



3º Grado en Ingeniería Informática

Transmisión de Datos y Redes de Computadores

TEMA 2. CAPA DE RED

(2021-2022)



TEMA 2. Índice

- © **2.1.** Enrutamiento estático y dinámico. (2h)
- © **2.2.** Protocolos de enrutamiento. (2h)
- © **2.3.** El problema del direccionamiento en IPv4. (4h)

APLICACIÓN

PRESENTACIÓN

SESIÓN

TRANSPORTE

RED

ENLACE

FÍSICO





TDRC

Tema 2.2.

Protocolos de enrutamiento

Antonio M. Mora García



Tipos de enrutamiento

- **Estático:**

El administrador decide las rutas a seguir dentro de su red, mediante la *configuración manual* de los routers, definiendo sus tablas de enrutamiento.

- **Dinámico:**

Existen ***algoritmos o protocolos*** entre routers que establecen de forma dinámica las mejores rutas de acuerdo a un *criterio definido* por el administrador.

Protocolos de enrutamiento

CARACTERÍSTICAS:

- **Exactitud:** encaminamiento entre los host origen y destino deseados (y no otros).
- **Robustez:** el algoritmo debe ser capaz de adaptarse ante posibles fallos en la red (buscar rutas alternativas).
- **Estabilidad:** el mantenimiento de la robustez no puede poner en compromiso el funcionamiento de la red, provocando cargas importantes o incluso bucles en algunas zonas.
- **Optimización:** debe proporcionar la mejor solución posible (considerando un criterio).
- **Imparcialidad:** debe haber un compromiso para que unos nodos no se vean perjudicados (por ser siempre elegidos o no elegidos en la ruta).
- **Eficiencia:** debe consumir los menores recursos posibles.
- **Simplicidad:** debe ser sencillo de implementar para que ofrezca más beneficios que el coste de ponerlo en funcionamiento.

Protocolos de enrutamiento

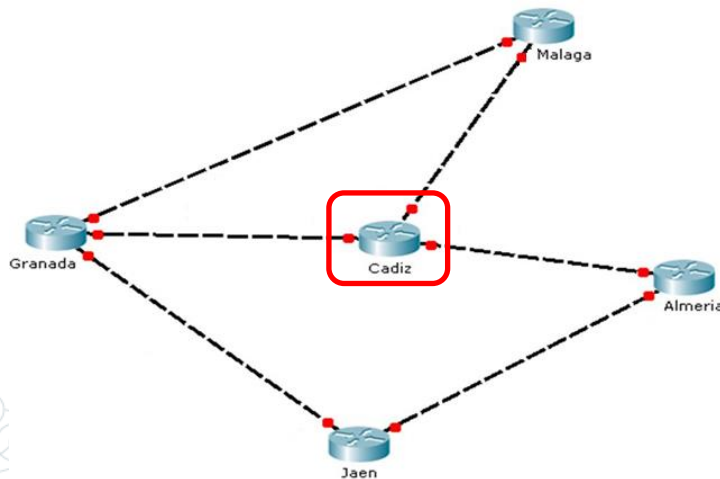
ESPECIFICACIÓN:

- **Criterio de decisión:** métrica usada para la decisión del encaminamiento (distancia, número de saltos, retardo, carga).
- **Instante de decisión:** momento en el que se toma la decisión de encaminamiento (al inicio de la comunicación, en cada salto).
- **Lugar de decisión:** puntos que toman las decisiones de encaminamiento (al inicio, en un nodo central de la red, en cada nodo por el que se pase).
- **Fuente de información:** procedencia de los datos considerados para hacer el encaminamiento (local, de los vecinos, de todos los nodos de la red).
- **Tiempo de actualización:** momento en el que se actualizan los datos para el encaminamiento (invariable, cada X segundos). De este factor depende la adaptabilidad del protocolo a los cambios en la red.

Protocolos de enrutamiento

ALGORITMOS DE ENRUTAMIENTO (RESPECTO AL LUGAR DE DECISIÓN):

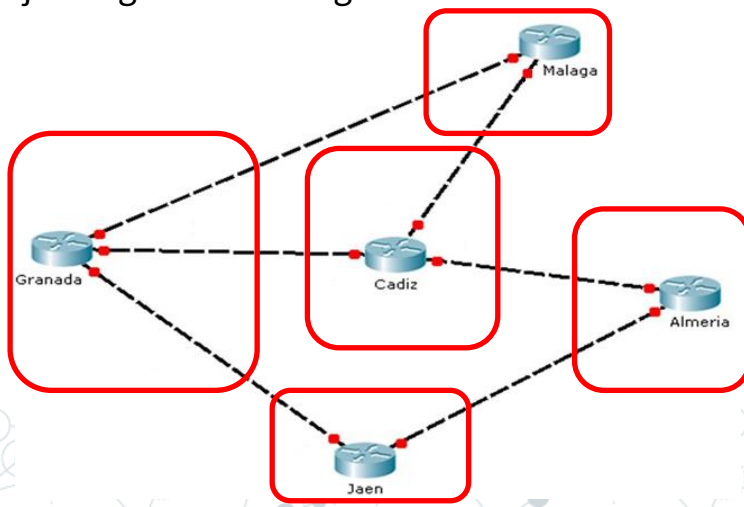
- **Centralizados:** un nodo recibe la información de todos los demás, crea el grafo completo de la red y determina las rutas óptimas. Después pasa esa información a los demás routers de la red.
 - Ventajas → rutas calculadas conociendo la red completa
 - Desventajas → mucho tráfico alrededor del nodo central. Actualización de cambios lenta. Inconsistencias temporales (los vecinos reciben las tablas antes)



Protocolos de enrutamiento

ALGORITMOS DE ENRUTAMIENTO (RESPECTO AL LUGAR DE DECISIÓN):

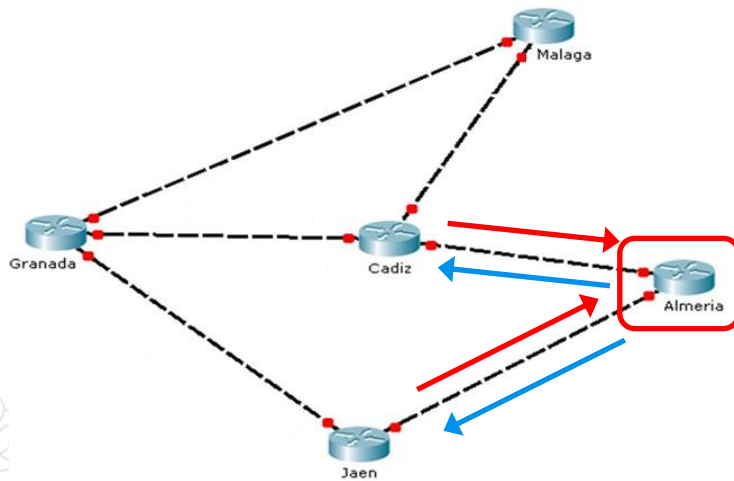
- **Aislados:** cada nodo es responsable de calcular sus propias tablas, sólo considerando información local. Se basa en el retardo medio medido en cada una de sus interfaces de salida. Se puede usar inundación (*flooding*) y enviar los paquetes por todas las líneas de salida menos por la que llega.
 - Ventajas → muy simple, el paquete llega al destino, en el menor tiempo posible.
 - Desventajas → gran sobrecarga de tráfico en la red. Solución: *Spanning Tree*



Protocolos de enrutamiento

ALGORITMOS DE ENRUTAMIENTO (RESPECTO AL LUGAR DE DECISIÓN):

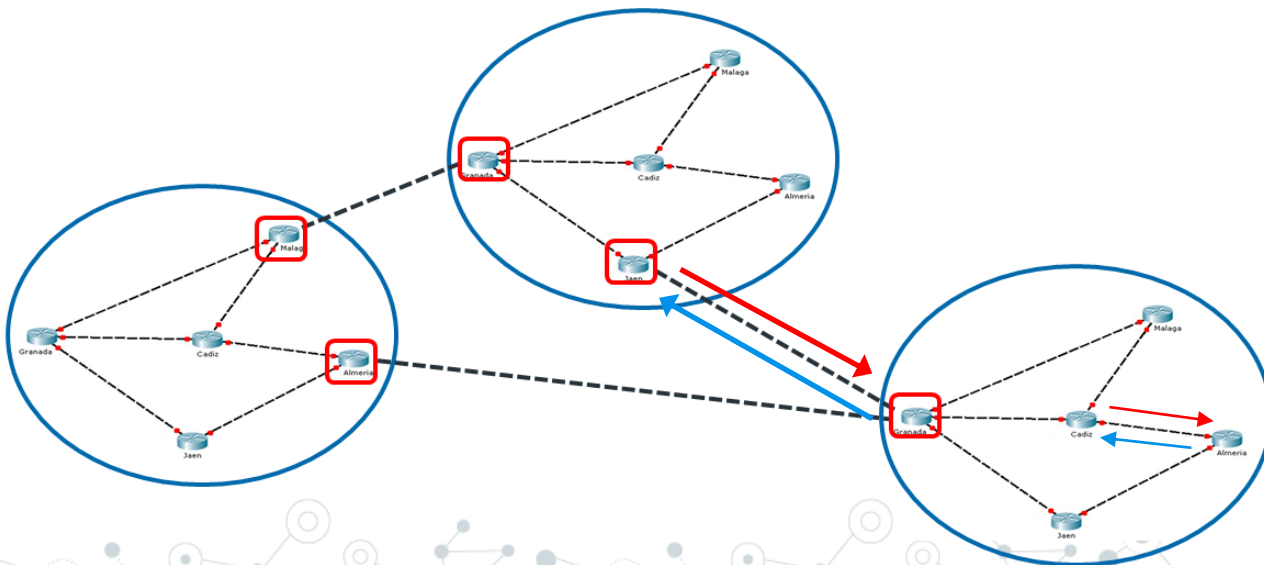
- **Distribuidos:** cada nodo es responsable de calcular sus propias tablas, pero intercambia información con sus vecinos, por lo que se cuenta con más información que en el caso aislado. Con la información recibida (tablas de enrutamiento de los vecinos) cada nodo actualizará su propia tabla.
 - Ventajas → se genera menos tráfico que en el caso aislado. Mejores rutas que en ese mismo caso.
 - Desventajas → actualizaciones lentas. Rutas sub-óptimas temporalmente (hasta que se haya propagado la noticia de un cambio en la red a todos los nodos).



Protocolos de enrutamiento

ALGORITMOS DE ENRUTAMIENTO (RESPECTO AL LUGAR DE DECISIÓN):

- **Jerárquicos:** se divide la red en regiones con varios nodos, que siguen un esquema de actualización distribuido entre ellos (dentro de la región) y, a su vez, se sigue un esquema distribuido entre regiones (sólo nodos determinados de cada región).

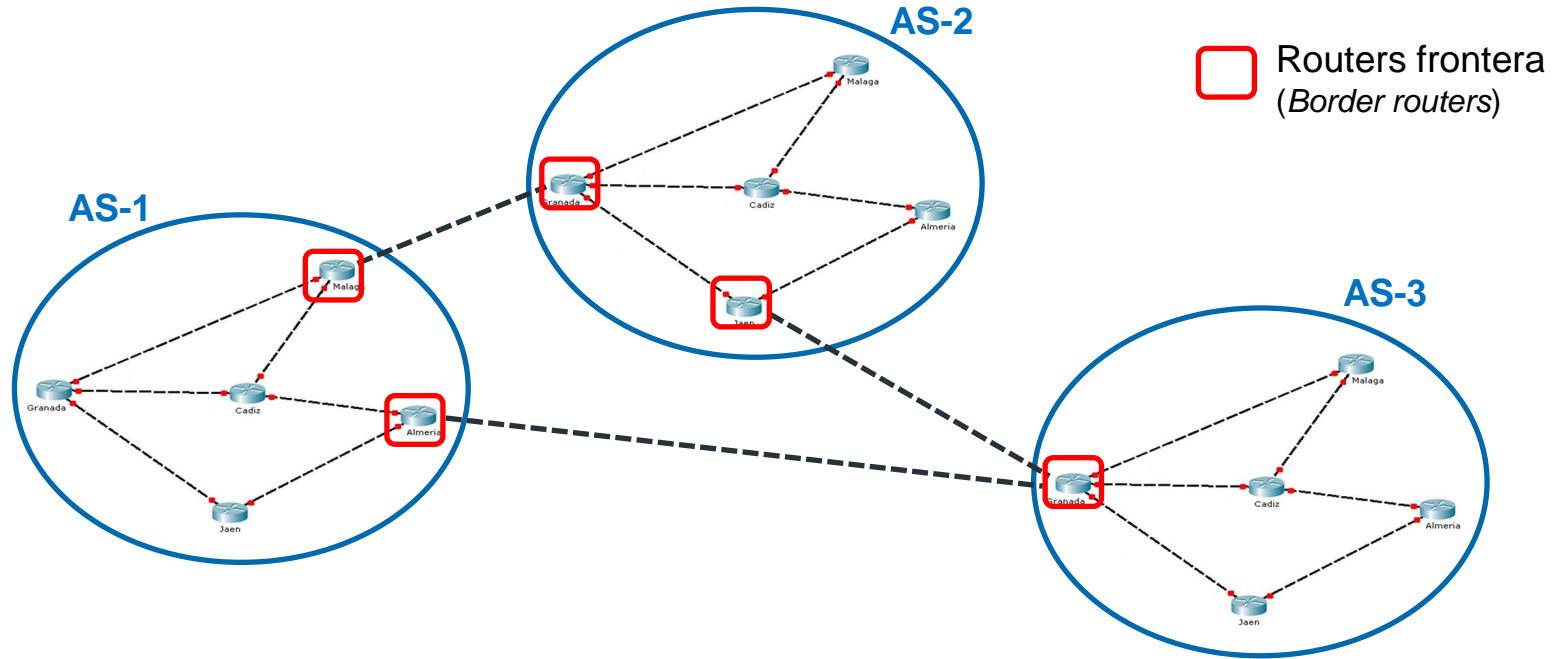


Sistemas Autónomos

- Un SA es una red definida por una **única política administrativa**, es decir, tiene una gestión de rutas común e independiente de otros sistemas.
- Para **facilitar la administración** y **aumentar la escalabilidad** Internet se jerarquiza en **Sistemas Autónomos (SA)**.
- Cada **SA se identifica por un entero de 16 bits** (*DESDE 2007 ES 32-BITS*). Ej: Rediris → AS766
- Cada **SA informa a los otros SAs** de las **redes accesibles**.

Existe un router responsable de esto, denominado **router exterior** (o *router frontera*).
- Se aplica **encaminamiento dinámico** (mediante algoritmos automáticos).

Enrutamiento en Internet



Protocolos de enrutamiento

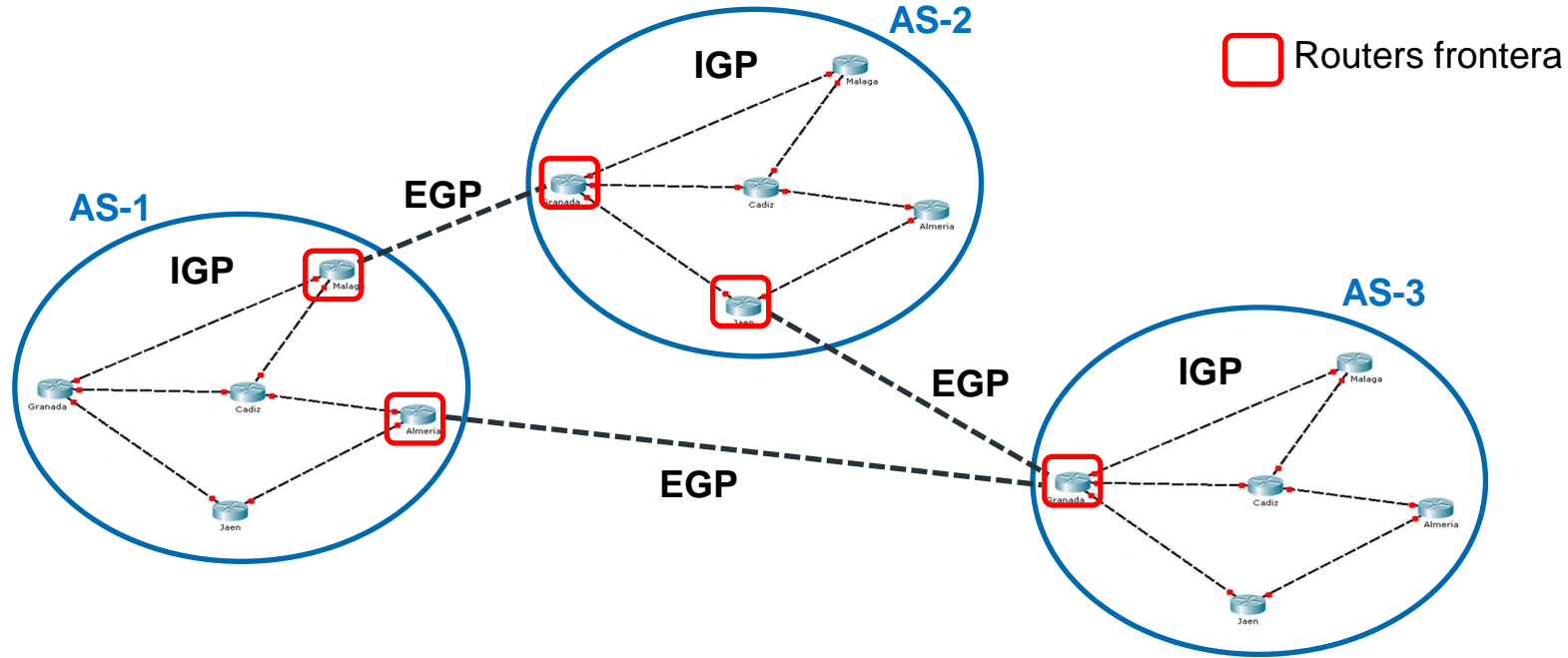
- **IGP** (*Interior Gateway Protocol*):

Son los que se dan **dentro de un sistema autónomo**. Ejemplos: RIP (*RFC 2453*) y OSPF (*RFC 2328*).

- **EGP** (*Exterior Gateway Protocol*):

Son los que utilizan **distintos sistemas autónomos entre sí**, como podrían ser distintos ISPs. Ejemplo: BGP (*RFC 4271*).

Enrutamiento en Internet



IGP (Interior Gateway Protocol)

- **Vector distancia:**

Los routers construyen su tabla de rutas con el único conocimiento de la distancia (métrica) y el siguiente salto (*next hop*) para llegar a la red de destino.

Ejemplo: RIP

- **Estado del enlace:**

Los routers necesitan conocer previamente toda la topología de la red (cómo se conectan todos los nodos entre sí) antes de generar su tabla de enrutamiento.

Ejemplo: OSPF

- **Híbrido:**

Una mezcla entre ambos. Ejemplo: EIGRP

Vector Distancia

- No conoce la topología de la red. **Solo conoce distancia** (métrica) **y *next hop***.
- Esta distancia puede ser un número que indica: longitud del enlace, número de saltos, latencia (tiempo medio) u otros valores.
- **Actualizaciones frecuentes** (necesarias ya que se desconoce la topología).
- Las actualizaciones que se **envían** son **copias de la tabla de rutas** con el incremento correspondiente al enlace hasta el router al que se envían.
- Cada router lo comunica **a sus vecinos**, no a todos.
- Consume **poca CPU y poca memoria**.

Vector Distancia

- Utiliza un **Algoritmo de Bellman-Ford** para calcular las rutas.
 - Cada router calcula la distancia (métrica) a todos los demás dentro de su red (dentro del AS en el que está). Almacena esta información en una tabla.
 - Cada router envía su tabla a todos sus vecinos (directamente conectados con él).
 - Cuando un router recibe las tablas de distancias de sus vecinos, éste calcula la ruta más corta a los demás routers y actualiza su tabla para reflejar los cambios.
- **Problemas:**
 - Mala escalabilidad: si la red crece, el algoritmo adquiere mucha complejidad.
 - Convergencia lenta: si hay cambios en la topología de la red, éstos tardan en informarse a todos los routers, ya que las actualizaciones se distribuyen nodo por nodo.

Vector Distancia (cuenta al infinito)

- Problema en el que se producen bucles por culpa de actualizaciones de rutas publicadas por la misma interfaz, es decir, desde el mismo router al que se quiere llegar.



- > La distancia de R1 a los hosts de la Red A es 1
- > La distancia de R2 a esos hosts es 2 (a través de R1)
- > La distancia de R3 a esos hosts es 3 (a través de R2)

Vector Distancia (cuenta al infinito)

- Si se cae el enlace entre R1 y la Red A.



> **!!!R1 podría acceder a Red A a través de R2, con distancia total 3!!!**

> La actualización posterior haría que R2 pudiese llegar a la Red A en 4 saltos (a través de R1→R2→R1)

> La actualización posterior haría que R3 pudiese llegar a la Red A en 5 saltos

... etc, etc

Vector Distancia (soluciones a la cuenta al infinito)



- **Horizonte dividido** (*split horizon, SH*): Un router sólo publica una ruta a un destino por una interfaz (enlace) por la que no pase el camino a dicho destino.
> R2 no publicaría su ruta hasta Red A a través de la interfaz que llega a R1.
- **Envenenamiento inverso** (*poison reverse, PR*): La ruta que se aprende por una interfaz es publicada por esa misma interfaz pero con métrica infinita.
- **Envenenamiento de ruta** (*route poisoning, RP*): Si una ruta falla, se comunica mediante difusión a todos los nodos indicando métrica infinita (red inalcanzable).
- **Tiempo de espera** (*hold down time, HDT*): Después de conocer que la ruta a una red ha fallado, el router deja pasar un tiempo (60 segs) antes de aceptar nuevas rutas a ese destino.
- **Actualización por disparo** (*triggered update, TU*): En cuanto se conoce que la ruta a una red no está disponible se publica inmediatamente al resto de routers sin esperar al periodo de actualización.

RIP

- **Routing Information Protocol** (RFC 1058, 2453, 4822).
- Es un protocolo de **enrutamiento IGP**.
- Adopta un **algoritmo vector-distancia** (Bellman-Ford Distribuido)
- La métrica que emplea es el **número de saltos**.
- Una red directamente conectada a un router tiene coste 0.
- **Máximo de 15 saltos** (16 sería considerada distancia infinita o no alcanzable).
- Periódicamente (por defecto **cada 30 segundos**) cada **router RIP recibe de todos sus vecinos y envía a todos sus vecinos** (dirección multicast 224.0.0.9) los **vectores-distancia para todos los posibles destinos**.
- De entre ellos, para un **destino dado**, se **selecciona como salto siguiente el vecino que anuncie el menor coste**, considerando en la métrica para ese destino el coste + 1 (coste para alcanzar ese vecino desde el router actual).

Problema
convergencia lenta:
Las malas noticias tardan en propagarse

RIP

- El router que recibe el paquete sabe que el **next-hop es el origen del paquete** (IP origen del paquete RIP).
- En el paquete con las rutas no hay máscaras asociadas a ellas (en Versión 1).
- Protocolo de **capa de aplicación**. Para el intercambio de información se usan datagramas UDP sobre el puerto reservado 520.
- En un paquete de RIP pueden ir **hasta 25 rutas**.
- Si la tabla de rutas a enviar es mayor se tendrán que enviar varios paquetes (varios datagramas UDP).

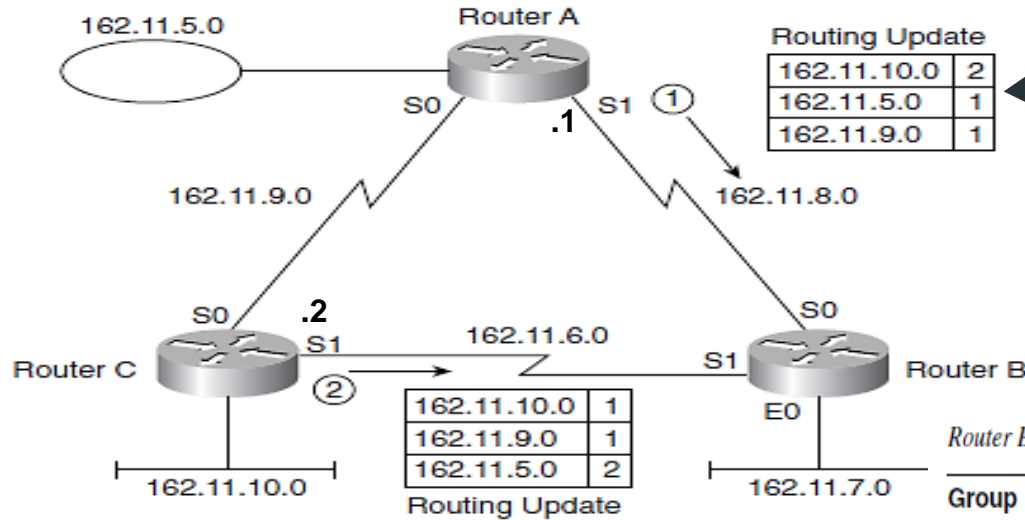
RIP v2:

- Se envían también las máscaras de red.
- Compatible con redes con máscaras de tamaño variable (VLSM, *Variable Length Subnet Mask*).
- Permite autenticación.

RIP (Escenario 1)

Múltiples rutas a la misma subred con igual métrica

Routers A and C Advertising to Router B



1.- RouterA envía actualización a RouterB indicando en cuántos saltos RouterA puede alcanzar estas redes + 1

3.- Router B compila el mejor camino posible y actualiza el coste. Además, aparecen dos rutas de igual coste (métrica) a la misma red.

2.- RouterC envía actualización a RouterB indicando en cuántos saltos RouterC podría alcanzar estas redes + 1

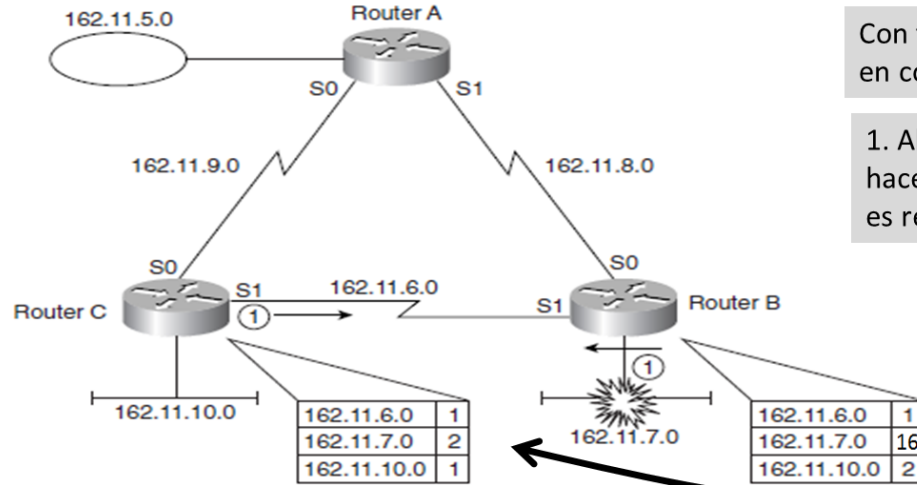
Router B Routing Table,

Group	Outgoing Interface	Next Router	Metric
162.11.5.0	S0	162.11.8.1	1
162.11.6.0	S1		0
162.11.7.0	E0		0
162.11.8.0	S0		0
162.11.9.0	S0	162.11.8.1, 162.11.6.2	1
162.11.10.0	S1	162.11.6.2	1

RIP (Escenario 2)

Sin horizonte dividido

¡PROBLEMAS!



Con tablas erróneas la red tarda más en converger

1. Al caerse 162.11.7.0, RouterB hace route poisoning (métrica 16 es red inalcanzable)

1. Por casualidad, RouterC envía una actualización simultánea a RouterB. RouterC incumple la regla del Split Horizon para la red 162.11.7.0

Router B Routing Table, After Subnet 162.11.7.0 Failed and Update from Router C Is Received

Group	Outgoing Interface	Next Router	Metric
162.11.6.0	S1		0
162.11.7.0	S1		2
162.11.10.0	S1	162.11.6.2	1

Router C Routing Table, After Subnet 162.11.7.0 Failed and Update from Router B Is Received

Group	Outgoing Interface	Next Router	Metric
162.11.6.0	S1		0
162.11.7.0	S1		16
162.11.10.0	E0		1

2.- Esto causa que la tabla de rutas de Router B sea errónea !!

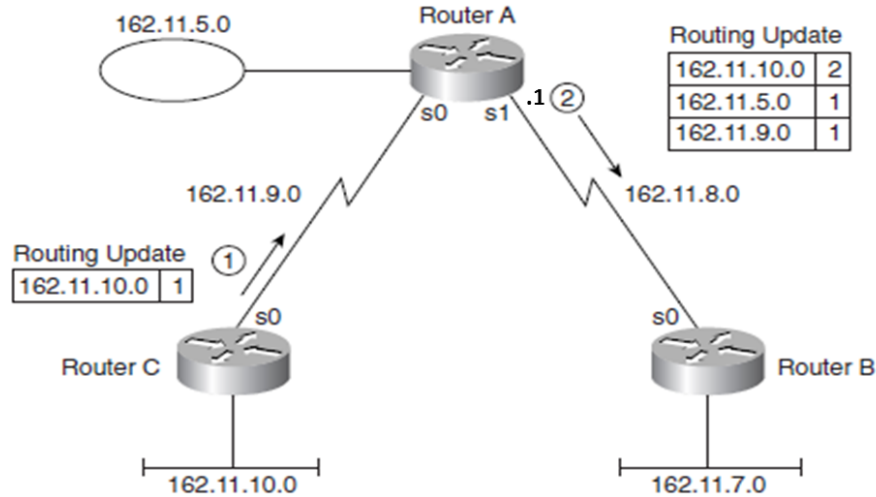
3.- Tabla correcta en RouterC, pero no ha evitado tabla de rutas corrupta en RouterB. Tendrá que hacer Route poisoning en próxima actualización. Convergencia lenta

RIP (Escenario 3)

Con horizonte
dividido

TODO OK

Router A Advertising Routes Learned from Router C



Situación: La red ha convergido y todos los routers tienen en sus tablas de rutas todas las redes (5 en total). Hay total conectividad

1.- RouterC envía una actualización a RouterA. Sólo contiene aquellas rutas que no fueron aprendidas de RouterA o que tienen en común

2.- RouterA actualiza a RouterB. Sólo contiene aquellas rutas que no fueron aprendidas de RouterB o que tienen en común

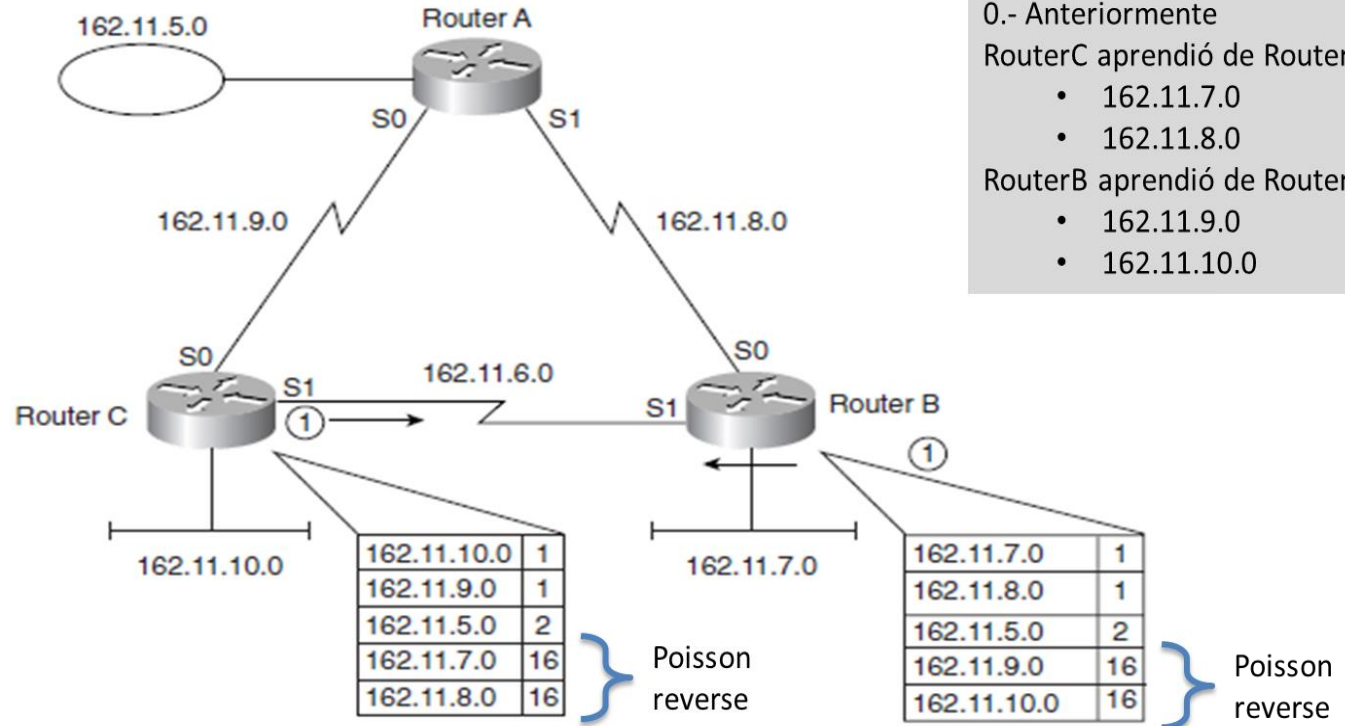
Router B Routing Table, After Receiving Update in Figure 6-3

Group	Outgoing Interface	Next Router
162.11.5.0	S0	162.11.8.1
162.11.7.0	E0	
162.11.8.0	S0	
162.11.9.0	S0	162.11.8.1
162.11.10.0	S0	162.11.8.1

RIP (Escenario 4)

Con
envenenamiento
inverso

TODO OK



0.- Anteriormente
RouterC aprendió de RouterB:

- 162.11.7.0
- 162.11.8.0

RouterB aprendió de RouterC:

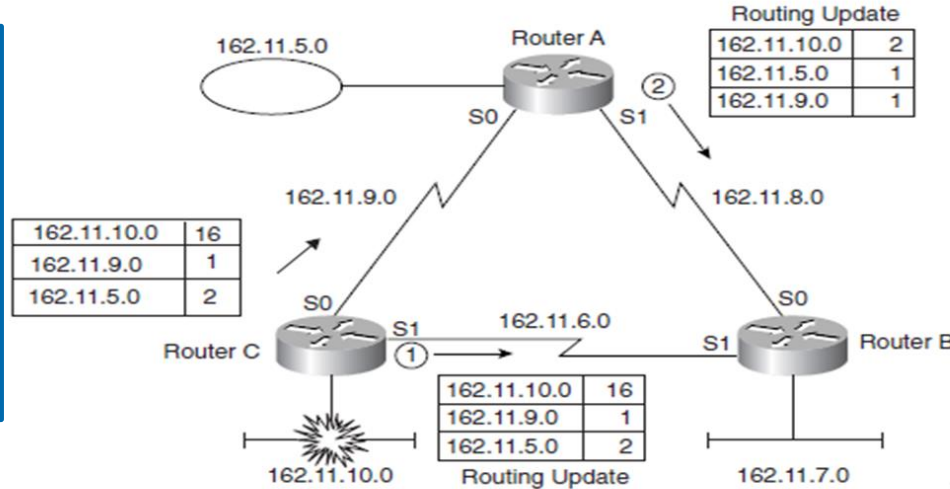
- 162.11.9.0
- 162.11.10.0

1.- En las actualizaciones mutuas entre RouterB y RouterC, las rutas aprendidas del otro son incluidas en las actualizaciones, pero con métrica infinita 16 (poisson reverse). Por tanto, se trata de una reafirmación activa del concepto split horizon.

RIP (Escenario 5)

Cuenta al infinito:
Sin Tiempo de
espera (HDT) +
Envenenamiento de
ruta (RP)

¡PROBLEMAS!



Router B Routing Table After Updates in Figure 6-7 Are Received

Group	Outgoing Interface	Next Router	Metric
162.11.5.0	S0	162.11.8.1	1
162.11.6.0	S1		
162.11.7.0	E0		
162.11.8.0	S0		
162.11.9.0	S0	162.11.8.1	1
162.11.10.0	S0	162.11.8.1	2

1.- Al caerse la red 162.11.10.0, RouterC envía actualizaciones con RP a RouterA y RouterB.

2.- Una actualización de RouterA sale hacia RouterB justo antes de que llegue la actualización con RP de RouterC y después de que ésta llegue al RouterB

3.- RouterB no considera HDT y cree ahora que tiene una ruta hasta 162.11.10.0 a través de RouterA (2 saltos)

4.-RouterB actualizará a RouterC (3 saltos)

5.- RouterC actualizará a RouterA (4 saltos)

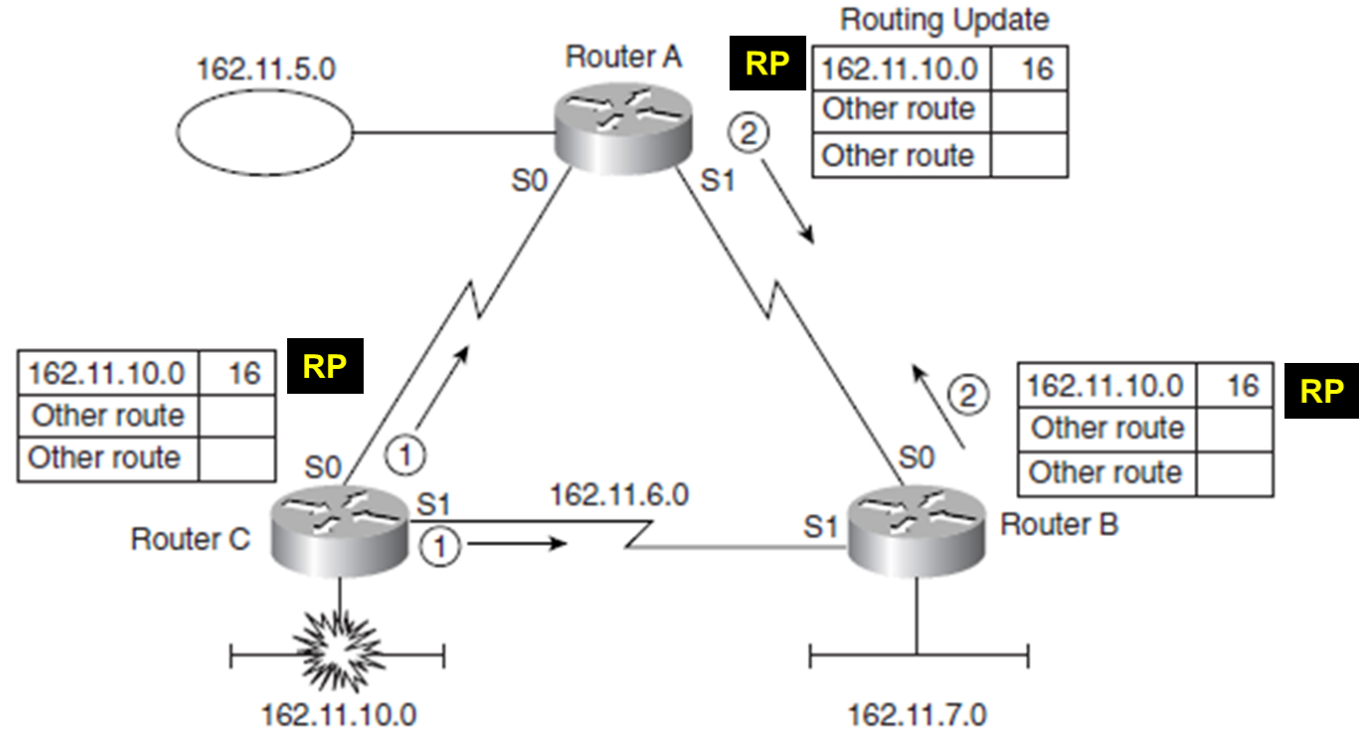
6.- RouterA actualizará a RouterB (5 saltos)

...

RIP (Escenario 6)

Con actualización
por disparo

TODO OK



1.- En cuanto falla una red, no se espera a la actualización periódica (e.g. RIPv2 cada 30 segundos), sino que se publica una actualización de forma inmediata con un **RP (Route Poisoning)**

Estado del enlace

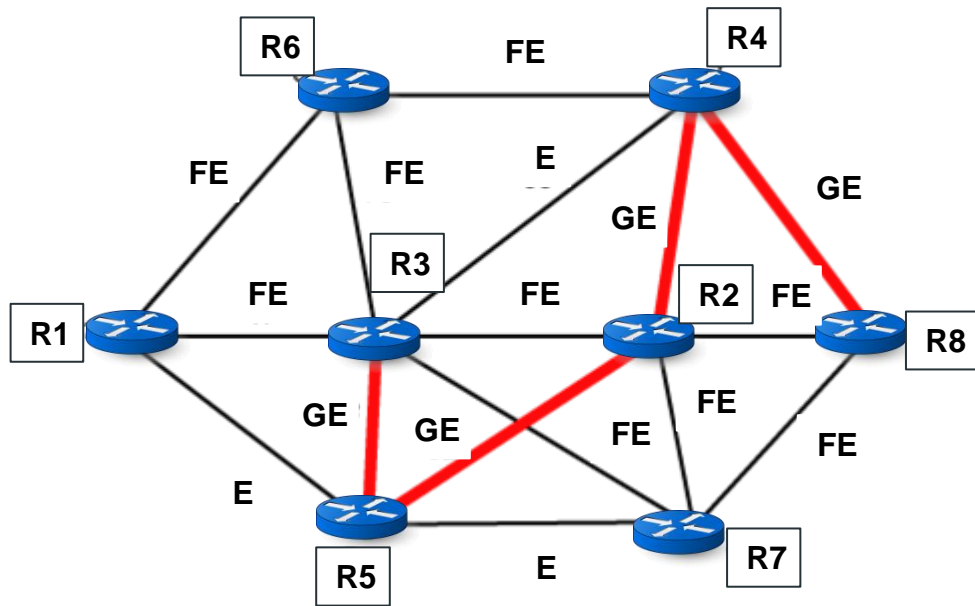
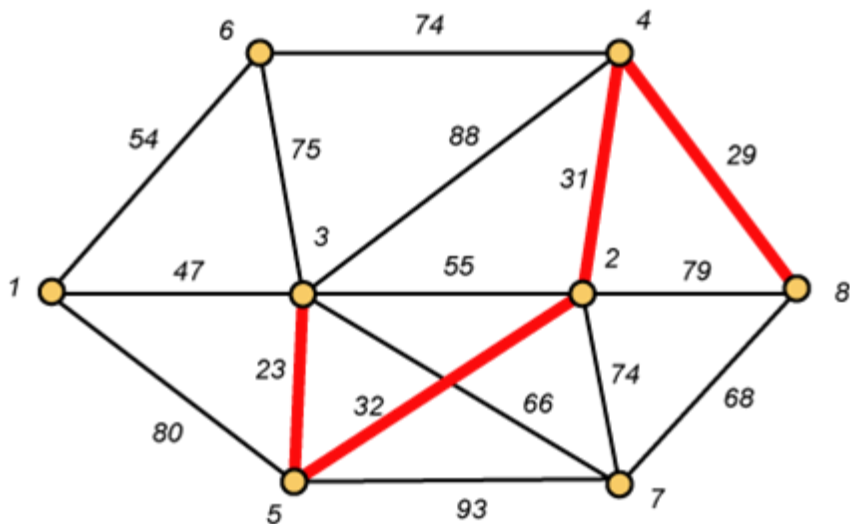
- Requiere conocerse **la red completa**.
- Cada router informa de **conexiones con sus vecinos** (nodos directamente conectados con él).
- Construye un **paquete LSP** (***Link State Packet***) que dice quién es el router, la lista de sus vecinos (nodos/redes alcanzables) y sus respectivas distancias desde él.
- Envía su LSP **por inundación** a todos los routers de la red.
- **Se reciben los LSP** de todos los nodos de la red.
- **Mejor convergencia**, ya que no hay calculo para obtener las rutas.

MEJORAS

- La inundación se hace reenviando las LSP por todas las interfaces excepto por la que se recibió.
- Los LSP se numeran para detectar y descartar duplicados, además tienen un TTL (Sólo se envían los LSP nuevos y que no están expirados).

Estado del enlace (Algoritmo de Dijkstra)

- Busca los caminos mínimos desde un nodo a todos los demás en un grafo (o una red).



E → Ethernet

FE → Fast Ethernet

GE → Gigabit Ethernet

OSPF

- **Open Shortest Path First** (RFC 2328).
- Es un protocolo de **enrutamiento IGP**.
- Basado en el **Estado de los Enlaces**.
- Se **publican los estados** por difusión/inundación.
- Maneja **distintas tablas (BD)**: vecinos, topología, rutas.
- Los routers envían a los demás routers de la red **paquetes** en los que informan de las **redes a las que están directamente conectados** y los routers vecinos que tienen.
- Al final todos los routers conocen el grafo de la topología con el coste de cada enlace.
- Las **actualizaciones** se hacen **sólo cuando hay cambios en la red**.
- **Mejor convergencia**, ya que no hay que hacer cálculos sobre las rutas a difundir.

OSPF

- Aplica el **algoritmo de Dijkstra** para el cálculo de los caminos mínimos.
- **A partir de las rutas** se **construyen** las **tablas de encaminamiento** de cada router.
- El **coste por defecto** que se considera en OSPF para cada enlace es: **coste = $10^8/\text{BW}$** .
Ej: para un enlace con BW = 1 Mbps
$$\text{coste} = 10^8/10^6 = 100$$
- El **coste** de los enlaces **se podrá determinar en tiempo real** → un administrador o un algoritmo automático.
- Permite calcular **rutas alternativas** y hacer **balanceo de carga**.
- Se pueden considerar **distintas métricas** y **dar prioridad a unos enlaces** sobre otros.
- El coste de los enlaces puede ser diferente de unos a otros e incluso según el sentido.
- **No tiene un coste infinito.**
- Funciona directamente **sobre protocolo IP**.

OSPF

MENSAJES:

- **hello** → para descubrir vecinos y routers adyacentes. Comprobar estado de los enlaces.

Los cambios en la red se detectan enviando mensajes HELLO cada cierto tiempo (10s) a los vecinos que tiene cada router en su tabla de vecinos.

Si alguno no responde, ha habido un cambio (se ha caído). También se detectan nuevos vecinos usando esos paquetes HELLO.

- **database description** → mensajes con los datos de las tablas que se manejan (vecinos, topología).
- **link status request/update/ack** → descripciones del estado de los enlaces.

Los routers intercambian paquetes LSA (*Link State Advertisements*) indicando las redes a las que están directamente conectados y sus respectivos routers adyacentes.

OSPF (terminología)

Vecino:

- Router con conexión directa con el que intercambio información de rutas.
- Se descubren con los paquetes de *Hello* (cada 10 seg)
- Todos los routers OSPF, direccion IP multicast 224.0.0.5

Área:

- Regiones dentro de la red.
- Backbone es el área central de la red. A ella se conectan todas las demás áreas.

Adyacencia:

- Formada por 2 routers vecinos.
- Cada router forma una adyacencia con el Designated router (DR) y el Backup DR (BDR).
- Una adyacencia consiste en el intercambio de enlaces de topología conocidos.

OSPF (terminología)

DR/BDR

- Responsable de formar las adyacencias con los demás routers vecinos para garantizar que todos tienen la misma tabla topológica.
- El BDR hace las mismas funciones que el DR cuando éste cae.
- Se eligen mediante el protocolo de *Hello*:
 - Los dos que tengan configurados la prioridad o las IPs más altas son elegidos DR y BDR.
- Sólo se da en una red multiacceso (routers unidos por un conmutador), pero no en enlaces punto a punto.
- El DR será el encargado de hacer adyacencias con otros DR de otras redes/áreas.

OSPF (terminología)

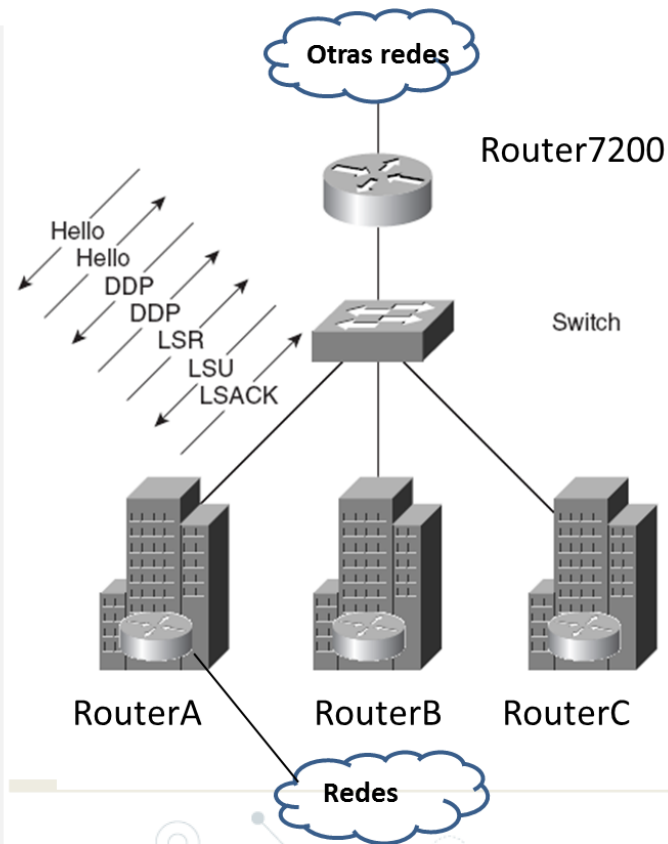
Tablas

- **Tabla de vecinos.**
- **Tabla de Topología:**
 - Tabla con los enlaces conocidos dentro de un área.
 - Los enlaces se descubren a través del intercambio de información al formar adyacencia.
 - Todos los routers en el mismo área tienen la misma tabla de topología.
 - Desde cada Router se ejecuta el algoritmo de Dijkstra sobre ella para calcular el mejor camino (SPF) a cada red destino. El resultado es la tabla de rutas.
- **Tabla de rutas.**

OSPF (Ejemplo)

Ejemplo: Generación Tabla de rutas

- 1.- RouterA se une por primera vez a una red OSPF.
- 2.- Envía paquetes de Hello para descubrir a sus vecinos en la misma red y conocer al DR y BDR
- 3.- Descubre a todos los routers vecinos. Router7200 es el DR, por tanto establece adyacencia con él.
- 4.- Se intercambian información de los enlaces/redes que conoce cada uno. Paquetes OSPF
 - DDP: Database descriptor packets: Usado para mandar info para sincronización de tabla de topología
 - LSR: Link State Request. Petición de mayor información (e.g. Después de un DDP por inconsistencias)
 - LSU: Link State Update. Respuesta de un LSR
 - LSACK: Ack de LSU
- 5.- Una vez que la tabla de topología converge, ambos routers aplican el algoritmo de Dijkstra (SPF) a la Tabla de Topología y construyen sus tablas de rutas.
- 6.- El Router7200 también forma adyacencia con RouterB y RouterC y repite 4 y 5
- 7.- Router 7200 será el encargado de formar adyacencia con otros DR y BDR en Otras redes y repetir pasos 4 y 5

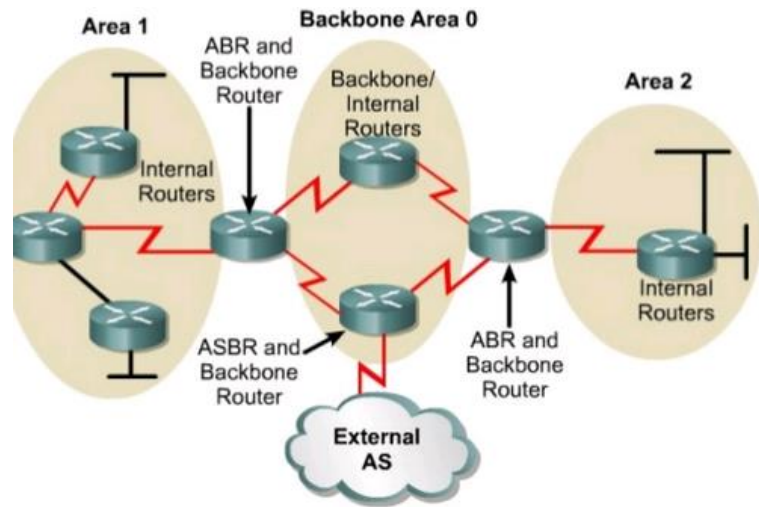


Nota: En el ejemplo sólo se contempla la adyacencia con el DR, no se ha considerado la formada con el BDR

OSPF (terminología)

Escalabilidad

- Al principio todos los routers están en el área 0 o de backbone.
- Cuando el número de routers OSPF se hace muy grande, el número de adyacencias no es manejable, incluso con DR/BDR, por lo que se define una nueva área en la que se van incluyendo nuevos routers.
- Las áreas son regiones que confinan las adyacencias de los routers que la componen
- Cada nueva área definida se conecta al área 0
- ABR: La interconexión entre áreas se hace mediante routers frontera llamados ABR (Area Border Router) que impiden que se formen adyacencias entre routers de distintas áreas. También son los encargados de hacer conocer a un área las redes alcanzables de la otra.
- ASBR: La interconexión con Internet se hace con un ASBR (Autonomous System Border Router)



<https://www.slideshare.net/ghostlinexawey/ospf-multi-area>

Vector distancia vs Estado del Enlace

Vector Distancia

- No conoce la topología de la red. Solo conoce distancia (métrica) y *next-hop*.
- Actualizaciones frecuentes (necesarias ya que se desconoce la topología). Convergencia lenta.
- Las actualizaciones que se envían son copias de la tabla de rutas con el incremento correspondiente.
- Una única tabla, la de rutas.
- Consume poca CPU y memoria.

Ejemplo: *RIPv2*

Estado del Enlace

- Todos los routers deben conocer la topología al completo del área o de la red para después calcular la tabla de rutas.
- Actualizaciones sólo en caso de cambios topología (necesarias para poder calcular la tabla de rutas).
- Las actualizaciones son informes de cambios de topología.
- Tres tablas: rutas, adyacencias (vecinos que envían las actualizaciones) y topología.
- Consumo de memoria. También gasto de CPU en el momento del cálculo de rutas.

Ejemplo: *OSPF*

BGP

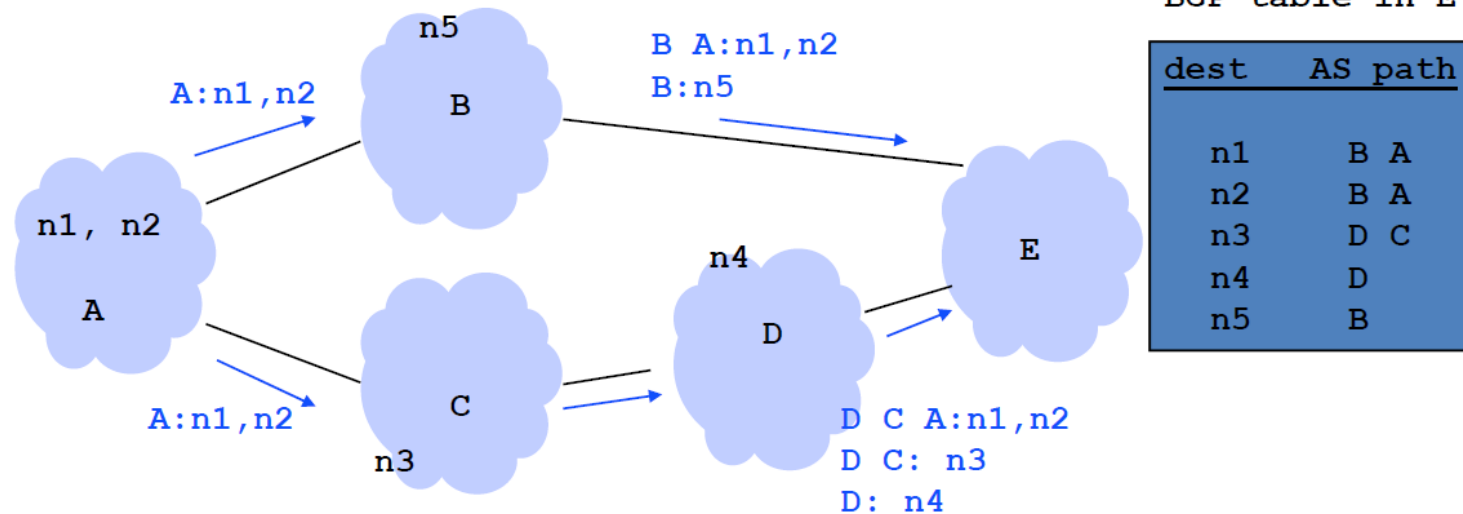
- **Border Gateway Protocol** (RFC 4271).
- Es un protocolo de **enrutamiento EGP**.
- Se utiliza entre Sistemas Autónomos. Lo hacen los **routers frontera** (deben soportar BGP).
- Es el más utilizado para enrutamiento en Internet.
- Funciona sobre **TCP en el puerto 179**.
- Es un protocolo **path-vector** o **vector-camino** (diferente de vector-distancia) en el que **se consideran políticas** (enrutamiento en Internet está muy influenciado por el aspecto económico/tarificación).
- Un **vector-camino** contiene una sucesión de Sistemas Autónomos (ASs) por lo que pasar para alcanzar un destino.
- La **Tabla BGP** contiene un destino y la ruta para alcanzarlo.
- Se puede utilizar dentro de un AS (i-BGP) o entre ASs (e-BGP).

BGP

- Cada vez que hay un cambio en una ruta se envían las rutas conocidas a los ASs vecinos.
- Se indican los dominios dentro del AS (puede haber varios) y el camino para alcanzarlos.

EJEMPLO:

(los 'n' son dominios)



Distancia Administrativa (*Administrative Distance*)

- **Medida** (usada por routers CISCO) para **seleccionar rutas** cuando hay varias alternativas hacia un destino.
- Indica la **calidad/fiabilidad de la fuente de enrutamiento**.

Fuente de enrutamiento	Distancia Administrativa
Conexión directa	0
Ruta estática	1
BGP (externo)	20
OSPF	110
RIP	120

menor valor DA →
mayor calidad

Ejemplo salida router Cisco

Seville#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

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, * - candidate default

U - per-user static route, o - ODR

RIP



DA



1 salto



away of last resc... et

```
10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
R    10.1.2.0/24 [120/1] via 10.1.5.252, 00:00:19, Serial1
R    10.1.1.0/24 [120/1] via 10.1.6.251, 00:00:22, Serial0
C    10.1.6.0/24 is directly connected, Serial0
C    10.1.5.0/24 is directly connected, Serial1
R    10.1.4.0/24 [120/1] via 10.1.6.251, 00:00:22, Serial0
      [120/1] via 10.1.5.252, 00:00:19, Serial1
C    10.1.3.192/26 is directly connected, Ethernet0
```



Igual métrica

Prioridad de enrutamiento

- En caso de que se disponga de **varias fuentes de enrutamiento para llegar a una red**, no todas las rutas aparecerán en la tabla de rutas. Existen unos **criterios**:
 - La fuente con **menor distancia administrativa** se queda.
 - En caso de **igual DA**, la ruta con **menor métrica** se queda.
 - En caso de **igual DA e igual métrica**, **ambas se quedan** en la tabla de rutas, pudiéndose producir balanceo/compartición de carga
- Redes superpuestas** (entradas solapadas en la tabla):
 - Se encamina mediante la ruta a la red más específica (valor máscara mayor).

Ejemplo:

Entradas
definidas con
enrutamiento
estático

Prioridad ?

Prioridad ?

Prioridad ?



	Destino	Siguiente
	0.0.0.0/0	R3
	192.168.32.0/24	R1
	192.168.32.0/22	R2

Red con menos
hosts primero

Ejercicio



- Un router CISCO tiene configurada la IP 10.20.30.1/24 en la interfaz FE0/0.

- Las rutas estáticas (según la sintaxis):

ip route <direc red> < mascara> <next-hop>

ip route 20.30.40.0 255.255.255.128 10.20.30.2

ip route 30.30.40.0 255.255.255.192 10.20.30.3

- Por otra parte, ha aprendido por RIP (R) y OSPF (O):

R 30.30.40.0/26 → 10.20.30.4 en 4 saltos

R 40.20.30.0/25 → 10.20.30.5 en 4 saltos

R 40.20.30.0/25 → 10.20.30.6 en 3 saltos

O 40.20.30.0/24 → 10.20.30.7 en coste 1000

O 40.20.30.0/25 → 10.20.30.8 en coste 100

¿Cómo quedaría definida su tabla de enrutamiento?

¿Por dónde enrutará al host 40.20.30.1?

¿y al 40.20.30.129?

Ejercicio



- Un router CISCO tiene configurada la IP 10.20.30.1/24 en la interfaz FE0/0.

- Las rutas estáticas (según la sintaxis):

ip route <direc red> <maska> <next-hop>

ip route 20.30.40.0 255.255.255.128 10.20.30.2

ip route 30.30.40.0 255.255.255.192 10.20.30.3

- Por otra parte, ha aprendido por RIP (R) y OSPF (O):

R 30.30.40.0/26 → 10.20.30.4 en 4 saltos

R 40.20.30.0/25 → 10.20.30.5 en 4 saltos

R 40.20.30.0/25 → 10.20.30.6 en 3 saltos

O 40.20.30.0/24 → 10.20.30.7 en coste 1000

O 40.20.30.0/25 → 10.20.30.8 en coste 100

¿Cómo quedaría definida su tabla de enrutamiento?

	DESTINO	MÁSCARA	SIGUIENTE	DA	METRIC
	C	10.20.30.0	255.255.255.0	-	0
	S	20.30.40.0	255.255.255.128	10.20.30.2	1
	S	30.30.40.0	255.255.255.192	10.20.30.3	1
X	R	30.30.40.0	255.255.255.192	10.20.30.4	120 4
X	R	40.20.30.0	255.255.255.128	10.20.30.5	120 4
X	R	40.20.30.0	255.255.255.128	10.20.30.6	120 3
	O	40.20.30.0	255.255.255.0	10.20.30.7	110 1000
	O	40.20.30.0	255.255.255.128	10.20.30.8	110 100

¿Por dónde enrutará al host 40.20.30.1?

¿y al 40.20.30.129?

Ejercicio



- Un router CISCO tiene configurada la IP 10.20.30.1/24 en la interfaz FE0/0.

- Las rutas estáticas (según la sintaxis):

ip route <direc red> <maska> <next-hop>

ip route 20.30.40.0 255.255.255.128 10.20.30.2

ip route 30.30.40.0 255.255.255.192 10.20.30.3

- Por otra parte, ha aprendido por RIP (R) y OSPF (O):

R 30.30.40.0/26 → 10.20.30.4 en 4 saltos

R 40.20.30.0/25 → 10.20.30.5 en 4 saltos

R 40.20.30.0/25 → 10.20.30.6 en 3 saltos

O 40.20.30.0/24 → 10.20.30.7 en coste 1000

O 40.20.30.0/25 → 10.20.30.8 en coste 100

¿Cómo quedaría definida su tabla de enrutamiento?

	DESTINO	MÁSCARA	SIGUIENTE	DA	METRIC
C	10.20.30.0	255.255.255.0	-	0	
S	20.30.40.0	255.255.255.128	10.20.30.2	1	
S	30.30.40.0	255.255.255.192	10.20.30.3	1	
O	40.20.30.0	255.255.255.0	10.20.30.7	110	1000
O	40.20.30.0	255.255.255.128	10.20.30.8	110	100

¿Por dónde enrutará al host 40.20.30.1?

Por 10.20.30.8

¿y al 40.20.30.129?

Por 10.20.30.7

Ejercicio



- A partir de la tabla de enrutamiento anterior, reconstruya la red:

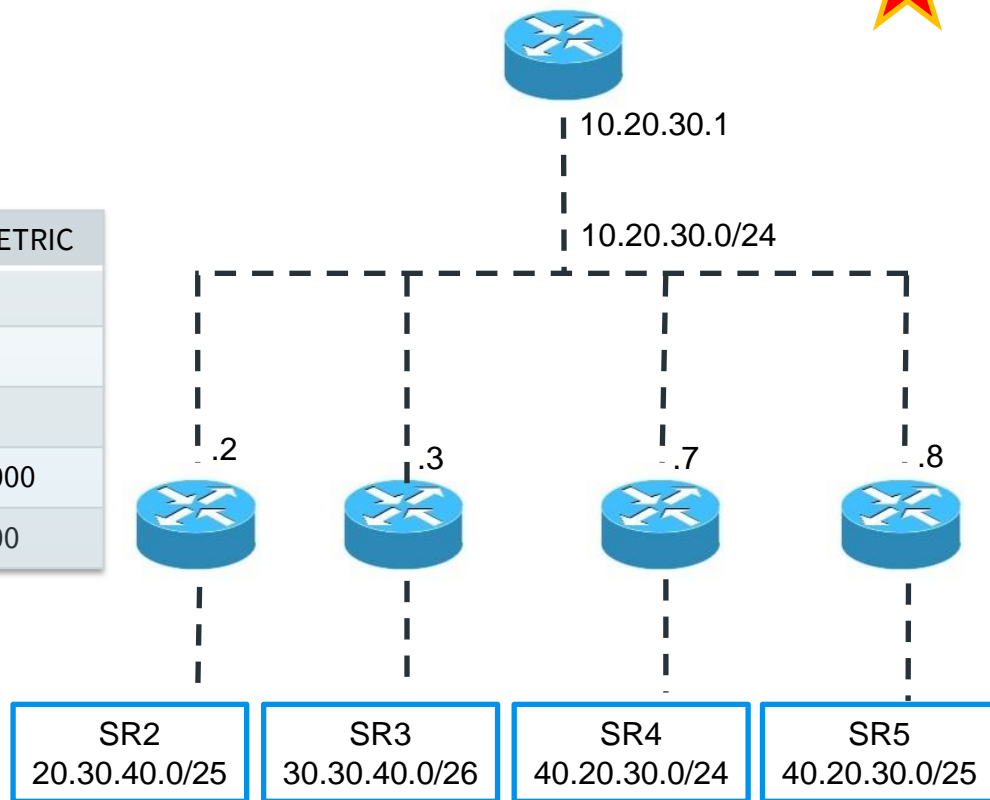
	DESTINO	MÁSCARA	SIGUIENTE	DA	METRIC
C	10.20.30.0	255.255.255.0	-	0	
S	20.30.40.0	255.255.255.128	10.20.30.2	1	
S	30.30.40.0	255.255.255.192	10.20.30.3	1	
O	40.20.30.0	255.255.255.0	10.20.30.7	110	1000
O	40.20.30.0	255.255.255.128	10.20.30.8	110	100

Ejercicio



- A partir de la tabla de enrutamiento anterior, reconstruya la red:

	DESTINO	MÁSCARA	SIGUIENTE	DA	METRIC
C	10.20.30.0	255.255.255.0	-	0	
S	20.30.40.0	255.255.255.128	10.20.30.2	1	
S	30.30.40.0	255.255.255.192	10.20.30.3	1	
O	40.20.30.0	255.255.255.0	10.20.30.7	110	1000
O	40.20.30.0	255.255.255.128	10.20.30.8	110	100



En resumen

PROTOCOLO	MÉTRICA	DIST. ADMINIST.	CARACTERÍSTICAS
RIP (v2)	Número de saltos	120	Vector distancia (Bellman-Ford). Actualizaciones periódicas (30"). Cada router envía la información a sus vecinos. Genera tráfico durante las actualizaciones. Poco coste computacional (memoria y CPU). No escalable: el máximo número de saltos es 15.
OSPF	Coste del enlace (Depende del ancho de banda o puede ser definida por el administrador)	110	Estado del enlace. Algoritmo Dijkstra (Shortest Path First). No actualizaciones periódicas. Sólo se ejecuta cuando hay un cambio topológico. Alto coste computacional durante la convergencia del algoritmo. Tabla de vecinos y topología. Escalabilidad basada en confinamiento por áreas.

Bibliografía y enlaces

- P. García-Teodoro, J.E. Díaz-Verdejo, J.M. López-Soler. Transmisión de datos y redes de computadores, 2ª Edición. Editorial Pearson, 2014.
- James F. Kurose, Keith W. Ross. Redes de computadoras. Un enfoque descendente. 7º Edición. Editorial Pearson S.A., 2017.
- Behrouz A. Forouzan. Transmisión de datos y redes de comunicaciones, 4º Edición. Editorial Mc Graw Hill 2007.
- Ernesto Ariganello. Redes Cisco: guía de estudio para la certificación CCNA Routing y Switching.
- RIP Version 2 <https://tools.ietf.org/html/rfc2453>
- OSPF Version 2 <https://tools.ietf.org/html/rfc2328>
- A Border Gateway Protocol 4 (BGP-4) <https://tools.ietf.org/html/rfc4271>

The background of the slide features a complex, light gray network pattern. It consists of numerous small circles, some of which are solid gray and others are hollow with a gray outline. These circles are interconnected by a web of thin, light gray lines, creating a dense, interconnected mesh that covers the entire slide area.

¿Alguna duda?