

Redes

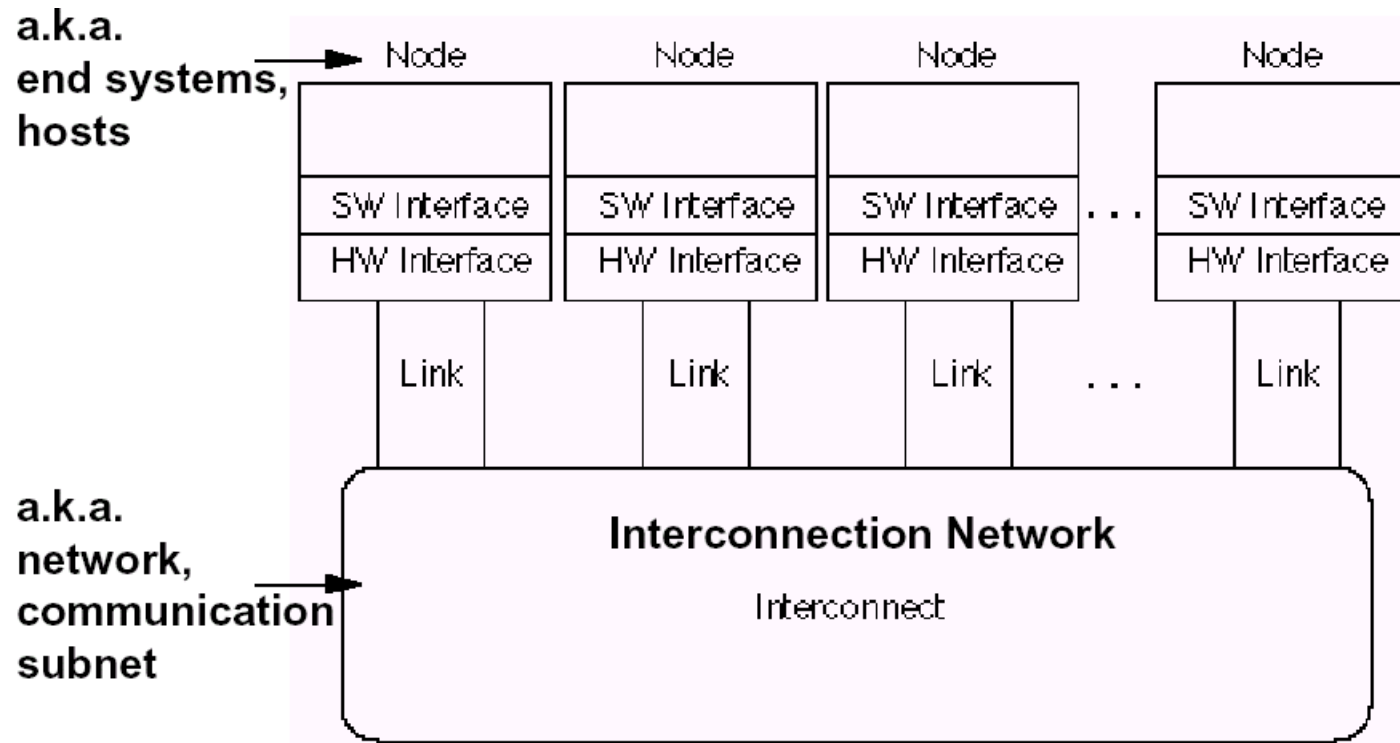
- ❖ Introducción
- ❖ Una red elemental
- ❖ Concepto de protocolo
- ❖ Medidas de rendimiento de una red
- ❖ Conexión computador-red
- ❖ Conexión de más de dos computadores
- ❖ Arquitecturas de protocolo

1. Introducción: objetivos y motivación

- ❖ Objetivo principal:
 - Comunicación entre computadores
- ❖ Objetivo eventual:
 - Formar un **gran computador** conectando una colección de computadores que comparten recursos (*cluster*)
- ❖ Redes y diseño de computadores
 - También **dentro del computador** los **conmutadores** (típicos de redes) están reemplazando a los buses como base de la comunicación => Para diseñar un computador hay que conocer técnicas propias de redes
 - Los computadores de hoy deben estar **capacitados para conectarse a redes**

Introducción:

Estructura genérica de una red



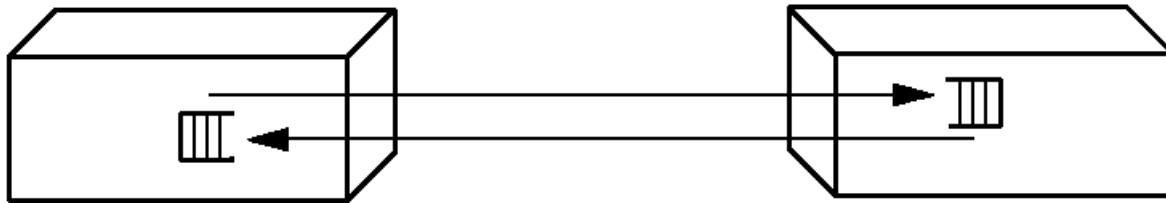
Introducción: tipos de redes

- ❖ En función del número de nodos y su proximidad:
 - **WAN** (*Wide Area Network*)
 - ✓ **Miles** de computadores distribuidos por todo el mundo.
Ejemplo: **ATM**
 - **LAN** (*Local Area Network*)
 - ✓ **Cientos** de computadores distribuidos en un edificio o en un campus. Ejemplo: **Ethernet**
 - **SAN** (Originalmente “*System A. N.*”, pero recientemente “*Storage A. N.*”)
Puede conectar **cientos** de nodos a **menos de 100 m** de

Tipos de redes (cont.)

- ❖ Por el número de nodos hay **solapamiento**, y por ende, **competencia**, buses/SAN, SAN/LAN y LAN/WAN
- ❖ Distintos ámbitos => diferencias de nomenclatura:
 - WAN: ámbito de telecomunicaciones e Internet
 - LAN: ámbito de empresa, trabajo en grupo
 - SAN: ámbito de supercomputación, almacenamiento
- ❖ ***Internetworking***: conexión de varias redes =>
 - Necesidad de estándares de comunicación para convertir información de una clase de red a otra. Ej.: **Internet**

2. Una red elemental



❖ Dos computadores

- Una conexión unidireccional en cada sentido
- Una cola (FIFO) en cada computador

Comunicación en la red

❖ Comunicación elemental:

- Un computador solicita un dato enviando la **dirección** que quiere leer
- El otro le envía el **dato**

❖ Formato elemental

- Campo (1 bit) para distinguir si lo que se envía es un dato o una dirección: **encabezamiento** (*header*)
- Campo que contiene la dirección o el dato: *payload* (“carga útil”)

Problemas en un caso más realista

- ❖ Si la red conecta **más de dos computadores**
 - Añadir al formato un campo para **determinar el nodo destino**
- ❖ Si en el nodo destino hay **varios procesos en ejecución**
 - Es preciso algún modo de **distinguir entre procesos** (por ejemplo ampliar el encabezamiento)
- ❖ El **mensaje puede deteriorarse** en el camino
 - Necesario un **campo de detección de error (C.D.E.)**
- ❖ El mensaje **puede perderse**
 - Si transcurrido un **tiempo** no se recibe respuesta. El mensaje se da por perdido. La solicitud **se reenvía**

Pasos software del emisor de un mensaje en nuestra red elemental

- ❖ La aplicación **copia** los datos por enviar a un *buffer* del sistema operativo
- ❖ El SO calcula el **CDE**, construye el mensaje e inicia el **temporizador**
- ❖ El SO coloca el mensaje en el **interfaz hardware** de red y le ordena que lo envíe a través de la red de interconexión
- ❖ ¿Se recibe señal de **reconocimiento** antes de que transcurra el tiempo establecido?
 - **Sí: se libera** el espacio en el *buffer* ocupado por la copia del mensaje
 - **No (*time-out*): se reenvía** el mensaje y se reinicia el temporizador

Pasos software del receptor

- ❖ El proceso es el **inverso**, es decir, los pasos son:
 - El SO copia los datos del **interfaz hardware** de red en el *buffer* del SO
 - El SO **calcula el CDE** en función de esos datos
 - **¿CDE calculado coincide con el recibido?**
 - ✓ **No: se borra** el mensaje recibido (el emisor lo reenviará transcurrido un tiempo)
 - ✓ **Sí:**
 - El receptor envía un **mensaje de reconocimiento** al emisor
 - El SO **copia los datos** en el espacio de direcciones del usuario y deja a la **aplicación** que **continúe** su ejecución

3. Concepto de protocolo

- ❖ **Protocolo: conjunto de reglas que gobiernan el intercambio de información a través de la red**
 - Estas reglas son de tres tipos:
 - ✓ **Sintácticas** (formatos de datos, niveles de señal...)
 - ✓ **Semánticas** (significado de los campos de control que aparecen en los mensajes)
 - ✓ **De temporización** (sincronización y secuenciación)

Concepto de protocolo (cont.)

- ❖ El protocolo debe **garantizar** un **buen funcionamiento** de la comunicación => debe resolver cuestiones como:
 - **Fiabilidad** (mensaje recibido = mensaje enviado)
 - Respetar **orden bytes** en cada nodo (little/big endian)
 - Conocer el **orden temporal** en que se enviaron los mensajes recibidos en un nodo
 - Mantener un funcionamiento correcto aunque el *buffer* del SO se llene
 - **Etc.**
- ❖ Un protocolo se materializa con el **algoritmo** que el **software** sigue para la **comunicación** de los nodos

4. Medidas de rendimiento de una red..

❖ Ancho de banda (*bandwidth*)

- **Máxima velocidad de propagación** de información por la red de interconexión
 - ✓ Se incluyen todos los bits del mensaje (no sólo el *payload*)

❖ Tiempo de recorrido (*time of flight*)

- Tiempo desde que el **primer bit** del mensaje es enviado por la red hasta que llega al receptor

Medidas de rendimiento de una red..

❖ Tiempo de transmisión

- **Tiempo** que tarda el mensaje **en pasar por la red** (sin incluir el tiempo de recorrido), es decir:
 - ✓ Tiempo desde que llega al receptor el primer bit del mensaje hasta que llega el último
- **Si no hay otros mensajes** con los que compartir la red de interconexión, el **tiempo de transmisión** es **$\text{tamaño del mensaje} / \text{ancho de banda}$**

Medidas de rendimiento de una red..

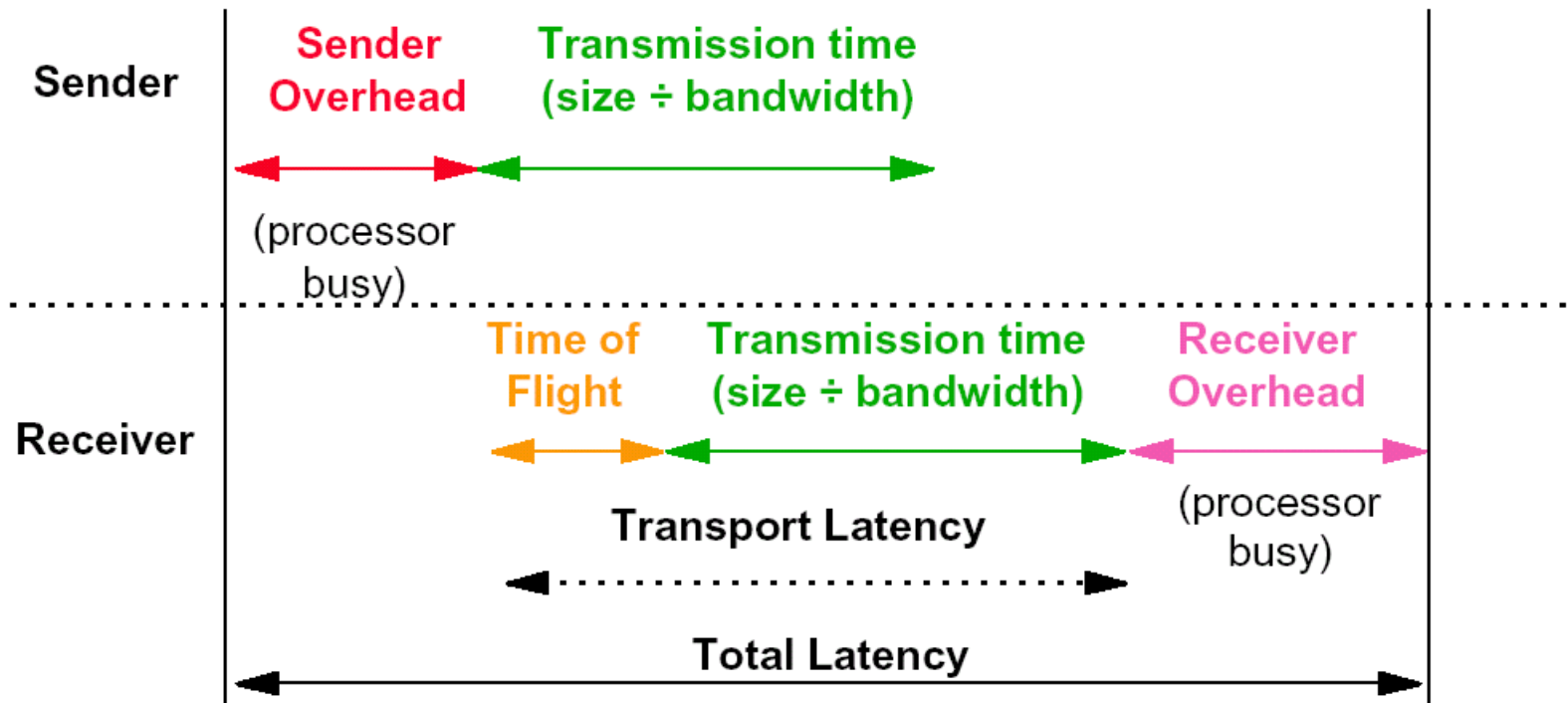
❖ Latencia de transporte

- Tiempo que el mensaje está en la red, es decir,
tiempo de recorrido + tiempo de transmisión

❖ Tiempo gastado por el emisor/receptor (*sender/receiver overhead*)

- Tiempo que tarda el (hardware y software del) emisor/receptor en **colocar** el mensaje en la red/ **recoger el mensaje** de la red
 - ✓ El procesador está **ocupado** durante este tiempo
 - ✓ En general, *receiver overhead* > *sender overhead*
 - Por ejemplo, el receptor debe pagar el coste de una interrupción

Medidas de rendimiento de una red: latencia total



$$\text{Total Latency} = \text{Sender Overhead} + \text{Time of Flight} + \text{Message Size} \div \text{BW} + \text{Receiver Overhead}$$

Ejemplo de la latencia total en redes SAN, LAN y WAN

- ❖ Supongamos: *sender overhead* = 80 μs ; *receiver overhead* = 100 μs ; tamaño mensaje = 10^4 B; ancho de banda = 1000 Mb/s; velocidad de propagación de las señales = $2 \cdot 10^5$ km/s
 - Tiempo transmisión = $8 \cdot 10^{-2} \text{ Mb} / 10^{-3} \text{ Mb}/\mu\text{s} = 80 \mu\text{s}$
- ❖ Distancia nodos: SAN: 10 m; LAN: 500 m; WAN: 10^3 km
- ❖ **Latencia total:**
 - **SAN:** $80 \mu\text{s} + 10 \text{ m} / (2 \cdot 10^8 \text{ m/s}) + 80 \mu\text{s} + 100 \mu\text{s} \cong 260 \mu\text{s}$
 - **LAN:** $260 \mu\text{s} + 500 \text{ m} / (200 \text{ m}/\mu\text{s}) = 262,5 \mu\text{s}$
 - **WAN:** $260 \mu\text{s} + 1000 \text{ km} / (0,2 \text{ km}/\mu\text{s}) = 5260 \mu\text{s}$

Otros aspectos relacionados con la latencia en redes

- ❖ Los **protocolos** son **más complejos** conforme consideramos redes con **mayor distancia** entre nodos.
Razones:
 - Mayor **probabilidad** de **errores** de transmisión
 - El tiempo de recorrido aumenta => los **tiempos** gastado por el **emisor** y el **receptor pierden importancia** => pueden ser mayores
- ❖ El *sender overhead* puede **solaparse** con la latencia de transporte y el *receiver overhead*
 - Salvo si el emisor necesita una respuesta antes de enviar el siguiente mensaje

Ancho de banda efectivo

❖ Por definición, **ancho de banda efectivo** =
tamaño del mensaje / latencia total

❖ Si definimos *overhead* =
sender overhead + time of flight + receiver overhead
resulta **Latencia total** =

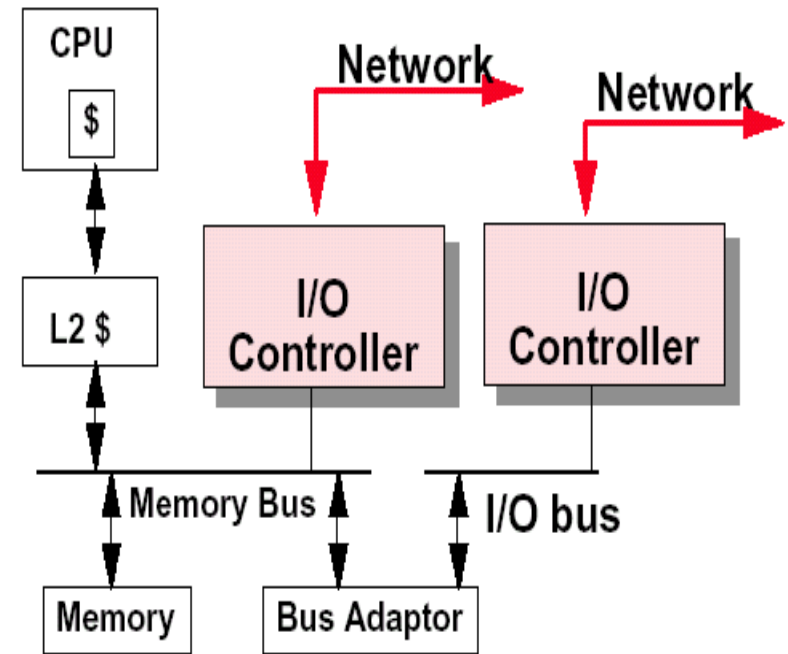
overhead + tamaño mensaje / ancho de banda

- Para **amortizar** el coste de un *overhead* alto se necesitan **tamaños de mensaje grandes**

5. Conexión computador-red..

❖ ¿Conectar la red al bus de memoria o al de E/S?

- Conectar al **bus de memoria** si se busca un **alto rendimiento**
 - ✓ Sólo **algunas SAN** se conectan al bus de memoria
- **Bus E/S** permite una **conexión más sencilla**
 - ✓ Las **WAN, LAN** y casi todas las **SAN** se conectan al bus de E/S



Conexión computador-red

❖ ¿Cómo debe ser la **E/S a la red: programada, DMA o interrupciones?**

■ Para **enviar mensajes**

- ✓ Si los mensajes son **largos**, mejor **DMA**
- ✓ En caso contrario, mejor salida programada

■ Para **recibir mensajes**

- ✓ Es mejor usar **interrupciones** si se reciben pocos por unidad de tiempo
- ✓ En caso contrario, mejor entrada por sondeo (*polling*)

6. Conexión de más de dos computadores

❖ Medio interconexión:

■ Compartido

✓ Bajo coste y limitado rendimiento

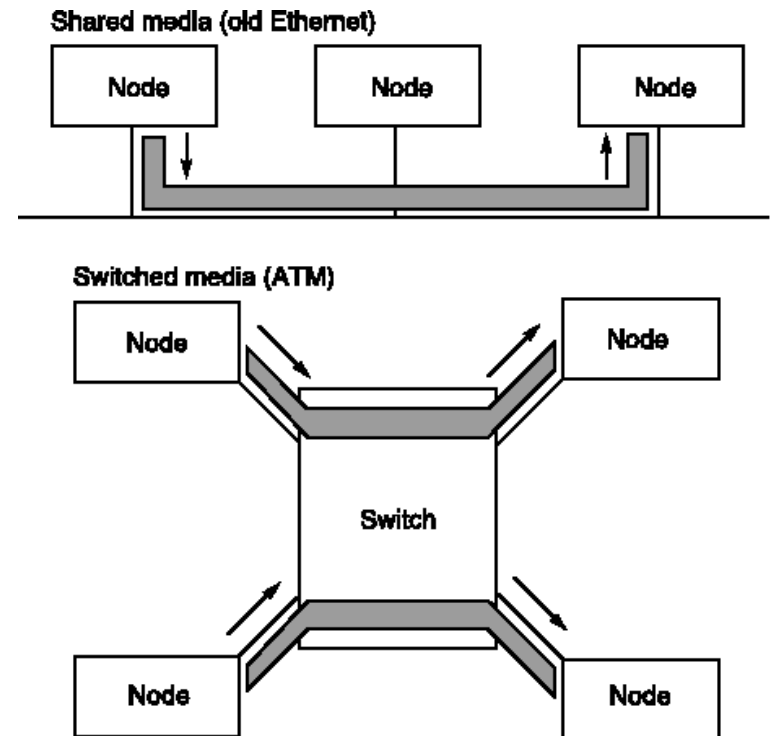
✓ Arbitraje necesario

■ Conmutado

✓ Hay una **línea dedicada** entre el nodo y un *switch*

- Del *switch* parten líneas dedicadas a los otros nodos

✓ No se precisa arbitraje



Arbitraje con un medio de interconexión compartido

- ❖ Arbitro **centralizado**: sólo en redes pequeñas
- ❖ Arbitraje **distribuido**
 - Por **detección de colisiones** (*carrier sensing*)
 - ✓ El rendimiento baja cuando hay un uso intensivo de la red
 - Mediante un vale (*token*) que se pasa de un nodo a otro y da derecho a acceder a la red

Medio de interconexión conmutado..

- ❖ Los conmutadores permiten **comunicación** directa (**punto a punto**) entre **varios pares de nodos a la vez**
 - **Mayor ancho de banda** que una red con un medio de interconexión compartido
 - **Interfaz** del nodo con la red **más sencillo**
 - Se consume **tiempo en los *switches*** pero se **ahorra** el tiempo del **arbitraje**
 - Buena **escalabilidad**

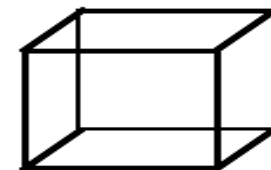
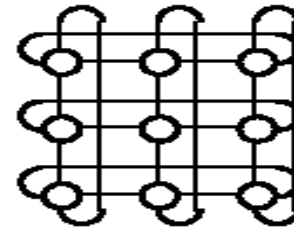
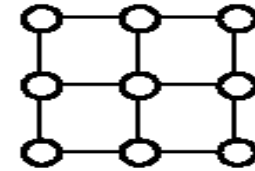
Medio de interconexión conmutado

- ❖ **Conmutadores** que eran tan grandes como un computador hoy pueden integrarse en un único chip =>
 - **Están reemplazando a los buses y las redes con un medio de interconexión compartido**
- ❖ Otros **nombres** para las redes *conmutadas*: *data switching exchanges, multistage interconnection networks, interface message processors (IMPs)*

Topologías de los conmutadores: clasificación..

❖ Estáticas (conmutador distribuido)

- Las **rutas** entre dispositivos son **fijas**
- Hay un **pequeño conmutador en cada nodo**
- Ejemplos: malla, anillo, toro, hipercubo

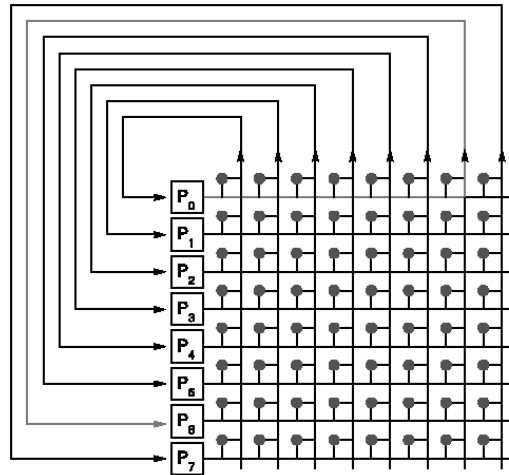


Topologías de los conmutadores: clasificación

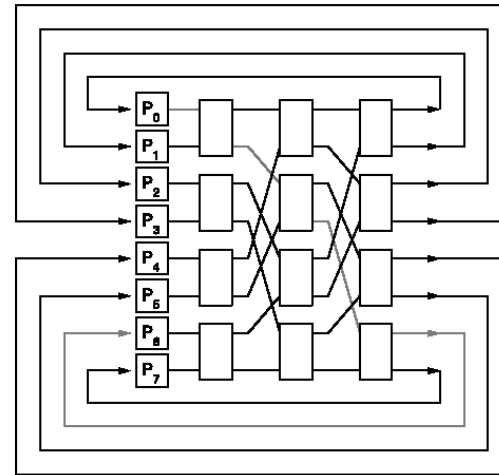
❖ **Dinámicas** **(conmutador centralizado)**

- Las **rutas** entre dispositivos son **reconfigurables** mediante el control de los **elementos de conmutación** de la red
- Ejemplos: barras cruzadas, omega, árbol grueso (*fat tree*)

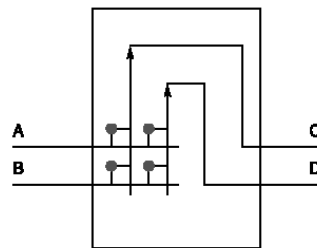
Red de barras cruzadas y omega



a. Crossbar

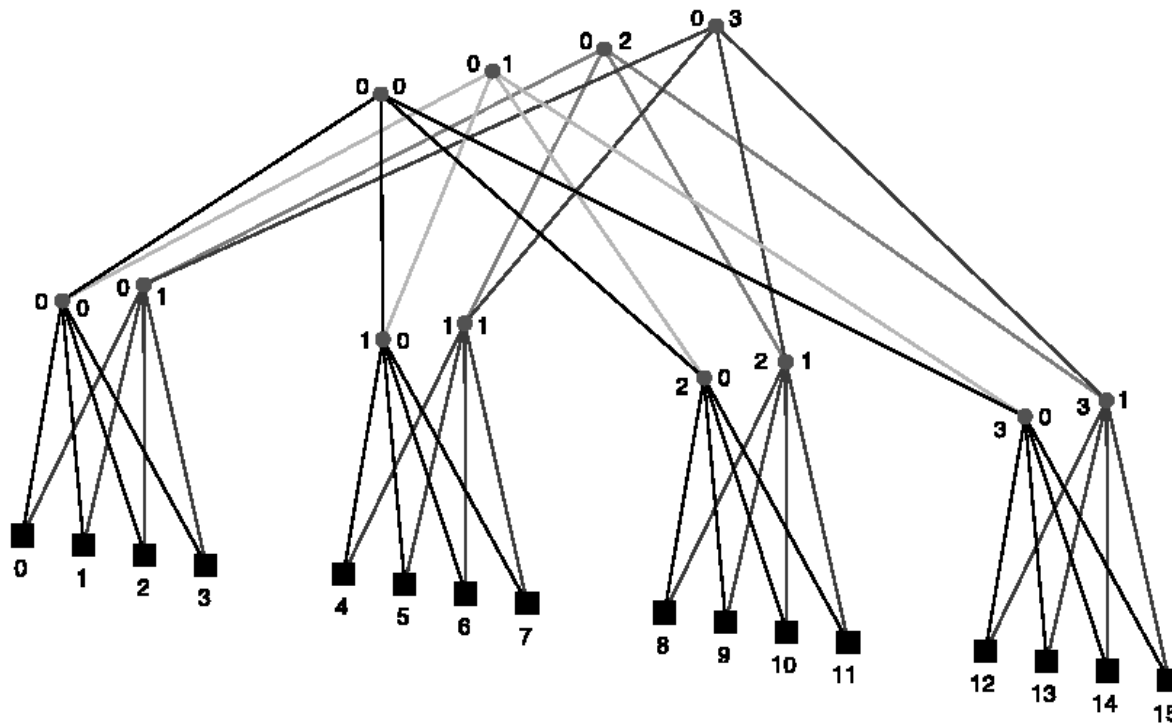


b. Omega network



c. Omega network switch box

Ejemplo de árbol grueso



Principales características de los conmutadores..

❖ Con/sin bloqueo

- En una red **con** bloqueo la **conexión simultánea de varios pares de nodos (disjuntos)** no es siempre posible
 - ✓ Ejemplo, red **omega** (las conexiones P0-P6 y P1-P7 son imposibles a la vez)
- Ejemplo red **sin** bloqueo: barras **cruzadas**
 - ✓ No obstante hay conflicto si dos nodos quieren conectarse a la vez con un mismo destino

Principales características de los conmutadores

- ❖ **Grado:** número de enlaces de un nodo
- ❖ **Diámetro:** máximo número enlaces en un camino mínimo entre dos nodos
- ❖ **Distancia media** entre nodos
- ❖ **Bisección:** mínimo número de enlaces que atraviesa una división imaginaria de la red en dos partes con igual número de nodos (± 1)
 - **Ancho de banda de la bisección:** la suma de los anchos de banda de esos enlaces

Conmutadores distribuidos y sus características

N nodos	Indicadores de coste		Indicadores de rendimiento		
	Grado	Nº. enlaces	Diámetro	Dist. media	Bisección
Anillo	2	N	$N/2$	$N/4$	2
Malla 2D	≤ 4	$2(N - N^{1/2})$	$2*(N^{1/2} - 1)$	$2*N^{1/2}/3$	$N^{1/2}$
Toro 2D	4	$2*N$	$N^{1/2}$	$N^{1/2}/2$	$2*N^{1/2}$
Hipercubo nD	$n = \log_2 N$	$n*N/2$	n	$n/2$	$N/2$

Red de barras cruzadas vs. red multietapa (ej., red omega)

- ❖ **Mayor ancho de banda** por nodo
- ❖ **Latencia** (mínima para transferir un dato)
 - **Constante** en red de **barras cruzadas**
 - **$O(\log N)$** en **red multietapa**
- ❖ **Coste:** proporcional al número de elementos de conmutación
 - **N^2** en topología de **barras cruzadas** =>
gía de **barras cruzadas** =>
e en **redes pequeñas**
en **red omega**

redes mayores pero con **N** grande el coste es elevado

Conmutación de circuitos y conmutación de paquetes

❖ Conmutación de circuitos

- **Antes de iniciarse la transmisión se establece un camino físico** entre el nodo fuente y el destino

❖ Conmutación de paquetes

- **Motivación:** no tener reservado un **camino** que puede estar **desocupado** la mayor parte del tiempo
- La información se divide en **paquetes o frames**
- Cada **paquete** incluye una **información de control** que permite que siga un **camino**, a través de los **nodos de conmutación de la red**, que termine en el nodo **destino**

Técnicas de conmutación de paquetes: datagrama y circuito virtual

- ❖ Función de los nodos de conmutación de la red
 - **Almacenan el paquete y lo colocan en cola sobre una línea de salida**
- ❖ **Datagramas**
 - Cada **nodo** de conmutación debe **decidir el mejor camino** a seguir por cada paquete (datagrama) que le llega
 - Los **paquetes** pueden ser **recibidos** por el nodo destino **en orden diferente** al de emisión
 - ✓ El software del nodo destino debe ordenarlos
- ❖ **Circuito virtual**
 - Se establece un **camino** que es **seguido por todos los paquetes** del mensaje

Comparación datagrama y circuito virtual..

❖ **Ventajas circuito virtual**

- **Los paquetes llegan en orden**
- **Mejor control de errores**
 - ✓ Si un paquete no llega a un nodo del camino, éste puede solicitar su **retransmisión** al nodo anterior
- **Los paquetes viajan más rápido** porque no hay que tomar una decisión de encaminamiento por cada paquete que llega a un nodo

Comparación datagrama y circuito virtual

❖ Ventajas datagramas

- No existe una fase de **establecimiento de llamada** (la fase en que se establece el circuito virtual)
- **Más flexibilidad**
 - ✓ Si hay **congestión** en una zona de la red, los paquetes pueden seguir **caminos alejados de la zona** de congestión
- **Más seguridad**
 - ✓ Si un **nodo falla** los paquetes pueden encontrar un **camino que no pase por él**
 - En el caso de circuito virtual todos los caminos que atraviesan ese nodo fallarán

Elementos de conexión de redes

❖ *Bridges* (puentes)

- **Conectan varias LAN**
- Operan a nivel del protocolo Ethernet
- **Más sencillos y baratos** que los *routers*

❖ *Routers* (encaminadores)

- **Dispositivos con capacidad de procesamiento que:**
 - ✓ **Conectan varias redes similares o diferentes** (p. ej., varias LAN, dos WAN, una LAN y una WAN)
 - ✓ **Resuelven diferencias de direccionamiento** en las redes **para que la información pase de una red a otra siguiendo el camino adecuado para alcanzar el destino**
 - ✓ Operan al nivel del **protocolo Internet (IP)**
 - ✓ Generalmente, son **más lentos que los puentes**

7. Arquitecturas de protocolo

- ❖ **Transferencia información por la red:** en última instancia es una **transferencia entre aplicaciones**
- ❖ El programador de aplicaciones **no** debería **preocuparse por los detalles** de la red
 - Esto se consigue organizando en **capas** las **funciones de comunicación** a través de una red
 - El modo en que se organiza en capas un protocolo se denomina **arquitectura del protocolo**

Un modelo de tres capas..

❖ Comunicaciones → **tres agentes**:

- **Aplicaciones** (ej.: transferencia ficheros)
- **Computadores y redes**

❖ Un modelo sencillo: organizar la transferencia de información en **tres capas**

- Capa de aplicación
- Capa de transporte
- Capa de acceso a la red

Un modelo de tres capas..

❖ Capa de aplicación

- Contiene un módulo distinto para cada tipo de aplicación (p. ej., un módulo para transferir ficheros)

❖ Capa de transporte

- Se encarga de que el intercambio de datos sea **seguro**
- Es una capa compartida por todas las aplicaciones

Un modelo de tres capas..

❖ Capa de acceso a la red

- Se encarga del **intercambio de datos entre el computador y la red**
- Las características de esta etapa dependen del tipo de red
- Permite que el software que está por encima de esta capa funcione independientemente de las características concretas de la red

Un modelo de tres capas

- ❖ Cada computador tiene una **dirección de red**
- ❖ Cada **aplicación** en el computador tiene su propia **dirección** denominada **SAP** (*Service Access Point*)
- ❖ Ejemplo de comunicación:
 - La aplicación asociada al **SAP1** en el computador **X** quiere **transmitir un mensaje a** la aplicación asociada al **SAP2** del computador **Y**
 - Cada capa recoge la **información de la capa superior** y le añade **información de control**
 - ✓ El resultado se denomina **PDU** (*Protocol Data Unit*)

Ejemplo de comunicación en el modelo de tres capas..

- La **aplicación** en X pasa el mensaje a la capa de transporte con información de control para que lo envíe al SAP2 de Y
- La **capa de transporte en X**:
 - ✓ Posiblemente, fracciona el mensaje en unidades más pequeñas
 - ✓ **Añade** información de **control** a cada unidad (crea PDU de transporte). Por **ejemplo**:
 - **SAP destino, código de detección de error**

Ejemplo de comunicación en el modelo de tres capas..

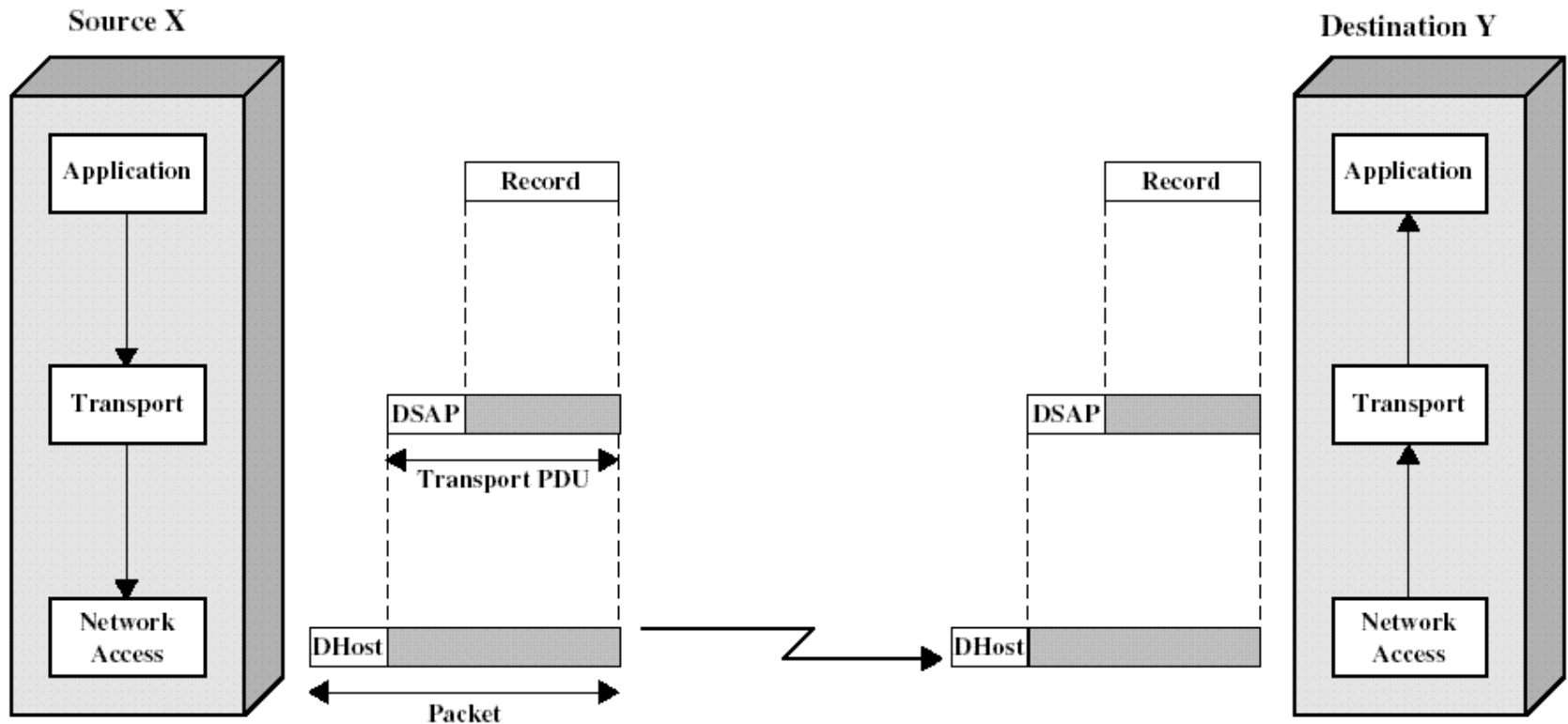
■ La **capa de acceso a la red en X**

- ✓ Recibe PDU de transporte
- ✓ Le añade una cabecera con información de control (crea PDU de red). Por ejemplo:
 - Dirección del computador destino
- ✓ Envía a la red la PDU de acceso a la red
- ✓ (Obsérvese que esta capa no necesita conocer la dirección del SAP destino)

Ejemplo de comunicación en el modelo de tres capas

- La **capa de acceso a la red en Y**:
 - ✓ Recibe la PDU de red, le **elimina** su cabecera
 - Lo que queda es una PDU de transporte que es enviada a la capa de transporte
- La **capa de transporte en Y**:
 - ✓ Examina el código de detección de error y el SAP destino
 - ✓ **Elimina la información de control** para obtener el mensaje que debe enviarse al SAP destino
 - ✓ Si no ha habido error, **envía el mensaje a la aplicación** asociada al SAP2 (el SAP destino)

Esquema de comunicación en el modelo de tres capas



Arquitectura de protocolos TCP/IP

- ❖ *Transmission Control Protocol / Internet Protocol*
- ❖ Permite que computadores de **redes diferentes** se comuniquen de modo **fiable y eficiente**
- ❖ No hay un modelo oficial TCP/IP pero se pueden distinguir **cinco capas: aplicación, transporte, internet, acceso a la red y física**
 - El uso estricto de todas las capas no es obligatorio
 - ✓ Por ejemplo, hay protocolos de aplicación que operan directamente en la capa internet

Capas protocolos TCP/IP..

❖ Aplicación

- Entre los **protocolos** de esta capa mencionaremos:
 - ✓ **HTTP: (*Hyper Text Transport Protocol*)**: Protocolo que permite transmitir páginas web
 - **Página web = documento hipermedia** (texto + imágenes + enlaces +...) escrito en **HTML (*Hyper Text Markup Language*)**
 - El **conjunto de servidores** que emplean el protocolo HTTP es lo que se denomina **World Wide Web (WWW)**
 - ✓ **SMTP (*Simple Mail Transfer Protocol*)**: recurso básico de correo electrónico
 - ✓ **FTP (*File Transfer Protocol*)**: para transferencia de archivos
 - ✓ **Telnet**: permite conectarse a un computador remoto y ejecutar programas y comandos en él

Capas protocolos TCP/IP..

❖ Transporte

- Suele seguir el protocolo **TCP**

❖ Internet

- Protocolo **IP**
- Ofrece servicios de **encaminamiento a través de varias redes**
 - ✓ Se implementa **en los sistemas finales y en los encaminadores (*routers*)**

Capas protocolos TCP/IP

❖ De acceso a la red

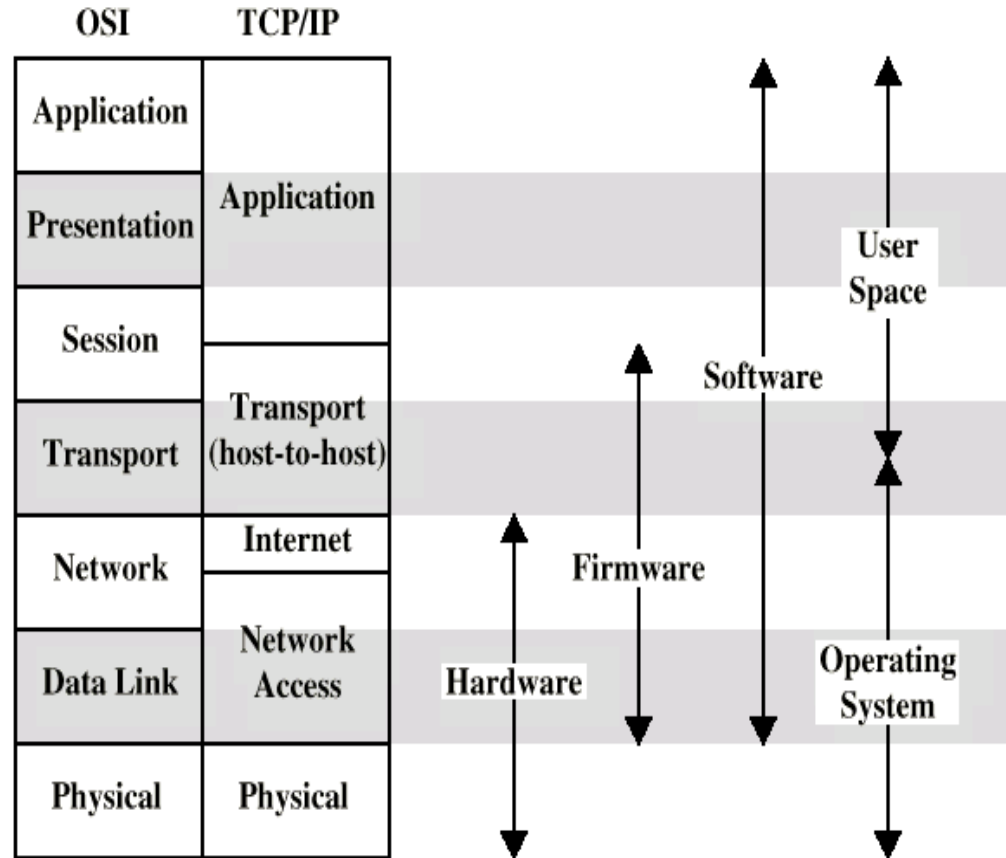
- Responsable del **intercambio de datos entre el sistema final y la red**

❖ Física

- **Interfaz física** entre los **nodos** y la **red** de interconexión
 - ✓ Características del medio de transmisión, naturaleza de las señales, velocidad de los datos...

Modelo OSI

- ❖ *Open Systems Interconnect*
- ❖ Pretende ser el marco de referencia para el desarrollo de protocolos estándares



Bibliografía

- ❖ Patterson D. A. and J. L. Hennessy [2002]: *Organización y diseño de computadores, La interfaz hardware/software*. McGraw-Hill. Madrid
- ❖ Stallings, W. [1997]: *Comunicaciones y Redes de Computadores*. 5ª edición. Madrid: Prentice Hall Iberia.