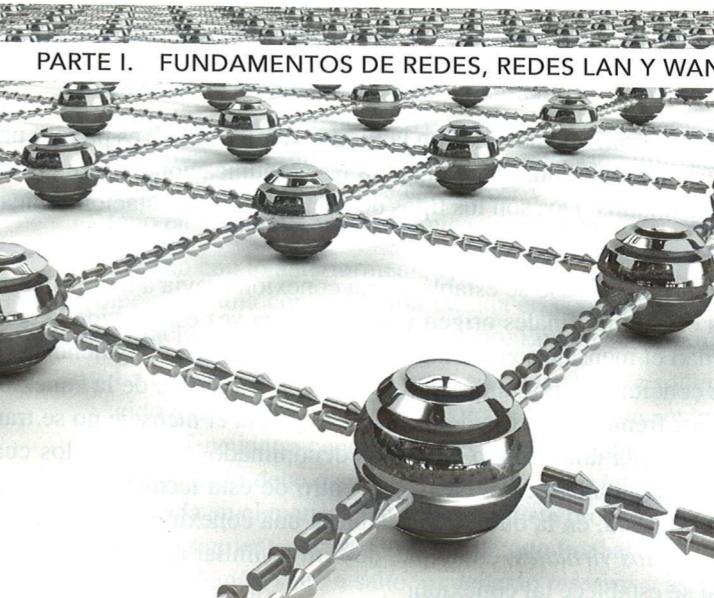


9. La banda ISM 2,4 GHz es, como ya se ha indicado, no licenciada, por lo que múltiples sistemas la utilizan. Obtenga un listado de las tecnologías/sistemas de comunicaciones que operan en dicha banda.
10. Considere un acceso ADSL en el que se utiliza el rango 25-200 kHz para el canal ascendente y el rango 250 a 1.000 kHz para el descendente. Durante el proceso de inicialización de la línea se ha medido una SNR de 30 dB en el rango 25-650 kHz y de 20 dB en el rango 650-1.000 kHz. Se utilizan subcanales con un ancho de banda de 4 kHz por canal, según el estándar. Calcule:
 - a) La capacidad máxima de cada uno de los subcanales ascendentes y las de cada uno de los subcanales descendentes.
 - b) De acuerdo con el estándar, se utiliza QAM en cada uno de los subcanales. ¿Cuál es el tamaño de las constelaciones en cada subcanal?
 - c) ¿Cuáles son las capacidades globales del sistema?
11. El estándar DOCSIS 2.0 permite la utilización de anchos de banda de subida de 0,2, 0,4, 0,8, 1,6, 3,2 y 6,4 MHz con codificación QPSK, QAM16 o QAM64. Obtenga una tabla con las velocidades de transmisión del canal ascendente que pueden obtenerse.
12. Realice una comparativa de precios de las tecnologías de acceso disponibles en su vecindario. Para ello, evalúe el coste por Mbps de cada una.

BIBLIOGRAFÍA

- Carbou, R., Díaz, M., Expósito, E., Román, R.: *Digital Home Networking*, John Wiley & Sons, 2011.
 Excentis: *Cable Network handbook*, 2009. Available at <http://docbox.etsi.org/>.
- Feit, S.: *Local Area High Speed Networks*. MacMillan, 2000.
- Gast, M.S.: *802.11 Wireless Networks*, O'Reilly, 2002.
- IEEE 802 LAN/MAN Standards Committee. <http://www.ieee802.org>
- León-García, A.; Widjaja, I.: *Redes de Comunicación. Conceptos Fundamentales y Arquitecturas Básicas*. McGraw-Hill, 2001.
- Shay, W.: *Understanding Data Communications and Networks*, Thomson, 2004, 3.^a edición.
- Stallings, W.: *Comunicaciones y Redes de Computadores*. Prentice-Hall, 2004. 7.^a edición.
- Tanenbaum, A. D.: *Computer Networks*. Prentice-Hall, 2011. 5.^a edición.
- USB IF: *Universal Serial Bus Specification. Revisión 2.0*. 2000. <http://www.usb.org>
- WiMAX Forum: *Mobile WiMAX – Part I: A Technical Overview and Performance Evaluation*, 2006.
- WiMAX Forum: *WiMAX End-to-End Network Systems Architecture – Stage 2: Architecture Tenets, Reference Model and Reference Points*, 2006.
- ZigBee Alliance: *ZigBee Specifications*, 2007. <http://www.zigbee.org>.



ENCAMINAMIENTO E INTERCONEXIÓN DE REDES

CAPÍTULO

6

- 6.1. Introducción
- 6.2. Conceptos y técnicas de comutación
- 6.3. Encaminamiento
- 6.4. 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 cualesquier 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 *comutación* 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 precisa 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.

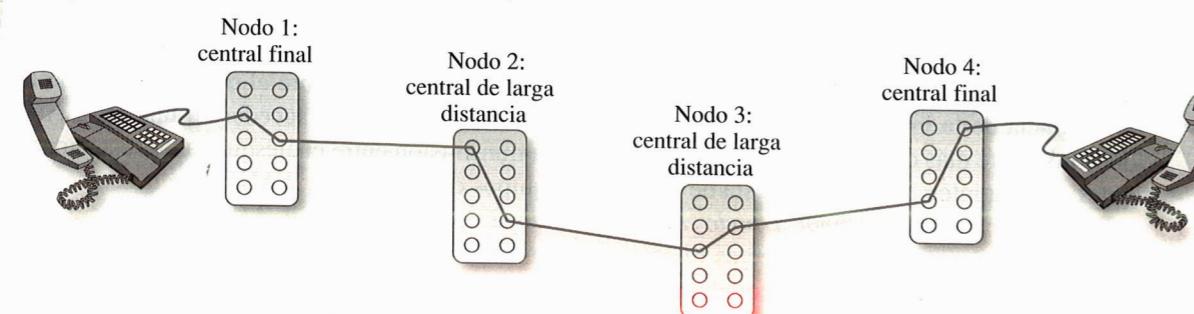


Figura 6.1. Ejemplo de conexión telefónica mediante conmutación de circuitos.

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

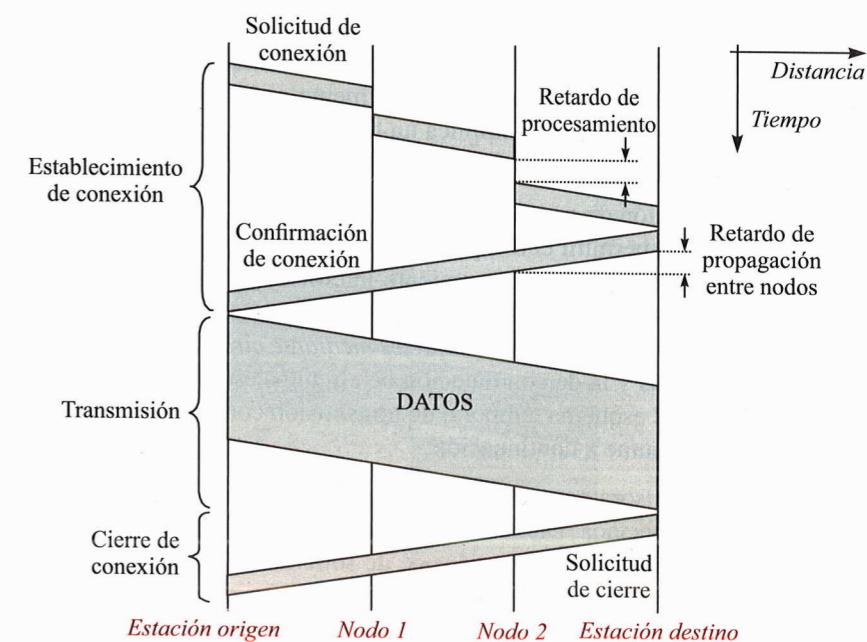


Figura 6.2. Esquema de transmisión mediante conmutación de circuitos entre dos estaciones finales con dos nodos intermedios.

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

- No se establece conexión previa a la transmisión de los datos.
 - El mensaje se divide en bloques más pequeños denominados *paquetes* o, derivado del nombre de la técnica, *datagramas*.
 - 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.
- 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:

- 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

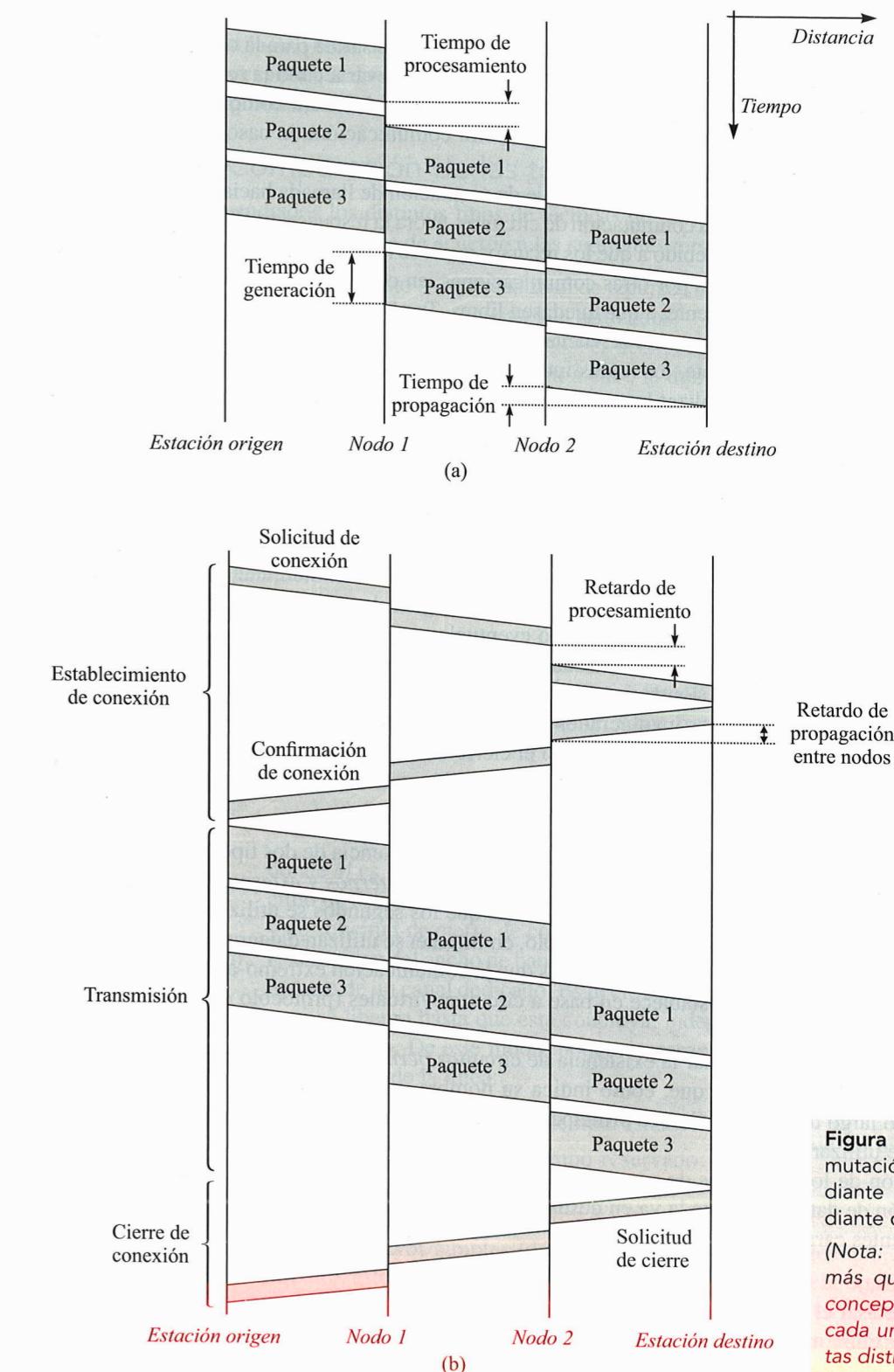


Figura 6.3. Ejemplo de comutación de paquetes: mediante datagramas (a) y mediante circuitos virtuales (b).
(Nota: el esquema (a) no es más que una representación conceptual, pudiendo seguir cada uno de los paquetes rutas distintas.)

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 habrá 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 *conmutación 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:

— Conmutación 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-línea 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.

- 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 T_c será:

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

En conclusión, $T_{CC} = T_e + T_t + T_c$ 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:

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

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, $T_{CPD} = T_t$:

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

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

- Establecimiento de conexión. Los pasos seguidos en este proceso son:

- 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:

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

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

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

- Propagación del mensaje hacia la estación origen:

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

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

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

En consecuencia, el tiempo de establecimiento T_e en este caso será:

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

- Transmisión de datos:

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

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

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

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

- b) Siguiéndole la pista al último paquete, los procesos sobre el mismo serán:

- Generación en los N nodos intermedios, esto es, en el resto de enlaces:

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

- Procesamiento en cada nodo intermedio:

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

- Propagación en cada salto:

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

Así:

$$T_t = \frac{P \cdot M}{[R_1 \cdot (P - H_c)]} + \sum_{i=2}^L \frac{P}{R_i} + \sum_{i=1}^L D_i + \sum_{i=1}^N T_i^{CPCV} \text{ segundos}$$

- Cierre de conexión. Como en el caso de la comutación de circuitos, con la salvedad de que la transmisión sufre retardos en los nodos intermedios:

- a) Generación del mensaje de solicitud de cierre en cada salto:

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

- b) Procesamiento en los nodos intermedios y en el receptor del paquete:

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

- c) Propagación del mensaje en cada salto:

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

Así:

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

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):

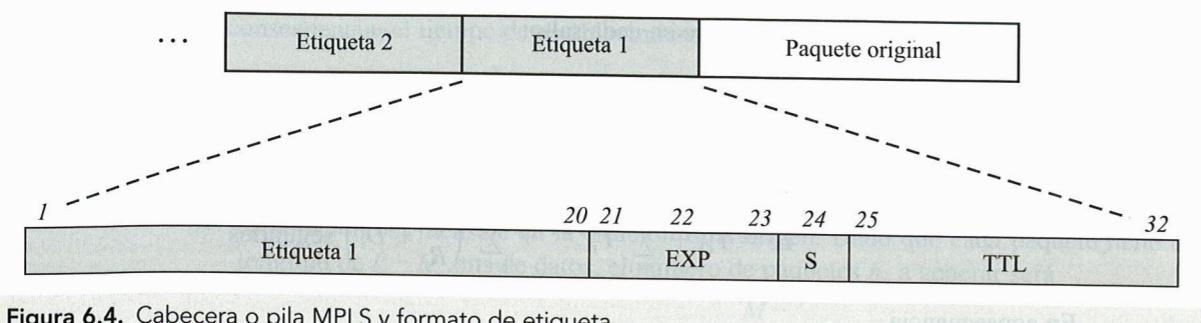


Figura 6.4. Cabecera o pila MPLS y formato de etiqueta.

- 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 commutación 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 commutación 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 commutación 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 commutación 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 commutación de circuitos y de commutación de paquetes mediante circuitos virtuales, y para cada paquete en el caso de servicios no orientados a conexión, como es la técnica de commutación 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:

$$\begin{aligned} A_i &\leftarrow \min [A_i, A_j + C_{ji}] && \rightarrow \text{concatenación de caminos si el último término, } A_j + C_{ji}, \\ &&& \text{es menor} \\ \text{Ir al paso 2} &&& \end{aligned}$$

— *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

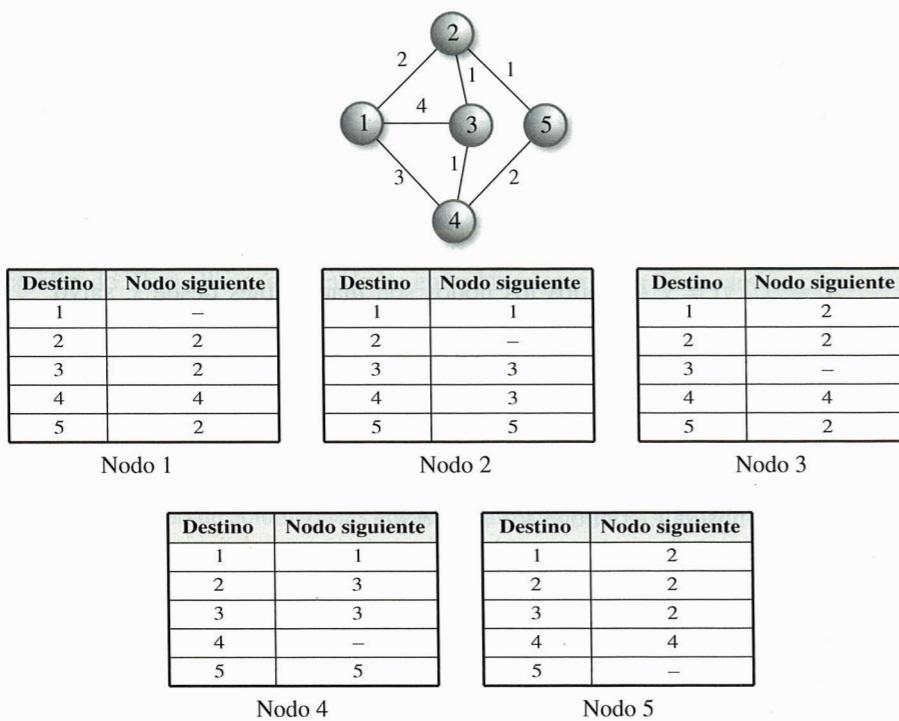


Figura 6.5. Ejemplo de topología de red, con costes por enlace, y tablas de encaminamiento de mínimo coste resultantes asociadas a los distintos nodos.

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

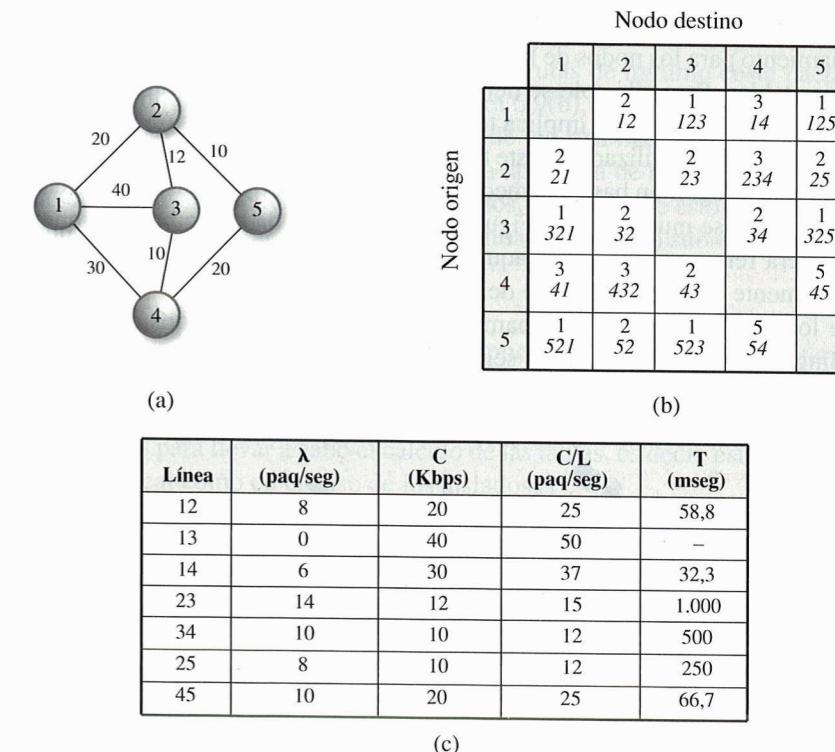


Figura 6.6. Encaminamiento basado en flujo: topología ejemplo y capacidades de línea full-duplex, en kbps, asociadas (a); matriz de encaminamiento y tráfico estimado entre cada par de nodos origen-destino (b); análisis de retardo medio sufrido en cada enlace suponiendo una longitud de paquete igual a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

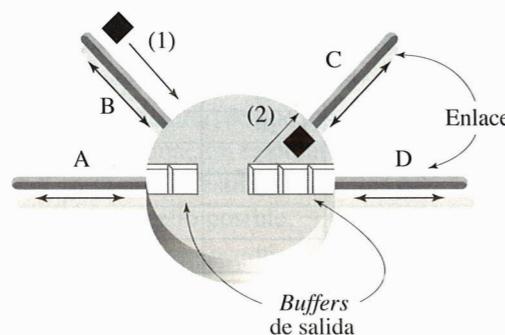


Figura 6.7. Ejemplo de nodo en el que se utiliza encaminamiento aislado con información local basada en retardo. Dado que la cola de salida más corta es la de la línea C, y supuesto que todas ellas tienen la misma velocidad, el paquete recibido (con fondo negro), (1), se reenviaría sobre ella, (2).

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.

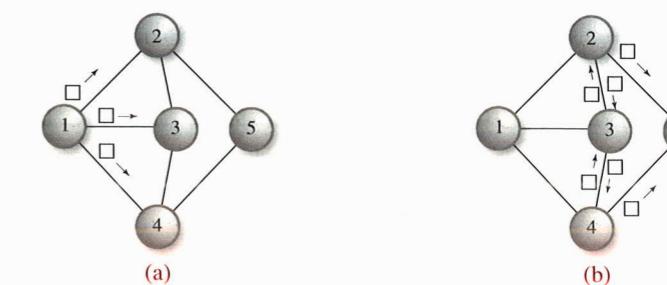


Figura 6.8. Ejemplo de transmisión de un paquete desde el nodo 1 al nodo 5 siguiendo el esquema de inundaciones: primer salto (a) y segundo salto (b).

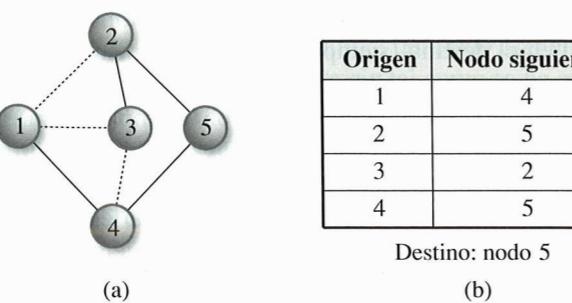


Figura 6.9. Árbol de expansión para la topología de la Figura 6.8 tomando como nodo raíz el 5 y métrica el número de saltos (a). Rutas de encaminamiento de mínimo coste desde el resto de nodos a dicho nodo destino (b).

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

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.

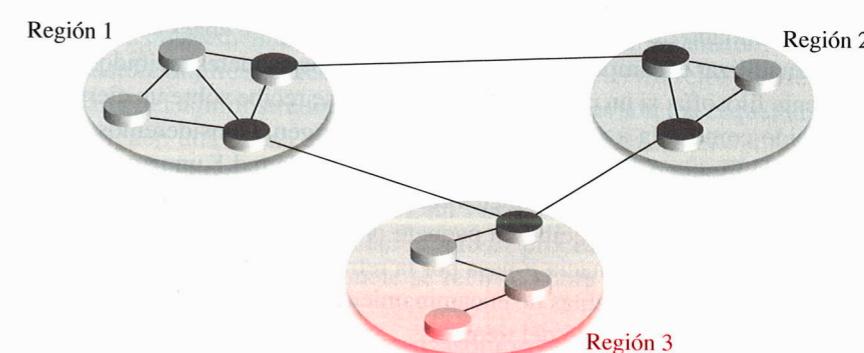


Figura 6.11. División de una red en regiones para la estimación jerárquica de las tablas de encaminamiento.

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 deseé 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 N2 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 N1 recibe un paquete originado, por ejemplo, por F sobre la red Y, N1 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.

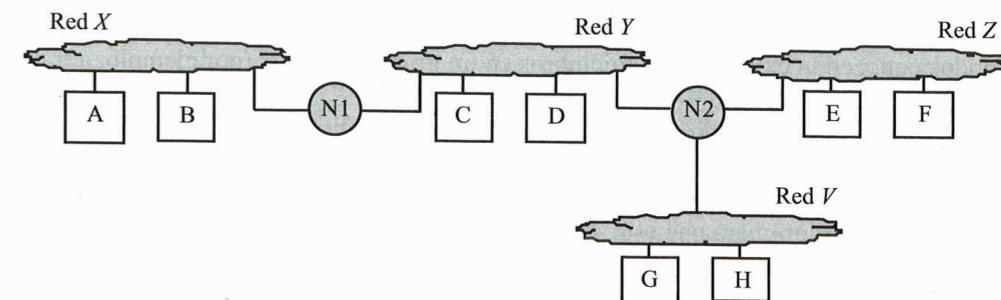


Figura 6.12. Configuración de interconexión con cuatro redes y dos nodos, N1 y N2.

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 *multidesino 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 conmutación 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 comutación y encaminamiento son directamente aplicables a una interconexión de redes.

Sí 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 possibilitar 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.

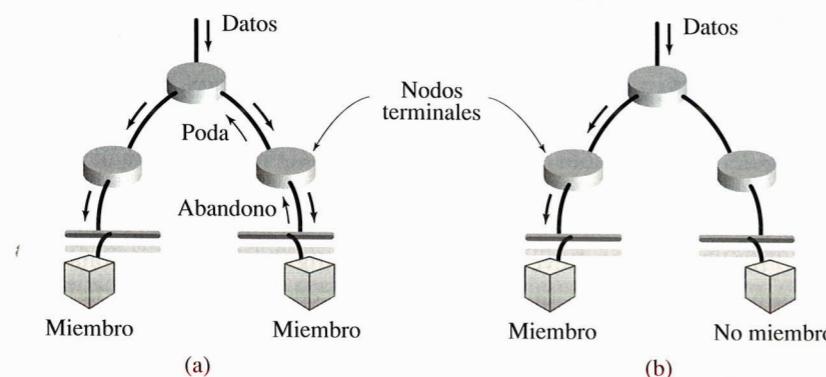


Figura 6.13. Modificación del árbol multicast en RPM ante el abandono del grupo por parte de uno de los miembros del mismo.

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

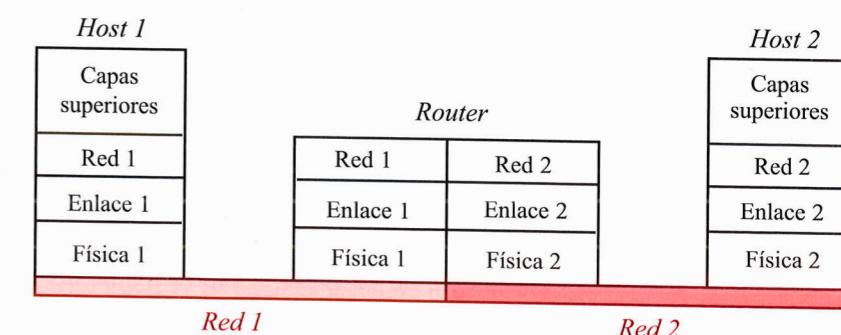


Figura 6.14. Modelo de red en una interconexión a través de un router.

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* deco-dificará, por un lado, la señal recibida para pasárla 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)).

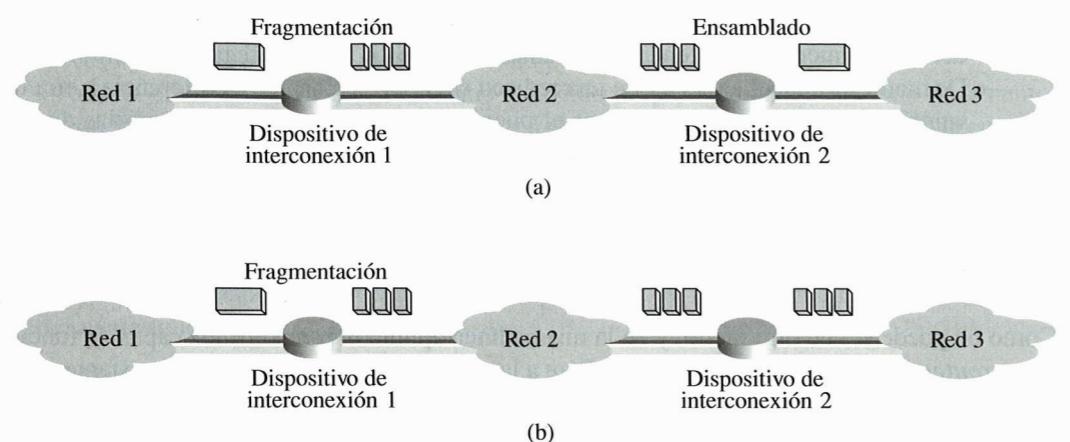


Figura 6.15. Fragmentación transparente (a) frente a no transparente (b).

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* (comutador) 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 comutador 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 multidiestino 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 comunicación 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.

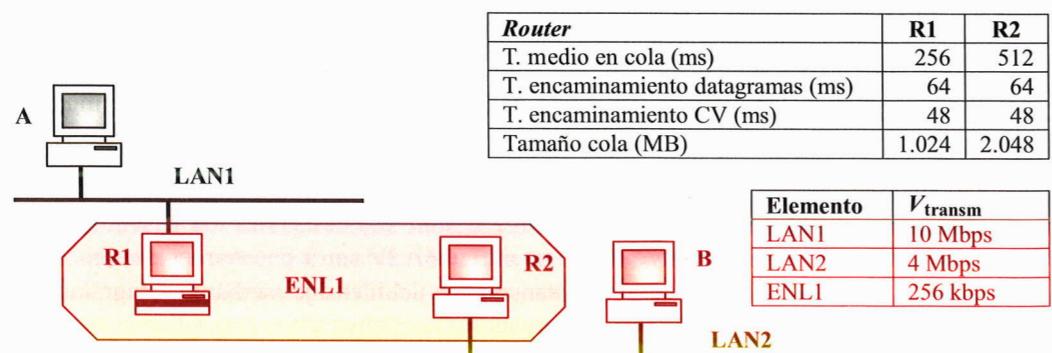
- a) Esquematice gráficamente el proceso involucrado en el envío completo del mensaje M si aceptamos que el tiempo implicado se puede aproximar a través de la siguiente expresión:

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

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)



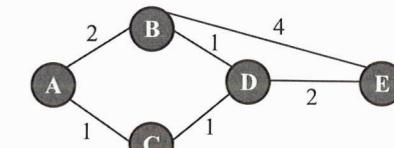
para llegar desde A a B. Los parámetros más relevantes de las redes se detallan en la tabla adjunta. Determine el tamaño mínimo de un archivo para su transmisión mediante circuitos virtuales (CV) que mediante conmutación de paquetes, sabiendo que el tamaño de la cabecera en CV es de 32 bytes, mientras que en paquetes es de 40 bytes.

- NOTA:** Suponga despreciables los tiempos de acceso al medio y de propagación entre 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. Consideremos 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 conmutación 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 conmutación 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?



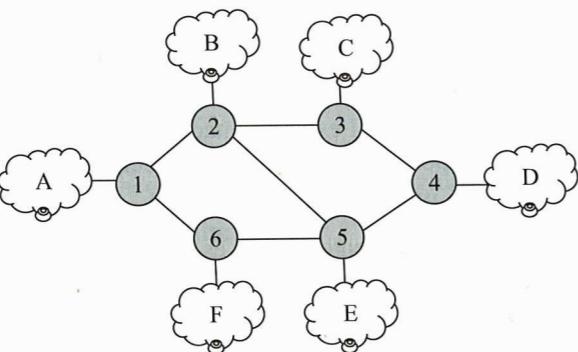
13. Se dispone de la topología de red inferior adjunta, donde se implementa un algoritmo distribuido de actualización de tablas de routing basado en el número de saltos. Supuesto que la actualización comienza para todos los nodos en $t = 15$ s con una periodicidad de 30 s y que las tablas de cada nodo 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:

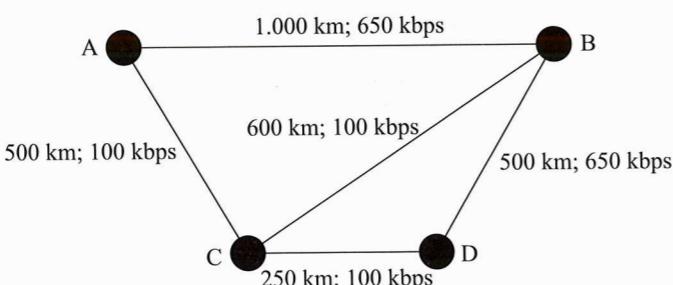
- i. ¿Cuáles serían las tablas de encaminamiento de los nodos justo antes de iniciarse su actualización en $t = 15$ s?
- ii. ¿En qué instante temporal se consigue ahora la estabilidad de las tablas?
- iii. ¿Se modifican estas respecto de las obtenidas en a)?



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:

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

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



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.