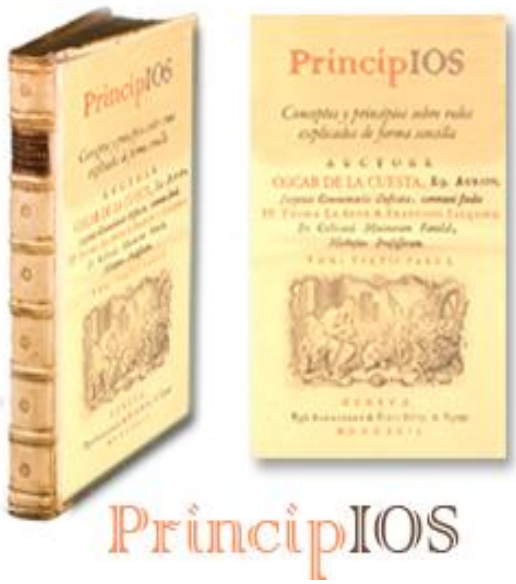


CONCEPTOS Y PRINCIPIOS SOBRE REDES EXPLICADOS DE FORMA SENCILLA

REV 1.0 beta

Gran parte de la información de este libro blanco la he buscado por internet de webs donde es lítica la copia y/o distribución de contenido (ver anexo final).

He realizado diversas tareas para crear esta guía:

Adaptado, aclarado, corregido, ampliado, dado formato y organizado numerosa información al archivo **PDF** para **una correcta visualización en cualquier dispositivo, además de las ventajas que ofrece este formato respecto a portabilidad y tamaño**. El documento PDF es de **carácter gratuito** para beneficiar a los usuarios que forman parte de la comunidad. A su vez permite al autor repasar los conocimientos adquiridos y servir de fuente para consultas específicas.

Deseo que os sea al menos de interés.

Observación:

La idea del logo, proviene del libro de Isaac Newton conocido como Principia, pero haciendo alusión a los comandos IOS empleados en los routers y conmutadores CISCO.

Este documento ha sido protegido con la herramienta gratuita MyPDFTreasure que he desarrollado y puedes descargar de la web:
http://www.palentino.es/blog/proteger-un-pdf-de-forma-facil-gratuita-firma-digital-y-permisos/

Tema I, Nociones y conceptos teóricos sencillos.

1 – Redes

2- TCP-IP

3 - Redes locales

4- Enrutamiento IP

5- Frame Relay

6- C.D.P.

7- N.T.P.

8- N.A.T

9- DHCP

Tema I

1 .Redes

1.1. Elementos básicos de una red

Para comenzar, en una red es necesario la existencia de un emisor, un receptor y un medio por el cual se comunican. Tanto emisor y receptor han de conocer el mismo lenguaje y utilizar el mismo sistema para que el diálogo entre ellos sea posible.

El emisor y el receptor pueden ser cualquier aparato que pueda transmitir datos y/o interpretar datos que reciban, mientras que el medio puede ser físico (cable de cobre, ...) o inalámbrico (ondas electromagnéticas, ...).

El lenguaje será un protocolo diseñado según el fin que se le vaya a dar y que tiene como objetivo que ambos dispositivos se entiendan.

1.2 El Modelo OSI y el modelo TCP/IP

El modelo OSI se basa en un conjunto de capas donde cada capa se comunica con la superior e inferior y tiene una función concreta, éste modelo dio paso al modelo TCP/IP que tiene mucho en común pero se suele explicar el modelo OSI ya que es más general.

Dentro del modelo **OSI hay 7 capas**, para tener una noción de redes con las tres capas inferiores bastaría para entenderlo pero aquí se incluirá la 4ª capa.

Las tres capas superiores se usan en aplicaciones de redes como VPN, algún protocolo de enrutamiento, servicios de red como SMTP (correo electrónico), HTTP (páginas web), ...

Las capas del modelo OSI y su correlación con TCP/IP son:

<table>
<tr><th>OSI</th><th>TCP/IP</th></tr>
<tr><td>Aplicación</td><td rowspan="3">Aplicación</td></tr>
<tr><td>Presentación</td></tr>
<tr><td>Sesión</td></tr>
<tr><td>Transporte</td><td>Transporte</td></tr>
<tr><td>Red</td><td>Internet</td></tr>
<tr><td>Enlace de datos</td><td rowspan="2">Red</td></tr>
<tr><td>Física</td></tr>
</table>

Cada capa tiene su unidad de datos que viene a ser el nombre que recibe el conjunto de datos a transmitir según va pasando de capa en capa, a esto se le llama PDU (Unidad de Datos del Protocolo) y en el modelo OSI, las PDU de las 4 capas inferiores son:

Transporte	Segmento
Red	Paquete
Enlace de datos	Trama
Física	Bits

Los bits serían los impulsos eléctricos o los ópticos dependiendo de si el medio es eléctrico o de fibra óptica. La capa física se encarga de ofrecer acceso al medio.

Cada tarjeta de red Ethernet (**NIC** - Network Interface Card) tiene una dirección física establecida de fábrica que consta de **6 bytes (48 bits)**, normalmente en las NIC y en los sistemas operativos, está representada en formato hexadecimal y cada byte separado por un guion "-" (por ejemplo: 01-02-03-AB-CD-EF) y en el caso del Cisco IOS, está representada por tres grupos de cuatro caracteres en formato hexadecimal separados por un punto (por ejemplo: 0102.03AB.CDEF).

Esta dirección **MAC** es única para cada NIC y consta de 3 bytes que identifican al **fabricante** y otros **tres bytes que identifican a la tarjeta**. La capa de "Enlace de datos" se encarga de establecer el protocolo de interconexión de los dispositivos. En la trama se encuentran los datos de la capa de red a los que la capa de enlace de datos añade una cabecera con la dirección MAC del dispositivo de origen y destino así como una cola con un control de errores y un delimitador.

La capa 3 se encarga de seleccionar **la mejor ruta desde el origen hasta el destino**. Aquí es donde los protocolos de enrutamiento (RIP, OSPF; EIGRP, IS-IS, ODR) toman decisiones basadas en las direcciones IP. En esta capa se añade una cabecera a la información que llega desde la capa de Transporte. En esta cabecera se incluye la dirección IP del dispositivo de origen y de destino.

La capa 4 capa de Transporte determina **si la información se enviará de forma fiable o no**, en esta capa se añade una cabecera a la PDU que viene de la capa de Sesión con información del puerto de origen y de destino y el protocolo de transporte mediante el que se enviará.

Si se usa **TCP** significa que la transmisión será fiable ya que mediante los números de secuencia se verifica que todos los datos lleguen al destino y si no llegan se usan las **ventanas TCP** para retransmitir la información de nuevo, si se usa **UDP** significa que aunque los datos se vayan a transmitir, **no se asegura que vayan a llegar al destino ya que no lleva ningún control de flujo** por lo que envía los datos y "espera" que vayan a llegar bien, pero si no llegan "tampoco pasa nada".

1.3 Direcciones IP (IPV4)

Una dirección **IP** (teniendo en cuenta **IPv4** pero no IPv6) consta de 4 bytes, cada byte tiene 8 bits, y un bit no es más que un 1 o un 0. Normalmente las direcciones IP se expresan en dotación decimal separando cada byte con un punto (por ejemplo 192.168.1.23). Una dirección IP identifica un dispositivo (PC, router, etc) en una red. Las direcciones IP asignables a host se pueden clasificar en **tres clases** (A, B, C), también se pueden clasificar en **públicas, privadas y reservadas**.

La dirección IP tiene dos partes, una parte que identifica a la red (o subred) y una parte que identifica al dispositivo. Por esto, la dirección IP sin más no tiene ningún sentido ya que necesita algún modo de indicar cuál es la porción de red y cuál la de host (de dispositivo), de esto se encarga la **máscara de red**.

La máscara de red consta también de 4 bytes separados por puntos. Si una dirección IP se pasa a binario y su máscara de red también y se sitúan correlacionando bit con bit, la parte de **la dirección IP que se corresponde con 1 en la máscara, es la parte de red, mientras que la parte que se corresponde con los 0 es la parte de host**. La parte de red es la parte que no puede cambiar en toda la red, mientras que dentro de la misma red, cada dispositivo tendrá una combinación distinta en la parte de host.

Por ejemplo, tomando como ejemplo la dirección IP 192.168.1.23 y su máscara de red 255.255.255.0:

```
Primero se pasa la IP a binario:

192 --> 11000000

168 --> 10101000

001 --> 00000001

023 --> 00010111

Ahora pasemos la máscara:

255 --> 11111111

255 --> 11111111

255 --> 11111111

000 --> 00000000

Se colocan una encima de la otra bit a bit:

11000000.10101000.00000001.00010111

11111111.11111111.11111111.00000000
```

Como se ve, la parte de red está compuesta totalmente de 1 y la de host de 0, en el caso de este ejemplo, hay 24 bits a 1 y 8 a 0, esto en la nomenclatura **CIDR** (del inglés **Classless Inter-Domain Routing**) se indicaría **/24**, donde 24 es la cantidad de bits a 1 en la máscara de red.

Viendo esto, será más sencillo entender la clasificación de una IP de host según su clase:

- Clase A:

  En binario, el primer byte comienza por 0. Tiene los **8 primeros bits** de la **máscara de red a 1** (255.x.y.z). El rango es de 1 a 126. **127.x.y.z** está reservado por los sistemas operativos como dirección de **loopback** o **localhost**.

  Ejemplo:

  ```
  115.2.56.3
  01110011.00000010.00111000.00000011

  255.0.0.0
  11111111.00000000.00000000.00000000
  ```

  En este caso se podría representar como 115.2.56.3**/8**

- Clase B:

  En binario, el primer byte comienza por 10. Tiene los 16 primeros bits de la máscara de red a 1 (255.255.y.z). El rango es de 128 a 192. 169.254.y.z está reservado como una dirección asignada de forma automática.

  Ejemplo:

  ```
  161.33.26.7
  10100001.00100001.00011010.00000111

  255.255.0.0
  11111111.11111111.0000000.00000000
  ```

  En este caso se podría representar como 161.33.26.7**/16**

- Clase C:

  En binario, el primer byte comienza por 110. Tiene los 24 primeros bits de la máscara de red a 1 (255.255.255.z). El rango es de 192 a 223.

  Ejemplo:

  ```
  213.97.49.120
  11010101.01100001.00110001.01111000

  255.255.255.0
  11111111.11111111.11111111.00000000
  ```

  En este caso se podría representar como 213.97.49.120**/24**

Aparte de las direcciones IP reservadas, las RFC1597 y RFC1918 (Request For Comments) especifican que cada clase tiene un rango de IP de uso privado, así los rangos privados de IP son:

- Clase A: 10.0.0.0/8 (Bloque CIDR 10/8).
- Clase B: comprende de 172.16.0.0/16 a 172.31.255.255/16 (Bloque CIDR: 172.16/12).
- Clase C: comprende de 192.168.0.0/24 a 192.168.255.255 (Bloque CIDR: 192.168/16).

También existe la clase D y E.

Así como las clases A, B y C son de **Unicast** (unidifusión - un origen a un destino), las direcciones de clase **D** son de **Multicast** (multidifusión - **un origen a más de un destino**) y tienen por máscara 255.0.0.0 (Bloque CIDR: 224/8), por ejemplo, servicios como el Imagenio de Telefónica se sirven de direcciones multicast para retransmitir cada canal por una dirección IP de clase D.

Las direcciones de clase E son para otros fines. En el rango de clase E se encuentra la dirección 255.255.255.255 que de no ser bloqueada por los routers, designaría a todos los hosts que hay en internet, por lo que de nuevo, si no fuese porque los routers bloquean los **broadcasts**, haciendo un ping a la dirección 255.255.255.255 responderían todos los hosts que hay en internet!!!

Bueno, en éste último párrafo he nombrado la palabra broadcast, que significa difusión a todos, es decir, para hacerlo sencillo con un ejemplo, es como una emisora de radio o televisión, que dentro del radio donde abarca la señal todo el mundo recibe la información.

Todo rango de direcciones IP, sea clase A, B o C tiene dos direcciones que no pueden ser usadas jamás:

- **IP de red: todos los bit de host tienen valor 0.**
- **IP de broadcast: todos los bits de host tienen valor 1.**

La IP de red tiene el objetivo de identificar a la red, por ejemplo de cara a los protocolos de enrutamiento, etc.

La IP de broadcast identifica a todos los hosts, es decir, cualquier dispositivo con una IP dentro de un mismo rango responde a su IP y a la IP de broadcast. La utilidad ping que sirve para comprobar la conectividad IP (capa 3) en una red, en el sistema operativo Windows no siempre funciona, pero en el sistema operativo de Cisco (Cisco IOS - Internet Operative System), se puede apreciar este hecho:

R2511RJ#**ping 192.168.1.255** (vemos como se hace ping a la ip de broadcast sobre iOS)

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.255, timeout is 2 seconds:

Reply to request 0 from 192.168.1.10, 4 ms
Reply to request 0 from 192.168.1.254, 4 ms
Reply to request 0 from 192.168.1.35, 4 ms
Reply to request 1 from 192.168.1.10, 1 ms
Reply to request 1 from 192.168.1.254, 4 ms
Reply to request 1 from 192.168.1.35, 4 ms
Reply to request 2 from 192.168.1.10, 1 ms
Reply to request 2 from 192.168.1.254, 4 ms
Reply to request 2 from 192.168.1.35, 4 ms
Reply to request 3 from 192.168.1.10, 4 ms
```

```
Reply to request 3 from 192.168.1.254, 8 ms
Reply to request 3 from 192.168.1.35, 4 ms
Reply to request 4 from 192.168.1.10, 1 ms
Reply to request 4 from 192.168.1.254, 4 ms
Reply to request 4 from 192.168.1.35, 4 ms
```

Cisco IOS envía 5 paquetes (PDU capa 3) a la IP 192.168.1.255 y le responden 3 IP diferentes 5 veces seguidas. Esto es porque cada dispositivo de la red responde a cada paquete que le envía el dispositivo de origen cuando se especifica la IP de broadcast.

Excluyendo la dirección de red y la de broadcast queda que según las clases tenemos disponibles los siguientes host:

- Clase A: (2^24)-2 = 16.777.214 hosts
- Clase B: (2^16)-2 = 65.524 hosts
- Clase C: (2^8)-2 = 254 hosts

Dada una IP y una máscara de red, se puede identificar la dirección de red mediante la operación lógica AND. La puerta lógica AND de dos entradas tiene la siguiente tabla de verdad:

Tabla de la verdad

A	B	Salida
0	0	0
0	1	0
1	0	0
1	1	1

Como se ve, lo que hace es un producto lógico, que no es una multiplicación ya que por ejemplo:

En binario 10*10 = 100 (4 en decimal)

En binario 10 AND 10 = 10 (2 en decimal)

Esto es porque la operación lógica AND lo que hace es un producto bit a bit, es decir:

```
2 AND 4:

2 --> 010
 &     ***
4 --> 100
- AND ---
0     000
```

De la misma manera, para saber la dirección IP de red a partir de la dirección IP de un host y su máscara de red, siguiendo el ejemplo de la dirección IP 192.168.1.23/24:

Primero se pasa la IP a binario:

192 --> 11000000

168 --> 10101000

001 --> 00000001

023 --> 00010111

255 --> 11111111

255 --> 11111111

255 --> 11111111

000 --> 00000000

Se colocan una dirección encima de la otra bit a bit y se hace la operación lógica AND:

11000000.10101000.00000001.00010111

******** ******** ******** ********

AND 11111111.11111111.11111111.00000000

11000000.10101000.00000001.00000000

Que pasado a decimal queda:

11000000 = 192 |

10101000 = 168 | == 192.168.1.0/24

00000001 = 001 |

00000000 = 000 |

Rango valido de direcciones IP asignables a host para 8 bits:

2^8 = 256-2 = 254.

192.168.1.1 a 192.168.1.254

Mascara 255.255.255.0

Esto es útil cuando parece que todo está correcto pero hay algún dispositivo en la red que no se comunica con el resto de la red y no se sabe porque, se busca cual es la dirección de red y el rango y se verifica que la IP esté dentro del rango de la subred. En el caso de máscaras de red por defecto no tiene mucho sentido buscar la IP de red pero cuando se hacen subredes o VLSM donde cada red tiene una máscara distinta, entonces sí que pueden darse con frecuencia confusiones en los rangos de IP.

Y qué son subredes y VLSM? Una subred es una red donde parte de los bits asignados a hosts pasan a ser de red, por ejemplo una dirección IP clase C con la máscara por defecto: 192.168.1.0/24. Se quiere dividir en 6 redes para los 6 departamentos de una empresa.

Para crear 6 redes necesitamos 3 bits (2^3=8), y ahora toca "robar" los bits de host para hacer la subred y modificar la máscara:

```
255.255.255.0 = 8 bits de host

0 = 00000000 en binario

Necesitamos 3 bits para subredes

11100000 = 224 en decimal

255.255.255.224 (CIDR: /27)
```

Vemos que la nueva máscara de red será 255.255.255.224, ahora veamos las subredes que hay y el rango asignable de cada subred:

```
Cuarto Byte: 00000000 = 0

IP de red: 192.168.1.0/27

De la IP: 192.168.1.1

A la IP: 192.168.1.30

Broadcast: 192.168.1.31



Cuarto Byte: 00100000 = 32

IP de red: 192.168.1.32/27

De la IP: 192.168.1.33

A la IP: 192.168.1.62

Broadcast: 192.168.1.63



Cuarto Byte: 01000000 = 64

IP de red: 192.168.1.64/27

De la IP: 192.168.1.65
```

A la IP: 192.168.1.94

Broadcast: 192.168.1.95

Cuarto Byte: 01100000 = 96

IP de red: 192.168.1.96/27

De la IP: 192.168.1.97

A la IP: 192.168.1.126

Broadcast: 192.168.1.127

Cuarto Byte: 10000000 = 128

IP de red: 192.168.1.128/27

De la IP: 192.168.1.129

A la IP: 192.168.1.158

Broadcast: 192.168.1.159

Cuarto Byte: 10100000 = 160

IP de red: 192.168.1.160/27

De la IP: 192.168.1.161

A la IP: 192.168.1.190

Broadcast: 192.168.1.191

Ya tenemos las 6 subredes, aún se podrían hacer dos subredes más, pero como solo se buscan 6 pues se deja aquí.

En este caso se han usado los bits del último byte, pero por ejemplo, si se necesitan 20 subredes y 1000 host en cada subred, necesitaremos:

2^5 = 32 --> 5 bits para subredes.

(2^10)-2 = 1022 --> 10 bits para host.

Se necesitan 5 bits para subredes y 10 bits para host lo que en total son 15 bits, si una dirección IP tiene 32 bits, necesitaremos una IP de red de clase A o B para poder usar 15 bits a nuestro antojo. El hecho es que como mucho hay 24 bits para "jugar" y dividirlos como queramos, si los puntos que separan cada byte despistan o incomodan a la hora de calcular subredes, siempre se pueden escribir todos los bits de host disponibles a 0 y a partir de ahí poner los puntos, en este caso disparatado en el que necesitamos 15 bits se podría hacer de la siguiente forma.

Está claro que necesitamos 15 bits, por lo que usaremos una dirección IP de clase B, para darle un toque de realismo a este ejemplo tomaremos el primer bloque privado que consta en la RFC1918 y haremos subredes a partir del mismo:

IP de clase B inicial: 172.16.0.0/16

Bits de subred: 5

Bits de host: 10

Tenemos 16 bits en la máscara disponibles para nosotros:

255.255.0.0 = 11111111.11111111.00000000.00000000

5 bits para subredes:

11111111.11111111.11111100.00000000

Dirección IP de red inicial:

172.16.0.0 = 10101100.00010000.00000000.00000000

Bits disponibles para redes en el tercer byte: 6

Bits disponibles para host: 10

Inicialmente la parte de la IP para host tiene todos los bits a 0.

Para las IP de red, comenzamos escribimos los cinco ceros que necesitamos para las redes:

Subred 1: 00000

Incrementamos el bit de menos peso (el de la derecha):

Subred 2: 00001

Subred 3: 00010

Subred 4: 00011

Subred 5: 00100

Subred 6: 00101

Subred 7: 00110

Subred 8: 00111

Subred 9: 01000

Subred 10: 01001

Subred 11: 01010

Subred 12: 01011

Subred 13: 01100

Subred 14: 10000

Subred 15: 10001

Subred 16: 10010

Subred 17: 10011

Subred 18: 10100

Subred 19: 10101

Subred 20: 10110

Pasos a hacer ahora:

1. A cada subred se le colocan diez 0 a la derecha que serán los bits de host.

2. Hemos de llenar 16 bits por lo que delante del número de cada subred ponemos un 0.

3. Hacemos dos grupos de 8 dígitos y los separamos por un punto.

4. Pasamos las cifras a decimal.

5. Colocamos al principio el prefijo de red 172.16.

Muestra:

Subred 1: 0 + 00000 + 0000000000 -> 00000000.00000000 = 00.0 => 172.16.0.0

Subred 2: 0 + 00001 + 0000000000 -> 00000100.00000000 = 04.0 => 172.16.4.0

```
Subred 3:  0 + 00010 + 0000000000 -> 00001000.00000000 = 08.0 => 172.16.8.0

Subred 4:  0 + 00011 + 0000000000 -> 00001100.00000000 = 12.0 => 172.16.12.0

Subred 5:  0 + 00100 + 0000000000 -> 00010000.00000000 = 16.0 => 172.16.16.0

Subred 6:  0 + 00101 + 0000000000 -> 00010100.00000000 = 20.0 => 172.16.20.0

Subred 7:  0 + 00110 + 0000000000 -> 00011000.00000000 = 24.0 => 172.16.24.0

Subred 8:  0 + 00111 + 0000000000 -> 00011100.00000000 = 28.0 => 172.16.28.0

Subred 9:  0 + 01000 + 0000000000 -> 00100000.00000000 = 32.0 => 172.16.32.0

Subred 10: 0 + 01001 + 0000000000 -> 00100100.00000000 = 36.0 => 172.16.36.0

Subred 11: 0 + 01010 + 0000000000 -> 00101000.00000000 = 40.0 => 172.16.40.0

Subred 12: 0 + 01011 + 0000000000 -> 00101100.00000000 = 44.0 => 172.16.44.0

Subred 13: 0 + 01100 + 0000000000 -> 00110000.00000000 = 48.0 => 172.16.48.0

Subred 14: 0 + 10000 + 0000000000 -> 00110100.00000000 = 52.0 => 172.16.52.0

Subred 15: 0 + 10001 + 0000000000 -> 00111000.00000000 = 56.0 => 172.16.56.0

Subred 16: 0 + 10010 + 0000000000 -> 00111100.00000000 = 60.0 => 172.16.60.0

Subred 17: 0 + 10011 + 0000000000 -> 01000000.00000000 = 64.0 => 172.16.64.0

Subred 18: 0 + 10100 + 0000000000 -> 01000100.00000000 = 68.0 => 172.16.68.0

Subred 19: 0 + 10101 + 0000000000 -> 01001000.00000000 = 72.0 => 172.16.72.0

Subred 20: 0 + 10110 + 0000000000 -> 01001100.00000000 = 76.0 => 172.16.76.0
```

Ya tenemos las IP de cada una de las redes con capacidad para 1000 host cada una, ahora solo haría falta asignar las IP a redes físicas y configurar los host con las IP dentro de cada rango. Aquí sí que se puede prestar a confusión la asignación de IP host por lo que si hay una red donde un host no tiene conectividad se puede hacer la operación AND de IP y máscara y así ver si la IP de ese host está dentro del rango.

Como se ve, hay un truquillo que es que una vez has determinado la cantidad de bits que necesitas para hacer las subredes, miras el bit de menor peso (el de más a la derecha) de la parte de red (en este caso los 6 primeros bits del tercer byte), calculas su valor en decimal, y cada subred en decimal se incrementa el valor de ése bit:

5 bits de subred + 3 bit para hacer los 8 del tercer byte:

0 + 00001 + 00 --> El 1 tiene valor 4 al pasarlo a decimal y así vemos que las subredes incrementan de 4 en 4 el tercer byte:

Subred 1: 00000000.00000000 = 00.0 => 172.16.0.0

Subred 2: 00000100.00000000 = 04.0 => 172.16.4.0

Subred 3: 00001000.00000000 = 08.0 => 172.16.8.0

Subred 4: 00001100.00000000 = 12.0 => 172.16.12.0

...

Y ahora que el tema de subredes ya está por la mano, se va a por el **VLSM** que es hacer **subredes dentro de subredes**!

Vamos allá con un ejemplo ya que la forma para hacerlo es la misma que una subred normal:

Si son las 10 y te viene un jefazo que acaba de llegar hace 5 minutos a la empresa y te dice algo así como:

"Mira, acabo de llegar y el informático me ha dicho que faltan IP en la empresa, he llamado al RIPE (organismo que asigna las IP en Europa) y les he dicho que quería comprar todo un bloque de clase C y que cuanta mayor cantidad de 2 (el número de la suerte del jefazo!!!) tenga la IP mejor, y me han dado el siguiente bloque completo: 222.22.2.0/24. De este rango, necesito 120 IP para los servidores del datacenter, luego necesito 6 subredes para los enlaces WAN hacia cada oficina de los que 4 han de tener 4 IP asignables y los otros 2 solamente 2 IP asignables, además cada una de las 5 oficinas ha de tener una subred con 3 IP asignables y sobretodo ahorrando el máximo de IP para asignar en un futuro.

Un enlace WAN es una conexión entre diferentes localizaciones, de router a router, solamente hay dos host. Lo que se llama un enlace punto a punto. Las IP de más pueden ser usadas para NAT estático por ejemplo.

Primero vamos a ordenar los datos para poder organizarnos y comenzar a asignar bits y máscaras:

- 1 red con 120 host.
- 4 redes con 4 host.
- 5 redes con 3 host.
- 2 redes con 2 host.

Para 120 host necesitamos 7 bits (2^7 = 128 -2 = 126).

Para 4 host necesitamos 3 bits (2^3 = 8 - 2 = 6). Con 2 bits nos quedaríamos cortos 2^2 = 4 - 2 = 2. Como vemos para 3 host también necesitamos 3 bits, por lo que podemos unificar y decir que necesitamos 9 redes donde podamos asignar 4 host. Ya hemos hecho las cosas más sencillas!

Para las redes con 2 host necesitamos 2 bits (2^2=4-2=2).

Ahora las subredes, lo haré a mi manera, que comienzo de la subred mayor a la menor:

Nos piden:

1er paso: 1 subred de 7 bits.

2o paso: 9 subredes de 3 bits.

3er paso: 5 subredes de 2 bits.

Tenemos:

8 bits disponibles: 00000000

1er paso:

1 subred con 7 bits de host:

1 bit de subred + 7 bits de host.

222.22.2.<0 + 0000000> = 0

Tenemos dos subredes hechas:

222.22.2.0/25

222.22.2.128/25

Mascara:

255.255.255.<10000000> = 128

Ya tenemos una subred hecha. Hemos "utilizado" un bit de la porción de host para la IP de red y nos queda otra subred que no usamos para nada con la IP: 222.22.2.128/25

2o paso:

IP disponible:

222.22.2.128/25 -> 7 bits disponibles.

El cuarto byte lo tenemos así: 10000000

De estos 8 bits, el primero no podemos ponerlo a 0 ya que se solaparía con la primera subred y habría problemas.

Así pues, tenemos 7 bits para "jugar":

9 subredes con 3 bits para host.

9 subredes = 4 bits.

1 + 0000 + 000

1 + 0001 + 000

1 + 0010 + 000

1 + 0011 + 000

1 + 0100 + 000

1 + 0101 + 000

1 + 0110 + 000

1 + 0111 + 000

1 + 1000 + 000

(Todavía se pueden hacer 7 subredes más).

Esto es:

1 bit que se usa para la primera subred.

4 bit que se usan para las 9 subredes.

3 bit disponibles para asignar 4 host en cada subred.

Ahora toca calcular la máscara de subred, y esta vez uso los 5 bits de mayor peso para cada subred, así pues la máscara en el 4 byte tendrá 5 bits a 1:

255.255.255.<11111000> = 248

En total se usan 29 bits para la máscara ahora.

Y las IP para cada una de las subredes será:

222.22.2.128/29

222.22.2.136/29

222.22.2.144/29

222.22.2.152/29

222.22.2.160/29

222.22.2.168/29

222.22.2.176/29

222.22.2.184/29

222.22.2.192/29

Todas ellas con mascara 255.255.255.248

De nuevo, se puede ver que de subred a subred se incrementa en 8 porque el bit de la parte de red con menor peso está en la cuarta posición y en decimal equivale a 8.

Bueno, ya tenemos hecha casi toda la faena, ahora nos queda calcular las dos redes para las ADSL de su casa y la de casa de su hermano.

3er paso:

Ultima subred usada:

222.22.2.192/29

Esto significa que la siguiente red a usar va a ser la 222.22.2.200/29 pero como el jefe tiene alergia a los 0 y le gustan más los números pares, vamos a usar la última subred disponible con la máscara anterior.

Si seguimos contando, las subredes que todavía podemos hacer son:

222.22.2.200/29

222.22.2.208/29

222.22.2.216/29

222.22.2.224/29

222.22.2.232/29

222.22.2.240/29

222.22.2.248/29

En el cuarto byte ahora tenemos usados 5 bits para red, y nos piden: 2 subredes de 2 host.

2 subredes = 2^1 = 2 -> 1 bit para red.

2 host = 2^2 = 4 - 2 = 2 -> 2 bits para host.

Necesitamos 3 bits, hasta ahora hemos estado usando 5 bits de los 8 que tenemos por lo que podemos meter esta subred dentro de una del paso anterior.

Vamos a usar la red 222.22.2.248/29 para las dos subredes que nos quedan.

Cuarto byte actual: 11111 + 000

5 bits de red + 3 bits de host.

Subred 1: 11111 + 0 + 00

Subred 2: 11111 + 1 + 00

Esto es: 5 bits de red + 1 bit de subred + 2 bits para host.

Las subredes que hemos hecho ahora son:

222.22.2.248/30

222.22.2.252/30

Con mascara: 255.255.255.252

Direccionamiento IP actual:

Red para 120 host: 222.22.2.0/25

Mascara: 255.255.255.128

Redes para 4 host (y hasta 6):

222.22.2.128/29

222.22.2.136/29

222.22.2.144/29

222.22.2.152/29

222.22.2.160/29

222.22.2.168/29

222.22.2.176/29

222.22.2.184/29

222.22.2.192/29

Mascara: 255.255.255.248

Redes para 2 host: 222.22.2.248/29

222.22.2.252/29

Mascara: 255.255.255.252

Para uso futuro hay disponible:

222.22.2.200/29

222.22.2.208/29

222.22.2.216/29

222.22.2.224/29

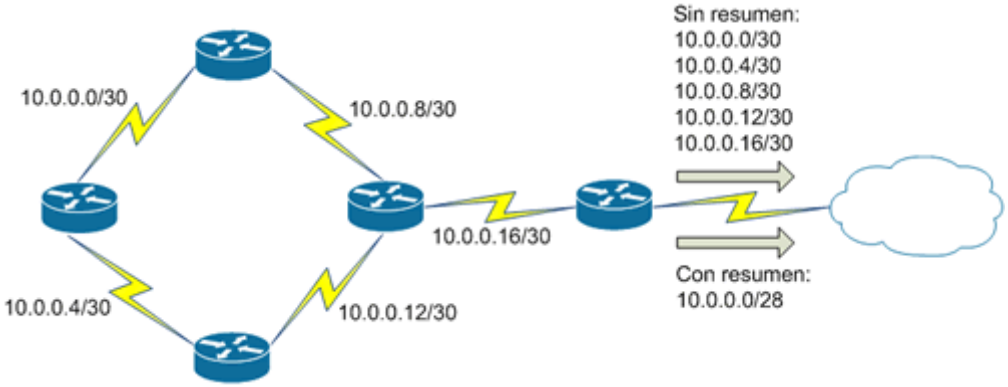
222.22.2.232/29

222.22.2.240/29

Es decir, para uso futuro nos quedan 6 redes con capacidad de 6 host cada una.

No hay más!

Lo contrario a las subredes son las supraredes o super-redes, también llamado resumen de ruta, algo muy útil cuando dos enrutadores intercambian las redes que tienen conectadas cada uno de ellos ya que así se publican menos redes, la tabla de enrutamiento de los routers tiene menos entradas, se rebaja la carga en la memoria del dispositivo con lo que se aumenta la velocidad y eficiencia a la hora de tomar decisiones de enrutamiento.



El calcular una supranet que englobe a un grupo de redes contiguas es tan sencillo como poner a 0 todos los bits de la parte de red de las subredes que vaya variando, es decir:

Tenemos las redes con las IP siguientes:

1. 200.70.16.0/24

2. 200.70.17.0/24

3. 200.70.18.0/23

4. 200.70.20.64/26

5. 200.70.20.128/26

6. 200.70.20.192/26

7. 200.70.30.128/25

Tal y como decía, una suprared tiene la misión de englobar a varias redes, para englobar a varias redes ha de tomarla parte de red común a todas y en el resto poner los bits a 0 como si fuesen de host.

En las redes se ve que el primer y segundo byte es común a todas, por lo que se mantendrá como esta, es decir, que la suprared comenzara por "200.70.", el cuarto byte varía entre ellas y además según la máscara es la parte de host, por lo que los bits estarán a 0 y la suprared la calcularemos a partir de los bits en el tercer byte, por lo que pasamos a binario el tercer byte de cada red:

1. 16 = 00010000

2. 17 = 00010001

3. 18 = 00010010

4. 20 = 00010100

5. 20 = 00010100

6. 20 = 00010100

7. 30 = 00011110

Se puede ver que de los 8 bits, los únicos que se mantienen constantes son los 4 primeros, por lo que los 4 últimos bits se pondrán a 0, quedando el tercer byte como "00010000" que en decimal es 16.

Por lo que la suprared que englobara a las redes será: 200.70.16.0

Y la máscara de red abarcara los bits de los dos primeros bytes y cuatro bits del tercer byte, que en total serán: 2*8 + 4 = 20 bits (255.255.240.0)

Quedando definitivamente:

IP de red: 200.070.016.0

Mascara: 255.255.240.0

Hasta aquí la parte del protocolo IP. Otras cosas como el DHCP y el NAT que están bastante relacionadas con esto lo pondré dentro de las configuraciones de routers Cisco ya que hay mucho que tocar de esto también.

Protocolos de enrutamiento también irá en otro apartado dentro de "nociones" para entender qué es el enrutamiento y en qué se basan los protocolos de enrutamiento para tomar decisiones y llenar la tabla de enrutamiento.

Hasta aquí la parte del protocolo IP. Otras cosas como el DHCP y el NAT que están bastante relacionadas con esto lo pondré dentro de las configuraciones de routers Cisco ya que hay mucho que tocar de esto también.

Protocolos de enrutamiento también irá en otro apartado dentro de "nociones" para entender qué es el enrutamiento y en qué se basan los protocolos de enrutamiento para tomar decisiones y llenar la tabla de enrutamiento.

1.4. Protocolos de transporte

Tanto TCP como UDP usan puertos para dar servicio a la capa de sesión (capa 5), haciendo un símil, cada puerto sería como un cajón de una cajonera, donde cada cajón está asociado a una aplicación, entonces los datos que van al puerto 21 son para subir archivos al servicio FTP, los datos que van al puerto 25 para el servidor de correo electrónico, los datos del puerto 80 para el servidor de páginas web. Los puertos son **bidireccionales**, es decir, se puede enviar y recibir datos por ellos si la aplicación asociada un puerto está diseñada para ello.

TCP:

El protocolo **TCP se dice que está orientado a conexión y es fiable**, esto significa que el dispositivo de origen y el de destino tienen una conexión establecida, están en comunicación y cuando en la transmisión se pierden datos por el camino el receptor se da cuenta, se lo dice al emisor y éste los reenvía.

Ante todo hay que explicar tres conceptos de este protocolo:

- Número de secuencia: se trata de un número único que identifica cada transmisión de datos entre origen y destino (y viceversa).
- Ventanas deslizantes: se envían X paquetes y en cuanto se recibe la confirmación (ACK) de que uno de los paquetes ha sido recibido, envía el siguiente, esto aumenta la eficiencia de la transmisión ya que en versiones anteriores se enviaba un paquete y se esperaba la confirmación antes de enviar otro paquete, lo que dejaba mucho tiempo la conexión vacía en cada sentido, posteriormente se establecían tamaños de ventana de X paquetes y hasta que no se recibían las confirmaciones de cada uno de los paquetes de la ventana, no se transmitía otra ventana. Lo que se consigue con la ventana deslizante es aprovechar al máximo el tiempo de uso de la línea ya que si el tamaño de ventana es de 10, tal y como va recibiendo la confirmación de un paquete va enviando uno más, por ejemplo, con un tamaño de 10, envía 10 paquetes, en cuanto recibe la confirmación del primer paquete envía el siguiente, cuando recibe confirmación del 2º paquete, envía otro y así sucesivamente, de forma que siempre hay 10 paquetes de los cuales está a la espera de recibir confirmación pero nunca se deja de transferir esperando las confirmaciones. El tamaño de la ventana viene dado por un campo de 16 bits en la cabecera del protocolo TCP.
- Escalado de ventana: es un campo en la cabecera TCP donde se especifica mediante 16 bits (aunque solo se usan 14) un múltiplo al campo del tamaño de la ventana, de esta forma se consigue aumentar el tamaño de la ventana al producto de: <tamaño de la ventana> * 2 ^ <bits escalado de la ventana>. El tamaño máximo de la ventana con el escalado máximo es de 1Gbit (1.073.741.824 bits), lo cual significa que el protocolo TCP, actualmente, permite que la ventana abarque hasta 1Gigabit.

Los paquetes que usa TCP son:

- SYN: Sirve para sincronizar el emisor y receptor, en otras palabras, es el aviso de que se quiere establecer una conexión.
- ACK: Es el acuse de recibo de que los datos del emisor han llegado.
- FIN: Indica fin de la conexión.

El establecimiento de la conexión se dice que es a tres fases ya que se envían 3 paquetes:

Emisor	- <SEQ=X><CTL=SYN> ->	Receptor
	* <- <SEQ=Y><ACK=X+1><CTL=SYN,ACK> -	
	- <SEQ=X+1><ACK=Y+1><CTL=ACK> -> *	
	- DATOS ->	

Con * se indica el momento en el cual la conexión queda establecida en cada extremo.

La transmisión de datos:

Emisor	- DATOS <sec#=100> ->	Receptor
	<- ACK <sec#=101> -	
	- DATOS <sec#=102> ->	
	<- ACK <sec#=103> -	
	...	
	- DATOS <sec#=n> ->	
	<- ACK <sec#=n+1> -	

Cuando el receptor no responde con un ACK y el número de secuencia adecuado, el emisor interpreta que se han perdido datos, reduce la ventana y los envía de nuevo hasta que recibe acuse de recibo del receptor con los números de secuencia adecuados a cada transmisión.

Existe otro paquete, el SACK, que es capaz de indicar concretamente qué bytes se han perdido para que así el emisor solo retransmita esos datos y aumentar la eficiencia de la transmisión.

El fin de la conexión consta de 4 fases:

Emisor	- FIN <sec#=x> ->	Receptor
	<- ACK <sec#=x+1> -	
	<- FIN <sec#=y> -	
	- ACK <sec#=y+1> -> *	

Con el asterisco "*" se indica donde se da por finalizada la conexión.

El protocolo TCP también hace uso de sistemas para verificar la integridad de los datos y temporizadores para calcular retrasos en la red y así ajustar el funcionamiento para un rendimiento óptimo. Sus aplicaciones son cualquiera donde se quiera garantizar la entrega de los datos.

Protocolo UDP:

El protocolo UDP, a diferencia de TCP/IP, no se asegura de que los paquetes vayan a llegar a su destino. Si hay que enviar datos de un sitio a otro, UDP crea un paquete con los siguientes campos:

- Puerto de origen.
- Puerto de destino.
- Longitud del mensaje (paquete).
- Checksum (comprobación de integridad).
- Datos.

Esto significa que UDP coge los datos de la capa de sesión, le añade el puerto de origen, el puerto de destino, cuenta la longitud del paquete, hace unas operaciones para que el destino pueda verificar si los datos están en buen estado o se han corrompido, y lo envía. El puerto de origen y el checksum son opcionales.

UDP se usa más que nada para aplicaciones que no requieran que todos los datos lleguen al destino para dar un servicio correcto, es decir, que si se pierden datos no pasa nada. Por ejemplo un streaming de video, si se pierden unos cuantos frames de un video, puedes seguir viendo el video, ahora bien si se pierde una cantidad X de datos de un fichero, el fichero estará corrupto y lo más probable es que sea ilegible.

Por otro lado, si se usase TCP para una transmisión de video y una cantidad X de datos se hubiese perdido y el tamaño de ventana fuese muy elevado, en el reenvío podría llegar frames anteriores a los últimos que se han mostrado, esto ocasionaría que viendo el video hubiese un momento que no viésemos nada, luego viésemos una imagen y luego los frames anteriores a la imagen que acabamos de ver, lo que haría la visualización bastante incómoda.

Hay protocolos que tienen ambos protocolos de transporte, por ejemplo FTP y TFTP.

- FTP utiliza TCP como protocolo de transporte.
- TFTP utiliza UDP como protocolo de transporte.

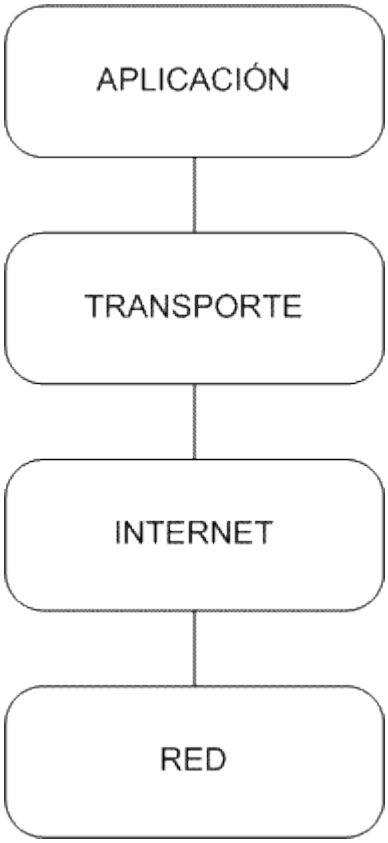
FTP: File Transfer Protocol. Puerto 21 TCP.

TFTP: Trivial File Transfer Protocol. Puerto 69 UDP.

2 .TCP-IP

2.1. Elementos básicos de una red

El conjunto de protocolos TCP/IP, del inglés Transfer Control Protocol / Internet Protocol, es un esquema alternativo al modelo OSI que de la misma manera que el modelo OSI, divide las funciones de la red en capas, en este caso en 4 capas:



Cada una de las capas tiene una función bien definida. La capa de red (también llamada de acceso a la red) define los aspectos físicos de la conexión y como se transmitirán los datos por el medio. La capa de Internet establece el camino y gestiona las conexiones. La capa de transporte se encarga de entregar los datos en el destino. Y finalmente la capa de aplicación se encarga de adaptar los datos a la aplicación a la cual están destinados.

Según la capa los datos que se manejan se denominan con un nombre concreto:

Aplicación	Mensaje
Transporte	Paquete
Internet	Datagrama
Red	Trama

En una comunicación entre dispositivos, cada capa se comunica con la misma capa del dispositivo de destino.

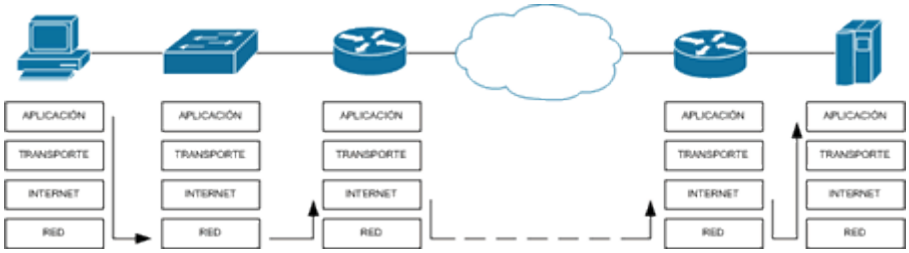
Dependiendo de los dispositivos, estos trabajan con la información de una capa u otra:

- Capa RED: conmutadores (switch), puentes (bridge).
- Capa INTERNET: enrutadores (routers).
- Capa TRANSPORTE: cortafuegos (firewalls), redes privadas virtuales (VPN), ...
- Capa APLICACIÓN: ordenadores personales (PC), servidores, algunos cortafuegos, ...

Algunos protocolos de cada capa son:

- RED: CDP, ARP, ...
- INTERNET: IP, ICMP, ...
- TRANSPORTE: TCP, UDP.
- APLICACIÓN: HTTP, SMTP, TFTP, FTP, Telnet, ...

Por ejemplo, de manera simplificada, si un usuario de un PC quiere visitar una página web, los datos entre el PC de origen y el servidor de destino seguirían un camino similar a este:



2.2 Las direcciones IP

El protocolo IP, utiliza direcciones de 32 bits en formato de cuatro bytes en formato decimal separados por puntos. Por ejemplo: 192.168.21.1

Estas direcciones, llamadas direcciones IP, van acompañadas de otro conjunto de 4 bytes separados por puntos llamado máscara de red. La máscara de red permite saber a un dispositivo cual es la parte de la dirección IP que se refiere a la dirección de la red y cual designa a un host determinado a través de una operación lógica llamada operación Y (del inglés AND).

La tabla de verdad de la puerta lógica del mismo nombre es como se muestra a continuación:

Puerta lógica

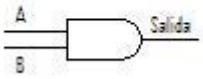


Tabla de la verdad

A	B	Salida
0	0	0
0	1	0
1	0	0
1	1	1

En conclusión, se puede decir, que el resultado solamente será 1 cuando ambas entradas tengan valor 1.

La operación AND no es un producto normal, sino que es un producto bit a bit:

```
2 = 010      | 3 = 011
4 = 100      | 5 = 101
             |
2 * 4 = 8    | 3 * 5 = 15
2 Y 4 = 0    | 3 Y 5 = 1
             |
 2 = 010     | 3 = 011
 Y   ***     | Y   ***
 4 = 100     | 5 = 101
--   ---     | -   ---
 0 = 000     | 1 = 001
```

En el caso de las direcciones IP, la parte de la dirección que se refiere a la red tiene los bits de la máscara con valor 1 mientras que la parte que designa a los dispositivos de la red, tiene los bits a 0. Teniendo una dirección IP y una máscara, se puede determinar que los bytes en los que la máscara tiene valor 255 son parte de la dirección IP de red, los bytes que tienen valor 0 pertenecen a la parte de la IP que se asignará a los dispositivos:

```
192.168.021.3            |      172.16.32.5
255.255.255.0            |      255.255.0.0
                         |
Parte de red:            |      Parte de red:
192.168.21               |      172.16
Parte de host:           |      Parte de host:
3                        |      32.5
                         |
Direccion IP de red:     |      Direccion IP de red:
  192.168.021.3          |        172.016.32.5
Y 255.255.255.0          |      Y 255.255.00.0
---------------          |      --------------
  192.168.021.0          |        172.016.00.0
```

Dependiendo de cómo comienza el primer byte en binario, las direcciones IP se clasifican en cinco clases:

- Clase A: el primer byte comienza por "0xxxxxxx". El rango es de 1.0.0.0 a 127.255.255.255. Máscara 255.0.0.0.
- Clase B: el primer byte comienza por "10xxxxxx". El rango es de 128.0.0.0 a 191.255.255.255. Máscara 255.255.0.0.
- Clase C: el primer byte comienza por "110xxxxx". El rango es de 192.0.0.0 a 223.255.255.255. Máscara 255.255.255.0.
- Clase D: el primer byte comienza por "1110xxxx". El rango es de 224.0.0.0 a 239.255.255.255. Uso para multicast.
- Clase E: el primer byte comienza por "11110xxx". El rango es de 240.0.0.0 a 254.255.255.255. Para uso futuro.

Las direcciones IP de clase A, B y C son asignables a dispositivos, las direcciones de clase D se usan para difusión, la dirección 255.255.255.255 hace referencia a todos los dispositivos de una red.

Dada una IP y una máscara, hay un rango de IP que pueden ser asignables excepto dos:

- Dirección de red: tiene todos los bits de host a 0. Indica cual es la red en la que se encuentra un host.
- Dirección de difusión: tiene todos los bits de host a 1. Abarca todos los host que hay en una red.

Al hacer ping a una dirección de difusión, todos los host de la red deberían responder:

```
R2511RJ#ping 192.168.1.255

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.255, timeout is 2 seconds:

Reply to request 0 from 192.168.1.250, 4 ms
Reply to request 0 from 192.168.1.10, 8 ms
Reply to request 1 from 192.168.1.10, 4 ms
Reply to request 1 from 192.168.1.250, 4 ms
Reply to request 2 from 192.168.1.10, 4 ms
Reply to request 2 from 192.168.1.250, 4 ms
Reply to request 3 from 192.168.1.10, 4 ms
Reply to request 3 from 192.168.1.250, 4 ms
Reply to request 4 from 192.168.1.10, 4 ms
Reply to request 4 from 192.168.1.250, 4 ms
```

Por esto, si tenemos un dispositivo con dirección IP "172.16.45.6" y máscara "255.255.0.0", el rango de IP asignable de la red será:

Primero buscamos la dirección IP de la red en la que se encuentra el dispositivo:

172 Y 255 = 172

016 Y 255 = 016

045 Y 000 = 000

006 Y 000 = 000

Por lo que tenemos que la dirección IP de la red es: 172.16.0.0

Al tener la máscara de red dos bytes enteros a 1, tenemos otros dos bytes para host.

2 bytes = 16 bits.

Ahora para saber la cantidad de IP que se encuentran en 2 bytes, hemos de elevar 2 a la potencia 16:

2^16 = 65536 direcciones IP.

Como de estas 65536 direcciones IP se ha de restar la dirección de red y la de difusión, nos quedan 65534 direcciones IP asignables:

IP de RED: 172.16.0.0

Direcciones asignables:

172.16.0.1 ... 172.16.255.254

Dirección de DIFUSION: 172.16.255.255

2.3 Subredes

El hacer subredes, es simplemente coger una red de clase A, B o C y ampliar la máscara restando bits a la parte de host. Si una empresa tiene la IP de red "192.168.3.0" con la máscara "255.255.255.0" y necesita asignar direcciones IP en cinco redes pero solamente tiene asignada la dirección de red especificada, el proceso es el siguiente:

IP de red: 192.168.003.0

Mascara: 255.255.255.0

Necesitamos asignar IP en cinco redes:

2^2 = 4 --> no se llega.

2^3 = 8 --> 3 bits para hacer subredes.

La máscara de red indica que el único byte que tenemos disponible para usar los bits de host es el cuarto, por lo que de los 8 bits del cuarto byte cogemos tres para hacer las cinco subredes:

000|00000 = 000

001|00000 = 032

010|00000 = 064

011|00000 = 096

100|00000 = 128

Todavía se pueden asignar tres subredes más que de momento quedaran sin usar:

101|00000 = 160

110|00000 = 192

111|00000 = 224

En el cuarto byte, los bits para cada dirección serán:

IP de red: XXX|00000

IP asignable: XXX|00001 ... XXX|11110

IP de difusión: XXX|11111

Ahora las direcciones IP de red, asignables y de difusión para cada red serán las siguientes:

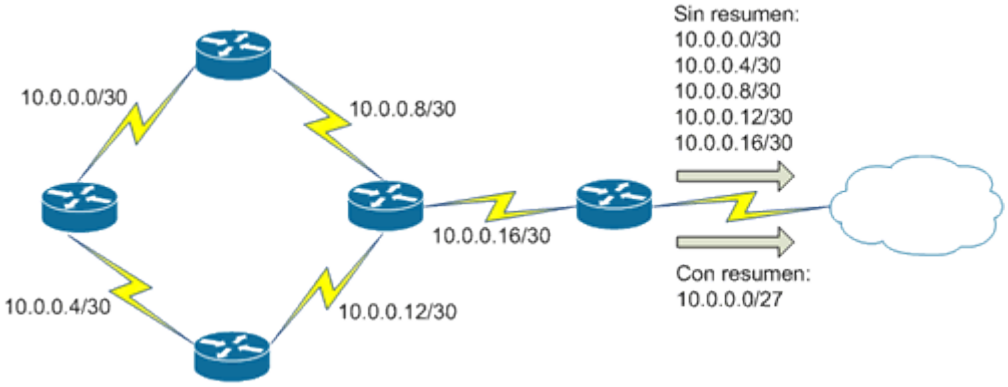
Primera red:
IP red: 192.168.003.000
Asignable: 192.168.003.001 ... 192.168.003.030
IP difusion: 192.168.003.031
Primera segunda:
IP red: 192.168.003.032
Asignable: 192.168.003.033 ... 192.168.003.062
IP difusion: 192.168.003.063
Primera tercera:
IP red: 192.168.003.064
Asignable: 192.168.003.065 ... 192.168.003.094
IP difusion: 192.168.003.095
Primera cuarta:
IP red: 192.168.003.096
Asignable: 192.168.003.097 ... 192.168.003.126
IP difusion: 192.168.003.127

El hacer subredes es útil para no desperdiciar direcciones IP, por ejemplo, si se quieren conectar dos oficinas con una línea dedicada y se tiene un rango de IP público con máscara 255.255.255.0 con la que se pueden asignar 254 direcciones IP, si no se pudiese dividir en subredes con menos IP, con una sola conexión entre dos sedes donde se necesitan dos direcciones IP (una para el router de cada sede y como mucho alguna más para hacer NAT estático y/o dinámico), se estarían desperdiciando 252 direcciones. Actualmente se asignan rangos de direcciones IP más pequeños y se usa VLSM que se trata de crear subredes dentro de subredes para así aprovechar al máximo los rangos de IP según las necesidades.

El VLSM se tocará después de las supraredes en el apartado "VLSM y CIDR".

2.4 Superredes y VLSM – CIDR

Las supraredes (también llamado resumen de ruta) son direcciones IP de red que abarcan varias redes, se utilizan básicamente para reducir la cantidad de direcciones IP que una red privada publica a Internet, de esta manera se reduce la carga de la tabla de enrutamiento. La superred tiene una máscara que :



Tanto la dirección IP como la máscara de la superred tiene todos los bits de host, y los asignados a subredes con valor 0. El método para obtener la dirección IP y máscara de una superred, es el siguiente:

Mascara /30 = 255.255.255.252

Direcciones IP:

10.0.0.00

10.0.0.04

10.0.0.08

10.0.0.12

10.0.0.16

El byte que cambia es el cuarto, se pasa a binario:

00 = 00000000

04 = 00000100

08 = 00001000

12 = 00001100

16 = 00010000

Centrándonos en el cuarto byte, los tres primeros bits y se mantendrán como están, los bits del cuarto al sexto cambian y tendrán valor 0 y los dos últimos bits pertenecen a host, como la suprared es una dirección IP de red, estos bits estarán a 0, por lo que quedara:

000|000|00 = 0

10.0.0.0

El cuarto byte de la máscara en un principio era:

252 = 11111100

Siendo la parte que estaba a 0 la parte asignable a los dispositivos, ahora se ha de reducir y los bits que cambiaban en las direcciones IP han de ponerse a 0 en la máscara de red:

111|000|00 = 224

Por lo que el resultado es que nos queda una red que engloba a todas las redes con la IP y la máscara siguiente:

010.000.000.000

255.255.255.224 (/27)

VLSM viene del inglés "Very Large Subnet Mask" es un método de hacer subredes dentro de subredes, de forma que el desaprovechamiento de direcciones IP es mucho más reducido que haciendo subredes.

Por ejemplo, si una empresa decide usar la dirección IP 192.168.9.0 con máscara 255.255.255.0 y ha de asignar IP en los siguientes departamentos:

- Centro de datos: 40 dispositivos.
- Oficina: 15 dispositivos.
- Tres líneas dedicadas punto a punto (de router a router) y una de ellas con dos direcciones IP para hacer NAT estático.
- Telefonía IP: 23 dispositivos.

Primero de todo hay que ver los bits que necesitaremos para cada subred teniendo en cuenta que hay que restar del rango la dirección de red y de broadcast que no se usan:

Teniendo en cuenta que cada dispositivo tiene una IP, lo que nos piden es:

Centro de datos: 40 IP + 2 Red y dif.

Oficina: 15 IP + 2 Red y dif.

Dos lineas dedicadas: 2 IP cada una + 2 Red y dif.

Una linea dedicada: 4 IP + 2 Red y dif.

Telefonia IP: 23 IP + 2 Red y dif.

Calculamos los bits que se necesitan para asignar a host en cada red:

1 bit -> 2^1 = 2 host

2 bit -> 2^2 = 4 host

3 bit -> 2^3 = 8 host

4 bit -> 2^4 = 16 host

5 bit -> 2^5 = 32 host

6 bit -> 2^6 = 64 host

7 bit -> 2^7 = 128 host

8 bit -> 2^8 = 256 host

Centro de datos: 6 bits

Oficina: 5 bits

Dos lineas dedicadas: 2 bits

Una linea dedicada con NAT: 3 bits

Telefonia IP: 5 bits

La direccion IP y mascara con la que partimos es:

192.168.009.0

255.255.255.0

Por lo que tenemos los últimos 8 bits de host para cambiar según las necesidades que tengamos:

Primera subred: centro de datos -> 6 bits de host.

00|000000 --> 000 ==> 192.168.009.000

11|000000 --> 192 ==> 255.255.255.192

IP red: 192.168.009.000

Asignable: 192.168.009.001 ... 192.168.009.062

IP dif.: 192.168.009.063

Mascara: 255.255.255.192

Segunda subred: oficina -> 5 bits de host.

010|00000 --> 064 ==> 192.168.009.064

111|00000 --> 224 ==> 255.255.255.224

IP red: 192.168.009.064

Asignable: 192.168.009.065 ... 192.168.009.094

IP dif.: 192.168.009.095

Mascara: 255.255.255.224

Tercera subred: telefonia IP -> 5 bits de host.

011|00000 --> 096 ==> 192.168.009.096

111|00000 --> 224 ==> 255.255.255.224

IP red: 192.168.009.096

Asignable: 192.168.009.097 ... 192.168.009.126

IP dif.: 192.168.009.127

Mascara: 255.255.255.224

Cuarta subred: linea dedicada con NAT -> 3 bits de host.

10000|000 --> 128 ==> 192.168.009.128

11111|000 --> 248 ==> 255.255.255.248

IP red: 192.168.009.128

Asignable: 192.168.009.129 ... 192.168.009.134

IP dif.: 192.168.009.135

Mascara: 255.255.255.248

Quinta subred: linea dedicada -> 2 bits de host.

100010|00 --> 136 ==> 192.168.009.136

111111|00 --> 252 ==> 255.255.255.252

IP red: 192.168.009.136

Asignable: 192.168.009.137 ... 192.168.009.138

IP dif.: 192.168.009.139

Mascara: 255.255.255.252

Sexta subred: linea dedicada -> 2 bits de host.

100011|00 --> 140 ==> 192.168.009.140

111111|00 --> 252 ==> 255.255.255.252

IP red: 192.168.009.140

Asignable: 192.168.009.141 ... 192.168.009.142

IP dif.: 192.168.009.143

Mascara: 255.255.255.252

CIDR, del inglés "Classless Inter-Domain Routing", es un sistema para indicar el prefijo de red y la cantidad de bits con valor 1 que tiene la máscara de red (siempre comenzando por la izquierda), de esta forma se puede ver de forma jerárquica como están asignadas las direcciones IP.

En el ejemplo anterior, la IP inicial era "192.168.9.0" y la máscara "255.255.255.0", con CIDR esto podría indicarse como "192.168.9.0/24". Y las subredes del ejemplo anterior podrían indicarse de manera similar:

IP inicial: 192.168.009.0

Mascara: 255.255.255.0

255 = 11111111

255 = 11111111

255 = 11111111

000 = 00000000

Total 24 bits a 1 --> /24

CIDR: 192.168.9.0/24

1ª subred: 192.168.009.0

Mascara: 255.255.255.192

255 = 11111111

255 = 11111111

255 = 11111111

192 = 11000000

Total 26 bits a 1 --> /26

CIDR: 192.168.9.0/26

2ª subred: 192.168.009.64

3ª subred: 192.168.009.96

Mascara: 255.255.255.224

255 = 11111111

255 = 11111111

255 = 11111111

224 = 11100000

Total 27 bits a 1 --> /27

CIDR: 192.168.9.64/27

CIDR: 192.168.9.96/27

4ª subred: 192.168.009.128

Mascara: 255.255.255.248

255 = 11111111

255 = 11111111

255 = 11111111

248 = 11111000

Total 29 bits a 1 --> /29

CIDR: 192.168.9.128/29

5ª subred: 192.168.009.136

6ª subred: 192.168.009.140

Mascara: 255.255.255.252

255 = 11111111

255 = 11111111

255 = 11111111

252 = 11111100

Total 30 bits a 1 --> /30

CIDR: 192.168.9.136/30

CIDR: 192.168.9.140/30

Ahora supongamos que en el centro de datos quieren dividir el rango de IP asignado a ellos en dos rangos de 30 direcciones IP cada uno para dos salas distintas:

IP asignada al centro de datos: 192.168.9.0/26

En el cuarto byte se cambia la máscara para dividir la IP de red inicial en dos subredes iguales:

Red Sala A: 000|00000 = 000

Red Sala B: 001|00000 = 032

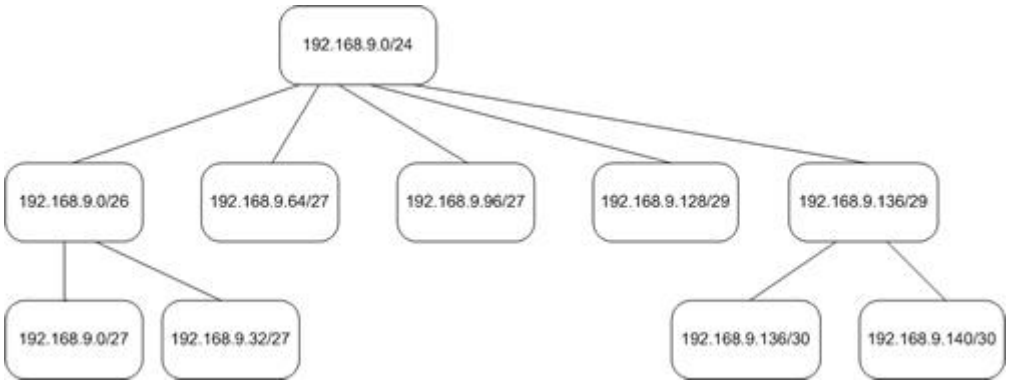
Mascara: 111|00000 = 224

IP Sala A: 192.168.009.000/27

IP Sala B: 192.168.009.032/27

Mascara: 255.255.255.224

La distribución jerárquica del bloque CIDR inicial podría representarse así:



2.5 El protocolo TCP

El protocolo TCP es un protocolo de la capa de transporte del modelo TCP/IP. Se le llama que es un protocolo fiable ya que tiene mecanismos para asegurar la entrega de los datos. Da servicio mediante los puertos, que es como una cajonera donde cada cajón sería un puerto y se asocia a una aplicación. El campo para puertos es de 16 bits, esto significa que hay 65535 puertos disponibles, de los cuales, los puertos del 0 al 1023 son puertos asignados por la IANA, los siguientes puertos del 1024 hasta el 49151 son puertos asignados por otras empresas y del 49151 al 65535 son puertos dinámicos, asignados por las aplicaciones pero que no tienen sentido fuera de una conexión en concreto, es decir, una misma aplicación puede usar dos puertos entre el 49151 y el 65535 en dos momentos distintos. Cada uno de los puertos es bidireccional, es decir, que si la aplicación que lo usa está diseñada para ello, puede enviar y recibir simultáneamente por el mismo puerto.

TCP añade una cabecera al paquete de la capa de aplicación con la siguiente información:

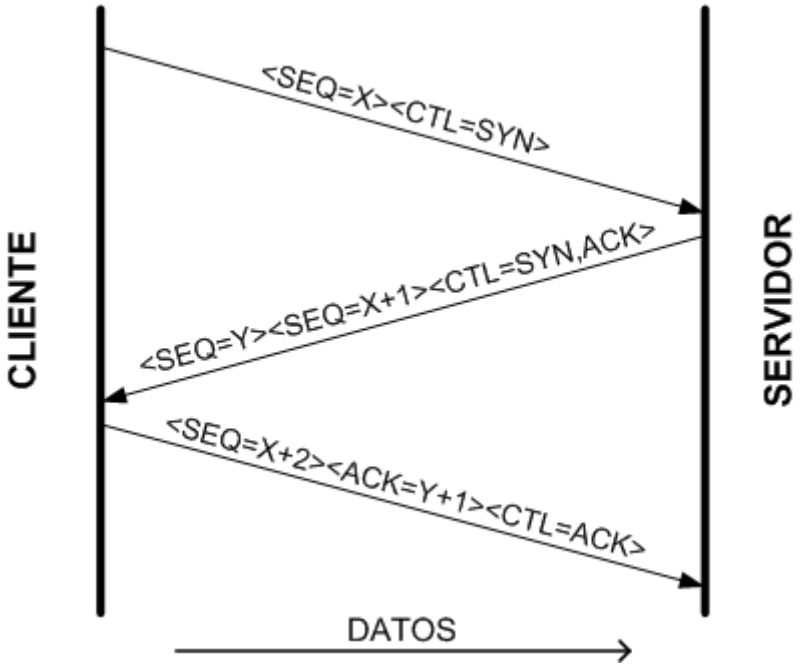
- Puerto de origen: número de puerto desde el cual el dispositivo abre la conexión. 16 bits.
- Puerto de destino: número de puerto al cual el dispositivo de destino quiere conectarse. 16 bits.
- Número de secuencia: número único que identifica el paquete transmitido. 32 bits.
- Número de acuse de recibo: número de secuencia del paquete del cual se va a confirmar la recepción. 32 bits.
- Longitud cabecera TCP: establece la longitud de la cabecera TCP para saber en qué punto de los datos recibidos acaba la cabecera y comienzan los datos. 4 bits.
- Flags: son 8 bits cada uno de los cuales tiene un significado.
- Ventana: indica la longitud de la ventana, que vendría a ser la cantidad de bytes que el receptor espera recibir antes de tener que enviar la primera confirmación de recepción. 16 bits.
- Checksum: es una suma para verificar que no haya errores ni en la cabecera ni en los datos. 16 bits.
- Puntero urgente: indica donde se encuentran datos urgentes. 16 bits.

- Opciones: sirve para especificar opciones adicionales que no están especificadas en la cabecera TCP.

TCP tiene tres tipos de paquetes:

- SYN: inicia el establecimiento de conexión.
- ACK: acuse de recibo. Confirma la recepción de un paquete.
- FIN: termina la conexión.

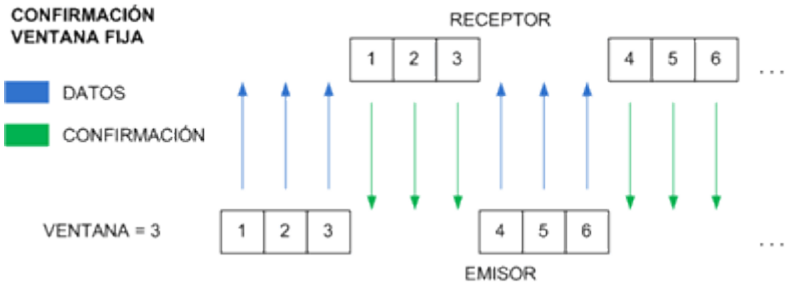
El establecimiento de conexión se efectúa en tres fases:



Donde "SEQ" es el número de secuencia y "CTL" es el tipo de paquete.

El protocolo TCP es un protocolo que entrega los datos de manera fiable, esto lo hace esperando confirmación de la correcta entrega de los datos que ha enviado. Para la confirmación de entrega existen varios métodos:

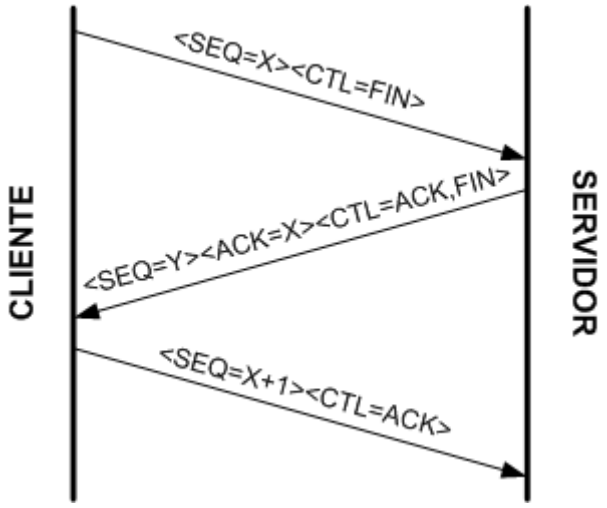
- Confirmación uno a uno: con este sistema, cada paquete entregado se confirma antes de enviar el siguiente.
- Ventana fija: se establece una cantidad de paquetes que van a ser enviados sin esperar confirmación y en cuanto se hayan recibido las confirmaciones de todos los paquetes se volverá a enviar la cantidad de paquetes fijada en la ventana y de nuevo se esperará a recibir la confirmación de recepción de todos los paquetes antes de enviar de nuevo. Se podría decir que se trata de enviar X cantidad de paquetes de forma seguida y hasta que no has recibido la confirmación de todos ellos no se envía de nuevo X cantidad de paquetes.
- Ventana deslizante: se establece una cantidad de paquetes que van a ser enviados sin esperar confirmación, pero en este caso, en cuanto se recibe la confirmación del primer paquete enviado, se envía un nuevo paquete, y así sucesivamente. De este modo siempre hay 10 paquetes en espera de confirmación pero mientras haya algo que transmitir no deja de haber tráfico en ambos sentidos. Este es el método que usa TCP.



El tamaño de ventana se establece durante el establecimiento de conexión y puede tener un valor hasta 65535. Existe la opción 3 de TCP la cual establece un múltiplo que aumenta el tamaño de ventana a 30 bits lo que equivale a tener una ventana de 1 Gigabit.

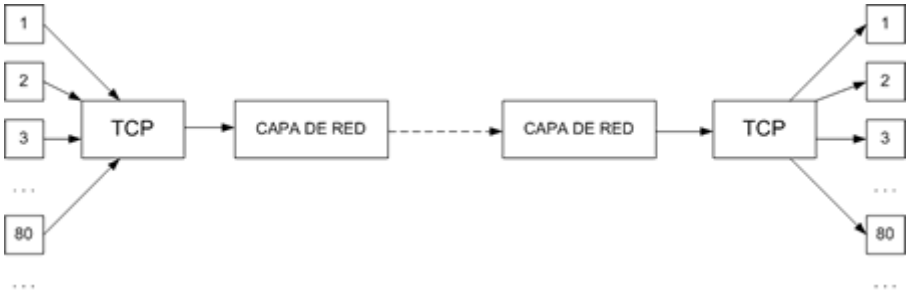
En caso de que un paquete no sea recibido, el emisor lo detecta ya que no recibe el paquete ACK con el número de secuencia correspondiente y reenvía el paquete de nuevo. En caso de que el paquete de confirmación sea el que se ha perdido y no llega al emisor, éste lo interpreta como que no se ha recibido el paquete y lo reenvía, pero el destinatario del paquete al detectar que el número de secuencia de dicho paquete ya ha sido confirmado lo descarta, de esta forma se evitan paquetes repetidos.

Para finalizar una conexión, TCP usa los paquetes "FIN":



TCP usa 16 bits para los puertos de origen y destino. Los puertos podrían describirse como un número identificativo de servicio, una aplicación puede usar más de un puerto por IP, pero un mismo puerto no puede ser utilizado por más de una aplicación en una misma dirección IP. Esto significa que si un dispositivo como puede ser un servidor, tiene asignada solamente una dirección IP, cada uno de los puertos podrá ser utilizado por una aplicación solamente, en cambio si un servidor tiene más de una IP configurada en alguna/s de sus interfaces, podrá abrir cada uno de los 65536 puertos en cada IP y ser usados por diferentes aplicaciones.

La información que TCP recibe por un puerto, la envía a la capa de red añadiéndole la cabecera TCP y una cola con un campo de control de errores. La capa de red se encarga de hacer llegar los datos al destino y en el destino TCP envía los datos al puerto especificado en el campo "puerto de destino" de la cabecera:



Un zócalo es el conjunto que hacen tres elementos necesarios para establecer una conexión: protocolo, dirección IP y puerto. Por ejemplo: "http://192.168.1.1:80". Esto se usa básicamente para servicios de la capa de aplicación.

2.6 El protocolo UDP

El protocolo UDP, del inglés User Datagram Protocol, es un protocolo llamado de máximo esfuerzo, esto significa que no garantiza la entrega de los datos. Al igual que TCP, el protocolo UDP usa puertos para que las aplicaciones envíen datos.

Este protocolo añade una cabecera a los datos recibidos de la capa de aplicación con la siguiente información:

- Puerto origen: puerto del dispositivo que está transmitiendo. 16 bits.
- Puerto destino: puerto del dispositivo al cual se va a transmitir. 16 bits.
- Longitud de los datos: longitud de la cabecera y los datos encapsulados, el valor mínimo es 8. 16 bits.
- Checksum: para comprobación de errores. 16 bits.

De estos campos, el de longitud de los datos y checksum son opcionales, por lo que lo estrictamente obligatorio es especificar el puerto de origen y destino, que es lo mínimo necesario para establecer una conexión.

Con estos datos, UDP lo único que puede hacer es enviar datos, y con los datos recibidos verificar la integridad en el mejor de los casos.

El principal motivo para usar UDP es la poca carga que acarrea, la cabecera UDP consta de 8 bytes mientras que la cabecera TCP puede tener hasta 60 bytes.

3 .Redes Locales

Para la comunicación entre dos dispositivos, bien sea en una red con muchos dispositivos o en una conexión entre dos dispositivos, se necesitan tres datos básicos:

- Un medio de transmisión de los datos: actualmente hay transmisiones por conductores eléctricos, conductores ópticos y transmisiones de ondas electromagnéticas las cuales se transmiten por el espacio sin usar ningún medio físico. Este aspecto entraría dentro de la capa 1 del modelo OSI y TCP/IP.
- Un protocolo de comunicación común: todos los dispositivos han de transmitir los datos con el mismo lenguaje para que se puedan entender entre ellos, de la misma manera que sucede en las conversaciones entre personas. Esto pertenece a la capa 2 del modelo OSI y capa 1 de TCP/IP.
- Un identificador para cada dispositivo: cada dispositivo ha de ser identificable independientemente para que en caso de que se transmitan datos destinados a él sepa diferenciarlos de los destinados a cualquier otro dispositivo. También pertenece a la capa 2 del modelo OSI y capa 1 de TCP/IP.
- Un camino desde el emisor al receptor: al poder tratarse de dispositivos distantes físicamente y poder haber diferentes caminos para llegar desde el origen al destino, hay que disponer de algún sistema mediante el cual los datos del emisor vayan a llegar de forma fiable o no al receptor. Dependiendo del protocolo, esto puede darse en la capa 2 y la capa 3 del modelo OSI y en la capa 1 y capa 2 del modelo TCP/IP.

Los medios más utilizados actualmente son el cobre y las redes inalámbricas. También se usa la fibra óptica aunque más bien como enlaces troncales que soportan más carga pero no para los terminales usados por los usuarios finales.

El protocolo de comunicación más usado en redes locales es Ethernet, antiguamente se usaba también Token Ring, que fue un predecesor de Ethernet y hubo algún fabricante que creó interfaces de red para crear redes locales usando ATM (Marconi por ejemplo).

El identificador de dispositivo del protocolo Ethernet es en realidad el identificador de la interfaz de red, y se llama dirección MAC (del inglés *Medium Access Control* - Control de Acceso al Medio), también fue usada antiguamente en redes Token Ring y actualmente en redes Ethernet y Wifi.

La dirección MAC está formada por 6 bytes expresados en hexadecimal y separados por guiones, aunque cada fabricante de software o hardware lo expresa de una manera distinta (Cisco en tres grupos de dos octetos separados por un punto, Microsoft separa cada uno de los seis octetos con guiones, ...). Esta dirección está dividida primeramente en dos grupos, los tres primeros bytes se refieren a un código asignado al fabricante por el IEEE (organismo encargado de establecer estándares) y los tres últimos bytes se refieren al identificador de la interfaz.

Así, por ejemplo, la dirección MAC del PC que estoy usando es: D8-D3-85-79-47-64. Esto indica:

- D8-D3-85: Identificador del fabricante, en este caso Realtek.
- 79-47-64: Identificador del dispositivo, esta parte es única para cada dispositivo para cada identificador de fabricante.

Dentro del primer byte, los dos últimos bits también tienen un significado:

- Último bit:
  - Valor 0: dirección MAC de unicast. También llamada individual.
  - Valor 1: dirección MAC de multicast. También llamada de grupo.
- Penúltimo bit:
  - Valor 0: dirección MAC asignada por el fabricante. También llamada global.
  - Valor 1: dirección MAC asignada por el administrador. También llamada local.

Por este motivo, si en un dispositivo que permita cambiar la MAC se intenta poner una dirección MAC en la cual el último bit tenga valor 1 (multicast) el sistema tal vez no de error pero no la registrará en memoria. Por otro lado, si se especifica una MAC de forma manual y se pone el penúltimo bit a 0 (asignada por el fabricante) el sistema no dará error y la usará pese a ser asignada localmente. Hay que tener cuidado con cambiar MAC para evitar direcciones repetidas lo que podría llevar a tener problemas en la red.

Más adelante se explicará la conveniencia de que cada interfaz tenga una MAC distinta y no que un dispositivo use la misma MAC en todas sus interfaces.

El sistema de entrega de datos, llamado LLC (de Logical Link Control) tiene dos modos:

- LLC tipo 1: es un método de entrega de máximo esfuero, lo que significa que no se garantiza que se vayan a entregar los datos.
- LLC tipo 2: es un método orientado a conexión, donde existe una confirmación de recepción. Pero este método no se usa en las redes locales, es el método usado por HDLC en enlaces punto a punto por ejemplo.

LLC establece un "puerto" de origen y uno de destino, aquí los puertos tienen el nombre de Puntos de Acceso a Servicio (Service Access Point), y tienen un campo de 8 bits, aunque de estos 8 bits, dos están destinados a especificar si el puerto está asignado de forma local o global y si los datos para ese puerto van destinados a uno o varios protocolos de nivel superior:

- Último bit:
  - Valor 0: individual.
  - Valor 1: grupo / conjunto de protocolos.
- Penúltimo bit:
  - Valor 0: global, asignado por un organismo oficial.
  - Valor 1: local, asignado en la LAN.

Por lo cual, restando estos dos bits quedan seis 6 bits disponibles para puertos lo que da un total de 2^6 = 64 puertos. Parecen muy pocos pero hay un comodín para que puedan usarse más puertos, este comodín es que junto con los tres bytes que un fabricante recibe para asignar las direcciones MAC, recibe dos bytes (16 bits) más que puede usar para identificar protocolos, esto se explica a continuación.

El puerto de origen recibe el nombre de SSAP (*Source Service Access Point*) y el puerto de destino recibe el nombre de DSAP (*Destination Service Access Point*) y normalmente en una comunicación SSAP y DSAP tienen el mismo valor. Bien pues, en caso de que tanto DSAP como SSAP tengan un valor hexadecimal de "AA" (en binario sería "10101010") significa que la cabecera incluye un campo que indica el número de protocolo. Este valor SSAP y DSAP de "AA" se le llama SNAP SAP (*SubNetwork Access Protocol Service Access Point* - Punto de acceso al Servicio Protocolo de Acceso a la Subred) y el campo de protocolo que espera encontrar tiene una longitud de dos bytes (16 bits), los dos bytes que se indicaban al final del párrafo anterior. El identificador de protocolo no es exclusivo de una organización, un mismo protocolo puede ser usado por dispositivos de varias organizaciones.

A continuación se pueden ver primero una captura de pantalla de un paquete de un protocolo estandarizado por el IEEE (*Spanning-Tree* - Se tratará a continuación) sin los campos SSAP y DSAP a "AA" y sin campo de identificador de protocolo, y en la segunda captura es un protocolo propietario de Cisco (CDP - Se trata en Nociones - CDP) por lo que tiene los campos SSAP y DSAP con valor "AA" y campo de identificador de protocolo (PID):

```
Logical-Link Control
  DSAP: Spanning Tree BPDU (0x42)
  IG Bit: Individual
  SSAP: Spanning Tree BPDU (0x42)
  CR Bit: Command
⊟ Control field: U, func=UI (0x03)
    000. 00.. = Command: Unnumbered Information (0x00)
    .... ..11 = Frame type: Unnumbered frame (0x03)
```

```
Logical-Link Control
  DSAP: SNAP (0xaa)
  IG Bit: Individual
  SSAP: SNAP (0xaa)
  CR Bit: Command
⊟ Control field: U, func=UI (0x03)
    000. 00.. = Command: Unnumbered Information (0x00)
    .... ..11 = Frame type: Unnumbered frame (0x03)
  Organization Code: Cisco (0x00000c)
  PID: CDP (0x2000)
```

Se puede ver que ambas capturas tienen especificado el bit de individual o grupo como individual (valor 0).

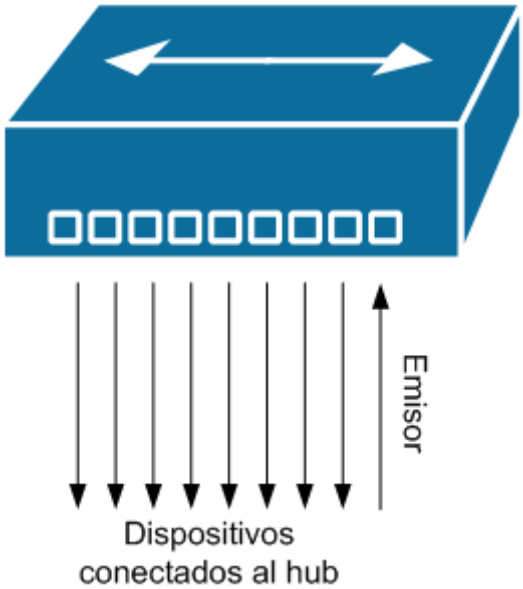
El campo de control, indica el tipo de paquete. En LLC tipo 1 que es el que se usa en redes locales, la longitud siempre será de 1 byte y puede tener los valores siguientes:

- UI - *Unnumbered Information*: indica que es un datagrama. Un datagrama es un paquete con la información mínima para poder ser encaminado desde el origen al destino.

- XID - *eXchange IDentification*: para intercambiar información de dispositivos. Hay dos tipos: "*command*" y "*response*":
    - Paquete "*command*": envía la identificación del emisor junto con los tipos LLC que soporta.
    - Paquete "*response*": es la respuesta al paquete "*command*" e indica los mismos datos pero del dispositivo receptor.
- TEST: sirve para comprobar que la comunicación entre un emisor y un receptor es posible. También tiene dos tipos de paquetes "*command*" y "*response*". El contenido son datos arbitrarios.

3.1 Concentradores

Los concentradores, más habitualmente conocidos por su nombre inglés de "*hub*", son dispositivos similares a los repetidores, trabajan en la capa física del modelo OSI (capa 1) y capa de red de TCP/IP (capa 1), los cuales lo único que hacen es reenviar la información que reciben, sin tomar ningún tipo de decisión más que discernir el puerto por el cual han recibido los datos y no reenviarlos por el mismo puerto:



El comportamiento de los concentradores y los repetidores es como si todos los dispositivos estuvieran conectados al mismo cable y recibieran todos los datos que envían todos los dispositivos, y cada interfaz de red recoge los datos destinados a ella según la dirección MAC tal y como está explicado en el apartado 1.

El problema de este sistema, es cuando dos estaciones transmiten simultáneamente, en ese caso, se da lo que se llama una colisión. Dada la naturaleza de ethernet como medio compartido, el hecho de que ocurran colisiones está contemplado y cuando ocurren hay mecanismos para detectarlas y ejecutar un método de reenvío para entregar los datos.

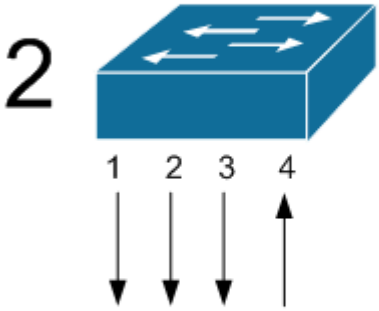
El método que utilizan para decidir si transmiten, detectar colisiones y reenviar los datos se llama CSMA/CD del inglés "*Carrier Sense Multiple Access with Collision Detection*" y existe otro método de transmisión llamado MACA "*Multiple Access with Collision Avoidance*" que viene a ser algo así como un sistema mediante el cual una estación, antes de transmitir "pide permiso" al resto de estaciones conectadas al medio o al concentrador, y cuando recibe autorización transmite los datos, de esta forma se evita que dos estaciones transmitan datos simultáneamente aunque puede darse una colisión en el momento de "pedir permiso", por este motivo cabe la posibilidad de que se implemente MACA para transmitir datos, pero para los datos de control seguir usando CSMA/CD.

Para solucionar el problema de las colisiones se inventaron los puentes y los conmutadores ("*switch*").

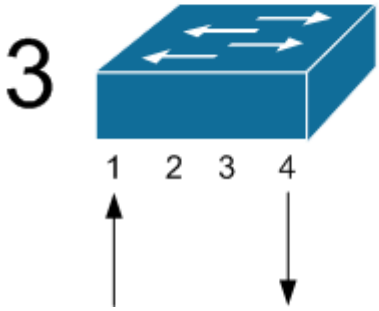
3.2 Conmutadores y puentes

Los puentes y conmutadores, más conocidos como "*bridge*" y "*switch*", son dispositivos que trabajan en la capa de Enlace de Datos del modelo OSI (capa 2) y en la capa de Red de TCP/IP (capa 1). Su funcionamiento en un primer momento es similar al de un concentrador con la diferencia de que aprende donde se encuentra cada dispositivo según su dirección física (dirección MAC) y en las sucesivas veces que ha de transmitir un paquete donde la cabecera indica que va destinado a una dirección que ya tiene registrada, envía el paquete directamente por el puerto correspondiente.El proceso que se sigue es el siguiente:

```
Switch#show mac-address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
Switch#
```




```
Switch#show mac-address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    18a9.0555.763e    DYNAMIC     Fa0/4
Total Mac Addresses for this criterion: 1
Switch#
```




```
Switch#show mac-address-table dynamic
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    000d.658a.c130    DYNAMIC     Gi0/1
   1    18a9.0555.763e    DYNAMIC     Fa0/4
Total Mac Addresses for this criterion: 2
Switch#
```

1. En el momento inicial, el conmutador no tiene ninguna dirección MAC registrada en la memoria.
2. Cuando la estación conectada al puerto número 4 envía datos, el conmutador guarda la dirección MAC de origen que aparece en la cabecera de la trama (PDU capa 2 modelo OSI) en la tabla interna que guarda en memoria, y envía los datos por todos los puertos excepto el puerto por el cual recibió la trama. Hasta aquí se comporta como un concentrador.
3. Cuando la estación a la cual van destinados los datos responde, el conmutador guarda la dirección MAC que figura como origen de la trama en la memoria y como la dirección MAC de destino ya la tiene guardada en memoria, envía el paquete directamente al puerto correspondiente y sucesivamente lo que vaya dirigido a cualquiera de las dos direcciones MAC que tiene registradas lo enviará directamente al puerto correspondiente, y en caso de que haya datos que vayan dirigidos a una dirección MAC que no tiene registrada, realizará el paso 2 para aprender tantas direcciones como vaya recibiendo, aunque en la práctica el límite de direcciones MAC que un switch para uso empresarial de tipo Cisco suele tener es de 2048 entradas.

El hecho de guardar la dirección física en memoria agiliza el tráfico de datos de manera notable dado que entre dos puertos se establece un canal dedicado donde solo los dispositivos conectados en cada puerto recibe los datos, evitando colisiones con otros dispositivos, y al evitar las colisiones el rendimiento aumenta de manera muy notable en momentos en los que se transmiten muchos datos, ya que aunque el ancho de banda no sea del 100% para todos los puertos cuando se transmite simultáneamente, sí que hay un flujo de datos constante -o casi constante- y esto se nota cuando se hace un cambio de concentrador a conmutador.

El principal problema con el hecho de guardar las direcciones MAC en memoria, es que si se da el caso de que una estación deja de estar disponible o si es cambiada físicamente de puerto (puede que el puerto se haya dañado, ...), cual es el momento para decidir que una estación deja de estar disponible en un puerto determinado? El método elegido fue establecer un temporizador, de forma que si una estación deja de transmitir por el puerto que tiene el conmutador en memoria, se borra la dirección MAC de la tabla y si se reciben datos de esa estación se guarda su dirección MAC en la tabla de nuevo y si otra estación envía datos destinados a la estación cuya MAC ha sido borrada, se realizaría el proceso del paso 2 del esquema anterior.

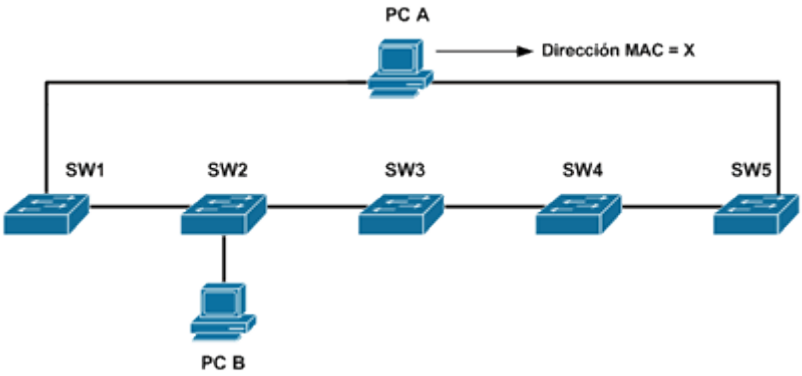
El siguiente reto pasa a ser el hecho de establecer el temporizador para borrar la dirección MAC de la memoria. Si se establece un periodo demasiado breve, habría muchos cambios en la memoria y se ejecutaría repetidamente el paso 2 del esquema provocando colisiones y disminuyendo el rendimiento. Por otro lado, un temporizador demasiado largo provocaría que en una red donde varios conmutadores se conectan entre ellos mediante otro conmutador central, en el momento en el que la estación cambiase de puerto, del conmutador A al conmutador B, el conmutador central podría estar enviando datos destinados a la estación en cuestión pensando que todavía se encuentra conectada al conmutador A, por lo que la pérdida de conectividad se alargaría innecesariamente. El temporizador en los conmutadores Cisco es de 3 minutos de forma predeterminada aunque puede ser cambiado.

La diferencia entre un puente y un conmutador, es que un puente tiene solamente dos puertos que enlazan dos redes mientras que un conmutador tiene varios puertos a los cuales se conectan estaciones u otros dispositivos (enrutadores, ...) aunque también puede interconectar varios conmutadores o conectar con un puente y así unir dos conmutadores, ... Lo importante es entender que un conmutador y un puente hacen la misma función a nivel lógico y hay que decir, que inicialmente, tanto los conmutadores como los puentes eran llamados puentes y el término conmutador fue acuñado por la industria, no por los diseñadores del sistema.

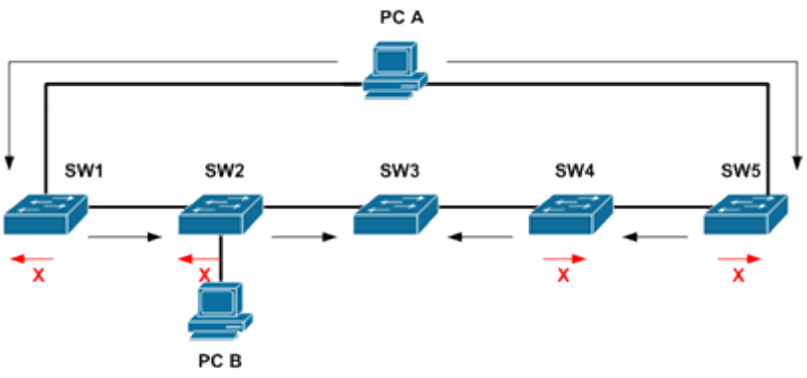
Problemas con las redes de puentes y conmutadores

Este punto está destinado a ver algunos posibles problemas que puede haber con las redes con puentes y el motivo de que las direcciones MAC se asignen por interfaz y no por estación, es decir, por qué una estación no transmite datos usando la misma dirección MAC por todas sus interfaces en lugar de transmitir con una MAC u otra dependiendo de la interfaz por la cual transmite, y por aquí comenzaré.

Si una estación tiene solamente una interfaz de red, no habría ningún problema en que se asignase una dirección MAC a la estación en lugar de a la tarjeta de red ya que el resultado sería el mismo, con un solo punto de acceso a la red no habría conflictos para llegar a la estación de destino. El problema viene cuando una estación tiene más de una interfaz y ambas se conectan a la misma red pero en diferentes conmutadores para tener redundancia y la dirección MAC fuese única para todas las interfaces de la estación:

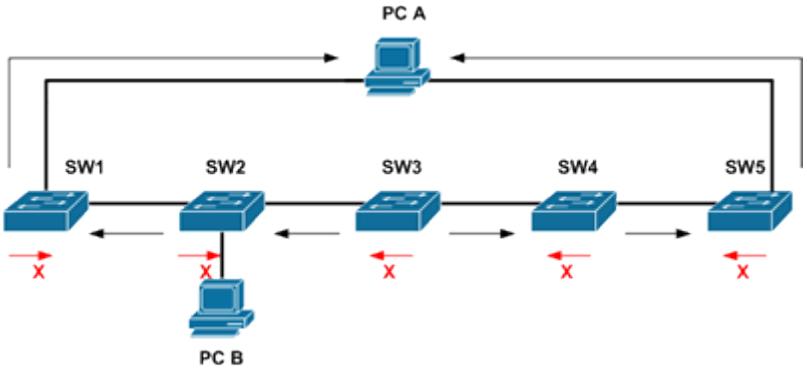


Teniendo en cuenta que la estación tiene una dirección MAC con un valor cualquiera X y que transmite simultáneamente un paquete al PC B por ambas interfaces hacia los conmutadores y que en los conmutadores no hay tráfico que aumente la latencia lo suficiente como para que el paquete enviado por un extremo (SW1) llegue al otro extremo antes que el paquete enviado por el otro extremo (SW5) haya cruzado un solo conmutador, pasaría lo siguiente:



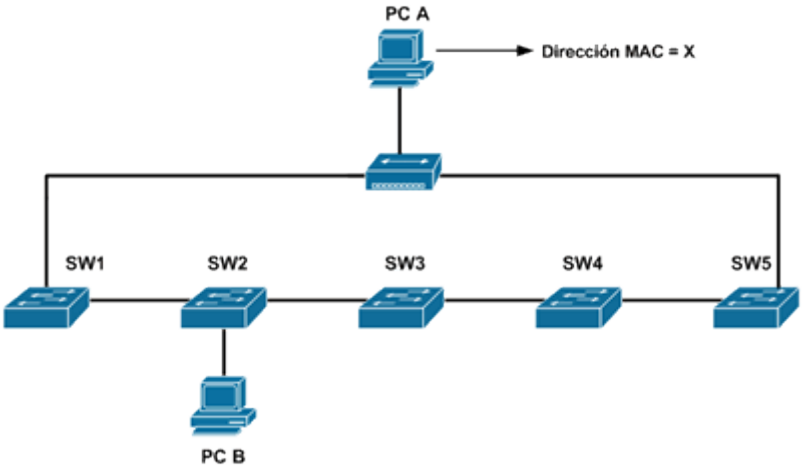
En un primer momento, el paquete se dirige a través de SW1 y SW2 los cuales guardan en su tabla que para alcanzar la MAC X hay que ir a la izquierda y los conmutadores SW4 y SW5 guardan en su tabla que para alcanzar la MAC X han de enviar los datos a su derecha. La tabla de cada conmutador aparece en rojo.

El problema llegaría cuando el paquete atravesase el conmutador SW3 desde un extremo, por ejemplo desde la derecha primero y SW3 guardaría que la MAC X se encuentra a su derecha y reenviaría el paquete a su izquierda hacia SW2 - SW1 y estos cambiarían el registro de su tabla indicando que la MAC X se encuentra a su derecha a partir de este momento, y del mismo modo, SW3 recibiría un paquete desde su izquierda indicando que la MAC X se encuentra a la izquierda y modificaría el registro de su tabla interna indicando que se encuentra a la izquierda y reenviando el paquete hacia la derecha a los conmutadores SW4 y SW5, los cuales modificarían sus tablas para enviar los datos dirigidos a la MAC X hacia su izquierda, ahora el recorrido de los datos y las tablas de cada conmutador serían las siguientes:

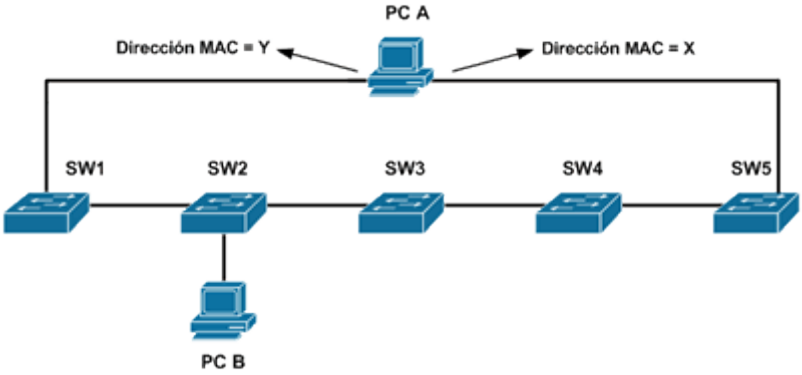


Como se puede ver, el problema que habría ahora para enviar datos a la estación con MAC X es que si otra estación con MAC Y estaba conectada al conmutador SW1 o SW2 y transmitía hacia el PC A, los datos irían a la derecha y al llegar al SW3, este los mandaría a la izquierda y viceversa, si se transmitiese al PC A desde SW3 o SW4 o SW5 los datos irían a la izquierda hasta llegar al SW2 y este los enviaría a la derecha haciendo imposible el envío hacia el PC A.

Esto mismo sucedería si una estación con una interfaz y por lo tanto una dirección MAC se conectase a un concentrador y el concentrador a su vez a dos conmutadores distintos pero conectados entre ellos a través de uno o más conmutadores:

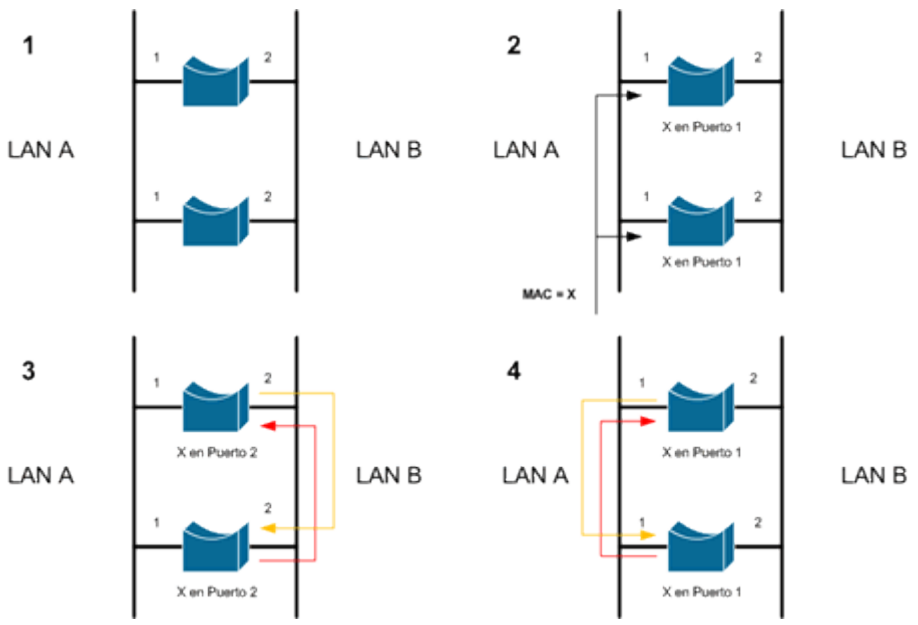


Por este motivo las soluciones a este problema sería añadir otra interfaz al PC A y cada interfaz tener una dirección MAC independiente:



De esta forma los conmutadores del SW1 al SW5 van a saber que para llegar a la MAC X han de enviar los datos a la derecha y para llegar a la MAC Y han de enviar los datos a la izquierda.

Otro peligro en las redes de conmutadores, que es catastrófico hasta el punto de dejar inutilizada a la red, es el hecho de producirse bucles de conmutación. Un bucle de conmutación se produce en redes donde existen enlaces redundantes entre conmutadores, los enlaces redundantes son importantes para hacer que una red siga estando disponible en caso de fallar un enlace y evitar que parte de la red quede aislada pero también los enlaces redundantes provocan bucles de conmutación y a diferencia de los bucles de enrutamiento donde hay unos contadores para limitar la cantidad de veces que un paquete puede estar circulando por la red, en los bucles de conmutación no hay ningún control sobre la cantidad de veces que un paquete puede estar circulando por lo que cuando se forma un bucle de conmutación, los datos estarán circulando para siempre hasta que se corte el bucle y dejen de transmitirse esos datos:



En el ejemplo se dan las siguientes fases:

1. Los puentes no tienen ninguna entrada en su tabla de direcciones MAC.
2. Una estación transmite un paquete con dirección MAC de origen X, los puentes guardan en la memoria que la estación con MAC = X se encuentra conectada al puerto 1.
3. Uno de los puentes va a ser el primero que transmitirá el paquete recibido en la LAN A a la LAN B, lo que hará que el otro puente interprete que la estación X se encuentra en la LAN B y guardará el paquete en la cola de la LAN A.
4. El puente envía el paquete de la LAN B a la LAN A y el otro puente renueva la entrada que dice que la estación X se encuentra en A y lo envía a la LAN B, este paquete será enviado a la LAN A por el otro puente y así sucesivamente.

Como se ve, un paquete inicial se multiplica por dos y no deja de circular de una red a otra, ocupando espacio constantemente en las colas de los puertos conectados a cada LAN. Con un paquete el impacto podría no notarse excesivamente, pero con un uso regular de la red, el impacto sería tan notable que llegaría un momento en el que no se podría trabajar con la red.

En la misma topología, por cada puente o conmutador que se añadiese se generaría un paquete más provocando el mismo efecto. La solución a este problema es el llamado Protocolo de Árbol Extensible (*Spanning-Tree Protocol* - STP).

3.3 El protocolo STP

El Protocolo Spanning-Tree (STP por sus siglas en inglés), fue desarrollado por Radia Perlman mientras trabajaba en *Digital Equipment Corporation* (DEC) en los años 80.

La idea del protocolo STP es que cualquier red de conmutadores y puentes puede simplificarse en una red en forma de árbol donde no hay ningún enlace redundante y la raíz del árbol es un conmutador o puente que establece la configuración que el resto de conmutadores y puentes han de tener. Para simplificar, cuando diga puentes me referiré a conmutadores y puentes salvo que especifique en algún caso únicamente conmutadores o únicamente puentes. La unidad de datos usada por los puentes se llama BPDU (del inglés "*Bridge Protocol Data Unit*").

El puente raíz se elige de forma dinámica. Las variables que STP utiliza para tomar decisiones son:

- Identificador del puente raíz: es la dirección MAC del dispositivo elegido raíz.
- Identificador del puente transmisor: es la dirección MAC del puente que transmite el paquete.
- Prioridad del puente: un valor global del punete el cual asigna una mayor importancia al puente para ser elegido raíz del árbol. Menor valor equivale a mayor prioridad.
- VLAN: número de VLAN en la que se encuentra asignado el puerto.
- Coste del enlace: un valor que establece el coste que tiene un enlace según su velocidad.
- Puerto: del puente que va a ser controlado por STP.

El orden de prioridad de las variables es:

1. Prioridad. Menor valor = Mayor prioridad.
2. ID de puente (dir. MAC) raíz transmitido. Valor numérico menor = más posibilidades de ser elegido raíz.
3. Coste del enlace por el cual se ha recibido el paquete.
4. ID de puente (dir. MAC) transmisor del paquete. Valor numérico menor = más posibilidades de ser elegido raíz.
5. Puerto por el cual transmite el puente transmisor del paquete.
6. VLAN: el protocolo STP se ejecuta en base a cada VLAN en los puentes de Cisco Systems, por lo que puede ser que un puerto esté en estado bloqueado para la VLAN 10 pero esté en estado de reenvío para la VLAN 20.

Los estados de STP por los que pasa un puerto son:

- ***Blocking*:** en este estado, un puerto no reenvía tramas, es el primer estado en el que se encuentra un puerto al encender un puente. Durante este estado se envía una BPDU por cada puerto. Características del estado de bloqueo:
  - Se descartan tramas recibidas.
  - Se descartan tramas recibidas en otro puerto y que tengan que ser enviadas por el puerto bloqueado.
  - No aprende direcciónes MAC.
  - Recibe BPDU.
- ***Listening*:** tras el estado de bloqueo, si STP considera que un puerto ha de estar activo, pasa a este estado. Características del estado de escucha:
  - Descarta tramas recibidas por el puerto.
  - Descarta tramas recibidas en otro puerto y que tengan que ser enviadas por el puerto en escucha.
  - No aprende direcciones MAC.

    - Recibe BPDU.
- ***Learning***: tras el estado de escucha, el puerto pasa al estado de aprendizaje. Características del estado de aprendizaje:
    - Descarta tramas recibidas por el puerto.
    - Descarta tramas recibidas en otro puerto y que tengan que ser enviadas por el puerto en aprendizaje.
    - Aprende direcciones MAC de estaciones.
    - Recibe BPDU.
- ***Forwarding***: tras el estado de aprendizaje, el puente pasa al estado de reenvío. Características del estado de reenvío:
    - Acepta las tramas entrantes y envía las tramas que correspondan por el puerto.
    - Aprende direcciones MAC de estaciones.
    - Recibe BPDU.
- ***Disable***: en este estado el puerto no participa ni en STP ni reenvía tramas. Este estado se da cuando se desactiva la interfaz por configuración o bien se desconecta el cable o bien no detecta ningún dispositivo conectado.
    - Descarta tramas entrantes y salientes.
    - No aprende direcciones MAC.
    - No recibe BPDU.

Parámetros configurables:

- ***Forward delay***: es la duración del estado de escucha ("*Listening*") y aprendizaje ("*Learning*"). Se establecen estos periodos ya que si en una red de puentes hay un cambio y STP cambiase el estado de enlaces de bloqueados al modo de reenvío, podrían formarse bucles por un periodo de tiempo breve que podrían saturar la red. Estableciendo este temporizador se pretende garantizar que en ningún momento se formarán bucles de conmutación. Un tiempo demasiado breve aquí provocaría que pudieran formarse bucles mientras que un tiempo demasiado largo provocaría tiempo innecesario sin servicio.
- ***Hello timer***: es el intervalo de envío de paquetes de configuración (con los que se decide quien será el puente raíz). Cuanto más breve sea el periodo menos probabilidades de falta de servicio en caso de caída de un enlace habrá pero más cpu, ancho de banda y memoria consumirá afectando al rendimiento, y al revés, cuando más amplio sea este periodo, más tardará STP en reconstruir el árbol pudiendo provocar falta de servicio.
- ***Max Age***: periodo de validez de un paquete de configuración en memoria, cuando caduca este periodo, si no recibe ningún paquete más desde el puente raíz, comienza a enviar BPDU indicándose a si mismo como puente raíz.

Aquí se puede ver una captura de pantalla con toda la información relevante de una BPDU de Spanning Tree con la información que se ha indicado hasta ahora:

```
⊟ IEEE 802.3 Ethernet
  ⊞ Destination: Spanning-tree-(for-bridges)_00 (01:80:c2:00:00:00)
  ⊞ Source: Cisco_44:23:c1 (00:08:a3:44:23:c1)
    Length: 38
    Trailer: 0000000000000000
⊟ Logical-Link Control
    DSAP: Spanning Tree BPDU (0x42)
    IG Bit: Individual
    SSAP: Spanning Tree BPDU (0x42)
    CR Bit: Command
  ⊞ Control field: U, func=UI (0x03)
⊟ Spanning Tree Protocol
    Protocol Identifier: Spanning Tree Protocol (0x0000)
    Protocol Version Identifier: Spanning Tree (0)
    BPDU Type: Configuration (0x00)
  ⊟ BPDU flags: 0x00
      0... .... = Topology Change Acknowledgment: No
      .... ...0 = Topology Change: No
  ⊟ Root Identifier: 4096 / 1 / 00:09:7c:3f:31:80
      Root Bridge Priority: 4096
      Root Bridge System ID Extension: 1
      Root Bridge System ID: 00:09:7c:3f:31:80
    Root Path Cost: 23
  ⊟ Bridge Identifier: 32768 / 1 / 00:08:a3:44:23:c0
      Bridge Priority: 32768
      Bridge System ID Extension: 1
      Bridge System ID: 00:08:a3:44:23:c0
    Port identifier: 0x8001
    Message Age: 2
    Max Age: 20
    Hello Time: 2
    Forward Delay: 15
```

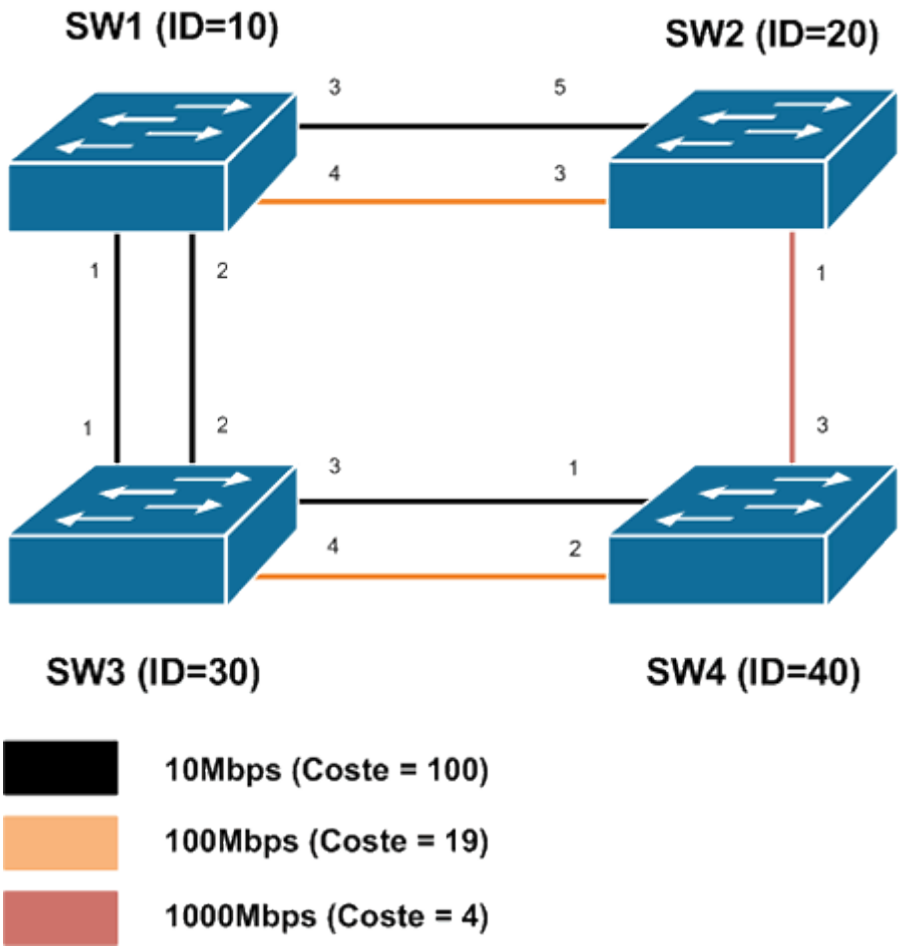
Cuando hay un cambio en un puerto que participa en STP el puente activa la "*flag*" llamada "*Topology Change*" y le da un valor binario de 1 para informar a los otros puentes de que ha habido un cambio.

La extensión que aparece como "*Root Bridge System ID Extension: ...*" es la VLAN en la que se está ejecutando STP.

El puente raíz se caracteriza por tener todos los puertos en estado de reenvío ("*forwarding*") aunque no tiene porque ser el único que tenga todos los puertos en este estado.

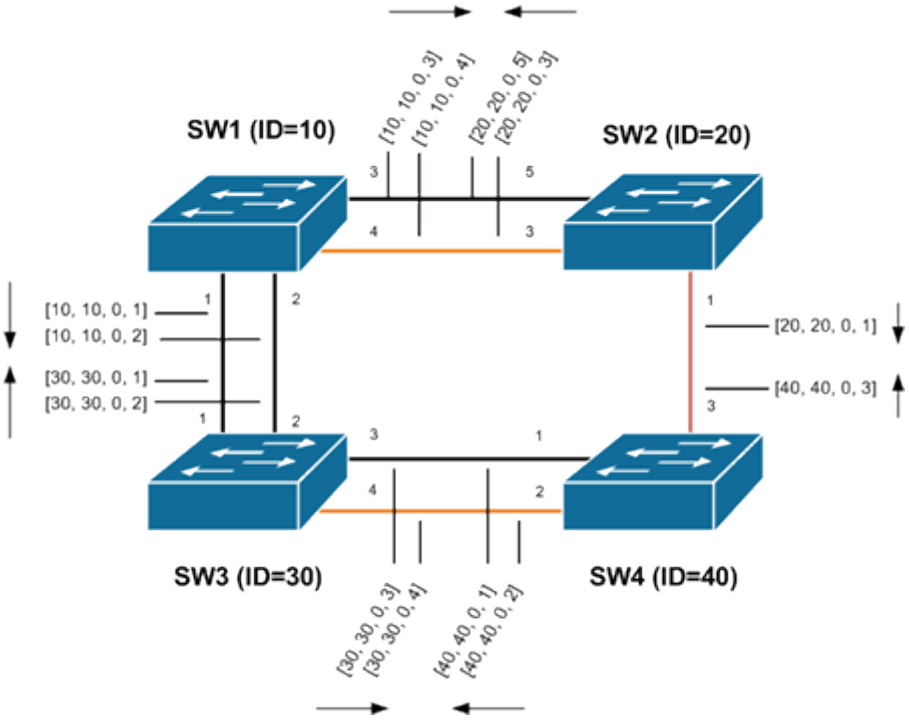
El siguiente ejemplo se simplificará suponiendo que todos los puertos se encuentran configurados en la misma VLAN, todos los puentes tienen la misma prioridad y se omitirá la prioridad del puerto dado que al tener todos la misma prioridad no afectará. Se describirá paso a paso:

Primer paso:



Topología de red inicial, diferentes puentes enlazados entre si con enlaces redundantes de distintas velocidades excepto el enlace entre SW2 - SW4 que no tiene enlace de respaldo y el enlace entre SW1 - SW3 que tiene ambos enlaces de la misma velocidad.

Segundo paso:

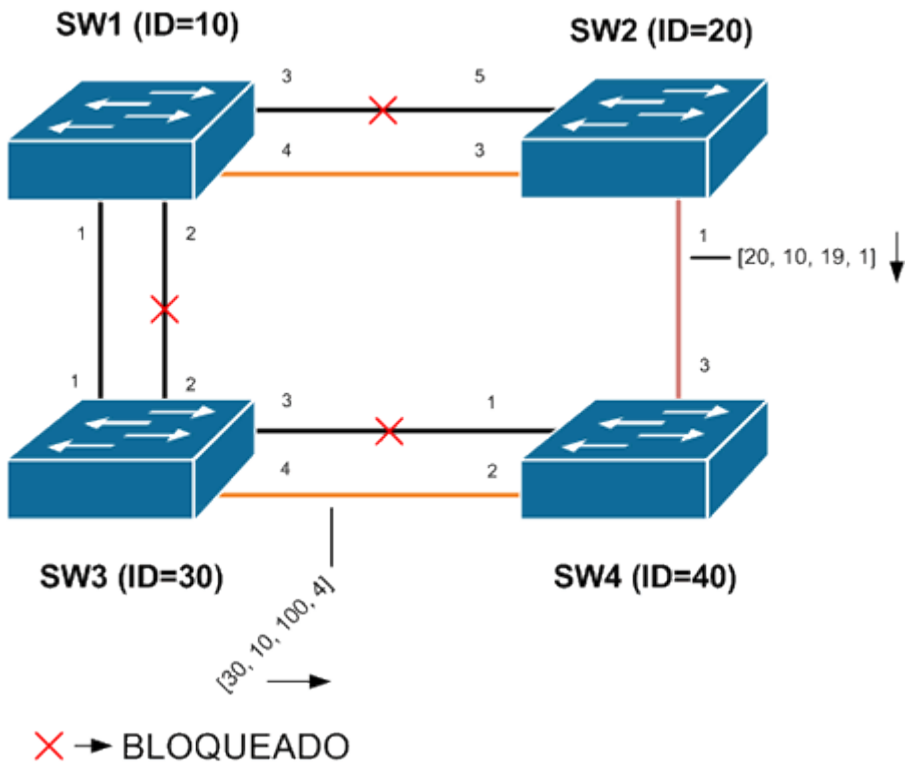


Cada puente envía por todos los enlaces activos una BPDU con los siguientes datos:

- Identificador del puente transmisor. En el caso de los puentes Cisco es la dirección MAC de la VLAN 1.
- Identificador del puente raíz. En el primer momento cada puente se considera a sí mismo como puente raíz. En el caso de los puentes Cisco es la dirección MAC de la VLAN 1.
- Coste del enlace. El coste del enlace es la suma de costes desde el puente que transmite la BPDU hasta el puente raíz. Como en el primer momento todos los puentes transmiten como puente raíz a si mismos, el coste del enlace es 0.
- Identificador de puerto. Es el número de puerto.

Tras este paso los puentes con enlaces redundantes (unidos por dos o más enlaces) decidirán qué enlaces estarán activos basándose en la **velocidad del enlace** o **identificador de puerto** y si ellos son puente raíz o es el puente transmisor de la BPDU basándose en el **identificador del puente** (ID).

Tercer paso:



Explicación de por qué se han bloqueado los puertos:

- SW1 - SW2: Enlaces redundantes. Se bloquea el enlace de 10Mbps (coste 100) por tener un coste superior al de 100Mbps (coste 19).
- SW1 - SW3: Enlaces redundantes. Se bloquea el enlace entre los puertos 2 - 2 ya que el identificador de ambos puertos (2) es superior al del otro enlace (1). En caso de que hubiese sido un enlace entre los puertos SW1:2 - SW3:1 se hubiese bloqueado igualmente el mismo enlace dado que aunque en el SW3 el puerto sería inferior al del otro enlace, el identificador de SW1 es inferior al de SW3, en este momento SW3 considera a SW1 el puente raíz, por lo que SW1 manda.
- SW3 - SW4: Enlaces redundantes. Se bloquea el enlace de 10Mbps por tener un coste superior al de 100Mbps como en el enlace SW1 - SW2.

Tras decidir que puertos se bloquean basándose en la información intercambiada en un primer momento, los puentes comparan los identificadores que reciben y envían por los otros puertos una BPDU con su identificador, el identificador del que ellos ahora consideran puente raíz, coste hasta el puente raíz y puerto por el cual se transmite.

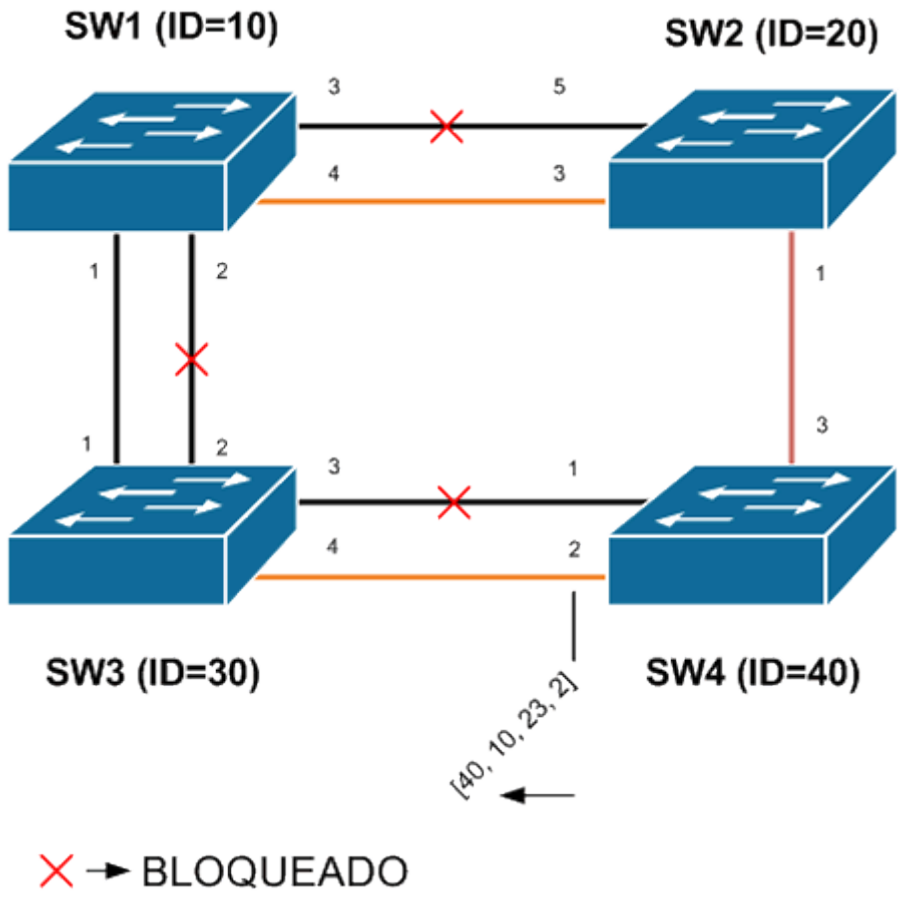
Las dos BPDU del ejemplo indicarían:

- SW3 a SW4:
  - 30: Identificador del puente transmisor.
  - 10: Identificador del puente raíz, como el propio ID (30) transmitido en el paso anterior es superior al ID (10) recibido del vecino, ya no se considera a sí mismo puente raíz.
  - 100: es el coste por defecto de un enlace de 10Mbps en STP. En este caso, entre SW3 y SW1 hay un enlace de 10Mbps, cuando SW4 reciba

la BPDU le sumará el coste del enlace entre SW2 y SW4 al coste recibido.
    - 4: puerto por el cual transmite.
- SW2 a SW4:
    - 20: Identificador del puente transmisor.
    - 10: Identificador del puente raíz, como el propio ID (20) es superior al ID (10) recibido del vecino ya no se considera a si mismo puente raíz.
    - 19: es el coste por defecto de un enlace de 100Mbps en STP. En este caso, entre SW2 y SW1 hay un enlace de 100Mbps, cuando SW4 reciba la BPDU le sumará el coste del enlace entre SW2 y SW4 al coste recibido.
    - 1: puerto por el cual transmite.

Tanto SW2 como SW3 omiten enviar la BPDU con la información recibida de SW4 ya que han determinado que no será elegido puente raíz por sus parámetros y no es necesario informar al vecino.

Cuarto paso:



Al ser SW4 el puente más remoto dentro del bucle, en este caso, es quien decidirá cómo se desarrollará el árbol. Comparemos los datos que recibió en el paso anterior, en negrita está el valor que toma como mejor:

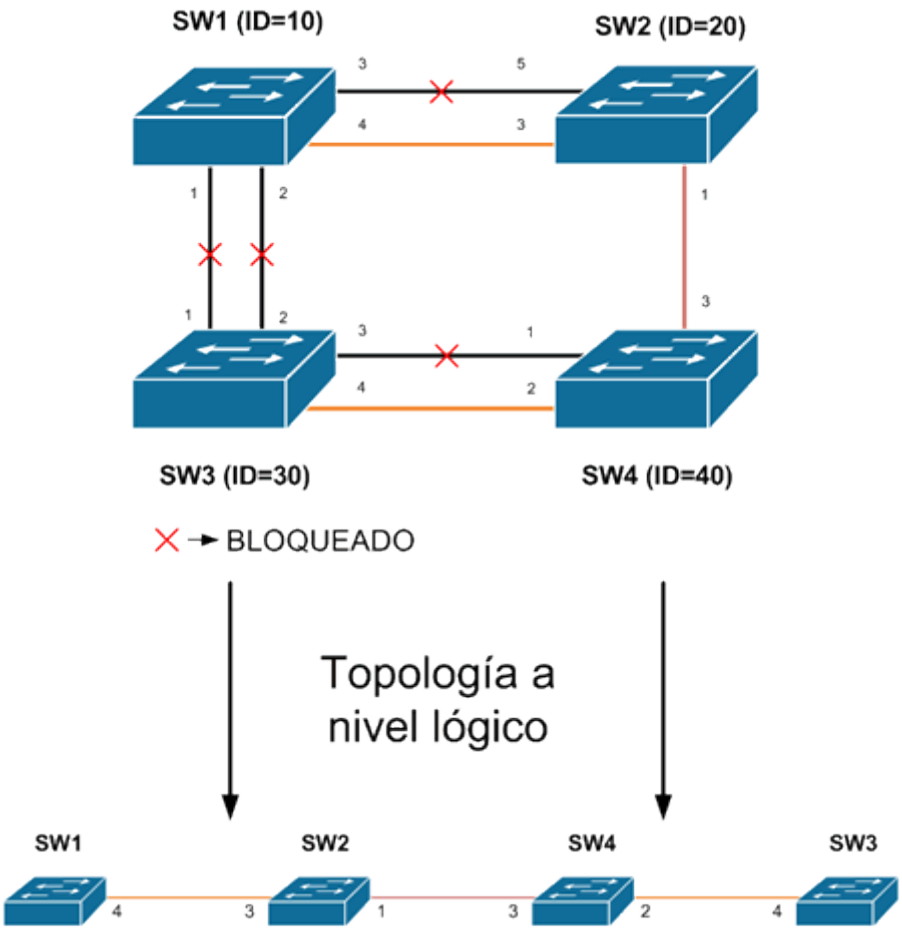
- Identificador del transmisor:
  - SW2: **20**
  - SW3: 30
- Identificador del puente raíz:
  - SW2: **10**
  - SW3: **10**
- Coste hasta el puente raíz:
  - SW2: **23**
  - SW3: 119
- Puerto por donde le transmiten el paquete:
  - SW2: **1**
  - SW3: 4

Según el orden de prioridades a la hora de tomar decisiones:

1. Identificador de puente raíz. Mejor valor = 10 (SW2 y SW3).
2. Coste del enlace. Mejor valor = 23 (SW2). A partir de aquí se decide ya que el mejor camino es por SW2.
3. Identificador de puente transmisor. Mejor valor = 20 (SW2).
4. Puerto por donde envía la BPDU el transmisor. Mejor valor = 1 (SW2).

Por lo que SW4 envía una BPDU a SW3 indicándole los datos recibidos de SW2 cambiando el ID del transmisor por su propio ID y añadiendo el coste del enlace SW2 - SW4 al coste recibido en la BPDU transmitida.

Quinto paso:



Tras comparar SW3 los datos recibidos de SW4 con los que tenía almacenados, SW3 decide poner el enlace SW1:1 - SW3:1 en estado de bloqueo ya que la información recibida de SW4 es mejor que la recibida directamente de SW1.

NOTA: El hecho de que SW3 ponga el puerto en estado de bloqueo significa que no reenvía datos, el puerto sigue recibiendo BPDU por si hubiese algún cambio y de hecho SW1 tiene el puerto en estado de reenvío, es decir, funcionando normalmente mientras que SW3 tiene el puerto en estado de bloqueo.

La configuración de STP se describirá cuando esté disponible dentro del apartado "Conmutación - Spanning Tree".

3.4 Redes locales virtuales (VLAN)

Las VLAN (del inglés "*Virtual Local Area Network*") están definidas por el 802.1q, se podrían describir como redes dentro de las redes, de hecho, en una red física puede haber una o más VLANs.

Se describirán muy brevemente ya que se explicarán más a fondo en la sección "Conmutación - VLAN y VTP" cuando esté disponible.

El funcionamiento descrito en pocas palabras viene a ser que se crea una VLAN en un conmutador, al crear la VLAN se le asigna un número, ése número será la etiqueta que se asignará a los puertos.

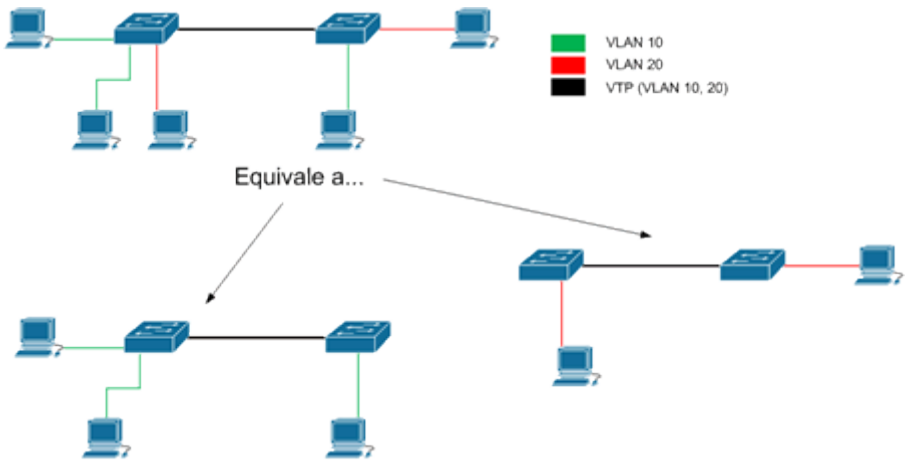
Una vez creada la VLAN se asignan puertos a dicha VLAN, una vez hecho esto, a todas las tramas que el conmutador reciba por dicho puerto se le colocará la etiqueta en la cabecera y se transmitirá la trama solamente a un puerto asignado a la misma VLAN, si el puerto con el dispositivo de destino está asignado a otra VLAN habrá que colocar un enrutador para que interconecte las dos redes.

Como se ha dicho, un puerto solamente puede estar asignado a una VLAN, si dos puertos de dos conmutadores distintos están conectados directamente pero están asignados a dos VLANs distintas, las tramas no se transferirán entre los conmutadores.

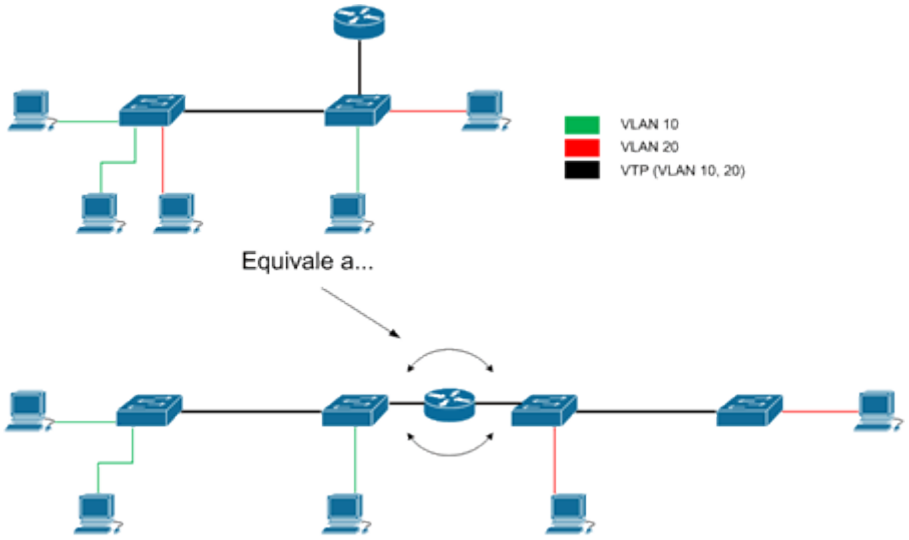
Para interconectar dos conmutadores entre ellos donde cada ambos tienen varias VLAN comunes configuradas y se quieren interconectar entre ellas, en lugar de asignar un puerto por VLAN en cada uno de los conmutadores, se puede configurar un puerto troncal al cual se le especificará el tráfico de las VLANs que se permiten y el de cuales no. El puerto troncal también se usa para interconectar conmutadores con varias VLAN con un enrutador y así permitir que los dispositivos de cada una de ellas puedan comunicarse.

El protocolo usado para establecer los enlaces troncales entre VLAN es el protocolo VTP (del inglés "VLAN Trunking Protocol" - Protocolo Troncal de VLAN).

Ejemplo sencillo descriptivo del funcionamiento de las VLAN:

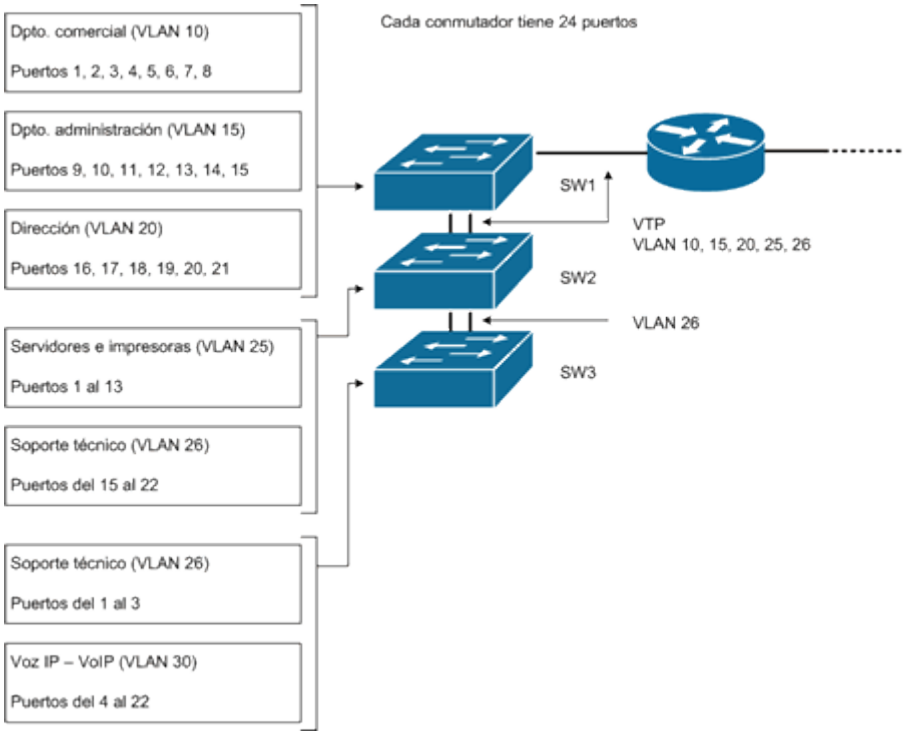


En este caso serían como dos redes independientes y sin conexión entre ellas. Para que hubiese conexión habría que añadirse un enrutador:



Al añadir el enrutador y configurar el enlace troncal pasaría a haber conectividad entre las dos VLAN, ahora para restringir la conectividad o limitar los servicios a los que pueden acceder las estaciones entre VLANs habría que aplicar Listas de Control de Acceso (ACL) en las subinterfaces del enrutador.

Un ejemplo más fiel al uso en la realidad de VLAN y VTP sería el siguiente:



En este caso se tiene una empresa con los departamentos:

- Dirección
- Departamento de administración
- Departamento comercial
- Soporte técnico

Y aparte dos redes asignadas a los servicios:

- Servidores e impresoras
- Voz IP

Cada departamento y servicio tiene unos puertos asignados por el administrador y establece que todos los departamentos pueden conectarse a los servidores e impresoras en red pero la red de Voz IP está aislada del resto de la red porque el servidor de VoIP se encuentra dentro de la misma VLAN y el administrador tiene acceso físico a la máquina por lo que no es necesario poder acceder desde otra red al mismo.

El enlace entre SW1 y SW2 y el enlace entre SW1 y el enrutador, son de tipo troncal dado que ha de pasar el tráfico de varias VLAN mientras que como en el SW3 solamente ha de pasar la VLAN 26 que también se encuentra configurada en el SW2, no es necesario configurar el enlace como VTP sino configurándolo en la VLAN 26 ya se transferirán los datos.

Como se ha dicho antes, entre puertos de la misma VLAN los datos se transfieren, bien estén en el mismo conmutador o bien en distintos conmutadores conectándolos entre ellos, si hay que conectar dos conmutadores para unir una sola VLAN configurada en ambos, configurando el puerto o los puertos que los interconectan en la misma VLAN hará que las estaciones conectadas en las VLAN se puedan comunicar entre ellas sin necesidad de un enrutador.

Esto se tocará de manera más extensa dentro del apartado "Conmutación" cuando esté disponible.

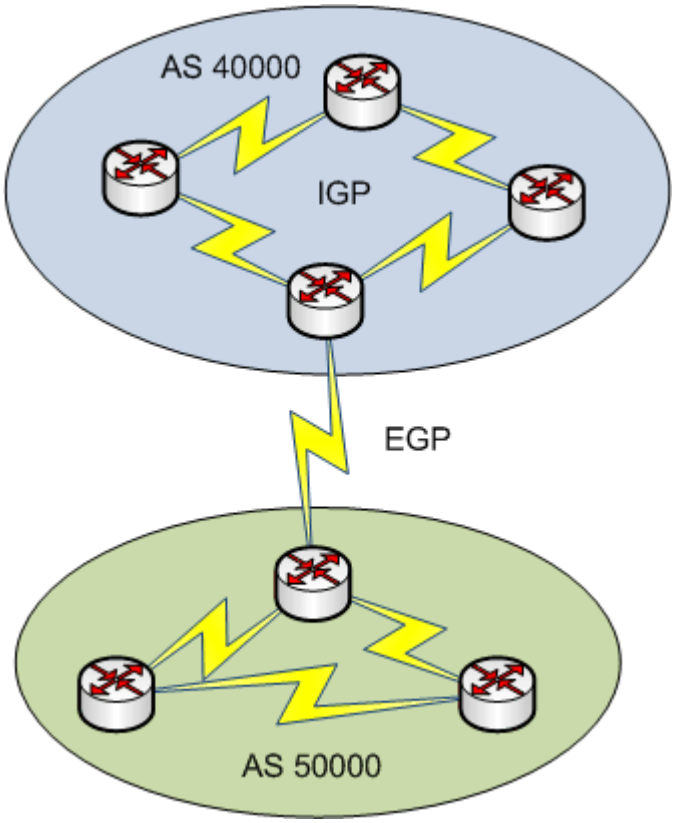
4 .Enrutamiento IP

El enrutamiento es el hecho de transmitir los datos desde un origen a un destino por un camino u otro. Tanto los protocolos de enrutamiento dinámicos como el enrutamiento estático toman decisiones basadas en la información de la capa 3 del modelo OSI, es decir, las direcciones IP.

Una red es un conjunto de dispositivos conectados, estos dispositivos están diseñados para transmitir datos, para que los datos lleguen de un dispositivo a otro que no están conectados directamente hace falta algún sistema que busque el camino para ir de un sitio a otro.

Los protocolos de enrutamiento se pueden clasificar en dos tipos:

- IGP: Protocolos que trabajan dentro de un AS.
- EGP: Protocolos que intercambian información entre Sistemas Autónomos.



Los protocolos de enrutamiento se pueden diferenciar también en otros dos grandes grupos según la información que transmiten en las actualizaciones:

- Con clase: no transmiten la máscara de red para cada red. Ejemplo: RIPv1, IGRP.
- Sin clase: transmiten la máscara de red para cada red. Ejemplo: RIPv2, EIGRP, OSPF, ISIS, BGP, ODR.

El hecho de que transmita o no la máscara de red, tiene más relevancia de la que pueda parecer ya que si no se transmite la máscara, el protocolo de enrutamiento "piensa" que todas las redes tienen la misma máscara y en caso de que se haya subredes con máscaras de longitud distinda (VLSM) como por ejemplo que unas

tengan /30 y otras /28, el protocolo de enrutamiento no tomará las decisiones adecuadas y habrá problemas en la red. El hecho de que el protocolo crea que todas las redes tienen la misma máscara de red, es lo que lleva a llamarlo con clase ya que no distingue entre redes de clase A, B, C, solo se queda con las de una clase.

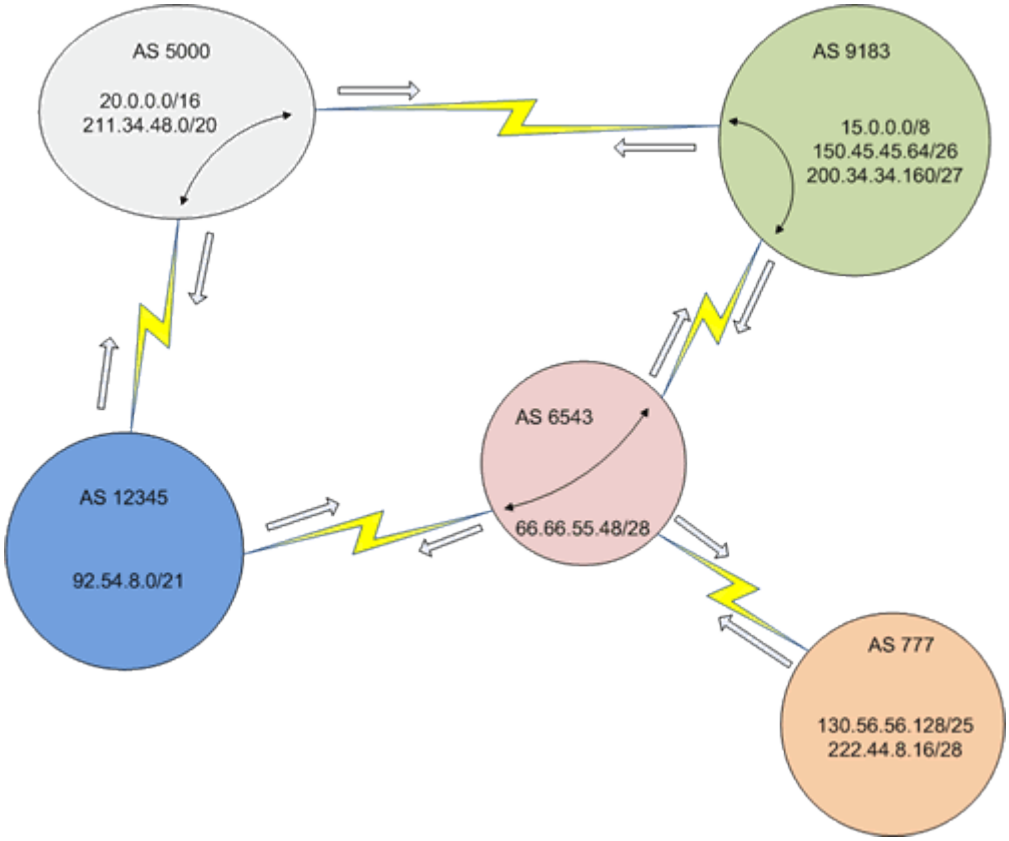
Al transmitir la máscara de red, los protocolos de enrutamiento pueden distinguir las subredes y transmitir los datos por el camino que sea más adecuado.

Otra clasificación divide a los protocolos de enrutamiento en tres grupos:

- Vector-Distancia: toman sus decisiones en función de los saltos hasta un destino. Ejemplo: RIPv1, RIPv2, IGRP, EIGRP.
- Estado de Enlace: crean un mapa completo de la red y así calculan el camino más rápido. Ejemplo: OSPF, IS-IS.
- Políticas: basa sus decisiones en políticas establecidas por el administrador. Ejemplo: BGP.

Cuando una red está gestionada por un mismo grupo de administradores, a esto se le llama Sistema Autónomo (AS en inglés), está registrado en el organismo correspondiente según su localización geográfica (RIPE, ARIN, APNIC, LACNIC, ...) y se identifica con un número que puede ir del 0 al 65535 (siendo los asignables a AS enrutados el rango del 1 al 54271).

Un sistema autónomo a la vista de otros sistemas autónomos es un conjunto de prefijos CIDR a los cuales se puede llegar mediante una ruta u otra, pero un AS no conoce la estructura (topología, velocidad de los enlaces, ...) de los otros Sistemas Autónomos:



Los Sistemas Autónomos (AS) intercambian información de las redes que conocen, se pueden establecer políticas para que determinadas redes se publiquen por un enlace u otro o que otras redes no sean transmitidas a los AS vecinos. Se pueden distinguir entre dos tipos de Sistemas Autónomos:

- Ordinarios: publican a los AS vecinos las redes que se encuentran dentro del AS.
- De tránsito: los AS de tránsito publican además de las redes que se encuentran dentro del mismo AS, las redes de otros AS. Este tipo de Sistema Autónomo es lo que se llama un Proveedor de Servicios de Internet (ISP).

Por lo que en el gráfico anterior, los AS 12345 y 777 serían de tipo ordinario y los AS 5000, 6543 y 9183 serían de tránsito.

En esto se profundizará más en el apartado BGP.

Las rutas que almacena el enrutador, se guardan en la tabla de enrutamiento. En el caso de protocolos de enrutamiento dinámicos la tabla se llena con las rutas que recibe en las actualizaciones de enrutamiento, en el caso del enrutamiento estático, se configura manualmente y quedan almacenada en la tabla de enrutamiento. Se puede dar el caso de que para una misma red haya varias rutas, en este caso siempre tienen preferencia las rutas con AD menor y métrica menor.

El proceso para incluir rutas en la tabla de enrutamiento es el siguiente:

- Cuando un router recibe una ruta, lo primero que hace es verificar que la IP del siguiente salto responde, en caso de que no responda la ruta queda descartada directamente.
- Para una misma red de destino, el router elige la fuente / protocolo / ruta con Distancia Administrativa (AD) menor y descarta las otras, aunque pueden quedar en una base de datos -EIGRP/OSPF- por si la ruta elegida pasase a no ser operativa y fuese necesario buscar una nueva.
- En caso de que para una misma red de destino haya más de un camino, siempre se elegirá el camino con métrica inferior, en caso de que haya más de un camino con la métrica menor, se añadirán todos a la tabla de enrutamiento. En el caso de EIGRP se puede dar el caso de que según la configuración se añadan rutas con métricas distintas.
- En caso de que el enrutador reciba una ruta hacia una dirección IP con propiedades mejores (por ejemplo: máscara más específica -/24 es más específica que /16-), la ruta se añade a la tabla de enrutamiento.

En el caso del Cisco IOS, el enrutador, en su tabla de enrutamiento, guarda los siguientes datos:

- Red de destino.
- Máscara de la red.
- Distancia Administrativa: es la prioridad que tiene uno u otro protocolo. Siendo 0 la mayor prioridad y 255 la menor.
- Métrica: el coste que calcula el protocolo de enrutamiento que tiene la ruta para llegar a la red de destino.
- Siguiente salto: es la IP de la interfaz del router vecino, directamente conectada al router del cual estamos viendo la tabla de enrutamiento.
- Tiempo desde que se ha recibido la última actualización con esta información.
- Interfaz por donde saldrán los datos.

Un ejemplo de tabla de enrutamiento, tal y como están configurados los enrutadores de pruebas sería:

```
R3640#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

     E1 - OSPF external type 1, E2 - OSPF external type 2

     i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2

     ia - IS-IS inter area, * - candidate default, U - per-user static route

     o - ODR, P - periodic downloaded static route


Gateway of last resort is not set

   102.0.0.0/32 is subnetted, 1 subnets

O E1    102.102.102.102 [110/1592] via 200.100.50.1, 00:07:37, Serial2/0

   50.0.0.0/28 is subnetted, 1 subnets

C      50.30.25.16 is directly connected, Ethernet3/0

   103.0.0.0/32 is subnetted, 1 subnets

O E1    103.103.103.103 [110/1592] via 200.100.50.1, 00:07:37, Serial2/0

   2.0.0.0/32 is subnetted, 2 subnets

C      2.2.2.7 is directly connected, Loopback123

S      2.2.2.8 [253/0] via 200.100.50.9

   101.0.0.0/32 is subnetted, 1 subnets

O E2    101.101.101.101 [110/1] via 50.30.25.17, 00:07:38, Ethernet3/0

   200.100.50.0/24 is variably subnetted, 5 subnets, 4 masks

O IA    200.100.50.16/28 [110/4687] via 200.100.50.1, 00:07:38, Serial2/0

C      200.100.50.8/29 is directly connected, Serial2/2

C      200.100.50.9/32 is directly connected, Serial2/2

C      200.100.50.0/30 is directly connected, Serial2/0

C      200.100.50.1/32 is directly connected, Serial2/0

   172.16.0.0/29 is subnetted, 1 subnets

O E1    172.16.0.0 [110/1592] via 200.100.50.1, 00:07:40, Serial2/0

   10.0.0.0/8 is variably subnetted, 11 subnets, 5 masks

O E2    10.0.0.10/32 [110/1] via 50.30.25.17, 00:07:40, Ethernet3/0

O E2    10.0.0.8/29 [110/1] via 50.30.25.17, 00:07:40, Ethernet3/0

O E1    10.0.0.9/32 [110/1592] via 200.100.50.1, 00:07:40, Serial2/0

O E2    10.0.0.2/32 [110/1] via 50.30.25.17, 00:07:40, Ethernet3/0

O E2    10.0.0.0/29 [110/1] via 50.30.25.17, 00:07:40, Ethernet3/0

O E2    10.0.0.1/32 [110/1] via 50.30.25.17, 00:07:40, Ethernet3/0
```

```
O E2    10.0.0.18/32 [110/1] via 50.30.25.17, 00:00:08, Ethernet3/0

O E2    10.0.0.16/28 [110/1] via 50.30.25.17, 00:07:40, Ethernet3/0

O E1    10.0.0.32/27 [110/1592] via 200.100.50.1, 00:07:41, Serial2/0

O E2    10.0.0.64/30 [110/1] via 50.30.25.17, 00:07:41, Ethernet3/0

O E2    10.0.0.128/30 [110/1] via 50.30.25.17, 00:07:41, Ethernet3/0

    192.168.0.0/30 is subnetted, 1 subnets

O       192.168.0.0 [110/1572] via 200.100.50.1, 00:08:01, Serial2/0

    60.0.0.0/29 is subnetted, 1 subnets

O E2    60.80.40.8 [110/1] via 50.30.25.17, 00:07:41, Ethernet3/0
```

Enrutamiento estático

El enrutamiento estático permite configurar manualmente el camino que deben tomar los datos para ir de un lugar a otro. Se puede configurar la distancia administrativa independientemente para cada ruta, especificar una ruta por donde enviar los datos para redes que desconoce (ruta por defecto).

En la configuración de una ruta estática puede especificarse o bien la interfaz por donde enviar los datos o bien la dirección IP del siguiente salto. De forma predeterminada, en caso de especificarse la interfaz la AD es de 0 y si se especifica el siguiente salto la AD es 1.

Se puede configurar de forma que la IP del siguiente salto sea la puerta de enlace obtenida de un servidor DHCP.

Enrutamiento bajo demanda (ODR – On Demand Routing)

El enrutamiento bajo demanda es un sistema de Cisco, solamente se puede aplicar en una topología "Hub and Spoke" donde un enrutador central, al cual el resto de enrutadores están conectados, usa los paquetes CDP propios de Cisco para darles una ruta por defecto que apunte hacia él mismo. Por otra parte, los enrutadores remotos informan al enrutador central de las redes que tienen directamente conectadas y así el router central sabe dónde está cada red y los enrutadores remotos saben que todo lo han de enviar al router central y para que lo gestione.
Es un protocolo sin clase.

Protocolo RIP (Routing Information Protocol)

El protocolo RIP es de tipo Vector-Distancia, la métrica es el número de saltos de una red a otra.
La versión 1 de RIP es un protocolo con clase y envía las actualizaciones cada 30 segundos por broadcast.
La versión 2 de RIP es un protocolo sin clase y envía las actualizaciones cada 90 segundos por multicast.

En las actualizaciones envía todo el contenido de la tabla de enrutamiento.

Protocolo EIGRP (Enhaced Interior Gateway Routing Protocol)

El protocolo EIGRP es un protocolo propietario de Cisco, es de tipo híbrido, lo que significa que tiene características de los protocolos Vector-Distancia y Estado de Enlace. Es un protocolo sin clase, usa el sistema DUAL para establecer la ruta más corta y mantiene contacto periodico con los vecinos para detectar cambios o caídas. Cuando hay un cambio envía una actualización solamente con la información del cambio a los routers vecinos.

Este protocolo calcula las métricas mediante una compleja fórmula que toma en cuenta varias variables que son el ancho de banda, carga de un enlace, retardo total de la ruta y confiabilidad. Además cada una de estas variables se puede ponderar para ajustar la configuración a la red y así optimizar el tiempo de convergencia y adaptación a los cambios en la topología.

El predecesor de EIGRP fue el protocolo IGRP, el cual era de tipo con clase. Cisco, para hacer más sencilla la conversión de IGRP a EIGRP hizo que fuesen compatibles, para convertir la métrica de IGRP a EIGRP se ha de multiplicar IGRP * 256, y dividir entre 256 para pasar de EIGRP a IGRP. Esto se debe a que la métrica de IGRP es de 24 bits y la de EIGRP es de 32, 2^8 = 256.

Tanto IGRP como EIGRP guardan varias tablas en la memoria para así llevar un control de la red y reducir el tiempo de convergencia cuando haya un cambio. Están diseñados para ser implementados en redes jerárquicas, admite resumen de ruta.

Protocolo OSPF (Open Shortest Path First)

El protocolo OSPF es un protocolo de estado de enlace, tiene en memoria un esquema de toda la red y es bastante veloz a la hora de convergir aunque no tanto como EIGRP.

Es un protocolo de enrutamiento diseñado para interconectar redes jerárquicas, admite resumen de ruta y divide la red en areas. Los enrutadores que ejecutan OSPF intercambian diferentes tipos de paquetes llamados LSA (Link-State Advertisement). Las áreas pueden ser de diferentes tipos. En cada tipo de área se intercambian unos tipos u otros de LSA.

OSPF ejecuta el algoritmo Dijkstra para establecer cual es el camino más corto. La métrica se forma a partir de una sencilla operación en la cual el valor que establece el resultado el el parámetro "bandwidth" de cada interfaz física.

Cada enrutador tiene un identificador en formato de dirección IPv4.

Protocolo IS-IS (Intermediate System - Intermediate System)

El protocolo IS-IS es un protocolo de estado de enlace. De la misma manera que OSPF ejecuta el algoritmo de Dijkstra para establecer el camino más corto. La métrica es el costo de los enlaces y en el caso de los enrutadores Cisco, el costo ha de ser introducido manualmente en la configuración de la interfaz o sino toma por defecto el valor de 10.

A diferencia de los otros protocolos, IS-IS no usa el protocolo IP para transmitir información entre enrutadores. Esto le permite ser flexible a la hora de utilizar un

direccionamiento IPv4 o IPv6 ya que OSPF por su parte está diseñado para IPv4 y se tuvo que diseñar de nuevo (OSPFv3) para que soportase IPv6.

Al igual que OSPF, cada enrutador tiene un identificador, que en este caso es una dirección CLNS, un sistema diseñado por DEC y donde los host de una red son ES (End Systems) y los enrutadores son IS (Intermediate Systems).

Hay tres tipos de IS (router), los de nivel 1, nivel 2 y nivel 1-2 (abarcan ambos niveles). Los ES solamente establecen adyacencias con los IS de nivel 1, los IS de nivel 1 con los IS de nivel 1 y nivel 1-2 en la misma área; los IS de nivel 1-2 con otros IS de nivel 1 en la misma área, de nivel 1-2 de cualquier área y de nivel 2 en cualquier área; los IS de nivel 2 con otros IS de nivel 1-2 en cualquier área y de nivel 2 en cualquier área.

Redistribución entre protocolos

La redistribución de rutas es el hecho de que el router, en el cual se ejecutarán dos o más protocolos de enrutamiento (incluyendo al enrutamiento estático como un protocolo de enrutamiento), dará a conocer al protocolo A las rutas que conoce el protocolo B (para que se informe al revés, de A a B se ha de hacer explícitamente, el hecho de redistribuir de un protocolo a otro no significa una redistribución recíproca, la redistribución se hace en la configuración de cada protocolo de enrutamiento).

A la hora de redistribuir rutas, hay métodos para filtrar que determinadas rutas no se redistribuyan o cambien las propiedades de las rutas. En concreto hay dos opciones:

- distribute-list: son listas de control de acceso (ACL) simples que filtran redes de las actualizaciones de enrutamiento que envía o recibe un enrutador.
- route-map: son conjuntos de ACL simples donde se pueden filtrar redes e incluso cambiar parámetros como la métrica, etc...

Protocolo BGP (Border Gateway Protocol)

El protocolo BGP es un protocolo basado en políticas, lo que significa que no se basa en el número de saltos ni velocidad de enlaces ni nada parecido, sino que toma decisiones basándose en las políticas (reglas) que el administrador de la red ha implementado.

La misión de BGP es intercambiar información entre los Sistemas Autónomos (AS), dentro de un AS los routers también establecen sesiones, estas son llamadas sesiones iBGP (internal BGP) mientras que las sesiones entre AS son llamadas eBGP (external BGP). Cada enrutador ejecutando BGP dentro de un AS se conecta con los otros enrutadores que ejecutan BGP.

BGP tiene mecanismos para garantizar que no se producen bucles de enrutamiento, de todas formas pueden ocurrir. También hay lo que se llaman "agujeros negros" que producen que el tráfico vaya a un destino que no es el que toca. Esto se puede producir cuando un AS de tránsito (ISP) redistribuye rutas de otros AS con un prefijo más largo que el que el AS original, es decir, redistribuye rutas más precisas que las redistribuidas por el AS propietario de dichas rutas. Por ejemplo:

El AS xxxxx pública:

- 111.11.0.0/18
- 123.48.0.0/12

El AS yyyyy redistribuye:

- 111.11.0.0/24
- 111.11.1.0/24
- 111.11.2.0/24
- ...

Al redistribuir el AS yyyyy redes más precisas en lugar de una suprared que abarque todas las subredes (como hace el AS xxxxx), el resto de AS toman por preferible las subredes /24 en lugar de las supranet /18, en lugar de algo muy general, es algo más concreto, y se le da prioridad sobre otras redes con máscaras más generalistas.

Algo así fue lo que ocurrió en el incidente del AS 7007 en el año 1997 y que llevó a cambios importantes en las configuraciones BGP de los ISP. En este incidente el AS 7007 comenzó a publicar de repente sin ningún motivo aparente rutas con prefijo /24 lo que hizo que todo el tráfico de Internet del momento fuese dirigido a una conexión de 45Mbps durante unas horas e incluso una vez desconectada la línea las rutas seguían apuntando a la conexión del AS 7007 hasta que las rutas fueron propagadas de nuevo desde los AS propietarios y el tráfico volvió a fluir de nuevo con normalidad.

Otro protocolo de enrutamiento EGP que apenas se usa ya es FSPF.

5. Frame Relay

5.1 Introducción

La tecnología Frame Relay, ya en desuso con la implementación de ATM, apareció a finales de los años 80.

Esta tecnología coloca los datos en tramas de tamaño variable (al contrario de ATM donde las tramas son de 48+5 bytes), las tramas se envían por circuitos virtuales que pueden ser permanentes (PVC) o conmutados (SVC). En su momento era una alternativa económica a las líneas dedicadas y permitía transmitir una cantidad de datos elevada -según los criterios de la época-.

Los circuitos virtuales se identifican por el DLCI (Data Link Connection Identifier), cada circuito tiene un DLCI, y un switch FR no puede tener repetido ningún DLCI. De esta forma en una sola conexión física puede haber hasta 991 DLCI dependiendo de las extensiones LMI:

- ANSI/q933a: del 0 al 15 y del 993 al 1023 están reservados, se pueden asignar del 16 al 992 por lo que hay 977 disponibles.
- Cisco: del 0 al 15 y del 1008 al 1023 están reservados, se pueden asignar del 16 al 1007 por lo que hay 991 disponibles.

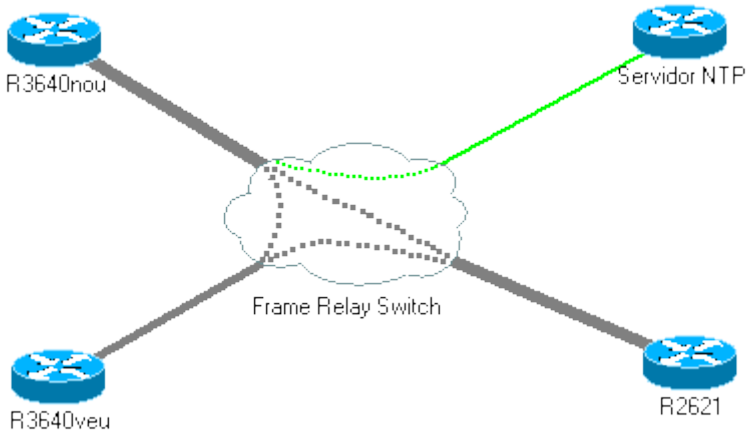
Las interfaces pueden ser Punto a Punto (Point-to-Point, PtP) o Punto a Multipunto (Point-to-MultiPoint, PtMP). Las interfaces PtP se comportan como una linea dedicada normal y las interfaces PtMP como un entorno de Acceso Múltiple sin Broadcast (NonBroadcast Multiple Access, NBMA).

5.2 Aplicación

Aunque la tecnología FR está cayendo en desuso, las características que tiene, a nivel práctico, son parecidas a las de ATM y es una forma sencilla y bastante económica de poder implementar un entorno NBMA en comparación a las alternativas y así poder comprobar el funcionamiento de algunos protocolos de enrutamiento en un pequeño laboratorio casero.

Se usarán PVC para así tener la ventaja de que con una interfaz poder establecer varios circuitos virtuales e implementar un enlace PtP y otro PtMP de malla completa.

5.3 Topología de laboratorio



En el esquema inicial de la red, numeré los dispositivos del 1 al 9 (sin contar el Switch FR y contando el Servidor NTP), por lo que según el esquema que me hice, cada uno de los dispositivos que aparece en la imagen tiene el siguiente número:

- R3640nou: 2
- R2621: 3
- R3640veu: 4
- Servidor NTP: 9

Las líneas punteadas de la imagen son los circuitos virtuales, como se ve hay dos colores distintos. La línea punteada de color verde es un circuito virtual PtP y las grises son circuitos virtuales PtMP formando una malla completa. Hay que decir que al ser interfaces configuradas en modo Punto a MultiPunto no hay PVC como tal, sino que se relaciona la IP de un vecino con un DLCI para que el Switch FR envíe las tramas a donde corresponda. La configuración del Switch FR es la misma si se hace con subinterfaces PtP o configurando las subinterfaces como PtMP. El hecho de configurar los routers con subinterfaces PtMP permite que todos estén en la misma subred, mientras que si se configurase la malla mediante subinterfaces PtP cada PVC tendría una subred. Por eso que aunque en el entorno NBMA se diga que hay PVC, en realidad todo forma parte de una misma red y actúa como si fuese una LAN pero sin difusión (broadcast).

Para calcular la canidad de PVC necesarios para formar la malla completa se utiliza la fórmula:

$$PVC = (n * (n-1)) / 2$$

En este caso son 3 routers y quedaría:

$$PVC = (3*(3-1))/2 = (3*2)/2 = 6/2 = 3$$

Por lo que se necesitan 3 PVC para formar la malla completa entre los 3 routers y son las 3 líneas grises punteadas de la imagen.

Sabiendo la cantidad de PVC, ahora hay que asignar los DLCI para cada subinterfaz y aprovechando que cada router está numerado se usarán esos mismos números para asignar los DLCI, así estos serán los DLCI:

- R3640nou (2) - R2621 (3) = 123 y 132
- R3640nou (2) - R3640veu (4) = 124 y 142
- R2621 (3) - R3640veu (4) = 134 y 143

Para el enlace PtP entre R3640nou y el Servidor NTP los DLCI serán:

- R3640nou (2) - Servidor NTP (9) = 209 y 902

El primer número corresponde al DLCI del primer router hacia el Switch FR y el segundo número al DLCI del segundo router al Switch FR.

Para la configuración del Switch FR, que es un router Cisco 2521, se trata simplemente de relacionar el DLCI de una trama que llegue al Switch por una interfaz y que lo envíe por otra interfaz con otro DLCI.

Antes de nada se configurará el Switch FR y posteriormente los cuatro routers conectados a la nube FR.

Switch FR (2511):

```
frame-relay switching

interface Serial0
 description To R3640nou.
 no ip address
 encapsulation frame-relay IETF
 clock rate 1000000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 123 interface Serial1 132
 frame-relay route 124 interface Serial2 142
 frame-relay route 209 interface Serial3 902

interface Serial1
 description To R2621.
 no ip address
 encapsulation frame-relay IETF
 clock rate 2000000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
```

```
 frame-relay route 132 interface Serial0 123
 frame-relay route 134 interface Serial2 143

interface Serial2
 description To R3640veu.
 no ip address
 encapsulation frame-relay IETF
 clock rate 64000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 142 interface Serial0 124
 frame-relay route 143 interface Serial1 134

interface Serial3
 description To NTP Server (R2511RJ).
 no ip address
 encapsulation frame-relay IETF
 clock rate 64000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 902 interface Serial0 209
```

Cuando un router va a actuar de conmutador Frame Relay, hay que introducir el comando de configuración global:

Router(config)#frame-relay switching

Descripción de la configuración de la interfaz conectada a R3640nou:

interface Serial 0	Nombre y número de la interfaz.
no ip address	Al ser Switch FR, no toma decisiones ni tiene conectividad de capa 3, por lo que no se configura IP, y aunque se configurase una no tendría efecto.
encapsulation frame-relay IETF	Se configura la encapsulación de tipo frame-relay de tipo IETF.
clock rate 1000000	Velocidad de datos de 1Mbit. Es el reloj que sincroniza los dos extremos de la conexión.
frame-relay lmi-type ansi	Extensiones LMI de tipo ANSI (necesarias para hacer PtMP)

frame-relay intf-type dce	Se especifica que esta interfaz es DCE (tiene configurado el reloj de sincronización)
frame-relay route 123 interface Serial1 132 frame-relay route 124 interface Serial2 142 frame-relay route 209 interface Serial3 902	Se configuran las rutas para las tramas en función del DLCI. La primera ruta indica que cuando reciba una trama con DLCI origen 123 que la ha de enviar por la interfaz Serial 1 con DLCI 132.

Las interfaces del Switch FR que estén configuradas como FR (puede haber interfaces conmutando tramas y otras enrutando paquetes) han de ser siempre DCE.

6. C.D.P. - Cisco Discovery Protocol

6.1 Introducción

El protocolo de descubrimiento de Cisco Systems, es un protocolo de capa 2 (no es de transporte sino que actúa en la capa de enlace de datos) que tiene la finalidad de dar a conocer la presencia de un dispositivo Cisco a sus vecinos. Al ser un protocolo propietario de Cisco, solamente es configurable en los enrutadores de esta marca.

Cuando se configura CDP en un enrutador y en una interfaz, CDP irá enviando actualizaciones por la interfaz cada cierto intervalo de tiempo, en estas actualizaciones se envía una cantidad de información considerable, entre otros datos, se publican los siguientes:

- Nombre del dispositivo adyacente.
- Direcciones IP.
- Plataforma.
- Tipo de dispositivo.
- Interfaz por la que se llega al dispositivo adyacente.
- Interfaz del dispositivo adyacente a la que está conectado el dispositivo.
- Versión del sistema operativo.
- Versión de CDP.

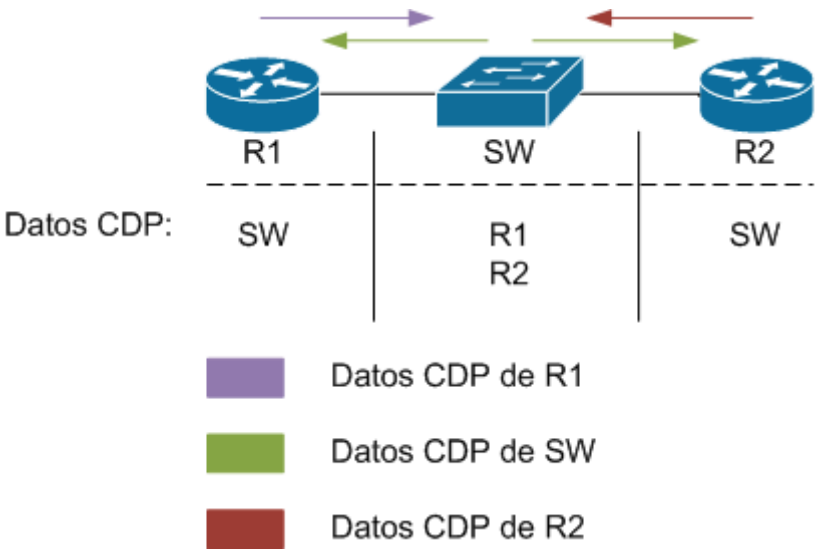
Debido a que CDP se encapsula directamente en la capa 2 del modelo OSI, la información proporcionada por CDP es útil para diagnosticar problemas de conectividad de capa 3 ya que aunque no haya ninguna dirección IP configurada los paquetes son transferidos por la red.

El protocolo CDP utiliza la dirección multicast de capa 2 "01-00-0c-cc-cc-cc" y el número de protocolo que usa en redes con encapsulación Ethernet y serial es "0x2000", en redes con encapsulación PPP usa el número de protocolo "0x0207". El OUI (código de organización) de CDP es "0x00000c" que indica que es de Cisco.

Al ser un protocolo que se encapsula en la capa 2 del modelo OSI, se ejecuta en cualquier dispositivo que actúe en la capa 2 o superior, es decir, entre otros se ejecuta en conmutadores, enrutadores, cortafuegos, puntos de acceso Wifi, ...

Existen dos versiones, ambas utilizan el mismo sistema de transmisión, la misma dirección de multicast, lo único que diferencia ambas versiones, es que en la versión 2 hay más mensajes de error para diagnosticar posibles errores de configuración como podría ser diferente configuración de dúplex.

Hay que tener en cuenta que CDP se transmite entre dispositivos adyacentes, es decir, directamente conectados, en caso de que haya dos enrutadores conectados a un switch (todos Cisco), los datos CDP de cada router quedarán registrados en el switch y en cada router habrá los datos del switch, pero cada enrutador no tendrá datos del otro enrutador:



Configuración

El protocolo CDP se configura primeramente en el modo de configuración global y posteriormente por interfaz. Aquí se configurará CDP versión 2 en un switch y dos router conectados al switch, como en el esquema anterior.

Los comandos de configuración global que hay que introducir en el dispositivo son los siguientes:

```
Switch(config)#cdp                          advertise-v2
Switch(config)#cdp                 timer              60
Switch(config)#cdp               holdtime            180
Switch(config)#cdp run
```

Con estos comandos se establece la siguiente configuración:

- **cdp advertise-v2:** publicará la información de la versión 2 de CDP.
- **cdp timer 60:** enviará un paquete CDP para 60 segundos.
- **cdp holdtime 180:** guardará una entrada en la tabla de vecinos durante 180 segundos.
- **cdp run:** se ejecuta CDP en el dispositivo.

A nivel de interfaz, el comando que hay que introducir en la interfaz donde queremos que se envíen paquetes CDP es:

```
Switch(config-if)#cdp enable
```

Y en caso de que queramos deshabilitar el envío de paquetes CDP en una interfaz, hay que introducir el modificador "no" delante del comando:

```
Switch(config-if)#no cdp enable
```

Por defecto, una vez que se configura CDP en el modo de configuración global, se envían paquetes CDP por todas las interfaces, por lo que en un primer momento, tras

activar CDP (comando "cdp run") solamente es necesario desactivar CDP en las interfaces por las que no queramos que se envíe información.

La configuración es la misma en cualquier dispositivo.

Verificación

El protocolo CDP es útil para resolver problemas de conectividad en la capa 2 y capa 3. En caso de que todo esté correctamente configurado, no mostrará ningún error, en caso contrario irá mostrando avisos de los fallos que detecte.

Los comandos para ver la información de CDP son:

- **show cdp entry <nombre | *>:** muestra toda la información del dispositivo indicado (nombre) o de todos los dispositivos (*), muestra la misma cantidad de información que el comando "**show cdp neighbors detail**".
- **show cdp interface:** muestra la encapsulación y los parámetros CDP de una interfaz si se especifica una o bien de todas las interfaces si no se especifica ninguna.
- **show cdp neighbors [<interfaz>] [detail]:** muestra los dispositivos adyacentes conectados directamente, en caso de que se especifique una interfaz, solamente muestra los dispositivos conectados directamente a esa interfaz y si se añade el modificador "detail" muestra toda la información de todos los dispositivos vecinos conectados o bien si se especifica una interfaz, muestra con detalle los datos de los dispositivos conectados a esa interfaz.
- **show cdp traffic:** muestra estadísticas de CDP.
- **show cdp:** muestra la versión CDP de los paquetes enviados, el intervalo entre paquetes y el tiempo que se guardará la información recibida en memoria.

Los comandos de depuración disponibles para CDP son los siguientes:

- **"debug cdp packets":** da información de los paquetes recibidos de cada vecino.
- **"debug cdp adjacency":** muestra información de las adyacencias ya establecidas, es decir, no muestra cuando se forma una adyacencia, sino cuando se rompe básicamente.
- **"debug cdp events":** muestra información de los cambios que se dan relacionados con CDP.
- **"debug cdp ip":** este comando es útil básicamente cuando se configura el protocolo de enrutamiento ODR (enrutamiento bajo demanda) ya que ése protocolo usa CDP para descubrir las redes que cada enrutador tiene conectadas, las guarda e inserta una ruta estática en los routers conectados, es la única función que CDP efectúa en la capa de red.

Ahora se cambiará la interfaz fastethernet del switch a half dúplex y se forzará que la interfaz ethernet del enrutador R2610 esté en modo full-dúplex, lo que habría de provocar mensajes de alerta de CDP por error de configuración:

"show cdp entry R2610":

"show cdp interface":

"show cdp neighbors":

"show cdp traffic":

"show cdp":

Los comandos de depuración disponibles para CDP son los siguientes:

- **"debug cdp packets":** da información de los paquetes recibidos de cada vecino.
- **"debug cdp adjacency":** muestra información de las adyacencias ya establecidas, es decir, no muestra cuando se forma una adyacencia, sino cuando se rompe básicamente.
- **"debug cdp events":** muestra información de los cambios que se dan relacionados con CDP.
- **"debug cdp ip":** este comando es útil básicamente cuando se configura el protocolo de enrutamiento ODR (enrutamiento bajo demanda) ya que ése protocolo usa CDP para descubrir las redes que cada enrutador tiene conectadas, las guarda e inserta una ruta estática en los routers conectados, es la única función que CDP efectúa en la capa de red.

Ahora se cambiará la interfaz fastethernet del switch a half dúplex y se forzará que la interfaz ethernet del enrutador R2610 esté en modo full-dúplex, lo que habría de provocar mensajes de alerta de CDP por error de configuración:

```
Switch#show cdp entry R2610

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

Device ID: R2610

Entry address(es):

  IP address: 192.168.0.2

Platform: cisco 2610,  Capabilities: Router

Interface: FastEthernet0/17,  Port ID (outgoing port): Ethernet0/0

Holdtime : 142 sec

Version :

Cisco Internetwork Operating System Software

IOS (tm) C2600 Software (C2600-IX-M), Version 12.2(8)T4,  RELEASE SOFTWARE (fc1)

TAC Support: http://www.cisco.com/tac

Copyright (c) 1986-2002 by cisco Systems, Inc.

Compiled Sat 04-May-02 08:06 by ccai

advertisement version: 2

Duplex: full (Mismatch)
```

Management address(es):

Se puede ver que casi al final de todo indica: "**Duplex: full (Mismatch)**". Esto significa que no coinciden los modos de transmisión en ambos dispositivos y provoca que periódicamente se muestre la siguiente notificación:

May 18 20:50:31: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/17 (not full duplex), with R2610 Ethernet0/0 (full duplex).

May 18 20:51:32: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/17 (not full duplex), with R2610 Ethernet0/0 (full duplex).

May 18 20:52:32: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/17 (not full duplex), with R2610 Ethernet0/0 (full duplex).

May 18 20:53:32: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/17 (not full duplex), with R2610 Ethernet0/0 (full duplex).

7. N.T.P. – Network Time Protocol

Introducción

El protocolo NTP sirve para sincronizar el tiempo entre aparatos a través de una red IP utilizando el protocolo de transporte UDP puerto 123.

Se divide en jerarquías que van del 0 (superior) al 3 (inferior) aunque dependiendo de la versión de NTP puede ir hasta el 255.

Cada nivel se llama "stratum" pero para hacer el texto más sencillo, lo seguiré denominando nivel.

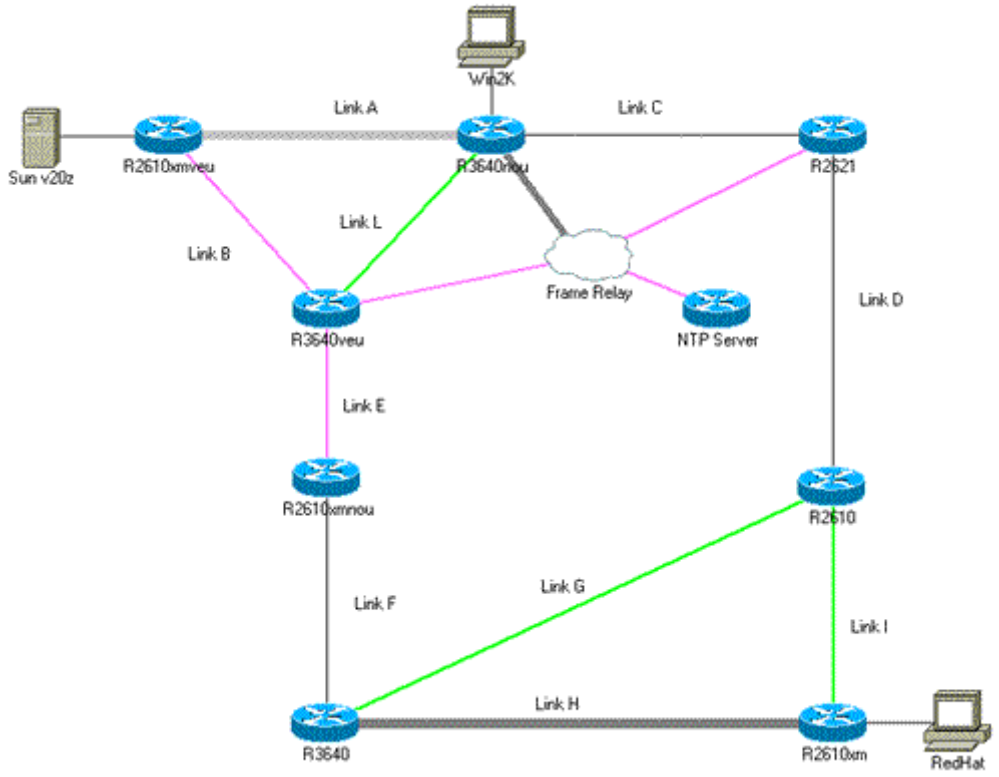
- Nivel 0: se encuentran las fuentes de tiempo, como relojes atómicos, receptores GPS, ... Van conectadas directamente a un servidor del nivel 1.
- Nivel 1: se encuentran servidores conectados directamente a una fuente de tiempo y hacen de servidor para los servidores de nivel 2.
- Nivel 2: los servidores de nivel 2, normalmente toman como fuente de datos a varios servidores de nivel 1 y el algoritmo de NTP se encarga de determinar la mejor fuente y tomarla de referencia, por lo que es posible que un servidor de nivel 2 tenga la hora más precisa que uno de nivel 1. Suelen servir directamente al dispositivo cliente.
- Nivel 3: Aquí entra cualquier dispositivo que tome la hora de una fuente de datos en la red como pueden ser PC, routers, ...

Aplicación

El protocolo NTP se usa básicamente para sincronizar dispositivos con una única referencia horaria.

En el caso de mi laboratorio, la aplicación que tiene el sincronizar todos los dispositivos es en el caso de monitorizar procesos como los protocolos de enrutamiento, poder ver qué sucede en un mismo momento en dos dispositivos adyacentes o distantes.

Topología de laboratorio



En este caso, para que la hora siempre esté sincronizada independientemente del protocolo de enrutamiento que se use, para comunicarse entre ellos, hay configuradas unas rutas estáticas especificando interfaz de salida y con distancia administrativa de 254 para que si en algún momento se ejecuta un protocolo de enrutamiento que publique todas las direcciones IP de un router -como ODR-, no haya lugar a confusiones.

El servidor NTP se conecta a internet y tiene configurados dos servidores, el resto de routers consultan al servidor NTP.

Comandos de configuración de cada dispositivo:

```
NTP Server

R2610veu

R3640nou

R2621

R3640veu

R2610xmnou

R2610

R3640

R2610xm
```

Analizando las configuraciones del router que hace de servidor NTP y de un cliente para ver la diferencia, por ejemplo el R3640:

Servidor NTP:

`ntp broadcastdelay 170000`	Este comando especifica el retraso que hay del router al servidor de destino.
`ntp clock-period 17179778`	Es un parámetro referente al reloj interno del router y que se va ajustando.
`ntp master 3`	Indica que es del nivel 3 (stratum 3)
`ntp server 128.59.59.177`	Servidor de sincronización.
`ntp server 198.82.1.204 prefer`	Servidor de sincronización preferido.

R3640:

El comando que se diferencia del servidor es el siguiente:

`ntp server 10.0.0.18 source Loopback123 prefer`	Este comando indica que la petición al servidor se ha de hacer desde la interfaz Loopback 123 (y con la dirección IP de dicha interfaz).

Esto es debido a que como comentaba antes, he usado un direccionamiento específico (2.2.2.x/32) solo para las peticiones NTP y en cada router hay una interfaz lógica Loopback con número 123 (el puerto de NTP) con IP 2.2.2.x donde x es un número que tiene cada router y las rutas estáticas apuntan hacia cada router por un camino.

Aparte de estos comandos también es interesante configurar la zona horaria para que así los routers cambien la hora automáticamente, para Barcelona se haría con los siguientes comandos:

```
(config)#clock timezone CET 1
(config)#clock summer-time CEST recurring last Sun Mar 2:00 last Sun Oct 3:00
```

Verificación

Para ver las conexiones a servidores NTP, se ejecuta el comando: `show ntp associations`

```
R3640#sh ntp associations

address ref clock st when poll reach delay offset disp
*~10.0.0.18 198.82.1.204 3 95 64 206 64.6 2889.0 2022.2
* master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

Para ver más información de NTP en el router: `show ntp status`

```
R3640#show ntp status
Clock is unsynchronized, stratum 16, no reference clock
nominal freq is 250.0000 Hz, actual freq is 250.8695 Hz, precision is 2**24
reference time is CF15E533.6D54FB32 (01:30:43.427 CET Fri Feb 5 2010)
clock offset is 222.1252 msec, root delay is 239.56 msec
root dispersion is 16118.64 msec, peer dispersion is 16000.00 msec
```

Para ver información sobre el proceso NTP, aquí hay un par de comandos de depuración:

Tráfico de paquetes: `debug ntp packet`
Ajuste del periodo del reloj: `debug ntp adjust`

8 . N.A.T. – Network Address Translation

Inicialmente los dispositivos tenían poca capacidad y los componentes eran caros, lo cual hacía que cuanto más ajustado fuera la cantidad de datos a tratar y más eficientemente tratados fueran estos, menor coste tendría. Cuando se diseñó el protocolo IP, tal vez por este motivo, se estableció que las direcciones IP tuvieran una longitud de 32 bits (según RFC 760 y 791). Con 32 bits la cantidad máxima de direcciones IP es igual a 2^32 direcciones, es decir, algo más de 4096 millones de direcciones IP distintas, descontando los rangos D y E de direcciones no asignables a host, la cantidad de direcciones disminuye sustancialmente, por lo que la cantidad final de direcciones asignables es bastante más limitada y eso provoca que tengan un coste bastante alto.

En terminología NAT, hay dos tipos de direcciones IP:

- Privada: direcciones IP que no se enrutan, generalmente se usan las que están definidas en el RFC 1918, y al ser rangos privados pueden usarse como se quiera.
- Pública: direcciones IP enrutables, asignadas por IANA (organismo encargado de asignar las direcciones IP) y que tienen un coste.

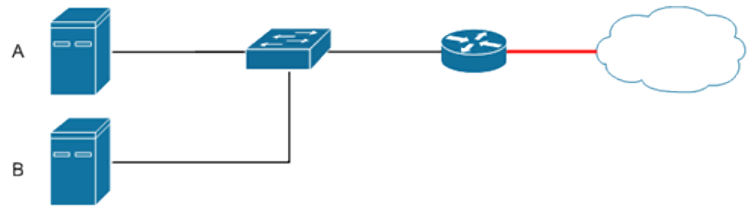
Esto explica el modo en que NAT ayuda a ampliar las direcciones disponibles, ya que puede haber hasta 16581375 dispositivos con direcciones IP distintas con acceso a internet con una sola dirección IP pública, en la práctica no sería posible que hubiese tantas direcciones IP, pero el NAT en teoría lo permitiría.

Hay varios tipos de NAT:

- NAT estático: una dirección IP privada tiene asignada una IP pública fija, que siempre será la misma.
- NAT dinámico: muchas direcciones IP privadas pueden tomar una dirección IP pública, cuando un host con IP privada quiere salir a internet, tomará una dirección IP pública si hay disponible, en caso de que no la haya habrá que esperar hasta que una asignación caduque. Las asignaciones tienen un temporizador tras el cual si no se está utilizando se elimina y la dirección IP pública asignada queda libre para otro host privado que quiera usarla.
- NAT con sobrecarga: en este caso una o muchas direcciones IP privadas tienen asignada una sola dirección IP pública, cuando una dirección IP privada quiere salir a internet, en lugar de tener una IP pública solo para ese dispositivo, se le asigna un puerto público por cada petición de puerto privado que hay. Si un host tiene dos aplicaciones que quieren salir a internet, NAT le asigna dos puertos públicos, uno para cada puerto privado. Cuando caduca el temporizador el puerto público queda disponible para otra petición de puerto de una IP privada, es posible hacer que desde el exterior se pueda acceder al puerto privado pero esta opción ha de ser configurada explícitamente.
- PAT: es una asignación estática de puertos públicos a puertos y IP privadas. Se diferencia entre los protocolos TCP y UDP.

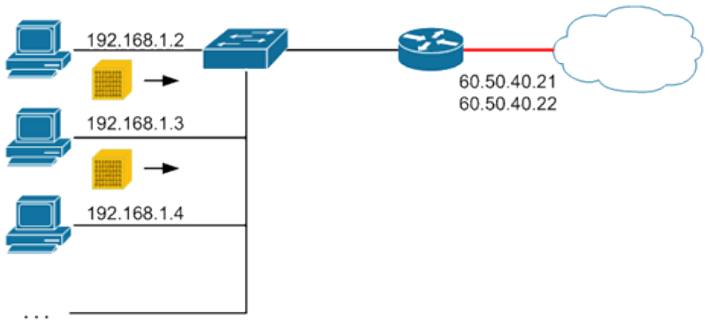
Aquí se representa con un gráfico cada tipo:

N.A.T. estático:



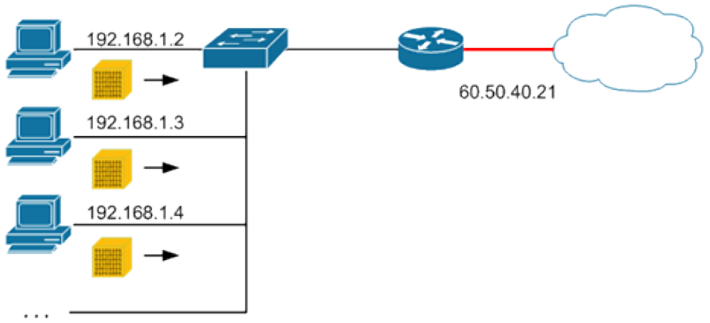
Host	IP privada	IP pública
A	192.168.0.10	60.50.40.31
B	192.168.0.11	60.50.40.32

N.A.T. dinámico:



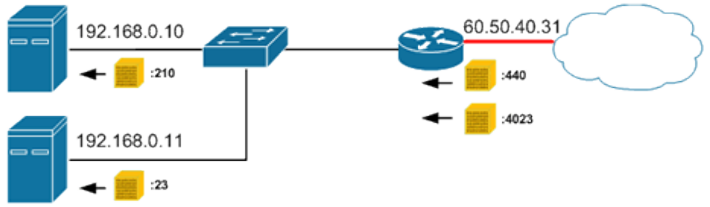
IP privada	IP pública
192.168.1.2	60.50.40.21
192.168.1.3	60.50.40.22

N.A.T. con sobrecarga:



Red privada	IP pública
192.168.1.2:3328	60.50.40.21:1500
192.168.1.3:3421	60.50.40.21:1503
192.168.1.3:1544	60.50.40.21:1506
192.168.1.4:3328	60.50.40.21:1507
192.168.1.4:2228	60.50.40.21:1509

Traducción de puertos P.A.T.:



IP privada	IP pública
192.168.0.10:80	60.50.40.31:80
192.168.0.10:110	60.50.40.31:110
192.168.0.10:210	60.50.40.31:440
192.168.0.11:21	60.50.40.31:21
192.168.0.11:20	60.50.40.31:20
192.168.0.11:23	60.50.40.31:4023

Como se puede ver en el gráfico, el NAT estático es asignar a cada host de la red interna tuviese una dirección IP pública. El NAT dinámico es como si cada host interno tuviese una dirección IP pública aunque de forma temporal. En el NAT con sobrecarga todos los hosts tienen una dirección IP pública pero cada host tiene unos puertos abiertos según establece una conexión hacia el exterior aunque temporalmente. En el PAT o traducción de puertos, cada puerto externo está asignado permanentemente a un host y puerto interno para poder ser alcanzado desde el exterior permanentemente.

Los cuatro tipos de NAT pueden ser combinados en un mismo enrutador.

En algunos lugares he visto la siguiente subclasificación de los tipos de NAT:

- SNAT: del inglés "*Source NAT*", es cuando se traduce la dirección o el puerto de origen.
- DNAT: del inglés "*Destination NAT*", es cuando se traduce la dirección o el puerto de destino.

Según esta clasificación, los cuatro tipos de NAT descritos se clasificarían como:

- SNAT: NAT dinámico, NAT con sobrecarga.
- DNAT: NAT estático, PAT.

N.A.T. estático

Como se ha dicho en el punto anterior, el NAT estático es simplemente asignar una dirección IP pública fija a una dirección IP privada también fija.

Este tipo de NAT es adecuado sobre todo para servidores que se encuentren en una red privada y tengan que ser plenamente accesibles desde el exterior, digo plenamente porque si solo es necesario acceder a un puerto en cada uno, la alternativa de la traducción de puertos puede ser viable como se verá en el apartado correspondiente.

En el NAT estático, si se quiere bloquear el acceso a determinados puertos desde el exterior, se puede hacer con una ACL extendida.

Sintaxis:

```
Router(config)#ip nat inside source static <IP int.> <IP ext.>
Router(config)#interface <interfaz interna>
Router(config-if)#ip nat inside
Router(config)#interface <interfaz externa>
Router(config-if)#ip nat outside
```

Los datos que es necesario saber para configurar el NAT estático son:

- Dirección IP pública.
- Dirección IP privada.
- Interfaz conectada a la red privada.
- Interfaz conectada a la red pública.

Sabiendo estos dos datos el NAT estático tiene una fácil configuración:

Router#configure terminal

Router(config)#ip nat inside source static 192.168.101.253 10.0.0.5

Router(config)#int fastethernet 0/0

Router(config-if)#ip nat inside

Router(config-if)#int serial 0/0

Router(config-if)#ip nat outside

Hecho esto ya se puede verificar si la traducción está bien configurada viendo las traducciones, en este caso se hace un ping del host 192.168.101.253 a la IP externa 172.16.0.1:

R2610xmveu#sh ip nat tr

Pro Inside global Inside local Outside local Outside global

icmp 10.0.0.5:512 192.168.101.253:512 172.16.0.1:512 172.16.0.1:512

--- 10.0.0.5 192.168.101.253 --- ---

Para depurar el proceso NAT, se puede ejecutar el comando siguiente:

1> NAT*: i: icmp (192.168.101.253, 512) -> (172.16.0.1, 512) [3563]

2> NAT*: s=192.168.101.253->10.0.0.5, d=172.16.0.1 [3563]

3> NAT*: o: icmp (172.16.0.1, 512) -> (10.0.0.5, 512) [3563]

4> NAT*: s=172.16.0.1, d=10.0.0.5->192.168.101.253 [3563]

Esto es solamente un ping, el paquete enviado y la respuesta, lo que muestra cada línea es:

1. Paquete originario de 192.168.101.253, en la parte interna o red privada (i), con destino a la dirección 172.16.0.1. Número de secuencia 3563.
2. Origen cambiado de 192.168.101.253 a 10.0.0.5, destino 172.16.0.1.
3. Paquete originario de 172.16.0.1, en la parte externa o red pública (o), con destino a la dirección 10.0.0.5. Nº. de secuencia 3563.
4. Origen 172.16.0.1. Destino cambiado de 10.0.0.5 a 192.168.101.253.

El NAT provoca que se añadan las líneas 2 y 4, donde el enrutador de manera transparente cambia las direcciones de origen y destino según los paquetes sean salientes o entrantes respectivamente.

N.A.T. dinámico

El NAT dinámico consiste en que una dirección IP de la red privada (interna) tenga asignada una dirección IP de la red pública (externa). Si hay más

Los datos necesarios para configurar este tipo de NAT son:

- Red interna que podrá salir con el NAT.
- Rango de direcciones externas asignables (una o varias).
- Interfaz conectada a la red interna.
- Interfaz conectada a la red externa.

Sintaxis:

```
Router(config)#access-list <nº ACL> permit <red interna>
<wildcard>
Router(config)#ip nat pool <nombre> <1ª IP> <ult. IP> <netmask
<máscara>|prefix-length <bits de máscara>> [<add-route|type
<rotary|match-host>>]
Router(config)#ip nat inside source list <nº ACL> pool <nombre>
<reversible>
Router(config)#interface <int. interna>
Router(config-if)#ip nat inside
Router(config)#interface <int. externa>
Router(config-if)#ip nat outside
```

Descripción de los parámetros:

- **nº ACL**: número de ACL, puede ser cualquier número dentro de los rangos de ACL simples (1~99, 1300~1999).
- **red interna**: dirección IP de red interna.
- **wildcard**: wildcard (inversa de la máscara o los bits de red a 0 y los de host a 1) de la red interna.
- **nombre**: nombre identificativo del rango de direcciones externas.
- **1ª IP**: dirección IP menor del rango externo.
- **ult. IP**: dirección IP mayor del rango externo.
- **máscara**: máscara de las direcciones IP externas.
- **bits de máscara**: longitud de la máscara de direcciones IP externas.
- **rotary**: rotación de direcciones, si una misma dirección IP interna tiene asignada una dirección IP externa, expira la asignación y vuelve a tener otra asignación, con esta opción se le asignarán las direcciones de manera rotativa.
- **match-host**: en este caso, lo que se hace es que en la dirección IP externa, la parte de host coincide con la dirección IP interna. Por ejemplo, si la red interna es 192.168.1.0/24 y el rango externo de direcciones abarca toda un rango como 10.0.0.0/24, la dirección IP interna 192.168.1.34 tendrá asignada la dirección IP externa 10.0.0.34, y la dirección interna 192.168.1.67 la dirección externa 10.0.0.67.
- **add-route**: añade una ruta para publicar la entrada.
- **reversible**: indica que puede haber tráfico en sentido externo->interno.
- **int. interna**: interfaz conectada a la red interna.
- **int. externa**: interfaz conectada a la red externa.

En el ejemplo se hará que la red interna 192.168.101.0/24 tenga disponible las direcciones IP públicas 10.0.0.3 y 10.0.0.4 y las asigne de manera rotativa, simultáneamente habrá configurado el NAT estático del punto anterior.

Configuración:

1> R2610xmveu(config)#access-list 10 permit 192.168.101.0 0.0.0.255

2> R2610xmveu(config)#ip nat pool NAT-dinamico 10.0.0.3 10.0.0.4 prefix-length 29 type rotary

3> R2610xmveu(config)#ip nat inside source list 10 pool NAT-dinamico

4> % Pool NAT-dinamico is a rotary-type pool, unexpected behavior may result.

5> R2610xmveu(config)#int fa0/0

6> R2610xmveu(config-if)#ip nat inside

7> R2610xmveu(config-if)#int serial0/0

8> R2610xmveu(config-if)#ip nat outside

Descripción por líneas:

1. Se define la ACL indicando la red interna.
2. Se crea el rango de direcciones públicas asignables a las redes internas, con máscara de 29 bits y asignado de manera rotativa.
3. Se configura la traducción con los parámetros configurados en las dos líneas anteriores.
4. Se indica un error dado que como las traducciones pueden ser bidireccionales, en caso de configurarse una traducción dinámica y en la red interna hubiese un servidor accesible desde el exterior, en cuanto la asignación (IP interna-IP externa) caducase y fuese eliminada, las peticiones externas seguirían dirigiéndose a la asignación externa que tenía el servidor y los paquetes irían destinados a la nueva dirección interna que tuviese asignada la dirección externa.
5. Se cambia al modo de configuración de la interfaz interna.
6. Se establece que es la parte interna del NAT.
7. Se cambia al modo de configuración de la interfaz externa.
8. Se establece que es la parte externa del NAT.

En la parte interna hay un host con IP 192.168.101.6/24, se dejará un ping permanente hacia la dirección externa 172.16.0.1 y se irá borrando las traducciones NAT dinámicas para ver que en cuando se hace una nueva asignación la dirección pública va cambiando entre las dos disponibles:

```
R2610xmveu#show ip nat translations icmp
Pro Inside global   Inside local      Outside local    Outside global
icmp 10.0.0.3:512   192.168.101.6:512  172.16.0.1:512   172.16.0.1:512
R2610xmveu#clear ip nat translation *
R2610xmveu#show ip nat translations icmp
Pro Inside global   Inside local      Outside local    Outside global
icmp 10.0.0.4:512   192.168.101.6:512  172.16.0.1:512   172.16.0.1:512
R2610xmveu#clear ip nat translation *
R2610xmveu#show ip nat translations icmp
Pro Inside global   Inside local      Outside local    Outside global
icmp 10.0.0.3:512   192.168.101.6:512  172.16.0.1:512   172.16.0.1:512
R2610xmveu#clear ip nat translation *
R2610xmveu#show ip nat translations icmp
Pro Inside global   Inside local      Outside local    Outside global
icmp 10.0.0.4:512   192.168.101.6:512  172.16.0.1:512   172.16.0.1:512
```

Depurando el proceso NAT para la red especificada en la ACL anterior y enviando pings de un solo paquete, vemos lo siguiente:

```
R2610xmveu#clear ip nat translation *
R2610xmveu#show ip nat translations
Pro Inside global   Inside local      Outside local  Outside global
--- 10.0.0.5        192.168.101.253  ---            ---
R2610xmveu#debug ip nat 10
Apr 28 16:04:24.782: NAT*: s=192.168.101.6->10.0.0.3, d=172.16.0.1 [18719]
Apr 28 16:04:24.782: NAT*: s=172.16.0.1, d=10.0.0.3->192.168.101.6 [18719]
R2610xmveu#clear ip nat translation *
R2610xmveu#
Apr 28 16:04:31.371: NAT: free twice, pool NAT-dinamico address 10.0.0.3!
Apr 28 16:04:37.179: NAT*: s=192.168.101.6->10.0.0.4, d=172.16.0.1 [18744]
Apr 28 16:04:37.179: NAT*: s=172.16.0.1, d=10.0.0.4->192.168.101.6 [18744]
R2610xmveu#clear ip nat translation *
```

R2610xmveu#

Apr 28 16:04:41.221: NAT: free twice, pool NAT-dinamico address 10.0.0.4!

Apr 28 16:04:51.689: NAT*: s=192.168.101.6->10.0.0.3, d=172.16.0.1 [18771]

Apr 28 16:04:51.693: NAT*: s=172.16.0.1, d=10.0.0.3->192.168.101.6 [18771]

R2610xmveu#clear ip nat translation *

R2610xmveu#

Apr 28 16:04:55.719: NAT: free twice, pool NAT-dinamico address 10.0.0.3!

Se parte de que solamente hay la traducción NAT estática configurada en el apartado anterior, pero no hay ninguna traducción NAT dinámica. Cuando se hace un ping, se muestra la traducción, inicialmente la traducción es de 192.168.101.6 a 10.0.0.3, cuando se eliminan las asignaciones y se hace otro ping, la asignación pasa a ser 192.168.101.6 a 10.0.0.4, al repetir el proceso de eliminar asignaciones y hacer ping, la asignación pasa a ser de 192.168.101.6 a 10.0.0.3 otra vez y así sucesivamente, si hubiese tres direcciones IP públicas configuradas iría asignándolas de forma secuencial.

El NAT dinámico con sobrecarga es una "extensión" del NAT dinámico, se explicará en el siguiente apartado partiendo de este.

N.A.T. dinámico con sobrecarga (P.A.T.)

El NAT dinámico con sobrecarga se podría decir que es una extensión del NAT dinámico. Se podría decir que se parte de la base de que en un host dentro de una red privada generalmente solamente usa un número limitado de puertos para los diferentes servicios que en éste se estén ejecutando, por este motivo no es necesario que cada host de la red interna disponga de una dirección IP pública con los 65535 puertos disponibles sino que solo es necesario que use un puerto u otro según lo necesite.

Partiendo de esto y teniendo en cuenta que no todos los servicios que abren puertos en un host necesitan salir de la red interna ni estar constantemente abiertos, un grupo de hosts pueden funcionar perfectamente con los 65535 puertos que ofrece una sola dirección IP, y además, en caso de que con una sola dirección IP no haya suficiente, se puede asignar un rango de direcciones IP externas, cuando una IP tenga todos los puertos ocupados pasará a usarse la siguiente IP y así sucesivamente hasta completarlas todas.

En este caso también puede configurarse para que en lugar de usar un rango de direcciones IP, todos los host salgan por una interfaz tenga la dirección IP que tenga.

Sintaxis con rango de direcciones:

```
Router(config)#access-list <nº> permit <red interna> <wildcard>
Router(config)#ip nat pool <nomb.> <1ª IP> <últ. IP>
<másc.|long.>
Router(config)#ip nat inside source list <nº> pool <nomb.>
overload
```

Si se quiere que todos los hosts salgan por una interfaz:

```
Router(config)#access-list <nº> permit <red interna> <wildcard>
Router(config)#ip nat inside source list <nº> interface
<interfaz> overload
```

Parámetros:

- **nº**: número identificativo de la ACL.
- **red interna**: red interna para hacer NAT.
- **wildcard**: inverso de la máscara de la red interna.
- **nomb.**: nombre identificativo del rango de IP externo.
- **1ª IP**: dirección IP inferior del rango externo.
- **Últ. IP**: dirección IP superior del rango externo.
- **Másc.**: máscara de red del rango externo de direcciones IP.
- **Long.**: longitud en bits de la máscara.
- **Interfaz**: interfaz que se va a asignar como interfaz conectada a la red externa.

Ejemplo:

Se configurará NAT para que la red 192.168.101.0/24 tenga NAT dinámico con sobrecarga con la dirección IP pública 10.0.0.4:

1> Router(config)#access-list 10 permit 192.168.101.0 0.0.0.255

2> Router(config)#ip nat pool nat-dinamico-sobrecarga 10.0.0.4 10.0.0.4 netmask 255.255.255.248

3> Router(config)#ip nat inside source list 10 pool nat-dinamico-sobrecarga overload

4> Router(config)#interface fastethernet 0/0

5> Router(config-if)#ip nat inside

6> Router(config-if)#interface serial 0/0

7> Router(config-if)#ip nat outside

En la configuración de NAT dinámico con sobrecarga en relación con el NAT dinámico el único cambio que hay es que en la declaración de la traducción (línea 3) se añade el parámetro "*overload*".

Desde dos hosts de la parte interna se hará ping y telnet a un enrutador situado en la parte externa:

```
R2610xmveu#show ip nat translations

Pro Inside global   Inside local       Outside local    Outside global

icmp 10.0.0.4:513   192.168.101.6:512  172.16.0.1:512   172.16.0.1:513

tcp 10.0.0.4:1039   192.168.101.6:1039 172.16.0.1:23    172.16.0.1:23

tcp 10.0.0.4:1041   192.168.101.6:1041 172.16.0.1:80    172.16.0.1:80

icmp 10.0.0.4:512   192.168.101.10:512 172.16.0.2:512   172.16.0.2:512

tcp 10.0.0.4:1029   192.168.101.10:1029 172.16.0.1:23   172.16.0.1:23

tcp 10.0.0.4:1031   192.168.101.10:1031 172.16.0.1:80   172.16.0.1:80
```

Depuración de esta configuración NAT:

```
1>  NAT:Reserved 10.0.0.4 for PAT

2>  ipnat_add_dynamic_cfg_common: id 1, flag 5, range 1

3>  id 1, flags 0, domain 0, lookup 0, aclnum A, aclname 10, mapname  idb
0x00000000

4>  poolstart 10.0.0.4   poolend 10.0.0.4

5>  NAT:[0] Allocated Port for 192.168.101.6 -> 10.0.0.4: wanted 1046 got 1046

6>  NAT:[0] Allocated Port for 192.168.101.6 -> 10.0.0.4: wanted 1047 got 1047

3>  NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1032 got 1032

8>  NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1033 got 1033

9>  NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1034 got 1034

10> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1035 got 1035

11> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1036 got 1036

12> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1037 got 1037

13> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1038 got 1038

14> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1039 got 1039

15> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1040 got 1040

16> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1041 got 1041

17> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1042 got 1042
```

18> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1043 got 1043

19> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1044 got 1044

20> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1045 got 1045

21> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1046 got 1024

22> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1047 got 1047

23> NAT:[0] Allocated Port for 192.168.101.10 -> 10.0.0.4: wanted 1048 got 1048

24> NAT:[0] Allocated Port for 192.168.101.6 -> 10.0.0.4: wanted 1048 got 1025

En la línea 1 se indica que se ha reservado la dirección 10.0.0.4 para la traducción de puertos. Las líneas 2 y 3 son parámetros del proceso NAT, la línea 4 indica que el rango de direcciones usables es de 10.0.0.4 a 10.0.0.4. En las líneas 5 y 6 se han establecido dos conexiones, una Telnet y otra HTTP hacia el host 172.16.0.1 desde 192.168.101.6, posteriormente, mientras estas conexiones están abiertas pero inactivas, se abren conexiones HTTP y Telnet desde 192.168.101.10 hasta que el puerto del host de origen 192.168.101.10 coincida con un puerto abierto por el otro host (192.168.101.6), esto sucede en la línea 21 y 24, donde al coincidir el puerto origen de ambos hosts, el proceso NAT asigna un puerto distinto (y posteriormente a los paquetes se les modificará la cabecera cuando vayan destinados a ese puerto).

Traducción estática de puertos (P.A.T.)

La traducción estática de puertos, otro método de PAT, consiste en abrir un puerto de la parte externa del NAT y asignarlo a un host y un puerto interno.

La principal diferencia con NAT es que un puerto externo siempre estará destinado al mismo host y puerto interno hasta que se cambie la configuración.

Sintaxis:

```
Router(config)#ip nat inside source static <tcp|udp> <IP int.>
<p. int.> <IP ext.|interfaz> <p. ext.> [<route-map>]
```

Parámetros:

- **tcp**: si el puerto al que se va a aplicar NAT utilizará el protocolo TCP.
- **udp**: si el puerto al que se va a aplicar NAT utilizará el protocolo UDP.
- **IP int.**: dirección IP del host en la parte interna.
- **p. int.**: nº. de puerto del host interno.
- **IP ext.**: dirección IP externa.
- **interfaz**: si se quiere que sea cual sea la dirección IP externa el puerto se abra en una interfaz determinada, se especificaría este parámetro, esto es útil si la interfaz externa es única o de un tipo determinado, o en caso de que la IP externa sea variable por usar DHCP.
- **p. ext.**: nº. de puerto en la interfaz o dirección IP externa.
- **route-map**: si se quiere aplicar un mapa de rutas para filtrar o modificar parámetros, se especificaría aquí el route-map para este fin.

En este caso se configurarán las traducciones PAT siguientes:

- UDP/10.0.0.5:69 -> UDP/192.168.101.6:6969
- TCP/serial0/0:23 -> TCP/192.168.101.6:23

La interfaz serial0/0 tiene la dirección IP 10.0.0.1.

Configuración:

Router(config)#ip nat inside source static udp 192.168.101.6 6969 10.0.0.5 69

Router(config)#ip nat inside source static tcp 192.168.101.6 23 interface serial0/0 23

Router(config)#int fastethernet 0/0

Router(config-if)#ip nat inside

Router(config-if)#int serial 0/0

Router(config-if)#ip nat outside

Primero se muestran las traducciones sin sesiones abiertas, después se abre una conexión Telnet desde un host externo al puerto 23 de la dirección IP 10.0.0.1 para ver una traducción activa y posteriormente se abre una sesión TFTP (puerto 69), a continuación se muestran las traducciones:

```
Router#sh ip nat translations
Pro Inside global     Inside local      Outside local     Outside global
tcp 10.0.0.1:23       192.168.101.6:23  ---               ---
udp 10.0.0.5:69       192.168.101.6:6969 ---              ---

Router#sh ip nat translations
Pro Inside global   Inside local      Outside local     Outside global
tcp 10.0.0.1:23     192.168.101.6:23  172.16.0.1:11000  172.16.0.1:11000
tcp 10.0.0.1:23     192.168.101.6:23  ---               ---
udp 10.0.0.5:69     192.168.101.6:6969 ---              ---

Router#sh ip nat translations
Pro Inside global     Inside local      Outside local     Outside global
tcp 10.0.0.1:23       192.168.101.6:23  ---               ---
udp 10.0.0.5:69       192.168.101.6:6969 172.16.0.1:56601 172.16.0.1:56601
udp 10.0.0.5:69       192.168.101.6:6969 172.16.0.1:59053 172.16.0.1:59053
udp 10.0.0.5:69       192.168.101.6:6969 ---              ---
```

Hay otro método de configurar traducciones NAT para protocolos concretos usando puertos no estándar. La sintaxis es:

```
Router(config)#ip nat service <protocolo> <tcp|udp> port
<puerto>
```

Parámetros:

- **protocolo**: protocolo del que se quiere modificar el puerto.
  - H225: permitir protocolo H323-H225.
  - allow-h323-keepalive: permitir los *keepalives* para H323.
  - allow-sip-even-rtp-ports: permitir puertos pares para SIP.
  - fullrange <tcp|udp> port <puerto>: asignar todos los puertos disponibles del 1 al 65535 a un puerto local.
  - list: asignar puertos mediante una ACL de la 1 a la 2699.
  - mgcp: Media Gateway Control Protocol
  - ras: protocolo H323-RAS
  - rtsp: Real Time Streaming Protocol
  - sip: protocolo SIP.
  - skinny: protocolo skinny.

Temporizadores y limitadores N.A.T.

Las traducciones dinámicas de NAT, es decir, el NAT dinámico y el NAT sobrecargado, por el tipo de servicio que es, requiere de algún criterio para establecer cuando se puede eliminar una traducción dinámica de la memoria y dejarla disponible para otro host que quiera establecer una conexión.

Para configurar un temporizador, se hace desde el siguiente modo: **Router(config)#ip nat translation <temporizador> ...**

Los temporizadores son:

- **timeout <sg|never>**: elimina una traducción NAT dinámica después de "sg" (entre 0 y 536870) segundos o nunca si se especifica "never".
- **max-entries**: establece un máximo de asignaciones NAT, hay varios modificadores:
  - **<1-2147483647>**: número máximo de traducciones NAT.
  - **all-host <1-2147483647>**: número máximo de traducciones para cada dispositivo.
  - **all-vrf <1-2147483647>**: número máximo de traducciones para cada VRF.
  - **host <IP> <1-2147483647>**: número máximo de traducciones para el dispositivo con la IP especificada.
  - **vrf <ident.> <1-2147483647>**: número máximo de traducciones para el VRF con el identificador especificado.
  - **list <nº|nombre> <1-2147483647>**: número máximo de traducciones para los hosts de la ACL especificada.
- **dns-timeout <0-536870|never>**: tiempo máximo para las entradas NAT del protocolo DNS.
- **icmp-timeout <0-536870|never>**: tiempo máximo para las entradas NAT del protocolo ICMP.
- **port-timeout <0-536870|never>**: tiempo máximo para las traducciones de puertos TCP y UDP.

- **pptp-timeout <0-536870|never>**: tiempo máximo para las traducciones de PPTP.
- **tcp-timeout <0-536870|never>**: tiempo máximo para las traducciones TCP.
- **syn-timeout <0-536870|never>**: tiempo máximo para las entradas NAT de conexiones TCP cuando se recibe un paquete SYN sin más datos a continuación.
- **finrst-timeout <0-536870|never>**: tiempo máximo para las entradas NAT de conexiones TCP cuando se ha recibido un paquete FIN o un paquete RST.
- **routemap-entry-timeout <0-536870|never>**: tiempo máximo para una entrada creada a partir de un mapa de rutas (route-map).
- **udp-timeout <0-536870|never>**: tiempo máximo para las entradas NAT de conexiones UDP.

Cuando usar y no usar N.A.T. y P.A.T.

La traducción de direcciones IP y puertos es muy útil en cuanto se quiere que un grupo de hosts de una red privada puedan acceder a Internet usando la mínima cantidad de direcciones IP públicas, pero este servicio también tiene sus inconvenientes o situaciones en las que no debe usarse, aquí pongo una lista de situaciones en las que es interesante usar y que no se debe usar NAT.

Situaciones en las que puede ser interesante utilizar NAT:

1. La primera de todas y la más clara es cuando hay un conjunto de hosts y un reducido número de direcciones IP públicas. Puede ser interesante usar NAT dinámico o NAT dinámico con sobrecarga.

2. La segunda situación, es cuando se tiene un grupo de servidores en una red local y alguno de sus servicios ha de ser accesible desde internet sin querer usar una dirección IP para cada servidor, se puede configurar el enrutador para que cada servicio de los servidores salga por la dirección IP pública del enrutador. En este caso sería interesante usar PAT.

Situaciones en las que puede NO debe usarse NAT:

1. El principal problema de NAT, es que cambia la dirección de origen o de destino (dependiendo del tipo de NAT) en la cabecera IP, esto no es un problema cuando las aplicaciones confían en que la capa de enlace de datos, red y transporte, pero hay aplicaciones que por cuestiones de seguridad (verificar la no interceptación y/o modificación del paquete) también guardan datos como la dirección IP de origen en el contenido del paquete, en este caso el destino recibe un paquete con una dirección de origen/destino traducida, y al compararla con la almacenada en el contenido del mensaje y verificar que son distintas, descartaría el paquete y provocaría la imposibilidad de comunicación entre aplicaciones en dos servidores. En este caso debería asignarse una dirección IP pública al host que ejecuta la aplicación.

2. Otro escenario donde no debería usarse NAT, es cuando se configura una VPN, las VPN interconectan distintas ubicaciones permitiendo la movilidad, y de cara al usuario y las aplicaciones que se comuniquen por la red, las distintas ubicaciones aparecen como si se encontrasen en el mismo lugar. Para el correcto funcionamiento de una VPN el servicio NAT no debe estar activado, ya que aunque sean redes privadas las interconectadas por una VPN, la VPN no requiere más que una sola dirección IP pública, y los datos de los diferentes servicios aparecen de cara al exterior como un mismo servicio. Se podría decir que una VPN sustituye al servicio NAT cuando

solamente se quieren interconectar dos ubicaciones sin posibilidad de navegar por Internet.

Hay que tener en cuenta también que NAT es un servicio que está constantemente vigilando los datos que se reciben para ver si ha de traducirlos o no, esta vigilancia y la posterior traducción si toca consume un tiempo de CPU y la tabla de traducciones ocupa memoria RAM, por lo que si se configura NAT en enrutadores que por protocolos de enrutamiento, servicios, tráfico, ... ya están bastante cargados, el tener configurado y ejecutando el servicio NAT puede incrementar la carga y bajar el rendimiento.

9 . Protocolo de Configuración de Hosts (D.H.C.P.)

El protocolo DHCP tiene la finalidad de configurar desde cero un dispositivo conectado a una red, para hacerlo, tiene cuatro paquetes distintos donde cada uno de ellos tiene una tarea concreta para conseguir ese fin, los cuatro paquetes básicos para obtener una configuración son según su orden: DHCHDISCOVERY, DHCPOFFER, DHCPREQUEST, DHCPACK.

DHCP es un protocolo que ofrece de manera dinámica la configuración, no es necesario configurar el dispositivo, previamente hubo el protocolo BOOTP, el cual requería una configuración más compleja en el lado del servidor ya que había que crear un fichero individual para cada dispositivo con todos los parámetros, mientras que en DHCP se crea un rango de direcciones IP asignables y el servidor las va asignando secuencialmente.

El protocolo DHCP se dice que también ayuda a ahorrar direcciones IP porque un proveedor de internet (ISP) puede evaluar estadísticamente la cantidad de clientes que están conectados simultáneamente y a partir de estos datos tener más clientes que direcciones IP disponibles, ya que cuando un cliente no está conectado, su dirección IP puede ser asignada a otro usuario y así ahorrar comprar direcciones y dinero.

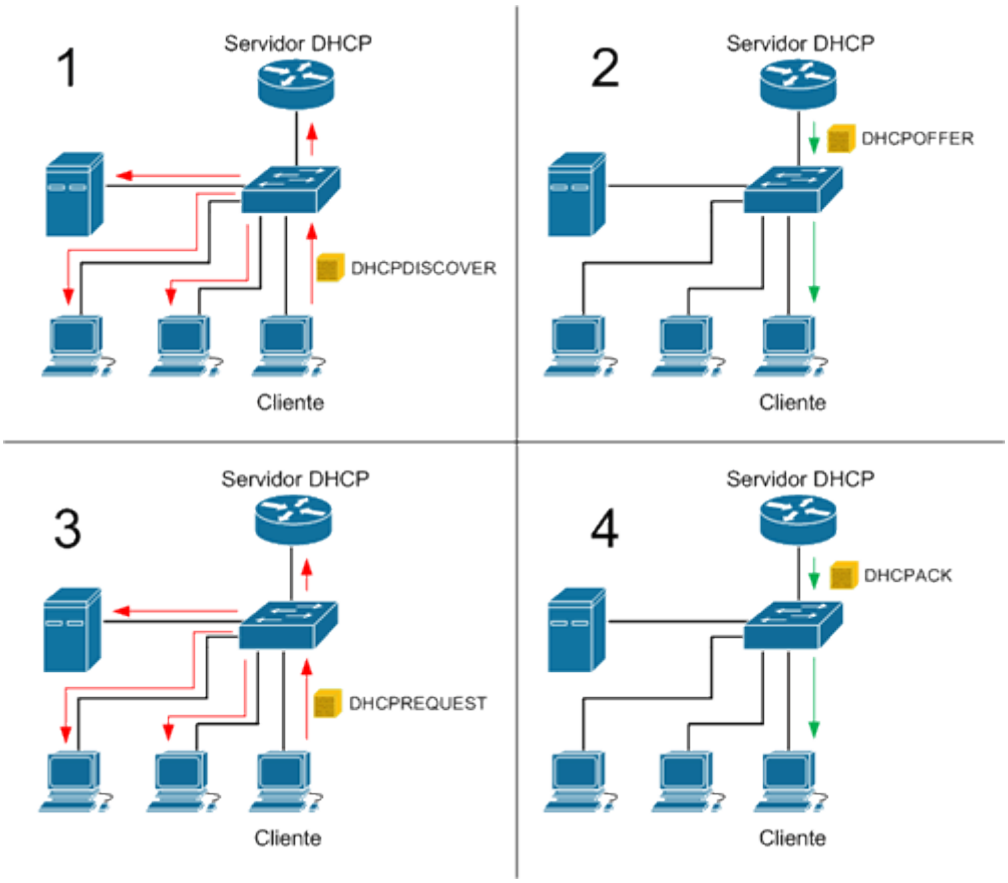
Paquetes DHCP por orden de uso:

1. DHCPDISCOVER: es el primer paquete usado en el proceso, este paquete es enviado por el cliente para "descubrir" cuáles son los servidores DHCP disponibles en la red. Este paquete tiene como dirección de destino la dirección de difusión "255.255.255.255", lleva entre otros datos la dirección física (MAC) del dispositivo que está realizando la petición. En el campo de opciones hay la opción 53 especificada con el valor 1 (valor para DHCPDISCOVER), la opción 55 indicando los parámetros necesarios para la configuración. La lista de parámetros se puede obtener aquí. En este paquete el cliente puede indicar una dirección IP que tuviese anteriormente como dirección IP deseada, el servidor puede aceptar que mantenga esa dirección o asignarle una nueva.
2. DHCPOFFER: en este paquete el servidor incluye los datos especificados en la opción 55, se envía por unicast hacia el dispositivo que ha realizado la petición. Cuando el servidor envía este paquete reserva una dirección para este dispositivo relacionando la dirección IP con la dirección MAC.
3. DHCPREQUEST: este paquete también se envía por difusión desde el cliente al servidor, indicando que recoge los datos ofrecidos.
4. DHCPACK: con este paquete el servidor le confirma al dispositivo que la asignación es para él. También se puede incluir información en este paquete como el tiempo de cesión ("*lease time*") u otra información que el cliente haya pedido.

Aparte de los cuatro paquetes principales, hay otros paquetes:

- DHCPNAK: lo envía el servidor cuando un cliente envía un paquete DHCPDISCOVER e incluye una dirección IP como la deseada, si el servidor no acepta que use esa dirección, se envía este paquete.
- DHCPDECLINE: lo envía el cliente al servidor indicando que la dirección ya está en uso.
- DHCPRELEASE: lo envía el cliente al servidor indicando que va a ser desconectado de la red y así su dirección quede disponible para ser usada por otro dispositivo.
- DHCPINFORM: lo envía el cliente al servidor pidiendo información adicional, en este momento el cliente ya tiene configurada la dirección IP.

Esquema:



En una red pequeña, el servidor DHCP tendrá una carga pequeña ya que habrá pocos dispositivos, en el caso de una red grande, que puede incluír hasta varias localizaciones, el servidor DHCP requerirá ser más potente para soportar una carga mayor, o que el servidor DHCP se encuentre en otra localización, en este caso hay la posibilidad de configurar lo que sería un asistente que reenvía los paquetes de los clientes hacia otro servidor DHCP fuera de la red de manera transparente, simplemente hace de intermediario, este tipo de servidor se llama "*dhcp relay-agent*".

También cabe la posibilidad de que en una red exista más de un servidor DHCP, en este caso el dispositivo cliente tomará y responderá a la asignación del servidor del

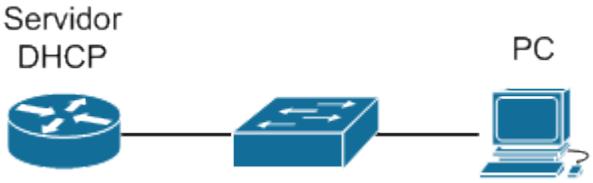
cual reciba el primer paquete. Los paquetes se diferencian por un campo que tienen los paquetes DHCP llamado XID y que es el identificador de transacción.

Las asignaciones normalmente se realizan de manera dinámica, aunque dado que las asignaciones se realizan relacionando dirección IP con dirección MAC, se pueden realizar asignaciones estáticas de manera que un dispositivo con una MAC concreta tenga siempre una dirección concreta. También puede darse el caso de que en una red se conecten dispositivos con direcciones IP configuradas manualmente que se podrían solapar con direcciones asignables por el servidor, lo que hace el servidor para prevenir esto es enviar paquetes ICMP echo (ping) para verificar que las próximas direcciones que va a asignar están libres.

El protocolo DHCP está definido en las RFC 2131, 2132, 2241, 2242, 2485, 2563, 2610, 2937, 3004, 3011, 3046, 3118, 3256, 3361, 3397, 3442, 3495, 3594, 3634, 3679, 3527, 3825, 3925, 3993, 4014, 4030, 4039, 4174, 4243,4280, 4388, 4578, 4702, 4776, 4833, 5010, 5071, 5107, 5192, 5223, 5417, 5678, 5859, 5908, 5969, 5986 y 6011, la lista de parámetros de la opción 53 se puede ver en esta URL. DHCP utiliza el protocolo de transporte UDP y los puertos 67 en los clientes y 68 en los servidores.

Configuración básica

Para realizar una configuración básica de un servidor DHCP se usa la siguiente topología:



En este caso se hará que el servidor DHCP envíe los siguientes parámetros al cliente:

- Dirección IP
- Máscara de red
- Puerta de enlace
- Validez de la concesión: 10 minutos.
- Se habilitará el registro de conflicto de direcciones.

Para la configuración necesitamos saber los siguientes datos:

- Rango de direcciones asignables: 192.168.101.0/24
- Dirección IP de la puerta de enlace: 192.168.101.3

Configuración:

1> Router(config)#ip dhcp pool DHCP-LAN

2> Router(dhcp-config)#network 192.168.101.0 255.255.255.0

3> Router(dhcp-config)#default-router 192.168.101.3

4> Router(dhcp-config)#lease 0 0 10

5> Router(dhcp-config)#exit

6> Router(config)#ip dhcp conflict logging

Explicación por líneas:

1. Se crea una configuración DHCP llamada DHCP-LAN.
2. Se establece que el rango asignable sea la red 192.168.101.0 con máscara 255.255.255.0.
3. Indica que la puerta de enlace sea 192.168.101.3 (enrutador de la red).
4. Se indica que las asignaciones tengan una validez de 0 días, 0 horas 10 minutos.
5. Se sale del modo de configuración DHCP.
6. Se activa el registro de conflictos IP.

Cuando se ha acabado de configurar el servidor DHCP y con dos hosts conectados a él, se hace que pidan una dirección IP automáticamente y el proceso de depuración indica:

1> DHCPD: Sending notification of ASSIGNMENT:

2> DHCPD: address 192.168.101.6 mask 255.255.255.0

3> DHCPD: htype 1 chaddr 18a9.0555.763e

4> DHCPD: lease time remaining (secs) = 600

5> DHCPD: checking for expired leases.

6> DHCPD: Sending notification of DISCOVER:

7> DHCPD: htype 1 chaddr 0040.9531.eb8b

DHCPD: remote id 020a0000c0a8650300000000

DHCPD: circuit id 00000000

DHCPD: Seeing if there is an internally specified pool class:

DHCPD: htype 1 chaddr 0040.9531.eb8b

DHCPD: remote id 020a0000c0a8650300000000

DHCPD: circuit id 00000000

8> DHCPD: requested address 169.254.220.83 is not on subnet 192.168.101.0.

9> DHCPD: Adding binding to radix tree (192.168.101.1)

DHCPD: Adding binding to hash tree

10> DHCPD: assigned IP address 192.168.101.1 to client 0100.4095.31eb.8b.

11> DHCPD: Sending notification of ASSIGNMENT:

12> DHCPD: address 192.168.101.1 mask 255.255.255.0

13> DHCPD: htype 1 chaddr 0040.9531.eb8b

14> DHCPD: lease time remaining (secs) = 600

DHCPD: checking for expired leases.

DHCPD: checking for expired leases.

Explicación de las líneas numeradas:

1. Indica que ha enviado un paquete DHCPOFFER.
2. La dirección asignada es 192.168.101.6 con máscara 255.255.255.0
3. Dirección MAC del cliente 18:a9:05:55:76:3e
4. Tiempo restante hasta que expire la asignación 600 segundos.
5. Comprueba si hay alguna asignación que haya caducado.
6. Recibido un paquete DHCPDISCOVER.
7. Dirección MAC del cliente 00:40:95:31:eb8b
8. Como el cliente tenía configurada una dirección APIPA, le pide al servidor seguir usándola y al estar en un rango fuera del permitido le niega usarla.
9. Añade la asignación al árbol de asignaciones. El árbol radix ("radix tree") y hash ("hash tree") son dos métodos de ordenar datos, en este caso, direcciones IP.
10. Asigna la dirección IP 192.168.101.1 al dispositivo con dirección MAC 40:95:31:eb:8b
11. Envía un paquete DHCPOFFER con los datos asignados.
12. Dirección IP asignada 192.168.101.1 con máscara 255.255.255.0
13. Dirección MAC del cliente 0040.9531.eb8b
14. Tiempo restante hasta que expire la asignación es de 600 segundos.

Y la tabla de asignaciones muestra:

```
R2610xmveu#show ip dhcp binding

Bindings from all pools not associated with VRF:

IP address       Client-ID/          Lease expiration        Type
                 Hardware address/
                 User name

192.168.101.1    0100.4095.31eb.8b   Feb 15 2010 06:56 PM    Automatic

192.168.101.6    0118.a905.5576.3e   Feb 15 2010 06:54 PM    Automatic
```

Aquí se ven los datos de las asignaciones, la primera columna es la dirección IP asignada, la segunda columna es la dirección MAC del dispositivo al cual se le ha asignado la dirección IP, la tercera columna indica cuando caduca la asignación y la cuarta columna si se trata de una asignación estática o dinámica.

4. Asignación estática de direcciones IP

Partiendo de la configuración básica anterior, se creará una asignación estática, la dirección IP 192.168.101.254 será asignada al PC con dirección MAC 18:A9:05:55:76:3E.

Configuración:

```
1> Router(config)#ip dhcp pool Servidor.253

2> Router(dhcp-config)#host 192.168.101.253 255.255.255.0

3> Router(dhcp-config)#client-identifier 0118.a905.5576.3e
```

Explicación por líneas:

1. Se crea una configuración DHCP llamada "Servidor.253"
2. Se establece la dirección IP 192.168.101.253 con máscara 255.255.255.0.
3. Se establece que la dirección IP sea asignada al dispositivo con identificador "0118.a905.5576.3e", el "01" indica el tipo de interfaz, en este caso 01.

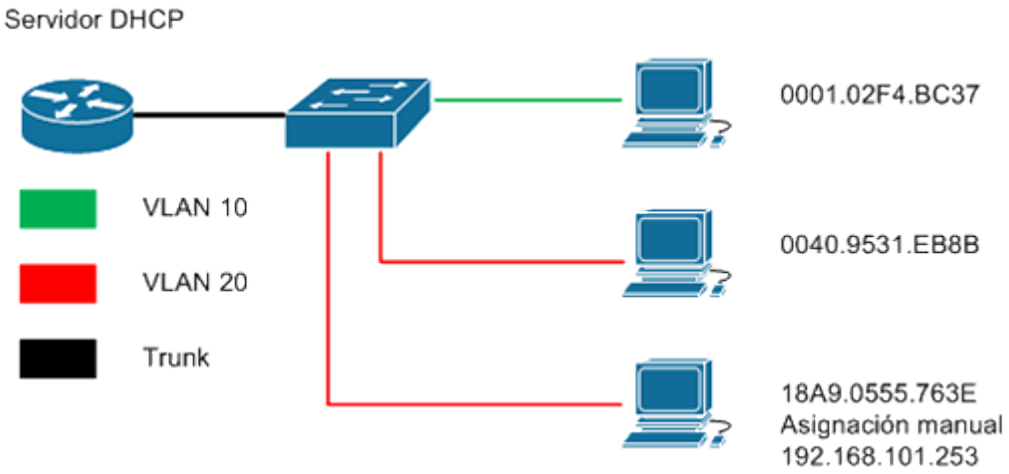
La lista de traducciones, con dos dispositivos, uno con asignación estática y otro con asignación dinámica, muestra las siguientes entradas:

```
R2610xmveu#sh ip dhcp binding

Bindings from all pools not associated with VRF:

IP address       Client-ID/          Lease expiration        Type

                 Hardware address/

                 User name

192.168.101.1    0100.4095.31eb.8b   Feb 16 2010 03:28 PM    Automatic

192.168.101.253  0118.a905.5576.3e   Infinite                Manual
```

Servidor DHCP en red con VLAN

Lo que se ha visto hasta ahora es el funcionamiento y configuración de un servidor DHCP en una red donde todos los dispositivos están interconectados, en el caso de una red dividida con VLANs, cada VLAN se comporta como una red distinta y todas ellas van conectadas a la misma interfaz del enrutador.

Topología:



Los dispositivos en la VLAN 10 tendrán configurada la red 192.168.101.0 con máscara 255.255.255.128, mientras que los dispositivos en la VLAN 20 tendrán configurada la red 192.168.101.128 con máscara 255.255.255.128. El enrutador tendrá configurada la primera IP de cada rango.

Configuración:

R2610xmveu(config)#ip dhcp pool VLAN10

R2610xmveu(dhcp-config)#network 192.168.101.0 255.255.255.128

R2610xmveu(dhcp-config)#default-router 192.168.101.1

R2610xmveu(dhcp-config)#exit

R2610xmveu(config)#ip dhcp pool VLAN20

R2610xmveu(dhcp-config)#network 192.168.101.128 255.255.255.128

R2610xmveu(dhcp-config)#default-router 192.168.101.129

R2610xmveu(dhcp-config)#exit

R2610xmveu(config)#ip dhcp pool VLAN20-Servidor.253

R2610xmveu(dhcp-config)#host 192.168.101.253 255.255.255.128

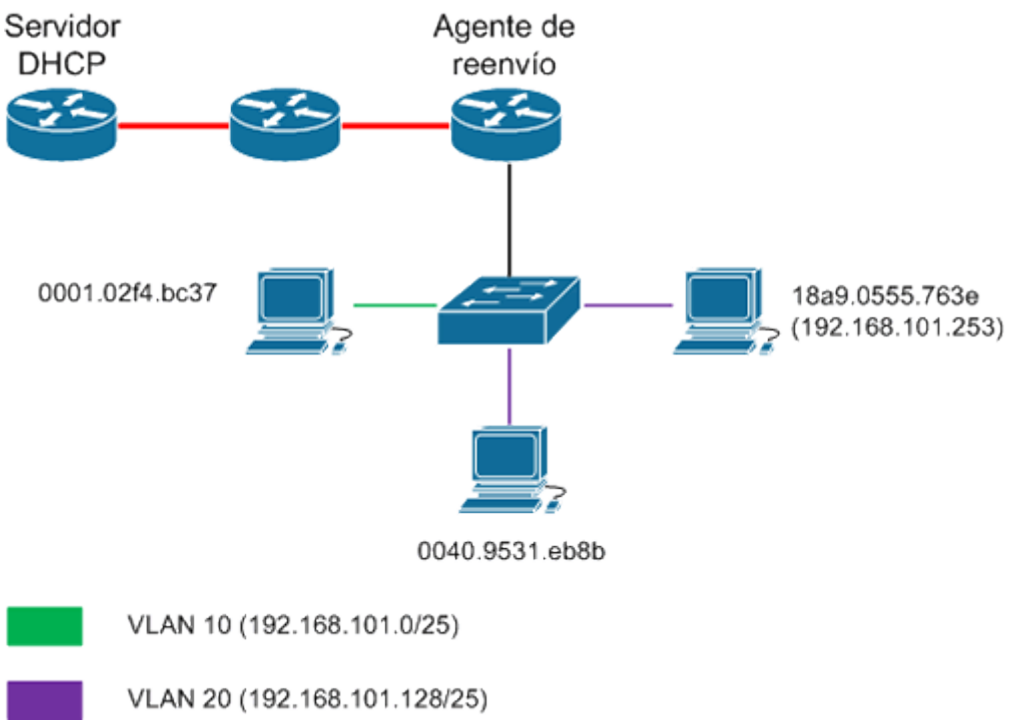
R2610xmveu(dhcp-config)#client-identifier 0118.a905.5576.3e

Reenvío de peticiones DHCP

En algunos casos el servidor DHCP encargado de asignar direcciones se encuentra centralizado para diversas redes, en estos casos se puede configurar un agente de reenvío de peticiones.

El servicio consiste en que cuando el enrutador recibe un paquete DHCPDISCOVER lo reenvía al servidor DHCP configurado, al responder el servidor, reenvía el paquete al dispositivo que estaba realizando la petición.

La topología usada será la siguiente:



La dirección IP del "Servidor DHCP" es 172.16.0.1, y la dirección IP de las interfaces del "Agente de reenvío" son 192.168.101.1 para la VLAN 10 y 192.168.101.129 para la VLAN 20.

En este caso se han eliminado las traducciones NAT ya que sino el agente de reenvío envía los paquetes al servidor DHCP con la dirección de origen interna y el servidor DHCP solo tiene conocimiento de la red externa. También es importante configurar explícitamente las direcciones IP que no se podrán asignar para evitar que se solapen.

Configuración del Servidor DHCP:

1> RouterDHCP(config)#ip dhcp excluded-address 192.168.101.1

2> RouterDHCP(config)#ip dhcp excluded-address 192.168.101.129

3> RouterDHCP(config)#ip dhcp pool VLAN-10

4> RouterDHCP(dhcp-config)#network 192.168.101.0 255.255.255.128

5> RouterDHCP(dhcp-config)#default-router 192.168.101.1

6> RouterDHCP(dhcp-config)#lease 0 0 5

7> RouterDHCP(dhcp-config)#exit

8> RouterDHCP(config)#ip dhcp pool VLAN-20

9> RouterDHCP(dhcp-config)#network 192.168.101.128 255.255.255.128

10> RouterDHCP(dhcp-config)#default-router 192.168.101.129

11> RouterDHCP(dhcp-config)#lease 0 0 10

12> RouterDHCP(dhcp-config)#exit

13> RouterDHCP(config)#ip dhcp pool VLAN-20_Servidor.253

14> RouterDHCP(dhcp-config)#host 192.168.101.253 255.255.255.128

15> RouterDHCP(dhcp-config)#client-identifier 0118.a905.5576.3e

Las líneas 1 y 2 son para excluír las direcciones IP de la interfaz del enrutador, de la línea 3 a la línea 12 se configuran los dos rangos DHCP para cada subred y en la línea 13, 14 y 15 la asignación manual de dirección IP como en los casos anteriores.

Configuración del agente de reenvío:

1> RouterRA(config)#interface FastEthernet0/0.10

2> RouterRA(config-if)#ip helper-address 172.16.0.1

3> RouterRA(config-if)#exit

4> RouterRA(config)#interface FastEthernet0/0.20

5> RouterRA(config-if)#ip helper-address 172.16.0.1

El comando para reenviar las peticiones DHCP al servidor es el especificado en las líneas 2 y 5.

La depuración del proceso muestra como las peticiones DHCP son reenviadas al servidor DHCP (IP 172.16.0.1):

DHCPD: Sending notification of DISCOVER:

DHCPD: htype 1 chaddr 18a9.0555.763e

DHCPD: remote id 020a0000c0a8658100000014

DHCPD: circuit id 00000000

DHCPD: setting giaddr to 192.168.101.129.

1> DHCPD: BOOTREQUEST from 0118.a905.5576.3e forwarded to 172.16.0.1.

2> DHCPD: forwarding BOOTREPLY to client 18a9.0555.763e.

DHCPD: Forwarding reply on numbered intf

DHCPD: creating ARP entry (192.168.101.253, 18a9.0555.763e).

DHCPD: unicasting BOOTREPLY to client 18a9.0555.763e (192.168.101.253).

DHCPD: Finding a relay for client 0118.a905.5576.3e on interface FastEthernet0/0.20.

DHCPD: Seeing if there is an internally specified pool class:

DHCPD: htype 1 chaddr 18a9.0555.763e

DHCPD: remote id 020a0000c0a8658100000014

DHCPD: circuit id 00000000

DHCPD: there is no pool for 192.168.101.129.

DHCPD: setting giaddr to 192.168.101.129.

DHCPD: BOOTREQUEST from 0118.a905.5576.3e forwarded to 172.16.0.1.

DHCPD: forwarding BOOTREPLY to client 18a9.0555.763e.

DHCPD: Forwarding reply on numbered intf

DHCPD: creating ARP entry (192.168.101.253, 18a9.0555.763e).

DHCPD: unicasting BOOTREPLY to client 18a9.0555.763e (192.168.101.253).

En la línea 1 se indica que la petición se reenvía a la dirección "172.16.0.1", la línea 2 indica que se reenvía la respuesta al cliente con dirección MAC "18a9.0555.763e".

El protocolo DHCP usa el puerto 67 y 68 con el protocolo de la capa de transporte UDP, pero los servidores no solamente usan el protocolo UDP con el protocolo DHCP, por ejemplo, el protocolo UDP también se usa para algunas comunicaciones del sistema, como NetBIOS, y lo que hace el agente de reenvío es transmitir los paquetes UDP hacia la IP establecida.

Los números entre paréntesis es el número de puerto, como se ve en las primeras líneas hay varios paquetes con los puertos 67 y 68, que son los puertos usados por el protocolo DHCP, posteriormente hay varias líneas con los puertos 137 y 138, estos puertos son los que usa NetBIOS, y como se muestra, también se envían los paquetes a la dirección 172.16.0.1, el servidor DHCP. Si hubiese un virus u otro programa dañino que se comunicase con el protocolo UDP también sería enviado, por lo que es interesante filtrar los paquetes UDP que se reenvían.

Para hacer este filtrado hay dos métodos, indicar que los paquetes destinados a algunos puertos no sean enviados, o bien con una ACL.

A continuación se indicará que los paquetes con destino a los puertos UDP 67 y 68 sean reenviados y los destinados al puerto 137 no:

RouterRA(config)#ip forward-protocol udp 67

RouterRA(config)#ip forward-protocol udp 68

RouterRA(config)#no ip forward-protocol udp 137

El resultado es que los paquetes destinados al puerto 137 no se reenvían pero los destinados a cualquier otro puerto.

Esto se debe porque por defecto se renvían todos los paquetes UDP cuando se activa el agente de reenvío, y solo hay alguna actuación cuando se especifica que no se envíen con el comando "no ip forward-protocol ...", por lo que si hay muchos protocolos conocidos que deben filtrarse, la configuración puede ser algo pesada.

Por otro lado, con el uso de ACL se puede conseguir filtrar los paquetes de manera más genérica:

```
RouterRA(config)#ip access-list extended 100

RouterRA(config-ext-nacl)#10 permit udp any eq 67 host 255.255.255.255 eq 68

RouterRA(config-ext-nacl)#100 deny udp any any

RouterRA(config-ext-nacl)#110 permit icmp any any

RouterRA(config-ext-nacl)#120 permit ip any any

RouterRA(config-ext-nacl)#exit

RouterRA(config)#interface fastethernet 0/0.10

RouterRA(config-subif)#ip access-group 100 in

RouterRA(config-subif)#exit

RouterRA(config)#interface fastethernet 0/0.20

RouterRA(config-subif)#ip access-group 100 in

RouterRA(config-subif)#exit
```

El puerto 67 es el puerto usado por los clientes DHCP, el puerto 68 es el usado por los servidores DHCP.

En este caso, si es necesario permitir uno o más puertos solamente hay que añadir una línea por puerto o incluso abrir todo un rango de puertos con una línea configurándolas con número de secuencia menor de 100, de esta manera el agente de reenvío seguirá enviando los paquetes a la dirección IP especificada, pero los paquetes se filtrarán antes de entrar en el enrutador. Esta ACL también deja pasar cualquier paquete ICMP e IP, por lo que se podría navegar por internet, hacer telnet, ping, ... solo filtra los paquetes UDP que no sean DHCP.

Depurando los paquetes UDP con la ACL aplicada se ve que solamente los paquetes DHCP (puertos 67 y 68) se transmiten:

UDP: rcvd src=0.0.0.0(68), dst=255.255.255.255(67), length=308

UDP: sent src=192.168.101.1(67), dst=172.16.0.1(67), length=308

UDP: rcvd src=172.16.0.1(67), dst=192.168.101.1(67), length=308

UDP: sent src=192.168.101.1(67), dst=192.168.101.3(68), length=308

UDP: rcvd src=0.0.0.0(68), dst=255.255.255.255(67), length=320

UDP: sent src=192.168.101.1(67), dst=172.16.0.1(67), length=320

UDP: rcvd src=172.16.0.1(67), dst=192.168.101.1(67), length=308

UDP: sent src=192.168.101.1(67), dst=192.168.101.3(68), length=308

UDP: rcvd src=0.0.0.0(68), dst=255.255.255.255(67), length=308

UDP: sent src=192.168.101.129(67), dst=172.16.0.1(67), length=308

UDP: rcvd src=172.16.0.1(67), dst=192.168.101.129(67), length=308

UDP: sent src=192.168.101.129(67), dst=192.168.101.131(68), length=308

UDP: rcvd src=0.0.0.0(68), dst=255.255.255.255(67), length=320

UDP: sent src=192.168.101.129(67), dst=172.16.0.1(67), length=320

UDP: rcvd src=172.16.0.1(67), dst=192.168.101.129(67), length=308

UDP: sent src=192.168.101.129(67), dst=192.168.101.131(68), length=308

Y comprobando los paquetes que han sido filtrados por la ACL:

RouterRA#show ip access-lists 100

Extended IP access list 100

10 permit udp any eq bootpc host 255.255.255.255 eq bootps (12 matches)

100 deny udp any any (85 matches)

110 permit icmp any any

120 permit ip any any

Ha dejado pasar 12 paquetes DHCP y ha bloqueado 85 paquetes que usaban el protocolo UDP. Aunque se ha configurado los puertos 67 y 68 para que sean filtrados, al visualizar las estadísticas de la ACL se ve que cambia el número por el "nombre" que Cisco le da a los puertos: 67-bootpc (cliente BOOTP, antecesor de DHCP) y 68-bootps (servidor BOOTP).

Base de Datos de asignaciones DHCP

Existe la posibilidad de que el servidor DHCP guarde una base de datos en un servidor externo aparte de en su memoria.

En este caso se configurará DHCP con todos los dispositivos en una misma red y sin reenvíos, donde habrá un servidor TFTP con dirección IP estática (asignada directamente en el S.O.) y allí se guardará la tabla de asignaciones.

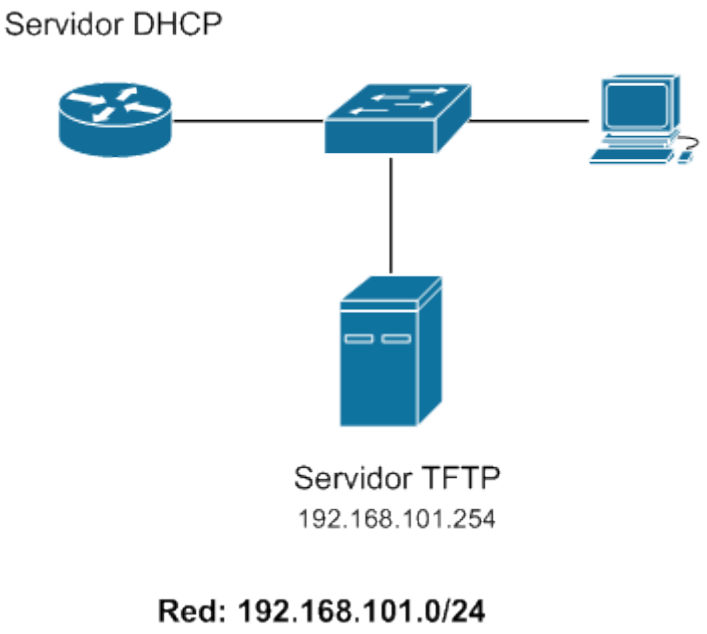
La sintaxis es:

```
Router(config)#ip dhcp database <URL> write-delay <sg> timeout
                                <sg>
```

Parámetros:

- URL: cadena completa indicando el protocolo, host/IP y ruta donde se debe guardar la BD.
- wirte-delay: tras haber un cambio, el retraso antes de guardar la BD en el servidor.
- timeout: cuanto tiempo ha de esperar antes de considerar el servidor como caído.

Topología:



Configuración:

Router(config)#ip dhcp pool DHCP-LAN

Router(dhcp-config)#network 192.168.101.0 255.255.255.0

Router(dhcp-config)#default-router 192.168.101.3

Router(dhcp-config)#exit

Router(config)#ip dhcp excluded-address 192.168.101.3

Router(config)#ip dhcp excluded-address 192.168.101.254

Router(config)#ip dhcp database tftp://192.168.101.254/dhcp-bd write-delay 60 timeout 5

Captura del fichero de texto de la BD:

Dirección C:\TFTP\dhcp-bd

```
"time" May 02 2010 01:40 AM
"version" 2
!IP address      Type  Hardware address   Lease expiration       VRF
192.168.101.131 id     0100.4095.31eb.8b  May 03 2010 01:39 AM

!IP address       Type      Hardware address  Interface-index


!IP address      Interface-index  Lease expiration      Server IP address  Hardware address  Vr
"end"
```

Aquí se puede ver en la cuarta línea la única asignación que hay ahora mismo (dir. IP 192.168.101.131).

Depuración del proceso:

DHCPD: Sending notification of DISCOVER:

DHCPD: htype 1 chaddr 0040.9531.eb8b

DHCPD: remote id 020a0000c0a8650300000000

DHCPD: circuit id 00000000

DHCPD: Seeing if there is an internally specified pool class:

DHCPD: htype 1 chaddr 0040.9531.eb8b

DHCPD: remote id 020a0000c0a8650300000000

DHCPD: circuit id 00000000

DHCPD: checking for expired leases.

DHCPD: dhcpd_age_bindings could not lock semaphore.

DHCPD: Adding binding to radix tree (192.168.101.1)

DHCPD: Adding binding to hash tree

DHCPD: assigned IP address 192.168.101.1 to client 0100.4095.31eb.8b.

DHCPD: Sending notification of ASSIGNMENT:

DHCPD: address 192.168.101.1 mask 255.255.255.0

DHCPD: htype 1 chaddr 0040.9531.eb8b

DHCPD: lease time remaining (secs) = 86400

1> !!

2> DHCPD: writing bindings to tftp://192.168.101.254/dhcp-bd.

3> DHCPD: writing address 192.168.101.1.

4> DHCPD: wrote automatic bindings to tftp://192.168.101.254/dhcp-bd.

En las cuatro últimas líneas, por orden, indica:

1. Los signos de admiración cerrados "!" indican que ha escrito en el servidor.
2. Indica que va a escribir la tabla de asignaciones en el servidor que configurado.
3. Indica las direcciones que se escriben, en este caso solamente hay un dispositivo que obtenga una dirección IP de manera dinámica.
4. Indica que ya se ha escrito la tabla de asignaciones.

Parámetros configurables

En los ejemplos anteriores, se ha mostrado diferentes configuraciones a nivel de servidor, donde solamente se configuraban la puerta de enlace y la red o asignaciones manuales de direcciones IP, pero en el mundo real normalmente son necesarios más parámetros, por ejemplo uno o más servidores DNS, dominio, ...

Aquí pongo la lista de parámetros adicionales que se pueden configurar:

Parámetro y sintaxis	Descripción
`domain-name <dominio>`	Indica el dominio al cual añadir el dispositivo.
`dns-server <dns1> [<dns2> ... <dns8>]`	Servidores DNS a ser usados por el cliente. Hasta 8 servidores DNS distintos.
`next-server <dhcp1> [<dhcp2>.. <dhcp8>]`	En caso de que haya varios servidores DHCP, se asignan al cliente de manera secuencial.
`bootfile <fichero>`	En caso de que el cliente deba cargar un fichero, se especifica aquí.
`netbios-name-server <nb1> [<nb2>...<nb8>]`	Servidores NetBIOS WINS. Hasta 8 en orden de preferencia.
`netbios-node-type <tipo>`	Tipo de nodo NetBIOS para un cliente Windows.
`default-router <gw1> [<gw2>...<gw8>]`	Se pueden especificar hasta 8 puertas de enlace.
`client-name <nombre>`	Nombre del cliente al qual se le hace una asignación DHCP manual.
`ip dhcp ping packets <num>`	Cantidad de *pings* que hace el servidor DHCP a la red de direcciones especificadas para verificar que están disponibles antes de asignarlas.
`ip dhcp ping timeout <milisegundos>`	Tiempo que esperará respuesta del ping un servidor DHCP antes de dar por disponible la dirección IP.
`ip dhcp bootp ignore`	Ignora las peticiones BOOTP.
`option <código> [instance <núm>] [ascii <cadena> | hex <cadena|dir. IP>]`	Añade opciones DHCP al servidor, la lista de opciones está disponible en un enlace en la introducción de este apartado.
`import all`	Importa todas las opciones de un servidor DHCP, usado cuando la configuración está en otro servidor.

Todos los comandos se ejecutan desde el modo de configuración del servidor DHCP: "**Router(dhcp-config)#**".

Liberación de asignaciones

Algunos sistemas operativos implementan la posibilidad de cuando van a ser cerrados, el sistema operativo avisa al servidor DHCP de que la dirección IP pasa a estar disponible, en el caso de Windows al cerrar el sistema operativo no avisa de que la dirección IP pasa a estar libre, ahora bien, si se ejecuta el comando "**ipconfig /release**" para borrar la información IP de la memoria, el sistema operativo avisa al servidor DHCP indicando que la asignación pasa a estar libre y así poder asignar la dirección IP liberada a otro cliente sin tener que esperar a que finaliza el tiempo de concesión.

Liberación de dirección IP:

1> DHCPD: Sending notification of TERMINATION:

2> DHCPD: address 192.168.101.1 mask 255.255.255.0

3> DHCPD: reason flags: RELEASE

4> DHCPD: htype 1 chaddr 0040.9531.eb8b

5> DHCPD: lease time remaining (secs) = 86095

6> DHCPD: returned 192.168.101.1 to address pool DHCP-LAN.

Explicación línea a línea:

1. Avisa de finalización de concesión.
2. La dirección que se libera es 192.168.101.1 con máscara 255.255.255.0
3. El motivo de la finalización es un paquete DHCPRELEASE.
4. Tipo de interfaz y dirección física (dirección MAC) de la interfaz.
5. Tiempo de concesión restante en segundos.
6. Pone como disponible la dirección IP 192.168.101.1.

Anexo: Referencias y consideraciones del documento PrincipIOS

He adaptado, aclarado, corregido, dado formato y organizado numerosa información al archivo **PDF** para **una correcta visualización en cualquier dispositivo, además de las ventajas que ofrece este formato respecto a portabilidad y tamaño**. El documento PDF es de **carácter gratuito** para beneficiar a los usuarios que forman parte de la comunidad. A su vez permite al autor **repasar** los conocimientos adquiridos y servir de fuente de inspiración para consultas específicas.

El texto de este libro y/o documento ha sido modificado y organizado por el autor, y gran parte obtenido de las páginas www.redespracticas.com del experto Carlos Paco
http://www.redespracticas.com/?Njs=t&pag=txtSobreMi.php

En dicho sitio se pone de manifiesto que el contenido de dicha página puede ser **usado, redistribuido, copiado y modificado libremente, con la mención necesaria**.

Los iconos de las topologías de red son propiedad de **Cisco Systems**, de acuerdo con su política los símbolos se pueden usar libremente pero no pueden ser modificados:

http://www.cisco.com/web/about/ac50/ac47/2.html

Hay alguna forma de Microsoft Visio, las cuales de acuerdo con la descripción que la ayuda indica:

"Las formas de patrón de Visio suministradas por Microsoft Corporation están protegidas por las leyes de derechos de autor. Puede copiarlas, reorganizarlas y modificarlas para su propio uso, así como distribuir los dibujos que las contengan. No puede vender ni distribuir las formas de patrón de Visio, ya sean originales o modificadas."

Se entiende que las "formas de patrón" sin contenido adicional no pueden ser distribuidas pero sí cuando forman parte de un dibujo donde se encuentran más formas.

Reitero que el carácter de esta guía es gratuito y **sin ánimo de lucro**, **con el mero propósito de ayudar a los usuarios de Internet y difundir conocimiento siguiendo la filosofía de mi blog**.

www.palentino.es

Gracias por la lectura, gracias a los contribuidores.