

Chương 2: Giao thức và kiến trúc phân tầng

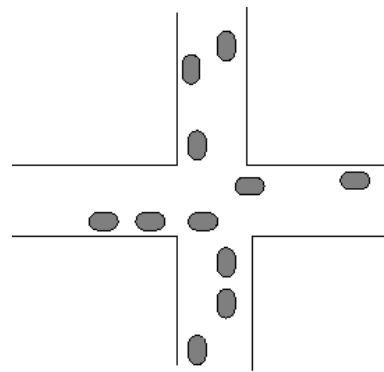
Protocols and Layered Architecture

Giao thức và kiến trúc phân tầng:

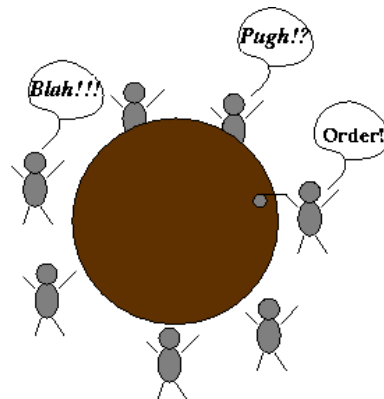
- ❑ Thế nào là giao thức (protocols)?
- ❑ Kiến trúc phân tầng và các hệ thống cấp bậc giao thức (protocol hierarchies)
- ❑ Các vấn đề cần lưu ý khi thiết kế các tầng
- ❑ Phương thức hoạt động: hướng kết nối và phi kết nối
- ❑ Mô hình tham chiếu OSI (The Open Systems Interconnection Reference Model)
- ❑ Mô hình TCP/IP (The Transmission Control Protocol/Internet Protocol Model)
- ❑ So sánh giữa mô hình tham chiếu OSI và TCP/IP

Giao thức (Protocols)

- ❑ Giao thức: những luật giúp những thành phần mạng (network elements) hội thoại với nhau
- ❑ Giao thức định nghĩa sự thỏa thuận (agreement) giữa những thực thể ngang hàng (peering entities)
 - Khuôn dạng và ngữ nghĩa của thông điệp được trao đổi
- ❑ Giao thức trong cuộc sống hàng ngày:
 - Luật giao thông, thảo luận bàn tròn...



Traffic Control



Floor Control

Kiến trúc phân tầng hay hệ thống cấp bậc giao thức!?

MMT: phức tạp!

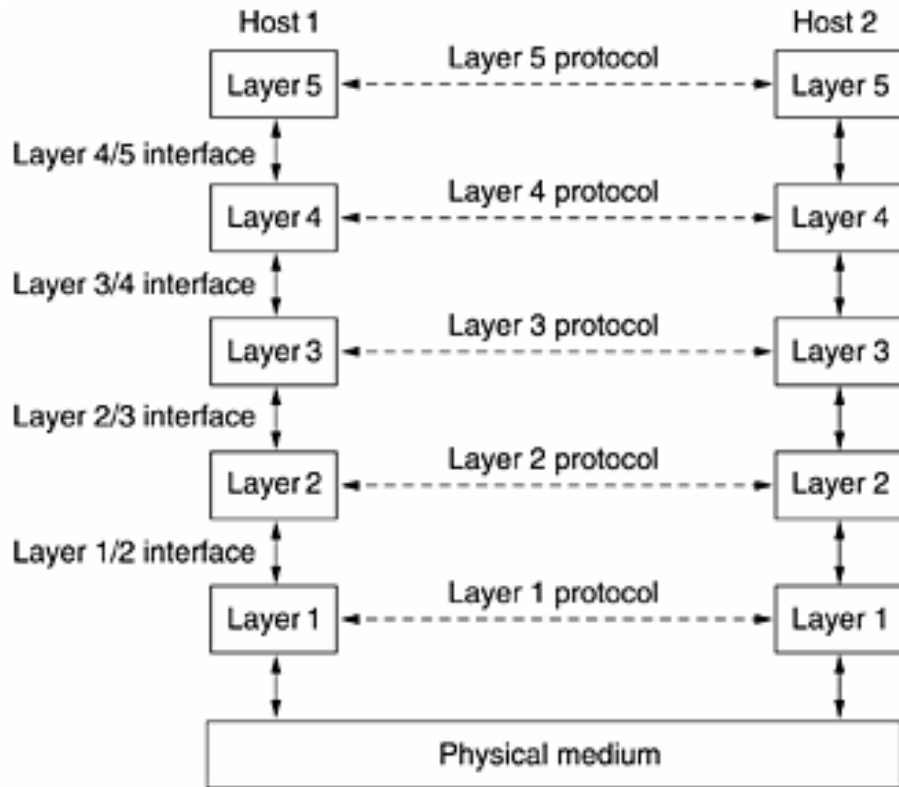
□ nhiều "mảnh":

- trạm (hosts)
- bộ định tuyến (router)
- phương tiện truyền thông
- ứng dụng (applications)
- giao thức (protocols)
- phần cứng, phần mềm (hardware, software)

Câu hỏi:

Cách nào để thiết lập cấu trúc mạng?

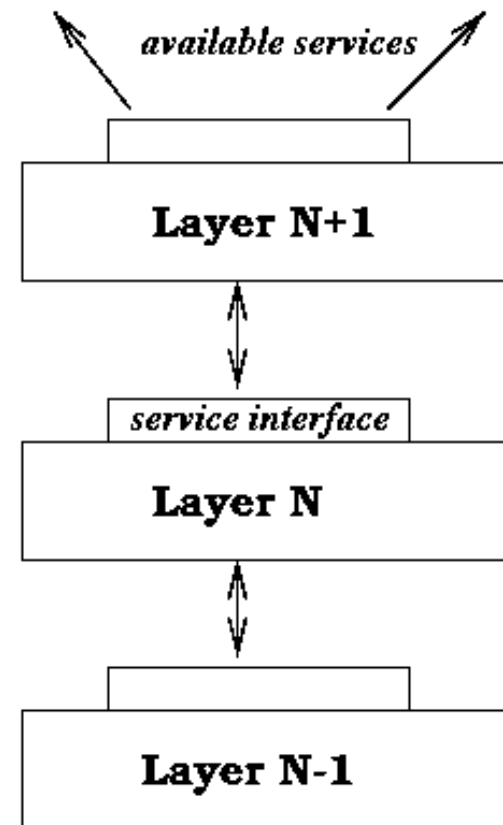
Kiến trúc phân tầng



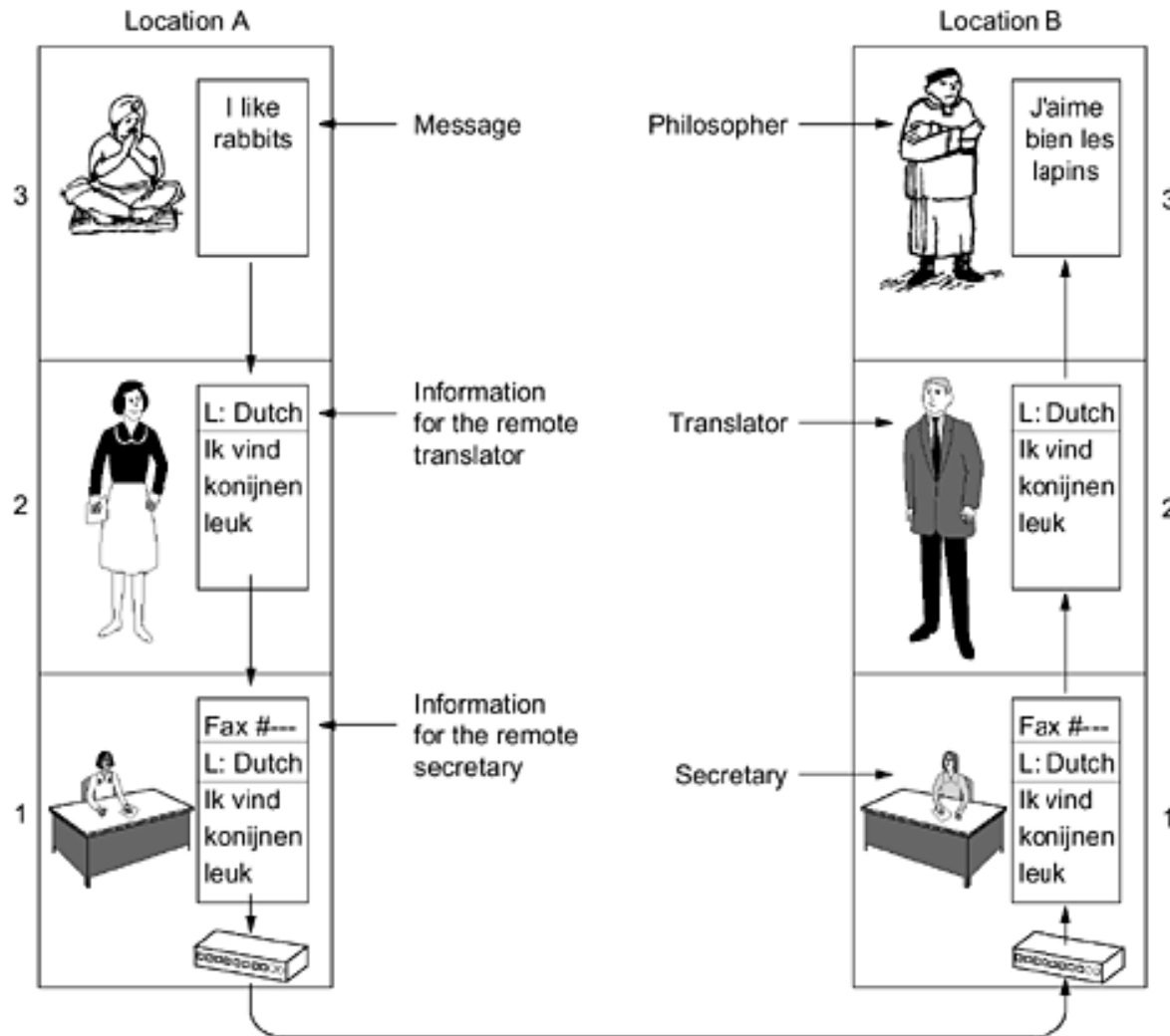
- Các tầng (Layers)
- Giao diện (Interfaces): định nghĩa các thao tác nguyên thủy và các dịch vụ mà tầng dưới cung cấp cho tầng trên
- Giao thức (Protocols): được sử dụng để thi hành (implement) các dịch vụ
- Một tập của các tầng và giao thức được gọi là một kiến trúc mạng (Network Architecture)

Kiến trúc phân tầng (2)

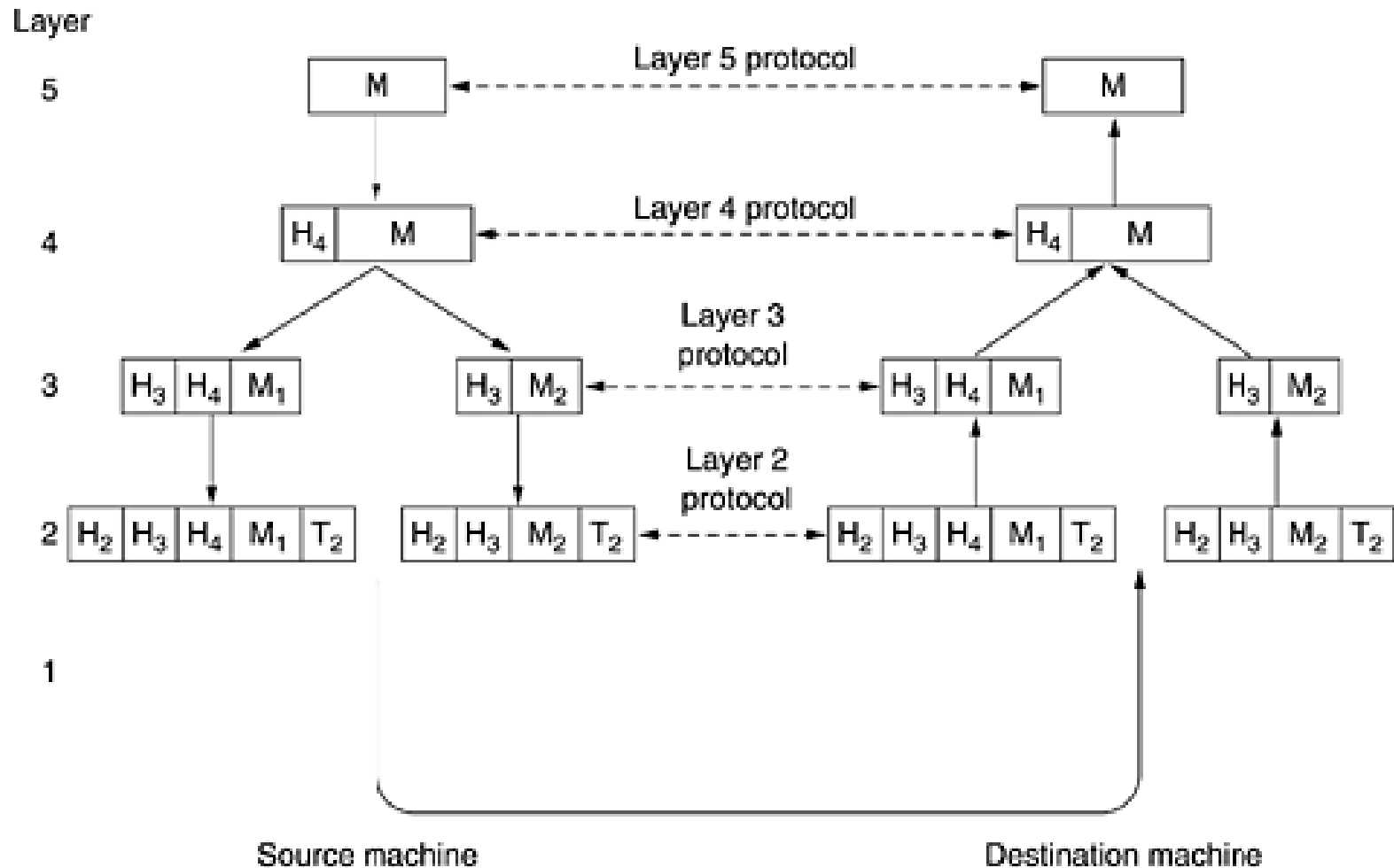
- ❑ *Phân tầng* giúp đơn giản hóa kiến trúc của hệ thống phức tạp
- ❑ Tầng N dựa vào dịch vụ (*services*) từ tầng N-1 để cung cấp dịch vụ cho tầng N+1
- ❑ Dịch vụ cần đến từ tầng dưới độc lập với sự thực thi (*implementation*) của nó
 - Tầng N thay đổi sẽ không ảnh hưởng đến các tầng khác
 - Che dấu thông tin và sự phức tạp
 - Tương tự như lập trình hướng đối tượng



Ví dụ về kiến trúc phân tầng



Truyền thông Vật lý, logic



Các vấn đề thiết yếu khi thiết kế các tầng

- ❑ Cơ chế định danh người gửi, nhận?
- ❑ Truyền dữ liệu theo chế độ nào: đơn công (simplex), bán song công (half-duplex), song công (full-duplex)?
- ❑ Kiểm soát lỗi? (Error control)
- ❑ Kiểm soát luồng? (Flow control)
- ❑ Tháo rời (disassembling) và ráp lại (reassembling) các thông điệp dài
- ❑ Dồn và tách kênh (Multiplexing & demultiplexing)
- ❑ Chọn đường

Dịch vụ hướng kết nối (Connection-oriented Service)

□ Người gửi - Sender

- Yêu cầu "kết nối" đến người nhận
- Chờ đợi Mạng thiết lập kết nối
- Duy trì kết nối trong khi gửi dữ liệu
- Ngắt kết nối khi hết nhu cầu

□ Mạng - Network

- Nhận yêu cầu kết nối
- Thiết lập kết nối và thông báo cho người gửi
- Truyền dữ liệu qua mỗi kết nối
- Giải phóng kết nối khi người gửi yêu cầu

Dịch vụ phi kết nối (Connectionless Service)

□ Người gửi - Sender

- Tạo các packet để gửi
- Đánh địa chỉ người nhận trong mỗi gói
- Truyền gói tin cho mạng để chuyển đi

□ Mạng - Network

- Sử dụng địa chỉ đích để chuyển tiếp gói tin
- Giao gói tin đến nơi nhận

So sánh giữa hướng kết nối và phi kết nối (Connection-Oriented vs. Connectionless)

□ Connection-Oriented

- Telephone System, Virtual Circuit Model
- Đường dẫn được thiết lập trước khi dữ liệu được gửi
- Chỉ cần định danh mỗi kết nối
- Tất cả dữ liệu đi cùng một đường

□ Connectionless

- Postal System, Datagram Model
- Không cần thiết lập đường dẫn trước khi truyền dữ liệu
- Gói tin chứa địa chỉ nơi nhận
- Mỗi gói tin được xử lý độc lập

So sánh giữa hướng kết nối và phi kết nối (Connection-Oriented vs. Connectionless)

❑ Connection-Oriented

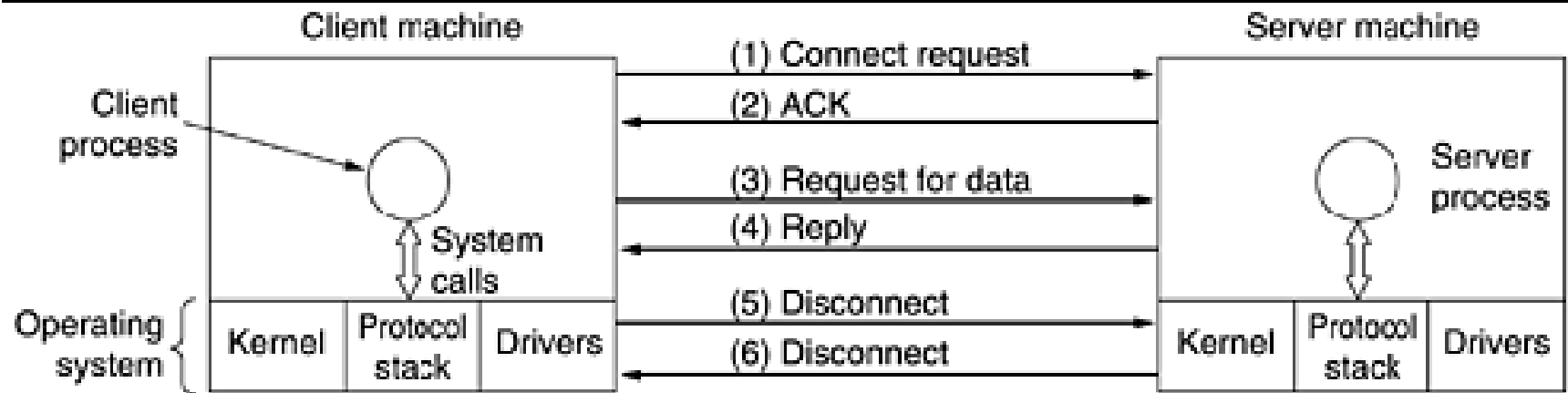
- "Gánh nặng" thiết lập kết nối
- Thiết bị chuyển tiếp phải lưu giữ trạng thái của các kết nối đang hoạt động
- Có thể đặt trước dải thông

❑ Connectionless

- Không trạng thái và ít gánh nặng
- Không thể đặt trước tài nguyên
- Cho phép broadcast/multicast

Ví dụ về các hàm dịch vụ nguyên thủy (service primitive)

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection



Một số tổ chức định chuẩn

International Standards Organization (ISO)

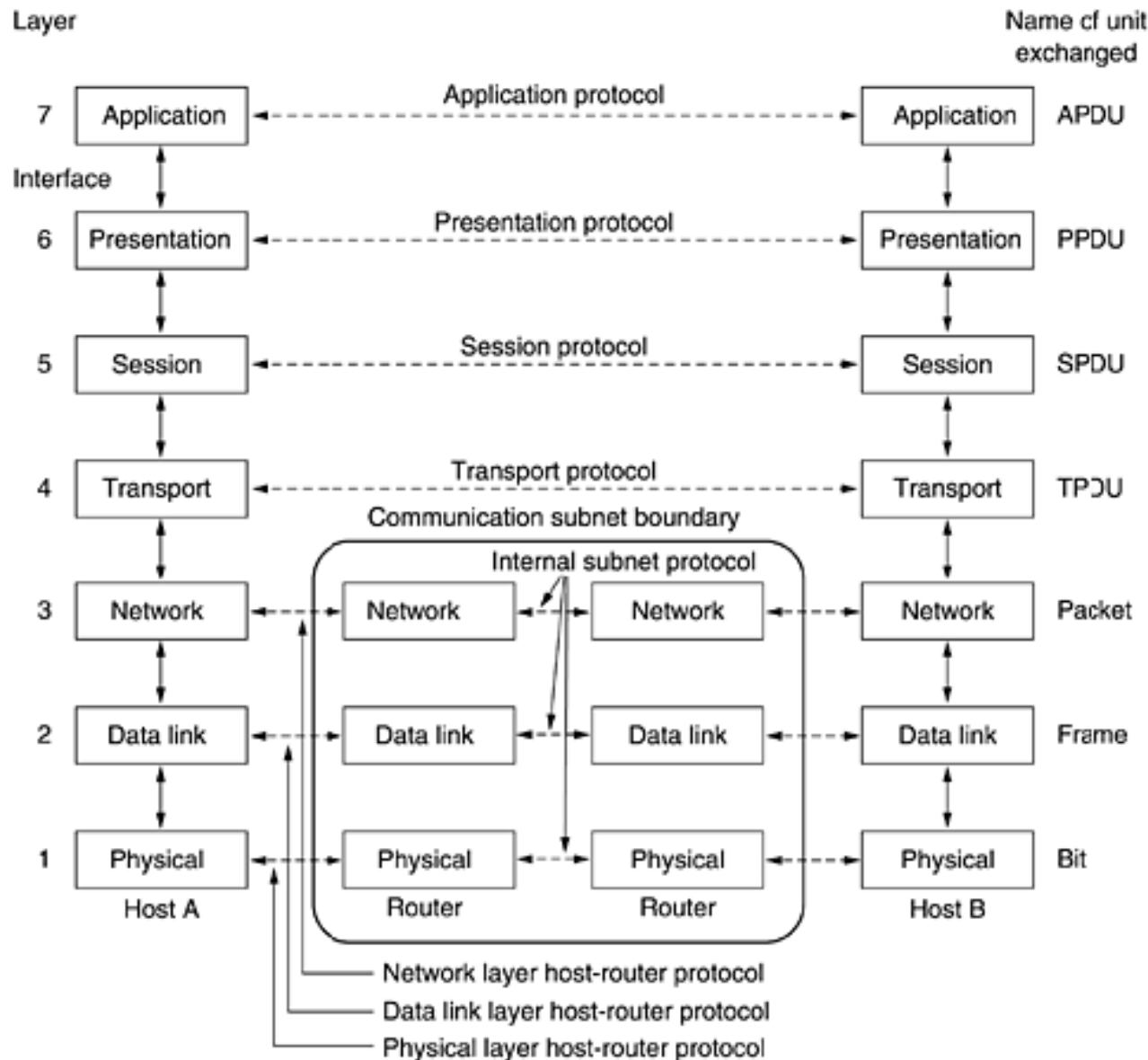
**International Telecommunications Union–
Telecommunication Standards Sector (ITU-T)**

Institute of Electrical and Electronics Engineers (IEEE)

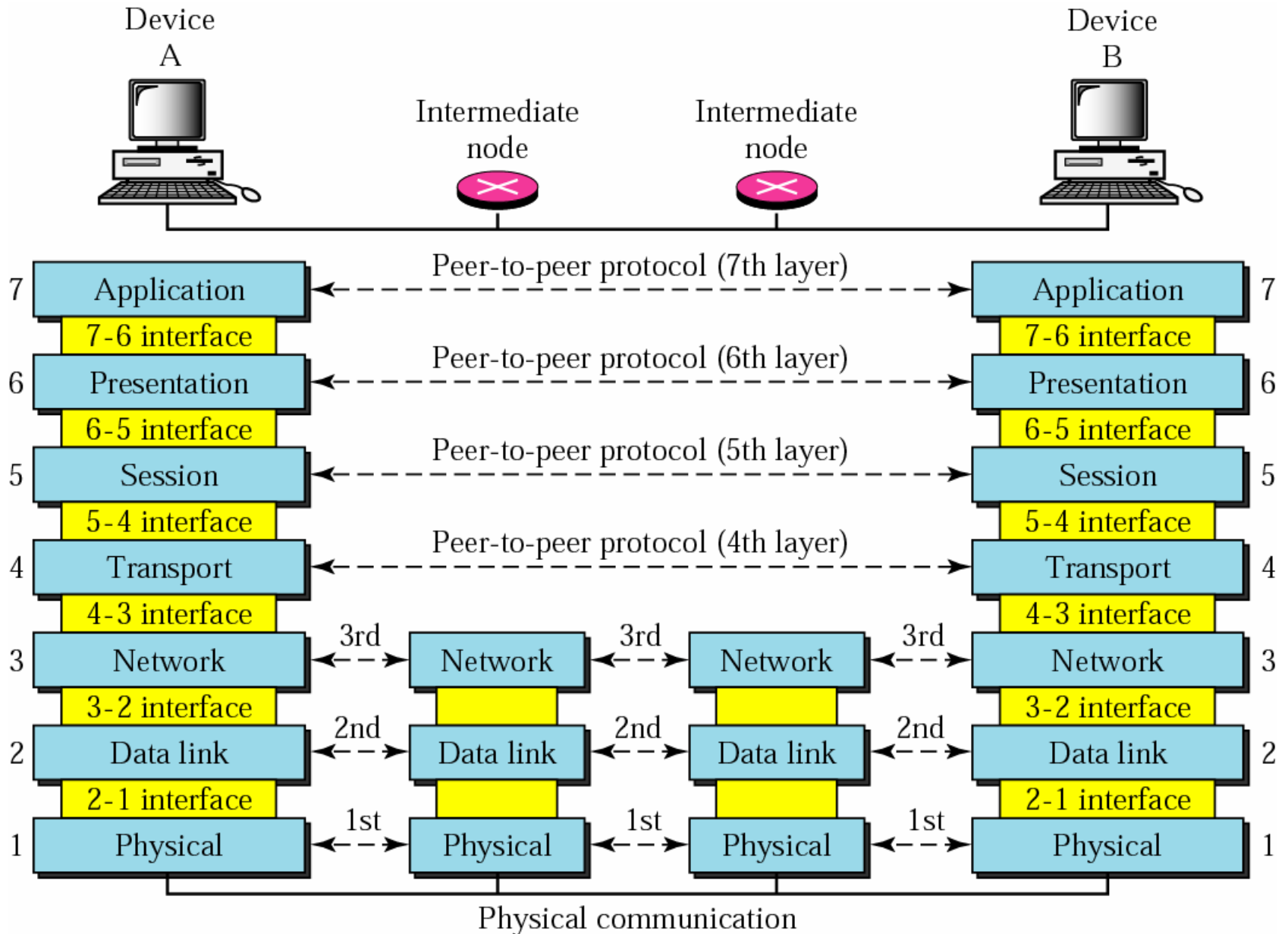
Electronic Industries Alliance (EIA)

Telecommunications Industry Association (TIA)

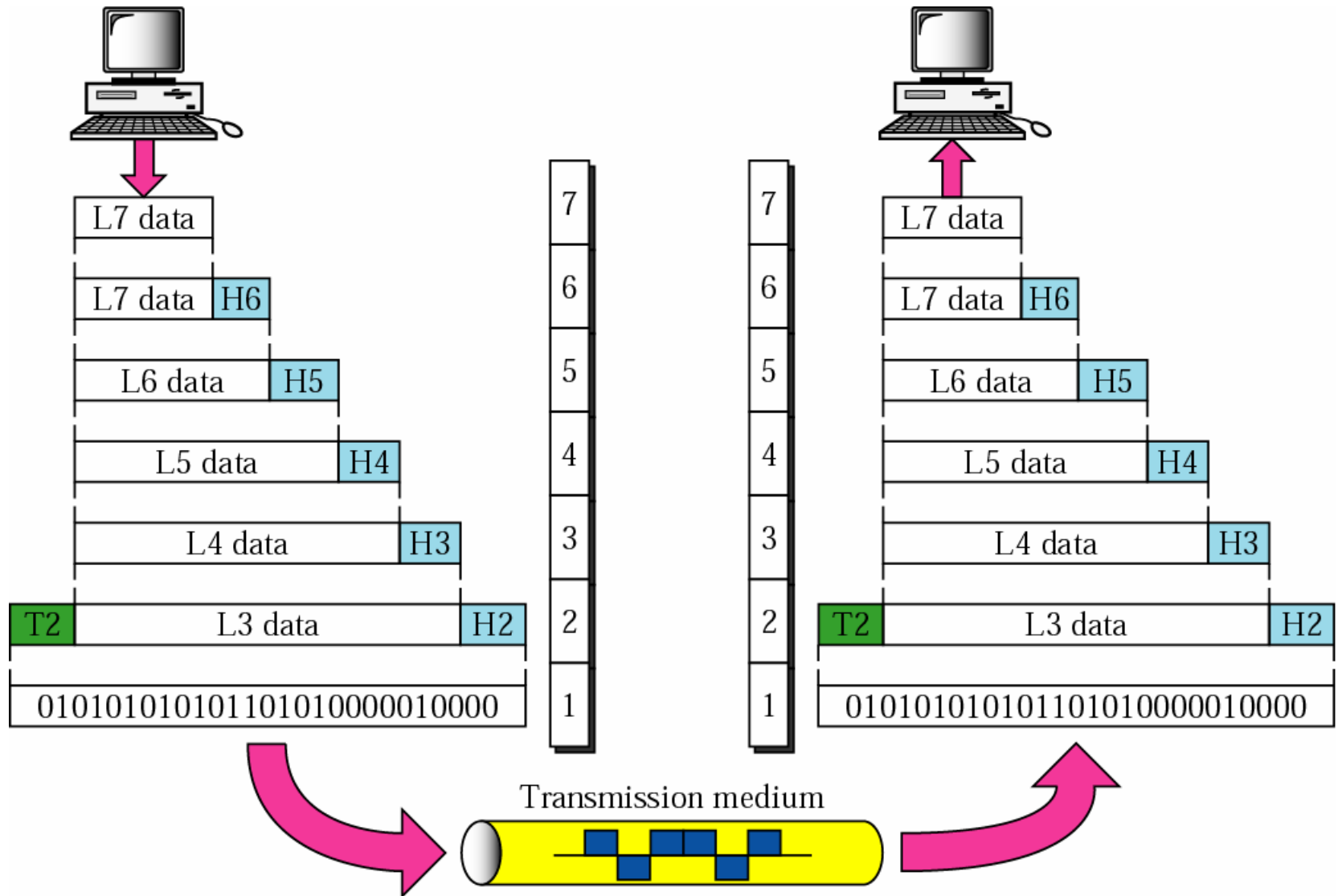
Mô hình tham chiếu OSI



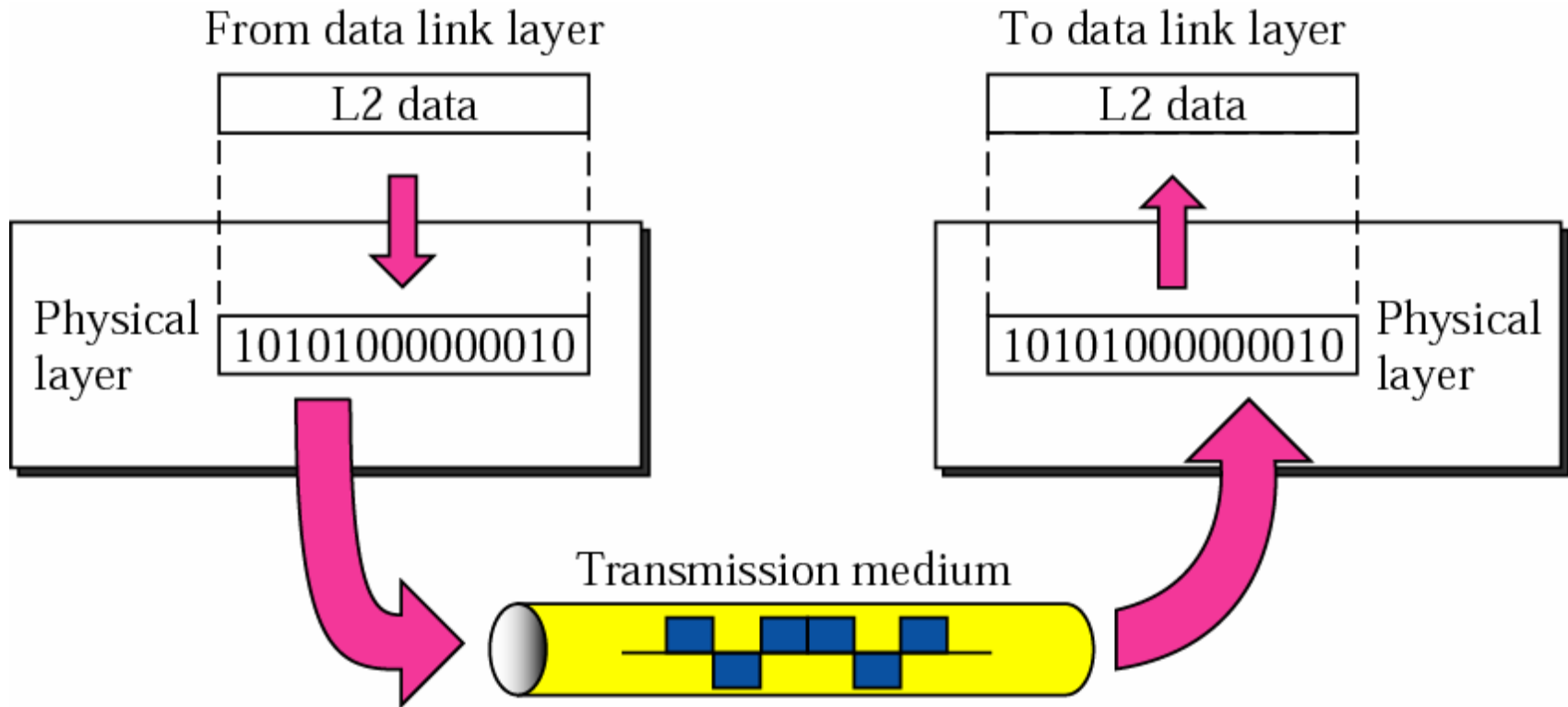
Các tầng trong mô hình OSI



Minh họa về trao đổi thông tin trong mô hình OSI



Tầng Vật lý (Physical Layer)

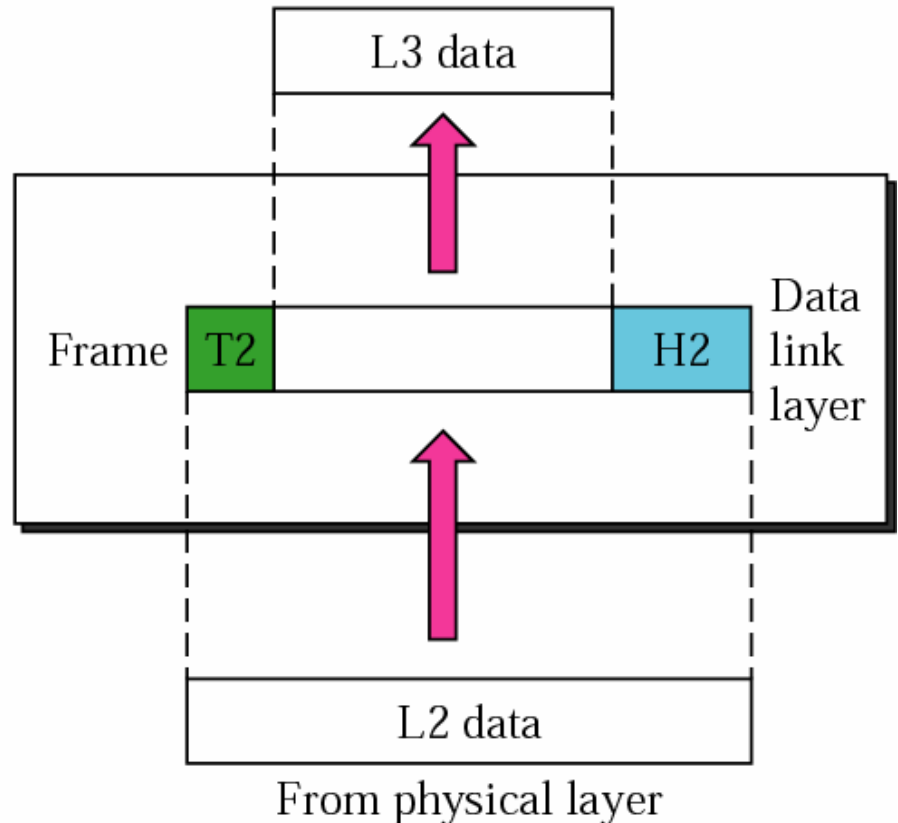
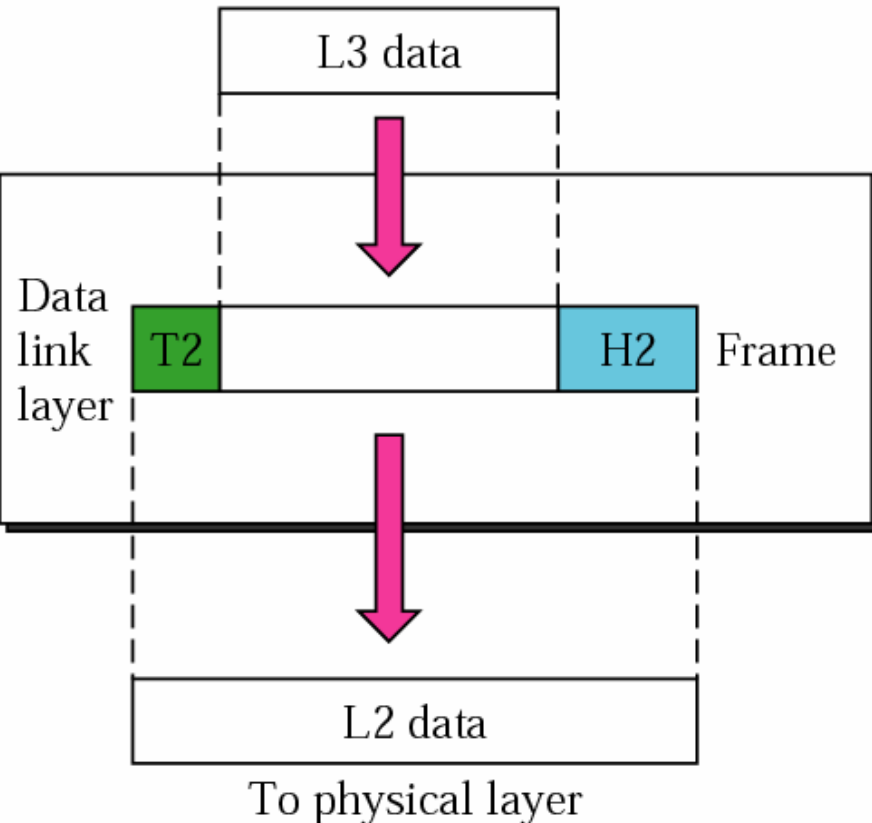


- ❑ Truyền dòng bit “tươi” (raw bits) qua đường truyền vật lý
- ❑ Giải quyết các giao diện (interfaces) thời gian, điện, cơ; phương tiện truyền vật lý

Tầng Liên kết dữ liệu (Data Link Layer)

From network layer

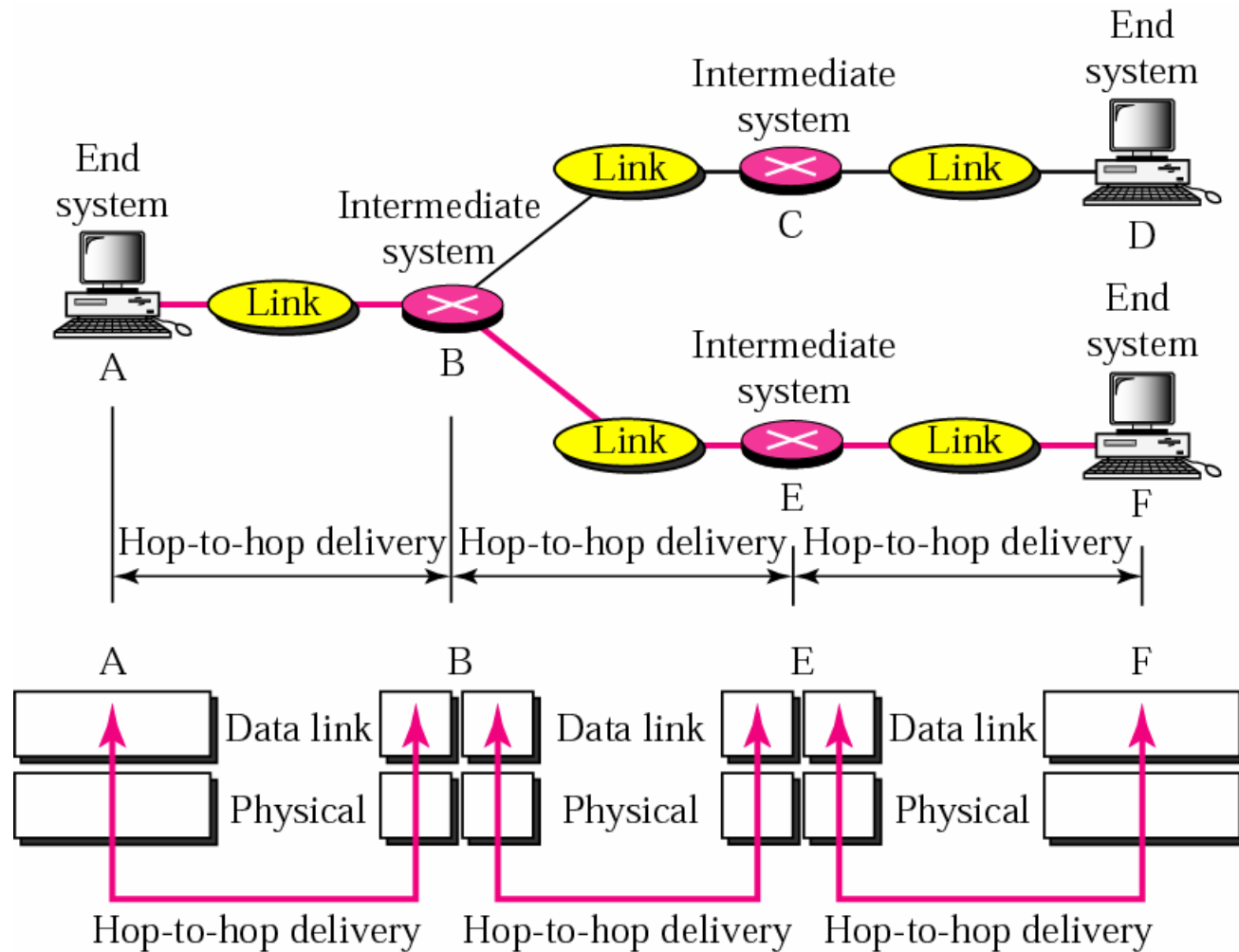
To network layer



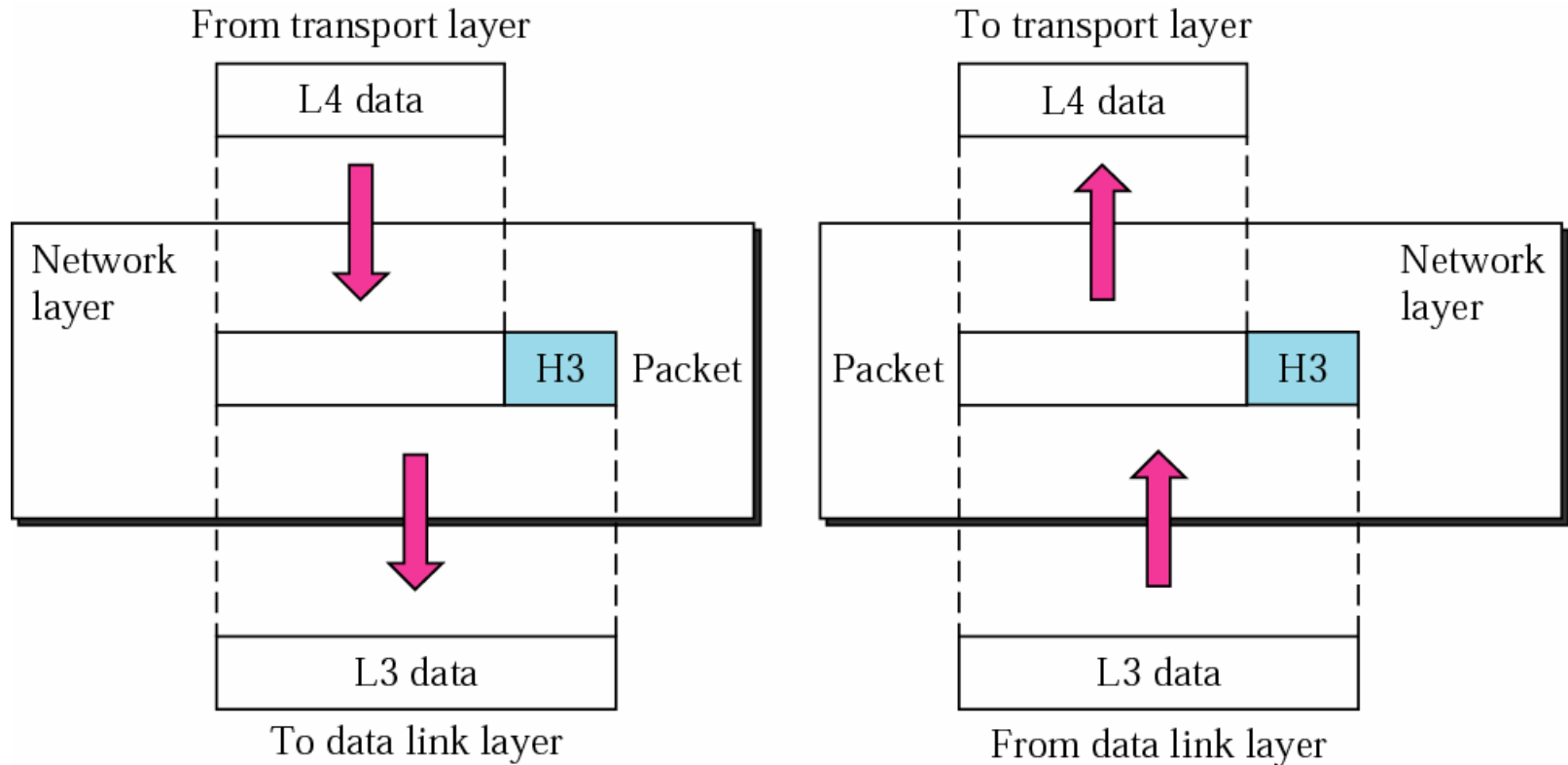
- Truyền dữ liệu giữa các nút (nodes) láng giềng
 - Định khung, kiểm soát lỗi, kiểm soát luồng
 - Điều khiển truy cập phương tiện truyền

Sự phân phát nút-nút

Node-to-node delivery



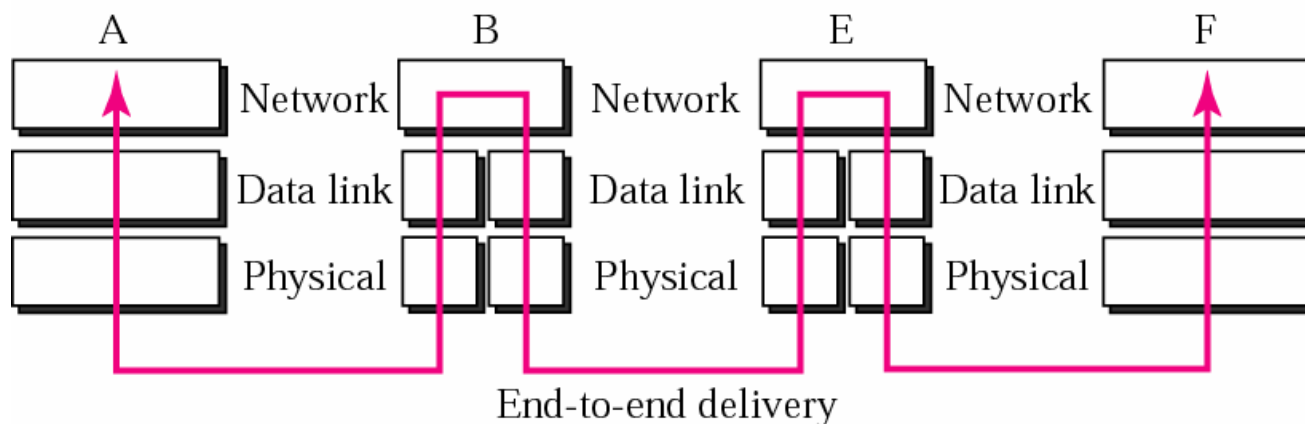
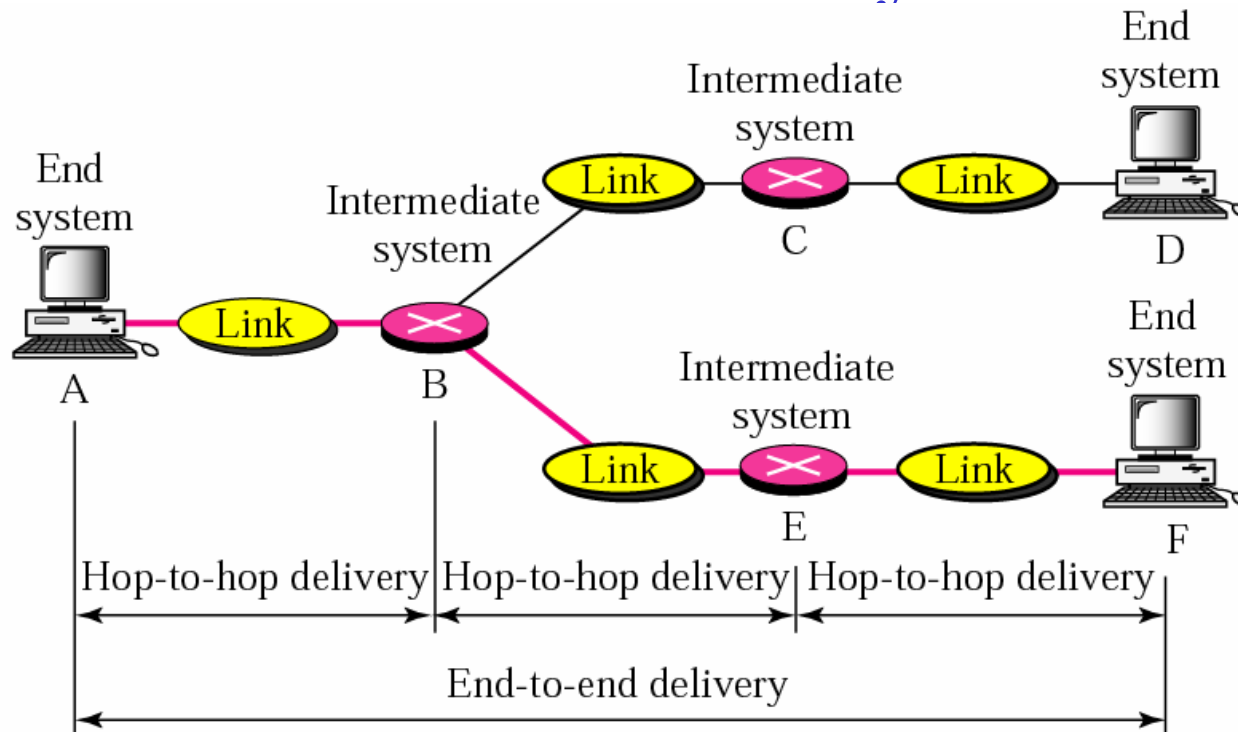
Tầng Mạng (network layer)



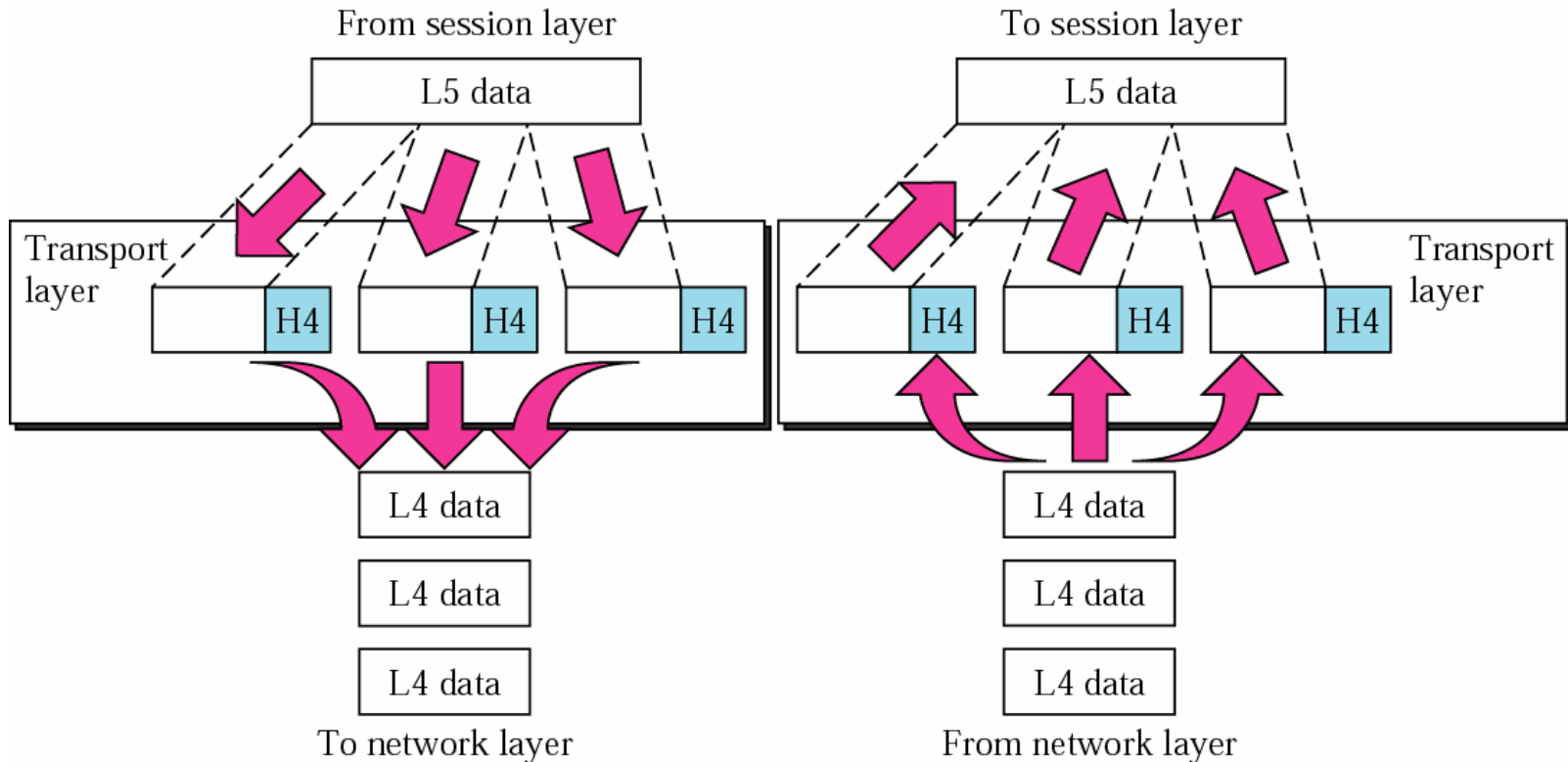
- Đánh địa chỉ
- Định tuyến cho các gói tin trên mạng
- Tránh các liên kết bị tắc nghẽn/hỏng

Sự phân phát cuối-cuối

End-to-end delivery



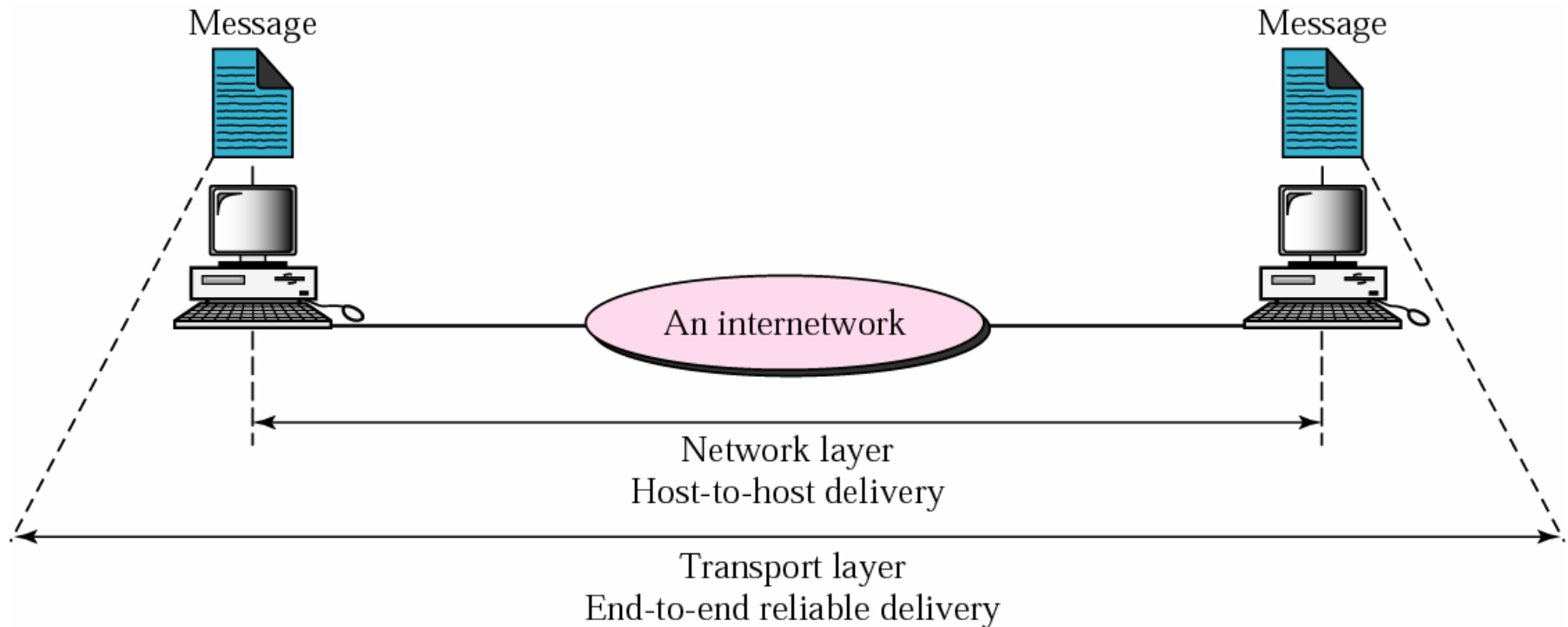
Tầng Vận chuyển (transport layer)



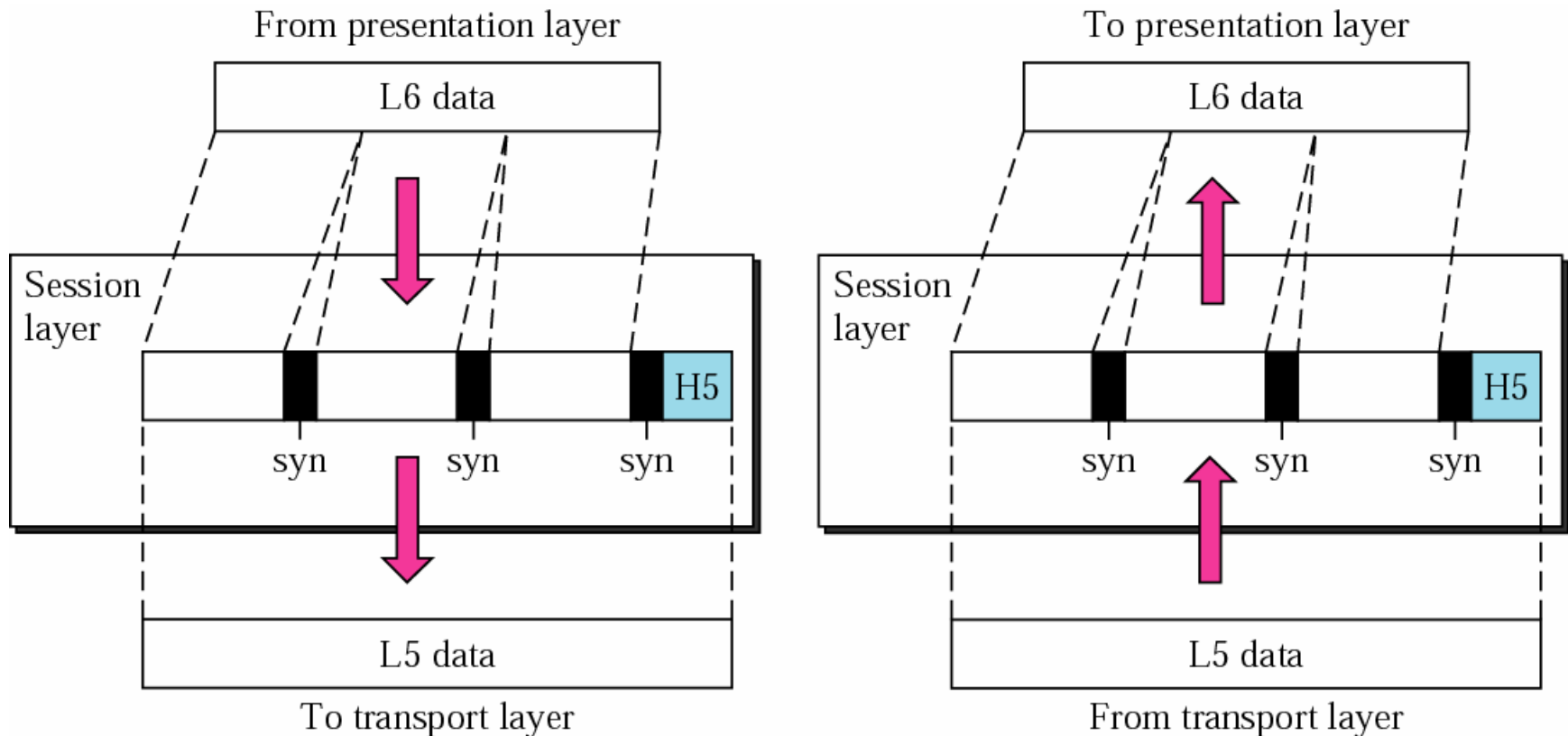
- Vận chuyển dữ liệu từ nơi gửi đến nơi nhận
- Thực hiện vận chuyển tin cậy, đúng thứ tự; kiểm soát lỗi/luồng

Phân phát thông điệp tin cậy từ hệ thống đầu cuối này đến hệ thống đầu cuối kia

Reliable end-to-end delivery of a message

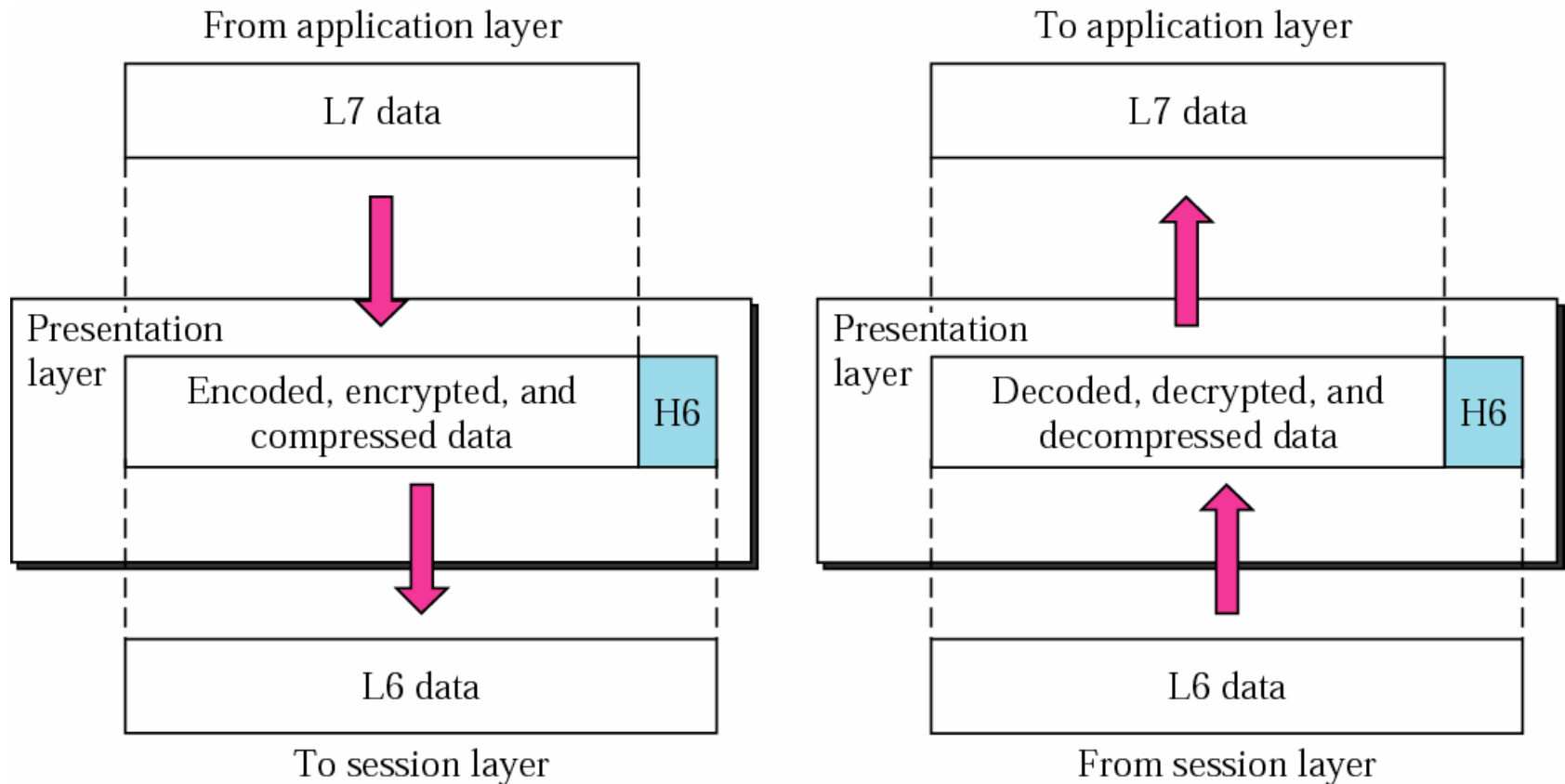


Tầng Phiên (session layer)



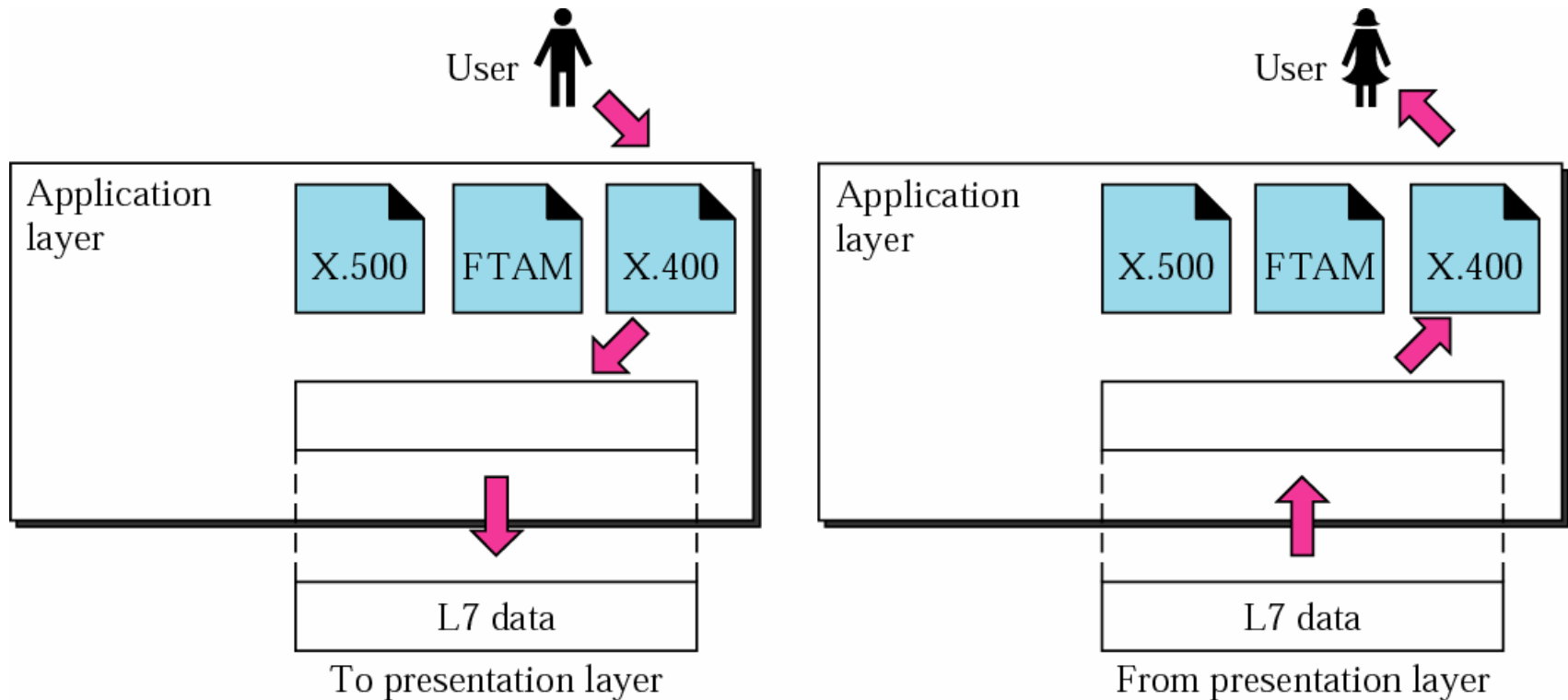
- Thiết lập phiên truyền thông (xác thực ...)
- Khôi phục phiên truyền khi gặp sự cố (phiên truyền bị đứt)

Tầng Trình diễn (presentation layer)



Chuyển đổi dữ liệu về một khuôn dạng chung

Tầng Ứng dụng (application layer)



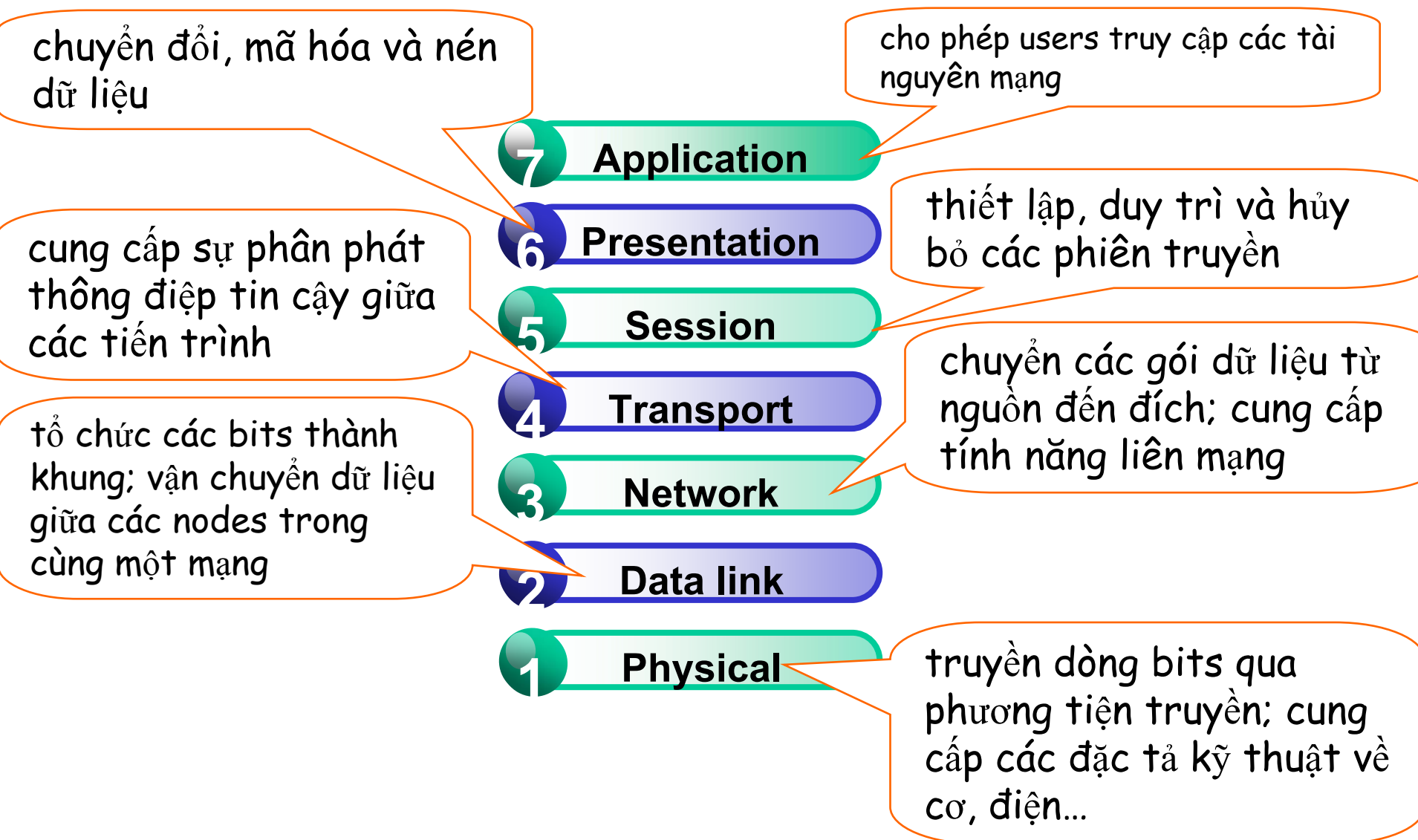
- Truyền thông giữa tiến trình – tiến trình
- Các tầng khác tồn tại để hỗ trợ tầng này

X500: Directory Service (Dịch vụ thư mục)

X400: hay là MHS Message Handling Systems

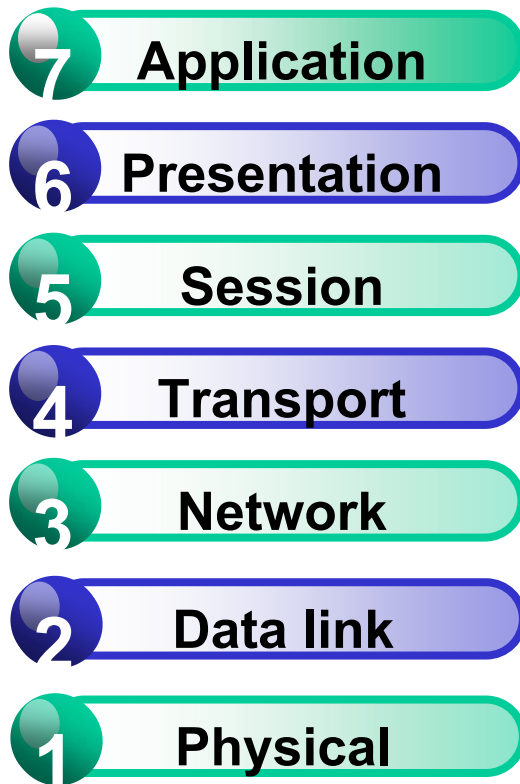
FTAM: File Transfer, Access and Management

Tóm tắt các tầng trong mô hình OSI

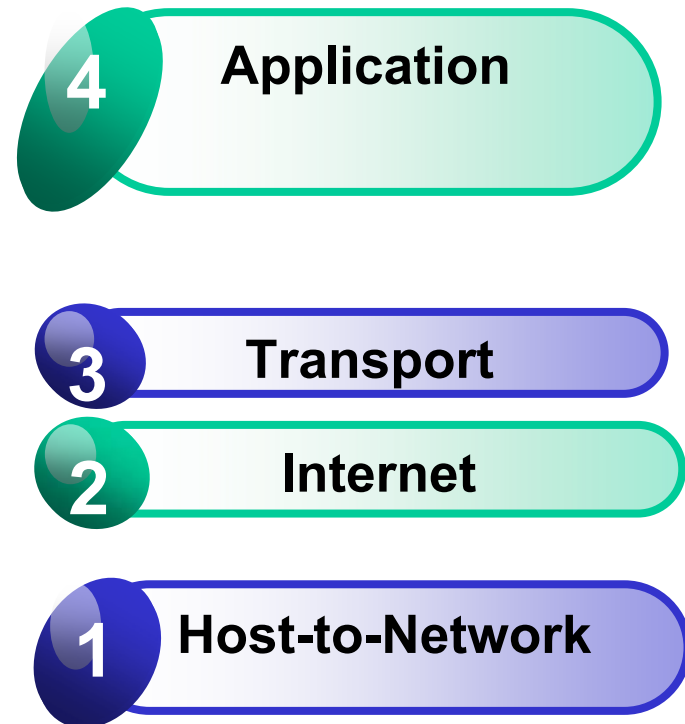


Mô hình tham chiếu TCP/IP

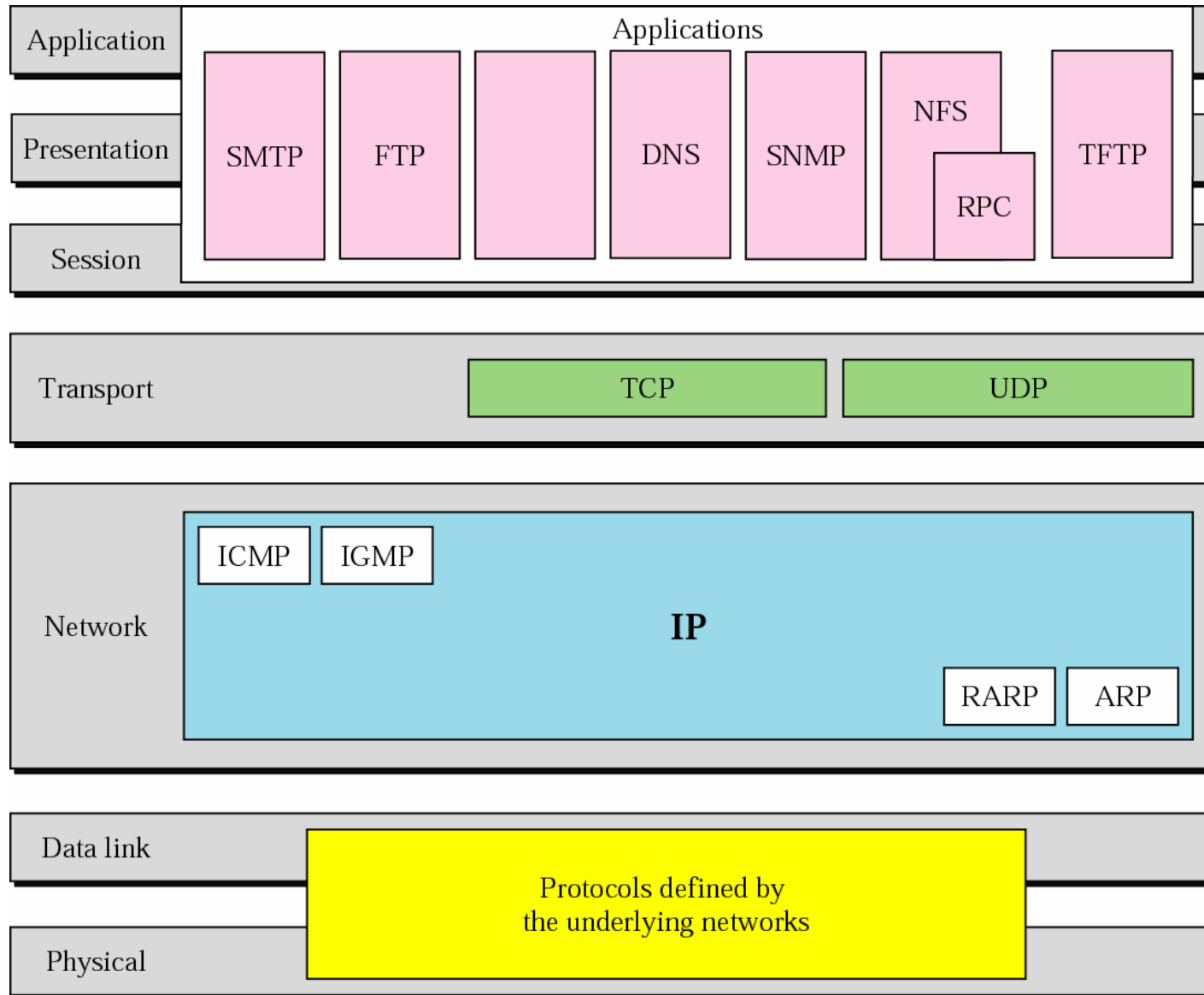
OSI Model



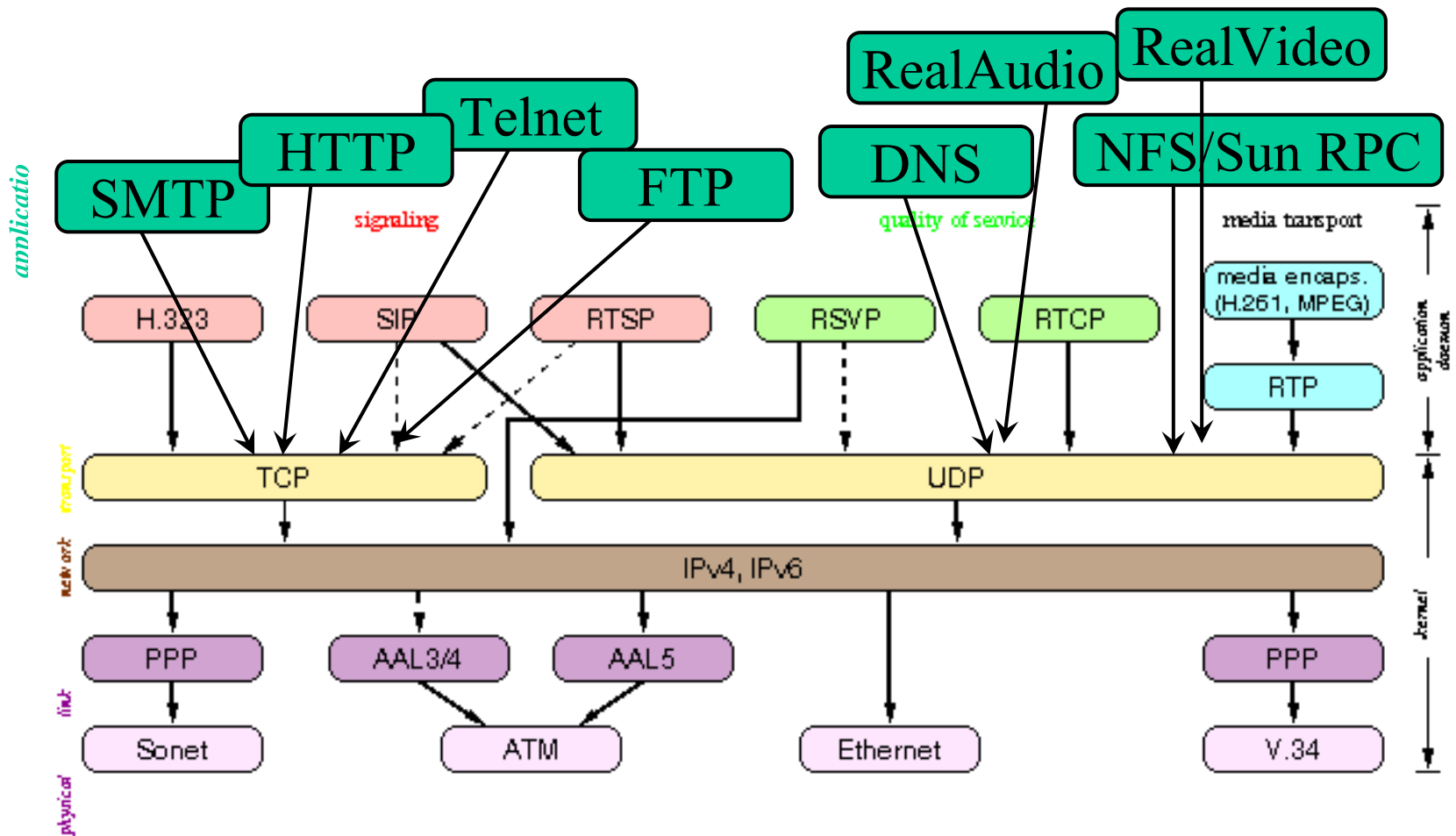
TCP/IP Model



TCP/IP and OSI model



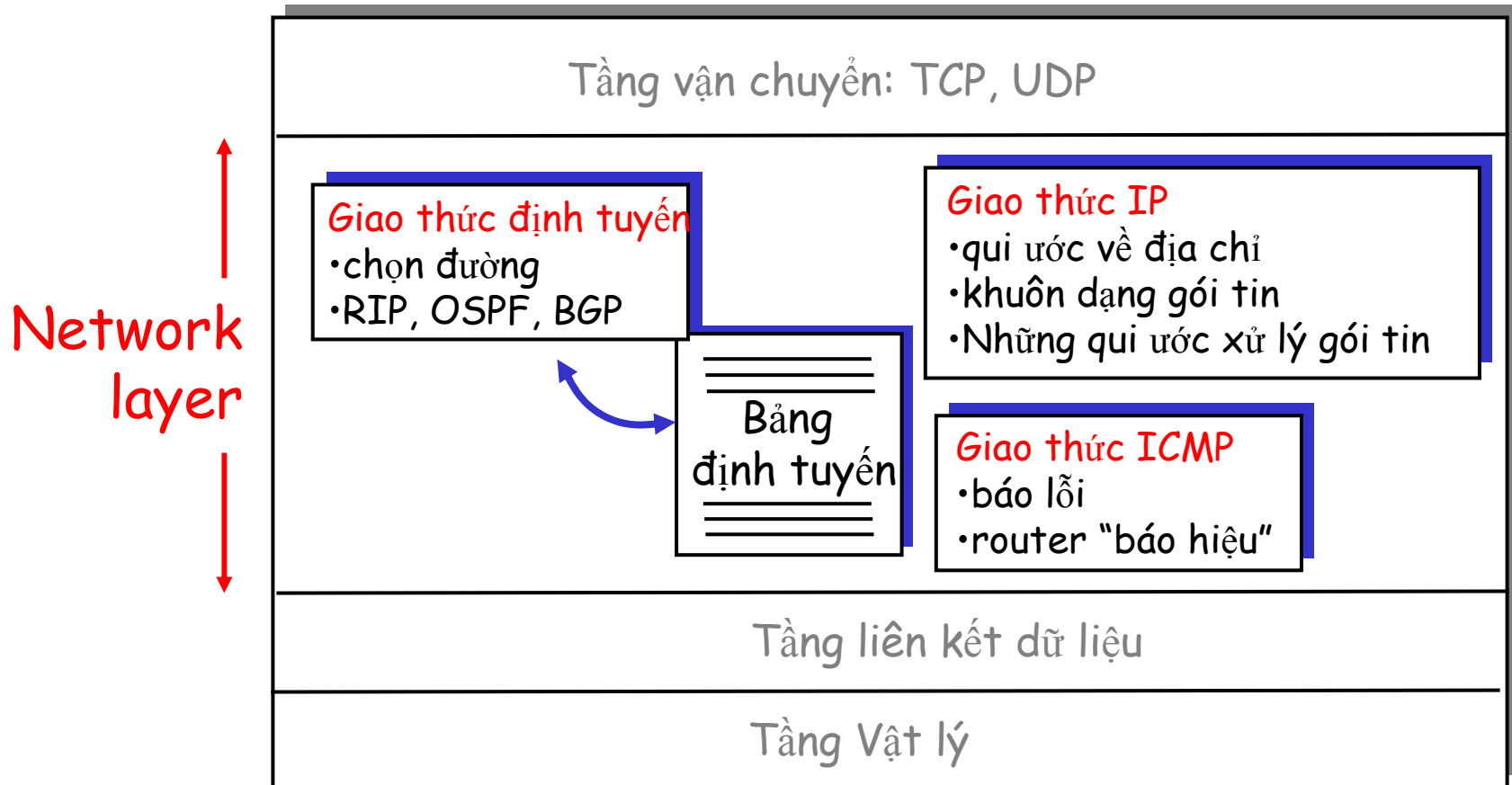
Internet Protocol "Zoo"



Tầng Internet

The Internet (Network) layer

Các thành phần (chức năng) chính của tầng mạng trên Internet
(được thực hiện tại các host và router)



Internet Protocol (IP)

- ❑ Tương kết các công nghệ mạng khác nhau và cung cấp sự kết nối toàn cầu trong một thế giới không đồng nhất.
- ❑ Tạo cảm giác về một mạng đơn đồng nhất.
- ❑ Gán địa chỉ logic duy nhất, toàn cầu cho mỗi host
- ❑ Phân giải địa chỉ
 - Ánh xạ từ địa chỉ logic sang địa chỉ vật lý để thực hiện sự phân phát các gói tin.

Internet Protocol

- ❑ Dịch vụ datagram phi kết nối, không tin cậy
- ❑ Các gói tin chứa địa chỉ nguồn và đích
- ❑ Các packets được truyền đi độc lập với nhau
- ❑ Các packets có thể bị mất
 - Phục hồi lỗi tùy thuộc vào các giao thức đầu cuối (end-to-end protocols)

Vận chuyển giữa các nút trong một mạng

- ❑ Sử dụng các cơ chế truyền dữ liệu của tầng liên kết dữ liệu
 - Example: Ethernet, Token Ring, PPP
- ❑ Ánh xạ từ địa chỉ logic (IP address) sang địa chỉ vật lý (MAC address)
 - Address Resolution Protocol (ARP) - giao thức phân giải địa chỉ

Các giao thức vận chuyển

Giao thức TCP:

- ❑ *Hướng kết nối:* phải thiết lập kết nối giữa client, server
- ❑ *Vận chuyển tin cậy* giữa bên gửi và nhận
- ❑ *Kiểm soát luồng (flow control):* bên gửi không làm ngập bên nhận
- ❑ *Kiểm soát tắc nghẽn:* điều chỉnh tốc độ bên gửi khi mạng quá tải

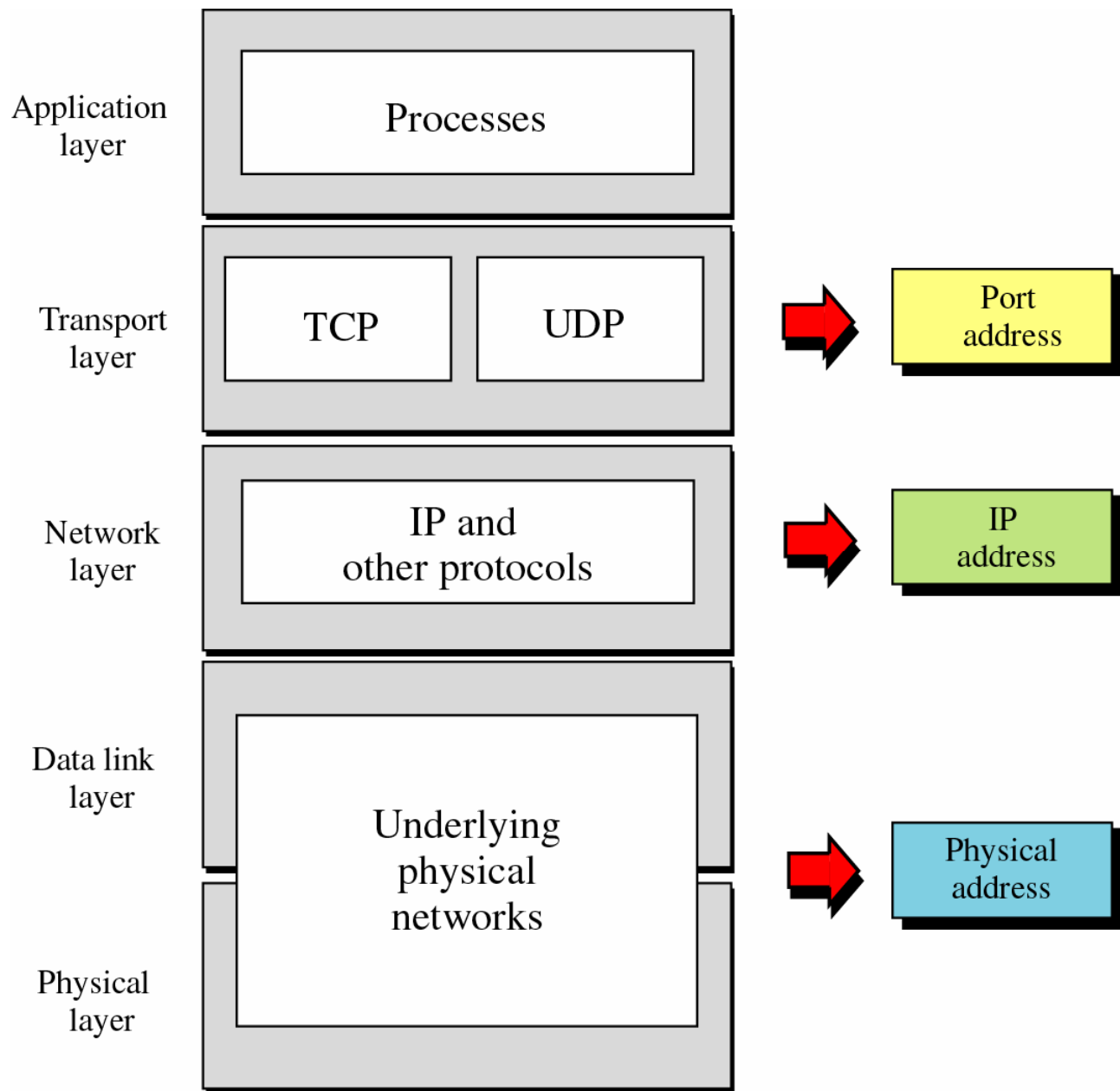
Giao thức UDP:

- ❑ Vận chuyển dữ liệu không tin cậy giữa bên gửi và nhận
- ❑ Không cung cấp: thiết lập kết nối, kiểm soát luồng, kiểm soát tắc nghẽn

Tại sao dùng UDP?

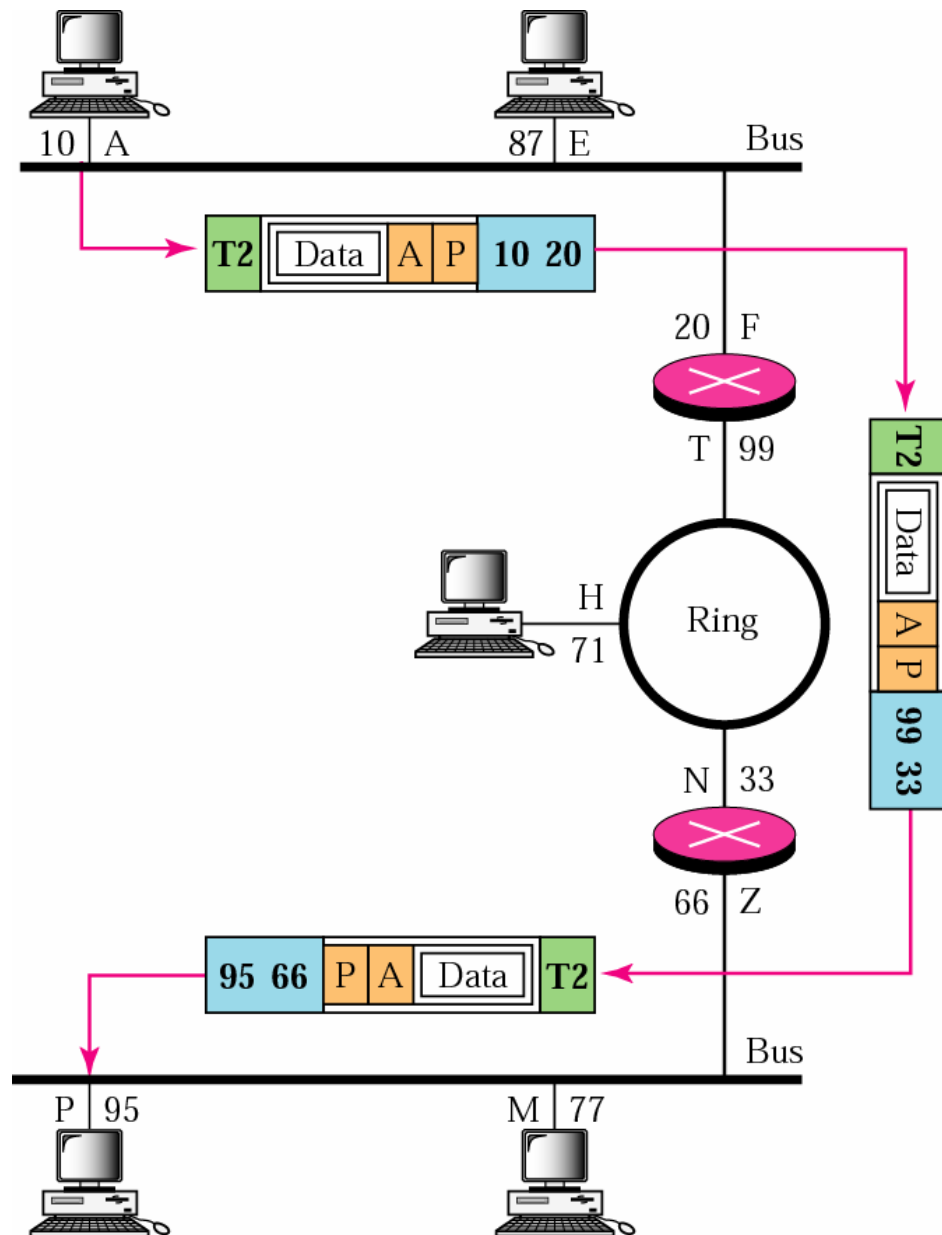
Triết lý của Internet

- ❑ Mạng chỉ cung cấp những dịch vụ thiết yếu nhất
 - IP chuyển dữ liệu phi kết nối, không tin cậy
- ❑ Các chức năng gia tăng khác được thực hiện ở các đầu cuối
 - Phục hồi lỗi và kiểm soát luồng được thực hiện bởi TCP
- ❑ Ứng dụng đầu cuối biết nhiều hơn
 - Mất packet có thể chấp nhận được đối với voice
- ❑ Mọi sự kiểm soát được đưa về các hệ thống đầu cuối.

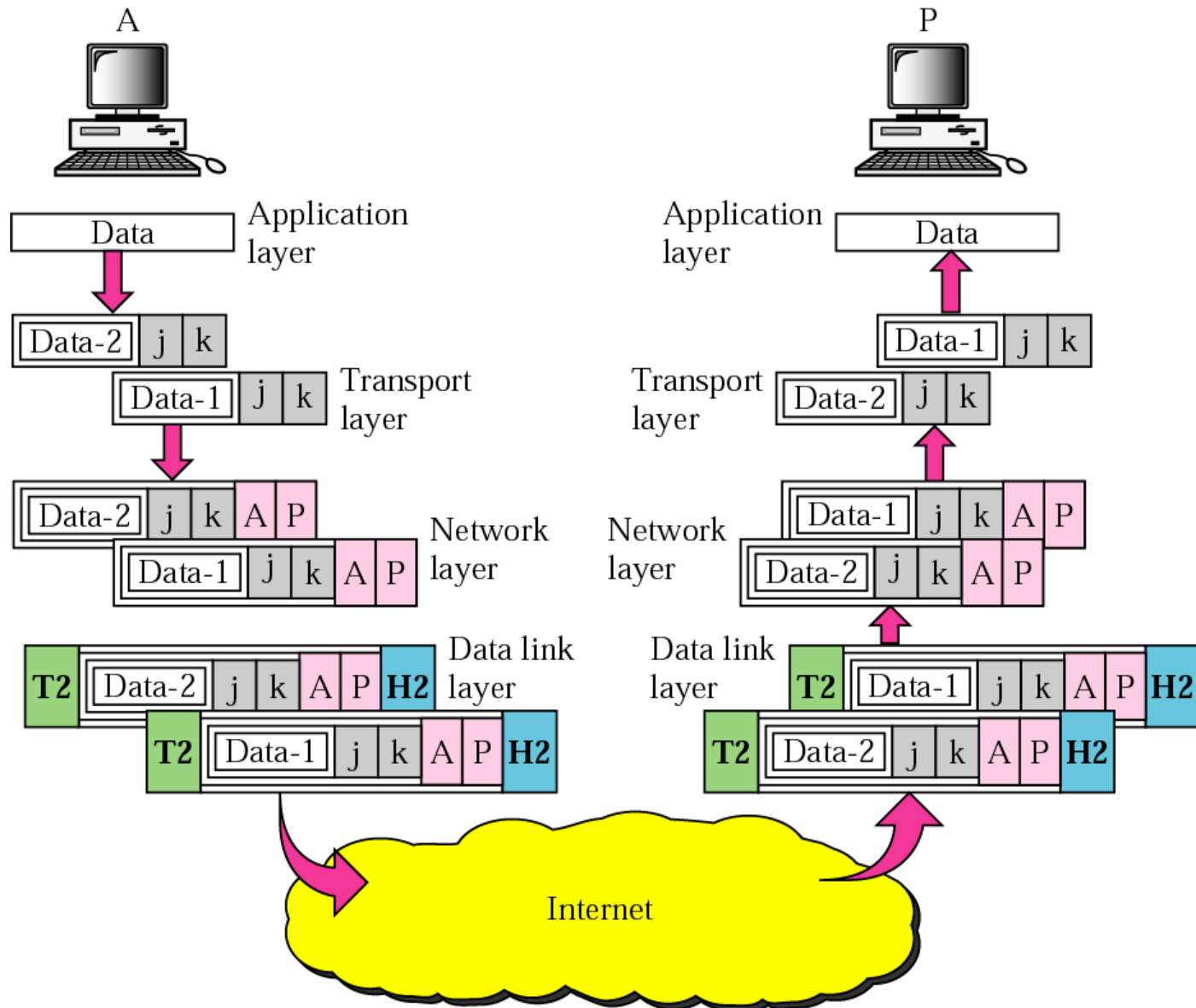


**Mối quan
hệ giữa
các tầng
và địa chỉ
trong
TCP/IP**

Địa chỉ IP (IP addresses)



Địa chỉ cổng (port addresses)



Mô hình Client-Server

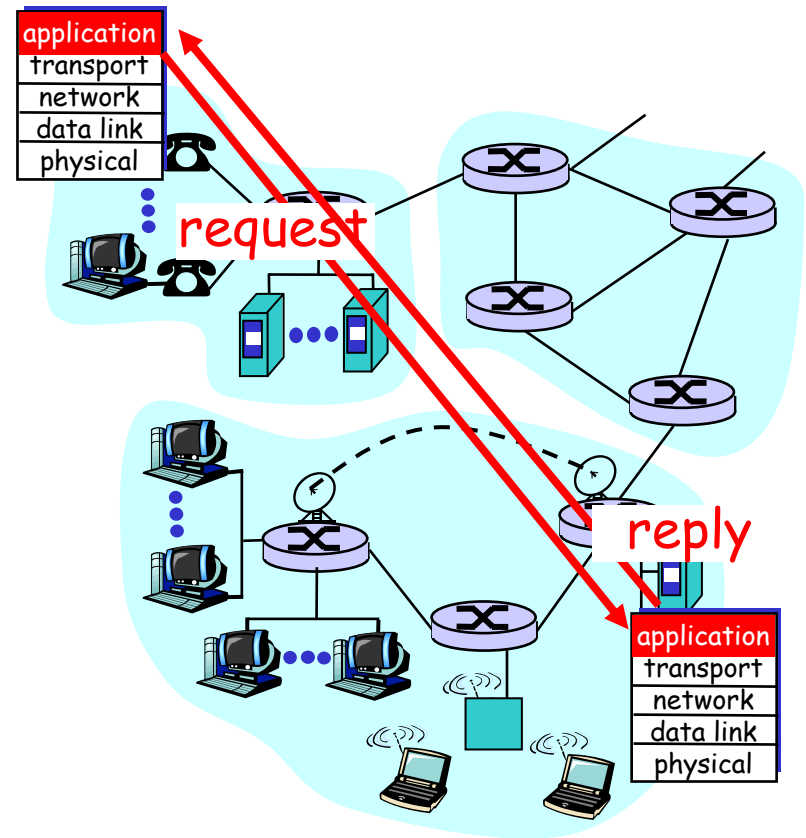
Thông thường một ứng dụng mạng gồm 2 phần:
client và *server*

Client:

- ❑ khởi tạo kết nối với server ("nói trước")
- ❑ thường là yêu cầu dịch vụ nào đó từ server

Server:

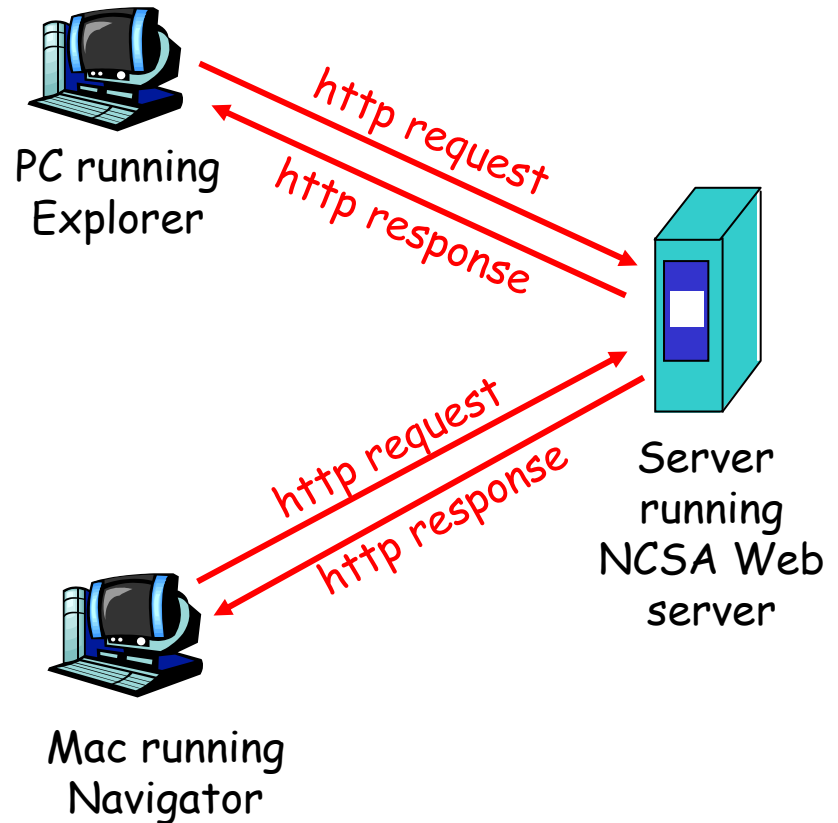
- ❑ Cung cấp dịch vụ được yêu cầu cho client



Ví dụ: Dịch vụ Web (giao thức HTTP)

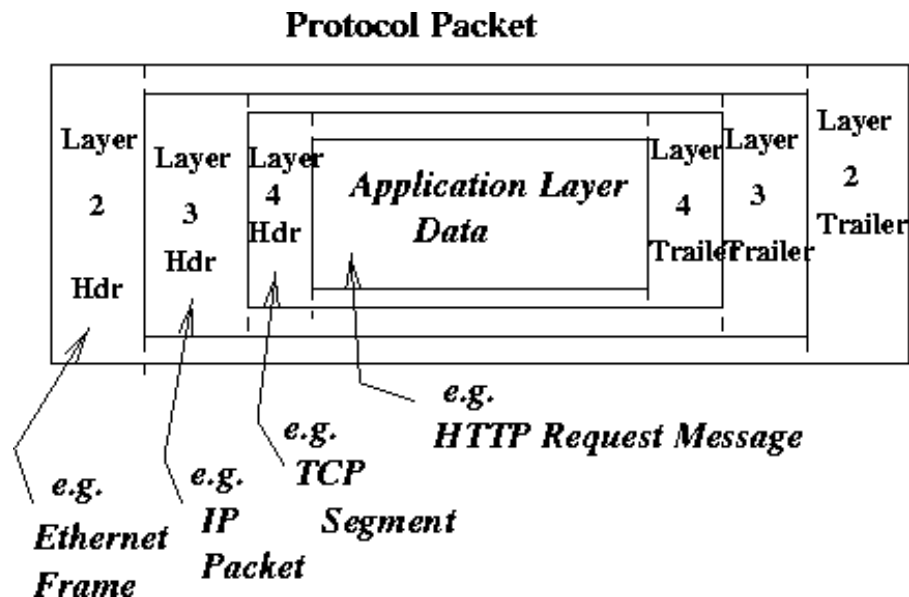
HyperText Transfer Protocol

- giao thức tầng ứng dụng cho dịch vụ Web
- mô hình client/server
 - *client*: là trình duyệt, nó yêu cầu, nhận và "hiển thị" các đối tượng Web
 - *server*: là Web server, nó gửi các đối tượng để trả lời cho các yêu cầu từ client

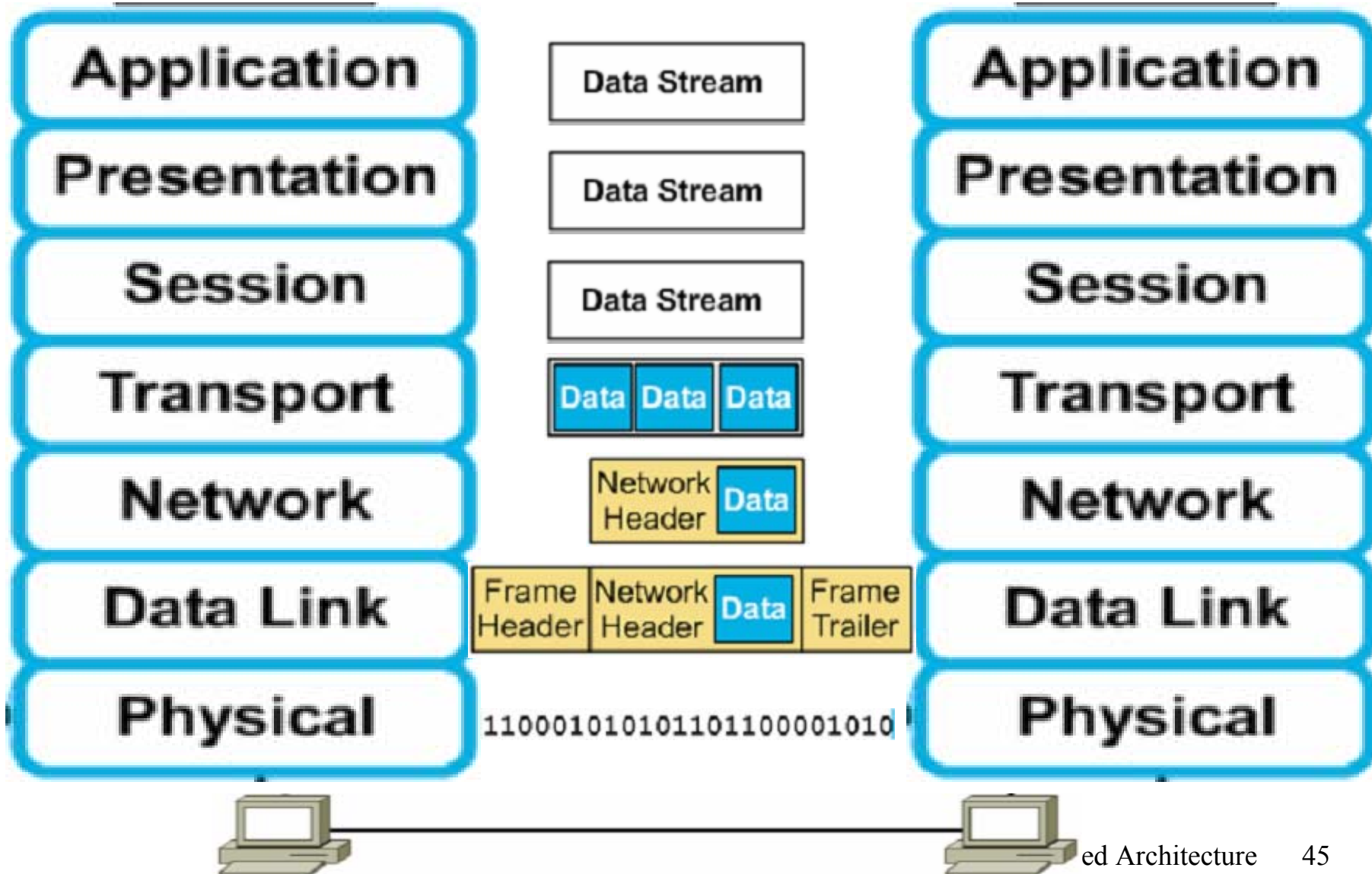


Đơn vị dữ liệu giao thức và đơn vị dữ liệu dịch vụ

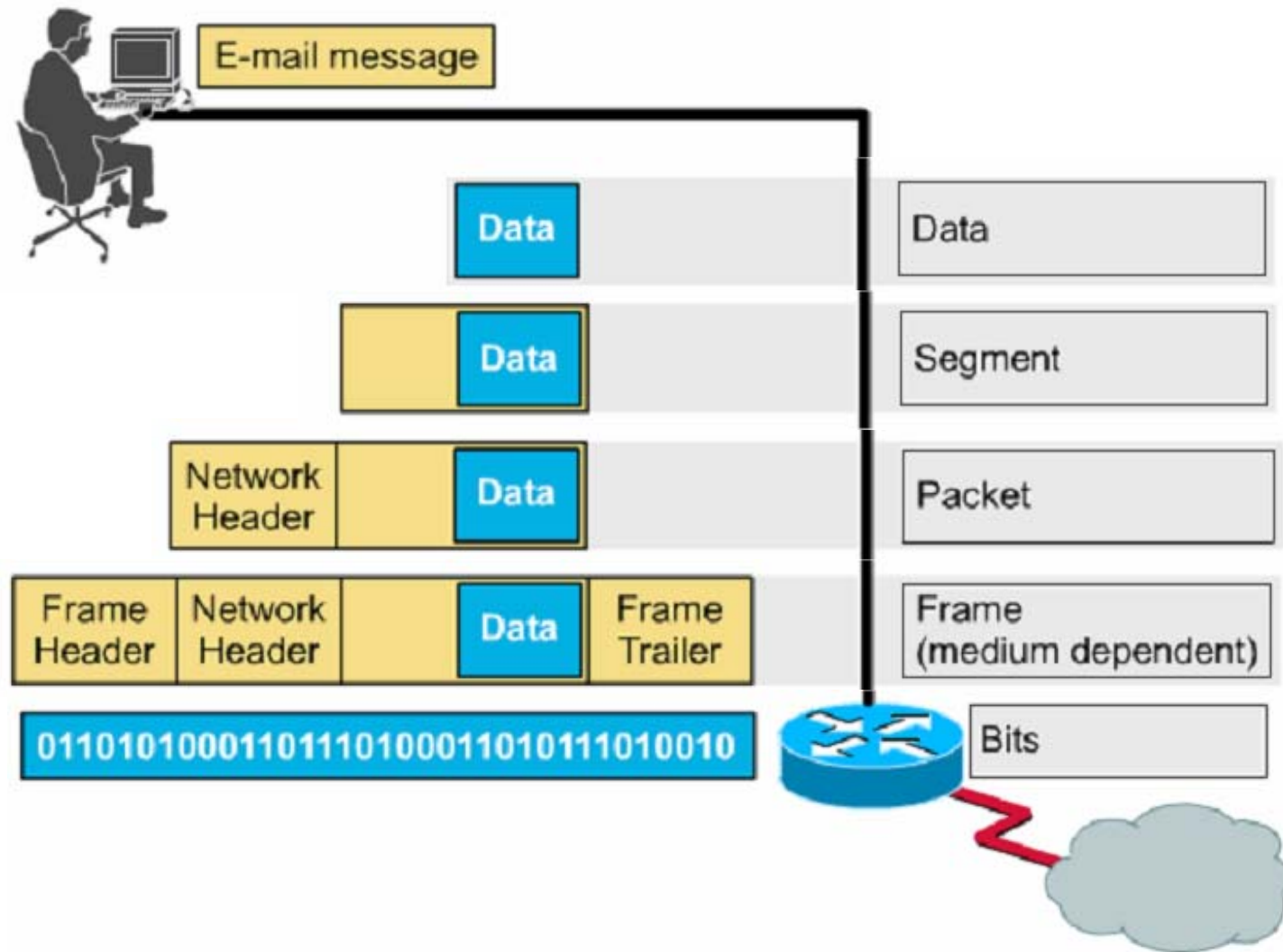
- Đơn vị dữ liệu giao thức - Protocol data units (PDUs): các gói dữ liệu được trao đổi giữa các thực thể đồng tầng
- Đơn vị dữ liệu dịch vụ - Service data units (SDUs): các gói dữ liệu do tầng phía trên đưa xuống
- Dữ liệu tại một tầng được bao bọc trong gói (packet) tại tầng dưới
 - *Envelope within envelope: PDU = SDU + header or (optional) trailer*



Minh họa quá trình đóng gói



Một ví dụ về sự đóng gói: gửi E-mail



Nhận xét về việc phân tầng

□ Ưu điểm

- Module hóa giúp cho việc duy trì và cập nhật dễ dàng hơn

□ Nhược điểm

- Tầng nào nên thực hiện chức năng gì?
 - Dựa trên nền tảng hop-by-hop hay end-to-end
- Sự trùng lặp chức năng giữa các tầng
 - Tầng liên kết dữ liệu và tầng vận chuyển đều có chức năng phục hồi lỗi

So sánh giữa OSI và TCP/IP

□ Giống nhau:

- ✓ Một chồng giao thức độc lập
- ✓ Chức năng của các tầng na ná giống nhau (roughly similar)

□ Khác nhau (về mặt mô hình tham chiếu):

- ✓ Số tầng (7 vs. 4)
- ✓ OSI phân biệt rõ: services, interfaces, protocols => protocols được che dấu tốt hơn và dễ dàng thay thế hơn khi công nghệ thay đổi
- ✓ OSI: mô hình được nghĩ ra trước, protocols được phát minh sau
- ✓ OSI: tầng mạng hỗ trợ cả hai connectionless & connection-oriented communication; tầng vận chuyển chỉ hỗ trợ connection-oriented communication