Manual de referencia de Cisco IOS

Conceptos y comandos principales para realizar las PEC y práctica

PID_00262485

Àlex Macia Pérez
Eduard Lara Ochoa

Tiempo mínimo de dedicación recomendado: 7 horas

Universitat Oberta de Catalunya

Àlex Macia Pérez
Profesor colaborador de la UOC.

Eduard Lara Ochoa
Profesor colaborador de la UOC.

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Joan Melià Seguí (2019)

Primera edición: febrero 2019

Av. Tibidabo, 39-43, 08035 Barcelona
Diseño: Manel Andreu
Realización editorial: Oberta UOC Publishing, SL

Índice

Introducción..... 5

1. Introducción a Cisco IOS..... 7
1.1. Acceso al IOS 7
1.2. Modos de operación 10
1.3. Inspección del estado 11
1.4. Ayuda en la introducción de comandos 12
1.5. Arquitectura del sistema 16
1.6. Sistema de archivos 19

2. Configuración de conmutadores en Cisco IOS..... 23
2.1. Seguridad y acceso remoto 28
2.2. Caso práctico 32
2.3. Configuración de conmutadores: VLAN 34
2.4. Configuración de conmutadores: tabla MAC 40

3. Configuración de enrutadores Cisco IOS..... 42
3.1. Estructura de un enrutador (*router*) 42
3.1.1. Consulta del estado del enrutador (comando `show`) 42
3.2. Configuración de enrutadores con Cisco IOS 43
3.2.1. Configuración de interfaces 43
3.2.2. Configuración de interfaces Ethernet 44
3.2.3. Configuración de interfaces serie 45
3.2.4. Tabla ARP 46
3.3. Encaminamiento estático 48
3.3.1. Ruta por defecto (Default Gateway) 52
3.4. Encaminamiento entre VLAN 53
3.4.1. Características del VLAN Trunking Protocol 55
3.4.2. Estándar IEEE-802.1Q 56
3.4.3. Ejemplo de encaminamiento entre diversas VLAN 56
3.5. Configuración de los protocolos de encaminamiento 59
3.5.1. Distancia administrativa de un protocolo de encaminamiento 60
3.5.2. Encaminamiento con OSPF 61
3.5.3. Encaminamiento con EIGRP 69
3.6. Redes IPv6 75
3.6.1. Tipo de direcciones IPv6 76
3.6.2. Ejemplo: generación de direcciones *Link Local*..... 77
3.6.3. Ejemplo de configuración de interfaces Ethernet con direcciones *Global Unicast*..... 79
3.6.4. Ejemplo de direccionamiento estático a IPv6 81

3.6.5. Ejemplo de encaminamiento entre diversas VLAN con IPv6 84
3.7. Encaminamiento con OSPFv3 88
3.7.1. Ejemplo: encaminamiento con OSPFv3 88

Bibliografía..... 93

Introducción

Este módulo sirve de introducción a la operación del sistema Cisco IOS, utilizado por los dispositivos Cisco (enrutadores, conmutadores). Si bien este sistema operativo es complejo y tiene muchas opciones, los conceptos que se presentan aquí son los más básicos para poder seguir la asignatura si partimos desde cero.

Dispositivos que se consideran en este documento

Los ejemplos de comandos y configuraciones que se presentan en este módulo se han realizado principalmente con el programa de simulación de redes Packet Tracer. Se trata de una herramienta desarrollada por Cisco y es parte de los recursos de su programa de formación Network Academy.

En el apartado de ayuda del programa podéis encontrar, entre otras informaciones, la referencia completa de los comandos del IOS disponibles. Basta con acceder a los contenidos de la ayuda, a Configuring Devices, apartados Router IOS, Router IOS 15 y Switch IOS.

En este módulo nos centraremos principalmente en el funcionamiento del IOS que implementan conmutadores y enrutadores. Pero más allá del hardware específico, todas las funcionalidades que se describen en el presente módulo aplican al conjunto de dispositivos que trabajan sobre un sistema IOS.

Finalmente, también es importante comentar que el IOS se ha convertido en un estándar *de facto* para el software de los dispositivos de red y la mayoría de las empresas del sector implementan sistemas muy similares; por lo tanto, los contenidos siguientes tienen un alcance bastante amplio en relación con la extensa variedad de hardware disponible en el mercado.

1. Introducción a Cisco IOS

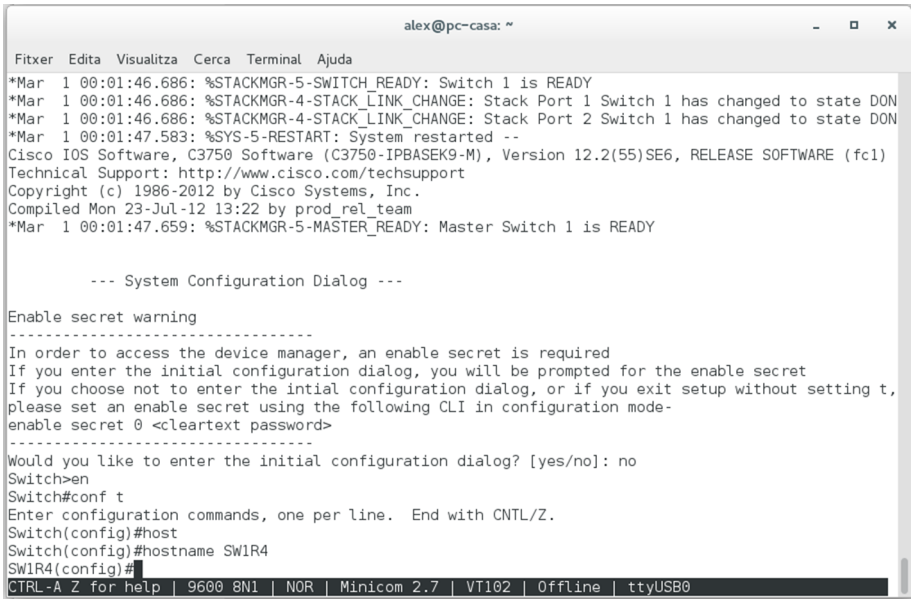
Este apartado resume el acceso y los modos de operación de Cisco IOS. Además, se hace una breve introducción a la arquitectura del sistema y el sistema de archivos de los dispositivos Cisco.

1.1. Acceso al IOS

Cisco IOS (IOS, siglas que hacen referencia a Internetwork Operating System) es el nombre comercial del sistema que integran los dispositivos de red de Cisco. Se configura por medio de una interfaz de línea de comandos (CLI - Command Line Interface), que diferencia entre varios niveles o modos de privilegio y funcionalidad.

El conjunto de comandos disponibles en cada modo es diferente y el usuario debe cambiar de modo en función de lo que quiera hacer. En la figura 1 se muestra un ejemplo de la interfaz de línea de comandos del IOS.

Figura 1. Configuración desde el CLI



Hay varias maneras de acceder al IOS para configurar los diferentes dispositivos. Las más habituales son:

- conexión directa por **consola**
- acceso **remoto** por red

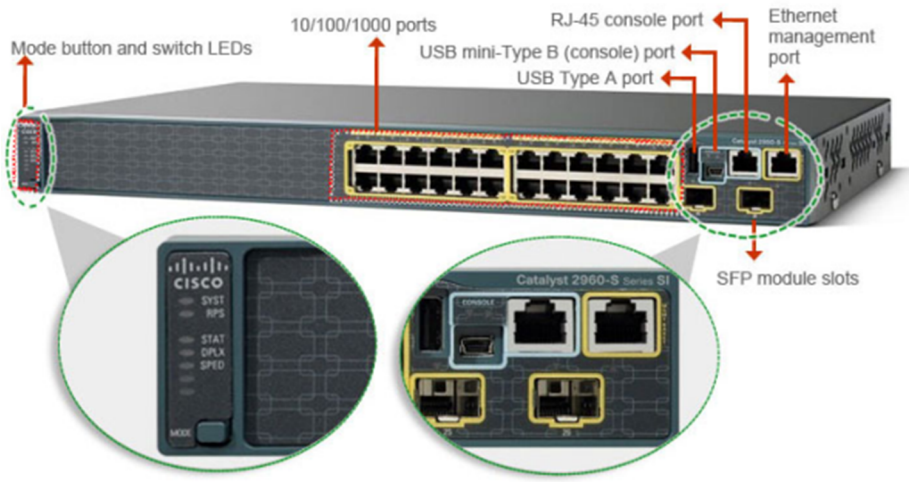
La primera vez que se configura un conmutador o un enrutador, hay que conectarse directamente mediante un **cable de consola**. Todos los dispositivos gestionados incluyen uno o más puertos de gestión, también llamados puertos de consola o *out-of-band*.

Out-of-band

Out-of-band, o fuera de banda, es un término heredado de la nomenclatura de las telecomunicaciones y se refiere a puertos que trabajan aparte de la banda de frecuencias por donde se transmiten los datos y la voz.

Los puertos de consola siempre están separados de los puertos de red e indicados claramente. Los podemos encontrar en el panel frontal, posterior o incluso excepcionalmente en el lateral, y pueden tener diferentes formatos: **RJ45** (consola o Ethernet de gestión), **RS-232 o DB9, USB mini (tipo B)**. En la figura 2 se muestran algunos ejemplos de puertos de gestión de un conmutador.

Figura 2. Panel frontal de un conmutador serie CISCO Catalyst 2960



Fuente: Cisco

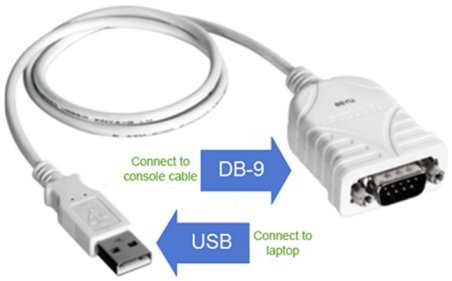
La conexión de consola siempre está disponible, aunque la configuración no sea correcta, esté incompleta o incluso si el IOS no se puede cargar. Es por eso que se utiliza también durante los procedimientos de recuperación de las diferentes situaciones de desastre, si el archivo de sistema está dañado o no se encuentra durante el proceso de arranque, o si hemos perdido u olvidado las contraseñas y hay que recuperar el acceso o restablecer la configuración de fábrica.

Para completar la conexión física se conecta un cable de consola desde el puerto de gestión del dispositivo que se quiere configurar hasta el puerto serie de un ordenador (COM1, COM2...), o alternativamente a un puerto USB utilizando algún adaptador de puerto serie a USB. En las figuras 3 y 4 se muestran ejemplos de estos elementos.

Figura 3. Cable de consola, serie DB-9 a consola RJ-45

Fuente: https://dcloud-cms.cisco.com/help/connect_console

Figura 4. Adaptador puerto serie (DB-9) a USB



Fuente: https://dcloud-cms.cisco.com/help/connect_console

Para gestionar las comunicaciones por el cable de consola hace falta disponer de algún programa de emulación de terminal serie, por ejemplo: PuTTY (Windows), minicom (Linux), Kermit (Unix), Z-Term (MacOS). A continuación se muestra la configuración del puerto serie «/dev/ttyS0» desde el programa minicom. Estos son los parámetros habituales, aunque la velocidad puede variar entre diferentes fabricantes:

```
Bits por segundo (Baud)     :  9600
Bits de datos               :     8
Paridad                     :    no
Bits de parada              :     1
Control de flujo            :    no
```

Por otra parte, también se puede acceder al IOS **por red** utilizando algún protocolo de conexión remota, normalmente telnet o ssh. Este último funciona sobre comunicaciones cifradas y es el método recomendado.

A diferencia del acceso por consola, que requiere estar físicamente junto al dispositivo, este procedimiento tiene la ventaja de que se puede usar desde cualquier terminal con acceso a la misma red que el dispositivo, siempre y cuando la configuración de seguridad de la red lo permita, que el dispositivo tenga establecida la configuración de red y que los servicios de acceso remoto estén activos y correctamente configurados.

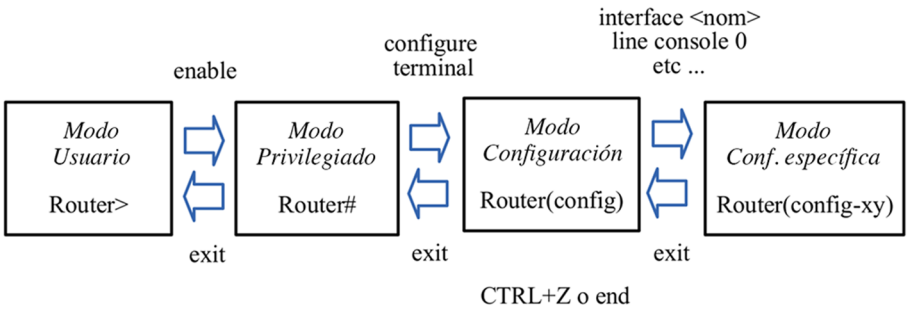
Así pues, el acceso remoto es opcional y no siempre estará disponible o será necesario configurarlo. No hay que olvidar que los conmutadores no requieren direccionamiento de red (dirección IP y puerta de enlace) para realizar su

función, o que el hecho de estar disponible remotamente supone un mayor riesgo de seguridad. Más adelante se detalla la configuración del IOS para permitir el acceso remoto tanto por telnet como cifrado por ssh.

1.2. Modos de operación

Uno de los aspectos más importantes del IOS es que se estructura en una serie de modos de operación, a los que se accede de manera secuencial, por niveles. Dependiendo del modo de operación actual, los comandos que se pueden ejecutar cambian totalmente. En todo momento se puede saber en qué modo estamos por la forma del símbolo del sistema (*prompt*). El usuario puede acceder a cada uno de los modos o volver a un modo anterior, como se muestra en la figura 5.

Figura 5. Modos de operación y comandos de navegación entre modos



El modo usuario es el inicial al entrar en el dispositivo. Solo permite ejecutar algunos comandos de consulta para visualizar su estado. No lo utilizaréis casi nunca. Se puede pasar a modo privilegiado con el comando `enable`.

El modo privilegiado permite visualizar el estado del dispositivo con más detalle y gestionar los sistemas de archivos. Como es el punto de entrada al control del dispositivo, normalmente se protege con una contraseña. También permite ejecutar los comandos `ping`, `telnet` y `ssh`. Se puede avanzar en el modo de configuración con el comando `configure terminal`.

El modo de configuración solo permite ejecutar comandos que cambian la configuración general del dispositivo. Por ejemplo, el nombre del *host* o su servidor DNS.

Se puede entrar en modo interfaz utilizando la instrucción `interface <nombre interfaz>`. El nombre debe ser completo y acepta espacios (por ejemplo: `interface FastEthernet 0/0`). El modo interfaz permite configurar una interfaz de red concreta. Por ejemplo, la dirección IP. Hay un modo asociado a cada interfaz concreta del dispositivo.

Para retroceder al modo anterior, se usa el comando `exit`. Es posible retroceder directamente a modo privilegiado con CTRL + Z o con el comando `end`.

Ejemplo

```
Router>enable
Router#configure terminal
Router(config)#interface FastEthernet 0/0
Router(config-if)#exit
Router(config)#exit
Router#exit
```

1.3. Inspección del estado

La utilidad principal del modo privilegiado es poder inspeccionar el estado del dispositivo mediante variaciones del comando `show`. Este comando tiene muchas opciones y permite visualizar datos de todo tipo.

Algunas opciones de utilidad o que se usan a menudo son:

- `show running-config`: muestra la configuración en ejecución.
- `show ip interface brief`: muestra un resumen del estado de las interfaces.
- `show interfaces`: muestra información detallada de las interfaces de red.
- `show interfaces <nombre interfaz>`: muestra información detallada de una interfaz concreta.
- `show cdp neighbors`: permite listar dispositivos Cisco vecinos conectados.

Ejemplo

```
SWP1>
SWP1>
SWP1>enable
SWP1#show running-config
Building configuration...

Current configuration : 2567 bytes
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname SWP1
!
```

```
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!




RINT>enable
RINT#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES unset  up                    up
GigabitEthernet0/0.4   192.168.4.1     YES manual up                    up
GigabitEthernet0/0.10  172.10.0.1      YES manual up                    up
GigabitEthernet0/0.20  172.20.0.1      YES manual up                    up
GigabitEthernet0/0.99  unassigned      YES unset  up                    up
GigabitEthernet0/1     unassigned      YES unset  administratively down down
Serial0/0/0            unassigned      YES unset  up                    down
Serial0/0/1            unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
```

1.4. Ayuda en la introducción de comandos

Los comandos pueden tener muchas opciones y se construyen ordenadamente. Las opciones disponibles en cada momento dependen de la opción inmediatamente anterior y las posibilidades se despliegan, por tanto, en forma de árbol.

Así, los comandos resultantes se componen de una serie de opciones interdependientes y que no pueden intercambiar su posición. Por suerte, el IOS incorpora una serie de utilidades para facilitar su uso. Por ejemplo, el IOS dispone de los mecanismos siguientes para facilitar la introducción de comandos:

- Introduciendo `?` se puede acceder a la ayuda. Si se hace a medio comando, muestra la ayuda vinculada a las opciones introducidas hasta el momento (por ejemplo: `show ip?`).

- La tecla de tabulador permite autocompletar la introducción de un comando.

- No es necesario introducir el texto completo del nombre de un comando o una opción. Si no hay ambigüedad en el comando introducido, el sistema lo resuelve automáticamente (por ejemplo: en lugar de `configure`

`terminal` se puede escribir `conf t`, o en vez de `show running-config` se puede escribir `sh run`).

- El comando `do` permite ejecutar comandos del modo privilegiado desde cualquier modo de configuración, general o específico. Esto nos ahorra tener que cambiar de modo.

La ayuda que se muestra usando `?` después de cada opción despliega una lista *indentada* de dos columnas con el conjunto de nuevas opciones disponibles. En la primera columna indica el valor concreto de la opción o de los valores posibles, y en la segunda columna, una descripción breve de la misma.

Además, sigue una nomenclatura que también facilita identificar los valores de las opciones disponibles a medida que se construye cada comando.

Si solo indica un texto o carácter sin *indentación* y sin añadir ninguna descripción, significa que es la única opción posible:

1) `<0-9>` representa un rango de valores posibles, en este caso un entero entre 0 y 9.

2) `WORD` hace referencia a una palabra cualquiera, sin espacios.

3) `LINE` quiere decir que se puede escribir una frase con espacios, si fuera necesario.

4) Los diferentes formatos de direcciones: MAC, IPv4 e IPv6.

- **dirección IPv4:** A.B.C.D
- **dirección MAC:** H.H.H
- **link-local:** X: X: X: X :: X dirección IPv6
- **prefijo IPv6:** X: X: X: X :: X / <0-128>

5) `<Cr>` hace referencia a *carriage return* e indica que **el comando está completo** y se puede ejecutar. Mientras no aparezca esta opción significa que el comando necesita más opciones.

El uso de mayúsculas también permite diferenciar de un vistazo aquellas opciones que son variables.

Ejemplo

```
Router(config)#enable password ?
   7      Specifies a HIDDEN password will follow
   LINE   The UNENCRYPTED (cleartext) 'enable' password
   level  Set exec level password
```

```
Router(config)#interface GigabitEthernet ?
   <0-9>  GigabitEthernet interface number
Router(config)#interface Giga?
GigabitEthernet
Router(config)#interface Giga ?
   <0-9>  GigabitEthernet interface number
Router(config)#interface Giga 0?
/
Router(config)#interface Giga 0/?
   <0-24>  GigabitEthernet interface number
Router(config)#interface Giga 0/1

Router(config-if)#
Router(config-if)#mac-address ?
   H.H.H  MAC address
Router(config-if)#mac-address 0001.0001.0001 ?
   <cr>
Router(config-if)#mac-address 0001.0001.0001
Router(config-if)#ipv6 address ?
   WORD                General prefix name
   X:X:X:X::X          IPv6 link-local address
   X:X:X:X::X/<0-128>  IPv6 prefix
   autoconfig          Obtain address using autoconfiguration

Router(config)#ip route ?
   A.B.C.D  Destination prefix
Router(config)#ip route 192.168.0.0 ?
   A.B.C.D  Destination prefix mask
Router(config)#ip route 192.168.0.0 255.255.255.0 FastEthernet 0/0 ?
   <1-255>  Distance metric for this route
   <cr>
Router(config)#ip route 192.168.0.0 255.255.255.0 FastEthernet 0/0 20 ?
   <cr>
Router(config)#ip route 192.168.0.0 255.255.255.0 FastEthernet 0/0 20
```

En caso de error, el IOS muestra información que sirve para poder identificar la causa del mismo y hacer las correcciones necesarias:

- El comando está incompleto, falta añadir alguna opción más.

- La sintaxis no es correcta. Es importante fijarse en el marcador «^» que aparece debajo del comando y que indica exactamente el carácter donde empieza a no ser válido.

- Si no es ninguno de los errores anteriores, muestra una breve descripción del problema, por ejemplo hace referencia a un recurso inexistente.

Ejemplo

```
Router(config)#interface gigabitEthernet
% Incomplete command.
Router(config)#interface gigabitLAN 0/1
               ^
% Invalid input detected at '^' marker.
Router(config)#interface gigabitEthernet 0
                    ^
% Invalid input detected at '^' marker.
Router(config)#interface gigabitEthernet 0/10
%Invalid interface type and number
Router(config)#
```

Para completar el análisis de los diferentes mecanismos de ayuda integrados en el IOS, hay que comentar también las **opciones para buscar y filtrar** la información de salida de los comandos `show` y `more`. Las operaciones disponibles son:

- Buscar: «`begin expression`» comienza en la primera línea en la que se encuentra la expresión.

- Filtro: «`include expression`» solo muestra las líneas que contienen la expresión.

- Filtro: «`exclude expression`» solo muestra las líneas que no contienen la expresión.

La opción correspondiente se indica a continuación del comando precedida por la barra vertical «`|`», como en una redirección o *pipe* de Linux. Para la expresión que se quiere buscar o filtrar se usan expresiones regulares. Por ejemplo, podéis consultar solo las líneas que contienen cierta palabra en el archivo de configuración `show running-config | include secret`, o consultar la información del dispositivo a partir de la línea que contiene un texto concreto `show version | begin FastEthernet`.

Ejemplo

```
SW1#show running-config | include secret
enable secret 5 $1$mERr$ILwq/b7kc.7X/ejA4Aosn0

SW1#show running-config | include secret|hostname
hostname SW1
enable secret 5 $1$mERr$ILwq/b7kc.7X/ejA4Aosn0

SW1#show version | begin FastEthernet
24 FastEthernet/IEEE 802.3 interface(s)
```

```
2 Gigabit Ethernet/IEEE 802.3 interface(s)

63488K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address       : 0002.4A86.2754
Motherboard assembly number     : 73-9832-06
Power supply part number        : 341-0097-02
...

SW1#show interfaces | include FastEthernet0/([3-5])
FastEthernet0/3 is down, line protocol is down (disabled)
FastEthernet0/4 is down, line protocol is down (disabled)
FastEthernet0/5 is down, line protocol is down (disabled)

SW1#show interfaces | include FastEthernet|duplex
FastEthernet0/1 is down, line protocol is down (disabled)
   Half-duplex, 100Mb/s
FastEthernet0/2 is down, line protocol is down (disabled)
   Half-duplex, 100Mb/s
...
```

1.5. Arquitectura del sistema

Los dispositivos de red no dejan de ser ordenadores, pero extremadamente especializados para realizar su tarea. Se pueden encontrar los mismos componentes de hardware a excepción de los periféricos: procesadores, memorias de sistema, unidades de almacenamiento, placas base y en general más puertos de los que se encontrarían en un dispositivo de usuario final. En cuanto al software, también disponen de un sistema operativo (por ejemplo, Cisco IOS) con los diferentes componentes de gestión.

El sistema de archivos se organiza en diferentes memorias de capacidades y prestaciones diversas según el uso que le dará el IOS. La memoria de mayor capacidad y donde se guarda, por ejemplo, la imagen del IOS (archivo comprimido del sistema) se denomina **Flash**. Es una memoria de lectura y escritura y en ella se pueden guardar imágenes de versiones actualizadas del IOS o copias de los archivos de configuración. Normalmente está integrada en la placa base del dispositivo, pero algunos modelos disponen de ranuras o puertos USB que permiten ampliarla.

En la memoria principal **RAM** se cargan el sistema en ejecución, la configuración en funcionamiento *running-config*, las estructuras de datos y en general todo lo necesario para que el IOS funcione, pero es volátil y por lo tanto la información no se mantiene cuando el dispositivo se detiene.

Por esta razón, los dispositivos también incorporan una memoria de acceso rápido, de poca capacidad, que se puede escribir y que no es volátil, llamada **NVRAM** (RAM no volátil), donde solo se guarda la configuración que se carga en el arranque *startup-config*.

También hay una memoria **ROM** (solo lectura) que contiene el sistema de arranque llamado System Bootstrap o monitor de la ROM, que se encarga de inicializar el hardware y localizar el archivo con la imagen del sistema para cargarlo y arrancar. Este incluye una interfaz CLI que funciona como sistema de recuperación *rommon*.

En la tabla 1 se indican las diferentes memorias que se pueden encontrar en los dispositivos y sus características más importantes.

Tabla 1. Tipo de memorias existentes en los dispositivos

Memoria	R/W	Volátil	Contenido principal
ROM	R	No	Sistema de arranque System Bootstrap o monitor de la ROM. Incluye el sistema de recuperación *rommon*.
FLASH	R/W	No	Imagen comprimida del sistema IOS, aunque se le pueden añadir nuevas imágenes o guardar configuraciones.
NVRAM	R/W	Sí	La configuración de inicio *startup-config*.
RAM	R/W	No	El sistema en ejecución y las estructuras de datos necesarios para el funcionamiento, por ejemplo: • la configuración en ejecución *running-config* • los búferes • tablas de sistema

Por lo tanto, el proceso de arranque de los dispositivos consiste en los pasos siguientes:

1) POST (Power-On Self-Test), inicialización y comprobación del hardware.

2) Carga del sistema de arranque System Bootstrap desde la memoria ROM a la memoria principal (RAM) para ejecutarlo.

3) El sistema de arranque localiza la imagen del IOS, la descomprime y la carga en la memoria para ejecutarla.

4) El IOS se ejecuta y copia la configuración de inicio *startup-config* en la memoria *running-config*; los cambios en la configuración siempre se llevan a cabo sobre esta última.

Durante el proceso de arranque se muestra un resumen de la información del sistema, tanto del hardware como del software. Esta información se puede recuperar en cualquier momento con el comando `show version`. En la tabla 2 se recogen a modo de ejemplo algunos de los parámetros de un enrutador Cisco 2901.

Tabla 2. Parámetros de sistema del enrutador Cisco 2901

Parámetro de sistema	Valor
Versión del IOS	15.1(4)M4
Archivo de imagen del sistema	flash0:c2900-universalk9-mz.SPA.151-1.M4.bin
Tamaño memoria no volátil NVRAM	255K
Tamaño memoria FLASH	249856K (250 MB)
Tamaño memoria RAM	491520K/32768K (500 MB)
Interfaces de red integradas	2 x GigaEthernet

Ejemplo

```
Router>
Router>en
Router#show version
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.1(4)M4, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thurs 5-Jan-12 15:41 by pt_team

ROM: System Bootstrap, Version 15.1(4)M4, RELEASE SOFTWARE (fc1)
cisco2901 uptime is 3 minutes, 43 seconds
System returned to ROM by power-on
System image file is "flash0:c2900-universalk9-mz.SPA.151-1.M4.bin"
Last reload type: Normal Reload

[...]

If you require further assistance please contact us by sending email to
export@cisco.com.
Cisco CISCO2901/K9 (revision 1.0) with 491520K/32768K bytes of memory.
Processor board ID FTX152400KS
2 Gigabit Ethernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
255K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)

[...]
```

La memoria se muestra con el formato 491520K/32768K; la primera es la memoria RAM que utiliza el sistema y la segunda hace referencia a la memoria para gestionar los paquetes en las colas de salida (búferes).

Se pueden encontrar dos grandes familias de publicaciones del software Cisco IOS, versiones 12.x o 15.y (esta última es la versión actual), aunque muchos dispositivos todavía implementan versiones 12.x. También hay otras variantes del sistema como IOS XE, que posiblemente será el sucesor del IOS, OS-XR, enfocado a hardware para proveedores de servicios, y NX-OS para hardware de centros de datos.

Enlace de interés

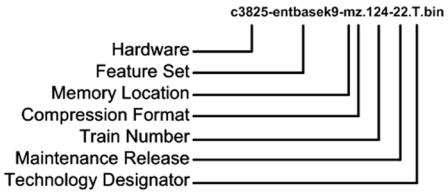
Sobre las redes IOS y NX-OS podéis consultar el enlace siguiente:https://bit.ly/2SlXZBk.

El nombre del archivo de imagen del sistema suele seguir una nomenclatura que también aporta información relevante. En la figura 6 se puede observar cómo se estructura el nombre de la imagen de un enrutador Cisco 3825.

Enlace de interés

En el enlace siguiente podéis encontrar una herramienta que permite comparar las funcionalidades entre dos versiones diferentes del sistema para cada dispositivo: https://bit.ly/2MJ8ARU.

Figura 6. Estructura del nombre de la imagen de un enrutador Cisco 3825



Fuente:https://www.cisco.com/c/en/us/about/security-center/ios-nx-os-reference-guide.html

Enlace de interés

Para más información al respecto podéis consultar los enlaces siguientes: Cisco IOS and NX-OS Software Reference Guide y Understanding Cisco IOS Naming Convention.

1.6. Sistema de archivos

El sistema de archivos o ficheros en general interactúa con la memoria Flash y puede transferir y recibir archivos del exterior, por ejemplo para hacer copias de respaldo remotas de las configuraciones o cargar actualizaciones del IOS.

Para la transferencia de archivos remotos se utiliza el protocolo TFTP (Trivial File Transfer Protocol), que es una versión ligera del protocolo FTP (File Transfer Protocol) que funciona sobre UDP (puerto 69) a nivel de Transport. Algunos modelos admiten también FTP para transferir los archivos o incluso HTTP para la descarga de archivos desde los entornos de gestión web.

El sistema de archivos también permite el intercambio de los archivos de configuración con las otras memorias de escritura RAM y NVRAM del dispositivo.

Algunos comandos de manipulación de los archivos de sistema de ficheros son:

- `dir <memoria>`: muestra el contenido de la memoria indicada.
- `mkdir <nombre directorio>`: crea un nuevo directorio en la memoria Flash.

- `copy <origen> <destino>`: copia el contenido del archivo origen al destino.
- `del <nombre archivo>`: borra el archivo de la memoria Flash.

Ejemplo

```
Router#dir ?
   WORD     Directory or file name
   flash0:  Directory or file name
   flash1:  Directory or file name
   flash:   Directory or file name
   nvram:   Directory or file name
   <cr>
Router#dir flash:
Directory of flash0:/

      3  -rw-    33591768          <no date>  c1900-universalk9-mz.SPA.151-4.M4.bin
      2  -rw-       28282          <no date>  sigdef-category.xml
      1  -rw-      227537          <no date>  sigdef-default.xml

255744000 bytes total (221896413 bytes free)
Router#
Router#mkdir backups
Create directory filename [backups]?
Created dir flash:backups

Router#copy ?
   flash:          Copy from flash: file system
   ftp:            Copy from ftp: file system
   running-config  Copy from current system configuration
   startup-config  Copy from startup configuration
   tftp:           Copy from tftp: file system
Router#copy startup-config flash:
Destination filename [startup-config]? backups/startup-config

608 bytes copied in 0.416 secs (1461 bytes/sec)
Router#dir flash:
Directory of flash0:/

      4  drw-           0          <no date>  backups
      5  -rw-         608          <no date>  backups/startup-config
      3  -rw-    33591768          <no date>  c1900-universalk9-mz.SPA.151-4.M4.bin
      2  -rw-       28282          <no date>  sigdef-category.xml
      1  -rw-      227537          <no date>  sigdef-default.xml

255744000 bytes total (221895805 bytes free)
Router#
```

```
Router#copy flash: tftp:
Source filename []? c1900-universalk9-mz.SPA.151-4.M4.bin
Address or name of remote host []? 150.150.150.150
Destination filename [c1900-universalk9-mz.SPA.151-4.M4.bin]?

Writing c1900-universalk9-mz.SPA.151-4.M4.bin....!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!![...]
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 33591768 bytes]

33591768 bytes copied in 3.952 secs (892458 bytes/sec)
Router#
```

La configuración de los dispositivos se basa en dos archivos especiales: el que está guardado en memoria no volátil (*startup-config*) y el que está cargado en la memoria RAM y se usa para la operación del dispositivo cuando está en marcha (*running-config*). Al iniciar el sistema, el archivo *running-config* se genera copiando en la memoria los contenidos del archivo *startup-config*.

Estos archivos contienen el texto con la secuencia de comandos IOS que habría que introducir por línea de comandos para llegar a la configuración actual.

Los comandos que se han de consultar y manipular en modo privilegiado son:

- `show <nombre archivo>`: muestra en pantalla el contenido del archivo.
- `copy <archivo origen> <archivo destino>`: copia el contenido del archivo de configuración origen en el archivo de configuración destino.
- `write`: persiste los cambios en ejecución, equivalente al comando `copy running-config startup-config`.
- `erase startup-config`: borra el archivo de configuración inicial de la memoria no volátil NVRAM.

Todo cambio en la configuración del dispositivo se hace siempre sobre el archivo *running-config*. Por lo tanto, si los cambios se quieren hacer persistentes, hay que copiar este archivo en *startup-config*.

Por otro lado, justo con el proceso inverso, copiar el archivo *startup-config* sobre *running-config*, podemos deshacer los cambios actuales y volver a poner el dispositivo en la configuración inicial. También se puede reiniciar sin guardar las nuevas configuraciones y el dispositivo iniciará sin estos cambios.

Ejemplo

```
Router#dir nvram:
Directory of nvram:/

No files in directory
```

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname RBCN
RBCN(config)#exit
RBCN#
%SYS-5-CONFIG_I: Configured from console by console

RBCN#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
RBCN#dir nvram:
Directory of nvram:/

   238  -rw-         531          <no date>  startup-config

531 bytes total (237588 bytes free)

RBCN#
RBCN#erase startup-config
Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]
[OK]
Erase of nvram: complete
%SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
RBCN#dir nvram:
Directory of nvram:/

No files in directory

RBCN#
```

2. Configuración de conmutadores en Cisco IOS

La primera vez que se accede a configurar un enrutador con IOS, este todavía no tiene ninguna configuración guardada en la memoria no volátil (NVRAM) y se activa el asistente de configuración inicial System Configuration Dialog, que permite configurar el dispositivo de manera guiada respondiendo a una serie de preguntas.

Ejemplo

```
          --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: yes

[...]

Would you like to enter basic management setup? [yes/no]: yes
Configuring global parameters:

   Enter host name [Router]: RBCN

   The enable secret is a password used to protect access to
   privileged EXEC and configuration modes. This password, after
   entered, becomes encrypted in the configuration.
   Enter enable secret: itsasecret
```

En general no utilizaréis este asistente porque es más lento. Con un poco de práctica es mucho más fácil hacer las configuraciones manualmente o cargar un archivo de configuración y luego hacer los ajustes necesarios a mano.

Para establecer cualquier configuración hay que acceder al modo de configuración, donde se puede configurar **nombre del dispositivo** `hostname WORD` y establecer una **clave de acceso** a este modo para evitar que cualquier usuario pueda hacer modificaciones sin autorización. Esta clave se puede guardar cifrada en el archivo de configuración `enable secret LINE` o en texto plano `enable password LINE`. Solo tenéis que configurar una de las dos opciones.

El nombre del dispositivo no es sensible a mayúsculas o minúsculas y no debe ser demasiado largo (menos de diez caracteres). Solo puede contener letras, dígitos y algunos caracteres especiales pero no puede contener espacios. En todo caso, la recomendación es que los nombres empiecen siempre por una letra seguida de letras o dígitos o guiones, si procede.

Enlace de interés

La guía de referencia de Cisco IOS recomienda seguir las directrices del documento RFC 1178 «Choosing a Name for Your Computer» para escoger los nombres de los dispositivos. La podéis consultar en: https://tools.ietf.org/html/rfc1178.

Otro aspecto importante que hay que saber a la hora de utilizar el IOS es que para **desactivar cualquier configuración** se utiliza la negación del mismo comando `no comando` en el mismo modo de configuración. Por ejemplo, para desactivar la clave cifrada de acceso al modo de configuración global tendréis que ejecutar `no enable secret` en este mismo modo.

Ejemplo

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname SW1
SW1(config)#
SW1(config)#enable secret itsasecret
SW1(config)#exit
SW1#
SW1#sh run | include secret|password
no service password-encryption
enable secret 5 $1$mERr$ILwq/b7kc.7X/ejA4Aosn0
SW1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW1(config)#no enable secret
SW1(config)#enable password itsasecret
SW1(config)#ex
SW1#
SW1#sh run | include secret|password
no service password-encryption
enable password itsasecret
SW1#exit

[...]

SW1>en
Password:
SW1#
```

Otra de las primeras configuraciones que deberéis hacer es el direccionamiento de red del dispositivo. Las direcciones IP se establecen en el modo de configuración específico de cada interfaz. Así, por ejemplo, en un enrutador

hay que acceder a la configuración de la interfaz `interface <nombre interfaz>` y luego indicar la dirección y la máscara `ip address A.B.C.D A.B.C.D`.

A menudo las interfaces de los dispositivos están desactivadas por defecto, y en este caso hay que activarlas con el comando `no shutdown`. Si la interfaz ya está activa, este comando no tiene ningún efecto, por lo tanto es una buena práctica que os acostumbréis a **activar siempre las interfaces después de configurar el direccionamiento**.

También es posible configurar el direccionamiento dinámico y que automáticamente se pidan los parámetros al servidor DHCP que haya disponible, en este caso el comando es `ip address dhcp`.

Una vez las direcciones están configuradas en el IOS, también hay disponibles algunas herramientas de utilidad para comprobar la conectividad, como el clásico `ping`, el comando de seguimiento de la traza de una ruta `traceroute` y los clientes `telnet` y `ssh` para establecer conexiones remotas. Encontraréis todos estos comandos tanto en modo usuario como en modo privilegiado.

Ejemplo

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface GigabitEthernet0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#do sh ip inter brief | include GigabitEthernet0/0
GigabitEthernet0/0     192.168.1.1     YES manual up                    up
R1(config-if)#
R1(config-if)#no ip address
R1(config-if)#ip address dhcp
%DHCP-6-ADDRESS_ASSIGN: Interface GigabitEthernet0/0 assigned DHCP address 192.168.1.10,
mask 255.255.255.0, hostname Router0

R1(config-if)#
R1(config-if)#do sh ip inter brief | include GigabitEthernet0/0
GigabitEthernet0/0     192.168.1.1    YES DHCP   up                    up
R1(config-if)#end
R1#
R1#ping 192.168.1.100

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.100, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/1 ms
```

```
R1#exit

...

R1>traceroute 192.168.1.100
Type escape sequence to abort.
Tracing the route to 192.168.1.100

   1   192.168.1.100   10 msec   0 msec    0 msec
```

Si bien el direccionamiento IP es necesario para el funcionamiento de los enrutadores, y cualquier interfaz activa debe tener como mínimo una dirección única en el contexto correspondiente, los conmutadores no necesitan ninguna configuración de red para su funcionamiento normal, que se desarrolla como enlace y se relaciona solo con las direcciones físicas (direcciones MAC).

Sin embargo, para poder acceder remotamente a un conmutador o para conectarse remotamente desde el mismo conmutador a otro dispositivo este debe disponer necesariamente de direccionamiento de red. A diferencia de los enrutadores, el direccionamiento de red de los conmutadores no se configura en una interfaz física, sino en alguna de las interfaces virtuales disponibles `VLAN <id>`. Aunque el dispositivo no tenga ninguna estructura de redes virtuales configurada, siempre tiene definida la interfaz virtual de la VLAN 1.

Ejemplo

```
Switch#conf t
Switch(config)#interface vlan 1
Switch(config-if)#
Switch(config-if)#ip address 192.168.1.2 255.255.255.0
Switch(config-if)#no shutdown
Switch(config-if)#exit
Switch(config-if)#do wr
```

Adicionalmente, en el direccionamiento del dispositivo puede ser necesario configurar la **puerta de enlace** de la red `ip default-gateway A.B.C.D` o también el **servidor de nombres** `ip name-server A.B.C.D` y el **nombre del dominio** `ip domain-name WORD`.

En todo caso, estas configuraciones solo afectan a la operativa local de administración, para poder comunicarse con equipos remotos o utilizar el nombre de *host* de estos equipos, en lugar de las correspondientes direcciones IP, pero en ningún caso afectan al desarrollo normal de la función principal del dispositivo y son opcionales.

Ejemplo

```
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname SW1
SW1(config)#ip default-gateway 192.168.1.1
SW1(config)#ip name-server 192.168.1.100
SW1(config)#ip domain-name office.test
SW1(config)#ip domain-lookup
SW1(config)#do wr
Building configuration...
[OK]
SW1(config)#exit
SW1#
%SYS-5-CONFIG_I: Configured from console by console

SW1#ping host1.office.test
Translating "host1.office.test"...domain server (192.168.1.100)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.33, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/2 ms
```

A menudo os pasará que ejecutaréis comandos incorrectos o en un modo que no les corresponde. En estos casos, cuando el IOS no encuentra el comando, activa la resolución de nombres (DNS) para comprobar si hay algún *host* con ese nombre. Esta consulta bloquea el CLI hasta que obtiene respuesta del servidor de nombres, o si no está configurado, que es lo más habitual, hasta que pasa el tiempo de espera, lo que puede ser exasperante.

Para evitar esta situación se puede desactivar la resolución de nombres, que se encuentra activada por defecto, utilizando la negación del comando `no ip domain-lookup`.

Ejemplo

```
Router#cmdnotfound
Translating "cmdnotfound"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address

Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#no ip domain-lookup
Router(config)#ex
Router#cmdnotfound
Translating "cmdnotfound"
```

```
% Unknown command or computer name, or unable to find computer address
```

2.1. Seguridad y acceso remoto

Anteriormente se ha explicado cómo asegurar el acceso al modo privilegiado estableciendo una contraseña y cómo, de manera análoga, se pueden asegurar también los diferentes accesos al CLI, por consola `line console 0` o el acceso remoto por medio de las líneas virtuales `line vty <X Y>`. Las líneas virtuales determinan el total de **conexiones remotas concurrentes** disponibles; normalmente hay hasta **dieciséis líneas virtuales** numeradas del 0 al 15, y se configuran tantas como sea necesario, o todas a la vez, por ejemplo, para configurar hasta cinco líneas concurrentes `line vty <0 4>`.

Es importante destacar que para que el acceso remoto sea funcional **el acceso al modo de configuración global debe estar protegido con una contraseña**. En caso contrario el IOS de Cisco implementa una restricción de seguridad que impide acceder al modo de configuración si el acceso al CLI es por alguna de las líneas virtuales `line vty`.

Desde la configuración específica de línea se pueden activar diferentes tipos de autenticación con el comando `login` o desactivarlos con la correspondiente `no login`. Por defecto, la autenticación básica sin parámetros `login` requiere un *password* que se guarda en texto plano en el archivo de configuración, comando `password LINE`.

Por otro lado, se pueden definir **usuarios locales** desde el modo de configuración global `username WORD [secret|password] LINE` y activar la **autenticación local** contra estos desde la configuración de línea `login local`. También son posibles otros métodos de autenticación más complejos contra servidores AAA (Authentication Authorization Accounting, por ejemplo, Radius o TACACS+), que quedan fuera del alcance de este manual.

Adicionalmente, se pueden filtrar los tipos de comunicaciones de entrada y salida permitidas con `transport [input|output] <protocol>`; por ejemplo, para limitar el acceso remoto al protocolo ssh podéis utilizar `transport input ssh`.

Ejemplo

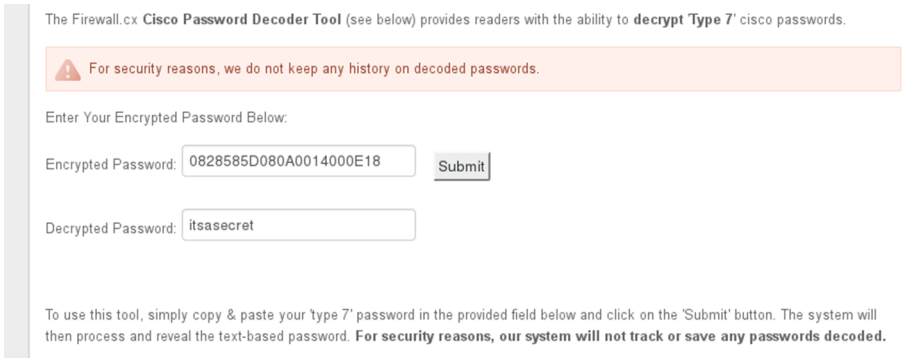
```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#enable password itsasecret
Router(config)#line console 0
Router(config-line)#password itsasecret
Router(config-line)#login
Router(config-line)#exit
```

```
Router(config)#username myuser password 1234
Router(config)#line vty 0 15
Router(config-line)#no login
Router(config-line)#exit
Router(config)#line vty 0 4
Router(config-line)#transport input ssh
Router(config-line)#login local
Router(config-line)#do wr
Building configuration...
[OK]
```

Como habéis podido ver anteriormente, en algunos casos es posible establecer las contraseñas para que se guarden cifradas en los archivos de configuración `secret` y, en otros casos, aparecerán sin cifrar `password`. Por esta razón, el IOS ofrece la posibilidad de cifrar todas las contraseñas que no estén cifradas, usando el comando `service password-encryption`.

Hay que tener en cuenta que el cifrado del servicio «password-encryption» es muy débil y hay muchas herramientas que permiten descifrar las contraseñas que hacen uso de este servicio. En la figura 7 podéis ver un ejemplo: «0828585D080A0014000E18» es el resultado de cifrar la clave «itsasecret».

Figura 7. Clave descifrada. Servicio «password-encryption»



Por otro lado, el IOS permite configurar mensajes de sistema que se muestran a los usuarios en diferentes situaciones y que se pueden utilizar como información o advertencia, a la vez que evitan, por ejemplo, que en caso de un acceso no autorizado el responsable pueda alegar desconocimiento.

Para configurar los mensajes se utiliza el comando `banner`: por ejemplo, se puede mostrar una advertencia justo antes de pedir las credenciales de acceso al sistema `banner login LINE` o añadir una información que se muestre siempre que un usuario acceda al sistema, aunque no se requiera autenticación `banner motd LINE`. Los mensajes comienzan y terminan indicando el mismo delimitador y pueden contener espacios y saltos de línea.

Ejemplo

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#no ena secret
Router(config)#ena password itsasecret
Router(config)#service password-encryption
Router(config)#ex
Router#
Router#sh run | include password
service password-encryption
enable password 7 0828585D080A0014000E18
username myuser password 7 08701E1D5D
password 7 0828585D080A0014000E18
Router#
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#banner motd #
Enter TEXT message.  End with the character '#'.
This is the Message of the day #

Router(config)#

....


Press RETURN to get started.

This is the Message of the day

Router>
```

Finalmente, cabe comentar los servicios de acceso remoto que implementan los dispositivos: telnet y ssh. Por lo que respecta al **telnet**, no es necesaria ninguna configuración, el servicio está activo y es funcional en todos los dispositivos. Esto no evita ciertos requerimientos, que se han comentado con anterioridad, para poderlo usar:

- El dispositivo debe disponer de un direccionamiento de red que permita la comunicación con los equipos remotos.

- Se debe establecer una contraseña para el acceso a la configuración global; de lo contrario, ningún equipo remoto podrá acceder a este modo.

El servicio **ssh**, además de los requerimientos anteriores, para poder funcionar necesita que se genere la pareja de claves asimétricas pública-privada con el comando `crypto key generate rsa`. El algoritmo para generar las claves es RSA (Rivest-Shamir-Adleman) y hay que tener en cuenta lo siguiente:

- Antes de generar las claves, el dispositivo debe tener configurados el nombre de *host* `hostname` y el dominio IP `ip domain-name`. El nombre de las claves resultantes tiene el formato del nombre totalmente cualificado FQDN (Fully Qualified Domain Name) del dispositivo. HOSTNAME.DOMAIN-NAME.

- A la hora de generar las claves hay que indicar su tamaño en bits. Los valores habituales son 512, 1024 o 2048. A medida que se incrementa el tamaño de las claves también tardan más en generarse, y dependiendo del dispositivo las claves de 2048 bits pueden tardar minutos o incluso horas.

- Para activar la versión 2 del protocolo ssh disponéis del comando `ip ssh version 2`.

- También se puede configurar, por ejemplo, el número máximo de intentos de autenticación antes de que la sesión se haya desconectado `ip ssh authentication-retries <0-5>`, y el tiempo en segundos para completar el inicio de sesión `ip ssh time-out <1-120>`.

Ejemplo

```
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname RBCN
RBCN(config)#enable secret itsasecret
RBCN(config)#ip domain-name example.com
RBCN(config)#crypto key generate rsa
The name for the keys will be: RBCN.example.com
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

RBCN(config)#ip ssh version 2
*març 1 0:8:26.509: %SSH-5-ENABLED: SSH 1.99 has been enabled

RLAN24(config)#
RLAN24#show ip ssh
```

```
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
RLAN24#
```

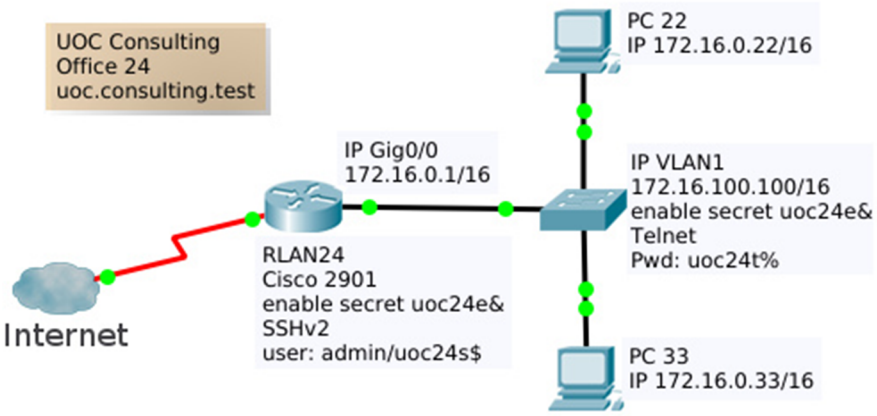
2.2. Caso práctico

Ved también

En el apartado 3 de este manual se tratará más detalladamente la configuración de los enrutadores.

En el caso práctico siguiente se propone configurar el acceso remoto por ssh y telnet a un enrutador y a un conmutador, respectivamente. En el esquema de la figura 8 se muestran algunos de los dispositivos de la red de una de las oficinas de la empresa ficticia UOC Consulting, con el dominio corporativo «uoc.consulting.test». En esta red los ordenadores se conectan a un conmutador, y este se conecta a su vez con el enrutador que hace las funciones de puerta de enlace y da acceso a internet al resto de dispositivos.

Figura 8. Esquema red **del caso práctico 1**. Oficina 24 UOC Consulting



En este mismo esquema se muestra el direccionamiento IPv4 y también el acceso remoto que se debe configurar tanto al enrutador como al conmutador para poder gestionarlos desde cualquier ordenador de la oficina. El enrutador RLAN24 debe estar accesible por ssh con las credenciales indicadas, mientras que el acceso al conmutador será por telnet utilizando la clave indicada.

Se pide implementar las configuraciones indicadas en todos los dispositivos y comprobar el funcionamiento de los accesos por ssh y telnet.

A continuación, se propone la solución implementada con el software de simulación de redes Packet Tracer. Se incluyen las configuraciones desde CLI de los dos dispositivos de red paso a paso.

Solución. Servicio ssh en el enrutador

```
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Router(config)#inte gig0/0
Router(config-if)#ip addr 172.16.0.1 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#hostname RLAN24
RLAN24(config)#enable secret uoc24e&
RLAN24(config)#ip domain-name uoc.consulting.test
RLAN24(config)#crypto key generate rsa
The name for the keys will be: RLAN24.uoc.consulting.test
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

RLAN24(config)#ip ssh version 2
*març 1 0:8:26.509: %SSH-5-ENABLED: SSH 1.99 has been enabled
RLAN24(config)#username admin secret uoc24s$
RLAN24(config)#line vty 0 15
RLAN24(config-line)#no login
RLAN24(config-line)#exit
RLAN24(config)#line vty 0 1
RLAN24(config-line)#transport input ssh
RLAN24(config-line)#login local
RLAN24(config-line)#do wr
Building configuration...
[OK]
RLAN24(config-line)#
```

Solución. Servicio telnet en el conmutador

```
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface vlan 1
Switch(config-if)#ip address 172.16.100.100 255.255.0.0
Switch(config-if)#no shutdown

Switch(config-if)#
%LINK-5-CHANGED: Interface Vlan1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up

Switch(config-if)#exit
Switch(config)#enable secret uoc24e&
Switch(config)#line vty 0 15
```

```
Switch(config-line)#no login
Switch(config-line)#exit
Switch(config)#line vty 0 1
Switch(config-line)#password uoc24t%
Switch(config-line)#login
Switch(config-line)#do wr
Building configuration...
[OK]
Switch(config-line)#
```

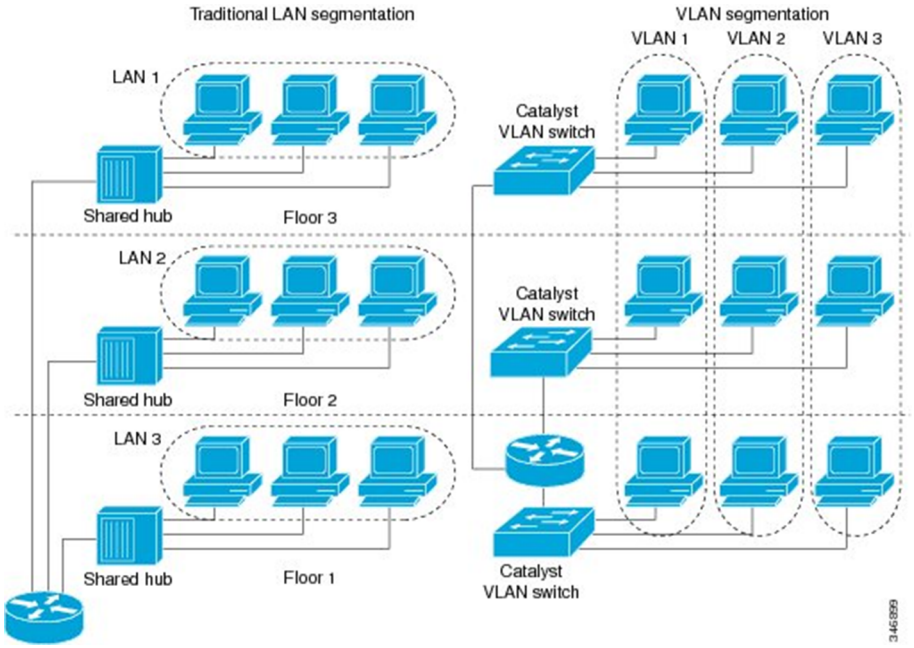
2.3. Configuración de conmutadores: VLAN

Una de las principales funcionalidades de los conmutadores es la posibilidad de dividir los dispositivos de una red local en segmentos o grupos funcionales más pequeños sin necesidad de tener en cuenta su ubicación física, ni tener que incrementar el hardware.

Cada segmento se llama **red virtual o VLAN** y tiene asignado un identificador numérico propio. El término "virtual" hace referencia a que, si bien todos los dispositivos en una red local están conectados entre sí utilizando conmutadores, es decir, se puede seguir un camino físico entre dos dispositivos cualesquiera de la red, los miembros de una red virtual se comportan como si estuvieran solos sin compartirla con nadie más.

Por lo tanto, la implementación de redes virtuales aporta una mayor flexibilidad a la hora de organizar los equipos de una red corporativa y garantiza que las comunicaciones de cada una de ellas estarán aisladas del resto, ya que el segmento de la red que conforma cada VLAN es un dominio de difusión independiente. Podéis ver un ejemplo de esto en la figura 9.

Figura 9. Comparativa de la segmentación tradicional respecto a la implementación con VLAN



Fuente: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/lanswitch/configuration/15-s/lsw-15-s-book/lsw-vlan-cfg-rtg.html

Para empezar, hay que identificar claramente en cuántos segmentos o grupos de equipos se dividirá la red, y a qué grupo pertenece cada equipo. Para implementar desde el IOS de un conmutador el esquema de redes virtuales hay que crear una VLAN para cada grupo y asignarle un número: desde el modo de configuración global con el comando `vlan <id>` se crea la VLAN y entra en el modo de configuración específico, desde donde opcionalmente se le puede dar un nombre, `name WORD`.

Los puertos de los conmutadores en que se conectan los equipos se asignan a las VLAN correspondientes. Estos tipos de puertos que conectan equipos se denominan **puertos de acceso**. En cada conmutador solo hay que definir las VLAN de los dispositivos que se conectan directamente a sus puertos, aunque si queréis podéis crearlas todas.

En el modo de configuración de cada interfaz se establece su rol de acceso con el comando `switchport mode access`, y luego se indica a qué VLAN pertenece `switchport access vlan <id>`. **Un puerto de acceso solo puede pertenecer a una VLAN**.

Para verificar la distribución de VLAN de los puertos de un conmutador se pueden ejecutar los comandos `show vlan` o `show vlan brief`, que muestran a qué VLAN está asignado cada puerto, o también para consultar información concreta de una VLAN `show vlan id <id>` o `show vlan name WORD`.

Normalmente es una buena práctica configurar todos los puertos de las diferentes VLAN según los requerimientos, y es habitual que varios puertos compartan la misma configuración. Para facilitar dicha configuración, el IOS permite acceder a configurar **rangos de puertos** todos a la vez con el comando `interface range <tipo interfaz> <0-9> / <0-48> - <0-48>`.

Ejemplo

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 5
Switch(config-vlan)#name MANAGE
Switch(config-vlan)#vlan 10
Switch(config-vlan)#name RRHH
Switch(config-vlan)#exit
Switch(config)#interface range fastEthernet 0/1-10
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#exit
Switch(config)#interface fastEthernet 0/24
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 5
Switch(config-if)#do wr
Building configuration...
[OK]
Switch(config-if)#end
Switch#show vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Gig0/1, Gig0/2
5    GESTIO                            active    Fa0/24
10   RRHH                              active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10

[...]

Switch#
Switch#show vlan id 10

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
10   RRHH                              active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10
```

```
VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
10   enet  100010     1500  -      -      -        -    -        0      0
```

A la hora de conectar dos conmutadores entre ellos, aparece la necesidad de comunicar dispositivos de las mismas VLAN repartidos entre ambos conmutadores. En una primera aproximación se podría usar un puerto de cada conmutador de las diferentes VLAN definidas para conectarlos entre ellos. Es fácil ver que esto tiene el inconveniente de tener que utilizar tantos puertos de los conmutadores como VLAN estén definidas para hacer esta interconexión, y es un desperdicio de recursos.

Para resolver el problema anterior se puede configurar un único puerto que permita el tráfico de datos de múltiples VLAN. El comando en el modo de configuración de interfaz es `switchport mode trunk`, y para estos puertos llamados **troncales** se puede configurar el tráfico de las *VLAN* admitidas, mediante la indicación de un rango inclusivo `switchport trunk allowed vlan <id inicial> - <id final>` o también una lista discreta de los identificadores de VLAN `switchport trunk allowed vlan <ID1>, <ID2>, ...` También posteriormente se pueden añadir o quitar VLAN asignadas al puerto troncal `switchport trunk allowed vlan add <id>` o `switchport trunk allowed vlan remove <id>`.

Para poder gestionar el tráfico de datos de múltiples VLAN en un único puerto troncal, el conmutador que envía los datos por el puerto troncal previamente les añade la información de la VLAN a la que pertenece (*tag* o). Así, al recibir las tramas el receptor extrae esta información para identificar a qué VLAN corresponde y las vuelve a dejar tal como estaban originalmente, sin la etiqueta. En la figura 10 podéis observar cómo afectan estos cambios a la estructura de la trama Ethernet.

Figura 10. Estructura de la trama Ethernet VLAN 802.11q (dot1q)

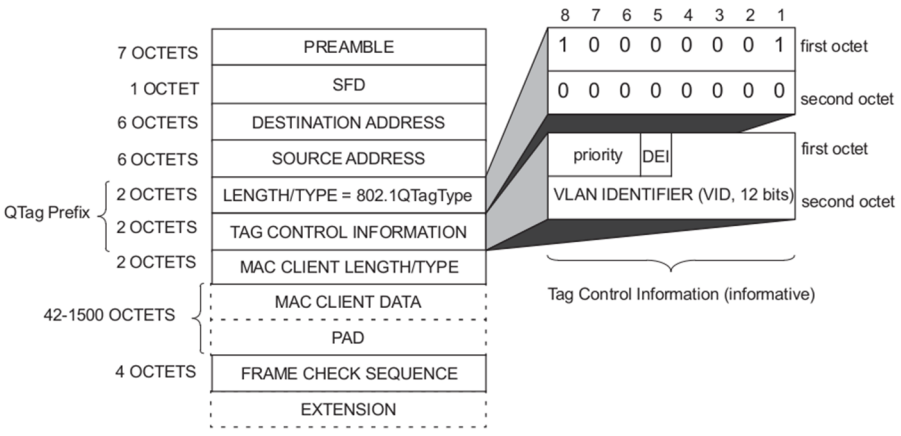


Figure G-1—Example of IEEE 802.3 MAC frame format

Fuente: https://standards.ieee.org/standard/802_1Q-2014.html

Tagging

El mecanismo de añadir la información de la VLAN a las tramas se llama ***tagging*** o etiquetado, razón por la que estos puertos troncales también se llaman ***tagged ports***. Los detalles de implementación de las redes virtuales se describe en la norma IEEE 802.1q, que es el origen del término dot1q.

Los puertos troncales también aceptan tramas sin etiqueta. En este caso el conmutador asigna estos datos a la VLAN nativa. Cada puerto troncal solo puede tener configurada una única VLAN nativa, que por defecto es la VLAN 1. Para configurarla tenéis el comando `switchport trunk native vlan <id>`.

En algunos dispositivos CISCO también se puede utilizar otro protocolo para encapsular las tramas en los enlaces troncales llamado ISL Inter-Switch Link. En este caso, habrá que establecer explícitamente el tipo de encapsulamiento `switchport trunk encapsulation [isl | dot1q]` en los puertos troncales. Este protocolo es propiedad de CISCO, mientras que la norma 802.1q es el estándar utilizado en todas partes y, por tanto, el método recomendado; además, ISL está en desuso y los nuevos modelos no lo implementan.

Cualquier trama que llegue a un puerto troncal desde la misma VLAN que tiene configurada el puerto troncal se transmite por este sin etiqueta. Es obligatorio que los dos puertos conectados al enlace troncal tengan configurada la misma VLAN nativa.

Para consultar la información de los puertos troncales disponéis del comando `show interfaces trunk` y para obtener información detallada de una interfaz concreta `show interfaces <nombre interfaz> switchport`.

Ejemplo

```
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 99
Switch(config-vlan)#exit
Switch(config)#interface gigabitEthernet 0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 5-20
Switch(config-if)#switchport trunk allowed vlan add 99
Switch(config-if)#do sh  int trunk
Port        Mode         Encapsulation  Status        Native vlan
Gig0/1      on           802.1q         trunking      99

Port        Vlans allowed on trunk
Gig0/1      5-99

Port        Vlans allowed and active in management domain
Gig0/1      5,10,20

Port        Vlans in spanning tree forwarding state and not pruned
```

```
Gig0/1      5,10,20
```

El hecho de que la estructura de las tramas de datos se modifique en los puertos troncales implica que estos puertos no son compatibles con los puertos de acceso de otros conmutadores, ni con los puertos de otros dispositivos como ordenadores o enrutadores.

En una red segmentada en varias VLAN independientes haría falta un enrutador para cada VLAN para gestionar las comunicaciones con el exterior (incluyendo los dispositivos de las otras VLAN), o bien utilizar un puerto troncal que admite múltiples VLAN conectado a un único enrutador, pero hay que configurar el puerto del enrutador para entender las tramas etiquetadas que le llegan por este enlace troncal.

Así, para configurar un puerto de un enrutador conectado a un puerto troncal de un conmutador, primero a partir del puerto físico del enrutador, hay que crear tantas subinterfaces como VLAN admita el troncal. La instrucción es `interface <nombre interfaz>.N`, donde `.N` es un sufijo numérico. Es una buena práctica escoger para el sufijo el mismo valor del identificador de la VLAN que gestionará la subinterfaz.

Cada subinterfaz actuará como puerta de enlace de los dispositivos de una VLAN, por tanto tendrá una dirección lógica de la misma red. Además, para que acepte los datos etiquetados con la información de la VLAN, se configurará el encapsulamiento con el comando `encapsulation dot1Q <id>`, indicando el identificador de esta. Hay otras soluciones al encaminamiento entre VLAN que no se detallarán en este manual.

Ejemplo

```
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface gigabitEthernet 0/1
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

Router(config-if)#exit
Router(config)#interface gigabitEthernet 0/1.5
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1.5, changed state to up

Router(config-subif)#encapsulation dot1Q 5
Router(config-subif)#ip address 192.168.5.1 255.255.255.0
```

```
Router(config-subif)#no shutdown
Router(config-subif)#do wr
Building configuration...
[OK]
Router#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0     unassigned      YES unset  administratively down down
GigabitEthernet0/1     unassigned      YES unset  up                    down
GigabitEthernet0/1.5   192.168.5.1     YES manual up                    down
Vlan1                  unassigned      YES unset  administratively down down
Router#
```

2.4. Configuración de conmutadores: tabla MAC

La tabla de direcciones MAC es la tabla que utiliza el conmutador para su función principal, que es conmutar las tramas que llegan por un puerto hacia uno de los otros puertos de salida. En la tabla MAC se guarda información de las direcciones físicas de los dispositivos como, por ejemplo, por qué puerto hay que enviarle los datos, en qué VLAN está y el tipo u origen de esta información. La tabla se puede consultar con el comando `show mac-address-table`.

Sin hacer ninguna configuración, el conmutador es capaz de gestionar la información de esta tabla de forma automática a partir de las direcciones origen de las tramas que recibe por los diferentes puertos. En este caso los datos aparecerán en la tabla como dinámicas `DYNAMIC`. Esta información registrada automáticamente tiene caducidad y es volátil. Si pasa cierto tiempo sin utilizarse, se borra de la tabla, y si se reinicia el dispositivo también. El contenido dinámico también se puede borrar con el comando `clear mac-address-table dynamic`.

Siempre que el conmutador recibe una trama y debe decidir a qué puerto enviarla, consulta la tabla de direcciones MAC. Si el destinatario no aparece, lo envía a todos los puertos para que la transmitan esperando encontrarlo, y esta trama se difunde por toda la red. La mayor parte de este tráfico será finalmente descartado. Para evitar la difusión de tramas por toda la red buscando dispositivos que no se encuentran en la tabla de direcciones MAC, se puede añadir información directamente en la tabla. Estos datos se muestran como estáticos `STATIC` en la tabla y son permanentes, no se borran ni siquiera después de un reinicio. El comando para añadir información estática de un dispositivo es `mac-address-table static H.H.H vlan <id> interface <nombre interfaz>`.

Se puede vaciar temporalmente la tabla de todas las entradas dinámicas o estáticas con el comando `clear mac-address-table`, pero hay que tener en cuenta que las configuraciones estáticas están guardadas en el archivo de configuración y, por lo tanto, después de reiniciar volverán a aparecer. Para desactivar una entrada estática hay que utilizar la negación del comando:

```
no mac-address-table static H.H.H vlan <id> interface <nombre interfaz>
```

Solo tiene sentido hacer esta configuración para añadir información de dispositivos relevantes en la red y que sean destinos habituales de las comunicaciones, por ejemplo la puerta de enlace, un servidor o una impresora.

Ejemplo

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#mac-address-table static 000C.000C.000C vlan 5 interface GigabitEthernet 0/1
Switch(config)#ex
Switch#
%SYS-5-CONFIG_I: Configured from console by console

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

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0040.0baa.ba19    DYNAMIC     Gig0/1
   5    000c.000c.000c    STATIC      Gig0/1
Switch#clear mac-address-table dynamic
Switch# sh mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   5    000c.000c.000c    STATIC      Gig0/1
```

3. Configuración de enrutadores Cisco IOS

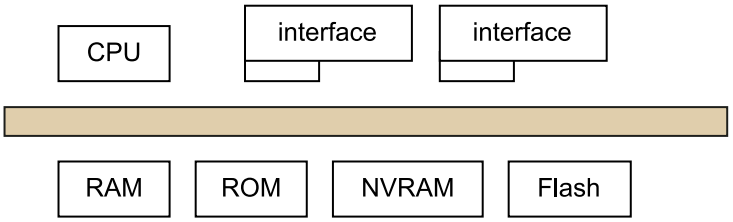
Este apartado sirve como una breve introducción a la operación de enrutadores con el sistema Cisco IOS, empleado por los dispositivos Cisco. Si bien este sistema operativo es complejo y tiene muchas opciones, los conceptos que se presentan aquí son los más básicos para poder seguir los objetivos de aprendizaje de la asignatura si partimos desde cero.

3.1. Estructura de un enrutador (*router*)

Un enrutador (*router*, en inglés) IP es un computador especializado en conmutar datagramas IP. Dependiendo de las prestaciones que debe ofrecer, su estructura interna es más o menos compleja y especializada, pero para los modelos de gama baja, podemos pensar en una estructura similar a la de un PC: CPU, memoria, buses e interfaces de red. Para el almacenamiento de datos es habitual utilizar memoria ROM, memoria Flash y memoria RAM y RAM no volátil (NVRAM):

- RAM: código, tablas de encaminamiento, búferes, memoria caché ARP, etc.
- NVRAM (no volátil): archivo de configuración *startup-config*.
- Flash (no volátil): imagen del IOS.
- ROM (no volátil): parte de la imagen IOS, código *bootstrap*.

Figura 11. Diagrama de bloques de un enrutador Cisco



3.1.1. Consulta del estado del enrutador (comando `show`)

Podemos consultar el estado de un enrutador mediante variaciones del comando `show`. Este comando tiene muchas opciones y permite visualizar datos de todo tipo, por lo que es muy útil usar la ayuda para listarlos (opción `?`).

Dependiendo del tipo de información que queremos consultar, el comando es ejecutable desde el modo usuario, o bien necesitamos los privilegios del modo privilegiado.

Las opciones que utilizaremos más habitualmente son:

- `show running-config`: muestra el archivo de configuración que está activo en el enrutador.
- `show startup-config`: muestra el archivo de configuración que está grabado en la NVRAM.
- `show ip <parameter>`: muestra los parámetros asociados a la configuración del protocolo IP.
- `show ip route`: muestra la tabla de encaminamiento IP.
- `show ip interface brief`: muestra un resumen del estado de las interfaces.
- `show interfaces`: muestra información detallada de las interfaces de red.
- `show interface <nombre>`: muestra información detallada de una interfaz concreta.
- `show cdp neighbors`: permite listar dispositivos Cisco vecinos conectados.

La tabla de encaminamiento es una información que no se considera privilegiada y que puede ser consultada desde el modo usuario. Sin embargo, el contenido de los archivos de configuración sí se considera privilegiado y estos archivos solo pueden ser visualizados desde el modo privilegiado.

3.2. Configuración de enrutadores con Cisco IOS

Los siguientes subapartados introducen la configuración genérica de interfaces a los enrutadores, junto con la configuración de interfaces Ethernet y la configuración de interfaces serie. También se indica el método para consultar la tabla ARP.

3.2.1. Configuración de interfaces

Un enrutador puede tener diferentes interfaces de diferentes protocolos de nivel 2 (Ethernet, Serial, ADSL). Cada una de estas interfaces se puede configurar solo con una dirección IP.

A nivel básico, para configurar una interfaz de red solo es necesario indicar su dirección IP, la máscara de subred, y activarla:

- `ip address <dirección IP> <máscara de subred>`: configura la dirección IP.
- `no shutdown/shutdown`: activa o desactiva la interfaz.

El comando `ip address` permite asignar una dirección IP a una interfaz. La configuración de las interfaces se hace desde el modo de configuración de un enrutador.

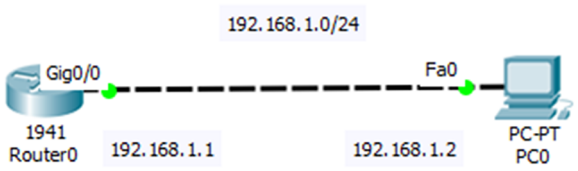
El comando `no shutdown` es necesario para activar la interfaz. Por defecto, al arrancar el enrutador todas las interfaces están desactivadas. El comando `shutdown`, en su defecto desactivaría administrativamente una interfaz.

3.2.2. Configuración de interfaces Ethernet

Ejemplo 1: configuración de un enlace Ethernet entre un enrutador y un ordenador

En la figura 12a se muestra un enlace Ethernet que conecta la interfaz GigabitEthernet0/0 del Router0 y la interfaz FastEthernet0 del ordenador PC0.

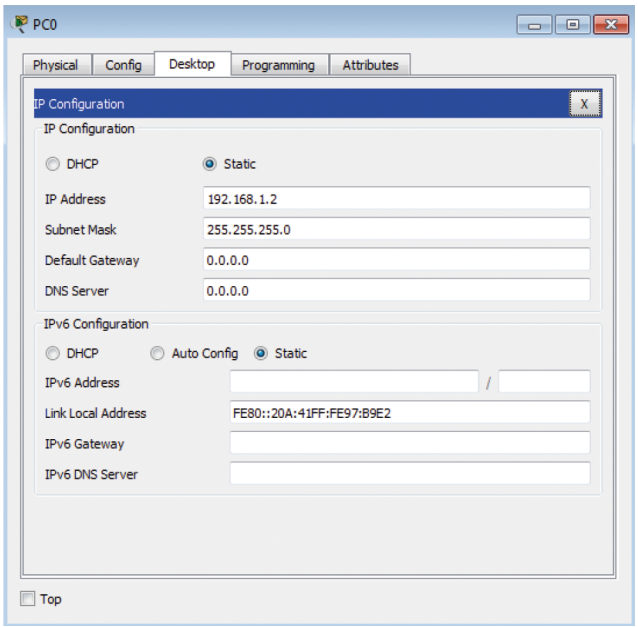
Figura 12a. Enlace Ethernet entre un enrutador y un ordenador



Las configuraciones de las interfaces del enrutador y del ordenador se muestran en la figura 12b. Por ejemplo, para configurar la interfaz del enrutador se ha utilizado el comando `ip address`, y para activarla, `no shutdown`.

Figura 12b. Configuración de las interfaces del enrutador Router0 y del ordenador PC0

```
Router>enable
Router#configure terminal
Router(config)#interface Gig0/0
Router(config-if)#ip address
192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#exit
Router#
```

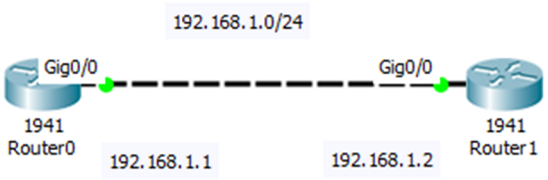


Una vez configuradas las dos interfaces, se puede hacer un *ping* con éxito entre los dos dispositivos.

Ejemplo 2: configuración Ethernet entre dos enrutadores

En la figura 13a se muestra un enlace Ethernet que conecta la interfaz GigabitEthernet0/0 de Router0 con la interfaz GigabitEthernet0/0 de Router1.

Figura 13a. Conexión Ethernet entre dos enrutadores



En la figura 13b se muestran las configuraciones de las interfaces Ethernet de los dos enrutadores, donde los comandos `ip address` y `no shutdown` son imprescindibles para hacer esta tarea. Una vez configuradas las interfaces, ya se consigue visibilidad a nivel IP entre los dos equipos.

Figura 13b. Configuraciones de las interfaces GigabitEthernet de Router0 y Router1

```
Router>enable
Router#configure terminal
Router(config)#hostname Router0
Router0(config)#interface Gig0/0
Router0(config-if)#ip address
192.168.1.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#
```

```
Router>enable
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#interface Gig0/0
Router1(config-if)#ip address
192.168.1.2 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#
```

3.2.3. Configuración de interfaces serie

Las interfaces serie están diseñadas para que en la situación más normal se conecten a un operador de telecomunicaciones por medio de un DCE (es decir, un módem o una terminación de red, TR). El DCE es lo que normalmente da el reloj y por tanto fija la velocidad de modulación y de transmisión.

Si se conectan dos puertos serie de enrutador (DTE-DTE) se debe utilizar un cable serie, donde uno de los dos puertos debe actuar como DCE dando el reloj. En principio, desde el punto de vista del enrutador, cualquiera de los dos puede actuar como DCE, así que es importante saber qué conector del cable marcará el puerto que actuará como DCE.

Una vez se sabe cuál es el puerto que actúa como DCE, este debe aportar el reloj. Esta opción se activa con el comando `clock rate Bw`, donde Bw indica la velocidad (bits por segundo) a la que debe trabajar la línea. En el puerto DTE no se debe ejecutar este comando.

Ejemplo: configuración de enlace serie entre dos enrutadores

En la figura 14a se muestra un enlace serie entre la interfaz Serial SE0/1/0 de Router0 y la interfaz Serial SE0/1/0 de Router1. La configuración de las dos interfaces se muestra en la figura 14b. Las interfaces *Serial* se configuran de forma similar a las interfaces *Ethernet* (con los comandos `ip address` y `no shutdown`), con la diferencia de que hay que fijar la velocidad de transmisión en la interfaz DCE. En este caso, la interfaz *Serial* de Router0 es la que actúa como DCE y donde hay que asignar la velocidad con el comando `clock rate 4000000`. La interfaz *Serial* de Router1 es DTE y no hay que hacer nada más.

Una vez configurado el enlace serie, los dos enrutadores se pueden comunicar a nivel IP.

Figura 14a. Conexión serie entre dos enrutadores

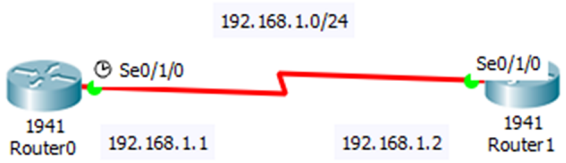


Figura 14b. Configuración de las interfaces serie de los dos enrutadores

```
Router>enable
Router#configure terminal
Router(config)#hostname Router0
Router0(config)#interface se0/1/0
Router0(config-if)#ip address
192.168.1.1 255.255.255.0
Router0(config-if)#clock rate 4000000
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#
```

```
Router>enable
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#interface se0/1/0
Router1(config-if)#ip address
192.168.1.2 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#
```

3.2.4. Tabla ARP

Para consultar la tabla ARP de un enrutador se puede usar el comando `show arp`, y para borrar las entradas se pueden utilizar los comandos `clear arp` y `clear arp-cache`.

Ejemplo: tabla ARP de un enrutador

La red de la figura 15a está formada por dos ordenadores PC0 y PC1 conectados vía Ethernet a un mismo enrutador Router0. Las configuraciones de las interfaces de los enrutadores y los dos ordenadores se muestran en la figura 15b.

Figura 15a. Red formada por dos ordenadores conectados a un enrutador

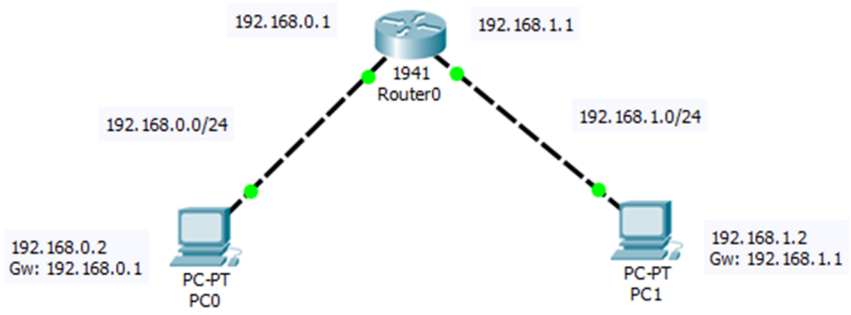
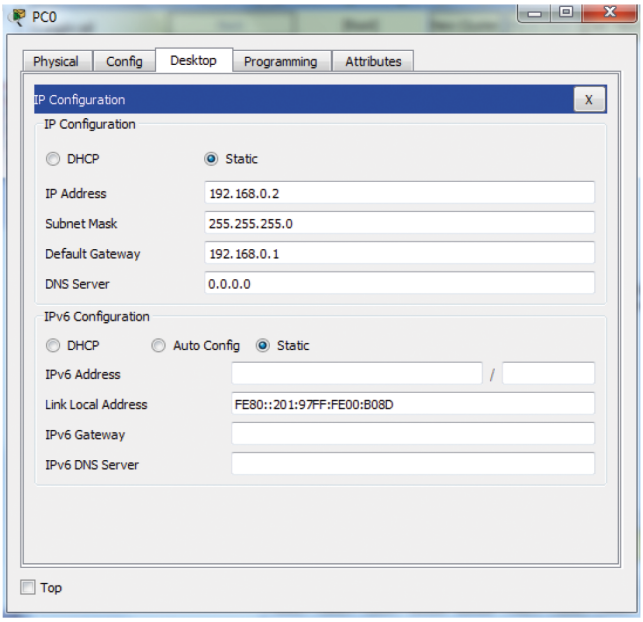


Figura 15b. Configuración de los dos ordenadores PC0 y PC1 y de Router0

```
Router>enable
Router#configure terminal
Router(config)#interface Gig0/0
Router(config-if)#ip address 192.168.0.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface Gig0/1
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)
```



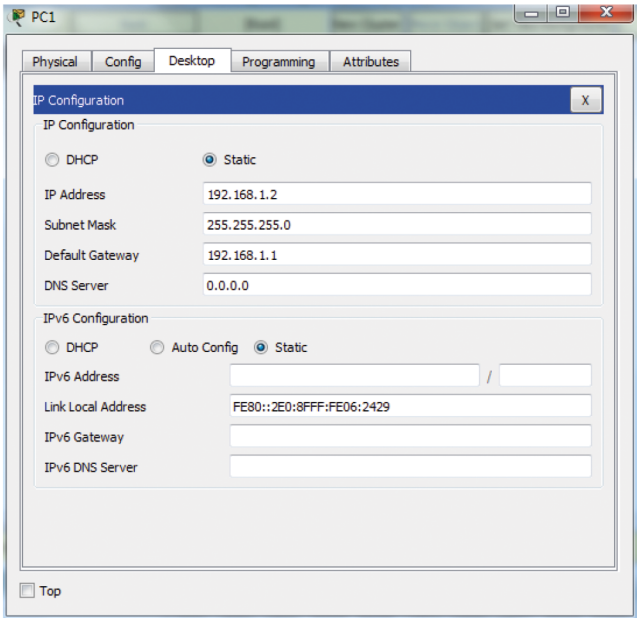
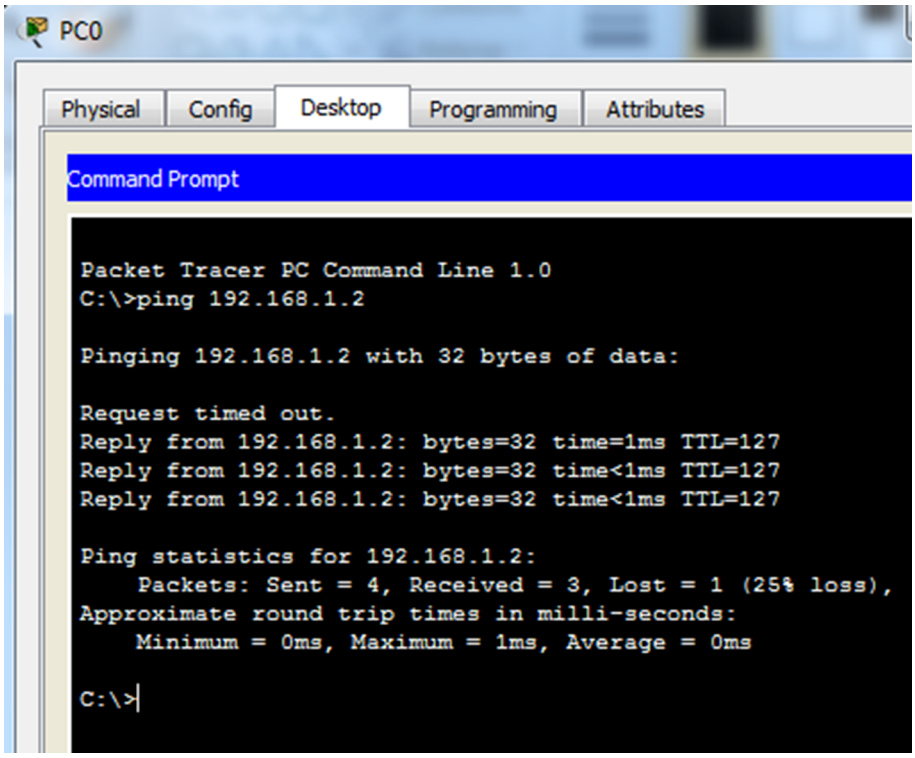


Figura 15c. *Ping* realizado desde PC0 hacia PC1



Una vez configuradas todas las interfaces, desde PC0 se realiza un *ping* hacia la dirección de PC1 (figura 15c). Este *ping* provoca que se hagan dos resoluciones ARP, una en cada red Ethernet, que permitirán al enrutador conocer las direcciones MAC de las interfaces de los dos ordenadores. El comando `show arp` permite visualizar estas direcciones MAC y las interfaces del enrutador por las que se han aprendido:

```
Router#show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.0.1 - 000C.856C.6201 ARPA GigabitEthernet0/0
Internet 192.168.0.2 0 0001.9700.B08D ARPA GigabitEthernet0/0
Internet 192.168.1.1 - 000C.856C.6202 ARPA GigabitEthernet0/1
Internet 192.168.1.2 0 00E0.8F06.2429 ARPA GigabitEthernet0/1
Router#
```

3.3. Encaminamiento estático

Una ruta estática es creada por un administrador de forma manual. Para configurar un encaminamiento estático a un enrutador se utiliza el comando `ip route` en modo de configuración:

```
Router(config)# ip route <IP red destino> <máscara destino>
<IP Gateway>.
```

Los parámetros del comando `ip route` son:

- Dirección de la red destino.
- Máscara asociada a esta red.
- Dirección de la interfaz del enrutador o la dirección del siguiente salto (*gateway*) por donde se debe encaminar el paquete.

Ejemplo: encaminamiento estático entre dos enrutadores

La red de la figura 16a está formada por tres subredes IP y dos enrutadores, de modo que cada enrutador solo está directamente conectado a dos subredes. Las configuraciones de las interfaces de los enrutadores Router0 y Router1 y de los ordenadores PC0 y PC1 se muestran en la figura 16b. Al tratarse de interfaces Ethernet se utilizan básicamente los comandos `ip address` y `no shutdown`, como hemos visto en apartados anteriores.

Figura 16a. Red formada por dos enrutadores y tres subredes IP

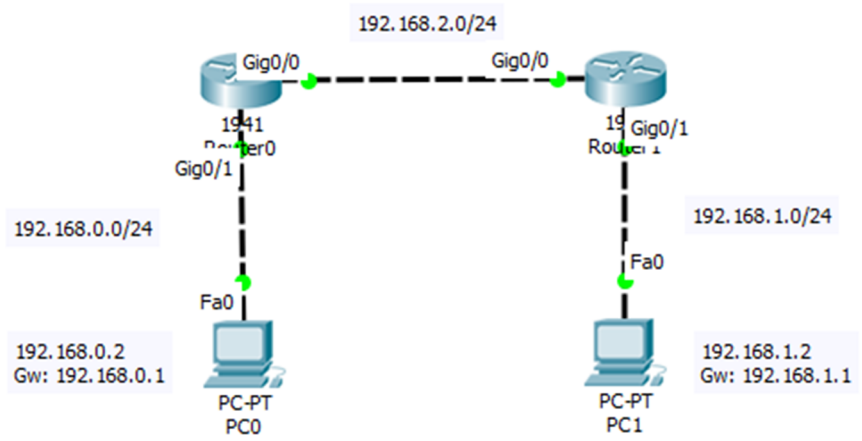
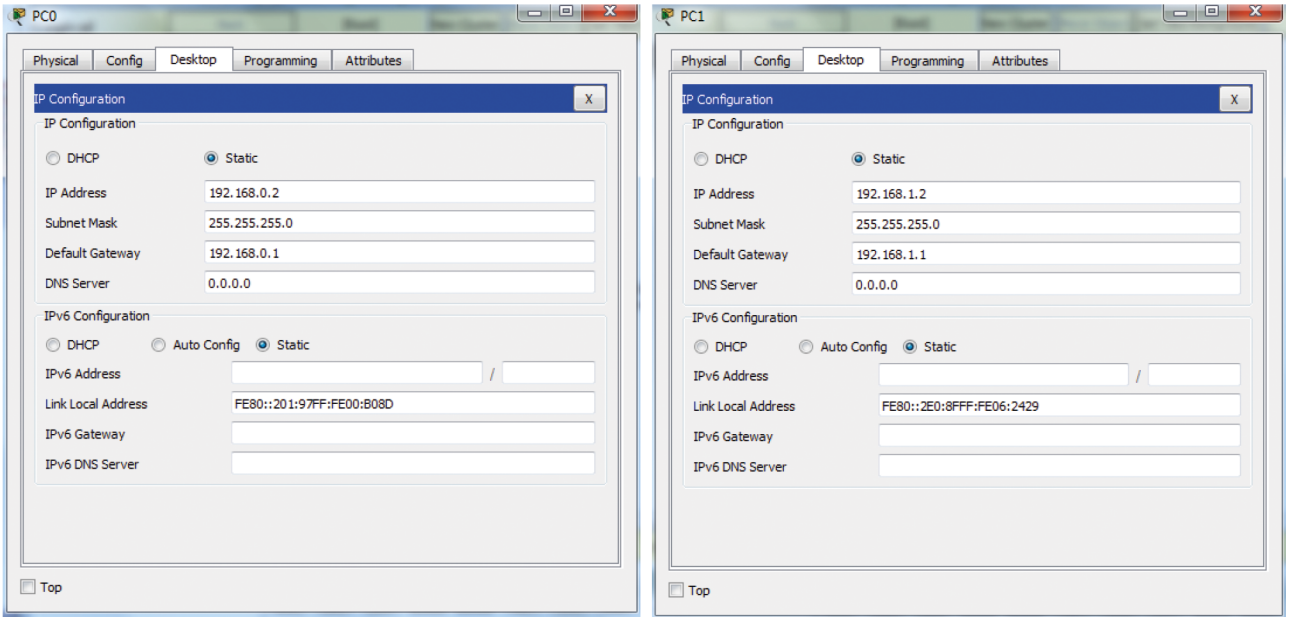


Figura 16b. Configuración de las interfaces de los ordenadores y enrutadores

```
Router#configure terminal
Router(config)#hostname Router0
Router0(config)#interface Gig0/1
Router0(config-if)#ip address
192.168.0.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#interface Gig0/0
Router0(config-if)#ip address
192.168.2.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
```

```
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#interface Gig0/0
Router1(config-if)#ip address
192.168.2.2 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#interface Gig0/1
Router1(config-if)#ip address
192.168.1.1 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
```



Es muy importante destacar que cuando se configura una interfaz de un enrutador, de manera automática se añade una entrada a su tabla de *routing*, que se corresponde con esta red directamente conectada.

Figura 16c. Tablas de *routing* del Router0 y Router1

```
Router0#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.0.0/24 is directly connected, GigabitEthernet0/1
L 192.168.0.1/32 is directly connected, GigabitEthernet0/1
192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.2.0/24 is directly connected, GigabitEthernet0/0
L 192.168.2.1/32 is directly connected, GigabitEthernet0/0
```

```
Router1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.1.0/24 is directly connected, GigabitEthernet0/1
L 192.168.1.1/32 is directly connected, GigabitEthernet0/1
192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.2.0/24 is directly connected, GigabitEthernet0/0
L 192.168.2.2/32 is directly connected, GigabitEthernet0/0
```

Si se ejecuta el comando `show ip route` en el Router0 (tal como muestra la figura 16c), se puede observar que por el hecho de haber configurado sus interfaces con las direcciones 192.168.0.1 y 192.168.2.1, se han añadido a su tabla de *routing* las redes directamente conectadas 192.168.0.0/24 y 192.168.2.0/24 con el distintivo 'C'. De manera similar, también se puede ver que Router1 tiene registradas las redes 192.168.1.0/24 y 192.168.2.0/24 a su tabla de *routing* con el distintivo 'C'.

Por lo tanto, se puede decir que Router0 conoce las redes 192.168.0.0/24 y 192.168.2.0/24, pero desconoce la red 192.168.1.0/24, y por otro lado, Router1 conoce las redes 192.168.1.0/24 y 192.168.2.0/24, pero no la red 192.168.0.0/24. Para una conectividad completa es necesario que todos los enrutadores conozcan todas las subredes implicadas en el encaminamiento de los paquetes.

Esta problemática se puede resolver de dos maneras:

- Con encaminamiento estático: un administrador de red añade manualmente las rutas que son desconocidas por las tablas de *routing*.

- Con encaminamiento dinámico: los protocolos de encaminamiento se encargan de propagar y mantener esta información entre las tablas de *routing* de los dispositivos.

Con encaminamiento estático y para que todas las rutas sean conocidas, se deberían ejecutar los siguientes comandos `ip route` en los enrutadores correspondientes:

```
Router1(config)#ip route 192.168.0.0 255.255.255.0
192.168.2.1

Router0(config)#ip route 192.168.1.0 255.255.255.0
192.168.2.2
```

3.3.1. Ruta por defecto (Default Gateway)

La ruta por defecto (o Default Gateway) evita que las tablas de los enrutadores tengan que almacenar todas las redes destino de internet. Normalmente un enrutador solo especifica las rutas más cercanas. El resto de rutas se indican mediante una ruta o Gateway por defecto. La ruta por defecto indica a un enrutador que cualquier paquete que no sepa cómo encaminar (destino desconocido), lo encamine por esta ruta.

Una ruta por defecto es aquella que tiene como red destino 0.0.0.0 y máscara 0.0.0.0, tal como muestra la tabla 3.

Tabla 3. Entrada de una ruta por defecto a una tabla de *routing* de un enrutador

Destino	Máscara	Gateway	Interfaz
0.0.0.0	0.0.0.0	IP Gateway	eth0

En un enrutador Cisco se puede introducir una ruta por defecto en su tabla de *routing* con el comando `ip route`, de la manera siguiente:

```
Router(config)#ip route 0.0.0.0 0.0.0.0 <IP Gateway>
```

La utilidad de la ruta por defecto se explica más claramente con el algoritmo de búsqueda de ruta, que utiliza el enrutador dentro de la tabla de *routing*, llamado Longest Prefijo Match (búsqueda por el prefijo más largo), y que se muestra en el código siguiente.

```
encontrado=false;
```

```
for i=1 to num_filas_tabla_routing {
     if (@IPdestino_datagrama AND Mascara[i] == Destino[i]) {
             Interface[i] ← Datagrama; //Se encamina el datagrama por la interfaz correspondiente
             encontrado =true;
             break;                                  //Es para la búsqueda
     }
}
if (!encontrado){                                 //Si no coincide con ninguna entrada de la tabla
      descarta_datagrama();               //Se descarta el datagrama
      envia_notificacion();                   //Se envía un mensaje ICMP de error
}
```

Un enrutador ordena las entradas de su tabla de *routing* según sus máscaras, de más específicas (255.255.255.255) a menos específicas (0.0.0.0) o, dicho de otro modo, de máscaras con más 1's a máscaras con menos 1's. Debido a esta ordenación, la ruta por defecto, en caso de que exista, ocuparía siempre la última entrada en la tabla de *routing*.

Cuando un enrutador recibe un datagrama, extrae su dirección IP destino, y accede a la tabla de *routing* para iniciar la búsqueda de una interfaz de salida. Empezando por la primera entrada, va comprobando que el AND bit a bit entre `IP_destino_datagrama` y la máscara de la entrada sea igual al destino de aquella entrada. Cuando encuentra una coincidencia, deja la búsqueda y encamina el paquete por la interfaz de la entrada coincidente. Si no encuentra ninguna coincidencia, este paquete se descarta0.

AND bit a bit

AND bit a bit es la operación lógica binaria que permite comprobar la coincidencia entre direcciones IP a gran velocidad, necesaria en cualquier enrutador, y en especial los que gestionan muchos paquetes.

Una ruta por defecto evitaría en última instancia que el paquete fuera descartado, ya que una entrada con dirección destino 0.0.0.0 y máscara 0.0.0.0 siempre encontrará coincidencia con cualquier IP destino (`IP_destino_datagrama AND 0.0.0.0 = 0.0.0.0`).

3.4. Encaminamiento entre VLAN

Definimos una VLAN como una red *broadcast*. Cada uno de los puertos de un enrutador es una red *broadcast* por definición y por tanto una red IP. Para ahorrar puertos de enrutador se pueden crear redes *broadcast* (redes IP) utilizando conmutadores que tengan módulos software de nivel 3. Esto significa que con un puerto de enrutador conectado a un conmutador se pueden crear tantas VLAN (redes *broadcast*) como el software del conmutador lo permita. Un conmutador Cisco de la gama 2950 permite crear hasta 1024 VLAN.

Es evidente que si un puerto de enrutador debe soportar N VLAN (N redes IP), el puerto deberá tener N direcciones IP, una por cada VLAN creada. También es evidente que para viajar desde una VLAN a otra se debe pasar obligatoriamente por el enrutador. Es decir, no se puede ir de una VLAN a otra directamente por medio del conmutador, al igual que el tráfico *broadcast* de nivel 2 (por ejemplo,

las tramas ARP) no se propagan entre VLAN diferentes. Para conseguir esta segmentación de nivel 3 se utiliza un protocolo específico llamado *trunking*. El estándar de *trunking* es el protocolo IEEE802.1Q.

Cuando encendemos un conmutador Cisco, todos los puertos pertenecen a la VLAN nativa. La VLAN nativa por definición es la VLAN-ID = 1. Si se define una VLAN para un uso específico es mejor usar otras VLAN-ID diferentes a 1.

A continuación, detallamos los pasos para realizar un encaminamiento entre VLAN:

1) Definición de las VLAN en un conmutador:

```
Sw(config)#vlan VLAN-ID                    //VLAN-ID puede tener el rango entre 0001 y 1005
Sw (config-vlan)# name VLAN-NAME
Sw (config-vlan)#exit
Sw (config)#
```

2) Asignación de VLAN a los puertos de un conmutador. Se debe utilizar el comando `switchport`:

```
Sw(config)# interface e0/1
Sw(config-if)# switchport mode access  //Define VLAN en modo estático
Sw(config-if)# switchport access vlan VLAN-ID  //Asigna el puerto a la VLAN con identificador VLAN-ID
Sw(config-if)# exit
```

3) Definición del enlace entre el conmutador y el enrutador como un enlace de tipo *trunk*. El enlace *trunk* es el que pertenece a todas las VLAN creadas. Debe estar asignado a la VLAN nativa (VLAN = 1):

```
Sw(config)# interface fastethernet0/1
Sw(config-if)# switchport mode trunk
Sw(config-if)# exit
Sw(config)# exit
```

4) Comandos para comprobar la configuración del conmutador:

- `show vlan`: muestra todas las VLAN existentes.
- `show vlan VLAN-ID`: muestra parámetros de una VLAN determinada.
- `show interfaces IFACE switchport`: muestra el modo administrativo (acceso estático), el modo de acceso de la VLAN (vlan-id), etc.
- `show interfaces IFACE trunk`: muestra parámetros de un enlace tipo *trunk*.

5) Configuración del enrutador para que entienda las VLAN creadas:

El enlace del enrutador debe ser de tipo *trunk* y debe tener tantas direcciones IP como VLAN hayan sido creadas. Por ello se deben crear tantas subinterfaces lógicas en la misma interfaz física como VLAN se hayan definido. A cada subinterfaz se le asignará una encapsulación dot1q (corresponden al protocolo de *trunking* IEEE 802.1Q), junto con el identificador de VLAN-ID. También se le asignará una IP:

```
R(config)# int fastethernet 0/0
R(config-if)# no ip address
R(config-if)# int fastethernet 0/0.1
R(config-subif)# encapsulation dot1q VLAN-ID2
R(config-subif)# ip address @IP2 MASK2
R(config-subif)# exit
R(config-if)# int fastethernet 0/0.2
R(config-subif)# encapsulation dot1q VLAN-ID3
R(config-subif)# ip address @IP3 MASK3
R(config-subif)# exit
R(config-if)# exit
R(config)# exit
```

En la tabla de *routing* debe haber una entrada para cada subinterfaz y su subred IP.

3.4.1. Características del VLAN Trunking Protocol

Algunas de las características del VLAN Trunking Protocol son:

- Fue desarrollado para gestionar la transferencia de tramas de diferentes VLAN en una única línea física o *trunk*.
- Evita enlazar un cable para cada VLAN entre dos conmutadores o entre un enrutador y un conmutador.
- Un enlace *trunk* agrupa múltiples enlaces virtuales sobre un único enlace físico (figura 17a), añadiendo etiquetas especiales en las tramas para identificar a qué VLAN pertenece (figura 17b).

Figura 17a. Ejemplo de VLAN Trunking Protocol

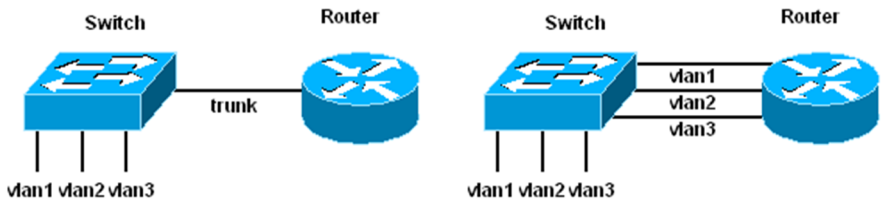
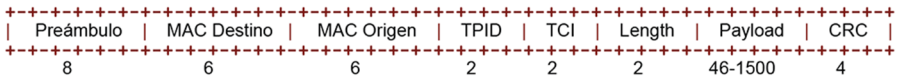


Figura 17b. Etiquetado de la trama Ethernet con el estándar IEEE 802.1Q



3.4.2. Estándar IEEE-802.1Q

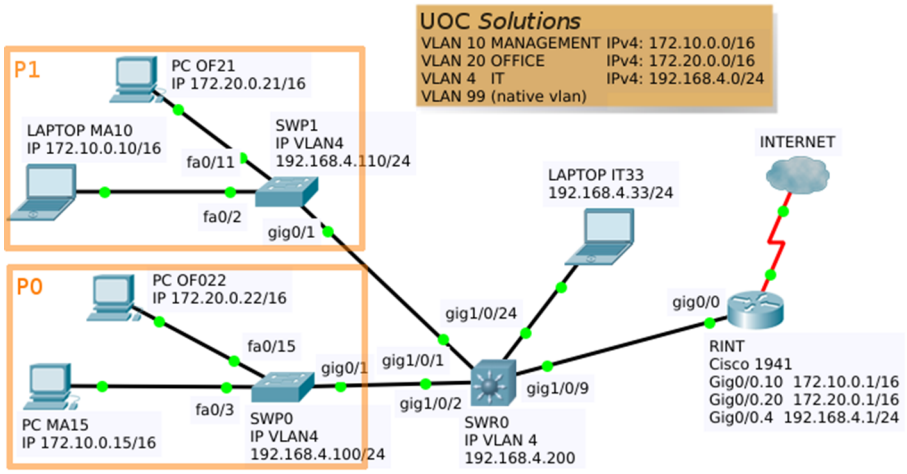
Añade una etiqueta de 4 bytes en la cabecera Ethernet de la trama para indicar a qué VLAN pertenece. El *tag* está formado por los campos siguientes (figura 17b):

- TPID (Tag Protocol Identifier): lleva el valor 0x8100 cuando el *tag* se ha añadido a una trama Ethernet.
- TCI (Tag Control Information): lleva el VLAN ID a la que pertenece la trama.

3.4.3. Ejemplo de encaminamiento entre diversas VLAN

En el siguiente caso práctico se propone configurar la distribución de redes virtuales (VLAN) de la empresa ficticia UOC Solutions, tal como se muestra en la figura 18.

Figura 18. Esquema red **del caso práctico 2**. UOC Solutions



Las oficinas de esta empresa ocupan dos plantas de un edificio comercial, los equipos de cada planta se conectan a un conmutador (SWP0 y SWP1), los conmutadores de planta se conectan por un enlace troncal al conmutador principal (SWR0), y este se conecta también por un enlace troncal al enrutador (RINT) que da el acceso a internet.

Los dispositivos de la red de esta empresa están segmentados en tres grupos: ordenadores de dirección (VLAN 10), ordenadores de empleados (VLAN 20) y equipamientos diversos del departamento de informática (VLAN 4).

Cada VLAN tiene asignada una red IPv4 para configurar sus equipos. En el esquema de la figura 18 se indican las direcciones de los equipos y de los dispositivos de red. Se han puesto solo unos cuantos equipos a modo de ejemplo.

En la tabla 4 se indica la distribución de puertos para los dos modelos de conmutadores.

Tabla 4. Configuración VLAN de puertos de los conmutadores UOC Solutions

Dispositivo	Puertos de acceso		Puertos troncales		VLAN nativa
	Puertos	VLAN	Puertos	VLAN	
Conmutadores SWP0 y SWP1	fa0 / 1-4	10	gig0 / 1-2	4-99	99
	fa0 / 5-23	20			99
	fa0 / 24	4			99
Conmutador SWR0	gig1 / 0/24	4	gig1 / 0 / 1-23	4-99	99

Se pide configurar la distribución VLAN en los conmutadores según los requerimientos indicados, y configurar el enrutador para que permita la comunicación entre todos los equipos de las diferentes VLAN.

A continuación se propone la solución implementada con el software de simulación de redes Packet Tracer. Se incluyen las configuraciones desde CLI paso a paso de los conmutadores SWP1 y SWR0, y del enrutador RINT, y las pruebas de conectividad entre los dispositivos finales.

Solución. Configuración conmutador SWP1

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname SWP1
SWP1(config)#vlan 4
SWP1(config-vlan)#name IT
SWP1(config-vlan)#vlan 10
SWP1(config-vlan)#name MANAGEMENT
SWP1(config-vlan)#vlan 20
SWP1(config-vlan)#name OFFICE
SWP1(config-vlan)#vlan 99
SWP1(config-vlan)#exit
SWP1(config)#interface range fastEthernet 0/1-4
SWP1(config-if-range)#switchport mode access
SWP1(config-if-range)#switchport access vlan 10
SWP1(config-if-range)#exit
SWP1(config)#interface range fastEthernet 0/5-23
```

```
SWP1(config-if-range)#switchport mode access
SWP1(config-if-range)#switchport access vlan 20
SWP1(config-if-range)#ex
SWP1(config)#interface fastEthernet 0/24
SWP1(config-if)#switchport mode access
SWP1(config-if)#switchport access vlan 4
SWP1(config-if)#exit
SWP1(config)#interface range gigabitEthernet 0/1-2
SWP1(config-if-range)#switchport mode trunk
SWP1(config-if-range)#switchport trunk allowed vlan 4-99
SWP1(config-if-range)#switchport trunk native vlan 99
SWP1(config-if-range)#exit
SWP1(config)#interface vlan 4
SWP1(config-if)#ip address 192.168.4.110 255.255.255.0
SWP1(config-if)#no shutdown
SWP1(config-if)#do wr
Building configuration...
[OK]
SWP1(config-if)#
```

Solución. Configuración conmutador SWR0

```
Switch>en
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#hostname SWR0
SWR0(config)#vlan 4
SWR0(config-vlan)#name IT
SWR0(config-vlan)#vlan 10
SWR0(config-vlan)#name MANAGEMENT
SWR0(config-vlan)#vlan 20
SWR0(config-vlan)#name OFFICE
SWR0(config-vlan)#vlan 99
SWR0(config-vlan)#exit
SWR0(config)#interface gigabitEthernet 1/0/24
SWR0(config-if)#switchport mode access
SWR0(config-if)#switchport access vlan 4
SWR0(config-if)#exit
SWR0(config)#interface range gigabitEthernet 1/0/1-23
SWR0(config-if-range)#switchport trunk encapsulation dot1q
SWR0(config-if-range)#switchport mode trunk
SWR0(config-if-range)#switchport trunk native vlan 99
SWR0(config-if-range)#switchport trunk allowed vlan 4-99
SWR0(config-if-range)#do wr
Building configuration...
Compressed configuration from 7383 bytes to 3601 bytes[OK]
[OK]
```

```
SWR0(config-if-range)#
```

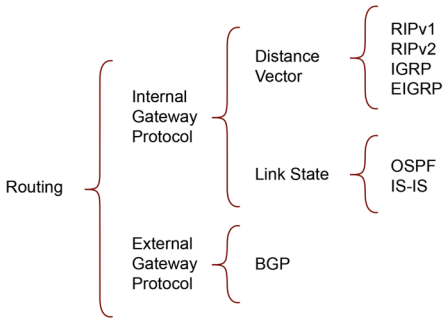
Solución. Configuración enrutador RINT

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname RINT
RINT(config)#interface gigabitEthernet 0/0.4
RINT(config-subif)#encapsulation dot1Q 4
RINT(config-subif)#ip address 192.168.4.1 255.255.255.0
RINT(config-subif)#no shutdown
RINT(config-subif)#exit
RINT(config)#interface gigabitEthernet 0/0.10
RINT(config-subif)#encapsulation dot1Q 10
RINT(config-subif)#ip address 172.10.0.1 255.255.0.0
RINT(config-subif)#no shutdown
RINT(config-subif)#exit
RINT(config)#interface gigabitEthernet 0/0.20
RINT(config-subif)#encapsulation dot1Q 20
RINT(config-subif)#ip address 172.20.0.1 255.255.0.0
RINT(config-subif)#exit
RINT(config)#interface gigabitEthernet 0/0.99
RINT(config-subif)#encapsulation dot1Q 99 native
RINT(config-subif)#do wr
Building configuration...
[OK]
RINT(config-subif)#
```

3.5. Configuración de los protocolos de encaminamiento

Los enrutadores deben construir una tabla de rutas para saber dónde redirigir los paquetes que viajan por la red. Para lograrlo, intercambian entre sí información sobre las redes que tienen directamente conectadas a sus interfaces. Para que un enrutador pueda llevar a cabo este intercambio, se debe configurar un protocolo de encaminamiento. En la figura 19 se muestra la clasificación de los principales protocolos de encaminamiento.

Figura 19. Clasificación de los principales protocolos de encaminamiento



Para ver la tabla de rutas actual de un enrutador se puede ejecutar, en modo privilegiado, el comando `show ip route`.

Hay diferentes tipos de protocolos de encaminamiento y un dispositivo solo puede intercambiar datos con otros con que comparta algún protocolo. Para configurar cada protocolo, hay que entrar en un modo especial de configuración de rutas, accesible desde el modo de configuración con la instrucción `router <nombre protocolo>`.

3.5.1. Distancia administrativa de un protocolo de encaminamiento

Es una medida utilizada en los enrutadores Cisco que define la fiabilidad de un protocolo de encaminamiento. Cuando hay dos o más rutas hacia un mismo destino, de diferentes protocolos de encaminamiento, la distancia administrativa ayuda en la elección de la mejor ruta (la más fiable). Las rutas con distancia administrativa baja son las más fiables. Por ejemplo, una ruta OSPF (ver la sección 2.5.2), con distancia administrativa 110, es considerada más fiable que una ruta RIP con distancia administrativa 120. La tabla 5 muestra las distancias administrativas por defecto que utilizan los enrutadores Cisco.

Tabla 5. Distancias administrativas que utilizan los enrutadores Cisco

Protocolo	Distancia administrativa
Directamente conectado	0
Ruta estática con Gateway de siguiente salto	1
Ruta EIGRP resumida	5
BGP externa	20
EIGRP interna	90
IGRP	100
OSPF	110
RIP	120

Protocolo	Distancia administrativa
EGP	140
EIGRP externa	170
BGP interna	200
Desconocida (descartada)	255

3.5.2. Encaminamiento con OSPF

OSPF (Open Shortest Path First) es un protocolo de encaminamiento de estado del enlace considerado de puerta de enlace interna. Hace uso del algoritmo SPF (llamado también de Dijkstra), para calcular la ruta más corta (de más bajo coste) entre dos nodos. Su métrica se denomina *cost*, y tiene en cuenta diversos parámetros tales como el ancho de banda y la congestión de los enlaces.

Para configurarlo en su forma más simple, se debe indicar un número de proceso en el comando `router`:

```
Router(config): router ospf <process id>
```

Este valor process ID no segmenta la red, solo sirve como identificador del proceso interno en el dispositivo. Por lo tanto, es de importancia local y puede diferir entre enrutadores.

Hay que publicar las redes conectadas al enrutador, que se introducen con el comando `network`:

```
Router(config-router): network 192.168.10.0 0.0.0.255 area 0
```

Sin embargo, en este caso se debe especificar una *wildcard*, que equivale a la máscara de subred, pero invirtiendo los 1's y los 0's. Por ejemplo, para publicar la subred 192.168.10.0/24, hay que usar como *wildcard* 0.0.0.255. También hay que añadir un parámetro con el número de área, que es realmente el valor que segmenta las zonas donde se publica OSPF (de forma similar al número de sistema en EIGRP, protocolo que veremos en el subapartado 3.5.3).

En todos los casos, en modo de configuración de rutas es posible desactivar la publicación en una interfaz con el comando `passive-interface`:

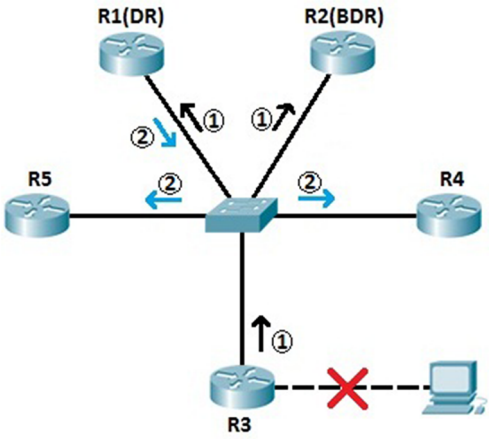
```
Router(config-router): passive-interface <nom interfície>
```

Esto evita desperdiciar ancho de banda en redes donde no hay ningún otro enrutador con el que intercambiar información.

Relaciones de adyacencia en OSPF: enrutadores DR y BDR

OSPF se basa en establecer y mantener relaciones de vecindad para que los enrutadores puedan compartir información de encaminamiento y del estado del enlace. Cada enrutador debe conectarse al menos a un enrutador en cada red IP a la que esté directamente conectado. Pero en redes multiacceso *broadcast* (como Ethernet), el número de adyacencias posibles crece cuadráticamente con el número de enrutadores existentes (figura 20).

Figura 20. Enrutadores DR y BDR en una red multiacceso Ethernet



La solución consiste en designar un enrutador que haga de portavoz de esa red o segmento (Designated Router o DR). Como esta solución representa tener un único punto de fallo, se elige un segundo enrutador designado como *backup* por si falla el principal (Backup Designated Router o BDR). Los enrutadores DR y BDR actúan como punto central para el intercambio de información de encaminamiento OSPF en una red multiacceso concreta. Cada encaminamiento solo intercambiará información (establecerá adyacencias) con el DR y el BDR. Estos distribuirán información de topología a cualquier otro enrutador dentro de la misma red multiacceso, reduciendo considerablemente el tráfico OSPF. Cuando el DR cae, el BDR se hace con el control y se elige otro BDR.

Elección DR/BDR

En los enlaces de punto a punto, no se elige un DR y un BDR, ya que solo hay dos enrutadores conectados directamente. En las redes LAN, se debe elegir un DR y un BDR que representen la red. Se utilizan dos reglas para elegir un DR y un BDR:

- El enrutador con la máxima prioridad OSPF se convertirá en un DR. Por defecto, todos los enrutadores tienen una prioridad de 1. Manualmente se puede configurar la prioridad OSPF con el comando `ip ospf priority <value>`.

- Si no se decide la prioridad, se elegirá el enrutador con el *router* ID más alto.

Asignación del *router* ID en un enrutador

Existe el siguiente orden de precedencia en la asignación del *router* ID en un enrutador:

- Es posible asignar manualmente el valor en cada proceso de OSPF con el comando `router-id X.X.X.X`.

- En caso de que no se tenga configurado de forma manual el *router* ID, los enrutadores utilizan la dirección IPv4 más alta configurada en las interfaces *loopback* para asignar el *router* ID (siempre que las interfaces no estén desactivadas).

- Por último, si no se tiene configurado el valor de *router* ID de manera manual, o no se ha configurado ningún interfaz *loopback* en estado activo, se utilizan las direcciones IPv4 de las interfaces físicas como valor para este parámetro. También es necesario que la interfaz se encuentre en estado activo.

Paquetes de OSPF

Algunos de los paquetes de OSPF son:

- *Hello*: ayudan a establecer y mantener relaciones de adyacencia con los enrutadores vecinos. Los paquetes *Hello* son emitidos cada diez segundos por la interfaz de un enrutador para indicar que continúa activo y para mantener y establecer relaciones de vecindad. Es el tiempo para determinar si un vecino ha caído o no.

- *LSA* o anuncio del estado del enlace: anuncian la relación entre el estado de las interfaces y adyacencias de un enrutador, con los cambios que se producen en la red. Avisan a los enrutadores de la red de los cambios de topología.

Funcionamiento básico de OSPF

Debemos tener en cuenta lo siguiente:

- Mediante el envío de LSA, los enrutadores intercambian su conocimiento de la red (métricas) con el resto de enrutadores de la red. Los LSA se envían por *flooding*: se reenvía un mensaje por todas las interfaces de un enrutador, excepto por la que ha llegado. Si a un enrutador le llega por segunda vez el mismo LSA, entonces lo descarta.

- Solo se envían ante cambios de la topología. Toda la red es advertida por medio de un LSA.

- Cada enrutador utiliza la información de los paquetes *Hello* y *LSA* recibidos para construir una base de datos topológica de la red.

- En la base de datos topológica se le aplica el algoritmo Dijkstra (SPF) para calcular la ruta más corta hacia un destino.

- Se obtiene el árbol SPF hacia todos los destinos donde el enrutador raíz es el enrutador local.

- Se seleccionan las mejores rutas del árbol SPF y se insertan en la tabla de *routing*.

Ejemplo de encaminamiento con OSPFv2

Dada la red de la figura 21a, se quiere activar el protocolo OSPF en los enrutadores Router0 y Router1 para que conozcan todas las rutas de la red.

En la figura 21b se muestra la configuración de las interfaces de los dos enrutadores y los dos ordenadores. Hay que tener en cuenta la velocidad de la interfaz serial DCE de Router0 con el comando `clock rate`. En la figura 21c se muestra la activación del protocolo OSPF en cada enrutador. Se ha utilizado la forma completa de activación, si bien es cierto que en este caso los comandos `router-id` y `passive-interface` no son necesarios o se podrían omitir. Después de configurar Router1, su sistema ya indica que ha contactado y establecido una relación de adyacencia con el enrutador vecino 1.1.1.1 (que es el Router0).

En la figura 21d se observan las tablas de *routing* de los dos enrutadores. Por ejemplo, Router0 tiene dos redes directamente conectadas (192.168.0.0/24 y 192.168.1.0/30) y ha aprendido la red 192.168.2.0/24 mediante OSPF, con una distancia administrativa de 110 y una métrica de 65. En la figura 21e se puede ver que la conectividad entre todos los ordenadores de la red es completa.

Figura 21a. Red formada por tres subredes y dos enrutadores.

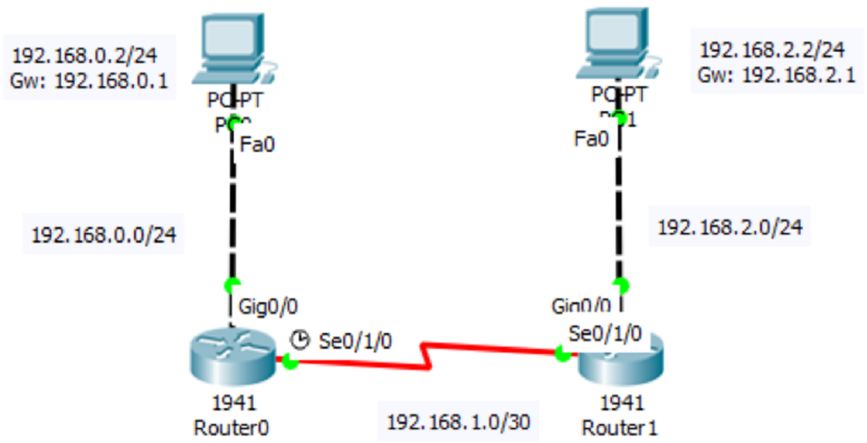
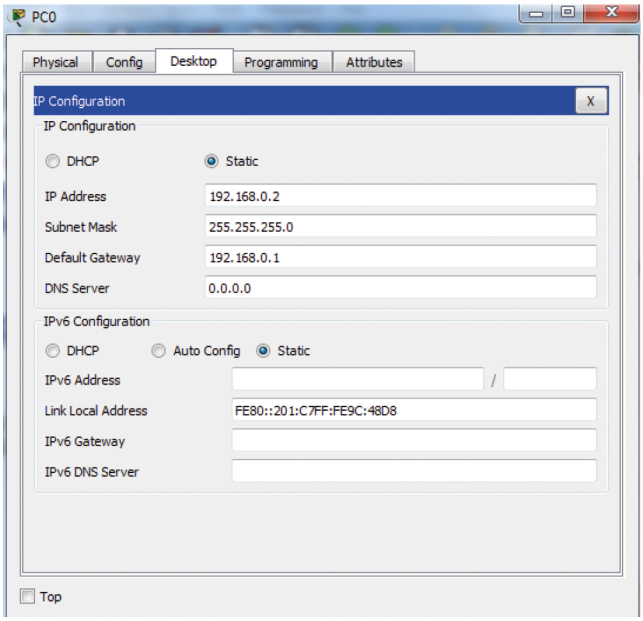


Figura 21b. Configuración de las interfaces de los enrutadores y los ordenadores

```
Router>enable
Router#configure terminal
Router(config)#hostname Router0
Router0(config)#interface Gig0/0
Router0(config-if)#ip address
192.168.0.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#interface se0/1/0
Router0(config-if)#clock rate 4000000
Router0(config-if)#no shutdown
Router0(config-if)#ip address
192.168.1.1 255.255.255.252
Router0(config-if)#exit
```

```
Router>enable
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#interface Gig0/0
Router1(config-if)#ip address
192.168.2.1 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#interface se0/1/0
Router1(config-if)#no shutdown
Router1(config-if)#ip address
192.168.1.2 255.255.255.252
Router1(config-if)#exit
```



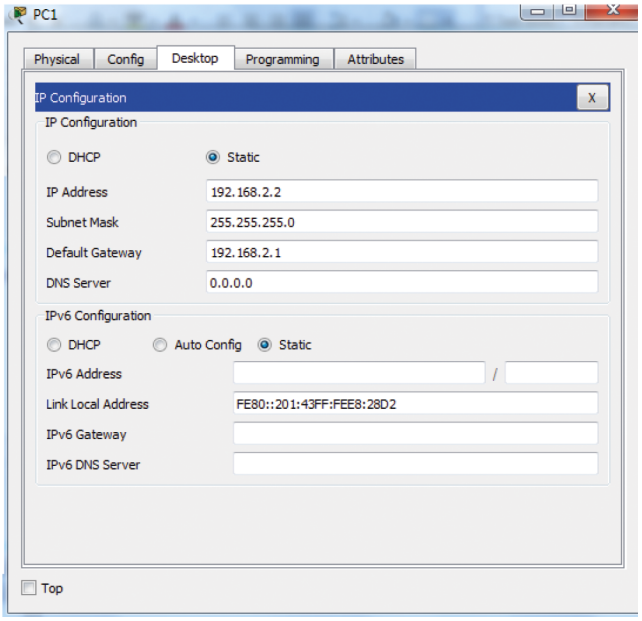


Figura 21c. Activación del protocolo OSPF en cada enrutador

```
Router0(config)#router ospf 10
Router0(config-router)#router-id
1.1.1.1
Router0(config-router)#passive-
interface Gig0/0
Router0(config-router)#passive-
interface Gig0/1
Router0(config-router)#network
192.168.0.0 0.0.0.255 area 0
Router0(config-router)#network
192.168.1.0 0.0.0.3 area 0
```

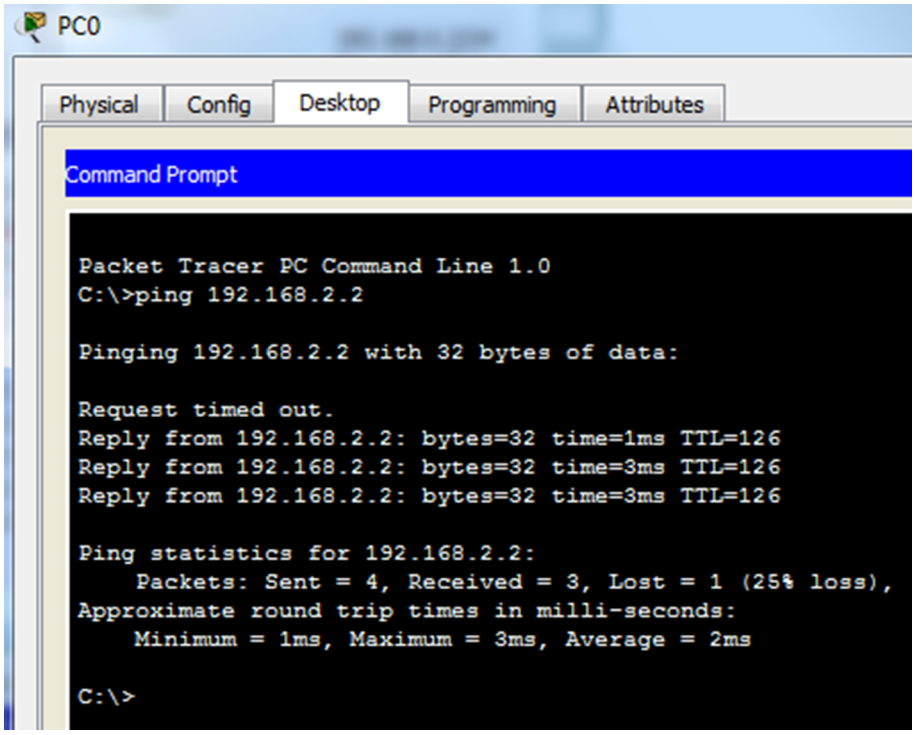
```
Router1(config)#router ospf 10
Router1(config-router)#router-id
2.2.2.2
Router1(config-router)#passive-
interface Gig0/0
Router1(config-router)#passive-
interface Gig0/1
Router1(config-router)#network
192.168.2.0 0.0.0.255 area 0
Router1(config-router)#network
192.168.1.0 0.0.0.3 area 0
00:33:11: %OSPF-5-ADJCHG: Process
10, Nbr 1.1.1.1 on Serial0/1/0
from LOADING to FULL, Loading Done
```

Figura 21d. Observación de la tabla de *routing* de Router0 y Router1

```
Router0#show ip route
Codes: L - local, C - connected,
S - static, R - RIP, M - mobile,
B - BGP
D - EIGRP, EX - EIGRP external, O
- OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 -
OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2
- IS-IS level-2, ia - IS-IS inter
area
* - candidate default, U - per-
user static route, o - ODR
P - periodic downloaded static
route
Gateway of last resort is not set
192.168.0.0/24 is variably
subnetted, 2 subnets, 2 masks
C 192.168.0.0/24 is directly
connected, GigabitEthernet0/0
L 192.168.0.1/32 is directly
connected, GigabitEthernet0/0
192.168.1.0/24 is variably
subnetted, 2 subnets, 2 masks
C 192.168.1.0/30 is directly
connected, Serial0/1/0
L 192.168.1.1/32 is directly
connected, Serial0/1/0
O 192.168.2.0/24 [110/65] via
192.168.1.2, 00:00:48,
Serial0/1/0
```

```
Router1#show ip route
Codes: L - local, C - connected,
S - static, R - RIP, M - mobile,
B - BGP
D - EIGRP, EX - EIGRP external, O
- OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 -
OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2
- IS-IS level-2, ia - IS-IS inter
area
* - candidate default, U - per-
user static route, o - ODR
P - periodic downloaded static
route
Gateway of last resort is not set
O 192.168.0.0/24 [110/65] via
192.168.1.1, 00:00:00,
Serial0/1/0
192.168.1.0/24 is variably
subnetted, 2 subnets, 2 masks
C 192.168.1.0/30 is directly
connected, Serial0/1/0
L 192.168.1.2/32 is directly
connected, Serial0/1/0
192.168.2.0/24 is variably
subnetted, 2 subnets, 2 masks
C 192.168.2.0/24 is directly
connected, GigabitEthernet0/0
L 192.168.2.1/32 is directly
connected, GigabitEthernet0/0
```

Figura 21e. Comprobación de la conectividad entre los ordenadores



Comandos de observación del estado de OSPF

A continuación se muestran una serie de comandos que nos pueden ayudar a averiguar el estado del protocolo OSPF una vez activado:

```
Router0#show ip ospf neighbor

Neighbor ID   Pri        State   Dead   Time   Address   Interface
2.2.2.2   0   FULL/ - 00:00:37   192.168.1.2   Serial0/1/0
```

```
Router0#show ip ospf database
OSPF Router with ID (1.1.1.1) (Process ID 10)
Router Link States (Area 0)
Link ID    ADV Router    Age    Seq#    Checksum Link count
1.1.1.1    1.1.1.1       366    0x80000005 0x002441 3
2.2.2.2    2.2.2.2       366    0x80000005 0x00b1ac 3
```

```
Router#show ip ospf
Routing Process "ospf 10" with ID 1.1.1.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x000000
Number of opaque AS LSA 0. Checksum Sum 0x000000
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
```

```
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
        Area BACKBONE(0)
                Number of interfaces in this area is 2
                Area has no authentication
                SPF algorithm executed 2 times
                Area ranges are
                Number of LSA 2. Checksum Sum 0x00d9eb
                Number of opaque link LSA 0. Checksum Sum 0x000000
                Number of DCbitless LSA 0
                Number of indication LSA 0
                Number of DoNotAge LSA 0
                Flood list length 0
```

```
Router0#show ip ospf interface
      GigabitEthernet0/0 is up, line protocol is up
      Internet address is 192.168.0.1/24, Area 0
      Process ID 10, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
      Transmit Delay is 1 sec, State DR, Priority 1
      Designated Router (ID) 1.1.1.1, Interface address 192.168.0.1
      No backup designated router on this network
      Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      No Hellos (Passive interface)
      Index 1/1, flood queue length 0
      Next 0x0(0)/0x0(0)
      Last flood scan length is 1, maximum is 1
      Last flood scan time is 0 msec, maximum is 0 msec
      Neighbor Count is 0, Adjacent neighbor count is 0
      Suppress hello for 0 neighbor(s)
Serial0/1/0 is up, line protocol is up
      Internet address is 192.168.1.1/30, Area 0
      Process ID 10, Router ID 1.1.1.1, Network Type POINT-TO-POINT, Cost: 64
      Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0
      No designated router on this network
      No backup designated router on this network
      Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:04
      Index 2/2, flood queue length 0
      Next 0x0(0)/0x0(0)
      Last flood scan length is 1, maximum is 1
      Last flood scan time is 0 msec, maximum is 0 msec
      Neighbor Count is 1 , Adjacent neighbor count is 1
      Adjacent with neighbor 2.2.2.2
      Suppress hello for 0 neighbor(s)
```

```
Router0#show ip protocols
```

```
Routing Protocol is "ospf 10"
   Outgoing update filter list for all interfaces is not set
   Incoming update filter list for all interfaces is not set
   Router ID 1.1.1.1
   Number of areas in this router is 1. 1 normal 0 stub 0 nssa
   Maximum path: 4
   Routing for Networks:
         192.168.0.0 0.0.0.255 area 0
         192.168.1.0 0.0.0.3 area 0
   Passive Interface(s):
         GigabitEthernet0/0
         GigabitEthernet0/1
   Routing Information Sources:
   Gateway    Distance    Last Update
   1.1.1.1    110    00:11:08
   2.2.2.2    110    00:11:08
   Distance: (default is 110)
```

```
Router#debug ip ospf events
OSPF events debugging is on
Router#
00:43:47: OSPF: Rcv hello from 2.2.2.2 area 0 from Serial0/1/0 192.168.1.2
00:43:47: OSPF: End of hello processing
00:43:57: OSPF: Rcv hello from 2.2.2.2 area 0 from Serial0/1/0 192.168.1.2
00:43:57: OSPF: End of hello processing
```

3.5.3. Encaminamiento con EIGRP

EIGRP (Enhanced Interior Gateway Routing Protocol) es un protocolo de encaminamiento vector distancia avanzado, propiedad de Cisco, que combina las ventajas de los algoritmos de vector distancia y del estado del enlace. EIGRP mejora las propiedades de convergencia y es más fácil de configurar que OSPF. Utiliza el cálculo de métrica siguiente:

Métrica = [K1 * ancho de banda + ((K2 * ancho de banda) / (256-carga)) + (K3 * retraso)] * [K5 / (confiabilidad + K4)]

Cuando K4 y K5 son 0, la ecuación de la métrica queda:

Métrica = ancho de banda + retraso

Para configurar el protocolo EIGRP, hay que añadir al final de la instrucción `router` un número de sistema autónomo:

```
Router(config): router eigrp <number sa>
```

El número de sistema autónomo se utiliza para identificar todos los enrutadores que pertenecen a la red. Solo intercambiarán datos los equipos configurados con el mismo valor.

Posteriormente solo hará falta introducir las direcciones IP de las redes que se quieran publicar con el comando `network`:

```
Router(config-router): network <ip xarxa> <wildcard>
```

Solo se pueden poner redes directamente conectadas al dispositivo vía alguna interfaz. También hay que especificar una *wildcard*, que equivale a la inversión de la máscara de la red.

Con el comando `no auto-summary` se indica al enrutador que no resuma las rutas que tiene:

```
Router(config-router): no auto-summary
```

Por defecto, el protocolo EIGRP resume automáticamente las subredes en la red principal de clase cada vez que atraviesan una frontera entre dos redes principales diferentes. El comando `no auto-summary` desactiva este comportamiento y hace publicidad de las subredes. Supongamos que un enrutador tiene dos subredes, 172.16.8.0/24 y 172.16.4.0/24, de clase B y una subred, 10.2.0.0/16, de clase A. Cuando `auto-summary` está habilitado, el enrutador anunciará solo la red principal mayor resumida de clase B 172.16.0.0/16 por su interfaz de clase A. Con el comando `no auto-summary` anunciaría las dos subredes.

Ejemplo de encaminamiento con EIGRP entre dos enrutadores

Dada la red de la figura 22a, se quiere activar el protocolo EIGRP en los enrutadores Router0 y Router1 para que conozcan todas las rutas de la red.

En la figura 22b se muestra la configuración de las interfaces de todos los dispositivos de la red, teniendo en cuenta la velocidad de la interfaz serial DCE (comando `clock rate`).

Para activar el protocolo EIGRP en los dos enrutadores (figura 22c) se utilizan los comandos:

- `network`, para indicar las redes directamente conectadas al enrutador y que tomarán parte del proceso EIGRP.
- `no auto-summary`, que controla que se anuncien las redes sin resúmenes.

En la figura 22c, después de configurar Router1, su sistema ya indica que ha contactado y establecido una relación de adyacencia con el enrutador vecino 192.168.30.1.

En la figura 22d se observan las tablas de *routing* de los dos enrutadores. Por ejemplo, Router1 ha aprendido la red 192.168.10.0/24 gracias a EIGRP, la cual tiene una distancia administrativa de 90 y una métrica de 2.172.416. La figura 22e muestra que la conectividad entre todos los ordenadores de la red es completa.

Figura 22a. Red formada por tres subredes y dos enrutadores

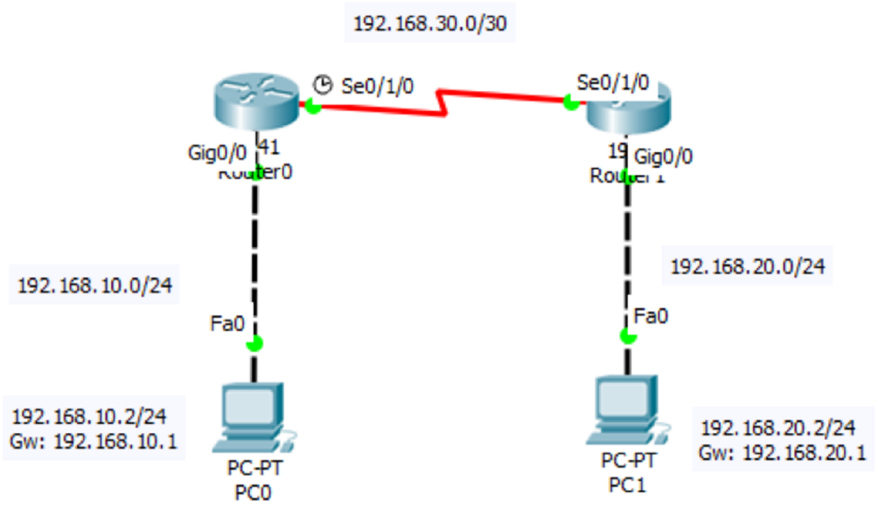
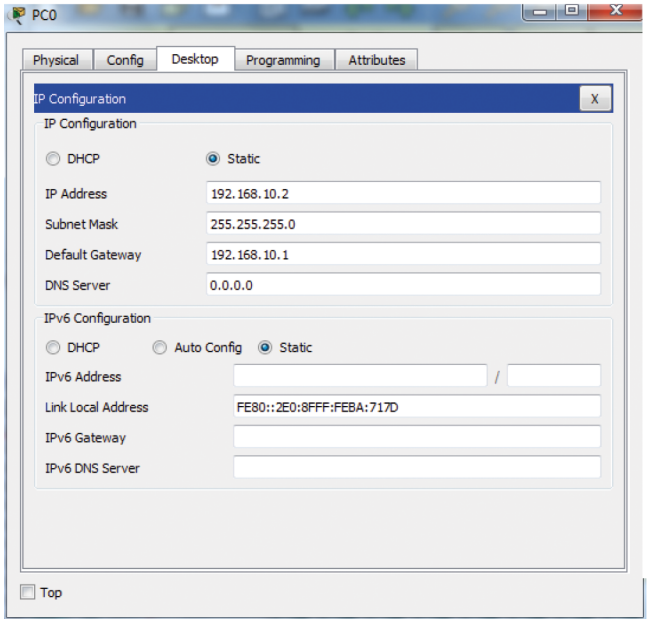


Figura 22b. Configuración de las interfaces de los enrutadores y los ordenadores

```
Router>enable
Router#configure terminal
Router(config)#hostname Router0
Router0(config)#interface Gig0/0
Router0(config-if)#ip address
192.168.10.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#interface se0/1/0
Router0(config-if)#clock rate 4000000
Router0(config-if)#no shutdown
Router0(config-if)#ip address
192.168.30.1 255.255.255.252
Router0(config-if)#exit
```

```
Router>enable
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#interface Gig0/0
Router1(config-if)#ip address
192.168.20.1 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#interface se0/1/0
Router1(config-if)#no shutdown
Router1(config-if)#ip address
192.168.30.2 255.255.255.252
Router1(config-if)#exit
```



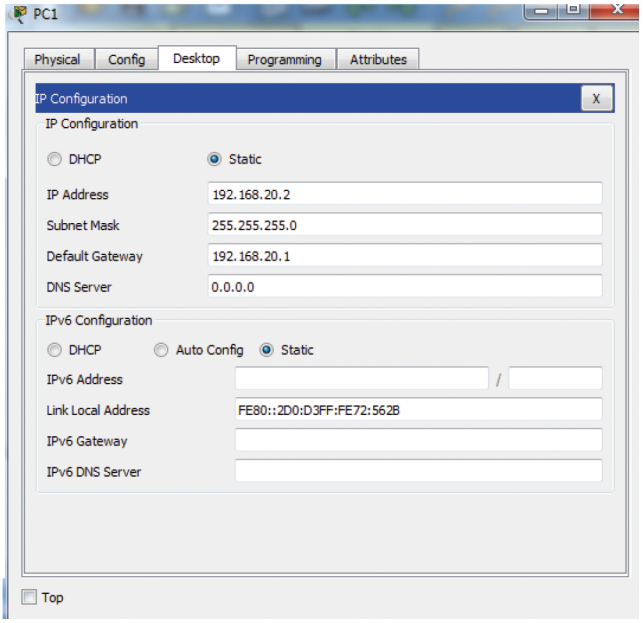


Figura 22c. Activación del protocolo EIGRP en cada enrutador

```
Router0(config)#router eigrp 10
Router0(config-router)#network
192.168.10.0 0.0.0.255
Router0(config-router)#network
192.168.30.0 0.0.0.3
Router0(config-router)#no auto-summary
Router0(config-router)#exit
Router0(config)#
```

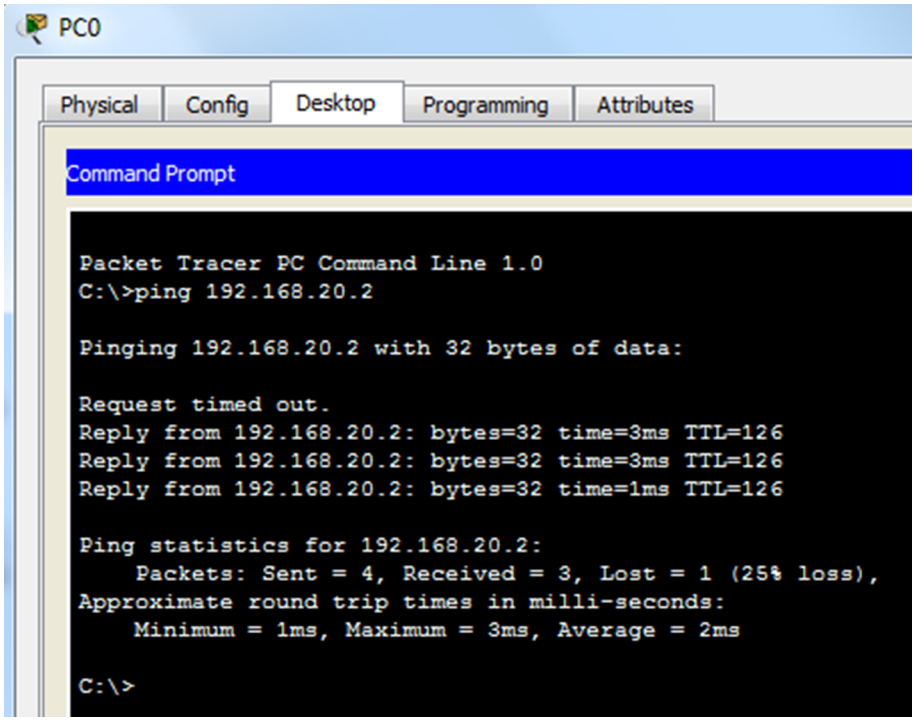
```
Router1(config)#router eigrp 10
Router1(config-router)#network
192.168.20.0 0.0.0.255
Router1(config-router)#network
192.168.30.0 0.0.0.3
Router1(config-router)#no auto-summary
Router1(config-router)#exit
Router1(config)#
%DUAL-5-NBRCHANGE: IP-EIGRP 10:
Neighbor 192.168.30.1
(Serial0/1/0) is up: new adjacency
```

Figura 22d. Observación de la tabla de *routing* de Router0 y Router1

```
Router0#show ip route
Codes: L - local, C - connected,
S - static, R - RIP, M - mobile,
B - BGP
D - EIGRP, EX - EIGRP external, O
- OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 -
OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2
- IS-IS level-2, ia - IS-IS inter
area
* - candidate default, U - per-
user static route, o - ODR
P - periodic downloaded static
route
Gateway of last resort is not set
192.168.10.0/24 is variably
subnetted, 2 subnets, 2 masks
C 192.168.10.0/24 is directly
connected, GigabitEthernet0/0
L 192.168.10.1/32 is directly
connected, GigabitEthernet0/0
D 192.168.20.0/24 [90/2172416]
via 192.168.30.2, 00:00:07,
Serial0/1/0
192.168.30.0/24 is variably
subnetted, 2 subnets, 2 masks
C 192.168.30.0/30 is directly
connected, Serial0/1/0
L 192.168.30.1/32 is directly
connected, Serial0/1/0
```

```
Router#show ip route
Codes: L - local, C - connected,
S - static, R - RIP, M - mobile,
B - BGP
D - EIGRP, EX - EIGRP external, O
- OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1,
N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 -
OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2
- IS-IS level-2, ia - IS-IS inter
area
* - candidate default, U -
per-user static route, o - ODR
P - periodic downloaded static
route
Gateway of last resort is not set
D 192.168.10.0/24 [90/2172416]
via 192.168.30.1, 00:00:53,
Serial0/1/0
192.168.20.0/24 is variably
subnetted, 2 subnets, 2 masks
C 192.168.20.0/24 is directly
connected, GigabitEthernet0/0
L 192.168.20.1/32 is directly
connected, GigabitEthernet0/0
192.168.30.0/24 is variably
subnetted, 2 subnets, 2 masks
C 192.168.30.0/30 is directly
connected, Serial0/1/0
L 192.168.30.2/32 is directly
connected, Serial0/1/0
```

Figura 22e. Comprobación de la conectividad entre los ordenadores



Comandos de observación del estado de EIGRP

A continuación, se muestran una serie de comandos que nos pueden ayudar a averiguar el estado del protocolo EIGRP una vez activado:

```
Router#show ip protocols
Routing Protocol is "eigrp 10 "
   Outgoing update filter list for all interfaces is not set
   Incoming update filter list for all interfaces is not set
   Default networks flagged in outgoing updates
   Default networks accepted from incoming updates
   Redistributing: eigrp 10
   EIGRP-IPv4 Protocol for AS(10)
      Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
      NSF-aware route hold timer is 240
      Router-ID: 192.168.10.1
      Topology : 0 (base)
         Active Timer: 3 min
         Distance: internal 90 external 170
         Maximum path: 4
         Maximum hopcount 100
         Maximum metric variance 1
   Automatic Summarization: disabled
   Automatic address summarization:
   Maximum path: 4
   Routing for Networks:
      192.168.10.0
```

```
      192.168.30.0/30
   Routing Information Sources:
      Gateway Distance Last Update
      192.168.30.2 90 5195
   Distance: internal 90 external 170
```

```
Router1#show ip eigrp interfaces
IP-EIGRP interfaces for process 10

              Xmit Queue Mean Pacing Time Multicast Pending
Interface    Peers    Un/Reliable   SRTT   Un/Reliable   Flow Timer   Routes
Gig0/0         0         0/0         1236     0/10         0            0
Se0/1/0        1         0/0         1236     0/10         0            0
Router#
```

```
Router1#show ip eigrp neighbors
IP-EIGRP neighbors for process 10
H Address         Interface     Hold Uptime     SRTT     RTO     Q     Seq
                                (sec)           (ms)           Cnt    Num
0 192.168.30.2      Se0/1/0    12 00:52:10       40      1000    0      3
Router#
```

3.6. Redes IPv6

IPv6 es la versión 6 del Protocolo de Internet (IP), diseñada para sustituir el actual IPv4 en internet, ya que el espacio de direccionamiento IPv4 está a punto de agotarse y la IANA está repartiendo los últimos grupos de direcciones.

El tamaño de las direcciones IPv6 es de 128 bits, cuatro veces mayor que el tamaño de las direcciones IPv4 de 32 bits.

Las direcciones IPv6 se representan en ocho grupos de cuatro dígitos hexadecimales. Por ejemplo, 2001:0DB8:85A3:08D3:1119:8A2E:0220:AB34 es una dirección IPv6 válida.

Si un grupo de cuatro dígitos toma el valor nulo "0000", entonces puede ser comprimido con la notación "::" o ":0:". Por ejemplo, 2001:1111:1111:0000:2222:2222:2222:2222 es la misma dirección que 2001:1111:1111::2222:2222:2222:2222 o la dirección 2001:1111:1111:0:2222 :2222:2222:2222.

Si una dirección IPv6 tiene más de una serie de grupos nulos consecutivos, también se puede comprimir con "::", pero siempre respetando que no haya ambigüedades. Por ejemplo, la dirección 2001::2222::3333 no es válida porque no queda claro cuántos grupos nulos hay en cada lado.

Otra regla es que los ceros iniciales en un grupo pueden ser omitidos. Por ejemplo, la dirección IPv6 2001:0111:0222::0333 también se puede representar como 2001:111:222::333.

IPv6 utiliza la notación de la máscara compacta como ya se utilizaba en IPv4. Por ejemplo, la dirección 2001:20:20:20::1/64 indica que 64 bits es la parte de subred y los otros 64 bits representan la parte de *host*.

3.6.1. Tipo de direcciones IPv6

Los tipos de direcciones IPv6 pueden identificarse teniendo en cuenta los primeros bits de cada una:

1) Direcciones *Global Unicast* en IPv6

Las direcciones *Global Unicast* en IPv6 son el equivalente a las direcciones IP públicas IPv4. Son direcciones únicas en el mundo y se configuran una por cada interfaz de *host*. Las direcciones IPv6 *Global Unicast* son direcciones IPv6 enrutables de internet.

Los tres primeros bits de estas direcciones deben ser los valores 001 en notación binaria, por lo tanto el prefijo de estas direcciones IP siempre tendrá un valor hexadecimal de 2xxx o 3XXX con una máscara /3, que establece el Global Routing Prefix.

Una dirección *Global Unicast* IPv6 de 128 bits se puede dividir en las partes siguientes:

- Prefijo Global (3 primeros bits).
- Red asignada a las organizaciones (45 bits siguientes).
- Subred (16 bits siguientes, para 2^{16} posibles subredes).
- *Host* (64 bits, donde cada subred puede soportar 2^{64} nodos).

2) Direcciones *Link Local*

Estas direcciones IPv6 locales permiten la comunicación entre dispositivos en un enlace local. El prefijo de la dirección IPv6 local es fe80::/10, y especifica que esta dirección solo es válida en este enlace físico local.

- Las direcciones IPv6 *Link Local* no son enrutables.
- Normalmente se asignan automáticamente.
- Solo se comunica con otras interfaces en el enlace.
- Pueden duplicarse fuera de los enlaces locales, ya que no interferirán entre sí.

Se puede generar automáticamente a partir de la dirección MAC de la misma interfaz con el comando:

```
Router(config-if)# ipv6 enable
```

También se pueden indicar manualmente con el comando:

```
Router(config-if)#ipv6 address FE80::1 Link Local
```

3) Direcciones *multicast*

Una transmisión *multicast* envía paquetes a todas las interfaces que son parte del grupo *multicast*. El grupo viene representado por la dirección IPv6 destino del paquete. Las direcciones IPv6 *multicast* comienzan con FF (FF00::/8). Las direcciones IPv6 *multicast* más importantes son:

FF02::1 - Todos los nodos en el segmento local de la red.

FF02::2 - Todos los enrutadores en el segmento local de la red.

4) Direcciones *loopback*

La dirección *loopback* IPv6 es la

0000:0000:0000:0000:0000:0000:0000:0001/128,

que también puede representarse como ::1/128. Se utiliza por un nodo para enviar un paquete IPv6 a sí mismo. Una dirección *loopback* IPv6 funciona igual que una dirección *loopback* IPv4 (corresponde con la dirección 127.0.0.1 de IPv4). No es asignable a ninguna interfaz física.

5) Direcciones *anycast*

Una transmisión *anycast* envía paquetes a solo una de las interfaces asociadas con la dirección, no a todas las interfaces. Esta interfaz es normalmente la interfaz más cercana, tal como define el protocolo de *routing*.

3.6.2. Ejemplo: generación de direcciones *Link Local*

En la figura 23a se muestra un enlace Ethernet entre un enrutador y un ordenador. Se pretende hacer un *ping* entre ambos dispositivos utilizando solo direcciones *Link Local* de IPv6.

En el ordenador PC0 ya hay una dirección *Link Local* configurada por defecto (figura 23c). Por lo tanto, basta con generar una dirección *Link Local* para la interfaz GigabitEthernet0/0 de Router0. Esto se hace con el comando `ipv6 enable`, como se puede observar en la figura 23c.

En la figura 23b se muestra la dirección MAC de la interfaz GigabitEthernet0/0 del enrutador Router0: **0001:4235:D201**. Se puede comprobar observando el resultado del comando `show ipv6 interface brief` (figura 23d) que la dirección *Link Local* IPv6 generada en la interfaz Gig0/0 tiene una parte de su dirección MAC: FE80::**201:42**FF:FE**35:D201**.

Desde PC0 se puede ejecutar un *ping* con éxito hacia la dirección *Link Local* de la interfaz GigabitEthernet0/0 generada anteriormente (figura 23e).

Figura 23a. Enlace Ethernet entre un enrutador y un ordenador

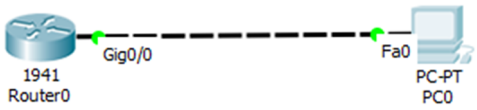


Figura 23b. Direcciones MAC de las interfaces del enrutador

```
Port                Link   VLAN   IP Address       IPv6 Address                MAC Address
GigabitEthernet0/0  Up     --     <not set>        <not set>                   0001.4235.D201
GigabitEthernet0/1  Down   --     <not set>        <not set>                   0001.4235.D202
Vlan1               Down   1      <not set>        <not set>                   0010.111A.0678
Hostname: Router
```

Figura 23c. Configuración de las direcciones *Link Local* en el enrutador y en el ordenador

```
Router>enable
Router#configure terminal
Router(config)#interface Gig0/0
Router(config-if)#ipv6 enable
Router(config-if)#no shutdown
Router(config-if)#exit
```

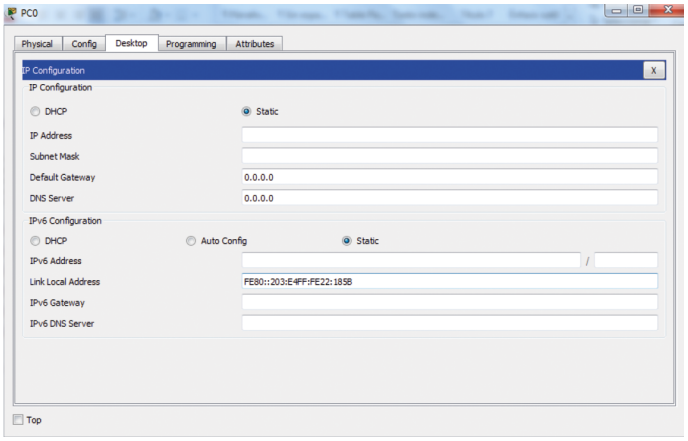
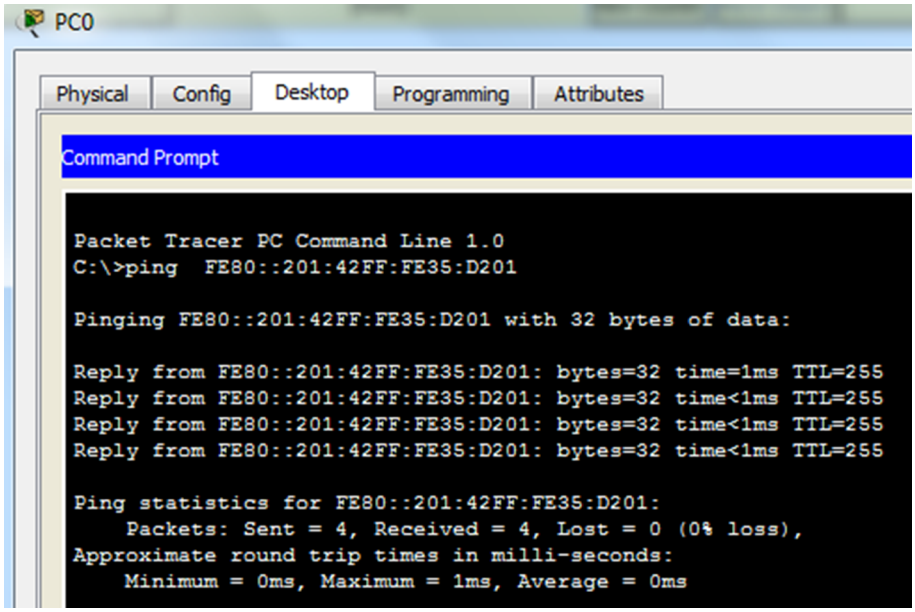


Figura 23d. Resultado del comando `show ipv6 interface brief` en el enrutador

```
Router#show ipv6 interface brief
GigabitEthernet0/0 [up/up]
FE80::201:42FF:FE35:D201
GigabitEthernet0/1 [administratively down/down]
Vlan1 [administratively down/down]
```

Figura 23e. *Ping* realizado con direcciones *Link Local* entre enrutador y ordenador



3.6.3. Ejemplo de configuración de interfaces Ethernet con direcciones *Global Unicast*

En la figura 24a se muestra un enlace Ethernet entre las interfaces de un enrutador y de un ordenador. Se quiere conseguir conectividad IP entre ambos equipos utilizando direcciones *Global Unicast* de IPv6, como por ejemplo las de la red 2001:10:10:10::/64. En la figura 24b se muestra la configuración IPv6 de las interfaces del enrutador y del ordenador. El enrutador utiliza el comando `ipv6 address` para configurar su interfaz con una dirección IPv6. Una vez realizada la configuración, se puede hacer un *ping* con éxito entre los dos dispositivos (figura 24c).

Figura 24a. Enlace Ethernet entre un enrutador y un ordenador con direccionamiento IPv6

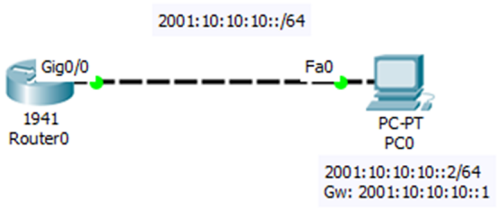


Figura 24b. Configuración de las interfaces de los enrutadores y del ordenador

```
Router>enable
Router#configure terminal
Router(config)#interface Gig0/0
Router(config-if)#ipv6 address
2001:10:10:10::1/64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

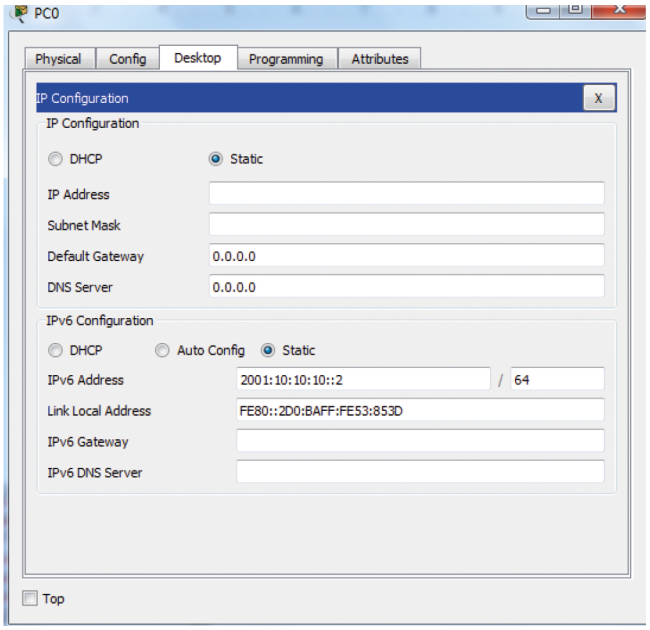
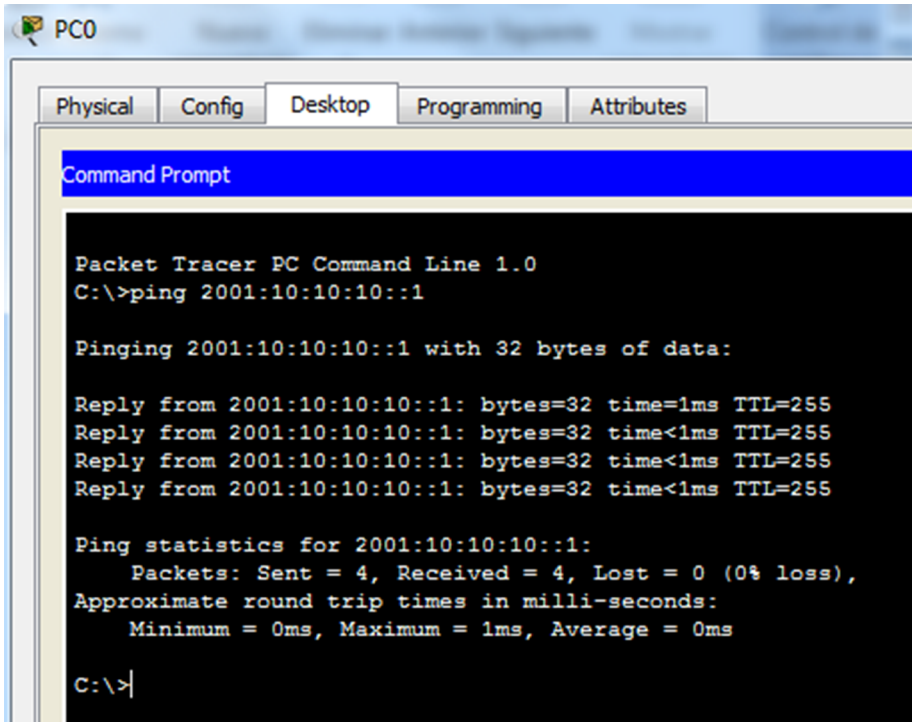


Figura 24c. Comprobación de la conectividad entre PC0 y Router0



Comando IPv6 unicast-routing

`Ipv6 unicast-routing` es el comando necesario para habilitar el reenvío de datagramas de unidifusión IPv6.

Para deshabilitar el reenvío de datagramas de unidifusión IPv6, se debe utilizar la negación de este comando.

Comando IPv6 route

Es el equivalente del comando `ip route` en IPv4. Sirve tanto para indicar rutas estáticas al enrutador, como para dar a conocer rutas no directamente conectadas a la tabla de *routing*.

3.6.4. Ejemplo de direccionamiento estático a IPv6

Dada la red de la figura 25a, se quiere conseguir visibilidad IP entre los dos ordenadores PC0 y PC1, pertenecientes a subredes IPv6 diferentes. Cada enrutador está directamente conectado a dos subredes, compartiendo la subred 2001:20:20:20::/64. En la figura 25b se muestran las configuraciones de las interfaces de los enrutadores Router0 y Router1 (con la instrucción `ipv6 address`) y de los ordenadores PC0 y PC1.

Figura 25a. Red formada por dos enrutadores y tres subredes IPv6

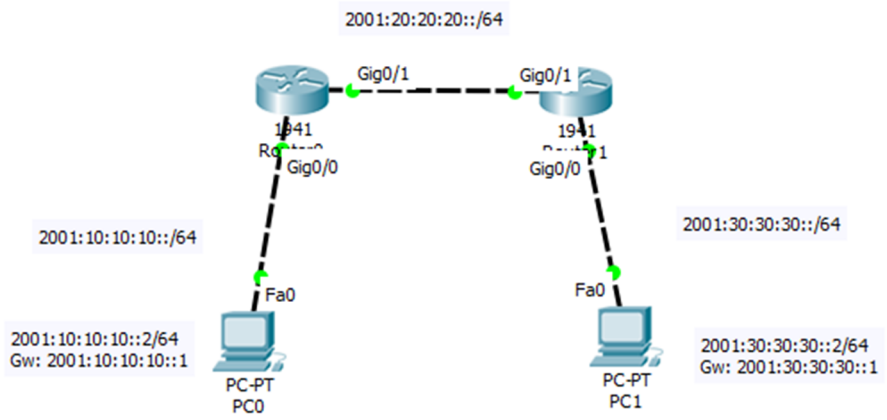
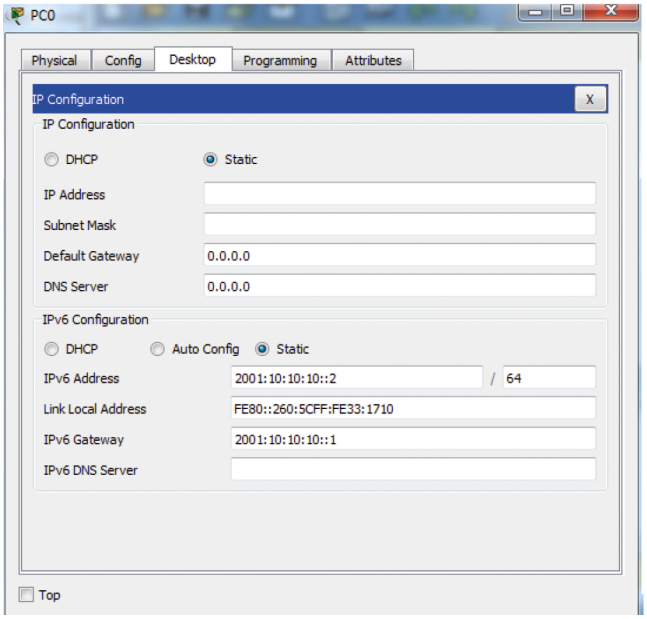
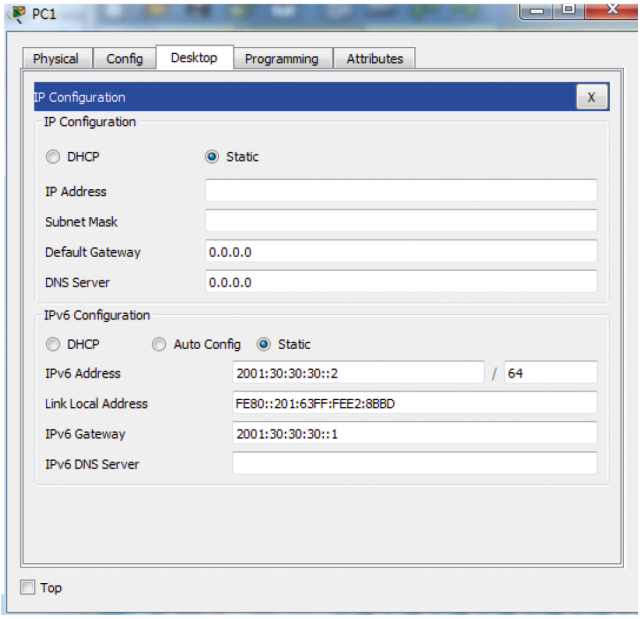


Figura 25b. Configuración de las interfaces de los enrutadores y los ordenadores

```
Router0(config)#ipv6 unicast-
routing
Router0(config)#interface
GigabitEthernet0/0
Router0(config-if)#no shutdown
Router0(config-if)#ipv6 address
2001:10:10:10::1/64
Router0(config-if)#interface
GigabitEthernet0/1
Router0(config-if)#ipv6 address
2001:20:20:20::1/64
Router0(config-if)#no shutdown
Router0(config-if)#exit
```

```
Router1(config)#ipv6 unicast-
routing
Router1(config)#interface
GigabitEthernet0/0
Router1(config-if)#no shutdown
Router1(config-if)#ipv6 address
2001:30:30:30::1/64
Router1(config-if)#interface
GigabitEthernet0/1
Router1(config-if)#ipv6 address
2001:20:20:20::2/64
Router1(config-if)#no shutdown
Router1(config-if)#exit
```





La situación es la misma que con IPv4: Router0 conoce las redes 10 y 20 porque están directamente conectadas, pero desconoce la red 30, e igualmente Router1 conoce las redes 20 y 30, pero desconoce la red 10. Por lo tanto, para que los enrutadores de la red conozcan todas las rutas, hay que ejecutar los comandos `ipv6 route` siguientes en los enrutadores correspondientes:

```
Router0(config)#ipv6 route
2001:30:30:30::/64
2001:20:20:20::2
```

```
Router1(config)#ipv6 route
2001:10:10:10::/64
2001:20:20:20::1
```

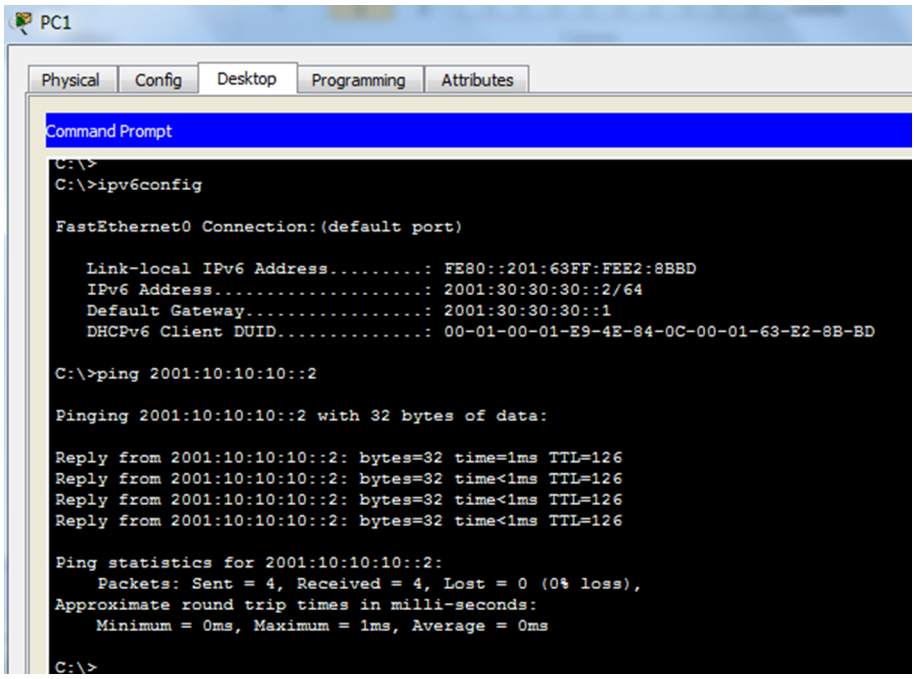
En la figura 25c se muestran las tablas de *routing* de Router0 y Router1. Se ha añadido al Router1 la red 10 con el distintivo S (estático), que tiene una distancia administrativa de 1 (mucha fiabilidad). En el Router0 ha añadido la red 30 de forma estática. En estas condiciones, un *ping* entre PC1 y PC0 funciona correctamente (figura 25d):

Figura 25c. Tablas de *routing* del Router0 y Router1

```
Router0#show ipv6 route
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local,
S - Static, R - RIP, B - BGP
U - Per-user Static route, M -
MIPv6
I1 - ISIS L1, I2 - ISIS L2, IA -
ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter,
OE1 - OSPF ext 1, OE2 - OSPF ext
2
ON1 - OSPF NSSA ext 1, ON2 - OSPF
NSSA ext 2
D - EIGRP, EX - EIGRP external
C 2001:10:10:10::/64 [0/0]
 via GigabitEthernet0/0, directly
connected
L 2001:10:10:10::1/128 [0/0]
 via GigabitEthernet0/0, receive
C 2001:20:20:20::/64 [0/0]
 via GigabitEthernet0/1, directly
connected
L 2001:20:20:20::1/128 [0/0]
 via GigabitEthernet0/1, receive
S 2001:30:30:30::/64 [1/0]
 via 2001:20:20:20::2
L FF00::/8 [0/0]
 via Null0, receive
Router0#
```

```
Router1#show ipv6 route
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local,
S - Static, R - RIP, B - BGP
U - Per-user Static route, M -
MIPv6
I1 - ISIS L1, I2 - ISIS L2, IA -
ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter,
OE1 - OSPF ext 1, OE2 - OSPF ext
2
ON1 - OSPF NSSA ext 1, ON2 - OSPF
NSSA ext 2
D - EIGRP, EX - EIGRP external
S 2001:10:10:10::/64 [1/0]
 via 2001:20:20:20::1
C 2001:20:20:20::/64 [0/0]
 via GigabitEthernet0/1, directly
connected
L 2001:20:20:20::2/128 [0/0]
 via GigabitEthernet0/1, receive
C 2001:30:30:30::/64 [0/0]
 via GigabitEthernet0/0, directly
connected
L 2001:30:30:30::1/128 [0/0]
 via GigabitEthernet0/0, receive
L FF00::/8 [0/0]
 via Null0, receive
Router0#
```

Figura 25d. *Ping* con éxito realizado entre PC1 y PC0



3.6.5. Ejemplo de encaminamiento entre diversas VLAN con IPv6

La red de la figura 26a está formada por un enrutador y un conmutador con soporte para VLAN. Se quiere segmentar la red en tres VLAN (en este caso tres subredes IPv6), utilizando el módulo de nivel 3 del que dispone el conmutador. En el primer paso se deben configurar los ordenadores (figura 26b). A continuación, se deben crear las VLAN y asignarlas a cada puerto correspondiente del conmutador (figura 26c). Se debe configurar también el puerto *trunking* del conmutador. Finalmente, se deben configurar las subinterfaces del enrutador (figura 26c). Se deben definir y configurar tantas subinterfaces lógicas sobre la misma interfaz física Gig0/1, como el número de VLAN definidas en la red.

Una vez realizadas todas las configuraciones, en la figura 26d se pueden observar las entradas, en la tabla de *routing* del enrutador, correspondientes a las diferentes subredes IPv6 definidas sobre la misma interfaz física.

La conectividad entre ordenadores es total, como se puede observar en la figura 26e.

Figura 26a. Red segmentada en tres VLAN

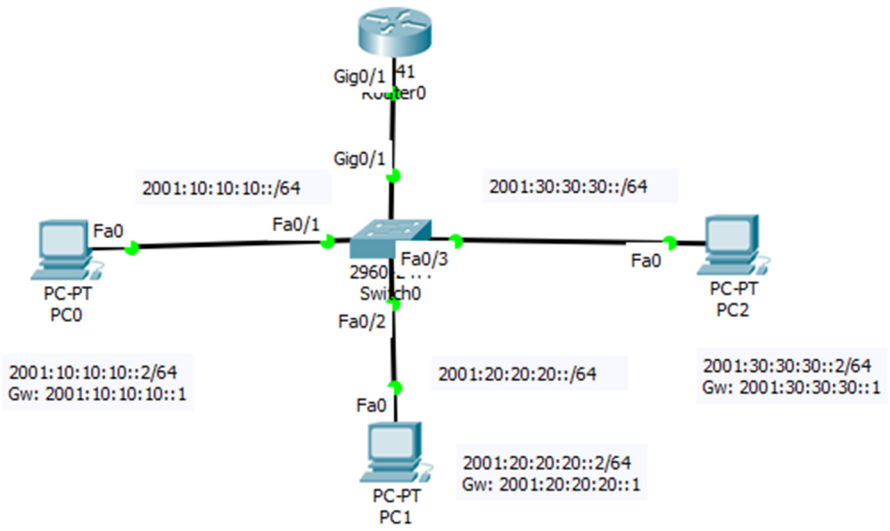


Figura 26b. Configuración de los ordenadores

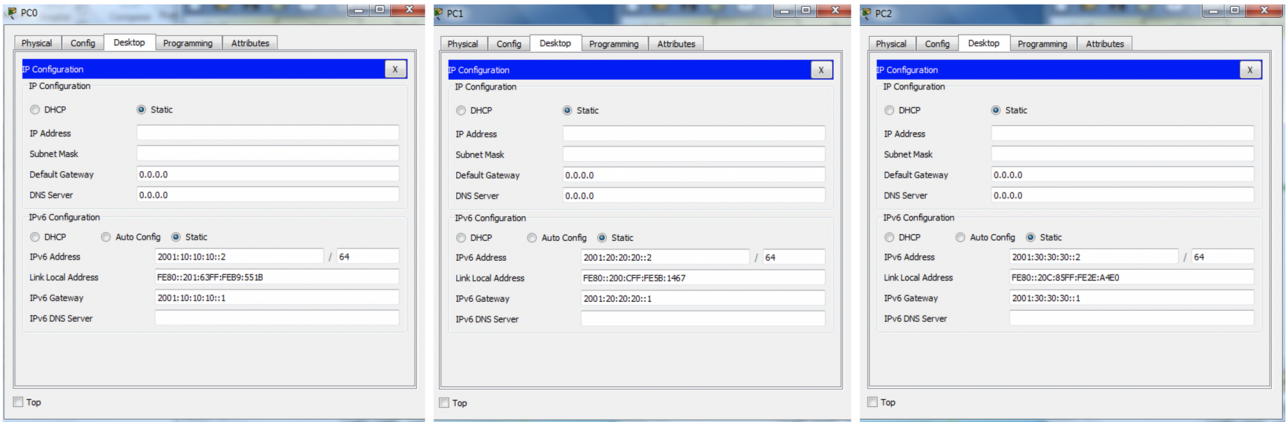


Figura 26c. Configuración del conmutador y las subinterfaces del enrutador

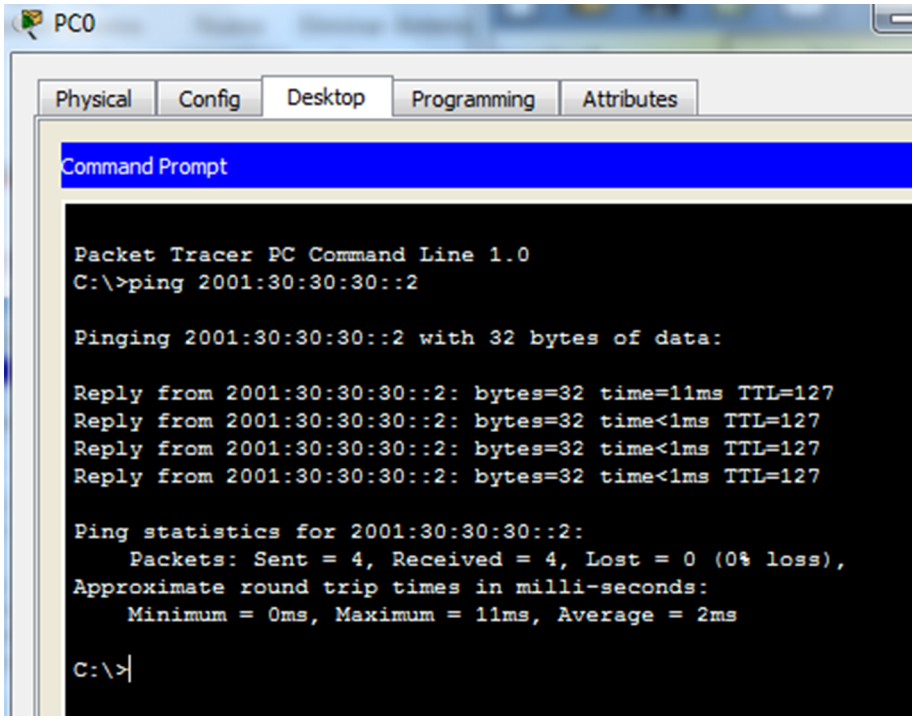
```
Switch>enable
Switch#configure terminal
Switch (config)#vlan 10
Switch (config-vlan)#exit
Switch (config)#vlan 20
Switch (config-vlan)#exit
Switch (config)#vlan 30
Switch (config-vlan)#exit
Switch (config)#interface fa0/1
Switch (config-if)#switchport
mode access
Switch (config-if)#switchport
access vlan 10
Switch (config-if)#exit
Switch (config)#interface fa0/2
Switch (config-if)#switchport
mode access
Switch (config-if)#switchport
access vlan 20
Switch (config-if)#exit
Switch (config)#interface fa0/3
Switch (config-if)#switchport
mode access
Switch (config-if)#switchport
access vlan 30
Switch (config-if)#exit
Switch (config)#interface Gig0/1
Switch (config-if)#switchport
mode trunk
Switch (config-if)#exit
```

```
Router>enable
Router#configure terminal
Router(config)#ipv6 unicast-
routing
Router(config)#interface gig0/1
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface
gig1/0.10
Router(config-
subif)#encapsulation dot1Q 10
Router(config-subif)#ipv6 address
2001:10:10:10::1/64
Router(config-subif)#exit
Router(config)#interface
gig0/1.20
Router(config-
subif)#encapsulation dot1Q 20
Router(config-subif)#ipv6 address
2001:20:20:20::1/64
Router(config-subif)#exit
Router(config)#interface
gig0/1.30
Router(config-
subif)#encapsulation dot1Q 30
Router(config-subif)#ipv6 address
2001:30:30:30::1/64
Router(config-subif)#exit
Router(config)#exit
```

Figura 26d. Tabla de *routing* de Router0

```
Router#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
U - Per-user Static route, M - MIPv6
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
D - EIGRP, EX - EIGRP external
C 2001:10:10:10::/64 [0/0]
 via GigabitEthernet0/1.10, directly connected
L 2001:10:10:10::1/128 [0/0]
 via GigabitEthernet0/1.10, receive
C 2001:20:20:20::/64 [0/0]
 via GigabitEthernet0/1.20, directly connected
L 2001:20:20:20::1/128 [0/0]
 via GigabitEthernet0/1.20, receive
C 2001:30:30:30::/64 [0/0]
 via GigabitEthernet0/1.30, directly connected
L 2001:30:30:30::1/128 [0/0]
 via GigabitEthernet0/1.30, receive
L FF00::/8 [0/0]
 via Null0, receive
Router#
```

Figura 26e. Comprobación de la conectividad entre PC0 y PC2



3.7. Encaminamiento con OSPFv3

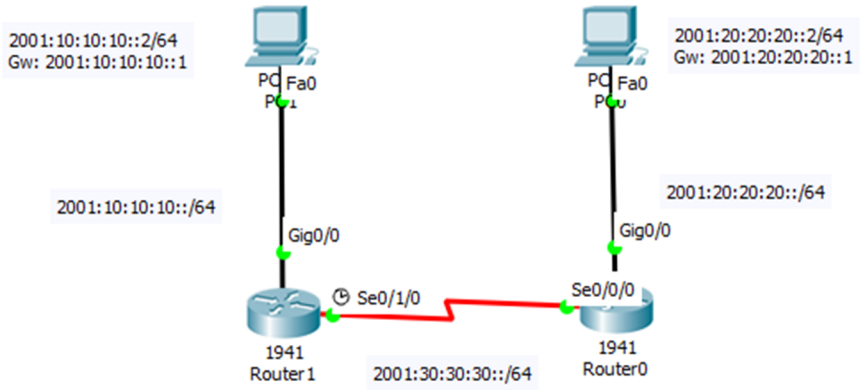
OSPFv3 es la versión del protocolo de encaminamiento *link-state* OSPF que permite trabajar con direcciones IPv6 (OSPFv2 trabaja con direcciones IPv4). Hay una diferencia básica muy importante sobre la forma de configurar y activar el protocolo OSPF entre las dos versiones:

- OSPFv2 se activa en el modo de configuración (`Router(config)#`)
- OSPFv3 se activa en el modo de configuración específica de cada interfaz implicada en el proceso OSPF (`Router(config-if)#`)

3.7.1. Ejemplo: encaminamiento con OSPFv3

Dada la red de la figura 27a, formada por solo dos enrutadores y tres subredes IPv6, el objetivo final es activar el protocolo OSPF versión 3, para que todos los enrutadores conozcan todas las subredes de la red principal y se pueda realizar un *ping* con éxito entre los dos ordenadores.

Figura 27a. Red formada por dos enrutadores con direccionamiento IPv6



En primer lugar, se deben configurar las interfaces Serial y Ethernet de los enrutadores Router0 y Router1 y las interfaces Ethernet de los ordenadores PC0 y PC1 (figura 27b). Como se trata de direcciones IPv6, se utilizará el comando `ipv6 address` para configurar todas las interfaces de los enrutadores. Además, en la interfaz serial DCE de Router1 se debe configurar la velocidad del enlace con el comando `clock rate`.

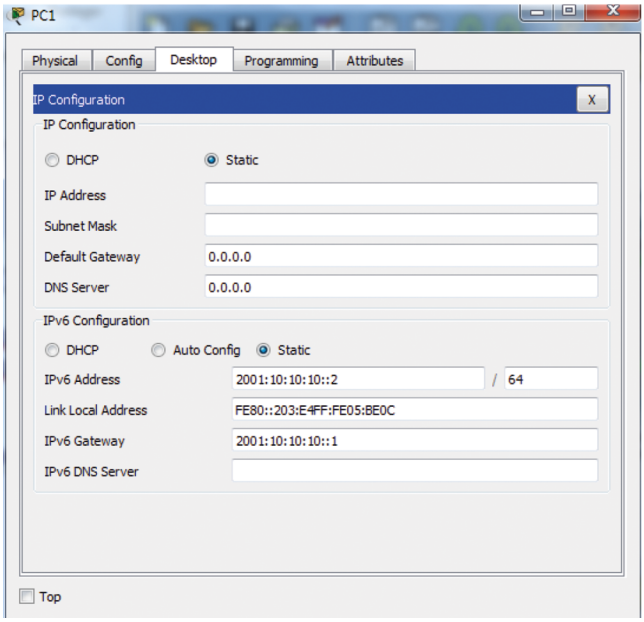
Como se detalla en la figura 27c, la activación del protocolo OSPFv3 se hace en cada interfaz del enrutador implicada en la transmisión de los paquetes *Hello* y *LSA*. Se debe utilizar el comando `ipv6 ospf` con dos identificadores: el del proceso OSPF y el del área.

Una vez activado este protocolo, se puede comprobar en la figura 27d que las redes aprendidas con OSPF tienen el distintivo 'O'. En la figura 27e se muestra un *ping* ejecutado con éxito entre los ordenadores PC0 y PC1.

Figura 27b. Configuración de las interfaces de los enrutadores y los ordenadores

```
Router>enable
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#interface Gig0/0
Router1(config-if)#ipv6 address
2001:10:10:10::1/64
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#interface se0/1/0
Router1(config-if)#clock rate 4000000
Router1(config-if)#no shutdown
Router1(config-if)#ipv6 address
2001:30:30:30::1/64
Router1(config-if)#exit
Router1(config)#
```

```
Router>enable
Router#configure terminal
Router(config)#hostname Router0
Router0(config)#interface Gig0/0
Router0(config-if)#ipv6 address
2001:20:20:20::1/64
Router0(config-if)#no shutdown
Router0(config-if)#exit
Router0(config)#interface se0/0/0
Router0(config-if)#no shutdown
Router0(config-if)#ipv6 address
2001:30:30:30::2/64
Router0(config-if)#exit
Router0(config)#
```



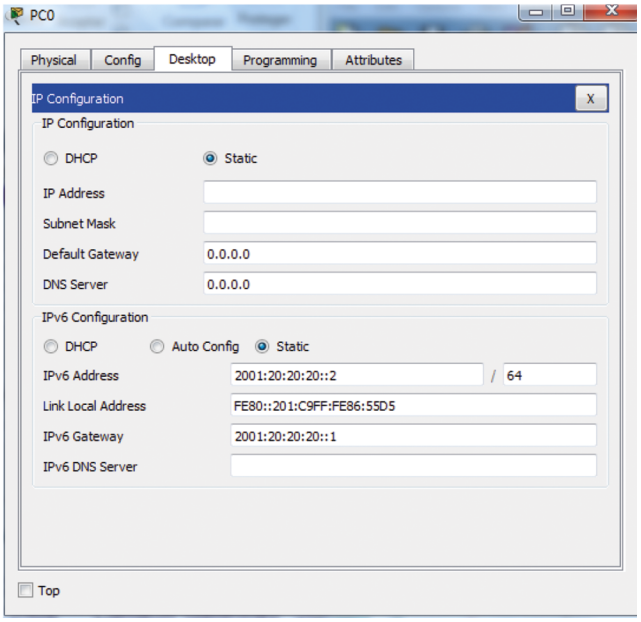


Figura 27c. Activación del protocolo OSPFv3 en cada enrutador

```
Router1>enable
Router1#conf term
Router1(config)#ipv6 unicast-
routing
Router1(config)#ipv6 router ospf
100
Router1(config-rtr)#router-id
1.1.1.1
Router1(config-rtr)#exit
Router1(config)#int Gig0/0
Router1(config-if)#ipv6 ospf 100
area 0
Router1(config-if)#exit
Router1(config)#int se0/1/0
Router1(config-if)#ipv6 ospf 100
area 0
Router1(config-if)#exit
Router1(config)#
```

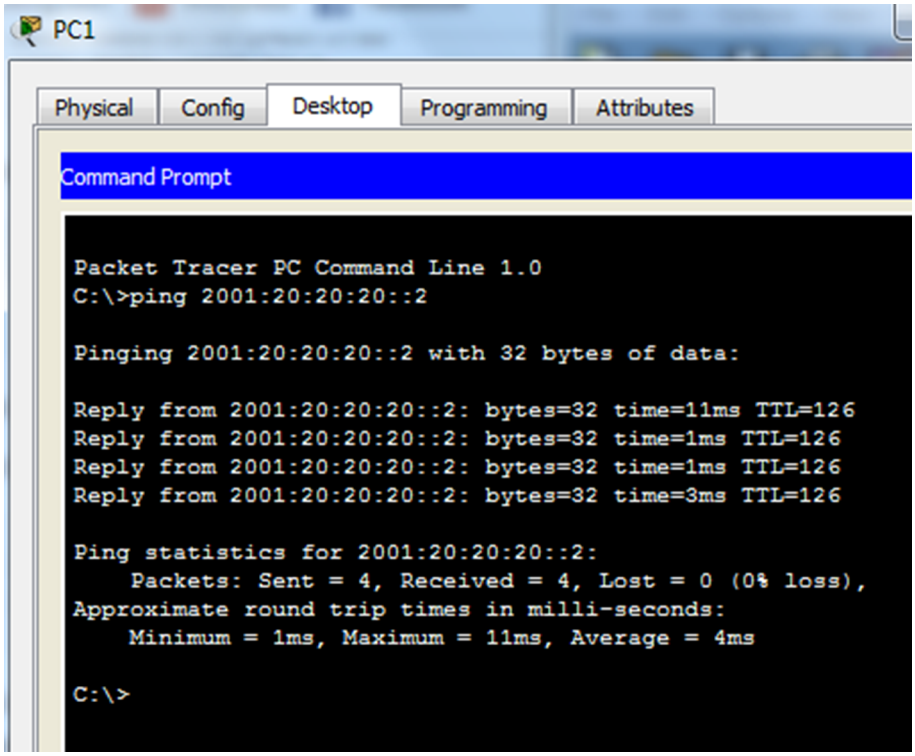
```
Router0>enable
Router0#conf term
Router0(config)#ipv6 unicast-
routing
Router0(config)#ipv6 router ospf
100
Router0(config-rtr)#router-id
2.2.2.2
Router0(config-rtr)#exit
Router0(config)#int Gig0/0
Router0(config-if)#ipv6 ospf 100
area 0
Router0(config-if)#exit
Router0(config)#int se0/0/0
Router0(config-if)#ipv6 ospf 100
area 0
Router0(config-if)#exit
Router0(config)#
```

Figura 27d. Tabla de *routing* de Router0 y Router1

```
Router1#show ipv6 route
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local,
S - Static, R - RIP, B - BGP
U - Per-user Static route, M -
MIPv6
I1 - ISIS L1, I2 - ISIS L2, IA -
ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter,
OE1 - OSPF ext 1, OE2 - OSPF ext
2
ON1 - OSPF NSSA ext 1, ON2 - OSPF
NSSA ext 2
D - EIGRP, EX - EIGRP external
C 2001:10:10:10::/64 [0/0]
 via GigabitEthernet0/0, directly
connected
L 2001:10:10:10::1/128 [0/0]
 via GigabitEthernet0/0, receive
O 2001:20:20:20::/64 [110/65]
 via FE80::20A:41FF:FEDD:E601,
Serial0/1/0
C 2001:30:30:30::/64 [0/0]
 via Serial0/1/0, directly
connected
L 2001:30:30:30::1/128 [0/0]
 via Serial0/1/0, receive
L FF00::/8 [0/0]
 via Null0, receive
Router1#
```

```
Router0#show ipv6 route
IPv6 Routing Table - 6 entries
Codes: C - Connected, L - Local,
S - Static, R - RIP, B - BGP
U - Per-user Static route, M -
MIPv6
I1 - ISIS L1, I2 - ISIS L2, IA -
ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter,
OE1 - OSPF ext 1, OE2 - OSPF ext
2
ON1 - OSPF NSSA ext 1, ON2 - OSPF
NSSA ext 2
D - EIGRP, EX - EIGRP external
O 2001:10:10:10::/64 [110/65]
 via FE80::200:CFF:FE99:6901,
Serial0/0/0
C 2001:20:20:20::/64 [0/0]
 via GigabitEthernet0/0, directly
connected
L 2001:20:20:20::1/128 [0/0]
 via GigabitEthernet0/0, receive
C 2001:30:30:30::/64 [0/0]
 via Serial0/0/0, directly
connected
L 2001:30:30:30::2/128 [0/0]
 via Serial0/0/0, receive
L FF00::/8 [0/0]
 via Null0, receive
Router0#
```

Figura 27e. Comprobación de la conectividad entre PC0 y PC1



Bibliografía

Una parte de los contenidos de este documento se han consultado de la web de Cisco o los materiales de los cursos CCNA de Netacad:

https://www.cisco.com/

https://www.netacad.com/

Además, algunos de los recursos se han obtenido del foro de la comunidad Cisco, y las páginas de los organismos IEEE (Institute of Electrical and Electronics Engineers), responsable de los estándares de la mayoría de tecnologías de red e IETF (Internet Engineering Task Force) , responsable de los estándares de internet y la publicación de los documentos RFC (Request For Comment):

https://community.cisco.com/

https://standards.ieee.org

https://tools.ietf.org/