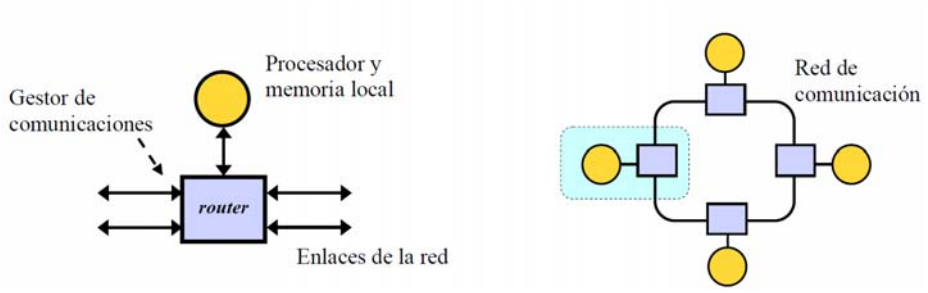
REDES FORMADAS POR ENCAMINADORES DE MENSAJES

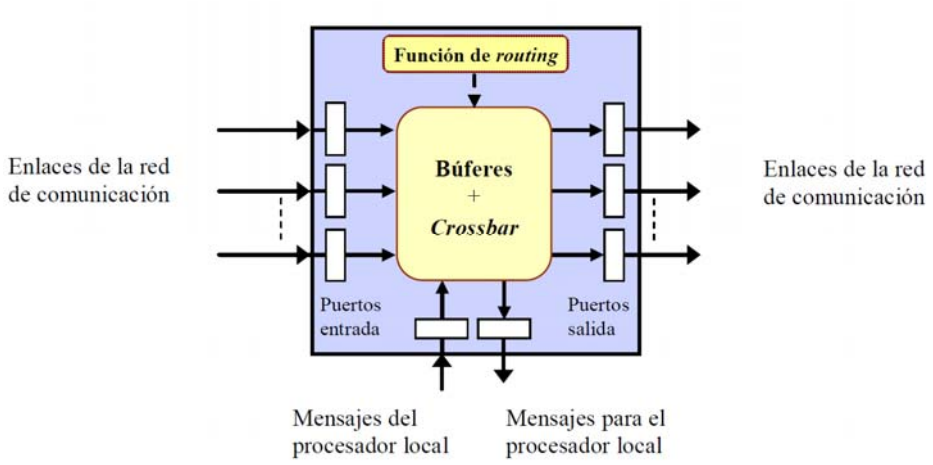
En las redes multietapa que acabamos de analizar, los enlaces entre conmutadores se establecen dinámicamente en función de las necesidades de comunicación, y los nodos y los dispositivos de la red están "separados". Tenemos un segundo tipo de redes, en el que cada nodo de la red utiliza un dispositivo propio para gestionar la comunicación, un encaminador de mensajes; estas redes se conocen como redes estáticas, Para formar la red, se conectan entre sí los dispositivos específicos de gestión de mensajes de cada nodo, los **encaminadores de mensajes** (*routers*), de acuerdo a una determinada topología. Este tipo de redes son las que habitualmente utilizan los sistemas paralelos actuales, sobre todo si el número de procesadores que se desea conectar es elevado.



1.- Encaminadores de mensajes

Como ya sabemos, la comunicación entre procesadores en sistemas paralelos de memoria distribuida se realiza mediante paso de mensajes. Para enviar un mensaje de un procesador a otro se pasa dicho mensaje al encaminador de mensajes local, desde donde irá avanzando, encaminador a encaminador, hasta el nodo destino.

La siguiente figura representa, de manera esquemática, un encaminador de mensajes típico. Por un lado, tenemos cierto número de puertos de entrada y salida, mediante los cuales se va a formar la red, más un puerto específico para la conexión con el procesador local. Por otro, un autómata que decidirá en cada caso el puerto de salida correspondiente a un mensaje que se ha recibido en uno de los puertos de entrada, bien para pasarlo a otro encaminador, bien para pasarlo al procesador local.



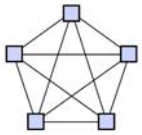
Es posible que un mensaje quede bloqueado en algún punto intermedio de su recorrido, debido a que no esté libre el camino de salida que necesita para seguir avanzando. Un poco más adelante concretaremos qué hay

que hacer en esos casos, aunque es habitual que el encaminador disponga de búferes para almacenar mensajes en esas circunstancias.

Un encaminador de mensajes no es sino un tipo de conmutador algo más complejo. El objetivo de ambos dispositivos es el mismo, conectar entradas y salidas para abrir un camino a los mensajes; es la organización de la red la que los hace algo diferentes. El encaminador de mensajes es un elemento más de los nodos que forman la red, no como los conmutadores de una red Omega, que son independientes de los nodos de la misma, como consecuencia de ello, por ejemplo, en el caso de las redes estáticas no todos los mensajes recorren el mismo número de pasos en la red: unos nodos están más cerca que otros.

2.- Topologías de red más utilizadas

Como ya hemos visto, la red ideal es la que conecta todos con todos, un *crossbar*, tal como el de la figura.



Es claro que se trata de una topología de difícil aplicación si el número de procesadores a conectar es elevado: el número de enlaces crece cuadráticamente y el grado de los nodos (número de conexiones) es grande (y no es constante).

Tenemos que analizar por tanto las alternativas más viables y que más se utilizan en los sistemas MPP actuales. En general, cada nodo de las topologías que vamos a analizar consta de procesador, memoria y encaminador de mensajes; los enlaces son siempre bidireccionales.

2.1.- Redes de una dimensión: la cadena y el anillo

Aunque no es una topología que se utilice en casos reales, analicemos como punto de partida dos redes de una sola dimensión: la cadena y el anillo.

cadena anillo

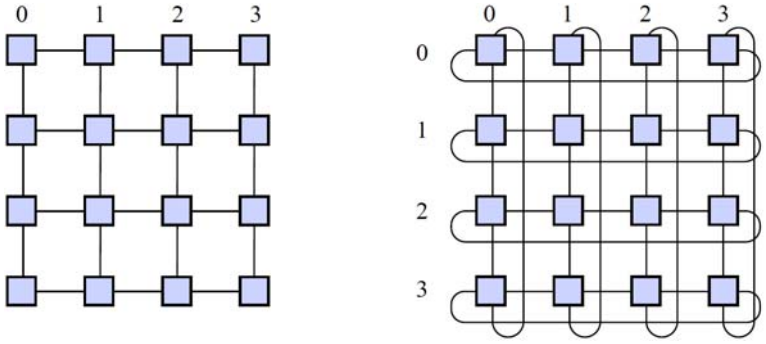
El grado de ambas redes es 2 (2 enlaces por nodo). El anillo es regular y simétrico, pero la cadena no. La tolerancia a fallos es baja: basta con que falle un enlace, cualquiera, para romper la cadena, o dos enlaces en el caso del anillo. Los parámetros de distancia (siendo P, el número de procesadores, par) son los siguientes (ver apéndice del capítulo):

	diámetro	distancia media
cadena →	$P-1$	$(P+1)/3 \quad \rightarrow P/3$
anillo →	$P/2$	$P^2/4(P-1) \quad \rightarrow P/4$

Las redes de una dimensión no son adecuadas para conectar un número alto de procesadores. Pero pueden generalizarse fácilmente a 2, 3 o más dimensiones.

2.2.- Mallas y Toros (*mesh, torus*)

Las mallas y los toros son las topologías resultantes de generalizar la cadena y el anillo respectivamente a *n* dimensiones. En la figura aparecen una malla y un toro de 2 dimensiones. Hemos representado redes cuadradas, pero el número de procesadores por dimensión puede ser cualquiera. Para obtener un toro a partir de una malla, basta con enlazar entre sí, formando anillos, los nodos de los bordes de la malla en cada dimensión (con otro tipo de conexión entre los nodos de los bordes, se obtiene otro tipo de redes).



En general, se utilizan mallas o toros de *n* dimensiones ($n = 2$ o 3), con *k* nodos por dimensión. Por tanto, el número total de nodos de la red es $P = k^n$. En el caso de dos dimensiones, $P = k \times k$ (si la red no es cuadrada, $k1 \times k2$).

El grado de ambas redes es $2n$ (hay dos enlaces por dimensión), un valor bajo. El toro es regular y simétrico, pero la malla no, ya que el grado de todos los nodos no es el mismo (por ejemplo, en dos dimensiones, los nodos de los vértices sólo tienen 2 enlaces, los de los lados 3, y el resto 4), por lo que los nodos no tienen la misma imagen de la red.

La tolerancia a fallos es alta, ya que hay muchos caminos para ir de un nodo a otro. En el caso peor, hay que quitar *n* enlaces en la malla (los enlaces de un nodo de uno de los vértices) y $2n$ en el toro (en dos dimensiones, los cuatro enlaces de cualquiera de los nodos) para romper la red; por tanto, la arco-conectividad es *n* y $2n$ respectivamente. Sin embargo, en general, se pueden eliminar (estropear) muchos más enlaces de la red sin que ésta pierda la conectividad entre todos sus nodos. Respecto al número de enlaces, una malla *tiene* $n \times k^{n-1} \times (k-1)$ enlaces, y un toro $n \times k^n$.

Para dividir la red en dos partes iguales (bisección), es necesario eliminar *k* enlaces en la malla y $2k$ enlaces en el toro (debido a los anillos).

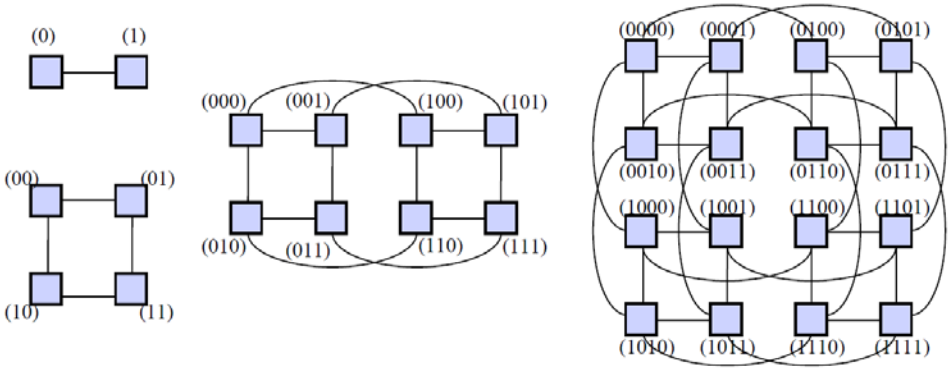
Los parámetros de distancia de ambas redes son los siguientes (*k* par; ver apéndice):

	diámetro	distancia media
malla →	$n \times (k-1)$	$n\frac{k^2-1}{3k}\frac{k^n}{k^n-1} \rightarrow n \times (k/3)$
toro →	$n \times k/2$	$n\frac{k}{4}\frac{k^n}{k^n-1} \rightarrow n \times (k/4)$

La distancia máxima de la malla corresponde a ir de un extremo a otro en cada dimensión; en el toro, en cambio, la distancia máxima por dimensión nunca es mayor que medio anillo. Simplificando el cálculo para el caso en el que el número de nodos sea grande, la distancia media en una malla es un tercio del número de nodos en cada dimensión, y en un anillo un cuarto.

2.3.- Hipercubos (*hypercube*)

Un hipercubo es una malla de *n* dimensiones que sólo tiene dos nodos por dimensión, por lo que también se le denomina *n*-cubo binario. En las siguientes figuras tenemos hipercubos de 1, 2, 3 y 4 dimensiones. Para generar un hipercubo de *n* dimensiones, hay que construir dos hipercubos de *n–1* dimensiones y enlazar, uno a uno, los nodos de la misma posición en cada red.



Cada nodo de la red tiene una dirección de n bits, (x_{n-1}, x_{n-2}, ..., x_1, x_0), en la que cada bit valdrá 1 o 0 en función de la posición que ocupe el nodo en la correspondiente dimensión. Así, el nodo (x_{n-1}, x_{n-2}, ..., x_1, x_0) está conectado con los nodos (/x_{n-1}, x_{n-2}, ..., x_1, x_0), (x_{n-1}, /x_{n-2}, ..., x_1, x_0), (x_{n-1}, x_{n-2}, ..., /x_1, x_0), (x_{n-1}, x_{n-2}, ..., x_1, /x_0) [/ = *not*]; es decir, con aquellos nodos cuya dirección difiere únicamente en un bit. Por ejemplo, el nodo 0000 está conectado con los nodos 1000, 0100, 0010 y 0001.

Un hipercubo de *n* dimensiones tiene 2^n nodos. Un hipercubo de *P* nodos es de $log_2 P$ dimensiones. El grado de la red es *n* (o sea, $log_2 P$), la dimensión de la misma, y no es constante, ya que crece con el tamaño de la red. El hipercubo es simétrico y regular. El número de enlaces es alto, $(P/2) \times log_2 P$, y en la misma medida es alta también la tolerancia a fallos; la arcoconectividad es *n* (hay que eliminar los n enlaces de un nodo dado para romper la red).

La bisección de un hipercubo es de *P/2* enlaces (para formar un hipercubo de *P* nodos unimos uno a uno los nodos de dos hipercubos de *P/2* nodos).

Los parámetros de distancia del hipercubo son los siguientes (ver apéndice):

	diámetro	distancia media
hipercubo →	n	$n \times 2^{n-1} / (2^n - 1) \rightarrow n / 2$

La distancia máxima es el número de dimensiones, ya que sólo se puede dar un paso en cada dimensión; si el número de nodos es grande, la distancia media resulta ser la mitad del número de dimensiones.

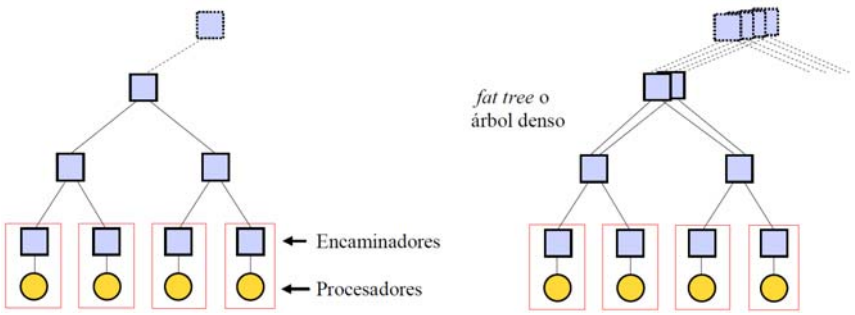
Los hipercubos presentan dos inconvenientes claros. Por una parte, el grado no es constante, lo que quiere decir que el número de enlaces del procesador (mejor, del interfaz de la red) no es constante, sino que varía en función del tamaño de la red. Por tanto, es complicado ampliar la red, ya que habría que cambiar todos los elementos de la red (o tener prevista desde el principio la posible ampliación del sistema). Además, la construcción de hipercubos de muchos nodos no es sencilla, ya que el cableado de la red es muy denso. Por otra parte, no se puede construir un hipercubo con cualquier número de procesadores, sino sólo con potencias de 2. Por ejemplo, 512 o 1024 procesadores, pero no 800.

Topologías tales como mallas, toros e hipercubos pueden englobarse dentro de una clase más general, denominada ***k-ary n-cube***. Se trata de redes de n dimensiones, con *k* nodos en cada dimensión. Los nodos de cada dimensión se unen mediante un anillo (o una cadena) y los enlaces son unidireccionales (por ejemplo, un toro 4×4 sería un *4-ary 2-cube*).

2.4.- Árboles y árboles densos (*fat tree*)

Otra topología muy utilizada en los sistemas paralelos MPP es el árbol. Un árbol puede tener estructuras diversas; la más utilizada consiste en disponer los procesadores en las hojas del árbol y los encaminadores de mensajes en el resto de los nodos, tal como aparece en la siguiente figura.

El grado del árbol es *k*, el número de nodos que salen de cada nodo (como excepción, el grado no representa en este caso el número de enlaces). Así, el número de niveles del árbol resulta ser $log_k P$. En la figura se representa un árbol binario, con $k = 2$.



En principio, la red no es regular, ya que el nodo raíz y las hojas sólo tiene dos enlaces, mientras que los intermedios tienen tres. Sin embargo, si los procesadores únicamente están en las hojas del árbol, la visión que tiene cada uno de ellos de la red es la misma, por lo que podríamos decir que, para los procesadores, es regular.

Considerando el tráfico de paquetes, la red es muy desequilibrada. Por ejemplo, todo el tráfico que se genere en la mitad derecha del árbol, y que vaya dirigido a procesadores situados en la mitad izquierda, tiene que utilizar el encaminador de mensajes de la raíz, zona de la red que va a estar muy saturada. Además, si se estropeara un enlace de ese nodo o el propio encaminador, la red quedaría inconexa. Para superar ese problema, las redes en forma de árbol que se utilizan en la realidad disponen de mayor cantidad de recursos —enlaces y encaminadores— según se avanza hacia la raíz, con lo que la capacidad de gestionar mensajes es mayor en las zonas en las que el tráfico va a ser mayor. A este tipo de árbol se le conoce como árbol denso (*fat tree*). Por tanto, la bisección de un *fat tree* es *P/2*.

Los parámetros de distancia de un árbol son los siguientes (grado k; ver apéndice):

	diámetro	distancia media
árbol →	$2 \log_k P$	$\frac{2P}{P-1}\log_k P - \frac{2}{k-1}$ $\rightarrow 2 \log_2 P - 2 \quad (k=2)$

En las redes implementadas mediante conmutadores, como las redes Omega, los diferentes componentes que forman la red están perfectamente diferenciados —por un lado, los procesadores y, por otro, los conmutadores— y no hay una relación directa entre ellos. En el caso de las redes estáticas, en cambio, cada procesador (o, tal vez, un conjunto pequeño de ellos) utiliza un encaminador de mensajes privado para gestionar la comunicación. Los árboles que hemos analizado toman la apariencia de redes dinámicas, porque los procesadores sólo están en los nodos hoja del árbol; los demás elementos de la red sólo se encargan de gestionar los mensajes. De hecho, se puede demostrar que los árboles densos y las redes *butterfly* son isomorfos ("equivalentes"). La única diferencia es que para llegar al destino en una red *butterfly* es necesario atravesar toda la red (la distancia siempre es la misma), pero en los árboles no siempre es necesario llegar hasta el nodo raíz, porque se puede volver hacia atrás en cualquiera de los encaminadores intermedios (y de esta forma hacer el camino más corto).