

# Máster en Ingeniería Informática

---

## *Redes de Nueva Generación*

**Profesor:**

*Dr. Juan Carlos Fabero Jiménez (UCM)*



# Contenidos

---

- Tema 1: IP de nueva generación: IPv6
- Tema 2: Encaminamiento interno: OSPF
- Tema 3: Encaminamiento externo: BGPv4
- Tema 4: Encaminamiento troncal: MPLS
- Tema 5: Redes definidas por software: SDN

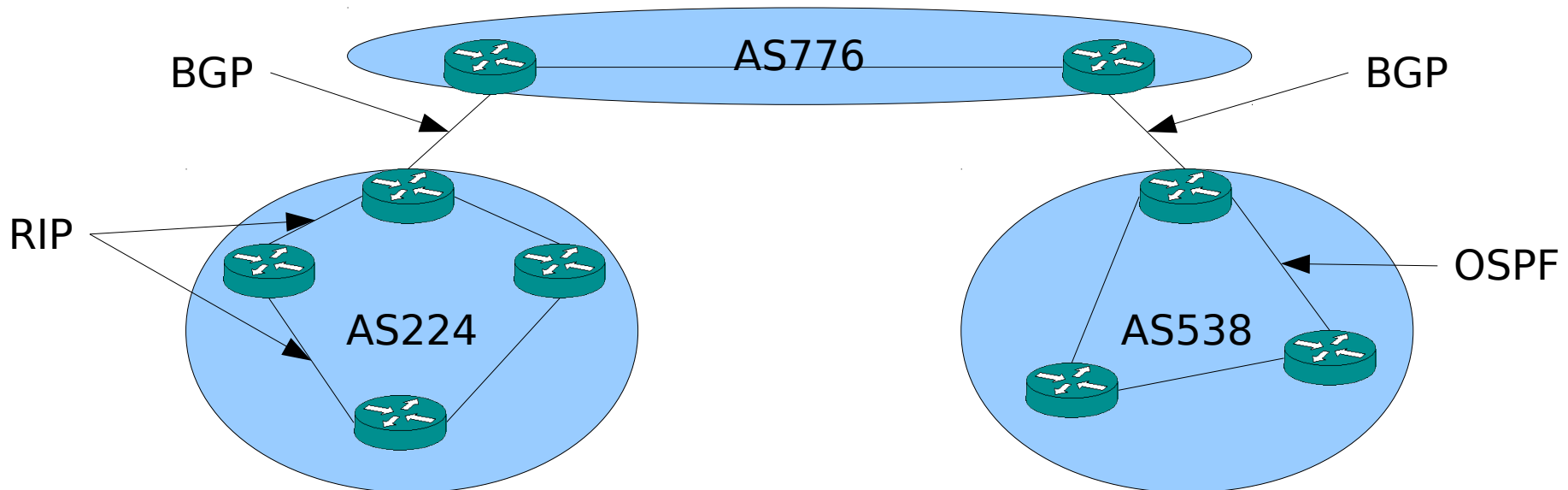
# Protocolos de Encaminamiento



*Introducción*

# Sistemas Autónomos

- **Conjunto conexo de redes IP y encaminadores bajo el control de una o varias organizaciones y con política de encaminamiento común (RFC1930)**
  - Utiliza un protocolo de pasarela interior (IGP, Internal Gateway Protocol). Generalmente OSPF o RIP.
  - Se comunica con otros AS mediante un protocolo de pasarela de frontera (BGP, Border Gateway Protocol)



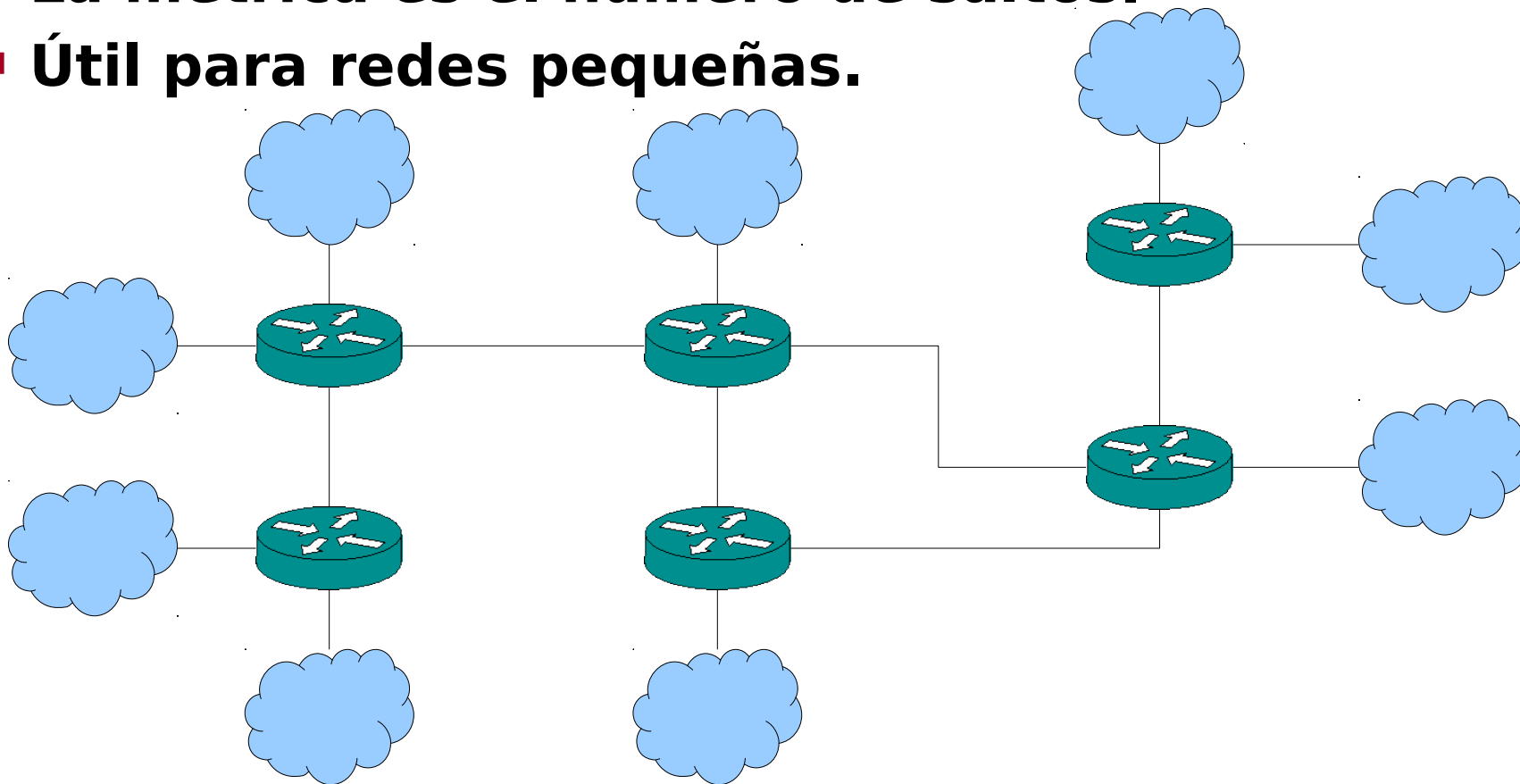
# Protocolos de Encaminamiento Interior



*RIP*

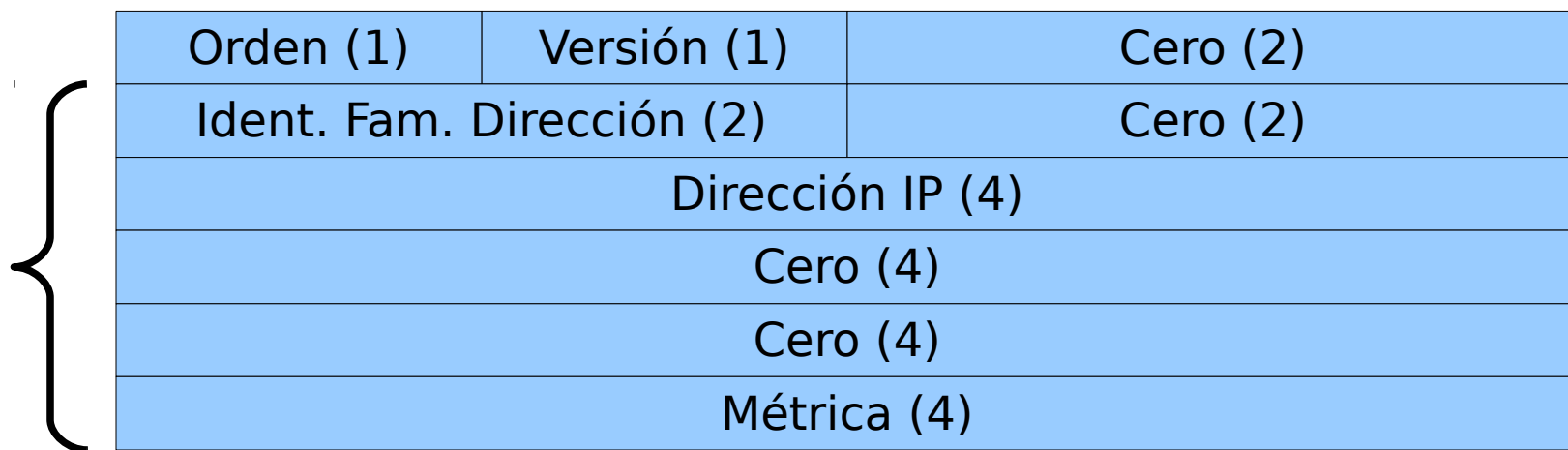
# RIP (Routing Internet Protocol)

- Definido en los RFC 1058 (v1) y 2453 (v2).
- Pertenece a la familia de protocolos de vector-distancia.
- La métrica es el número de saltos.
- Útil para redes pequeñas.



# RIPv1

## ■ Formato del mensaje de actualización



The diagram illustrates the structure of an RIPv1 update message. It consists of a table with six rows. The first two rows are grouped by a large curly brace on the left. The first row contains three fields: 'Orden (1)', 'Versión (1)', and 'Cero (2)'. The second row contains two fields: 'Ident. Fam. Dirección (2)' and 'Cero (2)'. The remaining four rows are single-field entries: 'Dirección IP (4)', 'Cero (4)', 'Cero (4)', and 'Métrica (4)'.

Orden (1)	Versión (1)	Cero (2)
Ident. Fam. Dirección (2)	Cero (2)	
Dirección IP (4)		
Cero (4)		
Cero (4)		
Métrica (4)		

## ■ Observaciones

- Las 5 últimas líneas (entrada RIP) se repiten para cada destino.
- El número máximo de saltos es 16 (inalcanzable).
- Orden:
  - Request (1)
  - Response (2)

# RIPv2

## ■ Formato del mensaje de actualización

Orden (1)	Versión (1)	Cero (2)
Ident. Fam. Dirección (2)		Marca de ruta (2)
Dirección IP (4)		
Máscara de red (4)		
Siguiendo salto (4)		
Métrica (4)		

## ■ Observaciones

- Marca de ruta (route tag): distingue entre rutas RIP y externas.
- El número máximo de saltos es 16 (inalcanzable).
- Si AFI==0xFFFF, significa que se utiliza autenticación.



# Protocolos de Encaminamiento Interior



*Encaminamiento en Linux*

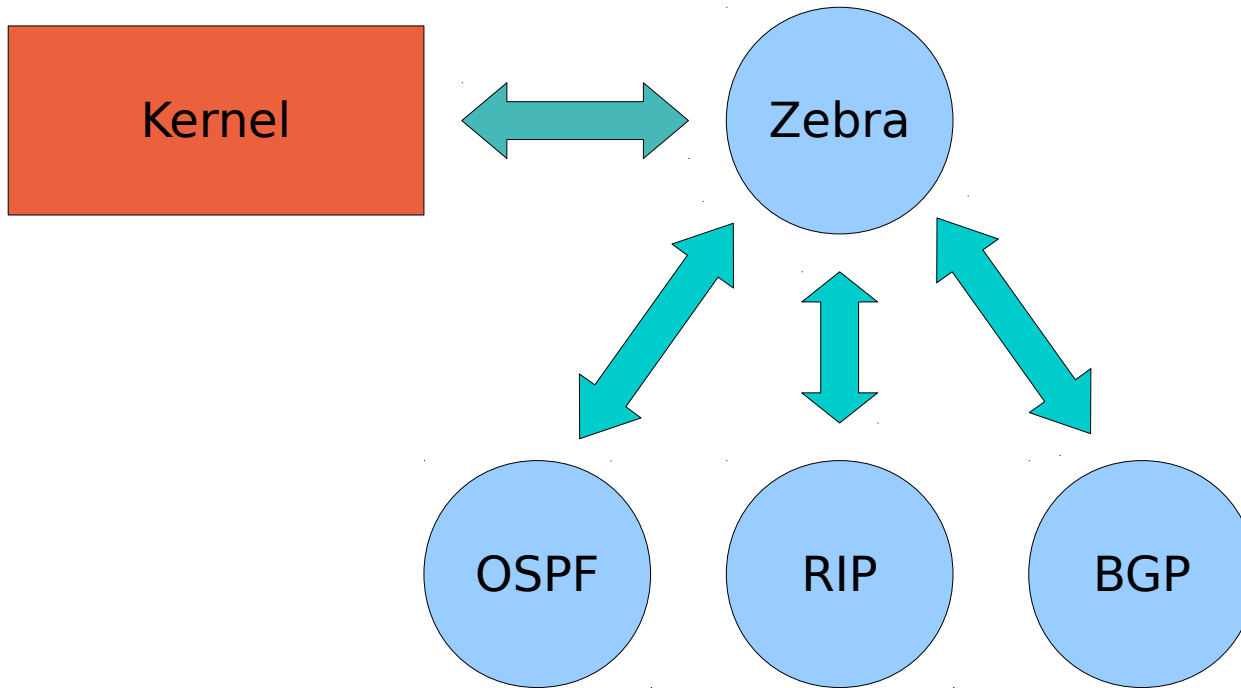
# Encaminamiento en GNU/Linux

---

## ■ Quagga (Zebra)

- Quagga es el sucesor (no oficial) de Zebra.
- Superdemonio genérico de encaminamiento.
- Soporta los siguientes protocolos:
  - RIP (versiones 1 y 2)
  - RIPng (RIP para IPv6)
  - OSPFv2
  - OSPFv3 (OSPF para IPv6)
  - BGPv4 y BGPv4+
  - IS-IS
  - Otros...

# Zebra



## ■ Definición de demonios

### ■ /etc/quagga/daemons

This file tells the quagga package which daemons to start.

```
#
# Entries are in the format: <daemon>=(yes|no|priority)
# 0, "no" = disabled
# 1, "yes" = highest priority
# 2 .. 10 = lower priorities
#
# When activation a daemon at the first time, a config file, even if it is
# empty, has to be present *and* be owned by the user and group "quagga", else
# the daemon will not be started by /etc/init.d/quagga. The permissions should
# be u=rw,g=r,o=.
# When using "vtysh" such a config file is also needed. It should be owned by
# group "quaggavty" and set to ug=rw,o= though.
#
zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=yes
ripngd=yes
isisd=no
```

# Zebra

## ■ Configuración de Zebra

- /etc/quagga/zebra.conf

```
!! Zebra configuration saved from vty
! 2005/11/28 14:13:23
!
hostname zebra
password zebra
enable password zebra
!
```

- Conectar mediante vtysh

```
root@uml1:/etc/quagga# vtysh
uml1# configure terminal
uml1(config)# ip forwarding
uml1(config)# ipv6 forwarding
uml1(config)# end
uml1# write
Configuration saved to /etc/quagga/zebra.conf
[OK]
uml1# exit
root@uml1:/etc/quagga#
```

# Zebra

## ■ Configuración de RIPd

- Si necesario, reiniciar quagga: `service quagga restart`
- Conectar con zebra mediante `vttysh`:

```
# vtysh
```

```
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
uml1# configure terminal
```

```
uml1(config)# router rip
```

```
uml1(config-router)# network eth0
```

```
uml1(config-router)# network eth1
```

```
uml1(config-router)# network eth2
```

```
uml1(config-router)# network eth3
```

```
uml1(config-router)# end
```

```
uml1# write
```

```
Building Configuration...
```

```
Configuration saved to /etc/quagga/zebra.conf
```

```
Configuration saved to /etc/quagga/ripd.conf
```

```
[OK]
```

```
uml1# quit
```

## ■ Configuración de RIPd

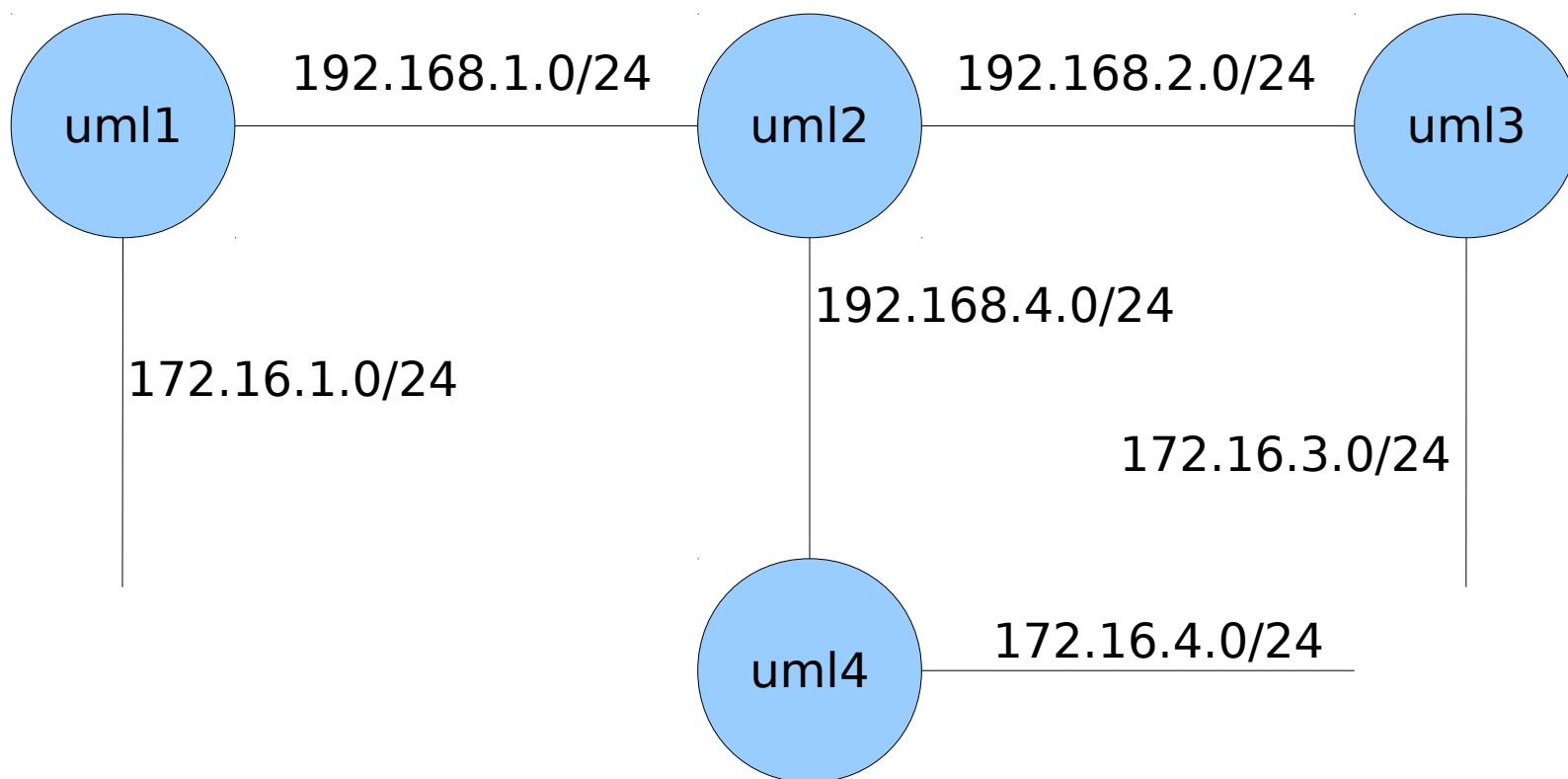
- Se ha creado el archivo /etc/quagga/ripd.conf:

```
!  
hostname ripd  
password zebra  
!  
!  
router rip  
 network eth0  
 network eth1  
 network eth2  
 network eth3  
!
```

# RIPv1

## ■ Ejercicio:

- Crear la siguiente configuración de red y utilizar RIPv1 en las máquinas uml.
- Comprobar las tablas de rutas y los mensajes intercambiados.





## ■ Configuración de RIPngd

- Activar el demonio ripngd en /etc/quagga/daemons
- Reiniciar quagga
- Conectar con zebra mediante **vttysh**:

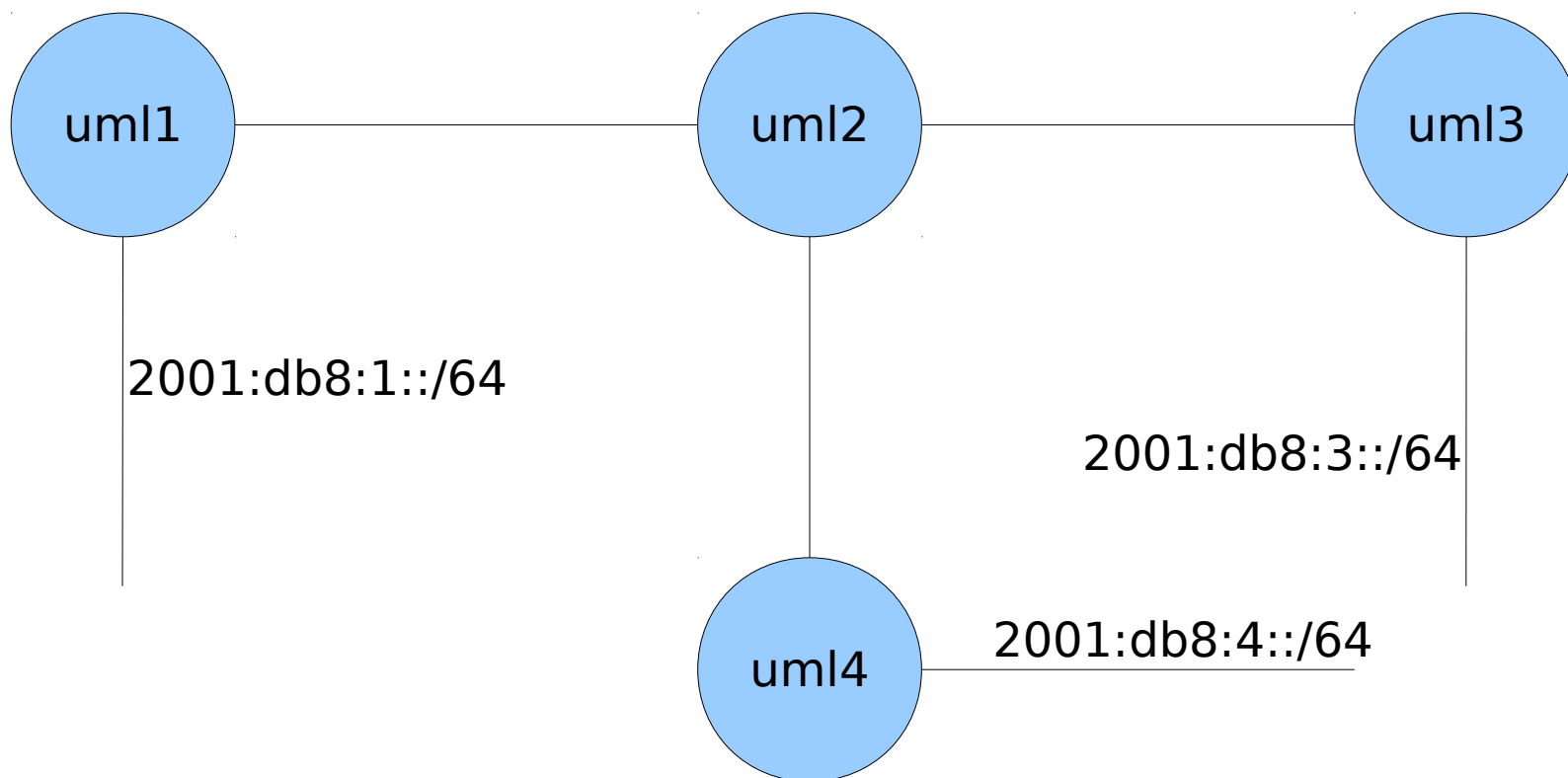
```
# vtysh
Copyright 1996-2005 Kunihiro Ishiguro, et al.

uml1# configure terminal
uml1(config)# router ripng
uml1(config-router)# network eth0
uml1(config-router)# network eth1
uml1(config-router)# network eth2
uml1(config-router)# network eth3
uml1(config-router)# end
uml1# write
Building Configuration...
Configuration saved to /etc/quagga/zebra.conf
Configuration saved to /etc/quagga/ripngd.conf
[OK]
uml1# quit
```

# RIPv1

## ■ Ejercicio:

- Crear la siguiente configuración de red y utilizar RIPv1 en las máquinas uml.
- Comprobar las tablas de rutas y los mensajes intercambiados.



# Protocolos de Encaminamiento Interior

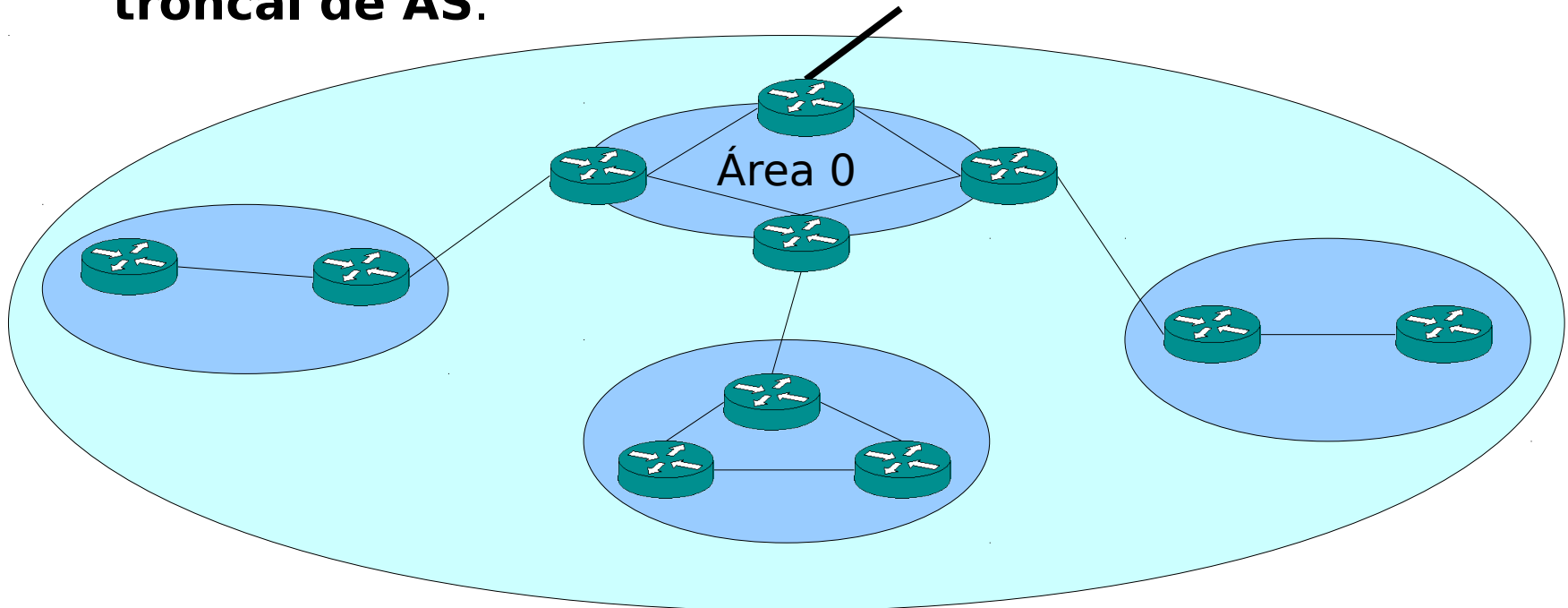


<i>OSPF</i>
-------------

# OSPF

## ■ Open Shortest Path First

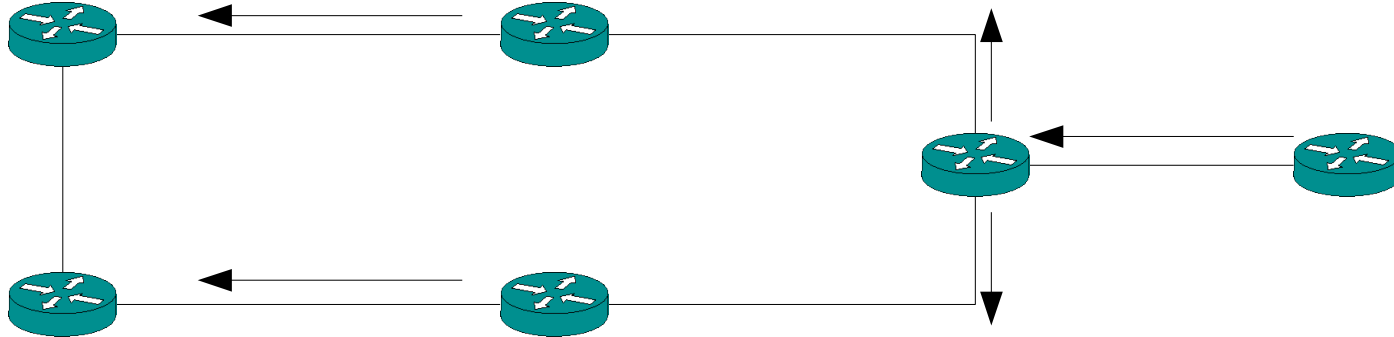
- Es el IGP más utilizado.
- Pertenece a la familia de encaminamiento por *estado del enlace*.
- Se puede dividir el SA en **áreas**.
- Cada área se conecta al área 0, que recibe el nombre de **troncal de AS**.



# OSPF

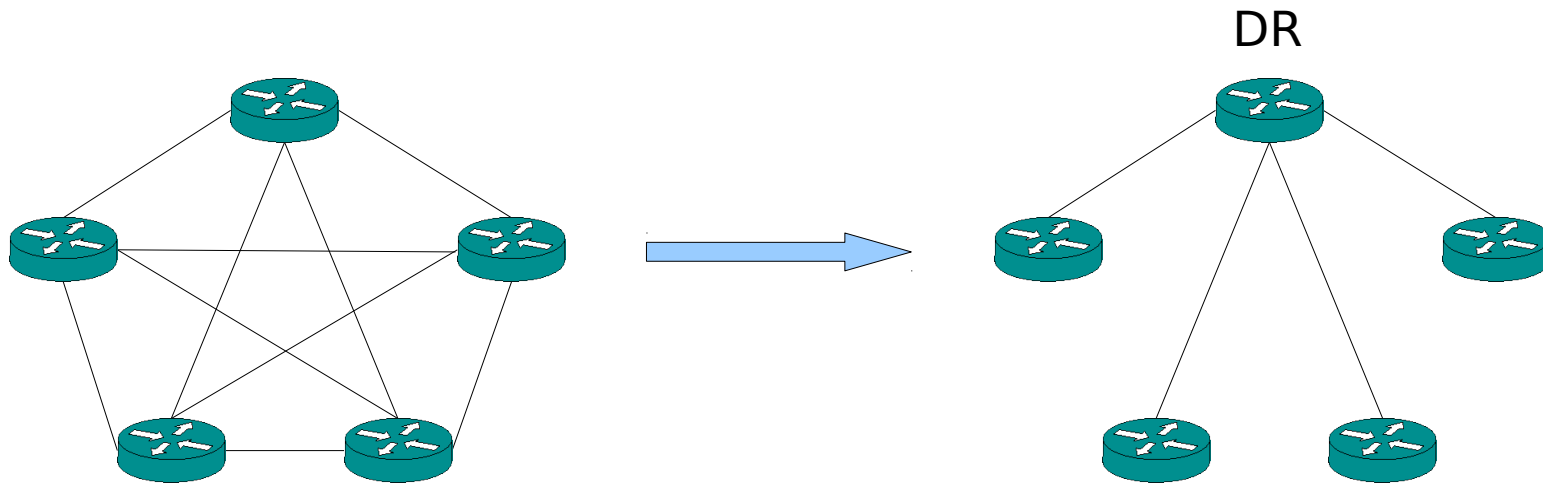
## ■ El problema de la inundación

- SPF necesita información topológica del estado de todos los enlaces.
- La información se recaba en cada encaminador y se difunde por la red.
- Es necesario acotar el proceso de inundación.
- Mediante el router-id y la edad del anuncio.



## ■ Encaminador Designado

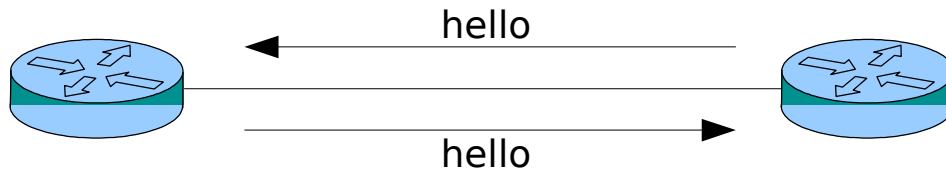
- Para disminuir el número de mensajes intercambiados **en una red de difusión**, uno de los encaminadores es elegido como encaminador designado (*DR, Designated Router*)
- Para aumentar la fiabilidad, se designa a otro encaminador como copia de seguridad (*BDR, Backup Designated Router*).
- Los demás encaminadores crean adyacencias sólo con el DR y el BDR.



# OSPF

## ■ Funcionamiento

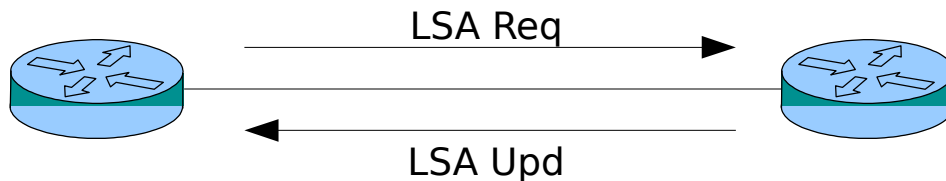
- Cada encaminador descubre a sus vecinos (hello)



- Se envía una descripción de la base de datos (DB Description)



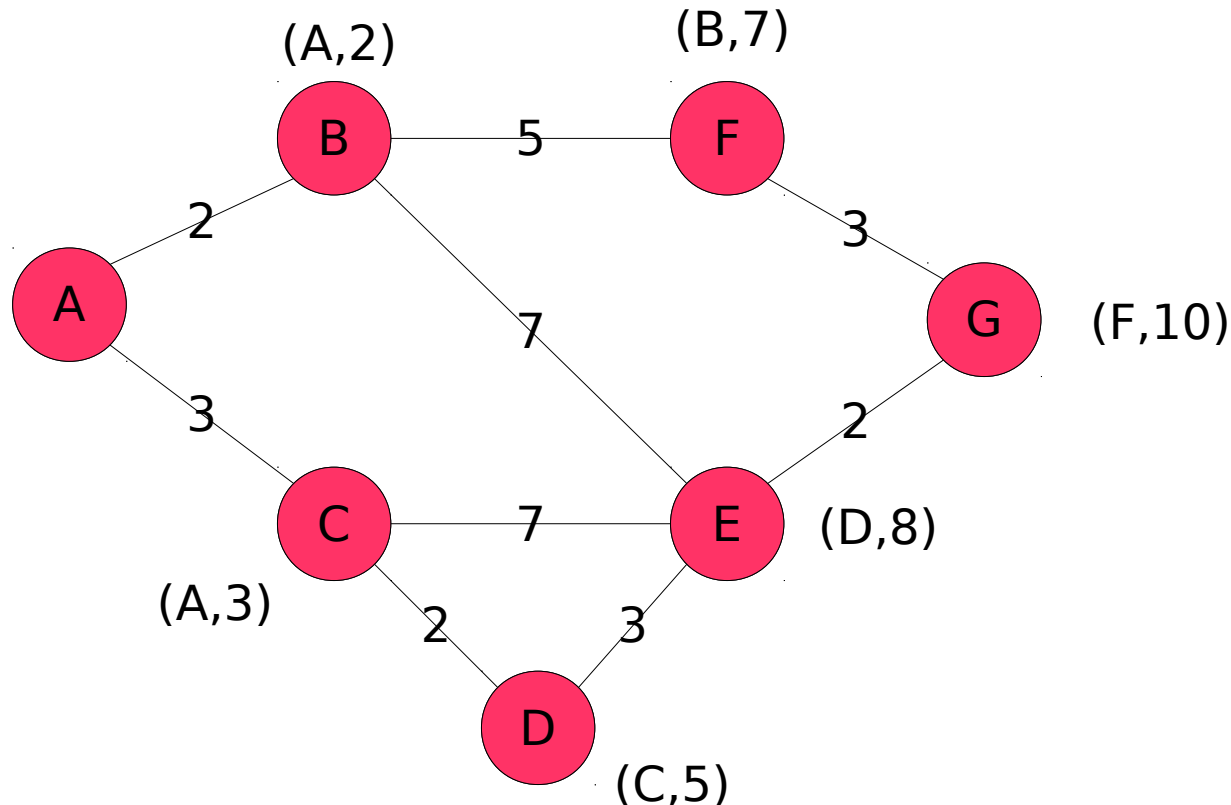
- Se envía el estado de los enlaces (LSA, Link State Advertisement)



# OSPF

## ■ Cálculo de rutas (SPF)

- Una vez que se han intercambiado los estados del enlace, cada encaminador realiza el cálculo de rutas.
- Ejemplo:





# Protocolos de Encaminamiento Interior



*OSPFv2*

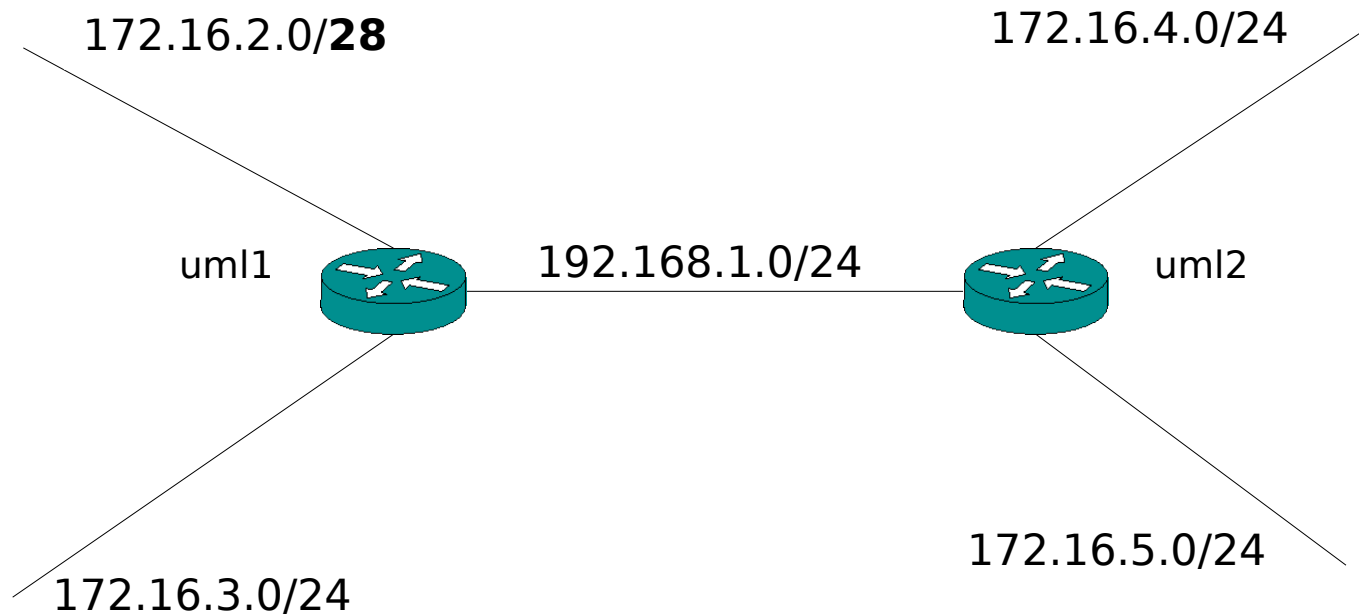
## ■ Características

- Definido en el RFC 2328 (abril de 1998).
- Emplea un modelo jerárquico de 2 niveles: todas las áreas deben conectarse directamente al área 0 (troncal).
- El cálculo del árbol SPF (*Shortest Path First*) se realiza de manera independiente en cada área.
- Generalmente ofrece una mejor convergencia que los algoritmos de vector distancia (RIP).
- Baja utilización de la red en el estado estacionario (sólo mensajes *hello* periódicos)
- No soporta IPv6.

# OSPFv2

## ■ Ejemplo:

- Crear la siguiente topología

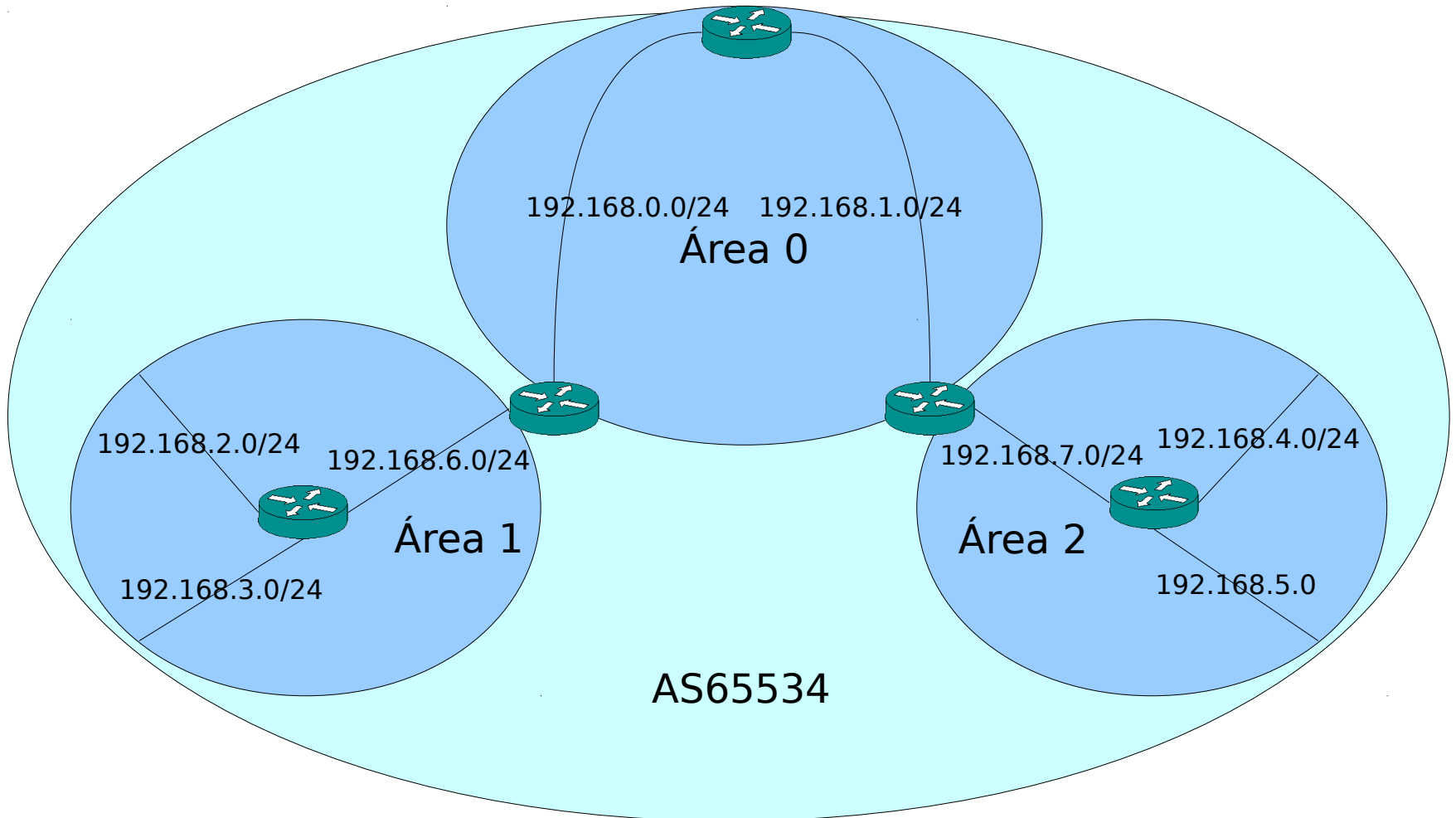


```
router ospf
router-id 0.0.0.1
network 192.168.1.0/24 area 0
network 192.168.2.0/28 area 0
network 192.168.3.0/24 area 0
passive-interface eth1
passive-interface eth2
```

```
router ospf
router-id 0.0.0.2
network 192.168.1.0/24 area 0
network 192.168.4.0/24 area 0
network 192.168.5.0/24 area 0
passive-interface eth1
passive-interface eth2
```

# OSPFv2

## ■ Ejemplo:



# Protocolos de Encaminamiento Interior



*OSPF: definiciones*

# OSPF: Definiciones

---

## ■ Áreas:

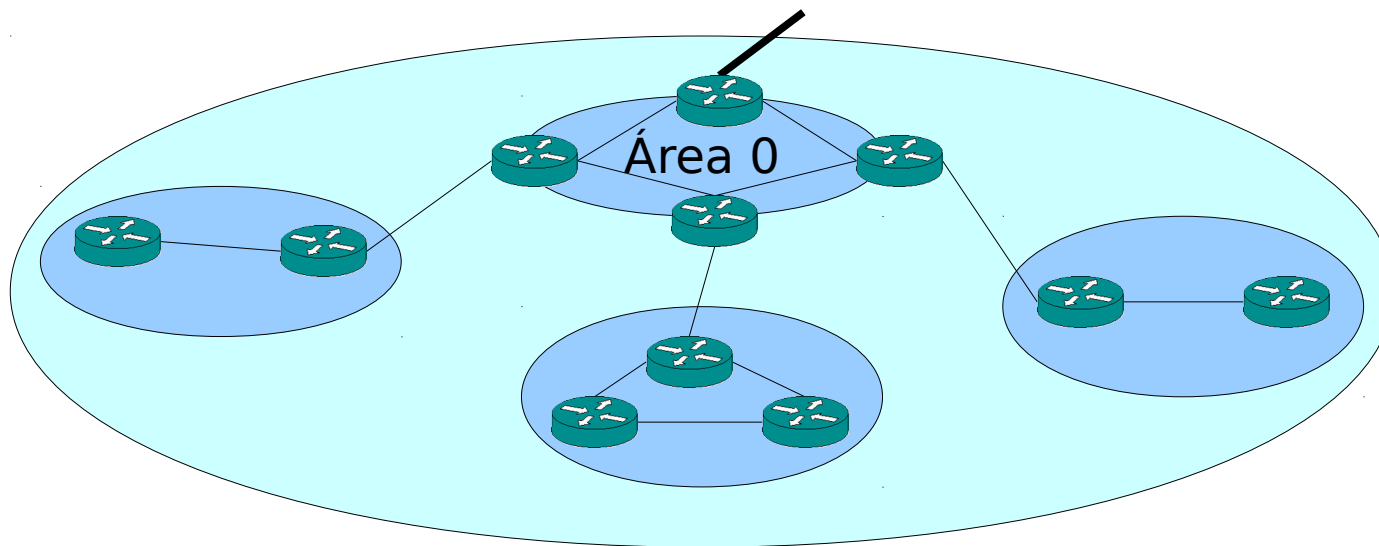
- En una red grande, el algoritmo SPF puede ser muy costoso computacionalmente.
  - OSPF permite dividir la red en áreas.
- Un área:
  - Es un conjunto lógico de redes y encaminadores.
  - Puede coincidir con fronteras administrativas o topológicas.
  - Cada área se distingue por un número de 32 bits.
- Ventajas:
  - Dentro del área, cada encaminador mantiene la misma información topológica del área. No conocen las topologías fuera del área. Sólo conocen rutas a los destinos externos al área.
    - Menor información topológica en cada dispositivo.
  - La mayor parte de los LSA se distribuyen sólo dentro del área.
    - Se limita el número de mensajes de actualización.
  - El algoritmo SPF se limita a la topología del área.
    - Menor coste computacional.

# OSPF: Definiciones

## ■ Áreas

### ■ Área troncal (*backbone*)

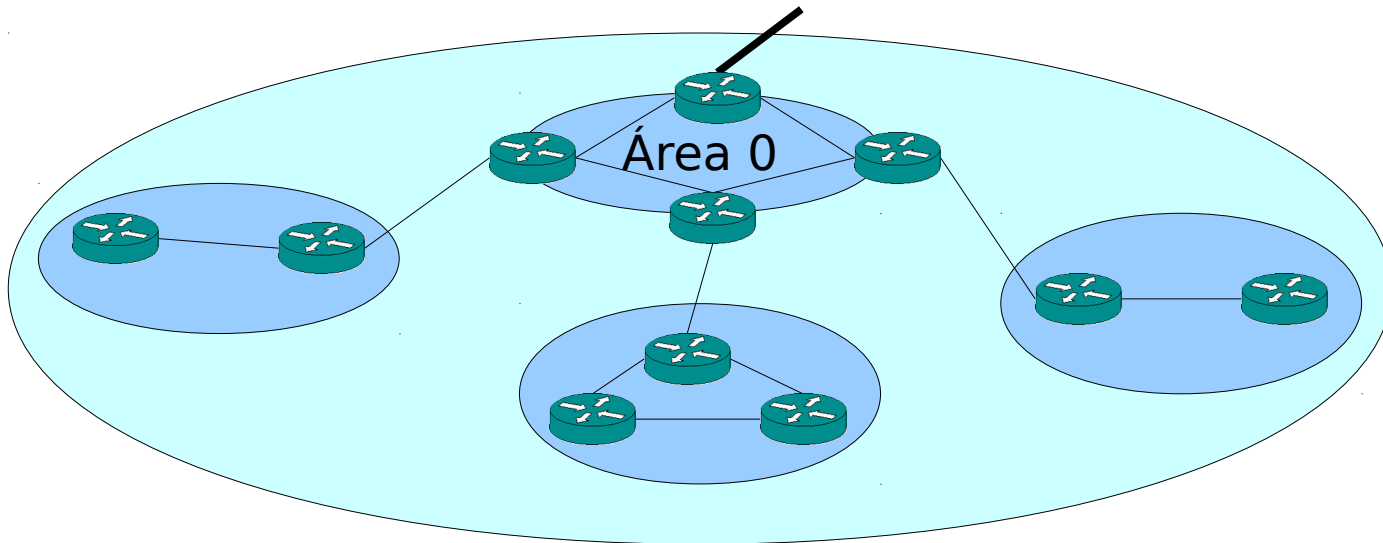
- Todas las redes OSPF contienen al menos un área, que se identifica con el número 0.
- Recibe el nombre de troncal.
- Cualquier otra área debe estar conectada físicamente con la troncal.
- Otras áreas inyectan información de encaminamiento en la troncal, que a su vez la distribuye a otras áreas.



# OSPF: Definiciones

## ■ Tipos de Encaminadores

- Internos: pertenecen sólo a un área.
  - Sólo mantienen información topológica de su área.
- Troncales: en el área troncal.
- Frontera de área (ABR): interconectan un área con la troncal.
  - Mantienen bases de datos de topología separadas para cada área.
- Frontera de AS (ASBR): intercambian información de encaminamiento entre la red OSPF y otros algoritmos.

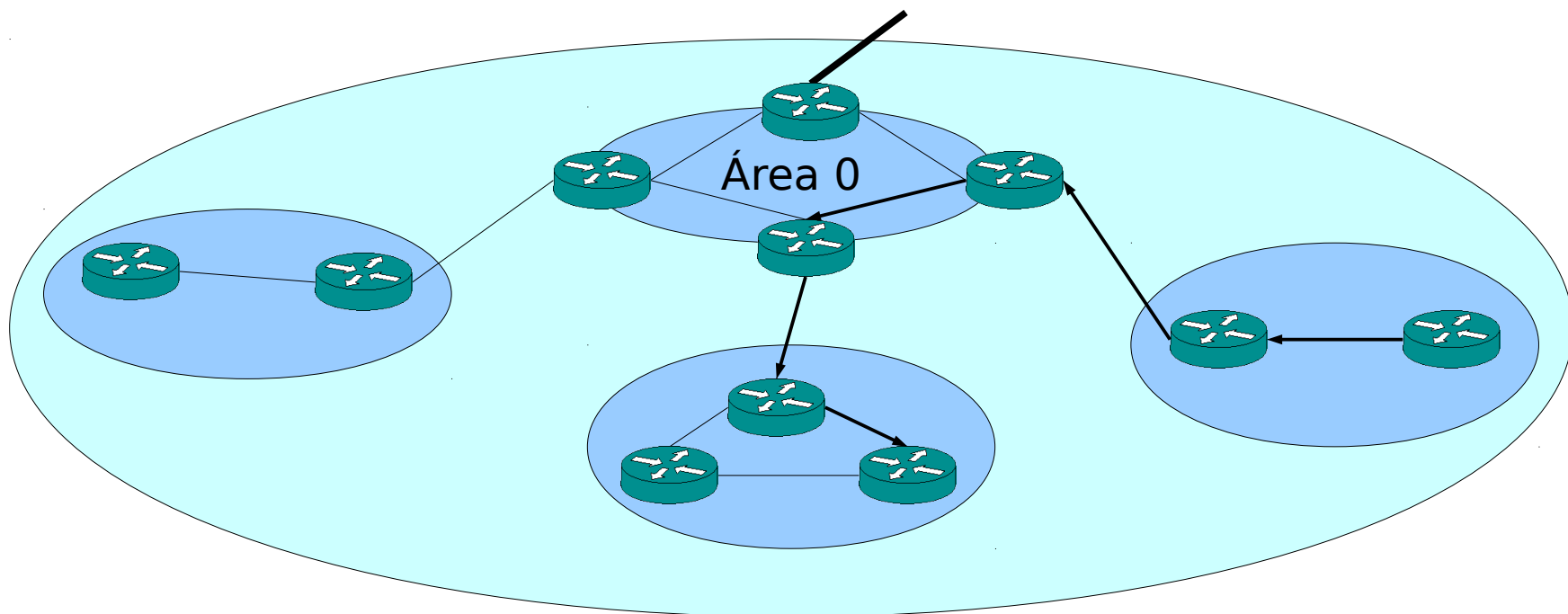




# OSPF: Definiciones

## ■ Encaminamiento

- Cada datagrama se encamina hacia la frontera de área.
- Puesto que los encaminadores frontera de área pertenecen a la troncal, se encamina el datagrama hacia la frontera del área de destino.
- Se encamina el datagrama hacia el destino final.



# OSPF: Definiciones

---

## ■ Base de datos de estado del enlace

- Contiene la información topológica sobre el área.
  - Dispositivos.
  - Enlaces físicos.
- Contiene rutas hacia destinos exteriores al área.

## ■ Anuncio de estado del enlace (*Link State Advertisement, LSA*)

- El contenido de un LSA describe un componente de la red (encaminador, enlace o ruta externa).
- Se intercambian LSA entre encaminadores adyacentes.
  - Con el fin de actualizar la base de datos de estado del enlace.
- Cuando un encaminador genera o modifica un LSA, debe anunciar el cambio por toda la red (área).
- Cuando se recibe un LSA, se actualiza la información topológica y se reenvía el LSA a cada encaminador adyacente (inundación controlada y fiable).

# OSPF: Definiciones

---

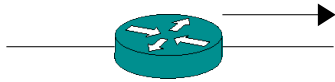
## ■ Información en los LSA

- Encaminador (*router*) (tipo 1): Describe el estado de los interfaces (enlaces) de cada encaminador. Se difunde por toda el área.
- Red (*network*) (tipo 2): Contiene los encaminadores conectados a una red de difusión. Se genera por el DR. Se difunde por toda el área.
- Resumen (*summary*) (tipos 3 y 4): Se genera por un ABR. La información se anuncia en la troncal, y luego se inyecta en las otras áreas.
  - Tipo 3: describe rutas a destinos que pertenecen a otras áreas (destinos inter-áreas).
  - Tipo 4: describe rutas a encaminadores ASBR.
- Externos (*external AS*) (tipos 5 y 7): describe rutas a destinos fuera de la red OSPF. Son generados por los ASBR. Se difunden por todas las áreas de la red OSPF.

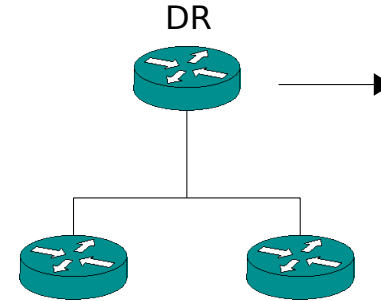
# OSPF: Definiciones

## ■ Tipos de LSA

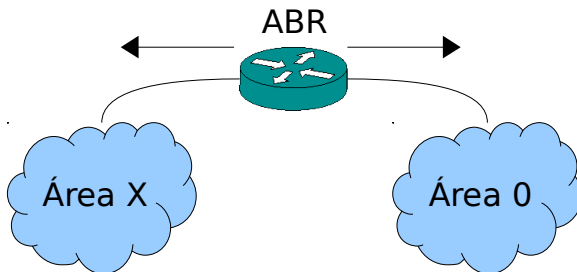
Router



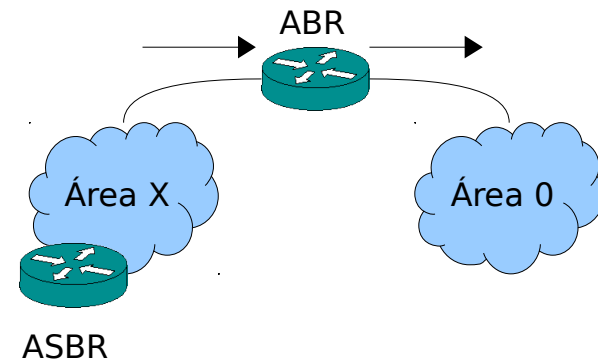
Network



Summary



External



# Protocolos de Encaminamiento Interior

---

*OSPF: Comunicación entre vecinos*

# OSPF: Comunicación entre vecinos

---

## ■ Proceso OSPF

- Todo encaminador OSPF debe realizar una serie de tareas:
  - Descubrimiento de vecinos.
  - Elección del encaminador designado DR (si procede).
  - Establecimiento de adyacencias.
    - Para el intercambio de información topológica.
  - Sincronización de la base de datos.
  - Cálculo de rutas mediante SPF.
- Para realizar estas tareas, los encaminadores intercambian entre sí distintos tipos de mensajes.

# OSPF: Comunicación entre vecinos

## ■ Todos los mensajes OSPF comparten la misma cabecera

- Version=2
- Packet Type
  - 1=Hello
  - 2=Database Description
  - 3=Link State Request
  - 4=Link State Update
  - 5=Link State Acknowledgment
- Authentication Type
  - 0=No Authentication
  - 1=Simple Password
- Authentication Data
  - Contraseña si Type=1

1	Version
1	Packet Type
2	Packet Length
4	Router ID
4	Area ID
2	Checksum
2	Authentication Type
8	Authentication Data

# OSPF: Descubrimiento de vecinos

---

## ■ Saludo (*hello*)

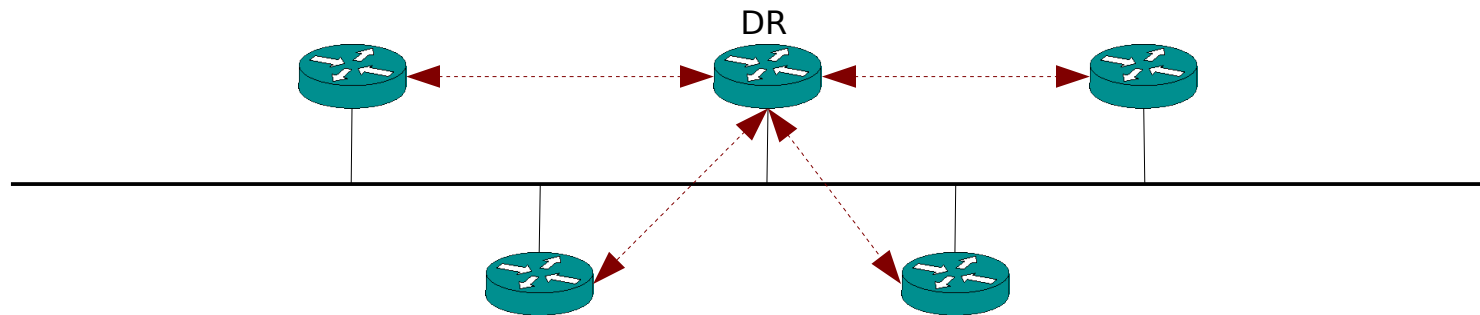
- Para descubrir las redes/encaminadores conectados directamente y el coste del enlace.
- Encaminadores conectados directamente establecen una relación de vecindad.
  - Cada mensaje *hello* porta los identificadores de los encaminadores (RID) que han sido descubiertos a través de ese enlace.
  - Cuando un encaminador recibe un *hello* con su propio RID, establece una relación de vecindad con el emisor.
- Para crear una relación de vecindad:
  - Ambos encaminadores deben pertenecer a la misma área.
  - Deben coincidir en el método de autenticación (si se ha definido).
  - Deben definir los mismos intervalos de *hello* y *dead*.
  - Ambos deben estar de acuerdo en la indicación de *stub* del área.
- Los mensajes *hello* se intercambian periódicamente entre vecinos.



# OSPF: Descubrimiento de vecinos

## ■ Elección del encaminador designado, DR

- En redes de acceso múltiple.
  - Para evitar crear adyacencias de todos con todos.
- Cada mensaje hello incluye la prioridad del encaminador, el identificador del DR (Designated Router) y el del BDR (Backup Designated Router).
- Para elegir el DR:
  - El encaminador con mayor prioridad.
  - Si igual, el encaminador con RID más alto.
- Una vez elegido el DR (y el BDR):
  - Establece adyacencias con cada encaminador del enlace.



# OSPF: Establecimiento de adyacencias

---

- **Dos encaminadores OSPF son adyacentes cuando han sincronizado su información topológica.**
  - No todo vecino es adyacente.
- **Dos pasos:**
  - Paso 1:
    - Intercambio de descripción de datos de enlace.
      - Lista de los LSA almacenados en la base local.
  - Paso 2:
    - Cada encaminador solicita de sus vecinos los LSA más recientes

# OSPF: Comunicación entre vecinos

---

- **Descripción de base de datos (*DBDesc, Data Base Description*)**
  - Contiene el conjunto de LSA contenidos en la base de datos local del encaminador.
  - Cada LSA lleva una marca de tiempo.
- **Solicitud de estado de los enlaces (*LSReq, Link State Request*)**
  - Solicita el envío del LSA a los encaminadores adyacentes.
- **Actualización del estado de los enlaces (*LSUpdate, Link State Update*)**
  - Contiene el LSA solicitado.
- **Confirmación del estado de los enlaces (*LSAck, Link State Acknowledge*)**
  - Confirmación del LSA recibido.

# Protocolos de Encaminamiento Interior



*OSPF: Máquina de estados*

# OSPF: Máquina de estados

---

## ■ Se definen una serie de estados para cada vecino y los eventos asociados.

- Down: Estado inicial. No se ha recibido información a través de ese enlace.
- Attempt: En redes de no difusión. El vecino parece inactivo. Se intenta restablecer la vecindad.
- Init: Se ha recibido un paquete *hello* pero el RID local no está listado en él.
- 2-way: Comunicación bidireccional, vecindad establecida.
- ExStart: los vecinos están comenzando a formar la adyacencia.
- Exchange: los dos vecinos están intercambiando sus bases de datos.
- Loading: los dos vecinos están sincronizando sus bases de datos.
- Full: los dos vecinos son adyacentes y sus bases de datos están sincronizadas

# Protocolos de Encaminamiento Interior

---

*OSPF: Redistribución de rutas*

# OSPF: Redistribución de rutas

---

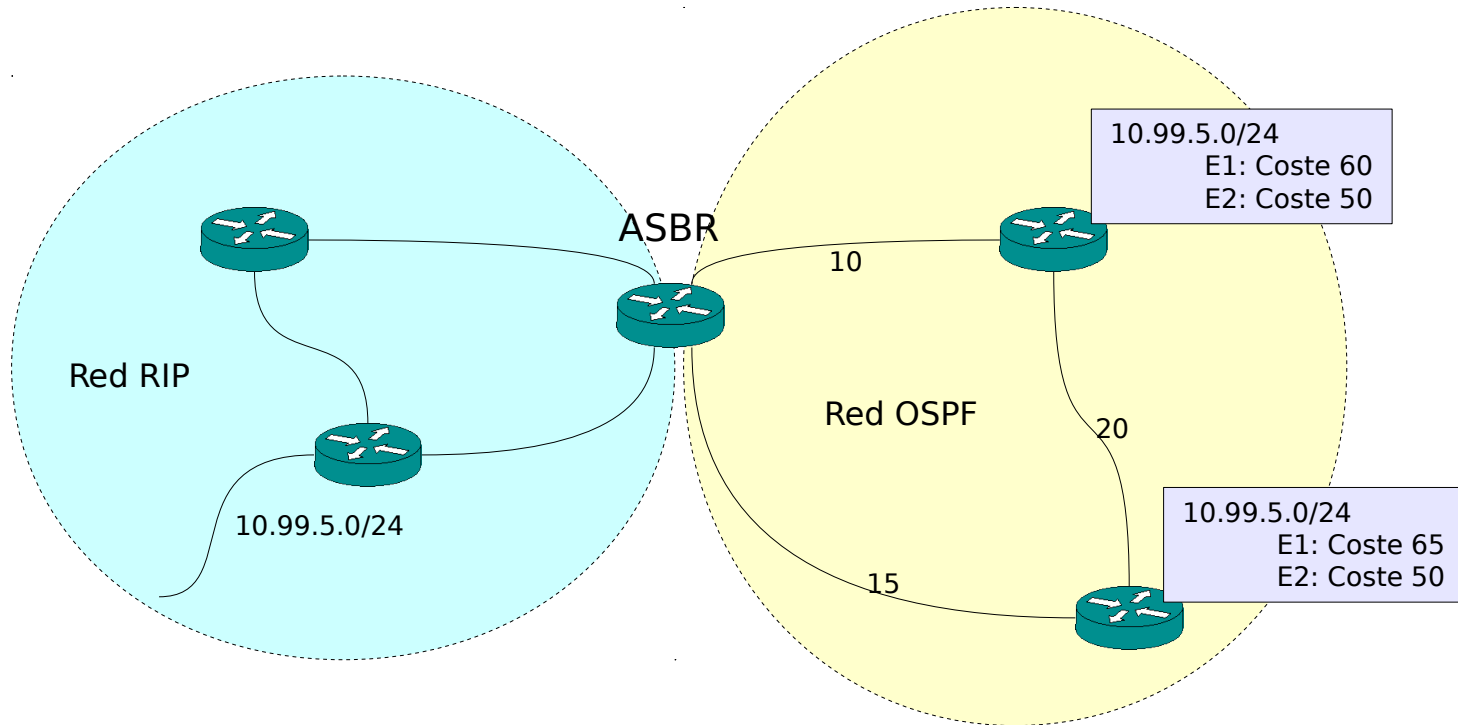
## ■ Definición

- Es el proceso de introducir rutas externas dentro de la red OSPF.
  - Las rutas pueden ser estáticas o aprendidas mediante otro protocolo de encaminamiento.
- Las rutas externas se anuncian en un ASBR.
- Dos partes:
  - Parte externa: representa la porción de la ruta exterior a la red OSPF. El ASBR asigna un coste externo a esta parte.
  - Parte interna: representa la porción de la ruta dentro de la red OSPF. Su coste se calcula según el algoritmo OSPF.
- Dos tipos de rutas externas:
  - Tipo External 1: el coste total es la suma del coste externo más el coste OSPF.
  - Tipo External 2: el coste total es siempre el coste externo. Se ignora el coste OSPF de alcanzar al ASBR.

# OSPF: Redistribución de rutas

## ■ Ejemplo

- El ASBR redistribuye la ruta hacia 10.99.5.0/24 en OSPF. La subred pertenece a la red RIP. El coste externo se ha configurado como 50.





# Protocolos de Encaminamiento Interior

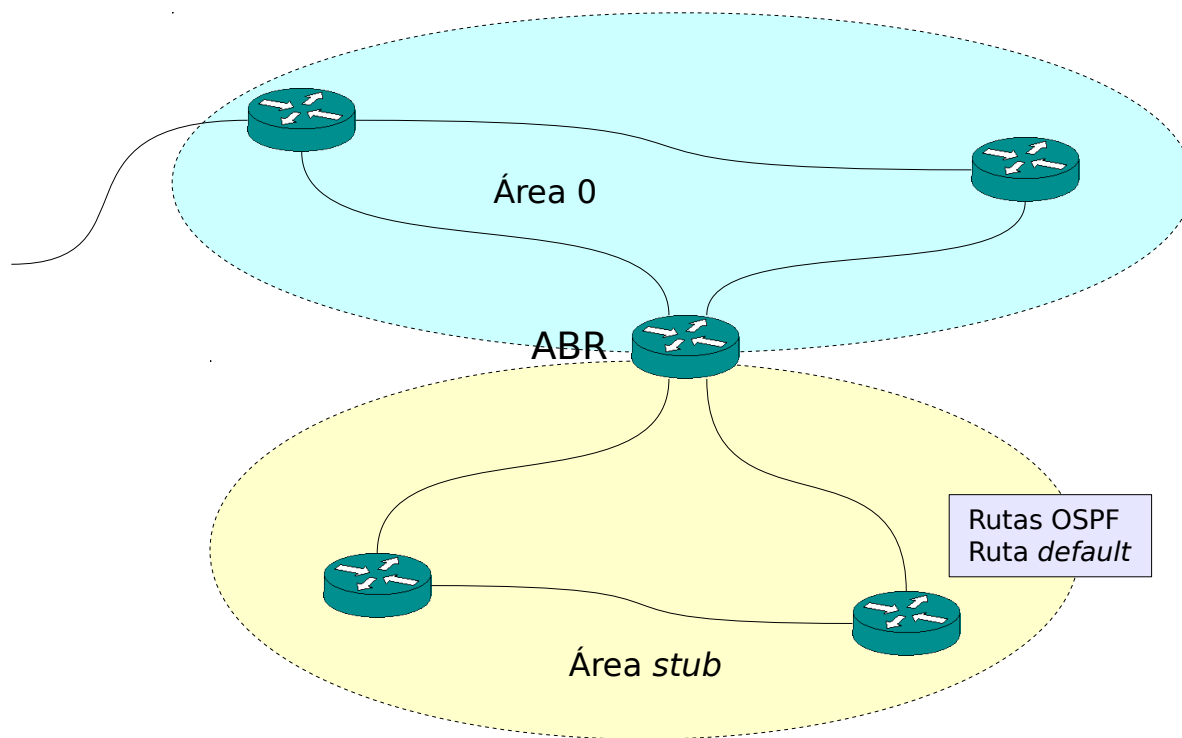


*OSPF: Áreas stub*

# OSPF: Áreas *stub*

## ■ Definición

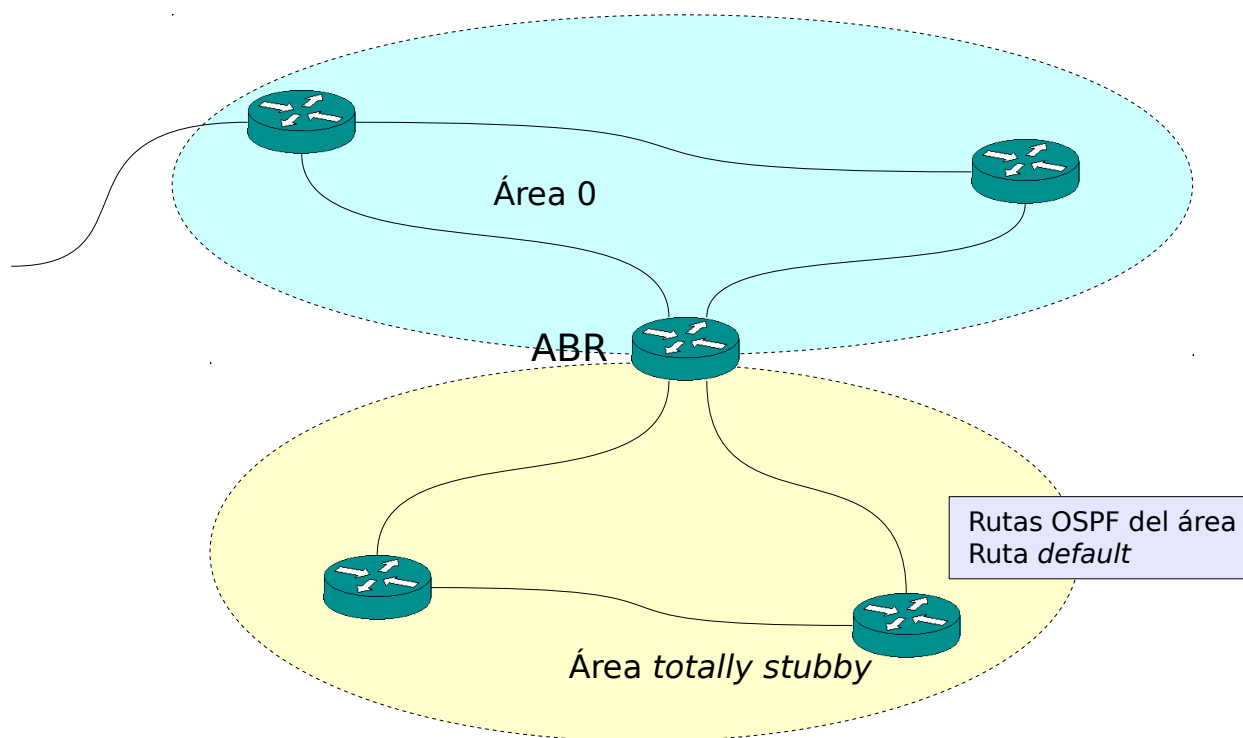
- Un área *stub* (terminal) es un área que no contiene información a rutas externas.
  - En lugar de ello, el ABR genera una ruta por defecto.
- Un área *stub* no puede contener un ASBR.



# OSPF: Áreas *totally stubby*

## ■ Definición

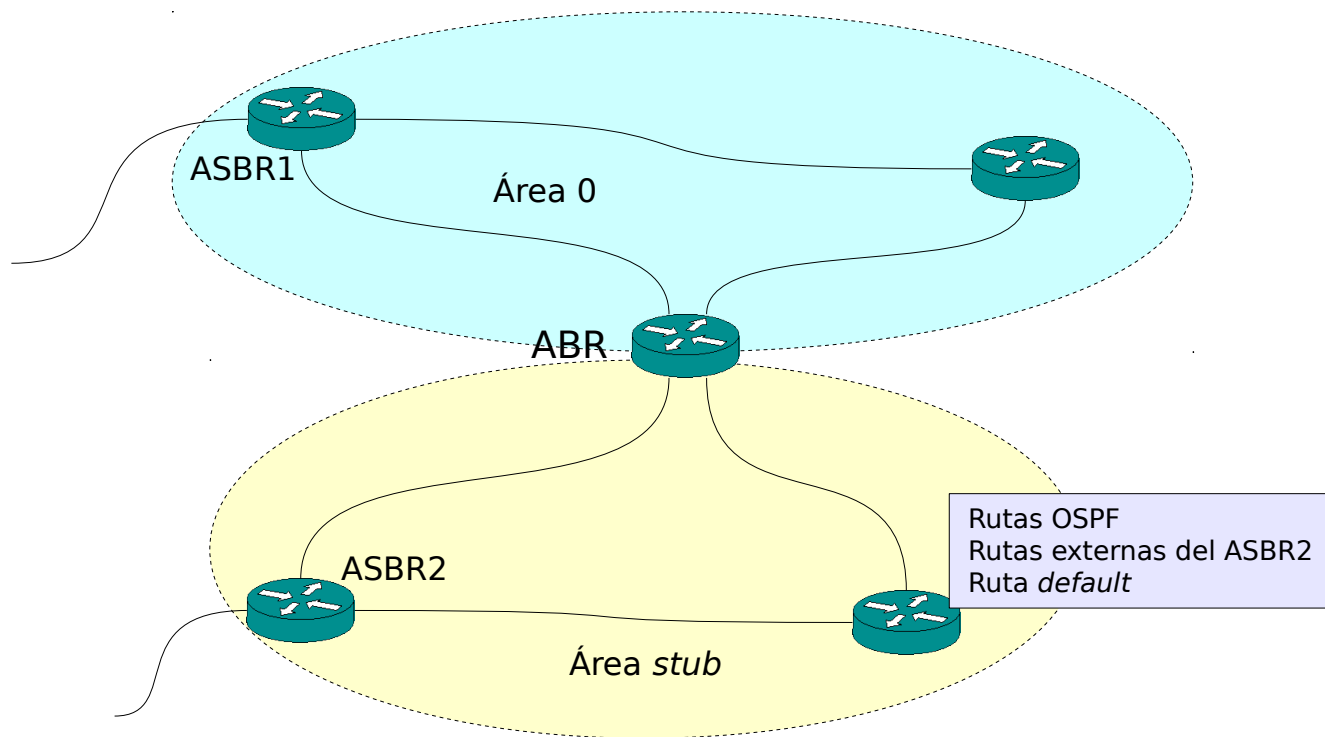
- Un área *totally stubby* (totalmente terminal) es un área que no contiene información a rutas fuera del área
  - En lugar de ello, el ABR genera una ruta por defecto.
- Un área *totally stubby* no puede contener un ASBR.



# OSPF: Áreas NSSA

## ■ Definición

- Un área *NSSA* (*not-so-stubby area*) es un área stub que contiene un ASBR.
- El ABR que la une con la troncal no inyecta rutas externas al área, pero propaga las generadas por el ASBR dentro de la NSSA en la troncal.



# Protocolos de Encaminamiento Interior

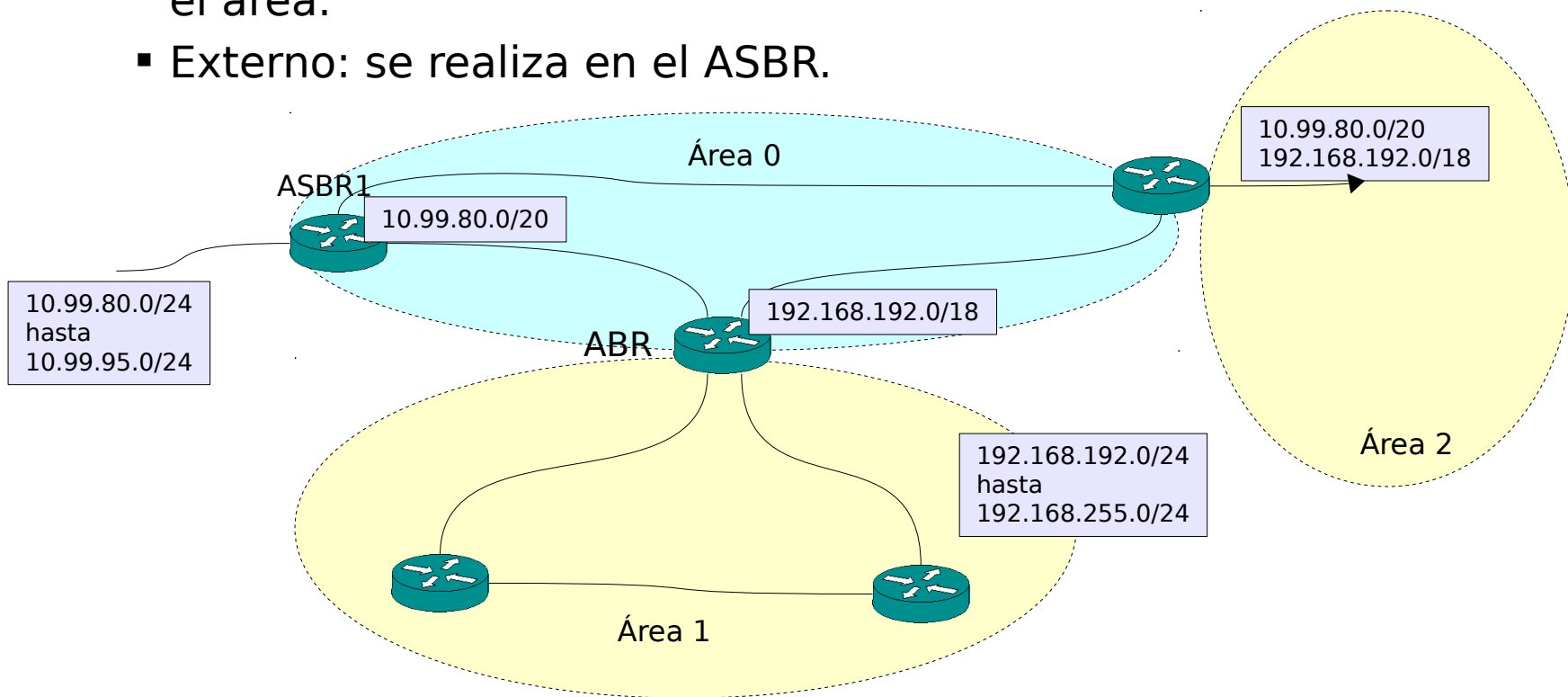


*OSPF: Agregado de rutas*

# OSPF: Agregado de rutas

## ■ Definición

- Es el proceso de resumir varias redes consecutivas en una sola.
- Dos tipos de agregados:
  - Inter-área: se realiza en el ABR para agregar los anuncios desde el área.
  - Externo: se realiza en el ASBR.



# Protocolos de Encaminamiento Interior

---

*OSPF: Configuración en GNU/Linux*

## ■ CLI (Command Line Interface)

- Su sintaxis es parecida a la de IOS de Cisco.
- Quagga abre un puerto TCP por cada protocolo activo, accesible normalmente desde localhost.
- Con telnet a dicho puerto entramos en modo cli y podemos ver o modificar la configuración y el estado del protocolo correspondiente.
- Tiene diversos niveles de acceso:
  - Sólo lectura: el modo de entrada. Sólo se puede ver la configuración y el estado actual, pero no modificar.
  - Modo enable: se accede mediante la orden enable (nos pedirá la contraseña especificada en el archivo de configuración). Se puede modificar la configuración en curso, escribirla en un archivo, restaurarla desde un archivo...



# OSPFv2

## ■ CLI (Command Line Interface)

### ■ Ejemplo de sesión

```
vmachine:/etc/quagga# telnet 127.0.0.1 ospfd
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.9).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
ospfd> show ip ospf route
===== OSPF network routing table =====
N   192.168.0.0/24      [10] area: 0.0.0.0
      directly attached to eth0
N   192.168.1.0/24      [20] area: 0.0.0.0
      via 192.168.0.1, eth0
N   192.168.2.0/24      [20] area: 0.0.0.1
      via 192.168.6.4, eth1
N   192.168.3.0/24      [20] area: 0.0.0.1
      via 192.168.6.4, eth1
N   192.168.6.0/24      [10] area: 0.0.0.1
      directly attached to eth1
```

# OSPFv2

## ■ CLI (Command Line Interface)

- Se puede acceder a la ayuda con `help`
- Si se pulsa `?` muestra las alternativas posibles para completar una orden:

```
ospfd> show ip ospf (se pulsa ?)
border-routers for this area
database      Database summary
interface     Interface information
neighbor      Neighbor list
route         OSPF routing table
<cr>
ospfd> show ip ospf
```

- **Ejercicio:** comprobar el estado del encaminador uml2

## ■ CLI (Command Line Interface)

- Cambio de configuración en curso
  - Activar el modo enable
  - Entrar en el modo de terminal de configuración:

```
ospfd# configure terminal
ospfd(config)# (se pulsa ?)
access-list  Add an access list entry
debug        Debugging functions (see also 'undebug')
end          End current mode and change to enable mode.
exit         Exit current mode and down to previous mode
help         Description of the interactive help system
interface    Select an interface to configure
ip           IP information
list         Print command list
log          Logging control
no           Negate a command or set its defaults
route-map    Create route-map or enter route-map command mode
router       Enable a routing process
service      Set up miscellaneous service
show         Show running system information
write        Write running configuration to memory, network, or terminal
```

## ■ CLI (Command Line Interface)

### ■ interface <iface>

- Se emplea para configurar parámetros de un interfaz:
  - **authentication**: activa la autenticación en este interfaz
  - **authentication-key**: la clave de autenticación
  - **cost**: El coste del interfaz (puede no ser simétrico)
  - **dead-interval**: periodo antes de declarar muerto a un vecino
  - **hello-interval**: periodo entre paquetes HELLO
  - **message-digest-key**: Clave de autenticación de huellas de mensajes
  - **mtu-ignore**: desactiva la detección de desacuerdo en la MTU
  - **network**: tipo de red (broadcast, NBMA, p-t-p...)
  - **priority**: prioridad del encaminador para ser elegido DR
  - **retransmit-interval**: periodo entre retransmisiones de avisos de pérdida de enlace

## ■ CLI (Command Line Interface)

### ■ router ospf

- **area**: parámetros de área OSPF (coste, autenticación, filtros de rutas, enlace virtual, área stub...)
- **auto-cost**: calcula el coste OSPF según el ancho de banda del enlace
- **compatible**: lista de compatibilidad OSPF (con RFC1583)
- **default-information**: control de la distribución de la información de encaminamiento predeterminada
- **default-metric**: métrica de las rutas redistribuidas
- **distance**: define una distancia adicional para rutas externas, inter-área o intra-área
- **distribute-list**: filtra redes en la actualización de rutas
- **end**: sale del modo actual y cambia a modo *enable*
- **log-adjacency-changes**: traza los cambios en estado de adyacencia
- **max-metric**: distancia máxima/infinita de OSPF
- **mpls-te**: configura parámetros MPLS-TE
- **neighbor**: especifica un encaminador vecino
- **network**: activa encaminamiento en una red IP
- **no**: desactiva una orden o la restaura a sus valores predeterminados

## ■ CLI (Command Line Interface)

- router ospf
  - **ospf**: órdenes específicas de ospf
  - **passive-interface**: suprime el envío de actualizaciones de ruta sobre un interfaz
  - **quit**: vuelve al modo cli anterior
  - **redistribute**: controla la redistribución de información que proviene de otros protocolos (sólo en ASBR)
  - **refresh**: ajusta parámetros de refresco
  - **router-id**: identificador del proceso ospf
  - **show**: muestra información del sistema actual
  - **timers**: ajusta intervalos de temporización de encaminamiento

## ■ Definición de demonios

### ■ /etc/quagga/daemons

This file tells the quagga package which daemons to start.

```
#  
# Entries are in the format: <daemon>=(yes|no|priority)  
# 0, "no" = disabled  
# 1, "yes" = highest priority  
# 2 .. 10 = lower priorities  
#  
# When activation a daemon at the first time, a config file, even if it is  
# empty, has to be present *and* be owned by the user and group "quagga", else  
# the daemon will not be started by /etc/init.d/quagga. The permissions should  
# be u=rw,g=r,o=.  
# When using "vtysh" such a config file is also needed. It should be owned by  
# group "quaggavty" and set to ug=rw,o= though.  
#  
zebra=yes  
bgpd=no  
ospfd=yes  
ospf6d=no  
ripd=no  
ripngd=no  
isisd=no
```

## ■ Configuración básica

- En el archivo `/etc/quagga/ospfd.conf` se configuran los distintos aspectos de OSPF.
- Los principales parámetros son:
  - **Network:** indica las redes involucradas en el algoritmo de manera que *ospfd* conozca los interfaces activos. Cada red se asocia con el área correspondiente.
  - **Router-id:** permite identificar de manera única a cada encaminador participante. Sigue la notación IP.
  - **Priority:** la prioridad del encaminador para ser elegido como encaminador designado (DR o BDR).
  - **Redistribute:** especifica qué rutas (de otros protocolos) deben inyectarse en OSPF (sólo en ASBR).
  - **area X.X.X.X range A.B.C.D/M:** agregado de rutas inter-área (sólo ABR).
  - **area X.X.X.X stub [no-summary]:** definición de un área *stub*.



# OSPFv2

## ■ Configuración básica

### ■ Ejemplo:

```
root@uml1:~# vtysh
```

```
Hello, this is Quagga (version 0.99.23.1).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
uml1# configure terminal
```

```
uml1(config)# router ospf
```

```
uml1(config-router)# router-id 0.0.0.1
```

```
uml1(config-router)# redistribute rip
```

```
uml1(config-router)# redistribute bgp
```

```
uml1(config-router)# network 192.168.0.0/24 area 0
```

```
uml1(config-router)# end
```

```
uml1# write
```

```
Building Configuration...
```

```
Configuration saved to /etc/quagga/zebra.conf
```

```
Configuration saved to /etc/quagga/ospfd.conf
```

```
[OK]
```

```
uml1# exit
```

```
root@uml1:~#
```

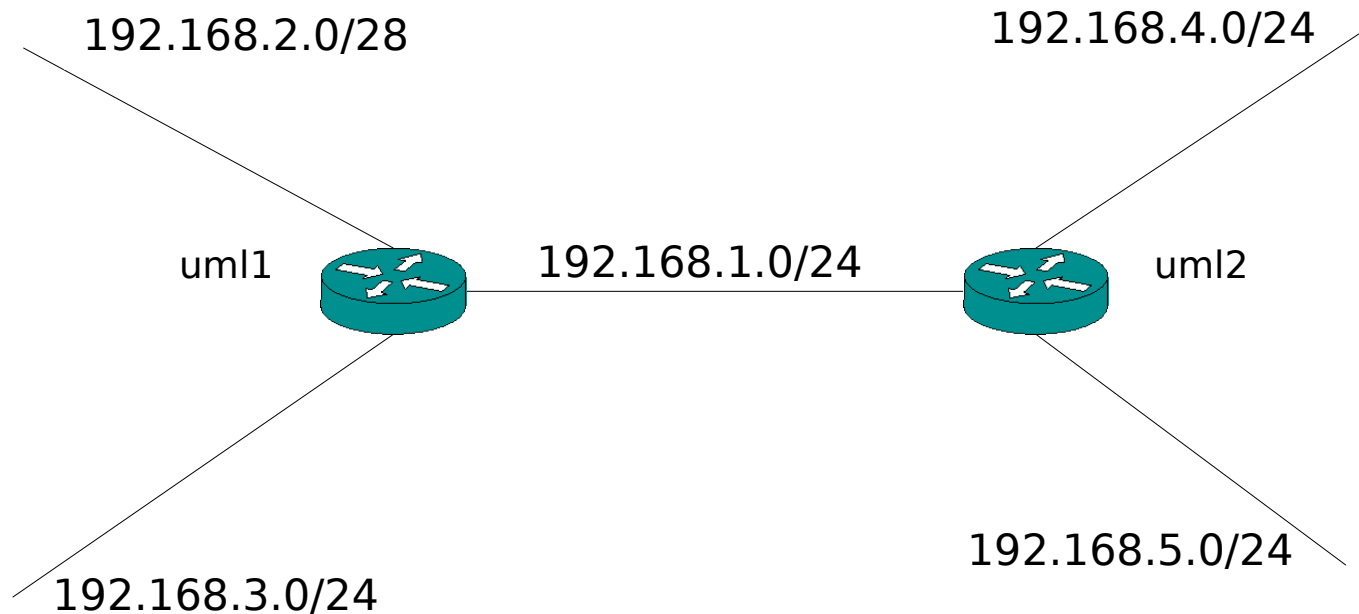
→ Sólo en ASBR

Sección de router

# OSPFv2

## ■ Ejemplo:

- Crear la siguiente topología



```
router ospf
router-id 0.0.0.1
network 192.168.1.0/24 area 0
network 192.168.2.0/28 area 0
network 192.168.3.0/24 area 0
passive-interface eth1
passive-interface eth2
```

```
router ospf
router-id 0.0.0.2
network 192.168.1.0/24 area 0
network 192.168.4.0/24 area 0
network 192.168.5.0/24 area 0
passive-interface eth1
passive-interface eth2
```

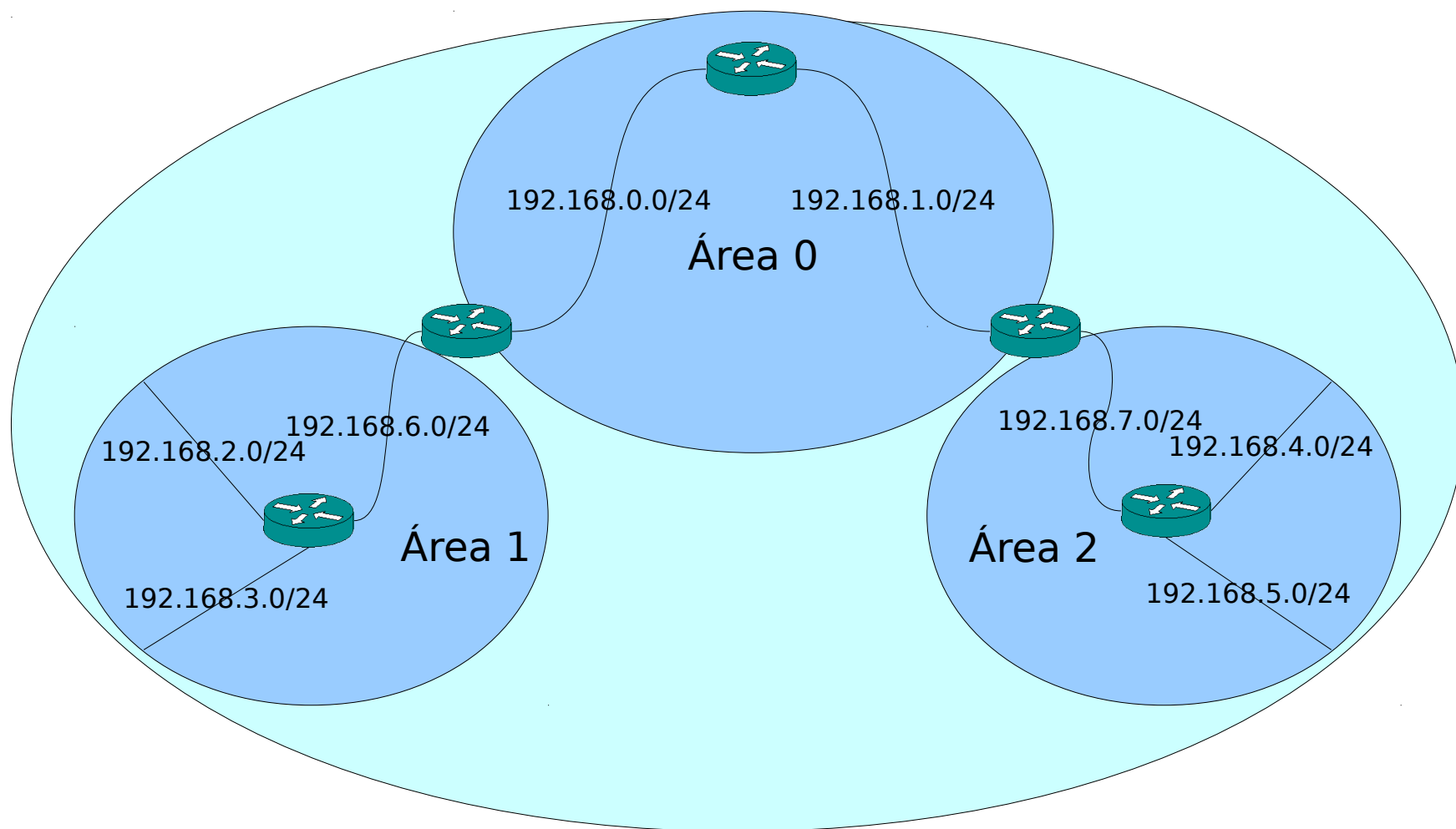
## ■ Áreas

- Para controlar el tamaño de las tablas de rutas se puede dividir el AS en áreas.
- Un área es un conjunto de redes y encaminadores agrupados con un identificador común.
- Un identificador de área es un número entero de 32 bits, en notación punto-decimal (como IP) o entero.
- El área 0 (equivalente a 0.0.0.0) recibe el nombre de troncal.
- Todas las áreas deben conectarse con la troncal.

# OSPFv2

## ■ Ejercicio:

- Configurar 5 máquinas virtuales para crear el siguiente AS



# OSPFv2

## ■ Configuración básica con áreas

### ■ Ejemplo:

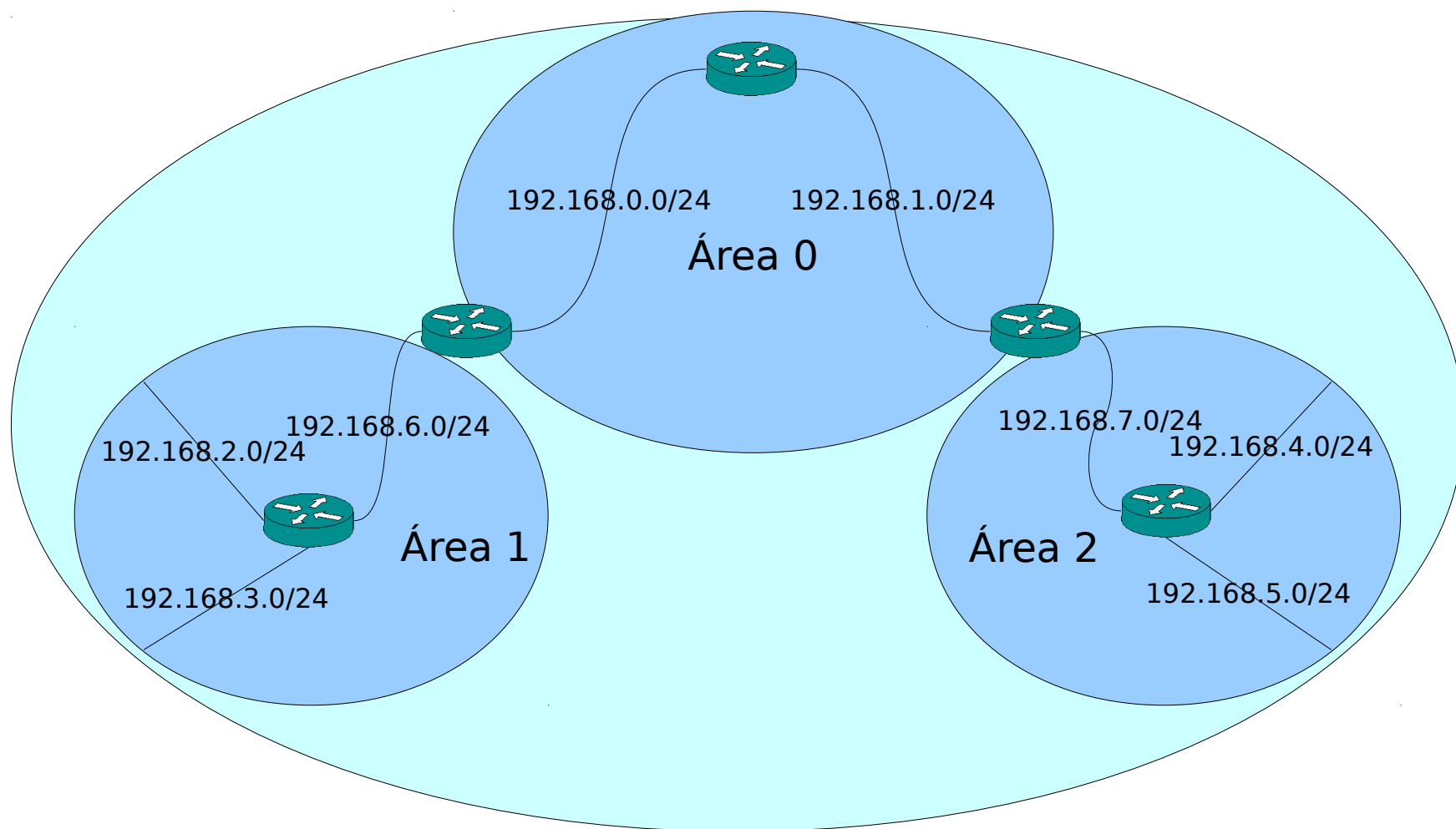
```
! *- ospf *-  
!  
! OSPFd configuration file  
!  
!  
hostname ospfd  
password zebra  
!enable password please-set-at-here  
!  
router ospf  
  router-id 0.0.0.2  
  
  network 192.168.0.0/24 area 0  
  network 192.168.6.0/24 area 1  
!  
log stdout
```

**Definición de áreas en ABR**

# OSPFv2

## ■ Ejercicio:

- Configurar 5 máquinas virtuales para crear el siguiente AS



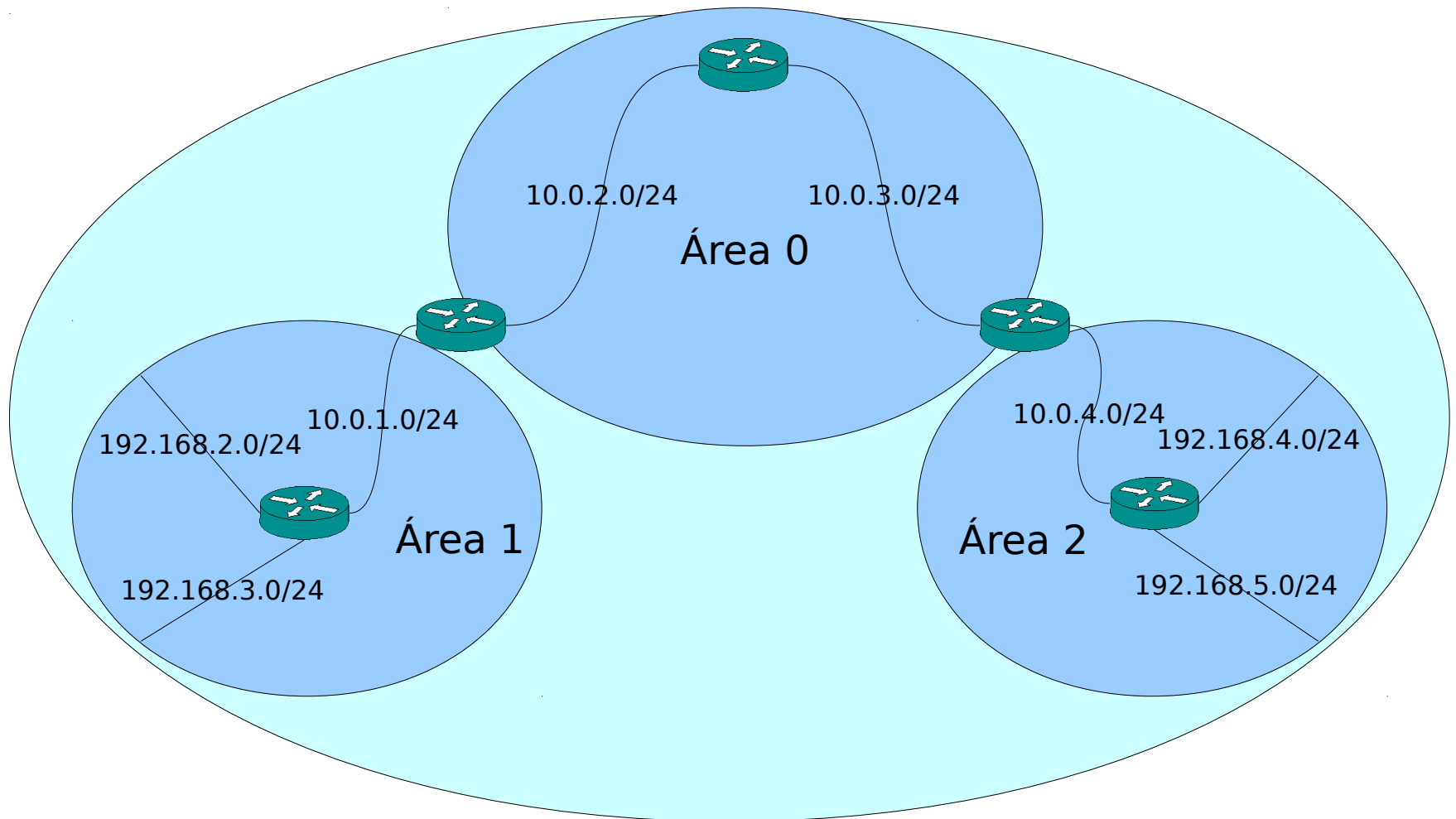
## ■ Notas:

- En los encaminadores internos de las áreas 1 y 2 se podría poner:  
network 0.0.0.0/0 area 1 (ó 2)
- Pero entonces hay un problema. ¿Cuál? ¿Cuál sería la solución?
- Comprobar las rutas que se añaden en cada encaminador
- Definir las áreas 1 y 2 como *stub*, en todos los encaminadores del área:  
area 1 stub
- Comprobar que se añade una ruta por defecto.
- Definir el área 1 como totally stubby: en el encaminador ABR:  
area 1 stub no-summary
- Comprobar que desaparecen las redes OSPF de fuera del área.

# OSPFv2

## ■ Ejercicio:

- Configurar 5 máquinas virtuales para crear el siguiente AS





# OSPFv2

## ■ Ejercicio:

```
router ospf
ospf router-id 0.0.0.1
network 10.0.0.0/24 area 0
network 10.0.1.0/24 area 1
network 192.168.0.0/24 area 0
network 192.168.1.0/24 area 0
area 1 stub no-summary
area 1 range 192.168.6.0/23
passive-interface eth1
passive-interface eth2
```

```
router ospf
ospf router-id 0.0.0.4
network 10.0.1.0/24 area 1
network 192.168.6.0/24 area 1
network 192.168.7.0/24 area 1
area 1 stub
passive-interface eth1
passive-interface eth2
```

```
router ospf
ospf router-id 0.0.0.2
network 10.0.0.0/24 area 0
network 10.0.2.0/24 area 2
network 192.168.2.0/24 area 0
network 192.168.3.0/24 area 0
passive-interface eth2
passive-interface eth3
```

```
router ospf
ospf router-id 0.0.0.3
network 10.0.2.0/24 area 2
network 192.168.4.0/24 area 2
network 192.168.5.0/24 area 2
passive-interface eth1
passive-interface eth2
```

## ■ Ejercicio:

- Comprobar las tablas de rutas de los encaminadores:  
ip route show
- Aparecen las direcciones privadas 10.0.0.0/8.
- Para suprimirlas, añadir en la sección **router ospf** (sólo ABR):  
area <X> range 10.0.0.0/8 not-advertise
- Probar a hacer traceroute desde uml3 a uml1.
- Comprobar con ayuda de wireshark qué sucede.

## ■ Ejercicio:

- En el caso de que todos los encaminadores estuvieran en el área 0, las rutas a 10.x.0.0/16 seguirían apareciendo.
- Para evitarlo:

```
router ospf
  distribute-list not-private out connected

access-list not-private deny 10.0.0.0/8
access-list not-private permit any
```

# Protocolos de Encaminamiento Interior



*OSPFv3*

## ■ Características

- Definido en el RFC5340 (julio 2008) y actualizado en RFC 6845, 6860 y 7503.
- Específico para IPv6.
- Basado en OSPFv2, con algunas mejoras.
- Distribuye prefijos IPv6.
- Utiliza directamente IPv6 (no necesita encapsulamiento adicional)
- Tipos de interfaces:
  - P2P (Point To Point)
  - P2MP (Point To Multipoint)
  - Broadcast
  - NBMA (Non Broadcast Multiple Access)
  - Virtual

# OSPFv3

---

## ■ Link LSA

- Un único LSA por link.
- Inundación de ámbito “enlace local”
- Proporciona dirección de encaminador de enlace local.
- Enumera todos los prefijos IPv6 asociados con el enlace.

## ■ Inter-Area Prefix LSA

- Enumera los destinos fuera del área pero dentro del SA.
- Se crea un resumen del área, que se inyecta en las demás áreas.
- Se origina en un ABR.
- Sólo se inyectan en la troncal las rutas intra-área.

# OSPF6d

## ■ Configuración de OSPFv3

- Mediante CLI (vtysh)

```
interface eth0
  ipv6 ospf6 cost 2
  ipv6 ospf6 hello-interval 60
  ipv6 ospf6 dead-interval 240
  ipv6 ospf6 retransmit-interval 5
  ipv6 ospf6 priority 5
  ipv6 ospf6 transmit-delay 1
  ipv6 ospf6 instance-id 0

interface eth1
  ipv6 ospf6 passive
  ipv6 ospf6 cost 50

router ospf6
  router-id 1.1.1.26
  area 0.0.0.0 range 2001:db8:400:139::/64
  area 0.0.0.22 range 2001:db8:200::/48
  interface eth0 area 0.0.0.0
  interface eth1 area 0.0.0.22
```

# OSPF6d

## ■ Verificación del estado mediante CLI

```
root@uml1:~# vtysh
```

```
Hello, this is Quagga (version 0.99.23.1).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.
```

```
uml1# show ipv6 ospf6 neighbor
```

Neighbor ID	Pri	DeadTime	State/IfState	Duration I/F[State]
1.1.1.170	30	-327806:-3:-56	Full/BDR	2d19:05:01 eth0[DR]

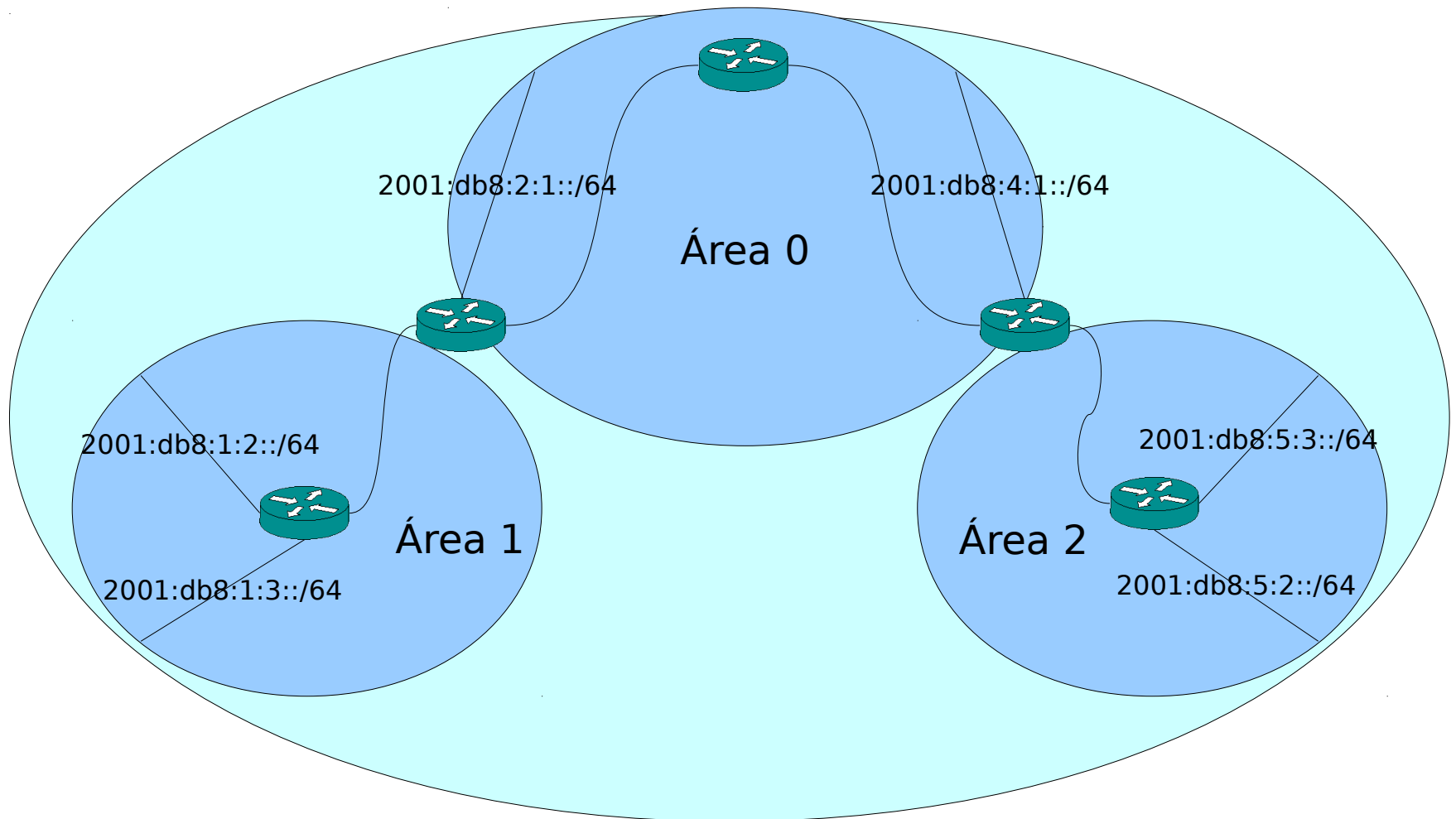
```
uml1#
```



# OSPFv2

## ■ Ejercicio:

- Configurar 5 máquinas virtuales para crear el siguiente AS



# OSPFv3

## ■ Solución

### ■ Para uml2

```
interface eth0
  ipv6 ospf6 cost 10

interface eth1
  ipv6 ospf6 cost 10

interface eth2
  ipv6 ospf6 cost 15
  ipv6 ospf6 passive

interface eth3
  ipv6 ospf6 cost 15
  ipv6 ospf6 passive

router ospf6
  router-id 0.0.0.1
  interface eth0 area 0.0.0.0
  interface eth2 area 0.0.0.0
  interface eth3 area 0.0.0.0
  interface eth1 area 0.0.0.1
```