

Transporte fiable

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Arquitectura de Redes, Sistemas y Servicios

upna

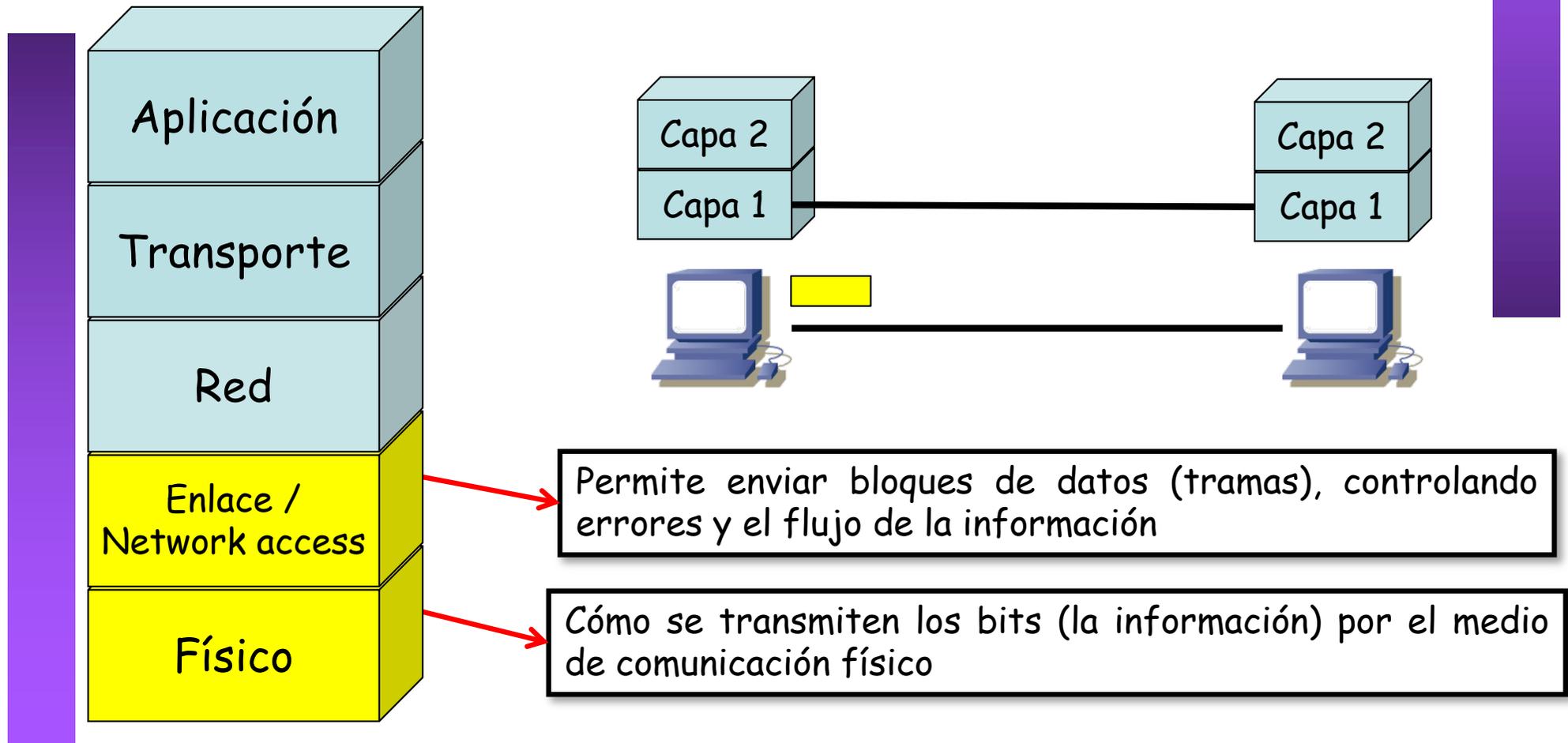
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS
Área de Ingeniería Telemática

Capas 1, 2 y 3

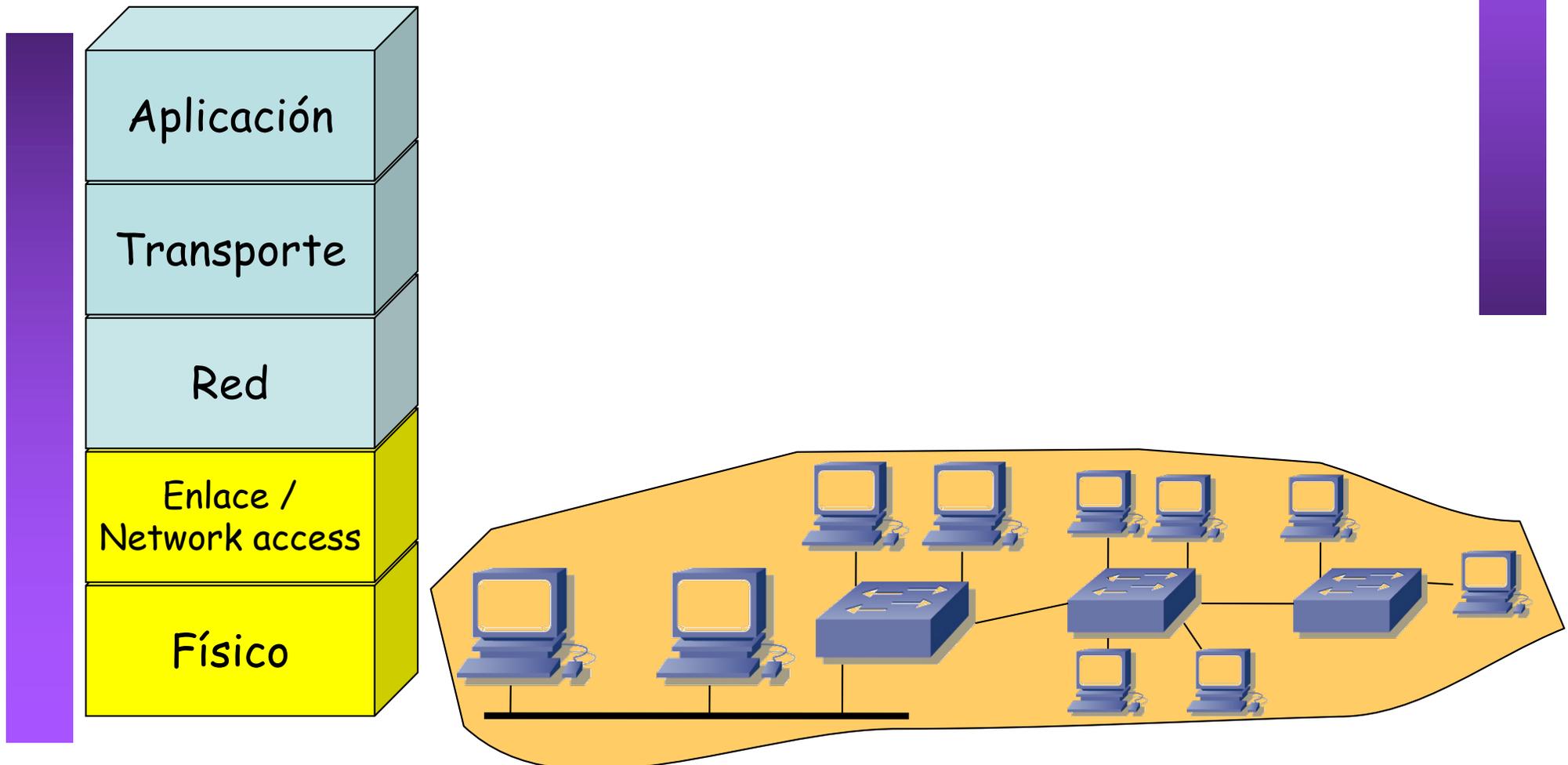
Nivel de enlace

- La comunicación básica en un enlace requiere los niveles 1 y 2
- Con eso podemos hacer llegar las tramas (PDUs de capa 2) de un equipo a otro adyacente



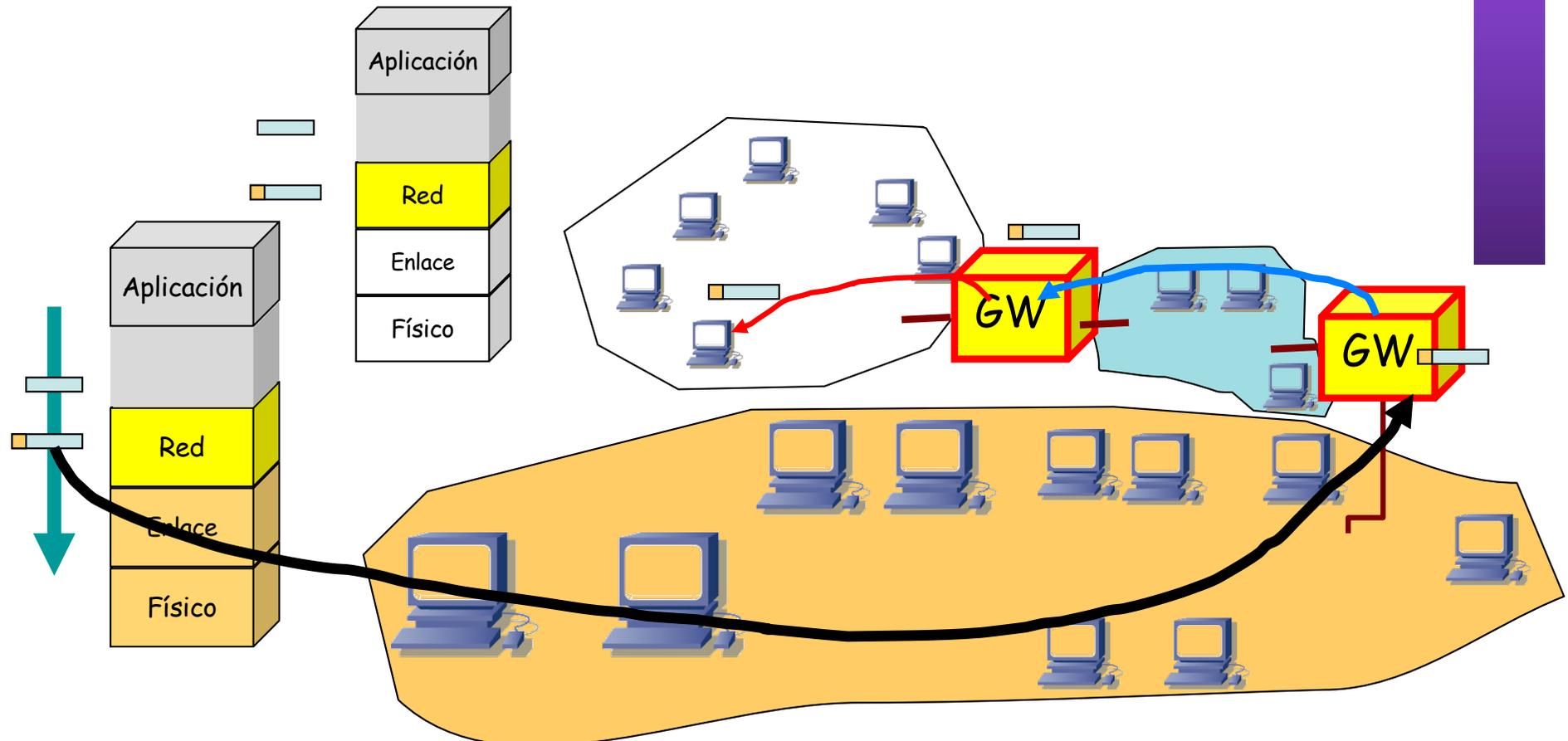
Comunicación dentro de una red

- Origen y destino del paquete están en la misma red
 - Dos hosts
 - Un host y un “gateway” con otra red
 - Dos “gateways”
- La red puede ser una LAN, MAN o WAN



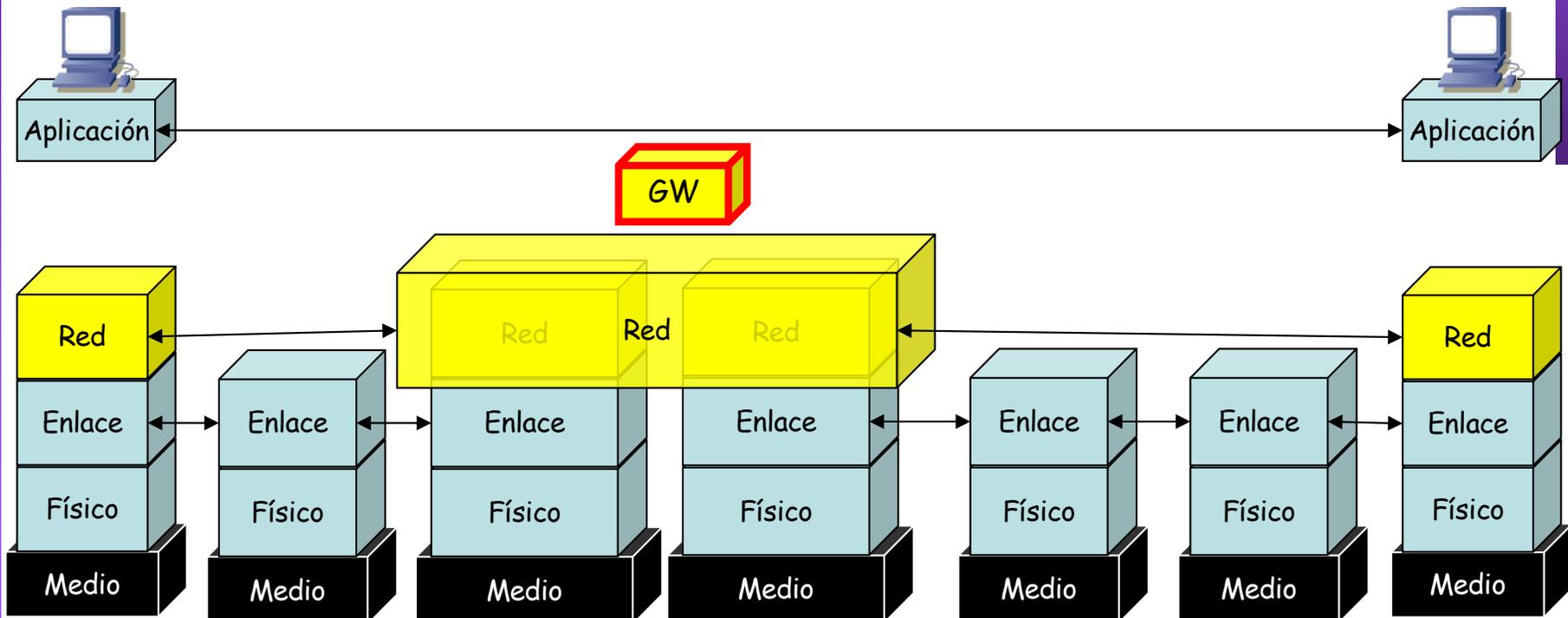
Capa 3

- Protocolo de capa 3 incluye las pasarelas de interconexión
- Esas pasarelas deben conocer el camino al destino
- Nos da comunicación extremo a extremo
- Los datos del paquete de capa 3 (ej: datos del paquete IP) llegan al host destino sin alterar



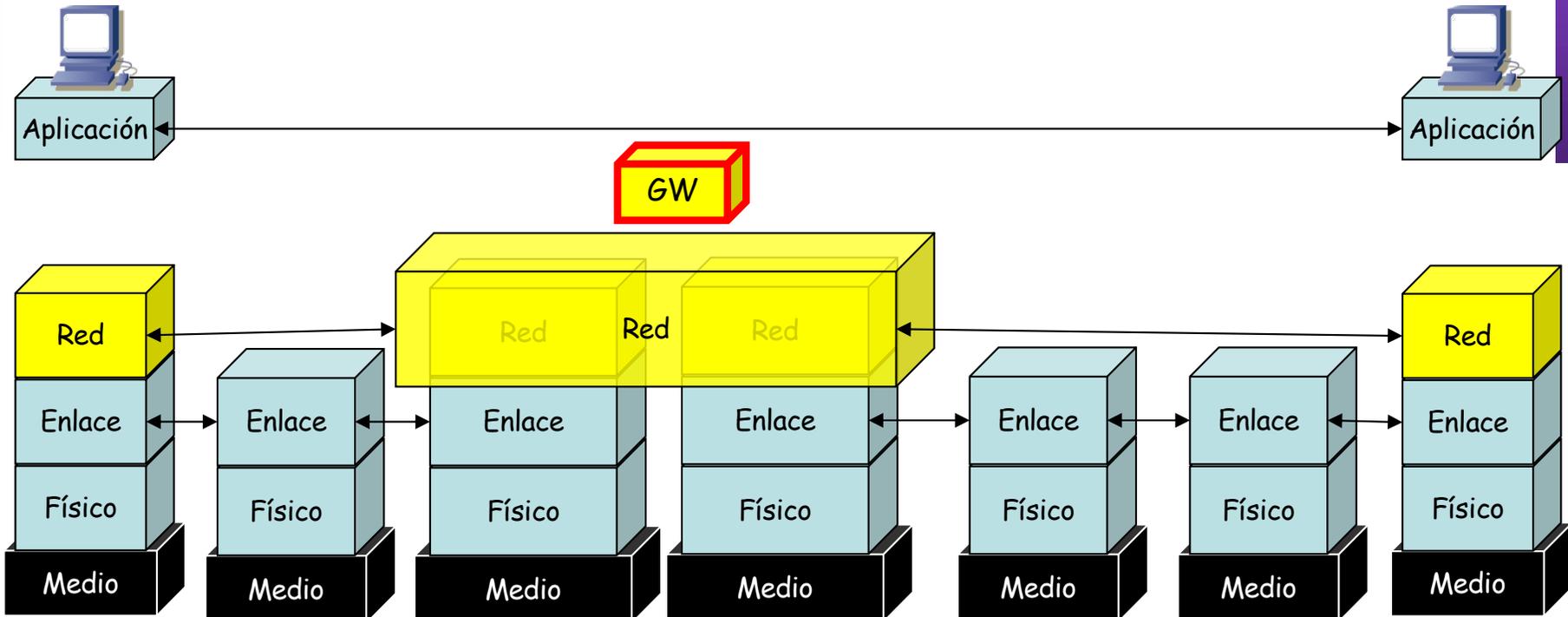
Diferentes equipos

- Las pasarelas de capa 3 (conmutadores capa 3, routers) procesan la cabecera de ese nivel
- Toman decisiones en base a campos en esa cabecera
- Por ejemplo, en base a la dirección (capa 3) destino



1, 2, 3, 4...

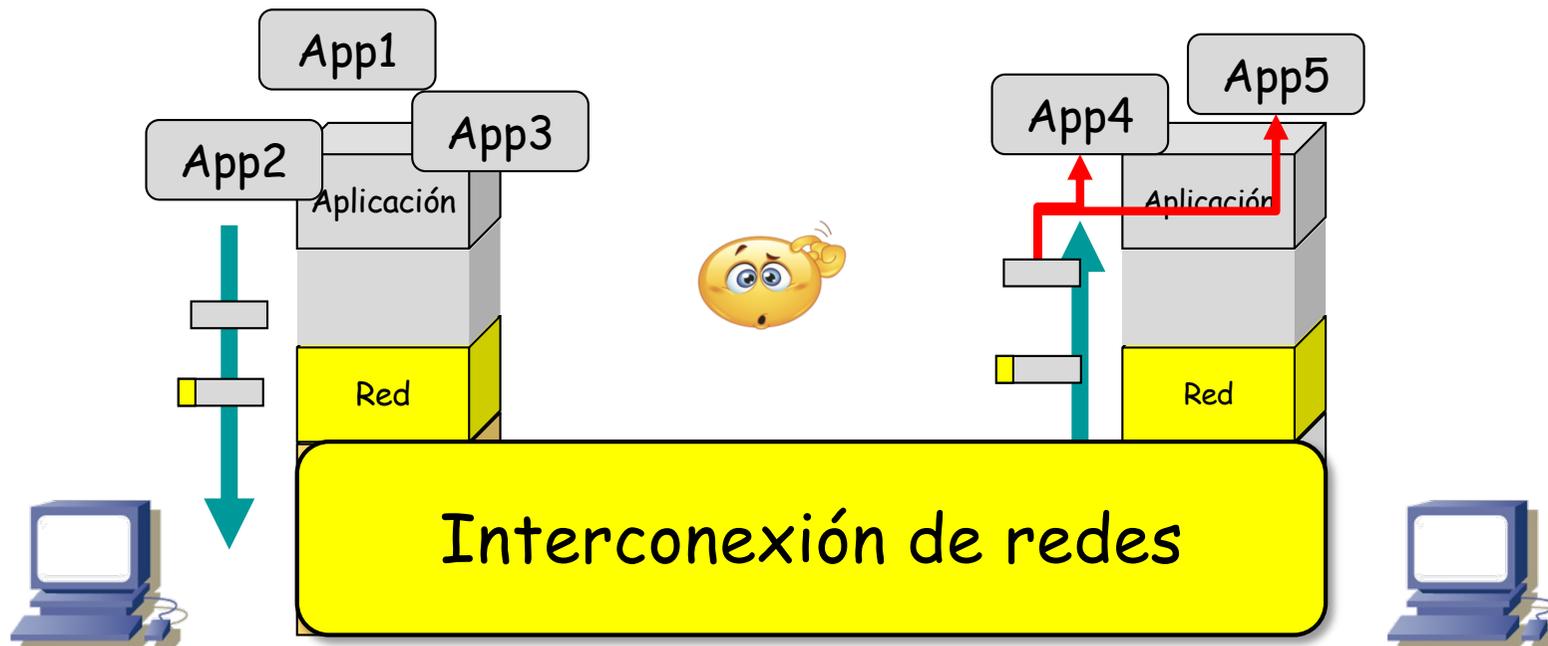
- Capa 1+2 es todo lo que necesitamos para enviar un paquete a otro host de la misma red
- Capa 1+2+3 es todo lo que necesitamos para enviar un paquete a otro host de la misma u otra subred
- Eso quiere decir por ejemplo que la Internet lo que provee es un servicio de conectividad en capa 3
- ¿Qué nos añade la capa 4? (“Nivel de transporte”)



Multiplexación en Capa 4

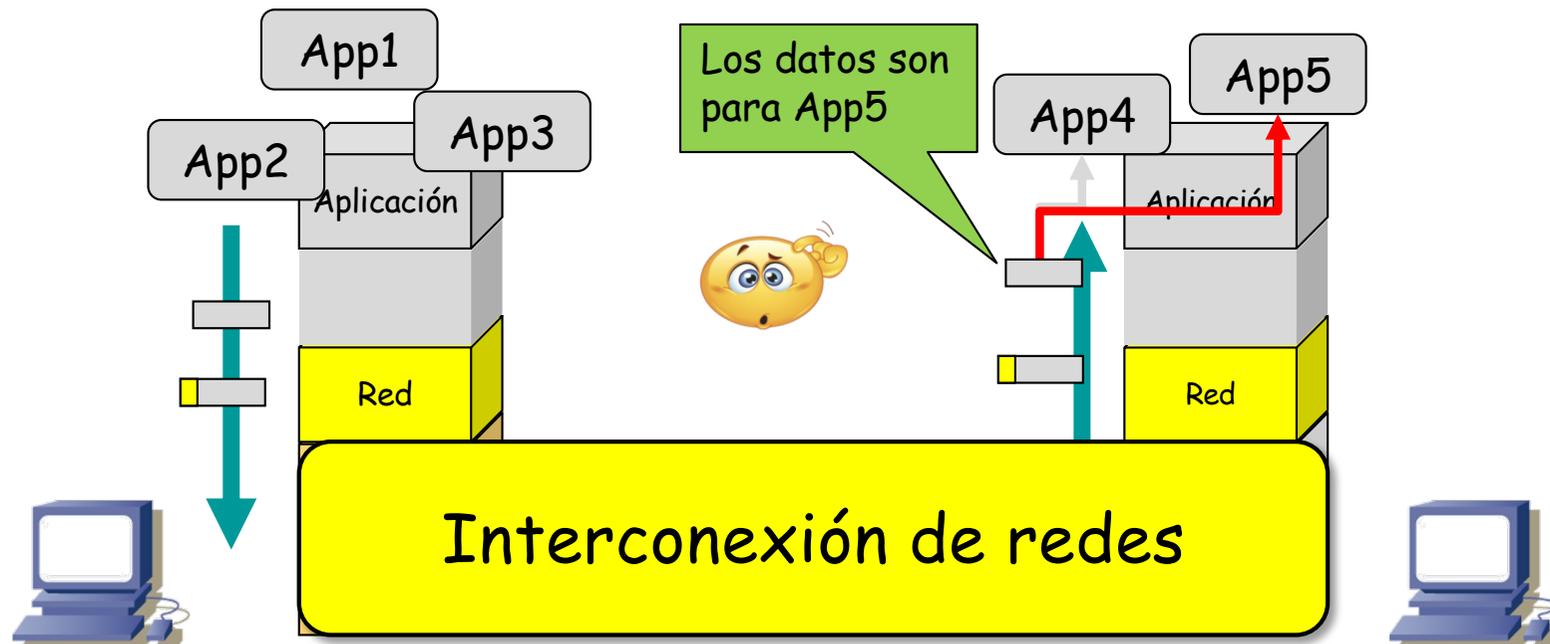
Aplicaciones

- Las aplicaciones pueden comunicarse enviando paquetes, de capa 3 de un host al otro, que contengan los datos de la App
- ¿Cómo sabe el host que recibe un paquete de capa 3 que es para él?: Direcccionamiento en capa 3
- ¿Pero cómo sabe el host destino de un paquete de capa 3 **para qué aplicación** de las que tiene corriendo es?



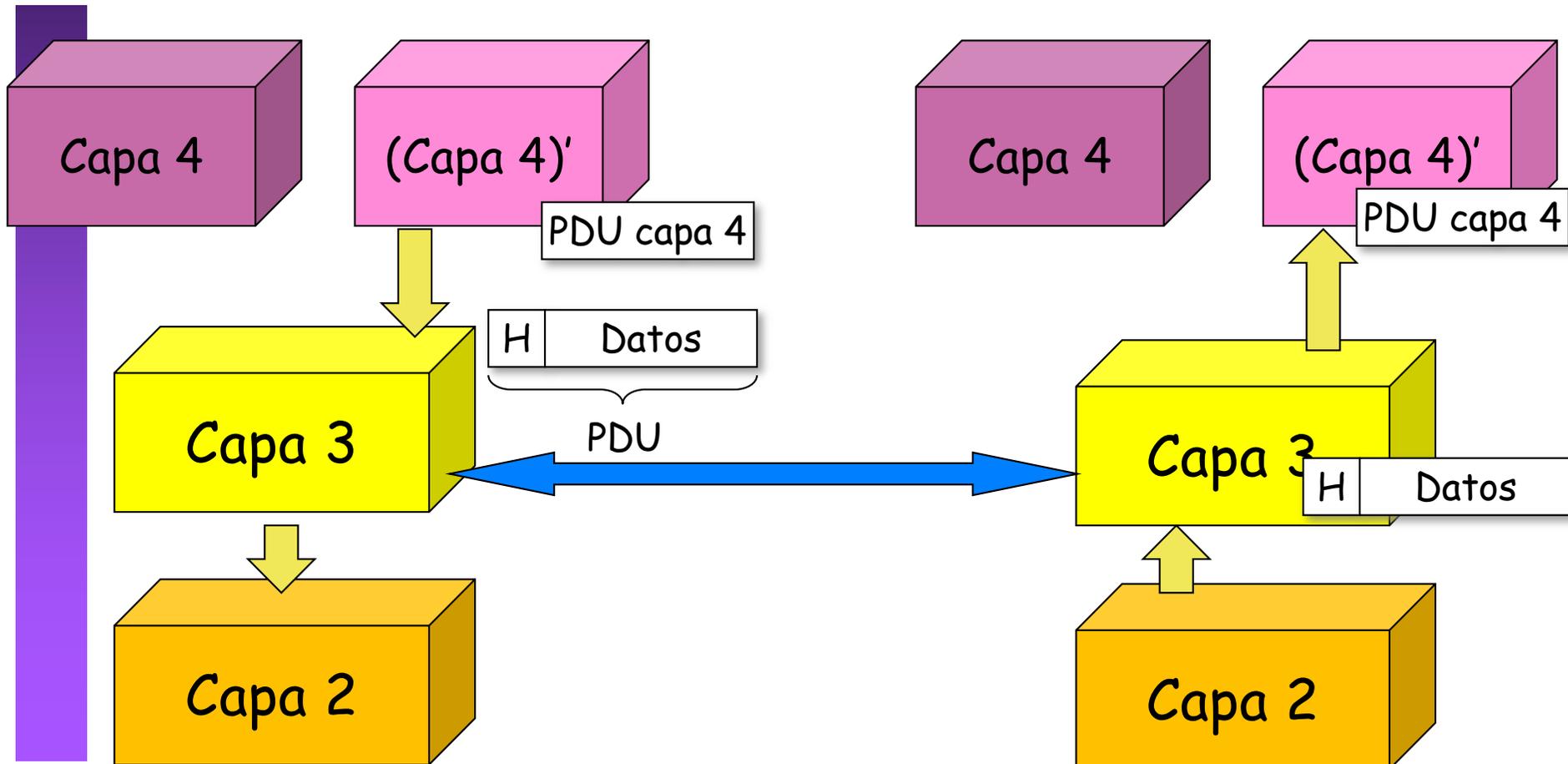
Direccionamiento

- Necesitamos poder “direccionar” a la aplicación destino
- También indicar cuál es la aplicación origen para que se le pueda contestar
- Normalmente algún campo en el paquete servirá para eso
- El protocolo de capa 3 no suele tener esta funcionalidad
- Es una tarea de protocolo de capa 4 (Nivel de Transporte)



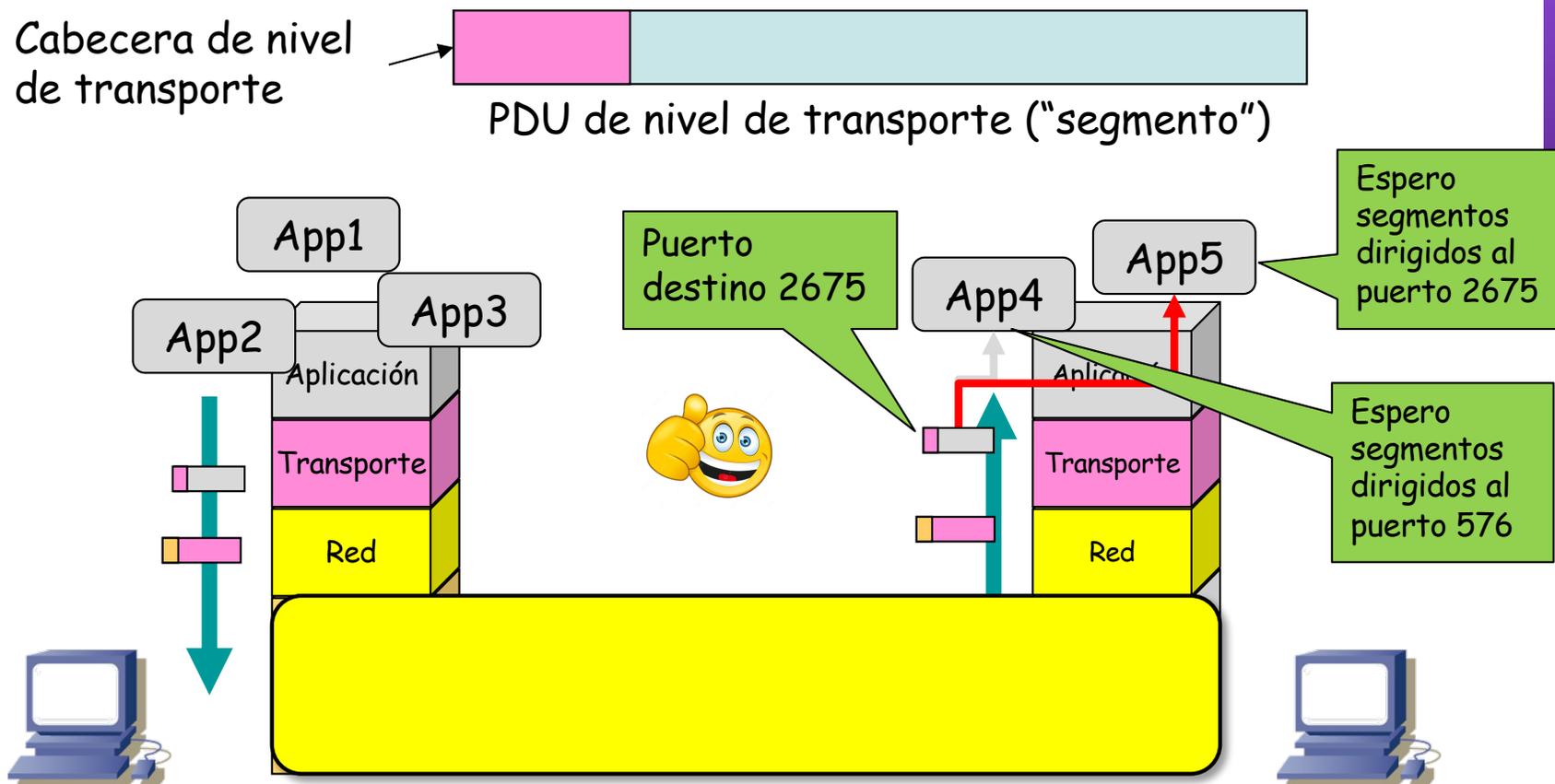
Multiplexación en capa 3

- Lo que sí puede multiplexar el protocolo de capa 3 es múltiples protocolos de capa 4
- Por ejemplo: el protocolo IP tiene un campo en la cabecera para indicar los datos de qué protocolo son (UDP, TCP, otros...)
- Lo mismo que en capa 2 Ethernet hace el Ethertype



Nivel de Transporte: Mux

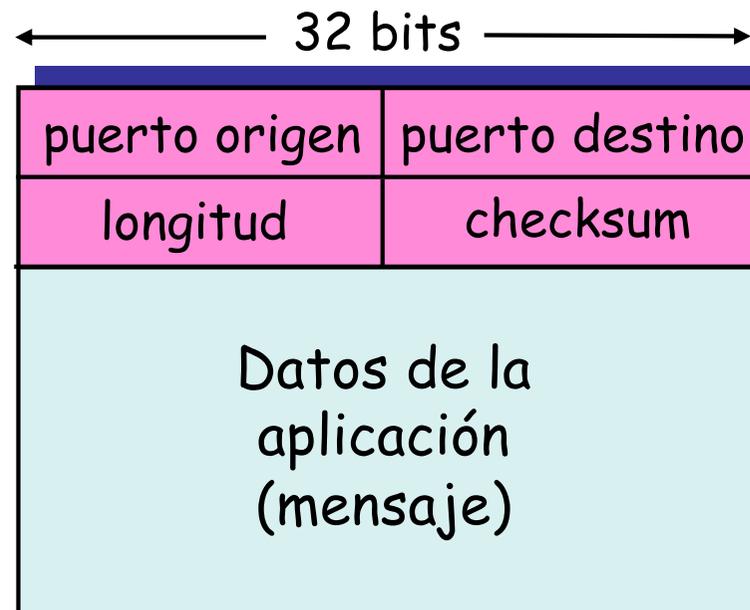
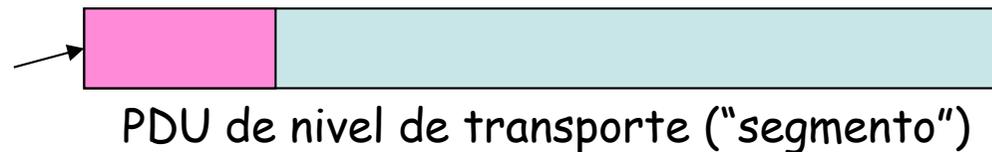
- El protocolo de nivel de transporte suele dar esta funcionalidad
- Por ejemplo, en el caso de los protocolos UDP y TCP ese valor se conoce como el “puerto” y son campos en la cabecera de nivel 4
- “Puerto destino” identifica a la App destino
- “Puerto origen” identifica a la App origen



Ejemplo: UDP

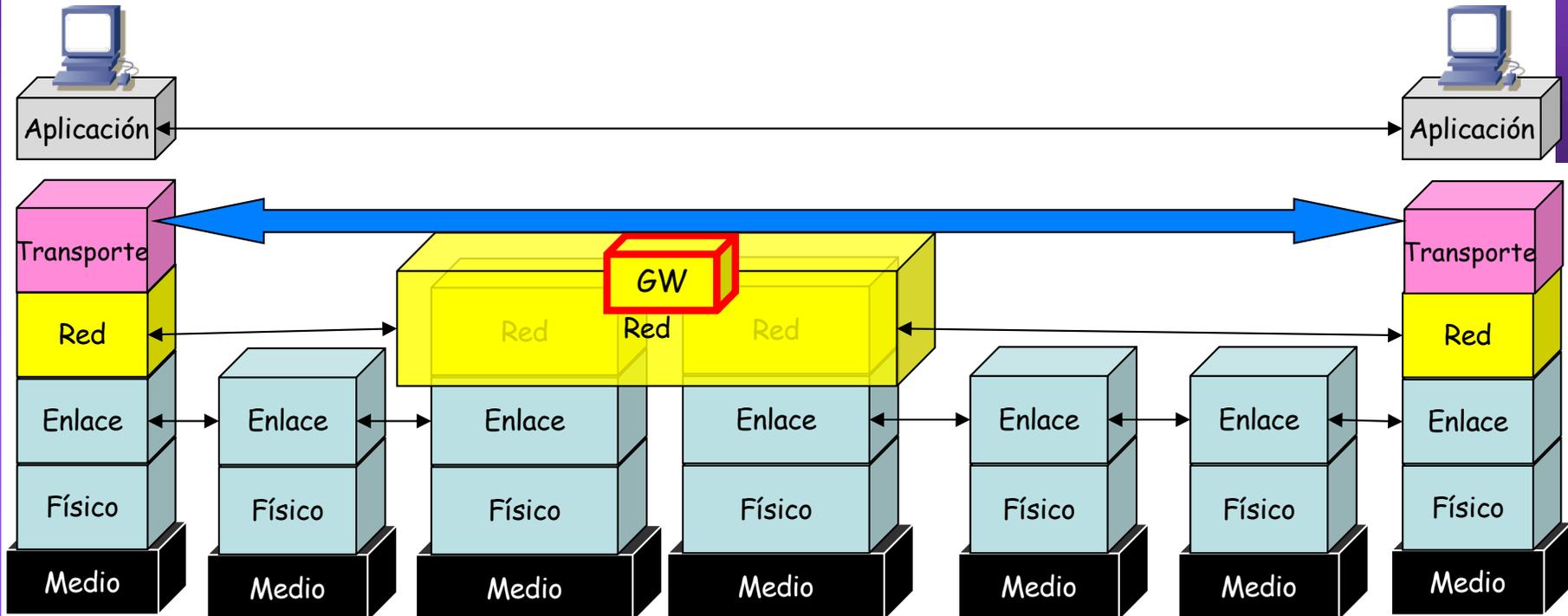
- El protocolo de nivel de transporte suele dar esta funcionalidad
- Por ejemplo, en el caso de los protocolos UDP y TCP ese valor se conoce como el “puerto” y son campos en la cabecera de nivel 4
- “Puerto destino” identifica a la App destino
- “Puerto origen” identifica a la App origen

Cabecera de nivel de transporte



Nivel de Transporte

- El primer nivel implementado exclusivamente en los hosts extremo
- Se dice que su comunicación es "extremo-a-extremo"
- Protocolo de capa 3 e inferiores tratan su PDU (cabecera capa 4 y datos) como datos
- Los equipos de red no entienden la cabecera de capa 4 y actúan exclusivamente en base a las cabeceras inferiores



upna

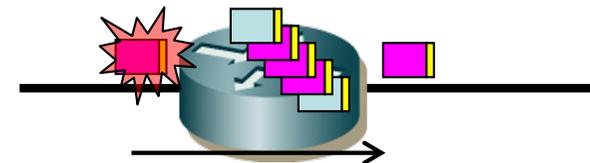
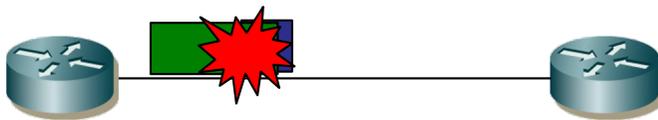
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS
Área de Ingeniería Telemática

Transporte fiable y capas

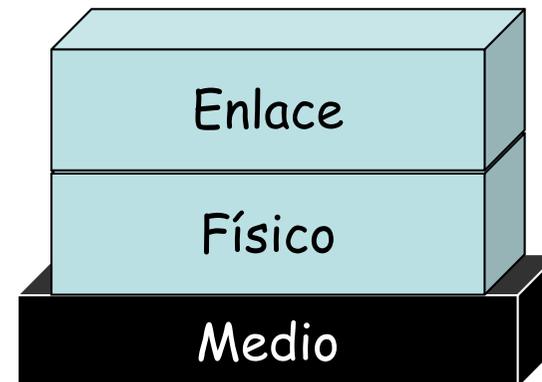
Transporte fiable

- Los paquetes se pueden “perder”
- Pueden corromperse y llegar estropeados o no llegar (corrupción en cabecera que impide llegar al destino)
- Pueden descartarse por congestión (buffer lleno)
- Pueden duplicarse
- Pueden llegar desordenados
- Normalmente las pérdidas (errores o descartes) las resolveremos con retransmisiones
- Los desórdenes y duplicidades requerirán que numeremos de alguna forma lo enviado para detectarlos y reordenarlos



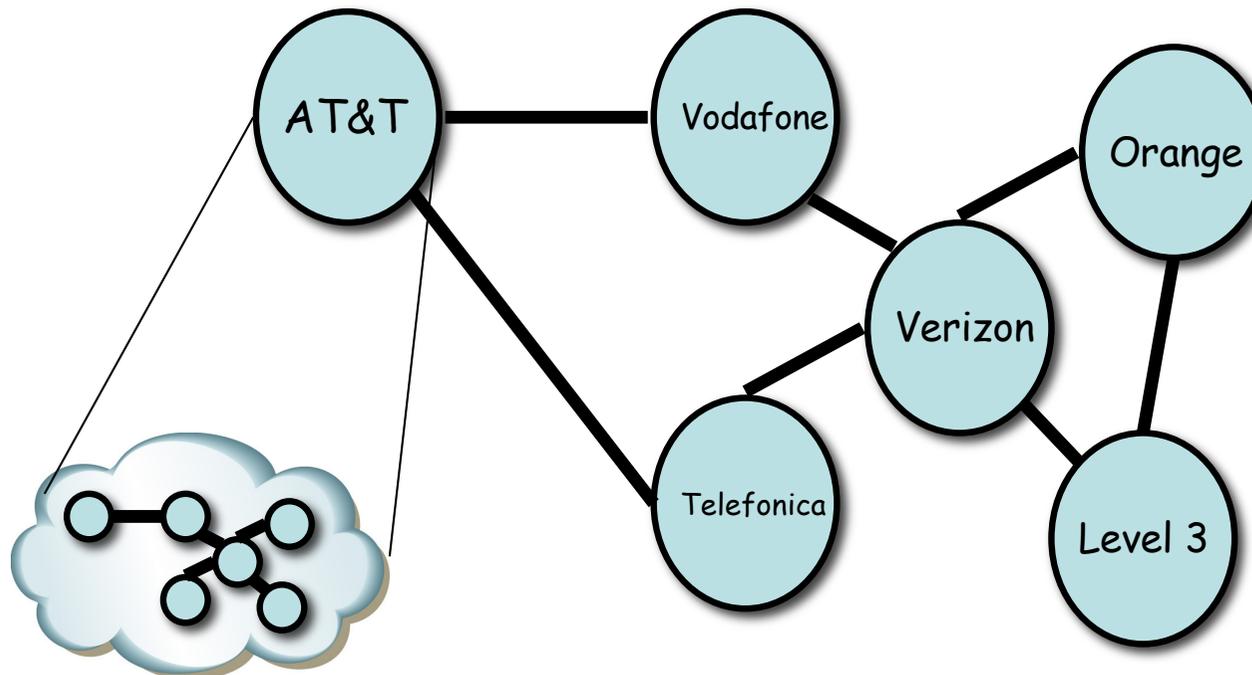
Entrega fiable y capa 2

- El nivel de enlace podría añadir mecanismos de entrega fiable
- Por ejemplo, WiFi:
 - Lo intenta hacer con sus confirmaciones (aunque puede cansarse de retransmitir)
 - No hay posibilidad de desorden en ese salto único
- Algunas tecnologías de nivel de enlace implementan mecanismos de retransmisión y otras no
- Por ejemplo, Ethernet:
 - En caso de descarte por buffer lleno en conmutador el paquete se ha perdido, no hay retransmisión
 - En hubs, en caso de colisión hay retransmisión desde el host origen, pero no garantiza, puede cansarse de hacer retransmisiones
- Lo que vamos a trabajar sobre transporte fiable tiene sentido en capa 2
- Por ejemplo, HDLC



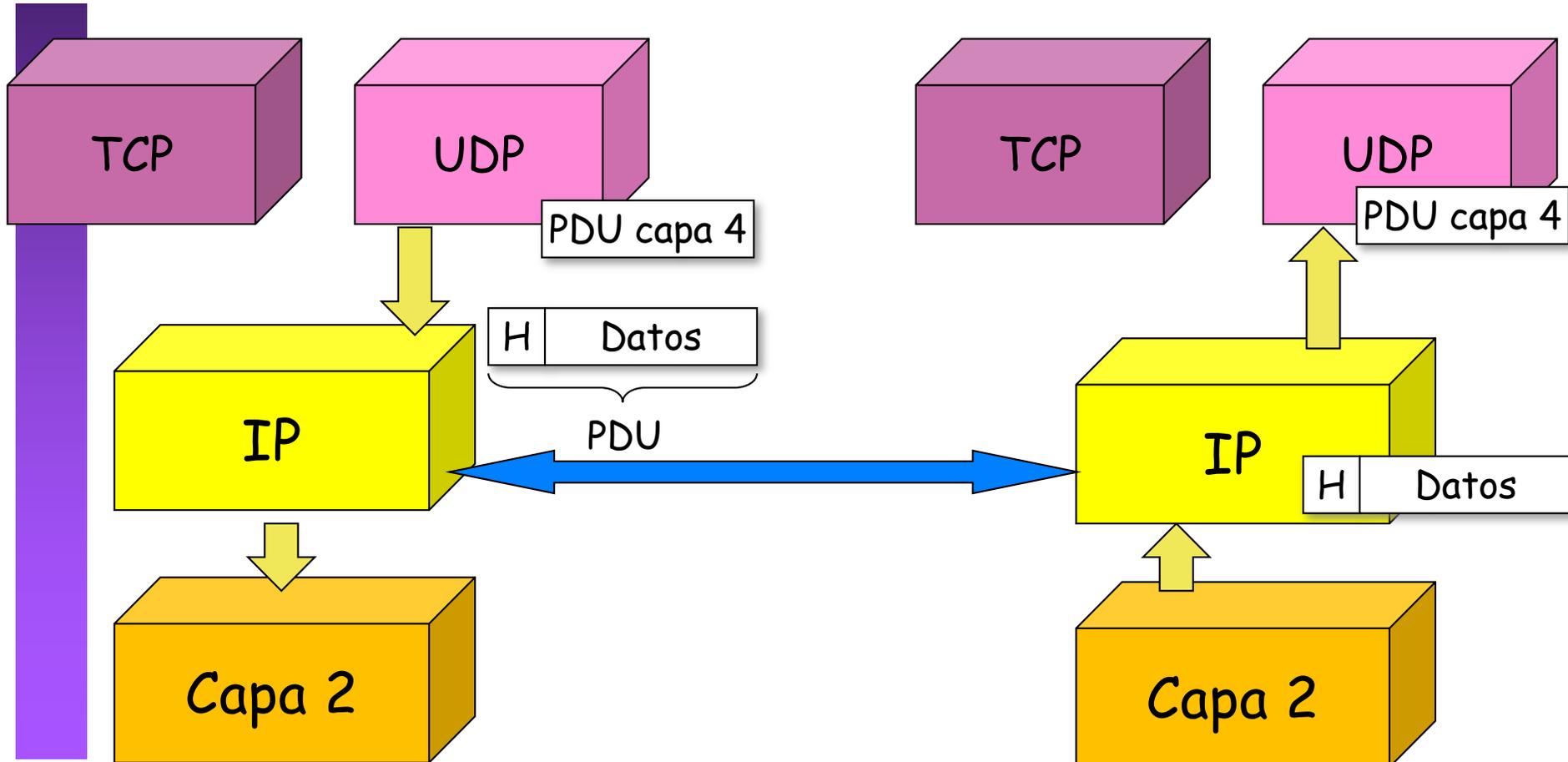
Transporte fiable y capa 3

- En una interconexión de redes, extremo a extremo damos el peor servicio de los servicios ofrecidos por cada salto
- Si en un salto se pueden perder los paquetes, extremo a extremo se pueden perder
- IP ofrece así un servicio *best effort*
- El transporte fiable suele ser tarea del protocolo de capa 4 OSI (por ejemplo, en TCP)



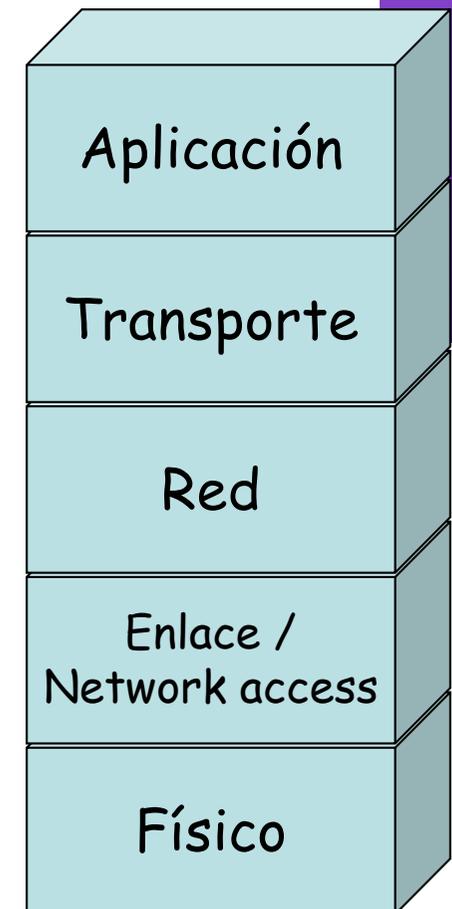
Transporte fiable en Internet

- Dos protocolos principales de nivel de transporte: TCP y UDP
- TCP ofrece multiplexación, transporte fiable (recuperación ante pérdidas y desórdenes), control de conexión, control de flujo, control de congestión
- UDP ofrece... multiplexación.
- ¿Una aplicación que emplee UDP puede lograr un envío “fiable” extremo a extremo? Sí, si se lo implementa ella en nivel de aplicación



Transporte fiable en...

- En tecnologías capa 2:
 - No en Ethernet, ni WiFi, ni ATM
 - En ATM sí hay entrega en orden
- En tecnologías capa 3:
 - No en IPv4, ni IPv6, ni CLNP, ni AppleTalk, ni IPX
 - Sí en X.25
- En protocolos capa 4:
 - No en UDP ni en TP0
 - Sí en TCP, SPX, ATP y TP4
- En la aplicación:
 - Si el nivel de transporte no lo ofrece y la aplicación lo requiere
 - En la propia aplicación se deberán implementar los mecanismos:
 - Reconocer pérdidas
 - Hacer retransmisiones
 - Reordenar (si es deseado)



upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS
Área de Ingeniería Telemática

Control de flujo

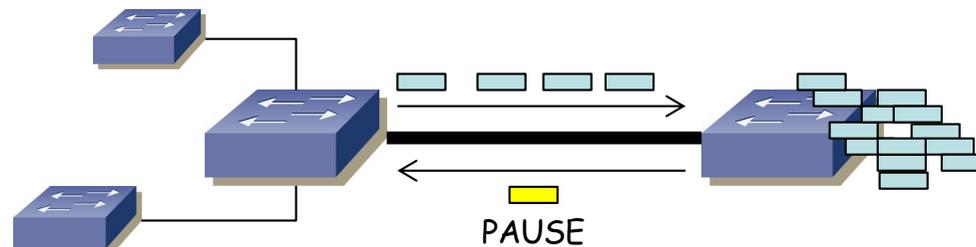
Flow control

- Las pérdidas pueden producirse en el host receptor
- Normalmente porque la aplicación que procesa los datos no puede hacerlo al mismo ritmo que se reciben
- Llegará el punto en el que se le acumularán tantos datos en memoria para procesar que tendrá que descartarlos
- No es un problema de la red, ella ha hecho llegar los paquetes al destino correctamente
- El protocolo de capa 4 puede implementar control de flujo para que el destino pueda regular el ritmo al que el origen envía los datos
- Controla la velocidad a la que se envía el flujo



Flow control en capa 2

- Flow-control también tiene sentido en el nivel de enlace
- Por ejemplo, para que un conmutador regule el ritmo de paquetes que está recibiendo de otro adyacente
- Cuando receptor cerca de saturación de buffer
- Ethernet tiene soporte para esto
- Nos centraremos en control de flujo en capa 4 (TCP)



upna

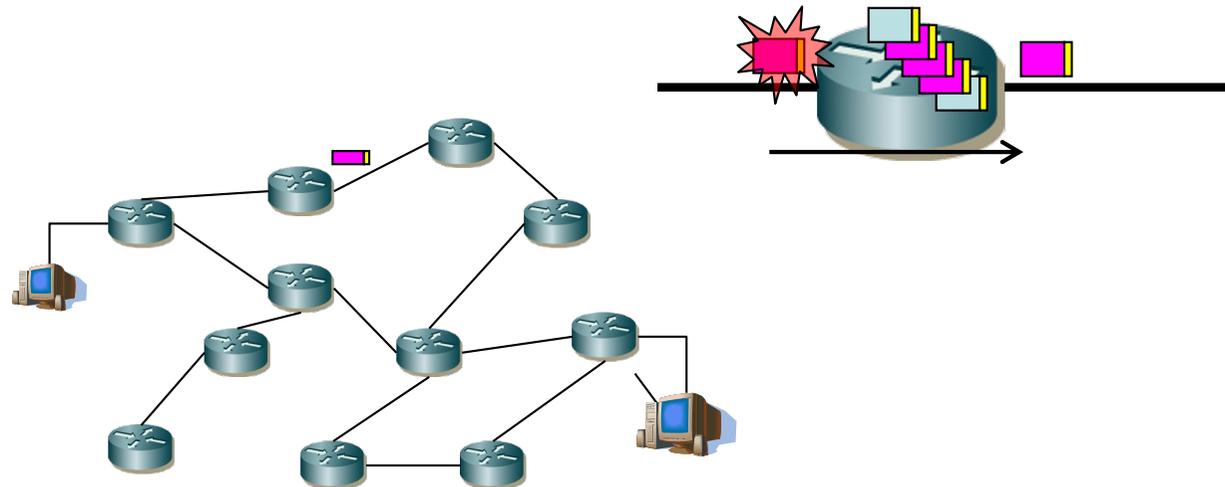
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS
Área de Ingeniería Telemática

Control de congestión

Control de congestión

- Pérdidas por buffer lleno en conmutador
- Se habla de que la red sufre “congestión”
- Evitarlo requiere que los hosts se coordinen para enviar más despacio
- La abuela de la Internet llegó a “colapsar” por congestión (más retransmisiones que datos útiles)
- Protocolos de capa 4 suelen implementarlo (TCP)
- Controla la velocidad a la que se envía el flujo
- Control de flujo + control de congestión van a regular la velocidad a la que envía paquetes el nivel de transporte en una conexión
- Por ejemplo a qué velocidad un usuario se descarga un fichero



upna

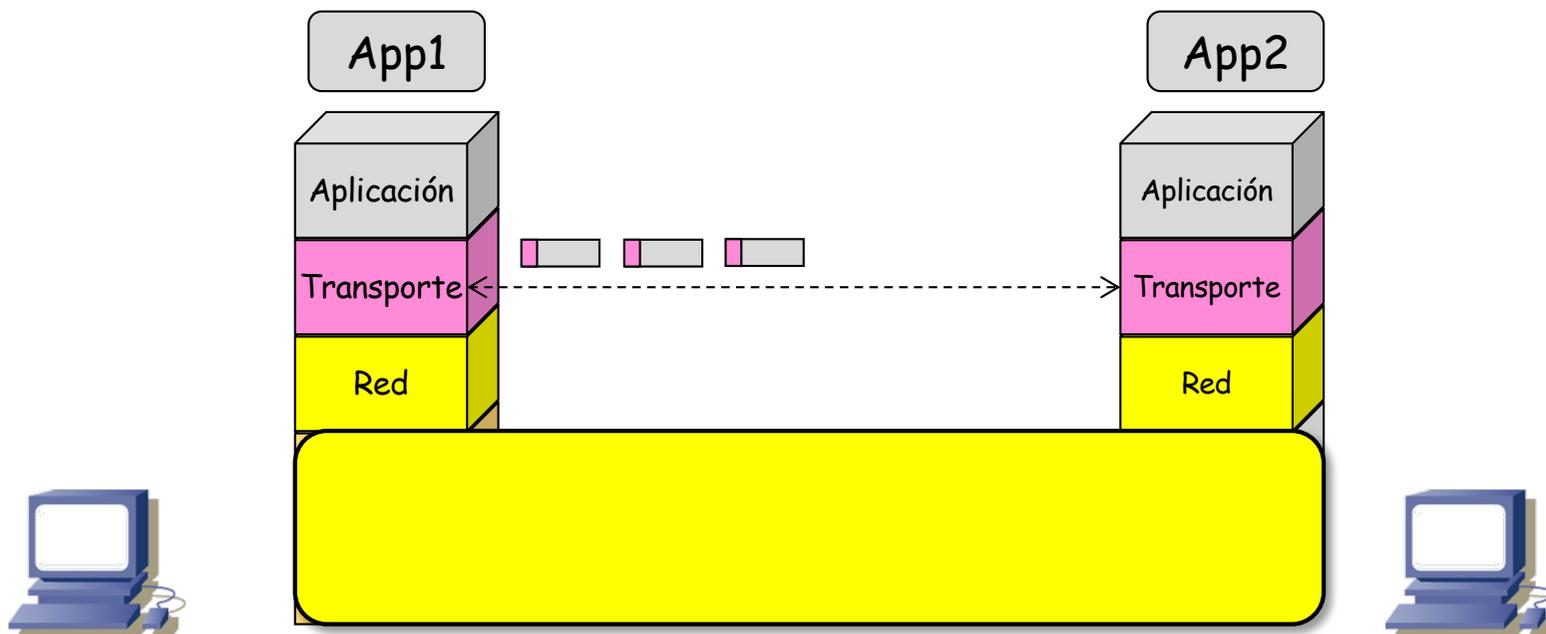
Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS
Área de Ingeniería Telemática

Control de conexión

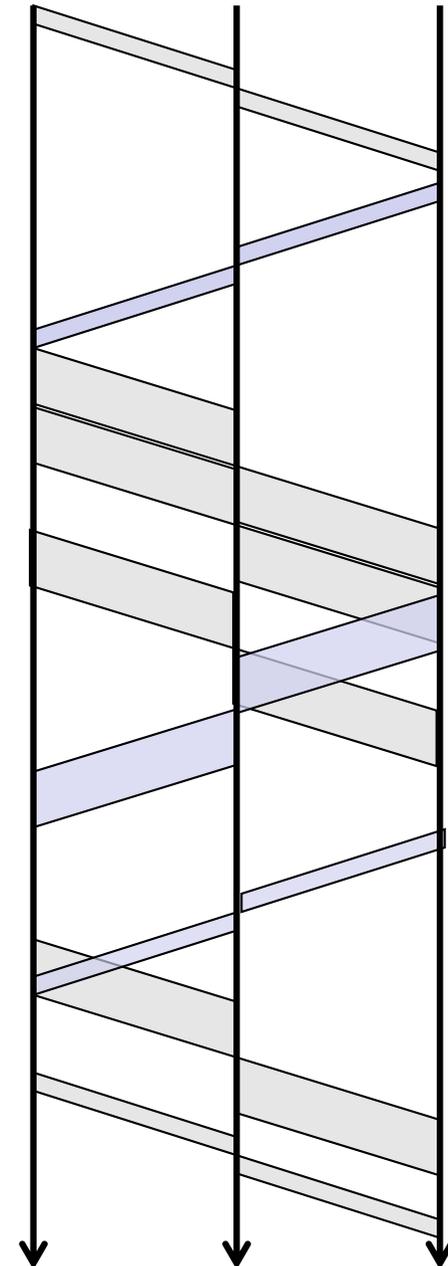
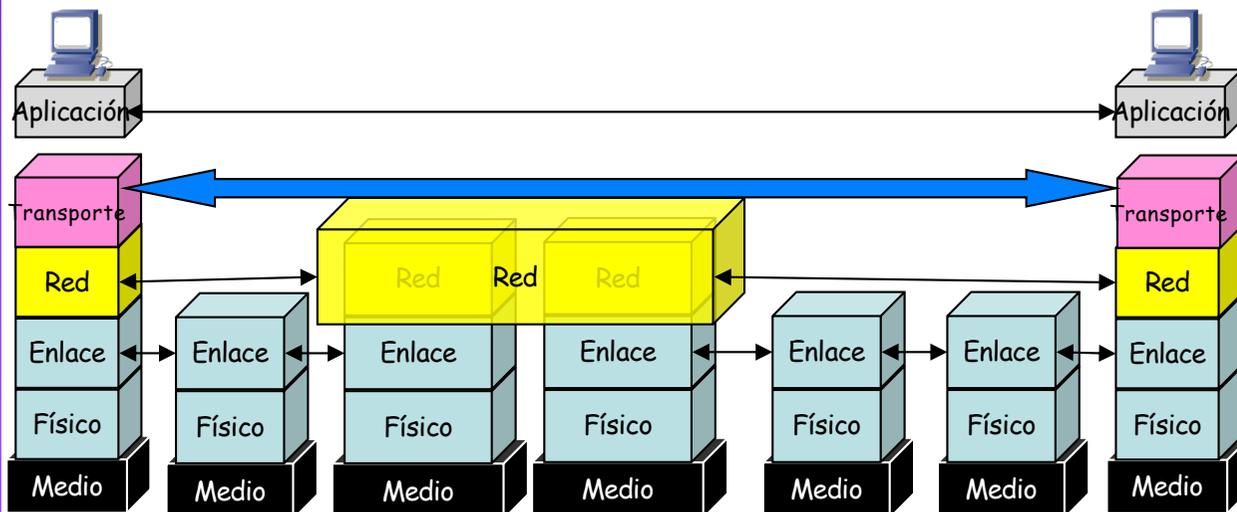
Orientado a conexión

- Los segmentos de capa 4 llegan sin cambios extremo-a-extremo
- Mantener el orden para los segmentos de una misma “conversación”
- Esa “conversación” la llamamos una “conexión” por sus similitudes:
 - Segmentos se intercambian entre los niveles 4 extremo para acordar el inicio de la misma (similar a circuitos virtuales, pero en capa 4)
 - A partir de ahí el intercambio de datos
 - Fase de finalización
- Es un acuerdo virtual o lógico entre los extremos no hay nada físico
- Capa 3 e inferiores solo ven que circulan segmentos



Orientado a conexión

- Conexiones capa 4 son conceptos lógicos en los equipos extremo
- Ningún equipo de red tiene conocimiento de ellas
- ¿Conexiones en capa 3?
 - Involucrarían a los conmutadores capa 3
 - Lo más parecido serían los circuitos virtuales en tecnologías como ATM (discutido si capa 2 ó 3)
- Conexiones capa 4 dan habitualmente un servicio de comunicación full-duplex



Mecanismos en capa 4

- Multiplexación
- Transporte fiable
 - Retransmisiones
 - Entrega en orden
- Control de flujo
- Control de congestión
- Control de conexión (“orientado a conexión”)