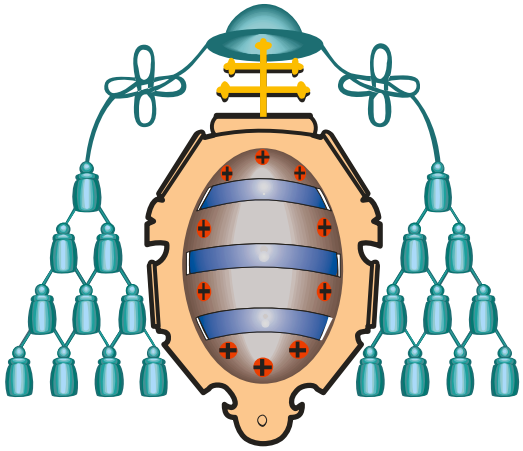
Ingeniería de redes

Grado en Ingeniería Informática en Tecnologías de la Información

Prácticas de laboratorio



UNIVERSIDAD DE OVIEDO

Área de Ingeniería Telemática

Contenidos

1

Introducción a Cisco IOS

1.1. Objetivos

En esta práctica se van a analizar los mecanismos de acceso y configuración de equipos de interconexión, y más específicamente algunas de las características básicas del sistema operativo Cisco IOS (Internetworking Operating System), presente en los equipos de dicho fabricante.

En la mayor parte de las prácticas de esta asignatura se va a emplear el simulador Packet Tracer. Dicho simulador ofrece un soporte bastante completo y realista de Cisco IOS, que servirá para familiarizarse con dicho sistema operativo sin que sea necesario tener acceso físico a equipos reales.

1.2. Introducción

Numerosos aspectos de las redes de comunicaciones requieren de la intervención de un administrador que los configure para prestar una funcionalidad determinada. Como ejemplos de estas configuraciones necesarias están la dirección IP y la máscara de subred en un *router* (funcionalidad mínima), los algoritmos de encaminamiento deseados, los mecanismos para evitar bucles en las redes locales o las técnicas de filtrado de tráfico.

Para poder particularizar todos estos aspectos en función del objetivo de la red, de los rangos de direcciones a utilizar, etc., la mayor parte del equipamiento de interconexión dispone de un sistema operativo configurable. Pese a las diferencias entre los sistemas operativos de los diferentes fabricantes, un aspecto que suelen compartir son los mecanismos de acceso a la interfaz de configuración.

1.3. Métodos de acceso y configuración de equipos de interconexión

Típicamente, los sistemas operativos de los equipos de interconexión ofrecen al usuario una interfaz en modo texto por línea de comandos, en la cual, mediante la ejecución de una serie de instrucciones, el administrador del equipo puede configurar el funcionamiento del equipo o consultar cierta información.

A esta interfaz por línea de comandos (CLI; Command Line Interface) se puede acceder mediante diversos mecanismos, que podemos clasificar de forma genérica en las siguientes categorías:

- **Configuración local**. Permite a los administradores configurar el equipo a través de una conexión a un puerto especial del equipo de interconexión, conocido como puerto de consola. Se realiza, habitualmente, desde el puerto serie de un PC al puerto de consola del equipo. Esta

conexión, dadas sus características, se lleva a cabo desde un PC cercano al equipo de interconexión, situado como máximo a una distancia igual a la longitud del cable de consola. Aunque, una vez configurado el direccionamiento IP y otros mecanismos de funcionamiento, se pueda recurrir a métodos de configuración remota, en general los primeros pasos de configuración de un equipo deben realizarse a través de la conexión de consola.
Las características de la conexión al puerto de consola de un equipo de interconexión dependen del fabricante del mismo. Una vez conectado el equipo al PC mediante el cable de consola (Figura 1.2), hay que utilizar software de emulación de terminal que permita acceder a la CLI del mismo. Entre las herramientas de emulación de terminal más extendidas podemos destacar *HyperTerminal* en sistemas operativos *Windows* y *Minicom* en sistemas operativos *Linux*.

- **Configuración remota**. Si el equipo que se va a configurar tiene correctamente asignado el direccionamiento IP, y la red de comunicaciones de la que depende tiene capacidad para encaminar tráfico IP, podemos emplear mecanismos de configuración remota como *telnet*, *ssh*, *HTTP* o *SNMP*. Lanzando una sesión de uno de estos protocolos a una de las direcciones IP que tenga asignado el equipo de comunicaciones, accederemos de forma remota a la CLI, siempre que, claro está, dicho mecanismo de acceso esté habilitado en el equipo de interconexión. La configuración remota tiene como ventaja el que no necesitamos desplazarnos a configurar el equipo. El principal inconveniente es la dependencia total del direccionamiento IP y de su correcto funcionamiento; si se modifica el direccionamiento IP del equipo, el protocolo de encaminamiento, las políticas de filtrado de tráfico, etc., el resultado será, en el mejor de los casos, la desconexión de la sesión a través de la cual estamos configurando el equipo. En el caso de haber cometido algún error en la configuración es posible que no podamos recuperar la conectividad con el equipo, siendo necesario desplazarse a su ubicación física para configurarlo.

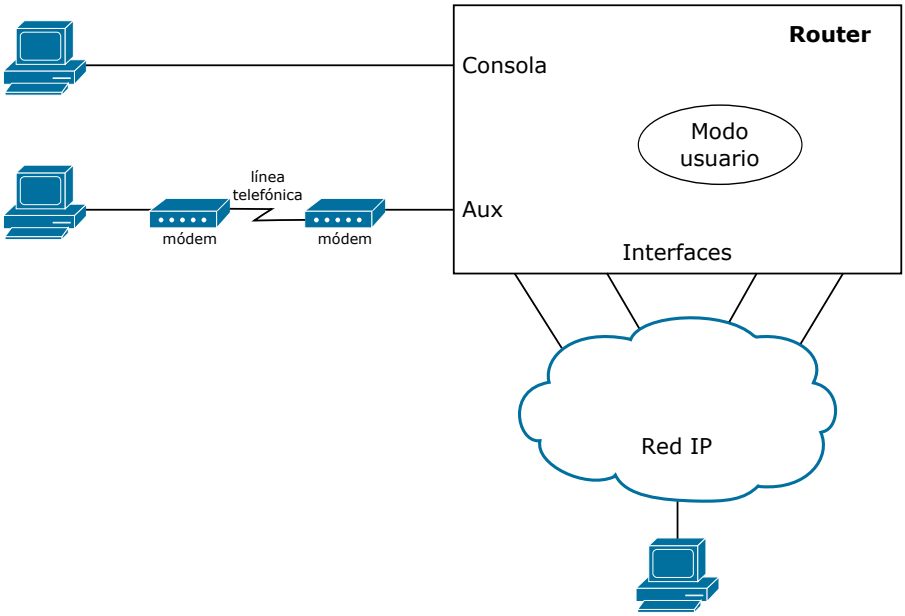


Figura 1.1: Métodos de acceso y configuración de equipos de interconexión.

En cuanto a los mecanismos de configuración remotos, hay que destacar que muchos de ellos transmiten todo el tráfico (usuarios, contraseñas e información) en abierto (sin cifrar) a través de la red, con los riesgos de seguridad que ello conlleva. En este grupo podemos incluir los protocolos *telnet* y *HTTP*. Las versiones 1 y 2 de *SNMP* también transmiten su información en abierto a través de la red.

De cara a garantizar la seguridad de las comunicaciones, y evitar que un atacante pueda obtener información sensible sobre nuestros equipos de interconexión mediante el análisis del tráfico intercambiado, hemos de utilizar mecanismos de configuración seguros como *ssh*. De esta forma, tanto el proceso de autentificación como la transmisión de información viaja cifrada por la red de comunicaciones.

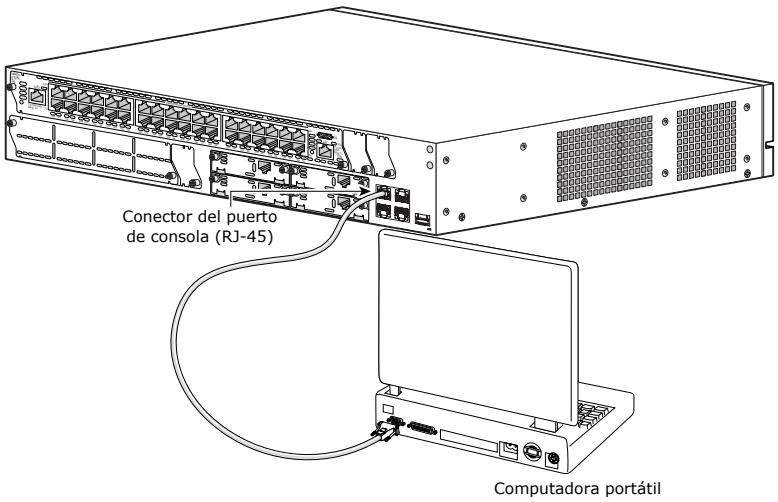


Figura 1.2: Conexión desde el puerto serie con conector DB-9 de un portátil al puerto de consola de un equipo de interconexión.

1.4. Sistema operativo Cisco IOS

Cada fabricante define los parámetros específicos de conexión al puerto de consola de sus equipos (tipo de conectores, patillaje y tasa binaria). Típicamente, los equipos Cisco disponen de un puerto de consola con conector RJ-45 o DB-9. El patillaje del cable a utilizar es totalmente cruzado: la patilla 1 de un extremo a la patilla 8 del otro, la patilla 2 a la 7, etc. En cuanto a la tasa binaria, típicamente es de 9600 bps, con 8 bits de datos, sin paridad y con 1 bit de parada.

La configuración inicial del equipo debe realizarse mediante este puerto de consola.

1.5. Modos de funcionamiento

En cualquiera de los modos de funcionamiento disponibles podemos obtener ayuda acerca de la sintaxis concreta de un comando mediante la tecla de tabulación (⟺), que autocompleta comandos de sintaxis unívoca, o mediante el símbolo ? , que nos muestra información sobre las posibilidades de las que disponemos.

Para agilizar la introducción de los comandos se permite no escribir todos sus caracteres, siendo necesario introducir únicamente aquéllos que hagan que el comando no se confunda con ningun otro de su mismo nivel. P.ej. para introducir el comando `configure terminal` basta con escribir `conf t`.

1.5.1. Modo usuario

El sistema operativo Cisco IOS dispone de dos modos de trabajo diferenciados: modo usuario y modo privilegiado. Al acceder a la línea de comandos nos encontramos en modo usuario, que se identifica mediante el símbolo **>** con el que termina el indicador de dicha línea de comandos. En este modo disponemos de permisos para consultar ciertos parámetros de funcionamiento del equipo, como pueden ser el hardware instalado y el tiempo desde el último inicio (`show version`), estadísticas de algunos protocolos (`show snmp`) o comprobar la conectividad del *router* (`ping` o `traceroute`).

1.5.2. Modo usuario con privilegios

Sin embargo, si queremos consultar información sensible, como el fichero de configuración o la tabla de rutas del equipo hemos de pasar a modo privilegiado, que se identifica mediante el símbolo **#** con el que termina el indicador de dicha línea de comandos. El paso de un modo a otro se realiza mediante los comandos `enable` y `disable` como puede verse en la Figura 1.3.

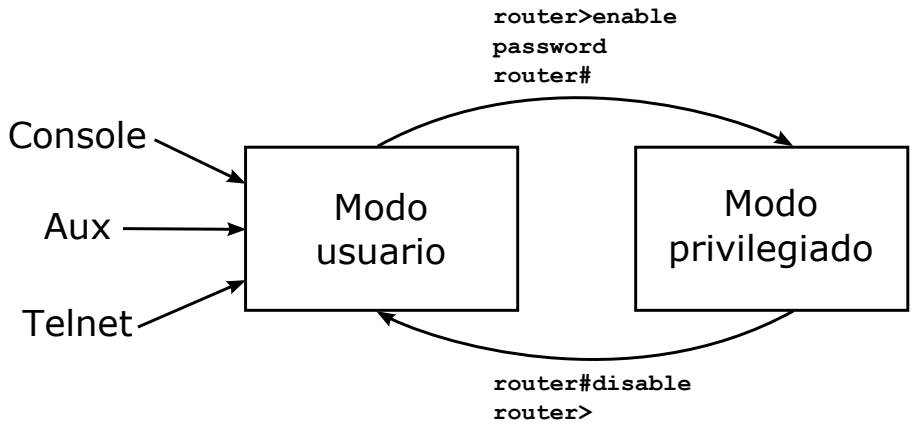


Figura 1.3: Modos de funcionamiento del sistema operativo Cisco IOS.

1.5.3. Modo de configuración

Si deseamos modificar el comportamiento del equipo, tras entrar en modo privilegiado hemos de entrar a su vez en modo configuración tecleando: `configure terminal`. En la Figura 1.4 se puede observar la descripción del modo configuración: cualquier comando tecleado con la sintaxis correcta afecta a la configuración activa del equipo, modificando su comportamiento.

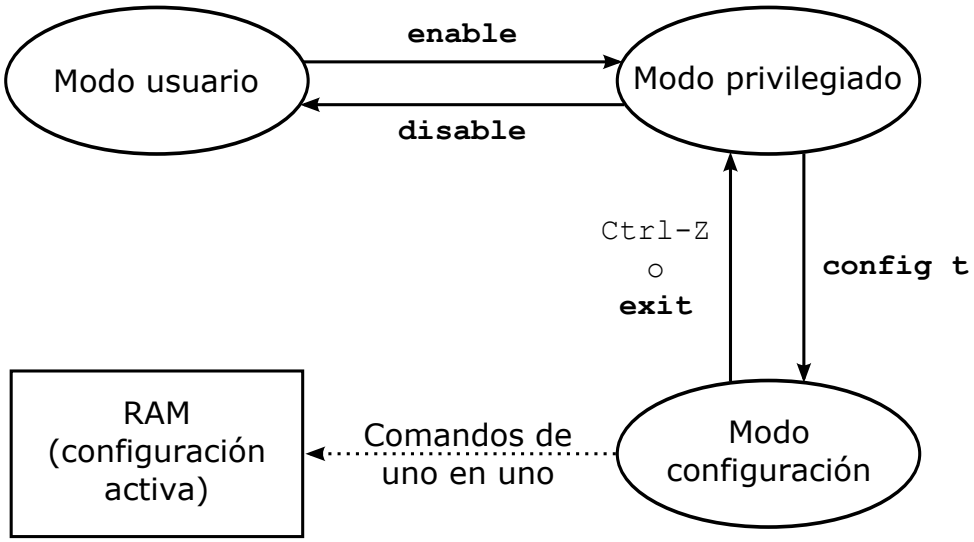


Figura 1.4: Modo de configuración en un equipo Cisco.

Dentro del modo de configuración, los comandos están organizados siguiendo una estructura de árbol; cada rama de este árbol se encarga, por ejemplo, de la gestión de una interfaz del equipo, de la gestión de un protocolo de encaminamiento determinado, etc. En la Figura 1.5 puede verse un ejemplo en el que, tras entrar en modo de configuración y definir la contraseña de usuario con privilegios, se modifica el nombre del equipo. Podemos ver cómo, efectivamente, el cambio realizado afecta inmediatamente al comportamiento del equipo, modificándose el indicador de la línea de comandos. Tras ello, se entra en modo de configuración de una de las interfaces del equipo (`Serial 0`) y se le asigna una descripción. Para salir de este modo de configuración hemos de teclear `exit` para volver al modo de configuración global, y de nuevo `exit` para salir del modo de configuración, o Ctrl + C para salir directamente.

Los cambios realizados en el modo de configuración se reflejan inmediatamente en el comportamiento del equipo y se pueden observar en el fichero de configuración cargado en memoria RAM (`running-config`). Si queremos que estos cambios de configuración sigan presentes en el siguiente reinicio del equipo, hemos de salvar esta configuración en la RAM no volátil. El proceso para hacerlo se describe en la siguiente sección.

```
router>enable
router#configure terminal
router(config)#enable password TEXTO
router(config)#line console 0
router(config-line)#login
router(config-line)#password cisco
router(config-line)#exit
router(config)#hostname router4
router4(config)#interface Serial 2/0
router4(config-if)#description Enlace con router 5
router4(config-if)#exit
router4(config)#exit
router4#
router4#show running-config
```

Figura 1.5: Ejemplo de configuración en un equipo Cisco IOS.

1.6. Tipos de memoria y estructura del sistema de ficheros

El funcionamiento de un equipo de interconexión Cisco viene marcado, fundamentalmente, por dos ficheros de configuración: el fichero de configuración en ejecución (`running-config`) y el fichero de configuración de arranque (`startup-config`). El fichero de configuración en ejecución marca el comportamiento actual del equipo, mientras que el fichero de configuración de arranque define el comportamiento que va a tener el equipo tras el próximo reinicio.

El proceso habitual de configuración de un equipo es el siguiente:

- Acceder a la CLI mediante alguno de los métodos descritos anteriormente.
- Realizar los cambios de configuración necesarios sobre el fichero de configuración en ejecución (`running-config`).
- Comprobar el correcto funcionamiento del equipo.
- Depurar el funcionamiento del mismo comprobando la conectividad y, generalmente, observando el fichero de configuración del mismo.
- Una vez comprobamos que el equipo funciona correctamente, salvaguardamos el fichero de configuración en ejecución en el fichero de configuración de arranque (`startup-config`).

Siguiendo este procedimiento, guardando únicamente en la configuración de arranque los cambios que comprobemos que son correctos, disponemos de un mecanismo para recuperar el control del equipo en caso de un fallo grave en la configuración: reiniciar el equipo.

Un equipo Cisco, generalmente, cuenta con 4 tipos de memoria:

- **DRAM** (Dynamic RAM). En esta memoria se encuentra la configuración activa del equipo (`running-config`). Es una memoria volátil que se reinicia cada vez que se reinicia el equipo, cargándose los contenidos de la configuración de arranque.
- **ROM**. En esta memoria se almacenan el código de arranque del equipo hasta el momento en que la imagen de IOS esté cargada y se le entregue el control. Además, almacena un sistema operativo de funcionalidades mínimas utilizable en casos de emergencia en los que, por la razón que sea, no se disponga de una imagen de IOS válida.
- **FLASH** (EEPROM, PCMCIA). Aquí se almacena típicamente la imagen de IOS que el equipo carga en el proceso de arranque. El punto exacto de almacenamiento de esta imagen de IOS es uno de los parámetros que se le pueden indicar al router en el proceso de configuración.
- **NVRAM** (RAM no volátil). Almacena la configuración de arranque del equipo (`startup-config`). Cada vez que se reinicie el equipo se cargará el contenido de este fichero y desaparecerá cualquier cambio efectuado sobre la `running-config` que no se haya guardado.

1.7. Gestión de ficheros en un equipo Cisco

Al igual que en prácticamente cualquier sistema operativo, el comando básico de gestión de ficheros es el comando `copy`. Este comando nos permitirá pasar contenidos de la NVRAM a la RAM y viceversa, alterando el comportamiento actual del equipo o el comportamiento del equipo tras el próximo reinicio (Figura 1.6).

Adicionalmente, la mayor parte de los equipos de interconexión disponen del protocolo *TFTP* (Trivial FTP), que permite el intercambio de archivos sin autentificación a través de una red. En un equipo Cisco, el comando `copy`, además de permitirnos gestionar de forma local los ficheros, nos permite copiar a/desde un equipo remoto. Tras teclear el comando `copy` con origen o destino en `tftp`, se solicitará información sobre la dirección IP del servidor *TFTP* y sobre el nombre del equipo al/desde el que copiar. Además, los equipos Cisco incluyen la posibilidad de cargar en el arranque un fichero de configuración que resida en un equipo remoto en lugar del almacenado en su RAM no volátil.

Este soporte permite salvaguardar no sólo ficheros de configuración sino también imágenes de IOS en un servidor remoto. Este es el mecanismo que permite actualizar la imagen del sistema operativo en caso de ser necesaria una actualización (funcionalidades no soportadas o *bugs*).

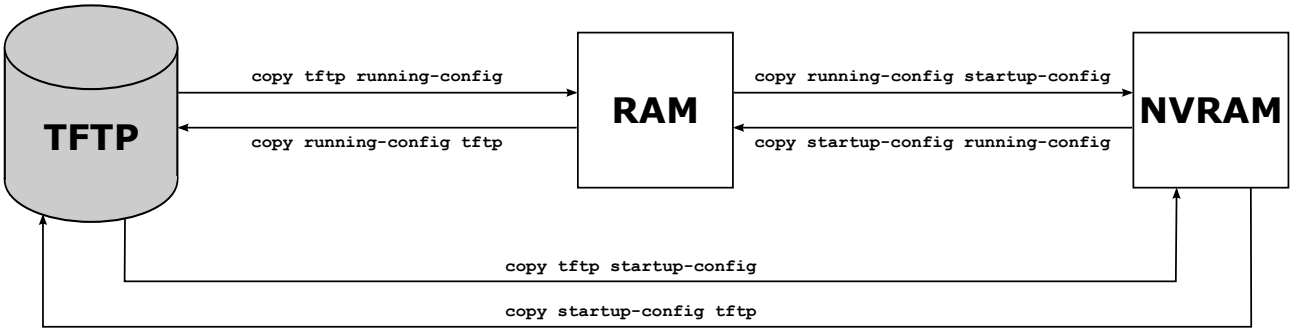


Figura 1.6: Copia en el sistema de ficheros del sistema operativo Cisco IOS.

1.8. Nomenclatura de las interfaces

Un equipo de interconexión contará con una serie de puertos mediante los que se conecte a la red de comunicaciones. Por ejemplo, un *switch* LAN con 24 puertos *FastEthernet* 10/100 tendrá 24 interfaces *FastEthernet*.

Dentro del sistema operativo Cisco IOS, cada interfaz tiene un nombre que la identifica. Este nombre está formado por la concatenación del tipo de interfaz (*Ethernet*, *FastEthernet*, *GigabitEthernet*, *Serial*, *BRI* [Acceso básico RDSI], ATM, etc.), el módulo en el que se encuentra (el número de tarjeta) y el orden que ocupa dentro de esa interfaz. Por ejemplo, el primer puerto del primer módulo de un *switch* LAN con dos tarjetas de 24 puertos *FastEthernet* se identificará como *FastEthernet 0/1* (los módulos se numeran de 0 a n-1 y los puertos de 1 a n). Por tanto, si quisiésemos comprobar el estado y las estadísticas de uso del ese puerto, ejecutaríamos el comando `show interfaces FastEthernet 0/1`.

IMPORTANTE: Packet Tracer numera las interfaces de un modo diferente a un equipo Cisco real. Cada interfaz del equipo pertenece a un módulo diferente y los módulos empiezan a numerarse en 0. De la misma forma, las interfaces comienzan a numerarse en 0.

1.9. Algunos comandos de Cisco IOS

- Información genérica del equipo:
  `show version`
- Contenido del fichero de configuración activa:
  `show running-config`

- Contenido del fichero de configuración de arranque:
  `show startup-config`
- Contenido de la tabla de direcciones MAC de un *switch*:
  `show mac-address-table`
- Estadísticas sobre las interfaces del equipo:
  `show interfaces`
- Estadísticas sobre las interfaces IP del equipo:
  `show ip interface brief`
- Mostrar contenidos del sistema de ficheros:
  `dir [flash: | nvram: | pcmcia | ...]`
- Reiniciar el equipo desde línea de comandos:
  `reload`

1.10. Ejemplos de configuración

A continuación se presentan varios ejemplos con la secuencia de comandos necesaria para configurar los distintos tipos de interfaces de un equipo de comunicaciones. También se incluyen las configuraciones de diversos mecanismos de encaminamiento que se emplearán en sesiones posteriores de prácticas.

- Interfaz Fast Ethernet. Es necesario configurar su dirección IP y la máscara de subred, empleando para ello el comando `ip address <dirección> <máscara>`.

```
Router0>enable
Router0#configure terminal
Router0(config)#interface FastEthernet 0/0
Router0(config-if)#ip address 192.5.5.1 255.255.255.0
Router0(config-if)#no shutdown
```

- Interfaz serie (modo DCE). Se han omitido los dos primeros comandos, que serán los mismos que en el caso anterior. Al tratarse de una interfaz que debe proporcionar la señal de reloj para sincronizar la comunicación habrá que indicar la velocidad de dicho reloj (mediante el comando `clock rate`).

```
Router0(config)#interface Serial 2/0
Router0(config-if)#ip address 201.100.11.1 255.255.255.0
Router0(config-if)#clock rate 64000
Router0(config-if)#no shutdown
```

- Interfaz serie (modo DTE). Igual a la anterior pero sin el comando `clock rate`.

Nota: El comando `no shutdown` emitido al final de las listas de comandos anteriores sirve para activar la interfaz correspondiente. El comando equivalente para desactivarla es `shutdown`.

- Rutas estáticas. El formato general del comando es: `ip route <red_destino> <máscara> <gateway>`

```
Router0>enable
Router0#configure terminal
Router0(config)#ip route 219.17.100.0 255.255.255.0 201.100.11.2
```

- Rutas dinámicas (RIP). El formato general del comando es: `network <dirección_red>`

```
Router0>enable
Router0#configure terminal
Router0(config)#router rip
Router0(config-router)#network 172.16.2.0
```

1.11. Acceso a la consola de los dispositivos de comunicaciones en Packet Tracer

El simulador Packet Tracer ofrece la posibilidad de configurar los equipos mediante una conexión de consola, e incorpora una implementación bastante completa del sistema operativo Cisco IOS de los equipos simulados. El acceso a dicha consola se puede realizar de dos maneras:

- Mediante la pestaña CLI del cuadro de diálogo de configuración del dispositivo, como se muestra en la Figura 1.7.
- Mediante una conexión de consola realizada desde un PC y ejecutando un programa de emulación de terminal en dicho PC. Esta última opción es la más realista y la que se empleará preferentemente en el desarrollo de las prácticas.

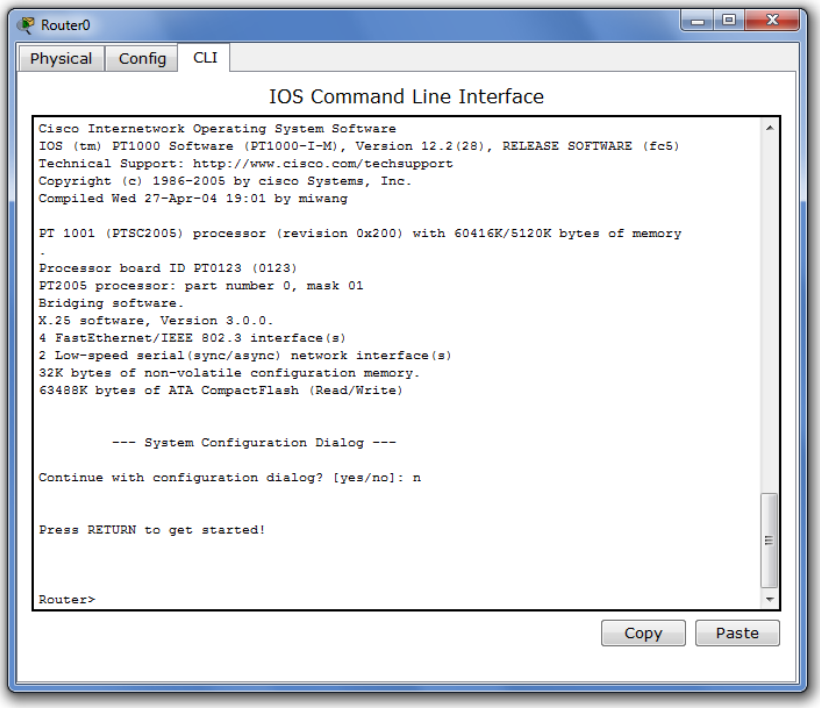


Figura 1.7: Acceso a la interfaz de línea de comandos (CLI) en Packet Tracer.

Como ayuda adicional para el aprendizaje de los comandos de Cisco IOS, el cuadro de diálogo para la configuración de los dispositivos en modo gráfico presenta en su parte inferior una zona en la que se muestran los comandos que sería necesario introducir para aplicar los cambios de configuración realizados mediante la interfaz gráfica.

1.12. Ejercicio propuesto

En el simulador Packet Tracer, crea una infraestructura de red formada por un *router*, dos *switches* LAN y cuatro PC, como la que puede verse en la Figura 1.8. Asigna el direccionamiento IP en las interfaces del *router* y de los PC para que el tráfico originado en cualquier PC pueda llegar a cualquier otro. Realiza la configuración de los equipos a través del puerto de consola.

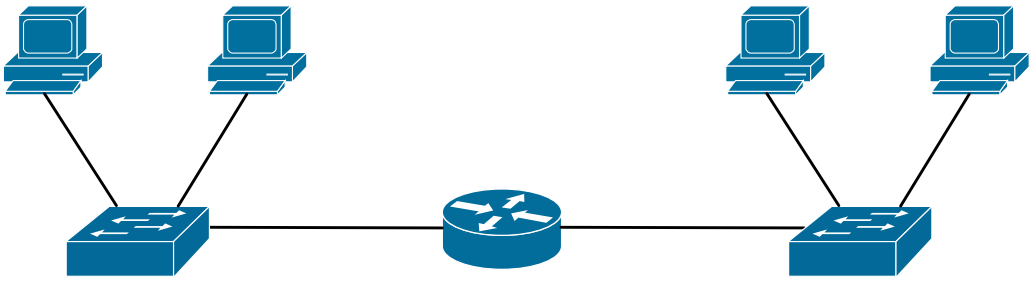


Figura 1.8: Escenario de ejemplo para el ejercicio propuesto.

- Instrucciones para entrar en modo de configuración de la interfaz y asignar una dirección IP a la interfaz.
  `router(config)#interface Tipo Módulo/Número`
  `router(config-if)#ip address DIR_IP MASK`
- ¿Cuál es el contenido del fichero de configuración?
  `router#show running-config`
- ¿Cuál es la tabla de direcciones MAC de los *switches*? ¿Por qué cada uno de los *switches* no conoce ningún equipo que no esté conectado a su misma interfaz del *router*?
  `switch#show mac-address-table`
- Diferencias entre modo usuario/superusuario y configuración.

  1. En modo usuario:
     - ¿Cuál es la tabla de rutas del equipo de encaminamiento?
       `router>show ip route`
     - ¿Cuál es el estado de sus interfaces y el direccionamiento IP que tiene asignado?
       `router>show ip interface brief`
     - ¿Dispones de conectividad con otros equipos del entorno?
       `router>ping DIR_IP`
     - ¿Cuál es la ruta que siguen los paquetes a lo largo de la red?
       `router>traceroute DIR_IP`
     - ¿Qué comandos se teclearon con anterioridad?
       `router>show history`
     - ¿Qué información muestra el comando `show version`?
     - ¿Puedes ver la configuración en ejecución del equipo? ¿Y la de arranque?
  2. En modo superusuario:
     - ¿Puedes ver la configuración en ejecución del equipo? ¿Y la de arranque?
     - Reinicia el equipo (`reload`) ¿Qué configuración tiene el equipo?
     - ¿Puedes copiar la configuración en ejecución en la configuración de arranque? ¿Qué configuración tiene ahora el equipo tras reiniciarlo?
     - ¿Puedes ejecutar los comandos del usuario sin privilegios? En caso afirmativo, ¿Cuál consideras que es la necesidad de disponer de un modo de usuario sin privilegios?
  3. En modo de configuración:
     - ¿Puedes ver el fichero de configuración del equipo? ¿Y la tabla de rutas?

2

Conmutación LAN

2.1. Objetivos

La aparición de los *switches* (conmutadores LAN) permite reducir las colisiones en una red local, segmentándola en múltiples dominios de colisión. Pese a la aparente sencillez de estos despliegues, hay multitud de mecanismos que se ejecutan de cara a permitir la utilización más eficiente del ancho de banda y evitar las colisiones. El objetivo de esta práctica es que el alumno tome contacto con estos conceptos, analizando los mecanismos de funcionamiento de las redes locales.

2.2. Introducción

En una red local, todos los equipos conectados a un mismo *hub* pertenecen a lo que se conoce como dominio de colisión. La utilización de un *switch* LAN permite segmentar la red en múltiples dominios de colisión, mejorando las prestaciones que obtienen los equipos conectados a la red.

Para ello, los *switches* han de implementar una serie de mecanismos, como el aprendizaje de direcciones MAC, las decisiones de reenvío/filtrado de tramas y los algoritmos necesarios para evitar bucles en la red.

2.3. Aprendizaje hacia atrás

Los *switches* basan su funcionamiento en el envío de una trama únicamente por los puertos necesarios. Es decir, que las tramas destinadas a un equipo con dirección MAC A sólo se reenvíen hacia el puerto en el que se encuentra dicho equipo. Esto, obviamente, mejora la eficiencia en la utilización del ancho de banda y aumenta la seguridad de la red.

Para poder llevar a cabo estas funciones, los *switches* han de ser capaces de adquirir conocimiento acerca de la posición de los clientes. Su mecanismo de aprendizaje se conoce como "aprendizaje hacia atrás", y pasa por las siguientes fases:

- En un primer momento, el *switch* no conoce la posición de ninguno de los equipos de la red.
- La primera trama que reciba (**trama 1**) la reenviará por todos sus puertos, excepto por el que la recibió, para asegurarse de que le llegue a su destinatario previsto. Al recibir esta trama, el *switch* apunta la dirección MAC de origen y el puerto por el que la recibió en su tabla de direcciones MAC (**MAC 1**, **puerto 1**). Este proceso se repite con cada trama que reciba y para la que no tenga una entrada en la tabla de direcciones MAC.
- Cuando el *switch* reciba la respuesta a la **trama 1**, consultará su tabla de direcciones MAC, encontrará la entrada relacionada con el primer envío (**MAC 1**), y sabrá cuál es el puerto por el que debe enviar dicha trama (**puerto 1**).

Las entradas en las tablas de direcciones MAC no se almacenan indefinidamente. De esta forma la red se adapta a cambios en la topología (cambios en la distribución de los elementos de interconexión o cambios en la posición de los clientes en la red). El tiempo de vida de las entradas de la tabla de direcciones MAC es un parámetro que, por lo general, se puede configurar en la red.

2.4. Decisiones de reenvío/filtrado de tramas

La lógica que debe implementar un *switch* cuando recibe una trama, a la hora de tomar la decisión de reenvío/filtrado de la misma, es la siguiente:

- Si la dirección de destino es de multidifusión (*multicast*/*broadcast*), debe reenviarla por todos los puertos excepto por el que la recibió.
- Si la dirección de destino es única (*unicast*), pero no hay una entrada en la tabla para esa dirección MAC, debe reenviarla también por todos los puertos excepto por el que la recibió.
- Si la dirección es única y existe una entrada en la tabla con dicha dirección MAC, y el puerto de destino no es el mismo por el que recibió la trama, debe reenviar la trama por ese puerto de destino.
- En otro caso debe filtrar la trama. Esto puede darse en el caso de tener múltiples equipos conectados a un *hub* y éste, a su vez, conectado a un puerto de un *switch*. En este caso, cuando dos equipos conectados al *hub* se envíen tráfico entre sí, el *switch* lo recibirá, siendo necesario filtrarlo, no reenviarlo.

2.5. Topologías redundantes

En un despliegue de red local pueden instalarse enlaces redundantes para proteger el servicio frente a posibles fallos en algunos de dichos enlaces. El problema de estas topologías es la aparición de bucles.

Los bucles en las redes locales pueden causar múltiples problemas, como reenvíos múltiples de la misma trama, inestabilidad de la tabla de direcciones MAC y tormentas de *broadcast*.

En la Figura 2.1 puede verse un ejemplo gráfico de los problemas que pueden aparecer en una red local con las topologías redundantes. En esa topología existen dos enlaces que cumplen la misma funcionalidad: unir el servidor con el router de salida.

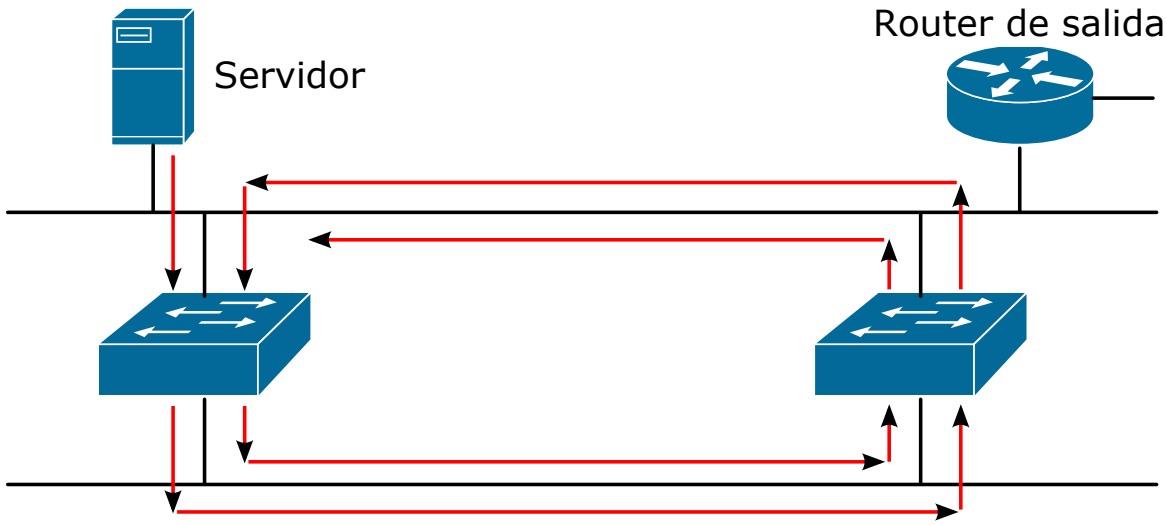


Figura 2.1: Topología redundante en una red LAN.

Se producirá una tormenta de *broadcast* cuando uno de los equipos de la red, como por ejemplo el servidor, envíe tráfico de difusión (*broadcast*). En caso de que el servidor deba enviar tráfico hacia una dirección IP del exterior de la red, lo primero que tendrá que hacer es resolver la dirección MAC del destino mediante una petición *ARP*. Está petición *ARP* viajará en forma de tráfico de difusión dirigido a todos los equipos de la red. Los dos *switches* recibirán la trama de difusión y la tratarán de la misma forma: la reenviarán por todos sus puertos menos por el que la recibieron. Cuando reciban el primer reenvío volverán a efectuar la misma operación. Este fenómeno de reenvío sucesivo se llevará a cabo a la velocidad de conmutación de los *switches*, consumiendo todos sus recursos de CPU e inhabilitando la red. Además, el tráfico generado por esta tormenta de *broadcast* también lo recibirán todos los PC de la red, provocando el colapso de sus CPU.

El caso del reenvío múltiple de la misma trama *unicast* está relacionado con la inestabilidad de la tabla de direcciones MAC. En un caso como el anterior, la primera trama a un destino dado se reenviará por todos los puertos del *switch* menos por el que se recibió. En este caso, si una trama originada en el servidor va destinada a una dirección MAC no conocida, dicha trama se reenviará por todos los puertos menos por el que se recibió. De esta manera, los dos *switches* introducirán en su tabla de direcciones MAC una entrada indicando que la MAC del servidor está en su interfaz superior. Cuando reciban la trama por la interfaz inferior, supondrán que la topología ha cambiado y cambiarán la entrada existente en la tabla de direcciones MAC por una en la que se indique que ahora está situada en el puerto inferior. Este proceso se repetirá indefinidamente.

2.6. *Spanning Tree Protocol* (STP)

El protocolo encargado de eliminar bucles en las redes locales se conoce como *Spanning Tree* (STP). Este protocolo, mediante el intercambio de tramas entre los *switches*, permite seleccionar qué partes de una infraestructura basada en *switches* deben deshabilitarse para evitar la aparición de bucles. Esta inhabilitación se lleva a cabo bloqueando de forma lógica los enlaces redundantes. STP se ejecuta cada vez que se detecta un cambio en la topología (caída/aparición de un nuevo enlace), para adaptar los enlaces bloqueados a la situación actual de la red.

Cada equipo de una LAN en la que se ejecute STP tiene un identificador, formado por dos partes concatenadas:

- Un identificador de prioridad de 2 bytes. Este identificador puede manipularlo un administrador, siendo 0 la máxima prioridad. Es la parte más significativa del identificador de STP.
- La dirección MAC base del dispositivo (6 bytes). Un *switch* tiene múltiples interfaces Ethernet. El fabricante asigna una dirección MAC base, y sumando a esta MAC el número de puerto se obtiene la dirección de cada uno de los puertos del *switch*.

Los dispositivos que ejecutan STP en una red local se intercambian tramas de gestión, llamadas BPDU (Bridge Protocol Data Units) a intervalos regulares. Mediante estas tramas descubren los identificadores de sus vecinos y los costes hacia cada equipo de la red, que se calculan sumando los costes de cada enlace atravesado por una BPDU. Estos costes de los enlaces vienen dados por el ancho de banda de las líneas, y se presentan en la siguiente tabla:

Velocidad del enlace	**Coste (Revisión IEEE)**
10 Gbps	2
1 Gbps	4
622 Mbps	6
155 Mbps	14
100 Mbps	19
45 Mbps	39
16 Mbps	62
10 Mbps	100
4 Mbps	250

2.7. Estados de STP

Un puerto en STP puede estar en uno de los siguientes estados (Figura 2.2):

- **Bloqueado** (**Blocking**). El puerto se ha deshabilitado para evitar bucles. Es un estado final de STP.
- **Escuchando** (**Listening**). Escuchando las BPDU para negociar la adaptación a una nueva topología. Es un estado intermedio de STP.
- **Aprendiendo** (**Learning**). Escucha tráfico recibido desde los usuarios pero todavía no lo reenvía. Se utiliza para apreder direcciones MAC y reducir la inundación de tramas inicial.
- **Reenvío** (**Forwarding**). El puerto recibe y reenvía tráfico. Es un estado final de STP.

Como se puede observar en la Figura 2.2, el paso de **Bloqueado** a **Reenvío** no es ni mucho menos instantáneo. Aunque los tiempos en cada estado son configurables, en su versión predeterminada un puerto puede emplear unos 50 segundos en pasar de **Bloqueado** a **Reenvío** ante un cambio de topología.

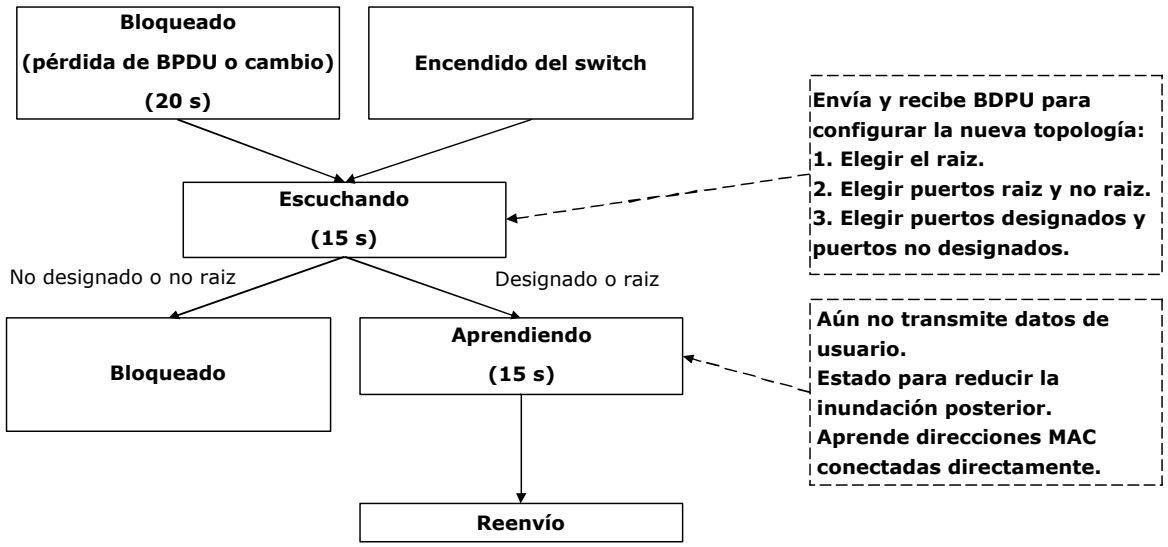


Figura 2.2: Estados del protocolo Spanning Tree.

En estado **Bloqueado** un puerto ni envía ni recibe tramas de datos.

En estado **Escuchando** un puerto ni envía ni recibe tramas de datos.

En estado **Aprendiendo** un puerto recibe tráfico y aprende direcciones pero no reenvía.

En estado **Reenvío** un puerto recibe y envía tráfico.

2.8. Ejecución de STP

La ejecución del protocolo STP pasa por las siguientes fases, que se ilustran en la Figura 2.3.

1. Elección del elemento **raiz (root)**. En una red en la que se ejecute STP sólo puede existir un elemento **raiz**, que será el que tenga el menor identificador de prioridad. El elemento **raiz** tiene todos sus puertos marcados como **Designados** o, lo que es lo mismo, en estado **Reenvío** (enviando y recibiendo tráfico).
2. Elección de los puertos **raiz** de los elementos **no raiz**. En cada elemento **no raiz** de la red se selecciona un puerto **raiz** como aquel con menor coste al elemento **raiz** (por el que se haya recibido la BPDU de menor coste desde el elemento **raiz**). Este puerto se sitúa en estado **Reenvío**.

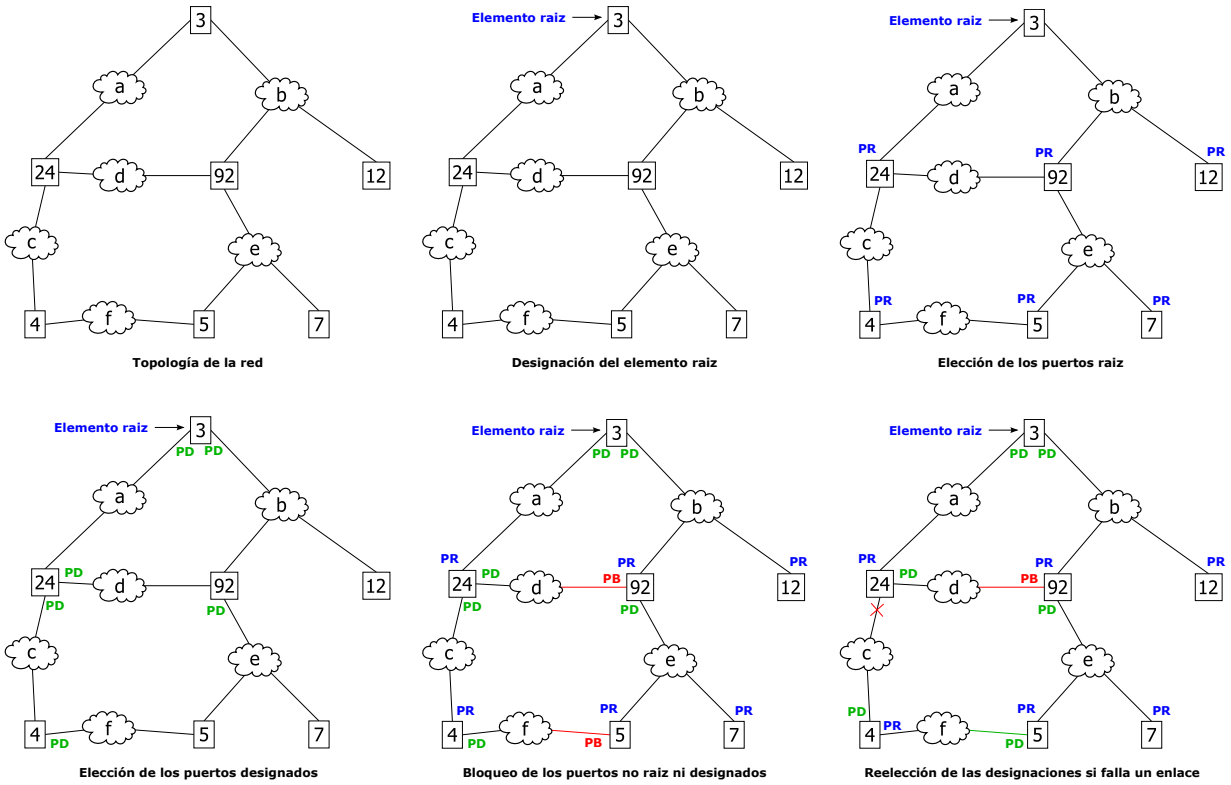


Figura 2.3: Fases en la ejecución del protocolo STP.

3. En cada segmento de la red se selecciona un puerto como **Designado**. Este puerto es el del *switch* que envía la BPDU de menor coste originada en el **raiz**. En cada segmento de la red sólo puede existir un puerto en estado **Designado**.

4. Si un puerto no se seleccionó en ninguna de las fases anteriores pasa a estado **Bloqueado**.

En la Figura 2.4 se puede observar un ejemplo de ejecución de STP, en el que el administrador no ha manipulado la elección del **raiz**: todos los elementos de la red tienen el mismo identificador de prioridad. Por lo tanto, la elección de **raiz** y la solución de los posibles empates se realizan en función de la dirección MAC base de los dispositivos. Situando en estado **Bloqueado** el puerto 1 del Switch Y se deshace la posibilidad de aparición de bucles en la red.

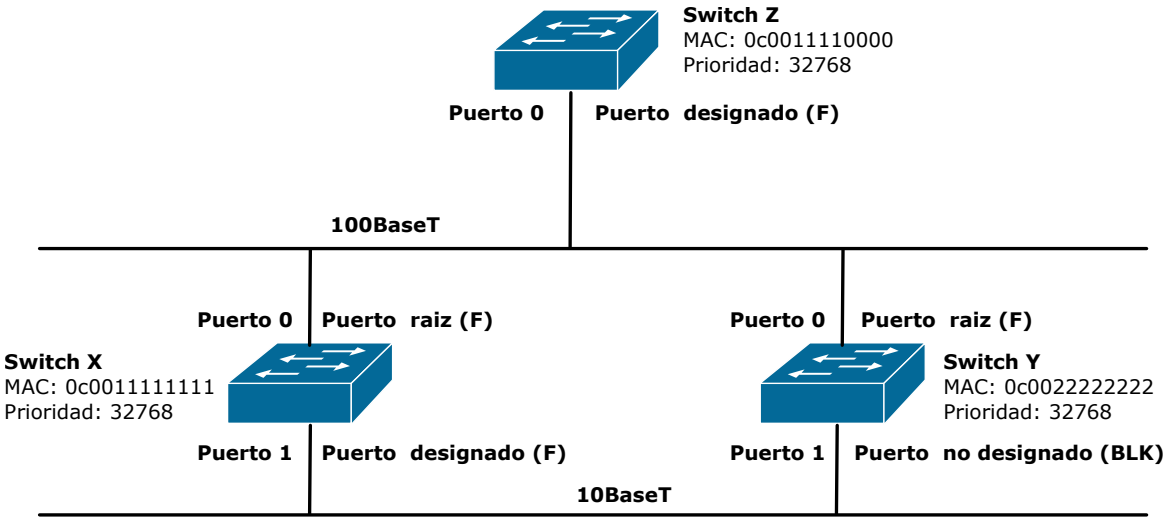


Figura 2.4: Ejemplo de ejecución de Spanning Tree.

La elección del elemento **raiz** en STP se puede manipular intencionadamente, de forma que el elemento de mayor capacidad de conmutación y con líneas de mayor ancho de banda se seleccione

como elemento **raiz**. Si analizamos algunos casos de ejecución de STP podemos ver cómo el elemento **raiz** forma la parte central de la arquitectura, el núcleo de nuestra red LAN, y por lo tanto debe ser el equipo que soporte una mayor cantidad de tráfico.

2.9. Transiciones rápidas

La transición a estado **Reenvío** desde que se activa un puerto, o desde que se detecta un cambio en la topología, es relativamente lenta. En aquellos puertos para los que exista la seguridad de que no se van a producir bucles (puertos a los que se conectan directamente equipos finales), se puede habilitar la transición directa a estado **Reenvío**. Hay que tener mucho cuidado a la hora de seleccionar qué puertos se sitúan en transición rápida, ya que, en caso de configurar como de transición rápida un puerto en el que se puedan producir bucles, se corre el riesgo de provocar una tormenta de *broadcast* e inhabilitar la red y los equipos conectados a ella.

2.10. Reparto de carga y topologías redundantes

Inhabilitar enlaces de la red nos permite evitar bucles y disponer de enlaces redundantes en caso de que algún equipo o enlace deje de funcionar. Sin embargo, mientras no haya cambios de topología, los enlaces redundantes permanecen inactivos. Además, existen muchas situaciones en las que el ancho de banda de una sola línea de comunicaciones no es suficiente para enlazar dos *switches* de alta capacidad.

Para permitir el reparto de carga entre varias líneas sin que STP inhabilite enlaces, los fabricantes de equipos han dispuesto soluciones propietarias como Etherchannel, en el caso de Cisco, o Multilink Trunking, en el caso de Nortel. En ambas soluciones se permite unir en el mismo grupo de enlaces hasta 8 enlaces individuales, consiguiendo un ancho de banda agregado 8 veces superior al de una línea individual o repartiendo el tráfico en función de la VLAN a la que pertenezcan.

2.11. Seguridad en capa 2

El uso de algunas redes locales, o de una parte de ellas, permite establecer políticas de seguridad basadas en el reconocimiento de direcciones MAC. Por ejemplo, en un puerto de un *switch* al que se conecte un servidor o el enlace que da acceso a ese servidor, se pueden establecer políticas de filtrado que sólo habiliten como dirección origen válida la MAC del servidor o servidores. De esta forma, cualquier persona que consiguiese acceso a un puerto red no podría cursar tráfico a través de él.

Hay que destacar que estas medidas de seguridad no siempre son fácilmente aplicables o factibles: en redes locales en las que los equipos cambien a menudo de posición (por ejemplo, redes locales con segmentos inalámbricos), o despliegues de red en una sala en la que se conecten personas externas a la empresa cada vez que se organiza una reunión.

2.12. CDP (Cisco Discovery Protocol)

Con el objetivo de simplificar el descubrimiento del entorno de red, los equipos Cisco disponen del protocolo CDP, que se ejecuta en la capa 2 y se encarga de averiguar el tipo de equipos existentes en su entorno. Este protocolo se encuentra activado de manera predeterminada en todos los equipos Cisco y no precisa de direccionamiento IP asignado al equipo. Un equipo con CPD activado envía de forma periódica anuncios con información sobre sí mismo como la siguiente:

- Tipo de equipo.
- Versión de imagen de IOS que ejecuta.
- Tipo de puerto por el que está enviando el anuncio CDP.

En algunas interfaces, como por ejemplo a las que están conectados usuarios de la red, no resulta recomendable proporcionar información sensible sobre la propia red. Para ello es posible deshabilitar tanto el protocolo en general como el envío de anuncios en las diferentes interfaces.

2.13. Simulación

En esta parte de la práctica se pretende que el alumno entre en contacto con algunas de las configuraciones típicas de los *switches* y los comandos necesarios para ello. Para realizar la práctica se emplearán algunos de los comandos ya examinados en la práctica anterior y se introducirán algunos nuevos.

2.13.1. Lista de comandos básicos

En esta parte de simulación se emplearán los siguientes comandos de IOS:

- Ver la tabla de direccionamiento MAC de un switch:
  `show mac-address-table`
- Ver información sobre las interfaces:
  `show interfaces FastEthernet <0-9>/<0-0>`
- Ver información relativa al filtrado realizado en un puerto determinado:
  `show port-security interface FastEthernet <0-9>/<0-0>`
- Deshabilitar manualmente la interfaz (modo de configuración de una interfaz):
  `shutdown`
- Habilitar manualmente la interfaz (modo de configuración de una interfaz):
  `no shutdown`
- Asignar nombre a un equipo (modo de configuración):
  `hostname nombre`
- Configuración de CDP:
  - Deshabilitar/Habilitar CDP:
    `[no] cdp run`
  - Deshabilitar/Habilitar CDP en una interfaz (modo de configuración de una interfaz):
    `[no] cdp enable`
- Información sobre dispositivos Cisco cercanos:
  `show cdp neighbors`
- Configuración de seguridad en la capa 2:
  - Habrá que acceder al modo de configuración del *switch*, y después acceder a la interfaz en la que se quiere aplicar el filtrado.
  - Una vez en la interfaz, se debe activar la seguridad del puerto (`port-security`) según los criterios elegidos:
    - Filtrar por dirección MAC. En el formato de dirección MAC utilizado cada número N representa 2 bytes de la dirección MAC:
      `Switch#switchport port-security mac-address N.N.N`
    - Filtrar por número de equipos:
      `Switch#switchport port-security maximum N`
    - Establecer el comportamiento en caso de violación:
      `Switch#switchport port-security violation shutdown`
  - Finalmente se activará ese filtrado en la interfaz correspondiente.
    `Switch#switchport port-security`

2.14. Ejercicio

Crea un entorno de red formado por tres *switches* con enlaces redundantes entre ellos tal y como se muestra en la Figura 2.5.

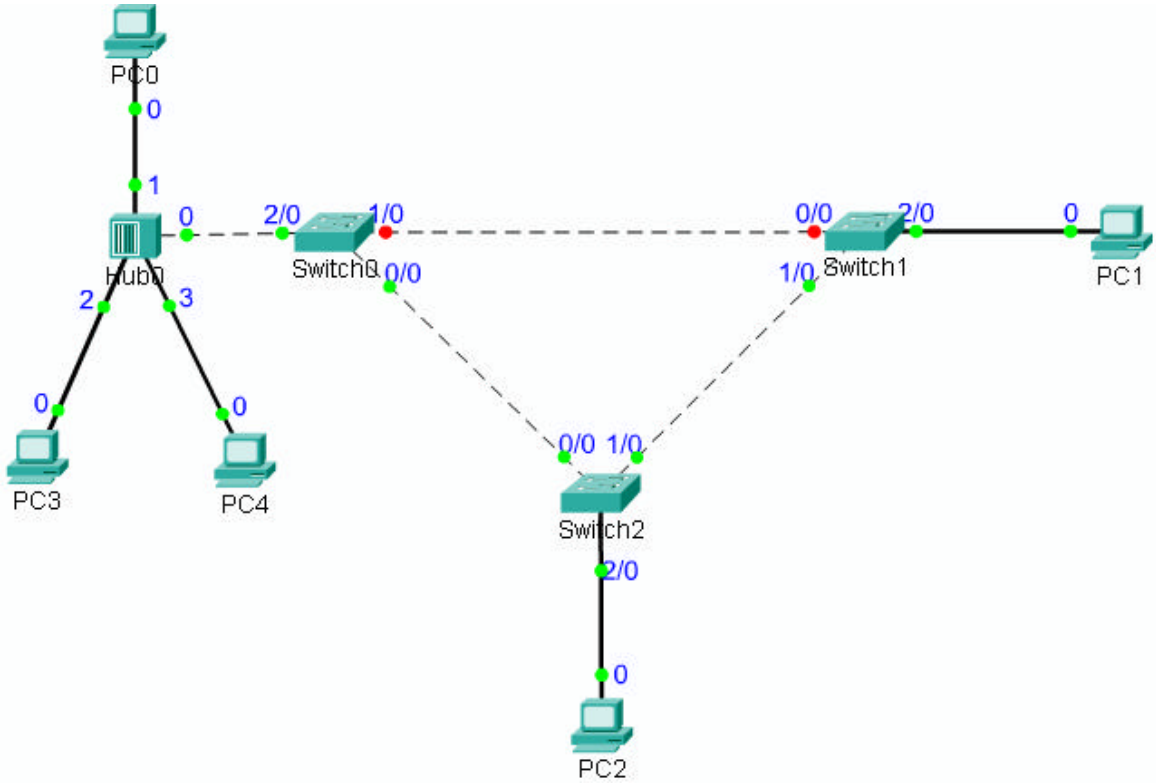


Figura 2.5: Conexión de los *switches* para el ejercicio.

2.14.1. Decisiones de reenvío/filtrado

Analiza la tabla de direcciones MAC de los tres *switches*. ¿Cuál es la razón para que los *switches* aprendan esas direcciones MAC en esos puertos en concreto?

Observa los procesos de filtrado y envío de los *switches*. Para ello envía tráfico entre PC0 y PC3, tráfico entre PC3 y PC1 y tráfico de *broadcast* desde PC0, y observa el intercambio de tramas, así como las tablas MAC de los *switches* involucrados. ¿Cuántos dominios de *broadcast* tenemos? ¿Y cuantos dominios de colisión?

Comprueba también las tablas de direccionamiento MAC, observando a través de qué interfaz son capaces de acceder a los diferentes PC. ¿Es óptimo el camino que siguen todos los paquetes entre todos los orígenes y todos los destinos?

2.14.2. STP

Como se puede observar, desde el primer momento en que se pasa a simulación uno de los enlaces entre los *switches* ha dejado de estar activo. Eso es consecuencia del STP que utiliza el simulador, aunque el algoritmo que emplea no es exactamente idéntico al real. Este algoritmo se activa automáticamente.

Es necesario aclarar que en los sistemas reales existen comandos específicos que permiten ver información directa sobre STP, y que su convergencia aparece en las interfaces indicando el estado en que se encuentran.

2.14.3. Protocolo CDP

Es interesante observar el resultado de CDP en los *switches* y la información que se intercambian. Recuerda que puede ser necesario configurarlo previamente. Fíjate en los *switches* que ve cada uno.

2.14.4. Seguridad en la capa 2

Para finalizar el ejercicio se verá una de las características configurables en los *switches* y que permite ofrecer "seguridad" y "control" en nuestras redes. Para ello se va a emplear el comando `port-security`. Este comando está asociado a cada interfaz y permite fijar las direcciones MAC válidas que se pueden conectar a un puerto determinado, o también limitar el número de equipos que puede haber en un puerto concreto. Además permite establecer el procedimiento que se debe seguir en caso que se conecte un equipo más de los permitidos o con una dirección MAC no autorizada.

Debes configurar la interfaz FastEthernet 2/0 del Switch0 utilizando esta característica. En un caso filtra por dirección MAC y en otro por número de dispositivos.

El proceso a realizar se describe a continuación:

- En primer lugar accede al modo de configuración del *switch*. Una vez en configuración, accede a la interfaz en la que se quiere aplicar el filtrado.
- Una vez en la interfaz, activa la seguridad del puerto (`port-security`) según los criterios elegidos:
  - Filtrar por dirección MAC:
    `Switch#switchport port-security mac-address N.N.N`
  - Filtrar por número de equipos:
    `Switch#switchport port-security maximum N`
  - Establecer el comportamiento en caso de violación:
    `Switch#switchport port-security violation shutdown`
- Finalmente, activa ese filtrado en la interfaz correspondiente.
  `Switch#switchport port-security`

Ve a simulación y verifica que se ha comportado como esperabas cuando has introducido algún equipo no autorizado.

2.15. Ejemplo de despliegue

Dado el despliegue que se muestra en la Figura 2.6, en el que dos *switches* Cisco Catalyst LAN se encuentran unidos por tres enlaces, dos de ellos a 100 Mbps y uno a 1 Gbps, y dada la siguiente información, obtenida de la línea de comandos de los equipos:

```
SW1#show spanning-tree active          SW2#show spanning-tree active
Interface                              Interface
Name              Cost     Sts         Name              Cost     Sts
----------------- -------- ----------  ----------------- -------- ----------
Fa0/10            19       FWD         Fa0/10            19       BLK
Fa0/11            19       FWD         Fa0/11            19       BLK
Gi0/1             4        FWD         Gi0/1             4        FWD

SW1#show spanning tree
VLAN0001
 Spanning tree enabled protocol ieee
 Root ID Priority 32769
    Address 000a.b79e.7640
    This bridge is the root
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

```
SW2#show spanning tree
VLAN0001
 Spanning tree enabled protocol ieee
 Root ID Priority 32769
    Address 000a.b79e.7640
    Cost 4 Port 26 (GigabitEthernet0/2)
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)
    Address 000a.b79e.a700
    Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
    Aging Time 300
```

- ¿Por qué finaliza la ejecución del STP con esas decisiones?
- ¿Cómo se podría modificar el resultado de la ejecución y seleccionar como raíz de la topología al otro *switch*? ¿Qué comando se utilizaría?

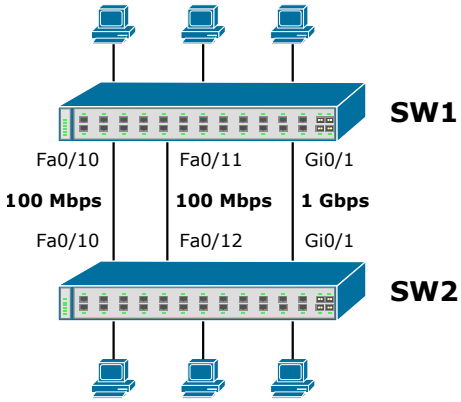
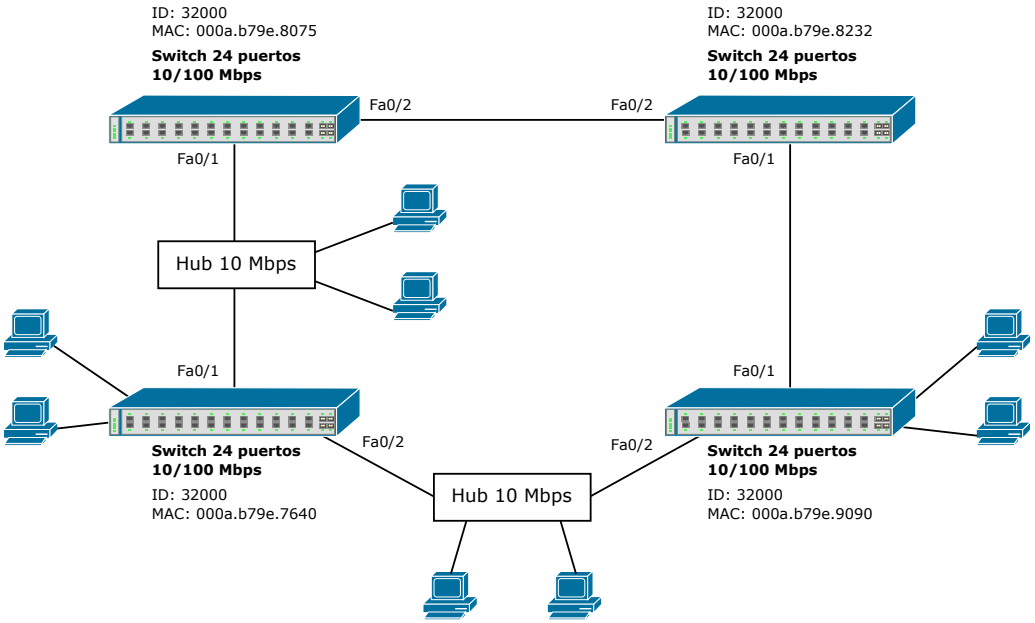


Figura 2.6: Despliegue de ejemplo.

2.16. Cálculo de STP

En el despliegue de red local de la figura. ¿Cuál sería el resultado de la ejecución del STP?



¿Considera adecuado el resultado? En caso de que detecte algún problema, ¿Cómo lo solucionaría?

3

Conmutación LAN avanzada

3.1. Objetivos

La aparición de los conmutadores LAN (*switches*) permite reducir las colisiones en una red local, segmentándola en múltiples dominios de colisión. Sin embargo, sin mecanismos adicionales, todos los equipos de la red local pertenecen al mismo dominio de difusión, con todas las limitaciones que esto conlleva en cuanto a número de equipos conectados y seguridad. En esta práctica se introducirán el concepto de LAN virtual (VLAN) que ayuda a superar dichas limitaciones.

3.2. Introducción

En una red local, todos los equipos conectados a un mismo *hub* pertenecen a lo que se conoce como dominio de colisión. La utilización de un *switch* LAN permite segmentar la red en múltiples dominios de colisión, mejorando las prestaciones que obtienen los equipos conectados a la red.

Pese a esta mejora en las prestaciones, con los mecanismos analizados hasta ahora, todos los equipos conectados a un *switch*, o a un grupo de ellos interconectados entre sí, pertenece al mismo dominio de difusión (o dominio de *broadcast*). Dicho de otra forma, el tráfico de *broadcast* generado por uno de los equipos alcanzará a todo el resto de los equipos de la red.

La primera implicación negativa de esto es la limitación en cuanto a las prestaciones que se pueden alcanzar en la red. El consumo de ancho de banda provocado por el tráfico de *broadcast* en una red local, no debería superar el 20 % del ancho de banda nominal de la red. A medida que el número de equipos conectados a la red se incrementa, aumenta el tráfico de *broadcast* generado, pudiendo llegar, si se incrementa excesivamente el número de equipos conectados, a provocar una caída significativa en las prestaciones. El tráfico de *broadcast* es utilizado, por ejemplo, por el protocolo ARP (Address Resolution Protocol), por el protocolo DHCP (Dynamic Host Configuration Protocol), por el protocolo NetBios (sistemas operativos Microsoft),... Debido a esto, el número máximo de equipos que debería conectarse a un mismo dominio de *broadcast* está limitado.

La segunda implicación es el nivel de seguridad de la red. En una red local, el tráfico de *broadcast* será recibido por todos los equipos de la red, permitiendo a un usuario malintencionado detectar actividad de otros equipos de la red con el objetivo de iniciar un posible ataque o simplemente analizar sus actividades. Asimismo, el proceso de aprendizaje de direcciones de un conmutador LAN provoca que los primeros paquetes dirigidos a un cierto destino lleguen a todos los equipos de la red. Finalmente, los equipos pertenecientes a una misma red local podrían comunicarse a nivel de capa 2 directamente, limitando las posibilidades de implantar medidas de seguridad en la red.

3.3. LAN Virtual (VLAN)

Una LAN virtual se puede definir como un dominio de difusión lógico que permite unir equipos situados en diferentes segmentos físicos de una red local. Todos los equipos de un misma VLAN pertenecerán a un dominio de difusión propio, diferente al de las otras VLAN del equipo, independientemente de su localización física. Ahora, los equipos pueden agruparse por su función en la red, por el departamento al que pertenezcan, por el proyecto en el que participen. De esta forma, estamos dividiendo una red agrupando a los equipos por funciones en la red, no por su ubicación física, mejorando la seguridad y las prestaciones. Ahora, cada VLAN se comporta como un *switch* lógico diferente.

Como puede verse en la Figura 3.1, los equipos conectados a bocas del *switch* asignadas a la misma VLAN, comparten dominio de *broadcast* independientemente de la localización física de los equipos. Asimismo, mediante la segmentación en VLAN conseguimos modificar la distribución lógica de la red independientemente de la distribución física. Si asignamos el puerto del SW2 al que esta conectado el PC2 a la VLAN 4, el equipo pasará a compartir dominio de difusión, o lo que es lo mismo, red local, con el departamento de facturación en lugar de con el departamento de sistemas. Esta es otra de las ventajas que ofrecen la segmentación en VLAN: la flexibilidad.

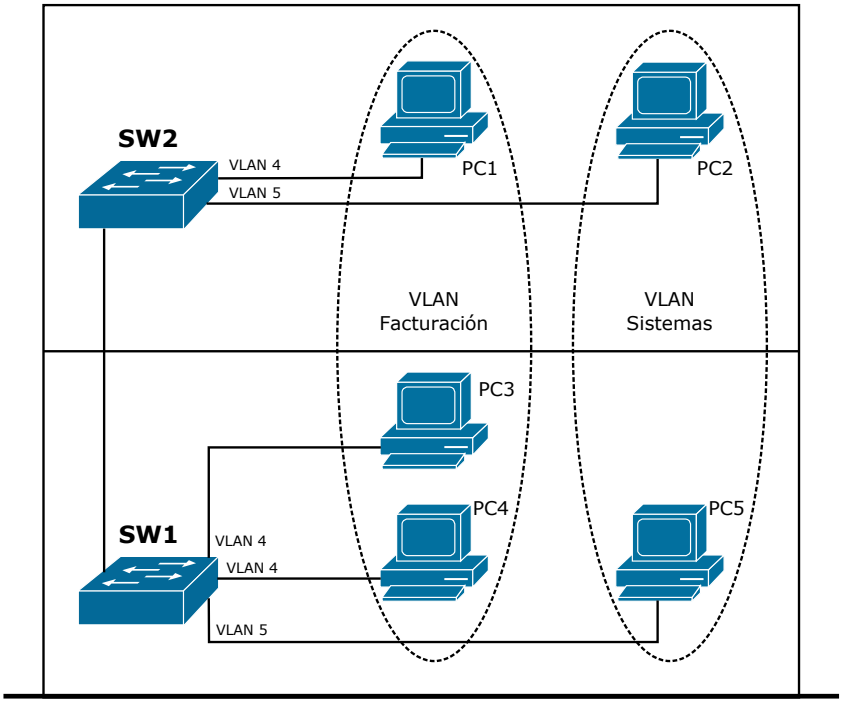


Figura 3.1: División en VLAN.

3.4. Mecanismos de funcionamiento de las VLAN

Cada VLAN, independientemente de la localización física de las interfaces que la forman, se comporta como un *switch* físico independiente. De esta forma, dentro de cada VLAN se implementa por separado:

- Mecanismo de aprendizaje de direcciones.
- Decisiones de filtrado o reenvío.
- Mecanismos para evitar bucles (STP). Dado que una VLAN se comporta como un *switch* lógico, dentro de cada VLAN debe ejecutarse una instancia diferente del protocolo STP.

Una trama originada en un puerto perteneciente a una determinada VLAN solo será retransmitida hacia un puerto perteneciente a la misma VLAN. De esta forma, queda limitado el alcance tanto del tráfico *unicast* como *multicast* o *broadcast*.

3.5. Puertos troncales

Una infraestructura de red local genérica, como la de la Figura 3.1, estará formada por más de un *switch* LAN. Para permitir que la definición de VLAN abarque más de un *switch*, hay que implementar algún mecanismo que permita diferenciar el tráfico de las diferentes VLAN.

En la Figura 3.1 puede verse como los dos *switches* están unidos por un solo enlace troncal. Sobre este enlace troncal se debe cursar el tráfico de todas y cada una de las VLAN, no solamente de una como en el caso de los puertos a los que están conectados los equipos finales de usuario. En esta situación, para mantener los mecanismos de funcionamiento de las diferentes VLAN, y que una trama sólo llegue a aquellos puertos que pertenecen a la misma VLAN por la que se emitió, debemos marcar la pertenencia de cada trama a cada una de las VLAN. A este proceso de marcado se le denomina comúnmente etiquetado o encapsulado de tramas, y a los puertos de los enlaces troncales sobre los que viaja la definición de múltiples VLAN, puertos troncales. En la Figura 3.2 puede observarse un ejemplo de estos mecanismos:

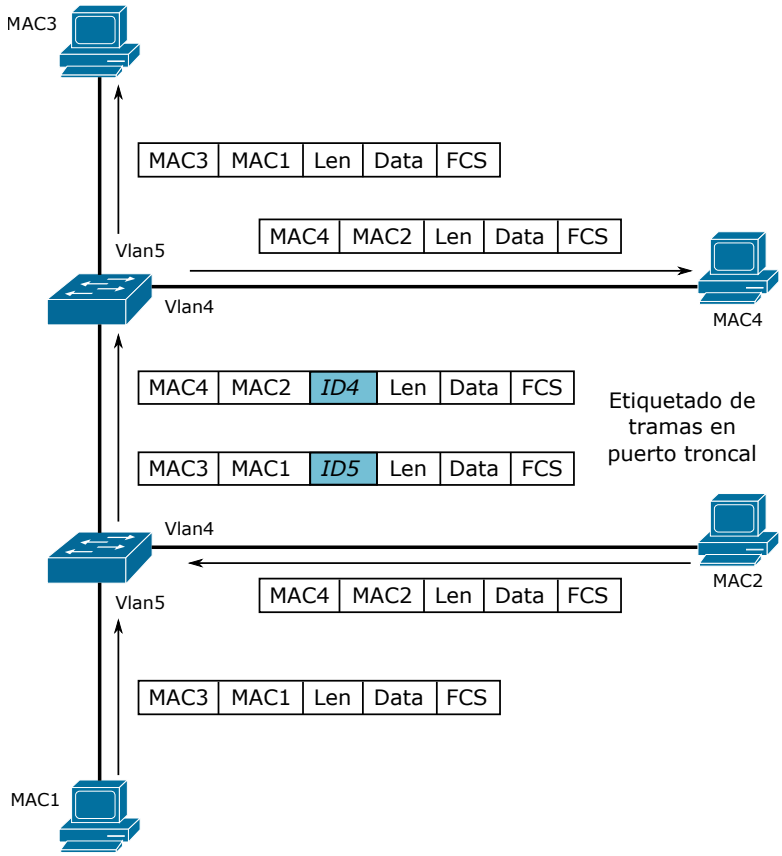


Figura 3.2: Puertos troncales y etiquetado de tramas.

A la hora de etiquetar tramas podemos seleccionar diversos métodos. Si trabajamos con equipos Cisco, los dos principales son:

1. **ISL** (Inter-Switch Link). Es un protocolo propietario de Cisco que encapsula las tramas Ethernet dentro de una cabecera ISL de 26 bytes y un campo CRC de 4 bytes. Dentro de los 20 bytes de la cabecera ISL, uno de los campos contiene el identificador de la VLAN a la que pertenece la trama. El problema principal de ISL es el hecho de ser un protocolo propietario de Cisco, lo que impide utilizarlo cuando se interconectan equipos de diferentes fabricantes.

2. Etiquetado **IEEE 802.1q**. Es un protocolo de etiquetado estándar desarrollado por el IEEE, que permite el etiquetado de tramas entre equipos de diferentes fabricantes. Se basa en la introducción de nuevos campos entre la cabecera Ethernet y el campo de datos, tal y como se ve en la Figura 3.3.

Definiendo las VLAN adecuadas en cada *switch* de la infraestructura de red, asignado interfaces del *switch* a cada una de las VLAN, y definiendo de forma adecuada los mecanismos de etiquetado de

tramas, permitimos extender el alcance de las VLAN, permitiendo que los equipos de una misma VLAN compartan dominio de *broadcast* independientemente de su localización física.

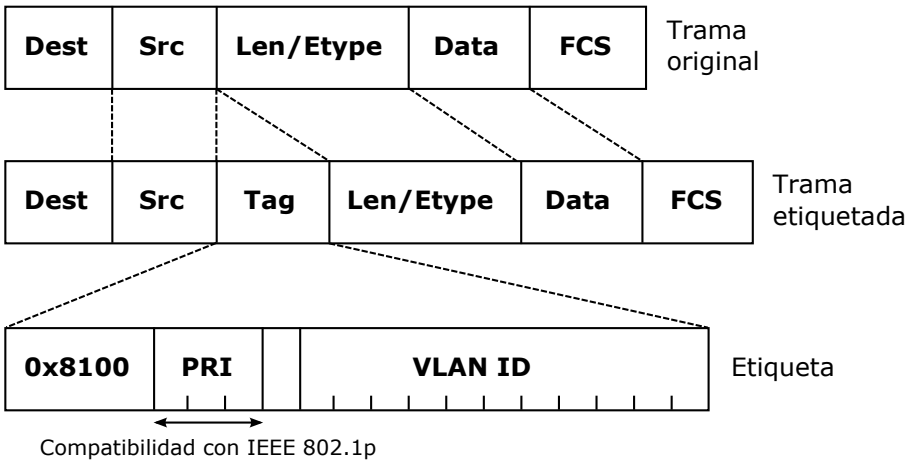


Figura 3.3: Etiquetado de tramas IEEE 802.1q.

3.6. VTP (VLAN Trunking Protocol)

Gracias a los mecanismos analizados con anterioridad, es posible extender la definición de VLAN a múltiples equipos de una red. En esta situación es necesario mantener, de forma manual, la definición de VLAN en cada equipo de la red y la consistencia en estas definiciones. En una red como la de la Figura 3.4, en la que existen múltiples dispositivos de interconexión, esto es una tarea compleja y que puede llevar a numerosas fuentes de inconsistencias (VLAN eliminadas sólo en una parte de la red, segmentos o departamentos diferentes identificados por la misma VLAN, etc.). Para solucionar todos estos problemas surge el protocolo VTP.

VTP se basa en mantener en un punto centralizado de la red (servidor VTP) las definiciones de todas las VLAN de cara a mantener la consistencia. Las actualizaciones VTP se transmiten de forma automática desde el servidor mediante los enlaces troncales entre *switches*.

3.7. Modos de funcionamiento de VTP

VTP se basa en la definición de dominios, servidores, clientes VTP y equipos transparentes. Para que un equipo comience a intercambiar información VTP es necesario añadirlo a un dominio VTP.

Por defecto, un equipo Cisco se encontrará en modo VTP servidor. En un equipo servidor VTP, un administrador podrá crear, modificar o suprimir VLAN. Cuando se detecta un cambio en la definición de las VLAN, este cambio de transmitirá hacia todos los equipos del dominio VTP mediante los enlaces troncales. En el equipo servidor, la información sobre las VLAN se almacena en RAM no volátil (NVRAM).

En un equipo que funcione en modo cliente no es posible crear, modificar ni suprimir VLAN. Este equipo sincroniza la definición de sus VLAN con el equipo servidor. En estos equipos, la definición de VLAN no se almacena en NVRAM.

Un *switch* que se configure en modo transparente no procesará los mensajes VTP recibidos, simplemente reenviará los mensajes del mismo dominio VTP. En un equipo en modo transparente se pueden añadir, modificar o eliminar definiciones de VLAN, pero dichas definiciones no se transmitirán hacia otros equipos de la red.

Los mensajes VTP se transportan sobre tráfico de *broadcast* e incorporan el nombre de dominio VTP, una contraseña opcional y un número de revisión. Si este número de revisión es más alto que el de la configuración almacenada, indica que la información recibida es más reciente, en cuyo caso debe sustituirse la información almacenada por la recibida.

3.8. Filtrado de tráfico VTP

Los enlaces troncales han de transportar el tráfico de todas las VLAN. Sin embargo, es posible que existan situaciones en las que un *switch* reciba tráfico de *broadcast* dirigido a una VLAN en la que no tenga ninguna interfaz. Para eliminar este tráfico en los puntos de la red donde no sea necesario, y aumentar así las prestaciones en la red, surge el filtrado de tráfico VTP.

En la Figura 3.4 puede observarse un ejemplo de funcionamiento de filtrado de tráfico VTP. El tráfico de *broadcast* originado en la VLAN Y sólo consumirá ancho de banda en los enlaces troncales que conduzcan a otros *switches* que tengan alguna interfaz perteneciente a la misma VLAN.

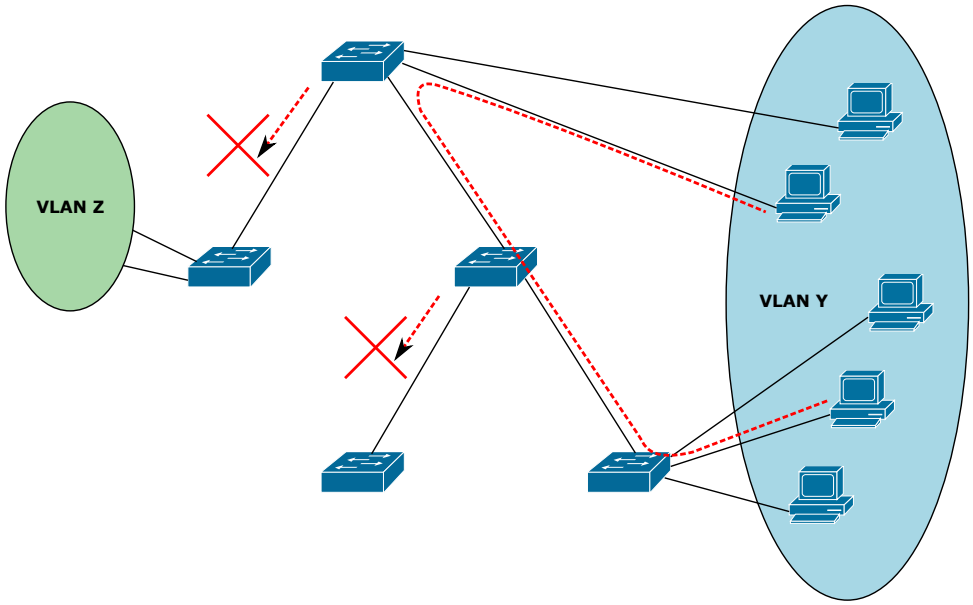


Figura 3.4: Filtrado de tráfico VTP.

3.9. Comunicación entre VLAN

Dado que cada VLAN es un dominio de *broadcast* diferenciado, y que un equipo perteneciente a una VLAN sólo puede enviar tráfico a la misma VLAN, hemos de habilitar un mecanismo de una capa superior que permita la comunicación segura entre VLAN. Una VLAN a todos los efectos se comporta como una red local independiente; una red con capacidad de comunicar los equipos en capa 2.

Para intercomunicar VLAN diferentes, la alternativa más sencilla es recurrir a un *router* que las interconecte. En este caso, cada VLAN utilizará un rango de direcciones diferente. Existirá un *router* con tantas interfaces como VLAN existan en nuestra red. Como puede verse en la Figura 3.5, cada una de las interfaces del *router* estará conectada a una interfaz del *switch* perteneciente a una VLAN diferente. Obviamente, el *router* tendrá asignada, en cada una de sus interfaces, una dirección IP del mismo rango que la VLAN a la que esté interconectado.

Dado que un *router* es un equipo especializado en encaminar tráfico entre diferentes redes IP, simplemente configurando el direccionamiento IP de sus interfaces ya será capaz de encaminar tráfico entre las redes a las que pertenecen las direcciones IP asignadas a sus interfaces. Es decir, sólo conocerá las redes a las que está directamente conectado, pero este conocimiento es suficiente para encaminar tráfico entre ellas.

Esta opción, conceptualmente sencilla, no es aplicable en el caso de una división en un número de VLAN elevado, y en ese caso deberemos recurrir a una infraestructura de *routers* más compleja. Entonces ya no será suficiente con el conocimiento local de cada uno de los *routers*, y deberá habilitarse un mecanismo de encaminamiento entre ellos. Existen otras soluciones, como la división en subinterfaces en el *router* o la utilización de *switches* de capa 3 que integran capacidades de encaminamiento IP, pero no se tratarán en esta práctica.

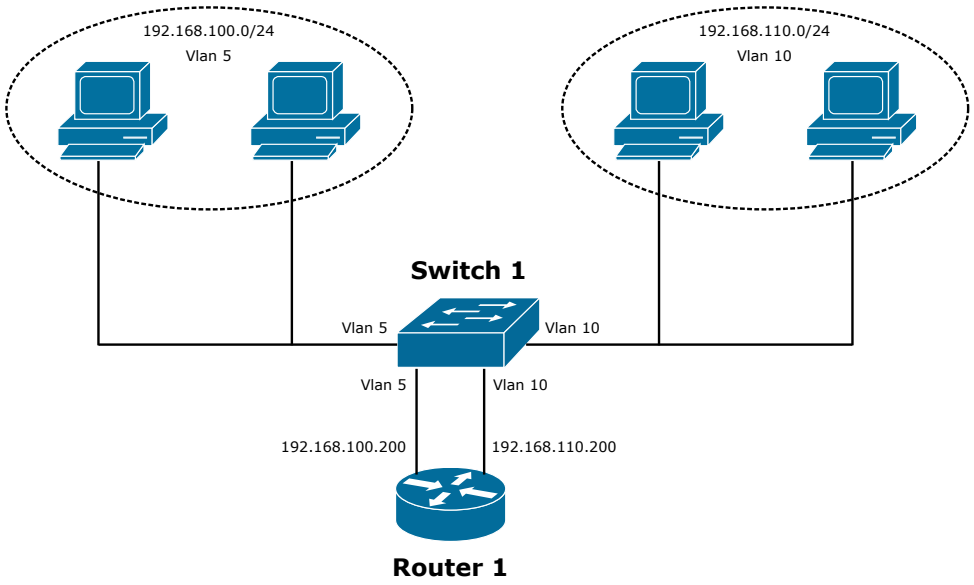


Figura 3.5: Interconexión de VLAN mediante *router*.

3.10. Otras opciones de trunking. Dynamic Trunking Protocol

Hemos visto como, en un caso genérico, el administrador ha de decidir cuáles de los puertos de un *switch* van a actuar como puertos de acceso o como puertos de interconexión con otro *switch* y por tanto tratar tramas etiquetadas. Para facilitar la gestión de la red los fabricantes de equipos han diseñado múltiples protocolos propietarios.

En el caso de Cisco, ha desarrollado el protocolo Dynamic Trunking Protocol, habilitado por defecto en equipos como los Catayst 2950. Dicho protocolo permite que dos *switches* reconozcan dinámicamente, sin intervención del administrador, qué puertos son troncales y negocien el tipo de etiquetado que van a utilizar.

3.11. Configuración de VLAN en Cisco IOS

1. Creación de una VLAN. Debe realizarse en dos fases:

   a) Desde el modo de configuración, dar de alta la VLAN con el siguiente comando.
   ```
   Switch(config)#vlan NUM
   ```

   b) Asignar una dirección IP y una descripción a una VLAN (todos sus puertos tendrán asignada la misma IP para poder gestionar el equipo) :
   ```
   Switch(config)#interface vlan NUM
   Switch(config-if)#ip address DIR_IP MÁSCARA
   Switch(config-if)#description 'TEXTO DESCRIPTIVO'
   ```

2. Asignación de puertos a las VLAN (en el modo de configuración de una interfaz):
   ```
   Switch(config)#interface Type Num/Mod
   Switch(config-if)#switchport access vlan NUM
   ```

3. Definición de los puertos troncales (en el modo de configuración de la interfaz a establecer como puerto troncal):
   ```
   Switch(config)#interface Type Num/Mod
   ```
   !! Establecer el puerto como troncal de unión entre *switches*
   ```
   Switch(config-if)#switchport mode trunk
   ```
   !! En un equipo real deberíamos establecer el tipo de etiquetado de tramas, pero en el simulador no es posible
   ```
   Switch(config-if)#switchport trunk encapsulation [isl/dot11q]
   ```
   !! Añadir VLAN a la lista de admitidas en el *trunking*
   ```
   Switch(config-if)#switchport trunk allowed vlan add NUM
   ```

!! Permitir algunas VLAN/todas/todas salvo algunas/eliminar una
`Switch(config-if)#switchport trunk allowed vlan add/all/except/remove NUM 4`

4. Mostrar información relacionada con las interfaces del equipo (estado, tráfico recibido, enviado, errores detectados, etc.)
   `Switch#show interfaces [Type Mod/Num] [Vlan]`
5. Mostrar información relacionada con las VLAN definidas en el equipo y la asignación de puertos:
   `Switch#show vlan`
6. Resumen con el estado de las interfaces del equipo:
   `Switch#show ip interfaces brief`

3.12. Configuración del direccionamiento IP en un router

1. Asignación de una dirección IP a una interfaz de un *router*:
   `Router(config)#inteface Type Num/Mod`
   `Router(config-if)#ip address DIR_IP MÁSCARA`
2. Resumen con el estado de las interfaces del equipo:
   `Router#show ip interfaces brief`
3. Rutas conocidas por el *router*:
   `Router#show ip route`

3.13. Ejercicio propuesto

Realice el despliegue de red mostrado en la Figura 3.6 mediante división en VLAN y definición de puertos troncales. La empresa utiliza el rango de direcciones `192.168.100.0/24`. El Departamento A y el Departamento B deben tener capacidad para direccionar 32 equipos. Hay 9 servidores corporativos.

Compruebe que todos los equipos se pueden comunicar con el resto y razone acerca del camino seguido por las tramas en su camino por la red. ¿Considera que hay algún enlace sobrante en la red? ¿Cuál es la razón?

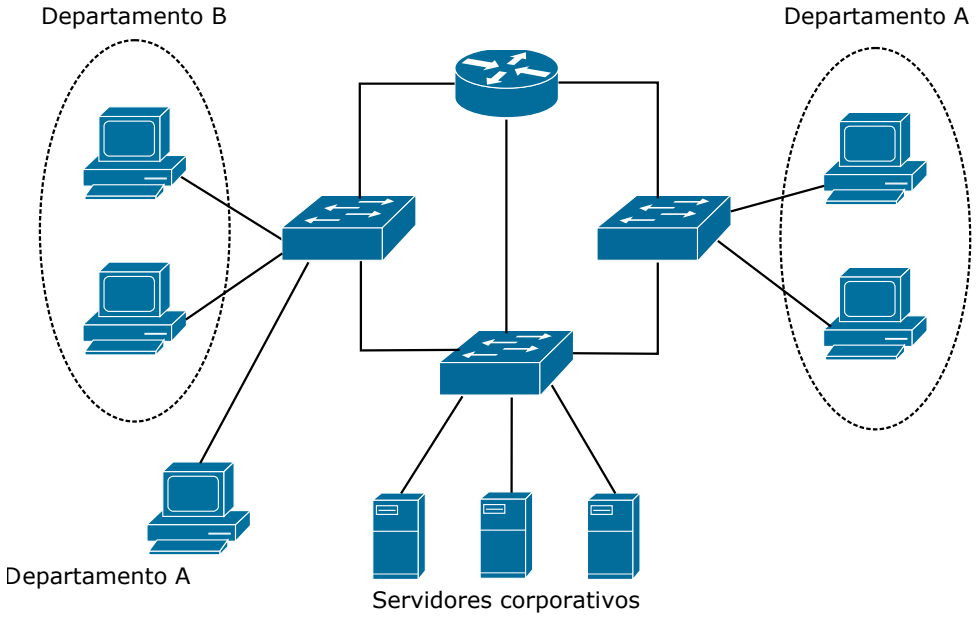


Figura 3.6: Despliegue de red del ejercicio planteado.

4

Direccionamiento IP

4.1. Objetivos

El diseño y asignación de direcciones IP a los equipos pertenecientes a una red es uno de los puntos más relevantes en la configuración de redes. Asignar adecuadamente las direcciones, optimizando su distribución, es una de las tareas más importantes y complejas de los administradores de una red, sobre todo cuando dicha red cuenta con un número elevado de dispositivos.

En esta práctica se recordarán algunos conceptos básicos sobre direccionamiento IP mediante ejercicios teóricos y prácticos. El objetivo no es enseñar estos conceptos al alumno, sino facilitar que los recuerde y aplique. Por tanto, las descripciones de este guión no son más que un complemento a lo explicado en la parte teórica de esta asignatura y de asignaturas anteriores.

4.2. Direcciones IP

Una dirección IP es un identificador único que se asigna a cada dispositivo en Internet o en una red TCP/IP. Es una dirección lógica que el administrador de red asigna a cada uno de los equipos de la red.

La dirección IP está formada por 32 bits (4 bytes), que permiten identificar la red a la que pertenece el equipo en cuestión y al equipo dentro de dicha red. Por tanto, está dividida en dos partes:

- **Netid**. Identifica a la red. Debe asignarlo un organismo oficial si la red va a formar parte de Internet, y se utiliza para el encaminamiento.
- **Hostid**. Identifica al dispositivo (*host*) dentro de la red. Lo asigna el administrador de la red.

La representación de la dirección IP no suele hacerse mediante bits, sino mediante una codificación en decimal de los 4 bytes que la forman separados por puntos:

```
192.168.100.2
```

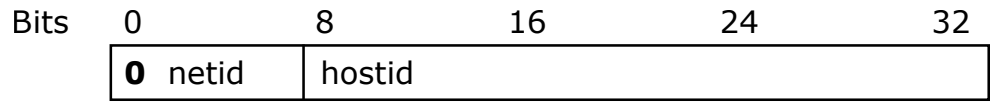
Como se ha dicho, una parte de dicha dirección identifica a la red y la otra al *host*:

$$\underbrace{\mathbf{192}\ .\ \mathbf{168}\ .\ \mathbf{100}\ .}_{NetId}\quad\underbrace{\mathbf{2}}_{HostId}$$

Inicialmente, las direcciones IP se dividieron en clases para permitir que redes de diferente tamaño dispusiesen de un número de direcciones distinto.

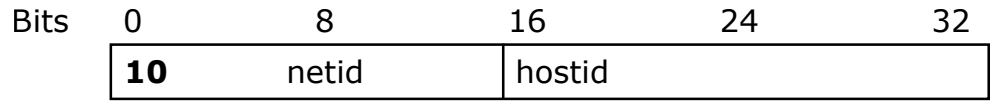
Los administradores de red decidieron realizar una clasificación de las direcciones en función del número de bits que se utilizan para fijar el identificador de red (*netid*), formando así las clases. Bajo esta premisa se ha realizado una división en 5 clases:

- **Clase A**. Dispone de un total de 126 redes con más de 16 millones de *hosts* en cada red.



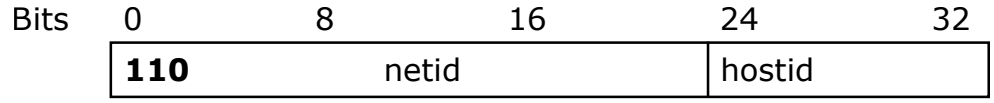
  El valor del primer byte, es decir del *netid*, está dentro del rango: 1-126.
  El primer y último valor para el *netid* en esta clase serían el identificador 0 ($netid_{bits}$= 00000000) y 127 ($netid_{bits}$= 01111111) respectivamente. No obstante, estos dos valores tienen un significado especial y no se asignan a ninguna red. El *netid* con valor 0 identifica a "esta red", es decir, la red a la que el *host* está conectado físicamente. El *netid* con valor 127 es la dirección de *loopback*. Cuando un equipo manda un paquete a esta dirección, dicho paquete no llega a la red sino que atraviesa descendentemente la arquitectura de protocolos y luego ascendentemente. Se suele emplear para comprobar que no hay ningún error en la implementación de la arquitectura de protocolos del equipo.

- **Clase B**. Dispone de más de 16000 redes con 65534 *hosts* en cada red.



  El valor del primer byte del *netid* está dentro del rango: 128-191.

- **Clase C**. Dispone de más de 2 millones redes con tan sólo 254 *hosts* en cada red.



  El valor del primer byte del *netid* está dentro del rango: 192-223.

- **Clase D**. Empleada para direcciones de multidifusión (*multicast*).
- **Clase E**. Empleada en investigación.

En este punto es necesario destacar algunos aspectos no especificados en la división en clases, como por ejemplo el hecho de que en todas las redes hay dos valores del *hostid* que no se pueden emplear por ser direcciones reservadas:

- Todos los bits del *hostid* tienen valor 0. Esa dirección IP con el *hostid* a cero identificaría a la red en su totalidad, no a un equipo.
- Todos los bits del *hostid* tienen valor 1. Esa dirección IP sería la de difusión (*broadcast*) de la red.

También es importante destacar que existen varios rangos de direcciones que están reservados para su uso en redes privadas (intranets), y cuyo empleo no es válido en Internet, quedando reducida su utilización a dichas redes privadas. Una máquina conectada directamente a Internet no debe emplear ninguna de estas direcciones. Estas redes reservadas son:

- Red de clase A. **10.0.0.0**
- Redes de clase B. **De 172.16.0.0 a 176.31.0.0**
- Redes de clase C. De **192.168.0.0 a 192.168.255.0**

La asignación de una clase a una organización para crear una red es algo complejo, sobre todo cuando el número de dispositivos a interconectar es relativamente grande. Por ello, en muchas ocasiones las redes se dividen en subredes que permiten mejorar aspectos de asignación y gestión de las mismas.

4.3. Otros conceptos: asignación dinámica

En la mayoría de las redes son los administradores de red los encargados de configurar y gestionar el direccionamiento IP. En redes con un número elevado de usuarios este proceso puede resultar tedioso y largo. Una forma de facilitarlo es utilizar algún procedimiento que automatice la asignación de direcciones y la configuración de los equipos.

Por otra parte, la escasez de direcciones IP también ha llevado a que los proveedores de acceso a Internet (ISP) empleen técnicas de asignación dinámica de direcciones IP en las conexiones contratadas por sus usuarios. Esto facilita la gestión del direccionamiento, asignando direcciones únicamente a los equipos que las necesitan en un momento determinado y no a todos los clientes.

Una de las técnicas empleadas para la asignación dinámica de direcciones IP a los equipos de una red consiste en utilizar el protocolo DHCP (Dynamic Host Configuration Protocol). Para poder utilizar este protocolo se requiere un servidor, que será el encargado de la asignación de direcciones, por lo que es necesario indicarle previamente el rango de direcciones disponibles en la red. Los equipos cliente deben tener activada la opción de asignación dinámica, para que al iniciarse en la red soliciten al servidor DHCP una dirección IP con la que poder comunicarse. Será el servidor quien, en función de diversos parámetros, devuelva a los clientes la configuración de red básica que tienen que emplear en la red, como dirección IP, máscara de subred, puerta de enlace (*gateway*), servidor de DNS, etc. En la actualidad, la mayor parte de los *routers* del mercado tienen incorporada la funcionalidad de servidor DHCP.

4.4. Ejercicio teórico

Se dispone de la red `192.168.16.0/20`, que se quiere dividir en subredes para formar la red que se muestra en la Figura 4.1:

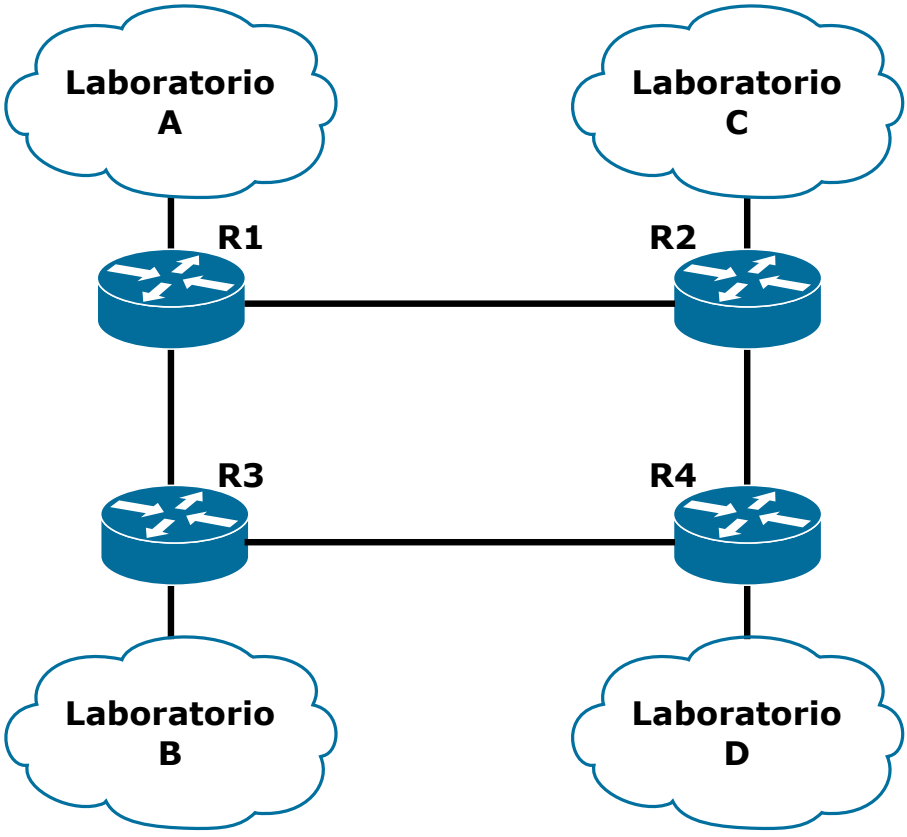


Figura 4.1: Esquema de red del ejercicio.

El laboratorio A va a necesitar espacio para 300 direcciones IP. El laboratorio D va a necesitar 120 direcciones IP y los laboratorios B y C sólo 30 direcciones IP cada uno. Aplicando VLSM, calcula las direcciones de red y la máscara de subred de cada una de las subredes que es necesario crear.

4.5. Simulación de conexión entre redes LAN

Generalmente las redes LAN no se suelen montar aisladas y se requiere la comunicación con otras redes. La tarea de este apartado consistirá en el montaje de una conexión típica entre redes LAN, implementando la red del ejercicio teórico en el simulador *Packet Tracer*. Como se trata de redes que se encuentran en el mismo edificio será posible conectarlas mediante *routers* sin tener que pedir licencias ni contratar circuitos a terceros.

Como se ve en la Figura 4.1, las cuatro redes están interconectadas entre sí mediante 4 *routers*. Para la simulación se va a considerar que las LAN de los laboratorios están formadas por un *switch* y varios equipos: 3 PC en las redes más grandes (laboratorios A y D) y 2 PC en las más pequeñas (laboratorios B y C).

Se deben configurar adecuadamente tanto los equipos como los *routers*, de acuerdo a los resultados obtenidos en el ejercicio teórico, para que sea posible la comunicación entre todas las redes, prestando especial atención a las direcciones IP que se asignan a cada una de las interfaces de los *routers*.

Finalmente se debe verificar el correcto funcionamiento de la red.

5

Encaminamiento estático

5.1. Objetivos

El objetivo de esta práctica es que el alumno adquiera los conocimientos necesarios para la configuración del encaminamiento estático en una red de comunicaciones. Para ello, primeramente se describirán los mecanismos básicos del encaminamiento entre redes. Una vez adquiridos estos conocimientos, el alumno podrá analizar las limitaciones obvias del encaminamiento estático y la necesidad de recurrir a encaminamiento dinámico como solución general para el intercambio de información entre redes. Como parte final se describirá el procedimiento de configuración del encaminamiento estático en una red formada por varios *routers* Cisco.

5.2. Introducción

En la práctica anterior se ha analizado el caso simplificado de un *router* conectado directamente a varias redes. En este caso, el *router*, mediante el análisis del direccionamiento IP asignado en sus interfaces (dirección IP y máscara), obtiene las redes a las que está directamente conectado y por tanto es capaz de encaminar tráfico entre ellas.

En una situación genérica, una red no estará formada por un único *router* que esté directamente conectado a todas las redes a las que debe cursar tráfico, sino que estará formada por varios *routers*, interconectados entre ellos, uniendo equipos distribuidos por toda la red. En la Figura 5.1 podemos ver un ejemplo. El router R2 tendrá configurada en la interfaz i0 una dirección del rango `192.168.100.32/28`, en la i1 una dirección del rango `192.168.100.46/28` y en la i3 una dirección del rango `192.168.100.132/30`. Por tanto, mediante su conocimiento local de la red, sabe cómo alcanzar esas tres redes y es capaz de encaminar tráfico entre ellas. Su tabla de rutas, como puede verse en la Figura 5.2, estará formada por tres entradas, una para cada destino conocido. Este *router*, cuando reciba un paquete, buscará si la dirección IP de destino coincide con alguna de las entradas de su tabla de rutas. Si coincide con alguna, procesará el paquete y lo enviará por la interfaz indicada en la tabla de rutas hacia el destino adecuado.

Lo mismo ocurre con R3 y las redes `192.168.100.132/30` y `192.168.100.128/30`, y con el *router* R1 y las redes `192.168.100.16/28` y `192.168.100.128/30`. Sin embargo, sin habilitar mecanismos adicionales, R2 desconoce la existencia de la red `192.168.100.16/28` y desconocerá cómo hacerle llegar un paquete.

Con el objetivo de superar estas limitaciones, tenemos dos opciones:

- Gestionar de forma manual el conocimiento de la red del que dispone cada equipo.
- Habilitar un mecanismo de encaminamiento dinámico.

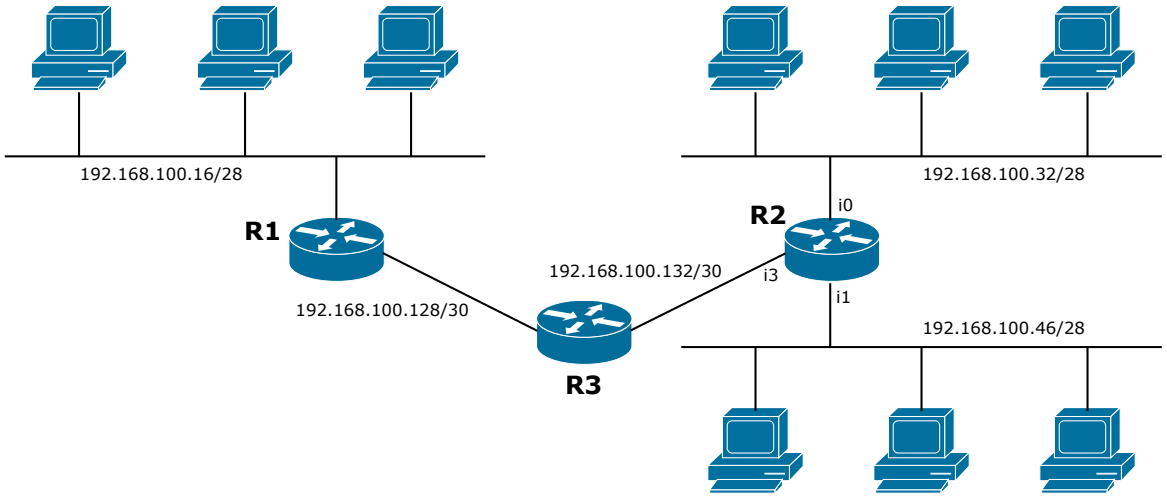


Figura 5.1: Interconexión de VLAN mediante Router.

```
R2#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set

      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.32/28 is directly connected, FastEthernet0/0
C        192.168.100.46/28 is directly connected, FastEthernet1/0
C        192.168.100.132/28 is directly connected, FastEthernet3/0
R2#
```

Figura 5.2: Tabla de rutas de R2 cuando sólo está configurado el direccionamiento IP.

5.3. Encaminamiento estático

El encaminamiento estático se basa en que el administrador, de forma manual, introduzca las rutas hacia todos los destinos necesarios, indicando para cada destino cuál es el siguiente salto a nivel IP. En el ejemplo de la Figura 5.3, podemos ver cómo, con dos rutas estáticas, los equipos finales de ambos extremos de la red pueden comunicarse entre ellos. En el *router* de la izquierda hay una ruta estática indicando que, para llegar a la red `192.168.102.0/24`, hay que utilizar como siguiente salto el equipo con dirección IP `192.168.101.2`.

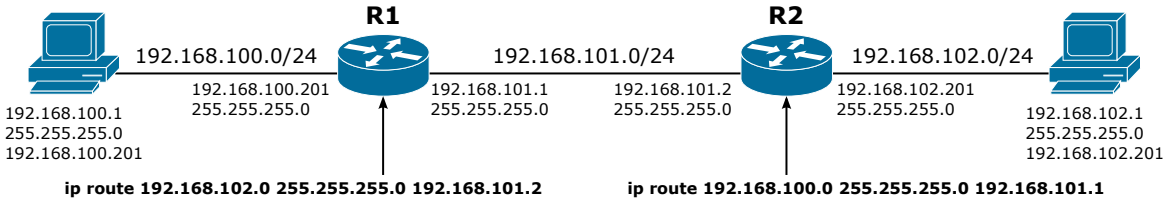


Figura 5.3: Rutas estáticas.

Repitiendo este proceso para cada ruta necesaria es posible habilitar el encaminamiento hacia todos los destinos que se desee alcanzar. Hay que indicar que, en una infraestructura de red más compleja, para habilitar una ruta desde el *router* origen hasta el *router* final, habría que ir habilitándola salto a salto. No es suficiente con indicar en el *router* origen cual es el siguiente salto, también es necesario indicarlo en los *routers* intermedios.

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set

C        192.168.100.0/24 is directly connected, FastEthernet0/0
C        192.168.101.0/24 is directly connected, FastEthernet1/0
S        192.168.102.0/24 [1/0] via 192.168.101.2
R1#
```

Figura 5.4: Tabla de rutas con una ruta estática.

5.4. Ruta predeterminada

En muchas ocasiones, hay un grupo elevado de rutas hacia las que se viaja por un camino común. Este es el caso, por ejemplo, del R1 de la Figura 5.5. Existe un número limitado de rutas internas a la red, y al resto de los destinos se accederá a través de un *router* del operador que provea la línea de acceso.

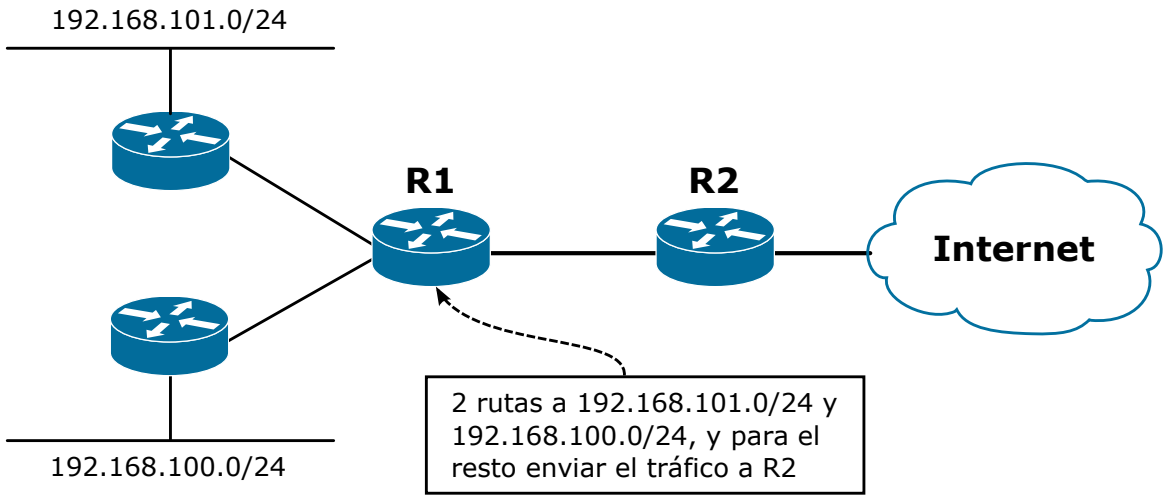


Figura 5.5: Ejemplo de la utilidad de una ruta predeterminada.

Para evitar, en casos como el anterior, tener que dar de alta un número muy elevado de rutas, se puede recurrir a lo que se conoce como ruta predeterminada. Esta entrada en la tabla de rutas significa que cualquier paquete que el *router* no sepa cómo encaminar con una entrada concreta de su tabla, debe enviarla hacia el equipo que se le indique en la misma. De esta forma, en el caso de la Figura 5.5, R1 podría reducir su tabla a tres entradas: 2 rutas a `192.168.100.0/24` y `192.168.101.0/24` y una entrada de ruta predeterminada apuntando a R2. Una posible tabla de rutas para R1 puede verse en la Figura 5.6. Marcada como "*Gateway of last resort*" o como destino `0.0.0.0/0` aparece la entrada de ruta predeterminada apuntando a un equipo de la red.

5.5. Configuración del encaminamiento estático en Cisco IOS

Definición de una ruta estática. Puede hacerse de dos formas:

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is 192.168.69.26 to network 0.0.0.0

     192.168.69.0/24 is subnetted, 3 subnets
C       192.168.69.16/30 is directly connected, FastEthernet0/0
C       192.168.69.20/30 is directly connected, FastEthernet1/0
C       192.168.69.24/30 is directly connected, FastEthernet2/0
S    192.168.100.0/24 [1/0] via 192.168.69.18
S    192.168.101.0/24 [1/0] via 192.168.69.22
S*   0.0.0.0/0 [1/0] via 192.168.69.26
R1#
```

Figura 5.6: Posible configuración de R1.

1. Indicando la dirección IP del siguiente dispositivo que debe procesar el paquete en su tránsito a través de la red
   `Router(config)#ip route RED MÁSCARA IP-SIG-SALTO`
2. Indicando la interfaz por la cual se debe emitir el paquete
   `Router(config)#ip route RED MÁSCARA INTERFAZ`

Definición de la ruta predeterminada:

```
Router(config)#ip route 0.0.0.0 0.0.0.0 IP-PUERTA-ENLACE
```

Examen de la tabla de rutas:

```
Router#show ip route
```

5.6. Ejercicio propuesto 1

En el esquema de red de la Figura 5.7, habilitar los mecanismos de encaminamiento necesarios para que:

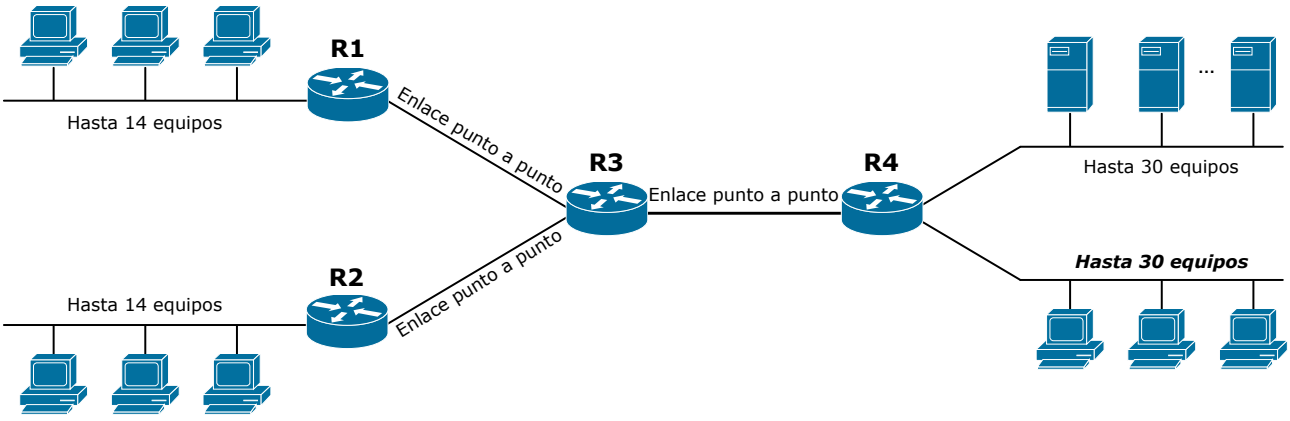


Figura 5.7: Esquema de red para el ejercicio propuesto número 1.

- Cada *router* disponga de una tabla de rutas con el menor número de entradas posible.
- Todas las redes deben tener capacidad para intercambiar tráfico con las restantes redes del despliegue, salvo la red marcada en cursiva y negrita, que sólo debe poder intercambiar tráfico

con los servidores. Sólo manipulando las rutas definidas en cada uno de los equipos, ¿es posible cumplir plenamente el objetivo marcado?

- En todas las redes, la capacidad ("Hasta n equipos") incluye al *router* de salida.

5.7. Ejercicio propuesto 2

En el despliegue siguiente, ¿en qué puntos de la red resultaría especialmente útil la definición de una ruta predeterminada en lugar de una entrada por cada red alcanzable? ¿Por qué?

Si sobre el mismo despliegue no se utilizase en ningún punto ruta predeterminada, ¿qué modificaciones supondría añadir una nueva red dependiente de R1?

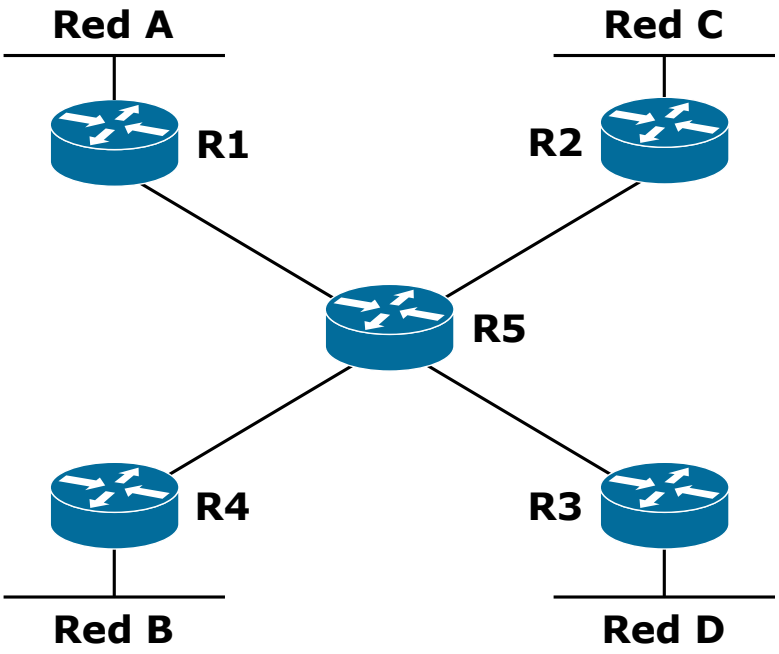


Figura 5.8: Esquema de red para el ejercicio propuesto número 2.

6

Encaminamiento dinámico. RIP

6.1. Objetivos

El objetivo último de esta práctica es que el alumno adquiera los conocimientos necesarios para la configuración de un algoritmo de encaminamiento dinámico en una red de comunicaciones. Para ello se describirán los diferentes tipos de protocolos de encaminamiento y se analizará en detalle el funcionamiento de RIP, uno de los protocolos de encaminamiento más sencillo y casi universalmente soportado por los equipos de interconexión de nivel 3. Como parte final se describirá el procedimiento de configuración de RIP en una red formada por varios *routers* Cisco.

6.2. Introducción

En la práctica anterior hemos visto los fundamentos del encaminamiento IP y hemos descrito el procedimiento de habilitación del encaminamiento estático en una red de comunicaciones. Como se pudo comprobar, el encaminamiento estático requiere que un administrador decida cuál es el mejor camino para los paquetes (en caso de que exista más de uno) y lo configure de forma manual en cada uno de los equipos. Entre las principales desventajas de esta forma de trabajar están las siguientes:

- Es inmanejable en redes de tamaño medio o grande.
- Cualquier cambio de topología supone la intervención manual del administrador de cara a introducir/modificar esa nueva ruta. Esta situación se produce tanto al añadir nuevos enlaces en la red como ante caídas en los mismos; si un enlace desaparece de la red, las rutas que atravesasen dicho destino quedan todas inhabilitadas hasta que el administrador intervenga e introduzca una nueva ruta.
- No se aprovechan posibles caminos alternativos, ya que todos los paquetes siguen la ruta establecida por el administrador.

De cara a resolver todos estos problemas, y hacer viable la explotación de una red de tamaño medio/grande, surgen los protocolos de encaminamiento dinámico. Estos protocolos se basan en el traspaso de conocimiento entre los diferentes *routers* de una red: cada *router* transmite hacía otros *routers* de la red información con su conocimiento local de la red. Dicho de otra forma, si B sabe como llegar a A, y B le dice a C que para llegar a A debe pasar por B, C sabe como llegar a A.

Un *router*, mediante su conocimiento local del entorno y la información recibida de los otros equipos de la red, calculará una distancia a cada destino en la red. En función de esta distancia calculada decidirá cuál es el mejor destino a una red dada.

Mediante la utilización de una algoritmo de encaminamiento dinámico, el administrador sólo debe realizar una tarea inicial de configuración en la que habilita un algoritmo de encaminamiento

determinado y configura qué rutas, de las que conoce el equipo configurado, anunciará hacia al resto de los *routers*. Gracias al intercambio de información entre los *routers*, periódico o cuando se produce un cambio en la topología en la red, la red se adapta a cambios sin la intervención del administrador.

6.3. Tipos de protocolos de encaminamiento

Podemos establecer diversas clasificaciones de los protocolos de encaminamiento en función de una gran multitud de factores, entre los que podemos destacar el tipo de métrica utilizada para calcular la distancia a un destino dado, y si el algoritmo tiene como objetivo encaminar de forma interna o externa a un sistema autónomo.

6.3.1. Interno/Externo

Una primera clasificación de los protocolos de encaminamiento puede realizarse en función de si el objetivo del algoritmo es tomar decisiones de encaminamiento entre los equipos de un mismo sistema autónomo o entre sistemas autónomos diferentes.

Un sistema autónomo no es más que un conjunto de *routers* bajo el mismo control administrativo, y que por tanto siguen una política común de explotación de la red. Es decir, forman la red de un determinado operador o corporación.

Un algoritmo cuyo objetivo sea el encaminamiento dentro de un sistema autónomo deberá ser capaz de gestionar un número no excesivamente elevado de rutas, con lo cual la velocidad de convergencia debe ser elevada. Además, no guarda memoria sobre cada una de las rutas calculadas, solamente conoce un coste para llegar a un destino determinado y se selecciona la mejor de las rutas.

Por el contrario, los protocolos destinados a encaminar entre sistemas autónomos deben ser capaces de gestionar cantidades muy elevadas de rutas y deben guardar memoria de los caminos atravesados (qué sistemas autónomos se atraviesan para llegar a un determinado destino). De esta forma es posible establecer acuerdos entre operadores, asignar prioridades a las rutas y emplear toda una serie de métricas adicionales en función del camino atravesado.

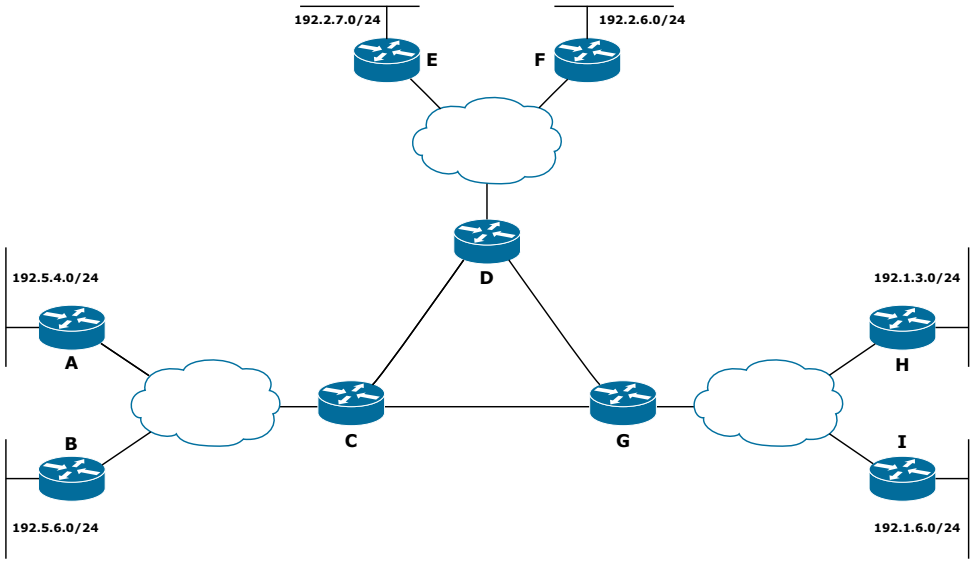
6.3.2. Vector de distancias/Estado de enlace

Los protocolos de vector de distancias se basan en que cada *router* de la red envíe a sus vecinos un resumen de cada prefijo que conoce y la distancia estimada hasta ese destino. La distancia puede estar medida, por ejemplo, en número de saltos hacia un destino. Por el mecanismo de funcionamiento de estos protocolos, en los que un cambio detectado por un equipo en un punto de la red se transmite a los vecinos, los cuales tras procesar el cambio, en la siguiente ejecución, transmitirán este cambio a su vez hacia sus vecinos, estos protocolos presentan las siguientes características:

- Convergen lentamente. Observando el esquema de red de la Figura 6.1 podemos comprobar como la aparición de nuevos enlaces se propaga rápidamente (a un salto por ejecución). Sin embargo, sin mecanismos adicionales, la caída de un enlace se propaga de una forma mucho más lenta. Durante el proceso de aprendizaje, por parte de todos los *routers* de la red, de la imposibilidad de alcanzar un destino, conocido como "cuenta hasta infinito" se producen incongruencias en las tablas de encaminamiento. Existen técnicas, como la de horizonte dividido (*split horizon*), que permiten evitar la lentitud del proceso de cuenta hasta infinito. En dicha técnica la distancia a un destino se envía como infinito sobre el enlace seleccionado como siguiente salto para ese destino. Sin embargo, también tiene sus limitaciones y existen topologías en las que no evita la cuenta hasta infinito y las incongruencias en las tablas de encaminamiento.
- Los diferentes equipos de la red pueden disponer de diferentes mapas de la misma o, incluso, de mapas inconsistentes.

Los protocolos de estado de enlace se basan en que cada *router* anuncia hacia todos los otros equipos de la red cada vecino que tiene y el estado de la línea que lo une con él. La información transmitida en el enlace es mucha menos que en el caso de los protocolos de vector de distancias (menor consumo de ancho de banda), supone el procesado de menos información (menor consumo de recursos en los *routers*) y además convergen mucho más rápido que los protocolos de vector de distancias.

En los protocolos de estado de enlace, todos los equipos de la red acaban contando con un mapa topológico exacto de la estructura de la red, no como en el caso anterior que solo conocen a sus vecinos y las rutas que estos conocen; en este tipo de protocolos, un *router* dado anuncia hacia todos los otros equipos de la red cuales son sus vecinos y cuál es el estado del enlace hacia ellos.



Ejecución	A	B	C	D	E	F	G	H	I
1	∞	∞	∞	∞	∞	∞	∞	∞	1
2	∞	∞	∞	∞	∞	∞	2,I	2,I	1
3	∞	∞	3,G	3,G	∞	∞	2,I	2,I	1
4	4,C	4,C	3,G	3,G	4,D	4,D	2,I	2,I	1

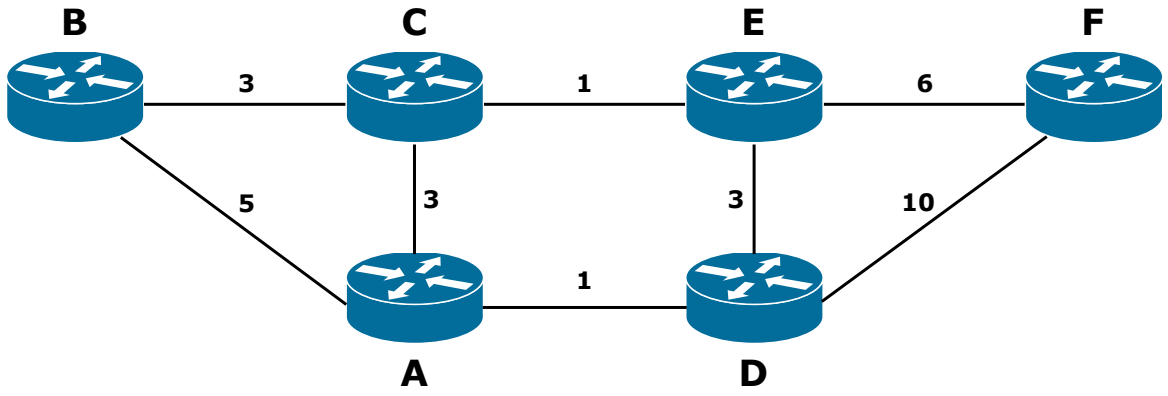
Figura 6.1: Ejemplo de propagación de buenas noticias en protocolos de vector de distancias.

Ejecución	A	B	C	D	E	F	G	H	I
1	4,C	4,C	3,G	3,G	4,D	4,D	2,I	2,I	∞
2	4,C	4,C	3,G	3,G	4,D	4,D	3,H	3,G	3,G
3	4,C	4,C	4,G	4,G	4,D	4,D	4,H	4,G	4,G
4	5,C	5,C	5,G	5,G	5,D	5,D	5,H	5,G	5,G
5	6,C	6,C	6,G	6,G	6,D	6,D	6,H	6,G	6,G
6	7,C	7,C	7,G	7,G	7,D	7,D	7,H	7,G	7,G
...	...	...	...	...	...	...	...	...	...
15	15,C	15,C	15,G	15,G	15,D	15,D	15,H	15,G	15,G
16	∞	∞	∞	∞	∞	∞	∞	∞	∞

Figura 6.2: Ejemplo de propagación de malas noticias en protocolos de vector de distancias (caída del enlace hacia la red I).

Como ejemplo de algoritmo de estado de enlace podemos mencionar por su gran extensión y sus funcionalidades OSPF (Open Shortest Path First). Los protocolos de estado de enlace ejecutan lo

que se conoce como algoritmo de Dijsktra. Un ejemplo de ejecución de este algoritmo puede verse en la Figura 6.3. Como ejemplo de algoritmo de vector de distancias podemos mencionar, por su simplicidad y extensión, RIP (Routing Information Protocol). Este tipo de protocolos ejecutan lo que se conoce como algoritmo Bellman-Ford distribuido.



Iteración	SPF	Lista de candidatos
1	A	D→(D,1) C→(C,3) B→(B,5)
2	D	C→(C,3) E→(D,4) B→(B,5) F→(D,11)
3	C	E→(D,C,4) B→(B,5) F→(D,11)
4	E	B→(B,5) F→(D→E,10)
5	B	F→(D→E,10)
6	F	vacía

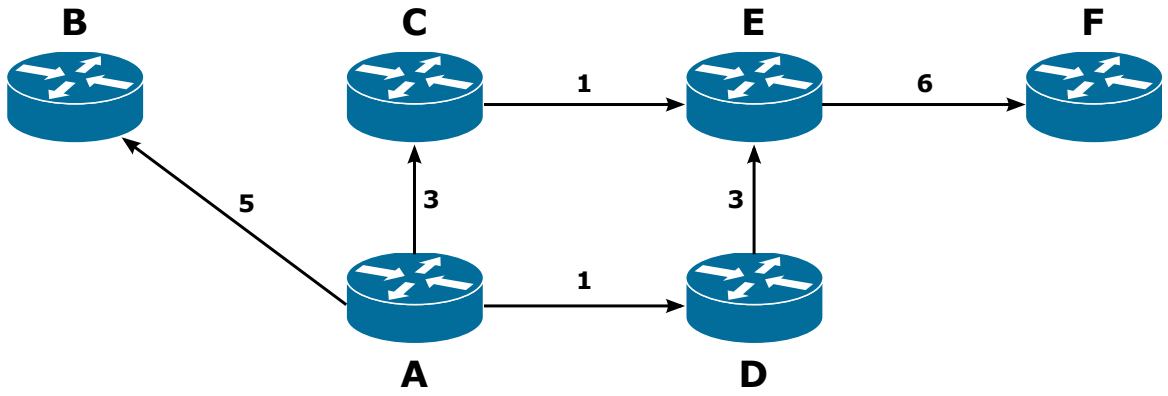


Figura 6.3: Ejemplo de cálculo del algoritmo de Dijsktra.

6.4. Encaminamiento basado en clases/sin clases

Otra clasificación de los protocolos de encaminamiento puede realizarse en función de si incluyen o no la máscara de red en los anuncios.

A aquellos protocolos de encaminamiento que no incluyen máscara de red en los anuncios se los conoce como de encaminamiento basado en clases (*classful*). Al no incluir la máscara de red en los anuncios se ven obligados a dar por supuesta una estructura uniforme en la división del direccionamiento IP en la red. Estos protocolos de encaminamiento suponen lo siguiente al recibir un anuncio de red:

- Si el anuncio pertenece a una red que forma parte de la misma clase que la interfaz por la que se recibe, se le aplica la misma máscara que tenga la interfaz. Debido a esto no soportan el trabajo con máscaras de red de tamaño variable.
- Si el anuncio pertenece a una clase diferente a la de la dirección IP por la que se recibe el anuncio, se le aplica la máscara de su clase.

En este tipo de protocolos la sumarización de rutas es automática. Al transmitir un anuncio más allá del limite de nuestra red y aplicarle la máscara de clase, los anuncios de múltiples sub-redes de una clase se reducen a un solo anuncio de la clase.

Los protocolos de encaminamiento que sí incluyen la máscara de red en sus anuncios se denominan de encaminamiento sin clases (*classless*). Estos protocolos permiten utilizar máscaras de red de tamaño variable (VLSM; Variable Lenght Subnet Masking) y en ellos la sumarización de rutas es manual, es decir, el administrador decide cuándo debe realizarse a la vista de la estructura de la red.

6.5. RIPv1

RIPv1 es uno de los protocolos de encaminamiento más extendidos y más sencillos. Es un algoritmo de vector de distancias cuya métrica es el número de saltos hacia un destino dado.

Las redes directamente alcanzables por un *router* se marcan como de coste 0. Los vecinos se considera que están a distancia 1. La distancia mayor que puede existir entre dos puntos de nuestra red (diámetro de la red) es de 15 ya que 16 se ha elegido como valor para los destinos inalcanzables.

RIP, como todos los protocolos de encaminamiento dinámicos, se basa en el intercambio de información entre los *routers* de la red. Los mensajes intercambiados viajan sobre tráfico UDP de *broadcast* destinado a la dirección `255.255.255.255` y al puerto `520`. Este intercambio de información tiene lugar cada 30 segundos o cada vez que sucede un evento significativo (una interfaz de un *router* se desactiva o se activa, se recibe una actualización de un *router*, una entrada agota su tiempo de vida).

Un *router* RIP, en su versión predeterminada, si no recibe actualizaciones hacia una destino dado en 90 segundos, actualiza el siguiente salto a cualquier *router* que anuncie la misma o menor métrica. Si a los 180 segundos no recibe actualizaciones, marca el destino como inalcanzable.

6.5.1. Formato de paquete

Un paquete IP RIPv1 tiene un tamaño de 512 bytes y está formado por 9 campos:

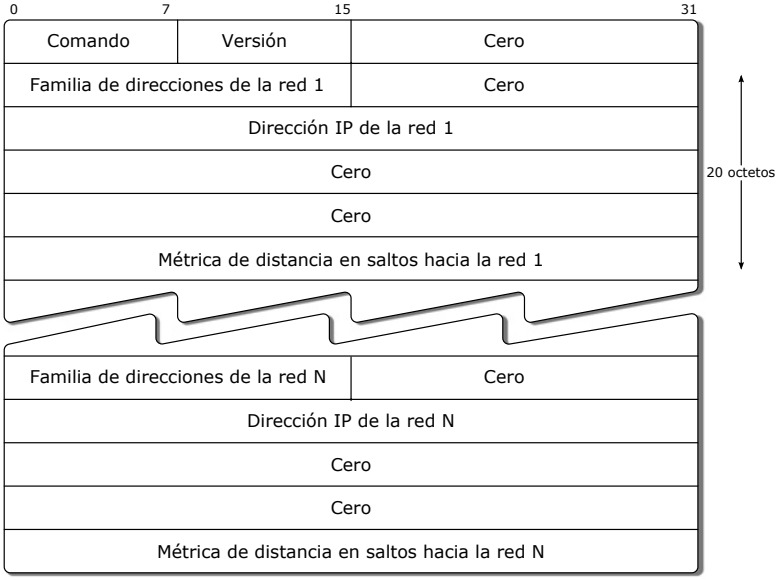


Figura 6.4: Paquete RIPv1.

- **Comando** (1 octeto) Indica si el paquete es una solicitud (de otro *router* preguntando por parte de la tabla de rutas) o una respuesta (una actualización periódica de la tabla de rutas o una respuesta).
- **Versión** (1 octeto).

- **Cero** (2 octetos). No utilizado en la versión 1 (RFC 1058). Compatibilidad con versiones pre estándar de RIP.
- **Identificador de familia de direcciones** (AFI; Address-family identifier) (2 octetos). RIP admite la transmisión de información sobre rutas para múltiples protocolos. En el caso del protocolo IP el AFI es 2.
- **Cero** (2 octetos).
- **Dirección** (4 octetos). Dirección de destino para esa entrada.
- **Cero** (4 octetos).
- **Cero** (4 octetos).
- **Métrica** (4 octetos). Indica cuántos saltos (*routers* intermedios) hay que dar hasta un destino dado. Este valor está entre 1 y 15 para una ruta válida o 16 para una ruta inalcanzable.

Los campos *AFI*, *Dirección* y *Métrica* se pueden repetir hasta 25 veces en un mismo paquete RIP que anuncie destinos IP.

Como podemos observar, RIPv1 no incluye la máscara de red en los anuncios, razón por la cual es un protocolo de encaminamiento basado en clases que no permite el trabajo con máscaras de red de tamaño variable.

6.5.2. Virtudes y defectos

Como ventajas del protocolo RIPv1 podemos destacar las siguientes:

- Es sencillo de configurar.
- Está disponible prácticamente en cualquier *router* del mercado.
- Admite reparto de carga entre, como máximo, 5 rutas. Este reparto de carga se realiza entre todas las líneas hacia un destino por las que se haya calculado el mismo coste.

Como desventajas del uso de este algoritmo podemos mencionar:

- RIPv1 no soporta VLSM, con las limitaciones que esto impone en la división de los rangos de direccionamiento, obligando a trabajar con subredes de tamaño idéntico.
- RIPv1 no soporta autentificación de los mensajes. Los mensajes del protocolo viajan en abierto (sin cifrar) sobre tráfico de *broadcast*. Cualquier equipo de la red (incluido un equipo final de usuario) puede comenzar a introducir mensajes UDP destinados a la dirección `255.255.255.255` y puerto `520` con el objetivo de crear inconsistencias en las tablas de rutas o redirigir el tráfico a través de él.
- Convergencia lenta al igual que todos los algoritmos de vector de distancias. Esto provoca que no sea adecuado en redes grandes (con diámetro siempre menor o igual que 15) o en redes que sufran muchos cambios, ya que la lentitud de convergencia puede provocar que el algoritmo no converja nunca.
- Métrica excesivamente sencilla que no permite, por ejemplo, tener en cuenta el ancho de banda de las líneas ocupadas.
- Reparto de carga sólo efectivo entre líneas de igual ancho de banda
    - *pinhole congestion*

6.6. RIPv2

A la vista de las limitaciones en cuanto a la división del direccionamiento IP y a los problemas de seguridad, se desarrolló una nueva versión de RIP, RIPv2.

RIPv2 es un protocolo de encaminamiento basado en vector de distancias, al igual que RIPv1. Las mejoras que introduce son las siguientes:

- Introduce las máscara de red en los anuncios.
- El intercambio de mensajes entre *routers* tiene lugar sobre tráfico *multicast* al grupo `224.0.0.9`.
- Implementa autentificación entre *routers* basada en contraseña.

6.6.1. Formato de paquete

Un paquete IP RIPv2 consta de los siguientes campos:

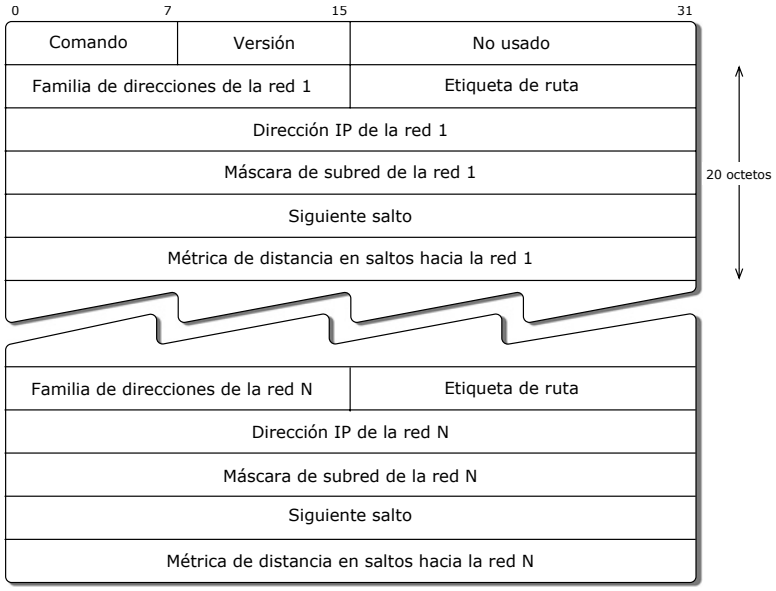


Figura 6.5: Paquete RIPv2.

1. **Comando** (1 octeto).
2. **Versión** (1 octeto). Si implementa cualquiera de los campos de RIPv2 o autentificación este valor es igual a 2.
3. **No usado** (2 octetos).
4. **Identificador de familia de direcciones** (2 octetos). Igual que en RIPv1 salvo una excepción:
   - Si `AFI=0xFFFF` el resto del paquete es información de autentificación. En las versiones estándar sólo implementa contraseña.
5. **Etiqueta de ruta** (2 octetos). Permite distinguir entre rutas internas (aprendidas por RIP) y externas (anunciadas por otros protocolos)
6. **Dirección IP** (4 octetos).
7. **Máscara de subred** (4 octetos). Si este campo es cero no se transmite ninguna máscara
8. **Siguiente salto** (4 octetos). Indica la dirección IP del siguiente salto al que debe enviarse el paquete.
9. **Métrica** (4 octetos). Indica el número de saltos (*routers*) hasta el destino.

6.7. Configuración de RIPv1/v2

1. Habilitar RIP como protocolo de encaminamiento:
   `router rip`
2. Cambio de la versión de RIP:
   `version [1|2]`
3. Asociación de redes con el algoritmo de encaminamiento RIP. Debemos incluir aquí las redes que queremos que el *router* incluya en sus anuncios hacia otros *routers* y las redes a las que pertenezcan las interfaces por las que el *router* queremos que envíe y reciba anuncios de RIP:
   `network A.B.C.D`
4. Definición de una interfaz pasiva. Esto permite definir interfaces sobre las que el protocolo no enviará ni recibirá actualizaciones de rutas del protocolo RIP. Aumenta la seguridad del protocolo evitando enviar/recibir actualizaciones sobre interfaces a las que, por ejemplo, sólo haya equipos de usuario conectados.
   `passive-interface Tipo Módulo/Número`
5. Deshabilitación de la autosumarización:
   `no auto-summary`
6. Comprobar el estado de la tabla de rutas:
   `show ip route`
7. Verificar el protocolo de encaminamiento que se está ejecutando en un *router* [No disponible en el simulador]:
   `show ip protocol`
8. Mostrar la información relacionada con las actualizaciones enviadas/recibidas en un *router* en el que se ejecute RIP [No disponible en el simulador]:
   `debug ip rip`

NOTA: En el proceso normal de depuración del funcionamiento de un equipo de interconexión, tras observar el contenido del fichero de configuración en ejecución (`show running-config`), se pueden detectar errores en la configuración de uno o varios elementos. Para anular un comando tecleado anteriormente, desde modo configuración, y a su vez, dentro del sub-modo adecuado (por ejemplo, configuración de interfaz) se utiliza la palabra clave **no** seguida de todo o una parte del comando erróneo:

1. Anular la dirección ip asignada a una interfaz:
   `no ip address`
2. Anular una ruta estática:
   `no ip route IP MASCARA`
   `no ip route IP MASCARA SIG_SALTO`
3. Anular RIP como protocolo de encaminamiento:
   `no router rip`
4. Anular la definición de un anuncio de red:
   `no network A.B.C.D`
5. Anular la definición de una interfaz pasiva:
   `no passive-interface Tipo Mod/Num`

6.8. Ejercicio propuesto 1

1. En el esquema de red de la Figura 6.6, habilite el encaminamiento mediante el protocolo RIP. Debe configurar el direccionamiento IP de todos los equipos y comprobar que se dispone de conectividad desde todos los puntos de la red. ¿Qué versión de RIP ha de utilizar? ¿por qué?

2. ¿A que direcciones de la red puede hacer `ping`? Tenga en cuenta equipos finales de usuario y equipos de interconexión. ¿Por qué? Compárelo con la configuración que realizó en el caso de la práctica de direccionamiento estático.

3. ¿Puede conseguir que los equipos de la red `192.168.100.128/27` sólo se comuniquen con los de la red `192.168.100.64/27`?

4. Manteniendo el tamaño de las redes y utilizando RIPv1, utilice el direccionamiento IP que crea conveniente para poder encaminar paquetes entre todos los equipos de usuario. ¿Considera eficiente este segundo despliegue?

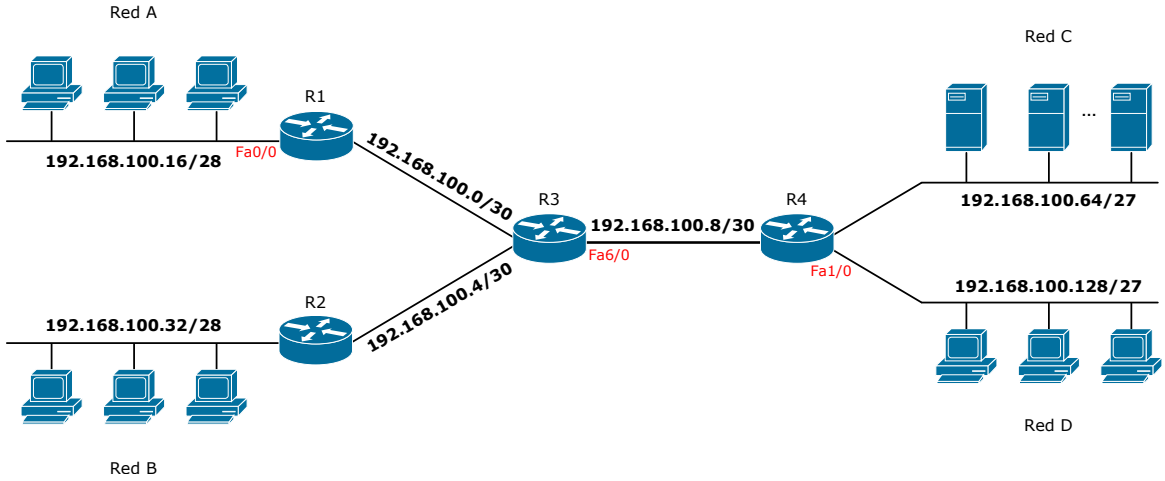


Figura 6.6: Esquema de red ejercicio propuesto.

6.9. Ejercicio propuesto 2

Si se utiliza encaminamiento RIP en la siguiente red, ¿qué modificaciones supondría añadir una nueva red dependiente de R1?

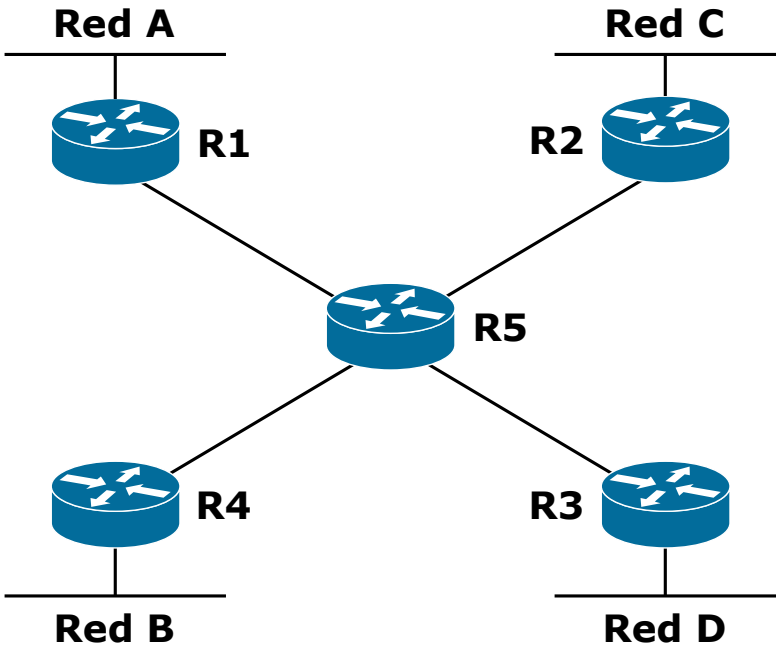


Figura 6.7: Esquema de red para el ejercicio propuesto número 2.

7

Control de tráfico. Listas de acceso

7.1. Objetivos

El objetivo de esta práctica es la toma de contacto con las técnicas de control de tráfico basadas en listas de acceso, uno de los mecanismos más extendidos y utilizados en las redes de comunicaciones. Se analizarán sus mecanismos de funcionamiento, las reglas básicas de diseño y sus principales virtudes y limitaciones. Finalmente se particularizará su uso para el sistema operativo IOS.

7.2. Introducción

Hasta este momento hemos tratado cómo proporcionar conectividad entre equipos y a diferentes niveles: desde entornos de red local a redes de área extensa. Sin embargo, hemos estado obviando un aspecto, la seguridad. A medida que el uso de Internet es más generalizado, surge la necesidad de proteger las redes de todo aquel tráfico que se considere dañino y de restringir las acciones que pueden realizar ciertos equipos. Como ejemplo podemos mencionar el caso de un *firewall* (cortafuegos) situado en la salida a Internet de una red corporativa. Este equipo ha de permitir que el tráfico originado en el interior de la red, y destinado a aquellos servicios que se consideren adecuados, viaje por Internet y que aquel tráfico generado como respuesta a estas peticiones sea recibido. Sin embargo, las conexiones iniciadas en Internet y con destino en el interior de la red, sólo tendrán sentido si el equipo de destino es un servidor corporativo accesible a cualquier usuario.

Uno de los mecanismos de control de tráfico más extendidos, tanto en *routers* como en *firewalls*, son las listas de acceso. Una lista de acceso no es más que un conjunto de reglas que definen qué trafico puede y cuál no atravesar un equipo de interconexión. En esta práctica trataremos el tema de las listas de acceso, describiendo los distintos tipos existentes y su funcionamiento, permitiendo definir filtros sobre el tráfico que atraviesa un determinado equipo de interconexión.

7.3. Tipos de listas de acceso

Existen, al menos, dos tipos de listas de acceso:

- Listas de acceso estándar.
  - Permite chequear el trafico IP en función de la dirección origen de los paquetes.
  - Permiten o deniegan toda la pila de protocolos que se encuentre sobre el nivel IP.
  - Utilizan identificadores entre 1 y 99 y entre 1300 y 1999.

- Listas de acceso extendidas.
  - Permite chequear el tráfico IP en función de la dirección IP origen, IP destino, protocolo utilizado y puertos origen y destino.
  - Permiten o deniegan un protocolo específico.
  - Utilizan identificadores entre 100 y 199 y entre 2000 y 2699.

Existen otros tipos de listas de acceso, como las listas de acceso basadas en nombre, que se tratarán en apartados posteriores y que son simplemente una mejora de los mecanismos de manipulación de las listas de acceso extendidas.

7.4. Sentido de aplicación de las listas de acceso

Una lista de acceso, tras definir las diferentes reglas de filtrado de las que se compone, es necesario aplicarla sobre alguna interfaz del equipo en cuestión. A la hora de aplicarla sobre una interfaz hemos de seleccionar el sentido en el que se aplica:

- Listas de acceso aplicadas a la entrada (Inbound Access List)
  - Los paquetes se chequean con la lista de acceso antes de ser enviados hacia la interfaz de salida.
  - El hecho de permitir el paquete en una lista de acceso de este tipo significa continuar su procesado.
  - Evita tener que realizar las tareas de toma de decisión en el encaminamiento.
- Listas de acceso aplicadas a la salida (Outbound Access List)
  - Se toma primero la decisión sobre la interfaz de salida y posteriormente se aplica la lista de acceso existente en sentido de salida, en caso de que exista alguna.
  - Ahora permitir significa meter el paquete en el *buffer* de salida de la interfaz en concreto.

Basándonos en el tipo de información que maneja cada uno de los tipos de listas de acceso mencionados con anterioridad para permitir o denegar el tráfico, y teniendo en cuenta que lo que estamos definiendo son filtros sobre el tráfico que intenta atravesar un equipo, podemos hacer una primera aproximación a su utilidad y limitaciones.

Las listas de acceso estándar, como sólo filtran el tráfico en función de la dirección IP origen, hemos de situarlas lo más cerca posible del destino. De otra forma, descartarán tanto el tráfico que deseemos filtrar como todo aquel originado en la misma IP. En cambio, las listas de acceso extendidas han de situarse lo más cerca posible del origen, ya que podemos filtrar el tráfico en función de origen y destino, permitiendo filtrar el tráfico no deseado originado en una cierta dirección IP y permitir el resto procedente de la misma IP.

7.5. Procesado de una lista de acceso

Como ya se mencionó con anterioridad, las listas de acceso están formadas por una serie de reglas en las que se define qué tráfico se permite y qué tráfico se deniega en una cierta interfaz.

El orden en el que se introduzcan las reglas de una lista de acceso es significativo en su funcionamiento: si un paquete encuentra un regla que lo permite o lo deniega, se aplica esa acción sin terminar de recorrer la lista de acceso. Por ejemplo, si situamos una regla genérica que permita el tránsito de todo el rango de direcciones `192.168.100.0/24` y posteriormente, en la misma lista de acceso hay una regla que deniega la dirección `192.168.100.111`, esta segunda regla no tendrá efecto, ya que en el recorrido de la lista de acceso el paquete entrará en la primera regla y se detendrá el procesado de la lista de acceso. Un ejemplo de procesado en una lista de acceso puede verse en la Figura 7.1. Cuando un paquete encaja en la definición de una regla, se acepta o se deniega su procesado, deteniendo la comprobación del resto de la lista de acceso.

Cualquier lista de acceso, al menos en equipos Cisco, dispone de una regla implícita que deniega todo el tráfico no denegado o permitido con anterioridad. Por lo tanto, cualquier lista de acceso ha de disponer de alguna regla que permita algún tipo de tráfico; en caso contrario se denegará todo.

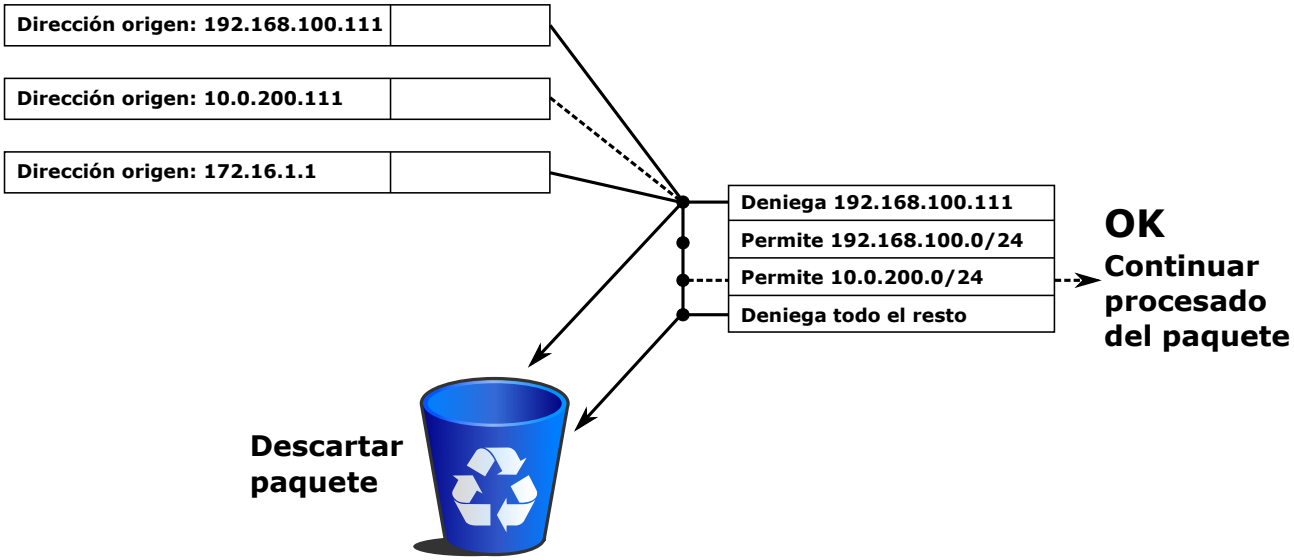


Figura 7.1: Procesado de paquetes en una lista de acceso.

7.6. Diseño de una lista de acceso

A modo de guía de diseño, la implementación de una lista de acceso podría seguir los siguientes pasos:

1. Utilizar un número del rango asignado al tipo de lista de acceso utilizado.
2. Organizar las entradas (reglas) en la lista de acceso de la siguiente forma:
   - Colocar las reglas para situaciones comunes primero.
   - Colocar primero las reglas más específicas.
3. Al final de toda lista de acceso hay una orden de denegación implícita de todo el tráfico que no cumpla una regla anterior.
   - Si queremos permitir todo el tráfico que no cumpla alguna regla hay que incluir dicha orden.
4. Debe definirse la lista de acceso antes de aplicarla a una interfaz.
   - Una lista de acceso vacía sobre una interfaz permite todo el tráfico.
5. Se pueden definir múltiples listas de acceso por interfaz pero sólo una por protocolo.
6. Ni en las listas de acceso estándar ni extendidas se puede añadir/eliminar entradas individuales.
   - Sí es posible hacerlo en las lista de acceso identificadas por nombre.
   - Para facilitar el proceso de diseño, y no cometer múltiples errores en la entrada de comandos, provocando la necesidad de borrar la lista de acceso entera y volver a dar de alta comando a comando, podemos seguir el siguiente procedimiento:
     - Diseñar la lista en cualquier editor de texto, prestando especial atención al orden de las reglas.
     - Cortar desde el editor y pegar en la línea de comandos las reglas definidas.

Además de estas reglas, hay una serie de consideraciones que deberíamos tener en cuenta a la hora de diseñar las reglas de filtrado de tráfico:

- Hay una serie de direcciones IP reservadas que, por lo general, no tiene sentido habilitar. Entre ellas podemos destacar la dirección de bucle local (`127.0.0.0/8`) y las direcciones IP *multicast* (`224.0.0.0/4`).
- Por lo general, hemos de denegar el tráfico que se intente cursar con direcciones IP privadas sobre una infraestructura pública. Los rangos de direcciones privadas son: `10.0.0.0/8`, `172.16.0.0/12` y `192.168.0.0/16`.
- Denegar tráfico del cual se sepa que tiene una dirección origen situada en otro punto de la red, por ejemplo, paquetes desde el exterior de nuestra red con dirección origen en una de las direcciones internas.
- Por lo general, a una infraestructura de comunicaciones (equipos de interconexión) sólo tiene sentido que dirijan tráfico los administradores de la misma. Es necesario controlar quién dispone de acceso a las líneas de acceso remoto a los equipos de interconexión, evitando que cualquier persona pueda ocupar una línea de `telnet` o `ssh` o intente iniciar sesión en el equipo.

Hemos de fijarnos en que, dependiendo del tipo de lista de acceso, el punto más adecuado para su aplicación será diferente. Imaginemos una infraestructura de red como la de la Figura 7.2.

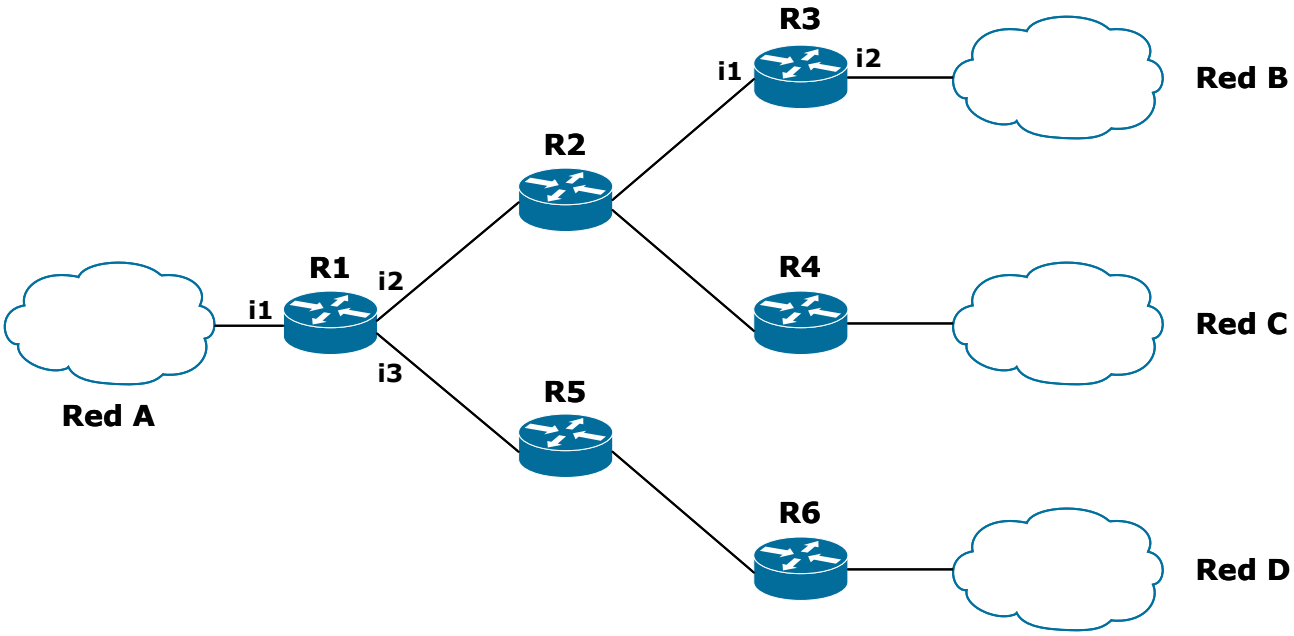


Figura 7.2: Punto de aplicación de una lista de acceso.

Si el objetivo de nuestra lista de acceso fuese filtrar el tráfico originado en la Red A y dirigido hacia la Red B, con listas de acceso estándar el punto adecuado sería la interfaz i2 del router R3 en sentido de salida, mientras que en el caso de una lista de acceso extendida, el punto adecuado sería la interfaz i1 de R1 en sentido de entrada. En otro caso, estaríamos provocando una de las dos posibles situaciones siguientes:

- Filtrando más tráfico del objetivo de la lista de acceso.
- Permitiendo circular por la red tráfico que finalmente tendrá que ser filtrado y, por tanto, siendo ineficientes en la gestión del mismo.

7.7. Configuración de listas de acceso estándar

Definición de una regla de una lista de acceso estándar:

```
Router(config)#access-list Núm-lista-acceso {permit|deny} origen máscara_comodín
```

- El número de la lista debe estar comprendido entre 1 y 99.
- La máscara comodín (*wildcard*) es equivalente a la máscara de subred, pero intercambiando el papel de los valores `1` y `0`.
- La máscara comodín predeterminada es `0.0.0.0`, que hace referencia a un único equipo y que no hace falta especificar si se emplea la palabra **host**.
- Para especificar cualquier equipo se puede emplear `0.0.0.0/255.255.255.255`, o usar la palabra **any**.
- Una misma lista de acceso puede estar formada por una o más reglas de este tipo.
- El comando `show ip interface` sirve para ver la configuración de las interfaces, incluida la información sobre las listas de acceso que tienen aplicadas.

Añadir una descripción a la lista de acceso (útil en configuraciones complejas):

```
Router(config)#access-list Núm-lista-acceso remark TEXTO
```

Aplicar la lista de acceso sobre una interfaz determinada:

```
Router(config)#interface Tipo Modulo/Número
Router(config-if)#ip access-group Núm-lista-acceso {in|out}
```

- Si no se indica nada el sentido predeterminado es de salida.

Mostrar las listas de acceso definidas en el equipo:

```
Router#show ip access-lists
```

Eliminar una lista de acceso:

```
Router(config)#interface Tipo Módulo/Número
Router(config-if)#no ip access-group Núm-lista-acceso
Router(config-if)#exit
Router(config-if)#no access-list Núm-lista-acceso
```

7.8. Configuración de listas de acceso extendidas

Definición de una regla de una lista de acceso extendida:

```
Router(config)#access-list Núm-lista-acceso {permit|deny} protocolo origen
máscara_comodín [operador puerto(s)] destino máscara_comodín [operador puerto(s)]
```

- `protocolo` puede ser `IP`, `TCP`, `UDP` o `ICMP`.
- `operador` puede ser `eq` (equal; igual), `lt` (less than; menor que), `gt` (greater than; mayor que), `neq` (not equal; distinto) o `range` (rango).
- La máscara predeterminada es `0.0.0.0`.

Añadir una descripción a la lista de acceso (útil en configuraciones complejas):

```
Router(config)#access-list Núm-lista-acceso remark TEXTO
```

Aplicar la lista de acceso sobre una interfaz determinada:

```
Router(config)#interface Tipo Módulo/Número
Router(config-if)#ip access-group Núm-lista-acceso {in|out}
```

- Si no se indica nada el sentido predeterminado es de salida.

Mostrar las listas de acceso definidas en el equipo:

```
Router#show ip access-lists
```

7.9. Configuración de listas de acceso basadas en nombre

Definición de una lista de acceso basada en nombre:

```
Router(config)#ip access-list {standard|extended} nombre
```

Definición de una regla de una lista de acceso basada en nombre:

```
Router(config-{std-|ext-}nacl)#{permit|deny} protocolo origen máscara_comodín
[operador puerto(s)] destino máscara_comodín [operador puerto(s)]
```

- `protocolo` puede ser `IP`, `TCP`, `UDP` o `ICMP`.
- `operador` puede ser `eq` (equal; igual), `lt` (less than; menor que), `gt` (greater than; mayor que), `neq` (not equal; distinto) o `range` (rango)
- La máscara predeterminada es `0.0.0.0`.
- Ahora las reglas se pueden dar de alta/baja de forma individual.
  - `no {permit|deny} {condiciones de la regla en la lista de acceso}`

Aplicar la lista de acceso sobre una interfaz determinada:

```
Router(config)#interface Tipo Módulo/Número
Router(config-if)#ip access-group Nombre-lista-acceso {in|out}
```

- Si no se indica nada el sentido predeterminado es de salida.

Mostrar las listas de acceso definidas en un equipo:

```
Router#show ip access-lists
```

7.9.1. Restriccciones:

- No están disponibles en versiones anteriores a la Cisco IOS 11.2.
- No se puede dar el mismo nombre a dos listas de acceso aunque sean de tipos distintos.

7.10. Control de acceso remoto a un equipo

Un *router*, típicamente, dispone de 5 líneas de terminal virtual (5 accesos *telnet* simultáneos) que permiten su configuración remota. Hemos de controlar los accesos a estas líneas, definiendo aquellos rangos de IP desde los que está permitido configurarlo. De otra forma, con las líneas de terminal virtual sin proteger, un atacante podría realizar los siguientes ataques:

- Ataque por fuerza bruta intentando averiguar usuarios y contraseñas.
- Ataque de denegación de servicio, ocupando las 5 líneas de terminal virtual, impidiendo que los administradores puedan acceder a ellas.

Configuración:

```
Router(config)#access-list Núm permit origen máscara_comodín
Router(config)#line vty {vty# | vty-range}
Router(config-line)#access-class Núm-lista-acceso {in|out}
```

7.11. Ejercicios propuestos

7.11.1. Listas de acceso estándar

A la vista del esquema planteado en la Figura 7.3, implementa las listas de acceso estándar que cumplan las siguientes restricciones:

- Los equipos de Red D sólo deben poder acceder a Red C.
- Los equipos de Red B no deben poder acceder a Red C.
- Los equipos de cualquier red, que no sean de la Red C, no deben llegar a Red D
- El equipo `192.168.100.17` es considerado "indeseable" en nuestra red y debemos garantizar que no pueda cursar tráfico fuera de su red local.
- Definir lista de acceso:
  - Mínimo número de reglas posible (eficiencia).
  - Lugar adecuado de la red (cerca/lejos de la fuente o cerca/lejos del destino).
  - Sentido correcto (entrada/salida).

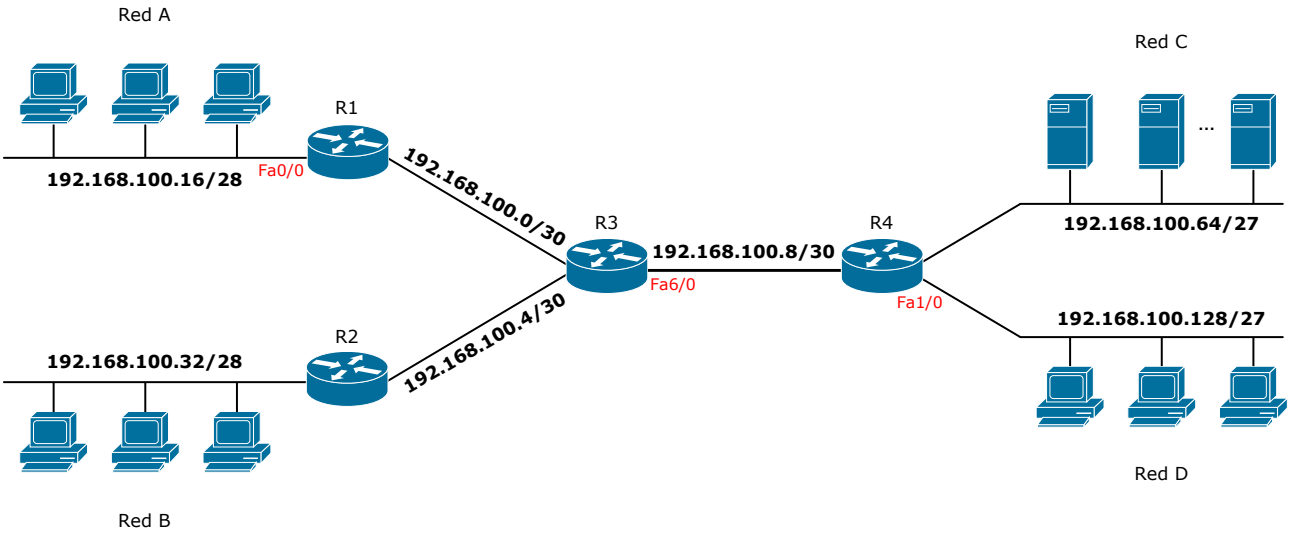


Figura 7.3: Esquema de red ejercicio listas de acceso estándar.

7.11.2. Listas de acceso extendidas

A la vista del esquema de red de la Figura 7.4, implementa las listas de acceso extendidas que cumplan las siguientes restricciones:

- *Dirección* y *Facturación* sólo deben poder comunicarse a través de los servidores corporativos, nunca directamente. Deben poder acceder a todos los servicios de cualquiera de los servidores.
- Desde redes situadas en Internet sólo se debe poder acceder a los puertos de servicio estándar de *HTTP* (`80`) y *SMTP* (`25`) de los servidores con dirección `.1` y `.2`. El resto de los puertos deben permanecer "cerrados".
- Los equipos de la oficina B deben poder acceder a cualquier puerto de los servidores corporativos.

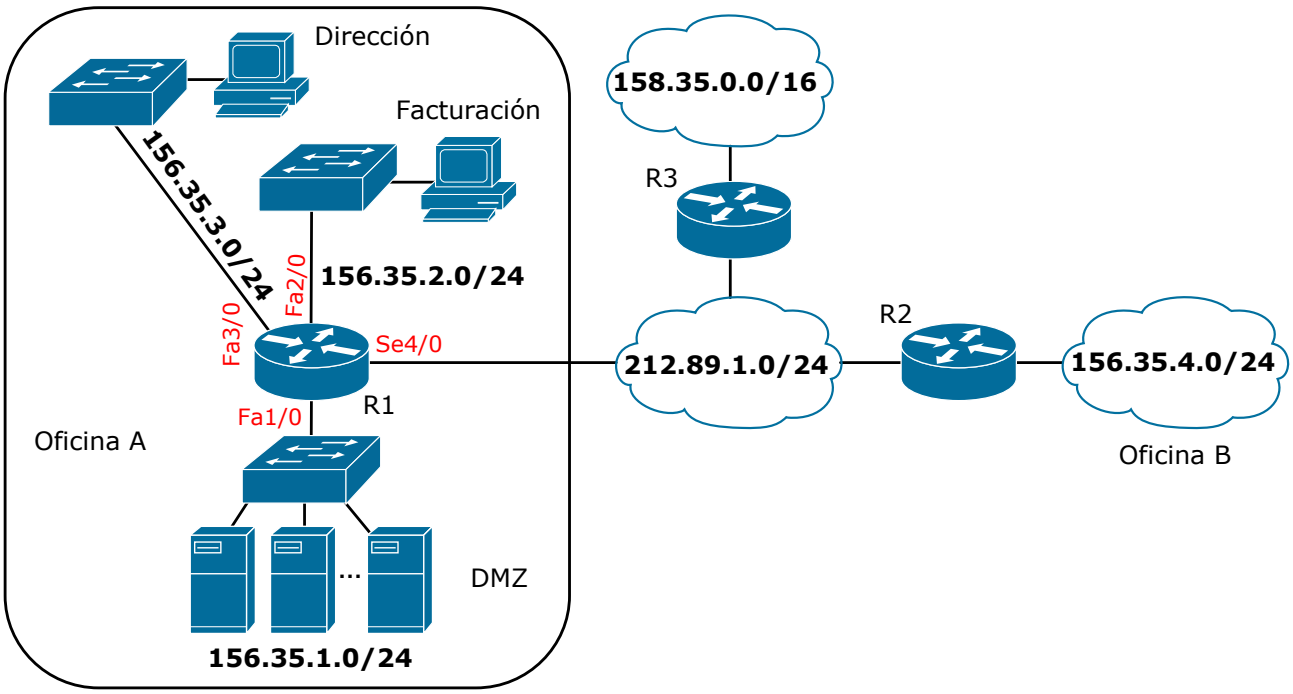


Figura 7.4: Esquema de red ejercicio listas de acceso extendidas.

8

Técnicas de traducción de direcciones y puertos. NAT/PAT

8.1. Objetivos

El objetivo de esta práctica es la toma de contacto con las técnicas de traducción de direcciones y de puertos (NAT/PAT) y su implementación en los equipos de interconexión. A través de la interfaz de comandos del sistema operativo IOS el alumno deberá aplicar las técnicas de traducción necesarias para que los equipos de una organización, utilizando direccionamiento privado, puedan acceder a Internet mediante traducciones sobre un rango limitado de direcciones públicas. Asimismo, se analizarán los problemas relacionados con estas técnicas de traducción y sus soluciones. Finalmente se plantearán soluciones en las que una organización desee situar en su red interna, utilizando direccionamiento privado, un equipo servidor accesible desde Internet.

8.2. Introducción

En los últimos años, las expectativas de crecimiento de Internet se han visto desbordadas. Acompañando a la enorme evolución tecnológica sufrida tanto en las redes de interconexión como en las de acceso, han ido apareciendo nuevos servicios que han convertido Internet en uno de los medios más poderosos y atractivos para el intercambio de información. La universalización en el acceso a Internet ha multiplicado el número de equipos conectados y por lo tanto las direcciones utilizadas; al ritmo de crecimiento de los últimos años, se esperaba que el direccionamiento IPv4 se agotase en un breve plazo.

Este problema de agotamiento del direccionamiento es una de las razones del desarrollo de IPv6, que entre otras ventajas dispone de una capacidad de direccionamiento mucho mayor: de 32 a 128 bits.

Pese a todo, y la vista de la lentitud en la adopción de IPv6 por parte de los operadores de comunicaciones y las empresas, es necesario desarrollar nuevas técnicas que permitan continuar el crecimiento de Internet basándose en el direccionamiento IPv4. Las técnicas diseñadas se pueden dividir esencialmente en dos bloques:

- Asignación óptima de direcciones: Mediante CIDR (Classless Inter-Domain Routing).
- Reutilización de direcciones: Mediante NAT/PAT (Network Address Translation/Port Address Translation).

8.3. Direccionamiento IP. Rangos de direcciones privadas.

Dado que el espacio de direccionamiento IPv4 es limitado, una de las posibilidades que surgen a la hora de solucionar los problemas de agotamiento del espacio de direcciones es la reutilización. Dicha reutilización debe realizarse de una forma ordenada. En Internet dos equipos diferentes no pueden tener la misma dirección, ya que la dirección de un equipo es el identificador que permite diferenciar los paquetes dirigidos a/procedentes del mismo.

Para ello, dentro del espacio de direccionamiento de IPv4 se han definido ciertos rangos como de direccionamiento privado. Dichos rangos de direcciones sólo se pueden emplear de forma interna a una organización y no pueden utilizarse para identificar de forma unívoca a un equipo en Internet, como en el caso del direccionamiento público. Además, el direccionamiento privado presenta otra diferencia fundamental respecto al direccionamiento público: no hay ningún organismo central que lo controle y lo administre. La IANA (Internet Assigned Numbers Authority), es la organización encargada de asignar el espacio de direccionamiento público, que al ser un recurso limitado tiene un precio elevado. Las direcciones privadas se tratan de forma diferente: cualquiera puede tomar un rango de los reservados y utilizarlo libremente de forma interna a su red. Un ejemplo gráfico de todo esto puede verse en la Figura 8.1.

Los rangos reservados para direccionamiento privado son los siguientes:

- `10.0.0.0/8`.
- `172.16.0.0/12`.
- `192.168.0.0/16`.

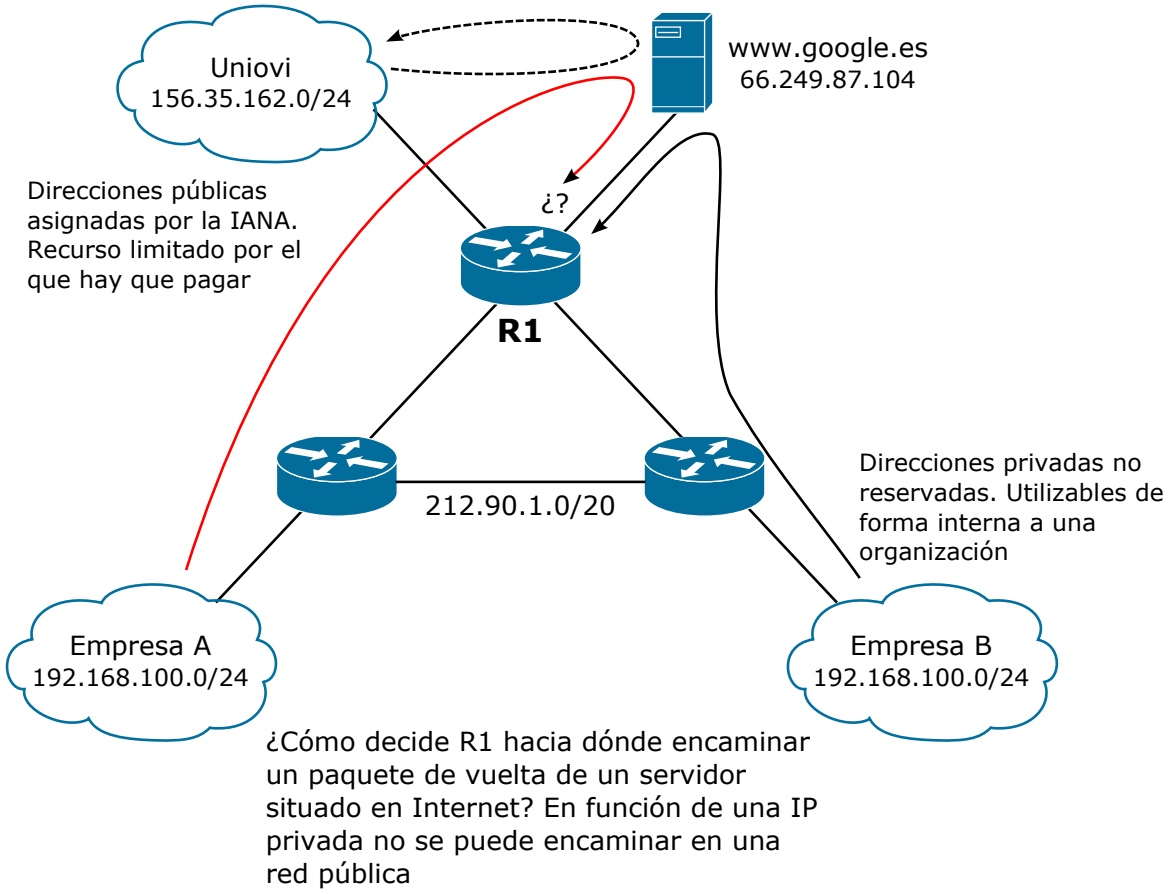


Figura 8.1: Diferencias entre direccionamiento público y privado.

8.4. Traducción de direcciones y puertos

Hasta ahora hemos visto cómo una empresa que necesite conectividad entre los equipos de una cierta zona puede utilizar un rango de direcciones privadas, distribuyendo el direccionamiento según sus necesidades dentro de esta red. La única restricción que se ha de cumplir es que estos equipos no pueden conectarse a Internet directamente. Quedándonos en este punto sólo hemos resuelto una pequeña parte de los problemas, ya que todas las empresas y particulares que precisen conectividad a Internet no podrían recurrir a esta solución.

Para solucionar este problema se diseñaron las técnicas de traducción de direcciones y de puertos (NAT/PAT), que combinan información proveniente de los niveles IP (direcciones de red) y TCP/UDP (puertos). Dichas técnicas se basan en traducir las direcciones privadas en una o varias direcciones públicas que sí puedan encaminarse a través de Internet. En la Figura 8.2 podemos ver un ejemplo de una traducción de direcciones y puertos.

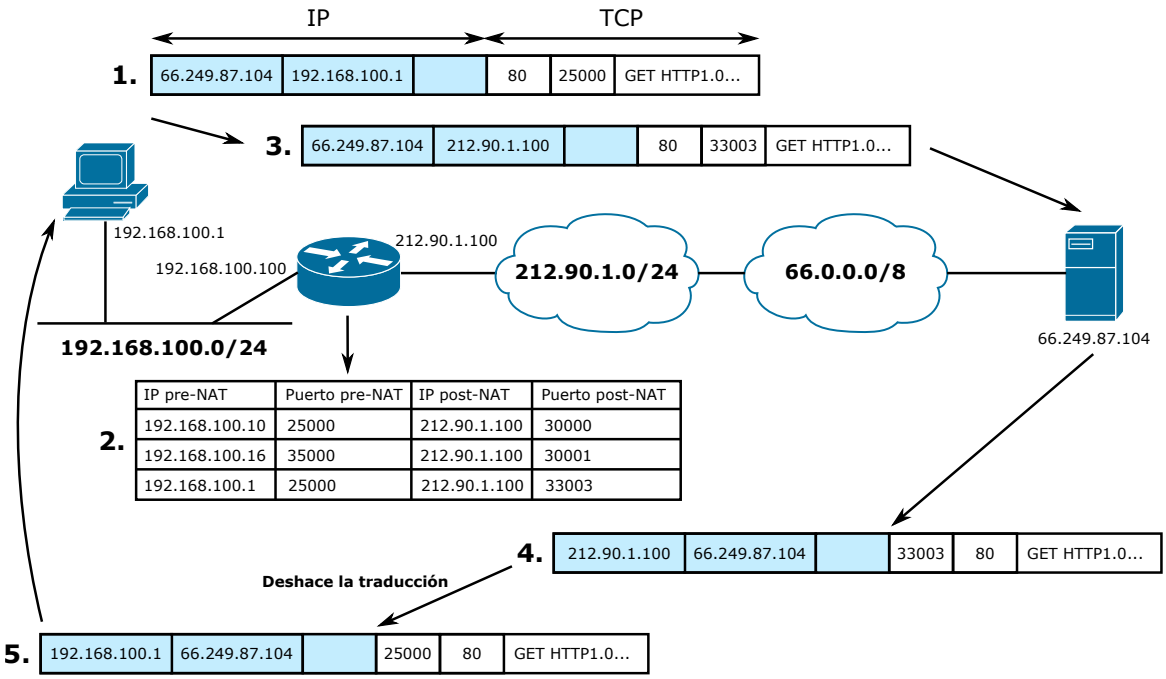


Figura 8.2: Ejemplo de traducción de direcciones y puertos.

Se puede apreciar que hay dos zonas claramente diferenciadas: una con direccionamiento privado, que pertenece a la red interna de una organización (rango de direcciones `192.168.100.0/24`) y otra con rango de direccionamiento público. Un equipo de la red que utiliza direccionamiento privado (`192.168.100.1`) desea obtener una página web del servidor con dirección `66.249.87.104`. Para ello envía un paquete dirigido a la dirección IP del servidor, al puerto de destino `80` y con dirección origen en su IP y el puerto en el que espera la respuesta. El equipo encargado de realizar las tareas de traducción hace lo siguiente:

- Introduce en la tabla de traducciones una nueva entrada con la dirección IP origen y el puerto de origen del paquete (IP y puerto pre-NAT)
- Sustituye en el paquete la dirección origen por la dirección IP pública de la interfaz sobre la que vaya a enviar el paquete.
- Sustituye el puerto origen por un puerto disponible propio del equipo que implemente las traducciones de direcciones y puertos.
- Introduce en la tabla de traducciones, relacionada con la entrada pre-NAT del paso 1, la dirección IP pública de la interfaz sobre la que vaya a enviar el paquete y el puerto localizado en el paso anterior (IP y puerto post-NAT).
- Envía el paquete sobre la interfaz adecuada, camino del destino.

El paquete llegará al servidor, que elaborará su respuesta. Dicha respuesta irá destinada a la IP y puerto post-NAT. El equipo encargado de las traducciones recibirá el paquete, consultará en la tabla de traducciones, y encontrará la entrada relacionada. Deshará la traducción con la información de la tabla, y enviará el paquete por la interfaz adecuada hacía el equipo que originó la petición.

Como puede comprobarse, basándose en una sola dirección IP pública y jugando con la información de los puertos, puede darse acceso a Internet a múltiples equipos simultáneamente (tantos como el número de puertos de los que disponga el equipo que realiza las traducciones). Es recomendable que la capacidad de procesamiento del equipo encargado de realizar las traducciones sea lo suficientemente elevada como para poder llevar a cabo todas estas tareas sin una influencia aparente en las acciones de los usuarios. Algunos equipos de gama baja (por ejemplo, *routers* inalámbricos) sufren caídas de prestaciones de hasta el 50 % cuando el equipo tiene que realizar traducciones respecto a casos en los que los dos equipos están en el interior de la red.

8.5. Problemas de las técnicas de traducción de direcciones y puertos.

Algunos protocolos (*RTSP*, *HTTP*, *FTP*, etc.) llevan como información del protocolo datos sobre direccionamiento IP y puertos. De esta forma, y teniendo en cuenta que no se ha mencionado nada sobre el tratamiento de la parte de datos del paquete traducido, cuando un paquete de estos protocolos atraviese un equipo NAT/PAT, la información que viaje en él dejará de ser válida. Un ejemplo de esto puede verse en la Figura 8.3, en la que un protocolo sin determinar lleva en la parte de datos el puerto en el que espera la respuesta. Cuando el destino reciba este paquete la respuesta viajará destinada al puerto indicado en el paquete, no al puerto en el que el equipo NAT/PAT esperaría la respuesta.

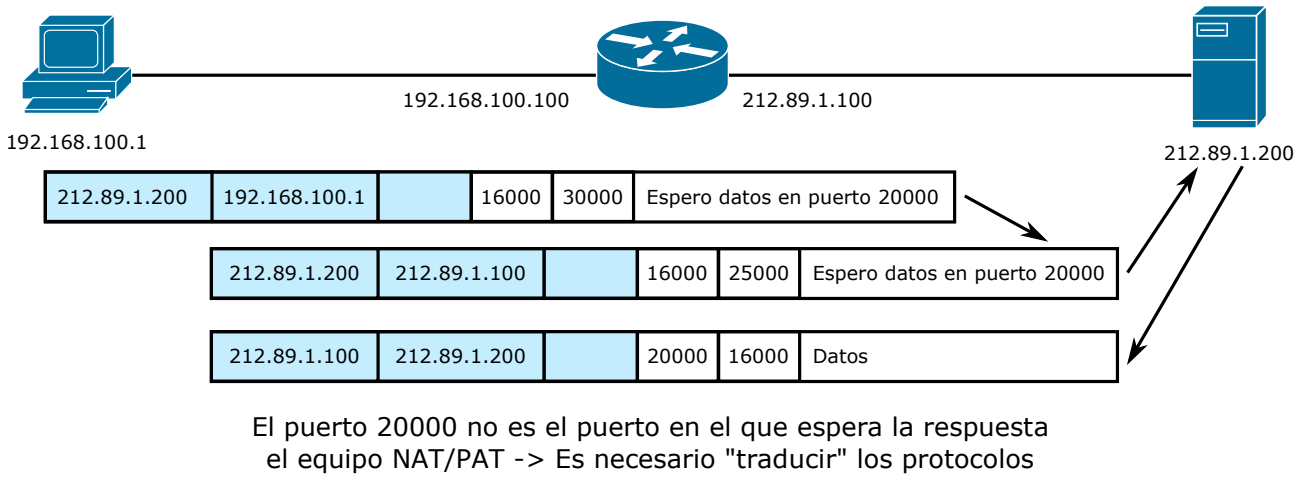


Figura 8.3: Problemas de NAT/PAT.

Una posible solución sería traducir el protocolo para que tenga en cuenta estas cuestiones. Sin embargo, dicha solución implica modificar la carga de datos de un paquete, obligando a recalcular por completo los campos de comprobación de los protocolos, con la posible influencia que esto pueda tener en las prestaciones alcanzadas en función de la capacidad del equipo que realiza las traducciones.

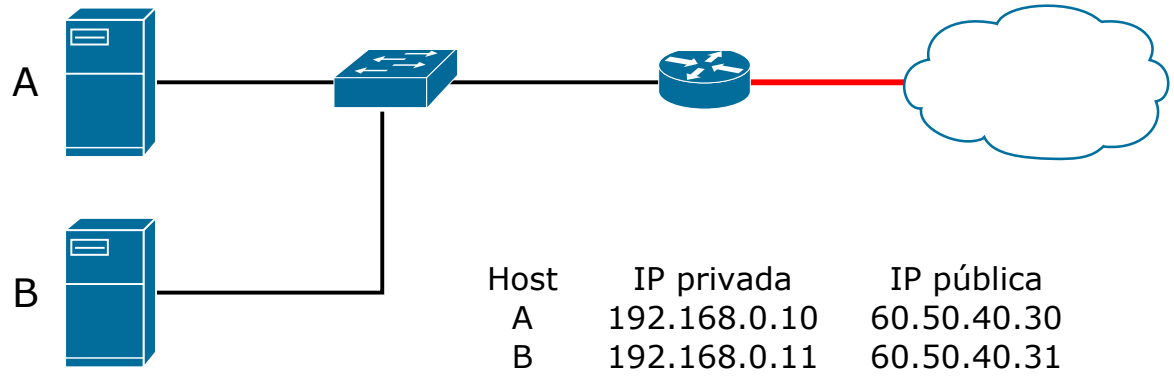
Desde el punto de vista de una red corporativa de un cierto tamaño, que utilice técnicas NAT/PAT para acceder a Internet, nos queda un problema por resolver: cómo disponer un servidor en la zona privada de nuestra red, utilizando direccionamiento privado, y que aún así sea accesible desde Internet. Esa es una de las finalidades de los distintos tipos de NAT que se presentan en la siguiente sección.

8.6. Tipos de NAT

8.6.1. NAT estático

Consiste simplemente en 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. La particularidad de tener que estar plenamente accesibles se debe a que, si sólo es necesario acceder a un puerto en cada uno, la alternativa de la traducción de puertos puede ser viable como se verá en la sección correspondiente.



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

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

Y la secuencia de comandos que habrá que ejecutar para realizar dicha configuración es:

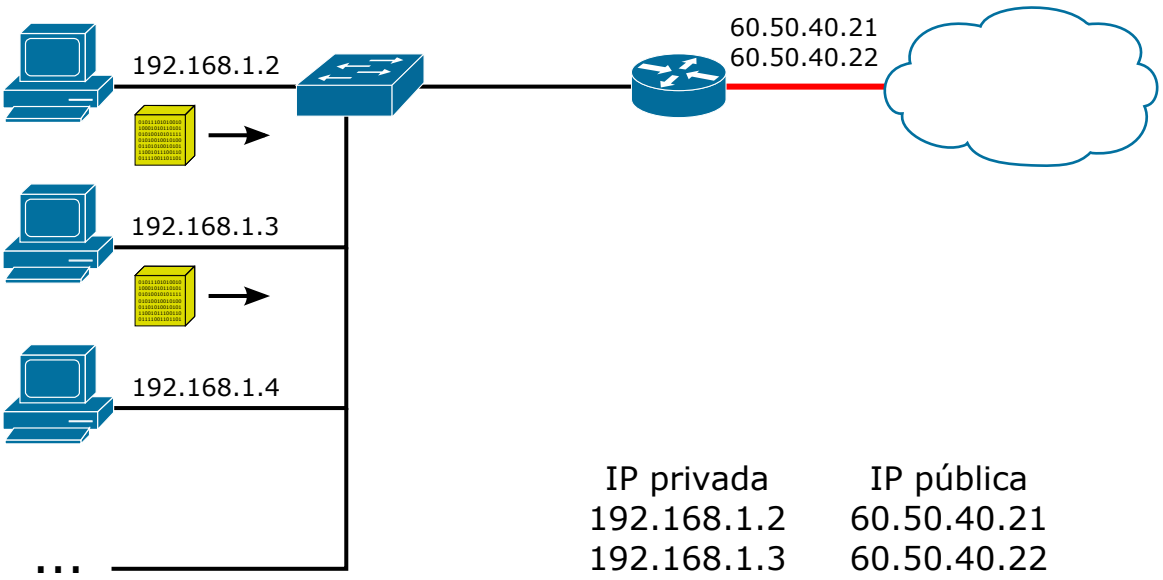
```
Router(config)#ip nat inside source static [protocolo] <IP int.> [puerto] <IP ext.> [puerto]
Router(config)#interface <interfaz interna>
Router(config-if)#ip nat inside
Router(config)#interface <interfaz externa>
Router(config-if)#ip nat outside
```

De manera opcional, permite filtro por protocolo, dirección origen y destino y puerto origen y destino.

Si se quiere bloquear el acceso a determinados puertos desde el exterior, todas estas reglas de traducción deben ir acompañadas por una ACL extendida y demás medidas de seguridad que se considere adecuado.

8.6.2. NAT dinámico

Consiste en que una dirección IP de la red privada (interna) tenga asignada una dirección IP de la red pública (externa), pero con la particularidad de que muchas direcciones IP privadas pueden tomar una dirección IP pública. Cuando un equipo con IP privada quiere salir a Internet, tomará una dirección IP pública si hay disponible. En caso de que no la haya deberá esperar hasta que caduque una asignación. 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 equipo privado que quiera usarla.



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.

Y la secuencia de comandos que habrá que ejecutar para realizar dicha configuración es:

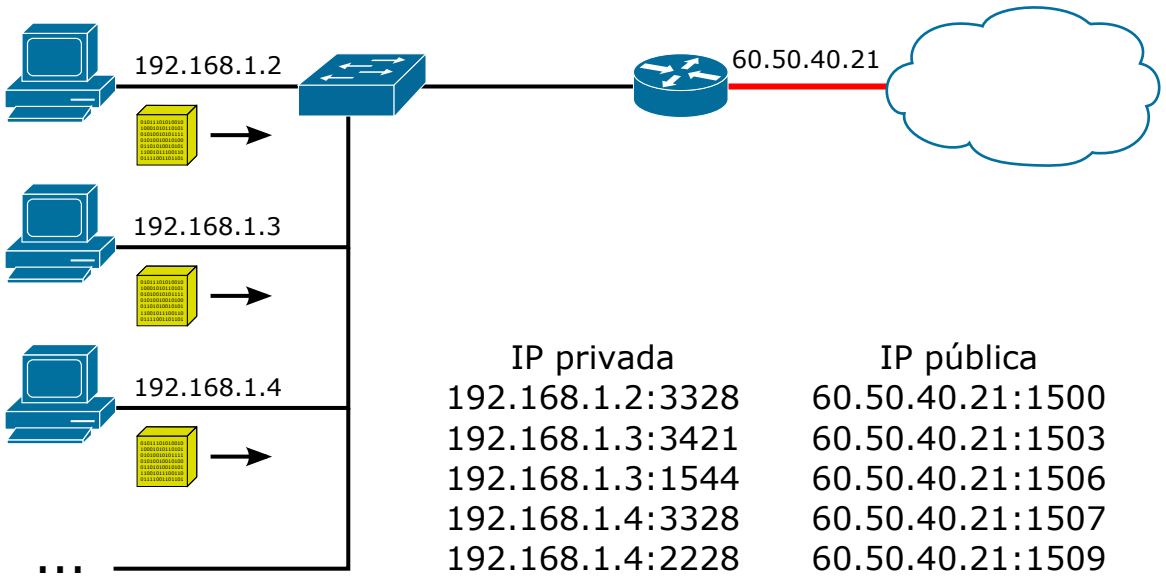
```
Router(config)#access-list <nº ACL> permit <red interna> <máscara_comodín>
Router(config)#ip nat pool <nombre> <1ª IP> <ult. IP>
              <netmask <máscara>|prefix-length <bits de máscara>>
Router(config)#ip nat inside source list <nº ACL> pool <nombre>
Router(config)#interface <interfaz interna>
Router(config-if)#ip nat inside
Router(config)#interface <interfaz externa>
Router(config-if)#ip nat outside
```

Descripción de los parámetros:

- nº ACL: Número de ACL, que puede ser cualquier número dentro de los rangos de ACL simples (1~99, 1300~1999).
- red interna: Dirección IP de la red interna.
- máscara_comodin: Inversa de la máscara de la red interna.
- nombre: Nombre identificativo del rango de direcciones externas.
- 1ª IP: Menor dirección IP del rango externo.
- ult. IP: Mayor dirección IP del rango externo.
- máscara: Máscara de las direcciones IP externas.
- bits de máscara: Longitud de la máscara de las direcciones IP externas.
- Interfaz conectada a la red interna.
- Interfaz conectada a la red externa.

8.6.3. NAT dinámico con sobrecarga

En este caso una o muchas direcciones IP privadas tienen asignada una única dirección IP pública. Cuando una dirección IP privada quiere salir a Internet, en lugar de tener una IP pública sólo para ese dispositivo, se le asigna un puerto público por cada petición de puerto privado que realice. Si un equipo 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 se debe configur explícitamente.



El NAT dinámico con sobrecarga es una extensión del NAT dinámico. Se parte de la base de que un *host* de una red privada suele usar un número limitado de puertos para los diferentes servicios que se están ejecutando en él. 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 sólo 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 *host* pueden funcionar perfectamente con los 65535 puertos que ofrece una única dirección IP. Además, en caso de que con una única 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.

```
Router(config)#access-list <nº> permit <red interna> <máscara_comodín>
Router(config)#ip nat pool <nombre> <1ª IP> <últ. IP> <máscara|longitud>
Router(config)#ip nat inside source list <nº> pool <nombre> overload
```

Si se quiere que todos los *host* salgan por una interfaz:

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

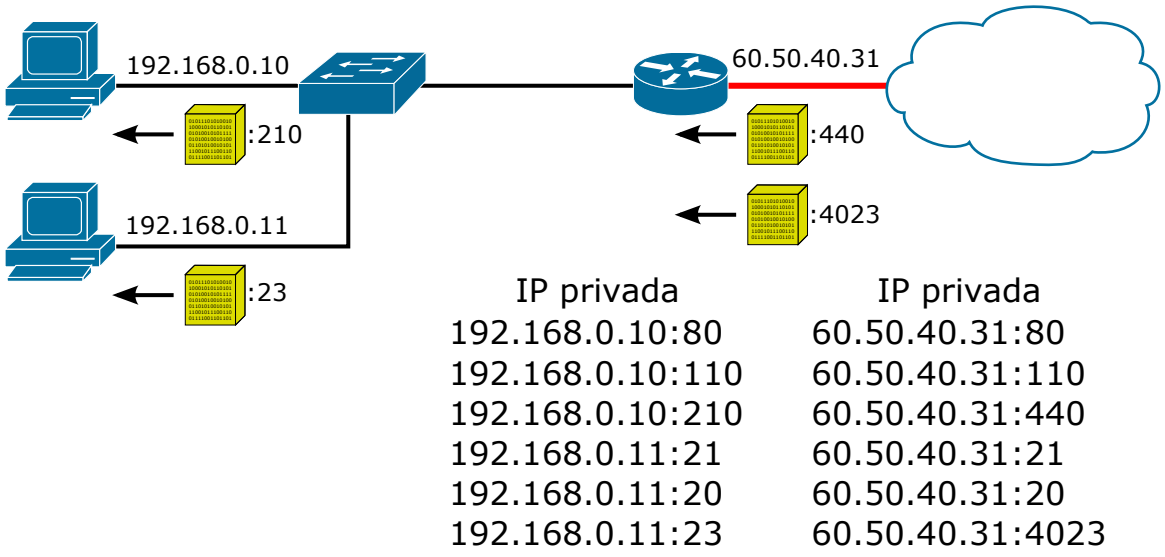
Parámetros:

- nº: Número identificativo de la ACL.
- red interna: Red interna para hacer NAT.
- máscara_comodín: Inverso de la máscara de la red interna.

- nombre: Nombre identificativo del rango de IP externo.
- 1ª IP: Dirección IP inferior del rango externo.
- últ. IP: Dirección IP superior del rango externo.
- máscara: Máscara de red del rango externo de direcciones IP.
- longitud: Longitud en bits de la máscara.
- Interfaz: Interfaz que se va a asignar como interfaz conectada a la red externa.

8.6.4. PAT

Es una asignación estática de puertos públicos a puertos y direcciones IP privadas. Se diferencia entre los protocolos TCP y UDP.



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.

```
Router(config)#ip nat inside source static <tcp|udp>
                <IP interna> <puerto interno>
                <IP externa|interfaz> <puerto externo>
```

Parámetros:

- tcp: Si el puerto al que se va a aplicar NAT va a utilizar el protocolo TCP.
- udp: Si el puerto al que se va a aplicar NAT va a utilizar el protocolo UDP.
- IP interna: Dirección IP del *host* en la parte interna.
- puerto interno: nº. de puerto interno del *host*.
- IP externa: 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.
- puerto externo: nº. de puerto en la interfaz o dirección IP externa.

8.6.5. Resumen de los tipos de NAT y PAT

Como se ha explicado en las secciones correspondientes, el NAT estático consiste en asignar a cada *host* de la red interna 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 *host* 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 de manera permanente.

8.7. Ejercicio propuesto

En el esquema de red de la Figura 8.4, y teniendo en cuenta los requisitos expuestos a continuación, implemente las técnicas de traducción y filtrado de tráfico necesarias:

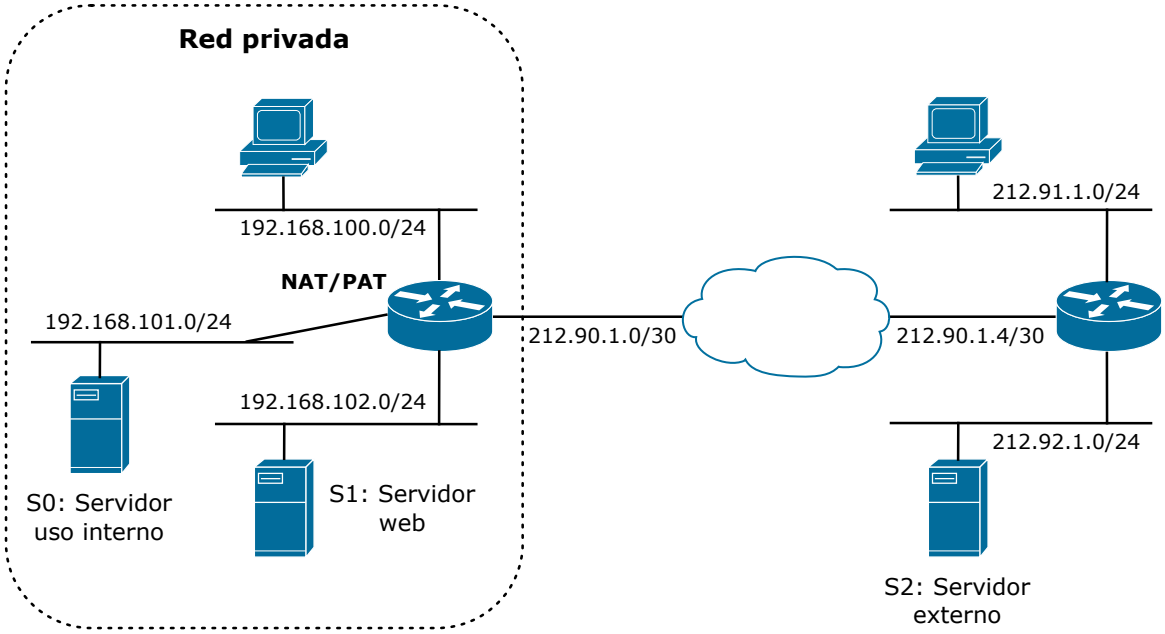


Figura 8.4: Esquema de red correspondiente al ejercicio propuesto.

Requisitos:

- Los PC situados en la red `192.168.100.0/24` han de poder enviar tráfico hacia cualquier punto de Internet y recibir las posibles contestaciones.
- El servidor S0 es sólo de uso interno; ni usuarios que no sean de la red privada han de poder acceder a él ni debemos permitir que tráfico originado en dicho servidor salga al exterior de la red.
- El servidor S1 es el servidor web corporativo, en el que se encuentra el catalogo de productos, al que debe acceder todo aquel que lo desee. Dicho servidor atiende peticiones en el puerto 80.