

# Arquitecturas de redes e Internet

Programación y administración de redes - Semana 2

Grado en *Ingeniería Informática*

Departamento de Informática. Universidad de Jaén

# Objetivos

## General

*Aprender a identificar los elementos principales de una arquitectura de red, sus protocolos, servicios, tipos de comunicación y los modelos existentes*

## Específicos

- **Aspectos** a considerar al diseñar una arquitectura de red
- Identificar las características básicas de **protocolos, interfaces y servicios**
- Diferenciar entre comunicación **lógica y física**
- Reconocer similitudes y diferencias entre las arquitecturas de red de **Internet y OSI**
- Conocer la utilidad y funcionamiento del **encapsulamiento de datos**

# Arquitecturas de redes de computadores



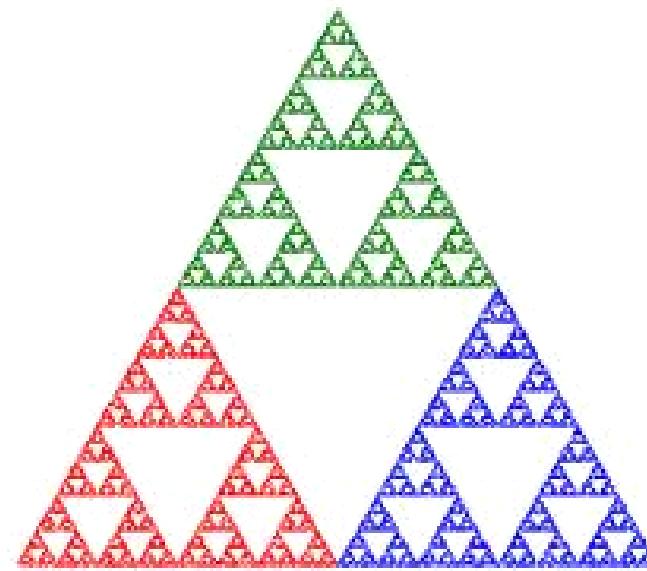
# Arquitectura de redes - Contexto

## Definición

*En el contexto de la arquitectura de computadores aparece el sistema de comunicaciones por red que define, a su vez, una arquitectura en capas*

Aplicaciones
Sistema operativo
Firmware
Hardware: Procesador, Memoria, E/S (Red)

Capas de abstracción en la  
*Arquitectura de computadores*



Aplicación
Transporte
Red
Enlace
Física

Capas de abstracción en una  
*Arquitectura de redes*

# Arquitectura de redes - Diseño

## Problema

*Al diseñar una red de computadores es preciso tener presentes multitud de aspectos, de ahí que se recurra a definir una **Arquitectura** adecuada*

- Medios y modos de transmisión
- Establecimiento y liberación de la conexión
- Formatos de paquetes
- Control de errores: detectarlos y recuperar
- Control de flujo
- Direccionamiento
- Fragmentación y ensamblaje de datos
- Control de paquetes, para evitar pérdidas o desórdenes de estos
- Control de la congestión
- Encaminamiento o enrutado de mensajes: escoger el mejor camino a la hora de enviar un mensaje
- Calidad del servicio
- Multiplexación de información en una misma línea
- Seguridad
- ...



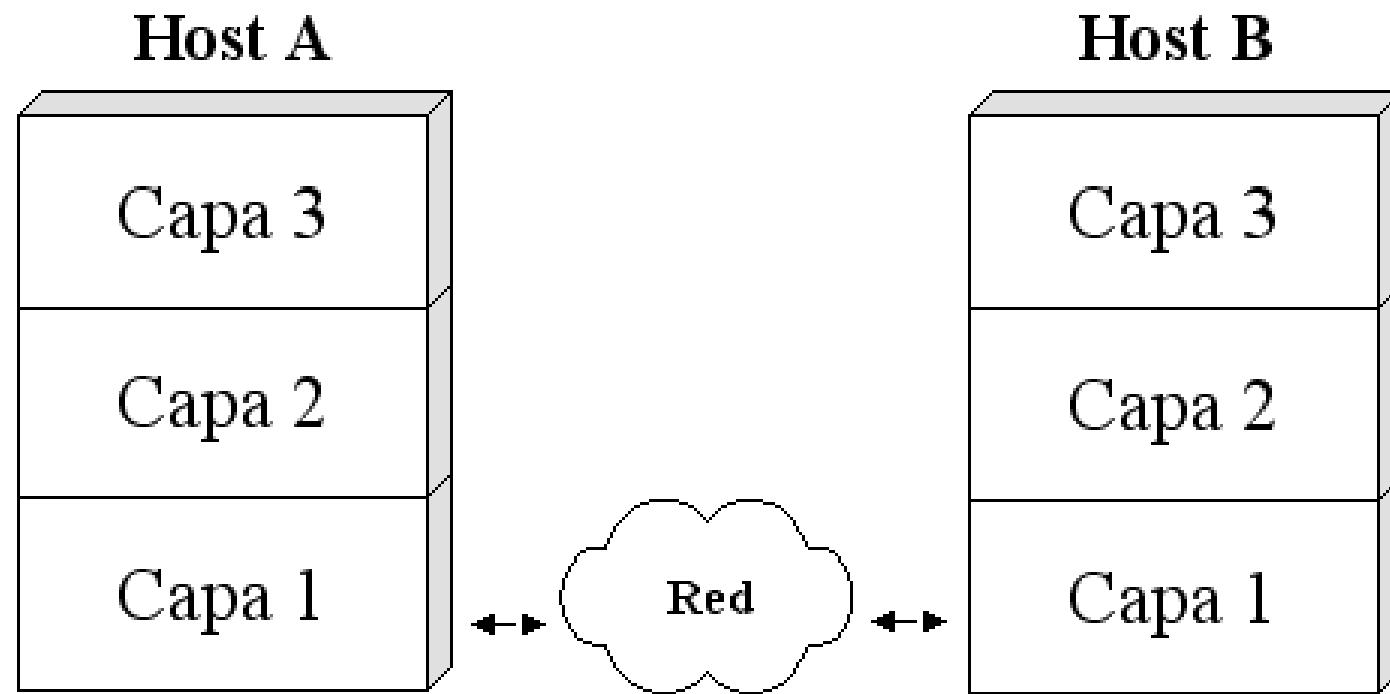
# Arquitectura de redes - Conceptos

- Las redes se desarrollan de forma estructurada en **capas** o **niveles** para facilitar su diseño y mantenimiento ya que
  - Permite identificar y relacionar las complejas piezas del sistema
  - Facilita el mantenimiento y actualización del sistema (sustituir la implementación de una capa sin que afecte al resto del sistema)
- Cada capa realiza una serie de funciones más o menos relacionadas, y necesarias para la transmisión de información (**cohesión**)
- Cada una de ellas se construye sobre una predecesora
- Una capa ofrece una serie de **servicios** a su capa superior, liberando a esta del conocimiento de los detalles internos de su realización.
- Normas habituales en el diseño de una red por capas son:
  - Diseño secuencial de capas
  - Minimizar la información que se transfiere entre ellas (**desacoplamiento**)

# Arquitectura de redes - Ejemplo

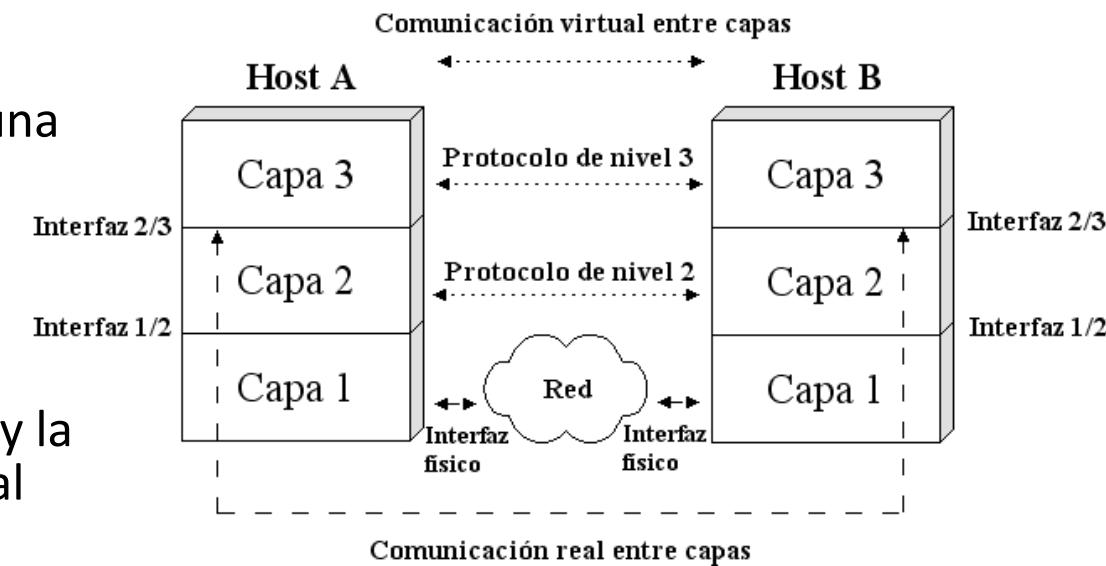
## Descripción

*A y B son dos host (equipos finales) que se comunican entre sí usando una arquitectura de redes compuesta de tres capas*



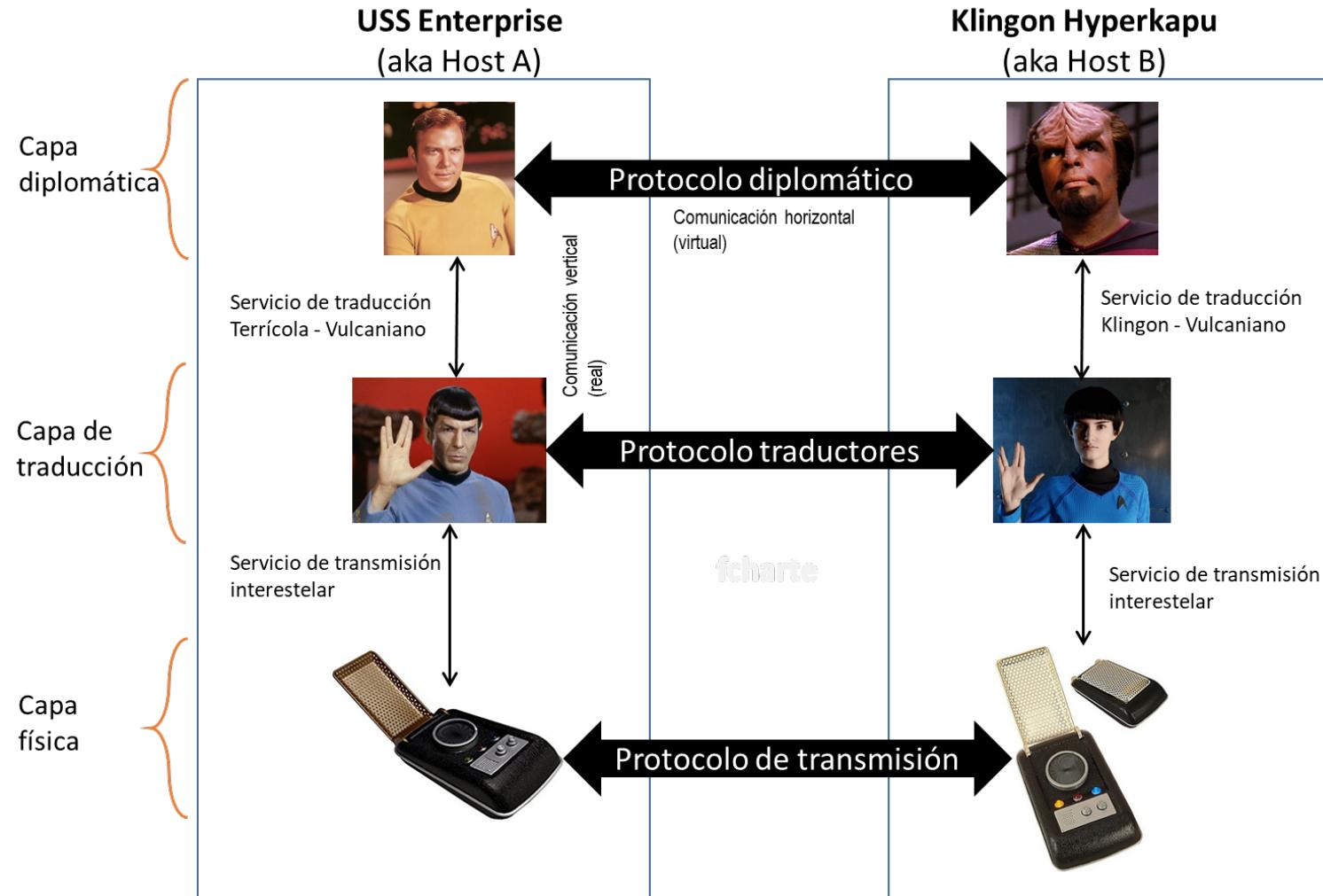
# Relación entre protocolos, servicios e interfaces

- La capa  $n$  de un ordenador, siempre conversará con la capa  $n$  de otro ordenador (**comunicación horizontal/virtual**)
- Se denomina **protocolo** al conjunto de reglas que rigen esta conversación
- Definiremos **interfase** o **interfaz** como la implementación del conjunto de **servicios** que hay entre dos capas, y que define una serie de primitivas y reglas de comunicación
- Una capa hará uso de un servicio llamando a las **primitivas** implementadas en este
- La conversación mantenida mediante un protocolo es **virtual** y la conversación **real** atravesará todas las capas desde la capa  $n$  al medio físico y de este a la capa  $n$  del otro ordenador
- Un buen diseño de esta interfaz minimiza la información que se transfiere entre capas y **simplifica la sustitución** de una capa en un momento dado (p.e. reemplazar las líneas telefónicas por canales satélite)



# Arquitectura de red imaginaria

*Se emplean tres protocolos distintos, uno por capa existente*



# Arquitectura de redes - Resumen

- Se denomina **arquitectura de red** al conjunto de capas y protocolos, no formando parte de ella los detalles de implementación
- Al conjunto de protocolos definidos también se le conocen como **pila de protocolos**
- Normalmente, para cada arquitectura existen varios protocolos y para cada protocolo varias implementaciones
  - Las **implementaciones cambian** continuamente
  - Los protocolos ocasionalmente se modifican o aparecen nuevos que conviven con los antiguos o los dejan anticuados
  - Sin embargo, la arquitectura **raramente se modifica** una vez definida

# Arquitecturas de red OSI vs Internet



# Modelos de referencia

## Definición

*Se denomina modelo de referencia a las arquitecturas de red aceptados como estándares, ya sean oficiales o de hecho. Los modelos de referencia guían el diseño de múltiples tecnologías de transmisión de información*

## Modelos existentes

### – Modelo de referencia OSI

- El modelo de referencia OSI (*Open Systems Interconnections*) pretende normalizar la conexión de sistemas heterogéneos
- Define las diferentes fases por las que deben pasar los datos para viajar de un dispositivo a otro sobre una red de comunicaciones
- Desarrollado por la ISO entre 1977-83, se basó en la especificación más famosa en redes que era SNA (*Systems Network Architecture*) de IBM

### – Pila de protocolos TCP/IP

- Es el modelo que sigue Internet

# OSI - Capas y su finalidad

- **Capa física:**

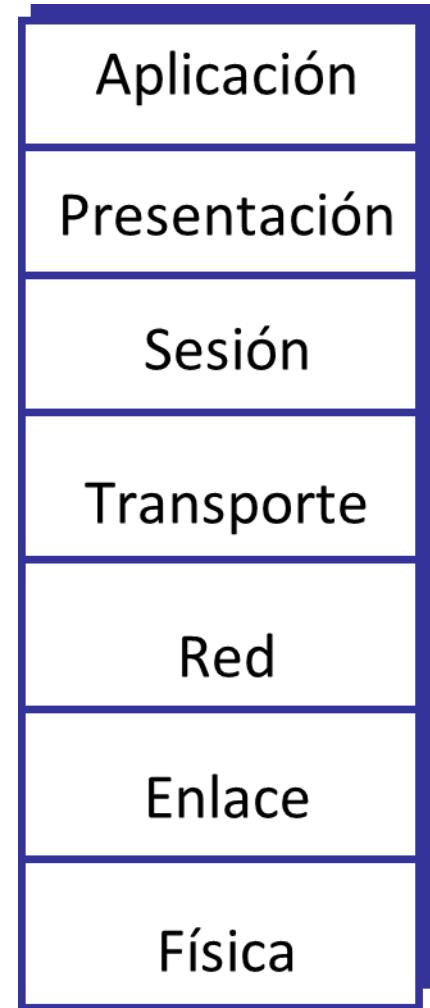
- Objetivo: definición de lo que es un **bit** en una transmisión y transmitirlo
- Define los **medios de transmisión** (conectores, cables, etc)
- Define los **modos de transmisión** (modulación, codificación, voltajes, temporización, etc.)
- Define como establecer y liberar la **conexión**
- **Sincronismo**

- **Capa de enlace:**

- Objetivo: transmisión entre dispositivos directamente conectados
- Definición y reconocimiento de los límites de **tramas** (entramado)
- Control de **errores**
- Control del **flujo** en la comunicación

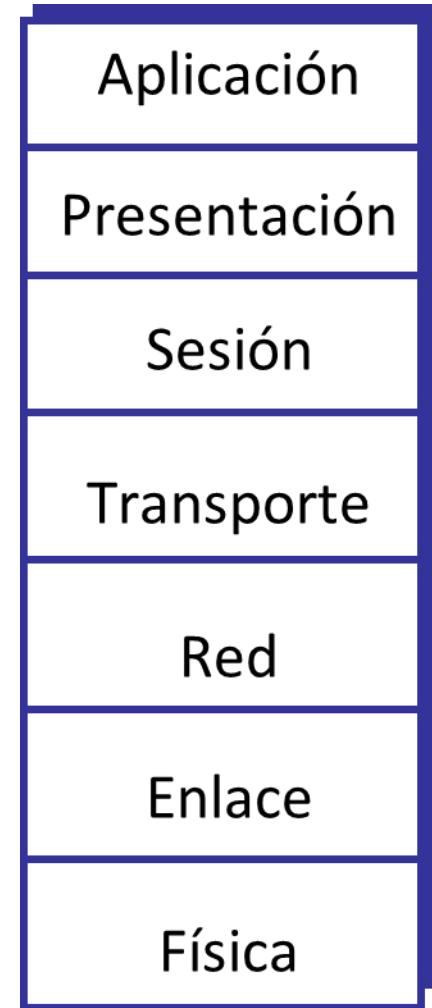
- **Capa de red:**

- Objetivo: transmisión entre dispositivos no directamente conectados
- Enrutamiento de **paquetes** en la subred: definición del camino para llegar al destino
- Control de la **congestión**
- Interconexión de **redes heterogéneas**



# OSI - Capas y su finalidad

- **Capa de transporte:**
  - Objetivo: definición de la comunicación extremo a extremo (ocultando la subred)
  - Control de **errores**
  - Control de **flujo**
  - **Multiplexación** de varias conexiones
- **Capa de sesión:**
  - Objetivo: establecimiento de “sesiones” entre usuarios de diferentes máquinas
  - Gestiona el control del **diálogo** (utilizando *tokens*)
  - Establece puntos de **sincronización**, pudiendo recuperar partes de transmisiones, sin retransmitir todo en caso de error
- **Capa de presentación:**
  - Objetivo: definir aspectos sintácticos y semánticos de la información
  - Define estructuras de **representación abstractas** para comunicar ordenadores que utilizan representación interna diferente
  - **Compresión** de datos y aplicar técnicas de criptografía
- **Capa de aplicación:**
  - Diseño de las aplicaciones que al final utiliza un usuario



# Modelo de referencia TCP/IP (Internet)

## Arquitectura

*Más simple que el modelo OSI, fusionando las capas de **sesión**, **presentación** y **aplicación** en una **única capa de aplicación***

## Variaciones

Existen dos variaciones, usándose una u otra según autor:

### – Cinco capas

- Conserva las capas inferiores del modelo OSI
- Se distingue entre capa de **Enlace** y **Física**
- Será la versión que usemos principalmente en la asignatura

### – Cuatro capas

- Fusiona las capas de **Enlace** y **Física** en una sola
- La capa, llamada **Host-Red**, asume todas las funciones de las capas de **Enlace** y **Física** del modelo OSI



# Comparativa OSI - TCP/IP

- **Diferente objetivos:**

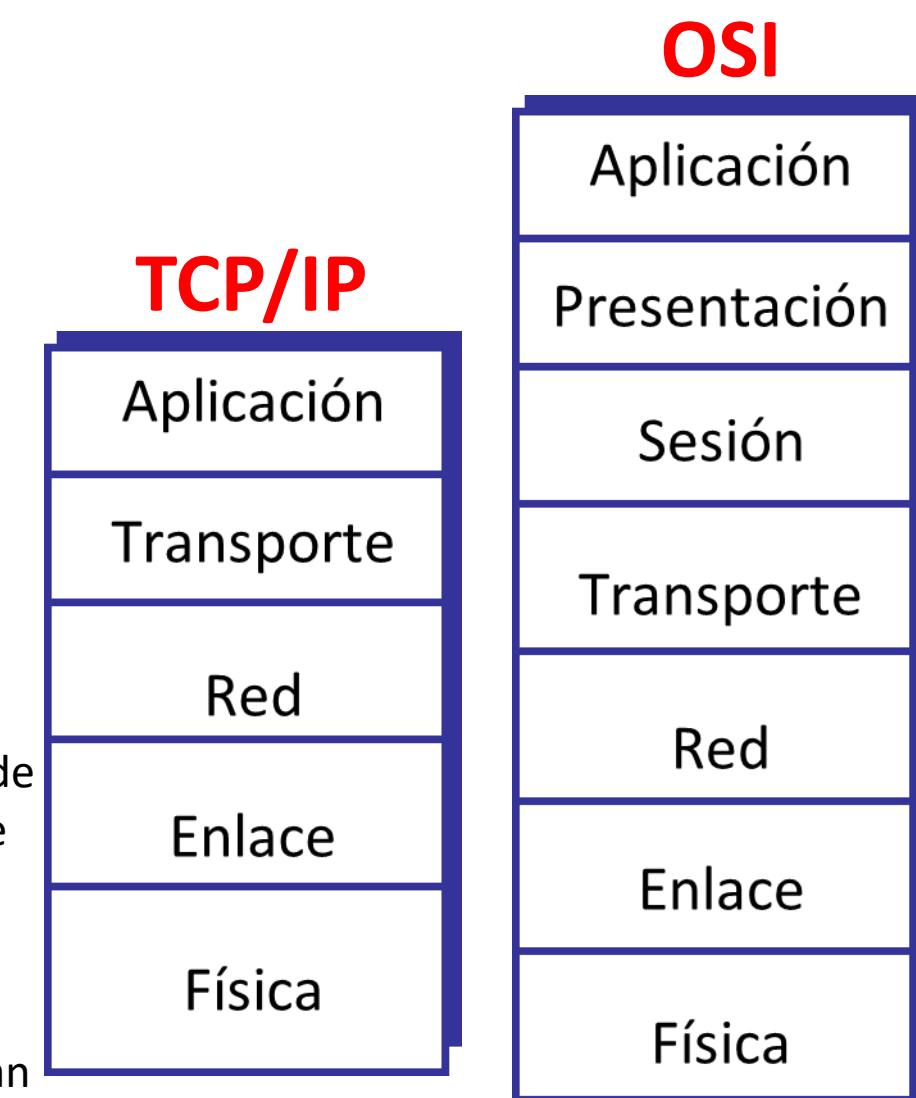
- OSI es una arquitectura más académica y teórica, en la que **primero se define el modelo** y después los protocolos, mientras que con TCP/IP el proceso fue a la inversa

- **Diferentes características:**

- OSI, como modelo académico, es **más formal y abstracto** que TCP/IP, permite explicar otras redes, y hace una clara distinción entre servicios, interfaces y protocolos
- En contraposición, **TCP/IP es un modelo más práctico** y no distingue tan claramente las funciones de algunas capas

- **Diferente desarrollo:**

- TCP/IP se desarrolla libremente (sus especificaciones son libres – **RFCs**), de forma distribuida y a gran velocidad. Está orientado a la interconexión de redes. Además **no define nada** en sus capas *hardware* por lo que es independiente de la tecnología
- OSI presenta mayor complejidad en su modelo con elementos a veces repetitivos y confusos. Además las organizaciones involucradas demandan mucha burocracia

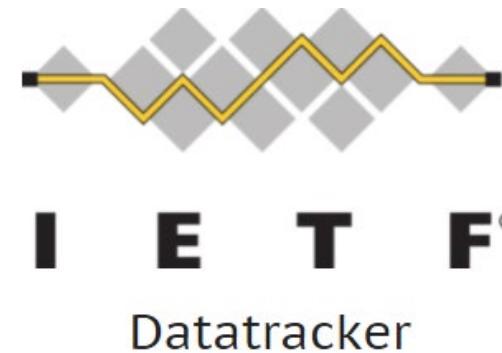


## La documentación oficial de Internet

*Los RFC son documentos accesibles de forma pública y representan la documentación oficial sobre el funcionamiento de Internet y sus tecnologías*

### Cómo consultar un RFC

- **Abre la página** <https://datatracker.ietf.org> e introduce el número de RFC o bien el protocolo o tecnología que quieras consultar
- Prueba a buscar el **número 1180** y haz clic en el único resultado que aparece
- El RFC 1180 es un tutorial del funcionamiento de TCP/IP y **lo puedes descargar** en múltiples formatos
- Examina el apartado *2.1. Basic Structure*, en el que se expone la arquitectura de capas de TCP/IP



# Implementación de capas en un ordenador

## Descripción

*La funcionalidad de cada una de las capas TCP/IP es implementada ya sea en hardware (controlador de red) o software (controladores, SO)*

## Esquema de un host PC

### – Adaptador de red (hardware)

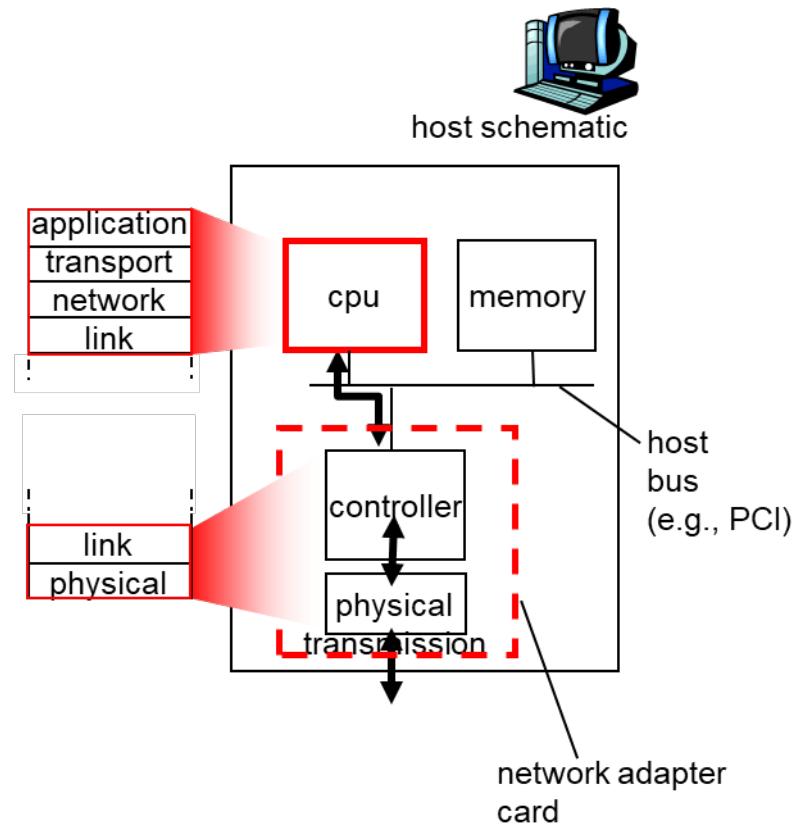
- Se ocupa de las funciones más cercanas al medio físico
- Suele implementar las capas física y de enlace

### – Controladores/SO (software)

- Se ocupan de las funciones de capas intermedias como la de red y transporte

### – Aplicaciones (software)

- Implementan los protocolos de más alto nivel, asociados a la capa de aplicación por regla general



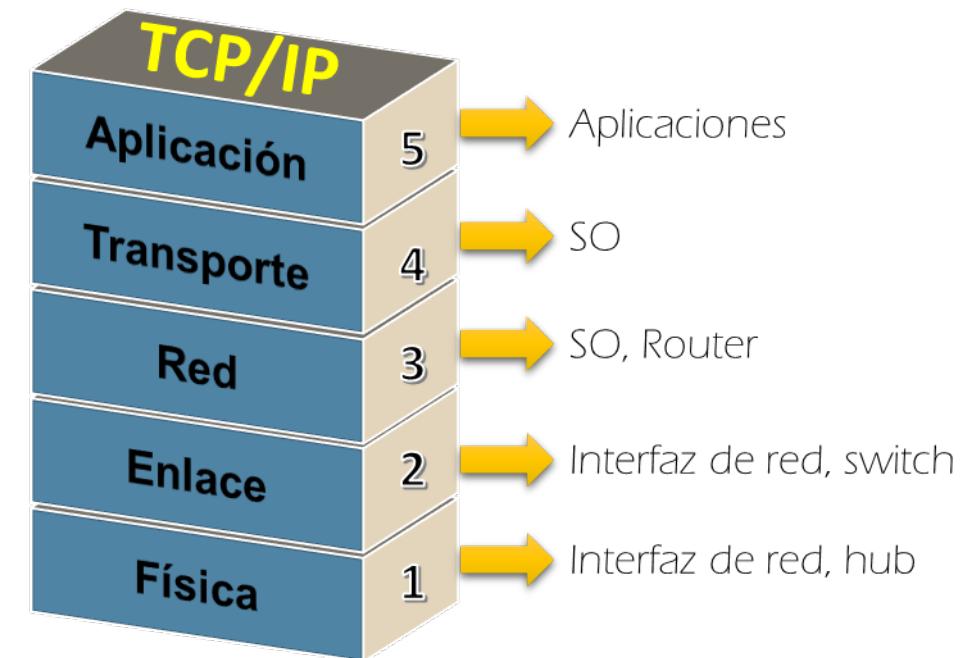
# Implementación de capas en equipamiento de red

## Descripción

No solo los ordenadores han de implementar las capas TCP/IP, también el equipamiento intermedio de red ha de contar con las capas necesarias para realizar su trabajo

## Esquema general

- Capas físicas
  - Todo equipamiento de red ha de contar con ella
  - En un PC está en la interfaz de red, también cuentan con ella los *hub*, *switches* y *routers*
- Capa de enlace
  - Implementada por los PC en su interfaz de red, los *switches* y *routers*
- Capa de red
  - Cuentan con ella los hosts finales (PC) y los *routers*
- Capas superiores
  - Únicamente son precisas en los host finales, como los PC



# Protocolos en una arquitectura de red



# Protocolo - Definición y ejemplo

## Definición

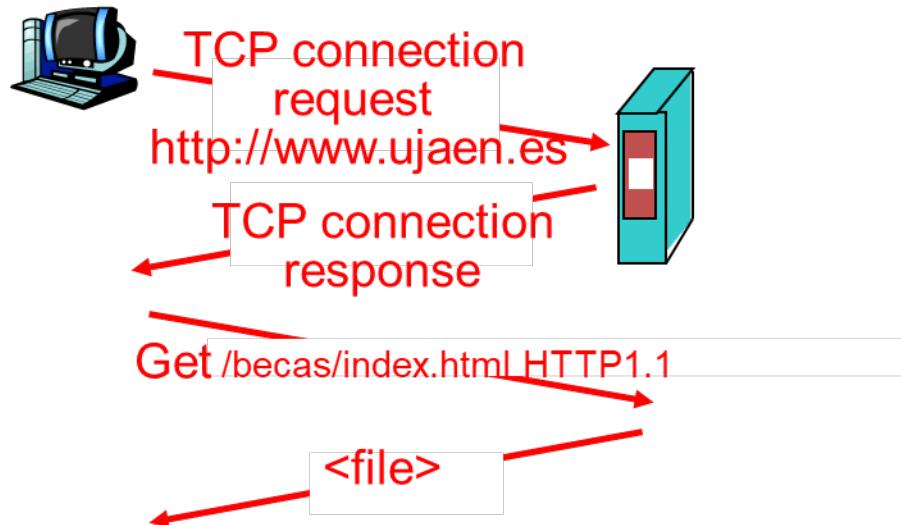
*Se denomina protocolo al conjunto de reglas que rigen la comunicación a nivel de una capa concreta entre dos host conectados*

### Protocolo humano



tiempo  
↓

### Protocolo de Internet



# Protocolo - Conceptos

## Definición ampliada

*El protocolo establece las reglas de conversación: sintaxis, semántica, temporización, etc., entre los pares de capas, determinando qué mensajes se envían, quién los envía, en qué momento, etc.*

## Conceptos

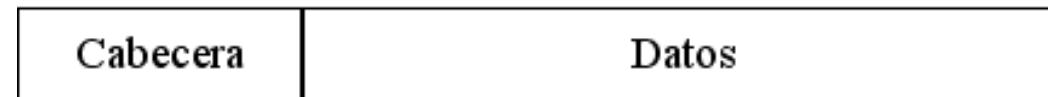
- **PDU** (*Protocol Data Unit*). Unidad de datos del protocolo: denominación dada a los mensajes que envía un protocolo
- **SDU** (*Service Data Unit*). Unidad de datos del servicio: además del mensaje (PDU) a enviar también incluyen una **cabecera** con datos de control
- La denominación PDU/SDU puede ser más específica **dependiendo de la capa** a la que se haga referencia:
  - **NPDU** (*Network PDU*): PDU de la capa de red
  - **TPDU** (*Transport PDU*): PDU de la capa de transporte

# PDU - Estructura

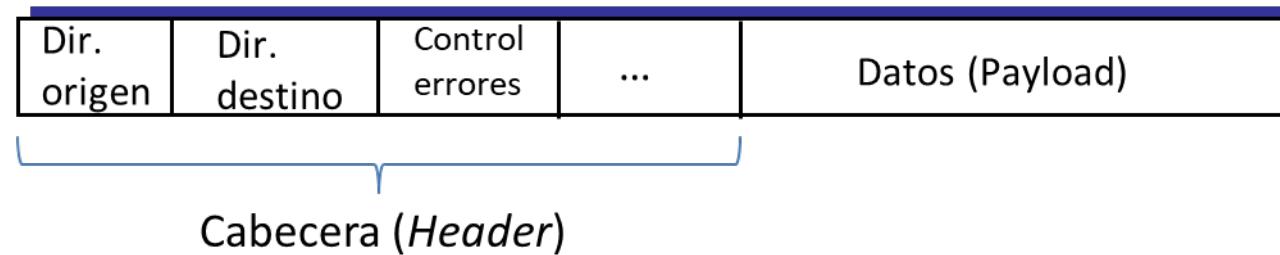
## Descripción

*La PDU de un protocolo se divide en dos partes: cabecera y datos*

PDU



*Los campos de la cabecera suelen implementar **funciones del protocolo***



*El **payload** es la parte útil de la PDU desde la perspectiva del usuario*

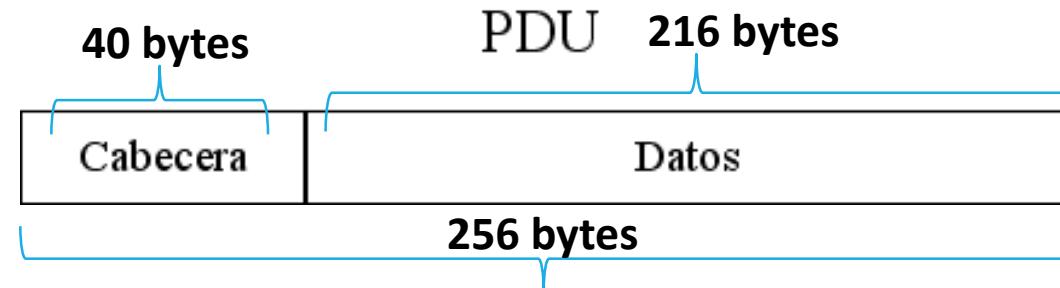
# Protocolo - Eficiencia

## Definición

*La eficiencia de un protocolo se mide como la relación entre la longitud del campo de datos en su PDU y la cantidad de información transmitida*

## Razonamiento

- **Utilidad.** La finalidad esencial de un protocolo es facilitar el **transporte de datos**
- **Implementación.** Para realizar su trabajo, los protocolos precisan campos de control que deben incluirse en la PDU (cabecera)
- **Objetivo.** Minimizar la información de la cabecera en la medida de lo posible, reduciendo así el *overhead*
- **Ejemplo**



$$40 / 256 = 0.15625$$

15% de overhead aprox.

# Servicios en una arquitectura de red



# Servicio - Conceptos

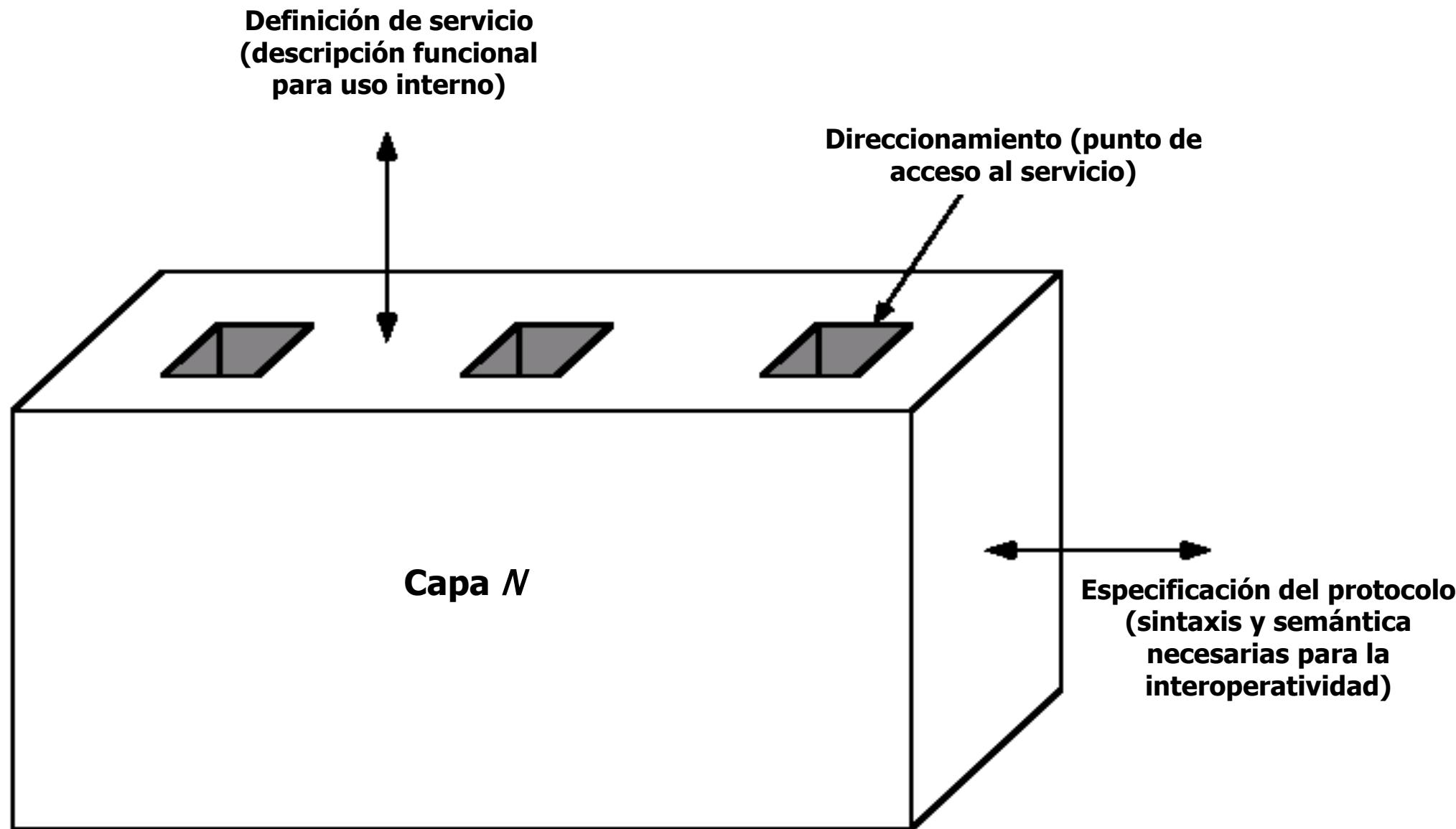
## Definición

*Un servicio es ofrecido por una capa de la arquitectura de red a la capa inmediatamente superior a ella*

## Conceptos

- **Primitivas.** Mecanismo por el que un servicio ofrece sus funciones a otra capa:
  - establece\_conexión(destino, puerto, ...)
  - envía\_datos(destino, puerto, datos, ...)
  - libera\_conexión(destino, ...)
- **SAP** (*Service Access Point*). Punto de acceso al servicio: dirección a través de la cual la capa superior accede a los servicios, por ejemplo puertos TCP o UDP
- Cada capa de la arquitectura cuenta con sus propios SAP:
  - **TSAP** (*Transport SAP*): direcciones de la capa de transporte
  - **NSAP** (*Network SAP*): direcciones de la capa de red

# Protocolos, servicios y puntos de acceso - Relación



# Servicio - Categorización

## Definición

*Los servicios ofrecidos pueden agruparse en dos categorías*

## Tipos de servicios

- Servicio orientado a conexión (**CONS**, *Connection Oriented Network Service*):
  - Para enviar información hay que **establecer una conexión**, enviar información y liberar la conexión
  - La información sigue toda el mismo camino y llega en el mismo **orden** en el que sale del emisor
  - Analogía del sistema telefónico
  - Normalmente a una conexión establecida se le denomina *circuito* (virtual o no, ya lo veremos)
- Servicio no orientado a conexión (**CLNS**, *ConnectionLess Network Service*)
  - La información se suele **dividir en mensajes** poniéndole a cada uno una dirección de destino.
  - Estos mensajes pueden seguir cualquier camino
  - Los mensajes **pueden desordenarse**
  - Analogía del servicio de correos

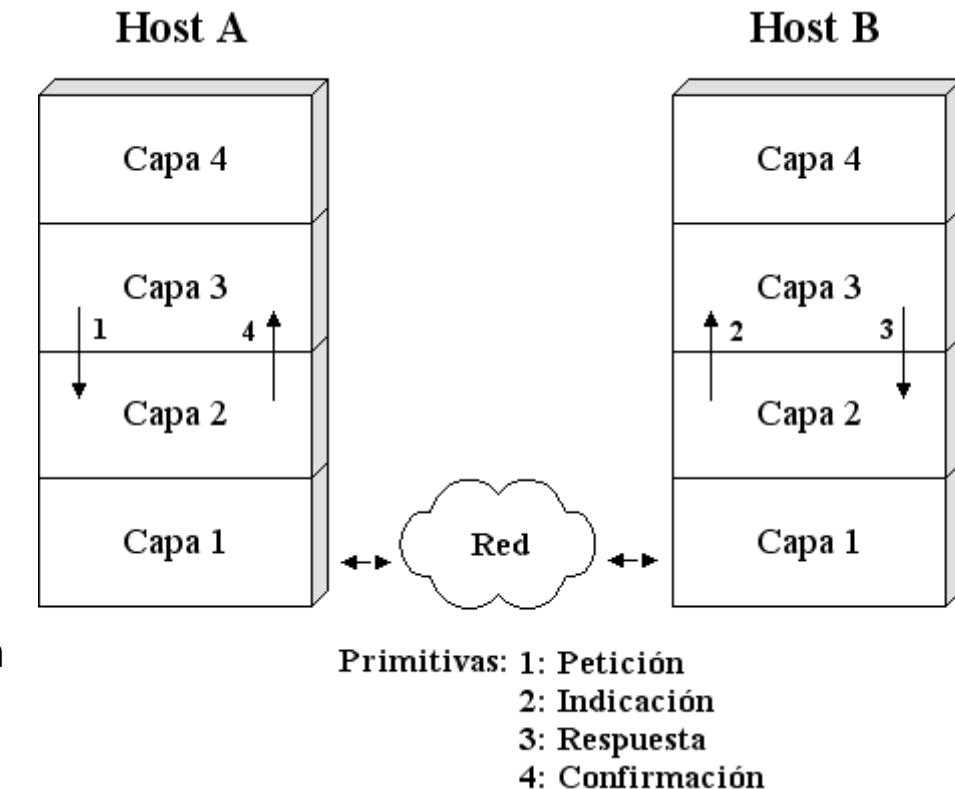
# Servicio - Primitivas

## Definición

Las primitivas indican al servicio que debe **efectuar una acción** o bien **notifican la acción** tomada por una entidad para

## Tipos de primitivas

- Existen **cuatro tipos** de primitivas principales:
  - **Petición o solicitud (request)**: una entidad pide que se realice un trabajo
  - **Indicación (indication)**: una entidad es informada acerca de un evento
  - **Respuesta (response)**: una entidad responde a un evento
  - **Confirmación (confirm)**: una entidad va a ser informada de su solicitud
- Dentro de un servicio las diferentes **fases** pueden ser:
  - **Confirmadas**: existen primitivas de petición, indicación, respuesta y confirmación
  - **No confirmadas**: existen primitivas solo de petición e indicación
- **Ejemplos**: 1) el establecimiento de la conexión suele ser confirmado, 2) la transmisión de datos puede serlo o no



# Servicio - Ejemplo de servicio CONS (FTP)

## Cliente

1. CONNECT.request -> Juan pulsa botón de conectar

4. CONNECT.confirm -> La aceptación de conexión llega al cliente  
5. DATA.request -> Enviamos datos (archivo solicitado)

8. DATA.indication -> Nos llegan los datos  
9. DISCONNECT.request -> Nos desconectamos

## Servidor

2. CONNECT.indication -> Al servidor le llega la petición de conexión  
3. CONNECT.response -> El servidor responde con la aceptación de conexión al cliente

6. DATA.indication -> La petición llega al servidor  
7. DATA.request -> El servidor nos envía el archivo

10. DISCONNECT.indication -> El servidor recibe indicación de desconexión

tiempo  
↓

# Comunicación lógica y física



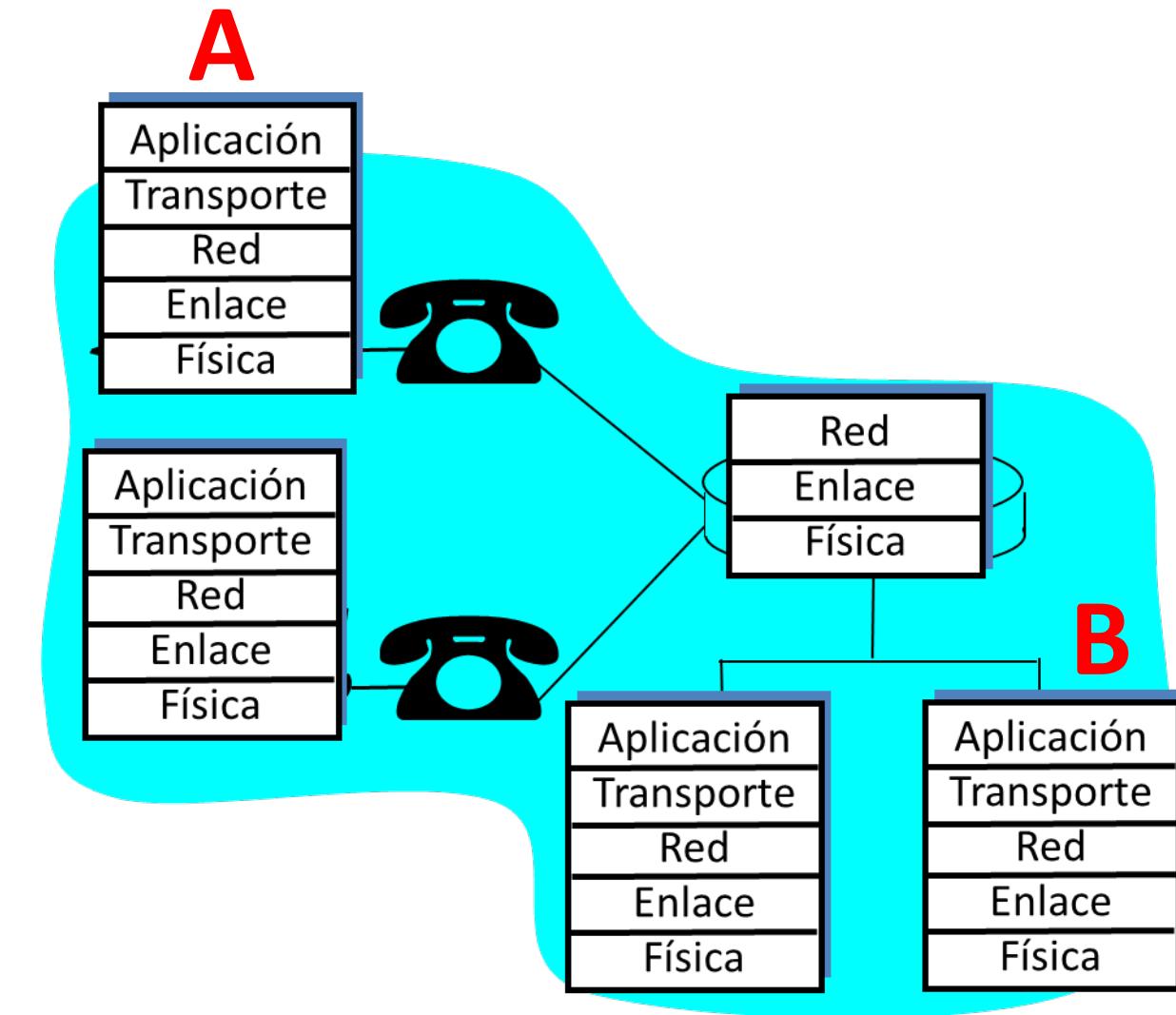
# Comunicación lógica vs física

## Definición

*La comunicación entre dos host en una red de ordenadores puede verse desde una perspectiva lógica o física*

## Ejemplo

- Un programa en el **host A** quiere comunicarse con otro del **host B**
- El primero puede ser un navegador web y el segundo un servidor web
- El mensaje enviado desde la capa de aplicación de A debe transferirse hasta alcanzar la capa de aplicación de B



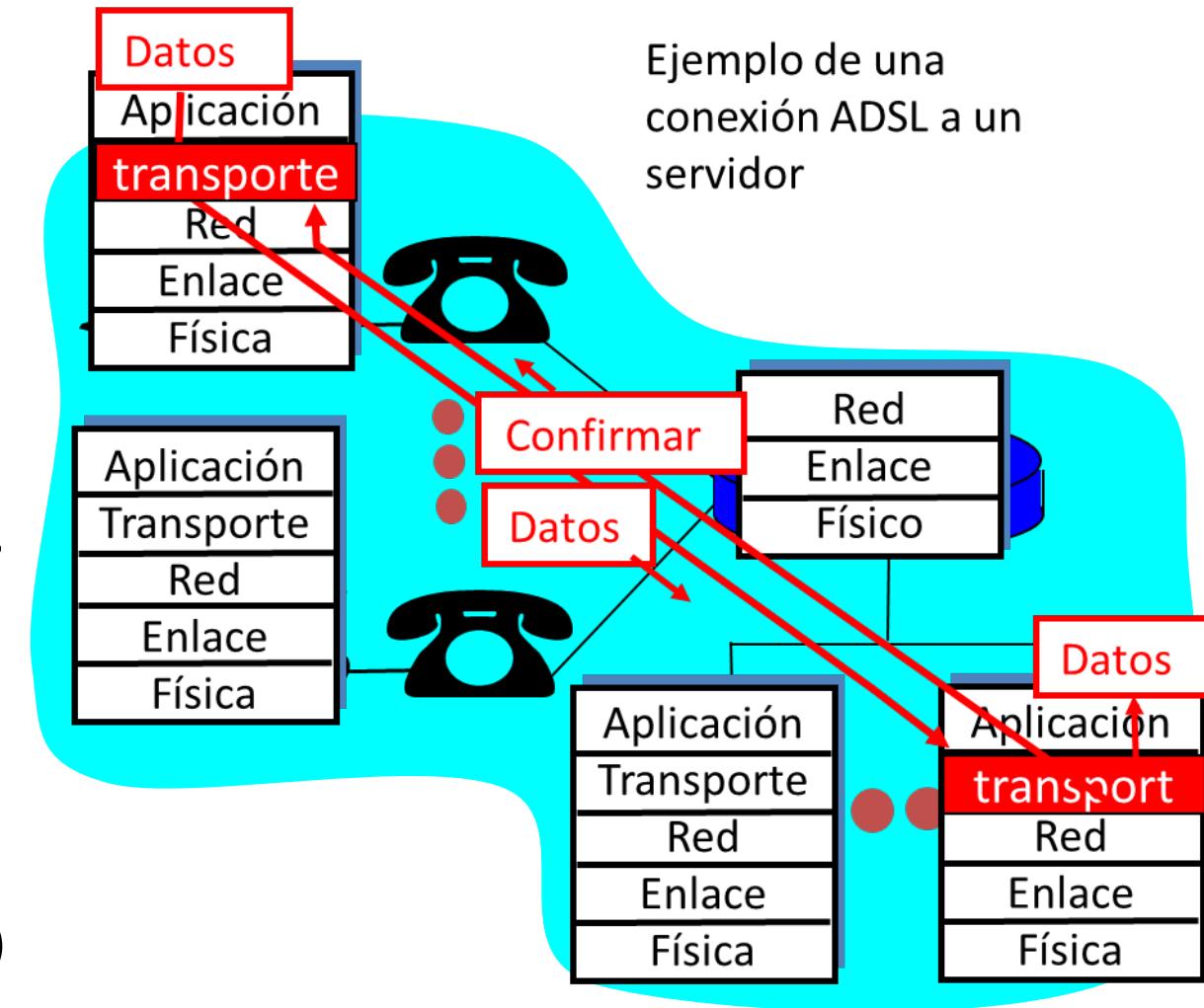
# Comunicación lógica

## Propósito

*Anализar relación entre protocolos y servicios en una comunicación por red*

## Ejemplo

- Comunicación entre capas de aplicación y transporte
  - La capa de aplicación usa las primitivas de la capa de transporte para acceder a sus servicios
  - La llamada a una primitiva suele implicar la generación y envío de un mensaje de un protocolo de la capa que ofrece el servicio
- Comunicación lógica/virtual
  - Para la capa de aplicación del primer host el intercambio de mensajes se produce con la misma capa del segundo host (abajo a la dcha.)



# Comunicación física

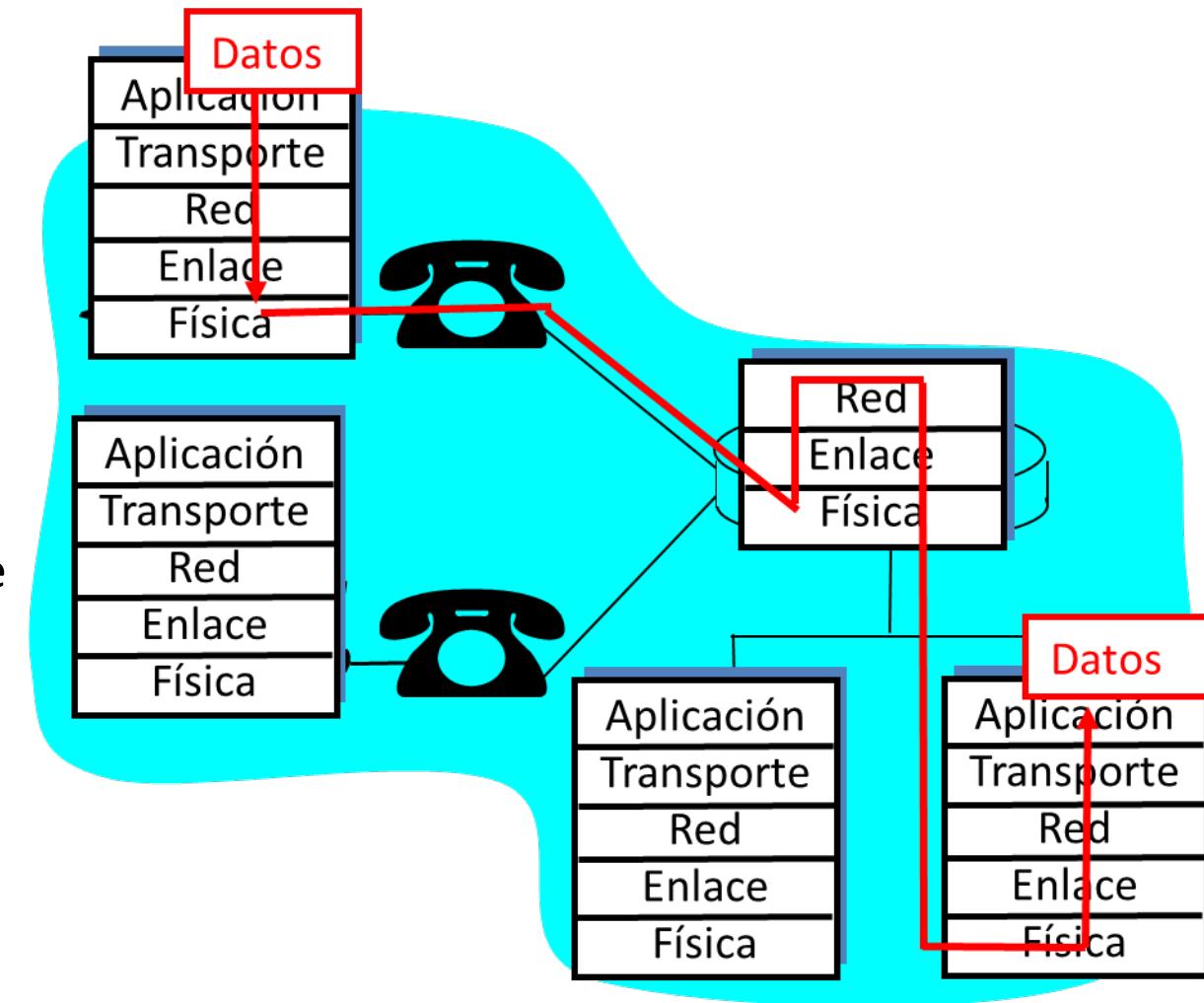
## Propósito

*Analizar relación entre protocolos y servicios en una comunicación por red*

## Ejemplo

### – Comunicación física/real

- La capa de aplicación usa las primitivas de la capa de transporte
- La capa de transporte recurre a las primitivas de la capa de red
- Al alcanzar la capa física la información se transmite por un medio hasta a otro elemento
- En el host de destino se sigue el proceso inverso, de forma que la información recibida por la capa física pasa a la de enlace, de esta a la red y así sucesivamente hasta la de aplicación



# Proceso de encapsulamiento de datos



# Encapsulamiento de datos

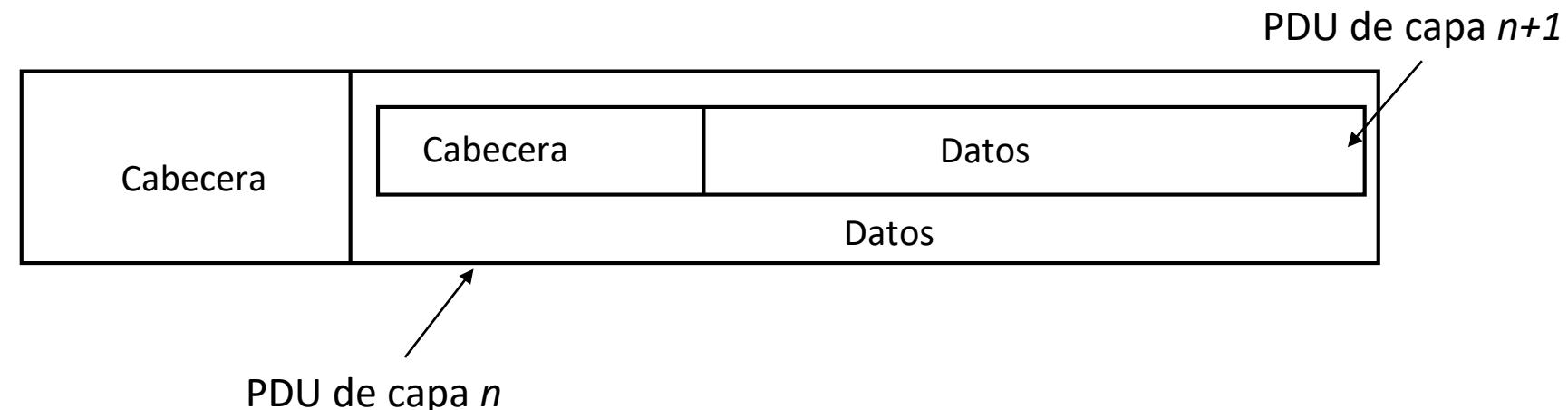
## Definición

*El proceso de encapsulamiento en una arquitectura de red consiste en empaquetar la PDU de cada capa en la de la capa inferior y, análogamente, llevar a cabo el proceso inverso en el host de destino (desencapsulamiento)*

## Finalidad

*El encapsulamiento de datos permite que los servicios de cada capa operen exclusivamente sobre los campos de su cabecera, tratando las cabeceras de capas superiores como parte del paquete de datos*

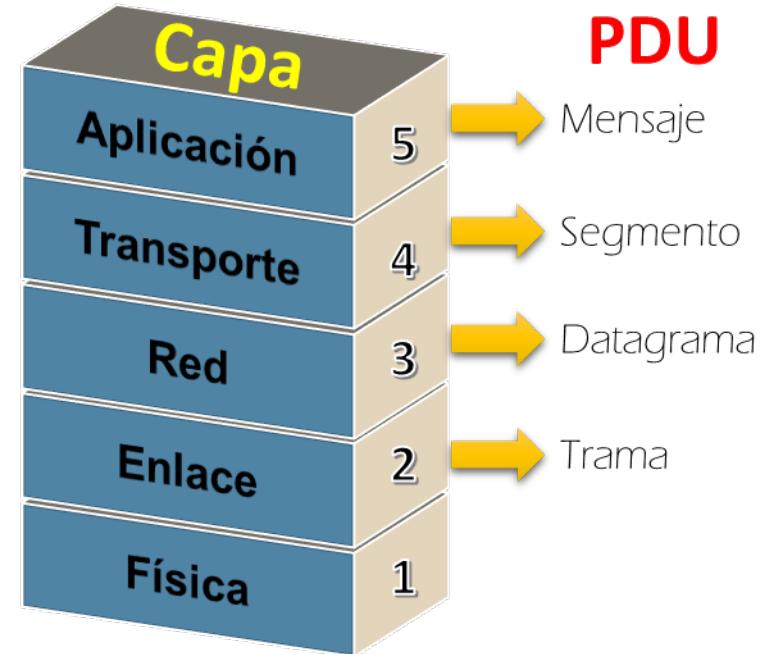
## Diagrama



# Encapsulamiento de datos

## Pasos

- **Encapsulamiento:** se produce en el host de origen o emisor
  - El **mensaje de la capa de aplicación**, junto con su cabecera, se empaqueta en el *payload* de la PDU de la capa de transporte
  - El **segmento de la capa de transporte**, junto con su cabecera, se empaqueta en el *payload* de la PDU de la capa de red
  - El **datagrama de la capa de red**, junto con su cabecera, se empaqueta en el *payload* de la PDU de la capa de enlace
  - La **trama de la capa de enlace**, junto con su cabecera, se empaqueta y envía a la capa física
- **Desencapsulamiento:** se produce en el host de destino o receptor
  - Se lleva a cabo el proceso inverso, extrayendo la trama, de esta el datagrama, etc.



# Cuestiones clave de este tema



# Cuestiones clave

## Qué deberías saber

*Al inicio de este tema se planteaban unos objetivos específicos que deberían permitirte **responder a las siguientes cuestiones** clave*

## Cuestiones

- ¿Qué es una arquitectura de red y cuáles son los modelos existentes?
- ¿Cuál es la distribución de funcionalidad en las capas de TCP/IP?
- ¿Qué son los protocolos, interfaces y servicios y la relación entre ellos?
- ¿Cómo se lleva a cabo la comunicación entre los *host* de una red?
- ¿Cuál es el funcionamiento de la comunicación horizontal/virtual y vertical/real y el encapsulamiento de datos?

# Material adicional

## Descripción

*En las siguientes diapositivas encontrarás material complementario que te ayudará a profundizar en algunos aspectos concretos de los conceptos abordados esta semana*

## Contenidos

- **Encapsulamiento de datos** a través de un ejemplo paso a paso en estas mismas diapositivas y complementado con un vídeo explicativo
- **Modelos OSI y TCP/IP** comparados con detalles sobre las diferentes capas que define cada arquitectura

# Ejemplo de encapsulamiento de datos paso a paso

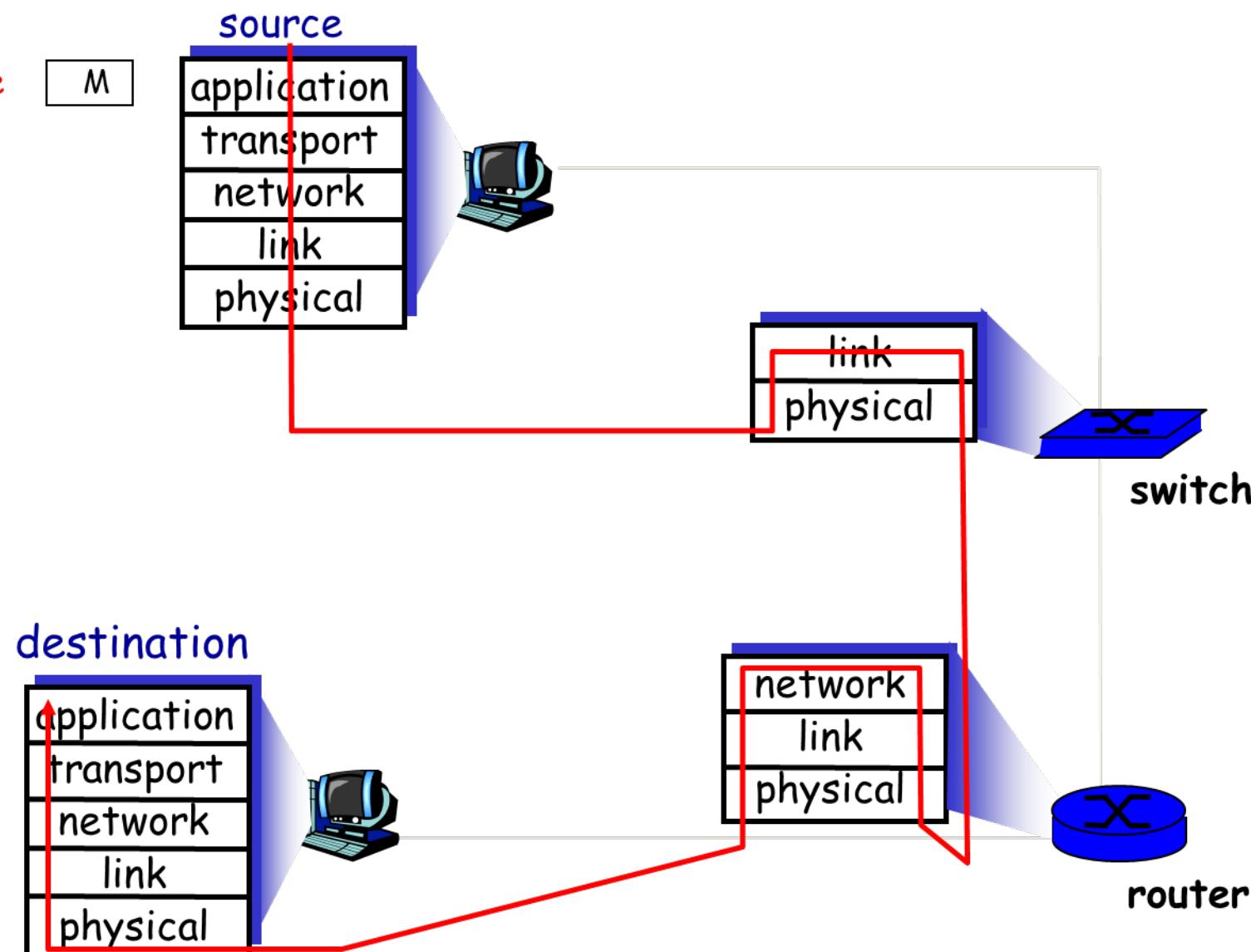


# Encapsulamiento de datos - Ejemplo

Paso 1

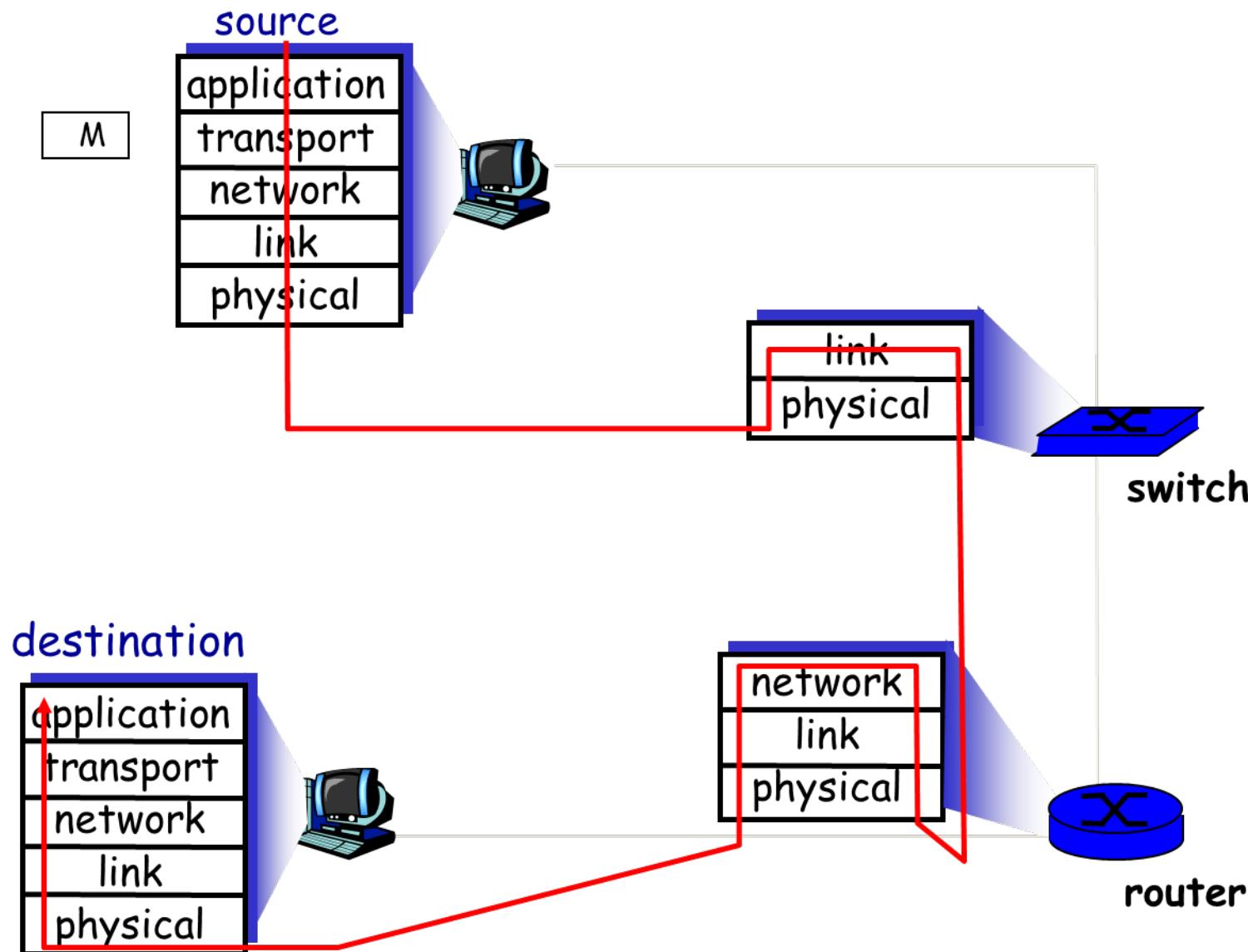
message

M



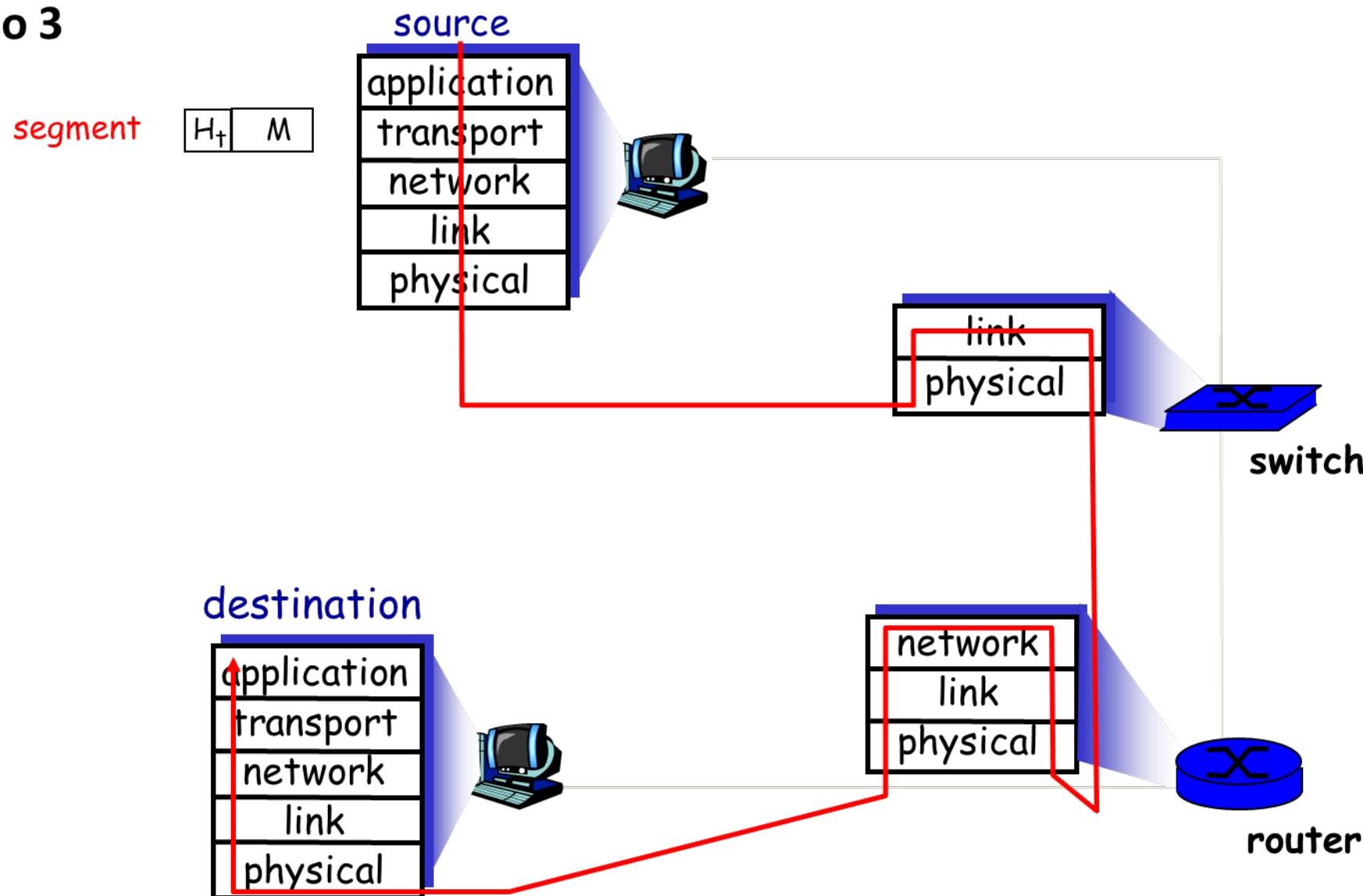
# Encapsulamiento de datos - Ejemplo

Paso 2



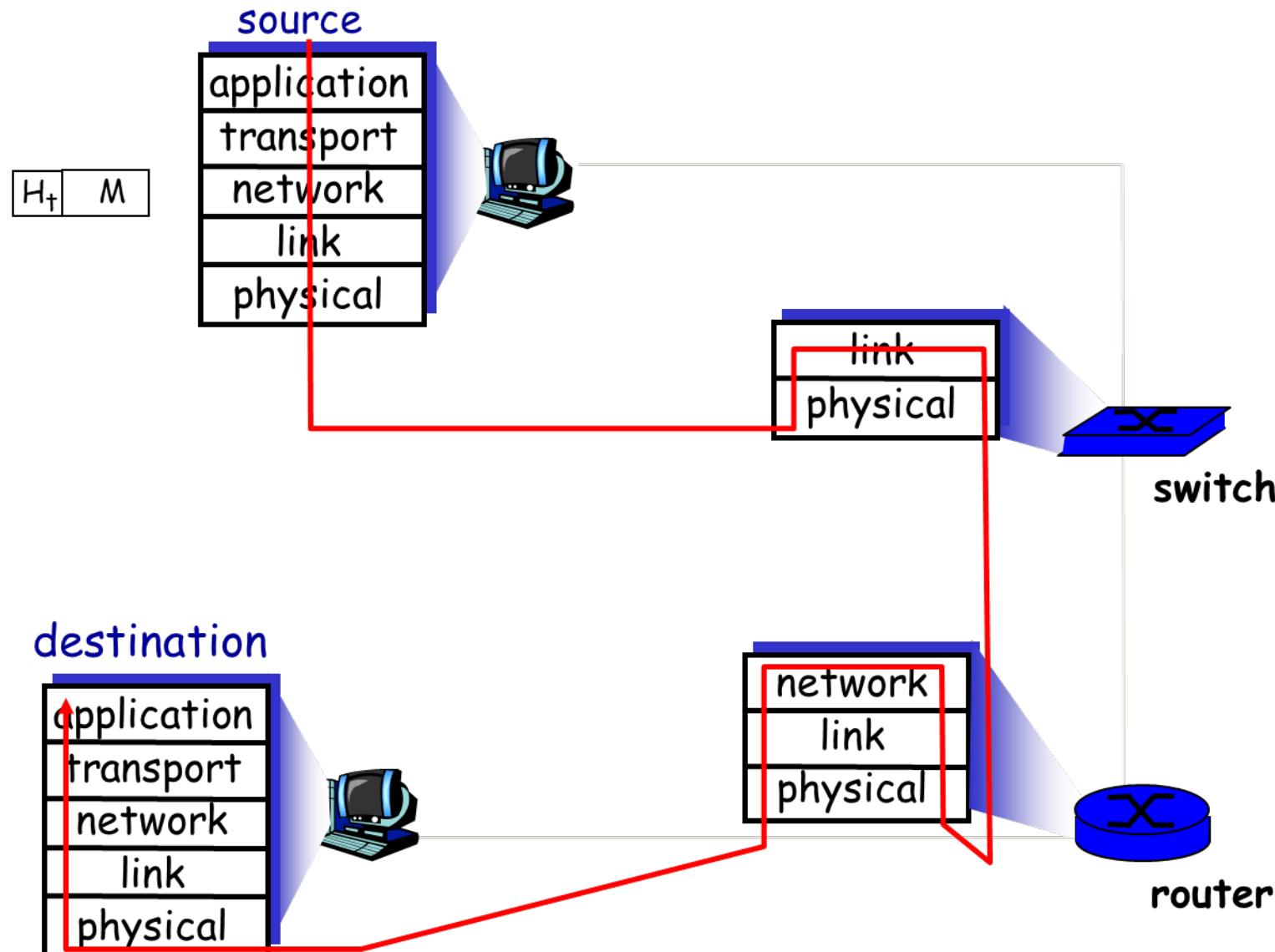
# Encapsulamiento de datos - Ejemplo

## Paso 3



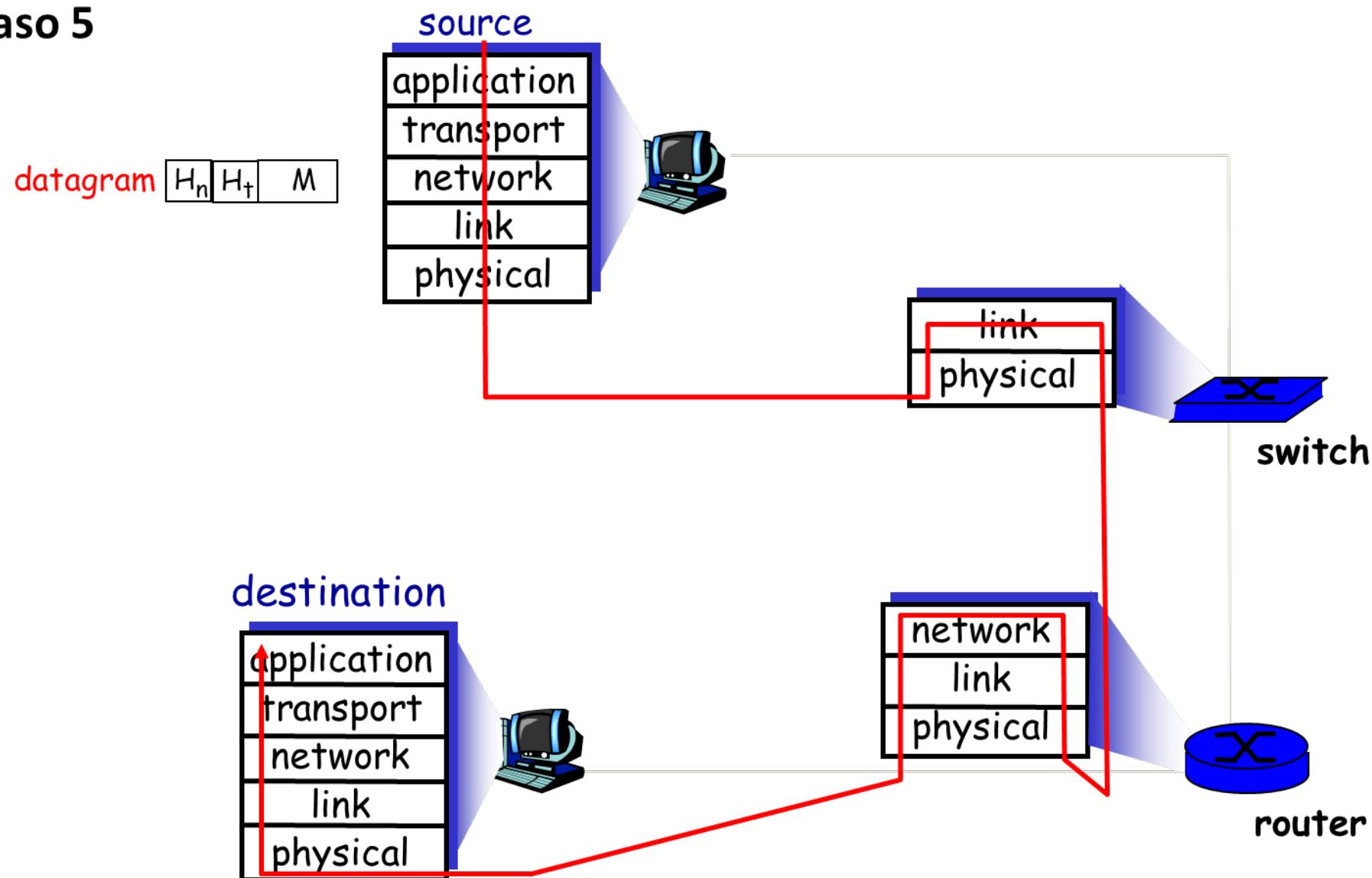
# Encapsulamiento de datos - Ejemplo

Paso 4



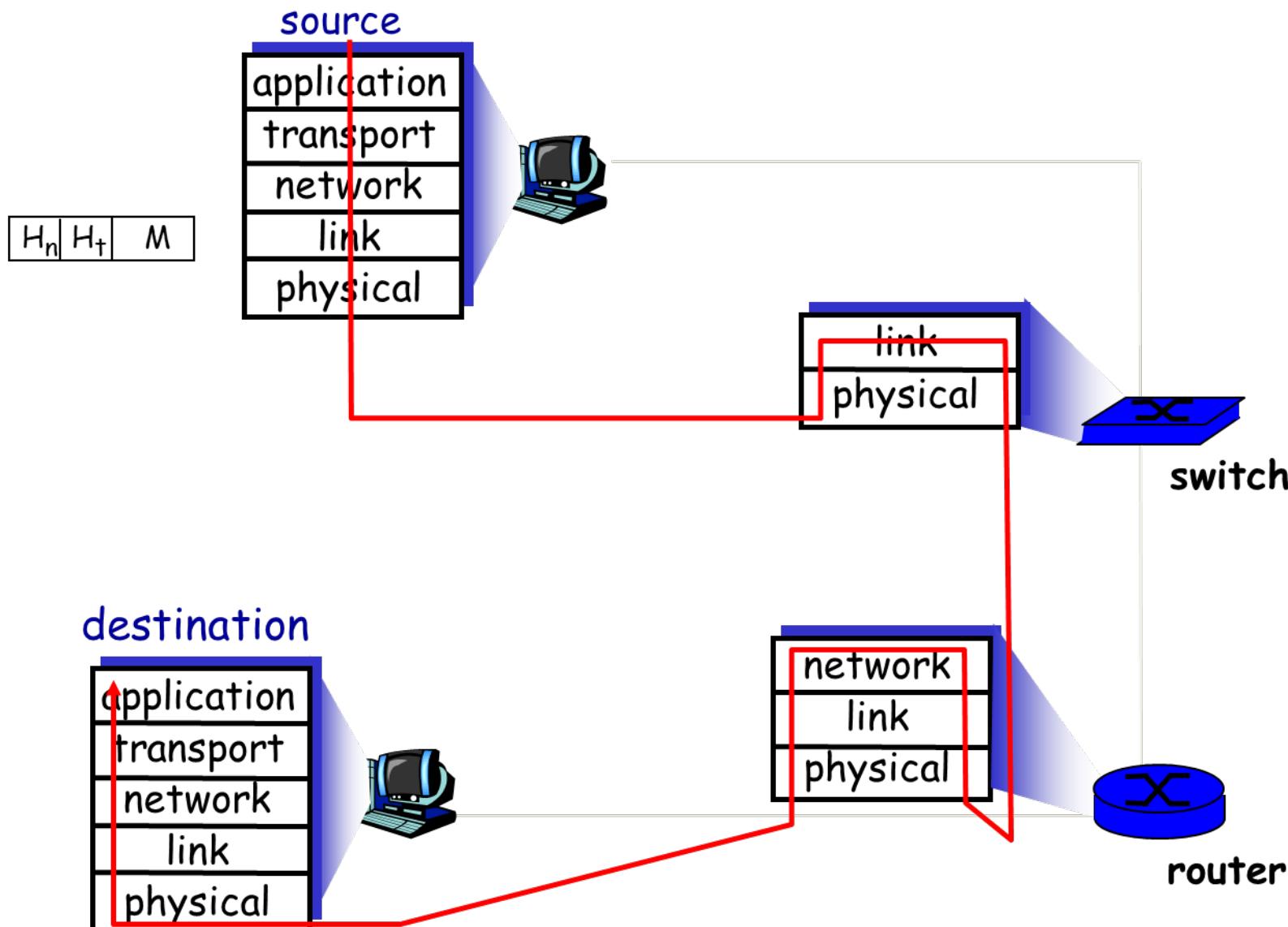
# Encapsulamiento de datos - Ejemplo

Paso 5



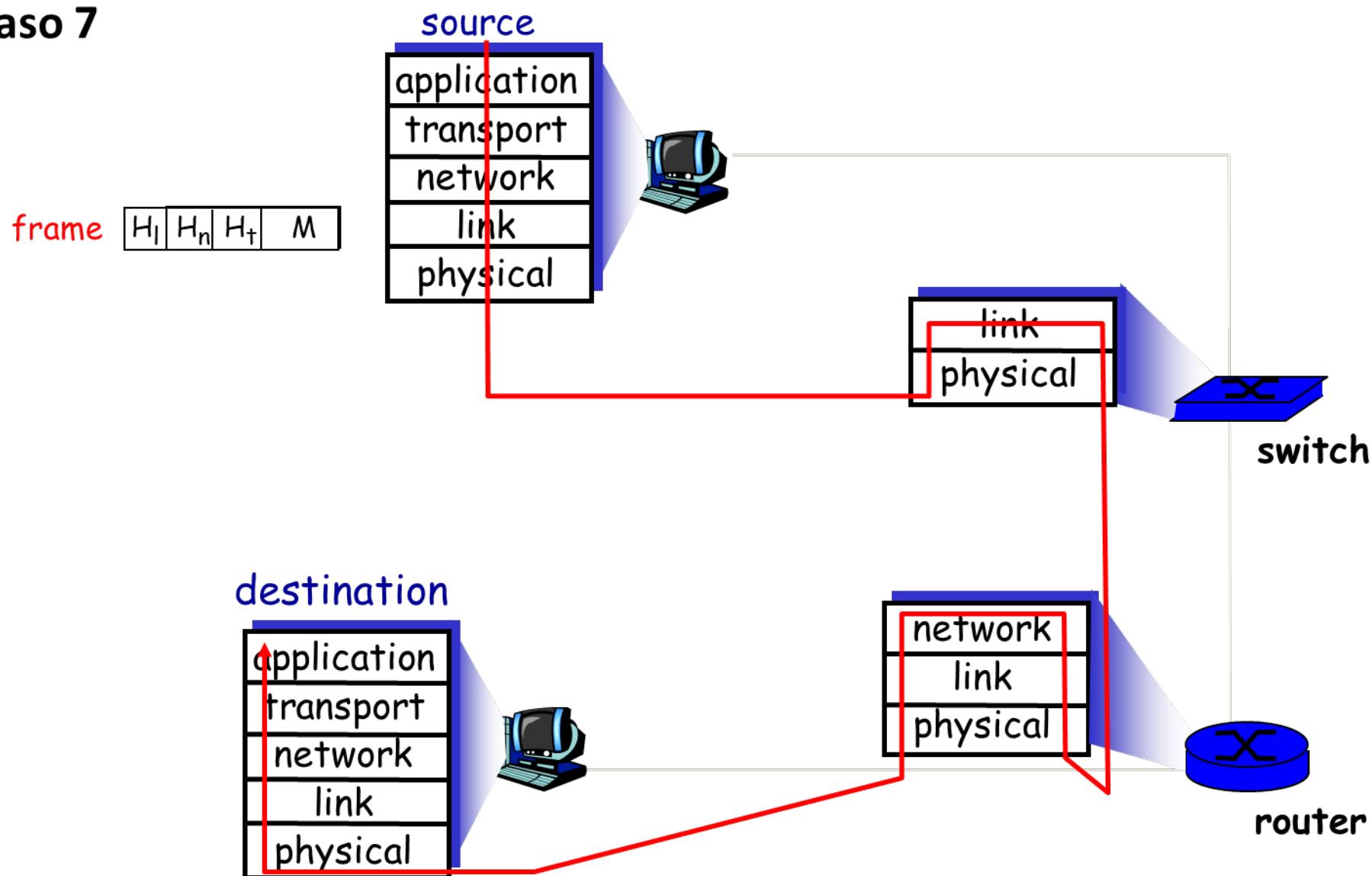
# Encapsulamiento de datos - Ejemplo

Paso 6



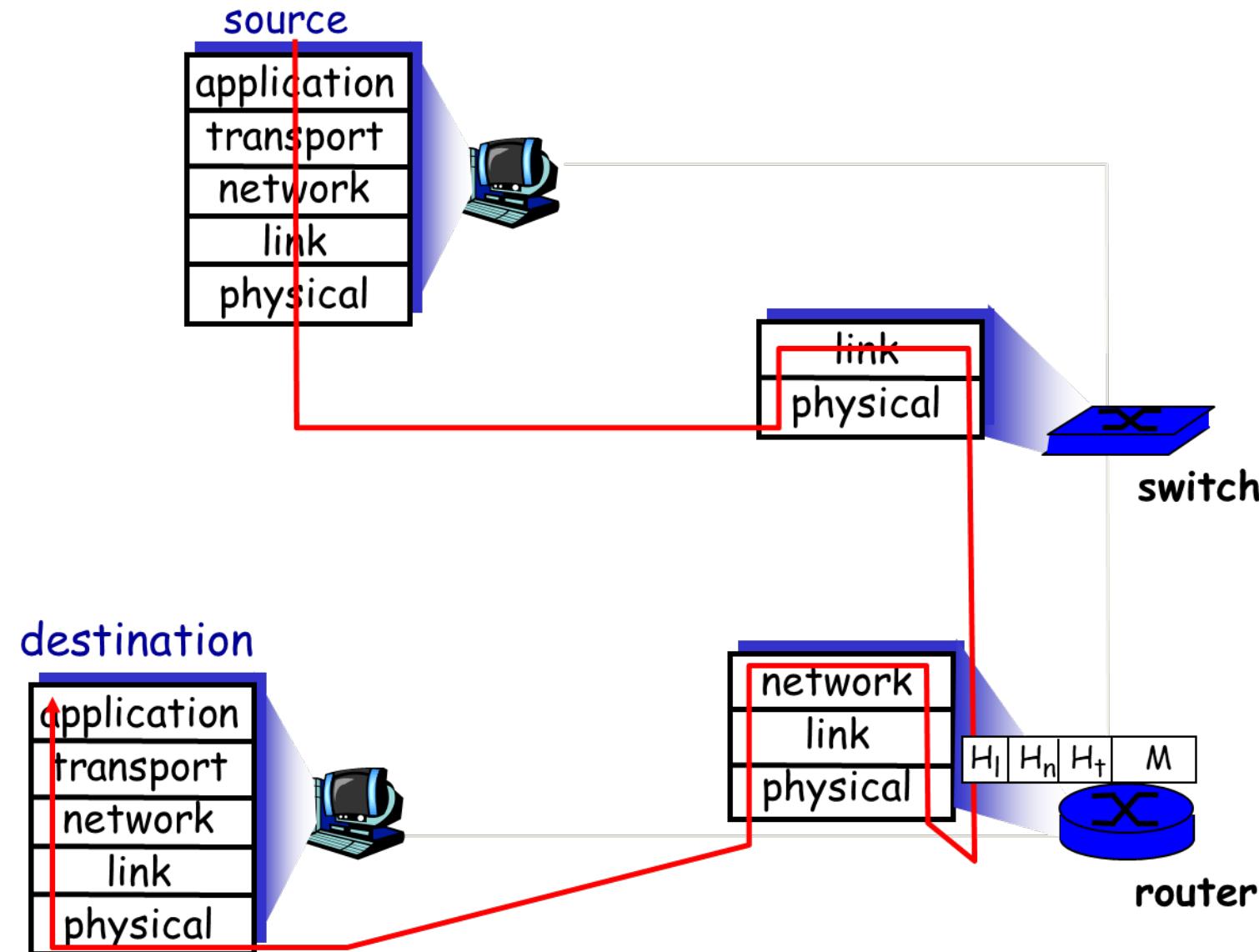
# Encapsulamiento de datos - Ejemplo

Paso 7



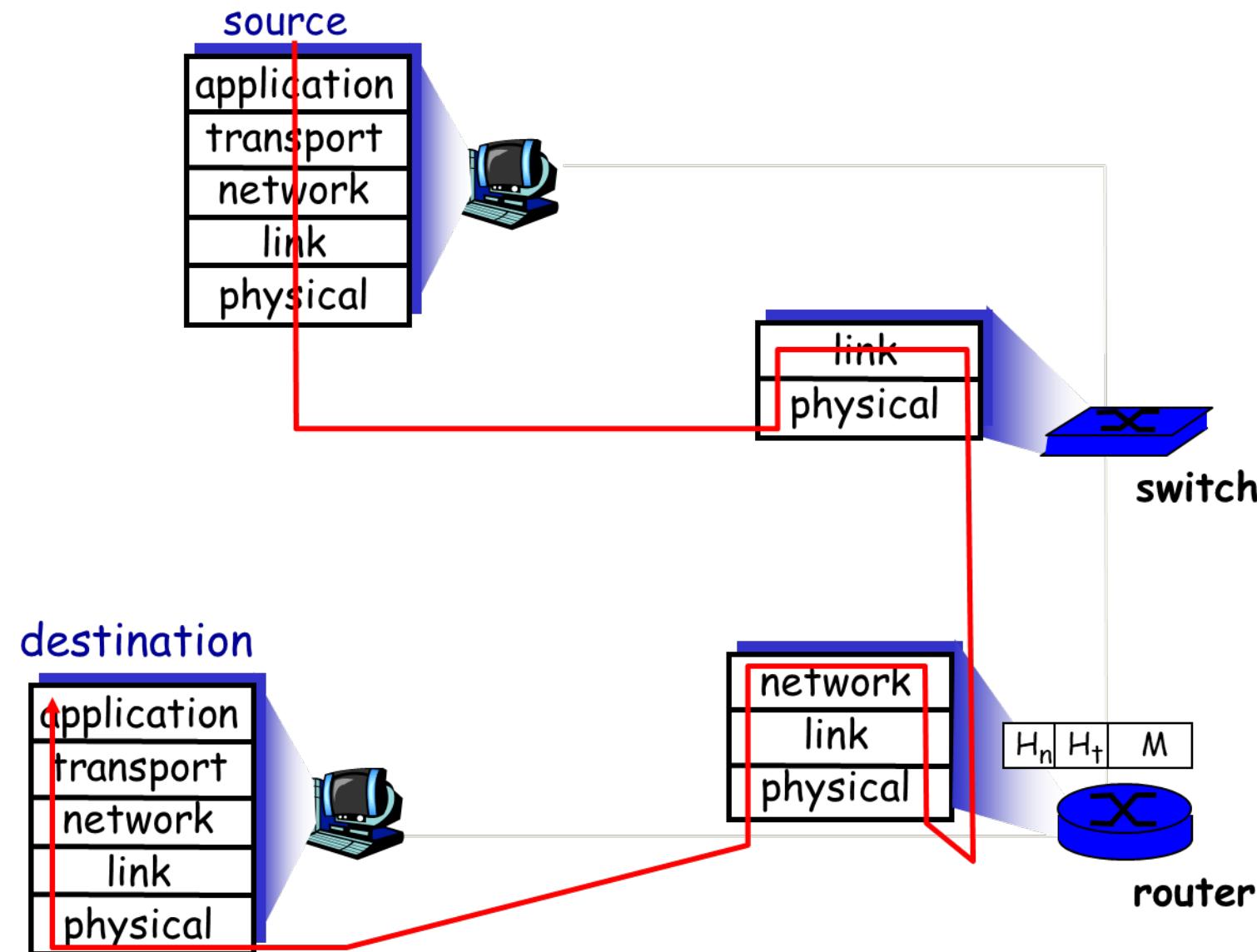
# Encapsulamiento de datos - Ejemplo

Paso 8



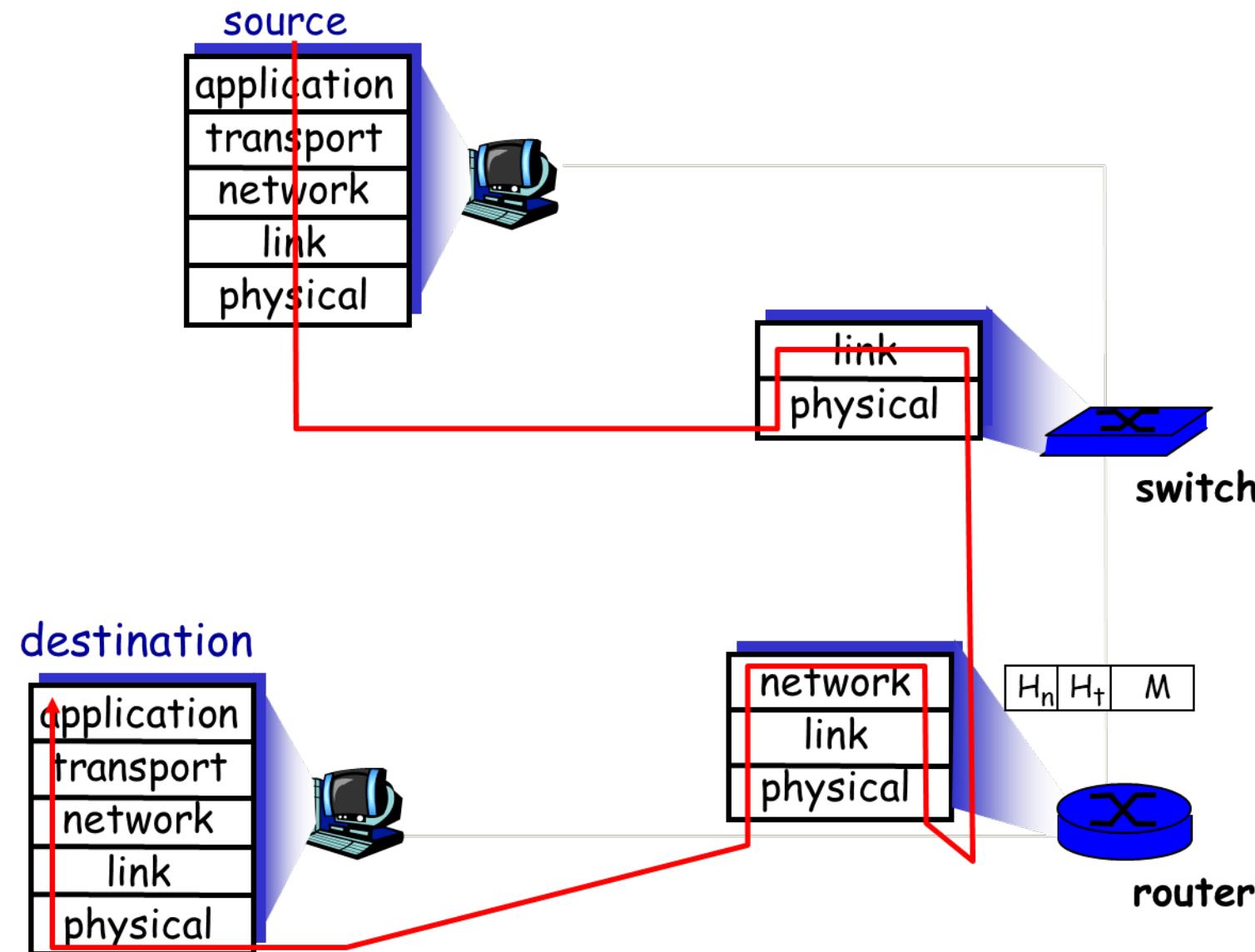
# Encapsulamiento de datos - Ejemplo

Paso 9



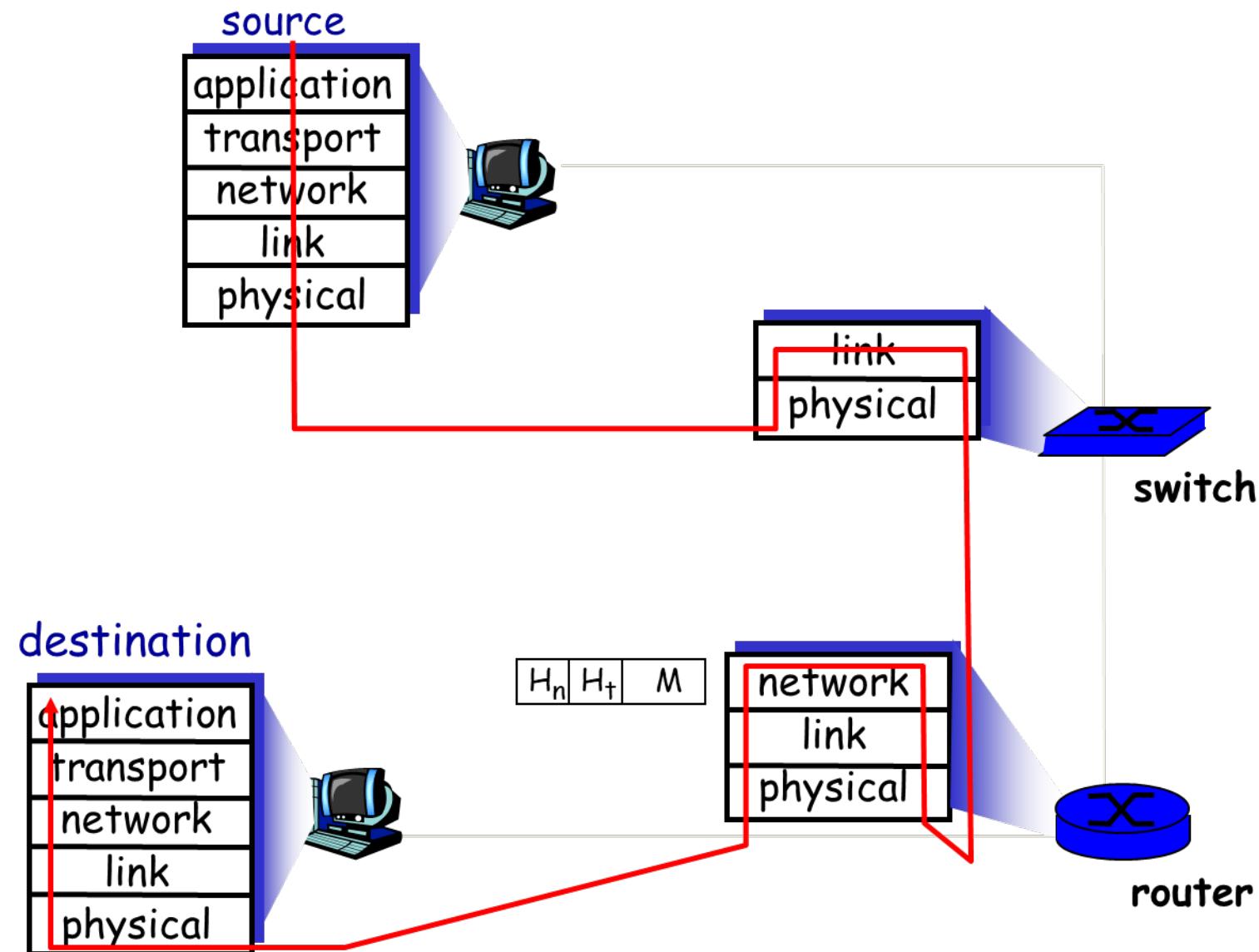
# Encapsulamiento de datos - Ejemplo

Paso 10



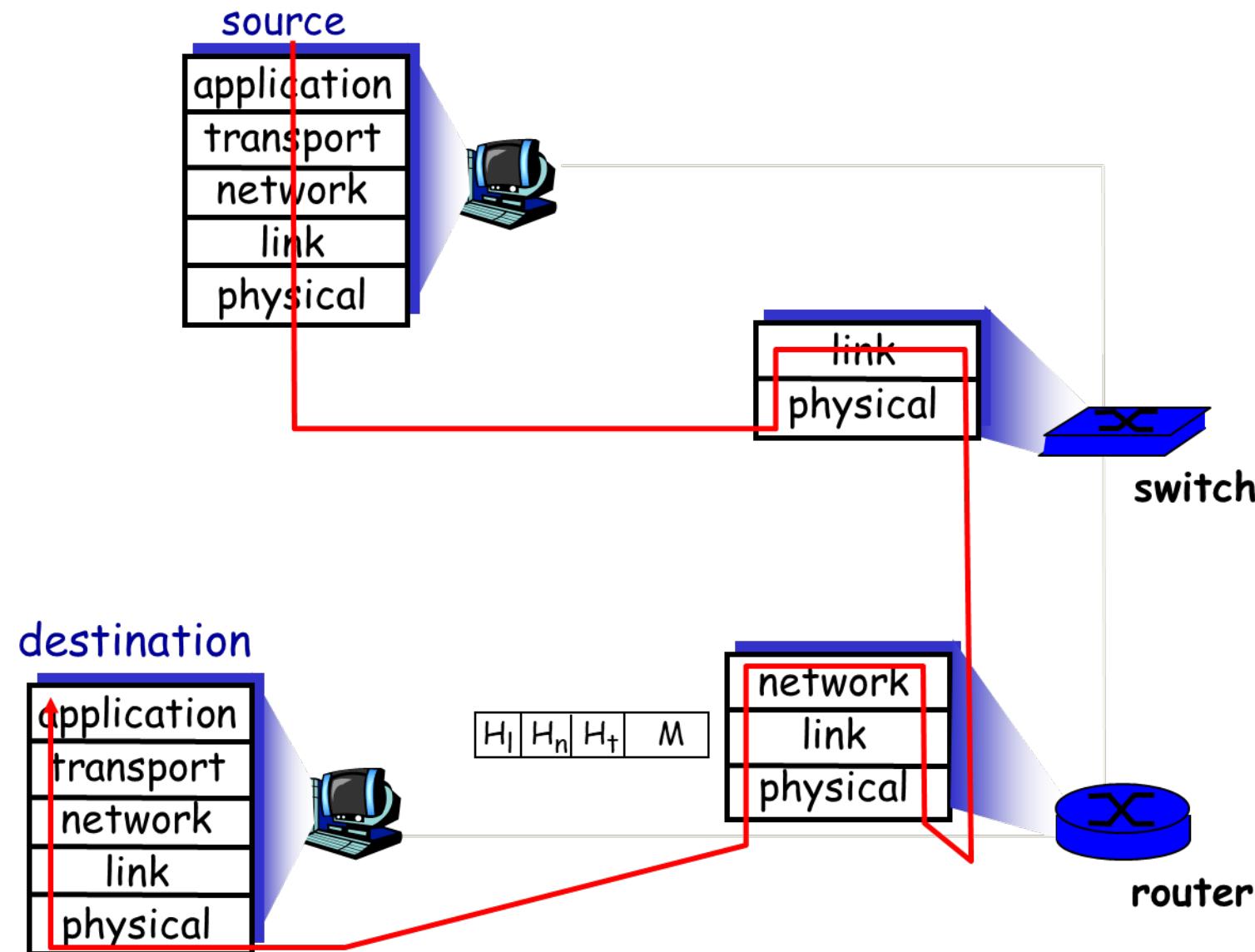
# Encapsulamiento de datos - Ejemplo

Paso 11



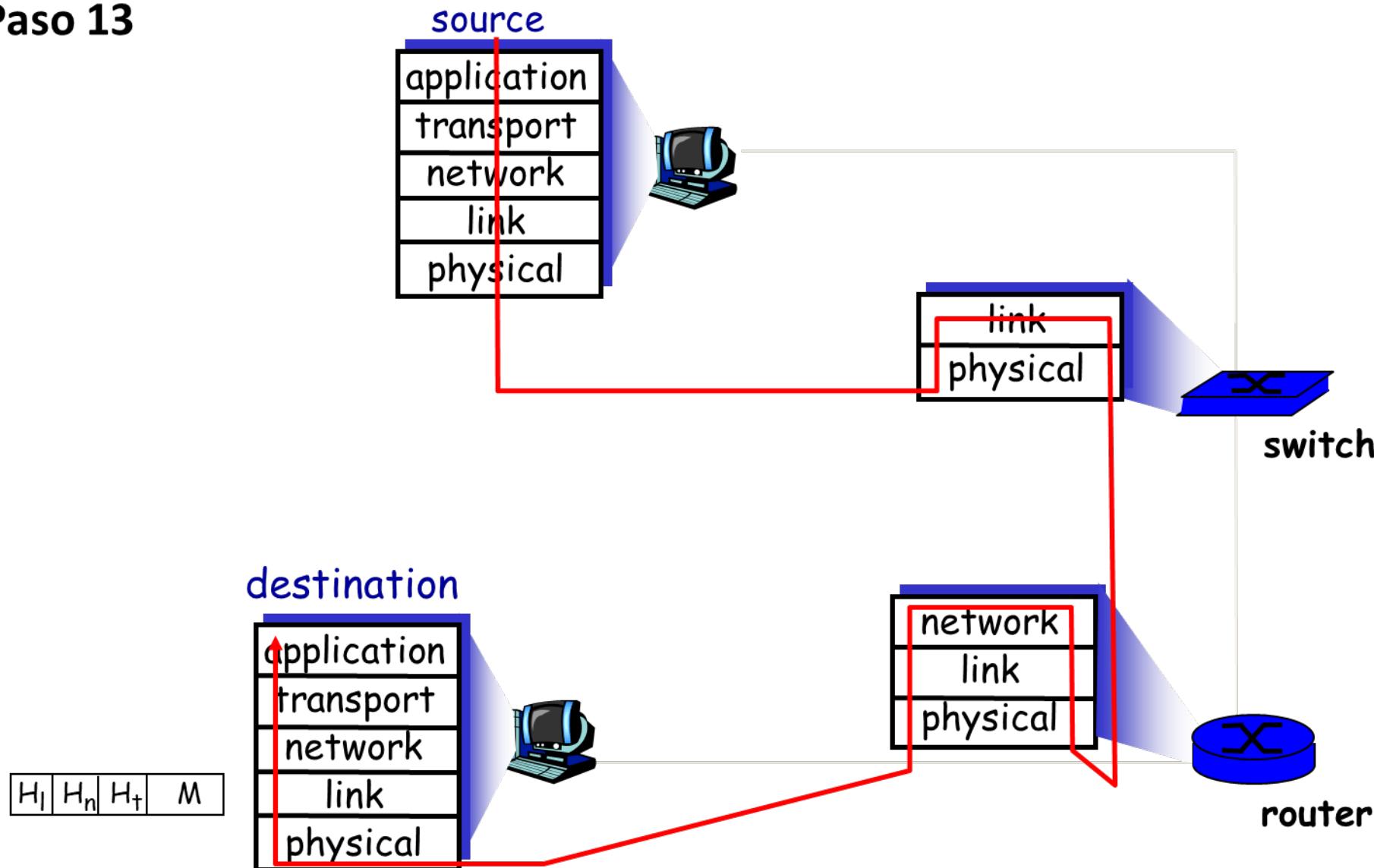
# Encapsulamiento de datos - Ejemplo

Paso 12



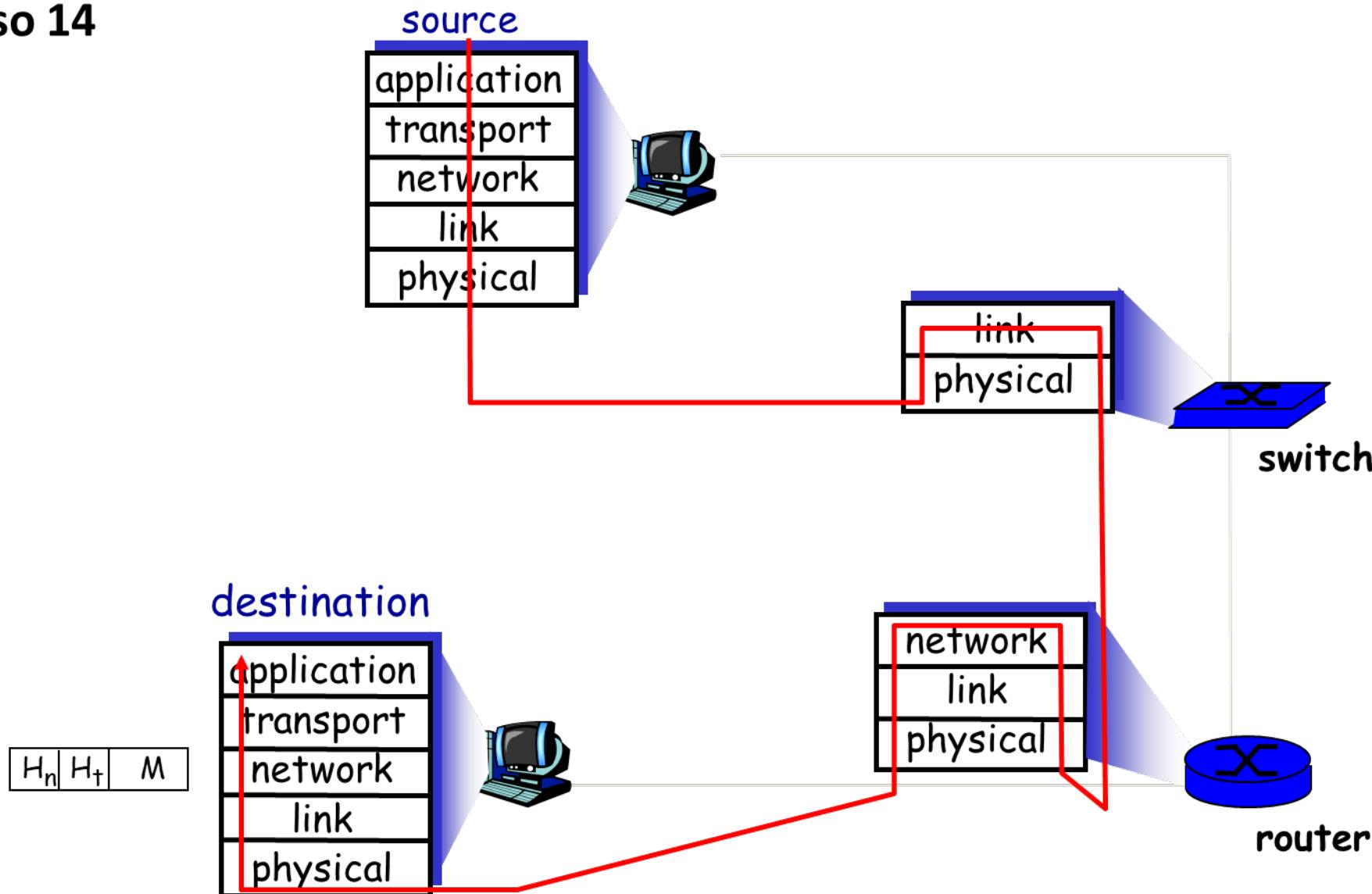
# Encapsulamiento de datos - Ejemplo

Paso 13



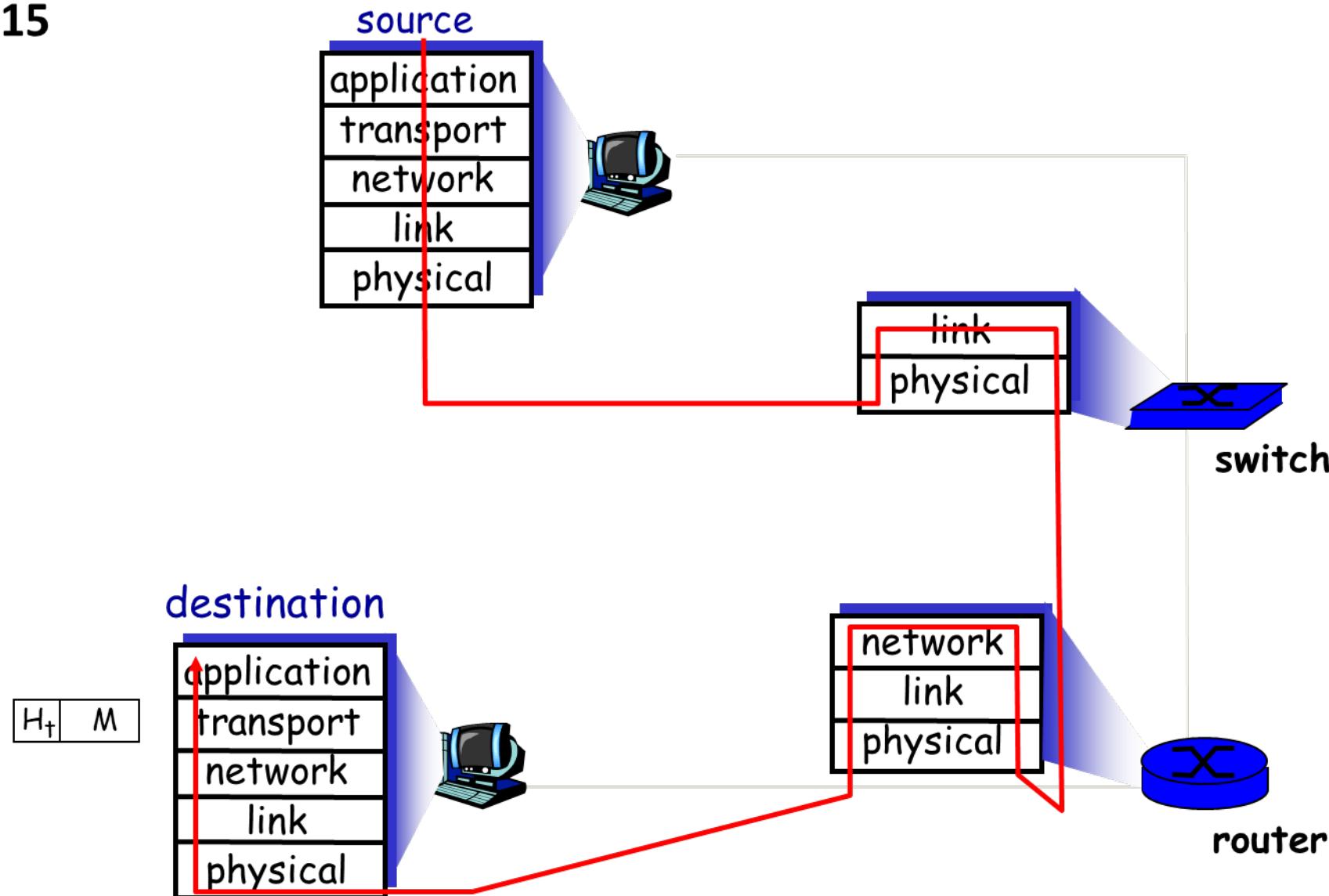
# Encapsulamiento de datos - Ejemplo

Paso 14



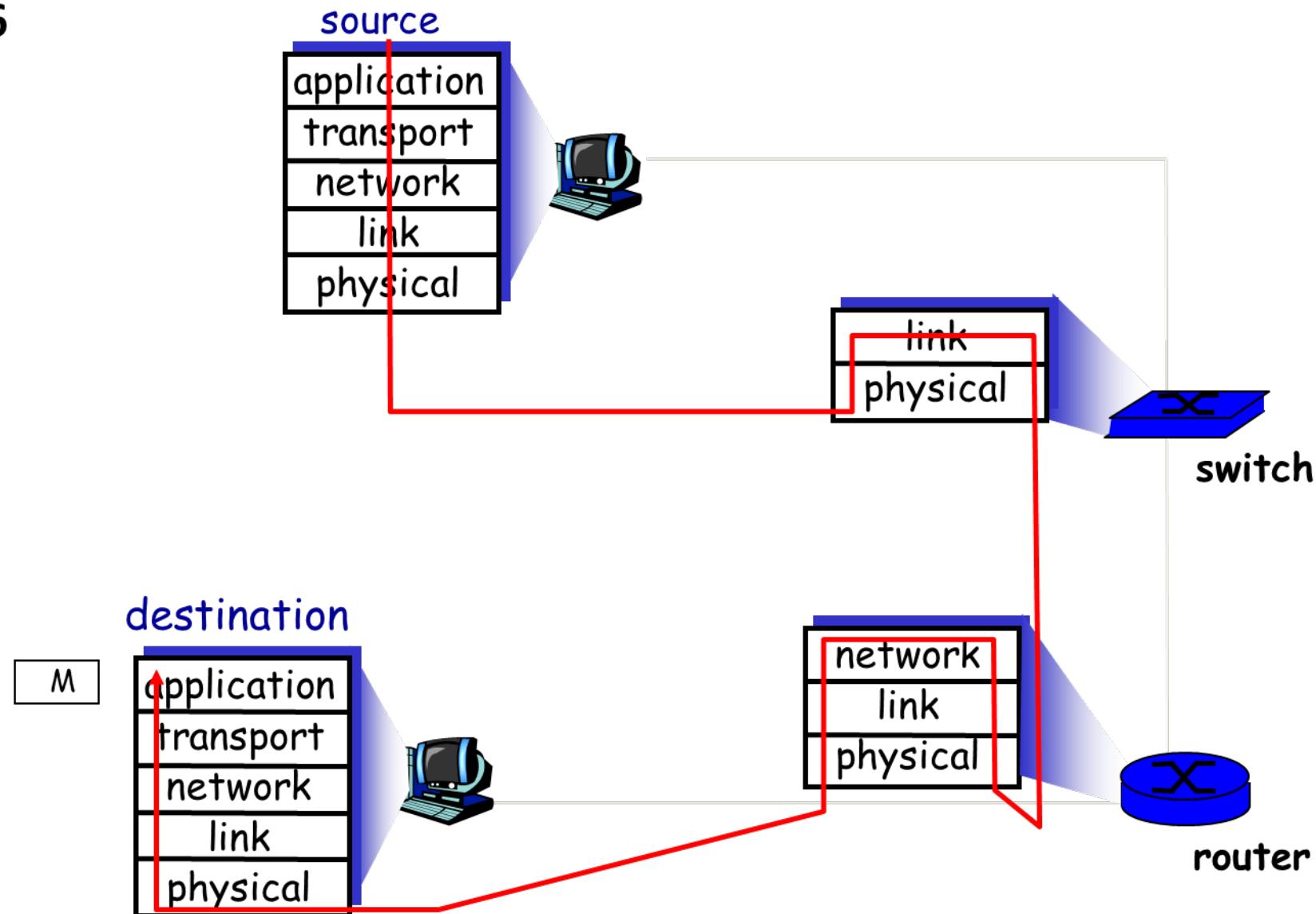
# Encapsulamiento de datos - Ejemplo

Paso 15



# Encapsulamiento de datos - Ejemplo

Paso 16



# Encapsulamiento de datos - Vídeo

[https://youtu.be/d\\_0En4l9Xkw](https://youtu.be/d_0En4l9Xkw)

Disponible en PLATEA

The screenshot shows a YouTube video player with the following details:

- Title:** Encapsulamiento de datos en TCP/IP
- Video Player Controls:** A play button, a progress bar, and a volume icon.
- Thumbnail:** A diagram of two computer monitors connected to servers via dashed lines, labeled "PID".
- Content Area:** Three diagrams illustrating network layers:
  - Red (Orange Box):** Shows a network of multiple hosts connected to a central router, with a play button icon.
  - Enlace (Yellow Box):** Shows two hosts, A and B, connected to a switch, with a play button icon.
  - Física (Grey Box):** Shows a close-up of a network cable, with a play button icon.
- Text Labels:** Descriptions for each layer:
  - Red: Conexión entre múltiples equipos en Direcciones IP para facilitar el enrute. Datagramas IP con datos y dirección.
  - Enlace: Conexión entre múltiples equipos en Direcciones MAC – Concepto de LAI. Tramas con secuencias de datos delimitados.
  - Física: Conexión directa entre dos equipos. Pulso eléctricos/ópticos/electromagnéticos.
- Page Footer:** Programación y administración de redes - Universidad de Jaén

# Comparación entre arquitecturas OSI y TCP/IP



# OSI vs TCP/IP

# Recursos

- **The TCP/IP Guide** (hipervínculo a la dcha.) ofrece un resumen detallado de las diferencias entre estos dos modelos de arquitectura de red
  - **Redes de computadoras e Internet** (ver bibliografía básica de la asignatura, accesible a través de la biblioteca digital de la UJA), en el apartado 1.5 del primer capítulo facilita una descripción de las capas y diferencias entre OSI y TCP/IP

<http://tcpipguide.com/free/tcpiparchitecturedandthetcpipmodel-2.htm>

