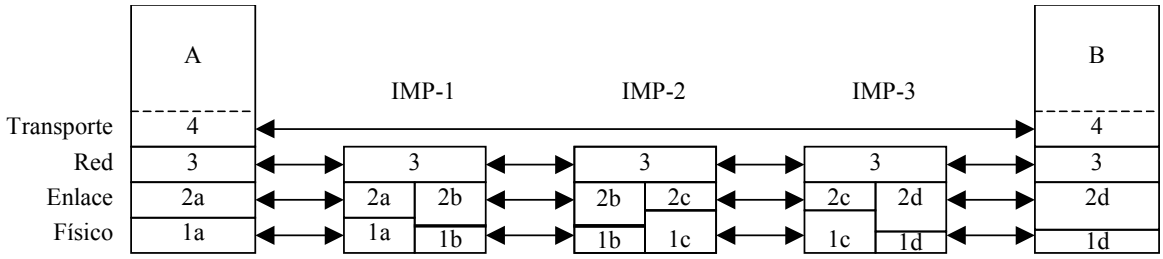
TEMA 5: LAS CAPAS DE RED Y DE TRANSPORTE

INDICE TEMA 5

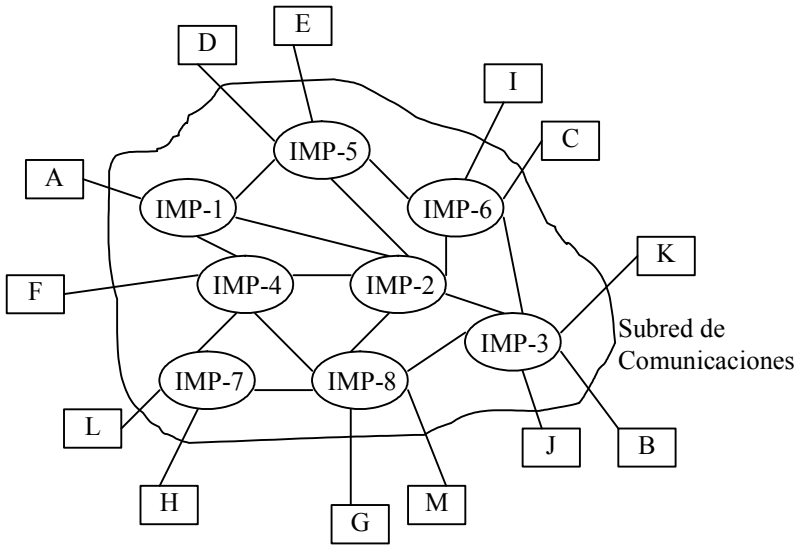
1. LA CAPA DE RED

1.1 Introducción

La capa de red se ocupa de la obtención de paquetes procedentes de la fuente y de encaminarlos durante todo el camino hasta alcanzar su destino. Para alcanzar su destino puede surgir la necesidad de hacer varios saltos en nodos intermedios a lo largo del recorrido. La capa de red es la capa que se ocupa de la transmisión extremo a extremo de la información mediante el diálogo entre las entidades homólogas de la capa de red de cada nodo intermedio que interviene en la comunicación. De esta manera la capa de transporte mantendrá ya directamente un diálogo extremo a extremo con la entidad homóloga del otro extremo de la comunicación (y no con nodos intermedios).



Los nodos intermedios se denominan también IMP (Procesadores de Intercambio de Mensajes), Routers o Encaminadores, aunque hay que puntualizar que en el caso de las dos últimas denominaciones, se supone que tienen capacidad para el encaminamiento de paquetes.



Para poder alcanzar sus objetivos, la capa de red habrá de conocer la topología de la subred de comunicación y seleccionar trayectorias apropiadas dentro de ella, tendrá que evitar la sobrecarga de algunas líneas (si hay otras sin tráfico) y resolverá los problemas derivados del hecho de que fuente y destino puedan residir en redes diferentes.

1.2 Problemas de diseño de la capa de red

Vamos a estudiar los puntos a considerar por todo diseñador de la capa de red: servicios proporcionados a la capa de transporte, encaminamiento de paquetes a través de la subred, control de congestión y conexión de múltiples redes entre sí.

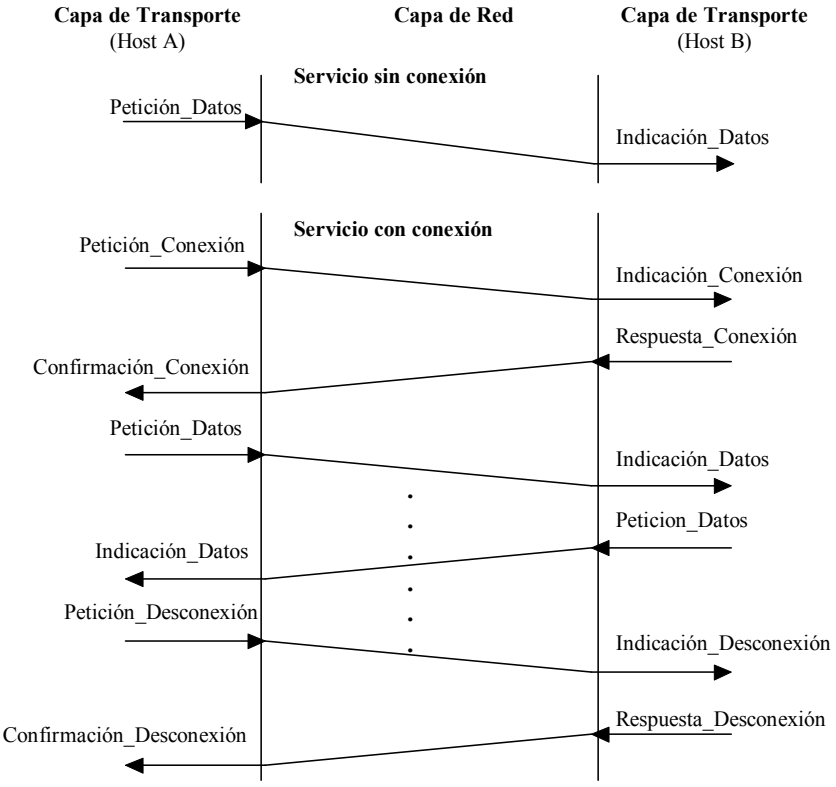
1.2.1 Servicios proporcionados a la capa de transporte

Debido a que en algunas redes de área extensa (por ejemplo X.25) la capa de red opera en los IMP y la capa de transporte opera en los equipos de los usuarios, los límites entre la capa de red y la de transporte en estas redes coincide con el límite entre la subred pública de transmisión de datos y el equipo del usuario. Por tanto, los servicios proporcionados por la capa de red definen los servicios proporcionados por la subred.

En un principio, la ISO sólo aceptó un servicio de red orientado a conexión, principalmente debido a que las compañías proveedoras de servicios portadores necesitaban algún mecanismo que les permitiese calcular los cargos por tiempo de conexión. Sin embargo, los que estaban a favor de un servicio sin conexión, como el de las redes IP, siguieron luchando, hasta que la ISO tuvo que aceptar ambos tipos de servicio.

Este problema surgió en diferentes capas de la arquitectura OSI, por lo que nos encontramos con los dos tipos de servicio en distintos niveles. Más aún, es posible que servicios orientados a conexión de una capa (por ejemplo, la de red) estén soportados por servicios sin conexión en capas inferiores (la de enlace) y viceversa.

En la figura siguiente se muestra un ejemplo de la utilización de las primitivas básicas de servicios sin conexión y orientados a conexión de la capa de red.



La mayor parte de las primitivas tienen parámetros. Por ejemplo, para establecer una conexión, se usa la primitiva *Petición_Conexión*. Esta primitiva especifica la dirección de red a la que se quiere conectar y la dirección de red del que hace la llamada. También

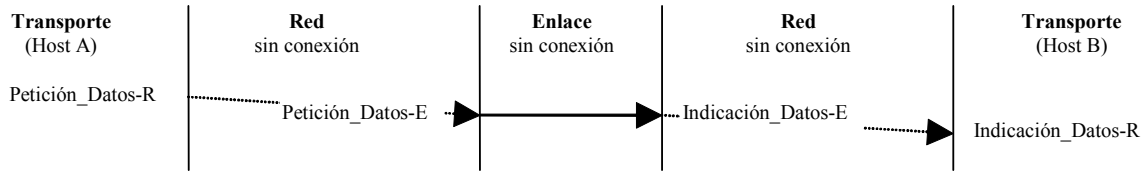
contiene otros parámetros que se utilizan para solicitar servicios adicionales: normalmente estos servicios son *negociados* entre las dos partes. Un parámetro adicional es el de la calidad del servicio proporcionado por la conexión. Habrán de especificarse unos mínimos de calidad aceptables por la entidad que solicita la conexión para que esta se realice con éxito, así como la calidad que realmente se desea. Entre los requisitos de calidad están cosas como el retardo, la tasa de error, el coste, etc.

El resto de primitivas, en una gran parte, tiene también parámetros que influyen en el comportamiento y la funcionalidad del servicio utilizado.

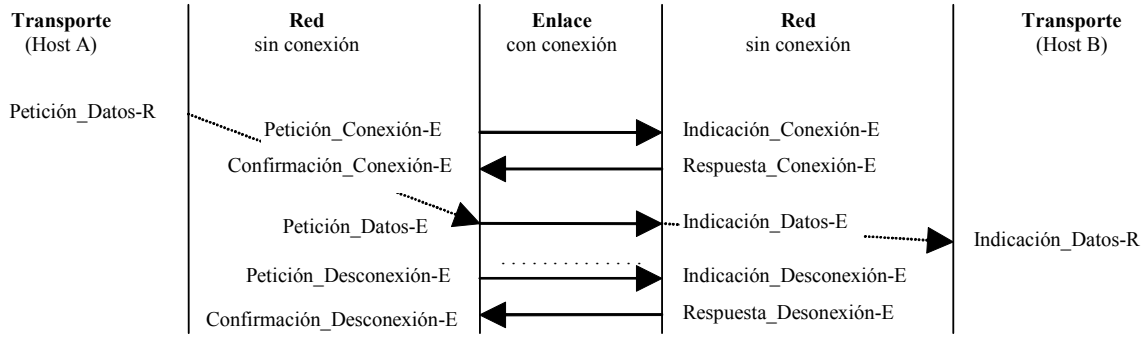
En cuanto a las posibles combinaciones entre los servicios proporcionados por las capas de Red y de Enlace, las siguientes figuras representan ejemplos de cada una de ellas.

a) Una **capa de red sin servicio de conexión** sobre una **capa de enlace sin servicio de conexión**, prácticamente convierte peticiones de envío de paquetes de datos provenientes de la capa de transporte en peticiones de envío de una o varias tramas a través de la línea de enlace. Ninguna de las dos capas asegura que los datos lleguen a su destino, que no existan duplicados, ni el correcto orden de los paquetes en la capa de red o de las tramas en la de enlace.

a)



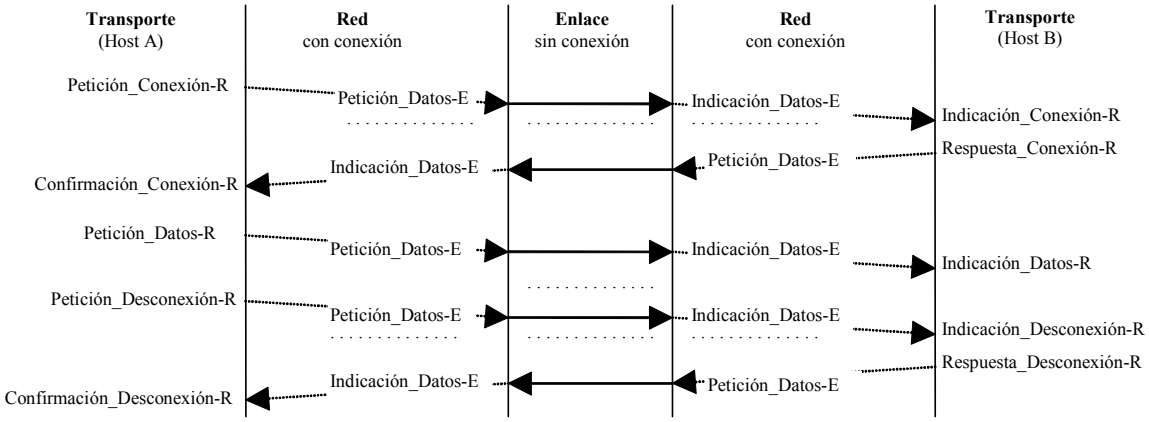
b)



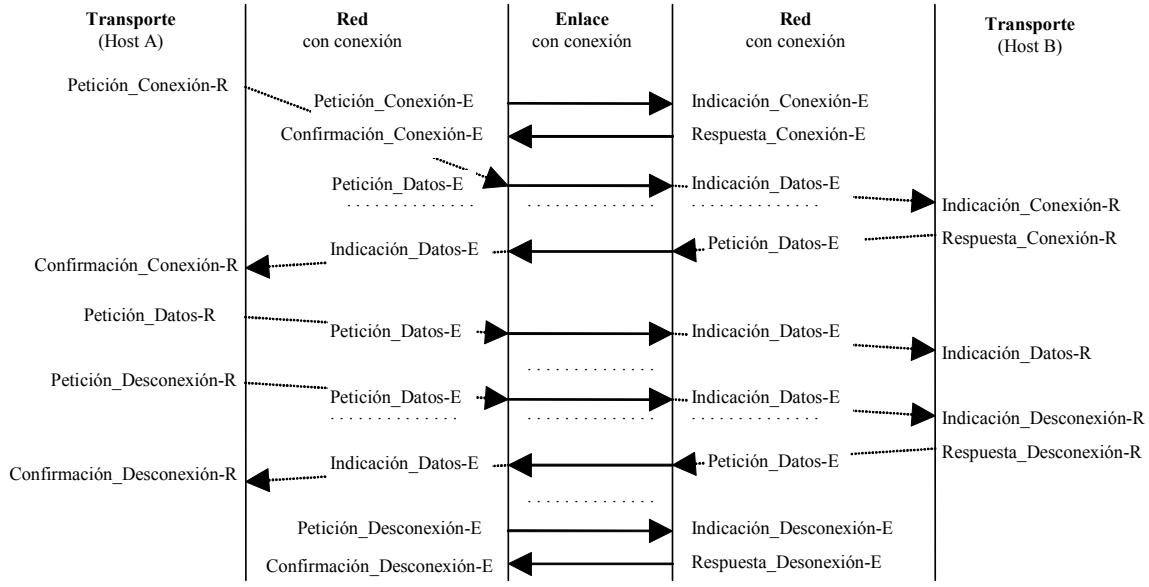
b) Una **capa de red sin servicio de conexión** trabajando sobre una **capa de enlace con servicio de conexión**, puede fiarse de esta en cuanto a que las tramas van a llegar con seguridad, sin duplicados y en orden, a pesar de que las líneas físicas de transmisión no sean muy fiables. Las funciones de esa sofisticada capa de enlace recuperaran las tramas dañadas, eliminan duplicados, etc. para hacer que la línea sea fiable. Sin embargo la capa de red no asegura esto a la de transporte. Paquetes de datos, pueden perderse, duplicarse o cambiar de orden por el camino, en los saltos entre routers a través de otras capas de enlace no tan fiables o por fallos de los propios routers. En estos casos resulta bastante frustrante que el nivel de fiabilidad obtenido mediante funciones complejas de la capa de enlace, se pierda en la capa de red. Además, aunque lo más común es que una conexión de enlace dé servicio a muchos paquetes de la capa de red, es posible que se den situaciones en que el envío de cada paquete requiera el establecimiento, envío y liberación de una conexión de enlace, lo que da lugar a un rendimiento poco eficiente.

c) Una **capa de red con servicio de conexión** trabajando sobre **una capa de enlace sin servicio de conexión**, tiene la laboriosa tarea de asegurar una conexión fiable a la capa de transporte, a pesar de que la capa de enlace no sea fiable. Por lo tanto deberá de implementar mecanismos que le permitan almacenar paquetes para su posible retransmisión, asegurarse de la correcta recepción de los mismos por la entidad homóloga de la capa de red en el siguiente nodo de la red, descartar posibles duplicados, mantener la correcta secuencia de los paquetes, etc. Si no es capaz de llevar a cabo esta labor para una determinada conexión, lo habitual es que se aborte la conexión y se notifique el error a la capa de transporte.

c)



d)



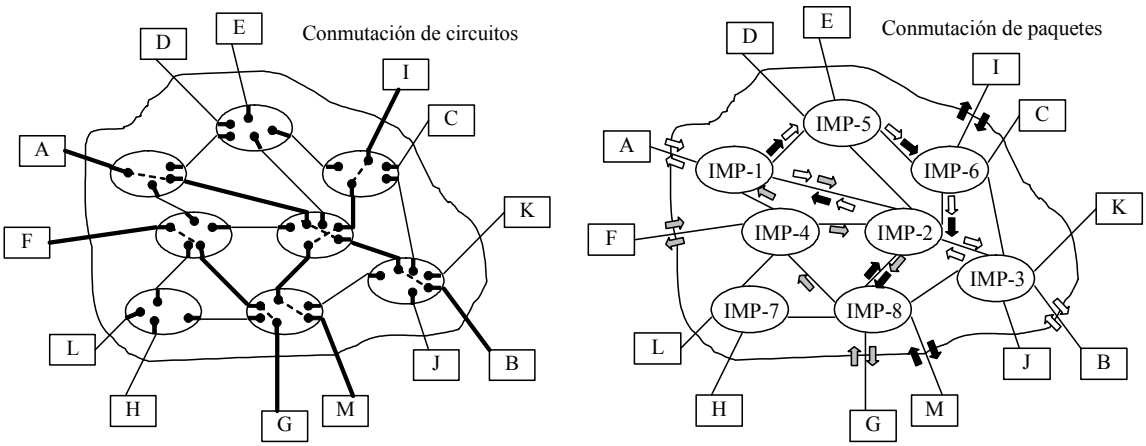
d) Una **capa de red con servicio de conexión** trabajando sobre **una capa de enlace con servicio de conexión**, puede tener una implementación muy sencilla si simplemente se "mapeasen" las peticiones que hace la capa de transporte en peticiones análogas sobre la capa de enlace. Sin embargo esto no suele ser posible ya que una conexión de red necesita normalmente de la cooperación de nodos intermedios que tienen que trabajar sobre capas de enlace que tienen otras características. Lo habitual es que la conexión de enlace exista incluso previamente a la llegada de la petición de conexión procedente de la capa de transporte, y que exista más allá de la desaparición de esta para dar servicio a

futuras conexiones. En todo caso suele suponer que dos capas implementan funciones redundantes para mantener la fiabilidad de las conexiones, cuando bastaría que solo lo hiciese la capa de red.

1.2.2 Organización interna de la capa de red

Una vez vistos los dos tipos de servicio que ofrece la capa de red, hay que pararse a ver cómo trabaja internamente.

En primer lugar podemos distinguir entre **redes de conmutación de circuitos** y **redes de conmutación de paquetes**, también conocidas como redes de almacenamiento y reenvío (store and fordward). En las primeras, al establecer la comunicación, los canales físicos que unen ambos extremos quedan reservados para uso exclusivo hasta que la conexión se libera, y no es necesario reservar recursos en los nodos intermedios para el almacenamiento temporal de la información. En el caso de redes de conmutación de paquetes, cada nodo intermedio recibe mensajes en forma de paquetes de datos y los almacena hasta que los reenvía hacia su destino final o a otro nodo intermedio.

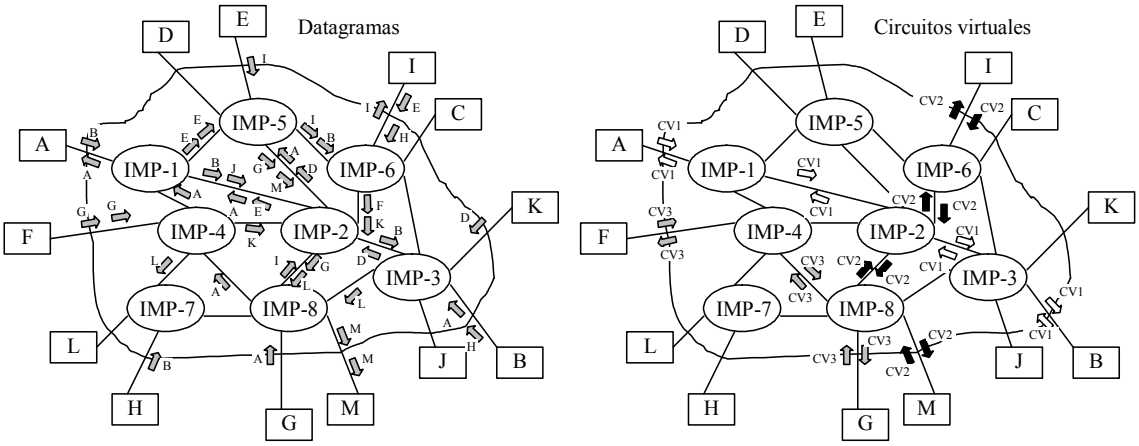


En las redes para transmisión de datos se suele optar por la solución de la conmutación de paquetes, ya que la reserva de un canal físico para la transmisión de datos, que suele ser un proceso que se produce a ráfagas, hace que la capacidad del canal físico se aproveche muy poco. Además, en el instante que se desean transmitir datos masivamente el flujo máximo está limitado por la capacidad máxima del canal. Por ello suele ser un esquema más adecuado para circuitos analógicos de voz, vídeo, etc. Por el contrario, mediante la conmutación de paquetes se comparten las capacidades de los distintos circuitos físicos entre comunicaciones simultáneas consiguiendo un mejor aprovechamiento.

Centrándose en la conmutación de paquetes, existen dos filosofías para la organización interna de la subred: como una red de **datagramas** (denominados así por analogía con los telegramas o el sistema postal) o mediante **circuitos virtuales** (denominados así por su analogía con los circuitos físicos establecidos por el sistema telefónico con redes de conmutación de circuitos).

Hay que señalar que el servicio ofrecido (orientado a conexión o sin conexión) es un tema independiente de la estructura de la subred (de circuitos virtuales o de datagramas). Teóricamente las cuatro combinaciones son posibles, aunque algunas sean más lógicas que

otras. Es más lógico y común que una red con servicio sin conexiones se estructure internamente como una red de datagramas y que una red con servicio orientado a conexión se estructure como una red de circuitos virtuales.



Los circuitos virtuales se utilizar normalmente en subredes cuyo servicio principal está orientado a conexión. La idea principal de los circuitos virtuales es evitar la toma de decisiones de encaminamiento para cada paquete transmitido. En lugar de esto, cuando se establece una conexión, se selecciona una ruta que va desde la máquina origen hasta la máquina destino, y se utiliza dicha ruta para todo el tráfico que circule por la conexión, como en el sistema telefónico. Durante el establecimiento de la ruta, los IMP que van a intervenir en la misma se intercambian mensajes para actualizar sus tablas de circuitos virtuales y reservar recursos (memoria, buffers, etc.) para la conexión. Cuando se libera la conexión, se desecha el circuito virtual, se borran sus entradas de las tablas de encaminamiento y se liberan los recursos. El coste de estos recursos hace que cada IMP pueda soportar un número limitado de conexiones. Si este número se ha alcanzado los siguientes intentos de conexión serán rechazados hasta que se liberen recursos de alguna conexión.

En cambio, con una subred de datagramas, no se determina la ruta anticipadamente, aún cuando el servicio esté orientado a conexión. Cada paquete se encamina independientemente, por lo que paquetes sucesivos podrán viajar por rutas diferentes. A pesar de que tienen que realizar más trabajo, las subredes de datagramas son más robustas y se adaptan mejor a los fallos y la congestión que las subredes de circuitos virtuales.

1.2.2.1 Encaminamiento en redes de circuitos virtuales

Si los paquetes que circulan por un circuito virtual dado siguen siempre la misma ruta a través de la subred, cada IMP debería recordar hacia dónde expedir paquetes para cada uno de los circuitos virtuales abiertos que pasen a través de él. Cada IMP deberá mantener una tabla, con una entrada por cada circuito virtual abierto. Cada paquete que circule por la subred, deberá contener un campo con el número de circuito virtual, además del resto de campos. Cuando el paquete llegue a un IMP, éste conocerá la línea por la que llegó, así como el número del circuito virtual. Con esta información, se reexpedirá el paquete al IMP apropiado.

Cuando se finaliza la utilización de un circuito virtual, habrá que indicar este hecho para que los IMP puedan realizar tareas de actualización en sus tablas.

1.2.2.2 Encaminamiento en redes de datagramas

En el caso de que utilicemos datagramas, los IMP no necesitan almacenar tablas con los circuitos virtuales, ya que éstos no existen. En su lugar, almacenarán una tabla que indica qué salida deben utilizar para cada uno de los posibles IMP destinatarios. Estas tablas también son necesarias cuando se utilizan los circuitos virtuales, para determinar la ruta empleada por el paquete durante el establecimiento del circuito.

Cada datagrama deberá contener la dirección completa del destinatario. Cuando llega un paquete, el IMP busca una línea de salida y lo reexpide a través de ella.

1.2.2.3 Comparación de circuitos virtuales y datagramas en el interior de la subred

Los circuitos virtuales y los datagramas tienen sus ventajas y sus inconvenientes. Vamos a ver los dos aspectos de cada uno de ellos. En la subred, la discusión circuitos virtuales frente a datagramas se fundamenta en el equilibrio entre el espacio de memoria que se consume y el ancho de banda que se logra.

Los circuitos virtuales utilizan números de circuito en lugar de direcciones completas para identificar el origen y el destino de la comunicación. Si los paquetes tienden a ser muy pequeños, el hecho de tener que incorporar en el paquete direcciones completas el lugar de identificadores de circuito virtual, puede representar una sobrecarga significativa, y por tanto, un bajo aprovechamiento del ancho de banda disponible.

Para operaciones relacionadas con el proceso de negociaciones (por ejemplo pagos con tarjetas de crédito), la sobrecarga que conlleva el establecimiento y finalización de un circuito virtual puede desaconsejar su uso. Si se espera que la mayor parte del tráfico sea de este tipo (con el intercambio de pocos datos), tiene poco sentido utilizar circuitos virtuales.

Los circuitos virtuales también tienen un problema de vulnerabilidad. Si por ejemplo falla un IMP, todos los circuitos virtuales que pasan por él tendrán que ser abortados. En cambio, si se usan datagramas y cae el IMP, sólo sufrirán problemas aquellos usuarios cuyos paquetes estaban en la cola de espera del IMP en ese momento. El uso de datagramas también permite balancear el tráfico de la subred gracias a que las rutas se pueden modificar a mitad de una conexión.

1.2.3 Encaminamiento

La función real de la capa de red consiste en el encaminamiento de paquetes, desde la máquina origen hasta la máquina destino. En la mayoría de las subredes, los paquetes necesitarán realizar múltiples saltos para terminar el viaje. Los algoritmos que seleccionan las rutas y las estructuras de datos que utilizan representan una de las áreas principales del diseño de la capa de red.

El algoritmo de encaminamiento es aquella parte del software correspondiente a la capa de red que es responsable de decidir sobre qué línea de salida se deberá transmitir un paquete que llega. Si la subred utiliza internamente circuitos virtuales, la decisión de encaminamiento se toma durante el establecimiento del circuito virtual y luego se mantiene fija para el resto de los paquetes que utilizan ese circuito virtual. Si la subred usa datagramas, la decisión se toma cada vez que llega un paquete y de forma independiente para cada uno de ellos aunque lleven el mismo destino.

Independientemente del momento en que se tome la decisión, existen ciertas propiedades deseables para todo algoritmo de encaminamiento: corrección, simplicidad, robustez, estabilidad, justicia y optimalidad.

Los algoritmos de encaminamiento se pueden agrupar en dos clases principales:

a) **Algoritmos no adaptativos**: no basan sus decisiones de encaminamiento en mediciones ni estimaciones del tráfico o la topología actuales de la red; más bien, la elección de la ruta a utilizar para ir de la i a la j (para toda i y j) se determina anticipadamente, fuera de línea, y se carga en los IMP cuando la red se arranca. A este procedimiento se le denomina en ocasiones encaminamiento **estático**.

b) **Algoritmos adaptativos**: intentan cambiar sus decisiones de encaminamiento para reflejar los cambios de topología y tráfico actuales. Existen tres familias de algoritmos adaptativos. Los algoritmos **centralizados** utilizan información recogida en toda la subred para intentar tomar decisiones óptimas. Los algoritmos **aislados** operan de forma separada en cada IMP y sólo utilizan la información que está disponible en él, como la longitud de las colas de espera. Los algoritmos **distribuidos** utilizan una combinación de información local y global.

1.2.3.1 Encaminamiento centralizado

Cuando se utiliza un encaminamiento centralizado, en alguna parte de la red hay un CCE (Centro de Control de Encaminamiento). Periódicamente, cada IMP transmite la información de su estado al CCE (por ejemplo, la lista de sus vecinos activos, las longitudes actuales de las colas de espera, cantidad de tráfico procesado desde el último informe de estado, etc.) El CCE recoge toda esta información, y con base en el conocimiento de la red completa, calcula las rutas óptimas de todos los IMP a cada uno de los IMP restantes. A partir de esta información, construirá nuevas tablas de encaminamiento que distribuirá a todos los IMP.

Una ventaja del encaminamiento centralizado es que los IMP se despreocupan de calcular el encaminamiento. Por contra, tiene importantes desventajas. Si la subred se tiene que adaptar a un tráfico variable, el cálculo del encaminamiento se tendrá que realizar con bastante frecuencia. Para una red grande, este cálculo llevará unos segundos. Si el propósito del cambio es adaptarlo a los cambios en la topología de la red y no tanto a cambios de tráfico, no habría demasiados problemas, dependiendo de lo estable que fuese la topología.

Además, si el CCE falla, la subred estará de pronto en una situación muy problemática. Una posibilidad es tener una máquina de respaldo, pero esto conlleva desperdiciar un ordenador.

Otro problema es el relacionado con la distribución de las tablas de encaminamiento. Los IMP próximos al CCE recibirán primero sus tablas nuevas y podrán cambiar a las nuevas rutas antes que los IMP localizados más lejos hayan recibido las suyas. Bajo estas circunstancias pueden presentarse inconsistencias que lleven a que ciertos paquetes se retarden. Entre estos paquetes puede haber alguna tabla de encaminamiento para IMP distantes, con lo que el problema se realimenta a si mismo.

Un último problema es la fuerte concentración de tráfico encaminado sobre las líneas que conducen al CCE.

1.2.3.2 Encaminamiento aislado

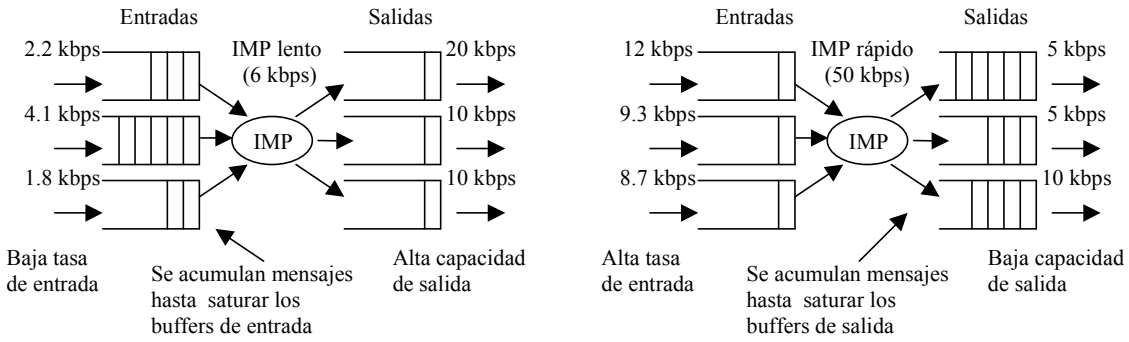
Las decisiones de encaminamiento son tomadas por los IMP basándose únicamente en la información que ellos mismos hayan reunido. No intercambian, por tanto, información con otros IMP. Sin embargo, tratan de adaptarse a los cambios de topología y tráfico que se presenten. A estos algoritmos se les conoce comúnmente como algoritmos de encaminamiento adaptables aislados.

1.2.3.3 Encaminamiento distribuido

En este tipo de algoritmos, cada IMP intercambia periódicamente información de encaminamiento explícito con cada uno de sus vecinos, con el fin de mantener una tabla de encaminamiento en donde figura la línea de salida más adecuada hacia cada IMP, y alguna estimación del tiempo la distancia hacia él. Estos algoritmos son bastante efectivos, pero presentan el problema del aumento artificial del tráfico para que los IMP se intercambien la información. Además, habría que decidir en qué momento se realizan dichos intercambios.

1.2.4 Congestión

Cuando tenemos muchos paquetes en la subred, el rendimiento se degrada. Esta situación se conoce con el nombre de congestión. La congestión puede estar producida por varios factores. Si los IMP son muy lentos para efectuar las distintas tareas que tienen asignadas, las colas pueden crecer, independientemente de que las líneas de transmisión tengan suficiente capacidad. Por otra parte, aún cuando la CPU del IMP fuese infinitamente rápida, el crecimiento de las colas de espera surgirá, cada vez que la velocidad del tráfico de entrada exceda la capacidad de las líneas de salida.



La congestión tiende a realimentarse, volviéndose todavía peor. Si un IMP no tiene memorias temporales desocupadas, deberá ignorar los nuevos paquetes que llegan. Cuando se desecha un paquete, al IMP que lo envió le vencerá un temporizador, con lo que retransmitirá el paquete hasta que reciba un acuse de recibo. Por tanto, deberá almacenar el paquete hasta que llegue correctamente al siguiente nodo, cuando en condiciones normales ya habría liberado el espacio que ocupaba. De esta manera la congestión se va extendiendo entre los IMP.

Finalmente, es importante señalar la diferencia existente entre el control de la congestión y el control de flujo. El control de la congestión tiene que ver con la seguridad de que la subred sea capaz de transportar el tráfico ofrecido. A diferencia de esto, el control de flujo se refiere al tráfico punto a punto entre un emisor y un receptor dados. Su trabajo consiste en asegurar que, en caso de que haya un emisor muy rápido, éste no inunde con información a un receptor que trabaja de forma más lenta.

1.2.5 Interconexión de redes

El problema del encaminamiento, es aún más complejo si se interconectan redes y/o circuitos de enlace de datos que no utilizan los mismos protocolos. El hecho de tener diferentes protocolos implica diferentes formatos para los paquetes, procedimientos de control de flujo, reglas de acuse de recibo, etc. Por lo tanto, ante un paso de una red a otra, será necesaria la realización de conversiones. Existen un gran número de redes, circuitos de enlace de datos y protocolos diferentes.

En el modelo OSI, la interconexión de redes se realiza en la capa de red. La capa de red, siempre que sea necesario, se puede dividir en tres subcapas que son de abajo a arriba: la subcapa de acceso a la subred, la subcapa de mejora de la subred y la subcapa de interconexión de redes. Como subred se entiende en este caso el conjunto formado por la capa de enlace y la capa física de la red o circuito de enlace de datos.

El propósito de la **subcapa de acceso a la subred** consiste en soportar el protocolo de la capa de red para la subred que específicamente se esté utilizando, es decir, hacer que las unidades de datos de la capa de red puedan ser transmitidas correctamente hasta el siguiente IMP o destinatario a través de la subred. **La subcapa de mejora de la subred** está diseñada para armonizar las subredes que ofrecen diferentes servicios y conseguir que el nivel de servicios proporcionados por todas las subredes sea el mismo. La función de la **subcapa de interconexión de redes** es el encaminamiento de las unidades de datos hacia cualquiera de las subredes a las que está conectado el IMP que serán, para esta subcapa, similares en cuanto a sus servicios gracias a las subcapas inferiores. Cuando un paquete llega a un IMP se lleva a la subcapa de interconexión de redes, que lo revisa y decide si se reexpide y hacia qué subred.

A parte de los IMP o encaminadores, existen otros tres tipos de equipos que pueden funcionar como retransmisores de información en una arquitectura de comunicaciones. En función de la capa en la que realicen su trabajo y del servicio que presten reciben diferentes denominaciones:

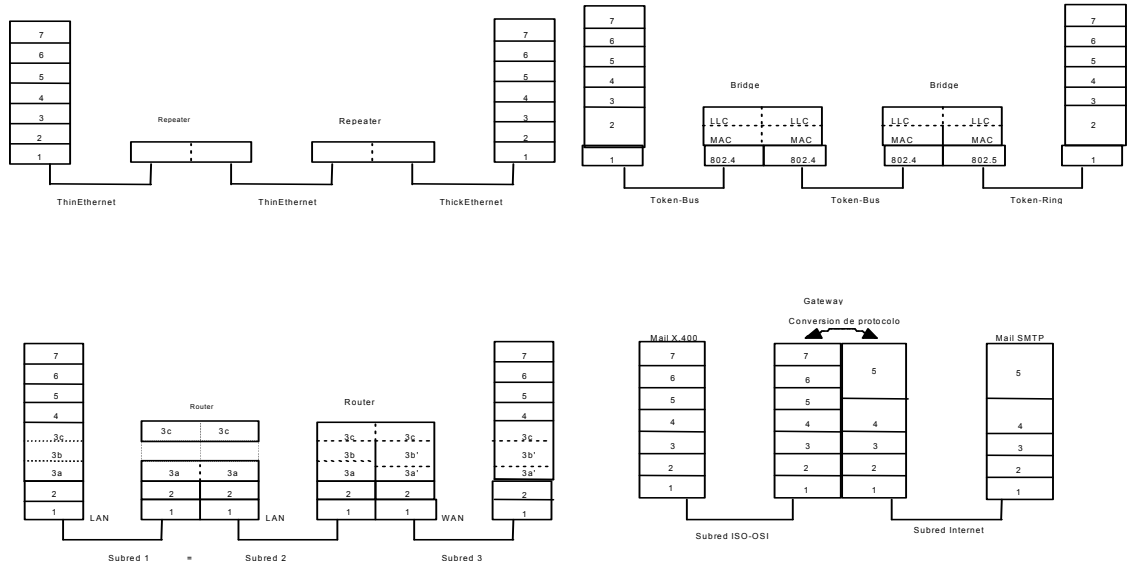
a) **Repetidores (repeaters)**: Se encuentran en la capa 1 (nivel físico). Se limitan a copiar los bits individualmente entre distintos segmentos de cable. Sólo amplifican señales eléctricas, y son necesarios para proporcionar corriente que permita excitar cables de longitud considerable.

b) **Puentes (bridges)**: Se encuentran en la capa 2 (nivel de enlace). Permiten almacenar y reexpedir tramas entre diferentes redes LAN. Un puente acepta una trama completa y la pasa a la capa de enlace, donde se comprueba el código de redundancia e incluso las direcciones físicas que van en las tramas, pudiendo realizar un filtrado de las tramas o funciones básicas de encaminamiento. Entonces, la trama se transmite a la capa física para que se reexpida hacia una subred diferente.

c) **Encaminadores (routers)**: Se sitúan en la capa 3 (nivel de red) y son los que se describen en este apartado como elementos para la interconexión de redes (IMP). Almacenan y reexpiden paquetes entre subredes. En algunos casos y en muchas configuraciones de sistemas se denomina a estos dispositivos Pasarelas (Gateway) o Puertas de Acceso. Cuál es la denominación más correcta es tema de discusión.

d) **Pasarelas (gateway)**: También denominados Convertidores de Protocolo. Se sitúan a partir de la capa 4 y proporcionan interconexión en capas superiores. Un ejemplo de convertidor de protocolo es un retransmisor que traduce el protocolo

de transporte del modelo OSI, al protocolo utilizado en la interconexión de redes ARPA (TCP). Otro puede ser el caso de los equipos encargados de hacer de pasarela entre usuarios de correo electrónico de aplicaciones que trabajan sobre arquitecturas diferentes, como SMTP (Simple Mail Transfer Protocol) sobre TCP/IP, X.400 sobre OSI o VMSMail sobre DECNET.



1.3 Algoritmos de Encaminamiento

1.3.1 Encaminamiento por el camino más corto

Normalmente se implementa como un algoritmo centralizado, es decir un nodo calcula las rutas más óptimas y las transmite a todos los demás. La operación puede realizarse offline antes de poner en marcha la red y manteniendo luego fijas las rutas haciendo que el encaminamiento sea estático, o bien, se recalcula periódicamente o cuando la red cambia para que sea el encaminamiento sea adaptativo. La idea consiste en construir un grafo de la subred, en el que cada nodo represente un IMP, y cada arco equivalga a una línea de comunicación. Para escoger una ruta entre un par de IMP dados, el algoritmo sólo determina el camino más corto que existe entre ellos.

Lo único que nos queda por definir es el concepto de camino más corto. Tenemos diferentes posibilidades:

a) El de menor número de saltos (nodos atravesados)

b) El de menor distancia en km.

c) El de menor retardo promedio de espera en cola y de transmisión.

d) El de mayor ancho de banda, etc.

En el caso general, las etiquetas de los arcos se podrían calcular como una función de la distancia, ancho de banda, promedio de tráfico, coste de comunicación, longitud promedio de la cola de espera, retardo medio, etc.

Una vez etiquetados los arcos del grafo, sólo queda aplicar algún algoritmo de cálculo de caminos mínimos, como por ejemplo el desarrollado por Dijkstra.

1.3.2 Algoritmo de la patata caliente

Es un caso de encaminamiento aislado. En el momento en que llega un paquete, el IMP trata de deshacerse de él tan rápido como le sea posible, poniéndolo en la cola de espera de salida más corta. Se pueden obtener variantes de esta idea combinándola con el encaminamiento estático: cuando llega un paquete, el algoritmo de encaminamiento toma en cuenta tanto el peso estático de las líneas como las longitudes de las colas de espera.

1.3.3 Algoritmo de aprendizaje hacia atrás

Otro caso de encaminamiento aislado que consiste en incluir la identidad del IMP origen en cada paquete, junto con un contador que se incrementa cada salto. Si un IMP ve llegar un paquete en la línea k, procedente del IMP H, con la cuenta de 4 saltos, sabe que H no puede estar más lejos de cuatro saltos sobre la citada línea. Si su mejor ruta actual hacia H se estima en más de cuatro saltos, marca la línea k como la elegida para el tráfico hacia H y registra la distancia estimada en cuatro saltos. Pasado cierto tiempo, cada IMP descubrirá el camino más corto hacia cualquier otro IMP.

1.3.4 Inundación

Es un caso extremo del encaminamiento aislado, en el que cada paquete que llega se transmite por todas las líneas de salida, excepto por la que llegó. Con la inundación se genera un número considerable de paquetes duplicados; de hecho, un número infinito, a no ser que se tome alguna medida. Una de las medidas es tener un contador de saltos en la cabecera de los paquetes, el cual se decrementa con cada salto, desechándose el paquete cuando el contador llegue a cero. Idealmente, el contador habrá de inicializarse con un valor correspondiente a la distancia entre origen y destino. Si el emisor no conoce la distancia, puede iniciar el contador con el valor del peor caso, es decir, el valor del diámetro completo de la subred.

En algunas aplicaciones, la inundación no resulta ser muy práctica, pero sí tiene algunos usos importantes. Por ejemplo, en aplicaciones militares la robustez que ofrece este mecanismo es algo deseable ante el hecho de que varios IMP puedan ser destruidos. En aplicaciones de bases de datos distribuidas, algunas veces es necesario actualizar todas la bases de datos concurrentemente, en cuyo caso, la inundación puede ser de gran utilidad.

Una variante de la inundación que es un poco más práctica, es la inundación selectiva. En este algoritmo, los IMP transmiten los paquetes sólo por aquellas líneas que van en la dirección correcta.

1.3.5 Encaminamiento jerárquico

A medida que crece el tamaño de la red, las tablas de encaminamiento de los IMP también crecen en forma proporcional. No sólo se produce un aumento de la cantidad de memoria consumida por la tabla, sino que también aumenta el tiempo necesario para explorarla. También se hace necesario un mayor ancho de banda, para poder transmitir los informes de estado que se guardan. Para reducir estos problemas, se recurre al encaminamiento jerárquico.

Cuando se utiliza encaminamiento jerárquico, los IMP se dividen en regiones, en las cuales cada uno de los IMP conoce todos los detalles sobre la manera de encaminar los paquetes para alcanzar sus respectivos destinos dentro de su propia región, pero desconocen la estructura interna de otras regiones. El número de niveles en la jerarquía aumentará a medida que aumenta el tamaño de la red.

En este caso, cada IMP tiene una tabla con entradas disponibles para cada IMP de su misma región. Los IMP de otras regiones se concentran en un único IMP local que es el que hace de enlace hacia dicha región. Las estrategias de encaminamiento dentro de cada región y a nivel global pueden ser distintas (estáticas, adaptativas, centralizadas, etc …).

El precio a pagar es un posible aumento en la longitud del camino al tener que encaminar todo el camino hacia una región a través de un determinado IMP.

1.4 Algoritmos de control de la congestión.

Vamos a ver tres estrategias para el control de la congestión (existen más). Estas estrategias se basan respectivamente en la asignación de recursos de forma anticipada, en desechar paquetes cuando no se pueden procesar y en restringir el número total de paquetes en la subred

1.4.1 Preasignación de buffers

Si se utilizan circuitos virtuales, es posible resolver el problema de la congestión de la siguiente manera. Cuando se establece el circuito virtual, se van actualizando las tablas de cada uno de los IMP y se reserva espacio para los buffers del circuito. Una pequeña modificación del algoritmo de establecimiento podría hacer que cuando uno de los paquetes de solicitud de llamada llega a un IMP y todos los buffers están reservados, se deberá proceder a buscar una ruta alternativa o bien, devolver una señal de red ocupada al extremo que llama (como cuando al intentar una llamada telefónica recibimos el tono o mensaje de red telefónica ocupada que nos impide establecer la llamada).

Al reservar espacio en cada IMP para cada circuito virtual, siempre habrá un lugar para almacenar cualquier paquete que llegue hasta que pueda ser reexpedido. Por ejemplo, sea un protocolo IMP-IMP de parada y espera. Un buffer por circuito virtual por IMP es suficiente para circuitos simplex, y uno por cada dirección, para circuitos dúplex. Cuando llega un paquete, el acuse de recibo (ACK) no se devuelve al IMP transmisor hasta que el paquete haya sido reexpedido. Esto es debido a que el acuse de recibo no sólo significa que se ha recibido un paquete, sino que también se está en condiciones de recibir otro. Si el protocolo IMP-IMP permite múltiples paquetes pendientes de acuse de recibo, cada IMP tendrá que dedicar un grupo completo de buffers equivalente al número máximo de paquetes que pueden estar pendientes de acuse de recibo para cada circuito virtual para poder eliminar completamente el problema de la congestión.

Cuando cada uno de los circuitos virtuales que pasan por un IMP tiene suficiente espacio en buffers dedicado a él, la conmutación de paquetes llega a ser muy parecida a la conmutación de circuitos. En ambos casos, hay una fase previa de establecimiento de la conexión, y también se necesita tener recursos asignados permanentemente, haya o no tráfico. Es imposible que se presente congestión en los circuitos establecidos, ya que todos los recursos necesarios para el tráfico han sido reservados. En ambos casos, hay un uso de recursos potencialmente ineficiente porque los recursos asignados a la conexión que no estén siendo utilizados no pueden ser utilizados por nadie más. Los nuevos intentos de

conexión, cuando todos los recursos están ocupados, deberán esperar a que se liberen recursos cuando finalicen alguna o algunas de las conexiones en curso.

Debido al gran coste que representa tener un conjunto de buffers asignados a un circuito virtual posiblemente inactivo, algunas subredes sólo lo utilizan en aquellos casos en los que es imprescindible tener un retardo muy pequeño y un ancho de banda fijo disponible.

1.4.2 Descarte de paquetes

Con este mecanismo, no se reserva absolutamente nada por adelantado. Si llega un paquete y no existe lugar disponible en el IMP, simplemente se descarta. Si la subred ofrece un servicio sin conexión, no hay nada más que hacer: la congestión se resuelve simplemente mediante el descarte de paquetes. Si la subred ofrece un servicio con conexión, en algún lugar deberá haber una copia del paquete para que se pueda retransmitir después. Una posibilidad consiste en hacer que el IMP que transmitió el paquete descartado espere un tiempo y retransmita el paquete hasta que sea recibido.

Pero no se pueden descartar paquetes alegremente: en el caso de que el paquete que llegue sea un paquete de acuse de recibo, se podría liberar el buffer. Sin embargo, si el IMP no tiene buffers disponibles, no podrá examinar el paquete. La solución consiste en reservar permanentemente un buffer por línea de entrada, con el fin de poder inspeccionar los paquetes que lleguen.

Si la congestión tiene que ser evitada mediante el descarte de paquetes, será necesario tener una regla para indicar cuándo se deberá conservar o descartar un paquete. En ausencia de cualquier regla, una sola línea de salida podría acaparar en un IMP todos los buffers disponibles, dado que se asignan sencillamente según la regla del primero que llega es el primero en ser atendido.

Aunque descartar paquetes es muy sencillo, tiene algunas desventajas: una de las más importantes es el ancho de banda necesario para los duplicados. Un punto muy relacionado con esto es la duración del temporizador de reenvío: si el plazo es muy corto, los duplicados pueden ser generados cuando no se necesiten, empeorando todavía más la congestión. Si es muy largo, los tiempos de transmisión sufrirán las consecuencias.

1.4.3 Control isarrítmico de la congestión

Un planteamiento directo para controlar la congestión es limitar el número de paquetes presentes en la subred. Al método que mantiene constante el número de paquetes que circulan por la subred se le denomina isarrítmico. En este método, existen permisos que circulan por la subred. Siempre que un IMP quiere transmitir un paquetœntregado por el equipo de un usuario, primero debe capturar un permiso y después destruirlo. Cuando el IMP destinatario saca el paquete de la subred, regenera el permiso. Con estas reglas aseguramos que el número de paquetes de la subred nunca excederá del número inicial de permisos.

Este método tiene algunos problemas: aunque asegura que la subred, como un todo, no llegará a congestionarse, no garantiza que un IMP determinado quede de repente abrumado por paquetes.

En segundo lugar, cómo distribuir los permisos no será fácil. Para evitar que un nuevo paquete sufra un gran retardo mientras el IMP local trata de conseguir un permiso,

los permisos deberán estar uniformemente distribuidos, de tal manera que cualquier IMP tenga algunos.

Tercero, si por alguna razón los permisos llegan a ser destruidos (errores de transmisión, mal funcionamiento de un IMP,...), la capacidad de transporte de la red se reducirá para siempre. No hay ninguna manera sencilla de determinar cuántos permisos existen todavía, mientras la red esté funcionando.

2. LA CAPA DE TRANSPORTE

La capa de Transporte ofrece a los usuarios de sus servicios (usuarios o capas superiores) un transporte extremo a extremo de los datos. Este transporte se realiza mediante un protocolo o diálogo también extremo a extremo con la entidad homóloga de la capa de Transporte en el nodo destinatario. Si ese servicio es fiable, la capa de Transporte será responsable del establecimiento, control y liberación de las conexiones de transporte para los usuarios del servicio. Aunque, como se ha visto ya en otras capas, es posible dar un servicio no fiable, sin conexiones.

Cuando se usan conexiones en ambas capas, la capa de Transporte mantiene normalmente una conexión de red para cada una de las conexiones de usuario que están activas. Pero puede multiplexar varias conexiones de usuario sobre una sola de red o una única conexión de usuario sobre varias de red según convenga.

Los datos de la capa de sesión se pasan a la entidad del protocolo de transporte que los encapsula en una o varias unidades de datos del protocolo de transporte (Transport Protocol Data Unit, TPDU). El protocolo de transporte aísla las capas más altas de los detalles relativos a los servicios de comunicación. Pueden definirse tres tipos de calidad de servicios de red:

1. Tipo A: Conexiones con una tasa de errores residuales aceptable (errores que no han sido detectados por la capa de red) y una tasa de señalización de errores aceptable (errores detectados, pero no corregidos, por la capa de red).
2. Tipo B: Conexiones con una tasa de errores residuales aceptable, pero con una tasa de señalización de errores inaceptable.
3. Tipo C: Conexiones con una tasa de errores residuales inaceptable para el usuario del servicio de transporte.

El modelo de referencia ISO/OSI ha definido cinco clases de protocolos de transporte capaces de manejar varios tipos de requisitos de usuario y adaptarse a los tres tipos de redes definidas:

a) Clase 0: Protocolos simples. No mejora el protocolo de red.
b) Clase 1: Protocolos con recuperación básica de errores.
c) Clase 2: Protocolos con multiplexación.
d) Clase 3: Protocolos con recuperación de errores y multiplexación.
e) Clase 4: Protocolos con detección y recuperación de errores.

Las clases 0 y 2 se han pensado para redes tipo A; las clases 1 y 3 se han pensado para redes tipo B; la clase 4 está pensada para redes tipo C.

Los mecanismos del protocolo de transporte variarán pues en función de la calidad del servicio de red. Para simplificar, se van a presentar estos mecanismos a continuación considerando primero la situación más simple, con un servicio de red fiable, para posteriormente dar una idea de la complejidad que puede llegar a tener sobre un servicio de red no fiable.

2.1 Mecanismos sobre un servicio de red fiable

Se supone que el servicio de red puede aceptar mensajes de tamaño arbitrario y los va a enviar en secuencia hacia su destino con una seguridad prácticamente del 100%. Ejemplos de este tipo de servicio los ofrecen la red X.25 o las redes Frame Relay con protocolo de control LAPF. Las cuestiones a considerar en el diseño de la capa de transporte son las cuatro siguientes.

2.1.1 Direccionamiento

Cuando un usuario de la entidad de transporte desea realizar una transferencia de datos con o sin conexión con un usuario de otra entidad de transporte, el usuario de destino debe especificarse con la siguiente información:

a) **Identificación de usuario**: Suele tratarse de un número de puerto con el que la entidad de transporte distingue a sus distintos usuarios.

b) **Identificación de la entidad de transporte**: En ocasiones no es necesario ya que en cada estación sólo hay una entidad de transporte para dar servicio a todos los usuarios. Pero si hay más de una (p.e. TCP y UDP) es necesario especificar cual es la entidad de transporte destinataria de los datos.

c) **Identificación de la estación**: Se denomina normalmente dirección de la estación y en el caso de una red global es una identificación única que distingue a cada estación de las demás.

2.1.2 Multiplexación

Respecto a la interfaz con la capa superior, se implementa una función multiplexación/demultiplexación, es decir, múltiples usuarios usan la misma entidad de transporte en la estación y son distinguidos por números de puerto o puntos de acceso al servicio.

Si se considera conveniente, las funciones de la entidad de transporte pueden incorporar la multiplexación de varias conexiones de usuario sobre una sola conexión de red o de una única conexión de usuario sobre varias conexiones de red simultáneas. Ambos casos son poco frecuentes, sobre todo el último.

2.1.3 Control de flujo

Si bien el control de flujo en la capa de enlace es relativamente sencillo, en la de transporte resulta bastante complejo por dos razones fundamentales:

a) El control de flujo supone la interacción entre los usuarios, la entidad de transporte y el servicio de red.

b) El retardo de transmisión entre entidades de transporte es generalmente grande y, lo que es peor, variable.

En cualquier caso la solución por la que se opta es un mecanismo de ventana, como el utilizado por algunos protocolos de enlace. En este caso existen dos variantes:

a) Ventana de tamaño fijo.

b) Ventana de tamaño variable mediante asignación de créditos.

El primer caso es el ya estudiado para la capa de enlace, donde la ventana es de tamaño fijo y siempre inferior al rango de los números de secuencia que se utilizan para numerar los paquetes. La ventana se desplaza cada vez que llega el acuse de recibo de alguno de los paquetes ya enviados.

En el segundo caso, la ventana se va cerrando a medida que llegan los acuses de recibo desplazándose el extremo inferior pero no el superior de la ventana, a no ser que el receptor envíe créditos para ampliar o reducir la ventana. Este mecanismo se describirá más adelante.

2.1.4 Establecimiento y liberación de la conexión

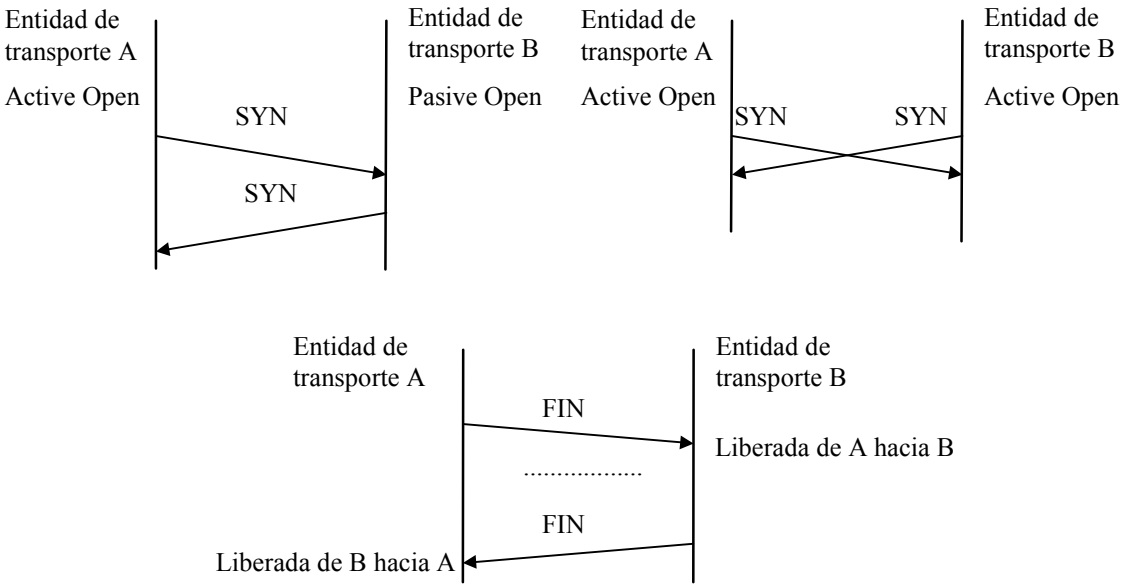
Incluso con servicios fiables de red es necesario establecer y liberar conexiones para ofrecer un servicio orientado a conexión. Los objetivos de establecer una conexión son:

a) Permitir a cada extremo asegurarse de la existencia del otro.

b) Permitir la negociación de parámetros opcionales (tamaño de ventana, calidad del servicio, etc.)

c) Poner en marcha la reserva de recursos para la conexión (espacio en memoria, entradas en tablas de conexiones, etc.)

Con un servicio fiable de red, la conexión se puede establecer con un saludo en ambos sentidos, generalmente denominado *handshake de doble vía*. El nodo que inicia la conexión (en situación *Active Open*) envía un segmento SYN para sincronizar y el otro nodo (que debería estar en situación *Passive Open*) contesta con otro SYN. La conexión ya está establecida y comenzará el intercambio de datos. Una alternativa en que los dos nodos intenten abrir simultáneamente la conexión (ambos en situación *Active* Open) enviando casi simultáneamente un SYN cada uno. El resultado es el mismo, pues cada extremo dará la conexión por abierta una vez que tras su transmisión le llegue el SYN del otro. En cualquiera de los casos un extremo puede rechazar o abortar una conexión con el envío de un segmento RST.



La liberación de la conexión es también muy simple, tras terminar A de enviar todos los segmentos de datos, enviará un segmento FIN, y B ya no esperará más datos de A. Sin

embargo, la conexión permanecerá abierta en el otro sentido por lo que A ha de permanecer a la escucha de los segmentos que B tenga todavía pendientes de enviar hasta que, tras enviar el último, envíe también un segmento FIN.

2.2 Mecanismos sobre un servicio de red no fiable

Se supone que el servicio de red puede perder ocasionalmente segmentos y que debido a los retardos variables del tránsito los segmentos pueden llegar fuera de secuencia. Ejemplos de este tipo de servicio los ofrecen las redes basadas en IP o las redes Frame Relay que usan sólo el núcleo del protocolo de control LAPF. La combinación de inseguridad y no secuenciamiento crea problemas a todos los mecanismos descritos hasta ahora y las soluciones suelen crear nuevos problemas, sobre todo cuando se intenta diseñar un protocolo de transporte orientado a conexión.

2.2.1 Transporte ordenado, retransmisión y detección de duplicados

Para controlar el correcto secuenciamiento de los segmentos es necesario numerarlos secuencialmente. Normalmente se va incrementando en una unidad el número de secuencia de cada segmento, aunque existen otras posibilidades (TCP numera los bytes que van en cada segmento). Además, el receptor deberá confirmar la recepción de los segmentos con ACKs con numeración congruente con los segmentos recibidos, aunque para agilizar la comunicación se hace una confirmación acumulativa que evita enviar un ACK por cada segmento. Así, la recepción de un ACK con un determinado número de secuencia confirma la recepción del segmento con ese número de secuencia y todos los anteriores.

Dos eventos requieren la **retransmisión** de un segmento:

* El segmento puede llegar dañado al destino por lo que será descartado por la entidad de transporte receptora al comprobar el código de detección de errores.
* El segmento no llega al destino.

En cualquiera de los dos casos no se producirá el envío del ACK, por lo que el emisor deberá tener asociado un *temporizador de retransmisión* a cada segmento transmitido. En el caso de que este temporizador expire se retransmitirá el segmento. ¿Pero qué valor se debe establecer en el temporizador? Debería ser algo mayor que el retardo de ida y vuelta (envío del segmento y recepción del ACK). Sin embargo, este retardo es muy variable en una red de interconexión.

Existen lógicamente dos estrategias posibles: utilizar un temporizador con **valor fijo** o un **esquema adaptativo**. El primer caso no se adaptará a situaciones cambiantes de la red si no que se fijará basándose en un comportamiento típico de la misma. Si se elige muy grande, el protocolo será muy lento para dar respuesta a las pérdidas de segmentos y, si es muy pequeño, habrá retransmisiones innecesarias que cargan de trabajo a la red y, en caso de congestión, provocarán nuevas retransmisiones que tenderán a aumentar la congestión de la red. El esquema adaptativo tiene sus propios problemas. Si se fijara, por ejemplo el temporizador en función de la media de los retardos, tendremos problemas para su cálculo ya que a veces las confirmaciones son acumulativas (no se responde inmediatamente a cada segmento) y no se sabe que confirmaciones lo son de segmentos retransmitidos. Además, un valor medio no se adaptará a los cambios rápidos de las condiciones de la red.

Existen otros temporizadores que se utilizan en la entidad de transporte, recogidos en la siguiente tabla, y que serán comentados a lo largo del capítulo.

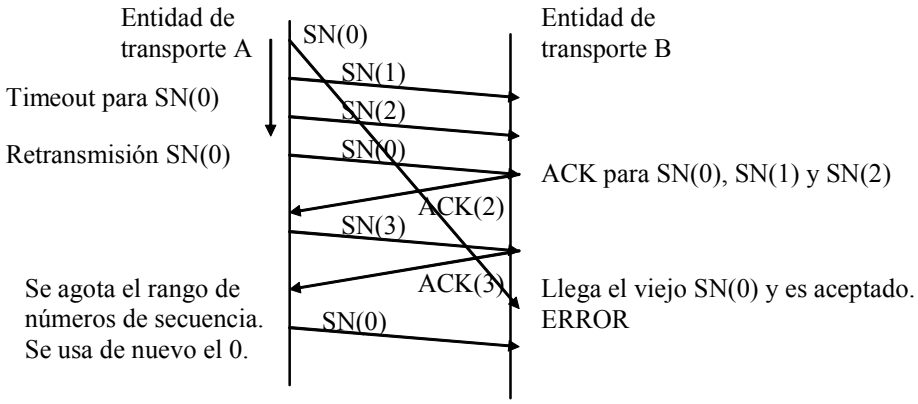
Temporizador de retransmisión	Para retransmitir un segmento no confirmado.
Temporizador de reconexión	Tiempo mínimo entre la liberación de una conexión y el establecimiento de otra con la misma dirección de destino.
Temporizador de ventana	Tiempo máximo entre segmentos ACK/CREDIT.
Temporizador de retransmisión de SYN	Tiempo entre intentos de establecer una conexión
Temporizador de persistencia	Para cancelar una conexión cuando no se confirman segmentos
Temporizador de inactividad	Para cancelar una conexión cuando no se reciben segmentos

Si un ACK se pierde o se agota prematuramente el temporizador de retransmisión, habrá retransmisiones que provocarán la existencia de segmentos **duplicados** en la red. El receptor ha de detectar adecuadamente esos duplicados, a lo que ayuda su número de secuencia, pero no es tan simple. Se pueden dar dos situaciones en el receptor:

a) Se recibe un duplicado antes de la liberación de la conexión.

b) Se recibe un duplicado tras liberar la conexión.

En el primer caso, generalmente el número de secuencia del segmento no será el esperado en ese momento. El receptor debe asumir que su confirmación se perdió y por lo tanto debe devolver un ACK. Por lo tanto, también el emisor del duplicado no debe confundirse si recibe varios ACKs para un mismo segmento. El problema aparece si el número de secuencia de ese segmento es válido en ese momento (ver figura).



Para evitar esto, el rango de los números de secuencia ha de ser lo suficientemente grande como para no agotarse en el tiempo que puede permanecer un segmento en la red. Este tiempo puede ser indefinido si no se limita con algún mecanismo la vida máxima de los datagramas en la red.

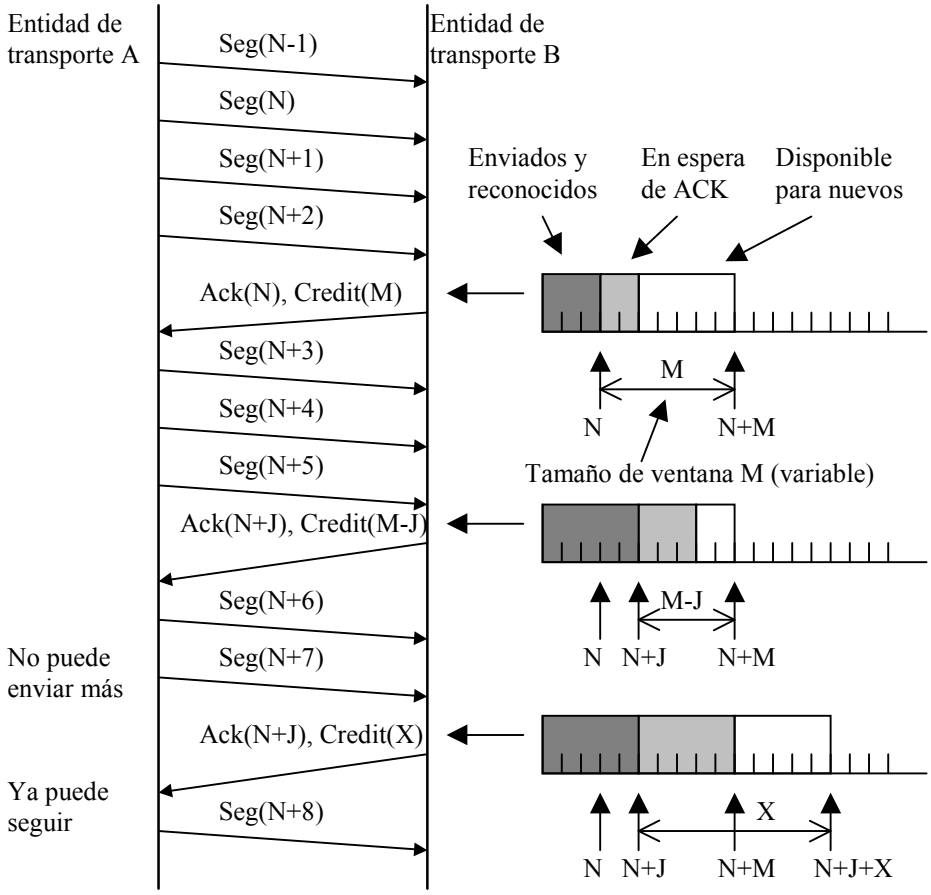
Aún es más sutil el problema que puede aparecer si el duplicado llega una vez que se ha liberado la conexión. Si no se ha establecido una nueva conexión o se ha establecido pero el número de secuencia del segmento no es válido en ella, se descartará y será contestado con un segmento RST (Reset) para abortar la antigua conexión si el otro extremo aún no lo había hecho y no habrá problema. Pero si el número de secuencia resulta válido en la nueva conexión se puede producir de nuevo un error. Hay varias soluciones posibles entre las que vamos a considerar dos:

a) Recordar el número de secuencia en el que finalizó la anterior conexión y continuar a partir de él (el rango del número de secuencia habrá de ser sobradamente amplio).

b) Usar un identificador específico de la conexión nuevo con cada conexión.

Los dos sistemas funcionan correctamente excepto en el caso de que el sistema se venga abajo. Al reiniciarse no será capaz de recordar los números de secuencia o identificadores de conexión que uso la última vez, pues puede no haber tenido tiempo de anotarlos en ningún soporte permanente e incluso de haberlo hecho podrían no ser fiables. Una solución simple es esperar a que expire el *temporizador de reconexión* antes de aceptar una nueva conexión con esa misma máquina, lo que por otro lado puede provocar retardos innecesarios.

Por otro lado, cuando un extremo de la conexión cae, generalmente el otro sigue activo hasta que el *temporizador de persistencia* o el *temporizador de inactividad* expira. El extremo que cae no debería iniciar conexiones hasta haber transcurrido un tiempo prudencial desde su caída para evitar usar números de secuencia que dieran lugar a errores en conexiones no finalizadas.



2.2.2 Control de flujo

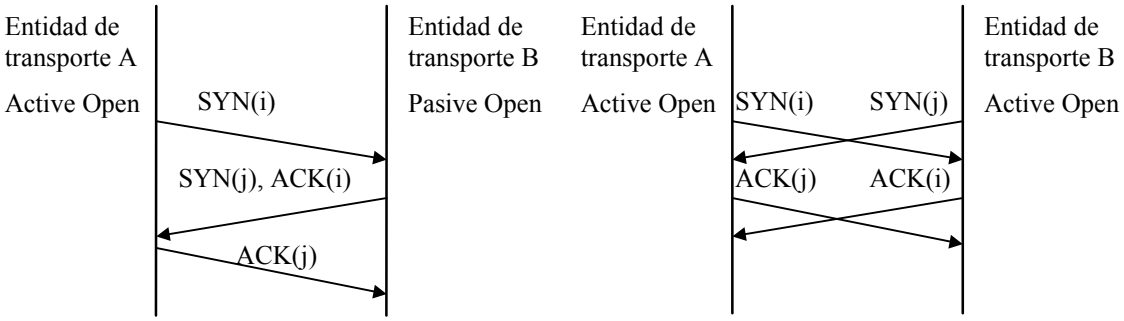
El mecanismo de control por ventana con asignación de créditos es suficientemente robusto para un servicio de red no fiable, con unas pequeñas mejoras. Supóngase que para confirmar segmentos y conceder créditos, se envían segmentos del tipo (ACK N, CREDIT M) donde se confirma la recepción hasta el segmento N y se permite la transmisión de los segmentos de N+1 hasta el N+M (algunos podrían haber sido ya transmitidos por el emisor gracias a segmentos ACK/CREDIT anteriores). Los créditos no son acumulables, es decir, no se suman a los otorgados en segmentos anteriores, sino que representan el número de segmentos que se pueden enviar a partir del último del que se ha hecho acuse de recibo.

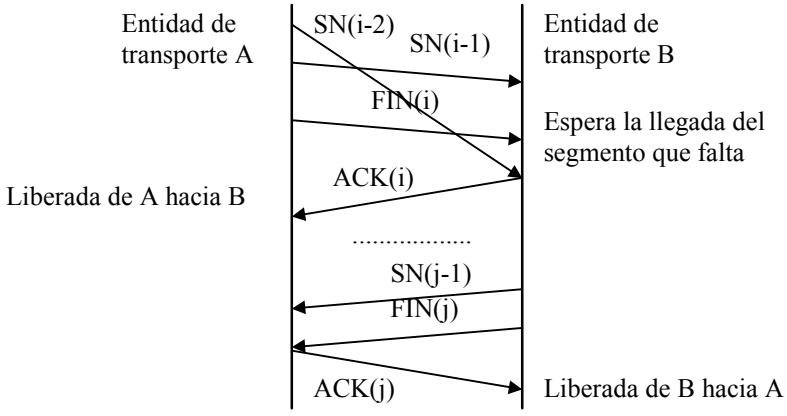
- Si se quieren confirmar J nuevos segmentos sin incrementar el espacio reservado a nuevos segmentos, se envía (ACK N+J, CREDIT M-J). De nuevo se da permiso para enviar segmentos hasta (N+J)+(M-J)=N+M.
- Si a partir de aquí se quieren otorgar más espacio se envía al emisor (ACK N+J, CREDIT X) siendo X>M-J (no se suele admitir que X<M-J, es decir, retrasar el puntero que marca el límite derecho de la ventana).

Si se pierde un segmento ACK/CREDIT no suele representar problema ya que posteriores retansmisiones y confirmaciones lo recuperarán. Si no ocurriera esto, un *temporizador de ventana*, reiniciado cada vez que se envía un segmento ACK/CREDIT, provocaría su retransmisión en caso de inactividad. Una alternativa o mejora del mecanismo sería realizar confirmaciones de los segmentos ACK/CREDIT.

2.2.3 Establecimiento y liberación de la conexión

El establecimiento se debe realizar ahora mediante un *handshake de triple vía*, en el que se intercambian como mínimo tres mensajes para que ambas entidades conozcan el número de secuencia inicial que va a utilizar la homóloga en la conexión y reciban la confirmación de que el suyo ha sido aceptado por la otra.





Como sucedía en el caso de un servicio de red fiable, el método resuelve las situaciones en que uno de los extremos es el que inicia la conexión o bien los dos la inician simultaneamente. Se han de tener en cuenta además, las consideraciones sobre los números de secuencia realizadas anteriormente, ya que en este caso incluso podrían llegar intentos de apertura de conexión antiguos debidos a retransmisiones por extinción del *temporizador de retransmisión de SYN.*

En la liberación de la conexión los segmentos con el mensaje de fin de conexión, FIN, deben llevar el número de secuencia correspondiente, así el receptor sabrá que es el

último segmento de la conexión y si faltan segmentos de datos esperará por ellos antes de confirmar con un ACK el segmento de FIN. Como la comunicación en la mayoría de los casos es full-duplex, el proceso se repite en el otro sentido de la conexión.

Tras el intercambio de los segmentos FIN y sus correspondientes confirmaciones, se suele esperar la llegada de posibles duplicados durante un intervalo igual a dos veces el tiempo máximo de vida esperado de un segmento, para asegurar que no quedan duplicados de segmentos de la conexión en la red y dar más robustez al inicio de nuevas conexiones.

3. BIBLIOGRAFÍA

Bibliografía consultada para la realización de este capítulo:

[STALLINGS 97]
Stallings, W. (1997).
Comunicaciones y redes de computadores, 5ª ed.
Prentice Hall Iberia.

[TANENBAUM 96]
Tanenbaum, A.S. (1996).
Computer Networks. (Third Edition).
Prentice-Hall.

[HALSALL 95]
Halsall, F. (1995).
Data Communications, Computer Networks and Open Systems.
Addison-Wesley.