

Ingeniería de los Computadores

Unidad 4. Redes de interconexión.

Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Introducción

Redes de interconexión en supercomputación:

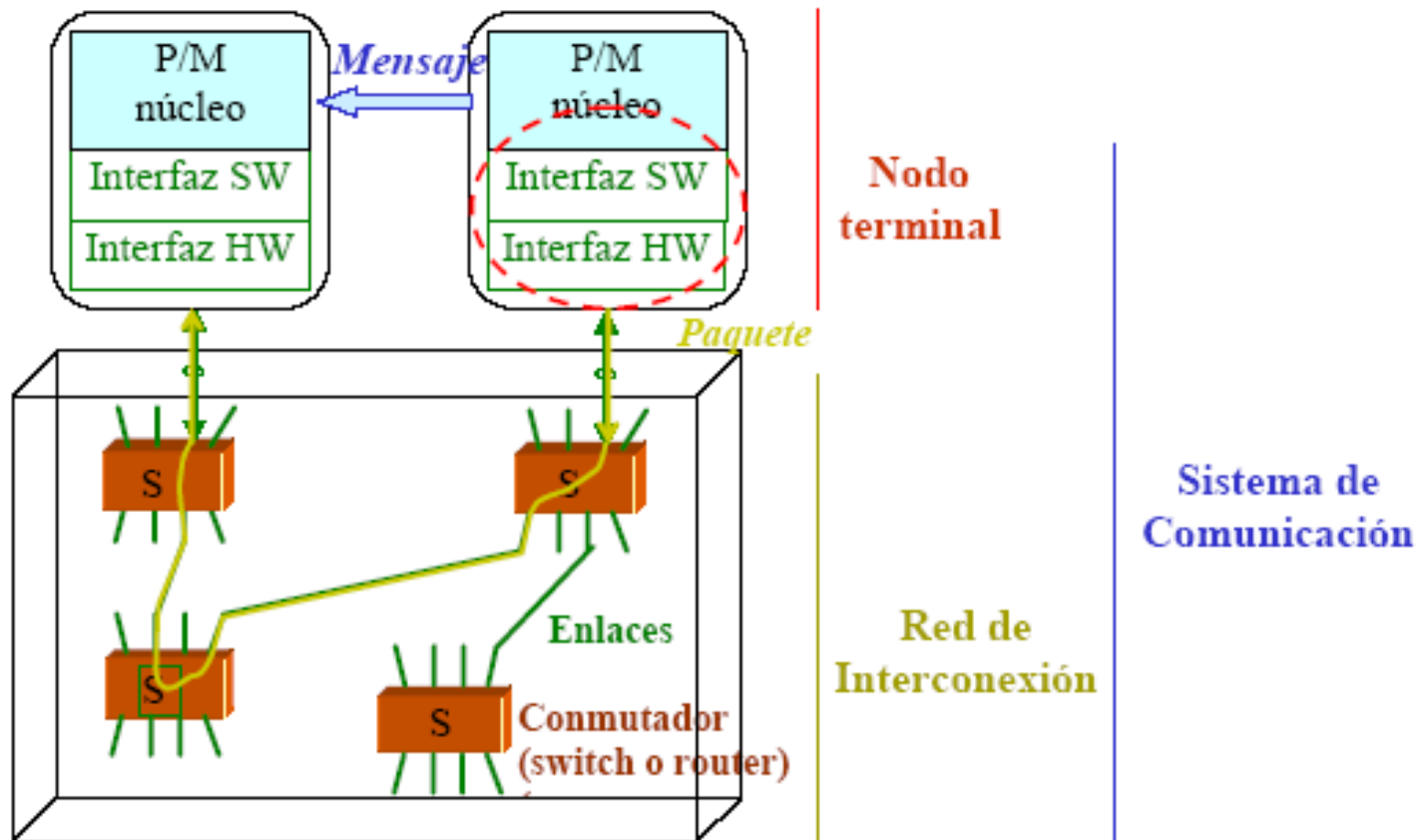
- Elemento fundamental en arquitecturas paralelas con **varios elementos de proceso que se comunican**
- La eficiencia en la comunicación es crítica: multiprocesadores, multicomputadores
- El diseño de la red condiciona: escalabilidad de la arquitectura, complejidad, tolerancia a fallos, etc.
- Aspectos relacionados: control de flujo y encaminamiento

Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Introducción

Estructura general del sistema de comunicación



Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Conceptos

Parámetros básicos:

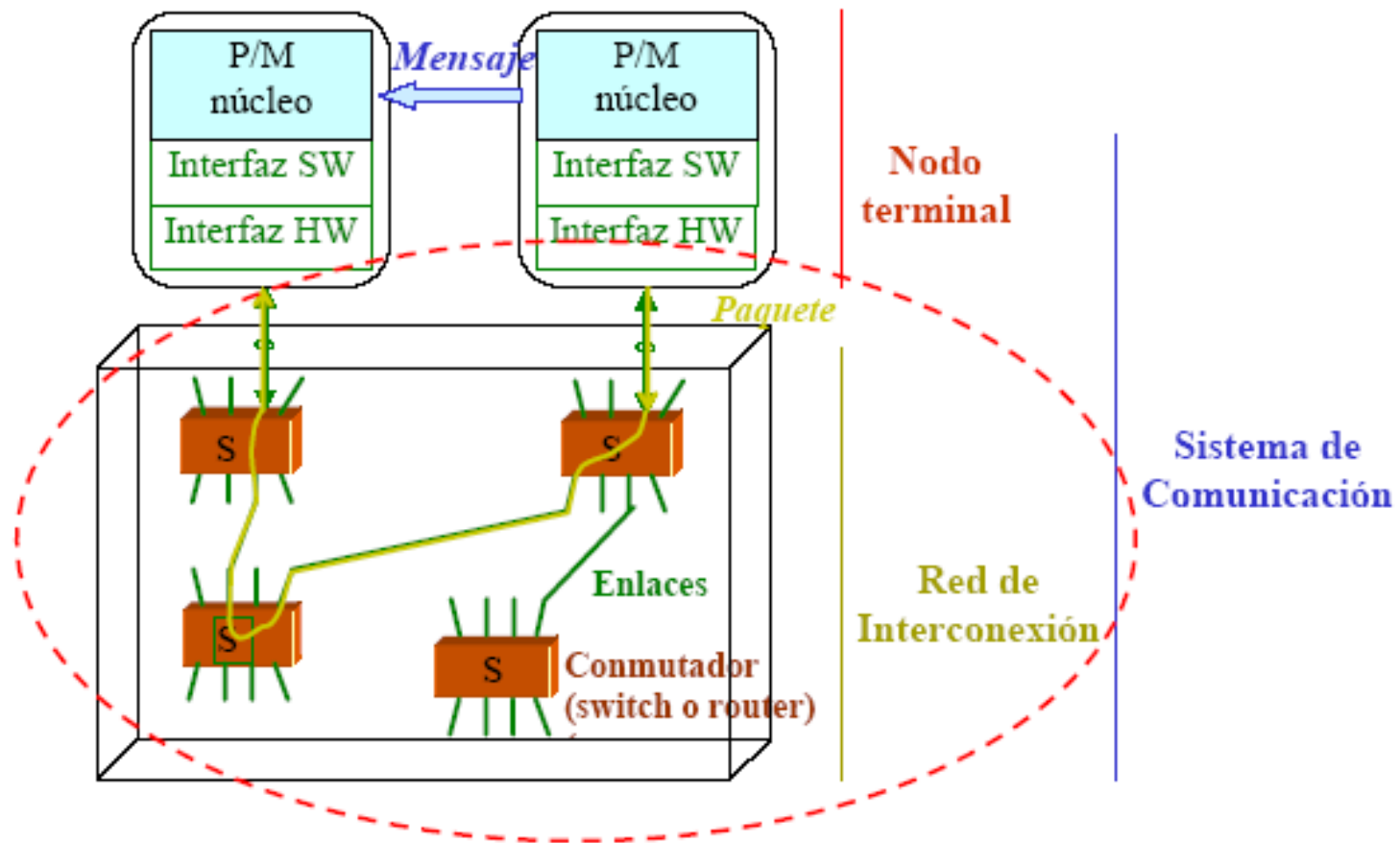
- Tamaño de la red: número de nodos (EPs, memorias, computadores)
- Grado del nodo (d - *degree*): número de canales de entrada y salida
- Nodos unidireccionales: grado de salida y grado de entrada
- Grado del nodo \rightarrow puertos de E/S (¿coste?)
- Diámetro de red: longitud máxima del camino más corto entre dos nodos cualquiera de una red.

Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Conceptos

Diseño de una red de interconexión



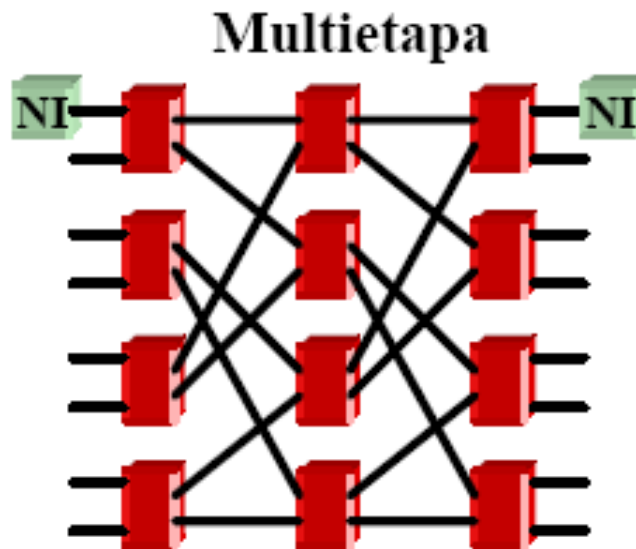
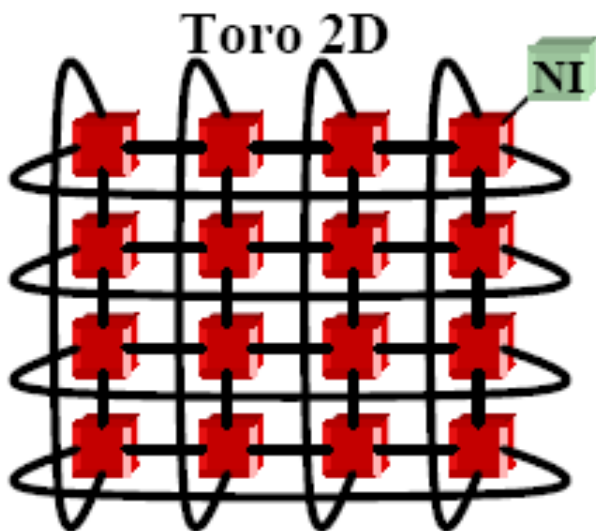
Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Conceptos

Diseño de una red de interconexión: **Topología**

- **Estructura** de interconexión física de la red. Se puede modelar mediante un grafo cuyos **vértices** son **conmutadores** o **interfaces de red** (a nodos de cómputo, a módulos de memoria, o a dispositivos de E/S) y los **aristas** son los **enlaces**.



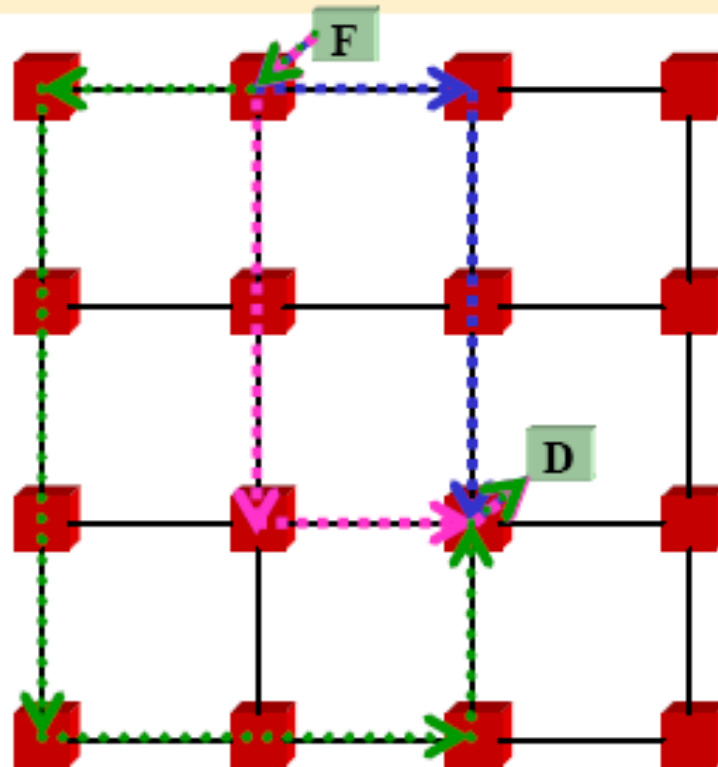
Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Conceptos

Diseño de una red de interconexión: **Encaminamiento**

- Determina el *camino* a seguir por un paquete desde el fuente al destino.



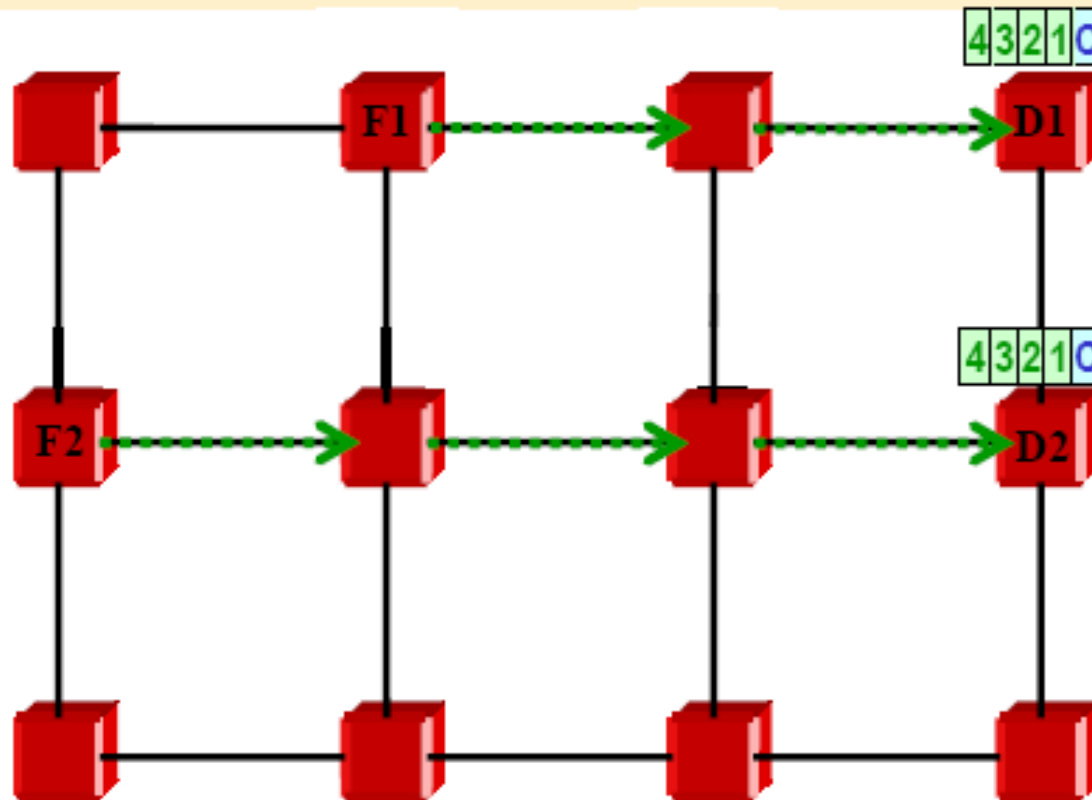
Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Conceptos

Diseño de una red de interconexión: **Estrategia de conmutación**

- Determina *cómo* los datos en un paquete atraviesan el camino hacia el destino.



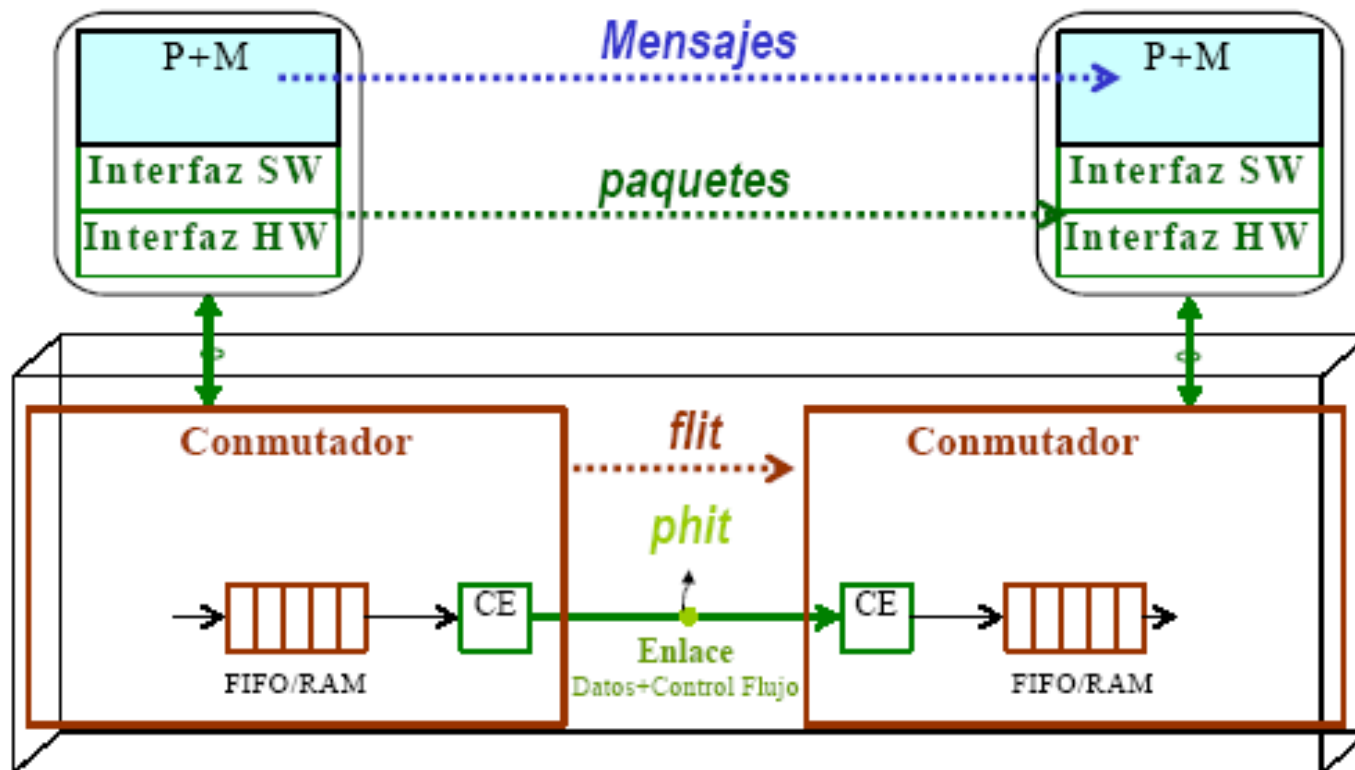
Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Conceptos

Diseño de una red de interconexión: **Control de flujo**

Determinan **cuándo** una unidad se mueven entre componentes del Sist. Comunicación, avanzando hacia el destino. **Arbitra** ante colisiones. Determina cómo y cuándo se asignan recursos (intra- e inter-conmutadores)

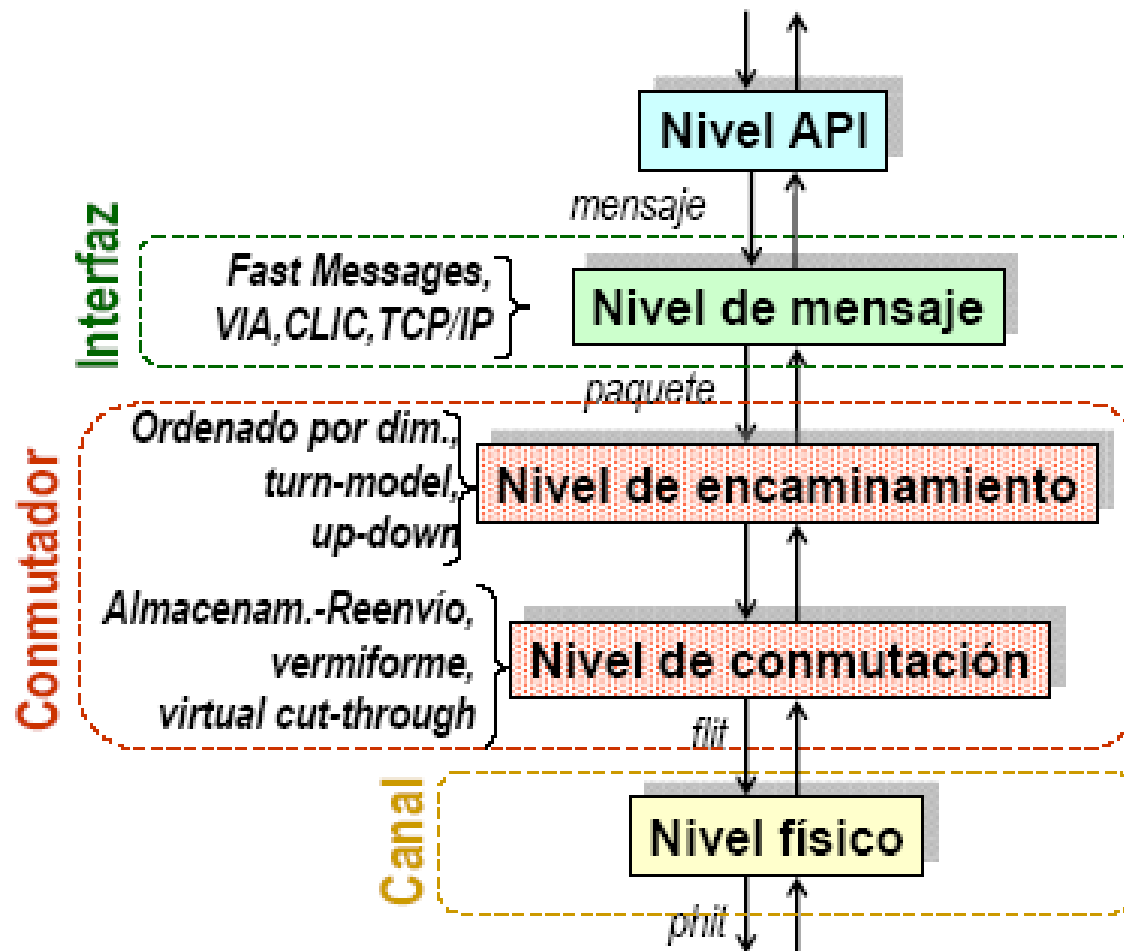


Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Conceptos

Diseño de una red de interconexión: **Niveles de servicios**



Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Clasificación

Clasificación de redes de interconexión

CLASES	Nº NODOS Y DISTANCIA	UTILIZACIÓN	DESARROLLO	EJEMPLOS
Diseñadas a medida	Nodos: unos pocos-decenas-cientos-miles	Multiprocesadores Multicomputadores Proc. matriciales	Arquitecturas de altas prestaciones.	-Cray X1 -Origin SGI -Sun Fire 15K
SAN: <i>System Area Network</i>	Nodos: decenas-cientos-miles Dist. decenas o cientos metros	Conecta comp. en habitación Interfaz software "ligera" (<i>lightweight</i>)	Redes a medida y LAN	-Estándares: <i>SGI, Infiniband</i> -OEM: <i>Myrinet, QsNet</i>
LAN: <i>Local Area Network</i>	Nodos: cientos Dist.<decenas km	Conecta comp. en edificio o campus	Estaciones de trabajo	-Fast Eth. -Gigabit Eth.
WAN: <i>Wide Area Network</i>	Nodos: miles Dist. miles km	Conecta comp. a nivel mundial	Telecomunicaciones	-ATM

Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Clasificación

Clasificación de redes de interconexión (1/2):

- Redes de **medio compartido**
 - Redes de área local
 - Bus de contención (Ethernet)
 - Token bus (Arcnet)
 - Token ring (IBM Token ring)
 - Bus de sistema (backplane bus) (Sun Gigaplane)
- Redes **directas** (estáticas basadas en router)
 - Topologías ortogonales (Malla, Toro, Hipercubo)
 - Otras topologías (Árbol, CCC, Estrella, ...)
- Redes **indirectas** (dinámicas basadas en conmutador)

Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Clasificación

Clasificación de redes de interconexión (2/2)

- Redes **indirectas** (dinámicas basadas en conmutador)
 - Topologías regulares
 - Barras cruzadas (Crossbar)
 - Redes de interconexión multietapa (MIN)
 - Con bloqueos (unidireccionales y bidireccionales)
 - Sin bloqueos (red de Clos)
 - Topologías irregulares
- Redes híbridas (redes jerárquicas)

Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Clasificación

Otra clasificación:

- Redes **directas**
 - Nodos conectados a subconjuntos de nodos
 - Escalabilidad
 - Comunicación entre los nodos mediante *routers*.
 - Canales unidireccionales o bidireccionales
- Redes **indirectas**
 - Comunicación a través de conmutadores
 - Topologías regulares (matriciales) e irregulares (NOWs)
- Redes **híbridas** (combinación de las anteriores)
- Redes **multibus**
- Redes **jerárquicas** (jerarquía de buses conectados mediante routers)
- Redes basadas en **clusters**
 - Nodos conectados (buses – fácil *broadcasting*) formando *clusters*
 - Clusters conectados entre sí (red directa - escalabilidad)

Redes de medio compartido

- Medio de transmisión compartido
- Arbitraje (resolución de conflictos)
- Sencillo Broadcast
- Ancho de banda limitado (escalabilidad limitada) -> cuello de botella
- Bus de sistema (arquitectura UMA: Proc -> Mem)
- Redes de área local
 - ➔ Ethernet (no determinista)
 - ➔ Token bus (determinista → aplicaciones de tiempo real)
 - ➔ Token ring (estructura en anillo)

Ingeniería de los Computadores

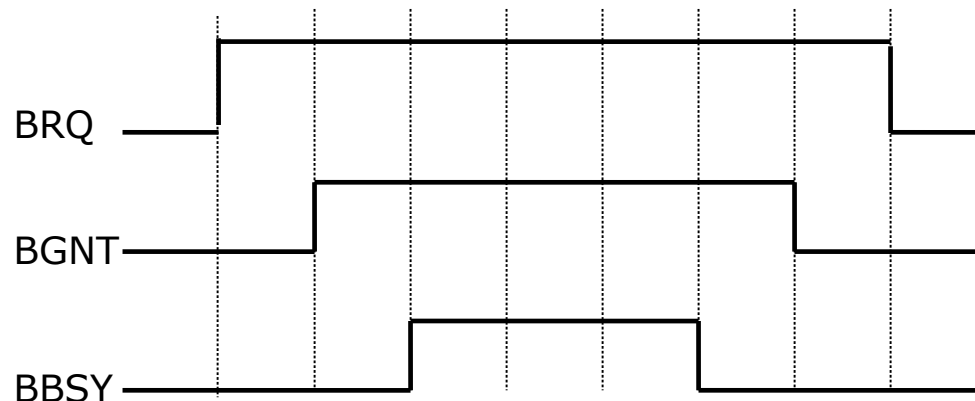
4.1 Introducción, conceptos y clasificación

Clasificación: redes de medio compartido

Redes de medio compartido (arbitraje del bus)

Prioridad estática. Señales de control:

- BRQ
- BGNT
- BBSY común



Ingeniería de los Computadores

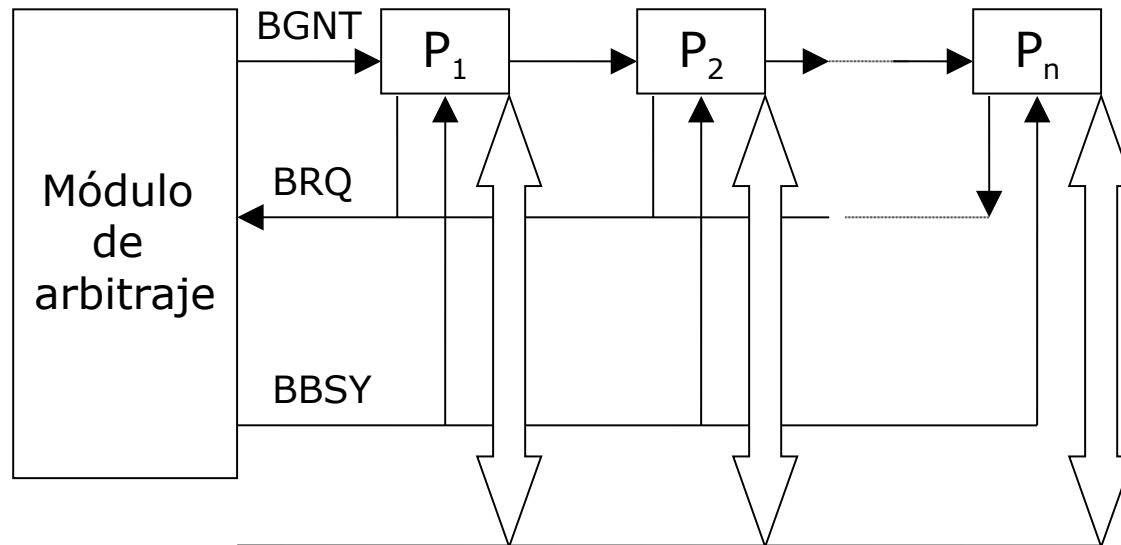
4.1 Introducción, conceptos y clasificación

Clasificación: redes de medio compartido

Redes de medio compartido (arbitraje del bus)

Prioridad estática. *Daisy Chain (centralizada-serie)*:

- BRQ común
- BGNT propagada
- BBSY común



Ingeniería de los Computadores

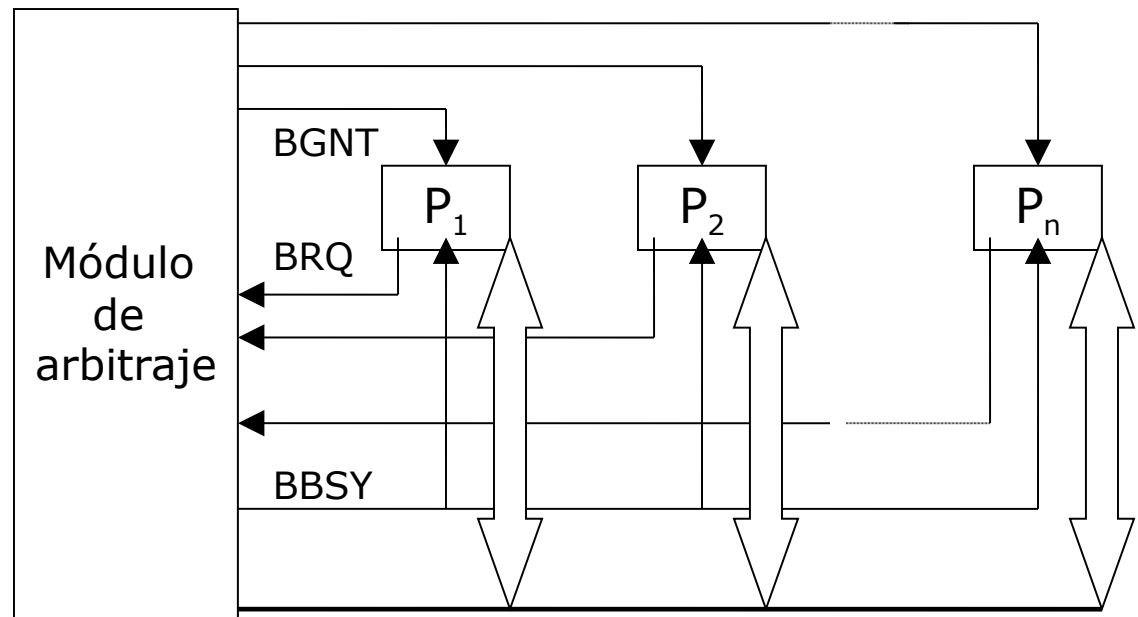
4.1 Introducción, conceptos y clasificación

Clasificación: redes de medio compartido

Redes de medio compartido (arbitraje del bus)

Prioridad estática. Codificador-decodificador de prioridad
(centralizada-paralela)

- BRQ individual
- BGNT individual
- BBSY común



Ingeniería de los Computadores

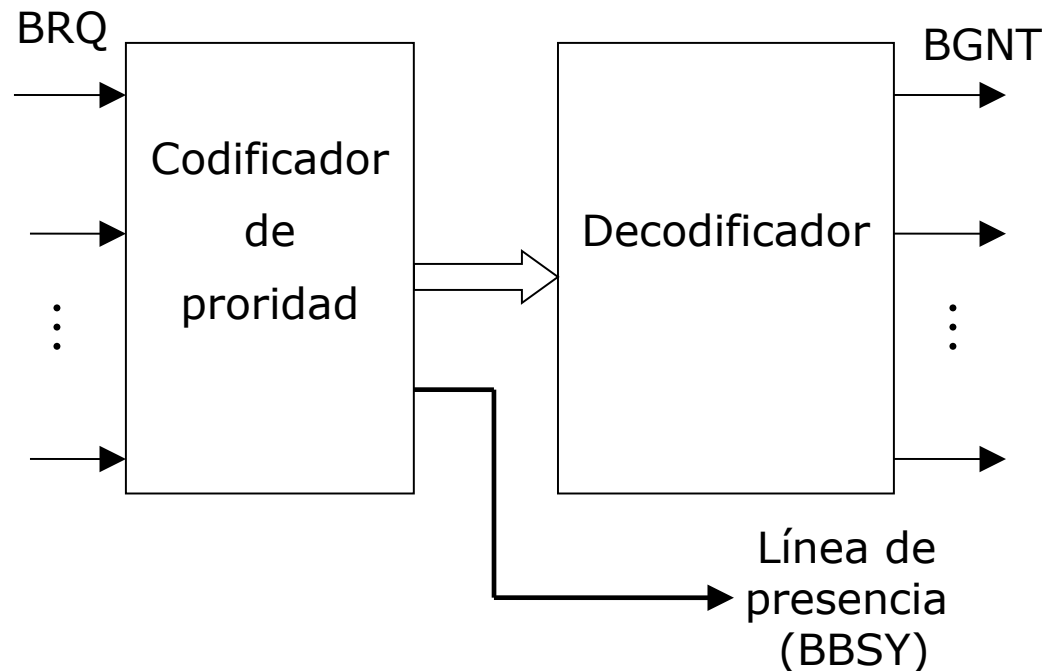
4.1 Introducción, conceptos y clasificación

Clasificación: redes de medio compartido

Redes de medio compartido (arbitraje del bus)

Prioridad estática. Codificador-decodificador de prioridad
(centralizada-paralela)

- BRQ individual
- BGNT individual
- BBSY común



Ingeniería de los Computadores

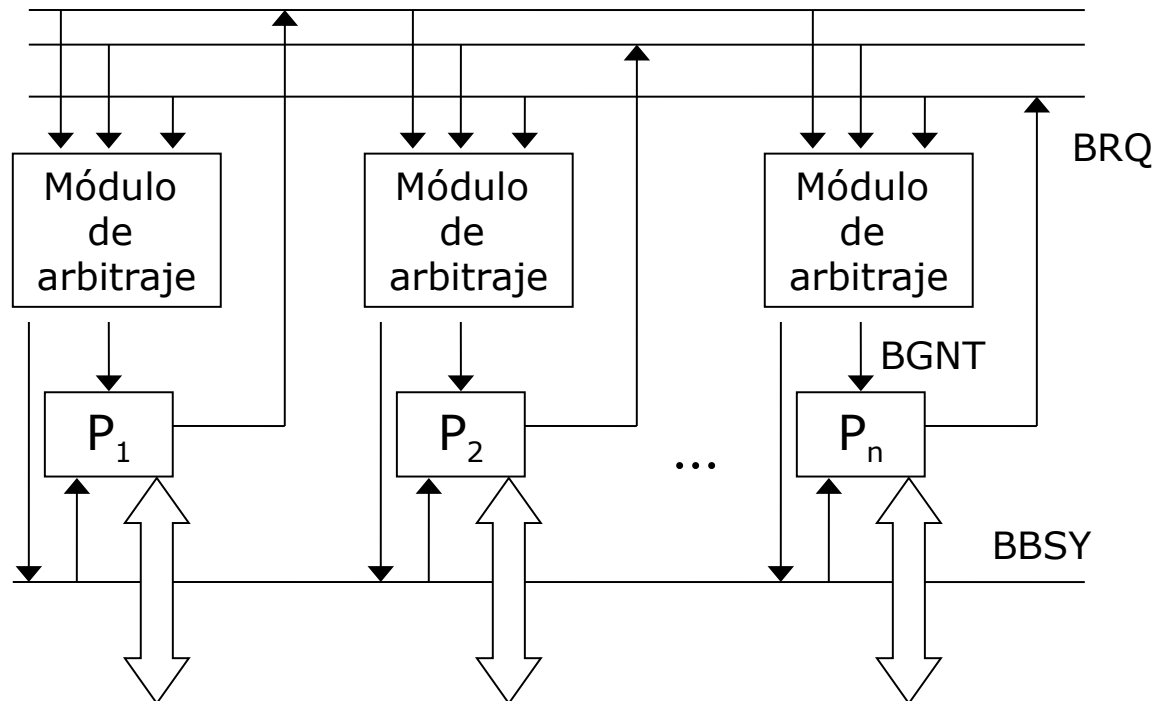
4.1 Introducción, conceptos y clasificación

Clasificación: redes de medio compartido

Redes de medio compartido (arbitraje del bus)

Prioridad estática. Autoarbitraje (distribuido-paralelo)

- BRQ individual
- BGNT individual
- BBSY común



4.1 Introducción, conceptos y clasificación

Clasificación: redes de medio compartido

Redes de medio compartido (arbitraje del bus)

- Multiplexación temporal

Ventajas:

- Asignación equitativa
- Simplicidad

Inconvenientes:

- Infrautilización del ancho de banda

- Prioridad dinámica

- LRU
- RDC (Rotating Daisy Chain)
- FCFS

Ingeniería de los Computadores

4.1 Introducción, conceptos y clasificación

Clasificación: redes de medio compartido

Redes de medio compartido (arbitraje del bus)

Ejemplo: Prioridad dinámica (LRU)

P_0	P_1	P_2	P_3	Acción
0	1	2	3	P_0 utiliza bus
0	1	2	3	P_2 solicita bus
1	2	0	3	P_2 utiliza bus
1	2	0	3	P_1 y P_3 solicitan bus
2	3	1	0	P_3 utiliza bus

¿Qué afecta al rendimiento?

Parámetros básicos

- Anchura de la bisección (B): mínimo número de canales que, al cortar, separa la red en dos partes iguales

Nota: El número de cables que cruzan la bisección es una cota inferior de la densidad de cableado

- Longitud del cable: efectos sobre la latencia
- Simetría: Una red es simétrica si es isomorfa a ella misma independientemente del nodo considerado origen
- Rendimiento
 - Funcionalidad: Indica cómo la red soporta el encaminamiento de datos, tratamiento de las interrupciones y sincronización.
 - Latencia: Indica el retraso de un mensaje

¿Qué afecta al rendimiento?

Parámetros básicos:

- Rendimiento
 - Ancho de banda. Velocidad máxima de transmisión de datos
 - Complejidad hardware. Coste de implementación (cables, conmutadores, conectores, etc.)
 - Escalabilidad. Capacidad de la red para expandirse de forma modular
 - Capacidad de transmisión. Número total de datos que se pueden transmitir a través de la red en una unidad de tiempo. (Punto caliente)

¿Qué afecta al rendimiento?

Diseño de una red de interconexión

- Topología -> grafo de interconexión
- Control de flujo -> método usado para regular el tráfico en la red
 - Mensaje
 - Paquete
 - Flit (FLoW control unIT)
- Encaminamiento -> método usado por un mensaje para elegir un camino entre los canales de la red
 - Determinista
 - Adaptativo

Topologías: Redes estáticas o directas

Redes estáticas o directas

Clasificación:

- Estrictamente ortogonales (malla, hipercubo, toro)
 - **(Estrictamente)** Cada nodo tiene al menos un enlace en cada dimensión.
 - **(Ortogonal)** Cada enlace supone un desplazamiento en **una dimensión**.
- No ortogonales (árbol)
- Propiedades
 - Grado (número de enlaces con otros nodos)
 - Diámetro (máximo camino más corto entre dos nodos)
 - Regularidad (todos los nodos tienen el mismo grado)
 - Simetría (se ve semejante desde cualquier nodo)

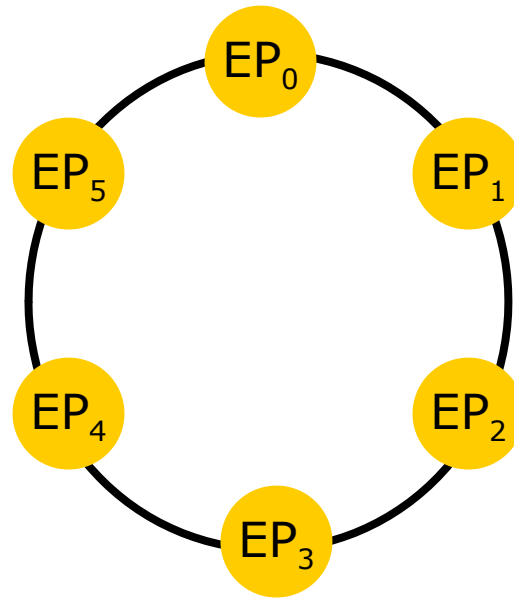
Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

Redes estáticas o directas. Anillo unidireccional

- Función de interconexión: $F_{+1}(i) = (i+1) \bmod N$
- Grado de entrada/salida: 1/1
- Diámetro: $N-1$



- ¿Anillo bidireccional?

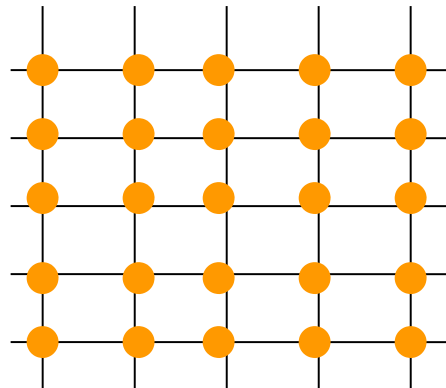
Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

Redes estáticas o directas. Ej. Malla abierta

- F. interconexión:
 $F_{+1}(i) = (i+1)$ si $i \bmod r \neq r-1$
 $F_{-1}(i) = (i-1)$ si $i \bmod r \neq 0$
 $F_{+r}(i) = (i+r)$ si $i \div r \neq r-1$
 $F_{-r}(i) = (i-r)$ si $i \div r \neq 0$
- Grado: 4
- Diámetro: $2(r-1)$, donde $N=r^2$



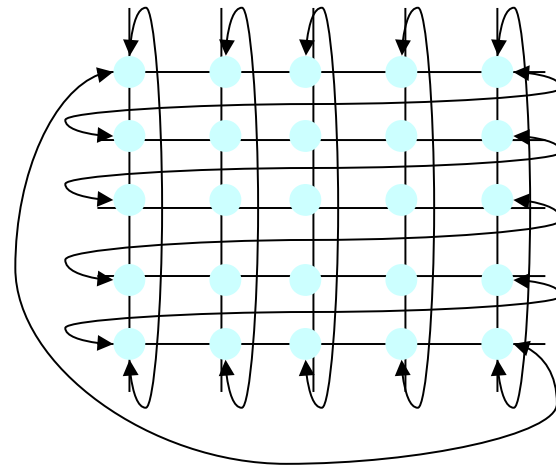
Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

Redes estáticas o directas. Ej. Malla Illiac

- F. interconexión:
 $F_{+1}(i) = (i+1) \bmod N$
 $F_{-1}(i) = (i-1) \bmod N$
 $F_{+r}(i) = (i+r) \bmod N$
 $F_{-r}(i) = (i-r) \bmod N$
- Grado: 4
- Diámetro: $(r-1)$, donde $N=r^2$



Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

Redes estáticas o directas. Ej. Redes n-cubos k-arias ó toros

- n dimensiones, k nodos
- Función de interconexión toro 2D:

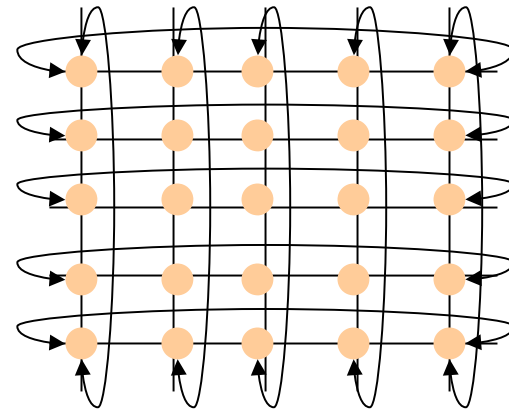
$$F_{+1}(i) = (i+1) \bmod r + (i \text{ DIV } r) \cdot r$$

$$F_{-1}(i) = (i-1) \bmod r + (i \text{ DIV } r) \cdot r$$

$$F_{+r}(i) = (i+r) \bmod N$$

$$F_{-r}(i) = (i-r) \bmod N$$

- Grado: 4
- Diámetro: $2 \cdot \left\lfloor \frac{r}{2} \right\rfloor$, donde $N=r^2$



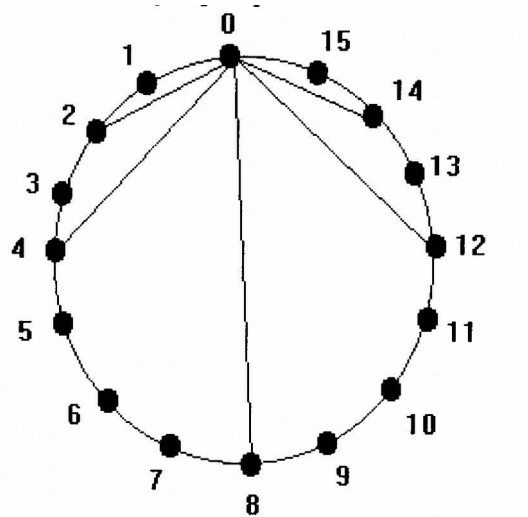
Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

Redes estáticas o directas. Ej. Desplazador barril

- F. interconexión:
 $B_{+k}(i) = (i + 2^k) \bmod N$
 $B_{-k}(i) = (i - 2^k) \bmod N$
 $K = 0 \dots n-1, n = \log N, i = 0 \dots N-1$
- Grado: $2n - 1$
- Diámetro: $n/2$

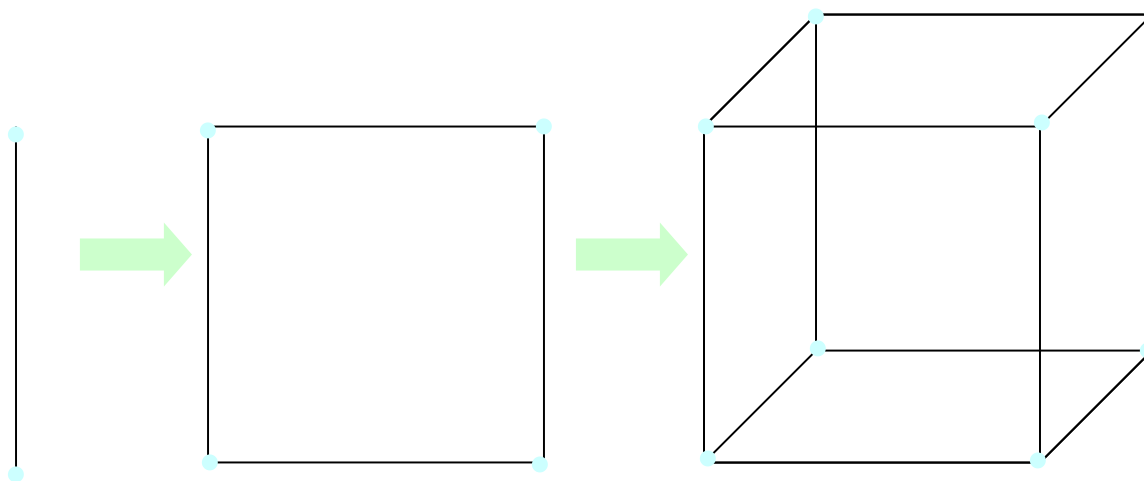


Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

- Redes estáticas o directas. Hipercubo
 - F. interconexión:
 - $F_i(h_{n-1}, \dots, h_i, \dots, h_0) = h_{n-1}, \dots, \bar{h}_i, \dots, h_0$
 - Grado: n ($n = \log N$)
 - Diámetro: n

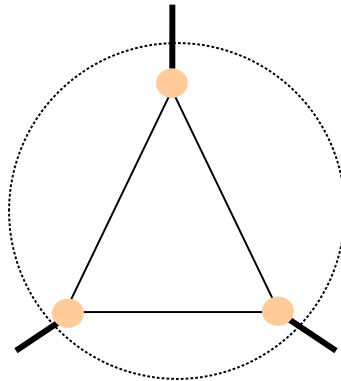


Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

Redes estáticas o directas. Ciclo cubo conectado (CCC) (red jerárquica)

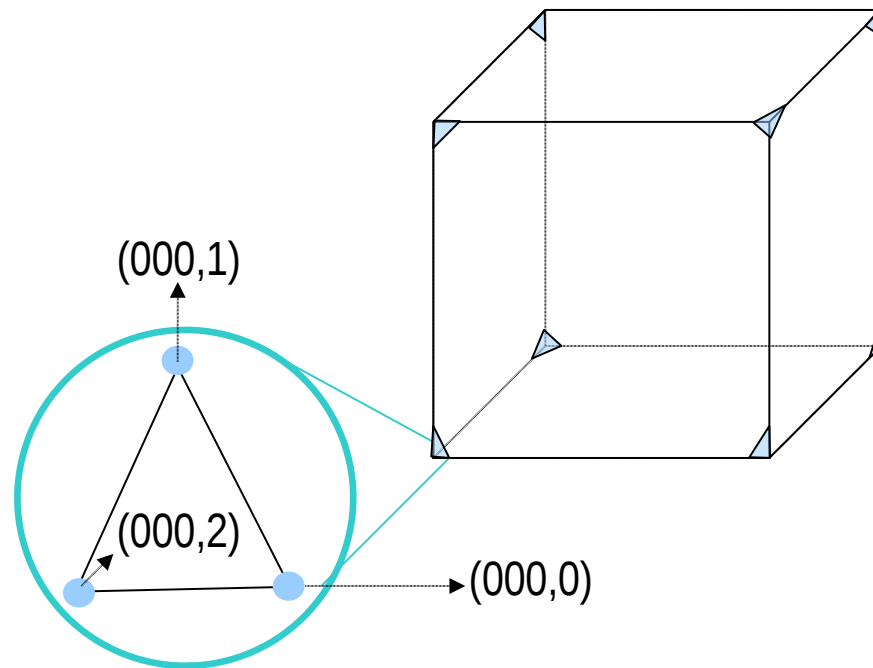


Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

Redes estáticas o directas. Ej. Red CCC (Ciclo-Cubo-Conectada)



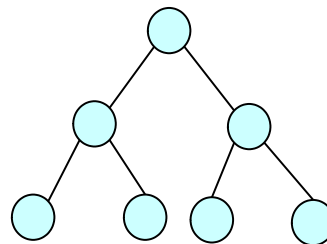
Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes estáticas o directas

Redes estáticas o directas. Ej. Árbol binario

- Balanceado: todas las ramas del árbol tienen la misma longitud
- Cuello de botella → nodo raíz
- $N \text{ (balanceado)} = 2^k - 1$ (k = niveles del árbol)
- Grado: 3
- Diámetro: $2(k-1)$



Redes indirectas o dinámicas

- Uso de conmutadores y árbitros
- Ejemplos
 - Redes crossbar
 - Redes de conexión multietapa (MIN)
- Modelo: $G(N,C)$
 - N , conjunto de conmutadores
 - C , enlaces (unidireccionales o bidireccionales) entre conmutadores
 - Canal bidireccional \rightarrow dos canales unidireccionales
 - Un conmutador puede tener conectados 0, 1 o más elementos (Procesadores, memorias, etc.)
- Distancia entre dos nodos: distancia entre los conmutadores que conectan los nodos más 2.

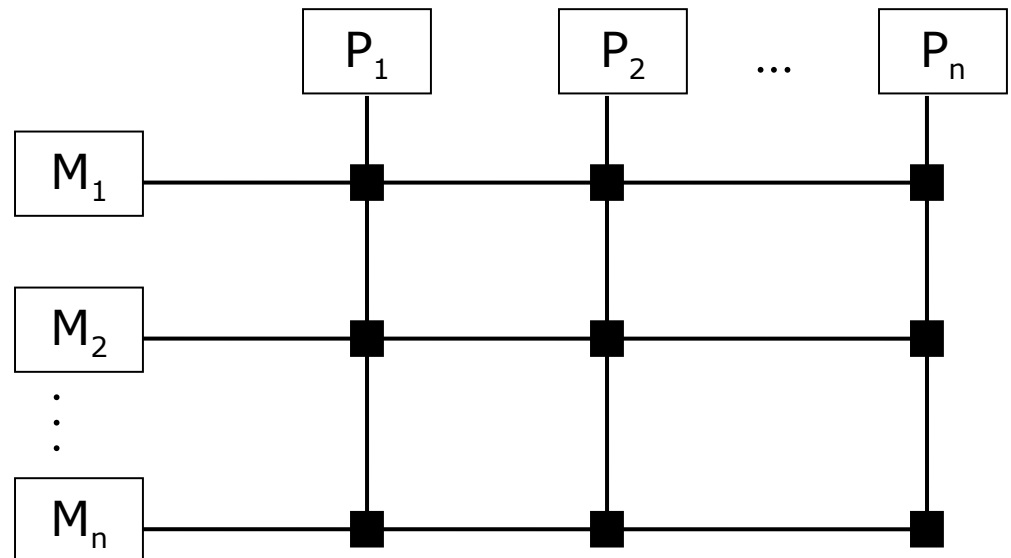
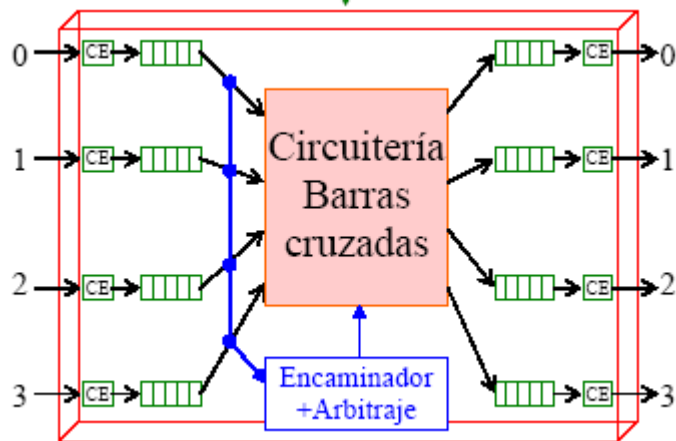
Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

Redes indirectas o dinámicas. Ej. Redes *crossbar*

- Conexión directa nodo-nodo
- Gran ancho de banda y capacidad de interconexión
- Conexión Proc. - Mem. → limitado por los accesos a memoria (columnas)
- Conexión Proc(N) - Proc(N) → máximo de N conexiones
- Coste elevado: $O(N \cdot M)$



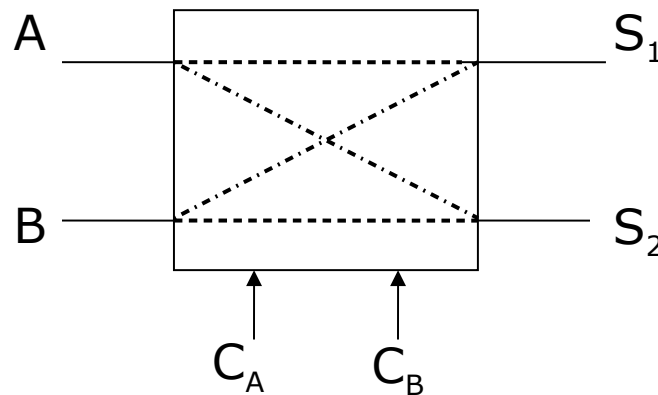
Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

Redes indirectas o dinámicas. Ej. Redes MIN

- Conectan dispositivos de entrada con dispositivos de salida mediante un conjunto de etapas de conmutadores, donde cada conmutador es una red *crossbar*.
- Concentradores \rightarrow n° entradas $>$ n° salidas
- Expansores \rightarrow n° salidas $>$ n° entradas



Ingeniería de los Computadores

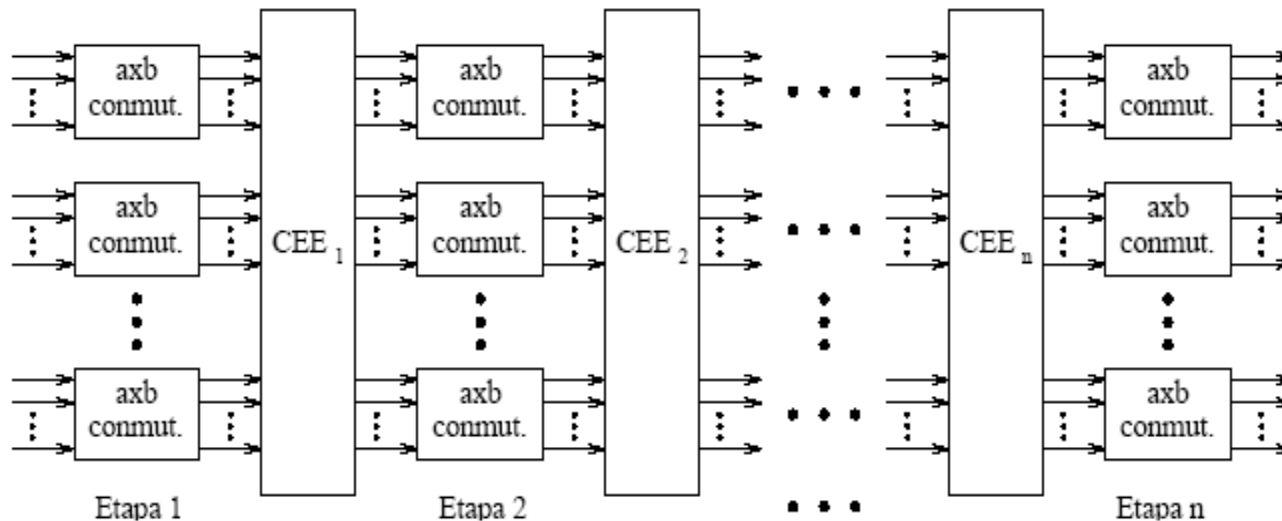
Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

Redes indirectas o dinámicas. Redes MIN

- Conexión de etapas adyacentes → Patrón de conexión
- Patrón basado en permutaciones: conmutadores con el mismo número de entradas y salidas.
- Ejemplo: barajado perfecto.

$$B(a_{n-1}, a_{n-2}, \dots, a_0) = (a_{n-2}, \dots, a_0, a_{n-1})$$



Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

Redes indirectas o dinámicas. Redes MIN

- Número de entradas a^n y número de salidas b^n (red $a^n \times b^n$)
 - ✓ n etapas de conmutadores (C_0, C_1, \dots, C_{n-1})
 - ✓ Conmutadores $a \times b$
 - ✓ $a^{n-1-i} \times b^i$ conmutadores en la etapa C_i
- Funcionalidad de los conmutadores: barras cruzadas, reducción, difusión
- Subred de interconexión entre etapas: R_0, R_1, \dots
- Tipos de canales: unidireccionales, bidireccionales

Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

Redes indirectas o dinámicas. Redes MIN – red Omega

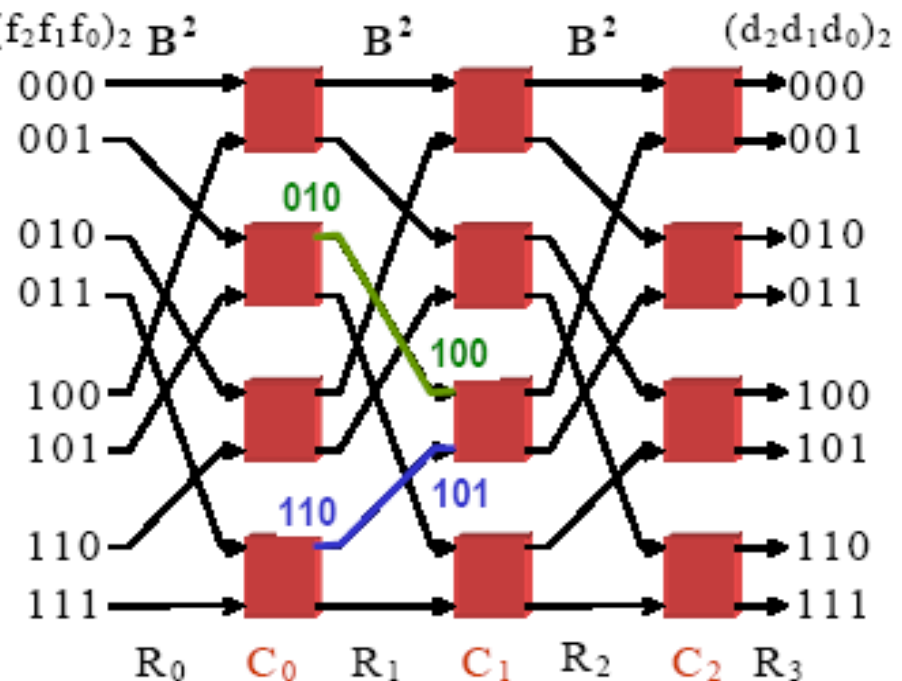
El patrón de conexión C_i es una permutación k -baraje perfecto a excepción del último (R_n) que es permutación 0

Red $k^n \times k^n$ ($8 \times 8 = 2^3 \times 2^3$):

- n etapas C_i (**3**),
- conmutadores $k \times k$ (**2×2**),
- k^{n-1} conm/etapa (**2^2**).

Subred R_i ($i=0, \dots, n-1$):

- Baraje- k perfecto
(**baraje-2 perfecto**).



$$B^k \left((f_{n-1}, f_{n-2}, \dots, f_1, f_0)_k \right) = (f_{n-2}, \dots, f_1, f_0, f_{n-1})_k$$

$$B^2 \left((f_2, f_1, f_0)_2 \right) = (f_1, f_0, f_2)_2$$

Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

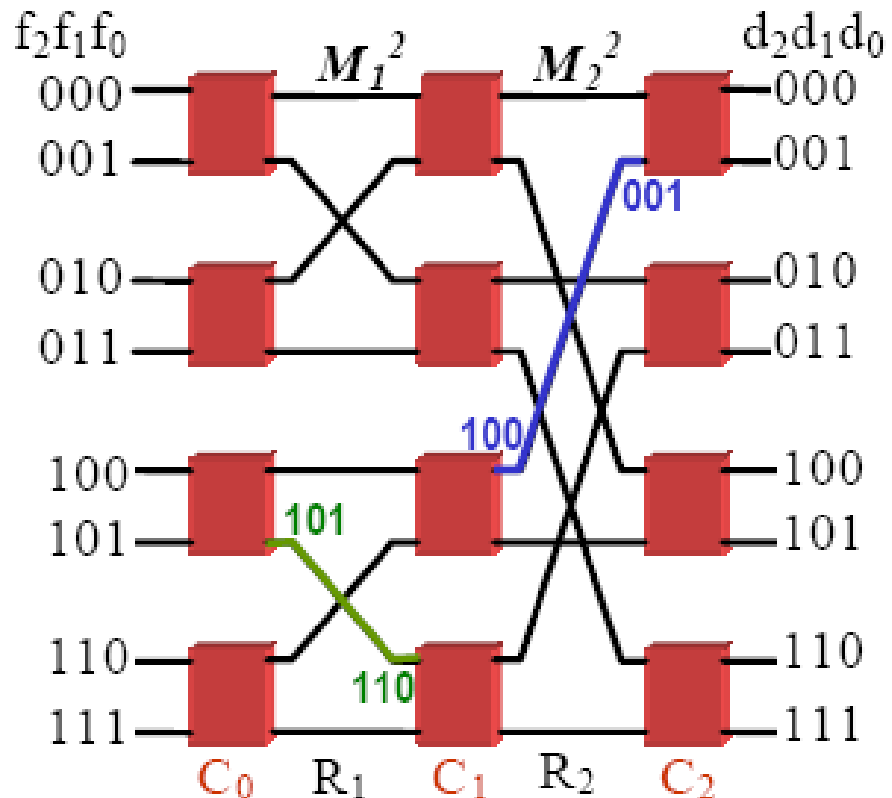
Redes indirectas o dinámicas. Ej. Redes MIN – red mariposa

– Red $k^n \times k^n$ ($8 \times 8 = 2^3 \times 2^3$):

- n etapas C_i (**3**),
- conmutadores $k \times k$ (**2x2**),
- k^{n-1} conm/etapa (**2²**).

– Subred R_i ($i=0, \dots, n-1$):

- Mariposa M_i^k



$$M_i^k ((f_{n-1}, f_{n-2}, \dots, f_{i+1}, \mathbf{f_i}, f_{i-1}, \dots, f_1, \mathbf{f_0})_k) = (f_{n-1}, f_{n-2}, \dots, f_{i+1}, \mathbf{f_0}, f_{i-1}, \dots, f_1, \mathbf{f_i})_k$$

$i=0, \dots, n-1$

$$M_2^2 ((\mathbf{f_2}, \mathbf{f_1}, \mathbf{f_0})_2) = (\mathbf{f_0}, \mathbf{f_1}, \mathbf{f_2})_2$$

Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

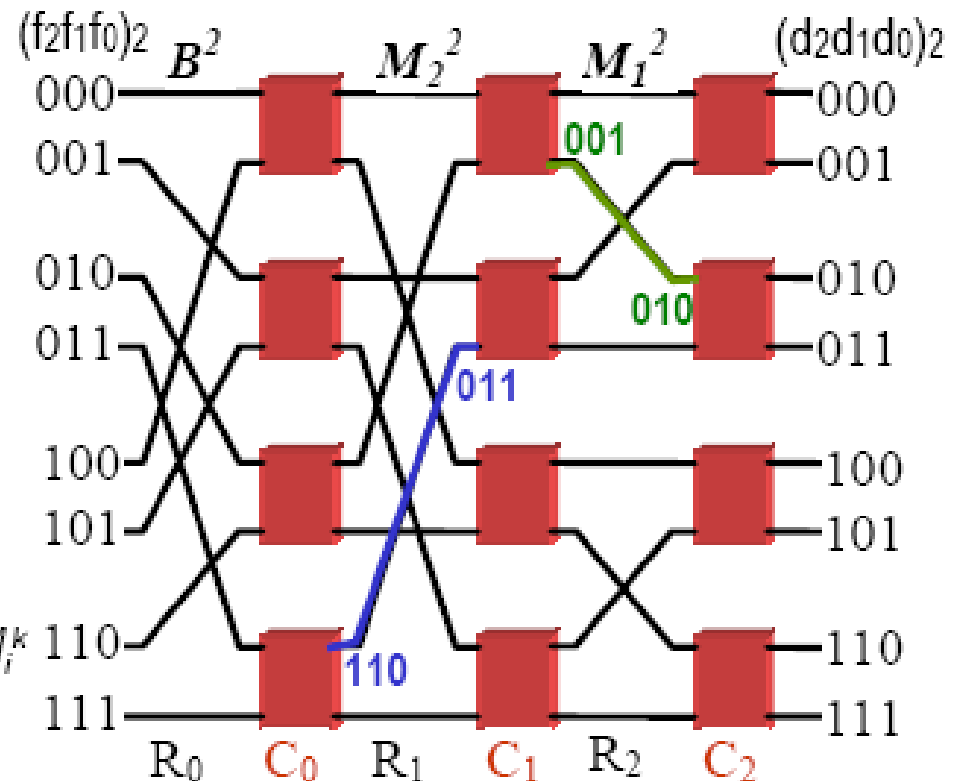
Redes indirectas o dinámicas. Ej. Redes MIN – red cubo

– Red $k^n \times k^n$ ($8 \times 8 = 2^3 \times 2^3$):

- n etapas C_i (**3**),
- conmutadores $k \times k$ (**2×2**),
- k^{n-1} conm/etapa (**2^2**).

– Subred R_i ($i=0, \dots, n-1$):

- R_0 : Baraje- k perfecto (**baraje-2 perfecto**).
- R_{n-i} ($i=1, \dots, n-1$): Mariposa M_i^k



$$M_i^k ((f_{n-1}, f_{n-2}, \dots, f_{i+1}, \mathbf{f_i}, f_{i-1}, \dots, f_1, \mathbf{f_0})_k) = (f_{n-1}, f_{n-2}, \dots, f_{i+1}, \mathbf{f_0}, f_{i-1}, \dots, f_1, \mathbf{f_i})_k$$

$i=0, \dots, n-1$

$$M_1^2 ((f_2, \mathbf{f_1}, \mathbf{f_0})_2) = (f_2, \mathbf{f_0}, \mathbf{f_1})_2$$

Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

Redes indirectas o dinámicas. Ej. Redes MIN – red delta

Red $a^n \times b^n$ (**$16 \times 9 = 4^2 \times 3^2$**):

- n etapas C_i (**2**),
- conmutadores $a \times b$ (**4×3**),
- $a^{n-1-i} \cdot b^i$ conm / C_i (**4, 3**).

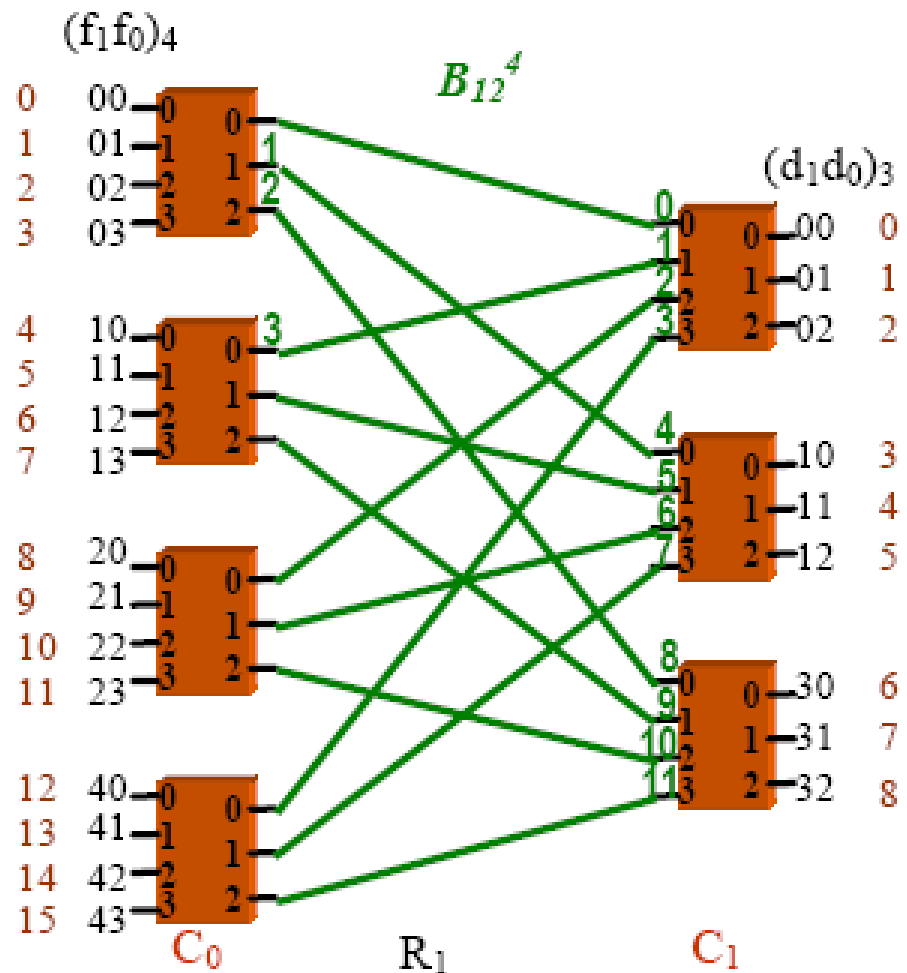
Subred R_i ($i=0$ o $1, \dots, n-1$):

- Baraje-a de c elementos

R_i : (**baraje-4 de 12 elementos**).

$$B_c^a(s) = \begin{cases} a \cdot s \bmod (c-1) & \text{si } s < c-1 \\ c-1 & \text{si } s = c-1 \end{cases}$$

$$B_{12}^4(s) = \begin{cases} 4 \cdot s \bmod (11) & \text{si } s < 11 \\ 11 & \text{si } s = 11 \end{cases}$$

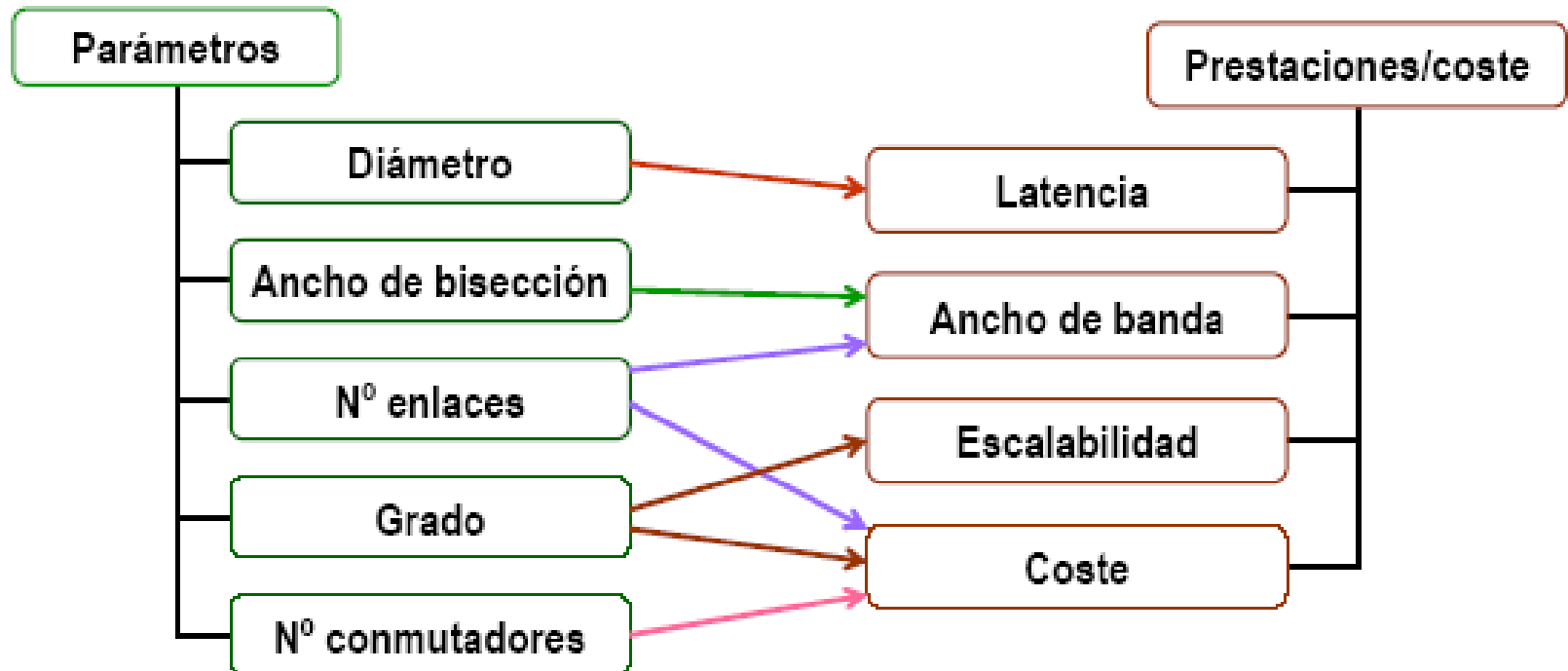


Ingeniería de los Computadores

Unidad 4.3 Topologías.

Topologías: Redes indirectas o dinámicas

Prestaciones:

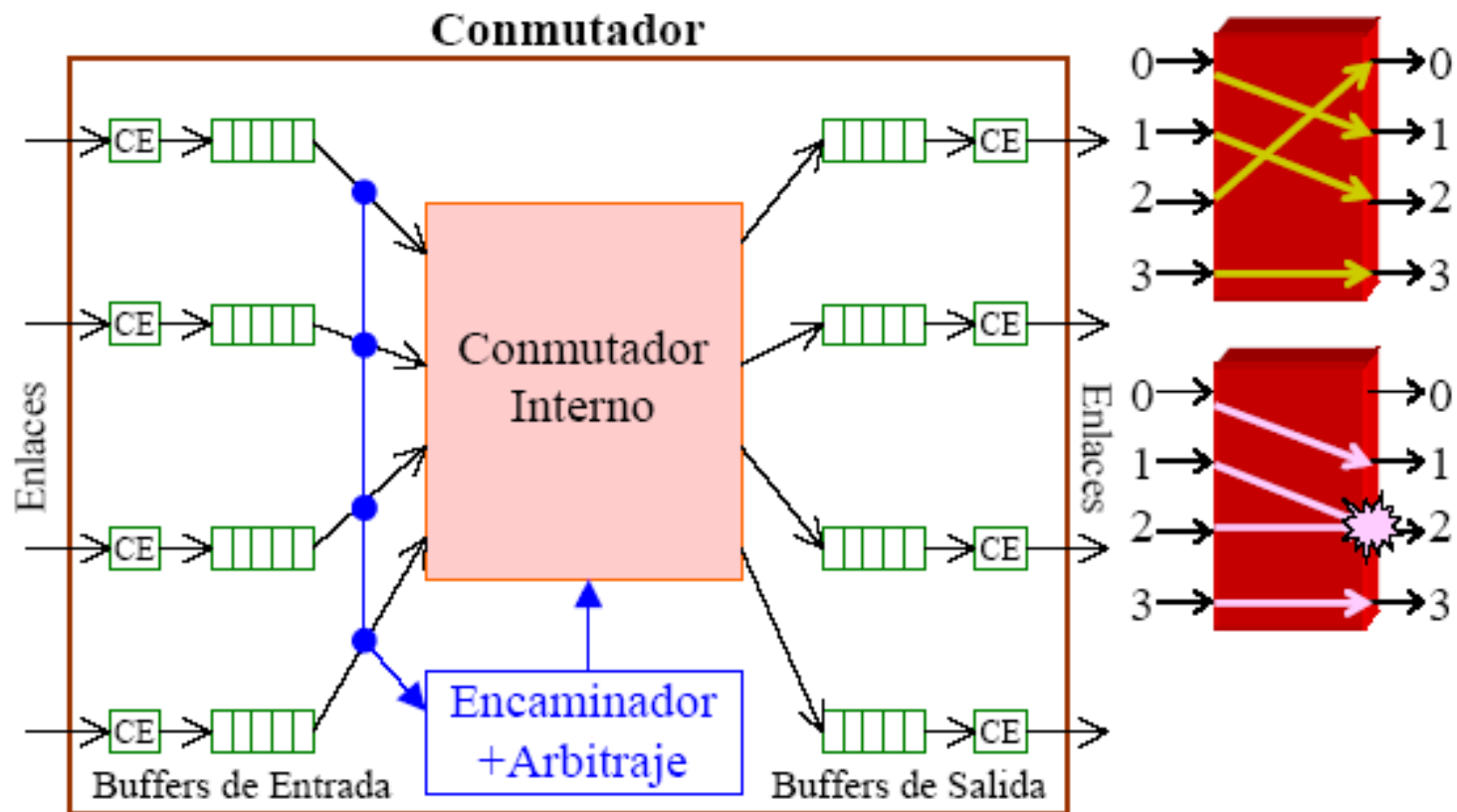


Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

El conmutador

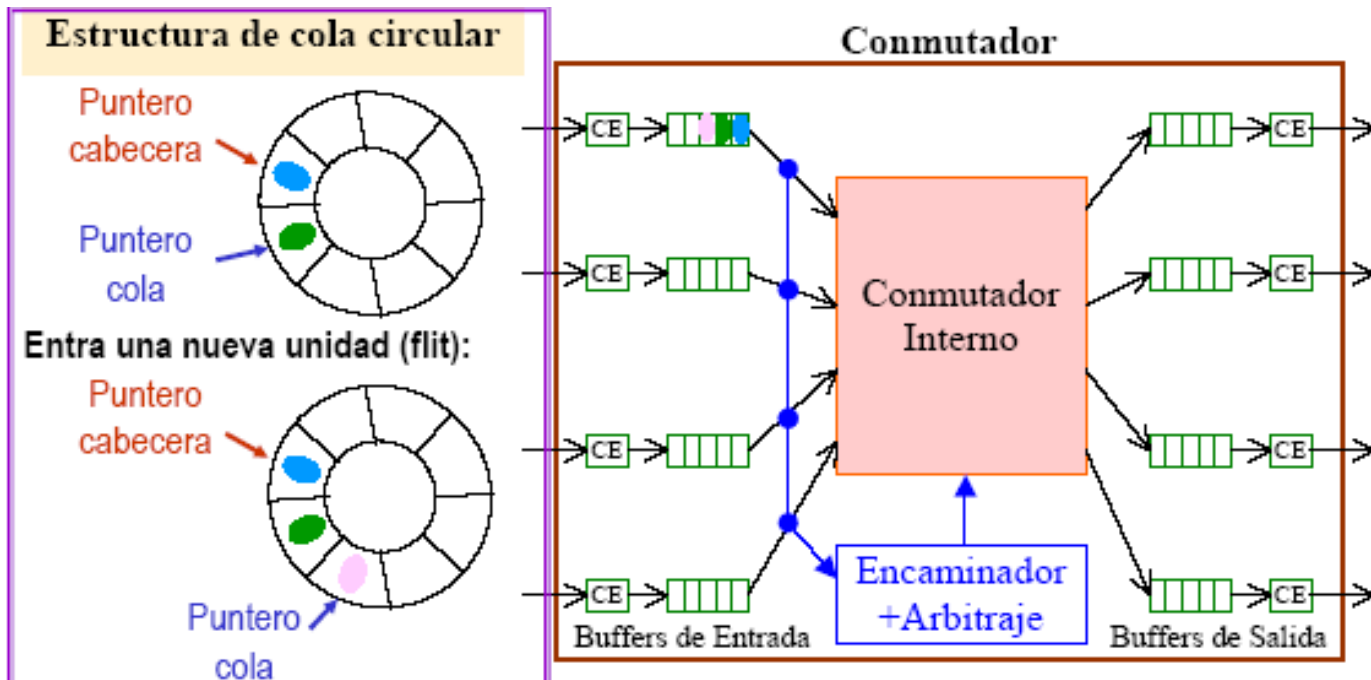


Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Buffers de entrada



Otra alternativa es una estructura de datos de lista enlazada

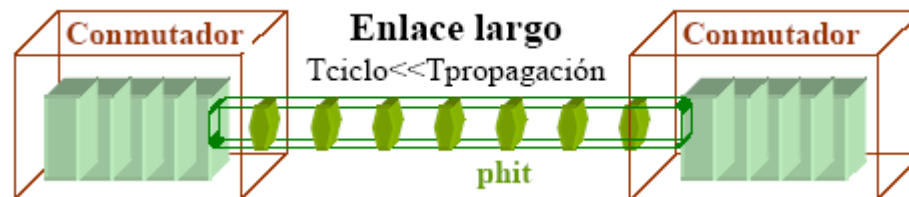
Coste de gestión mayor. Además de actualizar punteros de cabecera y cola hay que modificar punteros entre celdas y gestionar una lista con celdas libres.

Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

- Enlaces y canales.
 - ✓ Infraestructura: hilos eléctricos (cobre), fibras ópticas, etc.
- Anchura
 - ✓ Anchos: Si se transmite simultáneamente datos y control
 - ✓ Estrechos: Cuando se multiplexa en el tiempo datos y control
- Longitud
 - ✓ Cortos: 1 símbolo
 - ✓ Largos: Varios símbolos de forma simultánea



Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

- Enlaces: longitud
 - Cortos: El ciclo de red depende del retardo de propagación
 - Largos: Ciclo de red \ll retardo de propagación
- Velocidad del canal depende:
 - Energía empleada para transmitir por una línea
 - Distancia a atravesar
 - Ruido
 - Desplazamiento entre líneas de un enlace
 - Tamaño del *buffer* destino (enlaces largos)

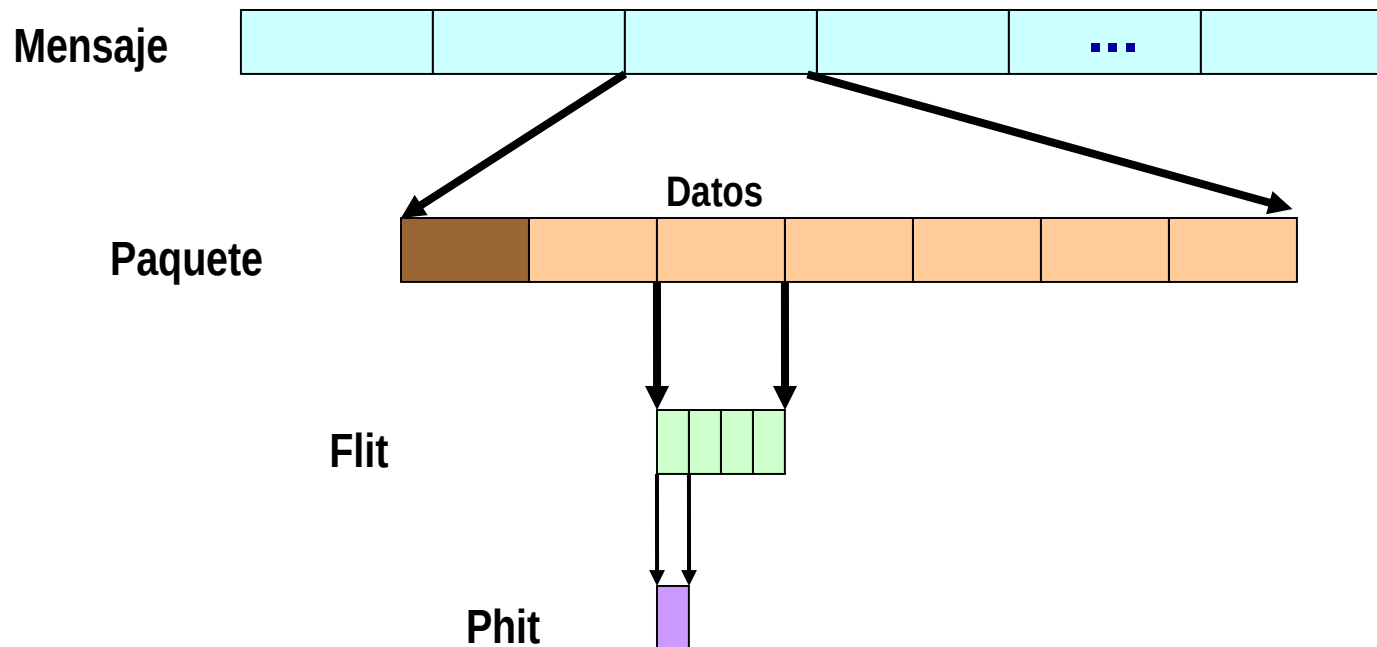
Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Técnicas de conmutación:

- Cuándo y cómo se conectan entradas y salidas de routers
- Cuándo se transfiere el mensaje por los caminos



Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

- Tipos de técnicas de conmutación
 - Almacenamiento y reenvío (S&F, Store and Forward)
 - Vermiforme (*Wormhole*)
 - *Virtual Cut-Through* (VCT)
 - Conmutación de circuitos (CC, *Circuit Switching*) (Origen en redes telefónicas)
 - Canales virtuales
- Comparación entre técnicas
 - Comparación cuantitativa: latencia de transporte
 - Comparación cualitativa: ancho de banda global

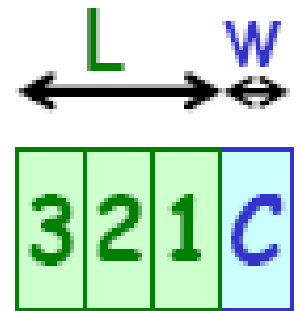
Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Se considera (a efectos de explicación teórica siguientes transparencias):

- 1 phit = 1 flit = W bits
- Cabecera = 1 flit
- Tamaño total del paquete = L bits + W bits (cabecera
- Distancia fuente-destino = D parejas conmutador-en
- Conmutadores con *buffer* independiente para cada entrada y salidas sin *buffer*
- T_w = tiempo para que un phit atravesase una etapa conmutador/enlace
- T_r = tiempo de encaminamiento (*routing*)



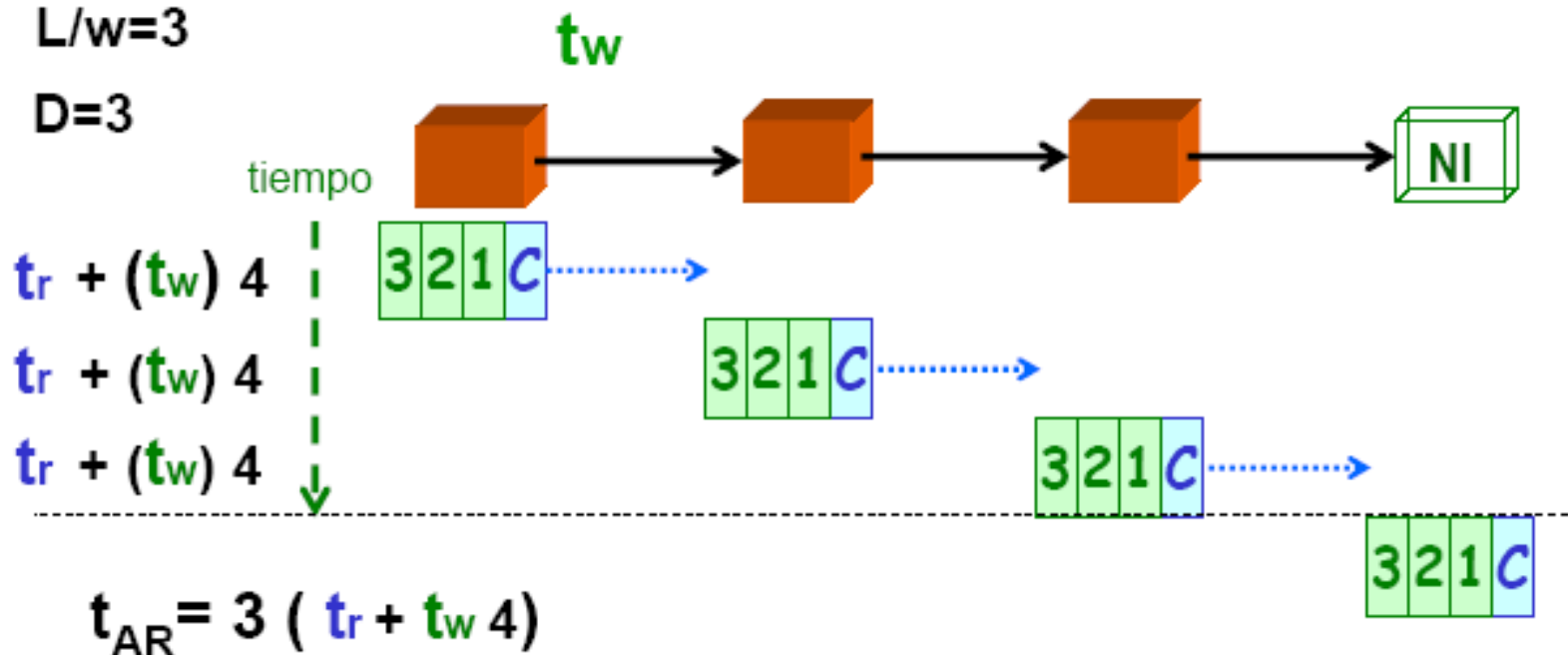
Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Store & forward (almacenamiento y reenvío)

- El conmutador almacena el paquete completo antes de ejecutar el algoritmo de encaminamiento y reenviar
- La unidad de transferencia (**paquete**) entre interfaces ocupa sólo un canal en cada instante
- Almacenamiento en conmutadores: múltiplos de un paquete (mínimo 1 paquete)
- Ancho de banda: El número de enlaces ociosos influye en este parámetro: para un tamaño de *buffer* mínimo (1 paquete), un paquete bloqueado deja ocioso un canal



Wormhole

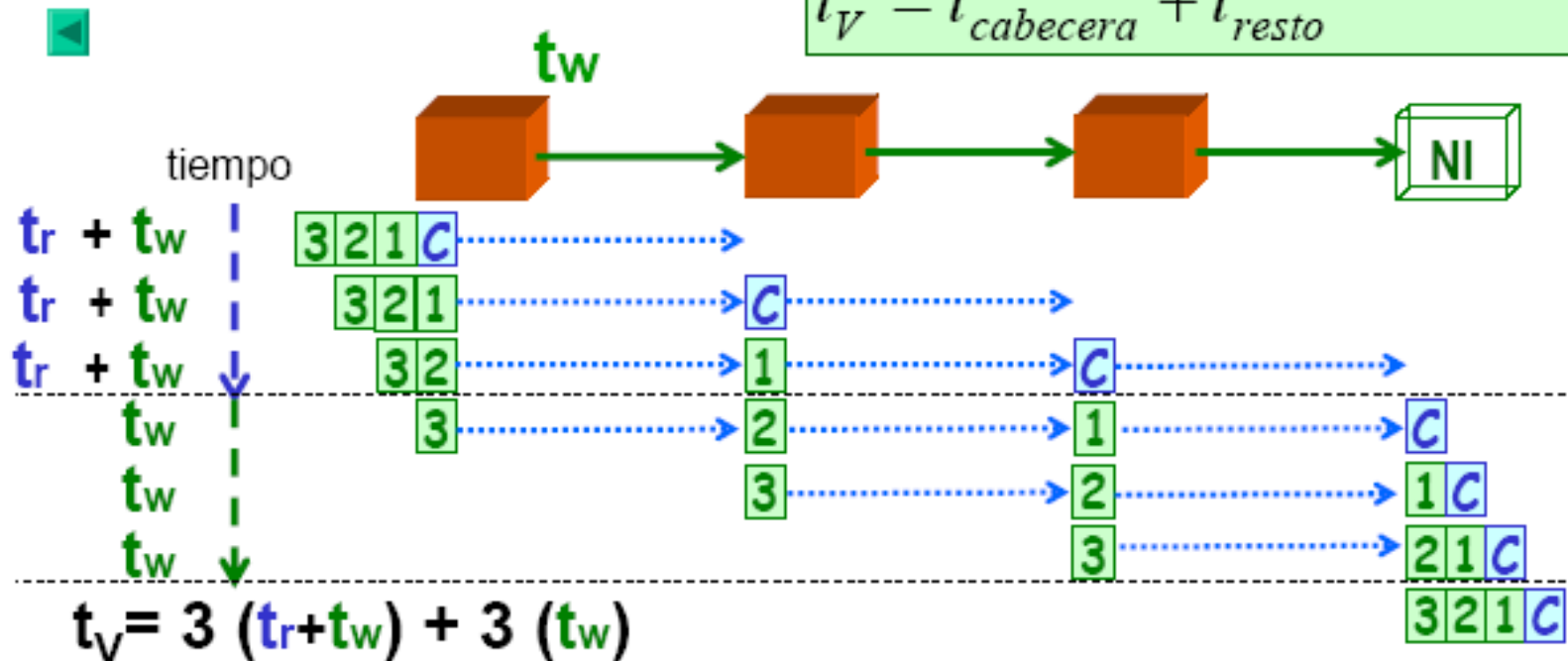
- En cuanto llega la cabecera al conmutador se ejecuta el algoritmo de encaminamiento y se reenvía
- La unidad de transferencia es el **mensaje**
- La transferencia se hace a través de un camino segmentado (n° etapas depende del n° de buffer). La unidad de transferencia puede ocupar varios canales en un instante
- Almacenamiento en conmutadores: múltiplos de un flit (mínimo 1 flit)
- Ancho de banda: El número de enlaces ociosos influye en este parámetro: para un tamaño de *buffer* mínimo (1 flit), un paquete bloqueado deja ociosos varios canales

Wormhole

- Latencia de transporte*
(buffer sólo en entradas de conmutadores)

$$t_V = D \cdot (t_r + t_w) + t_w \cdot \left\lceil \frac{L}{W} \right\rceil$$

$$t_V = t_{\text{cabecera}} + t_{\text{resto}}$$



Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

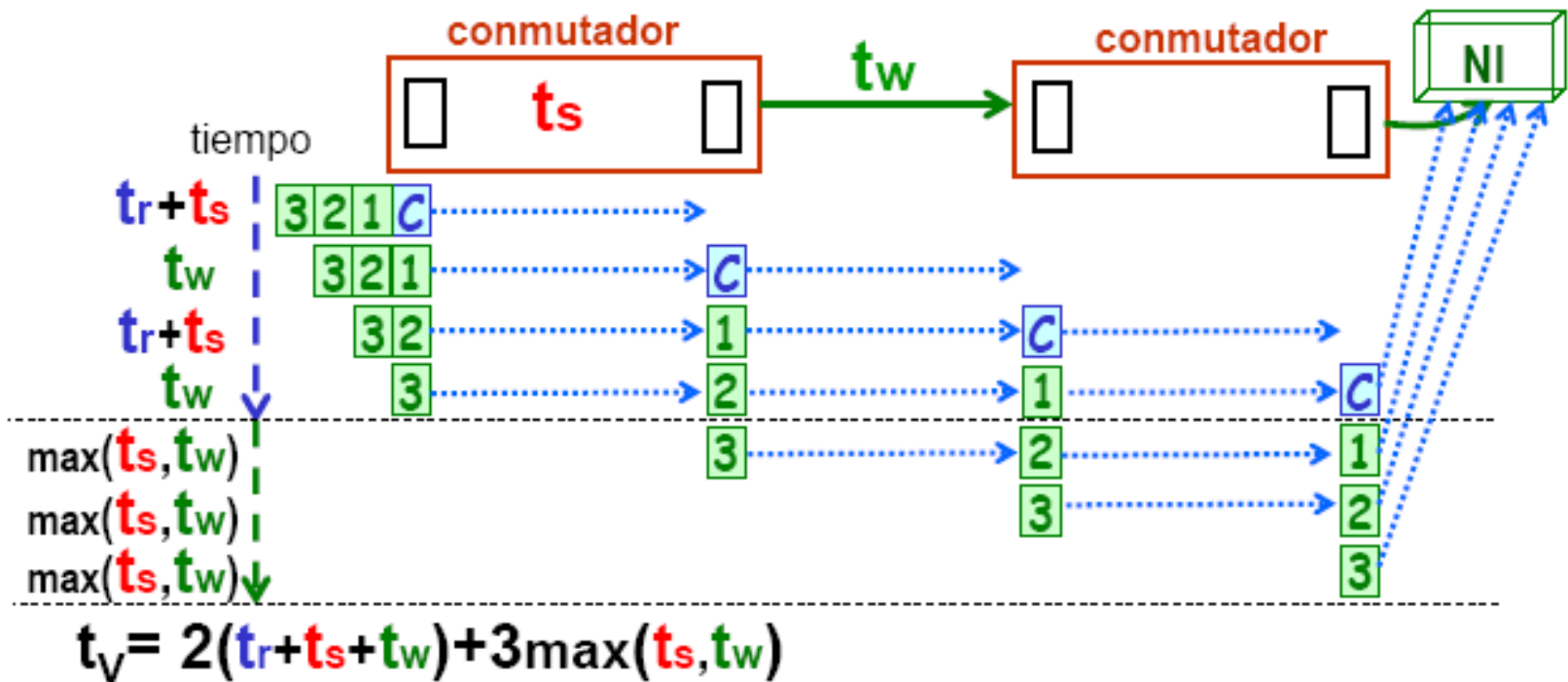
Técnicas de conmutación

Wormhole

- Latencia de transporte:
(buffer en entradas y salidas)

$$t_V = D \cdot (t_r + t_s + t_w) + \max(t_s, t_w) \cdot \left\lceil \frac{L}{W} \right\rceil$$

$$t_V = t_{cabecera} + t_{resto}$$



Virtual Cut-Through

- En cuanto llega la cabecera al conmutador se ejecuta el algoritmo de encaminamiento y se reenvía
- La unidad de transferencia es el **paquete**
- La transferencia se hace a través de un camino segmentado (n° etapas depende del n° de *buffer*). La unidad de transferencia puede ocupar varios canales en un instante
- Almacenamiento en conmutadores: múltiplos de un paquete (mínimo 1 paquete)
- Prestaciones:
 - Latencia = **Wormhole**
 - Ancho de banda = **Store & Forward**

Conmutación de circuitos

- Desde el fuente se envía una sonda (flit) que reserva el camino. El destino devuelve una señal de reconocimiento y el fuente comienza la transmisión
- La unidad de transferencia es el **mensaje**
- La transferencia se hace a través del canal entre fuente y destino (o un camino segmentado) reservado por la sonda
- Almacenamiento en conmutadores: los *buffers* almacenan la sonda
- Ancho de banda : Cuando la sonda queda bloqueada deja ociosos múltiples canales (tantos como la distancia del punto de bloqueo al fuente)

Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

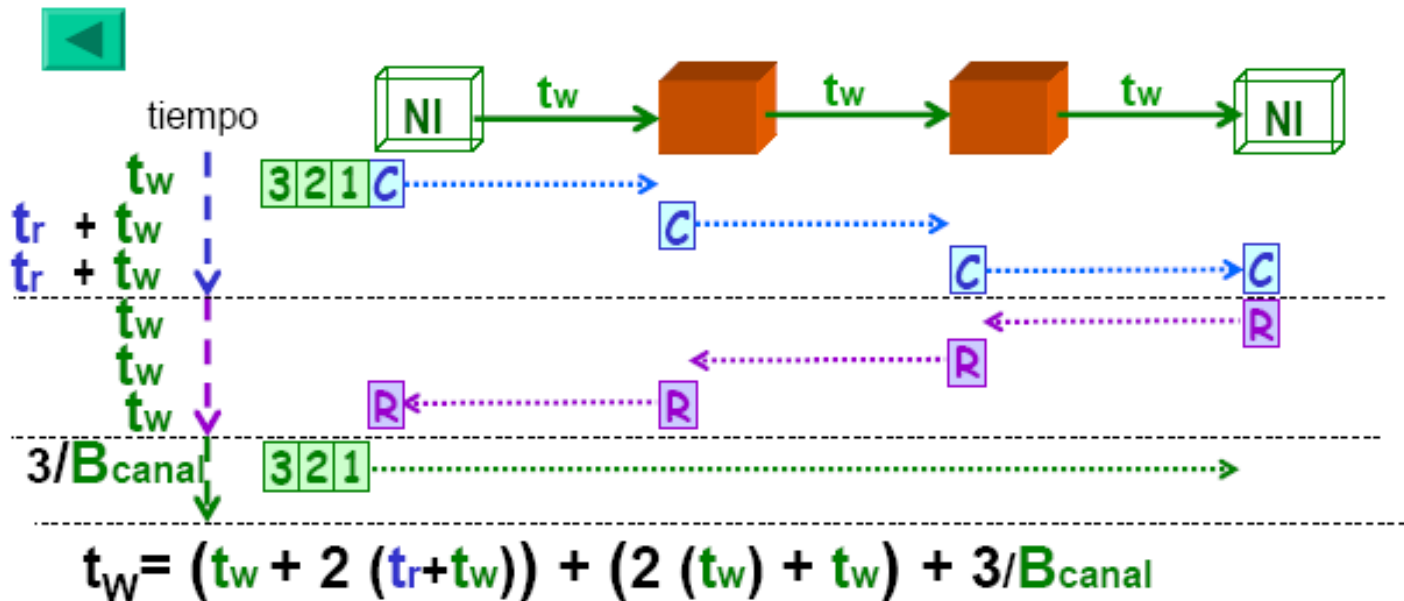
Técnicas de conmutación

Conmutación de circuitos

- *Latencia de transporte (si se establece 1 canal):*

$$t_{CC} = [t_w + D \cdot (t_r + t_w)] + [D \cdot (t_w) + t_w] + [1/B_{canal} \cdot \lceil L/W \rceil]$$

$$t_{CC} = t_{sonda} + t_{reconocimiento} + t_{datos}$$



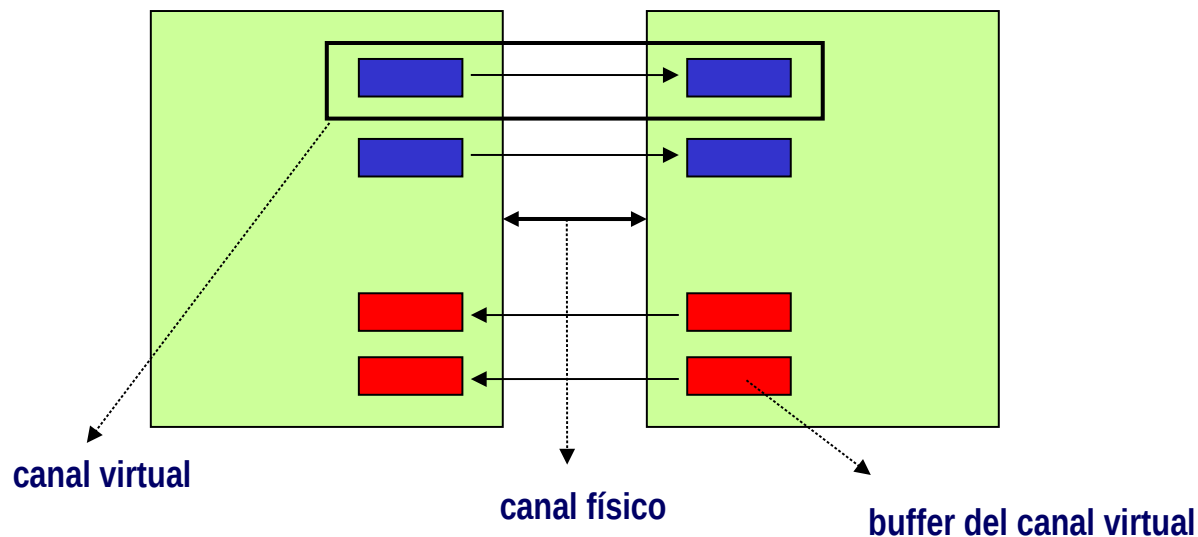
Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Canales virtuales

- Permiten que varios paquetes compartan el mismo enlace (a nivel de flit)
- Mejoran el ancho de banda y la latencia al disminuir la probabilidad de bloqueos
- Se aplica con el resto de mecanismos



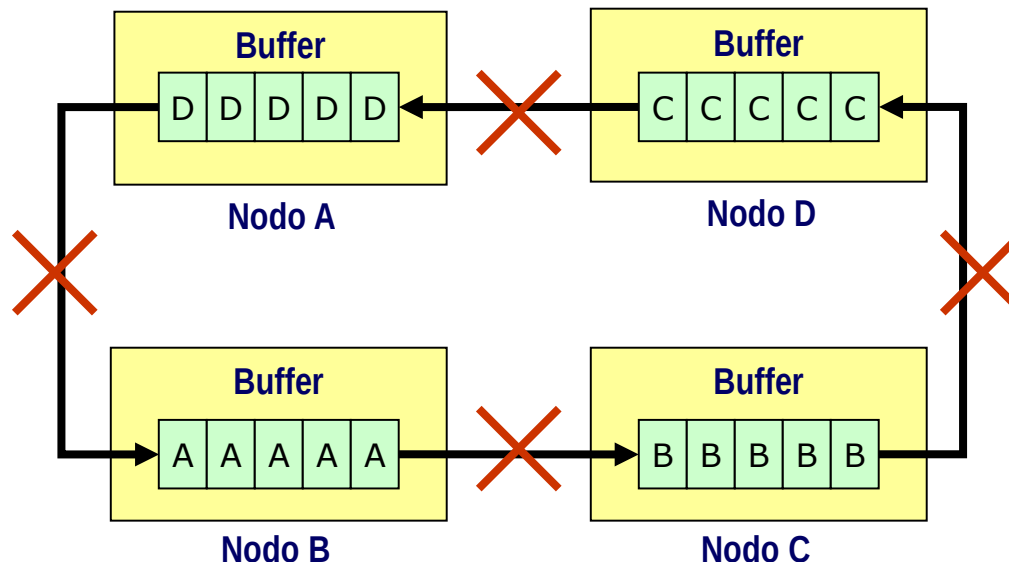
Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

¿Bloqueos?

- Algunos paquetes no pueden alcanzar el destino
- Capacidad de los *buffers* finita
- Canales ocupados



Bloqueos. Clasificación:

- **Interbloqueos** (*deadlocks*)
 - Recursos no disponibles para el avance de los paquetes
 - *Buffers* ocupados
 - Bloqueo permanente
- **Bloqueos activos** (*livelocks*)
 - Los paquetes nunca llegan a su destino
 - Canales ocupados por otros paquetes
 - Sólo ocurre si se permiten caminos no mínimos
- **Inanición** (los recursos siempre se asignan a otros paquetes)

Bloqueos. **Soluciones:**

- Inanición
 - Emplear un esquema de asignación de recursos correcto
 - Cola circular con distinta prioridad
- Bloqueos activos
 - Usar solo rutas mínimas
 - Usar rutas no mínimas restringidas
 - Dar mayor probabilidad a caminos mínimos respecto a no mínimos
- Interbloqueos
 - Prevención
 - Evitación
 - Recuperación

Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

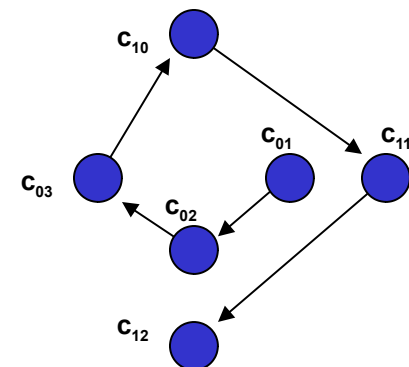
Técnicas de conmutación

Prevención de interbloqueos:

- Estrategia muy conservadora: se asignan todos los recursos para transmitir un mensaje antes de iniciar la transmisión
- Un flit de sondeo establece el camino
- Si existe bloqueo, retrocede y libera recursos

Evitación de interbloqueos

- Los recursos se asignan a medida que el mensaje atraviesa la red
- Un recurso se asigna a un paquete si el estado resultante es seguro (grafo de dependencias acíclico)



Ingeniería de los Computadores

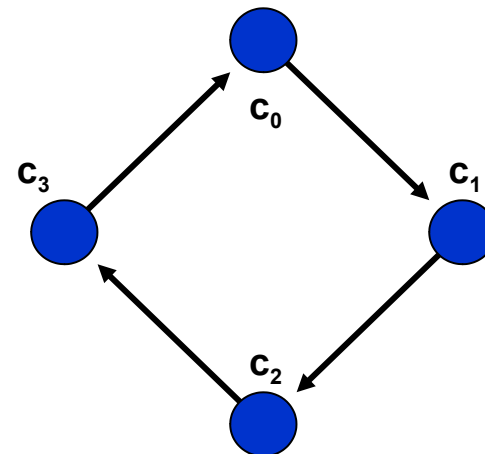
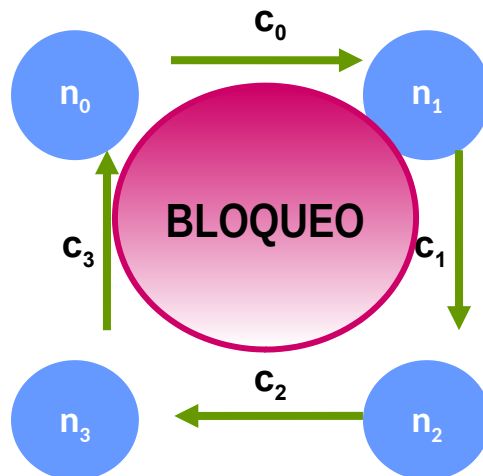
Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Evitación de interbloqueos – grafo de dependencias

Teorema: Una función de encaminamiento determinista F para una red R está libre de interbloqueos si sólo si no existen ciclos en su grafo de dependencia de canales

Ej. Anillo unidireccional

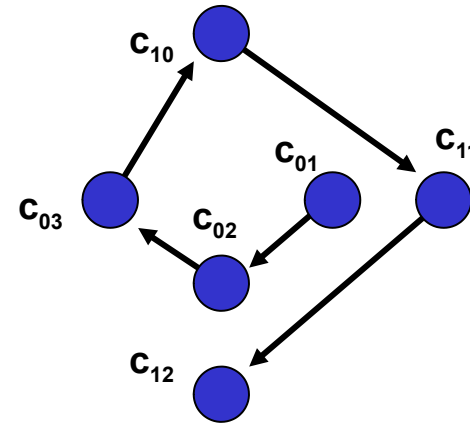
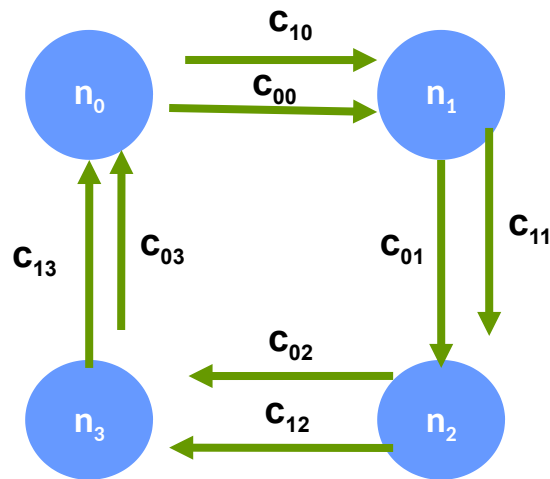


Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Evitación de interbloqueos – grafo de dependencias
Ej. Anillo unidireccional con canales virtuales



Función de encaminamiento:

Usar c_{0i} si $j < i$, o c_{1i} si $j > i$

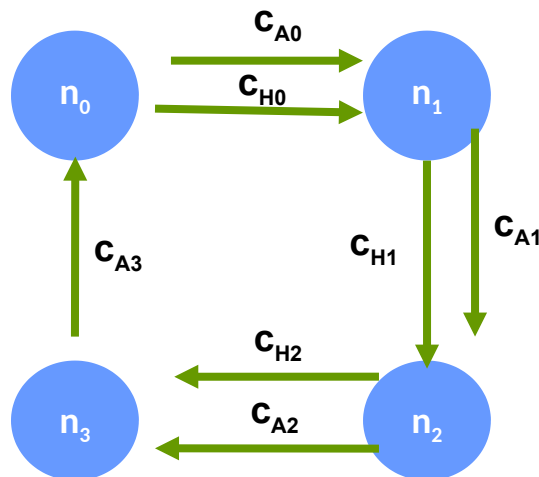
Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Evitación de interbloqueos – grafo de dependencias

Teorema: Una función de encaminamiento adaptativa F para una red R está libre de interbloqueos si, existiendo dependencias cíclicas, cada paquete encuentra un camino libre de bloqueos para llegar al destino



Función de encaminamiento:

Usar c_{Ai} si $j \neq i$, o c_{Hi} si $j > i$

Ingeniería de los Computadores

Unidad 4.4 Técnicas de conmutación

Técnicas de conmutación

Recuperación de interbloqueos

- Estrategia optimista: supone que rara vez ocurre un bloqueo
- Los recursos se asignan a los mensajes sin ninguna comprobación adicional
- Si existe bloqueo, se liberan bloqueos
- Se reasignan los recursos a otro mensaje
- Detección de bloqueos en el nodo origen

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Encaminamiento: Los algoritmos de encaminamiento establecen el camino que sigue cada mensaje o paquete

Propiedades derivadas:

- Conectividad: capacidad de encaminar desde cualquier nodo origen a cualquier nodo destino
- Adaptabilidad: capacidad de encaminar a través de caminos alternativos
- Evitación de bloqueos: capacidad de garantizar que los mensajes no se bloquearán en la red
- Tolerancia a fallos: capacidad de encaminar en presencia de componentes defectuosos

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmos de encaminamiento. Criterios de clasificación

- El número de destinos
- Quién toma la decisión del encaminamiento
- Cómo se realiza la implementación
- La adaptabilidad
- La progresividad
- La minimalidad del encaminamiento
- El número de caminos proporcionados

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmos de encaminamiento. Clasificación

Según número de destinos:

- Monodestino (*unicast*)
- Multidestino (*multicast*)

Según decisión de encaminamiento:

- Centralizados
- En origen (El nodo fuente especifica el camino y la ruta se almacena en la cabecera del paquete) → encaminamiento *street-sign*
- Distribuidos (Los nodos intermedios deciden hacia dónde encaminar) → Idóneo para topologías irregulares
- Multifase

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmos de encaminamiento. Clasificación

Según la implementación:

- Tablas (encaminamiento por intervalos)
- Máquinas de estados finitos (FSM) → topologías ortogonales (encaminamiento por orden de dimensión)

Según adaptabilidad

- Deterministas:
 - Siempre suministran el mismo camino
 - Rendimiento pobre si tráfico no uniforme
- Adaptativos:
 - Consideran el estado de la red
 - Totalmente adaptativos: pueden usar todos los canales
 - Parcialmente adaptativos: usan un subconjunto

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmos de encaminamiento. Clasificación

Según progresividad:

- Progresivos
- *Backtracking*: EPB (*Exhaustive Profitable Backtracking*)

Según minimalidad:

- Mínimos
 - ¿Algoritmos deterministas progresivos y mínimos?
- No mínimos
 - Mayor flexibilidad
 - Encaminamiento tolerante a fallos

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmo determinista: encaminamiento por orden de dimensión

- Topologías ortogonales
- Selección de canales sucesivos con orden específico
- Tipo determinístico
- La diferencia en una dimensión se anula antes de pasar a la siguiente
- Ejemplos:
 - *Street-sign* (fuente y sin tabla)
 - encaminamiento XY (distribuido y sin tabla)
 - encaminamiento *e-cube*
 - Intervalo (distribuido y con tabla de consulta)
- Libre de interbloqueos en mallas e hipercubos (en toros es necesario usar canales virtuales y establecer un orden en su utilización)

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Encaminamiento XY (ordenado por dimensión)

Ej. 3: malla 2D Ord. por dimensión 0-1

Entrada: actual $A=(a_1, a_0)_k$

destino $D=(d_1, d_0)_k$

Salida: Canal $cs=(D0-,D0+,D1-,D1+,I)$.

Procedimiento:

$dist0 = d_0 - a_0$; $dist1 = d_1 - a_1$;

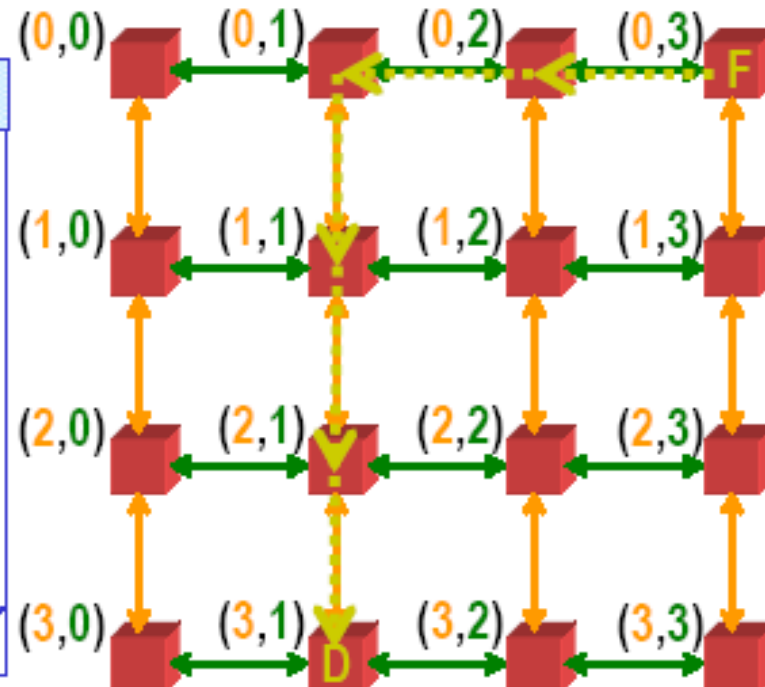
if ($dist0 < 0$) $cs = D0-$;

if ($dist0 > 0$) $cs = D0+$;

if ($dist0 = 0$ & $dist1 < 0$) $cs = D1-$;

if ($dist0 = 0$ & $dist1 > 0$) $cs = D1+$;

if ($dist0 = 0$ & $dist1 = 0$) $cs = I$;



Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Encaminamiento ordenado por dimensión en tablas



Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Modelo de giros (*turn-model*)

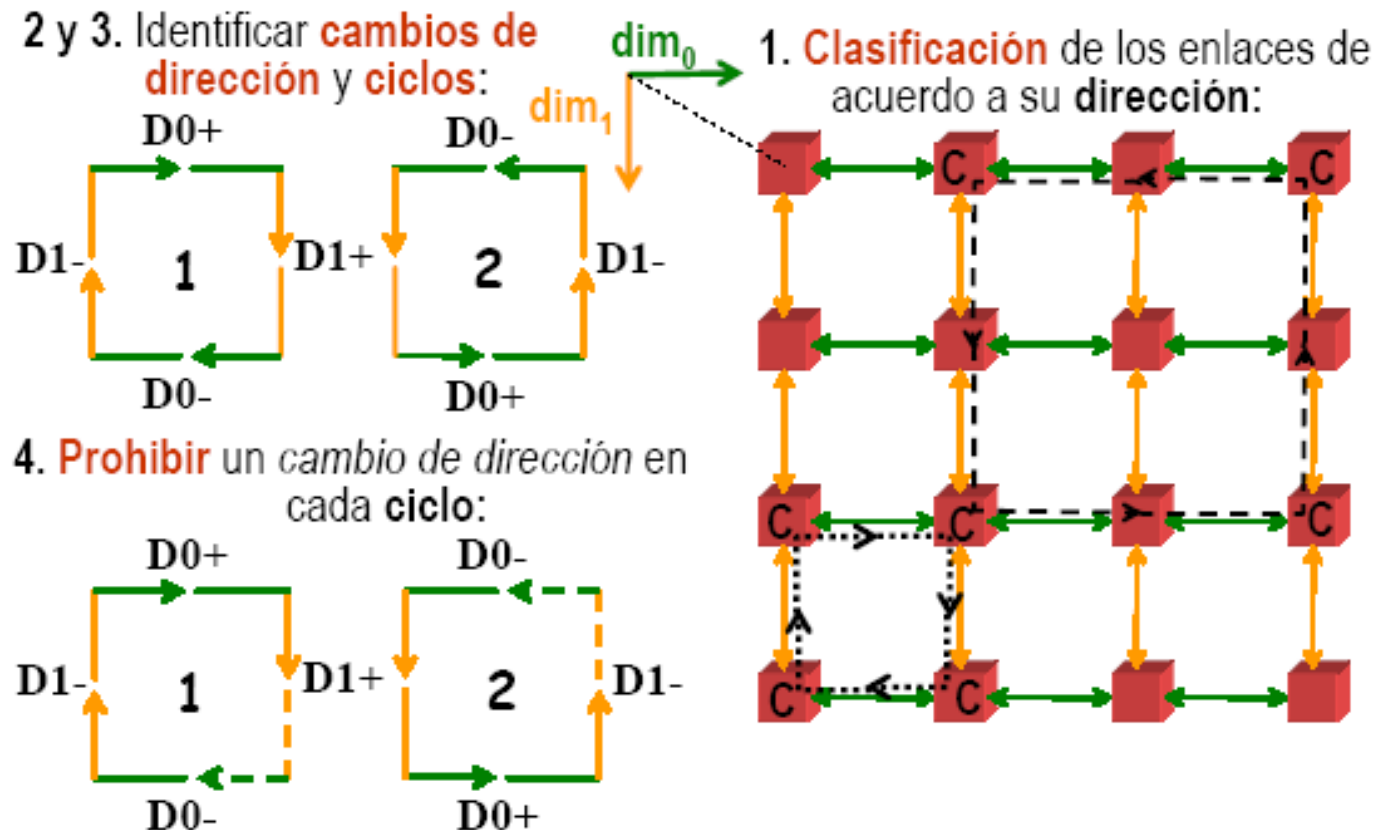
- Redes estáticas (topologías ortogonales) y redes dinámicas
- Ejemplo:
 - *West-First* en mallas 2D (distribuido, sin tablas, parcialmente adaptativo y puede ser mínimo o no mínimo)
- Interbloqueos
 - Ciclos que engloban varias direcciones → Se evitan prohibiendo al menos un cambio de dirección para cada ciclo
 - Ciclos sin cambio de dirección → Se evitan añadiendo canales virtuales y estableciendo un orden de uso

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmo West-First

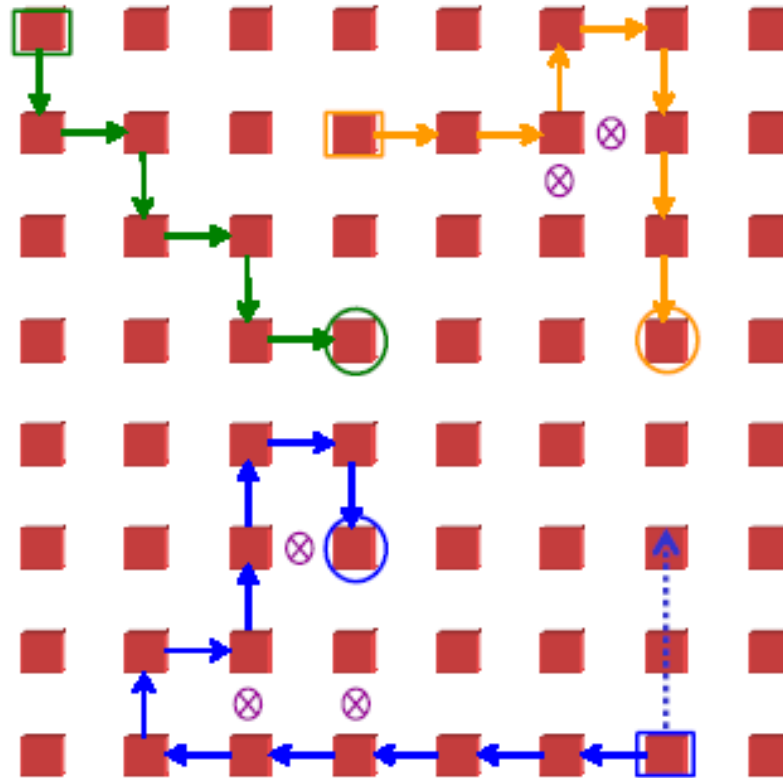


Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmo West-First – implementación no mínima



Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmo *West-First* para mallas 2D

Entrada: Actual $A = (a_1, a_0)$

Destino $D = (d_1, d_0)$

Salida: Canal cs

Procedimiento:

$dist0 = d_0 - a_0$; $dist1 = d_1 - a_1$;

if ($dist0 < 0$) $cs = D0-$;

if ($dist0 > 0$ & $dist1 > 0$) $cs = Sel(D0+, D1+)$;

if ($dist0 > 0$ & $dist1 < 0$) $cs = Sel(D0+, D1-)$;

if ($dist0 > 0$ & $dist1 = 0$) $cs = D0+$;

if ($dist0 = 0$ & $dist1 > 0$) $cs = D1+$;

if ($dist0 = 0$ & $dist1 < 0$) $cs = D1-$;

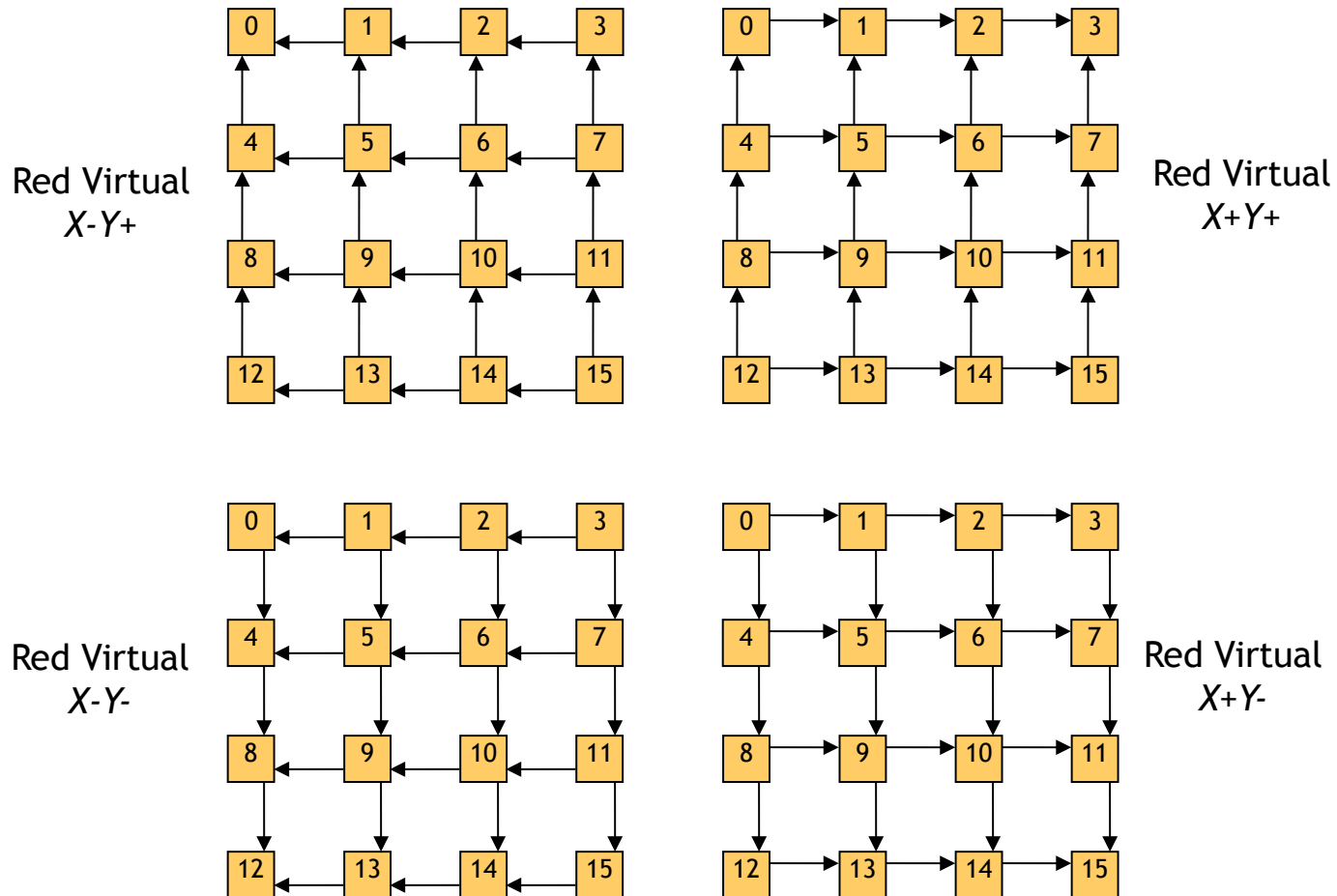
if ($dist0 = 0$ & $dist1 = 0$) $cs = I$;

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Algoritmo totalmente adaptativo: redes virtuales



Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Encaminamiento en redes tipo mariposa

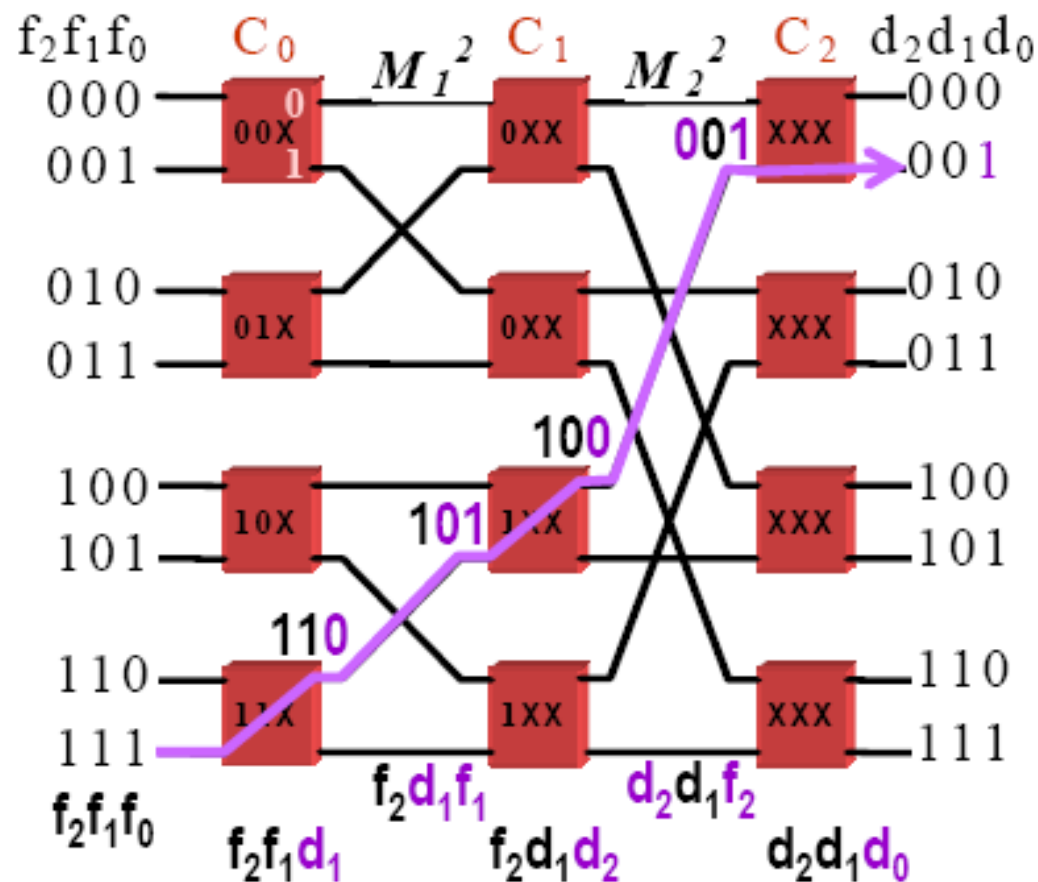
- Tipo determinístico
- Se usa la dirección del destino D en base b . Cada dígito controla una etapa de conmutadores
- No hay interbloqueo porque la topología no presenta ciclos
- Ejemplos:
 - ✓ Red Omega y red Cubo: d_i controla la etapa $n-i-1$
 - ✓ Red mariposa: d_i controla la etapa $i-1$ y d_0 la etapa $n-1$

Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Encaminamiento en redes tipo mariposa



Ingeniería de los Computadores

Unidad 4.5 Técnicas de encaminamiento

Técnicas de encaminamiento

Encaminamiento en redes tipo mariposa

