Switching, Routing and Wireless Essentials

1.1.6 Repaso de SVI Configuration.

REMEMBER that by default, the switch has all the ports and the management VLAN assigned to the VLAN 1 but is best practice to assign a different VLAN for this purpose such as VLAN 99.

VLAN CONFIGURATION

S1(config)# interface vlan 99

S1(config-if)# ip address [ip-address][subnetmask]

S1(config-if)# ipv6 address [ipv6-address]/[subnetmask]

S1(config-if)# no shut

S1(config-if)# end

S1# copy running-config startup-config

Task	IOS Commands
Enter global configuration mode.	S1# **configure terminal**
Enter interface configuration mode for the SVI.	S1(config)# **interface vlan 99**
Configure the management interface IPv4 address.	S1(config-if)# **ip address 172.17.99.11 255.255.255.0**
Configure the management interface IPv6 address	S1(config-if)# **ipv6 address 2001:db8:acad:99::11/64**
Enable the management interface.	S1(config-if)# **no shutdown**
Return to the privileged EXEC mode.	S1(config-if)# **end**
Save the running config to the startup config.	S1# **copy running-config startup-config**

DEFAULT GATEWAY CONFIGURATION

S1(config) ip default-gateway [ip-address]

S1(config) end

S1# copy running-config startup-config

Task	IOS Commands
Enter global configuration mode.	S1# **configure terminal**
Configure the default gateway for the switch.	S1(config)# **ip default-gateway 172.17.99.1**
Return to the privileged EXEC mode.	S1(config)# **end**
Save the running config to the startup config.	S1# **copy running-config startup-config**

OJO------> To verify the switch's physical and virtual interfaces status you can use the

Show ip interface brief & S***how ipv6 interface brief***

1.2.6 Network Access Layer Issues\

```
S1# show interfaces fastEthernet 0/18
FastEthernet0/18 is up, line protocol is up (connected)
Hardware is Fast Ethernet, address is 0025.83e6.9092 (bia 0025.83e6.9092)MTU 1500 bytes, BW 100000
Kbit/sec, DLY 100 usec,
```

1 -----> The line protocol is up line refers to the data link layer protocol and whether or not it is working.

1.3 SECURE REMOTE ACCESS

1.3.4 Configure SSH

Step 1 → Configure the IP domain name

S1(config)# ip domain-name [domain]

```
S1(config)# ip domain-name cisco.com
```

Step 2 → use the crypto key generate rsa command to generate a RSA key pair and enable the SSH server on the switch.

```
S1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
```

Step 3 → To use the local (not other server) authentication method, create a username and password.

S1(config)# *username* [username] *secret* [password]

```
S1(config)# username admin secret ccna
```

Step 4 → Configuring the VTY lines which allow the switch to have remote access using Telnet or SSH

S1(config)# line vty 0 15

S1(config-line)# transport input ssh

S1(config-line)# login local

S1(config-line)# exit

```
S1(config)# line vty 0 15
S1(config-line)# transport input ssh
S1(config-line)# login local
S1(config-line)# exit
```

Step 5→ enable SSH version 2

S1(config)# ip ssh version 2

```
S1(config)# ip ssh version 2
```

1.4 BASIC ROUTER CONFIGURATION

Remember the basic Router configuration

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# hostname R1
R1(config)# enable secret class
R1(config)# line console 0
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# service password-encryption
R1(config)#
```

Configure a banner to provide legal notification of unauthorized access, as s

```
R1(config)# banner motd #Authorized Access Only!#
R1(config)#
```

Save the changes on a router, as shown in the example.

```
R1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Remember how to configure Interfaces

```
R1(config)# interface gigabitethernet 0/0/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
R1(config-if)# description Link to LAN 1
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface gigabitethernet 0/0/1
R1(config-if)# ip address 192.168.11.1 255.255.255.0
R1(config-if)# ipv6 address 2001:db8:acad:2::1/64
R1(config-if)# description Link to LAN 2
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface serial 0/0/0
R1(config-if)# ip address 209.165.200.225 255.255.255.252
R1(config-if)# ipv6 address 2001:db8:acad:3::225/64
R1(config-if)# description Link to R2
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)#
```

1.5 Important commands to verify interfaces

Interface status -----> *show ip interface brief* and ***show ipv6 interface brief***

Verify running config ----> ***show running-config***

Verify commands used on interfaces -----> ***show running-config [interface]***

To verify routes ----> ***show ip route*** and ***show ipv6 route***

2 Switching in Networking

2.1.1 Remember that a LAN switch always maintains a table that serves as a forwarding point for traffic. The LAN switch forwards traffic based on the ingress port and the destination MAC address of an Ethernet frame. So in general terms there is an association between MAC addresses and ports on the port table.

The switch creates a MAC address table and it is stored in the content addressable memory (CAM). This table is maintained, and the switch populates it by recording the source MAC of every device connected to the ports.

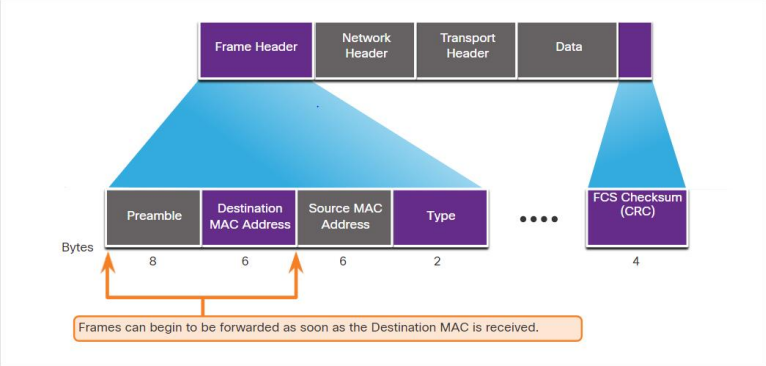
Whenever a frame enters the switch, this one is checked for new information to learn. It examines the source MAC address and port number (which is added to the frame whenever enters a switch).

If the source MAC and port do not exists on the switch, these are added to the MAC address table.

If the MAC address exists on the table, the switch updates both the refresh timer and the case the port is different, this one is changed as well.

A switch has 2 ways of forwarding frames. The switch has circuits called ASICs that reduce the frame-handling time.

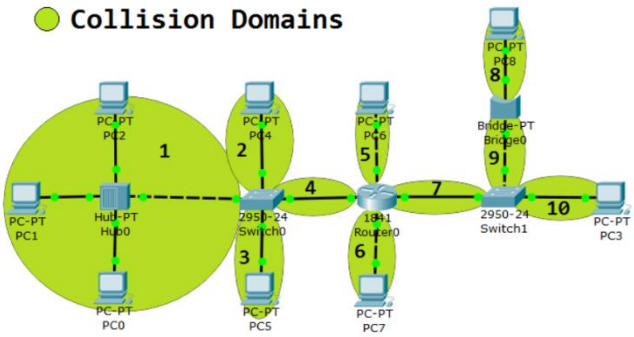
1.- Store-and -forward switching. This method uses CRC mathematical error-checking method to decide where to send the frame. This is the most common method. In the CRC method, the switch compares the FRAME CHECK SEQUENCE (FCS) value against it's own FCS calculations. If there is no physical or data-link errors the frame is forwarded.



2.- Cut-through switching à Less common method, simply forwards the frame after it check for the MAC address and port that it came from. Since this method does not apply the CRC method, it simply forwards every frame destined from a port to a destination MAC address.

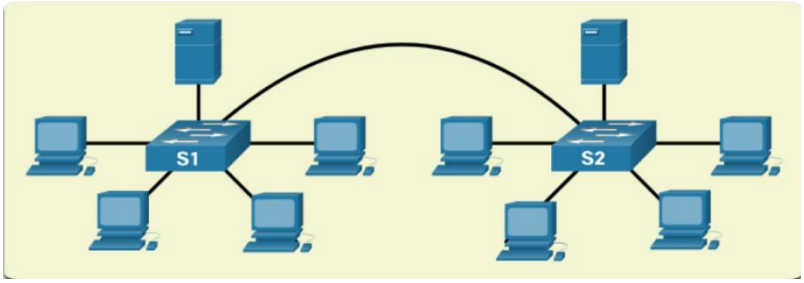
2.2.1 Collision Domains

REMEMBER that COLLISION DOMAINS are network segments that share the same bandwidth between devices and whenever 2 or more devices in such domain try to communicate at the same time, a collision happens.



If a switch port is operating on half-duplex, there could be a collision domain, but if a switch port is operating in full-duplex. Then no collision domain is created.

2.2.2 Broadcast Domains



Remember that a BROADCAST DOMAIN is created when 2 switched are interconnected. Only a network layer device, like a router, can separate such broadcast domain and it will also segment a Collision domain.

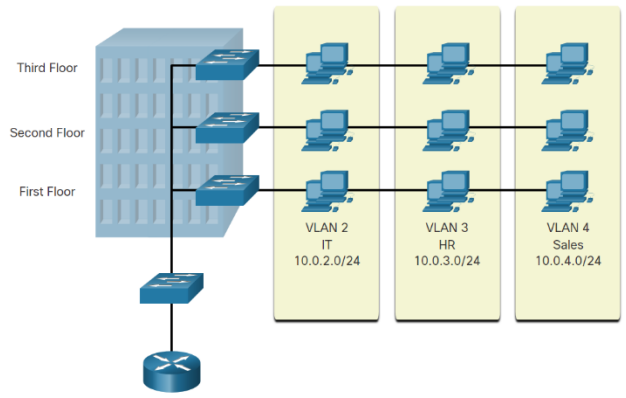
There exists broadcast frames that when a switch receives them, it forwards it to each one of its ports. These broadcast frames are used to locate other devices in the network and services. Keep in mind that if 2 switches are connected, the broadcast domain is increased, and therefor broadcast frames are sent across a larger network.

Since full-duplex by default prevents collision domains, switch ports running full duplex will always negotiate this first. They are required fro 1 Gbps ethernet speeds and higher.

3.- VLANS

3.1.1 VLAN definitions.

VLANS provide segmentation to a switched network. Think about it as if a group of devices connected to the same VLAN act as if they were connected physically to the same cable on the switch. Therefore, VLANs happen in a logical level instead of a physical level.



Each VLAN is considered a separate logical network and each device within such VLAN acts as if they are in their own separate logical network.

Unicast, broadcast and multicast packets are forwarded within each device included in the VLAN and packets that are not destined to any of the devices in such VLAN, must be forwarded for routing.

It is possible to create logical subnets within a network without the need of VLANS, nevertheless all devices connected to such subnets will still share the same broadcast domain and every broadcast packet not intended to be received by all devices, will still get it. Therefore, a VLAN creates a logical broadcast domain that can even surpass physical separation within the logical network (As seen in the image, VLAN 2 contains devices physically separated by switches, but still under the same logical subnet).

3.1.3 TYPES OF VLANS

a.- DEFAULT VLAN ----> On a Cisco router it is given the VLAN 1 name and from the start all ports are assigned to VLAN 1. Also important to mention that the navice VLAN by default is VLAN 1 and this one cannot be renamed or deleted.

b.- DATA VLAN ----> These VLANs separate users or computers into groups. It depends on the organizational structure.

c.- NATIVE VLAN -----> VLANs create tags that are applied to the frames to differentiate from which VLAN devices are communicating from. Trunk ports are used between switches to support this system of communication. In detail, an 802.1Q trunk port inserts a 4 byte tag in the Ethernet frame header to identify the VLAN from which the frame belongs.

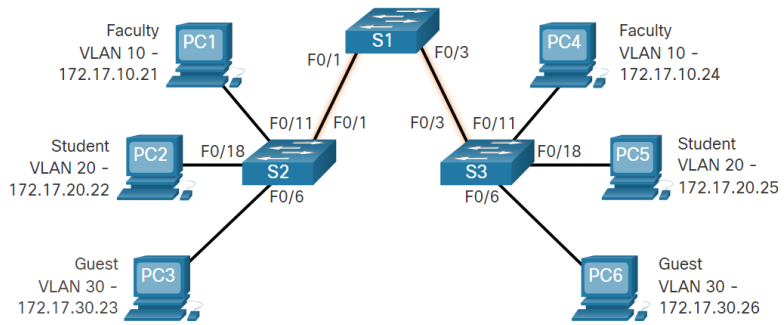
> →A switch may also need to send untagged frames across the trunk port. All of this untagged frames will be placed under the native VLAN.

d.- Management VLAN ----> VLAN dedicated for management that uses services like SSH HTTPS. By default VLAN 1 is confifured as the Management VLAN,.

e.- VOICE VLAN -----> In order to support voice over IP, a VLAN has to be dedicated to this purpose.

3.2.1 DEFINING VLAN TRUNKS

As stated before, VLAN TRUNKS enables VLAN traffic to be transferred throughout multiple switches. This enables different devices connected to the same VLAN but different switches to still communicate with one another. Important to mention that a trunk link can also be used between any 2 devices with a 802.1Q capable NIC.

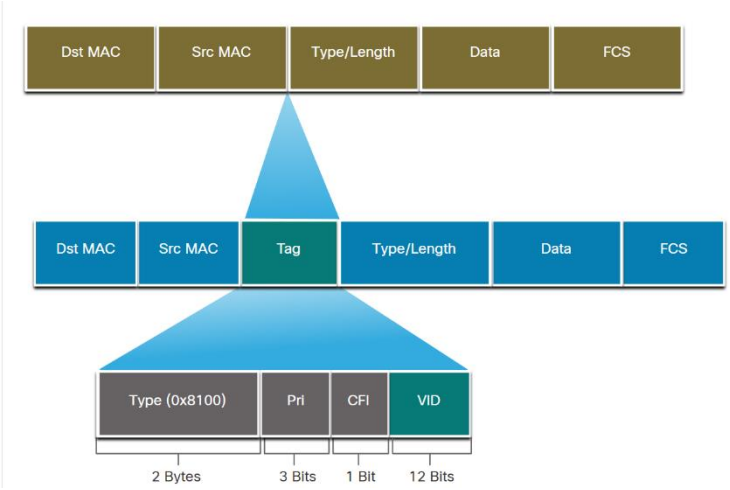


Imagine that this network does not have any trunk ports enabled or have VLANs. If any computer would send a broadcast frame out, it would be broadcasted to every single end point on the network, making it only 1 broadcast domain.

VLANs are configured on the switch's ports. Every device connected to these ports will belong to an IP sub network determined by the VLAN configuration given to that port. Therefore, a VLAN is equivalent to an IP network or subnet. Trunk ports are also configured to support specific VLAN traffic.

3.2.4 VLAN Identification with a Tag.

Whenever an Ethernet frame is created, it does not contain information on which VLANs device it comes from, this information is added on a trunk port. The IEEE 802.1Q HEADER allows ethernet frames to add VLAN tags identifiers. This 802.1Q header is a 4-byte tag inserted withing the original ethernet frame header.



3.2.5 Native VLANs and 802.1Q Tagging.

Trunk links need a native VLAN assigned so they know what tag to add to already untagged frames. Untagged frames are assigned to the Native VLAN. If by any case (sometimes devices supporting 802.1Q add VLAN tags to native VLAN traffic) a switch receives a tagged frame with the VLAN ID that is the same as the native VLAN, it drops the frame.

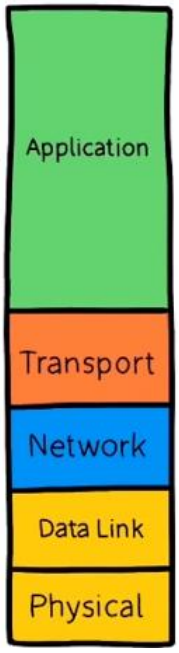
CBT NUGGETS GUIDE

CBT NUGGETS

Use Models to Understand IP Networking

Application and Transport Layers.

The TCP/IP model has 5 sections:



Different protocols and standards are run in either one of these layers from the model.

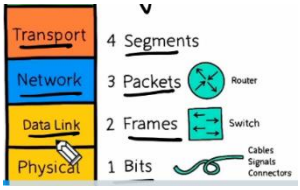
Application ------> Good way to think about is a sort of network function or service. HTTP/S and DNS run in the Application layer since the user interacts directly with a client (application) which is the browser.

Transport -------> UDP, TCP define what standard/protocol a network packet is going to be sent by. The application layer protocols like DNS or HTTP/s are associated with other protocols on the Transport layer. For example, HTTP requests use TCP protocol to communicate over the network.

Network ----> IP addresses come in here. When we want to communicate outside a LAN into a WAN, IP addresses are needed in order to be routed in the right matter. The mayor protocol here is the IP. IPv4 and IPv6 come in here both the source and destination IP addresses.

Data link ------> Mac addresses come in here.

Data in each layer is referred to as



LAB Instructions:

L3 IP Address → 10.10.0.50

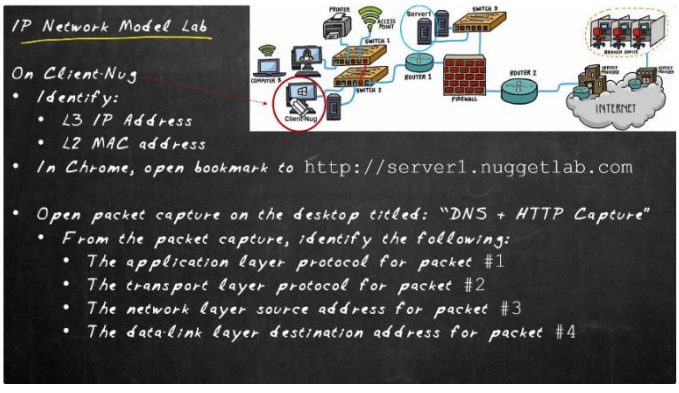
L3 MAC addres → `00-15-5D-00-04-08`

Application layer protocol for Packet 1 →

```
▸ Domain Name System (query)
```

Transport layer protocol for packet 2 → UDP

```
User Datagram Protocol, Src Port: 53, Dst Port: 64313
```



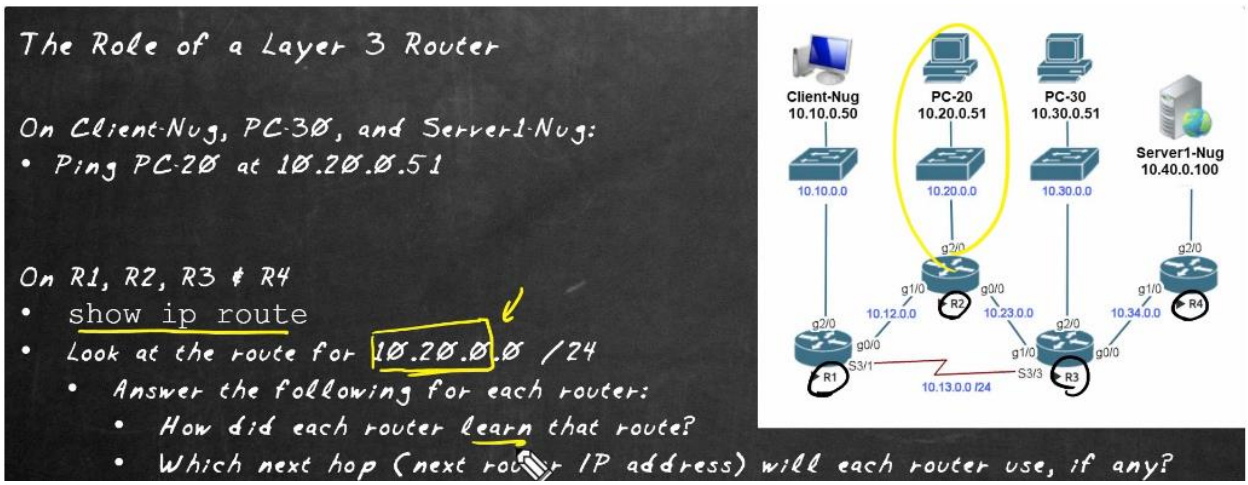
Network layer source address for packet 3 → 3 10.10.0.50 Datalink layer destination address for packet 4 → Dst: Microsoft_00:04:08 (00:15:5d:00:04:08)\

3.- Dynamic Routing protocols & static routing.

Routing tables are shared between routers. All the information is forwarded between routers. Some examples are: RIP, OSPF, EIGRP, BGP. Static routing protocols are a one-way-street. Most times static routes need to be manually input in the router. If a router needs to forward packets, they will know that it needs to either go outside the LAN or to another gateway.

Dynamic vs Static → Dynamic routing is implemented in networks consisting of more than just a few routers. Dynamic protocols are scalable and automatically determine the best route.

LAB



For router 1 →

Any traffic coming from 10.20.0 was learned statically and the next hop is G0/0 10.12.0.2
Any traffic coming from 10.30.0 was learned by OSPF
Any traffic coming from 10.40.0 was learned by OSPF

Fro router 2 →

The traffic from 10.20.0 was learned by local and directly connected and the next hop is G2/0

For router 3 →

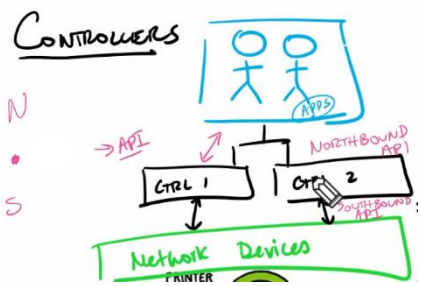
The traffic from 10.20.0 was leaned by EIGRP and the next hop is 10.23.0.2 G1/0

For router 4 →

The traffic from 10.20.0 was learned by OSPF and the next hop is 10.34.0.3 G1/0

4.- Use APs, NGFWs, and Controllers

A normal firewall has the capacity of reading through the traffic at a layer 3 level. A NGFW or Next Gen Firewall has the capacity of not only going through level 3 traffic but also at the APPLICATION layer. An IPS (Intrusion prevention System) serves as support to the NGFW that can analyze more in depth activities. NGFWs are capable of implementing policies.



A controller is a centralize management equipment. It provides visibility of all the devices inside a bigger network. It has a UI/Dashboard where you can manage all the devices. An API (Application programming interface) would come to be the interface for the user to control the application and the controllers. AT the same time the network devices have their own APIs capable of communicating with the above APIs mentioned. Famous controllers are Wireless Lan Controller (WLC) and DNA Center which enables you to manage virtually any network devices.

Also a WLAN controller (WLC) is helpful to manage multiple access points. They could be a physical controller or even virtual.

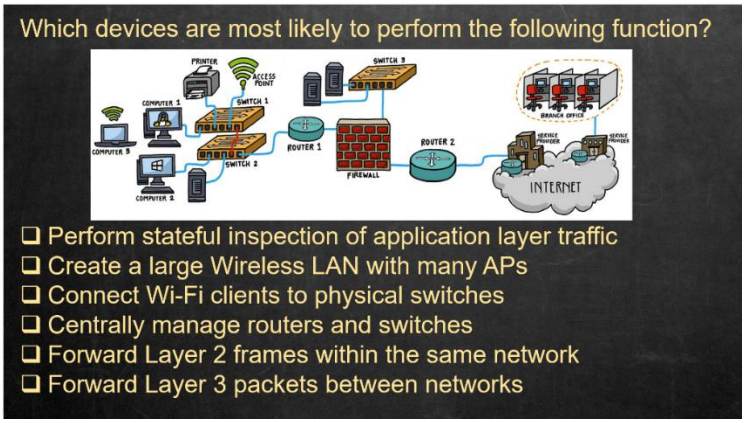
So Imagine we create several Wireless networks WLAN2/1. We would go to the controller and it will automatically configure all the Access Points.

ACCESS POINTS → Usually access points are built in a HOME ROUTER but in bigger physical networks, you can have them separately. A dedicated device connected to a switch port that allows wireless clients to connect to the network.

The Autonomous mode is the manual mode to handle and configure an Access point. Meaning setting it's SSID, checking for it's IP ADDRESS, etc/

Power over Ethernet (PoE) ---> 802.3af with 15.4W 802.3at with 30W and 802.3bt with 60W/100W As the name implies, power is transferred via ethernet to devices. These standards make "communication" between capable devices possible and help them determine the amount of power each need.

Lab/Validation



→NG Firewall

→WLC Wireless Controller

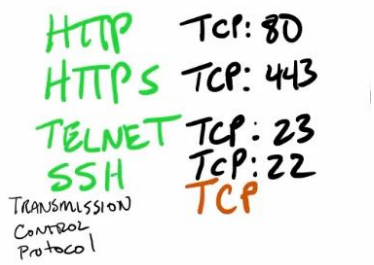
→Access Points

→Controller

→L2 Switch

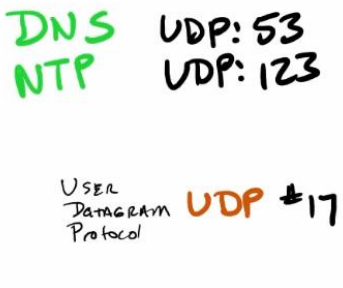
→Router

4.- **Use Layer 4 Transport Protocols**



Remember that 2 of the main protocols used at the layer 4 are: TCP AND UDP

Keep in mind that these protocols do not function alone by themselves. They need to be in conjunction with an application service that as well use transport layer protocols like HTTP/S or DNS or telnet/SSH.

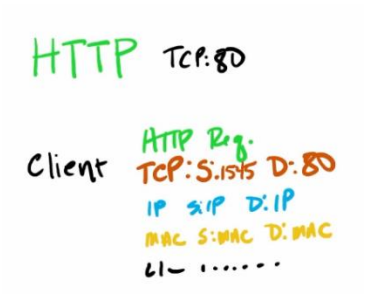


So remember that Application protocols NEED layer 4 transport protocols in order to communicate properly on the network.

The ICMP protocol although enclosed in an IP packet, it is considered as a L3 protocol. ICMP is a helper for the IP protocol. For example, the PING command instead of using protocols TCP or UDP, it uses ICMP.

Another example of a L4 protocol is the OSPF. A dynamic routing protocol that serves in the L4 transport.

When we talk about data being transmitted from a client to a server running at the application layer, we refer to as requests. For example, if a client is doing a HTTP request, it includes a source and destination port. In this case, HTTP request uses port 80.



The client needs a both a SOURCE PORT and DESTINATION PORT basically on every single layer.

So if the client is sending a request to a HTTP server that listens to port TCP:80 (HTTP), the client will need to send packets with a Source port in memory, let's say S:1545. Then it also needs to add an IP S and D (layer 3 header). At layer 2: S MAC and D MAC.

If the HTTP server responds back, it needs to return the request with the same info that it received the headers as. So, it will add all the Destination and source ports.

TCP VS UDP

Reliable and connection Oriented:

3 way handshake:

Packet 1 Client → SYNC

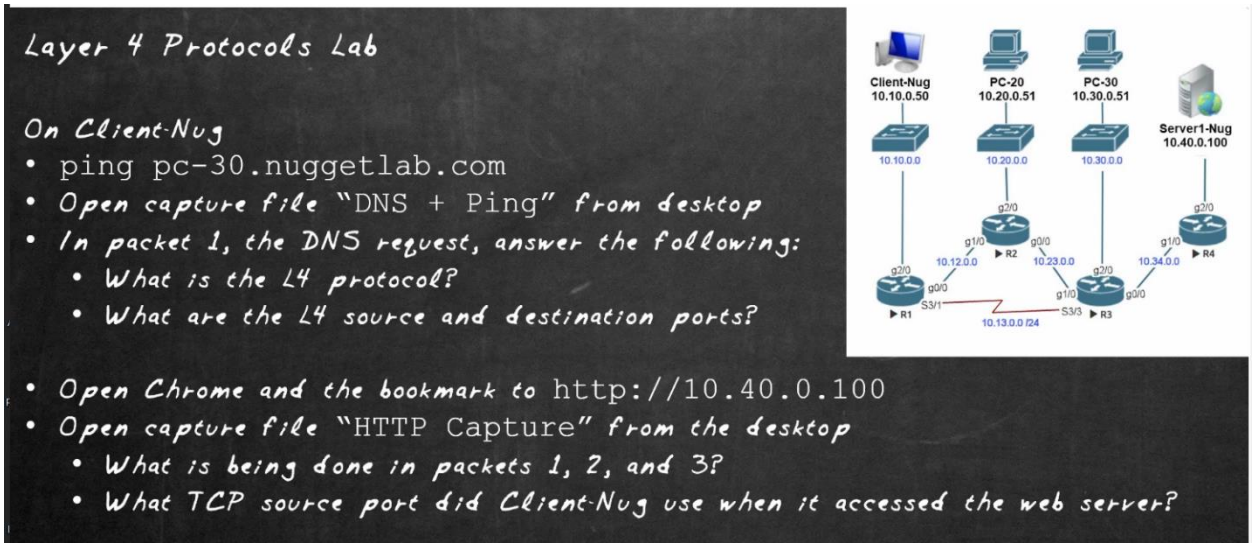
Packet 2 Server → ACK.SYN

Packet 3 Client → ACK

If data is lost, a RESEND is necessary.

This process goes back and forth, hence the reliability of the protocol.

Sends and receives requests without the 3WHandshake nor RESEND processes. Therefore, it is known as Connectionless and unreliable. But it is faster. Used for applications that do not need to lose time resending packets back and forth like voice calls.



LAB:

The layer 4 protocol in packet 1 is UDP

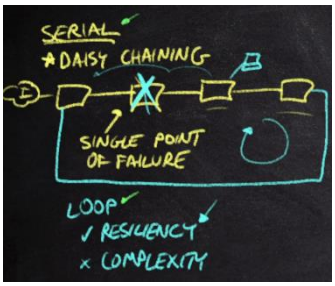
L4 source port → `Src Port: 57682, Dst Port: 53`

In packets 1,2,3 a 3 way handshake is being done.

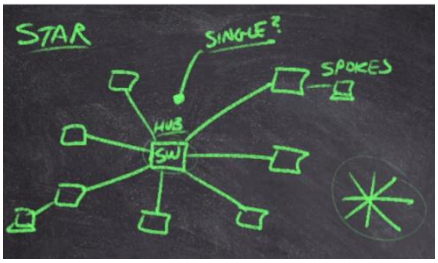
6.- **Describe the Hierarchical Network Model**

When building a network, there are 2 factors that come into play when the architecture does not have an intentional architecture: Scalability, Resiliency.

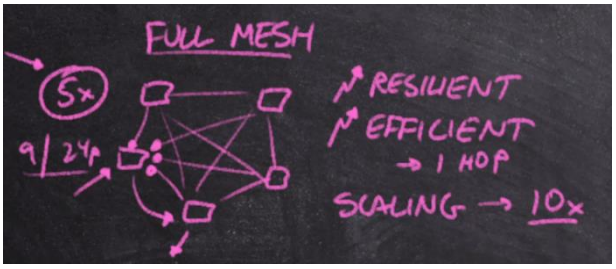
Basic Topologies:



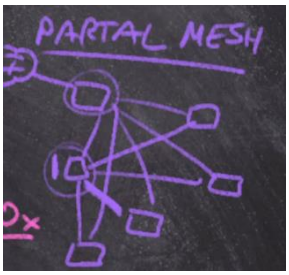
1. Serial → Network equipment connected in 1 line. Bad idea since if one of those equipment were to go down, the rest of the network on one side will not function either. One way to improve this, is to create a loop were RESILIENCY is created. On the other hand, complexity comes up as a negative aspect.



2. Star → With a main device (Switch) in the middle where other devices "Hang" on the outer edge. This topology is efficient. Between the 2 devices there is only 2 hops away. On the other hand, a SINGLE POINT OF FAILURE is created on the main hub, so it is not very resilient.



3. Full Mesh → Every device is connected somehow to every other. This creates a very RESILIENT (Since not a single device solely depends on another) and EFFICIENT network (Since every device is only 1 hop away from one another) but SCALABILITY is not good at all. The fact that when you need more devices connected, the full mesh becomes not doable.



4. Partial Mesh → Is a combination between star and mesh topologies. Getting the best of 2 worlds and a lot of practice comes in place when deciding how to connect your devices together.

The Hierarchical Network Model

We want a very Resilient, efficient and scalable network. That's the goal. So by combining the topologies, is the best way to create one with network BLOCKs.

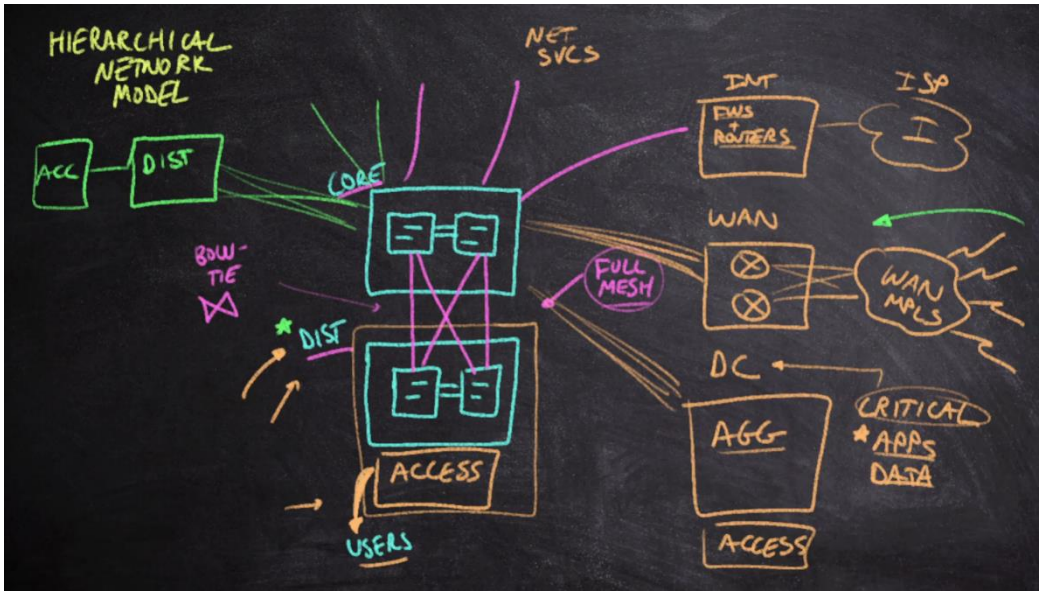
BLOCKS → Bigger blocks represent a set of network equipment, cables, LANS, etc with its own set of rules. There are a few of these blocks:

Internet block → made of firewalls and routers to be able to communicate with the ISP.

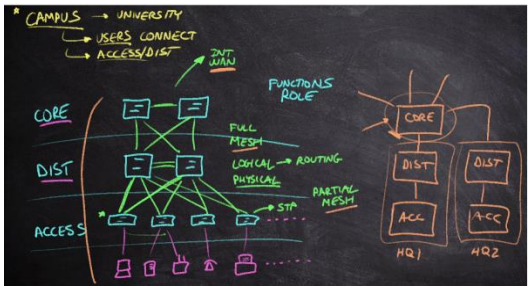
WAN Block →

Distribution block → It connects to an access block that ultimately connects to the users.

Aggregation → Stores apps and data so it is very critical to a network.



Enterprise Campus Networks



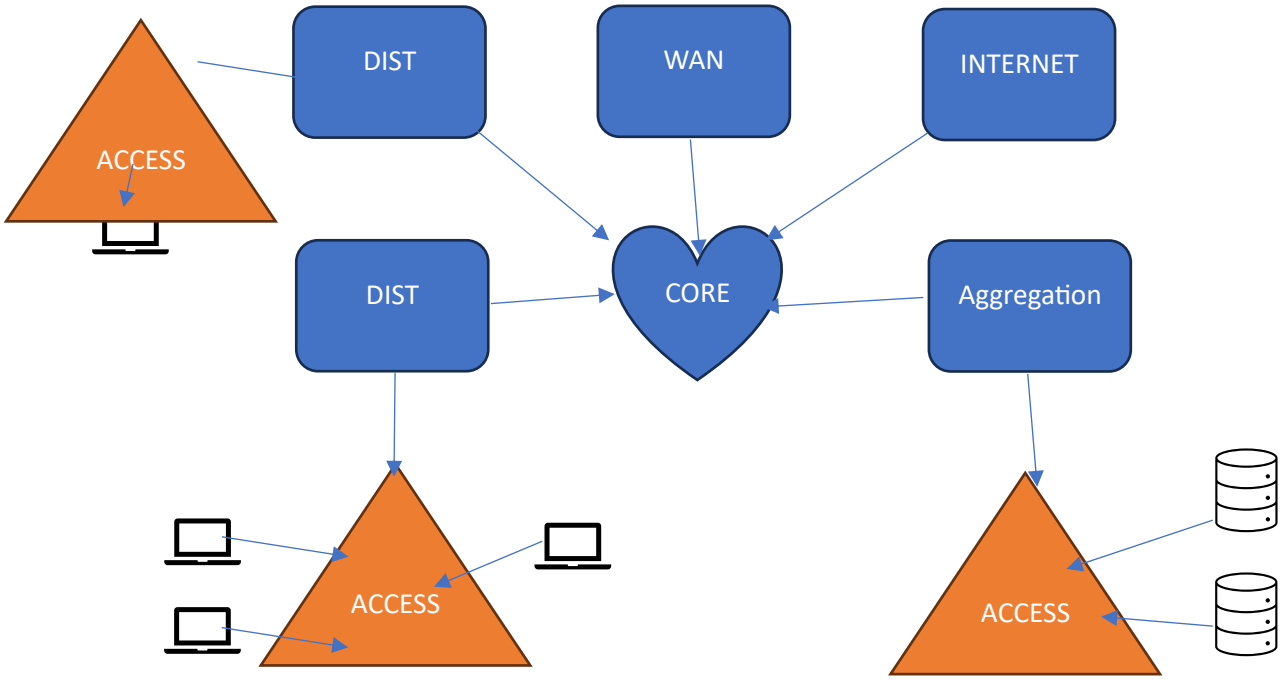
A campus in network is where users connect. It has an Access/Distribution layer. Remember that there are devices in each and every one of these layers serving a purpose with a particular set of rules. On the access layer, all of our endpoints will come in place. The access layer can have dosens of switches and Access points as well. The Distribution layer has several roles. In the physical realm, the distribution of traffic is necessary to the access layer. REMEMBER that these connect via a partial mesh topology and to the CORE layer via a full mesh topology.

On the logical side of things, routing comes in place when routing protocols are needed.

The CORE layer connects us out to the rest of the bigger network.

Validation

Starting with a network core, create a topology design that includes the following network blocks: Distribution, Aggregation, WAN, Internet, and Access.



7.- Describe the Hierarchical Network Blocks

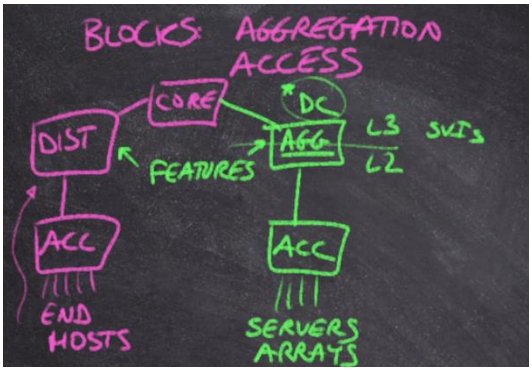
1.- ACCESS BLOCK → Where users connect. It is the primary block that we look at. We need switches where end devices are going to be attached to. As we connect the devices to an access layer switch, we will need to add identifications for the VLANS which is how we are going to separate our logical LANs on a network. Services are another important part of the access block, which is referred to as QoS. Security on Layer 1 is very important as well like shutting down ports that are not being used in a switch. On L2 security we can implement port security like the 802.1X standard.

2.- Distribution blocks → We would need L3 Services at the Distribution layers where routing and IP addresses come in place. Routing relationships upstream are created where routing protocols need to be running as well. We will also need to facilitate connectivity ports available for the Access layer devices.

As we need VLANs IDs in the access layer, we would need to facilitate the gateway for those VLANs in the name of VLAN INTERFACES or SWITCHED VIRTUAL INTERFACES (SVIs).

3.- CORE BLOCK → Since the CORE block is intended to connect to all the other blocks on the network, we will need to switch very fast on the L3 switching. We are performing both ROUTING and SWITCHING in here. We do not worry about QoS or VLANs or PoE. We need PERFORMANCE and BANDWIDTH. Sometimes these L3 switches are 800G fast! So these switches at the core layer or blocks are very Resilient to external physical problems. We also need a lot of available ports since ESCALABILITY is an important factor to consider.

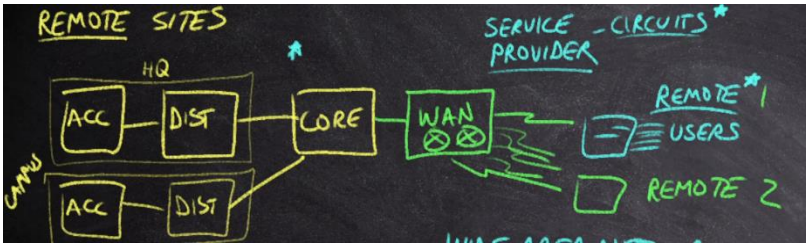
4.- AGGREGATION BLOCK → Data centers where data and apps are store are critical to the industry. In here, SERVERS are our best friends since they are big PCs, very powerful that run programs. Storage devices will also come in place here and this must be on a redundant basis so if some go down, other can continue to work. On the network side, Switches on the AGG block will need to be robust and high in performance and bandwidth to sustain high traffic.



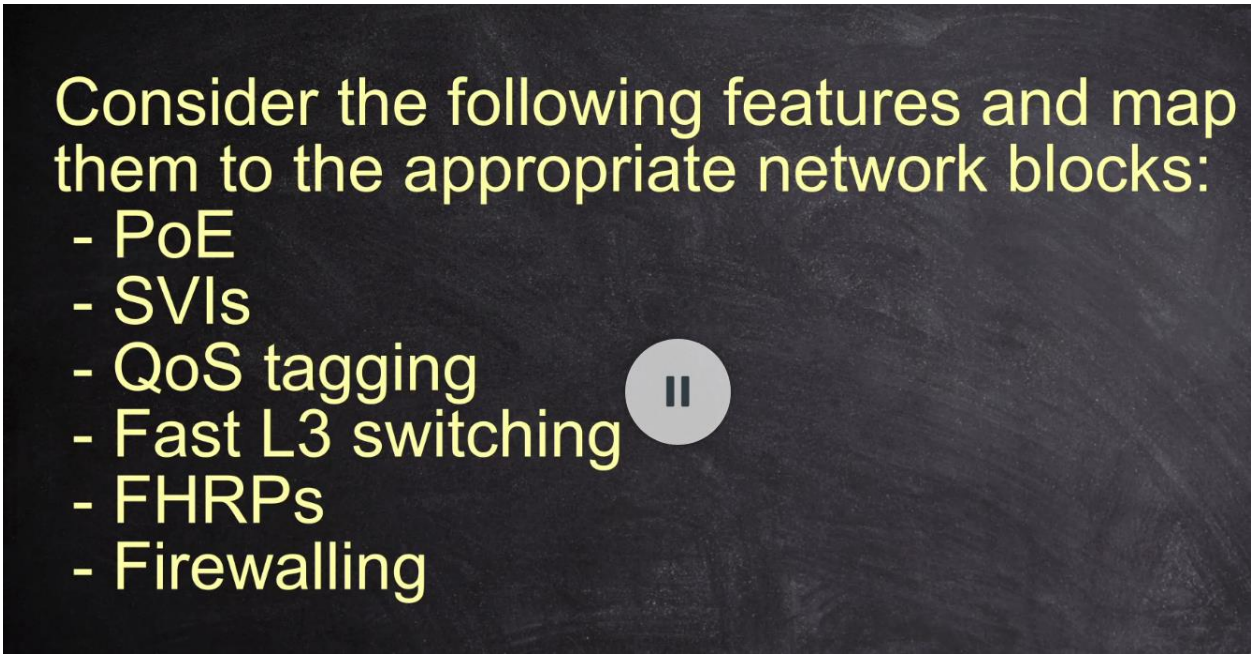
Keep in mind that conceptually, end devices are here as well, so logically an ACCESS BLOCK will also come in play side by side with the AGG BLOCK.

AGG blocks and Distribution Blocks are very similar. Both use L3 & L2 devices and protocols, Both connect to ACCESS BLOCKS. But the main difference relies in that the AGG BLOCK exists within the context of a data center and not primarily hosts, even though both are end devices in terms of networking.

5.- WAN BLOCK → Service Providers have circuits that run all across the world so Remote Users rely on these circuits to connect to the network. WAN blocks connect to the Core and provide connectivity to these Remote users. We are usually going to see routers for resiliency connecting to both the CORE block and then to the Remote users.



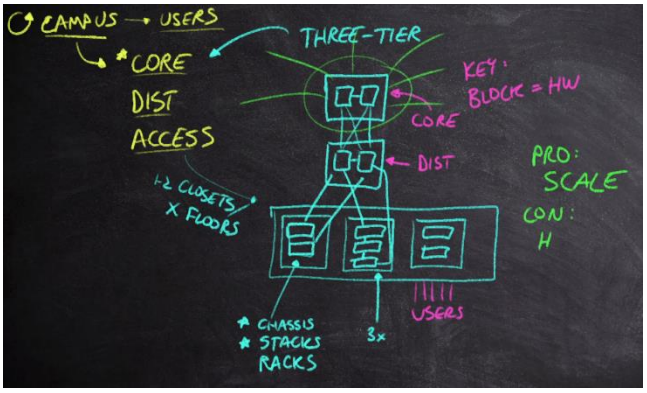
LAB/VALIDATION



- ➔ PoE: Access
- ➔ SVIs: Dist
- ➔ QoS tagging: Access
- ➔ FAST L3 SWITCHING: CORE
- ➔ FHRPs: Dist
- ➔ Firewall: Internet

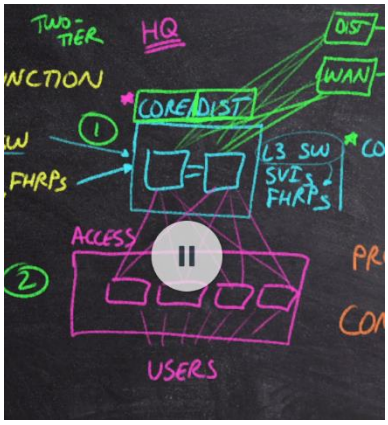
8.- Describe Network Topology Architectures

REMEMBER ---------- Enterprise Campus architecture → CAMPUS means 'Where' the users are located. With 3 primarily blocks here: Access, Core and Distribution.



1.- 3 Tier model architecture → A three tier model has all these 3 blocks working together. We'll have a couple of switches in the CORE and another couple on the DISTRIBUTION layer, all connected by a FULL MESH topology. We also have switches on the ACCESS layer. Each block in this architecture gets its own dedicated hardware; this is because we wouldn't want to give functionality from one block to another, that's the first reason why we have created these blocks to begin with.

A nice benefit of this architecture is that it SCALES pretty well, but then it would require a lot more hardware since functionality needs to be comprised in each block.



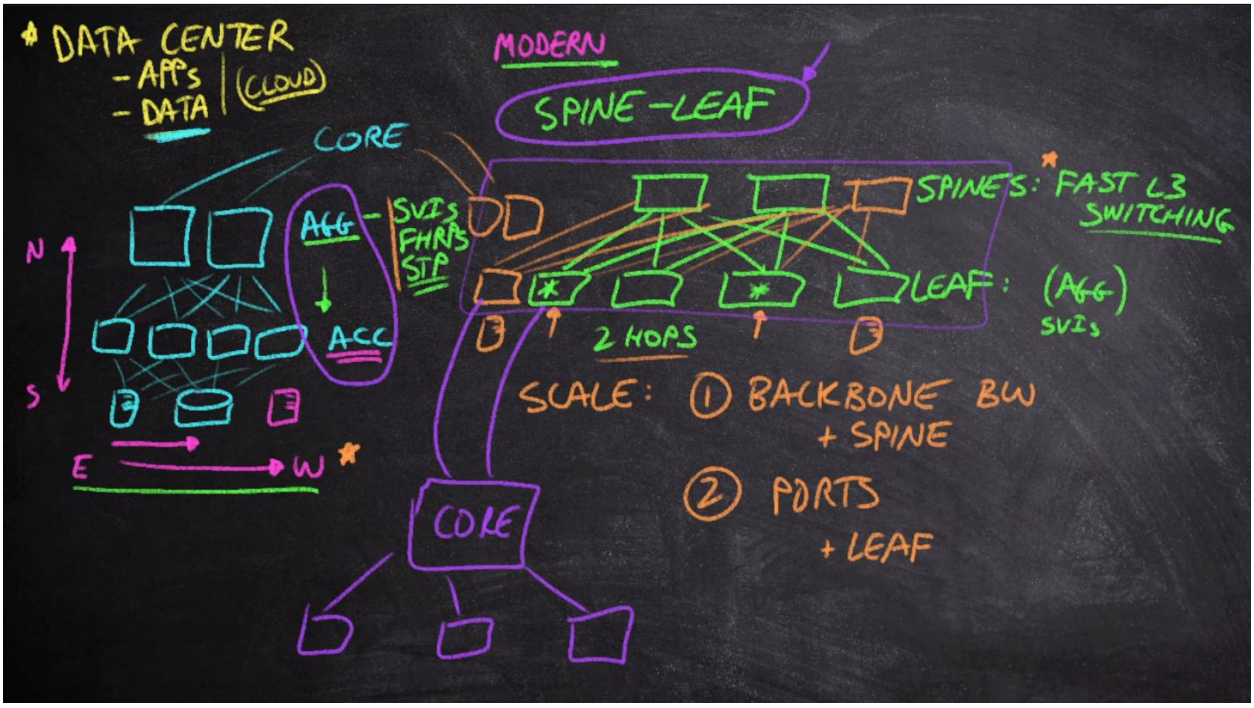
2.- 2 TIER architecture → Conceptually, a block is related to a network FUNCTIONALITY. Core switches need to be high in performance and robust. The distribution layer needs to deliver services like SVIs and First hand redundancy protocols (FHRP). But what would happen if we combine these functionalities into a COLLAPSED architecture with L3 Switches capable of handling all of this services and at the same time providing robustness and Resiliency.

In here, the CORE & DISTRIBUTION blocks are combined into one with the previously explained characteristics. This architecture has less hardware than a 3-way architecture and therefore reduce expenses but the SCALABILITY will not be as good since the CORE is needed to connect to all the other blocks in a network, and it is 'busy' working side by side with the DISTRIBUTION side, performance-wise will become a problem when it comes to scale our network up.

3.- SPINE-LEAF Architecture → Remember that the DATA CENTER is house to all the APPS and data that needs to be access consistently. A data center will need an AGGREGATION block with an ACCESS block as well. Each with switches, depending on the size, and for the end devices we have servers, and storage arrays instead of phones, pcs, etc.

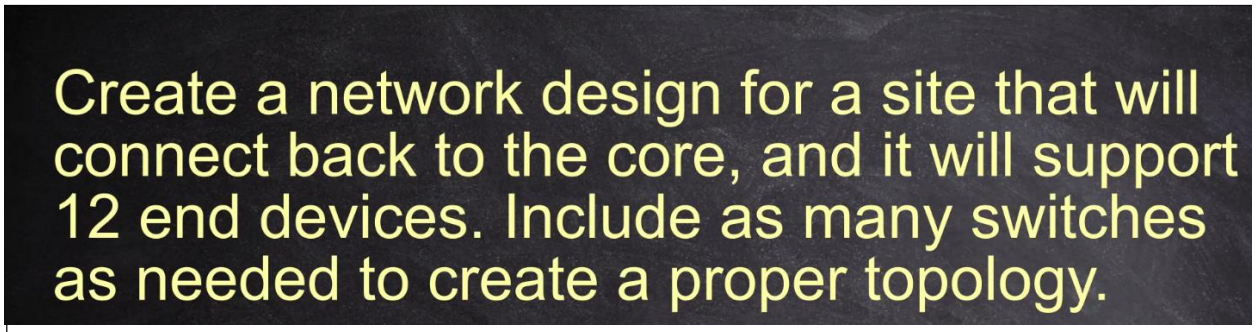
The SL arch is that one that has a spine layer which has CORE network L3 switching. This layer distincts itself from the AGG layer that serves more complex forwarding mechanisms vs the fast reliabel L3 switching on the SPINE layer. Then we have the LEAF layer that require another set of switches. We do not connect SPINE switches together and DONOT connect LEAF switches together. This way, we establish a rule of 2 hop distance between any 2 switches. This way we create a very good SCALABLE way for a DATA center to add hardware. This way, if we need more bandwidth, we'd simply add more SCALE switches; if we needed more ports, just add more switches to the LEAF layer. This architecture changes the game in

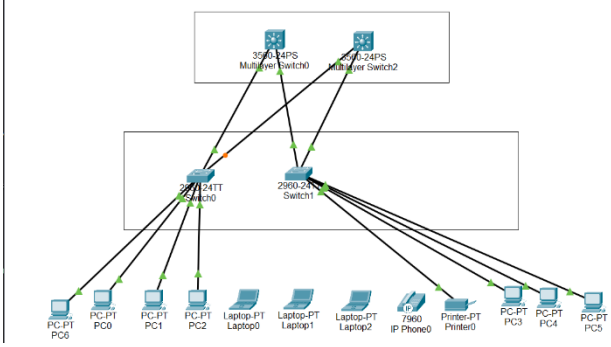
terms of how it can connect to the rest of the network. THE CORE block might connect solely to the LEAF switches.



5.- CLOUD & ON PREMISE architecture → What happens when we do not want to host our own data centers??? How do we make it happened between our own network to an offsite locations? Via a WAN CIRCUIT, VPN connections are also a choice through the Internet block.

LAB/VALIDATION



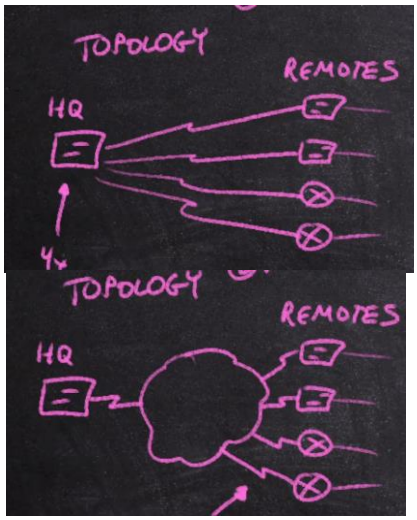


2 TIER architecture. Where the CORE and Distribution are combined in one and the access layer that connects the 12 devices together.

9.- Describe Wide Area Network (WAN) Technologies

WANs are used to connect remote site together into the rest of the block in the network. So how do you reduce the gap? A service provider builds out a service (network connectivity). They build out a network 'cloud' formation and the details of such network do not matter to the consumer.

Now on the Topology point of view there are a few things to consider:



1.- Point to point → Where our CORE block (HQ on the drawing), connects out individually to each remote site. But there is a big problem with scalability with this form.

For this, we use:

2.- Point to Multipoint → Where youi connect a single connection on your HQ to a single point and then another singel connection from each remote site. We represent this with the bolt connection. You can another connection for redundancy.

LEASED LINES

They are WAN circuits because we are leveraging circuit WAN technology. Some of these lines are built on old telephone networks. We used the bolt type connection to represent it or a cloud.

T1/E1 → Consists of phone lines bundled together that provides us with 1.544Mbps. The E1 provides 2.04 Mbps. They use RJ-45 connectors.

T3/E3 → We bundle a bunch of T1/E1 together. T1 provides 45Mbps and the E3 34Mbps.

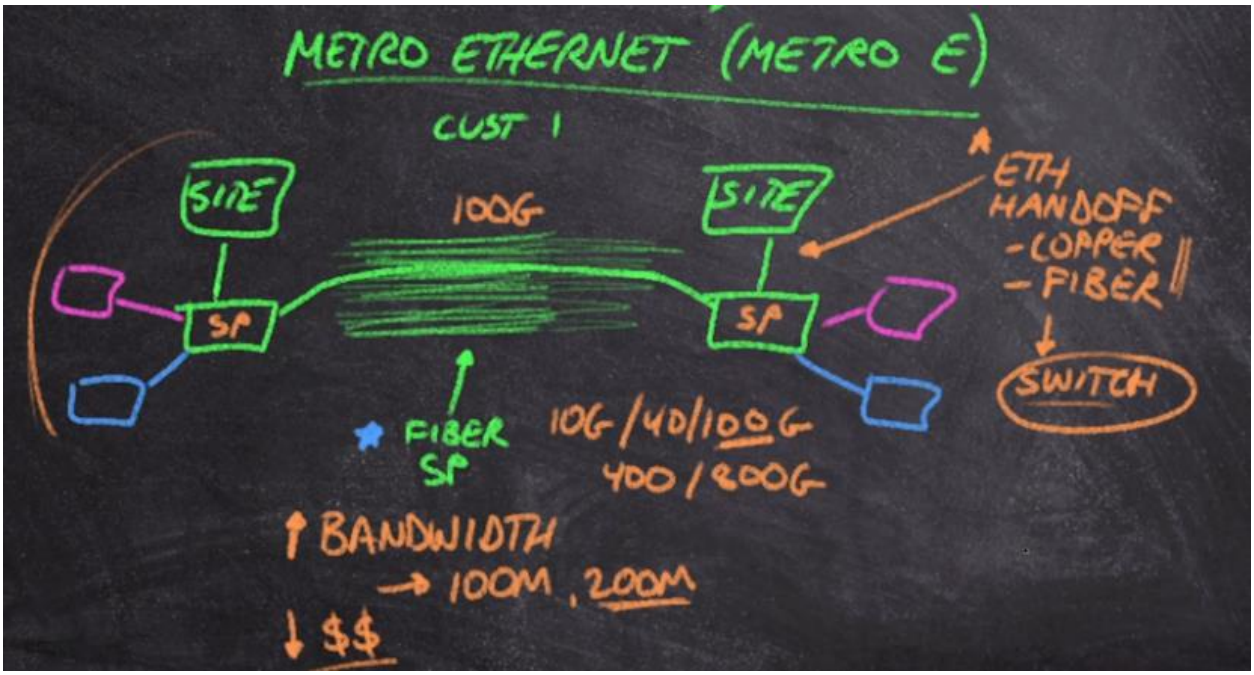
DISADVANTAGES → There are a few advantages of having a T1 line and it is that can be deployed anywhere. The downside of it can get pretty expensive. Another downside is that they do not provide a lot of bandwidth. Although they use RJ45, they do not use Ethernet protocol but another one called HDLC. For example, a home swithc uses Ethernet protocol and if connected to a T1 line/RJ45 it would not work. Then a router is used.

Ethernet Switching and Metro Ethernet

Since t1/E1 lines have several downsides, we want to push ethernet as far as possible into the users. A METROPOLITAN area network is leveraging short distances and start putting fiber optic into further distances.

What happens that is interesting is that service providers deploy a huge number of fiber cables from site to site and when needed to connect remote locations together, they can simply connect 2 switches each in one end. So they deploy ethernet connection for each site but the main distance is covered by underground fiber optic cables.

The connection betwenn sites and the switches are called ETH HANDOFFs that can both deploy copper or fiber for the costumer's needs. Now we can deploy huge amounts of bandwidth that serves as an advantage. Fiber can run at 10/40/100/400/800 Gigs. Fiber connections come at a lower cost as well.



MPLS (Multiprotocol label switching)

Now there is a problem of SECURITY and LOGISTICS when having to assure that packets destined to certain costumers need to arrive to their designated location instead of somewhere else.

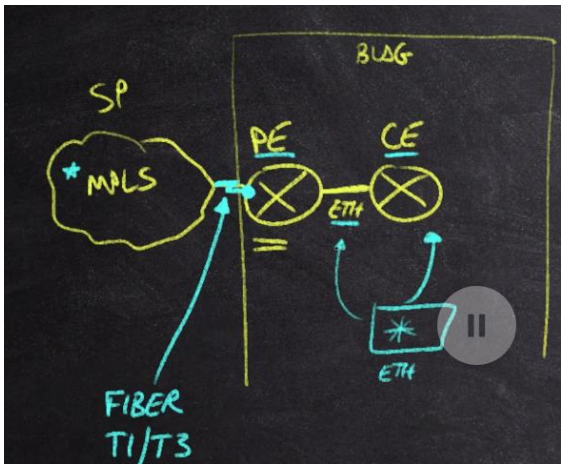
We could fix this problem with VLANs theoretically, but when a SP has tens of thousands of customers, you can not assign that many amount of VLANs because they simply do not exists.

Scalability in this case to provide privacy and simplicity will be MPLS. MPLS is a technology that uses labels to tag traffic.

1.- The Name of service →

2.- This exists at layer 3 → You send your routes to the SP. Meaning it works with routing protocols. The SP has routers participating in these advertisements using routing protocols to make sure proper routes are learned properly.

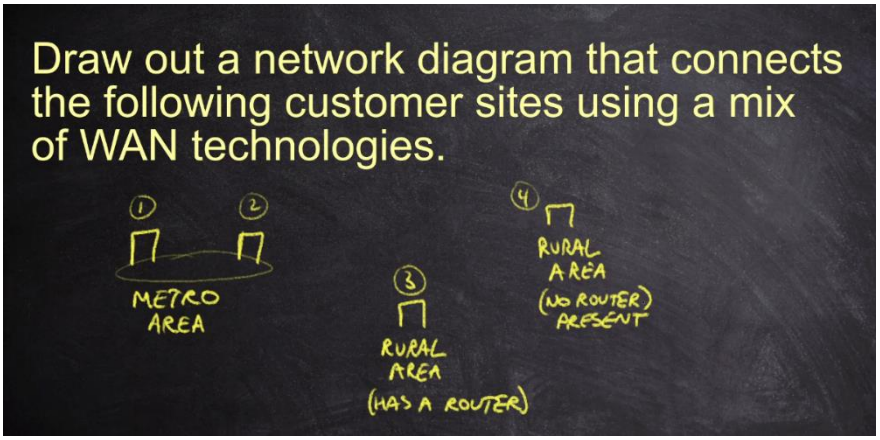
This all means that MPLS requres sharing routes wih the Service Provider.



The service provider will deploy routers at the costumers locations and enable routing routes via a routing protocol (usually BG or OSPF).

This routers that connect to the costumers are called a PROVIDER EDGE ROUTER and are sitting at the customers end. The way they connect this router back to their service centers can be Fiber or T1/T3. The costumer will always see an ethernet technology running and will not mind what technology the SP is using.

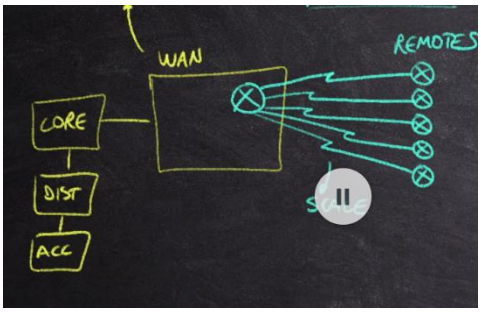
VALIDATION/LAB



10.- **Describe WAN Topologies and Connectivity**

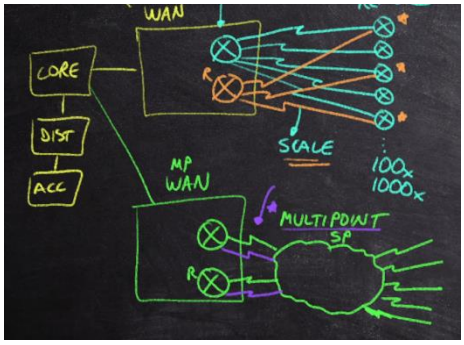
As described in the previous topic, there are 2 main WAN topologies, the Point to point and multipoint topologies. PtP provides a dedicated connection out to each remote site while a mulipoint connects all locations into the same network.

So imagine that we have all our blocks previously described. CORE, DIST AND ACC, including a WAN block. If:



PtP → Usinf individual connections from site to site. Imagine several emote sites that run T1 into our WAN block. But as said before, SCALABILITY is a huge problem here.

Multipoint → Where we create a network from rthe SP pov. They allow each individual site connect to this network using MPLS technology. A big advantage is that we do not need to worry about SCALABILITY.



Lit and Dark Fiber

Both use optical signals. A lit fiber service is shared between costumers deployed by the SP. A dark fiber connection is when the costumer itself lits up the fiber cable instead of the SP

Wireless Bridging

This is a bridge between 2 different networks using wireless technologies. This is used in locations that do not have any wired connections. So what you do is deploy a wireless bridge and connect those to an Access point or switch using a switch. This could be a PtP but can also be deployed in a Point to multi point fashion. This deployment is low on cost and easy to deploy. But it is not as reliable as a wired connection and the throughput as a wired connection.

L2 WAN Links

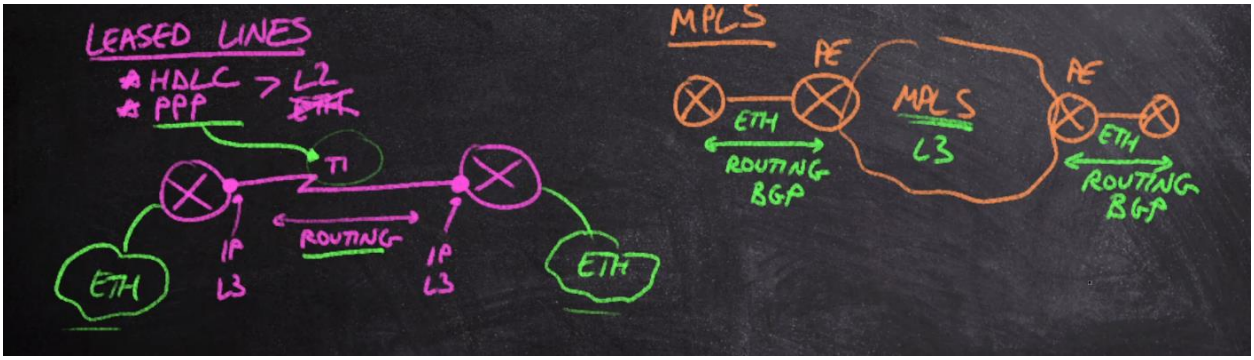
Remember that when we talk about L2 connectivity, we need to understand that norouting is occurring. So imagine we are deploying a L2 WAN link where we have switches connected with each other. This means that the devices can theoretically be part of the same subnet denoted by a VLAN. This happens

often when connecting DATA CENTERS. 2 Switches connected together with VLANs running can be programmed as trunk links but will this work when deployng WAN networks?

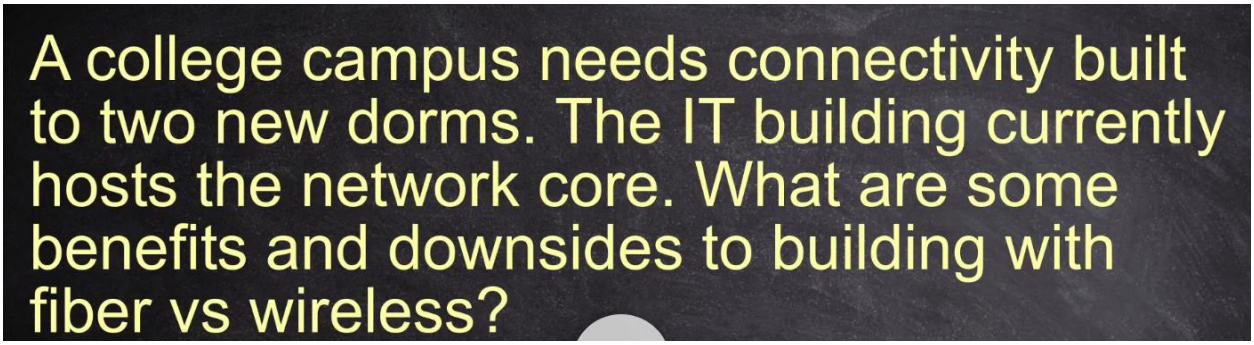
Most SP will not support this, but will actually deploy routers or L3 switches to supports routing and start using IP addressing.

L3 WAN Links

A L3 WAN link will use something else other than ethernet. Which makes sense given that L3 operates with routing protocols instead of L2 technology like ethernet. Remember that when talking about WAN deployments, we talk about 2 mayor ones like a point to point connection in lets say T1 line and an MPLS that rely on an external network from the SP. In the case of having leased lines on a PtP basis, we are going to have to route from one router to the other. The same happens with MPLS as a WAN technology. In this case, they deploy provider routers where connections are deployed with which enables us to only create a routing relationship with the PE routers.



LAB/validation



When building with FIBER one of the main advantages is scalability. Wireless deployments tend to be easy to deploy but are not as reliable nor scalable as fiber in this case. Fiber is also more expensive since we would need the infrastructure and the bandwidth is larger.

11.- Identify Copper Cabling and Termination

Signals Sent over Copper

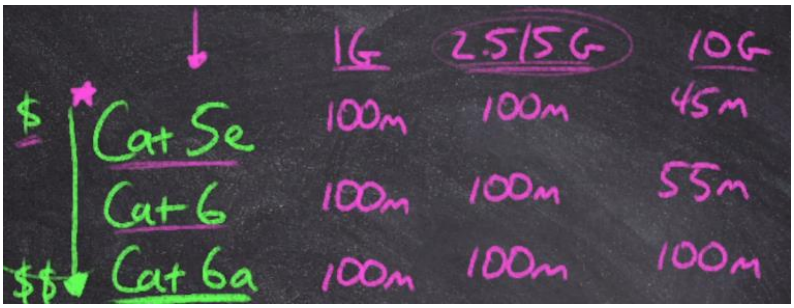
Signals sent over copper are electrical signals.

Shielded STP → It helps with interference. But must be properly grounded.

Unshielded UTP → It is in most cases the cable option seen more often.

Ethernet Cable Types

Ethernet has a Max distance of 100m. But the farther away, the lower the speed. So quality is a big important aspect. The better the cable, LESS CROSSTALK AND ERRORS will occur.

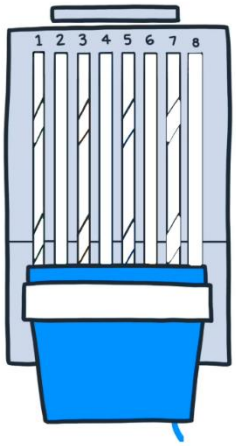


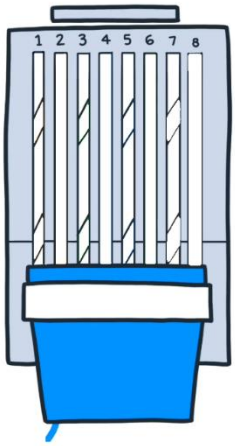
Ethernet Pinouts

A straight through cable is when both ends have the same standard T568A or T568B. A CROSSOVER cable is when each end is one of each. In both we have 8 pins, paired together in 2 pairs each.

A → G/G/O/B/B/O/Br/Br

B → O/O/G/B/B/G/Br/Br

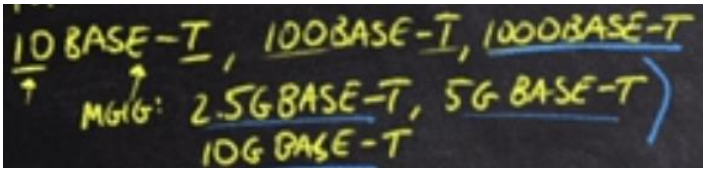




A → G/G/O/B/B/O/Br/Br

B→O/O/G/B/B/G/Br/Br

12.- Identify Copper Interface Transmissions



Straight-through cable → Ethernet cable that terminates on both ends in the same type. Remember that when we talk about twisted pair cables, they work with 4 pairs where one pair is used for TRANSMITTIN (TX) and one pair is used for RECEIVING (RX). There are rules that devices follow to know how to communicate through the cable when connected. These rules are:

MDI → We transmit in Tx: 1, 2 and Rx 3, 6

MDI-X → We Receive Rx: 3, 6 and Tx: 1, 2

So devices like a PC and a router use MDI but switches use MDI-X

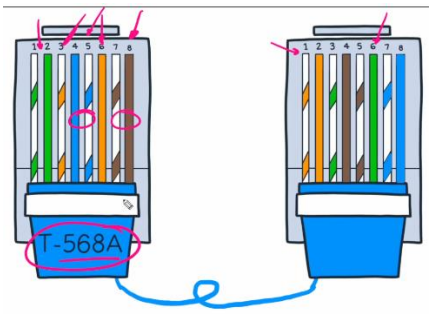
These 2 pairs are used on 10 and 100 Mbps but when it comes to Gigabit ethernet, the rules change where Transmitting and receiving are done on only 1 pair. Tx+Rx on a single pair.

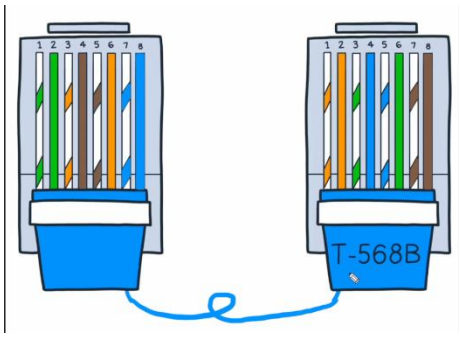
To summarize, different devices use one of these 2 standards. MDI and MDI-X. If we want to connect two devices that have different standards, we use a STRAIGHT-THROUGH cable and devices with same standards, we use CROSSOVER cables.

Crossover Cables

A Crossover cable has different terminations on both ends. Now, AUTO-MDIX is a feature that detects an error in cable usage and IF the device is running autoMDIX it will change the standard used. Starting at 1Gbps cables, AUTOMDIX is required, so crossover lacks importance.

But since GIGABIT requires all pairs functionality, if the need of a crossover cable comes, we would need to also swap the rest 2 pairs. So a crossover cable for a GIGABIT standard, would look like this. For type A and B





Console/Rollover Cables

It has a different pinout than T-568 A,B standards. Basically pin 1 – 8 connect to 8 – 1 on the other side.

T1/E1 and T3/E3 Cables

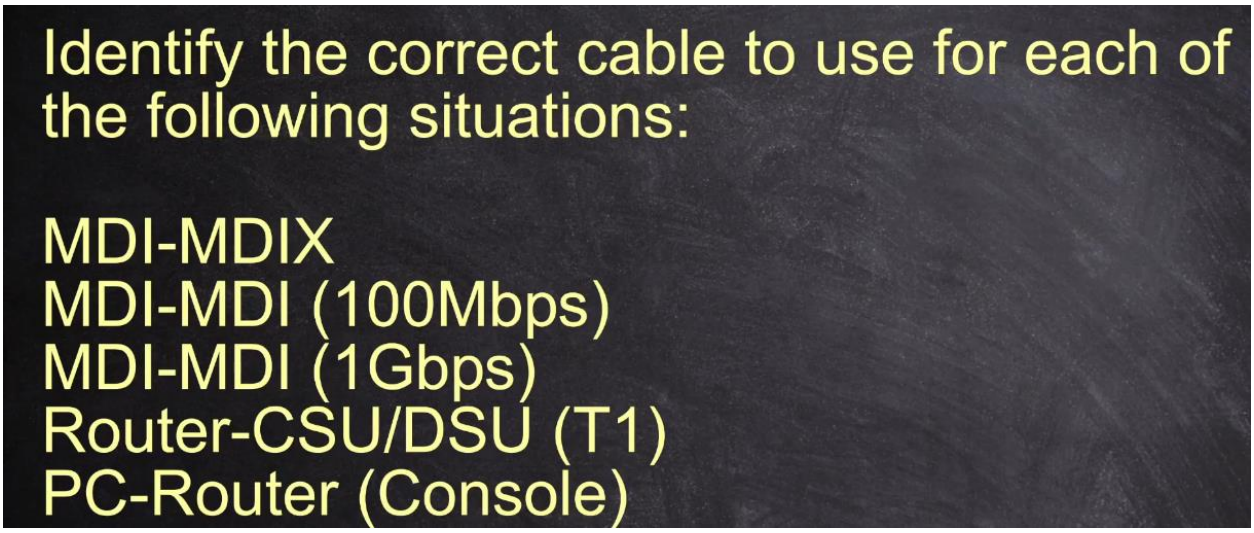
So far we have seen COPPER cables using RJ45. T1/E1 use copper and RJ45 as well. They only use 2 pairs even though they have 4 pairs just like the Tbase10 – 100M standards.

Tx 1, 2

Rx 4, 5

So we go back to the same concept, when wanting to connect two different devices running through a T1 cable like a ROUTER to a SP CSU/DSU port The Tx pins and Rx will be different. For example a SP port will have Rx 1, 2 and Tx 4, 5.

When wanting to connect two devices that are the same like 2 routers, we use a crossover cable that has both pairs Tx and Rx on their respective numbers.



1.- Using different standards means different devices, so a STRAIGHT THROUGH

2.- Same standard same devices, CROSSOVER with autoMDIX required

3.- Same standards same devices but at 1g+ means SPECIAL CROSSOVER A/B

4.- Using a T1 line from different devices means a STRAIGHT through

5.- Different devices to the CONSOLE rollover cable.

13.- Identify Fiber Optics and Cables

There are 3 main components to a FIBER OPTIC CABLE:

Core → Transmits the light

Cladding → protective barrier around the core that prevents light from escaping.

Jacket → Case that protects both sections of the cable.

Optical Transmissions

Optic cables can carry much more bandwidth than copper (Going up to 800Gmps). They can also run way longer (220m) compared to copper (100m).

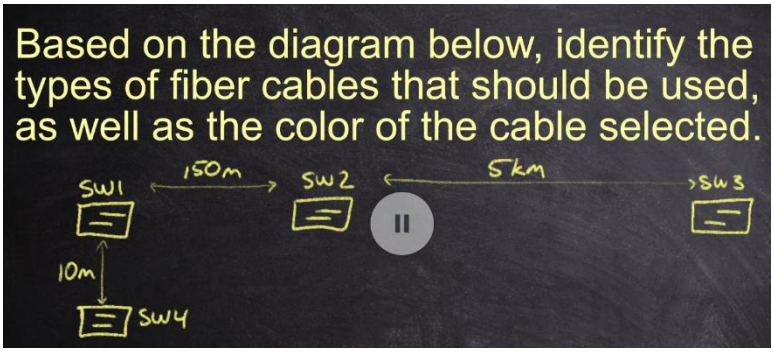
Since Copper works in pairs, fibers work with STRANDS to send and receive signals.

Single-mode Fiber (SMF) and Multimode Fiber (MMF)

Multimode is primarily used to deploy for shorter distances while single mode for longer distances.

Singel mode fiber has a YELLOW jacket and has a distance of 10 Km/40/80Km which is the limit of CISCO but can even go further. They tend to be pretty small with the core being 8 – 10.5 Micro meters. The Cladding would be 125 Micrometers. There is only one type of SINGLE MODE fiber called G.652

MULTIMODE tend to be ORANGE or AQUA. There are multiple types of Multimode fiber. OM1 – 5 being the last, the better. Multimode fiber have an element of Quality to them that increase the higher the number of the type. Distances also increase with the number. When we get to OM4 it can get to 400meters.



The first one would be a Multimode probably of OM2 since the distance is short. AQUA

The second one could still be a Multimode one of quality OM3,4. AQUA

The last one would be Single Mode. ALWAYS YELLOW.

14.- Identify Transceiver and Cabling Types

Long Range Transceivers

So far we have seen that we have Single mode and multi-mode fiber. Long Range Transceivers are used with Single Mode fiber. We see these as Lx, Lr, Lh.

A transceiver on a long

15.- Identify Cisco Transceivers and Compatibility

Transceivers come into 2 different types of technologies:

1.- SFP with three possible subcategories

SFP → 1 gbps

SFP+ → 10 Gbps

SFP28 → 25Gbps

SFP56 → 50Gbps

2.- QSFP Technology which leverages 4 streams of data and are bigger.

QSFP → 4 Gbps

QSFP+ → 40 Gbps

QSFP28 → 100 Gbps

QSFP56 → 200Gbps

QSFP-DD/800DD → 400/800 Gbps

16.- **Explain Ethernet Structure and Transmissions**

The ethernet frame is measured as Bytes. Data inside the ethernet frame is called the PAYLOAD. Before the payload comes a HEADER and then a TRAILER. The Ethernet frame has a maximum size of 1500 Bytes and the minimum is 64Bytes.

Ethernet Standards

Ethernet was created in the 70s by Xerox. Modern ethernet is managed by IEEE and started with 10Mbps called 802.3

Ethernet has evolved into 802.3u with 100Mbps → 802.3z 1Gbps FIBER → 802.3ab 1Gbps COPPER → 802.3ae 10Gbps FIBER / 802.3an 10Gbps COPPER → 802.3bz etc.

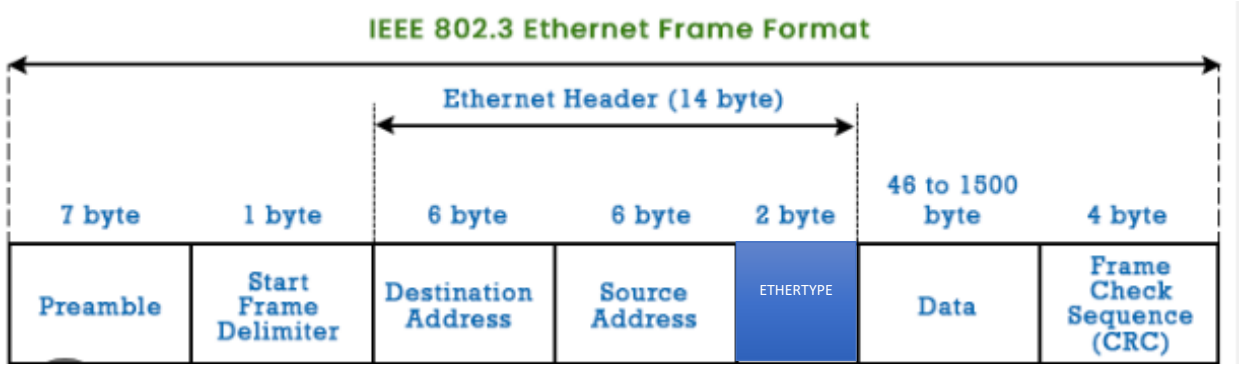
Layer 2 Ethernet Frame Structure

As explained above, an Ethernet frame has 3 main parts: The header + payload + trailer.

Frames in networking need structure which comes in terms of FIELDS so the application and network devices can interpret the data inside the frames. On an ethernet frame we have:

HEADER→ in the header we have 3 FIELDS DEST MAC/SOURCE MAC/ ETHERTYPE which provides us with information about the payload like the protocols used on L3.

TRAILER → Made by 1 FIELD and is made of the FCS or FRAME CHECK SEQUENCE that consists of a math problem called CRC. The purpose of the FCS is to run an error check detection to confirm that the data arrived correctly. It is basically a HASH.



Ethernet Transmissions

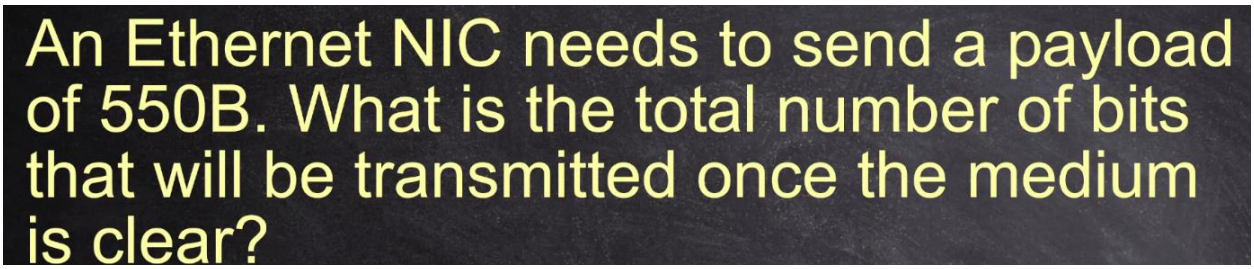
Before sending ethernet frames into a network, the NIC needs to create the preamble and other steps:

1.- Link detection → Assure that no one else is transmitting. The 'Medium' is the mechanism of transport, for example fiber optic or copper cables, wireless, etc. The MEDIUM is only going to accept 1 Tx (transmission) at a time. The way NICs do this is by sending 7 times a preamble consisting of 10101010 so when the receiver gets this information, it knows that the NIC is ready to send frames. Now, when the receiver is ready to receiver the ACTUAL frame, the SFD (Start Frame Delimiter) is added with a 10101011.

Half-Duplex and Collisions

Ethernet was created as a HALF-DUPLEX technology which means only 1 SENDER at a time (Like a walkie talkie). Meaning devices can only Tx OR Rx.

LAB



4608 bits

17.- Explain Ethernet Communications

CSMA/CD

When CSMA/CD rules are established, mediums need to:

1.- Listen if there are any one Tx? If they are, then they wait to send frames, if not:

2.- Start Tx → Start sending frame.

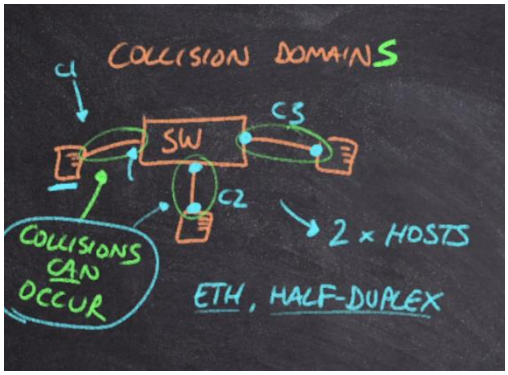
3.- Detect for any collisions → If there are collisions on the network happening, the least we would want is to send more traffic.

> Devices detect collisions by LENGTH and DISTANCE (1500 Byte and in the case of copper 100m). Through math, devices calculate how long they should take between sending and stopping between frames so others can start sending.

If there is a collision, devices let all other in a network know that there is a collision through a JAMMING SIGNAL so everyone stops sending frames.

Collision Domains

Collisions occur at L1. COLLISIONS DOMAINs are where collisions can occur. Collisions are more likely to happen with more hosts in a network. The same CSMA/CD rules apply, only one Tx at a time. Since HUBs are less intelligent than switches, they create one big collision domain when used to connect multiple devices.



Different to HUBs, SWITCHES are more intelligent and because of this, among other technologies, L2 protocols/rules/standards start playing a role. A switch is capable of reading MAC addresses (DEST AND SOURCE) and sending it through the particular interface where the DEST MAC is for. Now, COLLISION DOMAINS in this scenario are reduced and are created within every connection to the switch. In this case, a collision can occur between the host and the switch. If the hosts are running ETH in HALF DUPLEX operation, then collisions can occur although are reduced.

Half-duplex and Full-duplex Operations

We have talked about HALF-DUPLEX operation as being technologies when 2 devices communicate to each other by only Tx and Rx one at a time. This brings up many set of rules and concepts like CSMA/DC and COOLISION DOMAINS since we need to make sure that devices can still communicate between one another.

REMEMBER the characteristics of Half Duplex:

- ➔ ONLY 1 Tx
- ➔ Only able top Tx or Rx at a time
- ➔ BW is shared (Ex. 1000 devices with 10Mbps = 10Kbps per device)

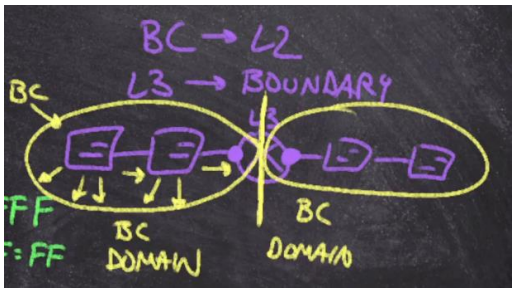
But HALF DUPLEX depended a lot on only 2 pairs being used. 1 for receiving and 1 for sending, but when FULL DUPLEX comes into play, we have 4 pairs now where 2 are sending and 2 receiving at the same time. This means that devices can now send through 2 pairs while the other is receiving through their 2 pairs and vice versa. FULL DUPLEX's characteristics:

- ➔ Many Tx at a time.
- ➔ You can Tx + Rx at the same time
- ➔ BW is dedicated to each device (Ex. 1Gbps can be used at the same time by everyone since everyone is doing 2 actions at the same time.)

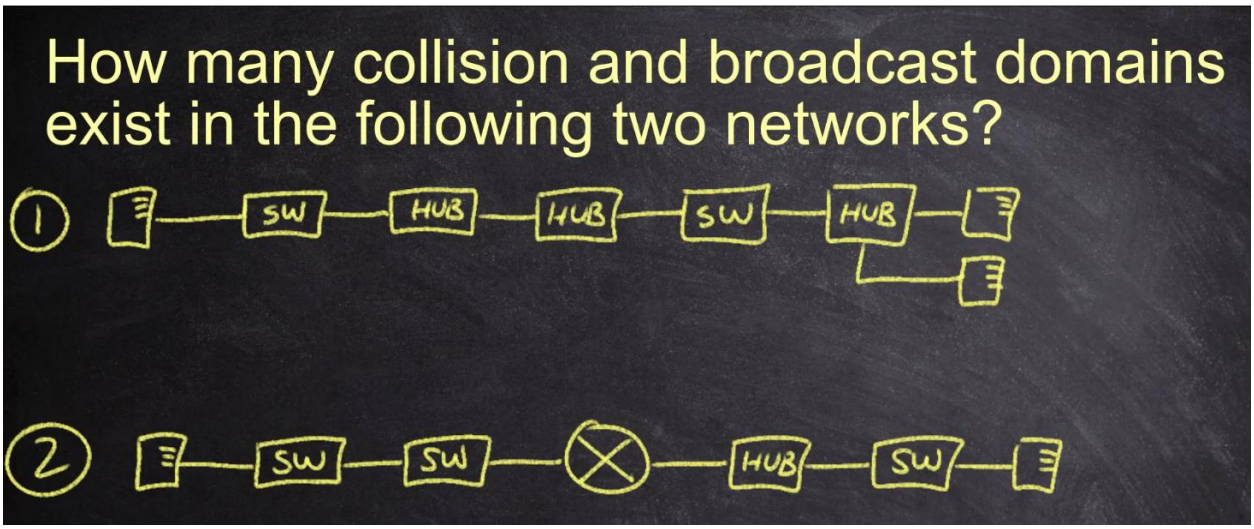
REMEMBER → → → → → → 10 Mbps require HALF DUPLEX. (HUBS only existed back then).

Ethernet Broadcasts

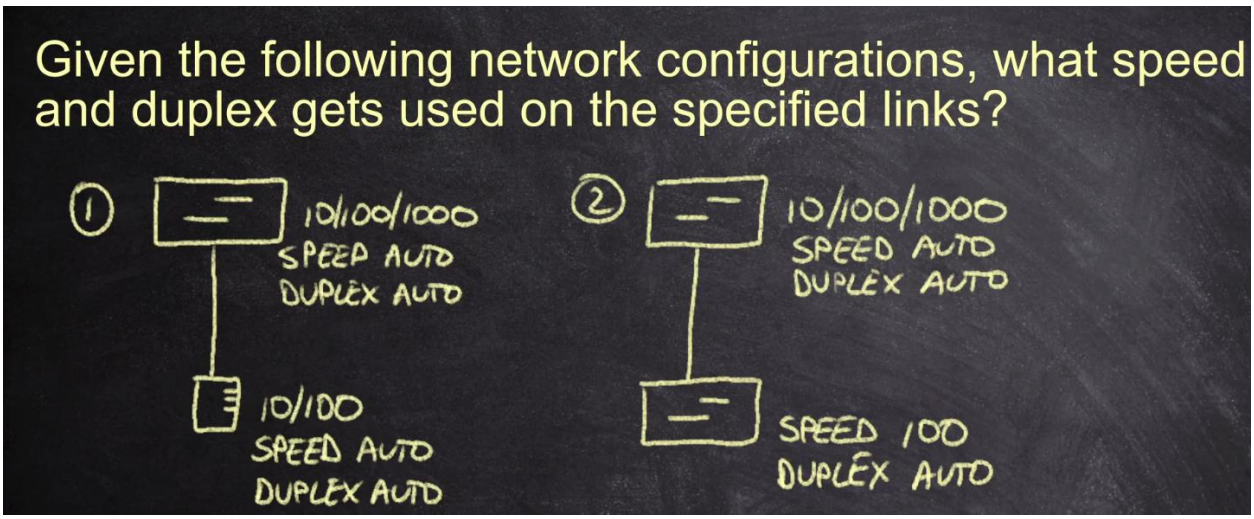
Ethernet delivers a functionality where an ETHERNET BROADCAST FRAME is sent through the entirety of a network. In this process the DEST MAC is set to 0XFF:FF:FF:FF:FF:FF (48 1s) and the switch understands this as a EB frame to be sent to every device. Now this introduces BROADCAST DOMAINS that will come to exists at L2, so at L3 happens a boundary (ROUTERS) where Broadcast domains stop to exist.



LAB/VALIDATION



18.- **Configure Ethernet Interface Speed and Duplex**



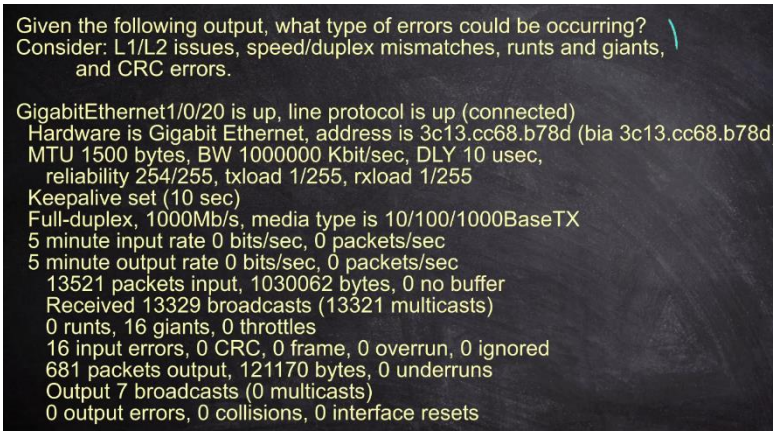
Both scenarios are going to come up to be at 100 Mbps and FULL duplex. But there is an scenario where if running at 100 Mbps, Cisco switches require to run at half duplex. So if we see this case, where one is running 100 M and at Auto duplex, and the other one at FULL duplex 100 Mbps, the mismatch is going to happen.

19.- **Identify Interface and Cable Issues**

Type of errors we might end up seeing:

Runt → 'Too small'. Ethernet frames have rules may never be <= 64Bytes with the payload being at least 46 Bytes. If the switch receives a smaller than 64 Bytes, it is going to drop it.

Giant → 'Too large' of a frame. The max size of the payload for an ethernet frame is 1500 Bytes. Called the MTU and this can actually be changed. We can go all the way up to 9000Bytes for example at Data centers. When changed, we call the MTU a JUMBO frame, but if a frame exceeds the limit, it is a GIANT.



Some packets have been dropped for being GIANT packets.

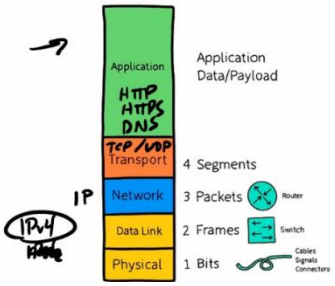
_____IMPORTANT_____

IF WE HAD A SPEED MISMATCH, WE WOULD BE AT A DOWN/DOWN STATE.

DUPLEX MISMATCHES CAN STILL HAPPEN EVEN IF THEY ARE UP/UP.

20.- **Understand IPv4 Addressing**

IPv4 Addressing Overview



Remember that at many layers in the model, we have several protocol working. In this module we will focus on L3 protocols like IPv4. Each network have their own IP terminations in order to know to what network hosts are part of. We have the NET part and the HOST part when we look at an IP address. The dividing line between the NET part and the HOST part is the MASK. An IPv4 Address is 32 Bit of data. And are composed of 8 bit each for each decimal position.

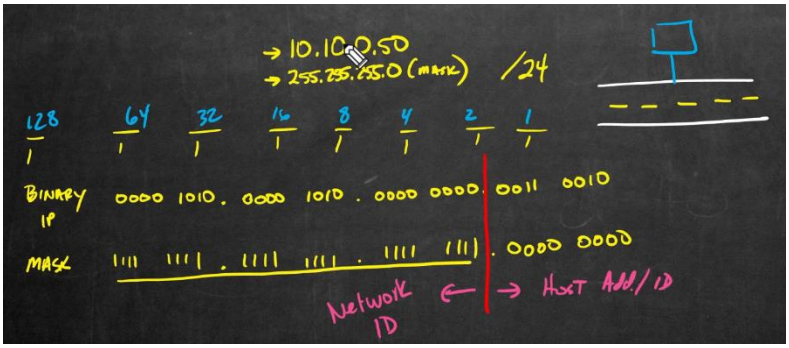
10.10.0.5 = 00000000.00000000.00000000.00000000

Ex. 11100011 = 2^7 + 2^6 + 2^5 + 2^1 + 2^0 = 128 + 64 + 32 + 2 + 1 = 227

Ex 73 =

Unveiling the Mask

The MASK is there to help the IP address (Host). In reality, if look more into the mask when represented by a /[NUMBER] it means that there are [number] of ones before the host.




```
1
2   Useful commands:
3
4   ! Cisco Router
5   # show ip interface
6   # show ip interface brief
7   # show ip interface brief | exclude unassigned
8   # show ip route
9   # show ip route connected
10
11   # config terminal
12   (config)# interface gig 2/0
13   (config-if)# ip address 10.10.0.1 255.255.255.0
14   (config-if)# no shutdown
15   (config-if)# end
16
17
18   ! Windows
19   ipconfig
20   ipconfig /all
21   ncpa.cpl
22   ping pc-20.nuggetlab.com
23   nslookup pc-20
24   ping pc-20
25   tracert -d pc-20
```

21.- **Configure IPv4 Private Addresses**

Classes of IPv4: In order to Identify each class, we look at the first number which is going to give us the information. They will have default MASKS although it may change

A → 0 – 127 /8 =255.0.0.0 B → 128 – 191 /16 = 255.255.0.0
C → 192 – 223 /24 = 255.255.255.0

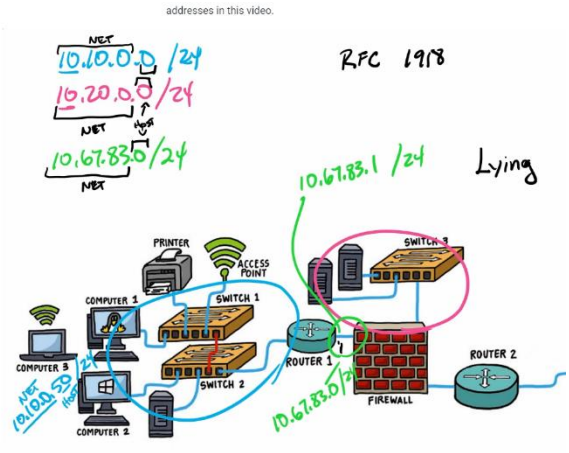
Private (RFC 1918) IPv4 Addresses

Private addressing was created to differentiate which hosts are part of our LAN instead of outside of the local network. The RFC 1918 has established that private addresses are the following:

10. ----------

172.16 – 31. ----------

192.168. ----------



If we look at this topology, we can see 2 subnets on BLUE AND PINK. We can assume that both are inside a LAN since they use PRIVATE addressing (10.-------). Look at the segment in green having assigned IP addresses to determine default gateways between ROUTER 1 and the FIREWALL of 10.67.83.1. But since they are inside the Private IPv4 address scheme, we can assign anything after the 10. -------- as long as the NETWORK ID makes sense for the rest of the hosts.

Now, a LAN using private addresses can still access WANs through a protocol called NAT/PAT.

Private IPv4 Address Planning

```
1   show ip int brief
2
3   config t
4   interface gig 0/0
5   no shutdown
6   ip address 10.10.0.1 255.255.255.0
7   interface gig 1/0
8   no shutdown
9   ip address 10.67.83.1 255.255.255.0
10   end
11
12   show ip int brief
```

22.- Use IPv4 Subnetting

When we subnet (10.67.83.0) /24 we take the 'parent' network and logically subdivide it into any amount of logical subnets. So how do we do this? We "chop" the subnet mask.

NET 10.67.83.0 /24

NET ID → 00001010 . 01000011 . 01010011 | 00000000

MASK → 11111111 . 11111111 . 11111111 | 00000000

If we want to create subnets given this example, we move the mask to the right, taking bits for our logical subnets. So let's move the mask 3 bits to the right.

NET ID → 00001010 . 01000011 . 01010011 . 0000 | 0 | 000

MASK → 11111111 . 11111111 . 11111111 . 1111 | 1 | 000

So now, our Mask would be /29 = 248. There will be less hosts of course, since we took out of 255 possible hosts (3 bits to our left) leaving only 7 possible hosts. So in summary, if we take 'x' amount of bits from the hosts portion, we take progressively more possible number of subnets: /25 => 2 subnets /26 => 4 subnets, /27 => 8 subnets

/25	/26	/27
1 Bit	2 Bits	3 Bits
0	00	000
1	01	001
	10	010
	11	011
		100
		101
		110
		111

So what actually happens to the parent network? 10.67.83.0. We Take the subnets as ranges and the way we do this is seeing all the possible positives 1s we have in the extra bits of our subnet mask:

/28 ========> . 1111 | 0000 | 10.67.83.0

1st subnet ------ . 0001 | 0000 | 10.67.83.16

2nd subnet ------. 0010 | 0000 | 10.67.83.32

3rd subnet ------ . 0011 | 0000 | 10.67.83.48

4th subnet ------ . 0100 | 0000 | 10.67.83.64

Notice how each range of possible new hosts inside each of the new subnets increments on ranges of the last bit 1 that we have moved our subnet mask to.

In summary, we'll have as many possible hosts for each new subnet as the last bit 1 that we have extended the subnet mask to. But remember that we need IP addresses for routers and leave one for our actual IP address for the network (Default gateways), so we leave out the -1 at the end of the last valid host address and +1 for the first valid host address.

Ex. For the subnet 10.67.83.32 /28, which of the following are valid IP addresses that can be configured on an interface?

10.67.83.32 /28 =====> 1111 0000 (increments of 16 each subnet)

.32 → .47 ---------- .33 → .46 It was only for this subnet

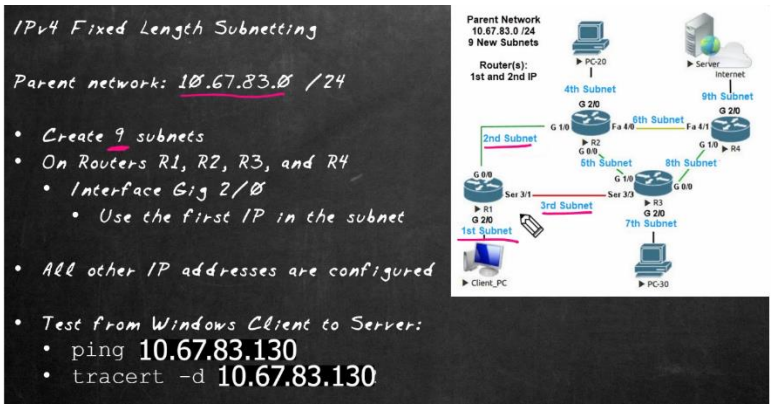
.48 → .63 ---------- .49 → .62

.64 → .79 ---------- .65 → .78

.80 → .95 ---------- .81 → .94

.96 → .111 ---------- .97 → .110

VALIDATION.LAB



To create 9 subnetworks, we would need to move the subnet to /28 which will provide us with 16 possible hosts for each subnet.

.0 → .15

.16 → .31

.32 → .47

.48 → .63

.64 → .79

.80 → .95

.96 → .111

.112 → .127

.128 → .143

23.- Use IPv4 Variable Length Subnets Masks (VLSM)

What happens if we don't want to not have equal hosts for each subnet?

Using Variable Length Subnet Mask (VLSM) is the solution for this. So what's the logic behind it?

1.- We write out the parent network. In our example it is 10.67.83.0 /24

2.- Number of hosts needed. Let's consider 60, 50, 20, 10, 5, 5 for an equal maximum number of 60 hosts. Now we see how that changes our subnet mask. To support a maximum number of 60 hosts, we would need to add +6 to /24 = /30. But only for the first one, or the subnet that requires it. Therefore the variable length comes into place. We are playing with what we need in terms of the MOTHER SUBNETMASK we already have.

The Starting Point for VLSM Calculations

The more subnets we have, the less hosts we are going to have per subnet. That's why when we are asked to divide a network x times to fulfill a need for subnets, we do not see the number of hosts each are going to have. So, if we are in need of 10 new subnets, we would need 16 hosts per subnet to fulfill the need.

In

1.- NET ADDRESS: 10.67.83.0 (60 hosts) -------- Move 6 bits (64 hosts) => /26 ----- .0 - .63 (-2 for broadcast and subnet) == 255.255.255.192

Based on our first, like a chain, we start doing the rest of the subnets.

2.- NET ADDRESS: 10.67.83.64 (50 hosts) ---- Move 6 bits (64 hosts) => /26 = 255.255.255.192

RANGE: .65 - .126 (.64 for NET ADRR and .127 for broadcast)

3.- NET ADDRESS: 10.67.83.128 (20 hosts) --- Move 3 bits (32 hosts) ➔ /27 = 255.255.255.224

RANGE: .130 – 158 (.128 for NET ADDRESS & .159 BROADCAST)

4.- NET ADDRESS: 10.67.83.160 (10 hosts) --- Move 4 bits (16 hosts) → /28 = 255.255.255.240

RANGE: .163 – 178 (.160 for NET ADDRESS & .175 broadcast)

5.- NET ADDRESS: 10.67.83.176 (5 hosts) --- Move 5 bits (8 hosts)➔ /29 = 255.255.255.248

RANGE: .177 – 182 (.176 FOR NET ADD & .183 BROADCAST)

6.- NET ADDRESS: 10.67.83.184 (5 hosts) – Move 5 bits (8 hosts) ➔ /29 = 255.255.255.248

RANGE: (.184 for NET ADDRESS 7 .191 BROADCAST) .185 - .190

VLSM Example Across Octets

Exercise. PARENT NETWORK 67.83.64.0 /19 =====> Notice how now our subnet mask would fall on the 3rd octet instead of the last one. Meaning that 67.83 part is the NET ID and our subnets would start in the 3rd octet. The process is the same.

1.- (901 hosts) NET ADDRESS: 67.83.64.0 from right to left how many bits do we need? (Finger game) 10 bits (From right to left meaning 32 bits – 10 = 22 bits OR /22) for 1024 hosts.

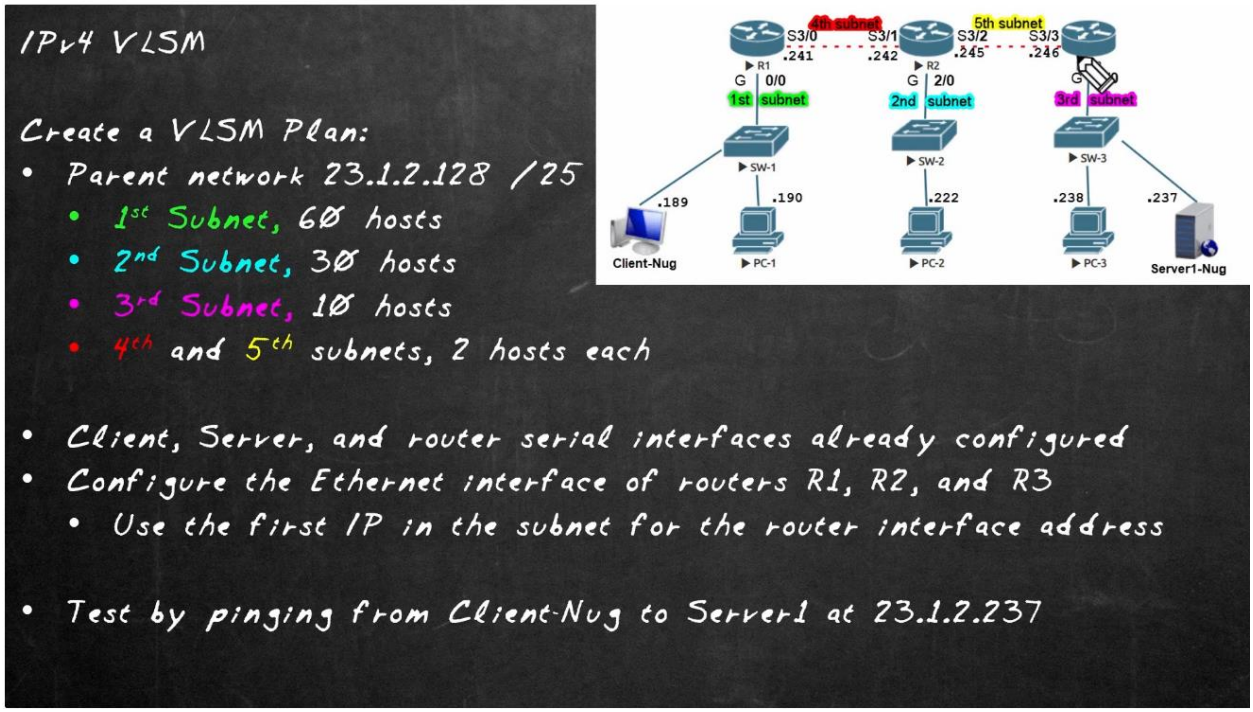
NET ADDRESS BROADCAST ADDRESS

1.- 901 hosts – NET ADDRESS: 67.83.64.0 /22	RANGE: .64.0 → .67.255	ACTRa: .64.1 → .67.254
2.- 502 hosts – NET ADDRESS: 67.83.68.0 /23	RANGE: .68.0 → .69.255	ACTRa: .68.1 → .69.254
3.- 203 Hosts – NET ADDRESS: 67.83.70.0 /24	RANGE: .70.0 → .70.255	ACTRa: .70.1 → .70.254
4.- 104 hosts – NET ADDRESS: 67.83.71.0 /25	RANGE: .71.0 → .71.127	ACTRa: .71.1 → .71.126
5.- 2 Hosts – NET ADDRESS: 67.83.71.128 /30	RANGE: .71.128 → .71.131	ACTRa:.71.129→71.130
6.- 2 Hosts – NET ADDRESS: 67.83.71.132 /30	RANGE: .71.132 → .71.135	ACTRa:.71.133→71.134

Ex. 23.1.2.128 /25

1.- 60 hosts – NET ADD: 23.1.2.128 /26	RANGE: .128 → .191	ACTRange: .129 → .190
2.- 30 hosts – NET ADD: 23.1.2.192 /27	RANGE: .192 → .223	ACTRange: .193 → .222
3.- 12 hosts – NET ADD: 23.1.2.224 /28	RANGE: .224 → .239	ACTRange: .225 → .238
4.- 2 hosts – NET ADD: 23.1.2.240 /30	RANGE: .240 → .243	ACTRange: .241 → .242
5.- 2 hosts – NET ADD: 23.1.2.244 /30	RANGE: .244 → .247	ACTRange: .245 → .246

VALIDATION



23.1.2.128 /25	128	64	32	16	8	4	2	1

1.- 60 hosts – NET ADD: 23.1.2.128 /26 RANGE:.128 → .191 ACTRA: .129 → .190

2.- 30 hosts – NET ADD: 23.1.2.192 /27 RANGE: .192 → .223 ACTRA: .193 → .222

3.- 10 hosts – NET ADD: 23.1.2.224 /28 RANGE: .224 → .239 ACTRA: .225 → .238

4.- 2 hosts – NET ADD: 23.1.2.240 /30 RANGE: .240 → .243 ACTRA: .241 → .242

5.- 2 hosts – NET ADD: 23.1.2.244 /30 RANGE: .244 → 247 ACTRA: .245 → .246

24.- Use Additional IPv4 Addressing

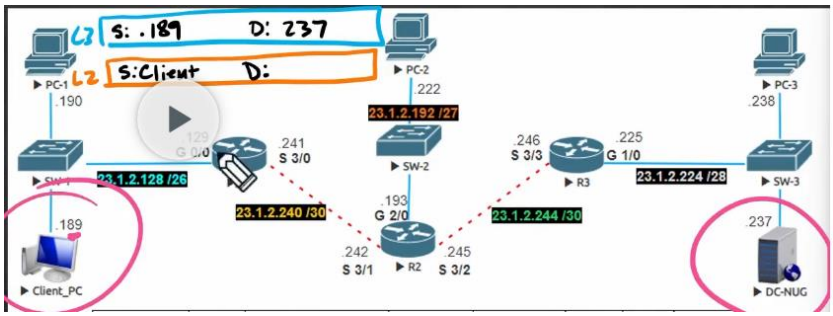
Unicast → Unicast means that it is coming from 1 IP address directed to another one like a PING request. Thanks to RFC 1918 We have private IP ADD within each IP class.

	CLASS A [0 - 127]	CLASS B [128 - 191]	CLASS C [192 - 223]
RFC1918	10. -------	172.16 – 31. -------	192.168.-------

If we were to capture the traffic from CLIENT-PC to DC-NUG on 23.1.2.128, we would see:

L3 S:.189 D: .237

L2 S:CLIENTMAC D: G0/0-R1 MAC



Through ARP, the client would ask 'who' is in it's network and create an ARP table for itself to know when to send WAN traffic and where to.

Same would happen if we captured traffic on 23.1.2.224:

L3 S:.189 D: .237

L2 S:R3 MAC D: DC-NUG MAC

IN OVERALL: Destination and source MAC addresses change depending on what hop they are on.

Broadcast → When a client wants to send all the devices to hear it. That is why we leave a broadcast address free. When we use PING [BROADCAST ADDRESS] the switch understands that it needs to forward it to all the devices. (1 to all)

This happens at L3 & L2 as well. At L2 would be FF:FF:FF:FF:FF:FF. Another variation at L3 255.255.255.255. ARP is a great example of a protocol that uses Broadcast IP addresses to properly function

Multicast → 1 to SOME devices. We use CLASS D 224 – 239. ------

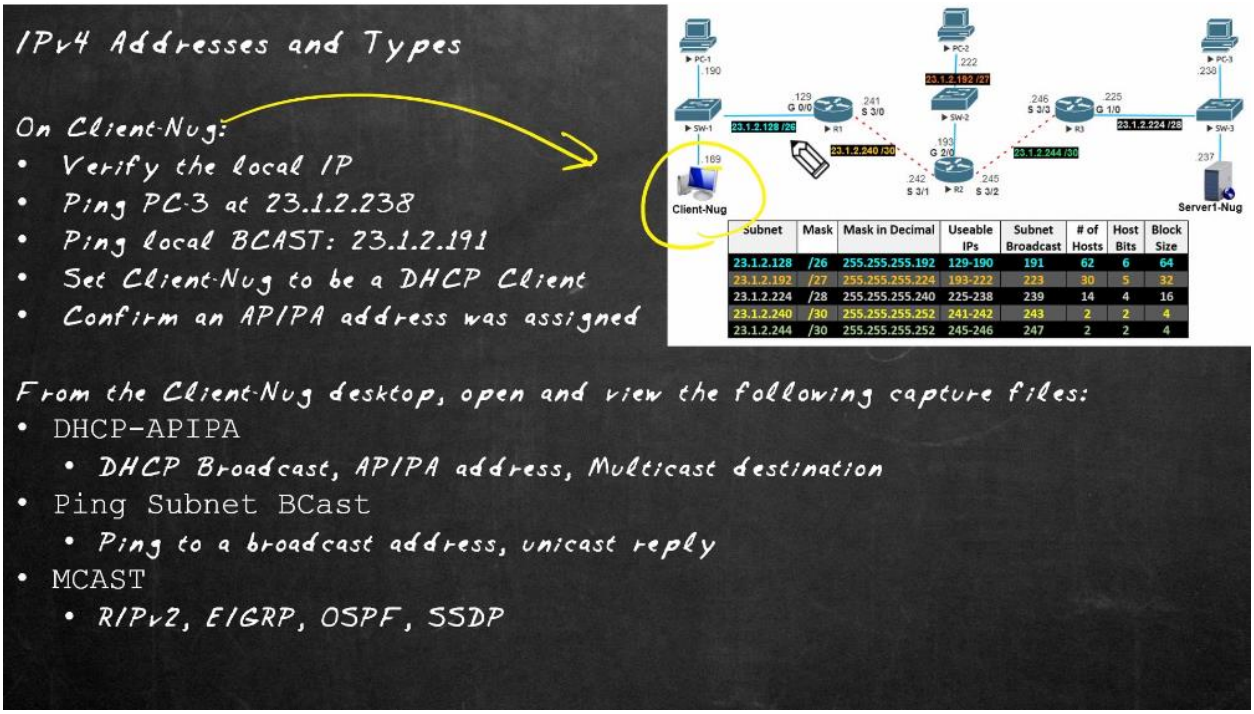
Many of the routing protocols need multicast addresses. OSPF, RIP are some examples.

Anycast → To the closest IP ADDRESS on the routing table.

LINK LOCAL IPv4 → DHCP services provide IPv4 to clients but when a client is not receiving a response, it will provide itself with an APIPA address 169.254.X.X. These addresses are only destined to be used inside a LAN hence not routable.

LOOPBACK → 127.X.X.X logically clients think this IP is themselves. If the same IP LOOPBACK address is advertised in more than 1 place in the network, the target can still connect to this IP ADDRESS that is the closest advertised to them.

Validation/LAB



DHCP-APIPA explanation → Since we set the client to 'find' or request an automatic IP ADDRESS but could not find any DHCP servers, it assigned an APIPA address to itself being 169.254.110.49. On the packet capture, we can see a first ARP request with destination 255.255.255.255 (BROADCAST) asking for a DHCP server, in fact it does it 3 times

```
1 0.0.0.0          255.255.255.255    DHCP    DHCP Discover - Transaction ID 0xeb3be4e2
2 0.0.0.0          255.255.255.255    DHCP    DHCP Discover - Transaction ID 0x16953121
3 0.0.0.0          255.255.255.255    DHCP    DHCP Discover - Transaction ID 0x8f13bd97
```

Ping Subnet BCast → In here we can see the use of ICMP protocol via PING command with an unicast address (being the broadcast address of the subnet). Notice how the client pc at layer 2 uses the mac broadcast address.

```
Ethernet II, Src: Microsoft_00:04:08 (00:15:5d:00:04:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
```

MCast → In here we see different routing protocols that use MULTICAST addresses. Notice the IP addresses that are in the CLASS D.

```
1 23.1.2.129       224.0.0.10         EIGRP
2 23.1.2.189       239.255.255.250    SSDP
3 23.1.2.129       224.0.0.9          RIPv2
4 23.1.2.129       224.0.0.5          OSPF
```

25.- Understand IPv6 Addressing

When we talk about IPv4, 6 addressing we talk about L3 protocols. They both have APPLICATION layer support, and use L4 transport protocols support like TCP & UDP.

IPv4 ➔ A.B.C.D /[subnet] 32 bits long 8 digits per segment

IPv6 ➔ A:B:C:D:E:F:G:H each segment represented by 16 bits OR equal to 1 HEX NUMBER. 128 bits long

The first 64 bits is the NETWORK PORTION, and the last 64 bits is the HOST ID.\

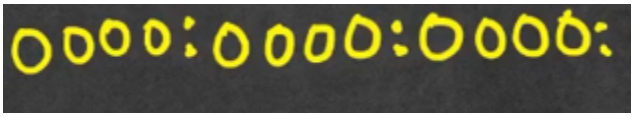
Let's see how to write one:

RULES:

➔ If we have leading 0s, we don't write them and assume there are 0s there.

This would be the same as writing :A:

⇨ Multiple groups of 0s we use dual :: but this trick can only be used once in one address.

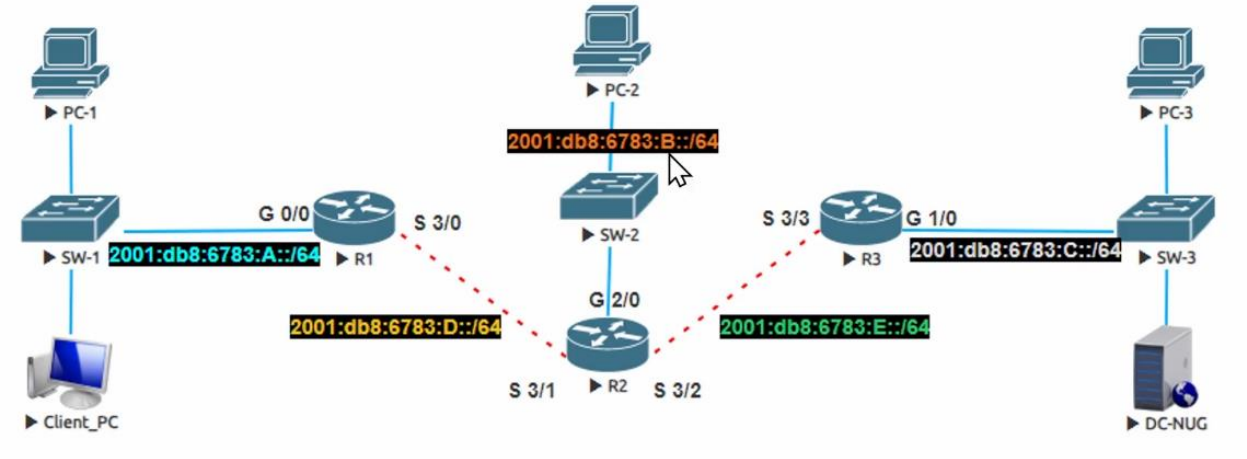


::

The final IPv6 address would look like this:

Address types:

Address	Intended Purpose	Example
2000-3FFF::	Global Unicast	2001:db8:6783:a::1 /64
FC00:: /7	Unique Local Unicast	Fc00:1234:abcd:bad:f00d::9 /64
::1 /128	Loopback	::1 /128
FE80:: /10	Link Local	Fe80::12:34:56::99:22
FF00:: /8	Multicast	FF02::9



Let's look at this example to guide ourselves.

GLOBAL UNICAST→ The first 4 characters 2000 – 3FFF: Think of it as a routable IP address on the internet publicly.

UNIQUE LOCAL UNICAST → starts with FC00. Very similar to the IPv4 PRIVATE ip addresses like the ones inside each IPV4 class described in RFC1918.

Loop Back → ::1 /128 which means all the bits are 0s.

Link Local → Remember that link local means NOT ROUTABLE and only used within our LAN.

Multicast →

→→→→IMPORTANT NOTE →→→→ Masks in IPv6 mean that we only care the first X bits considering it as our network ID and the rest for the host ID.

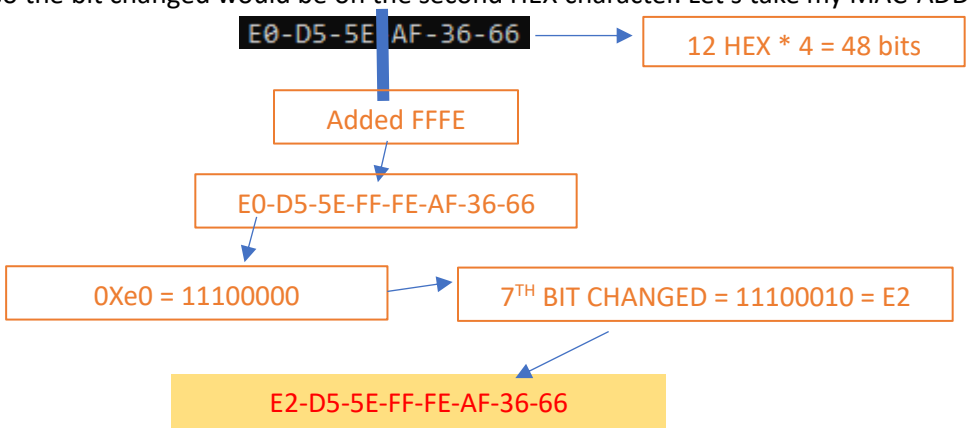
NDP Neighboring Disc Protocol.

`2000-3FFF::`

Just like ICMP using broadcast addresses in IPv4, NDP relies heavily on MULTICAST. One great example that NDP is used, happens when we look at routing protocols running on IPv6. Multicast works in terms of groups for routers to understand and separate multicast. For example, FF02::2 is a multicast address used for routing. Depending on the routing protocol using, it will have different numbers at the end.

Something interesting that happens with IPv6 is that clients can create their own **host address.** This process is called EUI-64, with the 64 representing how many bits are going to be for the host address.

EUI-64 takes the 48 bits of the MAC-ADDRESS of a device, chops it on the middle and adds extra bits. But the 7th bit will be changed over. Keep in mind that this is hexadecimal and that each hex character represents 4 bits, so the bit changed would be on the second HEX character. Let's take my MAC-ADDRESS as an example.



So there are still 64 bits more in order to create the IPv6. This will be given by the network portion + the host address just created. Let's consider `2001:db8:6783:A::/64`

It will end up being ➔ 2001:db8:6783:A::E2D5:5EFF:FEAF:3666

Link-local ADRRESSES `FE80:: /10`

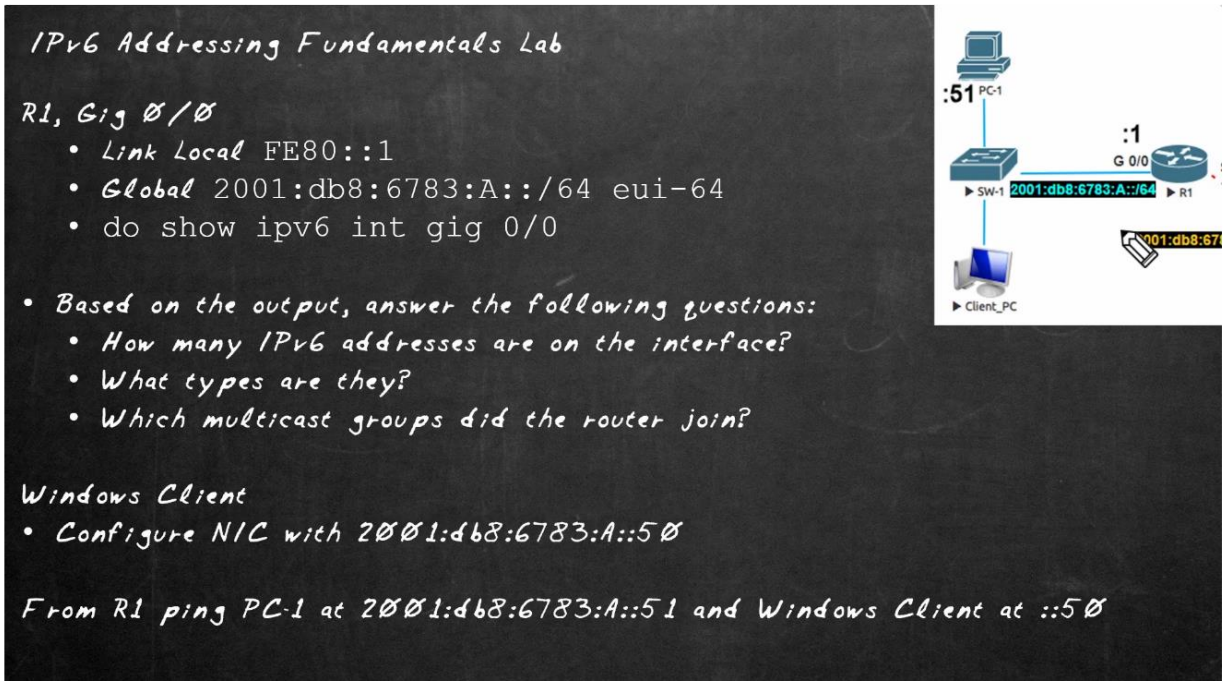
If we want to communicate locally, we use link-local addresses. Just like the RFC1918 has private IPv4 addresses for hosts to communicate within a LAN. When talking about link local addresses we have to remember that for the host portions of the IPv6s, the clients will be sharing the same HEX values. We can both directly configure the Link-local and the Global Unicast address. In some flavors, for the link local's host portion, it uses the same logic as the EUI-64 process of adding FFFE in the middle of the MAC address.

Multi Cast IPv6 `FF00:: /8`

```
Joined group address(es):
  FF02::1
  FF02::2
  FF02::9
  FF02::1:FF00:1
```

When we configure IPv6, they automatically joing groups in order to know where to send multicast frames.

Validation/LAB

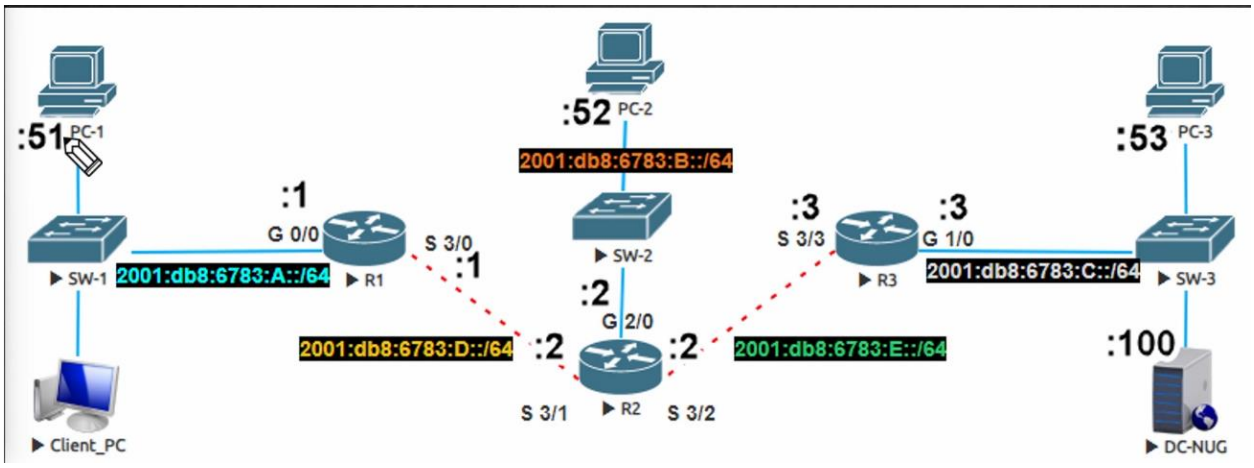


➔ There are 2 IPv6 addresses:

- The first one is a Global unicast `2001:DB8:6783:A::/64`
- The second one is a Link- local address `FE80::1`

26.- Configure IPv6 Addressing

Let's configure the following IPv6 network:



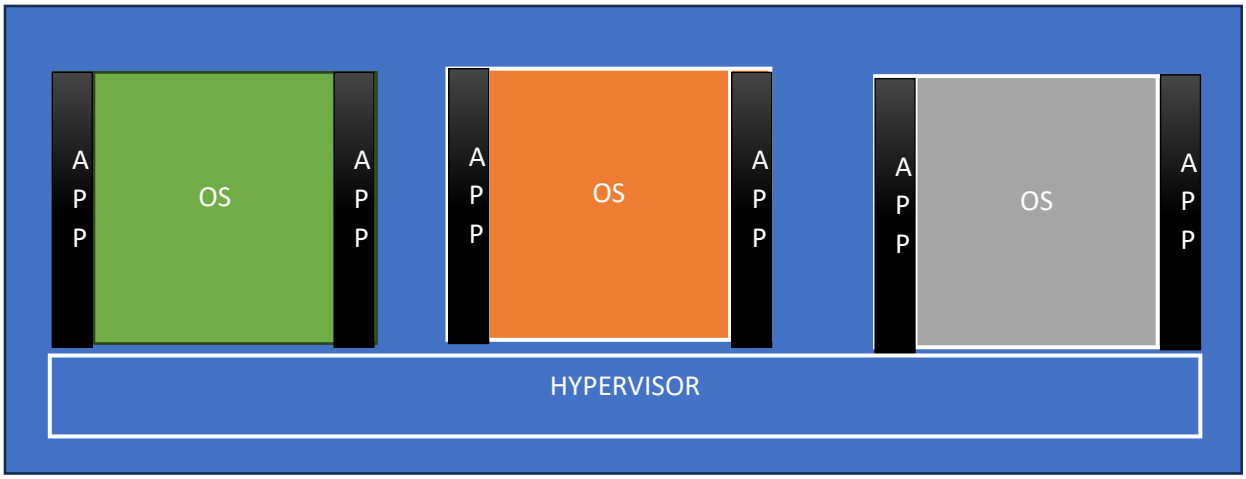
Important note → You can configure on the end devices, the IPv6 address statically or automatically (SLAAC) which enables NDP (Neighbor discovery protocol), assign themselves with the network portion of the IPV6 we see and then run DAD to verify there are not repeated host portions of the IPV6 assigned to themselves.

Important link-local → We can use the same Link local address on the same router because they are unique to only an specific subnet.

27.- Explain Virtualization Fundamentals

Remember when we talked about network blocks and how each connected with each other. Data Center blocks come into play when we talk about virtualization where APPS + DATA is stored, access, managed, controlled. It is a robust environment where a server runs. A server need a CPU, RAM, DISK, OS and network interfaces to run and APPS run in them.

If we run MULTIPLE APPS on 1 OS we might run into resources problems. Memory allocation, CPU usage play important roles in a high performance environment. Even some resources can not be shared. Like the TCP/UDP ports. Let's imagine 3 apps that require HTTP where they would require the same TCP port:80. A viable solution would be to deploy a piece of hardware which on top runs an OS which on top runs an APP. You do this repeatedly so no apps consumes resources needed for all three all for itself.



The main advantages of deploying virtualization is that:

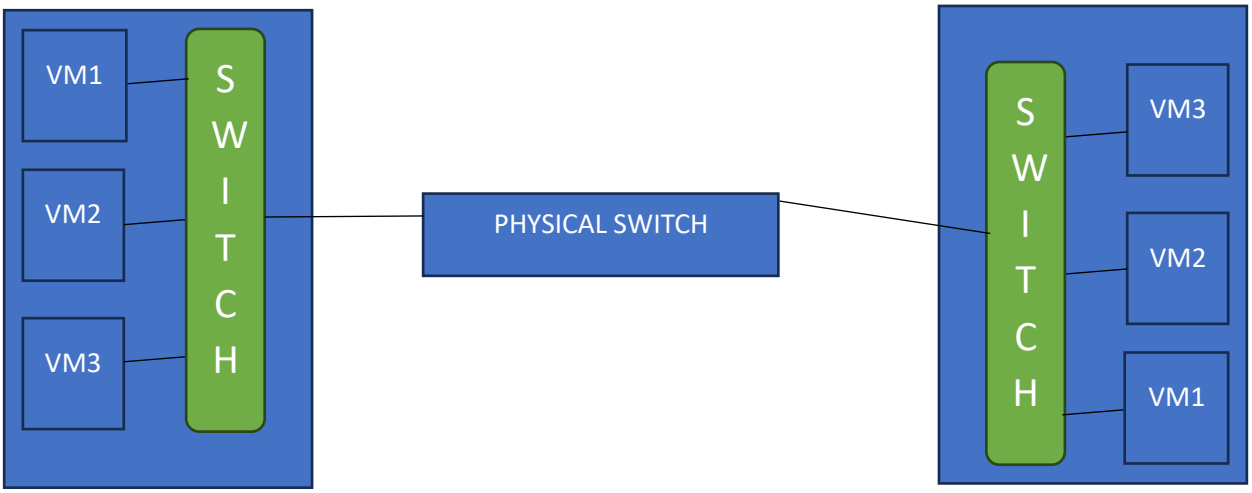
- ➔ Whenever we have scalability issues
- ➔ Management
- ➔ Resource management

On top of hardware, a HYPERVISOR runs, where Oss and apps on top of those OS run.

As said, we can virtualize networks. On the host, a virtual switch can run. It connects to all the hosts in order for them to communicate.

Validation.

Diagram out two virtual hosts, each with three VMs, and identify how traffic flows between VMs on one host compared to VMs on different hosts.



28.- Understand L2 Switch Forwarding

VLAN #	MAC Address	Type	Port
	...		Gig /
	... 6807		Gig 0/3
	... 6801		Gig 0/1

Remember that the mission of a switch is to forward L2 frames based on the L2 destination address (MAC address). The problem is that when a switch is first turned on, it does not recognize the hosts connected to it. It starts creating a 'MAC table' in which it recognizes the MAC address coming from a specific port. This MAC TABLE also has a determined age that given a certain time, deletes the information stored and it starts repopulating the table again.

Another important concept with L2 switching is ARP which is a request that uses BROADCAST to 'liste' to the frames sent from one client to the rest of the network.

Another concept is FLOODING. Which occurs when a switch does not have any information on where a host is connected to. So it FLOODS the frame to all the other ports (Except for the one that it came from).

In SUMMARY when:

UNICAST FLOODING → Unknown MAC address

AGING → When a MAC address hasn't been forwarded in 5 minutes

BROADCAST → ARP REQUEST

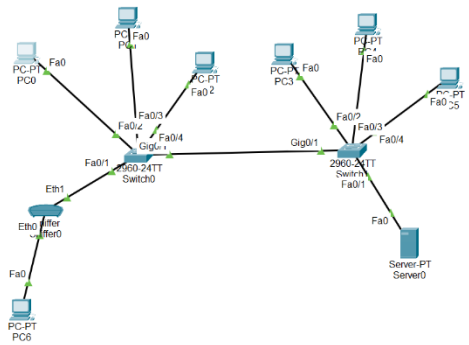
UNICAST FORWARDING → Known MAC address

Once the MAC tables are created, the switch will not bother any other ports when a frame needs to be forwarded.

A L2 BROADCAST address on the destination header looks like FFFF.FFFF.FFFF

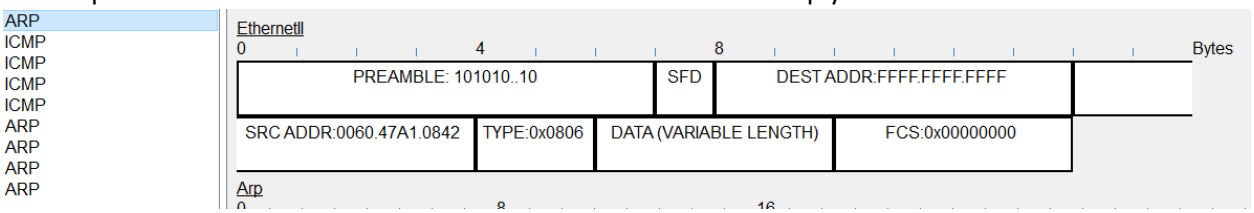
A L3 BROADCAST address depends on the subnet it is on but it would look like .255

Let's consider this lab



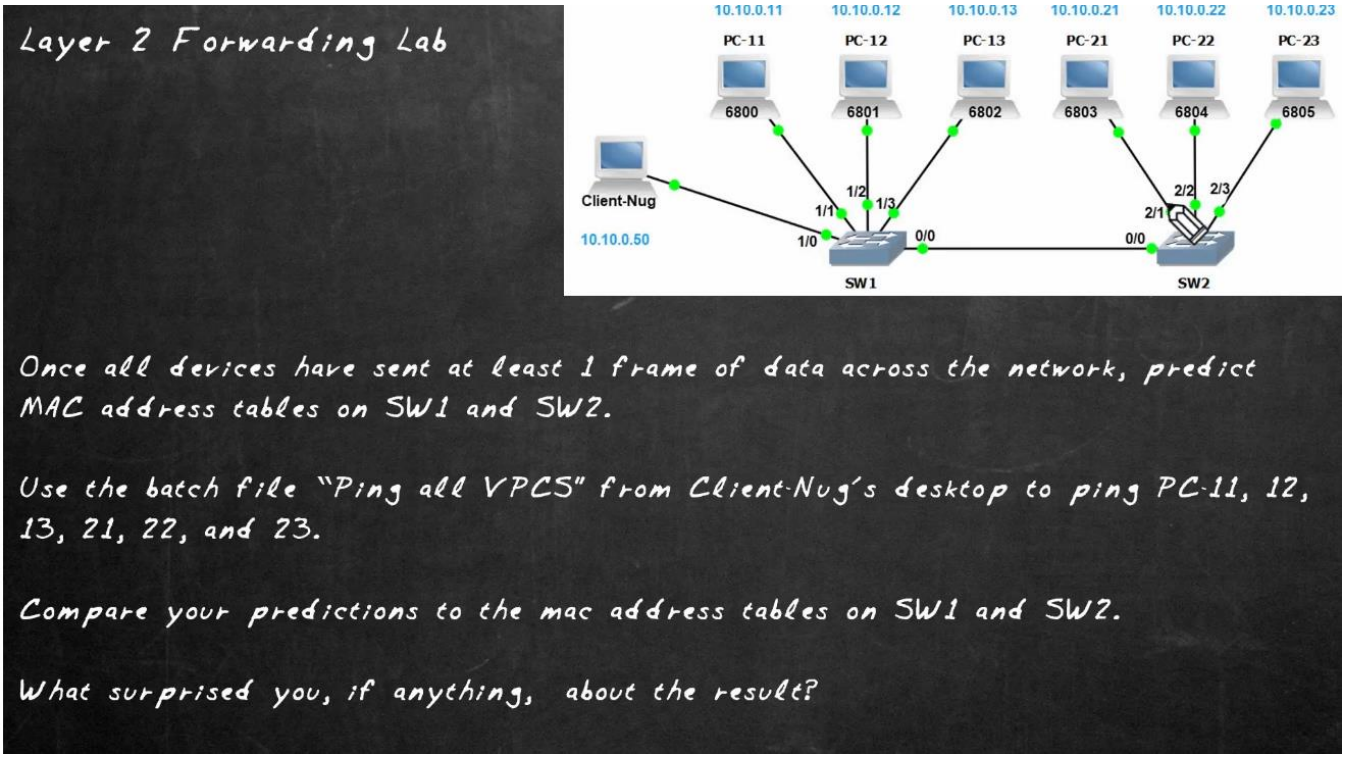
The mac address table on sw1 has been clear as well as all the arp tables on the hosts. PC0 sends a ping to the server and pc6 at the bottom left receives this packet with a broadcast MAC address of ff.ff.ff.ff being a broadcast. Remember that ARP is a protocol that maps IP addresses to MAC addresses and therefore when a host does not have knowledge of where another host is located at, it sends an

ARP request with a broadcast address so that the destination can reply back.



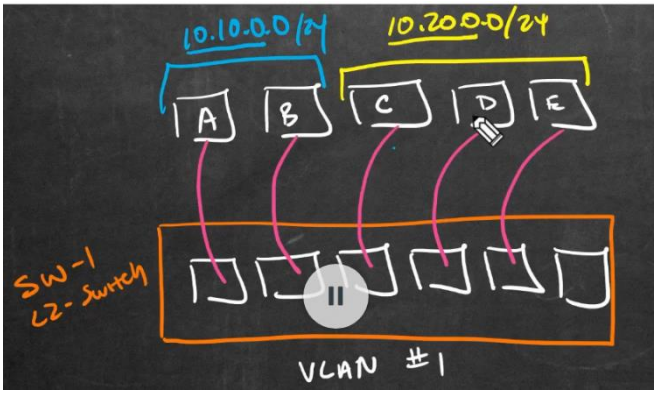
FLOODING → Is when a switch receives frame, looks at the DEST layer 2 address and does not have it on it's MAC address table so it floods it to all the other hosts but not the same interface that it received it from. So an Unicast flood is because of a frame destined to an unknown MAC address.

Validation



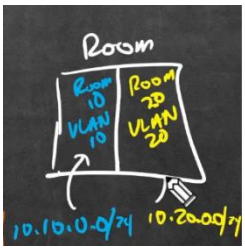
MAC Address	Type	Port
Client nug		1/0
PC11		1/1
PC12		½
PC13		1/3
PC21		0/0
PC22		0/0
PC23		0/0

30.- Understand L2 VLANs



A VLAN is like a room. It logically separates LANs inside a switch(es). Hosts inside the same VLAN are inside the same BROADCAST DOMAIN. They are logically segmented physical networks. Unicast, Broadcasts, multicast packets are forwarded or flooded to end devices connected to the same VLAN. Important to mention that VLANs occur as a L2 technology.

Let's imagine that we have 2 groups of PCs connected to the same switch (ALL SHARING THE SAME L2 BROADCAST DOMAIN) but each group on different subnets. Even both groups believe they are in different networks, their L2 broadcast domain remains relevant. A broadcast sent from A will still be sent to all the other ports, because the L2 broadcast domain is still the same.



Here is where VLANs come into play. They represent a separate logical L2 separation, even though they exists in the same switch.

THERE IS A DIFFERENCE BETWEEN L2 BROADCAST DOMAINS AND IP SUBNETS SEPARATIONS.

In summary, a L2 BROADCAST DOMAIN is a lot like a room where everyone inside can hear each other.

There is a correlation between an IP L3 SUBNET and it's own L2 BROADCAST DOMAIN in a well established network.

31.- Configure Inter VLAN Routing

Moving packets at L3 between 2 subnets. There are 3 main ways to implement inter VLAN routing.

1.- 1 router with 2 physical interfaces

2.- 1 physical router and 1 physical interface where 1 port on the switch is NOT going to be associated with any VLAN and we are going to call it TRUNK PORT

Where the TRUNK PORT is used to connect multiple VLANs inside the switch and forward traffic between VLANs and outside them. This is done by creating multiple logical interfaces to support each VLAN traffic that go through the same TRUNK PORT.

It's all about configuration in the router for the first type of configuration using only 1 router and 2 physicial interfaces

We would need to configure our default gateway in all our hosts and provide an IP address to the interface on the router that we want to have as a default gateway.

```
Router#show ip route | include C
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
C       10.10.0.0/24 is directly connected, FastEthernet0/0
C       10.20.0.0/24 is directly connected, FastEthernet0/1
Router#
```

Tracert to a host in the same VLAN 10 →

```
Tracing route to 10.10.0.51 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      10.10.0.51
```

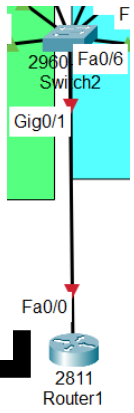
Tracert to a host in a different VLAN 20 →

```
C:\>tracert 10.20.0.51

Tracing route to 10.20.0.51 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      10.10.0.1
  2    0 ms      0 ms      0 ms      10.20.0.51
```

Inter VLAN Routing Using Router on a Stick (ROAS)

Now let's see how to configure a TRUNK PORT. Using 1 physical NIC connecting to the router and using 2 logical sub interfaces in order to provide connectivity to the 2 VLANS. On the switch we create the TRUNK PORT with a standard called 802.1Q

In this method, each traffic that comes from a certain VLAN are tagged so the trunk port knows where to direct traffic. Let's imagine an ARP frame coming in from a host located at VLAN 10 trying. The switch will know thanks to the tagged portion of the frame where to broadcast the ARP frame to the rest of the VLAN's 10 hosts.



In our topology, we'll configure int gig0/1 on the switch as our trunk port.

```
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk encapsulation dot1q
```

```
Gig0/1                          connected    trunk
```

In our router we will say that in our sub logical interface Fa0/0 .10 to be for our VLAN 10. And Fa0/0 .20 for our logical interface for VLAN 20.

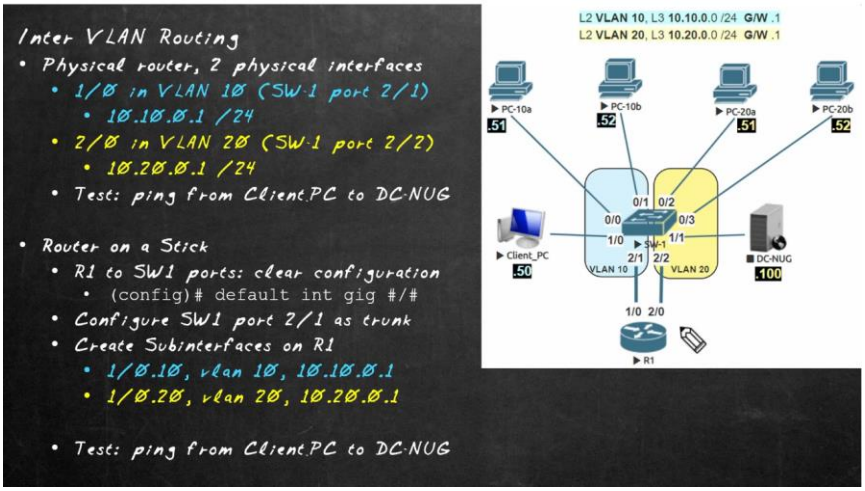
```
Router(config)#in f0/0.10
```

```
Router(config-subif)#encapsulation dot1Q 10
```

```
Router(config-subif)#ip address 10.10.0.1 255.255.255.0
```

```
Router(config)#do show ip int brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        unassigned      YES unset  up                    up
FastEthernet0/0.10     10.10.0.1       YES manual up                    up
FastEthernet0/0.20     10.20.0.1       YES manual up                    up
FastEthernet0/1        unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
```

VALIDATION/LAB



32.- Configure a Multi-Layer Switch

In order to set a password to go from USER MODE into GLOBAL CONFIG MODE

```
Switch(config)#enable secret Cisco123
```

Management ADDRESS → We can configure an IP Address to a interface on a switch for us to connect to it or for other reasons. We do so by creating a Logical L3 interface. This interface will be living in a VLAN. Then we can connect SSH, Telnet, etc to connect to that IP Address. This is called a Switch Virtual Interface. So basically as a physical interface that are associated with a VLAN, our SVI can also be associated with a VLAN.

```
Switch(config)#int vlan 1
Switch(config-if)#ip address 192.168.1.201 255.255.255.0
```

We would also need to set a default gateway.

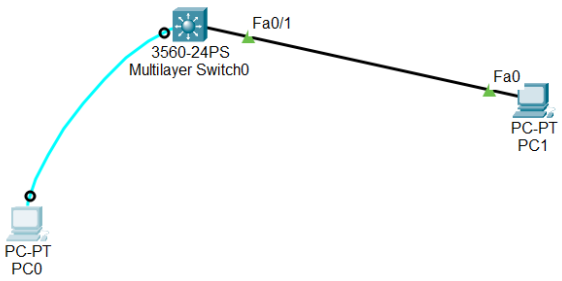
```
SW-1(config)#ip default-gateway 192.168.1.1
```

Connecting to a Switch IP Address Through a VTY Line

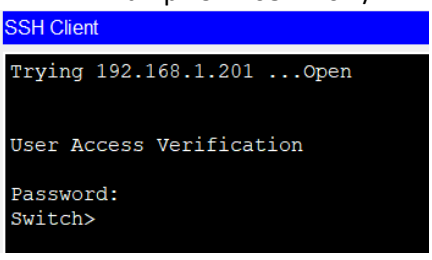
Imagine a VTY Line as having a virtual line from which you can remotely log in to your switch to manage it.

You set up these VTY lines by providing a password. So after having a VLAN SVI to connect to, we can set the VTY line as follows:

```
Switch(config)#line vty 0 15
Switch(config-line)#password Cisco
```

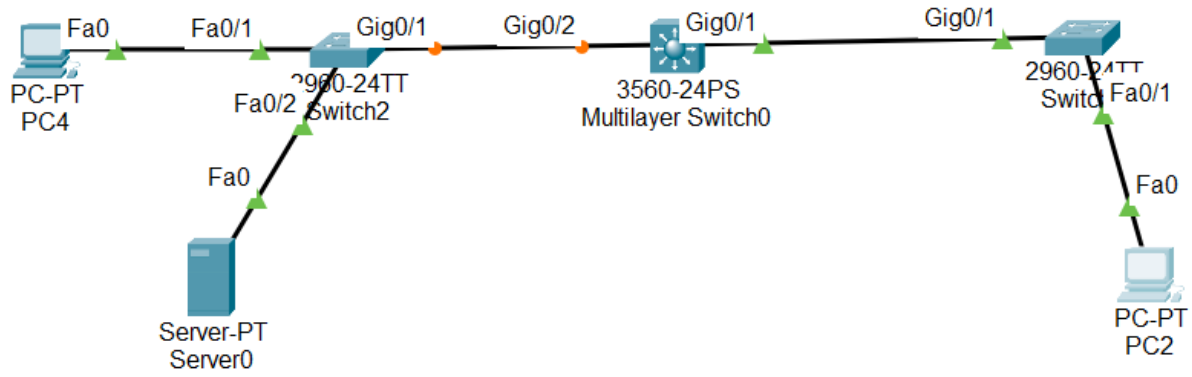


In our example, we set a Virtual interface SVI on multilayer switch on VLAN 1 with ip 192.168.1.201 /24 and after giving our PC-1 an IP ADDRESS we can use SSH to connect



Converting a Physical Layer 2 Switchport to a L3 Routed Port

Since we are working with L3 switches and these can route now at a L3 fashion, we can set one port specifically in our L3 switch to be able to route IP traffic while the others act as a L2 broadcast interfaces.



In this example we would need Gig0/2 and Gig0/1 interfaces to be able to route L3 TRAFFIC.

```
Switch(config)#int g0/1
Switch(config-if)#no switchport
Switch(config)#int g0/2
Switch(config-if)#no switchport
```

We can now see that these 2 interfaces are not associated to any VLANs and their status has changed to 'routed'.

Now, as a last step we would need to tell the switch to 'activate' routing with one simple command:

```
Switch(config)#ip routing
```

Now it makes sense to have multiple interfaces from which a router will route L3 traffic. And it makes more sense to make these interfaces, instead of physical ones, logical SVI inside a specific VLAN. We would apply the same steps that we hace implemented

Create VLAN → Get in VLAN → Assign IP address

In this next example we will have 1 L3 switch doing the L3 routing. With 2 VLANs each one with a SVI

Firs start by enabling routing with

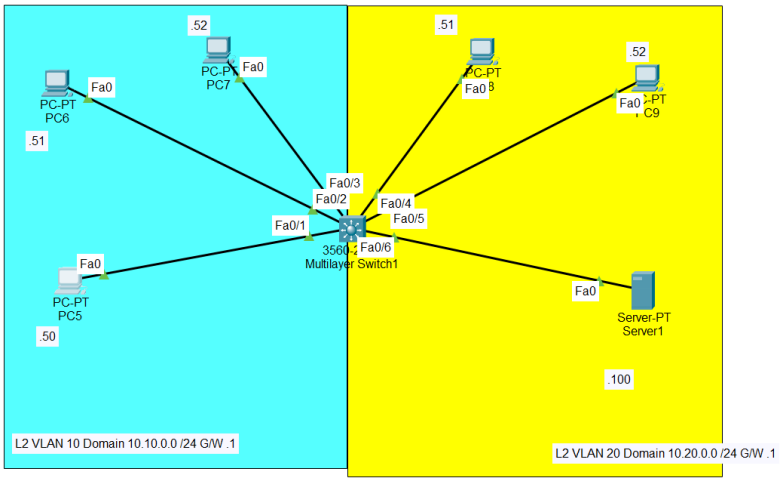
```
Switch(config)#ip routing
```

After creating both our VLANs, we go ahead and create 2 SVIs

```
Switch(config)#int vlan 10
Switch(config-if)#ip address 10.10.0.1 255.255.255.0
Switch(config)#int vlan 20
Switch(config-if)#ip address 10.20.0.1 255.255.255.0
```

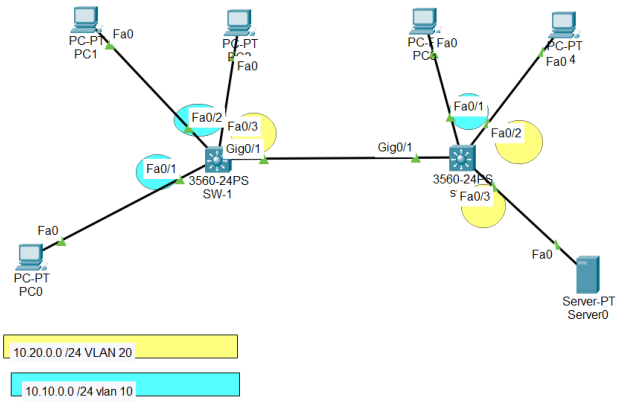
REMEMBER TO ADD ALL THE PHYSICAL INTERFACES TO THE VLANS THEY ARE SUPPOSED TO BE CONNECTED TO

Note: SVIs also work in L2 switches inside a VLAN so the switches at L2 know how to forward IP packets based on destination ip addresses.



33.- Configure Cisco 802.1Q Trunking

In this module we deal with VLANs interconnected between different switches. It all starts with setting a VLAN trunk port.



In our topology, we want to make interfaces Gig0/1 on both L3 switches as Trunk Ports using 802.1Q.

```
switch(config-if)# switchport trunk encapsulation dot1q
switch(config-if)# switchport mode trunk
switch(config-if)# no shutdown
```

Remember that Trunking is all about routing VLAN traffic from one physical port to another.

In other words, trunking is about EXTENDING THE L2 BROADCAST DOMAIN ACROSS 2 OR MORE SWITCHES.

The Cisco Native VLAN

As we have seen before, the VLAN all ports are associated with is VLAN 1. The NATIVE VLAN does not get TAGGED when sent through a TUNK PORT. Ports associated with VLAN 1 are in a different broadcast domain than the rest of the VLANs, as we already know. So let's consider an example where VLAN 1 get a domain network of 10.99.99.0 /24. When a trunk port receives traffic with no 802.1Q portion, that means that it belongs to the NATIVE VLAN.

In a conventional way, we have to let switches know what is their NATIVA VLAN when configuring TRUNK PORTS.

```
Switch(config)#int g0/1
Switch(config-if)#switchport trunk encapsulation dot1q
Switch(config-if)#switchport mode trunk
Switch(config-if)#no shut
Switch(config-if)#switchport trunk native vlan 1
Switch(config-if)#switchport trunk native vlan 3
```

```
Switch#show int trunk
Port        Mode         Encapsulation  Status        Native vlan
Gig0/1      on           802.1q         trunking      3
```

```
Switch#show int trunk
Port        Mode         Encapsulation  Status        Native vlan
Gig0/1      on           802.1q         trunking      3
```

Controlling Which VLANs Are Allowed on the Trunks

By default all VLANs are allowed but we can set some to not be able to be allowed over the trunk. In our example let's disable VLAN 20 traffic. We have multiple options to how we want to tell the CLI to remove, add, except a VLAN

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

So let's specify with the 'except' command that we want VLAN 20 to not be forwarded:

```
Switch(config-if)#switchport trunk allowed vlan except 20
```

Inter VLAN Routing

In this section we see how to combine our previous skill knowledge of Inter VLAN routing with a SVI. Remember that SVIs 'live' inside VLANs each assigned with IP addresses, at the end of the day, they ARE interfaces. Remember that we have about 2 ways of considering routing:

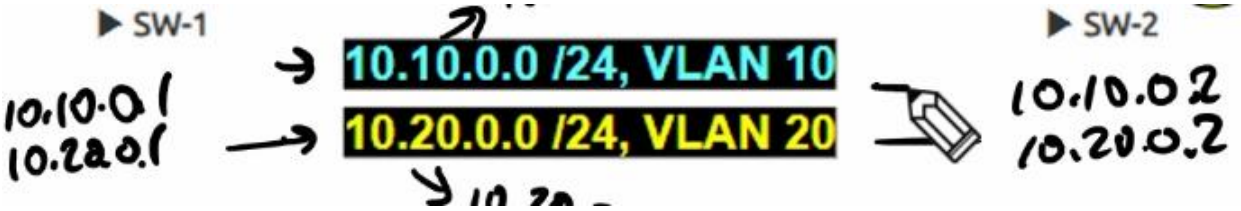
1.- Assign physical connections that will serve routing to every single VLAN.

2.- Router on a stick.

But when we have a L3 SWITCH we can simply create SVIs that lives in a VLAN and that will also serve as our DEFAULT GATEWAY.

So in our example, we create 2 SVIs each for VLAN 10/20 with IP ADDRESSES 10.10.0.1 ---------- 10.20.0.1

Since we have 2 L3 switches there are several ways to do this. We can put all the default gateways on either switch or we could create 2 different sets of D/Fs for each switch.



So each switch will have a SVI for each VLAN that will serve as a default gateway for every host connected to it.

Switch 1

```
Vlan10                 10.10.0.1       YES manual up                    up
Vlan20                 10.20.0.1       YES manual up                    up
```

Switch 2

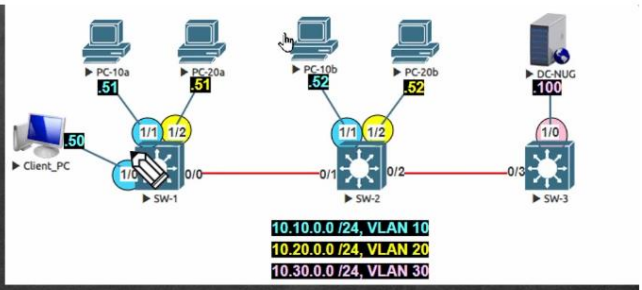
```
Vlan10                 10.10.0.2       YES manual up                    up
Vlan20                 10.20.0.2       YES manual up                    up
```

VALIDATION/LAB

802.1Q Trunking Lab

- Configure trunks with:
  - Encapsulation 802.1Q
  - Between:
    - SW-1 0/0 ← → 0/1 SW-3
    - SW-2 0/2 ← → 0/3 SW-3
  - Set native VLAN on trunks to 10
  - Allow only VLANs 1, 10, + 30 on the trunk

- Testing:
  - Ping over VLAN 10
    - PC-10a to PC-10b (10.10.0.52)
  - Ping over VLAN 30
    - Client PC to Server (10.30.0.100)
  - Attempt a Ping over VLAN 20
    - PC-20a to PC-20b



34.- Configure Cisco DTP

Dynamic Trunking Protocol is to negotiate trunking. DTP negotiates the ENCAPSULATION, and whether or not a trunk needs to be in place.

Negotiate ENCAP →

```
Switch(config-if)#switchport mode dynamic auto
```

IF BOTH ARE AUTO – AUTO NO TRUNK WILL BE FORMED

```
Switch(config-if)#switchport mode dynamic desirable
```

IF 1 IS AUTO – DESIRABLE A TRUNK WILL BE PLACED

IF BOTH ARE DESIRABLE – DESIRABLE THERE WILL BE A TRUNK

Also, if one is already configured as a TRUNK – AUTO/DESIRABLE a trunk link will be formed

By default on Cisco packet tracer, ports are designed by default as dynamic auto, so by default they will not form a trunk. In our example let's change the g0/1 port to DESIRABLE.

```
SW-1(config-if)#switchport mode dynamic desirable
```

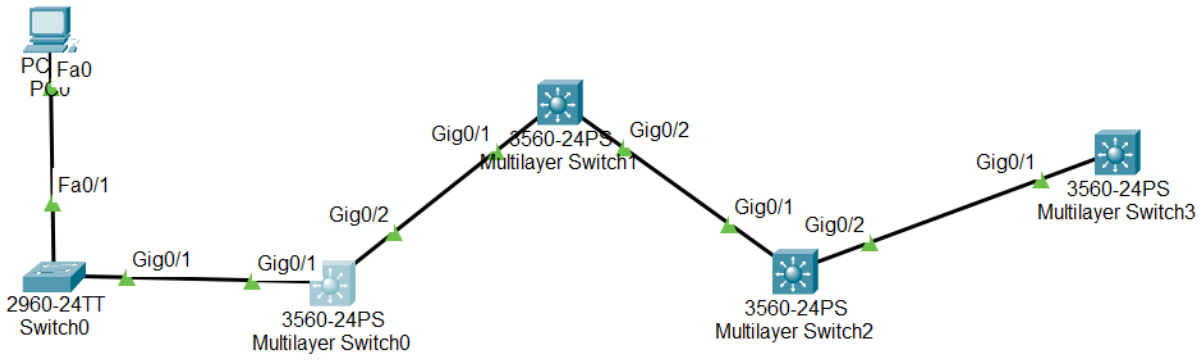
Now on SW-1 g0/1 and SW-2 g0/1 there will be a trunk link created.

```
SW-1#show int gig0/1 switchport
Name: Gig0/1
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
```

```
SW-2#show int g0/1 switchport
Name: Gig0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: trunk
```

Disabling DTP

Let's imagine that on our topology, we set SW-1 Gig0/1 as a trunk. Even though in our L3 Switch on G0/1



is set to dynamic auto, because DTP is still on, a trunk link is formed between the 2 ports.

```
Switch#show int g0/1 switchport
Name: Gig0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
```

NOW, IF WE SET A PORT TO BE IN A TRUNKING STATE, WE CAN THEN DISABLE DTP FOR GOOD.

```
Administrative Mode: trunk                    Administrative Mode: trunk
Operational Mode: trunk                       Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q  Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q     Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On                   Negotiation of Trunking: On
```

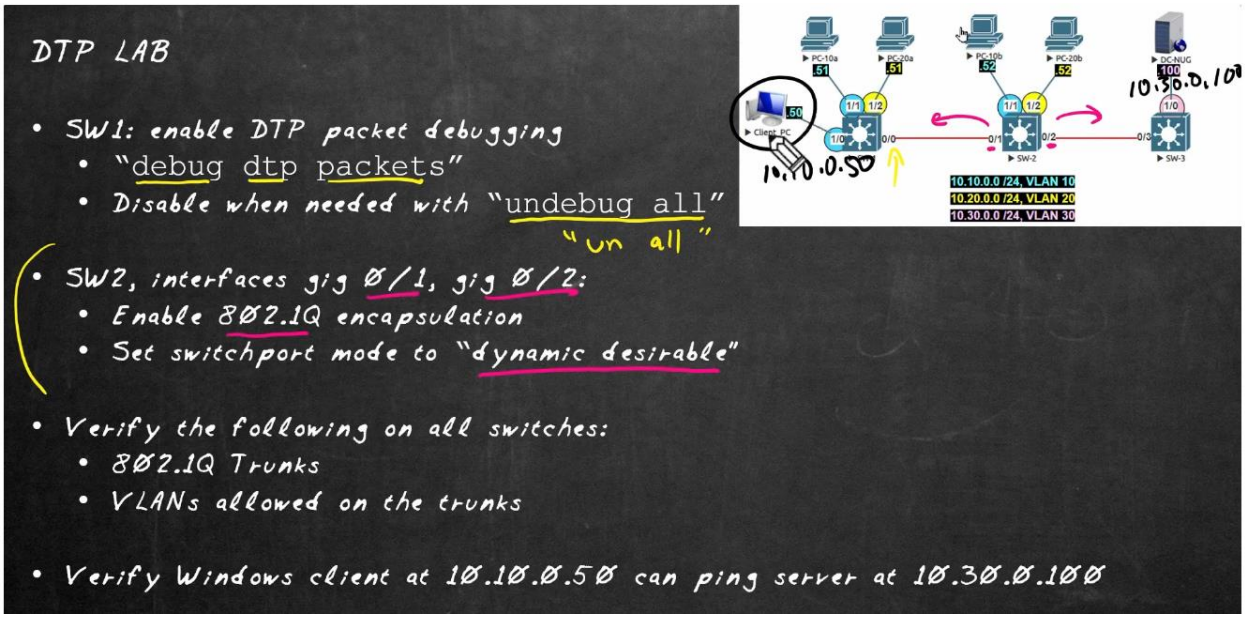
In here, I set both g0/1 on SW-1 & SW-2 as trunks. Then we can tell them to no negotiate to disable DTP

```
Switch(config-if)#switchport nonegotiate
```

```
Administrative Mode: trunk                    Administrative Mode: trunk
Operational Mode: trunk                       Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q  Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q     Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off                  Negotiation of Trunking: Off
```

NOTE: IF WE SET A PORT SIMPLY AS A SWITCHPORT IT WILL AUTOMATICALLY TURN DTP OFF

Confirm Access Ports and SVIs/lab validation



35.- VTP VLAN Trunking Protocol

What happens when one switch does not have a VLAN that needs to pass through it? The conversation gets completely dropped. One solution to maintain this conversation is to synchronize all the vlans via a VTP DOMAIN.

VTP works in existing trunk ports. If we create a new VLAN, this information can be transmitted to all the other trunking ports on all switches if they are part of the same VTP DOMAIN

```
Switch(config)# vtp domain OurVTP
Switch(config)# vtp password VTP-Password
```

So all switches are aware and 'asking' which one has the latest VTP information so they can all synchronize and be updated with the VLANs being created OR deleted.

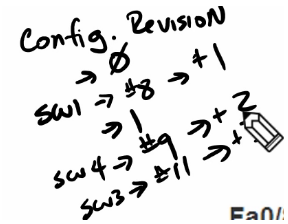
We can set modes for switches on a VTP domain to prevent accidental deletions or creations of VLANs.

VTP SERVER MODE → We can create/delete VLANs or FULL RIGHTS.

VTP CLIENT MODE → Clients get updated information regarding VLANs but can not create or delete VLANs.

TRANSPARENT MODE → The switch is not updating or delete or create VLANs based on what is going on the DOMAIN. Basically it does not update it's own files about it's VLANs.

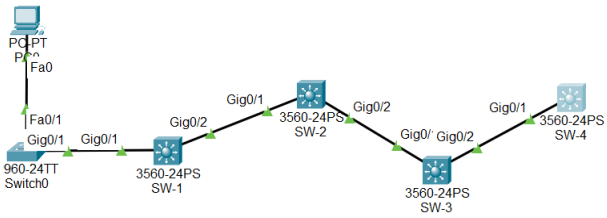
OFF mode → Not propagating or updating VLAN information on the DOMAIN.



The way the DOMAIN keeps track of the latest VLANs information is by using CONFIGURATION REVISION which is the variable the domain uses to add a change to the VLAN configuration in the domain and make this number shareable between all the switches for comparison.

MAKE SURE THAT WHEN CONNECTING A NEW SWITCH INTO A VTP DOMAIN, IT'S CONF REVISION NUMBER IS LOW, OTHER WISE EVERY OTHER SWITCH ON THE DOMAIN WILL WANT TO UPDATE TO THE VLAN INFORMATION THIS NEW SWITCH HAS.

Configure a VTP Domain



Calling our domain OurVTP and password OurPassword where the mode for all the switches will be Server mode.

Notice how by default the switches are on mode server VTP. We would need to let all switches be part of the domain by using the command on all of them:

```
Switch(config)#vtp domain OurVTP
```

```
Switch(config)#vtp password OurPassword
```

```
Switch(config)#do show vtp status
VTP Version capable             : 1 to 2
VTP version running             : 1
VTP Domain Name                 : OurVTP
```

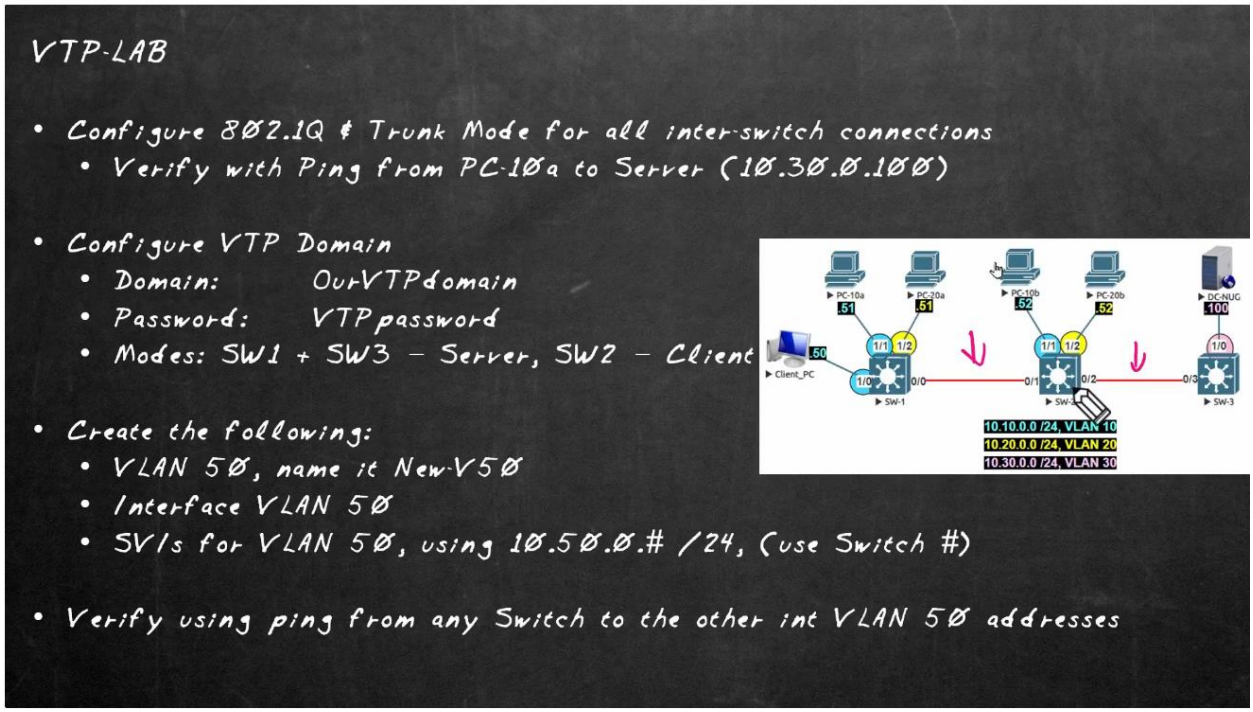
```
Switch#show vtp status
VTP Version capable             : 1 to 2
VTP version running             : 1
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0060.2F21.A000
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
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        : 5
Configuration Revision          : 0
```

At this point all of the switches will have a Conf Revision of 0 since no VLANs have been created. So let's create some VLANs on SW-1 and see how the domain changes. By creating only 1 VLAN 777 on SW-1 called Lucky-Demo, we will see both the Config Revision changing to an extra one to what it was on all of the switches as well as VLAN 777 created.

Testing with SVIs

Remember that when working with SVIs(L3) inside VLANs (L2) we provide IP addresses to the SVIs. We'll create a new VLAN 7 called SVI-TEST and provide an IP address of 10.7.0.[] /24. Now we should be able to ping to every SVIs' IP address on every switch.

Validation/lab

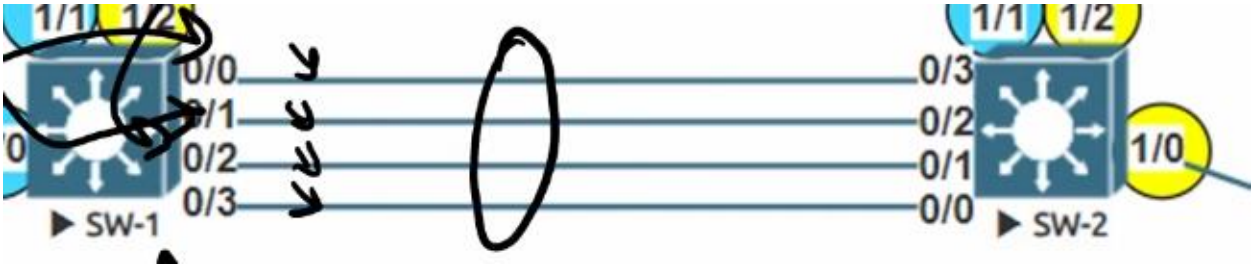


36.- Configure Cisco EtherChannel

REMEMBER THAT FOR L3 SWITCHES IF WE WANT A TRUNK PORT, TO SAY

```
Switch(config-if)#no switchport
```

Basically EtherChannel 'bundles' together several trunk connections.



We let the switches decide how they want to bound the traffic. We can enable this bundle to be a huge L2 Trunk, Access port, or even a L3 routed interface by saying 'no switchport'.

> Benefits: Fault tolerance where if one cable fails to function, the other can continue carrying traffic.

There is a concept called LOAD BALNCING which is a decision on 'how' the switch routes the traffic on it's different ports.

When configuring ether channel between 2 devices, we have options regarding the protocols for negotiation:

- ➔ PAgP: Port aggregation protocol. Old protocol. Used when links do not support the most recent protocol. Mode are Desirable and Auto. Which follows the same rules of thumb than the DTP. DESIRABLE START THE NEGOTIATION AND AUTO IS WILLING TO ACCEPT

  DES – AUTO ✓

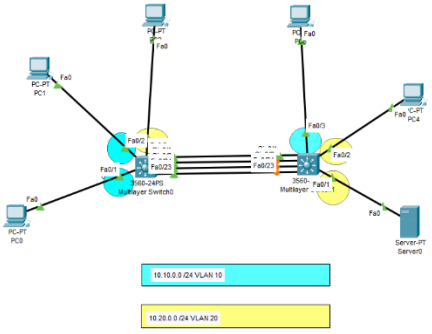
  AUTO – AUTO ✗

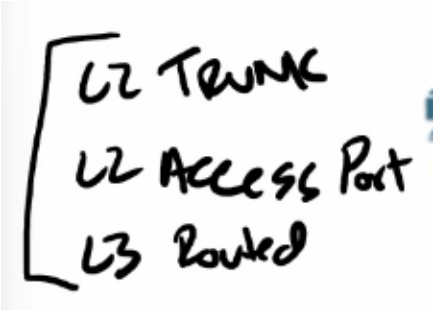
  DESIRABLE – DESIRABLE ✓

- ➔ LACP: Link aggregation control Protocol. Modes: ACTIVE starts negotiation PASSIVE willing to receive negotiation. SAME RULES APPLY from above

Keep in mind that for these protocols you can simply set the mode as ON TRUNK/PAgP/LACP.

Let's say for our lab we have to implement EtherChannel between SW-1 and SW-2 to make them the following types of ports





1.- L2 Access port For VLAN 10 using PAgP in desirable

```
Switch(config)#int range f0/4-7
Switch(config-if-range)#Switchport mode access
Switch(config-if-range)#switchport access vlan 10
```

Then by saying:

```
Switch(config-if-range)#channel-group 1 mode desirable
```

Depending on the mode type and protocol we want that port to be as.

L2 EtherChannel Trunk Port

In last part we saw how to bundle ports into a big access port. Now, let's make it a TRUNK

```
Switch(config)#int range f0/4-7
Switch(config-if-range)#switchport trunk e
Switch(config-if-range)#switchport trunk encapsulation do
Switch(config-if-range)#switchport trunk encapsulation dot1q
Switch(config-if-range)#switchport mode trunk

Switch(config-if-range)#channel-group 1 mode active
```

Depending on the type and protocol we want the port to use.

L3 EtherChannel Routed Interface

Remember when we used 'no switchport' and that port becomes a L3 interface just like a router port? A similar concept we can do in L3 switches. So let's do L3-Etherchannel with LAcP as Active/Active.

In this case, we would need to do it differently!

- ➔ Create Po# to create the port channel interface and indicate no switchport
- ➔ Assign the IP ADDRESS
- ➔ We go into CONFIG-RANGE for our port

➔ No switchport and assign the channel group that we just created.

CREATE Po1 on both switches and provide a logical IP address

```
Switch(config)#interface port-channel 1
Switch(config-if)#no switchport
Switch(config-if)#ip address 10.12.0.1 255.255.255.0
```

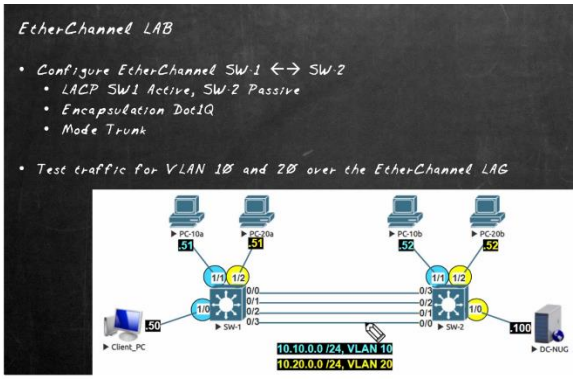
Then we go into our RANGE ports and say unite them into Po1

```
Switch(config)#int range f0/4-7
Switch(config-if-range)#no switchport
```

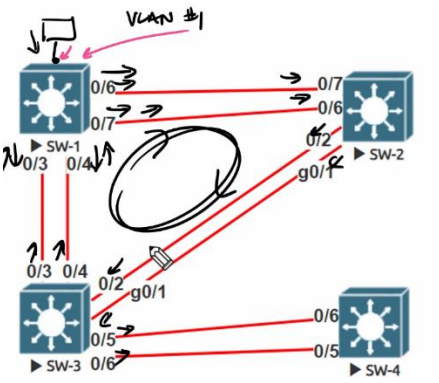
```
Switch(config-if-range)#channel-group 1 mode active
```

```
Switch#show etherchannel summary
Flags:  D - down        P - in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port


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

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------

1      Po1(RU)           LACP    Fa0/4(P) Fa0/5(P) Fa0/6(P) Fa0/7(P)
```

VALIDATION/LAB

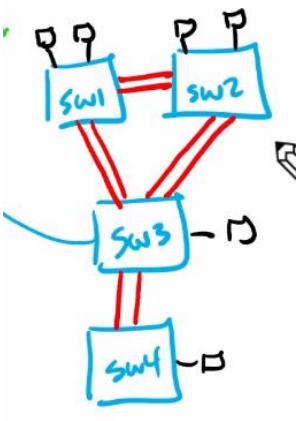


37.- Understand Cisco STP



Let's imaging that these ports are trunk now. And let's say that a traffic for broadcast from VLAN 1 comes in into 1 of the switch and gets forwarded by all of them nonstop. This creates a loop that causes many problems. This is known as a BROADCAST STORM. This causes a network mostly unusable.

The main problem is solved by STP. It looks at the topology of a network and starts BLOCKING some ports. So it prevents L2 loops.



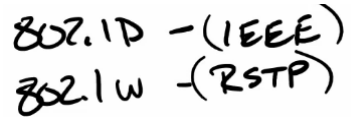
The way STP does this is through little messages called BPDU (BRIDGE PROTOCOL DATA UNIT). This messages are being sent constantly to prevent loops from being formed.

The best way to imagine how STP works is by picturing a tree with several branches that at the beginning of these branches we have our switches. STP prevents parallel lines of communication and that's the goal. So for example if we create more parallel lines of communication between multiple switches like in the picture, STP will identify these and block them when needed. STP also has fault tolerance to identify when it is appropriate to restate a link when it was blocked.

Cisco and Spanning Tree

The standard for STP is 802.1D originally, then 802.1W came. We can verify which one a switch is running by sending the command `Switch#show spanning-tree`

INSTANCES OF STP WILL BE RUNNING PER VLAN



If it appears that we are running IEEE or RSTP that means we are running the flavor showed on the left. In our case, we are running 802.1D (IEEE)

```
Switch#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
```

Electing a Root Bridge for Spanning Tree

If we want a loop-free topology, STP identifies a ROOT which is going to be called the ROOT BRIDGE. The ROOT is going to be a switch in this case. The root searches for the lowest bargain which means THE SWITCH WITH THE LOWEST BRIDGE ID. Behind the scene, the switches use BPDU to identify these bridge IDs, so let's see what is the bridge ID.

BRIDGE ID → Made of

Priority: By default 32,768 + [VLAN NUMBER] + [BASE MAC ADDRESS]

On our topology, you can see the root ID of the topology on different switches.

```
Switch#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID     Priority    32769
              Address     0002.1722.068E
This bridge is the root
```

in the case of SW-1 it won the root.

So in conclusion, the BRIDGE ID for SW-1 is the lowest of all 4 switches.

Port Roles and States

Two important concepts are

1.- ROLES

ROOT PORT → Forwarding towards the ROOT. These will only be found on switches that are NOT the root. They all are going to have a port forwarding towards the ROOT.

DESIGNATED PORT → Forwarding AWAY from the root. These can exists on both switches and the root.

We also can have alternate ports that STP decides to whether or not put it to forward towards the ROOT.

2.- States:

ROLES	STATE
ROOT	1.- Forwarding (FWD) → Actively forwarding towards the ROOT 2.- ✖ Not forwarding → These can be as blocking (BLK)
DESIGNATED	1.- Forwarding (FWD) → Actively forwarding towards the ROOT 2.- ✖ Not forwarding → These can be as blocking (BLK)

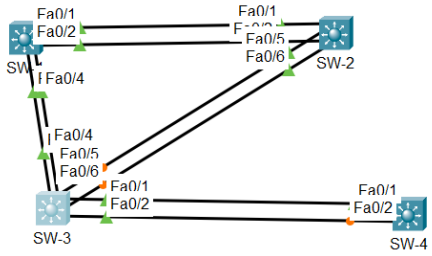
Non-Root Bridges Get 1 Root Port

Now we need to identify the Root Ports. In order to identify ROOT PORTS, switches do so by looking at the lowest cost. Remember that STP uses BPDUs to communicate STP information.

The ROOT BRIDGE has a cost of 0 so don't bother about choosing ROOT PORTS. BPDUs start at the ROOT BRIDGE and spread out to the other switches. For fast ethernet interfaces the base cost is 19 and for gigabit is 4.

So the formula is this: LOCAL COST + ADVERTISEMENT COST

So let's take our topology as an example:



It starts at the root bridge SW-1 with a cost of 0 propagated to the SW-3 AND SW-2. When the BPDU arrives at the fast ethernet port, it adds it's local cost (19) + the adv cost of 0 from the previous hop for a total of 19 cost. Now, it spreads out this information again. When it arrives to SW-3 from SW-2 it says 19 + 19 = 38 vs the 19 cost it came directly from SW-1. So then the switch decides on the ROOT port to be either Fa/03 or04 in this case.

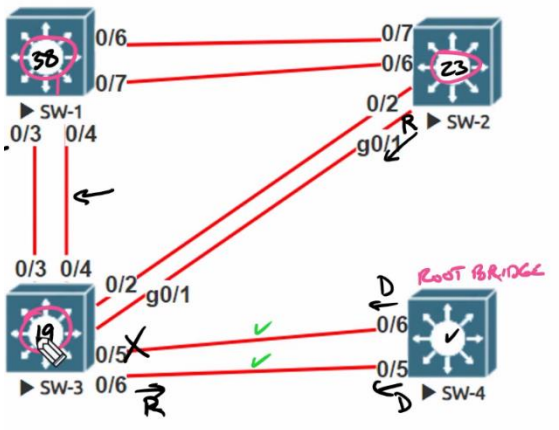
If the case of a tie comes, the tie-breaker will be:

1.- UPSTREAM SWITCH with the lowest Bridge ID. Remember that Bridge ID is the number all switches assign themselves during STP to identify the BRIDGE ID or ROOT.

Priority: By default 32,768 + [VLAN NUMBER] + [BASE MAC ADDRESS]

2.-if there is another TIE. The Advertised port priority. By default is 128.

3.- Advertised port number. Meaning f0/3 vs f0/4. 3 would be the lowest.



Designated Ports and Ports That Will Block

Now, we get to have a look at Designated ports. All the ports from the ROOT Bridge are going to be designated because they all forward away from the root.

For NON-ROOT BRIDGES (All the other switches other than the root) will have their designated ports by the role of which one has the lowest cost to reach the root. We have to think about network segments which means the segment in the network between 2 switches. In our topology would mean the links between each switch.

So the rule of thumb is: For a network segment, the switch with the lowest cost will have the DESIGNATED PORTS.

Between SW4 ---- SW3 ➔ Lowest is SW-4

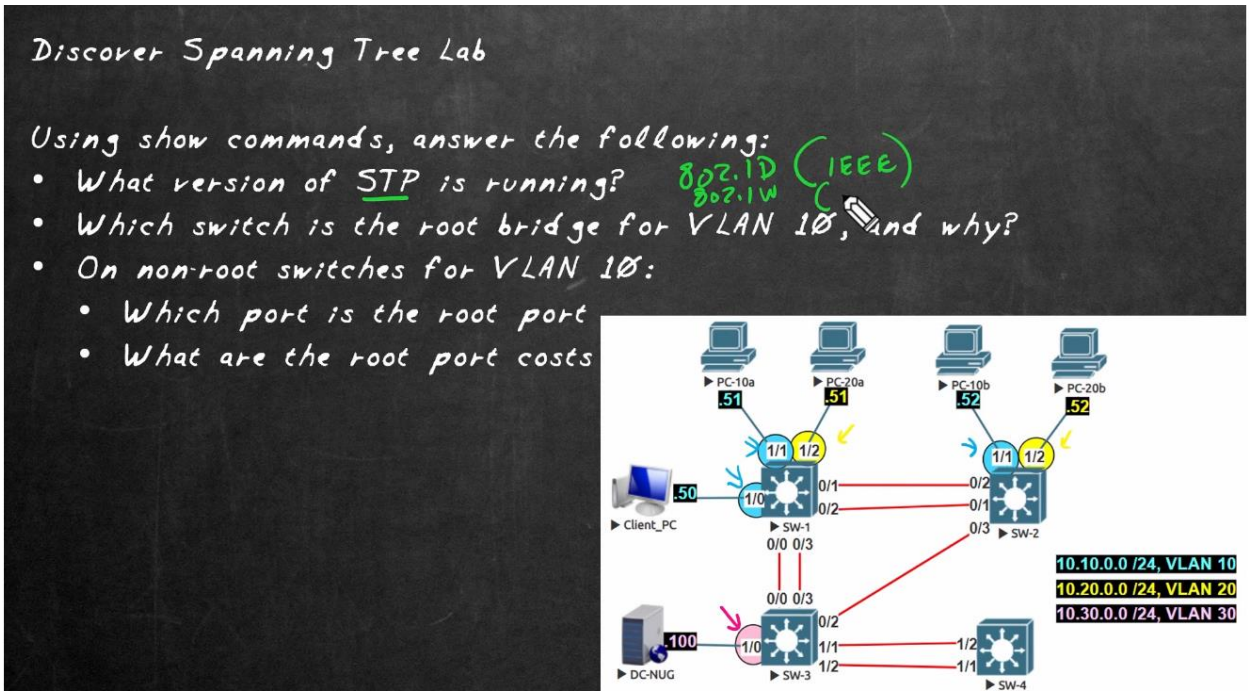
Between SW3 ---- SW1 ➔ Lowest is SW-3

Between SW3 ---- SW2 ➔ Lowest is SW-3

Between SW2 --- SW1 ➔ Lowest is SW-2

Once STP has put together all the Designated and Root ports, all the other ports will be put in a BLOCKING or DISGARDING state depending on the STP protocol using.

Validation/lab



1.- 802.1D IEEE

2.- The root bridge for VLAN 10 is SW-4 because it had the lowest Bridge ID number.

3.- For VLAN 10 on nonroot switches:

SW-3 ➔ G1/2 → Cost 4

SW-2 ➔ G0/3 → Cost 4

SW-1 ➔ G0/0 → Cost 4

38.- Configure Cisco STP

Recap → We got 2 main flavors of STP

802.1D IEEE

802.1W Rapid

The ROOT BRIDGE is the switch that is going to be the ROOT in the topology and is decided by the lowest BRIDGE ID [Priority + VLAN + MAC]

The root ports on each switch are going to be assigned by the port with the lowest cost and by the tie-breaker steps.

The designated ports will be assigned through each of the network segments between the switches. For the one who has the lowest cost. Remember that the cost comes from the ADVERTISEMENT COST + LOCAL COST.

Spanning Tree Modifications We May Want to Use

Delay → The default forward delayed timer is 15 seconds for any BPDU traffic coming in. After the first 15 seconds, the port will be put in a 'listening' state.

So 15 seconds listening for any BPDUs traffic coming in, 15 seconds for learning about MAC addresses. Then STP will put that port in a DESIGNATED FORWARD state.

WE CAN MODIFY THESE TIMERS.

We can use PORTFAST → It basically skips the listening and learning phase completely. The downside of portfast is that since you are basically telling STP to not pay attention to it's protocol of going through the listening and learning phases, there are potential loops that can be created it there is another switch being connected. You can enable portfast per vlan as:

```
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# spanning-tree portfast
```

You can enable PORTFAST GLOBALLY. So any port designated as switchport access, they will immediately go into PORTFAST mode.

```
Switch(config)# spanning-tree portfast default
```

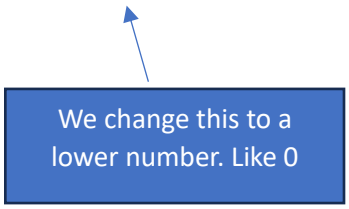
We can also enable PORTFAST on TRUNKs. By additionally adding the keyword

```
Switch(config-if)# spanning-tree portfast trunk
```

RAPID STP → Just goes straight to LISTENING state without learning state. If we are afraid that any BPDU traffic will come through a port with RAPID STP, we can enable BPDU Guard that will automatically block BPDU traffic on a access port. [Error disabling state]

We can also administratively control which switch we want to be the Root Bridge. And we do this by changing the PRIORITY variable that remember by default is 32,768.

BID ➔➔ PRIORITY + VLAN + MAC



Configure Port Fast on an Interface

Controlling the STP Root

When we tell a switch to be a root of a VLAN, since they know which switch is the actual root, they will simply lower the BRIDGE ID. The way we do this is by:

```
Switch(config)#spanning-tree vlan 10 ?
  priority  Set the bridge priority for the spanning tree
  root      Configure switch as root
  <cr>
```

We can configure a primary and secondary in case one switch goes off.

```
Switch(config)#spanning-tree vlan 10 root primary
```

Let's look at the priority for VLAN 10

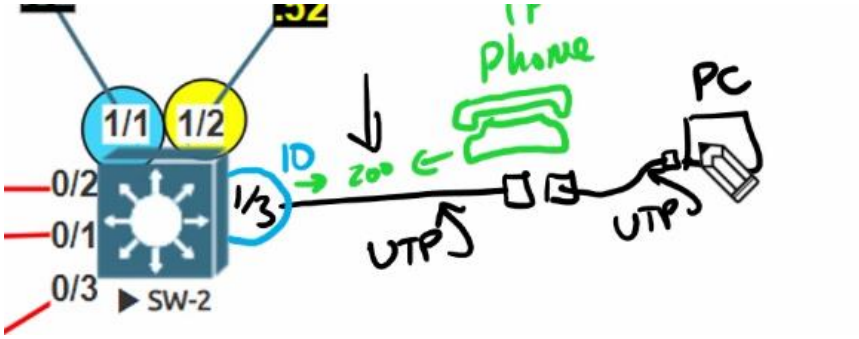
```
Switch(config)#do show spanning-tree vlan 10
VLAN0010
  Spanning tree enabled protocol ieee
  Root ID    Priority    24586
```

So switches simply 'beat' the BRIDGE ID of the switch that is the root of that particular vlan.

We can also set the priority manually with

```
Switch(config)#spanning-tree vlan 10 priority ?
  <0-61440>  bridge priority in increments of 4096
```

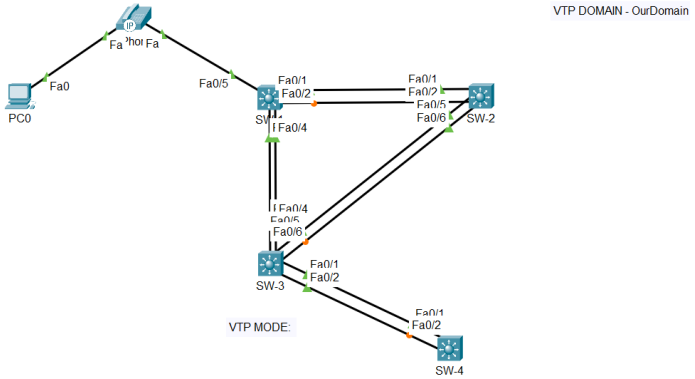
Configure Cisco Voice VLANs



It is convenient to isolate traffic for Voice devices into a VLAN. Even if making a port connected to an IP phone a ACCESS PORT would work, most likely there is another user pc along with the IP phone. We can connect the ethernet port of our PC into the phone and with an extra command, we can tell the switch port to tag 801.2Q tagging the voice traffic only and tagging the whatever vlan our pc is in.

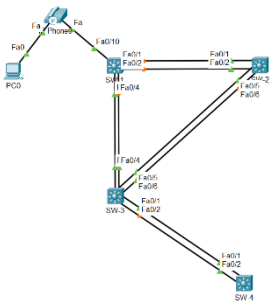
```
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# switchport voice vlan 200
```

Implementing a Voice VLAN



39.- Configure Cisco CDP and LLDP

CDP & LLDP is a L2 discovery protocol (they operate at the data link layer) that helps us identify connections between devices. CDP since it is from Cisco, would be great if we had all Cisco devices. CDP sends messages sporadically to help devices identify each other in networks. They advertise a bunch of information that we will see later on. Other vendors do not support CDP, instead they use LLDP. We can turn LLDP on in our Cisco devices as well to work with other vendor's equipment. By the way, when working with LLDP and wanted to connect to a VOICE VLAN IP phone, we use a capability called MED.



CDP Defaults

With this command we can see the timing between sending CDP traffic

```
Switch#show cdp
Global CDP information:
    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled
```

If we want to verify all our connections:

```
Switch#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
Device ID    Local Intrfce   Holdtme    Capability   Platform    Port ID
Switch       Por 1            177                    3560        Fas 0/1
Switch       Por 1            177                    3560        Fas 0/2
Switch       Por 1            177                    3560        Por 1
IP Phone     Fas 0/10         149            H P     7960
Switch       Fas 0/3          151                    3560        Fas 0/3
Switch       Fas 0/4          151                    3560        Fas 0/4
```

We can see all of the neighbors that SW-1 is getting CDP info from. The POR 1 means the etherchannel group 1 we created.

This is a great way to help us map a network in case we don't have a drawing representation.

Use CDP to Learn the IP of a Neighbor

In case we forget information about our network, we can go into more detail using CDP to identify all the information neighbor devices are sending. There are a couple of different ways:

With `SW-1#show cdp entry SW-2`

We can see a lot of information about SW-2 like IP addresses, interfaces involved, VTP domain.

Customize CDP

We can customize or change the behavior of CDP.

1.- We can turn it off -→ `SW-1(config)#no cdp run`

And to enable it back on → `SW-1(config)#cdp run`

2.- We can disable it interface by interface as well → `SW-1(config-if)#no cdp enable`

3.- We can also change the HOLDTIME for the entire box →

```
SW-3(config)#cdp holdtime 60
```

Enable LLDP

LLDP is not enabled by default in CISCO. To enable it we do:

```
SW-1(config)#lldp run
```

```
SW-1(config-if)#no lldp transmit
```

```
SW-1(config)#lldp holdtime 105
```

```
SW-1(config)#lldp timer 25
```

For these 2 commands, the LAST one will increase the advertisement timer to 25 seconds. And the FIRST one will increase the hold timer.

We can also disable the advertisement for let's say IP addresses for security reasons. To do this, we go

```
SW-2(config)#no lldp tlv-select
```

and we can specify a whole lot of options to leave out when LLDP sends out information to it's neighbors.

In the case of an IP address, we select

```
SW-2(config)#no lldp tlv-select  management-address
```

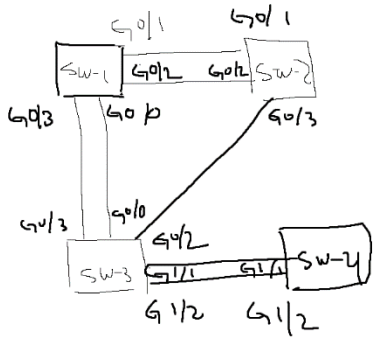
Validation/LAB

CDP and LLDP LAB

- Use CDP to document the connections between SW1, 2, 3, &4
  - From SW1, use CDP to discover the Mgmt. address of SW2
  - Test with a ping from SW1 to Mgmt. address on SW2
- Configure LLDP on all switches to match show LLDP:

```
Global LLDP Information:
    Status: ACTIVE
    LLDP advertisements are sent every 20 seconds
    LLDP hold time advertised is 65 seconds
```

- From SW4, use LLDP to identify:
  - Chassis ID and Mgmt. IP for SW3
- On SW3
  - Disable sending the Mgmt. IP info via LLDP
  - Disable sending and receiving LLDP on Gig 1/2




```
SW1#show cdp entry SW2
-------------------------
Device ID: SW2
Entry address(es):
  IP address: 10.10.0.2
```

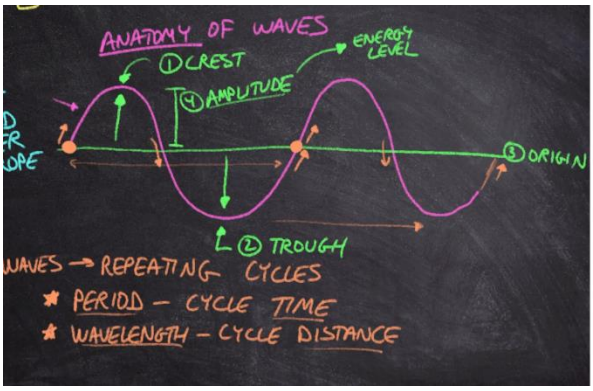
```
SW4#show lldp entry SW3

Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
----------------------------------------------
Local Intf: Gi1/1
Chassis id: 0cb4.d10c.0000
Port id: Gi1/2
Port Description: GigabitEthernet1/2
System Name: SW3

System Description:
Cisco IOS Software, vios_l2 Software (vios_l2-ADVENTERPRISEK9-M),
ARLY DEPLOYMENT DEVELOPMENT BUILD, synced to  V152_6_0_81_E
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2018 by Cisc

Time remaining: 50 seconds
System Capabilities: B,R
Enabled Capabilities: R
Management Addresses:
    IP: 10.10.0.3
```

```
SW3(config)#no lldp tlv-select management-address
```

Describe Wireless Principles

We are using light waves when we use Wireless technologies.

The anatomy of a wave is as follows:



41.- Describe Wireless Transmissions and Interference

Interference is when waves collide with each other in space. This could result in cancelling each other, which would result in no data being transmitted or in a lot of loss. This is if 2 waveforms were to be opposite exacts.

Sometimes we can have mutated waveforms that is going to be very difficult for a receiver to analyze it.

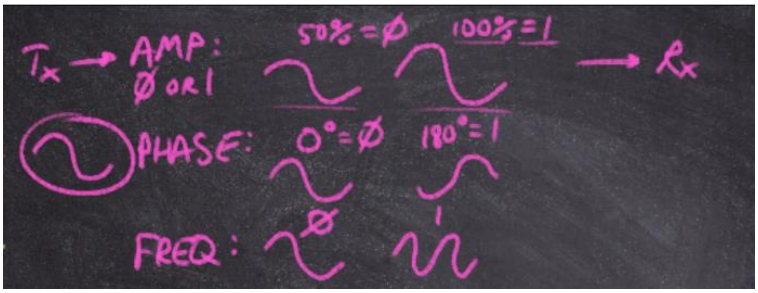
Let's consider 2 devices that use different frequencies techs like 2.4 and 5.0 GHz which ultimately would fix interference since both frequencies won't cancel each other.

Collisions and Overlapping Signals

Although Different frequencies cause mutated waveforms, the Rx receivers can take the mutated waveform and splits the 2 different frequencies. The process in place to achieve this is called FOURIER TRANSFORM

Modulation and Data Encoding

How do Tx and Rx interpret and are capable of put values of data into analog transmission? Since a simple waveform for a receiver can be interpreted as something different to another Rx. We use MODULATION for devices to change the properties of a waveform. But what properties can we change?



We can modify it's Amplitude --->

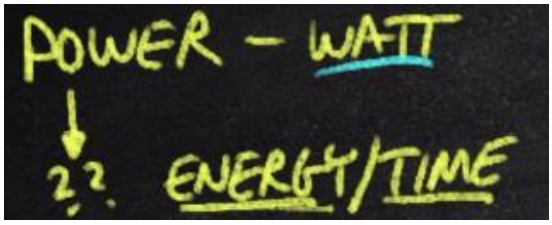
We can modify it's Phase -------->

We can modify it's Frequency ---> Speed up or decrease the frequency.

So devices can change rules of communication based on these variables.

42.- Calculate Wireless Measurements

Wireless Transmit Power



Power is measured in Watts. Power is a measure of energy over the course of time is being sent/received. Wifi sends wireless signals using some level of power. The levels for wifi are measured in mW or milliwatts 100 or lower. Since power is ENERGY/TIME the longer a signal propagates, the less power it will carry on, nevertheless a receiver can understand a 0.00001 mW power but there is a lower limit when this starts to become a problem.

Introducing the Decibel

A decibel is 1/10 of a Bel. The ide behind a decibel is to give us a more linear understanding of transmitting power. We use logarithms to make this conversion.

Decibels and Linearity

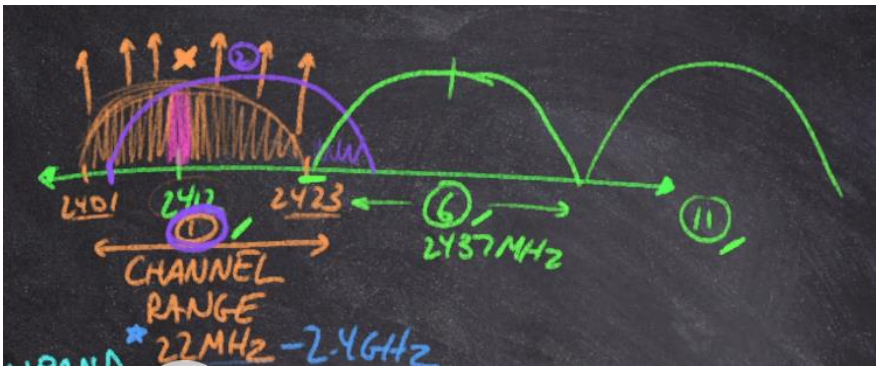
0 dBm = 1mW

43.- Explain Wireless Bands and Channels

Wireless Channels

Keep in mind that 1 GHz is 1000 MHz so even in 2.4 GHz bands, 2412 MHz is usual to see. Let's consider 2 different stations running at 2 different frequencies, let's say 2437 and 2412 MHz each, that should be good. What happens if a third device is sending data at frequencies at either one of those existing ones? This is called a POINT/NARROWBAND source of interference. Even it could happen that this third device works at ranges of frequencies (2400 – 2420 for example).

We can adapt the same concept for device 2 of sending data in ranges of frequencies. Although there will be interference exactly at 2412, the main range will be intact. This is the concept of a CHANNEL which isn't a single frequency but a RANGE of them.



For the 2.4 GHz frequency, we have 20 MHz channels in frequency size which translates to only 3 non overlapping channels.

Wireless Bands

So the bands are divided into channels. Let's break them down.

2.4 GHz – Original. It goes from 2.4 – 2.499 which only gives us around 72 MHz of possible frequencies that result in 3 non-overlapping 22 MHz and 11 that will overlap.

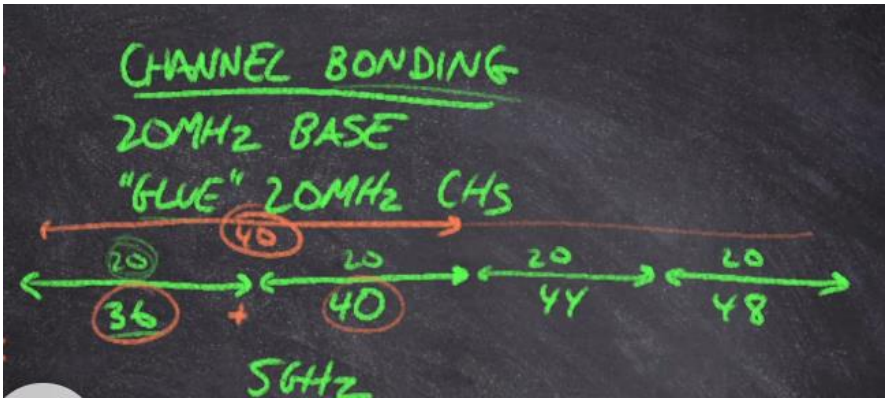
5.0 GHz → Improved and wider. It has a range of 500 MHz (5000 - 5500) and we get 25 non-overlapping channels. We categorized all these 25 channels into UNIBANDS.

6.0 GHz → We can use a bit of the 5 GHz and more of the 6 GHz frequencies so we get a lot MORE of channels. Around 59 non overlapping of a total of 1180 MHz for the entire 6.0 GHz.

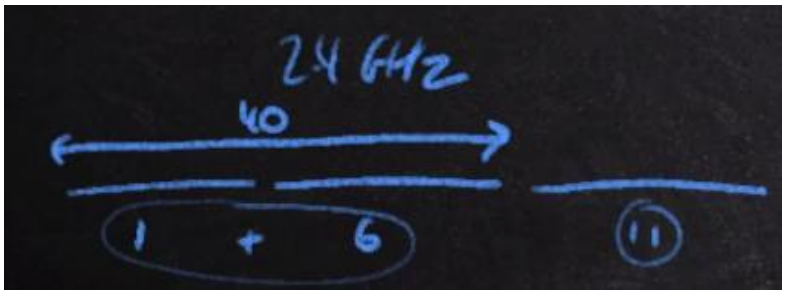
Bonded Channels

We can actually combine channels together and create more Throughput. If 1 channel has 20 MHz and we can add another 20 MHz channel of 40 MHz we can get a lot more throughput by doubling the channel width. We can use this concept with a lot more MHz's even up to 160 MHz making a single channel out of 8 times it's original size.

Let's say for example that in 5 GHz frequency where we have channels of 20 MHz in size, we can create 40, 80 or 160 MHz channels leveling down the number of channels possible each time (bad for congested areas in traffic) but adding more throughput.



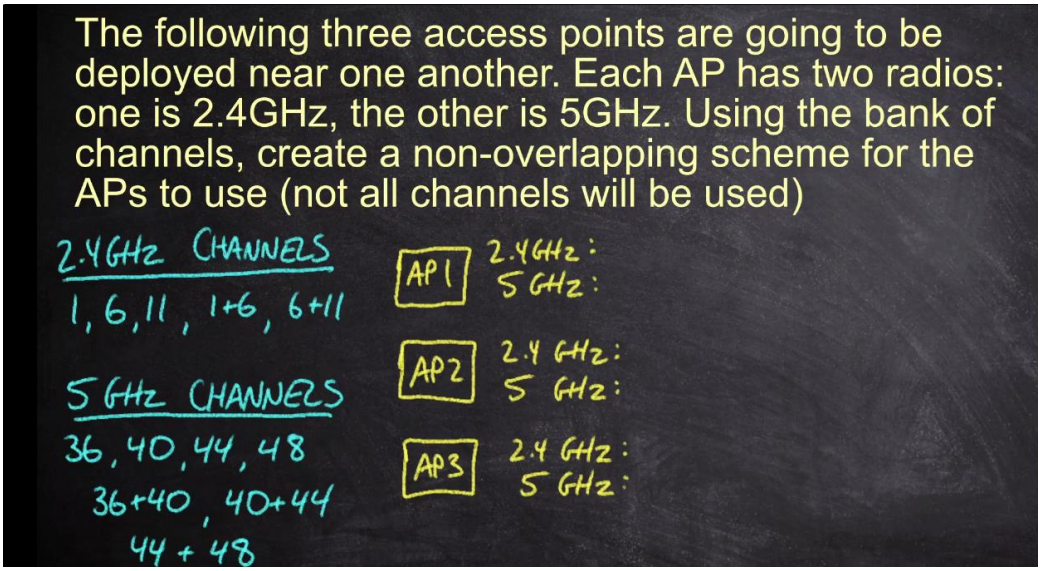
In 2.4 GHz frequency, we can only create even fewer throughput channels since the size of possible channels in nature is even lower than in 5.0 GHz.



Access Point Radios

An access point is a L1 and L2 device since it process ethernet frames that follow 802.11 standard. It has 2 ports, the first one is where all the devices connect to it. And the other one is where it connects to the LAN. The second connection can be a wired one.

VALIDATION



AP1: 2.4 = 1 5.0 = 36

AP2: 2.4 = 6, 5.0 GHz = 40 + 44

AP3: 2.4 GHz = 11. 5.0 GHz = 48

44.- Describe Wi-fi Standards

In Wifi standards exists so every device regardless of the vendor, can communicate. So IEEE has 2 major groups working for Ethernet protocols and Wireless protocols 802.3 and 802.11 respectively. The same concepts work on both, MAC addresses, IP addresses, and different L2 L3 protocols.

The WIFI ALLIANCE looked at the technology that IEEE were creating and noticed a vendor gap that you can't really test for all vendors if they are actually using a 802.11 'flavor'. So they created the Wifi that encompasses testing vendor products and guarantee trust. They also promote the name and the logo of Wifi and are basically the bridge between the consumer and the IEEE.

802.11 and 802.11a

802.11 the original came out in 1997. It allowed 1 Mbps – 2 Mbps.

> 802.11a came out in 1999. It introduced OFDM that got us upto 54 Mbps which is a huge gap compared to the original. It required the use of 5.0 Ghz band that made a lot of already existing devices running 2.4 Ghz to not be able to connect.

802.11b → 1999. Continue to use 2.4 Ghz and designed to be backwards compatible with 1 and 2 Mbps. It introduced 5.5 Mbps and 11 Mbps.

802.11g → 2003. Lasted for a lot of time. 2.4 Ghz and backwards compatible. Up to 54 Mbps.

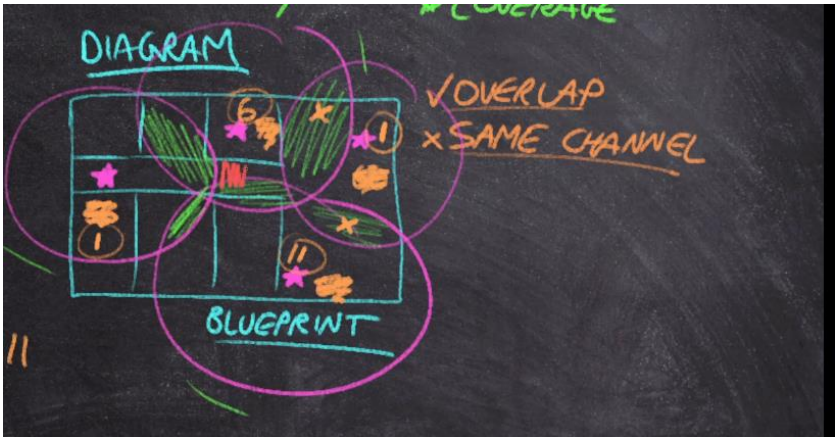
802.11n → 2009. Up to 600 Mbps. Both 2.4 and 5 Ghz bands. It introduced CHANNEL BONDING explained earlier for up to 20 / 40 Mhz channels. It also introduced MIMO meaning that now consumers were able to communicate with more throughput by adding more antennas and therefore increasing streams.

802.11ac → Wifi 5.

Describe Wireless Cells and Roaming

APs and Cells

When we talk about wireless signal we use a blueprint of the physical space we are going to be working with. We place circles that represent the area an AP is responsible for. We will want to make sure that we have enough coverage for all the physical space. Where these spaces overlap, it is ok to have these, but not SAME CHANNEL OVERLAP.



So For example on the image, we separate channel 36, 44 and 64 and it won't matter that there is a physical overlapping space as long as they are on different channels. When using larger frequencies like 5 or 6, we get more channels.

Noise Floor

Our focus is to have a good signal strength. To start with a BOUNDARY is an acceptable signal strength. Strength being measured in dBm, we need to make sure that the RECEPTION is good enough. Wireless noise is around us and needs to be taken into account when measuring what an acceptable signal strength is.

RSSI and SNR

So what makes a good signal?

To start with we measure the strength of a signal at the point of Rx and this is called the Received Signal Strength Indicator RSSI. We measure it in dBm. So let's say I receive a signal at -65 dBm.

But we have to consider NOISE in this. This is going to be evaluated in dBm. Ideally we want <= -90 dBm.

We compare the RSSI with the NOISE level and calculate the SNR or signal to noise ratio. Which simply is a division where we take the signal/noise.

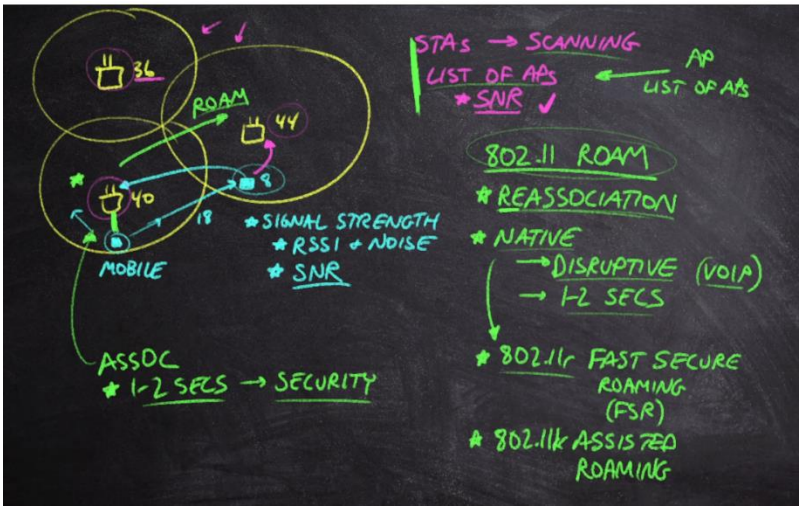
REMEMBER THAT dB are logarithmic functions and that represents exponents. If we divide exponents, it becomes a subtraction. 10 ^ 8 / 10 ^ 6 = 10 ^ 2

The same way if we take -64 – (-85) = +20 dB

So how much good enough? Cisco says that we need around 10dB. (non VOIP)

VOIP around 25 dB.

Station Roaming



What happens when a device gets to move around and still wants to be connected even though it goes out of the signal perimeter. Devices can track down a list of APs and consider the SNR of each AP. This is called 802.11 ROAM. In other words it makes a REASSOCIATION. Basically devices being connected wirelessly. On non VOIP connections, it lasts 1 – 2 seconds to readjust to the new AP but in VOIP connections 1 – 2 seconds to readjust is bad. 802.11r FAST SECURE ROAMING is made to jump security mechanisms to lower the waiting time. Another feature is 802.11k ASSISTED ROAMING is when an AP helps out a station and enables a list of Aps.

Service Set Identifiers (SSIDs)

SSIDs are identifiers that APs provide for the network names. It comes with a lot of properties like wireless security, network properties like bandwidth.

There are some limitations with this like ASCII (maximum of 32 characters).

Basic and Extended Service Sets

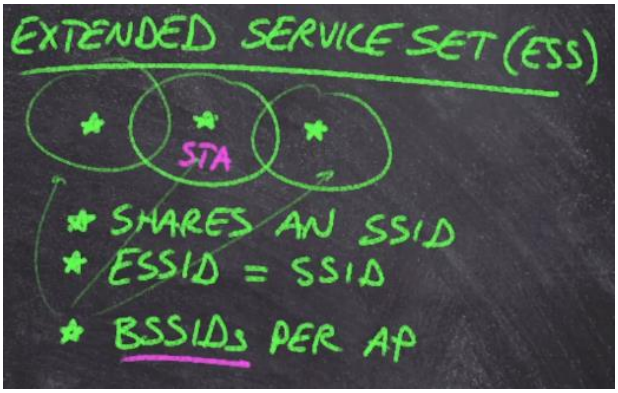
We have several service sets attached to the SSIDs like:

BSS → Basic Service Set. It defines a wireless network that is only broadcast by 1 AP.

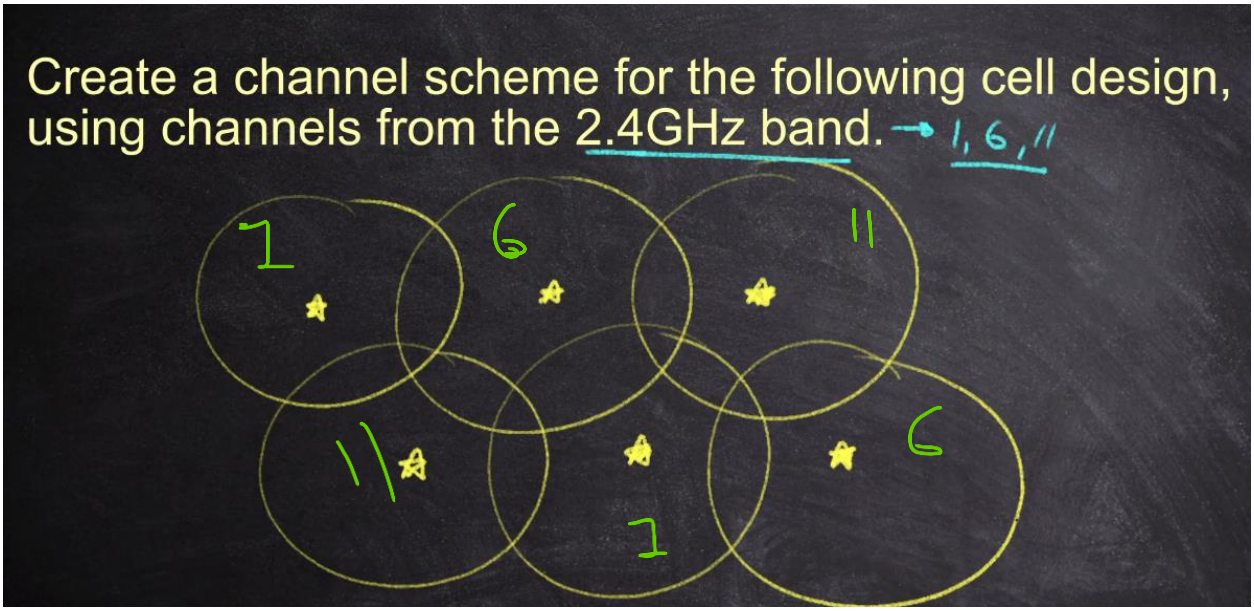
SSID → ASCII name of the network

BSSID → MAC address of the AP

ESS Extended Service Set → Several APs designed to work with one another. Every AP shares the same SSID. And they use the BSSID to know each separate AP inside the cluster.

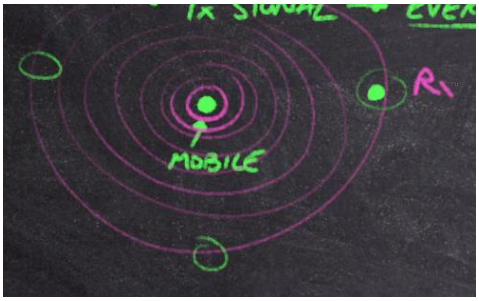


VALIDATION


Create a channel scheme for the following cell design, using channels from the 2.4GHz band. → 1, 6, 11
1
6
11
11
6
1

46.- Explain Wireless Security Principles

Wireless Security Challenges



One of the main problems with wireless is that we are working with waves. With wireless networking, threat actors can get in via the 'air'. It can both Rx wireless waves as well as create Tx them. So when we are talking about transmitting a signal, it transmit everywhere from the device (Depending on the transmit device).

From this, 2 main concepts join in: PRIVACY & ACCESS CONTROL.

Authentication and Encryption

Authentication is when we need to provide information to make sure that it is use who want to connect to a network. It is about proving identity.

How to prove identity:

- ➔ Username/Password

If a threat actor compromises these 2 variables, we need to go and cancel that account.

- ➔ Mutual authentication: Turns the focus around that is not the user. What happens when a user makes access to a threat's actor network (Man in the middle)? The idea is that the network provides a CERTIFICATION so the user knows it is connected to the right network.

Encryption → Masking data so that information can be transferred throughout a network. Encryption is basically scrambling text so that only Tx and Rx can understand what's being sent.

Wi-Fi Protected Access (WPA)

Having these 2 concepts in mind, we need a standard format within our wireless specifications. The WPA3 is based on a IEEE 802.11i which is used to maintain encryption and authentication.

WPA was not the first method to lock down wireless communications. 802.11 started with WEP, but it was very bad.

WAP introduced TKIP for ENCRYPTION and 802.1x for AUTHENTICATION.

WPA2 uses AES for ENCRYPTION and 802.1x for AUTHENTICATION.

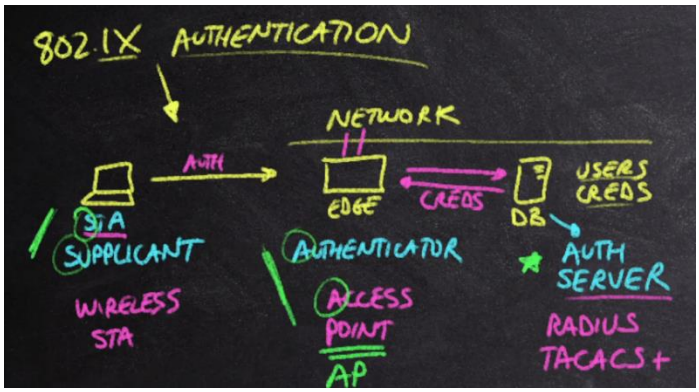
WPA3 patches some issues with WPA2 and provides high level security for certain scenarios and high-profile networks.

WPA Personal and Enterprise

We have several WPA standards. And it does not matter which one we are talking about, there will always be a:

- ➔ PERSONAL: Keep things simple. Used for residential, home applications. Small and medium businesses as well in terms of networks. To authenticate users, we use a pre-shared KEY known as PSK. The code inside WPA Personal provides encryption Keys and Prove ID.

- ➔ ENTERPRISE: Introduces more mechanisms like 802.1X + EAP. These set a robust mechanism to Authenticate and Encrypt data.



802.1X → Design for Authentication architecture. 802.1X defines a set of principles to do authentication. Imagine we have a network that consists of an edge device and database that stores user's credentials. The edge device is known as the AUTHENTICATOR which can be a switch or AP, and the server AUTH SERVER. The AUTH SERVER will use RADIUS or TACACS+ to authenticate.

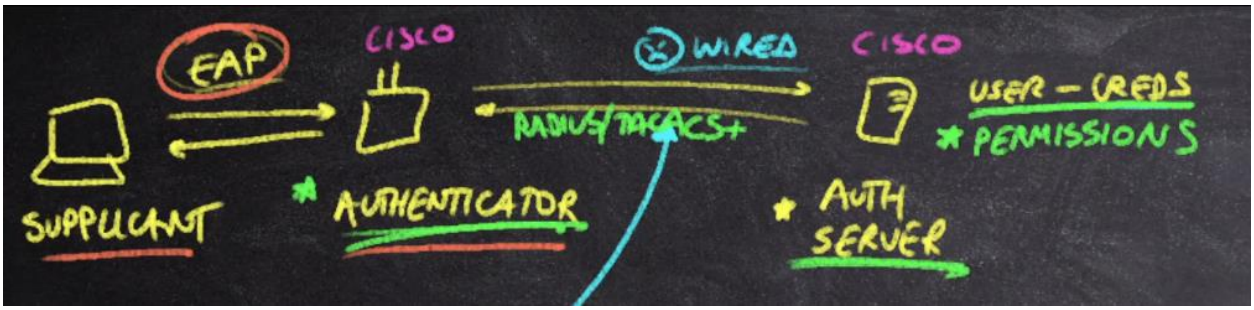
Another important part of 802.1X is the EXTENSIBLE AUTH PROTOCOL (EAP) → Which is a framework used for many other protocols to create EAP-TLS, EAP-TTLS, PEAP, LEAP, etc. EAP provides authentication mechanisms like USER/PASSWORDS, CERTS, etc.

RADIUS and TACACS+

EAP is not the only protocol needed for 802.1X to work. While EAP works during the Supplicant and Authenticator communication, on the backend of our server RADIUS or TACACS+ are working.

RADIUS → Relies on UDP and only the PASSWORD is encrypted.

TACACS+ → Created by CISCO but anyone can use it. It runs on TCP and all information is encrypted.

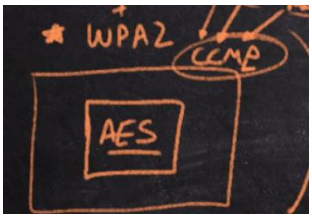


47.- Explain WPA Operation and Benefits

WPA Encryption

The mechanism used to scramble our data is called a CIPHER. A cipher is a set of rules that takes in inputs and applies these rules to output the ciphered message.

A cipher uses a KEY to encrypt a message and both Rx & Tx need this key to encrypt and decrypt.



Now, protocols are thought to be a layer outside the cipher, meaning that a cipher can work with different protocols. For example WPA which is the protocol uses RC4 which is the encryption cipher. WPA2 uses AES as an encryption cipher under CCMP protocol. (Remember that WPA2 is a standard with several components running to make this standard).

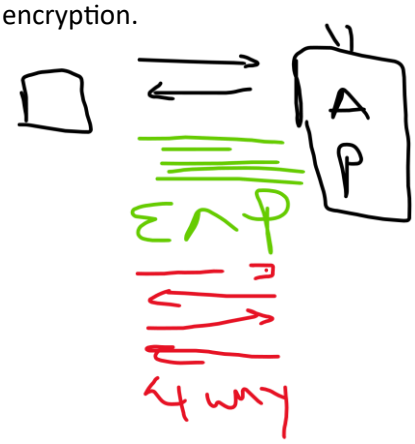
In essence, the core processes involved in WPA2 are:

- 802.1X or PSK (authentication)
- EAP (for enterprise)
- 4-way handshake (key management)
- AES-CCMP (encryption and integrity)
- Supplicant software (client-side)
- Access point software (router-side)

WPA 4-Way Handshake

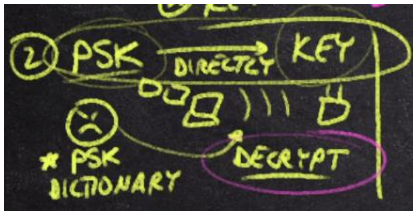
This is the way that both Tx & Rx can generate encryption keys. In essence, 4 messages are sent between Tx and Rx and it has to be confirmed that the keys are actually the same. But there is a difference between WPA personal and enterprise. On personal mode, we rely on a PRE shared key PSK that makes us not go through an authentication process like in enterprise.

On ENTERPRISE, since we use 802.1X with EAP where after the authentication process, EAP comes in place and manages the ways in which the user is going to authenticate. Whether it is through an USERNAME/PASSWORD or CERTS or OTP the AUTHENTICATOR needs to communicate with the server running RADIUS or TACACS+. We do ultimately do the 4 way handshake and keys are still being sent for encryption.



WPA3 Overview

Wpa2 had a couple of vulnerabilities like the KRACK attack that caused a threat actor to get the key from the 4 way handshake. WPA3 fixes this issue with a patch



Another vulnerability was that the PSK could get in the threat actors' hand through different methods, by brute forcing, library's help. Therefore with the key, being able to DECRYPT the communication. WPA3 still needs a passcode but changes the name of it. Instead of calling it a PSK, it is called SAE. Essentially performs authentication with different encryption keys and therefore, making it very hard to decrypt without a second key.

A third problem is that in OPEN NETWORKS, no authentication is used. This means that no encryption is also not in place. NO FIX but with another technology called OWE.

WPA3 still uses CCMP + AES.

Simultaneous Authentication of Equals (SAE)

We replace the PSK ----------> SAE. It is really a passcode. Which may seem as not accomplished much, but when you use PSK (WPA2) it performs 2 tasks of authentication and to derive the ENCRYPTION KEY.

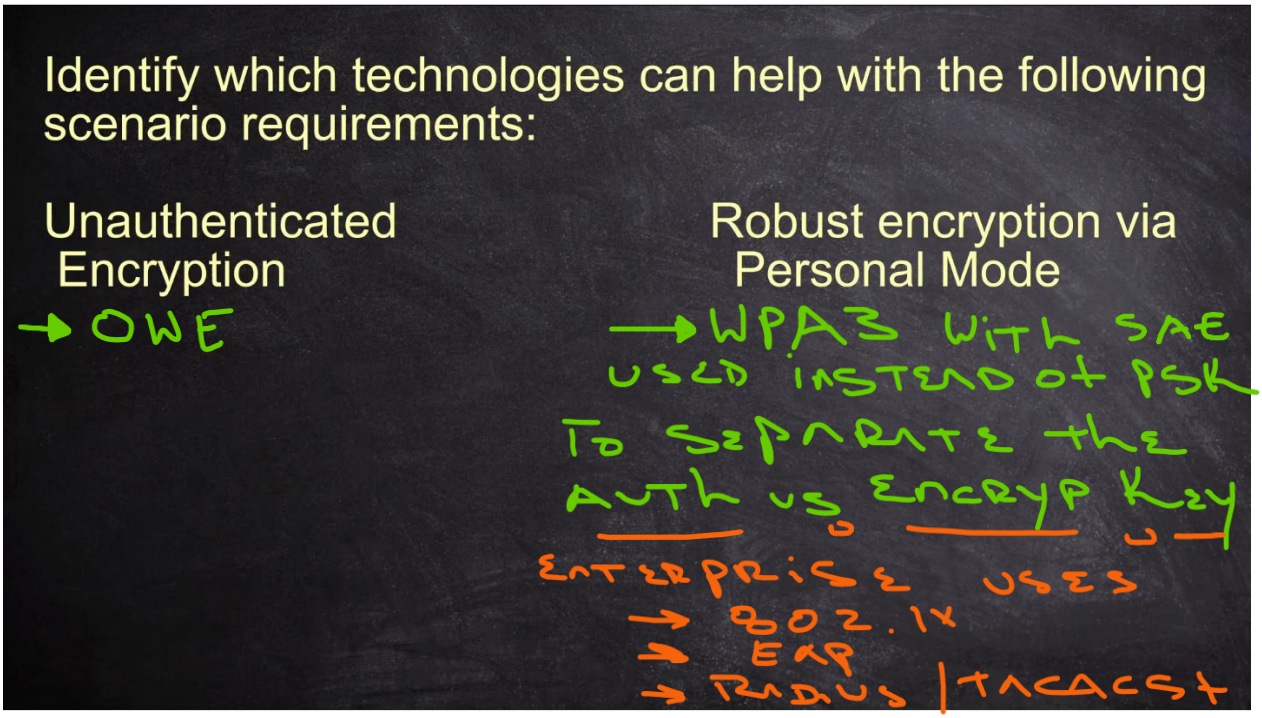
WPA3 personal uses SAE to authenticate but uses the DH KEY exchange process with the Access point (authenticator) to create the encryption key.

Opportunistic Wireless Encryption (OWE)

Remember that in WPA 2 the PSK gives us Authentication and Encryption. In theory we want both process to run separately. When using an open network, no authentication and no encryption is happening. But could we still provide encryption even with no authentication? Yes we can, OWE uses the same concept as SAE of allowing a network with open authentication and still provide encryption.

OWE uses the DH KEY exchange that generates an encryption key.

Validation



48.- Describe Wireless Network Components

Access Points (APs) and Stations (STAs)

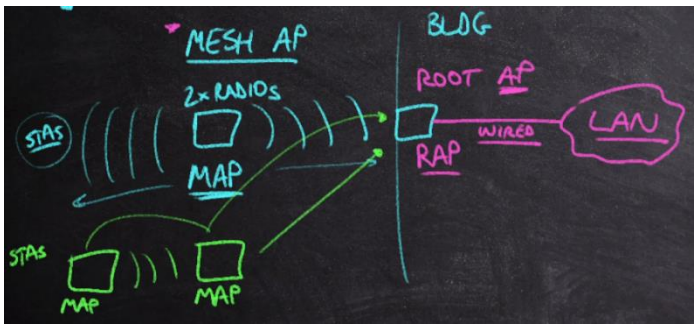
The Access point is designed to connect the wireless devices to a network. We have several capabilities like:

- ➔ Radio Support → 2.4, 5 or 6 Ghz
- ➔ Spatial stream support like
- ➔ Lightweight VS autonomous VS Cloud
- ➔ LAN port: We need to match this speed to the switch it is connected to.

We often refers to APs as STATIONS (sta)

Mesh APs and Bridges

Mesh APs come into place when we want both ends of our connections to our AP to be wireless. So we would need at least 2 radio antennas where one will be connected to the devices and the other one to a switch or L3 switch to connect to the rest of the LAN. We can also have another AP that is connected via ethernet to the LAN, these are called a ROOT AP. The wireless-wireless AP is called MAP. For this deployment, we would need not only multiple radios but a wall poer as well.



A wireless bridge is similar but it is used when you want to connect 2 locations or LANs wireless.

AP Options

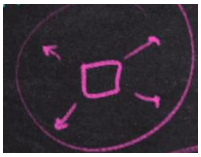
We can differentiate 2 APs groups in:

Indoor → Usually less expensive with internal antennas.

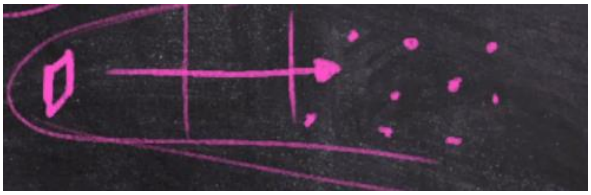
Outdoor → More expensive with external antennas.

Another important aspect to consider is that LAN PORTS are designed with the throughput in mind. Sometimes, it can be designed as MGIG multigig or more to support all the bandwidth. Keep in mind that the switch on the other side needs to support the same bandwidth.

Antenna Effects and Types



Omnidirectional → A wireless transmitter Tx would naturally radiates a spherical shape.



Planar →Patch: Semidirectional

→Panel:

Yagi: Longer range

Dish antenna: Directional antenna. Focus a LOT the beam where it does not spread out. Used to connect to a bridge

Distribution Systems and Media

The job of an AP is to take the traffic coming from a station and sending it to a switch. Stations are actually trying to connect TRHOUGH an AP.

A DISTRIBUTION SYSTEM is in reference to the upstream connection to an access point. In our case, wired or wireless. The DISTRIBUTION SYSTEM is the destination for out traffic.

Wireless LAN Controllers (WLCs)

Since in enterprise we deal with many APs, we need a way to manage them as a group. Most of our modern network are built on lightweight architectures which refers to a device called WLC that centralizes and controls all our APs.

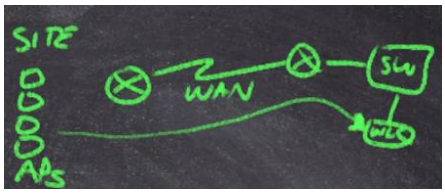
We have the advantage of FEEDBACK LOOP, that ensures proper functionality even when dynamic changes to our architecture or environment are being done. There are several features that we get from a WLC, we can get it as a hardware form but also virtualized, even the CLOUD!

49.- Explain Wireless Network Architectures

Autonomous Architecture

Basically without any WLC, all APs are individually and managed individually.

Lightweight Architecture



Network controllers started to come in. The WLCs started to gain FULL CONTROL of all the APs. It scales out well as far as the amount of APs the WLC is able to support.

In order for the APs to discover the WLCs, there are mechanisms like having a remote site with a bunch of APs that goes through a WAN, switches and find the WLC. This makes us manage our environment in a better way. This gives us

- ➔ Better updates on the firmwares
- ➔ Configurations are centralized
- ➔ Radio Resource Management which is a dynamic tuning of radio signals like the Tx power.

CAPWAP Tunneling

When we look at a lightweight AP connecting to a WLC we see configuration traffic, information, data. How exactly does this data is sent? The answer is TUNNEL.

A tunnel is a point to point connection between 2 points through an existing network. CAPWAP tunneling creates 2 tunnels:

- ➔ Control and Management
- ➔ DATA that the stations send to the AP and therefore to the WLC. The WLC sends it to a switch on the network.
  What this tunnel does is that data frames will not appear on the transmission between the AP and the WLC until it arrives to the switch.

Benefits:

1.- Control and management path

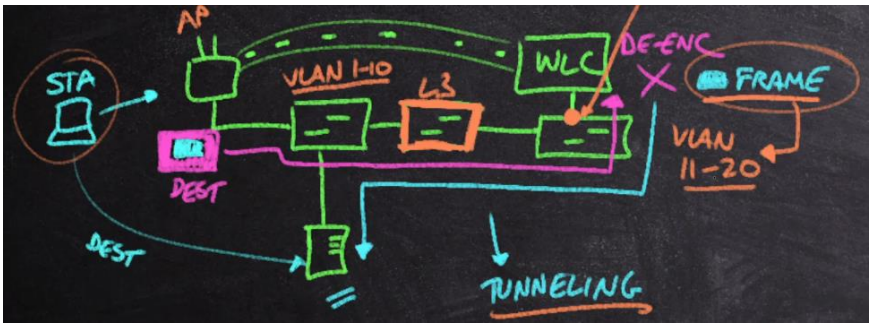
2.- Eliminates L3 ROAMING.

3.- Centralized management of the stations and this includes their VLANs, Subnets.

Tunneling and Data Flow

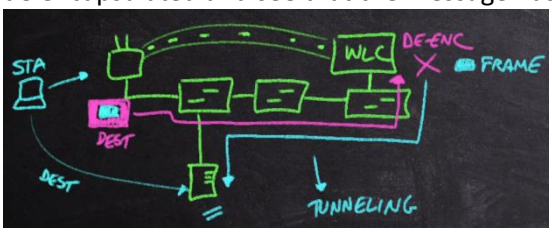
Tunneling is grabbing a message and encapsulating it into another frame. When sent to the other device, the encapsulation frame is stripped to leave the original message. In our case, the WLC will get a bunch of data/information, and it needs to centralize several devices.

There are several important aspects to consider with WLC, tunneling and CAPWAP tunneling, as well as VLANs.



To start with, VLANs become centralized when encapsulated via tunneling traffic. The logical reason being that VLAN traffic only shows up at the last point of interaction between hops (switches) and the WLC as shown in the picture. POINT OF PRESENCE is where a station pop into a network, or when a station is 'seen'.

Since APs are all managed and their traffic go through the WLC, encapsulation goes first to the WLC to be de encapsulated and see that the message has got to go to a destination.



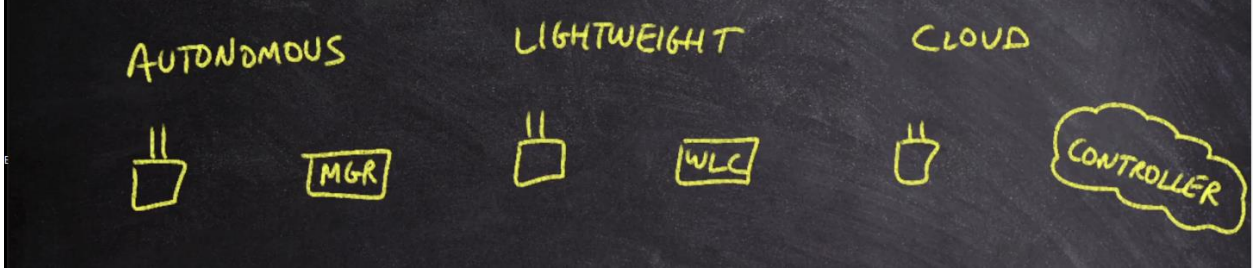
These 2 explanations go into DATA ENCAPSULATION which is optional, but the MANAGEMENT/CONTROL is not optional.

Cloud-Managed Architecture

Autonomous and lightweight are not the only architectures. An increase number of WLC started to become a problem, we no longer have a single point of management. Licensing started to become another big problem as well as management.

The CLOUD is simply somebody else's servers and network. Having the Wireless Controllers in the CLOUD and it brings several advantages like design, licensing, and scalability upwards becomes very simple. Also CAPWAP tunneling is discarded.

Identify the tunnels used in each of the architectures below.



50.- Explain AP Modes of Operation

Local Mode

Local mode acts in the way of forming 2 CAPWAP tunnels. One for the data for each host connected to that AP and the 2nd tunnel for the management. A big aspect of local mode is WLC FAILURE when loss of connection happens, the WLC fails. When this happens, the AP becomes disabled. So the local mode has 2 key properties:

Forms both CONTROL and DATA tunnels to the WLC

Must maintain an active connection at all times.

FlexConnect Mode

Cisco has a mode of connection called Flex Connect. We are still going to maintain a connection to the WLC but only the CAPWAP tunnel of Management.

Ion the case of FAILURE, APs will continue to stay online.

Flex connect is not intended to be deployed everywhere.

Limit to the amount of flex connect modes per APs

Since no data tunnel is deployed, the lack of L3 roaming can become a problem again

Intended to be deployed in an entire site. So each site has to be either local OR flex.

CISCO intends flex connects to REMOTE SITES. It is for when APs and WLC has separation between each other. If we expect regular losses of connectivity to our WLC, then FLEXCONNECT is a better choice

Monitor and Sniffer Modes

Thanks to WLC we can push changes on a daily even hourly basis. In broader terms, WLC ANALYSES the local mode APs' data. In order to improve this data collection process is:

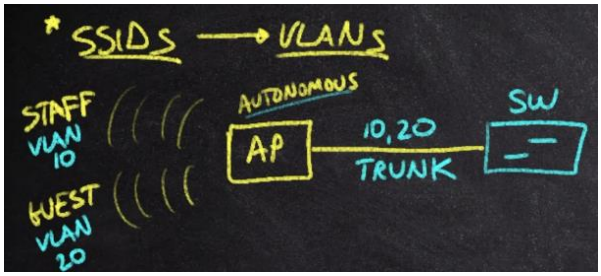
- ➔ Through a monitor AP mode that it's whole purpose would be to send data to the WLC for it to do a better analysis. This data would be RF data of the environment. These are not servicing stations by any means. They look for information through the channels of the SSIDs. They are looking for sources of interference, noise, rogue APs. This is very costly. Helps us troubleshoot L1 problems.
- ➔ Sniffer mode. This mode helps us analyze L2 troubles. This is a full AP deployed that captures packets.

Bridge and Flex+Bridge Modes

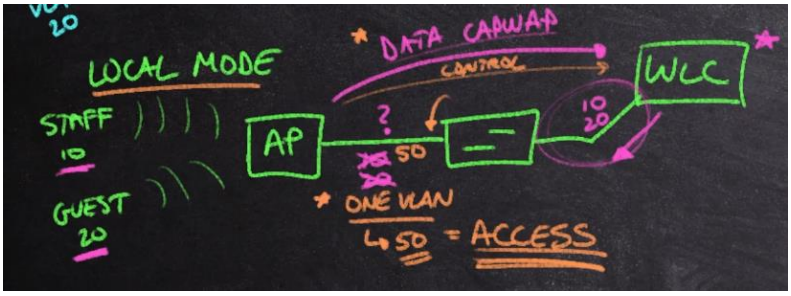
An important WAN situation is when wanting to connect 2 LAN sections wirelessly. Remember that a MESH architecture is when every point is connected to every nearby point. Stations near a remote location that need connection to a LAN, can be seen inside a MESH architecture where every single AP is connected to each other.

Flex + bridge → Combines 2 modes of operation, Flex connect and Mesh operation. APs will maintain a control tunnel to the WLC within all the MESH deployment. In the event of a WLC coming down, all the APs will stay online.

Access and Trunk Links



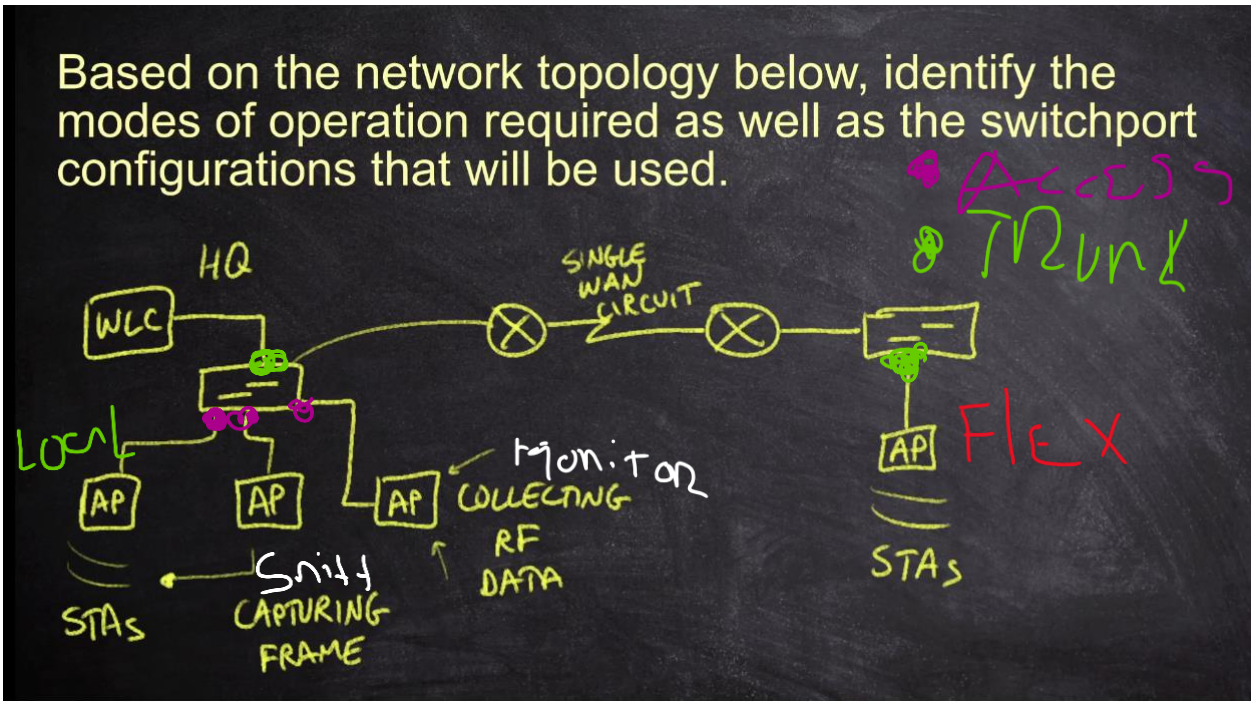
Since SSIDs are linked to VLANs, trunking comes in place when an autonomous AP is connected to a switch and trunking needs to be configured for all VLANs to pass through the connection. [Like the picture]



When looking at a Local Mode deployment, where APs need to connect via CAPWAP tunnels to the WLC, the AP will take all the traffic from the VLANs and send it to the DATA CAPWAP tunnel. But we don't actually need VLAN traffic in all connections since they are traveling through the DATA tunnel, all that matters is that we create ONLY 1 VLAN where the DATA & CTRL/MNGMT tunnel will go through. We don't actually need a trunk no more but a simple ACCESS PORT on the UPSTREAM SWITCH.

Now, since FLEX CONNECT does not use a data tunnel, all the traffic from the stations travel through switching in a normal l3 fashion. This makes trunking necessary again and in some way, we replicate the first example above.

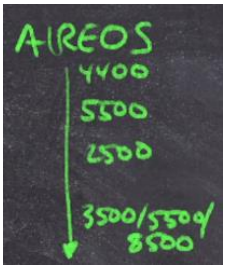
Validation



Local → When running WLC locally, it is better to deploy a local mode of operation and the AP is servicing stations.

Flex → Since it is in a remote location and we want to ensure connectivity, flex connect would be the go to choice.

51.- Explain Cisco WLC Architecture



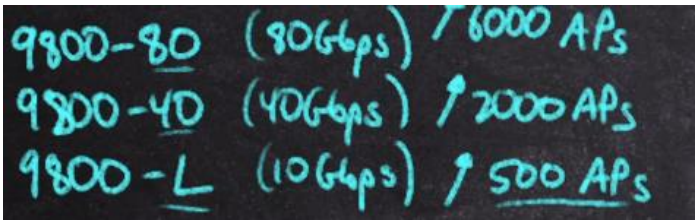
Legacy AireOS Controllers

Cisco acquired a lot of companies like AIRONET for their APs and AIRESPACE where they got their WLCs. AIRESPACE invented LWAPP but it got discarded by the industry and CAPWAP got introduced.

With the company's acquisition came their OS AIREOS which got retired. But for the scope of the CCNA, we'll be looking at this AIREOS that although retired, still is relevant. Nowadays, Cisco have developed IOS – XE that called their devices caralyst. Therefore configuration via CLI will feel different from catalyst vs older WLCs.

Catalyst 9800 Controllers

The current generation of devices is 9100 for APs and 9800 for WLCs. In the picture is the throughput and the amount of APs each 9800 model offers:



The software running here is the 9800 CL compatible with AWS CLOUD.

WLC Discovery

When an AP is connected to a network it should be able to find the WLC automatically and we strive for that.

- ➔ Broadcast in small networks within the same broadcast domain, it is very easy.
- ➔ For APs in different subnets than the WLC, the broadcast will not actually work. Instead, a DHCP check gets done. DHCP provides more than just IP addresses. The DHCP server can provide options like the default gateway.
  - DHCP option 43 is a customizable option that provides APs with IP addresses that belong to the WLC.
- ➔ DNS QUERY is another way to find WLC on a network through local domains and check if it resolves in a particular IP address.
- ➔ Manual backup.

The goal of WLC discovery is to create a list of WLC and then create a priority.

WLC Redundancy

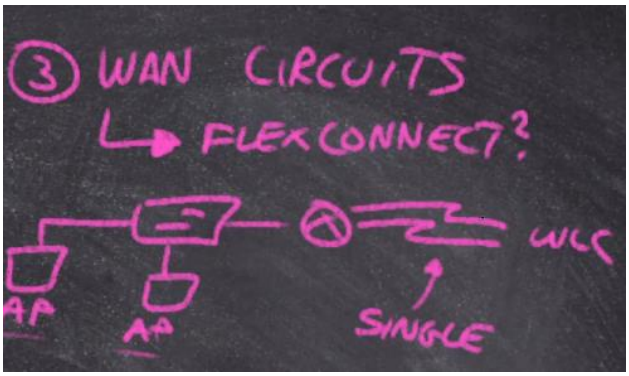
All non flexconnect APs will fail if a WLC fails. So redundancy is important.

- ➔ Actual extra hardware implementation
- ➔ Software VMs running

If the PATH fails:

- ➔ Extra connections established. WAN circuits.
- ➔ Deployment of LINK AGGREGATION (port channel) (EtherChannel) where we create a bundle of wires logically to create 1 logical connection with multiple physical connections. PORT CHANNEL MODE SHOULD BE ON, not any other

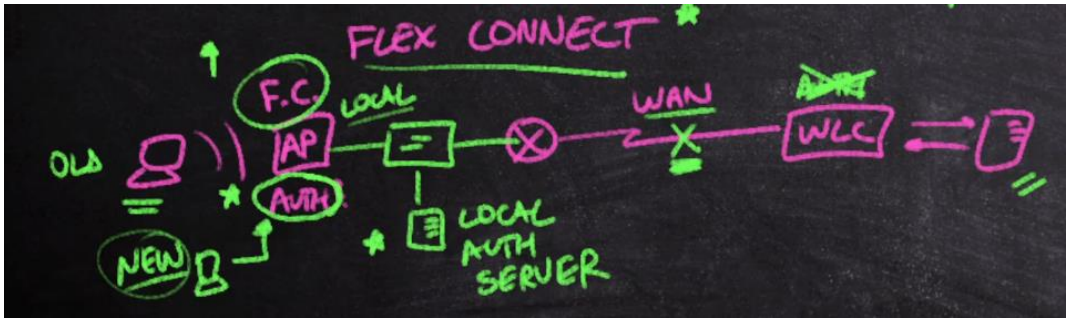
A questions comes up that is should we actually use Flex Connect? Which is great when problems like this show up. Since flex connect only runs 1 tunnel for management, data can flow still on the network since it does not need to travel first to the WLC to be un packed from it's tunnel frame.



WLC Authentication

Remember that WPA enterprise always use 802.1X /EAP for AUTHENTICATION. Remember also that the general process map of authentication is when we have:

SUPPLICANT ------> AUTHENTICATOR ----------> AUTH SERVER



Now that we have introduced WLCs, it will now act as the AUTHENTICATOR and communicates with the AUTH server.

With FLEX CONNECT, we have only 1 tunnel for management to the WLC. A question arrives that what happens if the remote connection to the WLC dies? Is the AP still going to be able to authenticate to the external AUTH server via the WLC for the supplicant? The solution is to have a local AUTH SERVER as a backup. We can still have an external one, but have to make sure that new supplicants still get authenticated into our network.

Lightweight Design Considerations

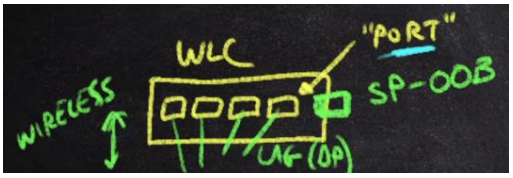
Remember the concept of POP that is the first time on the network that DATA traffic appears from a station after being de encapsuled by the WLC.

52.- Explain Cisco WLC Interfaces for AireOS

Physical Ports

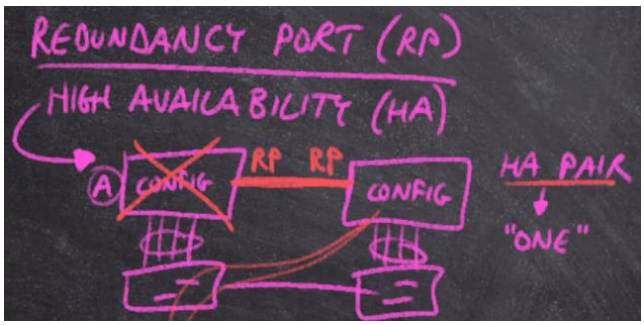
An important difference to consider is that ports are physical and interface virtual.

➔ The physical ports on a WLC are known as DISTRIBUTION PORTS. These connect to a LAN via a switch. Our wireless DATA will flow through these ports meaning that our CAPWAP data tunnel from our APs will be connected here. In here we will see VLANs as well, so this will need to be a trunk port to let VLAN traffic communicate with other LANs. Usually, we will apply EtherChannel to a bundle of links here.



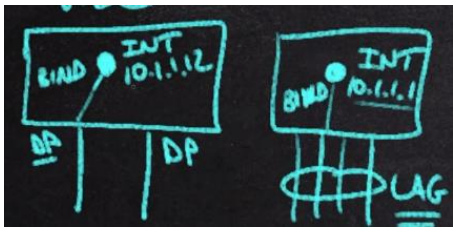
➔ Service ports: This will give us something called OOB Out of band Management access. This means that we will have a dedicated PHYSICAL PORT that does not share traffic with the DATA TRAFFIC and the Mgnmt CAPWAP tunnels. You can not have an IP address assigned to this physical port, therefore it will have to be bounded to a logical interface that we are going to call LOGICAL service-port.

➔ Redundancy port: Used for high availability. It takes redundancy to another level. We basically take 2 WLC that act as 1 on the network. We do this so if one of the WLC goes down, the other one will take place via the Redundancy port.



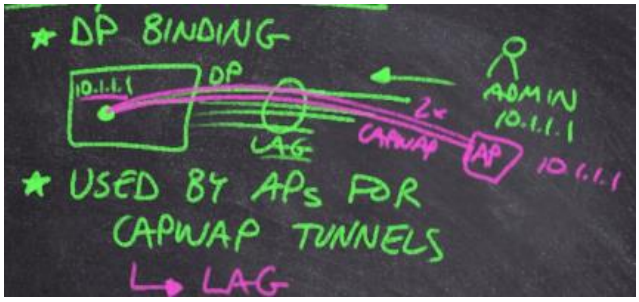
Logical Interfaces

Remember that on switches we can create virtual interfaces known also as VLAN interfaces or SVYs.



So each logical interface has it's own IP address and through distribution ports that remember receive the data traffic form the CAPWAP tunnel. Even it works if we create a bundle of connections [etherchannel] which is highly recommended. There 3 logical interfaces:

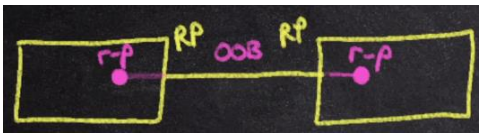
➔ Management: Allows us to manage the WLC. Obviously it will go through a distribution port just like the physical but in this case, 'responds' to the logical interface. Keep in mind that virtual interfaces are 'attached' or assigned to DISTRIBUTION PORT [similar to how a VTY is assigned to a VLAN]. ONLY WHEN THERE IS LAG INVOLVED, APs WILL USE THE MANAGEMENT INTERFACE TO CREATE OUR 2 CAPWAP TUNNELS.



➔ Virtual interface: This will be have an unreachable IP ADDRESS. It would be 192.0.2.1, this is because it is stablished for lab purposes meaning that the entire subnet is not reachable by the internet. It is used for internal functionality.

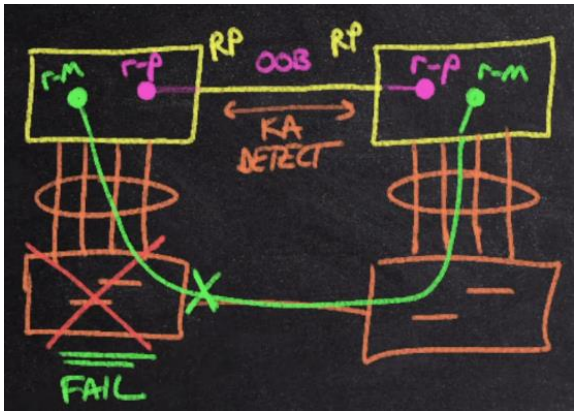
➔ LOGICAL Service-port: Since you can NOT flag an IP address to a service port from what we saw above, this will need to go to a SERVICE-PORT INTERFACE that bounds to our service port for OOB management access.

Redundancy Interfaces



Stateful Switch-over SSO is when we want to take 2 WLC and create 1 logical WLC. Imagine that we have 2 redundancy ports paired with 2 logical redundancy-ports. THIS IS AN OOB meaning that we are not sharing wireless traffic with our CAPWAP tunnels whatsoever as we have explained earlier.

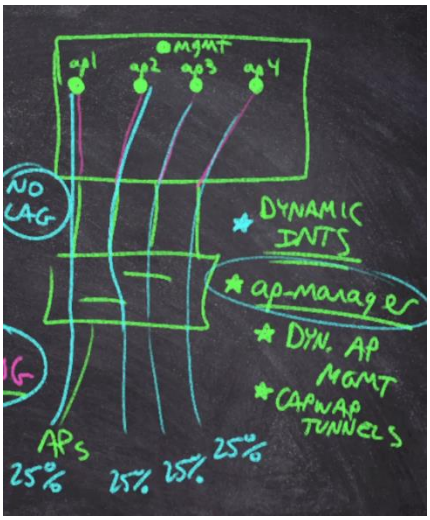
Now, we are going to create a second logical interface called REDUNDANCY-MANAGEMENT. Remember that in order for it to be redundancy around, devices need to communicate whether or not they are still online. BUT, what if another device that has no communication with the logical redundancy ports goes down? WLC bounded by the redundancy ports still need to know this, so the REDUNDANCY MANAGEMENT INTERFACE will be in bound with the rest of the network via the LAG to know what is going on with the rest of the network.



Dynamic Interfaces

How do we map the SSID to a specific VLAN? One of the ways is by leveraging a DYNAMIC INTERFACE. This interface is going to be created by the admin and we can create as many as we need and it will have a lot of parallels to the SVIs. We are mapping the INTERFACES to a VLAN so the idea here is to create a DYNAMIC INTERFACE attached to a VLAN and therefore will have access to the interface.

AP-Manager Interfaces

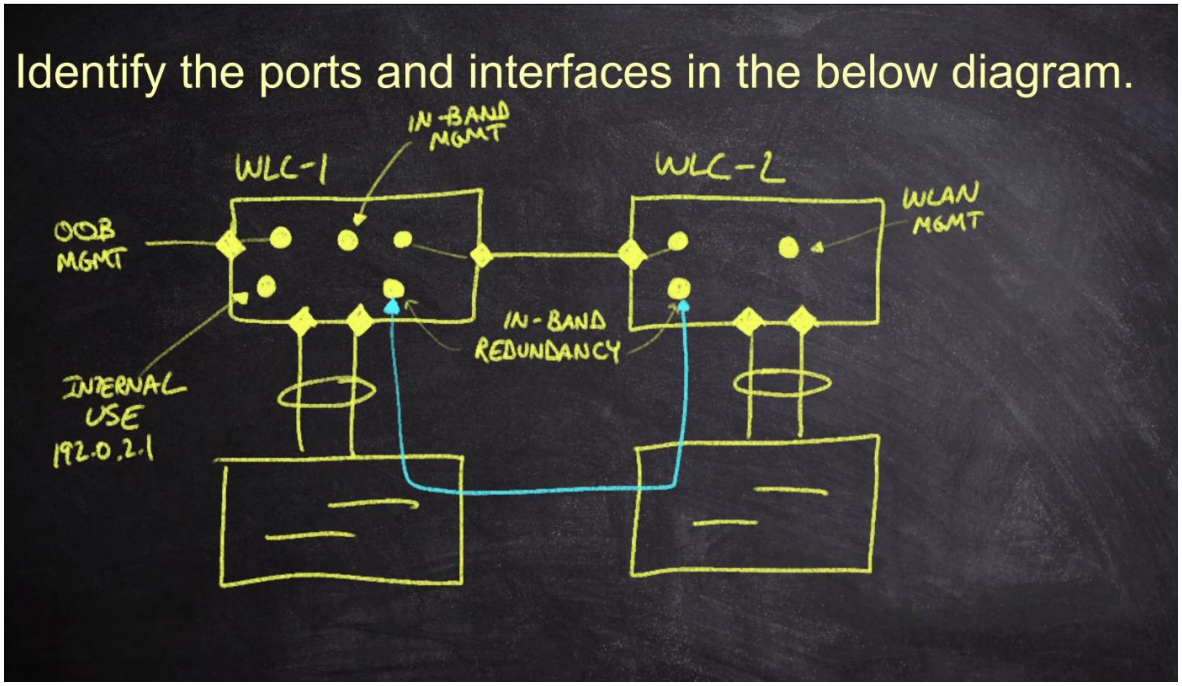


Remember that when LAG is being used, both tunnels will be terminated to the management interface. When LAG is not being used, the management interface will be bound to a single physical port meaning that all traffic will be coming in to as single port.

Ideally we want to use LAG for redundancy and better flow of traffic. But there is a 'trick' when no lag is being used so traffic does not need to be bounded to a single physical port.

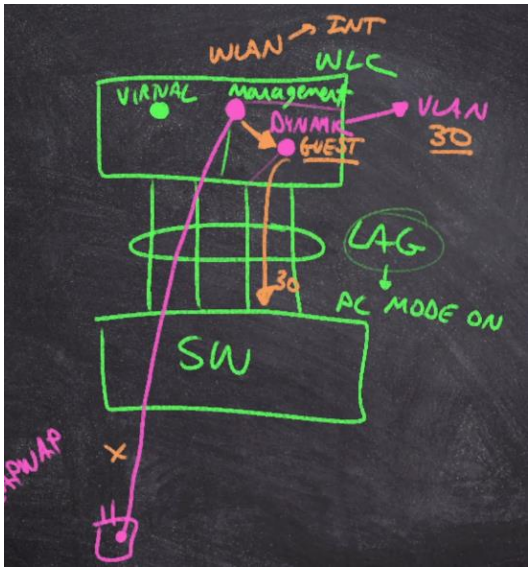
The conversation here is about LOAD-BALANCING so we can leverage dynamic interfaces studied before with this concept and we are going to create a dynamic interface called AP-MANAGER that supports CAPWAP tunnels on this interface. This will make that our APs connect to different dynamic interfaces. Remember that this is only used when no lag is used.

Validation



53.- Explain Cisco WLC Configurations

VLANs and Dynamic Interfaces



Remember that we need to configure dynamic interfaces that are mapped to VLANs. Dynamic interfaces need to be created and they are sort of like SVIs. In the picture we have virtual interfaces, a management interface that connects directly to the only port available LAG. But we will need to configure the DYNAMIC INT which are mapped to a VLAN. The CAPWAP connection has to be made from the AP to one of these interfaces that is the management one. The traffic then is transferred to one of the DYNAMIC INT which is the actual interface linked to the VLAN to be handed over to the network via the trunk ports on the WLC.

What really matters is that when we have a WLAN ---> We especify an Interface ---> That puts it out on a VLAN.

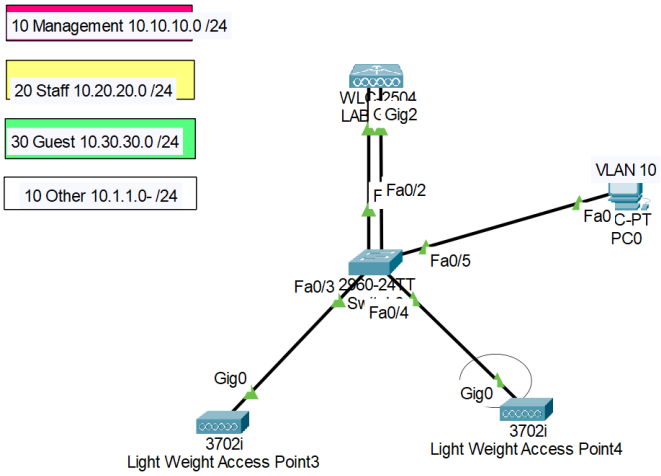
AP Groups

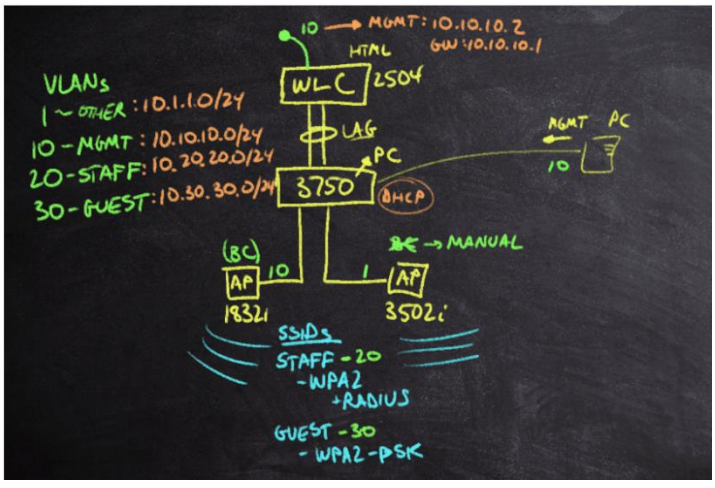
Imagine that we have multiple WLANs with multiple APs, some of them will share VLANs between them. We can create AP groups used on AIREOS controllers. APs will by default go into the default group.

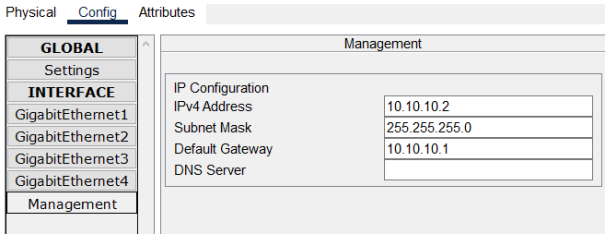
54.- Perform Cisco WLC Initial Configuration

Labbing with WLCs

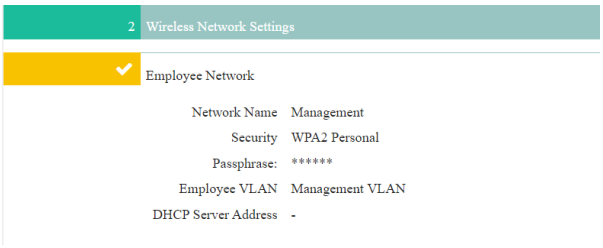
We are going to work with this lab at the moment.



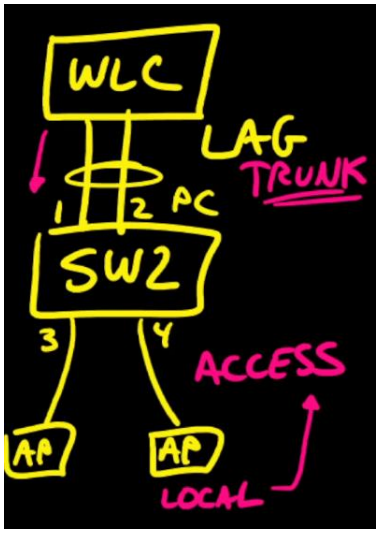




First thing is to configure our Mgnmt interface on our WLC which traffic will flow through VLAN 10. It's IP address will be 10.10.10.2 with it's gateway being 10.10.10.1. We were able to log in via the web browser. Next, we can set the VIRTUAL IP address that is not routable.



Switch Configuration



Remember that our APs will be running on LOCAL mode meaning they will be using CAPWAP tunnels to communicate with our WLC. So our fa0/3-4 on our switch will be SWITCHPORTS and the fa0/1-2 will be TRUNKS. Remember that there are 4 main ways that an AP can recognize a WLC on the network:

- ➔ Broadcast frames
- ➔ DHCP option 43
- ➔ DNS
- ➔ Manual configuration

If an AP is in the SAME VLAN as the WLC like our first AP is, then it will use a BROADCAST FRAME to acknowledge the presence of the WLC. However, since the other AP is in a different VLAN 1, we will have to manually configure it.

55.- Configure Cisco WLC WLANs

WLAN Configuration

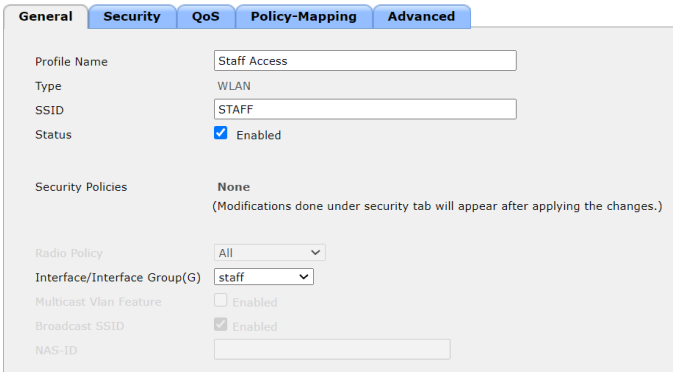
Keep in mind that the goal is that we want our APs to broadcast SSIDs for computers to join in. Therefore we need to configure WLANs that will encompass:

- ➔ SSID
- ➔ Parameters
- ➔ Security (WPA ENTERPRISE/PKA)
- ➔ Interface (VLAN to the SSID)

The WLAN will be applied to an AP group which are required. A default AP will see all APs and all WLANs.

For our lab we will create the STAFF & GUEST WLAN.

After we have created both our virtual interfaces for each WLAN Staff & Guest, we will link these and enable each WLAN



WLAN ID	Type	Profile Name	WLAN SSID	Admin Status	Security Policies
1	WLAN	Management	Management	Enabled	[WPA2][Auth(PSK)]
2	WLAN	Staff Access	STAFF	Enabled	None
3	WLAN	Guest Access	GUEST	Enabled	None

Wireless Security Review

Here the main thing to remember is that you have standards WPA → WPA → WPA 3 and these encompass encryption and authentication:

- ➔ Encryption: you have TKIP and AES (way better)
- ➔ Authentication: You have PSK and ENTERPRISE with deployments of 802.1X/EAP with RADIUS or TACACS+

Troubleshoot Cisco Wireless Networks

If the frequency of a wireless waveform increases, what happens to its wavelength?

A. Increases
B. Remains the same
C. Decreases ✓✓

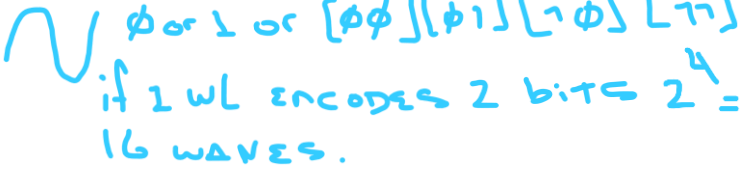
What best describes a wireless phase?

A. Cycles per second = f
B. Physical length of a wave
C. Distance from origin
D. Relative shift in position ✓✓

How does a wireless station know when a collision has occurred? CSMA/CA

A. An ACK frame never appears ✓✓
B. The collision is detected
C. The receiver reports the collision
D. The dB reading is too low

A wireless transmitter wants to encode 4 bits onto each waveform in a transmission. How many different waveforms will be needed for this?



A wireless transmission is made with 100mW of power. The transmission is received at 25mW. What decibel loss occurred?

100 mW ⟶ 25 mW 75% loss
100 ÷ 2 = 50 ÷ 2 = 25
-3 dB -3 dB = -6 dB

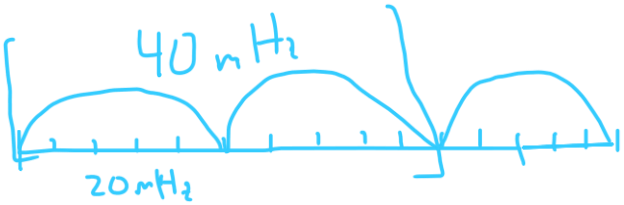
if dB ⟶ mW
±3 ⟶ 2x
±10 ⟶ 10x

A wireless transmission is made at 14dBm. How many mW is this transmission?

ϕ dBm = 1 mW
+ 10 dBm → *10
+ 10 dBm → *10
20 dBm → 100 mW
- 3 dBm → ÷ 2 mW
- 3 dBm → ÷ 2 mW
14 dBm 25 mW

What is a reasonable real-world expectation for a received signal (RSSI)?

A. -100dBm
B. -70dBm ✓✓
C. -20dBm
D. 0dBm
E. 20dBm



How many 40MHz bonded channels can be deployed in the 2.4GHz band?

A. Zero
B. One ✓✓
C. Two
D. ThreeT

Which band(s) can be used by each of the following Wi-Fi specs?

	2.4GHz	5GHz
A. 802.11a	✓	✓
B. 802.11b/g	✓	✓
C. 802.11n	✓	✓
D. 802.11ac	✓	✓
E. 802.11ax	✓	✓

What term is used to describe the level of background radiation in an environment?

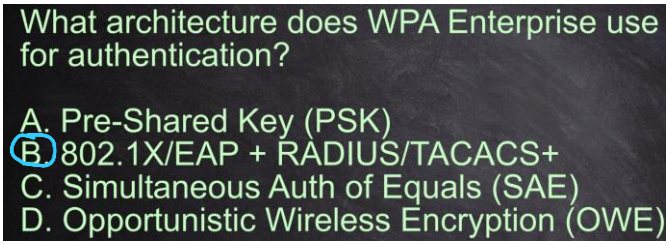
A. Signal
B. Noise -90
C. Interference
D. SNR
E. RSSI

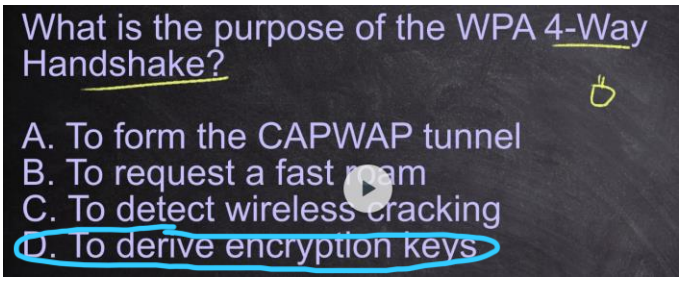
A wireless signal is received at -50dBm against a background noise level of -100dBm. What is the SNR?

A. +50dB
B. -50dB
C. 2dB
D. 1/2 dB

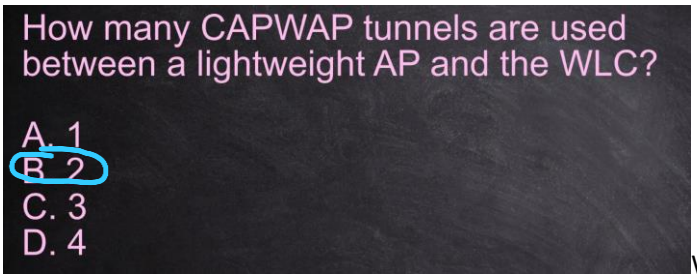
Signal / noise

dB → -50dBm - (-100dBm)
→ +50dB

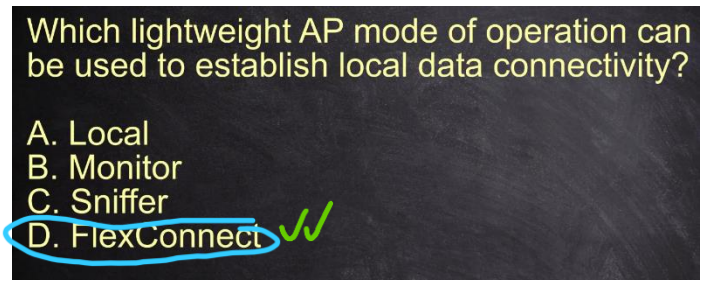




Autonomous
Does not need
a WLC



We have to ask ourselves if the AP is in local mode of FLEX CONNECT mode. Local mode has BOTH DATA & CONTROL tunnels.



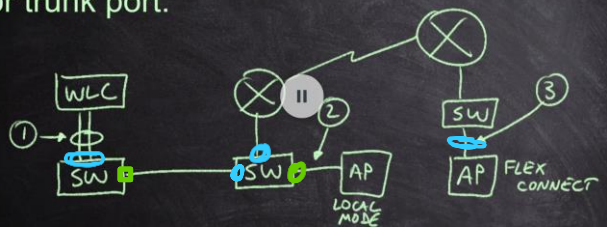
Local mode

Sends the traffic to the WLC via 2 CAPWAP tunnels.

flexConnect

Connects directly to a switch and sends traffic to the switch without going to the WLC.

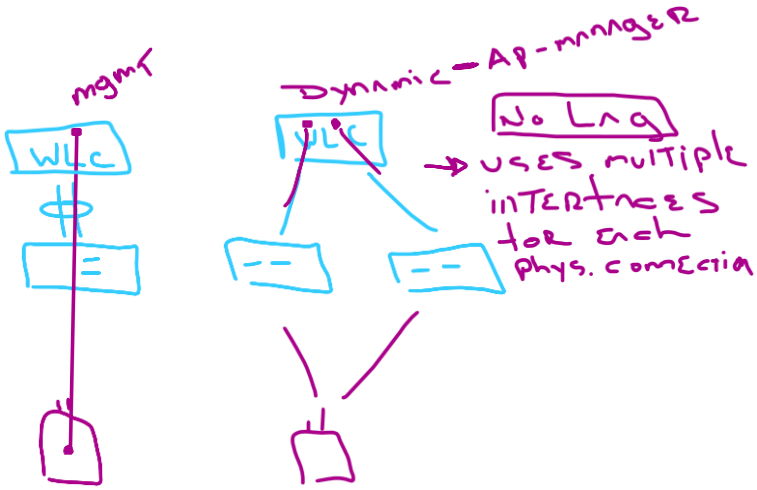
Given the diagram below, identify whether each link should be configured as an access or trunk port.



- Access
- Trunk

What are two considerations when forming a LAG on an AireOS WLC?

A. LACP is not allowed
B. Access mode must be configured
C. All ports must be in the LAG
D. FlexConnect is not supported
E. AP-Manager interfaces must be used



Which interface is used for CAPWAP termination when a LAG is in use?

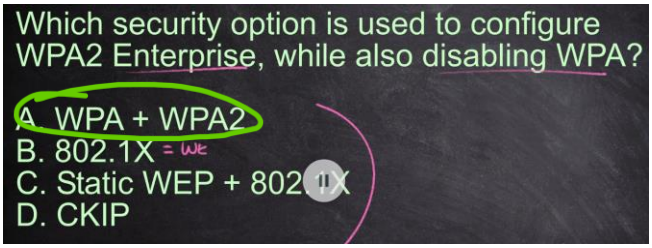
A. ap-manager
B. management
C. redundancy-management
D. dynamic

What feature can be used to restrict which SSIDs are used by which APs?

A. WLANs
B. VLANs
C. AP Groups
D. RADIUS/TACACS+

Which QoS level is considered the highest and is often used for VOIP?

A. Platinum
B. Gold
C. Silver
D. Bronze


Which security option is used to configure WPA2 Enterprise, while also disabling WPA?
A. WPA + WPA2
B. 802.1X
C. Static WEP + 802.1X
D. CKIP

56.- Configure IPv4 Static Routes

Static Route Overview

IP static routing works in the way that Routers are configured by administrators to route traffic on a specific path. The key here is that we have to configure the route.

On static routes you have 3 types:

- ➔ Static NETWORK routes: When a packet goes to a SUB NETWORK.
- ➔ Static HOST routes: When a packet goes to a specific known host.
- ➔ DEFAULT route: When the router does not know where to send it, simply send it here.

The way to configure a static NETWORK route is the following:

```
R1(config)#ip route 10.23.0.0 255.255.255.0 10.12.0.2
```

The way to configure a static /32 bit HOST route is the following:

Same process as the NETWORK route, just that we use a 32 bit mask.

```
R1(config)#ip route 10.23.0.3 255.255.255.255 10.13.0.3
```

The way to configure a static NETWORK route is the following:

```
R4(config)#ip route 0.0.0.0 0.0.0.0 10.34.0.3
```

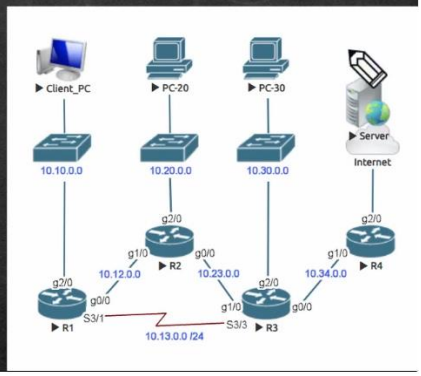
Validation

Static Route Lab

- R1: static network route to 10.23.0.0 /24 using R2 as next hop
  - ip route 10.23.0.0 255.255.255.0 10.12.0.2
  - ping 10.23.0.2
  - traceroute 10.23.0.2

- R1: static host route to 10.23.0.3 /32 using R3 as next hop
  - ip route 10.23.0.3 255.255.255.255 10.13.0.3
  - ping 10.23.0.3
  - traceroute 10.23.0.3

- R4: default route, using R3 as next hop
  - ip route 0.0.0.0 0.0.0.0 10.34.0.3
  - ping 10.23.0.3
  - ping 10.13.0.3



57.- Describe OSPF

Routing protocol that helps automize and share paths for routers. There are different flavors of OSPF:

1.- RIP & RIPv2 → It is still functional although old.

2.- EIGRP → Created by Cisco better than RIP.

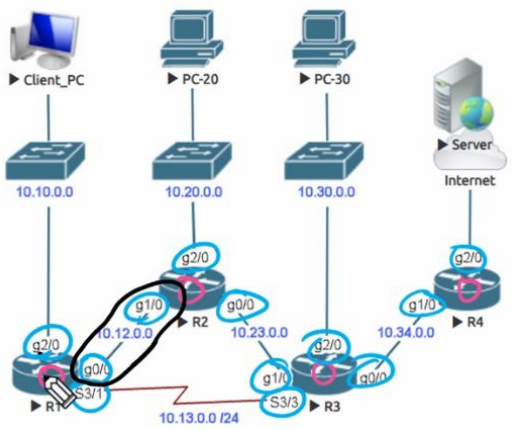
3.- OSPF → Supports IPv4 and IPv6.

On the Internet, BGP is used. Between ISP can supports a huge number of routes, making it a great choice for immense WANs seen on the internet.

Overview of OSPF

We have to enable it first and then specify which interface. Let's imagine OSPF working in areas, just like in the picture. Each area shares all of the important information, paths – interfaces – subnets. So imagine a large broadcast domain being shopped into smaller more manageable portions.

In our topology, let's separate the areas as follows:



To share inforamtion with each other, 2 routers would have to be what we call:

FULLY ADJACENT NEIGHBORS. In our topology, g0/0 R1& g1/0 R2. The requirements are the following:

- ➔ Unique Router ID called OSPF ID.
- ➔ Agree. 2 OSPF routers need to agree on **STTAMP (Subnet. Timers. Network Type (broadcast or PtP). Area of OSPF. MTU maximum transmission unit. Passwords for authentication)**

Designated Router

On each subnet segment, there will be a designated router and a BACKUP router. OSPF designers needed this because considering the worst case scenario of having multiple adjacent router's interfaces sharing information all the time, creating multiple adjacent routers and therefore a lot of redundancy in information.

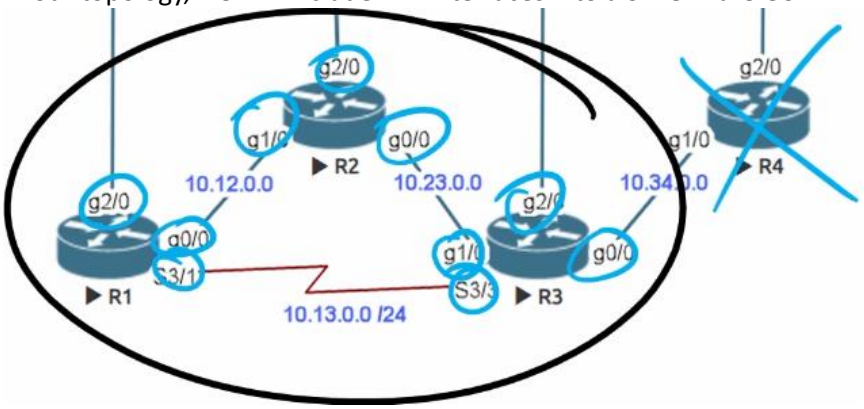
By electing a DR & BR for a specific network segment, all other routers can create 1 SINGLE ADJACENCY with the DR. The Back up router is there in case the DR goes off. In which case, the single adjacency would go the the BR.

Enable OSPF and Include All Interfaces

OSPF is a pogram basically, so it needs to start running on a router. We need to pay attention to:

- ➔ Process ID
- ➔ Which interfaces to include

In our topology, we will include ALL interfaces into a SINGLE are OSPF.



In order to include ALL interfaces, we have to use the following command:

```
Router#CONFIG
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf ?
  <1-65535>  Process ID
Router(config)#router ospf 1
Router(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

.We'll do this on every router

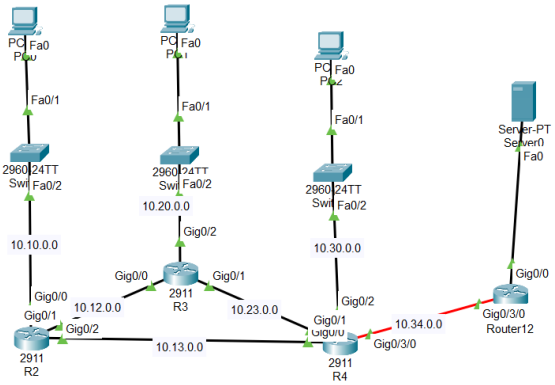
We can verify all interfaces running OSPF with this command

```
R1#show ip ospf interface brief
```

And the neighbors:

```
R1#show ip ospf neighbor
```

OSPF Router ID Selection Process



If we run the command ***show ip OSPF neighbor*** we will be able to see all the neighbor's IDs [ip addresses] related to our router

```
R2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.23.0.3         1   FULL/DR         00:00:31    10.12.0.3       GigabitEthernet0/1
10.34.0.4         1   FULL/DR         00:00:31    10.13.0.4       GigabitEthernet0/2
```

If we run show ip ospf we will see the ROUTER ID for the router we are in. These IDs are unique.

```
R2#show ip ospf
 Routing Process "ospf 1" with ID 10.13.0.2
```

```
R3#show ip ospf
 Routing Process "ospf 1" with ID 10.23.0.3
```

```
R4#show ip ospf
 Routing Process "ospf 1" with ID 10.34.0.4
```

But WHY did they choose these IDs????? → The reason they are unique is because at the moment OSPF starts running, The following happens:

➔ When no loopback interface has been configured on a router and NO ROUTER ID EITHER, the HIGHEST NUMERICAL INTERFACE will be chosen on other interfaces besides the loopback. This was our case in our topology.

Let's look at the ip interfaces on R1 as an example:

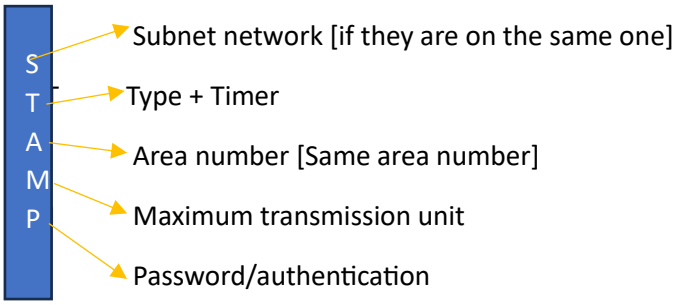
Highest numerical value interface

```
R2#show ip int brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     10.10.0.2       YES manual up                    up
GigabitEthernet0/1     10.12.0.2       YES manual up                    up
GigabitEthernet0/2     10.13.0.2       YES manual up                    up
GigabitEthernet0/1/0   unassigned      YES unset  administratively down down
GigabitEthernet0/2/0   unassigned      YES unset  administratively down down
GigabitEthernet0/3/0   unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
```

Of course, if there is a loopback address already configured, then it will choose that one.

Confirming OSPF Network Types and Timers

Remember that in order to have FULL ADJACENCY, routers have to agree on:



In regard to the TYPE → There are 2 types of connection

1.- Broadcast → Default for ethernet

2.- Point to Point → Default for serial

```
R2#show ip ospf interface g0/1

GigabitEthernet0/1 is up, line protocol is up
  Internet address is 10.12.0.2/24, Area 0
  Process ID 1, Router ID 10.13.0.2, Network Type BROADCAST, Cost: 1
```

Identify the DR and BDR for a Network Segment

FOR ETHERNET SUBNETS:

Designated routers and Backup Designated routers are PER SUBNET. The rest of the routers involved in a subnet are going to be called DR-OTHER.

So who wins? The interface with the HIGHEST PRIORITY wins. If they have the same priority, then the HIGHEST ROUTER ID will be chosen

Once the DR is assigned, the other with the highest priority numbers, will get the BDR role.

NOTE: once the DR & BDR have been established, and an extra router is added, the priorities won't change.

Validation

Describe OSPF Lab

Using show commands, answer the following questions about the topology:

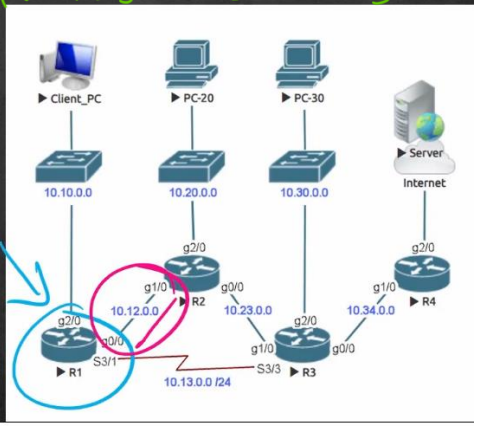
What is the OSPF Router ID for R1? 1.1.1.1
Who is the DR on the 10.12.0.0/24 network? R2
Why is the R2 OSPF router-id 2.2.2.2? A loopback ADD was configured

On R2 Gig 1/0, answer the following:

- What is the hello timer value? 10
- What area is this interface in? Area 0
- What neighbors does R2 have? 1.1.1.1, 3.3.3.3
- How did this interface get included in OSPF?

On R3, answer the following:

- How many interfaces are enabled for OSPF? 5
- Where is R3 acting as a BDR? 2 G1/0 → G0/0



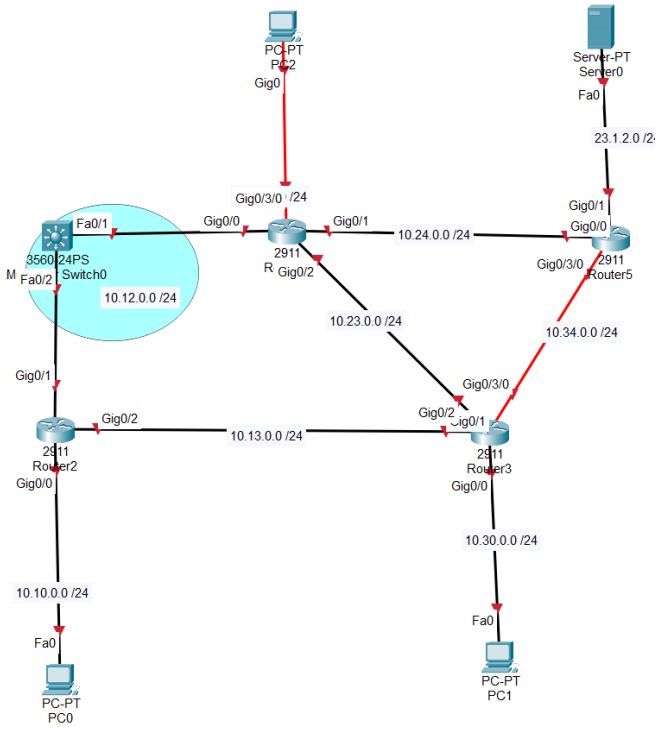
58.- Configure OSPF

Router ID, Interfaces, and DR

A recap: In order for a router to choose it's router ID

- ➔ If it is already configured
- ➔ Loopback address configured
- ➔ Highest numerical value of an interface

Let's learn how to manually configure it.



In our topology, we will be using AREA 0 ospf.

Since all of our routers' interfaces will be in a single are we can do it in 2 ways:

1.- Individually by every interface

2.- Network statement on global config mode.

GAMEPLAN RIDs:

R1 -> 1.1.1.1

R2 -> 2.2.2.2

R3 -> 3.3.3.3

R4 -> 4.4.4.4

We repeat this for all routers.

```
R1(config)#router ospf 1
R1(config-router)#rou
R1(config-router)#router-id 1.1.1.1
```

Adding OSPF Interfaces with Interface Commands

Now let's include the interfaces into our AREA 0 OSPF.

CAREFUL WITH PROCESS ID

```
R1(config-if-range)#do show ip ospf
 Routing Process "ospf 1" with ID 1.1.1.1
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0

R1(config-if-range)#ip ospf 1 area 0
```

```
R2(config)#int range g0/0-2, g0/3/0
R2(config-if-range)#do show ip ospf brief
show ip ospf brief
              ^
% Invalid input detected at '^' marker.

R2(config-if-range)#do show ip ospf
 Routing Process "ospf 1" with ID 2.2.2.2
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 0. 0 normal 0 stub 0 nssa
 External flood list length 0

R2(config-if-range)#ip ospf 1 area 0
```

```
R3(config)#int range g0/0-2, g0/3/0
R3(config-if-range)#ip ospf 1 area 0
```

```
R4(config)#int range g0/0-1, g0/3/0
R4(config-if-range)#ip ospf 1 area 0
R4(config-if-range)#no shut
```

Verifying OSPF Enabled Interfaces and IP Routing

So now, every router should know in their routing tables how to reach different subnets in the same area 0

OSPF Network Statement With All Wildcard Bits On

There is a faster way to include interfaces without going one by one.

The formula is on global config mode:

Network [sub_net] [mask] area [#]

For example: if we have network 10.12.0.1 0.0.0.0 → this means that for all interfaces on the subnet 10.12.0.1 on every single octet we will include these in area 0

If we have 10.0.0.0 0.255.255.255 → This means that we care about the interfaces that start with the octet 10 and the rest does not matter.

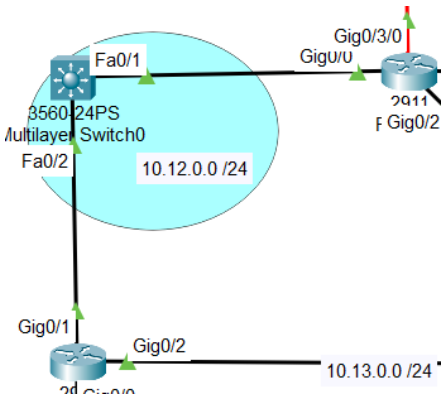
Adding Additional Routers and Interfaces to OSPF Area 0

We can go even one step beyond and look at the entirety of our network segments starting with a 10 octet. So the only octets that change [depending on the subnet] are the last 3. We can actually simply write on a note the commands and do it that way.

```
conf t
router ospf 1
network 10.0.0.0 0.255.255.255 area 0
int loop 0
ip ospf 1 area 0
end
```

Interface Priority Impacts the DR Elections

We can influence the decision on the DR in a subnet. There is a new priority number introduced to us that is the number we can change to manipulate the DR result. As an example, for our 10.12.0.0 subnet, by default the priority number is 1, so next they compare router IDs being R1 = 1.1.1.1 & R2 = 2.2.2.2.



59.-v Understand OSPF's Cost & Default Route

OSPF Default Costs and Default Route Overview

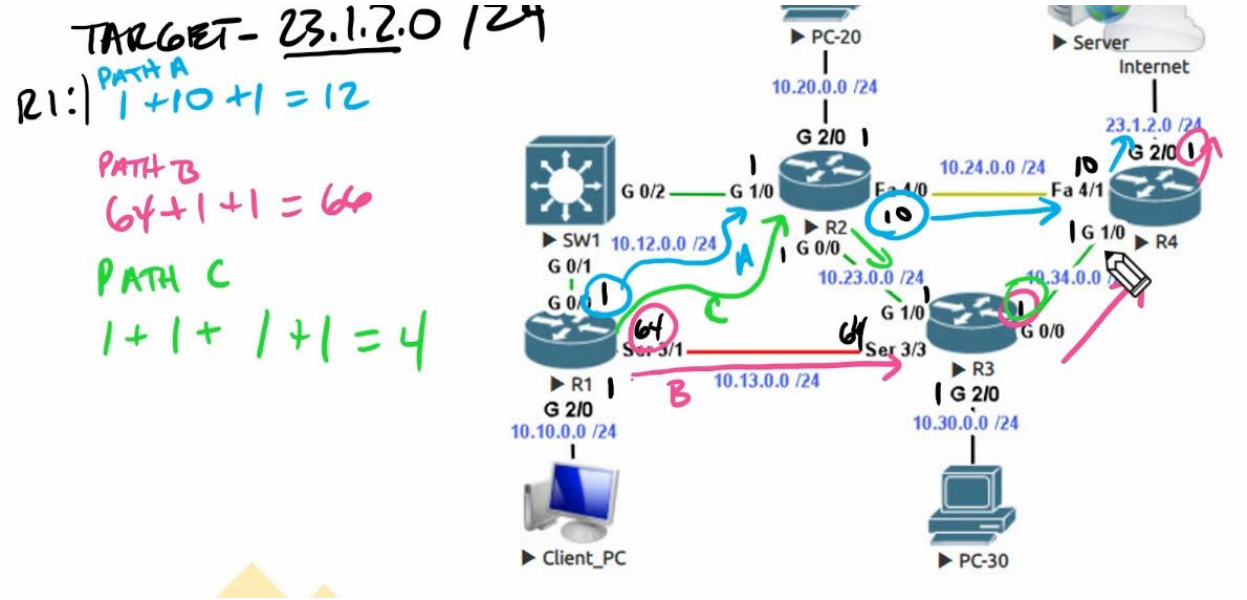
OSPF is referred to as LINK STATE ROUTING protocol. Link state routing protocols share frames called LSAs where information about the interface's states are shared constantly. These LSAs are shared within an area.

Imagine a router sharing → "I am connected to [subnet] on this [interface]"

This way, all routers can start building their Ip tables to know where to forward traffic.

Another important information shared is cost. In case of OSPF the lowest cost wins:

Each router has costs on their INTERFACES. Routers start calculating paths costs and will choose the lowest cost to reach a destination according to It's routing table.



This is great when working inside our LAN. When it comes to sending routable traffic over the internet, we use a DEFAULT ROUTE towards our router that is the last hop to the internet. We can also simply configuire 1 default route on our LAST ROUTER and this one will advertise it to the rest of our OSPF AREA 0.

Interface and Path Cost

One good way to see our costs is the following:

```
R1#show ip route ospf |
```

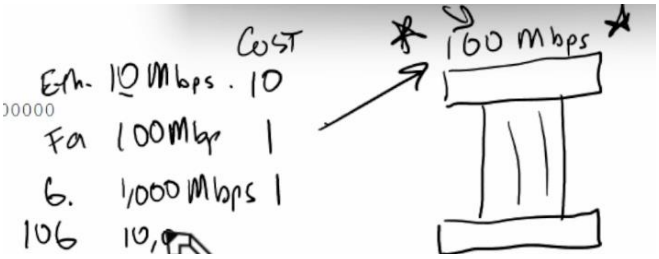
```
     10.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
O       10.20.0.0 [110/2] via 10.12.0.2, 00:10:20, GigabitEthernet0/1
O       10.23.0.0 [110/2] via 10.12.0.2, 00:10:20, GigabitEthernet0/1
                  [110/2] via 10.13.0.3, 00:10:20, GigabitEthernet0/2
O       10.24.0.0 [110/2] via 10.12.0.2, 00:09:25, GigabitEthernet0/1
O       10.30.0.0 [110/2] via 10.13.0.3, 00:29:31, GigabitEthernet0/2
O       10.34.0.0 [110/2] via 10.13.0.3, 00:29:31, GigabitEthernet0/2
```

COST

NOTE: Gig ethernet although faster than Fa, will not always mean it will choose that path. You have to see it from the router in which the frame is currently at.

Auto Cost Reference Bandwidth

There is a problem. By default a router might see an interface being Fa or Gigabit with a cost of 1 even though Gigabit is way faster. OSPF was invented decades ago, and we didn't count that someday we would have speed of 10 G or more. OSPF calculates the referred bandwidth speed by dividing the speed of an interface by 100 Mbps which is consider the highest.



100 / 10 = COST 10

100 / 100 = COST 1

100 / 1000 = rounded COST 1

100 / 10,000 = rounded COST 1

SO BY CHANGING THE REFERENCE SPEED OF 100 Mbps, WE WILL BE ABLE CHANGE THE COST.

For our topology, we will use 10,000 Mbps as the standard.

```
R2(config)#router ospf 1

R2(config-router)#auto-cost reference-bandwidth 10000
% OSPF: Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent across all routers.
```

MAKE SURE TO DO IT ON EVERY AREA 0 ROUTERS

Changing Interface Cost

We can actually manually change the cost of an Interface.

In the interface:

```
R2(config-if)#ip ospf cost 11
```

We can now see that has 1 more to it's cost

```
R2#show ip ospf int brief
Interface     PID   Area                IP Address/Mask          Cost  State  Nbrs F/C
Gig0/3/0        1   0                   10.20.0.2/255.255.255.0  10      DR   0/0
Gig0/2          1   0                   10.23.0.2/255.255.255.0  11      DR   0/0
Gig0/0          1   0                   10.12.0.2/255.255.255.0  10     BDR   0/0
Gig0/1          1   0                   10.24.0.2/255.255.255.0  10      DR   0/0
```

Default Routes in OSPF

There is a big problem!

If we do not have a default route, a router can reach and send traffic to the OSPF area that other router's are in, but if there is a destination address that is not in, it will drop the packet.

We can manually configure a default route. The way to do this is by setting a default router like in previous chapters:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 23.1.2.1
```

→ Notice that we would need to include this gateway into our OSPF area.

In our topology, we can set R4 to share it's own default ip route to an external network to the rest of the intermetal network 10.0.0.0.

```
R4(config-router)#default-information originate
```

Understand Administrative Distance

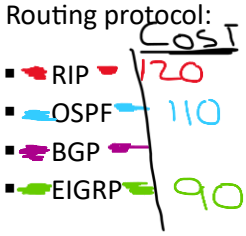
Administrative Distance Overview

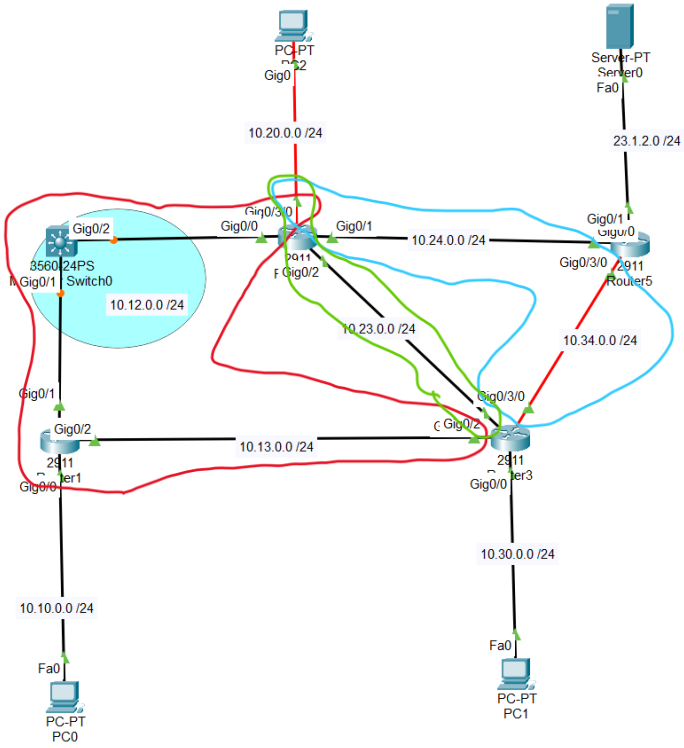
There are 3 ways for a router to learn routes:

1.- Directly connected

2.- Static routes

3.- Dynamic Routing protocol:

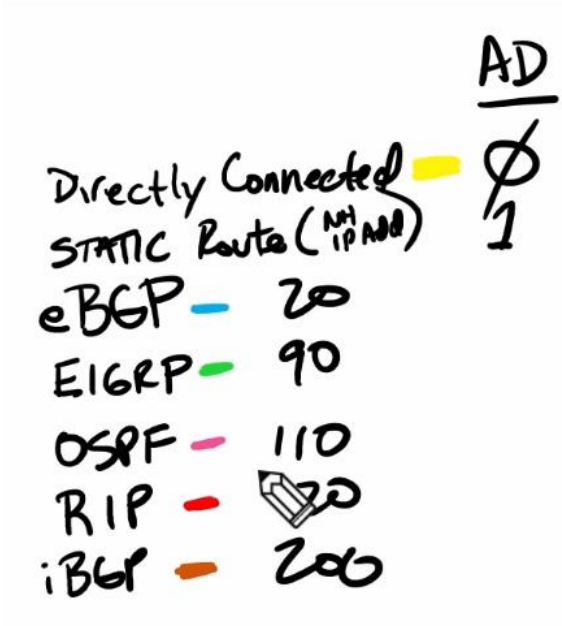




When it comes to routing protocols, several can be ran in a router, learning different routes via different routing protocols. In our topology, let's consider these routes / per routing protocol.

If we look at it carefully, we will see that router 3 will now know how to reach 10.20.0.0 /24 subnet vi 3 different protocols. Which one of these routes will it actually used?

Administrative distance is used when a router has learned distances by 2 or more different protocols. Just like Route costs that we looked at, it will look at the lower cost per protocol per route. [Notice the costs right next to each protocol]. In this case, on R3, to reach 10.20.0.0 /24 sub net it will choose the EIFRP route since it has lower cost.

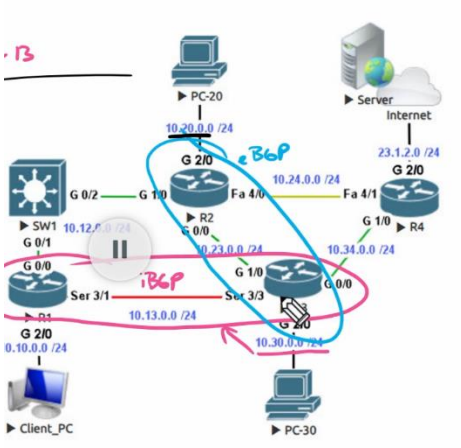


Default Administrative Distances

When it comes to deciding where to route traffic to a subnet, it will always come down to the lowest AD.

BGP Administrative Distances of 20 and 200

Let's look at iBGP [cost 200] and eBGP [cost 20] and reaffirm the difference in Administrative distance between the 2.

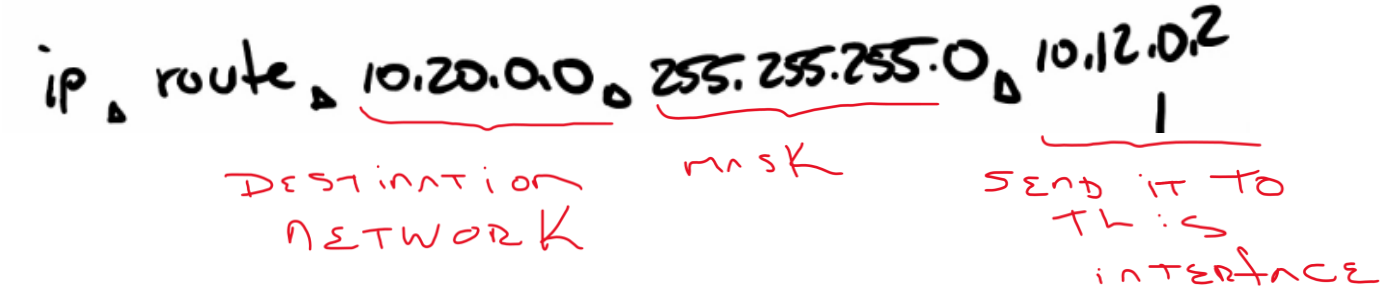


Static Routes with an Administrative Distance of 1

STATIC ROUTES CONFIGURED HAS A DEFAULT AD of 1

60.- Use IPv4 Floating Static Routes

Let's remind ourselves that a static route is manually configuring a route on an interface.

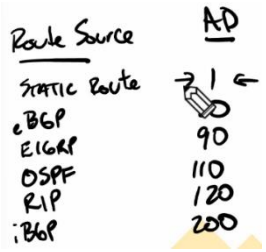


We can set a floating route that in simple terms would be a second option or backup route with the same syntax. But how do we tell the router which one is the PRIMARY and which one is the SECONDARY.

The way we do this is by ARTIFICIALLY CHANGING THE ADMINISTRATIVE DISTANCE.

Let's do an example with our topology

So in our topology if we were to set 2 static routes to the 10.20.0.0 /24 network on R1, both of them would have an AD of 1. What we could do is increase the AD for our backup route. Notice that the backup will not appear on the routing table, but it will be there 'floating' until the primary comes down, or simply is not available.




```
R1(config)#ip route 10.20.0.0 255.255.255.0 10.12.0.2
R1(config)#ip route 10.20.0.0 255.255.255.0 10.13.0.3 2
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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 23.1.2.1 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
C       10.10.0.0/24 is directly connected, GigabitEthernet0/0
L       10.10.0.1/32 is directly connected, GigabitEthernet0/0
C       10.12.0.0/24 is directly connected, GigabitEthernet0/1
L       10.12.0.1/32 is directly connected, GigabitEthernet0/1
C       10.13.0.0/24 is directly connected, GigabitEthernet0/2
L       10.13.0.1/32 is directly connected, GigabitEthernet0/2
S       10.20.0.0/24 [1/0] via 10.12.0.2
O       10.23.0.0/24 [110/20] via 10.13.0.3, 02:15:43, GigabitEthernet0/2
O       10.24.0.0/24 [110/20] via 10.12.0.2, 02:15:53, GigabitEthernet0/1
O       10.30.0.0/24 [110/20] via 10.13.0.3, 02:15:43, GigabitEthernet0/2
O       10.34.0.0/24 [110/20] via 10.13.0.3, 02:15:43, GigabitEthernet0/2
     23.0.0.0/24 is subnetted, 1 subnets
O       23.1.2.0/24 [110/30] via 10.13.0.3, 02:15:43, GigabitEthernet0/2
                    [110/30] via 10.12.0.2, 02:15:43, GigabitEthernet0/1
S*   0.0.0.0/0 [1/0] via 23.1.2.1
```

higher AD than the primary's static route '1'.

Establishing Initial Default Routes

```
R1(config)#ip route 0.0.0.0 0.0.0.0 10.12.0.2
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, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.12.0.2 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
C       10.10.0.0/24 is directly connected, GigabitEthernet0/0
L       10.10.0.1/32 is directly connected, GigabitEthernet0/0
C       10.12.0.0/24 is directly connected, GigabitEthernet0/1
L       10.12.0.1/32 is directly connected, GigabitEthernet0/1
C       10.13.0.0/24 is directly connected, GigabitEthernet0/2
L       10.13.0.1/32 is directly connected, GigabitEthernet0/2
S       10.20.0.0/24 [1/0] via 10.12.0.2
O       10.23.0.0/24 [110/20] via 10.13.0.3, 02:52:41, GigabitEthernet0/2
O       10.24.0.0/24 [110/20] via 10.12.0.2, 02:52:51, GigabitEthernet0/1
O       10.30.0.0/24 [110/20] via 10.13.0.3, 02:52:41, GigabitEthernet0/2
O       10.34.0.0/24 [110/20] via 10.13.0.3, 02:52:41, GigabitEthernet0/2
     23.0.0.0/24 is subnetted, 1 subnets
O       23.1.2.0/24 [110/30] via 10.13.0.3, 02:52:41, GigabitEthernet0/2
                    [110/30] via 10.12.0.2, 02:52:41, GigabitEthernet0/1
S*   0.0.0.0/0 [1/0] via 10.12.0.2
```

Adding Backup "Floating" Default Routes

So after putting in place our PRIMARY route from R1 to the 23.1.2.0 /24 network, we can set our backups and increase <1 our Ads

Remember that if we want to backup many routes with Backup static routes, we will need to see the 'protocol' that these routes were learned from.

Let's do an exercise where WE WILL CONFIGURE DEFAULT STATIC BACKUP ROUTES TO REACH 23.1.2.0 /24 ON EACH ROUTER, COMPARING THEM TO OUR PRIMARY ROUTES.

		Primary	Floating
		AD	NEXT HOP / AD
R1	Static	1	10.13.0.3 / 2
R2	OSPF	110	10.24.0.4 / 115
R3	OSPF	110	10.23.0.2 / 113

For R1

```
S*   0.0.0.0/0 [1/0] via 10.12.0.2

R1(config)#ip route 0.0.0.0 0.0.0.0 10.13.0.2 2
```

AD +1

For R2

```
O*E2 0.0.0.0/0 [110/1] via 10.24.0.4, 00:33:01, GigabitEthernet0/1

R2(config)#ip route ?
  A.B.C.D  Destination prefix
R2(config)#ip route 0.0.0.0 0.0.0.0 10.24.0.4 115
```

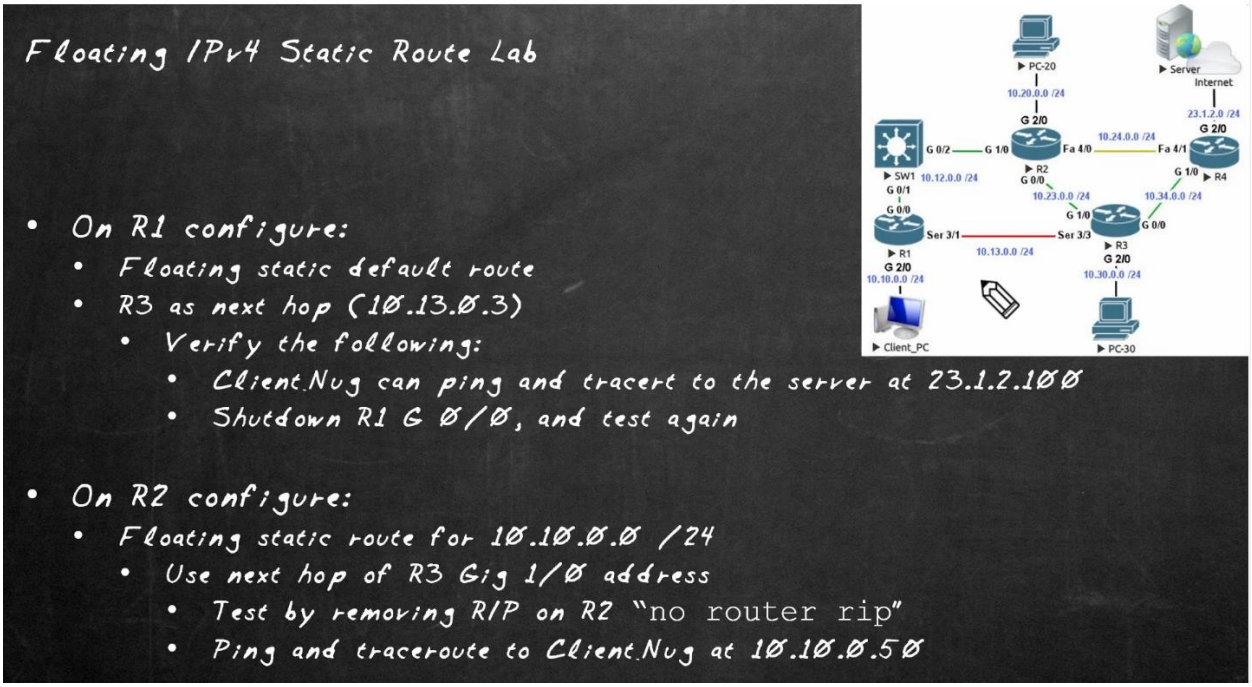
AD +5

For R3

```
O*E2 0.0.0.0/0 [110/1] via 10.34.0.4, 00:34:14, GigabitEther

R3(config)#ip route 0.0.0.0 0.0.0.0 10.23.0.2 113
```

AD +3

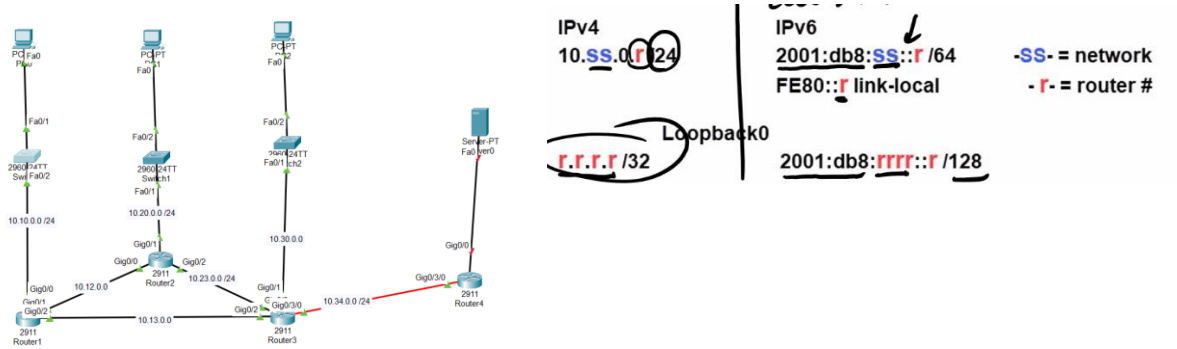
VALIDATION



61.- Use Dual Stack IPv4 and IPv6

We are going to see how to set up a network that works with both IPv4 & IPv6

Gameplan:




```
R1(config)#ipv6 unicast-routing
```
In order to enable IPv6 use this command

We will also add OSPF area 0 on all interfaces for IPv4.

And for IPv6 we are going to use RIP.

Configure IPv6 Global and Link-local Addressing

```
R1(config)#int g0/0
R1(config-if)#ipv6 ad
R1(config-if)#ipv6 address 2001:db8:10::1/64
R1(config-if)#ipv6 address fe80::1 link-local
```

Configure Basic IPv6 Routing Using RIP

After configuring all interfaces on all routers, they only know IPv6 routes directly connected to them. Using a routing protocol like RIP, will give our routers the ability to share ipv6 routes between them.

The way to do this is similar to OSPF. You go to each interface and declare them to run RIPv6

```
R1(config)#int range g0/0-2
R1(config-if-range)#ipv6 rip OurRIP enable
```

Name of the process

Now all our routers are sharing IPv6 routes via RIP. For some reason I had to configure each interface individually, the range command was not working.

Using IPv6 Anycast

Remember that an anycast address is that is a single address in multiple places. Like GOOGLES

3001::1 is Google's DNS IPV6 address. In our topology, we will add loopback addresses of 3001::1/128 to both R1 and R3

That way, when a client needs to go to our anycast address of 3001::1 it will go to the 'nearest' one, and it's up to the router's RIP scheme to decide this.

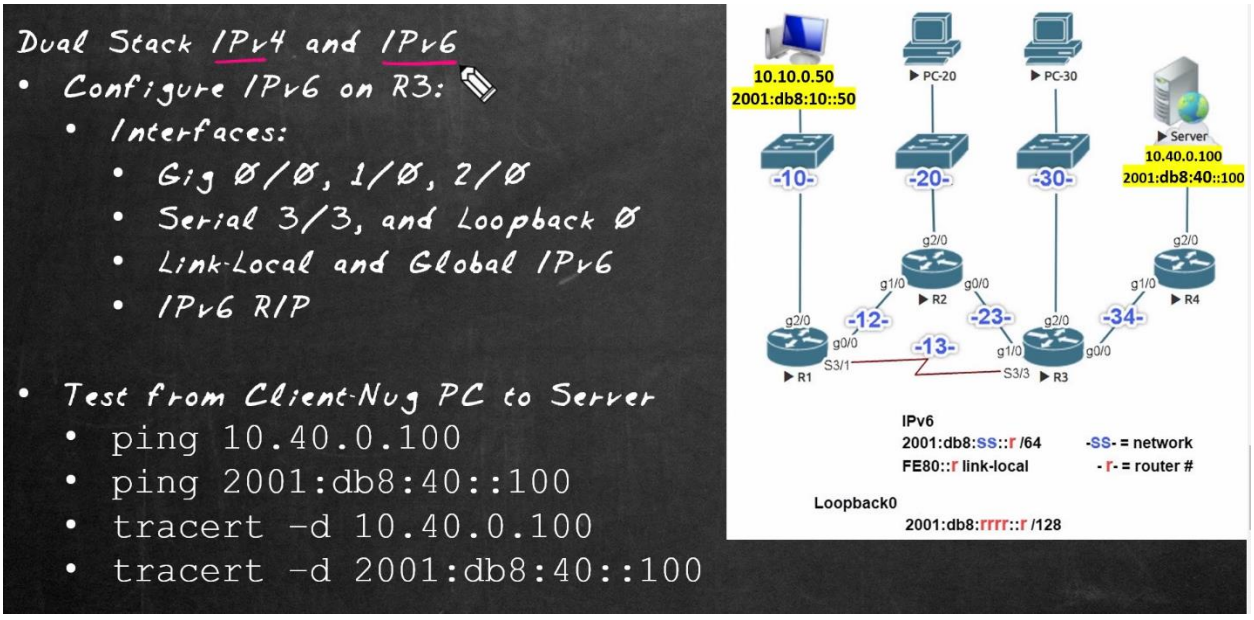
```
R1(config)#int loop 1

R1(config-if)#
%LINK-5-CHANGED: Interface Loopback1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed
state to up

R1(config-if)#ipv6 address 3001::1/128 anycast
```

```
R3(config)#int loop 1
R3(config-if)#ipv6 address 3001::1/128 anycast
```

Now we will also want for RIP to advertise this IPV6

```
R1(config-if)#ipv6 rip OurRIP enable
```

VALIDATION



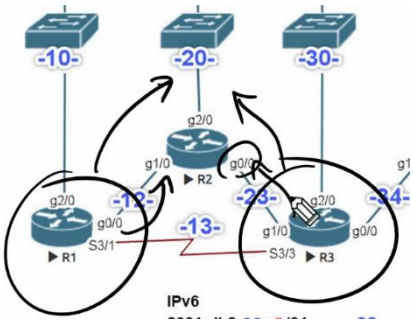
62.- Configure Static IPv6 Routes

They are very similar to how we set ipv4 STATIC ROUTES in which that we are giving our routers instructions on how to get to certain subnets.

If we are sending to a next-hp-address to a LINK LOCAL ADDRESS we would need to especify the interface that is going out from because there are multiple link local addresses for every interface.

If we want a backup OR floating IPV6 stating route, we can manipulate the AD as we did with IPv4.

Configure Static IPv6 Network Routes



We are going to train R1 & R3 to reach the -20- network on IPV6. We will use a Link Local address on R1 and a GLOBALLY routable ID for R3

On R1 we already have a route for the -20- subnet learned via RIP

```
R    2001:DB8:20::/64 [120/2]
```

So we will simply add a static route with an AD automatically set to 1

In R3 since we are going to use a GLOBAL ADDRESS, we can choose not to use the egress interface, BUT WITH A LINK LOCAL ADDRESS, WE HAVE TO ESPECIFY BOTH

```
R1(config)#ipv6 route 2001:db8:20::/64 g0/1 FE80::2
S    2001:DB8:20::/64 [1/0]
```

Now for R3:

```
R3(config)#ipv6 route 2001:db8:20::/64 2001:db8:23::2
```

Configure a Floating IPv6 Static Route

Remember that a floating static route is a backup. We just have to make sure that for both floating static routes, we set a higher AD value.

```
R1(config)#ipv6 route 2001:DB8:20::/64 g0/2 FE80::3 2
R3(config)#ipv6 route 2001:DB8:20::/64 2001:DB8:13::1 2
```

HIGHER AD

Configure a Static IPv6 Host Route

Static routes not only have to be towards networks. We can also set Hosts with /16 /8 bits or even /0 masks. Let's configure static Host Routes towards PC 2 on the -20- subnet.

```
R1(config)#ipv6 route 2001:db8:20::10/128 2001:db8:13::3
```

```
R1#show ipv6 static
IPv6 Static routes Table - default
Codes: * - installed in RIB, u/m - Unicast/Multicast only
            U - Per-user Static route
            N - ND Static route
            M - MIP Static route
            P - DHCP-PD Static route
            R - RHI Static route
*   2001:DB8:20::/64 via FE80::2, GigabitEthernet0/1, distance 2
*   2001:DB8:20::10/128 via 2001:DB8:13::3, distance 1
```

Configure IPv6 Primary and Floating Static Default Routes

Remember that a default route is when a router does not know where to send traffic to a specific address, it will send it to a default IPv6 address.

```
R2(config)#ipv6 route ::/0 2001:db8:23::3
```

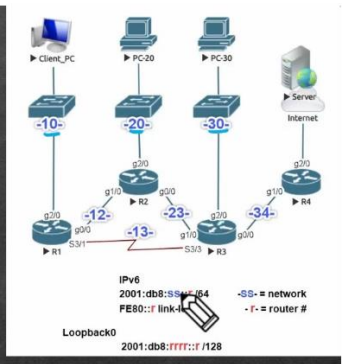
Default IPV6 address

Validation

IPv6 Static Routing Lab

R2: default route using R3, with backup through R1
R1 and R3:

- route to 2001:db8:20::/64 using R2 as next hop
  - R1 uses the next hop of R2 Global
  - R3 uses the next hop of R2 Link Local



R1: floating route to 2001:db8:23::/64 using R2 as next hop
R1: /128 host route to PC-20 using interface S3/1

R1 and R2: Add any required static routes to allow connectivity between the loopbacks on R1 and R2

- Test on R1 or R4 using the following method:
- ping 2001:db8:**rrrr**::**r** source loopback 0

63.- Routes That Must Win Twice

The Winning IPv4 Route Overview

There are 2 main things to consider when we say that a ROUTE MUST WIN to 'route' traffic:

1.- The first thing is that the ROUTES MUST BE INSTALLED IN THE ROUTING TABLE.

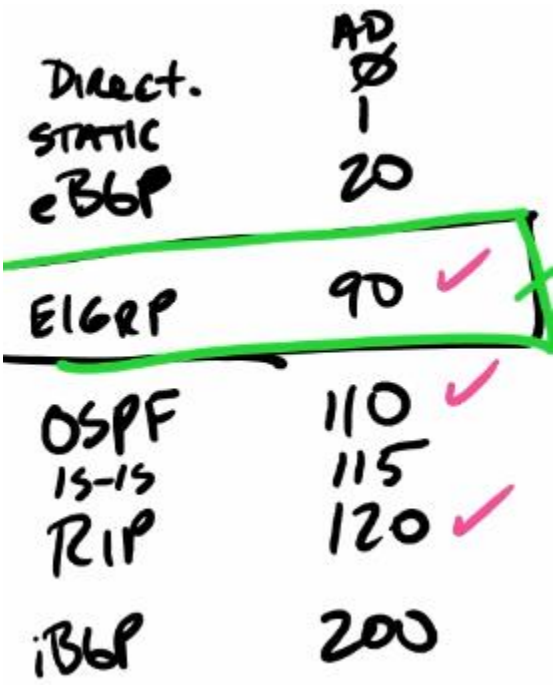
These are important to be considered because in case we have 2 identical networks like 23.1.2.0 /24 & 23.1.2.0 /25 only the /24 would be saved in the routing table.

THIS DOES NOT MEAN THAT THE ROUTE IS GOING TO BE USED

2.- If we were to configure a static route to 23.1.2.0 /25 [which is completely different from the /24 one] only the LONGEST MATCH [MASK] WILL BE USED.

→ Let's imagine that traffic arrives with a destination to 23.1.2.100, this address belongs to both /24 and /25 subnets. It turns out that the router will choose the /25 network.

Confirming Longest Match Wins, and the Use of AD



As an example, which AD's protocol will be chosen when trying to forward traffic to the 1.1.1.1 IP address?

0.0.0.0 /0 via Static Route
1.1.1.0 /29 via iBGP
1.1.1.0 /24 via OSPF
1.1.1.0 /30 via RIP
1.1.0.0 /16 via IS-IS

It does not matter the protocol that it's being used or it's AD. If we got several routes using different protocols to the same network ID but different subnet masks, the LARGEST SUBNET MASK will be chosen.

Equal Cost Routes, AD, and Longest Match Examples

Let's see this example

```
D        10.0.7.32/30
           [90/2170112] via 10.0.6.194, 01:05:52, GigabitEthernet0/0
           [90/2170112] via 10.0.6.1, 01:05:52, GigabitEthernet1/0
```

We can see that there are 2 possible routes to that subnet. We can also see that both routes were learned by EIGRP. Both routes have an AD of 90 and same cost. This tells us that in order to have 2 or more routes to a same subnet, these must have same COST & same AD.

Another example:

```
S        10.0.7.32/30 [1/0] via 10.0.6.194
S        10.0.7.32/31 [1/0] via 10.0.6.194
```

What route will be chosen to reach 10.0.7.33?

Both routes are static with the same AD and .33 falls in the same range for both. But /31 for being the largest mask, will be chosen to route traffic to that particular host.

Calculate the Range of an IPv4 Route

Let's see an example of a packet needed to be forwarded to 10.0.6.130 and all the possible routes there are in a router. Which one will it choose? At this point, we no longer care about the route's AD or cost since it can all fall in possible routes but which one has the LONGEST NUMBER OF BITS [mask] that matches that address

```
D        10.0.6.0/25 [90/3072] via 10.0.6.193, 01:26:23, GigabitEthernet1/0
R        10.0.6.128/25 [120/2] via 10.0.6.193, 00:00:20, GigabitEthernet1/0
D        10.0.6.128/26 [90/28416] via 10.0.6.193, 01:25:56, GigabitEthernet1/0
R        10.0.6.128/27 [120/2] via 10.0.6.193, 00:00:20, GigabitEthernet1/0
C        10.0.6.192/26 is directly connected, GigabitEthernet1/0
L        10.0.6.194/32 is directly connected, GigabitEthernet1/0
```

For 10.0.6.128 /25 → 7 bits left on last octet → 2 ^ 7 – 2 = 128 – 2 = 126 → RANGE : (128 - 255)

For 10.0.6.128 /26 → 6 bits left on last octet → 2 ^ 6 – 2 = 64 – 2 = 62 → RANGE : (128 - 191)

For 10.0.6.128 /27 → 5 bits left on last octet → 2 ^ 5 – 2 = 32 – 2 = 30 → RANGE : (128 - 159)

For 10.0.6.128 /26 → 4 bits left on last octet → 2 ^ 4 – 2 = 16 – 2 = 14 → RANGE : (128 - 143)

For 10.0.6.128 /32 → 0 bits left on last octet → ONLY 1 HOST

We can see here that all first 4 are possible subnets that traffic towards 10.0.6.130 could be sent to, but the largest number of bits [mask] would be 10.0.6.128 /26

What about 10.0.6.162? ➔ Only the first 2 could be considered, but the largest mask is /26.

What is the full range, including the subnet ID and the broadcast address, for the route: 10.0.6.0 /25

/25 → 7 bits left on last octet ➔ 2 ^ 7 – 2 = 128 – 2 = 126 ➔ RANGE (.0 - .127)

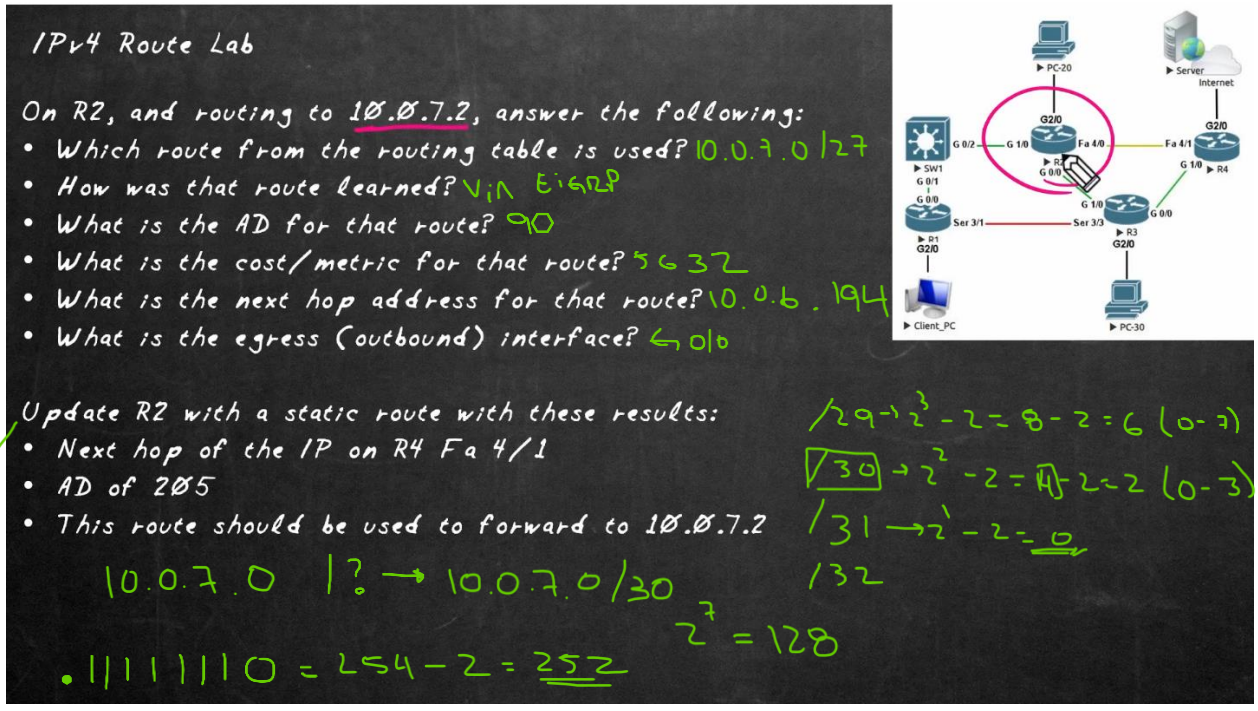
The Price of a Route: Cost | Hop Count | Metric

REMEMBER THAT when we talk about COST, it refers to the SUM OF THE EGRESS INTERFACESS TORWARDS THE FINAL DESTINATION.

Different protocols have different ways although similar to decide route costs.

- ➔ OSPF = Does it in the form of COST. It can tell differences between Fa vs Gig vs Ser. Each with a higher or lower cost making them better or worst.
- ➔ RIP = Not that good when it comes to dealing with a lot of information. RIP only uses HOP COUNTS. Does not consider interfaces, bandwidth.
- ➔ EIGRP = Uses COMPOSITE METRIC. Boils down to 2 factors → 1.- Worst bandwidth? So it can close up those options & 2.- SUMS OF DELAYS.

Validation

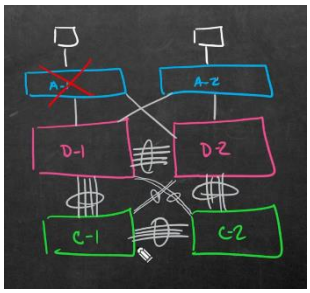



```
R2(config)#ip route 10.0.7.2 255.255.255.255 10.0.6.130
R2(config)#ip route 10.0.7.2 255.255.255.252 10.0.6.130 205
%Inconsistent address and mask
R2(config)#ip route 10.0.7.0 255.255.255.252 10.0.6.130 205
```

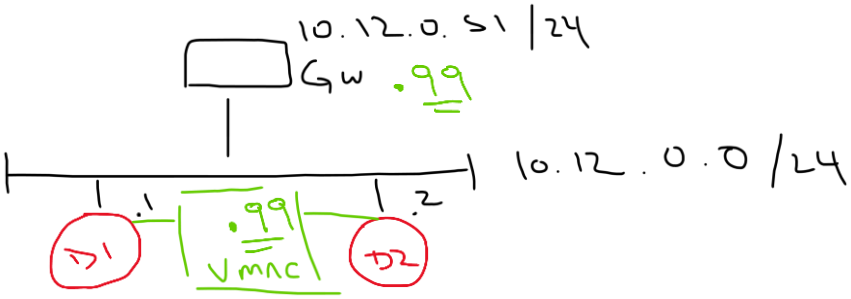
64.- Use First Hop Redundancy Protocols

First Hop Redundancy Protocol (FHRP) Overview

Fault Tolerance is the capacity to have a single fault and still 'tolerate' that. We build it in different ways:



- ➔ Physical: With multiple connections.
- ➔ Logical: Let's imagine in the example on the right that client connected to A-1 has a GW of 10.12.0.1 on Router D-1. What happens if the interface for it's VLAN running on D-1 goes off? Even if D-2 is running VLAN 12, it's gateway would be .2
  We can create fault tolerance by implementing FHRP
  First Hop Redundancy Protocol works by teaching routers to also use



The VIRTUAL IP ADDRESS OF .99 as a default GW. There will also be a VIRTUAL MAC ADDRESS assigned to this virtual IP. In case one physical interface GW shuts down, traffic will still be able to be routable via .99 Virtual GW that is also running on D2.

Cisco's Hot Standby Router Protocol (HSRP)

Created by Cisco. In our topology and for the sake of this explanation, we will use R1 and R2 as an example. For HSRP we use a group number for devices in the same VLAN.

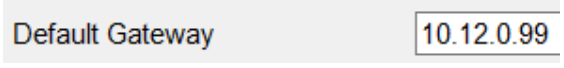
GAMEPLAN: HSRP group #5

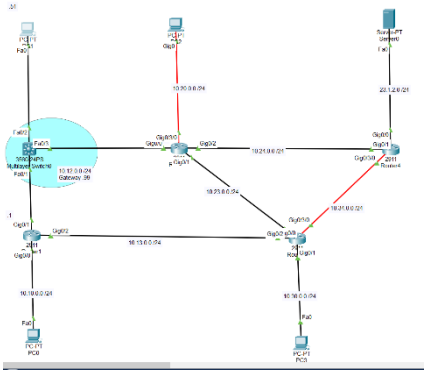
VIP: 10.12.0.99 that since both R1 & R2 are in the same group, both are going to support it.

Is noticeable to say that both routers in our HSRP group are constantly sending multicast frames to each other. Also, there will be an ACTIVE and STANDBY router, the priority will decide which one will be which.

Configure HSRP

Notice in our topology That PC 1 has a DG of 10.12.0.99 that is yet to be configured as a virtual Gateway on both R1 & R2 for GROUP 5





On R1

```
R1(config)#int g0/1
R1(config-if)#standby 5 ip 10.12.0.99
```

By default this will be the active virtual IP GATEWAY but we want to change that by changing the priority value.

```
R1(config-if)#do show standby
GigabitEthernet0/1 - Group 5
  State is Active
    5 state changes, last state change 00:44:33
  Virtual IP address is 10.12.0.99
  Active virtual MAC address is 0000.0C07.AC05
    Local virtual MAC address is 0000.0C07.AC05 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.474 secs
  Preemption disabled
  Active router is local
  Standby router is unknown
  Priority 100 (default 100)
  Group name is hsrp-Gig0/1-5 (default)
```

```
R1(config-if)#standby 5 priority 105
```

```
R1(config-if)#standby 5 preempt
```

Enable preempt to make the highest priority router, the active router

On R2

```
R2(config)#int g0/0
R2(config-if)#standby 5 ip 10.12.0.99
```

```
R2(config-if)#standby 5 preempt
```

Configure and Test VRRP

Virtual Router Redundancy Protocol is in case we don't have all Cisco routers. Funcitonality is the same in which we create a group. For our topology, we will use group #6 with a VIP of 10.12.0.6

The priority is set to 100 by default and preempt is also enabled by default.

```
R1(config-if)#vrrp 6 ip 10.12.0.6
```

```
R1#show vrrp
```

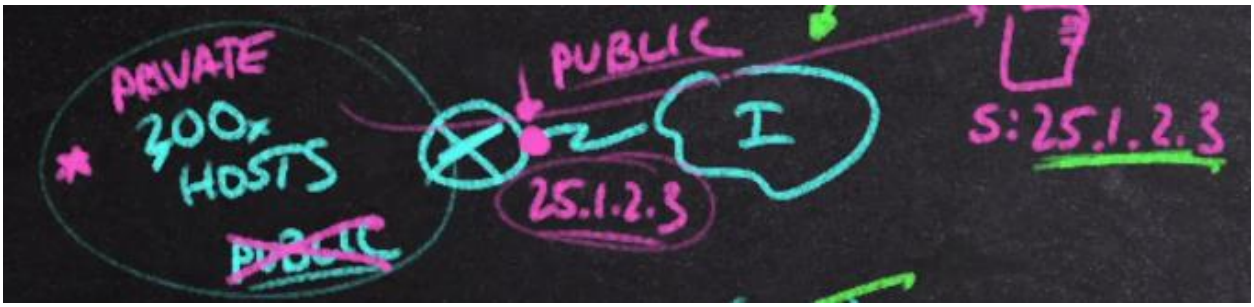
If the priority are the same, then the tiebreaker is the highest IP address on the interface.

65.- Explain Network Address Translation (NAT)

NAT helped us use a private IP address to route them on the INTERNET.

IPv4 has basically ran out of IP addresses. This was anticipated back in the 90s when IPv6 was starting to get introduced. NAT was borned as a result of this. NAT tries to extend IPv4 life span in which it leverages PRIVATE ADDRESSES with PUBLIC ADDRESSES.

When we have multiple Private IP addresses and they want to communicate outside it's own LAN, they will route with a single PUBLIC address. This makes us to be able to reuse private IP addresses and making the Internet a PUBLIC IP NETWORK.



Private IP Addressing

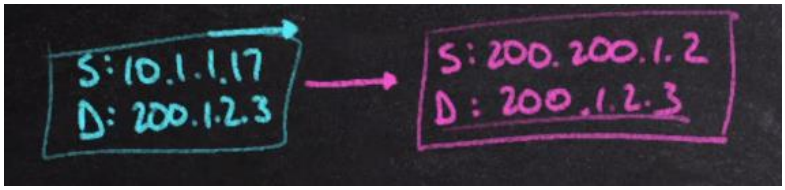
A private IP address is intended to be used in a network that will not use these addresses on the Internet.

RFC 1918 describes the use of PRIVATE ADDRESSES

- ➔ 10.0.0.0 /8: Used in ENTERPRISE. 2 ^ 24 = 16,777,216 possible addresses.
- ➔ 172.16.0.0 /12
    - On the second octet only 4 bits are used completely, leaving 4 possible bits to be used. This translates into 15 more possible subnets we can have making the range of the second octet 16 - 31
- ➔ 192.168.0.0 /16 Used in homes. 2 ^ 16 = ~65k addresses.

NAT Architecture

The router will translate the address inside the IP HEADER [SOURCE ADDRESS]. Therefore, creating a new packet.



When a packet returns, the DESTINATION address will be the NAT address that was changed for. Inside the router, NAT creates a translation table that saves these correlations.

Source and Destination NAT

We can also translate the DESTINATION address. Although very rare to do, the way this works is that when we want to DISGUISE A DESTINATION or RESOURCE.

It would be the same process in terms of the NAT router changing the destination instead of the source.

Address Pools and Overload

Public addresses come from pools. Let's explain this with an example.

We have the public address of 25.1.2.3 /24 and let's say that for our Pool of address we will have 25.1.3.0 /29 which will give us 6 address in total. .1 --- .6

The NAT router will map each host asking for one of these addresses in our pool. What happens if we fall out of addresses?

- ➔ We simply won't be able to assign any more addresses.
- ➔ But we have an overload option which means TO SHARE an IP address already being used. The way the Router does this is the LAST IP ADDRESS THAT IT USED, and then starts all over again assigning from the beginning.

There is another way NAT works and is simply by sharing the single PUBLIC address to all the private addresses but using PAT [more to come] for this to work.

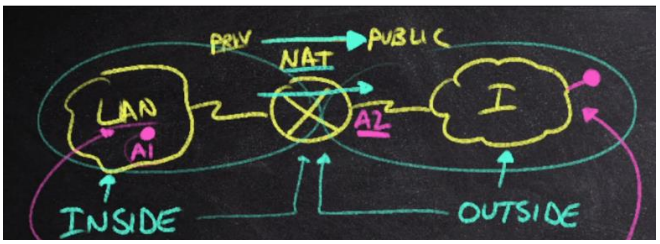
66.- Define NAT Operation

NAT Terminology

Let's say in this example that we are doing source NAT translation. Priv → Public. There are 2 terms we need to understand:

- ➔ Inside of the network: The LAN is our inside network. We are basically making 2 domains that when the packets transition between Lan -> Internet. When the packet traverses from the inside to the outside it is when NAT is performed. The ADDRESS TYPE can be described as LOCAL & GLOBAL.
    - LOCAL ADDRESSES → Addresses that are seen inside our LAN
    - GLOBAL ADDRESSES → Addresses seen on the internet.

      Hosts on the internet see as the address of an Inside host see them as a Global address.



- ➔ Outside of the Network

Dynamic NAT

Is primarily used when traffic is initiated from the inside to the outside. We talk about many hosts that share a POOL of addresses.

Static NAT

Why does the public IP matter? When we are initiating from the outside, a single non-changing IP address needs to exists in the NAT exchange when traffic from the outside tries to reach a specific host in our LAN. So let's say that our private IP address for a server is 10.17.4.5 and NAT assigns a non-changing PUBLIC address of 25.17.2.3 both routable and known on the internet.

Now, usually Organizations won't get a lot of PUBLIC IP ADDRESSES. So what is done in a real world scenario is that a low number of these are going to be used for STATIC NAT which a single one will be used for PAT.

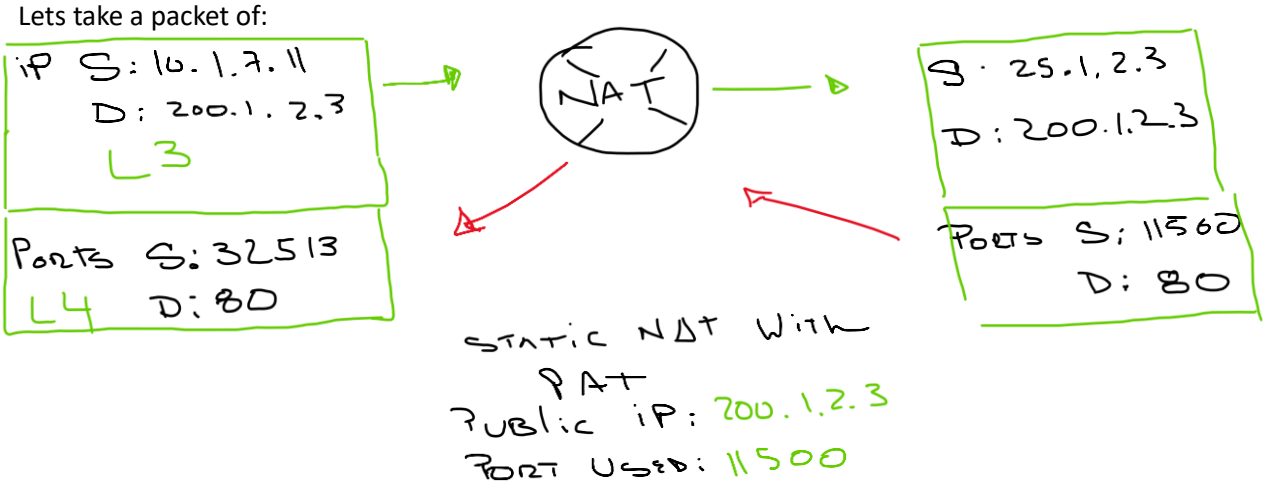
PAT

When looking at PAT the concept of OVERLOAD comes into the conversation. OVERLOAD is when many hosts are sharing a single translated IP address. When this happens, the NAT translator recognizes traffic coming in and out of our LAN by which PORTS are being used, therefore creating a correlation between PORTS -> INTERNAL IP ADDRESSES.

Let's say we have 200 hosts in our LAN and all of them share the same 25.1.2.3 PUBLIC ADDRESS. Meaning they all appear as if they are the same address on the Internet. How does the NAT translator decide? When using PAT, via ports.

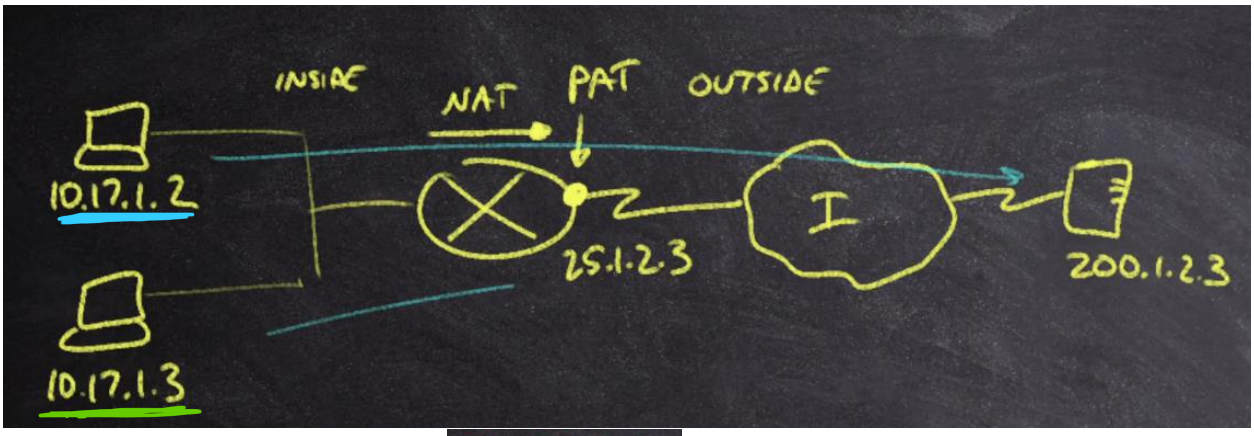
Remember that PORTS work at L4. TCP and UDP are part of this solution. The destination port of a HOST sending traffic to another one on the internet must match, but the SOURCE PORT IS ARBITRARY.

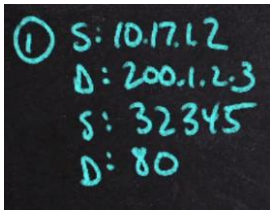
Lets take a packet of:

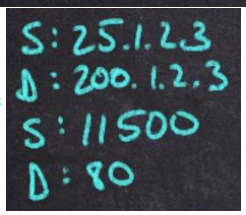


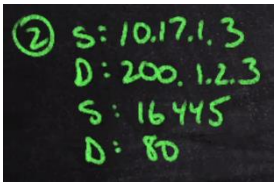
In other words, the SOURCE PORT IS USED FROM A MARKING PERSPECTIVE TO HELP MAP our internal IP ADDRESSES with the ports they used to send traffic.

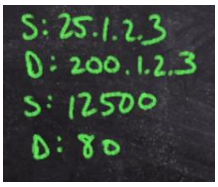
PAT Example

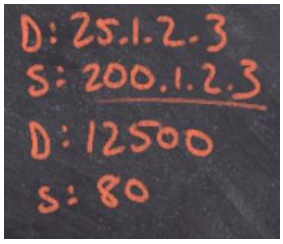








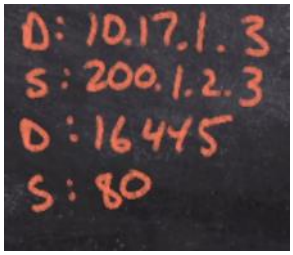




The NAT will look at this packet and by the D: PORTS, it

Will know that it is a response to the 10.17.1.3 host.

The only thing to remember is that PAT takes into consideration the SOURCE PORT when assigning them to a specific host inside our LAN.



NAT Address Types

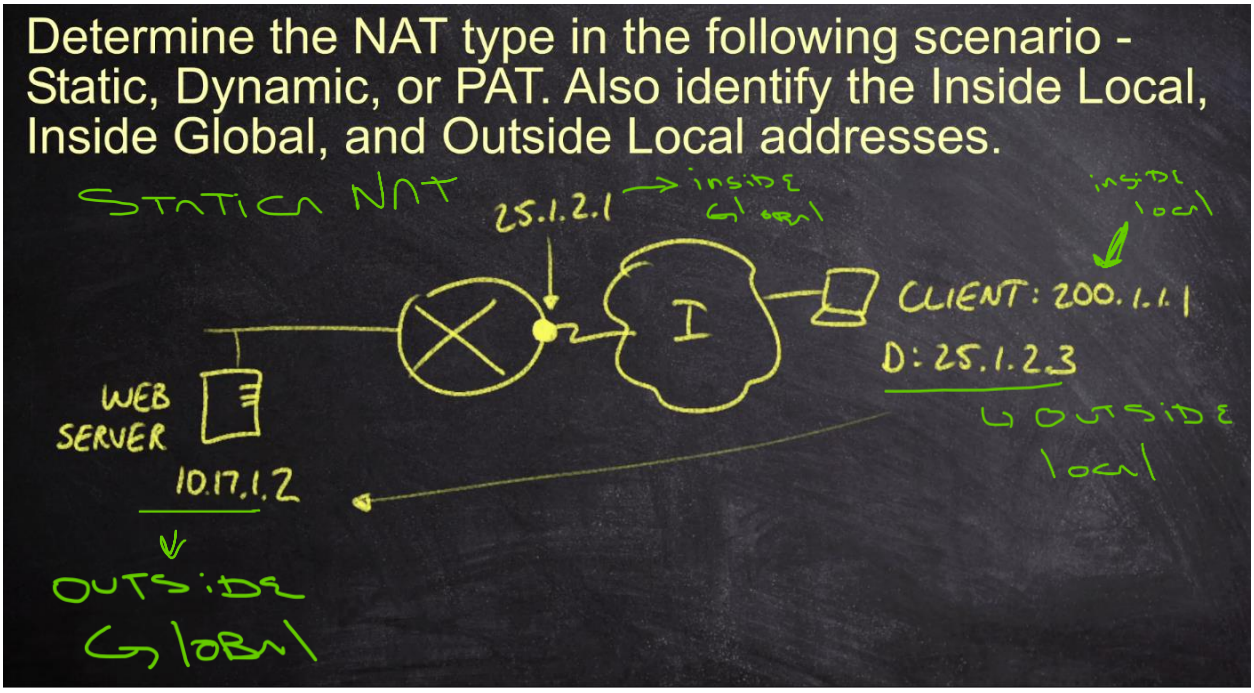
INSIDE LOCAL → address as seen by the LAN

Inside Global → Address as seen by the Outside network

Outside Local: Address of a host as seen by the outside network.

Outside Global: Global address on the internet as seen by everyone else.

Validation



67.- Configure Network Address Translation (NAT)

First we'd need to figure out which NAT we are deploying.

Firs we configure our INSIDE & OUTSIDE on an interface

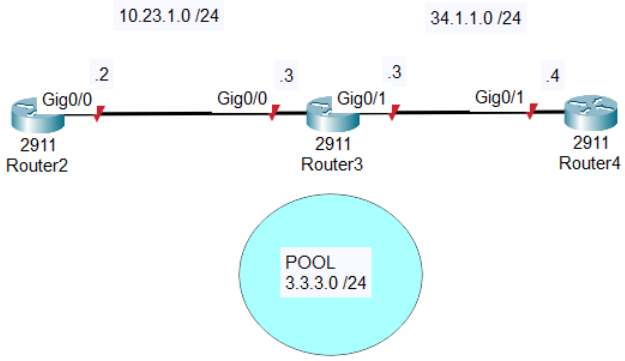
For OUTSIDE ZONES:

- ➔ Static: Configure GLOBAL STATIC CONFIGURATION. Simply map 1 IP ADDRESS – 1 IP ADDRESS
- ➔ Dynamic: CREATE POOL + ACL (Access control list).
- ➔ Interface: Where we configure PAT. With a global command we select which interface to use.

Topology and NAT Interfaces

1.- First, let's put together our Inside/outside boundaries

```
R3(config)#int g0/0
R3(config-if)#ip nat inside
R3(config-if)#int g0/1
R3(config-if)#ip nat outside
```



In our topology we will practice all types of NAT. Let's start with:

2.a.- Static: We will give a .2 for Router 2 gig0/0 and provide the NAT address 3.3.3.0 so Router 4 will see this address.

First tell the R3 where out INSIDE AND OUTSIDE boundaries reside

```
R3(config)#int g0/0
R3(config-if)#ip nat inside
```

```
R3(config)#int g0/1
R3(config-if)#ip nat outside
```

Then on global config, create the static configuration for our specific IP ADDRESS

(config)# **ip nat inside source static** [inside-local] [inside-global]

```
R3(config)#ip nat inside source static 10.23.1.2 3.3.3.2
```

(How the 'outside' will see traffic coming from 10.23.1.2

Configure Dynamic NAT

In this case, we are actually configuring our Pool.

Remember that the Pool requires the ACL which the purpose is to match IP addresses.

1.- Pool

(config)# **ip nat pool** [name] [address-start] [address-end] **netmask** [mask]

```
R3(config)#ip nat pool CBT 3.3.3.1 3.3.3.3 netmask 255.255.255.0
```

2.- Create ACL

```
R3(config)#ip access-list extended ANY
R3(config-ext-nacl)#permit ip any any
```

3.-
```
R3(config)#ip nat inside source list ANY pool CBT
```

Configure PAT

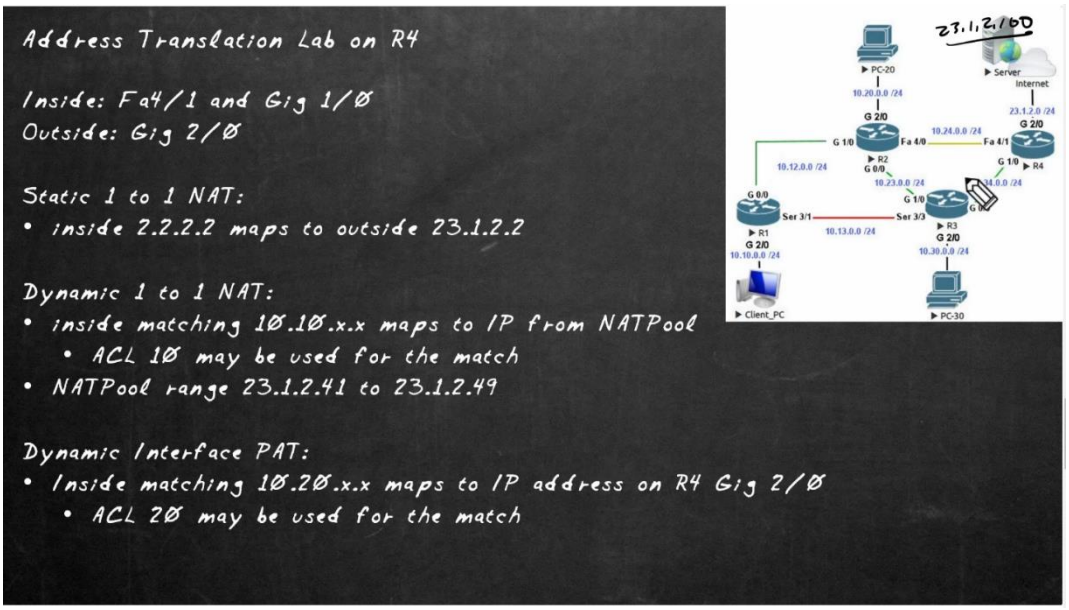
Remember that OVERLOAD is how Cisco specifies configuring PAT.

In our topology we will OVERLOAD g0/1 on R3 so that every IP address on our INSIDE network will be able to use that particular single IP address on R3. So NO POOL is being used.

We are STILL USING THE ACL previously configured.

```
R3(config)#ip nat inside source list ANY interface g0/1 overload
```

VALIDATION



For the Dynamic 1 to 1 NAT: When we match 10.10.x.x it means any traffic that comes from the 10.10.0.0 /24 network. For this, we:

1.- First create a NATPool that contains the IP addresses range of 23.1.2.41 – 49

```
R4(config)#ip nat pool NATPool 23.1.2.41 23.1.2.49 netmask 255.255.255.0
```

2.- Now we need to assign the ACL

```
R4(config)#ip nat inside source list 10 pool NATPool
```

```
R4(config)#do show ip nat st
Total translations: 1 (1 static, 0 dynamic, 0 extended)
Outside Interfaces: GigabitEthernet0/0
Inside Interfaces: GigabitEthernet0/2/0 , GigabitEthernet0/3/0
Hits: 0  Misses: 25
Expired translations: 0
Dynamic mappings:
-- Inside Source
access-list 10 pool NATPool refCount 0
 pool NATPool: netmask 255.255.255.0
       start 23.1.2.41 end 23.1.2.49
       type generic, total addresses 9 , allocated 0 (0%), misses 0
```

68.- Explain Dynamic Host Configuration Protocol (DHCP)

When using DHCP we are scaling usage of IP addresses. Basically, a dhcp server automatically assigns IP addresses to all host connected to the server.

A DHCP server does not only provide IP addresses, it also:

- ➔ Provides DFGW
- ➔ DNS server
- ➔ Other Options

To do this, we need to consider NETWORK ARCHITECTURE.

The main method is CLIENT → SERVER connection. The server has POOL of addresses, usually for each VLAN. It relays on a L2 Broadcast.

A DOMAIN CONTROLLER is a highly available service that the DHCP and DNS servers are connected to.

When it comes to DHCP there are a couple of implications like:

- ➔ Lease time of IP addresses.
- ➔ Pool running out of IP addresses

Leases will come with a timer which is configurable.

IPv6 have STATELESS AUTO CONFIGURATION which basically provides the hosts abilities to know in which networks they are in.

IPv6 hosts once they detect the network, they can create their own IPV6 via EUI-64. What IPv6 provides is the idea of administering pools, tracking addresses.

DHCP follows the DORA process. DHCP uses UDP port 68 [sending host] 67 [sending server]

D
O
R
A

------>DISCOVER: L2 BC → Where the l2 Source address is Host Mac and the L2 destination is ff:ff:ff:ff:ff:ff. L3 S: 0.0.0.0 DA: 255.255.255.255

<-------The OFFER is a packet that offers an IP address. L2 SA : Server MAC L2 DA: Host MAC. L3 SA: server IP and L3 DA: 255.255.255.255

--------->The host sends a REQUEST. L2 SA: host MAC L3 DA: Server MAC. L3 SA: 0.0.0.0 L3 DA 255.255.255.255

←-------- ACKNOWLEDGEMENT L2 SA: HOST DA: HOST

L3 SA: Server IP DA:STILL 255.255.255.255

Until this entire process is not locked in, the IP address on our DHCP pool is not going to be sent to the host.

DHCP Example

After assigning IP addresses to our interfaces, we need to create our POOL. We can have Cisco Routers work both as DHCP Clients and DHCP servers. POOLS ARE ASSIGNED PER SUBNET/PER VLAN. In our example we will have 10.1.1.0 /24 subnet and the DHCP will handout IP addresses within that range. We will use R2 as a DHCP server and R3 as the client.

```
Router(config)#ip dhcp pool LAB
Router(dhcp-config)#network 10.1.1.0 255.255.255.0
Router(dhcp-config)#default-router 10.1.1.2

Router(config)#ip dhcp excluded-address 10.1.1.1
```

ON R3 we will tell the interface that it will use DHCP

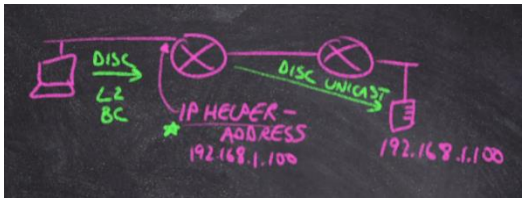
```
R3(config-if)#ip address dhcp
```

And we will be given a dhcp IP address of 10.1.1.3

DHCP Relay

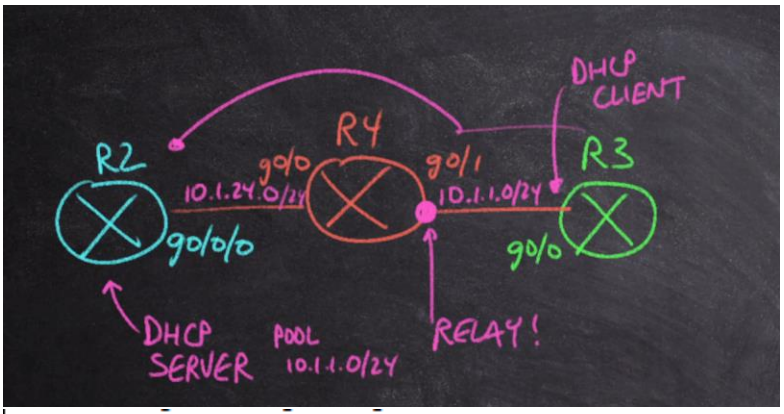
DHCP relies on L2 broadcasts which do not traverse L3 boundaries. Servers can be installed anywhere in the network, yet devices need L2 broadcasts to locate them. What happens when we need to traverse a router L3 for example?

DHCP RELAY helps us traverse a L3 router. We configure the specific interface in which traffic needs to traverse. What happens in the interface is that through RELAY, we assign the IP address of the DHPC server and whenever a L2 broadcast arrives, it will send it as a UNICAST to the DHCP server.



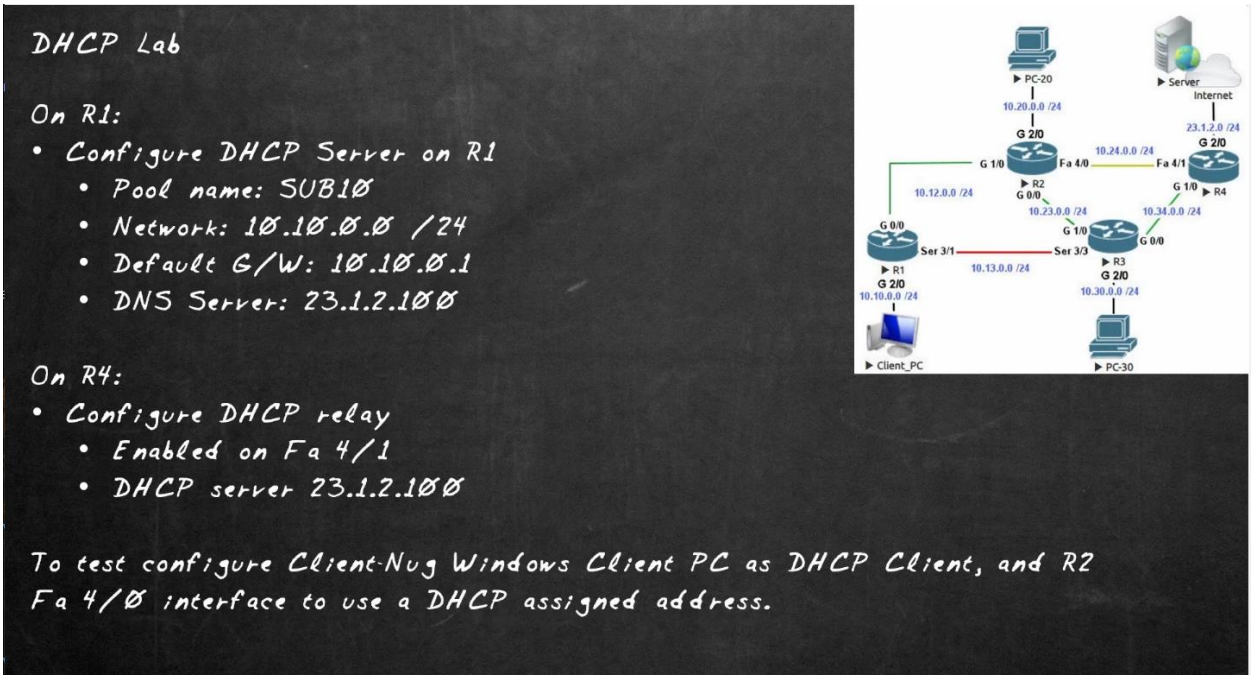
The HELPER-ADDRESS is the address we assign to the router to send back unicast to the DHCP server

Configure DHCP Relay




```
R4(config-if)#ip helper-address 10.1.24.2
```

VALIDATION



69.- Explain the Domain Name System (DNS)

DNS Servers and Hierarchy

When a client tries to communicate to a website [cbtnuggets.com] it will communicate to the local DNS SERVER where a database of NAMES → IP addresses.

The Client also needs to know WHERE the local DNS server is located.

DNS also works as a hierarchical structure where local DNS SERVERS need to communicate with upper-level servers that have more information about mapping IP addresses to domain names. The local DNS servers reach UPSTREAM.

DNS Process

When a host is attached to a network, it sends DHCP traffic to receive an IP address. Whenever there is IP addresses remember that they can start communicating to an outside network.

When a client sends DNS request, it uses UDP 53 and it responds on TCP 53. Both hosts and servers maintain a cache that needs to be updated with the DNS information.

In CISCO IOS-XE we configure a DNS SERVERS with the global command of:

ip name-server [ip address]

We will also need to configure the domain name

ip domain-name [domain name]

To input local DNS entries in our router.

Ip host R4 [ip address]

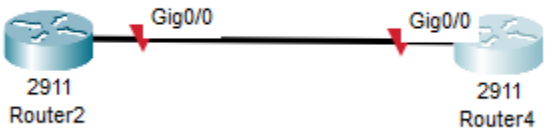
Enabling/disabling DNS lookup

[no] ip domain-lookup

Configure DNS

1.- Configure DNS server &

2.- Configure DNS domain name



```
R2(config)#ip name-server 1.2.3.4
```

```
R2(config)#ip domain-name example.com
```

3.- We will configure a local DNS entry to R4 and now we can ping R4

```
R2(config)#ip host R4 10.1.24.4
```

```
R2#ping R4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.24.4, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/0 ms
```

70.- Explain Simple Network Management Protocol (SNMP)

Simple network management protocol is a SERVER APPLICATION that helps us retrieve and push configurations. SNMP consists of a few architectural components:

1.- SNMP manager station → Capable of showing us the network management stations. Also shows us possible problems like Bandwidth/hardware problems, etc.

2.- SNMP devices → Under the hood of IOS-XE there is a SNMP software that each device needs to be running. We refer to these as SNMP agents. Through this software we are GETTING/SETTING SNMP management towards our devices inside a network.

3.- Database → Known as the Management information Base. It is a collection of all the individual components that can be managed. Each component is called OBJECT IDENTIFIER.

SNMP Operation

Get/Set → Types of SNMP traffic that a SNMP manager uses to either GET SNMP information from the agents or SET SNMP config commands.

Trap → Agents send traps traffic to alert the manager based on the fact that an alarm just happened. This allow the manager to take action and send traffic to the corresponding SNMP devices. The Manager then send an ACKOWLEDGEMENT to let the agent know that the message was received.

SNMP Versions

Version 2c → Insecure it used an unencrypted passcode set as a string to describe the relationship between the server and the SNMP devices.

Version 3 → Secure. Adds security to SNMP with AUTHENTICATION [username/password] and hashed with SHA/MD5. On top of that it adds ENCRYPTION the entirety of the SNMP traffic.

We will want to deploy V3 for an enterprise set up.

SNMPv3 IOS-XE Structure

Remember that the SNMP AGENTS have OIDs to identify themselves within the SNMP protocol/network. Ideally we want our SNMP manager to have full connection/control over every AGENT within our network or at least a partial control over many different technologies like

- ➔ VLANS
- ➔ Interfaces
- ➔ Virtual Interfaces
- ➔ ……etc

In order to set this control, we do so via a VIEW that describes the scope of access.

Organizations create an ISO tree where the OIDs are placed to identify different levels of hierarchy.

We also create GROUPS and USERS where we unite users into different groups and set up VIEWS to better determine the scope of access. In groups we also set up the SECURITY portion that includes authentication and encryption.

Configure SNMP Parameters

With all the configurations for SNMP, we are going to use snmp-server command first

```
Router(config)#snmp-server
```

80.- Configure Syslog and NTP

Log messages are important messages that are saved by a device. In networking, these log messages are saved in different places like the BUFFER and are delivered to the CONSOLE and have a few formats that are very important:

- ➔ Timestamp
- ➔ Facility [%. Hints what might be going on
- ➔ Severity level (0-7) High-level to low-level
- ➔ Mnemonic level (code/category)
- ➔ Description

Command ***show log***

```
Router#show logging
```

Syslog Servers

A network device has finite space to store messages known as "LOG BUFFER" of usually 4096 Bytes. When it gets full, the device takes the oldest message and discards it.

- ➔ But an ADMIN can change parameters like INCREASE the buffer size to store more messages.
- ➔ To OFFLOAD LOGS which means getting logs out of the device and storing them in a server that runs an application to store and manage logs. This server is also centralized so every device in a network can connect to it and send logs.

Logging Severity Levels

The LOG message has a level of severity. For example, let's say that an Interface went down or a CPU is overheating. Each of these have levels of severity. The severity ranges go from 0 – 7, being extremely severe to 7 the lowest.

7 is reserved to DEBUGGING messages

6 for INFORMATIONAL messages like the

5 for NOTIFICATIONS

4 is for WARNINGS

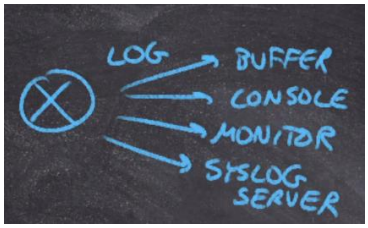
3 is for ERRORS

2 is for CRITICALS

1 is an ALERT

0 is an EMERGENCY

So remembering how a device organizes sending saving messages into a BUFFER, CONSOLE, MONITOR & then to a SYSLOG SERVER

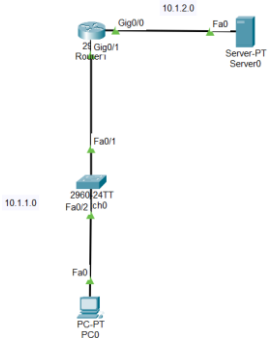


Buffer and Syslog Logging

With these commands we can:

1.- Change the buffer size →

```
Router(config)#logging buffered 8192
```



2.- Configure the Log Server →

```
Router(config)#logging host 10.1.2.100
```

And by entering show logging command again, we can see that we are in fact connected to the server

```
ESM: 0 messages dropped
    Trap logging: level informational, 5 message lines logged
        Logging to 10.1.1.1  (udp port 514,  audit disabled,
```

3.- Change the severity level. Keep in mind that changing the severity level to a set number will only show the severity messages of that number and below not above. Warning meaning 4

```
R6(config)#logging buffered warnings
```

Network Time Protocol (NTP)

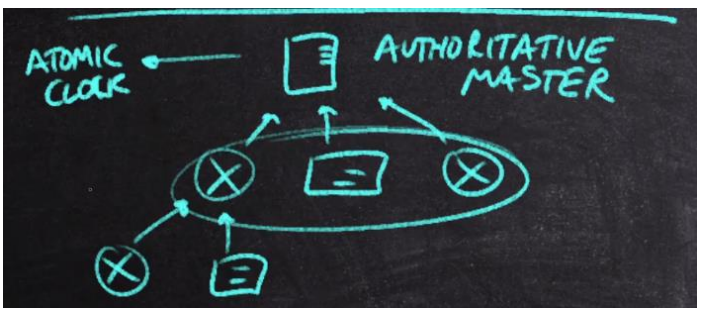
Network protocol that maintains synchronization with time in a network.

We create a hierarchy of

1.- Authoritative Device → Synced to an atomic Clock or other system that guarantees the time

2.- Network devices that point to the A.Device

We clasify these devices in tiers called stratum (1 - 15).



Configure NTP

In order to configure NTP in a network we need to:

1.- Configure a Local Clock with **(config) clock timezone [name] [delta]**:

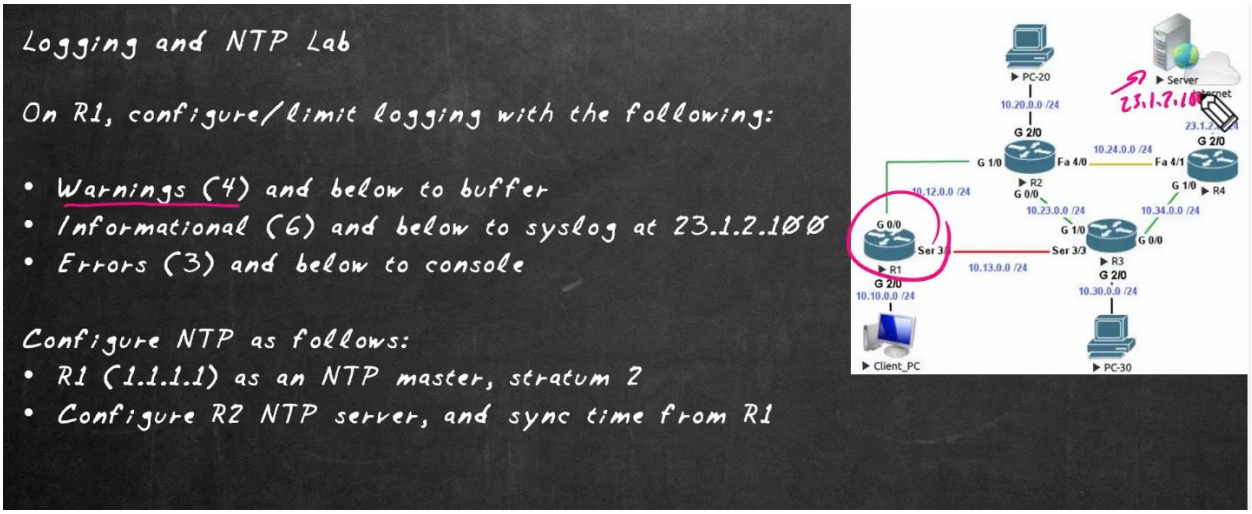
2.- Manually set the clock **clock set**:

3.- Make a NTP master router **ntp master [stratum]**:

4, 5.-

- **ntp server [ip]**: set the local system to sync with an upstream server
- **ntp peer [ip]**: set the local system to mutually sync with a peer

VALIDATION]



1.- Creating the NTP master router with Strat 2 `R1(config)#ntp master 2`

2.- Sync R2 to R1 `R2(config)#ntp server 1.1.1.1`

81.- Configure Cisco Device Management

Configurations remotely using VTY lines inside routers which helps us configure SSH.,

Telnet and SSH

Used to make remote connections.

VTY Lines

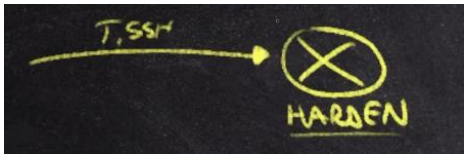
Used to make a virtual connection via SSH. In order to enable SSH or Telnet or even NONE of these, we will want to configure the VTY or RANGE of VTYs that we want SSH/telnet/none to run on:

```
Router(config)#line vty 0 4
```

```
Router(config-line)#transport input none
```

In this case we decided to run NONE of these remote connection protocols.

VTY Security



There are security features in a router we can implement.

Enable command

1.- Secure the enable command: There is a difference between password vs secret. A password is NOT encrypted while a secret is encrypted. So when we connect to a device and set the command 'enable', we will need to provide a password or secret preferably.

2.- LOGIN method to the router itself: How do we validate that the user trying to connect is capable of?

Login to the Router

2.1.- LOGIN command: Use a password configured on the VTY line itself.

2.2.- LOGIN LOCAL: Use a username and Password combo. DONE ON GLOBAL CONFIG

2.3.- NO LOGIN: Simply deny all attempts to login.

3.- ACL: Can restrict logins to specific IP ADDRESSSES. An ACL can be placed into a VTY.

VTY Configuration

Lets see these configurations:

1.- Enabled secret/password:

```
R3(config)#enable secret CBT12345
```

2.2.- LOGIN LOCAL with username and password:

```
R3(config)#username jeff secret cbtcisco
```

Enabling SSH

1.- What VERSION

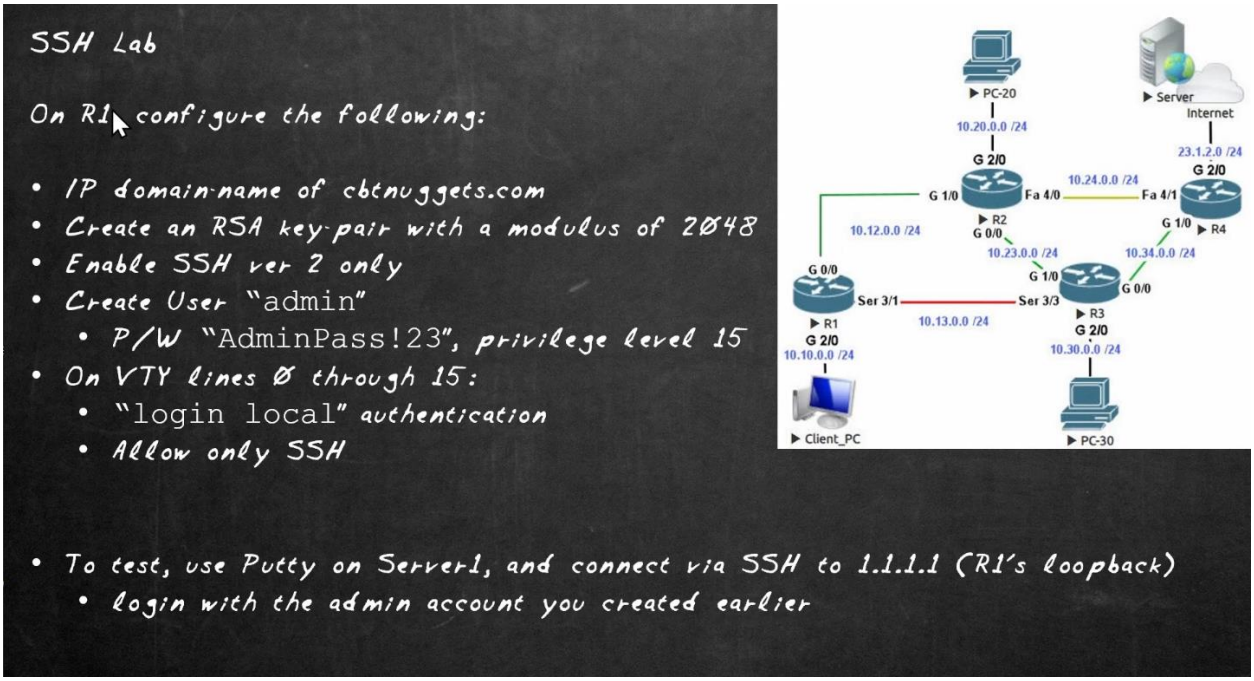
2.- Line: Transport input [all/ssh]

3.- Username/secret ----> In GLOBAL config

4.- KEYS → ip domain-name _____

Crypto key generate RSA modulus 2048

Configuring and Testing SSH & VALIDATION



We will configure R1 to be able to receive SSH connections.

First let's start with the domain name

```
R1(config)#ip domain-name cbtnuggets.com
```

Second the RSA key-pair

```
R1(config)#crypto key generate rsa general-keys modulus 2048
```

Third, let's enable SSHv2

```
R1(config)#ip ssh version 2
```

Fourth, we will create an user 'admin' with privilege 15 and P/W "AdminPass!23"

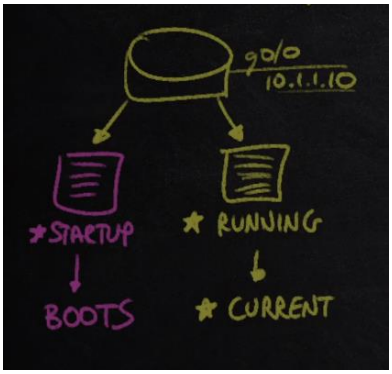
```
R1(config)#username admin privilege 15 secret AdminPass!23
```

Last, we will put a security barrier on every VTY line when someone wants to join the router

```
R1(config)#line vty 0 15
R1(config-line)#login local
```

82.- Configure IOS file Management

IOS Storage System



When we talk about storing data, of most importance is file images on the operating system.

The biggest storage unit is BOOT FLASH or FLASH.

There are 2 types of configurations running.

1.- Startup: Config when the system boots.

2.- Running: Current configuration operating in the system

When a device reloads, we don't keep the running config, it actually gets wiped. Once the system reloads, it will check for the startup config and it will copy it into the running config. So the running config will always change.

The command

Will copy the running config into the startup config.

This is stored on the NVRAM (Nonvolatile).

Managing IOS Files

We can show what's stored on the flash:/ with the dir command. We will see important info like the vlans.

Dir also has other uses like see what's stored on the NVRAM.

FOR EXAMPLE. We can actually copy some files using the copy command

```
Router#dir
Directory of flash0:/

    3  -rw-    33591768          <no date>  c2900-universalk9-mz.SPA.151-4.M4.bin
    2  -rw-       28282          <no date>  sigdef-category.xml
    1  -rw-      227537          <no date>  sigdef-default.xml
```

```
Router#dir nvram
Directory of nvram:/

No files in directory
```

```
Switch#dir flash
Directory of flash:/

    1  -rw-     4670455          <no date>  2960-lanbasek9-mz.150-2.SE4.bin
```

Trivial File Transfer Protocol (TFTP)

Since both the running and startup configurations are saved in FLASH, every time we want to copy a new Image, we want to copy it to FLASH.

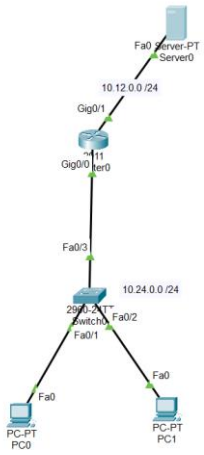
Usually if we want to update the configurations, networks have a NETWORK REPOSITORY that saves files and provides file system services to the entire network.

TFTP exists for this. Trivial means that the connection is not encrypted and It requires no authentication. It uses UDP which means that it does not rely on L4 reliability.

TFTP does check MD5 hashes.

TFTP and MD5 Demo

In our topology, we have a server running TFTP. In there we have a bunch of files that we'll want to copy into router0. In this example we are copying the file `ir800_yocto-1.7.2.tar`.




```
R1#copy tftp: flash:
Address or name of remote host []? 10.12.0.100
Source filename []? ir800_yocto-1.7.2.tar
Destination filename [ir800_yocto-1.7.2.tar]?

Accessing tftp://10.12.0.100/ir800_yocto-1.7.2.tar...
Loading ir800_yocto-1.7.2.tar from 10.12.0.100:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 2877440 bytes]
```

Now the file is in dir: on R0

```
R1#dir
Directory of flash0:/

    3  -rw-    33591768          <no date>  c2900-universalk9-mz.SPA.151-4.M4.bin
    4  -rw-     2877440          <no date>  ir800_yocto-1.7.2.tar
    2  -rw-       28282          <no date>  sigdef-category.xml
    1  -rw-      227537          <no date>  sigdef-default.xml
```

We want to VERIFY the file integrity with verify /md5

FTP, SFTP, and SCP

FTP is TCP-based that uses ports 21 for control and 20 for data

It is more complex and scalable than TFTP. It has the ability to create users, permissions, and manages network infrastructure better.

It uses USER/PASSWORDS which creates authentication.

IT DOES NOT USES ENCRYPTION

The file transfer protocol that does uses ENCRYPTION IS SFTP VIA SHH

To demonstrate FTP, we will use the same server and create a Username/Password

	Username	Password	Permission
1	cisco	cisco	RWDNL
2	admin	123456	RWDNL

83.- Describe Network Quality of Service (QoS)

Some traffic needs to be prioritized. For example, VoIP, Video needs low latency for a smooth experience. Other critical applications like Customer research or data center are critical might not need as low latency as the previously mentioned.

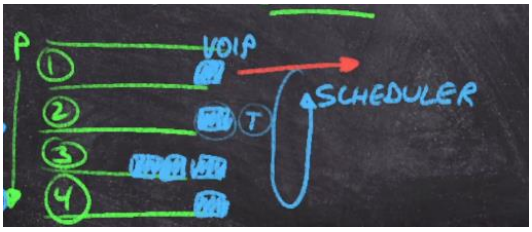
Quality of Service QoS is designed to fix the problem that we have. By default, networks TREAT ALL TRAFFIC THE SAME which will cause some major issues. We need to evolve the way we think about network prioritization.

For example, if we have to drop 2 packets between a TCP connection of an internet download and a VoIP continuous UDP connection, QoS can help us decide which packet to drop.

Queuing and Scheduling

Let's set up a problem.

We have a series of packets and the last one is a VoIP packet. FIFO stands for First In First Out, meaning that if a router receives such series of packets, it will forward the packets at the same order they came in.



QoS can provide prioritization for certain traffic so a network device can send specific packets first. Devices can create multiple QUEUES that will help organize these packets and will provide prioritization.

When a prioritized packet gets sent, a SCHEDULER will look at the queues and decide which packet to send via the PRIORITY.

Now, we are no longer running FIFO and instead started using a more advanced way of prioritizing packets. So even if a prioritized packet arrived after a lower prioritized one, the scheduler will know what to do.

Classification and Marking

We can classify traffic by TAGS [MARK]. A mark are bits which will accompany packets and will determine it's priority. This is part of a model called DiffServ where we have a number of devices and we do not make any reservations; instead each devices will simply send packets without worrying about what the next router will do.

Instead, we rely on the PER-HOP BEHAVIOUR where we will try to create an end-to-end QoS policy. Ex. Give VoIP ---> 10% of the BW.

For this, we create a TAG and set a BEHAVIOUR to it [drop it for example]. And this Behavior will be deployed into the network devices. QoS will tag the frame right when it comes in.

L2 and L3 Marking

So where to put these TAGS?

1.- At L2 → Known as CLASS OF SERVICE CoS. It is a FIELD that goes into the ETH header.

Of 3 bits (0 - 7)

Ø : Default
5 = VoIP
6 | 7 = Network Control

Some TAGS do not belong in the ETH header. Like the VLAN tag that will be added to it. CoS bits will go along with the TRUNK tag under the standard 802.1Q with the number 802.1P.

This means that if we don't have a trunk link, we don't have a CoS bit.

2.- At L3 IP precedence → Inside IP headers L3. Intended to provide End-to-End tagging different to L2 tagging since this last requires trunking to work.

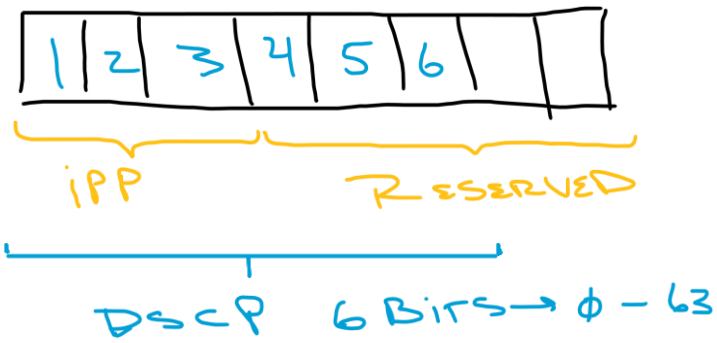
Also 3 bits long. [000]

So why not use always L3 tagging? Because L2 is helpful with L2 devices still that do not

DSCP Tagging

IP precedence at L3 is considered old. We need more bits.

When Type of Service field was invented with 8 bits, 3 were reserved for IP precedence. This became known as Differentiated Services Code Point (DSCP). The industry does not really need the 64 tagging values, but it gives us the ability to create different PER HOB BEHAVIOURS.



The PHB is defined by the following behaviours:

- ➔ Expedited FwD: VoIP value of 46

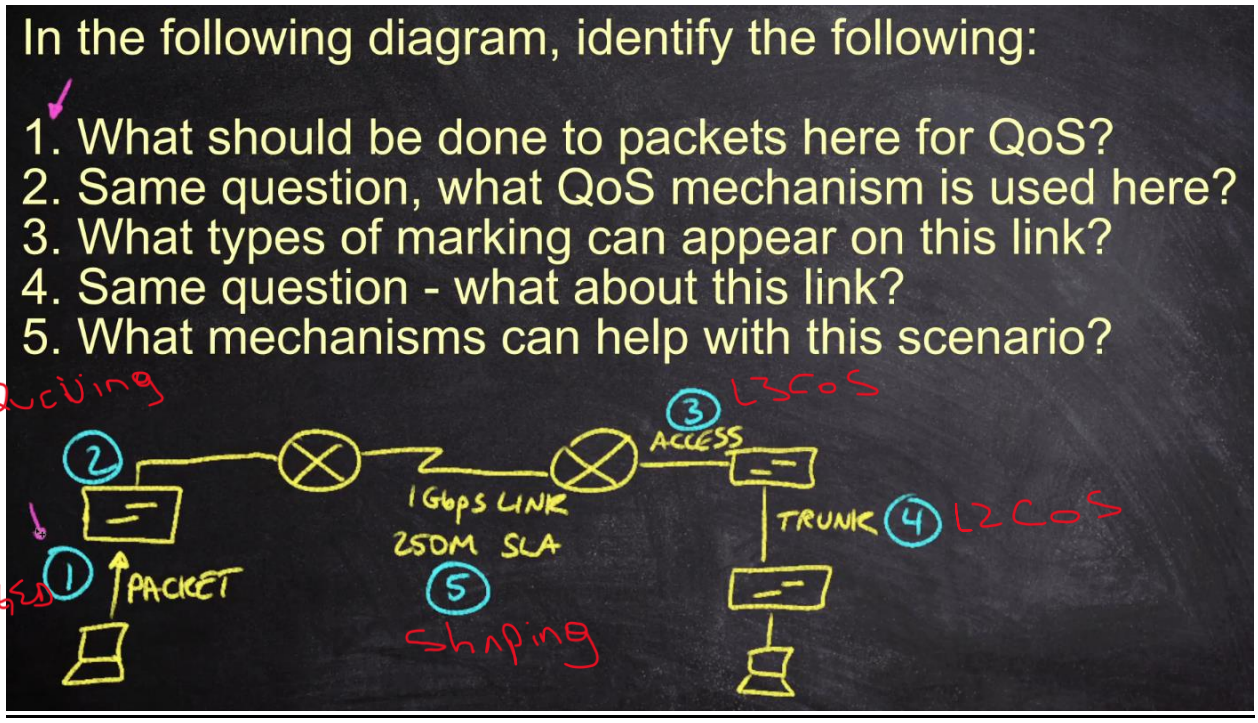
➔ Assured FwD: Many values that are defined by AFXY [Where XY represent other values]
- Ex. AF12, AF33

Shaping and Policing

Traffic shaping is a type of QoS that makes traffic 'flow' at a lower speed that the SLA (SERVICE LEVEL AGREEMENT) states.

Policing is used on ingress to drop any traffic that exceeds an SLA, with an option to allow for bursting traffic

VALIDATION



84.- Describe Network Security Fundamentals

Logical and Physical Security

A lot of secrutiy depends on our network architecture. Like Firewalls, wireless devices, we need to be able to enable SSH, ACLs.

There are LOGICAL configurations and PHYSICAL security.

PHYSICAL security represents access to the devices. For example, if a threat actor gets into a console he gets access to the PASSW Recovery, System config, backdoor which is a 'gate' that contains a user/pass for the threat actor.

What are some methods to protect these physical threats? → Locked doors → Locked racks → Cameras → Security guards.

Password Policies and MFA

Let's talk about DICTIONARY ATTACKS which are basically guesses done on a password. This comes hand in hand with educated guesses with important and known information on the target like birthdays, pet names, etc.

There is also a BRUTE FORCE attack which is an uneducated guess. These take common words, combinations, etc.

So a vulnerable password is a target to these attacks.

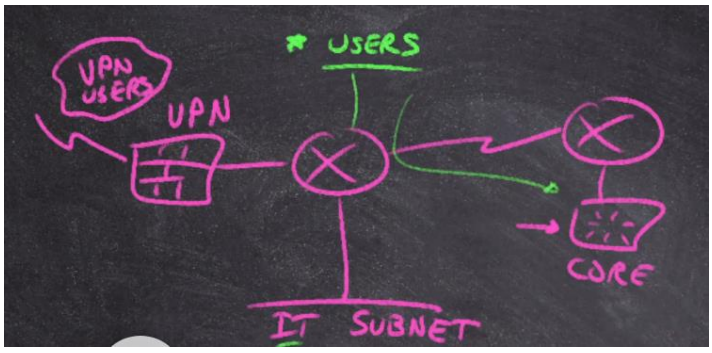
A lot of ORGANIZATIONS are moving to a MFA or 2FA. Which is the process that helps authenticate users with via different methods like a:

- ➔ One time password that expire quickly via a specific app.
- ➔ Certificate which is a FILE that resides on a device.
- ➔ Biometrics

Network Access Control (NAC)

Via ACL we can track subnets. Logins are linked to the IT SUBNETS which gives us the chance of seeing users trying to log in from a subnet they are not supposed to be in. Example in the image.

We can also see the LOCATION of these unusual connections, if using a VPN from another country, it is a good indication that something wrong is going on.



User Compliance

When we create a security policy we have to train our users to follow them and enforce compliance.

Common Network Attacks

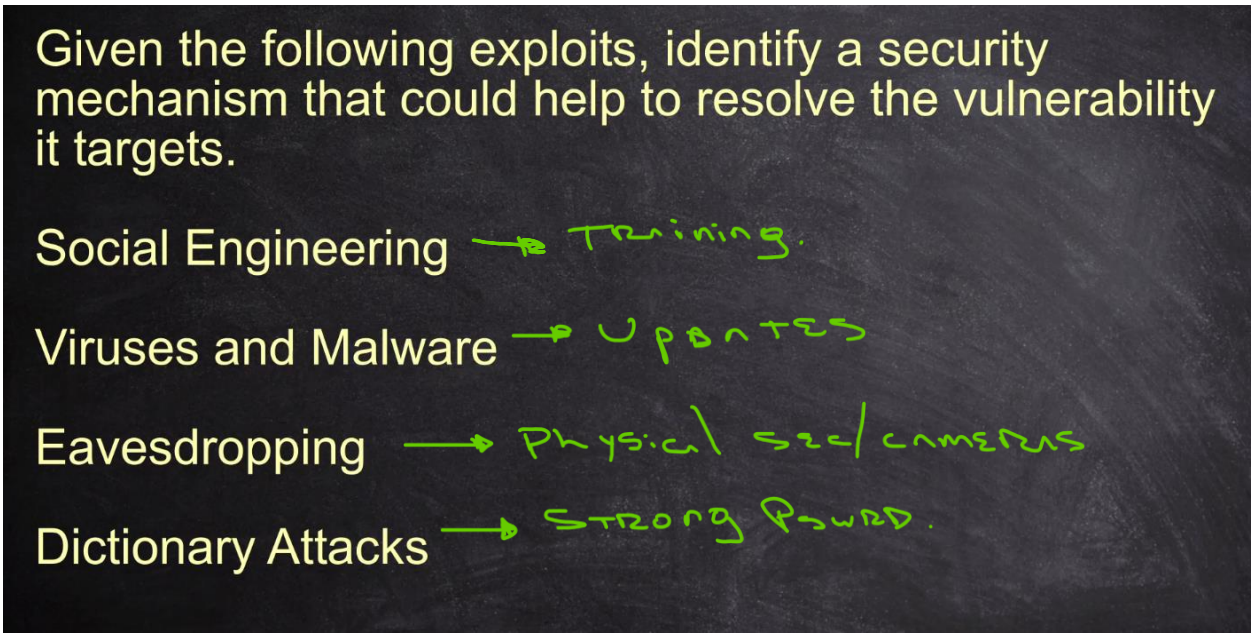
1.- Social engineering → Manipulation of human behavior.

2.- Spoofing → Insert yourself as a source address and pretending to be someone else for example being the S.MAC ADDRESS or the S.IP ADDRESS.

3.- DoS → Take down a service via flooding the service with a lot of traffic and overwhelming a device.

4.- Man in the middle → One of the most effective attacks. We are sending flow traffic to the threat actor and can decrypt the messages as well as change the data.

VALIDATION



85.- Create Standard IPv4 ACLs

Access Control List Overview

Remember that an Access Control List is a list that permit or denies statements. Ex. Deny or accept certain type of traffic.

We create these statement via IP addresses. On the Image we can see that Client PC wants to send traffic over to the server. We could create a 'bouncer' on a specific interface to permit or deny such access. We can also especify particular protocols to be deny or accepted in or out the interfaces. Here are several examples of what we can make our lists with:

- ➔ Accept/deny ANY type of traffic
- ➔ Accept/deny a specific traffic destined to a particular host
- ➔ Accept/deny a particular protocol
- ➔ Accept/deny a specific subnet and add a destination address

Routers will go through each and every one of these lists created to make a decision.

If a traffic DOES NOT show up on any of the created ACLs, the router will literally drop it.

ACLs can be used for:

1.- Filtering →

2.- NAT or PAT to help us identify who we want to be translated.

3.- QoS →

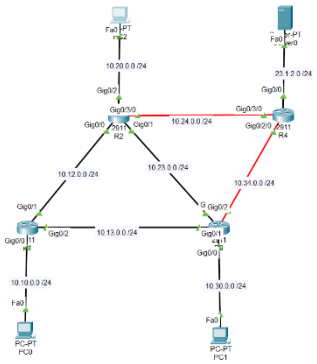
4.- Routing Protocols.

Capabilities of Standard vs Extended ACLs

STANDARD	EXTENDED
Permit/deny Source IP info only	Permit/deny Source & Destination & L4 ports
	

Creating a Standard Numbered ACL

Lets create a standard ACL first [1 - 99] with only a SOURCE IP ADDRESS



In our topology we are going to create a Standard Numbered ACL on g0/2 on R3. Here is the ACL

1.- Permit PC-0 on 10.10.0.0 /24 subnet

2.- Deny traffic that comes from 10.20.0.0 /19 [Even though it is a /24 subnet]

3.- Permit 10.0.0.0 /8 [The entirety of the network]

4.- NOT REQUIRED but Deny any log

Intended for traffic not intended to be denied will get denied.

NOTE → Keep in mind that the more specific conditionals on our list will want to go first and the broader ones last.

INSTRUCTIONS

1.- CHECK IF THERE ARE ANY ACLs first

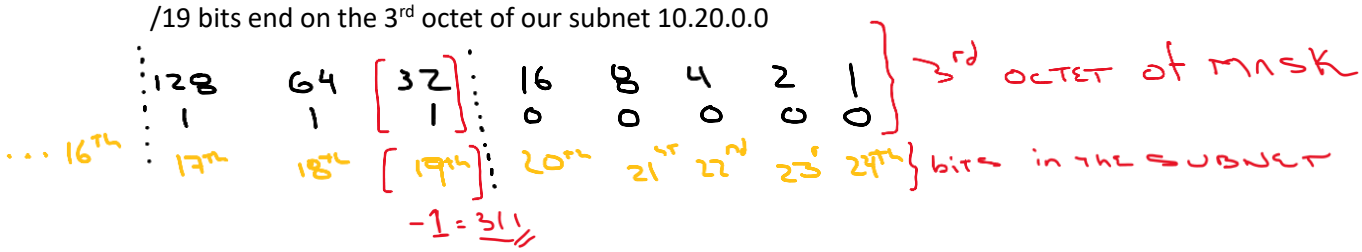
```
R3#show access-lists
R3#
```

2.- Create the first conditional to our Standard ACL

```
R3(config)#access-list 1 permit 10.10.0.2 0.0.0.0
```

3.- Create the second conditional. In this entry, we will have to play around with the wildcard bits since it is a /19. We will use a 'cheat' called block size.

/19 bits end on the 3rd octet of our subnet 10.20.0.0



The trick is to identify where the /19 lands on the octet of the mask and SUBSTRACT 1 to tell the WILDCARD BIT that number. It would mean that we do not care for the last 5 bits on that octet but we care about the first 3. It would look like this:

```
R3(config)#access-list 1 deny 10.20.0.0 0.0.31.255
```

4.- `R3(config)#access-list 1 permit 10.0.0.0 0.255.255.255`

```
R3#show access-lists
Standard IP access list 1
    10 permit host 10.10.0.2
    20 deny 10.20.0.0 0.0.31.255
    30 permit 10.0.0.0 0.255.255.255
```

Exercises

mask of /18 → 3rd octet → 0.0.63.255

10.0.16.0 /22 → 3rd octet → 0.0.3.255

172.16.32.0 255.255.240.0 → 3rd octet ⇒ .11110000. ... → 0.0.15.255

128 64 32 16

192.168.44.32 255.255.255.240 → 4th octet → 1 1 1 1 0 0 0 0

128 64 32 16 8 4 2 1

→ 0.0.0.15

Applying and Testing a Standard ACL

Now we have to APPLY THE ACL TO A INTERFACE (g0/2). We will need to especify for IN-bound or OUT-bound.

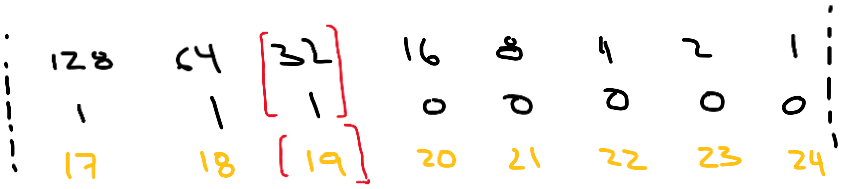
```
R3(config-if)#ip access-group 1 out
```

If you want to see what access lists are assigned to each interface, we use the "*show ip int g0/2*"

```
Outgoing access list is 1
Inbound  access list is not set
```

Let's verify each conditional:

1.- Permit traffic from 10.10.0.2 PC0



2.- Remember that 10.20.0.0 /19

The range for the first subnet is

10.20.0.0 --- 10.20.31.255

```
C:\>ping 23.1.2.100

Pinging 23.1.2.100 with 32 bytes of data:

Reply from 10.23.0.3: Destination host unreachable.
Reply from 10.23.0.3: Destination host unreachable.
Reply from 10.23.0.3: Destination host unreachable.
Reply from 10.23.0.3: Destination host unreachable.
```

Standard Named ACL

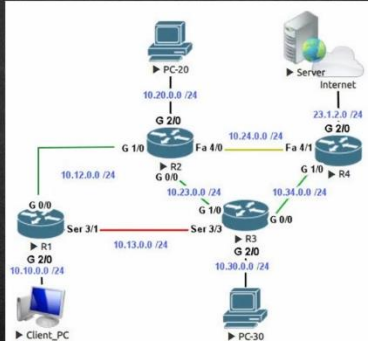
Instead of numbered, we can provide a name to our ACL.

```
R3(config)#ip access-list standard Our-Standard-ACL
R3(config-std-nacl)#per
R3(config-std-nacl)#permit ho
R3(config-std-nacl)#permit host 10.10.0.50
R3(config-std-nacl)#deny 10.20.0.0 0.0.31.255
R3(config-std-nacl)#permit 10.0.0.0 0.255.255.255
R3(config-std-nacl)#deny any lo
R3(config-std-nacl)#deny any log
```

VALIDATION

Creating Standard ACLs Lab

- Standard numbered ACL 1 on R3
- For testing, ping to 4.4.4.4
  - Permits traffic from 10.10.0.50 /32 (test from Client-NUG)
  - Permits traffic from 10.20.0.0 /19 (test from PC-20)
  - Permits traffic from 10.30.0.0 /16 (test from PC-30)
  - Logs all other Denied Traffic (test from R1)
  - Apply outbound on Gig 0/0
  - Test and Verify
- Bonus Points:
  - Reset lab, repeat, but use a named standard ACL



1.- Permits traffic from that particular client

```
R3(config)#access-list 1 permit 10.10.0.50
```

2.- 3rd OCTET

	128	64	32	16	8	4	2	1
	1	1	1	0	0	0	0	0
/19	17	18	19	20	21	22	23	24

→ 10.20.0.0 => 10.20.31.255

Wild card → 0.0.31.255

```
R3(config)#access-list 1 permit 10.20.0.0 0.0.31.255
```

3.- 2nd octet

	128	64	32	16	8	4	2	1
	1	1	1	1	1	1	1	1
/16	9	10	11	12	13	14	15	16

```
R3(config)#access-list 1 permit 10.30.0.0 0.0.255.255
```

→ 10.30.0.0 => 10.30.255.255

Wild card → 0.0.255.255

```
R3(config-if)#ip access-group 1 out

R3#show access-lists
Standard IP access list 1
    10 permit host 10.10.0.50
    20 permit 10.20.0.0 0.0.31.255
    30 permit 10.30.0.0 0.0.255.255
```

85.- Create Extended IPv4 ACLs

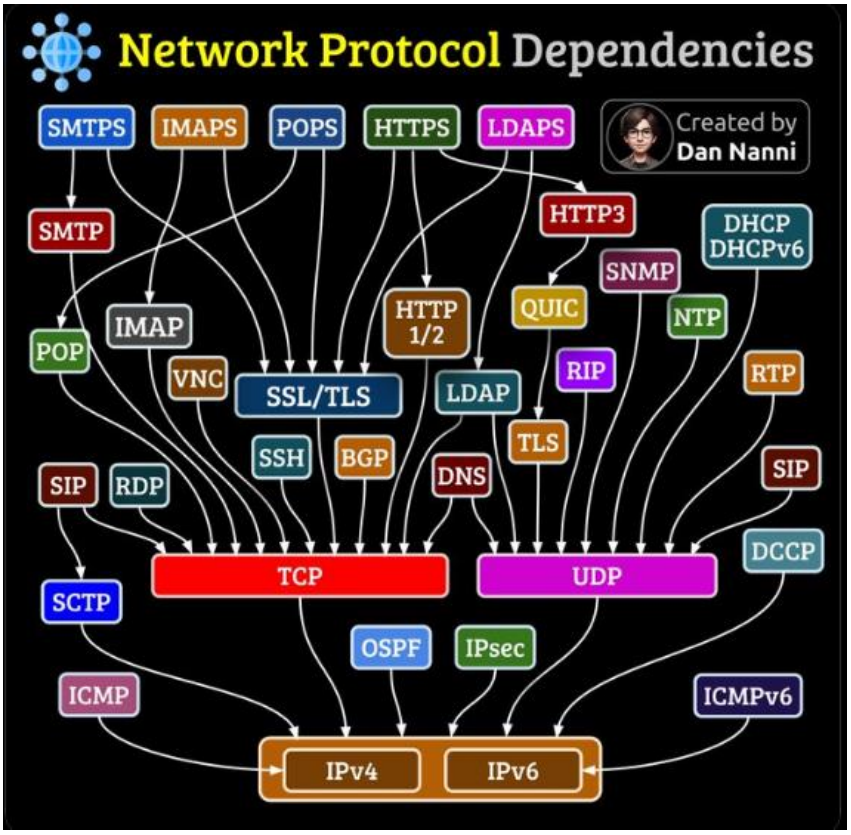
Remember that standard ACLs only create lists based on IP addresses.

The EXTENDED RANGE is #100 – 199 or we can NAME them

Remember that they are also processed from the top down.

We will practice in our topology an E.ACL on R4 at g0/0. These will be the conditionals.

1) Deny ICMP from 10.10.0.48 /29 to 23.1.2.100
2) Permit ICMP from any address to 23.1.2.0 /30
3) Permit HTTP from 10.10.0.0/24 to 23.1.2.0 /24
4) Permit DNS traffic from 10.0.0.0 /8 to 23.1.2.100
5) Permit all IP traffic from 1.1.1.1 to 23.1.2.100

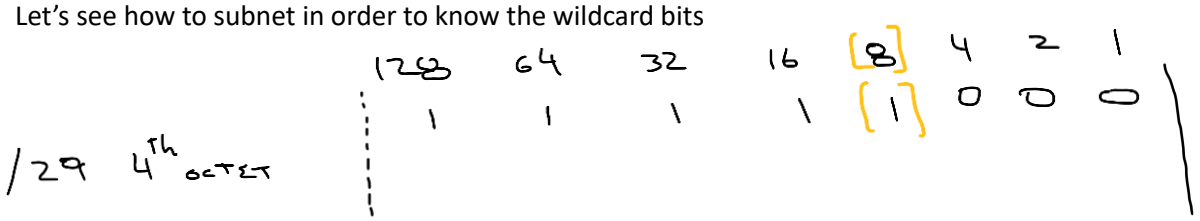


Create an Extended Numbered ACL

1st 2nd 3rd 4th
/8 /16 /24 /32

1) Deny ICMP from 10.10.0.48 /29 to 23.1.2.100

Let's see how to subnet in order to know the wildcard bits

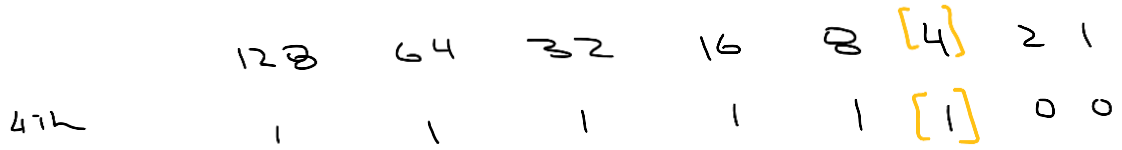


WC → 10.10.0.48 0.0.0.7

Range → 10.10.0.56

```
Router(config)#access-list 100 deny icmp 10.10.0.48 0.0.0.7 host 23.1.2.100
```

2) Permit ICMP from any address to 23.1.2.0 /30

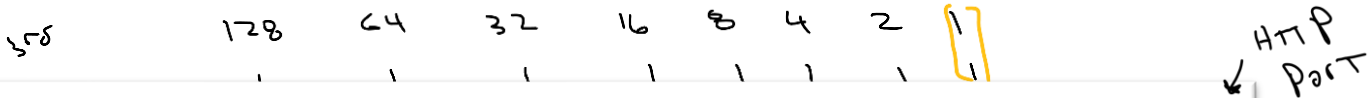


WC → 23.1.2.0 0.0.0.3

Range 23.1.2.3

```
Router(config)#access-list 100 permit icmp any 23.1.2.0 0.0.0.3
```

3) Permit HTTP from 10.10.0.0/24 to 23.1.2.0 /24




```
Router(config)#access-list 100 permit tcp 10.10.0.0 0.0.0.255 23.1.2.0 0.0.0.255 eq 80
```

23.1.2.0 0.0.0.255

4) Permit DNS traffic from 10.0.0.0 /8 to 23.1.2.100

```
Router(config)#access-list 100 permit udp 10.0.0.0 0.255.255.255 host 23.1.2.100 eq 53
```

5) Permit all IP traffic from 1.1.1.1 to 23.1.2.100

```
Router(config)#access-list 100 permit ip 1.1.1.1 0.0.0.0 host 23.1.2.100
```

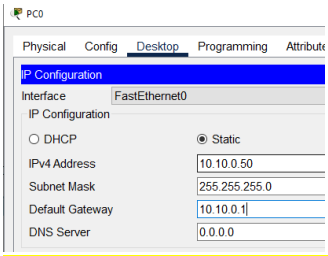
Apply and Test the Numbered ACL

Let's apply it to our G0/0 on R4

```
R4(config-if)#ip access-group 100 out
```

Let's check our conditionals

1.- Since the first conditional was to deny ICMP pings from 10.10.0.48 /29 our PC1 should not be able to ping.

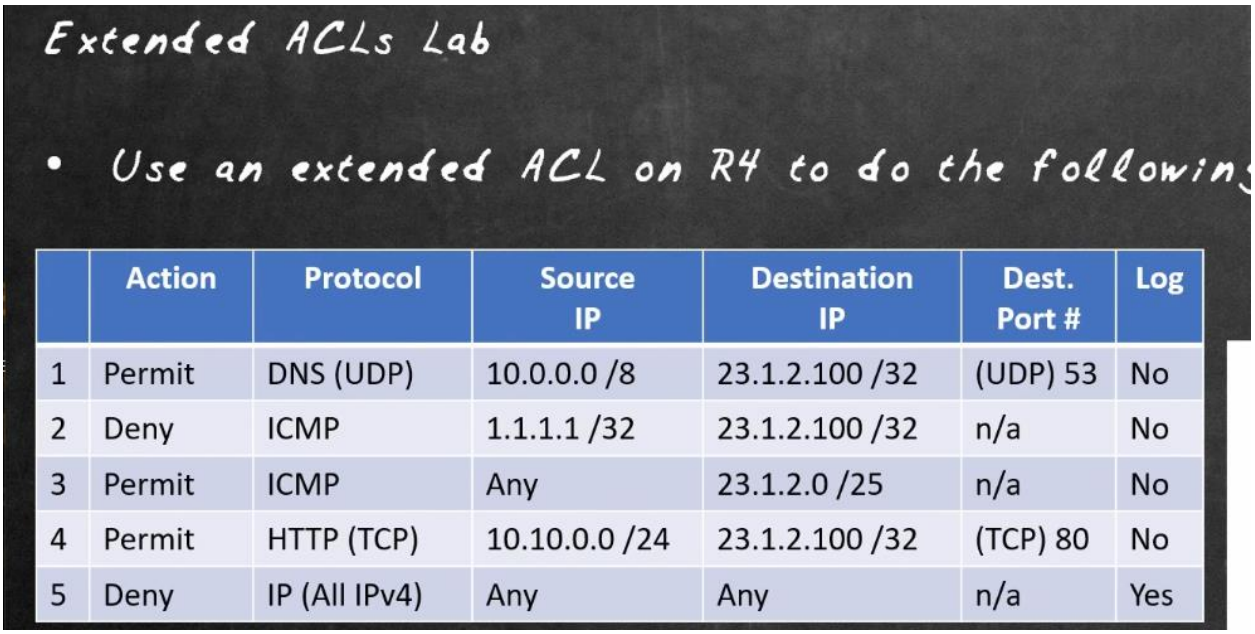



```
C:\>ping 23.1.2.100

Pinging 23.1.2.100 with 32 bytes of data:

Reply from 10.34.0.4: Destination host unreachable.
Reply from 10.24.0.4: Destination host unreachable.
```

we can even see the denys on R4

```
R4#show access-lists
Extended IP access list 100
    10 deny icmp 10.10.0.48 0.0.0.7 host 23.1.2.100 (4 match(es))
```

Validation



	Action	Protocol	Source IP	Destination IP	Dest. Port #	Log
1	Permit	DNS (UDP)	10.0.0.0 /8	23.1.2.100 /32	(UDP) 53	No
2	Deny	ICMP	1.1.1.1 /32	23.1.2.100 /32	n/a	No
3	Permit	ICMP	Any	23.1.2.0 /25	n/a	No
4	Permit	HTTP (TCP)	10.10.0.0 /24	23.1.2.100 /32	(TCP) 80	No
5	Deny	IP (All IPv4)	Any	Any	n/a	Yes

1st /8 2nd /16 3rd /24 4th /32

4th /25	[128]	64	32	16	8	4	2	1
	[1]	0	0	0	0	0	0	0

WC → 23.1.2.0 (0.0.0.127)

Range → 23.1.2.127

86.- Use ACLs with NAT and PAT

Let's review PAT and NAT.

In order to be communication between LANs and the internet we need address translation. We will pick a device at the edge of our network to make the translation.

- ➔ SOURCE NAT/PAT: A rule of thumb for this method is by thinking about it as a 1 to 1 translation. In our 10.0.0.0 /8 subnetwork, we will want each device to be routable on the internet with a given address, in our topologies 23.1.2.0 /24
- ➔ Many to 1 mapping PAT translation: We map all devices to 1 address and it is the job of the router at the edge of the network to 'remember' via PORTS which one belongs to which.

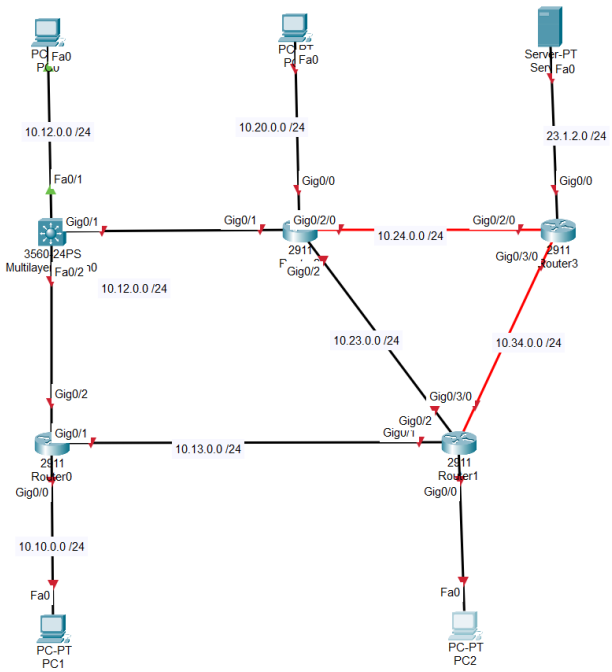
Now we can do this process STATICALLY [static NAT] that involves set up 1 to 1 manually.

Another option is to do it DYNAMICALLY using a POOL of addresses.

ACL, NAT, and PAT Game Plan

Address Translation Game Plan (R4)

- NAT Pool 23.1.2.64 /26
  - allowed for clients on
    - 10.10.0.32 /27
    - 10.12.0.48 /29
- PAT / Overload using G2/0
  - for all other clients on
    - 10.0.0.0 /8
    - except for
      - 10.30.0.51
      - (no NAT/PAT)



If you see on the first list of conditions, 1st 08/ 2nd 16/ 3rd 24/ 4th 32/

N.ID. 10.10.0.32/27 128 64 32 16 8 4 2 1

Range → 10.10.0.63 1 1 1 0 0 0 0 0

[BROADCAST]

10.10.0.33 1 1 1 1 1 0 0 0

↓

62

N.ID 10.12.0.48

Range 10.12.0.55

10.10.0.49 → 54

For the PAT overload on the second condition, we will use this method for every other device coming from the network except of the PC attached to R3./

Configure ACLs for Use with Address Translation

For the ACL we only need standard ACLs since we are not caring for anything else other than source IP addresses.

```
R4(config)#access-list 1 permit 10.10.0.32 0.0.0.31
R4(config)#access-list 1 permit 10.10.0.48 0.0.0.7
```

Let's do now the PAT overload creating another standard ACL 2

```
R4(config)#access-list 2 deny 10.30.0.51 0.0.0.0
R4(config)#access-list 2 permit 10.0.0.0 0.255.255.255
```

Configure IP NAT Interfaces

```
R4#show access-lists
Standard IP access list 1
    10 permit 10.10.0.32 0.0.0.31
    20 permit 10.10.0.48 0.0.0.7
Standard IP access list 2
    10 deny host 10.30.0.51
    20 permit 10.0.0.0 0.255.255.255
```

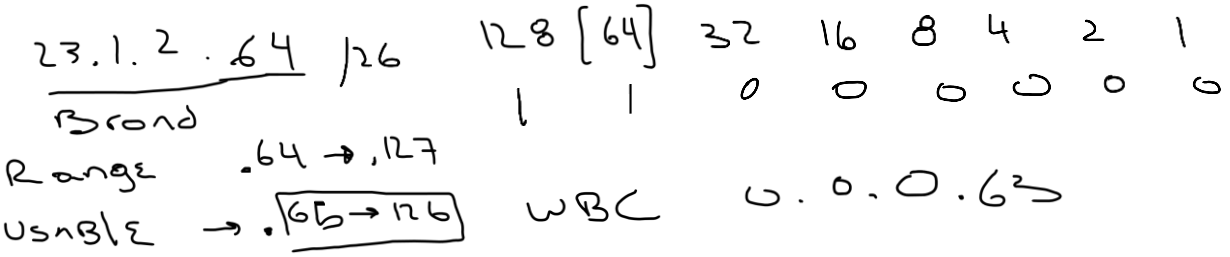
Here is our 2 ACL

These ACLs will be put to use on the interfaces on R4 G0/0 [OUTBOUND] and G0/2/0, g0/1 [INBOUND]

```
R4(config)#int range g0/1, g0/2/0
R4(config-if-range)#ip nat ?
  inside   Inside interface for address translation
  outside  Outside interface for address translation
R4(config-if-range)#ip nat inside
R4(config-if-range)#int g0/0
R4(config-if)#ip nat out
R4(config-if)#ip nat outside
```

Configure NAT POOL

When we look at the POOLs needed for this section we have to do some subnetting.



After knowing our RANGE, we can create the pool with the usable addresses

```
R4(config)#ip nat pool Our-NAT-Pool 23.1.2.65 23.1.2.126 netmask 255.255.255.192
```

Configure NAT Rules

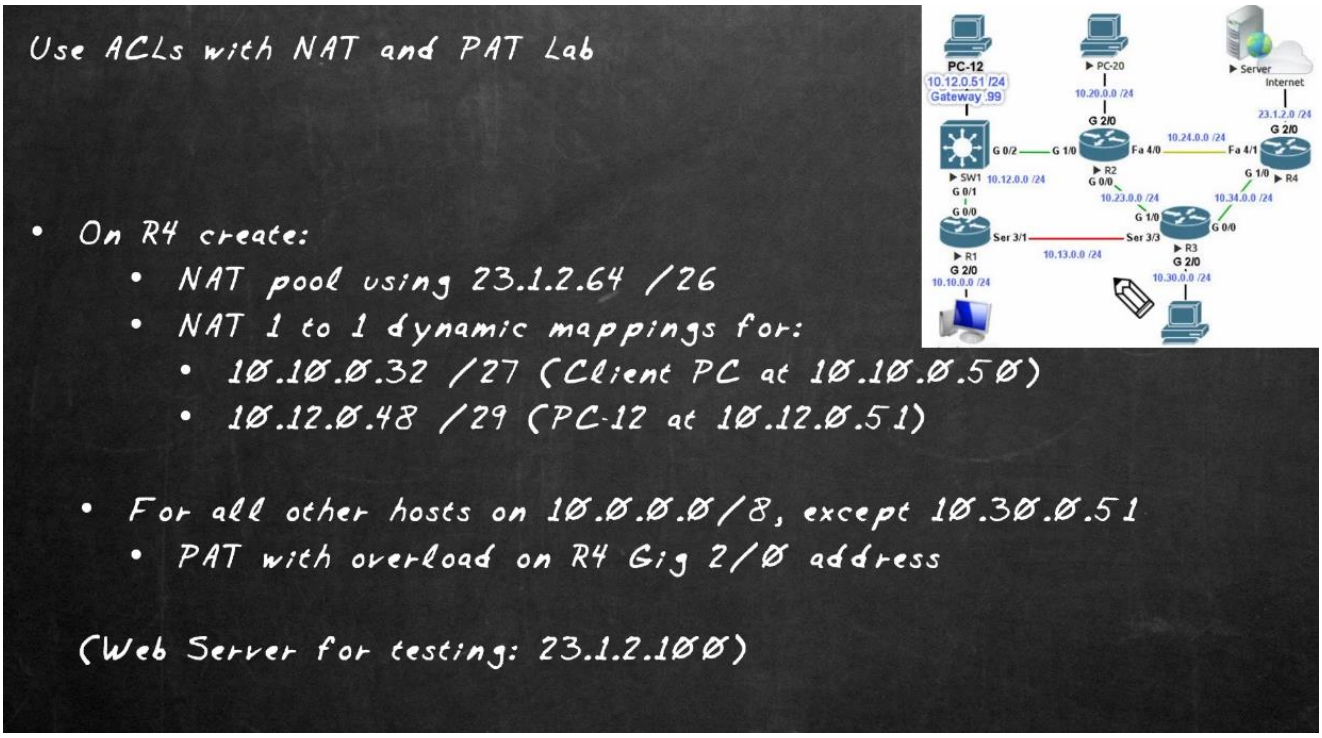
So the whole purpose of this is to tell the R4 that all the traffic coming in from the INBOUND interfaces, I want to do TRANSLATION.

```
R4(config)#ip nat inside source list NAT-ACL pool Our-NAT-Pool
```

Let's configure the OVERLOAD for traffic matching our 2nd ACL

```
R4(config)#ip nat inside source list PAT-ACL interface g0/0 overload
```

```
R4(config)#do show ip nat statistics
Total translations: 0 (0 static, 0 dynamic, 0 extended)
Outside Interfaces: GigabitEthernet0/0
Inside Interfaces: GigabitEthernet0/1 , GigabitEthernet0/2/0
Hits: 0  Misses: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
access-list NAT-ACL pool Our-NAT-Pool refCount 0
 pool Our-NAT-Pool: netmask 255.255.255.192
       start 23.1.2.65 end 23.1.2.126
       type generic, total addresses 62 , allocated 0 (0%), misses 0
```



Validation & Overview

Remember that NAT/PAT are standards that help us 'translate' inner IP local addresses to global addresses routable on the internet.

There are basically 2 ways of translation:

1.- Source NAT → Which gives us the chance to map 1 to 1 addresses from a private IP local address to a global address.

2.- PAT → Type of source NAT that map MANY to 1 global address and you differentiate them via PORTS

Now, you can make both statically and dynamically.

STATIC NAT → Mapping 1 to 1 addresses permanently. Ex, when you need to contact a server

STATIC PAT → Mapping 1 private address and port to a 1 global address and port. Ex, use for services like SSH.

DYNAMIC NAT → Maps 1 private to 1 global [from a POOL] of addresses.

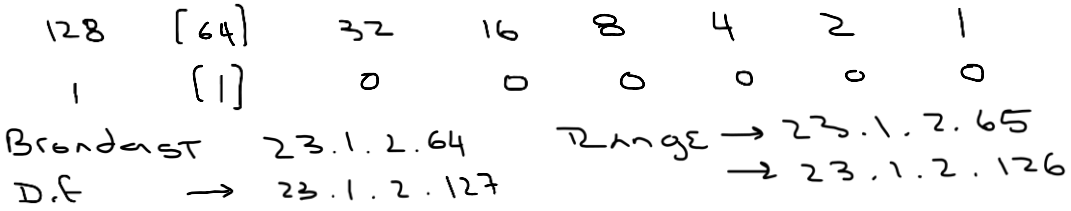
DYNAMIC PAT → Many PRIVATE addresses use a single global IP address, differentiated via ports.

So, ACLs are used to create conditionals that support rules on a network to restrict or allow traffic.

ACLs are great for PAT/NAT implementation because they specify which INTERNAL IP ADDRESS SHOULD UNDERGO PAT/NAT

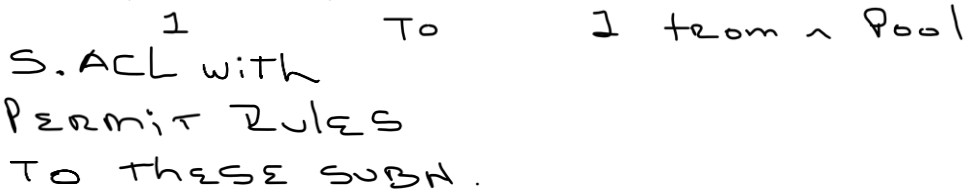
LAB

1.- On R4 create NAT pool using 23.1.2.64 /26



This will be the range of our NAT pool. Remember that NAT is done when mapping 1 private IP to 1 global IP address [1 to 1]. Static NAT happens when you manually do a 1 to 1 mapping. Dynamic NAT does 1 to 1 but from a POOL.

So for the first exercise, we are going to do DYNAMIC NAT 1 to 1 [pool] from 2 internal subnets 10.10.0.32/27 & 10.12.0.48/29 ----------> 23.1.2.65 < 23.1.2.126




```
R4(config)#ip nat pool NAT-Pool 23.1.2.65 23.1.2.126 netmask 255.255.255.192
```

Now we start mapping these 2 subnets to the pool of addresses.

We do this by creating a STANDARD ACL that we are going to map to our POOL

```
Router(config)#ip access-list standard NAT-ACL
Router(config-std-nacl)#permit 10.10.0.32 0.0.0.31
Router(config-std-nacl)#permit 10.12.0.48 0.0.0.7
```

Creating the standard ACL that will get mapped with the NAT-Pool

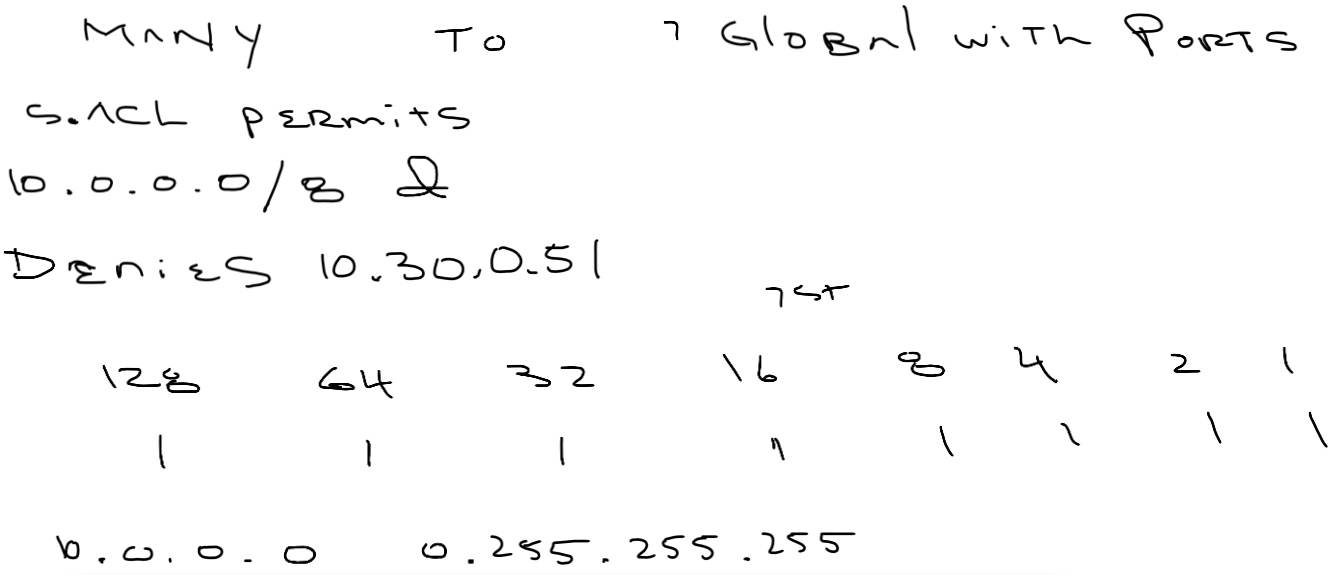
```
Router(config)#ip nat inside source list NAT-ACL pool NAT-Pool
```

Mapping the NAT-ACL where we permit traffic for 10.10.0.32/27 & 10.12.0.48/29 to the NAT-Pool. Therefore doing Dynamic NAT 1 to 1 [From POOL]

3.- For all other hosts on 10.0.0.0 /8 except 10.30.0.51 do PAT overload.

Now the same process happens here except that we will use PAT. Remember that PAT uses MANY TO 1 GLOBAL differentiated via ports.

We are creating another standard ACL that will permit traffic from all other subnet on 10.0.0.0/8 except for 10.30.0.51




```
Router(config)#ip access-list standard PAT-ACL
Router(config-std-nacl)#deny 10.30.0.51
Router(config-std-nacl)#permit 10.0.0.0 0.255.255.255
```

Now, we just have to mapt our S.ACL that permits traffic from all subnets on 10.0.0.0/8 except for 10.30.0.51 to our 1 GLOBAL IP address that will use ports to differentiate between each private host.

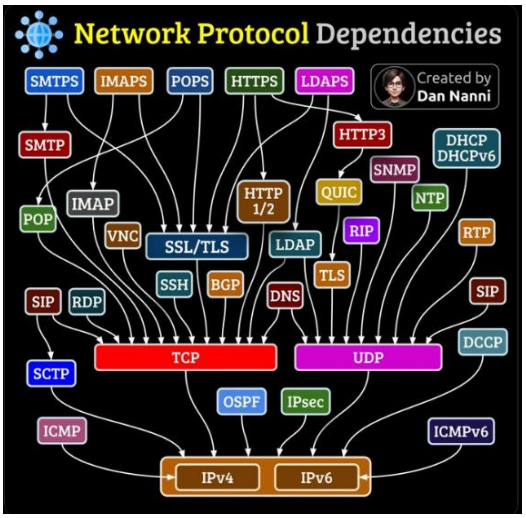
Therefore, doing DYNAMIC PAT MANY TO 1 using ports

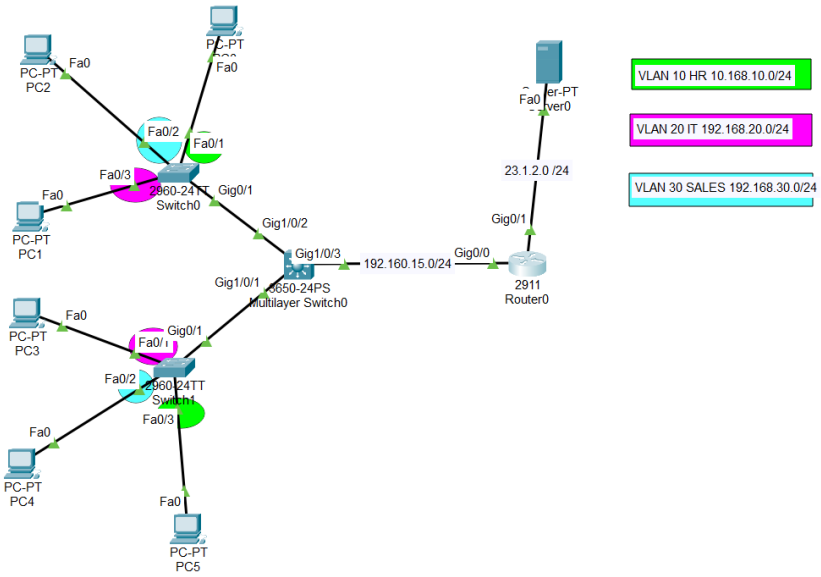
```
Router(config)#ip nat inside source list PAT-ACL interface g0/0 overload
```

Last step is to let the interfaces on R4 know which ones are bound to do inside & outside NAT.\

```
Router(config)#int range g0/3/0, g0/2/0
Router(config-if-range)#ip nat inside
Router(config-if-range)#exit
Router(config)#
Router(config)#int range g0/0
Router(config-if-range)#ip nat outside
```

EXTRA EXERCISE





We are going to practice:

DYNAMIC PAT → MANY [DYNAMIC.ACL] to 1 Global using ports

- ➔ Permit all traffic to access DHCP services [23.1.2.100] according to the following rules. except for PC5.
  - Permit DHCP on 192.168.10.0 /27 to 23.1.2.0 /24✔
  - Permit DHCP on 192.168.20.0 /29 to 23.1.2.0 /24
  - Permit DHCP on 192.168.30.0 /28 to 23.1.2.0 /24
  - DENY DHCP services to PC5 [192.168.10.3] ✔

```
Router(config)#access-list 100 permit udp 192.168.10.0 0.0.0.31 23.1.2.0 0.0.0.255 eq 67
Router(config)#access-list 100 permit udp 192.168.10.0 0.0.0.31 23.1.2.0 0.0.0.255 eq 68
Router(config)#access-list 100 permit udp 192.168.20.0 0.0.0.7 23.1.2.0 0.0.0.255 eq 67
Router(config)#access-list 100 permit udp 192.168.20.0 0.0.0.7 23.1.2.0 0.0.0.255 eq 68
Router(config)#access-list 100 permit udp 192.168.30.0 0.0.0.15 23.1.2.0 0.0.0.255 eq 67
Router(config)#access-list 100 permit udp 192.168.30.0 0.0.0.15 23.1.2.0 0.0.0.255 eq 68
```

- ➔ Allow all traffic to access the internet except for VLAN 10 [192.169.10.0 /24]

```
Router(config)#access-list 100 permit tcp 192.0.0.0 0.255.255.255 host 23.1.2.100 eq 80
Router(config)#access-list 100 deny tcp 192.168.10.0 0.0.0.255 host 23.1.2.100 eq 80
```

NOW remember that DYNAMIC PAT does MANY to 1 GLOBAL depending on ports.

So we map this E.ACL to our port g0/1 on Router0

```
Router(config)#ip nat inside source list 100 interface g0/1 overload
```

DYNAMIC NAT → 1 [STANDARD.ACL] to many [pool]

- ➔ Permit traffic for 192.168.10.0/29
- ➔ Permit traffic for 192.168.20.0/29
- ➔ Permit traffic for 192.168.30./27

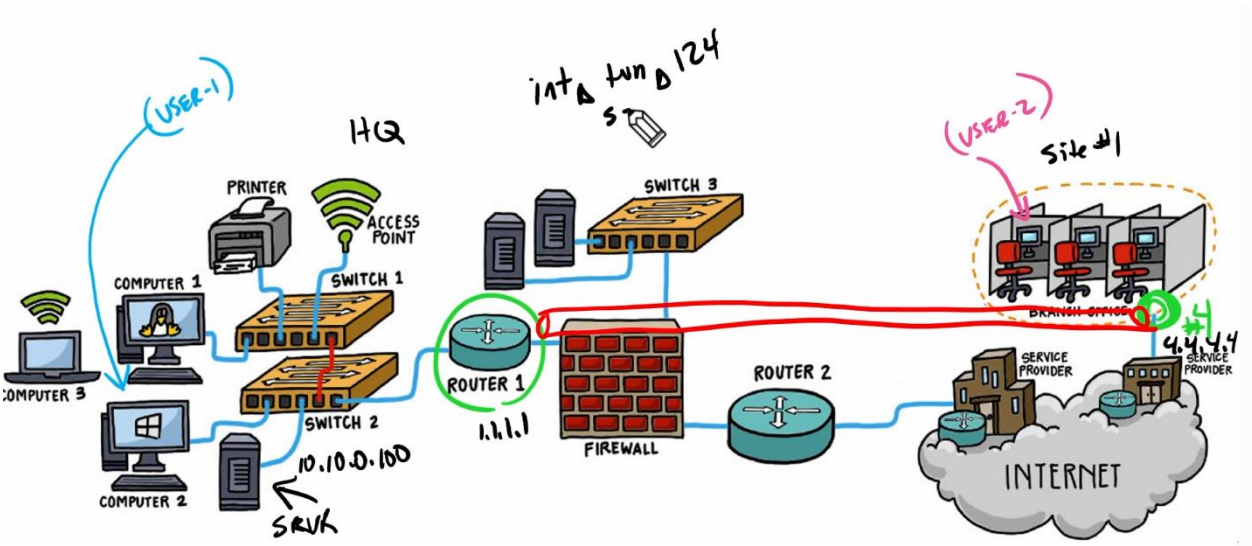
For each, we will set up rules for the ACLs and map them to the pool and the global address for NAT & PAT respectively.

87.- Describe Cisco VPNs

VPN Overview

A virtual NETWORK is usually set up between 2 firewall but for our sake, we will study VPNs [Virtual networks] between 2 routers. They are actually built from Client to Client, so you can set up a tunnel to a Host and Router.

Let's imagine we have 2 routers not connected physically to one another, we can build a VIRTUAL TUNNEL that is logically connecting these routers. Let's see the image as a reference.



R1
→ S: 1.1.1.1
D: 4.4.4.4 172.16.14.0 /24

R2
→ S: 4.4.4.4
D: 1.1.1.1

The point of a VPN is to OVERLAY or "put over" a virtual network over an existing infrastructure.

→ENCRYPTION on vpns

Another important point to consider with VPNs is encryption between both ends.

As mentioned at the beginning, we have

- ➔ Site – site VPNs as shown in the picture
- ➔ Remote Access VPN used for Hosts.

Since the connection is encrypted and uses Keys to decrypt and encrypt, the payload at the application layer won't be visible over the VPN.

There are a lot of different options to create these overlays over an infrastructure

1.- GRE [Generic Routing Encapsulation] → It's the form that you might think of when hearing VPN. Simply if a packet is coming out from a Host at 10.20.0.0 /24 to another host at 10.10.0.0 /24, the VPN packet will show up as if coming from 4.4.4.4 where the VPN is configured.

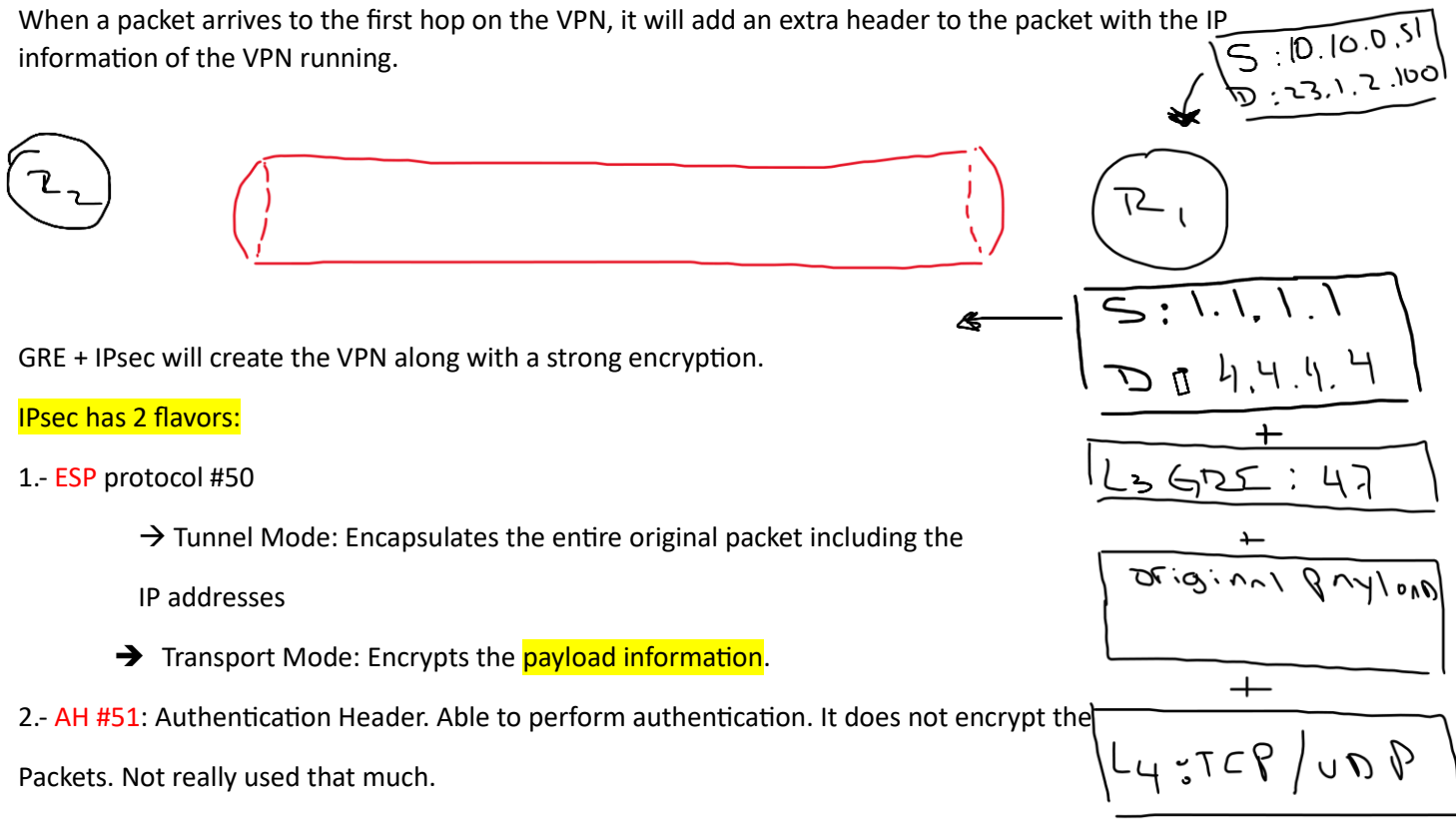
Now, we need an Encryption method:

- ➔ IPsec: Used for Site to site VPNs
- ➔ SSL/TLS used for Remote access VPNs.

IPsec Fundamentals

Important to mention that routing protocols need to be in placed.

When a packet arrives to the first hop on the VPN, it will add an extra header to the packet with the IP information of the VPN running.



GRE + IPsec will create the VPN along with a strong encryption.

IPsec has 2 flavors:

1.- ESP protocol #50

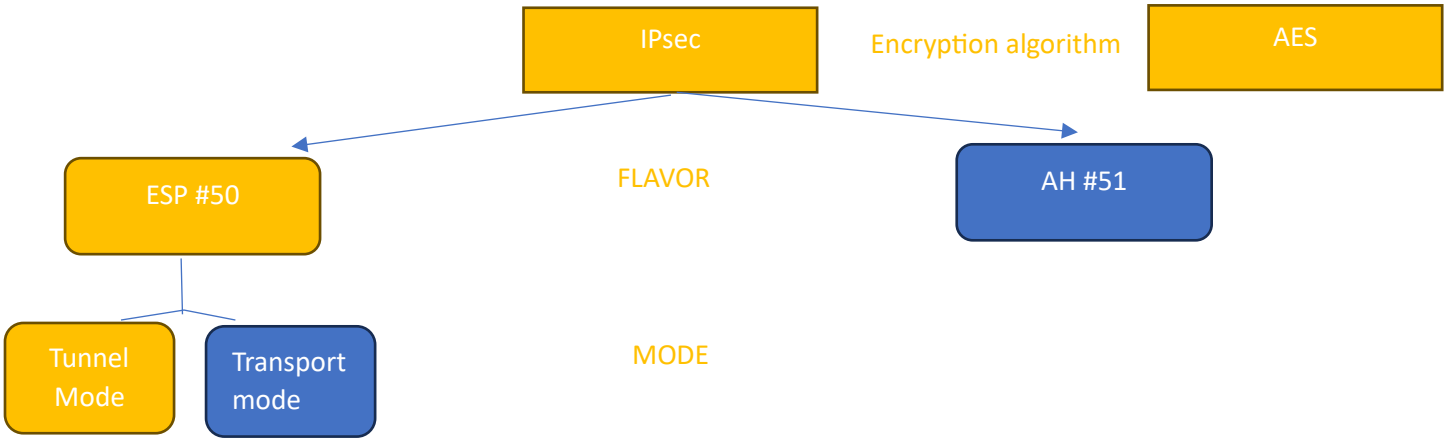
→ Tunnel Mode: Encapsulates the entire original packet including the IP addresses

- ➔ Transport Mode: Encrypts the payload information.

2.- AH #51: Authentication Header. Able to perform authentication. It does not encrypt the Packets. Not really used that much.

We will use ESP

There are multiple encryption algorithms which IPsec uses. For example AES.



IPsec Site to Site VPN Example

GRE Tunnel [Site to site since it is B/W 2 routers]

R1: 124 (S: 1.1.1.1, D: 4.4.4.4)

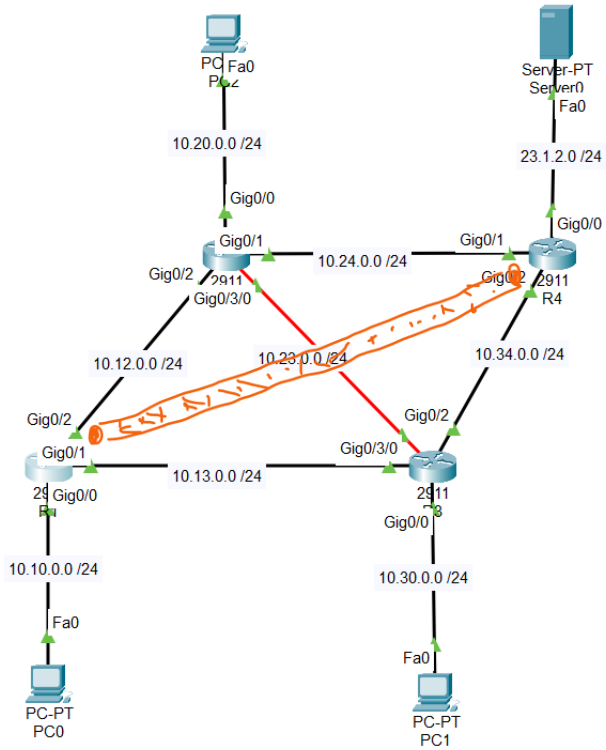
R4: 421 (S: 4.4.4.4, S: 1.1.1.1)

We can get the tunnel interfaces IP address

V.R1: 172.16.14.1 /24

V.R2: 172.16.14.4 /24

On top of this we will use IPsec for Encryption



In order to implement a VPN, we need to:

1.- Create an ACL to enable traffic on the VPN.

2.- Create an ISAKMP policy and key

3.- Create IPsec Transform-set

4.- Create CryptoMap

5.- Apply the Crypto Map to the interface

1.- create the E.ACL (In our case we want to allow traffic from 10.10.0.0 /24 to 23.1.2.0 /24)

```
R1(config)#access-list 100 permit ip 10.10.0.0 0.0.0.255 23.1.2.0 0.0.0.255
R1(config)#access-list 100 permit ip 23.1.2.0 0.0.0.255 10.10.0.0 0.0.0.255
```

Both from the 10.10.0.0 /24 → 23.1.2.0 /24 and vice versa

This has to be done on both ends of the VPN tunnel, in our case R1 & R4

2.- Create an ISAKMP policy and key

```
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#aut
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#group 5
```

On both routers

For the crypto key we will want to put on both ends the respective interface addresses in which the tunnel will begin.

```
R1(config)#crypto isakmp key secretkey address 10.24.0.4

R4(config)#crypto isakmp key secretkey address 10.13.0.1
```

This makes having the same key on both ends of our vpn.

3.-

Using ACLs and Hit Counts to Confirm Tunnel Use

In order to check the traffic being forwarded over the tunnel, we can see how TCP, UDP, ICMP, IP traffic is being encapsulated by ESP traffic (Encrypted traffic), we can add more conditionals to our ACL.

We'll create 2 ACLs named INBOUND [for G0/2] and OUTBOUND [g0/0] to allow ICMP, ESP, HTTP & IP

```
Extended IP access list INBOUND
    10 permit icmp any any
    20 permit esp any any
    30 permit tcp any any eq www
    40 permit ip any any
Extended IP access list OUTBOUND
    10 permit icmp any any
    20 permit esp any any
    30 permit tcp any any eq www
    40 permit ip any any
```

So we should ONLY SEE ESP TRAFFIC BEING forwarded, not HTTP, OR IP OR ICMP

Creating Static Routes to Send Traffic Through the Tunnel

How can we influence the router to route traffic through the tunnel? By influencing the routing table and creating a static route.

And we can set the static rout to use the next hop of 172.16.14.4 to use the tunnel to R4

```
R1(config)#ip route 23.1.2.0 255.255.255.0 172.16.14.4
```

Or you can also set the static route with the egress interface of the tunnel

```
R1(config)#ip route 23.1.2.0 255.255.255.0 tun 124
```

We also would want to do this on R4 for inbound traffic from 23.1.2.100 to our client.

```
R4(config)#ip route 10.10.0.0 255.255.255.0 tunnel 421
```

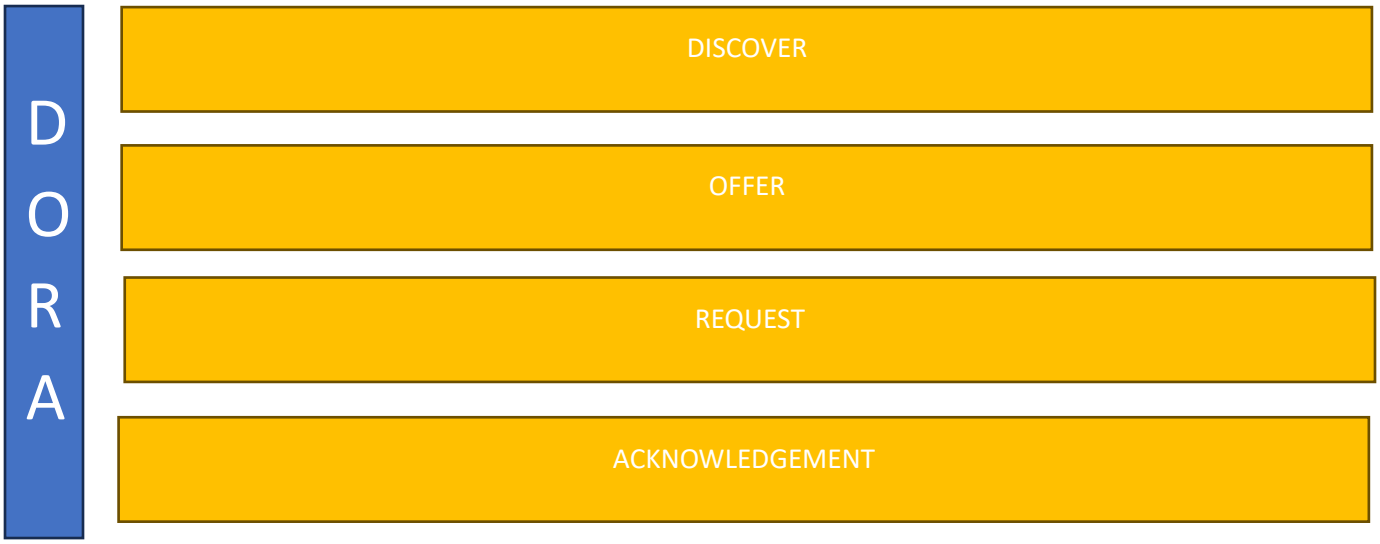
88.- Use Cisco IPv4 DHCP Snooping

We will study how to prevent unwanted DHCP servers from functioning on our servers?

IP DHCP snooping achieves this.

IPv4 DHCP Snooping Overview

Remember DORA from a previous DHCP topic, where:



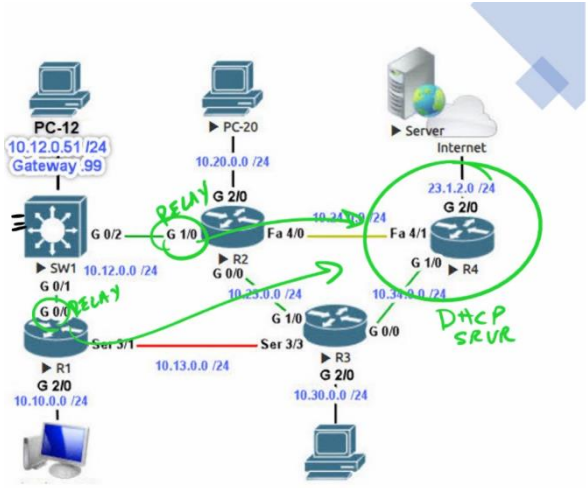
What if a hacker is connected to a network in whatever form where they can compromise a system and act as an DHCP server. Let's imagine that the DHCP client connected to the same network as the rogue instead uses a LOGICAL FIREWALL that basically states "Unless you are a DHCP trusted relay or server, the D.O.R.A process does not go through".

We just enable DHCP Snooping for the network connected to each of our interfaces. In the topology for this lab/exercise it will be VLAN 1.

So, any type of DORA message that try to come in from an UNTRUSTED PORT, will be denied.

In addition, a database will be created of all the DHCP clients that have been addressed to a DHCP address. With this database we can do later security features.

Game Plan for DHCP Services and DHCP Snooping



----> R4 will be our DHCP server

10.12.0.0 /26

G/W 10.12.0.2 .

DHCP Relay -→ R2 G1/0 & R1 G0/1

---> RELAY TO 4.4.4.4

----> On SW1 we will enable DHCP Snooping via VLAN1 where we will trust ports fa0/2 & g0/1

Configure R4 as a DHCP server

Let's enable DHCP services on R4

1.- Creation of pool

```
Router(config)#ip dhcp pool Our-DHCP-Pool
```

2.- Put on POOL & Default-router as main server

10.12.0.0 /26 128 [64] 32 16 8 4 2 1

1 [1] 0 0 0 0 0 0

B.C → 10.12.0.63

Range → 10.12.0.1 < 10.12.0.62

```
Router(dhcp-config)#network 10.12.0.0 255.255.255.192
Router(dhcp-config)#def
Router(dhcp-config)#default-router 10.12.0.2
```

```
R4#show ip dhcp pool

Pool Our-DHCP-Pool :
 Utilization mark (high/low)    : 100 / 0
 Subnet size (first/next)       : 0 / 0
 Total addresses                : 62
 Leased addresses               : 0
 Excluded addresses             : 0
 Pending event                  : none

 1 subnet is currently in the pool
 Current index        IP address range                    Leased/Excluded/Total
 10.12.0.1            10.12.0.1       - 10.12.0.62        0    / 0      / 62
```

Now R4 is a DHCP server with 1 pool that serves 10.12.0.0 /26

We could exclude IP addresses, and we have to do so since 10.12.0.1 & 10.12.0.2 are already being used by R1 & R2

INSTEAD, let's see how the DHCP server on R4 acts if we tell it to apply those 2 addresses

Configure DHCP Relay

On R2 →

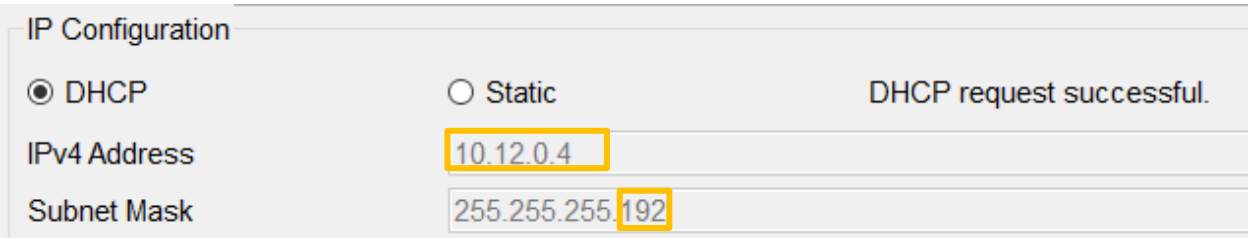
```
R2(config-if)#ip helper-address 4.4.4.4
```

```
R2#show ip interface g0/1
GigabitEthernet0/1 is up, line protocol is up (connected)
  Internet address is 10.12.0.2/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is 4.4.4.4
```

On R1 →

```
R1#show ip int g0/1
GigabitEthernet0/1 is up, line protocol is up (connected)
  Internet address is 10.12.0.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is 4.4.4.4
```

LETS SEE OUR PC ON 10.12.0.0 /24



NOW R4 WILL NOTICE A CONFLICT HAPPENING

```
R4(config-router)#%DHCPD-4-PING_CONFLICT: DHCP address conflict:  server pinged 10.12.0.2.
%DHCPD-4-PING_CONFLICT: DHCP address conflict:  server pinged 10.12.0.1.
```

Enable DHCP Snooping

```
Switch(config)#ip dhcp snooping vlan 1
```

This basically makes that any other traffic that is not on VLAN1, will not be trusted.

We must tell the snooping process to trust both the RELAYS and the DHCP SERVER on R4

Configuring DHCP Snooping Trusted Ports

Let's configure the ports that are trusted

```
Switch(config)#int range g0/1, f0/2
SW-1(config-if-range)#ip dhcp snooping trust
Switch(config-if-range)#ip dhcp snooping limit rate 50
```

Now we have told our SW-1 that g0/1 & f0/2 are trusted ports. So EVERY DHCP TRAFFIC COMING IN AND OUT OF THESE PORTS WILL BE ACCEPTED.

```
SW-1(config)#NO IP dhcp snooping information option
```

Do this command just because you have to. No more reasoning behind it.

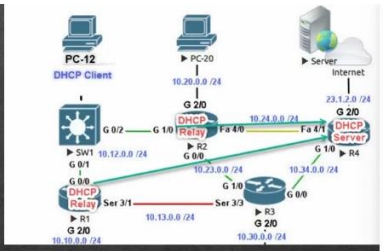
Using the Snooping Database and Agent

We will want to save all of the DHCP information and data into either a local server or our own file system inside the switch. For our purposes, we will do so to our switch.

```
Switch(config)#ip dhcp snooping database flash:/snoopy.txt
```

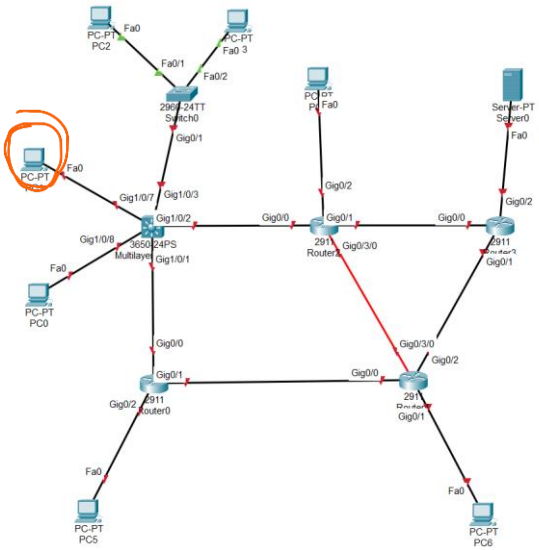
Validation

DHCP Snooping Lab



- Current settings:
  - R4 is a DHCP server, with a pool for 10.12.0.0
  - R1 gig 0/0, R2 gig 1/0 are enabled for DHCP relay to R4
- On SW1 enable DHCP Snooping for VLAN 1
  - Disable info option with: "no ip dhcp snooping information option"
  - Trust the ports to R1 and R2, and rate limit to 50 pps
- Test and verify

89.- Use Cisco l2 Port Security



Let's imagine for a second that the user on the selected computer tries to consume every single one of the IP addresses from the DHCP server.
They could send thousands of D. messages with different SOURCE MAC ADDRESSES for every single message. This is called DHCP STARVATION.

They could even run their own DHCP server that hands out new IP addresses.

We have studied that DHCP snooping solves part of this problem by conditioning DHCP traffic from and to our DHCP relays and servers.

----SWITCHPORT PORT-SECURITY defines how many MAC addresses can learned on a PORT.----

In this case, we are protecting our devices on a L2 level.

If we tell the SW connected to the malicious PC via port security that only 1 MAC address can be learned from that port and traffic violates that conditional, the port shuts down or error disables (we can play around with these rules).

Violation modes

- ➔ Protect: It will follow the port security if the conditional gets violated but does not tell 'anyone'. Will not send any SNMP messages nor save these results to a log. Not good practice if you want to troubleshoot your network.
- ➔ Restrict: It will also follow the conditions applied by the port security but more flexible when it comes to saving the results to a SNMP log or create a HIT counter.
- ➔ Shutdown: It follow the conditions from the port security and they get violated, it will shut down the port.

Por security can be implemented in the following ways

1.- Dynamic Secure (MAC) → When the switch saves MAC addresses as they come

2.- Static Secure (MAC) → Already configured, known Mac address

3.- Sticky Secure (MAC) → Will follow the way your set up a port to behave.

------ Security port can not be implemented if DTP is enabled -------

Configure and then Enable Port Security

```
SW1#show port-security interface f0/1
Port Security              : Disabled
Port Status                : Secure-down
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
```

Game play

We will enable l2 port security to interfaces f0/1-3 on SW1

- ➔ Max MAC addresses: 3
- ➔ Violation Mode: Change from Shutdown to restrict.
- ➔ Aging time: From 0 minutes to 3 minutes

```
SW1(config)#int range f0/1-3

SW1(config-if-range)#switchport port-security maximum 3

SW1(config-if-range)#switchport port-security violation restrict

SW1(config-if-range)#switchport port-security aging time 3
```

Port security is still NOT enabled. REMEMBER TO DISABLE DTP. Then you can enable port security.

```
SW1(config-if-range)#switchport nonegotiate
```

```
SW1(config-if-range)#do show port-security int f0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Restrict
Aging Time                 : 3 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 0
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 0
Last Source Address:Vlan   : 0000.0000.0000:0
Security Violation Count   : 0
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
                (Count)        (Count)         (Count)
---------------------------------------------------------------------
        Fa0/1         3           0                0          Restrict
        Fa0/2         3           0                0          Restrict
        Fa0/3         3           1                0          Restrict
---------------------------------------------------------------------
```

Manually Configuring Secure MAC Addresses

We can apply specific MAC addresses manually.

For PC1,l it's MAC address is 0060.3E6A.8165.

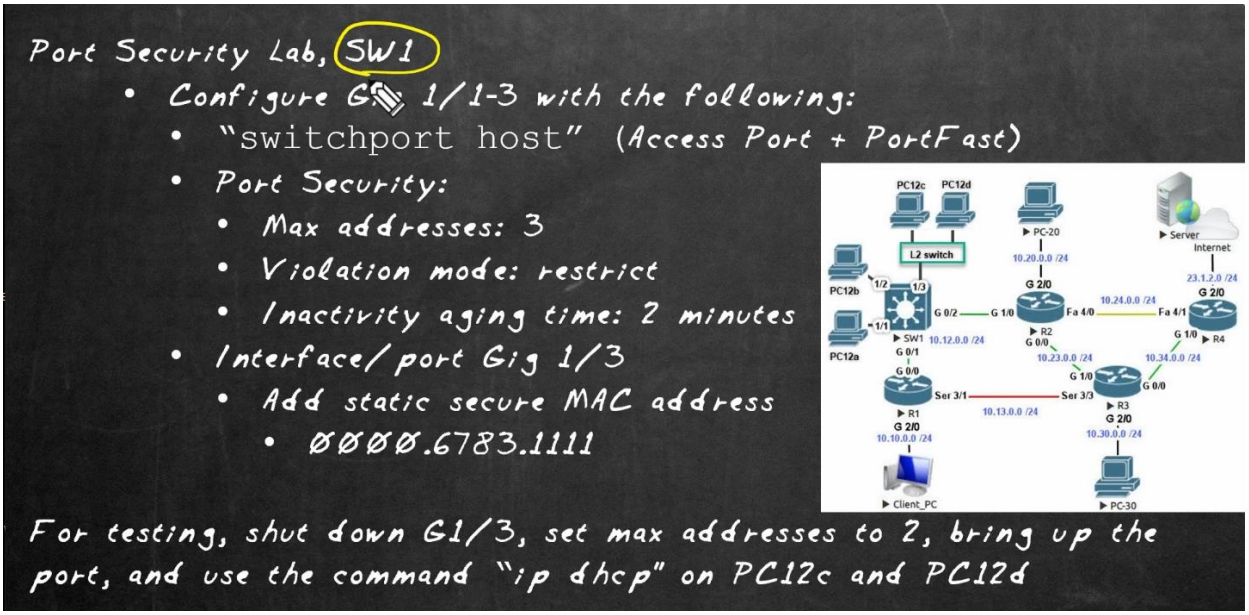
```
SW1(config-if)#switchport port-security mac-address 0060.3E6A.8165
```

This will mean that traffic coming in from this MAC, will be trusted.

When Port-Security Interacts with Other Features

There could be times when we can experience problems with Port Security. For example, if we are using ANY FHRP PROTOCOLS like HSRP and the router associated with an interface that has portsecurity shuts down, the switch has to change it flow of traffic to the secondary router with different port security policies.

Validation




```
SW1(config-if-range)#switchport port-security maximum 3
SW1(config-if-range)#switchport port-security violation restrict
SW1(config-if-range)#switchport port-security aging time 2
SW1(config-if-range)#switchport port-security
SW1(config-if)#switchport port-security mac-address 0000.6783.1111
```

```
SW1#show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                 (Count)       (Count)          (Count)
---------------------------------------------------------------------------
      Gi1/1              3            0                  0         Restrict
      Gi1/2              3            0                  0         Restrict
      Gi1/3              2            2                  3         Restrict
---------------------------------------------------------------------------
```

```
PC-12d> ip dhcp
DDD
```

90.- Understand l2 Security Controls

DAI and using DHCP Snooping Binding Information

There are a couple of L2 security controls that we have implemented like:

- ➔ DHCP Snooping → Where we train the switch where the trusted DHCP servers and relays are at. The switches will create a binding table between L2 MAC addresses / L3 IP addresses.
- ➔ Port security → Options related to MAC addresses.

Now we can study DAI which is Dynamic ARP inspection.

DAI is used against attacks that require l2 – l3 mapping. For example, an attacker can send gratuity ARP traffic saying that he is the G/W, therefore creating a MITM attack. To prevent this, DAI compares the ARP messages that come in and if a misleading information is caught, we can implement a violation status.

Now, how do we know which L2/L3 mapping to follow? We can manually configure it or work with DHCP snooping altogether where we know the trusted L2 & L3 addresses.

Layer 2 Controls To Protect Spanning Tree Protocol (STP)

Let's remember how switches decide the root and root ports. Behind the scenes are BPDUs being sent over the switches. Now let's imagine a hacker is able to create superior BPDUs and tell the network that the interface it is connected to is actually a ROOT port.

To fight this, we can enable BPDU GUARD in every interface connected to access ports on clients where we won't send any BPDUs packets there.

ROOT GUARD can also be enabled on root switches where we enable it on the interfaces of the root switch so it does not get confused if either switch connected to such interfaces start sending information that it is not true.

Mitigate VLAN Hopping Attacks

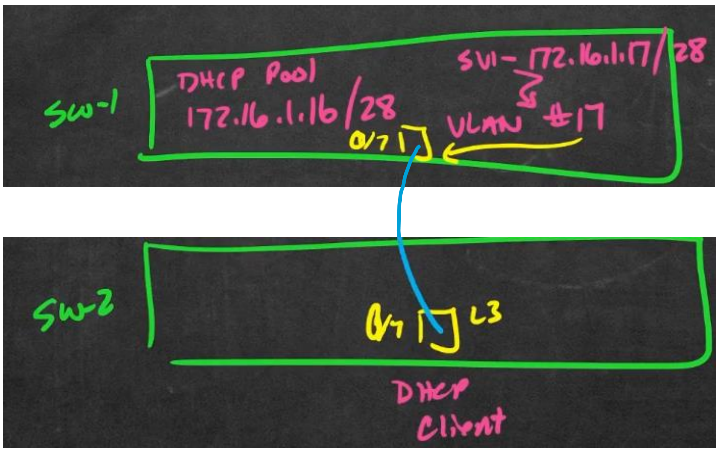
On ports that go to our clients, we want to disable DTP. We do not want CLIENTS being able to negotiate trunking benefits because VLAN traffic depends on it.

Remember that we can send ***switchport mode access or switchport host*** and they will disable DTP.

In order to prevent VLAN hoping we can also create a new native VLAN that is only for trunking that is not really used for anything else. For example, VLAN 66.

Enable DHCP Services and Snooping

DAI will rely on DHCP Snooping's binding table. Let's imagine having 2 switches where the SW-1 has a DHCP Pool of 172.16.1.16/28 and for it's SVI we will use 172.16.1.17/28 linked to VLAN #17



On SW-2 we will tell the interface connected to SW-1 to get its IP address from the DHCP server.

```
Switch(config)#vlan 17
Switch(config-vlan)#name Snoop-DAI
```

```
Switch(config)#ip dhcp pool DAI-Demo
```

```
Switch(dhcp-config)#network 172.16.1.16 255.255.255.240
Switch(dhcp-config)#default-router 172.16.1.17
```

SVI creation

```
Switch(dhcp-config)#int vlan 17
```

```
Switch(config-if)#ip address 172.16.1.17 255.255.255.240
```

Enable DHCP snooping for vlan 17

```
Switch(config)#ip dhcp snooping vlan 17
```

Allow vlan 17 on the SW-1 port

```
Switch(config-if)#switchport access vlan 17
```

Now, we will want to tell SW-2 to get an IP address from the DHCP server interface on SW-1. Also, we will check the snooping binding so we can then finally see the DAI.

Configure Dynamic ARP Inspection (DAI)

It basically goes like this. We know that DAI depends on the DHCP snooping binding table

Whenever a DHCP client wants to send a DHCP query to the DHCP query and the traffic goes through an interface that has DHCP snooping enable for a certain vlan and such vlan does not match with the table, the switch will block that incoming request.

STP BPDU Guard

When BPDU Guard is implemented on a port, it will get put on an err-disable state.

In our topology, lets put BPDU guard on SW-1 access ports.

```
Switch(config-if-range)#spanning-tree bpduguard enable
```

To bring it up again simply shut it down and then turn it on again.

BPDU Filtering

We can also take BPDU and disable it from scratch on ports that we KNOW it should never receive BPDU → STP traffic onto it.

We can apply it GLOBALLY or FILTERING. Let's do it first in 1 particular interface.

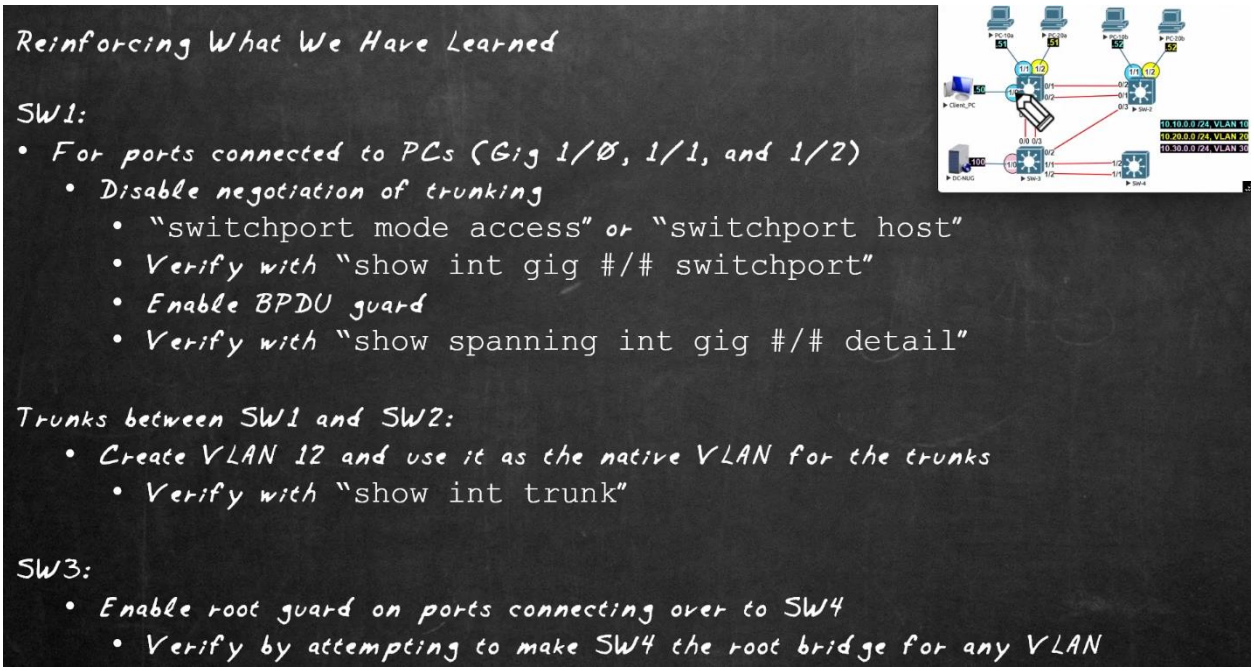
Loop Guard

To prevent a blocking port from transitioning to a forwarding state when BPDUs are no longer received

ROOT GUARD

Root Guard can prevent a switch from changing its root port, even when it sees better/superior BPDUs coming in on a port other than its current root port.

VALIDATION



91.- Describe Cisco AAA

RADIUS &TACACS+

In order to make AAA work as a server, we train the SERVER to the CLIENT and viceversa

RADIUS uses L4 UDP

- ➔ Encrypts only the password
- ➔ Lumps A/A/A

TACACS+ uses L4 TCP

- ➔ Encrypts the entire application layer
- ➔ Better separation for all A/A/A

TACACS+ is better implemented for admins on a network while RADIUS is used mostly for end uses.

AAA Local Authentication of Telnet

Let's make a plan to explain AAA using telnet.

→We are going to implement authentication on R1. By default AAA are not enabled in Cisco routers.

```
Router(config)#aaa new-model
```

→We then enable Telnet specifying the source of 23.1.2.100 by using a ACL

Remember that we use Lines VTY 0-4 in order to implement remote access to routers.

```
Router(config)#access-list 1 permit 23.1.2.100
Router(config)#access-list 1 deny any
```

```
Router(config)#line vty 0 4
```

```
Router(config-line)#transport input telnet
```

And just as any other interface in which you have to apply the ACL, the same is done for a VTY.

```
Router(config-line)#access-class 1 in
```

Lastly, How do we want to do the authentication? By default, the default uses the information in the router to authenticate.

```
Router(config-line)#login authentication default
```

AAA Local Authentication of SSH

The steps for SSH is a bit different since we have to enable encryption for SSH

1.- Verify routing information

2.- Key generation

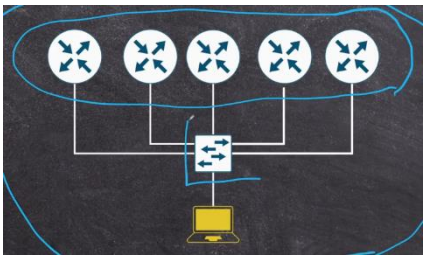
3.- AAA new-model

4.- Line VTY 0 – 15 (because we are doing it on the switch)

    For the transport protocol we say SSH v2

5.- Login authentication default (by default we use the users and passwords already configured in the switch)

92.- Understand Automation for Network Management



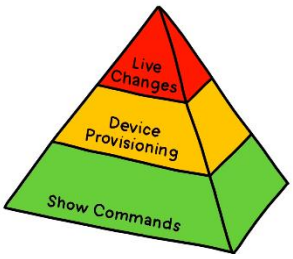
Consider the small network on the left. There are ways to configure the devices. Network admins have noticed that there are a lot of commonalities between devices, especially when they are within the same network.

Maybe we need to configure ip domain-name OR ip ssh version. There are a lot of commands that we would need to put into each and every single on of our devices.

We can use tools to automate these processes such as Python, Ansible, etc.

The 3 most important aspects to consider implementing network automation are TIME SAVE, SACALABILITY & MINIMIZE HUMAN ERROR.

What should we Automate?



Automation can also be deadly. So what should we automate?

1.- Let's see the show commands

2.- Start Pushing basic configurations like:

NOT AUTOMATE

1.- ACL

2.- QoS

3.- Routing paths and protocols

Push vs Pull

When we use ansible for example to manage the network, we want a 2 way conversation.

There are 2 architectural models in network automation. The PULL model you can conceptualize it as 'Always asking the manager node Am I ok?' and with this, ensure that configurations are being PULLED constantly.

The second model is the PUSH model. As long as the automation system can talk to all the devices [for example with SSH], the main system can generate configurations and PUSH them out to all the devices.

What is Ansible?

NR1 utility to use the PUSH model is ANSIBLE.

Written in python and it makes it Agentless.

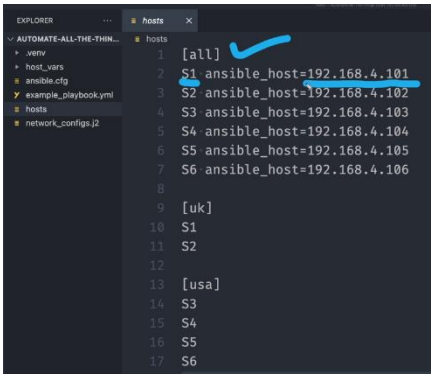
Ansible uses YAML which is a format to raise configurations which is HUMAN-READABLE

Ansible uses a PLAYBOOK which defines the instructions that we will tell ansible to push into which device.

The INVENTORY defines a list of devices configurable.

Ansible in Action

On Ansible we can see all of these components in action.



When you go to your server ansible host and see on hosts, notice how all devices show up with their IP addresses.

We also see GRIPS divided in [UK] & [USA].

Keep in mind that our PLAYBOOK is written in YAML.

When it comes to configurations, let's say VLANs, we will want to make a template that follows the same instructions but can adapt to each device properly. Cisco ANSIBLE SYNTAX allows us to do this. We won't have to learn YAML and Ansible configuration processes, just know that this process exists and you are capable of at least understand it. (Very similar to objects and classes)

Puppet and Chef

A PUPPET is inspired on Ruby. It uses PULL MODEL explained above that it's name Agent-Based. The device managing the network called 'Puppet Master' uses manifests.pp

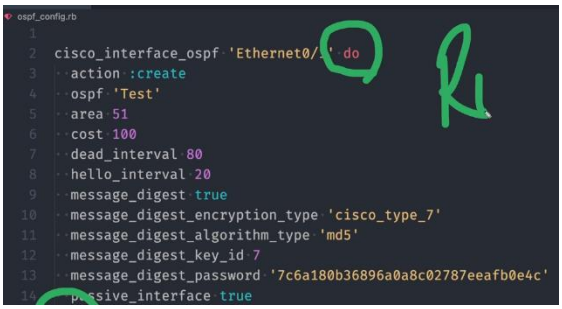
So for example if we are configuring OSPF for devices in the network, we will called it ospf.pp

Remember that as a PULL method, the devices on the network will be asking for the correct configuration constantly.

This is an example of a NTP configuration manifest file. In this case, puppet uses DSL language.

For CHEF, we have COOKBOOKS which have collections on RECIPES. The command utility is Knife.

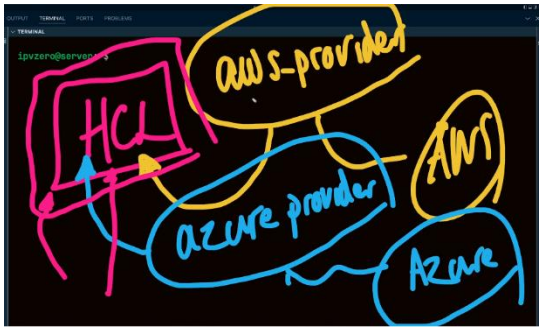
A recipe is similar to a Ansible playbook or a puppet manifest. It defines the configurations of devices.



What is Terraform?

It is a solution in the Cloud infrastructure. It operates by managing the cloud infrastructure via syntax text files called HCL. In this file we will list all the things we want in a DESIRED STATE (Ex. Stating the amount of servers and which OSs we will be using).

Terraform is Declarative, meaning that when it tracks the state of the configurations, and when a change needs to happen it will compare the changes. Terraform makes this happens via PROVIDERS, which is like a translation layer. When you write your file HCL and state that AWS will be your provider, AWS will be able to translate all the instructions all the instructions and make it a reality for AWS. The same way, if you specify AZURE to be your provider, this can happen as well. HCL is adaptable to each provider.



REMEMBER: A PROVIDER ACTS AS A TRANSLATION LAYER BETWEEN HCL FILES AND THE CLOUD PROVIDERS.

Terraform in Action

Lets see how terraform can be used to create a Virtual Machine in the cloud and a new tenant within Cisco's ACI

.tf is the extension of the HCL files used for cloud terraform.

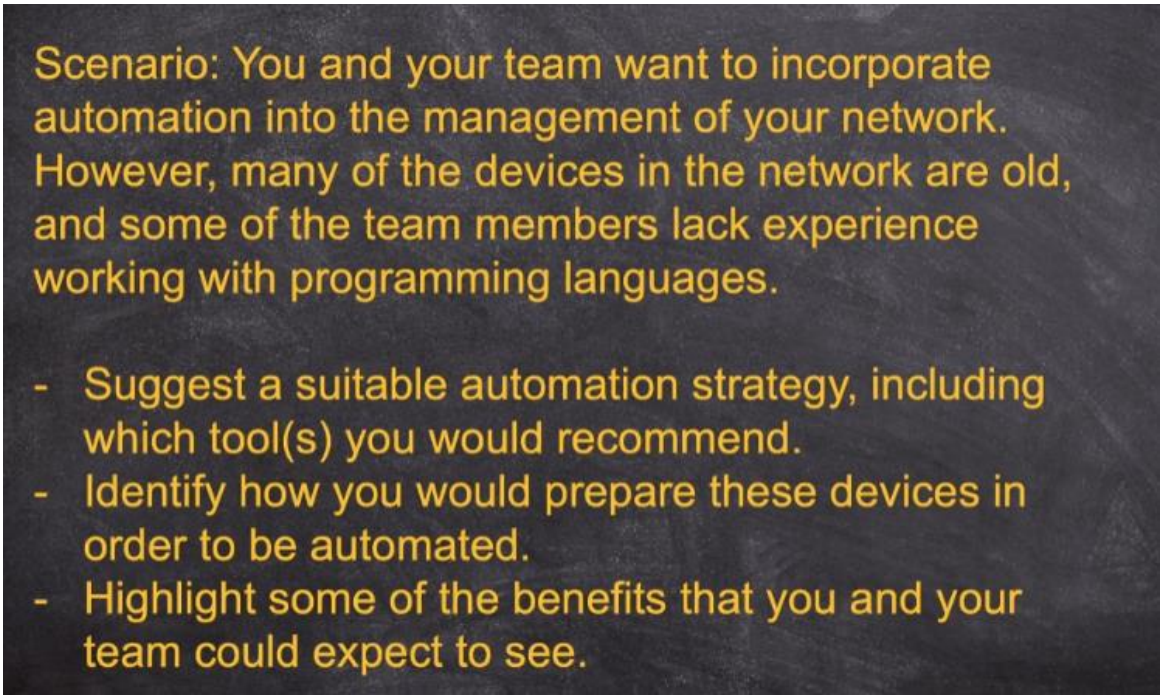
Artificial Intelligence

AI has 2 styles:

- ➔ Generative: Like CHATGPT, we can generate network configurations. The way AI does this is via past data that is fed to it.
- ➔ Predictive: In terms of a network, we can see sharp traffic demands. It get complicated to protect. What if you can feed the Historical data of your network to your AI to make

predictions in order to allocate resources, implement security measurements, prevent anomalies.

Validation



I would start slow by automating the configuration commands like R/O with the PUSH model, since they are old, an agentless solution would be proper. I would use Ansible since the team members are not used to programming languages.

We will also need to set up SSH with username and password.

We can see benefits such as making configuration changes faster and reliably. We will reduce human error and also the amount of operational cost.

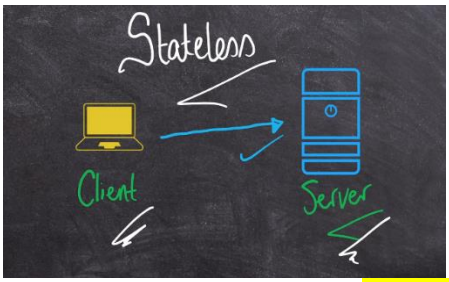
93.- Understand REST APIs and Data formats

What is an API?

An APPLICATION PROGRAMMING INTERFACE means a 'bridge' between 2 apps that need to communicate between each other.

We use APIs all the time, for example when we install an app on our phone developed in a certain language that needs to communicate with a server. The server needs an API to communicate with the client on your phone.

Understanding REST APIs



An architectural style for APIs is REST. These are often used for Web Services. RESTful uses a CLIENT ---- SERVER model. It is Stateless which means that the client nor the server do not need to remember or save any information about each other.

HTTP/S uses methods to specify the ACTION needed to be implemented in a web server. Headers are also important to specify the type of information being sent. For example, XML, Json, etc.

Let's imagine that we can take a very specific resource from a web server, we can identify this resource via an URI. For example when we change the main path on our browser.

https://en.wikipedia.org/wiki/Main_Page → https://en.wikipedia.org/wiki/Canada

Let's imagine that we have a server with an 192.168.1.1 and we path to a specific resource

https://192.168.1.1/bgp

the resource tells us that we want the bgp and the REST API is able to understand this, to map the URI to the resource we want.

Methods and Headers

CRUD is an important acronym which tells us that we can create resources that do not exist on the HTTP server. CRUD => Create Read Update Delete

For Create → The method is called POST.

Read → Method is GET.

Update → Method is PUT/PATCH

Delete → Method is DELETE. For example we can especify a resourse to delete via an URL.

https://192.168.1.1/loopback99 [Delete request]

HEADERS

Accept → When the client talks to a server

Content-Type → When we POST, we have to specify the type of content we are sending.

HTTP Status Codes

A status code is whenever a client sends a request, the server responds some type of status code.

There are 5 classes that range from 1 <= 5.

Class 1 ➔ Informational response. It communicates that the server received the request and to standby.

Class 2 ➔ This indicates that we have had a SUCCESS.

Class 3 ➔ Means redirection. The resource you are trying to get is perhaps not there, or it was moved temporarily. This depends on the last 2 numbers.

Class 4 ➔ Client Error. Bad Request. Perhaps a Syntax

Class 5 ➔ Server Error.

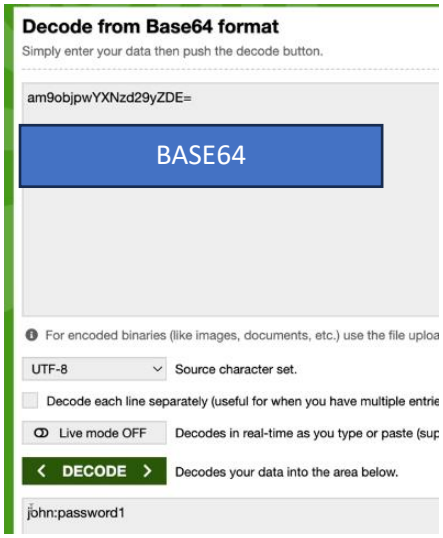
HTTP Authentication

Lets imagine a Server with a REST API so we can access it.

If we are a random person, it is not intelligent to allow anyone to make requests to a server.

Via authentication we can. The same way we have seen authentication in many other ways, we use password and username.

We will want to combine authentication with a powerful encryption method. HTTP does not encrypt, so BASIC AUTHENTICATION SENDS base64 encoded username and password that can easily be exploited.



Another solution is known as TOKEN.

With a TOKEN you give access to a bearer. The user simply presents the token and he will have access. You can revoke a token and they expire. The same way, if the traffic is not encrypted, you will have problems.

Another solution is API KEYS

These are valid for a indefinite period of time.

Understanding Structured Data

A structured data is for example JSON.

<---------Structured data looks like the image

This is good for automation and programmability.

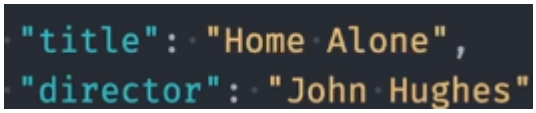
Python, JavaScript are examples of languages that are compatible with JSON and can interpret structured data.

Data Serialization and JSON

The ability to CONVERT data from a format like JSON, XML, YAML =======> python, javascript, etc

These type of data serialization languages like XML, YAML provide neutrality to communicate 2 applications.

1.- JASON has KEY/VALUE format



2;.- A JSON object separated by brackets we see in the image.

3.- An ARRAY is separated by squared brackets.

XML and YAML

This is the exact data as the previous JSON exercise ---->

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <movies>
    <title>Home Alone</title>
    <director>John Hughes</director>
    <published_year>1990</published_year>
    <genre>Comedy</genre>
    <rating>10</rating>
  </movies>
  <movies>
    <title>The Mighty Ducks</title>
    <director>Stephen Herek</director>
    <published_year>1992</published_year>
    <genre>Sports</genre>
    <rating>8</rating>
  </movies>
  <movies>
    <title>Back to the Future</title>
    <director>Robert Zemeckis</director>
    <published_year>1985</published_year>
    <genre>Science Fiction</genre>
    <rating>10</rating>
  </movies>
</root>
```

```yaml
movies:
- title: Home Alone
  director: John Hughes
  published_year: '1990'
  genre: Comedy
  rating: 10

- title: The Mighty Ducks
  director: Stephen Herek
  published_year: 1992
  genre: Sports
  rating: 8

- title: Back to the Future
  director: Robert Zemeckis
  published_year: 1985
  genre: Science Fiction
  rating: 10
```

We have the ANGLE brackets that give us the data

Indentation DOES NOT matter.

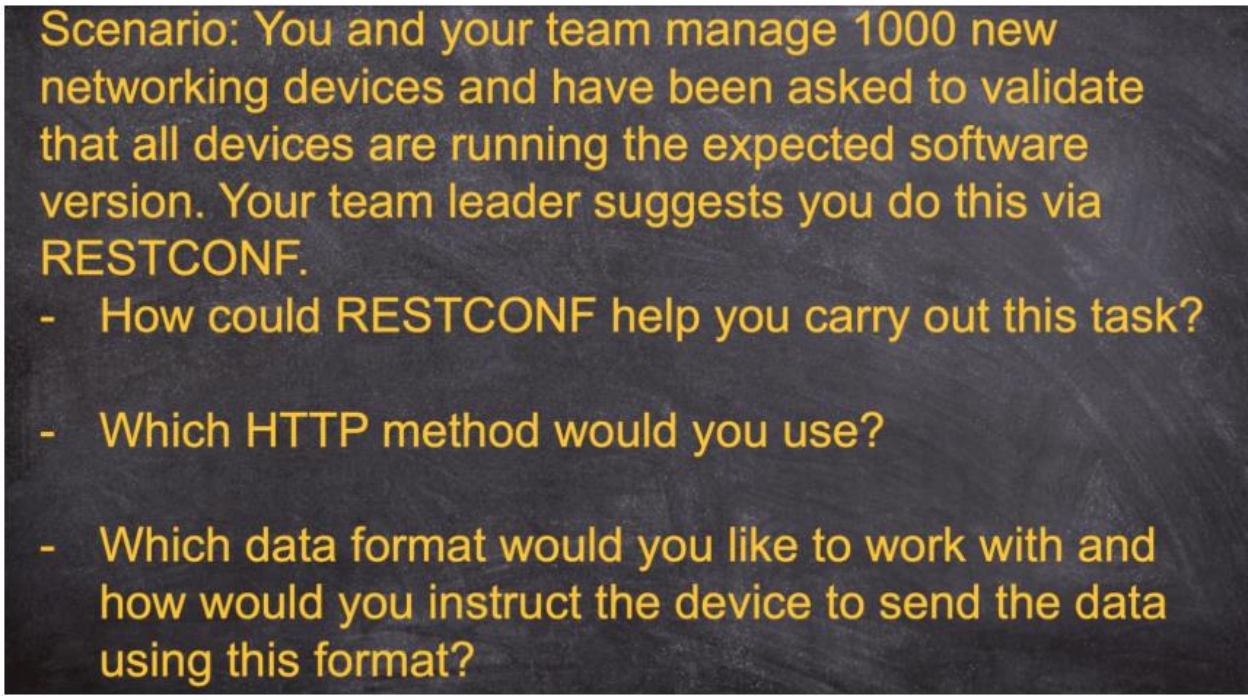
YAML is the one on the right. YAML DOES care about indentation.

Exploring RESTCONF

Remember that a router CAN act as an API

Cisco offers cisco devnet that gives you lab access to different cisco products like IOS XE catalyst.

VALIDATION



- ➔ RESTCONF can helps us by getting the structured DATA we need to organize it and read it. We can target and use this data with any language of our choosing.
- ➔ The HTTP method would be POST/PUT/PATCH since we are not doing anything else other than retrieving the data
- ➔ JSON. We tell the device via an UI of RESTCONF.

95.- Understand Software-Defined Networking

The Data Plane

1.- Logical planes → In our networks we have many devices doing different activities and running different protocols.

The data plan also known as the FORWARDING plane is all of those interactions or processes that our devices do when forwarding traffic and all of the protocols involved in that process are also encompassed in the Data plane.

NOTE: The switching logic happens at the ASIC of our router/switch.

The Control and Management Planes

The control plane is responsible for making decisions about how traffic should flow through the network. Routing protocols, MAC address learning, STP, ARP/ICMP redirects routing and forwarding tables are all processes that the control plane manage. It runs on the CPU

The Management PLANE is used to monitor and manage. SSH, TELNET, HTTPS, SNMP and any configuration changed via a GUI or CLI or API.

Software-Defined Networking Fundamentals

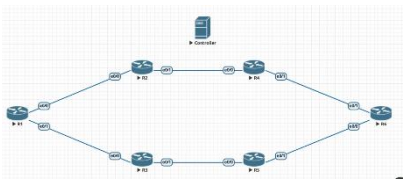
All planes live within all of our devices in the network.

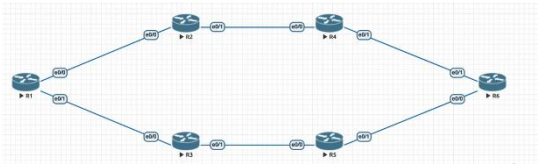
1.- NORTHBOUND API → When we want to communicate OUTSIDE our network. Ex. A controller wants to read or POST from another API.

2.- SOUTHBOUND API → When we want to communicate within our network. Ex. The controller wants to push out a new ACL. That communication from the controller to the device is via the SOUTHBOUND API. Another example is actually logging in to the the device via CLI SSH.

- Openflow
- Cisco Opflex
- Net Conf

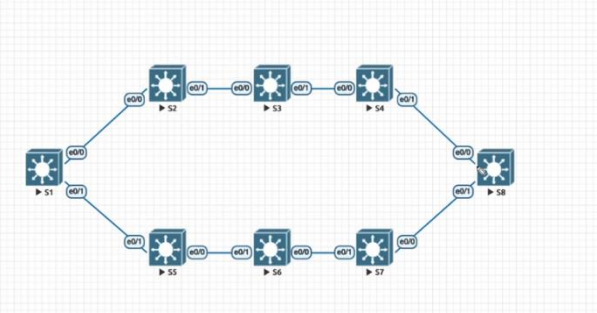
Comparing Traditional networks & Controller-based networks





A controller-based network is a subset of automation.

Understanding Software-Defined Access



Now that we understand what a software defined network is, we will see what SDA (software defined access) is. SDA is an implementation for Software defined networks. On campus enterprise networks SD-ACCESS are used. Cisco has a solution called DNAC.

Let's look at some concepts:

1.- Underlay → In few words is the physical infrastructure that supports the overlay network.

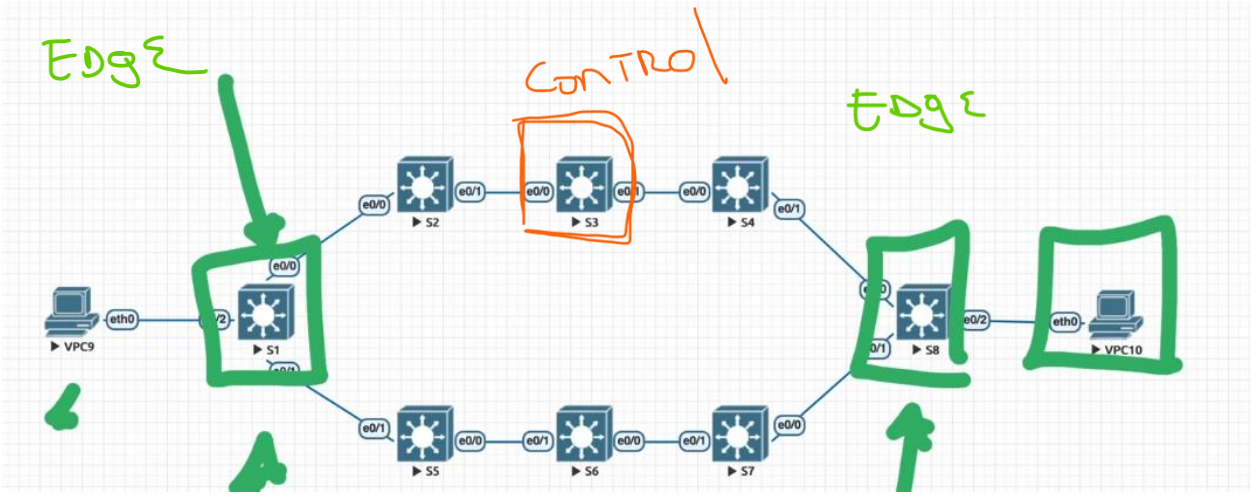
2.- Overlay → A VPN is a great example. Encapsulation serves as an overlay example that tells us a software defined solution built on top of an already existing infrastructure.

3.- Fabric → Combination of the Underlay and Overlay.

→ Fabric edge: devices on the edge of your network

➔ Fabric border: Suppose we have a Wide area network connected to a device that serves as the connection.

➔ Fabric control: Node that provides control plane capabilities.



Exploring DNA Center

Some of the advantages Digital Network Architecture gives is are:

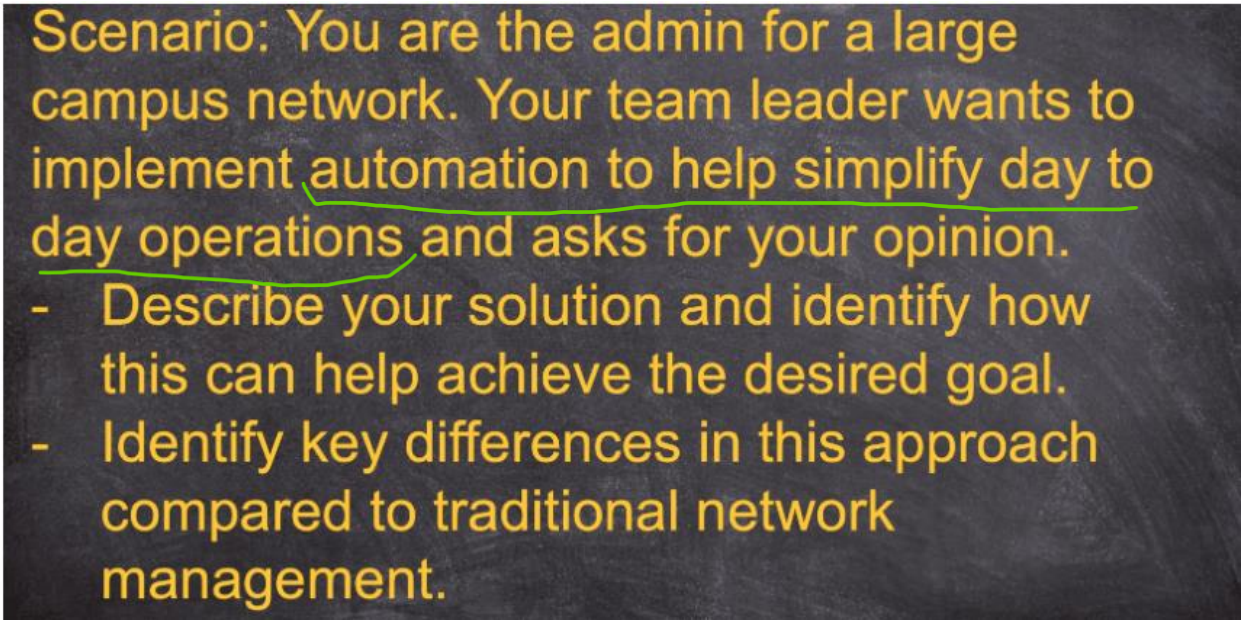
Cisco DNA (Digital Network Architecture) Center is a network controller used to automate SD-Access networks

1.- Creation of groups and people that we can set policies and procedures for all of them. Authentication is a great example that enables clients to roam through a network without the need of authenticating multiply times.

2.- We can also manage/add new devices.

3.- We can monitor the health of our devices as well as identify problems and suggest solutions.

VALIDATION



We are talking about the management plane. Since it is a Campus network, SD-Access along with DNA center is a great option.

We need a Controller where we will get a Centralized management.

The biggest differences are that we will have a Controller and a Centralized Control Plane. In a traditional network, we have a DISTRIBUTED CONTROL PLANE. We will also see the Northbound and Southbound interfaces that connect REST APIs with our controller.