# REDES CON SWITCHES MULTICAPA

## RUTEO DE VLANS

Instructor: L. Enrique López Olguín

## MANUAL DEL PARTICIPANTE

Participante:_____

Noviembre de 2017

"Ningún mar en calma hizo experto a un marinero"

# CONTENIDO DEL CURSO

1. LA CAPA 3 DEL MODELO OSI
   1.1. Las 7 capas del modelo OSI
   1.2. La capa de red y sus dispositivos
   1.3 Proceso de entrega de paquetes en una red ruteada

2. DETERMINANDO UNA RUTA
   2.1. Rutas estáticas, dinámicas y por default.
   2.2. Tablas de ruteo.

3. RUTEO DINAMICO
   3.1. Protocolos de ruteo
   3.2. Métodos de ruteo

4. NUMERACIÓN BINARIA
   4.1. Sistema binario y decimal
   4.2 Esquemas de direccionamiento IPv4 en subredes
   4.3 Mascara de subred
   4.4 *Subnetting* y VLSM
   4.5 Sumarización de rutas con VLSM

5. INICIALIZANDO UN ROUTER
   5.1. Configurando un Router desde CLI
   4.2 Configurando direccionamiento IP en un Router

6. HABILITANDO RUTEO
   6.1. Configuración de ruteo estático
   6.2. HSRP con ruteo estático
   6.3 Ruteo dinámico con EIGRP
   6.4 Configuración de EIGRP
   6.5 NAT, PAT y rutas por default

7. IMPLEMENTACION DE VLANS Y TRONCALES
   7.1. Entendiendo una VLAN
   7.2. Aplicando direccionamiento lógico en redes *switching*
   7.3 Operación de una VLAN
   7.4 Configuración de VLANs en Cisco
   7.5 Troncales con 802.1Q
   7.6 Configuración de troncales con 802.1Q

8. RUTEO DE VLANS
   8.1. Ruteo de Vlans
   8.2. Configurando ruteo inter-vlan

9. SWITHCES MULTICAPA
   - 9.1. *Switches* L2 L3
   - 9.2 Configurando *inter-vlan routing* con MLSs
   - 9.3 Configurando interfaces de MLS
   - 9.4 Usando DHCP con *switches* multicapa.

# 1.- LA CAPA 3 DEL MODELO OSI

## EL MODEO OSI

El modelo de referencia OSI (*open system interconection*) fue creado para ayudar a definir cómo funciona el proceso de red de una manera general, incluyendo todos los componentes de una red y la transmisión de datos. Entender la estructura y propósito del modelo OSI es necesario para entender cómo se comunica un *host* con otro. Recordemos que este es un modelo de referencia que proporciona una estructura para construir protocolos y ayudar a las personas a entender el proceso que envuelve las redes de comunicación.

Sin importar que tipo de conexión, sistema operativo o servicio sea, la realidad es que para que los dispositivos se puedan comunicar, algunas reglas deben existir y así como existen reglas para establecer la comunicación, también debe existir un medio.

Por ejemplo, cualquier lenguaje tiene reglas para formar oraciones usando palabras. Este lenguaje puede ser verbal o escrito, usando el aire o un papel como medio. La mayoría de los lenguajes tienen reglas que especifican la como poner las palabras en orden y como debes ser pronunciadas y escritas.

Muchas computadoras y sistemas operativos dentro de una organización son fabricados por diferentes compañías y usan diferentes tipos de programas para operar, sin embargo, si esos sistemas se van a comunicar con otros, deben usar un conjunto de reglas en común para lograr la comunicación de datos. Estas reglas que definen la manera en que los sistemas hablan entre si son llamados protocolos.

Muchos protocolos de red pueden ser usados para establecer comunicación entre dos sistemas, y cada uno de esos protocolos proporcionan funciones muy similares. Para proporcionar una manera de establecer una comunicación utilizando reglas "abiertas" para construir un protocolo de comunicación de datos, la *Internal Organization for Standardization* (ISO) creó el modelo de referencia OSI.

El modelo de referencia OSI es el principal modelo en el ámbito de las redes de datos. Al principio el desarrollo de las LANs, MANs y WANs fue caótico en muchos sentidos. A inicios de los 80s se vio un tremendo crecimiento en el número y tamaño de las redes. Cuando las compañías se daban cuenta de que se podían ahorrar mucho dinero y aumentar productividad usando la tecnología de las redes, implementaban más redes y expandían las que ya tenían tan rápidamente como las nuevas tecnologías y productos eran presentados.

A mediados de los 80s, las compañías empezaron a experimentar dificultades con todas las expansiones que querían hacer. Se volvió más difícil para las redes que usaban diferentes especificaciones comunicarse con otras. Las compañías se dieron cuenta que necesitaban dejar a un lado los sistemas de red cerrados, aquellos sistemas que eran desarrollados de manera propietaria. Para

corregir el problema de que las redes eran incompatibles e incapaces de comunicarse una con otra, la ISO investigó diferentes esquemas de red. Como resultado de esta investigación, la ISO creó un modelo que pudiera ayudar a los fabricantes a crear redes que fueran compatibles y que pudieran operar entre sí.

"El modelo de referencia OSI nació en 1984, fue un esquema descriptivo que la ISO creó. Este proporciona a los fabricantes un conjunto de estándares que aseguran la compatibilidad y la interoperabilidad entre varios tipos de tecnologías de red desarrolladas por compañías alrededor del mundo. "

## 1.1. Las 7 capas del modelo OSI

El modelo OSI está compuesto de 7 capas y cada una tiene una función en particular. A esta separación de funciones de red por capas es llamado "*layering*".

El modelo re referencia OSI define las funciones de red que ocurren en cada capa. Pero lo más importante es que este modelo facilita el aprendizaje de cómo los datos viajan a través de la red, es decir, este modelo explica como los datos viajan desde una aplicación en un host hasta la aplicación en otro host.

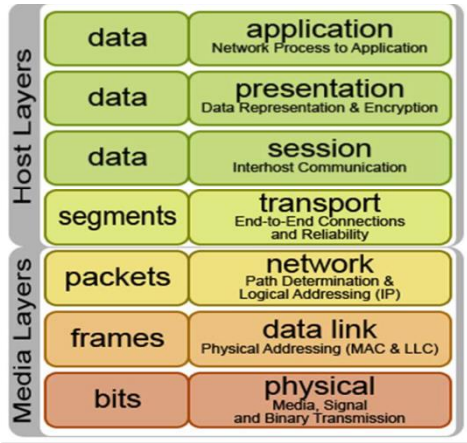



Figura 1. Las 7 capas del modelo OSI.

# 1.2. La capa de red y sus dispositivos

**"La capa de red se encarga de proporcionar conectividad y seleccionar la mejor ruta entre dos host que podrían encontrarse en redes diferentes. "**

Con el crecimiento de internet millones de usuarios se conectan alrededor del mundo diariamente y esta capa administra esta conectividad con ayuda de los dispositivos de capa 3 como herramienta

La capa 3 del modelo OSI o capa de red es la capa donde "vive" el protocolo IP o *internet protocol*. Este protocolo es el motor de internet ya que permite que la información sea dividida en porciones más pequeñas y sea enviada a través de múltiples rutas a un destino, permitiendo tener un sistema descentralizado.

Es en ésta capa es donde "viven" también protocolos de ruteo, direcciones lógicas y dispositivos que trabajando conjuntamente logran cumplir el objetivo de esta capa.

## DISPOSITIVOS DE CAPA 3

Los dispositivos de capa 3 o *Routers* proporcionan conectividad y selección de la mejor ruta entre dos host que podrían estar localizados en redes separadas geográficamente. En el caso de un host, esta es la ruta entre la capa de enlace de datos y las capas superiores.

**"Un *Router* es un dispositivo de capa 3 que funciona como una computadora de propósitos específicos y posee interfaces físicas, que usa las direcciones de capa 3 para encontrar la mejor ruta entre dos hosts utilizando protocolos de ruteo."**

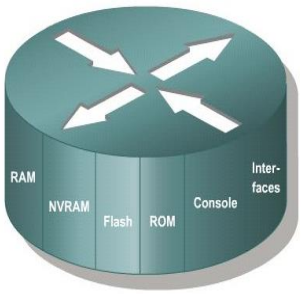



figura 2-Router.

Un *Router* o *Gateway* es un dispositivo de red que determina una ruta óptima para transmitir datos de una red a otra. el *Router* es un dispositivo especializado que corre programas y ejecuta algoritmos que ayudan en la entrega optima de trafico de red.

## CARACTERISTICAS

- Los *Routers* tienen componentes que podemos encontrar en una computadora. estos son:
    - CPU
    - *Motherboard*
    - RAM
    - ROM
    -
- los *Routers* poseen también interfaces de red, a las cuales son asignadas direcciones de red.
- Los *Routers* poseen estos tipos de puertos
    - Puerto de consola
    - puertos de red

## FUNCIONES

Determinación de la mejor ruta

Los *Routers* deben construir sus tablas de ruteo y asegurar que otros *Routers* sepan todos los cambios que hay en la red. Los *Routers* hacen esto usando protocolos de ruteo para comunicarse la información de la red entre ellos usando las tablas de ruteo. Se puede construir la tabla de ruteo de manera estática pero no es práctico en escalas grandes y además trae problemas cuando hay cambios en la topología.

Reenvío de paquetes

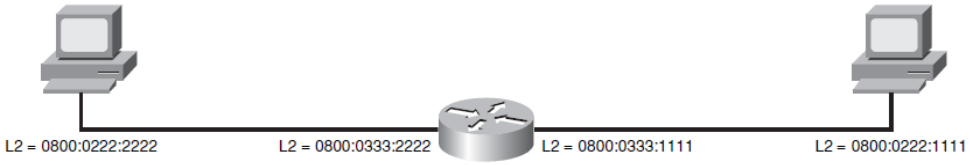
Los *Routers* usan las tablas de ruteo para determinar  a donde reenviar paquetes. Éstos reenvían paquetes a través de una interface de red hacia la red destino identificada por la dirección IP como dirección destino escrita en el paquete.

# 1.3 Proceso de entrega de paquetes en una red ruteada

Entender el proceso de entrega de paquetes es una parte fundamental para entender cómo funcionan los *Routers* dentro de una red.

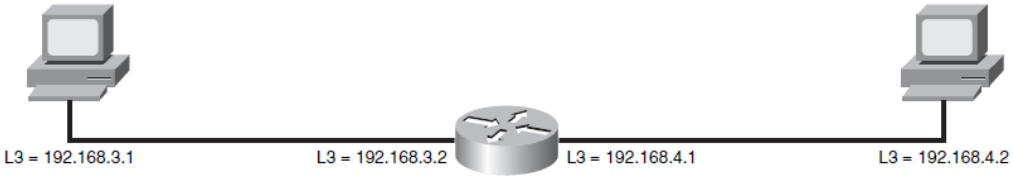
## DIRECCIONAMIENTO EN CAPA 2

Como ya sabemos, la comunicación *"host to host"* requiere direcciones de capa 2 o direcciones MAC para formar una trama *Ethernet*. Las direcciones MAC son asignadas a los dispositivos finales como lo son los *hosts*. Las interfaces físicas de los *Routers* proporcionan la funcionalidad para la capa 3 y tienen asignadas una dirección MAC. Esas direcciones son fundamentales en el proceso de entrega de paquetes. La figura de abajo muestra el direccionamiento que será usado para el ejemplo.




## DIRECCIONAMIENTO DE CAPA 3

Para poder mover datos desde una red a otra debe haber algún tipo de direccionamiento que identifique a una única red y a un único host. Para este ejemplo se usará IP como el direccionamiento de capa 3. La figura de abajo muestra el direccionamiento para cada dispositivo en toda la ruta. El *Router* tiene direccionamiento IP en cada interface. Estas serán el *Gateway* para cada host en cada red.

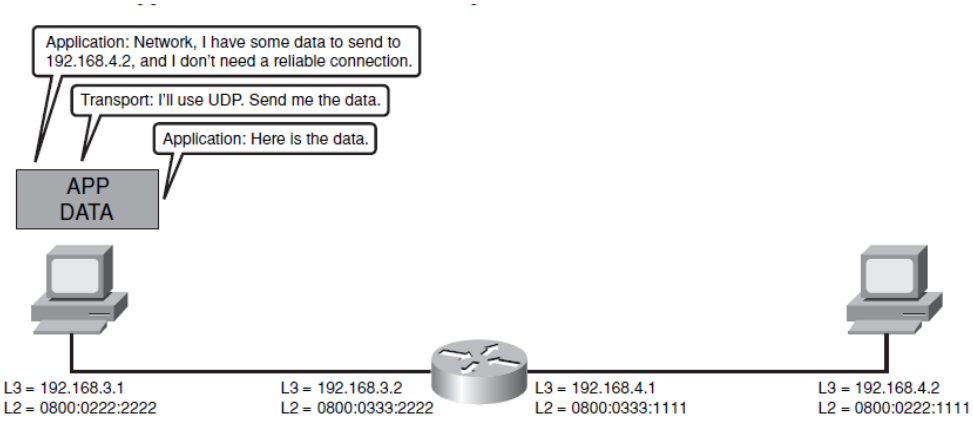


PROCESO DE ENTREGA DE PAQUETES

Los pasos para entregar un paquete en una red ruteada es similar a los pasos para entregar un paquete a través del servicio de paquetería. La clave es saber el remitente y su dirección así como el destinatario y su dirección.

Nótese que cada *host* pertenece a una red diferente así que cada uno debe tener configurado un Gateway. El Gateway es la interface del *Router* local.

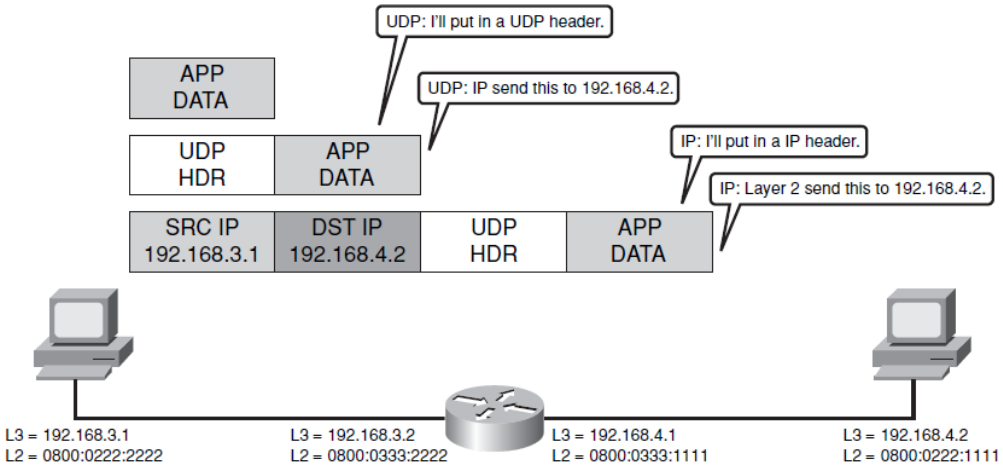
En la siguiente figura, el host 192.168.3.1 tiene datos de aplicación que quiere mandar al host 192.168.4.2. la aplicación prefiere usar UDP como transporte indicando que la confiabilidad en la entrega no es tan importante para esta aplicación.



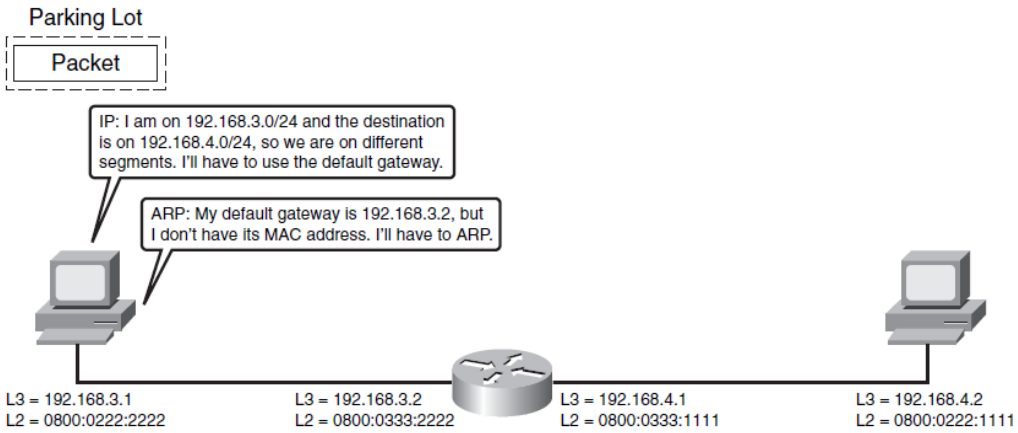

Debido a que no es necesario establecer una sesión. La aplicación puede empezar a enviar datos. UDP prepara una cabecera UDP y la pasa a IP con la instrucción de enviar el PDU a 192.168.4.2. IP encapsula el PDU en un paquete de capa 3 y lo pasa a capa 2.

Este ejemplo difiere de otros anteriores por que los paquetes (ahora tramas) en capa 2 son enviados entre host en diferentes redes: 192.168.3.0/24 y 192.168.4.0/24. Debido a que los *hosts* están configurados con una dirección IP y una máscara, se entiende que están en diferentes redes.

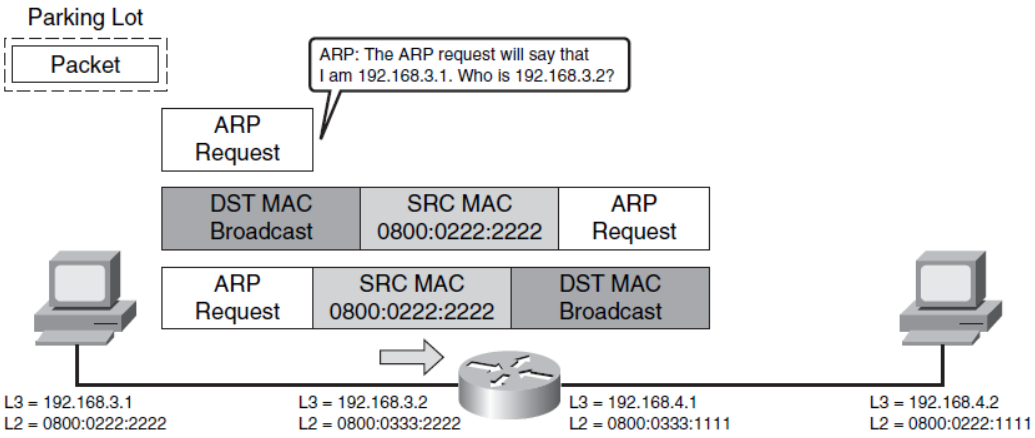
Debido a que el primer *host* no sabe cómo llegar a la red del segundo *host*, este primero debe enviar las tramas a su *default Gateway*, donde la trama puede ser reenviada. Si el *host* no tiene un mapeo de capa 2 para el *default Gateway* el *host* usa ARP para obtener el mapeo para el *Router*. Abajo se muestra el proceso.




Cuando se determina que los *host* están en diferentes redes.

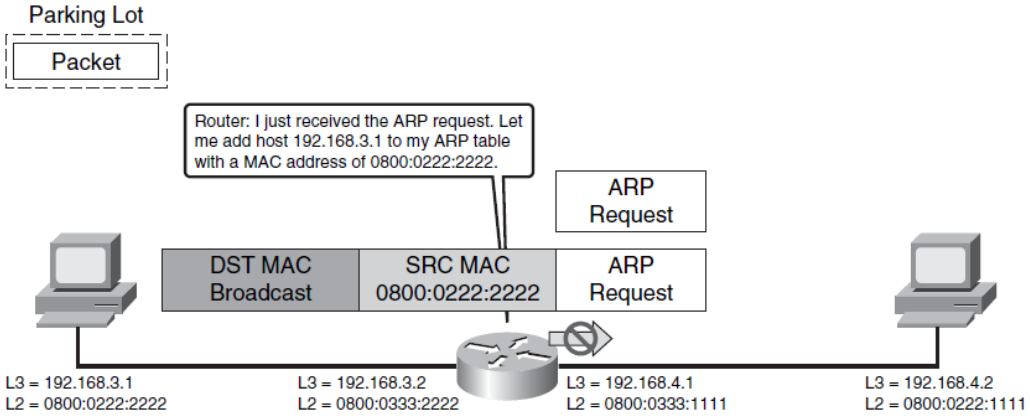



El usuario ha configurado la dirección 192.168.3.2 como el default Gateway. El *host* 192.168.3.1 envía entonces una solicitud ARP, y es recibido por el *Router*. La figura de abajo muestra la PC enviando una solicitud ARP.

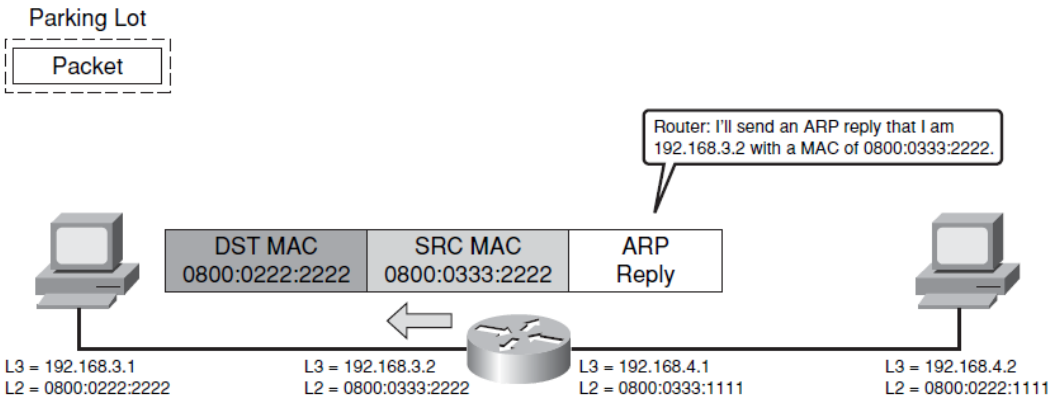



Debido a que ARP es una trama de capa 2 de tipo *broadcast*, no es enrutada a otras redes. La interface del *Router* en el segmento local reenviara la trama ARP al CPU del *Router* para ser procesada. El *Router* procesa la solicitud ARP como cualquier otro host.

L a figura de abajo muestra al Router procesando la solicitud ARP.

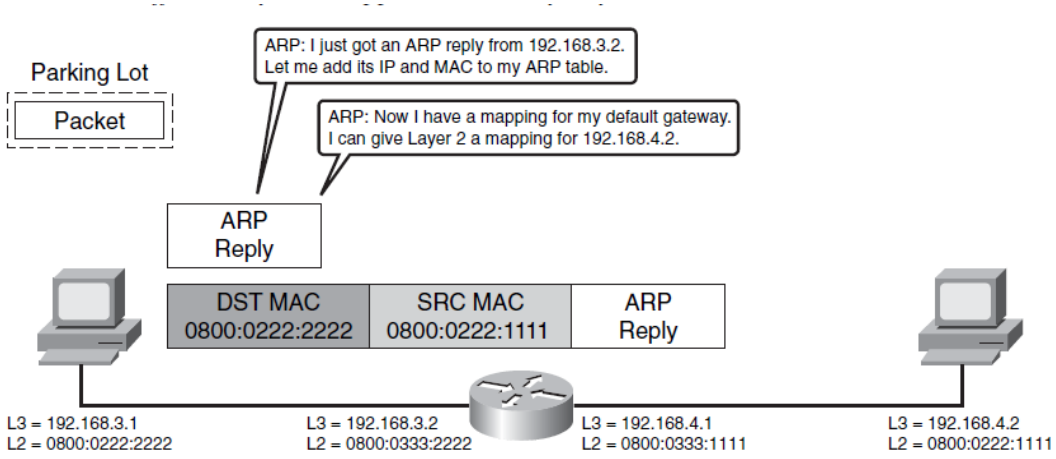



El *Router* actualizara su tabla ARP con la dirección MAC de la PC y entonces reenviara su respuesta a la solicitud ARP. Abajo se muestra.

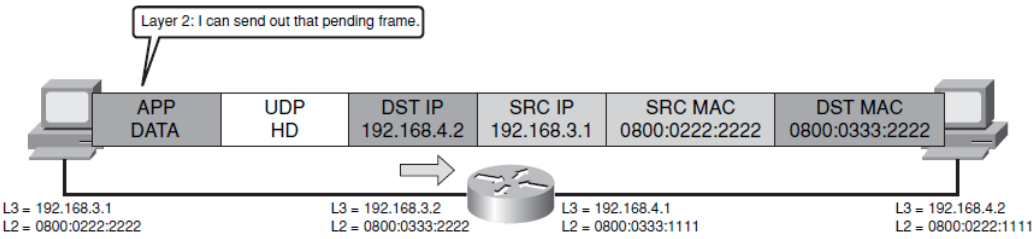



El host 192.168.3.1 recibe la respuesta ARP. Ahora puede formar una trama y enviar los datos de usuario. Debido a que el host destino está fuera de la red local, la dirección de capa 3 será mapeada a la dirección MAC de la interface del *Router* para formar la trama completa.

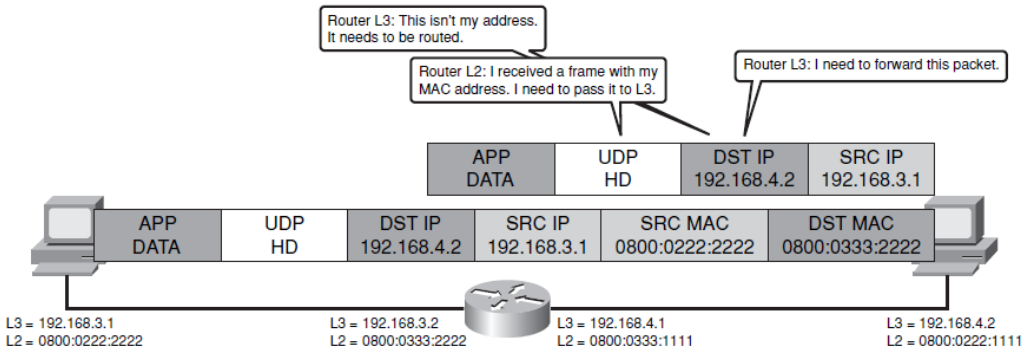
La figura de abajo muestra este paso.



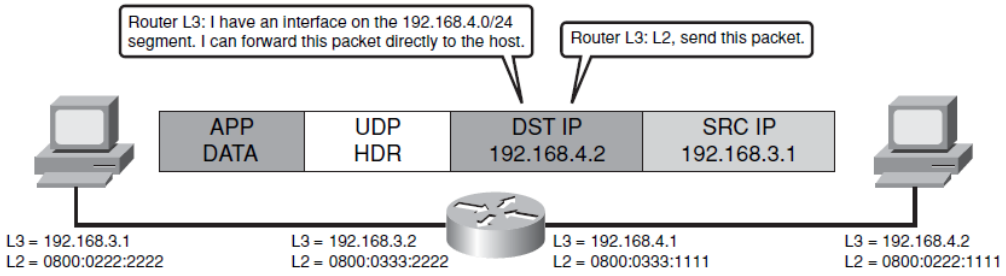

La trama pendiente es enviada con la dirección IP local y dirección MAC local de la fuente. Sin embargo la dirección IP destino es la del *host* remoto, pero la dirección MAC destino es la del *Gateway*. Abajo se muestra la trama como será enviada al *Router*.




Cuando la trama es recibida por el *Router*, el *Router* reconoce su propia dirección MAC y procesa la trama. A nivel de capa 3. El *Router* ve que la dirección IP destino no es su dirección, entonces pasa todos los paquetes que son para la red remota al proceso de ruteo. Abajo se ilustra.




El proceso de ruteo busca la dirección IP destino en la tabla de ruteo. En este ejemplo la red destino está directamente conectada. Debido a esto, el proceso de ruteo puede pasar el paquete directamente a la capa 2 para la interface apropiada.




| Destination | Next Hop | Interface |
|---|---|---|
| 192.168.3.0/24 | Connected | Fa0/0 |
| 192.168.4.0/24 | Connected | Fa0/1 |

El *Router* tendrá que enviar el paquete por la interface de salida hacia el host destino usando Ethernet. Esto requerirá que el *Router* conozca la dirección MAC destino del host. Si éste no la conoce, tendrá que usar el proceso ARP para obtener el mapeo de la dirección IP y MAC del destinatario.

# 2.- DETERMINANDO UNA RUTA


**"Durante la determinación de la mejor ruta para reenviar datos a través de la red, los *Routers* evalúan las rutas disponibles hacia los destinos remotos. "**

A continuación veremos cómo es que los *Routers* determinan la ruta más eficiente para el reenvío de paquetes.

## DETERMINANDO LA MEJOR RUTA

Puede haber diferentes rutas para llegar a la red destino, esas rutas pueden tener diferentes características como por ejemplo, latencia, tipo de medio, etc. el propósito de un *Router* es comunicarse con otros *Routers* para aprender y elegir la mejor ruta.

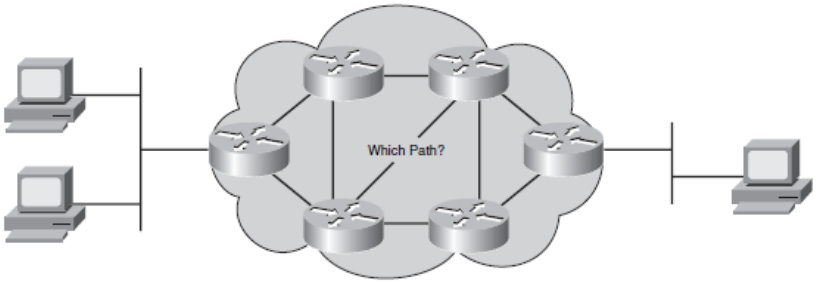



figura 3. selección de ruta

Estas rutas están almacenadas en las tablas de ruteo dentro del software del *Router* y son usadas para determinar a donde enviar un paquete basándose en la dirección destino. existen tres tipos de entradas para las tablas de ruteo para determinar la mejor ruta de un origen a un destino.

# 2.1. Rutas estáticas, dinámicas y por default.

Existen diferentes maneras de construir la mejor ruta que seguirán los paquetes desde un origen hasta un destino. Estas maneras dependen de las necesidades de la red y del administrador de red. A continuación se mencionan las tres formas de construir la mejor ruta a una red destino.

## RUTAS ESTATICAS

Este tipo de ruteo requiere que el administrador de la red manualmente introduzca la información de ruteo dentro de las tablas de ruteo para construir el camino o caminos que los paquetes seguirán. En las rutas estáticas el administrador debe de ir a cada uno de los *Routers* de la red para introducir la información de red que debe de llevar cada dispositivo en sus tabas de ruteo. Esta forma es un poco imprâctica cuando se trata de redes con un tamaño considerable debido al trabajo manual del administrador. Su principal ventaja es que con este método el *Router* ahorra trabajo de procesamiento.

## RUTAS DINAMICAS

Este tipo de ruteo construye las tablas de ruteo de manera dinámica en cada uno de los *Routers*, usando información de ruteo obtenida de otros *Routers* y que es compartida entre ellos gracias a los protocolos de ruteo. Este tipo de ruteo se usa en redes grandes ya que todo el proceso de llenado de la tablas de ruteo es de manera automática. El trabajo de un administrador disminuye de manera considerable, ya que solo se configura en cada *Router* el protocolo que se usará y las redes que participaran. Algunos protocolos de ruteo más comunes son RIP, RIPv2, OSPF, EIGRP.

## RUTAS POR DEFAULT

Este tipo de rutas se usan principalmente para conectar una red *stub* a internet, y se configura por lo general en conjunto con el método de ***NAT overloading***. Este tipo de enrutamiento reemplaza la necesidad de mantener una ruta explícita a cada red. La entrada de una ruta predeterminada puede configurarse o aprenderse de forma estática desde un protocolo de enrutamiento dinâmico.

# 2.2. Tablas de ruteo.

Como parte de determinar la mejor ruta, en el proceso de ruteo, el *Router* construye una tabla de ruteo de manera estática o dinámica. Esta tabla contiene información de las redes que un *Router* tiene configurada en sus interfaces y contiene también la información de otras redes y a través de que interfaces las conoce o aprendió. Esta información es configurada por el administrador o recabada de otros *Routers* con la ayuda de protocolos de ruteo. De esta manera cada *Router* sabe cómo alcanzar cada red hasta llegar a la red destino. A continuación se muestra el ejemplo de una tabla de ruteo.

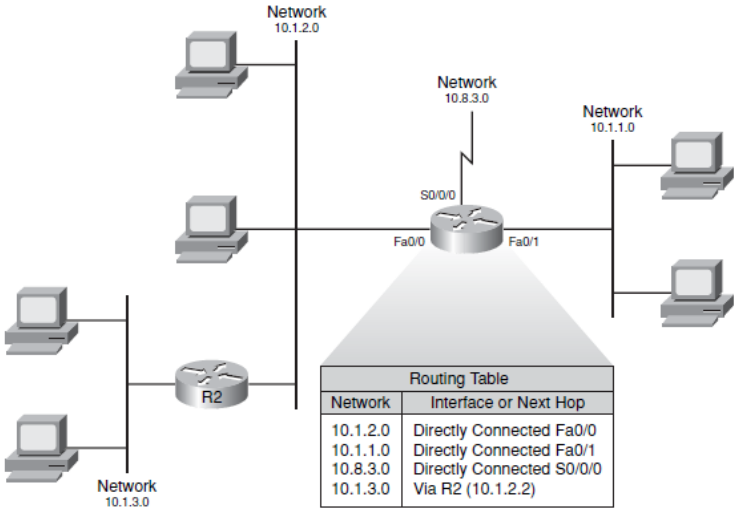



figura 4.- Tabla de ruteo

INFORMACION DE LA TABLA DE RUTEO

**"Una tabla de ruteo consiste en una lista ordenada de redes conocidas, estas redes han sido aprendidas dinámica o estáticamente. "**

Las tablas de ruteo contienen información de las redes que un *Router* tiene directamente conectadas e incluso de las redes que el *Router* puede alcanzar a través de otro *Router*, llamado "*Next hop Router*". Cuando un *Router* recibe un paquete de entrada, éste lee la dirección destino y consulta su tabla de ruteo para encontrar la mejor ruta. si no se puede encontrar ninguna entrada para esa dirección destino en la tabla de ruteo, el Router descarta el paquete.

# 3.- RUTEO DINAMICO

Dentro del ruteo dinámico podemos encontrar protocolos de ruteo dinámico, que son los responsables de que los *Routers* se comuniquen entre sí para poder compartir información de la red completa y así poder construir la mejor ruta desde una red origen hasta una red destino. Los protocolos de ruteo más usados son:

**RIP**: *Routing Information Protocol*
**RIPv2**: Evolución de RIP
**OSPF**: *Open Shortest Path First*
**EIGRP**: *Enhanced Interior Gateway Routing Protocol*

Cada protocolo de ruteo utiliza un método de ruteo para poder comunicar un *Router* con otro y compartir la información de red. Estos métodos se mencionan más adelante.

## 3.1 Protocolos de ruteo

Algunos protocolos de ruteo usan sus propias reglas y métricas para construir y actualizar tablas de ruteo de manera automática. Éstos protocolos son conocidos como protocolos de ruteo dinámico por que pueden ajustar de manera dinámica los cambios en la topología de red.

### METRICAS DE RUTEO

Cuando un protocolo de ruteol actualiza una tabla de ruteo, el objetivo principal del protocolo es determinar la mejor información para incluirla en la tabla de ruteo. Los algoritmos de ruteo generan un número llamado métrica para cada ruta en la red. Los protocolos de ruteo sofisticados basan su selección de la mejor ruta en múltiples métricas combinándolas en una sola métrica. Generalmente mientras menor sea el numero métrico, mejor será el camino. A continuación se muestran las métricas de ruteo.

Una métrica puede ser una sola característica o varias características de una ruta. Las métricas que son más comúnmente usadas por los protocolos son:

**BANDWIDTH**: es la capacidad del enlace.

**DELAY**: la cantidad de tiempo requerida para mover un paquete desde el origen hasta el destino. depende de enlaces intermedios, congestión de la red o de la distancia física.

**HOP COUNT**: Es el numero de Routers que el paquete tiene que atravesar antes de alcanzar el destino.

**COSTO**: Es un valor arbitrario asignado por el administrador de red o por el mismo sistema operativo. Es usualmente basado en el ancho de banda y preferencia del administrador.

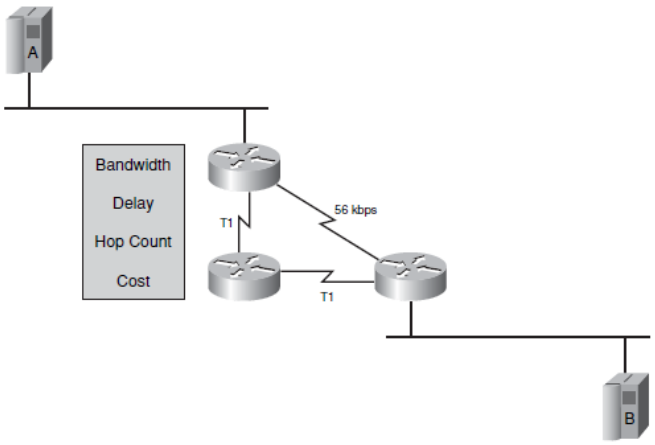



figura 5.- Métricas

## 3.2 Métodos de ruteo

Además de las métricas utilizadas para seleccionar rutas, también hay métodos para los protocolos de ruteo. La mayoría de los protocolos de ruteo están diseñados con unos de los siguientes métodos:

- *DISTANCE VECTOR*
- *LINK- STATE*

# RUTEO *DISTANCE VECTOR*

Con este método de ruteo, un *Router* no tiene que conocer el ruta completa a cada segmento de red; El *Router* sólo tiene que conocer el vector o la dirección por donde enviar el paquete. El método de ruteo vector distancia determina la dirección (Vector) y la distancia (saltos) a cualquier red.

**"*Distance Vector* envía notificaciones periódicas de sus tablas de ruteo a sus vecinos adyacentes cada 30 segundos por default, aunque no haya modificaciones en la red."**

los *Routers* que corren algún protocolo con el método vector distancia envían actualizaciones periódicas, aun si no ha habido cambios en la topología de red. Al recibir la tabla de ruteo de un vecino, el Router puede verificar todas las rutas conocidas y hacer cambios a su tabla de ruteo local basándose en las actualizaciones recibidas de sus vecinos.

Éste método es conocido como "ruteo por rumor" debido a un Router conoce la topología de red pero solo desde la perspectiva de la tabla de ruteo de sus vecinos.

En la figura de abajo se muestra como se determinan rutas en el método vector distancia.

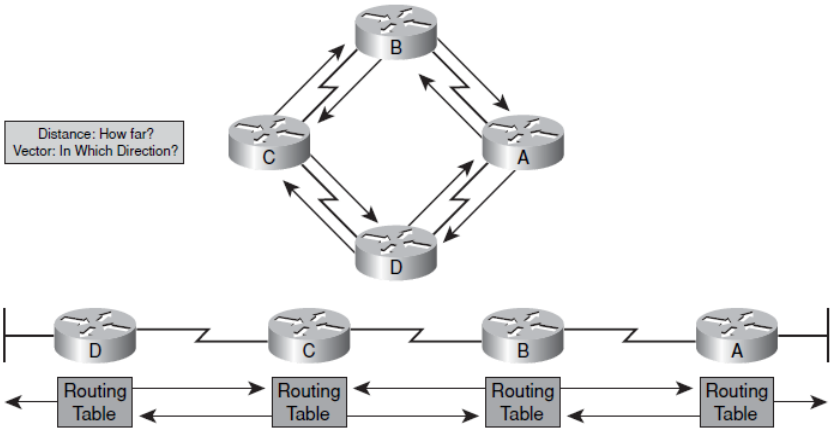
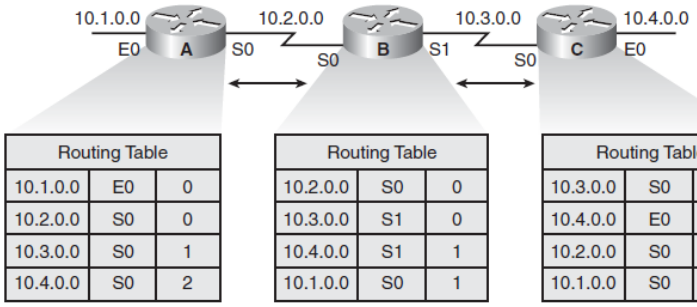



figura 6. *Distance Vector*

De esta manera el algoritmo acumula saltos (*Routers*) y los indica en una base de datos de la información de la topología. en vector distancia no es posible que un Router conozca la topología exacta de la red.

Por ejemplo:

El *Router* A aprende otras redes basándose en la información que recibe de el Router B. cada unas de las entradas en la tabla de ruteo tiene numero de saltos acumulados e indica que tan lejos está la red destino en la dirección indicada.




| Routing Table | | |
|---|---|---|
| 10.1.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S0 | 1 |
| 10.4.0.0 | S0 | 2 |

| Routing Table | | |
|---|---|---|
| 10.2.0.0 | S0 | 0 |
| 10.3.0.0 | S1 | 0 |
| 10.4.0.0 | S1 | 1 |
| 10.1.0.0 | S0 | 1 |

| Routing Table | | |
|---|---|---|
| 10.3.0.0 | S0 | 0 |
| 10.4.0.0 | E0 | 0 |
| 10.2.0.0 | S0 | 1 |
| 10.1.0.0 | S0 | 2 |

figura 7.- Tablas de ruteo

Cuando la topología de la red usando un protocolo en vector distancia cambia, debe de ocurrir una actualización en las tablas de ruteo.

El algoritmo en vector distancia hace que cada *Router* envíe su tabla de ruteo completa a su vecino directamente conectado. estas tablas de ruteo incluyen información acerca de el costo de la ruta (definido por la métrica) y la información de la dirección lógica de cada *Router* en la ruta para cada red que conoce.

## RUTEO *LINK- STATE*

En el ruteo "estado de enlace" cada Router trata de construir su propio mapa interno de la topología de la red. cada *Router* envía mensajes dentro de la red cuando se vuelve activo, escuchando los *Routers* a los cuales está directamente conectado y proporcionando información acerca de si el enlace de cada *Router* está activo. Los otros *Routers* usan esa información para construir un mapa de la topología de red y entonces, teniendo el mapa se puede escoger el mejor camino.

*Link State* responde rápidamente a los cambios de la red, enviando actualizaciones instantáneas cuando ha ocurrido un cambio en la red, enviando también actualizaciones periódicas (*refreshes*) en intervalos de tiempo largos. Cada 30 minutos.

Cuando la topología cambia, el dispositivo que detecta el cambio crea mensajes de actualización respecto a ese enlace (ruta), y esos mensajes de actualización son propagados a todos los *Routers* (corriendo el mismo protocolo). Cada *Router* toma una copia de esa actualización, actualiza su tabla de ruteo y reenvía a todos los demás *Routers*.

**Esta "inundación" del mensaje de actualización es necesaria para asegurar que todo los *Routers* actualicen su base de datos topológica antes de crear su tabla de ruteo actualizada que contenga las mejores rutas.**

A continuación se muestra la forma en que el método *link state* determina las rutas. Un ejemplo de un protocolo de ruteo que usa este método es OSPF (*open shortest path first*).

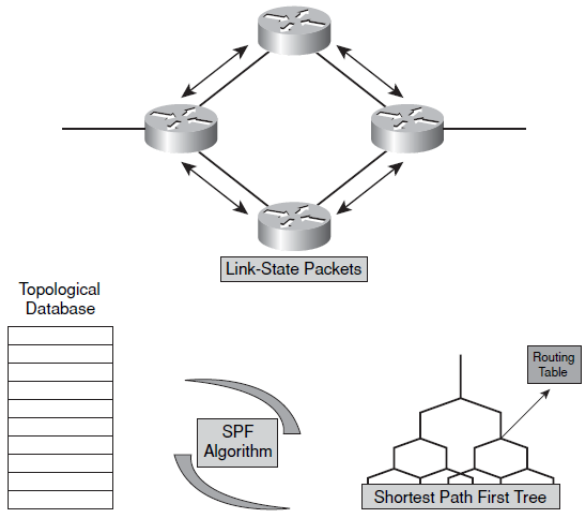



figura 8.- Método *Link State*

# 4.- NUMERACIÓN BINARIA

Todas las computadoras funcionan usando un sistema binario, es decir, de cambios de estados. estos estados pueden ser "prendido" o "apagado" como si fuese el interruptor de la luz en una casa. estos estados son representados con dos valores con los dígitos "1" y "0". estas representaciones son conocidas como valores binarios.

"Un dispositivo de red usa este sistema binario para definir su ubicación en la red. Las direcciones IP están representadas en notación binaria, es decir 1s y 0s pero se traducen a notación decimal para que los humanos puedan leerlas y escribirlas. "

## 4.1. Sistema binario y decimal

El sistema decimal (base 10) es un sistema numérico usado diariamente en las cuestiones matemáticas de las personas, mientras que el sistema binario (base 2 ) se usa en todas las operaciones que involucran una computadora.

En el sistema decimal los dígitos son 0,1,2,3,4,5...9. cuando se requiere una cantidad mayor de 9, la numeración decimal empieza desde 0 nuevamente anteponiendo el número anterior, por ejemplo 10, y continua hasta el 99.

El sistema binario utiliza solamente los números 0 y 1. por eso, el primer digito es 0 seguido del 1. si una cantidad mayor se necesita, el sistema binario se va hasta 10 y de ahí al 11. el sistema binario continua con 100, 101, 110, 111, 1000 etc. abajo se muestra unas equivalencias entre base 10 y base 2.

| Decimal Number | Binary Number |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 10 |
| 3 | 11 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

figura 9.- correspondencia de sistemas

## MSB (*MOST SIGNIFICANT BIT*) Y LSB (*LEAST SIGNIFICANT BIT*)

Mientras que la base numérica es importante en un sistema numérico, es importante también la posición de un digito ya que es el que confiere un valor. El numero 10 está representado por un 1 en la posición de las decenas y un cero en las la posición de las unidades. el numero 100 está representado por el numero 1 en el lugar de las centenas mientras que un numero 0 en las otras dos posiciones.

En los números binarios, el digito en el lado más a la derecha es el bit menos significativo (LSB) o de menor peso, y el digito mas a la izquierda es el bit más significativo (MSB) o de mayor peso.

La figura de abajo muestra el significado de la posición de los bits.

Base-10 Decimal Conversion - 63204829

| | MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|
| Base$^{Exponent}$ | $10^7$ | $10^6$ | $10^5$ | $10^4$ | $10^3$ | $10^2$ | $10^1$ | $10^0$ |
| Column Value | 6 | 3 | 2 | 0 | 4 | 8 | 2 | 9 |
| Decimal Weight | 10000000 | 1000000 | 100000 | 10000 | 1000 | 100 | 10 | 1 |
| Column Weight | 60000000 | 3000000 | 200000 | 0 | 4000 | 800 | 20 | 9 |

60000000 + 3000000 + 200000 + 0 + 4000 + 800 + 20 + 9 = 63204829

Base-2 Binary Conversion - 1110100 (233)

| | MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|
| Base$^{Exponent}$ | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| Column Value | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| Decimal Weight | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Column Weight | 128 | 64 | 32 | 0 | 8 | 0 | 0 | 1 |

128 + 64 + 32 + 0 + 8 + 0 + 0 + 1 = 233

figura 10.- Posición de los bits

Es importante entender el sistema de base 2 ya que el direccionamiento IPV4 consiste en una dirección formada por 32 dígitos del sistema binario. Cada digito es un bit. Los 32 bits están divididos en 4 grupos de 8 bits cada uno, es decir, en 4 octetos.

Cuando representamos una dirección IP lo hacemos por separado, ya que para convertirla a un valor decimal, es más fácil convertir por partes a convertir 32 bits juntos.
En una dirección IP el valor más alto que podemos encontraren un octeto es 11111111 el cual si lo representamos en decimal es igual a 255. El valor más bajo que se puede tener es 00000000 que es igual a un valor decimal de 0. Esto significa que se pueden tener 256 diferentes valores en un octeto, de 0 a 255.

# 2 ELEVADO A SUS POTENCIAS

Para entender como son usados los números binarios en el direccionamiento IP, se debe entender el proceso matemático de convertir del sistema binario a decimal y vice versa.

Para poder hacer conversiones solo es necesario convertir el número dos a las diferentes potencias hasta completar el octeto (8 posiciones) empezando desde el la potencia 0 en el lugar del LSB.

Esto se hace de la siguiente manera:

| Power of 2 | Calculation | Value |
|---|---|---|
| $2^0$ | Mathematical identity | 1 |
| $2^1$ | 2 | 2 |
| $2^2$ | 2*2 | 4 |
| $2^3$ | 2*2*2 | 8 |
| $2^4$ | 2*2*2*2 | 16 |
| $2^5$ | 2*2*2*2*2 | 32 |
| $2^6$ | 2*2*2*2*2*2 | 64 |
| $2^7$ | 2*2*2*2*2*2*2 | 128 |

figura 11.- valores en un octeto

# CONVERSION DE DECIMAL A BINARIO

Los nueros decimales se pueden convertir a sistema binario de la siguiente manera:

| Base$^{Exponent}$ | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|---|
| Place Value | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Example: Convert Decimal 35 to Binary | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

35 = $2^5$ + $2^1 + 2^0$

35 = (32 * 1) + (2 * 1) + (1 * 1)

35 = 0 + 0 + 1 + 0 + 0 + 0 + 1 + 1

35 = 00100011

# CONVERSION DE BINARIO A DECIMAL

Se puede convertir números binarios a decimal usando los valores de acuerdo a su posición basándonos en el valor que tiene el numero 2 con su respectiva potencia.

| Base$^{Exponent}$ | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|---|---|---|---|
| Place Value | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Example: Binary Number | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| Decimal Number Total: 185 | 128 | 0 | 32 | 16 | 8 | 0 | 0 | 1 |

1 0 1 1 1 0 0 1 = (128 * 1) + (64 * 0) + (32 * 1) + (16 * 1) + (8 * 1) + (4 * 0) + (2 * 0) + (1 * 1)
1 0 1 1 1 0 0 1 = 128 + 0 + 32 +16 + 8 + 0 + 0 + 1
1 0 1 1 1 0 0 1 = 185

EJERCICIOS

Nombre del participante_____ RPE_____

## Convertir de binario a decimal

10: _____
10110010: _____
1100: _____
10101100: _____
11001000: _____

## Convertir de decimal a binario

89 : _____
159:_____
16:_____
254_____

## Expresar la siguiente dirección IPV4 en binario

172.12.17.14 _____

## Convierte la siguiente dirección IP en decimal e indique si es pública o privada.

10111011110101011100111010111000_____

# 4.2. Esquemas de direccionamiento IPv4 en subredes

Las subredes (*subnetwork* o *subnet*) son ambientes de redes más pequeñas, ya que se segmenta la red en redes más pequeñas que tienen su propio direccionamiento. Para crear direccionamiento para subredes, algunos de los bits usados para la porción de host dentro de una dirección IP son "prestados" para crear subredes. A continuación se describe cómo funcionan las subredes y como se forman.

## SUBREDES

**"Los administradores de red a menudo necesitan dividir redes, especialmente redes grandes, en subredes para proporcionar flexibilidad de direccionamiento."**

Una compañía que ocupa un edificio de tres pisos podría tener una red dividida por pisos, y en cada piso, oficinas. Pendemos que el edificio es la red, los pisos son las tres subredes y las oficinas son las direcciones de host.

Una subred segmenta los host dentro de una red. Sin subredes, las redes tienen una topología plana. Una topología plana tiene una tabla de ruteo corta y depende de las direcciones de capa 2 para entregar paquetes. Las direcciones MAC no tienen una estructura jerárquica. Conforme la red va creciendo, el uso de ancho de banda de la red se vuelve menos y menos eficiente.

Las desventajas de una red plana son:

- Todos los dispositivos comparten el mismo ancho de banda
- Todos los dispositivos comparten el mismo dominio de broadcast
- Es difícil de aplicar políticas de seguridad por no hay fronteras entre los dispositivos.

En una red Ethernet conectada por *switches*, los *hosts* ven todo el trafico *broadcast*. En situaciones de mucho tráfico las redes podrían saturarse hasta llegar a un colapso. Para los usuarios, esto es percibido como una red lenta. Una red con subredes en estas situaciones separan las redes partiéndolas en múltiples subredes, cada una con su propio segmento de red.

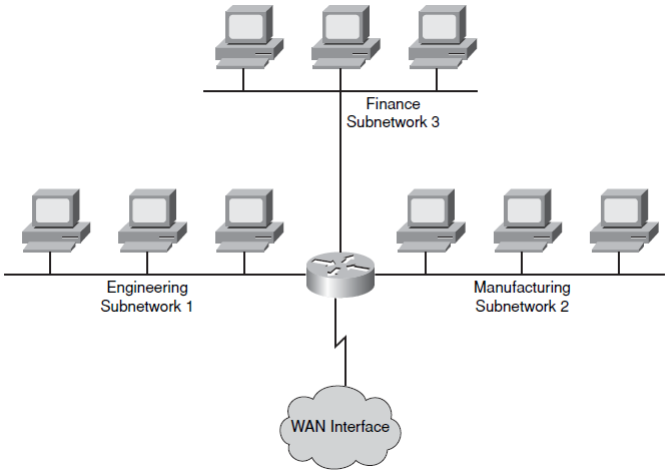



figura 12.- subredes

las ventajas de subnetear una red son las siguientes:

- Redes más pequeñas son más fáciles de administrar.
- El tráfico que circula por cada subred se reduce, lo cual mejora el rendimiento.
- Se pueden aplicar medidas de seguridad ya que están bien definidas las fronteras de cada subred.

## CREACIÓN DE UNA SUBRED


**El direccionamiento de una subred es creado tomando bits "prestados" de la porción de *host* de la clase A, B y C y pasándolos a la porción de red.**

Por los regular, un administrador de red asigna los segmentos de red de manera local. Así como una dirección IP, cada direccionamiento de una subred debe ser único.

Cuando hablamos de bits "prestados" de la porción de *host*, es importante notar que el numero de subredes adicionales que se van a crear se duplicaran por cada bit que sea prestado. Prestando un bit podemos crear dos posibles subredes $(2^1 = 2)$. Prestando dos bits podemos tener cuatro posibles subredes $(2^2 = 4)$. Prestando tres bits podemos tener 8 posibles subredes $(2^3 = 8)$, y así sucesivamente.

Cada vez que un bit es prestado desde la porción de los *hosts* a la porción de red, el número de subredes se incrementa por el numero 2 elevado a sus potencias. El número de posibles hosts se decrementa en cada subred.

## CALCULOS DE SUBREDES Y *HOSTS*

Una de las decisiones que se deben tomar cuando se crean subredes es determinar el número de subredes y *hosts* optimo. Para lograr esto, se necesita entender las clases de red y como usar los bits dentro de cada clase para crear redes y alojar direccionamiento suficiente para los hosts.

Cálculos para una clase C

Cada vez que un bit es prestado de la porción de *host* hay un bit menos que puede ser usado para el numero de *hosts*, así que el numero de *host* se decrementa por la potencia del 2.

Ejemplo.

Considere una red de clase C en el cual los 8 bits en el ultimo octeto son usados para los *hosts*. Por eso, hay 256 posibles números de *host*. Realmente el numero de *host* usables son 254, debido a las dos direcciones reservadas (id y broadcast). $\mathbf{2^n - 2 = hosts}$.

Ahora imagine que esta clase C se divide en subredes. Si 2 bits son prestados de los 8 bits que tiene por default el ultimo octeto, el campo de los *host* disminuye a 6 bits. Todas las posibles combinaciones de 0s y 1s que pueden ocurrir, determinan el número de subredes y el número de *host* por subred. El número original de *host* que era de 256 *host* disminuye a 64 ( ya que ahora tenemos 6 bits para *hosts*) y el número de subredes se incrementa a 4. Entonces ahora tendremos 4 subredes de 64 *host* cada una.

EJERCICIOS

Nombre del participante_____ RPE_____

DETERMINE EL NUMERO DE SUBREDES Y *HOSTS* SI SE PRESTAN 3 BITS EN UNA RED CLASE B.

# 4.3 Mascara de subred

"La máscara de subred es una combinación de bits (Os y 1s) que sirve para indicar a los dispositivos de red (Routers) qué parte de la dirección IP corresponde la red a la que pertenece esa dirección, incluyendo la subred, y qué parte es la correspondiente al *host*."

## COMO USAN LA MASCARA DE SUBRED LOS *HOSTS*

Los dispositivos finales usan la máscara de subred para comparar en la porción de red, si la dirección de red destino es local. Antes de que un *host* envíe un paquete a su destino, primero debe determinar si la dirección destino está en la red local. Esto lo hace comparando los bits en la porción de red en la dirección destino con los bits de la porción de red de la dirección origen.
En la figura de abajo se observa que el *host* A y el B están en la misma red local debido a la porción de red de su dirección IP.

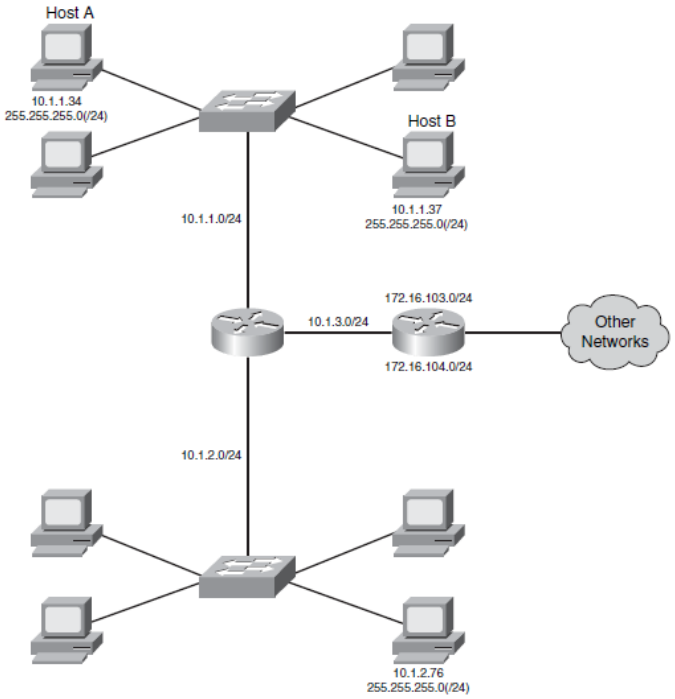



figura 13.- *hosts* locales

Debido a que los dos *hosts* están en la misma red, los *hosts* usan ARP para enlazar la dirección IP destino con la dirección MAC destino. Si no estuvieran en la misma red, el paquete debe ser reenviado a la dirección MAC del *Gateway* ( interface del *Router*) para que sea transmitida a la red destino.

## COMO USAN LA MASCARA DE SUBRED LOS *ROUTERS*

La máscara de subred identifica la parte de red de una dirección IP. Los *Routers*, como los hosts, necesitan esta información para determinar cómo reenviar un paquete hasta el destino deseado. Cuando un dispositivo determina que un paquete no pertenece a la red local, este lo debe reenviar al *Router* (a su *default Gateway*) en su red. El *Router* debe determinar hacia donde reenviar el paquete.

Todos los *Routers* tienen tablas de ruteo, dependiendo de la ubicación del *Router* en la red jerárquica, la tabla puede ser pequeña y simple o grande y compleja.

La figura de abajo muestra un paquete viajando desde *host* A hasta *host* B en redes diferentes.

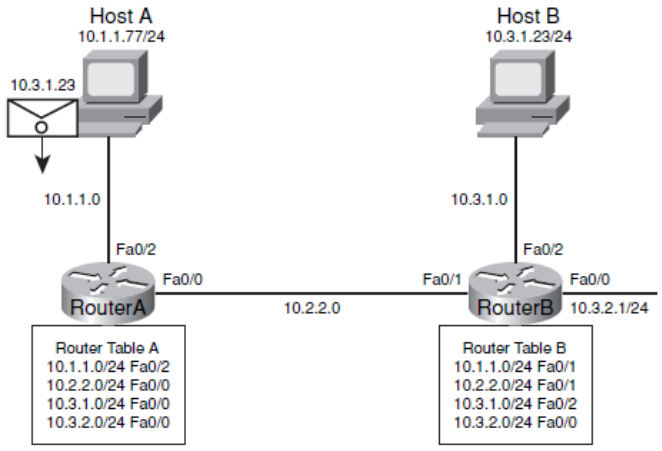



figura 14.- Tablas de ruteo

Los *Routers* llenan sus tablas de ruteo con todas las redes que tienen directamente conectadas o que has aprendido, para comparar las direcciones de la red destino del paquete que necesita ser reenviado. Si la red no está directamente conectada al *Router*, el *Router* almacena la dirección del próximo salto a la cual el *Router* debería mandarla.

La figura de arriba se describe a continuación.

**Paso 1.** El *host* A determina que la red destino no está dentro de la red local y requiere hacer uso del *default Gateway* (*Router* A). Así que el *host* A de debe establecer ARP con el *Router* A y entregarle la trama.

**Paso 2.** Debido a que la red 10.3.1.0/24 está directamente conectada al *Router* B en la interface fa0/2, el *Router* B usara ARP para determinar la dirección MAC del *host* B.

Cuando se configuran los *Routers*, cada interface está conectada a diferentes redes o subredes. Una dirección de host (usable) disponible de cada segmento se configura a cada interface del *Router* que corresponda a la red o subred.

La figura de abajo muestra como el *Router* tiene dos interfaces, una conectada al segmento 172.16.2.0 y la otra al 172.16.3.0. sin embargo las interfaces tienen configurada una dirección usable disponible.

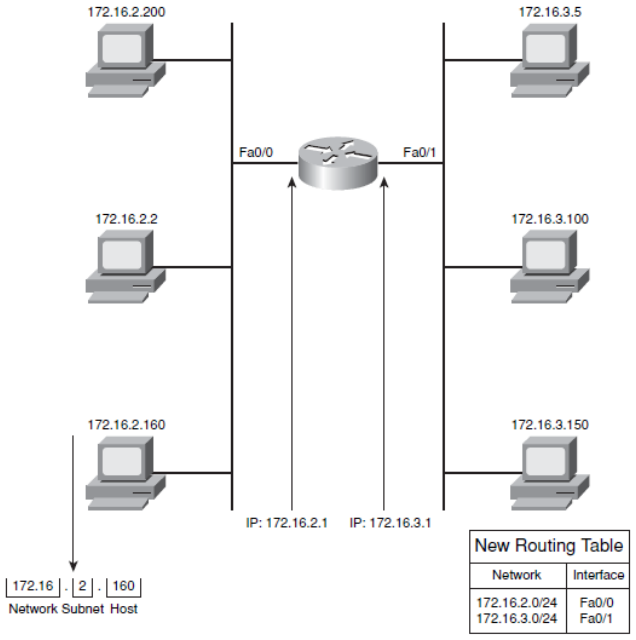



figura 15- Direccionamiento en las interfaces del *Router*

# OPERACIÓN DE LA MASCARA DE SUBRED

Aunque la máscara de subred usa el mismo formato de una dirección IP, no es una dirección IP como tal. Cada mascara de subred tiene 32 bits divididos en cuatro octetos, y está representada en notación decimal como una dirección IP.

**"En notación decimal la máscara de subred está representada con 1s en la porción de red y Os en la porción de host."**

Hay solo 8 valores validos de la máscara de subred por octeto para una subred. El campo de subred siempre va después del campo de red. Esto es, los bits prestados deben ser los primeros bits de la izquierda seguidos por los bits de host o Os en el o los octetos de host. La máscara de subred es la herramienta usada por los Routers para determinar que bits sin de red y subred y que bits de host.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | = | 128 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | = | 192 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | = | 224 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | = | 240 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | = | 248 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | = | 252 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | = | 254 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | = | 255 |

figura 16- mascaras de subred

Por ejemplo: teniendo la red 10.0.0.0, se toman 3 bits prestaos para obtener subredes.

10.0.0.0/11 = 11111111.11100000.00000000.00000000

Esta dirección se representa con una máscara en un numero decimal como 255.224.0.0

**La notación es entonces 10.0.0.0 255.224.0.0**

# 4.4 *Subnetting* y VLSM

## SUBNETTING

El método de *subnetting* es el proceso de dividir una red en subredes más pequeñas en su esquema de direccionamiento. Esto quiere decir que el direccionamiento de red se pude "subir" un nivel hablando en su división lógica para un mejor aprovechamiento en los espacios de direccionamiento.

Esta estrategia de aprovechamiento se puede aplicar siempre y cuando esta sea soportada por los protocolos de ruteo que se usan. Algunos de los protocolos de ruteo que soportan son RIPv2, OSPF y EIGRP.

## CALCULOS PARA SUBNETEAR

Recordemos que una dirección IP tiene 32 bits y está formada de dos partes, la parte de red y la parte de host. La longitud de la porción de red y de host depende de la clase A, B o C. Por lo tanto el numero de host por red depende de la clase.

Por default

| | 8 Bits | 8 Bits | 8 Bits | 8 Bits |
|---|---|---|---|---|
| **Class A:** | **Network** | Host | Host | Host |
| **Class B:** | **Network** | **Network** | Host | Host |
| **Class C:** | **Network** | **Network** | **Network** | Host |
| **Class D:** | Multicast | | | |
| **Class E:** | Research | | | |

figura 16.- IPs por clase

Las direcciones de subred se crean tomando bits prestados de la porción de host de clase A, B o C. Usualmente un administrador de red define el tamaño de las subredes dependiendo de las necesidades de la red.

Cada que un bit es prestado de la porción de host para la porción de red, el numero de subredes aumenta y el numero de host por subred disminuye.

Los siguientes son ejemplos de cómo las subredes crecen con cada bit prestado

Usando 3 bits prestados resulta en 8 posibles subredes $(2^3 = 8)$

Usando 4 bits prestados resulta en 16 posibles subredes $(2^4 = 16)$

Usando 3 bits prestados resulta en 32 posibles subredes $(2^5 = 32)$

Por ejemplo:

Dada la dirección IP de clase C 192.168.1.0/24, obtener 4 subredes. Indique cuantos host se pueden tener por subred. Indique la máscara de subred expresada en decimal

Mascara por default 11111111.11111111.11111111.00000000

Nueva mascara 11111111.11111111.11111111.11000000 en decimal 255.255.255.192

Para calcular el número de host tenemos que $2^6 = 64$ entonces 64-2 = 62 usables.

| ID de red | Inicio usables | Fin usables | broadcast |
|---|---|---|---|
| 192.168.1.0 | 192.168.1.1 | 192.168.1.62 | 192.168.1.63 |
| 192.168.1.64 | 192.168.1.65 | 192.168.1.126 | 192.168.1.127 |
| 192.168.1.128 | 192.168.1.129 | 192.168.1.190 | 192.168.1.191 |
| 192.168.1.192 | 192.168.1.193 | 192.168.1.254 | 192.168.1.255 |

EJERCICIOS

Nombre del participante_____ RPE_____

Dada la dirección de red 192.168.0.0/24. Obtenga  subredes para grupos de 13 hosts e indique cuantas subredes se tienen.
Indique las redes, broadcast y los rangos de host usables. Indique la nueva mascara en decimal.

Dada la dirección de red 10.0.0.0/8 Obtenga 8 subredes e indique cuantos host usables hay por subred.
Indique las redes, broadcast y los rangos de host usables. Indique la nueva mascara en decimal.

# *VARIABLE-LENGTH SUBNET MASK* (VLSM)

Antes de trabajar con VLS es importante tener conocimientos formes de *subnetting*.

Cuando se es asignada más de una máscara de subred para una red dada (*major network*), se considera una red con VLSM. Este método supera la limitación de un número fijo de subredes dentro de una red, pudiendo variar la máscara de subred a conveniencia.

**"VLSM es poder subnetear una subred. Es decir, un subneteo de un subneteo."**

VLSM proporciona la capacidad de incluir más de una máscara de subred dentro de una red y la capacidad de subnetear una red ya antes subneteada. Abajo se muestra una red con LVSM.

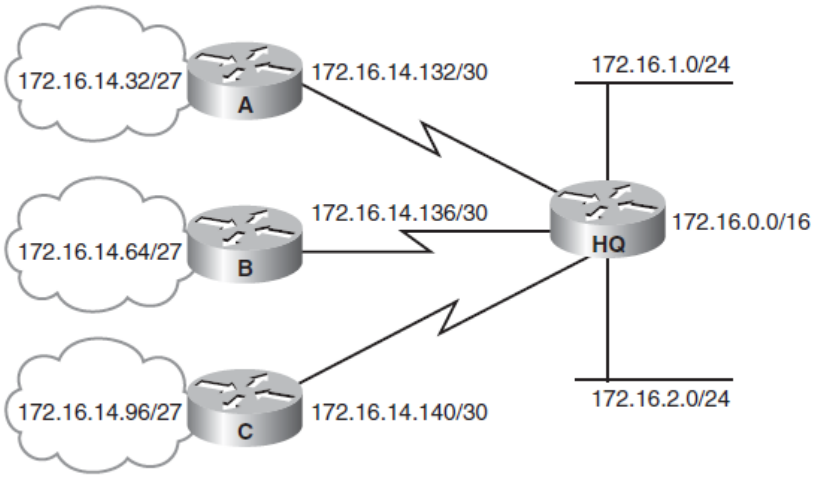



figura 17.- VLSM

VLS proporciona los siguientes beneficios.

Un aprovechamiento aun más eficiente del direccionamiento de red: sin el uso de VLSM los administradores deben implementar un simple subneteo y trabajar con mascaras fijas.

Por ejemplo. Considerando que la dirección 172.16.0.0/16 se divide en subredes usando mascara /24, y una de las subredes en el rango 172.16.14.0/24 requiere un menor número de direcciones de host algo así como una máscara /27. VLS permite que podamos tener múltiples mascaras dentro de una red.

Ejercicio:

Obtenga el esquema de direccionamiento teniendo en cuenta todas las subredes de la figura 17.

# 4.5 Sumarización de rutas con VLSM

En grandes redes, cientos incluso miles de direcciones de red pueden existir. En esos ambientes, no es deseable por los *Routers* mantener muchas rutas en sus tablas de ruteo. La sumarización de rutas, incluso llamadas "***SUPERNETTING***", puede reducir el número de rutas que el Router debe mantener representando una serie de redes en una sola dirección sumarizada.

La figura que sigue muestra el *Router* A que envía tres entradas de actualizaciones de ruteo sumarizadas en una sola dirección de red. En este ejemplo se sumariza todo el tercer octeto.

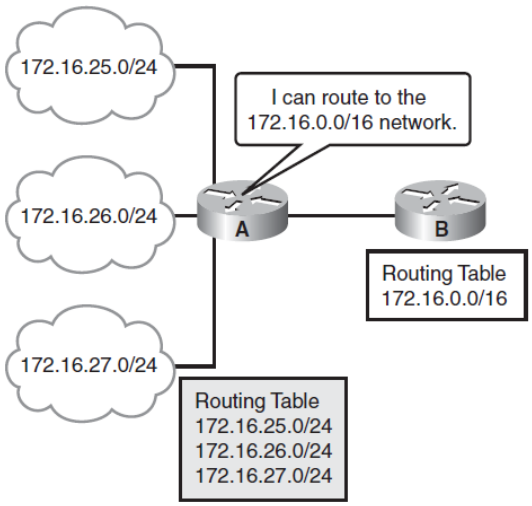



figura 18.- Sumarización de rutas

La sumarización de rutas es más eficiente dentro de un ambiente de subredes cuando el direccionamiento de red está en bloques continuos hablando de 2 a sus potencias. Por ejemplo 4, 16, o 512 pueden ser representadas por una sola entrada debido a su máscara de subred sumarizada.

Los protocolos de ruteo sumarizan o agregan rutas basándose en las redes compartidas dentro de la red. Los protocolos de ruteo sin clase como RIPv2, OSPF y EIGRP, soportan rutas sumarizadas basándose en las direcciones de subred o incluso de VLSM.

Supongamos que un Router recibe actualizaciones de las siguientes rutas.

172.16.168.0/24
172.16.169.0/24
172.16.170.0/24
172.16.171.0/24
172.16.172.0/24
172.16.173.0/24
172.16.174.0/24
172.16.175.0/24

Para determinar la ruta sumarizada, el Router determina el numero de bits del alto orden que hacen *match* en todas las direcciones. Es necesario convertir las direcciones a un formato binario para determinar el número de bits en común de las direcciones.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 172.16.168.0/24 = | 10101100 | | 00010000 | | 10101 | 000 | | 00000000 |
| 172.16.169.0/24 = | 172 | . | 16 | . | 10101 | 001 | . | 0 |
| 172.16.170.0/24 = | 172 | . | 16 | . | 10101 | 010 | . | 0 |
| 172.16.171.0/24 = | 172 | . | 16 | . | 10101 | 011 | . | 0 |
| 172.16.172.0/24 = | 172 | . | 16 | . | 10101 | 100 | . | 0 |
| 172.16.173.0/24 = | 172 | . | 16 | . | 10101 | 101 | . | 0 |
| 172.16.174.0/24 = | 172 | . | 16 | . | 10101 | 110 | . | 0 |
| 172.16.175.0/24 = | 172 | . | 16 | . | 10101 | 111 | . | 0 |
| | Number of Common Bits = 21<br>Summary: 172.16.168.0/21 | | | | | Noncommon<br>Bits = 11 | | |

Figura 19. Sumarización dentro de un octeto.

En la figura de arriba los primeros 21 bits hacen *match*. Por eso la mejor ruta sumarizada es 172.16.168.0/21

Para permitir al *Router* agregar el mayor numero de direcciones IP dentro de una ruta sumarizada, el plan de direccionamiento que el administrador elige debe ser jerárquico. Esto es importante cuando se usa VLSM.

Un diseño de VLSM permite el máximo aprovechamiento de las direcciones IP, así como actualizaciones de ruteo más eficientes cuando se una un direccionamiento jerárquico.

Por ejemplo en la figura que sigue, la sumarización ocurre en dos niveles.

El *Router* C sumariza y actualiza dos redes 172.16.32.64/26 y 172.16.32.128/26 en una sola actualización 172.16.32.0/24.

El *Router* A recibe tres diferentes actualizaciones de ruteo pero las sumariza en una sola actualización antes de propagarla en la red corporativa.

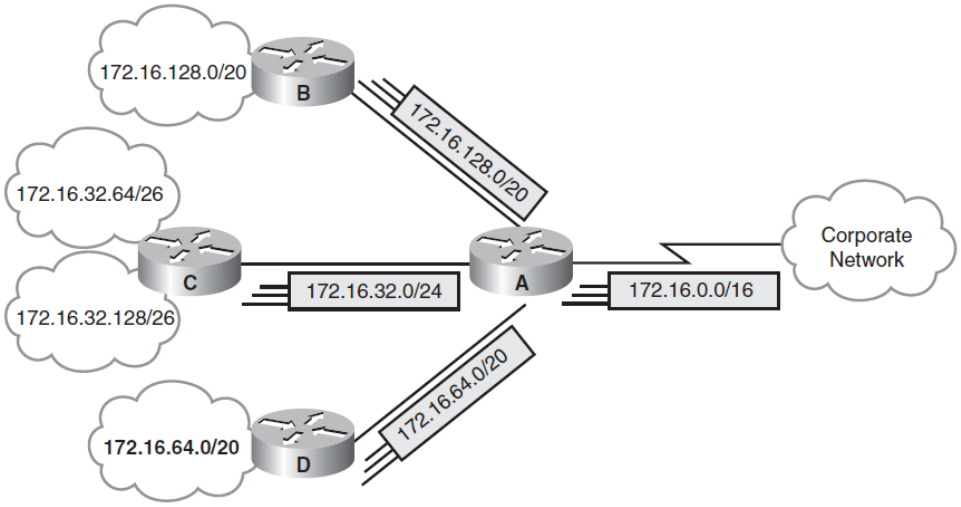



Figura 20. Sumarización en un ambiente VLSM

**"La sumarización de redes reduce el uso de memoria de los *Routers* y el tráfico en la red de los propios protocolos de ruteo."**

**NOTA:** Los protocolos de ruteo *classful* sumarizan automáticamente en las fronteras de cada red. Este comportamiento, podría no ser cambiado con RIPv1 e IGRP. Esto trae como consecuencia:

- Las subredes no son anunciadas a otra red principal diferente.
- Las subredes discontiguas no son visibles entre ellas.

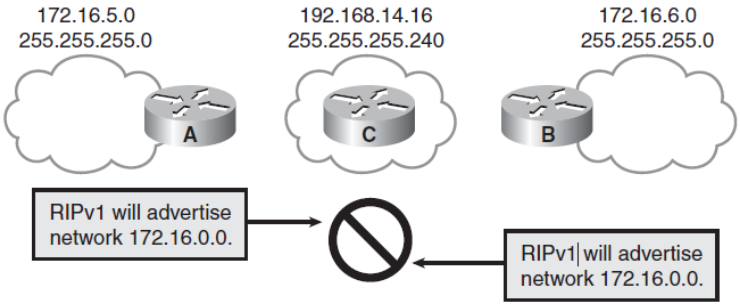



Figura 21. *Sumarización classful*

# 5.- INICIALIZANDO UN ROUTER

Un Router cisco se inicializa cuando es prendido por primera vez y no hay ninguna configuración salvada. Cuando termina de inicializar se puede introducir los primeros comandos de configuración. Los Routers deben de inicializar de una manera correcta y tener una configuración válida para operar en una red.

**"Cuando un Router inicia, éste busca un archivo de configuración. Si no lo encuentra, el Router ejecuta un cuestionario inicial de configuración. Este es un programa para realizar una configuración muy básica."**

Después de que un Router termina un POST y carga la imagen del IOS, busca el archivo de configuración en su NVRAM. La NVRAM del Router es un tipo de memoria que retiene su contenido aun cuando el dispositivo es apagado y vuelto a prender. Si el Router ya tienen un archivo de configuración en la NVRAM, el *prompt* de modo usuario aparece después de introducir una contraseña, si es que esta está configurada.

## 5.1 Configuración de un Router desde CLI

Cuando se configura un *Router* cisco desde la CLI o una terminal remota, el IOS de cisco proporciona comandos que funcionan de una manera jerárquica según el nivel de configuración.

Para propósitos de seguridad, existen dos primeros niveles de acceso. Estos son:

**Modo usuario:** comando para checar el estatus del *Router.*

**Modo privilegiado:** incluye algunos comandos para hacer cambios en la configuración del *Router.*

Es posible poner una barrera de seguridad en medio de estos dos niveles. Se una el comando ***enable password*** o ***enable secret.***

Se usa el comando ***exit*** para pasar del modo privilegiado a modo usuario.

Se puede usar el signo de interrogación (**?**) en el modo usuario o modo privilegiado para ver los comandos disponibles dentro de cada nivel.

Los siguientes dos comandos dentro del de CLI de cisco, se usan para hacer configuraciones un poco más complejas y especificas.

Para el modo **configuración global** se usa el comando *configure terminal* . Desde el modo configuración global , se pude accesar a todos los comandos de configuración general del equipo que incluyen.

- ***Interfaces***: configuraciones por interface
- ***Sub interfaces***: configuraciones por sub interface
- ***Line*** : línea de terminal
- ***Router***: configuración de protocolos de ruteo

Si tecleamos el comando ***exit***, salimos del nivel de configuración global a un nivel inferior.

Algunos comandos ejemplo que se hacen dentro del modo de configuración global son el de ***Hostname*** para nombrar al equipo, y el de ***enable password*** para usar seguridad en el equipo.

Después de hacer alguna configuración en el *Router*, se deban guardar los cambios desde la memoria RAM a la memoria NVRAM usando el comando ***copy running-config startup-config***

## 5.2 Configuración de direccionamiento IP en un *Router*

La función principal de un *Router* es el reenvío de paquetes desde una red a otra. Para hacer eso, se debe definir las características de las interfaces a través de las cuales los paquetes van a ser recibidos y enviados.

Las interfaces de un *Router* incluyen: la dirección IP de la interface, el método de encapsulación de la capa de enlace de datos, el tipo de medio y ancho de banda. Para poder entrar a la configuración de un *Router*, usamos el comando ***interface*** seguido del tipo de interface y el numero. El numero está indicado en el *hardware* del equipo.

**Cada interface de un Router debe tener su propia dirección IP usable que la identifique como única en la red.**

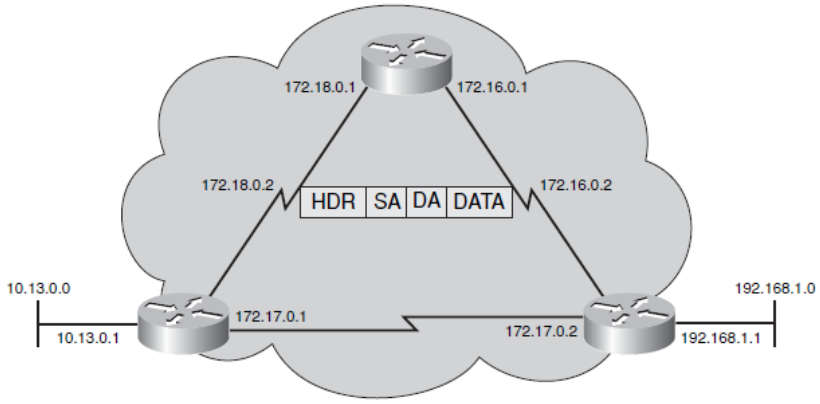



Figura 22. Direccionamiento por interface

Para poder configurar un Router cisco, se deben seguir los siguientes pasos.

Paso 1.- Entramos a modo configuración global.

Router# **configure terminal**

Paso 2.-identificamos la interface especifica que requiere una dirección IP.

**Router# interface fa0/0**

**P**aso 3.- se configura la dirección IP y mascara de subred a la interface usando una dirección usable.

**Router(config-if)# ip address 192.168.1.1 255.255.255.0**

Paso4.- habilitamos la interface para que cambie de administrativamente apagada a prendida (levantada)

**Router(config-if)# no shutdown**

## REVISIÓN DE CONFIGURACION DE INTERFACE

Cuando se completa la configuración de una interface, se debe de verificar usando el comando ***show interfaces*** para verificar el estado de la interface.

El comando ***show interfaces*** despliega el estatus y estadísticas de todas las interfaces del *Router*. Podemos. Podemos ver el estatus de una interface en particular indicando el número de la interface con el comando ***show interface fa0/0.***

```
RouterA# show interfaces

Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 00e0.1e5d.ae2f (bia 00e0.1e5d.ae2f)
  Internet address is 10.1.1.11/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:07, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     81833 packets input, 27556491 bytes, 0 no buffer
     Received 42308 broadcasts, 0 runts, 0 giants, 0 throttles
     1 input errors, 0 CRC, 0 frame, 0 overrun, 1 ignored, 0 abort
     0 input packets with dribble condition detected
     55794 packets output, 3929696 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 4 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Figura 23.- Comando ***show interfaces***

Uno de los parámetros más importantes es el estado de la interface para verificar la conectividad. Esto se ve en la siguiente figura.

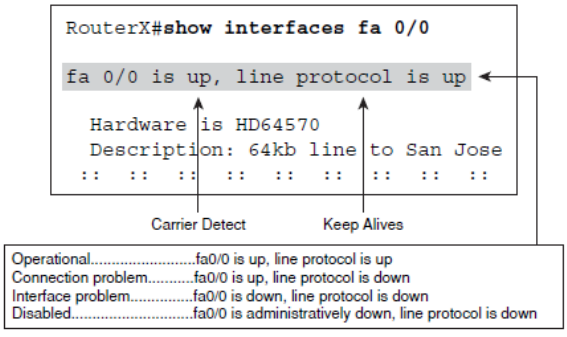



Figura 24.- estatus de interface

# 6.- HABILITANDO RUTEO

"Ruteo es el proceso que determina a donde enviar paquetes que tienen como destino una red fuera de la red local. Los *Routers* reúnen y mantienen información de ruteo para habilitar la transmisión y recepción de paquetes de información."

Conceptualmente, la información de ruteo toma la forma de entrada en una tabla de ruteo, con una entrada para cada ruta identificada. Se puede configurar de manera estática (manual) las entradas en la tabla, o el *Router* pude usar protocolos para crear y mantener la tabla de ruteo de manera dinámica para ajustar cambios de red cuando ocurran.

Para administrar eficientemente el direccionamiento de red, se debe entender la operación del ruteo estático y dinámico y el impacto que tienen en las redes.

Para poder enrutar paquetes, un *Router* debe de hacer lo siguiente:

- Identificar la red destino.
- Identificar fuentes de información de ruteo
- Identificar las posibles rutas para llegar al destino
- Definir la mejor ruta

La información de ruteo que un *Router* obtiene de otros *Routers* la pone en su tabla de ruteo. El *Router* confía en esta tabla ya que le indica que interfaces usar para el reenvío de paquetes.

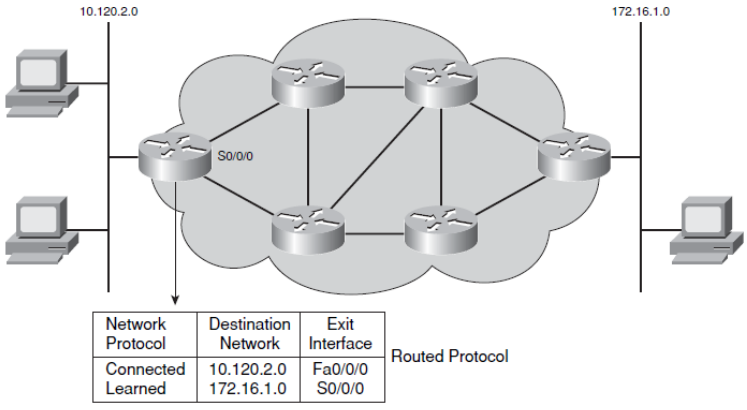


Figura 25.- Ruta hacia un destino

## 6.1 Configuración de Ruteo estático

El *Router* aprende rutas cuando un administrador configura rutas de manera manual. El administrador debe de actualizar de manera manual las rutas estáticas usando hay un cambio en la topología.

"Las rutas estáticas son comúnmente usadas cuando se requiere *ruteo* de una red a una red *stub*. Una red *stub* es una red alcanzada por una sola ruta. "

La sintaxis para la configuración de una ruta estática es la siguiente:

routerX(config)#**ip route** [network] [mask] + [address next hop router]

En la figura de abajo, el *Router* A esta configurado con una ruta estática para alcanzar la red 172.16.1.0 a través de una interface serial. Así mismo el *Router* B está configurado con una ruta estática para alcanzar las redes del *Router* A.

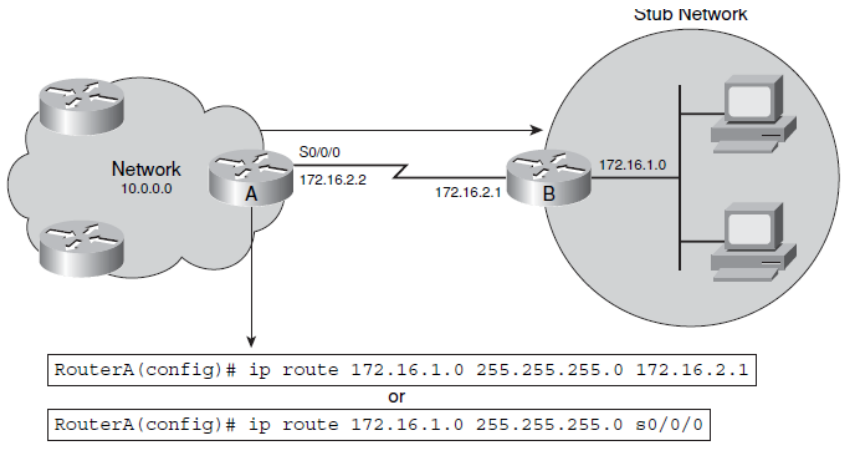



Figura 26.- Ejemplo de ruta estática

Para configurar una ruta estática, no olvidemos que el comando ***IP route*** es un comando de configuración global.

## VERIFICACIÓN DE CONFIGURACIOÓN DE RUTEO ESTÁTICO

Para verificar que la configuración de una ruta estática está bien configurada , usamos el comando ***show ip route*** , el cual nos muestra la tabla de ruteo del *Router* que se ha llenado de manera manual por el administrador. Las rutas estáticas estarán marcadas por **"S"**.

Un ejemplo del comando ***show ip*** route es:

```
RouterA# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     10.0.0.0/8 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Serial0/0/0
S*   0.0.0.0/0 is directly connected, Serial0
```

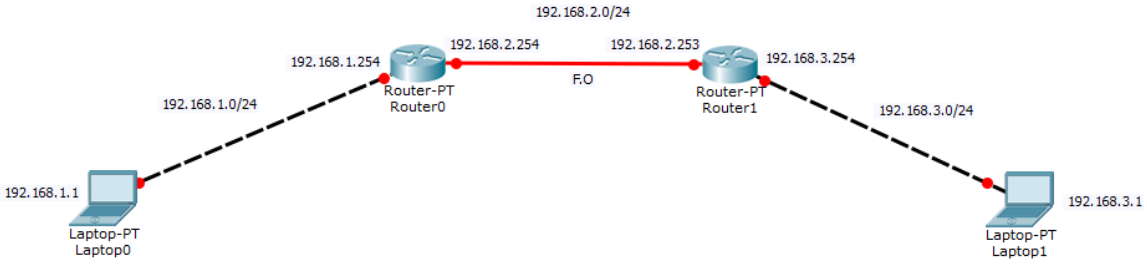
Figura 27.- Comando *Show IP route*

PRACTICA RUTEO ESTATICO

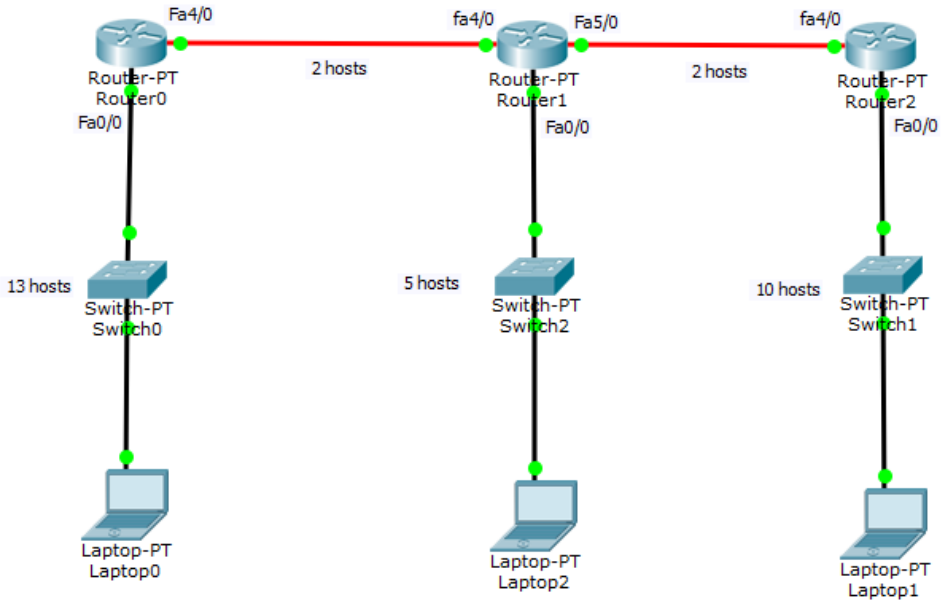
Nombre del participante_____ RPE_____

Tiempo 2hr max.

1.- Dada la topología, lograr comunicación usando ruteo estático. Se debe respetar el direccionamiento dado. **Laptop 0** debe tener comunicación con **laptop 1** a pesar de estar en redes diferentes. Verificar las tablas de ruteo.




2.- Dada la topología usar la red 172.16.0.0/16 y hacer *subnetting* para hacer un esquema de direccionamiento óptimo de acuerdo a las necesidades de la topología y lograr comunicación entre las tres laptop.




## 6.2 HSRP con ruteo estático

"HSRP (*Hot Standby Router protocol*) es un protocolo de capa 3 que proporciona redundancia en ruteo para el tráfico IP sin depender de la disponibilidad de un solo Router. HSRP no es un protocolo de *routing*."

Es un método usado para lograr alta disponibilidad de red al proporcionar redundancia en el "primer salto" de los host de una LAN configurada con *default Gateway*.

HSRP habilita un juego de interfaces de Router que trabajan juntas dando la apariencia de un solo *Router* virtual o *default Gateway* para los *host* en la LAN. Cuando HSRP es configurado en una red o segmento de red, proporciona un dirección MAC virtual y una dirección IP virtual que es compartida por el grupo de *Routers* configurados.

El *Router* virtual no existe. Solo es un ente que toma el control de los *Routers* que están configurados para proporcionar respaldo uno del otro, el cual asume el control si el *Router* designado como activo falla.

HSRP proporciona alta disponibilidad de red proporcionando redundancia para el trafico IP de los *host* en una red. En un grupo de interfaces de *Routers* , el *Router* activo es el *Router* que enruta los paquetes. El *Router standby* es el que toma las tareas de ruteo cuando el activo falla.

HSRP detecta cuando el Router activo falla, el Router en *standby* asume el control del grupo HSRP. Si hay más de dos *Routers*, un nuevo *Router standby* es elegido al mismo tiempo. Los dispositivos corriendo HSRP envían y reciben *multicast hello packets* para detectar que el *Router* ha fallado y designar un *Router* activo y un *Routers standby*. Cuando HSRP es configurado en una interface, la redirección de mensajes de *Internet Control message* Protocol (ICMP) es habilitado de manera automática para esta interface.

HSRP consta de 5 estados de operación:

1. **Initial:** Aquí aún NO está trabajando HSRP, éste estado es cuando la interface acaba de levantar.
2. **Listen:** Escucha los *Hellos Messages*, conoce la IP virtual.
3. **Speak:** Envía *Hellos Messages* y participa en la elección de los *Routers* (*Active/Passive*).
4. **Standby:** Es candidato para el próximo *active Router*
5. **Active**: Es el encargado de enviar paquetes hacia la IP virtual.

## HABILITANDO HSRP

El comando de configuración de interface ***standby ip*** activa HSRP en la interface configurada. Si una dirección IP es especificada, la dirección es usada como dirección designada para el grupo *hot standby*.

Para poder iniciar la configuración es necesario estar en modo privilegiado.

**Paso 1.-** usamos el comando ***configure terminal*** para entrar a modo configuración global.

**Paso 2.-** Pasamos al modo interface con el comando ***interface -id*** en la interface donde se quiere habilitar HSRP.

**Paso 3.-** Crear un grupo HSRP usando un numero y una dirección IP virtual. Se usa el comando ***standby*** [*group-number*] ***ip*** [*ip-address*]

Ejemplo:

**Router#** configure terminal
**Router(config)#** interface gigabitethernet 0/1
**Router(config-if)#** standby 1 ip [virtual IP]

## HSRP PREEMPT

La prioridad *standby* sirve para designar al *Router* activo del *Router standby*. El *Router* que posea la prioridad más alta, es que *Router* que el protocolo designa como activo. Por default la prioridad es 100. El número más alto representa una prioridad más alta, es decir, más probabilidad de ser el *Router* activo. Este número debe ser entre 1 y 255.

El comando ***standby priority*** es usado para encontrar el *Router* activo y el *Router standby* cuando un nuevo Router toma el control.

Para configurar la prioridad HSRP:

***Standby*** [*group-number*] ***priority*** [*priority number*]

Retomando el ejemplo anterior:

**Router#** configure terminal
**Router(config)#** interface gigabitethernet 0/1
**Router(config-if)#** standby 1 ip [virtual IP]
**Router(config-if)#** Standby 1 priority 150

Con el comando ***show standy*** podemos ver las configuraciones de HSRP

## HSRP *STANDBY TRACK*

La configuración de *standby track* en una interface liga la prioridad del *Router* HSRP con la disponibilidad de sus interfaces.

Cuando una interface rastreada falla, la prioridad HSRP en el *Router* en el cual ha sido configurado el rastreo se decrementa a 10. Si una interface no es rastreada, su cambio de estado no afecta a la prioridad del *Router* configurado.

La prioridad de la interface con el comando *standby track* especifica cuanto decrementa la prioridad cuando una interface rastreada se "cae". Cuando la interface "levanta", la prioridad se incrementa al mismo valor original.

Para configurar *standby track*:

***Standby*** [*group-number*] ***track*** [*type number*] [*interface-priority*]

Ya con el ejemplo anterior queda así:

**Router#** configure terminal
**Router(config)#** interface gigabitethernet 0/1
**Router(config-if**)# standby 1 ip [virtual IP]
**Router(config-if**)# standby 1 priority 150
**Router(config-if**)# standby 1 track fastethernet 0/1

NOTA: Es necesario agregar el comando standby [*gorup-number*] preempt. Este comando va acompañado del comando *track*, y hace que el Router que tiene la prioridad más alta, tome el control como Router activo.

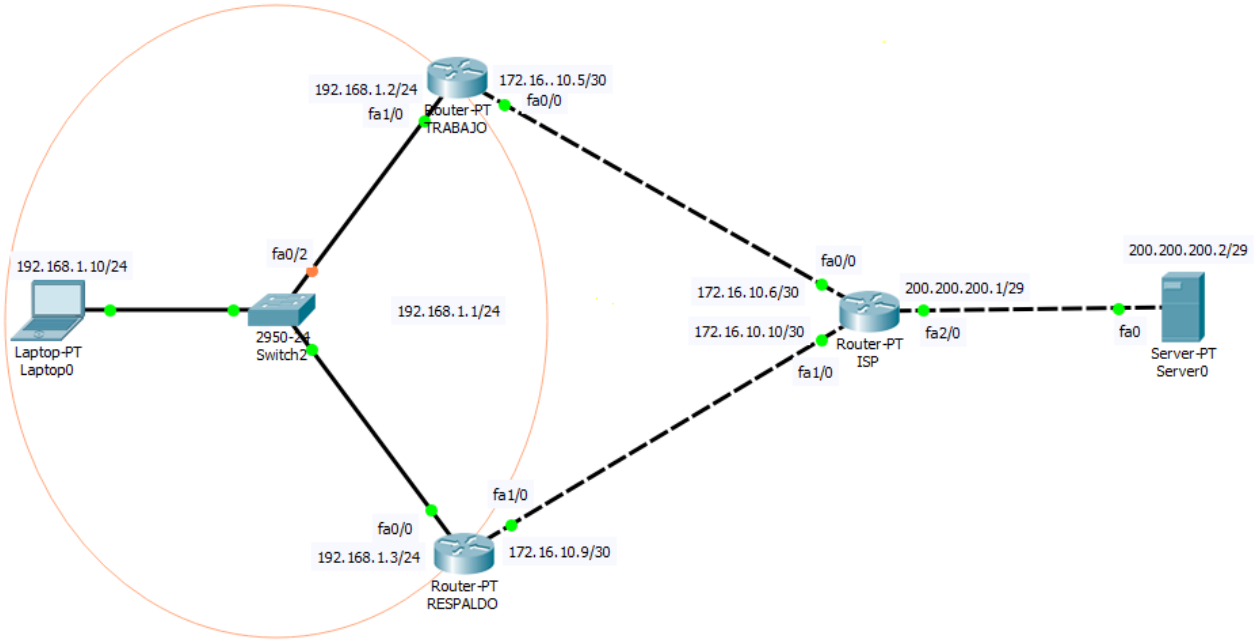
PRACTICA HSRP

Nombre del participante_____ RPE_____

Tiempo 1hr max.

1.- Dada la siguiente topología, construir en *packet tracert*. Lograr comunicación desde laptop 0 hasta el servidor 0 con alta disponibilidad. El *Router* activo debe ser TRABAJO. Las interfaces dentro del circulo naranja deben configurarse como *tracked*.

Usando el comando ***show standby brief,*** verificar funcionalidad.




Anote sus conclusiones aquí.

_____

_____

_____

_____

## 6.3 Ruteo dinámico con EIGRP

"Un protocolo de ruteo define las reglas que son usadas por un *Router* cuando éstos se comunica con sus vecinos, es decir, con otros *Routers*. El ruteo dinámico confía plenamente en un protocolo de ruteo, que sea quien disemine la información de ruteo por toda la red y que llegue de manera confiable a todos los *Routers* involucrados."

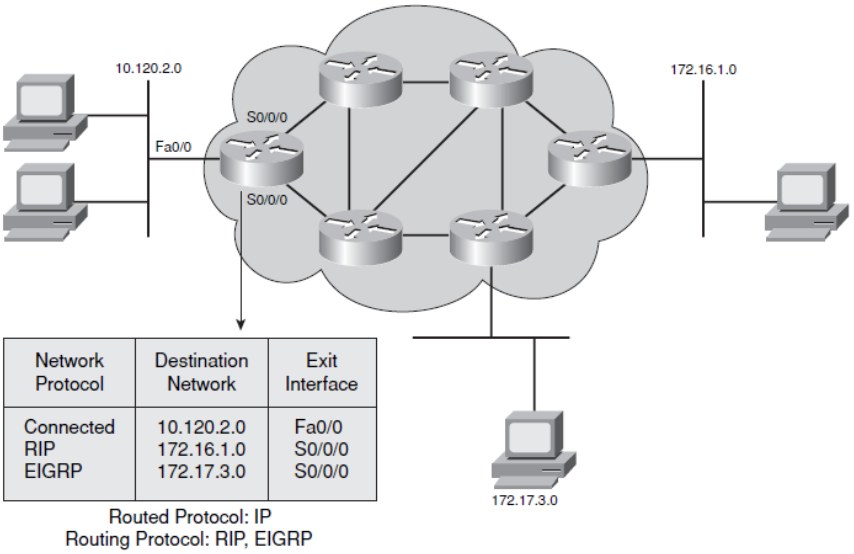



Figura 28.- Ruteo dinámico

Un ejemplo de la información que describe a un protocolo de ruteo es la siguiente:

- Como son transmitidas las actualizaciones
- Que información es transmitida
- Cuando es transmitida esa información
- Como localizar a los destinatarios de la información.

## TIPOS DE PROTOCOLOS DE RUTEO

Se tienen dos tipos de protocolos de ruteo. Los protocolos IGP (*interior gateway protocols*) y los BGP (*exterior gateway protocols*).

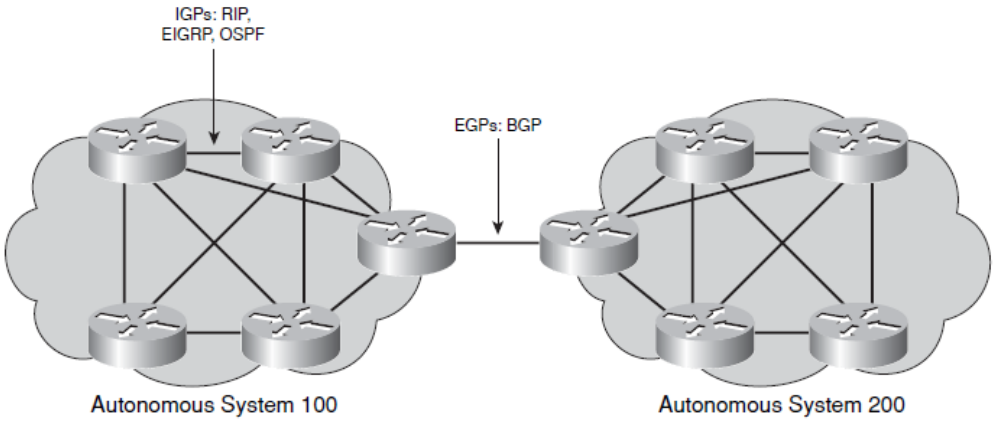



Figura 29.- Tipos de protocolos de ruteo

**Los IGP** son protocolos que son usados para intercambiar información de ruteo dentro de un sistema autónomo. *Routing Information Protocol* (RIP)versión 1, versión 2, EIGRP y OSPF son ejemplos de IGPs.

Además de el tipo de protocolo, como EIGRP, OSPF, etc., los protocolos de ruteo también son clasificados por su modo de funcionamiento. Los métodos son:

**Vector distancia:** el protocolo determina la dirección (vector) y distancia (saltos) para determinar la mejor ruta.

**Híbridos**: combinan aspectos de vector distancia y estado de enlace.

**Estado de enlace:** también conocido como algoritmo SPF (primero la ruta más corta) crea un mapa de la topología completa de la red, o al menos de la porción donde el Router está situado.

**"Múltiples protocolos de ruteo y rutas estáticas pueden ser usadas al mismo tiempo. Si varias fuentes de información de ruteo existen, un valor de distancia administrativa mide el grado de confiabilidad de cada fuente de información. "**

## DISTANCIA ADMINISTRATIVA

La distancia administrativa es el primer criterio que un Router utiliza para determinar qué protocolo de ruteo utilizar si dos protocolos proporcionan información de ruta para el mismo destino. La distancia

administrativa mide la fiabilidad de la fuente de la información de ruteo. La distancia administrativa tiene importancia local solamente y no se publica en actualizaciones de ruteo.
Nota: Cuanto más bajo sea el valor de la distancia administrativa, más confiable será el protocolo. Por ejemplo, si un *Router* recibe una ruta a cierta red de *Open Shortest Path First* (OSPF) (distancia administrativa predeterminada: 110) y de *Interior Gateway Routing Protocol* (IGRP) (distancia administrativa predeterminada: 100), el *Router* optará por IGRP porque es más confiable. Esto significa que el *Router* agrega la versión de la ruta de IGRP a la tabla de ruteo.
Si se pierde la fuente de la información derivada de IGRP (debido a un corte en el suministro eléctrico, por ejemplo), el software utiliza la información derivada de OSPF hasta que reaparezca la información derivada de IGRP.

**Es un valor de 0 a 255. Un protocolo de ruteo con una distancia administrativa baja es más confiable que uno con alta distancia administrativa.**

Por ejemplo, en la figura de abajo, si el Router A recibe información de una ruta hacia la red E de EIGRP y RIP al mismo tiempo, el *Router* A usa la distancia administrativa para determinar que EIGRP es más confiable. El *Router* A usa entonces agrega a su tabla de ruteo la ruta de EIGRP

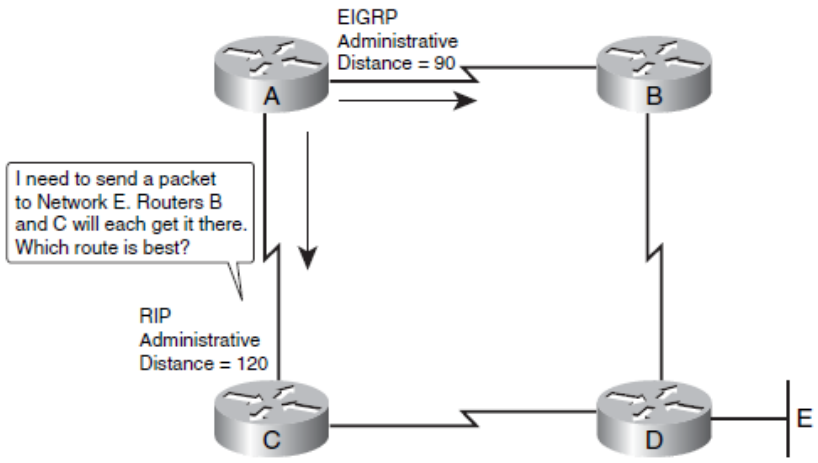



Figura 30.- Distancia administrativa

| | |
|---|---|
| Interfaz conectada | 0 |
| Ruta estática | 1 |
| Ruta de resumen del Enhanced Interior Gateway Routing Protocol (EIGRP). | 5 |
| External Border Gateway Protocol (BGP) | 20 |

| | |
|---|---|
| EIGRP Interno | 90 |
| IGRP | 100 |
| OSPF | 110 |

Figura 31.- Distancia administrativa

## *Enhanced Interior Gateway Routing Protocol* (EIGRP)

EIGRP es un protocolo de ruteo vector distancia avanzado desarrollado por cisco. EIGPR es utilizado para muchas topologías y medios. Es una red bien diseñada, EIGRP es escalable y proporciona una convergencia extremadamente rápida con el mínimo *overhead*. Por lo tanto es el protocolo mayormente usado para las redes con dispositivos cisco.

## INTRODUCCION

EIGRP es un protocolo propietario que combina las ventajas de *link-state* y *distance-vector*. EIGRP es un vector distancia avanzado o más bien, un protocolo de ruteo hibrido que incluye las siguientes características:

- **Convergencia rápida:** EIGRP una el *Diffusing Update Algorithm* (DUAL) para lograr convergencia rápida. Un Router que usa EIGRP almacena todas las rutas de respaldo disponibles hacia algún destino para que éste rápidamente rutas alternas. Si n existe una ruta apropiada de respaldo en la tabla de ruteo local, EIGRP consulta a sus vecinos para descubrir una ruta alterna.

- **Reduce el uso de ancho de banda:** EIGRP no hace actualizaciones periódicas. En lugar de eso, envía actualizaciones parciales cuando el camino o la métrica cambia para esa ruta. Cuando la información de la ruta cambia, DUAL envía una actualización solo de esa ruta en vez de la tabla de ruteo completa.

- **Ruteo *classless***: Debido a que EIGRP es un protocolo de ruteo *classless*, éste anuncia la máscara para cada red destino. El ruteo con la máscara permite a EIGRP soportar redes discontiguas y VLSM.

- **Menor *overead***: EIGRP usa *multicast* y *unicast* en lugar de *broadcast*. Como resultado, las estaciones finales no son afectadas por las actualizaciones de ruteo y solicitudes de información de la topología.

- **Balanceo de carga:** EIGRP soporta balanceo de carga en rutas desiguale, lo que hace que los administradores puedan hacer una mejor distribución del trafico.

- **Fácil sumarización**: EIGRP permite a los administradores crear rutas sumarizadas a donde sea dentro de la red en lugar de confiar en el enfoque tradicional de un *distance-vector* de hacer sumarización de rutas *classful* solo en las fronteras de las *major networks*.

Cada *Router* EIGRP mantiene una tabla de vecinos. Esta tabla incluye una lista de los *Routers* EIGPR directamente conectados que tienen una adyacencia éste Router.

Cada Router EIGRP mantiene una tabla topológica. La tabla topológica incluye entradas de rutas para todos los destinos que el *Router* aprende. **EIGRP elige las mejores rutas hacia los destinos de la tabla topológica y las coloca en la TABLA DE RUTEO**.

En la figura de abajo se muestra.

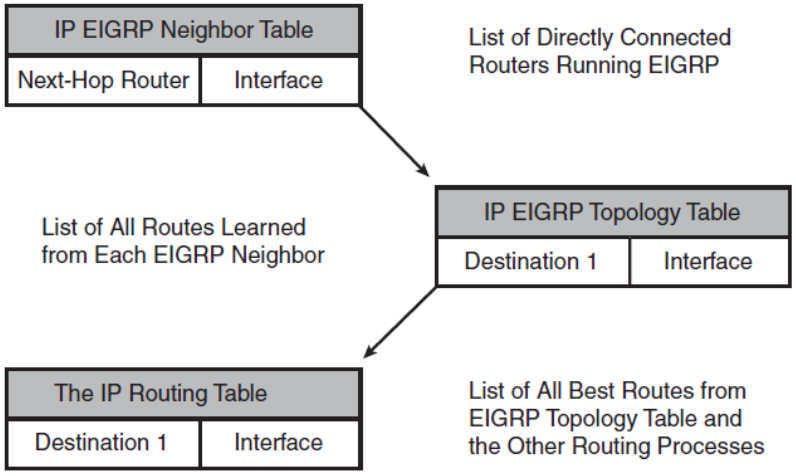



Figura 32.- Tablas EIGRP

"En EIGRP, la mejor ruta es llamada *sucessor route* mientras que la ruta de respaldo es llamada *feasible successor*. "

Para determinar la mejor ruta (*successor route*) y la ruta de respaldo (*feasible successor*) hacia un destino. EIGRP usa los siguientes parâmetros.

- ***Advertised distance***: Es la métrica para un vecino para alcanzar una red en particular.
- ***Feasible distance:*** Es la distancia anunciada por un vecino para una red en particular mas la métrica para alcanzar a ese vecino

Un *Router* compara todas las *feasible distance* para alcanzar una red en especial y entonces selecciona la *feasible distance* mas baja y la pone en su tabla de ruteo. La *feasible distance* para la ruta elegida se vuelve la métrica de ruteo para alcanzar esa red en la tabla de ruteo.

La base de datos topología de EIGRP contiene todas las rutas que son conocidas por cada vecino. El *Router* A y el B envían sus tablas de ruteo al *Router* C, la cual se muestra en la figura de abajo. Ambos *Routers* A y B tienen caminos a la red 10.1.1.0/24, así como a otras redes que no son mostradas.

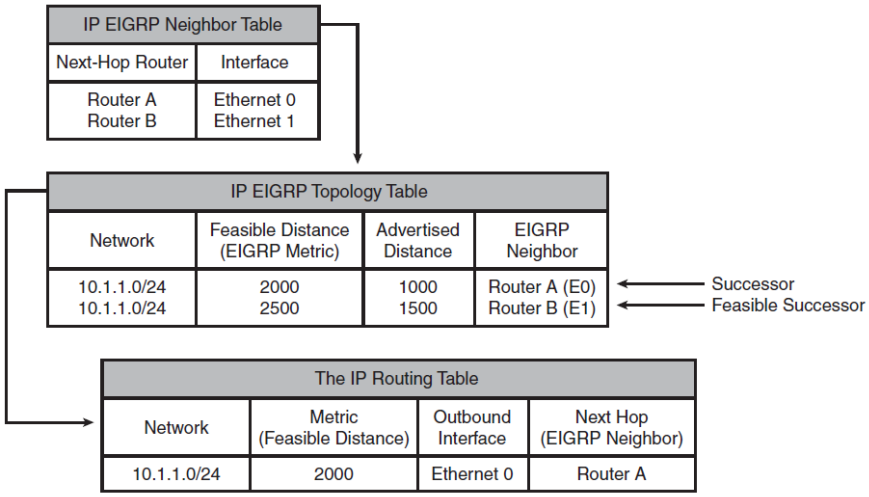



Figura 33.- Tablas EIGRP de *Router* C

El *Router* C tiene 2 entradas para alcanzar la red 10.1.1.0/24 en esta tabla topológica. La métrica EIGRP para el *Router* C para alcanzar el *Router* A y B es 1000. Adicional a este costo, la respectiva *advertised distance* para cada *Router*, y el resultado representa las *feasible distance* que el *Router* C debe viajar para alcanzar la red 10.1.1.0/24.

El *Router* C elige la *feasible distance* de menor valor (2000) y la instala en su tabla de ruteo como la mejor ruta para alcanzar a 10.1.1.0/24. La ruta con menor costo de *feasible distance* que es instalada en la tabla de ruteo es llamada *SUCCESSOR ROUTE*.

El *Router* C entonces elige una ruta de respaldo llamada ruta *FEASIBLE SUCCESSOR*, si una existe. Para una ruta que se convierte en *feasible successor*, el *Router* del próximo salto debe tener una *advertised distance* que es menor que la *feasible distance* de la *successor route* actual.

Si la *successor route* se vuelve invalida, posiblemente por un cambio en la topología o si un vecino cambia la métrica, DUAL revisa por rutas de respaldo (*feasible successor*) hacia la ruta destino. Si alguna es encontrada, DUAL la usa PREVINIENDO LA NECESIDAD DE RECALCULAR LA RUTA.

## 6.4 Configuración de EIGRP

Para crear un proceso de ruteo EIGRP solo se usan los comandos ***router eigrp*** y ***network***.

EIGRP requiere un numero de sistema autónomo (AS). El numero de sistema autónomo no tiene que ser registrado como es el caso cuando se usa protocolos de ruteo BGP, sin embargo, todos los Routers dentro del mismo sistema autónomo deben usar el mismo número de sistema autónomo para poder intercambiar información de ruteo unos con otros. La figura de abajo muestra una configuración EIGRP de una red simple.

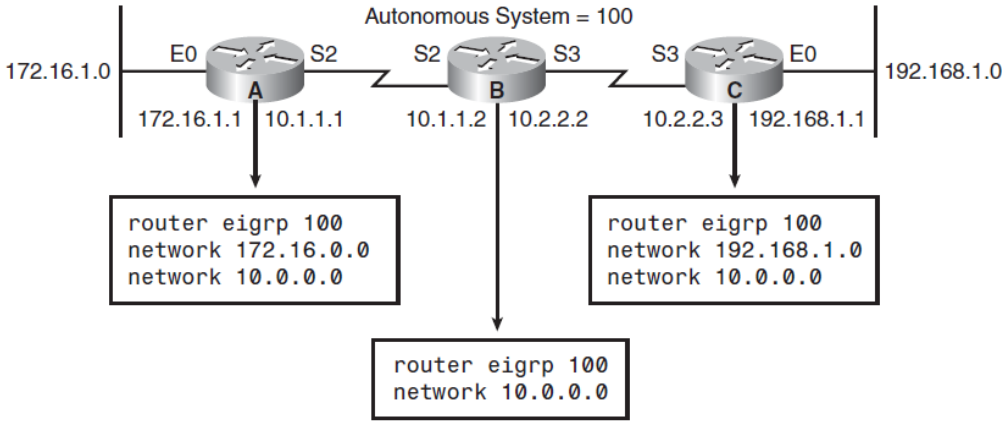



Figura 34.- configuración EIGRP

El comando ***network*** define una *major network* la cual el Router tiene directamente conectada.

El proceso de ruteo EIGRP busca interfaces que tienen una dirección IP que pertenecen a la red que está especificada con el comando *network* y empieza el proceso EIGRP en esas interfaces.

Ejemplo de comandos para EIGRP

| Command | Description |
|---|---|
| **router eigrp 100** | Enables the EIGRP routing process for AS 100 |
| **network 172.16.0.0** | Associates network 172.16.0.0 with the EIGRP routing process |
| **network 10.0.0.0** | Associates network 10.0.0.0 with the EIGRP routing process |

Figura 35.- Comandos EIGRP

## DESPUES DE CONFOGURAR EIGRP, SE PUEDEN USAR COMANDOS DE VERIFICACIÓN EIGRP.

***Show Ip route –*** muestra la tabla de ruteo

***Show ip protocols –*** Estado actual del proceso EIGRP

***Show ip eigrp neighbors –*** Despliega los vecinos que se han conocido por EIGRP

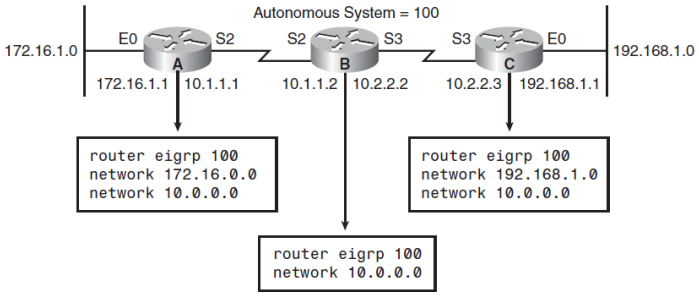
***Show ip eigrp topology–*** Muestra la tabla topológica

PRACTICA EIGRP

Nombre del participante_____ RPE_____

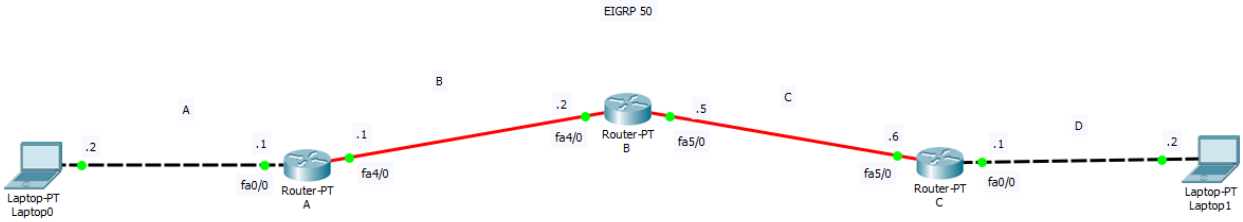
Tiempo 1 hr

1. Dada la topología. Realizar la configuración en *packet tracert* y lograr la adyacencia. Colocar *lap top* en la red 172.16.1.0 y otra en la red 192.168.1.0 para realizar pruebas de ping.




2. Dada la siguiente topología. Implementar *subnetting* para tener el direccionamiento óptimo para las necesidades de la topología. Usar la red 172.16.0.0/16 y generar subredes de 256 host. Usar la subred 5 para el segmento A y la subred 6 para el segmento D. Usar la red 192.168.14.0 y subnetear para tener subredes de 2 hosts. Usar la primer subred para el segmento B y la segunda subred para el segmento C.

.




## REDES DISCONTIGUAS EN EIGRP

"EIGRP automáticamente sumariza rutas en las fronteras de las *major networks*. En algunos casos, el administrador podría no necesitar la auto sumarización. Por ejemplo, si se tienen redes discontiguas se necesita deshabilitar la sumarización automática para eliminar la confusión del Router. Abajo se muestra en la figura un ejemplo de cómo la sumarización puede causar anuncios para la red 172.16.0.0 que es enviada por el *Router* A y el *Router* B al *Router* C

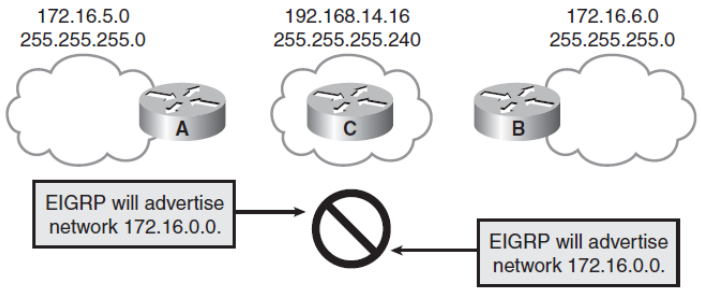



Figura 36.- Redes discontiguas

Para deshabilitar la sumarización automática, se usa el comando ***no auto-summary*** en la configuración de EIGRP del *Router*. Cuando este comando es usado, ambos, el *Router* A y B anunciaran la ruta especifica de la subred para una interface dada, como se muestra en la figura de abajo.

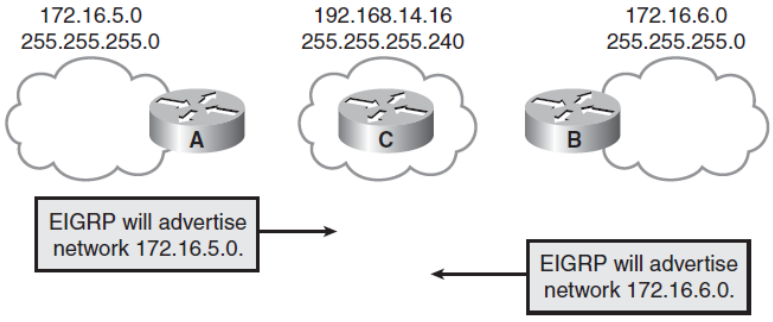



Figura 37.- Sin auto sumarización.

## 6.5. NAT y PAT

Los retos de hacer cada vez más escalable a internet con IPv4 son más y más grandes debido a que las direcciones IP se van terminando debido al creciente uso de dispositivos conectados a internet. Cisco se encargó de desarrollar un mecanismo llamado NAT (*Network Address Translation*) y PAT (*Port Address Translation*) para conservar el uso de direcciones registradas de IPv4 en grandes redes. Los mecanismos de traducción de direcciones NAT y PAT funcionan convirtiendo direcciones dentro de redes privadas a direcciones IPv4 legales registradas para poder tener alcanzabilidad en redes públicas como internet.

**"En resumen NAT y PAT traducen direcciones privadas de redes locales en direcciones públicas para poder ser enrutadas en internet"**

## INTRODUCCION A NAT y PAT

NAT está diseñado para la conservación de direccionamiento IPv4. NAT permite a redes privadas conectarse a internet usando direcciones no registradas. Usualmente NAT conecta dos redes y traduce las direcciones privadas (*inside local*) de una red interna en direcciones públicas (*inside global*) antes de que los paquetes sean enviados a otra red.

Como parte de ésta funcionalidad, se puede configurar NAT para anunciar una sola dirección para toda la red entera hacia el "mundo exterior". Anunciando una sola dirección se puede ocultar la red interna del "mundo exterior" , y esto sirve como una clase de seguridad.

Abajo se muestra un ejemplo de una traducción de una dirección privada en publica.

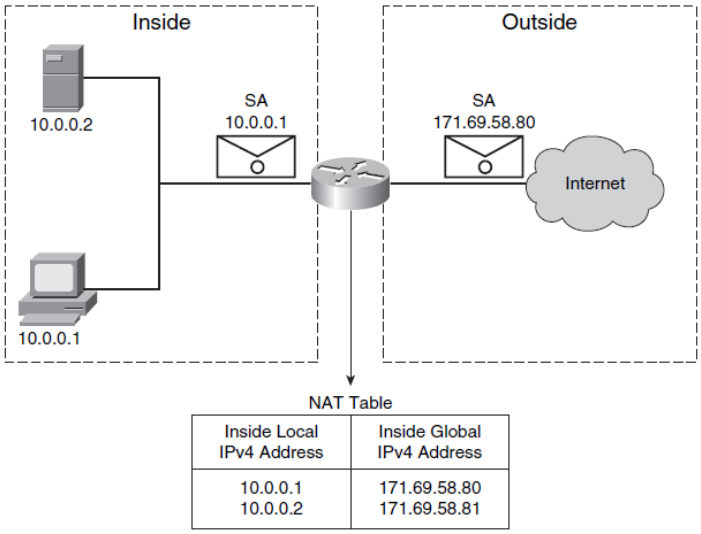



Figura 38.Traduccion de IPs con NAT

Cualquier dispositivo que este en medio de una red interna y una red pública, así como un firewall o Router usa NAT. Esto está definido en RFC 1631.

En terminología de NAT, la *inside network* es la red que está sujeta a traducción. La *outside network* se refiere a todas las demás direcciones. Usualmente son las direcciones validas de internet.

Se definen los siguientes términos para su mejor comprensión.

***Inside local address***: las direcciones IPv4 que son asignadas a un host de la *inside network*.
La *inside local address* es una dirección privada no asignada por algún ISP.

***Inside global address:*** es una dirección asignada por la NIC o un proveedor de servicios que representa una o más IPs privadas hacia el "mundo exterior".

***Outside local address:*** es la dirección IPv4 de un host externo como aparece en su red privada.

***Outside global address:*** es la dirección que se le asigna a un host de una red externa. Es una dirección publica ruteable en internet o red global.

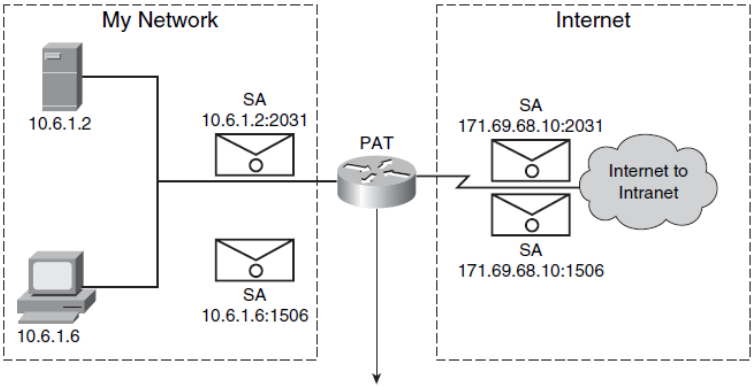
NAT tiene varias formas con las que puede funcionar.

**NAT estático:** es un mapeo de una dirección privada a una dirección pública. (una a una). Este método se usa cuando el dispositivo debe ser accesado desde la red externa.

**NAT Dinámico**: Es el mapeo de una dirección privada a una dirección global tomada de un grupo de direcciones globales.

**NAT Overloading:** es el mapeo de múltiples direcciones privadas a una sola dirección global usando diferentes puertos. *NAT Overloading* es conocido como **PAT** y es una forma de NAT dinámico.

Una de las principales características de NAT es PAT, es cual es conocido también como *Overload* en el IOS de cisco. PAT permite traducir múltiples direcciones locales en una sola dirección global. La figura de abajo muestra un ejemplo de PAT.




NAT Table

| Inside Local IPv4 Address | Inside Global IPv4 Address |
|---|---|
| 10.6.1.2:2031 | 171.69.68.10:2031 |
| 10.6.1.6:1506 | 171.69.68.10:1506 |
| 10.6.1.6:131 | 171.69.68.10:2032 |

Figura 39. PAT

PAT usa números de puertos para distinguir las traducciones de las direcciones privadas en una sola dirección global. Debido a que el puerto esta codificado en 16 bits, el número total de sesiones internas que NAT puede traducir es teóricamente de 65536.

## NAT (*Network Address Translation*)

Una dirección IP privada se debe traducir en una dirección IP global cuando se quiere comunicación con una red exterior. La traducción se puede traducir de manera estática o dinámica.

La figura de abajo muestra a un Router traduciendo una dirección local a una global.

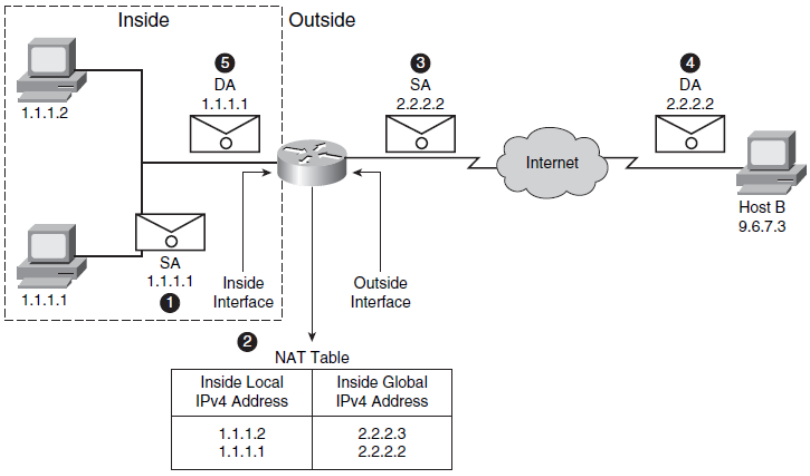



Figura 40. NAT

**Paso 1.-** El host 1.1.1.1 abre una conexión con el host B.

**Paso 2.-** EL primer paquete que el *Router* recibe del host 1.1.1.1 hace que el *Router* revise su tabla de NAT.

**Paso 3.-** El *Router* reemplaza la dirección local 1.1.1.1 con su traducción (dirección global 2.2.2.2) y reenvía el paquete.

**Paso 4.-** el host B recibe el paquete y responde al host 1.1.1.1 usando la dirección global 2.2.2.2

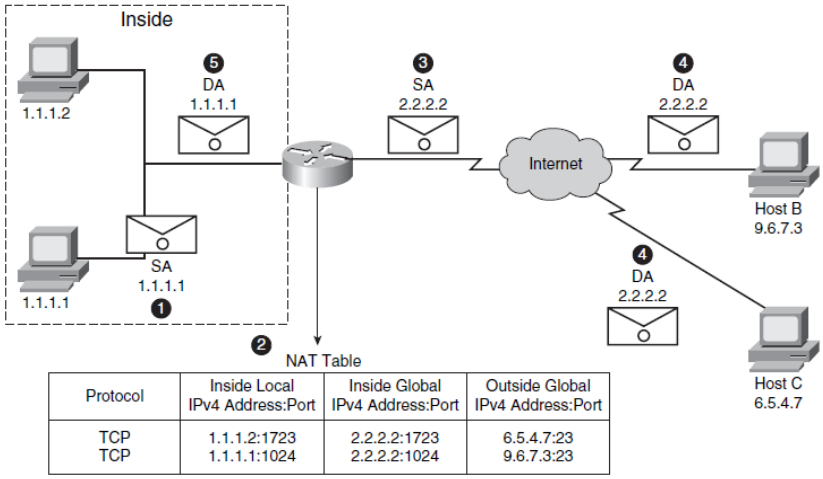
**Paso 5.-** Cuando el *Router* recibe el paquete con la dirección global, hace una traducción inversa y el paquete es enviado a la dirección local 1.1.1.1. el host 1.1.1.1 recibe el paquete y continua la conversación.

## PAT (*Port Address Translation*)

Es posible conservar direcciones globales haciendo que el *Router* use una sola dirección global para muchas direcciones locales. Cuando el ***NAT overloading*** es configurado, el *Router* maneja suficiente información de protocolos de capas más arriba como por ejemplo, números de puertos de TCP y UDP para traducir las direcciones privadas o locales en una sola dirección global.

**Cuando múltiples direcciones locales están mapeadas a una sola dirección global, los números de puertos de TCP o UDP hacen que cada host dentro de la red local se distinga de otros.**

La figura de abajo muestra la operación de PAT




| Protocol | Inside Local IPv4 Address:Port | Inside Global IPv4 Address:Port | Outside Global IPv4 Address:Port |
|---|---|---|---|
| TCP | 1.1.1.2:1723 | 2.2.2.2:1723 | 6.5.4.7:23 |
| TCP | 1.1.1.1:1024 | 2.2.2.2:1024 | 9.6.7.3:23 |

Figura 41. PAT

Host B y host C creen que están hablando con un solo host con dirección 2.2.2.2, pero, en realidad están hablando con host diferentes. El numero de puerto es la diferencia.

El Router ejecuta el siguiente proceso cuando ejecuta PAT.

**Paso 1.-** el host 1.1.1.1 abre una conexión con el host B

**Paso 2.-** el primer paquete que el *Router* recibe del host 1.1.1.1 hace que el *Router* revise su tabla NAT.

**Paso 3.-** el Router reemplaza la dirección local 1.1.1.1 con la dirección global elegida y envía el paquete.
**Paso 4.-** el host B recibe el paquete responde al host 1.1.1.1 usando la dirección global 2.2.2.2

**Paso 5.-**cuando el *Router* recibe el paquete con la dirección global, el *Router* realiza una búsqueda en su tabla NAT. Usando la dirección global y el puerto el *Router* realiza una traducción de vuelta a la dirección local 1.1.1.1 y reenvía el paquete al host 1.1.1.1. El host 1.1.1.1 recibe el paquete y continua con la conversación.

## Configuración de PAT

Para configurar PAT en una red para tener salida a internet se configuran los siguientes pasos.

Paso 1.- definir una lista de acceso que permita el direccionamiento que será traducido.

RouterX(config)# **access-list** access-list-number **permit** source [sourcewildcard]

Paso 2.- Relacionar la lista de acceso creada con la interface de salida.

RouterX(config**)# ip nat inside source list** access-list-number **interface** interface **overload**

Paso 3.- especificar la interface local y la interface global

RouterX(config)# **interface** type number
RouterX(config-if)# **ip nat inside**
RouterX(config-if)# **interface** type number
RouterX(config-if)# **ip nat outside**

Paso 4.- Revisar las traducciones con el comando

**show ip nat translations**

**con el comando ***cl ear ip nat translations*** limpiamos la tabla de traducciones**

PRACTICA PAT

Nombre del participante_____RPE_____

El ISP asignó una única dirección IP, 209.165.201.18, a su empresa para usarla en la conexión a Internet del *Router Gateway* de la empresa al ISP. Usará la traducción de la dirección del puerto (PAT) para convertir varias direcciones internas en la única dirección pública utilizable. Se probará, se verá y se verificará que se produzcan las traducciones y se interpretarán las estadísticas de NAT/PAT.

Armar la siguiente topología en *packet tracert*

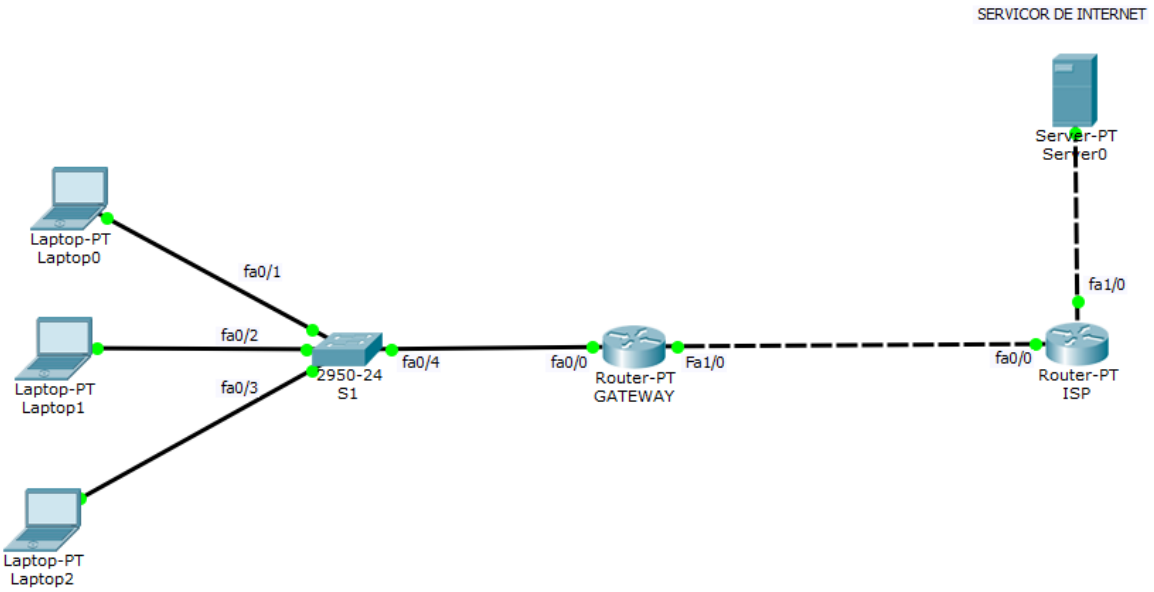



Tabla de direccionamiento

| Dispositivo | Interface | Dirección IP | Mascara | Gateway |
|---|---|---|---|---|
| GATEWAY | fa0/0 | 192.168.1.1 | 255.255.255.0 | |
| | Fa1/0 | 209.165.201.18 | 255.255.255.252 | |
| ISP | Fa0/0 | 209.165.201.17 | 255.255.255.252 | |
| | Fa1/0 | 192.31.7.1 | 255.255.255.0 | |
| LAPTOP 0 | Nic | 192.168.1.20 | 255.255.255.0 | 192.168.1.1 |
| LAPTOP 1 | Nic | 192.168.1.21 | 255.255.255.0 | 192.168.1.1 |
| LAPTOP 2 | Nic | 192.168.1.22 | 255.255.255.0 | 192.168.1.1 |

# 7.- IMPLEMENTACION DE VLANS Y TRONCALES

Una VLAN (virtual LAN) es un dominio lógico de *broadcast* que puede abarca múltiples segmentos físicos LAN. Proporcionan segmentación y son usadas para agrupar hosts que tienen un conjunto de requerimientos en común, independientemente de su ubicación física. Una VLAN tiene los mismos atributos que una LAN física., excepto que la primera permite agrupar hosts aun cuando no estén ubicados físicamente en la misma LAN física. Una VLAN incluso permite agrupar puertos en un *switch* para que se pueda limitar el tráfico *unicast*, *multicast* o *broadcast*. El tráfico que se genera en una VLAN en particular, solo pasa por los puertos que pertenecen a esa VLAN.

"Un dominio de broadcast son todos los dispositivos que están dentro del mismo segmento de red. El trafico *broadcast* no puede ser ruteado, es por eso que un dominio de *broadcast* es contenido siempre por la interfaz de un *Router*."

## 7.1. ENTENDIENDO UNA VLAN

Entender cómo opera una VLAN y que protocolos están asociados es importante para configurar, verificar y atacar problemas en redes *switching*.

Una red que tiene un diseño pobre, incrementa costos de soporte y reduce la disponibilidad de servicios. Algunos de los problemas que se presentan en una red de diseño pobre son los siguientes.

DOMINIOS DE FALLAS.- cuando las fronteras entre capa 2 y capa 3 no están claramente definidas, las fallas en cierta área de la red puede tener efectos de largo alcance.

DOMINIOS DE *BROADCAST*.- el broadcast existe en todas las redes. Muchas aplicaciones requieren broadcast para funcionar apropiadamente, por eso es imposible eliminarlo completamente. Para se necesita definir fronteras para prevenir serios problemas de *broadcast*, para eliminar su impacto negativo.

DIFICULTAD PARA ADMINISTRAR LA RED.- un diseño pobre de la red ya sea por mala organización o documentación pobre y la falta de identificación del flujo de tráfico culmina en dificultad para dar soporte y el tiempo de resolución de problemas se incrementa y la tarea se vuelve difícil.

La figura de abajo muestra una red en un solo dominio de broadcast. Las VLANS pueden ayudar a aliviar algunos de los problemas asociados con este diseño.

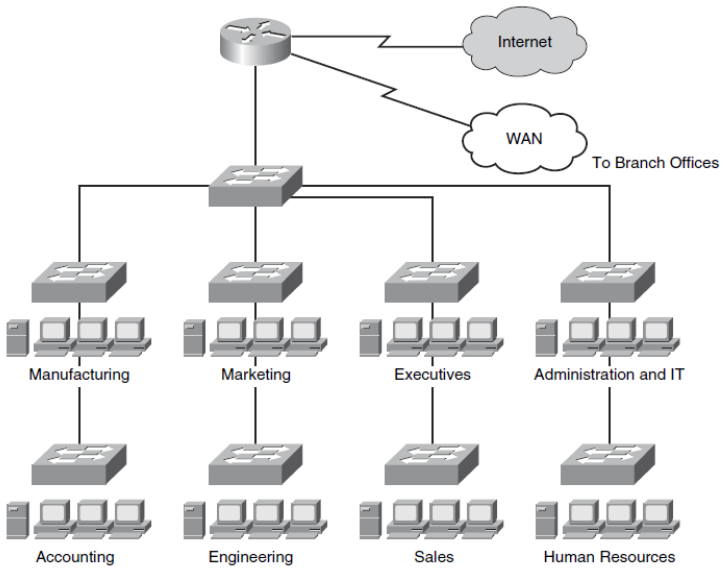



Figura 42.- dominio de *broadcast*

En una red *switching*, las VLANs proporcionan segmentación y organización. Usando la tecnología VLAN se puede agrupar puertos de diferentes *switches* en una "comunidad" como por ejemplo, trabajadores de un mismo departamento que comparten las mismas aplicaciones o DEIs que son alojados en un mismo servidor.

"VLAN puede existir en un solo *switch* o puede abarcar múltiples *switches*, en una misma ubicación física o separada geográficamente."

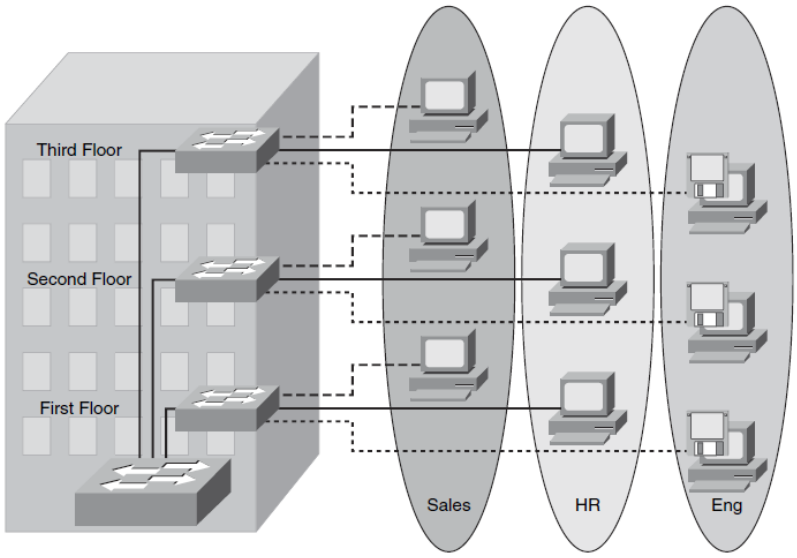



Figura 43.- VLANS abarcando múltiples *switches*

## 7.2. Aplicando direccionamiento IP en una red *switching*

Diseñar un modelo eficaz de red incluye una estructura ideal para superponer un esquema de direccionamiento IP jerárquico. A continuación se presentan algunas recomendaciones.

- Diseñar el esquema de direccionamiento IP para que bloques de $2^n$ direcciones (4,8,16,32, etc.) puedan ser asignadas a una subred en una cierta distribución de hosts.
- Tener un solo segmento de direccionamiento IP que corresponda a una sola VLAN. Cada VLAN es un dominio de *broadcast* separado.
- Cuando sea posible, tener subredes con el mismo número de direcciones para prevenir VLSM (*variable-lenght subnet mask*). Esto ayuda a minimizar errores y confusiones cuando se busquen errores o se configure nuevos segmentos.

En la figura siguiente se muestra un modelo de arquitectura de direccionamiento IP asignado a varios grupos en una red empresarial. Nótese que cada departamento tiene su propia subred y cada subred está asignada a una sola VLAN.

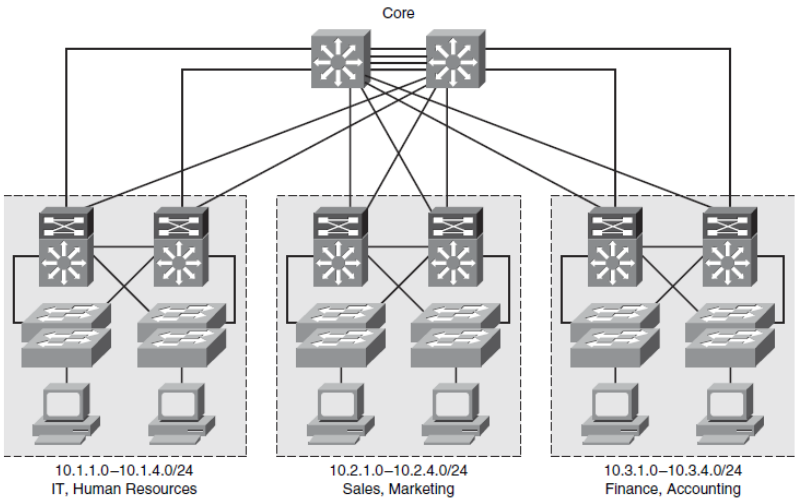



Figura 44.- direccionamiento IP por VLAN

## 7.3. OPERACIÓN DE UNA VLAN

Cada VLAN que se configura en un *switch* implementa aprendizaje de las direcciones, reenvío y toma de decisiones así como mecanismos de prevención de *loops* como si cada VLAN fuera un *switch* independiente.

Cada *switch* implementa VLANs para restringir el reenvío de tráfico solo a los puertos destino que estén en la misma VLAN que l puerto origen. Así que cuando una trama llega al puerto del *switch*, el *switch* debe retransmitir la trama solo a los puertos que pertenezcan a la misma VLAN. En esencia, un a VLAN que está operando en un *switch* limita las transmisiones *unicast*, *multicast* y *broadcast*. El tráfico originado en una VLAN llega solo a otros puertos en la misma VLAN.

Los puertos de un *switch* se configuran para pertenecer a una VLAN. Hay 3 tipos de afiliación para asignar un puerto a una VLAN.

*Static* VLAN: el administrador configura el puerto de manera manual para que pertenezca a una VLAN

Dynamic VLAN: los *switches* cisco soportan *dynamic* VLAN usando VLAN *Membership Policy Server* (VMPS). Es una manera de asignar VLANs a ciertos puertos a través de un servidor VMPS. Este servidor contiene un mapeo de direcciones MAC a VLANs asignadas.

*Voice* VLAN: La VLAN de voz es una VLAN creada en la configuración de default de algunos *switches* y está especialmente diseñada para el tráfico de voz ya que tiene una modificación en los parámetros de CoS.

## 7.4.- CONFIGURACION DE VLANs

Antes de comenzar con la creación de VLANs, se debe saber el número máximo de VLANs que el *switch* soporta. Un *switch* cisco de la serie catalyst puede administrar hasta 255 VLANS y se pueden usar los VID desde el 1 hasta el 4094. Las VIDs del 1002 al 1005 están reservadas.

```
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

Por default los *switches* tienen configurada la VLAN 1, que es la VLAN nativa de fábrica. Cisco utiliza la VLAN 1 para enviar anuncios VTP y CDP.

Para que se pueda tener comunicación remota con un *switch* para propósitos de administración, el *switch* debe tener una dirección IP. Esta dirección IP debe estar asignada a la VLAN de administración, que por default es la VLAN 1.

# COMANDOS PARA CONFIGURAR UNA VLAN EN UN SWITCH CISCO

| Command/Variable | Description |
|---|---|
| **vlan** *vlan-id* | ID of the VLAN to be added and configured. For *vlan-id*, the range is 1 to 4094 when the enhanced software image is installed and 1 to 1005 when the standard software image is installed. Do not enter leading zeros. You can enter a single VID, a series of VIDs separated by commas, or a range of VIDs separated by hyphens. |
| **name** *vlan-name* | (Optional) Specify the VLAN name, an ASCII string from 1 to 32 characters that must be unique within the administrative domain. |

Figura 45.- comandos para configurar una VLAN

Se usa el comando VLAN en modo de configuración global para crear una VLAN.

```
SwitchX# configure terminal
SwitchX(config)# vlan 2
SwitchX(config-vlan)# name switchlab99
```

Se usa el comando "**no**" anteponiéndolo para borrar la VLAN.

Para agregar una VLAN a la base de datos de VLANs, se asigna un número a la VLAN. La VLAN 1 es la VLAN de fábrica en un switch. El rango normal de VLANs están identificadas con un numero entre 1 y 1001, las VLANs entre 1002 a 1005 están reservadas para *Token Ring* y FDDI.

Los VIDs del 1 al 1005 están escritos en un archivo llamado vlan.dat (base de datos de VLANs). Se puede visualizar la lista de VLANs usando el comando ***show vlan*** en modo privilegiado. El archivo vlan.dat se almacena en la memoria *flash*.

Se usa el comando ***show vlan id*** [*vlan-number*] para visualizar información acerca de una VLAN en particular.

# ASIGNACION DE PUERTOS A VLANs EN SWITCH CISCO

Después de crear una VLAN, se puede asignar un puerto o número de puertos a una VLAN. Un puerto puede pertenecer solo a una VLAN al mismo tiempo.

En los *switches* cisco se configura la asignación de puerto a una VLAN desde la configuración de modo interface usando el comando *switchport access*. Por default todos los puertos pertenecen a la vlan 1.

```
SwitchX# configure terminal
SwitchX(config)# interface range fastethernet 0/2 - 4
SwitchX(config-if)# switchport access vlan 2

SwitchX# show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------
1    default                          active    Fa0/1
2    switchlab99                      active    Fa0/2, Fa0/3, Fa0/4
```

Figura 46.- configuración de un puerto de acceso.

Alternativamente, se usa el comando **show *interfaces*** *[interface]* ***switchport*** en modo privilegiado para visualizar la información de VLAN de una interface en particular, como se muestra a continuación.

```
SwitchX# show interfaces fa0/2 switchport
Name: Fa0/2
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 2 (switchlab99)
Trunking Native Mode VLAN: 1 (default)
--- output omitted ----
```

Figura 47.- información de VLAN de una interface en particular.

PRACTICA VLANS

ACTIVIDAD.- CONFIGURACIÓN DE VLANS EN UN SWITCH CISCO

Objetivo: configurar dos VLANS en un switch y configurar puertos asociados a dichas VLANs para probar conectividad.

1.-En un *switch* configurar vlan 10 y vlan 20

2.- Configurar el puerto 1 y 2 del *switch* de acceso a la vlan 10. El puerto 3 y 4 de acceso a la vlan 20

3.- Conectar una laptop con dirección IP 10.0.0.1/24 al puerto1 y otra laptop al puerto 2 con una IP 10.0.0.2/24.

4.- hacer ping desde Laptop 1 hasta Laptop 2 y verificar conectividad.

5.- cambiar laptop 2 al puerto 3 y verificar conectividad.

HACER UNA PAUSA Y PASAR AL PUNTO SIGUIENTE HASTA QUE SE LE INDIQUE

7.- integrar un nuevo *switch* con la configuración idéntica al primero y conectarlos entre ellos a través del puerto 2 con un cable *cross-over* y otro cable *cross over* a través del puerto 4.

Conectar una laptop en el *switch* 1 y una laptop en el *switch* 2. Probar conectividad entre VLANs

¿Que pude resumir con esta práctica?

# 7.5.- TRONCALES CON 802.1Q

Un puerto normalmente lleva solo el tráfico de una sola VLAN a la cual pertenece. Para que una VLAN pueda expandirse a través de múltiples *switches*, una troncal es requerida para conectar *switches*.

"Una troncal puede llevar tráfico de múltiples VLANs a través de dos *switches*"

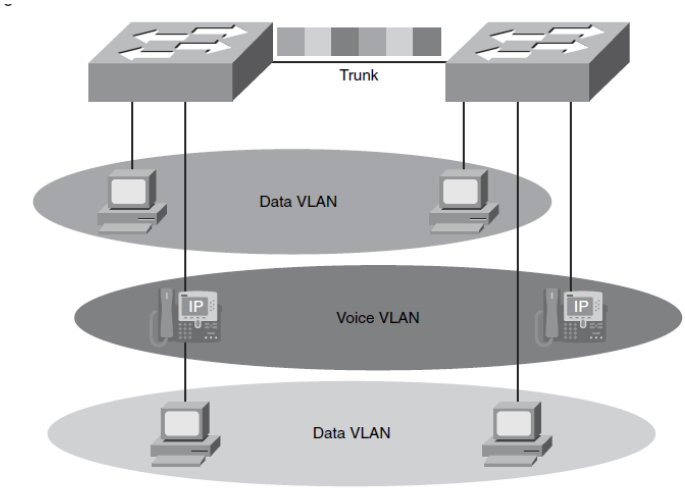



Figura 48.- VLAN *Trunk*

Una troncal es un enlace punto a punto entre dos o más interfaces de *switches* y otro dispositivo de red como otro *switch* o un *Router*. Una troncal lleva tráfico de múltiples VLANs sobre un solo enlace físico y permite extender la VLAN a través de la red entera.

La gran mayoría de *switches* de diferentes fabricantes soportan IEEE 802.1Q para interfaces *Fast Ethernet* y *Gigabit Ethernet*. Cisco posee un mecanismo para hacer troncales propietario llamado *Inter-Switch Link* (ISL).

802.1Q también conocido como dot1q es un proyecto del grupo de trabajo 802 de la IEEE que permite el paso de múltiples VLANs a través de un medio físico llamado troncal sin problemas de interferencia entre ellas usando un método de etiquetado de tramas. Cada trama pasa por la troncal con una etiqueta que la identifica dependiendo de la VLAN a la que pertenezca. Este identificador es conocido VLAN ID (VID).

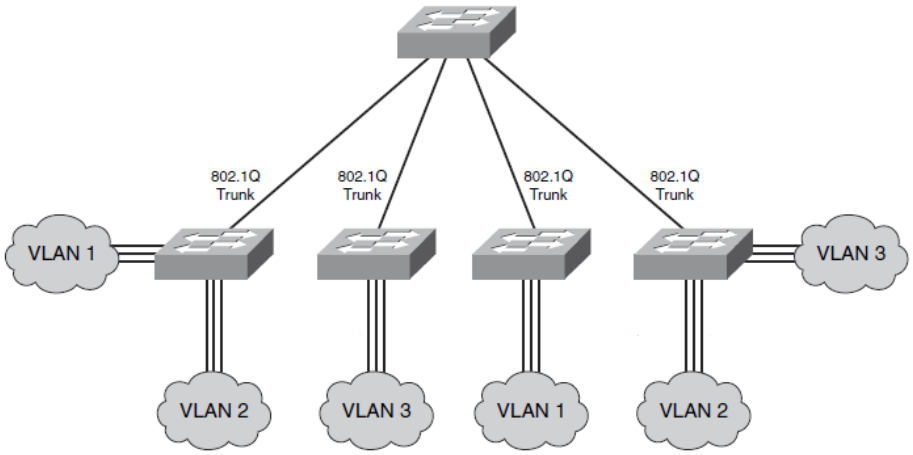



Figura 49.- troncales 802.1Q

"Las interfaces de un *switch* soporta diferentes modos. Se puede configurar una interface como troncal o no troncal. Las interfaces no troncales son conocidas como puertos de acceso en los *switches* cisco. Otros fabricantes usan la terminología *tagged* para los puertos troncales y *untagged* para los puertos de acceso."

Cada puerto 802.1Q es asignado a una troncal, y los puertos en dicha troncal estaban en por default en una VLAN antes de ser configurados como troncal, la VLAN nativa. Los puertos de la troncal necesitan estar en la misma VLAN nativa para poder establecer 802.1Q.Una VLAN nativa es usada en 802.1Q para enviar tramas sin etiquetar (*untagged*)como por ejemplo CDP, VTP etc. Por default la VLAN nativa es la VLAN 1 (VID 1)y todas las tramas sin etiquetar viajan por esta VLAN.

## TRAMAS 802.1Q

802.1Q usa un mecanismo interno de etiquetado (*tagging*) que inserta una etiqueta de 4 bytes en la trama original de Ethernet.

Es responsabilidad del *switch* revisar la etiqueta de 4 bytes y determinar donde entregar la trama. El campo *Ether Type* de 0x8100 indica a los dispositivos que la trama tiene una etiqueta 802.1Q. El campo de VID es el identificador de la VLAN, indica a que VLAN pertenece la trama.

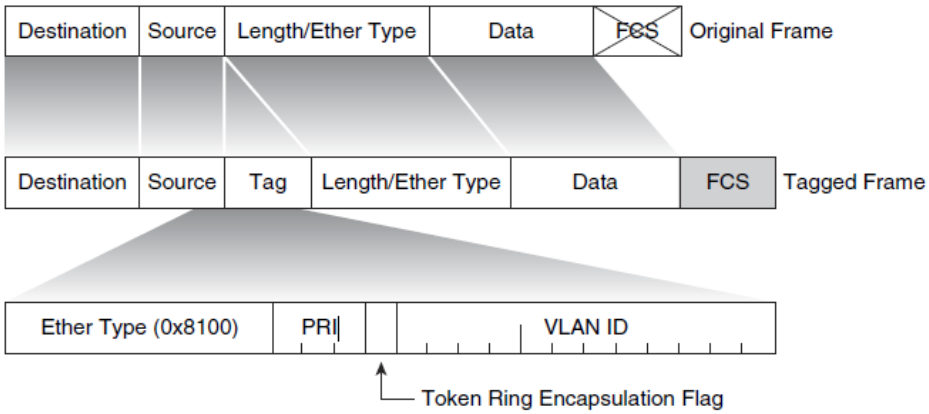



Figura 50.- formato de la trama 802.1Q

## VLAN NATIVA 802.1Q

Un puerto troncal 802.1Q y sus puertos asociados a la troncal tienen una VLAN nativa. 802.1q no etiqueta las tramas por la VLAN nativa. Por eso, los hosts ordinarios pueden leer las tramas nativas sin etiquetar pero no pueden leer ningún otra trama por que las tramas están etiquetadas. La figura de abajo muestra una trama de la VLAN nativa siendo distribuida a través de la red troncalizada.

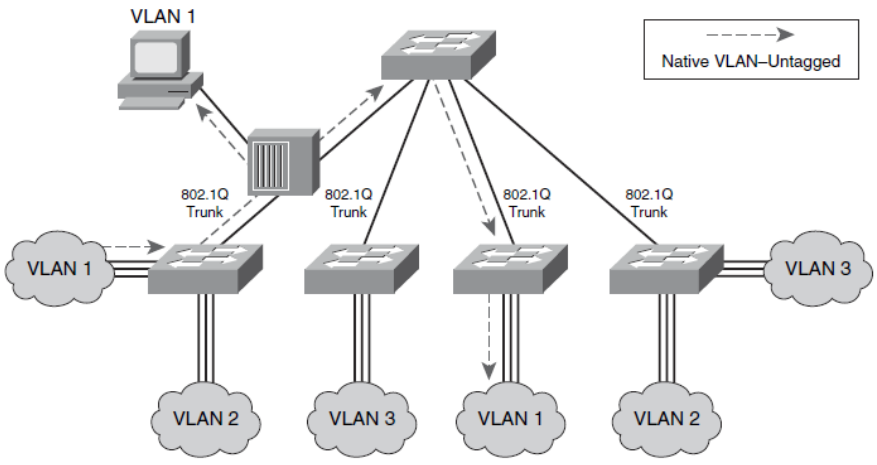



Figura 51.- trama sin etiquetar

# 7.6. CONFIGURACION DE TRONCALES 802.1Q

El protocolo 802.1Q o dot1q lleva tráfico de múltiples VLANs a través de un solo enlace físico entre *switches multivendor*.

Se deben de cuidar algunos aspectos para poder configurar 802.1Q, algunos aspectos son los siguientes:

- Asegurarse que la VLAN nativa en los puertos 802.1q sea la misma en las dos puntas. Si es diferentes pueden presentarse algunos problemas.
- Las tramas de la VLAN nativa no van etiquetadas.

Se usa el comando ***switchport mode*** en modo interface para configurar una interface *Fast Ethernet* o *gigabit Ethernet* para configurar la interface en modo troncal.

Hay cuatro opciones para el comando *switchport mode*

| Parameter | Description |
|---|---|
| trunk | Configures the port into permanent 802.1Q trunk mode and negotiates with the connected device to convert the link to trunk mode. |
| access | Disables port trunk mode and negotiates with the connected device to convert the link to nontrunk. |
| dynamic desirable | Triggers the port to negotiate the link from nontrunk to trunk mode. The port negotiates to a trunk port if the connected device is in trunk state, desirable state, or auto state. Otherwise, the port becomes a nontrunk port. |
| dynamic auto | Enables a port to become a trunk only if the connected device has the state set to trunk or desirable. Otherwise, the port becomes a nontrunk port. |

Figura 52.- *switchport mode*

PRACTICA TRONCALES

CONFIGURACIÓN DE TRONCALES EN UN SWITCH CISCO

Configurar dos VLANS en un *switch* y configurar puertos asociados a dichas VLANs para probar conectividad.

1.-En un *switch* configurar vlan 10 y vlan 20

2.- Configurar el puerto 1 y 2 del *switch* de acceso a la vlan 10. El puerto 3 y 4 de acceso a la vlan 20

3.- Conectar una laptop con dirección IP 10.0.0.1/24 al puerto1 y otra laptop al puerto 2 con una IP 10.0.0.2/24.

4.- hacer ping desde Laptop 1 hasta Laptop 2 y verificar conectividad.

5.- cambiar laptop 2 al puerto 3 y verificar conectividad.

7.- integrar un nuevo *switch* con la configuración idéntica al primero y conectarlos entre ellos a través de un puerto troncal.

Verificar conectividad entre VLANS.

# 8.- RUTEO DE VLANS

El ruteo es el proceso de determinar a donde mandar paquetes que van destinados a una red externa. Los Routers reúnen y mantienen información de ruteo a fin de lograr la transmisión y recepción de paquetes de manera correcta. Para el trafico que tiene que cruzar de una VLAN a otra, un proceso de capa 3 es necesaria.

A continuación se verá el proceso de operación de ruteo inter-VLAN usando el método ***Router on-a-stick.***

## 8.1. Ruteo de VLANs

La comunicación inter-vlan ocurre entre dominios de *broadcast* a través de un dispositivo de capa 3. En un ambiente de VLANs, las tramas son conmutadas solo entre puertos dentro de la mismo dominio de broadcast. Las VLANs otorgan particiones de red y separación de tráfico en capa 2. La comunicación inter-vlan no puede ocurrir sin un dispositivo de capa 3 como un Router. Se usa el IEEE 802.1Q para habilitar una troncal en una sub interface de un Router.

Ejemplo de una configuración ***Router on-a-stick***

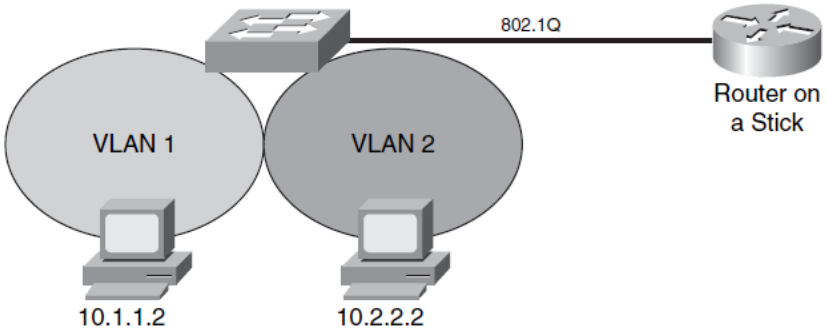



Figura 53.- *switchport mode*

El Router puede recibir paquetes en una VLAN y reenviarlos a otra VLAN. Para implementar ruteo inter-vlan, el Router debe saber cómo alcanzar todas las vlans que se requieren interconectar. Cada VLAN debe tener una conexión por separado al Router, y se debe habilitar una troncal 802.1Q en esas conexiones. El Router ya conoce las redes que tiene directamente conectadas. El Router debe aprender rutas a redes que no tiene conectadas directamente.

"Para soportar troncales 802.1Q, se debe de subdividir la interface física del Router en múltiples sub interfaces lógicas, una por VLAN. El resultado: interfaces lógicas llamadas sub interfaces."

Abajo se muestra un ejemplo.

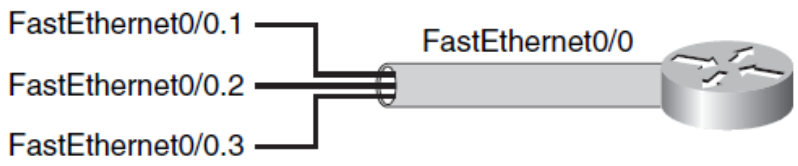



Figura 54.- Sub interfaces

"Sin la división de sub interfaces, se tendría que asignar una interface física por separado a cada VLAN."

En la figura de arriba, la interface Fa0/0 es dividida en múltiples sub interfaces: *FastEthernet* 0/0.1, *FastEthernet* 0/0.2 y *FastEthernet* 0/0.3.

# 8.2. Configurando ruteo inter-vlan

Para habilitar ruteo entre LANs en un switch, se necesita habilitar la configuración inter-VLAN.

En la figura de abajo, la interface *Fast Ethernet* 0/0 es dividida en múltiples sub interfaces. Cada sub interface representa al Router en cada una de las VLANs para la cual esté ruteando.

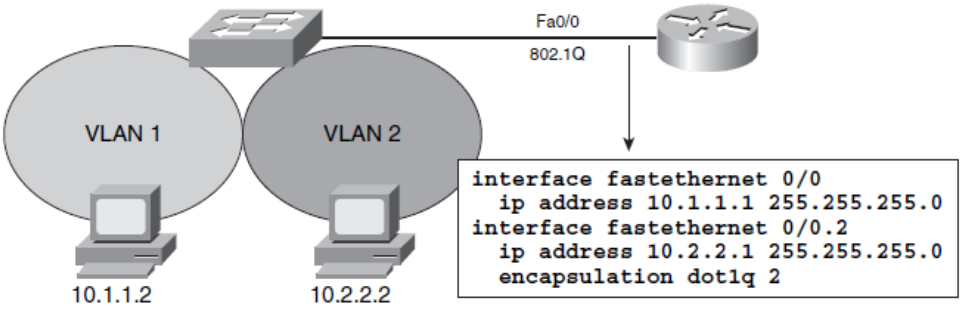



Figura 55.- Configuración inter-vlan

Se usa el comando ***encapsulation dot1q [vlan identifier]*** en cada sub interface para habilitar las troncales dot1q. Los números de interfaces no tienen que ser necesariamente el mismo que el numero de vlan dot1q. sin embargo, el administrador puede usar el mismo número de interface y de vlan para facilitar la administración.

Las tramas de la vlan nativa en 802.1q no llevan etiqueta. Por eso, la sub interface de la vlan nativa es configurada con el comando ***encapsulation 802.1q vlan identifier native.*** Se debe estar seguro que la vlan nativa asignada a la sub interface coincida con la vlan nativa en el *switch* al cual se conecta. Cada sub interface tendrá una única dirección IP para la vlan a la cual está asociado. Esta dirección será usa como *Gateway* para los hosts en esa vlan.

PRACTICA RUTEO INTER-VLAN

Nombre del participante_____RPE_____

Con la siguiente topología, simular lo siguiente.

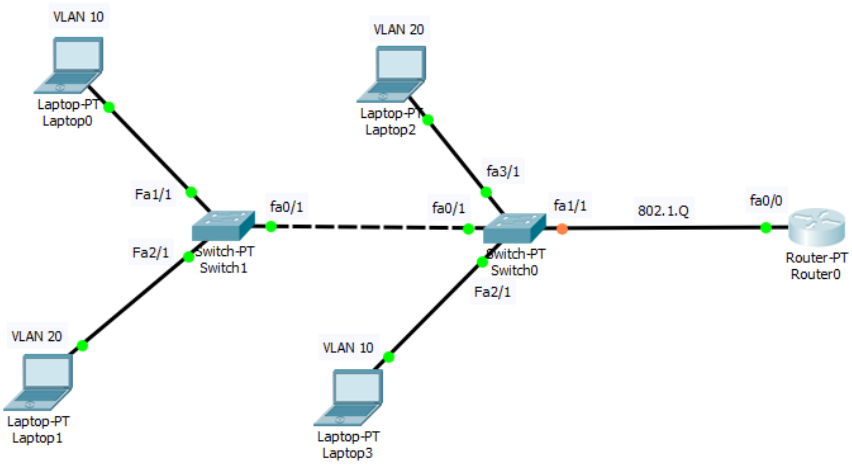
Configurar Vlans como se indica en la topología. Hacer una troncal entre ambos *switches*. Probar comunicación entre host de la misma Vlan.

Configurar 802.1q entre el *switch* y *Router* y configurar sub interfaces. Cada una correspondiente a una vlan. Asignar el siguiente direccionamiento.

Red 10.25.178.0 para vlan 10

Red 10.25.178.64 para vlan 20

Probar comunicación entre host de diferentes VLAN




Escriba el por qué cree que sea útil el ruteo Inter-Vlan

# 9. SWITHCES MULTICAPA o MLS

La implementación de ruteo entre VLAN usando *router-on-a-stick* requiere solamente una interfaz física en un *Router* y una interfaz en un *switch*, lo que simplifica el cableado del *Router*. Sin embargo, en otras implementaciones de ruteo entre VLAN, no se necesita un *Router* dedicado.

## 9.1. *Switches* L2 L3

Una red de capa 2 es definida como un dominio de *broadcast*. Una red de capa 2 incluso pude existir como una VLAN dentro de uno o mas *switches*. Esencialmente una VLAN está aislada de otras haciendo que los paquetes en una VLAN no puedan cruzar a otras. Para transportar paquetes entre VLAN se debe usar un dispositivo de capa 3. Tradicionalmente, está ha sido la función de un *Router*. El *Router* debe tener una conexión física o lógica para cada VLAN para que se puedan reenviar paquetes entre ellas. Como ya se mencionó con anterioridad, esto se conoce como ruteo inter-vlan.

El ruteo inter-vlan puede ser desarrollado por un Router externo que conecta a cada una de las VLANs en un *switch*. Se puede usar conexiones físicas separadas, o el *Router* puede accesar a cada VLAN a través de un solo enlace troncal. La figura de abajo ilustra este concepto. Un *Router* con múltiples conexione, una sola conexión troncal o un *switch* MLS.

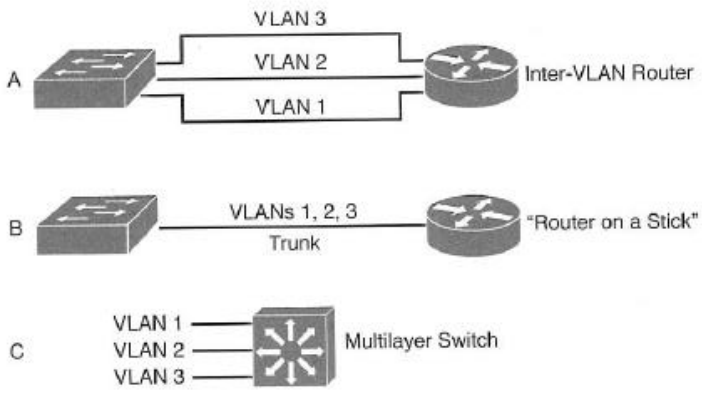



Figura 56.- Conexiones para ruteo inter-vlan

La parte C de la figura muestra cómo funcionan los *switches* y *Routers* se combinan para funcionar como un solo dispositivo. Un *switch* multicapa haciendo que un *Router* externo ya no sea necesario.

### TIPOS DE INTERFACES

Un *switch* multicapa desempeña funciones de ambas capas, capa 2 y capa 3. La función de *switch* ocurre entre interfaces que son asignadas a capa 2, es decir, VLANs o troncales. La función de *Router* ocurre entre cualquier tipo de interface, así como una interface que puede tener direccionamiento de capa 3 asignado.

Como con un *Router*, un *switch* multicapa o MLS puede tener asignada una dirección de capa 3 a una interface física. Incluso es posible asignarle una dirección de capa 3 a una interface lógica que representa a toda una VLAN. Esto es conocido como ***Switched virtual interface (SVI)***. Se debe tener en mente que la dirección de capa 3 que se configure se vuelve el *default Gateway* para cualquier *host* que esté conectado en esa VLAN. Este host utiliza esa interface de capa 3 para comunicarse afuera de su dominio de broadcast.

## 9.2 Configurando inter-vlan routing con MLSs

El ruteo inter-vlan primeramente requiere que el MLS sea habilitado para los protocolos de ruteo. Esto se hace con el comando "***ip routing***" en el modo de configuración global. Aparte de esto, se debe configurar rutas estáticas o ruteo dinámico.

Una interface en modo capa 2 o capa 3 se configura , dependiendo del uso de la interface usando el comando ***switchport*** en modo interface.

Se puede desplegar la información de la interface y verificar en qué modo está configurada actualmente usando el comando ***show interface*** *[type member/module/number]* ***switchport.***

"Si en la información desplegada se muestra *switchport*: *Enable*, indica que la interface esta en modo capa 2. Si se muestra como *Disabled*, el puerto está configurado en capa 3"

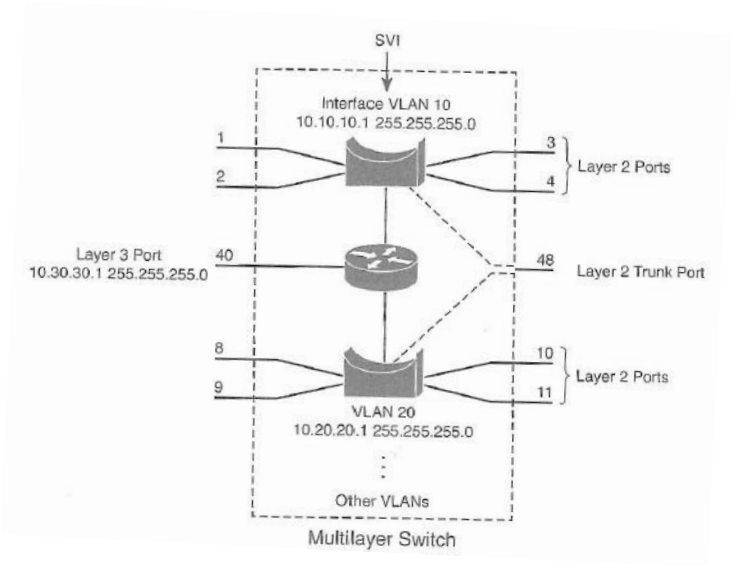



Figura 56.- Conexiones para ruteo inter-vlan

En la figura 56 se muestra los diferentes tipos de interfaces que se pueden usar con un *switch* MLS.

# 9.3 Configurando interfaces de MLS

## CONFIGURACION DE PUERTO EN CAPA 2

Si una interface esta en nodo capa 3 y se necesita reconfigurarla para capa 2 se usa el siguiente comando.

*Switch(config)#**interface** type member/module/number*
*Switch(config)#**switchport***

"El comando ***switchport*** pone a la interface en modo de capa 2, así que esta interface se puede usar para hacer troncales con otro *switch* o acceso a alguna VLAN."

# CONFIGURACION DE PUERTO EN CAPA 3

Los puertos físicos de un *switch* también pueden operar en capa 3, donde las direcciones de capa 3 pueden ser asignadas y entonces puede ocurrir ruteo. Una interface en modo capa 3 se configura de la siguiente manera.

*Switch(config)#**interface** type member/module/number*
*Switch(config-if)#**no switchport***
*Switch(config-if)#**ip address** ip-address mask*

El comando ***no switchport*** toma el puerto desde la capa 2 y lo pone en capa 3. Así se le puede configurar una dirección IP.

# CONFIGURACION DE PUERTOS SVI

En un *switch* multicapa, también se puede habilitar la función de capa 3 para una VLAN entera en el *switch*. Esto permite a una dirección de red ser asignada a una interface lógica. Esto es útil cuando un *switch* tiene muchos puertos asignados a una VLAN en común, y el ruteo es necesario para dicha VLAN.

En la figura 57 se puede ver como una dirección IP es aplicada al SVI llamada VLAN 10. Se debe notar que la SVI no tiene una conexión física por si misma hacia el exterior; para alcanzar una salida, la VLAN 10 se debe extender a través de un puerto de capa 2 o troncal hacia el exterior.

La interface lógica de capa 3 es conocida como SVI. Sin embargo, cuando ésta es configurada, se usan mas parámetros de configuración como el *vlan-id*, nombre, en comparación como una interface física.

La VLAN se debe definir y activar en el *switch* antes de que la SVI pueda ser usada. Se debe asegurar que la nueva *VLAN-Interface* también se puede habilitar con el comando de configuración de ***interface no shutdown.***

Se debe tener cuidado, un SVI no se puede volver activa hasta que al menos un puerto de capa 2

**"NOTA: La VLAN y el SVI se configuran por separado, aunque estos inter operan entre sí. Creando y configurando el SVI no se crea o configura la VLAN. Se debe definir cada uno independientemente."**

Un ejemplo de los siguientes comando se muestran con la VLAN 100, creándola y definiéndola como una SVI.

```
Switch(config)#vlan 100
Switch(config-vlan)#name Example_vlan
Switch(config-vlan)#exit

Switch(config)#interface vlan 100
Switch(config-if)#ip add 192.168.100.1 255.255.255.0
Switch(config-if)#no shutdown
```

## 9.4 DHCP con *swithces* multicapa

Cuando un *switch* es configurado con una dirección de capa 3 en una interface, éste se vuelve el *default gateway* de los host conectados a esa VALN o subred a la cual pertenece la interface. ¿Cómo le hacen esos host para usar la interface de capa 3 como su default Gateway?, así como, ¿Cómo saben esos host que dirección IP usar para sus propias identidades dentro de la red?.

Los host pueden tener una dirección IP asignada de manera manual o estática, mascara de subred , Gateway, etc. Esto podría ser apropiado para algunos dispositivos como lo son servidores, los cuales necesitan direcciones estables y reservadas, pero para la mayoría de hosts, la asignación de direcciones estáticas puede volverse una tarea administrativa muy difícil.

En lugar de eso, el protocolo *Dynamic Host Configuration Protocol* (DHCP) es usualmente usado para proporcionar direccionamiento IP a los *host* de manera dinámica a cualquiera que use este protocolo. DHCP está definido en el RFC2131 y está diseñado bajo la arquitectura de cliente servidor. Los host solicitan una dirección IP valida y el servidor otorga una dirección de un conjunto de direcciones disponibles.

Suponga que se conecta a una red, pero un no tienen una dirección IP. Se necesita solicitar una dirección valida vía DHCP. ¿Cómo se puede mandar un paquete al servidor DHCP sin tener una dirección IP válida para usarla como dirección fuente?. La respuesta está en la negociación DHCP, la cual se explica en los siguientes 4 pasos.

1.- **El cliente envía un mensaje "*DHCP Discover*" como *broadcast***: aun sin una dirección IP fuente valida, el cliente puede enviar un mensaje con una dirección broadcast para encontrar al servidor DHCP que se encuentra "escuchando".

2.- **El servidor DHCP contesta con un mensaje "*DHCP Offer*"**: la oferta contiene una dirección IP valida, mascara de subred, Gateway y otros parámetros.

El servidor incluso incluye su dirección IP para identificar quien está haciendo la oferta. (puede haber más de una oferta si más de un servidor DHCP recibe el mensaje broadcast del cliente). Debido a que el cliente aun no tiene una dirección valida, el servidor debe enviar la oferta en un mensaje broadcast para que el cliente pueda recibirla.

3.- **El cliente envía un mensaje "*DHCP Request*"**: cuando está satisfecho con la oferta , el cliente solicita formalmente hacer uso de la dirección ofertada. Se incluye un registro de la dirección ofertada para que solo el servidor que envió la oferta ponga a un lado la dirección IP solicitada.

4.-**El servidor DHCP responde con un mensaje "DHCP ACK"**: la dirección IP y los demás parámetros son asignados al cliente con una aprobación formal para su uso. El mansaje ACK es enviado como *unicast*.

Se debe notar que DHCP está diseñado para trabajar dentro de un dominio de broadcast. La mayoría de los mensajes en DHCP se intercambian como broadcast. Basado en esto, el servidor DHCP necesitaría estar ubicado en el mismo dominio de broadcast que los clientes. En este escenario, se tendría que tener un servidor DHCP dedicado conectado a la red y ubicado en la misma VLAN que el cliente. Se puede configurar un *switch* multicapa para operar como un servidor DHCP si se tiene configurado direccionamiento de capa 3 en una interface de un *switch* o en una SVI donde haya clientes alojados.

## CONFIGURACION DE SERVIDOR DHCP IPv4

Primero se configura una dirección de capa 3 en una interface del *switch* para que el *switch* pueda participar en actividades relacionadas con capa 3. Entonces se puede configurar un servidor DHCP que funcione de manera nativa en el *switch*. Se puede configurar un *pool* de direcciones que sean ofrecidas por el servidor DHCP así como direcciones que estén reservadas o que se vayan a asignar de manera estática. En todos los casos, las direcciones del servidor DHCP deben estar correlacionadas con la subred que se configura en la interface del Router. El *switch* entonces interceptará los paquetes broadcast de un cliente dentro de una VLAN. Se usa la siguiente secuencia de comandos para configurar un servidor DHCP:

Switch (config)# ***ip dhcp excluded-address*** start-ip end-ip
Switch (config)# ***ip dhcp pool*** pool-name
Switch (config-dhcp)#***network*** ip-address subnet-mask
Switch (config-dhcp)#***default-router*** ip-address
Switch (config-dhcp)#***lease*** {infinite | {days [hours [minutes]]}}
Switch (config-dhcp)#***exit***

Se usa el comando ***show ip dhcp binding*** para monitorear las direcciones asignadas por DHCP.

PRACTICA FINAL

Nombre del participante_____RPE_____

Con la siguiente topología y la siguiente información de direccionamiento, construir una red con ruteo de VLANs usando EIGRP y logar comunicación entre todas las subestaciones.

NOTA.- ninguna vlan debe pasar de una subestación a otra excepto la vlan 961, 962 (UTRs) y 806 (Gestión).

Crear una ruta (de capa 2) desde todas las subestaciones hacia el área de control central a través de la VLAN 961 y 962 para que las UTRs puedan alcanzar a la UTM y viceversa.

Crear un servidor DHCP en el *switch* de SE tula y en SE Nochistongo. Cada uno con su pool correspondiente. Colocar un AP en TULA y en NOC y probar con un host *wireless* la conectividad por DHCP.

Colocar una laptop ADMIN en el switch TULA en la vlan 50 y probar conectividad hacia SE RAT.

Configurar el swich ACC con la información de las vlans 961, 962 y 970 abajo mostrada y probar comunicación desde la UTM hacia las UTRs.

Cada subestación debe tener un puerto configurado como acceso a la VLAN de las UTRs excepto JIP.
**Todas las UTRs deben estar dirigidas hacia SW UTRs.**

TABLA DE SWITCHES

| DISPOSITIVO | VLAN LAN | VLAN DHCP | IP VLAN GESTION | VLAN UTRs |
|---|---|---|---|---|
| SW L2/L3 **TUL** | 50 - 10.25.181.0/26 | 10- 10.25.176.0/24 | 806.→ 10.25.191.238 | 961 |
| SW L2/L3 **NOC** | 6- 10.25.178.190/26 | 7-10.25.178.222/24 | 806.→ 10.25.191.226 | 961 |
| SW L2/L3 **RAT** | 50- 10.25.184.62/26 | 49-10.25.184.94/24 | 806.→ 10.25.191.233 | 961 |
| SW L2/L3 **HRC** | 20-10.25.187.62/26 | 21-10.25.187.126/24 | 806.→ 10.25.191.232 | 961 |
| SW L2/L3 **JIP** | 8- 10.25.191.126/26 | VLAN DE TULA | 806.→ 10.25.191.225 | 961 |
| SW UTRs | N/A | N/A | 806.→ 10.25.191.227 | 961,962 |

TABLA DE UTRs Y UTM

| SUBESTACION | IP | MASCARA | GATEWAY |
|---|---|---|---|
| TULA A | 172.12.17.23 | 255.255.255.240 | 172.12.17.30 |
| NOC | 172.12.17.36 | 255.255.255.240 | 172.12.17.46 |
| RAT | 172.12.17.34 | 255.255.255.240 | 172.12.17.46 |
| HRC | 172.12.17.38 | 255.255.255.240 | 172.12.17.46 |
| MAESTRA | 172.12.19.23 | 255.255.255.240 | 172.12.19.30 |

IFORMACION DE DIRECCIONAMIENTO PARA SWITCH ACC

| VLAN | IP | NOMBRE |
|---|---|---|
| 961 | 172.12.17.46 | UTRs SUBESTACIONES remotas |
| 962 | 172.12.17.30 | UTRs TULA |
| 970 | 172.12.19.30 | MAESTRA |

Topología: