

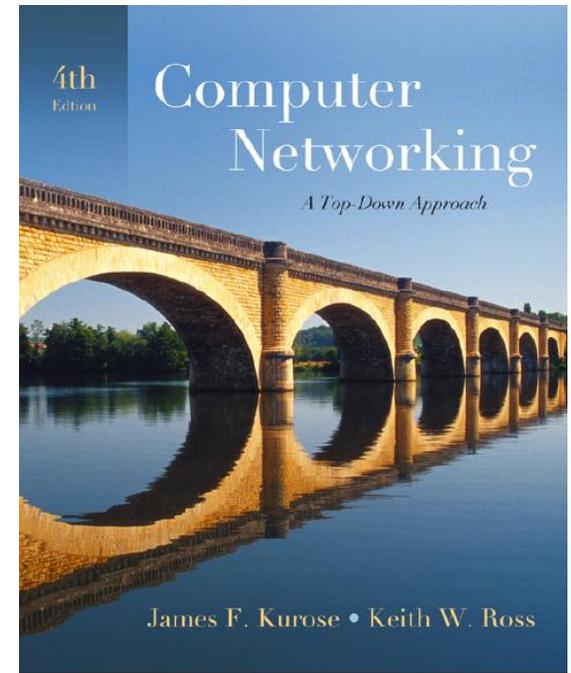
# Introducción a las Redes de Computadoras

## Capítulo 4 Capa de Red

Nota acerca de las transparencias del curso:

Estas transparencias están basadas en el sitio web que acompaña el libro, y han sido modificadas por los docentes del curso.

All material copyright 1996-2007  
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer Networking:  
A Top Down Approach ,  
4<sup>th</sup> edition.*

*Jim Kurose, Keith Ross  
Addison-Wesley, July  
2007.*

# Cap. 4: Capa de red

## Objetivos del capítulo:

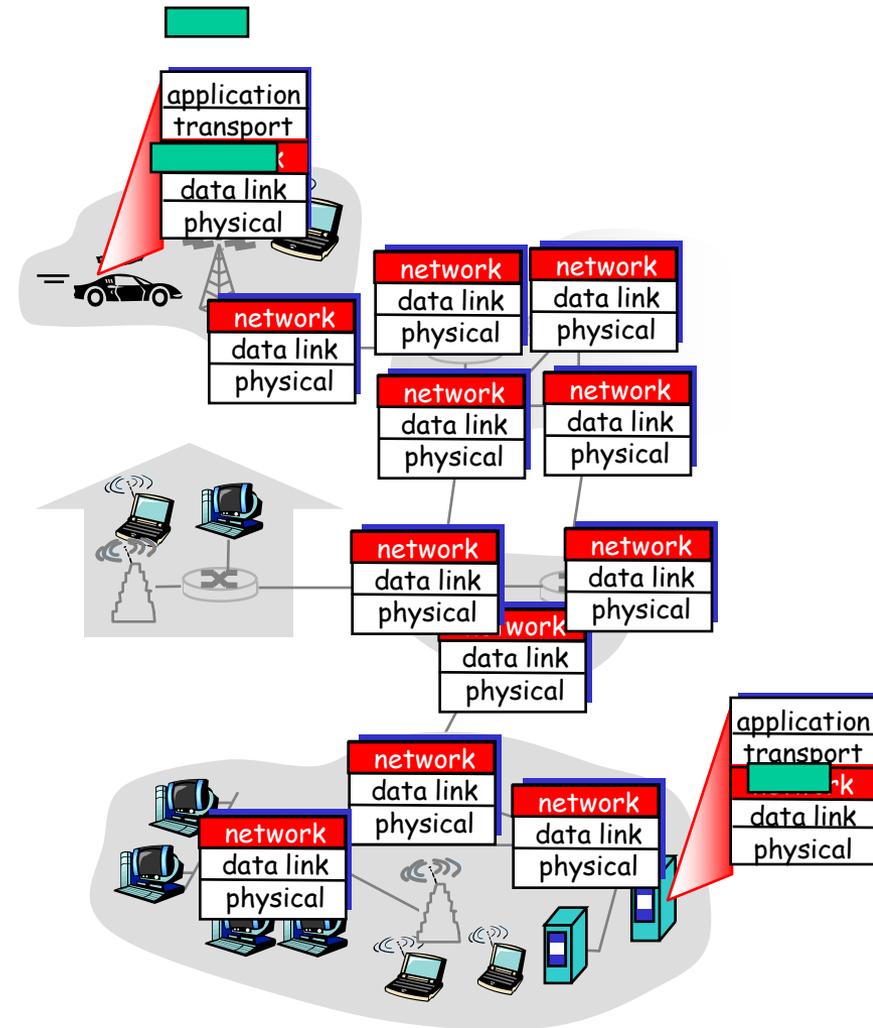
- ❑ comprender los principios que sustentan los servicios de capa de red:
  - modelos de servicios de capa de red
  - forwarding (reenvío) vs. routing (enrutamiento)
  - como funciona un router
  - routing (selección del camino o path selection)
  - escalabilidad
  - tópicos avanzados: IPv6, movilidad
- ❑ implementación en la Internet

# Cap. 4: Capa de red

- ❑ 4.1 Introducción
- ❑ 4.2 circuitos virtuales y datagramas
- ❑ 4.3 dentro de un router
- ❑ 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- ❑ 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- ❑ 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- ❑ 4.7 Broadcast y multicast

# Capa de Red

- transporte de segmentos desde host emisor a receptor
- el emisor encapsula segmentos en datagramas
- el receptor entrega segmentos a la capa de transporte
- Protocolos de capa de red *deben estar presentes* en cada host y router
- Los routers examinan el cabezal de todos los datagramas IP que reciben



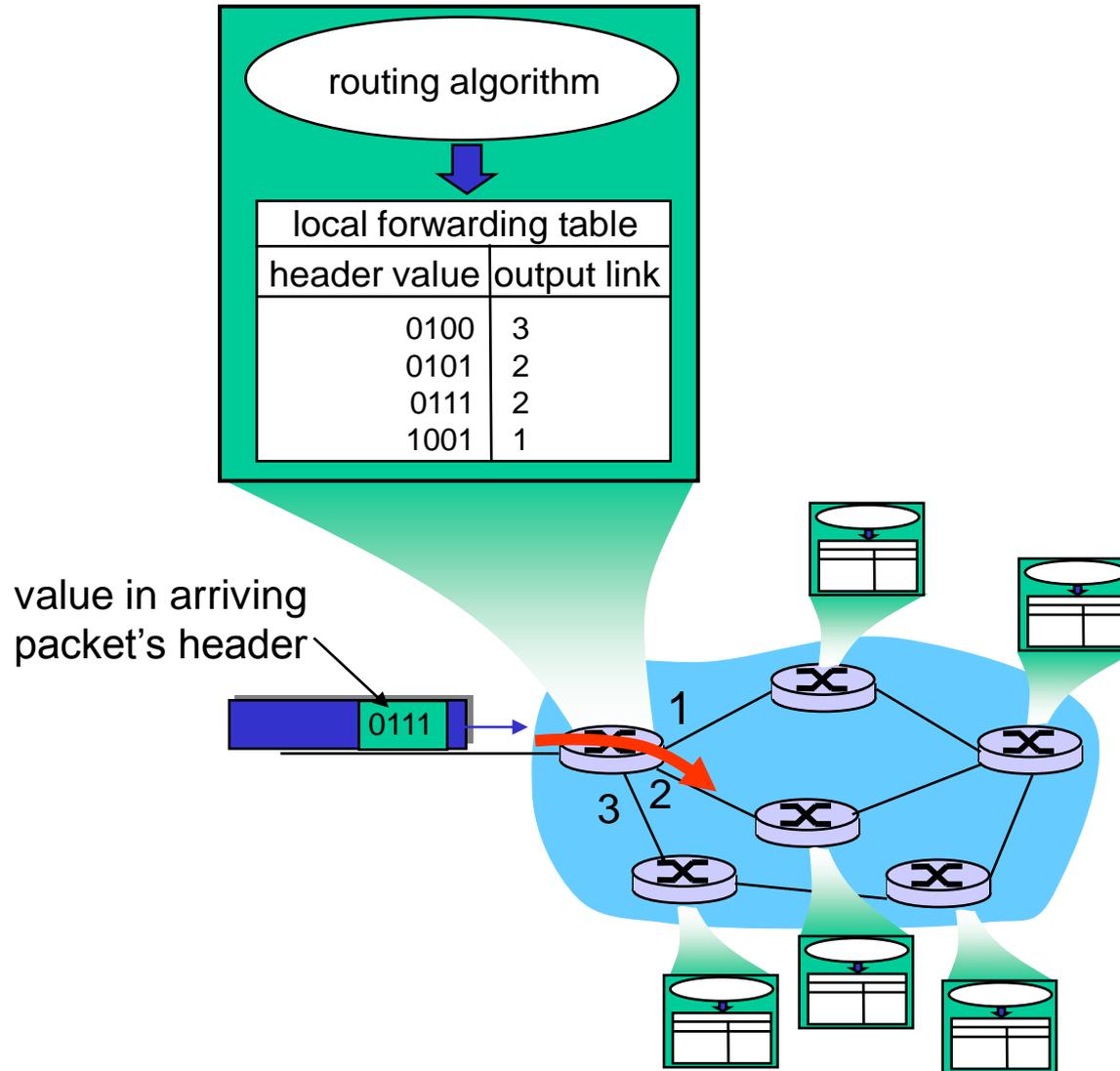
# Dos funciones clave de la Capa de Red

- ❑ *forwarding*: mover paquetes entre puertos de entrada y salida del router
- ❑ *enrutamiento*: determinar la ruta de los paquetes desde origen a destino
  - *algoritmos de enrutamiento*

## analogía:

- ❑ *enrutamiento*: proceso de planificación del viaje (p. ej. en avión) desde la salida a la llegada
- ❑ *forwarding*: proceso de pasar por cada punto de intercambio (p. ej. cambio de puertas en aeropuerto)

# Interacción entre routing & forwarding



# Establecimiento de la conexión

- ❑ 3a funcionalidad importante en *algunas* tecnologías de red:
  - ATM, frame relay, X.25
- ❑ antes de iniciar el flujo de datagramas, los extremos (hosts) y los routers intervinientes deben establecer una conexión virtual
  - routers están implicados
- ❑ servicio de conexión de capa de red vs. capa de transporte:
  - **red**: entre hosts (involucra routers cuando se establecen VCs)
  - **transporte**: entre dos procesos

# Modelo de Servicio de Red

**P:** Cual es el *modelo de servicio* del "canal" que transporta los datagramas del trasmisor al receptor?

## Ejemplo de servicios para datagramas individuales:

- ❑ entrega garantizada
- ❑ entrega garantizada con un retardo menos a 40 miliseg.

## Ejemplo de servicios para un flujo de datagramas:

- ❑ entrega de datagramas en orden
- ❑ ancho de banda mínimo garantizado
- ❑ restricciones en la variación del retardo (*jitter*)

# Modelos de Servicios de Red:

Network Architecture	Service Model	Guarantees ?			Congestion feedback	
		Bandwidth	Loss	Order Timing		
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

# Cap. 4: Capa de red

- ❑ 4.1 Introducción
- ❑ 4.2 circuitos virtuales y datagramas
- ❑ 4.3 dentro de un router
- ❑ 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- ❑ 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- ❑ 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- ❑ 4.7 Broadcast y multicast

# Servicios de Capa de Red orientados y no-orientados a conexión

- ❑ datagramas: servicio no-orientado a conexión
- ❑ circuitos virtuales (VC): servicio orientado a conexión
- ❑ análogo a los servicios de capa de transporte, pero:
  - **servicio:** host-a-host
  - **no se puede elegir:** la red provee uno u el otro
  - **implementación:** en el "core" de la red

# Circuitos Virtuales

“el camino de extremo a extremo se comporta como un circuito telefónico”

- tiene en cuenta parámetros de performance
- la red es responsable a lo largo del camino

- ❑ establecimiento de llamada antes del flujo de datos
- ❑ cada paquete tiene un identificador de VC (no la dirección del host de destino)
- ❑ *cada* router en el camino mantiene el estado de cada conexión
- ❑ se pueden *asignar* recursos de routers y enlaces (ancho de banda, buffers) para cada VC (recursos dedicados = servicio predecible)

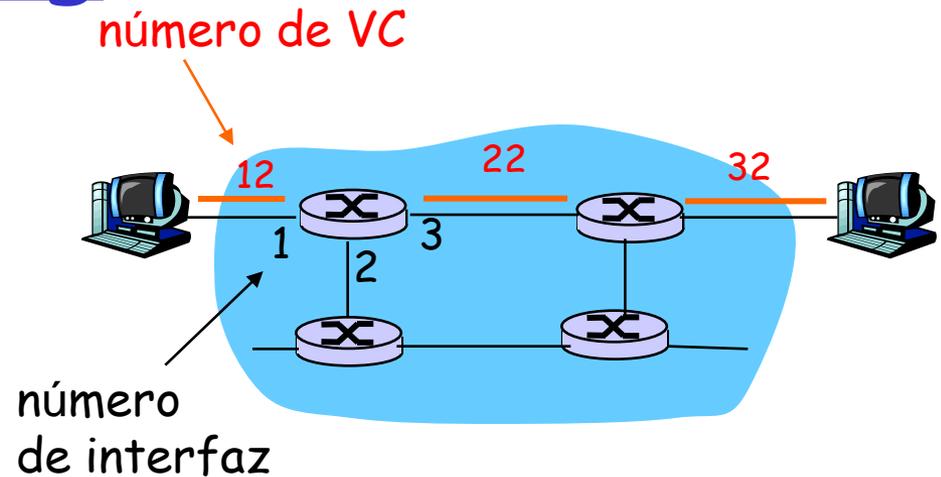
# Implementación de VCs

un VC consiste de:

1. camino (path) de fuente a destino
  2. número de VC, uno por cada enlace a lo largo del camino
  3. entradas en las tablas de forwarding de los routers a lo largo del camino: cross-conexiones
- ❑ los paquetes de un VC usan el número que lo identifica (en lugar de la dirección de destino)
  - ❑ el número del VC puede cambiar en cada enlace.
    - los números de VC tienen alcance local
    - el nuevo número de VC sale de la tabla de forwarding

# Tabla de forwarding

Tabla de forwarding en el router "nor-oeste":

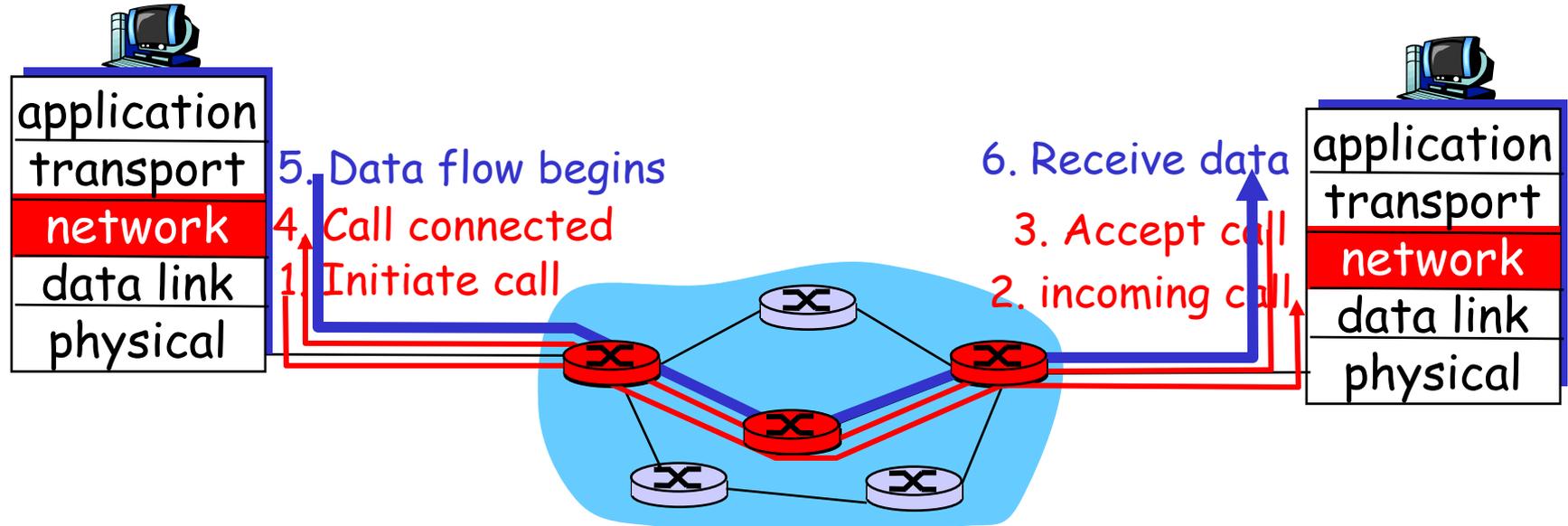


Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Los routers mantienen información de estado de las conexiones!

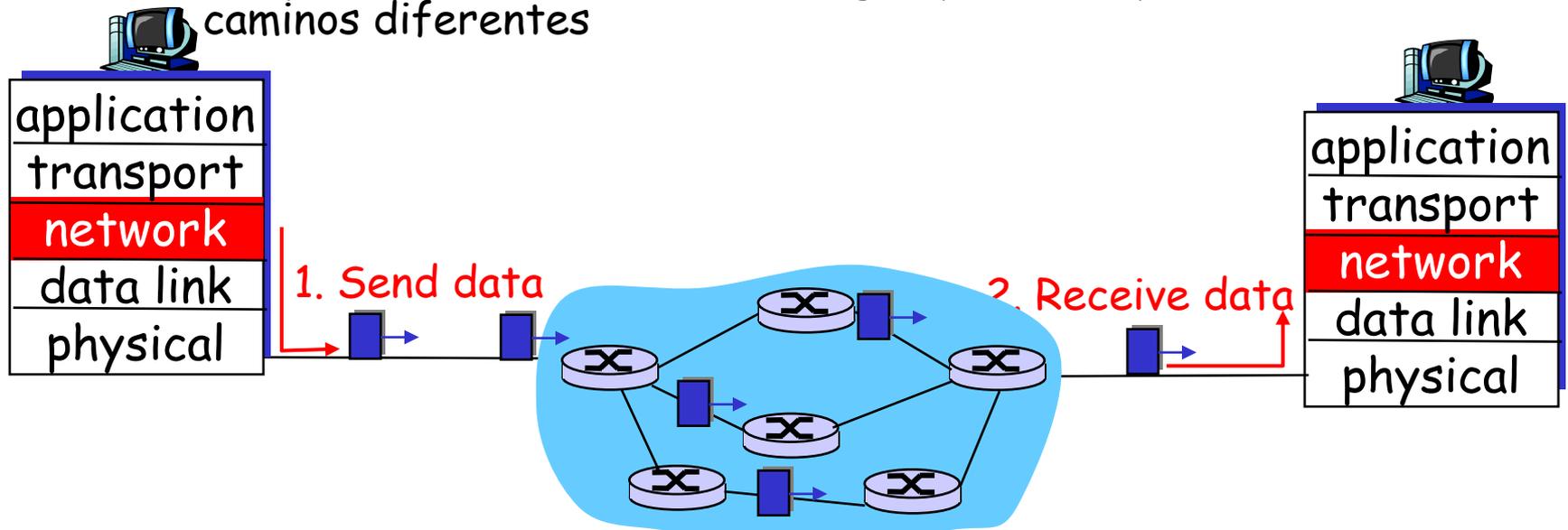
# Circuitos Virtuales: protocolos de señalización

- Usados para establecer, mantener y dar de baja los VCs
- usados en ATM, frame-relay, X.25
- "no se utilizan en Internet"
  - RSVP, MPLS...



# Redes de datagramas

- ❑ no existe procedimiento de establecimiento de conexión en capa de red
- ❑ routers: no mantienen estado de conexiones extremo a extremo
  - no existe en concepto de "conexión" a nivel de red
- ❑ los paquetes son encaminados utilizando la dirección de host destino
  - Los paquetes entre un mismo origen y destino pueden tomar caminos diferentes



# Tabla de forwarding

$2^{32} \sim 4$  mil millones  
de entradas posibles

<u>Destination Address Range</u>	<u>Link Interface</u>
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

# Longest prefix matching

<u>Prefix Match</u>	<u>Link Interface</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

## Examples

DA: 11001000 00010111 00010110 10100001      cuál interfaz?

DA: 11001000 00010111 00011000 10101010      cuál interfaz?

# Datagramas o VCs: por qué?

## Internet (datagramas)

- ❑ intercambio de datos entre computadores
  - servicio "elástico", sin requerimientos estrictos de tiempo
- ❑ end systems "inteligentes" (computadores)
  - adaptables, control de flujo y recuperación ante errores
  - red simple, "borde" complejo
- ❑ muchos tipos de enlaces
  - características diferentes
  - servicio uniforme

## ATM (VC)

- ❑ evolución desde la telefonía
- ❑ conversación entre seres humanos:
  - reqs. de tiempo estrictos, se necesita un servicio confiable
  - y garantizado
- ❑ end systems "tontos"
  - teléfonos
  - complejidad *en* la red

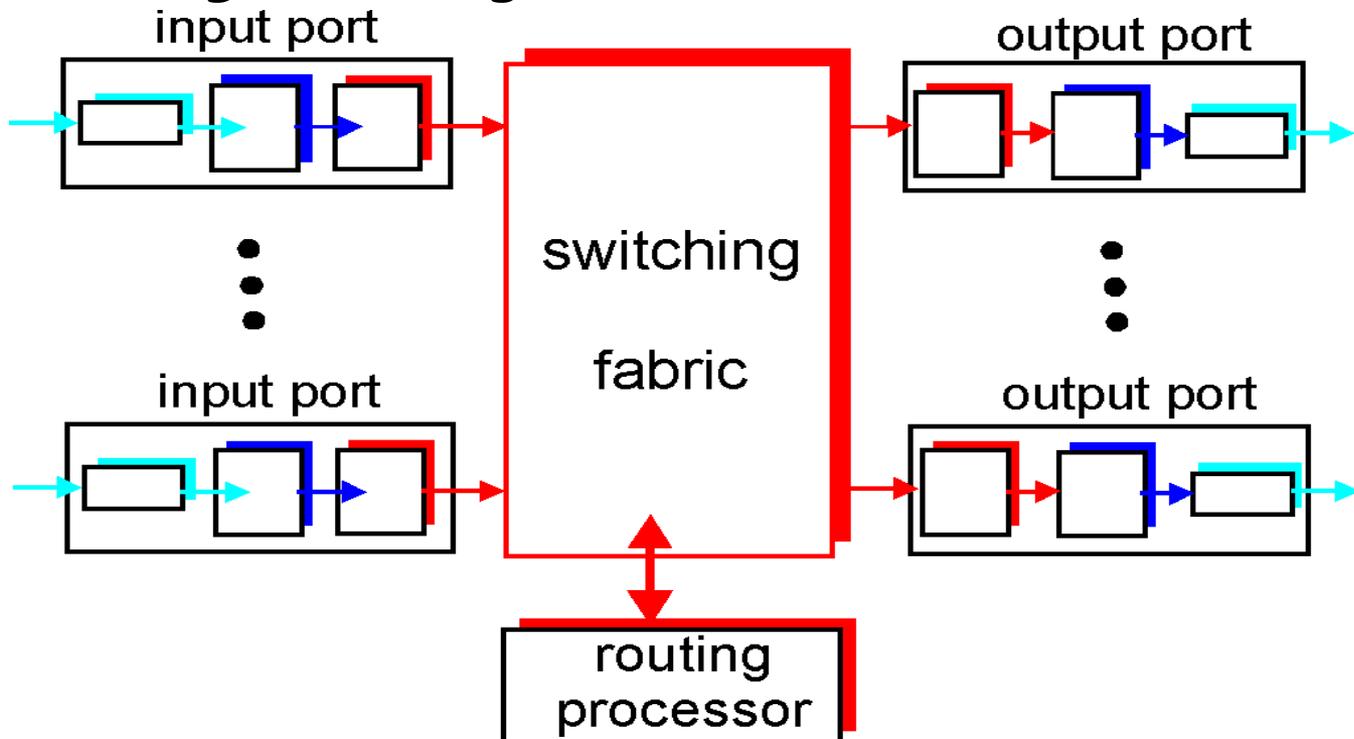
# Cap. 4: Capa de red

- ❑ 4.1 Introducción
- ❑ 4.2 circuitos virtuales y datagramas
- ❑ 4.3 dentro de un router
- ❑ 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- ❑ 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- ❑ 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- ❑ 4.7 Broadcast y multicast

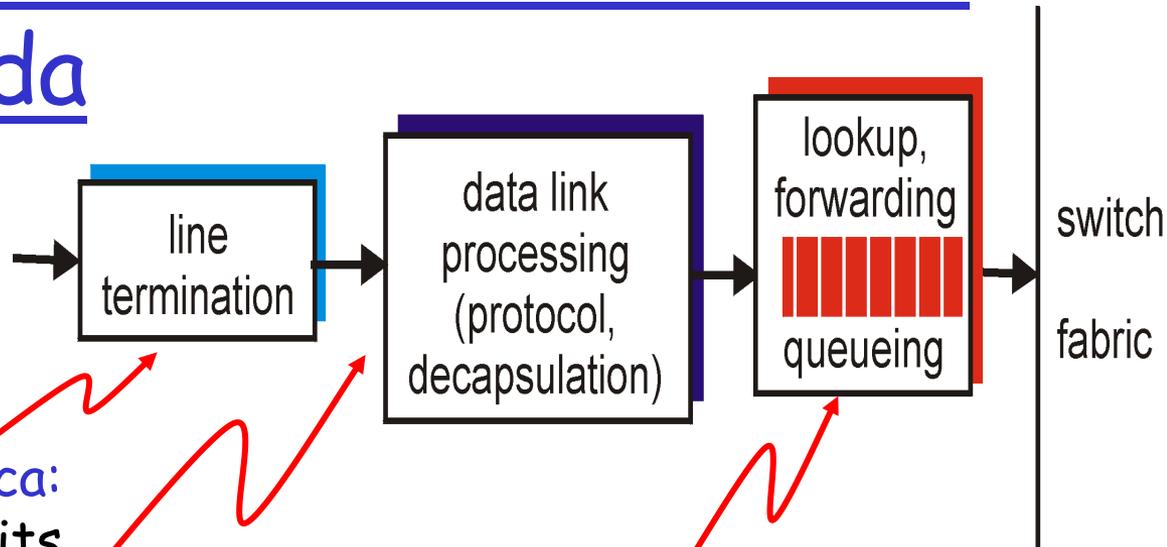
# Arquitectura del Router

Dos funciones fundamentales:

- ❑ ejecutar los algoritmos/protocolos de routing (RIP, OSPF, BGP)
- ❑ *forwarding* de datagramas entre entrada/salida



# Funciones de los Puertos de Entrada



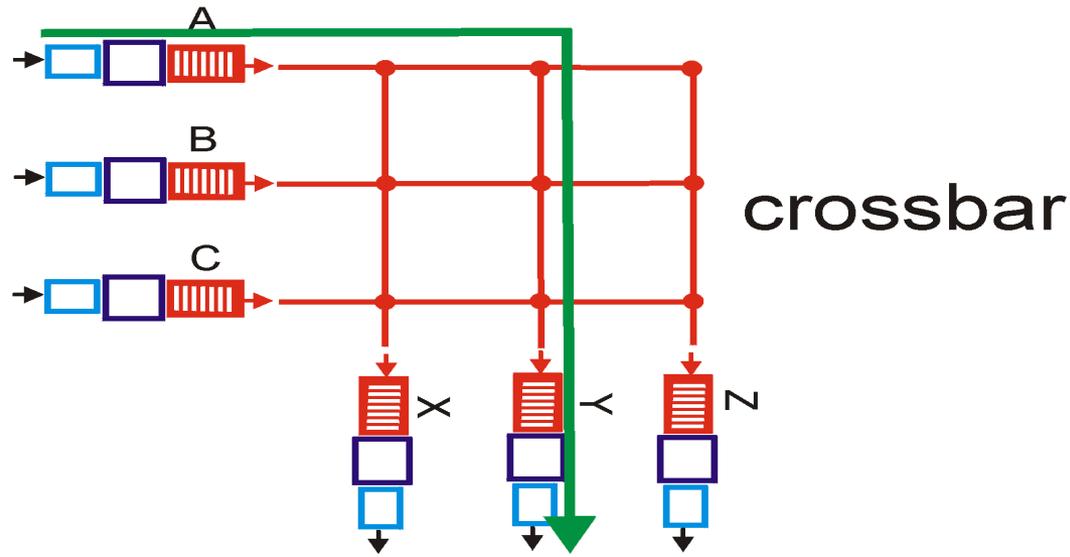
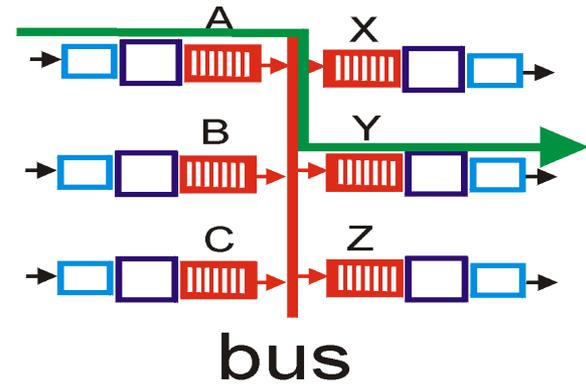
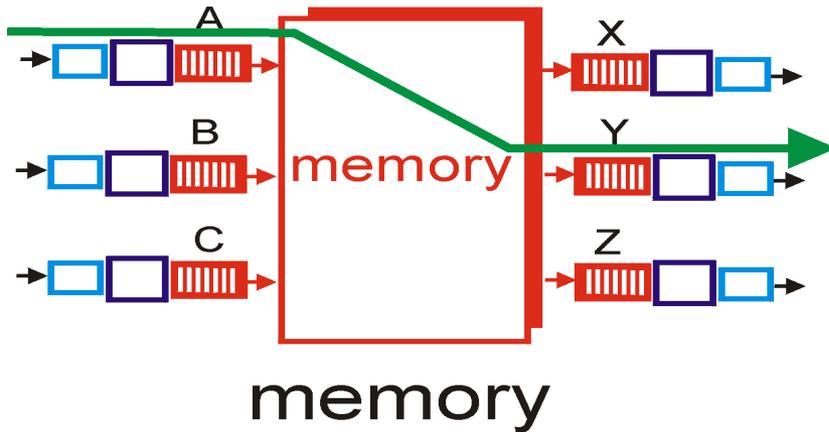
Capa física:  
recepción de bits

Capa de enlace:  
ej., ethernet  
ver cap. 5

## Conmutación descentralizada:

- Dado el destino del datagrama, buscar puerto de salida usando tabla de forwarding en memoria del puerto de entrada
- objetivo: completar procesamiento de entrada a 'velocidad de línea'
- queuing (encolamiento): si los datagramas llegan más rápido que la tasa de forwarding hacia el "switch fabric"

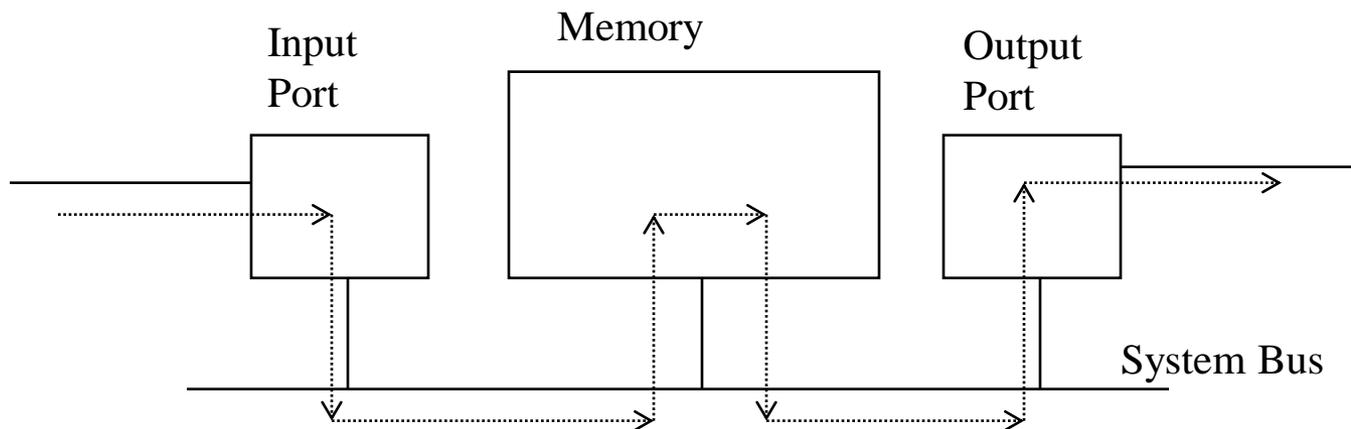
# Tipos de switching fabrics (matrices de conmutación)



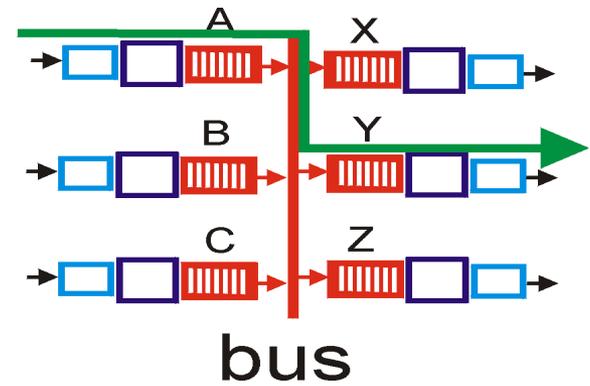
# Conmutación en memoria

## Routers de primera generación :

- ❑ computador tradicional con conmutación controlada directamente por la CPU
- ❑ los paquetes se copian a la memoria del sistema
- ❑ velocidad limitada por el ancho de banda de memoria (2 accesos al bus por datagrama)



# Conmutación en el bus

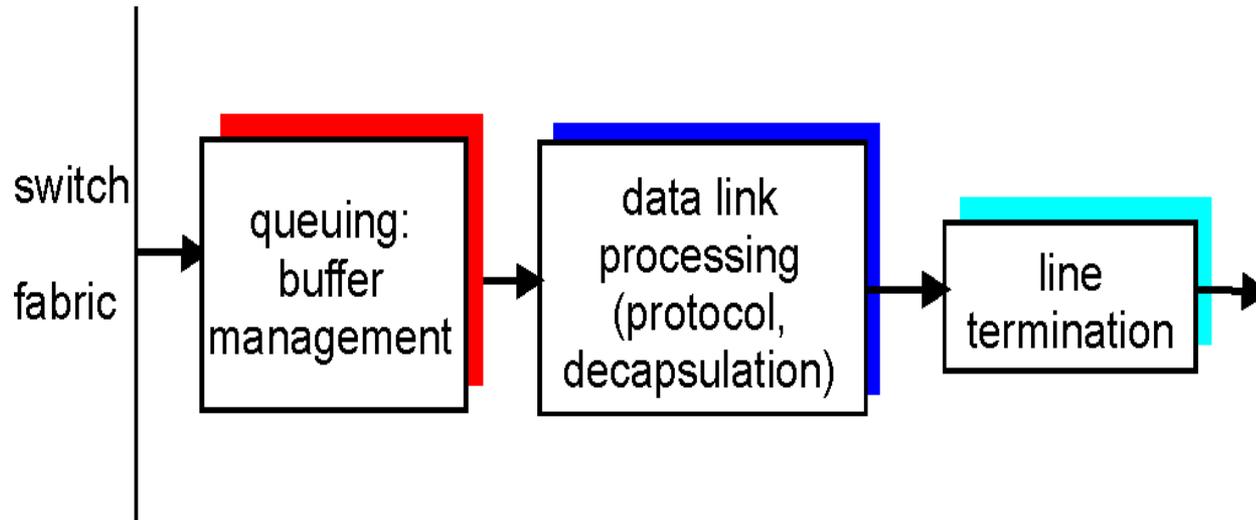


- ❑ el datagrama se copia del puerto de entrada al de salida por el bus compartido
- ❑ **bus contention:** velocidad de conmutación limitada por el ancho de banda del bus
- ❑ Ejemplo: 32 Gbps bus, Cisco 5600: velocidad suficiente para routers de acceso y empresariales

## Conmutación con una red de interconexión

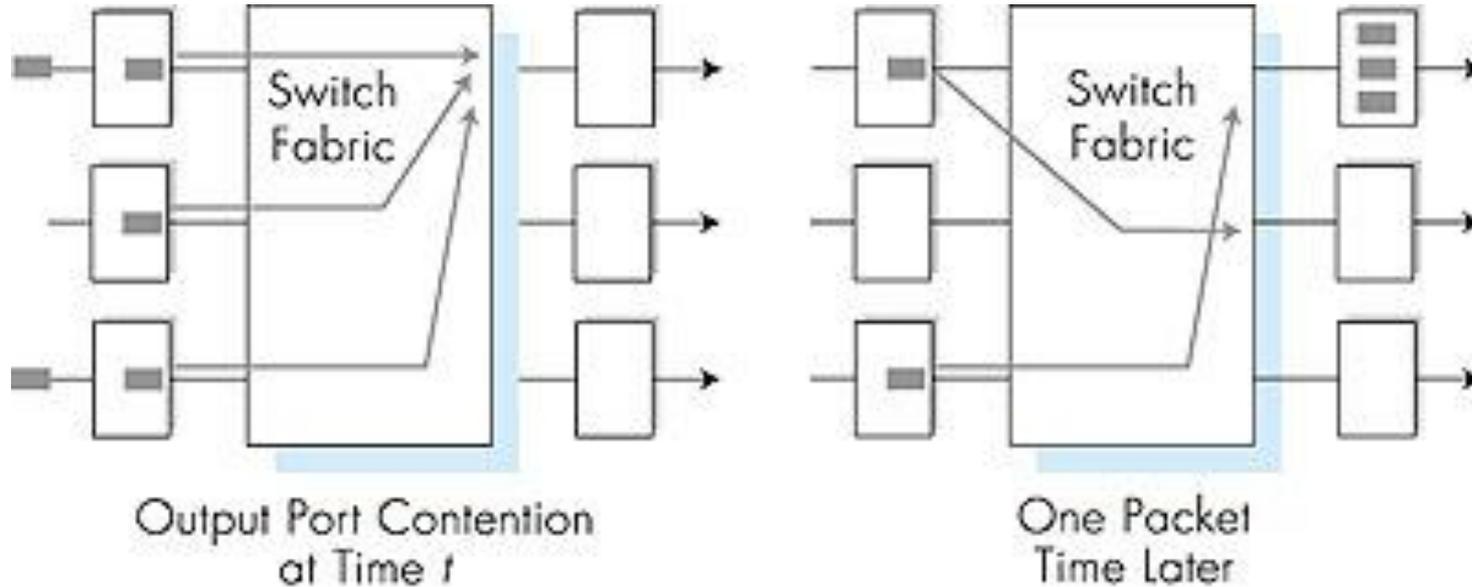
- ❑ supera las limitaciones de ancho de banda del bus
- ❑ redes de Banyan y otras inicialmente desarrolladas para interconexión de sistemas multiprocesadores
- ❑ diseño avanzado: fragmentación de datagramas en celdas de tamaño fijo, que se conmutan en la matriz
- ❑ Ej. Cisco 12000: conmuta 60 Gbps a través de la red de interconexión

# Puertos de Salida



- ❑ *buffering* (almacenamiento) requerido cuando los datagramas llegan desde la matriz más rápido que la tasa de transmisión
- ❑ *disciplina de scheduling* (despacho) elige datagramas en la cola para ser transmitidos

# Colas en Puertos de Salida



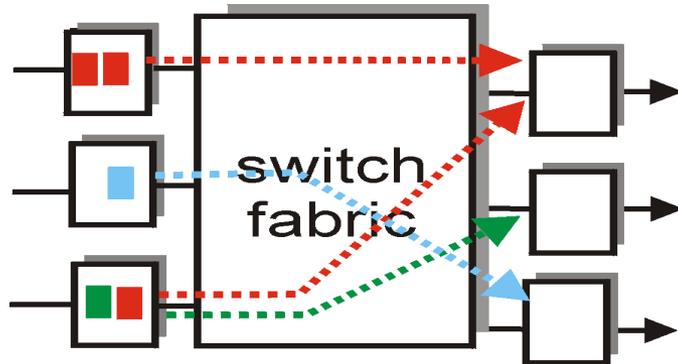
- ❑ *buffering* (almacenamiento) requerido cuando los datagramas llegan desde la matriz más rápido que la tasa de transmisión
- ❑ *pueden existir retardos y pérdidas debido a overflow del buffer del puerto de salida!*

# Tamaño del buffer?

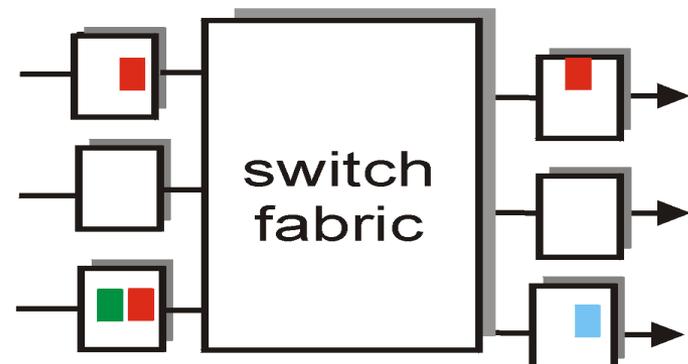
- regla de uso, RFC 3439: "buffering"  
promedio igual al producto del RTT típico  
(digamos 250 mseg) por la capacidad del  
enlace  $C$ 
  - ej.,  $C = 10$  Gps: buffer=2.5 Gbit
  - asume relativamente pocos flujos TCP
- Recomendaciones recientes: con  $N$  flujos,  
el buffer debe ser  $\frac{RTT \cdot C}{\sqrt{N}}$ 
  - asume  $N$  grande

# Colas en Puerto de Entrada

- si la matriz es más lenta que la combinación de los puertos de entrada -> se produce encolamiento
- **bloqueo Head-of-the-Line (HOL):** datagrama encolado al frente impide progresar al resto de la cola
- *pueden existir retardos y pérdidas debido a overflow del buffer del puerto de entrada!*



output port contention  
at time t - only one red  
packet can be transferred



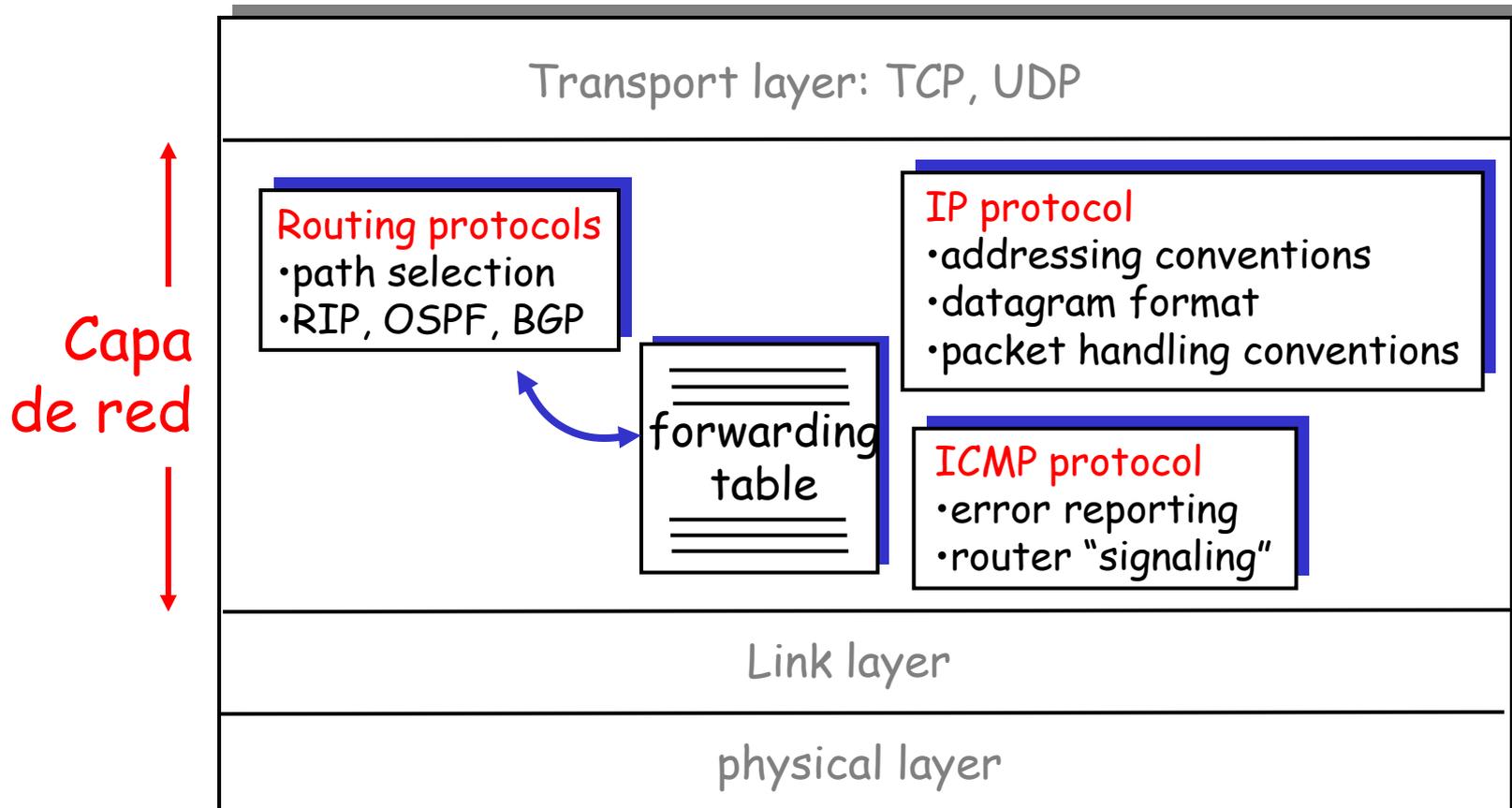
green packet  
experiences HOL blocking

# Cap. 4: Capa de red

- ❑ 4.1 Introducción
- ❑ 4.2 circuitos virtuales y datagramas
- ❑ 4.3 dentro de un router
- ❑ 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- ❑ 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- ❑ 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- ❑ 4.7 Broadcast y multicast

# La Capa de Red en Internet

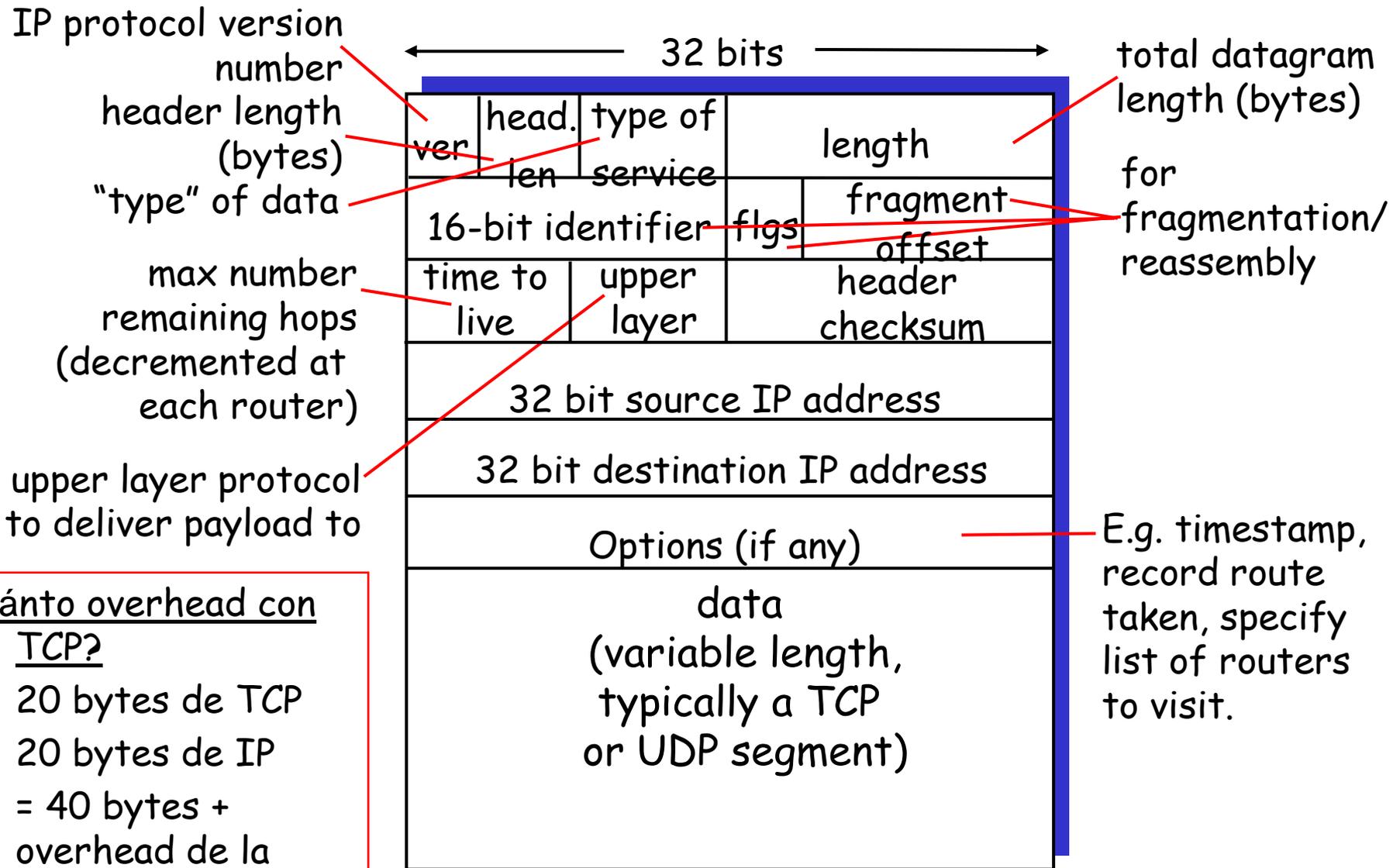
Funciones de capa de red en hosts y routers:



# Cap. 4: Capa de red

- ❑ 4.1 Introducción
- ❑ 4.2 circuitos virtuales y datagramas
- ❑ 4.3 dentro de un router
- ❑ 4.4 IP: Internet Protocol
  - formato de datagramas
  - direccionamiento IPv4
  - ICMP
  - IPv6
- ❑ 4.5 Algoritmos de enrutamiento
  - Link state
  - Distance Vector
  - Enrutamiento jerárquico
- ❑ 4.6 Enrutamiento en Internet
  - RIP
  - OSPF
  - BGP
- ❑ 4.7 Broadcast y multicast

# Formato del datagrama IP



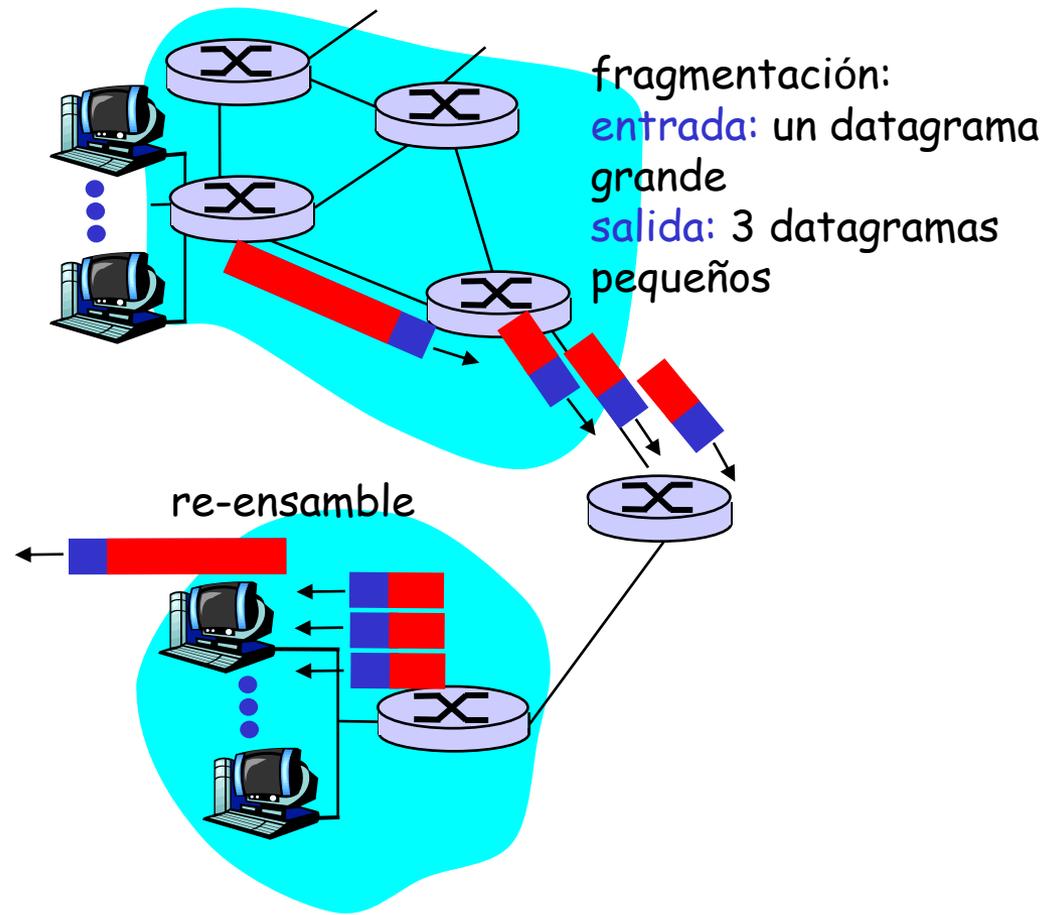
cuánto overhead con

TCP?

- ❑ 20 bytes de TCP
- ❑ 20 bytes de IP
- ❑ = 40 bytes + overhead de la aplicación

# Fragmentación & Re-ensamblado

- los enlaces tienen un tamaño máximo de la unidad de transmisión: MTU
  - enlaces diferentes, MTUs diferentes
- los datagramas IP muy grandes pueden ser fragmentados en la red
  - un datagrama se transforma en muchos datagramas
  - "re-ensamble" en el destino
  - se usan bits del cabezal IP para identificar y ordenar los fragmentos



# Fragmentación & Re-ensamblado

## Ejemplo

- ❑ datagrama de 4000 bytes
- ❑ MTU = 1500 bytes

	length	ID	fragflag	offset	
	=4000	=x	=0	=0	

un datagrama grande se transforma en muchos datagramas más pequeños

1480 bytes en campo de datos

offset =  $1480/8$

	length	ID	fragflag	offset	
	=1500	=x	=1	=0	

	length	ID	fragflag	offset	
	=1500	=x	=1	=185	

	length	ID	fragflag	offset	
	=1040	=x	=0	=370	