❖Tema 5. Multiprocesadores y Redes de Interconexión.

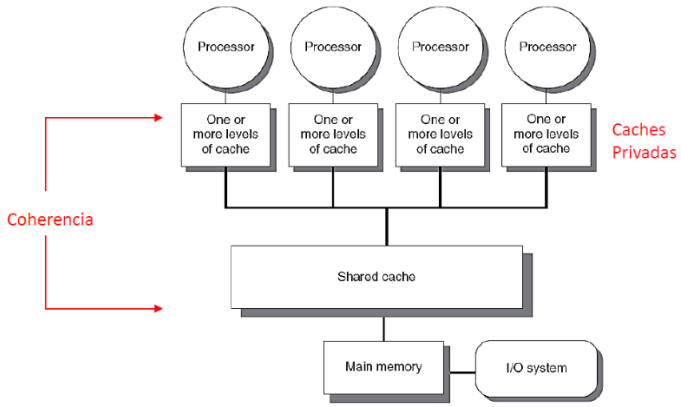
➢ 5.1 Multiprocesadores

Hemos visto en temas anteriores que una de las estrategias arquitectónicas para mejorar el rendimiento de los computadores es dotarlos de unidades de proceso independientes de forma que se pueda dividir la carga de trabajo en paralelo. Los chips estarán compuestos de varios procesadores (multiprocesador).

La primera cuestión que se nos plantea es cómo gestionar el tránsito de datos desde memoria al chip. Hay que tener en cuenta que pueden converger varias peticiones de uso de memoria por diferentes procesadores.

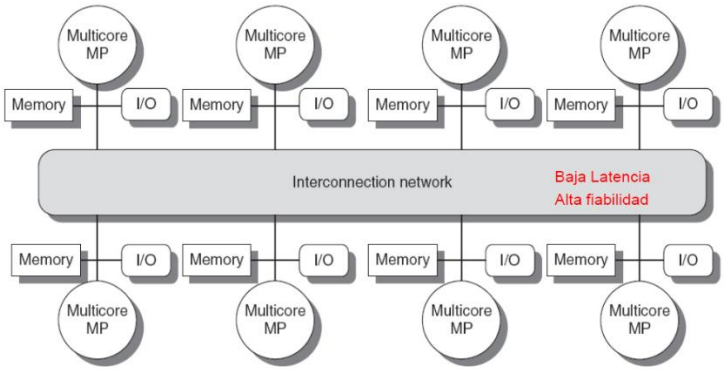
- Memoria centralizada. Nº reducido de procesadores.

Podemos separar las estrategias de comunicación en dos, un primer nivel más bajo con un Multiprocesador de CPU que tiene un reducido número de núcleos.



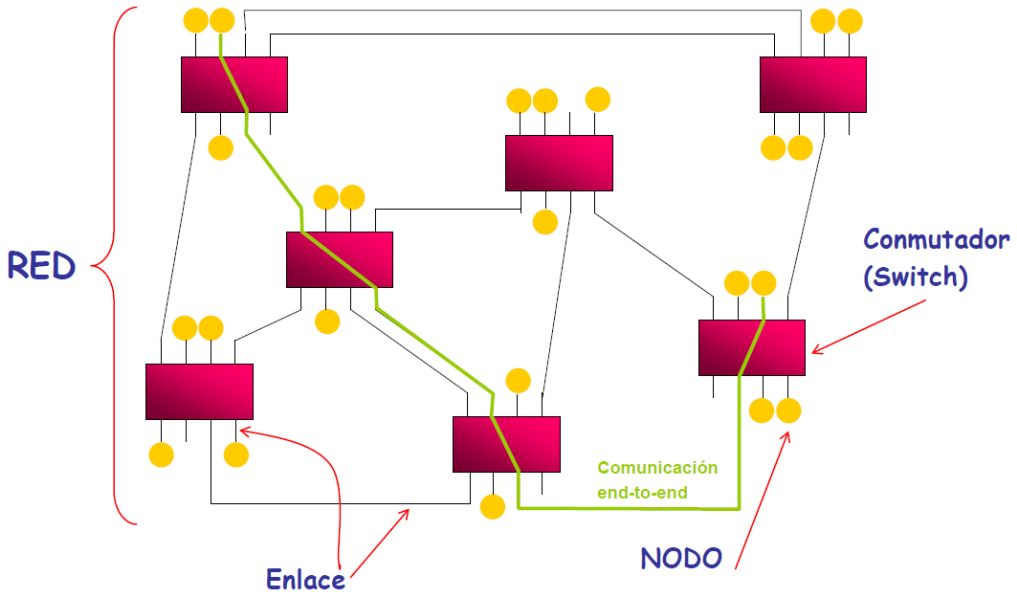
- Memoria distribuida. Nº elevado de procesadores.

Un segundo nivel más elevado con interconexiones entre multiprocesadores de memoria propia.



➢ 5.2 Redes de Interconexión.

Para la comunicación entre procesadores, ya sea dentro del mismo chip, conectados a una red de área local o comunicados a través de internet, diseñamos las redes de interconexión.



Nodo: centro de procesado, normalmente una CPU. Un nodo contiene, además de la CPU, un adaptador de interfaz para la red (network adapter).

Enlace: los enlaces son canales de comunicación, pueden tener soporte físico (cables de ADSL, fibra óptica, buses) o bien funcionar mediante transmisión de ondas en el aire (wi-fi, bluetooth, satélite).

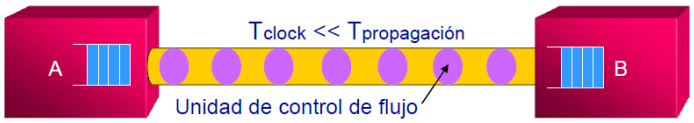
Conmutador: son centros neurálgicos de tránsito de información. Para evitar que las conexiones entre nodos tengan que ser "uno a uno" (lo que supondría un gasto excesivo en enlaces), se crean conmutadores que arbitran el paso de paquetes entre diferentes secciones de la red. Se les conoce también como Routers (enrutadores).

Mensaje: unidad lógica de comunicación en redes. Es la única unidad vista a nivel de aplicación de usuario.

Paquete: Unidad de transferencia entre interfaces de red (end-to-end). Es la menor unidad que contiene información de encadenamiento, esto quiere decir que los paquetes conforman mensajes y se debe indicar en todo momento la correlación entre paquetes para que el nodo de destino pueda recomponer el mensaje completamente (a veces incluso aunque los paquetes hubiesen llegado desordenados).

Los paquetes se encapsulan añadiéndoles cabecera y cola a la información neta del paquete. Estos añadidos llevan indicaciones de la estructura de la red, enrutado, correlación dentro del mensaje, etc.

5.2.1 Características de un enlace

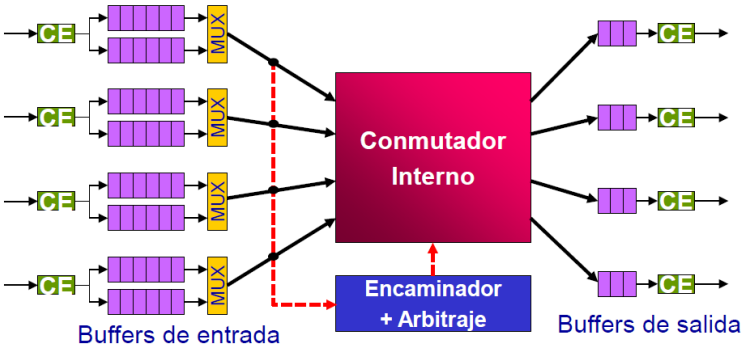


Un enlace es un canal, por este canal viajan los paquetes entre enrutadores o nodos. Para describir el comportamiento de un enlace lo caracterizamos con diferentes parámetros.

- Ancho del enlace: (w) en Bytes
- Velocidad de la señal (frecuencia): ($f = 1/\tau$) en herzios
- Ancho de Banda: $B = w \cdot f$
- Unidad de transferencia por ciclo: (Phit) (Physical unit = w)
- Unidad de control de flujo: (Flit) (Flow-control unit). Los paquetes están formados de Flits. Y los Flits a su vez por Phits.
- Enlace corto: solo un Phit en el enlace. En el enlace cabe muy poca información, pero como lo usamos para enlaces cercanos los Phit viajan rápido.
- Enlace largo: varios Phit en el enlace (enlace segmentado). Como la distancia es mayor tarda más, por eso se segmenta el enlace para que quepan varios phit (y también varios Flit).

Los mensajes se transfieren por paquetes que están formados por Flits. El primer Flit de un paquete es su cabecera, el último es su cola.

5.2.2 Conmutador/Router



Los conmutadores son centros de distribución de mensajes. Disponen de una lógica de arbitraje para que no se desvíen los paquetes del mismo mensaje, aunque existe una modalidad de enrutado adaptativo que puede mandar paquetes por distintos caminos si se detectan cuellos de botella (encaminamiento adaptativo).

Están caracterizados por el número de canales de entrada/salida y por la lógica de control.

5.2.3 Parámetros de la Red

Una red de interconexión se puede representar por un grafo de conmutadores y nodos unidos por canales de interconexión (dibujo de página 2).

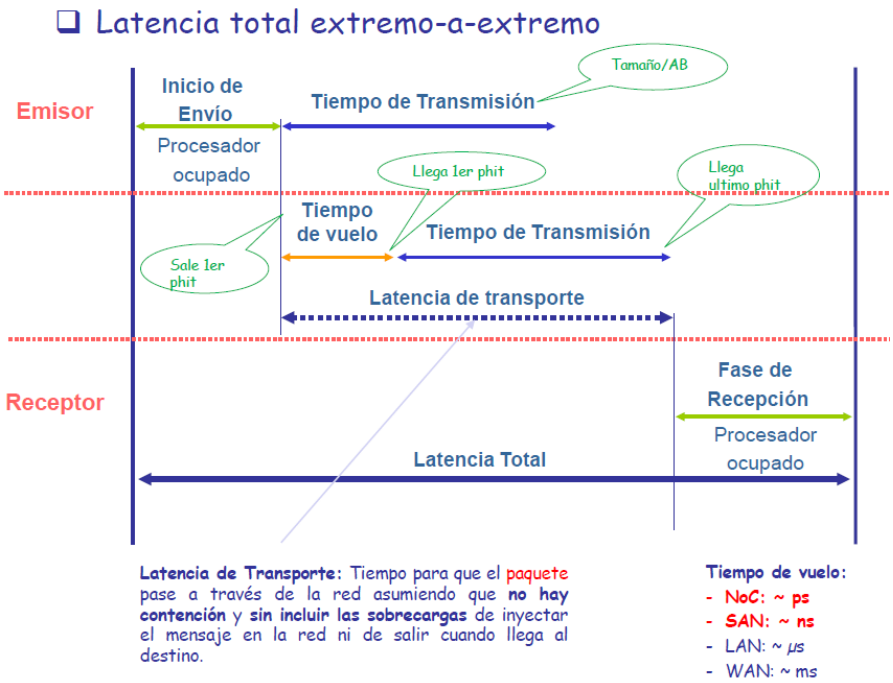
Ruta: Cuando un mensaje es enviado de un nodo a otro debe recibir una ruta, pero como los mensajes son fragmentados en paquetes es posible que paquetes del mismo mensaje viajen por rutas diferentes. Lo importante es que todos los paquetes lleguen a su destino para poder reconstruir el mensaje.

Topología: este parámetro nos habla de la estructura física de interconexión. Cuando hablamos de topología hablamos de la "forma" en que está fabricada la red. Si es regular, irregular, estática, dinámica, etc.

Encaminamiento: qué rutas pueden seguir los paquetes a través del grafo. Puede ser determinista o adaptativo.

Conmutación: describimos la forma en que se conmutan los paquetes o los circuitos de los enrutadores.

Control de flujo: en qué momento los paquetes se deben mover, si tienen que esperar en un enrutador o pueden continuar por un canal (que debe estar libre).



El tiempo de Vuelo es el tiempo neto que tarda en viajar una unidad de Phit entre el emisor y el receptor (lo recibe la tarjeta de red). El tiempo de Vuelo del primer Phit se podría equiparar al tiempo de latencia de las unidades segmentadas del VMIPS.

El procesador del receptor no se puede poner a trabajar hasta que el Network adapter haya recibido el mensaje completo.

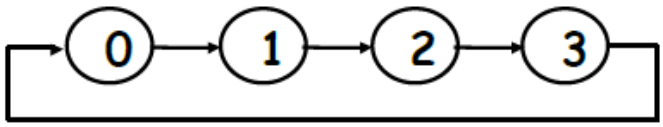
➢ 5.3 Redes Estáticas.

En las redes estáticas los nodos están conectados según unas reglas de adyacencia predefinidas, tienen una cierta toma de decisiones de encaminamiento, pero mediante reglas aritméticas básicas. No presentan routers/conmutadores de estructura compleja.

d → Grado de un nodo/conmutador. Número de canales de entrada/salida del nodo/conmutador.
D → Diámetro de la red. Es el camino más largo posible que tendría que recorrer un paquete siguiendo una ruta óptima. Viene a especificar la distancia máxima entre nodos, el número de aristas que tiene que recorrer.
F → Función de interconexión que describe el nodo al que se va a enviar. Puede tener varias modalidades dependiendo del número de enlaces (d) de que disponga el nodo.
$Simetría$ → si es simétrica todos los nodos son idénticos.

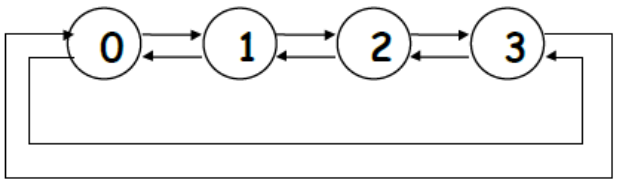
5.3.1 Anillo unidireccional de p nodos.

Es como una lista enlazada de conexión. Cada nodo puede enviar al siguiente (MOD P).



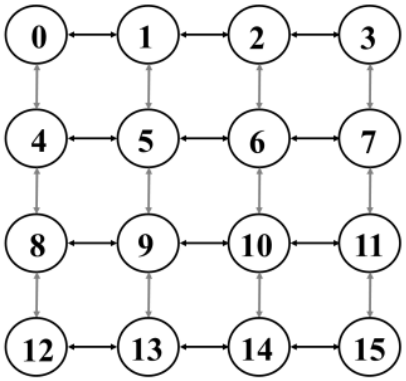
$$F(i) = (i+1) mod\ p \ \ , \ \ 0 \le i \le p$$
$$d = 1 \ \ , \ \ D = p - 1$$
$$Simetría = Sí$$

5.3.2 Anillo bidireccional de p nodos.



$$F_{+1}(i) = (i+1) mod\ p \ \ , \ \ 0 \le i \le p-1$$
$$F_{-1}(i) = (i-1+p) mod\ p \ \ , \ \ 0 \le i \le p-1$$
$$d = 2 \ \ , \ \ D = p/2 \ \ (división\ entera)$$
$$Simetría = Sí$$

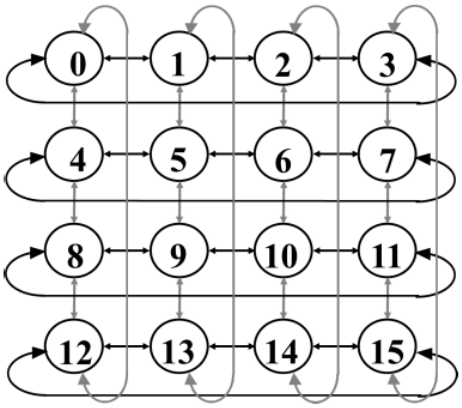
5.3.3 Malla 2D de (rxr) nodos



$p = r \cdot r$ Es bidireccional.
$$F_{+x}(i) = (i+1)$$
$$F_{-x}(i) = (i-1)$$
$$F_{+y}(i) = (i+r)$$
$$F_{-y}(i) = (i-r)$$
$$d = 4 \ \ , \ \ D = 2(r-1)$$
$$Simetría = No$$

Las operaciones tienen restricciones de frontera, los nodos de los laterales tienen menos canales que los nodos centrales. Cada nodo tiene al menos $d = 2$ enlaces.

5.3.4 Toro 2D de (rxr) nodos



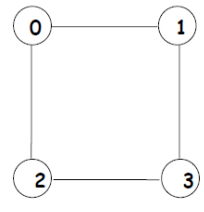
$p = r \cdot r$ Es bidireccional.

$$F_{+x}(i) = (i+1) mod\ r + (i/r) \cdot r$$
$$F_{-x}(i) = (i-1) mod\ r + (i/r) \cdot r$$
$$F_{+y}(i) = (i+r) mod\ p$$
$$F_{-y}(i) = (i-r+p) mod\ p$$
$$d = 4 \ , \ D = 2 \cdot (r/2)\ (división\ entera)$$
$$Simetría = Sí$$

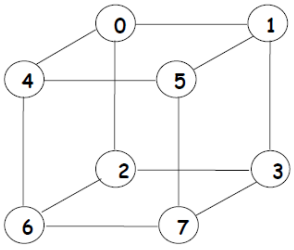
5.3.5 Hipercubo binario ó n-cubo binario.

$p = 2^n$ Nodos. $Simetría = Sí$

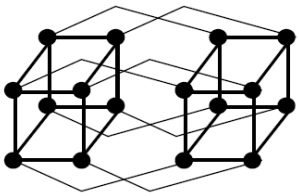
Cada nodo de la red tiene una dirección expresada en binario natural con n bits. La adyacencia se obtiene de la misma forma que en un mapa de Karnaugh, permutando uno de los n bits. Por lo tanto cada nodo tiene n vecinos con canales bidireccionales.



2-cubo d=2, D=2



3-cubo d=3,D=3


4-cubo d=4,D=4

5.3.6 n-cubo k-ario.

Es una evolución del hipercubo binario. En vez de base 2 se usa una base k. $p = k^n$ Nodos. Cada nodo tiene $2n$ vecinos porque a cada dígito (con $k > 2$) se le puede sumar o restar. Ejemplo:

Caso particular: Toro 2D 4x4 como 2-cubo 4-ario

Cada nodo: dir con n=2 dígitos en base k=4

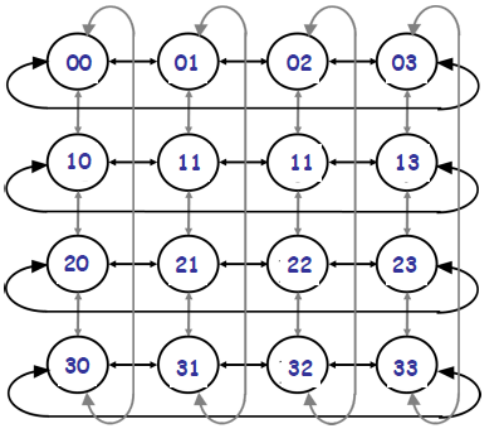
Funciones de interconexión: 2n=4 por nodo

$F_{1+}(p,q) = (p+1 \bmod 4, q)$ $F_{0+}(p,q) = (p, q+1 \bmod 4)$

$F_{1-}(p,q) = (p+4-1 \bmod 4, q)$ $F_{0-}(p,q) = (p, q+4-1 \bmod 4)$

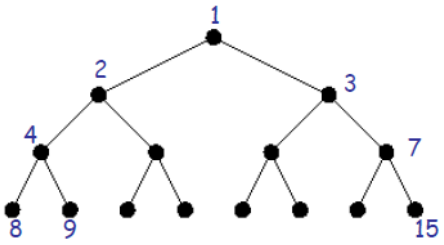
Ejemplos:

- $F_{1+}(3,0) = (0,0)$ $F_{0+}(3,0) = (3,1)$
- $F_{1-}(3,0) = (2,0)$ $F_{0-}(3,0) = (3,3)$

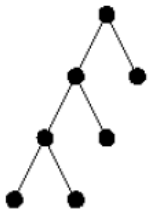


MACH MACH MACH MACH

5.3.7 Árboles binarios

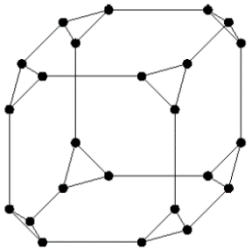


Equilibrado



No Equilibrado

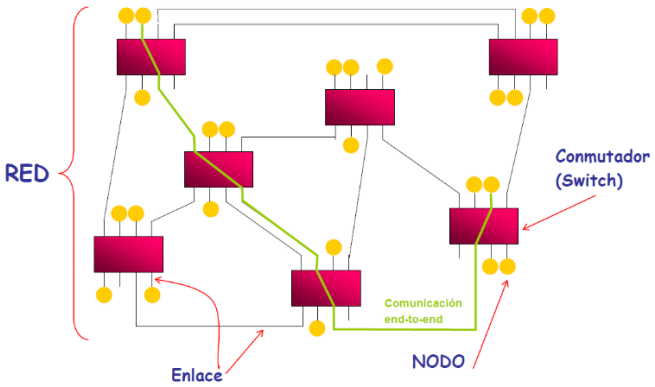
5.3.8 Topologías Híbridas



➢ 5.4 Redes Dinámicas.

Los EP(nodos) se conectan a un conmutador/Router reconfigurable de forma dinámica. Ahora los vértices del grafo son los conmutadores y las aristas los enlaces (los vértices no son los nodos).

Los nodos estarán "unidos" a un router físicamente, los routers serán el destino del mensaje en lo que se refiere a encaminamiento, y después el router se lo entrega al nodo/nodos propios destinatarios.



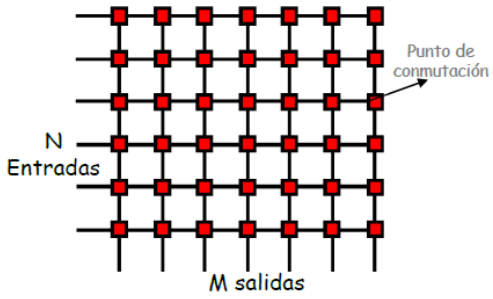
En la figura de ejemplo tenemos 7 vértices (routers) unidos por un conjunto de enlaces.

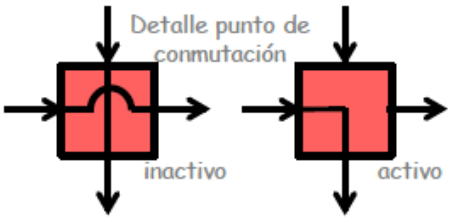
Cuando un router dispone de nodos (podría no tener ninguno, solo enlaces) se puede usar como origen/destino, si no tiene, simplemente es un puente de paso.

Los routers están unidos a los nodos "punto a punto", es decir, tienen un enlace propio para cada nodo de forma que si los nodos se quieren comunicar tienen que pasar por el router por muy cerca que estén.

5.4.1 Conmutadores Crossbar

Es un conmutador de N entradas y M salidas que permite un mínimo de (min(M,N)) conexiones punto a punto sin contención (sin esperas ni bloqueos). Es una "mini" red dentro del conmutador.





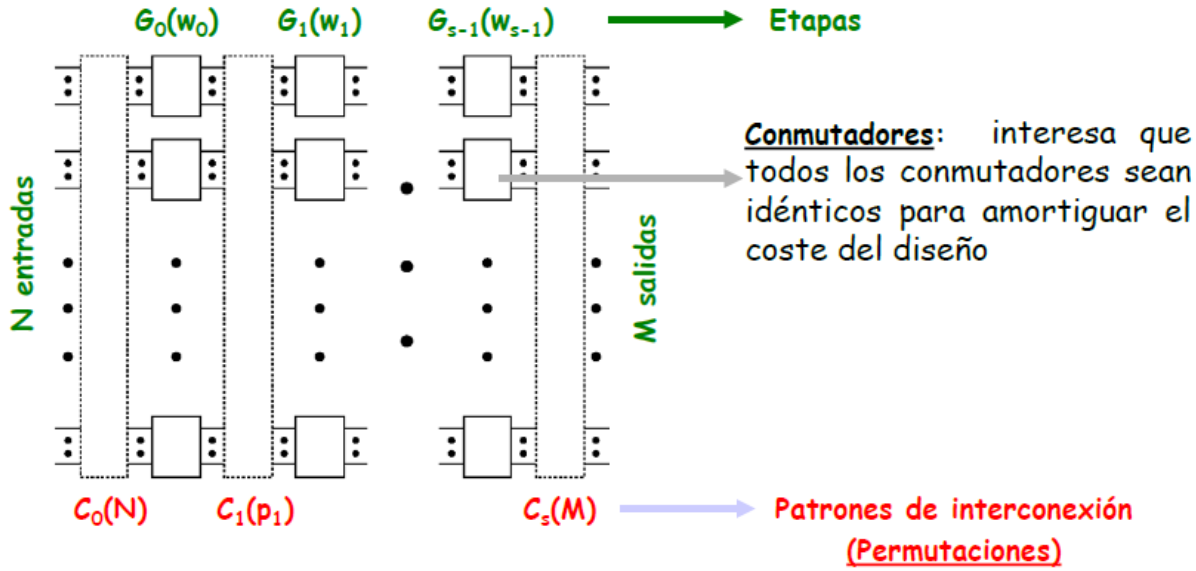
Cada "cruz" del entramado puede estar activada o desactivada. Si está desactivada simplemente transmite de arriba abajo y de izquierda a derecha lo que le llega de los vecinos. Sin embargo si está conectado realiza un corto entre lo que le llega de la izquierda y la salida inferior.

5.4.2 Redes dinámicas MIN

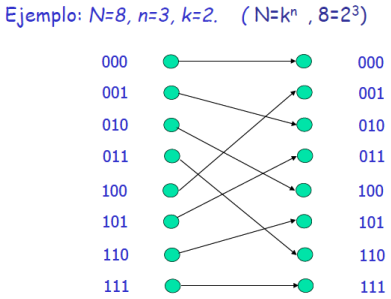
MIN ≡ Multistage Interconection Network

Se conectan dispositivos de entrada a dispositivos de salida a través de varias etapas compuestas por conmutadores. El rutado de los enlaces se hace según un protocolo de permutaciones que define el tipo de red.

Permutaciones
- Perfect Shuffle
- Butterfly
- Exchange



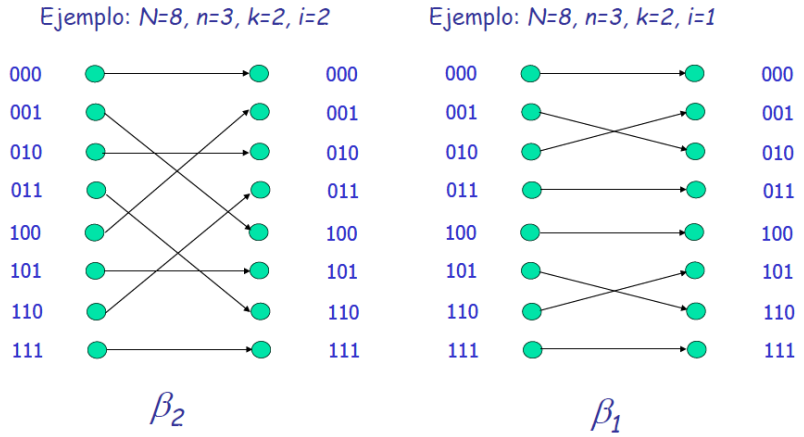
Perfect Shuffle (σ)



Se numeran las Entradas y las Salidas con n dígitos en base k. Se realiza la interconexión mediante la entrada y la salida que sale de hacer un ROTATE LEFT de los dígitos de la entrada.

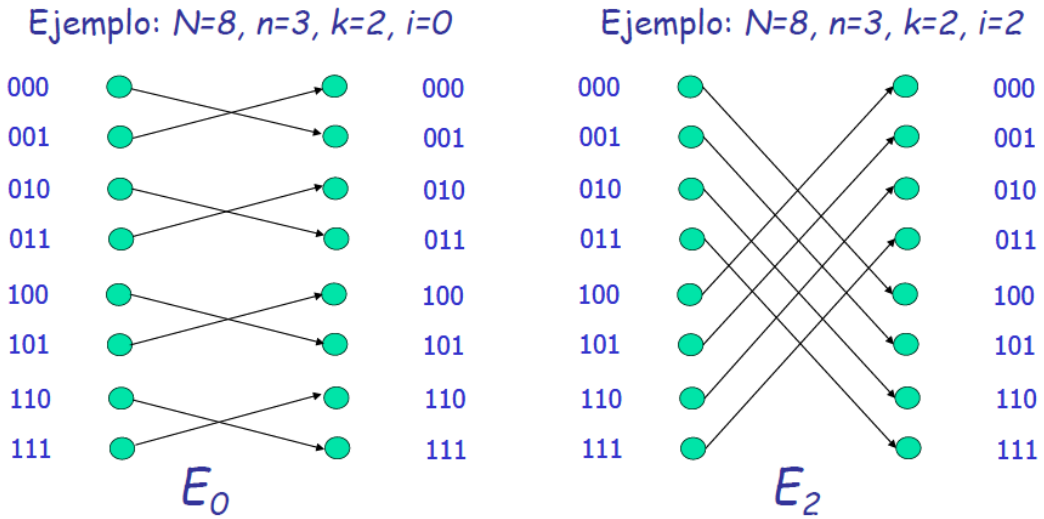
Permutación Butterfly (β)

La permutación β_i tiene un índice i que define su comportamiento. Para determinar la conexión entre entrada y salida se toman los n dígitos en base k de la entrada y se intercambian el dígito 0 con el dígito i. Siendo $i < n$.



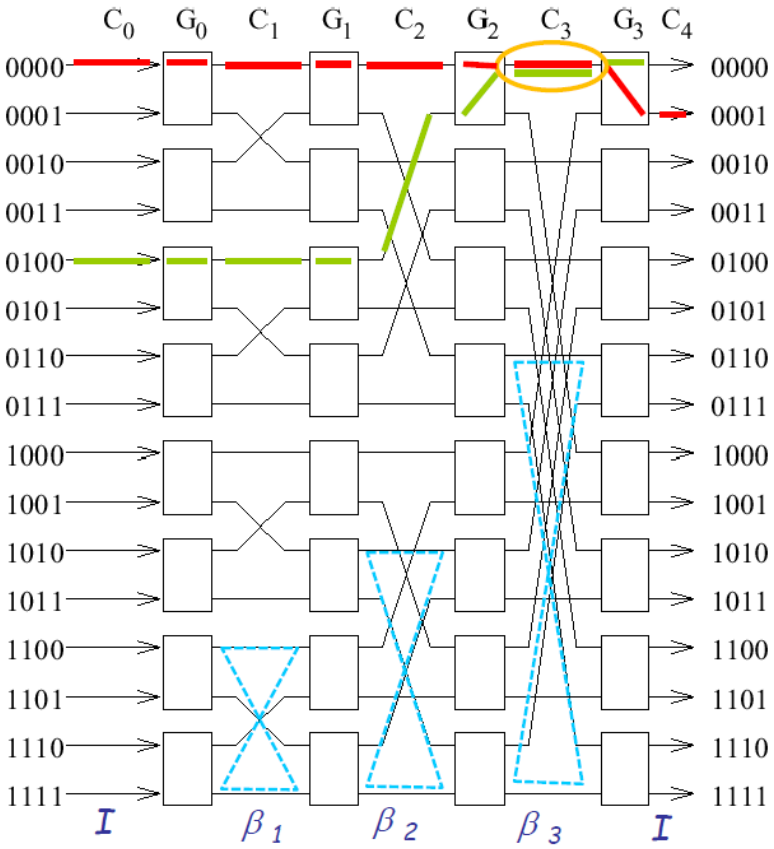
Permutación Exchange (E)

La permutación E_i también tiene un índice que lo define. La conexión se hace tomando los n dígitos en base k de la entrada y se modifica el dígito i. Siendo $i < n$. Si es binario, el bit i se niega.



5.4.3 Red MIN Butterfly

Usa permutaciones Butterfly entre cada etapa.



$$N = k^n$$

N → Número de Nodos.
k → Base de numeración. Grado.
n → Número de etapas.

Ejemplo: $k = 2,\ n = 4,\ N = 16$

La red tiene N nodos a la entrada y N nodos a la salida. En el ejemplo hay 4 arrays de conmutadores pertenecientes a las 4 etapas, pero las políticas de permutación son 5, tres son "internas a los conmutadores" y luego hay 2 más que son la entrada y la salida.

C_i → Algoritmo de permutación.

En una red Butterfly las permutaciones "exteriores" C_0 y C_n son siempre identitarias, eso quiere decir que son propiamente horizontales, no permutan. $C_0 = I$, $C_n = I$. Lo que entra a la etapa G_0 es el Nodo que está colocado a la misma altura así como lo que sale de la etapa G_3 se conecta directamente con los nodos de salida en horizontal.

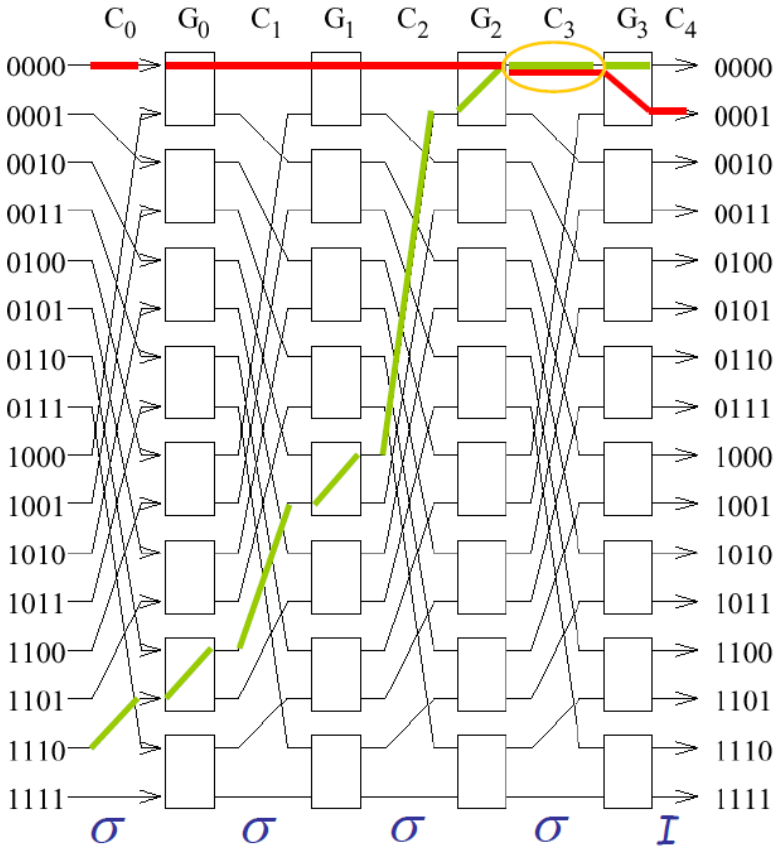
En las etapas intermedias usamos permutaciones $C_i = \beta_i$, por lo que en cada etapa es diferente.

La red debe garantizar que todos los N nodos de entrada puedan comunicarse con los N nodos de salida realizando algún trazado por los conmutadores. Esto provoca que haya comunicaciones incompatibles en cuestión de uso de canales. Por ejemplo, la ruta del 0000 al 0001 no es compatible con la ruta 0100 al 0000, los dos usan el mismo canal en C_3 (ver dibujo).

Los conmutadores 2x2 situados en las etapas $G_0 - G_3$ pueden cruzar las señales internamente en función de la información de rutado.

5.4.4 Red MIN Omega

Usa permutaciones Perfect Shuffle.



$$N = k^n$$

N → Número de Nodos.
k → Base de numeración. Grado.
n → Número de etapas.

Ejemplo: $k = 2,\ n = 4,\ N = 16$

La red tiene N nodos a la entrada y N nodos a la salida. En el ejemplo hay 4 arrays de conmutadores pertenecientes a las 4 etapas, pero las políticas de permutación son 5, tres son "internas a los conmutadores" y luego hay 2 más que son la entrada y la salida.

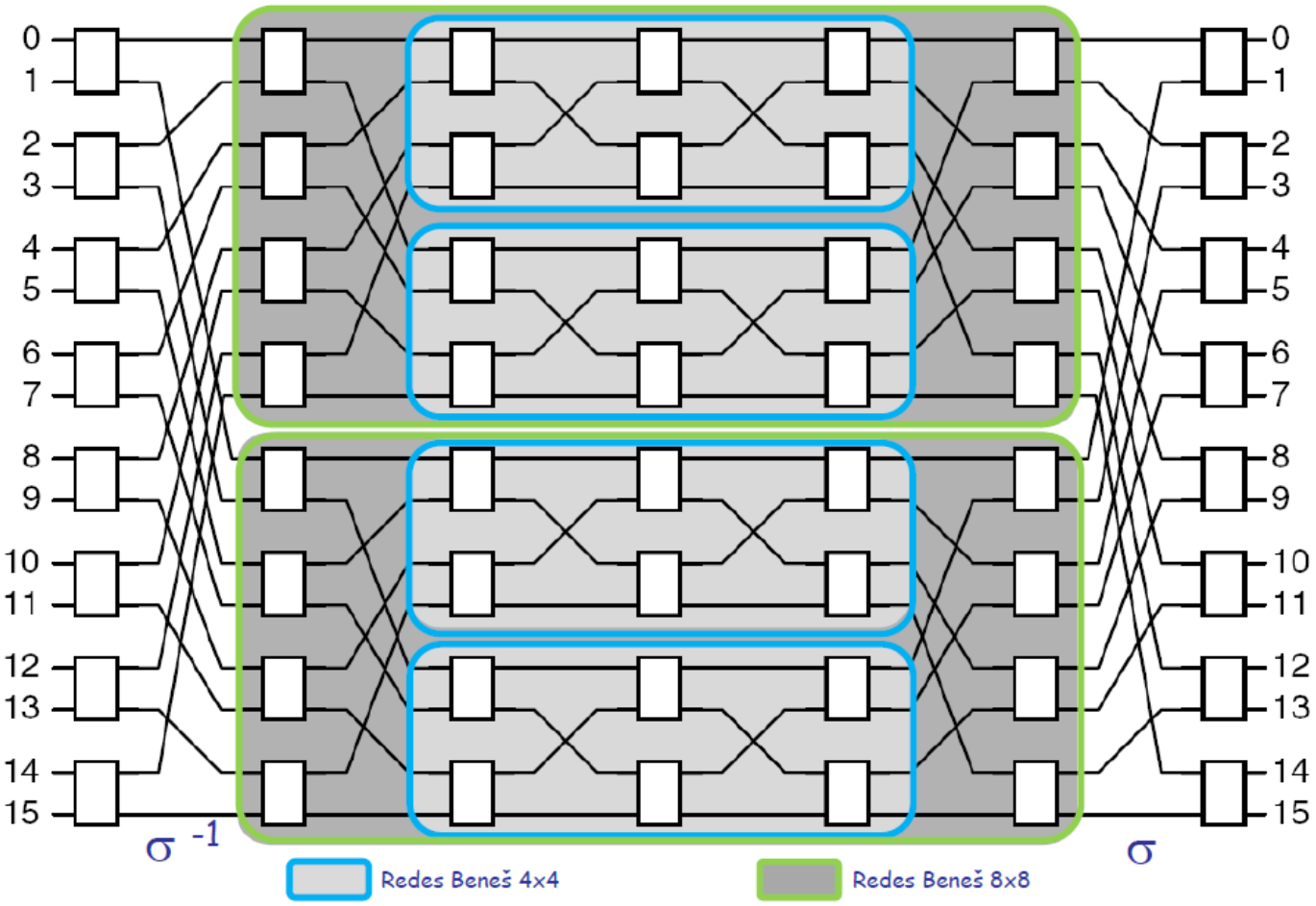
C_i → Algoritmo de permutación.

En la red Omega solo la última permutación es identitaria $C_n = I$. La permutación de entrada C_0 es "sigma" así como todas las intermedias. Como el perfect shuffle no tiene índice, es una rotación, todas las permutaciones son iguales en cada etapa $C_i = \sigma$.

Volvemos a tener problemas de bloqueo en determinados caminos cuando algunos nodos se intentan comunicar a la vez. En este caso la comunicación 0000 con 0001 no es compatible con 1110 al 0000.

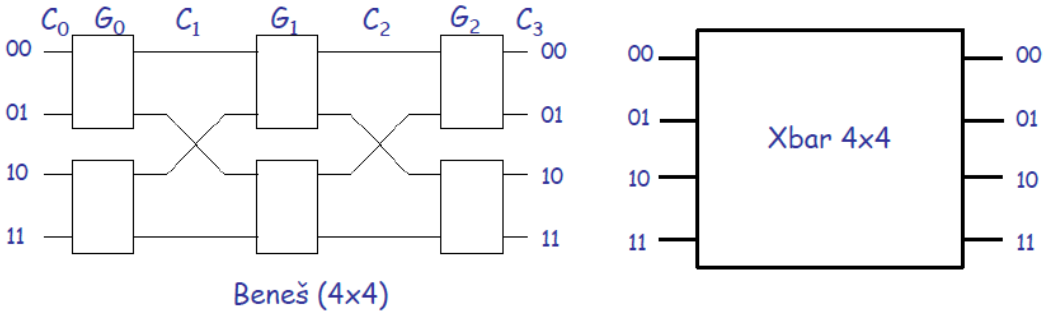
5.4.5 Red MIN Benes

La red Benes es una red más compleja, tiene más etapas que las redes Butterfly y Omega, pero consigue evitar el interbloqueo entre canales que sufren las anteriores.

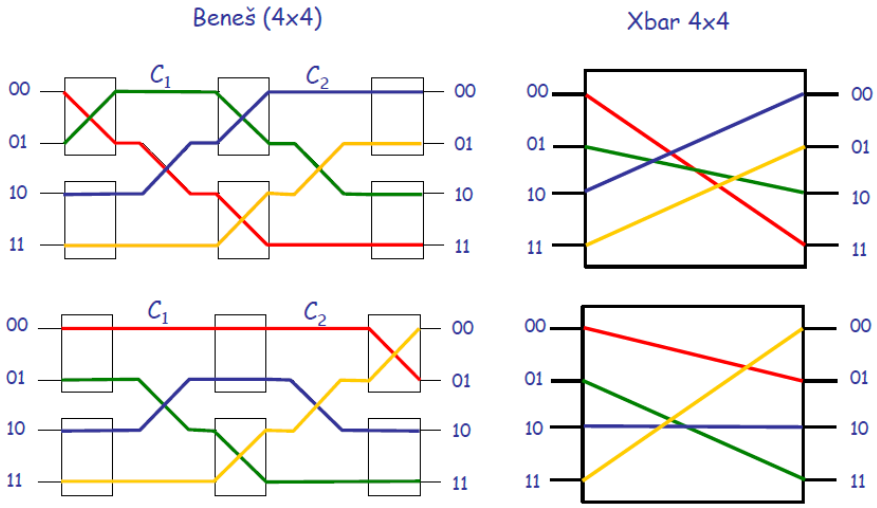


Para conectar N nodos en vez de tener $\log_k(N)$ etapas tiene $[\log_k(N) + (\log_k(N) - 1)]$. Se usan conexiones sigma (ROTATE LEFT) y sigma invertida (ROTATE RIGHT).

Esta red se suele fabricar de forma iterativa lo que facilita mucho su escalado. Vamos a ver más en detalle cómo funciona la red Benes 4x4 que es el bloque más sencillo.



La red Benes 4x4 funciona de forma idéntica a una Crossbar 4x4. Puede conectar todos con todos sin bloqueos.



Las permutaciones de entrada y salida son identidad y las permutaciones C_i intermedias son sima o sigma invertida.

$$C_1 = \sigma^{-1}$$

$$C_2 = \sigma$$

Las Xbar tienen precisamente esa característica, aunque su construcción interna es diferente.

➢ 5.5 Técnicas de Conmutación.

A la hora de realizar el rutado de los mensajes a través de las redes disponemos de diferentes métodos de toma de decisiones para los conmutadores (routers).

Existen dos estrategias bien diferentes para el envío de mensajes

- Conmutación de Circuitos
- Conmutación de Paquetes
  - Store & Forward
  - Cut-Through

Conmutación de Circuitos

- Se envía, previamente al mensaje, una cabecera que recorre todo el camino reservando los enlaces. Al alcanzar el destino retorna un ACK al origen.
- Cuando llega el ACK a origen se transmite el mensaje completo.
- Los enlaces de toda la ruta quedan bloqueados hasta que todo el mensaje recorre el trayecto completo.

Conmutación de Paquetes

- Cada decisión de rutado se realiza de forma dinámica mientras el mensaje progresa.
- Los enlaces que no se están usando en un momento dado quedan libres.
- Store & Forward espera a todo el mensaje en cada nodo. Cuando el mensaje está completo avanza al siguiente por donde pueda.
- Cut-Through envía cada FLIT desde el origen y avanzan de forma independiente sin esperarse.

5.5.1 Cálculo de tiempos con conmutación

Cuando se envía un mensaje de extremo a extremo de la red es interesante conocer el tiempo de latencia que consume el viaje del mensaje completo (sin contar lo que se tarda en procesar en origen o en destino, solo el viaje).

Para los cálculos se van a usar diferentes parámetros de la red y el rutado:

$h \rightarrow$ Número de enlaces que se van a atravesar en todo el viaje.
$n \rightarrow$ Número total de BITS que contiene el mensaje completo.
$B \rightarrow$ Ancho de banda de los enlaces. Medido en BITS/segundo.
$\Delta \rightarrow$ El tiempo necesario para la toma de decisión de la ruta del enlace siguiente y su conmutación.

Latencia con conmutación de circuitos

$$Tcc(n,h) = h \cdot \Delta + n/B$$

El primer sumando representa el tiempo necesario para que la cabecera recorra toda la red y vaya reservando los enlaces. El segundo sumando cuenta todo el recorrido neto del viaje del mensaje.

Latencia con Store & Forward

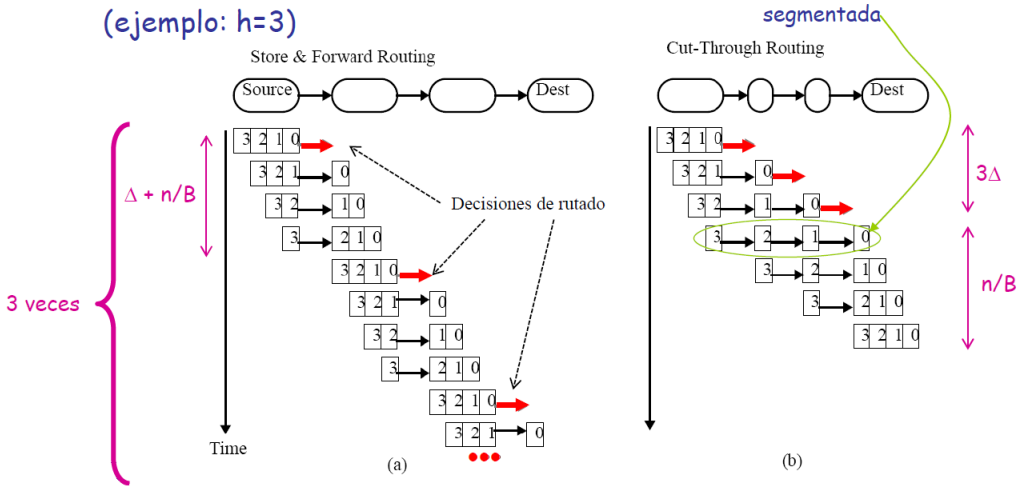
$$Tsf(n,h) = h \cdot \Delta + h \cdot (n/B)$$

La pega que presenta esta estrategia es que se pierde mucho ancho de banda debido a la espera del paquete en cada conmutador. Por eso el tiempo de transmisión del mensaje es mucho mayor, al no haber solapamiento entre en el uso de los enlaces.

Latencia con Cut-Through

$$Tct(n,h) = h \cdot \Delta + n/B$$

El tiempo promedio de esta estrategia es similar a la Conmutación de circuitos, con la ventaja de que los enlaces no quedan bloqueados todo el camino. Es como si tuviéramos una ruta segmentada.



➢ 5.6 Técnicas de Encaminamiento.

Cuando un mensaje tiene que seguir una ruta origen-destino se debe trazar un camino en la red que se compone de una secuencia de nodos. La configuración de este encaminamiento se puede hacer completa desde el origen (determinista) o se puede ir confeccionando a medida que avanzan los paquetes (adaptativo).

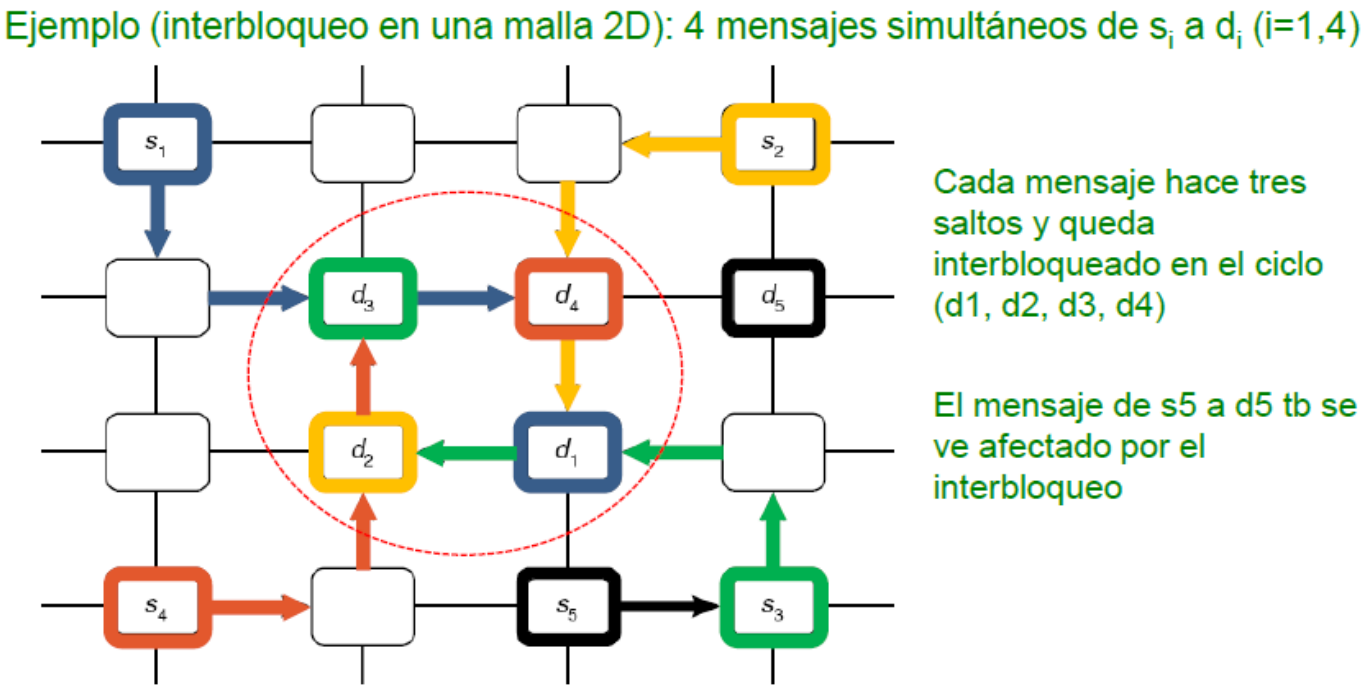
Lo que vamos a tratar aquí es la posibilidad de que, una vez se haya trazado un camino por varios mensajes, surjan problemas de interbloqueo o bucles infinitos.

Interbloqueo (Deadlock): un paquete queda a la espera de un evento que nunca puede ocurrir. Por ejemplo espera a que se libere un canal, pero el canal nunca se libera por estar bloqueado.

Espera indefinida (Starvation): un paquete está a la espera de un evento que puede ocurrir, pero que nunca ocurre. Normalmente por una mala gestión de la red.

Ciclos infinitos (Lifelock): un paquete está recorriendo nodos indefinidamente sin acabar de llegar a su destino.

Ejemplo de Deadlock (Red Malla 2D):



Ningún canal puede quedar libre porque ningún paquete termina de llegar a su destino y todo el centro de la red queda bloqueado.

Existen diferentes técnicas de encaminamiento para evitar los interbloqueos

- Selección de ruta en origen (se añade una cabecera con los nodos al mensaje)
- Dirigido por tablas (los nodos tienen tablas y los mensajes punteros que indexan las tablas)
- Enrutado de orden dimensional. (Dimension-order routing).

5.6.1 Dimension-order Routing

Vemos más en detalle esta técnica de enrutamiento. Se usa en redes conformadas por dimensiones (mallas, toroides e hipercubos).

Es una técnica adaptativa (los paquetes toman la decisión en tiempo de viaje), pero tiene una regla de encaminamiento que consiste en que el paquete no puede avanzar en una dimensión hasta que ha agotado el viaje en su desplazamiento en todas las dimensiones anteriores (que se ordenan según algún criterio, X Y Z o como sea).

Esta técnica evita el problema del interbloqueo, siempre acaban despachados los mensajes.