ARQUITECTURA DE COMPUTADORES II

AUTORES:
David Expósito Singh
Florin Isaila
Daniel Higuero Alonso-Mardones
Javier García Blas
Borja Bergua Guerra

Área de Arquitectura y Tecnología de Computadores
Departamento de Informática
Universidad Carlos III de Madrid

Julio de 2012

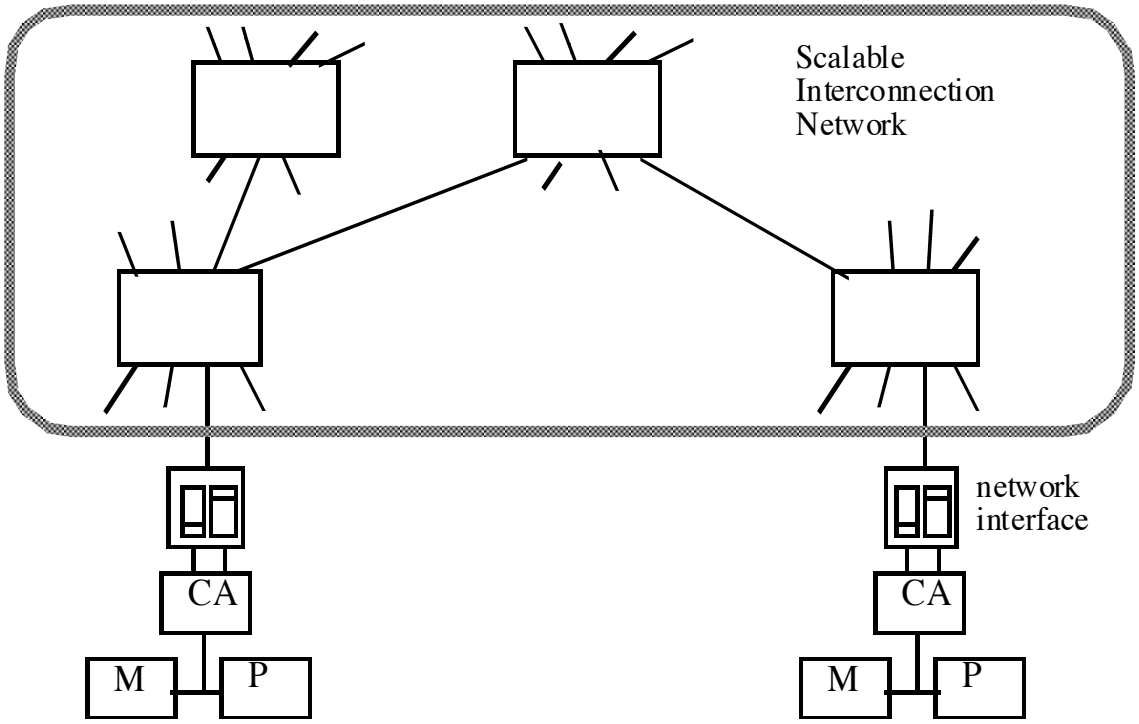
TEMA 5:
REDES DE INTERCONEXIÓN

Índice

Introducción

- Red de interconexión:
  - Sistema de interconexión: elemento *hardware* que permite la comunicación entre los *nodos* en un MP.
  - Elemento clave en una arquitectura paralela.
  - Objetivos:
    - Minimizar la latencia de comunicación.
    - 'Escalabilidad'.
    - Minimizar el coste.
    - Conseguir tantas comunicaciones simultáneas como sean necesarias.

Red de interconexión genérica



Introducción (II)

- Conceptos básicos y definiciones:
  - **Enlaces** (*links*):conjunto de cables (o fibras) que conducen una señal.
  - **Conmutadores** (*switches*): conectan un conjunto de **canales** de entrada a un conjunto de canales de salida.
  - **Canal**: emisor → enlace → receptor

Introducción (III)

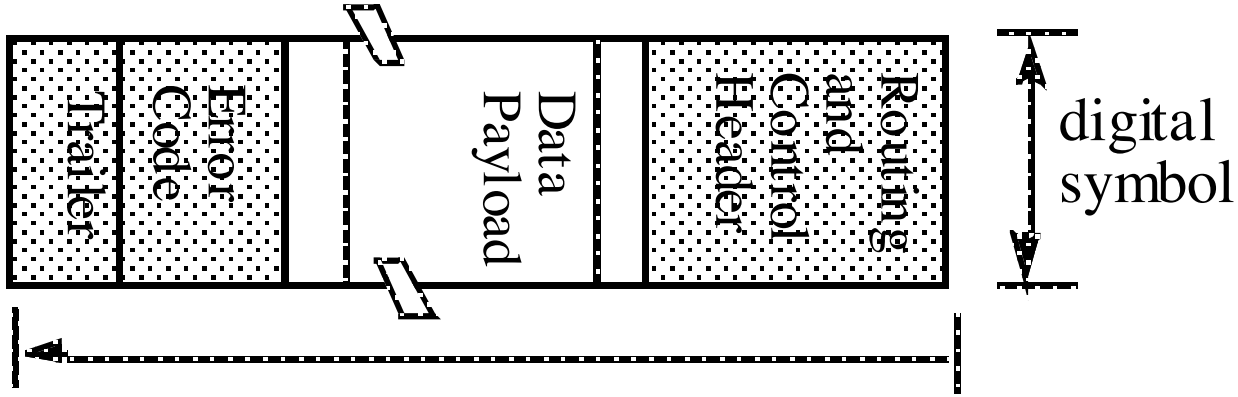
- Formalización de una red:
  - **Red**: grafo V = {nodos y conmutadores} conectados por un conjunto de **canales** $C \subseteq V \times V$
  - **Canal**: ancho w bits y la señal una frecuencia $f = 1/\tau$:
    - Ancho de banda del canal: $b = wf$ *bits/sec*.
    - ***phit*** [unidad física]: datos/ciclo
    - ***flit***: unidad básica de control en las transferencias.
  - **Grado del conmutador**: nº de canales que conmuta: entrada → salida
  - **Ruta**: secuencia de enlaces y conmutadores seguida por un mensaje

Introducción (IV)

- Caracterización de una red:
  1. **Topología**:
     - Estructura física de la interconexión
  2. Algoritmo de **encaminamiento** [*routing*]:
     - Decide el recorrido de un mensaje
  3. Estrategia de **conmutación**:
     - Cómo los datos en un mensaje atraviesan una ruta
  4. Mecanismo de **control de flujo**:
     - Cuándo un mensaje (o partes de) atraviesan una ruta

Introducción (VI)

- Los mensajes se descomponen en **paquetes**:



Sequence of symbols transmitted over a channel

Introducción (V)

- Evaluación de la **latencia** de comunicación:

latencia $(n)_{o \rightarrow d}$ = *overhead* +

+ ocupación del canal

+ retardo de encaminamiento

+ retardo por "contención"

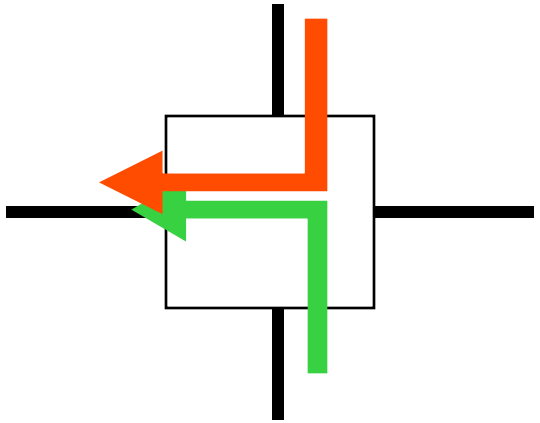
- Overhead: tiempo necesario para iniciar la operación de envío/recepción de un mensaje.
- Ocupación del canal = $(n+n_e)/b$
  - n: *data payload*
  - n_e: *packet envelope*
  - b: ancho de banda
- Retardo de encaminamiento: asociado al retardo del *switch*.
- Contención: conflicto en el acceso a un mismo recurso (*switch* o nodo).

Objetivos

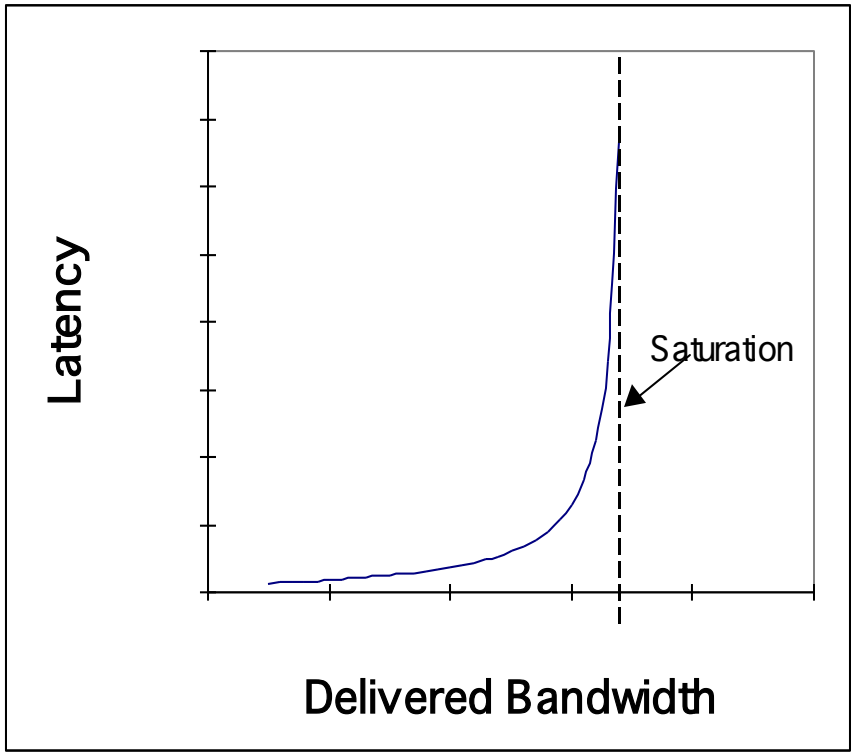
- Latencias menores posibles.
- Anchos de banda efectivos mayores posible.
- El mayor número de transferencias simultáneas
  - Grado de bisección.
- El menor coste posible.

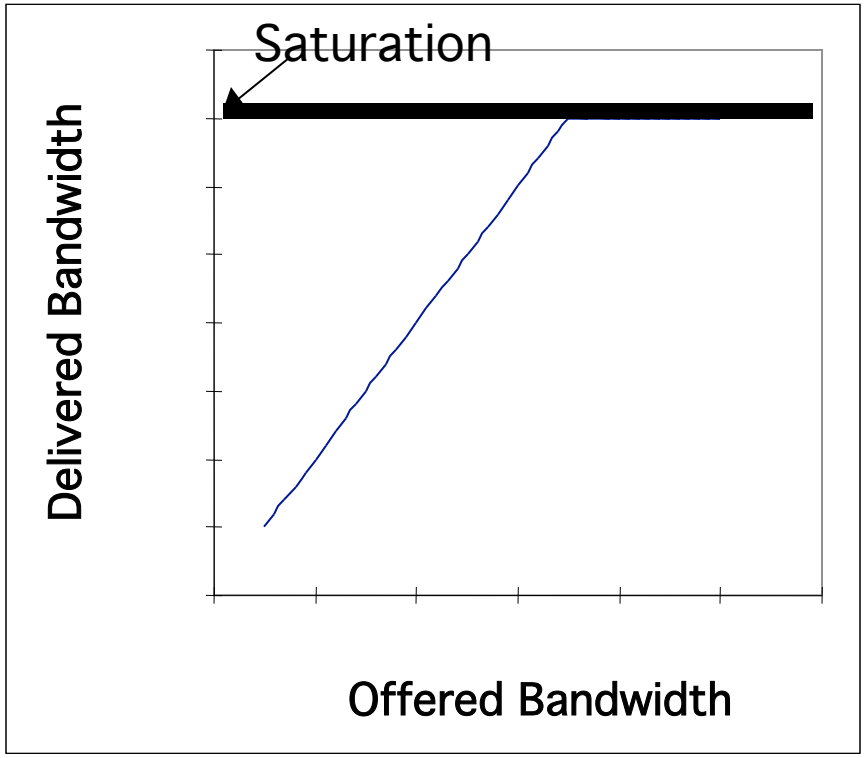
Introducción (VII)

- "Contención":
  - Dos paquetes que tratan de utilizar a la vez un mismo recurso.
  - Soluciones: ¡cuestión muy compleja!
    - control del flujo en el enlace.
    - Problema del árbol de saturación .
    - Multiprocesador como sistema cerrado.



Saturación


Latency
Saturation
Delivered Bandwidth


Saturation
Delivered Bandwidth
Offered Bandwidth

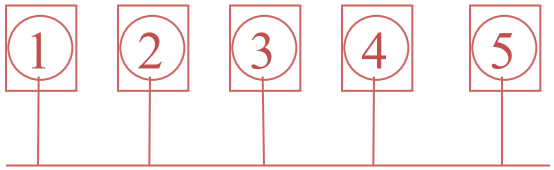
Topologías

- Índice:
  - Propiedades "topológicas"
  1. *Arrays* lineales y anillos
  2. Mallas y toros multidimensionales
  3. Árboles
  4. Mariposas
  5. Hipercubos
  - Comparación
  - Casos reales

Topologías

- Propiedades:
  - **Grado** (del *switch*): Número de canales de entrada - salida
  - **Distancia** (en una ruta): Número de enlaces que es necesario atravesar. Como mínimo es el camino más corto entre los dos nodos
  - **Distancia media:** de la distancia entre todos los pares de nodos
  - **Diámetro:** Longitud del máximo camino más corto entre dos nodos
  - **Bisección:** número de enlaces que se necesita suprimir para obtener dos redes iguales

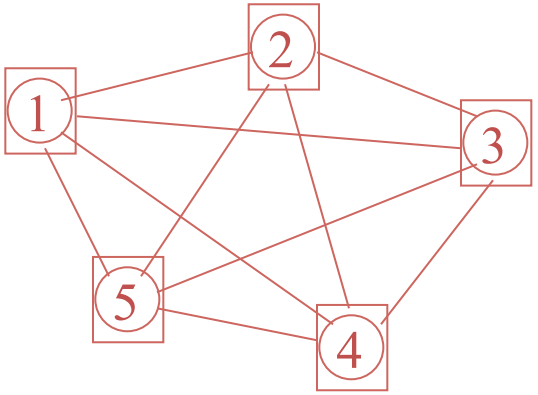
Bus (Red ethernet)



- Grado= 1
- Diámetro= 1
  - No es necesario enrutar
- Bisección= 1

Opción más simple y barata

Grafo completamente conectado



- Grado= n-1

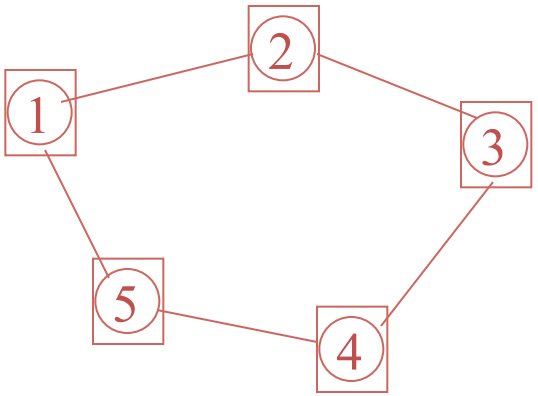
  Demasiado cara para grandes redes.
- Diámetro= 1
- Bisección=$\lfloor n/2 \rfloor \lceil n/2 \rceil$

Red estática

Cada nodo está conectado con los demás

Al dividir la red en dos cada nodo se conecta con n/2 nodos. Hay n/2 nodos.

Anillo



- Grado= 2
- Diámetro= $\lfloor n/2 \rfloor$

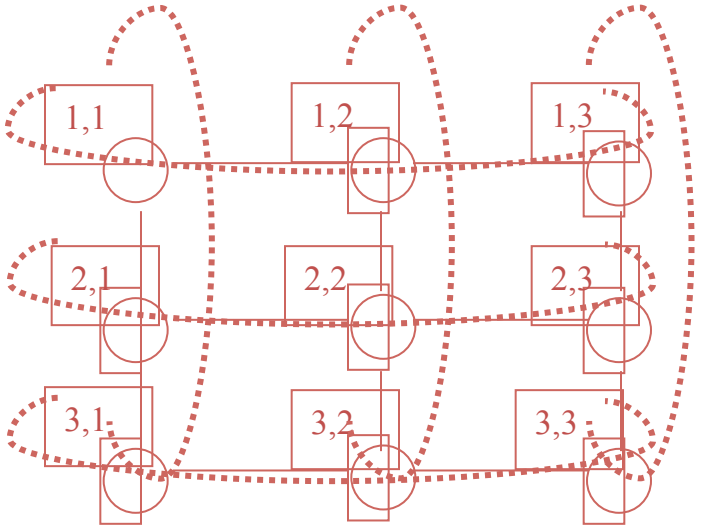
  Poco eficiente cuando hay muchos nodos
- Bisección= 2

Red estática

Nodo i conectado con nodos i+1 e i-1 módulo n.

– Ejemplos: FDDI, SCI, FiberChannel Arbitrated Loop, KSR1

Malla d-dimensional



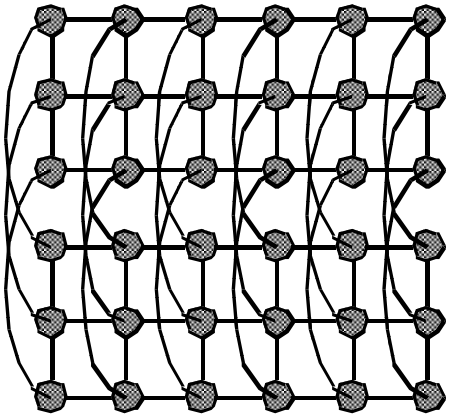
Para d dimensiones

- Grado= d
- Diámetro = d ($\sqrt[d]{n} - 1$)
- Bisección = ($\sqrt[d]{n}$) d–1

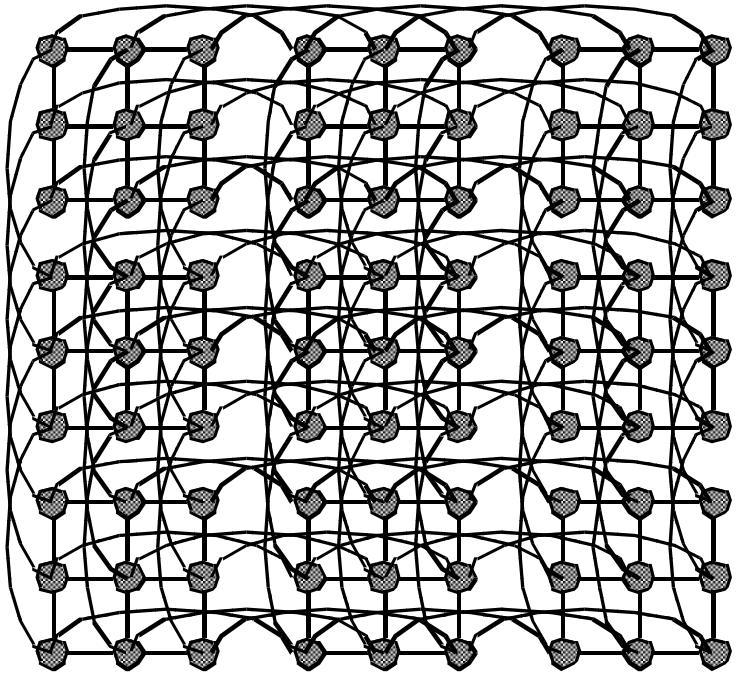
Red estática

Cray T3D yT3E.

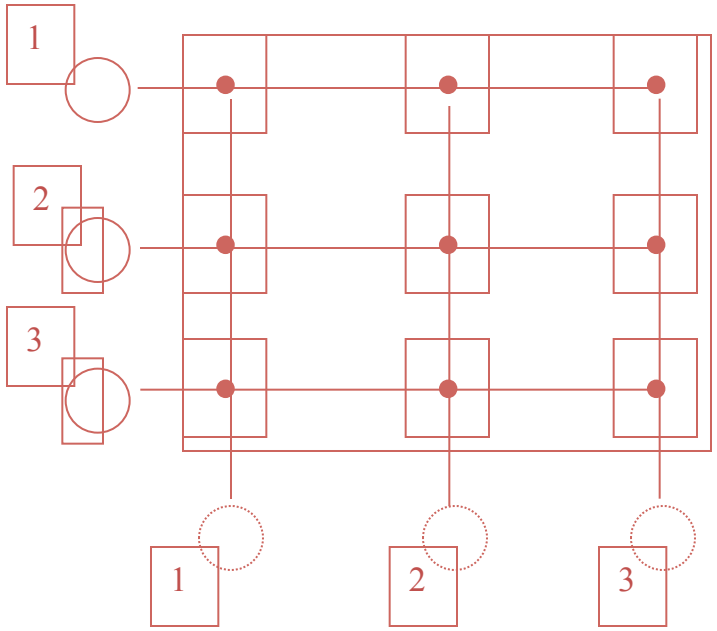
Mallas y toros multidimensionales



6 x 3 x 2



Crossbar



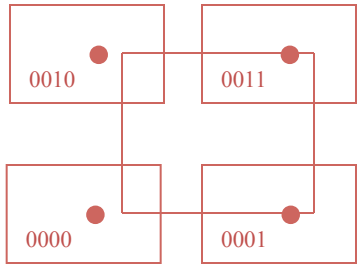
- Rápida y cara (n^2 switches)
- Normalmente empleada en conexiones entre procesadores y memorias
- Grado= 1
- Diámetro = 2
- Bisección= n/2

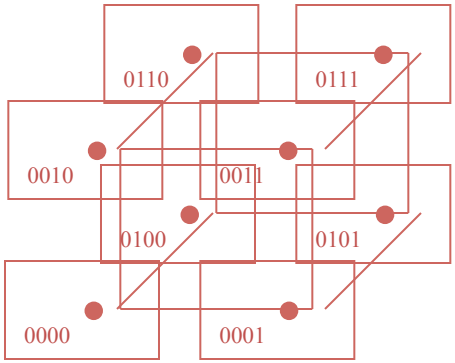
Ex: 4x4, 8x8, 16x16

• switch

Red dinámica

Hipercubo





Red estática

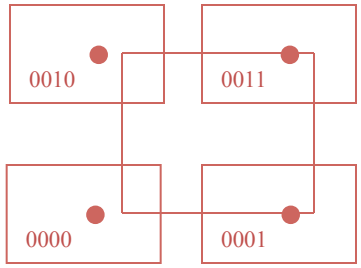
Distancia Hamming =

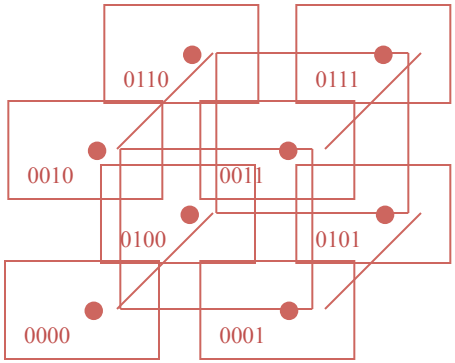
Número de bits en que se difiere la representación de dos números

Dos nodos están conectados si la distancia Hamming es 1

Encaminamiento de x a y mediante un gradiente empleando la distancia Hamming

Hipercubo





Intel iPSC/860,
SGI Origin 2000

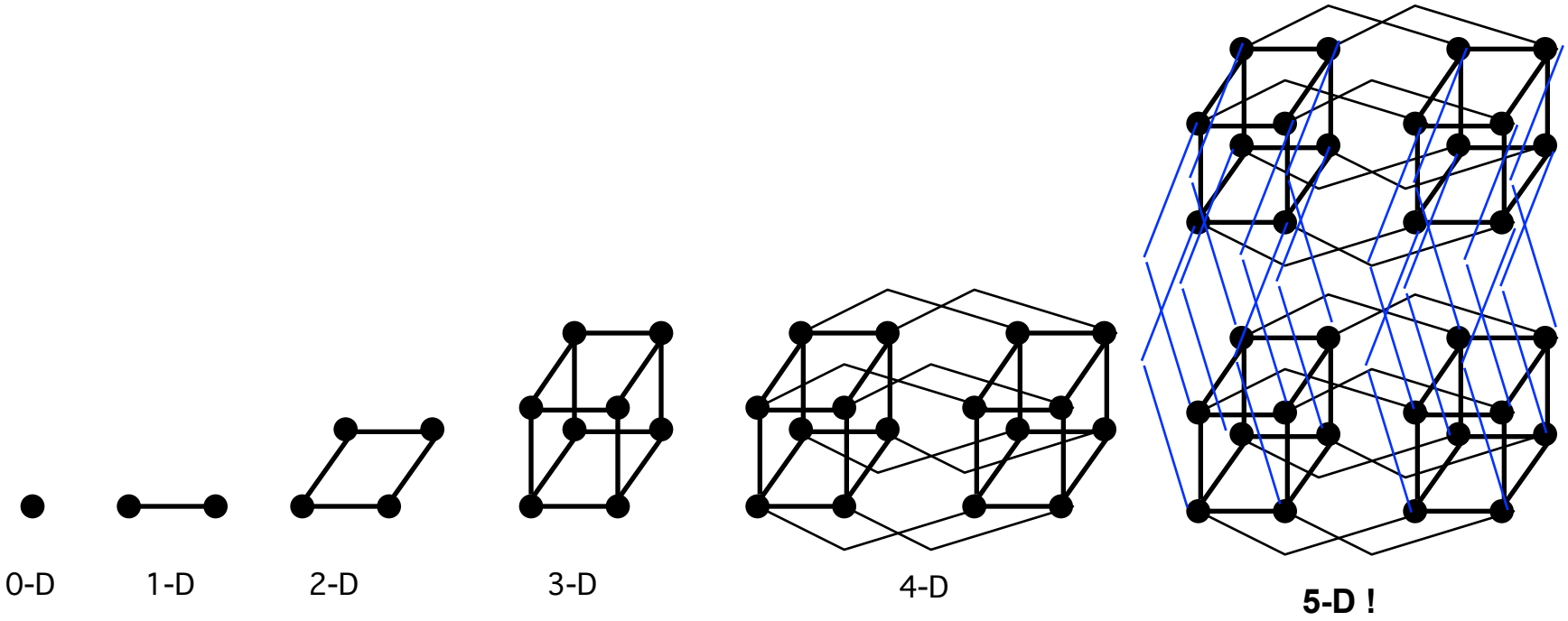
k dimensiones, n= 2^k nodos

- Grado= k
- Diámetro = k
- Bisección= n/2

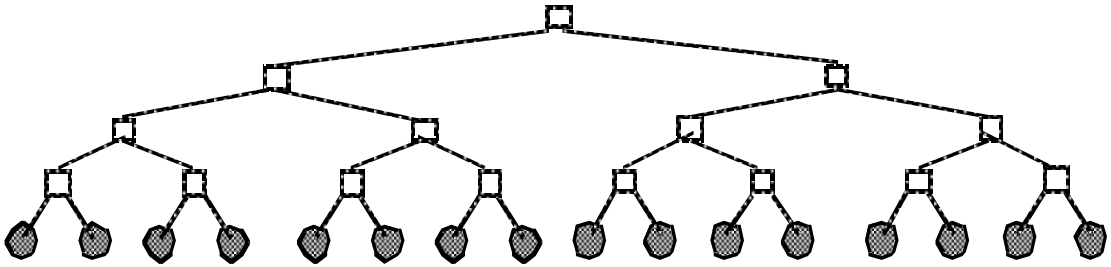
  Dos hipercubos (k-1) están conectados con n/2 vértices para formar un hipercubo de grado k.

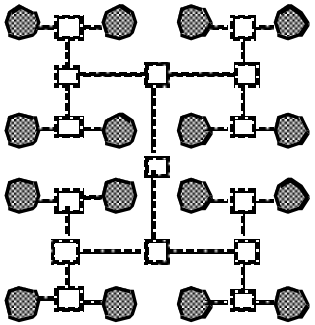
Hipercubos

- Tb. se llaman “n-cubos binarios”
- # de nodos = $N = 2^n$



Árboles [*trees*]



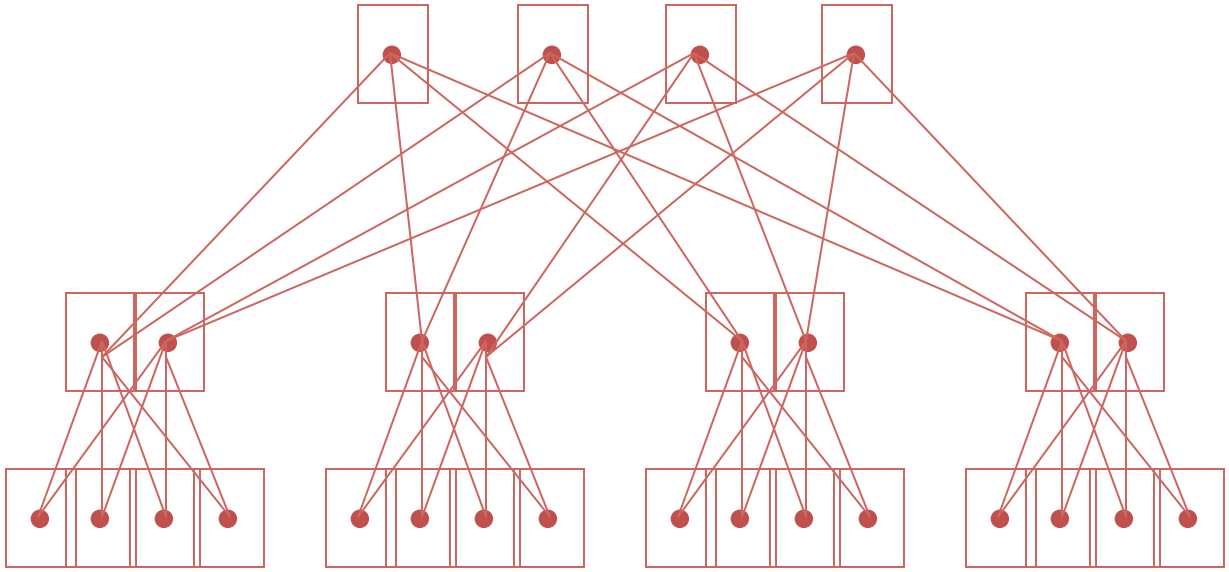


Fat Tree Clos-Network

- Diámetro árbol $2\log_2 n$
- Bisección árbol simple= 1
  - Aparición de cuello de botella
- Fat Tree:
  - Vértices en el nivel i tienen el doble de capacidad que en el nivel i-1
  - En el nivel i se emplean switches con 2^i entradas y 2^i salidas
  - También conocidas como Clos-networks

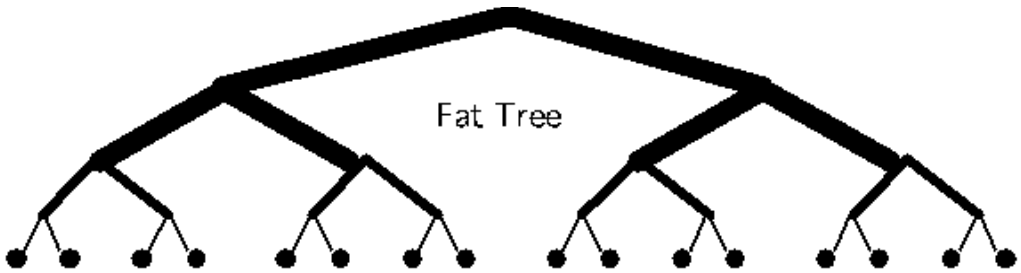
Fat Tree/Clos-Network

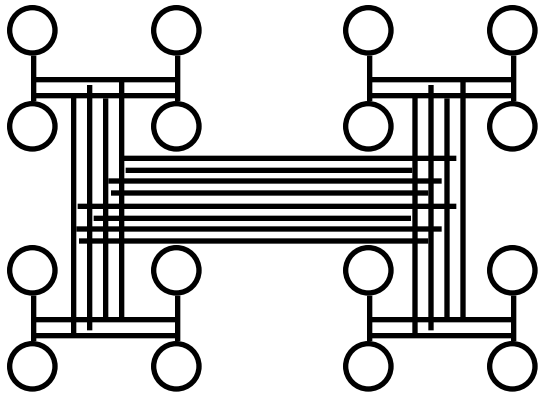
- Encaminamiento:
  - A través del nivel superior
  - Si hay alternativas, se elige de forma aleatoria.
  - Tolerancia a fallos.
- Diámetro $2\log_2 n$, bisección: n



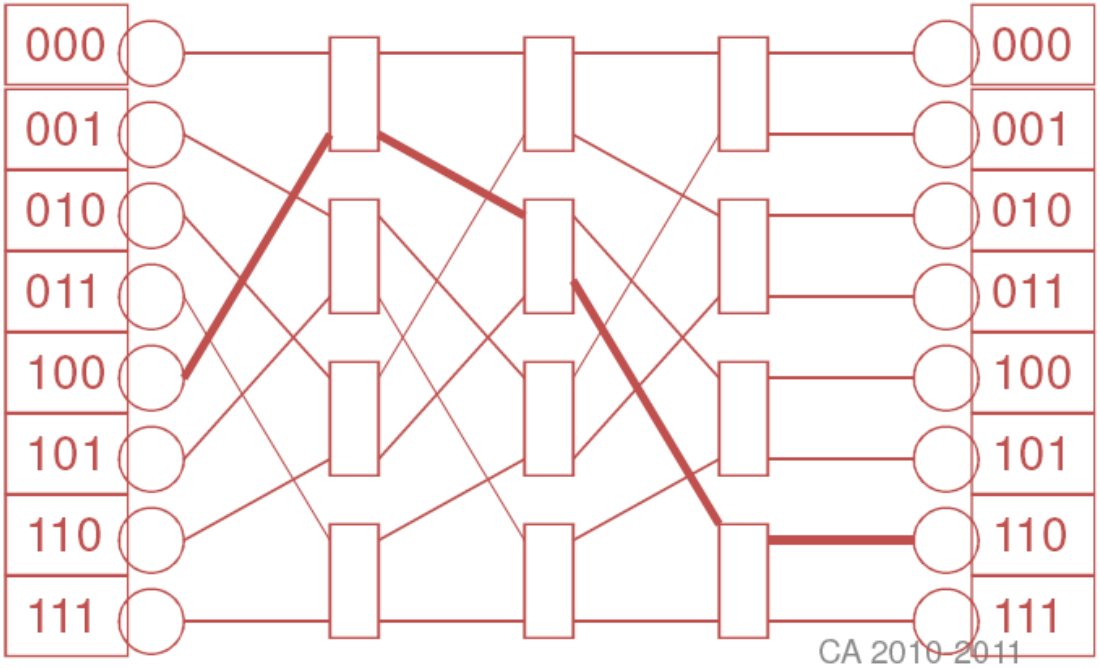
CM-5

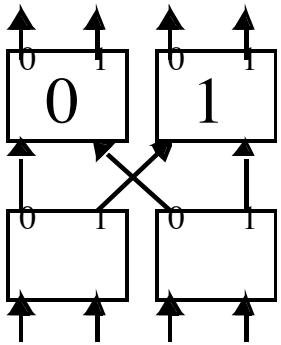
Fat-trees





Omega Network





building block

- Grado = 2 para los nodos, 4 para los swtiches
- Diámetro = $\log_2(n)$
- Bisección = n/2 (en promedio con comunicaciones homogéneas)

  = 1 (envío a un nodo) y n (envío cada nodo a si mismo).

Comparación

Topología	Grado	Diámetro	Distancia	Bisección	Dist. med. @ P=1024
1D Array	2	N-1	N / 3	1	huge
1D Ring	2	N/2	N/4	2	
2D Mesh	4	$2 (N^{1/2} - 1)$	$2/3\ N^{1/2}$	$N^{1/2}$	63 (21)
2D Torus	4	$N^{1/2}$	$1/2\ N^{1/2}$	$2N^{1/2}$	32 (16)
k-ary n-cube	2n	nk/2	nk/4	nk/4	15 (7.5) @n=3
Hypercube	n=log N	n	n/2	N/2	10 (5)

TOP 500

- Listado de **Interconnect Family share**
  - Número de equipos vs rendimiento

Topologías

Nombre	Topología
10GBit Ethernet	Star or Fat Tree
Infiniband 12x	Fat Tree
Myrinet	Clos
NUMAlink4	Fat Tree
Quadrics	Fat Tree
SCI/Dolphin	2D/3D Torus

Comparación de redes

Interconnect	Latency (microseconds)	Bandwidth (MBps)
GigE	**~29-120**	**~125**
GigE: GAMMA	~9.5 (MPI)	~125
GigE with Jumbo Frames	29-120	~125
GigE: Level 5	15	104.7
10 GigE: Chelsio (Copper)	9.6	~862
Infiniband: Mellanox Infinihost (PCI-X)	4.1	760
Infiniband: Mellanox Infinihost III EX SDR	2.6	938
Infiniband: Mellanox Infinihost III EX DDR	2.25	1502
Infinipath: HTX	**1.29**	**954**
Infinipath: PCI-Express	1.62	957.5
Myrinet D (gm)	~7.0	~493
Myrinet F (gm)	~5.2	~493
Myrinet E (gm)	~5.4	~493
Myrinet D (mx)	3.5	~493
Myrinet F (mx)	2.6	~493
Myrinet E (mx)	2.7	~493
Myri-10G	**2.0**	**1,200**
Quadrics	**1.29**	**~875-910**
Dolphin	4.2	457.5

Comparación de redes

Interconnect	8 Node Cost	24 Node Cost	128 Node Cost
GigE[1]	**$258.00**	**$944.00**	**$27,328.00**
GigE: GAMMA[2]	$258.00	$944.00	$27,328.00
GigE with Jumbo Frames[3]	$308.00	$944.00	$27,328.00
GigE: Level 5[4]	$4,060	$12,200	$83,360.00
10 GigE: Chelsio (Copper)[5]	$15,960.00	$62,280.00	$447,360.00
Infiniband: Voltaire[6]	$11,877.00	$23,084.00	$182,083.00
Infinipath[7]	**$13,810.00**	**$26,530.00**	**$207,860.00**
Myrinet D (gm/mx)[8]	$7,200.00	$21,600.00	$115,200.00
Myrinet F (gm/mx)[9]	$8,000.00	$24,000.00	$128,000.00
Myrinet E (gm/mx)[10]	$12,000.00	$36,000.00	$192,000.00
Myri-10G[11]	**$9,600.00**	**$28,800.00**	**$153,600.00**
Quadrics[12]	**$13,073.00**	**$43,698.00**	**$205,538.00**
Dolphin[13]	$7,800.00	NA	$140,160.00

Commutación

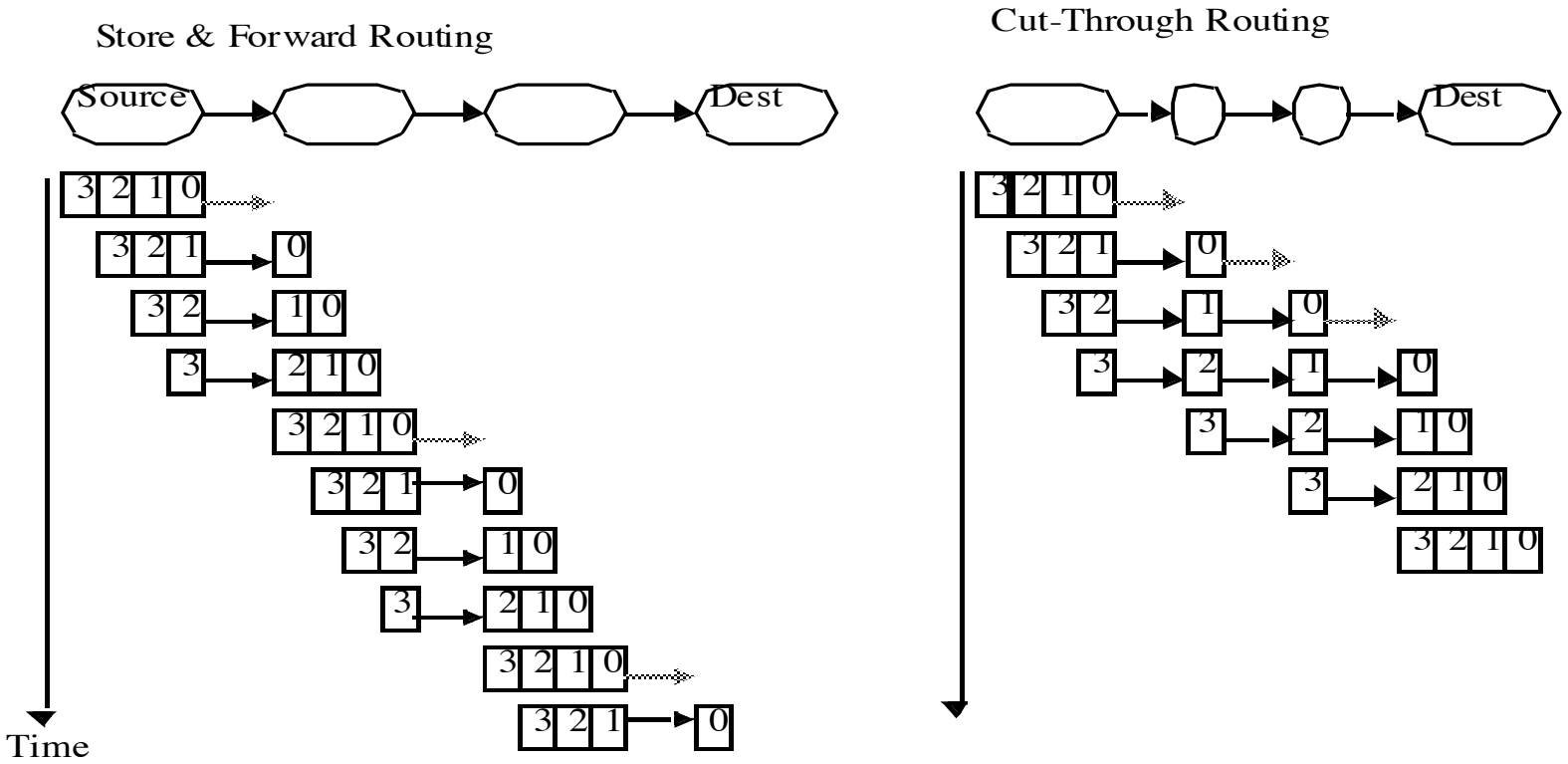
- Especifica cómo un mensaje recorre la red de un nodo a otro.
- Conmutación de paquetes
  - Un mensaje se divide en diferentes paquetes que se pueden enviar a través de rutas distintas.
  - Mejor utilización de los recursos de red.
- Conmutación de circuitos
  - Es establece el recorrido entre la fuente y el destino
  - Todos los paquete recorren el mismo camino

Commutación de paquetes

- Store-and-fordward
  - Cada paquete se almacena en el *switch* antes de enviarlo.
  - Cada paquete está en al menos dos nodos.
  - Memoria de almacenamiento.

- Cut-through
  - El encaminamiento se realiza en base a la cabecera.
  - El resto de los paquetes son reenviados siguiendo el mismo camino.
  - Estrategia similar a la conmutación de circuitos.

Conmutación:
store-and-forward vs. *cut-through*

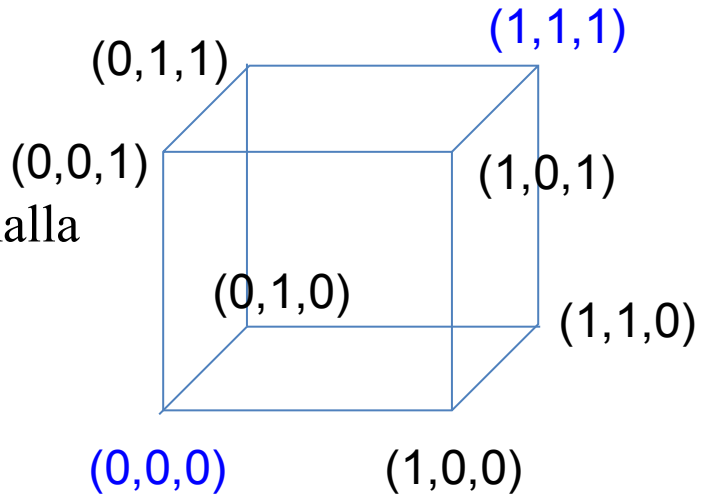


Encaminamiento

- El algoritmo de encaminamiento determina:
  - Cuál de todos los caminos posibles se usa como ruta
  - Cómo se determina la ruta
- Aspectos:
  - Mecanismo de encaminamiento
    - aritmético
    - selección basada en el origen
    - guiado por una tabla
    - computación
  - Propiedades de las rutas

Mecanismo de encaminamiento

- Selecciona el canal de salida para cada paquete entrante
- Aritmético:
  - Redes regulares
  - Ejemplo: Δx, Δy encaminamiento en una malla
    - oeste (-x) $\Delta x < 0$
    - este (+x) $\Delta x > 0$
    - sur (-y) $\Delta x = 0, \Delta y < 0$
    - norte (+y) $\Delta x = 0, \Delta y > 0$
    - local $\Delta x = 0, \Delta y = 0$
  - El switch requiere ALUs para encaminar.



Mecanismo de encaminamiento

- Basado en el origen:
  - La cabecera del mensaje contiene la indicación de los puertos seleccionados
  - Ejs.: Myrinet, Meiko CS-2

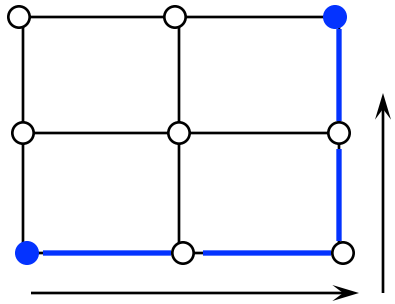
- Guiado por una tabla:
  - La cabecera del mensaje y / o la tabla contiene el índice para el siguiente tramo
  - Ejs. ATM, HPPI

Propiedades de los algoritmos de encaminamiento

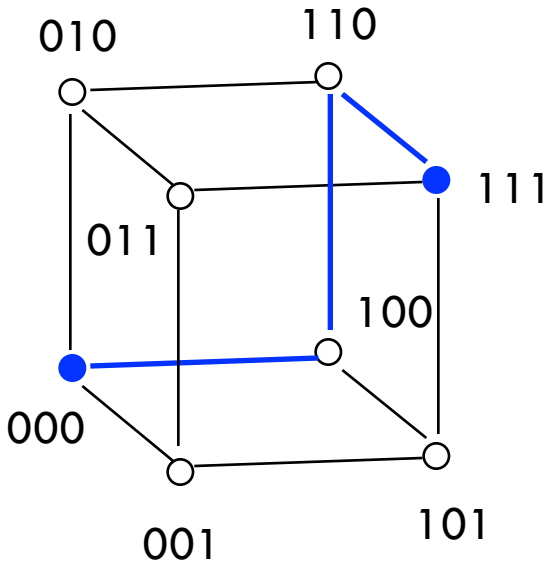
- Determinísticos:
  - La ruta vendrá determinada por el origen y el destino y no por elementos intermedios
- Adaptativos:
  - La ruta se podrá ver influida por el tráfico que se encuentre en el camino
- Minimales:
  - Sólo seleccionan siempre el camino más corto
- Libres de interbloqueos (*deadlocks*):
  - No se puede dar ninguna situación de tráfico en que se bloqueen todos los paquetes

Determinístico vs. Adaptativo

- Determinístico
  - K-cubo
    - (x1, y1) → (x2, y2)
    - Primero Dx = x2 - x1,
    - Después Dy = y2 - y1
  - Puede producir contención



- Adaptativo: ruta determinada por el grado de contención



Adaptativo

- Algoritmos determinísticos simples pueden originar problemas de contención
- Esencial para tolerancia a fallos.
- Puede mejorar el rendimiento de la red.

Encaminamiento

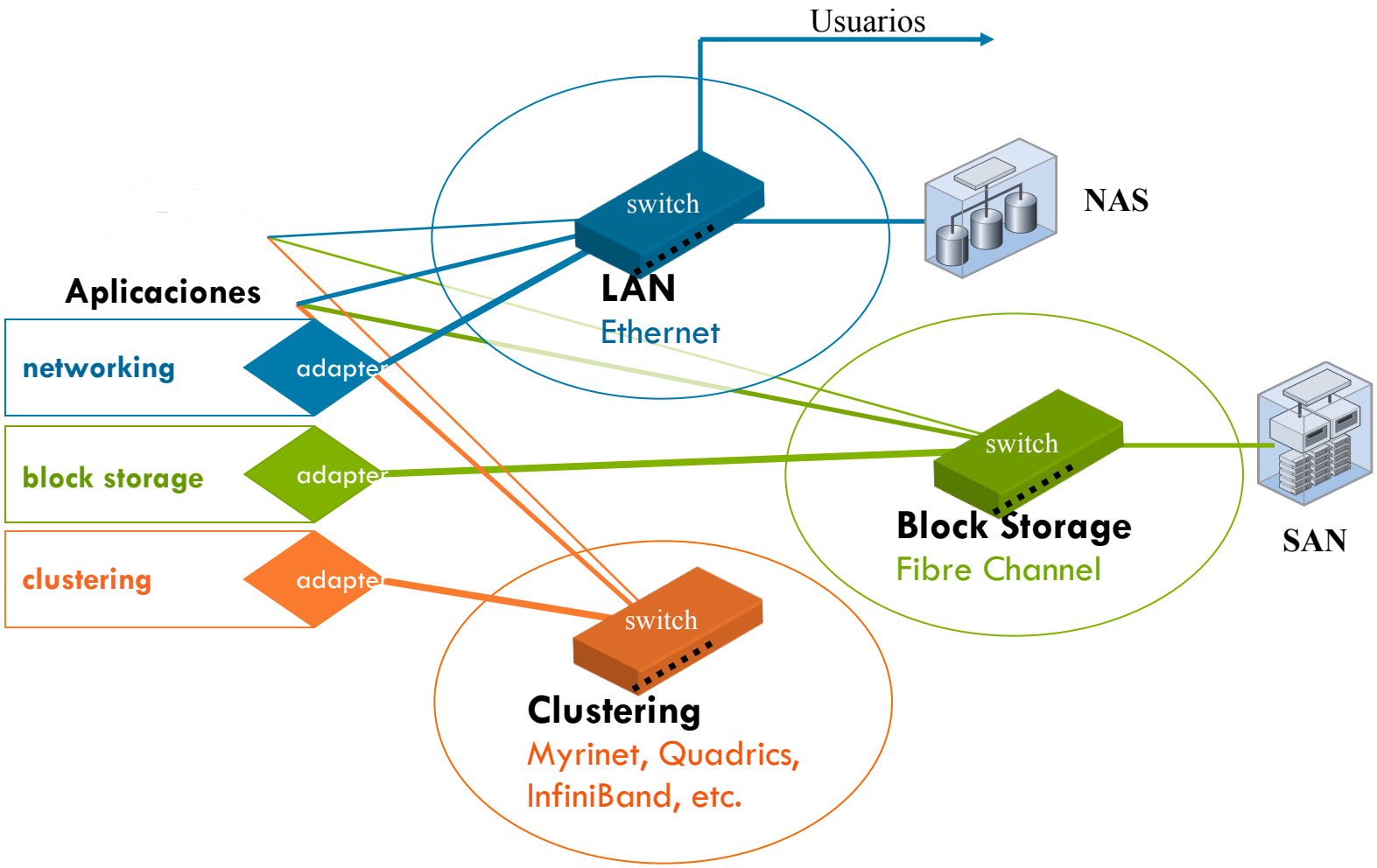
Basado en el origen

- La fuente especifica la ruta completa del paquete
- Switches de diseño simple
  - No estado de control.
- Myrinet
- No es adaptativo

Basado en tablas

- Pequeña cabecera: contiene un índice de la tabla de enrutamiento.
- Problemas: gestionar grandes tablas de enrutado (almacenar y actualizar).

Organización de un centro de cómputo



Bibliografía

- Culler et al., *Parallel Computer Architecture: A Hw/Sw Approach*, Capítulo 10.