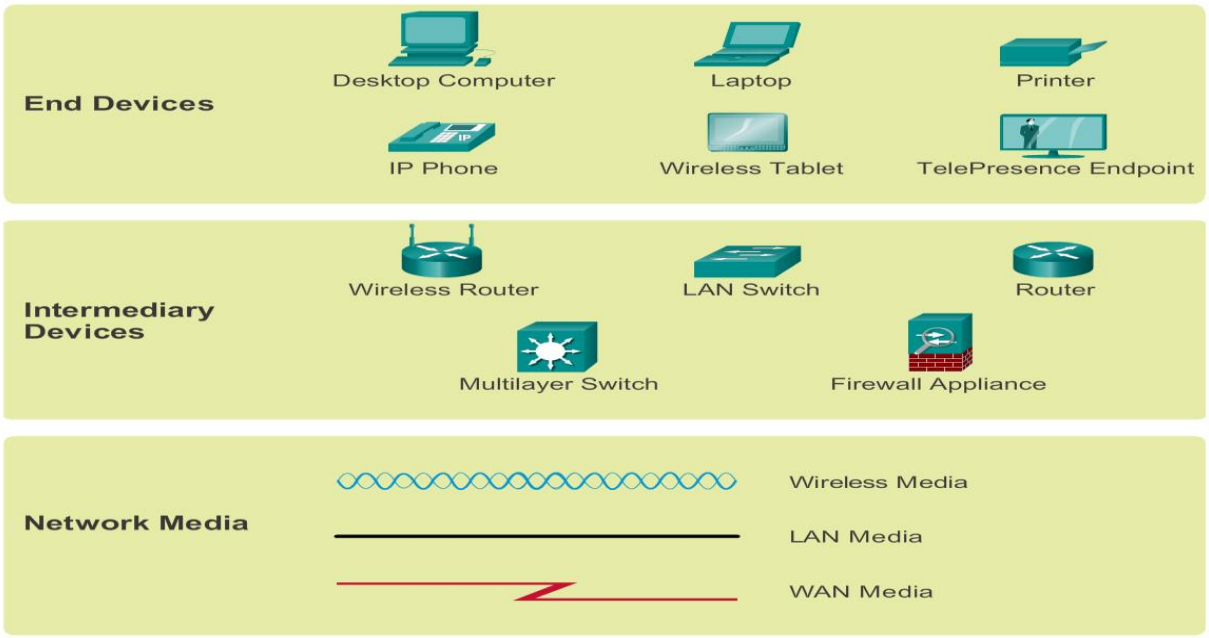
CURSO DE REDES CNNA GETAFE

COMPONENTES DE UNA RED

Hay que distinguir entre DISPOSITIVOS, MEDIOS y SERVICIOS



- TIPOS DE DISPOSITIVOS (Host): PCs, servidores, Tfnos IP, etc.

Todo dispositivo al que se le asigne una IP se le llama host o dispositivo final

Dispositivos intermedios: Switches, Routers, Firewalls, Puntos de acceso, etc: Estos dispositivos también pueden ser hosts.

- TIPOS DE MEDIOS (de interconexión): Fibra, Cable, Wifi

Medios WAN: Conectan redes LAN entre sí. Lo forman los Switches y Routers

Los medios WAN pueden ser:

- Privados: Con tecnologías FR, ATM, MPLS, RDSI…
- Públicos: Internet

Los medios LAN: son de dominio privado.

TIPOS DE SERVICIO: Los servicios son los contenidos.

Tipos de redes: PAN, LAN, MAN, WAN, SAN, CPDs

PAN: Personal Area Network

LAN: Local A.N.

MAN: Metropolitan A.N. (desaparecidas)

WAN: Wide A.N Los redes privadas tipo macrolan de telefónica o las iVPN se contratan. No dejan de ser sino WANs privadas. Actualmente la más moderna es MPLS

SAN: Storage A.N.. Se alojan servidores de almacenamiento

CPD: Centro de Procesamiento de Datos. Son recintos que solo se almacenan servidores de grandes empresas. Ej: 5 sucursales de El Corte Inglés. Cada sucursal tendrá su propia SAN y el total de las 5 SAN sería una CPD. Una CPD puede ser física o estar en la nube. Google ofrece CPDs en la nube.

La WAN pública (Internet) puede ser IPV4 o IPV6

- DIFERENCIAS ENTRE INTERNET, EXTRANET E INTRANET

INTERNET: El mundo, lo abarca todo

INTRANET: Red a la que solo pueden acceder miembros de la compañía.

EXTRANET: Red de menos confianza propia de empresas que colaboran entre sí. Forman parte distintas compañías

Tanto en extranet como en intranet, internet ejerce como medio de transporte

*El tipo más caro de conexión es un “Enlace Dedicado”

*En todo mensaje hay dos tipos de información:

Información de usuario: lo que a simple vista se ve

Información de Control y Protocolo: para ver esta información hay software como el WireShark

Conceptos de apoyo a la arquitectura de red:

- Tolerancia de fallo
- Escalabilidad
- Calidad de servicio
- Seguridad

Técnicas de Conmutación (pasar un paquete de una interface a otra) en WAN:

- Conmutación circuito: Se establece un camino dedicado único: PSTN, RDSI
- Conmutación paquetes: FR,ATM, MPLS, INTERNET

Da igual que la WAn sea pública o privada: es cuestión de técnicas de conmutación

Conceptos:

RAM: Aquí se ubica el archivo Running-Config, que es el archivo que se va generando mientras estamos configurando en línea de comando. Cuando estamos configurando un Router, toda la información que le vamos metiendo se copia en la memoria RAM. Como la RAM es una memoria volátil, para poder salvar esta configuración, hacemos un COPY RUNNING-CONFIG STARUP-CONFIG para que toda esa configuración se guarde en la NVRAM. Este archivo generado es el StartUp Config que es el archivo que le hace falta al S.O. para saber qué configuración tiene.

Cuando se arranca el router, va a buscar ese archivo (el starUp Conf.) a la NVRAM. Una vez que empezamos a configurar el Router, este archivo (StartUp Conf) sube a la RAM y es con el que trabajamos cuando configuramos. Por eso una vez terminado de configurar, debemos pasar toda esa información que ese momento tiene la RAM a la NVRAM con el comando: COPY RUNNING-CONFIG STARUP-CONFIG

ROM: Aquí se ubica el BootStrap. Aquí están las instrucciones que usa la CPU para arrancar el equipo. Revisa que los componentes esté bien hasta llegar al S.O. en la FLASH

NVRAM: Aquí se ubica el StartUp Config.

FLASH: Aquí está el S.O. IOS

Con el comando COPY RUNNING-CONFIG STARUP-CONFIG estamos copiando la configuración de la memoria RAM a la memoria NVRAM. **El comando copy se hace desde raíz: R1#:copy….**

CONTRASEÑAS:

Para la contraseña de enable, hay que estar en modo Conf: enable secret + la password

Para el resto de contraseñas, habrá que meterse en la línea correspondiente:

- Line console 0
- Line vty 0 4
- Line aux

\+ password + contraseña + login

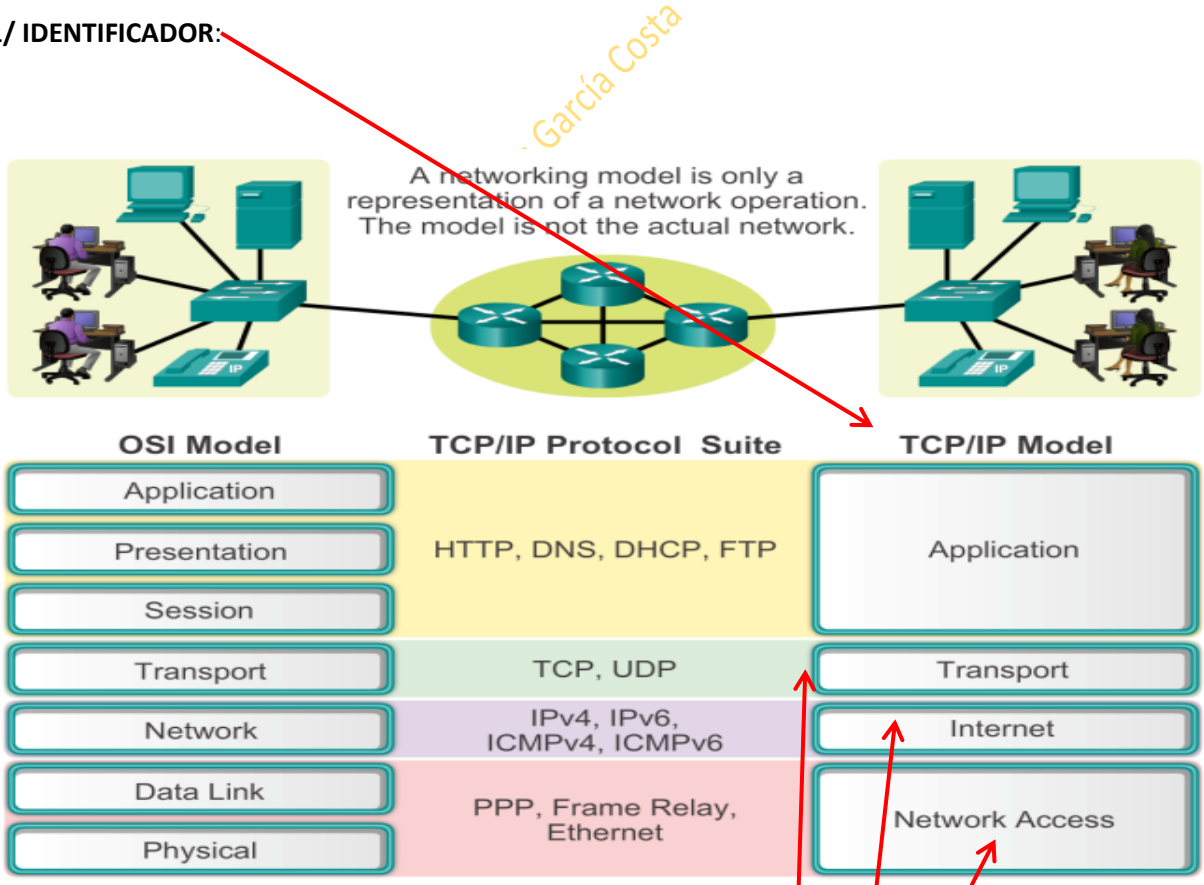
El comando Service password-encryption se hace desde config# y sirve para encriptar todas las contraseñas line

Todos los comandos show se hacen desde raíz: R1#:

El comando **show ip interface brief** nos muestra la configuración de todas las interfaces.

En los mensajes, las reglas son:

1/ IDENTIFICADOR:

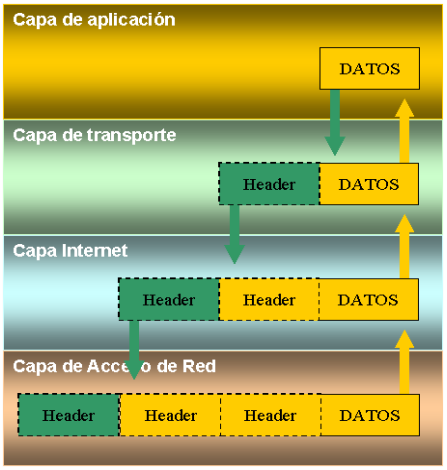


En el identificador de transporte, su identificador es el nº de puerto.

En el identificador de internet, su identificador es la dirección IPV4 ó IPV6

En el identificador de acceso a la red es ethernet y se llama dirección MAC

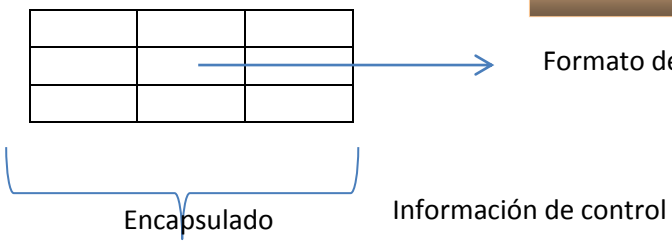
Se pueden tener esos 3 identificadores o solo 1



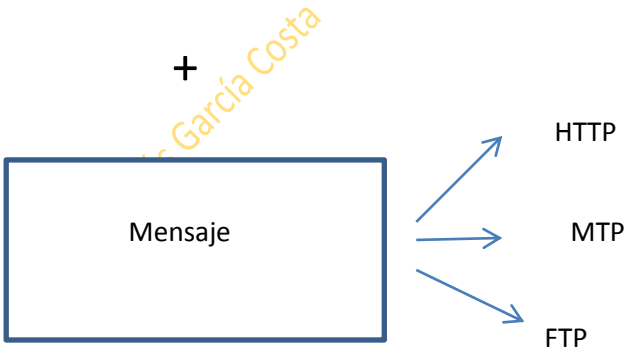
2/ PROTOCOLOS:

- Definen el formato del mensaje
- Definen la encapsulación

El encapsulado tiene dentro divisiones que se llaman formato del mensaje



`



Una MTU es la unidad máxima de tamaño por paquete. Si un mensaje supera el MTU, se divide en partes, cada una con su encapsulado. Cuando el mensaje es recibido en su totalidad, se reagrupa.

En una TRAMA, el preámbulo es la sincronización (lo que advierte al destinatario lo que le viene para que se prepare para recibirlo) y no cuenta en la suma de bytes.

Si la Trama es más de 1518 bytes, se fragmentará y si es menos de 64, se despreciará

La MTU es el tamaño mínimo por paquete. Este tamaño se puede modificar en el registro del S.O.

La MTU es la componen la Trama + los datos: su suma

Configuración interfaces

Para asignar la INTERFACE (puerto de conmutación con otra máquina) el comando es INTERFACE + el nombre que tenga el puerto. Ej: R1(Config)#Interface+ser1/0. Así el prompt quedará de la siguiente forma: R1(Config-if)#

Dentro de cada router hay que ir configurando uno a uno las distintas interfaces, bien sean fa (propias de la red LAN) o se (propias de las redes WAN). Una vez configurado la interface, hay que salirse de ella en línea de comandos del router y meterse en la siguiente que vayamos a configurar hasta terminar la configuración entera del router.

Los pasos para configurar cada interface son:

1º darle IP mediante el comando IP ADDRESS= se pone la ip que asignemos más la máscara de entrada correspondiente. Para corregir una ip mal puesta podemos hacer dos cosas. O la sobrescribimos directamente con el mismo comando más la ip nueva, o ponemos NO IP ADDRESS MAS LA DIRECCION MAL PUESTA Y SU MASCARA, Y DEPEUES VOLVEMOS A PONER LA BUENA.

LA IP DE LAS INTEFACES TIENE QUE SER LA MISMA QUE LA GATEWAY QUE HAYAMOS PUESTO EN LOS EQUIPOS

2º darle el comando NO SHUTDOWN para abrirlo

3º Si es una de los routers que necesita otorgarle velocidad, darle el comando CLOCK RATE + la velocidad que acostumbra a ser 64000.

Recordemos que en la interconexión entre dos routers, siempre uno de ellos debe otorgársele la velocidad.

Por defecto los puertos o interfaces, vienen cerrados en los routers y hay que abrirlos.

Cada LAN tendrá su propia red

La IP que se le dará a la interface es una IP de equipo. Para conectar un Router a otro hacen falta las dos direcciones IP de cada router. *(Ojo! No repetir es IP)*

Así que, cuando estamos en el prompt R1(Config-if)# el comando es IP ADDRESS 192.168.1.1 + la máscara 255.255.255.0. Esto es solo un ejemplo. La dirección Ip será la que le pongamos, y la máscara la que necesite.

Cuando se conectan dos routers entre sí, al menos uno de ellos hay que ponerle velocidad con el comando CLOCK RATE+LA VELOCIDAD=64000(en bits por segundo). De tal forma que sería: R1(Config-if)#clockrate 64000.

Después, para levantar el puerto (abrirlo) se usa el comando: NO SHUTDOWN. Esto hay que hacerlo en ambos puertos de los distintos routers. R1(Config-if)#NO SHUTDOWN

Conceptos:

Un loopback es hacer ping a nuestro propio host

La MAC va en capa 2

Cada interface del router es una red independiente

Cada red (interface del router) es un DOMINIO DE BROADCAST (de 1 a todos)

El Router separa redes

Los equipos que pertenecen a una misma red, usan la MAC (capa 2) para identificarse

Los equipos de distintas redes se comunican por IP (Capa 3)

La MAC es la dirección física de un equipo

TABLAS ARP:

En las tablas ARP (que son gestionadas por el propio host) se guardan la relación de IPs y MACs. Como comentábamos antes, los equipos que pertenecen a una misma red, usan la MAC (capa 2) para identificarse, mientras que los equipos de distintas redes se comunican por IP (Capa 3).

Cuando un equipo quiere comunicarse fuera de esa LAN, por ejemplo con internet, el equipo buscara la IP de la puerta de enlace en vez de la MAC .

Las tablas ARP se van guardando en el host.

Por eso, cuando un equipo manda un PING, lo primero que hace es consultar con tabla de enrutamiento ARP

IP	MAC

El paquete completo que envía el ping se llama ICMP y está compuesto por:

- IP de origen: Ej: 192.168.1.15
- IP de destino: 192.168.1.68
- MAC de origen: AAA
- MAC de destino: DDD

Paquete completo ICMP

Si las tablas de enrutamiento están vacías porque el host sea nuevo o se haya reiniciado, cuando se hace un ping, lo que se hace es enviar un **ARP REQUEST**, o lo que es lo mismo, un ping de broadcast. Cuando se envía un ping de broadcast, la IP la sabemos, pero la MAC no. Por eso la MAC es **FFF**, que equivaldría a un ping 255.255.255.255.

Una vez mandado el ping de broadcast o ARP REQUEST, cuando el equipo destinatario con la IP que pusimos nos responde, lo hará con un **ARP REPLAY**, que será unicast porque ya conoce el origen que ahora será destino.

Un ARP REQUEST también se usa para conseguir una dirección MAC.

- Cuando se manda un **PING a un host que pertenece a otra red**, se necesitará una puerta de enlace, ya que la red de destino podrá estar a 1 o más saltos.

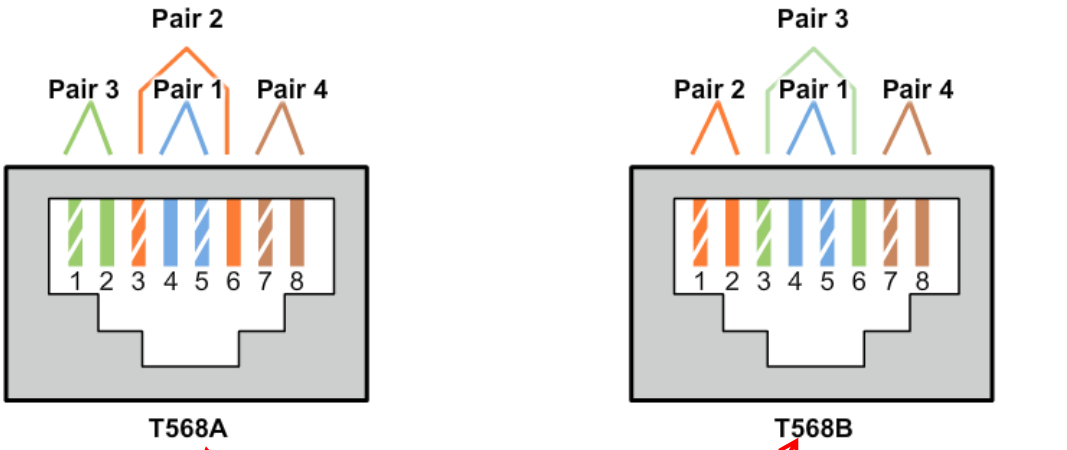
Ejemplo, si mandamos un ping desde 192.168.1.10 a 172.16.1.20 que evidentemente son redes distintas, nos hará falta además la dirección IP de la puerta de enlace 192.168.1.1 (interface del Router), por lo que hará falta también la MAC del Router para poder completar el paquete ICMP.

El Router, además de tener su propia tabla ARP también tiene una "tabla de rutas" (de las redes que conoce)

En distintas redes cuando la información llega al Router, al salir por la interface de la otra red, el Router mandará un ARP REQUEST para saber el destino host y éste le devolverá un ARP REPLAY y así generará su propia tabla ARP

*cada puerto/interface de un SW tiene una MAC distinta

CABLEADO



Cable Type	Standard	Application
Ethernet Straight-through	Both ends T568A or both ends T568B	Connects a network host to a network device such as a switch or hub.
Ethernet Crossover	One end T568A, other end T568B	• Connects two network hosts • Connects two network intermediary devices (switch to switch, or router to router)
Rollover	Cisco proprietary	Connects a workstation serial port to a router console port, using an adapter.

El modelo USA/Japón es el T568A y el europeo el T568B

El CABLE DIRECTO se usa para conectar equipos de distinta capa: Ej: Router es de capa3 con SW que es de capa 2

T568**B**----------T568**B** Cable directo

T568**B**----------T568**A** Cable cruzado

En cuestión de cables, hay que prestarle atención al modelo FTTH (fibra to that home)

Domino de colisión: Se produce en Hubs y Puntos de Acceso al ser medios compartidos. Es cuando por un medio solo puede ir a la vez un paquete de información. Los demás tendrán que esperar porque si no, colisionan. Por eso la información en estos medios es más lenta.

En un SW que tenga 24 interfaces tendrá 24 dominios de colisión. Cada interface del SW es 1 dominio de colisión. Un Hub con 24 puertos será 1 sólo dominio de colisión.

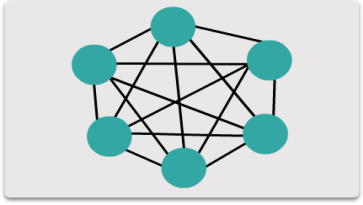
- Un SW limita los dominios de colisión.
- Un Router limita los dominios de broadcast.

Tipos de conexiones físicas:

- Punto a Punto
- Estrella o Hub and Spoke
- Todos con todos o Full Mesh

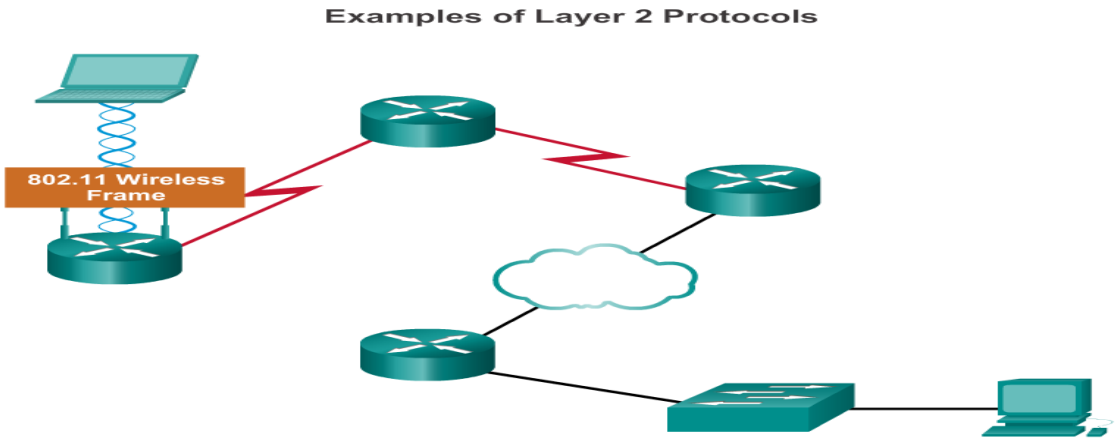

Point-to-point topology

Hub and spoke topology


Full mesh topology

Encapsulación: el protocolo de capa 2 que se esté usando, determinará el encapsulado.

Aunque sea el mismo medio de capa 2, también cambiará el encapsulado a l ser distintas MAC



La capa 3 no cambiará porque las IPs seguirán siendo las mismas, tanto la de origen como la de destino.

En una TRAMA, el preámbulo es la sincronización (lo que advierte al destinatario lo que le viene para que se prepare para recibirlo) y no cuenta en la suma de bytes.

Si la Trama es más de 1518 bytes, se fragmentará y si es menos de 64, se despreciará

La MTU es el tamaño mínimo por paquete. Este tamaño se puede modificar en el registro del S.O.

La MTU es la componen la Trama + los datos: su suma

Ethernet Protocol

A Common Data Link Layer Protocol for LANs

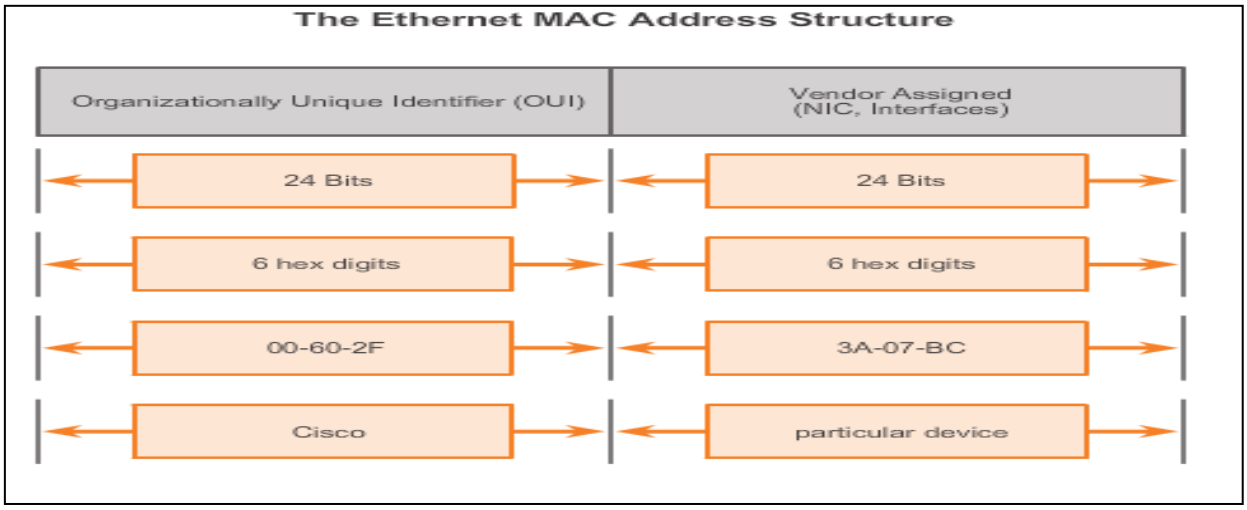
	Frame					
Field name	Preamble	Destination	Source	Type	Data	Frame Check Sequence
Size	8 bytes	6 bytes	6 bytes	2 bytes	46 - 1500 bytes	4 bytes

Preamble - Used for synchronization; also contains a delimiter to mark the end of the timing information
Destination Address - 48-bit MAC address for the destination node
Source Address - 48-bit MAC address for the source node
Type - Value to indicate which upper layer protocol will receive the data after the Ethernet process is complete
Data or payload - This is the PDU, typically an IPv4 packet, that is to be transported over the media.
Frame Check Sequence (FCS) - A value used to check for damaged frames

Tema 5

Las direcciones MAC compuestas por 6 pares de dígitos, se dividen en:

- Las 3 primeras identifican al fabricante
- Las 3 últimas al puerto



Las direcciones MAC pueden ser también de tres tipos:

- Unicast
- Multicast
- Broadcast

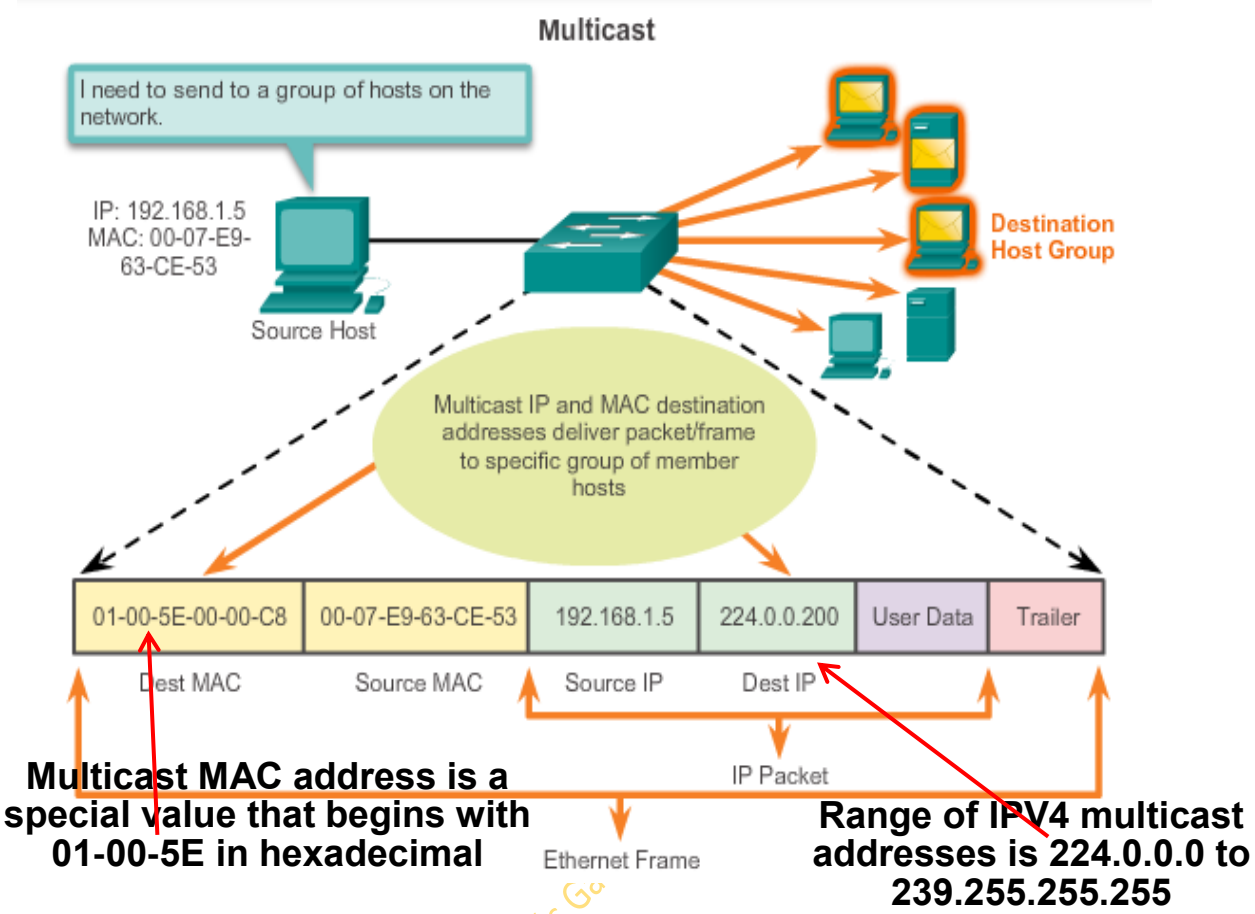
Para averiguar la dirección MAC de un dispositivo de nuestra red, hacemos primero un ping a esa dirección IP, ya sea la de la puerta de enlace o la de cualquier otro host. Al hacer el ping, quedará asociado una MAC a ese ping guardado en la tabla ARP. Esto es así porque si le damos arp –a este comando nos muestra todas las MACs que conoce. Un host puede no cono er todas las MACs porque no la s tiene guardadas en su tabla ARP. Por eso, si mandamos un ping a una dirección concreta de ip, esa información ya SÍ quedará guarda en la ARP con su MAC asociada.

Consecuentemente hacemos un ipconfig /all y vemos las direcciones ip. Mandamos un ping a la ip de la que queremos saber su MAC.

Acto seguido, con el comando arp –a nos mostrará todas las MACs asociadas a las ip que la red conoce y así podremos conseguir la MAC que queramos saber.

Un SW utiliza la dirección MAC de destino para conmutar (enviar) la trama y la MAC de origen para aprenderla. Los SW utilizan las MACs porque son dispositivos de capa 2

Un Router utilizará sólo la IP de destino para enrutar.



Si una Ip comienza por 224. Es una IP multicast, y si una MAC empieza por 01:00:5eE será una MAC multicast.

Conmutación de un SW y apredizaje:

Un SW tiene 2 funciones:

- Conmutar
- Aprendizaje

Las tablas CAM es donde los SWs guardan la información que van aprendiendo.

1º lo guardan en su buffer. El buffer puede ser por puerto/interface o compartido (un solo buffer para todas las interfaces del SW)

El SW va aprendiendo por que puerto le entró la información MAC y lo guardará en su tabla CAM. Estos valores los recuerdan solo 5 minutos. El tamaño de la CAM suele ser 65 K, pero dependerá del tipo de SW.

TABLA CAM

MAC	PUERTO	VLAN	MODO
BBB	1	1	DINAMICO

Otro tipo de broadcast que puede mandar un SW es cuando le llega una petición unicast pero si SW no conoce la MAC al no tenerla en su tabla CAM, tendrá que hacer un broadcast. Una vez mandado el broadcast, cuando el dispositivo le conetsta, el SW guardará esa información en su tabla.

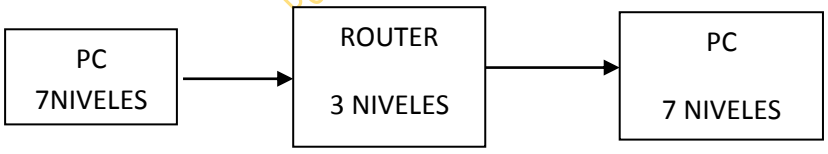
Si la memoria de una SW se llenara, pasaría a comportarse como un hub.

Un SW de capa 3 también enruta, no solo conmuta.

CONMUTAR: Pasar un paquete de una interface a otra. Para que un paquete se pueda conmutar hay que consultar la tabla de enrutamiento

ENRUTAR: Decidir cuál es el mejor camino para llegar a destino. El paquete pasa de un router a otro sin tocarse lo fundamental. Los datos no cambian. Cambia solo la trama para que pueda seguir su camino.

TABLA DE ENRUTAMIENTO: Donde se almacena todas las redes a donde el router sabe llegar. Se almacena en RAM. Lo que llega al router son tramas (3er nivel). El router trabaja en 3er nivel por que se basa en direcciones IP.

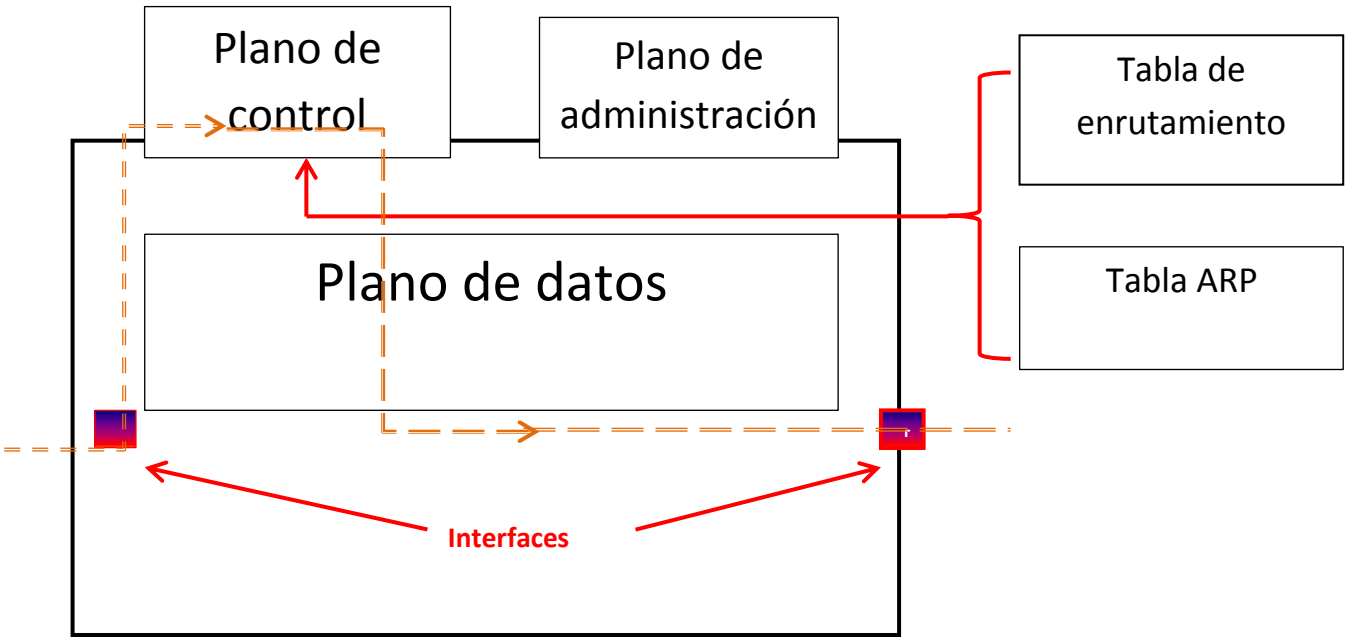


Un Switch solo conmuta. (algunos enrutan también, pero ya lo veremos)

Los SW de capa 3tienen dos tablas fundamentalmente:

- Tabla de enrutamiento (RIB)
- Tabla ARP

SW capa3

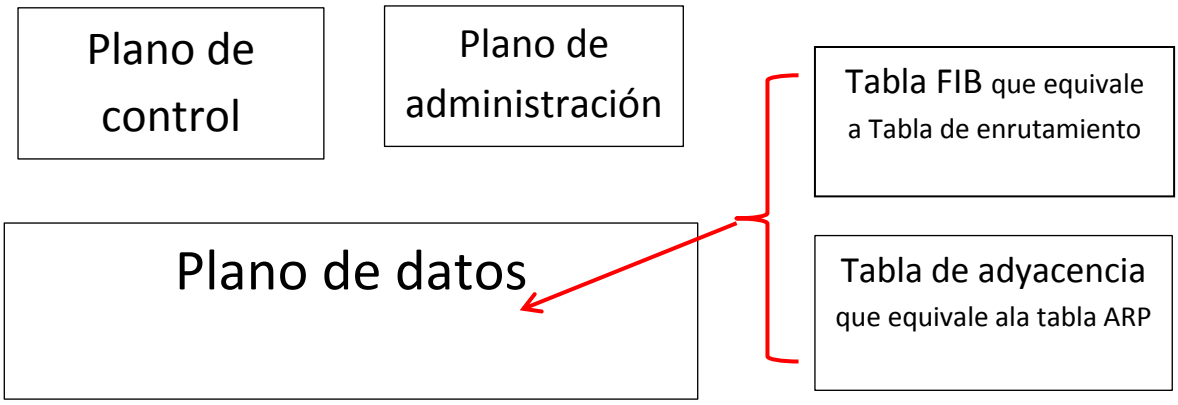


Dentro de un SW, su estructura es como la de arriba. Tiene 3 planos: de control, de administración y de datos. Es dentro del Plano de control donde se guardan la tabla de enrutamiento y la tabla ARP.

Cuando le llega información al SW por la interface, primero tiene que pasar por el plano de control para consultar con la tabla de enrutamiento y con la tabla ARP y después volver a bajar al plano de datos y salir por la interface de salida.

Sin embargo, con la tecnología **CEF** (Cisco Express Forwarding), se copia el plano de control en el plano de datos para que no haya que subir al plano de control. Así se aumenta el reenvío de datos por segundo.

Tecnología CEF



Una interface L3 etherchanel en un SW3 se comportará como un solo dominio de broadcast. Todas las interfaces tendrán la misma IP: misma red a pesar de ser distintos puertos.

El comando SW#show mac-address-table nos mostrará la tabla CAM del SW.

Capitulo 6

A un Router el origen de los paquetes le da igual. Él solo se limita a enrutar. Solo le interesa como llegar al destino.

La base de datos de un Router se llama tablas de enrutamiento.

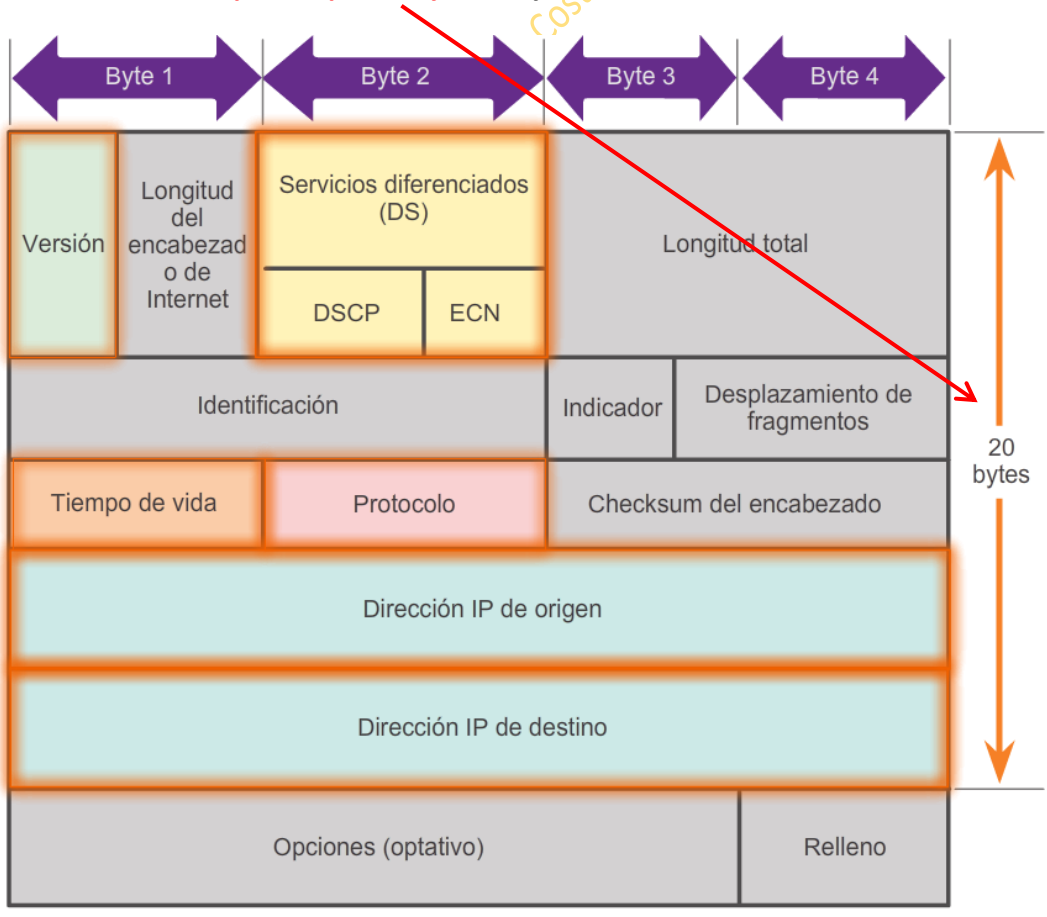
Un router solo usará la IP de origen si tiene orden de filtrar.

IPV4 funciona en Decimal

IPV6 funciona en Hexadecimal, mejora la calidad de los paquetes al tener espacio para las encapsulaciones. En IPV6 no hace falta NAT (NAT ahorra redes ante su escasez)

TCP/UDP en capa 4

El encabezado de IPV4 ocupa siempre 20 Bytes **1 Byte= 8 Bits**



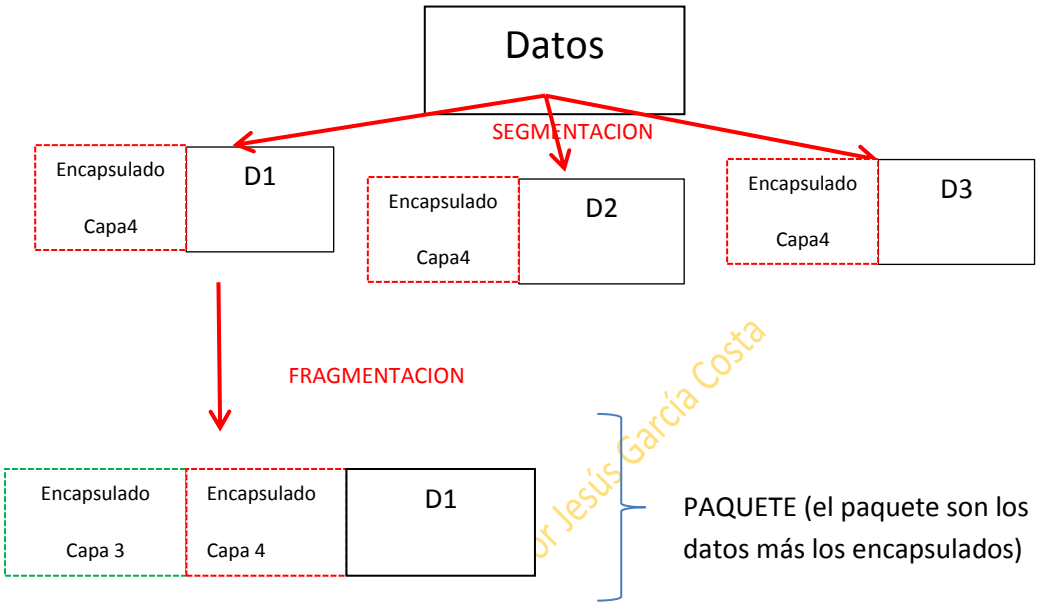
Diferencia entre SEGMENTACIÓN Y FRAGMENTACION

La Segmentación se produce en la capa 4, la capa de transporte.

Ej: un archivo de 2 Mg se tendrá que segmentar si la MTU tuviera un máximo de 1518 bytes en la capa de transporte. Así, la segmentación dividirá los datos en paquetes de como mucho 1518.

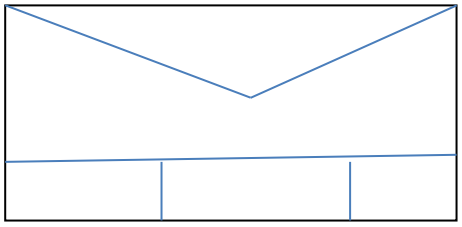
La Fragmentación se utiliza en capa 3 capa de red.

La interface de un Router o el S.O. son los que condicionan el tamaño de la MTU. De tal modo que si un segmento de tamaño 1500 no cupiera por una interface o el S.O. solo admitiera segmentos hasta 1000 bytes, ese segmento de 1500 bytes se fragmentará en 2: uno de 1000 y otro de 500. De tal forma que esos dos fragmentos formarían un segmento.



Un paquete fragmentado se divide en:

- Identificador
- Señalizador (**M**ore **F**ragments)
- Desplazamiento de paquete



Al ir cambiando de capa se le van añadiendo distintas encapsulaciones, lo que implica más información. Por eso el paquete va aumentando de tamaño, por lo que también es motivo o causa para la fragmentación.

Enrutamiento:

Un Router en su tabla de enrutamiento (RIB=Base de información del Router) tiene:

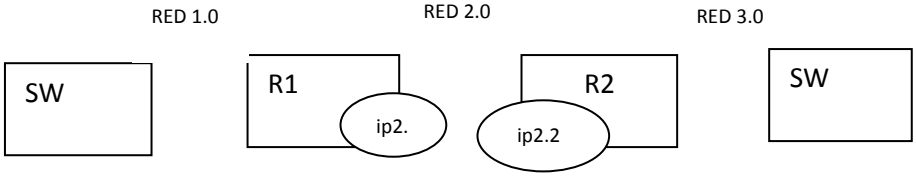
- Redes directamente conectadas
- Rutas estáticas
- Redes aprendidas por medio de un protocolo de enrutamiento.

Para copiar la configuración de Router, hacemos un Show Running-config y lo copiamos en texto plano

RUTAS ESTATICAS

Ruta Predeterminada: **La red 0.0.0.0 significa "Todas Las Redes" 2 elevado a 32. Su máscara será también la 0.0.0.0 y La Puerta de Enlace de esta Red será 10.0.0.1**

RUTA ESTATICA:



El router 1 sabrá llegar automáticamente a la red 1.0 y a la 2.0, pero no a la 3.0. Para que llegue a la 3.0 y la vea habrá que añadir una ruta a la tabla de enrutamiento. Hay dos maneras de hacerlo. Estática o automática.

Para añadir una ruta estática, se usa el comando IP ROUTE.

Hay que hacerlo paso a paso. Para que el router R1 vea la red 3 seria:

R1(Config)#ip route 192.168.30(dirección de red a la que queremos que llegue) + la máscara de la red 255.255.255.0 + indicar por donde tiene que llegar que en este caso sería por el router 2. Esto se puede hacer de dos formas:

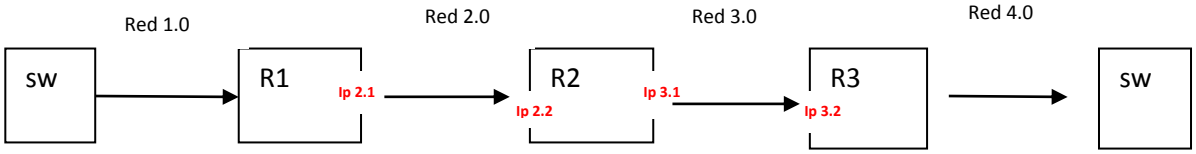
1- **Poniendo la IP de la interface del router de destino, en este caso 192.168.2.2**
2- **Poniendo la interface de salida del router de salida, en este caso, ser 2.0**

Así, en el caso 1 seria:R1(Config)#ip route 192.168.3.0 255.255.255.0 192.168.2.2

Y en el caso 2 seria: R1(Config)#ip route 192.168.3.0 255.255.255.0 ser 2/0

Lo más usual es la 1ª

Si hay más de un router por medio, habría que ir router por router añadiendo el camino. Salto a salto



De tal forma que si queremos que el router 1 vea la red 4.0, iríamos primero a la configuración del router 1 y la daríamos su ruta. El router 1 ve al 2 pero en la tabla de enrutamiento le ponemos la ip de destino que está en la red 4.0, y así el R2 sabe que si viene algo desde el R1 con esa ip. Debe dejarlo pasar. Después en el router 2 habrá que darle también su configuración.

Así quedaría:

R1(Config)#ip route 192.168.4.0(red de destino)255.255.255.0(mascara de destino) 192.168.2.2(ip de la interface de R2). Una vez que el R1 está hecho, iríamos al dos a seguir la ruta.

R2(Config)#ip route 192.168.4.0(el destino se mantiene)255.255.255.0+ ip de la interface de R3 192.168.3.2

Un PC podría funcionar como un router. Con el comando Router print en cmd vemos la tabla de enrutamiento de un PC

Con el comando **#debug ip packet** muestra el tráfico en tiempo real

Con el comando **#debug ip routing** muestra lo que ocurre en la tabla de enrutamiento. Cuando ponemos este comando se activa. Hasta que no pongamos el comando **# no debug ip routing** lo seguirá haciendo.

Configuración de un SW

Un SW aunque sea de capa 2 se le puede asignar una sola IP y una Puerta de Enlace. Esto se hace configurándolo en modo virtual SVY (1 solo si es capa 2 y varios si es capa 3)

VLAN-1

SW>enable

SW#conf t

SW(Config)#interface vlan 1

SW(Config-if)#ip address ….+ mask

+ no shutdown

Y para la default Gateway comando: **ip default-gateway ip** sin mask o

Cada SW tendrá su propia ip de default-gateway

Al La Vlan de administración también se le otorga una

Para copiar configuraciones a un servidor TFTP desde raíz **#copy startup-config tftp**:

Aquí nos pedirá la IP del servidor TFTP, se la damos y después nos preguntará que cómo queremos que se llame el archivo.

De esta forma podremos grabar/copiar la running-confif o lo que queramos.

Una vez que sabemos que lo tenemos copiado en el servidor TFTP y lo **quisiéramos recuperar**, el proceso sería: desde raíz **#copy tftp starup-config** Una vez hecho esto, nos preguntarála ip del servidor TFTP y después el nombre del archivo que queremos copiar.

Por último preguntará dónde lo queremos guardar.

Clases de redes públicas

Clase A: desde 0-127 /8 Mask: 255.0.0.0 2^8 redes y 2^{24} equipos

Clase B: desde 128-191 /16 Mask: 255.255.0.0 2^{16} redes y 2^{16} equipos

Clase C: desde 192-223 /24 Mask: 255.255.255.0 2^{24} redes y 2^8 equipos

Clase D: desde 224-239 Grupo multicast

Clase E: desde 240-255

Hay un pequeño grupo de ***direcciones privadas NO enrutables** tomadas de la división anterior. Se llaman RFC 1918:

- **De la Clase A: 10.0.0.0 /8**
- **De la Clase B: 172.16.0.0 /16 hasta la 172.31.0.0 /16**
- **De la clase C: 192.168.0.0 /24 hasta la 192.168.255.0 /24**

Una dirección enrutable es pública

Una dirección privada no es enrutable.

NAT traduce direcciones privadas en públicas. NAT y RFC 1918 van de la mano

/8 = 255.0.0.0 implica 2^8 redes y lo demás hosts Clase A: R.H.H.H

/16 = 255.255.0.0 implica 2^{16} redes y lo demás hosts Clase B: R.S.H.H

/24 = 255.255.255.0 implica 2^{24} redes y el resto a hosts Clase C: R.R.S.H

Direccionamiento IP especial:

0.0.0.0 /8 Identifica a un equipo sin IP

127.0.0.0 /8 Prueba la pila TCP/IP de la máquina sin salir (bucle interno para ver que funciona)

169.254.0.0 /16 Dirección A PIPA: Autoconfiguración del host. Sucede cuando la IP no se ha configurado ni manualmente ni por DHCP

255.255.255.255 /32

Conversión BINARIO/DECIMAL/HEXADECIMAL

Los 8 Bits de un octeto pueden estar a 0 ó a 1. Según la posición que estén tienen un valor distinto y su valor decimal es la suma de todos los bits que estén a 1

$$2^7 \times 0/1 + 2^6 \times 0/1 + 2^5 \times 0/1 + 2^4 \times 0/1 + 2^3 \times 0/1 + 2^2 \times 0/1 + 2^1 \times 0/1 + 2^0 \times 0/1$$

O lo que es lo mismo: Si el valor del bit está a 1, según su posición, sumarle el correspondiente valor: 128 + 64 +32 +16 +8 + 4 +2 + 1

Hexadecimal:

Hexadecimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Los valores son los que aparecen en el cuadro de tal forma que A=10 B=11 C=12, etc…

Si tenemos un número binario, para pasarlo a hexadecimal se divide el binario en dos de 4

11010110 de tal forma que la primera parte sería: 1101= 13=D y 0110=6 **D6**

Otro ejemplo:

1111 0001: primera parte: 1111=15=**F** segunda parte:0001=**1** **resultado: F1**

1101 0011: primera parte: 1101=13=**D** segunda parte:0011=**3** **resultado D3**

0010 1011: primera parte: 0010=**2** segunda parte:1011=11=**B** **resultado 2B**

Hexadecimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Mascara de red:

Para saber a qué red pertenece una IP multiplicamos según el patrón de la tabla AND la IP con su máscara, de tal forma que quedaría:

IP: 192.168.10.14

Máscara: 255.255.255.0

Tabla de verdad puerta AND

Entrada A	Entrada B	Salida AB
0	0	0
0	1	0
1	0	0
1	1	1

192.168.10.14 en binario sería:

11000000.10101000.00001010.00001110

Y la mascara 255.255.255.0:

11111111.11111111.11111111.00000000

Así se multiplica uno a uno entre sí la ip con la máscara y el resultado nos dará la red

SUBNETING

SLSM: máscara de red de longitud estática

VLSM: máscara de red de longitud variable

CIDR: sumarización

SLSM: máscara de red de longitud estática

Del 0 al 255 hay 256 IPs. Siempre una de ellas pertenecerá a la red y otra al broadcast. De tal forma que:

En la red 172.32.0.0 /16 implicará que la dirección 172.32.0.0 será la dirección de red y la 172.32.255.255 será la dirección de broadcast

En la red 91.0.0.0 /8 implicará que la dirección de red sea la 91.0.0.0 y la de brodcast sea la 91.255.255.255

Cuando se hace subneting, se toman bits de la parte que pertenece a los hosts, esto es:

De la dirección 192.168.1.HHHHHHHH si tomamos el primer bit de la parte de hosts tendremos: 2^1 redes. Es 2^1 porque es 1 bit el que se ha tomado para redes=2 redes y cada una de ellas con 2^7 hosts=128 hosts. Es 2^7 hosts porque quedan 7 bits de host.

Con esto habremos conseguido 2 redes y cada una con 128 IPs

Otro ejemplo: En esta ocasión vamos a tomar 5 bits de hosts para conseguir más redes.

192.168.1.HHHHHHHH

Las redes conseguidas serán 2^5=32 subredes y cada una de ellas con 2^3 =8 IPs disponibles, menos las dos de red y broadcast 8-2=6 IPs disponibles en cada red.

Este ejemplo además supondría que pasaría de ser un /24 a ser un /29 por que se tienen 5 bits mas que la /24 de tal forma que su máscara sería: 255.255.255. 128+64+32+16+8=248

Mas ejemplos de subneting

Recordamos que los bits red están todos a 1 (255) y los de equipo a 0.

Ejemplo de subneting

De la red 8.0.0.0 se pide:

- 32 subredes con la mayor cantidad de direcciones
- Hayar la subred numero 27

Es una red de clase A, por lo que tenemos 8 bits para red y 24 para hosts. Como necesitamos hacer 32, necesitaremos 2^5 direcciones para redes=32 por lo que quedan 2^{19} hosts=524288

8.RRRRRHHH.HHHHHHHH.HHHHHHHH

Los incrementos entre red y red será de 2^3 porque estos son los bits que han quedado en ese octeto para hosts=8

La máscara de 248 sale de sumar el valor binario de los 5 bits que hemos tomado para las redes 128+64+32+16+8=248

Al aumentar las redes de 8 en 8, quedará de la siguiente forma:

1ª red: 8.0.0.0 hasta la 8.7.255.255 y con una máscara de /13 = 255.248.0.0

2ª red: 8.8.0.0 hasta la 8.15.255.255 y con una máscara de /13=255.248.0.0

3ª red: 8.16.0.0 hasta la 8.23.255.255 y con una máscara de /13 =255.248.0.0

La 27ª: multiplicamos ese valor menos 1 (27-1) por la variación o saltos de red (8). Así 26x8=208, con lo que la red quedará:

27ª red: 8.208.0.0 hasta la 8.215.255.255 /13

La ultima red será: como son 32 redes, 32-1=31 31x8 que es el salto de red=248 por eso la última red sderá:

Última red: 8.248.0.0 hasta la 8.255.255.255 /13

VLSM: MÁSACARA DE RE DE LONGITUD VARIABLE

Los pasos que se siguen en subneting son:

- Ver cuantos bits hacen falta para los hosts y tomarlos
- Ver cuantos bits de host están libres en el octeto de red para saber de cuanto será el salto entre redes teniendo en cuenta que siempre se suma uno menos.
- Ver la máscara que se tiene atendiendo a los bits de red que se toman
- Tener en cuenta que en cuanto a los hosts o direcciones finales, siempre hay que reservar dos: uno para la propia red y otra dirección para el broadcast
-

Práctica VLSM con una red de Clsase B

DIRECCION DE RED: 129.17.0.0

PREFIJO: /16

MASCARA DE RED: 255.255.0.0

SE REQUIERE:

1º => 1 subred de 32.000 direcciones para PC's

2º subred de 1.000 direcciones para portátiles

3º=> 1 subred 200 direcciones para equipos de red

4º 1 subred de 40 direcciones para servidores

5º 2 subredes de 2 direcciones para WAN

129.17.HHHHHHHHH.HHHHHHHHH

1º Lo primero que nos piden es una subred de 32.000 direcciones, por lo que necesitaremos 2^n para conseguir 32000 que entes caso es 2^{15}=32.768. Asique tomamos esos 15 bits para nuestros PCs (32.768-2=32.766 hosts)

129.17.SHHHHHHH.HHHHHHHH => /17 (255.255.128.0.0)

El tomar este bit para la red hace que la máscara cambie de /16 a /17 con lo que pasará de 255.255.0.0 a 255.255.128.0

Los bits sobrantes del octeto de donde hemos tomado el de la red para los hosts son 7, por lo que los saltos de red serán 2^7=128 con lo que la siguiente red será: 129.17.128.0

Así tenemos:

Subred 1ª: 129.17.0.0 /17 (255.255.128.0) hasta la 129.18.127.255

Y **la red 2ª**: 129.17.128.0 /17 (255.255.128.0) hasta la 129.17.255.255

2º Nos piden una subred de 1000 direcciones para portátiles.

Para esto tomaremos como **raíz la red 2ª 129.17.128.0** /17 255.255.128.0

Como necesitamos 1000 direcciones para los portátiles, necesitaremos 10 bits para los hosts 2^{10}=1024 por eso el dibujo de la red sería: (1024-2=1022 hosts)

129.17.**R**SSSSSHH.HHHHHHHH => La máscara pasa de /17 a /22 porque hemos tomado 5 bits más para red (255.255.252.0)

Como los bits sobrantes del octeto de donde tenemos nuestra subred son 2, los saltos irán de 2^2=4

Así **la red 2.1** sería 129.17.128.0 hasta la 129.17.131.255 /22 255.255.252.0

La **red 2.2** sería 129.17.132.0 /22 255.255.252.0

3º Ahora queremos una subred para 200 equipos de red.

Tomamos como **raíz la red 2.2 129.17.132.0** /22 255.255.252.0

Como necesitamos 200 direcciones necesitaremos 2^8=256, por lo que el dibujo de la red quedaría: (256-2=254 hosts)

129.17.**RRRRRR**SS.HHHHHHHH => la máscara pasará de /22 a /24 porque hemos tomado 2 bits más para nuestra subred de tal forma que la máscara quedaría: 255.255.255.0

Como los bits sobrantes del octeto de red han sido 0 los saltos de red serán 2^0=1

Así **la red 2.2.1** sería de 129.17.132.0 hasta 129.17.132.255

La **red 2.2.2** quedaría: 129.17.133.0 hasta la 129.17.133.255

4º Ahora nos piden 40 direcciones para servidores.

Tomamos como **raíz la red 2.2.2 129.17.133.0** /24 255.255.255.0

Como nos piden 40 direcciones, nos harán falta 2^6=64 (64-2=62 hosts) de tal forma que el dibujo de la red sería:

129.17.**RRRRRRRR**.SSHHHHHH => La máscara pasará a ser de /24 a /26 al tomar dos bits más para red (255.255.255.192)

Los saltos de red serán 2^6=64 porque esos son los bits que quedan para hosts.

Así **la red 2.2.2.1** quedaría: 129.17.133.0 hasta la 129.17.133.63 con 64-2 =62 hosts

Y **la red 2.2.2.2** quedaría: 129.17.133.64

5º Por último nos piden 2 subredes de 2 direcciones.

Tomamos como **raíz la red 2.2.2.2 129.17.133.64** /26

Como solo nos piden 2 hosts por subred, nos hará falta 2 bits para hosts 2^2=4 ya que hay que tener en cuenta que dos hosts van a ser para la red y para broadcast, de tal forma que necesitaremos 2^2=4-2=2. Así el dibujo de la red sería:

129.17.**RRRRRRRR.RR**SSSSHH => /30 (255.255.255.252)

Como los bits sobrantes del octeto han sido 2, 2^2=4 serán los saltos de red.

Así las 2 redes con dos hosts para nuestra red WAN serán:

Red 2.2.2.2.1: 129.17.133.64 hasta la 129.17.133.67 /30

Y la **red 2.2.2.2.2** 129.17.133.68 hasta la 129.17.133.71 /30

Práctica VLSM con una red de Clsase A

DIRECCION DE RED: 12.0.0.0

PREFIJO: /8

MASCARA DE RED: 255.0.0.0

SE REQUIERE:

1º 2 subred de 16.000 direcciones para PC's

2º 4 subred de 1.600 direcciones para portátiles

3º 4 subred de 500 direcciones para equipos de red

4º 8 subred de 8 direcciones para servidores

5º 2 subredes de 2 direcciones para WAN

El mapa original de la red es este:

12.HHHHHHHH.HHHHHHHHH.HHHHHHHHH

1º: Nos piden 2 subredes de 16000 direcciones para PCs, así que necesitaremos 2^{14}=16384 lo que implica robar 14 bits a los destinados a hosts, de tal forma que el mapa sería:

12.SSSSSSSS.SSHHHHHH.HHHHHHHH => Y la máscara pasará de ser una /8 a una /18 (255.255.192.0.0)

Como los bits que quedan para hosts son 6, los saltos de red serán de 2^6=64

De tal forma las 2 redes que nos piden serán las:

1ª red: 12.0.0.0 hasta la 12.0.63.255

2ª red: 12.0.64.0 hasta la 12.0.127.255 con máscaras 255.255.192.0 /18

La 3º red que es la que tomaremos como raíz para los siguientes pasos será la:

12.0.128.0 hasta la 12.0.191.255 /18 255.255.192.0

2º Ahora nos piden 4 subredes de 1600 direcciones

Partimos de la red raíz 12.0.128.0.

Para conseguir 1600 direcciones nos hacen falta 2^{11}=2048 por lo que tomamos 11 bits a los destinados a hosts de tal forma que el mapa de red quedaría:

Estos son los 11 bits 2^{11} para poder conseguir los 1600hosts

12.**RRRRRRRR.RR**SSSHHH.HHHHHHHH => /21 (255.255.248.0)

Pasaría de ser un /18 a un /21 y con una máscara 255.255.248.0

Los saltos de red serán de 2^3=8 por que esos son los bits que han quedado en el octeto de red.

Así las 4 subredes que nos piden serían:

3.1/ 1º: 12.0.128.0 hasta la 12.0.135.255 con una máscara /21 255.255.248.0 y cada una con 16-2 hosts=14

3.2/ 2ª: 12.0.136.0 hasta la 12.0.143.255

3.3/ 3ª: 12.0.144.0 hasta la 12.0.151.255

3.4/ 4ª: 12.0.152.0 hasta la 12.0.159.255

Estas serían las 4 subredes que nos piden.

La 5ª es la que usaríamos como raíz para el siguiente paso y sería la:

La **3.5**/ 12.0.160.0 hasta la 12.0.167.255

3º Nos piden 4 subredes para albergar a 500 PCs.

Para poder llegar a 500 PCs nos hacen falta 2^9=512 (que serán 510 al restarle los de red y broadcast) Así que tomamos 9 bits para hosts de tal forma que el mapa quedaría:

12.**RRRRRRRR.RRRRR**SSH.HHHHHHHH => y la máscara pasaría de ser una /21 a ser una /23 (255.255.254.0)

Como en el octeto de red solo ha quedado un bit para host, los saltos serán de 2^1=2

De tal forma que las 4 subredes que nos piden para los 500 PCs serían:

1ª La 3.5.1: De la 12.0.160.0 hasta la 12.0.161.255 /23 y máscara 255.255.254.0

2ª La 3.5.2: De la 12.0.162.0 hasta la 12.0.163.255

3ª La 3.5.3: De la 12.0.164.0 hasta la 12.0.165.255

4ª La 3.5.4: De la 12.0.166.0 hasta la 12.0.167.255

La siguiente subred sería la raíz para las siguientes y sería la **3.6**:

12.0.168.0 hasta la 12.0.175.255

4º Ahora nos piden 8 subredes para 8 direcciones de servidores.

Como necesitamos 8 direcciones y sabemos que en todas las redes dos direcciones están reservadas para dirección de red y de broadcast, en vez de hacernos falta 2^3=8, nos hará falta 2^4=16, de tal forma que robamos 4 bits a los hosts, quedando un mapa de red:

12.**RRRRRRRR.RRRRRRRS**.SSSSHHHH => /28 (255.255.255.240)

Como los bits que han quedado en el octeto de red son 4, los saltos de red serán de 2^4=16

Así, las 8 direcciones de red que nos están pidiendo serán:

3.6.1/ 1º: Desde la 12.0.168.0 hasta la 12.0.168.15

3.6.2/ 2ª: Desde la 12.0.168.16 hasta la 12.0.168.31

3.6.3/ 3ª Desde la 12.0.168.32 hasta la 12.0.168.47

3.6.4/ 4ª: Desde la 12.0.168.48 hasta la 12.0.168.63

3.6.5/ 5ª: Desde la 12.0.168.64 hasta la 12.0.168.79

3.6.6/ 6ª: Desde la 12.0.168.80 hasta la 12.0.168.95

3.6.7/ 7ª: Desde la 12.0.168.96 hasta la 12.0.168.111

3.6.8/ 8ª: Desde la 12.0.168.112 hasta 12.0.168.127

La 9ª red sería la 12.0.168.128 hasta la 12.0.168.143

5º Por último nos piden dos subredes con dos direcciones para WAN. Teniendo en cuenta lo de los hosts, necesitaremos 2^2=4 para los hosts. Así, el mapa de red quedaría:

12.**RRRRRRRR.RRRRRRRR.RRRR**SSHH => /30 (255.255.255.252)

Como solo quedan dos bits de host, los saltos de red irán de 2^2=4

Como la red raíz en este caso es la 12.0.168.128, las dos redes que nos piden serán

1ª De la 12.0.168.128 hasta la 12.0.168.131

2ª Y la 12.0.168.132 a la 12.0.168.135 cada una de ellas con 2 direcciones disponibles y con una máscara /30 255.255.255.252

CDIR: SUMARIZAR

Es al revés que subnetear. Cuando hay dos redes parecidas, se sumarizan en los procesos de enrutamiento, cambiándoles también la máscara

Ejemplo:

Redes: 172.16.0.0 ********.00010000.********.********

172.17.0.0 ********.00010001.********.********

172.18.0.0 ********.00010010.********.********

En este caso como el primer octeto lo tienen igual, no hay porque comprarlo. En el segundo que ess donde está la variación, lo recorremos hasta trazar una línea imaginaria para ver hasta donde coinciden todos sus bits. En este caso, vemos que coinciden en valor 1 todos los bits de valor 16, por lo que la red resultante será la 172.16.0.0

Ese 1 tiene un valor de 16

La red originaria era de clase B y tenía una máscara /16 pero al ser sumarizada, la máscara en este caso será /14 porque es solo hasta el bit 14 hasta donde coinciden. A esto se le llaman super redes.

Más ejemplos:

192.168.1.0 /22 Será una super red al estar sumarizada. (Lo delata la máscara /22)

172.16.10.0 /17 Será una Sub red al estar subneteada, ya que su máscara es /17 en vez de /16

10.8.8.0 /16 será también una sub red al tener /16 y ser una clase A que tienen /8

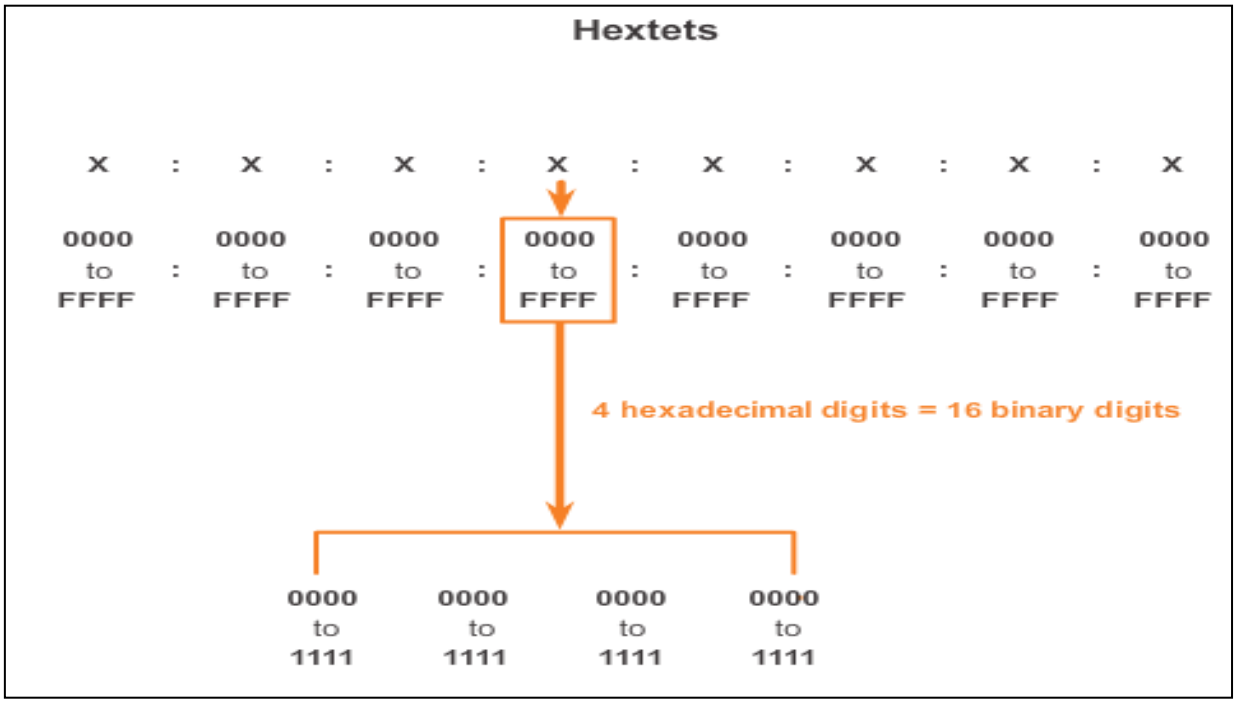
192.168.1.0 /24 sería una red ya que su máscara la tiene propia.

Capitulo 8: Direccionamiento IP

Broadcast dirigido: lanzar un ping a una dirección de broadcast de una subred, no a todos. Ej: ping a 172.16.4.255 /24 Este ping será solo dentro de la red .4

Link-local: parecido al broadcast dirigido pero limitado a la red 224.0.0.0 a la 224.0.0.255. Esta red NO es Enrutable.

IPV6: 8 bloques de 16 bits



De momento coexiste con ipv4.

Dual-Stack es un sistema que permite a un dispositivo coexistir con ambos protocolos: IPV4 y IPV6.

Una interface de un router moderno puede albergar en la misma interface IPV4 y IPV6.

ARP no funciona con IPV6

Dos dispositivos con distintos protocolos no pueden comunicarse.

Las direcciones unicast en IPV6 pueden ser de tres tipos

- **Unicast Global**: desde la 2000 hasta la 3FFF
- **Link-local** desde la FE80 a la FEBF (esta IP es anivel de segmento, no es enrutable y funciona parecido a la MAC)
- Unique-local:

Las direcciones Multicast son las FF00

La puerta de enlace de un host es la link-local de ese segmento.

Por interface puede haber múltiples direcciones Unicast Global, pero solo una Link-local

Habrá que poner una link-local diferente por cada segmento. No todas son FE80::1, también se usará la FE80::2,3,4,5,6,7. Lo habitual es poner una link-local por Router.

Los host con IPV6 tienen todos 2 IPs: una global y otra local-link.

En la dirección unicast global (de la 2000 a la 3FFF):

2001:DB8:ACAD:OOC8::/64

La parte en azul es la parte inmutable ya que viene dada por el proveedor de internet.

La parte en rojo es la parte copn la que jugamos nosotros y vamos poniendo distintas direcciones a las distintas LANs que tengamos.

La parte en verde que son 64 bits es la parte de host. Así esta parte para poder usarla, habrá que añadirle el valor de IP, porque si la dejáramos así, sería una dirección de red al terminar en 0. De tal forma, que una dirección para una interface o un equipo seria:

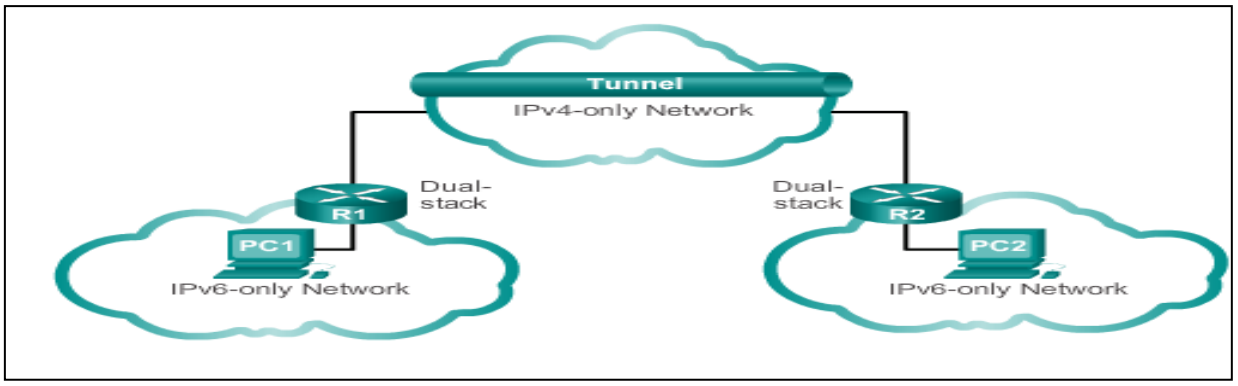
2001:DB8:ACAD:OOC8::1/64

La siguiente dirección de red sería:

2001:DB8:ACAD:OOC9::0/64

Cuando una IPV6 termina en ::=0. Lo que significa que está haciendo referencia a una red. Una IP de host nunca podrá ser terminada en :: ya que eso haría referencia a red. Para ser de host sería, :1 :2 :3…..

Tunnelling



Es un encapsulado de capa 3 para que dos dispositivos que hablan IPV6 pueden viajar por internet que todavía trabaja en IPV4

R1 Ipv6---------- INTERNET Ipv4 ----------R2 Ipv6

NAT64 es una especie de traducción y lo único que hace es sobrecargar la red

En una dirección Ipv6, se puede abreviar:

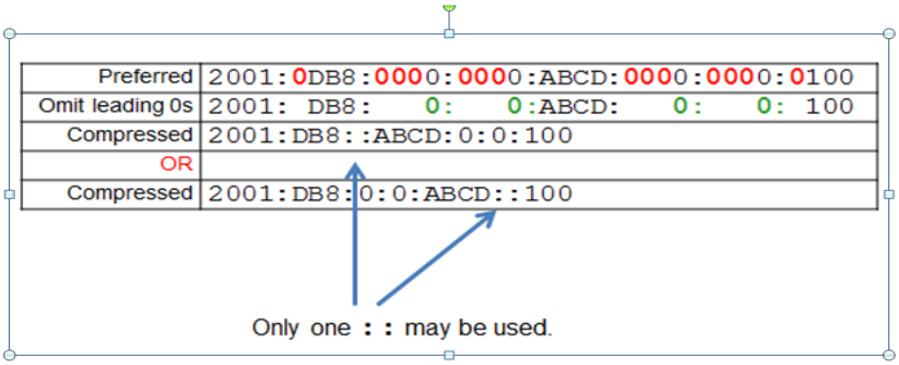
- Cuando tiene 0 a la izquierda, se suprime

- The first rule to help reduce the notation of IPv6 addresses is any leading 0s (zeros) in any 16-bit section or hextet can be omitted.
- 01AB can be represented as 1AB.
- 09F0 can be represented as 9F0.
- 0A00 can be represented as A00.
- 00AB can be represented as AB.

Preferred	2001:0DB8:000A:1000:0000:0000:0000:0100
No leading 0s	2001: DB8: A:1000: 0: 0: 0: 100
Compressed	2001:DB8:A:1000:0:0:0:100

- Cuando todo el hexteto está a 0 se ponen solo los dos puntos:
- Esta regla de poner solo los dos puntos soolo se puede repetir una vez:

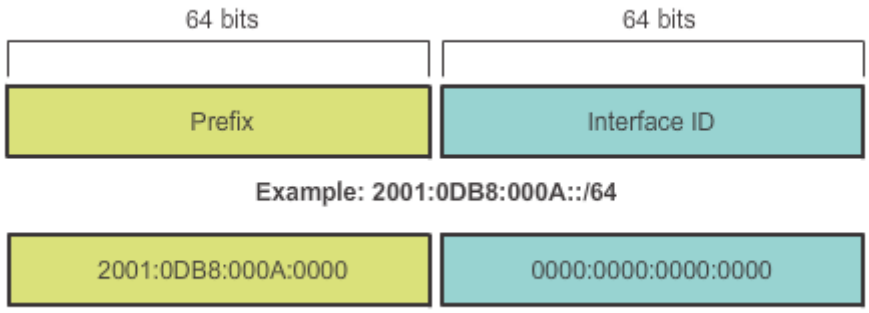
Example #1



Example #2

Preferred	FE80:0000:0000:0000:0123:4567:89AB:CDEF
Omit leading 0s	FE80: 0: 0: 0: 123:4567:89AB:CDEF
Compressed	FE80::123:4567:89AB:CDEF

En una dirección Ipv6 los primeros 64 bits son de red y los 64 siguientes son de host



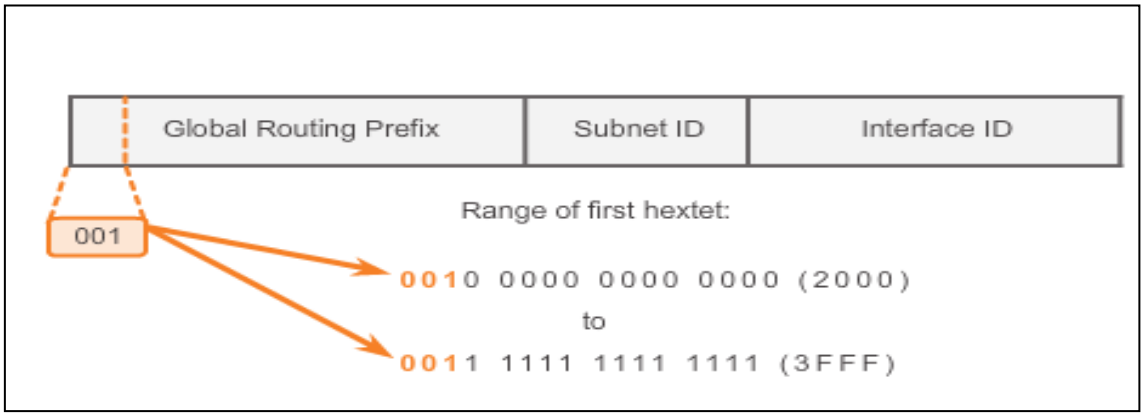
Con Ipv6 también tenemos ping Unicast, Multicast y uno que se llama Any cast, que es hacer ping al dispositivo más cercano al origen.

Link-local en Ipv6 se utiliza para comunicar direcciones que están dentro del mismo segmento. En Ipv6 una interface puede albergar a distintas LANs.; una Link-Local solo comunica equipos de dentro de una de esas múltiples LANs que puede haber dentro de una misma interface.

La dirección de Link-Local siempre empieza por FE80: y NO es Enrutable.

Slaac: Es una especie de auto dhcp: El host se autocompleta su dirección una vez que se le ha fdado la de red.

En las direcciones Ipv6 solo son enrutables las direcciones comprendidas entre 2000 y 3FFF



Ipv6 unicast-routing es el comando el ipv6. Al habilitarlo el Router empieza a mandar RAs para que los equipos se vayan autoconfigurando. Los equipos por su parte también pueden enviar solicitudes RSs.

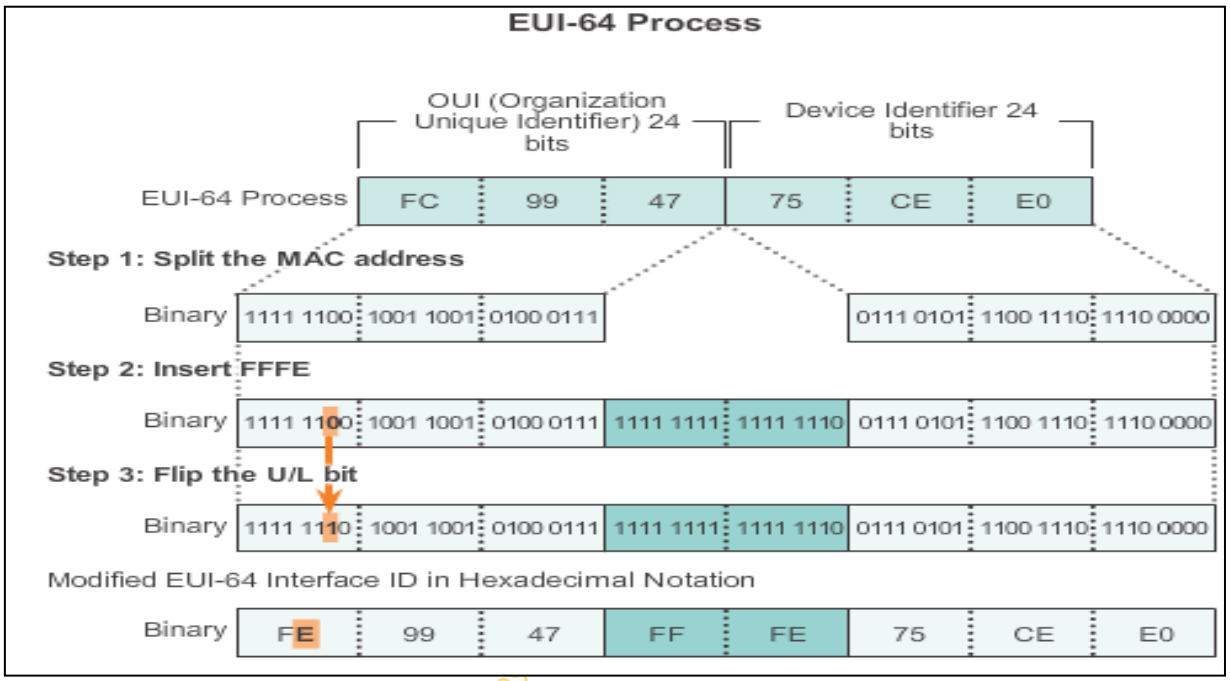
Cuando el router manda la información, puede enviar 3 tipos de RAs:

- Slaac only
- Slaac y Dhcp
- Dhcp only

- The **`IPv6 unicast-routing`** command enables IPv6 routing.
- RA message can contain one of the following three options:
  - SLAAC Only – Uses the information contained in the RA message.
  - SLAAC and DHCPv6 – Uses the information contained in the RA message and get other information from the DHCPv6 server, stateless DHCPv6 (for example, DNS).
  - DHCPv6 only – The device should not use the information in the RA, stateful DHCPv6.
- Routers send ICMPv6 RA messages using the link-local address as the source IPv6 address

Proceso EUI-64

Una MAC tiene 48 bits. Se divide la MAC en 2 de 24 y se meten en medio los 16 bits FFFE para tener 64. Después el 7º bit siempre se pone a uno




```
R1#show interface gigabitethernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is fc99.4775.c3e0
(bia fc99.4775.c3e0)
<Output Omitted>

R1#show ipv6 interface brief
GigabitEthernet0/0       [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:ACAD:1::1
GigabitEthernet0/1       [up/up]
    FE80::FE99:47FF:FE75:C3E1
    2001:DB8:ACAD:2::1
Serial0/0/0              [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:ACAD:3::1
Serial0/0/1              [administratively down/down]
    unassigned
R1#
```

Link-local addresses using EUI-64

FE80: = Link-local

Una IP se puede Autoconfigurar con EUI64:, lo que nos dará una IP con el hexteto: FFFE

Global Unicast es la dirección enrutable: también se pueden configurar la EUI64

Con EUI64 se configura la IP entorno a la MAC.

En IPV6 al configurar un Router siempre habrá 2 Ipv6:

- 1º La link-local para comunicarsae dentro de ese segmento
- 2º la global para comunicarse entre redes

La link local será un /10

La global será un /3

Una L /128 nos indica que es una IP dentro de la red

Las IPs de multicast más usadas son:

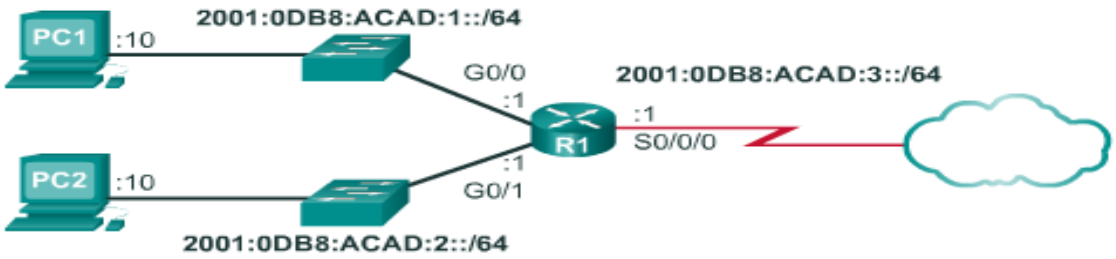
- FF02::1
- FF02::2

Ipv6 unicast-routing es es el comando para empezar a mandar RAs. La dirección de origen que manda en Router es siempre la de Link-local.

Siempre que se crea una ip global hay que configurar una local, sino, lo hará automáticamente.

Una vez que tiene asignada la ip el host se manda por DAD una consulta vecinal por si alguien más tuviera la misma dirección que acaba de obtener ese host. Esto se hace para evitar la duplicidad.

Configuracion IP global:




```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ipv6 address 2001:db8:acad:1::1/64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ipv6 address 2001:db8:acad:2::1/64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ipv6 address 2001:db8:acad:3::1/64
R1(config-if)#clock rate 56000
R1(config-if)#no shutdown
```

Configuracion IP Local-Link

```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ipv6 address fe80::1 ?
  link-local  Use link-local address

R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#
```

Transporte

Los datos siempre se segmentan al pasar de la capa de aplicación a la de transporte.

Se identifica la aplicación adecuada gracias al número de puerto.

El encabezado de TCP siempre ocupará 20 Bytes y el de UDP 8

TCP usa acuse de recibo y UDP no. Por eso UDP tiene menos sobrecarga.

Una aplicación puede haber sido programada para UDP o para TCP.

UDP está orientado a conexión.

Entre dispositivos UDP, más ligero y rápido pero menos seguro y confiable, se limitan los dispositivos a:

- El saludo de 3 vías
- Intercambio de información
- Finalización de la comunicación

El acuse de recibo que hace TCP se llama ACK

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65533	Private and/or Dynamic Ports

Registered TCP Ports:
1863 MSN Messenger
2000 Cisco SCCP (VoIP)
8008 Alternate HTTP
8080 Alternate HTTP

Well Known TCP Ports:
21 FTP
23 Telnet
25 SMTP
80 HTTP
110 POP3
194 Internet Relay Chat (IRC)
443 Secure HTTP (HTTPS)

Registered UDP Ports:
1812 RADIUS Authentication Protocol
5004 RTP (Voice and Video Transport Protocol)
5040 SIP (VoIP)

Well Known UDP Ports:
69 TFTP
520 RIP

Registered TCP/UDP Common Ports:
1433 MS SQL
2948 WAP (MMS)

Well Known TCP/UDP Common Ports:
53 DNS
161 SNMP
531 AOL Instant Messenger, IRC

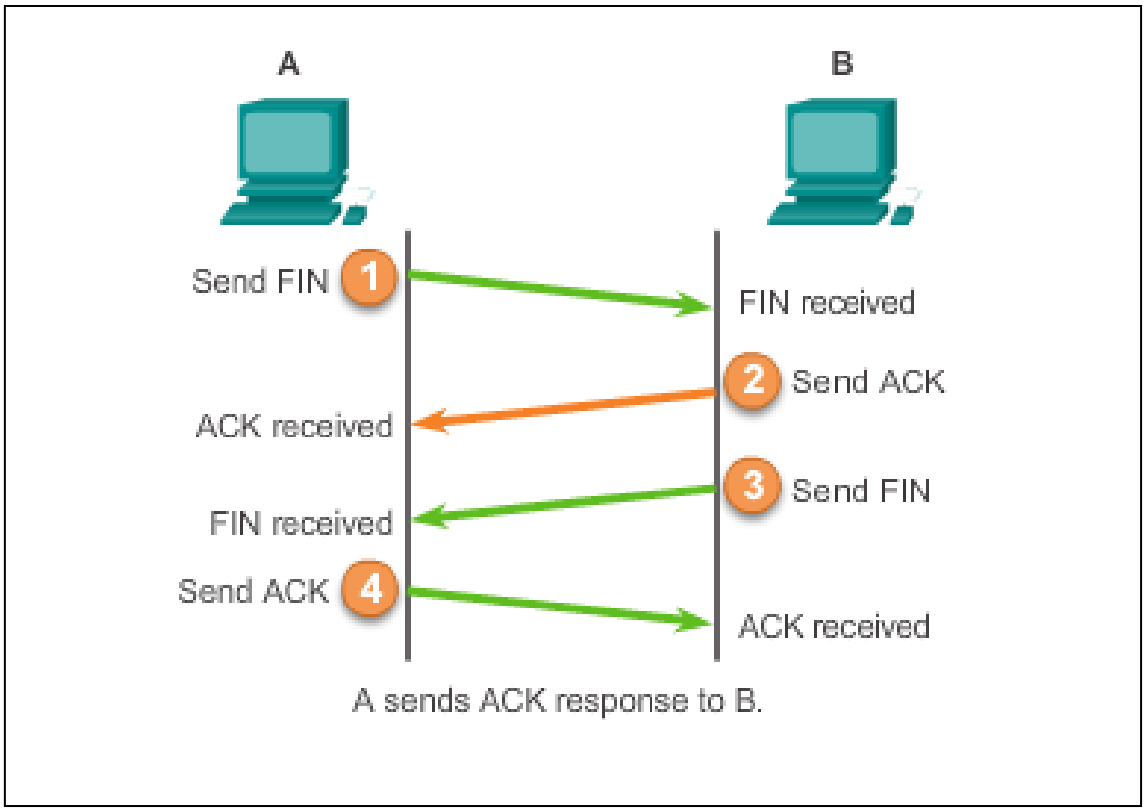
Cuando se realizan varias peticiones a puertos distintos se denomina muliplexado

El protocolo TCP, como necesita acuse de recibo, los procesos que sigue para conseguirlo son:

SYN: Intenta sincronizar

SYN-ACK: Respuesta a esa sincronización

ACK: Se le dice al receptor que se ha oido su respuesta afirmativa



FTP utiliza TCP

Balanceo de carga: todos los equipos/servidores funcionan a la vez por si se cae uno los demás asumen el servicio con más carga.

Tolerancia a fallos: Solo si falla el principal, entra en funcionamiento el segundo.

Protocolos de mantenimiento a distancia:

TELNET, SSH, HTTP, HTTPS, SNTP, RMON

```
Router(config)#service password-encryption
Router(config)#security password min-length 8
Router(config)#login block-for 120 attempts 3 within 60
Router(config)#line vty 0 4
Router(config-vty)#exec-timeout 10
Router(config-vty)#end
Router#show running-config
-more-
!
line vty 0 4
 password 7 03095A0F034F38435B49150A1819
 exec-timeout 10
 login
```

Comandos de seguridad:

R(Config)#service password-encryption encripta todas las contraseñas login

R(Config)#security password min-length 8 obliga a que las contraseñas tengan un mínimo en este caso de 8 caracteres

Dentro de line vty 0 4 el comando **exec-timeout 10** hace que en 10 minutos se autosalga del modo virtual


```
R1# conf t
R1(config)# ip domain-name span.com
R1(config)# crypto key generate rsa general-keys modulus 1024
The name for the keys will be: R1.span.com
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
R1(config)#
*Dec 13 16:19:12.079: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)# username Bob secret cisco
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# exit
```

Step 1: Configure the IP domain name.
Step 2: Generate one-way secret keys.
Step 3: Verify or create a local database entry.
Step 4: Enable VTY inbound SSH sessions.

El comando **R(Config)#ip domain-name** + el nombre de dominio ej: span.com

El comando **R(Config)#crypto key generate rsa general-keys modulus 1024** es para encriptar los datos en la comunicación

```
R3(config)#crypto key generate rsa

The name for the keys will be: R3.ccnasecurity.com

Choose the size of the key modulus in the range of 360 to 2048 for
your

  General Purpose Keys. Choosing a key modulus greater than 512 may
take

  a few minutes.


How many bits in the modulus [512]: 1024

% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

Con el commando **R(Config)#username Chus secret cisco** estamos creando un usuario que en este caso se llama chus y dandole una contraseña que en este caso es cisco. Así cuando nos vayamos a conectar por linea vty 0 4 nos pedirá usuario

Dentro de line vty 0 4 al poner login local le estamos diciendo la necesidad de logarse para acceder a esa linea vty 0 4 :

- **R(Config)#line vty 0 4**
- **R(Config-line)#login local**

Después con el commando **R(Config-line)#transport input ssh** estamos diciendo que para entrar en el modo line vty habrá que hacewrlo por ssh o también podría ser por telnet con el commando **R(Config-line)#transport input telnet**

Así los pasos son:

Antes de poner el comando **R(Config)#crypto key generate rsa,** hay que crear un dominio para que funcione la encriptación SSH. El Dominio se crea con el comando:

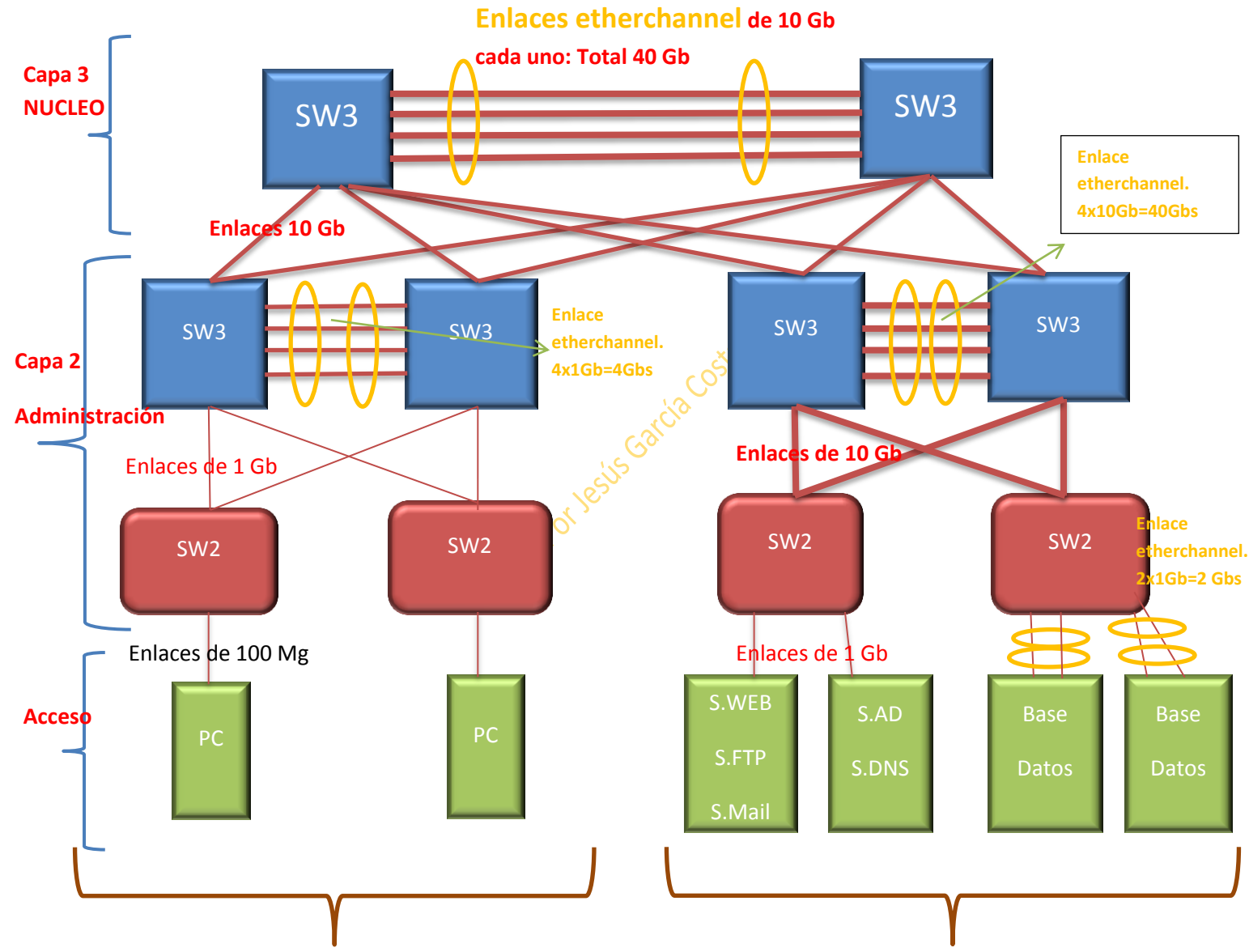
1- **R(Config)#ip domain-name +**costa54.com
2- Encriptamos la comunicación con **R(Config)#crypto key generate rsa general-keys modulus 1024**
3- Ponemos usuario y encriptamos: **R(Config)#username Chus secret cisco**
4- Llamamos a la linea vty 04 con **R(Config)#line vty 0 4**
5- Obligamos a que pida contraseña con **R(Config-line)#login local**
6- Obligamos a que se tenga que entrar por ssh con **R(Config-line)#transport input ssh**
7- También podría ser por telnet con el comando **R(Config-line)#transport input telnet**

Tema II: Ruting & Switching

Tipos de lenguaje para administración: HTTPS, SSH, SNMP3

Tipos de lenguaje para usuariuo (entre equipos): Ipsec, L2TP, L2F,PPTP

Una **red convergente** es aquella que soporta distintos tipos de tráfico: datos , voz y vídeo a la vez



La **REDUNDANCIA** es llegar al mismo sitios por distintos caminos

Los enlaces Etherchannel (2Cables) funcionan de manera lógica con la misma IP. Se comporta como un solo cable lógico. Si son 2 cables y cada cable soporta 1 Gb de datos, los enlaces serán de 2 Gbs

Entre Sws de igual capa también se usa etherchannel. 24 x 100 Mgs= 2,4 Gbs/Seg

Para tener más velocidad hay que poner más enlaces etherchannel. De lo que se trata es de conseguir: **Redundancia, alta disponibilidad, ancho de banda, resistencia ante fallos, jerarquía y flexibilidad**.

En el área de administración hay SWs que funcionan en capa 2 y 3. Un broadcast solo va por capa 2. En la capa del núcleo es donde se conectan los Routers.

Tablas CAM

MAC	PUERTO	VLAN	MODO
BBB	1	1	DINAMICO

Cada puerto de un SW es un dominio de colisión.

E principio y por definición, todos los puertos de un SW pertenecen a la VLAN 1. Por eso es bueno crear varias VLANs distintas a la 1 para administrarlas a nuestra conveniencia, y que no todos dependan de la 1.

Recuperación de contraseñas

La Bios de un Router es la ROMMON y la de un SW se llama BOOT SYSTEM

PROCEDIMIENTO DE RECUPERACION DE CONTRASEÑAS en un ROUTER

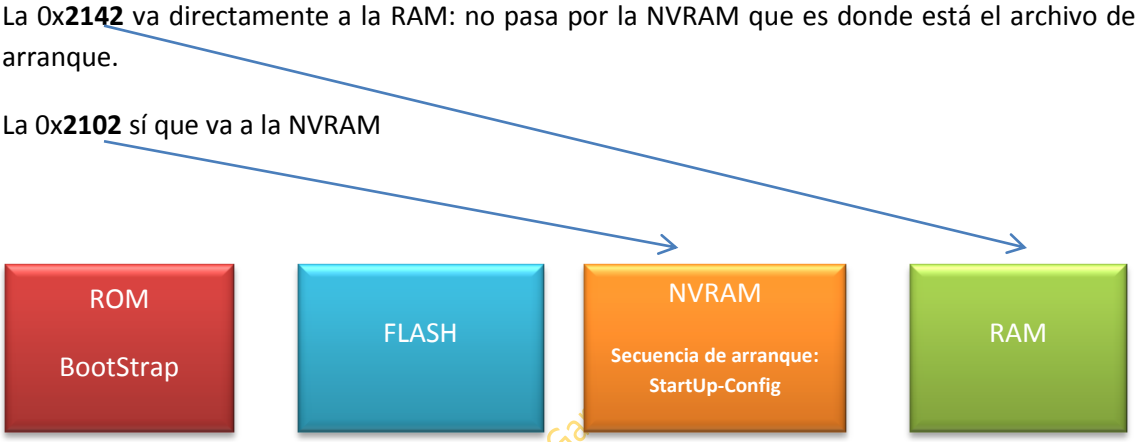
Paso 1: conectar al puerto consola
Paso 2: reiniciar el router
Paso 3: emitir la secuencia de escape Ctrl+break (pausa) para entrar al modo ROMmon
Paso 4: escribir el comando confreg 0x2142
Paso 5: escribir el comando reset y el router se reinicia
Paso 6: responder no a la pregunta de acceso a setup
Paso 7: escribir enable para acceder a modo privilegiado
Paso 8: escribir copy startup-config running-config
Paso 9: escribir show running, las contaseñas desencriptadas pueden seguir utilizandose, las encriptadas necesitan ser reconfiguradas
Paso 10: entra en el modo de configuración y configura una nueva enable secret
Paso 11: emite el comando no shutdown en las interfaces
Paso 12: escribe config-register 0x2102
Paso 13: guarda la configuración con el comando copy running startup

El comando confreg hace referencia al comportamiento del arranque del router. El comando que un router tiene por defecto es el 0x2102 que obliga al arranque as pasar por NVRAM, que es donde está grabada la contraseña que no recordamos.

Al cambiarle el comando 0x2102 por el de 0x2142 le estamos diciendo al Router que no pase por la NVRAM(que es donde está la secuencia de arranque) y que vaya directamente a la RAM (Running config), asi, una vez en la running, podremos volver a cambiar la contraseña, sin olvidarnos después de volver a poner el comando suoyo de origen, el 0x2102 para que una vez cambiada la contraseña, vuelva a hacer su proceso normal grabando su secuencia de arranque ya con la contraseña cambiada.

La 0x**2142** va directamente a la RAM: no pasa por la NVRAM que es donde está el archivo de arranque.

La 0x**2102** sí que va a la NVRAM



PROCEDIMIENTO DE RECUPERACION DE CONTRASEÑAS en un SWITCH

1- Quitamos la fuente de alimentación y lo volvemos a enchufar presionando a la vez el botón “mode” durante unos 15 seg
2- Cambiará el prompt al modo “boot-system”
   2.1 Escribimos el comando: flash_init
   2.2 Comando load_helper (para cargar los comandos de ayuda)
   2.3 Escribimos Dir flash para ver los directorios que hubiere
   2.4 Escribimos el comando: **rename flash:config.text flash:config.old**
   (Config text es el nombre que da el SW a su startUp-Config). Lo que estamos haciendo con este comando es creando “otro” archivo de arranque que no tiene la contraseña y que no recordamos)
3- Reset + yes
4- Ante la pregunta que salga, respondemos que NO, y estaremos en el modo enable.
   Estando aquí, antes de poner la contraseña nueva, volvemos a renombrar el archivo flash:

   rename flash:config.old flash:config.text

5- Ya cambiado el nombre a su original, copiamos su config.text (equivalente al starUp-Config) a la Ram con el comando: **copy flash:config.text system:running-config**
6- Ponemos la contraseña nueva

Un tfno. IP o una cámara IP se puede alimentar por un puerto RJ45 de un SW. Por eso hay puertos RJ45 que tienen esa capacidad: alimentar tfnos y cámara IP incluso PCs. Tener puertos cono Power-Ethernet sería un valor añadido que tiene un SW.

Administración de un SW:

Una VLAN puede ser hasta la 99.

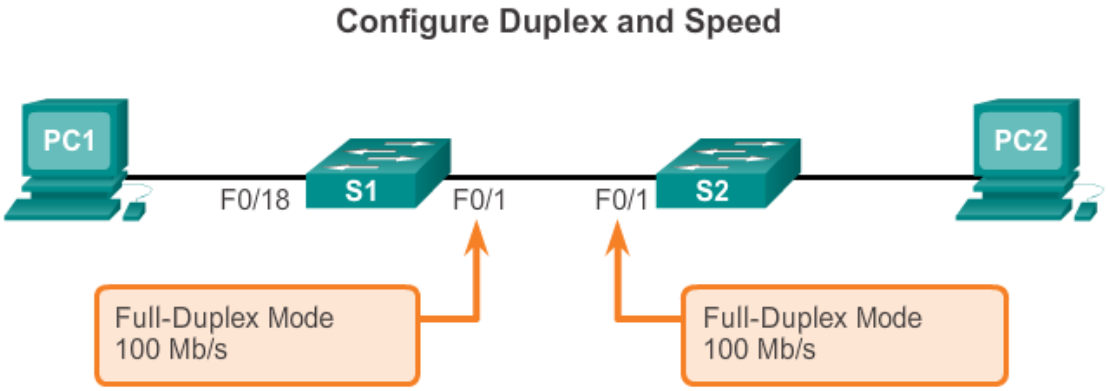
Creamos la VLAN 99 (SUI) virtual.

Una VLAN es equivalente a decir una red o una subred. Cada VLAN es un dominio de broadcast por lo que cada VLAN es como si fuera una red distinta. Dentro de un Sw podemos crear la VLANs que queramos. Podemos hacer por ejemplo: 4 VLANs 99 y le damos una IP. Esas 4 bocas pertenecerán a la VLAN 99 con su IP. Así hasta ocupar las 24 bocas de un SW.

Por defecto un SW asigna todos sus puertos (los 24)con la VLAN 1. Por eso es mejor cambiar esos valores por defecto parea administrar el SW a nuestra conveniencia quitandolde la VLAN 1. **El SW tiene su default-gateway para poder ser administrado no solo a nivel local sino desde otras redes**.

Cisco Switch IOS Commands	
Enter global configuration mode.	S1# **configure terminal**
Enter interface configuration mode for the SVI.	S1(config)# **interface vlan99**
Configure the management interface IP address.	S1(config-if)# **ip address 172.17.99.11**
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**

Otra característica de un SW es que puede ser Half-Dúplex o Full-Dúplex



Cisco Switch IOS Commands	
Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface FastEthernet 0/1`
Configure the interface duplex.	`S1(config-if)# duplex full`
Configure the interface speed.	`S1(config-if)# speed 100`
Return to the privileged EXEC mode.	`S1(config-if)# end`
Save the running config to the startup config.	`S1# copy running-config startup-config`

Comandos: dentro de la interface que queremos administrar:

- **SW(Config-if)#duplex full / half** (esto debe estar configurado en ambos extremos)
- **SW(Config-if)#speed + valor** en Mbs/seg

Si un SW está en "auto" trabajará a 100 Mg. Si la función de dúplex o half la tuviera en auto también, trabajaría por defecto en dúplex. Un SW siempre tiende por defecto a trabajar a la máxima velocidad y en full-dúplex, por eso siempre es recomendable dejar los valores en auto.

Después de haber configurado un SW siempre hacer un copy:

- SW#copy runnin-config startup-config

Auto MDIX: es una función que puede o tener tener un SW dependiendo de lo moderno y bueno que sea. Lo que hace es autoconfigurarse en cuanto si cable cruzado o directo. Entre dos SW para que funcione, esta función deben tenerla los dos, sino, no funciona.

Dispositivos de distinta capa=cable directo

Dispositivos de igual capa= cable cruzado

Ojo! Entre SWs de distintas capas, se usará siempre un cable cruzado, ya que un SW de capa 3, al trabajar también en capa 2, funcionara en principio como uno de capa 2 por lo que necesitará un cable cruzado con otro SW de capa 2 aunque funcionen en capas diferentes

Comandos para configurar todo por defecto en un SW

Configure auto-MDIX

Cisco Switch IOS Commands	
Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface fastethernet 0/1`
Configure the interface to autonegotiate duplex with the connected device.	`S1(config-if)# duplex auto`
Configure the interface to autonegotiate speed with the connected device.	`S1(config-if)# speed auto`
Enable auto-MDIX on the interface.	`S1(config-if)# mdix auto`
Return to the privileged EXEC mode.	`S1(config-if)# end`
Save the running config to the startup config.	`S1# copy running-config startup-config`

Comandos para show:

Verification Commands

Cisco Switch IOS Commands	
Display interface status and configuration.	`S1# show interfaces [interface-id]`
Display current startup configuration.	`S1# show startup-config`
Display current operating config.	`S1# show running-config`
Display information about flash file system.	`S1# show flash`
Display system hardware and software status.	`S1# show version`
Display history of commands entered.	`S1# show history`
Display IP information about an interface.	`S1# show ip [interface-id]`

Display the MAC address table.	`S1# show mac-address-table`
	OR
	`S1# show mac address-table`

SSH:

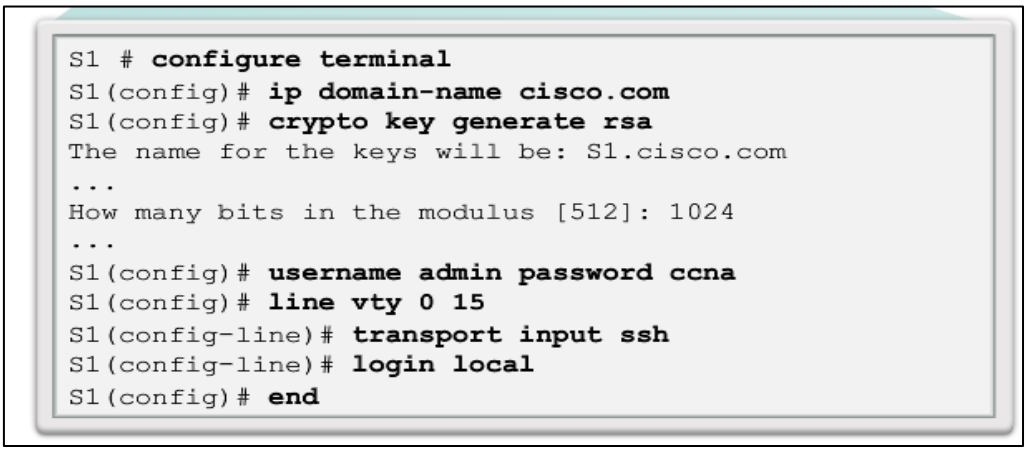
***para la configuración del rsa que encripta ssh hace falta siempre dos cosas**

- Nombre de dominio

- Nombre de equipo

Creamos nombre de dominio: **R(Config)#ip domain-name +**costa54.com

1- Encriptamos: **R(Config)#crypto key generate rsa** (RSA crea dos claves: encripta y desencripta)
2- Le damos **valor 1024** cuando pregunte.
3- Ponemos usuario y encriptamos: **R(Config)#username Chus secret cisco** (SSH necesita usuario y contraseña)
4- Llamamos a la linea vty 04 con **R(Config)#line vty 0 4** ó **15** (depende del numero de conexiones que tengamos a la vez)
5- Obligamos a que pida usuario y contraseña con **R(Config-line)#login local**
6- Obligamos a que se tenga que entrar por ssh con **R(Config-line)#transport input ssh**
7- No hay que olvidar que para habilitar la version 2 de ssh habrá que poner el comando**:** **R(Config)#ip ssh version 2**



Para usar SSH desde el PC nos hará falta un programa de software como puede ser el Putty o alguno parecido, para poder usar SSH como cliente.

Cuando el PC lanza un SSH al servidor, se crea un túnel de comunicación encriptada.

RSA proporciona:

- Algoritmo asimétrico
- Par de claves pública/privada
- Autenticación, Integridad y Confidencialidad

Telnet manda la información en texto plano por lo que es vulnerable.

SSH utiliza TCP y puerto 22

Telnet usa TCP y puerto 23

A nivel lógico, un dispositivo sin IP es como si no existiera. Por eso para administrar un SW hay que proporcionarse, bien por administración o por consola: interface vlan…: ip address….default-Gateway…

Desde línea de comandos de un PC, el comando para conectarse por SSH es:

SSH –L + nombre de susuario + IP

DHCP

Ataques de Dhcp:

- **Spoofing**: Servidor dhcp falso. Cuando un cliente solilcita una dirección IP por dhcp, lo hace mandando un broadcast, inundándolo todo y el servidor dhcp hace lo mismo. Un ataque de spoofing es cuando un pc actúa como un servidor dhcp falso o incluso como un router.

1- Los host mandan peticiones de direcciones IP por **dhcp discover** mediante un broadcast.
2- El servidor responde con un **offer dhcp**, que es unicast, en el que se encuentra:
   a. Dirección IP
   b. Máscara
   c. Puerta de enlace
   d. DNS
   e. Opciones adicionales

   En un ataque de spoofing el atacante al comportarse y hacerse pasar por un servidor de dhcp, mandará también offer dhcp

3- El cliente responde con un **request** al 1er servidor dhcp que le llegue mediante broadcast.
4- El servidor vuelve a responder confirmando la entrega (dándose por enterado que el cliente ya tiene el offer) con un **ACK=ok** o un **NACK=no tiene acuse de recibo** de manera unicast.

Como en un offer dhcp está también la puerta de enlace, cuando es el atacnate el que lo manda, manda su propia ip como puerta de enlace a todos los host que está atacando, con lo cual, todo el tráfico que salga desde los equipos atacados pasará primero por el atacante.

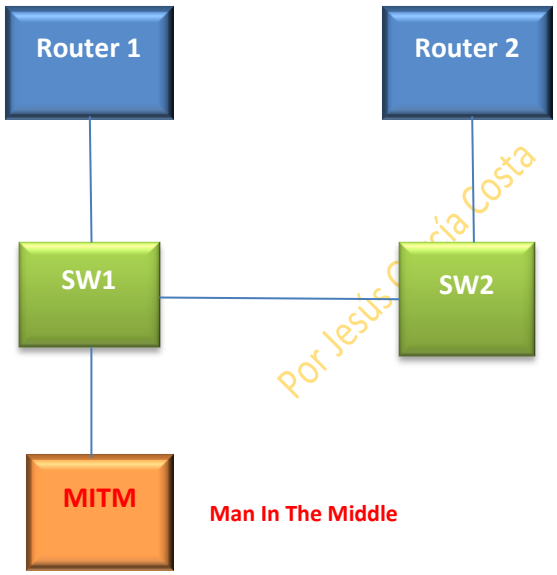
En ningún caso habrá duplicidad de IPs porque tanto el servidor legítimo como el atacante llevan un registro de las IPs que otorgan.

Otro ataque es que el atacante pide muchos requests al servidor legítimo hasta agotar su **pull**=recipiente, rango o ámbito de direcciones. Así, cuando un cliente legítimo pide una IP mediante un dhcp discover, al no quedar ya direcciones disponibles, el servidor no tiene más remedio que mandarle un NACK, con lo cual el cliente legítimo no tiene más remedio que autoasignarse una IP: la A PIPA= ip de clase B en el rango 169.254.0.1 a 169.254.255.254 con máscara 255.255.0.0 y consecuentemente, no tendrá salida a internet. Por ello, lo que el atacante consigue es un **ataque de denegación de servicio**.

Descubrimiento de equipos: LLDP / CDP

Hay dos formas de descubrimiento de equipos. Mediante el protocolo **LLDP** (genérico) o por el protocolo de Cisco que es el **CDP**.

Se hace por **tabla de vecinos**, de tal forma que:



En este diagrama, el SW1 tiene como vecino a SW2 y al R1

SW2 tendrá de vecino a SW1 y R2

R1 solo a SW1

R2 solo a SW2

La información que recoge las tablas de vecinos es:

- IP de administración
- VLANs
- IOS que se tiene
- VLAN Nativa
- Velocidad de Puerto

- Half/Full Dúplex
- Etc

Así se pueden auto-configurar los dispositivos entre sí.

Caso de haber algún BUG, el MITM podrá entrar y tener acceso a toda esa información, pudiendo invadir al SW con peticiones CDP.

1- Se obtiene información de los dispositivos
2- Se puede inundar a un dispositivo de peticiones

CDP se utiliza también para la alimentación de teléfonos IP: intercambio entre el teléfono y el SW. Siempre se recomienda tener activado el CDP para, entre otras cosas, otorgarle automáticamente el voltaje que el teléfono necesite.

CDP se utiliza también para mostrar con software adicional, consolas y gráficos de toda la red cisco de una empresa. Software como CCP, Mars, etc.

Gracias al CDP se puede auto-configurar un equipo.

Con el comando: **R(Config)#cdp run** se habilita el descubrimiento de vecinos.

Con el comando: **R(Config)#no cdp run** se deshabilita

Desde una interface el comando es: **R(Config-if)#cdp enable** o **R(Config-if)#no cdp enable**

Comandos de CDP**:**

SW#show cdp

SW#show cdp neighbors

Con cdp activado, si no se recibe señal vecinal cada x tiempo, se quitará el vecino sin señal de la tabla ARP

SW#show cdp neighbors detail

DHCP SNOOPING

1- Se habilita la herramienta con **SW#(Config)#ip dhcp snooping**
2- Decimos a que VLAN se aplica: **SW#(Config)#ip dhcp snooping vlan 10,20** (a las vlan 10 y 20 en este caso)
3- Vamos a las interfaces y las configuramos para que acepten o no dhcp con el comando: **SW(Config-if)#ip dhcp snooping trust**
   Se usa este comando porque al haber habilitado el **SW#(Config)#ip dhcp snooping** , por defecto se ponen todas las interfaces en **untrust**
4- Con el comando **SW(Config-if)#ip dhcp limit rate + valor** le estamos diciendo que acepte un número limitado de peticiones (el valor que le hemos puesto al comando) para evitar ataques

```
S1(config)# ip dhcp snooping
S1(config)# ip dhcp snooping vlan 10,20
S1(config)# interface fastethernet 0/1
S1(config-if)# ip dhcp snooping trust
S1(config)# interface fastethernet 0/2
S1(config-if)# ip dhcp limit rate 5
```

Con estas acciones se impide el ataque tanto por suplantación de servidor de dhcp, como de ataques de denegación de servicio por inundación de peticiones.

PORT SECURITY

Con esta herramienta se mitigan ataques de dirección MAC e inundaciones de la tabla CAM.

Se puede hacer de tres formas:

1- MAC **Manual** Estática poniendo la MAC y el Puerto
2- MAC **Dinámica**
3- MAC **Sticky**

Para activar el modo port security se hace con el comando:

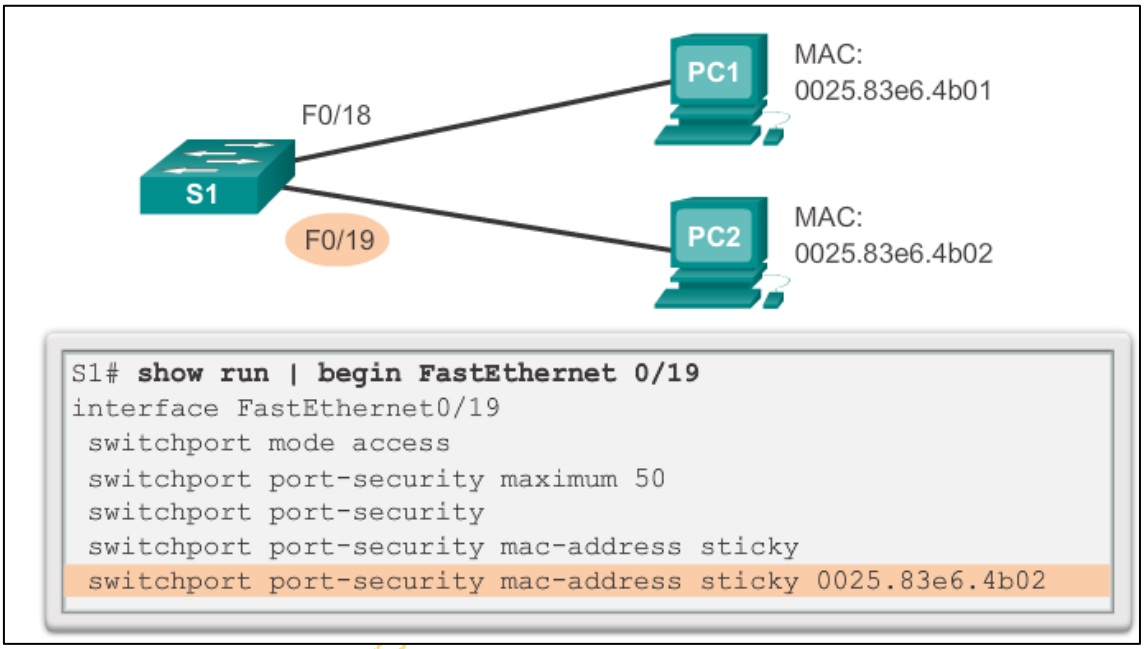
1- Dentro de la interface: **SW(Config-if)#switchport mode access** para activar el modo
2- **SW(Config-if)#switchport port-security** para habilitar el security.

Cisco IOS CLI Commands	
S1(config)#**interface fastethernet 0/18**	Specify the interface to be configured for port security.
S1(config-if)#**switchport mode access**	Set the interface mode to access.
S1(config-if)#**switchport port-security**	Enable port security on the interface.

3- Ya habilitado el port-security hay que decirle como va a aprender las MAC. Si es con el comando sticky sería: **SW(Config-if)#switchport port-security mac-address sticky**
Una vez habilitado el modo sticky, también se puede poner las MACs manualmente con el comando completo más la MAC en cuestión forzándolo así a que aprenda, de tal forma que el paso 4 sería:

4- **SW(Config-if)#switchport port-security mac-addres sticky + la MAC**

5- Además de configurar el modo de aprendizaje, para tener una buena seguridad, hay que configurar el valor máximo de MACs que puede aprender esa interface con el comando: **SW(Config-if)#switchport port-security máximum + valor**



6- Por último hay que configurar también la acción que se ha de tomar caso de violación de los parámetros configurados. Esto se hace con el comando:
SW(Config-if)#switchport port-security violation + la acción a tomar
La acción a tomar puede ser 3:

- **Protect:** descarta las tramas inválidas. Solo atiende a las direcciones MAC válidas proporcionadas
- **Restrict:** Solo se permiten las direcciones MAC válidas y si le llega una dirección no válida, la deniega y a la vez manda un mensaje SNMP a administración.
  Shutdown: Tira abajo la interface.
  *****_Para volver a levantar una interface caída por un violation shutdown , hay que entrar en esa interface y darle 1º otro shutdown y 2º un no shutdown_

Comandos Show del port-security:

SW#show mac-address table

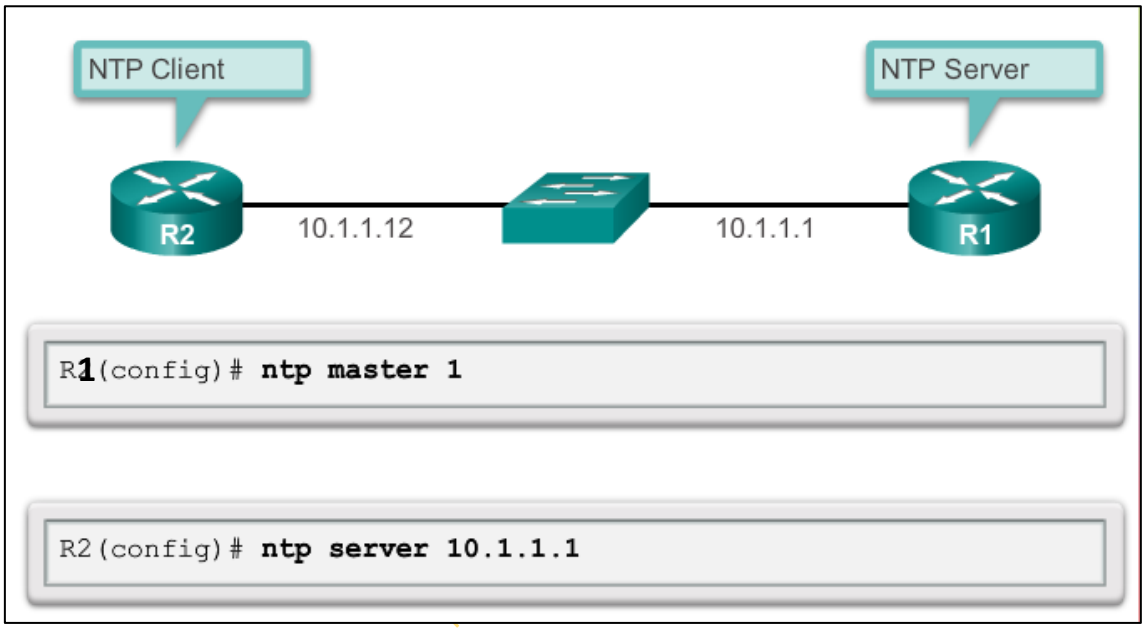
SW#show port-security interface fa 0/1

SW#show interface fa 0/1 status

Si en la meter el comando **show interface fa 0/1 status** apareciera el término **err-disable** significaría que esa interface se cerró por un violation shutdown. Una vez que entráramos en esa interface y le diéramos un shutdown y después un no shutdown, ese mensaje de error desaparecería al volver a darle un show interface fa 0/1 status.

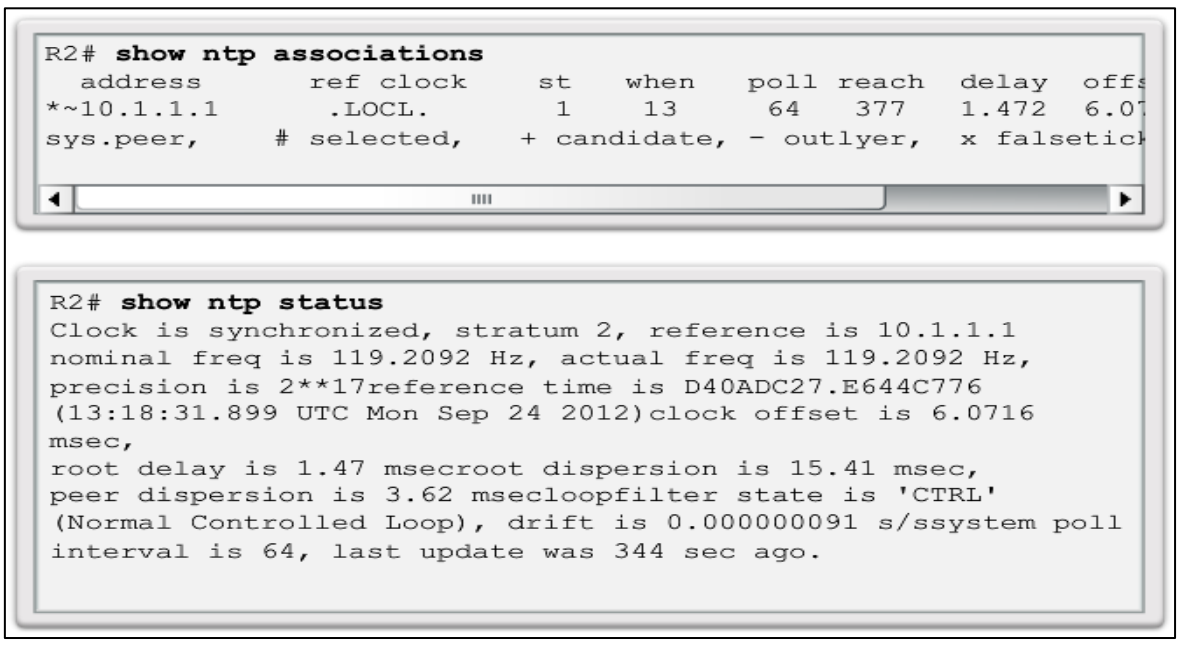
NTP

NTP sincroniza la hora. Se puede nutrir de dentro o de fuera, es decir, desde un servidor NTP de la red propia o de internet.



R(Config)#ntp master 1

R(Config)#ntp server + ip del servidor ntp



VLAN

VLAN=subred/red=dominio de broadcast

Lo normal es que cada vlan tenga una IP asociada

Vlan 10: 192.168.10.0 *SW *Host: Routers, terminales

Vlan 20: 192.168.20.0

Las direcciones de las Vlan se configuran en el SW. La red de esas Vlan se configura en el Router. Un SW entiende de Vlan. Un host entiende de direcciones IP.

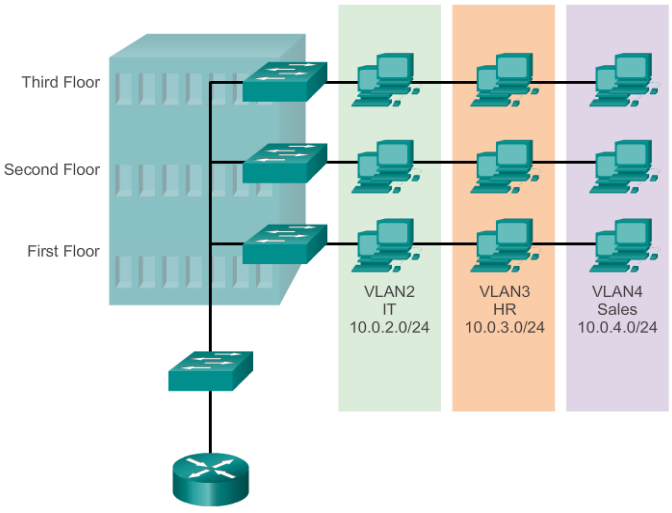
Las vlan se asocian a los puertos físicos. El SW diferencia entre tráfico de distintas vlan. Sólo el Router. O un SW de capa 3 puede comunicar distintas Vlan.

Un host que pertenezca a una vlan tendrá IP de esa Vlan, dentro de ese rango. La puerta de enlace de un host dependiente de una vlan así como la default-gateway de una Vlan será la IP de interface o subinterface del router.

Si la vlan 20 es la 192.168.20.0 la ip de un host será por ejemplo la 192.168.20.2 y su puerta de enlace la del Router.

Un SW añade 4 bytes más a la trama MTU para etiquetarla, pasando de 1518 a 1522.

Uno de los beneficios de las Vlan es que acotan el dominio de broadcast al comportarse como una red independiente. Una Vlan puede extenderse a lo largo de varios SW o varios pisos en una empresa, atendiendo a la configuración que hagamos.



- **Seguridad**
- **Reducción de costes**
- **Acota dominios de broadcast**
- **Mejora el rendimiento**
- **Mejora la gestiós**
- **Como en la diapositiva, se podría dar el caso de de SWs de una misma Vlan en distintos pisos:**
  Vlan 10: Para tráfico de datos de usuario
  Vlan 20: Tráfico de la Granja de Servidores
  Vlan 30 para tráfico de voz

Las Vlan no se ven entrre sí. Para ello haría falta un dispositivo de capa 3: SW de capa 3 o Router. Un dispositivo de capa 3 se rige con ACL´s.

Se puede y debe crear una vlan de administración separada del resto vlans.

Tipos de Vlan:

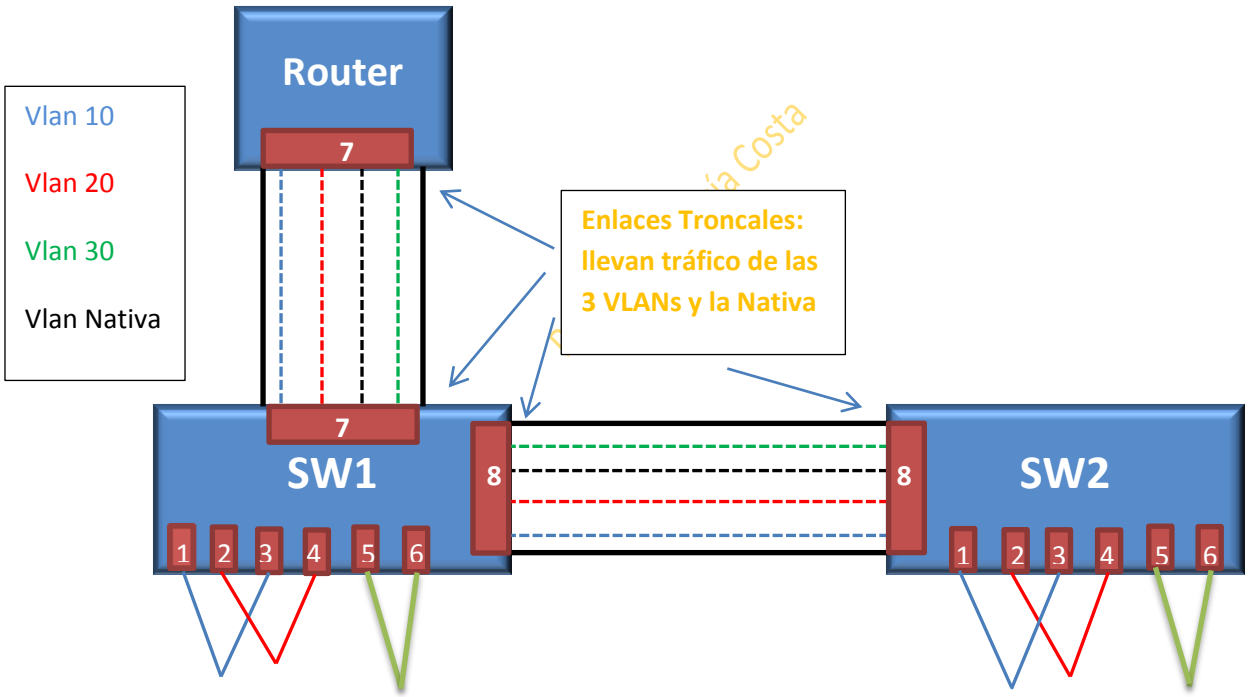
- Vlan de datos: Por ejemplo las de producción
- Vlan por defecto: Vlan 1
- Vlan Nativa: Sin etiqueta. Mantiene el valor 1518
- Vlan de admistración, gestión y/o monitorización

Por defecto los 24 puertos de un SW pertenecen a la Vlan 1.

La Vlan Nativa es una Vlan sin etiqueta. Su trama no tiene va sin etiqueta de SW por eso mide 1518 y no 1522.

El comando **#show vlan brief** es muy importante. Nos muestra las vlan de un SW y del tipo que son.

Tipos de interface y protocolo troncal:



Un SW tiene dos tipos de puertos:

- ACCESO
- TRONCAL

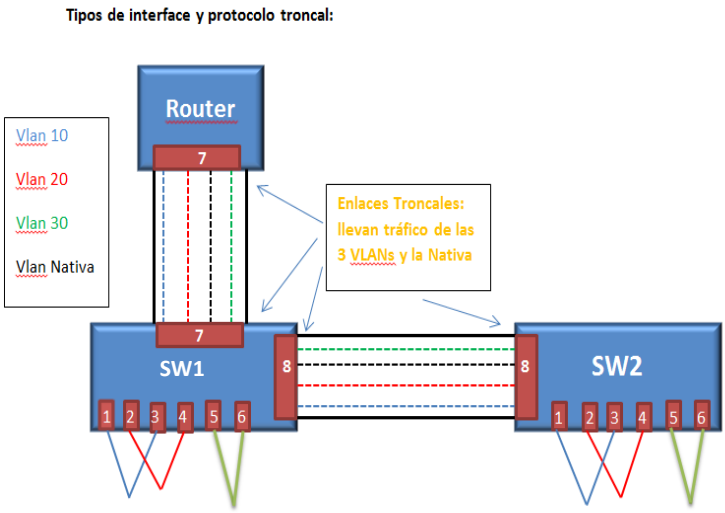
Las interfaces en modo acceso son donde se conectan los terminales, (del 1 al 6 en el ejemplo)

La **Membresía** es la asociación entre puerto físico y Vlan

La Vlan 1 no se puede eliminar.

Las interfaces del 1 al 6 del ejemplo, son interfaces en modo acceso. Una interface/puerto puede albergar a la vez una Vlan de acceso asociada y una Vlan de voz, pero no dos de voz a la vez o dos de acceso a la vez, solo una de cada a la vez.

En el ejemplo anterior podría haber una Vlan 40 con tráfico de voz dentro de las interfaces 1-6. Sin embargo y para que sirva de ejemplo, no se podría albergar una vlan de servidores, ya que ésta sería de datos y ya tienen esas interfaces asociadas tráfico de datos. (Solo una de cada).



Comunicación Intra-Vlan: Es la comunicación entre equipos de la misma Vlan aunque estuvieran en distintos SWs.

Comunicación Inter-Vlan: Es comunicación entre distintas Vlans. Se necesita un dispositivo de capa 3.

Interface en modo Troncal:

Aquí se permiten distintas Vlans por inetrface, como podemos ver, entre R1 y SW1 por el puerto 7 que está en modo troncal al igual que el puerto 8. Por esas interfaces 7 y 8 pasan las VLANs 10, 20 y 30.

Las tramas que pasan por las interfaces troncales se les mete una etiqueta que las diferencia de las otras tramas para que sepan a que Vlan pertenecen y sepan el camino.

Los enlaces troncales se usan para que distintas Vlans pueden trabajan en el mismo dispositivo. Además permiten que se comuniquen Vlans de distintos SWs.

La trama de 1518 viaja por el troncal con un valor de 1522 por que lleva la etiqueta de la Vlan a la que pertenece.

Aunque dos equipos estén compartiendo SW, si pertenecen a distinta Vlan no se verán entre sí y necesitarán de un dispositivo de capa 3 para poder hacerlo.

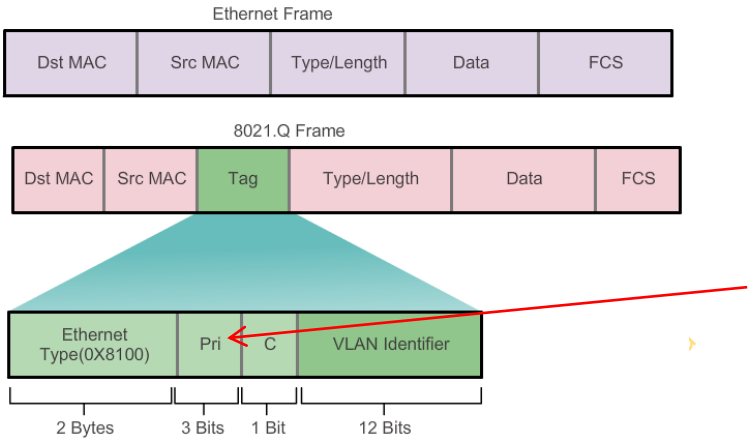
Cuando una trama llega al Router por un enlace troncal, el Rouetr debe tener su interface subdividida en subinterfaces, ya que cada Vlan tiene su IP propia y por una interface de un

Router en IPV4 solo podría albergar a una IP. Por eso la necesidad de hacer subinterfaces en los Router.

Una Vlan que no estuviera en un enlace troncal, buscaría su default-gateway propia del SW, pero precisamente al ser troncal, necesita llegar al Router para poder comunicarse con otras Vlans.

La etiqueta es fundamental para el tráfico de los troncales. Al llegar a destino, la etiqueta se deshecha. Poner y quitar etiquetas es trabajo del SW.

En modo acceso no hace falta protocolo, pero en modo troncal sí, al necesitar etiquetado ya que se tiene que encapsular la trama.



El protocolo 802.1Q es el que añade los 4 bits a la trama.

Se puede dar más o menos prioridad a una Vlan: A número más alto, más prioridad y menos retardo. El tráfico de gestión y control, así como el de voz, son los que mayor prioridad tienen

Las Vlan pueden ir de 0 a 4096.

Para aumentar el número de Vlan habría que hacer un QinQ, que es lo que hace una MetroLan: no es otra cosa que aumentar el etiquetado. El protocolo 802.1Q es el protocolo estándarpara el etiquetado, pero Cisco tiene el suyo que es el ISL.(20 bits)

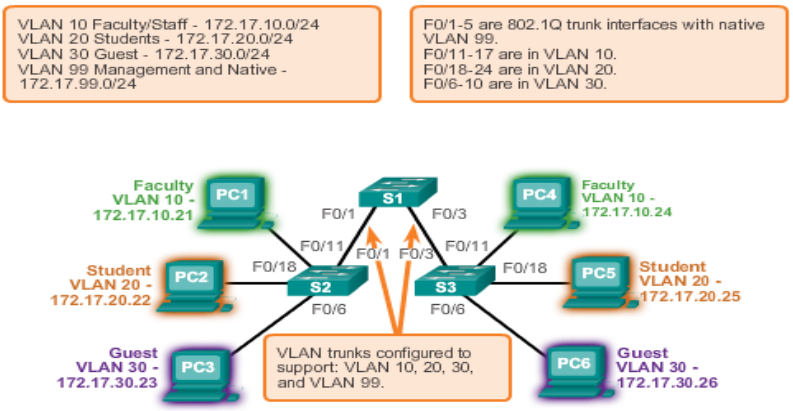
Vlan Nativa: Aquí se mandan las tramas sin etiqueta. Por defecto es igual a la Vlan 1.

Podemos hacer que la Vlan Nativa sea la que queramos pero generalmente Cisco le pone el valor 99. La Vlan Nativa solo está en los enlaces troncales.

Cisco Switch IOS Commands	
Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface interface_id`
Force the link to be a trunk link.	`S1(config-if)# switchport mode trunk`
Specify a native VLAN for untagged 802.1Q trunks.	`S1(config-if)# switchport trunk native vlan vlan_id`
Specify the list of VLANs to be allowed on the trunk link.	`S1(config-if)# switchport trunk allowed vlan vlan-list`
Return to the privileged EXEC mode.	`S1(config-if)# end`

Vlan de voz: Tiene prioridad ante los datos, y un retardo máximo de 150 milisegundos.

Vlan Troncal:



Configuración de VLAN:

Del 1 al 1005 es el denominado rango "normal", pero del 1002 al 1005 son los creados por defecto para token-ring y FDDI. Por lo tanto los valores que usaremos en el modo ethernet son del 1 al 1001.

Estas Vlan se guardan en el fichero **vlan.dat** que está en la memoria Flash.

VTP es un protocolo que solo puede usarse en el rango normal.

El rango extendido va del 1006 al 4096 y se almacena en la NVRam.

Para crear una Vlan:

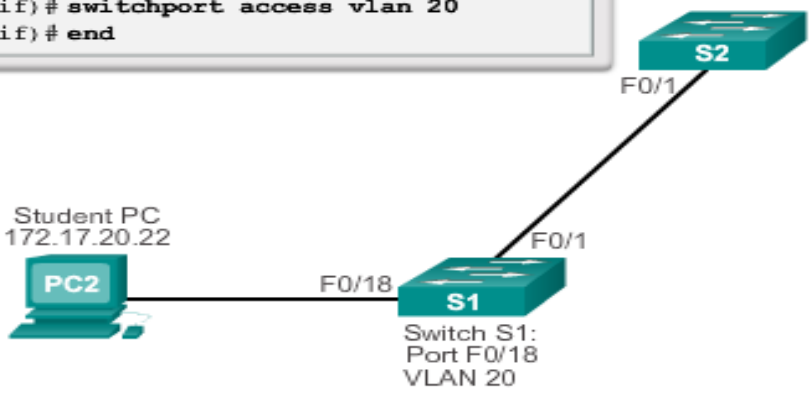
Cisco Switch IOS Commands	
Enter global configuration mode.	S1# **configure terminal**
Create a VLAN with a valid id number.	S1(config)# **vlan** vlan_id
Specify a unique name to identify the VLAN.	S1(config)# **name** vlan_name
Return to the privileged EXEC mode.	S1(config)# **end**

Una vez que la Vlan está creada según vemos el dibujo, nos metemos físicamente en la interface y asignamos a esa interface la vlan que pasará por allí:

- SW(Config-if)#switchport mode access
- SW(Config-if)#switchport Access vlan_ID(la que le pongamos)

Cisco Switch IOS Commands	
Enter global configuration mode.	S1 # **configure terminal**
Enter interface configuration mode for the SVI.	S1(config) # **interface** *interface_id*
Set the port to access mode.	S1(config-if) # **switchport mode access**
Assign the port to a VLAN.	S1(config-if) # **switchport access vlan** *vlan_id*
Return to the privileged EXEC mode.	S1(config-if) # **end**

```
s1# configure terminal
s1(config)# interface F0/18
s1(config-if)# switchport mode access
s1(config-if)# switchport access vlan 20
s1(config-if)# end
```



En un SW, las interfaces siempre están asociadas a una Vlan. Por defecto es a la Vlan 1. Para ver todo lo relacionado con las interfaces asociadas a una Vlan, usamos el comando **show interfaces vlan_id**.

Si una Vlan se elimina, el puerto asociado a ella pasa a inactivo.

Para configurar una interface troncal:

- Toda interface troncal necesita tener asociada una vlan nativa y debe ser la misma en ambos puertos troncales. Por eso, cuando es trancal entre SW y Router, a la parte del R también hay que decirle que es nativa: R(config-subif)#encapsulation dot1q 99 **native**
- Toda interface troncal necesita asociarse con VLANs

```
S1(config)# interface FastEthernet0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk native vlan 99
S1(config-if)# switchport trunk allowed vlan 10,20,30
S1(config-if)# end
```

Enlace troncal (entre switches)

Desde config me voy a la interface.

Para cambiar la vlan nativa:

1º le decimos que es modo troncal.

SW(Config-if)#switchport mode trunk

Ojo! No olvidar al configurar un troncal, que debe tener la misma configuración en ambos extremos de los SWs. Y si el otro extremo es u router, hay que decirle a la subinterface que hacemos que va a ser nativa: Ej:

R(Config)#interface gi 0/0.99

R(Config-subif)#encapsulation dot1q 99 native

R(Config-subif)#ip address _____

2º Le metemos la Vlan Nativa:

SW(Config-if)#switchport trunk native vlan + un valor decimal(99)

Es bueno crear una vlan nativa para quitarle trabajo a la vlan 1

3º le decimos que vlan permitirá.

SW(Config-if)#trunk allowed vlan + el vamor decimal de la vlan que queramos permitir. Así una a una de todas las vlan que permitamos.

2.1º Si queremos permitir todo:

SW(Config-if)#switchport trunk allowed vlan all

Las interfaces troncales de 2 SW deben tener la misma configuración

Atención a los commandos de verificación:

- **Show vlan brief**
- **Show interfaces vlan_ID**
- **Show interfaces f0/1 switchport** (para ver toda la condiguración de esa interface)

DTP

DTP es un protocol de negociación. Dos interfaces troncales deben tener la misma configuración, o troncal o de acceso, pero con este protocol de cisco, también se puede negociar el modo:

Estando dentro de la interface SW(Config-if)#**switchport mode dynamic** auto/desirable

Atendiendo a cómo esté cada interface, las combinaciones serán:

	Dynamic Auto	Dynamic Desirable	Trunk	Access
Dynamic auto	Access	Trunk	Trunk	Access
Dynamic desirable	Trunk	Trunk	Trunk	Access
Trunk	Trunk	Trunk	**Trunk**	Limited connectivity
Access	Access	Access	Limited connectivity	**Access**

Por defecto el DTP está habilitado.

Hay que conocer que hay ataques que se llaman de “doble etiquetado”

PVLAN: así se denominan a los puertos protegidos. Un Puerto protegido es aquel que impide cualquier tipo de tráfico con otro que también lo esté, permitiéndose todos los demás.

Con un show vlan brief veremos las interfaces que están asociadas, las membresías y al hacer esto en distintos SWs iremos viendo también a que Vlan pertenece cada interface.

Ojo! Comandos Show!!!!!!

Routing

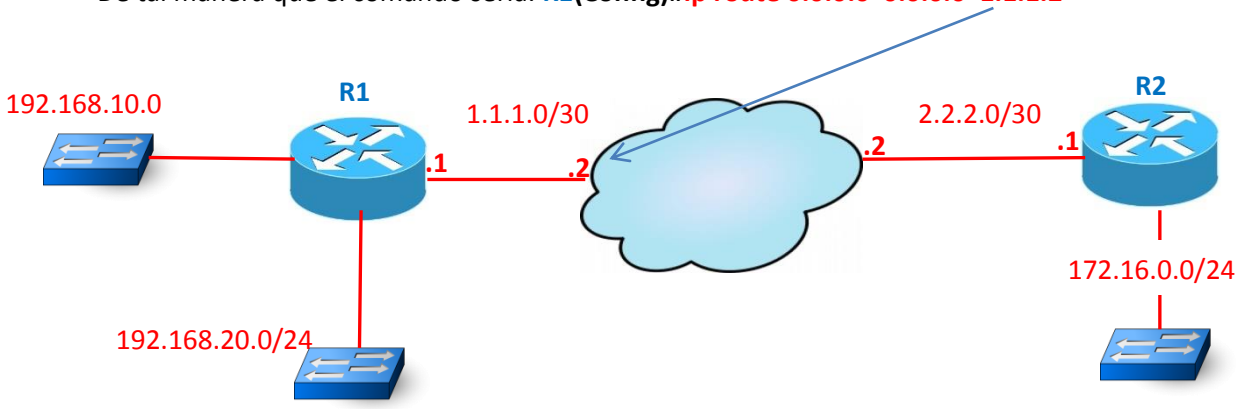
El Routing es la comunicación de equipos de distintas redes.

Memory	Volatile / Non-Volatile	Stores
RAM	Volatile	• Running IOS • Running configuration file • IP routing and ARP tables • Packet buffer
ROM	Non-Volatile	• Bootup instructions • Basic diagnostic software • Limited IOS
NVRAM	Non-Volatile	• Startup configuration file
Flash	Non-Volatile	• IOS • Other system files

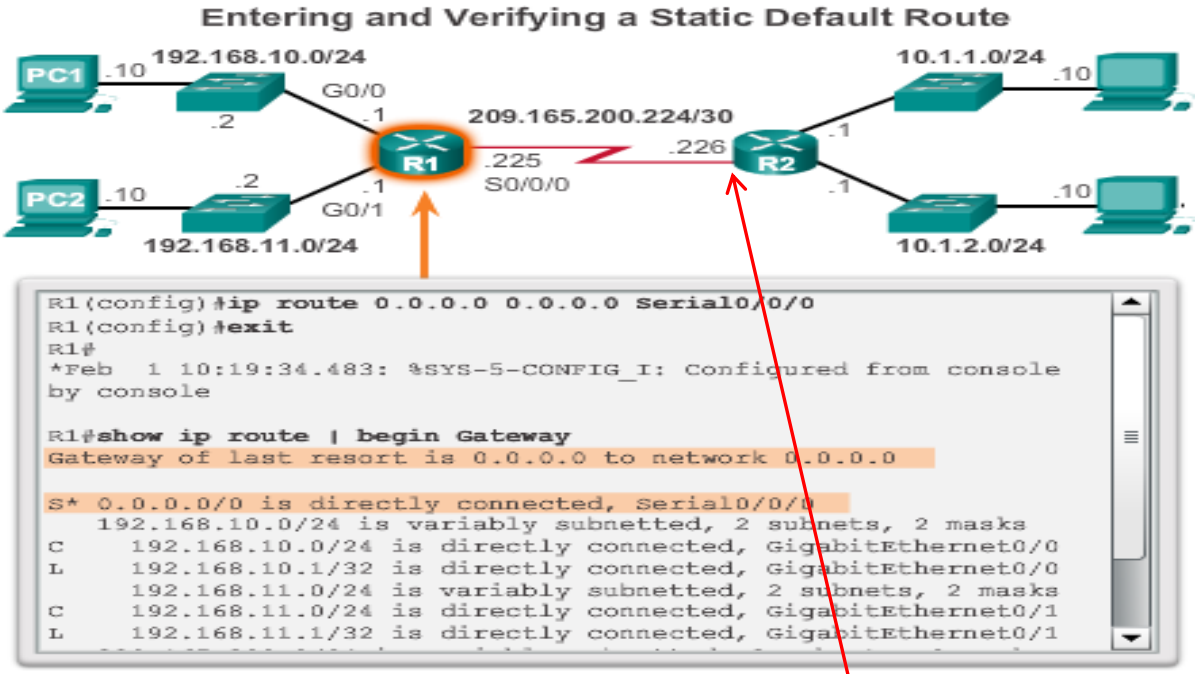
Rutas estáticas: lo aconsejable es meter la estática por defecto: 0.0.0.0 0.0.0.0 S0/0

Ruta Predeterminada: **La red 0.0.0.0 significa “Todas Las Redes”. Su máscara será también la 0.0.0.0 y La Puerta de Enlace será la interface del próximo salto.**

De tal manera que el comando sería: **R1(Config)#ip route 0.0.0.0 0.0.0.0 1.1.1.2**



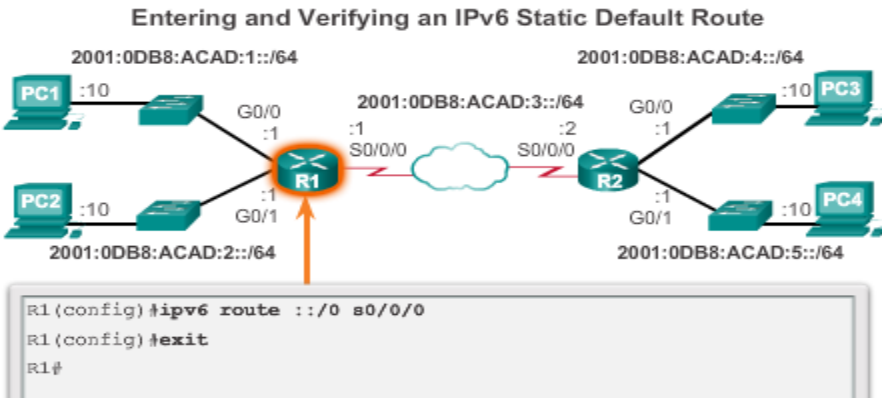
Cont. Ruta estática



En esta diapositiva, la ruta estática por defecto para R1 seria:

1- **Recursiva (de próximo salto): 0.0.0.0 0.0.0.0 209.165.200.226**
2- **Directa (por interface de salida): 0.0.0.0 0.0.0.0 serial 0/0/0**

Ruta estática en IPV6



R1(Config)#ipv6 route ::/0 s0/0/0

En IPv6, al ponerle la IP a una interface del Router **no hace falta ponerle la Link-local**. Lo hace él automáticamente con el EUI-64.

En IPV6 la link-local del router es la ip que se toma como próximo salto.

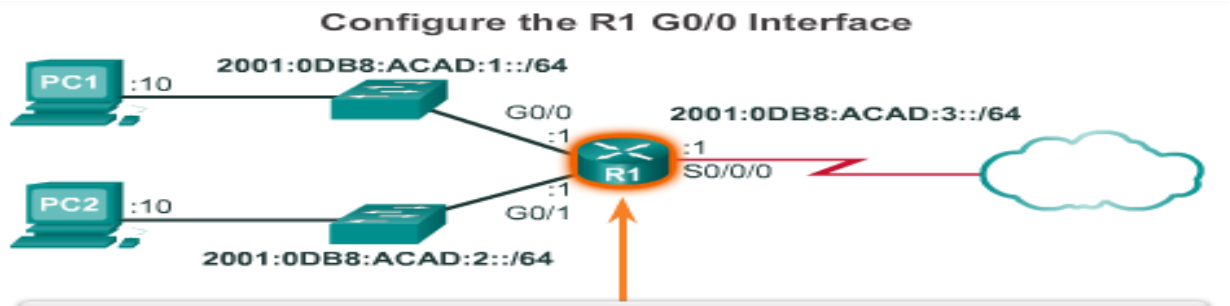
Rutas dinámicas: En un enrutamiento dinámico, los routers se mandan entre sí todas las redes a las que están directamente conectadas.

Comando **show ip route** para ver las rutas del router.

Tipos de conmutación(fordwarding): Process Switching, Fast switching, Cisco Express Fordwarding (CEF)

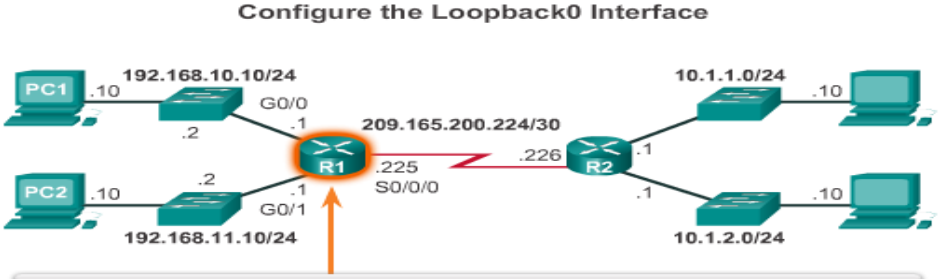
Comando **description**: se hace desde la interface Config-if#

El comando **clock-rate** se pone solo en el router **DCE** no en el DTE.




```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#description Link to LAN 1
R1(config-if)#ipv6 address 2001:db8:acad:1::1/64
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
*Feb  3 21:38:37.279: %LINK-3-UPDOWN: Interface
GigabitEthernet0/0, changed state to down
*Feb  3 21:38:40.967: %LINK-3-UPDOWN: Interface
GigabitEthernet0/0, changed state to up
*Feb  3 21:38:41.967: %LINEPROTO-5-UPDOWN: Line protocol on
Interface GigabitEthernet0/0, changed state to up
R1(config)#
```

La **interface loopback** sirve para hacer comprobaciones sobre el estado del Router. No se le asigna a ningún puerto físico del router. La loopback es importante en el protocolo OSPF



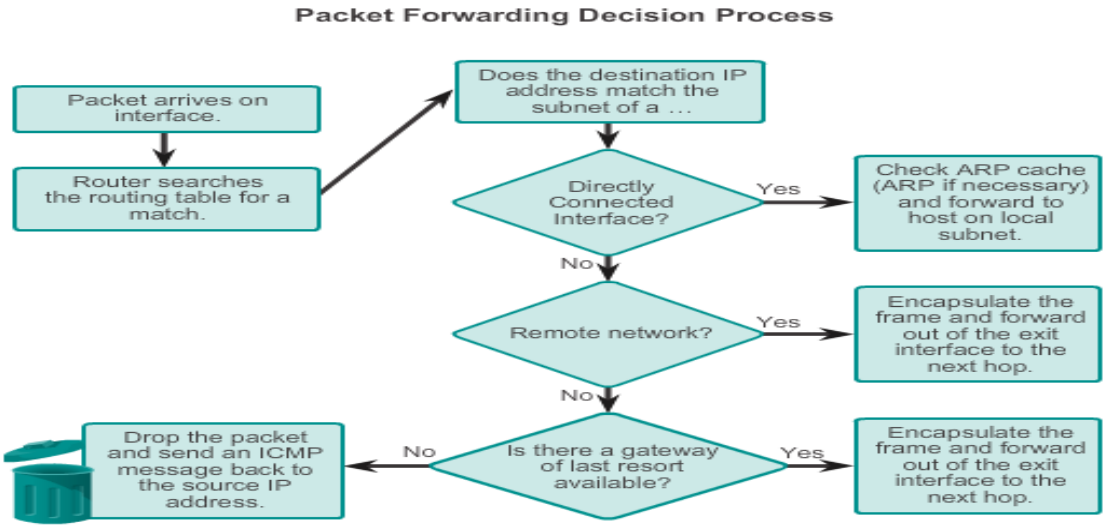

```
R2(config)#interface loopback 0
R2(config-if)#ip address 10.0.0.1 255.255.255.0
R2(config-if)#exit
R1(config)#
*Jan 30 22:04:50.899: %LINK-3-UPDOWN: Interface loopback0,
changed state to up
*Jan 30 22:04:51.899: %LINEPROTO-5-UPDOWN: Line protocol on
Interface loopback0, changed state to up
```

Comando **show ipv6 interface brief:** nos muestra las IPs de ipv6

```
R1#show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:ACAD:1::1
GigabitEthernet0/1      [up/up]
    FE80::FE99:47FF:FE75:C3E1
    2001:DB8:ACAD:2::1
Serial0/0/0             [up/up]
    FE80::FE99:47FF:FE75:C3E0
```

En el proceso de enrutamiento, según los paquetes van dando saltos (hops) de un sitio a otro, van teniendo distinto encapsulado.

Proceso de toma de decisiones de un router:



1º- Cuando llega el paquete, el router consulta su tabla de enrutamiento por si tuviera al alcance el destino

2º- Si está directamente conectado a destino chequea su ARP y lo enviá al host en esa red local

3º- Si el router no está conectado directamente a la red de destino encapsula la trama y le da salida por la interface del siguiente salto

A menor **métrica** = mejor camino.

Si la métrica coincide en dos caminos, se toma como referencia el balanceo de carga (repartir el trabajo)

Distancia administrativa: es la **confiabilidad**. La distancia administrativa pesa más que la métrica a la hora de tomar una decisión de enrutamiento. A igual distancia administrativa, se entrará a valorar la métrica.

Estos son los valores de distancia administrativa según protocolos:

Route Source	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
External EIGRP	170
Internal BGP	200

Dentro de una WAN/LAN pueden convivir distintos protocolos de enrutamiento.

La tabla de enrutamiento nos muestra las rutas directamente conectadas, las rutas remotas y las asociaciones con las redes o próximos saltos.

D=EIGRP O=OSPF R=RIP S=ESTATICA

C=RED CONECTAA DIRECTAMENTE L=IP DE ESA RED CONECTADA DIRECTAMENTE

Los protocolos varían según se es ipv4 o ipv6:

RIP 1 y 2 para ipv4 RIPNG para ipv6

OSPF 1 y 2 para ipv4 y OSPF 3 para ipv6

EIGRP trabaja en los dos

BGP para ipv4 y MP-BGP para ipv6

	Interior Gateway Protocols				Exterior Gateway Protocols
	Distance Vector		Link-State		Path Vector
IPv4	RIPv2	EIGRP	OSPFv2	IS-IS	BGP-4
IPv6	RIPng	EIGRP for IPv6	OSPFv3	IS-IS for IPv6	BGP-MP

Capitulo 5: Enrutamiento entre redes

Para que haya enrutamiento a través de un SW de capa 2, hará falta la ayuda de un router. Este enrutamiento se pude hacer de dos formas.

1- Creando varias VLANs en el Switch y usar una interface física del router por cada vlan que queremos enrutar.
2- Utilizar enlaces troncales en el SW y subinterfaces lógicas en el router para albergar las distintas VLANs

Evidentemente el modo 2 es el más adecuado: Se llama **Router-On-A-Stick**

En el router se configuran las subinterfaces fa 0/0/10 fa 0/0/20 fa 0/0/30

Para poder crear las subinterfaces en el router, hay que habilitar el protocolo 802.1Q y decirle a esa subinterface qué VLAN va a alojar: VLAN 10, VLAN 20 etc.

a- Creamos subinterfaces en el router. Estando en Config, ponemos el nombre de la interface física +.ID(valor que queramos dar).
Ejemplo: R1(Config)#interface Fa 0/0.**10**
b- El promp cambiará a R1(Config-subif)#. Le habilitamos el protocolo 802.1Q con el comando encapsulation DOT1Q + valor que queramos. Es muy aconsejable ponerle el mismo valor que le pusimos a la subinterface para que se identifique rápidamente. De tal forma que quedaría: R1(Config-subif)# encapsulation dot1q **10**
c- Le asignamos una IP a esa subinterface: R1(Config-subif)#172.17.10.1 255.255.255.0
d- El No shutdown se lo damos solo a la interface física, no a cada subinterface

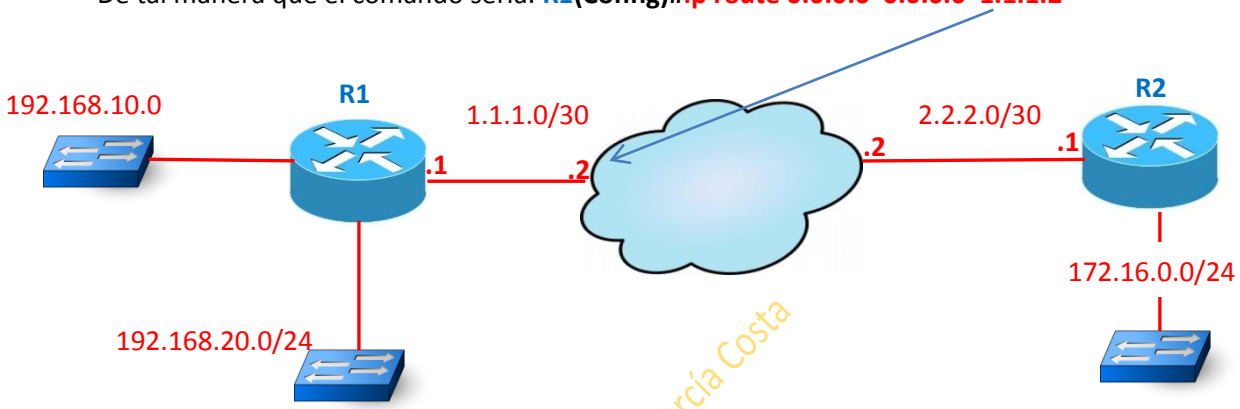
Una interface física de un router, tendrá tantas IPs o subinterfaces hasta 1001, que son las VLANs que puede alojar, pero meter tantas en una interface ralentizaría mucho el ancho de banda.

Capítulo 6: Enrutamiento estático

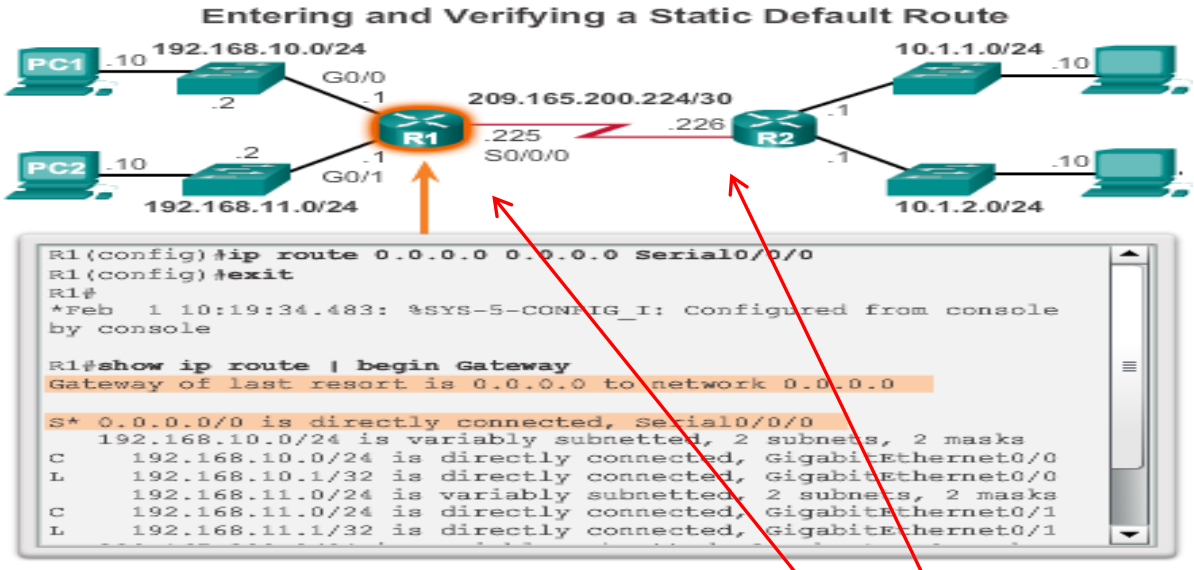
Rutas estáticas: lo aconsejable es meter la estática por defecto: 0.0.0.0 0.0.0.0 S0/0

Ruta Predeterminada: **La red 0.0.0.0 significa "Todas Las Redes". Su máscara será también la 0.0.0.0 y La Puerta de Enlace será la interface del próximo salto.**

De tal manera que el comando sería: **R1(Config)#ip route 0.0.0.0 0.0.0.0 1.1.1.2**



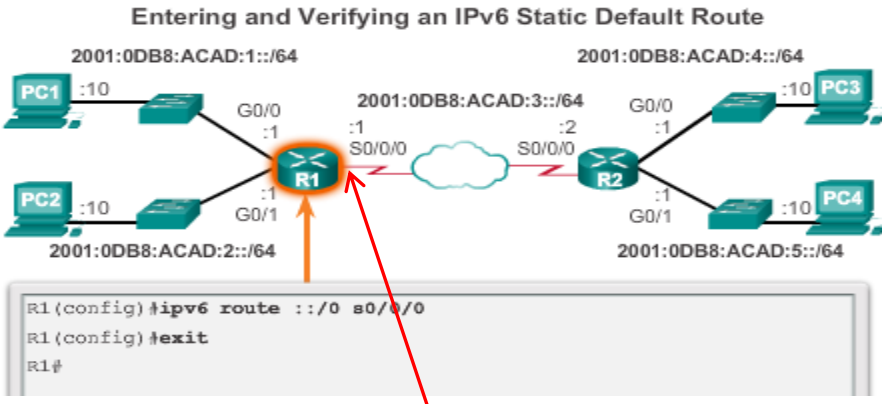
Cont. Ruta estática



En esta diapositiva, la ruta estática por defecto para R1 seria:

3- **Recursiva (de próximo salto): R1(Config)#ip route 0.0.0.0 0.0.0.0 209.165.200.226**
4- **Directa (por interface de salida): R1(Config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0**

Ruta estática en IPV6



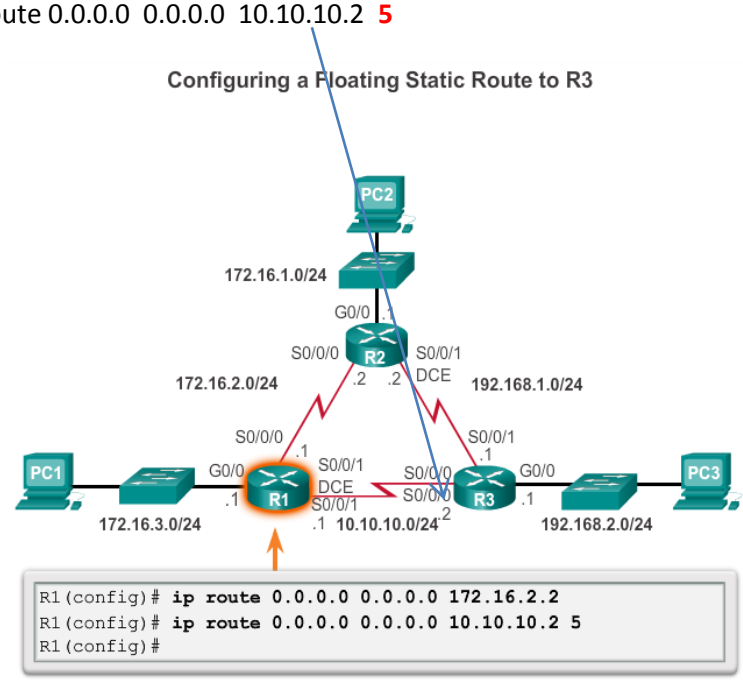
R1(Config)#ipv6 route ::/0 s0/0/0

A una ruta estática se le puede poner una **distancia administrativa manualmente**. Solo hay que definir la ruta y al final poner el valor decimal que queramos. Aprovechando el gráfico de arriba sería:

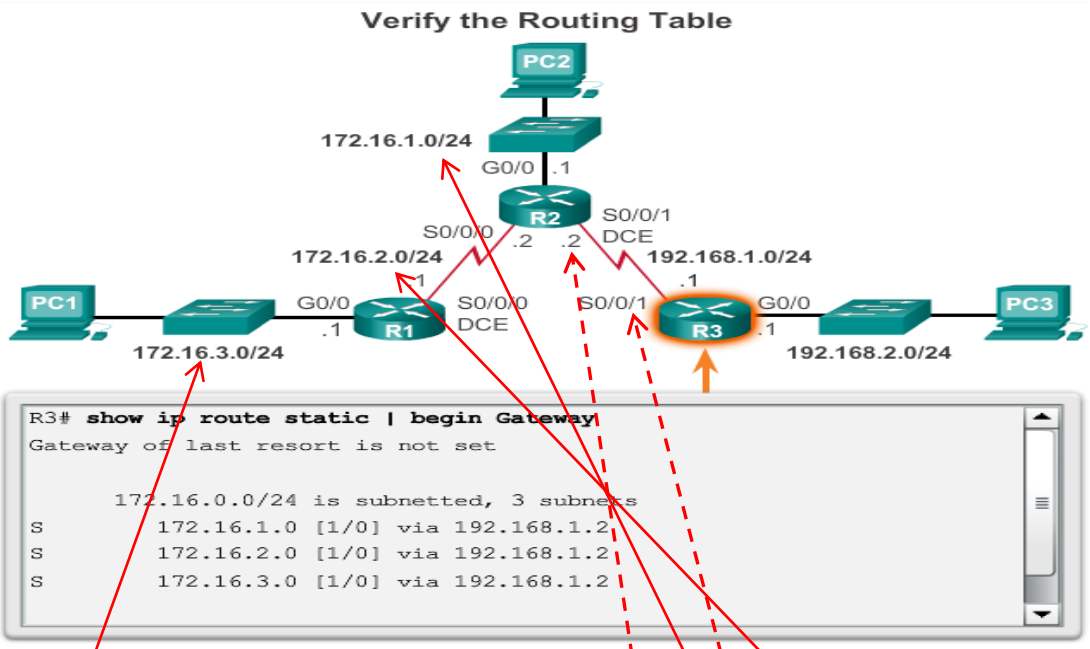
- R(Config)#ip route 0.0.0.0 0.0.0.0 se0/0/0 **5**
- **5** es el valor que le hemos dado manualmente para que tenga esa distancia administrativa.

Ruta estática flotante: Es aplicar una ruta estática con una distancia administrativa superior a 1, ya que 1 es su valor por defecto. La ruta estática flotante sirve para uso alternativo por si se desactivara la estática principal. En el ejemplo de abajo, una estática flotante sería:

R1(Config)#ip route 0.0.0.0 0.0.0.0 10.10.10.2 **5**

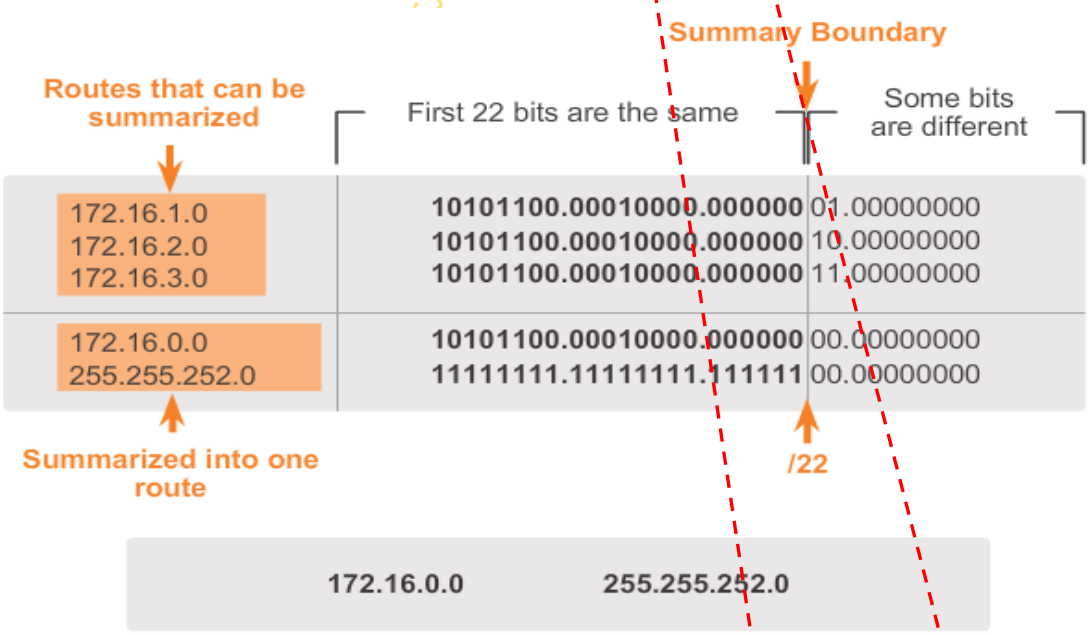


Rutas sumarizadas:



Sumarizar rutas es agrupar varias rutas en 1 sola, ya que que comparten bits entre sí. Es muy útil para calcular 1 sola ruta estática y que llegue a distintas redes.

En el ejemplo vemos que desde el R3 para llegar a las redes 172.16.1.0 172.16.2.0 y 172.16.3.0, tendríamos que hacer 3 rutas estáticas. Sin embargo, podemos sumarizarlas ya que comparten muchos bits iguales (todas son 172.16.) y así poner solo una ruta estática en vez de tres.



Como vemos, esas tres rutas, se pueden agrupar en 172.16.0.0, de tal forma que desde R3 ponemos: R3(Config)#ip route 172.16.0.0 255.255.252.0 192.168.1.2 / s 0/0/1

Red de Destino con su máscara

Interface de siguiente salto (Recursiva)

Interface de salida (Directamente)

Capítulo 7: Enrutamiento Dinámico

Cada protocolo de enrutamiento RIP, OSPF, etc, tiene su propia base de datos. La base de datos es distinta de la tabla de enrutamiento.

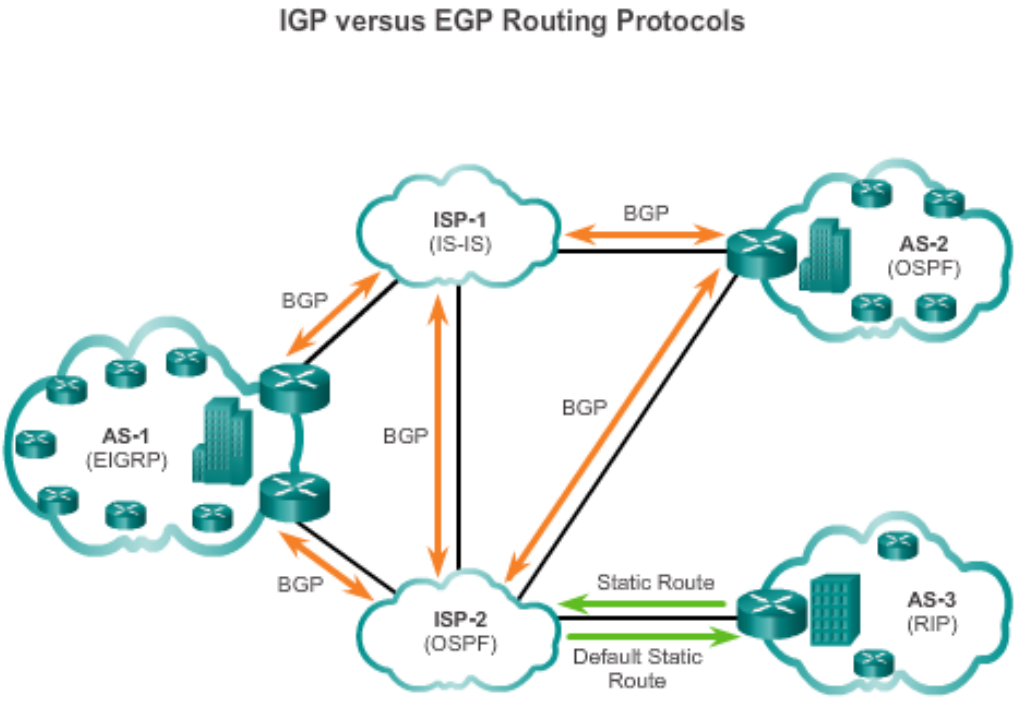
El algoritmo usado es importante en cada protocolo de enrutamiento.

Una ruta estática tiene una distancia administrativa por defecto de 1.

Si tenemos u protocolo dinámico, el que sea, y por ejemplo tuviera una distancia administrativa de 120, para crear una ruta flotante alternativa, necesariamente debería tener un valor superior al de 120, o sea, a partir de 121.

RIP V2

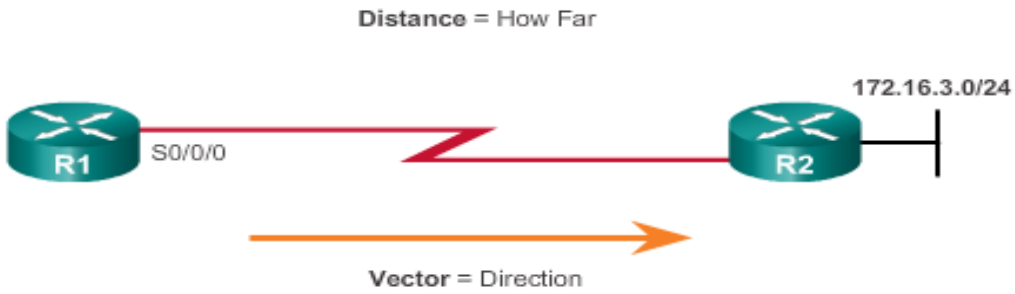
Todos los Routers de una red van intercambiando información. Cuando todos los Routers tienen ya la información de los demás, se llama Convergencia. El tiempo de convergencia del protocolo RIP es más elevado que en EIGRP y OSPF.



Dentro de los protocolos de enrutamiento hay que distinguir entre los **IGP** (Interior Gateway Protocols) tales como RIP, EIGRP, OSPF, IS-IS. Estos protocolos son los que se usan de manera interna.

Por otro lado están los **EGP** (Exterior Gateway Protocols) como puede ser el BGP, que son los que se usan para conectar ISPs entre sí). EGP consume muchos recursos.

Protocolos basados en **VECTOR DISTANCIA**: La distancia se mide en torno a la métrica: cuán lejos está una ruta de otra. El Vector (cómo se alcanza) es la dirección de la IP de de próximo salto.



Para R1, 172.16.3.0/24 es un salto de lejos (distance). Lo alcanza a través de R2 (vector).

Rip y EIGRP son protocolos basados en vector distancia. Utilizan Los Routers como “carteles”

Protocolos basados en **ESTADO DE ENLACE**: Todos los Routers se conocen toda la tipología de la red. Cada Router, al tener toda la información, sabe qué ruta tomar en cada situación. Cada Router difunde toda su información de todo lo que tiene directamente conectado. Así, todos los Routers conocen todo. A raíz de sa información, cada Router toma sus decisiones.

Si una red se eliminara, o si se incorporara una nueva a un Router, éste lo difundiría todos de manera inmediata. Así cara Router en la LAN actualizará su tabla de enrutamiento.

Los protocolos basados en estado de enlace consumen muchos recursos.

Protocolos **CLASSFULL (Con clase)**: Redes A,B y C. En las actualizaciones con protocolos classfull, no se envía las máscara de red, porque al ser con clase, se le presupone la máscara (al no contemplar ni subredes ni super redes sumarizadas)

Protocolos **CLASSLESS (Sin clase)**: Contemplan también super redes, subredes, VLSM, SLSM, CIDR y sumarizaciones. Aquí sí que se trabaja con máscara.

CARACTERISTICAS DE LOS DISTINTOS PROTOCOLOS DE ENRUTAMIENTO

	Distance Vector				Link State	
	RIPv1	RIPv2	IGRP	EIGRP	OSPF	IS-IS
Speed Convergence	Slow	Slow	Slow	Fast	Fast	Fast
Scalability - Size of Network	Small	Small	Small	Large	Large	Large
Use of VLSM	No	Yes	No	Yes	Yes	Yes
Resource Usage	Low	Low	Low	Medium	High	High
Implemenation and Maintenance	Simple	Simple	Simple	Complex	Complex	Complex

La métrica en **Rip** se mide por los saltos. Rip está constantemente actualizándose (Cada 30 segundos). Su algoritmo es: Belman-Ford. La limitación de Rip son 15 saltos y su distancia administrativa es de 120. RIP 1 y 2 utilizan el puerto 520.

RIPv1 versus RIPv2

Routing updates broadcasted every 30 seconds

Characteristics and Features	RIPv1	RIPv2
Metric	Both use hop count as a simple metric. The maximum number of hops is 15.	
Updates Forwarded to Address	255.255.255.255	224.0.0.9
Supports VLSM	✗	✓
Supports CIDR	✗	✓
Supports Summarization	✗	✓
Supports Authentication	✗	✓

Updates use UDP port 520

Completando el cuadro de arriba diremos que en RIPNG (Rip para ipv6)

- las actualizaciones se hacen a través de la dirección: FF02::9,
- el puerto es el 521, es classless: admite vlsm, cidr y sumarización al igual que Rip 2
- y en cuanto a seguridad es IPsec de tal forma que admite: autenticación, confidencialidad e integridad.

La métrica en **EIGRP** es compuesta (métrica compleja).

EIGRP solo se actualiza cuando hay algún cambio. Su algoritmo es Dual.

- En EIGRP las actualizaciones se difunden a través de la dirección 224.0.0.10 / FF02::A
- Es de tipo classless, permite autenticación en su versión ipv4 e IPsec en IPv6. Las actualizaciones son **limitadas y parciales**.
- Emiten 5 tipos de mensajes: Hellow, Update,Query, Reply y ACK.

EIGRP está enviando constantemente un "Hellow" para ver si sus vecinos están activos. Si hubiera algún vecino que se hubiera caído, lo eliminará o cambiará de su tabla de enrutamiento.

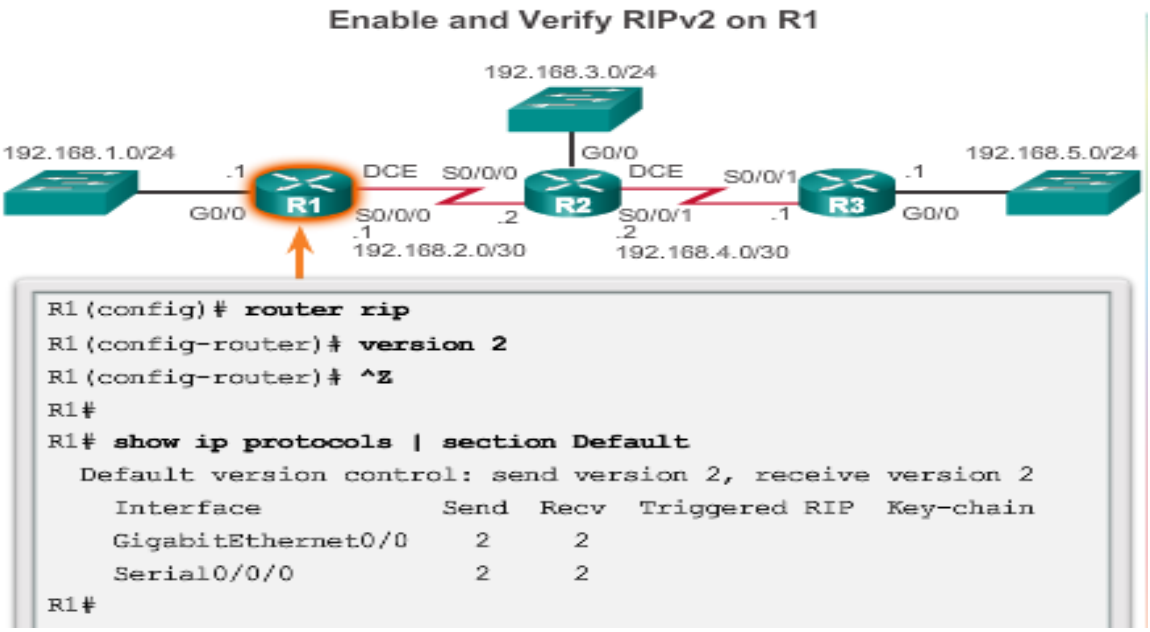
Distancia administrativa: se define como la confiabilidad de la ruta de origen. Las distancias administrativas por defecto que tiene EIGRP son:

- Rutas sumarizadas: 5
- Rutas internas: 90
- Rutas importadas: 170

Con el comando **show ip protocols** vemos los protolos en la tabla de enrutamiento de un Router

La métrica en OSPF utiliza en ancho de banda: COSTO. Su algoritmo es Dijstra.

Configuración RIP:



1- Habilitamos el protocolo: **R(Config)#router rip**
2- Si queremos la versión 2 de Rip: **R(Config-router)#**version 2
3- Declaramos las redes que el router ve directamente: **R(Config-router)# network + red**
Al estar poniendo el comando network, estamos declarando la red para las actualizaciones y además la dirección de envío de estas actualizaciones.
*la métrica en Rip es 15, porque es el número mayor de saltos que alcanza. Si su métrica fuera 16 significaría que el protocolo es inválido.
*Cuando un protoloco está en Rip 1, evidentemente manda todo en RIP 1, pero podría recibir en RIP 2 también, lo uínco que al ser Rip 1 Classfull y no saber de subredes ni super redes, no haría caso a la máscara que le llegara.

Cuando habilitamos RIP 2 no hay que olvidar deshabilitar la autosumarización que hace por defecto, con el comando R(Config-router)#**no auto-summary**

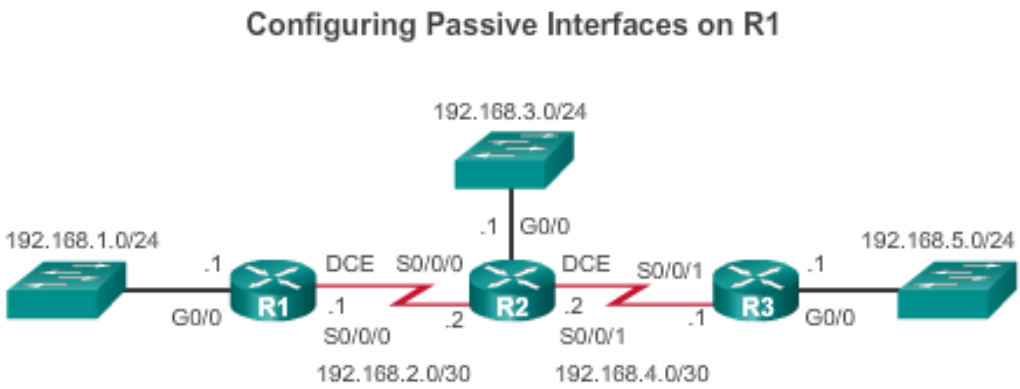
Rip 1 y 2 activan automáticamente la sumarización, de tal forma que si tuviéramos las redes:

- 172.16.10.0 /24
- 172.16.20.0 /24
- 172.16.30.0 /24
- 172.16.40.0 /24

Si no aplicáramos el comando no auto-summary, sólo enviaría la red sumarizada: 172.16.0.0 /16.

Rip 1 hace la autosumarización por defecto. Rip 2 hace lo mismo pero añade la máscara.

Como Rip está constantemente actualizándose y ocupando ancho de banda y recursos, con el comando **R(Config-router)# passive-interface** + la interface, estaremos desactivando las actualizaciones por esa inetrface, pero no impedirá que esa red se siga publicando.




```
R1(config)# router rip
R1(config-router)# passive-interface g0/0
R1(config-router)# end
R1#
R1# show ip protocols | begin Default
  Default version control: send version 2, receive version 2
    Interface              Send  Recv  Triggered RIP  Key-chain
    Serial0/0/0            2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.168.1.0
    192.168.2.0
  Passive Interface(s):
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.2           120      00:00:06
  Distance: (default is 120)

R1#
```

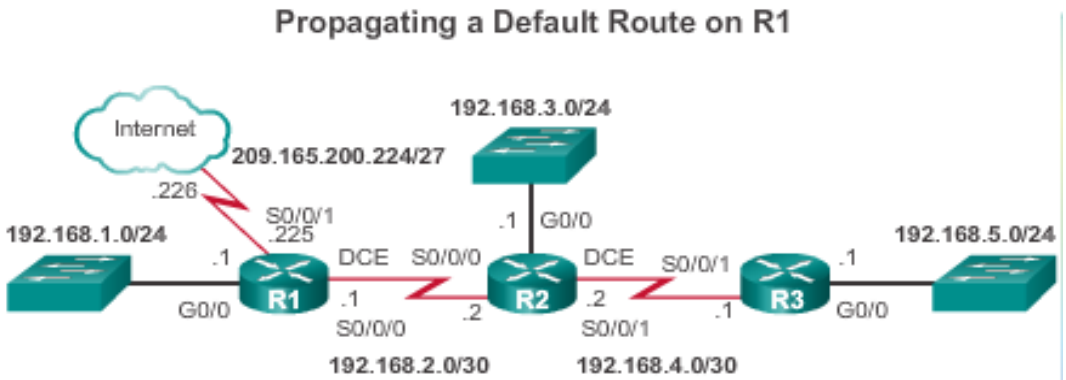
Para expandir una ruta por defecto se hace con el comando **R(Config-router)#default-information originate.**

De tal forma que:

1º: Creamos las ruta por defecto: R(Config)#ip route 0.0.0.0 0.0.0.0 s0/0/1 209.165.200.226

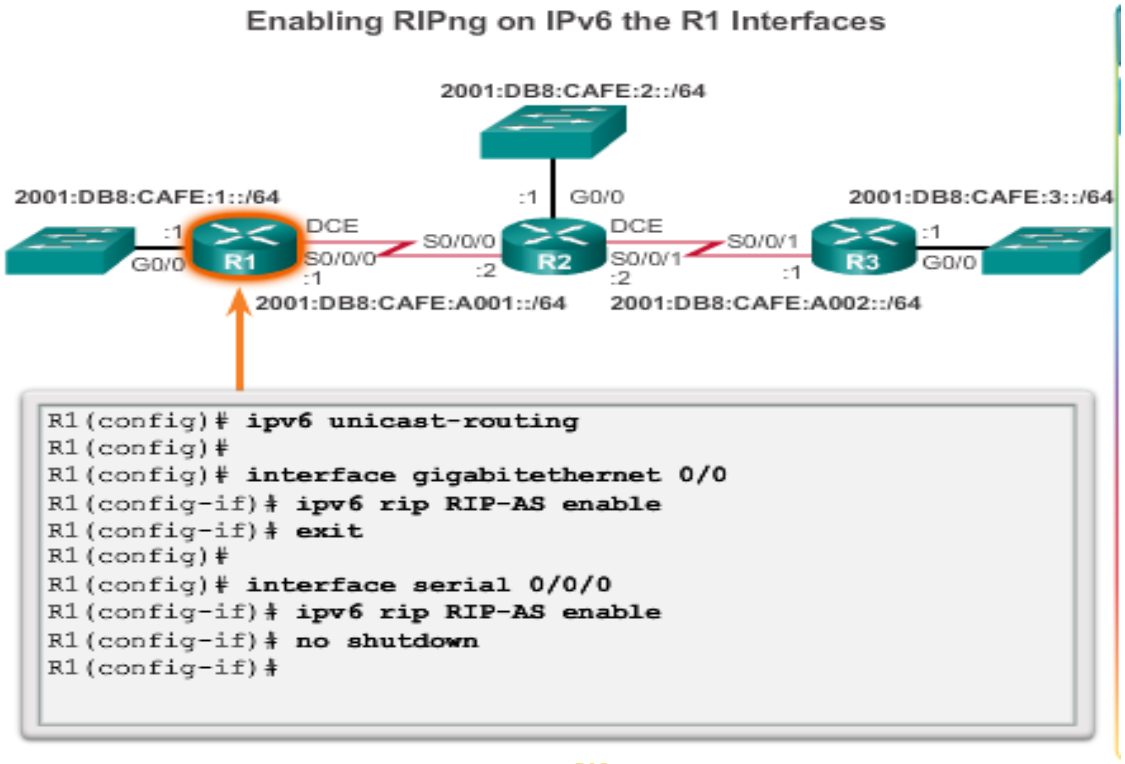
2º: Habilitamos el protocolo Rip

3º: Le decimos con el comando **R(Config-router)#default-information originate** que expanda esa ruta por defecto.



RIP IPV6

El protocolo Rip en IPv6 se activa con el comando: ipv6 unicast-routing



De tal forma que:

1º- Se habilita Rip en ipv6: **R(Config)#ipv6 unicast-routing**

2º- Le ponemos nombre al protocolo que vamos a aplicar: **R(Config)#ipv6 router rip Cisco**

*Podemos poner el nombre que queramos. En el caso de arriba le ha llamado RIP-AS, nosotros le estamos llamando ahora Cisco

El prompt cambiará a **R(Config-rtr)#**

3º Nos metemos en cada interface:

R(Config-rtr)#interface s 0/0/0

4º- - Habilitamos el nombre que hemos puesto a cada interface del router con el comando:

R(Config-if)#ipv6 rip Cisco enable

Así por cada interface.

El comando en ipv6 para ver los protocolos de la tabla es: **show ipv6 protocols**

Tipos de rutas:

- **Rutas de Nivel 1**
- **Rutas de Nivel 1 Padre**
- **Rutas de Nivel 2**
- **Rutas Finales**

Las rutas de nivel 1 son aquellas con clase , super redes y rutas por defecto

Una ruta de Nivel 1 también puede ser una ruta final.

Una ruta final es aquella que tiene una Ip de próximo salto o una interface de salida.

Todas las rutas de nivel 1 son finales.

Ruta de nivel 1 padre: se crean cuando hay subredes.

- 10.0.0.0 → Ruta de nivel 1 Padre
- 10.10.10.0 /24
- 10.10.20.0 /24 } Rutas de nivel 2 o hijas
- 10.10.30.0 /24

Las subredes siempre son rutas finales.

Si todas las subredes tienen la misma máscara la red padre también tendrá esa máscara.

Si alguna subred tuviera distinta máscara, la red padre tendría la máscara de origen de clase: prefijo de clase.

Una red padre nunca tendrá una ip de último salto o interface de salida.

A toda ruta de nivel 2 o hija, se le asigna una ruta padre. Sii solo hubiera una sola hija o subred, la máscara será la de la subred, no la de prefijo de clase.

Las rutas de nivel 2 siempre son rutas finales.

Ejemplos:

- 172.16.0.0 /16 S 0/0/1 Ruta Nivel 1 y Final
- 10.0.0.0 /8 Nivel 1 Padre
- 10.10.10.0 /24 Nivel 2 hija y Final

Los procesos de elección por parte del router para elegir ruta son:

1- Si la mejor opción es una ruta de un nivel 1 ruta final, entonces esta ruta se utiliza para enviar el paquete.
2- Si la mejor opción es una ruta padre de nivel 1, continúe con el siguiente paso.
3- El router examina las rutas secundarias (las rutas de subred) de la ruta padre como mejor opción.
4- Si hay una coincidencia con una ruta secundaria de nivel 2, esa subred se utiliza para enviar el paquete.

5- Si no hay coincidencia con ninguna de las rutas secundarias de nivel 2, se continua con el siguiente paso.
6- El router sigue buscando rutas de nivel 1 de superred en la tabla de enrutamiento como mejor opción, incluyendo la ruta por defecto, si es que existe.
7- Si hay un menor posibilidad de opción con una ruta de nivel 1, se elegirá una superred o una ruta por defecto para enviar el paquete.
8- Si no hay coincidencia con ninguna ruta en la tabla de enrutamiento, el router descarta el paquete.

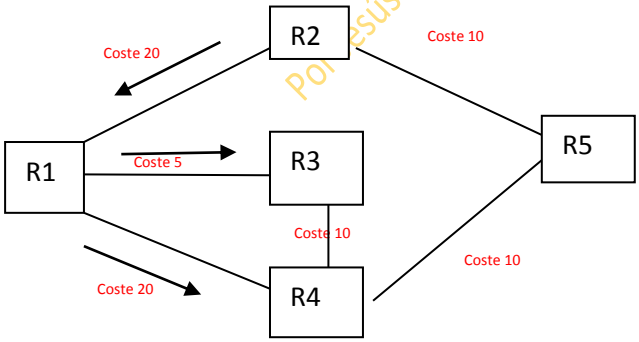
Cuando se pone una ruta por defecto 0.0.0.0 0.0.0.0 hay que decir también que es la Default-information originate para expandirla. R(Config-router)#default-information originate.

PROTOCOLOS DE ESTADO DE ENLACE: OSPF

OSPF utiliza base de datos t tabla de enrutamiento; son cosas distintas.

Se conocen también como algoritmos Shortest Path First (SPF=primero la ruta más corta). Estos protocolos se crean sobre la base de algoritmos SPF de Dijkstra.

A cada enlace se le asigna un coste, que está muy relacionado con el ancho de banda



Así por ejemplo, para ir de R2 a R3 será mejor por R5. Los costes son menores.

A mayor calidad de enlace, menor será el coste. Siempre se cogen las rutas con menor coste, independientemente de los saltos.

Los routers conocen sus redes directamente conectadas. Intercambian saludos. Cada router crea su propio paquete de estado (LSP) que incluye información sobre sus vecinos, como la ID, el tipo de enlace y el ancho de banda.

También crea su propia base de datos (tipológica). Así en la base de datos tendrá también el paquete de estado de sus vecinos. Se manda la información entre todos. Así cada router tendrá su propio paquete y su propia base de datos, donde guarda el paquete de estado de los demás routers.

Cada router hace una copia en su base de datos de cada paquete nuevo que le llega. Un router mandará a sus vecinos los paquetes que reciba. Así R1------R2------R3 R3 tendra el paquete de R1 por que se lo habrá mandado R2. Ojo solo se mandan los paquetes, no la base de datos.

Cuando todos los routers conocen a todos se produce lo que se llama convergencia. Así el tempo de convergencia es el tiempo que tardan todos los routers en conocerse.

Información que tiene un paquete:

- Tiene información de las interfaces de cada router.
- Red a la que está conectado por cada interface.
- Dirección de red.
- Tipo de red.
- Dirección IP.
- Coste del enlace.
- Vecinos a los que está conectado ese enlace(interface)

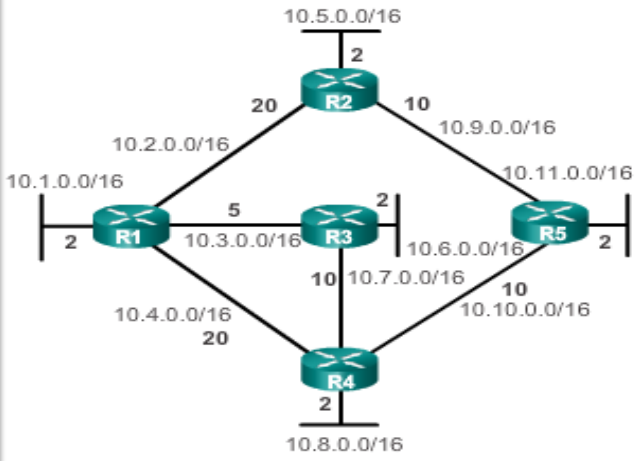
La información solo se manda entre vecinos. No es un broadcast.

Así los routers al recibir toda esa información se configuran mapas para calcular cual es la mejor ruta y su coste para llegar a cualquier sitio. Entiéndase por enlace la interface.

En la base de datos está la tipología de toda la red: cuando se han intercambiado todos los LSPs de tal forma que todos los Routers tienen la misma base de datos y por tanto, todos conocen la tipología de la red. Una vez que se tiene esta información, ya son los routers los que deciden que camino seguir hasta alcanzar una determinada red. Costo=Métrica.

Resulting SPF Tree of R1

Destination	Shortest Path	Cost
10.5.0.0/16	R1 → R2	22
10.6.0.0/16	R1 → R3	7
10.7.0.0/16	R1 → R3	15
10.8.0.0/16	R1 → R3 → R4	17
10.9.0.0/16	R1 → R2	30
10.10.0.0/16	R1 → R3 → R4	25
10.11.0.0/16	R1 → R3→ R4→ R5	27

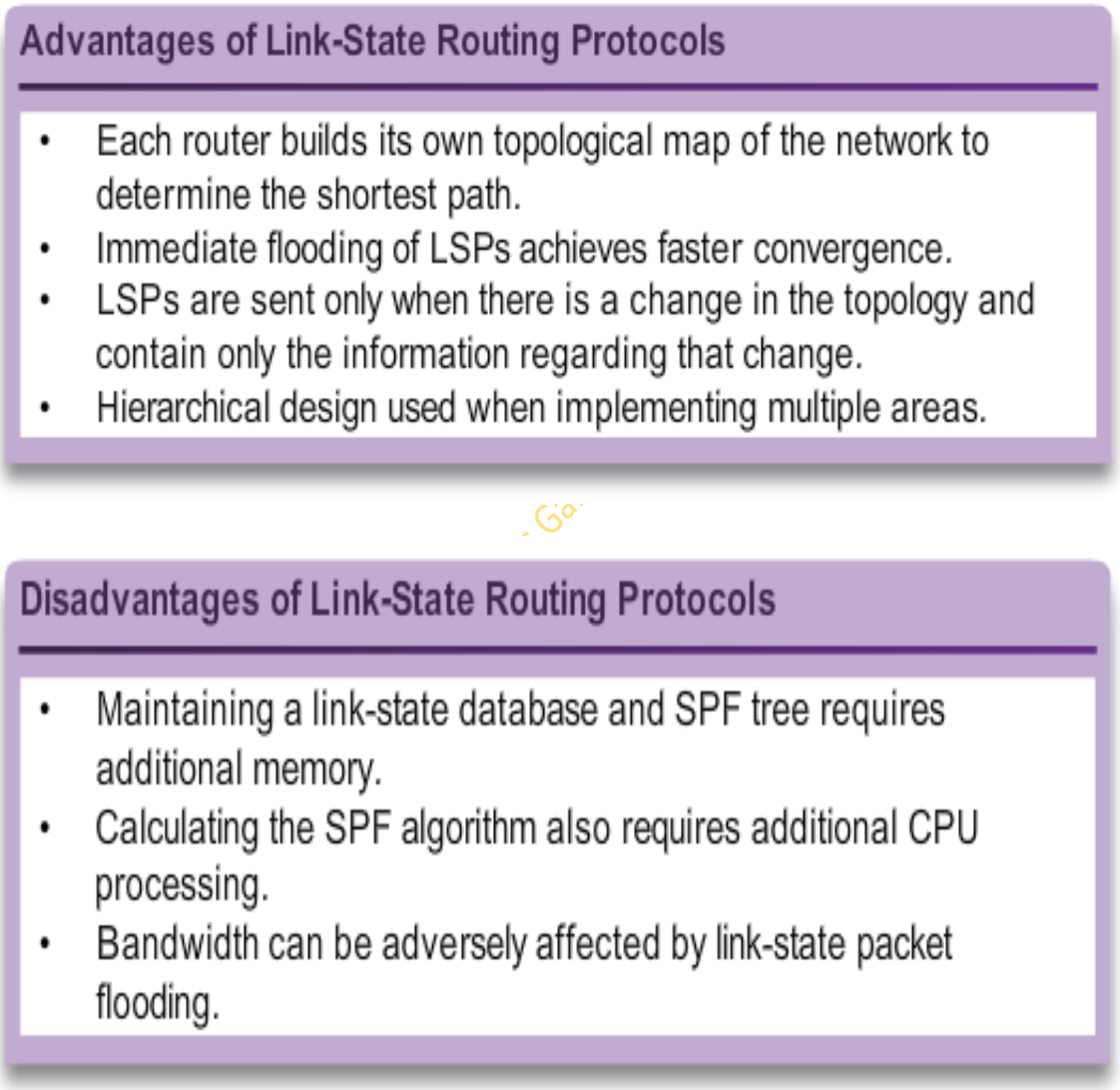


En OSPF también se puede usar el comando comando **R(Config-router)# passive-interface** + la interface.. Si dos routers conectados cada uno tuviera distintos protocolos, ejemplo: OSPF/RIP

habrá que poner el passive interface en la interface que les une para evitar que se manden los paquetes “Hellow”.

El algoritmo Dijstra es quien construye el **SPF TREE** que son los mejores caminos de un punto inicial con el cual se generará la tabla de enrutamiento. Tras todo esto se conseguirá la **Convergencia.**

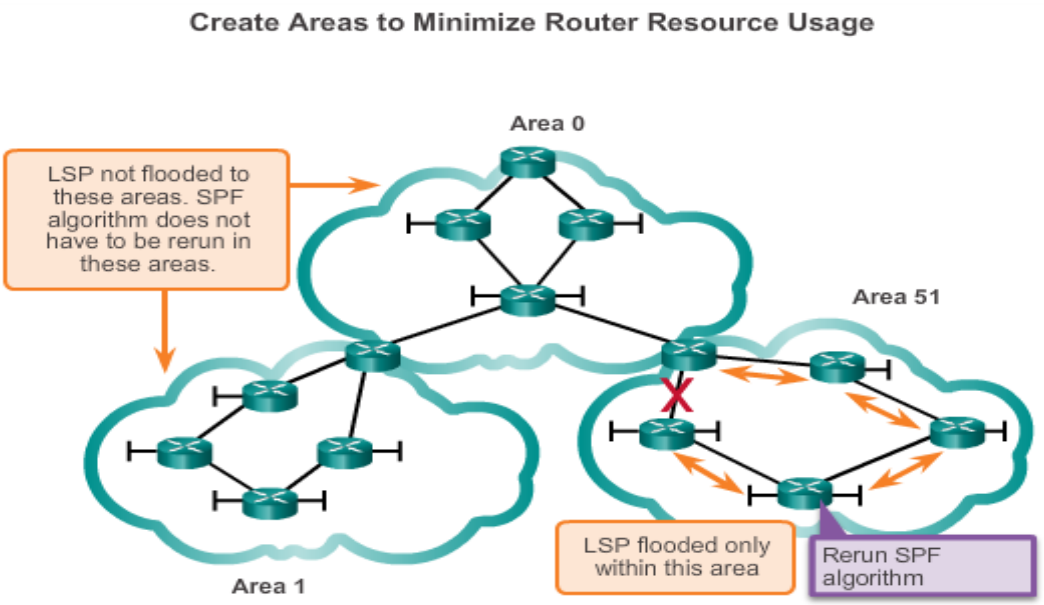
Ventajas y desventajas de los protocolos de estado de enlace:



Las tablas de enrutamiento siempre contemplan los caminos de menor coste: **SPF=Shortest Path First**

En OSPF las actualizaciones que se mandan entre sí los routers son parciales y limitadas al igual que en EIGRP.

OSPF es un protocolo que nace jerárquico. RIP y EIGRP son protocolos lineales. Los protocolos de estado de enlace OSPF IS-IS requieren un mayor consumo de cpu y ram y sus bases de datos son mayores que los protocolos basados en vector distancia.



La creación de áreas en OSPF es para evitar inundaciones por cambios en alguna red o equipo. De esta manera, solo el área afectada por algún cambio deberá ejecutar de nuevo el algoritmo Dijstra. Las demás áreas posteriormente solo actualizarán sus rutas sin tener que ejecutar de nuevo el algoritmo Dijstra e inundar a todos sus vecinos.

Capitulo 8: OSPF

OSPF manda distintos paquetes: Hellow, DBP, LSR, LSU, LSACK.

El encabezamiento del paquete OSPF:

1- Saludo
2- Descripción de datos (DBP)
3- Solicitud de estado de enlace (LSR)
4- Actualización de estado de enlace (LSU)
5- Acuse de recibo de estado de enlace (LSAck)

Los paquetes de saludo detectan vecinos y establecen adyacencias. Públicas pautas sobre qué routers deben estar de acuerdo para convertirse en vecinos.

Utilizados por redes de accesos múltiples para elegir un router designado (DR) y un router designado de respaldo (BDR).

El contenido del paquete de saludo tiene el identificador del router que relaiza la transmisión.

Intervalos de saludo:

- Por multicast (224.0.0.5)
- Enviados cada 30 segundos para segmentos NB MA
- Intervalo muerto. Para dar por muerto a un router vecino es 4 veces el tiempo de saludo.

Los paquetes de saludo también tienen información que se utiliza en la selección del router designado (DR). El DR es responsable de la actualización de todos los otros routers OSPF.

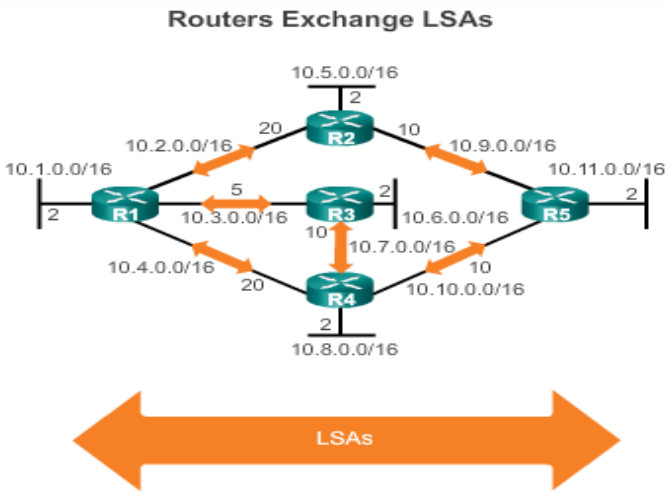
El router designado de respaldo (BDR) asume las responsabilidades del DR si este falla.

Actualizaciones de de estado de enlace.

- Cuando empieza a converger la red.
- Cuando hay un cambio topológico.

El algoritmo OSPF es igual que SPF.

La distancia administrativa en OSPF es 110.



La publicación de estado de enlace es el LSA

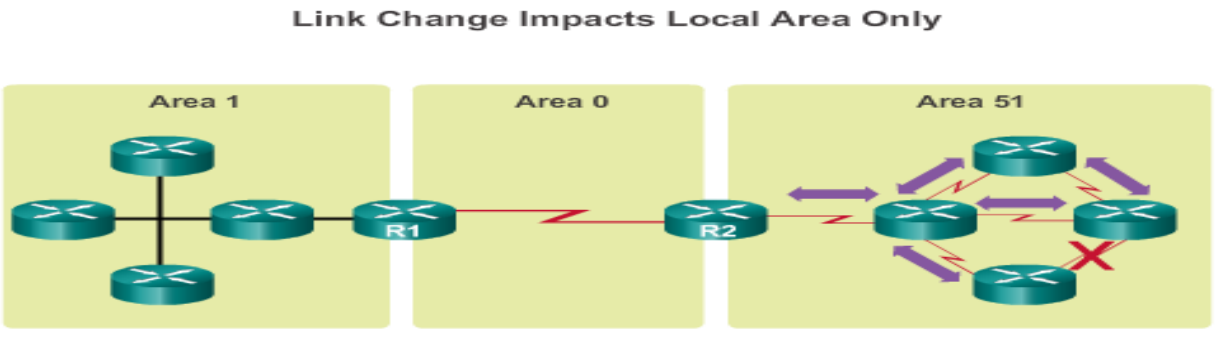
Los LSU son los que contienen la información LSA. Cada LSA es información de cada red directamente conectada. Un Router con 4 redes directamente conectadas, mandará4 LSAs mediante 1 solo LSU. Dentro de un LSU puede haber 1 o muchos LSAs. Hay 11 tipos de LSAs. La base de datos de un router en OSPF lo que alberga son precisamente las LSAs.

Además OSPF establece adyacencias.

OSPF Multitarea:

El área o backbone siempre tiene que estar creado. Para usar OSPF hay que crear áreas necesariamente. Se recomienda un máximo de 50 Routers por área.

Todas las áreas normales (así se llaman todas las áreas que no son la 0) se supone que tienen



- Link failure affects the local area only (area 51).
- The ABR (R2) isolates the fault to area 51 only.
- Routers in areas 0 and 1 do not need the run the SPF algorithm.

que estar conectadas con el área 0. El área 0 se le llama también “Área de tránsito”

En esta diapositiva, el R2 pertenece al área 0 y a la 51 a la vez y R1 a a la 0 y a la 1.

Al menos una interface de un Router debe estar conectada al área 0 de backbone. Si no es así, será un “**área aislada**”

Al estar dividido en áreas, caso de haber un fallo, o una incorporación de algún host, solo afectará a esa área.

Una virtual link es la forma de conectar el área 0 con un área aislada. No se recomienda tener virtual links.

Tipos de paquetes:

OSPF Packet Descriptions

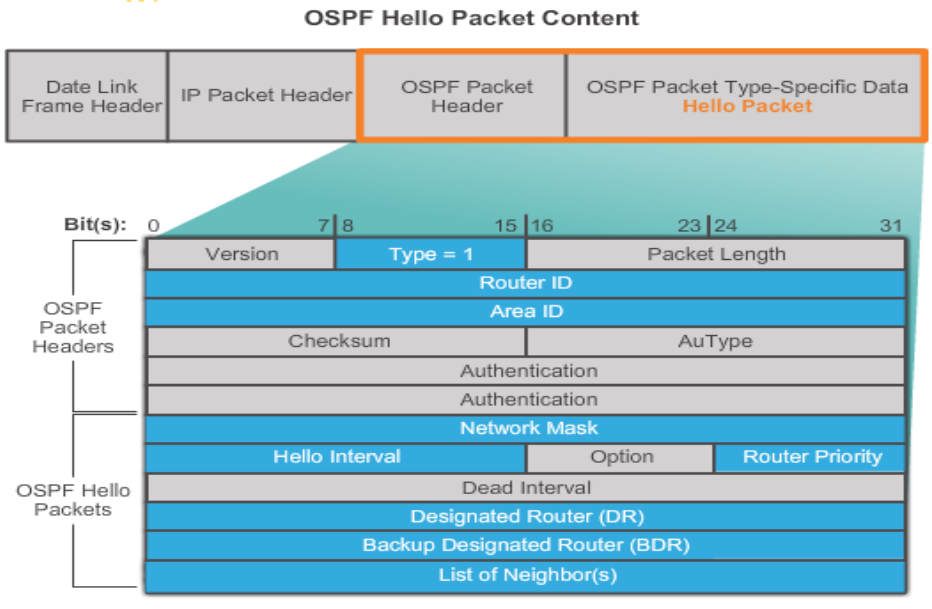
Type	Packet Name	Description
1	Hello	Discovers neighbors and builds adjacencies between them
2	Database Description (DBD)	Checks for database synchronization between routers
3	Link-State Request (LSR)	Requests specific link-state records from router to router
4	Link-State Update (LSU)	Sends specifically requested link-state records
5	Link-State Acknowledgment (LSAck)	Acknowledges the other packet types

Router ID: Dirección ipv4: identificador único. Es 1 por Router

Área ID: Área de pertenencia de un Router

Hellow interval: Periodicidad con la que se emite el “Hellow”

Una red multi-acceso es aquella que está conectada mediante un SW. Los paquetes hellow se actualizan cada 10 seg.



En una red wan con Frame Telay, Atm, etc, es cada 30 seg.

Intervalo muerto: es el período de tiempo que espera un Router a recibir los “hellow”. Los tiempos son x4: esto es: Si lo normal son 10 seg en llegar, si tarda 40 ese paquete se eliminará

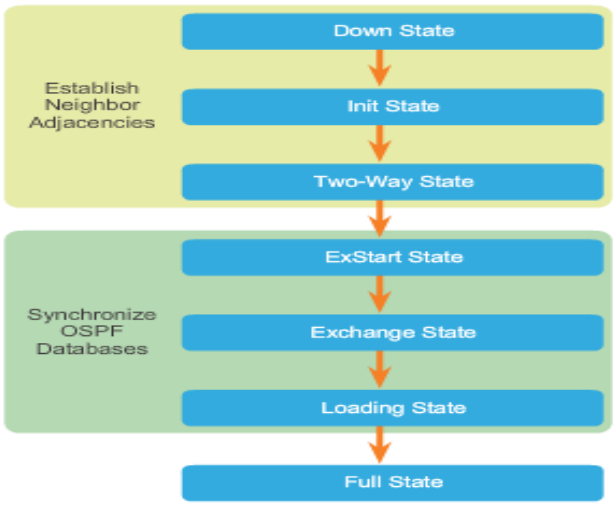
de la tabla de enrutamiento. En redes wan, como el tiempo normal es 30 seg, su intervalo muerto será de 2 minutos.

El proceso operacional de OSPF es:

1 Crea adyacencias.
2 Intercambia información
3 Calcula la mejor ruta
4 Alcanza la convergencia: Cuando todos los Routers de se conocen entre sí y su topología.

When an OSPF router is initially connected to a network, it attempts to:

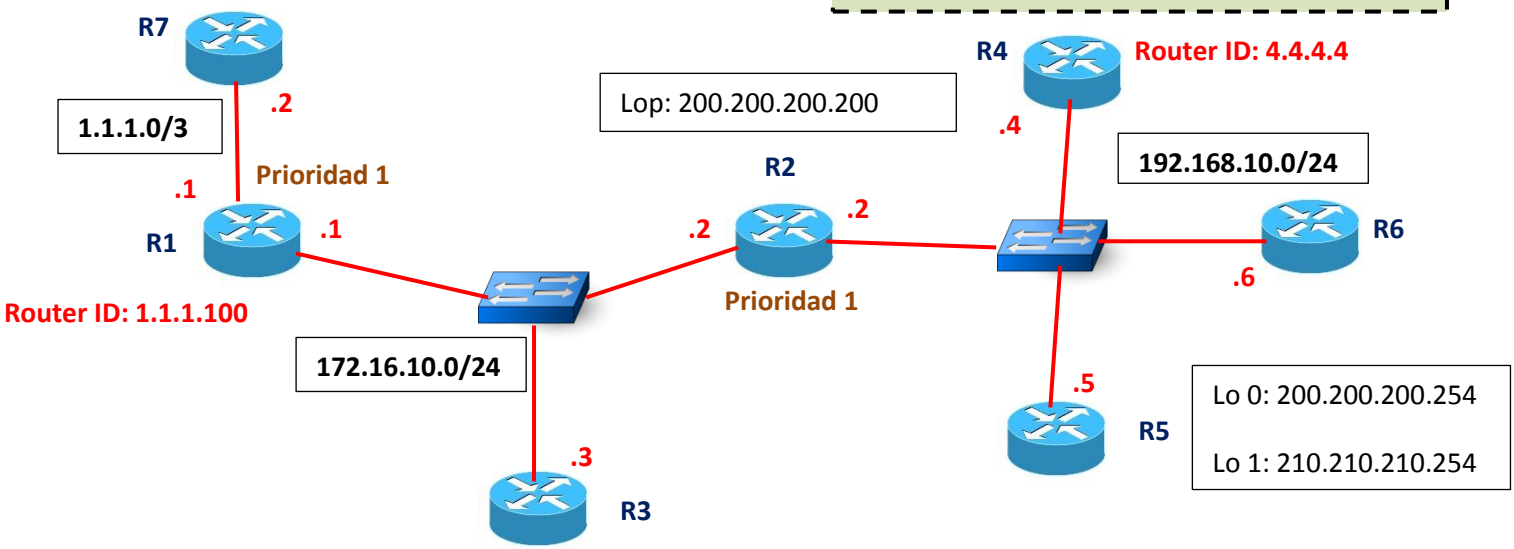
- Create adjacencies with neighbors
- Exchange routing information
- Calculate the best routes
- Reach convergence
- OSPF progresses through several states while attempting to reach convergence.



Down State

INI State

Two Way State: en este momento es cuando se crea la Tabla de Vecinos



Para saber el ID de un Router hay que atender a tres parámetros:

1- **Dirección puesta de manera Manual [Router-ID <ipv4>]**
2- **Dirección loopback más alta**
3- **Dirección IP de interface activa más alta**

Para elegir el DR/BDR:

1- **Prioridad de interface + alta** (generalmente están todos a 1)
2- **Router ID + alto**

Elegir el Router ID se puede hacer manualmente.

La dirección del Router ID puede ser la que queramos.

Si no se le mete una dirección manualmente, tomará como ID la loopback más alta (Interface Virtual) activa (en no shutdown)

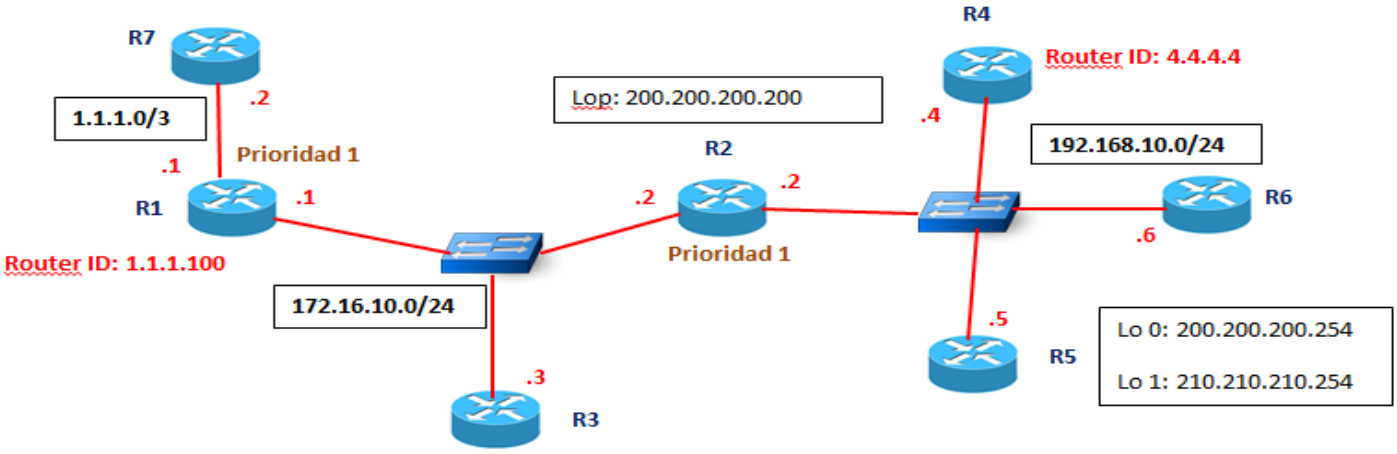
Si tampoco hubiera loopback configurada, se tomaría como Router ID la dirección IP de la interface activa más alta.

DR es el Router Designado.

BDR es el Router de respaldo.

En OSPF las interfaces tienen por defecto una prioridad de valor 1.

En el ejemplo, para elegir el Router ID en R1 sería la 1.1.1.100 porque la hemos hecho manualmente.



Para saber el ID de un Router hay que atender a tres parámetros:

1- Dirección puesta de manera Manual [Router-ID <ipv4>]
2- Dirección loopback más alta
3- Dirección IP de interface activa más alta

Para elegir el DR/BDR:

1- Prioridad de interface + alta (generalmente están todos a 1)
2- Router ID + alto

En R2 sería la loopback porque no se le ha configurado manualmente ninguna ID.

En R3 sería la 172.16.10.3 porque no tiene ni ID configurada manualmente ni loopback, de ahí que tomemos la dirección IP de la interface activa más alta que en este caso es 172.16.1.3

La selección de BDR y DR se hace por segmentos, por eso un Router podría ser a la vez DR de un segmento y BDR en otro. De hecho en el ejemplo de arriba, el DR en el segmento de la izquierda es R2 y el BDR R3. Sin embargo en la zona de la derecha, el DR será R5 por tener la loopback más alta y el BDR será R2.

Actualizaciones de de estado de enlace.

- Cuando empieza a converger la red.
- Cuando hay un cambio topológico.

El algoritmo OSPF es igual que SPF.

La distancia administrativa en OSPF es 110.

Autenticación: se puede poner usuario y contraseña. Autenticar es comprobar que alguien es quien dice que es.

Configuración de OSPF:

```
R1(config)# router ospf 10
R1(config-router)# ?
Router configuration commands:
  auto-cost               Calculate OSPF interface cost
                          according to bandwidth
  network                 Enable routing on an IP network
  no                      Negate a command or set its defaults
  passive-interface       Suppress routing updates on an
                          interface
  priority                OSPF topology priority
  router-id               router-id for this OSPF process
```

R1(Config)#router ospf + identificador del proceso (este identificador es solo para local. Valor decimal entre 1 y 65535)

Configurar OSPF:

1º Le decimos al router que va a trabajar en ese protocolo:

R1(Config)#router ospf + identificador del proceso (este identificador es solo para local. Valor decimal entre 1 y 65535)

R1(Config)#router ospf 1

2º Le metemos su Router ID

Comando router-id: para dar el nombre a un identificador. Valor decimal.

R1(Config)#router ospf 1

R1(Config-router)#router-id dirección ip Así se asigna un identificador.

Le decimos las redes que van a trabajar balo OSPF (ojo, la máscara wildcard)

3º Le decimos la red, le ponemos la mascara wildcard y el área en el que va a trabajar

R1(Config-router)#network 172.16.1.16 0.0.0.15 area 0 (generalmente es la 0. No tiene porque coincidir con la id del proceso)

Eso es todo!!!

Para configurar una interface en modo pasivo usando ospf, el commando es:

R1(Config)#router ospf + identificador del proceso

R1(Config-router)#passive-interface gi 0/0

En este caso no es como en eigrp. No hace falta que sea el mismo número para todos.

R1(Config)#router ospf 1

R1(Config-route)#network 172.16.1.16 0.0.0.15 area 1

```
R1(config)# router ospf 10
R1(config-router)# network 172.16.1.0 0.0.0.255 area 0
R1(config-router)# network 172.16.3.0 0.0.0.3 area 0
R1(config-router)# network 192.168.10.4 0.0.0.3 area 0
R1(config-router)#
R1#
```

Al poner network hay que poner la dirección de red + la máscara wildcard + area id

Máscara wildcard: sabiendo cual es la máscara de subred, la fórmula es: a la máscara total le resto la máscara de subred:

255.255.255.255

_

255.255.255.0

Wildcard: 0.0.0.255

En wildcard los bits que están a 0 tienen que coincidir y los bits a 1 son indiferentes:

Ej: 192.168.8.0 ⇨ 0.0.0.255

Una red es par cuando el último bit está a 1 . _ _ _ _ _ _ _ _
1
Si el último bit está a0 la red será impar.

Ej: 192.168.8.14 máscara wildcard asociada: 0.0.0.255

Si para la red 192.168.7.4 se le asociara una mascara wildcard 0.255.255.0 solo serían válidas las redes que empezaran por 192 y terminaran en 4.

Otro ej: máscara wildcard asociada de una mascara subred 255.255.255.252=0.0.0.3

Máscara total: 255.255.255.255

Máscara subred: 255.255.255.252

Máscara wildcard: 0. 0. 0. 3

R1(Config-route)#network + red + mascara wildcard +area

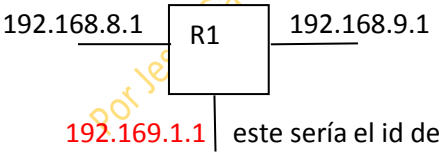
ID: Identificador del router:

Siempre es una ip. El criterio que sigue OSPF para asignar una ip a un router es:

Si hay comando router-id las ip será la que pongamos

Si no tiene ese comando, se pondrá como identificador la dirección ip más alta de las interfaces loopback.

Si no hay tampoco interfaces loopback configurados, se escoge la dirección ip más alta de las interfaces conectadas al router.



este sería el id del router (la ip más alta)

Comandos de verificación OSPF:

Show ip protocols

Show ip ospf

Show ip ospf interface

Interface loopback: Interface virtual

Configurar loopback en un router. Es una dirección ip virtual. Se usa mucho como administración.

R1(Config)#interface loopback 0

R1(Config-if)#ip address 55.55.55.55 (le ponemos la dirección que queramos) **255.255.255.255** (esta máscara la las loopback en principio será siempre esa)

```
R1(config)# interface loopback 0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# end
R1#
```

Interesa tener loopback porque así puedes llegar a ella por cualquier interface física. Así, si una interface física se cae, se podrá llegar a ese router por otra interface física mediante la interface loopback virtual.

La interface loopback no puede fallar=estabilidad OSPF

Comando router-id: para dar el nombre a un identificador. Valor decimal.

R1(Config)#router ospf 1

R1(Config-router)#router-id dirección ip Así se asigna un identificador.

```
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
% OSPF: Reload or use "clear ip ospf process" command, for
this to take effect
R1(config-router)# end
R1#
*Mar 25 19:46:09.711: %SYS-5-CONFIG_I: Configured from
console by console
```

Para modificar el identificador, comando:

R1 clear ospf + valor decimal

*ojo con esto: generalmente hay que hacer un reload para que cambie el ID.

Comando show ip ospf neighbor. Si no se muestra el identificador del vecino, es que no hay adyacencias. Si no hay adyacencias no habrá intercambio sobre la información del estado de enlace y por tanto, los árboles y las tablas de enrutamiento serán inexactas.

METRICA DE OSPF

El coste de una ruta se mide/calcula con la fórmula 10^8 dividido entre al ancho de banda real de la interface. De ahí que, un enlace por fibra óptica de 1 Gb=100.000.000 de ancho de banda, nos dé un valor de 1= al coste. Por eso un enlace de peor calidad, por fast Ethernet por ejemplo, de 1.000.000 al dividir 10^8 nos daría un coste de 100.

El ancho de banda de referencia por defecto es 100 Mbps.

Se puede modificar usando el comando: autos-cost reference-bandwidth.

El coste de una ruta es el coste de todos los enlaces que tiene que atravesar un paquete. El comando show interface muestra en ancho de banda de las interfaces. Ambos lados de un enlace serial deben configurarse con el mismo ancho de banda.

Para modificar virtualmente el ancho de banda es con el comando bandwidth. El comando ip ospf cost permite especificar manualmente el coste de una interface.

Ej: R1(Config)#interface serial

R1(Config-if)#ip ospf cost 1562 (el número que queramos)

Disminuyendo el coste se establece prioridad. Este comando ip ospf cost es mejor que bandwidth.

OSPF permite 5 tipos de redes, en las que pude trabajar:

1- Punto a punto.
2- Accesos múltiples con broadcast.
3- Accesos múltiples sin broadcast.
4- Punto a multipunto.
5- Enlaces virtuales.

OSPF EN REDES DE ACCESOS MULTIPLES.

Dos desafíos de las redes de accesos múltiples:

1 - Adyacencias múltiples.

2 – Flooding de LSA masivo.

Para evitar la saturación de LSA se establecen los DR y los BDR.

El criterio que se sigue es:

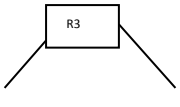
1 - Los routers que tiene mayor prioridad.

2 - Si todos tienen la misma prioridad, se toma el router con la ID más alta.

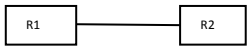
Cuando todos convergen, se establece quién es el DR Y BDR.

DRothers son todos los routers que no son ni DR ni BDR.

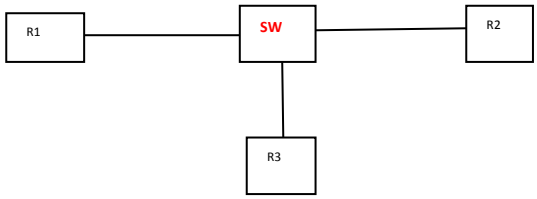
En estas redes la dirección multicast es 224.0.0.6 al DR y al BDR. El DR envía los LSA mediante la dirección multicast 224.0.0.5



Las elecciones de DR /BDR NO ocurren en las redes punto a punto:



Ocurren solo en las redes de accesos múltiples:



Criterios para la elección de DR/BDR:

1- Router con mayor prioridad.
2- BDR 2º router con mayor prioridad.
3- Si las prioridades son iguales, se tomara el router con la ID más alta.

El valor por defecto de la prioridad es 1.

El DR/BDR se elige cuando se habilita la interface del 1er router en la red de accesos múltiples.

Cuando se elige un DR, éste permance hasta que:

- El DR falla.
- El proceso OSPF en el DR falla.
- La interface de accesos múltiples en el DR falla.

Manipulación del proceso de selección (DR/BDR):

- Iniciaremos el OSPF en el router que queramos que sea el DR. El primer router que configuremos OSPF será el DR.
- Apagando la interface en todos los routers utilizando el comando no shutdown en el DR, luego en el que queramos que sea el BDR y por último en los demás routers. Esto se hace a capón.

Para manipularlo en línea de comando, usaremos el comando: ip ospf priority y le otorgamos un valor entre (0-255). R1(Config-if)#ip ospf priority (0-255)

Si ponemos el valor 0 implicará que ese router no podrá ser DR o BDR**.**

Redistribución de una ruta OSPF por defecto: R1(Config)#ip route 0.0.0.0.0.0.0.0 loopback 1

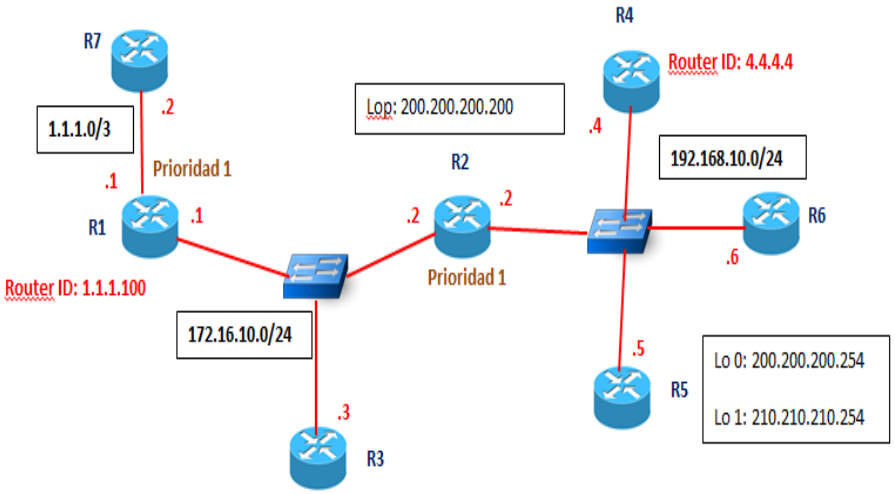
Requiere el uso del comando default-information origínate.

Para elegir el DR/BDR:

3- Prioridad de interface + alta
(generalmente están todos a 1)
4- Router ID + alto

R1(Config-router)#default-information originate.

En el ejemplo como la prioridad es siempre 1, se pasa a la interface active más alta, pore so R2 será el DR en el segment de la izquierda y R3 el BDR (comparando los Routers ID R2 es la más alta y R3 es la 2ª más alta)



En el segmento de la derecha el DR será R5 ya que su router ID es su loopback cuya dirección es 210.210.210.254 y el BDR sera R2 ya que su router ID es la loopback 200.200.200.200.

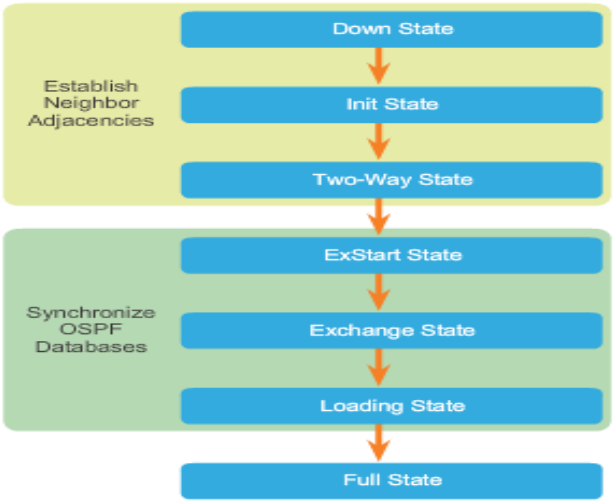
El resto de routers se llaman DRothers.

Para las **direcciones multicast**, **el DR y el BDR escuchan la dirección 224.0.0.6** BDR de manera pasiva. En caso de tener que actuar , lo hará el DR. **Los DRothers escucharán en 224.0.0.5**

Para saber el ID de un Router hay que atender a tres parámetros:

4- **Dirección puesta de manera Manual [Router-ID <ipv4>]**
5- **Dirección loopback más alta**
6- **Dirección IP de interface activa más alta**

En loading state, ya todos los LSDB son idénticos, ya que se han sincronizado y se podrá pasar a full state para crear el árbol SPF.



En OSPF también se puede configurar una passive interface:

```
R1(config)# router ospf 10
R1(config-router)# passive-interface GigabitEthernet 0/0
R1(config-router)# end
R1#
```

Verifying R1's OSPF Interfaces

```
R1# show ip ospf interface brief
Interface   PID   Area   IP Address/Mask   Cost   State   Nbrs F/C
Se0/0/1     10    0      192.168.10.5/30   15625  P2P     1/1
Se0/0/0     10    0      172.16.3.1/30     647    P2P     1/1
Gi0/0       10    0      172.16.1.1/24     1      DR      0/0
R1#
```

El comando R1#show ip ospf interface brief es importante.

Paquetes confiables:

LSU, LSR DBD son confiables ya que reciben respuesta

Paquetes no confiables:

LSAck, Hellow

En un mismo medioa físico podemos trabajar simultáneamente con OSPFV2 y OSPFV3.

En OSPFV3 la ID Router seguirá siendo IPV4 al igual que la loopback

Similitudes entre OSPFV" y PSPFV3:

OSPFv2 and OSPFv3	
Link-State	Yes
Routing Algorithm	SPF
Metric	Cost
Areas	Supports the same two-level hierarchy
Packet Types	Same Hello, DBD, LSR, LSU and LSAck packets
Neighbor Discovery	Transitions through the same states using Hello packets
DR and BDR	Function and election process is the same
Router ID	32-bit router ID: determined by the same process in both protocols

En OSPFV3 la link local es la dirección de origen.

La dirección de destino será la multicast FF02:5:FF0…

En OSPFV3 el comando network pasa a ser #ipv6 ospf “process-id” “area-id”

	OSPFv2	OSPFv3
Advertises	IPv4 networks	IPv6 prefixes
Source Address	IPv4 source address	IPv6 link-local address
Destination Address	Choice of: • Neighbor IPv4 unicast address • 224.0.0.5 all-OSPF-routers multicast address • 224.0.0.6 DR/BDR multicast address	Choice of: • Neighbor IPv6 link-local address • FF02::5 all-OSPFv3-routers multicast address • FF02::6 DR/BDR multicast address
Advertise Networks	Configured using the **network** router configuration command	Configured using the **ipv6 ospf process-id area-id** interface configuration command
IP Unicast Routing	IPv4 unicast routing is enabled by default.	IPv6 unicast forwarding is not enabled by default. The **ipv6 unicast-routing** global configuration command must be configured.
Authentication	Plain text and MD5	IPv6 authentication

Dado que en un a misma interface podemos incluir más de una IPV6, esto se puede utilizar para que en una misma interface distintas IPV6 tengan asignadas distintos “process-id”

Configuración IPV6

```
R1(config)# ipv6 unicast-routing
R1(config)#
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# description R1 LAN
R1(config-if)# ipv6 address 2001:DB8:CAFE:1::1/64
R1(config-if)# no shut
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# description Link to R2
R1(config-if)# ipv6 address 2001:DB8:CAFE:A001::1/64
R1(config-if)# clock rate 128000
R1(config-if)# no shut
R1(config-if)#
R1(config-if)# interface Serial0/0/1
R1(config-if)# description Link to R3
R1(config-if)# ipv6 address 2001:DB8:CAFE:A003::1/64
R1(config-if)# no shut
R1(config-if)# end
R1#
```

Step 1: Enable IPv6 unicast routing: **ipv6 unicast-routing**.
Step 2: (Optional) Configure link-local addresses.
Step 3: Configure a 32-bit router ID in OSPFv3 router configuration mode using the **router-id** *rid* command.
Step 4: Configure optional routing specifics such as adjusting the reference bandwidth.
Step 5: (Optional) Configure OSPFv3 interface specific settings. For example, adjust the interface bandwidth.
Step 6: Enable IPv6 routing by using the **ipv6 ospf area** command.

Configuración ID router OSPFV3

Assigning a Router ID to R1

```
R1(config)# ipv6 router ospf 10
R1(config-rtr)#
*Mar 29 11:21:53.739: %OSPFv3-4-NORTRID: Process OSPFv3-1-
IPv6 could not pick a router-id, please configure manually
R1(config-rtr)#
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)#
R1(config-rtr)# auto-cost reference-bandwidth 1000
% OSPFv3-1-IPv6: Reference bandwidth is changed. Please
ensure reference bandwidth is consistent across all routers.
R1(config-rtr)#
R1(config-rtr)# end
R1#
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 10"
  Router ID 1.1.1.1
  Number of areas: 0 normal, 0 stub, 0 nssa
  Redistribution:
    None
R1#
```


Recordar: Link local por interface

Pueden ser distintas o iguales en un mismo router

Habilitanto OSPFV3 en interfaces

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/1
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# end
R1#
R1# show ipv6 ospf interfaces brief
Interface   PID   Area      Intf ID  Cost    State  Nbrs F/C
Se0/0/1     10      0         7      15625   P2P    0/0
Se0/0/0     10      0         6      647     P2P    0/0
Gi0/0       10      0         3       1      WAIT   0/0
R1#
```

R(Config)#interface loopback 1

R(Config-if)#ip address 10.40.10.1 255.255.255.0

Si hemos creado por ejemplo 4 loopbacks en un mismo router, podemos publicarlas con un solo comando #network

10.10.10.0

10.10.20.0

10.10.30.0

10.10.40.0

#network 10.0.0.0 0.255.255.255 area 0

Passive interface

R(Config-router)#passive-interface default

R(Config-router)#no passive-interface fa 0/0

Comando show ip ospf database

El estado entre un DRother y un DR o BDR será full-state.

Entre los DRother será two-way

En broadcast: #ip ospf priority 0 : al dar valor 0 estamos diciendo que este router nunca va a ser DR o BDR en este broadcast.

Capitulo 9: ACLs

Ojo, una interface sólo puede tener como mucho 6 ACL: 2 x ip, 2 x ipx y 2 x Apple talk (una de entrada y otra de salida)

LISTAS DE CONTROL DE ACCESO: ACL

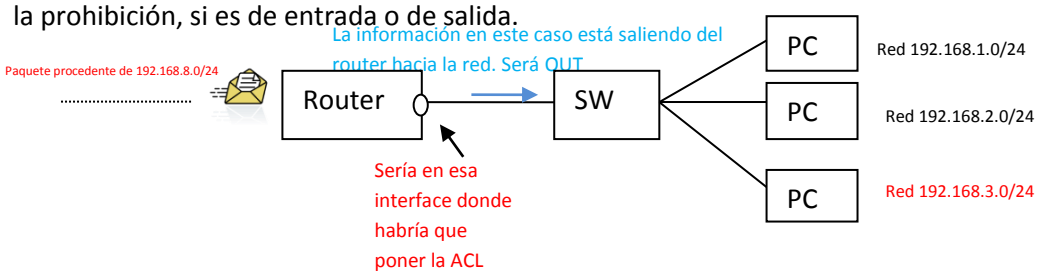
ACL: conjunto de instrucciones que deniega o permite tráfico (paquetes ip). Son una especie de mini-firewall.

Hay distintos tipos de ACL:

- Estándar: numeradas
  - Nombradas
- Extendidas numeradas
  - Nombradas

Estándar: Puede ser numerada o nombrada. Es estándar numerada cuando va del 1 al 99 y del 1300 al 1999

Para denegar tráfico de la red 192168.8.0/24 (la de origen) a la 192.168.3.0/24 (la de destino), lo recomendable es poner la ACL lo más cerca posible del destino de los datos: La interface que lleva a la red que queremos taponar. Además habrá que decirle a esta interface el sentido de la prohibición, si es de entrada o de salida.



Configuración ACL estándar:

1- En el router de destino:
R(Config)#ACCESS-LIST 1 (valor del 1 al 99 o del 1300 al 1999) [permit/deny]+dirección ip de origen+máscara wildcard de origen
=#access-list 1 deny 192.168.8.0 0.0.0.255

Información para el **Router** afectado
ACCESS-LIST

2- Para decir la interface:
En el router de destino R(Config)#interface fa 0/0 (esto es solo un ejemplo de interface)
R(Config-if)#IP ACCESS-GROUP 1 + añadir el sentido del tráfico (OUT/IN). En este caso, como la información esta saliendo del router hacia la red LAN, será out.
Al ponerlo así, denegamos todo, no solo los de la red 192.168.8.0, por que al poner el comando deny, por defecto se incluye un deny any. Por eso antes habría que ponerle el comando desde config: Access-list 1 permit any.

Información para la **interface** del Router afectado
ACCESS-GROUP

Si nos equivocáramos, habría que eliminar la lista entera. No bastaría con poner No.

R(Config)#no Access-list 1
R(Config-if)#no ip access-group 1 out.
Ojo! Usar el mismo valor decimal tb que Access-list al eliminar Access-group.

En una lista ACL hay que permitir o denegar todo al principio. Si quisiéramos añadir alguna instrucción más, habrá que cargarse todo y ponerlo de nuevo con la orden nueva. Por eso para poner una orden nueva, para no tener que repetir todo lo que hubiéramos hecho, es recomendable hacer un copy/paste en un block de notas para aprovecharlo después.

Para denegar lo que viniera de 1 equipo solo y no todo lo de la red a la que perteneciera ese equipo (permitírselo a todos los demás equipos de esa red) sería:

Ej: en la red 192.168.6.0 se quiere prohibir el 6.2 y permitir todos los demás de esa red y de las demás redes.

1 - R de destino(Config)#access-list 2 deny host 192.168.6.2 (aquí no se le pone máscara por que al poner HOST la máscara es 0.0.0.0)

2 - R de destino(Config)#access-list 2 permit 192.168.6.0 0.0.0.255

Red de origen con su máscara wildcard

Información para el **Router** afectado

3 - R de destino(Config)#access-list 2 permit any (se pone este permit any, por que por defecto termina poniendo deny any)

4 - R de destino(Config)#interface fa 2/0 (si fuera esa interface)

5 - R de destino(Config-if)#ip Access-group 2 + el sentido (out/in). En este caso sería out.

Información para la **interface** del Router afectado

RESUMEN:

1- Donde ubicar la ACL: lo más cerca del destino
2- En qué interface y en que sentido: asociar a la interface física más cercana al destino.
3- El sentido siempre se observa desde el punto de vista del router.

Para denegar una red:

R(Config)#access-list <1-99> deny + dirección de origen con su mascara

Así ir añadiendo todas las prohibiciones que queramos

Al final poner "permit any" (por defecto termina poniendo deny any)

Después ir a la interface afectada: R(Config)#interface fa 0/0

R(Config-if)#IP ACCESS-GROUP 1. Este uno responde al número de la lista de acceso(Access-list) a la que le queremos vincular, ya que en un mismo router, puede haber muchas ACLs que cada una de ellas le habremos asignado un valor decimal diferente con Access-list 1. Al final se le añade el sentido OUT/IN.

Para denegar un equipo en especial asociado evidentemente a una ip:

1º R(Config)#access-list n deny HOST 192.168.8.9 0.0.0.0 Así estaremos denegando implícitamente todo lo que venga de ese pc.

2º R(Config)#access-list n permit 192.168.8.0 para que deje pasar todo lo demás de esa red + su máscara wildcard. Si solo se denegara este equipo, no haría falta poner que permitiera esta red. Con el permit any esta red estaría incluida.

3º R(Config)#access-list n permit any para que permita todo lo demás.

4º Nos vamos a la interface correspondiente con R(Config)#interface fa 0/0

5º R(Config-if)#ip Access-group n + el sentido (OUT/IN)

Para denegar el acceso a todos los equipos que tengan la misma ip, por ej la 56 de distintas redes, por ej las redes 3,7 y 9 pondríamos.

1º R(Config)#access-list n deny + 192.168.3.56 (aqui no pondríamos mascara por que se supone la 0.0.0.0)

#access-list n deny + 192.168.7.56

#access-list n deny + 192.168.7.56

2º R(Config)#access-list permit any

3º Nos vamos a la interface correspondiente: R(Config)#interface fa 0/0

R(Config-if)#ip access-group n (out/in)

Con el comando SHOW ACCESS-LIST vemos todas las ACLs de ese router.

Ejemplo de configuración de ACLs estándar:

```
Router(config)#ip access-list [standard | extended ] name
```

Alphanumeric name string must be unique and cannot begin with a number.

```
Router(config-std-nacl)#[permit | deny | remark] {source
[source- wildcard]} [log]
```

```
Router(config-if)#ip access-group name [in | out]
```

```
R1(config)#access-list 1 remark Do not allow Guest workstation
through
R1(config)#access-list 1 deny host 192.168.10.10
R1(config)#access-list 1 remark Allow devices from all other
192.168.x.x subnets
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)#interface s0/0/0
R1(config-if)#ip access-group 1 out
R1(config-if)#
```

Example 2: Commenting a named ACL

```
R1(config)#ip access-list standard NO_ACCESS
R1(config-std-nacl)#remark Do not allow access from Lab
workstation
R1(config-std-nacl)#deny host 192.168.11.10
R1(config-std-nacl)#remark Allow access from all other networks
R1(config-std-nacl)#permit any
R1(config-std-nacl)#interface G0/0
R1(config-if)#ip access-group NO_ACCESS out
R1(config-if)#
```

LISTAS DE ACCESO EXTENDIDAS

Una ACL extendida son instrucciones también para filtrar tráfico. En una ACL estándar se prohíbe todo lo que venga o bien de una red o bien de un pc sin discriminar nada más. Una ACL extendida es más específica. Por ejemplo, se podría prohibir el tráfico de paquetes smtp y permitir los ftp. Es por ello que se puede filtrar tipos de tráfico, con lo que se concreta mucho más la filtración.

Una ACL extendida también puede ser numerada y su rango es <100-199> y <2000-2699>

Configuración de ACL extendida:

Comando Access-list <100-199> al poner este valor, la IOS ya sabe que va a ser una lista extendida, y los parámetros que espera.

Protocolo Protocolos para capa de aplicación (7)	**Nº de Puerto**	**TCP/UDP** Protocolo de transporte (4)
HTTP	80	TCP
FTP	21	TCP
TELNET	23	TCP
SMTP	25	TCP
DNS	53	TCP/UDP
TFTP	69	UDP
SNMP	161	UDP
RIP	520	UDP

Cada protocolo de aplicación va asociado a un protocolo de transporte. Recordamos que UDP es menos confiable pero menos pesado. DNS puede ser TCP o UDP.

Un número de puerto es un valor decimal con el que yo identifico un protocolo de la capa de aplicación (7). Por cada tarea, el equipo le otorga un número de puerto aleatorio: Ej: visitar la página del mundo y mientras tanto visitar también la página del país.

Ej: desde mi pc con dirección 192.168.1.1 voy a la 88.16.20.1 que corresponde al mundo y a la vez voy a 88.14.20.1 que corresponde al país.

Al visitar el mundo mi pc otorga aleatoriamente el puerto 45714.

Si el puerto de destino del mundo es el 80, la orden sería:

Dirección de mi pc	Puerto aleatorio de mi pc	Dirección ip del mundo	Puerto destino del mundo
192.168.1.1	45714	88.16.20.1	80

Dentro de los números de puerto hay de varias clases:

- Bien conocidos: ej: puerto 80(de destino) No se generan aleatoriamente: Rango<0-1023>
- Registrados: Rango<1024-49151>
- Aleatorios: Rango<49152-65535>

Un ejemplo gráfico de esto seria: la dirección ip es el nombre de la calle y el nº de puerto el nº y el piso.

En origen el puerto es aleatorio pero en destino NO. Para ver puertos conocidos, consultar google.

Ejemplo completo ACL estándar:

Access-list 10 permit 192.168.30.0 0.0.0.255

Ejemplo complete ACL extendida:

Access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80

Aquí el 103 nos dice ya que es una ACL extendida. Permit porque estamos permitiendo. TCP por que lo que vamos a permitir es un protocolo http que corresponde al puerto 80 y es TCP. 192.168.30.0 es la dirección de origen con su máscara wildcard. Any para que lo acepte todo. Eq implica que sea igual al puerto que pone a continuación. 80 nº de puerto afectado.

Los parámetros para los puertos de destino son:

- Eq = Igual. En este caso el puerto de destino será el 80.
- lt= less tan (menos que)
- gt= grather tan (mayor que)

Otro ejemplo:

Access-list permit/deny [tcp/udp/ip] + la dirección ip de origen:192.168.2.0 + máscara wildcard de origen:0.0.0.255 any + dirección ip de destino + [eq/lt/gt] + nº de puerto. En este caso usamos uno "bien conocido" como es el 23.

Si ponemos ip engloba udp y tcp. En este caso usaremos tcp porque es el puerto 23 y si consultamos la tabla vemos que es un protocolo telnet que es el asociado al puerto 23 y que es tcp.

1- Access-list 114 permit tcp 192.168.2.0 0.0.0.255 any eq 23a
   Ojo! Aquí también se deniega todo lo demás por defecto, con un comando oculto que sería a Deny any any
2- Access-list 114 permit tcp 192.168.2.0 0.0.0.255 any eq 21
3- Access-list 114 permit tcp 192.168.2.0 0.0.0.255 any eq 20

Estas tres instrucciones que estamos dando en nuestra ACL extendida implican que deneguemos todo pero que deje pasar telnet (23), ftp (21) y ftp-data (20)

Se puede poner el nº de puerto, (como hemos hecho en esos tres casos, o también se podría poner el nombre del protocolo: ….any eq telnet, …..any eq ftp …..any eq ftp-data.

Si lo pusiéramos al revés para permitir todo, acabaríamos poniendo permit ip any any.

Lo que se filtra sobre todo en este tipo de ACLs son

1 - Direcciones ip de origen y destino.

2 - Puertos tcp/udp de origen y destino

3 – Tipo de protocolo ip, icmp, udp o nº de protocolo.

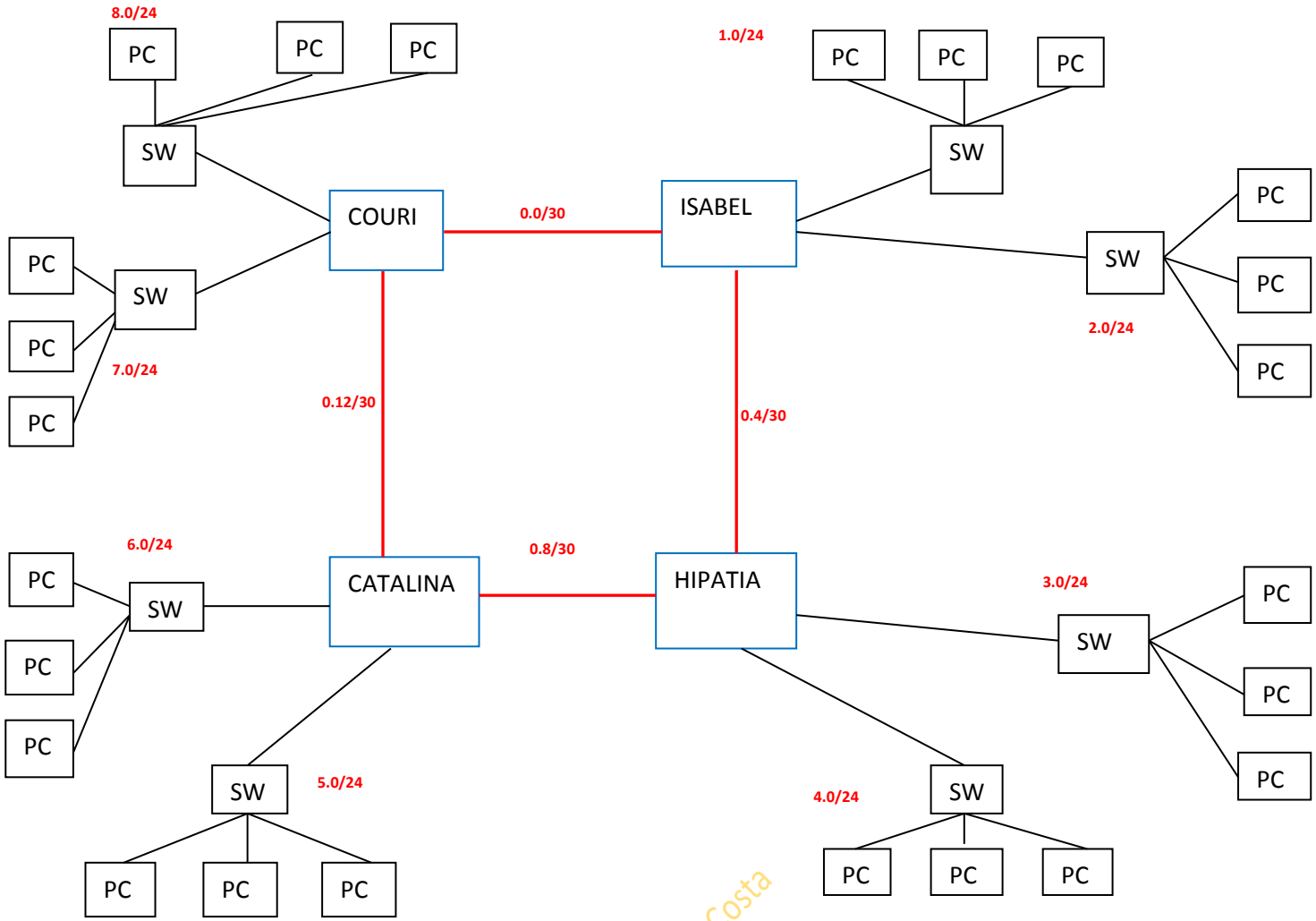
Después de haber creado la ACL, hay que decírselo a la interface.

R(Config)#interface fa 0/0

R(Config-if)#ip Access-group 2

**LAS ACLs EXTENDIDAS SUELEN COLOCARSE LO MAS CERCA POSIBLE DEL ORIGEN DE LOS DATOS* (al revés que en las estándar) y en sentido de entrada (IN).

LAS ACLs SON INDEPENDIENTES DEL PROTOCOLO DE ENRUTAMIENTO. SE PUEDEN IMPLEMENTAR CON RIP, RIP2, EIGRP Y OSPF.



En la red anterior se pide:

1º Se denegará el tráfico smtp, ftp y telnet con origen el equipo 192.168.3.2 y destino los equipos impares de las redes pares. Así mismo se denegará el tráfico tftp con origen la red 192.168.3.0 y destino las redes del router catalina. Se permitirá todo lo demás.

Será: 1º Nos vamos a hipatia porque la red 3.2 está en ese router.

Hipatia(Config)#access-list 101(asi le decimos que es ACL extendida)+ deny tcp

2º ponemos deny porque estamos denegando y tcp porque tanto smtp, tft y telnet son protocolos de transporte tcp.

3º ponemos host porque nos estamos refiriendo a un equipo en particular, el 3.2

Hipatia(Config)#access-list 101 deny tcp host 192.168.3.2

4º Seguimos poniendo los destinos, que son los equipos impares de las redes pares. Como está haciendo referencia otra vez a equipos y no a redes, sabemos que habrá que usar otra vez el comando host. Así iremos poniendo los equipos que pide red a red:

1er equipo impar de la red par

Hipatia(Config)#access-list 101 deny tcp host 192.168.3.2 host 192.168.2.3

1ª red par

5º Finalizamos poniendo el comando eq 25 porque queremos empezar denegando smtp que corresponde al puerto 25. Después habrá que ir poniendo eq con el valor de los demás protocolos: ftp=21 y telnet=23

Hipatia(Config)#access-list 101 deny tcp host 192.168.3.2 (origen) host 192.168.2.3 (destino) eq 25 (Valor del protocolo de transporte)

Así sucesivamente contemplando todos los equipos impares de las redes pares así como los valores eq según el valor del protocolo que queremos denegar.

Para denegar el tráfico tftp con origen la red 192.168.3.0 y destino las redes del router catalina, que son la 5.0 y la 6.0, nos volvemos a ir al router Hipatia, porque es el origen de la red 3.0:

Hipatia(Config)#access-**list** destino con su máscara wildcard= 192.168.5.0 0.0.0.255 (la 5.0 es la primera red que denegamos. Después habrá que hacer lo propio con la 6.0 que es la otra red de catalina) + eq 69 porque ese es el valor del protocolo de transporte de tftp.

Así la 1ª y 2ª orden total será:

Hipatia(Config)#access-list 101 deny udp 192.168.3.0 0.0.0.255 192.168.5.0 (Red 5 de catalina) 0.0.0.255 eq 69

Hipatia(Config)#access-list 101 deny udp 192.168.3.0 0.0.0.255 192.168.6.0 (Red 6 de catalina) 0.0.0.255 eq 69

Terminaremos poniendo permit ip any any y nos vamos a darle la asociación a la interface correspondiente.

Hipatia(Config)#interface fa 0/0 Hipatia(Config-if)#ip access-group 101 IN

El 101 hay que ponerlo porque es el mismo valor decimal que le dimos a Access-list e IN es el sentido de la prohibición.

2º se denegará todos los protocolos tcp con origen la red 1.0/24 y destino los equipos acabados en 3 y 4 de las redes de los routers Courie e Hipatia. Se permitirá todo lo demás.

Nos vamos al router Isabel porque es el origen de la red 1.0/24.

Isabel(Config)#access-list 101 deny tcp + red de origen 192.168.1.0 con su máscara 0.0.0.255 + host porque se está refiriendo a equipos, los acabados en 3 y 4 de las redes 7.0 y 8.0 (los de Couri) y las redes 3.0 y 4.0 (las de Hipatia).

Usaremos también el comando eq = 80(http), 21(ftp), 23(telnet) y 25(smtp) que corresponden a los valores de todos los protocolos tcp que son los que nos pide denegar.

Así quedará:

Isabel(Config)#access-list 101 deny tcp 192.168.1.0 0.0.0.255 host 192.168.7.3 eq 80

Isabel(Config)#access-list 101 deny tcp 192.168.1.0 0.0.0.255 host 192.168.8.3 eq 80

Isabel(Config)#access-list 101 deny tcp 192.168.1.0 0.0.0.255 host 192.168.7.4 eq 80

Isabel(Config)#access-list 101 deny tcp 192.168.1.0 0.0.0.255 host 192.168.8.4 eq 80

Isabel(Config)#access-list 101 deny tcp 192.168.1.0 0.0.0.255 host 192.168.3.4 eq 80

Isabel(Config)#access-list 101 deny tcp 192.168.1.0 0.0.0.255 host 192.168.4.4 eq 80…….

….Así sucesivamente hasta completar todas las redes y todos los valores eq.

Terminaremos poniendo permit ip any any y yéndonos a la interface correspondiente para hacer la asociación y el sentido: ip Access-group 101 IN

3º Se permitirá todos los protocolos udp con origen en el equipo 192.168.8.2 y destino los equipos pares de las redes impares. Se denegará todo lo demás:

Los protocolos udp son tftp(69) snmp(161) y rip(520)

Nos vamos al router Curie porque es el origen de la red 8.2

Couri(Config)#access-list 101 permit udp host 192.168.8.2 (no ponemos mascara al ser equipo y ponemos host precisamente por lo mismo) + los distintos destinos

Couri(Config)#access-list 101 permit udp host 192.168.8.2 host (aquí también se está refiriendo a equipos en concreto, por eso ponemos host tb) 192.168.1.2/1.4/3.2/3.4/5.2/5.4/7.2/7.4 eq 69/161/520.

Iremos poniendo sucesivamente todos los equipos pares de las redes impares tal como nos pide, así como iremos cambiando el valor eq para poner el correspondiente.

Como se deniega todo lo demás y por defecto pone deny any any no habrá que darle ninguna orden más.

Pasaremos directamente a la interface a darle la asociación y el sentido.

Las estándar lo más cerca del destino.

Las extendidas lo más cerca del origen.

ACLs NOMBRADAS

En las acl nombradas podemos intercalar instrucciones sin tener que hacerlo todo de nuevo como con las extendidas. Ana acl nombrada puede ser también estándar o extendida.

Características: se le asigna un nombre. Este nombre puede tener números también. Se sugiere que sea siempre en mayúscula. No hay acentos ni espacios y siempre ha de empezar por una letra. Se puede agregar o borrar entradas a la lista sin tener que cargárselo todo.

Con el comando remark podemos añadir comentarios. Si ponemos no remark, lo quitaremos: R(Config)#access-list remark ------

Ip Access-list [standar/extended] name

Std si es estandar

R(Config-std-nacl)# permit/deny remark + origen + mascara wildcard 8se puede finalizar también poniendo log para que lo guarde.

Después lo asociamos con la interface: R(Config-if)#ip Access-group name in/out.

Para verificar la acl: SHOW ACCESS-LIST NAME. Si no ponemos el nombre, saldrán todas.

Las listas nombradas permiten meter más instrucciones. Haciendo un show Access-list name al ver las entradas, el sistema por defecto les pone un valor a cada una de las entradas:

10 permit ……

20 deny……

30 deny……

} esto son solo ejemplos.

Para meter más instrucciones: R(Config)#ip Access-list standar name

R(Config-std-nacl)#15 permit/deny……

Como en cada entrada por defecto le ha asignado el sistema un valor (en este caso 10, 20 y 30) al poner nosotros el valor 15 le estaríamos diciendo que metiera esa instrucción en esa posición. Podíamos haber puesto 11 ó 12, 13, 14. Viendo el valor que tiene en el show acces-list name una instrucción, para añadir otra instrucción le asignaremos el valor que necesite para colocarla en la posición que necesitemos.

Para cargarnos una instrucción, poner NO (valor numérico que tenga esa instrucción)ej 10 permit/deny…..desde dentro de la acl: R(Config-std-nacl)#no 10 permit/deny…..

Si al final hay que permitir, acabar con PERMIT IP ANY ANY

Para una acl nombrada extendida:

R(Config)#access-list extended name

Ext al ser estándar

R(Config-ext-nacl)#permit……

Para denegar una lista entera:

NO IP ACCESS-LIST STANDAR/EXTENDED + NOMBRE

Después asociarlo con la interface: ip Access-group name in/out.

Comando established: con este comando se permiten los protocolos hechos desde dentro. Esto es, permite la vuelta de paquetes del protocolo que sea a su vuelta a pesar de que ese protocolo lo hayamos denegado en concreto. Se seguirá denegando todo lo que venga de ese protocolo de fuera. Si por ejemplo hemos prohibido la entrada de paquetes ping a un router,

los pines no entraran en nuestro ruter, salvo si nosotros hemos hecho un ping fuera de nuestra red, con established dejará pasar ese ping de vuelta para la respuesta.

Ejercicio 15.

1 - Conf.básica.

2 - Dividir una red clase B en 12 subredes.

3 - Conf. EIGRP

4 – Verificación

5 – Comprobar conectividad punto a punto

6.1 Se denegará el tráfico HTTP, TFTP, FTP y SMNP con origen el penúltimo equipo de la red H y destino los equipos acabados en 4 de las redes Sonta, Maserati y Ferrari. Se permite todo lo demás.

6.2 Se denegará todo el tráfico proveniente de las redes de los routers Bugatti, Aston y Ferrari y destino la red D. Se permite todo lo demás.

6.3 Se permitirá todo el trafico UDP con origen el 1er equipo de la red E y destino los equipos acabados en 8 de las routers Bugatti, Aston y Ferrari. Se denegará el resto.

6.1/ Será extendida porque singulariza protocolos. Por tanto poner ACL en origen.

El penúltimo equipo de la red H es del router Bugatti y es el 172.16.159.253 (comprobado al hacer el subneting)

Los comandos serán: 1º Bugatti(Config)#ip Access-list extended PAKITO (así la hemos llamado)

2º Bugatti(Config-ext-nacl)#deny tcp host 172.16.159.253 host 172.16.64.4 eq 80.

Los equipos acabados en 4 de los routers que pide son: 172.16.64.4/80.4/96.4/112.4/0.4/16.4

El tcp y el eq 80 irán cambiando según el protocolo hasta completar todos.

Terminamos poniendo permit ip any any, y lo asociamos con la interface con IP ACCESS-GROUP PAKITO + EL SENTIDO OUT.

6.2/ Al pedirnos que deneguemos todo el tráfico será una lista estándar. Por lo tanto en destino. La red de destino está en el router Sonta y es la 172.16.80.0.

Así que nos vamos a Sonta y denegamos todas las redes de los routers que nos piden:

IP ACCESS-LIST STANDAR CHOCOLATERO

Sonta(Config-std-nacl)#deny 172.16.144.0 0.0.15.255 y así sucesivamente con todas las redes que nos pide.

Terminamos poniendo ip permit any y asociándolo a su interface.

6.3/ Será extendida al hacer referencia a protocolos, por lo tanto en origen. El origen es 172.16.96.2. Será un host al hacer referencia a un solo equipo y se irán permitiendo los equipos (usando otra vez el host por cada equipo) que pide y cambiando el tcp/udp dependiendo del protocolo así como su valor eq.

Capítulo 10: DHCP

CONFIGURACION SERVIDOR DHCP EN ROUTERS CISCO (levantar el servicio DHCP)

*tb se podría hacer desde un servidor dedicado

Lo primero que hay que saber es que habrá que levantar tantos DHCP como redes LAN tenga el router. Para distinguir cada red, le daremos un nombre (identificador). En este caso llamaremos a nuestra red, “red1”. El comando que se usa es: IP DHCP POOL + IDENTIFICADOR+intro (estando en config)

R1(Config)# IP DHCP red1+intro

El prompt quedará entonces así; R1(DHCP-Config)#. Ahora hay que decirle a la red1 qué direcciones va a servir y se le dice poniendo el comando NETWORK+ La red que se va a asignar con su máscara correspondiente. De tal forma que quedaría así: si por ejemplo vamos a levantar servicio dhcp para la red 192.168.1.0:

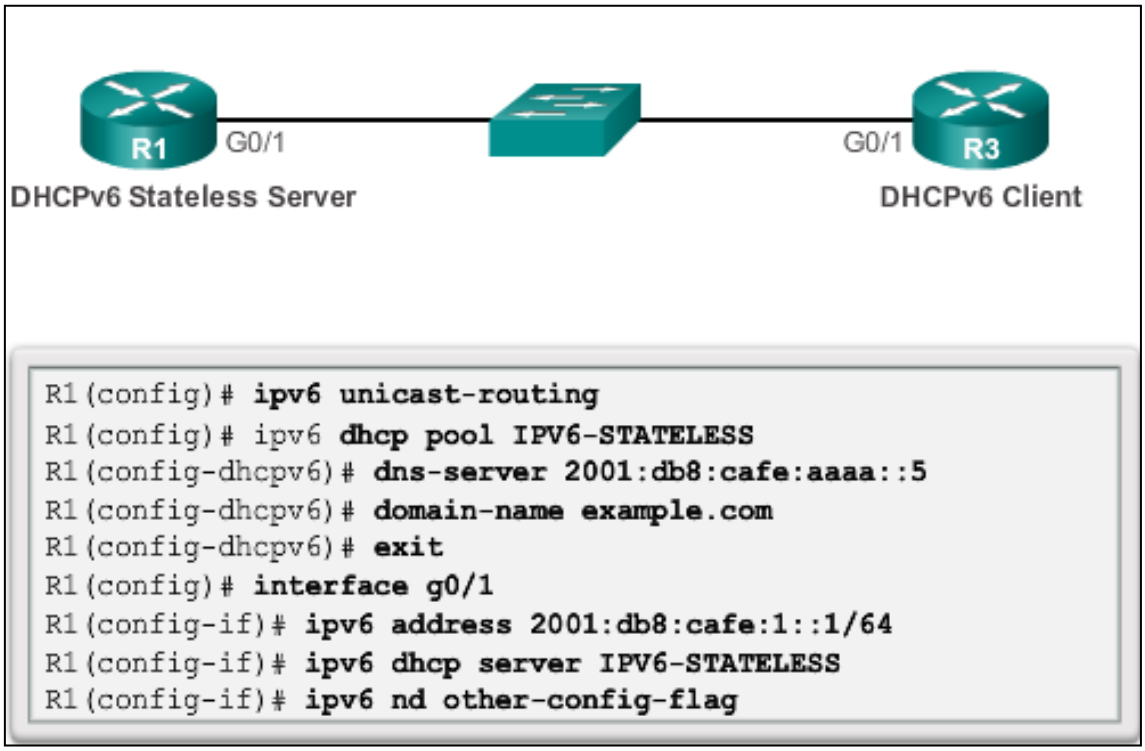
R1(DHCP-Config)#network 192.168.1.0 255.255.255.0

Ojo!! Hay que decirle que la IP de la puerta de enlace no la tome. Para ello, el comando es: DEFAULT-ROUTER+ la dirección de la puerta de enlace. Si la puerta de enlace es la 1.1, la secuencia completa seria: R1(DHCP-Config)#default-router 192.168.1.1 (aquí no hace falta poner la máscara)

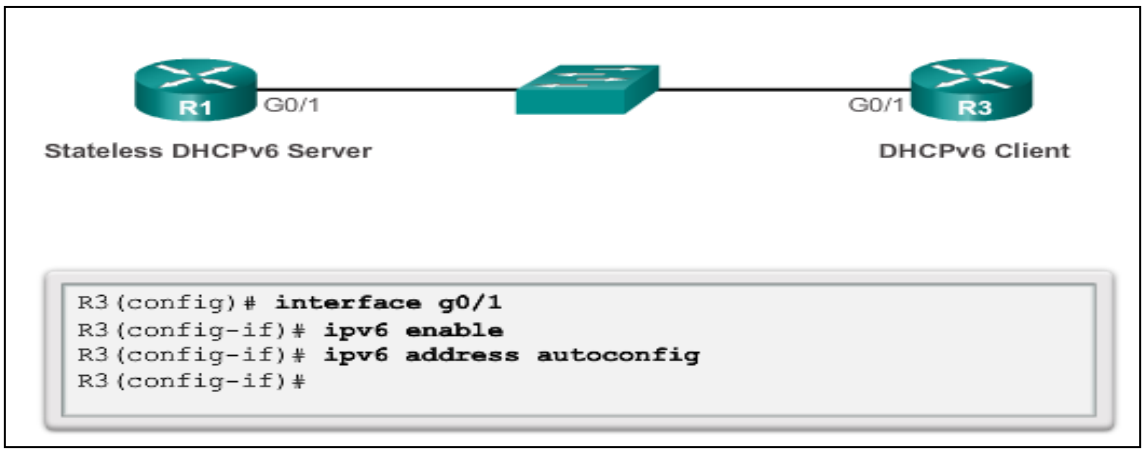
Normalmente hay un rango de IPs que se reservan para impresoras u otros dispositivos. Para excluir estas IPs, nos volvemos a Config R1(Config)# y ponemos el comando: IP DHCP EXCLUDED-ADDRESS y ponemos la primera y última dirección que vamos a excluir. Si quisiéramos excluir desde la 2 hasta la 19 seria así: R1(Config)#ip excluded-address 192.168.1.2 192.168.1.19

```
R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
R1(dhcp-config)# default-router 192.168.10.1
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com
R1(dhcp-config)# end
R1#
```

DHCP en IPV6 en stateless

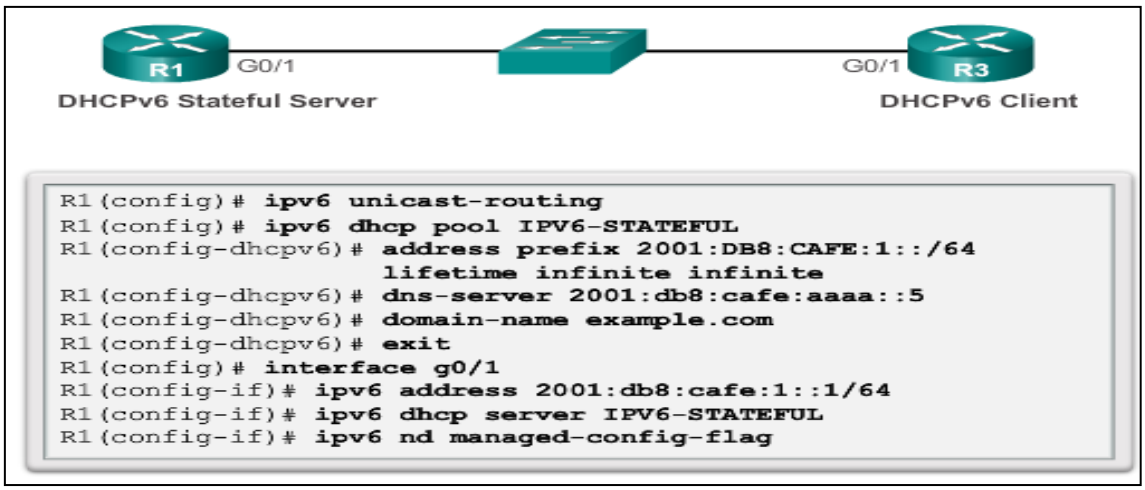



```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 dhcp pool IPV6-STATELESS
R1(config-dhcpv6)# dns-server 2001:db8:cafe:aaaa::5
R1(config-dhcpv6)# domain-name example.com
R1(config-dhcpv6)# exit
R1(config)# interface g0/1
R1(config-if)# ipv6 address 2001:db8:cafe:1::1/64
R1(config-if)# ipv6 dhcp server IPV6-STATELESS
R1(config-if)# ipv6 nd other-config-flag
```

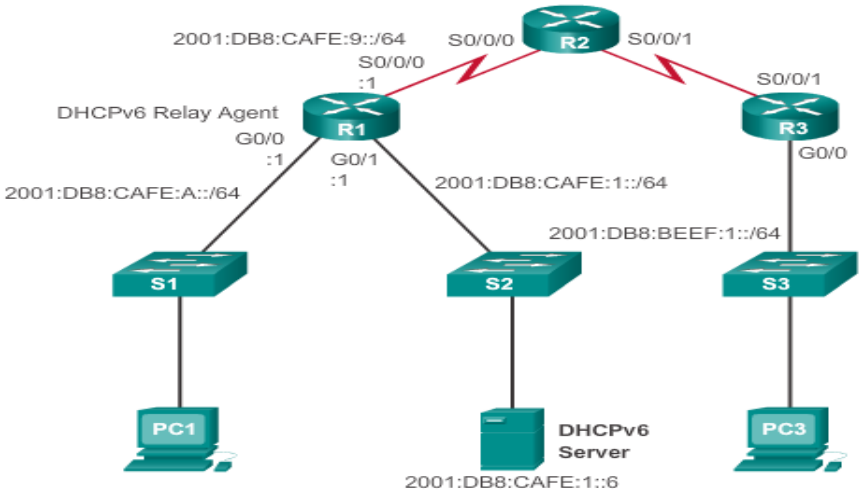



```
R3(config)# interface g0/1
R3(config-if)# ipv6 enable
R3(config-if)# ipv6 address autoconfig
R3(config-if)#
```

DHCP en IPV6 en statefull

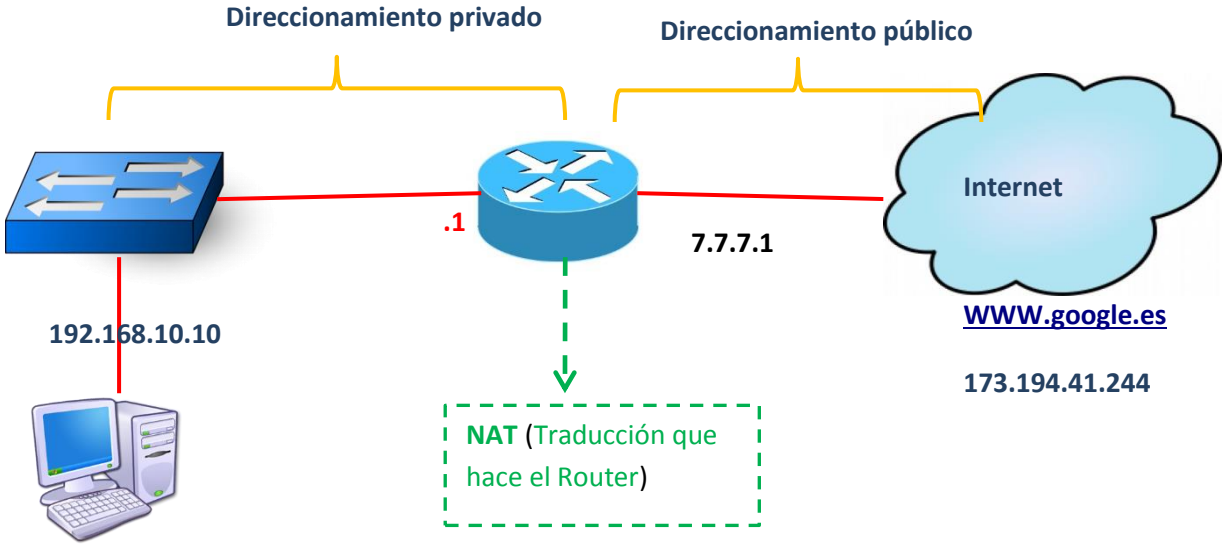



```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 dhcp pool IPV6-STATEFUL
R1(config-dhcpv6)# address prefix 2001:DB8:CAFE:1::/64
                    lifetime infinite infinite
R1(config-dhcpv6)# dns-server 2001:db8:cafe:aaaa::5
R1(config-dhcpv6)# domain-name example.com
R1(config-dhcpv6)# exit
R1(config)# interface g0/1
R1(config-if)# ipv6 address 2001:db8:cafe:1::1/64
R1(config-if)# ipv6 dhcp server IPV6-STATEFUL
R1(config-if)# ipv6 nd managed-config-flag
```




```
R1(config)# interface g0/0
R1(config-if)# ipv6 dhcp relay destination 2001:db8:cafe:1::6
R1(config-if)# end
R1# show ipv6 dhcp interface g0/0
GigabitEthernet0/0 is in relay mode
  Relay destinations:
    2001:DB8:CAFE:1::6
R1#
```

Capítulo 11: NAT: NETWORK ADDRESS TRASLATION
Traducción de direccionamiento de red (en IPV4)



NAT traduce el direccionamiento privado al público. Se implementa en los Routers directamente conectados a internet. También se puede usar en un Router que no sea perimetral (el que da acceso directo a internet), pero no suele ser lo habitual.

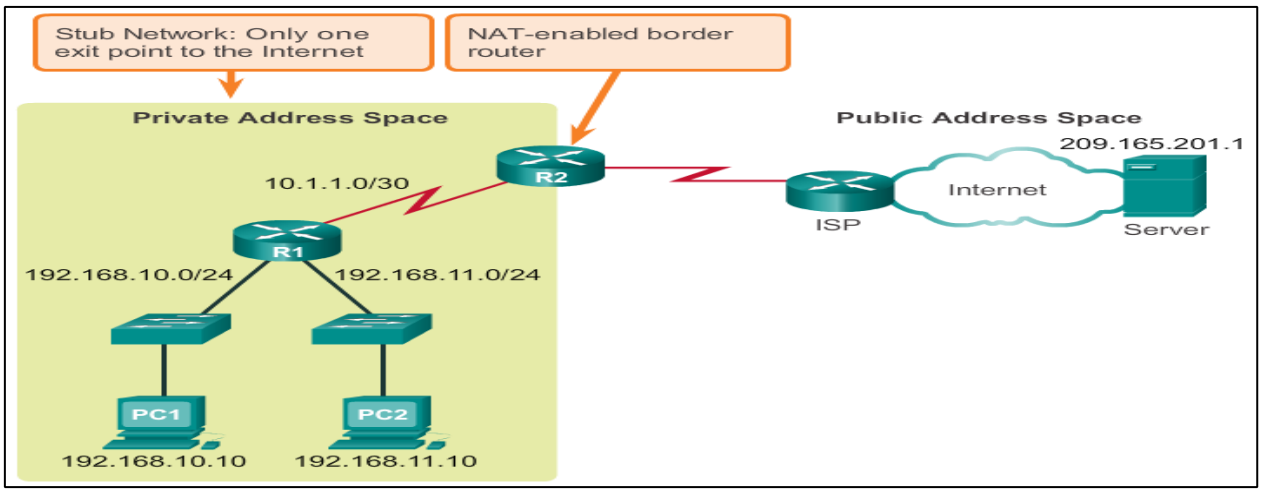
Las direcciones…

1- 10.0.0.0 hasta la 10.255.255.255
2- 172.16.0.0 hasta la 172.31.0.0
3- 192.168.0.0 hasta la 192.168.255.255

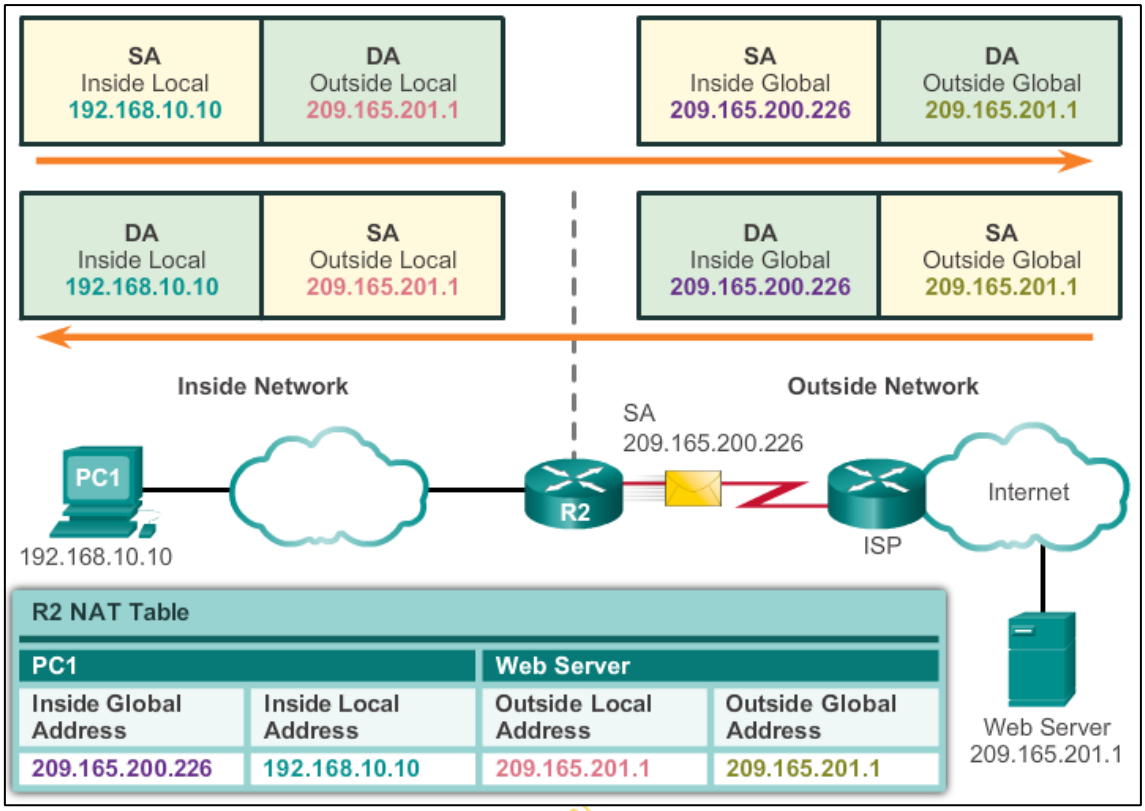
…son direcciones privadas.

Private Internet addresses are defined in RFC 1918:		
Class	RFC 1918 Internal Address Range	CIDR Prefix
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16

Un router NAT modificara la dirección de origen convirtiéndola de privada a pública. Por eso un router dentro de una misma red privada no necesitará NAT, pero sí el router de borde o perimetral que es el que sale a internet.



NAT tiene 4 tipos de direcciones:

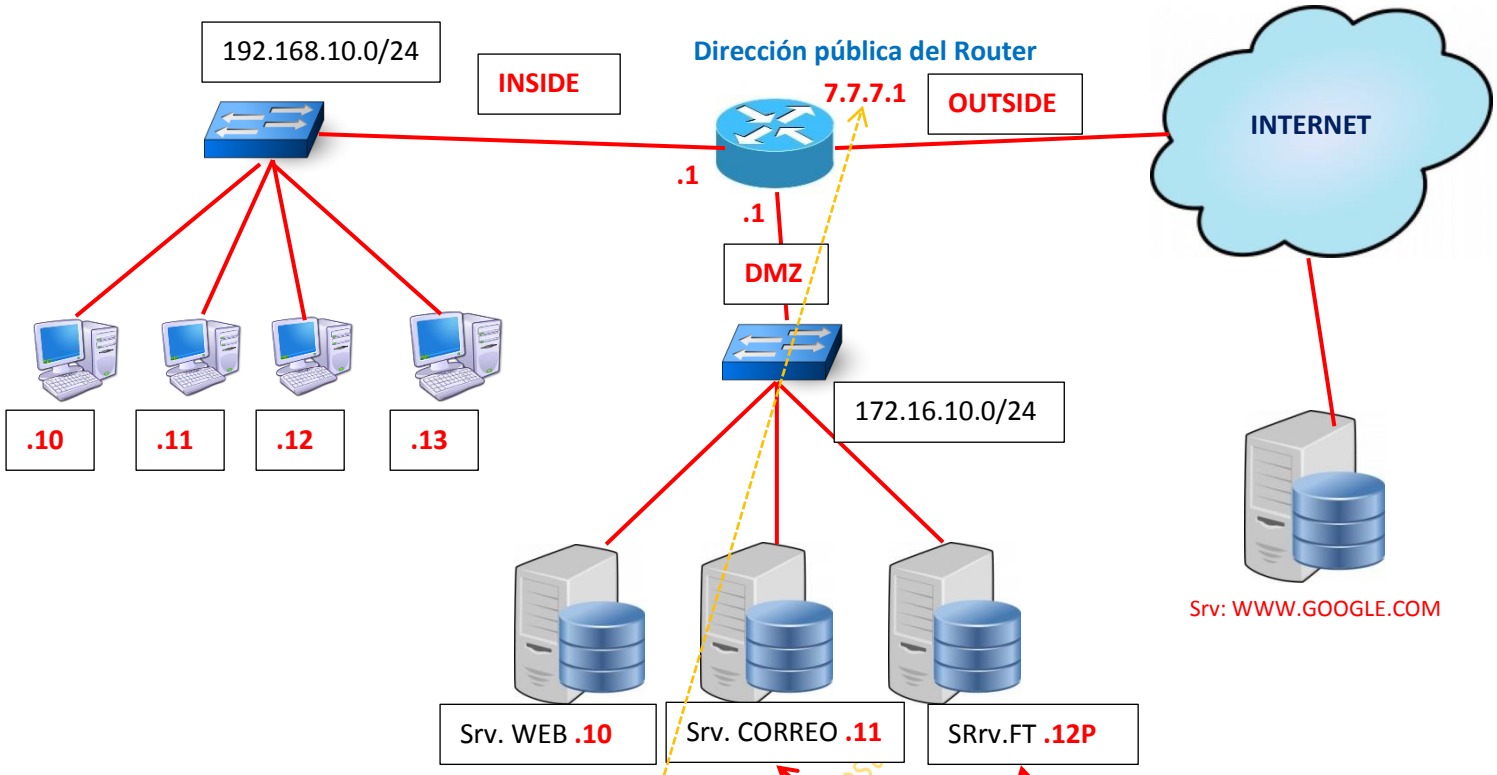


Como vemos en la diapositiva las direcciones son:

1- Dirección Inside Global
2- Dirección Inside Local
3- Dirección Outside Local
4- Dirección Outside Global

} Generalmente es la misma

NAT ESTATICO:



NAT ESTATICO: Se usan en las DMZs para que siempre salga con la misma IP pública (se usa en las grandes compañías)

Ejemplo de Tabla NAT:

Las direcciones 172.16.10.10 = 3.3.3.10

172.16.10.11 = 3.3.3.11

172.16.10.12 = 3.3.3.12

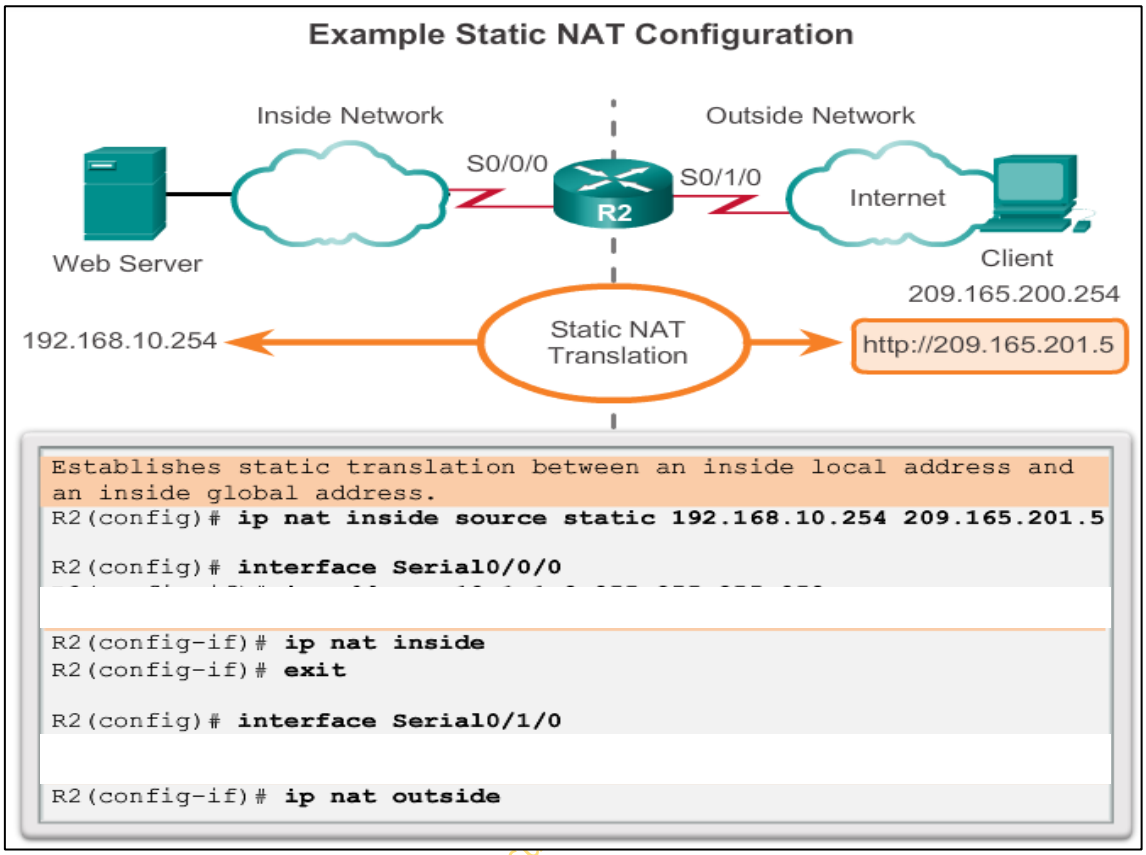
Estas tres direcciones privadas de los servidores, 172.16.10.10/11/12 saldrán a internet SIEMPRE con la misma IP pública asignada correspondiente, que en este caso como vemos son la 3.3.3.10/11/12

Las direcciones IP públicas se solicitan a las compañías de servicios de internet ISP, en este caso, para los servidores, que como vemos en este caso, son distintas a la dirección pública del router que es la 7.7.7.1. Por eso en el router hay que configurar las direcciones IP global y local. Así cuando se solicite la dirección 3.3.3.10 al llegar la petición desde internet al router, éste la redireccionará a la 172.16.10.10 que es a la dirección que está vinculado. El router al tener NAT lo transforma porque lo tiene configurado en su tabla NAT.

Este proceso también sucede al revés. Cuando el servidor responde desde su dirección 172.16.10.10, al salir a internet saldrá como la dirección 3.3.3.10. Todo esto NAT lo hace estáticamente.

Al fin y al cabo, el trabajo de NAT es pasar/traducir/transformar red local a una global y viceversa.

Configuración de NAT Estático:



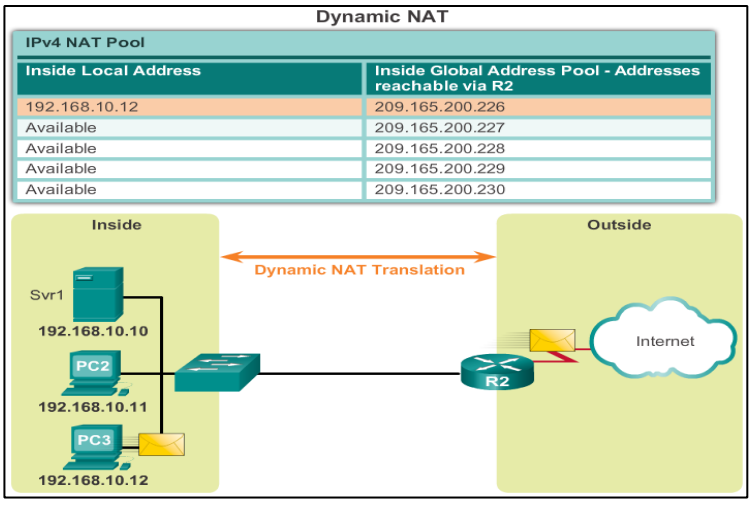
1: R(Config)#ip nat inside source static + ip local privada + ip pública

2: Nos metemos en la interface de entrada y le decimos que va a ser la interface de entrada:

R(Config-if)#ip nat inside

3: Nos vamos a la interface de salida y le decimos que va a ser de salida

R(Config-if)#ip nat outside



NAT DINAMICO: No traduce de una dirección a otra 1 a 1, sino que tenemos un Pull (baúl) de direcciones públicas. Estas direcciones disponibles en el pull se van asignando según van llegando las peticiones. Si nuestro pull de direcciones fueran por ejemplo 5, solo se podrían conectar 5 equipos a la vez: solo podría atender a 5 peticiones a la vez.

El pull de direcciones las da el proveedor de servicio, y además nos tiene que decir

también la máscara de red, que dependerá de la cantidad de IPs públicas que nos ofrezca.

Ejemplo: 200.200.200.0/29

Las direcciones se asignan según van llegando las peticiones. En NAT estático siempre sería la misma dirección, pero en NAT dinámico es según la orden de solicitud.

NAT dinámico utiliza ACLs

Configuración NAT Dinámico:

Dynamic NAT Configuration Steps	
Step 1	Define a pool of global addresses to be used for translation. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
Step 2	Configure a standard access list permitting the addresses that should be translated. **access-list** *access-list-number* **permit** *source*[*source-wildcard*]
Step 3	Establish dynamic source translation, specifying the access list and pool defined in prior steps. **ip nat inside source list** *access-list-number* **pool** *name*
Step 4	Identify the inside interface. **interface** *type number* **ip nat inside**
Step 5	Identify the outside interface. **interface** *type number* **ip nat outside**

PAT: Port Address Traslation

Varios equipos de una misma red utilizan una sola dirección pública porque cambian de puerto. PAT se conoce también como “NAT OVERLOAD”

TABLA NAT/PAT

	IP LOCAL	IP GLOBAL	PUERTO LOCAL (aleatorio)	PUERTO GLOBAL
PC 1	192.168.10.10	Siempre la misma dirección en todos los equipos, en este caso 7.7.7.1	Aleatorio, en este caso 2001	NAT asignado puerto global. En este caso 3101
PC 2	192.168.10.11	7.7.7.1	4122	3102
PC 3	192.168.10.12	7.7.7.1	8193	3103
PC 4	192.168.10.13	7.7.7.1	10100	3104

La dirección 192.168.10.10 al pasar por el router saldrá a internet con la dirección 7.7.7.1 que es la dirección pública.

La dirección 7.7.7.1:3101(puerto asignado) será la salida y el destino la dirección que sea, que en este caso ponemos la 8.8.8.8:80

Así, de manera local, cada equipo tiene su propia IP y un puerto que se asigna de manera aleatoria. Al salir a internet por la interface del router, lo hará con la IP pública 7.7.7.1 y además con un puerto asignado, de tal forma que al salir la petición va con una ip y puerto local 192.168.10.10:2001 y sale con una IP global y además un puerto asignado 7.7.7.1:3101

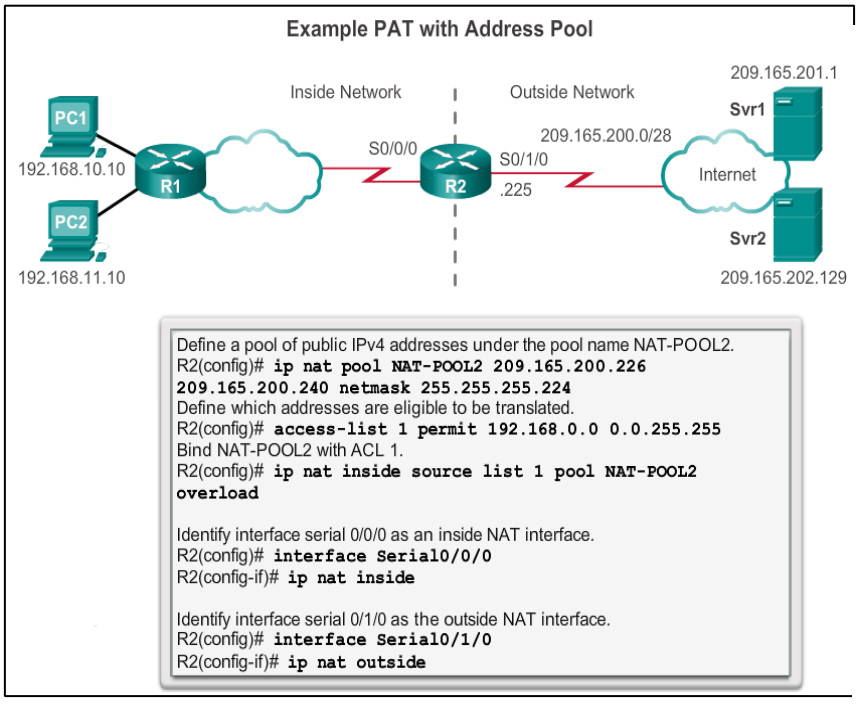
Al regresar los paquetes al router, lo harán también a la dirección 7.7.7.1 pero a distinto puerto. Por eso sabe a qué equipo debe responder. Por eso se puede usar solo una IP, gracias a los puertos.

En NAT Dinámico si el pull está lleno, se usará una misma dirección IP ya usada pero con distinto puerto.

Las ACLs se usan para distinguir entre lo que tiene que traducir y lo que no. NAT y ACL deben estar coordinados.

NAT Dinámico sobrecargado (NAT OVERLOAD) es lo mismo que PAT.

Configuración NAT OVERLOAD/ PAT



Step 1	Define a standard access list permitting the addresses that should be translated. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
Step 2	Establish dynamic source translation, specifying the ACL, exit interface and overload options. **ip nat inside source list** *access-list-number* **interface** *type number* **overload**
Step 3	Identify the inside interface. **interface** *type number* **ip nat inside**
Step 4	Identify the outside interface. **interface** *type number* **ip nat outside**

Port Fordwarding: La misma dirección IP del Router con distintos puertos

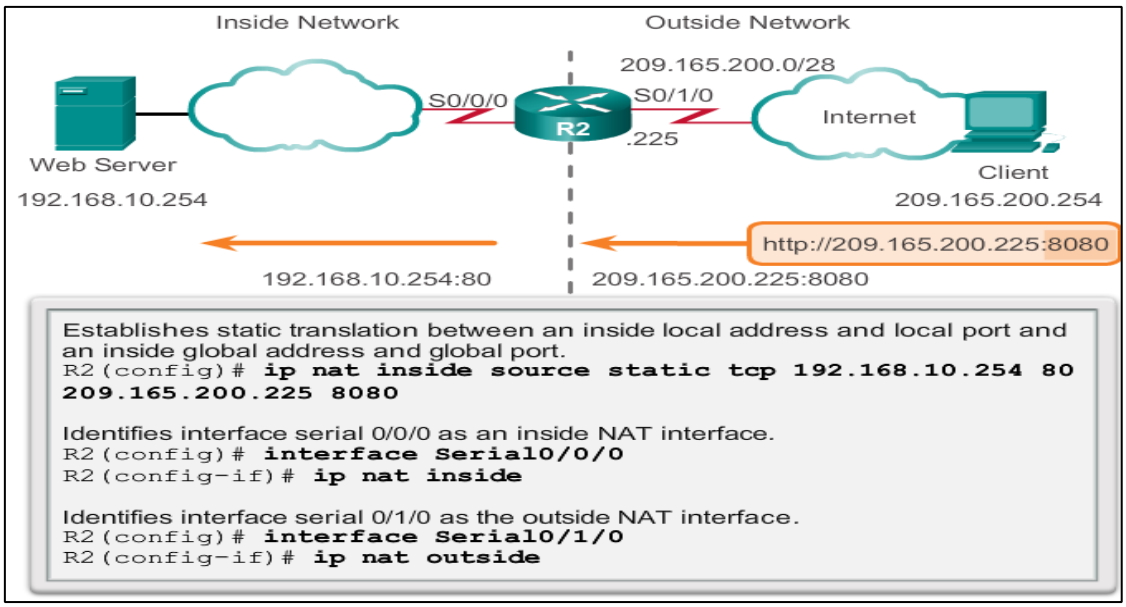


Tabla Port Fordwarding:

Si se apunta a 7.7.7.1:80 lo enviará a 192.168.1.10:80

Si se apunta a 7.7.7.1:21 lo enviará a 192.168.1.11:21

Si se apunta a 7.7.7.1:1042 lo enviará a 192.168.1.12:1042

Port Fordwarding se configura igual que NAT Estático pero añadiendo los puertos

La diferencia entre NAT Overload (PAT) y Port Fordwarding es que uno usa puertos aleatorios (PAT) y el otro puertos asignados (Port Fordwarding)

Port Fordwarding se configura igual que NAT Estático pero añadiendo los puertos

Tema III: Scaling Networks

Capítulo 1: Introducción a las redes escalables

Es sólo literatura. Ver diapositivas

Capítulo 2: Redundancia

La redundancia es hacer un camino de respaldo o de balanceo de carga por si nos fallara la conexión por defecto o principal.

Hay que tener en cuenta varias cuestiones cuando se hace redundancia:

- Puede generar **inestabilidad dentro de la tabla MAC** del Sw ya que le puede entrar PETICIONES DE DESTINO CON LA MISMA MAC EN DISTINTAS INTERFACES DEL SW.
- Se pueden generar **tormentas de broadcast**.
- Se pueden generar **bucles** por la transmisión múltiple de tramas unicast

Una tormenta de broadcast son muchos broadcast a la vez, lo cual a su vez provoca inestabilidad en las tablas CAM del SW y los convierte en inestables generando bucles. Si esto se produce, siempre irá en aumento y terminará por consumir todo el ancho de banda hasta un punto en que el SW se satura y se termina por una denegación de servicio.

Para evitar en la medida de lo posible todos estos inconvenientes derivados de la redundancia, está el **Spanning-tree**

Spanning-tree se asegura de que sólo hay un camino lógico entre todos los destinos en la red mediante el bloqueo de intencionalmente rutas redundantes que podrían causar un bucle.
Un puerto se considera bloqueado cuando los datos de usuario se impide que entren o salgan de ese puerto. Esto no incluye unidad de datos de protocolo de puente (BPDU) tramas que se utiliza STP para evitar bucles.
Existen todavía los caminos físicos para proporcionar redundancia, pero estos caminos se desactivan para evitar los bucles que se produzcan.
Si la ruta es siempre necesario para compensar un cable de red o fallo interruptor, STP vuelve a calcular las trayectorias y desbloquea los puertos necesarios para permitir que la ruta de acceso redundante para convertirse en activo.

Spanning-tree: es el protocolo que evita los bucles de capa 2. Como los SWs expenden los paquetes que les llegan por difusión, si un paquete no encontrara destino, se quedaría dando vueltas entre los SWs y podrían colapsar la red. Además el ST evita los ataques basados en tormentas.
ST escoge de entre todos los SWs uno que será el puente raíz: ROOT BRIDGE

Para escogerlo los SWs se intercambian tramas llamadas BPDUs (Bridge Protocol Dta Unit) y **deciden que SW será el root bridge atendiendo a dos criterios: 1º la prioridad y 2º la dirección mac.** Este valor se denomina BID. **El SW con menor valor de prioridad, será el elegido y caso de tener todos los SWs la misma prioridad, se tomará aquel con la dirección mac más baja**. Por defecto todos los SWs tienen un valor de prioridad de 32768+1(valor por defecto de la VLAN 1)+ la MAC.
Esta elección se toma por defecto, pero lo más recomendable es elegir como SW puente raíz (Root Bridge) al SW con mejor capacidad de gestión. El valor de la prioridad se puede modificar manualmente.
En el caso de **PVST+** (con enlaces troncales) que contempla también las VLAN, el valor de prioridad de la misma será 32768+ el valor decimal que le hayamos dado a la VLAN+ la MAC: si es la VLAN 30, su prioridad será 32768+30=32798+ la MAC
En el caso también de PSTV+ habrá tantos bridge-id como VLANs. Cada SW, de forma local, tendrá tantas instancias como VLANs tenga: si 4 VLANs ⇒ 4 Root-ID y 4 Bridge-ID.

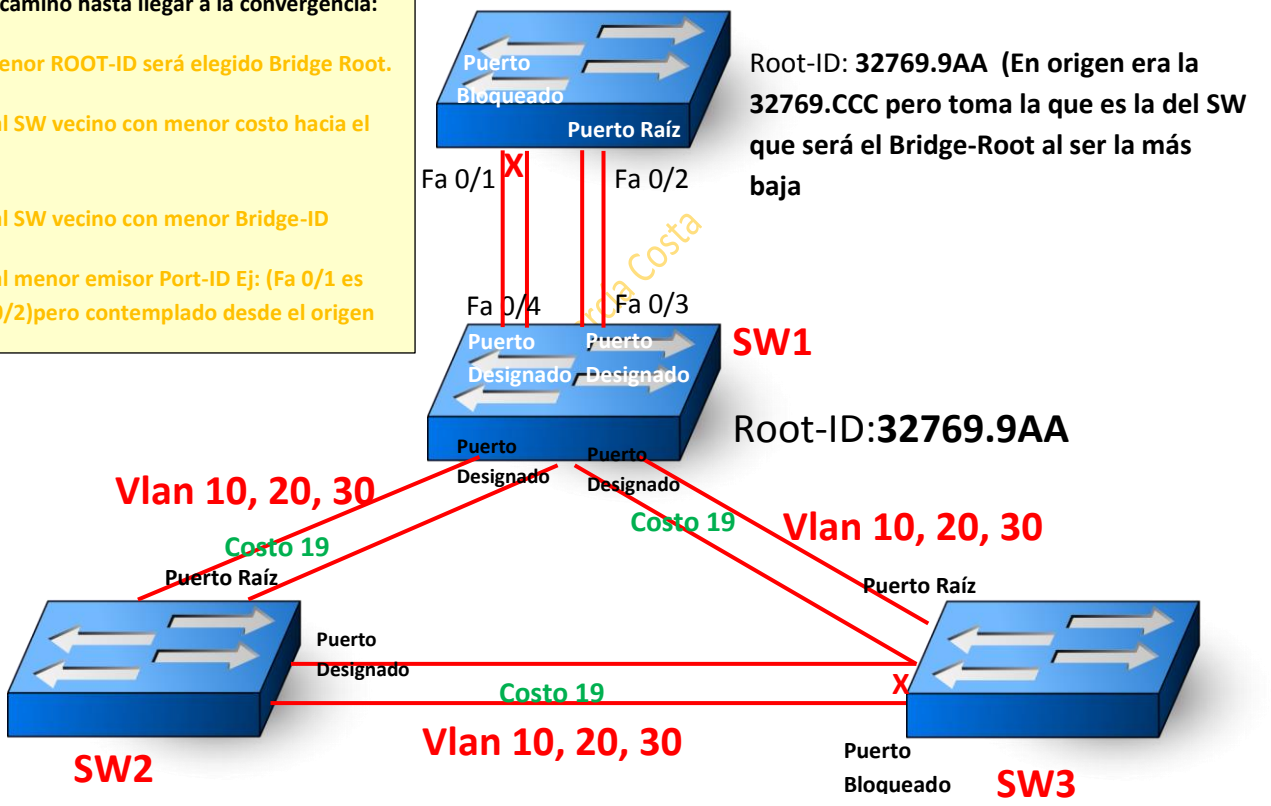
Selección del camino hasta llegar a la convergencia:

1º El SW con menor ROOT-ID será elegido Bridge Root.

2º Se prefiere al SW vecino con menor costo hacia el puente raíz.

3º Se prefiere al SW vecino con menor Bridge-ID

4º Se prefiere al menor emisor Port-ID Ej: (Fa 0/1 es menor que Fa 0/2)pero contemplado desde el origen



Root-ID: **32768+1.CCC** **(En origen era la 32769.CCC pero toma la que es la del SW que será el Bridge-Root al ser la más baja**

Root-ID: **32769.9AA (en origen era la 32769.AAA)**

Bridge-ID Vlan 1: **32768+1.AAA**

Bridge-ID Vlan **10: 32768+10.AAA**

Bridge-ID Vlan 20: **32768+20.AAA**

Los SWs de intercambian los BPDUs y el que tenga menor Root-ID se convertirá en en BRIDGE-ID. En nuestro caso será el SW1 porque su Root-ID es 32769.9AA

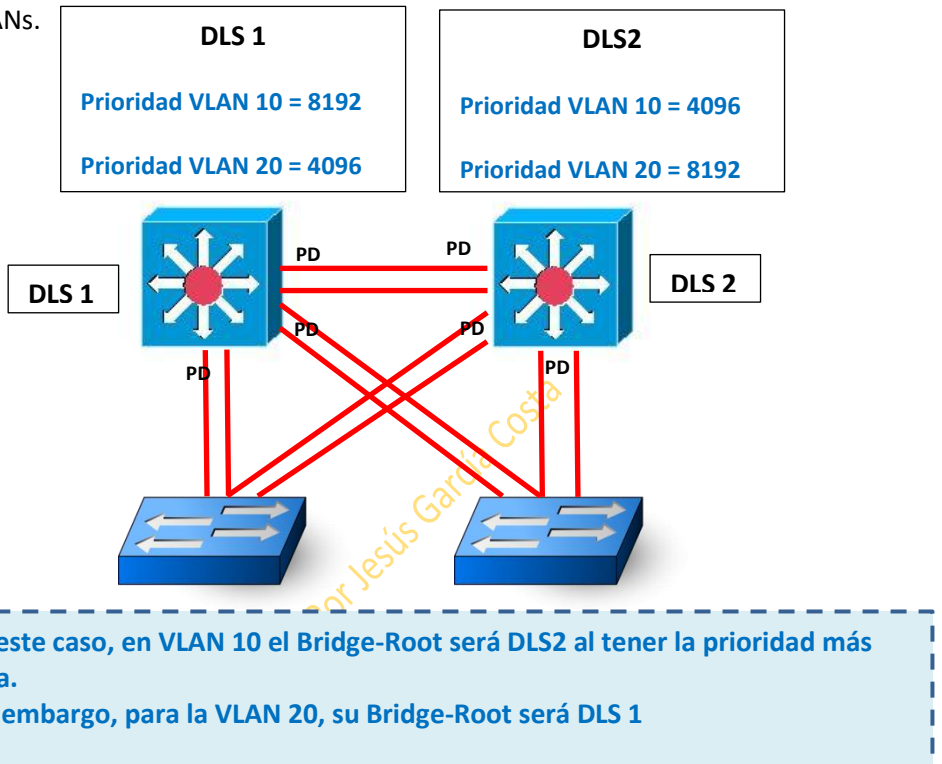
Root-ID: **32769.9AA (En origen la suya era la 32769.BBB)**

Bridge-ID Vlan 1: **32768+1.BBB**

Bridge-ID Vlan 10: **32768+1.BBB**

Bridge-ID Vlan 20: **32768+20.BBB**

La prioridad de un SW tiene que ser siempre múltiplo de 4096 que es valor que tienen 4 bits.
Además, se elegirá el puerto raíz que será el puerto con menor costo hacia el Bridge-ID
Hay una serie de costos por defecto según el tipo de enlace físico que se utilice. A mayor velocidad, menos costo.
Por eso, a pesar de estar más cerca del SW Bridge-ID si su enlace físico es peor y consecuentemente de mayor costo, no irá por el camino físicamente más corto, sino por el que menor coste tenga aunque tenga que dar más vuelta.
PVST+ requiere muchos recursos ya que al ir por las VLANs también, manda muchos BPDUs.
La prioridad siempre acostumbra a ser el SW que tenga la MAC más baja, ya que por defecto todos los SWs tienen la misma prioridad, así que es la MAC la que decide.
Pero la prioridad se puede cambiar manualmente para que el SW Bridge-Root sea el que decidamos nosotros. Es el valor más bajo el que conseguirá ser el Bridge-Root. Lo mismo pasa con las VLANs.



En este caso, en VLAN 10 el Bridge-Root será DLS2 al tener la prioridad más baja.
Sin embargo, para la VLAN 20, su Bridge-Root será DLS 1

El comando para dar prioridad a un SW y provocar cambios en el valor de prioridad:
SW(Config-if)#spanning-tree port-security 112 (112 es solo un ejemplo)

También si hay Vlanes podemos hacer que según vlan, cambie de SW ROOT y así acomodarse a la topología de esa vlan:
SW(Config)#spanning-tree vlan 1 priority 2500

Para darle coste manual a un puerto:
SW(Config-if)#spanning-tree cost 25

Una vez elegido el SW root bridge, para saber qué puerto se taponará para evitar los bucles, se hace atendiendo al coste de cada enlace de tal forma que el camino para llegar al root bridge sea el más rápido. Ej: si entre dos SWs existe una conexión de Giga y otros de Ethernet normal, se taponará la Ethernet normal para que el camino sea el de Giga ya que será más rápido. Elemental, vaya.
Todos los puertos del SW Root Bridge son "**puertos designados**". Esto es: **pueden enviar y recibir tramas.**

Los SWs que no son el Root Bridge pero que alguno de sus puertos dan al Root Bridge, se llaman "**puertos raíz**"(solo puede haber 1 puerto raíz en esos SWs)
Los **puertos designados** de los SWs que no son el root bridge son los susceptibles de ser bloqueados para evitar los bucles.
Un **puerto bloqueado** sí que puede recibir BPDUs pero no tramas.
Recordamos que las tramas no tienen campo ttl (tiempo para desactivarse) por ello son las que se evitan en los puertos bloqueados para evitar colapsos.
Los puertos caídos son shutdown.
Además de ataques de tormenta de broadcast, el ST también nos protege de los ataques de tramas de unicast duplicadas.

El ST también asegura que exista una solo ruta lógica entre todos los destinos de la red. Cisco tiene habilitado por defecto el protocolo STP (spanning tree protocol)
El algoritmo que usa STP es el STA. Es el STA el que escoge al root bridge.
Todos los SWs intercambian las tramas BPDUs y así escogen al root bridge y el que tenga un BID Mezcla del valor entre prioridad y mac) menor será el elegido. Cada BPDU tiene un BID que identifica al SW que envió la bpdu.
Con un show vemos que SW es el root.
Una vez decidido quien es Root y el camino a seguir, se le da una categoría a los puertos.
Puerto raíz: Puerto/interface de los SWs que no es el root, que está más cercano al root. Solo será 1
Puerto designado: todos los puertos que no son raíz y que aun pueden enviar y recibir tráfico, tramas y bpdu. Todos los puertos del root bridge son designados, pero también puede haber puertos designados en SW que no son el root. Solo se permite un puerto designado por enlace.
Puerto no designado: Pueden recibir bpdus pero no tramas. Son los puertos que impiden los bucles. Son los puertos bloqueados.
Puerto deshabilitado: los que están en shutdown.

- Todos los puertos del SW root son designados.
- No hay puertos raíz en el SW raíz.

Los estados de los puertos son:
Bloqueo: cuando no es designado. Permite bpdu pero no tramas
Escucha:
Aprendizaje:
Envío:
Con el portfast hacemos que se salte los pasos de escucha y aprendizaje.
Recordamos que el portfast solo se aplica en los SWs que dan a los hosts.

Si hay cambios en la topología de red, tiene que haber una forma de que el root bridge se entere. Cuando se entera, hace un broadcast para que se enteren los demás SWs.
TCN es una bpdu especial que envía un SW al root bridge cuando hay un cambio topológico que le afecta, como podría ser que una de sus interfaces pasara a shutdown, o al revés, etc…
TCA es la respuesta del Root Bridge al SW que le envió la TCN, como acuse de recibo.
Después el Root mandará un broadcast.

Roles de puerto:
- Puerto Raíz
- Puerto designado
- Puerto no designado (Bloquing)

Estado del puerto:

- Deshabilitado
- Bloqueado
- Escucha
- Aprendizaje
- Reenvío

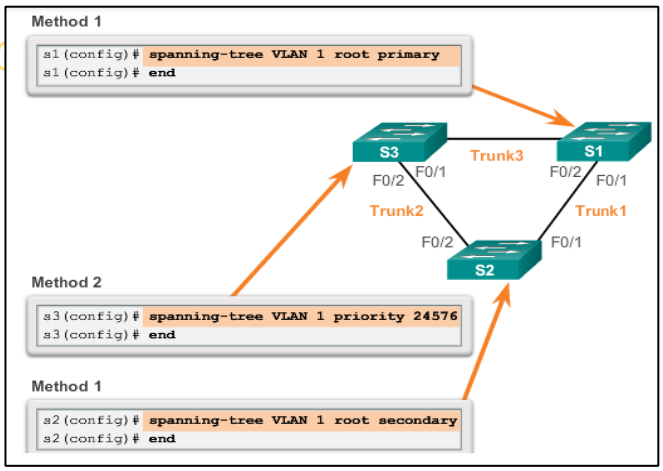
Temporizadores:

- Saludo: 2 segundos
- Retardo de envío: 15 segundos
- Envejecimiento de BPDU: 20 segundos

Operación de Spanning-Tree:

- Elección del puente raíz Roote Bridge: Vocero
- Elección del **puerto raíz**: Puerto/interface de los SWs que no es el root, que está más cercano al root. Solo será 1
- Elección de los **puertos designados**: : todos los puertos que no son raíz y que aun pueden enviar y recibir tráfico, tramas y BPDU. Todos los puertos del root bridge son designados, pero también puede haber puertos designados en SW que no son el root. Solo se permite un puerto designado por enlace.
- **Puertos bloqueados**

***El SW Bridge no tiene puerto raíz**

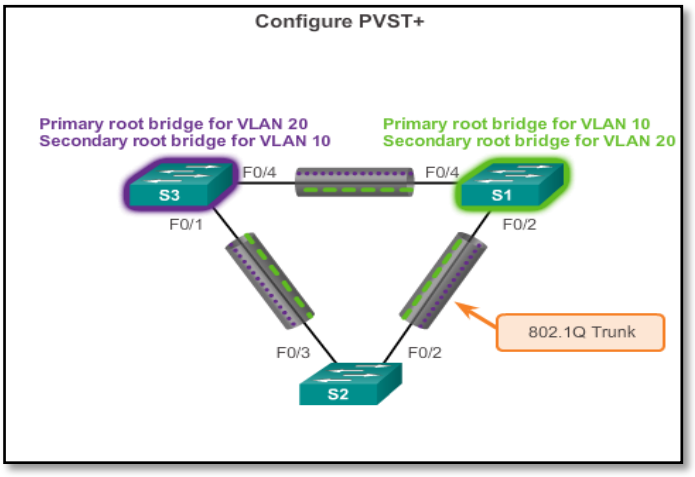


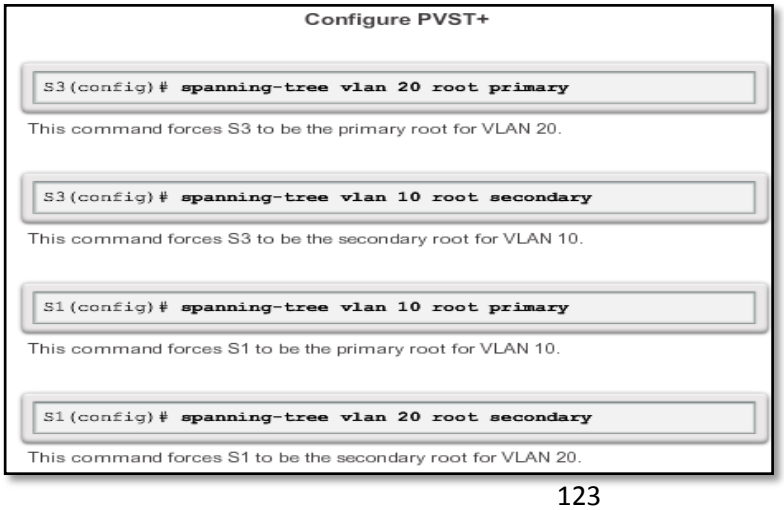
Configuración de Spanning-Tree:

1- Para habilitar STP **S(Config)#spanning-tree mode pvst/rapid-pvst** según el modo en el que estemos.

2- Para decir si una VLAN es primaria o secundaria: **S(Config)#spanning-tree vlan x root primary/secondary**

3- Para darle una prioridad en concreto manualmente a una vlan: **S(Config)#spanning-tree vlan x priority + valor** *Recordar que una misma Vlan puede ser primary en un SW y secondary en otro SW como vemos en el ejemplo de abajo.





4- Para habilitar **PortFast y BPDU Guard,** siempre será en las interfaces que conectan con dispositivos finales o hosts. Nos metemos en la interface y aplicamos el comando: **S(Config-if)#spanning-tree portfast**

5- Un vez habilitado el portfast, habilitamos también el BPDU Guard con el comando**:** **S(Config-if)#spanning-tree bpduguard enable**

```
S2(config)# interface FastEthernet 0/11
S2(config-if)# spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to
a single host. Connecting hubs, concentrators, switches,
bridges, etc... to this interface  when portfast is enabled,
can cause temporary bridging loops.
 Use with CAUTION

%Portfast has been configured on FastEthernet0/11 but will only
 have effect when the interface is in a non-trunking mode.
S2(config-if)# spanning-tree bpduguard enable
S2(config-if)# end
```

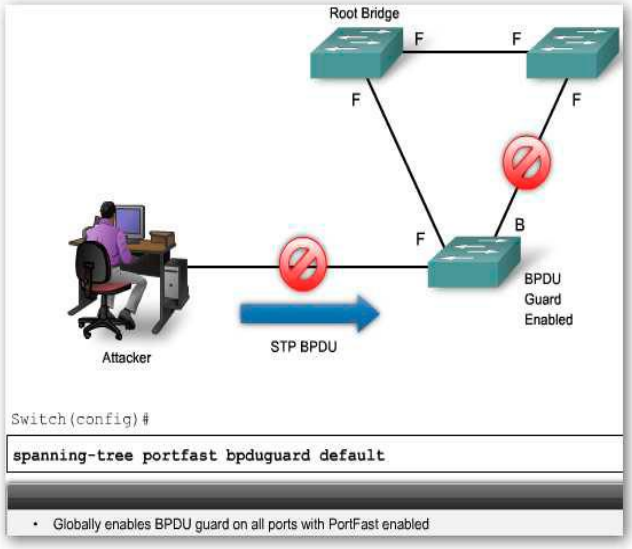
- Son las que nos interesan

PortFast, BackboneFast y UPLinkFast ayudan a la convergencia de STP.

BPDU Guard, BPDU Filter, Root Guard y VDLD ayudan a la seguridad de STP

Prevenir ataques que comprometen la capa 2 se puede configurar la seguridad de puerto, BPDU guard, Root guard, control de tormentas, SPAN y RSPAN.

Para que un puerto de un SW no tenga que estar pasando por todos los estados de aprendizaje, escucha, etc (estados de transición), usamos el PortFast. **PORTFAST**: Hace que los puertos/interfaces del SW se habiliten inmediatamente sin pasar por las 4 fases de los sw: escucha, aprendizaje, etc.



El PortFast siempre hay que hacerlo en las intefaces de acceso a los hosts o servidores, no a las interfaces conectadas a otros SWs. El portFast y el BPDUGuard se habiltan en este tipo de interfaces precisamente para evitar que un PC o SRV pudiera enviar BPDUs haciéndose pasar por un SW Bridge.
Si un PC mandara un BPDU sería un ataque y cconsecuentemente se generaría un error llamado “error-disabled”

Al configurar manualmente la prioridad en una VLAN habrá que hacerlo dando valores múltiplos de 4096.

Spanning-Tree Mode: Los SWs Cisco 2960 y 3560 pueden trabajar en modo RPVST +

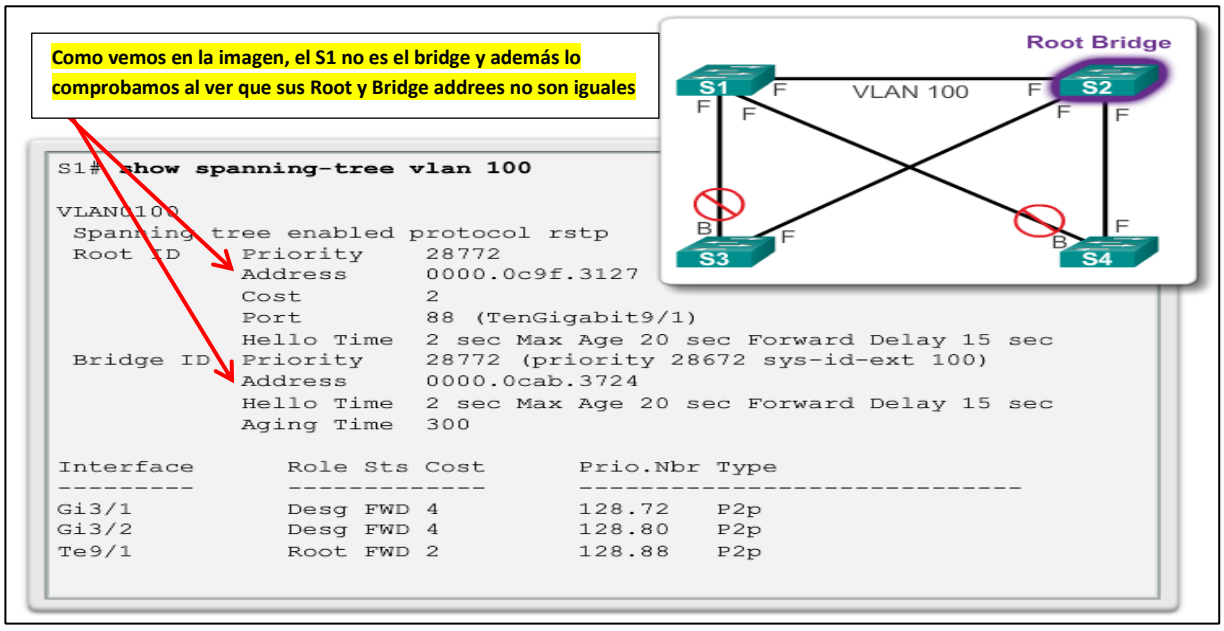
```
S1# configure terminal
S1(config)# spanning-tree mode rapid-pvst
S1(config)# interface f0/2
S1(config-if)# spanning-tree link-type point-to-point
S1(config-if)# end
S1# clear spanning-tree detected-protocols
```

Cisco IOS Command Syntax	
Enter global configuration mode.	`configure terminal`
Configure Rapid PVST+ spanning-tree mode.	`spanning-tree mode rapid-pvst`
Enter interface configuration mode and specify an interface to configure. Valid interfaces include physical ports, VLANs, and port channels.	`interface` *interface-id*
Specify that the link type for this port is point-to-point.	`spanning-tree link-type point-to-point`
Return to privileged EXEC mode.	`end`
Clear all detected STP.	`clear spanning-tree detected-protocols`

Recordar que se pueden modificar los costos y las prioridades manualmente.

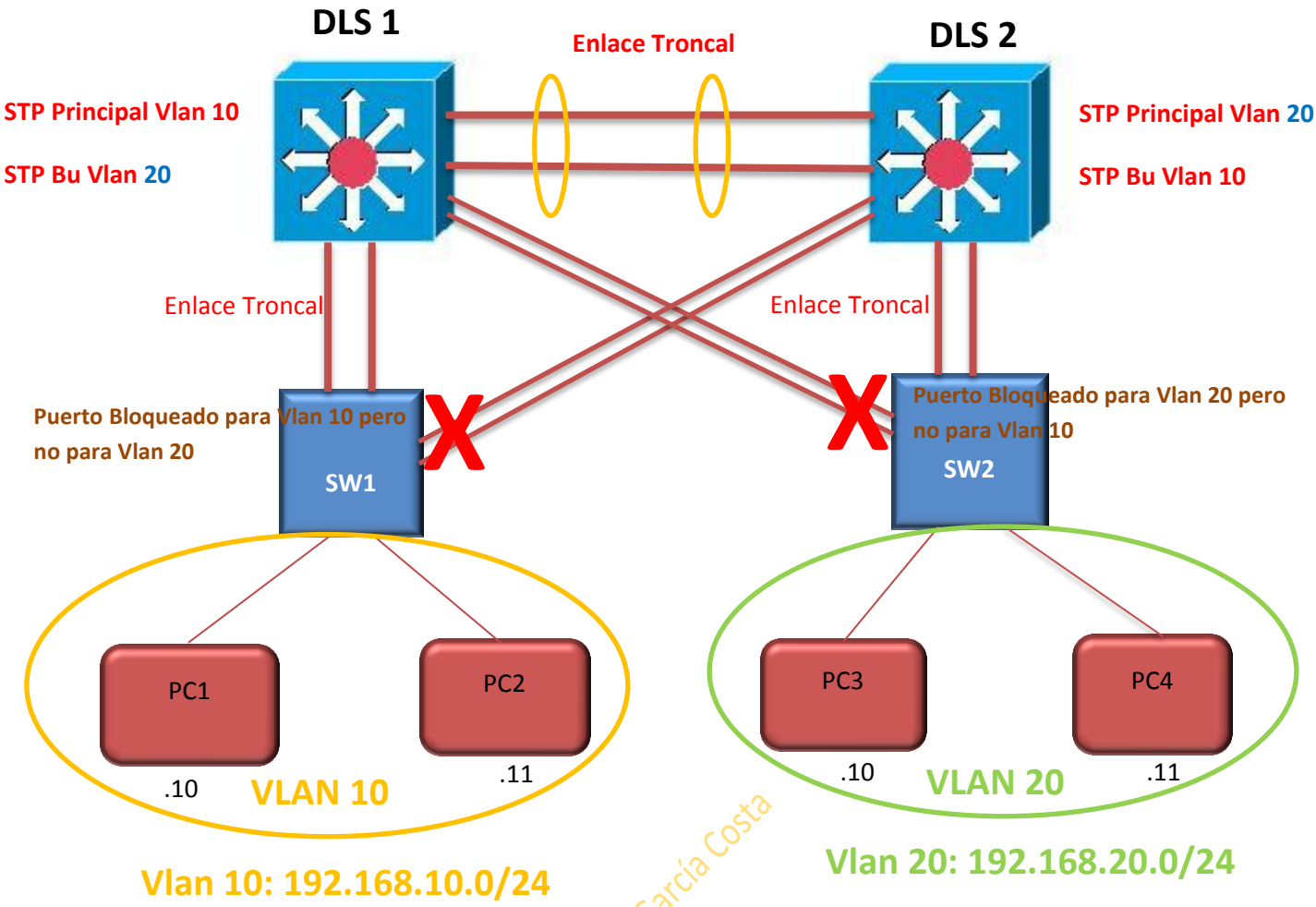
Recordar también que lo normal es que el SW Bridge esté en el núcleo, no en la parte de acceso.

Con el comando show spanning-tree vlan x podremos saber a simple vista si es o no el SW Bridge: si la address del root y del bridge son iguales, será el SW Bridge, sino, no.



Protocolos de redundancia de 1er salto:

- HSRP
- VRRP
- GLBP



STP traza el camino de un SW cualquiera de la red hacia el SW Bridge. En este escenario:

Para **DLS 1** el default Gateway de vlan 10 sería por ejemplo la primera dirección: 192.168.10.1 = SVI 10 y para la vlan 20 la 192.168.20.1 = SVI 20

Para **DLS 2** la Gateway de vlan 10 sería la 192.168.10.2 = SVI 10 y la de la vlan 20 192.168.20.2 = SVI 20

La default Gateway de vlan 10 será 192.168.10.1 porque solo hay un salto hasta conseguirla.

La default Gateway de vlan 20 será 192.168.20.2 porque solo hay un salto hasta conseguirla en vez de dos.

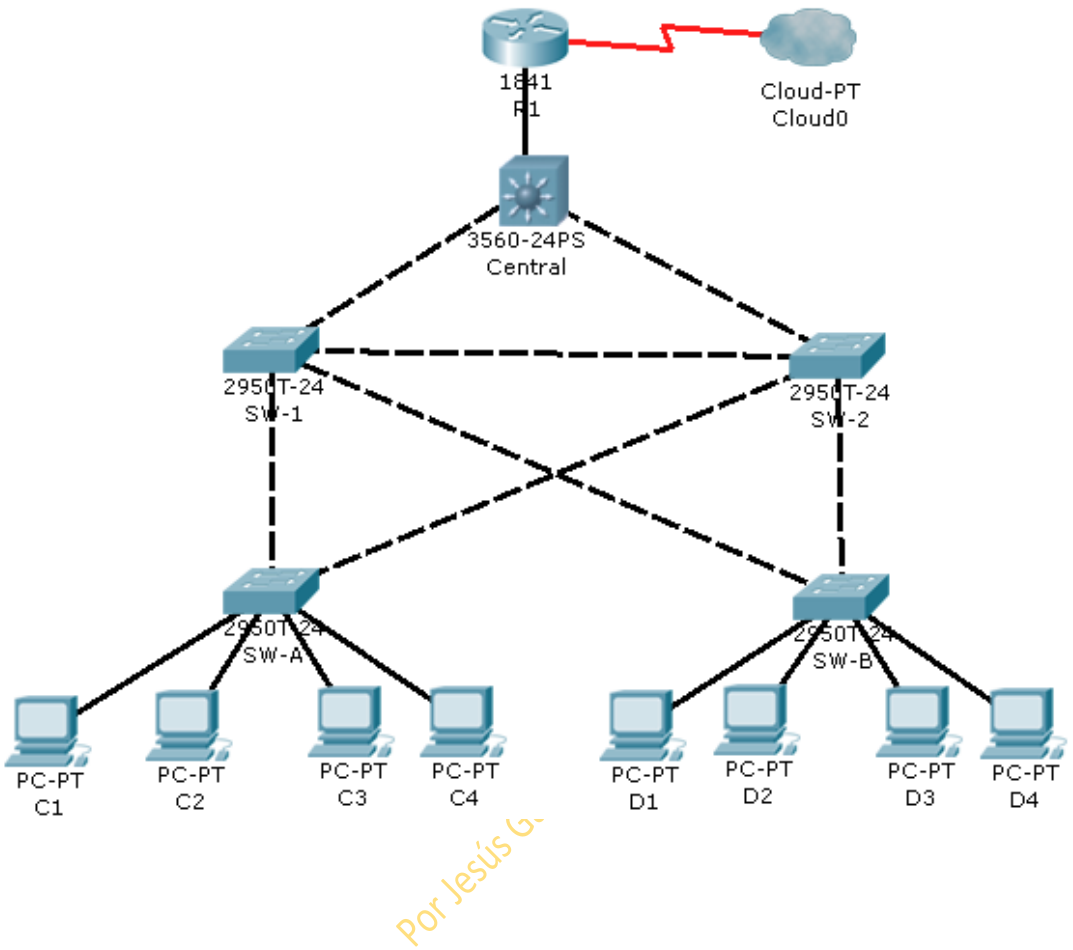
Si Router Virtual:

- SVI 10: 192.168.10.254
- SVI 20: 192.168.20.254

La default gateway para vlan 10 sería 192.168.10.254

La default Gateway para vlan 20 sería 192.168.20.254

Ejemplo de configuración ampliada de Spanning-Tree:



1º Configurar el rootbridge. Asignar al SW central como puente raíz=root bridge.

Nos metemos en el SW Central y Central(Config)#spanning-tree vlan 1 root primary
Con esto estamos configurando manualmente el spanning-tree en vez de dejar que lo hagan automáticamente los Sws entre sí.
Es más, le estamos diciendo al SW Central que será el root primario y al SW-1 que será el secundario. (es como asignar el DR y el BDR)
Para decirle al SW-1 que será el secundario lo hacemos con el comando:
SW-1(Config)#spanning-tree vlan 1 root secondary

2º Para protegernos de ataques STP:

2.1 Aplicamos portfast:

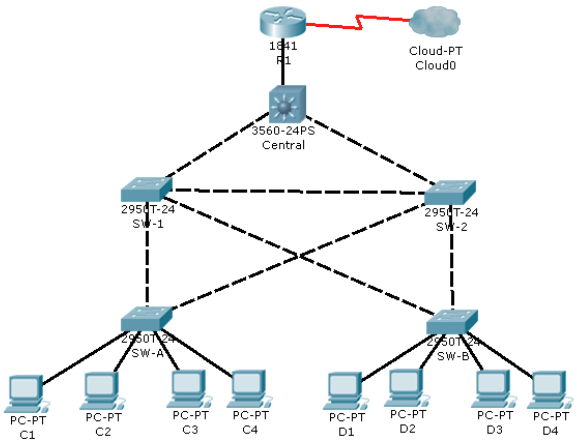
Nos vamos al SW-A y en config le ponemos el comando interface range para habilitar un rango entero de interfaces (desde hasta las interfaces que seleccionemos).
**en el ejercicio pone desde la 0/1 – 4 pero en nuestro ejercicio son de la 0/4 a la 0/7*

Con esto sólo nos hemos referido a las interfaces seleccionadas, pero todavía no hemos hecho nada con ellas.

```
SW-A(config)# interface range fastethernet 0/1 – 4
```

```
SW-B(config)# interface range fastethernet 0/1 - 4
```

Con el comando spanning-tree portfast habilitamos el portfast en todas esas interfaces ya que como vemos el prompt cambió a config-if-range.




```
SW-A(config-if-range)#
spanning-tree portfast
SW-B(config-if-range)#
spanning-tree portfast
```

Ojo!! El comando portfast solo se puede aplicar a interfaces que dan a hosts

La función PortFast causa que una interfaz de acceso, realice la transición del estado bloqueado a envío de forma inmediata, sin pasar por los estados de escucha y aprendizaje.

Como hemos visto, hacemos lo mismo con el SW-B

2.2 Habilitamos BPDU GUARD

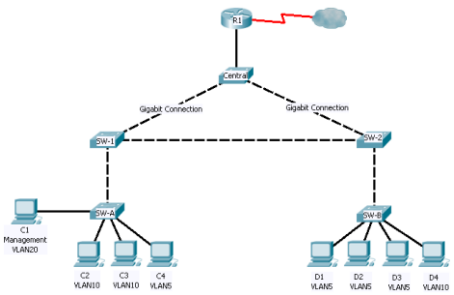
Cogemos el mismo rango de interfaces en ambos SWs y le aplicamos el comando:
SW-A/B(Config-if-range)#spanning-tree bpduguard enable

```
SW-A(config)# interface range fastethernet 0/1 - 4
SW-A(config-if-range)# spanning-tree bpduguard enable

SW-B(config)# interface range fastethernet 0/1 - 4
SW-B(config-if-range)# spanning-tree bpduguard enable
```


2.3 Habilitamos ROOT GUARD

* el Puerto/interface raíz de un SW es aquel que conecta con el SW ROOT. Aquí JAMÁS habilitar el root guard. Nos vamos al SW-1 y al SW-2 y como vemos en la imagen, una de sus interfaces da al root bried con lo cual habrá que tener cuidado de no habilitar en ellas el root guard. Sí a las demás interfaces.



En el ejercicio de muestra, las interfaces que elije son la 23 y la 24. En nuestro propio ejercicio serán la 3 y la 4 en ambos SWs. El comando para habilitar en las interfaces correspondientes el root guard es spanning-tree guard root

```
SW-1(config)# interface fa0/23
SW-1(config-if)# spanning-tree guard root
SW-1(config-if)# interface fa0/24
SW-1(config-if)# spanning-tree guard root
SW-2(config)# interface fa0/23
SW-2(config-if)# spanning-tree guard root
SW-2(config-if)# interface fa0/24
SW-2(config-if)# spanning-tree guard root
```

3.1 Habilitamos el control de tormentas

El control de tormentas hay que habilitarlo en los SWs Central, 1 y 2 ya que son éstos los SWs los principales, los cuales son los susceptibles de ser atacados de este modo.

Así que, nos vamos a todas las interfaces de cada cual y le damos el comando:

Central(Config-if)#storm-contol broadcast level 50

El control de tormentas lo podemos configurar para evitar ataques de broadcast, multicast o unicast. En este caso el ejercicio nos pide que sea broadcast, que por otra parte es lo que toma por defecto.

Level 50 lo ponemos porque ése es el umbral que le queremos poner como máximo precisamente para evitar las tormentas de paquetes.

```
SW-1(config)# interface gi1/1
SW-1(config-if)# storm-control broadcast level 50
SW-1(config-if)# interface fa0/1
SW-1(config-if)# storm-control broadcast level 50
SW-1(config-if)# interface fa0/23
SW-1(config-if)# storm-control broadcast level 50
SW-1(config-if)# interface fa0/24
SW-1(config-if)# storm-control broadcast level 50
**Repeat on SW-2 (gig1/1, fa0/1, fa0/23, and Central (gig0/1,
gig0/2, fa0/1) connection to other switches.
```

4.1 Configuramos como puertos básicos y deshabilitamos los puertos no usados.

Esto se hace en todas las interfaces que conectan con los hosts. Además le configuramos para que el número máximo de macs permitidas sean 2 y a su vez permitimos que las mac las aprenda dinámicamente y que el SW se apague en caso de violación de esta configuración para evitar ataques.

Para ello nos vamos a los SWs que dan soporte a los hosts que son el SW A y el B.

SW-A(Config)#interface fa 0/4, 0/5, 0/6 y 0/7 (en nuestro caso)

Una vez en las interfaces correspondientes le ponemos el comando:

1- SW-A(Config-if)#switchport mode access
   Tenemos que poner este comando para decirle a la interface que está en modo acceso, ya que *para configurar el port-security siempre debe ser en modo Access.*
2- Una vez lo tenemos ya en modo acceso, para configurar la seguridad le damos el comando: SW-A(Config-if)#switchport port-security
3- Una vez metidos en la configuración de la seguridad, para decirle que le permitimos como máximo 2 macs, lo hacemos con el comando:
   SW-A(Config-if)#switchport port-security maximum 2
4- Después para decirle que apague el SW si se viola esta norma, lo hacemos con el comando: SW-A(Config-if)#switchport port-security violation shutdown
5- Por último, para decirle que aprenda dinámicamente las macs lo hacemos con el comando: SW-A(Config-if)#switchport port-security mac-address sticky

Recordamos que en nuestro ejercicio, las interfaces donde hay que aplicar todas estas órdenes son de la fa 0/4 a la fa 0/7 y en los SWs A y B

```
Example:
SW-A(config)# interface FastEthernet 0/1
SW-A(config-if)# switchport mode access
SW-A(config-if)# switchport port-security
SW-A(config-if)# switchport port-security maximum 2
SW-A(config-if)# switchport port-security violation shutdown
SW-A(config-if)# switchport port-security mac-address sticky
```

```
**Repeat on other ports in SW-A and SW-B
```

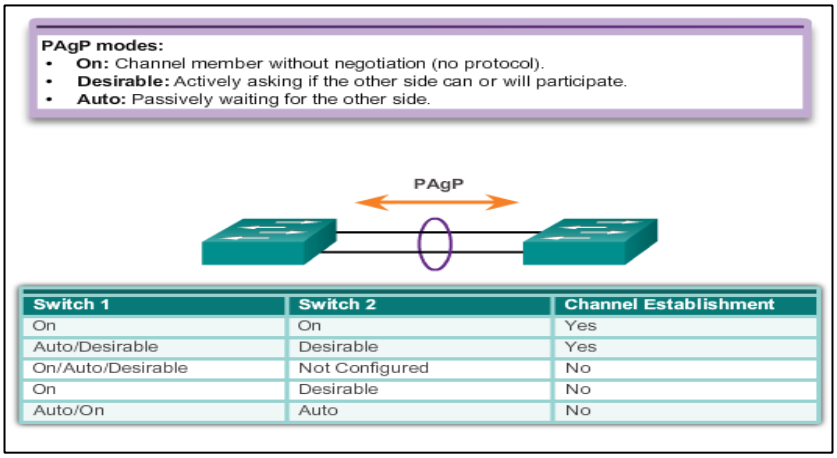
4.2 Verificamos y vemos que todo está ok.

4.3 Deshabilitamos las interfaces que no vamos a usar.
Para ello en vez de ir interface a interface en ambos SWs, ya que hay hasta 24, usamos range para cogerlas todas a la vez y decirles la orden de shutdown. Lo hacemos con el comando:
SW-A(Config)#interface range fa 0/8 – 24 Esas son en nuestro caso las interfaces que deshabilitaremos.
El prompt cambiará y se pondrá SW-A(Config-if-range)# y es entonces cuando le damos la orden de shutdown, de tal forma que quedará: SW-A(Config-if-range)#shutdown.
Hacemos lo mismo con el SW-B.

Capítulo 3: Conceptos de agregación de enlace Etherchannel

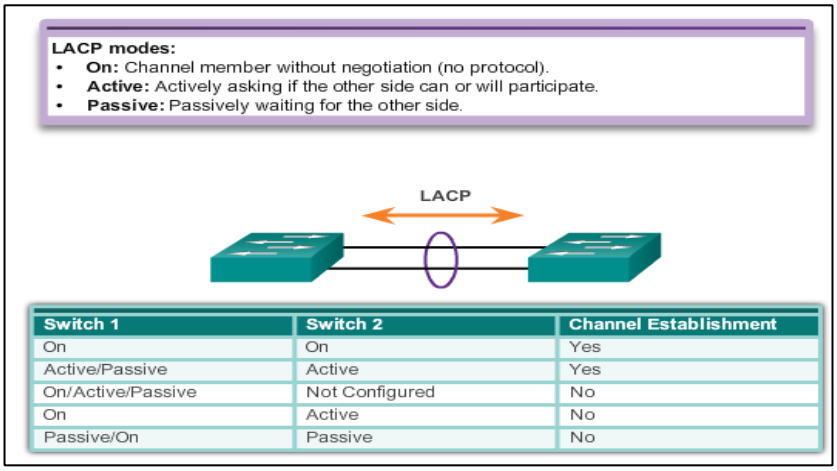
Hay principalmente dos protocolos:

1- **PAgP**: Port Aggregation Protocol



Switch 1	Switch 2	Channel Establishment
On	On	Yes
Auto/Desirable	Desirable	Yes
On/Auto/Desirable	Not Configured	No
On	Desirable	No
Auto/On	Auto	No

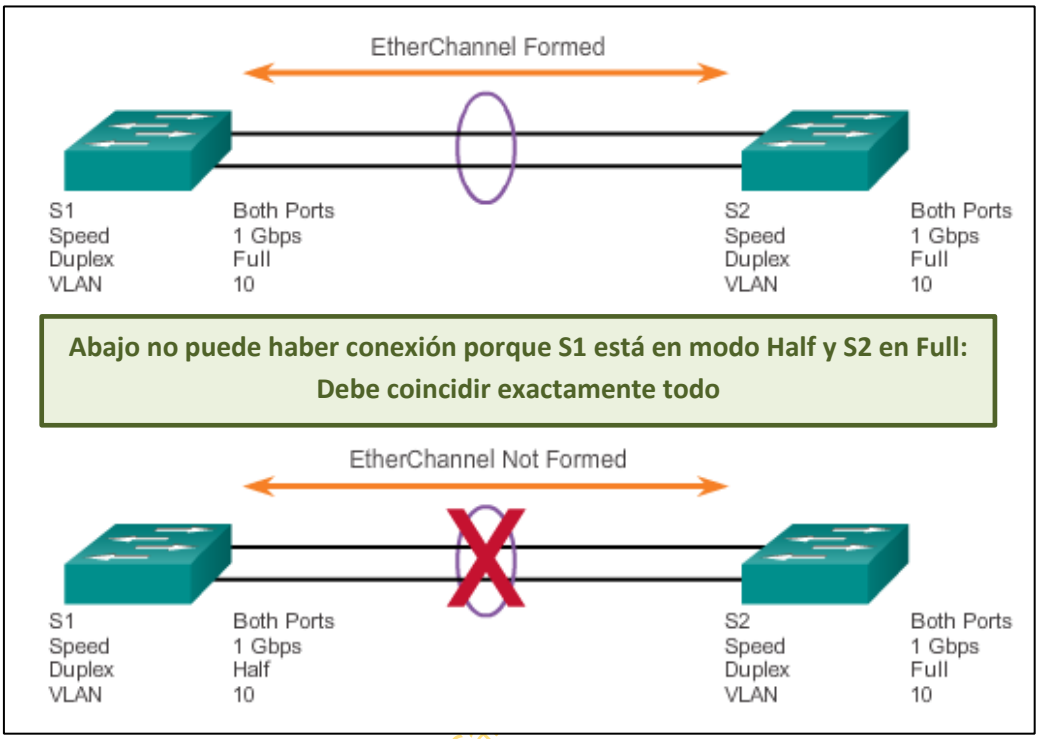
2- **LACP:** Link Aggregation Protocol



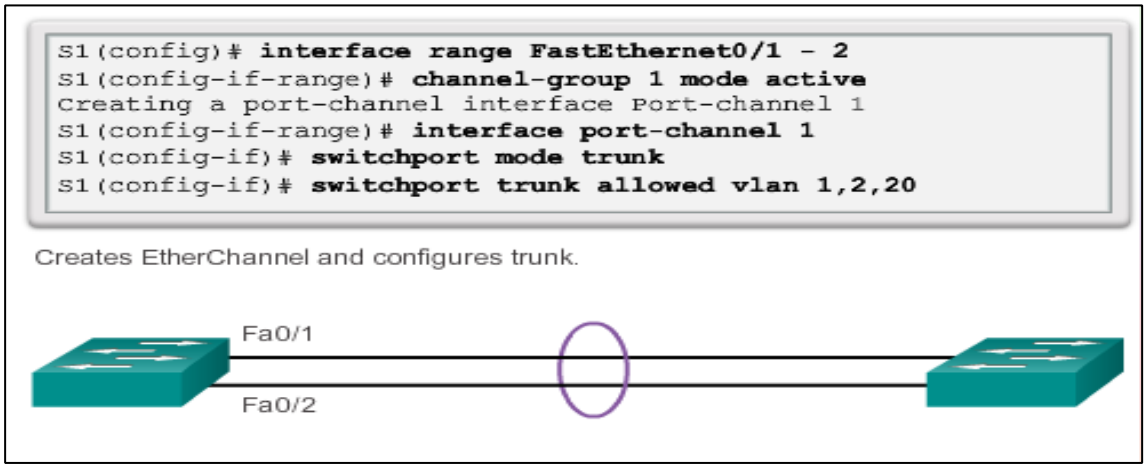
Switch 1	Switch 2	Channel Establishment
On	On	Yes
Active/Passive	Active	Yes
On/Active/Passive	Not Configured	No
On	Active	No
Passive/On	Passive	No

Las condiciones para que haya conexión son

1- La VELOCIDAD Y FULL DUPLEX tienen que coincidir
2- Los TRONCALES DEBEN TENER EXACTAMENTE LA MISMA CONFIGURACIÓN. Pueden estar también en modo acceso pero tendrían que estar los dos



Configuración de etherchannel:

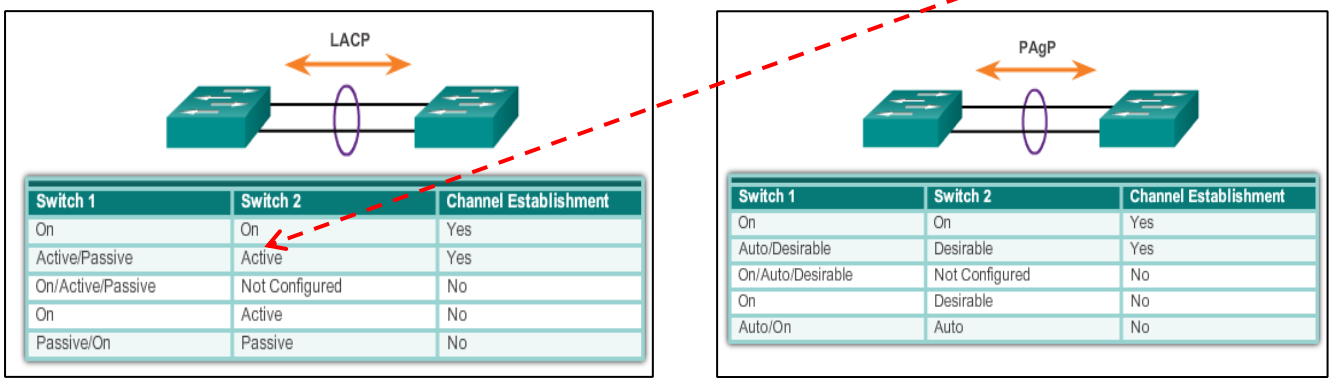


Tal como vemos:

1º Nos vamos al primer SW

1- Nos metemos en las interfaces de SW1 que van a soportar la conexión.

2- Les decimos que van a formar parte de un grupo con: **S(Config-if)#channel-group** +**1** (número que le queramos dar al grupo) + **mode** (en este caso ha cogido "**active**" pero podía haber sido cualquiera que admita según que protocolo usemos PAgP o LACP). Al



Switch 1	Switch 2	Channel Establishment
On	On	Yes
Active/Passive	Active	Yes
On/Active/Passive	Not Configured	No
On	Active	No
Passive/On	Passive	No

Switch 1	Switch 2	Channel Establishment
On	On	Yes
Auto/Desirable	Desirable	Yes
On/Auto/Desirable	Not Configured	No
On	Desirable	No
Auto/On	Auto	No

activar en este caso el modo active, implicitamente estamos activando el protocolo LACP, ya que active es uno de sus modos.

3- Le decimos que va a ser troncal: **S(Config-if)#switchport mode trunk**

4- Le decimos que vlan va a permitir: **S(Config-if)#switchport trunk allowed vlan 1,2,20**

```
S1(config)# interface range FastEthernet0/1 - 2
S1(config-if-range)# channel-group 1 mode active
Creating a port-channel interface Port-channel 1
S1(config-if-range)# interface port-channel 1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk allowed vlan 1,2,20
```

2º Nos metemos en el SW2 y lo configuramos exactamente igual

Lo único que puede cambiar, evidentemente son las interfaces del SW2 que no tienen por qué coincidir en número con las de SW1.

Los comandos Show siempre son importantes:

- **`show interface Port-channel`** – Displays the general status of the EtherChannel interface.
- **`show etherchannel summary`** – Displays one line of information per port channel.
- **`show etherchannel port-channel`** – Displays information about a specific port channel interface.
- **`show interfaces etherchannel`** – Provides information about the role of the interface in the EtherChannel.

Capítulo 4: Wireless LANs

Tecnologías Wireless:

Wireless networks can be classified broadly as:

- **Wireless personal-area network (WPAN)** – Operates in the range of a few feet (Bluetooth).
- **Wireless LAN (WLAN)** – Operates in the range of a few hundred feet.
- **Wireless wide-area network (WWAN)** – Operates in the range of miles.
- **Bluetooth** – An IEEE 802.15 WPAN standard; uses a device-pairing process to communicate over distances up to .05 mile (100m).
- **Wi-Fi (wireless fidelity)** – An IEEE 802.11 WLAN standard; provides network access to home and corporate users, to include data, voice and video traffic, to distances up to 0.18 mile (300m).
- **Worldwide Interoperability for Microwave Access (WiMAX)** – An IEEE 802.16 WWAN standard that provides wireless broadband access of up to 30 mi (50 km).
- **Cellular broadband** – Consists of various corporate, national, and international organizations using service provider cellular access to provide mobile broadband network connectivity.
- **Satellite Broadband** – Provides network access to remote sites through the use of a directional satellite dish.

Wireless Lan Control (WLC)

Wireless Control System (WCS)

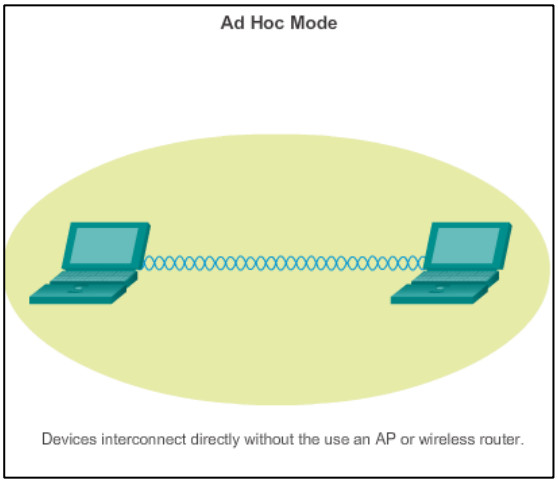
Generalmente un SSID corresponde a una VLAN diferente

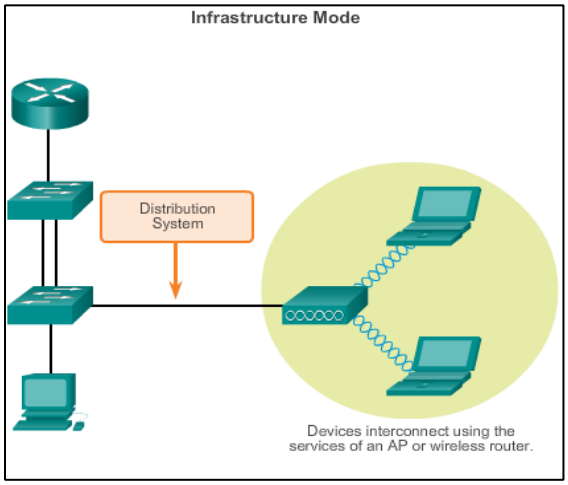
Modos de Wireless:

Ad-Hoc: punto a punto

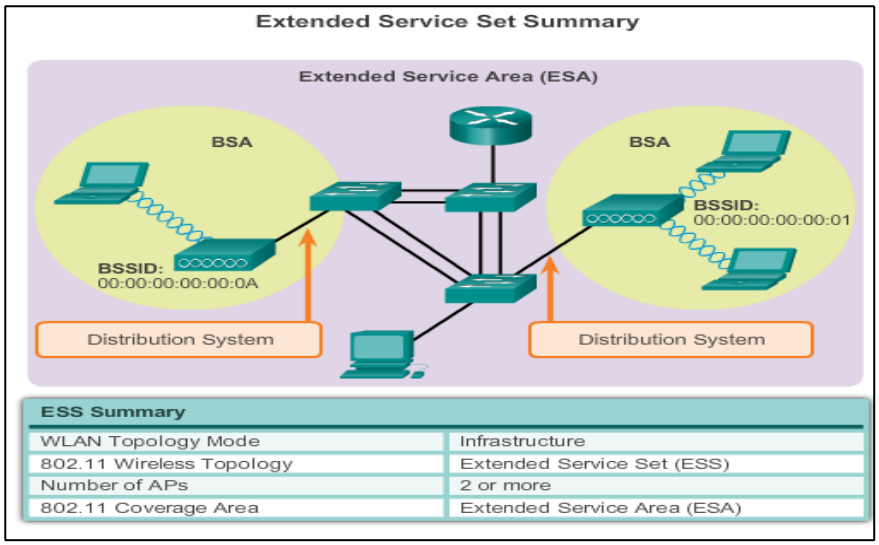
Infraestructura: desde un punto

de acceso.



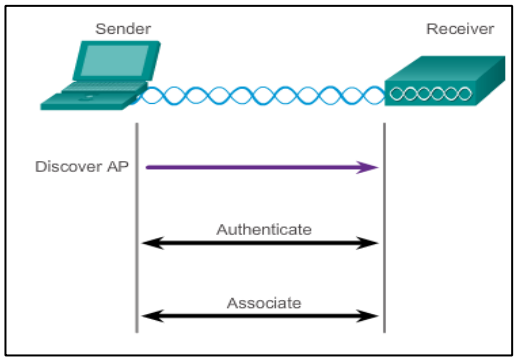


Sistema de Distribución: mediante cable vamos de cable a wifi.



Three-Stage Process: Para acceder de wifi a cable hacen falta tres pasos:

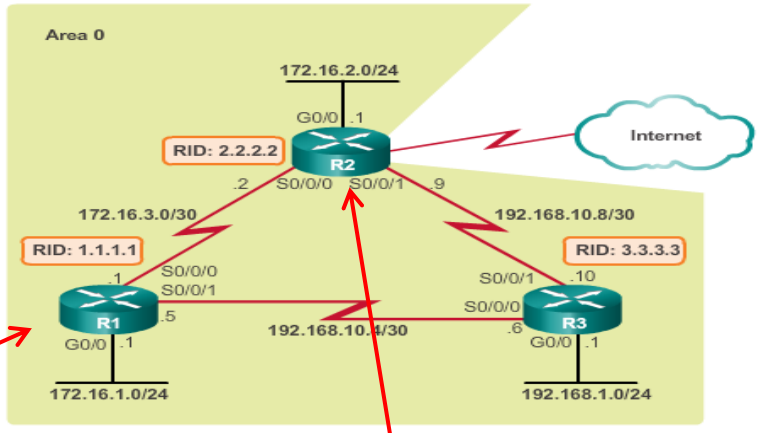
- **Discover**
- **Identicarse**
- **Asociarse**



En general este capítulo es literatura: ver diapositivas.

Capítulo 5: OSPF

En OSPF si se modifica el ancho de banda en un router (Bandwidth), habrá que hacerlo en todos. La **configuración en IPV4** es la siguiente:



R1

```
R1(config)# interface GigabitEthernet0/0
R1(config-if)# bandwidth 1000000
R1(config-if)# exit
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
R1(config-router)# auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent
across all routers.
R1(config-router)# network 172.16.1.0 0.0.0.255 area 0
R1(config-router)# network 172.16.3.0 0.0.0.3 area 0
R1(config-router)# network 192.168.10.4 0.0.0.3 area 0
R1(config-router)#
R1(config-router)# passive-interface g0/0
R1(config-router)#
```

R2

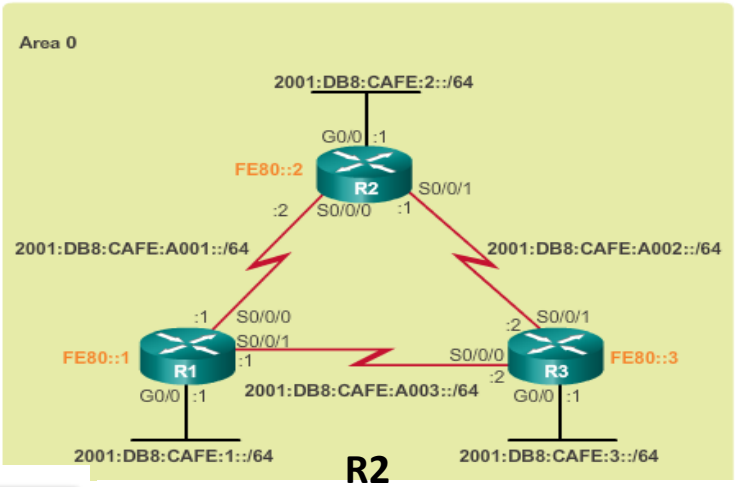
```
R2(config)# interface GigabitEthernet0/0
R2(config-if)# bandwidth 1000000
R2(config-if)# exit
R2(config)# router ospf 10
R2(config-router)# router-id 2.2.2.2
R2(config-router)# auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent
across all routers.
R2(config-router)# network 172.16.2.1 0.0.0.0 area 0
R2(config-router)# network 172.16.3.2 0.0.0.0 area 0
R2(config-router)# network 192.168.10.9 0.0.0.0 area 0
R2(config-router)#
R2(config-router)# passive-interface g0/0
R2(config-router)#
```

Los comandos SHOW son muy importantes:

- Show ip ospf neighbor
- Show ip protocols
- Show ip ospf
- Show ip ospf interface
- Show ip ospf interface brief

La configuración de OSPF en IPV6:

Al configurar en IPV6, para declarar las redes, en vez de meterse desde R(Config)# y poner todas las redes, nos metemos en cada interface del router y le decimos a qué área pertenece.



R1

```
R1(config)# ipv6 router ospf 10
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)# auto-cost reference-bandwidth 1000
% OSPFv3-10-IPv6:  Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent across all
routers.
R1(config-rtr)#
R1(config-rtr)# interface GigabitEthernet 0/0
R1(config-if)# bandwidth 1000000
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/1
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)# end
R1#
```

R2

```
R2(config)# ipv6 router ospf 10
R2(config-rtr)# router-id 2.2.2.2
R2(config-rtr)# auto-cost reference-bandwidth 1000
% OSPFv3-10-IPv6:  Reference bandwidth is changed.
        Please ensure reference bandwidth is consistent across all
routers.
R2(config-rtr)#
R2(config-rtr)# interface GigabitEthernet 0/0
R2(config-if)# bandwidth 1000000
R2(config-if)# ipv6 ospf 10 area 0
R2(config-if)#
R2(config-if)# interface Serial0/0/0
R2(config-if)# ipv6 ospf 10 area 0
R2(config-if)#
R2(config-if)# interface Serial0/0/1
R2(config-if)# ipv6 ospf 10 area 0
R2(config-if)# end
R2#
```

Tipos de Redes OSPF:

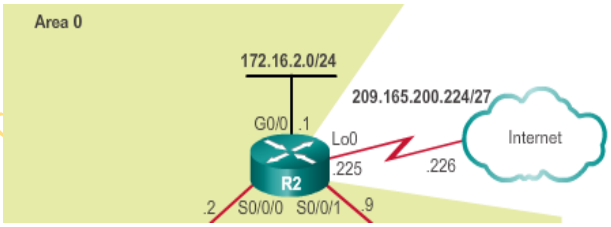
- **Punto a punto**
- **Broadcast Multiacceso**
- **No broadcast Multiacceso (NBMA)**
- **Punto a Multipunto**
- **Virtual Links**

Para **configurar la prioridad** nos metemos en la interface y le ponemos el comando:

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip ospf priority 255
R1(config-if)# end
R1#
```

Una interface Serial tiene por defecto una prioridad de 0 y una Fast Ethernet de 1.

Ruta por defecto en OSPF: (Redistribución de ruta; así se llama a la ruta por defecto en OSPF y se representa con O*E2)




```
R2(config)# ip route 0.0.0.0 0.0.0.0 209.165.200.226/lo0
R2(config)#
R2(config)# router ospf 10
R2(config-router)# default-information originate
R2(config-router)# end
R2#
```

O*E2 no acumula costo. OE1 acumula el costo de 1 + el valor de caa segmento: Si es OSPF 10, el costo que acumulará será 11.

El comando **default-information originate** se pone cuando se ha creado la ruta estática por defecto.

En IPV6 la ruta por defecto se configura:

```
R2(config)# ipv6 route ::/0 2001:DB8:FEED:1::2 Global link
R2(config)#
R2(config)# ipv6 router ospf 10
R2(config-rtr)# default-information originate
R2(config-rtr)# end
R2#
*Apr 10 11:36:21.995: %SYS-5-CONFIG_I: Configured from console by
console
R2#
```

Los comandos show una vez más son muy importantes:

- Show ipv6 route static
- Show ipv6 ospf interface s0/0/0

Modificación de los intervalos de tiempo para paquetes “Hellow” y “Dead”:

En IPV4:

```
R1(config)# interface serial 0/0/0
R1(config-if)# ip ospf hello-interval 5
R1(config-if)# ip ospf dead-interval 20
R1(config-if)# end
R1#
```

En IPV6:

```
R1(config)# interface serial 0/0/0
R1(config-if)# ipv6 ospf hello-interval 5
R1(config-if)# ipv6 ospf dead-interval 20
R1(config-if)# end
R1#
```

Los intervalos HELLO y DEAD deben ser IDENTICOS por interface

Seguridad en actualizaciones de OSPF:

- **Null** (sin autencación)
- **Simple password authentication**
- **Autenticación MD5**: por área o por interface.

Para configurar MD5:

- MD5 authentication can be enabled globally for all interfaces or on a per-interface basis.
- To enable OSPF MD5 authentication globally, configure:
  - **`ip ospf message-digest-key`** *key* **`md5`** *password* (interface configuration command)
  - **`area`** *area-id* **`authentication message-digest`** (router configuration command)
- To enable MD5 authentication on a per-interface basis, configure:
  - **`ip ospf message-digest-key`** *key* **`md5`** *password* (interface configuration command)
  - **`ip ospf authentication message-digest`** (interface configuration command)

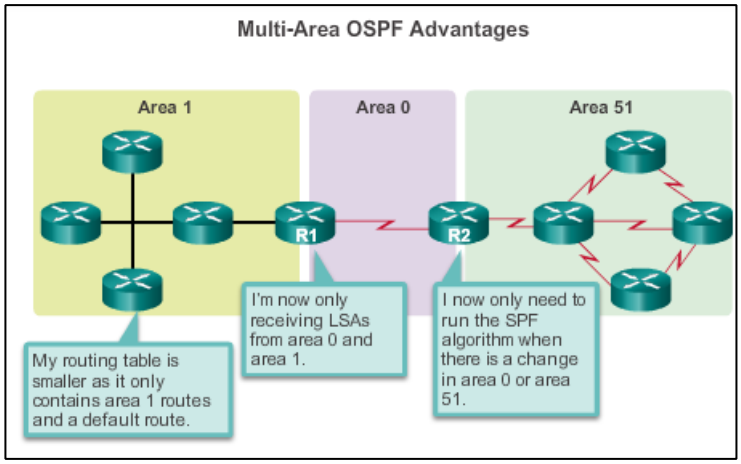
```
R1(config)# router ospf 10
R1(config-router)# area 0 authentication message-digest
R1(config-router)# exit
R1(config)#
*Apr  8 09:58:09.899: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2
on Serial0/0/0 from FULL to DOWN, Neighbor Down: Dead timer
expired
R1(config)#
*Apr  8 09:58:28.627: %OSPF-5-ADJCHG: Process 10, Nbr 3.3.3.3
on Serial0/0/1 from FULL to DOWN, Neighbor Down: Dead timer
expired
R1(config)#
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# exit
R1(config)#
R1(config)# interface Serial 0/0/0
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# exit
R1(config)#
R1(config)# interface Serial 0/0/1
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)#
```

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# ip ospf authentication message-digest
R1(config-if)# exit
R1(config)#
R1(config)# interface Serial 0/0/0
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# ip ospf authentication message-digest
R1(config-if)# exit
R1(config)#
R1(config)# interface Serial 0/0/1
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# ip ospf authentication message-digest
R1(config-if)# exit
```

Capítulo 6 OSPF MULTIAREA

Un área como mucho debe tener hasta 50 Routers. Más sería congestionar demasiado la red.

OSPF es un protocolo jerárquico. **Área de Backbone=Área 0=Área de Tránsito**



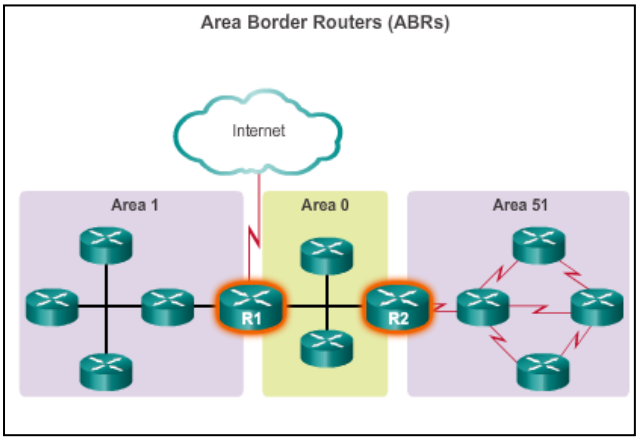
Tipos de router:

Routers internos: los que pertenecen a un área

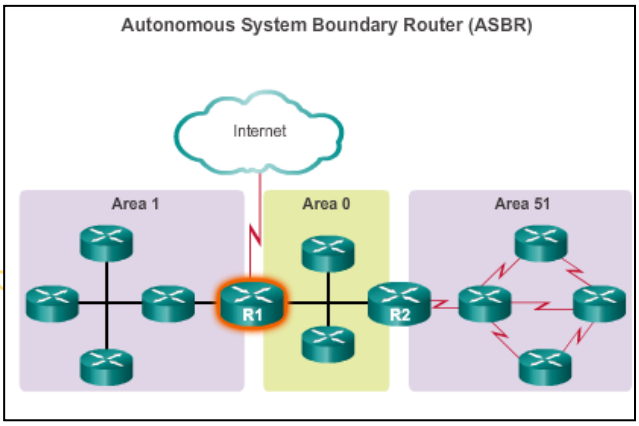
Routers de área: Los que perrtenecen a más de un área.

Routers de backbone: Los que pertenecen al área de backbone=área 0

Router ABR: Router de borde de área. Como mucho pueden pertenecer a 3 áreas. Como los routers deben tener la base de datos de todas las redes y routers directamente conectados, si fueran más de 3, su base de datos enloquecería. Deben ser los routers más potentes.



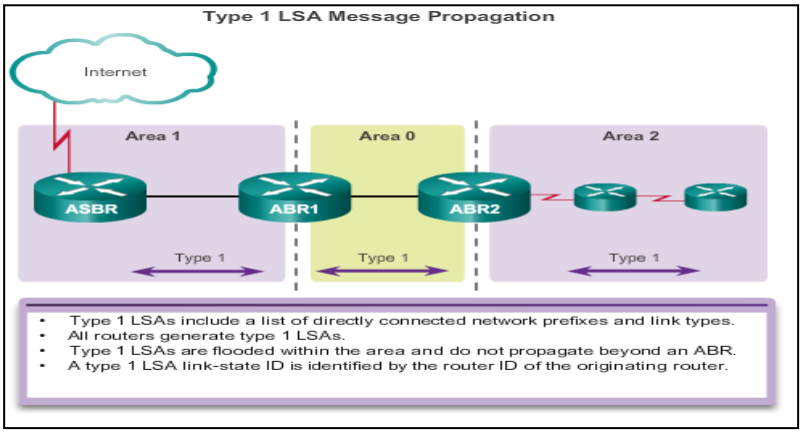
Routers ASBR: Son los routers que hacen de puerta de salida a otra área. Pueden albergar distintos protocolos de enrutamiento y distintos dominios de enrutamiento (3): Rutas estáticas, por defecto y de protocolo.



Tipos de LSA:

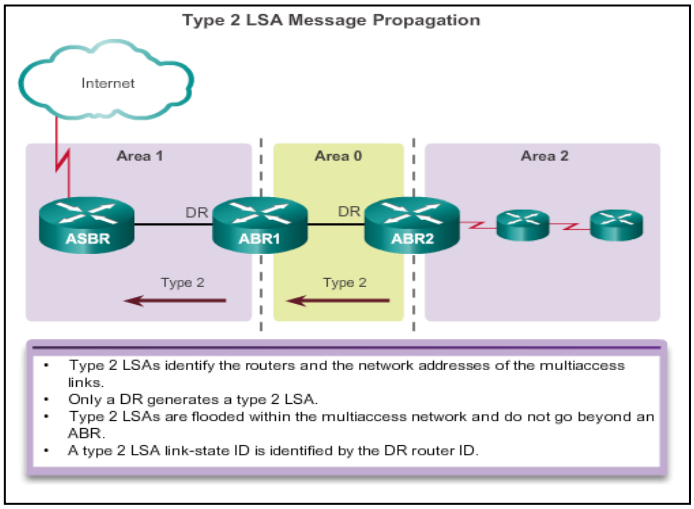
LSA tipo 1: dentro de un área. Cada área tiene su propia LSA de tipo 1.

Link State ID es un tipo de LSA identificado por el router ID. También llamado link tipo router.



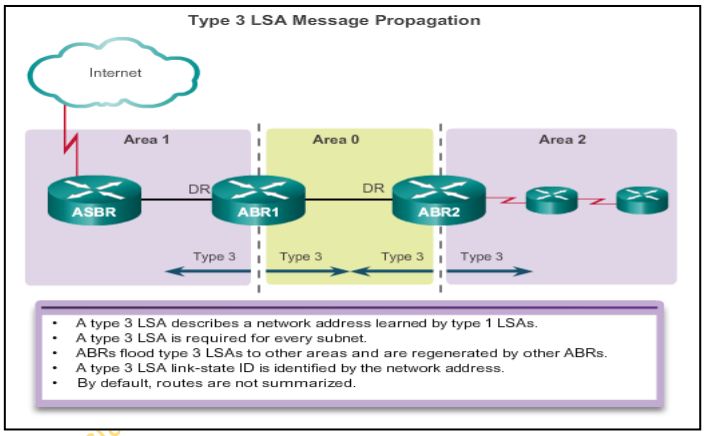
LSA de tipo 2: Router-ID DR

Link-state ID solo lo genera el DR. Solo hay LSA de tipo 2 en redes multiacceso, con SWs. También llamado link tipo broadcast.



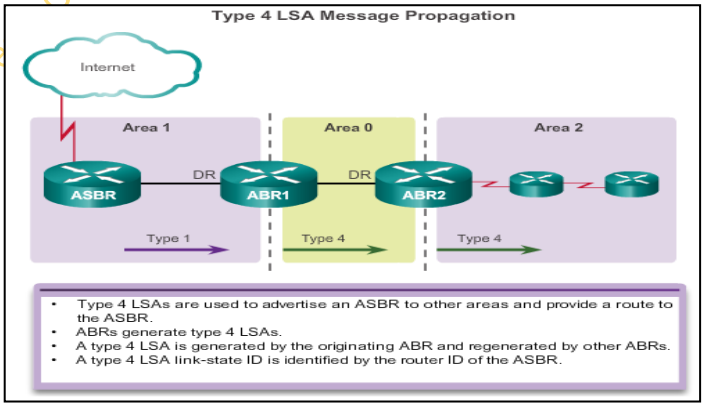
LSA de tipo 3: De red interna. Identificado por red.

Link-state ID: se retransmiten a todas las áreas. Los ABR pueden realizar sumarizaciones y los ASBR también. OSPF por defecto no hace sumarizaciones como RIP o EIGRP. Link llamado también tipo summary.



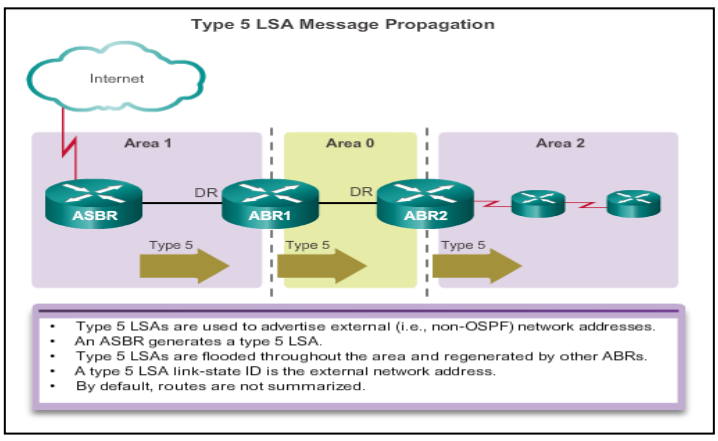
LSA tipo 4:

Link-state ID identificado por el router ASBR. Anuncia quien es el router ASBR del dominio OSPF.

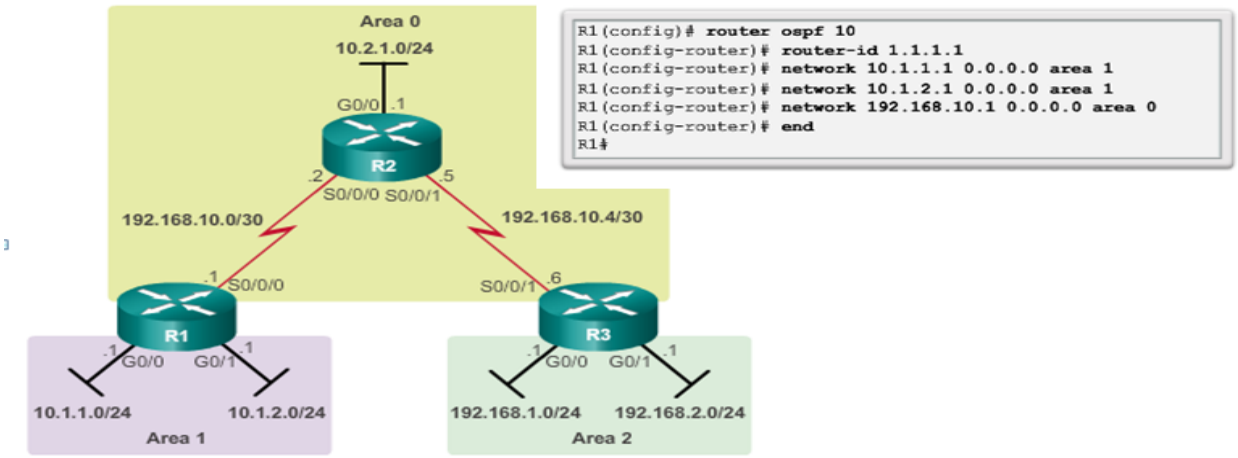


LSA de tipo 5: Para publicar acceso de red externas.

Link-state ID: red externa



Configuración:

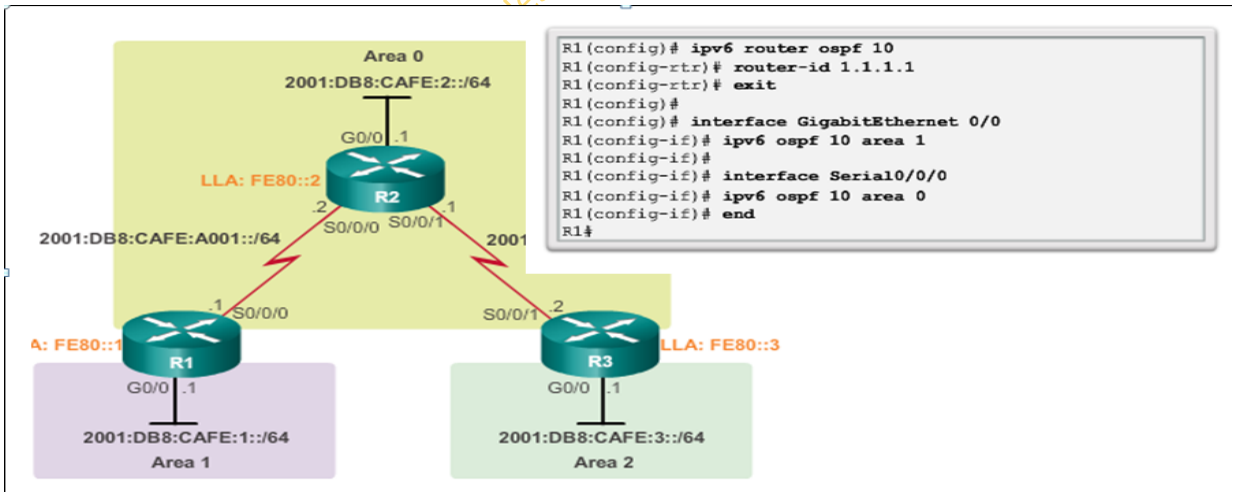


1º Definimos el proceso. En este caso es router ospf 10

2º Ponemos el router ID: En este caso 1.1.1.1

3º Declaramos las redes según áreas con: network + la red + wildcar + área

En ipv6



1º Habilitamos IPV6 con: ipv6 univast-routing

2º Habilitamos el proceso donde estamos con: ipv6 router ospf + valor del proceso

3º Metemos el router ID con router-id + ip

4º Nos metemos en cada interface del router y metemos:

4.1 Le decimos el proceso con ipv6 ospf + proceso + el área a la que pertenece esa interface.

Con show ip protocols vemos las rutas directamente conectadas

Con show ip route vemos las rutas que conoce el router

Con show ip ospf database vemos los link state id

Con show ip route vemos las redes y como las ha aprendido:

Router and Network Routing Table Entries

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

Gateway of last resort is 192.168.10.2 to network 0.0.0.0

O*E2 0.0.0.0/0 [110/1] via 192.168.10.2, 00:00:19, Serial0/0/0
     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.1.1.0/24 is directly connected, GigabitEthernet0/0
L       10.1.1.1/32 is directly connected, GigabitEthernet0/0
C       10.1.2.0/24 is directly connected, GigabitEthernet0/1
L       10.1.2.1/32 is directly connected, GigabitEthernet0/1
O       10.2.1.0/24 [110/648] via 192.168.10.2, 00:04:34, Serial0/0/0
O IA 192.168.1.0/24 [110/1295] via 192.168.10.2, 00:01:48,Serial0/0/0
O IA 192.168.2.0/24 [110/1295] via 192.168.10.2, 00:01:48,Serial0/0/0
     192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
C       192.168.10.0/30 is directly connected, Serial0/0/0
L       192.168.10.1/32 is directly connected, Serial0/0/0
O       192.168.10.4/30 [110/1294] via 192.168.10.2, 00:01:55,Serial0/0/0
R1#
```

O = Rutas dentro del área del router pero no directamente conectadas a él

OIA = Rutas dentro del área ospf y actualizadas

OE1 = Rutas externas a esa área con costo acumulativo

OE2 = Rutas externas a esa área con costo no acumulativo

O*E1= Rutas aprendidas externas y por defecto con costo acumulativo

O*E2= Rutas aprendidas externas y por defecto con costo no acumulativo+

OSPFv3 Routing Table Entries

```
R1# show ipv6 route
IPv6 Routing Table - default - 9 entries
Codes:C - Connected, L - Local, S - Static, U-Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND-ND Default,NDp-ND Prefix,DCE-Destination
       NDr - Redirect, O-OSPF Intra, OI-OSPF Inter, OE1-OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OE2  ::/0 [110/1], tag 10
     via FE80::2, Serial0/0/0
C    2001:DB8:CAFE:1::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L    2001:DB8:CAFE:1::1/128 [0/0]
     via GigabitEthernet0/0, receive
O    2001:DB8:CAFE:2::/64 [110/648]
     via FE80::2, Serial0/0/0
OI   2001:DB8:CAFE:3::/64 [110/1295]
     via FE80::2, Serial0/0/0
C    2001:DB8:CAFE:A001::/64 [0/0]
     via Serial0/0/0, directly connected
L    2001:DB8:CAFE:A001::1/128 [0/0]
     via Serial0/0/0, receive
O    2001:DB8:CAFE:A002::/64 [110/1294]
     via FE80::2, Serial0/0/0
L    FF00::/8 [0/0]
     via Null0, receive
R1#
```

Capítulo 7 EIGRP:

- El algoritmo que es usa es DUAL.
- Se calcula además un camino de respaldo.
- Las actualizaciones son limitadas y parciales; solo se actualiza cuando hay un cambio y solo a los routers afectados.
- No existe en la capa 4 de transporte TCP/UDP. Es RTP
- CIA: Confidencialidad, Integridad y Autenticación.
- En EIGRP hay 5 tipos de paquetes divididos en confiables y no confiables:
  Paquetes confiables: Update, Query y Reply
  Paquetes no confiables: Hello y Ack

PROTOCOLO EIGRP

Enhanced Internal Gateway Routing Protocol es una evolución de IGRP protocolo propio de Cisco mejorado. Solo los routers Cisco entienden este protocolo. Así que todos los routers que lo usen deben ser Cisco.

A diferencia de RIP que se basa en la métrica y distancia administrativa para tomar decisiones de enrutamiento, EIGRP se basa en mas parámetros:

1- Ancho de banda
2- Retraso
3- Confiabilidad
4- Carga


```
R1(Config)#router eigrp 1

R1(Config-route)#network + dirección ip de red
```

Por defecto son el ancho de banda y el retraso.

Los procesos de los mensajes en EIGRP son:

1- Encabezamiento de trama.
2- Encabezado de paquete IP.
3- Encabezado de paquete EIGRP.

Tipo/longitud/tipos de valor.

En rip solo eran origen y destino.

El encabezado del paquete EIGRP contiene:

- Campo de código de operación
- Nº de sistema autónomo (AS)

El protocolo EIGRP solo toma por defecto los dos primeros procesos, el ancho de banda y el retraso. Los otros factores (confiabilidad y carga) hay que decírselos manualmente.

Con EIGRP se puede ir a más de 15 saltos. No tiene límite. Además, EIGRP tiene su propio protocolo de transporte (RTP)=protocolo de transporte confiable. Ya sean protocolos IP,IPX o Apple talk, el RTP será diferente. A esto se le llama PDMs distintas.

EIGRP crea (además de la tabla de enrutamiento) dos estructuras de datos nuevas:

- Tabla de tipología
- Tabla de vecinos

La dirección multicast es 224.0.0.10

Hay 5 tipos distintos de paquetes que se pueden enviar entre dos routers EIGRP.

- Paquetes de saludo: detectan vecinos y forma adyacencias con ellos
- Paquetes de actualización: se usan para difundir la información de enrutamiento.
- Paquetes de reconocimiento: se usan para reconocer la recepción de los paquetes de actualización, consulta y respuesta.
- Paquetes de consulta: pueden usar unicast o multicast
- Paquetes de respuesta: Usan

} **DUAL**: usa los paquetes para la búsqueda de redes y evitar bucles de enrutamiento. t

Función del protocolo de saludo: Detecta routers vecinos y establece adyacencias. Manda paquetes de aludo cada 5 segundos y si después de 3 intentos sin respuesta de la interface del router de destino la dará por muerta. Este valor se puede cambiar. Se llama tiempo de HOLD.

Actualizaciones limitadas de EIGRP. EIGRP no envía actualizaciones cada 30 segundos como hace rip, sino que solo envía actualizaciones cuando hay un cambio en el estado de la ruta. Estas actualizaciones las hace por multicast con la ip 224.0.0.10.

Actualizaciones parciales: Envía solo información de la ruta que se ha modificado, no de todo.

Actualizaciones limitadas: Cuando una ruta se modifica, solo se notifica a los dispositivos afectados

Algoritmo de actualización difusa (DUAL). Es el método principal de EIGRP para evitar los bucles de enrutamiento.

Distancia administrativa: se define como la confiabilidad de la ruta de origen. Las distancias administrativas por defecto que tiene EIGRP son:

- Rutas sumarizadas: 5
- Rutas internas: 90
- Rutas importadas: 170

EIGRP es más fiable que rip.

Autenticación: se puede cifrar

Como EIGRP sumariza automáticamente igual que RIP, resumirá automáticamente las rutas en los limites classfull. Ojo si hacemos eigrp con subredes. Poner el no auto-summary

Packet Type	Description
Hello	Used to discover other EIGRP routers in the network.
Acknowledgement	Used to acknowledge the receipt of any EIGRP packet.
Update	Convey routing information to known destinations.
Query	Used to request specific information from a neighbor router.
Reply	Used to respond to a query.

Configuración EIGRP:

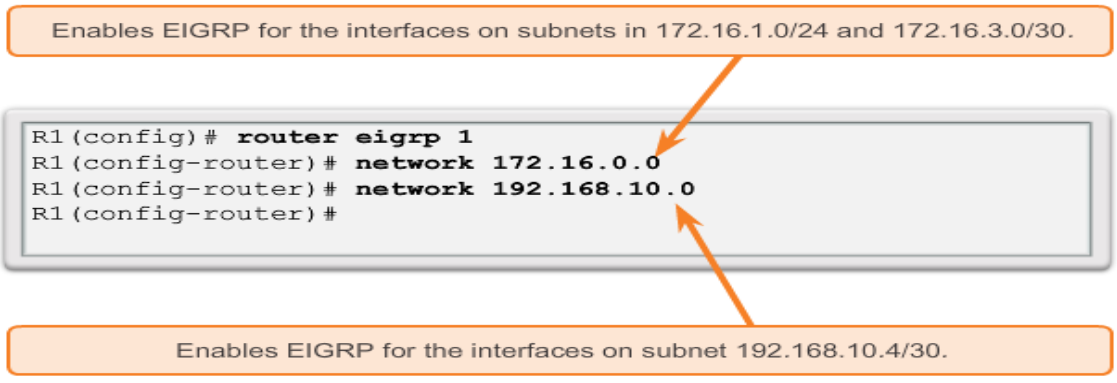
1º Habilitamos EIGRP: R1(Config)#router eigrp 1 (hay que poner un valor decimal del 1 al 65535)

A todos los routers se le otorgara el mismo valor decimal dentro del mismo dominio de enrutamiento.

En EIGRP también se usa el comando network por cada red directamente conectada.

```
R1(config)#router eigrp 1
R1(config-router)#
```

2º Declaramos las redes: R(Config-router)#network + la red

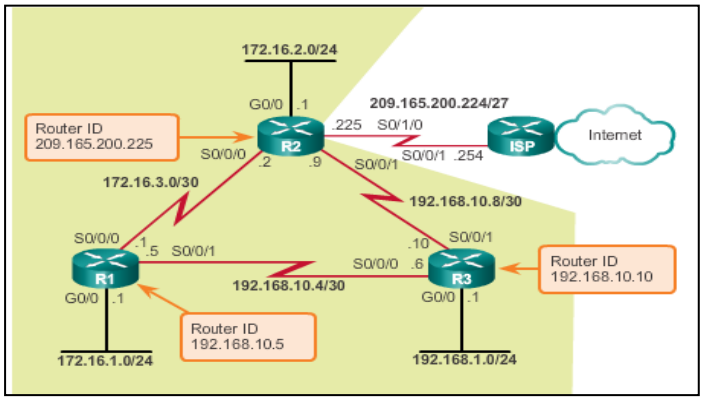


3º Ponemos la passive-interface a la red que corresponda

4º Ponemos la no-autosummary

En EIGRP no existen áreas. No hace falta tampoco poner máscara, pero sí aconsejable (la wildcar). Si no ponemos máscara, tomará por defecto la de clase.

EIGRP permite sumarizar. En el ejemplo de la derecha, para declarar las redes 172.16.3.0/30 y la red 172.16.2.0/24 (de R2), bastaría con declarar una red sumarizada que en este caso sería: 172.16.2.0/23 o la 172.16.0.0



También se configura en EIGRP las **passive-interface**:

```
Router(config)# router eigrp as-number
Router(config-router)# passive-interface interface-type
interface-number
```

OSPF, RIP y EIGRP tienen balanceo de carga de hasta 4 caminos.

Con **show interface +** la interface correspondiente, nos muestra el ancho de banda (BW) y el retardo (DLY).

Para cambiar la métrica usamos el comando bandwidth + valor

```
R1(config)# interface s 0/0/0
R1(config-if)# bandwidth 64
```

Concepto DUAL:

Sucesor: La mejor ruta con menos coste. Es la ruta que tiene mejor métrica.
Distancia factible: Es la métrica calculada más baja a lo largo de una ruta a la red de destino. (la mejor métrica, vaya, la métrica del sucesor)

Sucesor factible: Es una ruta de respaldo sin bucles al mismo destino que la ruta del sucesor. (es la ruta alternativa). No siempre se calcula una ruta suplente automáticamente.

*El factor vector distancia no funciona por topología, sino por la información que recibe un router de los dispositivos directamente conectados y la aumenta un salto a la ruta.

*Converger: Cuando todos los routers conocen a todos=tiempo de convergencia

*Bucle de enrutamiento: Cuando un paquete no llega a su destino y se queda dando vueltas entre routers. Cuando pasa esto, entra en juego el TTL (time to live) y se autodestruye.

En una red LAN (Ethernet) la resolución se hace en capa 2. No son paquetes ip. El spanning tree evita los bucles que hacen que se pueda caer la red entera.

Distancia reportada /notificada: El valor de métrica que un router dice a un arouter vecino con respecto a una red.

Para que haya un sucesor factible se debe dar la condición de factibilidad: EIGRP hará un sucesor factible solo cuando la distancia reportada de un vecino es menor que la distancia factible(FD) del router local en la misma red de destino. Esta condición no siempre se cumple, por lo que no siempre hay sucesor factible.

EIGRP además de la tabla de enrutamiento tiene la tabla de vecinos y la tabla de topología.

Para ver la tabla de tipología se usa el comando show ip eigrp topology. Esta tabla tendrá todas las rutas del sucesor y del sucesor factible.

Máquina de estado finito: máquina en la que se apoya EIGRP para tomar decisiones.

Por defecto EIGRP sumariza, al igual que rip 2. Para que no lo haga, hay que poner también el comando no auto-summary.

Diferencias entre EIGRP en IPV4 y IPV6: es prácticamente igual. Cambia la autenticación que en ipv6 es siempre con MD5

	EIGRP for IPv4	EIGRP for IPv6
Advertised routes	IPv4 networks	IPv6 prefixes
Distance vector	Yes	Yes
Convergence technology	DUAL	DUAL
Metric	Bandwidth and delay by default, reliability and load are optional	Bandwidth and delay by default, reliability and load are optional
Transport protocol	RTP	RTP
Update messages	Incremental, partial and bounded updates	Incremental, partial and bounded updates
Neighbor discovery	Hello packets	Hello packets
Source and destination addresses	IPv4 source address and 224.0.0.10 IPv4 multicast destination address	IPv6 link-local source address and FF02::10 IPv6 multicast destination address
Authentication	Plain text and MD5	MD5
Router ID	32-bit router ID	32-bit router ID

Configuración de EIGRP en IPV6:

1º Habilitamos ipv6 en el router con: R(Config)#ipv6 unicast-routing

2º Habilitamos EIGRP para ipv6 con: R(Config)#ipv6 router eigrp + valor del proceso (siempre usaremos el mismo**)**

3º Asignamos un router-ID a cada router con: R(Config-rtr)#eigrp router-id + dirección (que será la que queramos y de ipv4 a pesar de estar en ipv6**)**

4º Hacemos un No shutdown ya que EIGRP necesita hacer un no shutdown por cada proceso.

```
R2(config)#ipv6 unicast-routing
R2(config)#ipv6 router eigrp 2
R2(config-rtr)#eigrp router-id 2.0.0.0
R2(config-rtr)#no shutdown
R2(config-rtr)#
```

5º Ir a cada interface y publicar el proceso que hemos creado

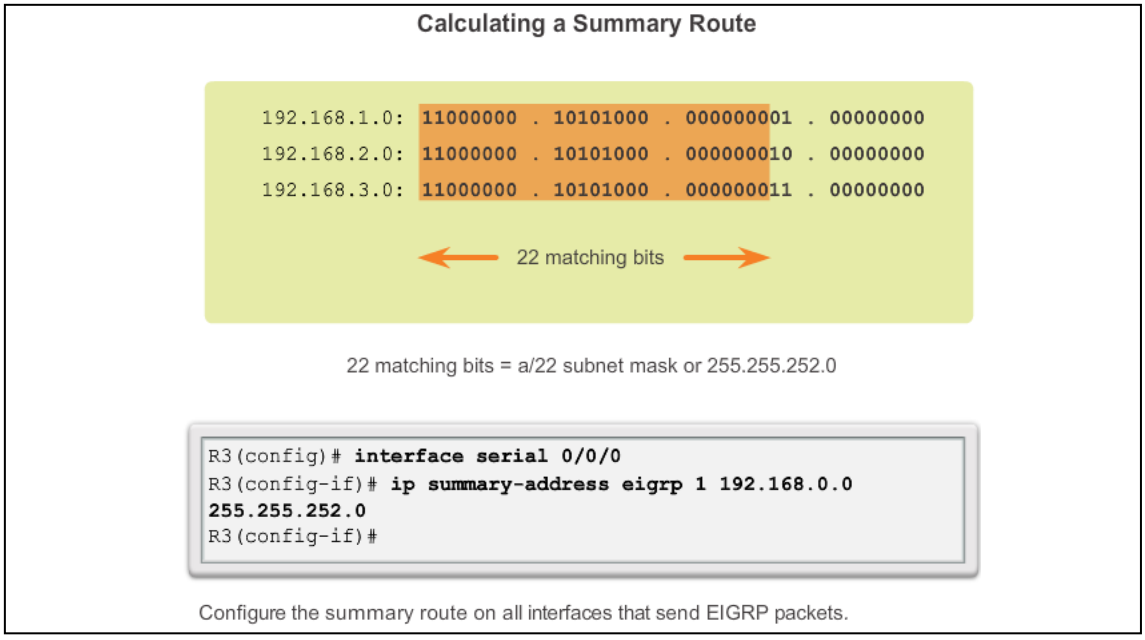
```
R1(config)#interface g0/0
R1(config-if)#ipv6 eigrp 2
R1(config-if)#exit
R1(config)#interface s 0/0/0
R1(config-if)#ipv6 eigrp 2
R1(config-if)#exit
R1(config)#interface s 0/0/1
R1(config-if)#ipv6 eigrp 2
R1(config-if)#
```

```
R2(config)#interface g 0/0
R2(config-if)#ipv6 eigrp 2
R2(config-if)#exit
R2(config)#interface s 0/0/0
R2(config-if)#ipv6 eigrp 2
R2(config-if)#exit
%DUAL-5-NBRCHANGE: EIGRP-IPv6 2: Neighbor FE80::1
(Serial0/0/0) is up: new adjacency
R2(config)#interface s 0/0/1
R2(config-if)#ipv6 eigrp 2
R2(config-if)#
```

Caso de caerse una red para volver a calcular los valores y ejecutar de nuevo el algoritmo dual y crear la nueva tabla topológica, usaremos el comando **R#debug eigrp fsm** así lo actualizamos. Una vez reconstruida la tabla topológica es aconsejable volver a desactivar este comando con **R#undebug all**

Capítulo 8: EIGRP TroubleShooting

Sumarización en EIGRP:



Redistribución de ruta estática por defecto en ipv4

```
R2(config)# ip route 0.0.0.0 0.0.0.0 serial 0/1/0
R2(config)# router eigrp 1
R2(config-router)# redistribute static
```

Redistribución de ruta estática por defecto en ipv6

```
R2(config)# ipv6 route ::/0 serial 0/1/0
R2(config)# ipv6 router eigrp 2
R2(config-router)# redistribute static
```

Configuración de paquetes Hello y Hold time Para cambiar el tiempo de hello, comando:

R(Config-if)#ip hello-interval eigrp as-number seconds

R(Config-if)#ip hello-interval eigrp 1 60

Ojo! Al cambiar el tiempo de hello, también habrá que cambiar el tiempo de hold. Lo ideal es poner el triple de tiempo de hello) *estos valores no se suelen cambiar

R(Config-if)#ip hold-time eigrp 1 180

Configuring EIGRP for IPv4 Hello and Hold Timers

```
R1(config)# interface serial 0/0/0
R1(config-if)# ip hello-interval eigrp 1 60
R1(config-if)# ip hold-time eigrp 1 180
```

Default Hello Intervals and Hold Times for EIGRP

Bandwidth	Example Link	Default Hello Interval	Default Hold Time
1.544 Mbps	Multipoint Frame Relay	60 seconds	180 seconds
Greater than 1.544 Mbps	T1, Ethernet	5 seconds	15 seconds

EIGRP además de la tabla de enrutamiento tiene la tabla de vecinos y la tabla de topología.

Para ver la tabla de tipología se usa el comando show ip eigrp topology. Esta tabla tendrá todas las rutas del sucesor y del sucesor factible.

Máquina de estado finito: máquina en la que se apoya EIGRP para tomar decisiones.

Por defecto EIGRP sumariza, al igual que rip 2. Para que no lo haga, hay que poner también el comando no auto-summary.

También se puede hacer una **autosumarización (hacer una superred) manual con el comando:**

Router(Config-if)#ip summary-address eigrp as-number(1) network-address subnet-mask

Ojo al hacer una superred. Habrá que cambiar la máscara. Ejemplo: si las redes 192.168.1.0 192.168.2.0 y 192.168.30 las quisiéramos agrupar en una superred, seria:

192.168.0.0 y con máscara 255.255.252.0 Ese 252 es por que comparten todos sus bits hasta el segundo octeto pero en el 3º al haber agrupado hasta el valor 3 para el cual harían falta dos bits ½ 1/1 1+2=3 si a 255 le restamos 3=252

Por defecto EIGRP usa solo el 50% del **ancho de banda** para intercambiar información. Este valor porcentual se puede cambiar con el comando:

Router(Config-if)#ip bandwidth-percent eigrp as-number percent

R(Config)#interface serial 0/0/0 (mascara/placa/puerto) esto es solo un ejemplo

R(Config-if)#bandwidth 64

R(Config-if)ip bandwidth-percent eigrp 1 50

Balanceo de carga:

Equilibrio de carga de igual costo es la capacidad de un router para distribuir el tráfico saliente mediante todas las interfaces que tienen la misma métrica de la dirección de destino. El Cisco IOS, por omisión, se permite el balanceo de carga utilizando un máximo de cuatro rutas de igual costo; Sin embargo, esto puede ser modificado. Con el comando de modo de configuración maximum-paths router, hasta 32 rutas de igual costo se pueden mantener en la tabla de enrutamiento.

```
Router(config-router)# maximum-paths valor
```

* Si el valor se establece en 1, el equilibrio de carga está desactivado.

Configuración de autenticación:

Step 1: Create a Keychain

```
Router(config)# key chain name-of-chain
Router(config-keychain)# key key-id
Router(config-keychain-key)# key-string key-string-text
```

Step 2: Configure EIGRP Authentication Using Keychain and Key

```
Router(config)# interface type number
Router(config-if)# ip authentication mode eigrp as-number md5
Router(config-if)# ip authentication key-chain eigrp as-number
name-of-chain
```

… y en IPV6:

```
R1(config)# key chain EIGRP_KEY
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string cisco123
R1(config-keychain-key)# exit
R1(config-keychain)# exit
R1(config)# interface serial 0/0/0
R1(config-if)# ip authentication mode eigrp 1 md5
R1(config-if)# ip authentication key-chain eigrp 1 EIGRP_KEY
R1(config-if)# exit
R1(config)# interface serial 0/0/1
R1(config-if)# ip authentication mode eigrp 1 md5
R1(config-if)# ip authentication key-chain eigrp 1 EIGRP_KEY
R1(config-if)# end
```

```
R1(config)# key chain EIGRP_IPV6_KEY
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string cisco123
R1(config-keychain-key)# exit
R1(config-keychain)# exit
R1(config)# interface serial 0/0/0
R1(config-if)# ipv6 authentication mode eigrp 2 md5
R1(config-if)# ipv6 authentication key-chain eigrp 2
                EIGRP_IPV6_KEY
R1(config-if)# exit
R1(config)# interface serial 0/0/1
R1(config-if)# ipv6 authentication mode eigrp 2 md5
R1(config-if)# ipv6 authentication key-chain eigrp 2
                EIGRP_IPV6_KEY
```

Tema VI: Conecting Networks

Capítulo 1: Redes jerárquicas

Además de NUCLEO, DISTRIBUCIÓN y ACCESO, puede haber otras secciones: Módulos

SAN: Storage Area Network

Servidores Virtuales: Hyper-V, Vmware ESXi

Servidores de balanceo de carga

Firewalls

IDS detección de intrusos/IPS prevención de intrusos

- **Access layer** – Provides workgroup or user access to the network.
- **Distribution layer** – Provides policy-based connectivity.
- **Core layer** – Provides fast transport between distribution switches.

Tipos de tráfico:

7 -Tráfico de red

5- Tráfico de voz

4-Videoconferencia

3-Datos críticos

2-Tráfico de usuario

1-Lo mejor posible

0-Tráfico carroñero

VSS: Sistema de SW Virtual (tipo Etherchannel). Convierte dos SWs en uno solo

El enlace se llama VSL

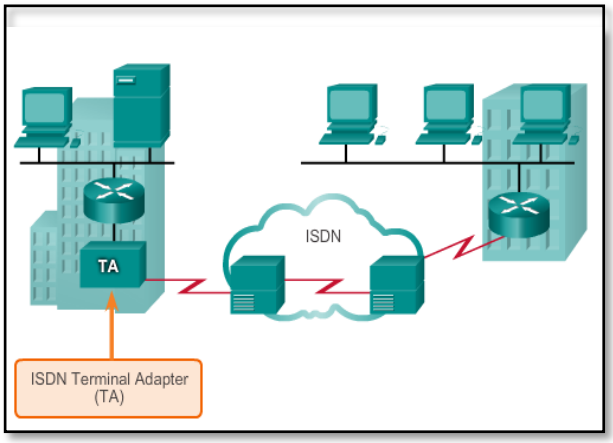
Capítulo 2: Conectando a red WAN

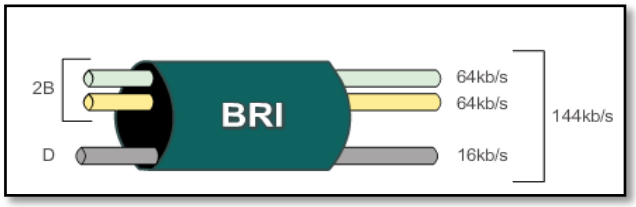
Tecnología de conmutación de circuitos:

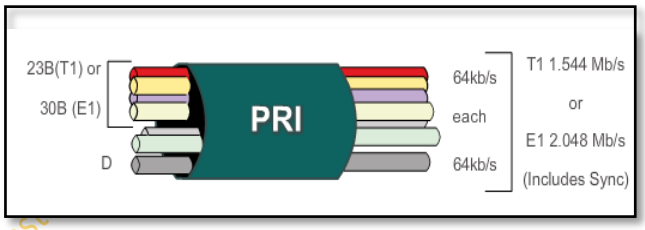
- RDSI (Protocolo Q931)
- PSTN

RDSI=ISDN

CSU/DSU: Equipo conversor de medio LAN a medio WAN. Sería también el DCE (Clock rate)





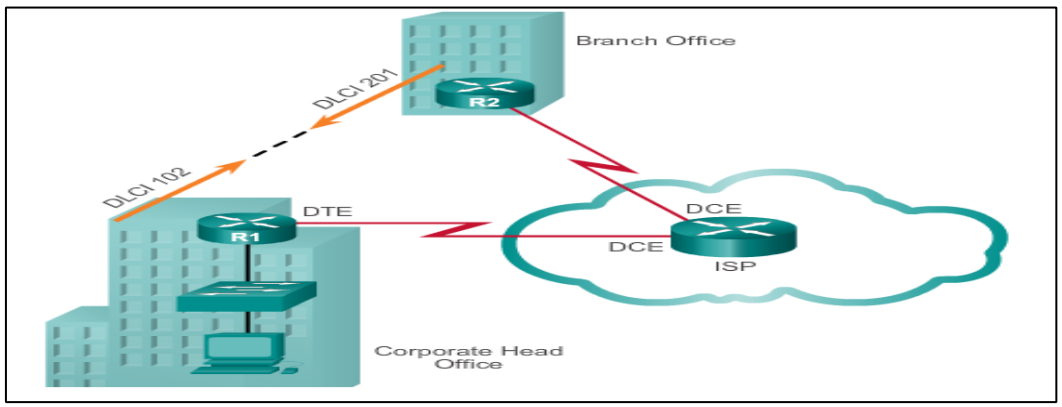


RDSI:

23 Canales B para T1 (1544 Mb/s)

30 Canales B para E1 (2048 Mb/s)

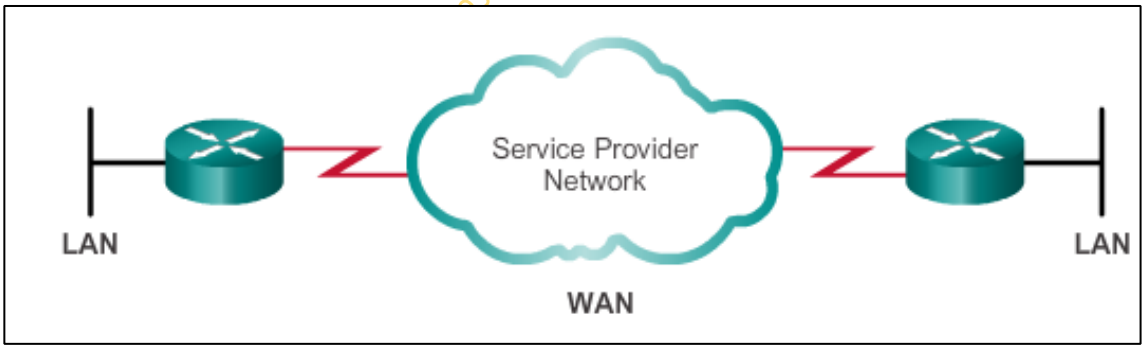
1 Canal D (Señalización)

- En un SW Frame Relay todos sus puntos son DCE (Clock Rate)
- PVC (Circuito Virtual Permanente) (Frame Relay)
- DLCI es el identificador de Frame Relay
- 

- ATM es más rápido y confiable que Frame Relay
- ATM sí que puede ofrecer QuOs, y Frame Relay No (en la WAN)
- SVC: Circuito Virtual Conmutado (casi no se utiliza ya)
- Ethernet WAN (Metro Ethernet)
    - Es MacroLan para Telefónica
- SW 3750 ME de Cisco son SWs especiales de capa 3
- ATOM: Any Transport Over MPLS
- EOMPLS: Ethernet Over MPLS
- (MPLS VPN L2; MPLS VPN L3)
- Asociar término DSLAM con DSL
- Las VPNs ofrecen seguridad gracias a IPsec
- VPN como los PVC son permanentes

Capitulo 3: Conexión Punto a Punto

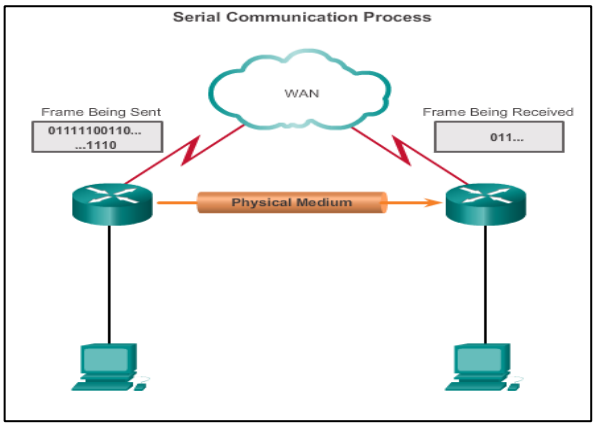
Las conexiones PPP unen 2 Routers : De LAN a LAN pasando por WAN entre 2 equipos distantes.



Son por enlaces seriales o por conexiones dedicadas.

Comunicación Serial: Para que haya PPP tienen que estar habilitados el PPP en ambos sitios. Una comunicación Serial correcta une LAN con WAN.

Hay 3 estándares de comunicación serial: RS-232, V35 y HSSi para fibra.



Enlaces Punto a Punto: Es una línea dedicada de Lan Lan pasando por Wan.

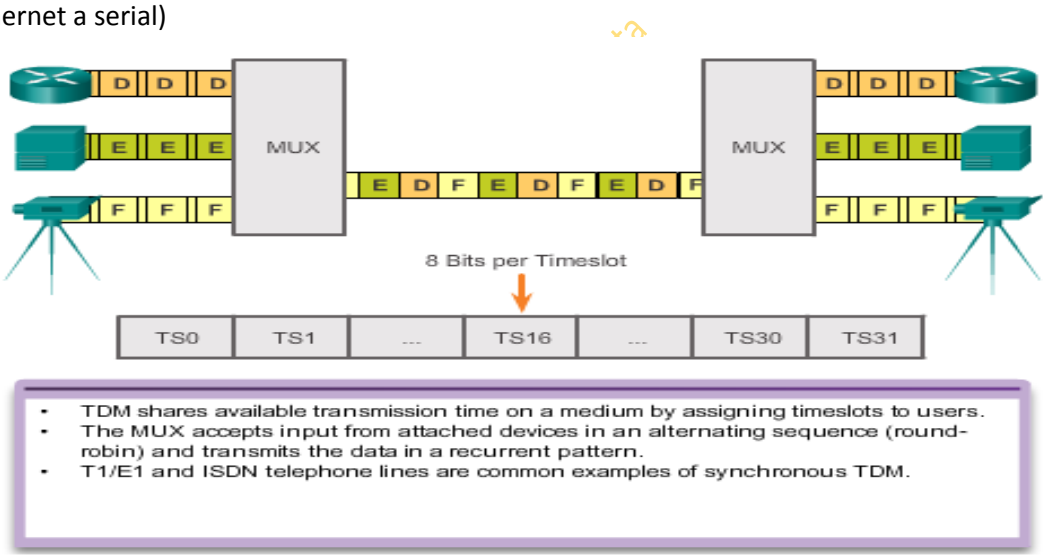
El CSU/DSU es el conversor de medios de ethernet a serial: de RJ45 a Smart o a V35, o a Winchester.

El Clock Rate del serial lo marca el proveedor de servicio.

Tanto cliente como proveedor de servicio tienen un CSU/CDU (Conversor de medio).

En la Wan se tiene una tasa de transferencia constante, según lo que paguemos. Una conexión Punto a Punto siempre es más cara.

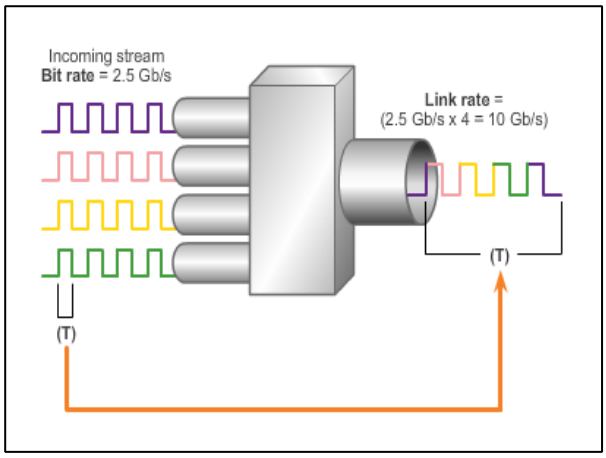
TDM: Time Division Multiplexing es la forma, la tecnología de convertir de un medio a otro (ethernet a serial)



STDM es una tecnología como TDM pero más eficiente.

El TDM para fibra óptica suele ser mediante **SDH**=Red Digital Síncrona y también suele ser por **SONET**=Red Optica Síncrona.

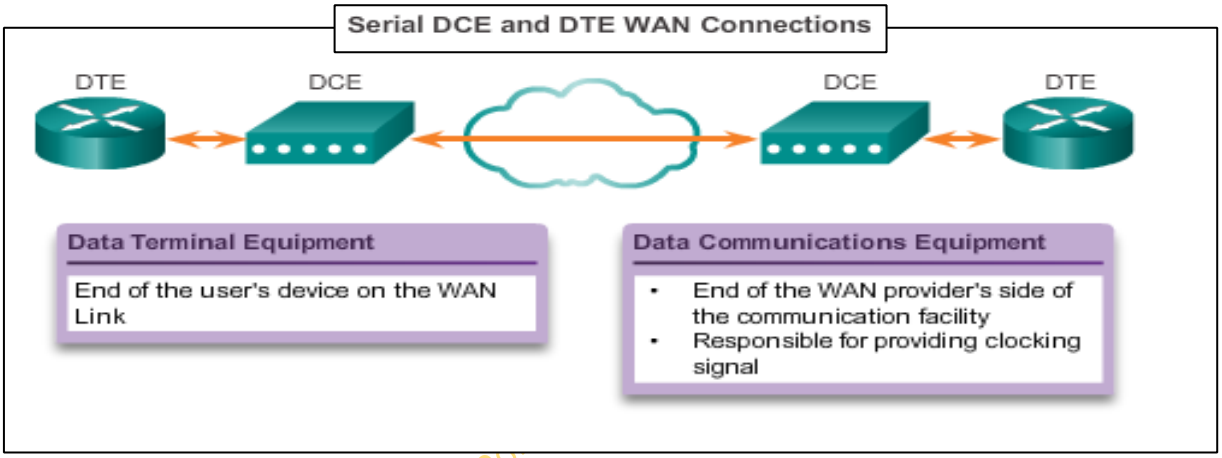
En estos dos casos, se le añade a las tramas información de control.



Punto de Demarcación: Es el punto que separa la responsabilidad del cliente y empieza la del proveedor de servicio y al revés. Lo normal es que el punto de demarcación sea el CSU/DSU.

DTE Vs DCE:

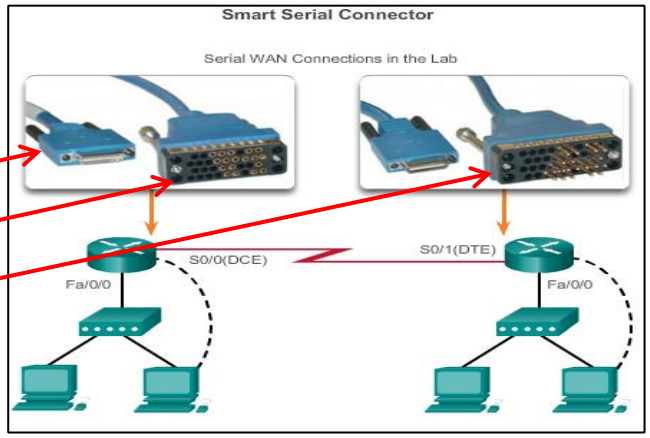
DTE - Comúnmente CPE, generalmente un router, también podría ser un terminal, ordenador, impresora o máquina de fax si se conectan directamente a la red de proveedores de servicios. DCE - Comúnmente un módem o CSU / DSU, es un dispositivo que se utiliza para convertir los datos del usuario del DTE a una forma aceptable para el servicio de enlace de transmisión proveedor de WAN. La señal se recibe en el mando a distancia DCE, que decodifica la señal de nuevo en una secuencia de bits; DCE distante entonces señala esta secuencia al DTE remoto.



Cables seriales:

Smart: es el azul pequeño que se conecta directamente en el Router.

Winchester: Es el grande que se conecta al CSU/DTU: La hembra es el del clock rate (DCE) y el macho es para el DTE.



Anchos de banda permitidos según el tipo de cable en enlaces dedicados:

*Cuadro de la derecha

Encapsulación de datos: Sin la encapsulación de los datos la transmisión no se podría realizar.

Carrier Transmission Rates

Line Type	Bit Rate Capacity
56	56 kb/s
64	64 kb/s
T1	1.544 Mb/s
E1	2.048 Mb/s
J1	2.048 Mb/s
E3	34.064 Mb/s
T3	44.736 Mb/s
OC-1	51.84 Mb/s
OC-3	155.54 Mb/s
OC-9	466.56 Mb/s
OC-12	622.08 Mb/s
OC-18	933.12 Mb/s
OC-24	1.244 Gb/s
OC-36	1.866 Gb/s
OC-48	2.488 Gb/s
OC-96	4.976 Gb/s
OC-192	9.954 Gb/s
OC-768	39.813 Gb/s

Encapsulación HDLC:

```
Router(config)# interface s0/0/0
Router(config-if)# encapsulation hdlc
```

- Enable HDLC encapsulation
- HDLC is the default encapsulation on synchronous serial interfaces

Para saber qué interface serial tiene el clock rate, además de con un show running, se haría con: **show controllers serial + interface**

La encapsulación de PPP (que es un estandar) **se basa en HDLC**

LCP configura el enlace y lo comprueba.

NCP: Negocia el tipo de protocolo.

HDLC solo encapsula, pero no da seguridad ni compresión a las tramas.

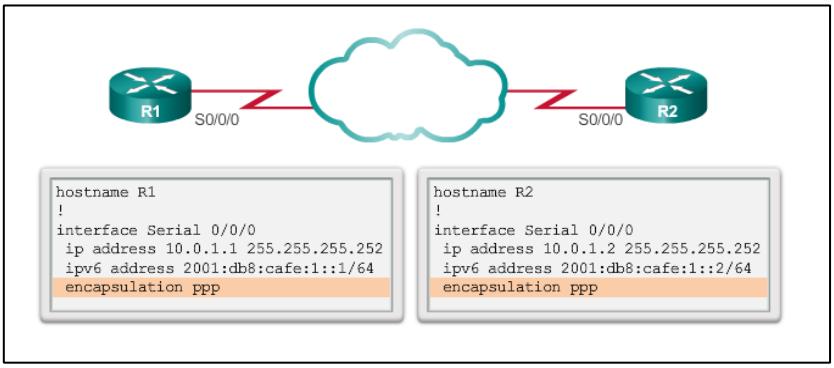
Establecimiento de sesión PPP:

Fase 1: LCP inicia la conexión y negocia las opciones

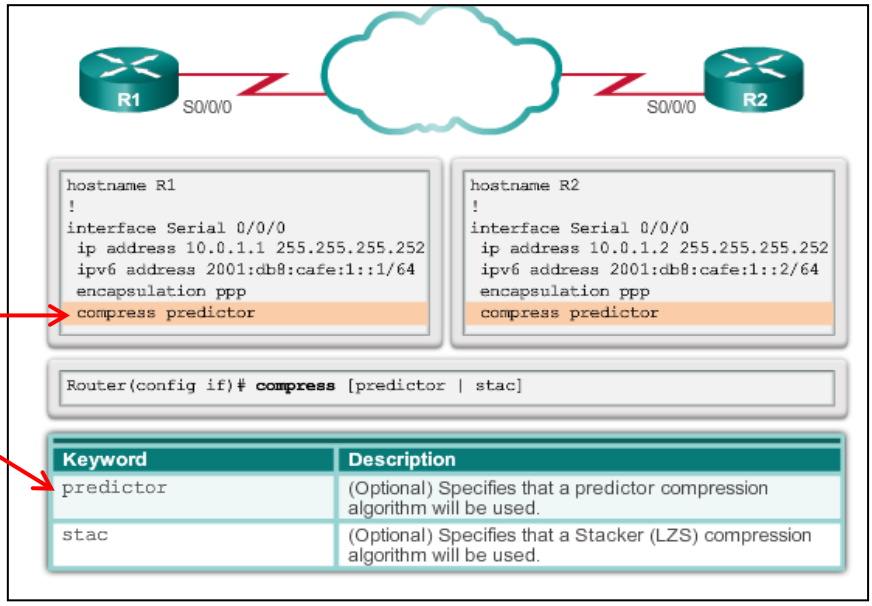
Fase 2: Se determina la calidad del enlace. LCP testea esta calidad.

Fase 3: Se negocia el protocolo de red por NCP

Opciones de configuración PPP:

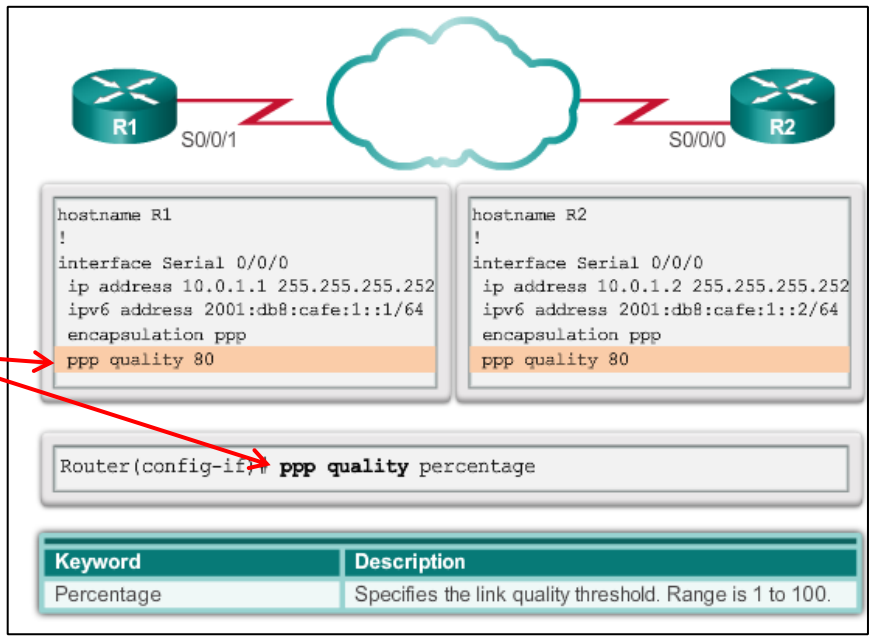


Comandos de compresion para PPP



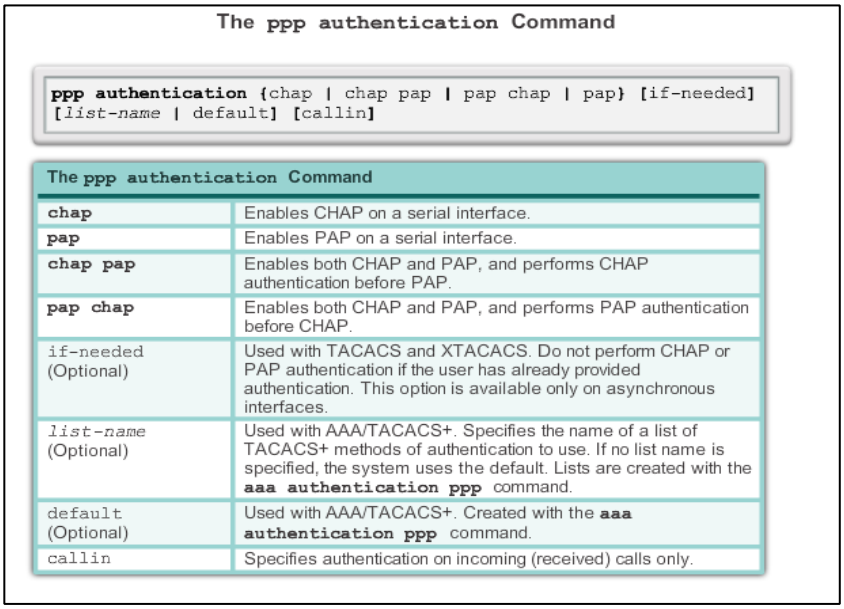
Keyword	Description
predictor	(Optional) Specifies that a predictor compression algorithm will be used.
stac	(Optional) Specifies that a Stacker (LZS) compression algorithm will be used.

Comandos de calidad



Keyword	Description
Percentage	Specifies the link quality threshold. Range is 1 to 100.

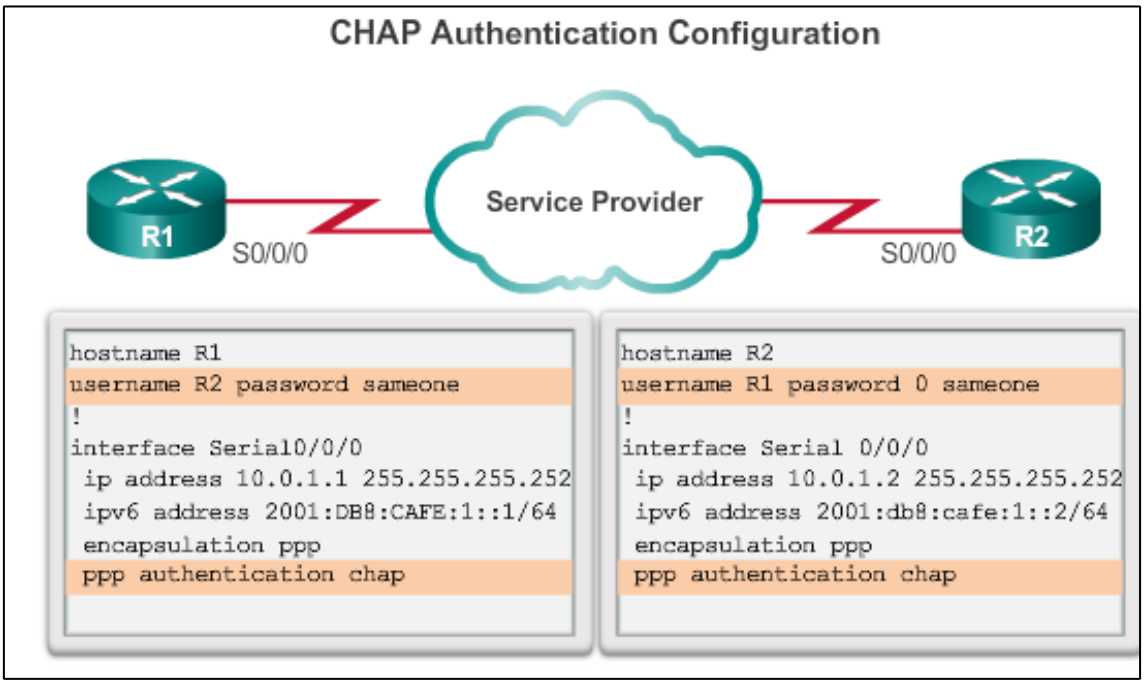
Comandos de Autenticación



The ppp authentication Command

```
ppp authentication {chap | chap pap | pap chap | pap} [if-needed]
[list-name | default] [callin]
```

The ppp authentication Command	
chap	Enables CHAP on a serial interface.
pap	Enables PAP on a serial interface.
chap pap	Enables both CHAP and PAP, and performs CHAP authentication before PAP.
pap chap	Enables both CHAP and PAP, and performs PAP authentication before CHAP.
if-needed (Optional)	Used with TACACS and XTACACS. Do not perform CHAP or PAP authentication if the user has already provided authentication. This option is available only on asynchronous interfaces.
list-name (Optional)	Used with AAA/TACACS+. Specifies the name of a list of TACACS+ methods of authentication to use. If no list name is specified, the system uses the default. Lists are created with the **aaa authentication ppp** command.
default (Optional)	Used with AAA/TACACS+. Created with the **aaa authentication ppp** command.
callin	Specifies authentication on incoming (received) calls only.



Configuración de ppp real:

1- **Estando en R(Config)#** nos metemos en la interface correspondiente
2- **Dentro de la interface R(Config-if)#** metemos el comando encapsulation ppp. Así quedaría: **R(Config-if)# encapsulation ppp**

Configuración real de autenticación:

En **R1**

1- (DATOS DEL OTRO) **Estando en R1(Config)#** metemos el usuario y la contraseña del router con el que nos vamos a conectar, no las contraseñas del nuestro. De tal forma que si nuestro router es R1 y nos queremos hacer ppp con R3 los comandos serán:

R1(Config)#username R3 secret class *(username y secret son los comandos) R3 y class son los nombres

De tal forma que el usuario que estamos metiendo es R3 que es con quien vamos a establecer la comunicación, no nuestro propio nombre y class será la contraseña también para cuando venga algo de R3 hacia R1, pero es la de R3 que se mete en la base de datos de R1.

2- Nos metemos en la interface y para habilitar (en este caso) pap, le metemos el comando:

R1(Config-if)#ppp authentication pap (en este caso hemos configurado una autenticación pap, pero podría haber sido también chap)

3- (DATOS NUESTROS) Para decirle a pap lo que tiene que siempre envíe su usuario (R1) y su contraseña (que en este caso va a ser cisco) metemos el comando**R1(Config-if)#ppp pap sent-username R1 password cisco**

Una vez configurado R1, habrá que configurar en este caso R3 que es con quien se va a establecer la comunicación ppp, de tal forma que para R3 sería:

En **R3**

1- (DATOS DEL OTRO) **Estando en R3(Config)#username R1 secret cisco** (estas son las credenciales de R1, no las de R3, porque es con R3 con quien se va a comunicar. Con esto, se lo estamos metiendo en la base de datos de R3)
2- Nos metemos en la interface y le habilitamos ppp pap (en este caso) con el comando: **R3(Config-if)#ppp authentication pap**
3- (DATOS NUESTROS) Le decimos a pap lo que tiene que hacer, enviar su propio nombre de usuario que es R3 y su propia contraseña que en este caso será class: **R3(Config-if)#ppp pap sent-username R3 password class**

Capítulo 4: Frame Relay

Frame Relay es una red de conmutación de paquetes.

Los SWs Frame Relay trabajan en capa 2.

En Frame Relay no hace falta un conversor de medios, va en tarjeta que hace la función.

Los SWs FR usan los DLCI para conmutar los paquetes hacia un destino.

Configuración Frame Relay Multipunto:

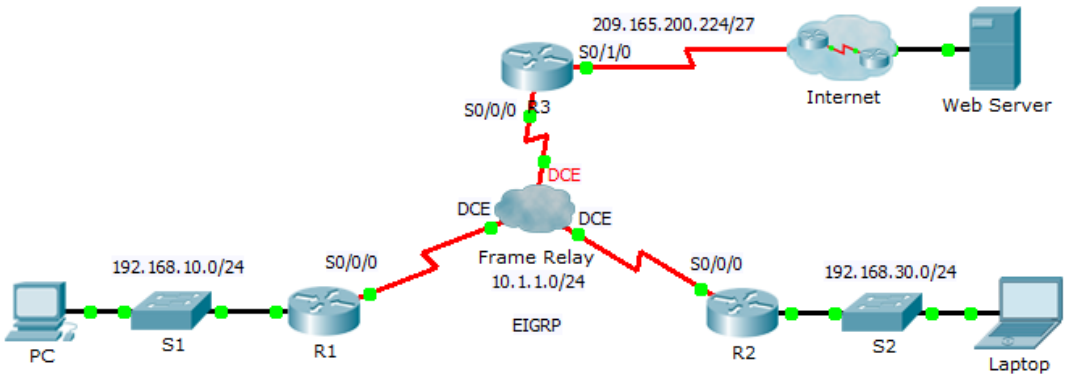
```
R1(config)# interface serial 0/0/1
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# encapsulation frame-relay
R1(config-if)# no frame-relay inverse-arp
R1(config-if)# frame-relay map ip 10.1.1.2 102 broadcast
cisco                          Esta es la Ip de destino
R1(config-if)# no shutdown
```

Configuración Frame Relay punto a punto:

```
R1(config)# interface serial 0/0/1
R1(config-if)# encapsulation frame-relay
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface serial 0/0/1.102 point-to-point
R1(config-subif)# ip address 10.1.1.1 255.255.255.252
R1(config-subif)# bandwidth 64
R1(config-subif)# frame-relay interface-dlci 102
R1(config-fr-dlci)# exit
R1(config-subif)# exit
R1(config)# interface serial 0/0/1.103 point-to-point
R1(config-subif)# ip address 10.1.1.5 255.255.255.252
R1(config-subif)# bandwidth 64
R1(config-subif)# frame-relay interface-dlci 103
R1(config-fr-dlci)#
```

Configuración mapa estático de Frame Relay:

1- Nos metemos en la interface
2- Habilitamos FR con el comando **R(Config-if)#encapsulation frame-relay**
3- Habilitamos FR en las interfaces de los otros Routers con los que vamos a conectar.



Según este escenario, para configurar R1 para usar un mapa estatico de FR, vamos a usar la DLCI 102 para comunicar R1 y R2

DLCI 103 para comunicar R1 y R3

Como ademas los routers deben poder soportar EIGRP multicast sobre 224.0.0.10 el comando broadcast deberá ser requerido, de tal forma que el comando resultante será:

R1(config)# **interface s0/0/0** Ip de destino. Es la ip de R2
R1(config-if)# **frame-relay map ip 10.1.1.2 102 broadcast**
R1(config-if)# **frame-relay map ip 10.1.1.3 103 broadcast**
Ip de destino. Es la ip de R3

En R2 usando la DLCI 201 para comunicar R2 a R1, el comando será:

R2(config-if)# **frame-relay map ip 10.1.1.1 201 broadcast**

..y de R2 a R3 usando la DLCI 203, el comando será:

R2(config-if)# **frame-relay map ip 10.1.1.3 203 broadcast**

En R3 usando la DLCI 301 para comunicar R3 con R1, el comando será:

R3(config-if)# **frame-relay map ip 10.1.1.1 301 broadcast**

…y de R3 a R2 usando la DLCI 302, el comando será:

R3(config-if)# **frame-relay map ip 10.1.1.2 302 broadcast**

Por último, para configurar ANSI como el tipo LMI en R1, R2 y R3, habilitamos el comando:
R1(config-if)# **frame-relay lmi-type ansi**

Para la configuración punto a punto:

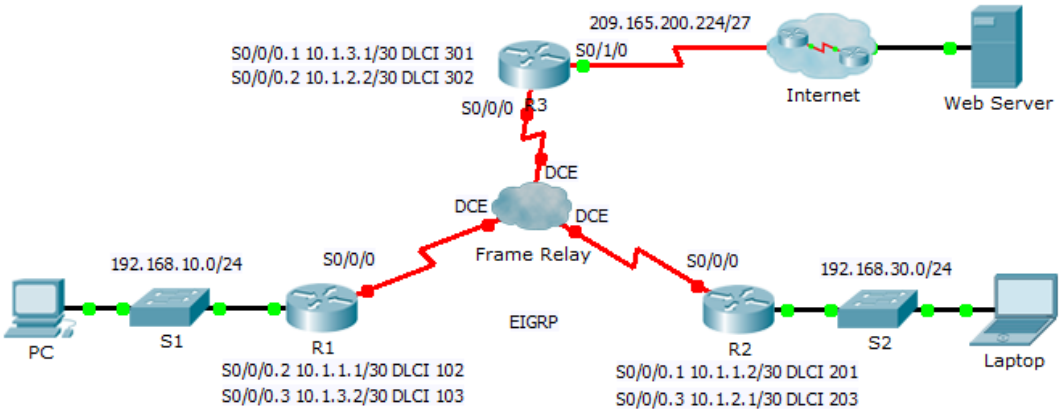
Paso 1: Configurar las subinterfaces en R1, R2, y R3.

a. Para configurar las interfaces de **R1** usaremos en este caso la **DLCI 102** para comunicar R1 con R2, y usaremos la **DLCI 103 para comunicar R1 con R3**.

R1(config)# **interface s0/0/0.2 point-to-point**
R1(config-subif)# **ip address 10.1.1.1 255.255.255.252**
R1(config-subif)# **frame-relay interface-dlci 102**
R1(config-subif)# **interface s0/0/0.3 point-to-point**
R1(config-subif)# **ip address 10.1.3.2 255.255.255.252**
R1(config-subif)# **frame-relay interface-dlci 103**

b. Habilitamos EIGRP autonomous system 1

R1(config)# **router eigrp 1**
R1(config-router)# **network 10.1.1.0 0.0.0.3**
R1(config-router)# **network 10.1.3.0 0.0.0.3**



Una vez hecho esto, habrá que ir a configurar las interfaces de R2 con los correspondientes DLCI. **DLCI 201** is used to communicate from **R2** to **R1**, while **DLCI 203** is used to communicate from **R2** to **R3**. Use the correct IP address in the **Address Table** for each subinterface.

d. Add the appropriate EIGRP entries to **R2** for autonomous system of 1.

e. Configure **R3** to use subinterfaces. **DLCI 301** is used to communicate from **R3** to **R1**, while **DLCI 302** is used to communicate from **R3** to **R2**. Use the correct IP address for each subinterface.

f. Add the appropriate EIGRP entries to **R3** for autonomous system of 1.

Cuando se configura Frame relay Punto a punto o a multipunto, las adyacencias entre routers habrá que hacerlas a mano ya que no estamos en un medio broadcast. Se hace con el comando neighbor + la ip del router adyacente. Una vez metidas las adyacencias, hacer un clear OSPF/EIGRP process.

En ipv6 la dirección ip será la Linjk-local

Para configurar un SW con Frame Relay:

1- Habilitamos FR con el comando: SW(Config)#Frame-relay switching
2- Nos metemos en la interface del SW: interface serial0/0/0
3- Le metemos el tipo de encapsulación: SW(Config-if)# encapsulation frame-relay
4- SW(Config-if)# frame-relay intf-type dce
5- SW(Config-if)# frame-relay route <DLCI IN> interface <Interface out> <DLCI out>
6- Le metemos el clock rate 64000
7- no shutdown

```
Switch FR:
!
Frame-relay switching
interface serial0/0/0
 encapsulation frame-relay
 frame-relay intf-type dce
 frame-relay route <DLCI IN> interface <Interface out> <DLCI out>
 clock rate 64000
 no shutdown
.
```

Capitulo 5 Nat: ya visto

Capítulo 6: Tecnologías de acceso a internet

Son soluciones de acceso a internet.

El segmento del que tratamos es el de la "última milla".

Al proveedor de servicio se le alcanza por dos medios:

1- Ruta por defecto (Cliente pequeño): 0.0.0.0/0
2- BGP (Empresa grande. Consume muchos recursos pero soporta múltiples redes)

VPN:

Una VPN es una canal seguro entre 2 puntos. Hay dos tipos de topologías:

Punto a punto: Ejemplo de conexión entre mi sede central en Madrid y la de Barcelona. Los dispositivos de conexión en ambos extremos de la conexión VPN conocen la configuración VPN de antemano.
La VPN permanece estática y los hosts internos no tienen conocimiento de la existencia de la VPN.
Las VPNs de punto a punto conectan redes completas entre sí.
Los hosts envían y reciben tráfico TCP/IP a través del gateway VPN, el cual puede ser un router, un firewall, un concentrador Cisco VPN o un Cisco ASA 5500.

El gateway de la VPN es responsable de encapsular y cifrar el tráfico saliente de un sitio y enviarlo a través del túnel VPN sobre Internet hacia el otro gateway de VPN en el sitio de destino.

Locales en remoto: se crea cuando la información de la VPN no se configura en forma estática, sino que se permite que la información cambie en forma dinámica.
La información requerida para establecer la conexión VPN, como puede ser la dirección IP del trabajador a distancia, cambia dinámicamente de acuerdo a la ubicación del mismo.
El host tiene instalado Cisco VPN Client, cuando el host intenta enviar tráfico destinado a la VPN, la aplicación Cisco VPN Client encapsula y cifra dicho tráfico antes de enviarlo a través de Internet hacia el gateway VPN en el borde de la red destino.
Un pc debe tener un software específico para permitir la conexión VPN a la red de destino.
Un enlace WAN dedicado vale una pasta y lo otorgan las ISPs. De ahí que el uso de una vpn sea mejor y más barato.

Hay IOS que soportan VPN y otras no. VPN se puede implementar por mediación de software o de hardware. A veces es recomendable meter un módulo de expansión hardware en el router ISR

Cisco IOS VPN SSL, es otra implementación de VPN de acceso remoto por web. Las VPNs con SSL no son un reemplazo completo de las VPNs con IPsec, si no que en muchos casos son complementarias.
Las VPNs con SSL no admiten el mismo nivel de seguridad criptográfica que IPsec
Las VPNs con SSL son compatibles con DMVPNs, Cisco IOS Firewalls, IPsec, Cisco IOS IPS, Cisco Easy VPN y NAT.
Una VPN con IPsec es más segura que con SSL.
Las VPNs con SSL no permiten la aceleración por hardware. Sólo disponen de soporte por software.

Hay 3 tipos de VPNs de capa 3:

- GRE
- MPLS
- IPsec

Para poder hacer una VPN necesitaremos apoyarnos en software o hardware.
Por VPN puede viajar voz, videoconferencia y datos.

Router UTM: Router bueno que ouede albergar distintos tipos de datos
Cableado: Usa el protocolo DOCSIS en capa 2
Componentes del cable: Hay dos tipos de equipos necesarios para enviar señales de módem digitales ascendente y descendente en un sistema de cable: Sistema de Terminación de Cable Módem (CMTS) en la cabecera del operador de cable.
Cable módem (CM) en el extremo del abonado.

Tipos de soluciones para banda ancha:

Cable - Ancho de banda es compartido por muchos usuarios.
DSL - ancho de banda limitada que es la distancia y minúsculas.
Fiber-to-the-Home - Requiere fibra de acceso a la red de superposición.
Celular / Móvil - La cobertura es a menudo un problema, el ancho de banda relativamente limitado.
Wi-Fi Mesh - Muchos municipios no cuentan con una red de malla desplegada.
WiMAX - Velocidad de bits está limitado a 2 Mb / s por abonado; tamaño de la celda

es de 1,25 millas (1-2 km.)
Satélite - Caro; capacidad limitada por suscriptor.

Capítulo 7: Conectividad sitio a sitio

Tipos de conexiones VPN:

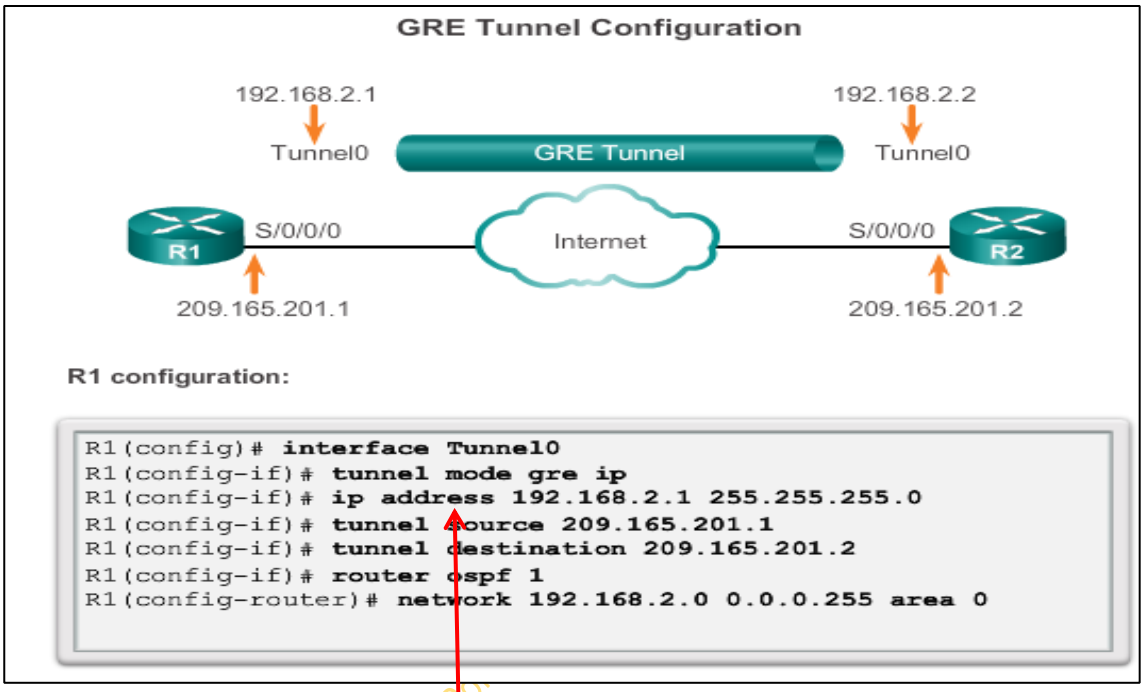
- **Según tipo de conexión**:
  - Side to side (sitio a sitio)
  - De acceso remoto mediante un cliente Cisco. Las VPNs de acceso remoto son conexiones Dial-Up: a petición/por marcado.
- **Según el nivel de confianza**:
  - Intranet
  - Extranet
  - Internet
- **Según el modelo TCP/IP**:
  - De acceso a la red
  - De la capa deinternet
  - De la capa de transporte
  - De la capa de aplicación
- **Según el protocolo**:
  - *Protocolo de acceso a la red*:
    - Frame Relay
    - ATM
    - MPLS
    - Metro-Ethernet
    - **L2TP**
    - **PPTP**
    - PPP
  - *Protocolo de Internet*:
    - **GRE** (en texto plano)
    - **IPSec** (más seguro)
    - GREoIPsec
    - DMVPN
    - DTS
  - *Protocolo de Transporte*
    - SSL
    - TLS

    } **TCP de capa 4 y puerto 443**
  - *Protocolo de aplicación*
    - SSh
    - HTTPS
    - SNMTPv3

Una VPN es un establecimiento de punto a punto pero puede ser seguro o no tan seguro. No cifra los datos necesariamente. Dependerá del tipo de VPN que utilicemos. Además puede ser con confidencialidad o sin ella.

VPN tipo GRE: Soporta tráfico unicast y multicast. IPsec solo soporta unicast. No es muy seguro en internet pero entre redes privadas puede ser muy útil.

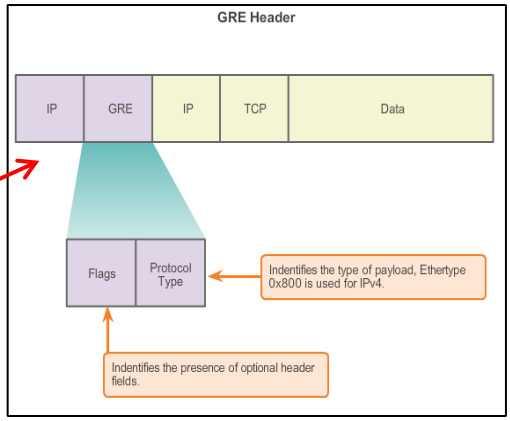
Configuración Túnel GRE:



1º- Creamos y habilitamos el túnel GRE:

R1(Config)#interface Tunnel0

R1(Config-if)#tunnel mode gre ip



2º- Le otorgamos una ip privada. Esta ip que se le pone tiene que ser de la misma red en ambos routers y puede ser la que queramos (ya que al haberle habilitado el túnel gre, la encapsulación de las tramas ha cambiado), siempre y cuando sea de la misma red: **R1(Config-if)#ip address 192.168.2.1 255.255.255.0**

3º- Le decimos la dirección ip pública de origen (La nuestra: Nuestra ip pública sumministrada por el proveedor de servicio, la ip del router): **R1(Config-if)#tunnel source 209.165.201.1**

4º Le decimos la dirección pública de destino: **R1(Config-if)#tunnel destination 209.165.201.2**

(en este caso las direcciones son consecutivas, pero no tiene porque ser así. Cada dirección pública puede ser muy diferente)

5º- Después de habilitar el túnel VPN GRE hay que habilitar un protocolo de enrutamiento, que en este ejemplo es OSPF. Si se habilitan/declaran las redes LAN, éstas podrán ser visibles entre routers a través del túnel VPN. Evidentemente dentro de un túnel VPN no hará falta NAT ya que por el túnel VPN los equipos piensan que están directamente conectados y no han salido a internet.

R2 configuration:

```
R2(config)# interface Tunnel0
R2(config-if)# tunnel mode gre ip
R2(config-if)# ip address 192.168.2.2 255.255.255.0
R2(config-if)# tunnel source 209.165.201.2
R2(config-if)# tunnel destination 209.165.201.1
R2(config-if)# router ospf 1
R2(config-router)# network 192.168.2.0 0.0.0.255 area 0
```

Aquí en el router 2 vemos que la dirección privada que le mete es de la misma red que en R1 pero distinta ip.

Por otro lado la ip pública de origen, la suya propia, coincide con la de destino de R1 y viceversa.

R1 configuration:

```
R1(config)# interface Tunnel0
R1(config-if)# tunnel mode gre ip
R1(config-if)# ip address 192.168.2.1 255.255.255.0
R1(config-if)# tunnel source 209.165.201.1
R1(config-if)# tunnel destination 209.165.201.2
R1(config-if)# router ospf 1
R1(config-router)# network 192.168.2.0 0.0.0.255 area 0
```

GRE Tunnel Commands

Command	Description
tunnel mode gre ip	Specifies GRE tunnel mode as the tunnel interface mode, in interface tunnel configuration mode.
tunnel source *ip_address*	Specifies the tunnel source IP address, in interface tunnel configuration mode.
tunnel destination *ip_address*	Specifies the tunnel destination IP address, in interface tunnel configuration mode.
ip address *ip_address mask*	Specifies the IP address of the tunnel interface.

IPSec VPNs:

IPSec es un conjunto de algoritmos criptográficos estándar. Son algoritmos para la:

- **Confidencialidad:** Cifra los datos.
- **Integridad**: Mantiene los datos sin cambios e íntegros.
- **Autenticación:** Verifica la identidad de la fuente de los datos que se envía, se asegura de que la conexión se realiza con el socio de comunicación deseado, IPsec utiliza Internet Key Exchange (IKE) para autenticar usuarios y dispositivos que pueden llevar a cabo la comunicación de forma independiente.
- **Intercambio de claves.**

CIA: confidencialidad, integridad y autenticación

Estos algoritmos los gestiona el administrador. No tiene porqué implementarse todos. Es a nuestra discreción.

Tipos de IPSec:

- Modo Tunnel (se añade un nuevo encabezado IP)
- Modo transporte (no se añade un nuevo encabezado IP)

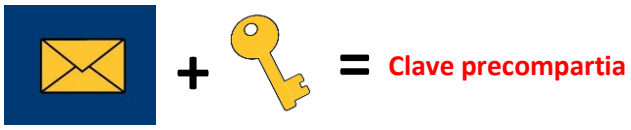
El modo Tunnel es el recomendado por la encapsulación y por el nuevo cifrado IPSec: El texto plano se convierte en cifrado. El tráfico que se cifra lo hacen las ACLs y se llaman Crypto-ACLs. Son ellas las que definen el tráfico interesante.

Confidencialidad con cifrado:

Paso de texto plano a texto cifrado: criptograma

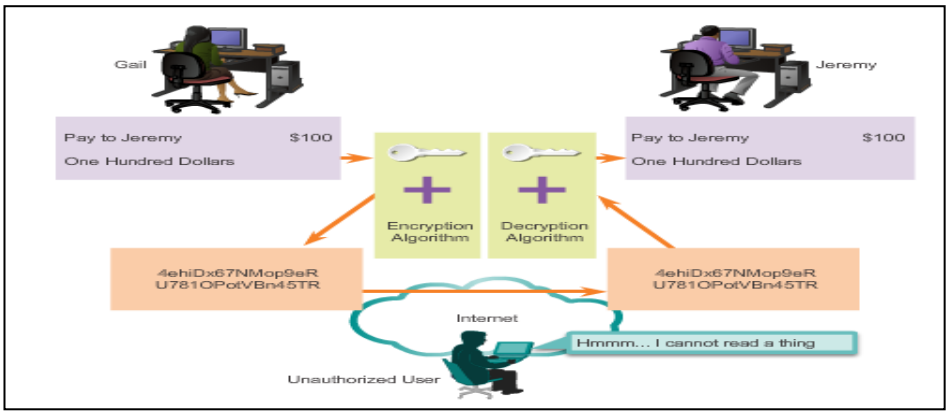
El algoritmo puede ser simétrico o asimétrico.

Una clave precompartida es simétrica. La clave precompartida la forman los datos + la clave en sí con el mismo algoritmo.



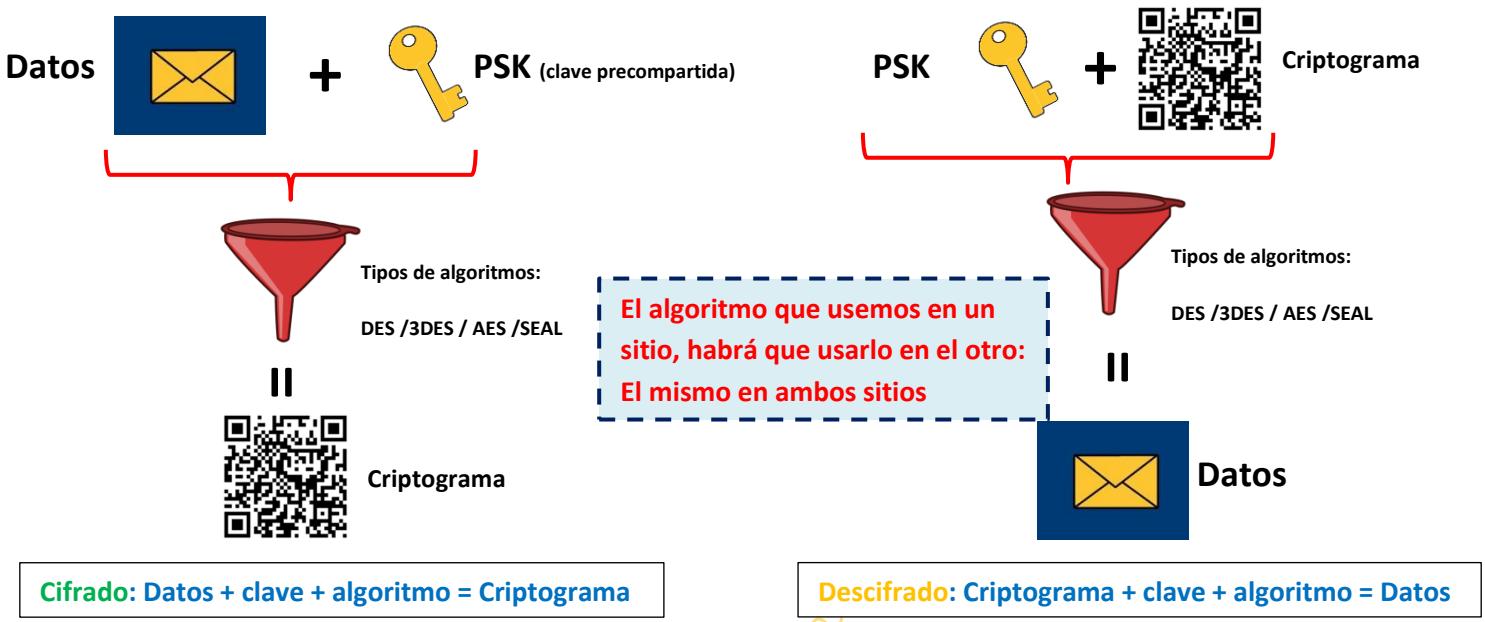
Un extremo del túnel cifra y el otro descifra.

Tiene que haber clave precompartida y con el mismo algoritmo en ambos extremos



Dos tipos de algoritmos cifrados:

- Simétricos
- Asimétricos



Algoritmos asimétricos:

- Utilizan dos claves:
  - Pública: es la que cifra
  - Privada: es la que descifra

A manda a B:

1º- A encripta: Los datos de A + la clave pública de B hace el criptograma que A mandará a B

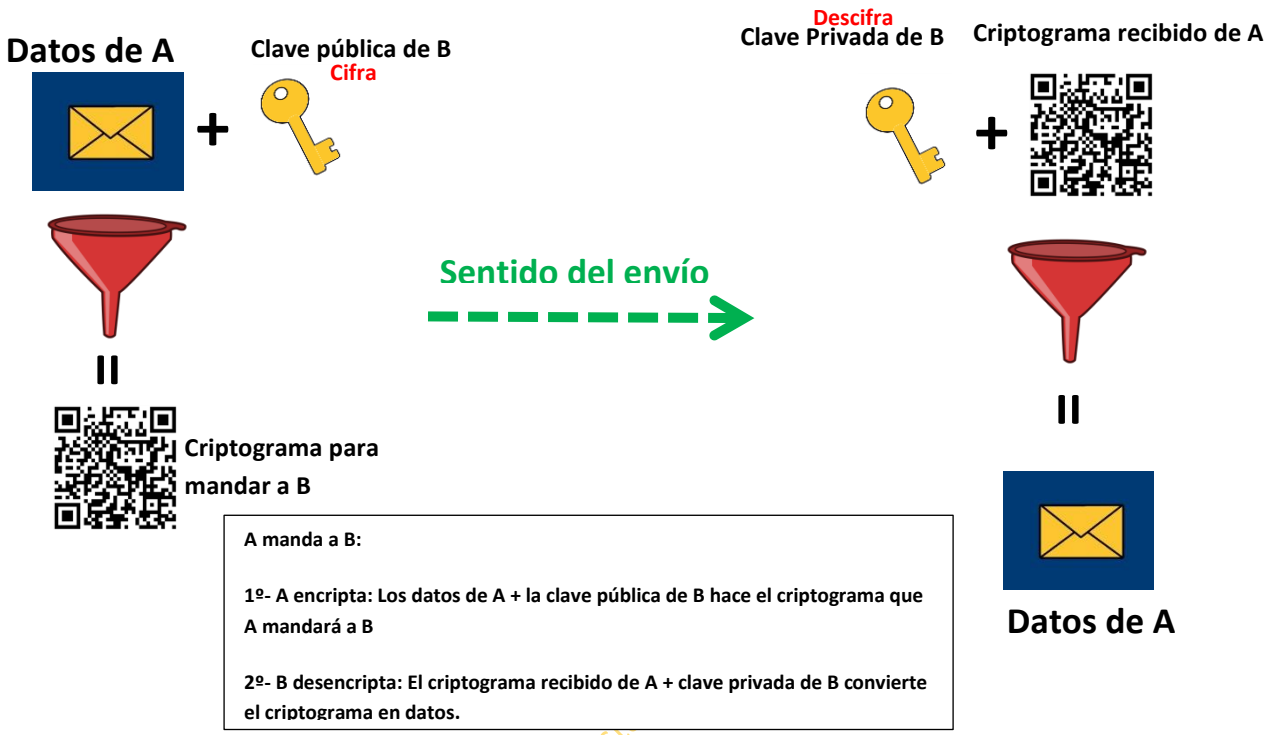
2º- B desencripta: El criptograma recibido de A + clave privada de B convierte el criptograma en datos.

B manda a A:

1º- B encripta los datos metiéndole la clave pública de A, lo que genera el criptograma que mandará a A.

2º- A desencripta el criptograma recibido de B con la clave privada de A, lo cual convertirá el criptograma en datos.

La clave privada es única. Solo la conoce el equipo o la persona.



A manda a B:

1º- A encripta: Los datos de A + la clave pública de B hace el criptograma que A mandará a B

2º- B desencripta: El criptograma recibido de A + clave privada de B convierte el criptograma en datos.

B manda a A

1º- B encripta los datos metiéndole la clave pública de A, lo que genera el criptograma que mandará a A.

2º- A desencripta el criptograma recibido de B con la clave privada de A, lo cual convertirá el criptograma en datos.

El algoritmo simétrico tiene una longitud de clave de 64 bits hasta 512. Son pequeños pero más rápidos.

El algoritmo asimétrico va desde 300 bits hasta 1024, 2048, 4096, etc.

PKI es la infraestructura de clave. Confianza depositada en un 3º para establecer una comunicación segura

PKCS: Certificado Digital

Protocolo DIFFIE-HELLMAN: para intercambiar las claves de manera segura en internet. DH asegura el intercambio de claves pero no cifra nada. DH creará esa clave.

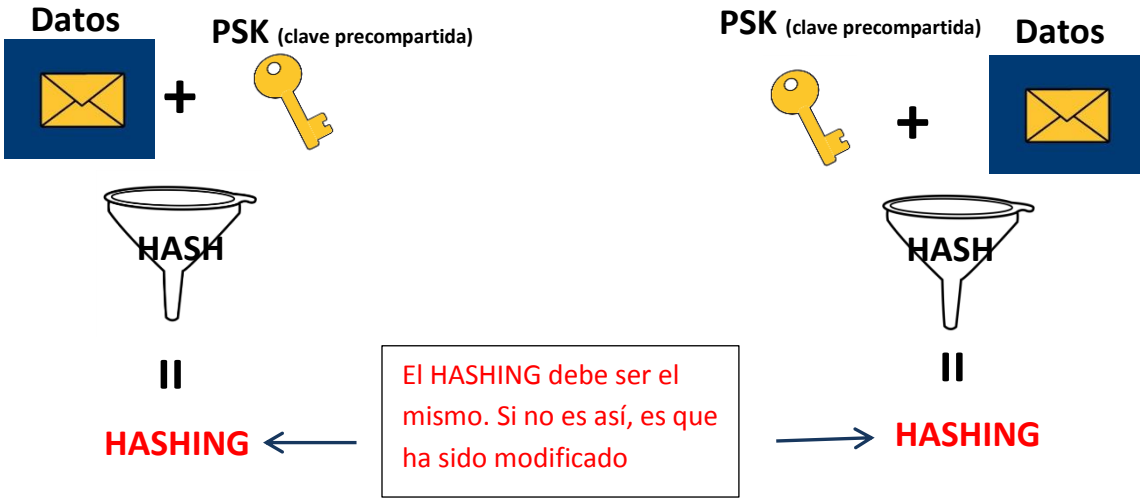
DH permite que dos dispositivos compartan clave de manera segura en un medio inseguro como internet.

Antes de establecer el túnel VPN habrá que negociar el DH.

Integridad algoritmo de HASH (algoritmo unidireccional) y siempre tiene una longitud fija.

Hash ofrece integridad pero no autenticación. Si se capturan los datos, al ir en texto plano, nada nos aseguraría que pudieran haber sido modificados.

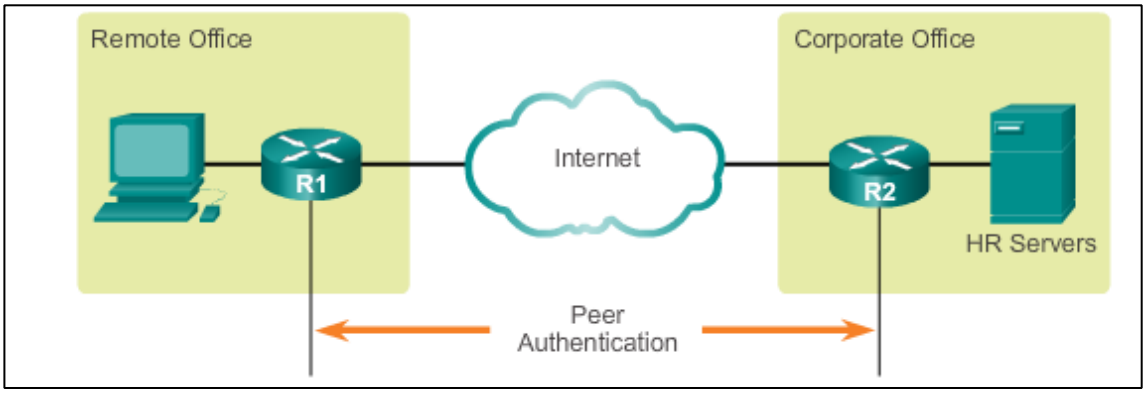
Por eso la clave precompartida es necesaria (PSK)



Si los datos fueran modificados, como no tiene el PSK, el hashing sería diferente, con lo cual se descartaría el paquete al haber sido corrompido. A esto se le llama HMDC: Hash-Based Message Authentication Code.

Los algoritmos de integridad con HMDC con HASH son: **MD5 y SHA**

Antes de hacer las VPNs los equipos deben autenticarse.



Una **Firma Digital** lleva intrínseca nuestra clave privada.

Si A quisirea comunicarse con B le solicitaria la clave pública a B. En ese momento podría mandar encriptado.

Estando en A:

Datos + Clave pública de B + RSA = datos cifrados

Para **sumar integridad**: **Datos + algoritmo de hash (MD5)= Hushing**

Y para **sumar autenticacion**: **Clave privada de A + RSA= Hushing cifrado= datos cifrados con hushing cifrado.**

Así tendremos confidencialidad, autenticación e integridad.

Estando en B:

1- **Confidencialidad**:

Datos cifrados llegados de A + clave privada de B + RSA= Datos

2- **Integridad**:

Creamos el Hushing: Datos + MD5= Hushing

3- **Autenticación**:

Una vez cifrado el hushing + la clave pública de A autenticamos que fue A quien originó el mensaje. Al aplicarle el algoritmo RSA, el resultado debe ser el mismo:

Hushing = Hushing

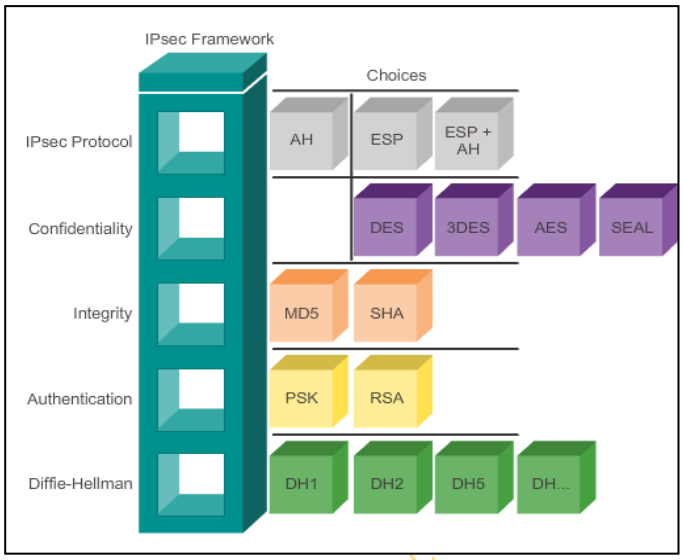
Al meter nuestra clave privada le estamos dando autenticación. Si solo le metiéramos la pública le estaríamos dando confidencialidad pero no autenticación.

Recordamos:

PSK: Clave precompartida usando canal seguro. Hay que establecerla antes del Tunel VPN. Es de algoritmos simétricos

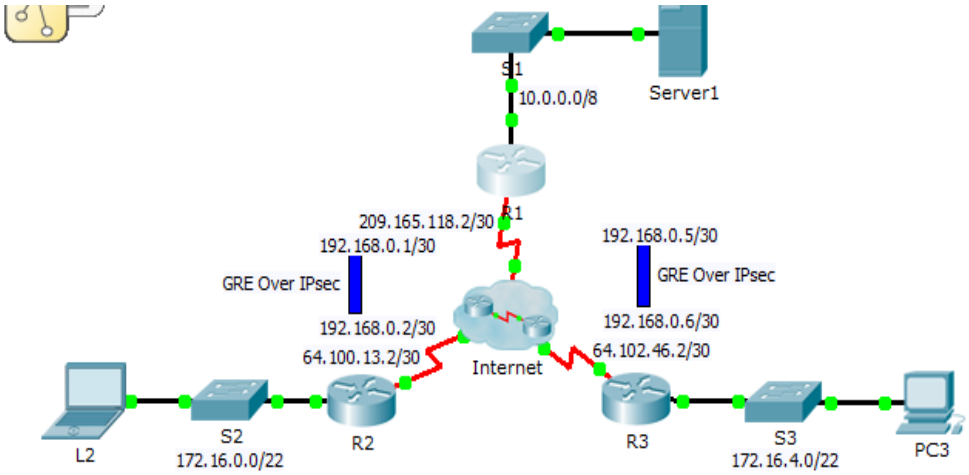
Firmas RSA: Cifran nuestra clave privada. Con ella añadimos autenticación.

PKI: Certificado Digital. Es añadir todavía más seguridad.



Configuración IPSec:

0- Creamos ACLs para permitir el tráfico ESP (IP 50), AH (IP 51) y ISAKMP (UDP 500) en los routers de borde.
1- SA (Asociación de seguridad) IKE fase 1
2- SA IKE Fase 2
3- Creamos el CRYPTO-MAP
4- Creamos la CRYPTP-ACL (para permitir solo el tráfico interesante)
5- Aplicamos el CRYPTP-MAP a la interface física o virtual



1- **Configuración de las ACL**

En este caso hay que configurar 2 ACLs:

1.1- Una para R1-R2 que en este caso será la 102 para permitir el tráfico de las redes de R1 y R2, y

1.2- Otra para R1-R3 que será la 103 para permitir el tráfico de las redes de R1 y R3.

1.1-R1(config)# **access-list 102 permit ip 10.0.0.0 0.255.255.255 172.16.0.0 0.0.3.255**

Hacemos otra ACL a la que llamaremos 103 para permitir también el tráfico de la LAN de R3 la red 172.16.4.0

1.2-R1(config)# **access-list 103 permit ip 10.0.0.0 0.255.255.255 172.16.4.0 0.0.3.255**

Las crypto ACLs se hacen en cada Router

En VPN al crear una ACL no está implícito el deny-any

Parámetros ISAKMP

Parámetro	Palabra clave	Valores aceptados	Valor por defecto	Descripción
encryption	des 3des aes aes 192 aes 256	Data Encryption Standard (DES) de 56 bits Triple DES AES de 128 bits AES de 192 bits AES de 256 bits	des	Algoritmo de cifrado del mensaje
hash	sha md5	SHA-1 (variante de HMAC) MD5 (variante de HMAC)	sha	Algoritmo de integridad del mensaje (hash)
authentication	pre-share rsa-encr rsa-sig	Claves precompartidas Nonces RSA cifrados Firmas RSA	rsa-sig	Método de autenticación de pares
group	1 2 5	Diffie-Hellman (DH) de 768 bits DH de 1024 bits DH de 1536 bits	1	Parámetros de intercambio de claves (identificador de grupo DH)
lifetime	*seconds*	Puede especificarse cualquier cantidad de segundos	86,400 segs (un día)	Tiempo de vida de SA establecidos por ISAKMP

2- **Configuración ISAKMP FASE 1: (en este caso, habrá que habilitar una fase 1 para R1-R2 y otra para R1-R3)**

Para R1-R2:

R1: lo hacemos con el comando crypto isakmp + policy + numero de la prioridad, que en este caso le hacemos coincidir con el número de la ACL

R1(config)# crypto isakmp policy 102

3- Después le decimos que tipo de encriptación va a tener (su algoritmo de cifrado), que en este caso va a ser AES. Podría haber sido: des, 3des, aes o seal.

R1(config-isakmp)# **encryption aes**

4- Ahora configuramos la autenticación con el comando **authentication pre-share. La autenticación podría haber sido pre-share, rsa-encr o rsa-sig**

R1(config-isakmp)# **authentication pre-share**

5- Le decimos a qué **tipo de Deffie-Hellman** va a tener: Podría ser group 1, 2 ó 5. En este caso hemos tomado el 5. Para decir qué tipo de DH tendrá es con el comando **group + número:**

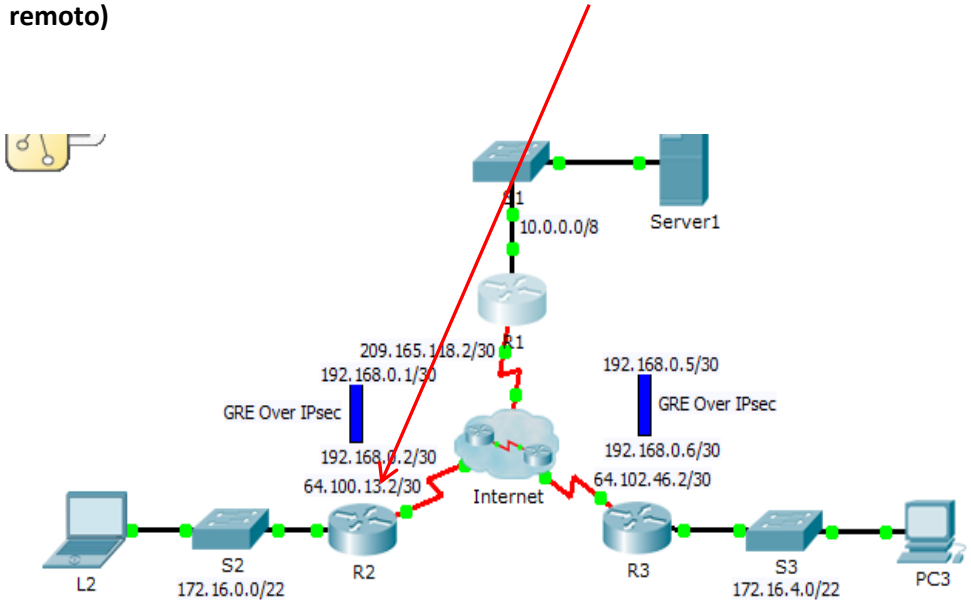
R1(config-isakmp)# **group 5**

R1(config-isakmp)# **exit**

6- Por último como hemos escogido antes PSK como método de autenticación, habrá que tener en cuenta que si se seleccionan PSK como método de autenticación, es necesario **configurar la clave precompartida**. En este caso la clave va a ser "cisco" por lo que habrá que añadir el comando:

R1(Config)#crypto isakmp key + la clave + comando address + ip de la dirección de destino.

R1(config)# **crypto isakmp key cisco address 64.100.13.2 (ip pública del Gateway remoto)**



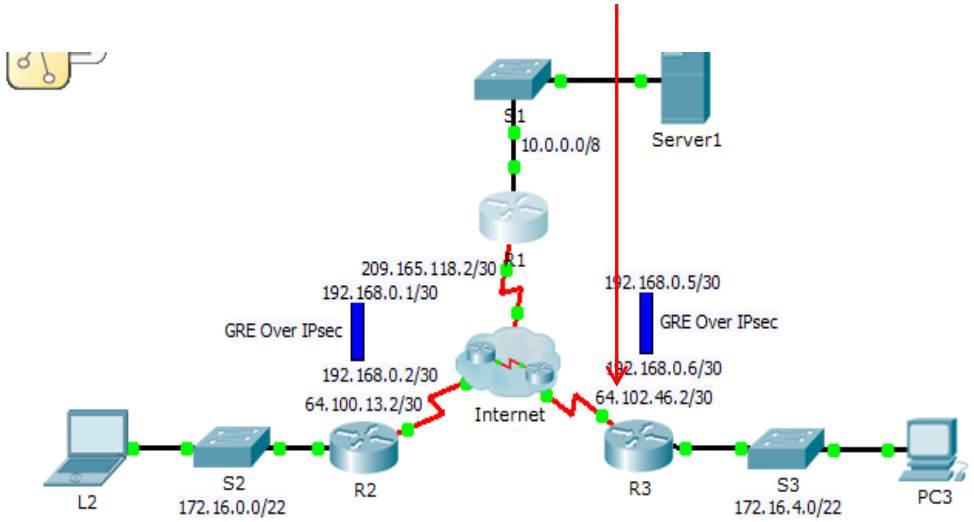
Para R1-R3:

Ya hemos configurado las políticas de la ACL 102, pero habrá que hacer también la de la 103 para la red LAN de R3.

La configuración es exactamente igual pero cambiando R1(config)# **crypto isakmp policy 102 por** R1(config)# **crypto isakmp policy 103**

Encriptación, autenticación y grupo 5 de DH seguirá siendo el mismo. Cambiaremos solo la dirección de la red que queremos aceptar que en este caso es la de la red de R3, la 64.102.46.2

R1(config)# **crypto isakmp key cisco address 64.102.46.2**



Configuración IPSec ISAKMP Fase 2: *(en este caso habrá que habilitar uno para R1-R2 y otro para R1-R3)

Se trata de **Configurar el conjunto de transformación**: Decidir si AH o ES

Los conjuntos de transformación se limitan a: 1 transformación AH y/o 1 ó 2 transformaciones ESP.
Se hace con el comando **R1(Config)#crypto ipsec transform-set (***esto configura qué tipo de políta vamos a usar en el túnel)*

Cada transformación representa un protocolo de seguridad IPsec (AH ó ESP) junto con un algoritmo asociado.

En el caso de ahora usaremos esp-aes y esp-sha-hmac:

Para R1-R2:

1- Con el comando **crypto ipsec transform-set** habilitamos la **fase 2** de IPSec ISAKMP.
1.1- Además hay que añadirle un nombre a esta fase que en este caso lo llamaremos R1_R2_Set.
1.2- Por último le añadimos el tipo de cifrado que le vamos a meter, que en este caso elegiremos esp-aes (podría haber sido: esp-3des, esp-des)
1.3- y el tipo de integridad que queremos, que en este caso usaremos esp-sha-hmac (podía haber sido también esp-md5-hmac)

Así el comando completo para activar la fase 2 quedará:

R1(config)# **crypto ipsec transform-set** R1_R2_Set **esp-aes esp-sha-hmac**

2- **Configuramos el Crypto-map**

El crypto-map es lo que se asocia a la interface. Se compone de:

Sólo puede asignarse un crypto-map a una interfaz.
La ACl es sólo un valor más de los 5 que componen el cripto-map.
Habrá distintos tipos de cripto-map dependiendo del tipo de tráfico.

Y su configuración es con el comando crypto map, de tal forma que sería:

R1(Config)#crypto map + nombre que le demos **+ valor decimal + ipsec-manual/ipsec-isakmp**. Vamos a llamar al crypto mapa R1_R2_Map con valor decimal de 102. De tal forma que quedaría:

Comando	Descripción
set	Utilizado con los comandos **peer**, **pfs**, **transform-set**, y **security-association**.
peer [*hostname* | *ip-address*]	Especifica los pares IPsec permitidos por dirección IP o nombre de host.
pfs [**group1** | **group2**]	Especifica el Grupo 1 o Grupo 2 DH.
transform-set [*set_name(s)*]	Especifica la lista de conjuntos de transformación en orden de prioridad. Cuando se utiliza el parámetro **ipsec-manual** con el comando **crypto map**, sólo puede definirse un único conjunto de transformación. Cuando se utilizan los parámetros **ipsec-isakmp** o **dynamic** con el comando **crypto map**, pueden especificarse hasta seis conjuntos de transformación.
security-association lifetime	Configura los parámetros de tiempo de vida de SA en segundos o kilobytes.
match address [*access-list-id* | *name*]	Identifica una ACL extendida por su nombre o número. El valor debe coincidir con los argumentos access-list-number o name de una ACL IP extendida previamente definida.
no	Utilizado para eliminar los comandos ingresados con el comando **set**.
exit	Sale del modo de configuración de crypto-map.

R1(config)# crypto map R1_R2_Map 102 ipsec-isakmp
Cuando metemos el comando crypto map para configurar el cripto mapa, el prompt cambiará a R1(Config-crypto-map)#

R1(Config-crypto-map)#

3- **Definimos la ip del otro extremo**, que en este caso para R1 es la 64.100.13.2 del túnel con el comando **Set peer**, de tal forma que quedaría:
R1(config-crypto-map)# set peer 64.100.13.2

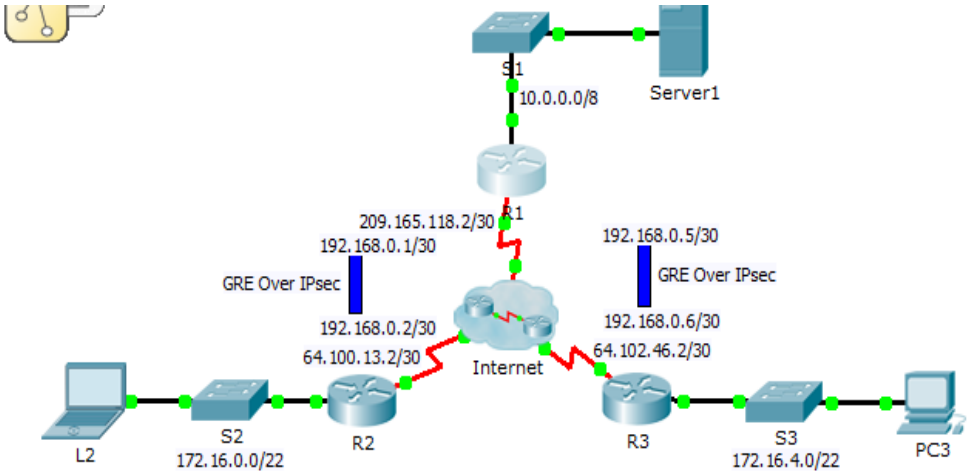
4- Asociamos la política IPSec que creamos antes con el nombre de R1_R2_Set al crypto mapa.
Esto se hace estando metidos en el prompt del crypto mapa (R1(Config-crypto-map)#), con el comando set transform-set + el nombre de la política que creamos que fue en este caso R1_R2_Set.
Nombre de la política creada: R1_R2-Set
Nombre del crypto Mapa: R1_R2_Map
Así, el comando completo quedaría:
R1(config-crypto-map)# **set transform-set** R1_R2_Set
Con ello asociamos la política ipsec al crypto map

5- Por último, asociamos la ACL con el crypto mapa con el comando **match address** + el **valor decimal** que le dimos a la ACL, que en este caso fue 102.
Así, quedaría: **R1(config-crypto-map)# match address *102 *(valor decimal que le dimos a la ACL)**

Para R1-R3:

Habrá que hacer lo mismo cambiando las IPs y los nombres de los cripto mapas para unir el tunel al otro extremo.

R1(config)#crypto ipsec transform-set R1_R3_Set esp-aes esp-sha-hmac (como vemos, hemos cambiado el nombre)
R1(config)#crypto map R1_R3_Map 103 ipsec-isakmp
R1(config-crypto-map)#set peer 64.102.46.2 (esta ip ha cambiado al ser el otro extremo del tunel)
R1(config-crypto-map)#set transform-set R1_R3_Set
R1(config-crypto-map)#match address 103



Asociación con la interface correspondiente: **Solo un crypto map por interface. Para meter otra, habría que hacer otro crypto map con el mismo nombre pero número de secuencia distinto, o crear subinterfaces para albergar los crypto mapas, de tal forma que podría ser:**

Nos metemos en la interface que en este caso es la serial 0/0/0 *creamos subinterfaces

R1(config)# **interface S0/0/0.10**
R1(config-subif)# **crypto map R1_R2_Map**

R1(config)# **interface S0/0/0.20**
R1(config-subif)# **crypto map R1_R3_Map**

Con el comando **show crypto isakmp sa** nos dice el túnel establecido
Con el comando **show crypto ipsec sa** nos dice los paquetes encapsulado y cifrados por lo cual vemos si ipsec está funcionando o no.

Ipsec así estará completamente configurado en R1. Ahora le metemos la configuración necesaria a R2 y a R3.

En R2:

Fase 1

1- Creamos las ACLs. : En este caso la dirección de origen es la Lan de R2, la 172.16.0.0/22 y la de destino, la Lan de R1 la 10.0.0.0/8, de tal forma que la ACL quedaría:

R2(config)# **access-list 102 permit ip 172.16.0.0 0.0.3.255 10.0.0.0 0.255.255.255**

*no hay que crear acl entre R2 y R3 porque no hay túnel ahí.

2- Creamos la política: R2(config)# crypto isakmp policy 102

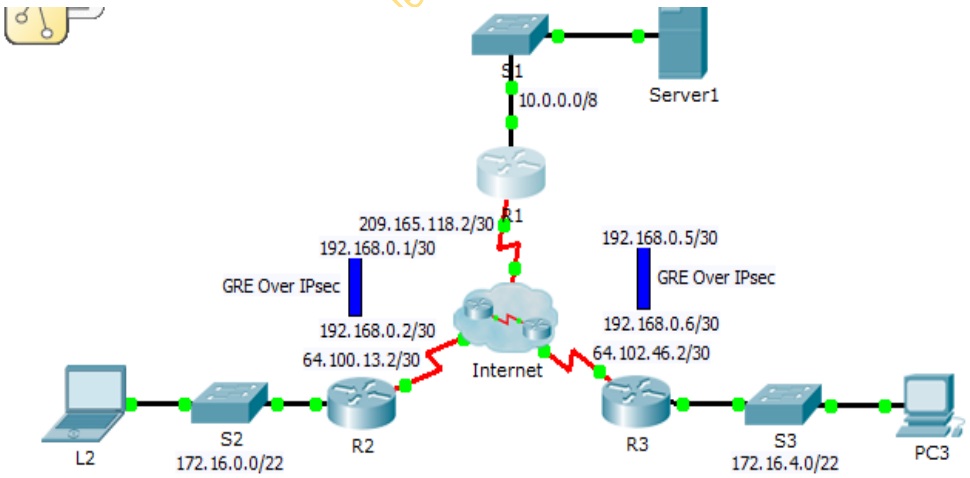
3 Configuramos el algoritmo de cifrado, que será el mismo de R1, o sea aes en este caso: R2(config-isakmp)# encryption aes

4- Configuramos el algoritmo de autenticación que tal como hicimos en R1 es pre-share: R2(config-isakmp)# Authentication pre-share

5- Definimos el tipo de Diffie-Hellman que tal como hicimos en R1 es group 5: R2(config-isakmp)# group 5

6- **configuramos la clave precompartida**. En este caso la clave va a ser también "cisco" . La dirección ip pública de destino en esta caso es la 209.165.118.2, por lo que el comando será:

R1(config)# **crypto isakmp key cisco address 209.165.118.2 (ip pública del Gateway remoto)**



Fase 2 de R2

R2(config)# **crypto ipsec transform-set *R1_R2_Set esp-aes esp-sha-hmac**

R1(config)# **crypto map *R1_R2_Map 102 ipsec-isakmp**

R1(config-crypto-map)# **set peer 209.165.118.2**

R1(config-crypto-map)# **set transform-set R1_R2_Set**

R1(config-crypto-map)# **match address 102**

R1(config-crypto-map)# **exit**

*Los nombres que hayamos puesto en R1 para el transform_set (en este caso **R1_R2_Set**) y para el crypto map (en este caso **R1_R2_Map**), deberán ser los mismos en R2. Lo mismo habrá que hacer en R3. Los nombres siempre los mismos, sino, la configuración estaría mal.

Asociación con la interface correspondiente:

Nos metemos en la interface que en este caso es la serial 0/0/0

R2(config)# **interface S0/0/0**

R2(config-if)# **crypto map R1_R2_Map**

En R3

Fase 1:

1- Crear la ACL:

Lo mismo que R2 pero teniendo en cuenta que la ACL será la 103 permitiendo el tráfico de la LAN de R3: En este caso la dirección de origen es la Lan de R2, la 172.16.4.0/22 y la de destino, la Lan de R1 la 10.0.0.0/8, de tal forma que la ACL quedaría:

R2(config)# **access-list 103 permit ip 172.16.4.0 0.0.3.255 10.0.0.0 0.255.255.255**

*no hay que crear acl entre R2 y R3 porque no hay túnel ahí.

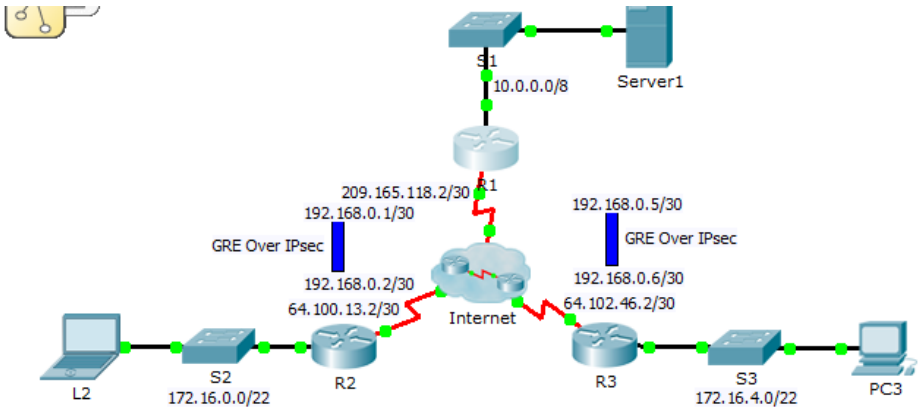
2- Creamos la política: R3(config)# crypto isakmp policy 103

3- Configuramos el algoritmo de cifrado, que será el mismo de R1y R2, o sea aes en este caso: R3(config-isakmp)# encryption aes

4- Configuramos el algoritmo de autenticación que tal como hicimos en R1y R2 es pre-share: R3(config-isakmp)# Authentication pre-share

5- Definimos el tipo de Diffie-Hellman que tal como hicimos en R1 es group 5: R3(config-isakmp)# group 5

6- **configuramos la clave precompartida**. En este caso la clave va a ser también “cisco” . La dirección ip pública de destino sigue siendo la de R1 que es la 209.165.118.2, por lo que el comando será: R3(config)# **crypto isakmp key cisco address 209.165.118.2**



Fase 2 en R3

R3(config)# **crypto ipsec transform-set *R1_R3_Set esp-aes esp-sha-hmac**

R3(config)# **crypto map *R1_R3_Map 103 ipsec-isakmp**

R3(config-crypto-map)# **set peer 209.165.118.2**

R3(config-crypto-map)# **set transform-set R1_R3_Set**

R3(config-crypto-map)# **match address 103**

R3(config-crypto-map)# **exit**

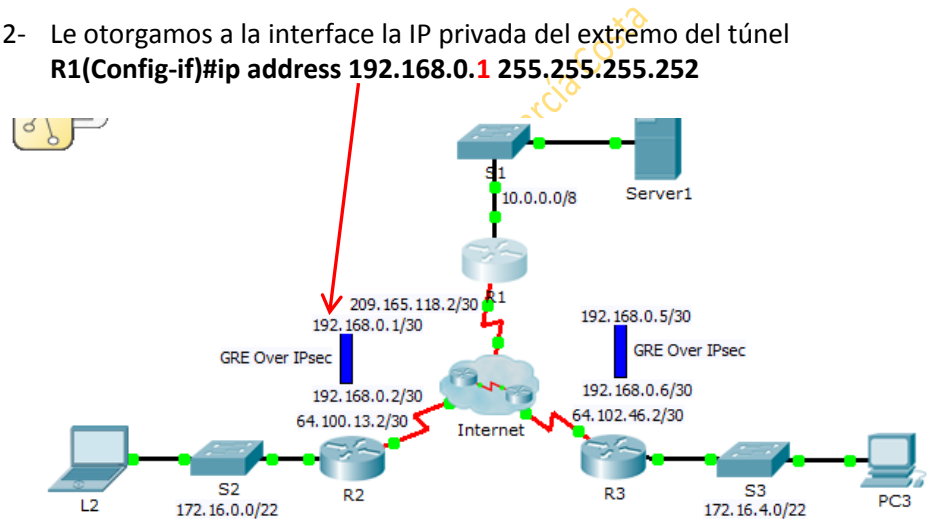
*Los nombres que hayamos puesto en R1 para el transform_set (en este caso **R1_R3_Set**) y para el crypto map (en este caso **R1_R3_Map**), deberán ser los mismos en R3. *Lo mismo se hizo en R2*. Los nombres siempre los mismos, sino, la configuración estaría mal.

Hemos visto la configuración de túnel GRE y de túnel IPsec, pero está también el modelo GREoIPSec.

a- Creamos el tunel entre R1 y R2 (*que se llamará tunnel 0*)

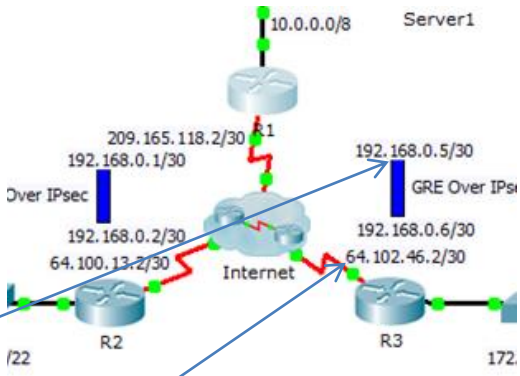
1- Nos metemos en la interface del túnel 0
R1(Config)#interface tunnel 0

2- Le otorgamos a la interface la IP privada del extremo del túnel
R1(Config-if)#ip address 192.168.0.1 255.255.255.252



3- Le decimos cual es la interface de salida del túnel:
R1(Config-if)#tunnel source s0/0/0 Así le estamos metiendo la interface/subinterface física correspondiente.
A la vez le decimos cual es la ip de destino.
R1(Config-if)#tunnel destination 64.100.13.2

4- Configuramos el túnel para transmitir el tráfico IP a través de GRE.
Lo hacemos con el comando: **R(Config-if)#tunnel mode gre ip**



b- Creamos el túnel entre R1 y R3 (*que se llamará tunnel 1*)

1- Nos metemos en la interface del túnel 1
R1(Config)#interface tunnel 1

2- Le otorgamos a la interface la IP privada del extremo del túnel
R1(Config-if)#ip address 192.168.0.5 255.255.255.252

3- Le decimos cual es la interface de salida del túnel:
R1(Config-if)#tunnel source s0/0/0
A la vez le decimos cual es la ip de destino.
R1(Config-if)#tunnel destination 64.102.46.2

4- Configuramos el túnel para transmitir el tráfico IP a través de GRE.
Lo hacemos con el comando: **R(Config-if)#tunnel mode gre ip**

*Hasta ahora en la configuración de los túneles GRE hemos configurado lo correspondiente a los extremos de R1, por lo que habrá que configurar el otro extremo del túnel gre tanto en R2 como en R3.

El túnel gre para R2 sería:

1- Nos metemos en la interface del túnel 0
R1(Config)#interface tunnel 0

2- Le otorgamos a la interface la IP privada del extremo del túnel
R1(Config-if)#ip address 192.168.0.2 255.255.255.252

3- Le decimos cual es la interface de salida del túnel:
R1(Config-if)#tunnel source s0/0/0
A la vez le decimos cual es la ip de destino.
R1(Config-if)#tunnel destination 209.165.118.2

4- Configuramos el túnel para transmitir el tráfico IP a través de GRE.
Lo hacemos con el comando: **R(Config-if)#tunnel mode gre ip**

El túnel gre para R3 sería:

1- Nos metemos en la interface del túnel 0 **no hace falta que se llame tunnel 1 como hicimos en R1. Al ser otro router le podemos seguir llamando tunnel 0*
R1(Config)#interface tunnel 0

2- Le otorgamos a la interface la IP privada del extremo del túnel
R1(Config-if)#ip address 192.168.0.6 255.255.255.252

3- Le decimos cual es la interface de salida del túnel:
R1(Config-if)#tunnel source s0/0/0

A la vez le decimos cual es la ip de destino.

R1(Config-if)#tunnel destination 209.165.118.2

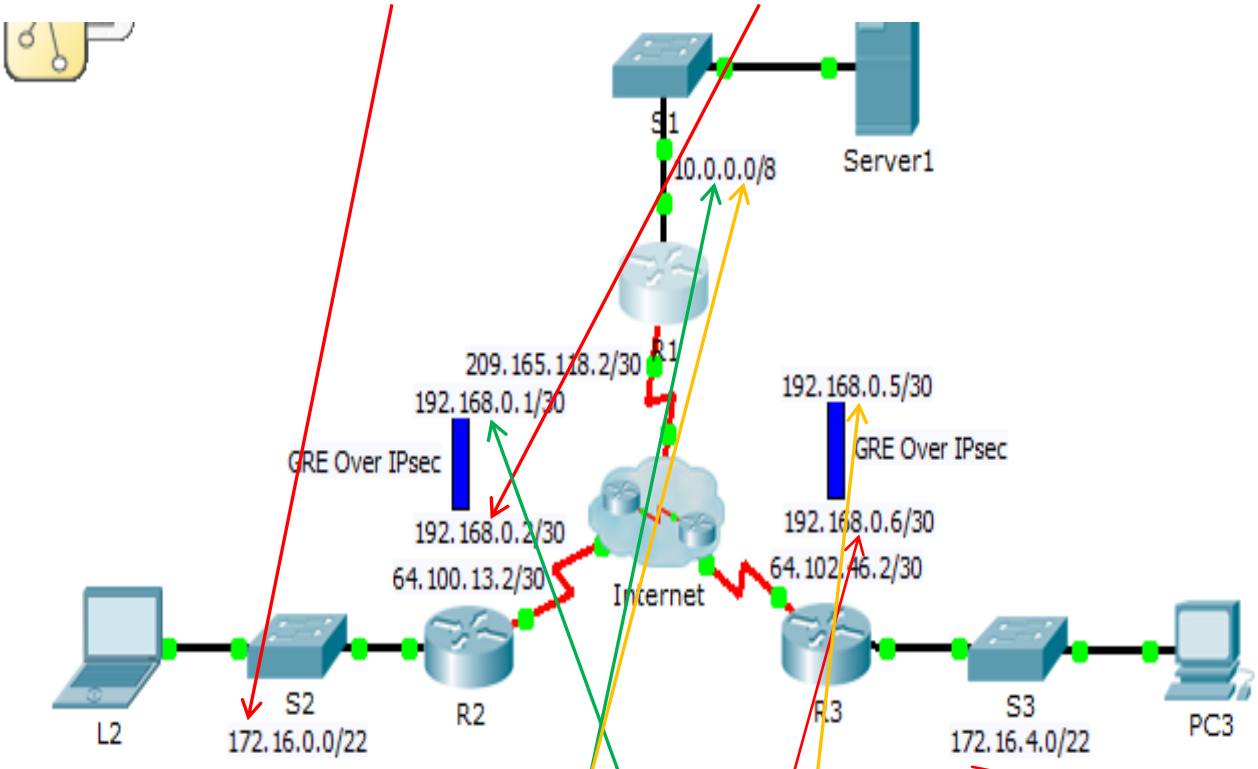
4- Configuramos el túnel para transmitir el tráfico IP a través de GRE.
Lo hacemos con el comando: **R(Config-if)#tunnel mode gre ip**

Una vez que la configuración de los túneles VPN se ha configurado, habrá que habilitar qué protocolo de enrutamiento va a pasar por ellos. Podrá ser ruta estática, o OSPF, EIGRP…etc.

En este caso vamos a habilitar rutas estáticas**:**

Para que R1 tenga una ruta estática dedicada privada con la red 172.16.0.0/22 de R2, la configuración será:

R1(Config)#ip route 172.16.0.0 255.255.255.252 192.168.0.2(ip del siguiente salto)



Y para que R1 tenga una ruta estática dedicada privada con la red 172.16.0.0/22 de R3, la configuración será:

R1(Config)#ip route 172.16.4.0 255.255.255.252 192.168.0.6(ip del siguiente salto)

Por último definiremos rutas estáticas también para que desde las redes de R2 y R3 lleguen a R1. Por tanto:

En R2: R2(Config)#ip route 10.0.0.0 255.0.0.0 192.168.0.1

Y en R3: R3(Config)#ip route 10.0.0.0 255.0.0.0 192.168.0.5