

# Informe

## Práctica 4: Enrutamiento dinámico con OSPF

### Laboratorio de Redes



Diego Cruz Rodríguez  
Universidad de La Laguna  
Ingeniería Informática  
3º Curso, 2º Semestre  
27/03/2020

# Objetivos

- Conocer el funcionamiento del protocolo OSPF (Open Shortest Path First) y sus características.
- Entender las diferencias con el protocolo RIP.
- Ser capaz de hacer funcionar el enrutamiento dinámico mediante OSPF en una red.
- Comprender la utilidad de las áreas de tipo stub y totally stub.
- Entender la sumarización de rutas y las repercusiones en el diseño del esquema de direccionamiento.
- Ser capaz de manejar los costes de los enlaces.
- Configurar la propagación de rutas por defecto en OSPF.

# Protocolo OSPF

Open Shortest Path First (OSPF) es un protocolo de direccionamiento de tipo enlace-estado, desarrollado para las redes IP y basado en el algoritmo de primera vía más corta (SPF). OSPF es un protocolo de pasarela interior (IGP). En una red OSPF, los direccionadores o sistemas de la misma área mantienen una base de datos de enlace-estado idéntica que describe la topología del área. Cada direccionador o sistema del área genera su propia base de datos de enlace-estado a partir de los anuncios de enlace-estado (LSA) que recibe de los demás direccionadores o sistemas de la misma área y de los LSA que él mismo genera. El LSA es un paquete que contiene información sobre los vecinos y los costes de cada vía. Basándose en la base de datos de enlace-estado, cada direccionador o sistema calcula un árbol de extensión de vía más corta, siendo él mismo la raíz, utilizando el algoritmo SPF.

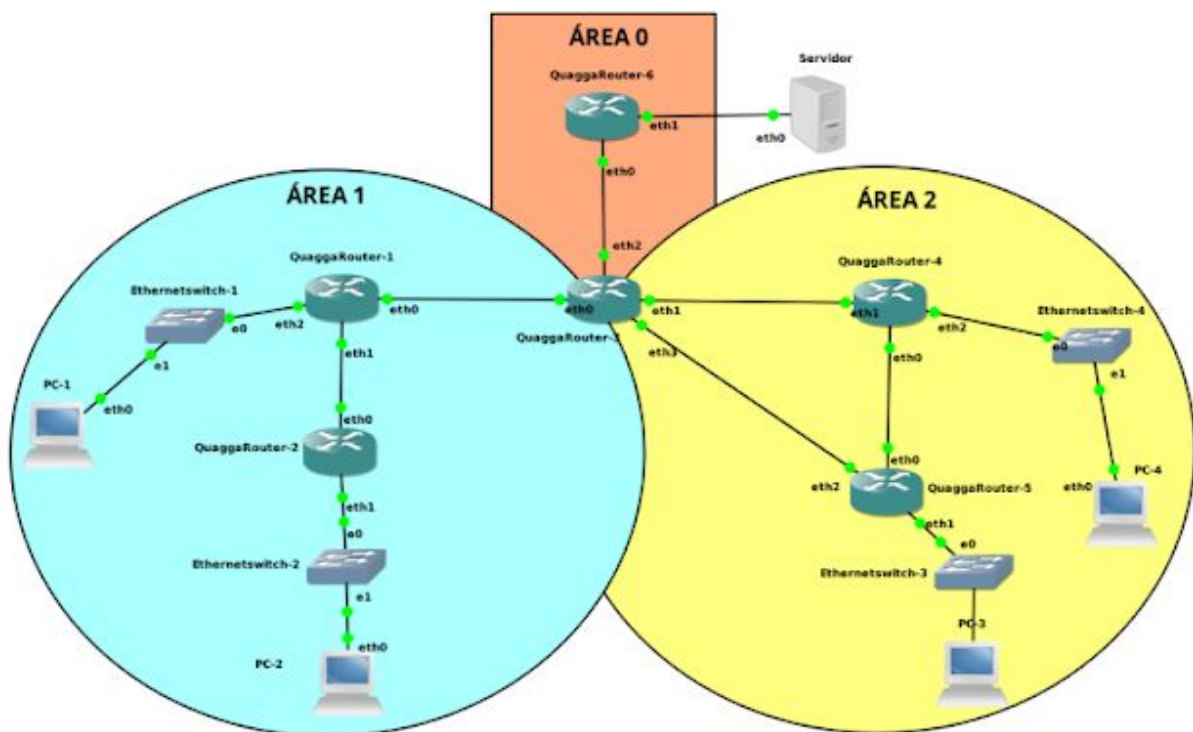
Las ventajas principales de OSPF son las siguientes:

- En comparación con los protocolos de direccionamiento de distancia-vector como el protocolo de información de direccionamiento (RIP), OSPF es más adecuado para servir entre redes heterogéneas de

gran tamaño. OSPF puede recalcular las rutas en muy poco tiempo cuando cambia la topología de la red.

- Con OSPF, puede dividir un sistema autónomo (AS) en áreas y mantenerlas separadas para disminuir el tráfico de direccionamiento de OSPF y el tamaño de la base de datos de enlace-estado de cada área.
- OSPF proporciona un direccionamiento multivía de coste equivalente. Se pueden añadir rutas duplicadas a la pila TCP utilizando saltos siguientes distintos.

## Topología



Esquema de la topología de red.

Dispositivo	Interfaz	Dirección/Máscara
QuaggaRouter-1	eth0	10.0.0.10/30
QuaggaRouter-1	eth1	10.0.0.21/30
QuaggaRouter-1	eth2	10.1.0.1/24
QuaggaRouter-2	eth0	10.0.0.22/30
QuaggaRouter-2	eth1	10.1.1.1/24
QuaggaRouter-3	eth0	10.0.0.9/30
QuaggaRouter-3	eth1	10.0.0.13/30
QuaggaRouter-3	eth2	10.0.0.6/30
QuaggaRouter-3	eth3	10.0.0.26/30
QuaggaRouter-4	eth0	10.0.0.17/30
QuaggaRouter-4	eth1	10.0.0.14/30
QuaggaRouter-4	eth2	10.2.1.1/24
QuaggaRouter-5	eth0	10.0.0.18/30
QuaggaRouter-5	eth1	10.2.0.1/24
QuaggaRouter-5	eth2	10.0.0.25/30
QuaggaRouter-6	eth0	10.0.0.5/30
QuaggaRouter-6	eth1	10.0.0.1/30
PC1	eth0	10.1.0.2/24
PC2	eth0	10.1.1.2/24
PC3	eth0	10.2.0.2/24
PC4	eth0	10.2.1.2/24
Servidor	eth0	10.0.0.2/30

Direcciones de red en cada interfaz.

## 1º Configurar el área 0

```

Debian(config)# hostname QuaggaRouter-6
QuaggaRouter-6(config)# router ospf
QuaggaRouter-6(config-router)# network 10.0.0.4/30 area0
% Unknown command.
QuaggaRouter-6(config-router)# network 10.0.0.4/30 area 0
QuaggaRouter-6(config-router)# network 10.0.0.0/30 area 0
QuaggaRouter-6(config-router)# exit
QuaggaRouter-6(config)# exit
QuaggaRouter-6# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

O 10.0.0.0/30 [110/10] is directly connected, eth1, 00:01:01
C>* 10.0.0.0/30 is directly connected, eth1
O 10.0.0.4/30 [110/10] is directly connected, eth0, 00:01:09
C>* 10.0.0.4/30 is directly connected, eth0
C>* 127.0.0.0/8 is directly connected, lo

```



## 2º Configurar el Resto de áreas

### Área 2

- Router 5

```
QuaggaRouter-5# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route
C>* 10.0.0.0/30 [110/30] via 10.0.0.26, eth2, 00:12:24
C>* 10.0.0.4/30 [110/20] via 10.0.0.26, eth2, 00:12:24
O>* 10.0.0.8/30 [110/20] via 10.0.0.26, eth2, 00:12:24
O>* 10.0.0.12/30 [110/20] via 10.0.0.17, eth0, 00:12:24
    * via 10.0.0.26, eth2, 00:12:24
C> 10.0.0.16/30 [110/10] is directly connected, eth0, 00:14:27
C>* 10.0.0.16/30 is directly connected, eth0
O>* 10.0.0.20/30 [110/30] via 10.0.0.26, eth2, 00:12:24
O> 10.0.0.24/30 [110/10] is directly connected, eth2, 00:12:28
C>* 10.0.0.24/30 is directly connected, eth2
O>* 10.1.0.0/24 [110/30] via 10.0.0.26, eth2, 00:12:24
O> 10.2.0.0/24 [110/10] is directly connected, eth1, 00:13:36
C>* 10.2.0.0/24 is directly connected, eth1
O>* 10.2.1.0/24 [110/20] via 10.0.0.17, eth0, 00:14:10
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-5#
```

- Router 4

```
QuaggaRouter-4# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route
C>* 10.0.0.0/30 [110/30] via 10.0.0.13, eth1, 00:16:20
O>* 10.0.0.4/30 [110/20] via 10.0.0.13, eth1, 00:16:20
O>* 10.0.0.8/30 [110/20] via 10.0.0.13, eth1, 00:16:20
O> 10.0.0.12/30 [110/10] is directly connected, eth1, 00:16:35
C>* 10.0.0.12/30 is directly connected, eth1
O> 10.0.0.16/30 [110/10] is directly connected, eth0, 00:18:44
C>* 10.0.0.16/30 is directly connected, eth0
O>* 10.0.0.20/30 [110/30] via 10.0.0.13, eth1, 00:16:20
O>* 10.0.0.24/30 [110/20] via 10.0.0.18, eth0, 00:12:39
    * via 10.0.0.13, eth1, 00:12:39
O>* 10.1.0.0/24 [110/30] via 10.0.0.13, eth1, 00:16:20
O>* 10.2.0.0/24 [110/20] via 10.0.0.18, eth0, 00:14:01
O> 10.2.1.0/24 [110/10] is directly connected, eth2, 00:18:04
C>* 10.2.1.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-4#
```

### Áreas 0,1 y 2

- Router 3

```
QuaggaRouter-3# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route
S>* 10.0.0.0/30 [110/20] via 10.0.0.5, eth2, 00:03:20
O> 10.0.0.4/30 [110/10] is directly connected, eth2, 01:37:28
C>* 10.0.0.4/30 is directly connected, eth2
S> 10.0.0.8/30 [110/10] is directly connected, eth0, 01:36:05
C>* 10.0.0.8/30 is directly connected, eth0
O> 10.0.0.12/30 [110/10] is directly connected, eth1, 01:34:30
C>* 10.0.0.12/30 is directly connected, eth1
O>* 10.0.0.16/30 [110/20] via 10.0.0.14, eth1, 00:03:25
    * via 10.0.0.25, eth3, 00:03:25
S>* 10.0.0.20/30 [110/20] via 10.0.0.10, eth0, 00:03:20
O> 10.0.0.24/30 [110/10] is directly connected, eth3, 01:33:48
C>* 10.0.0.24/30 is directly connected, eth3
S>* 10.1.0.0/24 [110/20] via 10.0.0.10, eth0, 00:03:20
O>* 10.2.0.0/24 [110/20] via 10.0.0.25, eth3, 00:03:30
O>* 10.2.1.0/24 [110/20] via 10.0.0.14, eth1, 00:03:30
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-3#
```

## Área 1

- Router 2

```
QuaggaRouter-2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 10.0.0.20/30 is directly connected, eth0
O 10.1.1.0/24 [110/10] is directly connected, eth1, 00:37:45
C>* 10.1.1.0/24 is directly connected, eth1
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-2#
```

- Router 1

```
QuaggaRouter-1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

O>* 10.0.0.0/30 [110/30] via 10.0.0.9, eth0, 00:01:02
O>* 10.0.0.4/30 [110/20] via 10.0.0.9, eth0, 00:01:02
O 10.0.0.8/30 [110/10] is directly connected, eth0, 00:01:19
C>* 10.0.0.8/30 is directly connected, eth0
O>* 10.0.0.12/30 [110/20] via 10.0.0.9, eth0, 00:01:02
O 10.0.0.20/30 [110/10] is directly connected, eth1, 00:01:15
C>* 10.0.0.20/30 is directly connected, eth1
O>* 10.0.0.24/30 [110/20] via 10.0.0.9, eth0, 00:01:02
O 10.1.0.0/24 [110/10] is directly connected, eth2, 00:00:39
C>* 10.1.0.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-1#
```

## 3º Configurar los Router-ID

```
QuaggaRouter-3# show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
RXmtL RqstL DBsmL
6.6.6.6 0 0 1 Full/Backup 32.611s 10.0.0.5 eth2:10.0.0.6
1.1.1.1 0 0 1 Full/DR 32.605s 10.0.0.10 eth0:10.0.0.9
4.4.4.4 0 0 1 Full/DR 32.623s 10.0.0.14 eth1:10.0.0.13
5.5.5.5 0 0 1 Full/DR 32.608s 10.0.0.25 eth3:10.0.0.26
QuaggaRouter-3#
```



## 4º Propagación de la ruta por defecto

```
QuaggaRouter-6(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.2
QuaggaRouter-6(config)# router ospf
ospf ospf6
QuaggaRouter-6(config)# router ospf
QuaggaRouter-6(config-router)# default-information originate
QuaggaRouter-6(config-router)# exit
QuaggaRouter-6(config)# exit
QuaggaRouter-6# write memory
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
```

## 5º Optimización de las tablas de enrutamiento

### Área 1

```
QuaggaRouter-1# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/11] via 10.0.0.9, eth0, 00:00:28
O 10.0.0.8/30 [110/10] is directly connected, eth0, 00:52:21
C>* 10.0.0.8/30 is directly connected, eth0
O 10.0.0.20/30 [110/10] is directly connected, eth1, 00:54:31
C>* 10.0.0.20/30 is directly connected, eth1
O 10.1.0.0/24 [110/10] is directly connected, eth2, 01:03:58
C>* 10.1.0.0/24 is directly connected, eth2
O>* 10.1.1.0/24 [110/20] via 10.0.0.22, eth1, 00:02:50
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-1#
```

### Área 2

- Antes

```
QuaggaRouter-5# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/10] via 10.0.0.26, eth2, 00:05:46
O>* 10.0.0.0/30 [110/30] via 10.0.0.26, eth2, 00:05:47
O>* 10.0.0.4/30 [110/20] via 10.0.0.26, eth2, 00:05:47
O>* 10.0.0.8/30 [110/20] via 10.0.0.26, eth2, 00:05:47
O>* 10.0.0.12/30 [110/20] via 10.0.0.17, eth0, 00:05:47
   * via 10.0.0.26, eth2, 00:05:47
O 10.0.0.16/30 [110/10] is directly connected, eth0, 00:07:27
C>* 10.0.0.16/30 is directly connected, eth0
O>* 10.0.0.20/30 [110/30] via 10.0.0.26, eth2, 00:05:47
O 10.0.0.24/30 [110/10] is directly connected, eth2, 00:35:46
C>* 10.0.0.24/30 is directly connected, eth2
O>* 10.1.0.0/24 [110/30] via 10.0.0.26, eth2, 00:05:47
O>* 10.1.1.0/24 [110/40] via 10.0.0.26, eth2, 00:05:47
O 10.2.0.0/24 [110/10] is directly connected, eth1, 01:00:59
C>* 10.2.0.0/24 is directly connected, eth1
O>* 10.2.1.0/24 [110/20] via 10.0.0.17, eth0, 00:07:27
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-5#
```

```

QuaggaRouter-4# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/10] via 10.0.0.13, eth1, 00:28:52
O>* 10.0.0.0/30 [110/30] via 10.0.0.13, eth1, 00:50:43
O>* 10.0.0.4/30 [110/20] via 10.0.0.13, eth1, 00:50:43
O>* 10.0.0.8/30 [110/20] via 10.0.0.13, eth1, 00:50:43
O 10.0.0.12/30 [110/10] is directly connected, eth1, 00:50:58
C>* 10.0.0.12/30 is directly connected, eth1
O 10.0.0.16/30 [110/10] is directly connected, eth0, 00:30:03
C>* 10.0.0.16/30 is directly connected, eth0
O>* 10.0.0.20/30 [110/30] via 10.0.0.13, eth1, 00:03:24
O>* 10.0.0.24/30 [110/20] via 10.0.0.13, eth1, 00:28:23
    * via 10.0.0.18, eth0, 00:28:23
O>* 10.1.0.0/24 [110/30] via 10.0.0.13, eth1, 00:03:24
O>* 10.1.1.0/24 [110/40] via 10.0.0.13, eth1, 00:03:24
O>* 10.2.0.0/24 [110/20] via 10.0.0.18, eth0, 00:30:03
O 10.2.1.0/24 [110/10] is directly connected, eth2, 01:19:42
C>* 10.2.1.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-4#

```

- Después

```

QuaggaRouter-4# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/11] via 10.0.0.13, eth1, 00:00:38
O 10.0.0.12/30 [110/10] is directly connected, eth1, 00:56:50
C>* 10.0.0.12/30 is directly connected, eth1
O 10.0.0.16/30 [110/10] is directly connected, eth0, 00:35:55
C>* 10.0.0.16/30 is directly connected, eth0
O>* 10.0.0.24/30 [110/20] via 10.0.0.13, eth1, 00:00:03
    * via 10.0.0.18, eth0, 00:00:03
O>* 10.2.0.0/24 [110/20] via 10.0.0.18, eth0, 00:35:55
O 10.2.1.0/24 [110/10] is directly connected, eth2, 01:25:34
C>* 10.2.1.0/24 is directly connected, eth2
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-4#

```

```

QuaggaRouter-5# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

O>* 0.0.0.0/0 [110/11] via 10.0.0.26, eth2, 00:00:09
O>* 10.0.0.12/30 [110/20] via 10.0.0.17, eth0, 00:00:09
    * via 10.0.0.26, eth2, 00:00:09
O 10.0.0.16/30 [110/10] is directly connected, eth0, 00:36:02
C>* 10.0.0.16/30 is directly connected, eth0
O 10.0.0.24/30 [110/10] is directly connected, eth2, 01:04:21
C>* 10.0.0.24/30 is directly connected, eth2
O 10.2.0.0/24 [110/10] is directly connected, eth1, 01:29:34
C>* 10.2.0.0/24 is directly connected, eth1
O>* 10.2.1.0/24 [110/20] via 10.0.0.17, eth0, 00:36:02
C>* 127.0.0.0/8 is directly connected, lo
QuaggaRouter-5#

```



## Paso 7. Manejo de los costes de enlace

Comprobación de la configuración de costes de enlace

```
QuaggaRouter-5# traceroute 1.1.1.1
traceroute to 1.1.1.1 (1.1.1.1), 30 hops max, 60 byte packets
 1  10.0.0.17 (10.0.0.17)  2.072 ms  1.421 ms  0.898 ms
 2  10.0.0.26 (10.0.0.26)  2.286 ms  3.834 ms  3.413 ms
 3  10.0.0.5 (10.0.0.5)    3.811 ms  5.555 ms  3.590 ms
 4  10.0.0.2 (10.0.0.2)    6.290 ms  6.478 ms  5.184 ms
 5  10.0.0.5 (10.0.0.5)    5.886 ms  4.510 ms  6.830 ms
```

## Referencias

- [https://www.ibm.com/support/knowledgecenter/es/ssw\\_ibm\\_i\\_71/rzajw/rzajwospf.htm](https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_71/rzajw/rzajwospf.htm)
- Enunciado de la práctica