



PARTE I. FUNDAMENTOS DE REDES, REDES LAN Y WAN

CAPÍTULO

6

- 6.1. Introducción
- 6.2. Conceptos y técnicas de conmutación
- 6.3. Encaminamiento
- 6.4. Interconexión de redes

ENCAMINAMIENTO E INTERCONEXIÓN DE REDES

6.1. Introducción

Si bien las distintas tecnologías de redes LAN presentadas a lo largo del capítulo precedente resultan de gran interés por cuanto que constituyen el entorno de trabajo directo de los usuarios finales, no debemos perder de vista que la verdadera relevancia actual de cualquier sistema de red radica en su coexistencia e intercomunicación con otros. Según se establece en el modelo OSI, es la capa de red la encargada de permitir tal intercomunicación de equipos y sistemas, en lo que viene a denominarse *interconexión de redes*.

Antes de abordar esta cuestión central se discutirá otra necesaria para posibilitar en la práctica la primera: la de *encaminamiento*. La función de encaminamiento («routing» en inglés) se refiere a la obtención de un camino o ruta entre un *host* emisor y un *host* receptor sobre el que llevar a cabo la transferencia de la información entre ambos. A diferencia de como suele suceder en una red LAN, donde no existe más que un único medio común para la transmisión entre cualesquiera dos estaciones finales, en una interconexión (sea esta de redes o de equipos en una misma WAN) debe buscarse una ruta de la forma *estación_final_1-nodo_1-nodo_2-nodo_3-...-estación_final_2* (Figura 1.4).

Adicionalmente a los dos aspectos referidos, los cuales se desarrollarán a lo largo de los apartados tercero y cuarto del capítulo, comenzará este estudiando el concepto de *conmutación* y los distintos tipos de técnicas existentes al respecto. Esta cuestión es la base para comprender la forma en que se lleva a cabo el reenvío de información salto-a-salto en una red de computadores y las características más relevantes del mismo.

6.2. Conceptos y técnicas de conmutación

A las redes que siguen una estructura como la mostrada en la Figura 1.4, en las que es preciso una conmutación entre nodos a fin de establecer la ruta adecuada para las comunicaciones entre un origen y un destino, se les llama *redes conmutadas*. Dos son los tipos de técnicas de conmutación básicas existentes en la actualidad:

1. *Comutación de circuitos*: en este caso se establece una conexión previa a la transferencia de la información entre las estaciones finales origen y destino. Una vez establecida la conexión, todos los datos que forman el mensaje se transmiten de forma secuencial siguiendo la misma ruta o circuito. Una vez concluida la comunicación se procederá al cierre de la conexión.
2. *Comutación de paquetes*: frente al de circuitos, en este esquema el mensaje no se transfiere secuencialmente como una sola unidad sino en trozos denominados *paquetes*, los cuales se retransmiten nodo a nodo hasta alcanzar el destino. Dentro de esta técnica de conmutación existen dos variantes: *datagrama*, en la que no se establece una conexión origen-destino previa a la transmisión, y *circuitos virtuales*, en la que, de forma similar a como sucede en conmutación de circuitos, sí se establece tal conexión.

Seguidamente se describen en mayor detalle las técnicas mencionadas.

6.2.1. Comutación de circuitos

La técnica de *comutación de circuitos* es la utilizada típicamente en una comunicación telefónica, caracterizada por el establecimiento de una conexión origen-destino con carácter previo a la transmisión de la información; conexión que se debe cerrar una vez transferidos los datos. Los pasos, por tanto, involucrados en una comunicación de esta naturaleza son tres:

1. *Establecimiento de conexión*: proceso correspondiente a la elección de la ruta origen-destino a seguir por el mensaje (Figura 6.1), lo cual implica la reserva de los recursos necesarios para posibilitar la comunicación pretendida.
Este paso correspondería a los procesos comprendidos entre el marcado en un extremo y la constatación del descolgado del teléfono en el otro en una comunicación telefónica.
2. *Transmisión*: es el intercambio de datos propiamente dicho entre las estaciones finales de origen y destino. A este respecto hemos de decir que el mensaje se transfiere en forma secuencial, desde el principio hasta el fin, siguiendo la ruta fijada en el proceso de establecimiento de la conexión.

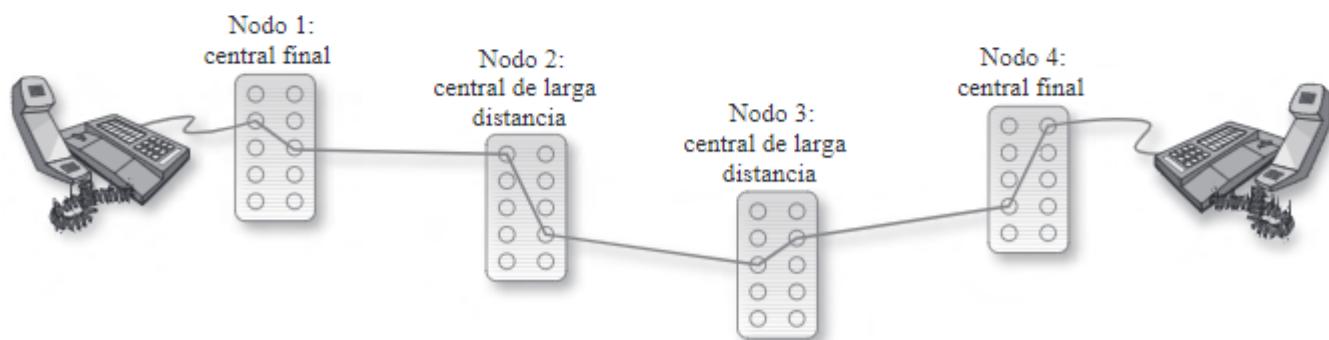


Figura 6.1. Ejemplo de conexión telefónica mediante conmutación de circuitos.

En el ejemplo de una comunicación telefónica, el proceso de transmisión se corresponde con la conversación telefónica como tal.

3. *Cierre de conexión*: proceso que consiste en la liberación de los recursos (ancho de banda típicamente) de la subred asociados a la conexión utilizada.

Es fácil constatar que este tercer y último paso está relacionado con el colgado del teléfono en el caso de una comunicación telefónica.

De acuerdo con lo anteriormente establecido, en la Figura 6.2 se muestra como ejemplo la evolución temporal de una comunicación basada en conmutación de circuitos con dos nodos intermedios entre las estaciones finales origen y destino. En ella se siguen los siguientes pasos:

- a) La estación final emisora genera un mensaje de solicitud de establecimiento de llamada dirigido a la estación de destino. Dicho mensaje se enviará en primera instancia hacia el nodo al que se encuentra conectada la estación origen.
- b) Dicho nodo debe decidir el siguiente nodo en la ruta en base a la dirección de destino especificada en el mensaje de solicitud. Este proceso de decisión de ruta tendrá lugar en todos los nodos siguientes hasta alcanzar el nodo final al que se encuentra conectada la estación de destino. Además, salto a salto, se deberán reservar y asociar los recursos (ancho de banda) necesarios para dar cabida a la comunicación deseada.
- c) La estación destino responde generando un mensaje de aceptación de llamada hacia el origen. Dado que la ruta ya se encuentra establecida y los recursos correspondientes reservados tras el paso b), el retorno de la respuesta no sufrirá demora en los nodos intermedios salvo, claro está, la debida a la propagación de las señales.
- d) La recepción de la confirmación en la estación final emisora significa el establecimiento efectivo de la conexión. En este punto, el emisor y el receptor están en disposición de intercambiarse

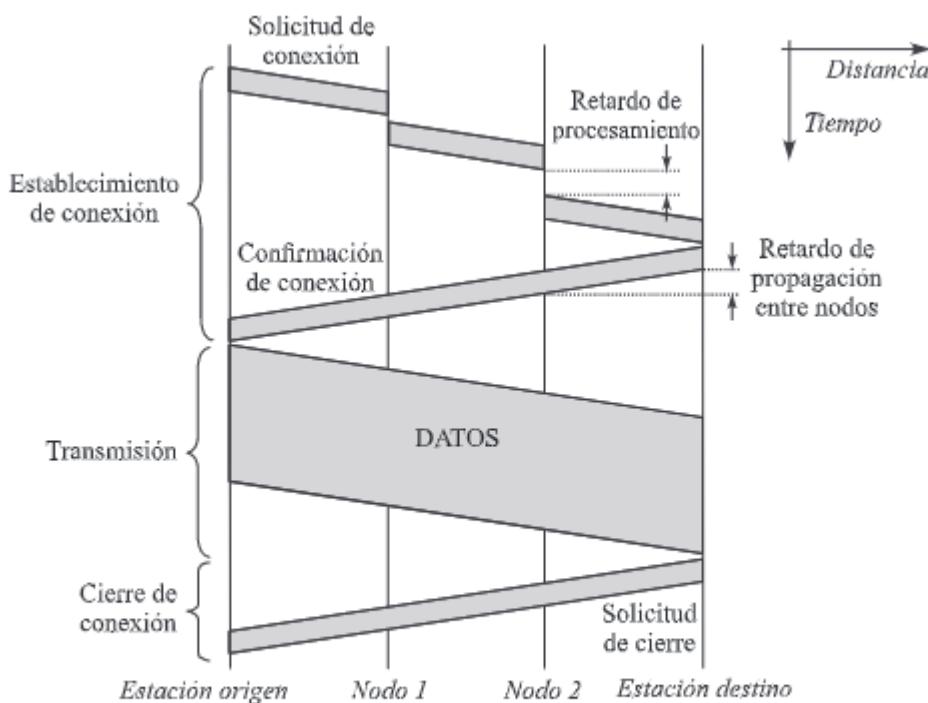


Figura 6.2. Esquema de transmisión mediante conmutación de circuitos entre dos estaciones finales con dos nodos intermedios.

la información correspondiente sobre el circuito establecido en forma secuencial, lo que significa que: (1) no se produce retardo en los nodos intermedios salvo el propio de propagación, y (2) los datos se reciben en el mismo orden con el que son transmitidos.

- e) Finalizada la transferencia de los datos se procederá al cierre de la conexión y a la correspondiente liberación de los recursos en los nodos intermedios. Ello se lleva a cabo a través del envío por parte de uno de los extremos de un mensaje de solicitud en este sentido.

6.2.2. Comutación de paquetes

La conexión referida en la comutación de circuitos implica la reserva estática de recursos de la subred (acceso a las líneas de transmisión, procesamiento de los nodos). Es decir, dichos recursos no pueden ser usados de forma simultánea por otra comunicación; y ello independientemente de si se utilizan de forma efectiva o no. Por ejemplo, en el caso de una conversación telefónica normal se estima que la línea permanece desocupada en torno al 50% del tiempo; esto es, ¡el canal dispuesto se desaprovecha en torno a la mitad!

Para evitar la infroutilización que de los recursos de la subred puede producirse en el esquema de comutación de circuitos, surge la técnica de *comutación de paquetes*. En su versión más genuina, conocida como *comutación de paquetes mediante datagramas*, el proceso de transmisión seguido se fundamenta en los tres principios siguientes (Figura 6.3(a)):

- a) No se establece conexión previa a la transmisión de los datos.
- b) El mensaje se divide en bloques más pequeños denominados *paquetes* o, derivado del nombre de la técnica, *datagramas*.
- c) Cada paquete se transmite independientemente del resto, de forma que pueden seguir caminos distintos y, en consecuencia, recibirse de forma desordenada en el receptor.

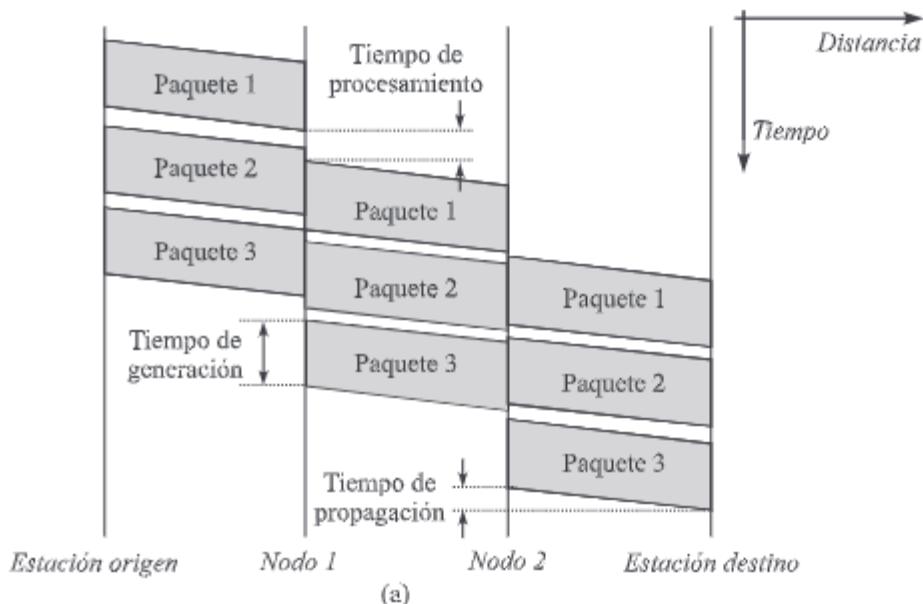
Para posibilitar su envío individualizado cada paquete debe contener información diversa, como el destino al que va dirigido.

- d) La transmisión de los paquetes se realiza en forma de almacenamiento y reenvío. Es decir, los nodos intermedios deben almacenar temporalmente cada paquete y procesarlo antes de llevar a cabo su retransmisión. Ello implica un tiempo de procesamiento en la retransmisión de todos y cada uno de los paquetes de que consta el mensaje.

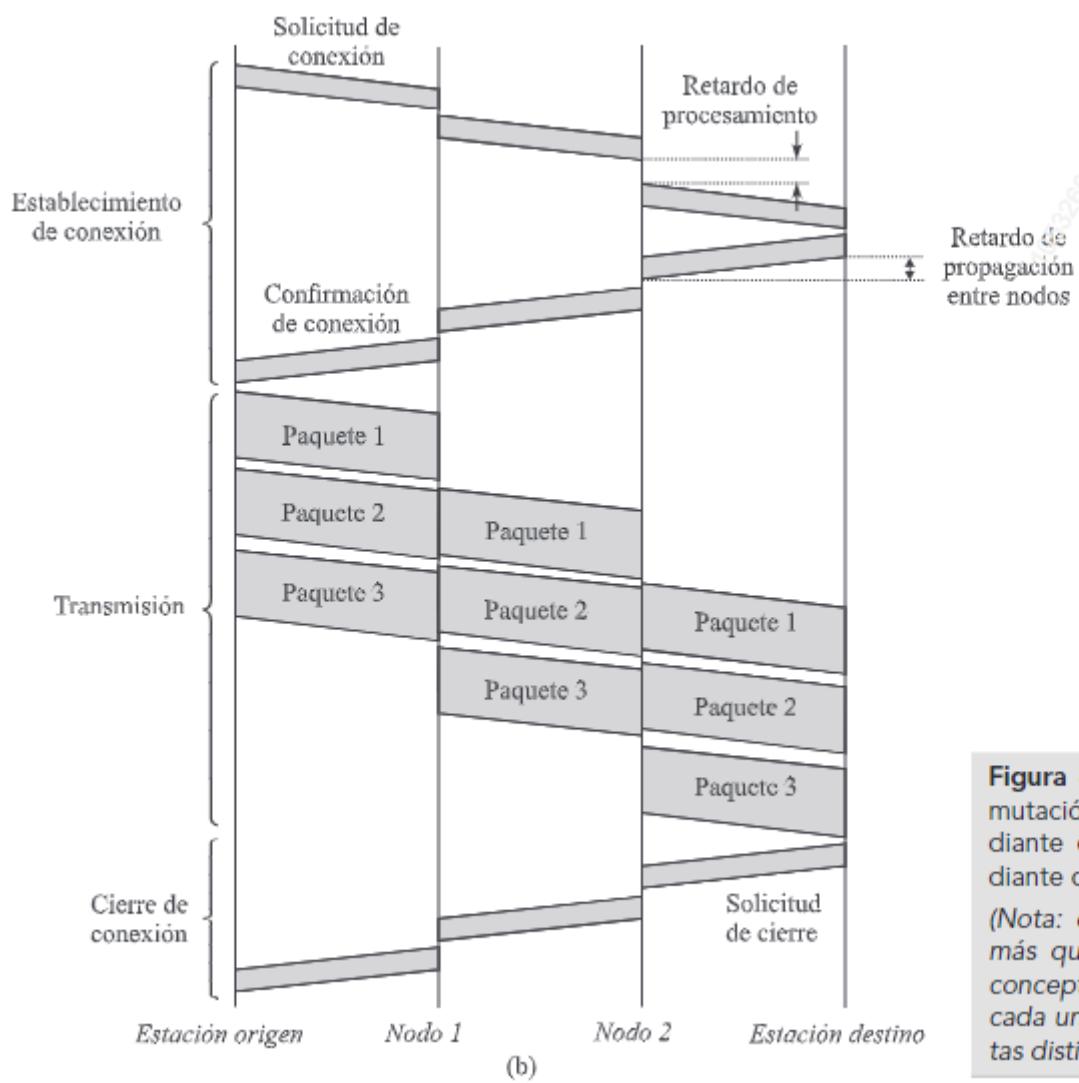
El esquema de comutación de paquetes mediante datagramas presenta como principal ventaja la robustez en la transmisión, al permitir el uso potencial de rutas alternativas para los distintos paquetes. Sin embargo, debido a los tiempos de procesamiento intermedios implicados en los reenvíos, resulta poco adecuado para servicios interactivos en los que se precisa una alta velocidad. Así, frente a esta técnica, existe la variante de *comutación de paquetes mediante circuitos virtuales*, la cual supone una mezcla entre la de datagramas y la de comutación de circuitos estudiada en el apartado anterior. En la Figura 6.3(b) se muestra el esquema temporal de transmisión correspondiente a este caso, siendo el proceso seguido el que se resume a continuación:

- a) La estación final emisora genera un mensaje de establecimiento de llamada hacia el nodo al que se encuentra conectada. Dicho nodo debe decidir el siguiente en la ruta en base a la dirección de destino especificada en el mensaje de solicitud. Este proceso de decisión de ruta se repetirá en todos los nodos siguientes hasta alcanzar el nodo final al que se encuentra conectada la estación de destino.

Frente al carácter dedicado de los recursos asociados a una conexión en el caso de la comutación de circuitos, en circuitos virtuales estos pueden ser compartidos en el tiempo entre varias comunicaciones. Es por ello que se llama virtual a la ruta establecida, constituyendo



(a)



(b)

Figura 6.3. Ejemplo de conmutación de paquetes: mediante datagramas (a) y mediante circuitos virtuales (b).

(Nota: el esquema (a) no es más que una representación conceptual, pudiendo seguir cada uno de los paquetes rutas distintas.)

así lo que se conoce como *circuito virtual*. Podemos decir, así, que en este caso no se realiza reserva de recursos más allá de la asignación de una línea de salida para la comunicación solicitada. De este modo, cuando un nodo intermedio desee llevar a cabo la retransmisión de un paquete correspondiente a una comunicación dada, primero deberá comprobar que la línea de salida asociada no está siendo ocupada por otra comunicación; en caso contrario, deberá esperar a que quede libre.

- b) La estación destino responde con un mensaje de aceptación de llamada hacia el origen. A diferencia de como sucede en la conmutación de circuitos, ahora la respuesta también sufrirá retardos en los nodos intermedios debido a que los recursos a que se refiere el circuito virtual establecido pueden estar siendo usados por otras comunicaciones, en cuyo caso habría que esperar, como hemos señalado anteriormente, a que quedasen libres. También a diferencia de conmutación de circuitos, es posible que el camino de vuelta sea diferente al de ida si los circuitos no son dúplex. En este caso, evidentemente, los nodos intermedios deben decidir también la ruta de vuelta y reservar los circuitos al realizar la retransmisión del mensaje de aceptación de la conexión.
- c) Dividido el mensaje en paquetes, cada uno de ellos se transmite siguiendo la misma ruta como en el caso de conmutación de circuitos, aunque en forma de almacenamiento y reenvío como en el esquema datagrama.

Cada paquete debe contener un identificador de camino virtual (de significado totalmente local) a través del que se posibilitará la retransmisión o conmutación adecuada del mismo hacia el nodo siguiente.

Como ya hemos señalado, todos los paquetes siguen la misma ruta, lo que significará su recepción ordenada en el receptor, pero eventualmente podrán producirse retardos en los nodos debido al carácter virtual y consecuente potencial utilización temporal de esta por parte de otras comunicaciones. Es de esperar, sin embargo, que todos estos retardos de procesamiento sean inferiores a los involucrados en la conmutación de datagramas.

- d) Finalizada la transmisión se llevará a cabo el cierre de la conexión, proceso que en este caso no consiste más que en la eliminación de la asociación comunicación-línea de salida o identificador de circuito virtual en cada nodo intermedio.

Aunque de forma breve, es conveniente señalar también la existencia de dos tipos de circuitos virtuales y datagramas desde la perspectiva de una red de paquetes: *internos* y *externos*. Los primeros se refieren al funcionamiento intrínseco de la red, mientras que los segundos se utilizan externamente a la red e independientemente de esta. Así, por ejemplo, en Internet se utilizan datagramas internos (protocolo de red IP, no orientado a conexión), mientras que la comunicación extremo-a-extremo entre los usuarios, externa a la subred, se establece en base a circuitos virtuales (protocolo de transporte TCP, orientado a conexión).

Así mismo hemos de mencionar la existencia de circuitos *permanentes*. Estos se caracterizan por no precisar establecimiento, sino que, como indica su nombre, se encuentran disponibles de forma permanente a lo largo del tiempo. Su uso principal es servir de canales de control y supervisión de la red en la que se utilizan.

Combinación de los esquemas de conmutación de circuitos y de paquetes mediante datagramas es la transmisión de datos mediante la ya en desuso técnica de *comutación de mensajes*, la cual presenta las siguientes características reseñables:

- a) El mensaje se transmite completo, sin fragmentarse.
- b) Sin precisar el establecimiento de una conexión previa a la transmisión, el mensaje se intercambia entre nodos adyacentes en forma de almacenamiento y reenvío tras realizarse en cada uno de ellos el procesamiento pertinente.

De este modo, la técnica de conmutación de mensajes es similar a la de paquetes mediante datagramas con la salvedad de que el mensaje se transmite en un único paquete. Ello presenta la ventaja de ser más rápida que la de datagramas al tiempo que más eficiente que las de circuitos, siempre y cuando la probabilidad de error en las transmisiones sea despreciable.

6.2.3. Comparación de las técnicas de conmutación

Una vez estudiados los distintos tipos de técnicas de conmutación, en este apartado se presenta una comparación de las mismas de acuerdo a las siguientes tres características:

- *Ancho de banda*, esto es, qué cantidad de recursos de la subred se ven involucrados en la transmisión y cómo de eficientemente son utilizados.
- *Complejidad de los nodos intermedios*, es decir, capacidades que deben presentar los nodos de conmutación de la subred para llevar a cabo la transmisión de la información origen-destino.
- *Latencia/retardo*, factor importante por cuanto que a partir de él se establece si el esquema en cuestión es apto o no para su uso en aplicaciones interactivas o de tiempo real.

De acuerdo con las especificaciones dadas en los apartados anteriores para los esquemas de conmutación, las prestaciones de cada uno de ellos desde el punto de vista de las características referidas son como sigue:

- *Commutación de circuitos*:

- *Ancho de banda*:

- a) Se utilizará el necesario para la transmisión de los datos que componen el mensaje, sin bits adicionales salvo los correspondientes al establecimiento y cierre de la conexión. Así, la capacidad efectiva (bits de datos frente al total) requerida para la transmisión pretendida será:

$$\text{capacidad efectiva} = \frac{M}{M + Se + Ce + Sc} \quad (6.1)$$

donde M es la longitud del mensaje a transmitir y Se , Ce y Sc la de los paquetes de solicitud de establecimiento de conexión, de confirmación de establecimiento de conexión y de solicitud de cierre de conexión, respectivamente.

- b) Uso estático del ancho de banda, lo que se traduce en su potencial desperdicio al tratarse de un canal dedicado. Es decir, los recursos se reservan al establecer la conexión y no se liberan hasta que esta concluya, independientemente de si se utilizan de forma efectiva o no. De este modo, podemos aproximar la eficiencia de uso al porcentaje de uso efectivo de la línea:

$$\text{eficiencia de uso} = \frac{\text{tiempo usado}}{\text{tiempo reservado}} = \text{porcentaje de uso efectivo} \quad (6.2)$$

- *Complejidad de los nodos intermedios*:

- a) Son dispositivos simples (de tipo electromecánico) en los que, una vez establecido el circuito, este viene a ser como un cable continuo sobre el que se efectúa la transmisión entre las estaciones origen y destino.
- b) No precisan memorias de almacenamiento temporal para los datos a transferir, más allá de la necesaria para los paquetes de gestión de la conexión.

- *Latencia/retardo:*

Una vez establecida la conexión, la transmisión se realiza sin retardos salvo los propios de propagación. Ello hace de esta técnica de conmutación la más adecuada para su uso en servicios interactivos o en tiempo real.

En el ejemplo mostrado al final de la revisión de las técnicas se lleva a cabo una estimación y comparación del tiempo total involucrado en la transmisión de un mensaje mediante circuitos, datagramas y circuitos virtuales.

- *Comutación de paquetes mediante datagramas:*

- *Ancho de banda:*

- a) Superior al estrictamente necesario para la transmisión de los datos de mensaje. Esto es así debido al hecho de que, puesto que no se establece conexión origen-destino y cada paquete se reenvía de forma independiente del resto, cada datagrama precisa bits suplementarios para, al menos: (1) indicar el origen y el destino del mismo, y (2) identificar el número de paquete dentro del mensaje completo a fin de que el receptor pueda reconstruirlo ante su potencial recepción desordenada.

La capacidad efectiva de bits involucrados en la transmisión mediante esta técnica de conmutación será:

$$\text{capacidad efectiva} = \frac{D}{D + H} \quad (6.3)$$

donde D es la longitud del campo de bits de datos en cada paquete enviado y H la longitud de la cabecera o bits suplementarios.

- b) En esta técnica se hace un uso dinámico del ancho de banda; es decir, no se desaprovecha como en el caso de la conmutación de circuitos, sino que los recursos solo se utilizan cuando son necesarios, pudiéndose compartir temporalmente con otras comunicaciones. Puede concluirse desde esta perspectiva que la eficiencia de uso de los recursos en la conmutación mediante datagramas es del 100 %.

- *Complejidad de los nodos intermedios:*

Además de las capacidades de encaminamiento propias de esta conmutación, necesarias para decidir el nodo siguiente en cada caso, los nodos requieren la existencia de memorias internas de almacenamiento temporal a fin de dar cabida a los paquetes de datos recibidos antes de proceder a su reenvío en su camino hacia el destino.

- *Latencia/retardo:*

Debido a que cada paquete se procesa de forma independiente en cada nodo, el tiempo consumido en ello, al que ha de añadirse el de almacenamiento en espera de la disponibilidad de recursos, dificulta su utilización en aplicaciones interactivas o de tiempo real.

Como ya se ha indicado, en el ejemplo más adelante se hace una estimación del tiempo total involucrado en la transmisión de un mensaje mediante datagramas.

- *Comutación de paquetes mediante circuitos virtuales:*

- *Ancho de banda:*

- a) De modo similar al caso de datagramas, el ancho de banda preciso para la transmisión en circuitos virtuales será superior al estrictamente necesario para el envío de los datos

de mensaje. Esto es así debido al hecho de que, adicionalmente a los mensajes de establecimiento y cierre de conexión, cada paquete debe incluir bits suplementarios para indicar el circuito virtual que identifica la ruta sobre la que se efectuará la transmisión. La capacidad efectiva será así:

$$\text{capacidad efectiva} = \frac{M}{M + n_p \times H + Se + Ce + Sc} \quad (6.4)$$

donde M es la longitud del mensaje original, Se , Ce y Sc el tamaño de los paquetes de solicitud de establecimiento, confirmación y cierre de conexión, respectivamente, n_p el número de paquetes en que se divide el mensaje y H el tamaño de los bits suplementarios de cada paquete. (*Nota:* Es de señalar que $M = n_p \times D$, siendo D es el tamaño del campo de datos de cada paquete; asumiendo, claro está, que todos los paquetes tienen el mismo tamaño.)

- b) Como en el esquema datagrama, el uso que del ancho de banda se hace en circuitos virtuales es dinámico, no desaprovechándose cuando la fuente no transmite como sucede en el esquema de conmutación de circuitos, al ser no dedicado el canal. Podemos concluir por tanto que la eficiencia de uso de los recursos en la conmutación mediante circuitos virtuales es también del 100 %.

- *Complejidad de los nodos intermedios:*

- a) Como en datagramas, además de las capacidades de encaminamiento propias de la conmutación, los nodos requieren de la existencia de memorias internas de almacenamiento temporal a fin de dar cabida a los paquetes recibidos antes de proceder a su reenvío.
- b) Adicionalmente es necesaria la consideración y gestión de números de circuito virtual o tablas de asociación comunicación-linea de salida.

- *Latencia/retardo:*

Aunque este esquema también es poco recomendable para comunicaciones en tiempo real, sus prestaciones en este sentido son mejores que las presentadas por la técnica de datagramas dado que los tiempos de procesamiento intermedios se prevén inferiores. Al estar ya fijada la ruta a seguir por todos los paquetes, estos tiempos se emplean fundamentalmente en la comprobación de uso actual de la línea de salida requerida.

A parte de las tres características anteriores, hemos de mencionar la elevada robustez en la comunicación que presenta la técnica de conmutación de paquetes mediante datagramas frente a las otras dos. Esta técnica surgió motivada principalmente por necesidades estratégicas (militares) en cuanto a: (1) conveniencia de que la subred encuentre caminos origen-destino alternativos ante potenciales fallos en ciertos nodos o enlaces, y (2) deseo de que la transmisión sea inmune a escuchas por parte de terceros no autorizados. Ambos objetivos se consiguen gracias al hecho de que, como se ha discutido, cada paquete puede viajar por una ruta distinta. Este esquema de conmutación constituye en la actualidad la base de las transmisiones en Internet, como se estudiará en los Capítulos 8 y 9.

En la Tabla 6.1 se resume la comparación teórica de las técnicas de conmutación estudiadas con anterioridad.

Tabla 6.1. Resumen de las características de las técnicas de conmutación de circuitos y de paquetes mediante datagramas y mediante circuitos virtuales

	Circuitos	Paquetes mediante datagramas	Paquetes mediante circuitos virtuales
Ancho de banda	Uso estático: canal dedicado	Uso dinámico: no hay reserva de recursos a priori	Uso dinámico: circuito no dedicado
	No existen bits suplementarios en la transmisión una vez establecido el circuito	Existencia de bits suplementarios en cada paquete	Existencia de bits suplementarios en cada paquete (menos que en datagramas), además de los mensajes de establecimiento y cierre
Complejidad de los nodos	Conmutadores simples	Necesidad de memoria de almacenamiento	Necesidad de memoria de almacenamiento más gestión de números de circuito virtual
Latencia	Adecuada para aplicaciones interactivas y en tiempo real	Poco adecuada para aplicaciones interactivas y en tiempo real	Aunque mejor que datagrama, también es poco adecuada para aplicaciones interactivas y en tiempo real
Robustez	Si un nodo o enlace falla, la comunicación finaliza	Si un nodo o línea cae, se buscan rutas alternativas	Si un nodo o línea falla, la comunicación falla

Estimación de la latencia de las técnicas de conmutación:

A modo de ejemplo y aclaración, seguidamente se lleva a cabo el cálculo del tiempo total involucrado en la transmisión de un mensaje de datos para las técnicas de conmutación de circuitos (*CC*) y de paquetes mediante datagramas (*CPD*) y mediante circuitos virtuales (*CPCV*) considerando los siguientes parámetros:

- M : longitud en bits del mensaje a enviar.
- R_i : velocidad de transmisión de cada enlace o línea i , en bps.
- P : longitud total en bits de los paquetes, tanto en *CPD* como en *CPCV*.
- H_d : bits de cabecera de los paquetes en *CPD*.
- H_c : bits de cabecera de los paquetes en *CPCV*.
- C : longitud en bits de los mensajes intercambiados para el establecimiento y cierre de la conexión, tanto en *CC* como en *CPCV*.
- N : número de nodos intermedios entre las dos estaciones finales.
- $T_i^{CC/CPD/CPCV}$: tiempo de procesamiento en segundos en cada nodo intermedio i , en *CC*, en *CPD* y en *CPCV*.
- T_O, T_D : tiempo de procesamiento en las estaciones origen y destino, respectivamente.
- D_i : retardo de propagación en segundos asociado a cada uno de los enlaces.

Para este estudio vamos a analizar los distintos procesos involucrados en cada uno de los esquemas. En este sentido, conviene mencionar que, dado que existen N nodos intermedios, el número de saltos o enlaces total existente será $L = N + 1$ habida cuenta de que las estaciones estarán conectadas a sus nodos finales respectivos.

— *Comutación de circuitos (CC):*

- Establecimiento de conexión. El tiempo involucrado en cada uno de los pasos seguidos en este proceso es:

- Generación del mensaje de solicitud de establecimiento en la estación origen y en los nodos intermedios, es decir, en cada enlace:

$$\sum_{i=1}^L \frac{C}{R_i}$$

- Propagación en cada salto:

$$\sum_{i=1}^L D_i$$

- Procesamiento del mensaje de solicitud en cada nodo intermedio y en la estación de destino:

$$\sum_{i=1}^N T_i^{CC} + T_D$$

- Generación del mensaje de confirmación en la estación receptora:

$$\frac{C}{R_L}$$

- Propagación, sin procesamiento intermedio, del mensaje hasta la estación origen:

$$\sum_{i=1}^L D_i$$

- Tiempo de procesamiento de la confirmación en la estación origen: T_O .

En consecuencia, el tiempo total de establecimiento de conexión, T_e , será:

$$T_e = T_O + T_D + \frac{C}{R_L} + \sum_{i=1}^N T_i^{CC} + \sum_{i=1}^L \left(2 \cdot D_i + \frac{C}{R_i} \right) \text{ segundos}$$

- Transmisión de datos. Este proceso presenta solo dos componentes temporales dado que, una vez establecido el circuito, no se producen retardos de procesamiento en los nodos intermedios:

- Generación del mensaje en el emisor:

$$\frac{M}{R_1}$$

- Propagación del mensaje hasta el destino:

$$\sum_{i=1}^L D_i$$

Así pues, el tiempo de transmisión T_t resulta:

$$T_t = \frac{M}{R_1} + \sum_{i=1}^L D_i \text{ segundos}$$

- Cierre de conexión. Como en el caso de la transmisión de datos, dos serán las componentes temporales involucradas en el desarrollo de este proceso:
 - Generación del mensaje de solicitud de cierre, donde, sin pérdida de generalidad, se ha supuesto (Figura 6.2) que es el extremo de la derecha, receptor, quien la realiza:

$$\frac{C}{R_L}$$

- Propagación del mensaje hasta el otro extremo:

$$\sum_{i=1}^L D_i$$

Así pues, el tiempo de cierre Tc será:

$$Tc = \frac{C}{R_L} + \sum_{i=1}^L D_i \text{ segundos}$$

En conclusión, $T_{CC} = Te + Tt + Tc$ resulta

$$T_{CC} = T_O + T_D + 2 \cdot \frac{C}{R_L} + \frac{M}{R_1} + \sum_{i=1}^N T_i^{CC} + \sum_{i=1}^L \left(4 \cdot D_i + \frac{C}{R_i} \right) \text{ segundos} \quad (6.5)$$

- *Comunicación de paquetes mediante datagramas (CPD).* En este caso no existe establecimiento ni cierre de conexión. A pesar de ello vamos a suponer que todos los paquetes siguen la misma ruta, con lo que el estudio se simplifica y bastará con seguir la pista al último paquete¹ (véase Figura 6.3):

- Transmisión de datos:

- Generación del mensaje en la estación final origen. Dado que cada paquete tiene una longitud de $P - H_d$ bits de datos, el número de paquetes n_p a generar será

$$n_p = \frac{M}{(P - H_d)}$$

operación que, por simplicidad, suponemos resulta en un número entero. Como cada paquete tiene una longitud de P bits, el tiempo de generación total de los paquetes a transmitir será

$$P \cdot \frac{M}{[R_1 \cdot (P - H_d)]}$$

- Como hemos establecido anteriormente, sigámosle la pista al último paquete. Generado este en el origen, los procesos siguientes sobre el mismo serán:
 - Generación en los N nodos intermedios, esto es, en el resto de enlaces:

$$\sum_{i=2}^L \frac{P}{R_i}$$

¹ Este modo de proceder es válido si aceptamos que las velocidades de transmisión asociadas a los enlaces y los tiempos de procesamiento de los nodos son tales que a medida que se generan los paquetes en un nodo o estación, se retransmiten sin inducir retrasos adicionales en los paquetes siguientes en los saltos sucesivos.

- Procesamiento en cada nodo intermedio:

$$\sum_{i=1}^N T_i^{CPD}$$

- Propagación en cada salto:

$$\sum_{i=1}^L D_i$$

En consecuencia, $T_{CPD} = Tt$:

$$T_{CPD} = \frac{P \cdot M}{[R_1 \cdot (P - H_d)]} + \sum_{i=2}^L \frac{P}{R_i} + \sum_{i=1}^L D_i + \sum_{i=1}^N T_i^{CPD} \text{ segundos} \quad (6.6)$$

- *Commutación de paquetes mediante circuitos virtuales (CPCV):*

- Establecimiento de conexión. Los pasos seguidos en este proceso son:
 - a) Generación del mensaje de solicitud de establecimiento en la estación origen y en los nodos intermedios, es decir, en cada enlace:

$$\sum_{i=1}^L \frac{C}{R_i}$$

- b) Propagación en cada salto:*

$$\sum_{i=1}^L D_i$$

- c) Procesamiento del mensaje de solicitud en cada nodo intermedio y en la estación de destino:*

$$\sum_{i=1}^N T_i^{CPCV} + T_D$$

- d) Generación del mensaje de confirmación en la estación receptora y en los nodos intermedios:*

$$\sum_{i=1}^L \frac{C}{R_i}$$

- e) Propagación del mensaje hacia la estación origen:*

$$\sum_{i=1}^L D_i$$

- f) Procesamiento del mensaje de confirmación en cada nodo intermedio y en la estación de origen:*

$$\sum_{i=1}^N T_i^{CPCV} + T_O$$

c) Propagación del mensaje en cada salto:

$$\sum_{i=1}^L D_i$$

Así:

$$Tc = T_O + \sum_{i=1}^N T_i^{CPCV} + \sum_{i=1}^L \left(\frac{C}{R_i} + D_i \right) \text{ segundos}$$

En consecuencia,

$$T_{CPCV} = \frac{P \cdot M}{[R_1 \cdot (P - H_c)]} + 2 \cdot T_O + T_D + 4 \cdot \sum_{i=1}^N T_i^{CPCV} + \sum_{i=1}^L \left(3 \cdot \frac{C}{R_i} + 4 \cdot D_i \right) + \sum_{i=2}^L \frac{P}{R_i} \text{ segundos} \quad (6.7)$$

Para concluir y completar este análisis, hemos de hacer hincapié en la adopción de una serie de asunciones adoptadas en el estudio realizado. Algunas de las más relevantes son las siguientes:

- Se ha supuesto que todos los paquetes generados en CPD siguen la misma ruta.
- Se ha aceptado que todos los paquetes de datos generados tienen exactamente la misma longitud.
- Como se ha comentado anteriormente en una nota a pie de página, se ha supuesto que los paquetes se retransmiten sin provocar retrasos adicionales en los paquetes siguientes en los saltos sucesivos.
- También se ha obviado el tiempo invertido en la creación de los distintos paquetes por parte del emisor correspondiente, es decir, en la composición campo a campo de los mismos.

6.2.4. Comutación basada en etiquetas: MPLS

Tomando como base los esquemas de comutación anteriores, han surgido otros cuyo objetivo principal es aumentar la velocidad de transferencia de la información. Ejemplos de ello son las técnicas de *retransmisión de tramas* (FR, del inglés «Frame Relay») y de *retransmisión de celdas* (ATM, «Asynchronous Transfer Mode»), las cuales persiguen aumentar el rendimiento de las comunicaciones. FR es una forma simplificada de la técnica de comutación de paquetes que transmite una variedad de tramas de datos para adaptarse a las necesidades de las aplicaciones. Por su parte, ATM ofrece un servicio orientado a conexión mediante circuitos virtuales sustentado en la comutación de paquetes cortos de tamaño fijo denominados *celdas*.

La técnica de *comutación basada en etiquetas* o MPLS, del inglés «MultiProtocol Label Switching», está reemplazando a esquemas como FR y ATM por su mayor fiabilidad, flexibilidad y rendimiento para la transmisión de datos de alta velocidad. En MPLS a cada paquete se le asigna una etiqueta de manera que las retransmisiones no precisan la inspección completa del paquete, sino solo de la etiqueta asociada. Se permite crear así circuitos extremo-a-extremo para grupos de paquetes (en lugar de individuales) a través de cualquier red de forma independiente de la tecnología utilizada en la capa 2: FR, ATM, T1/E1, etc.

Como hemos indicado, a través de MPLS se asigna una etiqueta a cada paquete. En realidad se trata de una o más etiquetas en lo que viene a ser la cabecera MPLS, también denominada *pila de etiquetas*. Cada una de las etiquetas en la pila tiene 32 bits de longitud, organizados en cuatro campos como sigue (véase Figura 6.4):

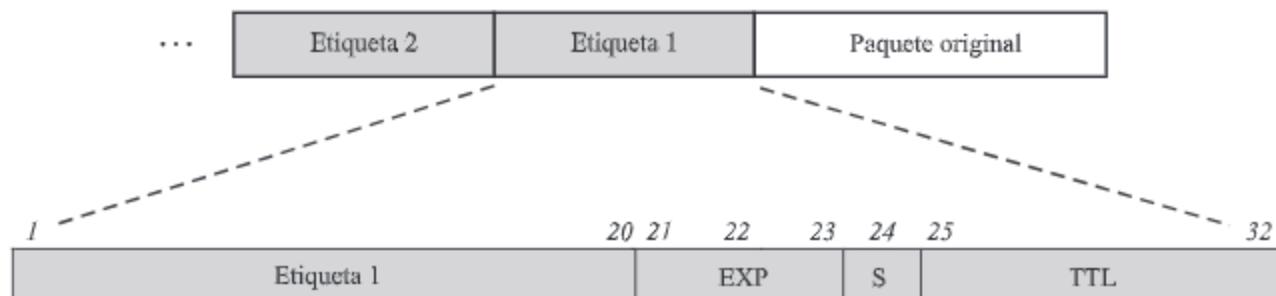


Figura 6.4. Cabecera o pila MPLS y formato de etiqueta.

- Un campo *etiqueta*, de 20 bits de longitud y significado local.
- Un campo *experimental* (EXP), de 3 bits de longitud, para especificar requisitos de calidad de servicio (QoS, «Quality of Service») y notificación explícita de congestión (ECN, «Explicit Congestion Notification») —véase Capítulo 7—.
- Un indicador *posición de pila* (S), de 1 bit, indicativo de si se trata (S = 1) o no (S = 0) de la etiqueta más antigua dentro de la pila del paquete.
- Un campo *tiempo de vida* (véase Capítulo 9), de 8 bits, para codificar el conteo de saltos del paquete.

La disposición de varias etiquetas sucesivas en la cabecera equivale al encapsulado múltiple de un paquete, pudiéndose por tanto agregar tráficos diferentes formando túneles. El funcionamiento conceptual básico de MPLS es así como sigue:

1. Los nodos que hacen conmutación basada en etiquetas se denominan LSR («Label Switch Routers»). Además, aquellos LSR que constituyen los puntos de entrada y salida de la red MPLS (esto es, los que insertan y eliminan las etiquetas en los paquetes) se denominan nodos LER («Label Edge Routers»).
2. Cada etiqueta hace referencia a una «marca» asociada a un conjunto de paquetes conocido como FEC («Forwarding Equivalence Class»), el cual queda definido por la ruta a seguir y por los requerimientos de QoS del flujo.
3. Las etiquetas se distribuyen entre los LSR y los LER mediante el protocolo de distribución de etiquetas o LDP («Label Distribution Protocol»). Se establecen así las denominadas *rutas de conmutación de etiquetas* o LSP («Label Switched Path»), las cuales se refieren, como hemos mencionado anteriormente, a túneles de transmisión a través de la red MPLS.
4. A partir de todo lo anterior, cada paquete entrante a la red MPLS es etiquetado con al menos una FEC dada.
5. En el LSR final de un túnel, la etiqueta más externa (esto es, la más nueva) se elimina de la pila o cabecera. De este modo, se desencapsula el paquete y se repite el procesamiento de reenvío, si es el caso, para la siguiente etiqueta en la pila.

6.3. Encaminamiento

Tarea fundamental en toda interconexión de red, la función de encaminamiento se refiere al conjunto de decisiones que deben tomarse a fin de establecer la ruta o rutas a seguir en una comunicación entre una estación final origen y una destino dadas. Como es evidente, una vez conocidas las rutas, los nodos

intermedios llevarán a cabo la retransmisión de los datos haciendo uso de alguna de las técnicas de conmutación previamente discutidas.

En lo que sigue se presentan distintas consideraciones al respecto de esta cuestión y se estudian diversos tipos de algoritmos de encaminamiento.

6.3.1. Fundamentos

Las características deseables que debe presentar todo algoritmo de encaminamiento son las siguientes:

- *Exactitud*. Esta característica se refiere a la necesidad (obvia) de que el encaminamiento sea tal que la comunicación se desarrolle «exactamente» entre las estaciones origen y destino deseadas, y no entre otras.
- *Robustez*. Como se planteó en la conmutación de paquetes mediante datagramas, resulta interesante la posibilidad de que la red sea capaz de encontrar rutas alternativas ante la potencial ocurrencia de fallos en ciertas líneas y/o nodos.
- *Estabilidad*. La característica anterior puede hacer que en la red aparezcan desplazamientos de carga que provoquen la aparición no deseada de bucles en la transmisión. Desde este punto de vista, hay que llegar a un compromiso entre robustez y estabilidad de funcionamiento.
- *Optimización*. Las rutas seleccionadas en cada momento deben ser las mejores de las posibles desde el punto de vista de un cierto criterio considerado en las tomas de decisión.
- *Imparcialidad*. La característica anterior puede conducir en ocasiones a situaciones claramente injustas en las que, por perseguir un interés global, se vean perjudicados intereses locales. Interesa por tanto llegar a un compromiso entre ambas características.
- *Eficiencia y simplicidad*. Adicionalmente a todas las características anteriores interesa que la implementación sea eficiente al tiempo que sencilla, de modo que los beneficios conseguidos por el empleo de la técnica en cuestión sean superiores al coste de su puesta en funcionamiento.

Los principales elementos o aspectos involucrados en la especificación de un algoritmo de encaminamiento dado, y a partir de los cuales se define este, son los siguientes:

1. El *criterio de decisión* hace referencia a la métrica usada en la decisión de encaminamiento, pudiendo ser de diverso tipo: distancia, número de saltos, retardo, eficiencia, etc.
2. El *instante de decisión* es el momento en que se toma la decisión de encaminamiento en los nodos: al inicio de la sesión en el caso de servicios orientados a conexión, como es el caso de la conmutación de circuitos y de conmutación de paquetes mediante circuitos virtuales, y para cada paquete en el caso de servicios no orientados a conexión, como es la técnica de conmutación de paquetes mediante datagramas.
3. El *lugar de decisión* se refiere al punto o puntos en que se toman las decisiones de encaminamiento. Así, podemos encontrar que estas pueden llevarse a cabo en un único lugar (por ejemplo, en el origen o en un nodo central), de manera distribuida y coordinada entre todos los nodos de la subred o de forma aislada en cada uno de ellos.
4. Relacionado con lo anterior, la *fuente de información* hace mención a la procedencia de los datos considerados en las tomas de decisión. Esta puede ser así, por ejemplo, local a cada nodo, procedente de los nodos adyacentes o de todos los nodos de la subred.
5. Por último, el *tiempo de actualización* es una cuestión de suma importancia puesto que establece la adaptación de las tomas de decisión a las condiciones cambiantes de la red. Podemos encontrar así algoritmos estáticos, en los que las decisiones son invariantes a lo largo del tiempo, o adaptables, caracterizados por realizar las decisiones de encaminamiento teniendo en cuenta las condiciones de la red en cada momento.

A partir de todos los aspectos mencionados podemos encontrar varias clasificaciones para los algoritmos de encaminamiento. En lo que sigue vamos a realizar un estudio en base a dos aspectos distintos: primeramente centraremos nuestra atención en el criterio de decisión y posteriormente en el lugar y la fuente de decisión. Sobre ambas cuestiones se presentarán y discutirán diferentes algoritmos.

Finalizaremos esta breve discusión introductoria diciendo que las rutas se especifican en forma de tabla, debiendo disponer cada nodo² de la suya propia. De este modo, a través de las tablas de encaminamiento cada nodo sabrá en un momento dado el nodo siguiente en el camino a seguir hacia un destino especificado. Para ello, cada una de las entradas en una tabla de encaminamiento tiene el siguiente formato básico: *<Destino, Siguiente nodo en la ruta al destino, Valor de métrica asociado a la ruta>*. Por ejemplo, la entrada *<5, 3, 3.4>* en la tabla de un nodo hipotético 2 significa que la mejor ruta del nodo 2 al nodo 5 es aquella que atraviesa el nodo 3, con un valor de métrica (retardo, distancia, ...) asociado igual a 3.4.

Por otra parte, las tablas pueden ser permanentes en el tiempo mientras no se decida establecer otras ante una eventualidad importante como, por ejemplo, cambios conocidos en la topología. También cabe la posibilidad de actualizarlas automáticamente para su adaptación dinámica a posibles variaciones en la red. Aunque un encaminamiento estático puede resultar útil en redes pequeñas y poco cambiantes, no es adecuado para redes de grandes dimensiones en las que las condiciones y la topología varían a lo largo del tiempo de forma impredecible. Por ello, lo más habitual es la disposición de algoritmos de encaminamiento adaptables, los cuales, no obstante, se diferencian operacionalmente de los estáticos solo en el hecho de que la estimación de las rutas se reitera de forma periódica en el tiempo en lugar de hacerlo una sola vez al inicio de la vida de la red.

6.3.2. Encaminamiento de mínimo coste

Al margen de la actualización o no de las tablas a lo largo del tiempo, todos los criterios ya mencionados (distancia, número de saltos, retardo, eficiencia, rendimiento) y algún otro más, como por ejemplo el coste económico, pueden ser utilizados para establecer si una ruta dada es mejor o no que otra. Es de destacar no obstante el amplio uso que se hace actualmente del número de saltos, al ser esta variable de fácil y rápido cálculo. Sea cual fuere el criterio concreto considerado, a partir de él se debe estimar el coste asociado a cada enlace o línea en la red. Así, el coste será proporcional a la métrica en caso de considerar esta igual al número de saltos o al retardo, e inversamente proporcional a ella en caso de considerar como medida, por ejemplo, la eficiencia o el rendimiento. Conocidos todos los costes asociados, la obtención subsiguiente de las rutas se suele realizar en base a un *algoritmo de mínimo coste*, estableciéndose en suma que la mejor ruta entre dos nodos dados es aquella que optimiza la métrica definida: menor distancia, menor número de saltos, menor retardo, mayor eficiencia, mayor rendimiento, etc.

Los dos esquemas más ampliamente utilizados para la elección de las rutas de mínimo coste son el de Dijkstra y el de Bellman-Ford. Ambos son iterativos y se especifican como sigue:

- *Dijkstra*: encuentra las rutas de mínimo coste entre un nodo origen dado y todos los demás desarrollando los caminos en orden creciente de coste. Para la especificación formal de este algoritmo definamos:

- N = número de nodos de la red.
- s = nodo origen.

² Es importante señalar que también los *hosts* o estaciones finales deben disponer de tablas de encaminamiento, si bien estas suelen ser más simples que las de los nodos intermedios de la red.

- L = lista de nodos incorporados por el algoritmo.
- C_{ij} = coste de la ruta entre los nodos i y j . Dicho coste será 0 si $i = j$ y mayor que este valor en otro caso. Si ambos nodos son inalcanzables directamente, se establece que $C_{ij} = \infty$.
- A_i = coste acumulado desde el nodo origen al nodo i , es decir, $A_i = C_{si}$.

El algoritmo de Dijkstra se formaliza en los siguientes tres pasos:

1. Inicialización:

$$\begin{aligned} L &= \{s\} && \rightarrow \text{la lista sólo consta del nodo origen} \\ A_i &= \infty, \forall i \neq s && \rightarrow \text{todos los nodos son inaccesibles desde } s, \text{ salvo} \\ A_s &= 0 && \rightarrow \text{él mismo} \end{aligned}$$

2. Obtención del nodo siguiente más cercano en el sentido del coste:

$$\begin{aligned} A_i &= \min_{j \notin L} A_j && \rightarrow \text{búsqueda del nodo vecino más próximo no incluido en } L \\ &&& \text{Añadimos } i \text{ a } L \rightarrow \text{incorporación del mismo a dicha lista} \\ &&& \text{Finalizamos si } L \text{ contiene los } N \text{ nodos} \end{aligned}$$

3. Actualización de los caminos de mínimo coste:

$$A_i \leftarrow \min [A_i, A_j + C_{ji}] \rightarrow \text{concatenación de caminos si el último término, } A_j + C_{ji}, \text{ es menor}$$

Ir al paso 2

— *Bellman-Ford*: encuentra las rutas de mínimo coste desde un nodo origen dado a distancia uno, a continuación los caminos de mínimo coste a distancia dos, y así sucesivamente. Para ello se basa en el siguiente principio: si el camino más corto de A a B pasa por C, los caminos AC y CB deben ser los más cortos.

De forma similar a como procedimos en el algoritmo de Dijkstra, definamos:

- s = nodo origen.
- C_{ij} = coste de la ruta entre los nodos i y j . Dicho coste será 0 si $i = j$ y mayor que este valor en otro caso. Si ambos nodos son inalcanzables directamente, se establece que $C_{ij} = \infty$.
- e = distancia de una ruta.
- $A_e(i)$ = coste acumulado desde el nodo origen s al nodo i con la condición de una distancia igual a e enlaces o saltos como máximo.

El algoritmo de Bellman-Ford, también llamado de Ford-Fulkerson, queda formalizado en los siguientes dos pasos:

1. Inicialización:

$$A_0(i) = \infty, \forall i \neq s \rightarrow \text{todos los nodos son inalcanzables desde el origen, a excepción de}$$

$$A_e(s) = 0, \forall e \rightarrow \text{él mismo, a distancia 0}$$

2. Actualización: de forma sucesiva $\forall e \geq 0$

$$A_{e+1}(i) = \min_j [A_e(j) + C_{ji}], \forall i \neq s \rightarrow \text{concatenación de caminos: } si = sj + ji$$

En la Figura 6.5 se muestra un ejemplo sencillo de construcción de tablas de encaminamiento. A partir de la topología dada en dicha figura, y conocidos los costes asociados a cada línea (aquí hemos

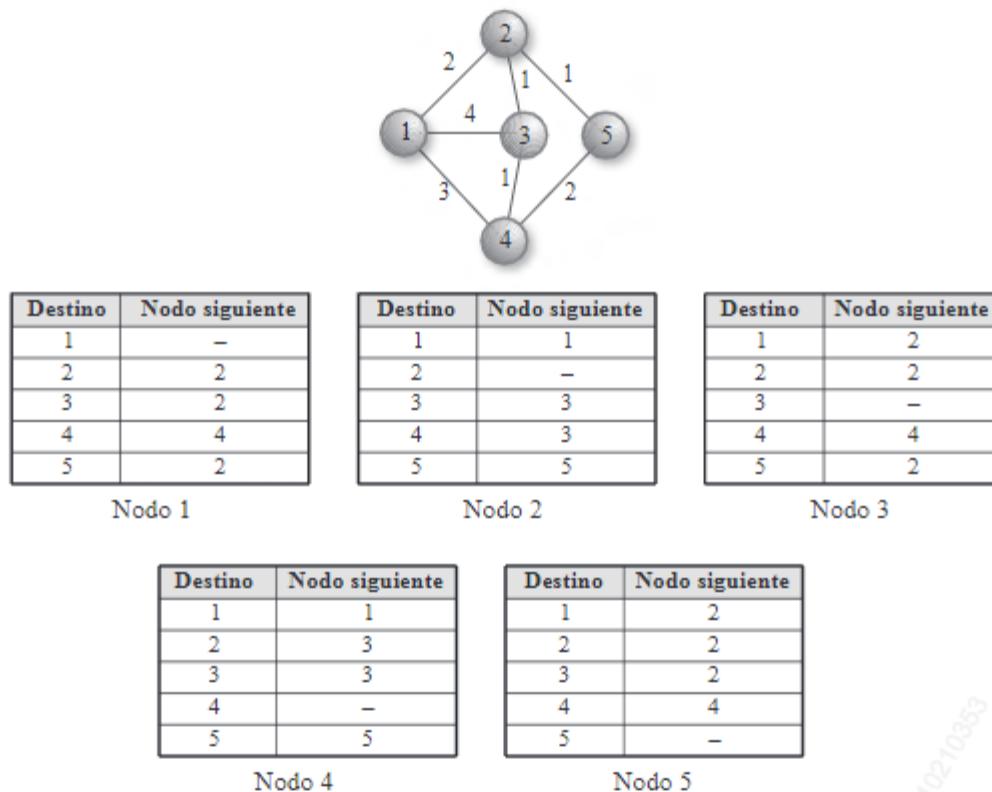


Figura 6.5. Ejemplo de topología de red, con costes por enlace, y tablas de encaminamiento de mínimo coste resultantes asociadas a los distintos nodos.

obviado si se trata de una métrica de retardo, rendimiento o cualquier otra), se calculan las tablas de encaminamiento para cada uno de los nodos de la subred de acuerdo al criterio de mínimo coste. Así, por ejemplo, según las tablas obtenidas para los nodos, la ruta para ir desde el nodo 1 al nodo 5 sería 1-2-5. Aunque los costes globales de las rutas no están especificados en la figura, es fácil comprobar que el asociado a la ruta antes citada es 3 (2 de la línea entre el nodo 1 y el 2, más 1 de la línea 2-5).

6.3.3. Algoritmos de encaminamiento según el lugar y la fuente de decisión

Aceptando el hecho previamente discutido de que los algoritmos de encaminamiento trabajan en base a un criterio de mínimo coste, a continuación estudiaremos diferentes tipos de esquemas desde el punto de vista del lugar de la red en el que se estiman las rutas para los diferentes destinos, así como de la información considerada para hacer ello posible.

Centralizados

Un algoritmo de encaminamiento se dice centralizado cuando el establecimiento de las tablas asociadas a los distintos nodos se realiza en un único punto de la red. Para ello es preciso que cada nodo envíe la información relativa al coste de las líneas asociadas hacia dicho nodo central. Este, una vez recopilada la información acerca de toda la subred, obtendrá las mejores rutas según el criterio de mínimo coste y enviará a cada nodo las respectivas tablas de encaminamiento a utilizar.

La principal ventaja de este esquema de encaminamiento radica en la optimización de las rutas debido al conocimiento global de toda la red. Sin embargo, presenta como gran desventaja la sobrecarga sufrida por el dispositivo central, además de la congestión de las líneas adyacentes al mismo, debido al elevado tráfico generado hacia y desde él. Esto provoca que el tiempo de respuesta de esta técnica sea generalmente alto, lo que la hace lenta en su potencial adaptación dinámica a situaciones cambiantes. Además, puesto que las tablas obtenidas se reciben antes en los nodos de la vecindad del punto central, es posible la aparición temporal de inconsistencias e inestabilidades en el encaminamiento de la red.

El ejemplo de la Figura 6.5 sirve como muestra de un algoritmo centralizado, ya que la estimación de las rutas parte del conocimiento y uso del estado global de la red. No obstante, seguidamente vamos a hacer mención a un algoritmo centralizado de cierta aceptación denominado *basado en flujo*. Lo que se hace en este caso es estimar las mejores rutas tomando como medida el retardo promedio de la subred, el cual se obtiene a partir del tráfico estimado para las distintas líneas o enlaces. Como ejemplo consideremos la topología y capacidades de línea (en kbps, *full-duplex*) dados en la Figura 6.6(a). Supongamos también las rutas establecidas para dicha topología; dichas rutas se indican en cursiva en cada celda de la matriz de encaminamiento de la Figura 6.6(b). En la misma matriz se indica el tráfico, en paquetes/segundo, transmitidos entre cada par de nodos origen-destino. Hemos de mencionar la consideración de tráfico simétrico; así, del nodo 2 al nodo 4 se transmiten los mismos paquetes/segundo que del 4 al 2: 3. A partir del tráfico global estimado para la subred, en la Figura 6.6(c) se

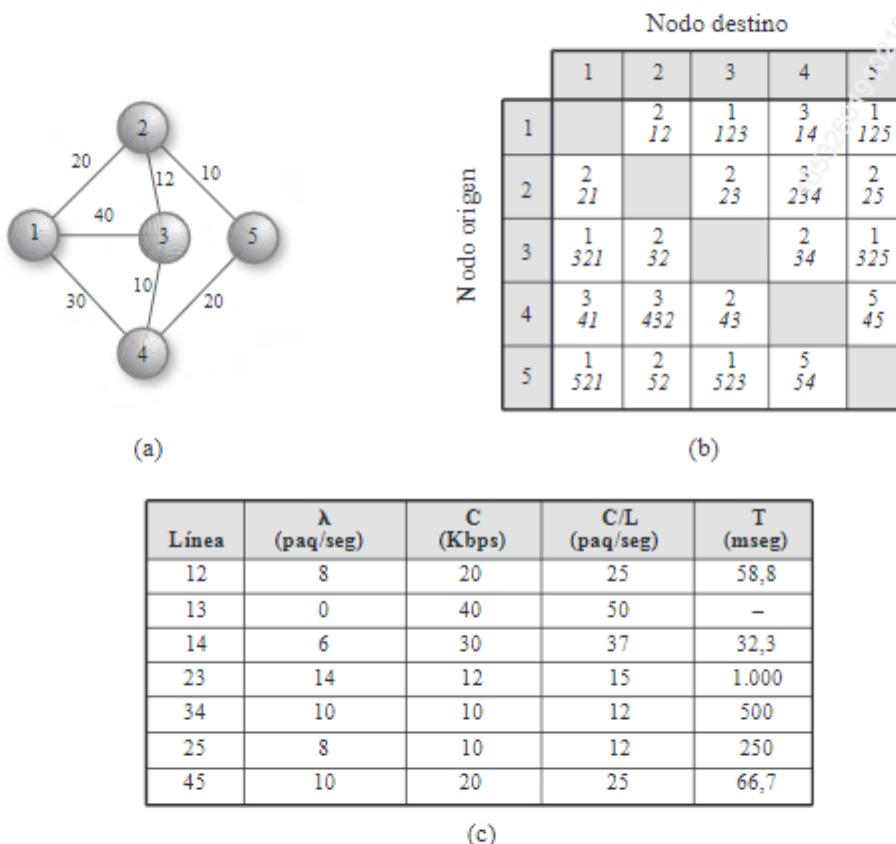


Figura 6.6. Encaminamiento basado en flujo: topología ejemplo y capacidades de línea *full-duplex*, en kbps, asociadas (a); matriz de encaminamiento y tráfico estimado entre cada par de nodos origen-destino (b); análisis de retardo medio sufrido en cada enlace suponiendo una longitud de paquete igual a 800 bits (c).

especifica el tráfico λ_i en paquetes/segundo para cada uno de los enlaces individuales; así, sobre la línea 1-2 se supone un total de 8 paquetes/segundo: 2 del nodo 1 al nodo 2, 1 del nodo 1 al nodo 3, 1 también del 1 al 5, y otros tantos en sentido contrario tal como se ha señalado en el carácter simétrico del tráfico: 2 del nodo 2 al 1, 1 del 3 al 1 y 1 también del nodo 5 al nodo 1.

A partir de esta información, y supuesta una longitud de paquete igual a 800 bits, se calcula el retardo medio T sufrido en la transmisión de un paquete para cada uno de los enlaces según la expresión (3.5). Finalmente, el retardo medio de la subred se obtiene como

$$T_{\text{subred}} = \frac{1}{\lambda} \cdot \sum_{i=1}^N \lambda_i \cdot T_i \quad (6.8)$$

donde λ_i es la carga en paquetes/segundo para cada uno de los N enlaces, T_i el retardo para cada uno de ellos, obtenido de (3.5), y λ la carga total de la red, obtenida esta como la suma de la correspondiente a cada uno de los N enlaces.

En el caso del ejemplo que nos ocupa resulta $T_{\text{subred}} = 398,8$ mseg. Para elegir entre dos o más matrices de encaminamiento (esto es, tablas de rutas) alternativas bastará con calcular para cada una de ellas, y de acuerdo al tráfico estimado sobre la red, el retardo medio que sufrirá un paquete, seleccionando finalmente aquel conjunto de rutas que proporcionen el menor retardo medio resultante.

Aislados

Frente a los esquemas centralizados, en los que existe un único nodo donde se estiman todas las tablas de encaminamiento para los nodos de la subred, en los aislados cada nodo es el responsable del cálculo de sus propias tablas. Para ello solo se tiene en cuenta información local, es decir, la actualización de las tablas de encaminamiento no implica trasvase alguno de información sobre la subred.

La métrica más usual utilizada en este tipo de algoritmos es el retardo, de modo que un nodo decide localmente la ruta a seguir en base a la medición periódica del tráfico asociado a cada una de sus líneas de salida. Así, como se muestra en la Figura 6.7, si un nodo dado recibe un paquete en un instante de tiempo, este será retransmitido sobre aquella línea de salida que implique un menor retardo, y ello independientemente del destino último del paquete.

Uno de los algoritmos de encaminamiento aislados más conocidos es el llamado de inundaciones («flooding» en inglés). El proceso seguido en él es simple: cada paquete recibido se retransmite sobre todas las líneas de salida salvo por la que se recibió, independientemente de su destino. Este

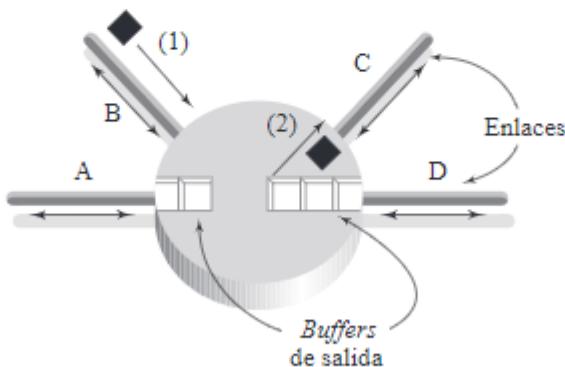


Figura 6.7. Ejemplo de nodo en el que se utiliza encaminamiento aislado con información local basada en retardo. Dado que la cola de salida más corta es la de la línea C, y supuesto que todas ellas tienen la misma velocidad, el paquete recibido (con fondo negro), (1), se reenviaría sobre ella, (2).

algoritmo presenta tres ventajas claras: (a) es de muy fácil implementación, (b) el paquete se recibirá con toda seguridad en el destino (siempre que exista al menos una ruta activa entre el origen y él) y (c) lo hará en el menor tiempo posible. No obstante, el principal inconveniente que plantea es el enorme tráfico que genera sobre la red. Supongamos, por ejemplo, la transmisión de un paquete desde el nodo 1 al nodo 5 en la topología dada en las Figuras 6.5 y 6.6. Como se observa en la Figura 6.8, es claro que en el primer salto se generarán 3 paquetes, uno sobre cada uno de los enlaces 1-2, 1-3 y 1-4. En el segundo salto se generarán 6 paquetes más: 2 por parte del nodo 2, uno hacia 5 y uno hacia 3; 2 por parte del nodo 3, uno hacia 2 y uno hacia 4; y 2 también por parte del nodo 4, uno hacia 3 y uno hacia 5. Siguiendo con este esquema, en un tercer salto se producirán 8 paquetes: 2 por parte del nodo 2, uno hacia 1 y uno hacia 5; 4 por parte del nodo 3, dos hacia 1, uno hacia 2 y uno hacia 4; y 2 por parte del nodo 4, uno hacia el nodo 1 y uno hacia el 5. En conclusión, la transmisión de un solo paquete desde el nodo 1 al nodo 5 producirá una carga total que irá aumentando a lo largo del tiempo y de forma tanto más creciente cuanto más compleja sea la topología considerada.

Aunque existen diversas soluciones posibles para resolver el problema de sobrecarga generado por el esquema de inundaciones, tales como limitar el número máximo de saltos al que puede propagarse un paquete o la identificación de paquetes para evitar retransmisiones duplicadas, quizás la más ampliamente adoptada se refiere al empleo de *árboles de expansión*, o ST («Spanning Tree» en inglés). En este caso se lleva a cabo la elaboración de una topología virtual completa sin bucles a partir de la consideración de un *nodo raíz*. En la Figura 6.9(a) se muestra el árbol de expansión basado en la minimización del número de saltos para la red de la Figura 6.8, tomando como nodo raíz el 5. Las líneas discontinuas reflejan rutas alternativas a nodos vecinos susceptibles de utilización en caso de necesidad. A partir de dicho árbol se obtienen las rutas de mínimo coste hacia el nodo raíz desde el resto de nodos; son las especificadas en la Figura 6.9(b). El envío basado en inundaciones sobre el árbol de expansión en lugar de sobre la topología de red real hace que el crecimiento de la carga sea lineal en lugar de exponencial como sucede ante la aparición de bucles.

A modo de conclusión de los algoritmos aislados, diremos que estos son más rápidos que los centralizados aunque es de esperar que las rutas obtenidas ahora sean sub-óptimas.

Distribuidos

Como en los aislados, los algoritmos de encaminamiento distribuidos se caracterizan por el hecho de que cada nodo es el responsable de calcular y, en su caso, actualizar sus propias tablas de encaminamiento. Sin embargo, a diferencia de los primeros, en los distribuidos los nodos intercambian información con sus vecinos para llevar a cabo el cálculo de las tablas, es decir, este no se realiza considerando solo información local como en el caso de los aislados.

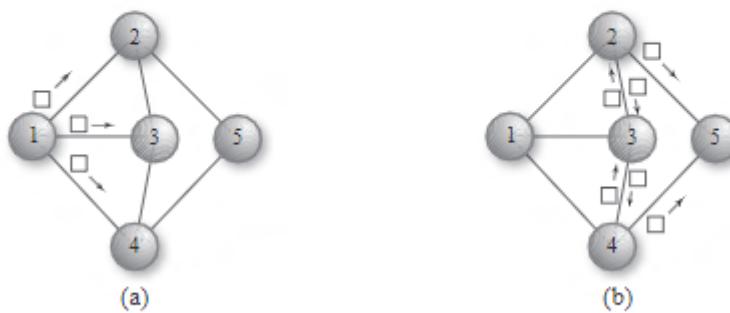
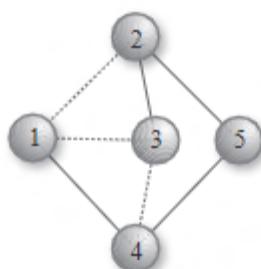


Figura 6.8. Ejemplo de transmisión de un paquete desde el nodo 1 al nodo 5 siguiendo el esquema de inundaciones: primer salto (a) y segundo salto (b).



(a)

Origen	Nodo siguiente
1	4
2	5
3	2
4	5

Destino: nodo 5

(b)

Figura 6.9. Árbol de expansión para la topología de la Figura 6.8 tomando como nodo raíz el 5 y métrica el número de saltos (a). Rutas de encaminamiento de mínimo coste desde el resto de nodos a dicho nodo destino (b).

Como ejemplo supongamos una métrica basada en retardo y las tablas de encaminamiento dadas en la Figura 6.10(a) correspondientes a los nodos 1, 2 y 4 de la red de la Figura 6.8 en un instante de tiempo dado. El proceso de actualización distribuido de la tabla de encaminamiento del nodo 3 sería el siguiente:

1. Intercambio de las tablas de encaminamiento con sus vecinos: 1, 2 y 4.
2. Estimación local del retardo sobre sus líneas de salida hacia los nodos vecinos (Figura 6.10(b)): líneas 3-1, 3-2 y 3-4.
3. Actualización de la ruta asociada a cada nodo destino como sigue:
 - Para cada nodo vecino, obtención del valor de retardo asociado a la ruta *nodo 3-nodo vecino-destino* como suma de dos componentes: (a) retardo del nodo vecino al destino

Destino	Retardo (mseg)
1	0
2	1
3	2
4	3
5	4

Nodo 1

Destino	Retardo (mseg)
1	3
2	0
3	2
4	4
5	1

Nodo 2

Destino	Retardo (mseg)
1	2
2	4
3	2
4	0
5	2

Nodo 4

(a)

Línea	Retardo (mseg)
31	2
32	4
34	2

Nodo 3

(b)

Destino	Nodo siguiente	Retardo estimado (mseg)
1	1	2
2	1	3
3	—	0
4	4	2
5	4	4

Nodo 3

(c)

Figura 6.10. Ejemplo de actualización distribuida de la tabla de encaminamiento del nodo 3 de la Figura 6.8: tablas de encaminamiento de los nodos vecinos, 1, 2 y 4, recibidas en un instante de tiempo dado (a); estimación local de los retardos de los enlaces a dichos nodos vecinos (b); actualización final en base a toda la información anterior (c).

según la tabla de encaminamiento correspondiente recibida en 1), y (b) retardo estimado en 2) para el enlace que une el nodo actual, 3, con el vecino considerado.

- Realizado el proceso anterior para los tres nodos 1, 2 y 4, se elegirá como nodo siguiente en la ruta hacia el destino aquel vecino a través del que resulte un valor de retardo total menor (Figura 6.10(c)).

El uso de información supra-local en este tipo de esquemas de encaminamiento hace que se consigan mejores rutas que con uno de tipo aislado. Por su parte, frente a uno centralizado, el distribuido es más rápido en el cálculo de las rutas y genera menos tráfico. No obstante estas ventajas, los algoritmos de encaminamiento distribuidos presentan un problema fundamental: la información manejada es quasi-local, por lo que: (1) las noticias tardan en propagarse por la red, de modo que hasta que un nodo se percate de una eventualidad ocurrida en una región alejada transcurrirá cierto tiempo, y, en consecuencia, (2) las rutas obtenidas podrán resultar temporalmente sub-óptimas, pudiendo producirse también inestabilidades en el encaminamiento.

Jerárquicos

Como consecuencia de la lenta propagación de noticias en la red a través del empleo de algoritmos de encaminamiento distribuidos, la estimación de las tablas resulta tanto más sub-óptima cuanto mayor sea el tamaño de la red. Para solucionar este hecho, al tiempo que se simplifican las tablas de encaminamiento y el tráfico intercambiado, en redes de grandes dimensiones como Internet se recurre al empleo de algoritmos de encaminamiento denominados *jerárquicos*, los cuales se caracterizan por ser distribuidos a dos niveles.

Dividida la red en regiones (Figura 6.11), en un primer nivel se procede a la actualización distribuida entre los nodos de cada una de las regiones de forma independiente entre ellas. En un segundo nivel se actualiza, también de forma distribuida, el encaminamiento entre regiones, para lo cual solo intervienen determinados nodos con acceso externo definidos al efecto (sombreados en Figura 6.11).

Por supuesto, es evidente el posible uso recursivo de este tipo de esquemas para dar lugar a jerarquías dispuestas en más de dos niveles.

6.3.4. Otros algoritmos de encaminamiento

Además de los esquemas de encaminamiento vistos anteriormente, existen otros que, si bien no podemos calificar como radicalmente distintos en su operación fundamental, sí merecen ser específicamente reseñados por diferenciarse en algún aspecto relevante de los hasta ahora considerados.

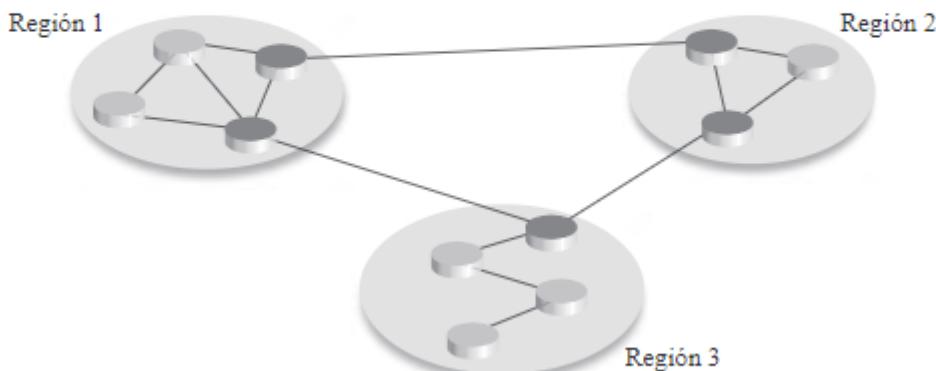


Figura 6.11. División de una red en regiones para la estimación jerárquica de las tablas de encaminamiento.

Encaminamiento desde el origen

Hasta aquí hemos aceptado que los nodos intermedios de la red intervienen activamente en la estimación de las rutas, y que son estos quienes llevan a cabo el encaminamiento de los paquetes recibidos. Una alternativa a este modo de proceder es el conocido como *encaminamiento desde el origen* («source routing» en inglés). En este la complejidad del proceso de encaminamiento no se encuentra en la subred, sino en las estaciones finales. Es decir, son estas y no los nodos intermedios quienes deben establecer la ruta a seguir por una transmisión dada. Para obtener la ruta correspondiente en caso de que sea desconocida por el nodo origen se utiliza el esquema de *descubrimiento de ruta*, cuyo procedimiento es el siguiente:

1. Una estación que desee conocer la ruta a un destino dado emite un mensaje de *solicitud de ruta*. Este se difunde mediante *inundaciones* sobre la red hasta alcanzar el destino.
2. En respuesta a la recepción de un mensaje de solicitud de ruta, la estación final destino genera un mensaje de *difusión de todas las rutas*. En él, cada nodo de la ruta almacenará su identidad a fin de que la estación final origen conozca la secuencia completa de nodos seguida desde el destino.
3. Recibida en el origen una respuesta por cada ruta posible, este podrá elegir la mejor de ellas y proceder a su almacenamiento en sus tablas de encaminamiento.
4. En la transmisión de datos posterior el origen especificará en cada paquete la secuencia de nodos a atravesar hasta alcanzar el destino, de modo que cada nodo de la ruta se limitará a reenviar el paquete hacia el siguiente nodo indicado tras él en la lista.

En el proceso de descubrimiento de ruta aparece un inconveniente ya apuntado en el uso del esquema de encaminamiento de inundaciones: la aparición de bucles en la topología de la red puede dar lugar a un incremento inaceptable de la carga de la red, además de inestabilidades en las transmisiones. Para evitar el problema se procede a la difusión de las tramas de descubrimiento según el esquema del árbol de expansión ya explicado.

Encaminamiento de aprendizaje hacia atrás

Para facilitar la administración de una subred, los nodos dispuestos en la misma suelen iniciarse con sus tablas de encaminamiento vacías. Entonces, ¿cómo se encaminan los datos recibidos? La respuesta es simple: basta con usar el esquema ya conocido de inundaciones, en el cual se lleva a cabo la retransmisión de la información sobre todos los enlaces de que dispone el nodo salvo por el que se recibió. Este esquema se suele complementar de nuevo con el uso de árboles de expansión.

A medida que transcurre el tiempo, las tablas se pueden ir completando y actualizando mediante el algoritmo denominado de *aprendizaje hacia atrás* (del inglés «backward learning»). Dicho algoritmo consiste en utilizar el campo de dirección origen de los paquetes recibidos para llenar las tablas según la siguiente filosofía: si un paquete de origen dado se recibe sobre un cierto enlace, dicho enlace será el establecido como ruta a seguir para alcanzar el origen. Consideremos, por ejemplo, la configuración dada en la Figura 6.12. Si el nodo N_2 recibe sobre la red Y un paquete originado por la estación A , anotará en sus tablas que los paquetes con destino a A deben retransmitirse sobre dicha red Y . Del mismo modo, si el nodo N_1 recibe un paquete originado, por ejemplo, por F sobre la red Y , N_1 anotará en sus tablas que la ruta hacia F pasa por la retransmisión también sobre la red Y . Es evidente que con este procedimiento las tablas de encaminamiento de los nodos tenderán a llenarse y estabilizarse automáticamente con el paso del tiempo.

En resumen, el proceso de encaminamiento seguido en *backward learning* es el siguiente:

1. Si el destino está en la misma red que el origen, el nodo intermedio descarta el paquete.

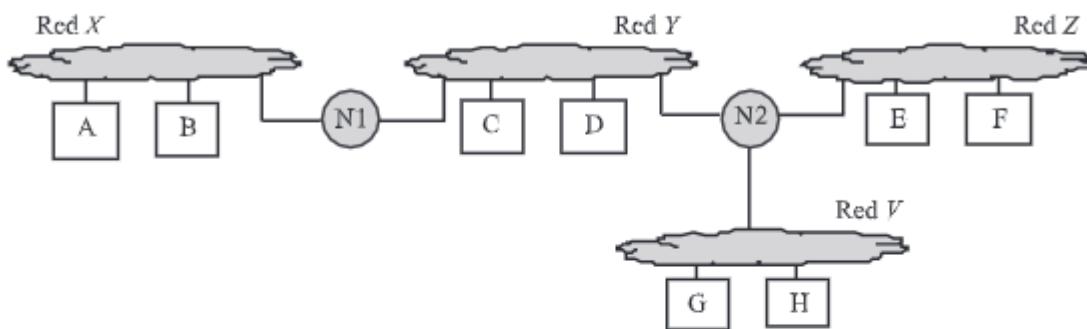


Figura 6.12. Configuración de interconexión con cuatro redes y dos nodos, N1 y N2.

2. Si las redes origen y destino son distintas, se retransmite el paquete según el siguiente esquema:
 - a) Si la ruta al destino está especificada en las tablas de encaminamiento, se procede a la retransmisión según se indica en ellas.
 - b) Si no lo está, se utiliza el esquema de inundaciones.
 - c) En cualquiera de los dos casos, se utiliza el algoritmo de aprendizaje hacia atrás a fin de actualizar las entradas en la tabla de encaminamiento, refrescando así aquellas rutas más viejas. Asimismo, las rutas que no hayan sido actualizadas o refrescadas durante un cierto periodo de tiempo se consideran obsoletas y se eliminan.

Por supuesto, el uso de este algoritmo de aprendizaje de rutas no excluye la utilización de otros esquemas de actualización ya vistos.

Encaminamiento multidestino y de difusión

Todos los esquemas de encaminamiento vistos con anterioridad se utilizan en comunicaciones *unicast* o de destino único. En transmisiones multidestino («multicast» en inglés), caracterizadas por ir dirigidas a varios destinos simultáneamente, no resulta adecuado el establecimiento de rutas uno-a-uno puesto que ello implicaría un uso excesivo, por innecesario, de recursos de la subred. Para evitar esto se establecen rutas comunes gracias a las cuales se reduce el tráfico preciso a generar (con la consecuente menor utilización de recursos) al tiempo que se maximiza el número de destinos deseados alcanzables. Un caso particular de un envío multidestino es el bien conocido de difusión («broadcast»), donde la información se dirige a todos los destinos existentes en la red.

Para transmisiones de esta envergadura puede pensarse una vez más en el empleo del esquema de inundaciones, complementado con la técnica del árbol de expansión. Sin embargo, utilizar un árbol de expansión implica, evidentemente, conocer este. Para ello puede procederse de forma manual (por parte del administrador) o automática. Diversos son los esquemas automáticos desarrollados que consiguen una buena aproximación al árbol de expansión óptimo, sin requerir para ello el conocimiento explícito de toda la estructura de la red. Uno de ellos es el algoritmo de *retransmisión por camino inverso* (RPF, «Reverse Path Forwarding»), el cual funciona en base al siguiente principio: «si un nodo recibe un paquete *multicast* procedente de X sobre la interfaz Y y esta está en la ruta de menor coste a X, el paquete se reenvía sobre todas las interfaces excepto sobre Y; en caso contrario se descarta el paquete». Una mejora a este esquema lo constituye el de *difusión por camino inverso* (RPB, «Reverse Path Broadcasting»): «si un nodo recibe un paquete *multicast* procedente de X, este se reenvía sobre todas las interfaces que se encuentren en los caminos de menor coste con destino a X». Esta técnica, a su vez, puede mejorarse mediante la variante *difusión por*

camino inverso truncada (TRPB, «Truncated Reverse Path Broadcasting»), en la que se asume que los nodos conocen la existencia de miembros en un grupo *multicast* (por ejemplo, a través del protocolo IGMP —ver Capítulo 9—).

Un esquema más avanzado es el de *multidestino por camino inverso* (RPM, «Reverse Path Multicast»). De modo similar a TRPB, RPM supone una adaptación dinámica del árbol puesto que cuando un nodo terminal detecta la modificación de la situación de miembros/grupos «debajo» suyo lo comunica al nodo «superior» para que este «pode» el camino de retransmisión correspondiente. El mecanismo es recursivo hacia arriba, existiendo también técnicas de «injerto» a fin de insertar ramas en el árbol ante la aparición de nuevos miembros en el grupo *multicast*. La Figura 6.13 muestra un ejemplo en el que un nodo elimina una rama del árbol ante la recepción de un mensaje de poda por parte de un nodo terminal.

6.4. Interconexión de redes

Todo lo anteriormente discutido sobre commutación y encaminamiento se refiere principalmente a redes de tipo WAN. Sin embargo, más allá de la existencia práctica de estas, la situación habitual con la que nos encontramos en el mundo real es la disposición de distintas redes (generalmente LAN, pero también de otros tipos) interconectadas entre sí a fin de globalizar los principios u objetivos perseguidos por las redes de computadores. Este conglomerado de redes distintas interconectadas es lo que se conoce formalmente como *interconexión de redes* («internetworking» en inglés), término de donde procede el nombre Internet de todos conocido.

Dado que, como ya se apuntó también en el Capítulo 1, una interconexión de redes es conceptualmente similar a una red WAN (salvo por el hecho de que entre dos nodos de la subred existe toda una red en lugar de una simple línea de transmisión), también los principios de commutación y encaminamiento son directamente aplicables a una interconexión de redes.

Si existe, sin embargo, un hecho diferencial no discutido hasta ahora en relación a una interconexión de redes: la posibilidad de que las redes accesibles a través de un nodo dado sean de naturaleza distinta. Por ejemplo, una Ethernet frente a una WLAN, o una LAN frente a WAN, etc. En tal caso, ¿cuáles son las funciones extra que debe realizar el nodo intermedio para posibilitar dicha comunicación heterogénea? En lo que sigue se analiza este particular, no sin antes discutir algunos aspectos básicos previos no tratados aún acerca de las interconexiones de redes.

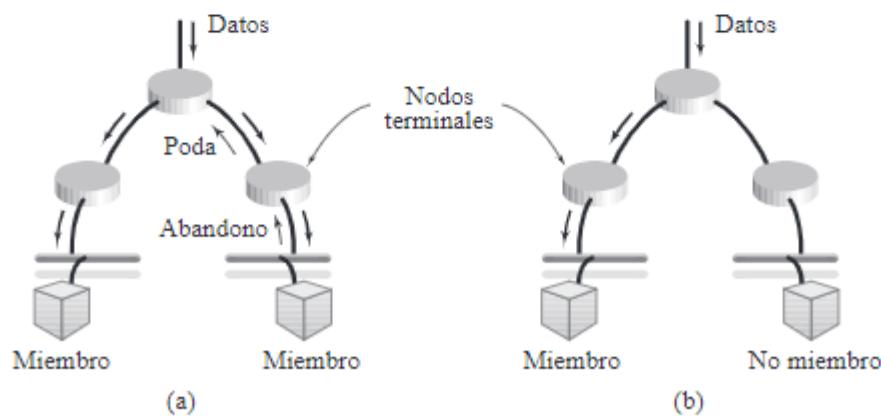


Figura 6.13. Modificación del árbol *multicast* en RPM ante el abandono del grupo por parte de uno de los miembros del mismo.

6.4.1. Principios de interconexión

Dos redes se interconectan a través de un dispositivo con la funcionalidad precisa para poder salvar las diferencias entre ellas. El tipo del dispositivo de interconexión en cuestión dependerá de la capa superior en la que opera, esto es, la capa más alta en la que se diferencian las redes interconectadas:

- *Repetidor* («repeater»): trabaja en la capa física y es un mero amplificador de señal que solventa el problema de atenuación de la señal en su propagación hacia el destino.
 - *Puente* («bridge»): opera hasta la capa de enlace a fin de posibilitar la interconexión de redes LAN de naturaleza diferente, por ejemplo una red WLAN con una Ethernet.
 - *Pasarela* («gateway»): funciona hasta la capa de red y permite la interconexión de sistemas cuya funcionalidad difiere a nivel 3.
 - *Dispositivo de capa superior*: trabaja por encima de la capa de red para posibilitar la comunicación entre protocolos de la capa correspondiente.

Aunque todos los dispositivos mencionados existen en la realidad y, como tales, pueden ser utilizados según el caso particular, en lo que sigue centraremos nuestra atención exclusivamente en el estudio de pasarelas. Por una parte, porque este tipo de dispositivos incluye implícitamente la funcionalidad de repetidores y puentes. Por otra, porque la existencia de dispositivos de capa superior a la de red resulta más conceptual que real dada la tendencia actual hacia la integración global IP, esto es, Internet o arquitectura TCP/IP.

Es de mencionar en este punto que las pasarelas suelen recibir en la práctica el nombre de *routers*, si bien es cierto que la funcionalidad real de ambos tipos de dispositivos es sustancialmente diferente. Así, un *router* no es más que un dispositivo con capacidad de encaminamiento, mientras que una pasarela, como ya se ha mencionado, permite la interconexión de redes diferentes a nivel de capa 3. No obstante este matiz, y de acuerdo con el uso práctico del término, en adelante hablaremos de *router* para referirnos genéricamente a todo dispositivo de encaminamiento que permite interconectar redes distintas.

En lo que sigue se discuten las principales características funcionales y de operación de los routers como dispositivos universales de interconexión.

6.4.2. Routers e interconexión de redes

El modelo de red para la interconexión de dos sistemas a través de un *router* es el que se muestra en la Figura 6.14. En esta se indica, por un lado, las capas de dos *hosts* situados en sendas redes, red 1 y red 2 y, por otro, que el *router* opera hasta la capa de red. Esta estructura es generalizable a la interconexión

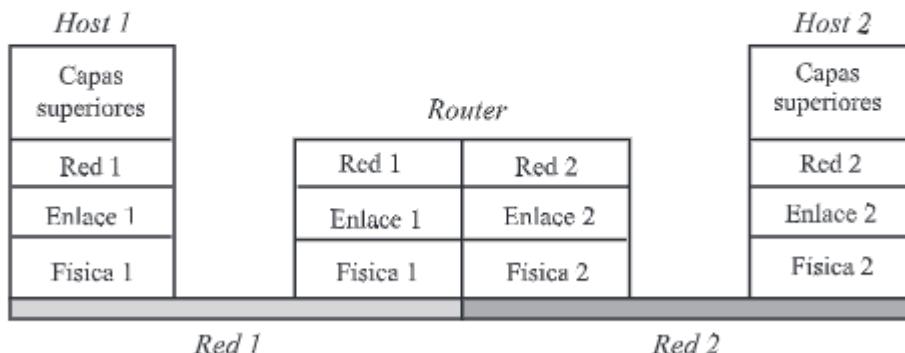


Figura 6.14. Modelo de red en una interconexión a través de un router.

de tantas redes como se desee, requiriéndose solo para ello la disposición en el *router* de una interfaz de acceso para cada una de las redes en cuestión.

Resulta obvio que el dispositivo, como elemento propio de todos y cada uno de los entornos que interconecta, debe «hablar» hasta la capa de red de cada uno de ellos y llevar a cabo la correspondiente «adaptación» a fin de posibilitar la comunicación bidireccional entre ellos. Dado que el *router* solo trabaja hasta la capa 3, el resto de capas superiores deben implementar idénticos protocolos si deseamos poder comunicar con éxito dos usuarios finales situados en los *hosts* 1 y 2.

Las funciones específicas a realizar por un *router* van a depender del tipo concreto de redes que interconecte, apuntándose conceptualmente capa a capa en lo que sigue dichas funciones dadas dos redes cualesquiera.

Capa física

Como es evidente, si deseamos interconectar, digamos, un medio cableado (p.e., Ethernet) con uno inalámbrico (p.e., IEEE 802.11), el dispositivo de interconexión debe recibir la información en un formato y retransmitirla en el otro. Para ello se requiere disponer de las interfaces físicas precisas en ambos medios (tarjeta RJ45, antena, etc.) además de la lógica necesaria para la representación adecuada de las señales en el canal (tipo de código binario, esquema de modulación, etc.). En suma, el dispositivo desde estar dotado de la funcionalidad propia de cada red.

A pesar de trabajar también a nivel físico, los dispositivos repetidores ya mencionados con anterioridad se diferencian de un *router* en que los primeros se limitan a recoger las señales recibidas y retransmitirlas (generalmente amplificadas en potencia) al segundo medio. En cambio, un *router* decodificará, por un lado, la señal recibida para pasarl a la capa de enlace y, por otro, en la segunda interfaz llevará a cabo el envío físico de los bits recibidos de la capa de enlace correspondiente. Y ello conforme a la comunicación real vertical ya discutida para las comunicaciones en el Capítulo 1.

Capa de enlace

A este nivel el *router* recibirá la secuencia de bits detectados en la capa física y procederá a interpretar la trama correspondiente en la capa de enlace, llevando a cabo la funcionalidad correspondiente: delimitación y, en su caso, dependiendo del protocolo considerado, control de errores y/o control de flujo. Extraída la información, esta se pasará a la capa de red. Por su parte, en la otra interfaz procesará los datos provenientes de la capa de red y los enviará sobre la física.

Es evidente de nuevo que el dispositivo de interconexión debe conocer los protocolos de enlace de las redes que interconecta. Esta capa, como es ya conocido, hace referencia en el caso de las redes LAN a las subcapas de enlace y MAC. Es decir, el dispositivo de interconexión no solo llevará a cabo un cambio en el formato o encapsulado de la información, sino también debe adaptarse al mecanismo concreto de acceso al canal utilizado: CSMA, protocolos libres de colisión, etc.

Además de los simples cambios en el formato/encapsulado, la funcionalidad de un *router* a nivel de enlace tiene en ocasiones implicaciones de gran alcance, y como tales merecen ser comentadas aunque sea brevemente:

- Algunos sistemas consideran prioridades en los envíos. En este caso, la prioridad de las tramas desde la red sin ella hacia la otra deberá ser introducida ficticiamente por el *router*.
- Asimismo, el *router* deberá mentir en ocasiones al emisor en cuanto a la confirmación de la recepción de las tramas en el destino, a fin de evitar la expiración de los temporizadores de reenvío en el origen.

En el caso de los puentes, dado que llevan a cabo la comunicación «directa» entre las capas de enlace, otras consideraciones de interés que deben hacerse son:

- Si deseamos enviar datos desde una red con MTU dada a otra con MTU menor, el puente deberá fragmentar y re-generar las tramas a transmitir sobre el segundo medio.
- De modo similar, el envío desde una red con mayor velocidad de transferencia a otra de menos velocidad deberá ser adaptado por el puente en base al empleo de memorias de almacenamiento temporal para evitar posibles pérdidas.

En suma, las implicaciones de una interconexión a nivel de enlace van en ocasiones conceptualmente bastante más allá del mero cambio de formato y reenvío de los datos.

Capa de red

Como no puede ser de otro modo, y en la misma línea apuntada para las otras capas, la funcionalidad de un *router* a nivel de la capa de red se refiere a la adecuación de las diferencias existentes a este nivel para las redes interconectadas. Aunque a este respecto pueden ser varias las disimilitudes entre ellas (p.e., tipo de direccionamiento), dos son los aspectos que aquí trataremos en relación con la funcionalidad principal descrita a lo largo del presente capítulo:

1. *Tipo de interconexión*: mediante circuitos virtuales concatenados o mediante datagramas.

Como ya conocemos, en el primer caso se establece una conexión no dedicada entre las estaciones finales origen y destino con carácter previo a la transmisión de los datos. Esta ruta es la que seguirán todos los paquetes emitidos por el origen, por lo que se recibirán ordenadamente en el destino.

Frente a ella, la interconexión basada en datagramas ofrece un servicio no orientado a conexión en el que cada paquete se transmite de forma independiente del resto, pudiendo recibirse, en consecuencia, de forma desordenada en el destino.

2. *Fragmentación y ensamblado*: transparente y no transparente.

Dado que, por lo general, las redes de una interconexión tienen MTU distintas, una función habitual a realizar por parte de los dispositivos de interconexión es la fragmentación a nivel 3 de los paquetes intercambiados entre dos redes. Pensemos la siguiente situación: un dispositivo de interconexión fragmenta los paquetes que van desde una red 1 a una red 2 de MTU inferior; atravesada dicha red 2, ¿tendría sentido que el dispositivo de interconexión siguiente en la ruta ensamblara, es decir, agrupara, los fragmentos ante el hipotético reenvío de estos sobre una red 3 de MTU superior a la de la red 2?

Si se procediese así, de modo que los paquetes llegasen al destino perfectamente ensamblados (tal como fueron enviados), estaríamos ante un esquema de *fragmentación transparente*. Es decir, en este caso es la propia subred la encargada de fragmentar y ensamblar los paquetes según la necesidad en cada momento y de forma «transparente» para el usuario final (Figura 6.15(a)).

No obstante esta posibilidad, lo cierto es que no tiene demasiado sentido actuar de esta manera. Esto es así por dos motivos:

- a) Si la transmisión se realiza en base a datagramas es posible que no todos los fragmentos sigan la misma ruta y, en consecuencia, será imposible realizar su posterior ensamblado.
- b) Aun en el caso de que todos ellos siguiesen la misma ruta, su ensamblado en un dispositivo de interconexión intermedio puede resultar una completa pérdida de tiempo ante una posible fragmentación necesaria posterior.

Así pues, la forma más usual de proceder a este respecto consiste en dejar que el ensamblado sea una función propia de las estaciones finales, liberando a la subred de tan tediosa (y generalmente estéril) tarea. Es lo que se conoce como *fragmentación no transparente* (Figura 6.15(b)).

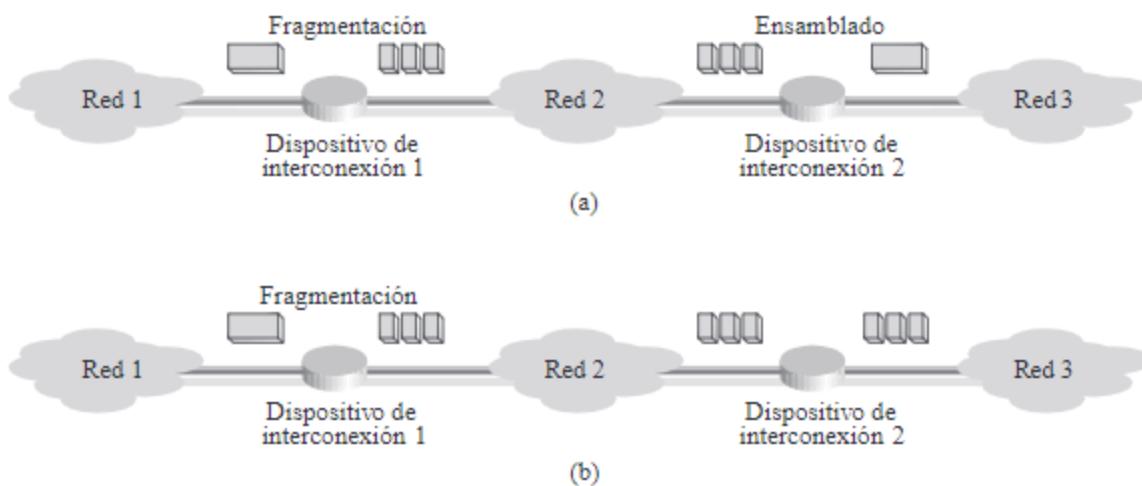


Figura 6.15. Fragmentación transparente (a) frente a no transparente (b).

Al margen de las dos cuestiones anteriores, es también de reseñar la necesaria adopción de funciones adicionales tales como el almacenamiento temporal de la información para adaptar la intercomunicación de redes de distintas velocidades y requerimientos.

6.4.3. VLAN y LAN conmutadas

De un modo u otro relacionado con la problemática a lo largo del tema tratada, en esta sección vamos a abordar el estudio breve de redes VLAN y LAN conmutadas. Ambas constituyen una alternativa a la interconexión directa a través de *routers* de redes físicas LAN, y resultan de gran interés para el despliegue efectivo de redes de usuario finales.

En primer lugar hemos de partir de la base de que suele ser habitual la necesidad de disponer varias redes LAN para la conformación de una red corporativa perteneciente a una empresa o institución, donde puedan existir diversos departamentos claramente separados en funcionalidad, acceso a recursos y/o permisos. Para ello puede hacerse uso de dispositivos de tipo *switch* (conmutador) interconectados entre sí en lo que conoce como *LAN conmutadas*. Los principios de funcionamiento de este tipo de entornos están sujetos a los ya descritos en el apartado anterior.

Una alternativa a la disposición de redes físicas separadas es la creación de *dominios de difusión* lógicamente diferenciados dentro de una misma red física. Es lo que se conoce como *red LAN virtual* o VLAN («Virtual LAN»). Las VLAN pueden clasificarse en cinco tipos:

- *VLAN de nivel 1*. También conocidas como «port switching», los miembros de la LAN se diferencian por el puerto al que están conectados al conmutador o medio físico que constituye la red.
- *VLAN de nivel 2 por dirección MAC*. En este caso se asignan los *hosts* a una VLAN en función de la dirección MAC de aquellos.
- *VLAN de nivel 2 por tipo de protocolo*. La VLAN viene determinada por el contenido del campo tipo de protocolo de la trama MAC. Por ejemplo, VLAN 1 al protocolo IP, VLAN 2 a IPX, etc.
- *VLAN de nivel 3 por direcciones de subred*. En este caso son los paquetes y no las estaciones finales quienes pertenecen a una VLAN u otra en función de su cabecera de nivel 3.
- *VLAN de niveles superiores*. La diferenciación entre varias VLAN puede hacerse en base a criterios relacionados con capas superiores, como, por ejemplo, un puerto TCP, una dirección de correo electrónico, etc.

El funcionamiento y configuración de las VLAN se sustenta en el empleo de protocolos como STP («Spanning Tree Protocol»), VTP («VLAN Trunking Protocol») o IEEE 802.1Q. El más común de ellos es el etiquetado IEEE 802.1Q, el cual considera un formato de trama similar a Ethernet al que se añaden dos octetos para codificar la prioridad.

RESUMEN

Centrado el tema precedente en el estudio de tecnologías y redes LAN, a lo largo de este sexto capítulo hemos realizado un primer acercamiento a las funciones y servicios de la capa de red según OSI.

Comenzó el tema analizando una cuestión fundamental en este tipo de redes, la conmutación. Se han presentado así las técnicas de conmutación de circuitos y de paquetes mediante datagramas y mediante circuitos virtuales. Para todas ellas se han introducido los respectivos principios de funcionamiento así como sus principales características, llevándose a cabo su comparación en base a parámetros tales como ancho de banda, complejidad de los nodos intermedios e idoneidad para su uso en aplicaciones interactivas y de tiempo real. Aunque de un modo somero, también se ha hecho mención a otros esquemas de conmutación más avanzados sustentados en los anteriores: FR, ATM y, en especial, MPLS, tecnología dominante en la actualidad en redes WAN.

Abordando otra de las funciones propias a implementar a nivel 3, seguidamente se ha procedido al estudio del encaminamiento como uno de los aspectos más importantes a considerar en una red de comunicaciones. Comentados los elementos más característicos de un algoritmo de encaminamiento, se han presentado las bases de los algoritmos de mínimo coste y una clasificación de los algoritmos a partir del lugar y la fuente de decisión. Además de estos esquemas, de tipo *unicast*, también se han discutido aspectos clave de las técnicas de encaminamiento para transmisiones multidestino y se han presentado otros esquemas de encaminamiento como el del origen y el de aprendizaje hacia atrás.

Para concluir el tema, una cuestión relevante también desarrollada ha sido la relativa a la interconexión de redes. En este marco se ha discutido la existencia de distintos tipos de dispositivos en función de la capa en que operan, centrando nuestra atención en el estudio de *routers*. También se ha hecho mención al despliegue de redes LAN conmutadas y LAN virtuales como alternativa a tener en consideración frente al uso de redes físicas diferenciadas.

EJERCICIOS

1. Justifique si es posible que, a pesar de que todos los elementos de una red funcionen correctamente, un mensaje dado se reciba en un destino inadecuado.
2. Explique el punto débil del siguiente razonamiento: «La conmutación de paquetes requiere que a cada paquete se le añadan bits de control y de dirección, lo que provoca un coste adicional en esta técnica. En conmutación de circuitos se establece un circuito transparente, no siendo necesario el uso de bits suplementarios. Por tanto, dado que no existe coste adicional en la técnica de conmutación de circuitos, la utilización de la línea es más eficiente que en conmutación de paquetes.»
3. Suponga la transmisión de un mensaje de 64 kB de longitud entre dos estaciones origen y destino separadas entre sí tres saltos, todos ellos de 10 Mbps de capacidad y 800 km de longitud. Aceptando que el tiempo de procesamiento en cada nodo es despreciable y que no existen errores en la comunicación:
 - a) ¿Qué tiempo conlleva la transmisión del mensaje mediante datagramas si el tamaño de los paquetes es 2 kB, con una cabecera de 30 bytes?
 - b) ¿Y mediante circuitos virtuales si los mensajes de control son de 62 octetos?

4. Compare desde el punto de vista de ancho de banda y latencia la técnica de conmutación de mensajes apuntada en el capítulo con la de datagramas.
5. Se desea transmitir un mensaje de M bits entre dos estaciones origen y destino separadas entre sí S enlaces, sobre una red de conmutación de paquetes mediante datagramas. D_i es el retardo de propagación en cada línea i (en m/s); el tiempo de procesamiento en cada nodo es T_i y P es la longitud total de cada paquete (en bits), con H bits de cabecera y L de datos. Calcule el tiempo total involucrado en la transmisión del mensaje M si se supone que la velocidad de cada enlace i (expresada en bps) es $V_1 < V_2 < \dots < V_{S-1} < V_S$. Señale y justifique las diferencias aparecidas en relación con la expresión obtenida para el ejemplo desarrollado en el capítulo para esta técnica de conmutación.
6. Repita el ejercicio anterior en la siguiente situación: $V_1 < V_2 < \dots < V_{K-1} < V_K > V_{K+1} > \dots > V_{S-1} > V_S$.
7. Suponga una red de conmutación de paquetes sobre la que se lleva a cabo un envío origen-destino basado en datagramas, para la que se especifican los siguientes parámetros:

M : tamaño en bits del mensaje a transmitir,

L : longitud en bits de todos y cada uno de los paquetes a considerar,

H : tamaño en bits de la cabecera de los paquetes,

S : número de saltos intermedios en la ruta origen-destino,

R_i : velocidad de transmisión en bps del enlace i -ésimo,

D_i : longitud en m del enlace i -ésimo,

V_i : velocidad de propagación en m/s correspondiente al enlace i -ésimo,

P_i : tiempo de procesamiento en s de un paquete en cada nodo.

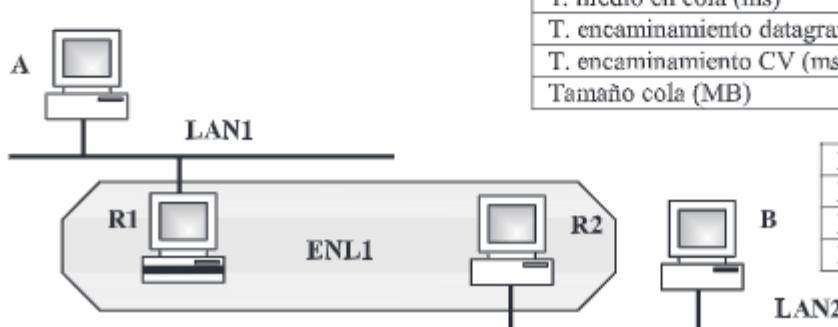
- a) Esquematice gráficamente el proceso involucrado en el envío completo del mensaje M si aceptamos que el tiempo implicado se puede aproximar a través de la siguiente expresión:

$$T_{\text{total}} = \sum_{i=1}^S \left[\frac{M}{L - H} \cdot \left(\frac{L}{R_i} + P_i \right) + \frac{D_i}{V_i} \right]$$

Explique el significado de cada uno de los sumandos en la ecuación.

- b) ¿En qué situaciones tendría sentido utilizar este esquema de envío frente al de datagramas tradicional?

8. Dos entidades paritarias de nivel de red en A y B intercambian paquetes de 1.504 bytes a través de una interconexión (véase la figura inferior). Los paquetes deben atravesar 2 nodos (R1 y R2)



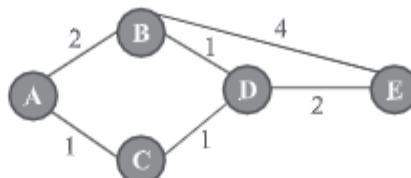
Router	R1	R2
T. medio en cola (ms)	256	512
T. encaminamiento datagramas (ms)	64	64
T. encaminamiento CV (ms)	48	48
Tamaño cola (MB)	1.024	2.048

Elemento	V_{transm}
LAN1	10 Mbps
LAN2	4 Mbps
ENL1	256 kbps

para llegar desde A a B. Los parámetros más relevantes de las redes, enlaces y *routers* se indican en la tabla adjunta. Determine el tamaño mínimo de un archivo para que resulte más rápida su transmisión mediante circuitos virtuales (CV) que mediante comutación de paquetes (CP), sabiendo que el tamaño de la cabecera en CV es de 32 bytes, mientras que en CP es de 48 bytes.

NOTA: Suponga despreciables los tiempos de acceso al medio y de propagación en LAN1 y LAN2 y que las tramas de confirmación y establecimiento tienen un tamaño despreciable.

9. Un mensaje de m bits se transmite por una ruta de L saltos en una red de paquetes como una serie de N paquetes consecutivos, cada uno de ellos con k bits de datos y h bits de cabecera. Suponga que $m \gg k + h$, que la velocidad de los enlaces es R bps y que los retardos de propagación y de cola son despreciables.
 - a) ¿Cuál será el número total de bits transmitidos?
 - b) ¿Cuál es el retardo total experimentado por el mensaje (es decir, el tiempo entre el primer bit transmitido por el emisor y el último recibido por el receptor)?
 - c) ¿Qué valor de k minimiza el retardo total?
10. Un servicio Internet actualmente en auge es el de «voz sobre IP» (VoIP, del inglés «Voice over IP»); en suma, la transmisión de voz sobre redes de comutación de paquetes mediante datagramas. ¿Cuáles son los principales retos a resolver en este tipo de servicios de cara a su adopción generalizada por parte de los consumidores, frente al tradicional de telefonía basada en comutación de circuitos?
11. Puede demostrarse que el algoritmo de encaminamiento de *inundaciones* puede utilizarse para determinar la ruta con menor número de saltos. ¿Se puede usar también para la obtención del camino con menor retardo? Razone la respuesta.
12. Considere la red mostrada en la figura adjunta, en la que se representan 5 nodos unidos con enlaces y para cada uno de ellos el retardo, en ms , sufrido por los paquetes al atravesarlo. Si se considera un protocolo de encaminamiento dinámico de tipo distribuido que toma como métrica el retardo:
 - a) Muestre la evolución de las tablas de encaminamiento hasta su estabilización, considerando un periodo de intercambio de las mismas de 10 segundos, comenzando este en $t = 0$ s. ¿En qué instante de tiempo se consigue la estabilidad?
 - b) Si suponemos que en $t = 45$ s cae el enlace DE, ¿en qué instante temporal volverán a ser estables las rutas?

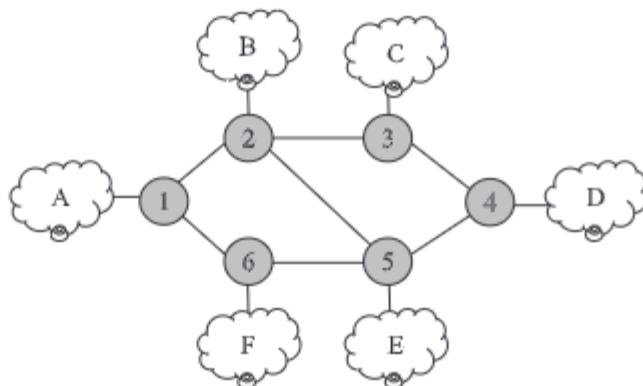


13. Se dispone de la topología de red inferior adjunta, donde se implementa un algoritmo distribuido de actualización de tablas de *routing* basado en el número de saltos. Supuesto que la actualización comienza para todos los nodos en $t = 15$ s con una periodicidad de 30 s y que las tablas de cada nodo están inicialmente vacías:
 - a) Indique cuáles serán las tablas de encaminamiento estables finales para cada uno de los nodos. ¿En qué instante de tiempo se alcanza esta situación?
 - b) Suponga ahora que los nodos implementan el algoritmo *backward learning* y que tiene lugar el siguiente envío de mensajes:

- Un *host* en la red C envía un mensaje a un *host* en la red E, en $t = 8$ s.
- Un *host* en la red B envía un mensaje a un *host* en la red D, en $t = 12$ s.

Responda a las siguientes tres cuestiones:

- i. ¿Cuáles serían las tablas de encaminamiento de los nodos justo antes de iniciarse su actualización en $t = 15$ s?
- ii. ¿En qué instante temporal se consigue ahora la estabilidad de las tablas?
- iii. ¿Se modifican estas respecto de las obtenidas en a)?



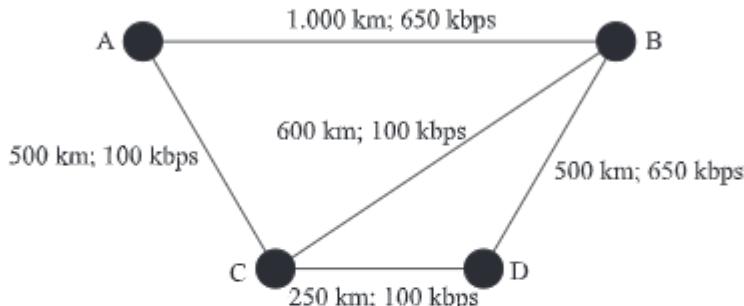
1210410

110

14. Actualice la tabla de encaminamiento del nodo D de la figura inferior suponiendo un protocolo de encaminamiento basado en distancia y que la expresión de la métrica para cada enlace es:

$$D = \alpha \times d + (1 - \alpha) \times B$$

donde d es la distancia de los nodos extremos en kilómetros, B el ancho de banda del enlace y $\alpha = 0,7$.



Las tablas de encaminamiento actuales son las siguientes:

Tabla nodo A		
Destino	Nodo siguiente	Distancia
A	—	—
B	B	760
C	C	380
D	C	580

Tabla nodo B		
Destino	Nodo siguiente	Distancia
A	A	760
B	—	—
C	C	450
D	D	410

Tabla nodo C		
Destino	Nodo siguiente	Distancia
A	A	760
B	B	450
C	—	—
D	D	205

Tabla nodo D		
Destino	Nodo siguiente	Distancia
A	B	955
B	B	410
C	C	205
D	—	—

15. Obtenga los árboles de expansión de la topología de red del Ejercicio 13, tomando como nodo raíz cada uno de los seis nodos existentes.
16. Discuta brevemente las funciones a implementar por parte de un *router* que interconectase una red Ethernet y una WLAN. ¿Y dos WLAN distintas?

BIBLIOGRAFÍA

- Black, U. N.: MPLS and Label Switching Networks. Prentice-Hall, 2002.
- Cisco: Configuring VLANs. Recurso online: <http://www.cisco.com/en/US/docs/switches/datacenter/nexus5000/sw/configuration/guide/cli/VLANs.html>.
- Duato, J.; Yalamanchili, S.; Ni, L. M.: Interconnection Networks: An Engineering Approach. Morgan Kaufmann, 2003.
- León-García, A.; Widjaja, I.: Redes de Comunicación. Conceptos Fundamentales y Arquitecturas Básicas. McGraw-Hill, 2001.
- Stallings, W.: High Speed Networks and Internets. Performance and Quality of Service. Prentice Hall, 2002. 2.^a edición.
- Stallings, W.: Comunicaciones y Redes de Computadores. Pearson Educación, 2004. 7.^a edición.
- Tanenbaum, A. S.; Wetherall, D. J.: Computer Networks. Prentice-Hall, 2011. 5.^a edición.



PROTOCOLOS PARA LA INTERCONEXIÓN DE REDES

- 9.1. Protocolo Internet: IP
- 9.2. Mensajes de control de Internet: protocolo ICMP
- 9.3. Encaminamiento dinámico en Internet
- 9.4. Encaminamiento multidestino en Internet

Tras la introducción a Internet realizada en el capítulo anterior, en este se da comienzo al estudio detallado de la arquitectura de red en que se basa aquella: TCP/IP. Siguiendo la metodología de la primera parte del texto, el presente capítulo se centra en la descripción de los servicios implementados en la más baja de las capas de TCP/IP, la de red, a través del estudio de los protocolos desarrollados y adoptados al efecto.

Comenzando por el más importante de ellos, y que constituye el núcleo de esta arquitectura, IP, tras él se presentarán otros protocolos de gran transcendencia como ICMP, orientado al control de ciertas eventualidades y situaciones en la subred, y RIP, OSPF y BGP, cuyo fin es la actualización dinámica de las tablas de encaminamiento de los *routers*.

Asimismo, se estudiarán las transmisiones *multicast* en Internet desde tres perspectivas: direccional, gestión de grupos y encaminamiento. En relación a cada una de ellas se discutirán, respectivamente, la arquitectura MALLOC, el protocolo IGMP y el protocolo PIM.

9.1. Protocolo Internet: IP

Como se comentó a partir de la Figura 8.5, la PDU de la capa de red se encapsula en el campo de datos de la PDU de la capa inferior (típicamente una trama de enlace) sobre la que se implementa la arquitectura TCP/IP. El protocolo de la capa red en TCP/IP es el conocido como protocolo Internet o IP

(«Internet Protocol»), encargado del encaminamiento de los datos y que presenta las siguientes características principales:

1. Ofrece un servicio no orientado a conexión basado en la técnica de transmisión de paquetes mediante datagramas, ya estudiada en el Apartado 6.2.2 del texto.
2. No implementa control de flujo ni de errores, proporcionando así un servicio de envío no fiable. Con ello se descarga a la subred de cómputo, relegando dichas funciones a las capas superiores, las cuales se implementan en las estaciones finales o *hosts*.
3. IP se dice de *mejor esfuerzo* («best effort») ya que trata de enviar los datos con las máximas garantías posibles, dentro, eso sí, de las limitaciones impuestas por las dos características anteriores.

9.1.1. Datagrama y funcionalidad IP

Las características antes referidas quedan reflejadas en el formato de la PDU IP, conocida como paquete IP o, haciendo referencia al tipo de transmisión llevada a cabo, *datagrama IP*. Detallado originalmente en el RFC 791, el formato del paquete IP en su versión 4 (IPv4) es el mostrado en la Figura 9.1. Comenzaremos haciendo mención a los siguientes cinco campos «menores» en cuanto a funcionalidad del protocolo:

- *V* (4 bits): versión del paquete IP; 4 para el caso de la Figura 9.1.
- *LC* (4 bits): longitud de la cabecera IP en palabras de 32 bits. Los campos que constituyen la cabecera IP son los sombreados en la Figura 9.1. De todos ellos, solo los campos *opciones* y *relleno* son optativos, por lo que el mínimo valor del campo *LC* es 5, es decir, la longitud mínima de la cabecera es 20 octetos. También hemos de indicar que la longitud máxima total permitida para la cabecera (incluidas las opciones) es 60 octetos.
- *Longitud total*: campo de 16 bits que indica el número de octetos de que consta el datagrama IP completo, incluido el campo *datos*.
- *Protocolo*: campo de 8 bits que identifica al protocolo que generó el paquete IP. Estos valores se especifican en el RFC 1700 (actualizado por el RFC 3232), siendo algunos ejemplos: 1 = ICMP, 2 = IGMP, 6 = TCP, 17 = UDP.
- *Datos*: campo de tamaño igual a *longitud total*-(*LC*×4) octetos, donde se encapsula la información correspondiente al protocolo que generó el paquete IP.

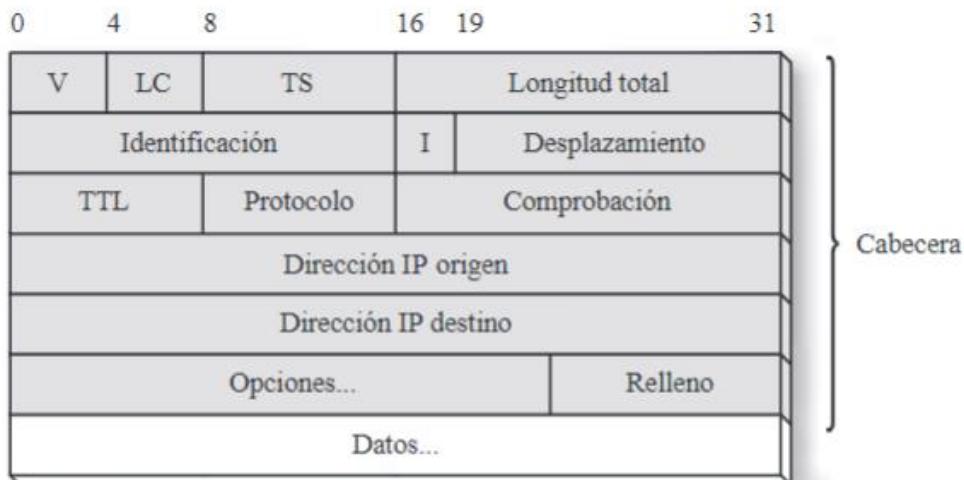


Figura 9.1. Formato del datagrama IPv4.

El resto de campos del paquete IP se describen en mayor profundidad en los siguientes apartados, organizados en base a la funcionalidad o servicio para el que están pensados.

Envío datagrama no fiable

Comentados los campos «menores» previos, a continuación vamos a centrar la explicación en otros más importantes en cuanto a la funcionalidad desarrollada. Comenzaremos haciendo referencia a aquellos relacionados con las dos primeras características mencionadas al comienzo del Apartado 9.1:

- Dado que la transmisión se basa en el esquema datagrama, cada paquete debe contener un identificador para posibilitar la reconstrucción del mensaje en el extremo receptor¹. Asimismo, deben especificarse las direcciones origen y destino a fin de permitir la identificación del remitente y el encaminamiento individualizado de cada paquete.

El primer hecho se recoge a través del campo de 16 bits *identificación* del datagrama IP (Figura 9.1), el cual no es más que el número de orden del paquete dentro del mensaje: 1, 2, 3, ... La segunda de las cuestiones mencionadas se corresponde con la especificación de los campos *dirección IP origen* y *dirección IP destino*, direcciones que, como se indicó en el Apartado 8.4 son de 32 bits para la versión 4 de IP aquí considerada.

- Por su parte, el carácter no fiable de IP se evidencia en la no existencia de campo de comprobación de errores alguno que controle la ocurrencia de estos en los datos transportados. En este sentido, aunque el datagrama IP incluye el campo *comprobación*, a través de él solo se controlan los campos que forman la cabecera del datagrama, pero no la información de las capas superiores incluida en el campo *datos*.

Frente a otros tipos de esquemas de comprobación de errores tales como el CRC estudiado en el Apartado 4.3.2 y usado, por ejemplo, en los estándares MAC IEEE 802, en IP se realiza un procedimiento mucho más simple: *complemento a 1 de la suma complemento a 1 de las palabras de 16 bits que componen la cabecera, excepto, por supuesto, el propio campo de comprobación*.

Un campo importante relacionado con el carácter no orientado a conexión del protocolo IP es el de *tiempo de vida* (TTL, «Time To Live»). Debido a posibles problemas de inconsistencia en las tablas de encaminamiento, podría ocurrir que un paquete dado entrase en un bucle en la transmisión, lo que resultaría a todas luces indeseable (especialmente desde el punto de vista de la posible congestión de la red). Para evitar este hecho, cada paquete se marca en origen con un tiempo máximo de vida, de forma que cada vez que este se retransmite sobre una línea de salida, el nodo intermedio correspondiente decrementa el campo TTL en uno. Además, en cada *router* se decrementa adicionalmente en tanto tiempo como haya permanecido almacenado localmente hasta su envío efectivo. Cuando el campo TTL alcanza el valor cero, el nodo correspondiente concluye que el paquete ha permanecido «demasiado» tiempo en la red y lo elimina de la misma.

Como es evidente, este campo implica la modificación de la cabecera de los paquetes IP por parte de los nodos intermedios, lo que conlleva el re-cálculo del campo *comprobación*.

Fragmentación IP

Según se deduce de los campos *longitud total* y *LC* ya comentados, el tamaño máximo del datagrama IPv4 es $2^{16} - 1 = 65.535$ octetos, de los cuales $65.535 - 20 = 65.515$ octetos como máximo corresponderán a datos. El transporte de una cantidad de datos tal supera con creces la MTU usual de redes

¹ Recuérdese del Capítulo 6, Apartado 6.4.2, las razones por las que no resulta aconsejable el ensamblado del mensaje en los nodos intermedios de la subred.

como las IEEE 802 estudiadas en el Capítulo 5. Es por ello que se hace preciso tener la capacidad de dividir el paquete IP en fragmentos de menor tamaño², de forma que puedan ser encapsulados y transmitidos en las redes subyacentes de transporte. La fragmentación de los datagramas IP se controla a través de los campos *indicadores* (I), *desplazamiento* e *identificación* de la siguiente forma:

- Ante la necesidad de fragmentar un paquete IP dado, cabría la posibilidad de disponer de un campo adicional al de *identificación* donde se especificase el orden de fragmento dentro del paquete. Dada la potencial necesidad de fragmentaciones posteriores, se emplea un esquema mucho más flexible basado en el uso del campo *desplazamiento* («offset» en inglés). Este, con una longitud de 13 bits, es un puntero que indica la posición relativa del primer byte del fragmento dentro del datagrama original. Esta posición no viene dada, como cabría esperar, en octetos, sino en grupos de 8 bytes, esto es, de 64 bits. La razón es simple: *para poder direccionar la longitud total del datagrama (16 bits de longitud) a través del campo desplazamiento (13 bits de longitud)*, se hace preciso introducir un factor 2^3 de manera que $2^{13} \times 2^3 = 2^{16}$.

Para aclarar esta cuestión pensemos a modo de ejemplo en la transmisión de un paquete IP de longitud total 4.200 octetos (20 de cabecera más 4.180 de datos) sobre una red Ethernet con MTU igual a 1.500 bytes (Figura 9.2). Suponiendo una cabecera de 20 octetos en todos los casos, la transmisión de dicho paquete se podría hacer dividiéndolo en dos fragmentos de 1.480 octetos de datos y uno de 1.220 ($4.180 = 2 \times 1.480 + 1 \times 1.220$). De esta forma, el campo *desplazamiento* del primer fragmento tomaría el valor 0 (es decir, el primer byte de datos del primer fragmento corresponde al inicial del datagrama original), el segundo el valor 185 ($185 \times 2^3 = 1.480$) y el tercero el valor 370 ($370 \times 2^3 = 2.960$). Si, por ejemplo, el segundo fragmento sufriese una división posterior, digamos, en dos fragmentos de 1.000 y 480 octetos de datos, bastaría con especificar en el campo *desplazamiento* de estos fragmentos los valores 185 ($185 \times 2^3 = 1.480$) y 310 ($310 \times 2^3 = 2.480 = 1.480 + 1.000$), respectivamente. Como

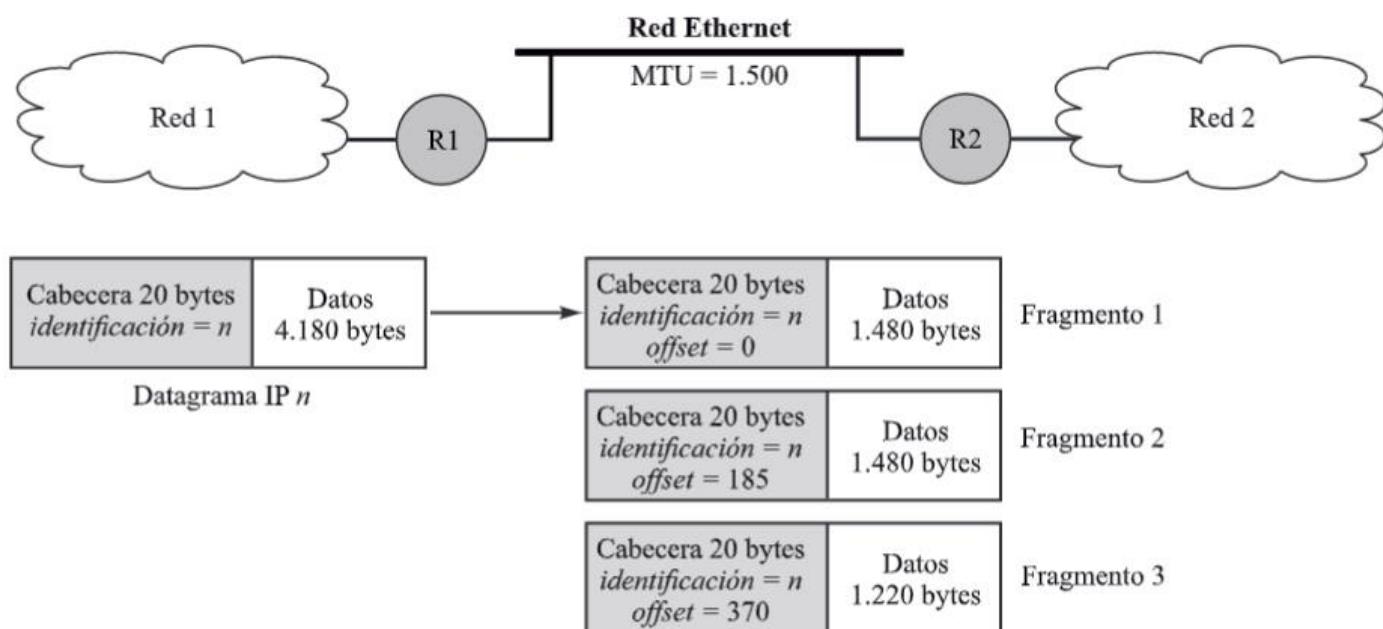


Figura 9.2. Ejemplo de fragmentación IP.

² De nuevo volvemos a hacer referencia al tema de la fragmentación de paquetes ya tratado en el Apartado 6.4.2 del texto.

se observa, el valor del campo *desplazamiento* del tercer fragmento ($370 \times 2^3 = 2.960$) sigue siendo válido.

Adicionalmente hay que señalar que, como es evidente, todos los fragmentos especificarán en el campo *identificación* el mismo valor de datagrama (el original) para permitir su correcto ensamblado en el receptor.

Este esquema permite, por tanto, identificar de forma unívoca, además de sencilla, cada fragmento IP dentro del datagrama original para el ensamblado posterior del mensaje completo en el destino.

- El campo *indicadores* (I) del paquete IP también está relacionado con la función de fragmentación. Este campo consta de 3 bits, si bien hemos de indicar que el situado más a la izquierda según se muestra en la Figura 9.1 no se utiliza. El siguiente bit situado a la derecha del anterior se denomina *DF* (del inglés «Don't Fragment») y se utiliza para indicar, por parte del emisor, la no autorización a fragmentar el paquete IP. En tal caso, ante la potencial imposibilidad de enviar el paquete, este podría ser rechazado por la red. El tercero de los bits del campo *I* es el conocido como *MF* («More Fragments») y sirve para señalar el último fragmento (*MF* = 0) de un datagrama frente al resto (*MF* = 1). Dicha distinción, junto con los campos *desplazamiento* e *identificación*, es necesaria para posibilitar la reconstrucción fiel del paquete y, en consecuencia, del mensaje original.

Tipo de servicio y opciones IP

Tras todos los anteriores, a continuación se explican los campos *tipo de servicio* (TS) y *opciones*:

- El campo de 8 bits *tipo de servicio* (TS, del inglés «Type of Service») está pensado para mejorar la eficiencia de las transmisiones IP a través de la especificación de ciertas preferencias de envío por el emisor. Como tal, este campo es desiderativo, es decir, no vinculante para los nodos intermedios. Con el formato indicado en la Figura 9.3(a), TS permite llevar a cabo la solicitud de la transmisión del datagrama según un criterio de mínimo retardo (bit 3 activo, D, «Delay»), de máximo rendimiento (bit 4 activo, T, «Throughput»), de máxima fiabilidad (bit 5 activo, R, «Reliability») o de mínimo coste económico (bit 6 activo, C, «monetary Cost»)³. Es decir, en un nodo dado, si existe la posibilidad de elección, el *router* elegirá la línea de salida de acuerdo con las preferencias expresadas en este campo del datagrama.

Adicionalmente al tipo de transmisión deseada (que no garantizada), los tres primeros bits del campo *TS* indican la prioridad, de 0 a 7, del paquete IP. Este subcampo permitiría, por tanto, dar prioridad a unos paquetes frente a otros.

0	1	2	3	4	5	6	7
Prioridad	D	T	R	C	Sin uso		

(a)

0	1	2	3	4	5	6	7
Copia	Clase	Número					

(b)

Figura 9.3. Campos tipo de servicio (a) y opciones (b) del datagrama IP.

³ El bit *C* del campo *TS* no existe en el RFC 791, sino en actualizaciones posteriores al mismo: RFC 1349, 2474 y 3260.

Para concluir el estudio del campo *TS* es de mencionar que este campo se usa en la práctica muy poco; de hecho, alternativamente en el RFC 2481 se propone una redefinición del mismo (en concreto los bits 6 y 7) para llevar a cabo la notificación explícita de congestión en conjunción con TCP.

- Con una longitud máxima permitida de 40 octetos, el campo *opciones* permite llevar a cabo algunas funciones de test y depuración. Cada opción especificada debe comenzar con el octeto indicado en la Figura 9.3(b). El primer bit, *copia*, indica si la opción en cuestión debe ser copiada en cada fragmento potencial del datagrama. Si este bit toma el valor 0, la opción solo se copiará en el primer fragmento y no en el resto.

Existen cuatro *clases* de opciones IPv4:

- 0 → Control de red o datagrama.
- 2 → Depuración y test.
- 1 y 3 → Reservados para uso futuro.

Las distintas opciones dentro de cada clase se diferencian entre sí mediante un *número*, siendo las más destacables las siguientes:

- a) *Registro de ruta*. Esta opción solicita que cada dispositivo de encaminamiento atravesado por el paquete especifique la dirección IP de la interfaz de salida dentro del mismo. Comenzando con el octeto mostrado en la Figura 9.3(b), con los subcampos *clase* = 0 y *número* = 7, el resto de campos de esta opción son los indicados en la Figura 9.4(a):
 - *Longitud*: número de octetos totales que forman la opción.
 - *Dirección IP i*: dirección IP de salida del *i*-ésimo dispositivo de encaminamiento atravesado por el paquete.
 - *Puntero*: próxima posición libre dentro del campo de opción *registro de ruta* en la que un nuevo dispositivo de encaminamiento almacenará su dirección.
- b) *Encaminamiento desde el origen*. Esta opción permite el encaminamiento del paquete siguiendo la ruta especificada por el emisor del mismo, según se discutió en el Apartado 6.3.4. De esta forma, cada dispositivo de encaminamiento se limitará a extraer la próxima dirección IP especificada a través del campo *puntero* (Figura 9.4(a)) y a retransmitir el paquete en consecuencia. El campo *puntero* será incrementado para cada nueva dirección IP accedida.

Existen dos tipos de encaminamiento desde el origen: *estricto* (*clase* = 0, *número* = 9) y *flexible* (*clase* = 0, *número* = 3). Ambos con el campo *copia* a valor 1, en el primero se trata de seguir exactamente la secuencia de direcciones IP especificadas en el campo de opción correspondiente. En cambio, en el segundo se deja libertad a la subred para que entre cada dos direcciones IP indicadas en el paquete se puedan atravesar otras.
- c) *Sello de tiempo*. Esta opción es similar a la de registro de ruta, pero grabándose no solo la dirección IP de los dispositivos de encaminamiento atravesados por el paquete, sino también el instante en el que esto sucedió. Dicha marca de tiempo (Figura 9.4(b)) se expresa a través de 32 bits e indica los milisegundos transcurridos desde medianoche UTC.

Como en los casos anteriores, el campo *puntero* se incrementa consecuentemente en cada salto e indica la posición dentro del campo opción *sello de tiempo* en la que el siguiente dispositivo de encaminamiento debe grabar su dirección y el sello de tiempo correspondiente.

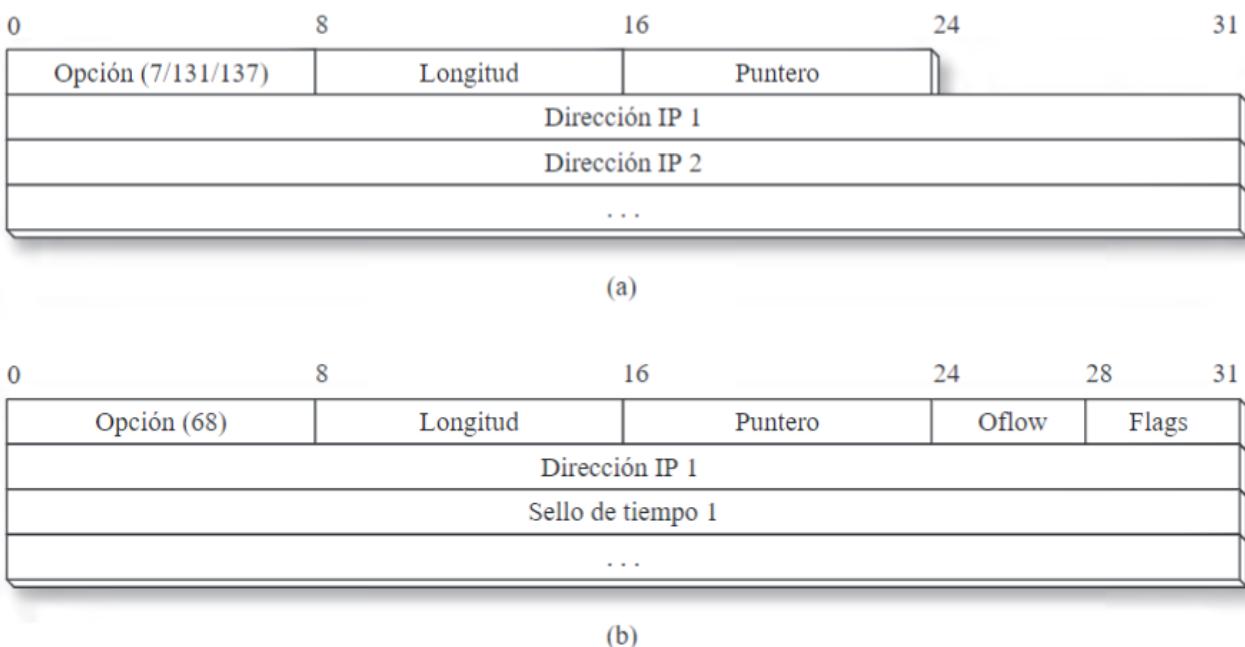


Figura 9.4. Formato de las opciones IP *registro de ruta y encaminamiento desde el origen* (a) y *sello de tiempo* (b).

Otros dos campos especificados en esta opción son los siguientes:

- *Flags*: de 4 bits de longitud, este campo indica la información a grabar por parte de los dispositivos de encaminamiento atravesados:
 - 0 → Grabar solo sello de tiempo.
 - 1 → Grabar dirección IP y sello de tiempo.
 - 3 → Las direcciones IP están especificadas por el emisor, de modo que un dispositivo de encaminamiento solo grabará el sello de tiempo si la siguiente dirección IP en la lista coincide con la suya.
- *Oflow*: campo de 4 bits utilizado para indicar el número de dispositivos de encaminamiento que no han podido grabar el sello de tiempo porque la longitud de la opción era demasiado pequeña.

Un comentario final acerca del campo *opciones* es que este no tiene forzosamente una longitud total múltiplo de 32 bits. Ello hace que se requiera un campo de *relleno* para, llegado el caso, completar la longitud del paquete hasta un múltiplo de 32 bits, haciendo así coherente el valor del campo *LC* del datagrama. Este campo indica, recordemos, el número de palabras de 32 bits de que consta la cabecera IP, dentro de la que se contabiliza el campo *opciones*. El campo *relleno* consiste en una secuencia de «todo ceros».

9.1.2. IPv6

Casi dos décadas después de la especificación formal de IPv4 apareció la versión 6 del protocolo IP: IPv6 (véase RFC 2460). Las características principales de este son:

- *Capacidades de direccionamiento extendidas*. En el caso de IPv6 se hace uso de direcciones IP de longitud 128 bits en lugar de las de 32 bits contempladas en IPv4.
- *Capacidad de etiquetado de flujo*. Cada paquete IPv6 se etiqueta con una marca identificativa del tráfico para el que el emisor desea, por ejemplo, una calidad de servicio dada.

- *Formato de cabecera simplificado y flexible.* Frente a la cabecera de formato fijo utilizada en IPv4, el datagrama IPv6 se desarrolla como una serie de cabeceras extendidas opcionales (véase Figura 9.5(a)). Esto proporciona una mayor simplicidad a la cabecera base, además de una mayor flexibilidad al protocolo, posibilitando su expansión natural a capacidades adicionales tales como nuevas tecnologías subyacentes o nuevas aplicaciones.
- *Autenticación y privacidad.* Como particularización de las cabeceras extendidas, hemos de destacar la existencia de unas específicas que permiten la provisión de seguridad en las transmisiones (véase Capítulo 12). Este nuevo aspecto de IP resulta de enorme interés actual en el contexto de las comunicaciones y la compartición de información.

El formato del paquete IPv6 es el indicado en la Figura 9.5(a), donde se muestra la disposición de las cabeceras extendidas tras la cabecera base. En relación a esta segunda, los campos que la componen son (Figura 9.5(b)):

- *Versión* (4 bits): para que los nodos intermedios puedan saber si se trata de un paquete IPv4 o IPv6, el primer campo del datagrama es, como en IPv4, el de versión. En el caso que nos ocupa el valor del campo será, obviamente, igual a 6.
- *Prioridad* (4 bits): campo que permite al origen indicar la prioridad deseada para sus paquetes, en relación a otros paquetes enviados. Los valores 0-7 se utilizan para el tráfico para el que el origen lleva a cabo control de congestión, y los valores 8-15 para el resto.
- *Etiqueta de flujo* (24 bits): todos los paquetes correspondientes a un mismo «flujo» serán transmitidos con las mismas direcciones IP origen y destino, la misma prioridad y la misma etiqueta de flujo. Esto permite a los nodos intermedios gestionar adecuadamente distintos flujos de datos, con diferentes requisitos de QoS (ver Apartado 7.2)
- *Longitud de datos* (16 bits): campo para indicar la longitud, en octetos, del campo de datos o *payload* del paquete.
- *Siguiente cabecera* (8 bits): identifica el tipo de cabecera de extensión que sigue inmediatamente a la fija de 40 octetos de IPv6. Si no existiese ninguna cabecera opcional tras la base, el valor de este campo sería 59.

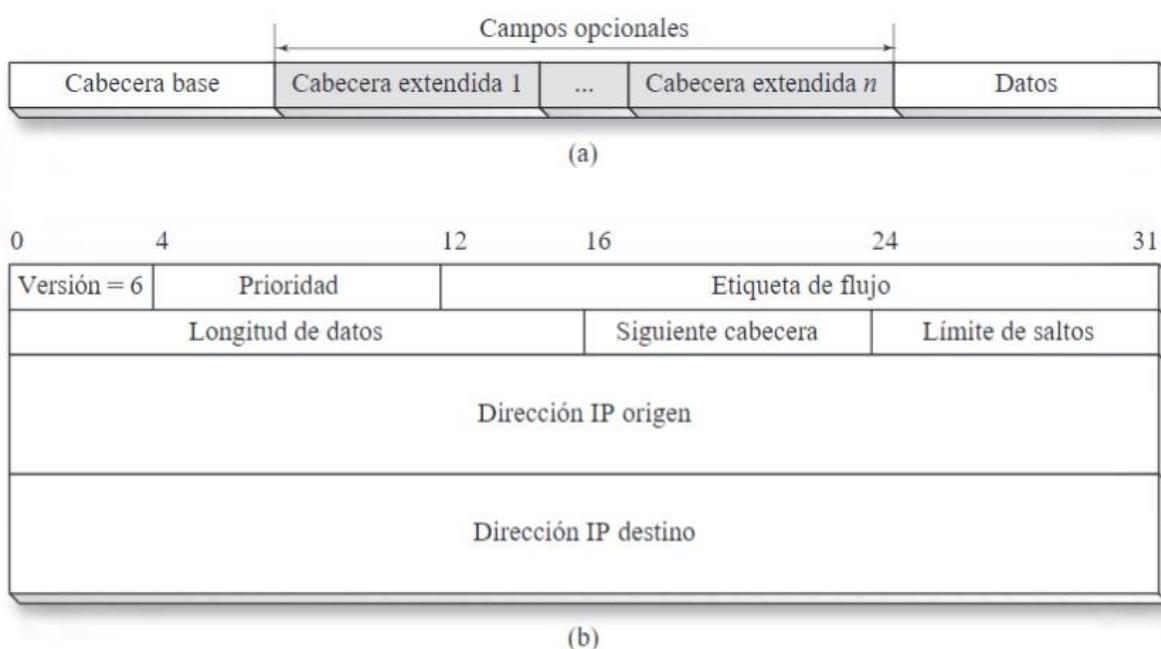


Figura 9.5. Formato general del paquete IPv6 (a) y campos de la cabecera base (b).

- *Límite de saltos* (8 bits): establece el número máximo de saltos permitido para un datagrama. Esto es, tiene la misma función que el campo TTL del paquete IPv4 pero expresado en saltos en lugar de en tiempo para reducir las necesidades de cómputo en los nodos intermedios. Así, cada nodo atravesado por el paquete decrementará este campo en 1, de manera que el paquete será descartado cuando el valor del campo sea 0.
- *Direcciones IP origen y destino* (128 bits cada una): como ya se ha indicado, las direcciones IPv6 son de 128 bits e identifican, como no puede ser de otro modo, las máquinas origen y destino del datagrama. Más adelante se profundiza en la nomenclatura general utilizada para este tipo de direccionamiento.

Tras la cabecera base, fija, en IPv6 pueden aparecer otros campos opcionales (Figura 9.5(a)) en lo que se denomina cabeceras de extensión, cuyo objetivo puede ser diverso (seguridad, fragmentación, etc.). Más adelante se discuten con más detalle estas cabeceras de extensión; antes de ello, sin embargo, veamos el direccionamiento utilizado en IPv6.

Direccionamiento IPv6

Como en el caso de IPv4, una dirección IPv6 es una etiqueta numérica que identifica una interfaz de red. En el caso de una dirección *unicast*, se trata de una interfaz específica conocida. En el caso de direcciones IP *multicast* (véase Capítulo 8) se identifica un grupo de dispositivos que participan de un mismo servicio. Es de mencionar que en IPv6 las direcciones *broadcast* se consideran un caso particular de las *multicast*.

Adicionalmente a estos tipos, IPv6 introduce un nuevo conjunto de direcciones conocidas como *anycast* (RFC 1546). Una dirección IP de este tipo identifica un grupo de interfaces de modo que un paquete *anycast* se envía a solo una de ellas, generalmente la más cercana.

Una dirección IPv6 se representa mediante ocho grupos de cuatro dígitos hexadecimales (expresados en letra minúscula), cada uno de los grupos representando en consecuencia 16 bits (2 octetos), y separados entre sí mediante el carácter «::» (véase RFC 2373). Ejemplo de todo lo anterior es la dirección IPv6 4ce2:0000:0000:28c9:82ea:dba9:07fa:0001.

Con objeto de simplificar la notación de las direcciones se permite la supresión de los 0 existentes en cada grupo. Así, la dirección ejemplo anterior quedaría 4ce2:0:0:28c9:82ea:dba9:7fa:1. También por simplicidad se acorta la dirección sustituyendo la secuencia de grupos «todo ceros» más larga por «::». En el caso anterior tendríamos 4ce2::28c9:82ea:dba9:7fa:1. En caso de que existiese más de una cadena de grupos «todo ceros» de la misma longitud, la sustitución se haría solo para la situada más a la izquierda. Por ejemplo, si tuviésemos la dirección 4ce2:0:0:28c9:0:0:7fa:1, la identificación final sería 4ce2::28c9:0:0:7fa:1.

Las direcciones *unicast* y *anycast* están compuestas usualmente, como sucede en IPv4, de dos partes lógicas: un prefijo de red de 64 bits usado para *routing* (véanse los protocolos de encaminamiento estudiados más adelante en este capítulo) y un identificador de interfaz de red (o *host*), también de 64 bits (Figura 9.6). Del grupo primero, los 16 bits menos significativos pueden utilizarse para identificar subredes dentro de una misma red. Frente a las direcciones anteriores, las *multicast* pueden tener distintos formatos dependiendo de la aplicación, si bien todas comienzan por el byte 11111111.

Prefijo para <i>routing</i> (64 bits)		Interfaz (64 bits)
Prefijo de red	Subred	Identificador de <i>host</i>

Figura 9.6. Formato general de direcciones *unicast* en IPv6.

Los rangos de direcciones de red se escriben en notación CIDR, de modo que, por ejemplo, la red 4ce2::28c9:0:0/96 comienza en la dirección 4ce2::28c9:0:0:0000:0000 y finaliza en la 4ce2::28c9:0:0:ffff:ffff. Por otro lado, a fin de facilitar el transitorio entre el direccionamiento IPv4 y el IPv6, el prefijo ::ffff:0:0/96 designa una dirección IPv4. Por ejemplo, la dirección IPv6 ::ffff:0:0:c0a8:10f correspondería a la IPv4 192.168.1.15. Adicionalmente a este hecho, está permitido escribir los 32 últimos bits de una dirección IPv6-IPv4 en la notación decimal con puntos. Así, la dirección anterior podría especificarse en IPv6 como ::ffff:0:0:192.168.1.15.

Cabe también mencionar la disposición de algunas direcciones *unicast* reservadas como son: ::/0, ruta por defecto; ::1/128, autobucle (*localhost*); ::/128, no especificada (p.e., un *host* iniciándose); y fc00::/7, comunicaciones locales. Del mismo modo, las direcciones *multicast* ff00::0/8 están reservadas para distintos usos y no pueden ser usadas para indicar grupos. También las 128 direcciones *anycast* más altas dentro de cada prefijo de subred (/64) están reservadas, lo que significa que tienen 57 bits a valor 1 seguidos de 7 bits que identifican la identidad *anycast*.

Para concluir esta breve discusión sobre el direccionamiento IPv6, hemos de hacer dos reseñas importantes. Por una parte, que una dirección puede tener un tiempo de uso limitado y que, frente al carácter habitual de direcciones únicas, estáticas en IPv4, una interfaz puede tener asociadas más de una dirección IPv6 y que estas pueden ser creadas temporalmente mediante cadenas aleatorias variables en el tiempo.

Por otro lado, comentar que si deseamos indicar una dirección IPv6 directamente en un enlace URL, esta ha de aparecer entre corchetes ('[]'). Por ejemplo, en el caso [http://\[2001:f8f:3400::34:47b9\]:8080](http://[2001:f8f:3400::34:47b9]:8080) estaríamos solicitando un recurso al puerto 8080 de la dirección 2001:f8f:3400::34:47b9.

Cabeceras de extensión

Como hemos mencionado con anterioridad, adicionalmente a la parte fija de la cabecera, IPv6 contempla la inclusión de una o más cabeceras opcionales a través de las cuales se posibilitan distintas funciones de interés. Son las denominadas cabeceras de extensión, encontrándose definidas en el RFC 2460 las siguientes, todas ellas con una longitud total múltiplo de 64 bits:

— *Fragmentación*. Como vimos en IPv4, las funciones de fragmentación y ensamblado son funciones necesarias para permitir la transmisión de paquetes de tamaño superior al soportado por las MTU asociadas a las rutas. Esto sigue siendo válido en IPv6 con la salvedad de que el proceso de fragmentación no se realiza en los nodos intermedios, sino exclusivamente en el origen. Para ello, previamente a la transmisión, el emisor debe llevar a cabo un procedimiento de *descubrimiento de MTU de ruta* en base al empleo del bit *DF* ya conocido de IP y protocolo ICMP (véase Apartado 9.2 más adelante), a fin de conocer la MTU mínima en la ruta hacia el destino. Fragmentados en consecuencia los paquetes de longitud mayor a la MTU mínima, estos se transmitirán hacia el destino en paquetes con la cabecera de extensión mostrada en la Figura 9.7(a). Los campos de esta cabecera son los siguientes:

- *Siguiente cabecera*: campo de 8 bits que indica el tipo de la siguiente cabecera extendida. Si no existiese ninguna más, es decir, si a continuación siguiese el campo *datos* del paquete IPv6, el valor del campo *siguiente cabecera* será 59.
- *Reservado*: campo a valor 0 ignorado en la recepción.
- *Desplazamiento*: campo de 13 bits que indica, en unidades de 8 octetos, la posición del fragmento respecto al inicio del paquete original sin fragmentar. Nótese la correspondencia de este campo con el homónimo visto para el paquete IPv4.

- *MF*: campo de 3 bits en el que los dos primeros toman el valor 0 y se indica a través del último que el fragmento en cuestión es (*MF* = 0) o no (*MF* = 1) el último de los que componen el datagrama original.
- *Identificación*: campo de 32 bits con el que se numera cada paquete del mensaje. Como en el caso de IPv4, todos los fragmentos correspondientes a un mismo datagrama tendrán el mismo valor de este campo. Obsérvese que la longitud de este campo en IPv4 era solo de 16 bits, permitiendo su ampliación al doble de bits una mejor adecuación de IPv6 a redes de alta velocidad y mensajes de gran tamaño.

De acuerdo con el RFC 1700, la cabecera de extensión de fragmentación se referencia con el valor 44 en el campo *siguiente cabecera* de la cabecera precedente a esta en el paquete IPv6.

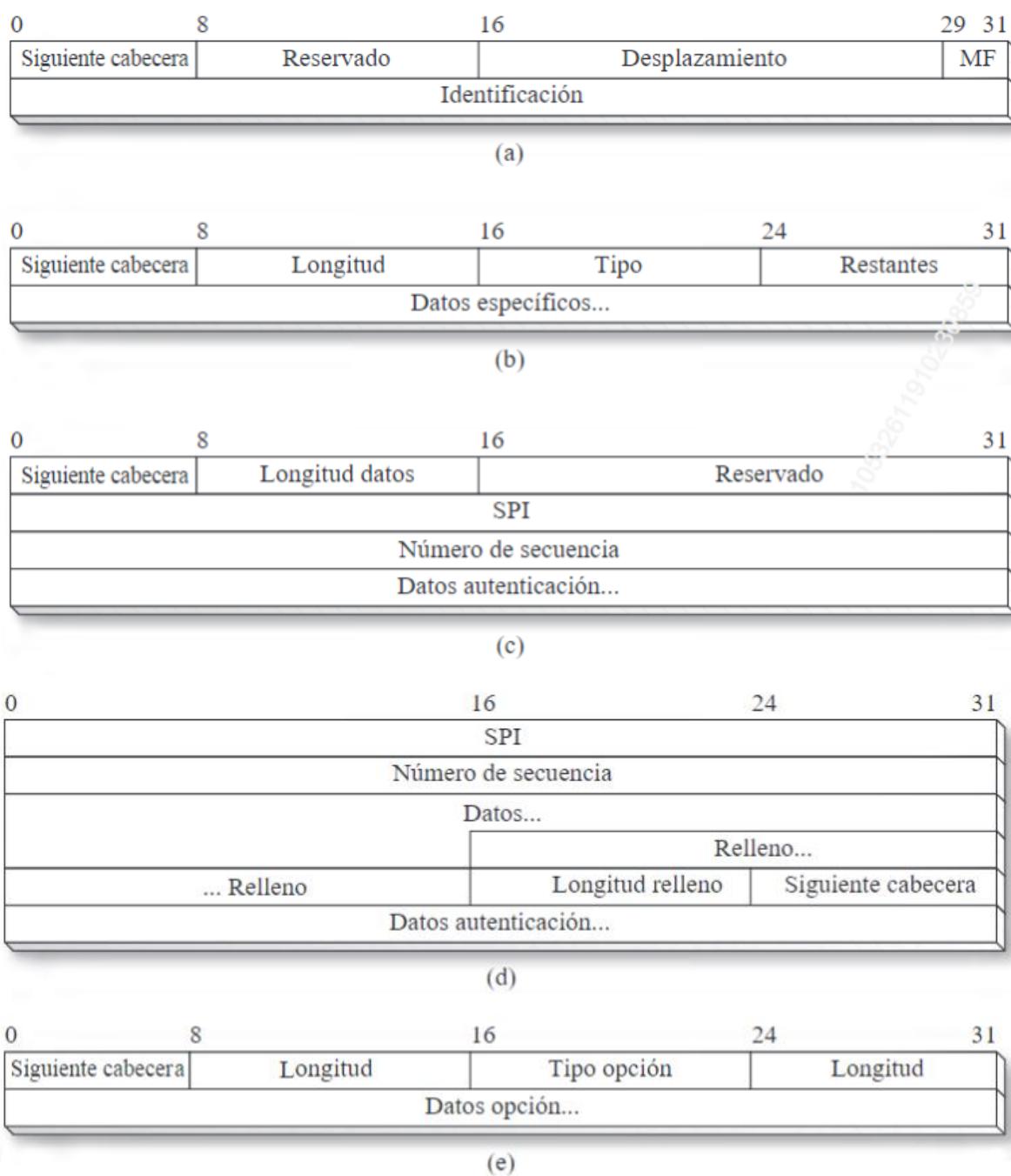


Figura 9.7. Cabeceras de extensión IPv6 de fragmentación (a), de encaminamiento (b), de autenticación (c), de encapsulado de seguridad (d) y genérica de opciones (e).

— *Encaminamiento.* Análoga a la opción de *encaminamiento desde el origen* o a la de *registro de ruta* en IPv4, esta cabecera IPv6 permite especificar por parte del origen la ruta a seguir por un paquete dado. El formato de esta cabecera es el mostrado en la Figura 9.7(b):

- *Siguiente cabecera:* indica el tipo de la siguiente cabecera extendida. Si no existiese ninguna más, esto es, si a continuación siguiese el campo datos del paquete IPv6, el valor del campo siguiente cabecera será 59.
- *Longitud:* longitud total de esta cabecera, en unidades de 8 octetos.
- *Tipo:* campo de 8 bits para indicar el tipo específico de encaminamiento (ver campo *datos específicos* más adelante o RFC para más detalles).
- *Restantes:* número de nodos intermedios que restan por visitar hasta llegar al destino.
- *Datos específicos:* campo de longitud variable definido por el tipo de encaminamiento. En el caso tipo = 0, el formato del campo *datos específicos* consiste en una lista de direcciones IP a visitar, precedida por 32 bits a valor 0.

La cabecera de extensión de encaminamiento se referencia con el valor 43 en el campo *siguiente cabecera* de la cabecera precedente a esta en el paquete IPv6. También cabe mencionar que esta opción está habitualmente deshabilitada en los *routers* por cuestiones de seguridad.

— *Autenticación y encapsulado de seguridad.* En los RFC 2402 y 2406 se describen, respectivamente, las cabeceras de extensión de autenticación (AH, «Authentication Header») y de encapsulado seguro de los datos (ESP, «Encapsulating Security Payload»). Aunque no se han descrito hasta aquí, es importante mencionar que ambos tipos de cabecera se pueden utilizar tanto en IPv4 como en IPv6.

Con la cabecera AH se permite la autenticación de un paquete y la comprobación de su integridad. Para ello, los campos de la cabecera de extensión directamente relacionados con esta función son los siguientes (Figura 9.7(c)):

- *SPI* («Security Parameters Index»): campo de 32 bits que, en combinación con la dirección IP de destino, identifica únicamente la asociación de seguridad para este paquete. Es decir, el conjunto de elecciones relacionadas con los algoritmos de cifrado, de resumen o *hash* y de autenticación a utilizar entre ambos extremos de la comunicación (véase Capítulo 12).
- *Número de secuencia:* valor monótonamente creciente a lo largo de la transmisión utilizado para la detección de paquetes duplicados. A través de esta técnica simple se intentan evitar ataques de repetición.
- *Datos autenticación:* campo de longitud variable que contiene el valor de comprobación de integridad y autenticación para este paquete. Algoritmos de autenticación empleados en el cálculo de este campo son DES (RFC 2405), MD5 (RFC 2403) y SHA-1 (RFC 2404) —véase Capítulo 12—.

La cabecera ESP proporciona confidencialidad, integridad y autenticación de los datos. Los campos de la cabecera ESP son los siguientes (Figura 9.7(d)):

- *SPI:* como en AH.
- *Número de secuencia:* como en AH.
- *Datos:* campo de longitud variable cuya naturaleza se especifica a través del campo *siguiente cabecera* existente más adelante; generalmente corresponde a una PDU de capa superior.
- *Relleno:* campo de longitud variable al que se recurre en caso de que el algoritmo de cifrado utilizado precise que la longitud del texto sea múltiplo de un cierto número de octetos.
- *Longitud relleno:* 8 bits para especificar la longitud del campo anterior.

- *Siguiente cabecera*: tipo de cabecera que sigue a esta en el paquete IPv6.
- *Datos de autenticación*: campo opcional donde se recogen los datos de autenticación o integridad en la forma descrita en el campo del mismo nombre de la cabecera AH anterior.

La cabecera de extensión de autenticación se referencia con el valor 51 en el campo *siguiente cabecera* de la cabecera precedente a esta en el paquete IP. Por su parte, la cabecera ESP se identifica mediante el valor 50 (ver RFC 1700). Como comentario adicional, decir que el uso de ambas cabeceras constituye el protocolo IPsec, ideado para proporcionar seguridad en las comunicaciones sobre IP (ver RFC 4301; Capítulo 12).

- *Otras opciones IPv6*. Adicionalmente a las cabeceras extendidas mencionadas, existen otras cuyo formato general es el indicado en la Figura 9.7(e). De entre ellas podemos destacar dos:

- *Salto-a-salto*: cabecera de referencia a valor 0 en el campo *siguiente cabecera* de la cabecera precedente, a través de la que se indican acciones a tomar por los dispositivos de enrutamiento atravesados en la ruta hacia el destino.
- *De destino*: frente a la anterior, esta opción especifica acciones que deben ser tomadas solo por el nodo destino del paquete.

La cabecera de extensión *de destino* se referencia con el valor 60 en el campo *siguiente cabecera* de la cabecera precedente a esta en el paquete IPv6.

Como ejemplo de algunas de las acciones a que se refieren las opciones anteriores podemos mencionar entre otras: rechazo del paquete, rechazo del paquete y envío en respuesta al origen de un mensaje ICMP de problema de parámetros (véase Apartado 9.2.1).

Para concluir el estudio de las cabeceras extendidas de IPv6, hemos de decir que se recomienda que el orden en que aparezcan en el paquete (si es que lo hacen) tras la cabecera base fija sea el siguiente: *salto-a-salto, de destino* (para las opciones a procesar en el primer destino más los destinos subsiguientes listados en la cabecera de encaminamiento), *encaminamiento, fragmentación, autenticación, encapsulado de seguridad y de destino* (para las opciones a procesar solo por el destino final del paquete).

9.2. Mensajes de control de Internet: protocolo ICMP

En el RFC 792 se describe el *protocolo de mensajes de control de Internet* (ICMP, «Internet Control Message Protocol»). ICMP es, como IP, un protocolo de la capa de red, lo que no significa que sea una alternativa al mismo, sino, por el contrario, un complemento. Además, ICMP es usuario de IP, es decir, los mensajes ICMP se encapsulan dentro de los paquetes IP.

Según se desprende del estudio de IP, cada dispositivo de encaminamiento funciona de forma relativamente independiente del resto, de manera que todo el sistema funcionará adecuadamente si, y solo si, lo hacen todos los dispositivos que lo forman. Desafortunadamente, esto no ocurre en un sistema real. Así, encapsulado en el datagrama IP, cuyo campo *protocolo* tomará el valor 1 (ver RFC 1700), el protocolo ICMP define un conjunto de mensajes para informar o señalizar sobre determinadas situaciones tales como inaccesibilidad de un destino, expiración del tiempo de vida de un datagrama IP, etc. Además, ICMP define otros mensajes adicionales (como el de eco) para facilitar el diagnóstico de posibles problemas en la red.

Todos los mensajes ICMP comienzan con los siguientes tres campos de cabecera (los sombreados en la Figura 9.8), con una longitud total de 32 bits:

- *Tipo* (8 bits): indica el tipo del mensaje.
- *Código* (8 bits): identifica un subtipo dentro del tipo.

- *Comprobación* (16 bits): complemento a 1 de la suma complemento a 1 de las palabras de 16 bits que componen el mensaje ICMP, usado para el control de errores.

Los mensajes ICMP más relevantes son los siguientes (Tabla 9.1 y Figura 9.8):

- *Eco*. Utilizada para testar la accesibilidad de un destino dado, esta funcionalidad ICMP implica la consideración de dos mensajes, uno de solicitud de eco (*tipo* = 8, *código* = 0) y otro de respuesta (*tipo* = 0, *código* = 0).

Implementado a través del comando de usuario *ping*, un emisor genera un mensaje ICMP de solicitud de eco como el mostrado en la Figura 9.8(a), que será contestado mediante un mensaje de respuesta por el destinatario, en caso de estar accesible. El campo *opcional* contiene un conjunto de datos arbitrarios que el receptor deberá devolver al emisor en la respuesta. Los campos *identificador* y *secuencia* se utilizan para hacer corresponder solicitudes con respuestas. Como resultado, *ping* muestra por pantalla el tiempo consumido hasta el destino especificado.

- *Destino inalcanzable*. El mensaje ICMP tipo 3 (Figura 9.8(b)) se genera cuando el destino IP especificado en un datagrama dado no es accesible. Diversas pueden ser las causas que motiven esta inaccesibilidad, lo cual se especifica a través del campo *código* con los siguientes valores: 0 → red inalcanzable, 1 → *host* inalcanzable, 2 → protocolo inaccesible, 3 → puerto inaccesible, 4 → se precisa fragmentación y el bit DF está activo, 5 → fallo en ruta de origen, 6 → red destino desconocida, 7 → *host* destino desconocido, 8 → *host* origen aislado, 9 → comunicación prohibida con la red destino, 10 → comunicación prohibida con el *host* destino, 11 → red inaccesible por el tipo de servicio (campo *TS* en el paquete IP), 12 → *host* inaccesible por el tipo de servicio (campo *TS* en el paquete IP).

Dado que gran parte de los elementos mencionados anteriormente se especifican en la cabecera del protocolo IP o en el de nivel superior, el mensaje ICMP de *destino inalcanzable* incluye tanto la cabecera IP como parte de la PDU de la capa superior (a través de los 64 primeros bits del campo *datos* del datagrama IP) a fin de que el origen que generó el paquete al que se refiere el mensaje ICMP pueda realizar las comprobaciones oportunas.

- *Ralentización del origen* (del inglés «source quench»). El tipo de mensaje 4 consiste en una notificación explícita de congestión hacia atrás tal como se comentó en el Apartado 7.1.3. A través de este mensaje ICMP (Figura 9.8(c)), un dispositivo de encaminamiento comunica al *host* origen del paquete cuya cabecera y primeros 64 bits se especifican en el campo de datos del mensaje, que se está produciendo congestión en los recursos utilizados en su transmisión. Al recibir este mensaje, el *host* correspondiente debe reducir el flujo de emisión en la forma que se comentará en el Capítulo 10 (Apartado 10.3.5).

Tabla 9.1. Principales mensajes ICMP.

Campo tipo	Mensaje ICMP
8/0	Solicitud/respuesta de eco
3	Destino inalcanzable
4	Ralentización del origen
5	Redirecciónamiento
11	Tiempo excedido
12	Problema de parámetros
13/14	Solicitud/respuesta de sello de tiempo
17/18	Solicitud/respuesta de máscara de red

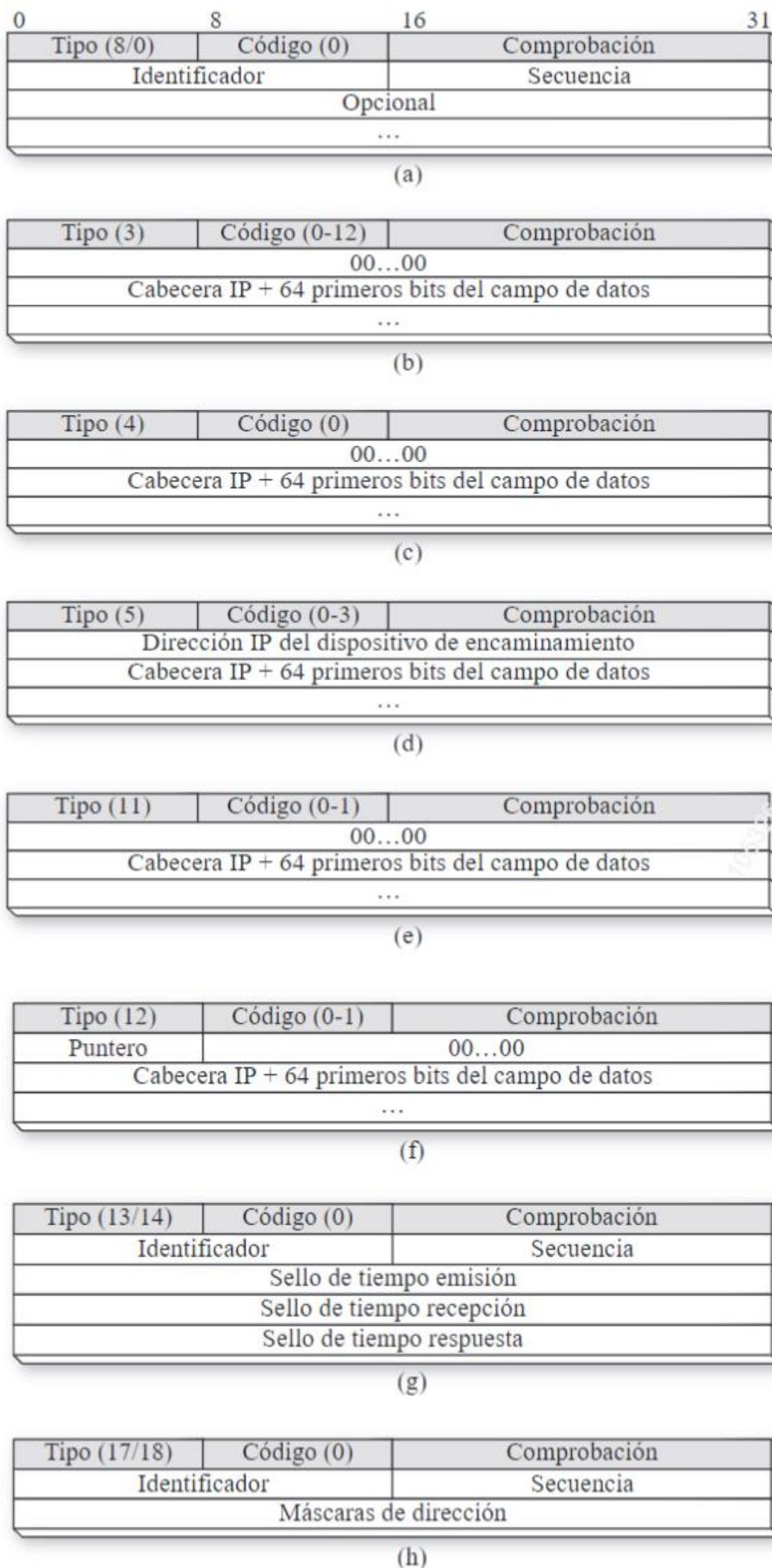


Figura 9.8. Formato de mensajes ICMP: (a) eco, (b) destino inalcanzable, (c) ralentización del origen, (d) re-direccionamiento, (e) tiempo excedido, (f) problema de parámetros, (g) sello de tiempo y (h) máscara de red. (Nota: los campos sombreados corresponden a la cabecera ICMP).

— *Redireccionamiento.* Los dispositivos de encaminamiento son los encargados de conocer y gestionar las rutas entre los distintos pares origen-destino. Así pues, en un momento dado, estos pueden comunicar a un *host* (que comparta una subred con ellos) que la ruta indicada para un destino especificado no es la mejor, sino que resulta más adecuada la que atraviesa el dispositivo cuya dirección IP se indica en el paquete ICMP de redireccionamiento, cuyo formato se muestra en la Figura 9.8(d).

Con este tipo de mensajes ICMP (*tipo* = 5) los *hosts* aprenderán las rutas de los dispositivos de encaminamiento. El campo *código* del paquete ICMP informa sobre el tipo de redireccionamiento al que se refiere el mensaje: 0 → para la red destino, 1 → para el *host destino*, 2 → para el tipo de servicio especificado en el campo *TS* del paquete IP y para la red, y 3 → para el tipo de servicio especificado y para el *host*.

- *Tiempo excedido.* Como mencionamos cuando se abordó el servicio no fiable ofrecido por el protocolo IP, resulta necesario un campo de tiempo de vida en cada datagrama (campo TTL) que evite la transmisión indefinida de los mismos (como resultado de la aparición de bucles en las tablas de encaminamiento). Decrementado en cada dispositivo de encaminamiento en la forma comentada en el Apartado 9.1.1, cuando este campo alcanza el valor 0 se descarta el datagrama correspondiente. Este hecho se comunica al *host* origen con el mensaje ICMP de tipo 11 cuyo formato se muestra en la Figura 9.8(e). Este mensaje también se utiliza para indicar la expiración del tiempo de ensamblado del datagrama a partir de sus fragmentos en el extremo receptor. Ambas situaciones se diferencian a través del campo *código*: 0 para indicar la expiración del campo TTL del datagrama y 1 para el tiempo de ensamblado.
- *Problema de parámetros.* Con el mensaje tipo 12 (Figura 9.8(f)) se indica la ocurrencia de un problema distinto a los anteriormente mencionados. Un valor 0 en el campo *código* indica que la cabecera IP está mal. En cambio, *código* = 1 significa la ausencia de una opción requerida; la señalada mediante el campo *puntero*.

Este tipo de mensaje solo se genera cuando el problema provoca que se descarte el datagrama.

- *Sello de tiempo.* En ocasiones resulta interesante conocer el tiempo total involucrado en el viaje de ida y vuelta entre dos estaciones finales. Es el denominado *tiempo de ida y vuelta* («Round-Trip Time», RTT). La estimación de dicho tiempo se lleva a cabo en base al mensaje ICMP de *sello de tiempo*. El mensaje tipo 13 (y *código*=0) corresponde a la solicitud, especificándose por parte del emisor el instante de tiempo en que esta se origina. Por su parte, el mensaje de respuesta (*tipo*=14, *código*=0) especifica el instante de tiempo en que se genera esta así como el instante en que se recibió la solicitud (Figura 9.8(g)). Los campos *identificador* y *secuencia* de estos mensajes ICMP se utilizan para hacer corresponder solicitudes con respuestas.

Como se ha comentado previamente, algunas implementaciones del comando de usuario *ping* no solo indican la accesibilidad o no del destino, sino también el tiempo del viaje de ida y vuelta a este. Esta información debe ser tomada con cierta reserva debido a la posible desincronización entre los relojes de los dispositivos en la red.

- *Máscara de red.* Como se verá en el próximo apartado, el encaminamiento en Internet precisa del conocimiento de la máscara de red asociada a cada destino. En este sentido existe un mensaje ICMP para la obtención de dicha información.

El mensaje tipo 17 (Figura 9.8(h)) corresponde a la solicitud de la máscara de red, la cual se especificará en el campo *máscara de dirección* del mensaje ICMP de respuesta (*tipo*=18). Como antes, los campos *identificador* y *secuencia* se utilizan para hacer corresponder solicitudes con respuestas.

Además de los anteriores, también hemos de mencionar, por útil, la definición de un nuevo tipo de mensaje ICMP en el RFC 1256. Son los de *descubrimiento de router* “router discovery”. El objetivo perseguido mediante esta nueva tipología de mensajes es evitar la necesidad de conocer a priori la existencia de nodos o *routers* en una red que permitan la interconexión con otras redes y/o dispositivos. Para ello, se definen dos mensajes dentro de esta tipología:

- *Anuncio de router* (*tipo*=9, *código*=0). Mensaje emitido vía *multicast* periódicamente por un *router* en la red para notificar su existencia, de modo que los *hosts* en el entorno tendrán conocimiento de este hecho de modo automático.

En estos mensajes ICMP se indicará, tras la cabecera ya conocida, no solo la dirección de un *router* dado, sino también todos los que este conoce en la misma (sub)red. Además se indica el número máximo de segundos de validez de estas direcciones.

- *Solicitud de router* (*tipo*=10, *código*=0). A través de este mensaje, son los *hosts* quienes preguntan explícitamente para conocer los *routers* disponibles. Frente al anterior, este mensaje ICMP no contiene información adicional de interés a la cabecera del paquete.

9.2.1. ICMPv6

En coherencia con la definición de IPv6, en el RFC 1885, actualizado posteriormente mediante los RFC 2463 y 4443, se define la versión 6 del protocolo ICMP: ICMPv6. Como en ICMPv4, los mensajes ICMPv6 se inicien con una cabecera de 32 bits, de campos *tipo* (8 bits), *código* (8 bits) y *comprobación* (16 bits). Para el cálculo del campo *comprobación* en este caso se considera no solo el paquete ICMP, sino también una seudo-cabecera compuesta por algunos campos de IPv6, entre los que cabe destacar *dirección IP origen*, *dirección IP destino* y *siguiente cabecera*.

Tras la cabecera, como sucede en IPv6, aparecerán o no cabeceras de extensión. En relación a ello, decir que la cabecera ICMPv6 estará identificada con el valor 58 en el campo *siguiente cabecera* en el paquete IPv6.

Los mensajes ICMPv6 se dividen en dos categorías: de error y de información. Los primeros se identifican a través del valor 0 en el bit más significativo de campo *tipo*, esto es, tienen valores en el rango 0-127. Por su parte, los mensajes de información toman valores en el rango 128-255.

Sin entrar en más detalles sobre los distintos tipos de mensajes ICMPv6 existentes, seguidamente se describe brevemente cada uno de ellos (Tabla 9.2):

- *Mensajes de error*:

- *Destino inalcanzable* (*tipo*=1, *código*=0-4). Mensaje originado en un *router* o en la capa IPv6 del nodo origen en respuesta a un paquete que no puede ser entregado a la dirección de destino por algún problema distinto a la aparición de congestión.

Tabla 9.2. Mensajes ICMPv6.

Tipo	Mensaje
1 / Error	Destino inalcanzable
2 / “	Paquete demasiado grande
3 / “	Tiempo excedido
4 / “	Problema de parámetros
128 / Información	Solicitud de eco
129 / “	Respuesta de eco
130-132 / “	Pertenencia a grupo

- *Paquete demasiado grande* (*tipo*=2, *código*=0). Mensaje enviado por un *router* cuando un paquete dado no pueda ser enviado por superar la MTU del enlace en cuestión.
- *Tiempo excedido* (*tipo*=3, *código*=0-1). Mensaje generado por un *router* hacia el origen de un paquete IPv6 que es descartado por haber alcanzado el campo *límite de saltos* el valor 0.
- *Problema de parámetros* (*tipo*=4, *código*=0-2). Mensaje motivado por la aparición de un problema al procesar algún campo del paquete.

— *Mensajes de información:*

- *Solicitud de eco* (*tipo*=128, *código*=0). Como en ICMPv4.
- *Respuesta de eco* (*tipo*=129, *código*=0). Como en ICMPv4.
- *Pertenencia a grupo* (*tipo*=130-132, *código*=0). Mensajes usados para la notificación de información relacionada con la gestión de grupos *multicast* (véase IGMP en el Apartado 9.4.2).

A parte de los mensajes mencionados, existen otros valores de uso reservado tanto para los mensajes de error como los de información. Al igual que en ICMPv4, se incluyen mensajes para el descubrimiento y anuncio de *routers* y de las máscaras. También se incluyen funciones relacionadas con el descubrimiento de vecinos y agentes, de uso en el direccionamiento *anycast*. Por otro lado, en los RFC correspondientes se hace mención expresa a consideraciones adicionales acerca de la seguridad de las transmisiones sobre el protocolo ICMPv6. Se indica así la posible ocurrencia de ataques a este protocolo y el uso de las cabeceras AH y ESP en IP para la provisión de confidencialidad, integridad y autenticación.

9.3. Encaminamiento dinámico en Internet

Dado un datagrama IP, el proceso de encaminamiento seguido para su retransmisión en cada uno de los nodos intermedios de la subred es el siguiente:

1. Extracción de la dirección IP de destino especificada en el datagrama: IP_D .
2. Para cada entrada en la tabla de encaminamiento, consistente en un identificativo de red de destino, IP_N , y la máscara asociada, M_N , además del siguiente nodo en la ruta a seguir hasta dicha red, se procede como sigue:
 - a) Se realiza la operación lógica AND, bit a bit, entre IP_D y la máscara de red M_N , obteniéndose el identificativo de red IP_R . Es decir, $IP_R = (IP_D \text{ AND } M_N)$.
 - b) Si $IP_N = IP_R$, o lo que es lo mismo, si el identificativo de red correspondiente a la entrada coincide con el obtenido tras aplicar la máscara asociada a la dirección IP de destino del paquete, dicho paquete se encaminará como se indica en la tabla, tomando como dirección física (MAC) de destino la del siguiente dispositivo de encaminamiento en la ruta.
 - c) Si, por el contrario, $IP_N \neq IP_R$, se procede a consultar la siguiente entrada en la tabla.
3. Para evitar situaciones de error, la última entrada de la tabla de *routing* suele hacer referencia a una ruta por defecto sobre la que se enviarán aquellos paquetes para los que no se encuentre ninguna coincidencia previa en la tabla.

Obsérvese de lo expuesto que las direcciones IP origen y destino de los datagramas permanecen inalteradas a lo largo de la ruta; en cambio, las direcciones físicas de las tramas sobre los que se encapsulan varían salto a salto. Además, resulta relevante el orden de aparición de las entradas en la tabla de encaminamiento, aplicándose la primera de ellas con prefijo más largo que verifique $IP_N = IP_R$.

Haciendo referencia a los esquemas de establecimiento de las tablas de encaminamiento estudiados en el Capítulo 6 del texto, hemos de mencionar que en Internet se implementa un esquema adaptable, dinámico, de tipo distribuido jerárquico. Esto es, por una parte, los nodos se intercambian sus tablas periódicamente en el tiempo a fin de actualizar y adaptar la información de *routing* y, por otra, la red se divide en regiones de modo que la actualización de las tablas se realiza a dos niveles separados: intra-región e inter-región.

Cada una de las regiones en que se divide Internet se denomina *sistema autónomo* (Figura 9.9), consistente en un conjunto de subredes administradas por una única autoridad de forma que en dicho entorno se puede implementar un algoritmo de encaminamiento independientemente de los considerados en otros sistemas autónomos. Son los conocidos como *protocolos de encaminamiento interiores* o IGP («Interior Gateway Protocol»). Aparte del conocimiento que cada dispositivo de encaminamiento dentro de un sistema autónomo dado debe tener del mismo, deben conocerse las rutas entre sistemas autónomos a fin de establecer una conectividad completa de toda la red. Para ello, en cada sistema autónomo se establece al menos un dispositivo de encaminamiento encargado de encaminar el tráfico entrante/saliente al sistema autónomo. Estos dispositivos son los denominados *dispositivos de encaminamiento exterior o de frontera* (R1 y R2 en la Figura 9.9(b)). El encaminamiento entre sistemas autónomos también se establece siguiendo un esquema distribuido, en este caso mediante un algoritmo común a todos los *routers* frontera como es el algoritmo EGP («Exterior Gateway Protocol») o el más actual BGP («Border Gateway Protocol»).

9.3.1. Protocolos de encaminamiento interiores

En este apartado se presentan dos algoritmos IGP implementados para la actualización dinámica de las tablas de encaminamiento internas a un sistema autónomo dado. Estos son RIPv1 y OSPF, de amplia adopción en Internet. Ambos, como se ha indicado anteriormente, presentan una característica común: son de naturaleza distribuida; es decir, la actualización de las tablas se realiza en base al intercambio periódico de las mismas entre nodos vecinos.

Protocolo de información de encaminamiento (RIP)

La versión 1 del algoritmo RIP («Routing Information Protocol») se especifica en el RFC 1058, encontrándose en los RFC 1388, 1723 y 2453 revisiones sucesivas de la versión 2 del mismo. Aunque con una función propia de la capa de red, encaminamiento, RIP se implementa sobre UDP (ver capítulo

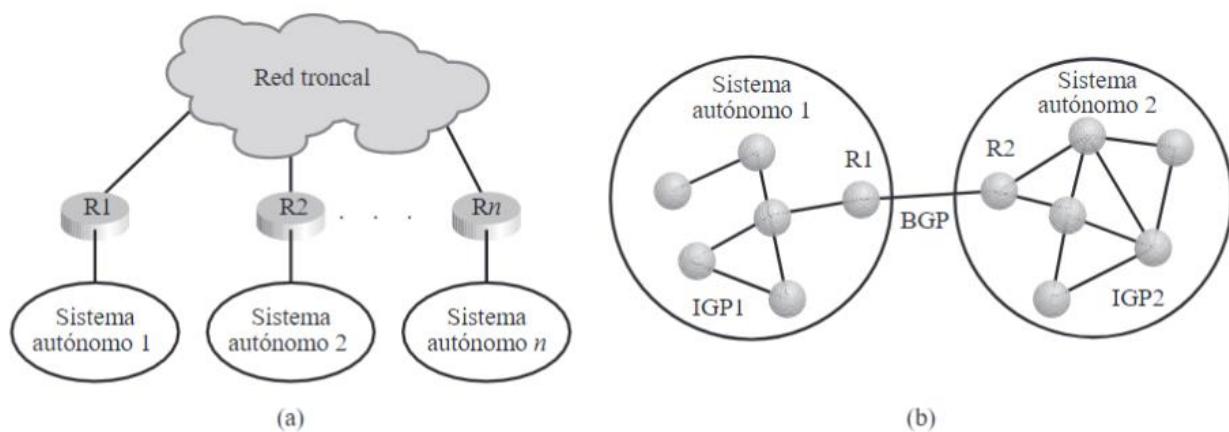


Figura 9.9. Visión conceptual de un sistema autónomo (a) y esquema de protocolos de encaminamiento interiores y exteriores (b).

próximo), es decir, se trata de un protocolo de la capa de aplicación. Ello no limita, sin embargo, su estudio en la capa de red, por cuanto que su funcionalidad pertenece a esta capa.

La principal característica de este protocolo es que se trata de un algoritmo de encaminamiento distribuido basado en *vector-distancia*, donde para cada entrada de la tabla correspondiente a un destino se especifica el número de nodos a atravesar hasta alcanzar el mismo, así como el nodo siguiente en la ruta. La principal ventaja de los algoritmos vector-distancia radica en su simplicidad. Por el contrario, presentan el problema conocido como de *convergencia lenta*. Esto significa que las noticias respecto a un cambio en la topología de la red se propagan lentamente sobre la misma. Dicha convergencia depende del diámetro de la red, definido este como el máximo número de saltos que separan los dos nodos más alejados de la misma.

Otro problema inherente a un algoritmo vector-distancia es el denominado *cuenta al infinito*. Consideremos la topología mostrada en la Figura 9.10. Es claro que la distancia de R1 a un *host A* situado en la red 1 es 0; la de R2 1, a través de R1; y la de R3 2, a través de R2. Si en un momento dado fallase el enlace de R1 a la red 1, la actualización distribuida de las tablas de encaminamiento «permitiría» alcanzar A desde R1 a través de R2 con una distancia total igual a 2. Una actualización posterior haría que R2 estableciese la ruta a A a través de R1 con una distancia igual a 3. De igual modo, en un instante de actualización posterior R1 actualizaría su ruta a A a través de R2 con un número de saltos igual a 4. Procediendo sucesivamente de esta forma, el número de saltos a A desde cualquier nodo se haría infinito, momento en el que se concluiría que la red 1 resulta inalcanzable.

Para intentar evitar el problema mencionado se implementa la técnica conocida como *horizonte dividido* («split horizon» en inglés), en la cual se opta por no propagar la información de encaminamiento correspondiente a un destino dado sobre aquella interfaz por la que atraviesa la ruta hacia dicho destino. Una técnica alternativa a la de horizonte dividido es la denominada «hold down». En este caso se opta por ignorar durante un cierto tiempo (típicamente 60 segundos) todos los mensajes de encaminamiento correspondientes a un destino dado para el que se ha detectado su inaccesibilidad con anterioridad. Otra opción consiste en el esquema «poison reverse», en el cual se comunica mediante difusión la inaccesibilidad de una red una vez detectada esta.

El formato de los mensajes correspondientes a la versión 2 de RIP, RIP-2, es el mostrado en la Figura 9.11. El encabezado de estos mensajes tiene una longitud de 32 bits y está compuesto por los siguientes tres campos:

- *Comando* (8 bits): campo donde se indica el tipo de mensaje RIP. Dos son los posibles valores aceptados: 1 → solicitud de información de encaminamiento y 2 → respuesta de encaminamiento. También se contemplan los valores 3, 4, 5 y 6, encontrándose los dos primeros obsoletos en la actualidad y los dos últimos, de sondeo («poll»), indocumentados.
- *Versión* (8 bits): número de versión de RIP.
- *Dominio* (16 bits): este campo, de valor igual a 0 en la versión 1 de RIP, permite en la versión 2 distinguir entre múltiples procesos de encaminamiento distintos. El valor que toma este campo es 0 por defecto.

Con los mensajes RIP (intercambiados periódicamente en un tiempo máximo de 30 segundos, según el RFC) se indican las distintas redes accesibles desde el nodo emisor y la distancia a la que se

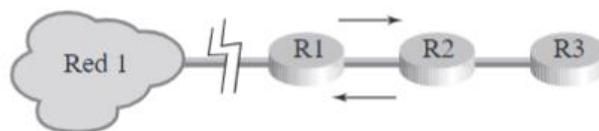


Figura 9.10. Problema de la cuenta al infinito.

0	8	16	31
Comando (1/2)	Versión (2)	Dominio	
Familia dirección 1 (2)		Etiqueta ruta 1	
		Dirección IP red 1	
		Máscara de red 1	
		Siguiente nodo hacia red 1	
		Distancia red 1	
Familia dirección 2 (2)		Etiqueta ruta 2	
		...	

Figura 9.11. Formato del paquete RIP versión 2 para redes IP.

encuentra cada una de ellas. El valor del campo *distancia* se especifica en número de nodos. Un valor 16 significa que la red correspondiente es inalcanzable; es decir, $16 \equiv \infty$. Adicionalmente a la distancia a la que se encuentra accesible, para cada red referenciada en el paquete se debe indicar⁴:

- la *familia de dirección* de que se trata (a valor 2 para direcciones IP),
- una *etiqueta de ruta* indicativa del número de sistema autónomo (para EGP y BGP),
- su *dirección IP* (32 bits en caso de tratarse de una red IP),
- su *máscara de red* (32 bits en caso de red IP) y
- la dirección correspondiente al *siguiente nodo* en la ruta para alcanzar dicha red (32 bits en caso de redes IPv4). Un valor 0 en este campo significa que el encaminamiento debe hacerse a través del nodo emisor del mensaje RIP.

Una capacidad interesante que ofrece la versión 2 de RIP es la posibilidad de llevar a cabo un intercambio de información de encaminamiento autenticado. Para ello, en la primera entrada del paquete RIP, y solo en ella, se especificará un valor hexadecimal 0xFFFF en el campo *familia dirección*. En tal caso, el tipo de autenticación se indicará en el campo *etiqueta ruta* a través del único valor aceptado: 2, significando con ello que los 16 octetos restantes de la entrada (campos *dirección IP red 1* a *distancia red 1* en Figura 9.11) corresponden a una palabra clave de paso o *password*.

Por último, hemos de comentar que, a fin de reducir la carga que supone el intercambio de información de encaminamiento, RIP-2 permite el envío multidestino («multicast») de los mensajes a través de la dirección IP 224.0.0.9, indicativa de «todos los nodos RIP».

Protocolo del primer camino más corto disponible (OSPF)

El protocolo de encaminamiento IGP OSPF («Open Shortest Path First») se basa en el procedimiento del *estado del enlace*; es decir, los nodos no intercambian distancias con sus vecinos, sino el estado de los enlaces a cada uno de ellos. Especificado en el RFC 1131 en su versión 1 y en el RFC 2328 en su versión 2 inicial, OSPF presenta las siguientes características principales:

1. Determina rutas alternativas, permitiendo el balanceo o reparto de carga mediante la distribución del tráfico sobre ellas y el envío de acuerdo a distintos tipos de servicio.
2. Facilita la gestión de las rutas mediante el establecimiento de *áreas* independientes.
3. Permite autenticar el intercambio de información entre dispositivos de encaminamiento a fin de introducir seguridad en la distribución de las rutas.

⁴ Los campos *etiqueta ruta*, *máscara red* y *siguiente nodo hacia red* toman el valor 0 en la versión 1 de RIP.

4. Minimiza la difusión a través de la definición de unos nodos concretos encargados de la actualización de las rutas; son los denominados *nodos designados*.
5. Permite intercambios de rutas mediante envíos *multicast*.
6. Como RIP, se implementa en la capa de aplicación.

En la Figura 9.12 se muestra el formato de los distintos mensajes OSPF en su versión 2, todos comenzando con la cabecera indicada en la subfigura (a). Los campos que forman esta son los siguientes:

- *Versión*: indica la versión del protocolo OSPF, siendo la actual la 2.
- *Tipo*: campo de 8 bits para indicar el tipo de mensaje. Este puede tomar cinco valores distintos:
 - 1 → mensaje *hello*, utilizado para testar la accesibilidad de un nodo vecino,
 - 2 → mensaje de *descripción de la base de datos*, que contiene la información de accesibilidad,
 - 3 → mensaje de *solicitud del estado del enlace*,
 - 4 → mensaje de respuesta al anterior para la *actualización del estado del enlace*,
 - 5 → mensaje de *confirmación del estado del enlace*.
- *Longitud*: campo de 16 bits que indica la longitud en bytes del mensaje OSPF.
- *Dirección IP del nodo origen*: dirección IP del *router* que emite el mensaje OSPF en cuestión.
- *Identificador de área*: número de 32 bits que identifica el área a la que pertenece el paquete.
- *Comprobación*: suma de comprobación del mensaje OSPF.
- *Tipo de autenticación*: campo de 16 bits que indica el tipo de autenticación requerido para el intercambio de mensajes OSPF. Tres son los posibles valores de este campo: 0 → no se precisa autenticación, 1 → uso de una palabra de paso “password” y 2 → autenticación cifrada a través del uso de una función resumen o *hash* (ver Capítulo 12).
- *Autenticación*: 64 bits usados en el proceso de autenticación en caso de que el campo *tipo de autenticación* tome uno de los valores 1 o 2.

Los mensajes OSPF mencionados anteriormente a través del campo *tipo* de la cabecera se expresan como sigue:

- Mensaje *hello* (*tipo* = 1). Este mensaje es intercambiado periódicamente entre dos nodos vecinos para comprobar su accesibilidad. Este paquete consta de los siguientes campos (Figura 9.12(b)):
 - a) *Máscara de red* asociada a la interfaz sobre la que se transmite el mensaje.
 - b) *Intervalo de tiempo*, en segundos, para el intercambio periódico de mensajes *hello*.
 - c) Campo *opciones* a través del que se permite el rechazo de vecinos debido a incompatibilidad de capacidades (*multicast*, inundaciones, etc.)
 - d) *Prioridad* del nodo emisor, necesaria para la elección del nodo designado.
 - e) *Tiempo de expiración* (en segundos) tras el que, si no se recibe contestación, se considera a un nodo vecino «muerto».
 - f) *Nodos designados* principal y de respaldo para la red sobre la que se envía el mensaje.
 - g) *Dirección IP* de cada uno de los nodos vecinos de los que el nodo en cuestión ha recibido mensajes de accesibilidad.
- Mensaje de *descripción de la base de datos* (*tipo*=2). A través de este mensaje, intercambiado entre nodos vecinos cuando se inicia su relación de vecindad, se lleva a cabo la especificación e identificación de los distintos enlaces accesibles desde ambos nodos. Uno de los nodos, *esclavo*, solo enviará este tipo de mensajes en respuesta a solicitudes enviadas por el *maestro*.

Cada mensaje de descripción de la base de datos consta de los siguientes campos (Figura 9.12(c)):

- MTU interfaz*, para indicar el tamaño máximo de paquete IP que se puede transmitir sin fragmentar.
- Campo *opciones*, como se ha comentado para el mensaje *hello*.
- Dado que la base de datos puede ser grande, esta puede partirse en varios paquetes. El primero estará indicado con el campo *I* = 1 y los restantes, salvo el último, con

Cabezera OSPF con <i>tipo</i> = 1		
Máscara de red		
Intervalo de tiempo	Opciones	Prioridad
Tiempo de expiración		
Nodo designado		
Nodo designado de respaldo		
Dirección IP vecino 1		
Dirección IP vecino 2		
...		

(b)

(c)

$M = 1$. Además, cada mensaje estará numerado secuencialmente a través del campo *secuencia base de datos*.

- d) El campo de 1 bit S indica si el nodo emisor es el maestro ($S = 1$) o el esclavo ($S = 0$).
- e) El resto del paquete consta de una serie de piezas descriptoras de la base de datos correspondiente al estado de los enlaces. Cada una de estas piezas es lo que se llama *cabecera LSA* (HLSA, «Header Link Status Advertisement») y está compuesta por los campos indicados en la Figura 9.12(d):

- *Edad* del enlace, en segundos, desde que este fue establecido.
- *Opciones*, como en el apartado b) anterior
- *Tipo* de enlace, el cual puede tomar cinco valores: 1 → de dispositivo de encaminamiento, 2 → de red, 3 → ruta a red, 4 → ruta a nodo frontera y 5 → a destino externo.
- *Identificador* que describe la porción de Internet especificada por el enlace. Los posibles valores de este campo dependen del tipo de enlace de que se trate.
- *Dirección del nodo notificador* que indicó la existencia de este enlace.
- *Número de secuencia* del enlace para su ordenación e identificación, permitiendo, por ejemplo, la detección de LSA duplicados.
- Suma de *comprobación* de los campos del LSA (ver mensaje tipo 4), incluyendo la cabecera excepto el campo *edad* del enlace.
- *Longitud* en bytes del LSA (ver mensaje tipo 4) incluida la cabecera.

— Mensaje de *solicitud del estado del enlace* ($tipo = 3$). Mostrado en la Figura 9.13(a), este tipo de mensajes OSPF se envía por parte de un nodo para requerir información acerca de un conjunto

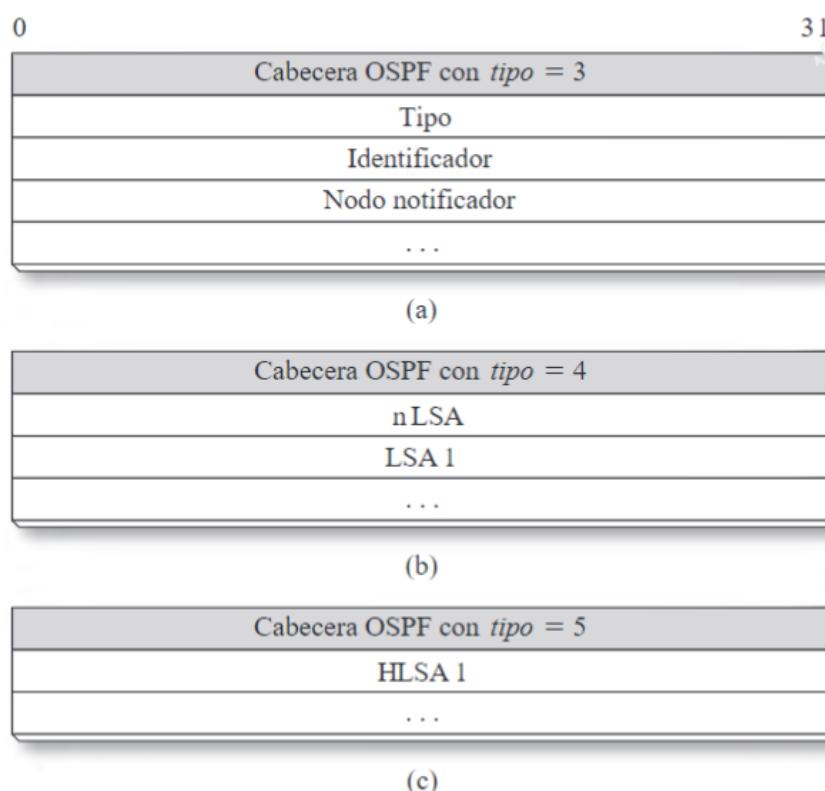


Figura 9.13. Mensajes OSPF solicitud del estado del enlace (a), actualización del estado del enlace (b) y confirmación del estado del enlace (c).

- específico de enlaces (LSA). De cada uno de ellos se debe indicar su *tipo*, su *identificador* y el *nodo que lo notificó*.
- Mensaje de *actualización del estado del enlace* (*tipo* = 4). Enviado en respuesta al anterior, este mensaje indica el número de anuncios de estado contenidos (*nLSA*) y la lista de los mismos (Figura 9.13(b)). Cada uno de los LSA está compuesto por la cabecera HLSA, ya comentada y mostrada en la Figura 9.12(d), además de datos específicos dependientes del tipo de enlace de que se trate. Entre otros datos especificados (identificador, máscara, etc.), los más relevantes son: tipo de servicio y métrica o coste asociado al enlace (para más detalles, ver RFC 2328).
 - Mensaje de *confirmación del estado del enlace*. Paquete utilizado para confirmar la recepción de un mensaje de actualización del estado del enlace. Como se indica en la Figura 9.13(c), este tipo de mensajes está compuesto por la cabecera OSPF con el campo *tipo* a valor 5 y una HLSA para cada LSA recibida en el paquete de actualización tipo 4.

Protocolos IGRP y EIGRP

Además de los protocolos de *routing* descritos, también merecen especial mención los propietarios de Cisco IGRP y EIGRP. El primero de ellos, *Interior Gateway Routing Protocol*, es, como RIP, un protocolo vector-distancia y fue desarrollado para solventar dos limitaciones de este para grandes redes: número máximo de saltos igual a 15 y uso de una sola métrica. En el caso de IGRP, el número máximo de saltos es 255 (100 por defecto) y se combinan varias métricas (ancho de banda, retardo, carga, MTU y fiabilidad) en una sola a través de una fórmula donde se hace uso de pesos para cada una de ellas.

Finalmente, comentar que la actualización de las tablas en IGRP se realiza por defecto cada 90 segundos y que se trata de un protocolo conforme a clase (*classful*), lo que significa que no se gestionan máscaras de subred. Esto es, si se trata de una red IP Clase A, los bits de red y *host* son los fijos ya conocidos: 7 y 24, respectivamente; 14 y 16 para Clase B; y 21 y 8 para Clase C.

Con posterioridad a IGRP, Cisco desarrolló el protocolo EIGRP («Enhanced Interior Gateway Routing Protocol»), el cual ha sustituido completamente al anterior. A diferencia de los protocolos previos, incluido IGRP, EIGRP se implementa directamente en la capa de red (es decir, sobre IP) y no en la de aplicación.

Las características principales de EIGRP son:

- Es conforme a CIDR y permite enmascaramiento variable (*classless*).
- Soporta balanceo de carga, en base a evitar bucles parciales y el uso de un algoritmo que calcula la cantidad de tráfico a enviar por cada camino.
- Permite usar diferentes *passwords* a lo largo del tiempo.
- Soporta autenticación MD5 entre los *routers*.
- El envío de la información de *routing* se refiere más bien a cambios en la topología y no tanto a tablas de encaminamiento completas.
- Realiza procesos de encaminamiento separados por protocolo (IP, IPv6, IPX, Apple Talk, etc.).

EIGRP implementa el algoritmo DUAL («Diffusing Update ALgorithm») para mejorar el encaminamiento en base a la eliminación de bucles en el entorno. Tres son las tablas usadas para el cálculo de las rutas:

- *Tabla de vecinos*: contiene información del conjunto de *routers* directamente conectados, disponiéndose una tabla distinta para cada protocolo posible. Cada entrada corresponde a un vecino, con la descripción de la interfaz de red y dirección, además de un contador para el intercambio periódico de paquetes *hello* para testar la accesibilidad del nodo en el tiempo.

- *Tabla de topología*: contiene una lista de las posibles redes de destino, junto con el coste asociado a la ruta y un «nodo sucesor» y un «nodo posible sucesor» para alcanzarla.
- *Tabla de rutas*: almacena las rutas reales a todos los destinos.

Como se ha indicado anteriormente, a diferencia de otros protocolos como RIP, EIGRP no se basa en el intercambio periódico de las tablas entre nodos vecinos, sino que se comunican los cambios habidos a lo largo del tiempo a partir de la definición de relaciones de vecindad.

9.3.2. Protocolo exterior BGP

A diferencia de como sucede en el encaminamiento interno, en el que cada sistema autónomo puede considerar un IGP independiente del resto (RIP, OSPF, EIGRP, etc.), el establecimiento de las rutas entre sistemas autónomos precisa de un algoritmo común implementado sobre los dispositivos de encaminamiento exteriores o frontera («border» en inglés). Inicialmente se especificó en el RFC 823 el protocolo GGP («Gateway-to-Gateway Protocol»), implementándose posteriormente el protocolo EGP («Exterior Gateway Protocol»), detallado en el RFC 904. A pesar del amplio uso que de este último protocolo se ha hecho, los problemas que a continuación se enuncian (ver RFC 1009) motivaron su sustitución por el *protocolo de pasarela frontera* (BGP, «Border Gateway Protocol»):

1. La conectividad global falla si un nodo frontera falla.
2. EGP solo establece una ruta para alcanzar cada sistema autónomo, no permitiendo balanceo de carga.
3. No interpreta ninguna de las métricas de distancia internas que aparecen en los mensajes de actualización de las tablas.

La última versión del protocolo BGP es la 4 (BGP-4), especificada en el RFC 1771 (actualizado por el RFC 4271). De forma análoga a EGP, el protocolo BGP presenta los siguientes tipos de mensajes:

- *Adquisición de vecino*, a fin de establecer los nodos frontera entre los que se efectuará el intercambio de la información de encaminamiento.
- *Accesibilidad de vecino*, con objeto de testar la alcanzabilidad de los mismos.
- *Información de encaminamiento*, para proceder a la actualización de las tablas.

Cada uno de los mensajes BGP comienza con los siguientes tres campos de 19 octetos (Figura 9.14(a)):

- *Marcador* (16 bytes): campo de autenticación que permite al destino verificar la identidad del emisor del mensaje.
- *Longitud* (2 bytes): indica el número de octetos que componen el mensaje BGP.
- *Tipo* (1 octeto): campo que indica el tipo de mensaje BGP de que se trata: 1 → *Apertura* (*Open*), 2 → *Actualización* (*Update*), 3 → *Notificación* (*Notification*) y 4 → *Accesibilidad* (*Keepalive*).

A continuación se comentan los cuatro tipos de mensajes BGP mencionados, según el orden seguido en su utilización:

- *Apertura* (*Open*) (*tipo*=1). A través de este mensaje (Figura 9.14(b)) se solicita una relación de vecindad con un nodo frontera adyacente perteneciente a otro sistema autónomo. Con una longitud mínima de 29 octetos, en este mensaje se especifica:
 - a) Un campo de *versión* del protocolo BGP para la correcta interpretación de los campos que lo forman. La versión actual es la 4.

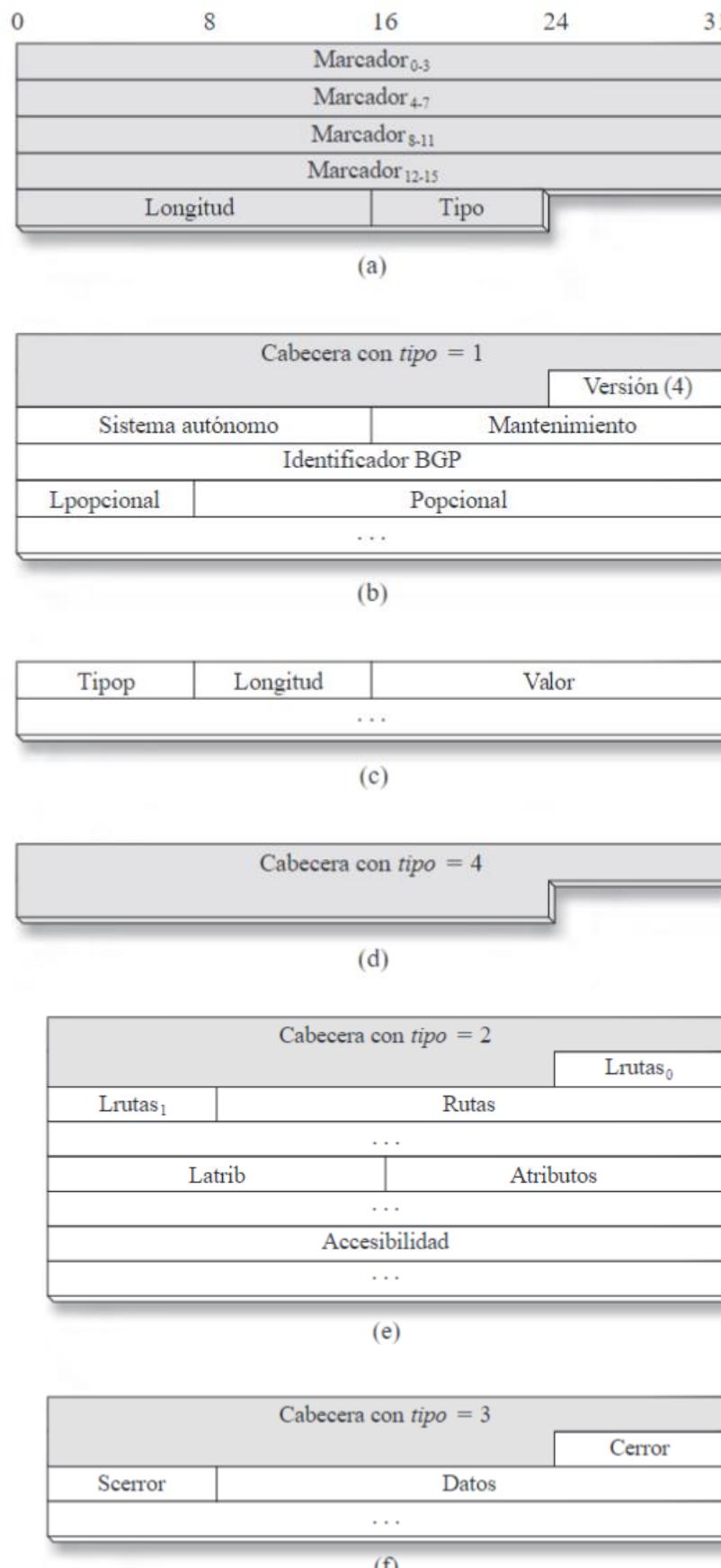


Figura 9.14. Cabecera BGP-4 (a) y mensajes Open (b), Keepalive (d), Update (e), y Notification (f). (c) corresponde al campo *popcional* en (b).

- b) El número de *sistema autónomo* del que forma parte el nodo emisor del mensaje.
- c) El tiempo de *mantenimiento*, referente al número máximo de segundos a considerar para el intercambio de mensajes de *Accesibilidad* y/o de *Actualización* entre vecinos.
- d) La dirección IP del nodo emisor (campo *identificador BGP*).
- e) Un campo de parámetros adicionales opcionales (*popcional*) cuya longitud se indica en el campo de 1 octeto de longitud *lpopcional* y cuyo formato es el que se especifica en la Figura 9.14(c).

Por el momento solo se considera un parámetro opcional (*tipop* = 1), el cual sirve para la autenticación del emisor y del receptor. El campo *valor* consta en este caso de 1 octeto para indicar el tipo de autenticación y de un número variable de bytes (*longitud-3*) correspondientes a los datos involucrados en dicha autenticación.

Si el nodo receptor de un mensaje *Open* acepta la relación de vecindad, deberá responder al emisor con un mensaje *Keepalive*.

- Accesibilidad (*Keepalive*). El mensaje *Keepalive* (Figura 9.14(d)) consiste solo en la cabecera BGP con el campo *tipo* a valor 4. Este mensaje hace las veces de mensaje eco y, como tal, sirve para testar la accesibilidad de los nodos vecinos. Si transcurre un tiempo igual al de mantenimiento especificado en el mensaje *Open* sin recibir un mensaje *Keepalive*, se concluye que el vecino identificado con anterioridad ha «muerto».

Además de para testar la accesibilidad, este mensaje BGP tipo 4 se utiliza como respuesta a un mensaje *Open*.

- Actualización (*Update*). A través de este mensaje BGP-4 tipo 2 (Figura 9.14(e)) se produce la actualización de las tablas de encaminamiento entre nodos pertenecientes a sistemas autónomos distintos. El mensaje puede incluir uno o los dos tipos de información siguientes: (a) una ruta particular a considerar a través de un conjunto de redes y (b) una lista de rutas a eliminar.

La primera situación precisa de los siguientes tres campos:

- a) *Latrib*: número de octetos de que consta el campo *atributos*.
- b) *Atributos*: lista de atributos aplicados a la ruta. Estos constan de 3 octetos: *tipo de atributo*, *longitud de atributo* y *valor de atributo*. Los tipos de atributos definidos son los siguientes: 1 → *origen* (información generada por un IGP o por BGP), 2 → *ruta_AS* (lista de sistemas autónomos atravesados por la ruta), 3 → *siguiente salto* (dirección IP del siguiente nodo frontera para alcanzar los destinos indicados), 4 → *multisalida* (indica varios puntos de salida hacia un sistema autónomo), 5 → *preferencia local* (indica una prioridad para una ruta interna al sistema autónomo), 6 → *agredado atómico* y 7 → *agregador* (utilizados para componer rutas con partes comunes).
- c) *Accesibilidad*: lista de direcciones IP correspondientes a los destinos alcanzables. Cada destino consta de dos campos: *longitud* y *prefijo*, siendo el primero un octeto para indicar la longitud del segundo.

Aunque el mensaje no la incluye, la longitud total del campo *accesibilidad* puede calcularse como: *longitud Update - 23 - latrib - lrutas*, donde el campo *lrutas* es como se especifica a continuación.

El segundo tipo de información que puede aparecer en un mensaje BGP *Update* es para la eliminación de rutas e involucra dos campos del paquete: *rutas*, de longitud variable y relativo a una lista de direcciones IP previamente anunciadas por este nodo y notificadas ahora para su borrado, y *lrutas*, de 2 octetos y cuyo fin es especificar la longitud del campo anteriormente mencionado. El formato del campo *rutas* es el mismo que el indicado anteriormente para el campo *accesibilidad*.

— *Notificación (Notification)*. Las posibles situaciones de error se indican con el mensaje BGP *Notification*. Como se muestra en la Figura 9.14(f), los campos que componen este mensaje, con campo *tipo* = 3 en la cabecera, son los siguientes:

- Cerror*: campo de 1 octeto para indicar el tipo de error de que se trata. Los valores que este campo puede tomar son los siguientes: 1 → error en la cabecera del mensaje, 2 → error en el mensaje *Open*, 3 → error en el mensaje *Update*, 4 → expiración del tiempo de mantenimiento, 5 → error en la máquina de estados finitos correspondiente al procedimiento, 6 → cese.
- Scerror*: campo de 1 byte para concretar aún más el tipo de error. En la Tabla 9.3 se indican los valores de este campo tal como se especifican en el RFC 1771.
- Datos*: campo de longitud variable donde se explica la razón del error para su posible lectura por parte de un humano.

La longitud mínima del mensaje *Notificación*, incluida la cabecera, es de 21 octetos, caso en el cual se considera el campo *datos* de longitud nula.

Un último comentario acerca de BGP hace referencia al hecho de que estos mensajes no se encapsulan sobre datagramas IP, sino sobre segmentos TCP. En este sentido, BGP, como sucede con RIP y OSPF, es un protocolo propio de la capa de aplicación y no de la de red.

Tabla 9.3. Mensajes BGP de Notificación.

Campo <i>cerror</i>	Campo <i>scerror</i>	Descripción
1	1	<i>Connection Not Synchronized</i>
	2	<i>Bad Message Length</i>
	3	<i>Bad Message Type</i>
2	1	<i>Unsupported Version Number</i>
	2	<i>Bad Peer AS</i>
	3	<i>Bad BGP Identifier</i>
	4	<i>Unsupported Optional Parameter</i>
	5	<i>Authentication Failure</i>
	6	<i>Unacceptable Hold Time</i>
3	1	<i>Malformed Attribute List</i>
	2	<i>Unrecognized Well-known Attribute</i>
	3	<i>Missing Well-known Attribute</i>
	4	<i>Attribute Flags Error</i>
	5	<i>Attribute Length Error</i>
	6	<i>Invalid ORIGIN Attribute</i>
	7	<i>AS Routing Loop</i>
	8	<i>Invalid NEXT_HOP Attribute</i>
	9	<i>Optional Attribute Error</i>
	10	<i>Invalid Network Field</i>
	11	<i>Malformed AS_PATH</i>

9.4. Encaminamiento multidestino en Internet

Un aspecto relevante a considerar en Internet, especialmente con la adopción de nuevos servicios del tipo multi-conferencia o vídeo *streaming*, es el de las transmisiones multidestino o *multicast*. Es decir, envíos generados por un único origen pero simultáneamente destinados a varios receptores. Como ya se comentó en el Apartado 6.3.4, es patente la conveniencia de abordar la provisión de este tipo de servicios con comunicaciones distintas a las *unicast*. A modo de ejemplo sencillo, en la Figura 9.15 se muestra una situación en la que un emisor envía datos correspondientes a un mismo servicio a 3 destinos. Es evidente que, si bien es posible hacerlo, una transmisión *unicast* implicaría multiplicar innecesariamente el volumen de recursos (ancho de banda y *buffer* en los *routers*) involucrados en la provisión del servicio. El carácter «innecesario» viene derivado del hecho de que, como se observa en la figura, parte de los paquetes pueden ser «agrupados» para optimizar el uso de los recursos. Ello se consigue transmitiendo un solo paquete *multicast* en la parte de la ruta que sea común desde el origen a los posibles destinos y replicando el paquete desde el nodo a partir del cual las rutas a los posibles destinos diverjan.

Sobre este tipo de transmisiones son tres los aspectos principales a analizar. Por una parte, y frente al empleo ya conocido de direcciones IP *unicast*, cómo se puede identificar unívocamente un grupo de destinos. Por otro lado, cómo se gestiona la creación y el mantenimiento de los grupos *multicast*, esto es, cómo se dan de alta y de baja puntos finales (destinos) de un cierto grupo. Por último, pero no menos importante, cómo se establecen las rutas *multicast* hacia los diferentes destinos.

Cada una de estas cuestiones se desarrolla en los siguientes apartados.

9.4.1. Direccionamiento IP *multicast*

La transmisión multidestino en IP es el equivalente en Internet del *multicast* y difusión hardware en las redes LAN (véase Apartado 5.2.2). En este sentido, la transmisión que da nombre al presente apartado permite el envío de datagramas IP a un conjunto más o menos amplio de *hosts* que constituyen un grupo; grupo que puede estar compuesto por miembros pertenecientes a redes físicas distintas. Como se estudió en el Apartado 8.4, las direcciones Clase D se utilizan para transmisiones multidestino IP

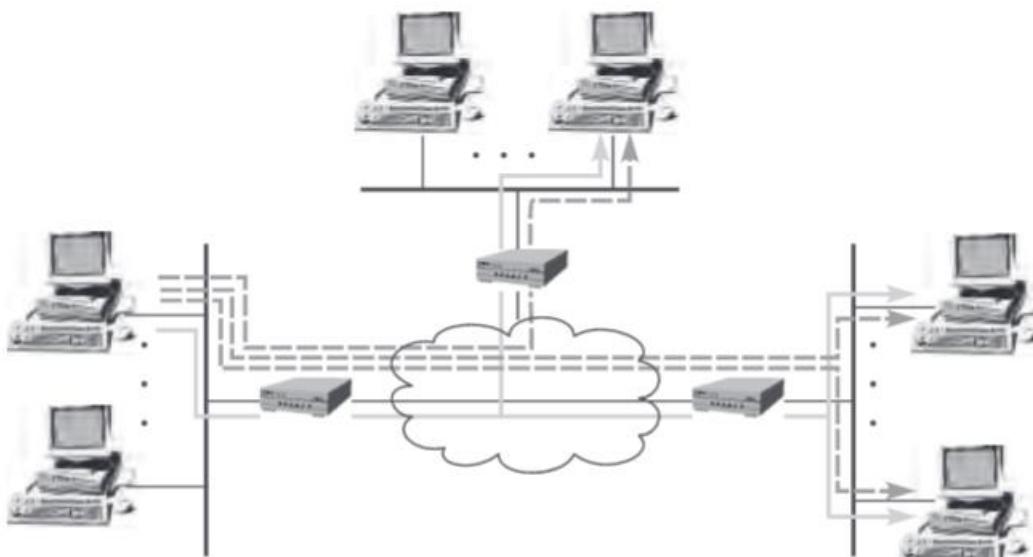


Figura 9.15. Ejemplo de transmisiones *multicast* (línea continua) frente a *unicast* (línea discontinua)

0	1	2	3	4		31
1	1	1	0		Identificador de grupo	

Figura 9.16. Direcciones IP Clase D multidestino.

(Figura 9.16), únicamente como direcciones destino y nunca como origen. El rango, por tanto, para estas direcciones es desde 224.0.0.0 a 239.255.255.255.

Es claro que, como se vio en el Capítulo 8 anterior, debe existir una correspondencia entre las direcciones IP multidestino y las direcciones hardware subyacentes, las cuales serán en este caso también multidestino. La conversión de direcciones multidestino IP a direcciones multidestino Ethernet se establece de la siguiente forma: *sustituir los correspondientes 23 bits menos significativos de la dirección multidestino Ethernet 01:00:5E:00:00:00 por los 23 bits menos significativos de la dirección IP*. Así, la dirección multidestino IP 224.10.8.21 se corresponderá con la dirección Ethernet multidestino 01:00:5E:10:08:21. Según se deduce de lo establecido, la no consideración de los 5 bits de dirección multidestino IP de posiciones 4 a 8 hace que la correspondencia entre ambos espacios de direcciones no sea unívoca; es decir, varios grupos *multicast* IP pueden tener la misma dirección *multicast* Ethernet. Así por ejemplo, los grupos IP 224.10.8.21, 233.10.8.21 y 224.138.8.21 se corresponden con el mismo grupo Ethernet 01:00:5E:10:08:21. De todo ello se concluye que el software IP de una estación final debe filtrar la información recibida a fin de rechazar aquellos datagramas no solicitados.

Una vez aclarado el formato para las direcciones IP *multicast* y cómo estas se asocian a direcciones *multicast* físicas, una cuestión importante adicional es cómo se lleva a cabo la asignación de una dirección IP *multicast* dada a un conjunto de usuarios o grupo. En este punto, sin embargo, hay que matizar que una dirección *multicast* lo que realmente identifica es una fuente de datos, de manera que los destinatarios interesados en recibir estos deben indicarlo explícitamente (esto es, suscribirse al grupo) a sus respectivos nodos de acceso para que procedan a realizar el encaminamiento de la información correspondiente.

Aclarado ello, la asignación de una dirección *multicast* a una fuente dada se puede realizar de tres formas posibles:

- *Asignación estática*. El IANA («Internet Assigned Numbers Authority») tiene reservadas distintas direcciones IP *multicast* (<http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml>). Así, las direcciones 224.0.0.1 y 224.0.0.2 significan, respectivamente, envío a todos los *hosts* y a todos los nodos de una red; mientras que 224.0.0.9 significa «todos los *routers* RIP-2».
- *Relativa a dominio*. En el RFC 2365 se describe que el rango de direcciones 239.0.0.0 a 239.255.255.255 se asigna localmente a nivel de dominio, de manera que estas direcciones no pueden ser utilizadas entre dominios separados.
- *Asignación dinámica*. Para permitir un uso adecuado y flexible, dinámico según necesidad, del espacio de direcciones *multicast*, existen procedimientos automáticos de asignación bajo demanda. Estos se fundamentan en la arquitectura MALLOC («Multicast Address aLLOCation architecture») especificada en el RFC 2908 (actualizado a su vez por el RFC 6308).

Como se indica en la Figura 9.17, en la arquitectura MALLOC se definen dos niveles de actuación: inter-dominio e intra-dominio. El primero permite la asignación de rangos de direcciones para cada dominio, mientras que el segundo se refiere a una coordinación intra-dominio para la asignación final de direcciones *multicast*. Asimismo, para posibilitar una mayor escalabilidad de la solución se definen clientes y servidores de asignación. Los primeros solicitan una

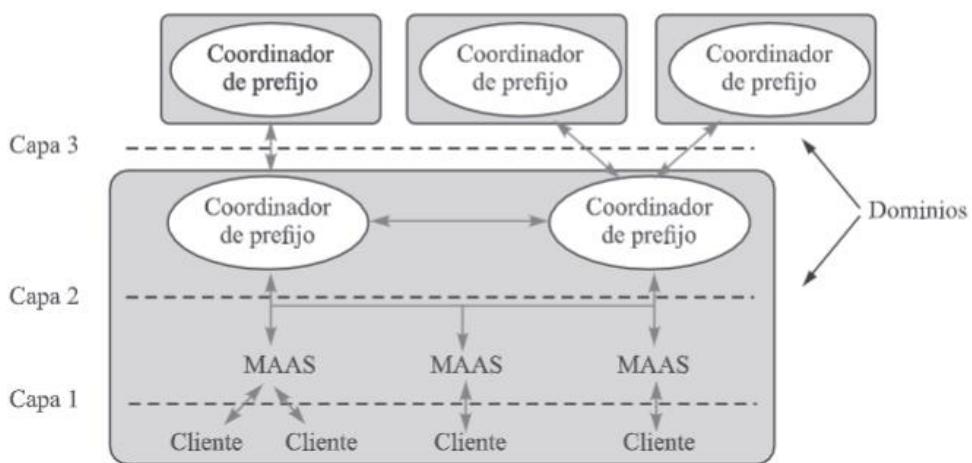


Figura 9.17. Arquitectura MALLOC.

dirección *multicast* como usuarios y los segundos, denominados MAAS («Multicast Address Allocation Server»), asignan y gestionan estas.

Si bien son varios los protocolos existentes para llevar a cabo la asignación y gestión dinámica de direcciones *multicast*, dos de los más usados son MASC, tanto para inter-dominio como para coordinación intra-dominio, y MADCAP, para las interacciones cliente-MAAS.

Protocolo de asignación de direcciones *multicast* MASC

El protocolo MASC («Multicast Address-Set Claim protocol»), definido en el RFC 2909, está pensado tanto para relaciones intra-dominio como inter-dominio. Así, un nodo puede realizar las siguientes funciones y relaciones respecto de otro remoto (Figura 9.18):

- *Paritario intra-dominio*, como los nodos P4a y P4b en la Figura 9.18, diciéndose *P4a internal_peer P4b*.
- *Hijo*, como el caso del nodo C6a respecto del P4a en la Figura 9.18. En este caso se designan *C6a child P4a*.
- *Padre*, como el nodo T2a respecto del P4a en la Figura 9.18. En este caso se dice *T2a parent P4a*.

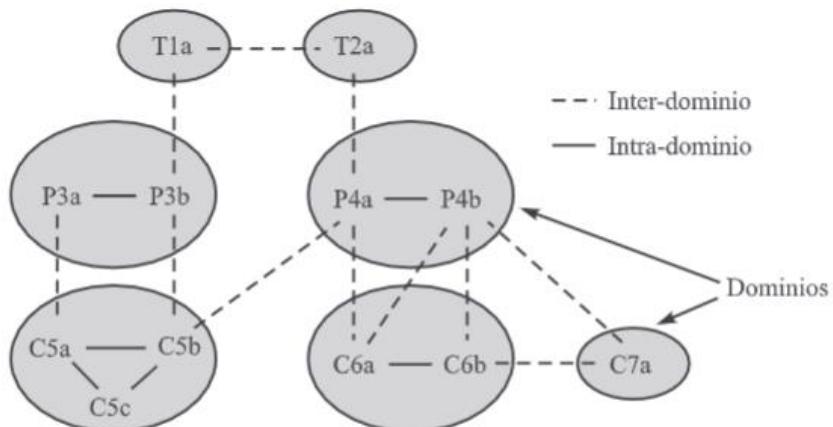


Figura 9.18. Ejemplo de topología jerárquica considerada en MASC.

— *Hermano o paritario inter-domino*, como T1a respecto de T2a en la Figura 9.18. En este caso se dice *T1a sibling T2a*.

A partir de lo anterior, los mensajes generados por un nodo dado se propagan hacia su padre, sus hermanos con el mismo padre y sus hijos. Los mensajes MASC están precedidos por la cabecera cuyo formato se indica en la Figura 9.19(a). De los campos que la componen es de destacar el campo *tipo*, a través del cual se indica el tipo concreto de mensaje MASC de que se trata⁵:

- *Open* (*tipo=0*). Primer mensaje intercambiado para establecer una conexión TCP entre los nodos. De entre los campos que componen este mensaje (Figura 9.19(b)), son de destacar dos:
 - *Rol*: papel del nodo emisor respecto del remoto (00→*internal_peer*; 01→*child*; 10→*sibling*; 11→*parent*).
 - *Tiempo*: tiempo de intercambio de mensajes *Keepalive* para testar la accesibilidad (aproximadamente 4 minutos).

0	16	24	31
Longitud		Tipo	Reservado

(a)

0	8	9	14	16	31
Longitud	0	Familia	Rol	Tiempo	
Identificador del dominio del emisor...					
Identificador del nodo MASC emisor...					
Identificador del dominio del parente...					
Parámetros opcionales...					

(b)

0	16	24	31		
Longitud α		Tipo α	Reservado α		
Reservado1	D	Familia	Rol	Reservado2	
Sello del tiempo de la solicitud					
Tiempo de vida de la solicitud					
Tiempo de tenencia (en <i>cache</i>) de la solicitud					
Identificador de dominio origen...					
Identificador de nodo origen...					
Dirección asociada al prefijo...					
Máscara...					
Parámetros opcionales...					

(c)

0	1	8	16	31
Código		Subcódigo		Datos

(d)

Figura 9.19. Cabecera de los mensajes MASC (a) y mensajes Open (b), Update (c) y Notification (d).

⁵ Nótense las similitudes evidentes existentes con el protocolo BGP-4.

- *Update* (*tipo*=1). Mensaje de intercambio de información en base a atributos, refiriéndose un atributo al conjunto de campos no sombreados en el mensaje de la Figura 9.19(c). De todos los campos existentes en un atributo, es de destacar el campo *tipoa* desde el punto de vista de la funcionalidad deseada para MASC; el cual puede tomar los siguientes valores: 0→PREFIX_IN_USE, 1→CLAIM_DENIED, 2→CLAIM_TO_EXPAND, 3→NEW_CLAIM, 4→PREFIX_MANAGED y 5→WITHDRAW.
- *Notification* (*tipo*=2). Mensaje definido para la notificación de posibles errores. El campo *código* se refiere al tipo de error concreto, el cual toma valores entre 1 y 7 (1→error en la cabecera, 2→error en mensaje *Open*, 3→error en mensaje *Update*, 4→tiempo de tenencia expirado, 5→error en la máquina de estados finitos, 6→error en el mensaje *Notification*, 7→cese), y el campo *subcódigo* hace mención a subtipos dentro de un tipo (por ejemplo, para el tipo 2 existen 13 subtipos y hasta 29 para el 3).

Adicionalmente a la terminología numérica de los errores a través de los campos mencionados, a través del campo *datos* se explicitan estos mediante texto.

- *Keepalive* (*tipo*=4). Mensaje de respuesta a uno *Open* y enviado periódicamente para comprobar la accesibilidad del vecino. Consta solo de la cabecera MASC ya conocida.

El procedimiento operacional del protocolo MASC es fácil de describir a partir de los mensajes antes referidos:

1. Primero se establecen las conexiones TCP oportunas y se fijan las relaciones de vecindad mediante los mensajes *Open* y *Keepalive*.
2. Con los mensajes *Update* se eligen las direcciones según el siguiente procedimiento:
 - a) Cuando un nodo necesita un espacio de direcciones, lo decide localmente (a partir del conocimiento que tiene de las direcciones de su padre, sus hermanos o la suya propia; ver RFC) y se lo indica a uno de sus padres, a sus hermanos con el mismo parentesco y a todos los paritarios intra-dominio.
 - b) Si se recibe una solicitud «colisionante», gana aquella que resulta de la comparación PREFIX_IN_USE > CLAIM_DENIED > CLAIM_TO_EXPAND > NEW_CLAIM. Si siguen coincidiendo, gana aquella con menor *sello de tiempo*, y si aún coinciden, el de menor *identificador de nodo origen*.
 - c) Finalmente, se fija el espacio de direcciones elegido y se notifica mediante un mensaje *Update* de tipo PREFIX_IN_USE.

Tras todo este proceso, queda fijado el rango de direcciones *multicast* inter-dominio o intra-dominio a utilizar en cada caso.

Protocolo de asignación de direcciones *multicast* MADCAP

La asignación concreta de una dirección IP a un usuario fuente de información *multicast* se realiza mediante protocolos como MADCAP («Multicast Address Dynamic Client Allocation Protocol»), el cual se define en el RFC 2730.

MADCAP se implementa sobre UDP (puerto 2535) y los mensajes tienen el formato especificado en la Figura 9.20. De los campos dispuestos, el relevante para la provisión de la funcionalidad pretendida es *tipo*, el cual indica el tipo de mensaje concreto de que se trata:

- *Mensajes cliente→servidor*:
 - DISCOVER (*tipo*=1): descubrimiento de servidores MADCAP para solicitudes.
 - GETINFO (*tipo*=8): obtención de parámetros de configuración (p.e., lista de dominios).

0	8	16	31
Versión	Tipo	Familia	
	xid		
	Opciones...		

Figura 9.20. Paquete MADCAP.

- REQUEST (*tipo=3*): solicitud de dirección *multicast*.
- RENEW (*tipo=4*): renovación de dirección *multicast* tras la expiración del periodo de validez de la misma.
- RELEASE (*tipo=7*): liberación de dirección *multicast* para su reutilización por parte de otros clientes.

— *Mensajes servidor→cliente:*

- ACK (*tipo=5*): confirmación positiva a mensajes del cliente.
- NAK (*tipo=6*): confirmación negativa a mensajes del cliente.
- OFFER (*tipo=2*): respuesta a DISCOVER indicando una dirección.

A partir de lo anterior, el funcionamiento básico del protocolo MADCAP es como sigue:

- Solicitado por parte de un cliente el conocimiento de un conjunto de direcciones *multicast* disponibles mediante el mensaje DISCOVER, el servidor informará de ello con un mensaje OFFER.
- Tras ello, el cliente solicitará una de las direcciones informadas mediante un mensaje REQUEST y, si todo va bien, el servidor la concederá mediante un mensaje ACK.
- Tras el uso de una dirección *multicast*, un cliente puede liberarla enviando un mensaje RELEASE. En todo caso, es de significar que la concesión de direcciones tiene un límite de tiempo, transcurrido el cual el cliente debe renovarla mediante un mensaje RENEW si desea continuar utilizándola.

Todos estos mensajes serán oportunamente confirmados por el servidor mediante ACK.

9.4.2. Gestión de grupos: IGMP

Tras conocer distintos esquemas de asignación de direcciones *multicast*, en este apartado se aborda el estudio de la gestión en cuanto a la incorporación y abandono de usuarios de un grupo *multicast*; esto es, de la contabilización de los usuarios finales interesados o no en recibir la información *multicast* relativa a un grupo dado. Antes de ello, sin embargo, es importante mencionar que existen distintos procedimientos para publicitar una dirección *multicast* asignada a una fuente de datos.

Los medios usuales para ello son dos: indicación explícita mediante servicios del tipo correo electrónico o página web, o mediante protocolos específicos diseñados al efecto como es el caso de SAP («Session Announcement Protocol»; RFC 2974). Este último caso se refiere a la difusión sobre la red de cierta información (como es la dirección de grupo *multicast*) necesaria para la provisión de un servicio dado.

Una vez que los usuarios finales conocen la dirección *multicast*, ya sí se está en disposición de llevar a cabo la gestión de los correspondientes grupos como sigue.

Para la gestión de los grupos se especificó el protocolo Internet IGMP («Internet Group Management Protocol»), del cual hay que señalar en primer lugar que solo se implementa en los *hosts* y *routers* hoja o finales; no así en los nodos intermedios de la red. Sobre esta cuestión incidiremos en el Apartado 9.4.3 más adelante.

0	8	16	31
Tipo	Tiempo	Comprobación	
			Grupo

Figura 9.21. Mensaje IGMP versión 2.

De modo análogo a ICMP, IGMP es parte integrante de la capa de red IP y se encapsula sobre dicho protocolo (campo *protocolo* del datagrama IP a valor 2). Especificado en su versión 2 en el RFC 2236 (versión 1 en RFC 1112 y versión 3 en RFC 3376), los mensajes IGMPv2 tienen el formato mostrado en la Figura 9.21. El significado de los distintos campos es el siguiente:

- *Tipo*: campo de 8 bits para especificar el tipo de mensaje IGMP de que se trata. Existen los siguientes cuatro tipos:
 - Consulta de pertenencia a grupo* («Membership Query» o simplemente «Query»): con valor 11 en hexadecimal, existen dos subtipos de este mensaje diferenciados por el campo de 32 bits *grupo* (ver más adelante). El primero, llamado «General Query», sirve para preguntar acerca de la existencia de *hosts* en cualquiera de los grupos establecidos. El segundo subtipo, «Group-Specific Query», se utiliza para consultar sobre la existencia de *hosts* en un grupo *multicast* concreto.
 - Informe de pertenencia* («Membership Report» o «Report»): mensaje de tipo 16 en hexadecimal, utilizado como respuesta a una consulta a través de «Query».
 - Abandono de grupo* («Leave Group»): mensaje enviado por un *host* cuando abandona un grupo al que pertenecía. El valor hexadecimal del campo *tipo* para este mensaje es 17.
 - Informe de pertenencia v1*: mensaje de *tipo* 12 en hexadecimal utilizado para compatibilidad con la versión 1 de IGMP.
- *Tiempo*: unidades de tiempo, en décimas de segundo, que expresan el intervalo máximo permitido para el envío de mensajes «Report». Este campo estará a valor 0 para todos los mensajes distintos de «Query», no considerándose por el receptor o receptores del paquete.
- *Comprobación*: campo de suma de comprobación del mensaje IGMP completo.
- *Grupo*: campo de 32 bits donde se indica la dirección del grupo multidestino IP. Este campo es el que diferencia los dos subtipos de mensajes «Query» existentes comentados anteriormente, siendo su valor 0 en el caso «General Query» y especificándose el grupo en cuestión para los mensajes «Group-Specific Query».

El procedimiento seguido por IGMP se basa en las siguientes tres premisas de funcionamiento:

1. Un nodo IGMP puede actuar o no como «consultante», o *querier* (es decir, emite mensajes «Query»).
2. Por defecto, debe existir un nodo «consultante» por cada red física, por lo que el estado natural de un nodo IGMP es el de «consultante».
3. Esto último es así a menos que el *router* IGMP detecte la existencia de otro «consultante» con menor dirección IP en la misma red, en cuyo caso pasa al estado de «no consultante» para dicha red.

A partir de estas premisas, el procedimiento IGMP seguido es el siguiente:

1. Cada nodo «consultante» debe transmitir periódicamente un mensaje «General Query» a cada una de las redes físicas a las que se encuentra conectado y de las que es «consultante».

2. Cuando un *host* interesado en un grupo recibe una consulta, establece un tiempo aleatorio de respuesta comprendido entre 0 y el valor especificado en el campo *tiempo* del «Query». De esta forma, cada vez que expira el temporizador asociado al contador, lleva a cabo la transmisión de un mensaje «Report» sobre la interfaz por donde ha recibido la consulta. Este informe se emite siempre y cuando no se detecte que otro *host* del grupo lo ha generado con anterioridad. En tal caso se reiniciará el contador.
3. Cada vez que un nodo recibe un informe relativo a un grupo dado por parte de un *host*, refresca un contador de vida asociado a dicho grupo. Por el contrario, si transcurrido un tiempo máximo no se recibe informe alguno acerca de un grupo dado, este se supondrá vacío y se eliminará, no transmitiéndose con posterioridad mensajes «Query» referentes a dicho grupo.
4. Cuando un *host* se incorpora a un grupo, deberá emitir un mensaje «Group-Specific Report» para informar de este hecho si es el primero de la red en incorporarse a dicho grupo. Para evitar el potencial problema de que se pierda el mensaje, el *host* lo retransmitirá de forma espaciada una o dos veces.
5. Por su parte, cuando un *host* abandona un grupo emitirá un mensaje «Leave Group» hacia todos los nodos mediante la dirección 224.0.0.2. Esto es obligatorio si el *host* en cuestión fue el último de la red en responder a un «Query» para el grupo en cuestión.
6. Cuando un nodo «consultante» recibe un mensaje «Leave Group», emite un mensaje «Group-Specific Query» para ese grupo. Si no recibe respuesta dentro de un intervalo de tiempo dado actuará como se ha indicado en 3.

A partir del procedimiento descrito, en la Figura 9.22 se muestra el diagrama de estados de los nodos IGMP, tanto en su estado «consultante» como «no consultante», especificado en los RFC.

9.4.3. Algoritmos de encaminamiento *multicast* en Internet

Como hemos indicado al comienzo del subapartado anterior, IGMP no se implementa en los nodos intermedios de la red. Entonces, ¿cómo saben estos redireccionar adecuadamente la información correspondiente al servicio provisionado hacia los *routers* finales, a los que se encuentran ligados los destinatarios que componen cada uno de los grupos *multicast* en un momento dado? La respuesta a esta cuestión viene de la mano de diversos protocolos de encaminamiento desarrollados específicamente para transmisiones *multicast*. Entre ellos cabe señalar DVMRP («Distance-Vector Multicast Routing Protocol», RFC 1075) y MOSPF («Multicast OSPF», RFC 1584), extensiones *multicast* de RIP y OSPF, respectivamente.

El principal inconveniente, sin embargo, de ambos protocolos es que son dependientes del esquema de encaminamiento usado en *unicast*. Así, si utilizamos DVMRP para *multicast*, debemos usar RIP para transmisiones *unicast*. Lo mismo sucede con MOSPF respecto de OSPF. Frente a ellos, el protocolo PIM («Protocol Independent Multicast») fue definido para evitar este tipo de dependencias, de manera que el protocolo a considerar a nivel *unicast* puede ser cualquiera de los disponibles. Seguidamente se describe PIM.

Protocolo PIM

El protocolo PIM tiene dos modos de funcionamiento: denso y disperso, los cuales hacen referencia a la cantidad (alta o baja) de grupos *multicast* existentes en el entorno de red. El primero de ellos, PIM-DM («PIM-Dense Mode»; RFC 3973), es el más simple de los dos, aunque resulta también menos escalable. Su operación básica es simple:

- Cuando un paquete *multicast* se recibe en un *router* PIM este lo difunde mediante inundaciones, evitándose la aparición de bucles usando la técnica *RPF*, ya vista en el Apartado 6.3.4.

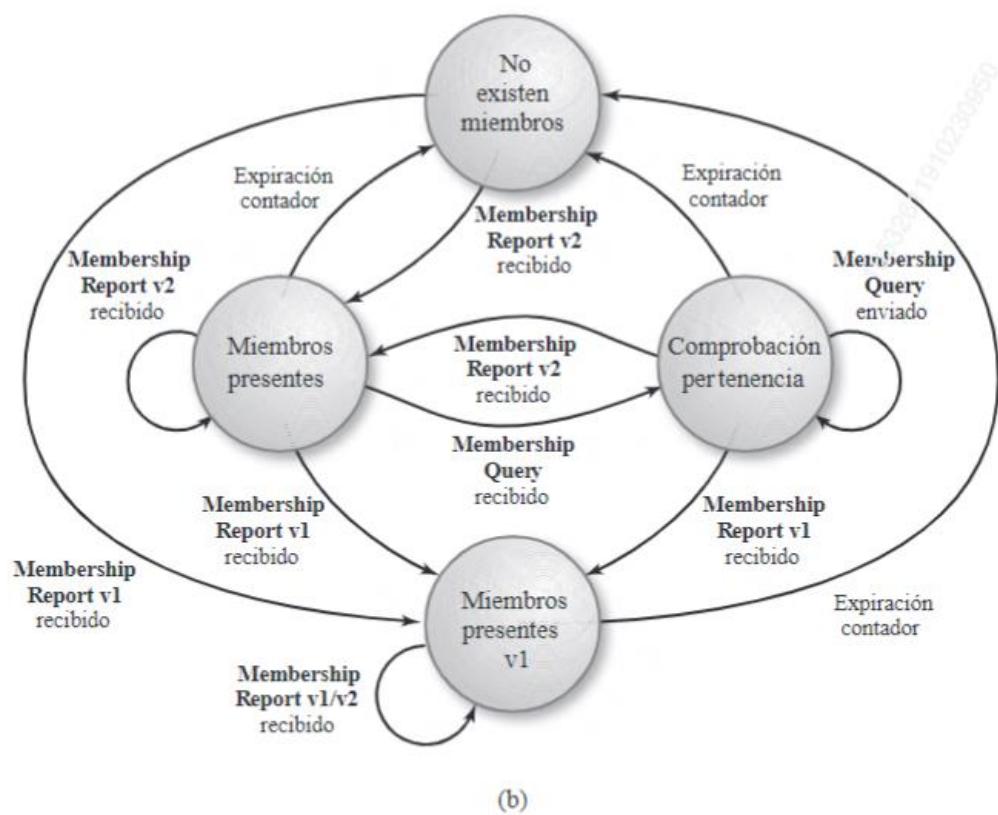
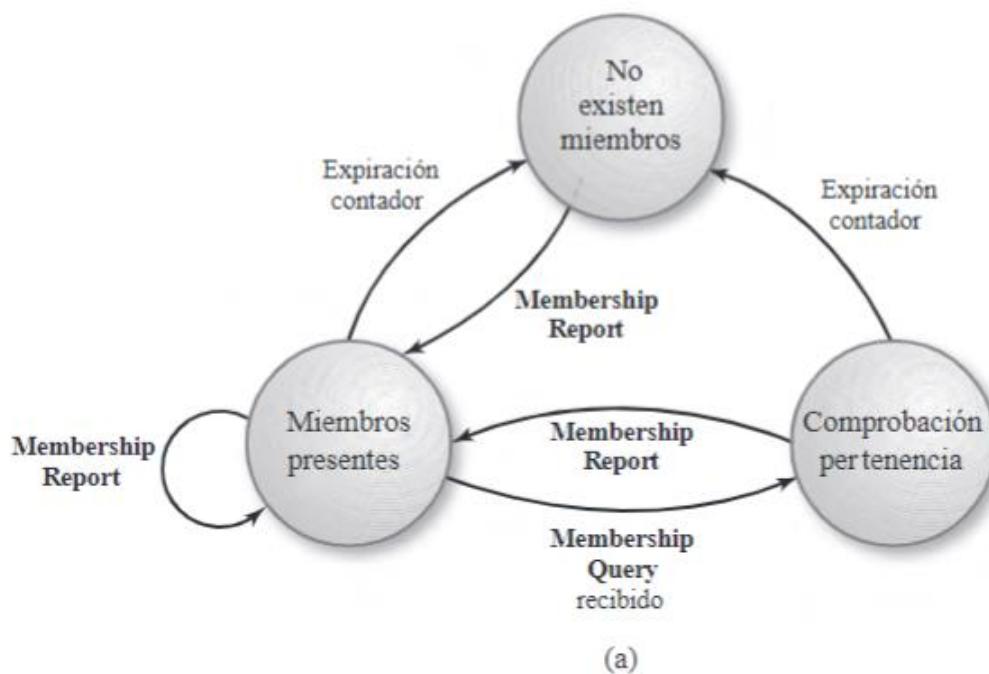


Figura 9.22. Diagrama de estados de los nodos IGMP en su función como «no consultante» (a) y como «consultante» (b).

- Implementado el algoritmo IGMP en los nodos terminales, estos pueden enviar hacia sus nodos «padre» mensajes de *poda* («prune» en inglés) en caso de no disponer de *hosts* usuarios por debajo interesados en un cierto grupo *multicast*. En este punto hay que señalar que el árbol de la red se construye tomando como punto raíz el origen o fuente de la información.

Estos mensajes pueden propagarse a su vez hacia arriba en el árbol para evitar transmisiones innecesarias. Es decir, si «debajo» de un cierto nodo no existe ningún otro con miembros de un grupo dado, aquel puede evitar la retransmisión de la información correspondiente (véase Figura 6.13).

- Una vez podada una rama del árbol, esta se puede restablecer con posterioridad si apareciesen nuevos miembros del grupo. Ello se lleva a cabo mediante mensajes de *injerto* («graft» en inglés).

Como hemos indicado anteriormente, el protocolo PIM en su versión dispersa, PIM-SM («PIM-Sparse Mode») es más complejo que la versión densa. Definido originalmente en el RFC 2362 y actualizado a través del RFC 4601, la arquitectura funcional se muestra en la Figura 9.23. De ella hay que destacar la existencia de los siguientes elementos:

- Dominios PIM, esto es, entornos donde se implementa PIM.
- Dominios no PIM.
- *Routers frontera multicast* o MBR («Multicast Border Router»), los cuales permiten la interconexión de los tipos de dominios anteriores y, en suma, el encaminamiento *multicast*.
- *Routers PIM designados* (RD), puntos *rendezvous* (RP) y nodos *bootstrap*, involucrados en el desarrollo del encaminamiento *multicast* como se explica a continuación.

La transmisión en PIM-SM es como sigue (véase Figura 9.24):

- Para que un *router* reciba datos *multicast* debe indicarlo explícitamente a sus vecinos hacia arriba en el árbol. Para ello, los nodos intercambian mensajes PIM de *incorporación* («join») y *poda* («prune»). Estos, de modo análogo a como sucede en PIM-DM, se sustentan en el empleo de IGMP en los nodos hoja o finales.
- PIM-SM hace uso de árboles compartidos, pero tomando como punto raíz un nodo denominado *rendezvous* (RP) en lugar del origen de los datos como sucede en PIM-DM. Así, el RP es el encargado de hacer las transmisiones *multicast* hacia todos los receptores del correspondiente grupo dentro del dominio.

La existencia de los *routers* RP es anunciada a través de nodos especiales llamados *bootstrap*.

- Para enviar al RP, el origen u orígenes de la información encapsulan los datos en mensajes de control PIM y se envían en modo *unicast* al RP desde el nodo designado (RD) de la red local del origen.

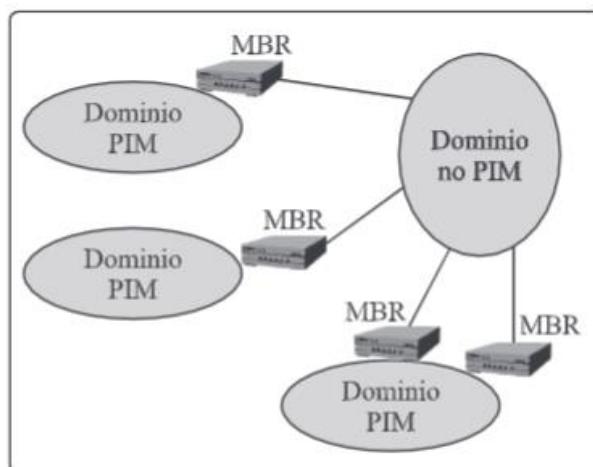


Figura 9.23. Arquitectura PIM-SM.

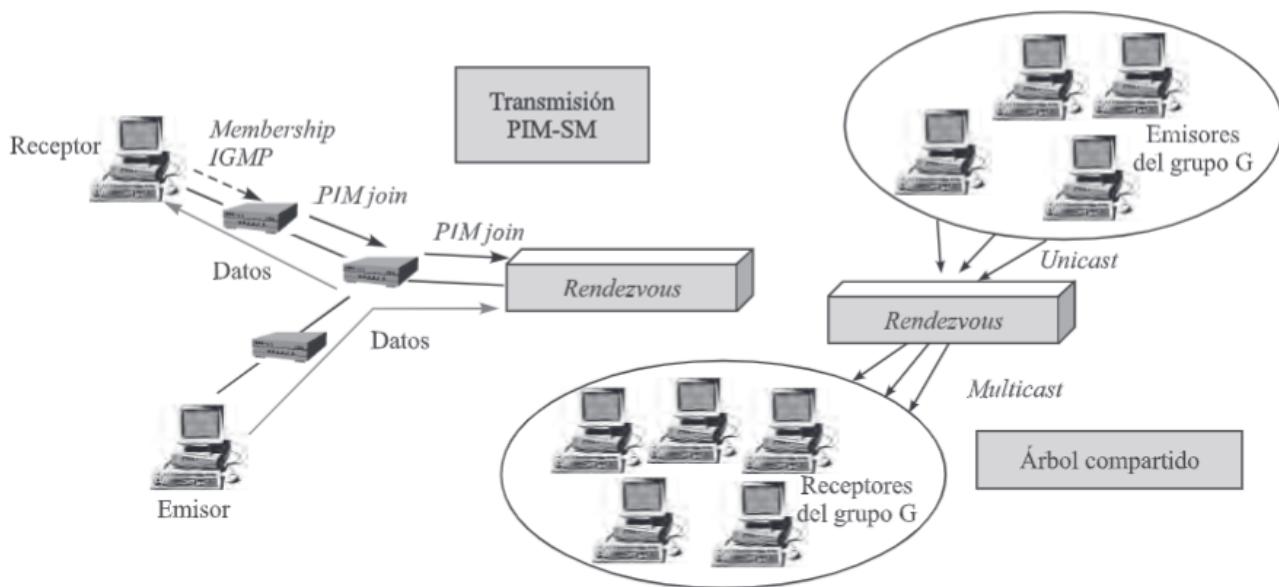


Figura 9.24. Funcionamiento operacional de PIM-SM.

El formato concreto de los mensajes PIM-SM es el indicado en la Figura 9.25, correspondiendo los campos sombreados a la cabecera. El más relevante de ellos desde el punto de vista de la funcionalidad pretendida es el campo *tipo*, el cual indica el tipo de mensaje PIM de que se trata. Los tipos disponibles en la versión 2 actual del protocolo son:

- *Hello* (*tipo=0*). Mensaje enviado periódicamente entre *routers* vecinos. El de dirección IP superior será el nodo designado o RD.
- *Register* (*tipo=1*). Mensaje sobre el que se encapsulan los datos enviados hacia los receptores de un grupo dado, y que se envía desde los RD al RP.
- *Register-Stop* (*tipo=2*). Contestación al mensaje anterior desde el RP para el control de flujo.
- *Join/Prune* (*tipo=3*). Incorporación/poda de nodos en el árbol de transmisión de la red.
- *Bootstrap* (*tipo=4*). Mensajes enviados mediante *multicast* al grupo ALL-PIM-ROUTERS (224.0.0.13) con información acerca del RP.
- *Assert* (*tipo=5*). Enviado a la dirección 224.0.0.13 para resolver la designación de RD.
- *Graft* (*tipo=6*). De uso solo en PIM-DM y cuya funcionalidad es el injerto de ramas en el árbol de transmisión sobre la red.
- *Graft-Ack* (*tipo=7*). De uso solo en PIM-DM y cuya funcionalidad es servir de confirmación al mensaje anterior.
- *Candidate-RP-Advertisement* (*tipo=8*). Mensaje enviado (*unicast*) periódicamente desde los RP candidatos hacia los nodos *bootstrap*.

0	5	8	16	31
Versión	Tipo	Reservado	Complicación	
Mensaje...				

Figura 9.25. Formato de mensaje PIM-SM.

Concluimos este apartado citando sin más la existencia de una tercera variante de PIM: PIM bidireccional (BIDIR-PIM), así como la disposición de algunos otros protocolos alternativos para llevar a cabo el encaminamiento *multicast* en Internet, como son MSDP («Multicast Source Discovery Protocol») y CBT («Core-Based Trees»). También es de mencionar el BGMP («Border Gateway Multicast Protocol») como protocolo *multicast* de tipo exterior, extensión de BGP.

RESUMEN

En este capítulo se ha abordado el estudio de las funciones propias de la capa de red en Internet. Como núcleo de la misma y soporte principal del resto de protocolos y servicios Internet se ha presentado el protocolo IP, el cual se caracteriza por ofrecer un servicio de transmisión datagrama no orientado a conexión y no fiable. A partir del formato del datagrama se ha comentado la capacidad de fragmentación y especificación de opciones IP.

Ante algunas evidentes limitaciones de la versión más conocida y usada de IP, IPv4, se ha discutido también su versión 6, IPv6. De ella se han destacado el direccionamiento extendido y las cabezas de extensión, entre otros aspectos relevantes.

A pesar de la naturaleza de mejor esfuerzo demostrada por IP, se hace necesaria la implementación de procedimientos que permitan un cierto nivel de control de la subred para monitorizar el funcionamiento global de esta. Se ha introducido así el protocolo ICMP, el cual, encapsulado sobre IP, establece varios mensajes de control con los que se gestionan determinadas circunstancias que pueden presentarse en la transmisión de los datagramas. Como extensión al uso de IPv6, también se ha presentado la versión 6 de ICMP, ICMPv6.

Una cuestión de suma importancia en toda red se refiere al encaminamiento y actualización de las tablas por parte de los nodos intermedios. Presentado el concepto de sistema autónomo y el carácter distribuido jerárquico de los protocolos de encaminamiento en Internet, en el Apartado 9.3 se estudian tanto protocolos interiores, en particular RIP, OSPF y los propietarios de Cisco IGRP y EIGRP, como protocolos exteriores, en concreto BGP.

Para concluir el capítulo se han discutido las transmisiones *multicast* en Internet. Tres han sido las cuestiones a este respecto tratadas: gestión de direcciones *multicast*, gestión de grupos *multicast* y encaminamiento *multicast* sobre la subred. En lo concerniente a la primera se ha estudiado la arquitectura MALLOC, y los protocolos IGMP y PIM al respecto de la segunda y tercera cuestiones, respectivamente.

EJERCICIOS

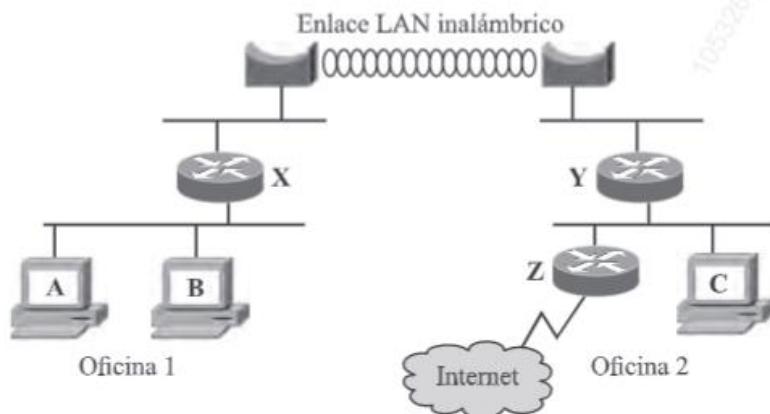
1. ¿Tendría sentido que el primer campo del paquete IP fuese otro distinto del de *versión*? Razona la respuesta.
2. Un nodo debe llevar a cabo la transmisión de un datagrama N , con un *payload* de longitud 1.500 bytes, correspondiente a una comunicación IP dada sobre una red de transporte con MTU igual a 1.200 octetos. Ante esta situación, ¿qué campos de la cabecera IP de los fragmentos será necesario modificar?
3. Un dispositivo de encaminamiento recibe un paquete que debe retransmitir sobre una red con MTU igual a 640 octetos. Si el paquete tiene una cabecera IP mínima y un campo de datos de 1.960 bytes, realice la fragmentación e indique los valores de los campos de la cabecera IP del paquete original y de cada fragmento según la siguiente tabla:

Paquete	Longitud cabecera	Longitud total	Protocolo	ID	MF	Offset
Original						
Fragmento 1						
...						

4. En el ejercicio anterior, ¿qué sucedería si el bit *DF* del paquete original estuviese especificado a valor 0?
5. Una empresa tiene dos oficinas (1 y 2) conectadas mediante un enlace LAN inalámbrico, como se ilustra en la figura inferior. Suponga que la empresa contrata una línea dedicada con un proveedor de Internet, el cual le ha asignado al *router* de acceso Z la dirección IP 192.169.15.6, con máscara de red de 30 bits. Suponga también que la empresa obtiene de su proveedor una dirección pública de red 150.214.60.0.

Utilizando las direcciones arriba mencionadas:

- Realice una asignación de las direcciones IP para los distintos equipos.
- Indique todas las tablas de encaminamiento.
- Indique qué haría si apareciera un nuevo grupo de ordenadores (D) en la oficina 1 con 70 nuevos usuarios.



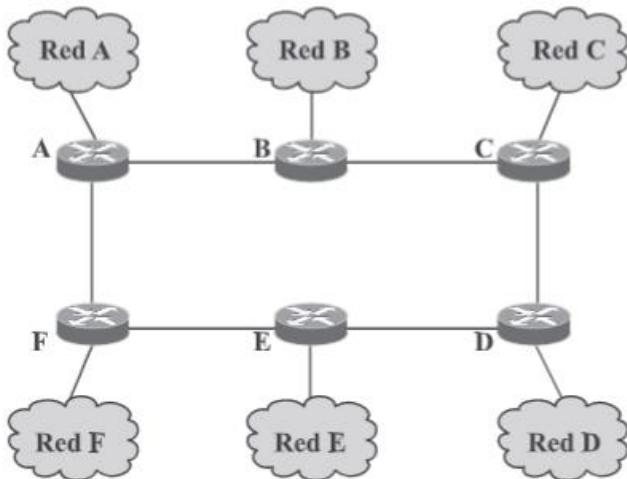
6. Se desea desplegar una red corporativa con acceso a Internet con las siguientes características básicas:
- Dos departamentos, cada uno con 5 equipos de trabajo, 1 servidor de ficheros y 1 impresora en red. Uno de los departamentos se encuentra ubicado en un edificio A y el otro en un edificio B separado 50 metros del primero.
 - Zona de servidores públicos: HTTP y DNS.
 - Aunque se usarán preferentemente direcciones privadas, también se dispone del rango de direcciones públicas 199.199.199.8/29 para la DMZ.
 - El *router* de acceso a Internet solo debe contener 4 entradas totales en su tabla de *routing*.
- Proponga y dibuje una topología completa para la red pretendida.
 - Lleve a cabo una asignación de direcciones IP y máscaras asociadas.
 - Indique la tabla de encaminamiento del *router* de acceso a Internet.

7. La dirección IPv6 1080:0000:0000:002c:0000:0000:0000:417a ¿de qué tipo es, *unicast* o *multicast*? Especifíquela en formato abreviado.
8. Indique la dirección IPv4 10.214.23.3 en formato IPv6.
9. Ponga un ejemplo y explique el funcionamiento de un servicio sustentado sobre transmisiones *anycast*.
10. Frente a IPv4, en IPv6 es el emisor, y no los nodos intermedios, el encargado de fragmentar los paquetes IP. ¿Qué implicaciones tiene desde el punto de vista del origen? ¿Qué ventajas y desventajas presenta esta alternativa frente a la tradicional?
11. Supongamos que, de modo análogo al enunciado del Ejercicio 3, se dispone de 1.960 bytes de datos que transmitir sobre una topología de red con MTU mínima de 640 bytes. Responda a las siguientes cuestiones sobre el proceso de fragmentación si consideramos el uso del protocolo IPv6:
 - a) Describa el proceso general de fragmentación seguido por el emisor.
 - b) ¿Cuántos datagramas IPv6 se generarán y cuál será el valor del campo *desplazamiento* de las cabeceras de extensión correspondientes?
 - c) ¿Cuál es el formato general de cada uno de los fragmentos desde el punto de vista de las cabeceras existentes?
12. La mayoría de los mensajes ICMP (p.e., *destino inalcanzable*, *tiempo excedido* y *problema de parámetros*) incluyen en el campo *datos*: «cabecera IP más los primeros 64 bits del campo *datos* del datagrama original». ¿Cuál es el objetivo de esta información?
13. El mensaje ICMP «source quench» se envía desde un *router* hacia el origen del paquete que motiva el mensaje. ¿Tendría sentido que estos mensajes fuesen destinados a un *router* o *routers* vecinos? Justifique la respuesta.
14. Suponga que un *host* dado envía un paquete IP sobre un cierto *router* hacia el destino pretendido. Si con posterioridad el *host* recibiese un mensaje ICMP de *tipo* = 5 y *código* = 0, ¿qué ocurriría?
15. Pruebe en su red de trabajo el comando *ping* y explique el resultado obtenido en relación a los mensajes ICMP involucrados. Repita el proceso con el comando *traceroute*.
16. El protocolo de encaminamiento RIP es susceptible de presentar el problema conocido como *cuenta al infinito*. ¿Es esta situación generalizable a otros protocolos IGP como, por ejemplo, OSPF? Justifique la respuesta.
17. Suponga que al *router* X en la red del Ejercicio 5 se le instala una tercera interfaz a una nueva red de dirección 150.214.70.0. Suponga también que el administrador opta por instalar el protocolo RIP en todos los equipos. Explique el funcionamiento de este protocolo de encaminamiento dinámico, identificando cada uno de los paquetes que aparecerían para llevar a cabo la actualización de las tablas de encaminamiento. (*Nota: Haga las suposiciones que estime necesarias*)

	ETH. ORIG.	ETH. DEST.	IP ORIG.	IP DEST.	DESCRIPCIÓN
1					
2					
3					
...					

18. Los *routers* de la figura adjunta tienen definidas las rutas a las redes que tienen conectadas directamente. El administrador de la red decide utilizar en dichos *routers* el protocolo RIP (en las

interfaces hacia otros *routers*) y activa dicho servicio siguiendo la secuencia temporal indicada a la derecha de la figura (en segundos).



- $t = t_0$ → Activación RIP en *router A*
- $t = t_0 + 5$ → Activación RIP en *router B*
- $t = t_0 + 10$ → Activación RIP en *router C*
- $t = t_0 + 15$ → Activación RIP en *router D*
- $t = t_0 + 20$ → Activación RIP en *router E*
- $t = t_0 + 25$ → Activación RIP en *router F*

- a) Explique el funcionamiento del protocolo de encaminamiento dinámico RIP mediante la descripción de los mensajes intercambiados entre los *routers* (indique origen/destino del mensaje, redes conocidas por el receptor tras recibir el mensaje, coste para alcanzar cada red y cuál es el primer *router* en la ruta hacia dicha red) hasta que las rutas se mantienen estables.

Suponga que solo se utilizan actualizaciones periódicas y que el primer mensaje periódico enviado por cada *router* es a los 30 segundos de haber arrancado el servicio RIP. Incluya en la descripción solo la accesibilidad a las redes A, B, C, D, E y F.

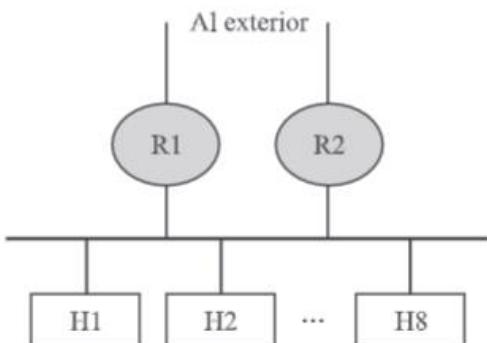
- b) ¿Cuál es el tiempo total transcurrido hasta que la situación de toda la red se ha estabilizado (desde el instante t_0)?

19. A través del campo *TTL* del datagrama IP se especifica el tiempo de vida máximo permitido para el paquete; sin embargo, dicho campo se suele especificar a valor 1 en transmisiones *multicast*. Justifique razonadamente el motivo de este hecho.

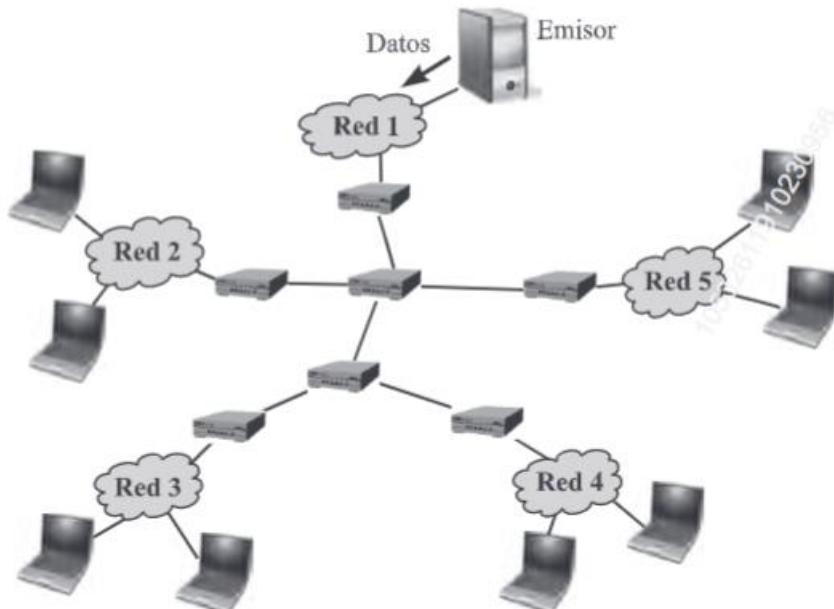
20. En relación al protocolo IGMP:

- a) Identifique y explique todos los contadores necesarios en un *router* «consultante».
 b) Identifique y explique todos los contadores en un *host* miembro de 2 grupos *multicast* distintos.

21. Disponemos de una red con la topología indicada en la figura adjunta y en la que tanto los *hosts* como los nodos R1 y R2 implementan el protocolo IGMP. Supuesta la existencia de los grupos *multicast* 224.135.22.4 y 224.7.22.4 por parte de R1 y R2,



- a) ¿Qué dispositivo/s emitiría/n el mensaje MQ «¿quién quiere formar parte de un grupo multicast?»? ¿Sería este mensaje de tipo GMQ o SGMQ?
- b) Los hosts H1, H3 y H5 están interesados en formar parte del grupo 224.135.22.4, mientras que solo el host H7 quiere pertenecer a 224.7.22.4. Indique los mensajes MR enviados en respuesta al MQ de a), contemplando los temporizadores involucrados.
- c) ¿Cómo abandonaría el host H3 el grupo al que está suscrito?
22. Suponga la infraestructura de red inferior, donde se dispone de un emisor *multicast* y se implementan los algoritmos PIM-DM e IGMP. Explique los mensajes intercambiados y el árbol de transmisión en las siguientes situaciones temporales:
- Inicialmente, ningún host en la red desea recibir información del grupo.
 - En t_0 , un host de Red 3 se incorpora al grupo para recibir la información correspondiente.
 - Posteriormente, en t_1 un host de Red 4 solicita la incorporación.
 - En t_2 el segundo host de Red 4 se incorpora también.
 - En t_3 el host de Red 3 se da de baja del grupo.
 - En t_4 uno de los hosts de Red 4 se da de baja del grupo.



23. Tal como se ha indicado en el tema, una posibilidad para dar a conocer la dirección del grupo *multicast* del ejercicio anterior es hacerlo mediante el protocolo SAP. Consulte el RFC correspondiente y describa el proceso asociado.

BIBLIOGRAFÍA

- Adams, A.; Nicholas, J.; Siadak, W.: *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)*. RFC 3973. Enero, 2005.
- Almquist, P.: *Type of Service in the Internet Protocol Suite*. RFC 1349. Julio, 1992.
- Braden, R.T.; Postel, J.: *Requirements for Internet Gateways*. RFC 1009. Junio, 1987.
- Comer, D.E.: *Internetworking with TCP/IP. Volume I: Principles, Protocols and Architecture*. 3.^a edición. Prentice Hall, 1995.
- Conta, A.; Deering, S.: *Internet Control Message Protocol (ICMP) for the Internet Protocol Version 6 (IPv6)*. RFC 1885. Diciembre, 1995.

- Deering, S.E.: *Host Extensions for IP Multicasting*. RFC 1112. Agosto, 1989.
- Deering, S.: *ICMP Router Discovery Messages*. RFC 1256. Septiembre, 1991.
- Deering, S.; Hinden, R.: *Internet Protocol, Version 6 (IPv6). Specification*. RFC 2460. Diciembre, 1998.
- Fenner, W.: *Internet Group Management Protocol, Version 2*. RFC 2236. Noviembre, 1997.
- Fenner, B.; Handley, M.; Holbrook, H.; Kouvelas, I.: *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*. RFC 4601. Agosto, 2006.
- Hanna, S.; Patel, B.; Shah, M.: *Multicast Address Dynamic Client Allocation Protocol (MADCAP)*. RFC 2730. Diciembre, 1999.
- Hedrick, C.L.: *Routing Information Protocol*. RFC 1058. Junio, 1988.
- Hinden, R.M.; Sheltzer, A.: *DARPA Internet gateway*. RFC 823. Septiembre, 1982.
- Hinden, R.; S. Deering.: *IP Version 6 Addressing Architecture*. RFC 2373. Julio, 1998.
- Kent, S.; Atkinson, R.: *IP Authentication Header*. RFC 2402. Noviembre, 1998.
- Kent, S.; Atkinson, R.: *IP Encapsulating Security Payload (ESP)*. RFC 2406. Noviembre, 1998.
- Kent, S.; Seo, K.: *Security Architecture for the Internet Protocol*. RFC 4301. Diciembre 2005.
- Kurose, J.F.; Ross, K.W.: *Computer Networking. A Top-Down Approach*. Addison Wesley, 2013. 6.^a edición.
- Madson, C.; Glenn, R.: *The Use of HMAC-MD5-96 within ESP and AH*. RFC 2403. Noviembre, 1998.
- Madson, C.; Glenn, R.: *The Use of HMAC-SHA-1-96 within ESP and AH*. RFC 2404. Noviembre, 1998.
- Madson, C.; Doraswamy, N.: *The ESP DES-CBC Cipher Algorithm with Explicit IV*. RFC 2405. Noviembre, 1998.
- Malkin, G.: *RIP Version 2*. RFC 2453. Noviembre, 1998.
- Meyer, D.: *Administratively Scoped IP Multicast*. RFC 2365. Julio, 1998.
- Mills, D.L.: *Exterior Gateway Protocol Formal Specification*. RFC 904. Abril, 1984.
- Moy, J.: *OSPF Specification*. RFC 1131. Octubre, 1989.
- Moy, J.: *Multicast Extensions to OSPF*. RFC 1584. Marzo, 1994.
- Moy, J.: *OSPF Version 2*. RFC 2328. Abril, 1998.
- Partridge, C.; Mendez, T.; Milliken, W.: *Host Anycasting Service*. RFC 1546. Noviembre, 1993.
- Postel, J.: *Internet Protocol*. RFC 791. Septiembre, 1981.
- Postel, J.: *Internet Control Message Protocol*. RFC 792. Septiembre, 1981.
- Radoslavov, P.; Estrin, D.; Govindan, R.; Handley, M.; Kumar, S.; Thaler, D.: *The Multicast Address-Set Claim (MASC) Protocol*. RFC 2909. Septiembre, 2000.
- Ramakrishnan, K.; Floyd, S.: *A Proposal to Add Explicit Congestion Notification (ECN) to IP*. RFC 2481. Enero, 1999.
- Rekhter, Y.; Li, T.: *A Border Gateway Protocol 4 (BGP-4)*. RFC 1771. Marzo, 1995.
- Reynolds, J.; Postel, J.: *Assigned Numbers*. RFC 1700. Octubre, 1994.
- Stallings, W.: *Comunicaciones y Redes de Computadores*. Pearson Educación, 2004. 7.^a edición.
- Stevens, W.R.: *TCP/IP Illustrated, Vol. 1. The Protocols*. Ed. Addison Wesley, 2000.
- Thaler, D.; Handley, M.; Estrin, D.: *The Internet Multicast Address Allocation Architecture*. RFC 2908. Septiembre 2000.
- Waitzman, D.; Partridge, C.; Deering, S.E.: *Distance Vector Multicast Routing Protocol*. RFC 1075. Noviembre, 1988.