

# ROUTING

|  |    |
|--|----|
| <u>1.- INTERCONEXIÓN DE REDES A NIVEL 3</u> .....                              | 5  |
| <u>1.1.-FUNCIONES DEL ROUTER</u> .....   | 6  |
| <u>1.2.-NIVEL DE RED</u> .....   | 7  |
| <u>1.3.-TIPOS DE SERVICIO</u> .....  | 7  |
| <u>1.4.-ESTRUCTURA DE LA RED</u> .....   | 7  |
| <u>1.4.1.-CIRCUITO VIRTUAL</u> .....   | 7  |
| <u>1.4.2.-DATAGRAMA</u> .....  | 8  |
| <u>1.5.-SERVICIO SIN CONEXIÓN</u> .....  | 8  |
| <br>   |    |
| <u>2.- ALGORITMOS DE ENCAMINAMIENTO</u> .....                                  | 11 |
| <u>2.1.- ENCAMINAMIENTO POR VECTOR DISTANCIA</u> .....                         | 12 |
| <u>2.1.1.- HOLD DOWN</u> .....   | 18 |
| <u>2.1.2.-SPLIT HORIZON</u> .....  | 22 |
| <u>2.1.3.-TRIGGERED UPDATES (Actualizaciones Desencadenadas)</u> .....         | 23 |
| <u>2.2.- ENCAMINAMIENTO POR ESTADO DE ENLACE</u> .....                         | 25 |
| <u>2.2.1.- ALGORITMOS DE INUNDACIÓN</u> .....                                  | 26 |
| <u>2.2.2.-CALCULO DE RUTAS</u> .....   | 28 |
| <br>   |    |
| <u>3.- PROTOCOLO DE ENRUTAMIENTO RIP</u> .....                                 | 31 |
| <u>3.1.-RIP v1</u> .....   | 32 |
| <u>3.1.1.- FUNCIONAMIENTO DE RIP PARA IP</u> .....                             | 33 |
| <u>3.1.2.- FORMATO DE MENSAJES DE RIP v1</u> .....                             | 34 |
| <u>3.1.3.- PROBLEMAS DE RIP v1</u> .....                                       | 35 |
| <u>3.2.- RIP v2</u> .....  | 36 |
| <u>3.2.1.-CARACTERÍSTICAS DE RIP v2</u> .....                                  | 36 |
| <u>3.2.2.-FORMATO DE MENSAJES DE RIP v2</u> .....                              | 37 |
| <u>3.2.3.-AUTENTICACIÓN EN RIP v2</u> .....                                    | 38 |
| <u>3.3.-ENTORNOS MIXTOS DE RIP v1 Y RIP v2</u> .....                           | 39 |
| <br>   |    |
| <u>4.- Open Shortest Path First (OSPF)</u> .....                               | 41 |
| <u>4.1.- CLASIFICACIÓN DE LOS ROUTERS OSPF</u> .....                           | 42 |
| <u>4.2.- FORMATO DE MENSAJE OSPF</u> .....                                     | 43 |
| <u>4.2.1.-FORMATO DEL MENSAJE HELLO</u> .....                                  | 44 |
| <u>4.2.2.-FORMATO DEL MENSAJE DE DESCRIPCIÓN DE LA BD</u> .....                | 45 |
| <u>4.2.3.-FORMATO DEL MENSAJE DE SOLICITUD DE ESTADO DE ENLACE</u> .....       | 45 |
| <u>4.2.4.-FORMATO DEL MENSAJE DE ACTUALIZACIÓN DE ESTADO DE ENLACE</u> .....   | 46 |
| <u>4.2.5.-FORMATO DEL MENSAJE DE ACUSE DE RECIBO DE ESTADO DE ENLACE</u> ..... | 46 |
| <u>4.2.6.-INFORMACIÓN DE ESTADO DE ENLACE</u> .....                            | 47 |
| <u>4.3.- MEJORAS DE OSPF FRENTE A RIP</u> .....                                | 47 |
| <br>   |    |
| <u>5.- MULTICASTING IP ROUTING</u> .....                                       | 49 |
| <u>5.1.- INTRODUCCIÓN</u> .....  | 50 |
| <u>5.2.- FUNCIONAMIENTO DEL MULTICAST</u> .....                                | 50 |
| <u>5.2.1.- DIRECCIONAMIENTO CLASE D</u> .....                                  | 52 |
| <u>5.3.- PROTOCOLO IGMP</u> .....  | 52 |

|  |    |
|--|----|
| <a href="#"><u>5.3.1.- MENSAJES IGMP</u></a> .....           | 54 |
| <a href="#"><u>5.4.-PROTOCOLOS DE ROUTING</u></a> .....      | 56 |
| <a href="#"><u>5.4.1.-MODO DENSO:</u></a> .....              | 56 |
| <a href="#"><u>5.4.2.- MODO DISPERSO:</u></a> .....          | 57 |
| <br>   |    |
| <a href="#"><u>6.- PROTOCOLOS EXTERIORES</u></a> .....       | 59 |
| <a href="#"><u>6.1.-SISTEMAS AUTÓNOMOS</u></a> .....         | 60 |
| <a href="#"><u>6.2.- PROTOCOLO BGP</u></a> .....             | 61 |
| <a href="#"><u>6.2.1.- FORMATO DE LOS PAQUETES</u></a> ..... | 62 |

# INTERCONEXIÓN DE REDES A N3

## 1.- INTERCONEXIÓN DE REDES A NIVEL 3

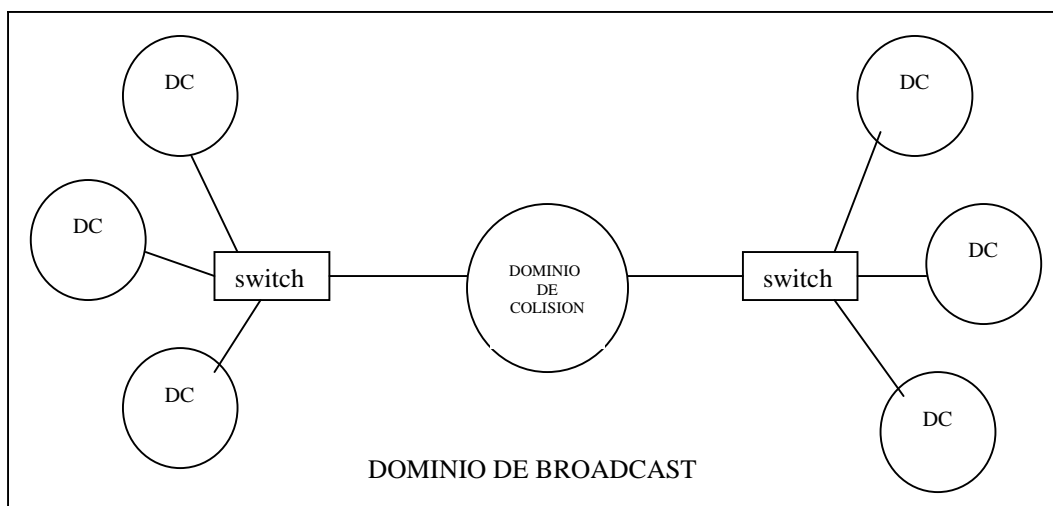
Hasta ahora habíamos supuesto:

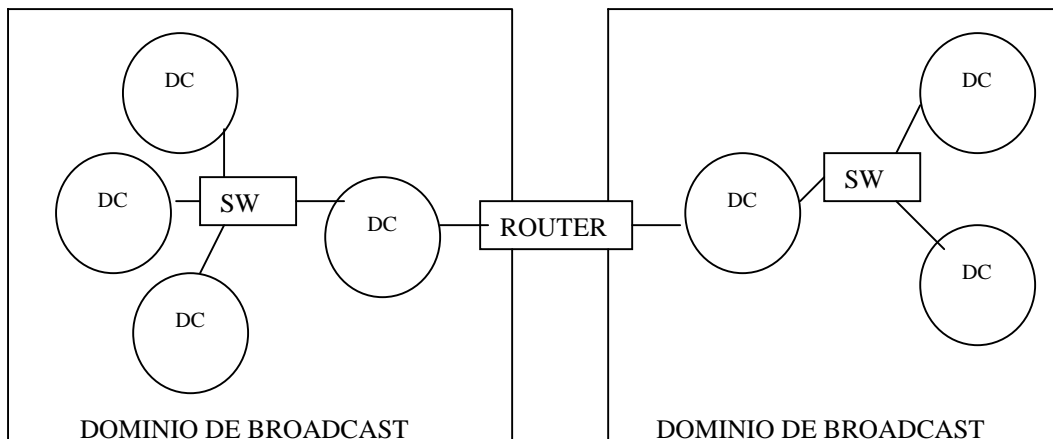
- Existía una sola red homogénea
- Utilización de los mismos protocolos.

Un conjunto de redes consta de múltiples redes separadas e interconectada mediante routers. Los datos se intercambian en paquetes entre origen y destino a través de un camino que implica múltiples redes y dispositivos de encaminamiento.

¿Cuáles son las razones para usar interconexión a N3?

- Existen problemas que no se pueden resolver a N2 o requieren muchas modificaciones. Ej.: diferentes MTUs.
- Cuando se busque segmentar la red con las metas de:
  - Limitar tormentas de Broadcast
  - Restringir acceso a servicios
  - Filtrado selectivo
  - Selección de rutas por donde viajará la información
  - Balanceo de carga
  - Máxima vida de una información en la red
- Compatibilizar sistemas que no lo son a N3, como IP, IPX, X.25, etc..





Por el contrario: la gestión de los routers es compleja, se debe conocer la arquitectura de la red y los parámetros relacionados con los protocolos de encaminamiento.

A mayor coste → Dispositivos más complejos.

### 1.1.-FUNCIONES DEL ROUTER

- Proporcionar un enlace entre 2 o más redes  
 - Encaminar y entregar datos entre sistemas finales operando en diferentes redes de forma transparente:

- Compatibilizar esquemas de direccionamiento si no existe un método global.
- Ocultar aspectos específicos de enlaces WAN

- Gestionar tamaños distintos de paquete
- Distintos interfaces de red
- Control de flujo

El protocolo más comúnmente utilizado para la interconexión de redes es IP.

## 1.2.-NIVEL DE RED

Es el responsable de llevar los paquetes de origen a destino, por tanto debe conocer la topología de la subred de comunicaciones y escoger las trayectorias adecuadas.

Debe resolver el problema de manejar distintas tecnologías cuando origen y destino están en redes distintas.

El N3 proporciona servicio al N4 y normalmente define el límite de la subred. La interface con el N4 debe estar bien definida:

1. Los servicios deben ser independientes de la tecnología de la subred
2. El N4 debe estar aislado de la cantidad, tipo y topología de las redes subyacentes
3. Las direcciones de red deben seguir un plan de numeración uniforme.

## 1.3.-TIPOS DE SERVICIO

El N3 puede proporcionar dos tipos de servicio al nivel superior, diferenciándose en donde reside la complejidad:

- Con Conexión → Complejidad en SUBRED
- Sin Conexión → Complejidad en HOSTS

A favor del servicio con conexión se puede decir que determinados servicios (audio/vídeo) es más fácil proporcionarlos mediante este tipo de servicio.

A favor del servicio sin conexión se dice que la red no debe sobrecargarse con características que pueden ser obsoletas en pocos años, además los hosts son cada vez más potentes y baratos.

## 1.4.-ESTRUCTURA DE LA RED

Básicamente existen dos posibilidades, una que utiliza conexiones y otra que no:

- Circuito Virtual
- Datagrama

### *1.4.1.-CIRCUITO VIRTUAL*

- Precisa establecimiento de conexión, en este momento se escoge el camino y se reservan los recursos para la comunicación.
- La red garantiza unos parámetros de calidad cuando se establece la llamada.
- Si el camino deja de estar disponible o no se pueden cumplir los parámetros de calidad se libera la comunicación.
- La ruta se utiliza para todo el tráfico que fluye por la conexión.
- Cuando se libera la conexión deja de existir el circuito virtual.

#### 1.4.2.-DATAGRAMA

- No se determina rutas por adelantado.
- Cada paquete se encamina de forma independiente, debe incorporar dirección fuente y destino.
- Dos paquetes consecutivos pueden llevar rutas distintas.
- La red puede perder, duplicar, retardar la información.

#### 1.5.-SERVICIO SIN CONEXIÓN

Es el elegido habitualmente para interconexión pues se trata cada paquete de forma independiente para que siga el mejor camino disponible. Hace uso al máximo de la capacidad de la red.

Aspectos como ordenación de la información, pérdida de paquetes, duplicados, serán tratados por el nivel superior.

Las partes en que se puede dividir un servicio de este tipo son:

- **Servicio Básico sin Conexión**
- **Relación con los Vecinos**
- **Routing**

##### ***Servicio Básico sin Conexión:***

- Comprende:
  - formato de los paquetes
  - mensajes de error (ICMP)
  - avisos de la red al nodo

ISO – CLNP (Connection Less...)  
TCP/IP – IP, ICMP

##### ***Relación con los Vecinos:***

- Se precisa saber quien está en nuestra LAN para intercambiar información de estado, caminos disponibles, etc..

ISO – ES/IS (End System/Intermediate System)  
TCP/IP – ARP, ICMP

##### ***Routing:***

- Se definen los protocolos y algoritmos que los routers utilizan para elegir caminos.

- Se subdivide la red global en subconjuntos para facilitar el problema de routing en grandes redes.
- Los subconjuntos pueden utilizar protocolos internos distintos y se comunican mediante un protocolo común.

“SISTEMAS AUTÓNOMOS” → TCP/IP

Dominios de Encaminamiento → ISO

- Cada subconjunto es gestionado por una única organización.
- Los protocolos de encaminamiento dentro de cada subconjunto se llaman:

ISO (Intradomain Routing Protocols)

IS – IS (Intermediate System -- IS)

TCP/IP (Internal Gateway Protocols)

RIP → Routing Information Protocol

OSPF → Open Shortest Path First

- Para conectar los diferentes subconjuntos:

ISO (Interdomain Routing Protocols)

IDRP

TCP/IP (External Gateway Protocols)

EGP → Exterior Gateway Protocol

BGP → Border Gateway Protocol



# ALGORITMOS DE ENCAMINAMIENTO

## 2.- ALGORITMOS DE ENCAMINAMIENTO

Básicamente existen dos tipos de algoritmos de encaminamiento que operan en entornos distribuidos:

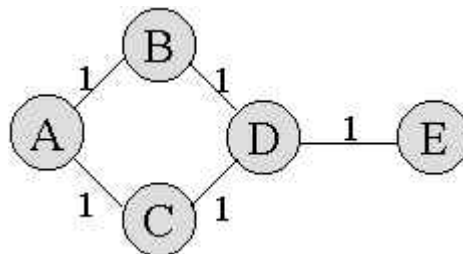
- **Vector Distancia**
- **Estado de Enlace**

Todos los protocolos más utilizados de encaminamiento en redes se basan en uno u otro (RIP-OSPF).

### 2.1.- ENCAMINAMIENTO POR VECTOR DISTANCIA

Fue el algoritmo original utilizado en ARPANET. Se le conoce también como algoritmo de Bellman-Ford, y es la base de RIP y RIP II, utilizado para calcular las rutas más cortas.

Cada nodo informa a sus nodos vecinos de todas las distancias conocidas por él, mediante vectores de distancias (de longitud variable según los nodos conocidos). El vector de distancias es un vector de longitud variable que contiene un par (nodo:distancia al nodo) por cada nodo conocido por el nodo que lo envía, por ejemplo (A:0;B:1;D:1) que dice que el nodo que lo manda dista "0" de A, "1" de B y "1" de D, y de los demás no sabe nada (ésta es la forma en la que un nodo dice lo que sabe en cada momento). El nodo solo conoce la distancia a los distintos nodos de la red pero no conoce la topología. Estos vectores de distancia se envían periódicamente y cada vez que varíe su vector de distancias. Veamos el siguiente ejemplo:



El vector de distancias de A sería:



Este vector de distancias de A llega al nodo B, el cual lo utiliza para actualizar el suyo:

$$V_{DB} = (B:0, A:1, D:1) \longrightarrow V_{DB} = (B:0, A:1, C:2, D:1)$$

Este VDB pasa al nodo A, el cual actualiza el suyo, etc.

Estos protocolos requieren que cada nodo calcule por separado la mejor ruta (enlace de salida) para cada destino marcado. Tras seleccionar la mejor ruta, un router envía vectores de distancia a sus vecinos, notificándoles la accesibilidad de cada destino y las correspondientes métricas asociadas con la ruta que se ha seleccionado para alcanzarlo. Paralelamente sus vecinos calculan también la mejor ruta para alcanzar cada destino disponible y luego notifican a sus vecinos (junto con las métricas asociadas) la que ellos han seleccionado para alcanzar su destino. A partir de los mensajes recibidos de los vecinos detallando el destino y las métricas asociadas seleccionadas, el router podría determinar que existe una ruta mejor a través de un vecino alternativo. El router notificara nuevamente a sus vecinos sus rutas seleccionadas para alcanzar cada destino. Este ciclo continua hasta que todos los routers hallan llegado a un común acuerdo sobre las mejores rutas para cada destino prefijado.

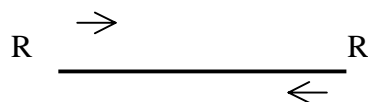
Con todos los vectores recibidos, cada nodo monta su tabla de encaminamiento ya que al final conoce qué nodo vecino tiene la menor distancia al destino del paquete, pues se lo han dicho con el vector de distancias.

La tabla de encaminamiento de cada nodo contiene un registro para cada router de la subred.

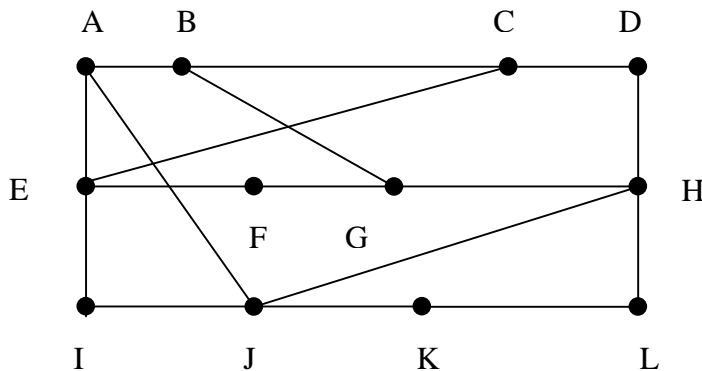
El envío de vectores de distancia entre nodos tiene lugar en el plano de control.

Si no se sabe a donde enviar un paquete, se descarta.

Nota: si la métrica es la distancia se cuenta 1 salto de router a router. Si es el retardo se envían paquetes de eco.



**Ejemplo:** Utilizando como métrica el retardo. El router conoce el retardo a sus vecinos. Cada  $T_{msg}$  cada router envía a sus vecinos una lista de retardos estimados y recibe lo propio de éstos. Con la información recibida cada router recalcula su tabla de encaminamiento.



Métricas estimadas por J:

JA \_\_\_\_\_ 8  
 JI \_\_\_\_\_ 10  
 JH \_\_\_\_\_ 12  
 JK \_\_\_\_\_ 6

Los datos que tenemos para obtener la tabla de encaminamiento final son los vectores de retardo recibidos de los vecinos de J mostrados en la siguiente tabla:

|   | A  | I  | H  | K  |
|---|----|----|----|----|
| A | 0  | 24 | 20 | 21 |
| B | 12 | 36 | 31 | 28 |
| C | 25 | 18 | 19 | 36 |
| D | 40 | 27 | 8  | 24 |
| E | 14 | 7  | 30 | 22 |
| F | 23 | 20 | 19 | 40 |
| G | 18 | 31 | 6  | 31 |
| H | 17 | 20 | 0  | 19 |
| I | 21 | 0  | 14 | 22 |
| J | 9  | 11 | 7  | 10 |
| k | 24 | 22 | 22 | 0  |
| L | 29 | 33 | 9  | 9  |

Ahora J procede a calcular su tabla de encaminamiento:

| Destino | Distancia | Ruta |
|---------|-----------|------|
| A       | 8         | A    |
| B       | 20        | A    |
| C       | 28        | I    |
| D       | 20        | H    |
| E       | 17        | I    |
| F       | 30        | I    |
| G       | 18        | H    |
| H       | 7         | H    |
| I       | 11        | I    |
| J       | 0         | J    |
| K       | 6         | K    |

$$\begin{aligned} JA + AG &= 8+18 = 26 \\ JI + IG &= 10+31 = 41 \\ JH + HG &= 12+6 = 18 \\ JK + KG &= 6+31 = 37 \end{aligned}$$

Estas son los distintos caminos para alcanzar G, y como podemos observar nos quedamos con la ruta que pasa por H por ser la de métrica más corta.

|   |    |   |
|---|----|---|
| L | 15 | K |
|---|----|---|

Ventajas del método:

- Muy sencillo.
- Muy robusto (gracias al envío periódico de información)
- Consumo de memoria bajo: cada nodo sólo ha de almacenar distancias con el resto de los nodos.

*Nota:* los nodos no tienen información topológica de la red completa, es decir, pueden conocer la distancia a nodos lejanos, pero no donde están.

Inconvenientes del método:

- Convergencia lenta (los vectores de distancia tardan en estabilizarse).
- Pueden aparecer bucles.
- Adaptabilidad a los cambios baja, ya que sólo sabe a quién tiene que reenviar un paquete, pero no tiene información de la topología.
- Consumo alto de capacidad: se transmiten vectores cuyo tamaño es del orden del número de nodos de la red pues cada nodo comunica a su vecino todas las distancias que conoce .

*Nota:* los bucles (situación que se da cuando los paquetes pasan más de una vez por un nodo) ocurren porque los criterios de los nodos no son coherentes, generalmente debido a que los criterios de encaminamiento o no han convergido después de un cambio en la ruta de un paquete; cuando por cualquier causa un paquete sufre un cambio de encaminamiento, la red tarda en adaptarse a ese cambio pues la noticia del cambio tiene que llegar a todos los nodos. Es en ese transitorio cuando se pueden dar los bucles, ya que unos nodos se han adaptado y otros no. El objetivo de los algoritmos de encaminamiento es detener el curso de los paquetes antes de que se produzcan bucles. Esto es importante sobre todo cuando se envían los paquetes por varias rutas simultáneamente (técnicas de inundación, etc...).

El problema clásico de convergencia por vector de distancia se denomina el problema de contar hasta el infinito y es consecuencia directa del esquema de anuncios asincrónicos. Cuando los enrutadores de RIP para IP agregan rutas a sus tablas de enrutamiento, dependiendo de las rutas anunciadas por otros enrutadores, sólo conservan en la tabla de enrutamiento la mejor ruta y sólo actualizan una ruta de costo inferior con una ruta de costo superior si la anuncia el mismo origen como la ruta de costo inferior actual. En determinadas situaciones, como se ilustra en las figuras 1 a 5, esto provoca el problema de contar hasta el infinito.

Se supone que en la red de la figura 1 se ha producido convergencia. Por motivos de claridad, no se incluyen los anuncios enviados por el enrutador 1 en la red 1 y el enrutador 2 en la red 3.

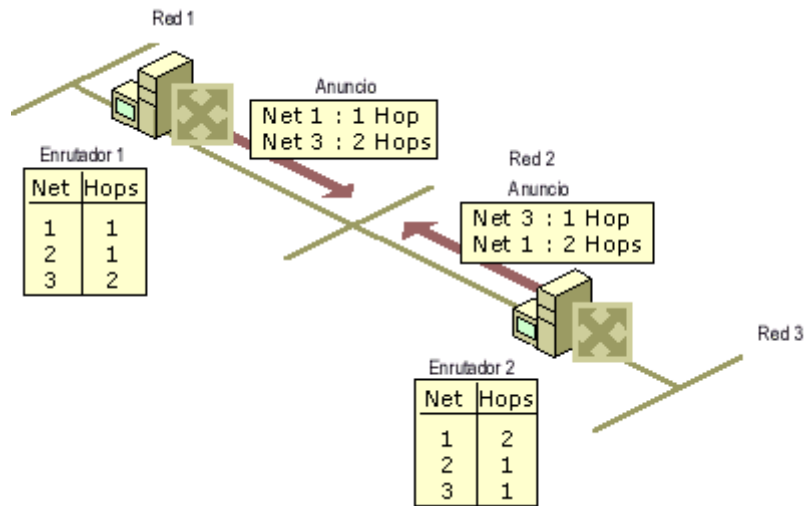


Figura 1 Red en la que se ha producido convergencia

Supongamos ahora que se produce un error en el vínculo del enrutador 2 a la red 3 y el enrutador 2 detecta dicho error. Tal como se aprecia en la figura 2, el enrutador 2 cambia el número de saltos para la ruta a la red 3 para indicar que no es accesible, que se encuentra a una distancia infinita. En el caso de RIP para IP, el infinito es 16.

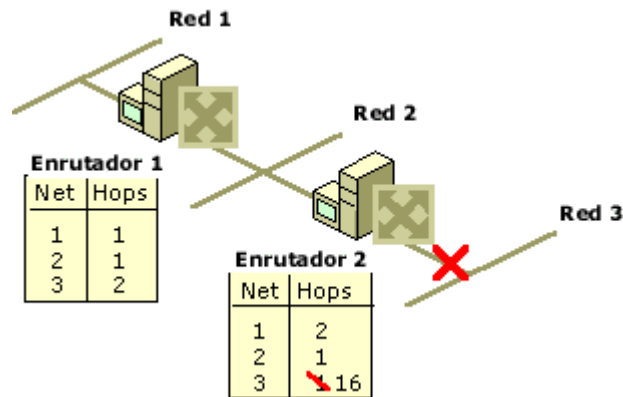


Figura 2 Error en el vínculo a la red 3

Sin embargo, antes de que el enrutador 2 pueda anunciar el nuevo número de saltos a la red en un anuncio programado, recibe un anuncio del enrutador 1. Dicho anuncio contiene una ruta a la red 3, que está a 2 saltos. Como la distancia de 2 saltos es una ruta mejor que 16 saltos, el enrutador 2 actualiza su entrada de la tabla de enrutamiento para la red 3 y cambia de 16 a 3 saltos, como se muestra en la figura 3.

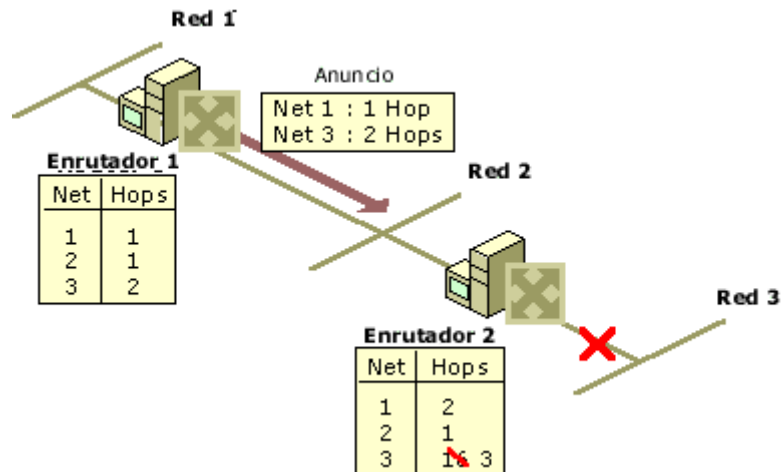


Figura 3 El enrutador 2 después de recibir el anuncio del enrutador 1

Quando el enrutador 2 anuncia sus rutas nuevas, el enrutador 1 advierte que la red 3 se encuentra a una distancia de 3 saltos a través del enrutador 2. Como la ruta a la red 3 en el enrutador 1 se obtuvo originalmente del enrutador 2, el enrutador 1 actualiza a 4 saltos su ruta a la red 3. (Consulte la figura 4.)

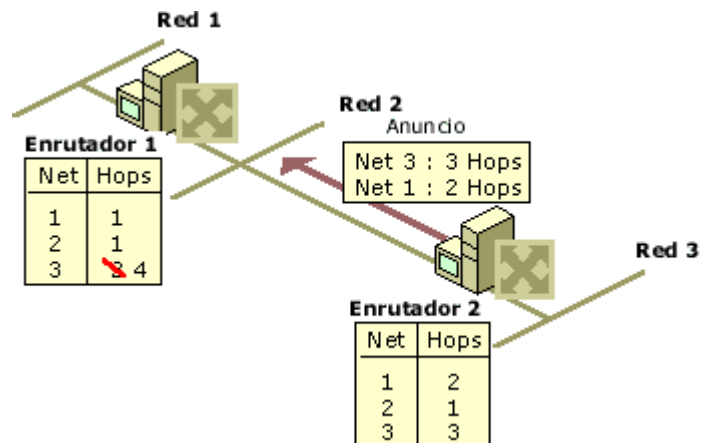


Figura 4 El enrutador 1 después de recibir el anuncio del enrutador 2

Quando el enrutador 1 anuncia sus rutas nuevas, el enrutador 2 advierte que la red 3 se encuentra a una distancia de 4 saltos a través del enrutador 1. Como la ruta a la red 3 en el enrutador 2 se obtuvo originalmente del enrutador 1, el enrutador 2 actualiza a 5 saltos su ruta a la red 3. (Consulte la figura 5.)

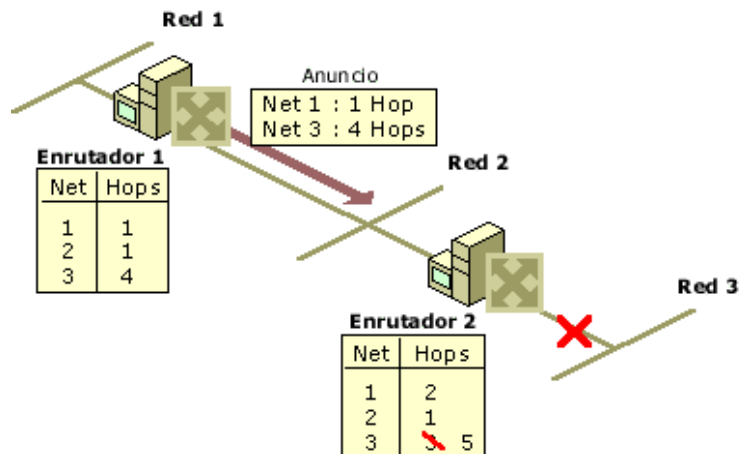


Figura 5 El enrutador 2 después de recibir otro anuncio del enrutador 1

Los dos enrutadores continúan anunciando rutas a la red 3 con un número de saltos cada vez mayor hasta que se llega al infinito (16). A continuación, la red 3 se considera no accesible y, finalmente, se agota el tiempo de espera de la ruta a la red 3 de la tabla de enrutamiento. Esta situación se denomina problema de contar hasta el infinito.

El problema de contar hasta el infinito es uno de los motivos por los que el número máximo de saltos de las redes RIP para IP está definido como 15 (16 significa no accesible). Si el número máximo de saltos fuera mayor, el tiempo de convergencia sería mayor cuando se produjera la cuenta hasta el infinito. Tenga en cuenta también que durante la cuenta hasta el infinito del ejemplo anterior, la ruta del enrutador 1 a la red 3 se realiza a través del enrutador 2. La ruta desde el enrutador 2 a la red 3 se efectúa a través del enrutador 1. Hay un bucle de enrutamiento entre el enrutador 1 y el enrutador 2 para la red 3 mientras dura el problema de contar hasta el infinito.

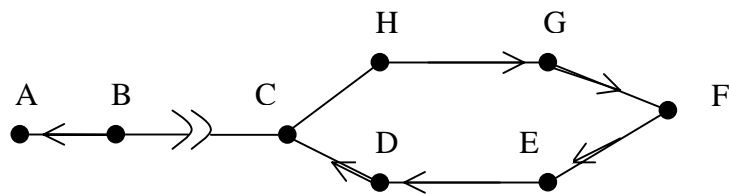
Se han propuesto varias soluciones para arreglar el problema de la 'convergencia lenta' en encaminamientos por vectores distancia:

- Hold Down
- Split Horizon
- Triggered Updates

### 2.1.1.- HOLD DOWN

Cuando cae un enlace, se espera un tiempo antes de conmutar hacia otro camino, este tiempo se da para que los routers vecinos conozcan la caída de la línea.

Ejemplo:



→ Camino óptimo desde cada encaminador hacia A (métrica : retardo)

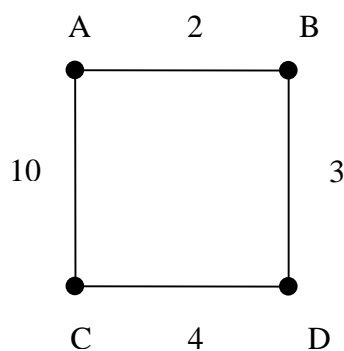
Supongamos que existen problemas entre B y C, ¿qué ocurriría?

C entraría en HOLD DOWN, D se da cuenta de que por C no puede enviar datos hacia A y entraría también en HOLD DOWN, así sucesivamente hasta que todos los routers se den cuenta de que el enlace entre B y C ha caído.

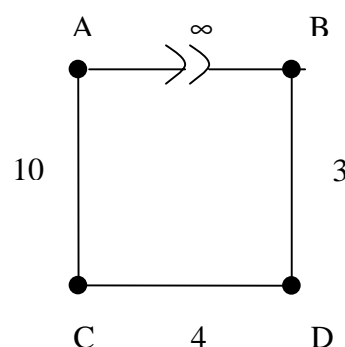
Tendremos que tener especial cuidado en elegir el tiempo de HOLD DOWN ya que si no se elige bien puede ocurrir que cuando le llega a H la noticia, C ya no esté en HOLD DOWN y encuentre H un camino por C y C por D y así se formaría un bucle infinito y funcionaría mal.

No es conveniente que el tiempo de HOLD DOWN sea muy alto porque la reacción de la red ante cambios positivos sería lenta.

Ejemplo: Mostrar la evolución de las tablas en el intento de adaptarse a la nueva realidad de la subred. Suponer un tiempo de Hold Down = 3



| Dest | IMP A<br>Coste por | IMP B<br>Coste por | IMP C<br>Coste por | IMP D<br>Coste por |
|------|--------------------|--------------------|--------------------|--------------------|
| A    | 0 --               | 2 A                | 9 D                | 5 B                |
| B    | 2 B                | 0 --               | 7 D                | 3 B                |
| C    | 9 B                | 7 D                | 0 --               | 4 C                |
| D    | 5 B                | 3 D                | 4 D                | 0 --               |



En el primer intercambio A y B entran en Hold Down ya que el enlace que los une ha caído.

| DEST | A          |
|------|------------|
| A    | ----       |
| B    | $\infty$ B |
| C    | $\infty$ B |
| D    | $\infty$ B |

HD1

| DEST | B          |
|------|------------|
| A    | $\infty$ A |
| B    | ----       |
| C    | 7 D        |
| D    | 3 D        |

HD1

| DEST | C    |
|------|------|
| A    | 9 D  |
| B    | 7 D  |
| C    | ---- |
| D    | 4 D  |

| DEST | D    |
|------|------|
| A    | 5 B  |
| B    | 3 B  |
| C    | 4 C  |
| D    | ---- |

En el segundo intercambio D se entera que enlace ha caído y entra también en Hold Down.

| DEST | A          |
|------|------------|
| A    | ----       |
| B    | $\infty$ B |
| C    | $\infty$ B |
| D    | $\infty$ B |

HD2

| DEST | B          |
|------|------------|
| A    | $\infty$ A |
| B    | ----       |
| C    | 7 D        |
| D    | 3 D        |

HD2

| DEST | C    |
|------|------|
| A    | 9 D  |
| B    | 7 D  |
| C    | ---- |
| D    | 4 D  |

| DEST | D          |
|------|------------|
| A    | $\infty$ B |
| B    | 3 B        |
| C    | 4 C        |
| D    | ----       |

HD1

En el tercer intercambio C se da cuenta que ha caído el enlace, entrando en Hold Down.

| DEST | A          |
|------|------------|
| A    | ----       |
| B    | $\infty$ B |
| C    | $\infty$ B |
| D    | $\infty$ B |

HD3

| DEST | B          |
|------|------------|
| A    | $\infty$ A |
| B    | ----       |
| C    | 7 D        |
| D    | 3 D        |

HD3

| DEST | C          |
|------|------------|
| A    | $\infty$ D |
| B    | 7 D        |
| C    | ----       |
| D    | 4 D        |

HD1

| DEST | D          |
|------|------------|
| A    | $\infty$ B |
| B    | 3 B        |
| C    | 4 C        |
| D    | ----       |

HD2

Al finalizar el tiempo de Hold Down en A y B, A busca un nuevo camino a través de C, no siendo así en B ya que no sabe que por C puede alcanzar a A.

| DEST | A    |
|------|------|
| A    | ---- |
| B    | 17 C |
| C    | 10 C |
| D    | 14 C |

| DEST | B          |
|------|------------|
| A    | $\infty$ D |
| B    | ----       |
| C    | 7 D        |
| D    | 3 D        |

| DEST | C          |
|------|------------|
| A    | $\infty$ D |
| B    | 7 D        |
| C    | ----       |
| D    | 4 D        |

HD2

| DEST | D          |
|------|------------|
| A    | $\infty$ B |
| B    | 3 B        |
| C    | 4 C        |
| D    | ----       |

HD3

D deja de estar en Hold Down, por lo tanto buscará un camino alternativo para alcanzar A, pero no encuentra ninguno ya que en este intercambio todavía se desconoce que puede hacerlo por C, que está en Hold Down.

| DEST | A    |
|------|------|
| A    | ---- |
| B    | 17 C |
| C    | 10 C |
| D    | 14 C |

| DEST | B          |
|------|------------|
| A    | $\infty$ D |
| B    | ----       |
| C    | 7 D        |
| D    | 3 D        |

| DEST | C          |
|------|------------|
| A    | $\infty$ D |
| B    | 7 D        |
| C    | ----       |
| D    | 4 D        |

| DEST | D            |
|------|--------------|
| A    | $\infty$ B,C |
| B    | 3 B          |
| C    | 4 C          |
| D    | ----         |

### HD3

El nodo C deja de estar en Hold Down, reconociendo un camino alternativo hacia A.

| DEST | A    |
|------|------|
| A    | ---- |
| B    | 17 C |
| C    | 10 C |
| D    | 14 C |

| DEST | B          |
|------|------------|
| A    | $\infty$ D |
| B    | ----       |
| C    | 7 D        |
| D    | 3 D        |

| DEST | C    |
|------|------|
| A    | 10 A |
| B    | 7 D  |
| C    | ---- |
| D    | 4 D  |

| DEST | D          |
|------|------------|
| A    | $\infty$ B |
| B    | 3 B        |
| C    | 4 C        |
| D    | ----       |

En este intercambio D ya sabe que puede alcanzar A a través de C.

| DEST | A    |
|------|------|
| A    | ---- |
| B    | 17 C |
| C    | 10 C |
| D    | 14 C |

| DEST | B          |
|------|------------|
| A    | $\infty$ D |
| B    | ----       |
| C    | 7 D        |
| D    | 3 D        |

| DEST | C    |
|------|------|
| A    | 10 A |
| B    | 7 D  |
| C    | ---- |
| D    | 4 D  |

| DEST | D    |
|------|------|
| A    | 14 C |
| B    | 3 B  |
| C    | 4 C  |
| D    | ---- |

En este último intercambio, B y por tanto todos los nodos tienen un camino alternativo para llegar a cualquier nodo de la red sin utilizar el enlace que ha caído.

| DEST | A    |
|------|------|
| A    | ---- |
| B    | 17 C |
| C    | 10 C |
| D    | 14 C |

| DEST | B    |
|------|------|
| A    | 17 D |
| B    | ---- |
| C    | 7 D  |
| D    | 3 D  |

| DEST | C    |
|------|------|
| A    | 10 A |
| B    | 7 D  |
| C    | ---- |
| D    | 4 D  |

| DEST | D    |
|------|------|
| A    | 14 B |
| B    | 3 B  |
| C    | 4 C  |
| D    | ---- |

## 2.1.2.-SPLIT HORIZON

El Split Horizon ayuda a reducir el tiempo de convergencia, ya que no permite a los enrutadores anunciar redes en la dirección desde la que se aprendieron dichas redes. La única información enviada en los anuncios de vector distancia es la de aquellas redes que se encuentran más allá del enrutador vecino en la dirección opuesta. No se incluyen las redes aprendidas del enrutador vecino.

El Split Horizon elimina la cuenta hasta el infinito durante la convergencia en redes con ruta de acceso única, y reduce las posibilidades de que aparezca el problema de contar hasta el infinito en las redes con múltiples rutas de acceso. La figura 6 muestra cómo el Split Horizon impide que el enrutador anuncie rutas en la dirección en la que las aprendió.

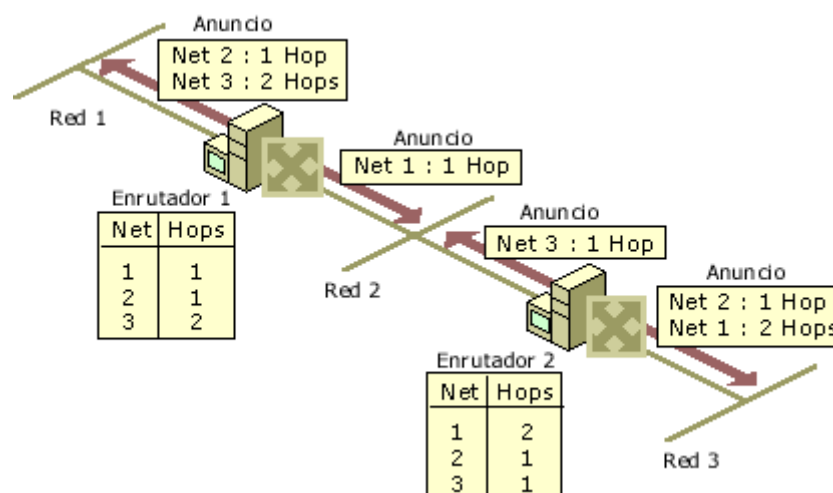


Figura 6 Horizonte dividido

- **SPLIT HORIZON CON RUTAS INALCANZABLES**

El Split Horizon con rutas inalcanzables se diferencia del Split Horizon simple en que anuncia todas las redes. Sin embargo, las redes aprendidas en una determinada dirección se anuncian con un número de saltos de 16, lo que indica que la red es inalcanzable. En una red con ruta de acceso única, el Split Horizon con rutas inalcanzables no tiene más ventajas que el Split Horizon simple. Sin embargo, en una red con rutas de acceso múltiples, el Split Horizon con rutas inalcanzables reduce en gran medida la cuenta hasta el infinito. La cuenta hasta el infinito puede seguir produciéndose en un sistema de redes con rutas de acceso múltiples porque las rutas a las redes pueden aprenderse de varios orígenes.

En la figura 7, el Split Horizon con rutas inalcanzables anuncia rutas aprendidas como inalcanzables en la dirección en la que se aprendieron. El Split Horizon con rutas inalcanzables tiene la desventaja de que se produce

una sobrecarga de mensajes de vector distancia adicionales, ya que se anuncian todas las redes.

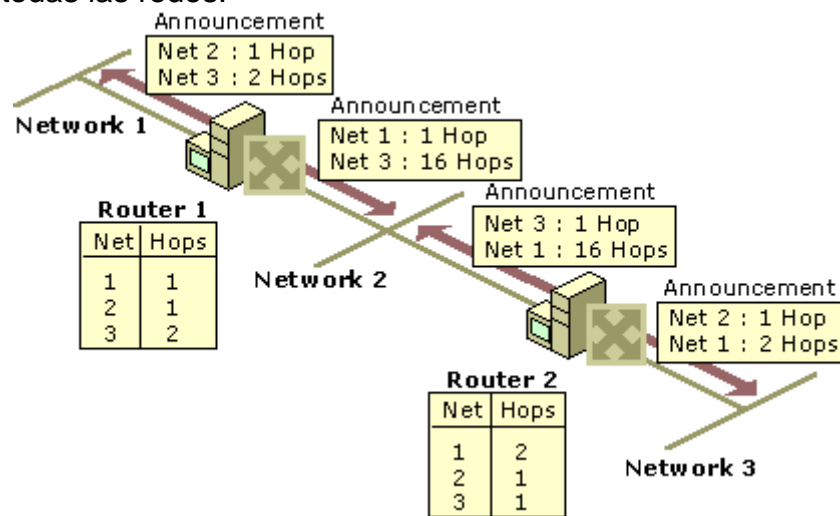


Figura 7 Horizonte dividido con rutas inalcanzables

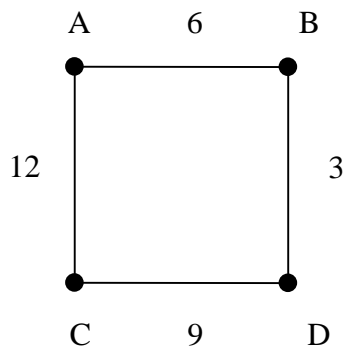
Como vemos el problema de conteo a infinito persiste.

### 2.1.3.-TRIGGERED UPDATES (Actualizaciones Desencadenadas)

El Triggered Updates permite que un enrutador anuncie los cambios en los valores de medida casi inmediatamente, en vez de esperar al siguiente anuncio periódico. El desencadenador es un cambio en la medida de una entrada de la tabla de enrutamiento. Por ejemplo, las redes que se convierten en no disponibles pueden anunciarse con un número de saltos de 16 a través del Triggered Updates. Hay que tener en cuenta que la actualización se envía *casi inmediatamente*, ya que hay un período de espera especificado normalmente en el enrutador. Si todos los enrutadores enviaran inmediatamente actualizaciones desencadenadas, cada actualización desencadenada podría provocar una cascada de tráfico de difusión en la red IP.

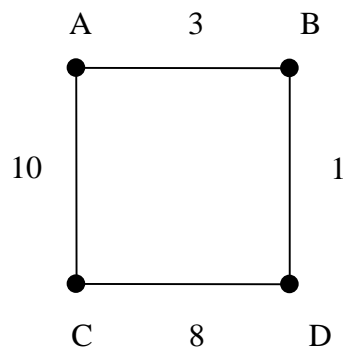
El Triggered Updates mejora el tiempo de convergencia de las redes RIP, pero a costa de tráfico de difusión adicional a medida que se propagan las actualizaciones desencadenadas.

Ejemplo: La subred de la figura utiliza un algoritmo de encaminamiento distribuido. En un momento dado las tablas son:



| Dest | IMP A<br>Coste por |    | IMP B<br>Coste por |    | IMP C<br>Coste por |    | IMP D<br>Coste por |    |
|------|--------------------|----|--------------------|----|--------------------|----|--------------------|----|
| A    | 0                  | -- | 6                  | A  | 12                 | A  | 9                  | B  |
| B    | 6                  | B  | 0                  | -- | 12                 | A  | 3                  | B  |
| C    | 12                 | C  | 12                 | D  | 0                  | -- | 9                  | C  |
| D    | 9                  | B  | 3                  | D  | 9                  | D  | 0                  | -- |

Mostrar la evolución de las tablas en el intento de adaptarse a la nueva realidad de la subred



| 1º INTERCAMBIO |                    |    |                    |    | 2º INTERCAMBIO     |    |                    |   |      |                    |    |                    |    |                    |    |                    |    |
|----------------|--------------------|----|--------------------|----|--------------------|----|--------------------|---|------|--------------------|----|--------------------|----|--------------------|----|--------------------|----|
| Dest           | IMP A<br>Coste por |    | IMP B<br>Coste por |    | IMP C<br>Coste por |    | IMP D<br>Coste por |   | Dest | IMP A<br>Coste por |    | IMP B<br>Coste por |    | IMP C<br>Coste por |    | IMP D<br>Coste Por |    |
| A              | 0                  | -- | 3                  | A  | 10                 | A  | 7                  | B | A    | 0                  | -- | 3                  | A  | 10                 | A  | 4                  | B  |
| B              | 3                  | B  | 0                  | -- | 11                 | D  | 1                  | B | B    | 3                  | B  | 0                  | -- | 9                  | D  | 1                  | B  |
| C              | 10                 | C  | 10                 | D  | 0                  | -- | 8                  | C | C    | 10                 | C  | 9                  | D  | 0                  | -- | 8                  | C  |
| D              | 6                  | B  | 1                  | D  | 8                  | D  | 0                  | C | D    | 4                  | B  | 1                  | D  | 8                  | D  | 0                  | -- |

## 2.2.- ENCAMINAMIENTO POR ESTADO DE ENLACE

Los algoritmos vector distancia tienen una convergencia lenta, incluso con Split Horizon, lo que hace que haya nodos que trabajen con información que ya no es cierta o que envíen información por rutas que ya no existen.

Con estado de enlace, los routers requieren información de la topología de red completa, lo que quiere decir que los mensajes serán más grandes a medida que haya más redes.

Para tener constancia de la topología de la red, cada nodo debe aprender la dirección y el coste a cada nodo vecino. Se debe construir un paquete (LSP, Link State Packet) que contiene una lista de nombres y el coste de alcanzarlos. El LSP se transmite a los demás routers, y cada router se queda con el LSP más reciente del otro router. Una vez conocido el mapa de la topología, cada router calcula el camino a cada destino.

Como los mensajes de actualización se envían a todos los routers, cada router construye su propia tabla de ruteo a partir de la misma información, reduciendo de esta forma la posibilidad de tener ciclos en el ruteo. En la práctica sin embargo estos protocolos no están completamente libres de ciclos de ruteo. En grandes redes los mensajes de actualización pueden propagarse de forma lenta, debido a que en el encaminamiento por estado de enlace todos los routers deben tener constancia de la topología de la red completa, y por lo tanto los routers obtendrán las actualizaciones en distintos momentos, esto puede provocar que dos routers calculen sus tablas de ruteo basándose en información distinta.

Para asegurar que cada router construya sus tablas con la misma información, los mensajes de actualización deben ser transportados en forma confiable y las actualizaciones del mismo router deben ser procesadas en forma secuencial.

Las tareas que debe desempeñar cada uno de los nodos son:

Envío de paquetes de ECO de forma periódica, identificándose a uno mismo si es un enlace punto a punto o a un grupo de direcciones predefinido en el caso de una LAN.

Una vez conocidos los vecinos se construye un LSP, periódicamente o al descubrir un nuevo vecino, al cambiar el coste de un enlace o al caer un enlace.

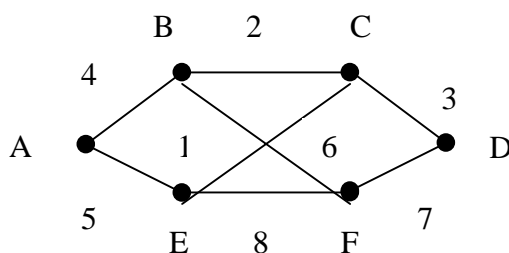
Y por último la distribución de los LSPs que se realiza mediante el algoritmo de Inundación se debe hacer correctamente, porque de no ser así pueden suceder varias cosas:

- que los routers tengan distintos LSPs, con lo que calcularan rutas basándose en informaciones diferentes.
- o que el nº de LSPs puede multiplicarse llegando a colapsar la red.

Mensaje LSP:

|                  |
|------------------|
| Fuente           |
| Nº de secuencia  |
| Edad             |
| Lista de vecinos |

En este ejemplo se muestran los distintos mensajes LSP para dicha red:



| A    | B    | C    | D    | E    | F    |
|------|------|------|------|------|------|
| SEC  | SEC  | SEC  | SEC  | SEC  | SEC  |
| EDAD | EDAD | EDAD | EDAD | EDAD | EDAD |
| B    | 4    | A    | 4    | B    | 2    |
| C    | 2    | D    | 3    | F    | 7    |
| E    | 5    | C    | 2    | D    | 3    |
| F    | 6    | E    | 1    |      |      |
|      |      | F    | 8    | E    | 8    |

### 2.2.1.- ALGORITMOS DE INUNDACIÓN

En este algoritmo los nodos no intercambian información de control. Cuando un paquete llega a un nodo de la red, lo que éste hace es conmutarlo por todos los puertos de salida sin mirar ninguna tabla de encaminamiento. De esta forma se asegura que el paquete llegue al destino.

*Problema:* si la topología presenta bucles el paquete estará dando vueltas de manera indefinida. Como consecuencia, se consume capacidad de red ilimitada. Además, llegan paquetes duplicados.

*Solución:* La solución pasa por limitar la vida del paquete en la red introduciendo un campo edad al LSP que actúa como un TTL, es decir, establecer una caducidad. Cuando se genera un paquete se incluye en un campo el número máximo de saltos que éste puede dar. Cada vez que ese paquete es conmutado el campo "número de saltos" se decrementa en una unidad hasta que sea cero, en cuyo caso los nodos ya no lo conmutan, sino que lo descartan. De esta manera aseguramos una existencia limitada de paquete dentro de la red.

Es necesario establecer el valor inicial del contador lo suficientemente alto como para que el paquete sea capaz de llegar al destino, pero tampoco excesivamente alto para que no consuma muchos recursos. La decisión que se adopta es inicializar el contador al número de saltos necesario para llegar desde cualquier nodo de la red a otro (El valor es el mismo para todos los paquetes generados, sea cual sea el nodo de la red que lo haga).

Otra solución es determinar si un nodo ha conmutado ya ese paquete. En este caso, un nodo origen marca el paquete generado con un número de secuencia de tal forma que la unicidad de ese paquete viene dada por el par (origen, secuencia). Cuando un nodo conmuta un paquete mira en tabla; si ya lo ha conmutado lo descarta, y si no lo ha conmutado lo introduce en la tabla. Como ventaja respecto de la solución anterior se consume menos capacidad de red, pero posee el inconveniente de que la tecnología del nodo es muy compleja y el tamaño de las tablas puede ser muy grande. Además esta solución puede dar mayores retardos que la anterior.

Como protección contra errores, en las líneas router-router todos los paquetes de estado de enlace requieren reconocimiento.

*Ventajas del método:*

- Detección de errores más sencilla (si un estado de enlace es infinito, significa que el nodo ha caído).
- Convergencia rápida.
- Alta adaptabilidad a los cambios, ya que los nodos tienen información de toda la red
- Menor consumo de capacidad: el tamaño del tráfico enviado es siempre el mismo independientemente del tamaño de la red.

*Inconvenientes del método:*

- Difusión.
- Consumo de memoria elevado: cada nodo almacena toda la topología de la red.

Haciendo referencia al ejemplo anterior ahora veremos el buffer de paquetes del router B:

Indicadores de:      Enviarse a C y F y  
Envío   Estado      reconocerse a A

| ORIGEN | SEC | EDAD | A C F | A C F | DATOS |
|--------|-----|------|-------|-------|-------|
| A      | 21  | 60   | 0 1 1 | 1 0 0 |       |
| F      | 21  | 60   | 1 1 0 | 0 0 1 |       |
| E      | 21  | 59   | 0 1 0 | 1 0 1 |       |
| C      | 20  | 60   | 1 0 1 | 0 1 0 |       |
| D      | 21  | 59   | 1 0 0 | 0 1 1 |       |

Paquete de estado de enlace recién llegado

Llegó 2 veces EAB, EFB.  
Se ha de enviar a C y reconocerse a A y F

Los indicadores de envío y de estado nos dicen a que nodos debemos enviar un paquete LSP y a cuales se deben reconocer la llegada de un paquete, porque como hemos comentado anteriormente en líneas router-router los LSP requieren reconocimiento.

Si llega una copia del estado de C desde el nodo F antes de reenviarse la cuarta entrada de la tabla, cambiarán los 6 bits de la tabla para indicar que el paquete ha de ser reconocido ante F y no enviarse, quedando la tabla de la siguiente forma:

| ORIGEN | SEC | EDAD | A C F | A C F | DATOS |
|--------|-----|------|-------|-------|-------|
| C      | 20  | 60   | 1 0 0 | 0 1 1 |       |

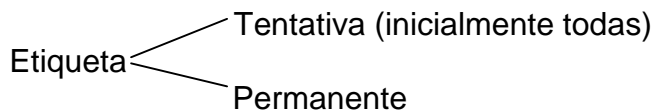
Como podemos observar en la tabla el campo edad para los nodos de A, C y F es de 60, ya que están conectados directamente al router B y sólo será este nodo el que decremente el campo dejándolo en 60.

Para los nodos E y D el campo tiene un valor de 59, esto es así porque para llegar a B tiene que pasar por un nodo intermedio, el cual decremente el campo a 60 y cuando llega al nodo B este realizará la misma acción dejándolo en 59.

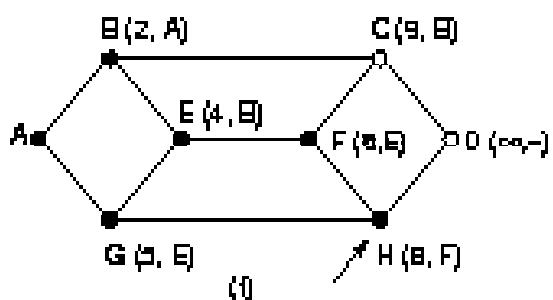
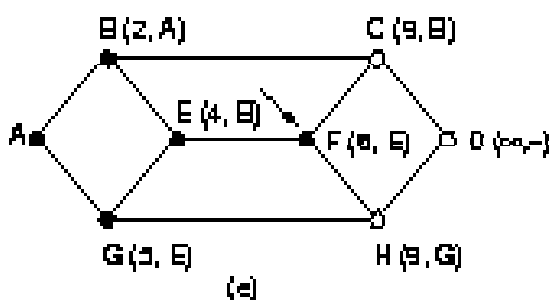
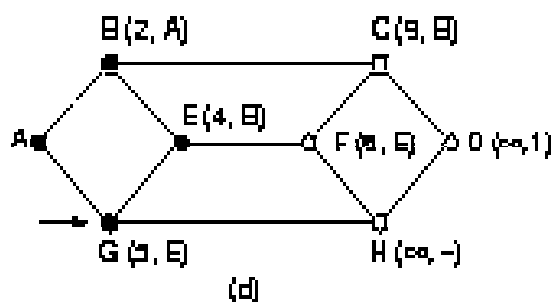
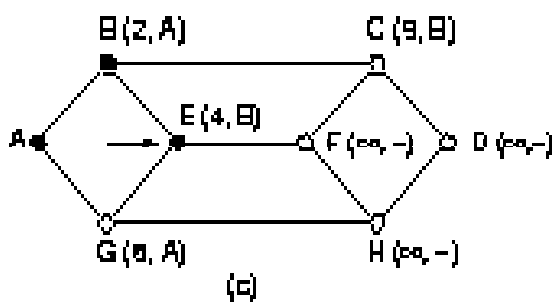
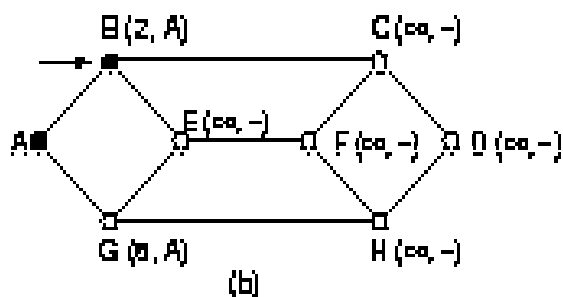
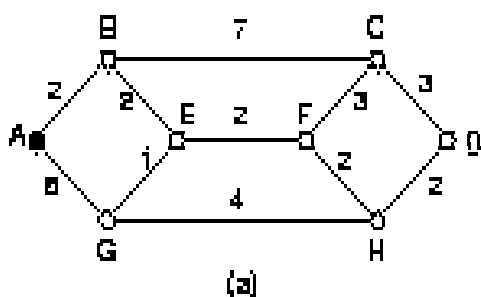
### 2.2.2.-CALCULO DE RUTAS

Cuando un router tiene un conjunto completo de LSPs tiene conocimiento de la topología de la red. El algoritmo más utilizado para calcular rutas óptimas es el de DIJKSTRA, en el que cada nodo obtiene un árbol del camino más corto donde se incluyen todos los destinos posibles en la red. Luego cada nodo envía los datos de su tabla de ruteo en forma de actualización a los demás nodos.

Cada nodo se etiqueta con su distancia al nodo de origen a través de la mejor trayectoria conocida. Inicialmente todos tienen infinito, tal como avanza el algoritmo pueden cambiar las etiquetas reflejando mejores trayectorias.



A continuación se estudiará un ejemplo del algoritmo de Dijkstra, para la red mostrada en la figura (a).



Paso a paso se explicará cada uno de las seis figuras:

- (a) Se marca el nodo inicial A como permanente. Se encaminan por turnos los nodos adyacentes reetiquetándolos con la distancia desde el nodo inicial A, también se indica desde que nodo se hizo la prueba para reconstruir luego el camino.
- (b) Habiendo examinado cada nodo adyacente se miran todos los tentativos en el grafo completo y hacemos permanentes al de la etiqueta más pequeña, este se convierte en el nuevo nodo de trabajo. En nuestro caso, B y G son los nodos tentativos, como B tiene la distancia más corta se convierte en el nodo permanente.
- (c) C y E pasan a ser los nodos tentativos, al tener E la distancia más corta, se convierte en el nodo permanente.
- (d) F y G serán ahora los nodos tentativos, G pasa a ser el nodo permanente al tener la distancia más corta.
- (e) Ahora los nodos tentativos son C, H y F, pasando a ser el permanente F.
- (f) C y H son los nodos tentativos, convirtiéndose en permanente el nodo H.

RIP

### 3.- PROTOCOLO DE ENRUTAMIENTO RIP

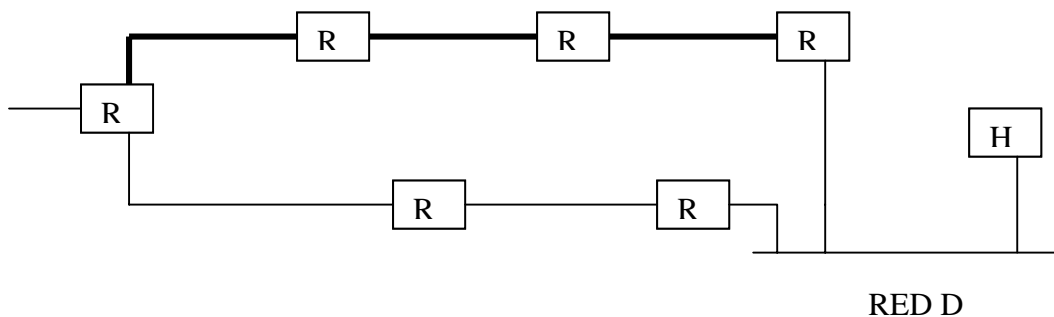
#### 3.1.-RIP v1

Hay dos versiones de RIP: La versión 1 (v1) de RIP está definida en RFC 1058 y la versión 2 (v2) en RFC 1723.

El protocolo RIP (*Routing Information Protocol*) para el IP facilita a los routers el intercambio de información de las direcciones IP de las redes (las direcciones 'alcanzables' por el *router*), y la 'distancia' de estas redes.

RIP está clasificado como un protocolo de vector de distancia, lo que implica que utiliza la distancia, medida en saltos de ruteo, para determinar el camino óptimo de un paquete (el número de saltos de RIP es independiente del campo TTL -Tiempo de vida- del encabezado IP).

Esto conlleva un problema fundamental: cuando elige una ruta, RIP no toma en cuenta la velocidad de los enlaces involucrados, ya que sólo entiende de saltos. Por ejemplo: si un camino que consiste exclusivamente en enlaces Fast Ethernet está un salto más lejos que un camino que incluye un enlace Ethernet de 10 Mbps, RIP seleccionará –como ruta óptima– al enlace Ethernet de 10 Mbps (es decir, escogerá la tecnología más lenta).



- Enlace Fast Ethernet. Tres saltos.
- Enlace Ethernet de 10 Mbps. Dos saltos.

RIP opera en uno de los dos modos siguientes: *activo* (normalmente lo usan los routers) y *pasivo* (normalmente lo usan los hosts). Ambos participantes RIP, activo y pasivo, escuchan todos los mensajes emitidos y actualizan sus tablas de enrutamiento según el algoritmo vector-distancia, la única diferencia que hay es que los nodos activos pueden anunciar rutas y los pasivos no.

Los ruteadores, cada 30 segundos, envían anuncios a los demás ruteadores. Cada dispositivo que recibe el mensaje incrementa en uno la cuenta de saltos. Si se reciben anuncios de varios ruteadores, se escoge el camino que apunta en la dirección del ruteador que tiene la cuenta de saltos

más baja. Por otro lado, si la ruta preferida no está disponible, se usará la vía con la cuenta de saltos más alta como respaldo.

Con RIP (y otros protocolos de ruteo), los ruteadores de la red deben efectuar un proceso para determinar caminos alternos cuando una ruta deja de estar disponible. Esta operación lleva el nombre de “convergencia”. Aquí radica un gran problema: RIP tarda mucho tiempo en convergir. Este protocolo está diseñado para esperar hasta haber perdido seis actualizaciones (un total de 180 segundos), antes de considerar que la ruta no es transitable. Después, RIP aguarda al próximo anuncio de otra ruta disponible, antes de actualizar la tabla de ruteo con la nueva vía.

Esto implica que pasarán por lo menos tres minutos antes de que RIP utilice una ruta de respaldo, tiempo más que suficiente para dos cosas: para que los usuarios noten un retraso y para que en la mayor parte de las aplicaciones se venza el plazo para detectar una respuesta. Desde luego, esto no representa un problema si sólo existe un camino hacia cualquier destino.

RIP para IP, al igual que la mayoría de los protocolos de enrutamiento por vector de distancia, anuncia sus rutas de forma asíncrona y sin confirmación. Esto puede dar lugar a problemas de convergencia. Sin embargo, puede habilitar modificaciones en los algoritmos de anuncio para reducir el tiempo de convergencia en la mayoría de las situaciones.

Los protocolos de ruteo también deben impedir que los paquetes viajen en círculos o caigan en ciclos de ruteo, problema que afecta a las redes que tienen enlaces redundantes. RIP supone que si hay más de 15 saltos de ruteo, desde un extremo de la red al otro, entonces deben existir ciclos. Por lo tanto, cuando una ruta llega a 16 saltos, este protocolo considera red inalcanzable. Obviamente, esto limita al RIP a redes en las que nunca es necesario pasar por más de 15 ruteadores.

Los problemas más graves de RIP se presentan en las redes grandes que poseen caminos redundantes.

### *3.1.1.- FUNCIONAMIENTO DE RIP PARA IP*

El funcionamiento normal de un enrutador RIP para IP consta de un proceso de inicialización (durante el cual el enrutador aprende las rutas de la red gracias a los enrutadores del entorno), un continuo proceso de anuncios periódicos y el anuncio adecuado de las rutas inalcanzables cuando el enrutador queda inactivo debido a una acción administrativa.

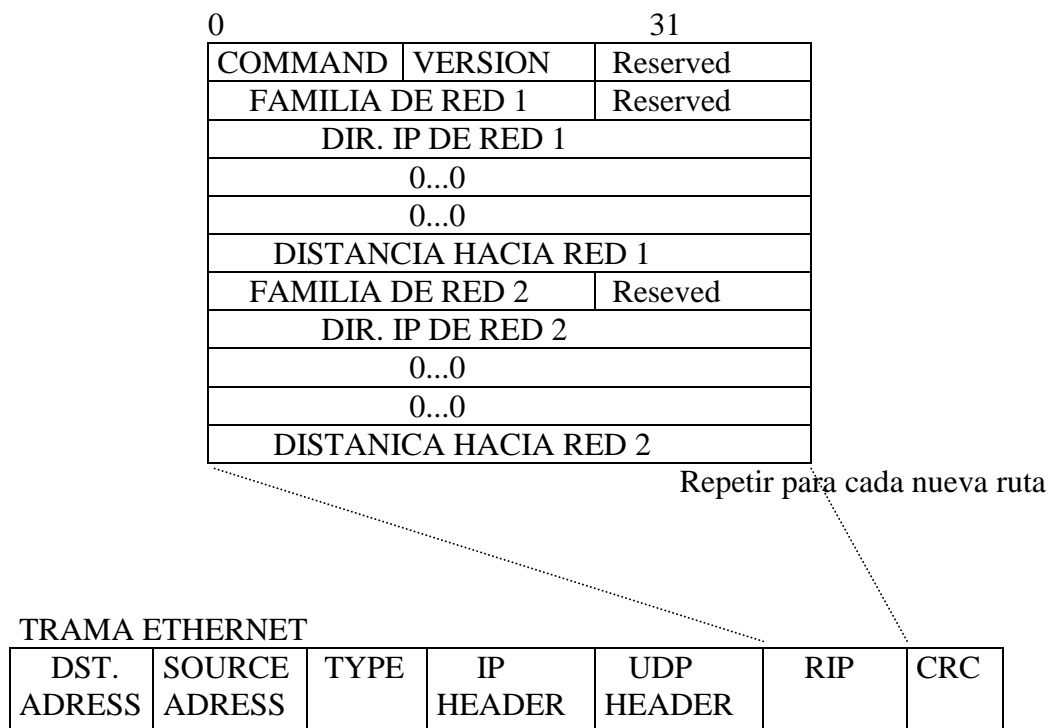
Tras el inicio, el enrutador RIP para IP anuncia las redes que tiene conectadas localmente en todas sus interfaces. Los enrutadores RIP del entorno procesan el anuncio RIP y agregan la red o las redes nuevas a sus tablas de enrutamiento.

El enrutador RIP que se está inicializando también envía una solicitud RIP general a todas las redes conectadas localmente. La solicitud RIP general es un mensaje RIP especial que solicita todas las rutas. Los enrutadores RIP del entorno reciben la solicitud RIP general y envían una respuesta de *unicast* al enrutador solicitante. Las respuestas se utilizan para crear la tabla de enrutamiento del enrutador RIP que se está inicializando.

Si el enrutador RIP almacenara una lista completa de todas las redes y todas las formas posibles de llegar a cada red, la tabla de enrutamiento podría tener cientos, o incluso miles, de entradas en el caso de una red IP grande con múltiples rutas de acceso. Puesto que sólo se pueden enviar 25 rutas en un único paquete RIP, las tablas de enrutamiento grandes tienen que enviarse como múltiples paquetes RIP.

### 3.1.2.- FORMATO DE MENSAJES DE RIP v1

Los mensajes RIP están encapsulados en un datagrama UDP (Protocolo de datagramas de usuario) enviado desde la dirección IP de la interfaz del enrutador y el puerto UDP 520 a la dirección IP de difusión de la subred y el puerto UDP 520. El mensaje RIP v1 consta de un encabezado RIP de 4 bytes y de hasta 25 rutas RIP. El tamaño máximo del mensaje RIP es de 504 bytes. Con el encabezado UDP de 8 bytes, el tamaño máximo del mensaje RIP es una carga IP de 512 bytes. La figura 8 ilustra el formato de los mensajes RIP v1.



- **Command:** Campo de 1 byte que contiene 0x01 ó 0x02. 0x01 indica una solicitud RIP de todas (una solicitud RIP general) o parte de las tablas de enrutamiento de los enrutadores del entorno. 0x02 indica una respuesta RIP que consta de toda o parte de la tabla de enrutamiento de un enrutador vecino. Se puede enviar una respuesta RIP como contestación a una solicitud RIP, o como un mensaje periódico o de actualización desencadenada.
- **Versión:** Campo de 1 byte que se establece con el valor 0x01 para RIP v1.
- **Familia de Red:** Campo de 2 bytes que identifica la familia de protocolos. Se establece con el valor 0x00-02 para indicar la familia de protocolos IP.
- **Dirección IP:** Campo de 4 bytes que se establece como el Id. de red IP que puede ser un Id. de red basado en clases, un Id. de red con subredes (anunciado sólo dentro de la red con subredes), una dirección IP (para una ruta de *host*) o 0.0.0.0 (para la ruta predeterminada). En el caso de una solicitud RIP general, la dirección IP se establece como 0.0.0.0.
- **Distancia hacia Red:** Campo de 4 bytes para el número de saltos a la red IP que debe ser un valor de 1 a 16. La métrica se establece como 16 en una solicitud RIP general o para indicar que la red es inalcanzable en una respuesta RIP (anuncio).

### 3.1.3.- PROBLEMAS DE RIP v1

RIP v1 se diseñó para las redes IP basadas en clases, donde el Id. de red puede determinarse a partir de los valores de los 3 primeros bits de la dirección IP de la ruta RIP. Como la máscara de subred no se incluye con la ruta, el enrutador RIP debe determinar el Id. de red basándose en un conjunto limitado de información.

RIP v1 no proporciona ninguna protección para evitar que un enrutador RIP malintencionado se inicie en una red y anuncie rutas erróneas o imprecisas. Los anuncios RIP v1 se procesan independientemente de cuál sea su origen. Un usuario malintencionado podría utilizar esta falta de protección para sobrecargar los enrutadores RIP con cientos o miles de rutas incorrectas o imprecisas.

## 3.2.- RIP v2

RIP versión 2 (v2), como se define en RFC 1723, intenta solucionar algunos de los problemas asociados a RIP v1.

### 3.2.1.-CARACTERÍSTICAS DE RIP v2

Para que las redes IP actuales minimizaran el tráfico de difusión, utilizaran subredes de longitud variable para ahorrar direcciones IP y aseguraran su entorno de enrutamiento frente a enrutadores mal configurados o malintencionados, se agregaron distintas características clave a RIP v2.

Anuncios RIP con *multicast*: En vez de difundir anuncios RIP, RIP v2 admite el envío de anuncios RIP a la dirección de *multicast* IP 224.0.0.9. Los nodos que no son RIP no se ven afectados por el tráfico de anuncios de los enrutadores RIP. Como los anuncios con *multicast* de RIP v2 se envían a 224.0.0.9 con un TTL de 1, no es necesario el uso de Internet Group Membership Protocol (IGMP, Protocolo de pertenencia a grupos de Internet, que veremos más adelante) para registrar la pertenencia al grupo del *host*.

La desventaja de esta nueva característica es que los nodos RIP silencioso también deben escuchar el tráfico de *multicast* enviado a 224.0.0.9. Si utiliza RIP silencioso, compruebe que los nodos RIP silencioso pueden escuchar anuncios RIP v2 con *multicast* antes de distribuir RIP v2 con *multicast*.

El uso de anuncios con *multicast* es opcional. También se admite la difusión de anuncios RIP v2.

Máscaras de subred: Los anuncios RIP v2 envían la máscara de subred junto con el Id. de red. Se puede utilizar RIP v2 en entornos de subredes, de superredes y de máscara de subred de longitud variable. Las subredes de un Id. de red no tienen que ser contiguas (pueden ser subredes disjuntas).

Autenticación RIP v2 admite el uso de mecanismos de autenticación para comprobar el origen de los anuncios RIP entrantes. En RFC 1723 se definió la autenticación por clave de acceso simple.

Los enrutadores RIP v1 son compatibles con RIP v2. RIP v1 se diseñó teniendo en cuenta su compatibilidad con versiones posteriores. Si un enrutador RIP v1 recibe un mensaje y la versión RIP del encabezado no es 0x01, no descarta el anuncio RIP pero sólo procesa los campos definidos de RIP v1. Además, los enrutadores RIP v2 envían una respuesta RIP v1 a una solicitud RIP v1, excepto cuando están configurados para enviar únicamente anuncios RIP v2.

### 3.2.2.-FORMATO DE MENSAJES DE RIP v2

Para asegurar que los enrutadores RIP v1 pueden procesar anuncios RIP v2, RIP v2 no modifica la estructura del formato de mensajes RIP. RIP v2 utiliza los campos que en RIP v1 se definieron como “Debe ser cero”.

El uso de los campos de comando, identificador de familia, dirección IP y métrica son los mismos que los definidos anteriormente para RIP v1. El campo de versión se establece como 0x02 para indicar un mensaje RIP v2.

Los mensajes de RIP-2 al igual que los de RIP-1 se componen de una cabecera de 32 bits seguida de un conjunto de entradas de 20 octetos cada una.

La cabecera sólo modifica la versión, la principal diferencia son las entradas:

|                   |           |
|-------------------|-----------|
| FAMILIA DE RED    | ROUTE TAG |
| DIRECCIÓN IP      |           |
| MÁSCARA DE SUBRED |           |
| SIGUIENTE SALTO   |           |
| DISTANCIA         |           |

- Route tag: Este campo se utiliza como método para marcar rutas específicas con propósitos administrativos. Su uso original, tal como se define en RFC 1723, fue distinguir las rutas que estaban basadas en RIP (internas al entorno RIP) de las que no lo estaban (externas al entorno RIP). La etiqueta de ruta es configurable en los enrutadores que admiten múltiples protocolos de enrutamiento.
- Máscara de subred: Este campo de 4 bytes contiene la máscara de subred del Id. de red en el campo de la dirección IP.
- Siguiendo salto: Este campo de 4 bytes contiene la dirección IP de reenvío (también denominada dirección de puerta de enlace) para el Id. de red en el campo de la dirección IP. Si el siguiente salto se configura como 0.0.0.0, se supone que la dirección IP de reenvío (el siguiente salto) de la ruta será la dirección IP de origen del anuncio de ruta.

El campo de siguiente salto se utiliza para evitar situaciones de enrutamiento que no sean óptimas. Por ejemplo, si un enrutador anuncia una ruta de *host* para un *host* que se encuentra en la misma red que la interfaz del enrutador que anuncia la ruta y no se utiliza el campo de siguiente salto, la dirección IP de reenvío para la ruta de *host* será la dirección IP de la interfaz del enrutador, no la dirección IP del *host*. Los demás enrutadores que reciban el anuncio en esa red reenviarán los paquetes destinados a la dirección IP del

*host* a la dirección IP del enrutador que efectúa el anuncio, en vez de reenviarlos al *host*. Esto crea una situación de enrutamiento que no es óptima.

Al utilizar el campo de siguiente salto, el enrutador anuncia la ruta de *host* con la dirección IP del *host* en este campo. Los demás enrutadores que reciban el anuncio en esa red reenviarán los paquetes destinados a la dirección IP del *host* a la dirección IP del *host*, en vez de reenviarlos al enrutador que efectúa el anuncio. Como el campo de siguiente salto se convierte en el campo de dirección de puerta de enlace en la tabla de enrutamiento IP, la dirección IP del campo de siguiente salto debe poderse alcanzar directamente mediante una interfaz de enrutador.

### 3.2.3.-AUTENTICACIÓN EN RIP v2

El proceso de autenticación de los anuncios RIP v2 utiliza la primera entrada de ruta del mensaje RIP para almacenar la información de autenticación. Se debe utilizar la primera entrada de ruta, con lo que queda un máximo de 24 rutas en un anuncio autenticado de RIP v2. Para indicar autenticación, el campo de identificador de familia se establece como 0xFF-FF. El campo de tipo de autenticación, normalmente utilizado como el campo de etiqueta de ruta para una ruta, indica el tipo de autenticación que se utiliza. La autenticación por clave de acceso simple utiliza el valor 0x00-11 para el tipo de autenticación.

Los 16 bytes que vienen a continuación del tipo de autenticación se utilizan para almacenar el valor de autenticación. En el caso de la autenticación por clave de acceso simple, el campo de valor de autenticación de 16 bytes almacena la clave de acceso justificada a la izquierda, rellena con caracteres nulos, con distinción de mayúsculas y minúsculas y en texto no cifrado. La figura ilustra el mensaje de autenticación de RIP v2.

|  |         |                         |
|--|---------|-------------------------|
| COMMAND                                | VERSION | Reserved                |
| FFFF                                   |         | TIPO DE AUTENTIFICACIÓN |
| DATOS DE AUTENTIFICACIÓN<br>(16 bytes) |         |                         |

Los enrutadores RIP v1 descartan la primera ruta de un anuncio autenticado de RIP v2 porque el identificador de familia para la ruta es desconocido.

La autenticación por clave de acceso simple para RIP v2 evita que en la red se coloquen enrutadores RIP no autorizados o mal configurados. Sin embargo, la clave de acceso simple no es segura porque se envía por la red como texto no cifrado. Cualquier usuario con un analizador de protocolos, puede capturar los paquetes RIP v2 y ver la clave de acceso de autenticación.

### 3.3.-ENTORNOS MIXTOS DE RIP v1 Y RIP v2

Se debe tener precaución al utilizar conjuntamente enrutadores RIP v2 y enrutadores RIP v1. Puesto que los enrutadores RIP v1 no interpretan el campo de máscara de subred en la ruta, los enrutadores RIP v2 no deben anunciar rutas que un enrutador RIP v1 pueda interpretar incorrectamente. Las máscaras de subred de longitud variable (VLSM) y las subredes disjuntas no pueden utilizarse en entorno mixtos.

En el caso de una interfaz que utilice RIP v2 para realizar anuncios de modo que los enrutadores RIP v1 puedan procesar las rutas anunciadas, los enrutadores RIP v2 deben resumir las rutas de subred cuando se anuncien fuera de un entorno con subredes. Una ruta de subred específica anunciada a un enrutador RIP v1 puede malinterpretarse como una ruta de *host*. Además, los enrutadores RIP v2 no pueden anunciar rutas de superredes. Un enrutador RIP v1 malinterpretaría la ruta como una única red, en vez de hacerlo como un intervalo de redes.

Si los enrutadores RIP v2 se encuentran en la misma red que los enrutadores RIP v1, la interfaz del enrutador RIP v2 debe configurarse para difundir sus anuncios. Los enrutadores RIP v1 no procesan los anuncios RIP v2 con *multicast*.

OSPF:

OPEN SHORTEST PATH FIRST

## **4.- Open Shortest Path First (OSPF)**

OSPF es un protocolo de ruteo del tipo estado de enlace, que soporta ruteo jerárquico dentro de un sistema autónomo. OSPF provee un muy rápido ruteo y soporta máscaras de subred de longitud variable. OSPF se derivó del protocolo de ruteo IS-IS de la OSI. Algunas características especiales de OSPF incluyen ruteo de múltiples caminos de costo y ruteo basado en un tipo de nivel superior de solicitudes del servicio (TOS Type-Of-Services). Por ejemplo, una aplicación puede especificar que ciertos datos son urgentes y si OSPF tiene enlaces de alta prioridad a su disposición, ellos pueden ser utilizados para transportar un paquete urgente. OSPF soporta uno o más métricas. Así un router tendrá que examinar de la cabecera IP la dirección destino y el tipo de servicio para seleccionar la ruta.

En OSPF, un router no intercambia distancias con sus vecinos. En vez de eso, cada router chequea el status de cada uno de sus enlaces con los routers adyacentes y envía a éstos la información recogida, la que se propaga de esta forma a través del sistema autónomo. Cada router captura esta información y construye su tabla de ruteo, y todos los routers involucrados tendrán la misma tabla de ruteo.

OSPF proporciona Balance de Carga, es decir, es capaz de distribuir el tráfico entre routers con igual coste hacia un destino.

OSPF permite que se agrupen juntas colecciones de redes y hosts. Esta agrupación, junto con todos los routers que tienen interfaces a cualquiera de las redes incluidas es llamada un *área*. Cada área ejecuta una copia separada del algoritmo de ruteo básico SPF, lo que implica que cada área tiene su propia base de datos topológica.

La topología de un área es invisible para cualquier dispositivo que no pertenezca a ella. Es decir, los router internos de un área específica no saben nada de la topología externa al área. Esta aislación es la que permite introducir un bajo tráfico de ruteo en la red, en comparación a compartir toda la información del sistema autónomo. Los routers que están conectados a múltiples áreas son llamados *routers de borde de área (ABR)*. Es así como dos routers que pertenecen a una misma área tienen, para esa área, una base de datos idéntica.

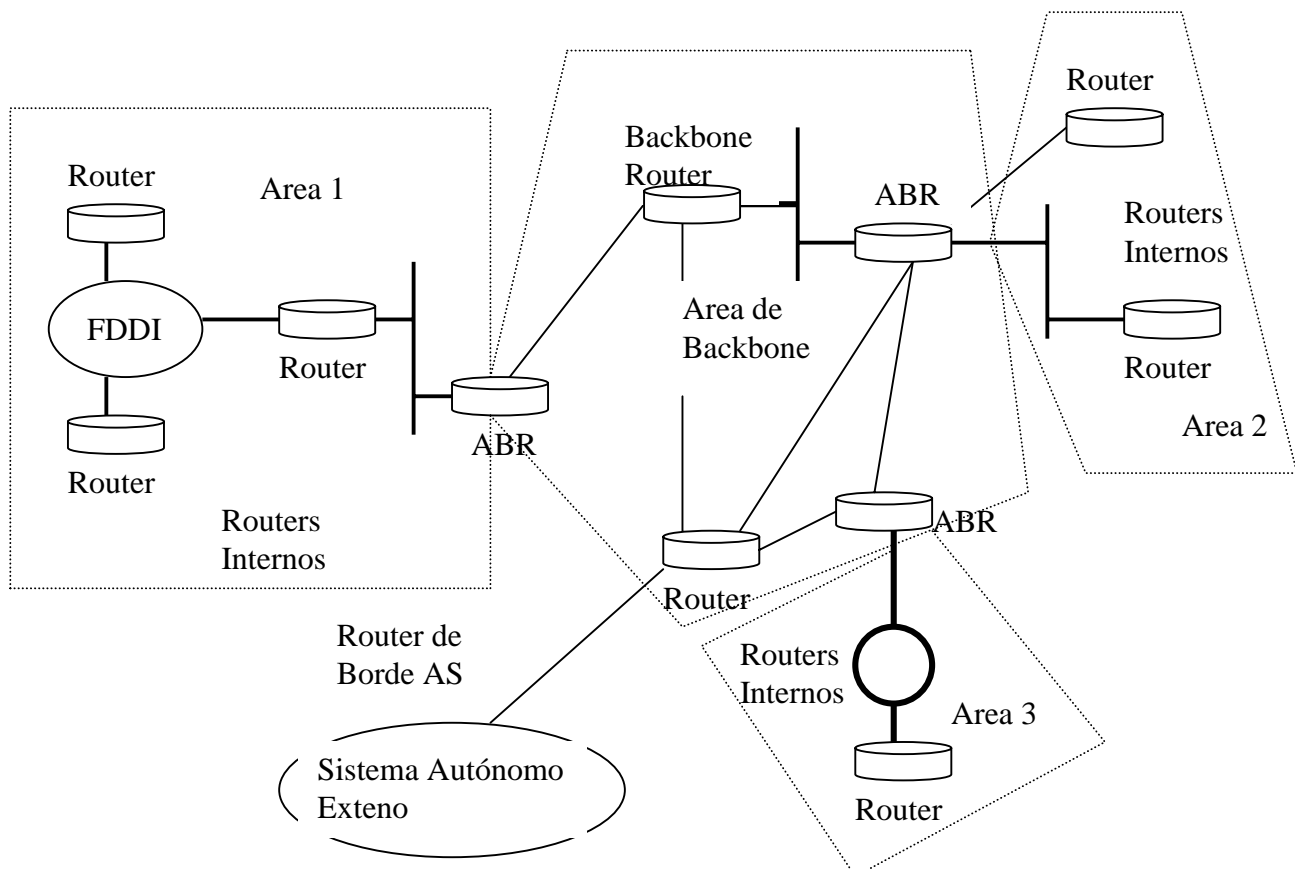
El ruteo en un sistema autónomo tiene dos niveles, dependiendo de si la fuente y el destino están en una misma área o no. El *ruteo intra-área* pertenece al primer caso, los paquetes son ruteados con información exclusivamente del área en cuestión. Esto protege al ruteo de la inyección de información corrupta. En el *ruteo Inter-área*, se obtiene información del o las áreas exteriores involucradas.

#### 4.1.- CLASIFICACIÓN DE LOS ROUTERS OSPF

Cada sistema autónomo tiene un área principal o *backbone*, al que están conectadas el resto de las áreas en una distribución de estrella. OSPF clasifica los enrutadores en cuatro tipos, en función del área o áreas a las que pertenezcan:

- Enrutadores *internos*, contenidos en una única área.
- Enrutadores *de borde de área*, conectados a varias áreas. Estos enrutadores necesitan la base de datos de ambas áreas y deben realizar, para cada una por separado, la obtención de rutas óptimas.
- Enrutadores *de backbone*. Estos enrutadores aceptan información de los enrutadores de borde de área con el fin de calcular la mejor ruta a todos los enrutadores. Esta información se propaga de regreso a los enrutadores de borde de área, quienes la divulgan a su área. Usando esta información, un enrutador a punto de enviar un paquete inter-área puede seleccionar el mejor enrutador de salida al backbone. A continuación, el paquete atraviesa el backbone hasta alcanzar el enrutador de borde perteneciente al área de destino. Finalmente, el mensaje se desplaza desde el enrutador conectado al backbone hasta el nodo destino. Todos los enrutadores de borde de área son automáticamente parte del backbone.
- Enrutadores *de borde de AS*, que se relacionan con enrutadores de otros sistemas autónomos.

Ahora mostraremos una figura en la que se reflejan los diferentes tipos de routers y relaciones en OSPF:



Para asimilar cambios de topología, OSPF opera realizando intercambios de información entre enrutadores adyacentes, considerando *adyacencia* como una relación formada entre los routers vecinos seleccionados con el propósito de intercambiar información de ruteo. Es muy frecuente que estos estén conectados a través de una red de multidifusión (LAN). En estas situaciones, todos ellos consideran al resto como sus vecinos inmediatos, lo que conlleva que la sobrecarga añadida por el mecanismo de encaminamiento se incremente considerablemente. Estos efectos indeseables se evitan mediante la elección de un *enrutador designado*, el cual asume que sólo él es adyacente a todos los demás. El enrutador designado intercambia información (LSP) con el resto de los enrutadores, y estos no intercambiarán información entre sí. Generalmente, tras un cambio de topología el enrutador designado es aquel que ya estaba designado antiguamente o (en caso de no haber ninguno) aquel con identificador de mayor prioridad.

#### 4.2.- FORMATO DE MENSAJE OSPF

OSPF es un protocolo que se ejecuta sobre IP, es decir, sus paquetes son transmitidos encapsulados dentro de paquetes IP, lo que se indica con el campo "protocolo" asignado a 89. Los paquetes OSPF tienen la misma cabecera de longitud fija, lo que favorece su codificación compacta y rápido procesamiento, a costa de reducir su extensibilidad futura. La cabecera se muestra en la siguiente figura. El significado de cada campo es el siguiente:

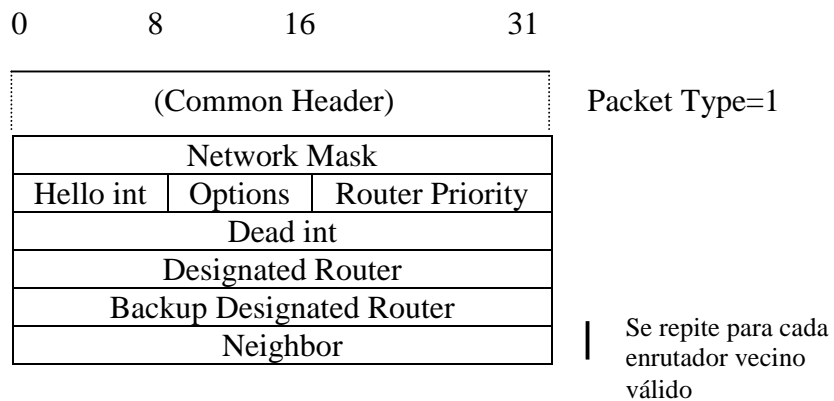
|                     |             |                     |    |
|---------------------|-------------|---------------------|----|
| 0                   | 8           | 16                  | 31 |
| Version             | Packet Type | Packet Length       |    |
| Router ID           |             |                     |    |
| Area ID             |             |                     |    |
| Checksum            |             | Authentication type |    |
| Authentication data |             |                     |    |
| Authentication data |             |                     |    |

- Version: versión del protocolo (2 en la actualidad).
- Packet type: identifica el tipo de mensaje.
  - 1 – Hello (pruebas de accesibilidad)  
Son periódicas y sirven para saber si el vecino es accesible
  - 2 – Descripción de la base de datos (Topología)
  - 3 – Petición de estado de enlace (para actualizar la Base de Datos Topológica)
  - 4 – Actualización de estado de enlace
  - 5 – Acuse de recibo de estado de enlace
- Packet length: número de octetos del paquete.
- Router ID: dirección IP del enrutador emisor.
- Area ID: identificador del área a la que pertenece el paquete.
- Checksum: código de error (similar al utilizado en IP).

- Authentication type:
  - 0 = sin autenticación
  - 1 = con clave simple
  - 2 = criptográfica
- Authentication data: clave de 64 bits

#### 4.2.1.-FORMATO DEL MENSAJE HELLO

Veamos a continuación cada uno de los cinco tipos de paquetes OSPF. En primer lugar, los paquetes *hello* permiten detectar cambios en el estado de los vecinos o de los enlaces que unen a un nodo con sus vecinos. Siempre son transmitidos entre vecinos inmediatos, y nunca recorren más de un enlace. La siguiente figura muestra el formato de estos paquetes donde se ha obviado la cabecera. El significado de cada campo es el siguiente:

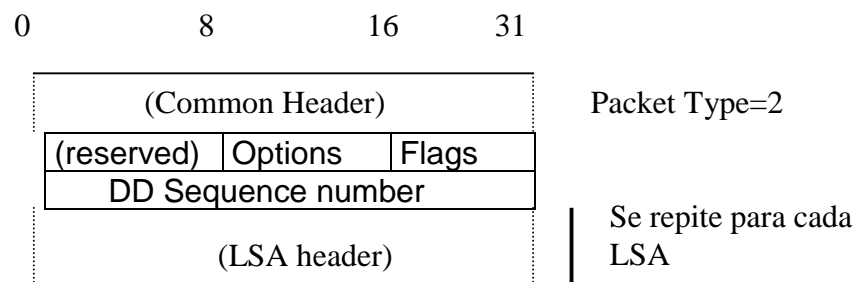


- Network mask: la máscara configurada para este enlace en el enrutador emisor. Si el receptor no comparte este valor, entonces rechaza el paquete y no acepta al emisor como vecino.
- Hello int: intervalo entre emisión de paquetes *hello* (expresado en segundos). Este campo también debe coincidir con la información del receptor.
- Options: ciertas opciones como el soporte de múltiples métricas.
- Router priority: prioridad aplicada en la elección de enrutadores designados (principales y de reserva).
- Dead int: intervalo de tiempo (expresado en segundos) en que un enrutador considera a otro desactivado si no recibe paquetes *hello*.
- Designated router: identificador del enrutador que el emisor considera enrutador designado, o cero si no considera ninguno.
- Backup designated router: identificador del enrutador que el emisor considera enrutador designado de reserva, o cero si no considera ninguno.

- Neighbors: lista de identificadores de vecinos activos, es decir, aquellos de los que ha recibido paquetes *hello* dentro del intervalo delimitado en el campo *dead int*.

#### 4.2.2.-FORMATO DEL MENSAJE DE DESCRIPCIÓN DE LA BD

Cuando se activa un enlace entre dos enrutadores, estos deben sincronizar la información topológica que poseen. Para ello, los dos nodos aceptan una relación maestro / esclavo definida en función del UID. El maestro comunica su información topológica al esclavo mediante paquetes de descripción de la base de datos (DD, *database description*), utilizando tantos como sea necesario (cada paquete se identifica por un número de secuencia). Por su parte, el enrutador esclavo confirma cada paquete DD enviando paquetes DD al maestro con el mismo número de secuencia, pero conteniendo la propia información topológica. El maestro no envía un nuevo DD en tanto en anterior no haya sido confirmado. Cuando un nodo ha terminado de transmitir su información topológica continua emitiendo paquetes vacíos hasta que termine el otro. La siguiente figura muestra el formato de estos paquetes. El significado de cada campo es el siguiente:

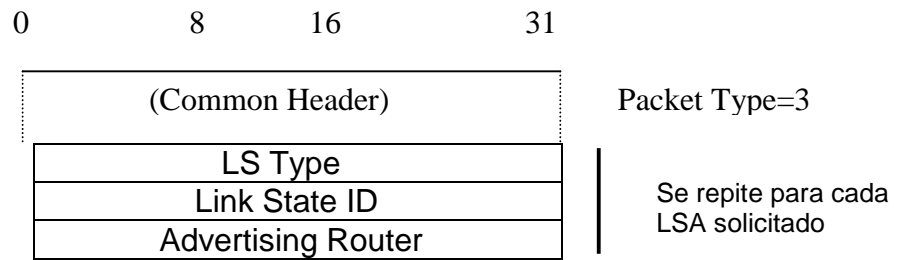


- Options: similar al campo *options* del paquete *hello*.
- Flags: bits cuyo estado activo indica que:
  - MS (master/slave): el emisor es el nodo maestro
  - M (more): no es el último paquete DD
  - I (init): es el primer paquete DD
- DD packet sequence number: número orden en la secuencia de paquetes de descripción de topología.
- LSA header: es la parte común a diferentes tipos de LSA (*link state advertisement*). Puede repetirse varias veces dentro del paquete. Como otros paquetes también incorporan LSAs, estos se describen al final.

#### 4.2.3.-FORMATO DEL MENSAJE DE SOLICITUD DE ESTADO DE ENLACE

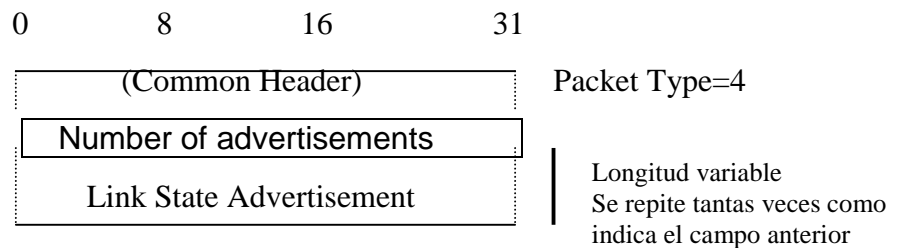
En cualquier momento, un enrutador puede solicitar información topológica a otro enrutador vecino. Para ello utiliza un mensaje de solicitud de estado de enlace (*link state request*). La siguiente figura muestra el formato de estos paquetes. Los campos *LS type*, *link state ID* y *advertising router* son en

realidad un fragmento de la cabecera de un LSA y se describen más adelante, en la descripción completa del LSA.



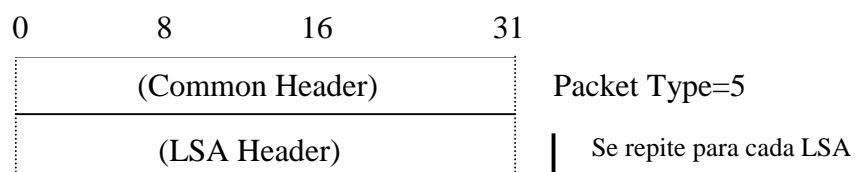
#### 4.2.4.-FORMATO DEL MENSAJE DE ACTUALIZACIÓN DE ESTADO DE ENLACE

Los paquetes de actualización de estado de enlace (*Link state update*) contienen uno o varios LSAs, y son difundidos por los routers para informar a todos los otros enrutadores sobre sus enlaces conectados directamente. La siguiente figura muestra el formato de estos paquetes.



#### 4.2.5.-FORMATO DEL MENSAJE DE ACUSE DE RECIBO DE ESTADO DE ENLACE

Los paquetes de acuse de recibo de estado de enlace (*Link state acknowledgment*) son paquetes de reconocimiento que proporcionan su confiabilidad a OSPF. Cada uno puede reconocer varios LSAs. La siguiente figura muestra el formato de estos paquetes.



#### 4.2.6.-INFORMACIÓN DE ESTADO DE ENLACE

Existen cinco tipos de LSAs. Todos ellos tienen una cabecera común. Primero se describe esta cabecera y después se comenta la información exclusiva de cada tipo. La cabecera tiene una longitud fija de 20 bytes, cuyo formato se muestra en la siguiente figura. El contenido de cada campo es el siguiente:

|                    |   |         |         |
|--------------------|---|---------|---------|
| 0                  | 8 | 16      | 31      |
| LS Age             |   | Options | LS Type |
| Link State ID      |   |         |         |
| Advertising Router |   |         |         |
| LS Sequence Number |   |         |         |
| LS Checksum        |   | Length  |         |

- LS age: indica el tiempo en segundos desde que el registro deseado de enlace fue anunciado.
- LS type: sirve para identificar el tipo de enlace, y los valores posibles son:
  - 1 = enlaces del enrutador
  - 2 = enlaces de la red
  - 3 = resumen de enlaces (subredes IP alcanzables)
  - 4 = resumen de enlaces (enrutadores alcanzables de sistemas vecinos)
  - 5 = resumen de enlaces (subredes IP alcanzables de sistemas vecinos)
- Link state ID: su significado depende del valor del campo anterior:
  - Type = 1 el ID del enrutador que generó la información
  - Type = 2 la dirección IP del enrutador designado de la LAN
  - Type = 3 la dirección IP del enlace que conecta la subred
  - Type = 4 el ID del enrutador de borde
  - Type = 5 la dirección IP del enlace que conecta la subred
- Advertising router: ID del enrutador (dirección IP) que generó la información.
- LS sequence number: número de secuencia del LSA.
- LS checksum: se aplica a la cabecera y contenido de los mensajes.
- Length: longitud del registro, incluyendo la cabecera.

#### 4.3.- MEJORAS DE OSPF FRENTE A RIP

Además de ser un protocolo de enlace en vez de distancia, OSPF tiene otras muchas características que lo hacen superior a RIP:

- OSPF puede calcular un conjunto separado de rutas para cada tipo de servicio IP. Esto quiere decir que para un mismo destino puede haber varias entradas en la tabla de ruteo, una por cada tipo de servicio.
- A cada interfaz se le asigna un costo. Este puede asignarse en función del ancho de banda de salida, seguridad, fiabilidad, etc. Pueden asignarse distintos costos para distintos servicios.
- Cuando existen varias rutas a un mismo destino, con idénticos costos, OSPF distribuye el tráfico por ambas rutas de forma equitativa.
- OSPF soporta subredes: una máscara de subred es asociada con cada ruta notificada. Esto permite que una única dirección IP de cualquier clase pueda ser dividida en múltiples subredes de varios tamaños. Las rutas a un host son notificadas mediante una máscara de subred con todos los bits a 1. Una ruta por defecto es notificada como una dirección IP de 0.0.0.0 con una máscara con todos los bits a 0.
- Los enlaces punto a punto entre routers no necesitan una dirección IP a cada extremo. Es lo que se conoce como redes no numeradas. De esta forma se ahorran direcciones IP.
- Es posible emplear un pequeño mecanismo de autenticación ya que es posible enviar un password.
- OSPF emplea multicast en vez de broadcast, para reducir la carga en los sistemas que no emplean OSPF.

Desde un punto de vista práctico, la diferencia más importante es que un protocolo de estado del enlace converge con mayor rapidez que un protocolo de vector de distancia. Por convergencia se entiende la estabilización después de cambios en la red, como caídas de router o de enlaces.

# MULTICASTING IP ROUTING

## **5.- MULTICASTING IP ROUTING**

### **5.1.- INTRODUCCIÓN**

A la hora de direccionar un host (interface) dentro de una red, se puede hacer uso de tres tipos diferentes de direcciones:

- Dirección unicast. Este tipo de dirección hace referencia a un único host (interface) dentro de la subred. Un ejemplo de dirección IP unicast es 192.168.100.9. Una dirección MAC unicast es, por ejemplo, 80:C0:F6:A0:4A:B1.
- Dirección broadcast. Con una dirección de este tipo se consigue direccionar a todos los hosts (interfaces) dentro de una subred. Una dirección IP broadcast es 192.168.100.255 y una dirección MAC broadcast es FF:FF:FF:FF:FF:FF.
- Dirección multicast. Este tipo de direcciones permite direccionar a un grupo concreto de hosts (interfaces) dentro de una subred.

Se usarán direcciones multicast cuando el destinatario de la información no sea una única máquina, pero tampoco se quiera hacer un broadcast a toda la red. Este escenario será típico de situaciones en las que se requiera el envío de información multimedia (audio o video en tiempo real) a varios hosts de la red. En casos como este no es óptimo, en términos de ancho de banda, establecer un envío unicast a cada uno de los clientes que quieran recibir la emisión multimedia. Establecer un envío broadcast tampoco es la solución, sobretodo si alguno de los clientes están fuera de la subred local desde la cual se realiza el envío.

Si un host se une a un grupo multicast, recibirá todo el tráfico unicast dirigido a él, el broadcast dirigido a toda la subred y el tráfico multicast dirigido al grupo al que se ha unido.

### **5.2.- FUNCIONAMIENTO DEL MULTICAST**

El contenido de la transmisión de flujos llega de cualquiera de estas dos maneras: unidifusión (unicast) punto a punto o multidifusión.

Con unidifusión los datos se envían en un flujo separado desde el origen a cada usuario que lo solicite. Este método funciona en situaciones en las que cada usuario desea un contenido diferente al del resto, pero cuando mucha gente quiere el mismo contenido al mismo tiempo. La entrega del mismo contenido a miles de usuarios sería inviable, ya que el ancho de banda necesario para el servidor sería prohibitivo por su coste.

La transmisión de flujos multidifusión elimina este problema recurriendo al método “uno a muchos”. En lugar de difundir cientos de flujos, el servidor envía sólo uno. El flujo se propaga entre los diversos usuarios que lo han solicitado.

De esta manera, el ancho de banda necesario, tanto para el servidor como para la red, se reduce.

Como vemos es una solución eficiente para mejorar el rendimiento de la red. El multicasting trabaja a partir de una arquitectura de red jerárquica. El flujo de datos IP multicast es capturado por los routers sólo cuando alguno de los ordenadores que tienen conectados está suscrito a este flujo de datos. Los routers envían los datos sólo hacia aquellos conmutadores y concentradores que tengan clientes a la escucha.

En el caso de querer trabajar con multicast en WAN, se necesitan routers con soporte multicast que se comuniquen entre ellos mediante algún protocolo de encaminamiento que contemple el multicast. Cuando un proceso en un host de una subred se asocia a un grupo multicast, este host envía un mensaje IGMP a todos los routers multicast de su subred, informándoles que cuando reciban un mensaje multicast destinado al grupo al cual él se ha asociado, lo envíen a la subred para que pueda recibirlo. Estos routers le comunicarán esta información al resto de routers multicast de tal forma que todos los routers sepan a quién deberán encaminar los mensajes multicast que le lleguen.

Los routers además envían de forma periódica mensajes IGMP al grupo 224.0.0.1 solicitando a los hosts información sobre los grupos a los cuales están asociados. Un host, al recibir este mensaje inicializa un temporizador con un valor aleatorio, y no contestará hasta que este temporizador llegue a cero. Con esto se evita que todos los hosts contesten a la vez, produciendo una sobrecarga innecesaria en la red. Cuando el temporizador de alguno de los hosts llegue a cero, enviará su contestación a la dirección del grupo multicast concreto del cual esté informando, por lo que el resto de hosts asociado a ese grupo verán la contestación, y anularán su temporizador no generando por tanto su respuesta. Esto se hace porque con un host que conteste es suficiente, al router únicamente le hace falta saber que hay un host interesado en determinado grupo en esa subred, con eso le basta para redirigir los mensajes multicast destinados al grupo, el resto de hosts los recibirán y no es necesario por tanto que también contesten ellos.

Si todos los hosts que estaban en un determinado grupo, se quitan del mismo, entonces ninguno contestará a los mensajes del router, quién al ver que ya no hay nadie interesado en determinado grupo en una subred, dejará de encaminar a la misma los mensajes destinados a ese grupo. Otra opción, implementada en IGMPv2, es que el propio host indique a los routers que ha abandonado un determinado grupo, enviando para ello un mensaje a la dirección 224.0.0.2.

Gran parte del interés del multicasting se ha dirigido a las aplicaciones multimedia. Las empresas están considerando seriamente esta opción para implementar aplicaciones como la enseñanza a distancia, la difusión de noticias sobre el escritorio y las reuniones de empresas virtuales. Otra aplicación a considerar es la distribución de software.

La tecnología IP multicast también puede ayudar a las empresas que necesiten transmitir pequeños archivos de datos a muchos usuarios.

Al igual que Internet, las intranets y el Web emprenden más servicios antes gestionados por sistemas cliente / servidor, la tecnología multicasting puede proporcionar un camino para la expansión de servicios sin necesidad de cambiar por completo una red.

### *5.2.1.- DIRECCIONAMIENTO CLASE D*

El espacio de direccionamiento IP se distribuye en cuatro grupos o clases de direcciones, las direcciones de clase A, B, C y D. La clase D está reservada para las direcciones multicast y tiene reservado el rango de direcciones IPv4 entre la 224.0.0.0 y la 239.255.255.255.

Las direcciones multicast IPv4 a nivel de red, deben mapearse sobre las direcciones físicas correspondientes al tipo de red con el se esté trabajando. Si se estuviese trabajando con direcciones a nivel de red unicast, se obtendría la dirección física asociada haciendo uso del protocolo ARP.

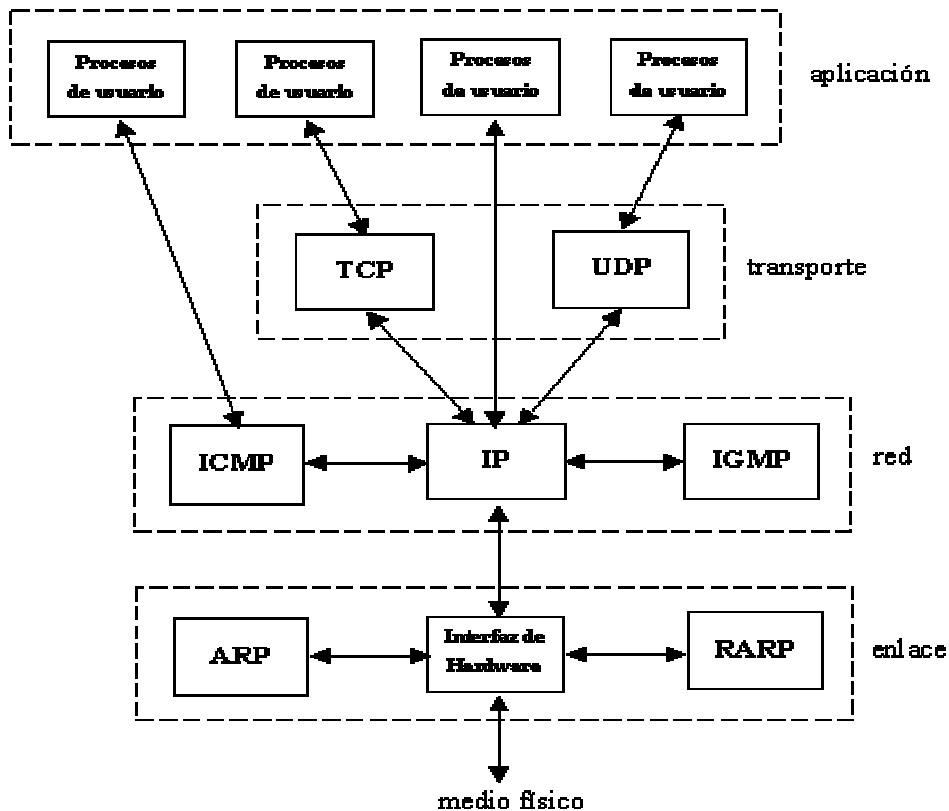
Entre otras direcciones multicast IPv4 reservadas encontramos las siguientes:

- La dirección 224.0.0.1 identifica a todos los hosts de una subred. Cualquier host con capacidades multicast que se encuentre en una subred deberá unirse a este grupo.
- La dirección 224.0.0.2 identifica a todos los routers con capacidades multicast de una subred.
- El rango de direcciones 224.0.0.0 - 224.0.0.255 está reservado para protocolos de bajo nivel. Los datagramas destinados a direcciones dentro de este rango nunca serán encaminados por routers multicast.
- El rango de direcciones 239.0.0.0 - 239.255.255.255 está reservado para usos administrativos. Las direcciones en este rango se asignan de forma local por cada organización pero no se asegura que no existan otras direcciones como esas fuera de la red de la organización. Los routers de la organización no deberán encaminar los datagramas destinados a direcciones dentro de este rango fuera de la red corporativa.

### 5.3.- PROTOCOLO IGMP

IGMP es usado por máquinas y routers que soportan multicasting. Informa a la red física sobre qué máquinas pertenecen actualmente a un grupo multicast. Esta información es requerida por los routers para saber cuando reenviar un datagrama multicast.

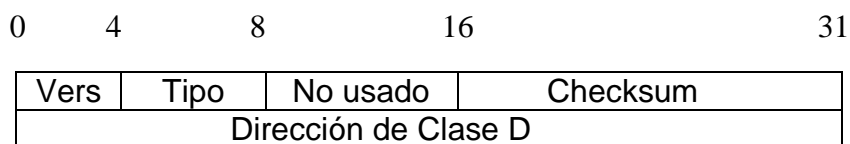
El protocolo IGMP aunque se vale de datagramas IP para transportar mensajes, pensamos en éste como una parte integral del IP, no como un protocolo separado.



IGMP funciona así: se asigna una dirección del rango de clase D al flujo multidifusión. Cualquier equipo que quiera recibir el flujo de datos colocará la dirección IP clase D de dicho flujo en cualquier interfaz que utilice para IP. Puesto que todos los clientes del flujo tienen la misma dirección clase D, la multidifusión se envía a una sola dirección y a muchos clientes. Los encaminadores multidifusión utilizan el protocolo de pertenencia a un grupo, como el IGMP (Internet Group Management Protocol o protocolo de gestión de grupos de Internet), para averiguar los equipos conectados a las subredes. Cuando un equipo desea unirse a un grupo, envía un mensaje IGMP al encaminador multidifusión, indicándole las sesiones que desea recibir. El encaminador multidifusión empieza a difundir las sesiones solicitadas a los miembros de una subred y cada miembro añade la dirección de identificación de grupo a su interfaz para empezar a recibir datos. La escalabilidad se incrementa a medida que más miembros se unen, ya que hay más posibilidades de localizar un encaminador multidifusión cerca de una red de flujo ascendente.

### 5.3.1.- MENSAJES IGMP

Los mensajes IGMP se envían en datagramas IP. La cabecera IP tendrá siempre un número de protocolo de 2, indicando IGMP y un tipo de servicio de cero (rutina). El campo de datos IP contendrá mensaje IGMP de 8 bytes con el formato mostrado en la figura que se muestra a continuación.



donde:

- Vers: Versión IP de 4 bits. Siempre 1.
- Tipo: Identificador del tipo de datagrama IGMP.
- Checksum: Información para chequeo de errores de transmisión, se realiza una suma complemento a 1 del mensaje IGMP, tal y como ocurría en ICMP.
- Dirección de clase D: Dirección de grupo multicast

Los principales tipos de mensajes son tres:

1) *Petición de miembros de un grupo*: mensaje enviado desde el mrouter a los hosts de su subred para preguntarles si quieren apuntarse a un grupo. Se utiliza una subred con capacidad multicast, usando la dirección IP 224.0.0.1 para preguntar a todos los sistemas multicast de la subred. Este tipo de mensaje lo envía el mrouter periódicamente, y espera las respuestas de los hosts de la subred, configurando sus tablas de encaminamiento multicast con la información recibida. Se mantiene un grupo apuntado en las tablas mientras se reciba respuesta de algún hosts de este grupo.

Los datos de los datagramas IP e IGMP de la petición son:

IGMP tipo = 1

IGMP dirección de grupo = 0

IP TTL = 1

IP dirección destino = 224.0.0.1 (a todos los hosts de la subred)

IP dirección fuente = la del router

2) *Informe de miembros de grupo*: mensaje de respuesta al anterior, desde los hosts al mrouter para informar que se quiere ser miembro de un grupo. Para no colapsar al mrouter, los hosts esperan un tiempo aleatorio (entre 0 y 10 segundos) antes de responder a la petición de miembros. Además si un host observa que otro de su grupo ya ha enviado un informe al mrouter, no es necesario que este envíe el suyo, ya que lo importante es que el mrouter se entere de que hay alguien en la subred que pertenece a un grupo determinado. La dirección de destino de un datagrama de informe es la del grupo,

de esta forma esta información llega a todos los miembros del grupo en la subred, además de llegar al mrouter.

Los datos de los datagramas IP e IGMP del informe son:

IGMP tipo = 2

IGMP dirección de grupo = dirección de grupo

IP TTL = 1

IP dirección destino = dirección de grupo

IP dirección fuente = dirección IP del host

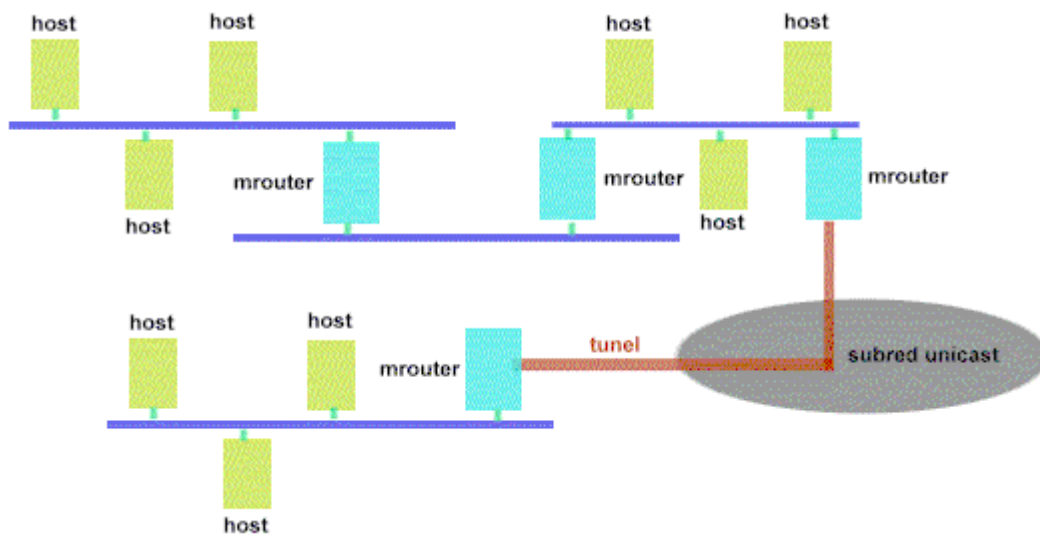
3) *Mensaje DVMRP (Distance Vector Multicast Protocol)*: mensaje enviado por los mrouter a sus vecinos para comunicarles los cambios habidos sobre miembros de grupos. Estos mensajes se pueden enviar bajo dos situaciones distintas:

**Mrouters vecinos conectados directamente con una subred multicast:** utilizan la dirección reservada 224.0.0.4.

**Mrouters vecinos conectados a través de un túnel:** transmiten datagramas IP multicast encapsulados dentro de datagramas IP unicast, que contiene en su cabecera la dirección de destino unicast del otro extremo del túnel.

Se pueden utilizar otros protocolos multicast como PIM y MOSPF que llevarían asociados sus propios tipos de mensajes. Estos tres tipos de mensajes no son propagados más allá de su subred. Utilizan un tiempo de vida (TTL) con valor 1. Si un datagrama tienen el campo TTL a cero es privado para el host origen, si el TTL es dos todos los hosts que sean miembros del grupo y todos los routers multicast reciben el datagrama, y los datagramas con otros valores para la dirección de destino los envía el router multicast como normales: decrementa el valor de TTL al menos un segundo.

En la siguiente figura se muestra un ejemplo de topología de red multicast con las distintas variantes que se han apuntado.



Esta figura se observa subredes multicast (color azul) conectadas directamente a través de m routers y la posibilidad de conexión de subredes aisladas mediante túneles por medio de redes IP unicast.

#### 5.4.-PROTOSCOLOS DE ROUTING

La multidifusión adopta una de las dos tecnologías de Spanning-Tree: Dense mode (modo denso) o Sparse Mode (modo disperso).

##### *5.4.1.-MODO DENSO:*

Los miembros del grupo están agrupados densamente en la red. Se emplean en grandes distribuciones, donde muchos equipos situados en la misma red o subred reciben los datos desde la misma localización. El modo denso también supone que el ancho de banda es suficientemente grande para soportar la transmisión. Protocolos de encaminamiento para distribuir datos en redes de modo denso, tenemos el DVMRP, el PIM-DM y el MOSPF.

DVMRP (Distance Vector Multicast Routing Protocol o protocolo de encaminamiento multimedia de vector distancias): Adopta la inundación como método para hacer llegar los datos multidifusión a su destino, aunque puede necesitar de un gran ancho de banda. Los encaminadores DVMRP asumen q todo el mundo conectado a la subred quiere recibir los datos. Este tipo de extensión del árbol lleva la información a todas las “hojas del árbol” de la forma mejor y más rápida. A medida que los miembros se unan o dejen el grupo, los encaminadores eliminarán las ramas del árbol donde no haya miembros, reduciendo de esta manera el ancho de banda utilizado.

DVMRP depende del camino más corto para su propagación. Los encaminadores DVMRP comprueban sus tablas de encaminamiento para determinar si tienen mejor ruta hacia el siguiente encaminador multidifusión. Comparando sus tablas de encaminamiento, los encaminadores DVMRP crean un camino eficiente para la transmisión de los datos. Si un encaminador determina mediante el protocolo IGMP que no tiene miembros a los que transmitir o no tiene el mejor camino, pedirá ser eliminado de la transmisión. Este protocolo emplea la técnica de broadcast para actualizar todos los encaminadores de la red.

PIM-DM (Protocolo Independent Multicast Dense Mode o multidifusión independiente del protocolo -Modo denso): Similar al DVRMP en su funcionamiento general, PIM-DM es la versión de modo denso del protocolo PIM, creado para proporcionar un protocolo de encaminamiento multidifusión estándar y escalable. La forma de propagación de los paquetes de datos de PIM-DM consiste en que cuando llega un paquete al encaminador, PIM-DM determina si se está utilizando el camino más corto hacia el origen. Si es así, el paquete se envía de forma descendente a todos los interfaces hasta que llegan

a los miembros, y las ramas que no se usan se eliminan. El funcionamiento es simple, pero puede generar sobrecarga y duplicación de paquetes.

MOSPF(Multicast Open Shortest Path First o primero el camino más corto abierto multidifusión): Es una extensión del OSPF orientada a tratar el tráfico multidifusión en lugar de dedicarse exclusivamente al de unidifusión. MOSPF encamina los datos por las conexiones de coste más bajo con un ancho de banda disponible, y el menor número de saltos se utiliza como criterio para determinar el mejor camino. Empleando este método, las rutas más congestionadas pueden evitarse si se les asigna un coste más elevado.

Cada encaminador MOSPF mantiene una visión completa de toda la red, creada a partir de la información del estado de los enlaces que los encaminadores intercambian entre sí. Esto puede limitar la escalabilidad, ya que los encaminadores, además de intercambiar información sobre el estado de los enlaces, envían datos IGMP sobre los miembros, reduciendo de esta manera el ancho de banda disponible para la transmisión en redes con muchos grupos de miembros. A medida que el árbol de multidifusión se va creando, cada encaminador MOSPF lleva a cabo una serie de cálculos para determinar el mejor camino para los paquetes. Sin embargo, esto se hace una vez para cada uno de los grupos, lo que mantiene la sobrecarga de la red en cotas bajas. A medida que el paquete atraviesa la red, cada encaminador realiza los mismos cálculos y crea el árbol final para los miembros.

#### *5.4.2.- MODO DISPERSO:*

El propósito es encontrar formas eficientes de llevar los datos a mucha gente que se encuentra dispersa en grandes áreas. Al contrario que el modo denso, que supone que hay un miembro en cada rincón de la red, el modo disperso asume que para transmisiones especializadas, los miembros están repartidos en pequeños grupos de la red. Los protocolos Sparse Mode también se han diseñado para funcionar bien sobre conexiones congestionadas o con un ancho de banda reducido.

Existen dos protocolos de modo disperso: CBT y PIM-SM. Ambos construyen el árbol de encaminamiento pidiendo a los encaminadores que participen en la creación del mismo. Los encaminadores de modo disperso se unen a una sesión multidifusión cuando un miembro solicita ser admitido.

CBT(Center-Based Trees o árboles basados en un centro): Simplifica el enfoque creando un árbol que es utilizado por todos los grupos y emplea una estructura de árbol basada en un encaminador central, desde el que fluyen los datos, a parte de la propia fuente de datos (servidor). Esto es ventajoso, ya que la información del estado del enlace se reduce, pues todos los grupos usan el mismo árbol. Por el contrario, el encaminador central se irá sobrecargando a medida que se vayan incorporando más miembros.

PIM-SM(Protocol Independent Multicast Sparse Mode o multidifusión independiente del protocolo -Modo disperso): Es parecido al CBT en su

topología, pero es más flexible. En lugar de un encaminador central, PIM-SM utiliza un RP (Rendevous Point o punto de encuentro), en el que los encaminadores descendentes se “reúnen”. Esto permite al PIM-SM crear un árbol compartido o uno basado en el camino más corto. De esta manera, cada grupo puede tener una estructura de árbol diferente dependiendo de cual funcione mejor.

CBT mantiene baja la cantidad de información sobre el estado de los enlaces, aunque podría no proveer la mejor ruta hacia un miembro. Si lo que se necesita es una baja latencia, PIM-SM puede crear mejores rutas.

Puesto que PIM es un protocolo estandarizado, es posible la interoperatividad entre PIM-DM y PIM-SM.

# PROTOCOLOS EXTERIORES

## 6.- PROTOCOLOS EXTERIORES

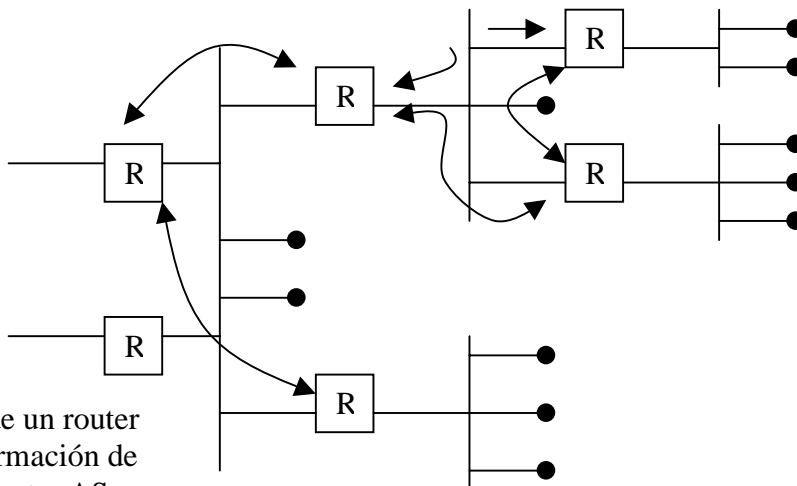
Es la interconexión de sistemas autónomos.

### 6.1.-SISTEMAS AUTÓNOMOS

Para permitir a cada red intercambiar su información de routing y definir su política de intercambio de tráfico se ha creado el concepto de Sistema Autónomo. Entendemos por Sistema Autónomo (AS, Autonomous System) la subred que es administrada o gestionada por una autoridad común, que tiene un protocolo de routing homogéneo mediante el cual intercambia información en toda la subred, y que posee una política común para el intercambio de tráfico con otras redes o sistemas autónomos.

Cada AS tiene asignado un número de identificación que lo diferencia en la Internet, estos números los asigna la misma autoridad que proporciona las direcciones IP.

En un AS existe un camino para cualquier par de nodos, excepto cuando hay fallos temporales.



Puede haber más de un router que transfiera información de accesibilidad hacia otra AS.

Esquema de un Sistema Autónomo

Las subredes ocultas dentro de un AS son accesibles desde Internet gracias a la difusión de información de accesibilidad que cada AS difunde hacia otras AS.

Como mínimo en la Internet se dan dos niveles jerárquicos de routing, el que se realiza dentro de un sistema autónomo (AS) y el que se efectúa *entre* sistemas autónomos. Al primero lo denominamos routing interno, o routing en el interior de la pasarela (pasarela es una antigua denominación de router). Al routing entre sistemas autónomos lo denominamos routing externo, o también routing exterior a la pasarela. Dado que los requerimientos en uno y otro caso son muy diferentes, se utilizan protocolos de routing distintos en uno y otro caso. Los protocolos de routing dentro del sistema autónomo se denominan

IGP (Interior Gateway Protocol), mientras que los utilizados entre sistemas autónomos se llaman EGP (Exterior Gateway Protocol).

Para conectar los diferentes sistemas autónomos TCP consta de dos protocolos:

EGP -> Exterior Gateway Protocol  
BGP -> Border Gateway Protocol

## 6.2.- PROTOCOLO BGP

El protocolo BGP (*Border Gateway Protocol*) es el protocolo de encaminamiento interdominio más utilizado en Internet. Una descripción rigurosa de la versión actual (BGP-4) puede encontrarse en [RFC 1771]. Este protocolo se diseñó para permitir la cooperación en el intercambio de información de encaminamiento entre enrutadores de distintos sistemas autónomos.

BGP construye un gráfico de sistemas autónomos basados en la información intercambiada entre los routers BGP. Este entorno gráfico se denomina en ocasiones árbol. En lo que concierne a BGP, Internet es un gráfico de SA, con cada SA identificado por un número de SA único. Las conexiones entre dos SA juntos forman una ruta de acceso, y el conjunto de información de rutas de acceso forma una ruta para llegar a un destino específico. BGP utiliza la información de ruta de acceso asociada con un destino dado para asegurar el enrutamiento entre dominios libre de bucles.

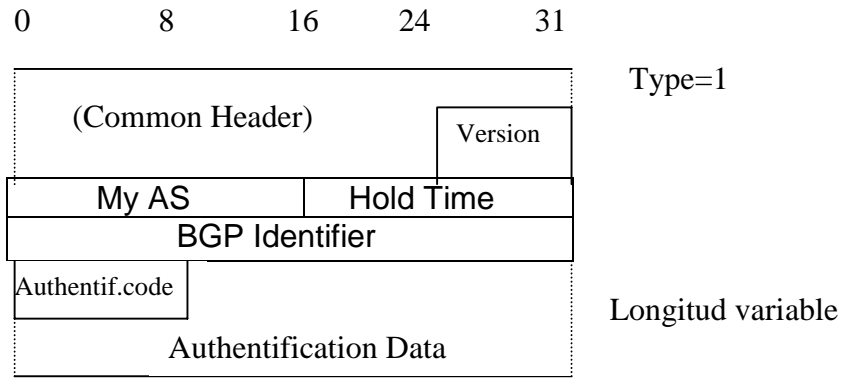
En terminología BGP los enrutadores se denominan pasarelas (*gateways*), y realizan tres procesos funcionales: adquisición de vecinos, detección de vecinos alcanzables y detección de redes alcanzables.

El término *adquisición de vecinos* implica que dos dispositivos de encaminamiento que comparten la misma subred física, pero que pertenecen a distintos sistemas autónomos, deciden intercambiar regularmente información de encaminamiento. Se requiere un procedimiento formal para la adquisición ya que uno de los dos dispositivos puede decidir no participar. En este procedimiento un dispositivo hace una oferta a otro (mediante un mensaje *open*), el cual puede aceptarla (mediante un mensaje *keepalive*) o rechazarla.

Una vez establecida la relación de vecindad, se utiliza el procedimiento de *detección de vecino alcanzable* para mantenerla. Cada miembro necesita estar seguro de que su pareja existe y está todavía comprometida con la relación. Para este propósito, periódicamente ambos dispositivos de encaminamiento se envían mensajes *keepalive*.

El último procedimiento es la *detección de redes alcanzables*. Cada pasarela mantiene una base de datos con las subredes que puede alcanzar y la ruta completa para hacerlo. Siempre que se modifica esta base de datos, la



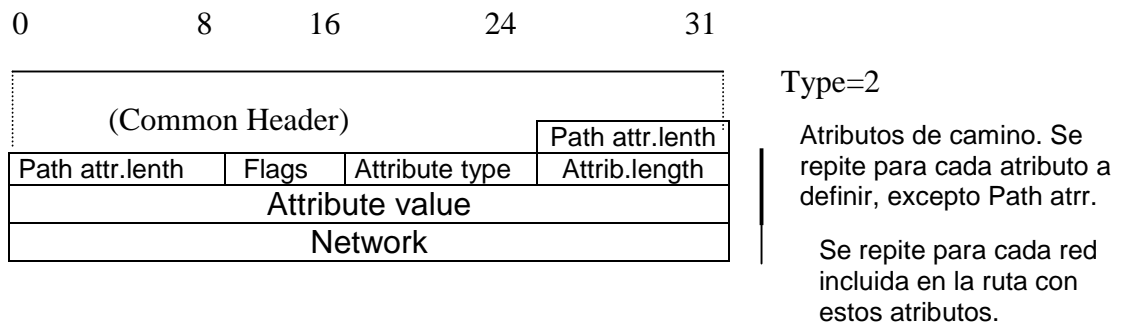


- Version: versión del protocolo, actualmente la 4.
- My autonomous system: indica el numero de AS del emisor.
- Hold time: tiempo que tiene que esperar el receptor antes de asumir que el emisor a caído. El emisor debe continuar emitiendo paquetes antes de que este tiempo se agote.
- BGP identifier: dirección IP del emisor. BGP considera como identificador de cada enrutador su propia dirección IP.
- Authentication code: define el sistema de autenticación empleado. En la actualidad este campo se asigna a cero.
- Authentication data: datos destinados a la autenticación del paquete. La longitud y el contenido de este campo dependen del campo anterior. De momento este campo no se utiliza y tiene una longitud de cero bytes.

Un mensaje *update* facilita dos tipos de información:

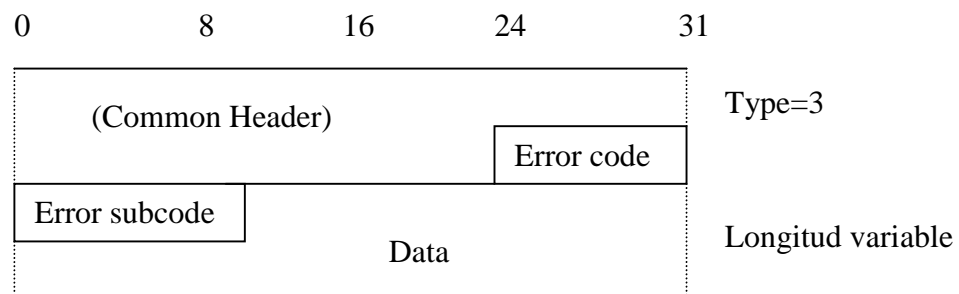
- 1) Información sobre una ruta particular a través del conjunto de redes. Dicha ruta se incorpora a la base de datos de cada dispositivo de encaminamiento que la recibe.
- 2) Una lista de rutas que fueron previamente anunciadas por este dispositivo de encaminamiento, y que ahora han sido eliminadas.

Estos dos tipos de información pueden ser proporcionadas simultáneamente en un único paquete. El formato del paquete *update* se muestra en la siguiente figura. El contenido de cada campo se describe a continuación:



- Path attributes length: longitud de rutas no factibles. Número de atributos de la ruta.
- Flags: diversos bits que indican la opcionalidad, transitividad y parcialidad del atributo.
- Attribute type: existen cinco tipos de atributos:
  - 1 = *origin*; el atributo ocupa 1 byte e indica si la información fue generada por un protocolo de encaminamiento interior (como OSPF) o por un protocolo de encaminamiento exterior (en particular, BGP).
  - 2 = *AS path*; el atributo es de longitud variable y enumera una lista de AS que atraviesa la ruta.
  - 3 = *next hop*; el atributo ocupa 4 bytes y proporciona la dirección IP del dispositivo de encaminamiento frontera que se debe usar para alcanzar los destinos indicados en el campo *network*.
  - 4 = *unreachable*; el atributo no ocupa lugar adicional
  - 5 = *inter-AS metric*; el atributo ocupa 2 bytes

El tercer tipo de paquete es el paquete *notification*. Un paquete de notificación es enviado por el enrutador R1 al enrutador R2 para explicar por qué deniega la conexión a R2. Su formato se muestra en la siguiente figura. El contenido de cada campo es el siguiente:



- Error code: para cada código de error existen diversos subcódigos que no detallaremos:
  - 1 = cabecera corrupta. El tipo de paquete es inaceptable, bien por errores de sintaxis o por cuestiones de autenticación.
  - 2 = paquete *open* corrupto
  - 3 = paquete *update* corrupto
  - 4 = tiempo de *hold down* expirado
  - 5 = Error en la máquina de estados finitos (errores de procedimiento)
  - 6 = Cese (cierre voluntario de una conexión)
- Data: contiene el comando ofensivo.

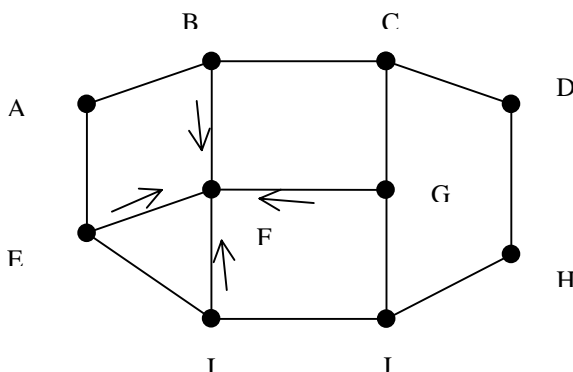
El cuarto y último tipo de paquete es el paquete *keepalive*. Este paquete no tiene otra información aparte de la cabecera mostrada en la siguiente figura. Este paquete se envía para inicializar el temporizador *hold down* del receptor antes de que expire, y el emisor no tiene información de interés que comunicar.

|                    |   |         |         |
|--------------------|---|---------|---------|
| 0                  | 8 | 16      | 31      |
| LS Age             |   | Options | LS Type |
| Link State ID      |   |         |         |
| Advertising Router |   |         |         |
| LS Sequence Number |   |         |         |
| LS Checksum        |   | Length  |         |

- LS age: edad estimada de la información (expresada en segundos).
- LS type: los valores posibles son los siguientes:
  - 1 = enlaces del enrutador
  - 2 = enlaces de la red
  - 3 = resumen de enlaces (subredes IP alcanzables)
  - 4 = resumen de enlaces (enrutadores alcanzables de sistemas vecinos)
  - 5 = resumen de enlaces (subredes IP alcanzables de sistemas vecinos)
- Link state ID: su significado depende del valor del campo anterior:
  - Type = 1 el ID del enrutador que generó la información
  - Type = 2 la dirección IP del enrutador designado de la LAN
  - Type = 3 la dirección IP del enlace que conecta la subred
  - Type = 4 el ID del enrutador de borde
  - Type = 5 la dirección IP del enlace que conecta la subred
- Advertising router: ID del enrutador que generó la información.
- LS sequence number: número de secuencia del LSA.
- LS checksum: código de redundancia ISO 8473 (anexo C).
- Length: tamaño en bytes.

BGP es un protocolo de 'VECTOR DE CAMINOS'. No mantiene el coste a cada destino, lleva el registro de la trayectoria seguida. Cada router informa a sus vecinos BGP sobre la trayectoria exacta que está utilizando.

Ejemplo:



Routers con BGP

- En la tabla de F, la ruta para llegar a D es:  
FGCD

- F recibe de sus vecinos acerca de D:

B – utilizó BCD

G – utilizó GCD

I – utilizó IFGCD] las descarta porque lo

E – utilizó EFGCD] atraviesan a él mismo.

Como vemos nos dan la trayectoria exacta de la ruta y no sólo la distancia.

- La decisión pasa por utilizar B o G.

BGP examina las rutas y las pondera, si se viola alguna restricción por 'política', recibe un peso  $\infty$ . El router selecciona la más corta. La función de ponderación no es parte de BGP sino que la definen los administradores del sistema.

BGP no tiene el problema de conteo a  $\infty$  típico, ya que si G cae o se desactiva la línea FG, F recibe rutas de los vecinos restantes: BCD – IFGCD – EFGCD. Las dos últimas rutas las descartan porque pasan por F, y seleccionan la ruta BCD quedando FBCD como trayectoria exacta.