

# Unidad 7

---



## Configuración y administración de protocolos dinámicos

### Planificación y Administración de Redes

# Índice



## 7.1. Tipos de enrutamiento

- 7.1.1. Enrutamiento estático
- 7.1.2. Enrutamiento dinámico

## 7.2. Enrutamiento estático

## 7.3. Mejoras en el diseño de redes y el enrutamiento

- 7.3.1. VLSM
- 7.3.2. CIDR

## 7.4. Enrutamiento dinámico

- 7.4.1. Sistema autónomo (AS)
- 7.4.2. Protocolos interiores y exteriores a un sistema autónomo
- 7.4.3. Protocolos según su algoritmo de enrutamiento
- 7.4.4. Protocolos híbridos
- 7.4.5. Clasificación de los protocolos de enrutamiento
- 7.4.6. Pasos para configuración de enrutamiento dinámico

## 7.5. RIP (v1)

## 7.6. RIP (v2)

## 7.7. RIPv6

## 7.8. OSPF

## 7.9. Características comunes de los protocolos de enrutamiento

## 7.10. Enrutamiento entre VLANs



## Introducción

Para que en una red se puedan interconectar dos enlaces, sobre todo cuando estos se encuentran a grandes distancias, necesitaremos que hay variedad en las rutas para que los destinos se encuentren.

Estas rutas se usan en gran mayoría para evitar que ningún equipo quede asilado de los demás en la red en caso de fallos de algún enlace. Además, cuando en ciertas rutas el tráfico es muy denso, se pueden usar otras para su descongestión.

Cuando una red como una WAN, de gran distancia, comienza a funcionar, necesita

de estas rutas sí o sí y determinar cuál de las rutas es la necesaria en cada caso. La técnica encargada de asignar las mejores rutas para cada paquete se conoce como enrutamiento y se engloba dentro del nivel de red.

Si queremos realizar un enrutamiento correctamente deberemos de tener muy claro los equipos y sus IPs.

El enrutamiento lo realizan los routers y en anteriores unidades ya hemos visto como trabaja en ciertos aspectos, ahora veremos los restantes.

## Al finalizar esta unidad

- + Conoceremos los tipos de enrutamiento que existen.
- + Sabremos manejar los protocolos presentes en el enrutamiento estático.
- + Podremos desarrollar las técnicas VLSM y CIDR.
- + Seremos capaces de determinar todos los tipos de protocolos que existen en el enrutamiento dinámico.
- + Sabremos como funcionan las tres versiones del protocolo RIP.
- + Podremos configurar de manera eficiente el protocolo OSPF.
- + Seremos capaces de enrutar distintas VLANs atendiendo a todo lo que tratemos a lo largo de la unidad.



# 7.1.

## Tipos de enrutamiento

Basándonos en como el *router* identifique las direcciones de destino, el enrutamiento será o estático o dinámico.

### 7.1.1. Enrutamiento estático

Las tablas de enrutamiento estático deben de ser definidas por el administrador de la red. Las entradas de la tabla de enrutamiento deben de crearse manualmente cuando una ruta no se va a mover y además, o solo existe esta, o el administrador piensa que siempre va a ser la más adecuada. Otras veces también se usan por temas de seguridad, ya que no se podrían captar por terceras personas con la intención de conocer el contenido de la red.

En este tipo de enrutamiento el *router* no conoce la topología de la red por sí solo, pero su velocidad de encaminamiento es mucho más alta.

### 7.1.2. Enrutamiento dinámico

Esta técnica de enrutamiento se fundamenta en la creación de protocolos de enrutamiento dinámico con los cuales se produce un intercambio de información entre varios *routers* y se crea de manera dinámica la tabla de enrutamiento.

Estos *routers* importan la topología de red por sí solos, siendo mucho más flexibles que los estáticos, pero con un menor rendimiento.

Los dos enrutamientos se pueden complementar entre sí, porque no son excluyentes.

Muchas veces, lo que se hace es configurar dinámicamente un *router* y añadir unas cuantas entradas estáticas por temas de seguridad.

El enrutamiento dinámico persigue los siguientes objetivos:

#### > Mejor ruta:

Dependiendo del criterio que se escoja, distancia, menor, tráfico o mayor capacidad, se conseguirá la mejor ruta y será el primordial objetivo.

#### > Simplicidad:

Se busca que la técnica de enrutamiento no nos consuma muchos recursos.

#### > Robustez:

Se persigue con el enrutamiento dinámico que existan mecanismos que eviten el bloqueo cuando haya fallos en la red, para que el *router* siga dando servicio.

#### > Rapidez de aprendizaje:

Cada vez que se produzca un cambio en la red, el protocolo dinámico permitirá que los *routers* conozcan este cambio lo antes posible y así modificar sus tablas de enrutamiento.

Los protocolos dinámicos más usados son RIP, EIGRP, IS-IS y OSPF.



# 7.2.

## Enrutamiento estático

Cada uno de los equipos que haya en la red, tienen una dirección IP que es única para ellos dentro de su red. Los *routers* también deben de contar con una única dirección IP por cada red a la que conecten que, además, en muchas ocasiones, será el *gateway* de los equipos de la red. Debe de tener tantas IPs como puertos activos.

Este tipo de enrutamiento se usa para establecer un formato jerárquico sobre todas las direcciones IP, primero red, después subred y por último el host. Este formato se usa con el fin de que el algoritmo que decide el enrutamiento sea mucho más sencillo y que los nodos de red se vean optimizados en cuanto a recursos.

### Tabla de enrutamiento

Todos y cada uno de los *routers* de una red deben de disponer de una tabla de enrutamiento donde se definen los destinos de los datos enviados.

La tabla de enrutamiento debe de contener alguna de las columnas siguientes:

- > Dirección IP de la red de destino
- > Máscara de la red de destino.
- > Siguiente salto o *next hop*. Aquí distinguimos dos posibilidades:
  - » La dirección IP del puerto del *router* al que se le entrega el paquete si la red de destino se encuentra directamente conectada. En *routers* CISCO es 0.0.0.0.
  - » La dirección IP del puerto del *router* siguiente, cuando la red de destino **no** esté directamente conectada.
- > Interfaz → nombre o dirección IP del puerto del *router* donde se entregará el paquete.
- > Métrica → mide el número de *routers* que debemos de atravesar o incluso cualquier parámetro que nos indique que nos va a costar dicho camino.

### Observaciones:

- > En la tabla de enrutamiento debemos de incluir todas las redes de destino que existen.
- > Puede que una red de destino aparezca en varias filas de una misma tabla, esto es correcto cuando hay distintas rutas hacia una misma red,
- > Cuando la red se encuentra conectada a internet, la red de destino siempre será 0.0.0.0/0, es decir, todas las direcciones no englobadas aquí.
- > Red destino y siguiente salto son las únicas columnas obligatorias.

### Pasos para configuración del enrutamiento estático en Packet Tracer

1. Construir la red
2. Configurar la IP
  - a. En estaciones de trabajo, IP, máscara y *gateway*.
  - b. En *routers*, IP y máscara en cada puerto. Hay que activar el puerto.
3. Tablas de enrutamiento estático:
  - a. Identificar las redes que haya y etiquetarlas.
  - b. Pestaña: *Config* → *Routing* → *Static*
  - c. Configurar la red que será de destino, la máscara y el siguiente salto para todas las redes.
4. Comprobar que los equipos tienen conexión.





# 7.3.

## Mejoras en el diseño de redes y el enrutamiento

Fue en los años 1980 cuando las IPs empezaron a escasear, lo que se veía en que los *routers* comenzaron a tener muchas más tablas de enrutamiento. Con la intención de intentar frenar estas circunstancias, se crean diversas técnicas que aportan flexibilidad.

### 7.3.1. VLSM

VLSM, son las siglas de las máscaras de subred de longitud variable, **Variable Length Subnet Mask**. Esta técnica surge como la primera de las técnicas que quieren evitar el desperdicio de IPs que tenía como consecuencia la técnica de diseño mediante clases.

#### Ejemplo

Si tenemos la red 192.168.1.0./24, nuestra intención es que tengamos un esquema de direccionamiento con el cual el espacio de la red este optimizado. Debemos de cumplir además con los siguientes requerimientos de subredes:

- > Una subred de 40 que se asigne a la VLAN de Alumnos de 1º.
- > Una subred de 80 que se asigne a la VLAN de Alumnos de 2º.
- > Una subred de 20 que se asigne a la VLAN de Alumnos de 3º.
- > Una subred de 5 direcciones para asignar a los enlaces entre *routers*.

Vamos a hacer la que sería nuestra posible solución:

1. Ordenamos las subredes en orden decreciente según el número de hosts: 80, 40, 20, 5
2. Si queremos disponer de los primeros 80 hosts, debemos de usar 7 bits, porque  $2^7 = 128$ , que es el siguiente mayor. Es decir, el prefijo de subred del primero de los bloques será /25.

Si la primera de las subredes que cogemos es la que hemos definido en primera instancia, tendríamos que S1 = 192.168.1.0/25. Y su broadcast es 192.168.1.127. En esta red el rango asignable es desde 192.168.1.1 hasta 192.168.1.126.

Ya tendríamos por lo tanto dos subredes, S1 = 192.168.1.0/25 y S2 = 192.168.1.128/25.

3. La siguiente subred que tenemos en realidad, es la de 40 hosts, para la que necesitamos 6 bits. Por lo tanto, el prefijo de red es /26.

La red será por lo tanto S2 = 192.168.1.128/26, su *broadcast* es 192.168.1.191, por lo tanto, su rango es desde 192.168.1.129 hasta 192.168.1.190.



Si dividimos S2 en otras 2 subredes, se quedan:

- » S2.1 = 192.168.1.128/26
- » S2.2 = 192.168.1.192/26

4. Lo siguiente que tenemos es la subred con 20 hosts, que esta usará 7 bits y por lo tanto, será, 192.168.1.192/27. Su *broadcast* será 192.168.1.223 u el rango irá desde la 193 hasta la 222.

Se nos quedan entonces:

- » S1 = 192.168.1.0/25
- » S2.1 = 192.168.1.128/26
- » S2.2.1 = 192.168.1.192/27
- » S2.2.2 = 192.168.1.224/27

5. El siguiente paso es coger la S2.2.2, y usarla para los 5 enlaces por *routers*. Esto es similar a coger 5 bits, Esto se traduce en que esta misma subred nos sirve para el ejemplo.

Subredes

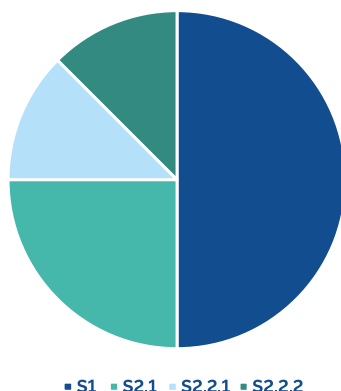


Imagen 1. Subredes del ejemplo VLSM

Por lo tanto, se nos queda el siguiente esquema:

Esquema de direcciones					
#	Nombre	Dirección	Broadcast	Rango IP	Máscara
S1	Alumnos de 2º	192.168.1.0/25	192.168.1.127	1 a 126	255.255.255.128
S2.1	Alumnos de 1º	192.168.1.128/26	192.168.1.191	129 a 190	255.255.255.192
S2.2.1	Alumnos de 3º	192.168.1.192/27	192.168.1.223	193 a 222	255.255.255.224
S2.2.2	Enlace	192.168.1.224/27	192.168.1.255	225 a 254	255.255.255.224

### 7.3.2. CIDR

Esta notación lo que hace es exponer la dirección IP y justo a continuación una barra que tiene seguidamente el número de unos que se contiene en la máscara de subred, un ejemplo de esto sería 192.168.1.10/24.



# 7.4.

## Enrutamiento dinámico

Para el enrutamiento dinámico no existe ningún protocolo que nos consiga todos los objetivos al 100%. Pero sí que hay algunos protocolos que nos ayudan más que otros dependiendo de la situación en la que nos encontremos.

Para poder clasificar dichos protocolos, necesitamos primeramente definir un tipo concreto de clasificación, y hay varios tipos que dependerá de diversas características.

### 7.4.1. Sistema autónomo (AS)

Un **sistema autónomo** o AS, de sus siglas en inglés, es una red que encontramos dentro de Internet y que se administra por una sola empresa u organización, que será o un proveedor de servicios o un organismo público. Los *routers* que se encuentran en dicho sistema, se comunican entre sí con el mismo protocolo de enrutamiento.

Los sistemas autónomos se encuentran perfectamente separados del resto de la red WAN donde se encuentre asilado. La ICANN se encarga del registro de todos los AS a través de sus Registros Internacionales. En Europa, el registro es RIPE-NCC.

### 7.4.2. Protocolos interiores y exteriores a un sistema autónomo

Si un protocolo funciona de manera interior o exterior a un sistema autónomo, su clasificación es la siguiente:

- > **IGP, Interior Gateway Protocol:** Funcionan dentro de un sistema autónomo. Estos son:
  - » RIP
  - » EIGRP
  - » RIPv2
  - » OSPF
  - » RIPv6
  - » IS-IS
  - » IGRP
- > **EGP, Exterior Gateway Protocol:** Son los protocolos que enrutan paquetes entre distintos sistemas autónomos y, por lo tanto, funcionan fuera de estos. Su protocolo más importante es BGP.





### 7.4.3. Protocolos según su algoritmo de enrutamiento

---

#### Vector de distancias

Los protocolos pertenecientes a este grupo funcionan de modo que cada uno de los *routers* solo envían la información a los *routers* más cercanos, por lo que no necesitan saber cual es la topología de la red. Las distancias entre ellos se mantienen y se usa el algoritmo de Bellman-Ford para calcular las rutas.

Estos protocolos, aunque lentos, son fáciles de manejar y bastante eficientes para redes con pocos hosts.

Se engloban aquí los siguientes protocolos:

- » RIPv1
- » RIPv2.
- » RIPv6
- » IGRP

#### Estado del enlace

Los protocolos de estado del enlace cuentan con unas tablas de enrutamiento bastante complejas donde se almacena la topología de toda la red en conjunto.

Cuando un *router* recibe la información, comienza a idear un árbol jerárquico con todos los destinos y enrutadores intermedios. Con esto se consigue que cada *router* pueda determinar por sí solo toda la ruta con saltos intermedios hasta el destino, sin importar los nodos de red que haya.

Aunque es el mejor método, su consumo de recursos es bastante más alto.

Se engloba aquí principalmente el protocolo OSPF.

### 7.4.4. Protocolos híbridos

---

Este es una mezcla de los dos tipos anteriores de protocolos.

Cuando la topología de red cambia, es cuando los *routers* envían la información a los demás. Las métricas usadas son más complejas que simplemente calcula los saltos.

En esta familia se adhieren los protocolos EIGRP e IS-IS.



### 7.4.5. Clasificación de los protocolos de enrutamiento

Aquí podemos ver una primera clasificación general de los protocolos de enrutamiento dinámico de los que hemos ido hablando:

	Clasificación de protocolos de enrutamiento dinámico				
	Protocolos interiores				Protocolos exteriores
	Vector de distancias		Estado del enlace		
Con clase	RIP	IGRP			EGP
Sin clase	RIP 2	EIGRP	OSPF 2	IS-IS	BGP v4
IPv6	RIPng	EIGRP v6	OSPF 3	IS-IS v6	BGP v6

Si partimos de esta clasificación, cuando un *router* tiene su tabla vacía o si no conoce aún la topología de red, se puede usar:

- > **Algoritmo de la patata caliente:** en cuanto el *router* recibe el paquete, lo envía lo más rápido posible por la ruta con menor tráfico.
- > **Algoritmo de inundación:** si no conocemos la ruta por la que enviar la información, se envía por todas las rutas existentes. Lógicamente el tráfico de la red aumenta considerablemente.

Si queremos evitar la saturación de la red, los *routers* eliminan los paquetes con el TTL excedido.

### 7.4.6. Pasos para configuración de enrutamiento dinámico

1. Construir la red.
2. Direccionamiento IP
  - a. Equipos terminales
  - b. Routers
3. Configuración de un protocolo de enrutamiento dinámico en cada *router*
4. Comprobación de la conectividad.



# 7.5.

## RIP (v1)

Dentro de los protocolos de enrutamiento dinámico, encontramos que uno de los más usados es RIP, *Routing Information Protocol*.

Su nombre es RIP, pero suele escribirse como RIP v1 para poder diferenciarlo de la segunda versión de este protocolo.

Sus principales características son:

- > Se trata de un protocolo de vector-distancia.
- > Si queremos calcular su métrica, solo tenemos en cuenta el número de salto. Su capacidad máxima es de 15 saltos, si alcanzamos los 16, el destino se considera inalcanzable.
- > Las actualizaciones de las tablas enviadas a los *routers* más cercanos se producen cada 30 segundos. Dichas actualizaciones se envían sin conexión mediante el protocolo UDP por la dirección de difusión y el puerto 520.
- > Siempre se va a enviar la información por la ruta que tenga métrica menor. Si tenemos varias rutas con la misma métrica, la carga de trabajo se balancea.

Este protocolo es sencillo de implementar y además su consumo de recursos es menor que el de otros, pero cuenta con ciertas limitaciones como las siguientes que nos hacen optar por su versión más avanzada:

- > No se tiene en cuenta la velocidad de transmisión de los enlaces, es decir, puede que las rutas, aunque con menos saltos, sea más lenta.
- > Como el mensaje de actualización de rutas se hace mediante difusión, el tráfico de la red se incrementa.
- > No soporta CIDR ni VLSM.
- > No existen mecanismos de seguridad que paren mensajes de actualización falsos.

### Configuración de RIPv1 en routers CISCO

1. Accedemos al *router* y lo habilitamos como vimos en la unidad referente a la administración de estos.
2. Accedemos al modo de configuración del *router*.
3. Lanzamos los siguientes comandos:

a. `router rip`

b. `network dirección_de_red`

```
Router(config)#router rip
Router(config-router)#network 192.168.1.0
Router(config-router)#
```

Imagen 2. Configuración de RIPv1

4. Comprobamos la ruta de red con el comando `show ip route`.

# 7.6.

## RIP (v2)

Este protocolo surge como una evolución a RIP, para mejorar los problemas que mostraba este. Se trata de un protocolo de enrutamiento dinámico sin clase y sus mejoras frente a su versión anterior son las siguientes:

- > Soporta subredes, CIDR y VLSM.
- > Los mensajes de actualización se envían por *multicas* a la dirección 224.0.0.9, lo que hace que el tráfico de red se vea francamente reducido.
- > Se necesita de autenticación para los mensajes de actualización, lo que incrementa en gran medida la seguridad.

### Configuración de RIPv2 en routers CISCO

Se usan los siguientes comandos:

1. `router rip`
2. `versión 2`
3. `network dirección_de_red`
4. `exit`

```
Router(config)#  
Router(config)#router rip  
Router(config-router)#version 2  
Router(config-router)#network 192.168.1.0  
Router(config-router)#exit  
Router(config)#
```

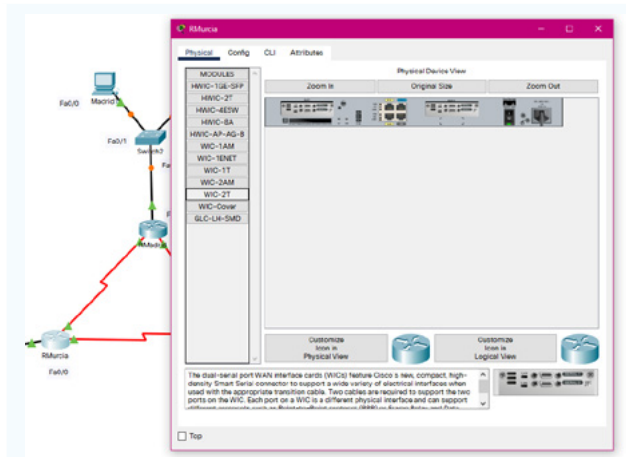
Imagen 3. Configuración de RIPv2

### Ejemplo

#### 1. Conexiones en los routers.

Lo primero que haremos, será configurar los dispositivos para poder conectarlos entre sí.

#### Instalación de los módulos



#### PASO 1:

Para poder conectar los router entre sí, necesitamos añadir el módulo WIC-2T

#### NOTAS:

No olviden encender el módulo en la configuración **Serial** que utilices.



Una vez configurado los *routers* con los módulos, los conectaremos entre sí.

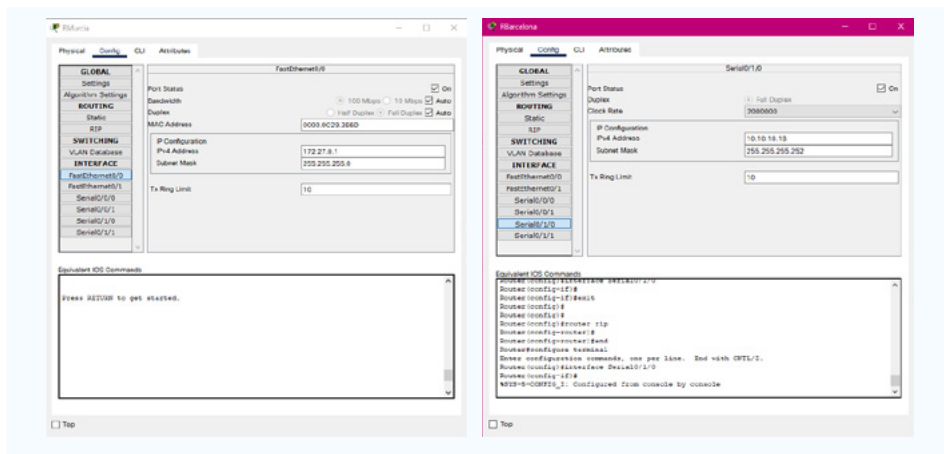
## 2. Estructura de red.

Seguimos con la estructura de la red, vamos a ir asignando direcciones IP, máscaras de subred y Gateway, tanto a los equipos como a los Routers.

### Estructura de red

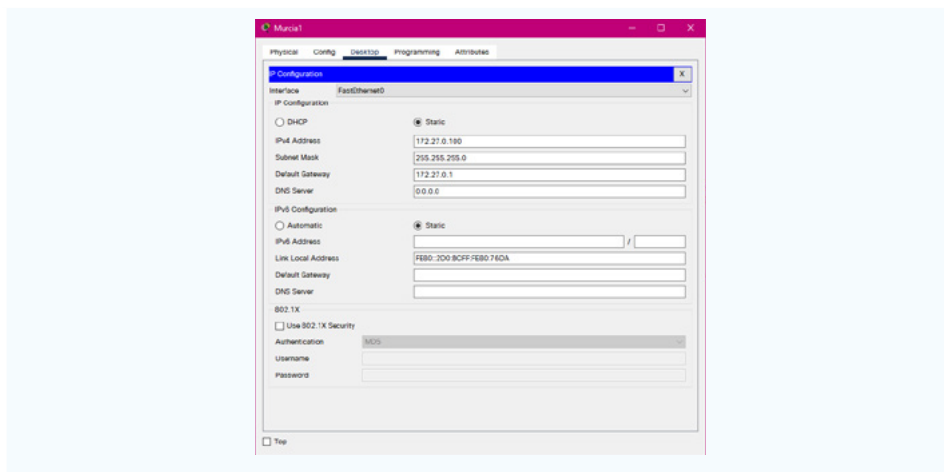
#### Configuración de los routers

Ingresamos tanto la interfaz **FastEthernet0/0** tanto la interfaz **Serial**



#### Configuración de los ordenadores

Ingresamos tanto la interfaz **dirección ip, máscara de red y Gateway**.



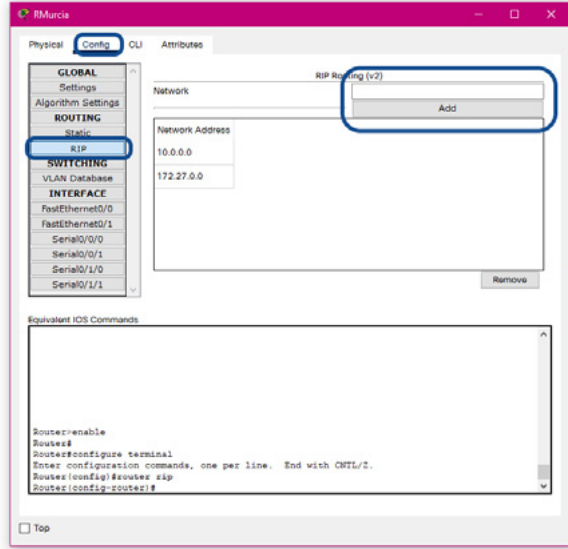


### 3. Configuración del protocolo RIPv2

Llegamos a la parte más importante de la práctica, que es en sí, donde reside el protocolo RIPv2.

Podemos realizarlo de dos maneras, mostraremos ambas, pero desarrollaremos la parte de **terminal**.

**Protocolo RIPv2**



**EXPLICACIÓN 1:**

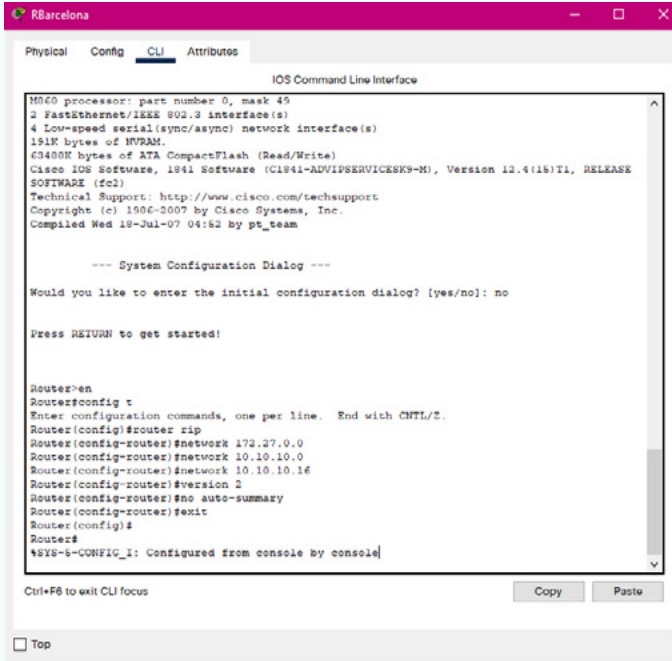
Para poder activar este protocolo, nos tendremos que posicionar en la configuración del router.

En esta primera explicación, veremos la parte "gráfica"

En esta parte, hay que ir introduciendo las redes que vayamos a trabajar.

#### EXPLICACIÓN 2:

La práctica en sí, es sencillo, tenemos que añadir las redes que vayamos a trabajar a cada router en la parte de enrutamiento RIP. Para conseguir este protocolo a través del terminal se realiza la siguiente configuración.







### Router Murcia

```
Router>en
Router#config t
Enter configuration commands, one per line. End with
CNTL/Z.
Router(config)#router rip
Router(config-router)#network 172.27.0.0
Router(config-router)#network 10.10.10.0
Router(config-router)#network 10.10.10.16
Router(config-router)#version 2
Router(config-router)#no auto-summary
Router(config-router)#exit
```

### Router Madrid

```
Router>en
Router#config t
Enter configuration commands, one per line. End with
CNTL/Z.
Router(config)#router rip
Router(config-router)#network 172.27.1.0
Router(config-router)#network 10.10.10.0
Router(config-router)#network 10.10.10.8
Router(config-router)#version 2
Router(config-router)#no auto-summary
Router(config-router)#exit
```

### Router Barcelona

```
Router>en
Router#config t
Enter configuration commands, one per line. End with
CNTL/Z.
Router(config)#router rip
Router(config-router)#network 172.27.2.0
Router(config-router)#network 10.10.10.8
Router(config-router)#network 10.10.10.16
Router(config-router)#version 2
Router(config-router)#no auto-summary
Router(config-router)#exit
```

### ¿Por qué usamos no auto-summary?

El comando no auto-summary evita que RIP haga un resumen automático de la red, si no lo hacemos así, los routers no van a ser capaces de conocer las subredes de esa red principal. De esta forma los forzamos a que publiquen las subredes tal como son.

Llegados a este punto, ya tenemos terminada la práctica. Teniendo una red capaz de poder dirigir el tráfico en casa de que algún *router* caiga o no funcione por cualquier motivo.



# 7.7.

## RIPng

RIPng, *Rip Next Generation*. Se trata de una extensión de RIPv2 que incorpora IPv6.

No se puede desarrollar este protocolo en todos los *routers* porque no todos soportan IPv6.

Las mayores diferencias entre ambos protocolos, RIP v2 y RIPng, son:

- > Soporta IPv6.
- > No se soporta la autenticación para los mensajes de actualizaciones en las tablas. Esto se debe a que los routers IPv6 suelen usar IPSec para temas de autenticación.
- > En RIPng no se pueden etiquetar las rutas mientras que en RIPv2 sí.
- > La dirección de multicast usada en RIPng es FF02::9.

### Configuración de RIPng en routers CISCO

Los comandos que más se usan en RIPng son:

Comandos para RIPng en CISCO	
Comando	Utilidad
<code>ipv6 unicast-routing</code>	Habilita el tráfico IPv6.
<code>ipv6 address direccion/longitud-prefijo eui-64</code>	Configura la dirección IPv6 en la dirección que seleccionemos.
<code>ipv6 router rip nombre</code>	Crea el modo de configuración de rip y accede a él.
<code>ipv6 rip nombre enable</code>	Configura rip en la interfaz que seleccionemos.

Cuando accedemos a un *router* IPv6 estará siempre deshabilitado.



# 7.8.

## OSPF

El protocolo OSPF, *Open Shortest Path First*, hace referencia al enrutamiento interior basándose en el estado del enlace para calcular cual es la ruta más adecuada.

Este protocolo es más rápido en comparación con otros basados en vector de distancias. La principal ventaja de los protocolos de estado de enlace es que solo notifican cuando sufre un cambio la red, lo cual es una ventaja porque el consumo del ancho de banda es mínimo.

OSPF no se basa en ningún protocolo de transporte de TCP/IP, este realmente se encapsula en un datagrama IP con código de protocolo 89. LA dirección de destino de dichos datagramas siempre será alguna de las *multicast* que reserva el protocolo: 224.0.0.5 o 224.0.0.6. El datagrama IP donde se contiene los datos de OSPF está encapsulado a su vez dentro de una trama Ethernet donde la dirección MAC de destino es alguna de las *multicast* 01-00-5E-00-00-05 o 01-00-5E-00-00-06.

Soporta VLSM y CIDR.

La versión 3 de este protocolo soporta IPv6 y además, extensiones multidifusión.

### Coste (métrica)

La medida de la métrica que usa OSPF se denomina coste. Este valor se apoya en los siguientes parámetros para dar un dato numérico final: ancho de banda y congestión de los enlaces. La interfaz de menor coste será posiblemente la más usada para el envío de los datos. Cuando haya cargas del mismo coste, el propio protocolo realizará un balanceo de la carga.

Para calcular el coste de OSPF se usa la siguiente formula:

**Coste Interfaz = Ancho de banda de referencia / Ancho de banda de la interfaz**

El ancho de banda de esta formula siempre suele ser de 100 Mbps,  $10^8$ . Es decir, por defecto, el coste de la interfaz siempre va a ser  **$10^8$  / ancho de banda de la interfaz**.

Los costes para las interfaces más usadas son:

- > Ethernet: Coste por defecto =  $10^8 / 10^7 = 10$ .
- > Fast Ethernet: Coste por defecto =  $10^8 / 10^8 = 1$ .
- > FDDI: Coste por defecto 1.
- > ATM: Coste por defecto 1.

Cualquier interfaz que sea más rápida de  $10^8$  tendrá un coste 1. Esta medida hace que no se pueda comparar de manera eficiente y con precisión las interfaces con un ancho de banda superior. Para poder usar estas medidas correctamente, habrá que modificar el valor del ancho de banda de referencia, lo que nos dará un valor más exacto.



En los *routers* CISCO lo más usado es el comando:

`auto-cost reference-bandwidth`

Este comando nos permite que se cambie el ancho de banda de referencia que OSPF usa para poder calcular las métricas.

### Tipos de paquetes

OSPF usa varios tipos de paquetes distintos que se usan para que todos los dispositivos de la red conozcan la topología de la red e informa del estado en el que están los nodos de la red. Cada tipo de paquete tiene un código numérico asociado que se refleja en el campo tipo de la cabecera OSPF. Los tipos son:

- > **Paquetes Hello tipo 1.** El *router* envía a los *routers* más próximos la lista de los conocidos, indicando como se relaciona con ellos.
- > **Paquetes de descripción de base de datos estado del enlace tipo 2.** Estos paquetes se usan para el intercambio de las bases de datos de estado del enlace entre nodos.
- > **Paquetes de solicitud del estado de los enlaces, tipo 3.** Son usados por algún *router* que quiera específicamente, conocer información acerca de alguna entrada que aparezca en alguna descripción de base de datos recibida.
- > **Paquetes de estado de enlace.** Son los usados para la información dada a la red cuando hay un cambio de estado en los enlaces de alguno de los *routers*.

### Áreas

Para trabajar con OSPF, lo que debemos de hacer es organizar un sistema autónomo de áreas, que identifiquemos en números de 32 bits. Estas áreas son grupos de enrutadores de los que extraemos información ya resumida que se envía a los demás enrutadores. La misma área también es una unidad de enrutamiento, o sea, que todos los *routers* que pertenecen a dicho área, guardan información topológica en su base de datos del estado del enlace. Con esto, lo que conseguimos es que, si hay un cambio en la red, en una parte en concreto, no cambia toda la red, por lo que el tráfico que genere este cambio se queda restringido a esta área únicamente.

Tenemos los siguientes tipos para la clasificación de las áreas en OSPF:

- > **Backbone:** se trata del núcleo de la red con OSPF implementado. Es también llamado área 0. Toda red OSPF cuenta con un *backbone* y este siempre va a mantener una conexión con todas las demás áreas, ya sea física o lógica.
- > **Stub:** estas son las áreas que no reciben rutas externas porque se han implementado en OSPF mediante otro protocolo de enrutamiento distinto.
- > **Not-so-stubby:** un subtipo dentro de las áreas *Stub*. Importan rutas externas que vengan de sistemas autónomos diferentes para enviarlas al *Backbone*. Si que no pueden exportar rutas del núcleo al exterior.



### Recomendaciones de diseño en *routers* CISCO

1. Un *router* no debe de estar al mismo tiempo en más de 2 o 3 áreas al mismo tiempo debido a sus limitados recursos.
2. Un área puede contener desde 30 hasta 90 *routers*, menos sería un malgasto y más sería una sobreexplotación del área.
3. Un mismo *router* no debe de tener más de 50 conexiones directas.
4. El área más importante que no hay que sobredimensionar es el área 0, porque se nos complicaría la escalabilidad.

### Configuración en *routers* CISCO

Los siguientes comandos son los más usados para la configuración de OSPF en un *router* CISCO:

Comandos para configurar OSPF en CISCO		
Comando	Modo	Uso
<code>show ip route ospf</code>	Router#	Muestra la tabla de enrutamiento para OSPF
<code>show ip ospf</code>	Router#	Información sobre OSPF.
<code>show ip ospf interface</code>	Router#	Muestra información de una interfaz que usa OSPF
<code>show ip ospf database</code>	Router#	Contenido De la BD OSPF
<code>router ospf id_proceso</code>	Router(config)#	Pasa a modo de configuración del protocolo OSPF en el <i>router</i> .
<code>network dirección_de_red máscara área id_area</code>	Router(config-router)#	Configura la red, la máscara inversa y el área a la que pertenece el <i>router</i> .
<code>passive-interface interfaz</code>	Router(config-router)#	No envía paquetes <i>Hello</i> y descarta los entrantes.
<code>do show ip ospf interface interfaz</code>	Router(config-router)#	Muestra la información sobre una interfaz interviniente en un enrutamiento OSPF.
<code>auto-cost reference-bandwidth valor</code>	Router(config-router)#	Se fija el valor del ancho de banda de referencia para OSPF.
<code>neighbor IP_vecino cost valor</code>	Router(config-router)#	Fija el valor del coste de un vecino.
<code>ip ospf cost valor</code>	Router(config-if)#	Fija el coste de una interfaz a un valor entre 1 y 65.535
<code>ip ospf priority valor</code>	Router(config-if)#	Establece la prioridad de una interfaz con el fin de facilitar que enrutador designado elegir.
<code>ip ospf hello-interval intervalo</code>	Router(config-if)#	Intervalo de los paquetes <i>Hello</i> . Por defecto, en Ethernet es cada 10 segundos.
<code>ip ospf dead-interval intervalo</code>	Router(config-if)#	Intervalo en el que consideramos un enlace como no operativo.



# 7.9.

## Características comunes de los protocolos de enrutamiento

### Métrica

Cualquier protocolo dinámico va a avalorar cual de las rutas es más adecuada midiendo rapidez y/o fiabilidad. Esta valoración cuantitativa se llama métrica.

Hay diferentes tipos de parámetros usados para medir la métrica, desde el número de saltos hasta la tasa de mensajes que alcanzan su mensaje frente a los perdidos.

### Equilibrado de carga

En casi cualquier protocolo dinámico, está aceptado que haya varias entradas para un mismo destino en su tabla de enrutamiento, aquí se pueden tener el mismo o distintos valores para la métrica. Con esto, lo que se persigue y casi siempre se consigue es que el tráfico quede repartido, ya que el *router* envía la información valiéndose de varias rutas parecidas y no siempre por la misma, aunque el destino vaya a ser el mismo. Esto que nos ayuda a descongestionar la red, se conoce como el balanceado de carga.

### TTL. Time to live

Cada vez que se comienza con una red, y los *routers* comienzan a funcionar, es muy común que en sus tablas de enrutamiento se produzcan una serie de entradas incoherentes que no tienen mucho sentido porque no se conoce toda la topología de la red. Hasta que conseguimos llegar al **estado de convergencia**, que es cuando toda la información contenida en las tablas de enrutamiento es correcta y consistente, hay muchos paquetes que no llegan bien a los destinos, y7 que se quedan circulando eternamente por la red. Para que esto no suceda, los paquetes tienen un TTL o tiempo de vida que determina los números de saltos totales que da el paquete antes de que un *router* lo elimine. Si cuando un paquete llega a un *router*, su TTL es 0, lo eliminará.

### Distancia administrativa

Si tenemos que dos o más protocolos nos ofertan una ruta al mismo destino, para saber que ruta tomar, los *routers* usan la distancia administrativa. Esta medida se basa en la confianza que se tiene acerca de la fuente que nos informa de dicha ruta. Esto solo cubre la parte local, es decir, cada *router* mide las suyas.

Cuanto más pequeño sea el valor de distancia administrativa, más confiable será y por tanto, antes elegida.

Las distancias administrativas más conocidas son:

Distancias administrativas	
Protocolo	Distancia administrativa
Interfaz directamente conectada	0
Ruta estática	1
EIGRP	5
BGP externo	20
EIGRP interno	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
EIGRP externo	170
Desconocido	255





# 7.10.

## Enrutamiento entre VLANs

Este tipo de enrutamiento se trata de intentar conectar equipos que se encuentran en VLANs distintas a través de un único *router* con una única interfaz. Para esto, lo que se hace es definir interfaces lógicas dentro de la interfaz principal, subinterfaces, de las cuales, debe de haber tantas como VLANs queramos conectar. En cada una de las subinterfaces tendremos una IP que pertenezca a la VLAN asociada. Esta IP será el *gateway* de todos los equipos que existan en una misma VLAN.

### Ventajas de configurar enrutamiento entre VLAN

- > Todos los equipos cuentan con salida a internet.
- > La red cuenta con dominios de segmentación más pequeños.
- > La posibilidad de que puedan conectar entre ellos equipos pertenecientes a diferentes VLAN.



 [www.universae.com](http://www.universae.com)

