Rev.R.Acad.Cienc.Exact.Fís.Nat. (Esp)
Vol. 109, Nº. 1–2, pp 75–86, 2016
XIX Programa de Promoción de la Cultura Científica y Tecnológica

Redes de comunicaciones de alta velocidad: ¿cómo funcionan?

José F. Duato Marín

Real Academia de Ciencias Exactas, Físicas y Naturales, Madrid

Universidad Politécnica de Valencia

RESUMEN

Las tecnologías de la información y las comunicaciones han cambiado notablemente la sociedad en las últimas décadas. El desarrollo de dispositivos con grandes potencias de cálculo y tamaño cada vez más pequeño está haciendo emerger un gran número de aplicaciones que eran impensables hasta hace poco. Pero para que muchas de esas aplicaciones puedan funcionar hace falta que existan potentes servidores accesibles a través de Internet, así como una red de comunicaciones de muy alta velocidad para acceder a los mismos. Es más, los propios servidores de Internet se construyen hoy en día con entre centenares y centenares de miles de procesadores, interconectados entre sí por redes de muy alta velocidad. Los propios chips de procesamiento que constituyen la base para esos servidores contienen varios núcleos de procesamiento, cuyo número irá en aumento. Ya existen procesadores con centenares de núcleos en un chip, interconectados por una red dentro del chip.

Todas estas redes, aunque diferentes en su diseño, características y prestaciones, están basadas en los mismos principios de funcionamiento. En este artículo de divulgación se presentan, de forma descriptiva e inteligible para los no expertos, los principios básicos de funcionamiento de todas estas redes. La presentación sigue un enfoque mixto ascendente y descendente. Desde el punto de vista del usuario final se sigue un enfoque descendente, empezando por los servicios ofrecidos y llegando hasta el detalle del funcionamiento. Desde el punto de vista del funcionamiento se sigue un enfoque constructivo ascendente, motivando la necesidad de nueva funcionalidad y presentando las soluciones adoptadas en cada caso.

1. INTRODUCCIÓN

Hoy en día, los ordenadores de todo tipo, incluyendo los portátiles, tabletas y móviles, son una herramienta común en el trabajo y para uso personal. El uso generalizado de los ordenadores ha sido posible gracias a la mejora continua de la relación prestaciones/coste y, lo que es más importante, a la disponibilidad de una amplia gama de aplicaciones y servicios. La forma en que se utilizan los ordenadores ha evolucionado mucho desde los primeros ordenadores personales. La amplia disponibilidad de acceso a Internet motivó el desarrollo de nuevos servicios basados en la red (WWW, comercio electrónico, etc.). Como consecuencia, la carga computacional se ha desplazado de los ordenadores a los *servidores* de Internet. Para poder atender decenas de miles de peticiones concurrentes de usuarios de todo el mundo, muchos de estos servidores requieren procesadores potentes y discos rápidos de gran capacidad. Pero, ¿cuál es el papel de las interconexiones en este contexto? ¿Realmente necesitamos comunicaciones de alta velocidad para proporcionar esos servicios? Muchas personas creen que con sólo aumentar el ancho de banda de su conexión a Internet es suficiente. Sin embargo, la situación es muy diferente, como se describe en las siguientes secciones.

2. EL PAPEL DE LAS INTERCONEXIONES EN LA ERA DE INTERNET

Las prestaciones de los servicios de Internet dependen en gran medida de las prestaciones de los servidores y de Internet. Proporcionar tiempos de respuesta bajos es cada vez más difícil a medida que el uso de los servicios de Internet crece, debido a que las solicitudes de muchos usuarios deben ser atendidas simultáneamente, imponiendo así requisitos de ancho de banda tanto en servidores como en Internet. El tiempo de respuesta y el ancho de banda dependen de muchos factores, incluyendo el número de usuarios simultáneos en un momento dado, la complejidad de la aplicación, la potencia de cálculo de clientes y servidores, el ancho de banda de acceso a los discos en el servidor, el ancho de banda de Internet y el ancho de banda entregado por los proveedores de servicios de Internet.

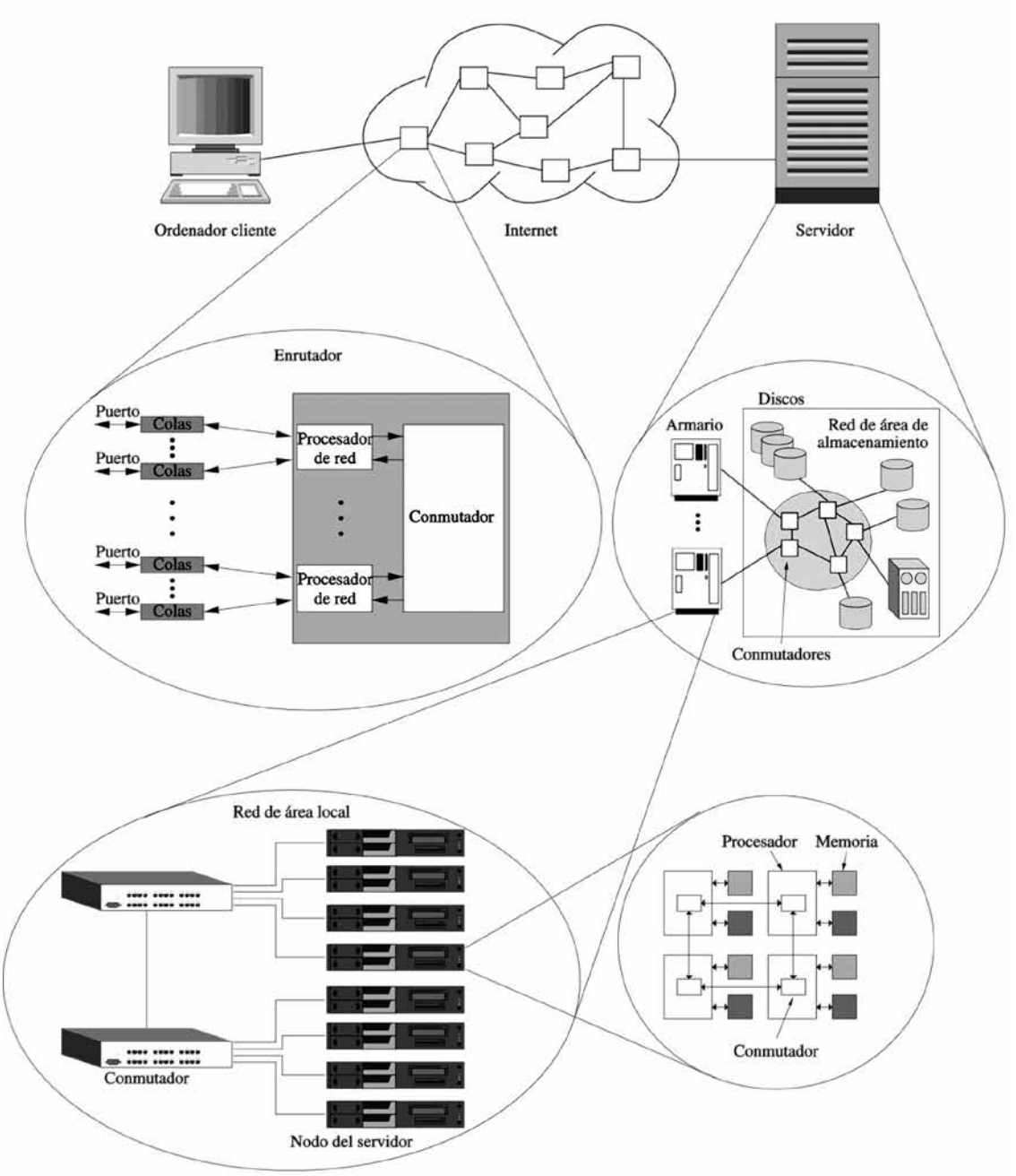

Figura 1. Vista global de la ubicación de diferentes redes de interconexión en los sistemas actuales de altas prestaciones.

Entre todos estos factores, nos interesa analizar los requisitos de comunicación. Si observamos cómo un cliente accede a un servidor de Internet, vemos que los datos se transmiten a través de varias redes. La figura 1 presenta una vista global que incluye la mayoría de las redes de interconexión involucradas. En particular, al acceder a servidores remotos a través de Internet, los mensajes de datos atraviesan uno o más *enrutadores* (*routers* en inglés) que usan el Protocolo de Internet (IP). Esos enrutadores requieren protocolos para comunicarse con otros dispositivos de Internet y para configurar las tablas de enrutamiento utilizadas para establecer una ruta a través de Internet. Dentro de cada enrutador IP, un conmutador se encarga de establecer una ruta desde el puerto de entrada al puerto de salida solicitado para cada mensaje entrante.

Una vez que la solicitud del usuario llega al servidor de Internet apropiado, se necesitan aún más interconexiones de alta velocidad para procesar la solicitud con un tiempo de respuesta razonable. Algunas solicitudes de usuario pueden requerir búsquedas complejas en bases de datos y/o crear páginas web dinámicamente. Además, miles o incluso decenas de miles de clientes pueden ser atendidos simultáneamente. Con el fin de proporcionar la potencia de cálculo requerida a un coste relativamente bajo, la mayoría de los servidores de altas prestaciones se construyen actualmente en torno a un *clúster*. La potencia de cálculo se puede aumentar fácilmente añadiendo más procesadores (es decir, más nodos de procesamiento). Sin embargo, para resolver el cuello de botella de acceso a disco y aumentar la fiabilidad, los grandes sistemas utilizan subsistemas de disco fiables externos (por ejemplo, matrices redundantes de discos (RAID)) e interconexiones de alta velocidad que conectan procesadores y subsistemas de disco.

Pero las interconexiones de alta velocidad no sólo son necesarias para interconectar procesadores y discos. Procesar una solicitud compleja puede requerir la ejecución de varios procesos. Estos procesos se pueden ejecutar en paralelo con el fin de reducir el tiempo de respuesta. Además, algunos procesadores pueden requerir servicios de otros procesadores (por ejemplo, acceder a una base de datos). En estos casos, los procesadores tienen que comunicarse entre ellos, requiriendo así otra interconexión de alta velocidad. En la mayoría de los casos, la tecnología de red de área local (LAN) más común (Gigabit o 10 Gigabit Ethernet) será suficiente. En otros casos, pueden ser necesarias interconexiones con mayores prestaciones, tales como InfiniBand®.

A su vez, los nodos de procesamiento se pueden diseñar como un multiprocesador para aumentar las prestaciones. Los nodos con doble procesador son muy comunes en los clústeres actuales, pero también los hay con un mayor número de procesadores, todos ellos interconectados entre sí. Finalmente, las elevadas escalas de integración de los circuitos actuales han hecho posible los procesadores multinúcleo, lo que requiere una red de interconexión dentro del chip para conseguir una comunicación rápida entre los núcleos.

Este ejemplo de acceso a un servidor de altas prestaciones a través de Internet ha revelado que las prestaciones de los sistemas informáticos basados en red dependen en gran medida del desarrollo de interconexiones de alta velocidad capaces de satisfacer las crecientes demandas tanto en los servidores como en los enrutadores IP. Las interconexiones de alta velocidad para diferentes áreas de aplicación comparten muchos problemas de diseño. Por lo tanto, estas interconexiones pueden analizarse conjuntamente. Además, las soluciones propuestas en un área pueden ser válidas para otra área.

3. ARQUITECTURA BÁSICA DE RED

La arquitectura de la red puede ser muy diferente dependiendo del número de dispositivos que necesitan comunicarse y de las limitaciones de diseño. La comunicación entre dos dispositivos se puede lograr fácilmente usando un *enlace punto a punto*. Cuando más de dos dispositivos tienen que intercambiar información, existen básicamente dos opciones: i) usar un *medio de transmisión compartido* que enlaza todos los dispositivos, o ii) usar un *conmutador* para establecer dinámicamente conexiones entre pares de dispositivos. El primer enfoque es más sencillo y se ha utilizado tradicionalmente para interconectar un pequeño número de dispositivos, pero tiene importantes limitaciones. En cambio, los conmutadores son más complejos pero son mucho más efectivos para interconectar varios dispositivos, siendo preferibles al diseñar interconexiones de alta velocidad.

Cuando el número de dispositivos a conectar excede del número de *puertos* disponible en un solo conmutador, es posible diseñar *redes* con múltiples conmutadores. Estas redes suelen basarse en enlaces punto a punto para conectar conmutadores y dispositivos entre sí. En estas redes, los mensajes enviados por dispositivos conectados a la red pueden requerir cruzar varios conmutadores antes de llegar a su destino.

Finalmente, los servicios de comunicación proporcionados por los conmutadores pueden no ser los adecuados para el resto del sistema. En este caso, los dispositivos se pueden conectar a la estructura del conmutador a través de una *interfaz de red*, que proporciona los servicios necesarios. La complejidad de las interfaces de red varía significativamente de un sistema a otro. Todas las opciones de interconexión anteriores se analizan brevemente en las siguientes secciones, que también introducen los principales servicios proporcionados por las interfaces de red.

3.1 Comunicación directa entre dos dispositivos

La comunicación directa entre dos dispositivos se realiza habitualmente utilizando un *enlace punto a punto*. Los dispositivos de comunicación suelen tener relojes independientes. Se requiere un *protocolo de comunicación* entre los dispositivos para que la comunicación se realice correctamente. En general, este protocolo permite al transmisor notificar la transmisión de datos, permitiendo también que el receptor notifique la recepción correcta y la disponibilidad de espacio para almacenar nuevos datos.

Para conseguir altas prestaciones, el ancho de banda del enlace debe ser lo más alto posible. El *ancho de banda* de un enlace es el producto de la frecuencia de reloj por el ancho de enlace. Los avances en la tecnología VLSI permiten velocidades de reloj más rápidas. Sin embargo, los datos no se propagan instantáneamente a través de un enlace. Por lo tanto, para utilizar la frecuencia de reloj de enlace máxima permitida por una tecnología dada, puede ser necesario inyectar nuevos datos en un enlace antes de que los datos previamente inyectados alcancen el otro extremo del enlace. Esto se conoce generalmente como *segmentación de canales*. En este caso, una secuencia de señales eléctricas se propaga a través de los hilos (o la luz se propaga a través de la fibra) como ondas, sin mezclarse entre sí, si bien hay una cierta atenuación de la señal que limita la aplicabilidad de esta técnica. Dependiendo de la frecuencia de reloj y la longitud del enlace, desde unos pocos bits hasta varios mensajes pueden estar viajando por el mismo enlace en un momento dado.

Suponiendo que el transmisor almacena los datos a transmitir en una única cola de memoria y que el receptor también almacena los datos recibidos en una cola, es necesario asegurarse de que no se excede la capacidad de la cola de recepción. Esto puede lograrse mediante un protocolo de *control de flujo* a nivel de enlace, que se encarga de notificar al transmisor la disponibilidad de espacio en el lado receptor. Una *unidad de control de flujo* es la cantidad de datos sobre los que se aplica el control de flujo. El espacio de almacenamiento se mide en unidades de control de flujo.

Los datos se almacenan en el lado de recepción antes de ser procesados. Existen varias cuestiones relacionadas con dicho almacenamiento, denominadas genéricamente *gestión de colas*. Habitualmente, varios flujos de información se multiplexan sobre un único enlace. Suponiendo que cada flujo se almacena en una cola física o lógica diferente, tanto en el lado del transmisor como del receptor, es necesario extender el protocolo de control de flujo para identificar adecuadamente cada flujo, almacenar sus datos en la cola correspondiente e implantar adecuadamente el control de flujo en cada cola. En este caso, la gestión de colas se vuelve más compleja. La selección de la cola en el lado del receptor puede basarse en información explícita proporcionada por el transmisor, calculada dinámicamente en el lado del receptor con el objetivo de optimizar el uso de recursos o reducir el tiempo de espera en la cola, o basada en la información proporcionada por el mensaje que se recibe (el destino de ese mensaje).

Además, cuando se multiplexan varios flujos de información sobre un enlace, se requiere un *planificador de transmisión* en el lado de transmisión para seleccionar el flujo para el que se transmitirá a continuación una unidad de control de flujo. Este planificador garantiza un uso justo de los recursos. Además, se puede requerir un *planificador de recepción* en el lado del receptor para programar la devolución de la información de control de flujo.

El planificador de transmisión gestiona las *prioridades*, si las hay, cuando selecciona una unidad de control de flujo para la siguiente transmisión. Además, algunos flujos de información requieren un ancho de banda garantizado y/o un límite garantizado en el retraso experimentado por cada unidad de control de flujo. Estas garantías se denominan generalmente *calidad de servicio*, o QoS. En este caso, el planificador de transmisión se vuelve mucho más complejo. Obviamente, sería imposible garantizar QoS si el ancho de banda total requerido por los flujos excediera el ancho de banda del enlace. Generalmente se requiere un protocolo de *control de admisión* para evitarlo.

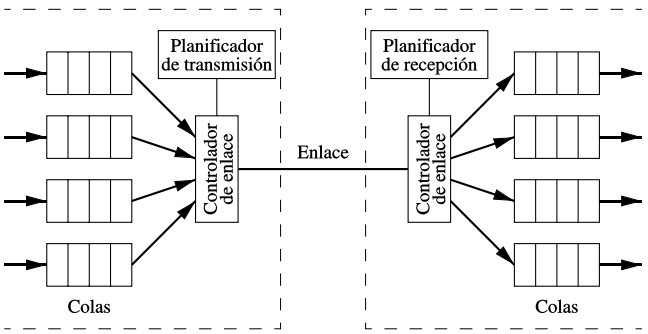

Figura 2.: Componentes requeridos para la transmisión de datos por un enlace punto a punto.

La figura 2 muestra un diagrama de bloques que incluye la mayoría de los elementos descritos anteriormente. Los *controladores de enlace* realizan el control de flujo. En la figura 3 se muestra una solución habitual que usa memoria RAM para las colas. Los gestores de colas pasan a ser gestores de memoria y contienen listas de punteros para realizar un seguimiento de las ubicaciones de memoria asociadas con cada cola, manteniendo también una lista de ubicaciones de memoria libre donde almacenar los mensajes entrantes.

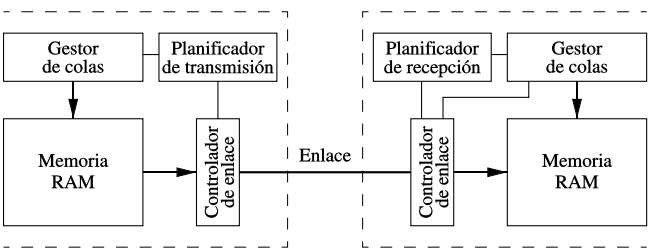

Figura 3. Solución habitual que usa memoria RAM para las colas.

3.2 Comunicación a través de un solo conmutador

La forma más sencilla de establecer comunicación entre varios dispositivos consiste en usar un medio compartido, frecuentemente denominado *bus*. Además de su simplicidad, un bus cuenta con dos características importantes. En primer lugar, garantiza la entrega en orden de todos los mensajes transmitidos a través de él. En segundo lugar, un bus permite a cada dispositivo transmitir información a todos los demás dispositivos conectados al bus, sin coste ni retrasos adicionales. Sin embargo, a pesar de sus ventajas, un bus no es adecuado para la comunicación de altas prestaciones. El motivo principal es que el ancho de banda debe ser compartido entre todos los dispositivos que se comunican a través de ese bus.

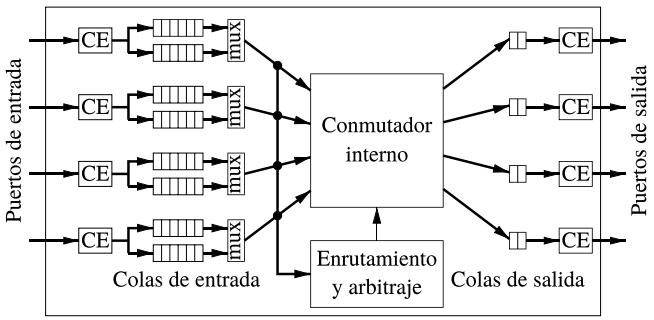

Figura 4. Arquitectura básica de un conmutador.

Cuando un bus no proporciona suficiente ancho de banda para interconectar un conjunto de dispositivos, se pueden lograr mayores prestaciones conectando esos dispositivos a través de un conmutador. Un *conmutador* es un circuito digital diseñado para interconectar varios dispositivos. Un conmutador normalmente incluye un conjunto de *puertos de entrada y salida* conectados a enlaces externos, un conjunto de memorias o *colas* para almacenar temporalmente unidades de control de flujo (o flits, para abreviar), un *conmutador interno* y algunos circuitos de control que proporcionan cierta funcionalidad que se describirá a continuación. La organización interna de un conmutador se conoce generalmente como *arquitectura del conmutador*. La figura 4 muestra un diagrama de bloques para la arquitectura básica de un conmutador. En

este diagrama, varias colas de entrada están asociadas con cada puerto de entrada. Ciertamente, esta no es ni la única opción ni la más eficiente. Sin embargo, esta arquitectura de conmutador es bastante popular y será útil para describir los problemas de diseño más relevantes.

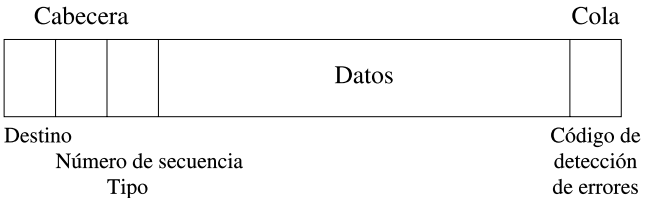

Figura 5. Ejemplo de formato de un paquete.

El comportamiento básico del conmutador es el siguiente. Cuando se recibe una unidad de control de flujo en algún puerto de entrada, el gestor de memoria determina la cola en la que debe almacenarse. Mientras tanto, el controlador de enlace (CE) realiza las acciones asociadas con el control de flujo. Para transmitir cada unidad de control de flujo por el siguiente fragmento de su ruta, es necesario establecer una conexión a través del conmutador interno al puerto de salida apropiado. Sin embargo, esto no puede hacerse sin tener información sobre el puerto de salida solicitado. Para ello, una o más unidades de control de flujo se agrupan generalmente en unidades más grandes, usualmente denominadas *paquetes*. La figura 5 muestra un ejemplo de formato típico de paquete. Los paquetes contienen una cabecera con información de destino. Esa cabecera se envía desde la cola de entrada correspondiente a la *unidad de enrutamiento y arbitraje*, que calcula el puerto de salida apropiado para ese paquete y solicita dicho puerto. En caso de solicitudes simultáneas al mismo puerto de salida, sólo se concede una solicitud y la unidad de enrutamiento y arbitraje instruye al conmutador interno para configurar la ruta correspondiente. A continuación, las unidades de control de flujo para ese paquete pueden ser transmitidas a las colas de salida, y desde allí a través del enlace de salida correspondiente. En resumen, la unidad de enrutamiento y arbitraje establece rutas dinámicamente a través del conmutador interno de acuerdo con las peticiones de los paquetes entrantes, posiblemente seleccionando entre peticiones que compiten por los recursos.

La *técnica de conmutación* determina cuándo se configura el conmutador interno y cuándo se pueden retransmitir los componentes de los paquetes (normalmente, unidades de control de flujo) a lo largo de la ruta correspondiente. Por ejemplo, cuando se usa *conmutación de paquetes*, un conmutador debe esperar a recibir un paquete entero antes de retransmitirlo. Pero existen técnicas de conmutación más agresivas, como *wormhole* (*agujero de gusano*), que empiezan a retransmitir un paquete nada más llega su cabecera. Las técnicas de conmutación están acopladas con el control de flujo para la transferencia de unidades de control de flujo a través de los enlaces de entrada, el conmutador interno y los enlaces de salida. Como además el control de flujo está acoplado con la gestión de las colas, conjuntamente determinan cómo se gestionan los paquetes cuando se bloquean debido a congestión en la red.

Además de decidir cuándo se establecerá una ruta a través del conmutador interno para un paquete dado, también es necesario decidir qué ruta se establecerá. Esta operación se conoce como *enrutamiento*, y es realizada por la unidad de enrutamiento y arbitraje. La forma más sencilla de realizarla en un único conmutador consiste en codificar directamente el puerto de salida requerido en la cabecera del paquete. Este problema se vuelve importante en redes con múltiples conmutadores y, por lo tanto, se analizará en la siguiente sección.

Como ya se ha indicado, varios paquetes pueden solicitar el mismo puerto de salida al mismo tiempo. Un *árbitro* selecciona entre las peticiones en disputa. El puerto solicitado se concede al ganador y los perdedores tendrán que esperar, quedando así bloqueados en las colas correspondientes. El control de flujo evita el desbordamiento de las colas. Este tipo de conflictos puede tener consecuencias muy negativas. No sólo retrasa a los paquetes que pierden el arbitraje, sino que éstos, a su vez, pueden bloquear a otros paquetes, produciendo así un efecto acumulativo. Esto es especialmente grave cuando algunos paquetes bloqueados en una cola bloquean a paquetes posteriores en la misma cola para los cuales su puerto de salida solicitado está libre. Esta situación se conoce como *bloqueo de cabeza de línea*. En este caso, el ancho de banda del enlace se desperdicia y puede conducir a una degradación significativa de las prestaciones.

3.3 Redes con múltiples conmutadores

Cuando el número de dispositivos a conectar excede un cierto valor que depende de la tecnología VLSI disponible en un momento dado, resulta imposible interconectar todos ellos usando un solo conmutador. En este caso, es posible diseñar redes con múltiples conmutadores. Estas redes suelen basarse en enlaces punto a punto para conectar conmutadores y dispositivos entre sí. En redes con múltiples conmutadores hay que especificar, al menos, el patrón de interconexión entre conmutadores y cómo seleccionar una ruta desde el dispositivo de origen al dispositivo de destino. Sin embargo, esto es más complejo de lo que parece. A medida que crece el tamaño de las redes, algunos problemas se vuelven más críticos y deben ser tratados (por ejemplo, prestaciones y fiabilidad). Además, algunos servicios adicionales, como las garantías de QoS, requieren mayor complejidad para conseguir que sean eficientes. En esta sección se presenta una visión general de los principales problemas de diseño relacionados con las redes con múltiples conmutadores.

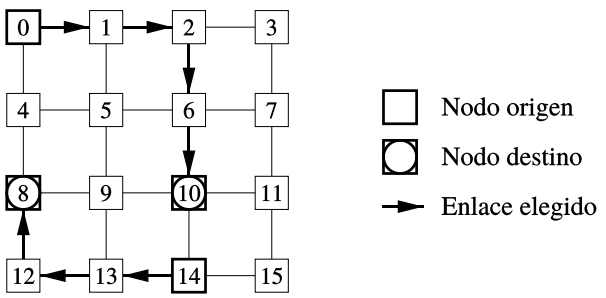

Figura 6. Ejemplo de red directa y de rutas entre nodos: Malla bidimensional.

El patrón de interconexión entre conmutadores define la *topología* de la red. Los componentes se pueden organizar de varias maneras. Una forma es que se conecte un solo dispositivo (o un grupo de ellos de tamaño fijo) a cada conmutador, que a su vez está interconectado a otros conmutadores siguiendo algún patrón de conexión (normalmente regular). Esta disposición se conoce como una *red directa* porque la correspondencia uno a uno entre grupo de dispositivos y conmutadores permite la integración del conmutador en la misma tarjeta o chip que el grupo de dispositivos y, por lo tanto, esas placas o chips están interconectados directamente entre ellos. La figura 6 muestra un ejemplo de red directa de pequeño tamaño con topología de malla bidimensional, incluyendo ejemplos de rutas entre nodos.

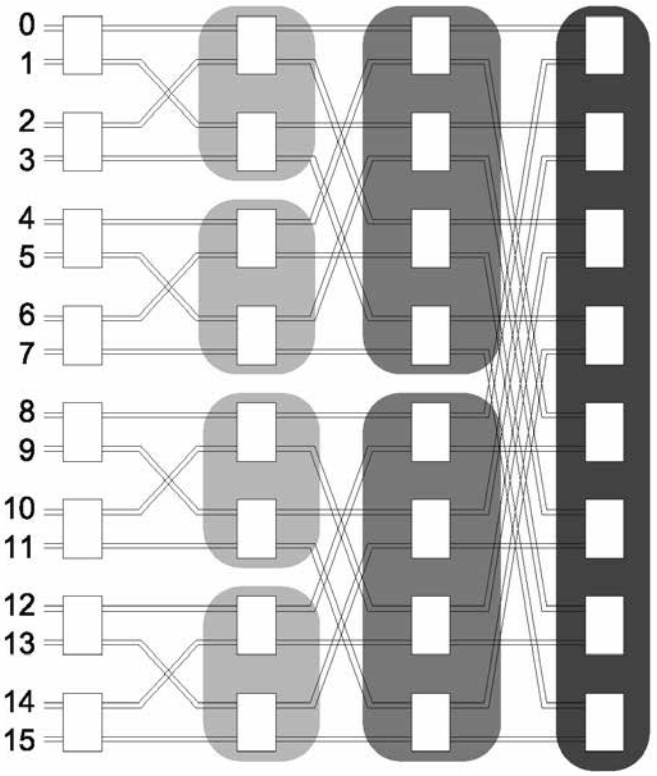

Figura 7. Ejemplo de red de interconexión multietapa. Cada columna de conmutadores corresponde a una etapa. Por claridad, los dispositivos conectados a la red se han representado mediante su dirección numérica.

Otra forma consiste en disponer conmutadores en varias etapas, cada una conectada a la siguiente etapa siguiendo algún patrón de interconexión regular, y uniendo dispositivos solamente a la primera etapa de conmutadores. Esta disposición se conoce generalmente como *red de interconexión multietapa*. La figura 7 muestra una red multietapa con conmutadores pequeños (2×2) y topología Benes plegada. Es un caso particular de la popular topología *árbol gordo* (*fat tree* en inglés). En general, las redes en las que sólo un subconjunto de conmutadores tienen dispositivos conectados a ellos se denominan *redes indirectas* o *redes conmutadas* porque los conmutadores se disponen generalmente en armarios externos, permitiendo así la interconexión de dispositivos indirectamente a tra-

vés de un conjunto de conmutadores. En ambos casos, pero especialmente en el caso de redes indirectas, el patrón de interconexión entre conmutadores puede no seguir ningún patrón regular. Tales redes se denominan generalmente *redes irregulares*. En general, podemos ver las redes que usan enlaces punto a punto como un conjunto de conmutadores interconectados, cada uno conectado a cero, uno o más dispositivos. Las redes directas corresponden al caso en que cada conmutador está conectado a un solo (grupo de) dispositivo(s). Las redes de interconexión multietapa corresponden al caso en el que los conmutadores están dispuestos en varias etapas y los conmutadores en etapas intermedias no están conectados a ningún dispositivo. Las redes irregulares corresponden al caso general.

La interconexión de varios conmutadores plantea nuevos problemas. En primer lugar, debe seleccionarse una topología adecuada para la red. A veces, la topología se decide en el momento del diseño. En otros casos, el objetivo es proporcionar suficiente flexibilidad al usuario final para que configure la topología deseada. En algunas áreas de aplicación (por ejemplo, supercomputadores), también es importante determinar cómo organizar físicamente todos los componentes de la red en chips, placas y armarios, de modo que los enlaces sean lo más cortos posible y el número de enlaces que atraviesan cada chip, placa o armario sea lo menor posible. La selección de una topología de red apropiada es crucial para obtener un buen diseño de componentes.

En las redes con más de un conmutador, los paquetes normalmente tienen que cruzar varios conmutadores antes de llegar a su destino. Por lo tanto, es necesario establecer una ruta para cada paquete para que pueda llegar a su destino lo más rápido posible. La figura 6 muestra un par de ejemplos de rutas en una red directa. El algoritmo para calcular dicha ruta se denomina *algoritmo de enrutamiento*. Este algoritmo puede ser calculado de forma distribuida (paso a paso en las unidades de encaminamiento y arbitraje en los conmutadores a lo largo de la ruta recorrida por un paquete), de forma centralizada para cada paquete (en el dispositivo origen o en su interfaz de red asociada) o de forma centralizada para toda la red (en un gestor de red). Cuando el algoritmo no se calcula de forma distribuida, se suelen usar unas *tablas de enrutamiento* en cada conmutador e interfaz de red para almacenar las rutas precalculadas. Cada conmutador debe contener los circuitos para calcular la ruta o la memoria para almacenar la información para enrutar cada paquete entrante. Los paquetes deben almacenar la información de destino en su cabecera. En algunas redes, la cabecera de cada paquete codifica la ruta completa para ese paquete.

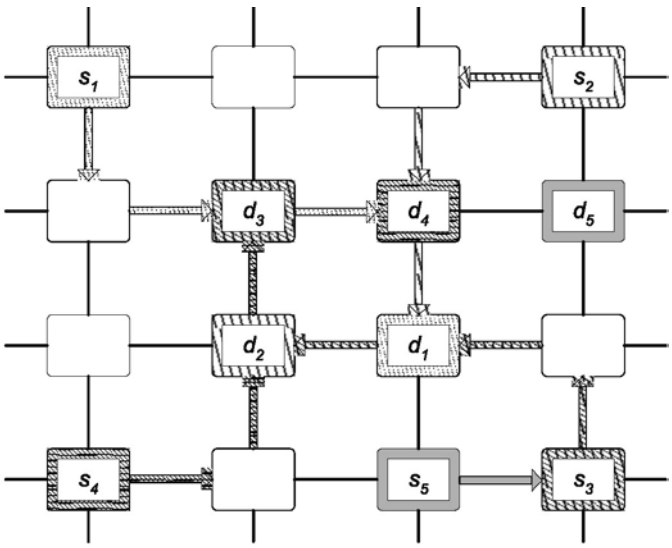

Figura 8. Ejemplo de interbloqueo en una malla bidimensional. Las flechas indican los fragmentos de ruta cuyas colas han sido reservadas por cada paquete con origen en s_i y destino en d_i.

El principal problema asociado con el enrutamiento es el interbloqueo. Se produce un *interbloqueo* cuando algunos paquetes no pueden avanzar hacia su destino porque las colas solicitadas por ellos están permanentemente llenas. Todos los paquetes implicados en un interbloqueo se bloquean para siempre, cada uno solicitando colas ocupadas por otros paquetes y ocupando colas solicitadas por otros paquetes. La figura 8 muestra un ejemplo de interbloqueo en una malla bidimensional. El interbloqueo se puede evitar diseñando cuidadosamente el algoritmo de enrutamiento. Existen técnicas bien conocidas para hacerlo, tales como imponer algunas restricciones de enrutamiento, por lo que ciertos caminos no se pueden utilizar para llegar a ciertos destinos. Estas restricciones de enrutamiento pueden reducir las prestaciones.

En general, existen múltiples rutas de cada dispositivo de origen a cada dispositivo de destino en una red

con múltiples conmutadores. Sin embargo, el algoritmo de enrutamiento puede diseñarse para proporcionar una sola ruta de cada origen a cada destino (denominado *enrutamiento determinista*). Esta estrategia suele conseguir un enrutamiento más rápido, pero la red generalmente se satura a una carga media más baja. Alternativamente, las diferentes rutas pueden utilizarse para evitar regiones congestionadas de la red o para proporcionar servicio en presencia de fallos. Las estrategias de *equilibrado de carga* intentan distribuir el tráfico entre rutas alternativas para que la utilización de los enlaces esté equilibrada (es decir, sea similar para diferentes enlaces), evitando o retrasando la aparición de zonas saturadas en la red. Una forma particular de conseguir equilibrar la carga consiste en recopilar información sobre el tráfico en la red y seleccionar la ruta de cada paquete de acuerdo con la disponibilidad de recursos, minimizando así la aparición de zonas congestionadas. Esta estrategia se conoce como *enrutamiento adaptativo* y la mayoría de las veces sólo utiliza información local sobre la utilización de recursos, minimizando así la sobrecarga. El principal problema introducido por el enrutamiento adaptativo es la *entrega de paquetes fuera de orden*, lo cual no es aceptable en algunas áreas de aplicación.

A veces, la misma información debe ser entregada a varios destinos. Esto se conoce como *enrutamiento multidestino*. En otros casos, varios dispositivos deben enviar cierta información al mismo dispositivo, que tiene que combinar todos los mensajes recibidos. Esta operación de comunicación se conoce como *reducción*. Estas primitivas de comunicación se denominan genéricamente *operaciones de comunicación colectiva*. Existen muchas otras operaciones de comunicación colectiva. Algunas de ellas son casos particulares. Por ejemplo, un caso particular del enrutamiento multidestino es la *difusión*, en la que se debe entregar la misma información a todos los dispositivos conectados a la red. Otras operaciones son combinaciones de operaciones más simples. Por ejemplo, una operación de comunicación de uso frecuente que implica varios dispositivos es la *sincronización por barrera*, y se utiliza para sincronizar dichos dispositivos. Consiste en una reducción (en la cual cada dispositivo participante notifica que alcanzó la barrera a un único dispositivo de coordinación) seguido por una difusión a todos los dispositivos participantes (notificando que se ha alcanzado la barrera).

El problema de enrutamiento multidestino se puede resolver enviando copias independientes de los mismos paquetes a cada destino. Aunque esta solución funciona, introduce una sobrecarga significativa en la red. En la mayoría de los casos, es posible organizar la comunicación de tal manera que algunos destinos de paquetes colaboren en la distribución de copias enviando esos paquetes a otros destinos, que a su vez también pueden reenviar copias a otros destinos hasta que se alcancen todos los destinos. Cuando esto no es posible o se requieren mayores prestaciones, es posible diseñar circuitos para el enrutamiento multidestino en los conmutadores. Además de proporcionar rutas simultáneas desde cada puerto de entrada a varios puertos de salida en el conmutador interno, esta estrategia requiere una codificación apropiada de todos los destinos en la cabecera de cada paquete. El enrutamiento multidestino también introduce nuevos escenarios potenciales de interbloqueo.

Algunos problemas no son específicos de las redes con múltiples conmutadores pero se vuelven más críticos y difíciles de resolver a medida que aumenta el tamaño de la red. Uno de estos problemas es la congestión. Ésta puede propagarse a través de la red, formando un *árbol de congestión* que crece desde el área congestionada inicial hasta los dispositivos origen que están transmitiendo paquetes hacia (o a través de) esa área. Esta situación sólo ocurre cuando una parte de la red se satura. Por lo tanto, sólo representará un problema serio si la carga media inyectada en la red representa habitualmente un porcentaje grande de su capacidad. En tal caso, los árboles de congestión pueden degradar significativamente las prestaciones de la red y se requiere alguna solución. La solución tradicional a este problema consiste en detectar la congestión, notificar esa situación a los dispositivos origen y limitar la inyección en esos dispositivos hasta que la congestión desaparezca. Esta estrategia se conoce como *control de congestión* y puede implantarse de maneras muy diferentes dependiendo de dónde se detecte la congestión, cómo se transmita esta información a los dispositivos origen y cómo se limite la inyección en dichos dispositivos.

Otro problema relacionado que también se agrava cuando la red se hace más grande es proporcionar soporte para QoS. Efectivamente, cuando aumenta el tamaño de la red, las rutas a través de cada conmutador

deben ser compartidas por un mayor número de flujos de paquetes. Aunque existen muchas más rutas alternativas en la red, su uso puede ser restringido por el algoritmo de enrutamiento para evitar interbloqueos. Por lo tanto, en redes con múltiples conmutadores, se hace aún más necesario utilizar algún protocolo de control de admisión para asegurarse de que hay suficiente ancho de banda disponible para cada conexión aceptada a lo largo de toda la ruta que utilizará.

La fiabilidad también puede llegar a ser más crítica cuando crece el tamaño de la red. Efectivamente, la probabilidad de que falle un componente aumenta con el número de componentes. Por lo tanto, las redes deben diseñarse de tal manera que un fallo en un componente no inutilice toda la red. Los enfoques tradicionales para abordar el problema de la fiabilidad consisten en incorporar componentes redundantes que se activan cuando se detecta un fallo o bien en diseñar un algoritmo de *enrutamiento tolerante a fallos*, que es capaz de enrutar paquetes hacia su destino en presencia de ciertos fallos en la red (siempre que el dispositivo destino no haya fallado). El enrutamiento tolerante a fallos aprovecha los caminos alternativos proporcionados por la topología. Esta estrategia suele ser más barata que replicar componentes de red pero, obviamente, las prestaciones se degradan cuando fallan algunos conmutadores y/o enlaces. El enrutamiento tolerante a fallos puede introducir nuevos escenarios de interbloqueo. En todos los casos se requieren técnicas de *detección de fallos* eficientes y fiables.

Más recientemente se ha propuesto otro enfoque para afrontar los fallos en la red. Algunas áreas de aplicación requieren proporcionar suficiente flexibilidad para que el usuario final pueda definir la topología de red requerida e incluso modificarla dinámicamente (adición y sustitución en caliente de componentes y dispositivos de red). Esta flexibilidad requiere algún *protocolo de configuración de red* para descubrir la topología de la red, calcular las tablas de enrutamiento y transferirlas a los conmutadores y/o interfaces de red. Sin embargo, no hay diferencia significativa entre el fallo de un componente de red y la eliminación en caliente del mismo componente. Por lo tanto, los mismos protocolos de configuración de red se pueden utilizar para realizar una *reconfiguración de la red* en caso de fallo.

Esta flexibilidad se suele proporcionar mediante conmutadores e interfaces de red configurables. Sin embargo, el uso de componentes de red configurables implica que se requiere alguna fase de inicialización antes de que la red esté completamente operativa. Entre otras cosas, esta fase de inicialización debe incluir el protocolo de configuración de red. Algunas operaciones de mantenimiento también pueden ser necesarias. Todas estas operaciones se denominan genéricamente *gestión de red* y pueden ser realizadas por un *gestor de red* dedicado o implantadas de forma distribuida en las interfaces de red y/o conmutadores.

3.4 Servicios proporcionados por la interfaz de red

La *interfaz de red* es el componente que establece la conexión entre los dispositivos conectados a la red y la propia red. Las interfaces de red varían mucho de una aplicación a otra. En algunos casos, consiste simplemente en un par de colas, que almacenan mensajes a transmitir y recibir, respectivamente. En otros casos, es un sofisticado componente que contiene varios procesadores de propósito general y especializados, varias decenas de megabytes de memoria RAM, varios dispositivos de acceso directo a memoria (DMA), interfaces con el sistema y enlaces de red, entre otros. Por lo tanto, la funcionalidad y los servicios proporcionados por la interfaz de red varían drásticamente de un sistema a otro. La principal función es el soporte para la transmisión y recepción de datos. Algunas interfaces de red proporcionan soporte para muchas colas. Este es el caso en la mayoría de las tarjetas de interfaz de red que se utilizan para conectar un nodo de computación a un clúster. Las interfaces de red también pueden incluir uno o más dispositivos DMA, pudiendo así acceder a datos en una memoria externa (por ejemplo, la memoria principal de los nodos de computación). En el caso de las redes directas, la interfaz de red suele estar integrada en cada nodo de la red. Sin embargo, las interfaces de red suelen estar conectadas a la red a través de un enlace en una red indirecta. En este caso, la interfaz de red también requiere control de flujo a nivel de enlace. Las interfaces de red también pueden proporcionar soporte para fragmentar mensajes en paquetes de tamaño fijo.

4. RELACIÓN ENTRE VARIABLES DE DISEÑO

Los mecanismos y técnicas presentados en la sección anterior pueden parecer arbitrarios en algunos casos. Sin embargo, son necesarios cuando la búsqueda de mayores prestaciones lleva a los diseñadores a afrontar problemas creados por limitaciones tecnológicas o por técnicas previamente introducidas. Por ejemplo, la tecnología limita el número de puertos que se pueden integrar de manera eficiente en un único conmutador. Para superar esta limitación, se pueden usar redes con múltiples conmutadores. Sin embargo, esto introduce la necesidad de seleccionar una ruta apropiada para cada paquete, dando así lugar al problema de enrutamiento. A su vez, algunas soluciones al problema de enrutamiento introducen los problemas de interbloqueo y de entrega fuera de orden. Por lo tanto, como muchos mecanismos y técnicas se introducen para resolver otros problemas, es obvio que diferentes áreas de aplicación con diferentes requisitos implantarán diferentes conjuntos de soluciones.

Además, los problemas y las soluciones propuestas no son independientes entre sí. No sólo las soluciones a algunos problemas pueden crear otros problemas. También las soluciones a un problema dado pueden ayudar a resolver otro problema. Por ejemplo, un protocolo de control de admisión para conexiones que requieran garantías de QoS puede solucionar el problema de congestión. Como ejemplo adicional, los protocolos de configuración de red, que son necesarios para soportar topologías arbitrarias definidas por el usuario, se pueden usar para mejorar la tolerancia de fallos simplemente añadiendo un mecanismo de detección de fallos adecuado en cada componente de red.

En general, las variables de diseño no son independientes. Diferentes variables de diseño pueden producir acciones complementarias hacia un objetivo común, y deben ser sintonizadas conjuntamente. Por ejemplo, al intentar obtener las máximas prestaciones, la tecnología VLSI y el consumo de energía determinan el ancho de banda de enlace máximo alcanzable. El diseño de componentes y la topología de red determinan cuántos enlaces pueden utilizarse para interconectar los conmutadores entre sí, así como la longitud de dichos enlaces. El número de enlaces puede estar limitado por el encapsulado del chip, los conectores de la placa o el cableado del armario. Las variables de diseño anteriores definen el ancho de banda máximo disponible entre los dispositivos conectados a la red. Sin embargo, para que un determinado conjunto de mensajes se transmitan, no es posible utilizar todo el ancho de banda proporcionado por la red porque los diferentes mensajes que llegan a un conmutador dado pueden solicitar el mismo puerto de salida, dejando así libres otros puertos de salida. En otras palabras, la utilización del enlace no está equilibrada y, como consecuencia, se pierde un cierto ancho de banda. Las técnicas de equilibrado de carga pueden utilizarse para equilibrar la utilización de los enlaces. El enrutamiento adaptativo es particularmente útil cuando las condiciones de tráfico no se conocen a priori, ya que selecciona la ruta de cada mensaje teniendo en cuenta la carga de tráfico actual. Por lo tanto, el equilibrado de carga y el enrutamiento adaptativo se pueden utilizar para maximizar las prestaciones alcanzables por la red, logrando así una mejor utilización del ancho de banda disponible. Por desgracia, la mayoría de las redes no se comportan de forma estable cuando se cargan a sus máximas prestaciones debido a que se forman árboles de congestión. En este caso, las estrategias de control de congestión ayudan a prevenir la expansión de los árboles de congestión, evitando así la degradación de las prestaciones. En resumen, la tecnología VLSI, los encapsulados y conectores, el diseño de componentes y la topología determinan el ancho de banda disponible. El equilibrio de carga y el enrutamiento adaptativo maximizan el uso de este ancho de banda (es decir, las prestaciones de la red).

BIBLIOGRAFÍA

1. José Duato, Introduction to Network Architectures. In José Flich, Davide Bertozzi (editors), Designing Network On-Chip Architectures in the Nanoscale Era, Chapman and Hall, 2010.

2. Timothy M. Pinkston, José Duato, Appendix E: Interconnection Networks. In John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Elsevier Publishers, 4th edition, 2006.

3. José Duato, Sudhakar Yalamanchili, Lionel Ni, Interconnection Networks: An Engineering Approach, Morgan Kaufmann, 2002.

4. William Dally, Brian Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann, 2003.