



COMPUTER FIFTH EDITION  
NETWORKING

*A Top-Down Approach*

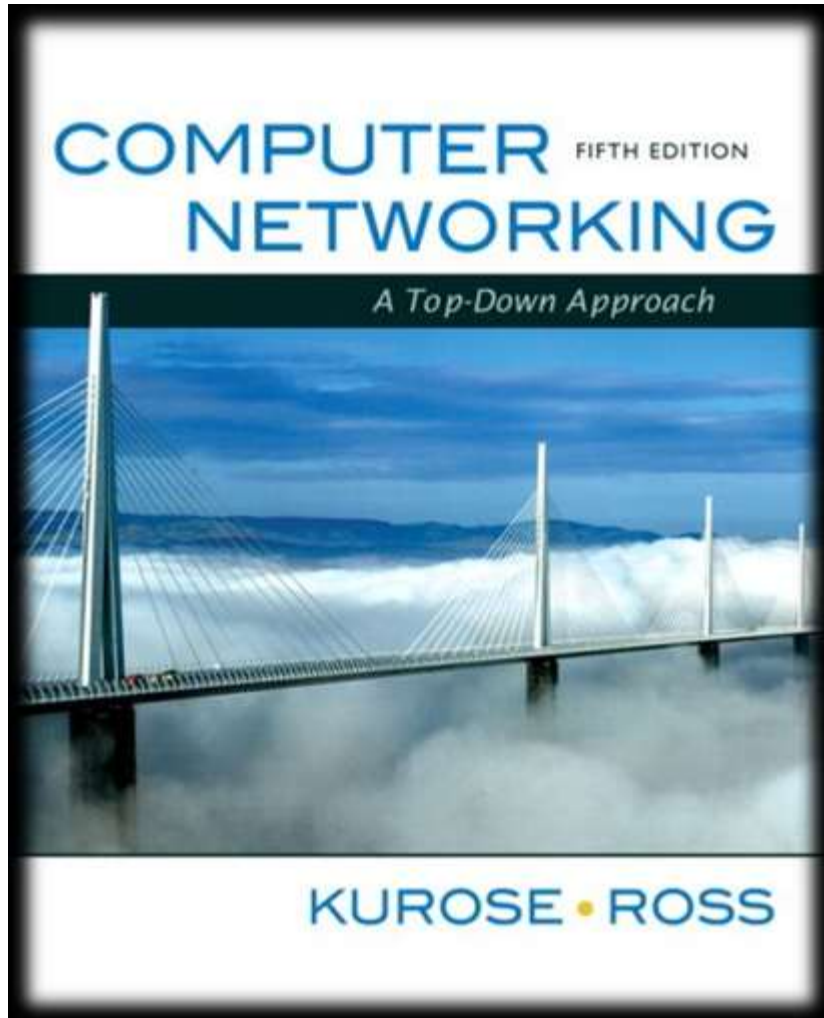


KUROSE • ROSS

Slides bài giảng  
**NHẬP MÔN MẠNG MÁY TÍNH**

Nhập môn mạng máy tính

# Nội dung môn học



- ❑ Ch1. Giới thiệu
- ❑ Ch2. Lớp Application
- ❑ Ch3. Lớp Transport
- ❑ Ch4. Lớp Network
- ❑ Ch5. Lớp Link & các mạng LAN

# Đánh giá

- ❑ Chuyên cần: 10%
- ❑ Kiểm tra: 20%
- ❑ Trắc nghiệm + vấn đáp: 70%



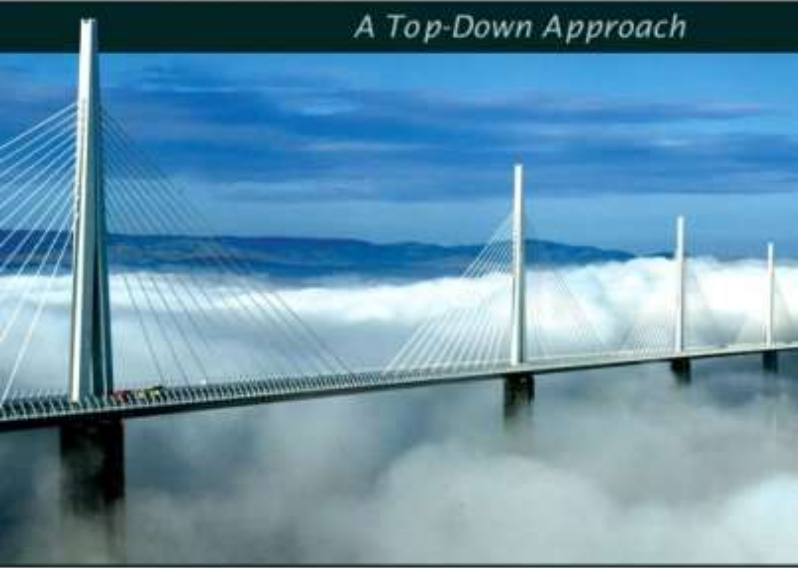
# Tài liệu tham khảo

- ❑ *Giáo trình Nhập môn mạng máy tính, Hồ Đắc Phương, NXB Giáo dục, 8/2011*
- ❑ *Computer Networking: A Top Down Approach Featuring the Internet, 3<sup>rd</sup> edition. Jim Kurose, Keith Ross Addison-Wesley, July 2004*  
(<http://www.mediafire.com/?j0p1pe5029250>)

# COMPUTER NETWORKING

FIFTH EDITION

*A Top-Down Approach*



KUROSE • ROSS

## Chương 1 – Giới thiệu MẠNG MÁY TÍNH VÀ NHỮNG KHÁI NIỆM MỞ ĐẦU

Nhập môn mạng máy tính

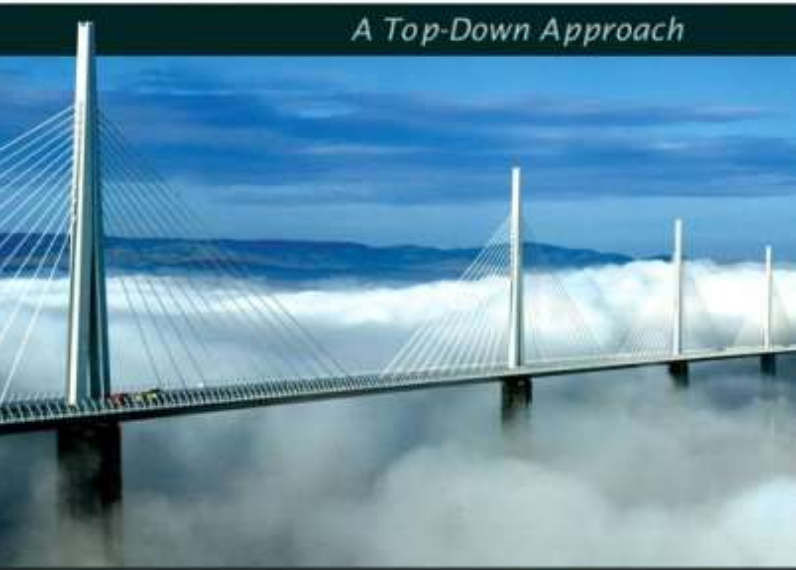
# Nội dung

- 1.1 - Mạng máy tính và ứng dụng trong đời sống
- 1.2 - Phân loại mạng (network taxonomy)
- 1.3. - Giao thức mạng (Protocol)
- 1.4 - Các mô hình tham chiếu (reference models).
- 1.5 - Chuẩn mạng máy tính (network standards).
- 1.6 - Hệ điều hành trong môi trường mạng.
- 1.7 - Mạng Internet.
- 1.8 - Một số mô hình mạng
- 1.9 - Lịch sử phát triển Internet

# Nội dung

## COMPUTER NETWORKING FIFTH EDITION

*A Top-Down Approach*



KUROSE • ROSS

1.1 - Mạng máy tính và ứng dụng trong đời sống.

# 1.1. Mạng máy tính và ứng dụng

- ❑ Mạng máy tính (computer network) là hệ thống bao gồm nhiều hệ máy tính đơn lẻ (nút mạng) được *kết nối* với nhau theo *kiến trúc* nào đó và có khả năng trao đổi thông tin.
  - Nút mạng (node): host, workstation, network component...
  - Kết nối (interconnected): *dây (cable), sóng (wave)*...
  - Kiến trúc (architecture): Chỉ ra phương thức xây dựng và hoạt động của mạng.
- ❑ Lợi ích của mạng:
  - Chia sẻ, trao đổi thông tin.
  - Tăng cường sức mạnh của hệ thống (distributed system, parallel system).

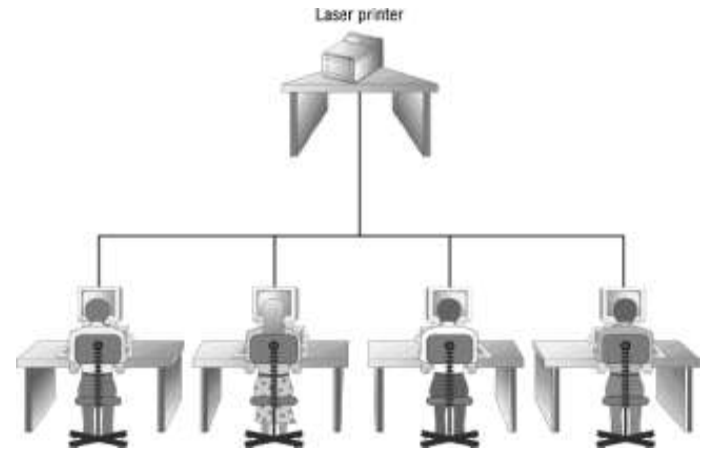
# Ứng dụng của mạng máy tính

- ❑ Mạng nội bộ (cơ quan, toà nhà)
  - Chia sẻ tài nguyên (máy in, ổ cứng, chương trình...).
  - Liên lạc trong mạng nội bộ cơ quan (local mail).
- ❑ Cung cấp dịch vụ (mô hình client/server).
  - Web, Email, search engine, tin tức.
  - Thương mại điện tử (ecommerce - electronic commerce).
- ❑ People online communication.
  - Chatting, conference
  - Điện thoại (PSTN, Mobile).

# Chia sẻ tài nguyên máy in



*CS before 2003*



*CS since 2003*

# Các thành phần của kiến trúc mạng máy tính

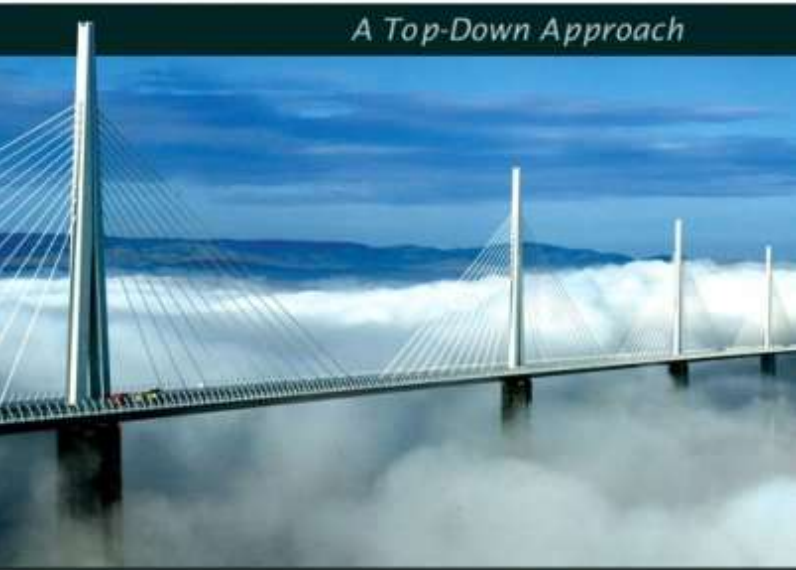
- ❑ Đường truyền vật lý (physical media)
  - Truyền *tín hiệu* giữa các hệ thống.
  - Hữu tuyến (cable) và vô tuyến (wireless).
  - Băng thông (bandwidth)
  - Tốc độ (speed) hay thông lượng (throughput):
    - ❖ Số lượng bit truyền được trong một giây (bps).
    - ❖ Số lượng thay đổi tín hiệu trong một giây (baud).
- ❑ Kiến trúc mạng (network architecture)
  - Hình trạng mạng (topology).
  - Giao thức (protocol).



# Nội dung

## COMPUTER NETWORKING FIFTH EDITION

*A Top-Down Approach*



KUROSE • ROSS

1.2 - Phân loại mạng (network taxonomy).

# 1.2. Phân loại (network classification, taxonomy)

## □ Theo topo mạng

- Broadcast: Truyền quảng bá.
- Point-to-point (switched): Truyền điểm-điểm.

## □ Theo kỹ thuật truyền (transmission technique)

- Circuit-switched
- Packet-switched

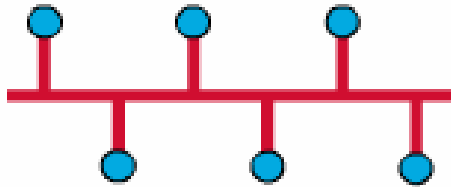
## □ Theo quy mô (scale)

- LAN
- MAN
- WAN

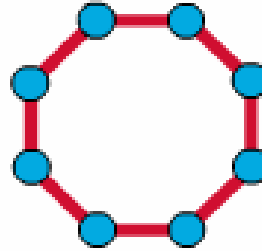


Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	
100 m	Building	
1 km	Campus	Local area network
10 km	City	
100 km	Country	Metropolitan area network
1000 km	Continent	
10,000 km	Planet	Wide area network
		The Internet

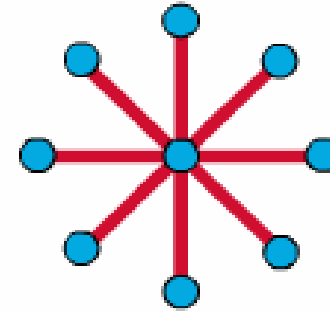
# Topology (broadcast & point-to-point)



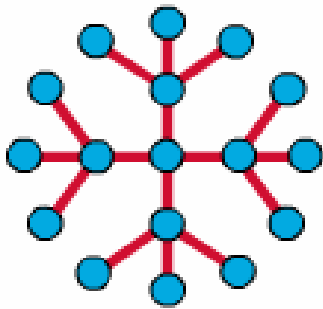
**Bus Topology**



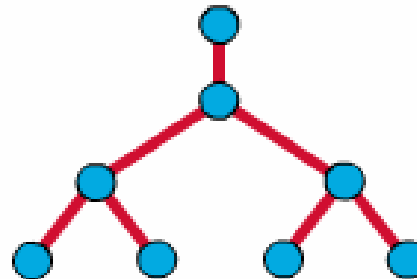
**Ring Topology**



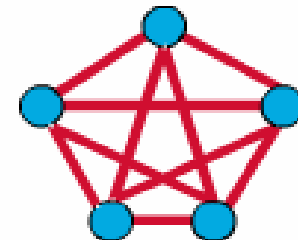
**Star Topology**



**Extended Star  
Topology**



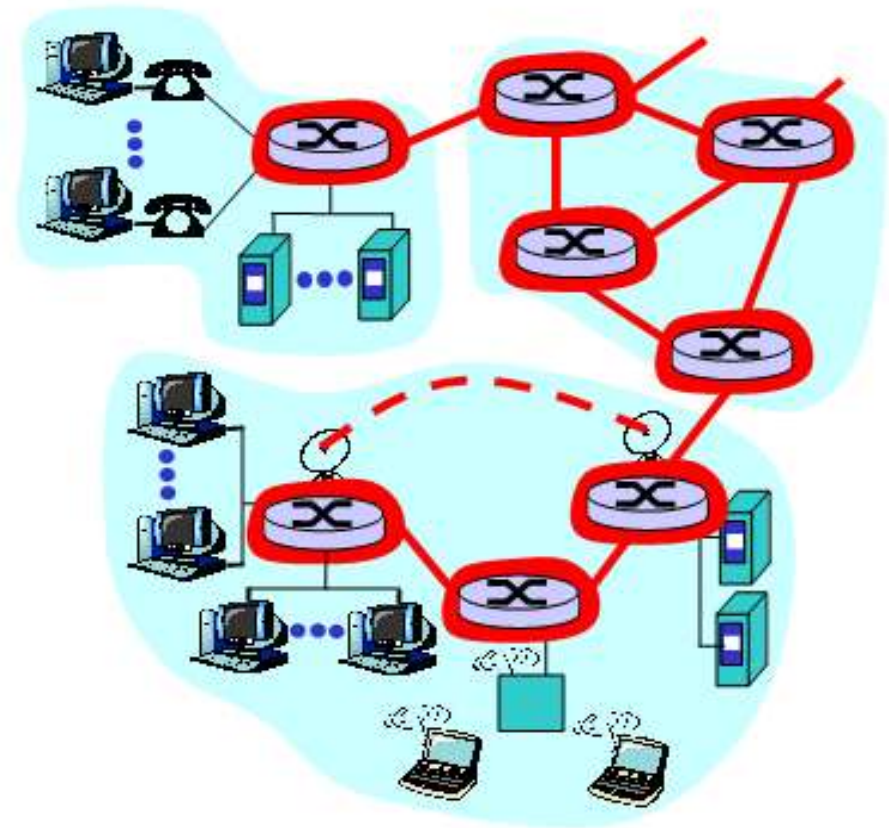
**Hierarchical  
Topology**



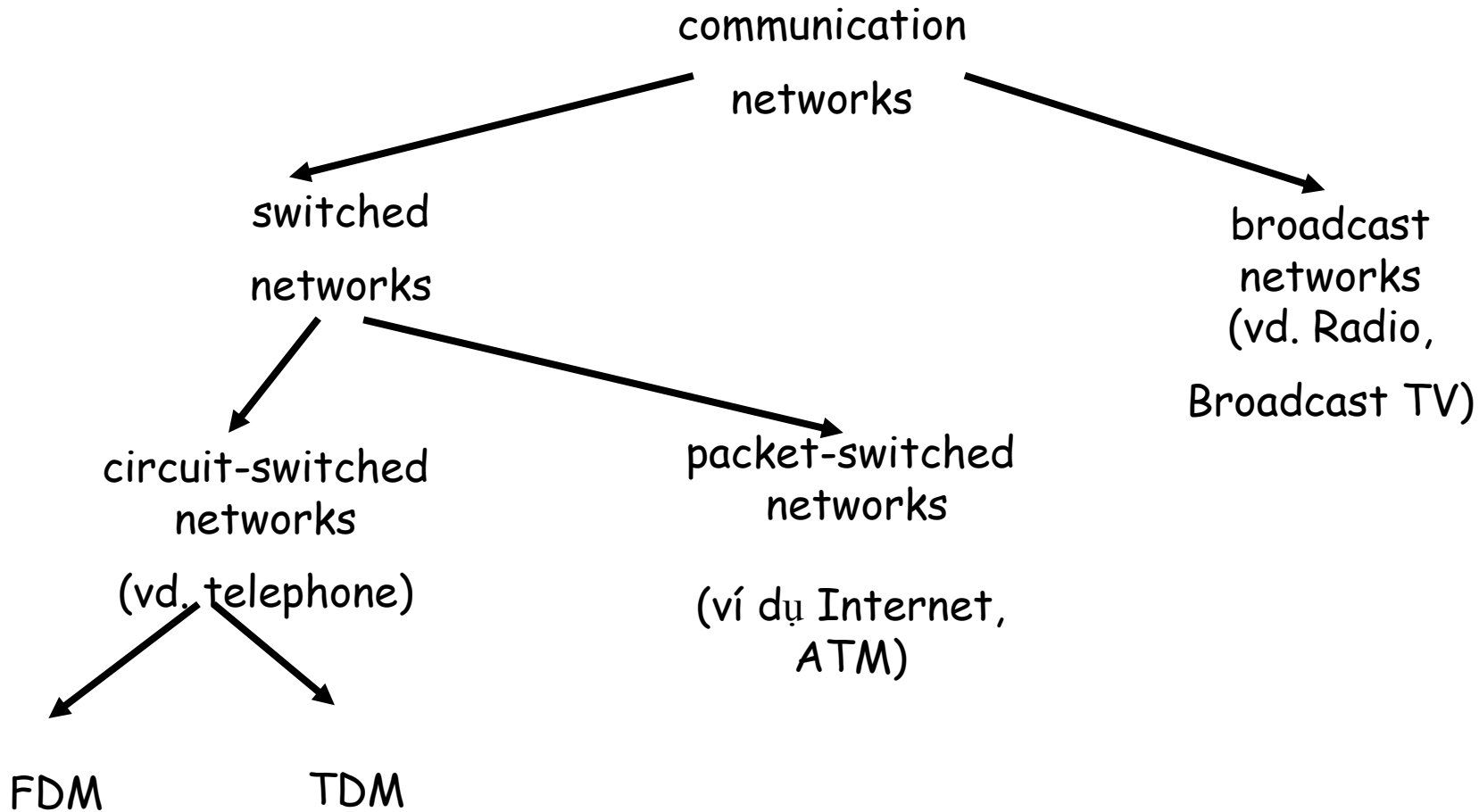
**Mesh  
Topology**

# Vận chuyển dữ liệu qua mạng

- ❑ Mạng máy tính gồm 2 phần
  - Lớp "Rìa"
  - Lớp "Lõi"
- ❑ Lớp lõi có nhiệm vụ chuyển dữ liệu từ thiết bị đầu cuối đến thiết bị đầu cuối, có 2 cách chuyển
  - Chuyển mạch ảo
  - Chuyển mạch gói

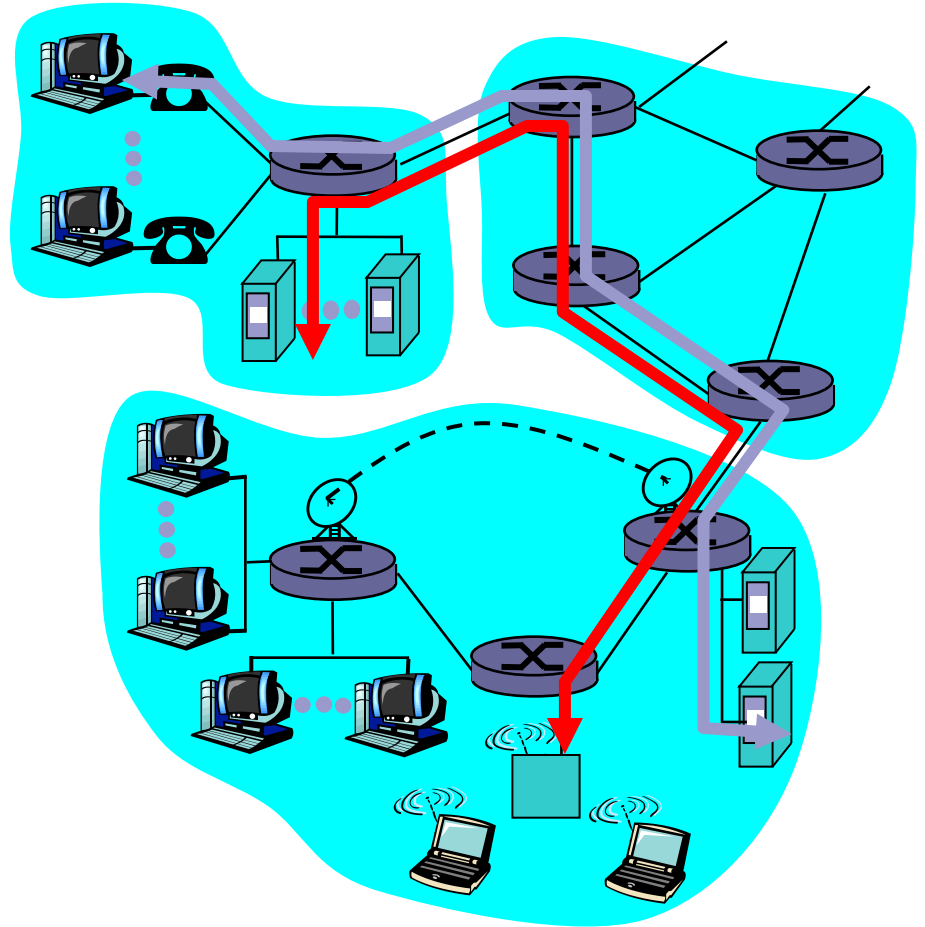


# Truyền dữ liệu trong mạng



# Circuit switching (chuyển mạch kênh, ảo)

- ❑ Khi hai nút muốn trao đổi thông tin → thiết lập kênh (circuit).
- ❑ Kênh được giữ riêng cho hai nút cho tới khi kết thúc phiên trao đổi.
- ❑ VD: Mạng điện thoại.

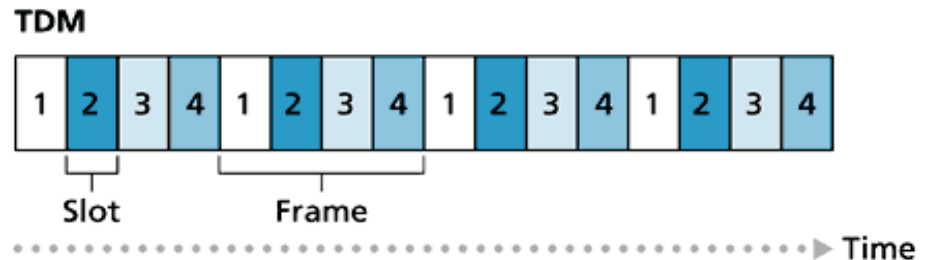
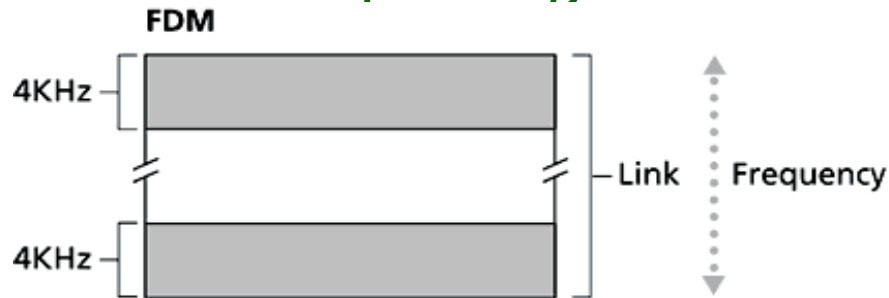


# Circuit switching: FDM & TDM


□ Hai phương pháp chia sẻ liên kết (link) trong circuit switching:

➤ FDM (Frequency Division Multiplexing).

➤ TDM (Time DM).



Key:

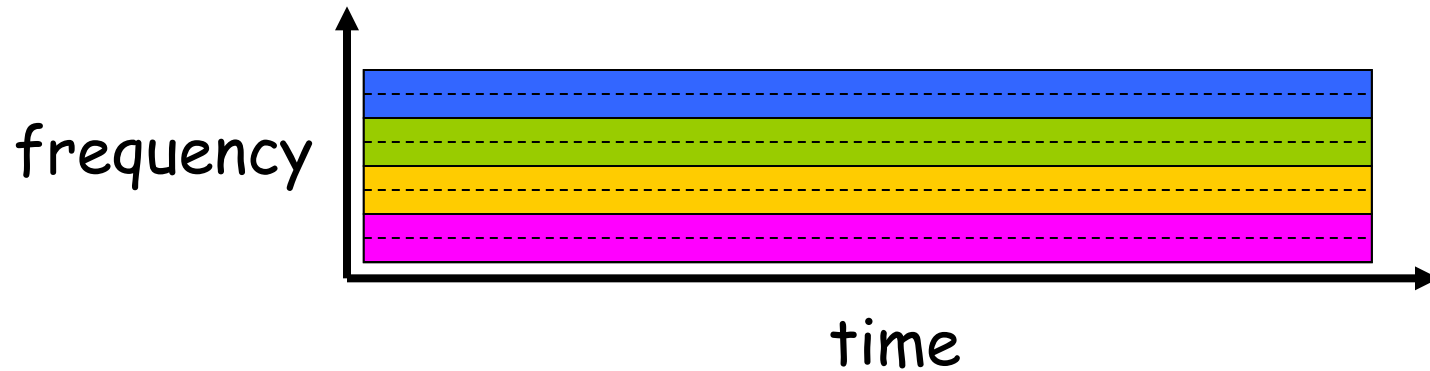
 All slots labeled "2" are dedicated to a specific sender-receiver pair.

# FDM & TDM example

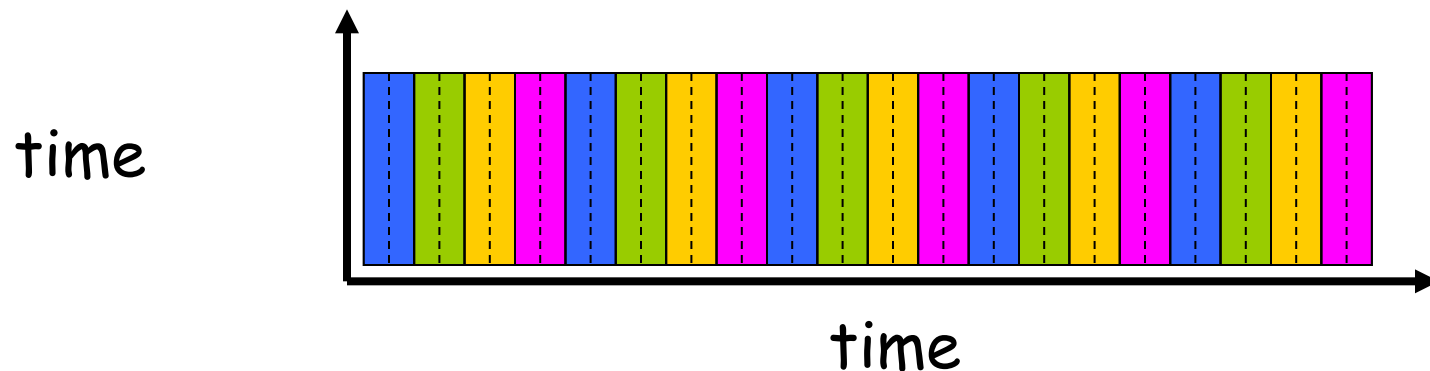
FDM

Example:

4 users

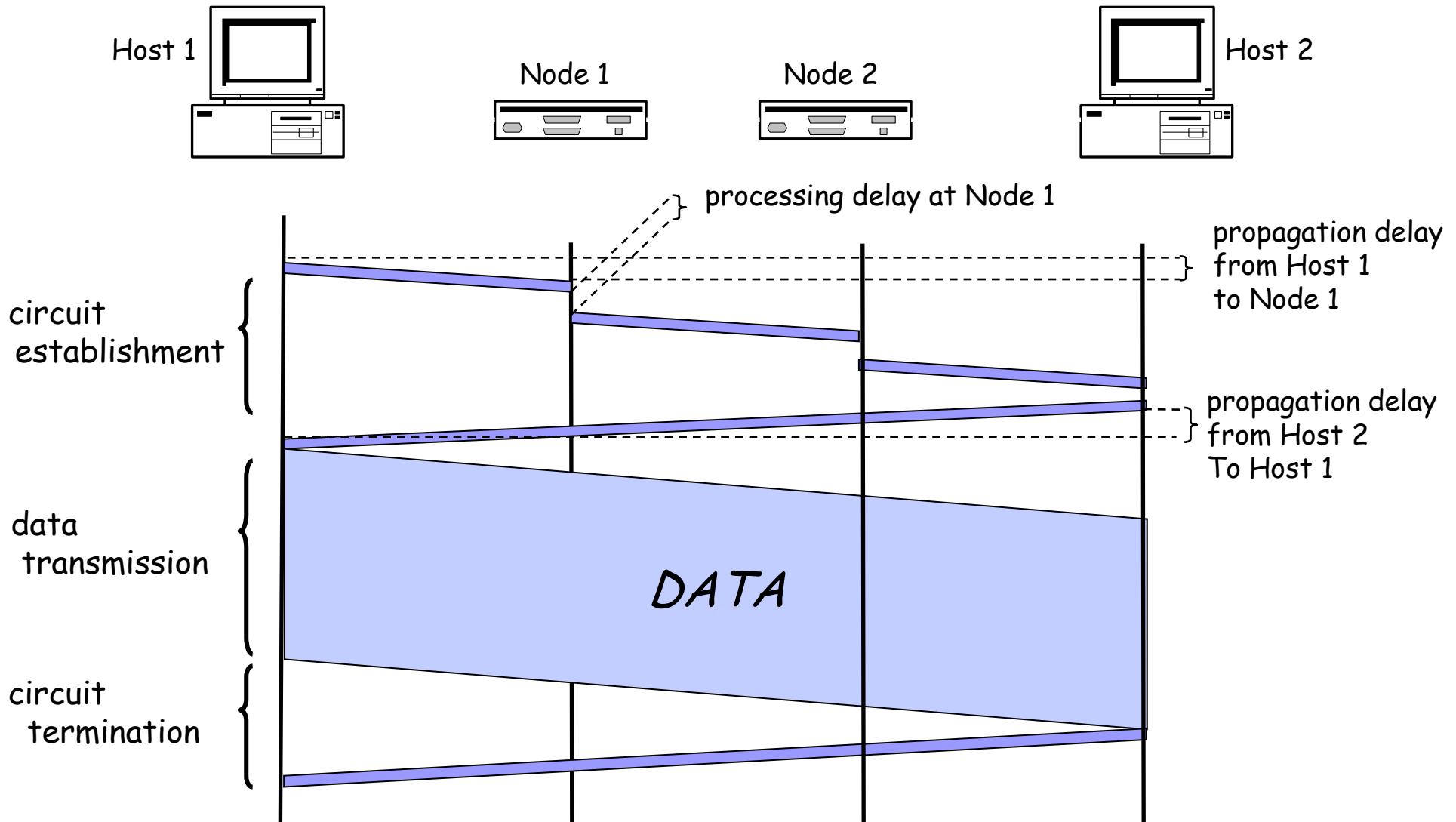


TDM

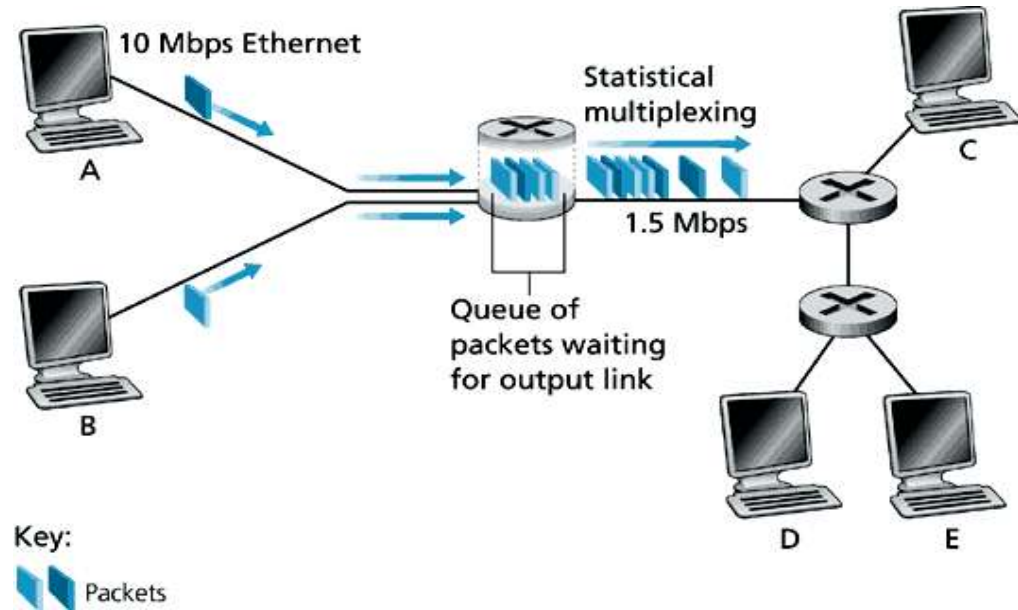
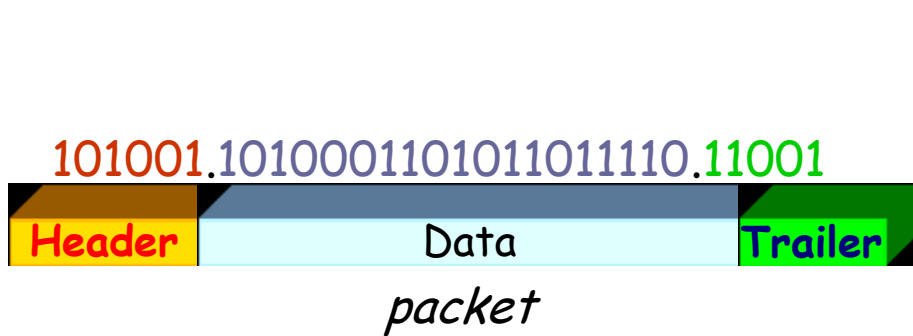




# Các giai đoạn của circuit switching

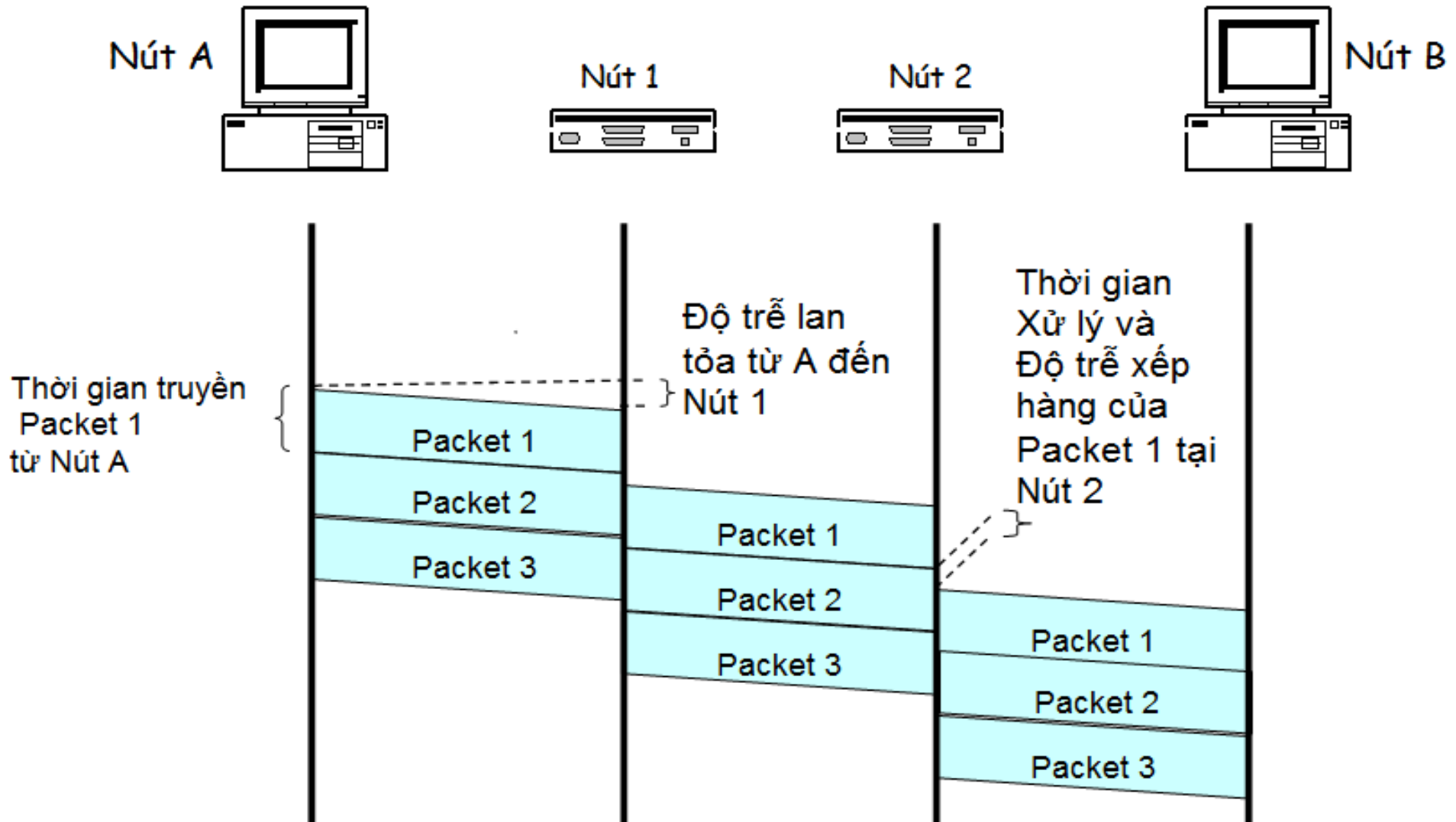


# Packet switching

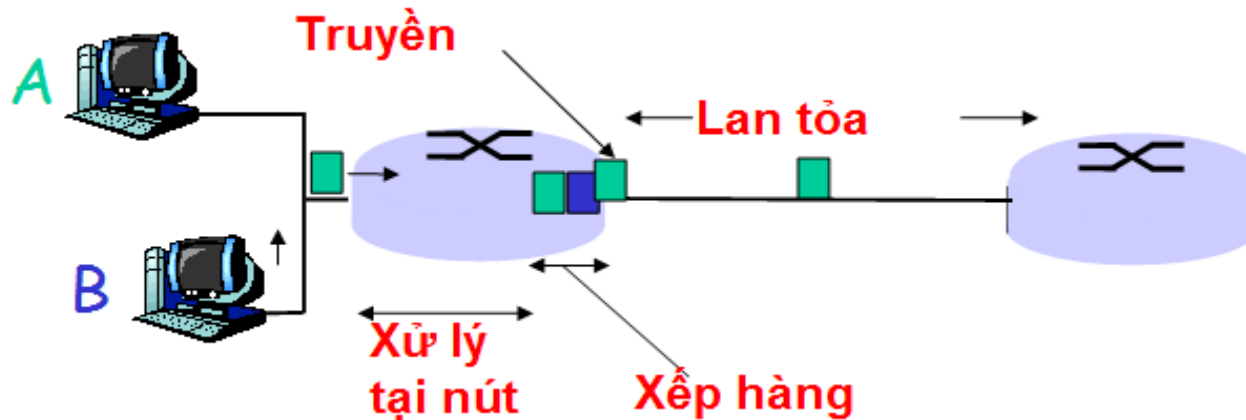


- ❑ Dữ liệu được chia thành các gói tin (packet). Mỗi gói đều có phần thông tin điều khiển (header, trailer) cho biết nguồn gửi, đích nhận...
- ❑ Các gói tin có thể đến và đi theo những đường khác nhau → dồn kênh (multiplexing), được lưu trữ rồi chuyển tiếp khi đi qua nút trung gian (store & forward).

# Thời gian trong Chuyển gói



# Độ trễ trong mạng chuyển gói



- ❑ Trễ do phải xử lý tại nút:
  - Kiểm tra xem có lỗi bit không?
  - Xác định đường ra
- ❑ Trễ do xếp hàng
  - Đợi tại cổng ra để truyền đi tiếp
  - Phụ thuộc vào mức độ tắc nghẽn của router

# So sánh mạch chuyển gói và mạch chuyển kênh

## ❑ Packet switching

- Không chiếm dụng đường truyền → cho phép nhiều người dùng hơn, hiệu suất sử dụng đường truyền cao.
- Không cần thiết lập kênh truyền (call setup).
- Có độ trễ gói tin.
- Cần phải có cơ chế khắc phục lỗi.

## ❑ Circuit switching

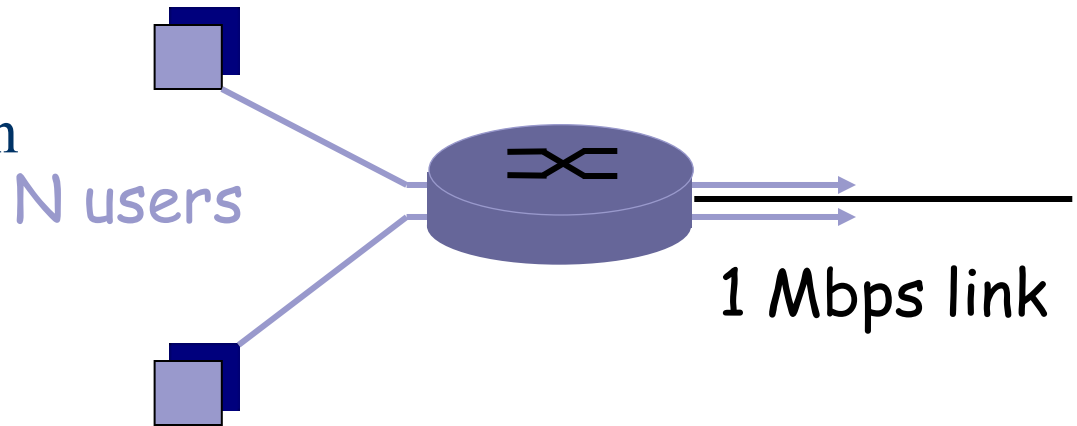
- Call setup
- Thích hợp với truyền tin chất lượng cao, tức thì.

# Mạch chuyển gói và mạch chuyển kênh

Mạch chuyển gói cho phép nhiều người sử dụng mạng hơn!

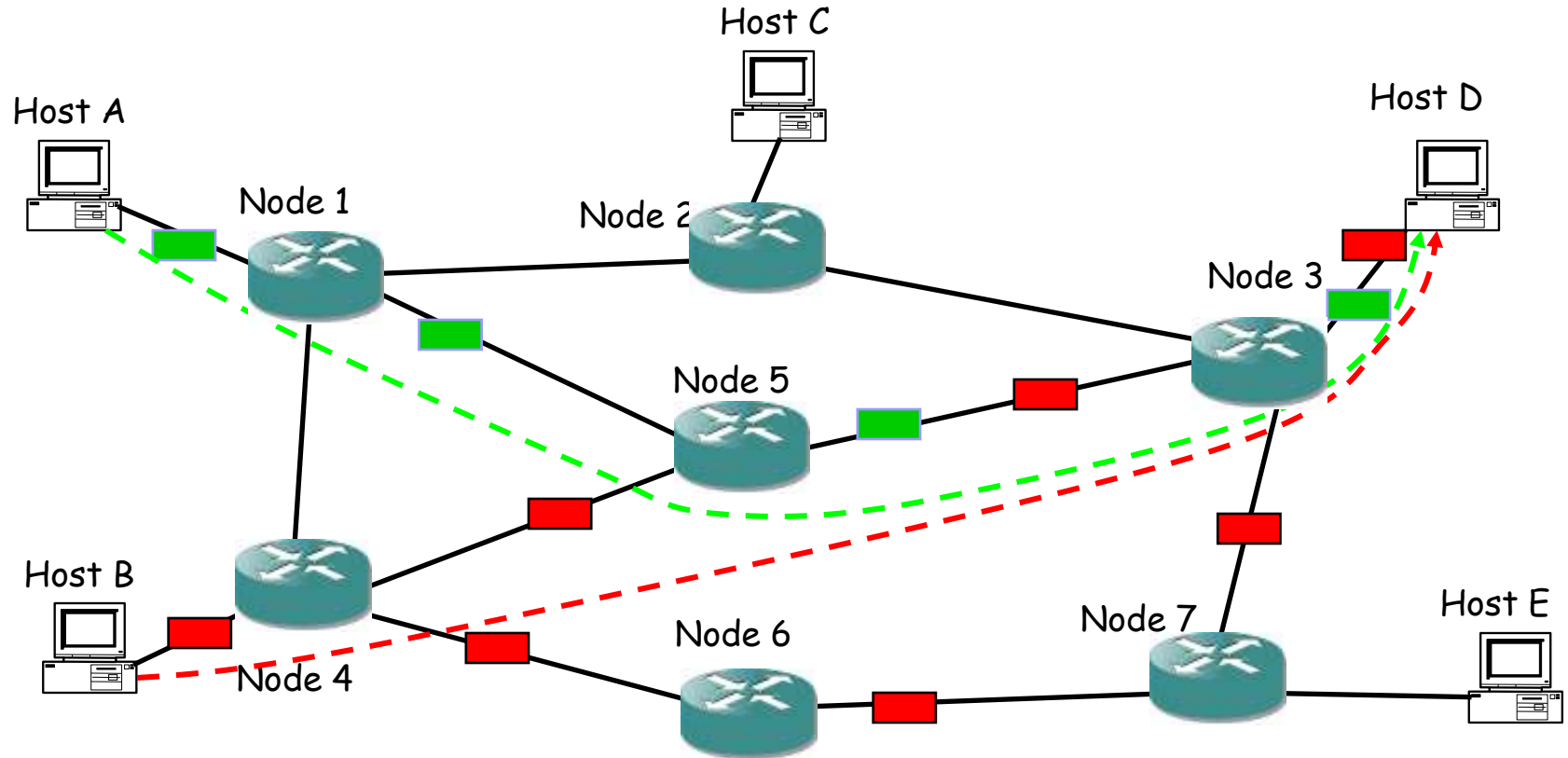
Với kết nối 1 Mbit

- ❑ **Mỗi người dùng cần :**
  - 100Kbps khi “active”
  - Sử dụng 10% thời gian
- ❑ **Mạch chuyển kênh:**
  - Cho phép 10 NSD
- ❑ **Mạch chuyển gói**
  - Cho phép 35 NSD, (khả năng có quá 10 “active” trong một khoảng thời gian nhỏ hơn 0.004)



# Packet switching: Datagram

Vd: Internet



*Datagram packet switching*

# Định tuyến

## ❑ Mạng chuyển mạch gói:

- Sử dụng Địa chỉ đích để xác định chặng kế tiếp
- Có thể đi theo nhiều tuyến đường khác nhau

## ❑ Chuyển mạch ảo:

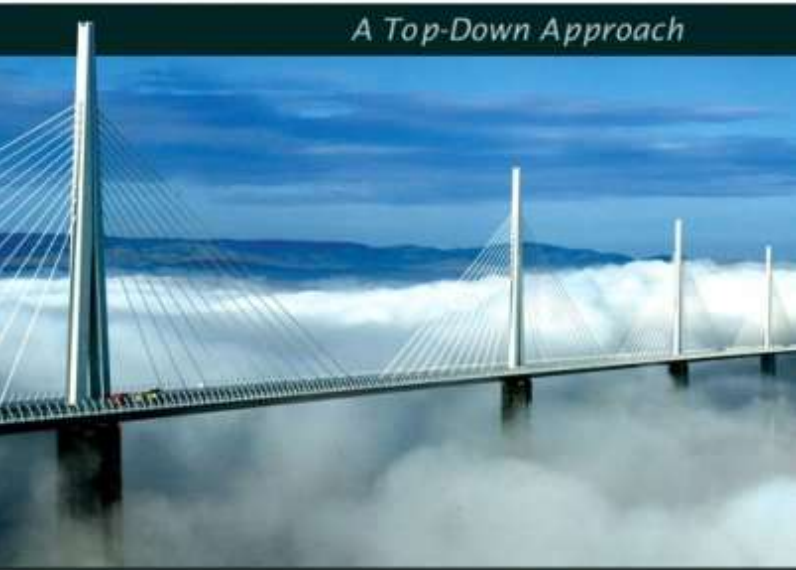
- Mỗi gói tin có một thẻ (Định danh mạch ảo), được sử dụng để xác định chặng kế tiếp
- Tuyến đường cố định được xác định tại thời điểm thiết lập kênh truyền
- Các routers duy trì trạng thái cho mỗi kênh truyền



# Nội dung

## COMPUTER NETWORKING FIFTH EDITION

*A Top-Down Approach*

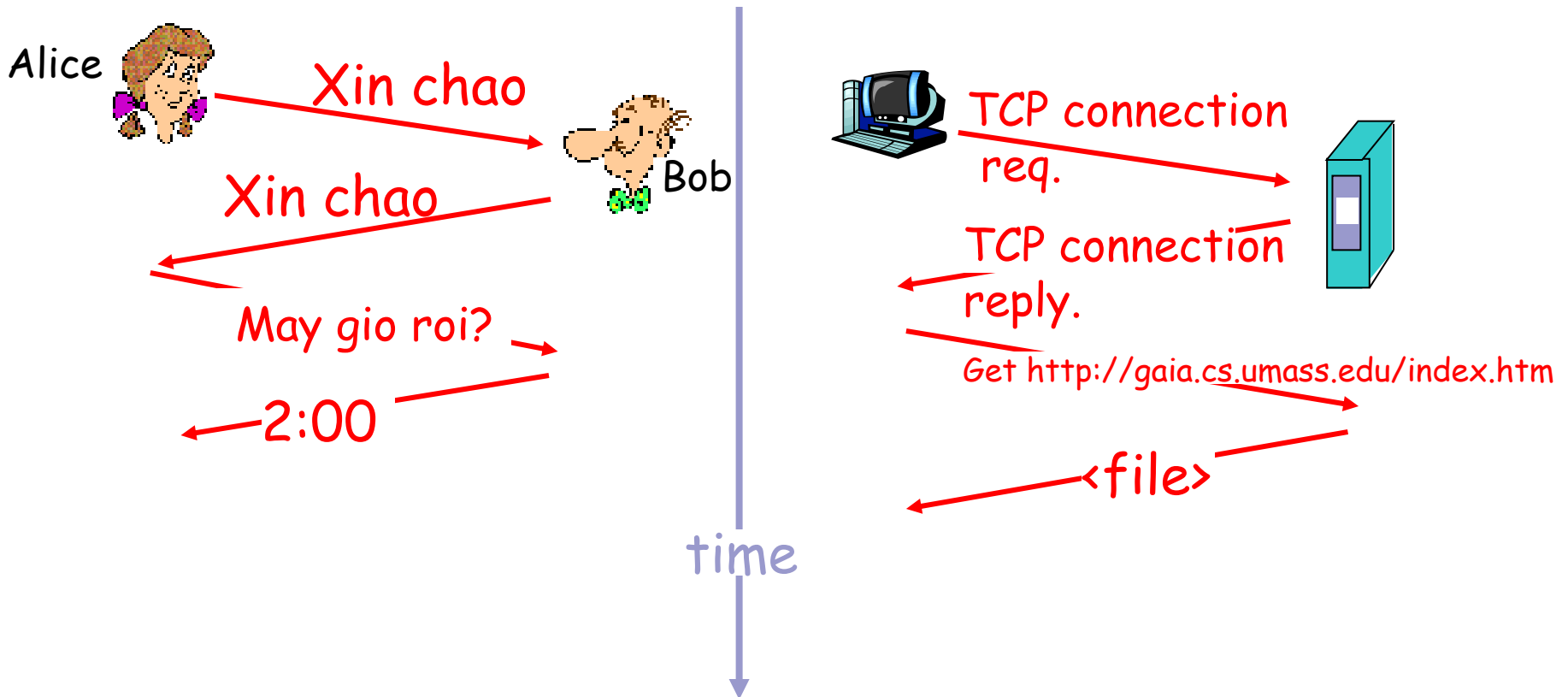


KUROSE • ROSS

### 1.3 - Giao thức mạng (software).

# 1.3. Giao thức mạng (Protocol)

- Là tập các qui tắc để các bên truyền thông có thể giao tiếp với nhau



# Giao thức mạng (Protocol)

## Khái niệm về giao thức

- Các thực thể của mạng muốn trao đổi thông tin với nhau phải bắt tay, đàm phán về một số thủ tục, quy tắc... Cùng phải “nói chung một ngôn ngữ”. Tập quy tắc hội thoại được gọi là giao thức mạng (Protocols).

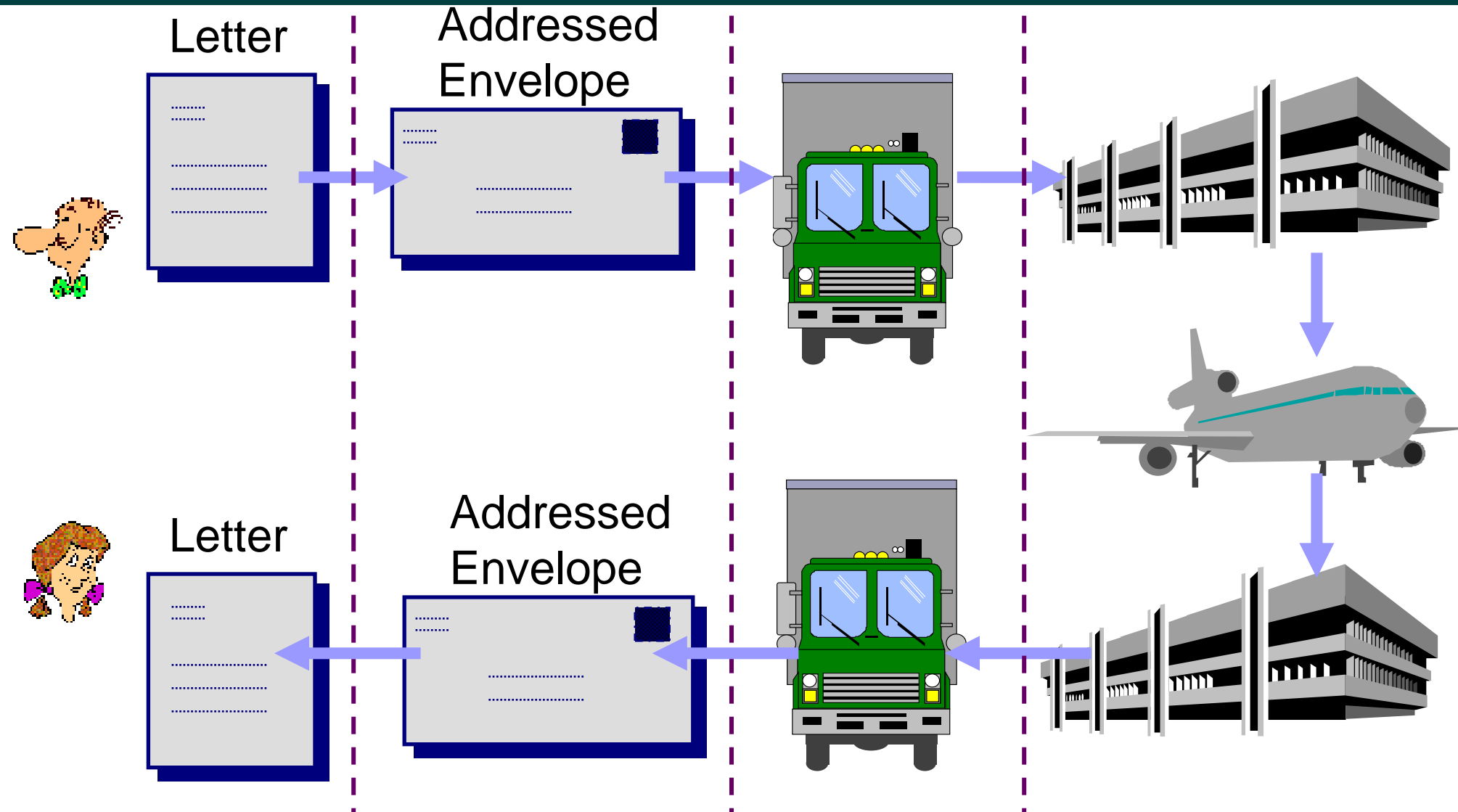
## Các thành phần chính của một giao thức

- ✓ Cú pháp: định dạng dữ liệu, phương thức mã hoá và các mức tín hiệu.
- ✓ Ngữ nghĩa: thông tin điều khiển, điều khiển lưu lượng và xử lý lỗi..

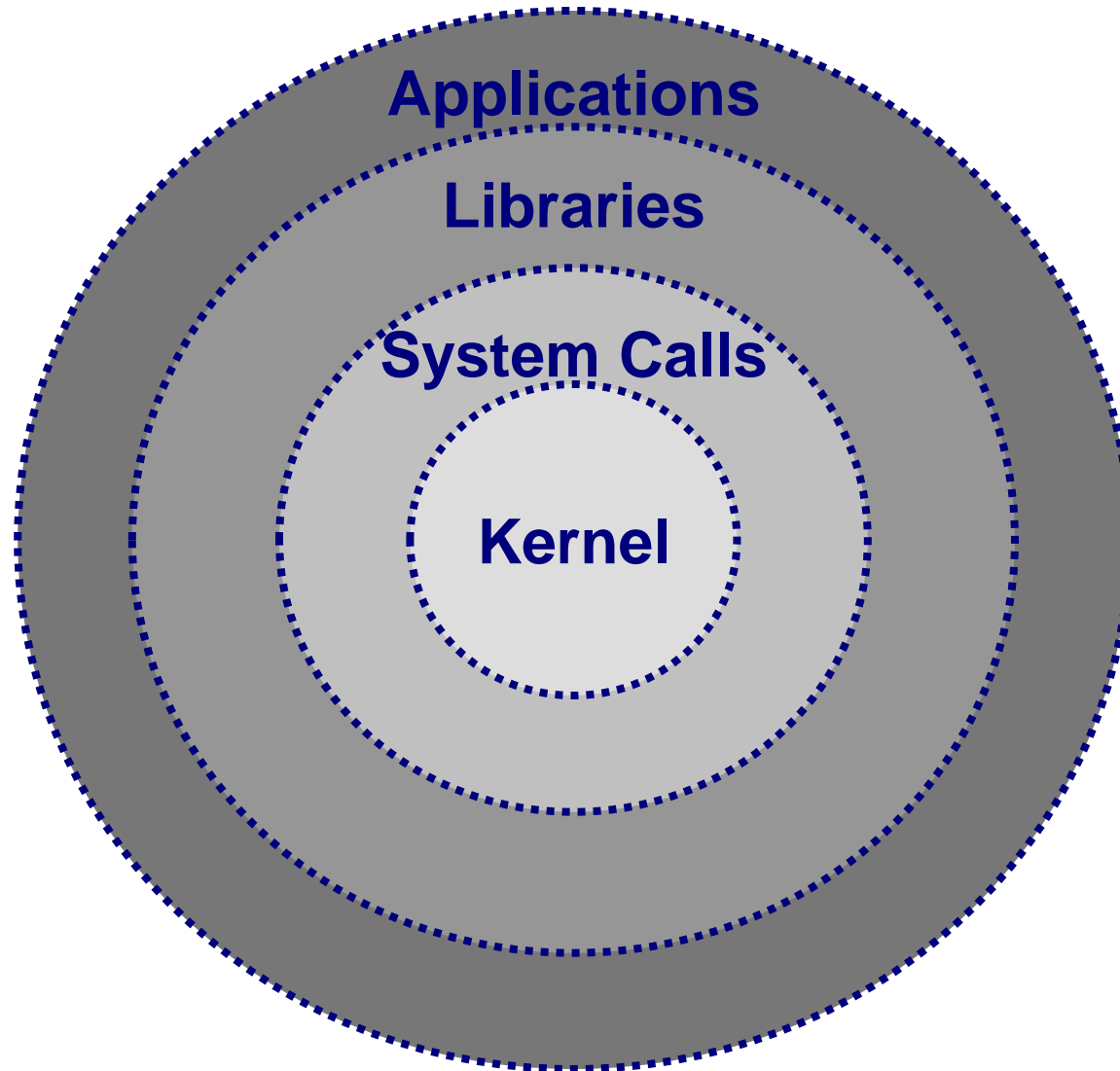
# Một số chức năng của giao thức

- ✓ Đóng gói
- ✓ Phân đoạn và hợp lại
- ✓ Điều khiển liên kết
- ✓ Giám sát
- ✓ Điều khiển lưu lượng
- ✓ Điều khiển lỗi
- ✓ Đồng bộ hoá
- ✓ Địa chỉ hoá

# Hệ thống gửi thư - Nguyên tắc phân tầng



# Unix Operating System Layers

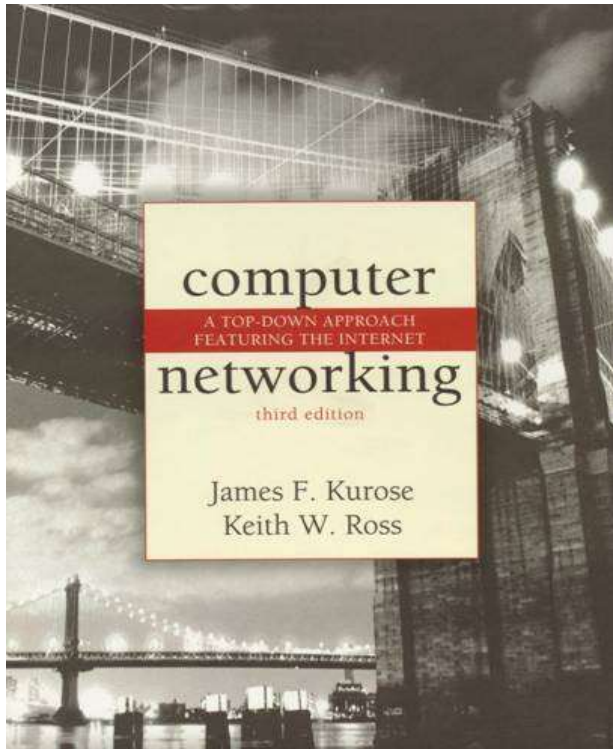


# Giao thức hướng nối và không hướng nối

Giao thức truyền thông được chia làm hai loại:

- ❑ Truyền tin có liên kết (connection-oriented): 3 giai đoạn
  - Thiết lập liên kết (handshaking).
  - Truyền dữ liệu (data transferring).
    - ❖ Có các cơ chế kiểm soát lỗi
  - Huỷ bỏ liên kết (terminating).
- ❑ Truyền thông không liên kết (connectionless): chỉ có giai đoạn truyền dữ liệu.
  - Thường sử dụng cho các tầng thấp, đường truyền có độ tin cậy cao.

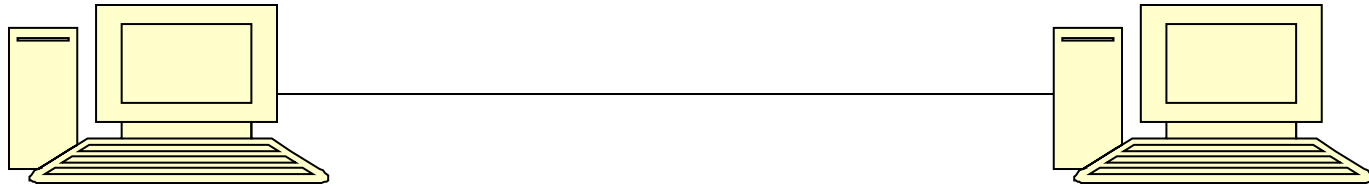




## 1.4 - Các mô hình tham chiếu (reference models).



# Truyền thông host-to-host



## ❑ Mô hình độc quyền

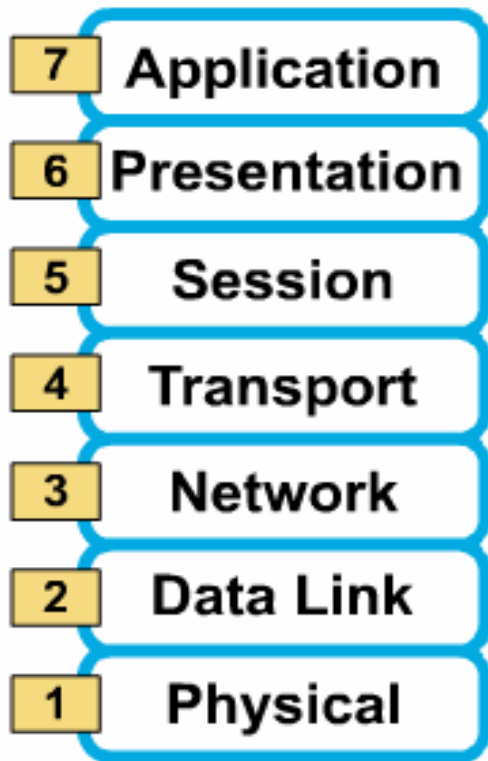
- Phần cứng
- Phần mềm

## ❑ Mô hình dựa trên sự chuẩn hóa

- Máy tính có thể cài nhiều phần của các hãng khác nhau
  - Các máy tính khác nhau có thể kết nối được với nhau
- => cần chuẩn hóa dựa trên cơ sở phân lớp

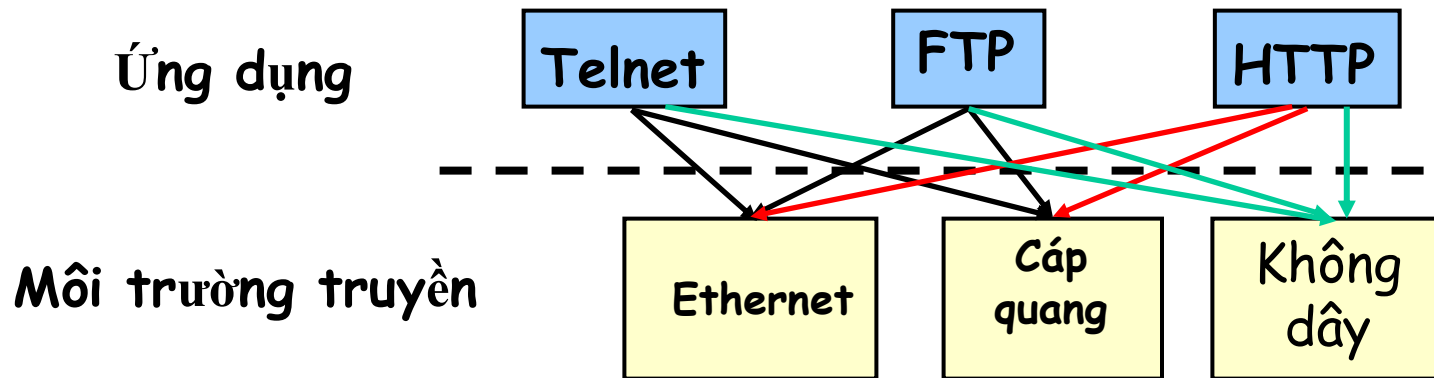
# Tại sao phải phân tầng?

## Why a Layered Network Model?

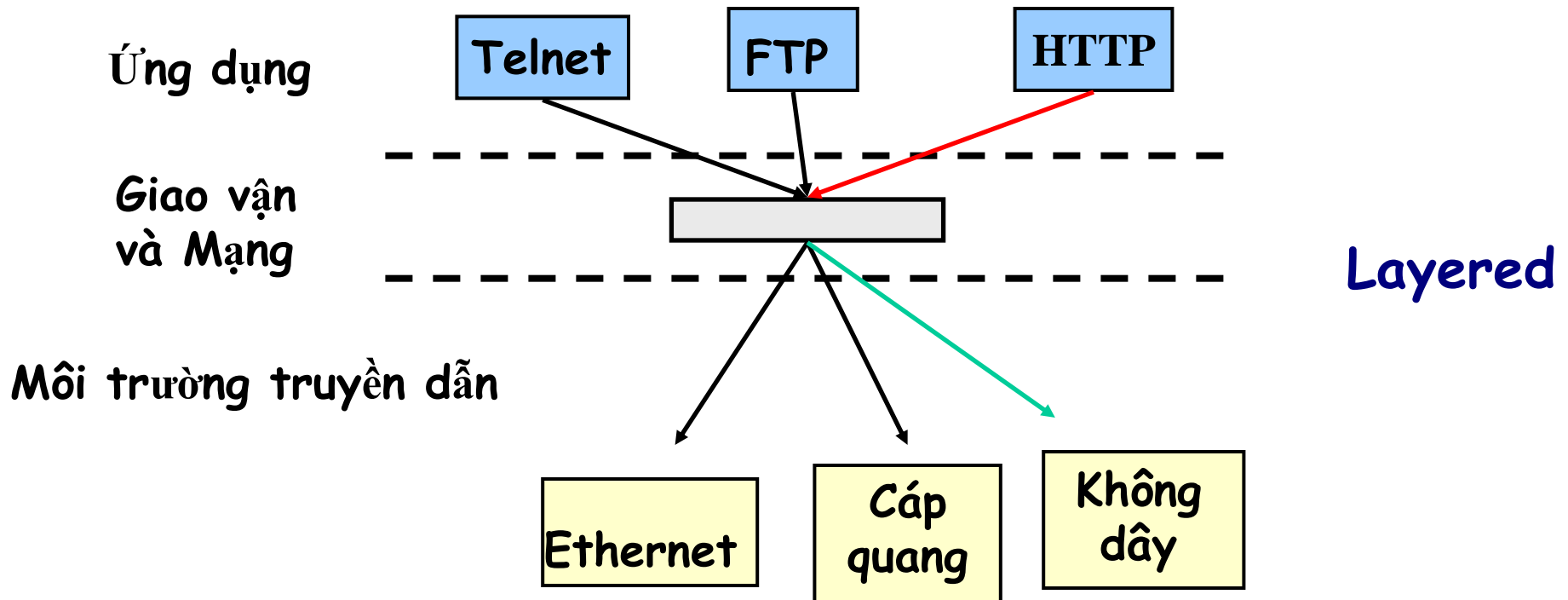


- ◆ Reduces complexity
- ◆ Standardizes interfaces
- ◆ Facilitates modular engineering
- ◆ Ensures interoperable technology
- ◆ Accelerates evolution
- ◆ Simplifies teaching and learning

# Mô hình không phân tầng



# Mô hình phân tầng



## 1.4. Các mô hình tham chiếu

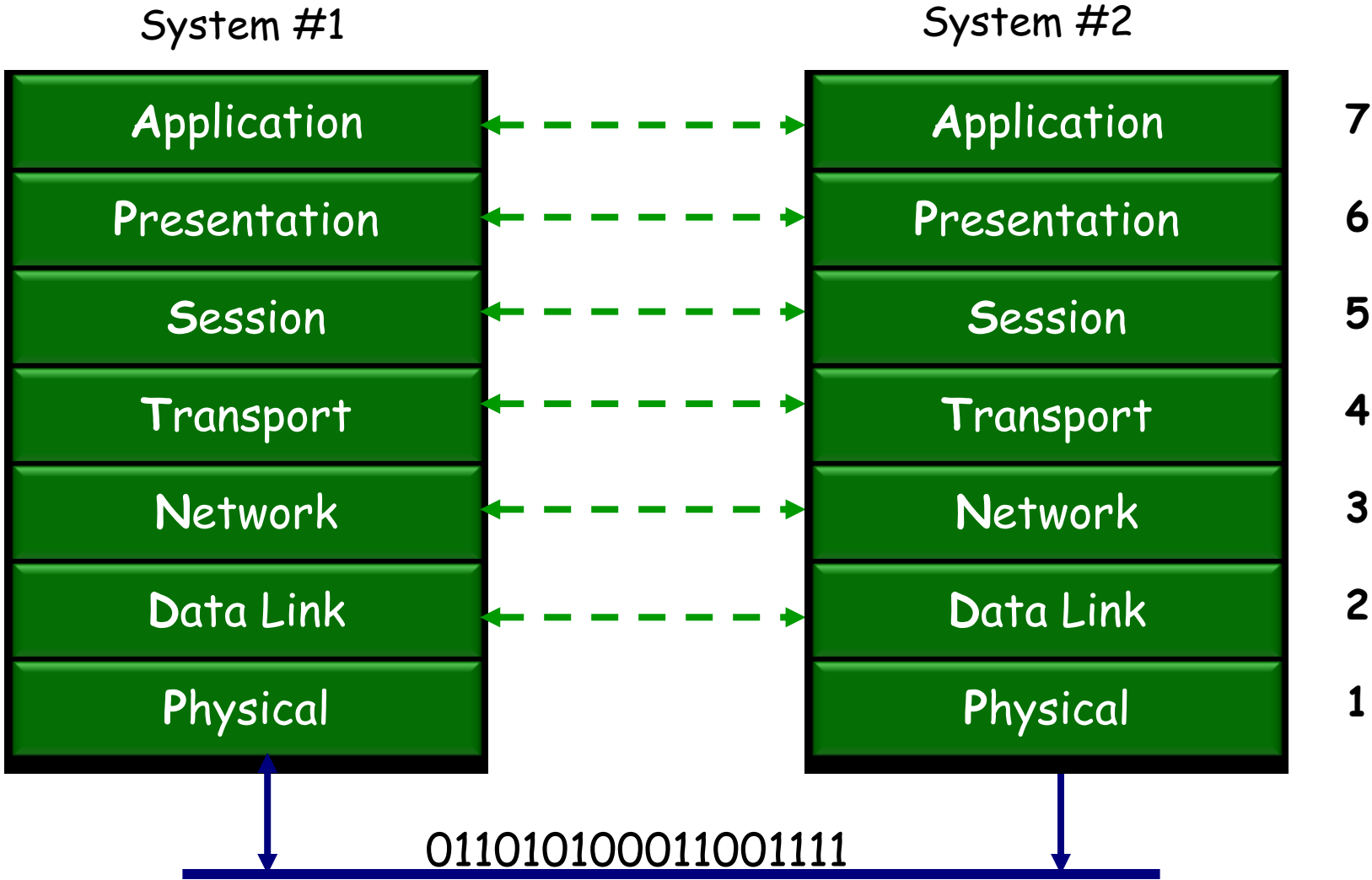
### ❑ Open System Interconnection Reference Model (OSI Reference Model)

- Đưa ra bởi ISO (International Organization for Standardization) năm 1984.
- Mô hình tham chiếu lý thuyết cho các hệ thống mở nói chung.
- 7 tầng: Physical, Data Link, Network, Transport, Session, Presentation, Application.

### ❑ TCP/IP Reference Model

- Sử dụng cho mạng Internet.
- 4 tầng: Host-to-network, Internet, Transport, Application.

# OSI layers



# Layer 1: The Physical Layer

- ❑ Chỉ có tầng vật lý của hai hệ thống được *kết nối* và truyền thông *trực tiếp* với nhau (wire/wireless).
- ❑ Các đặc tả vật lý (điện, điện từ...) nhằm đảm bảo sự kết nối và truyền *tín hiệu* giữa hai hệ thống.
- ❑ Một số yếu tố:
  - Cáp truyền (Cable).
  - Mức điện thế (voltage levels).
  - Phương pháp điều chế tín hiệu.
  - Chu kỳ tín hiệu, khoảng cách...

# Layer 2: The Data Link Layer

- ❑ Biến dữ liệu thô nhận được từ tầng vật lý thành dữ liệu có cấu trúc logic cụ thể hơn.
  - Framing.
    - ❖ 001101010 → Khung (frame) có cấu trúc.
  - Physical Addressing.
    - ❖ Dữ liệu đến từ đâu? Máy tính nào gửi đến?
    - ❖ Dữ liệu cần phải gửi tiếp đi đâu?
- ❑ Đảm bảo sự *tin cậy* của tín hiệu truyền giữa hai tầng vật lý.
  - Kiểm soát lỗi (error control).
  - Kiểm soát luồng (flow control).



# Layer 3: The Network Layer

- ❑ Định tuyến gói dữ liệu
- ❑ Cung cấp địa chỉ logic và chọn đường đi
- ❑ Kết nối các mạng có kiến trúc khác nhau.

# Layer 4: The Transport Layer

- ❑ Tầng trên cùng của quá trình truyền dữ liệu.
- ❑ Đảm bảo dữ liệu được truyền thông suốt và tin cậy giữa hai hệ thống (2 end-systems).
  - Thiết lập, quản lý các kết nối đầu cuối - đầu cuối (virtual circuits).
  - Kiểm soát lỗi, kiểm soát luồng (error detection and recovery, information flow control).

# Layer 5: The Session Layer

- ❑ Thiết lập và quản lý các phiên truyền thông giữa hai hệ thống.
  - Chứng thực (security authentication).
  - Thiết lập liên kết (connection establishment).
  - Huỷ bỏ liên kết (connection release).

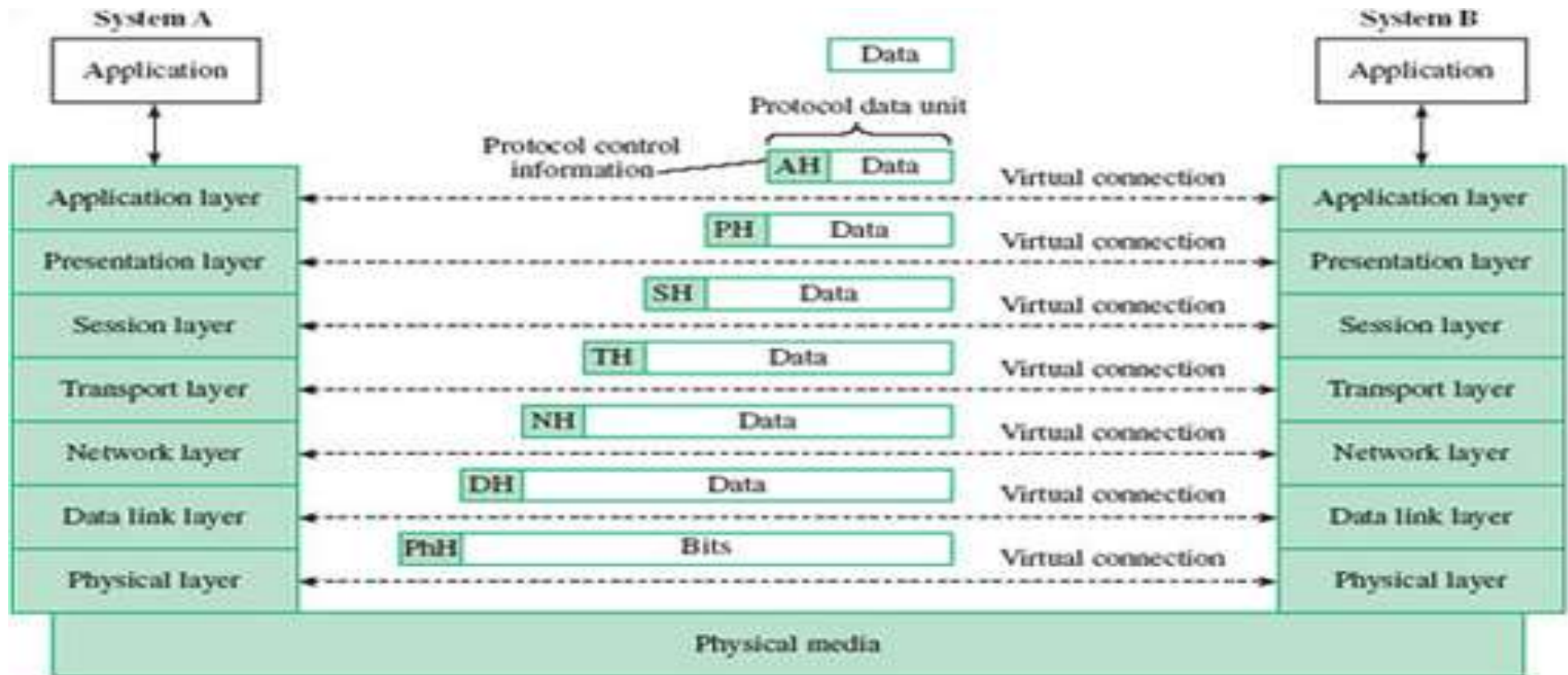
# Layer 6: The Presentation Layer

- Đảm bảo thông tin truyền từ ứng dụng của hệ thống truyền có thể đọc được bởi ứng dụng của hệ thống nhận.
  - Cú pháp và ngữ nghĩa của dữ liệu (syntax & semantic).
  - Định dạng dữ liệu (data formatting).
  - Chuyển đổi định dạng (format exchange).
  - Nén dữ liệu (data compression).
  - Mã hóa dữ liệu (data encryption)

# Layer 7: The Application Layer

- ❑ Cung cấp các dịch vụ để người dùng hay các ứng dụng như ftp, telnet, electronic mail,...
- ❑ Cơ chế xác thực người dùng

# Quá trình đóng gói dữ liệu

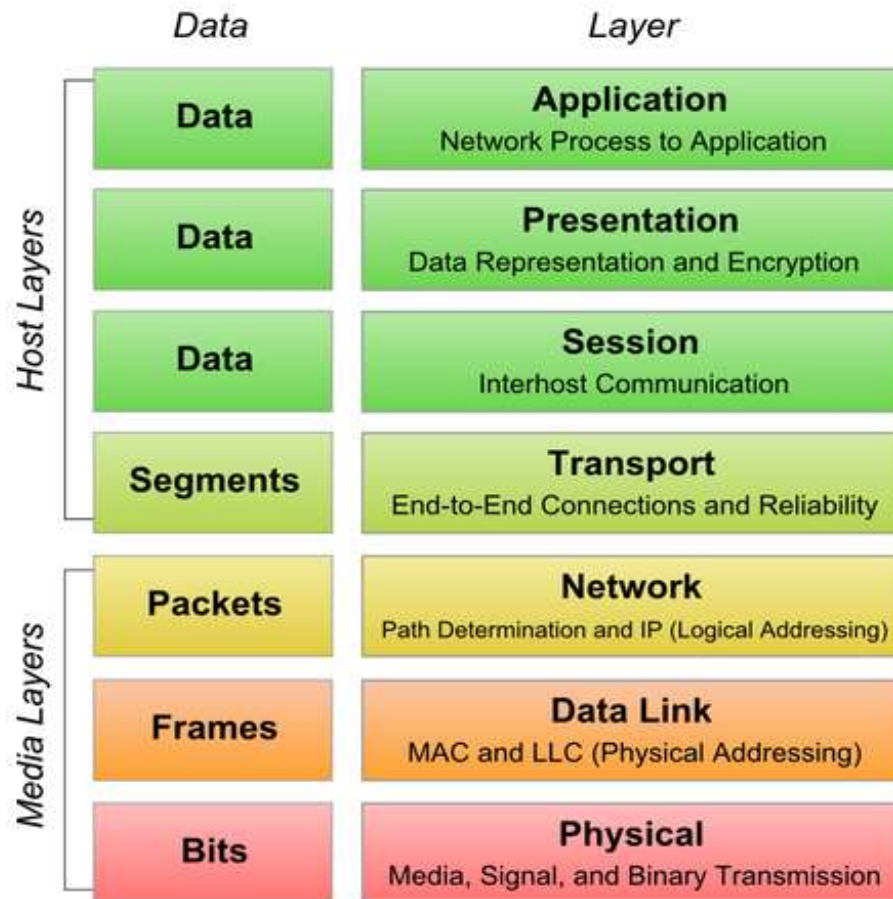


## Glossary

<b>AH</b> Application layer header	<b>TH</b> Transport layer header	<b>PhH</b> Physical layer header
<b>PH</b> Presentation layer header	<b>NH</b> Network layer header	
<b>SH</b> Session layer header	<b>DH</b> Data link layer header	

# Đơn vị dữ liệu của các tầng

## OSI Model

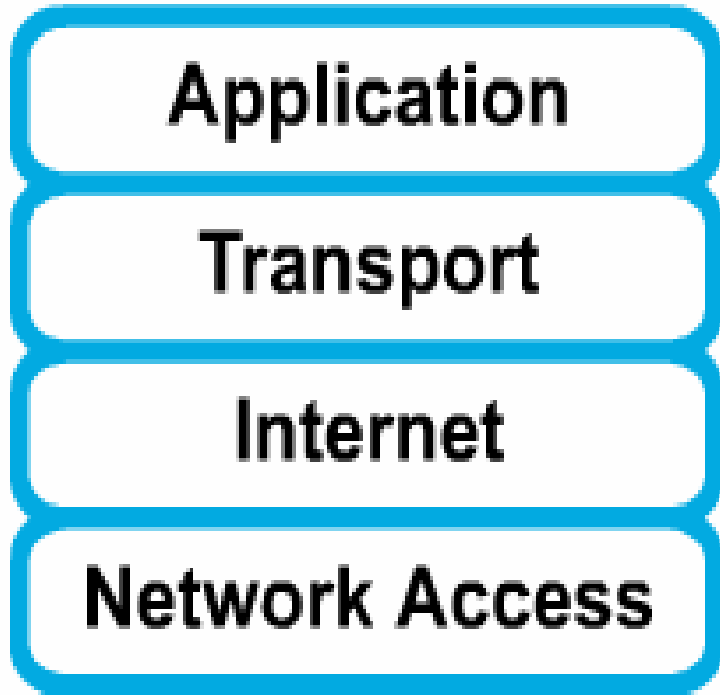


# TCP/IP Reference Model

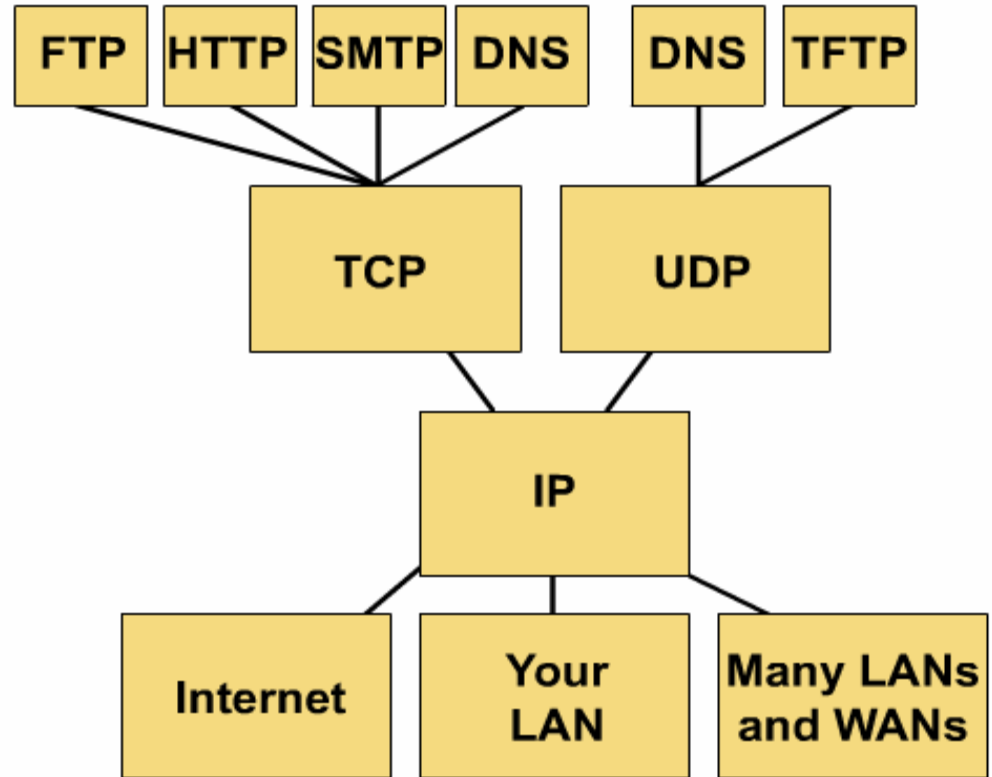
- ❑ Mô hình OSI chỉ mang tính chất lý thuyết, phục vụ nghiên cứu và học tập.
- ❑ TCP/IP là mô hình áp dụng cho mạng Internet.
  - TCP = Transmission Control Protocol.
  - IP = Internet Protocol.
  - TCP, IP là hai giao thức phổ biến trong họ giao thức TCP/IP.



# TCP/IP Layers & Protocols



Layers

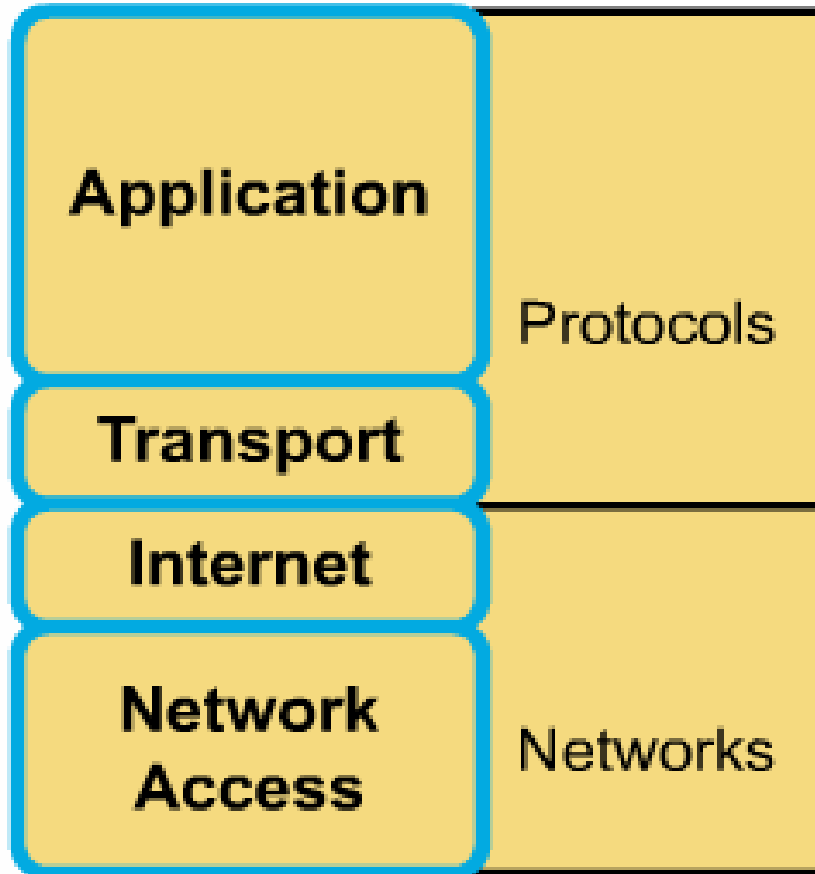


Protocols

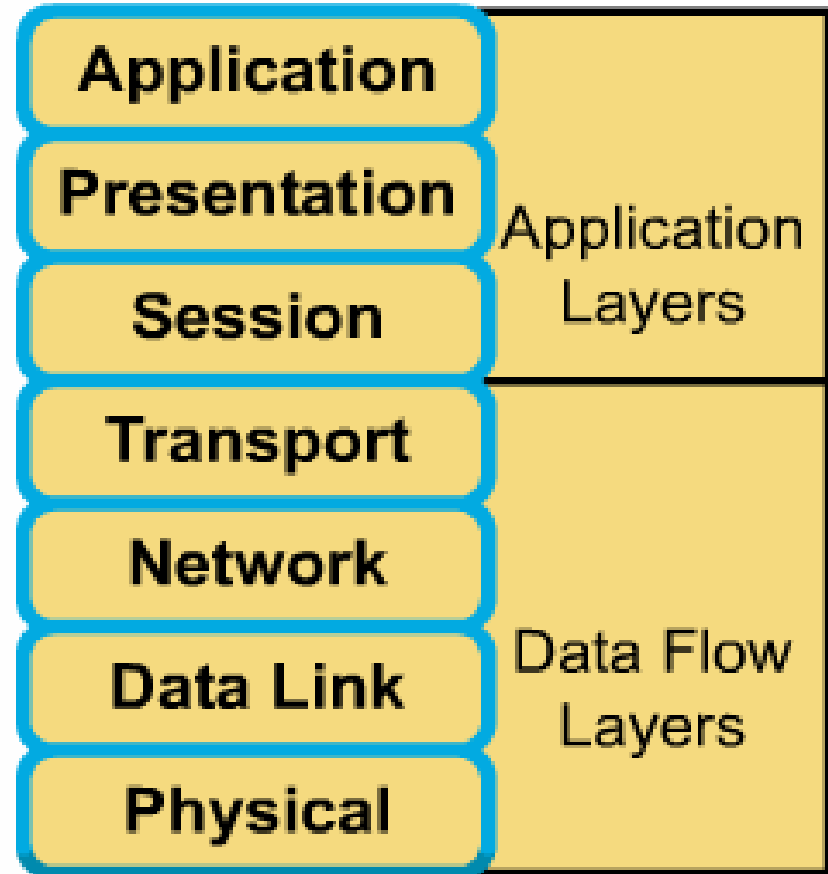
*Network Access = Host-to-network = Data link + Physical  
Network = Internet*

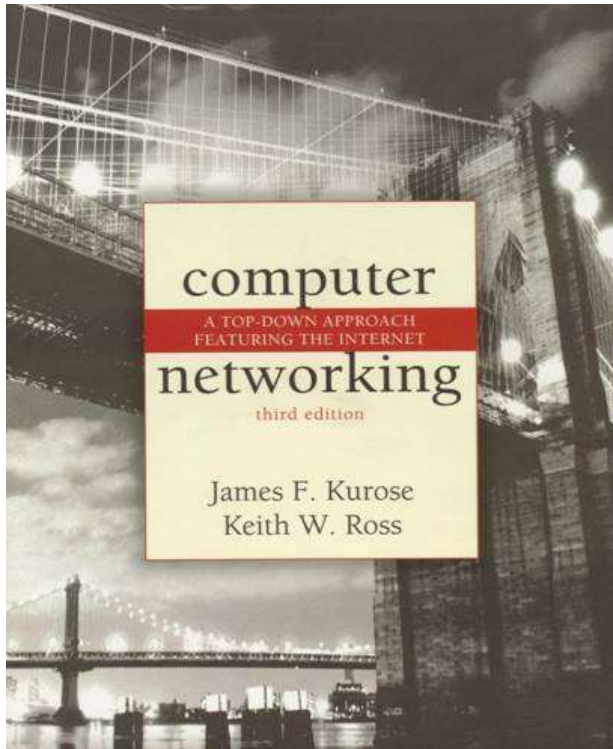
# Đổi sánh OSI và TCP/IP

TCP/IP Model



OSI Model





## 1.5 - Chuẩn mạng máy tính (network standards).

# 1.5. Network Standardization

## ❑ OSI:

- Chỉ nêu lên chức năng của từng tầng.
- Không chỉ ra cài đặt các tầng này như thế nào.
- Để hai hệ thống giao tiếp được với nhau thì chúng cần:
  - ❖ Được cài đặt các chức năng truyền thông chung.
  - ❖ Tổ chức thành cùng một tập tầng.
  - ❖ Hai tầng đồng mức phải có chung giao thức.

## ❑ Từ đó, cần phải chuẩn hoá (standardization)

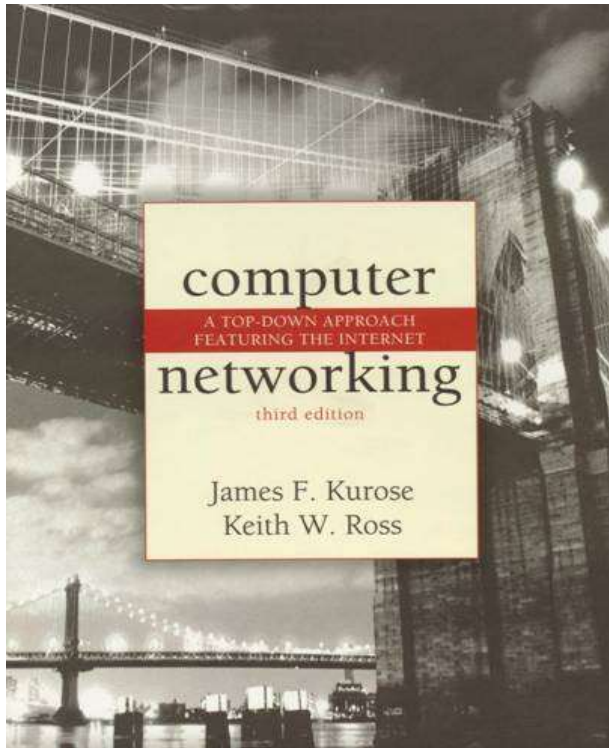
# Các tổ chức và chuẩn thông dụng

- ❑ ISO (International Organization for Standardization)
  - <http://www.iso.org>
- ❑ CCITT (Comité Consultatif International pour Télégraphe et Téléphone)
  - Không ban hành các chuẩn mà ban hành các khuyến nghị (X.25...).
- ❑ ANSI (American National Standard Institute)
- ❑ NIST (National Institute of Standards and Technology)
- ❑ IEEE (Institute of Electrical and Electronics Engineers)
- ❑ IAB (Internet Architecture Board).
- ❑ RFCs (Request For Comments).
- ❑ IRTF (Internet Research Task Force).
- ❑ IETF (Internet Engineering Task Force).
- ❑ ISOC (Internet Society)...

# Nhóm chuẩn IEEE 802 (LAN)

Number	Topic
802.1	Overview and architecture of LANs
802.2 ↓	Logical link control
802.3 *	Ethernet
802.4 ↓	Token bus (was briefly used in manufacturing plants)
802.5	Token ring (IBM's entry into the LAN world)
802.6 ↓	Dual queue dual bus (early metropolitan area network)
802.7 ↓	Technical advisory group on broadband technologies
802.8 †	Technical advisory group on fiber optic technologies
802.9 ↓	Isochronous LANs (for real-time applications)
802.10 ↓	Virtual LANs and security
802.11 *	Wireless LANs
802.12 ↓	Demand priority (Hewlett-Packard's AnyLAN)
802.13	Unlucky number. Nobody wanted it
802.14 ↓	Cable modems (defunct: an industry consortium got there first)
802.15 *	Personal area networks (Bluetooth)
802.16 *	Broadband wireless
802.17	Resilient packet ring

# Nội dung

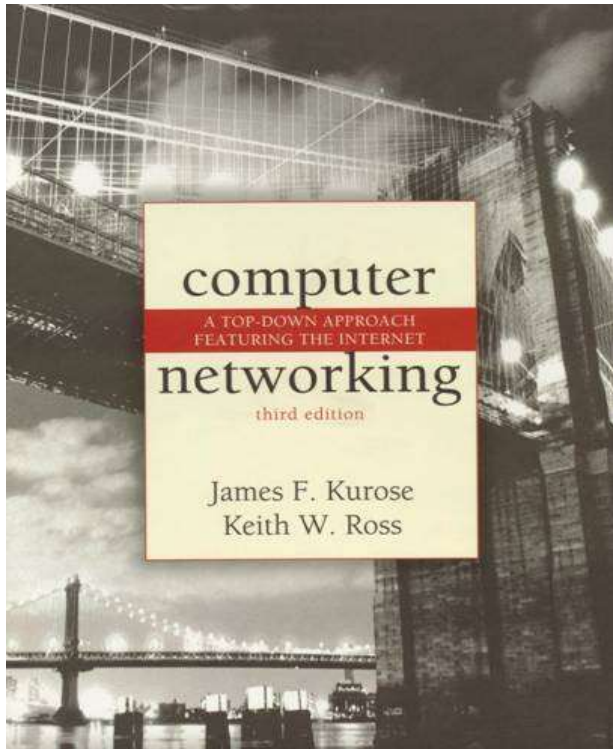


## 1.6 - Hệ điều hành trong môi trường mạng.

# 1.6. Hệ điều hành trong môi trường mạng

- ❑ Không hỗ trợ mạng
  - DOS
- ❑ Trạm làm việc
  - Windows 9x/NT Workstation/2000/XP/Vista/Server
  - MacOS
  - Linux...
- ❑ Máy chủ quản lý mạng, cung cấp dịch vụ mạng.
  - Windows NT/2000/2003/2008 Server
  - Novel NetWare
  - Unix, Linux, Sun Solaris...

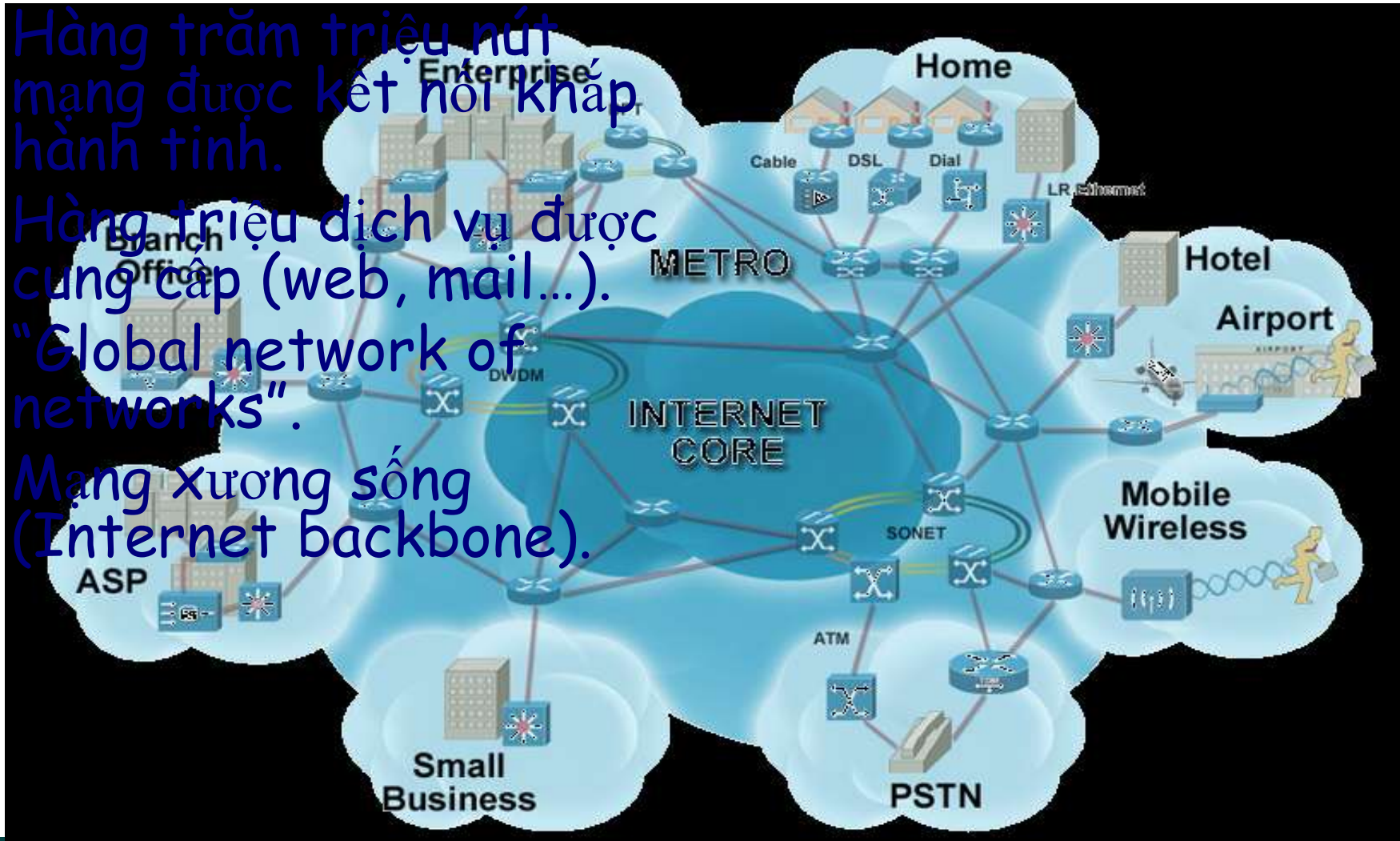




## 1.7 - Mạng Internet.

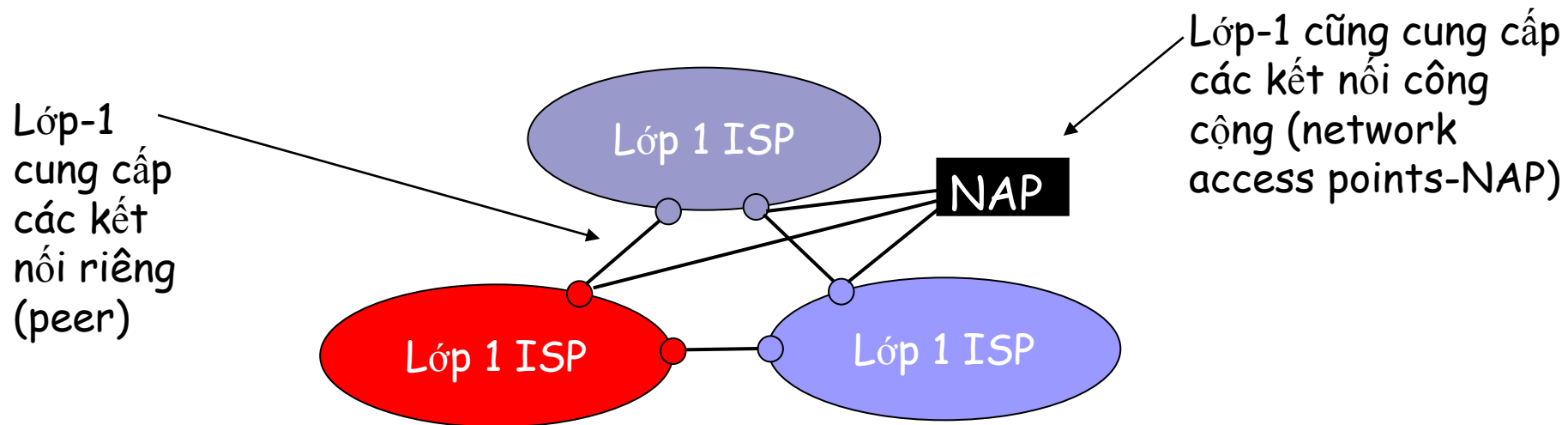
# 1.7. Mạng Internet

- Hàng trăm triệu nút mạng được kết nối khắp hành tinh.
- Hàng triệu dịch vụ được cung cấp (web, mail...).
- "Global network of networks".
- Mạng xương sống (Internet backbone).



# Kiến trúc Internet: mạng của các mạng

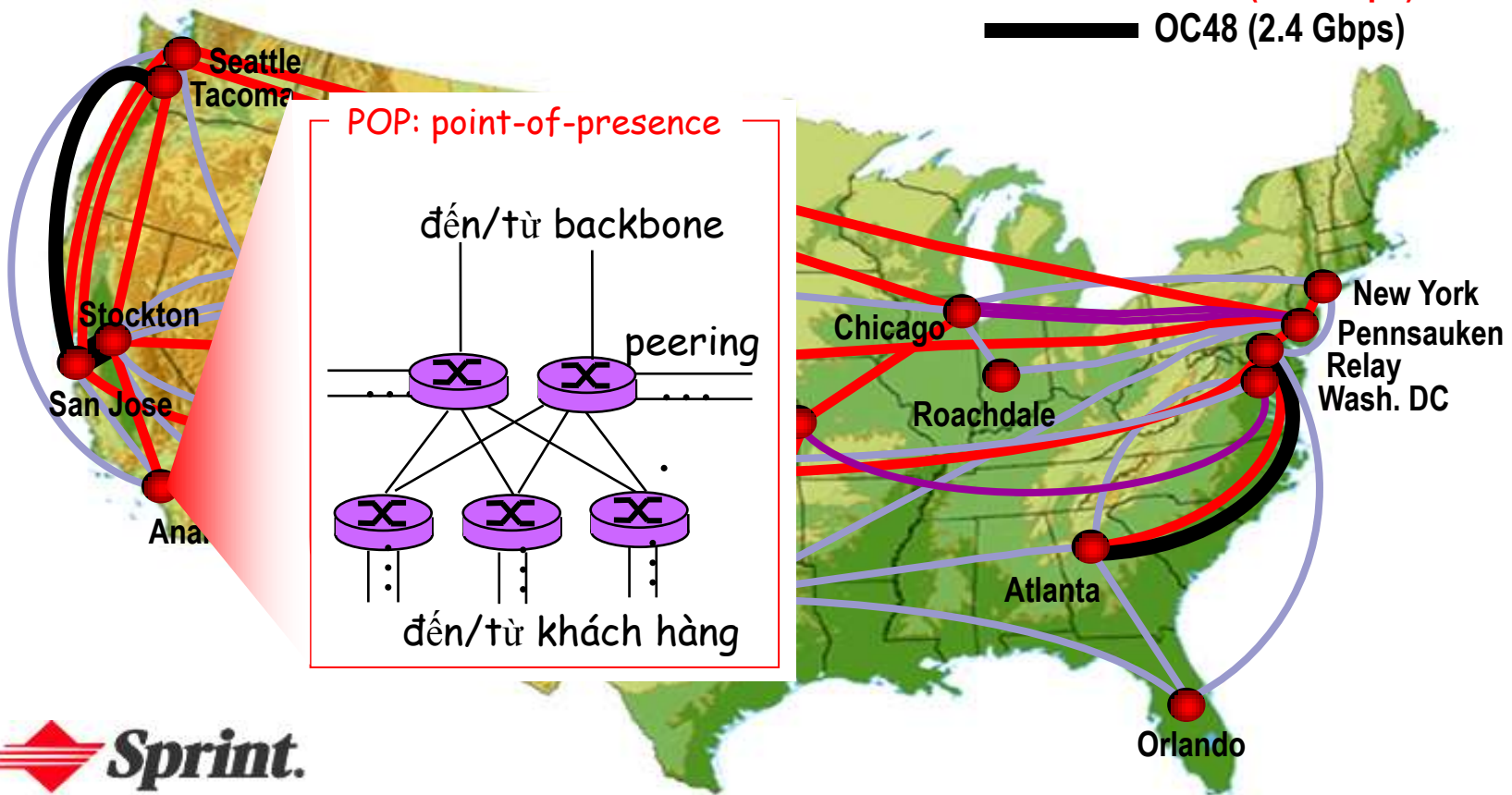
- ❑ roughly hierarchical - không có thứ bậc
- ❑ **Trung tâm: "lớp-1" là các ISP** (ví dụ: MCI, Sprint, AT&T, Cable and Wireless), bao trùm các quốc gia/toàn thế giới
  - **Đối xử như nhau**



# Lớp-1 ISP: ví dụ Sprint

## Mạng Sprint US backbone

- DS3 (45 Mbps)
- OC3 (155 Mbps)
- OC12 (622 Mbps)
- OC48 (2.4 Gbps)

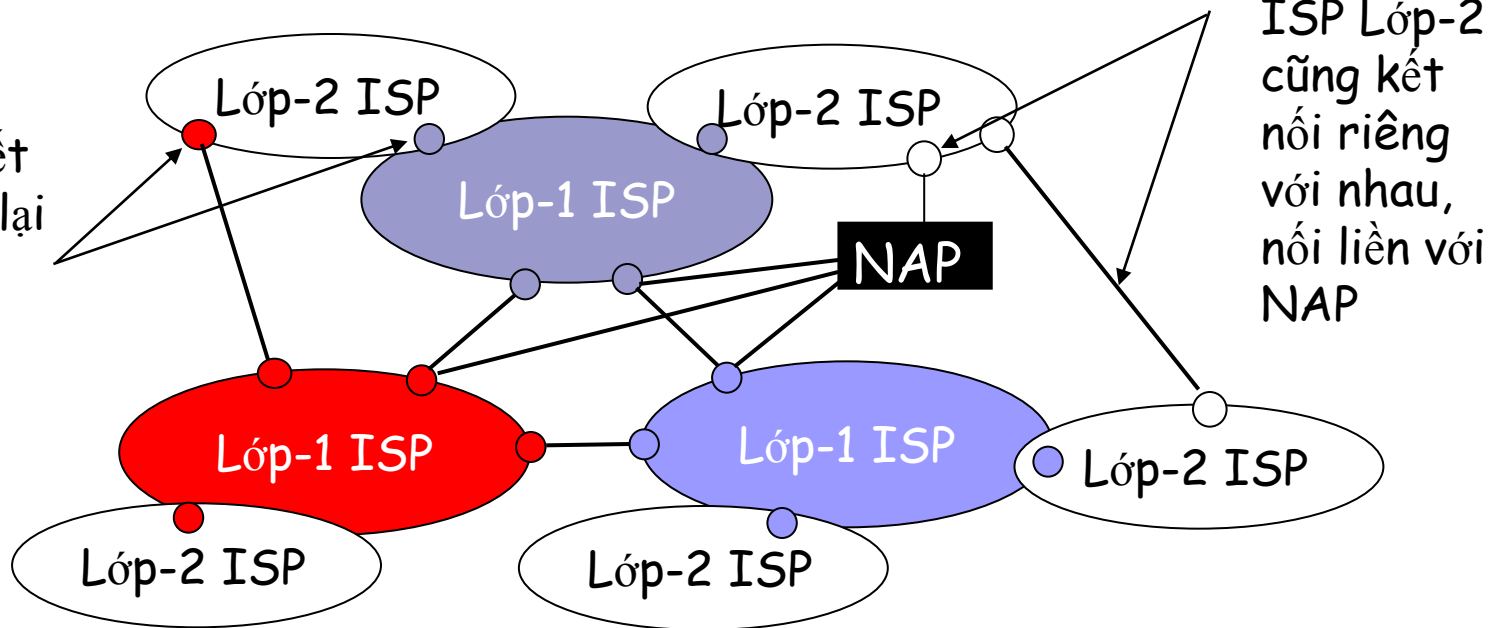


# Kiến trúc Internet: mạng của các mạng

## ❑ “Lớp-2” các ISP nhỏ hơn (thường là ISP vùng)

- Nối kết một hoặc nhiều ISPs lớp-1, cũng có thể một số ISP lớp-2

ISP Lớp-2 giúp  
ISP lớp-1 ISP kết  
nối với phần còn lại  
của Internet  
❑ Lớp-2 ISP là  
khách hàng của  
ISP lớp-1

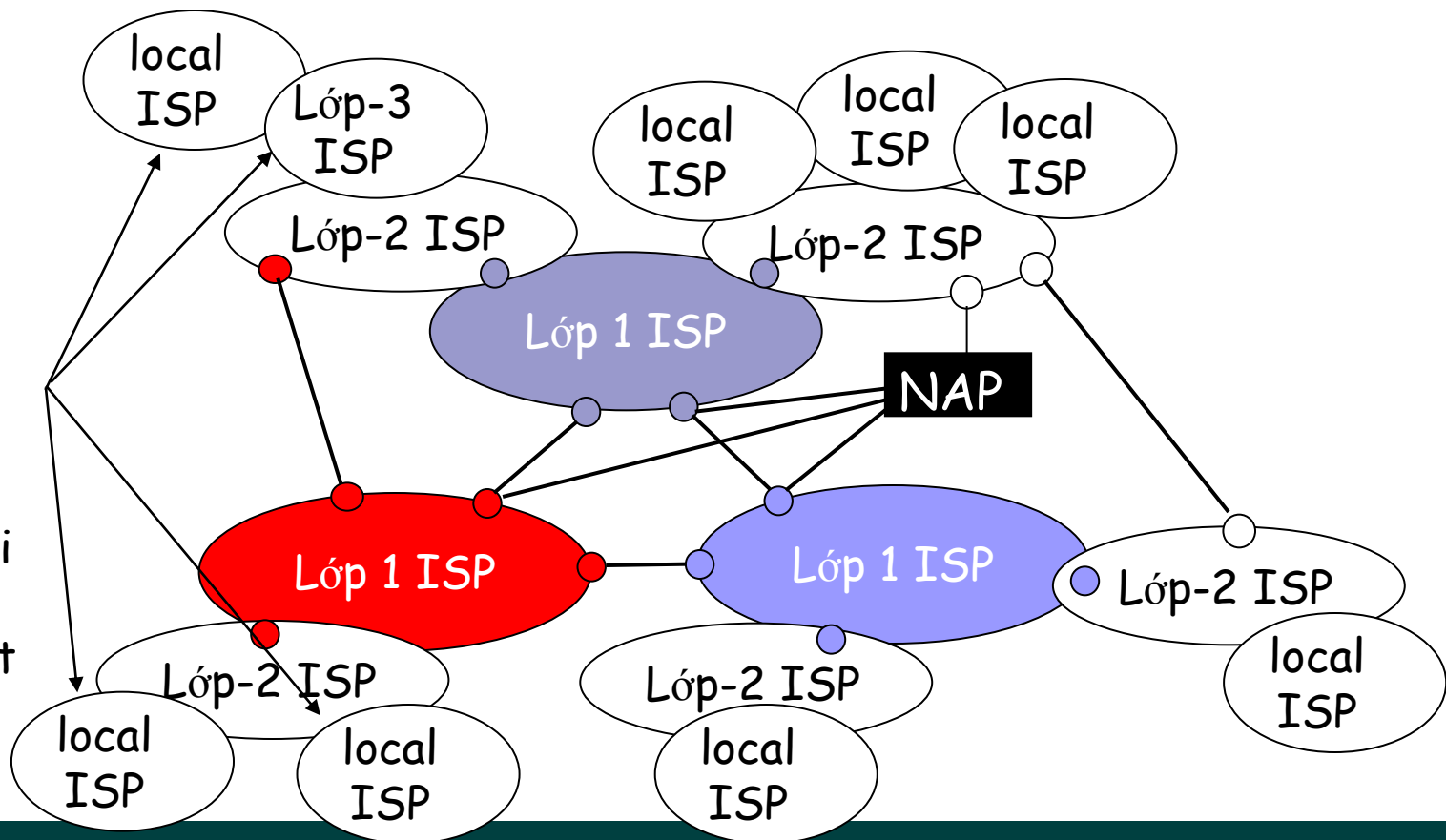


# Kiến trúc Internet: mạng của các mạng

## ISP "Lớp-3" và ISP địa phương (local ISP)

- hop cuối cùng "truy xuất" mạng (gần các hệ thống đầu cuối nhất)

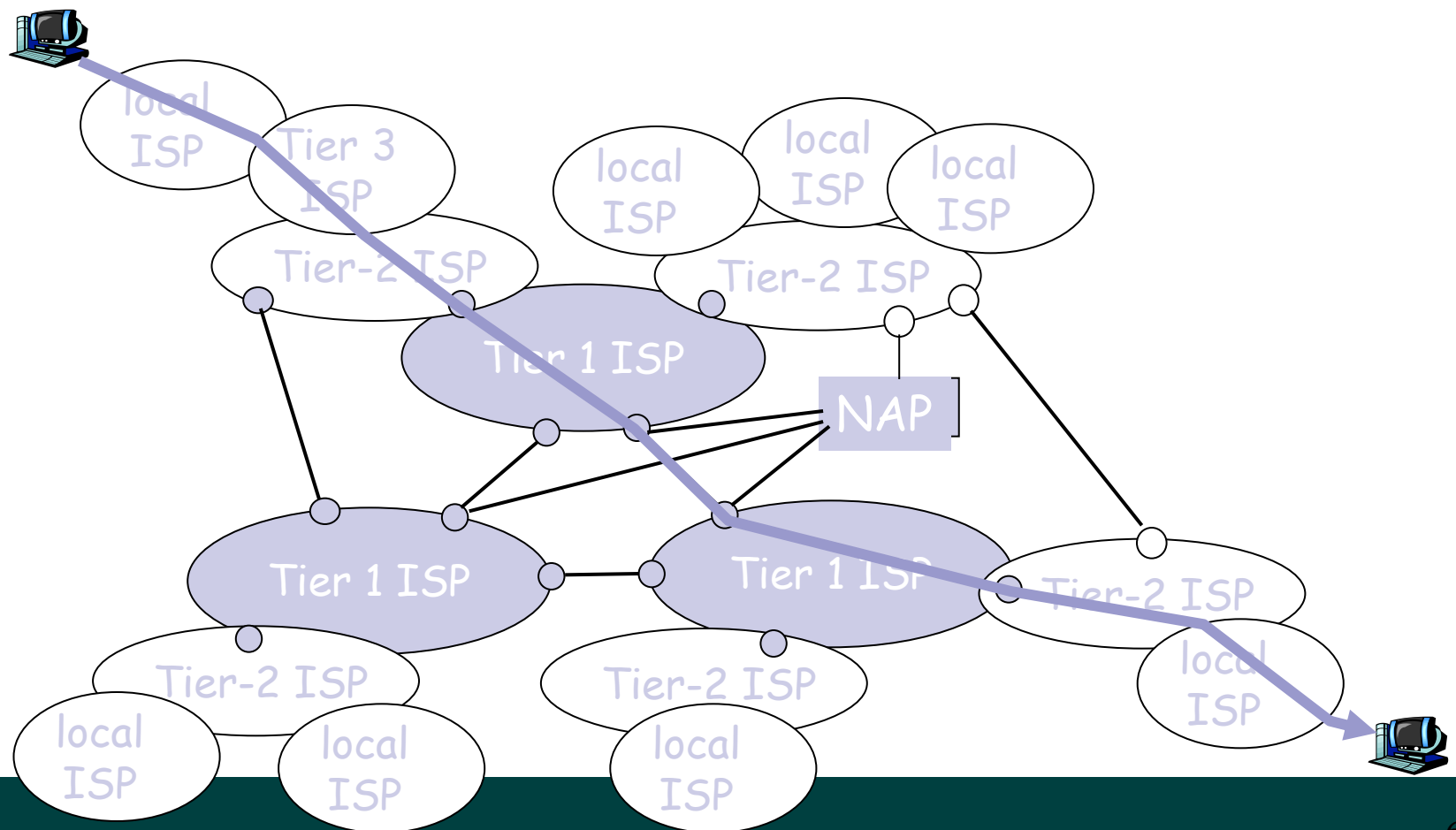
ISP địa phương và lớp-3 là khách hàng của các lớp cao hơn, kết nối chúng với phần còn lại của Internet



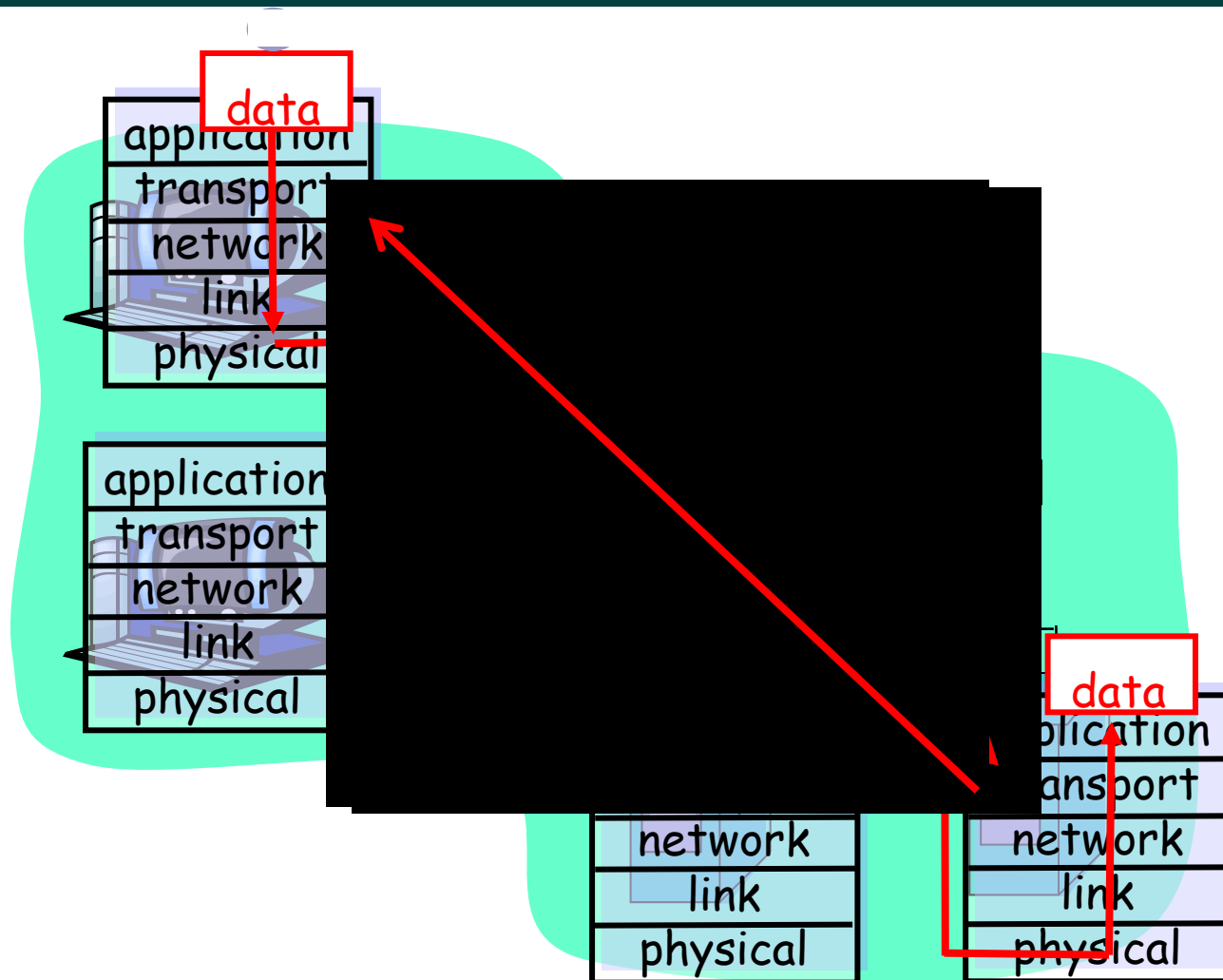


# Kiến trúc Internet: mạng của các mạng

- ❑ Một gói phải đi qua nhiều mạng!

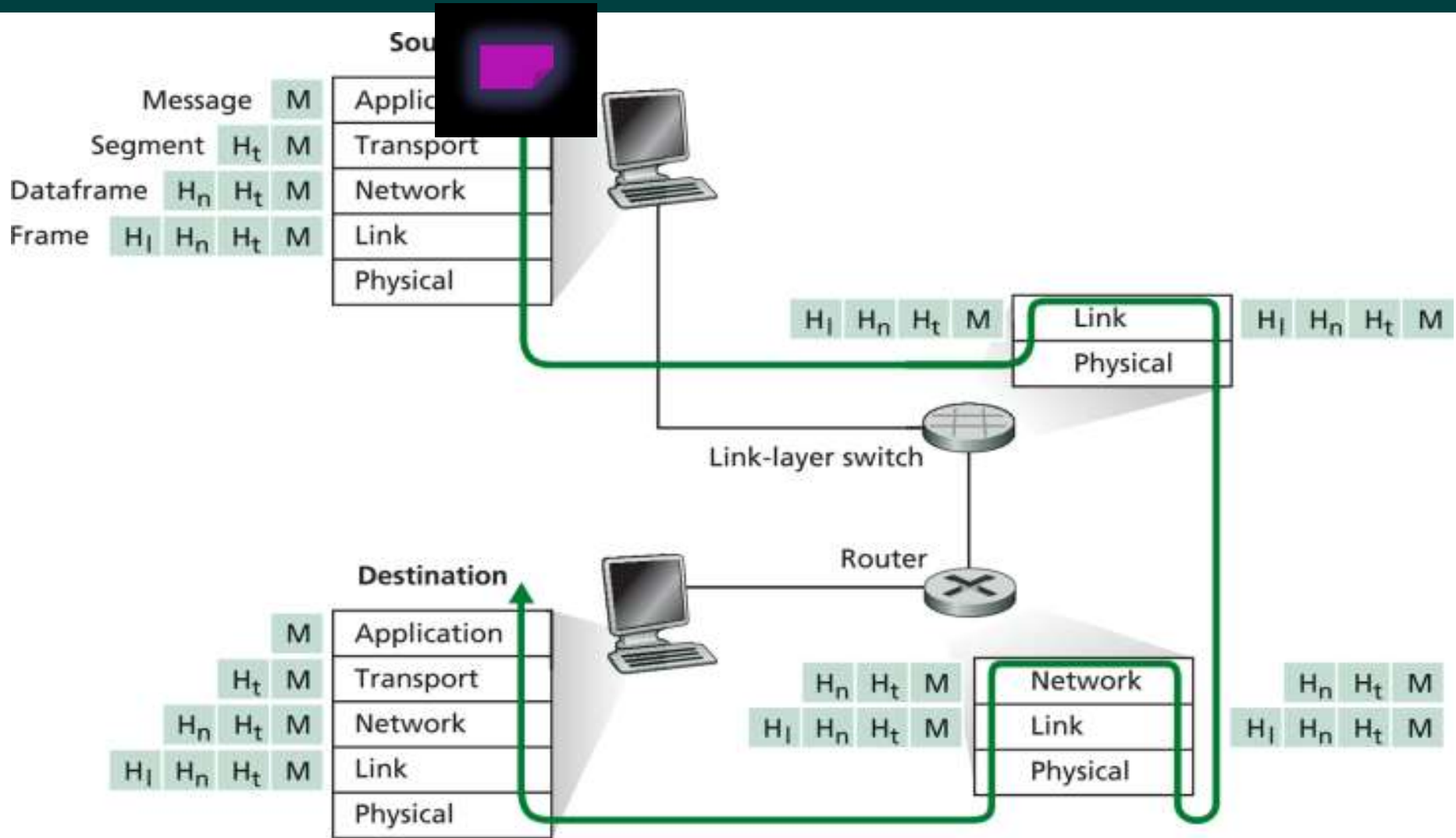


# Kênh truyền logic

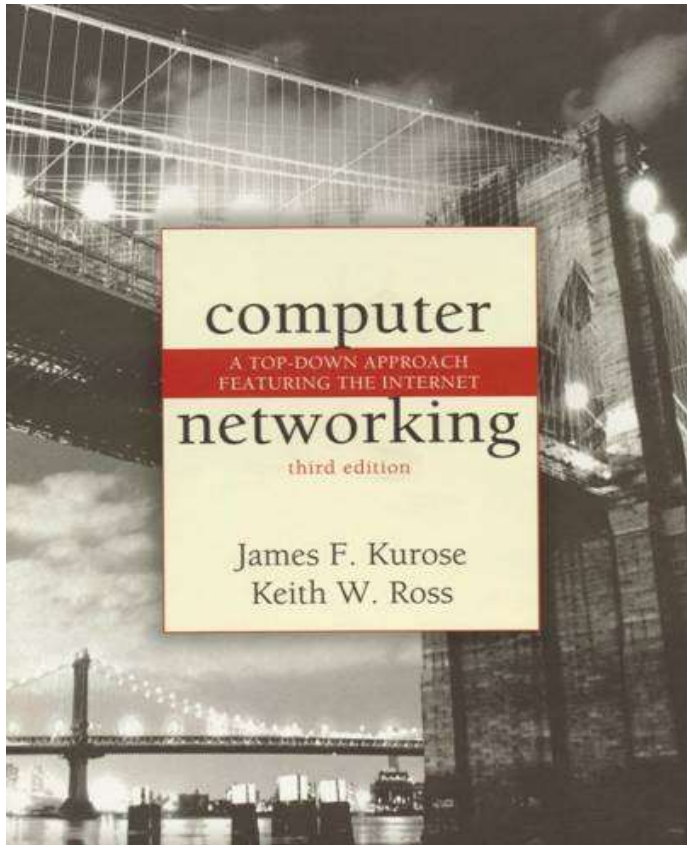




# Kênh truyền logic



# 1.8 - Một số mô hình mạng



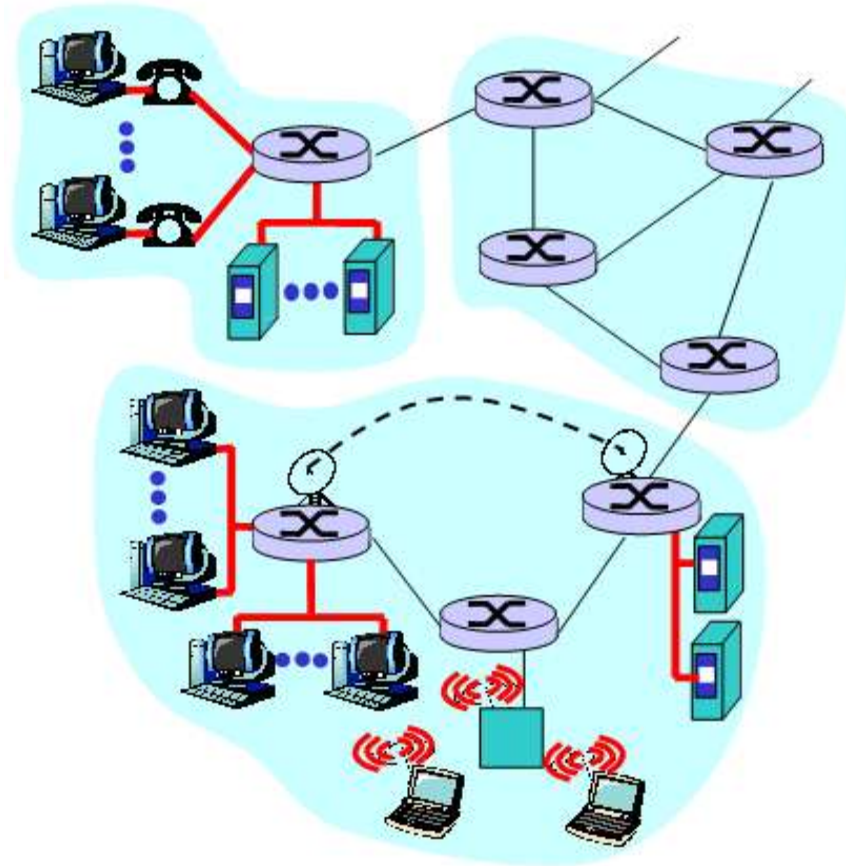
# Truy cập vào Mạng và Môi trường Vật lý

## ❑ Làm thế nào để kết nối thiết bị đầu cuối vào các Router ?

- Truy cập từ nhà riêng
- Truy cập từ cơ quan, trường học
- Truy cập qua mạng di động

## ❑ *Chú ý:*

- Băng thông của đường truyền (bps)
- Chia sẻ hay Dùng riêng?



# Truy cập từ nhà

## ❑ Quay số qua modem

- Truy nhập trực tiếp tới router, tốc độ lên đến 56Kbps. (theo lý thuyết)

## ❑ ISDN: mạng số các dịch vụ tích hợp : tốc độ 128Kbps cho mọi kết nối số tới router

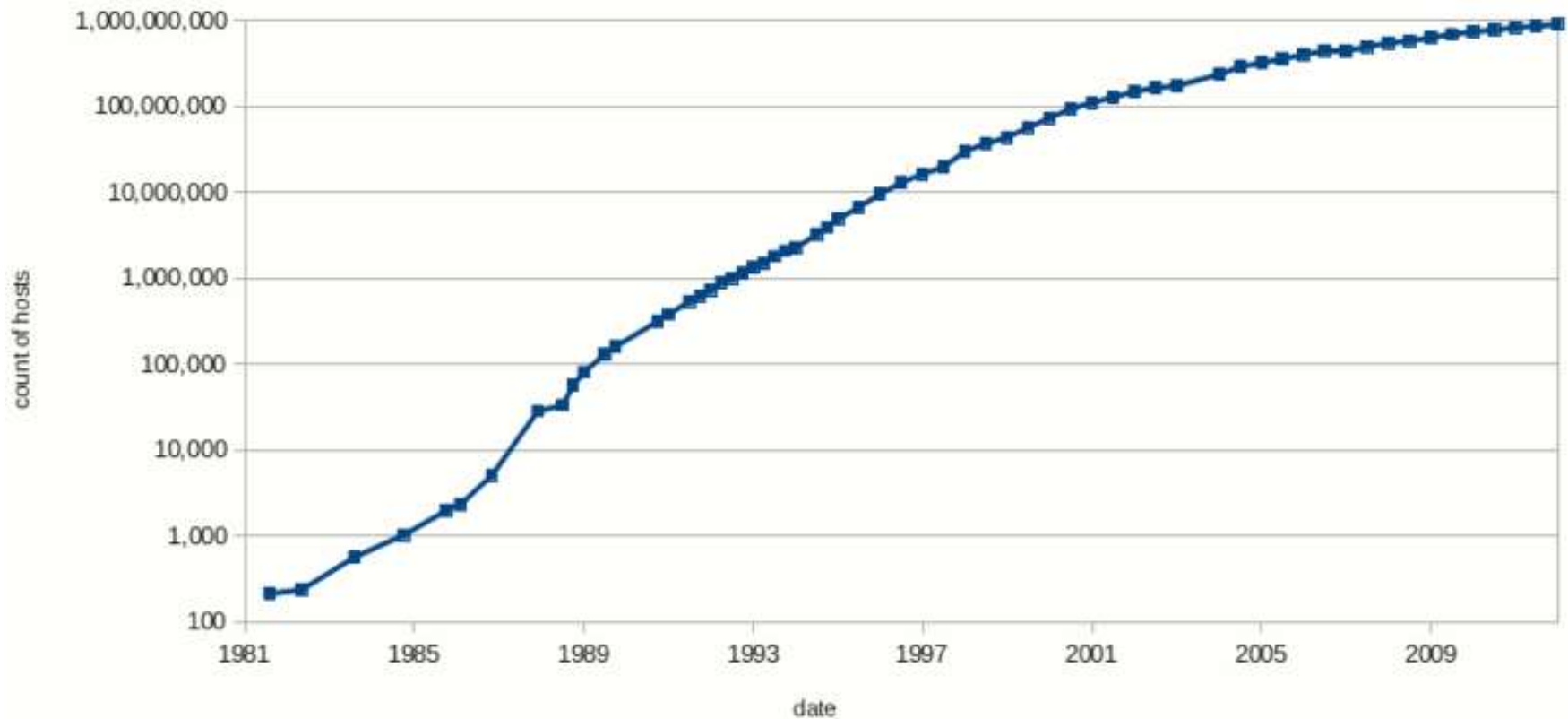
## ❑ ADSL: mạng thuê bao số không đồng bộ (asymmetric digital subscriber line)

- up to 1 Mbps home-to-router
- up to 8 Mbps router-to-home
- Sự khai triển ADSL

# 1.9 Lịch sử phát triển Internet

Internet hosts 1981-2012

<https://www.isc.org/solutions/survey/history>

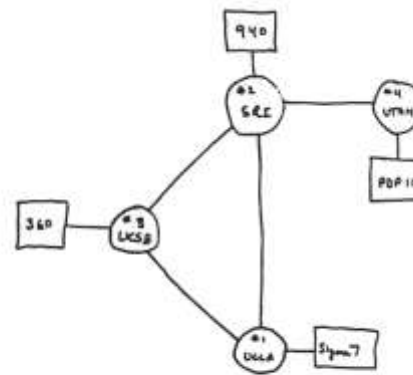


# Lịch sử phát triển Internet

*1960 - 1972: Thời kỳ có các nguyên lý chuyển gói*

- ❑ 1961: Kleinrock - chứng minh hiệu quả của chuyển gói
- ❑ 1964: Baran - chuyển gói trong các mạng quân đội
- ❑ 1967: ARPAnet hình thành từ Advanced Research Projects Agency
- ❑ 1969: nút ARPAnet đầu tiên hoạt động

- ❑ 1972:
  - ARPAnet phổ biến rộng rãi
  - NCP (Network Control Protocol) giao thức host-host đầu tiên
  - chương trình e-mail đầu tiên
  - ARPAnet có 15 nút



THE ARPA NETWORK

# Lịch sử phát triển Internet

*1972-1980: Internetworking, các mạng riêng và mới*

- ❑ 1970: ALOHAnet mạng vệ tinh ở Hawaii
- ❑ 1974: Cerf và Kahn - kiến trúc sư của mạng toàn cầu
- ❑ 1976: Ethernet tại Xerox PARC
- ❑ những năm 70: kiến trúc: DECnet, SNA, XNA
- ❑ Cuối những năm 70: chuyển các gói độ dài cố định (tiền thân của ATM)
- ❑ 1979: ARPAnet có 200 nút

**Nguyên lý mạng toàn cầu của Cerf và Kahn:**

- yêu cầu tối thiểu, tự quản-không thay đổi bên trong nào được đòi hỏi
- mô hình dịch vụ tốt nhất
- định tuyến phi trạng thái
- điều khiển tập trung

**định nghĩa kiến trúc của Internet ngày nay**

# Lịch sử phát triển Internet

*1980-1990: các giao thức mới, sự gia tăng phát triển*

- ❑ 1983: xuất bản TCP/IP
- ❑ 1982: định nghĩa giao thức email SMTP
- ❑ 1983: DNS định nghĩa cách chuyển đổi tên-địa chỉ IP
- ❑ 1985: giao thức FTP được định nghĩa
- ❑ 1988: điều khiển tắc nghẽn TCP
- ❑ Các mạng quốc gia mới: Csnet, BITnet, NSFnet, Minitel
- ❑ 100,000 hosts được kết nối vào liên minh các mạng



# Lịch sử phát triển Internet

*1990, những năm 2000: thương mại hóa, Web, các ứng dụng mới*

- ❑ những năm đầu 1990: ARPAnet ngừng hoạt động
  - ❑ 1991: NSF chấm dứt những hạn chế của NSFnet (ngừng hoạt động, 1995)
  - ❑ những năm đầu 1990: Web
    - hypertext [Bush 1945, Nelson 1960's]
    - HTML, HTTP: Berners-Lee
    - 1994: Mosaic, Netscape
  - ❑ những năm cuối 1990: thương mại hóa Web
- cuối những năm 1990 - những năm 2000:
- ❑ Nhiều ứng dụng ra đời: tin nhắn nhanh, chia sẻ file P2P
  - ❑ bảo mật mạng
  - ❑ Ước lượng khoảng 50 triệu host, hơn 100 triệu người dùng
  - ❑ liên kết backbone chạy với tốc độ Gbps

# Một số câu hỏi thảo luận

1. Mạng máy tính là gì? Chức năng cơ bản của mạng máy tính? Các thành phần chủ yếu của mạng máy tính?
2. Trình bày khái quát đặc trưng cơ bản của đường truyền: băng thông (bandwidth), thông lượng (throughput) và suy hao (attenuation)
3. Trong kỹ thuật chuyển mạch kênh, vai trò của địa chỉ như thế nào?
4. Hiểu thế nào là giao thức, vai trò của giao thức trong truyền thông?
5. Hiểu thế nào về mô hình chuẩn hóa mạng, tại sao phải chuẩn hóa?
6. Hiểu thế nào về mạng client/server, mạng peer to peer

Hết Chương 1

COMPUTER FIFTH EDITION  
NETWORKING

*A Top-Down Approach*

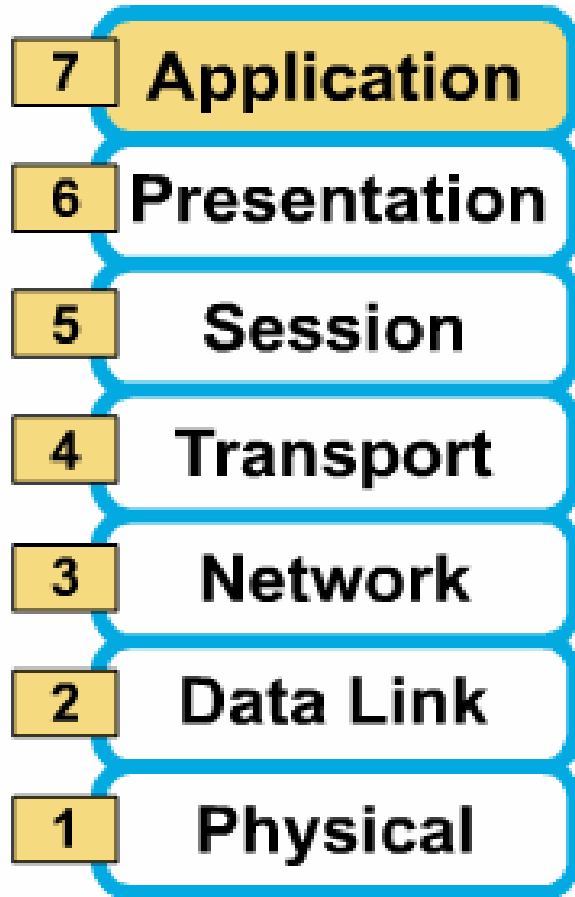


KUROSE • ROSS

## Chương 2 TẦNG ỨNG DỤNG - APPLICATION LAYER

Nhập môn mạng máy tính

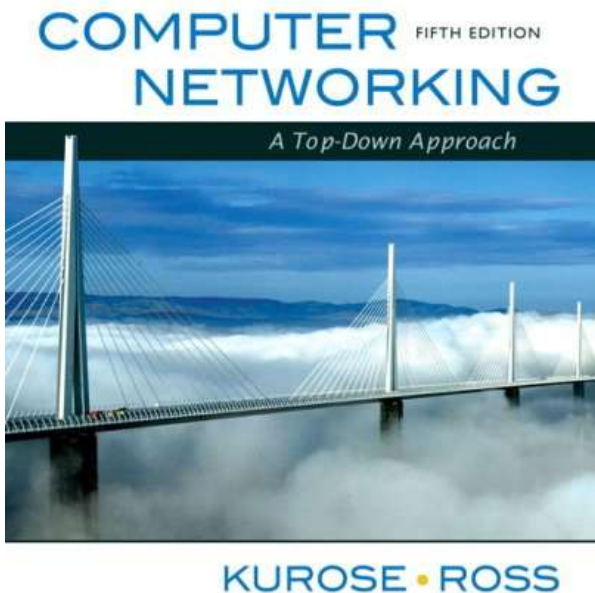
# Application Layer



## Network Processes to Applications

- Provides network services to application processes (such as electronic mail, file transfer, and terminal emulation)

# The Application Layer



- 2.1. Một số khái niệm và nguyên tắc.
- 2.2. Web & Hyper Text Transfer Protocol.
- 2.3. Web design and HTTP, Web programming.
- 2.4. File Transfer Protocol.
- 2.5. Electronic Mail Protocols.
- 2.6. Domain Name System.

## 2.1. Một số khái niệm và nguyên tắc

- Một số chương trình ứng dụng (network applications)
  - Email (Yahoo! Mail, Gmail, FPT mail...).
  - Web browser (MS Internet Explorer, Netscape Navigator...).
  - Instant messenger (Yahoo! Messenger, ICQ...).
  - P2P file sharing (Napster, KazaA, eMule...).
  - Internet games (Yahoo! Games).
  - Internet telephone (Skype, FPTFone, VNNFone...).
  - Real-time video conference.
  - Massive parallel computing.

# Một số giao thức (application-layer protocols)

- ❑ HTTP (Hyper Text Transfer Protocol)
  - Giao thức web.
- ❑ FTP (File Transfer Protocol)
  - Giao thức truyền tệp.
- ❑ SMTP (Simple Mail Transfer Protocol)
  - Giao thức truyền thư điện tử.
- ❑ POP (Post Office Protocol)
  - Giao thức nhận thư điện tử.
- ❑ IMAP (Internet Mail Access Protocol)
  - Giao thức nhận thư điện tử (khác POP3)
- ❑ DNS (Domain Name System)
  - DNS = hệ thống tên miền, giao thức DNS quy định quy tắc sử dụng tên miền.
- ❑ TFTP (Trivial FTP)
  - Một dạng khác của FTP.



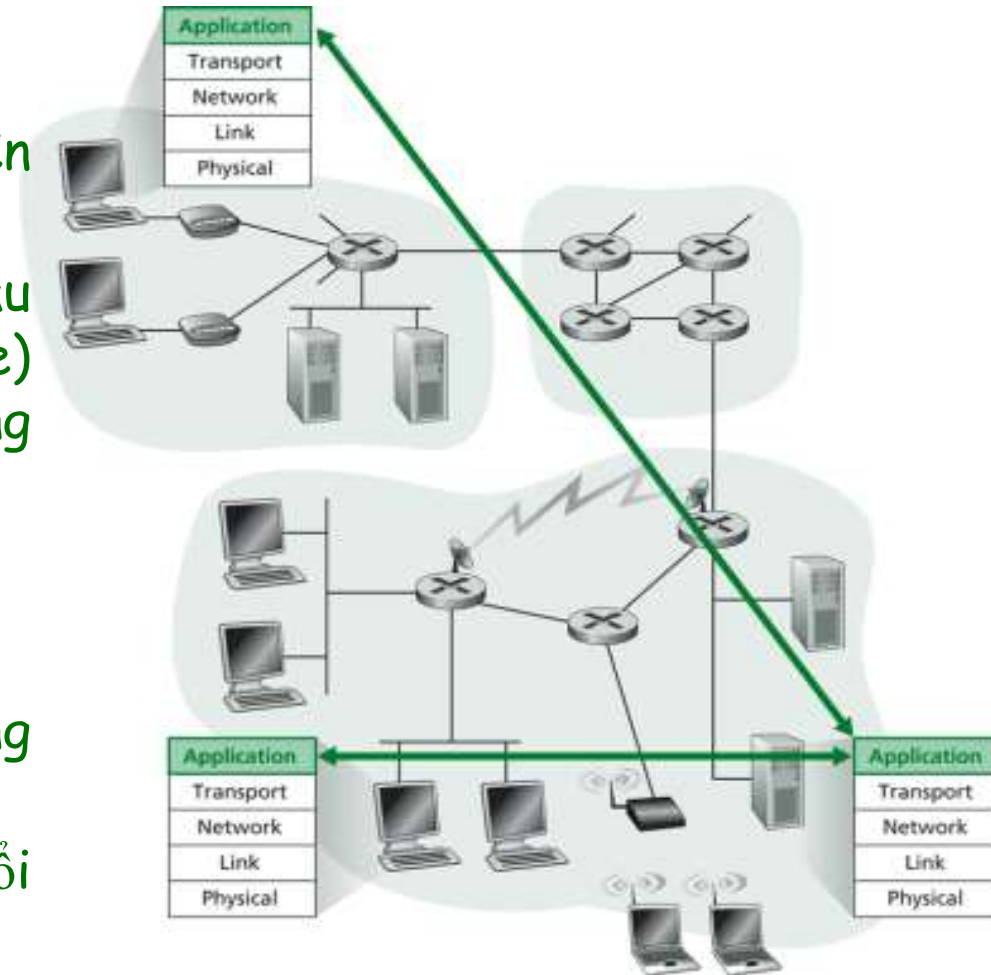
# Applications và Application-layer protocols

## □ Applications:

- Chương trình ứng dụng chạy trên hệ điều hành => 1 tiến trình
- Các tiến trình trao đổi với nhau qua các thông điệp (message) nhằm thực thi vai trò của ứng dụng.
- Vd: web browser, email client...

## □ Application-layer protocols:

- Là một thành phần của chương trình ứng dụng.
- Định nghĩa các quy tắc trao đổi thông điệp.
- Giao tiếp với các dịch vụ được cung cấp bởi tầng dưới (TCP, UDP).



# Các chương trình giao tiếp với nhau ntn?

- ❑ Các chương trình trên cùng một máy tính
  - Sử dụng các dịch vụ được cung cấp bởi hệ điều hành (*interprocess communication*).
- ❑ Các chương trình trên các máy tính khác nhau
  - Giao tiếp thông qua giao thức của tầng ứng dụng (*application-layer protocol*).
- ❑ Chương trình phía người dùng = *user agent*
  - Giao tiếp với người sử dụng (giao diện sử dụng).
  - Giao tiếp với các giao thức mạng.

# Các mô hình của ứng dụng

- ❑ Client-server
- ❑ Peer-to-peer (P2P)
- ❑ Lai giữa client-server và P2P

# Mô hình khách-phục vụ (client-server)

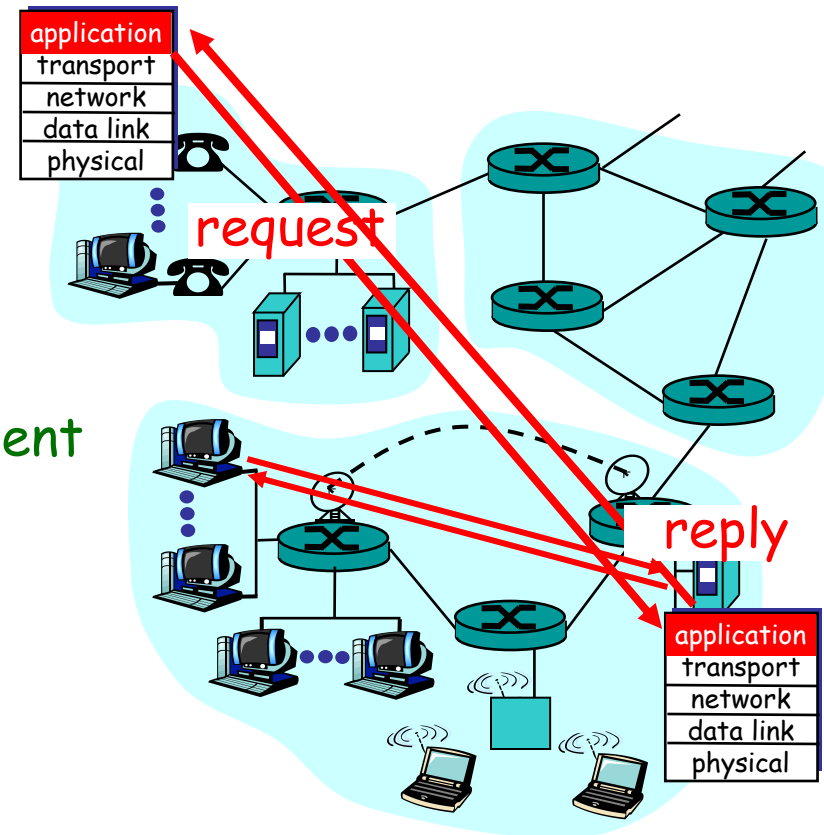
Mô hình khách-phục vụ được quen gọi là mô hình khách - chủ.  
Một ứng dụng cụ thể bao gồm hai phần:

## ❑ Client

- Thiết lập liên kết với server.
- Yêu cầu dịch vụ từ server.
- Vd: web browser.

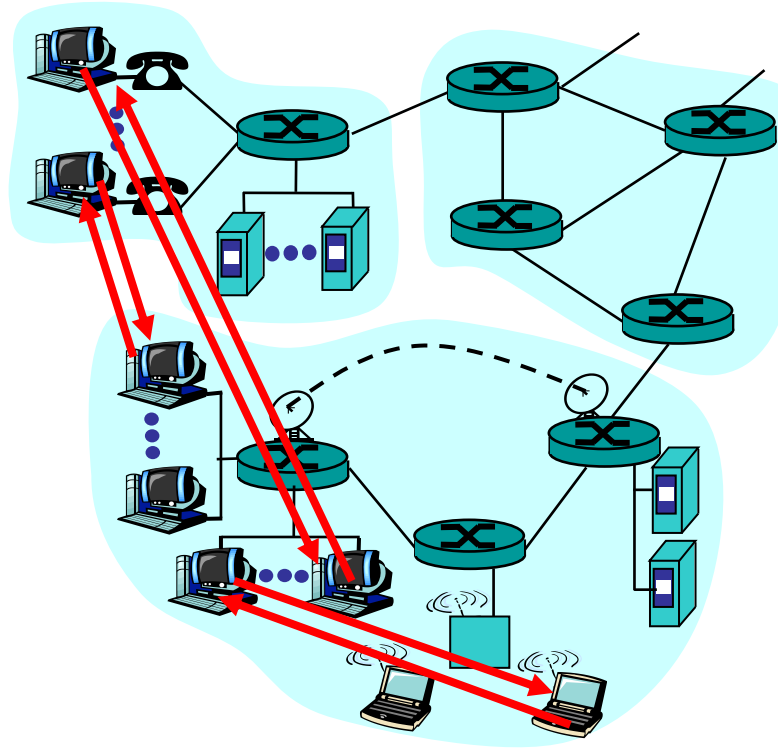
## ❑ Server

- Chờ và cung cấp các dịch vụ khi client yêu cầu.
- Vd: web server.

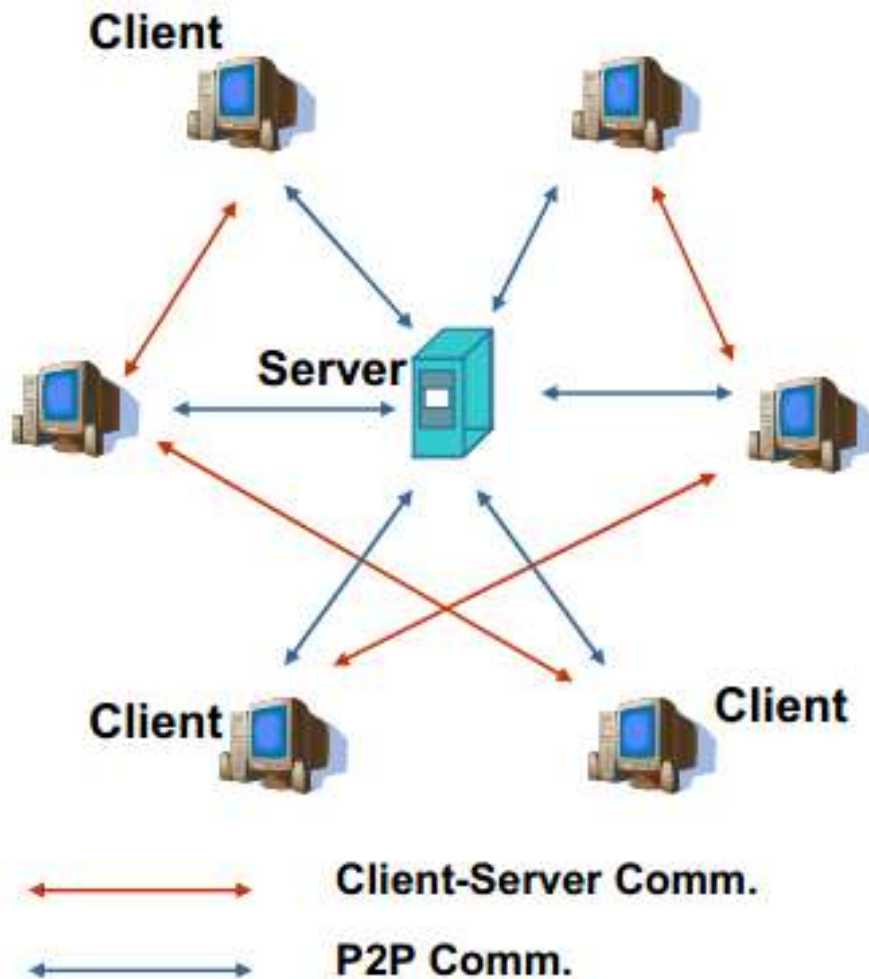


# Mô hình ngang hàng (Peer-to-peer)

- ❑ Không có máy dịch vụ cố định, các máy tính trong mạng lúc này là server (nhận yêu cầu kết nối) nhưng lúc khác lại là client (yêu cầu máy khác kết nối).
- ❑ Các máy tính là *ngang hàng*, tự do kết nối và giao tiếp đôi một.
- ❑ Địa chỉ của các máy tính tham gia giao tiếp liên tục thay đổi.



# Mô hình lai



- ❑ Máy chủ quản lý đăng nhập, thông tin sử dụng chung
- ❑ Các máy khách trao đổi trực tiếp với nhau sau khi đăng nhập
- ❑ Ví dụ: Yahoo Messenger, Skype,...

# Client-server & P2P examples

- ❑ CNN.com → Client-server
- ❑ Yahoo! Mail → Client-server
- ❑ Google → Client-server
- ❑ Yahoo! Messenger → Client-server & P2P
- ❑ Internet Relay Chat (IRC) → Client-server & P2P

# Yêu cầu đối với việc vận chuyển dữ liệu

## □ Data loss

- Các ứng dụng audio/video có thể chấp nhận mất mát, sai lệch dữ liệu trong khoảng cho phép.
- Email, file transfer yêu cầu dữ liệu phải chính xác.

## □ Timing

- Các ứng dụng online cần độ trễ (delay) nhỏ (phone, games).

## □ Bandwidth

- Các ứng dụng multimedia cần đường truyền tốc độ cao để đảm bảo chất lượng.
- Các ứng dụng email, file transfer mềm dẻo hơn, tùy theo tốc độ.



# Các yêu cầu giao vận của một số ứng dụng

<b>Application</b>	<b>Data loss</b>	<b>Bandwidth</b>	<b>Time Sensitive</b>
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	loss-tolerant	elastic	no
real-time audio/video	loss-tolerant	audio: 5Kb-1Mb video: 10Kb-5Mb	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps up	yes, 100's msec
financial apps	no loss	elastic	yes and no

# Internet Transport protocols: TCP & UDP

## ❑ TCP (Transmission Control Protocol)

- ❑ **Hướng nối** : Yêu cầu thiết lập kết nối giữa client và server.
- ❑ **Truyền dữ liệu tin cậy** giữa tiến trình gửi và nhận
- ❑ **Điều khiển lưu lượng** : bên gửi sẽ không làm "lụt" bên nhận.
- ❑ **Kiểm soát tắc nghẽn**: điều chỉnh tốc độ gửi khi mạng quá tải.
- ❑ **Không hỗ trợ** : thời gian, băng thông tối thiểu.

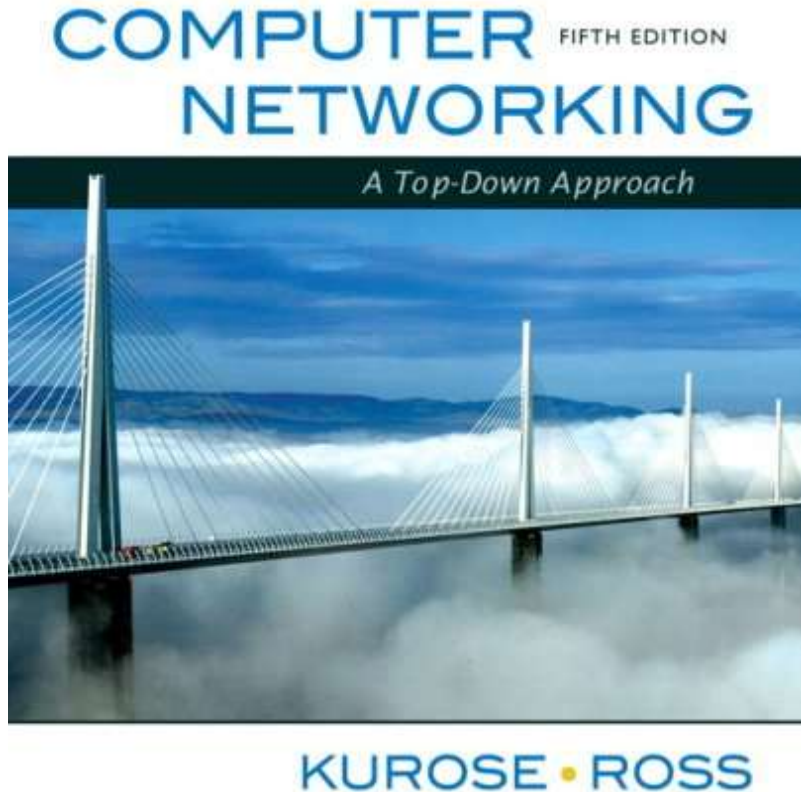
## ❑ UDP (User Datagram Protocol)

- ❑ **Truyền dữ liệu không tin cậy** giữa các tiến trình gửi và nhận.
- ❑ **Không hỗ trợ** : thiết lập kết nối, độ tin cậy, điều khiển lưu lượng, kiểm soát tắc nghẽn, thời gian, băng thông tối thiểu.

# Internet applications: TCP or UDP?

<b>Application</b>	<b>Application layer protocol</b>	<b>Underlying transport protocol</b>
e-mail	SMTP [RFC 821]	TCP
remote terminal access	telnet [RFC 854]	TCP
Web	HTTP [RFC 2068]	TCP
file transfer	ftp [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
remote file server	NSF	TCP or UDP
Internet telephony	proprietary (e.g., Vocaltec)	typically UDP

## 2.2. Web & Hyper Text Transfer Protocol.



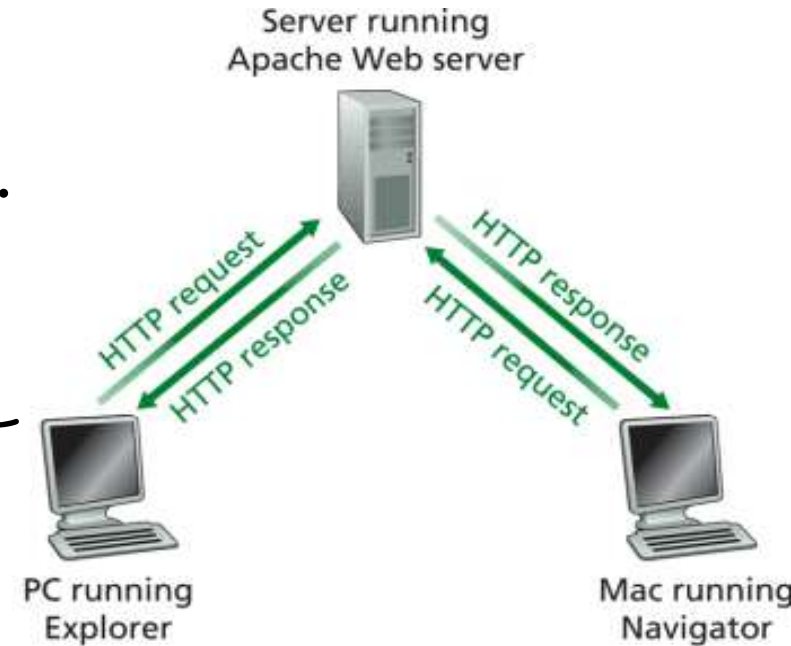
## 2.2. Web & HTTP

- ❑ Trang web (web page): một loại tài liệu có chứa nhiều đối tượng: text, image, audio, Java applet, HTML...
- ❑ Về cơ bản, web page = HTML file (base HTML).
  - HTML = Hyper Text Markup Language.
  - HTML sử dụng các thẻ (tag) để mô tả đối tượng chứa trong nó.
- ❑ Mỗi đối tượng được chỉ ra bởi địa chỉ URL (Uniform Resource Locator)
  - <http://www.qnu.edu.vn:80/cs/index.htm>
  - <http://www.qnu.edu.vn:80/cs/images/calendar.jpg>

Protocol      Host name & port      Path

# HTTP (Hyper Text Transfer Protocol)

- ❑ Là giao thức tầng ứng dụng.
- ❑ Sử dụng mô hình client/server.
  - Client (user agent):
    - Trình duyệt web (web browser)
    - Yêu cầu truy cập web thông qua URL (URL request).
  - Server
    - Máy phục vụ web (web server).
    - Gửi các đối tượng tới client thông qua phản hồi (response).
- ❑ HTTP version:
  - HTTP 1.0: RFC 1945
  - HTTP 1.1: RFC 2068



# HTTP: Quá trình trao đổi thông điệp

- ❑ Client (browser) thiết lập liên kết tới web server (TCP connection) qua cổng 80 (mặc định).
- ❑ Server đồng ý kết nối (accept).
- ❑ Các thông điệp (HTTP messages) được trao đổi giữa browser và server.
- ❑ Ngắt kết nối.

HTTP là giao thức "không trạng thái": không lưu lại yêu cầu của client

# HTTP example

Giả sử người dùng truy cập URL: [www.qnu.edu.vn/cs/index.htm](http://www.qnu.edu.vn/cs/index.htm)  
(trang web có text và 10 hình ảnh jpeg)

1a. http client thiết lập liên kết TCP với http server (process) tại địa chỉ [www.qnu.edu.vn](http://www.qnu.edu.vn), cổng 80 (ngâm định với http server).

1b. http server tại máy phục vụ [www.qnu.edu.vn](http://www.qnu.edu.vn) chờ yêu cầu kết nối TCP tại cổng 80, chấp nhận kết nối rồi thông báo với client.

2. http client gửi *http request message* (bao gồm cả URL) tới TCP connection socket

3. http server nhận request message, tạo ra *http response message* có chứa các đối tượng được yêu cầu rồi gửi vào socket.

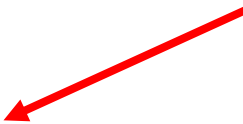
time





time



4. http server ngắt liên kết.
  5. http client nhận response message có chứa html file, hiển thị html. Sau đó, phân tích html file, tìm URL của 10 hình ảnh jpeg trong tài liệu.
  6. Bước 1-5 được lặp lại với từng hình ảnh.
- 

# HTTP Connection: Persistent & Non-persistent

## HTTP Không liên tục

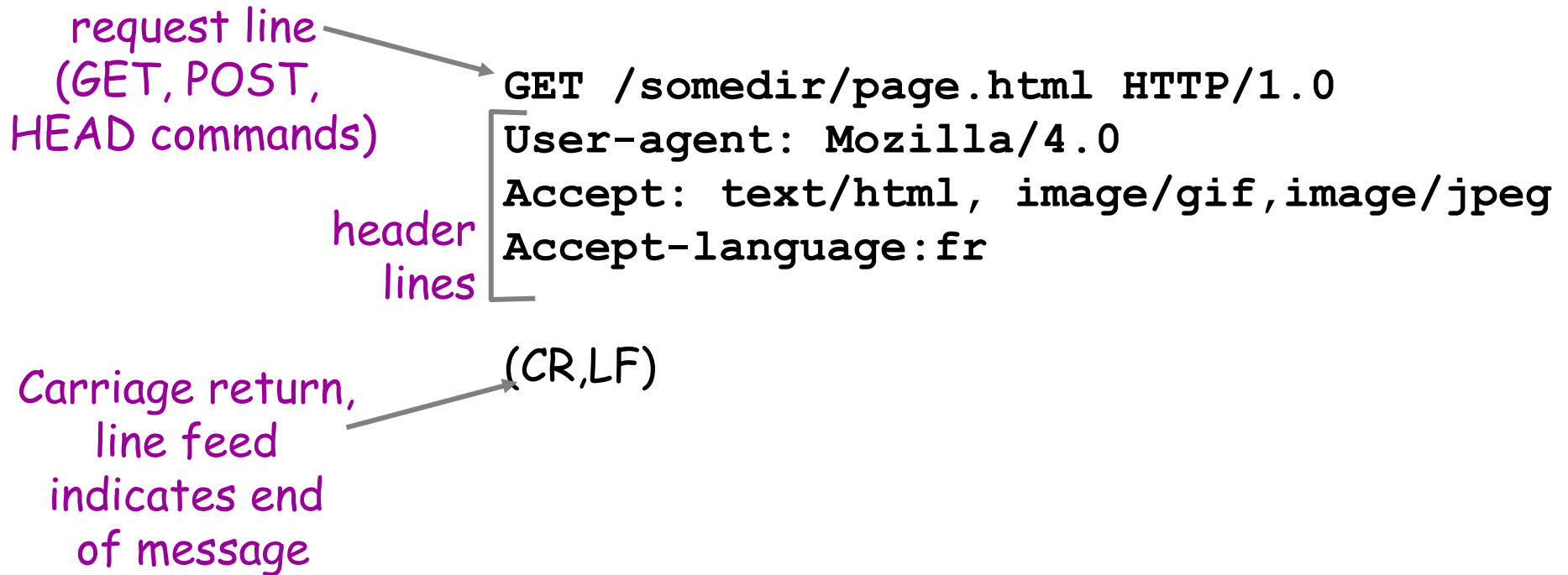
- ❑ Chỉ có một đối tượng web gửi qua liên kết TCP.
- ❑ Sử dụng mặc định trong HTTP/1.0

## HTTP liên tục

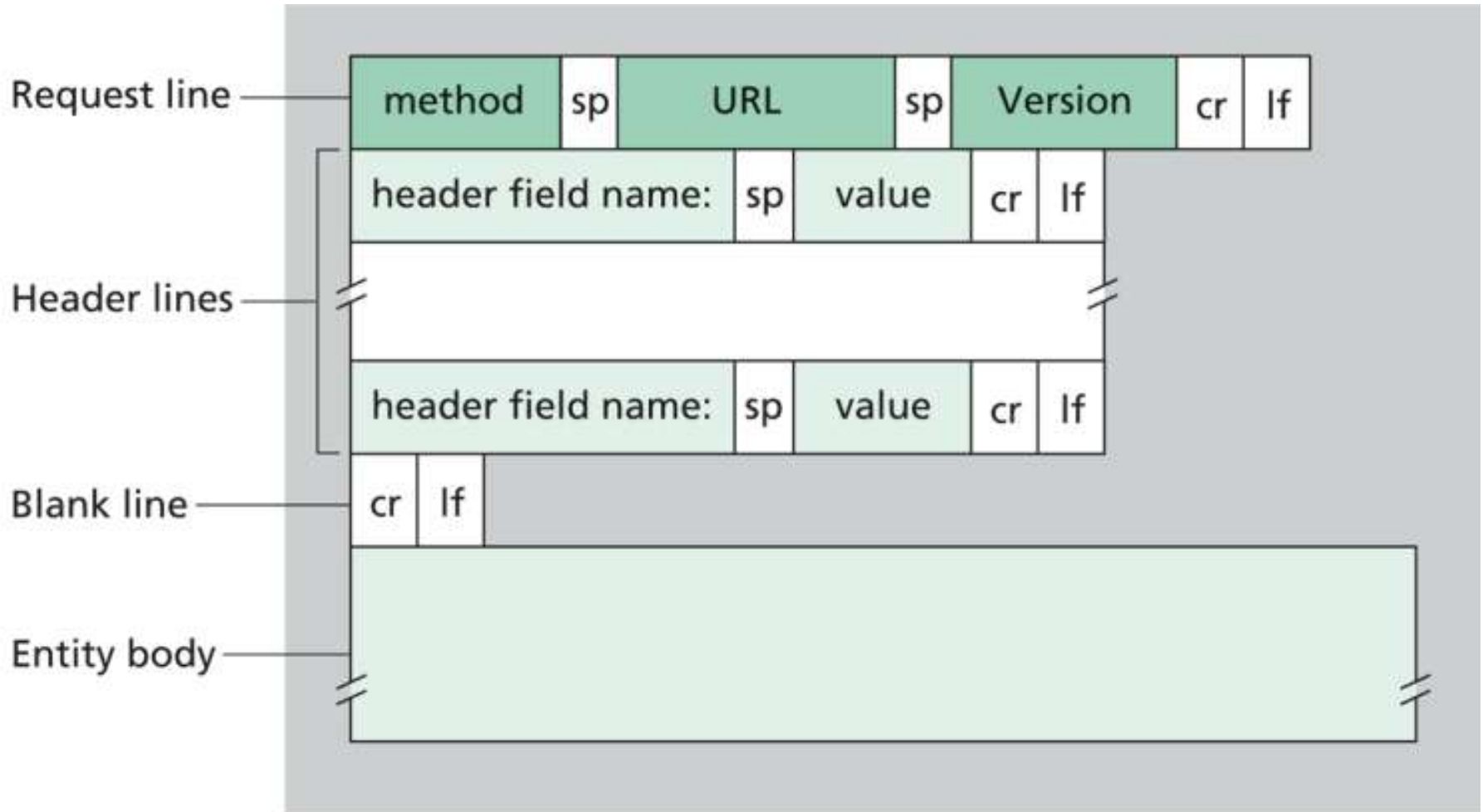
- ❑ Cho phép nhiều đối tượng được truyền trên cùng một liên kết.
- ❑ Client phân tích, tìm ra và gửi yêu cầu tới tất cả các đối tượng ngay sau khi nhận được trang HTML ban đầu (base HTML).
- ❑ HTTP 1.1 sử dụng liên kết loại này ở chế độ mặc định.

# HTTP Message Format: request & response

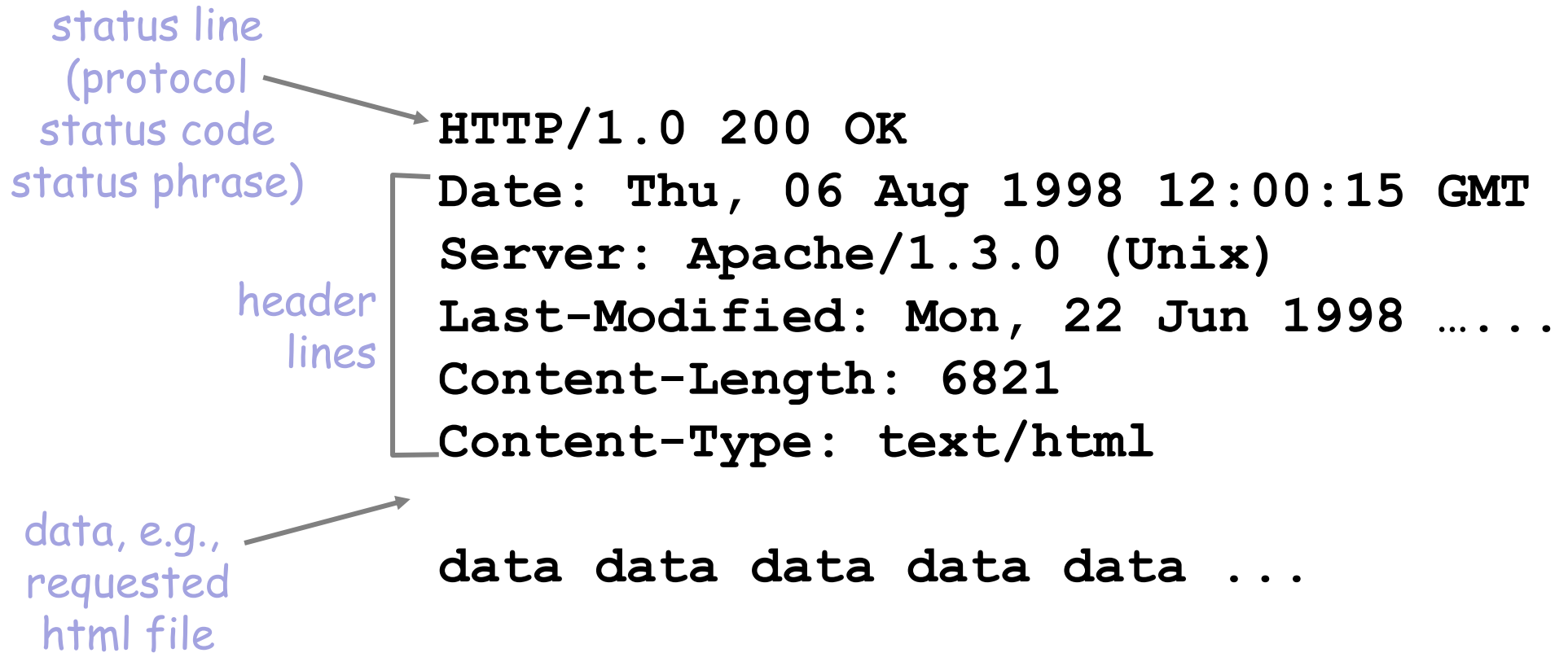
- ❑ Hai loại message: *request* và *response*.
- ❑ HTTP request message: dạng ASCII (đọc được!)



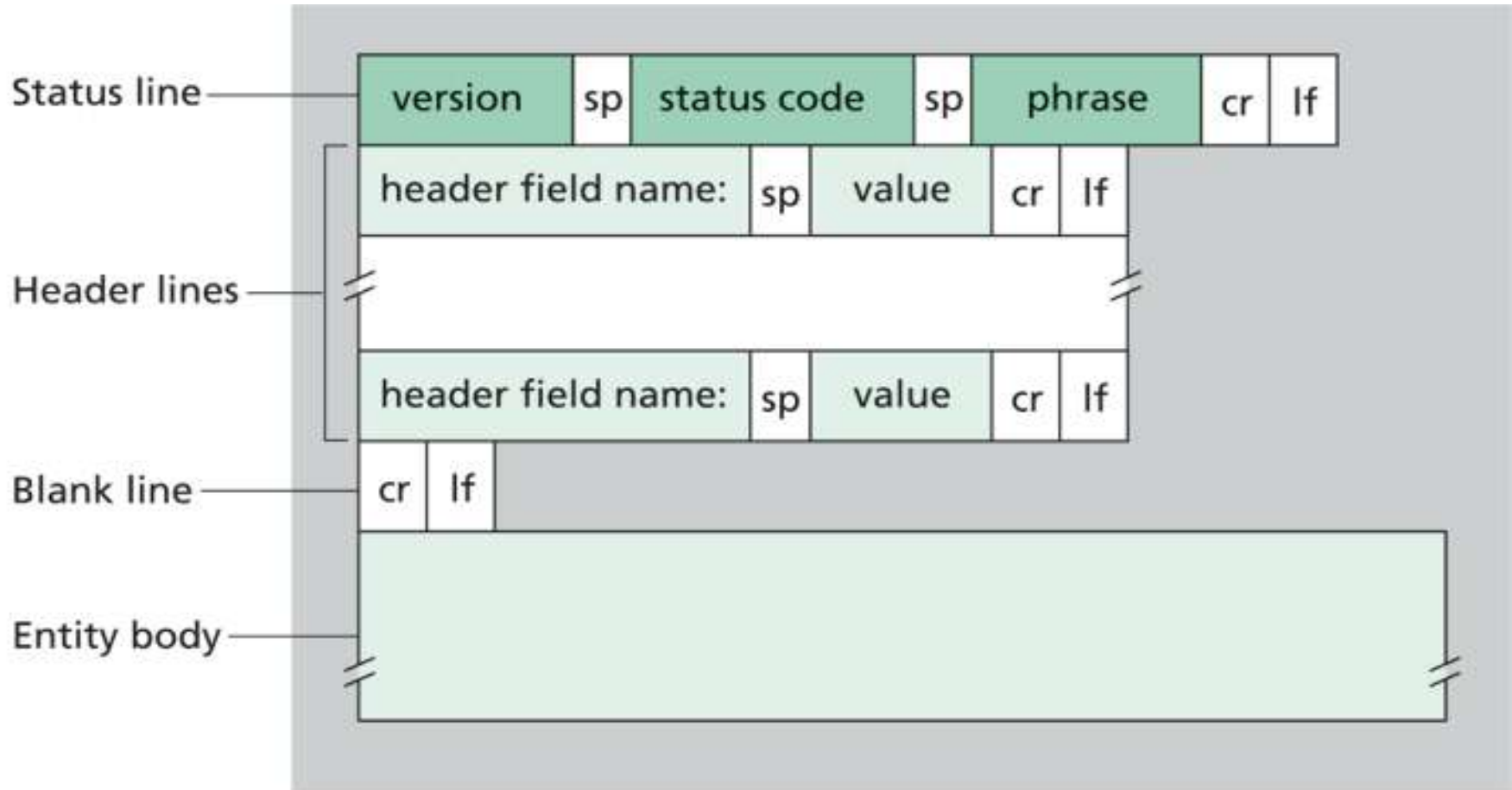
# HTTP request message



# HTTP response message



# HTTP response message



# HTTP response message: status codes

## 200 OK

- request succeeded, requested object later in this message

## 301 Moved Permanently

- requested object moved, new location specified later in this message (Location:)

## 400 Bad Request

- request message not understood by server

## 404 Not Found

- requested document not found on this server

## 505 HTTP Version Not Supported

# Ví dụ thực hành

## 1. Telnet tới web server:

```
telnet www.qnu.edu.vn 80
```

Mở một kết nối (TCP) tới cổng 80 tại web server của trường ĐHQN. Sau khi kết nối, tất cả ký tự gõ vào sẽ được gửi tới cổng 80.

## 2. Nhập vào *GET http request*.

```
GET /cs/index.htm HTTP/1.1
```

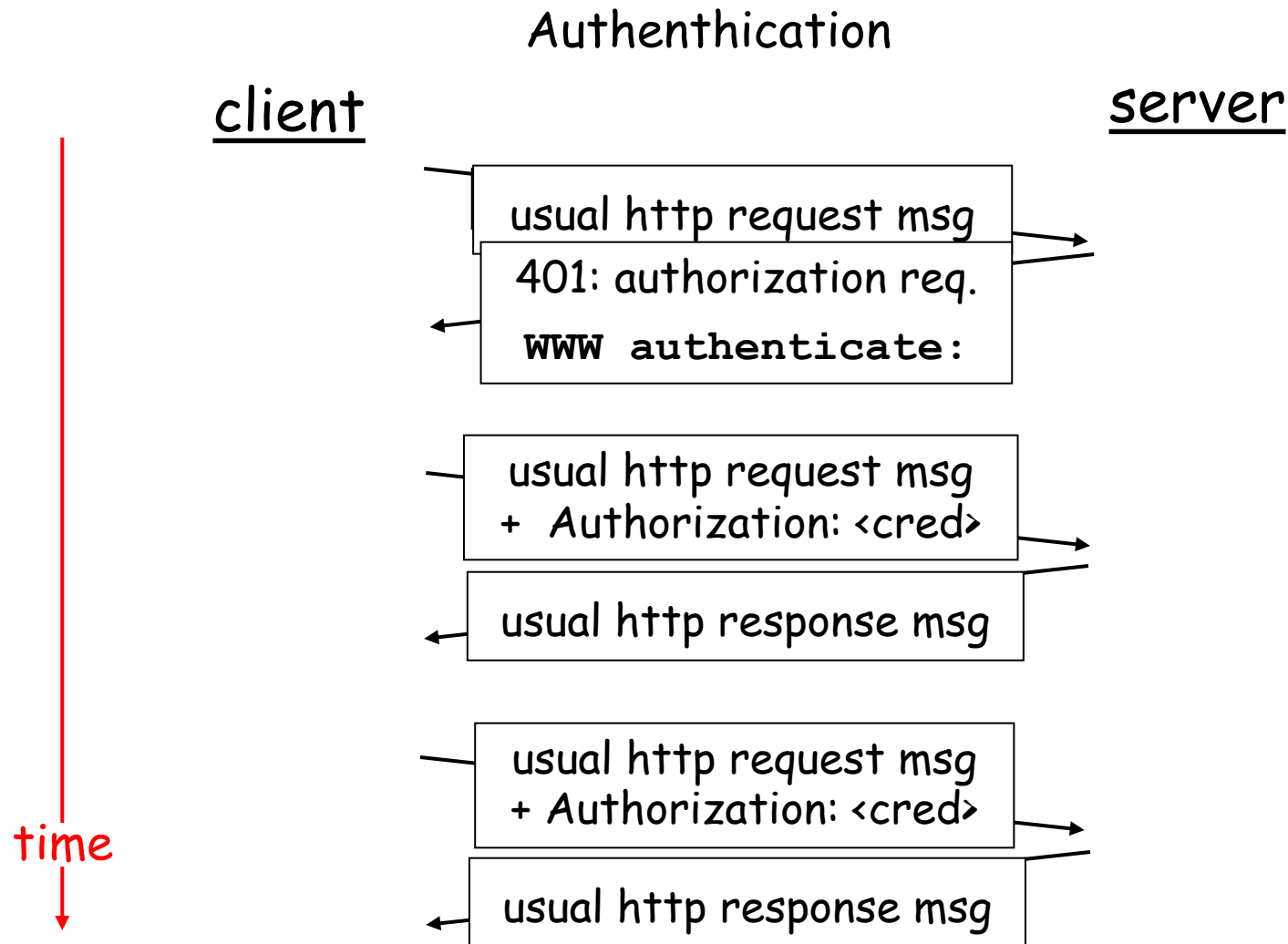
Đây là một *http request* đơn giản.

Nhập xong gõ 2 lần Enter để gửi tới server

## 3. Kiểm tra kết quả nhận được (*http response message*)!



# User-server interaction: authentication (kiểm chứng)



# Trạng thái User-server: các cookie

Nhiều Web sites dùng các cookie

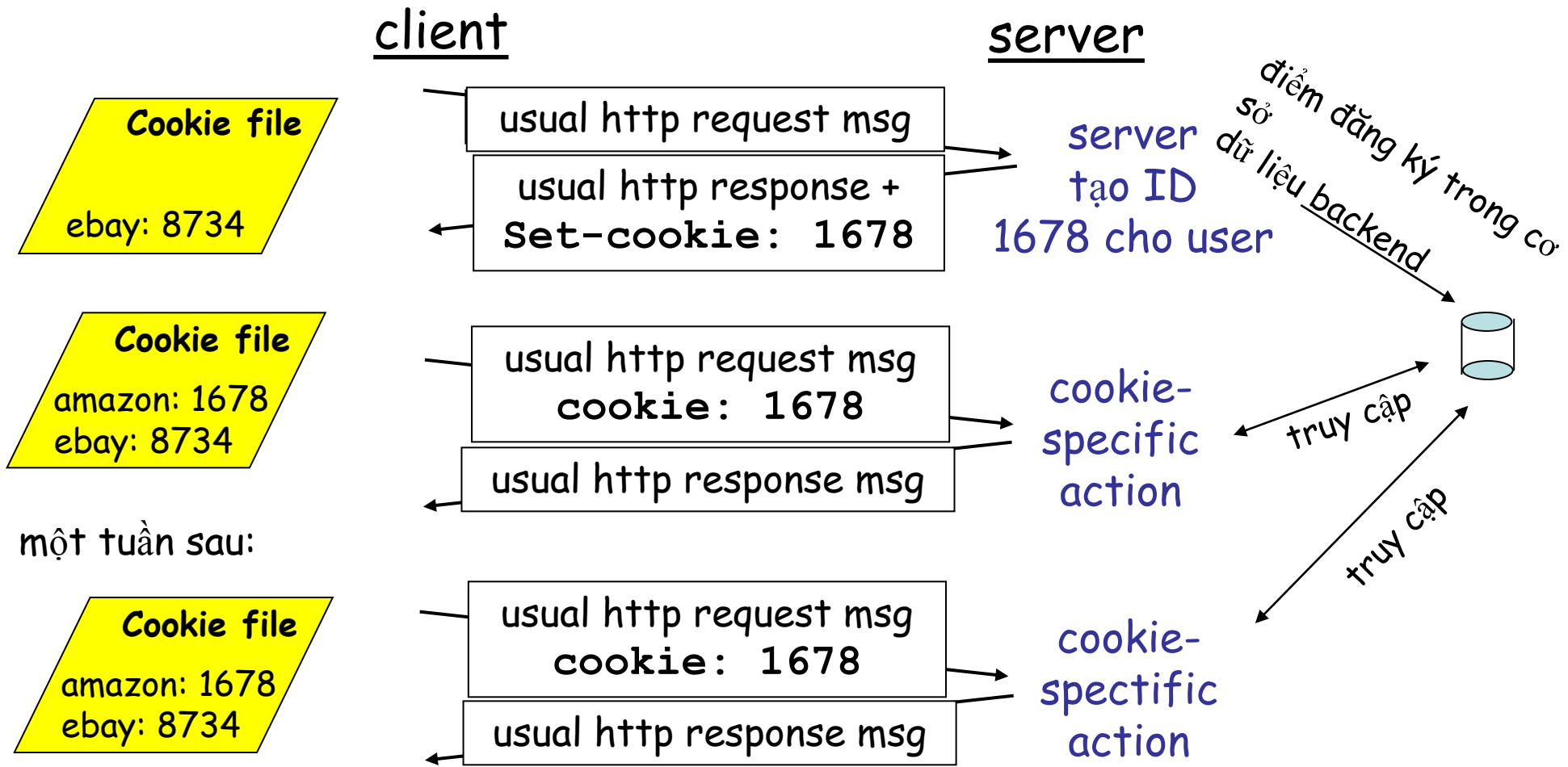
## 4 loại:

- 1) cookie header line của thông điệp phản hồi HTTP
- 2) cookie header line trong thông điệp yêu cầu HTTP
- 3) cookie file lưu trong host của user, quản lý bởi trình duyệt của user
- 4) cơ sở dữ liệu back-end tại Web site

## Ví dụ:

- Susan truy cập Internet luôn từ một PC
- She lần đầu tiên vào một e-commerce site xác định
- Khi yêu cầu khởi tạo HTTP đến site, site tạo một ID duy nhất và tạo một điểm đăng nhập trong cơ sở dữ liệu back-end cho ID đó

# Các cookie: lưu giữ "trạng thái" (tt.)



# các cookie (tiếp)

## Các cookie đem lại:

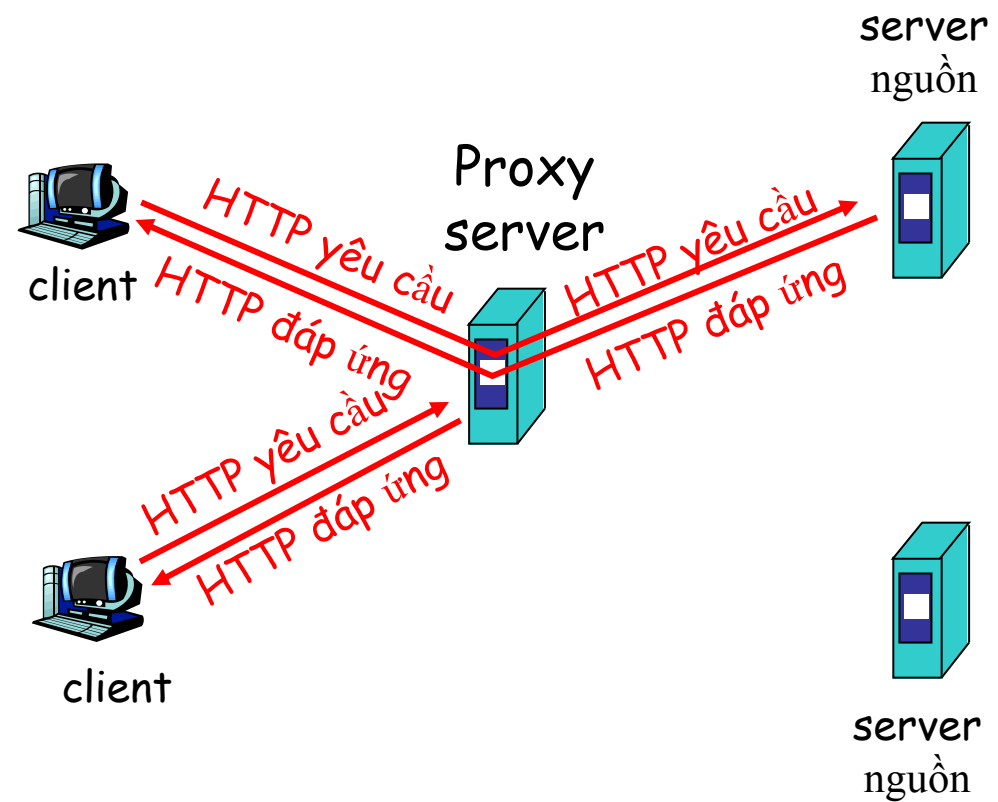
- ❑ Không cần nhập username và password nhiều lần
- ❑ các khuyến cáo
- ❑ trạng thái phiên làm việc của user (Web e-mail)

- ngoài ra
- các cookie và sự riêng tư.
- ❑ các cookie cho phép các site biết nhiều hơn về bạn
  - ❑ bạn có thể cung cấp tên và e-mail cho sites

# Web caches (proxy server)

**Mục tiêu:** thỏa mãn yêu cầu của client không cần liên quan đến server nguồn

- user thiết lập trình duyệt: truy cập Web thông qua cache
- trình duyệt gửi tất cả yêu cầu HTTP cho cache
  - đối tượng trong cache: cache trả về đối tượng
  - ngược lại cache yêu cầu đối tượng từ server nguồn, sau đó trả về cho client

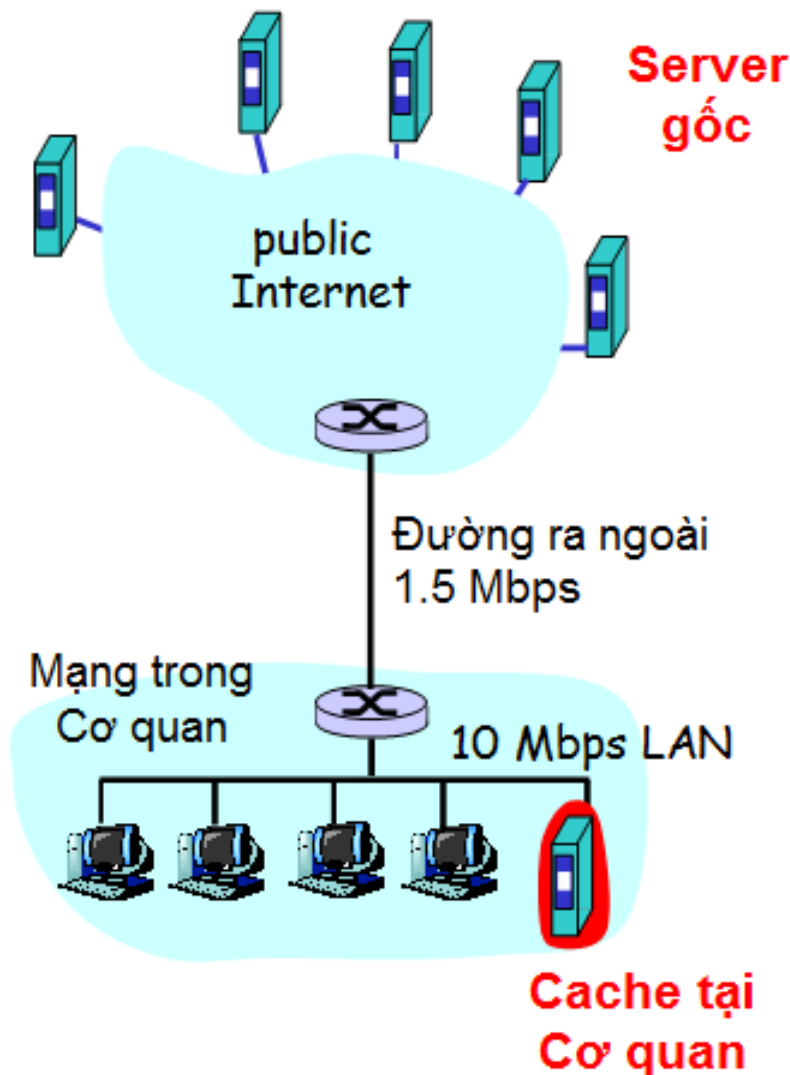


# Web caching

- ❑ Cache hoạt động tại cả client và server
- ❑ Tiêu biểu cache được cài đặt bởi ISP (trường học, công ty, ISP riêng)

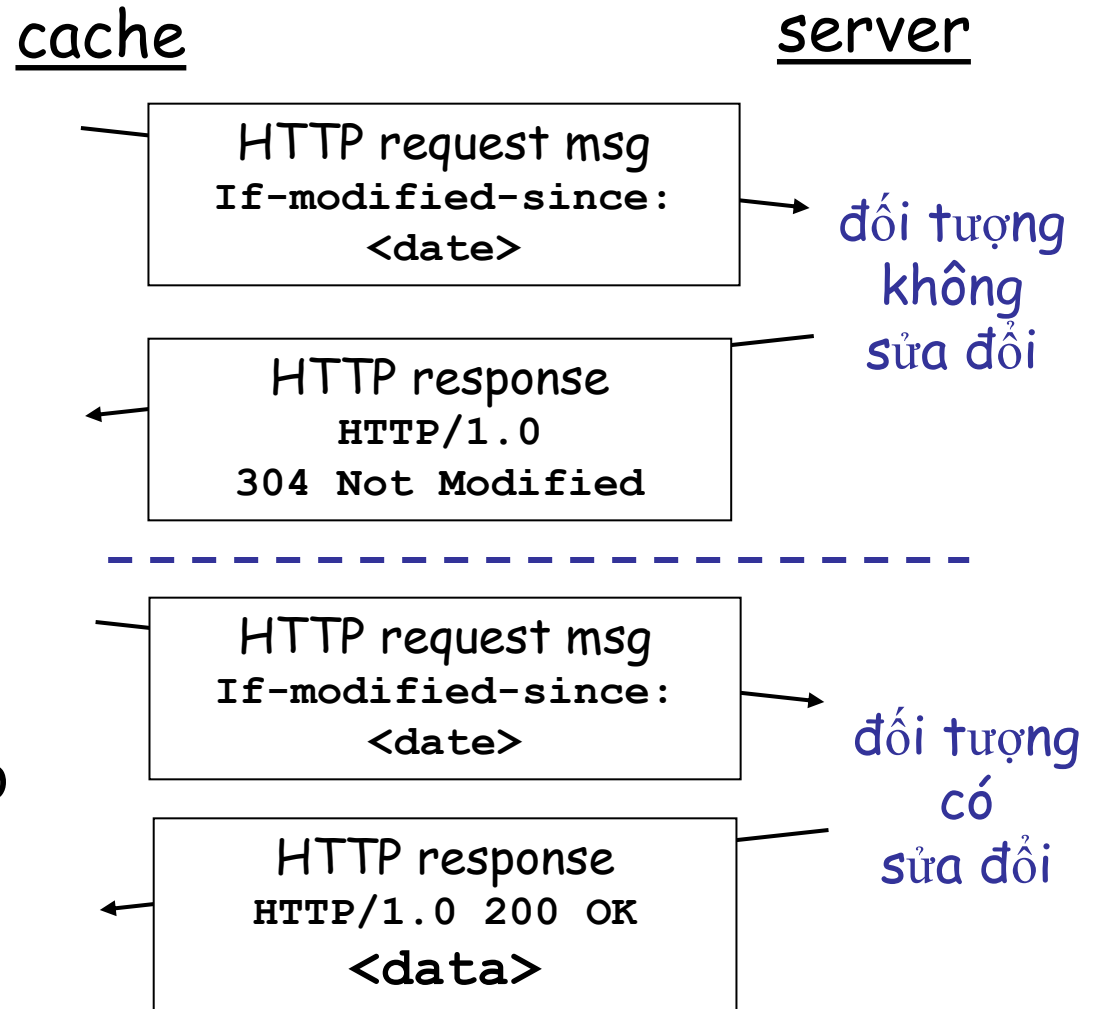
## Tại sao dùng Web caching?

- ❑ Giảm thời gian đáp ứng cho yêu cầu của client
- ❑ Giảm lưu thông trên liên kết truy cập
- ❑ Nhà cung cấp không quan tâm đến việc triển khai cache trên Internet



# GET có điều kiện

- ❑ **Mục tiêu:** không gửi đối tượng nếu cache đã cập nhật
- ❑ **cache:** xác định ngày của bản sao cache trong yêu cầu HTTP:  
**If-modified-since:**  
**<date>**
- ❑ **server:** đáp ứng không chứa đối tượng nếu bản sao cache đã cập nhật:  
**HTTP/1.0 304 Not Modified**

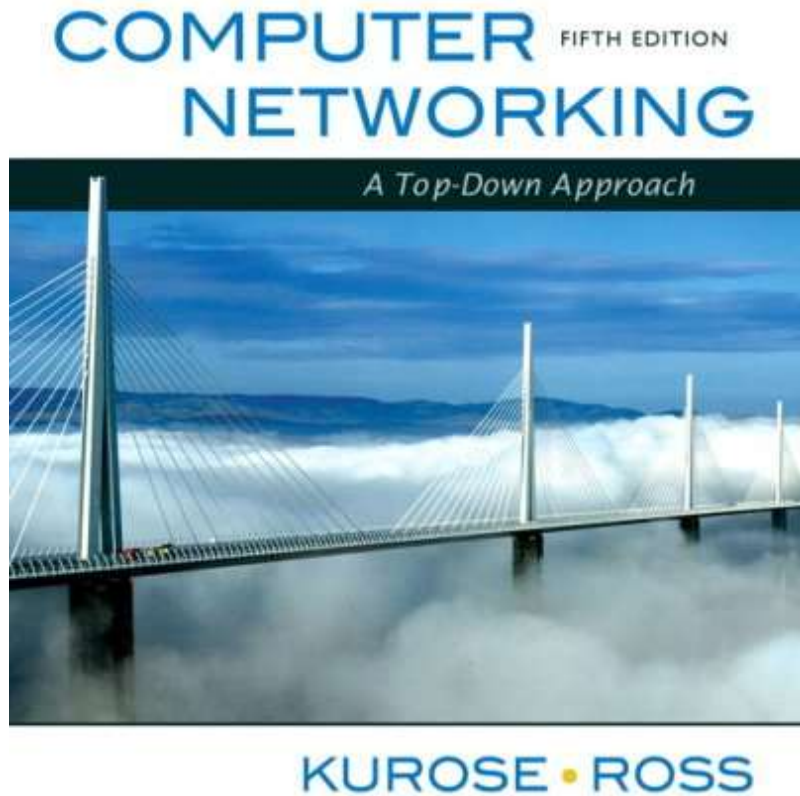


# Web servers

- ❑ IIS (Microsoft Internet Information Server)
  - Chạy trên môi trường Windows
  - Đi kèm Windows Server
- ❑ Apache web server
  - Mã nguồn mở.
  - Linux/Windows...
  - <http://www.apache.org>
  - Web server được dùng nhiều nhất.
- ❑ Tomcat, Netscape, Xitami...



## 2.3. Web design and HTTP, Web programming.



# Web design: HTML

- ❑ Thiết kế web: Tạo ra tài liệu HTML
- ❑ HTML (Hyper Text Markup Language)
  - Trong tài liệu HTML có thể có nhiều đối tượng
    - Text
    - Images
    - Hyper link
  - Sử dụng thẻ (tag) để diễn đạt đối tượng.
    - Thẻ mở, vd: <html>; <head>; <title>; <body>...
    - Thẻ đóng, vd: </html>; </head>;</title>;</body>...
- ❑ Công cụ:
  - Text editor: Notepad, Ultra Edit...
  - Microsoft FrontPage
  - Macromedia Dreamweaver
  - ...

# Web programming

- ❑ HTML = trang web tĩnh (Static web): Web server luôn trả về một trang HTML cố định khi nhận được yêu cầu của người dùng.
- ❑ Lập trình web: tương tác với người sử dụng thông qua web.
  - Java script, VB script.
  - ASP, JSP, PHP, Perl...

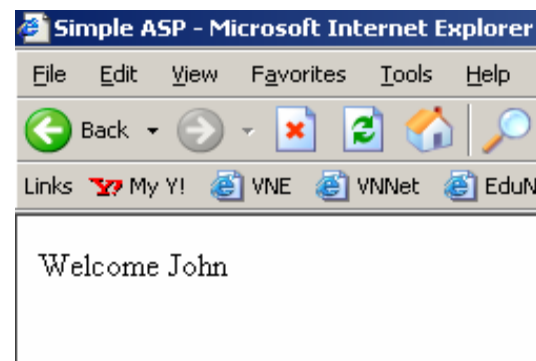
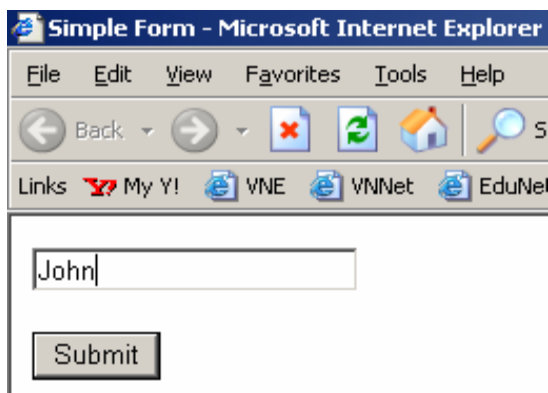
# HTML Form: Send user requests

HTML

```
<html>
<head>
<title>Simple Form</title>
</head>
<body>
<form method="POST" action = "simple.asp">
  <p><input type="text" name="myName" size="20">
  <p><input type="submit" value="Submit" name="B1">
</form>
</body>
</html>
```

ASP

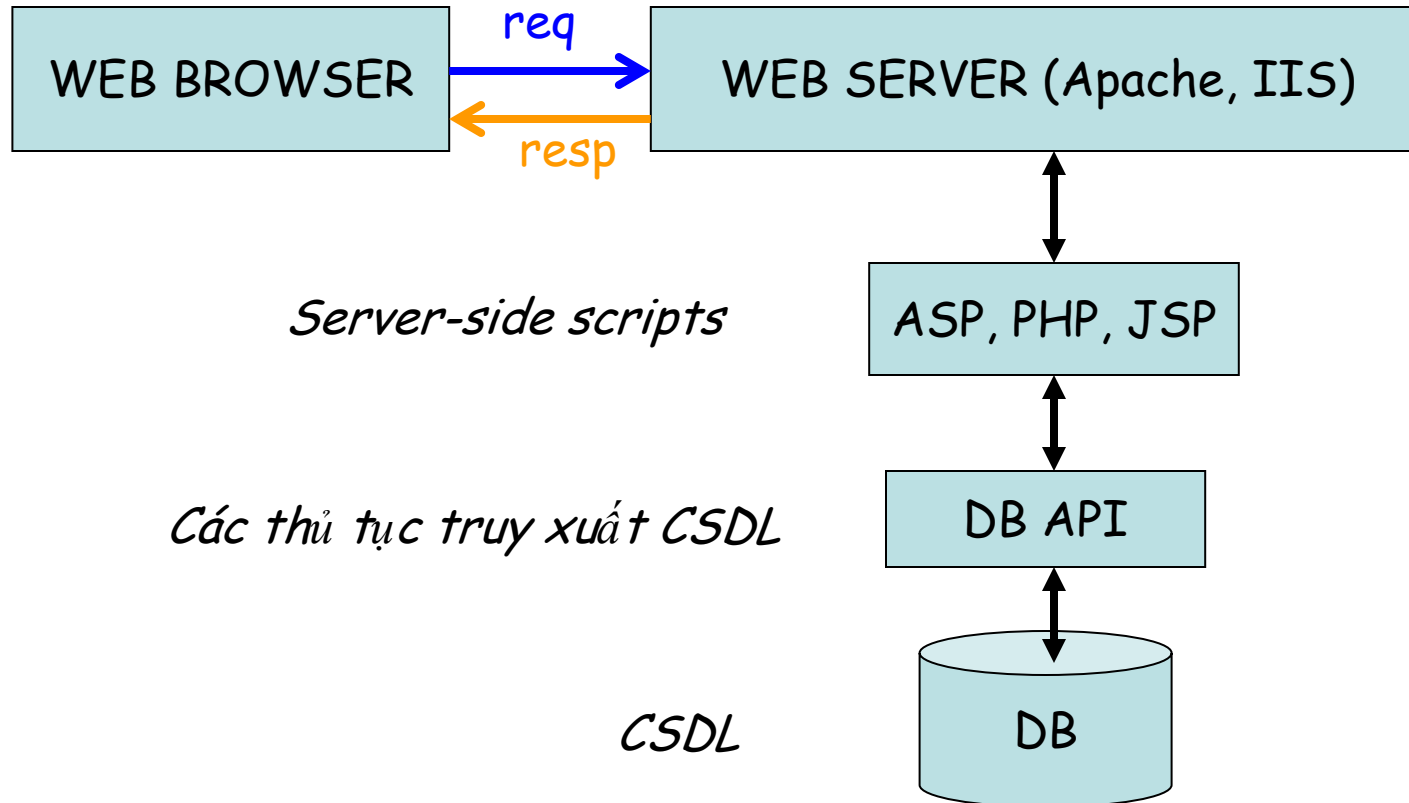
```
<%@ Language=VBScript %>
<HTML>
<HEAD>
<Title>Simple ASP</title>
</HEAD>
<BODY>
<%
  dim welcome
  dim myName
  myName=Request.Form("myName")
  welcome = "<h1>Welcome " & myName & "!</h1>"
  Response.Write (welcome )
%>
</BODY>
</HTML>
```



# ASP (Active Server Page)

- ❑ Microsoft/IIS.
- ❑ Một tài liệu ASP có thể coi như một tài liệu HTML có nhúng các script ASP, được ghi lại dạng tệp \*.asp, đặt tại web server.
- ❑ Mỗi khi được request, các script được chạy ở phía server, kết quả trả về client dạng HTML (client không thấy source).
- ❑ Các đối tượng:
  - Request → Client request.
  - Response → Server response.
  - Application
  - QueryString...
- ❑ Công cụ: MS Visual InterDev

# Mô hình 3 lớp trong lập trình web

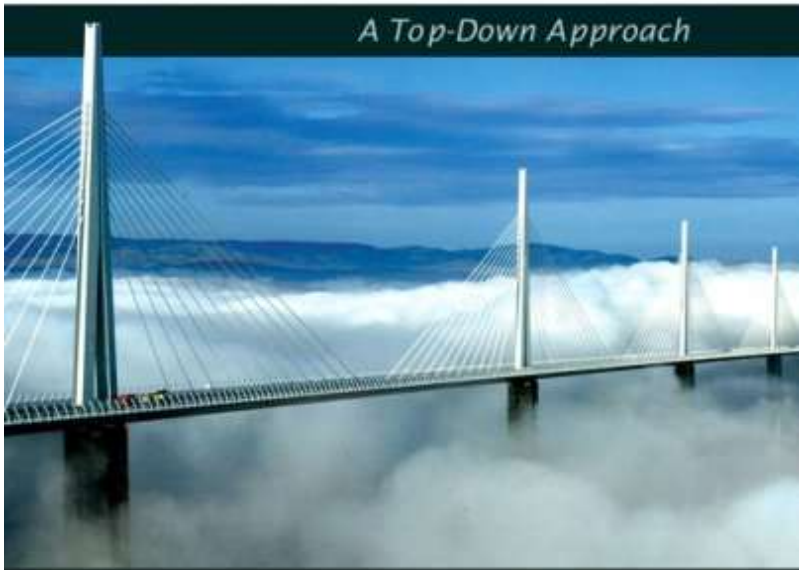


## 2.4 FTP

# COMPUTER NETWORKING

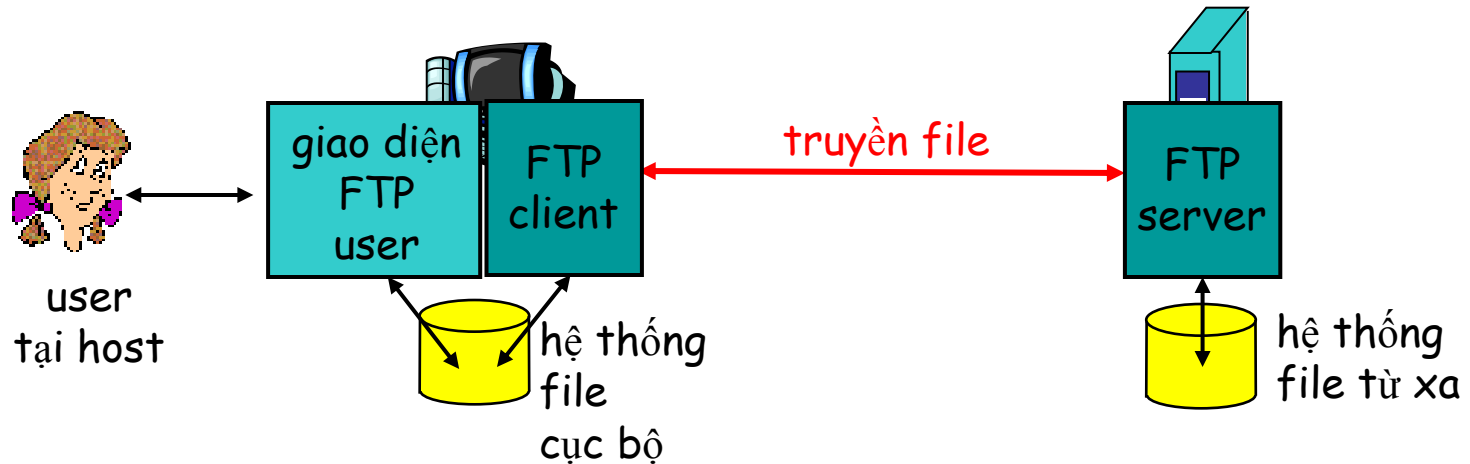
FIFTH EDITION

*A Top-Down Approach*



KUROSE • ROSS

# FTP: giao thức truyền file

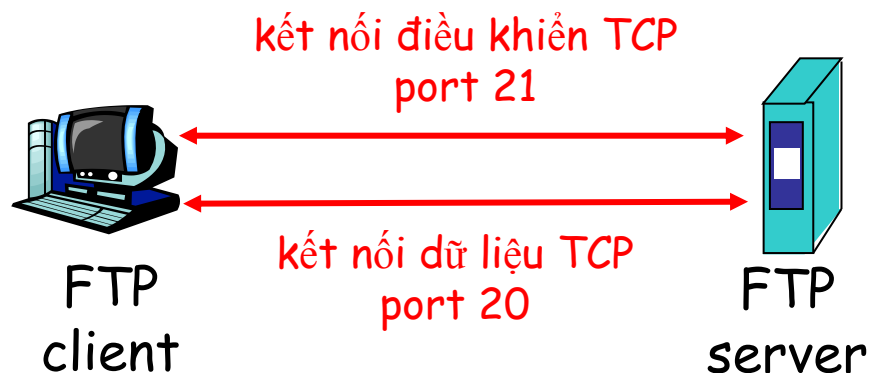


- ❑ truyền file đến/từ host từ xa
- ❑ mô hình client/server
  - *client*: phía khởi tạo truyền (đến/từ host ở xa)
  - *server*: host ở xa
- ❑ ftp: RFC 959
- ❑ ftp server: port 21



# FTP: kết nối dữ liệu, điều khiển riêng biệt

- ❑ FTP client tiếp xúc FTP server tại port 21, xác định TCP như giao thức transport
- ❑ Client lấy giấy phép thông qua kết nối điều khiển
- ❑ Client thao tác thư mục ở xa bằng việc gửi các lệnh thông qua kết nối điều khiển.
- ❑ Khi server nhận lệnh truyền file, server mở kết nối TCP thứ 2 (cho file) đến client
- ❑ Sau khi truyền 1 file, server đóng kết nối dữ liệu



- ❑ Server mở kết nối dữ liệu TCP khác để truyền file khác
- ❑ Sử dụng kênh điều khiển khác với kênh truyền dữ liệu gọi là “out of band”
- ❑ FTP server giữ lại “trạng thái”: thư mục hiện hành, giấy phép trước đó

# Các lệnh, phản hồi FTP

## Ví dụ các lệnh:

- ❑ gửi như văn bản ASCII trên kênh điều khiển
- ❑ `USER username`
- ❑ `PASS password`
- ❑ `LIST` trả về danh sách của file trong thư mục hiện hành
- ❑ `RETR filename` trích chọn (lấy) file
- ❑ `STOR filename` lưu (đặt) file vào trong host ở xa

## Ví dụ mã trả về

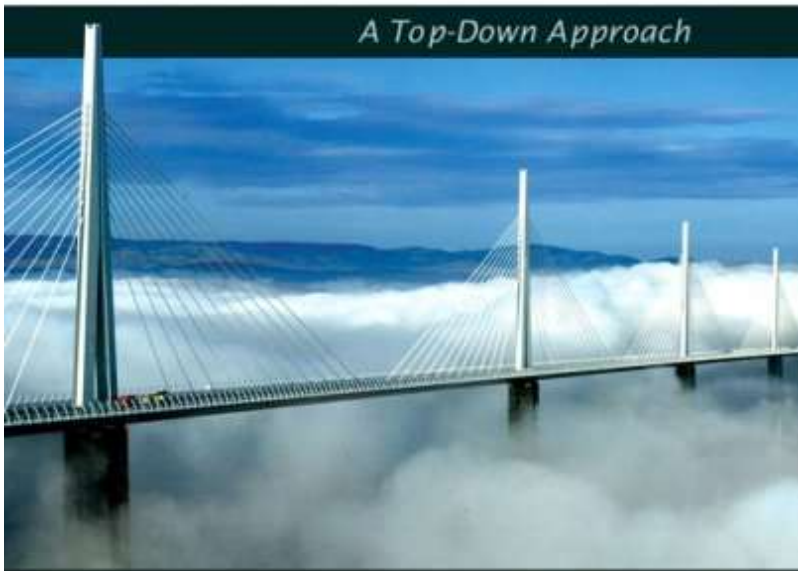
- ❑ mã trạng thái và cụm (như HTTP)
- ❑ 331 Username OK, password required
- ❑ 125 data connection already open; transfer starting
- ❑ 425 Can't open data connection
- ❑ 452 Error writing file

## 2.5 Electronic Mail

# COMPUTER NETWORKING

FIFTH EDITION

*A Top-Down Approach*



KUROSE • ROSS

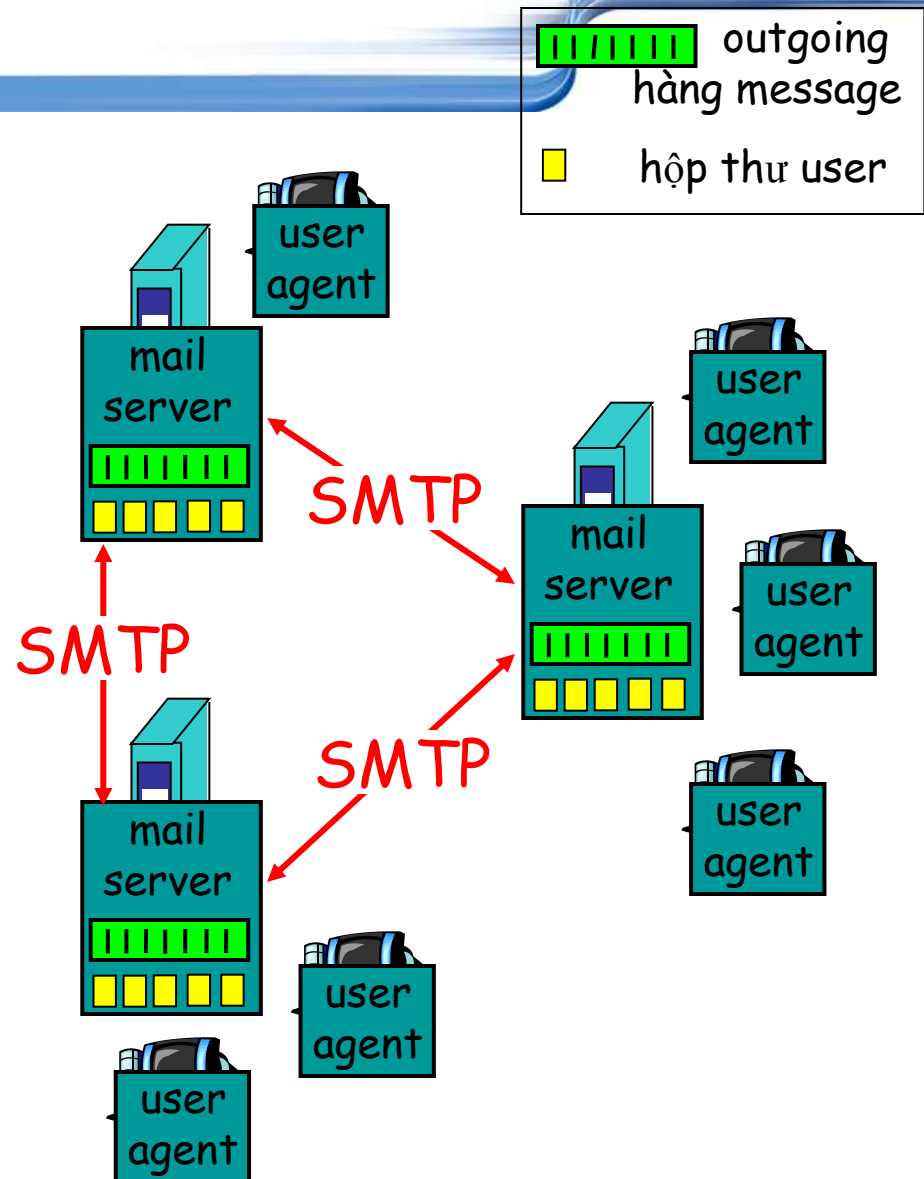
# Electronic Mail

## 3 thành phần quan trọng:

- ❑ user agents
- ❑ mail servers
- ❑ simple mail transfer protocol: SMTP

## User Agent

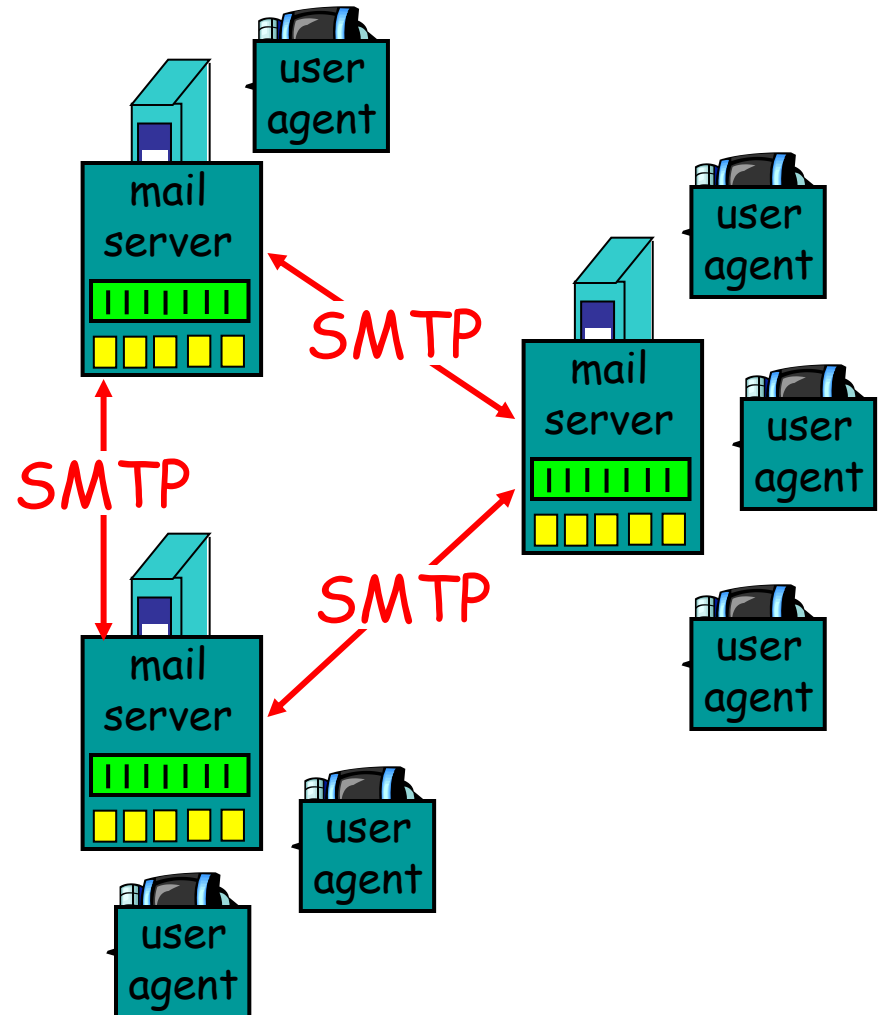
- ❑ còn gọi là "mail reader"
- ❑ viết, sửa đổi, đọc các thông điệp mail
- ❑ Ví dụ: Eudora, Outlook, elm, Netscape Messenger
- ❑ các thông điệp đi và đến được lưu trên server



# Electronic Mail: mail servers

## Mail Servers

- ❑ mailbox (hộp thư) chứa các thông điệp đến user
- ❑ hàng thông điệp cho các thông điệp email ra ngoài (chuẩn bị gửi)
- ❑ giao thức SMTP giữa các mail servers để gửi các thông điệp email
  - client: mail server gửi
  - "server": mail server nhận

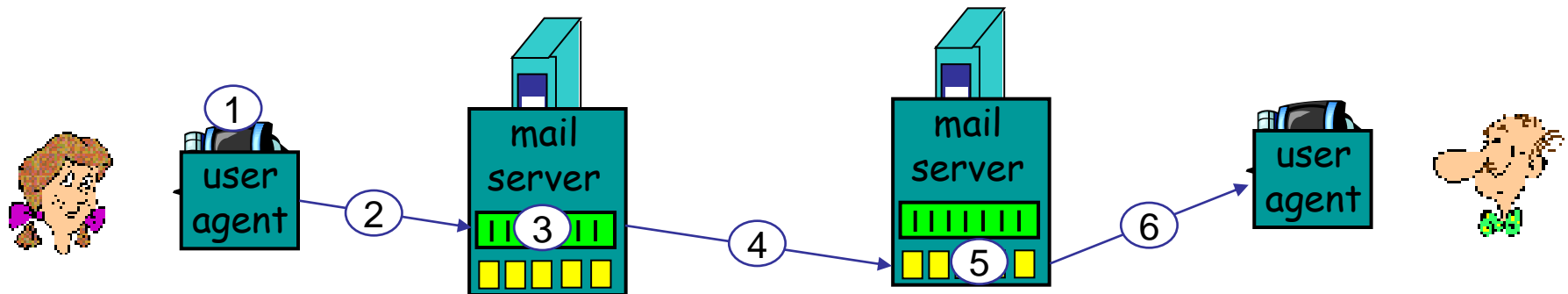


# Electronic Mail: SMTP [RFC 2821]

- ❑ dùng TCP để truyền tin cậy thông điệp email từ client đến server trên port 25
- ❑ truyền trực tiếp: server gửi đến server nhận
- ❑ 3 giai đoạn
  - bắt tay (chào hỏi)
  - truyền thông điệp
  - đóng
- ❑ tương tác lệnh/phản hồi
  - lệnh: văn bản ASCII
  - phản hồi: mã trạng thái và cụm
- ❑ các thông điệp phải ở dạng mã ASCII 7-bit

# Tình huống: Alice gửi cho Bob

- 1) Alice dùng UA viết thông điệp và "gửi đến"  
`bob@someschool.edu`
- 2) UA của Alice gửi thông điệp của cô ấy đến mail server; thông điệp được gia nhập vào hàng đợi
- 3) Phía Client của SMTP mở kết nối TCP với mail server của Bob
- 4) SMTP client gửi thông điệp của Alice trên kết nối TCP
- 5) mail server của Bob đặt thông điệp vào hộp thư của Bob
- 6) Bob kích hoạt trình user agent đọc thông điệp



# Ví dụ tương tác SMTP

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```



# Thử nghiệm tương tác SMTP:

- ❑ `telnet servername 25`
  - ❑ thấy 220 trả lời từ server
  - ❑ nhập các lệnh HELO, MAIL FROM, RCPT TO, DATA, QUIT
- lệnh trên cho phép bạn gửi email không cần dùng email client (reader)

# SMTP

- ❑ SMTP dùng các kết nối liên tục
- ❑ SMTP yêu cầu các thông điệp (header & body) phải ở dạng thức 7-bit ASCII
- ❑ SMTP server dùng CRLF.CRLF xác định kết thúc thông điệp

## So sánh với HTTP:

- ❑ HTTP: kéo
- ❑ SMTP: đẩy
- ❑ tất cả đều có tương tác lệnh/đáp ứng, các mã trạng thái ASCII
- ❑ HTTP: mỗi đối tượng được đóng kín trong thông điệp đáp ứng của nó
- ❑ SMTP: nhiều đối tượng được gửi trong thông điệp nhiều phần

# Dạng thức thông điệp email

SMTP: giao thức cho trao đổi các thông điệp email

RFC 822: chuẩn cho dạng thức văn bản:

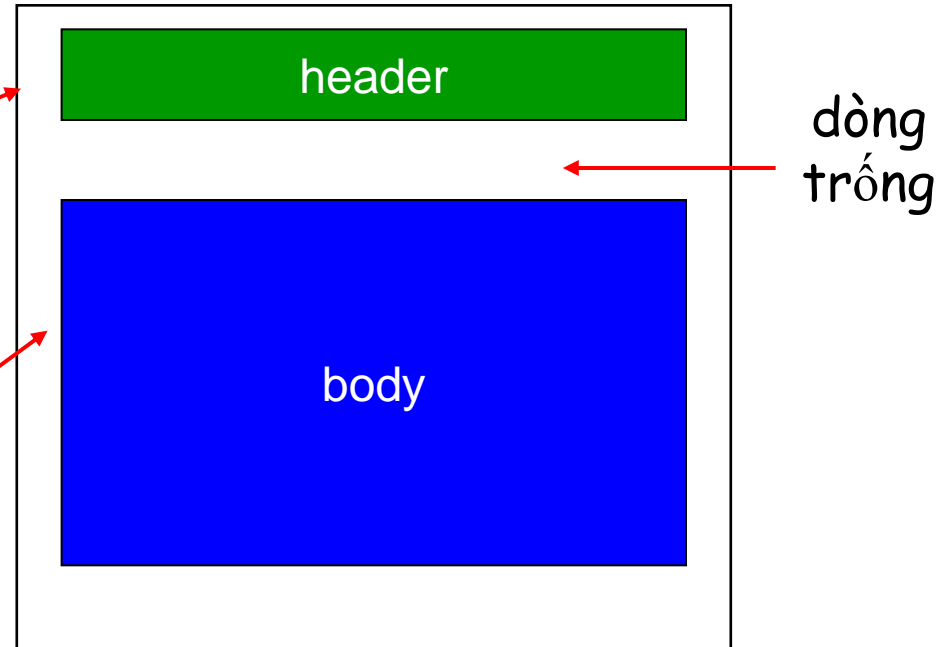
□ các dòng header, ví dụ:

- To:
- From:
- Subject:

*khác với các lệnh HTTP*

□ body

- "thông điệp", chỉ có các ký tự ASCII



# Dạng thức thông điệp: các mở rộng multimedia

- ❑ MIME: Multimedia Mail Extension, RFC 2045, 2056
- ❑ các dòng bổ sung trong header của thông điệp khai báo kiểu nội dung MIME

phiên bản MIME

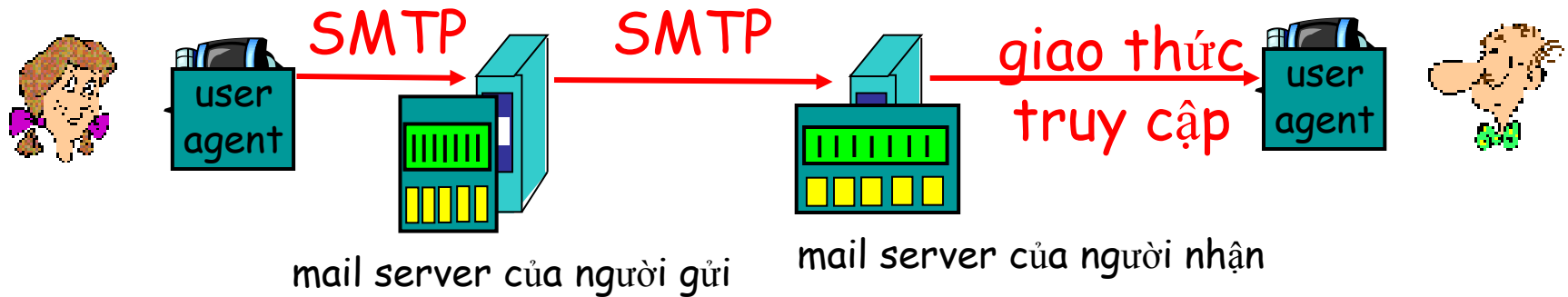
phương pháp dùng  
để mã hóa dữ liệu

kiểu, kiểu con,  
khai báo tham số  
dữ liệu multimedia

dữ liệu đã mã hóa

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....
.....
.....base64 encoded data
```

# Các giao thức truy cập email



- ❑ SMTP: truyền dẫn/lưu trữ vào server của người nhận
- ❑ Giao thức truy cập email: trích xuất từ server
  - POP: Post Office Protocol [RFC 1939]
    - cấp phép (agent <--> server) và download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - nhiều tính năng (phức tạp hơn)
    - điều khiển các thông điệp đã lưu trên server
  - HTTP: Hotmail , Yahoo! Mail,...

# Giao thức POP3

## giai đoạn cấp phép

- ❑ các lệnh phía client:
  - **user**: khai báo username
  - **pass**: password
- ❑ các đáp ứng phía server
  - **+OK**
  - **-ERR**

## giai đoạn giao dịch, client:

- ❑ **list**: liệt kê các số thông điệp
- ❑ **retr**: trích xuất thông điệp theo số
- ❑ **dele**: xóa
- ❑ **quit**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

# POP3 và IMAP

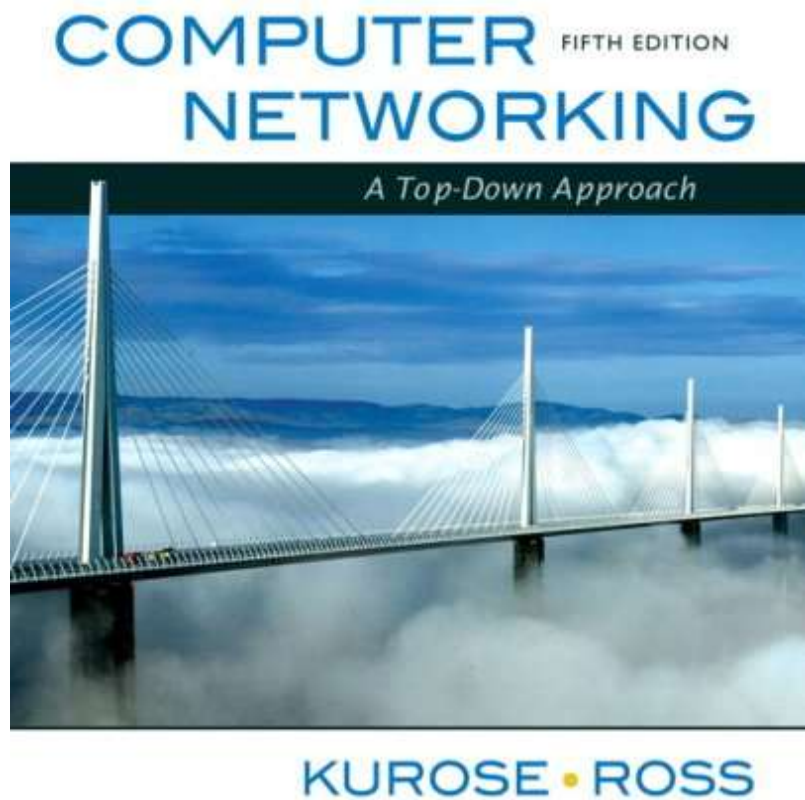
## nghiên cứu thêm về POP3

- ❑ Ví dụ trước dùng chế độ "tải xuống và xóa".
- ❑ Bob không thể đọc lại email nếu thay đổi client
- ❑ "tải xuống-và-giữ": sao chép các thông điệp trên các client khác nhau
- ❑ POP3 không giữ trạng thái của các phiên làm việc

## IMAP

- ❑ Giữ tất cả thông điệp tại 1 vị trí: server
- ❑ Cho phép user tổ chức các thông điệp theo dạng thư mục
- ❑ IMAP giữ trạng thái xuyên suốt các phiên làm việc:
  - các tên của thư mục và ánh xạ giữa ID của thông điệp và tên thư mục

## 2.6 DNS - Domain Name System





# DNS: Domain Name System

**Con người:** nhiều cách nhận dạng:

- tên, #hộ chiếu

**Internet hosts, routers:**

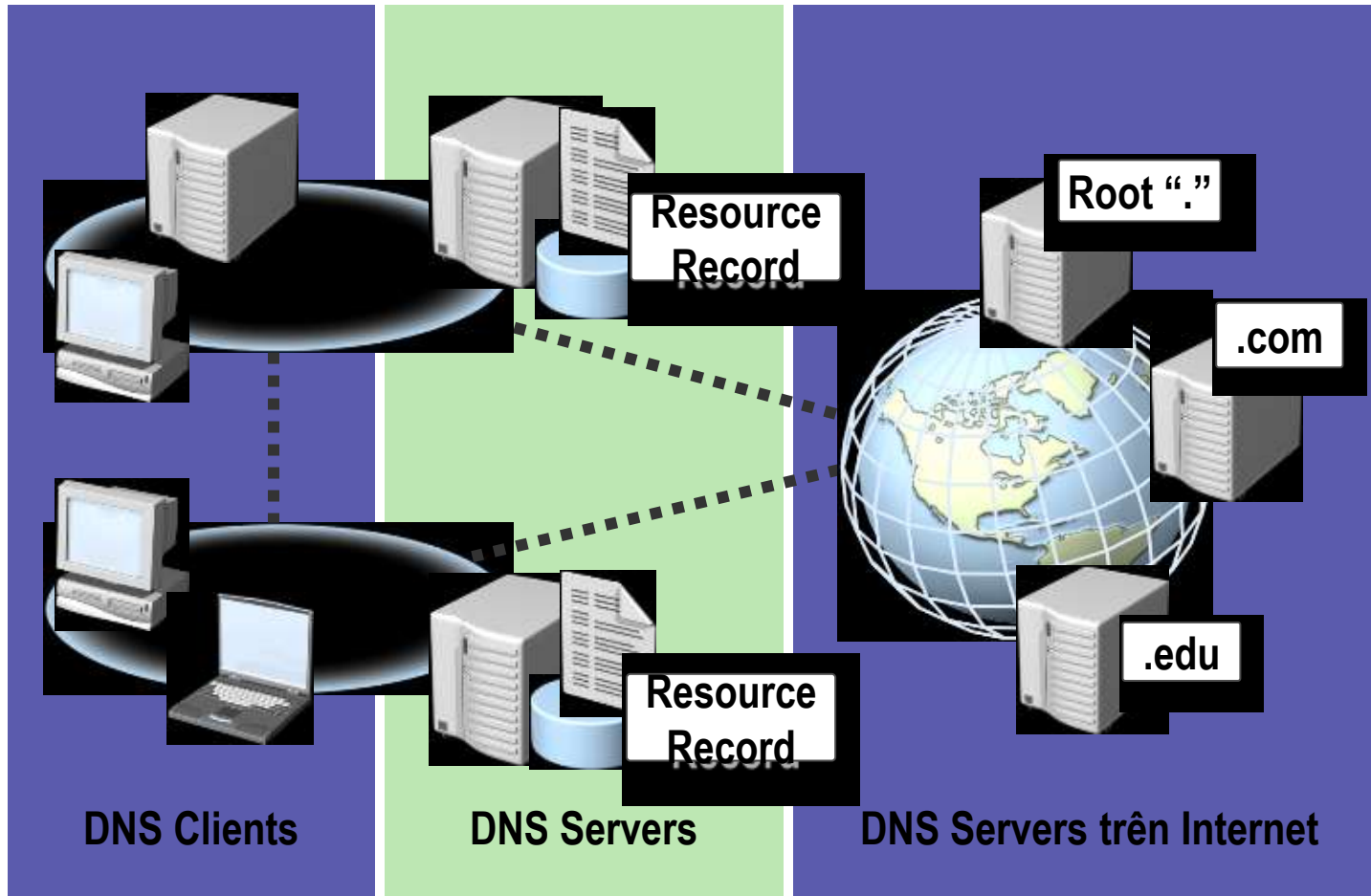
- địa chỉ IP (32 bit) dùng cho các gói định địa chỉ
- "tên", ví dụ:  
www.yahoo.com - dùng bởi con người

Ánh xạ giữa địa chỉ IP và tên?

**Domain Name System:**

- ❑  *cơ sở dữ liệu phân bố* hiện thực theo tổ chức phân cấp của nhiều *servers*
- ❑ *Là giao thức lớp application*

# Các thành phần trong dịch vụ DNS



## Các dịch vụ DNS

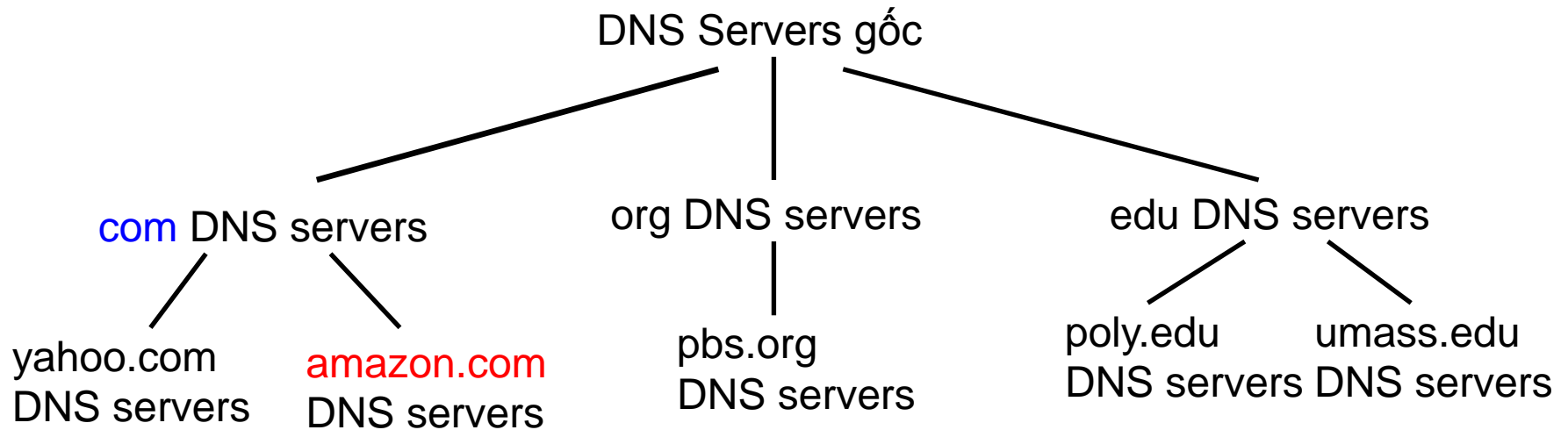
- ❑ Tên Host chuyển thành địa chỉ IP
- ❑ Bí danh Host
  - các tên đúng chuẩn và bí danh
- ❑ Bí danh Mail server
- ❑ Tải phân bố
  - Các Web server bản sao: tập các địa chỉ IP cho 1 tên đúng chuẩn

## Tại sao không tập trung hóa DNS?

- ❑ một điểm chịu lỗi
- ❑ lưu lượng
- ❑ khoảng cách cơ sở dữ liệu tập trung
- ❑ bảo trì

không *linh hoạt!*

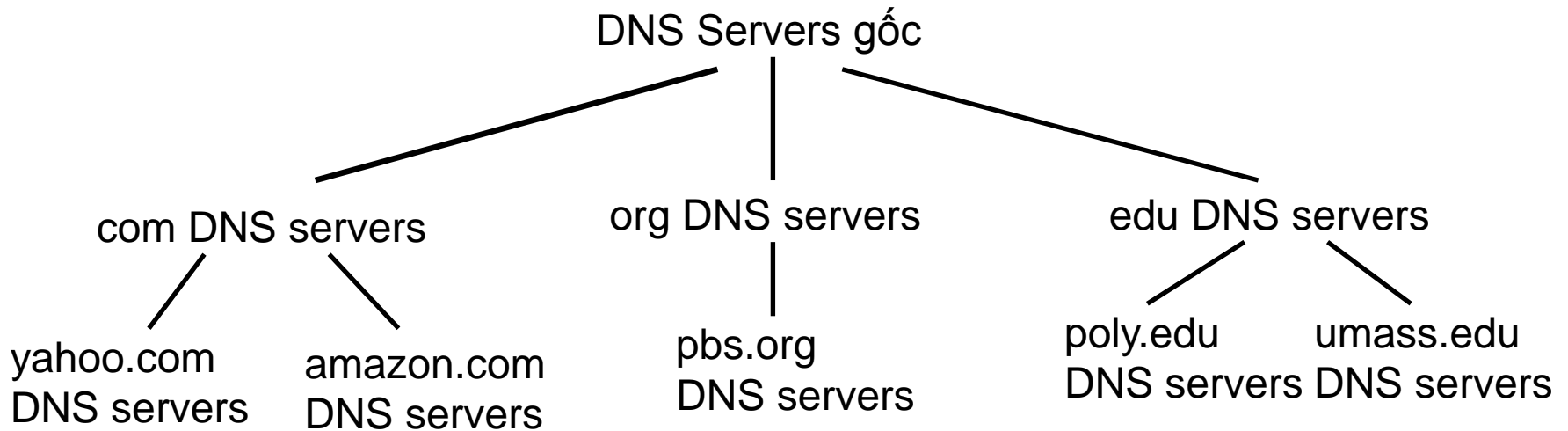
# Cơ sở dữ liệu cấu trúc, phân bố



## Client muốn IP cho [www.amazon.com](http://www.amazon.com):

- ❑ Client hỏi một server gốc (root) để tìm **com** DNS server
- ❑ Client hỏi com DNS server để lấy **amazon.com** DNS server
- ❑ Client hỏi amazon.com DNS server để lấy địa chỉ IP của **www.amazon.com**

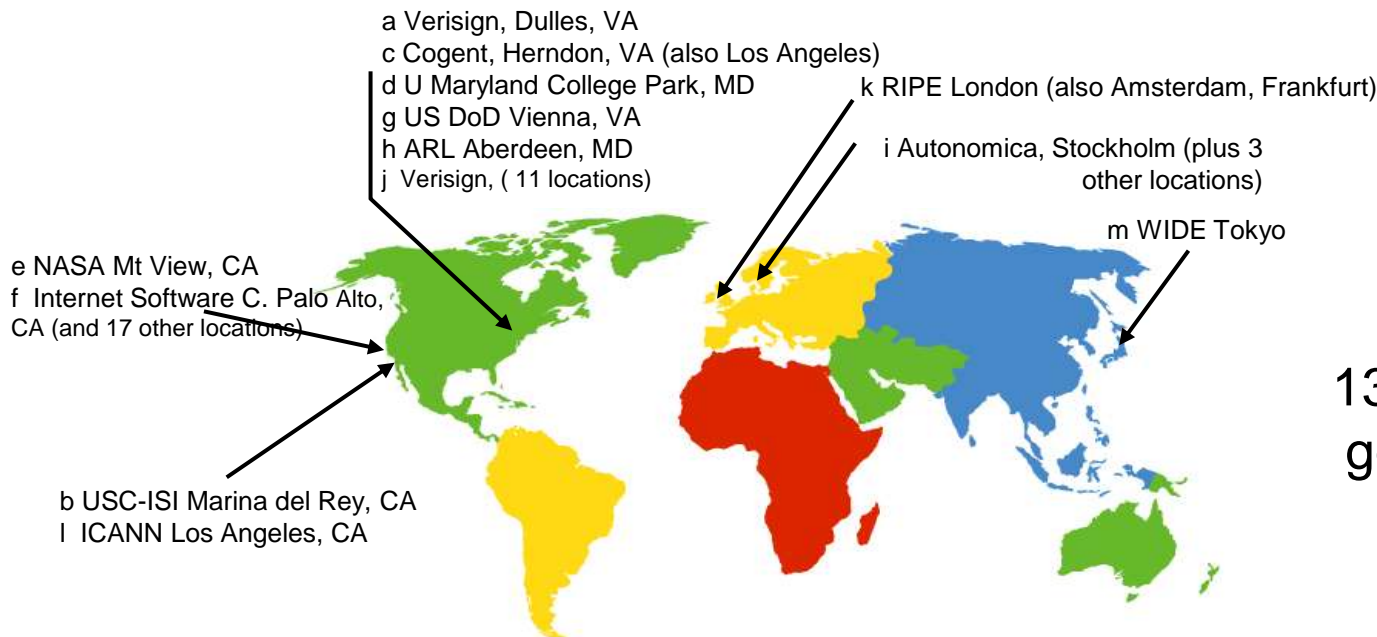
# Authoritative DNS server & Non-authoritative DNS server



- ❑ Authoritative DNS server (**DNS server có thẩm quyền**) sẽ thực hiện:
  - Kiểm tra cache, kiểm tra zone, gởi IP address cho truy vấn.
- ❑ Non-authoritative DNS server sẽ thực hiện:
  - Chuyển yêu cầu truy vấn không thể phân giải đến Forwarder server.
  - Sử dụng root hints server để trả lời cho truy vấn.

# DNS: các server tên gốc

- ❑ Local name server sẽ hỏi Root name server khi không xác định được ảnh xạ.
- ❑ Root name server:
  - Hỏi authoritative name server nếu không trả lời được
  - Nhận câu trả lời từ authoritative name server
  - Trả lời local name server



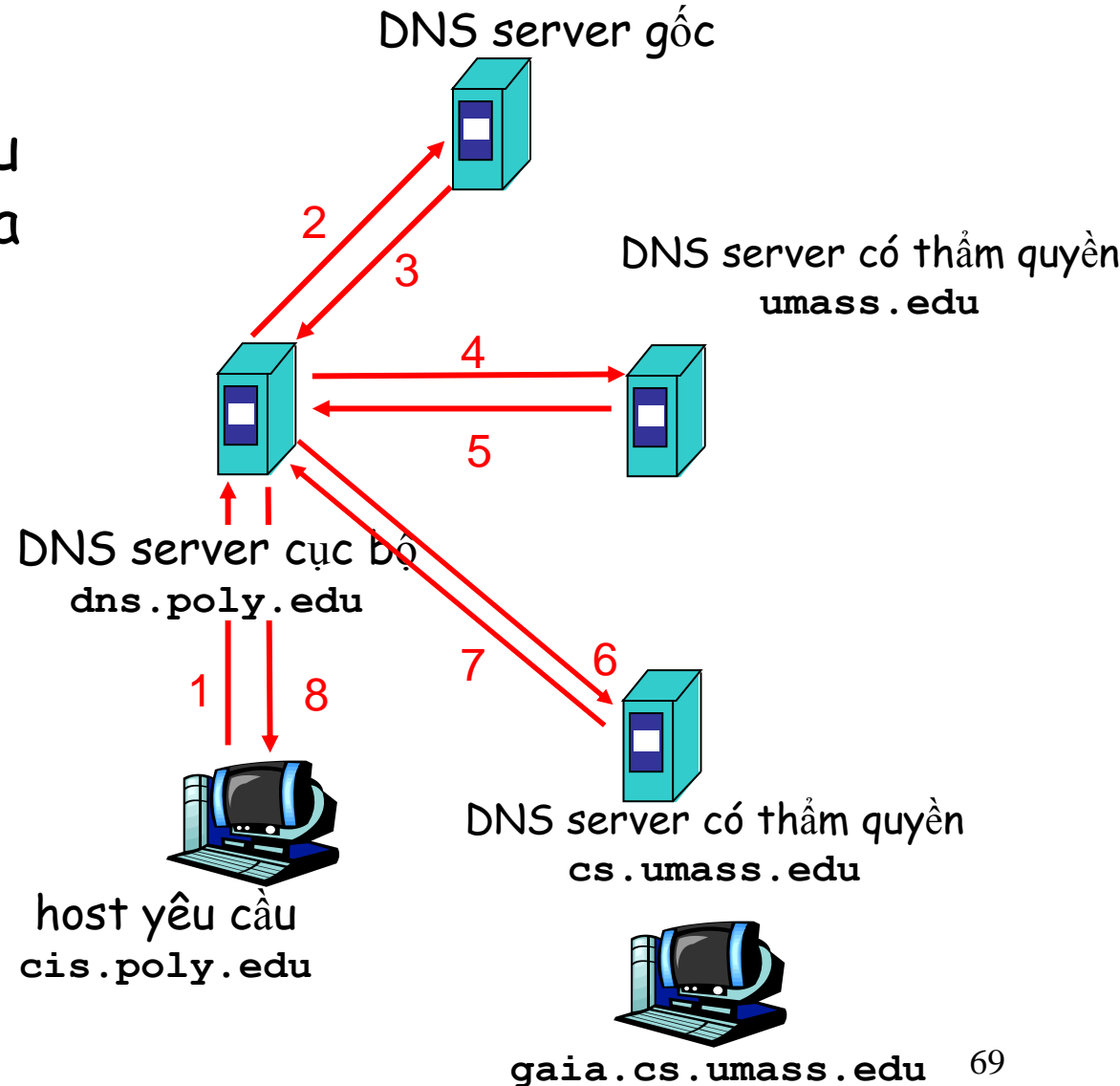
13 name servers  
gốc trên toàn cầu

# Server tên cục bộ

- ❑ Không hoàn toàn theo cấu trúc phân cấp
- ❑ Mỗi ISP (ISP cá nhân, công ty, trường học) có một server cục bộ như vậy.
  - cũng gọi là "server tên mặc nhiên"
- ❑ Khi một host tạo một truy vấn DNS, truy vấn đó được gửi tới DNS server cục bộ của nó
  - Hoạt động như một proxy, chuyển truy vấn vào cho tổ chức phân cấp

# Ví dụ

- Host tại `cis.poly.edu` muốn địa chỉ IP của `gaia.cs.umass.edu`

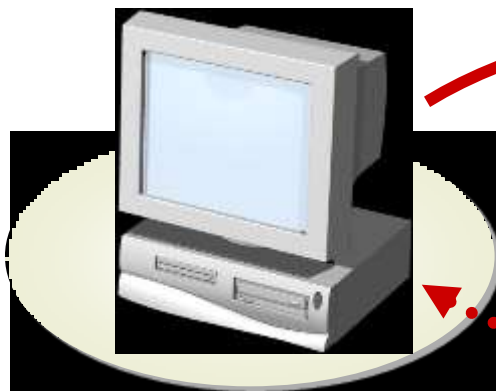




# Recursive query

*recursive query* được gửi tới DNS server, trong đó DNS client yêu cầu DNS server cung cấp đầy đủ thông tin cho truy vấn.

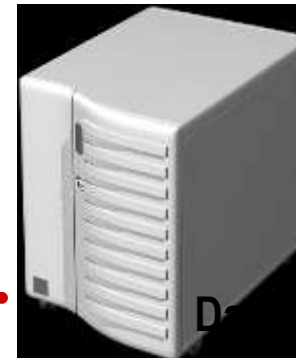
DNS server kiểm tra forward lookup zone và cache để trả lời truy vấn



Computer1

Recursive query for  
mail1.yahoo.com

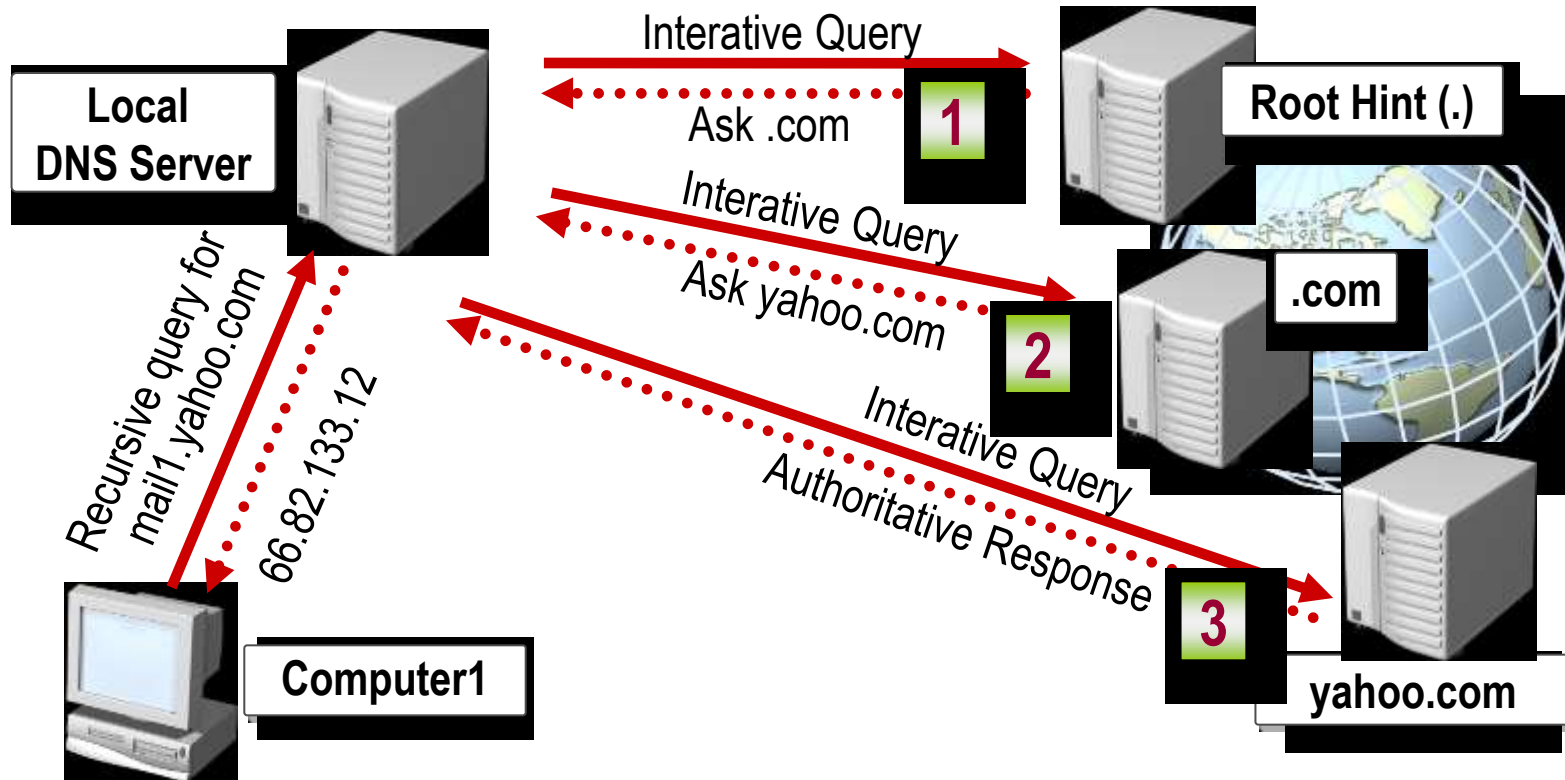
66.8.133.10



Local DNS Server

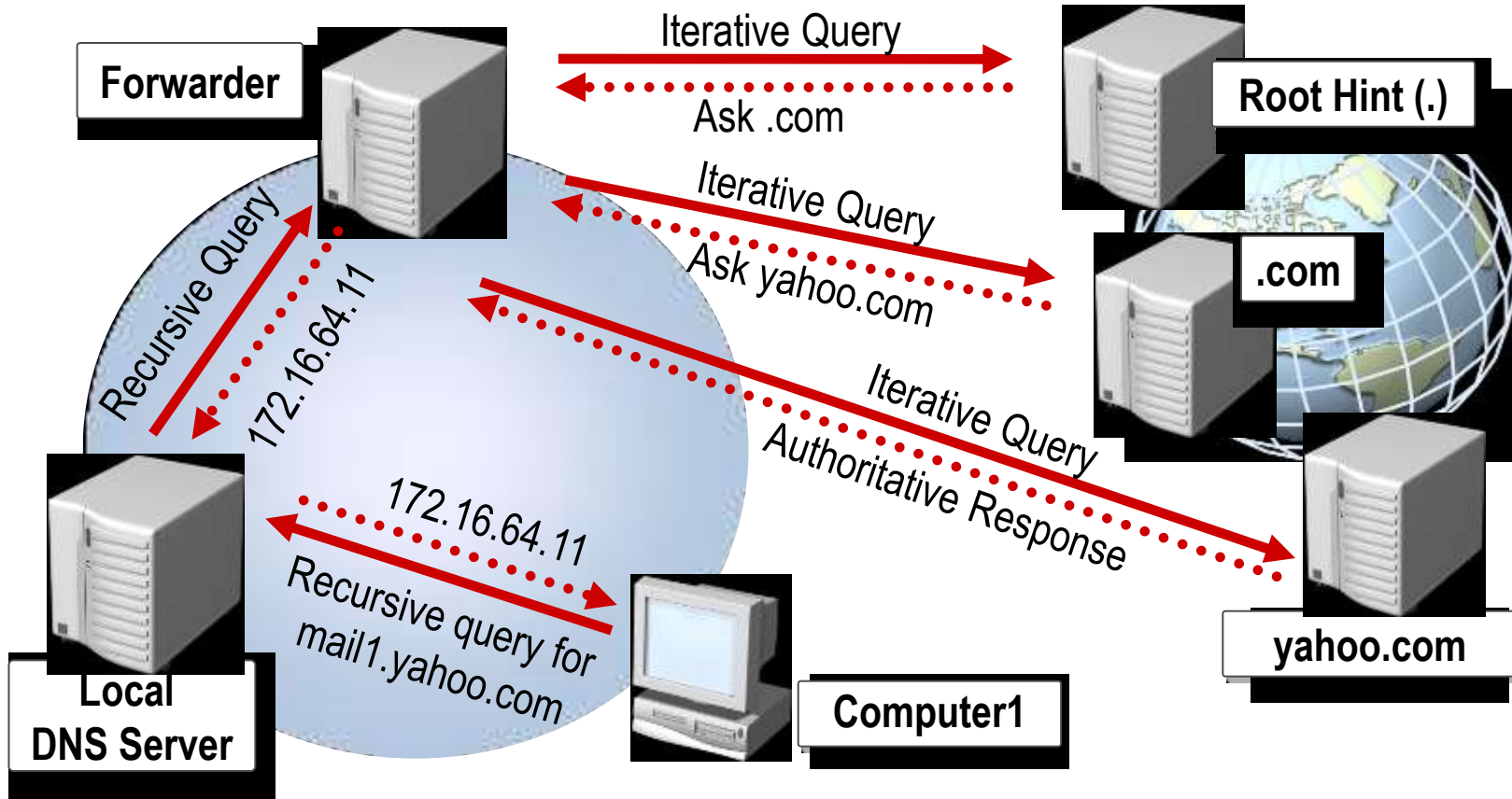
# Interactive query

*iterative* query là truy vấn được gửi tới DNS server trong đó DNS client yêu cầu DNS server cung cấp thông tin tốt nhất mà nó có chứ không tìm sự trợ giúp từ Name server khác. ACK của *iterative* query thường tham chiếu đến DNS server con trong DNS tree.



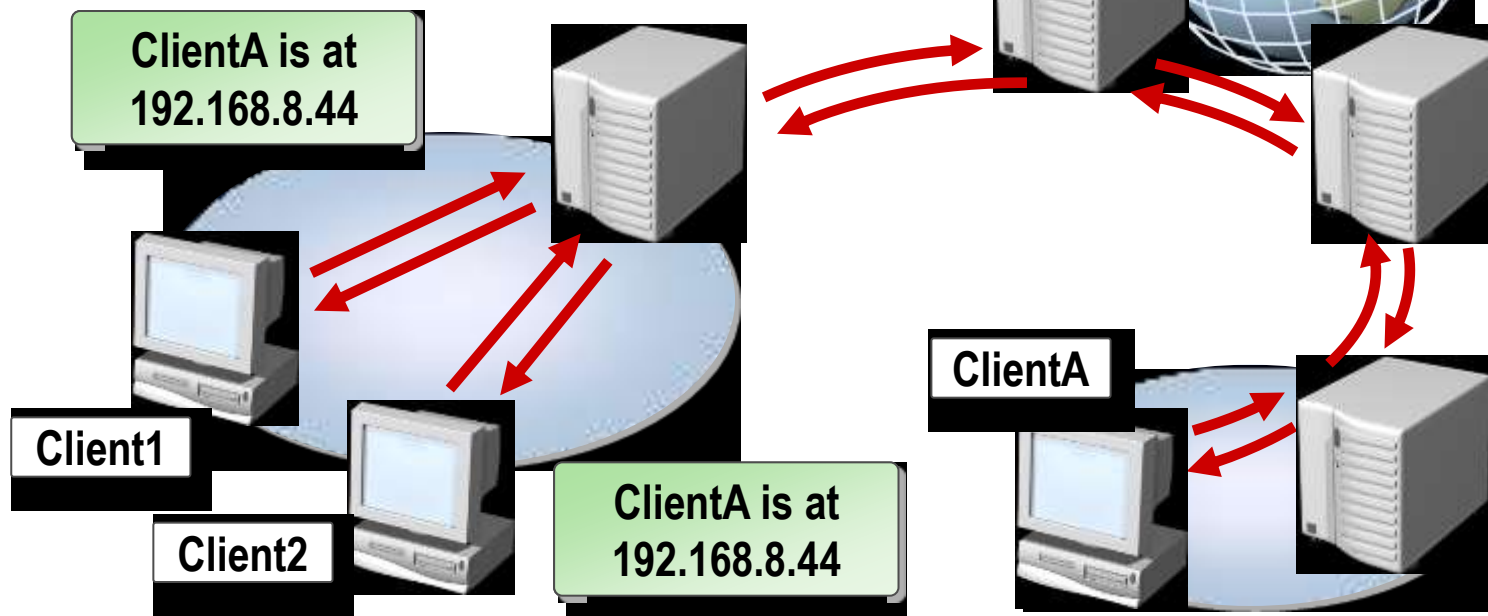
# Forwarder

*forwarder* cung cấp cơ chế chuyển yêu cầu truy vấn cho internal DNS servers ra ngoài external DNS server



# Caching DNS Server

Caching Table		
Host Name	IP Address	TTL
clientA.vnn.vn.	192.168.8.44	28 seconds



**Caching** là tiến trình lưu trữ tạm một số thông tin phân giải trước để cung cấp cho các lần phân giải sau này nhằm làm tăng tốc quá trình phân giải tên miền

# Caching DNS Server

- Khi “học” được thêm một ánh xạ, name server sẽ “ghi nhớ” ánh xạ này
  - Sau một khoảng thời gian, nếu thành phần nào trong cache không được sử dụng thì sẽ bị xóa bỏ.
- Cơ chế Cập nhật và Thông báo do IETF thiết kế:
  - RFC 2136
  - <http://www.ietf.org/html.charters/dnsind-charter.html>

# Các DNS record

DNS: cơ sở dữ liệu phân bố lưu trữ các record tài nguyên (RR)

dạng thức RR: (name, value, type, ttl)

## □ Type=A

- ❖ **name** là tên host
- ❖ **value** là địa chỉ IP

## □ Type=NS

- **name** là tên miền (vd: foo.com)
- **value** là tên host của server tên có thẩm quyền cho tên miền này

## □ Type=CNAME

- ❖ **name** là bí danh của tên “chuẩn” (tên thực)

www.ibm.com là tên thực

servereast.backup2.ibm.com

- ❖ **value** là tên chuẩn

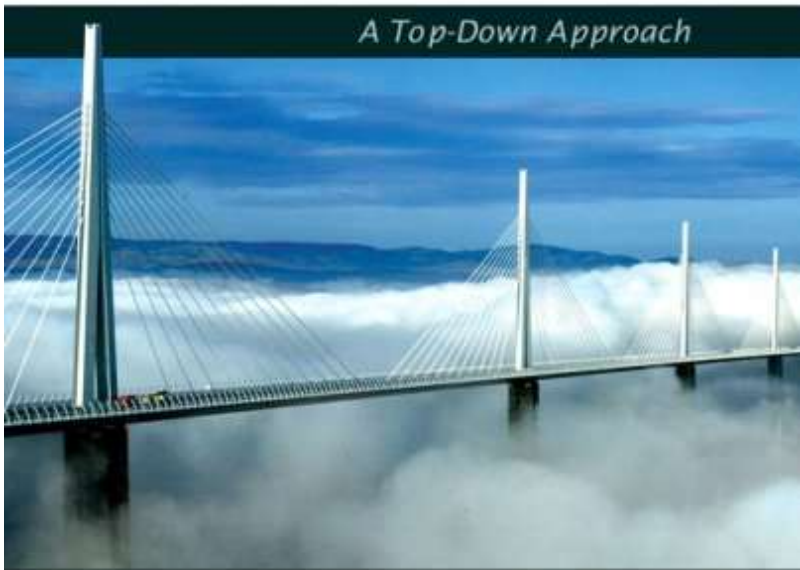
## □ Type=MX

- ❖ **value** là tên của email server liên kết với **name**

## 2.7 Lập trình socket với TCP

### COMPUTER FIFTH EDITION NETWORKING

*A Top-Down Approach*



KUROSE • ROSS



# Lập trình socket

Mục tiêu: nghiên cứu cách xây dựng ứng dụng truyền thông client/server dùng sockets

## Socket API

- ❑ đã được giới thiệu trong BSD4.1 UNIX, 1981
- ❑ sử dụng và giải phóng bởi ứng dụng
- ❑ mô hình client/server
- ❑ 2 kiểu dịch vụ lưu thông qua socket API:
  - datagram không tin cậy
  - tin cậy, truyền byte theo streaming

## socket

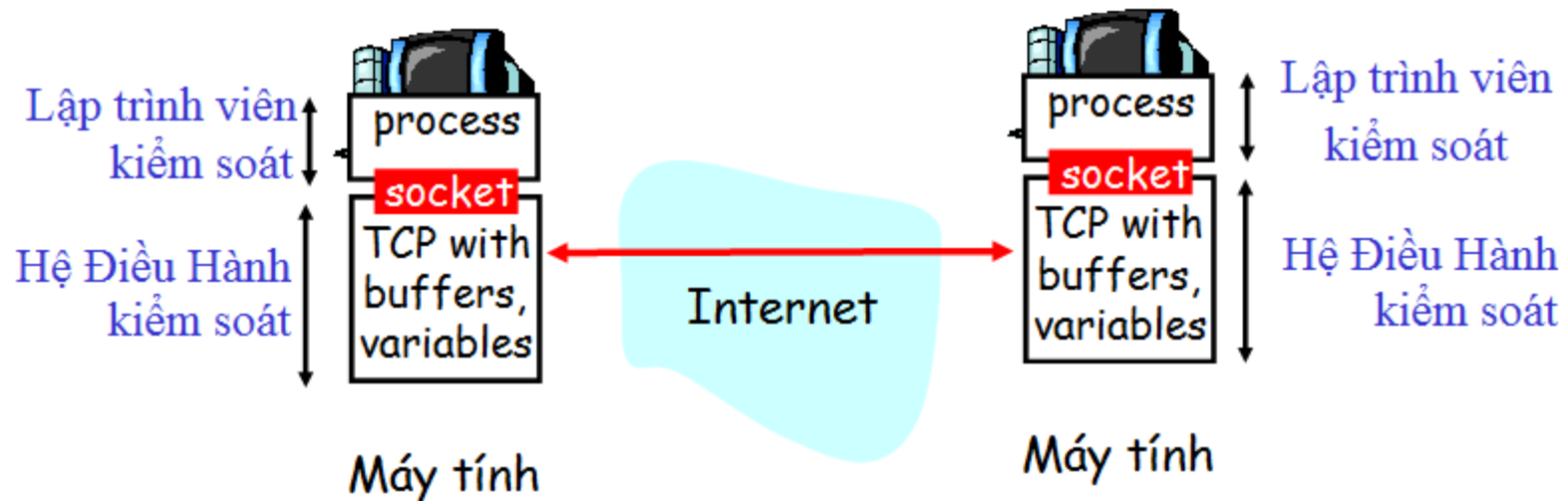
Là giao diện *nằm trên máy tính*, do ứng dụng tạo ra và quản lý, nhưng HĐH kiểm soát (là "cửa") thông qua đó tiến trình vừa gửi và nhận thông điệp từ các tiến trình ứng dụng khác (ở trên máy tính khác)



# Lập trình socket dùng TCP

**Socket:** một cánh cửa giữa tiến trình ứng dụng và giao thức transport (UCP hoặc TCP)

(Socket: a door between application process and end-end-transport protocol (UCP or TCP).)



**Dịch vụ TCP:** truyền tin cậy các **bytes** từ một tiến trình đến tiến trình khác

# Lập trình socket *với TCP*

Client phải liên lạc trước với server

- ❑ tiến trình server phải chạy trước
- ❑ server phải tạo socket (cửa) mời client đến tiếp xúc

Client liên lạc server bằng:

- ❑ tạo socket TCP client cục bộ
- ❑ xác định địa chỉ IP, số port của tiến trình server
- ❑ Khi **client tạo socket**: client TCP thiết lập kết nối với server TCP

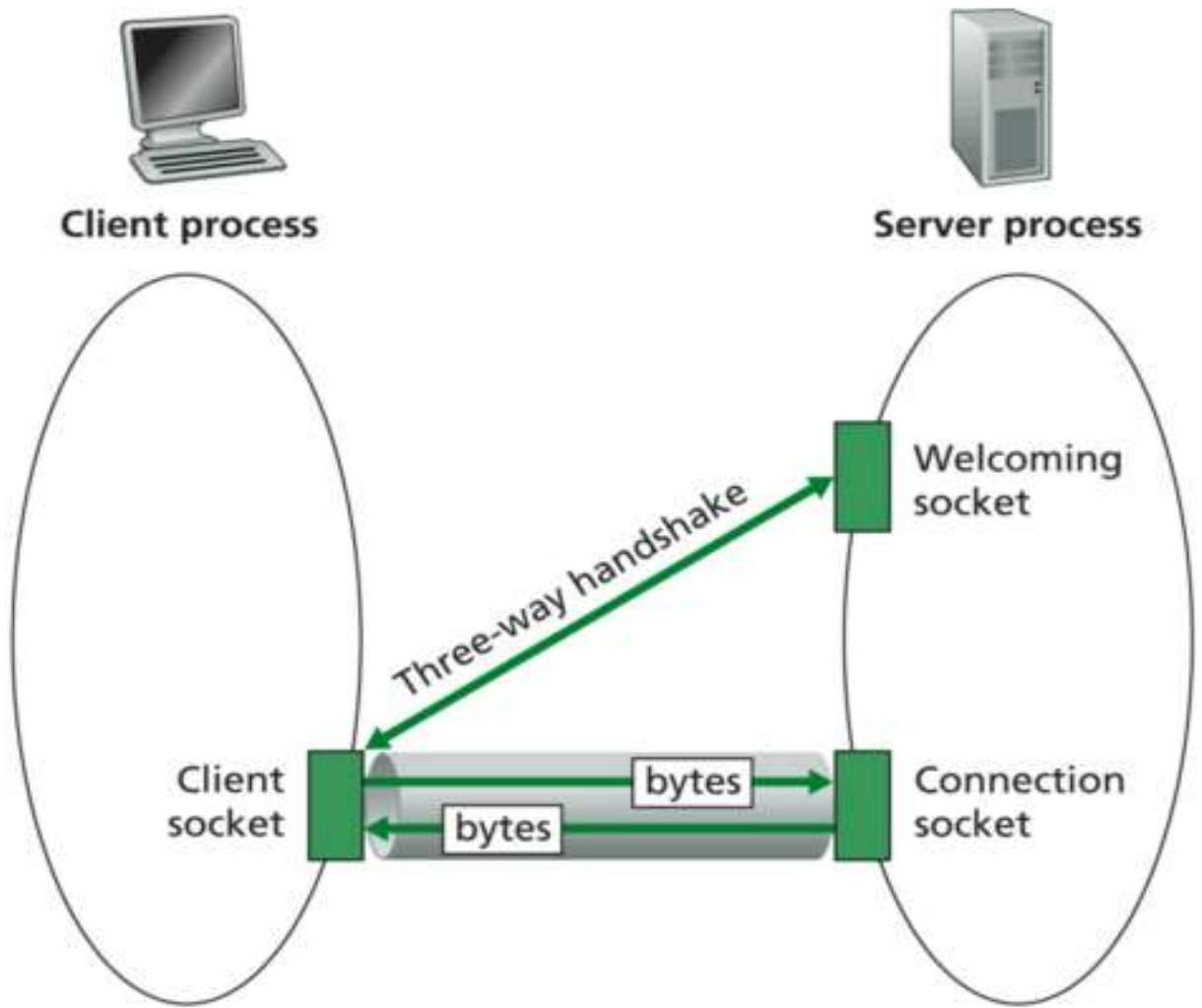
- ❑ Khi đã chấp nhận yêu cầu bởi client, **server TCP tạo socket mới** cho tiến trình server để truyền thông với client

- cho phép server "nói chuyện" với nhiều client
- số port dùng để phân biệt các client (xem trong chương 3)

Nhìn dưới góc độ ứng dụng

*TCP cung cấp việc truyền các byte tin cậy và theo thứ tự giữa client và server*

# Giao tiếp socket Client/server



# Giao tiếp socket Client/server: TCP

Server (chạy trên `hostid`)

Client

create socket,  
port=`x`, for  
incoming request:  
`welcomeSocket =  
ServerSocket()`

wait for incoming  
connection request  
`connectionSocket =  
welcomeSocket.accept()`

read request from  
`connectionSocket`

write reply to  
`connectionSocket`

....

TCP  
connection setup

create socket,  
connect to `hostid`, port=`x`  
`clientSocket =  
Socket()`

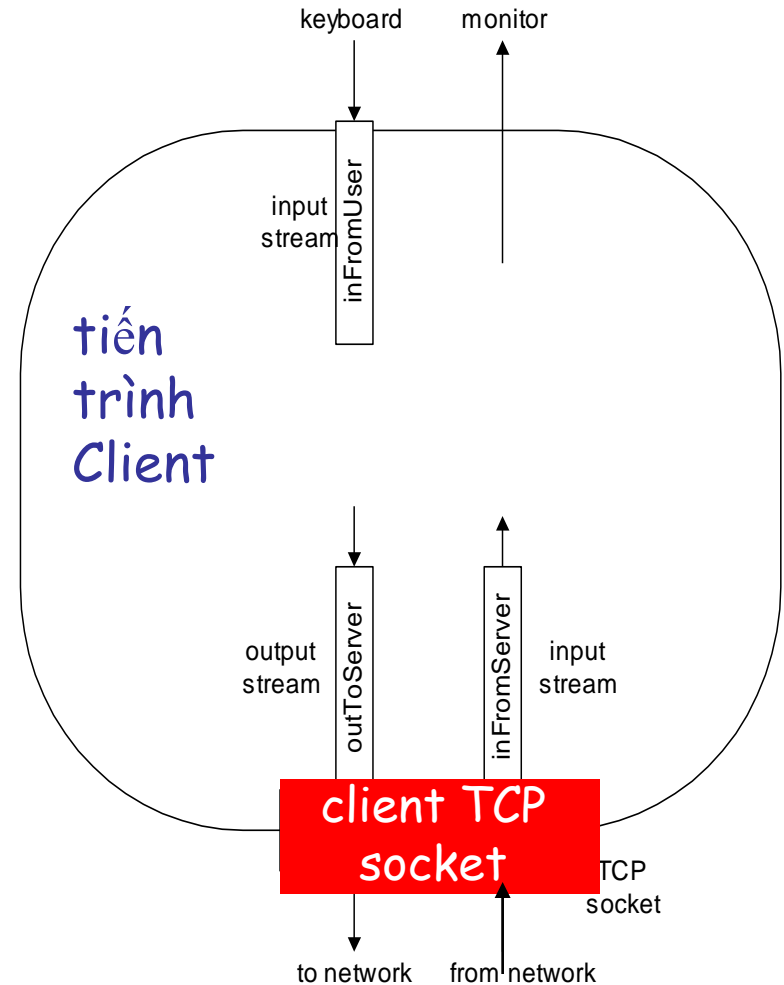
send request using  
`clientSocket`

read reply from  
`clientSocket`

close  
`clientSocket`

# Thuật ngữ Stream

- ❑ Một **stream** là một chuỗi các ký tự được "chảy" vào hoặc ra khỏi một tiến trình
- ❑ Một **input stream** để chỉ nguồn vào của một tiến trình, vd: bàn phím hoặc socket.
- ❑ Một **output stream** để chỉ nguồn ra của một tiến trình, vd: màn hình hoặc socket.



# Lập trình socket dùng TCP

## Ví dụ ứng dụng client-server :

- 1) client đọc các dòng từ input chuẩn (`inFromUser stream`) , gửi đến server thông qua socket (`outToServer stream`)
- 2) server đọc các dòng từ socket
- 3) server chuyển các dòng thành chữ hoa, gửi ngược trở lại cho client
- 4) client đọc, in các dòng đã sửa đổi từ socket (`inFromServer stream`)

# Ví dụ: Java client (TCP)

```
import java.io.*;
import java.net.*;
class TCPClient {
```

```
    public static void main(String argv[]) throws Exception
    {
```

```
        String sentence;
        String modifiedSentence;
```

input stream } tạo → BufferedReader **inFromUser** =  
new BufferedReader(new InputStreamReader(System.in));

client socket, } tạo  
kết nối vào server → Socket **clientSocket** = new Socket("hostname", 6789);

output stream } tạo  
gắn vào socket → DataOutputStream **outToServer** =  
new DataOutputStream(clientSocket.getOutputStream());

# Ví dụ: Java client (TCP)

tạo  
input stream  
gắn vào socket

```
BufferedReader inFromServer =  
    new BufferedReader(new  
        InputStreamReader(clientSocket.getInputStream()));
```

Gửi dòng  
đến server

```
sentence = inFromUser.readLine();
```

```
outToServer.writeBytes(sentence + '\n');
```

đọc dòng  
từ server

```
modifiedSentence = inFromServer.readLine();
```

```
System.out.println("FROM SERVER: " + modifiedSentence);
```

```
clientSocket.close();
```

```
    }  
}
```



# Ví dụ: Java server (TCP)

```
import java.io.*;
import java.net.*;
```

```
class TCPServer {
```

```
    public static void main(String argv[]) throws Exception
    {
```

```
        String clientSentence;
        String capitalizedSentence;
```

socket mời tiếp xúc  
tại port 6789

← tạo

```
        ServerSocket welcomeSocket = new ServerSocket(6789);
```

Chờ, client  
tiếp cận với  
server

```
        while(true) {
```

```
            Socket connectionSocket = welcomeSocket.accept();
```

Tạo input  
stream, gắn vào  
socket

```
            BufferedReader inFromClient = new BufferedReader(new
                InputStreamReader(connectionSocket.getInputStream()));
```

# Ví dụ: Java server (TCP)

Tạo output stream, gắn vào socket

```
DataOutputStream outToClient =  
    new DataOutputStream(connectionSocket.getOutputStream());
```

Đọc dòng từ socket

```
clientSentence = inFromClient.readLine();
```

```
capitalizedSentence = clientSentence.toUpperCase() + '\n';
```

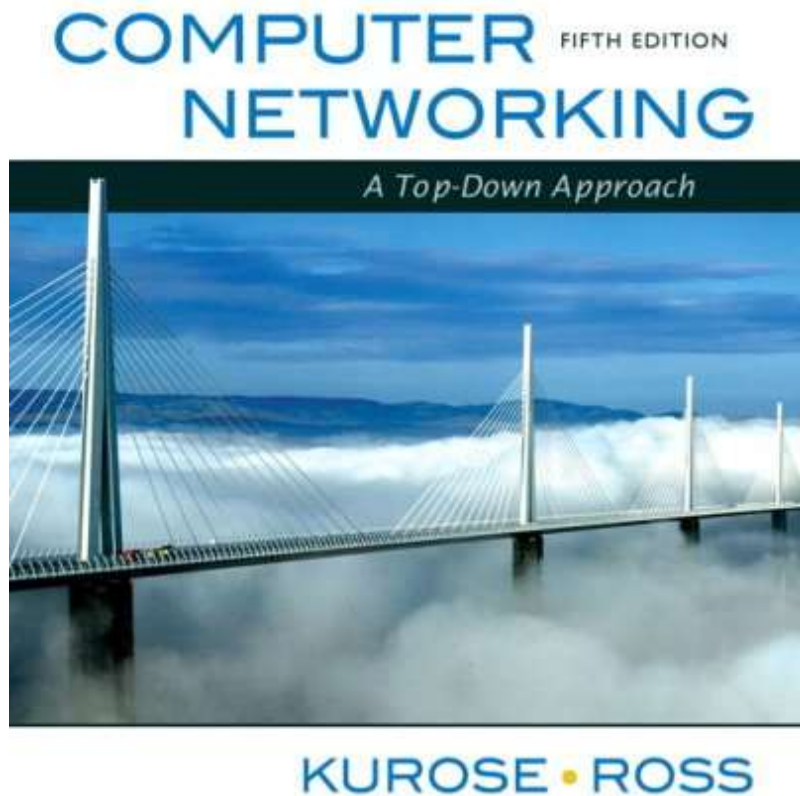
Viết dòng ra từ socket

```
outToClient.writeBytes(capitalizedSentence);
```

```
}  
}  
}
```

kết thúc vòng lặp while quay lại và chờ cho kết nối của client khác

## 2.8 Lập trình socket với UDP



# Lập trình socket *với UDP*

UDP: không "kết nối" giữa client và server

- ❑ không bắt tay
- ❑ người gửi rõ ràng gắn địa chỉ IP và port của đích vào mỗi gói
- ❑ phải trích địa chỉ IP, port của người gửi từ gói đã nhận

UDP: dữ liệu truyền có thể không theo thứ tự, hoặc bị mất mát

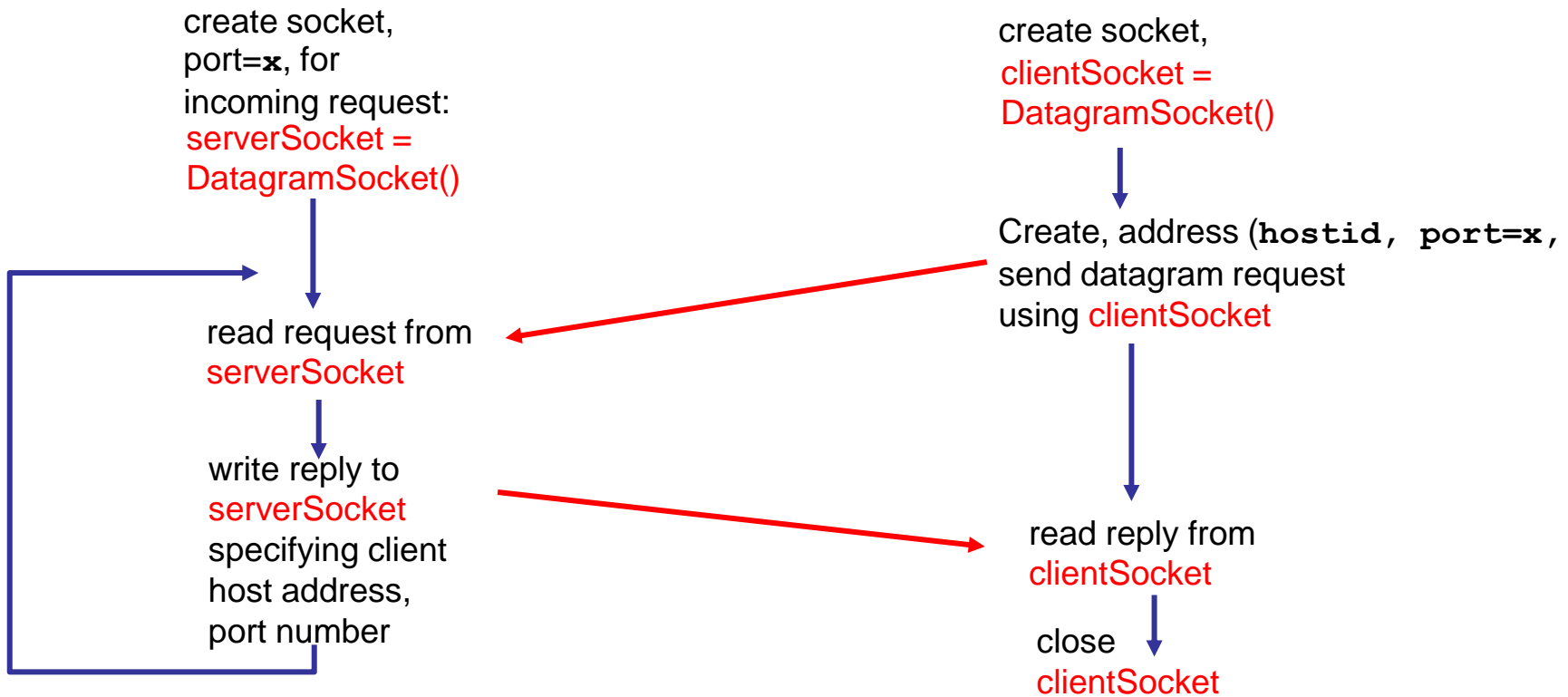
*góc nhìn ứng dụng*

*UDP cung cấp việc truyền không tin cậy một nhóm các byte ("datagrams") giữa client và server*

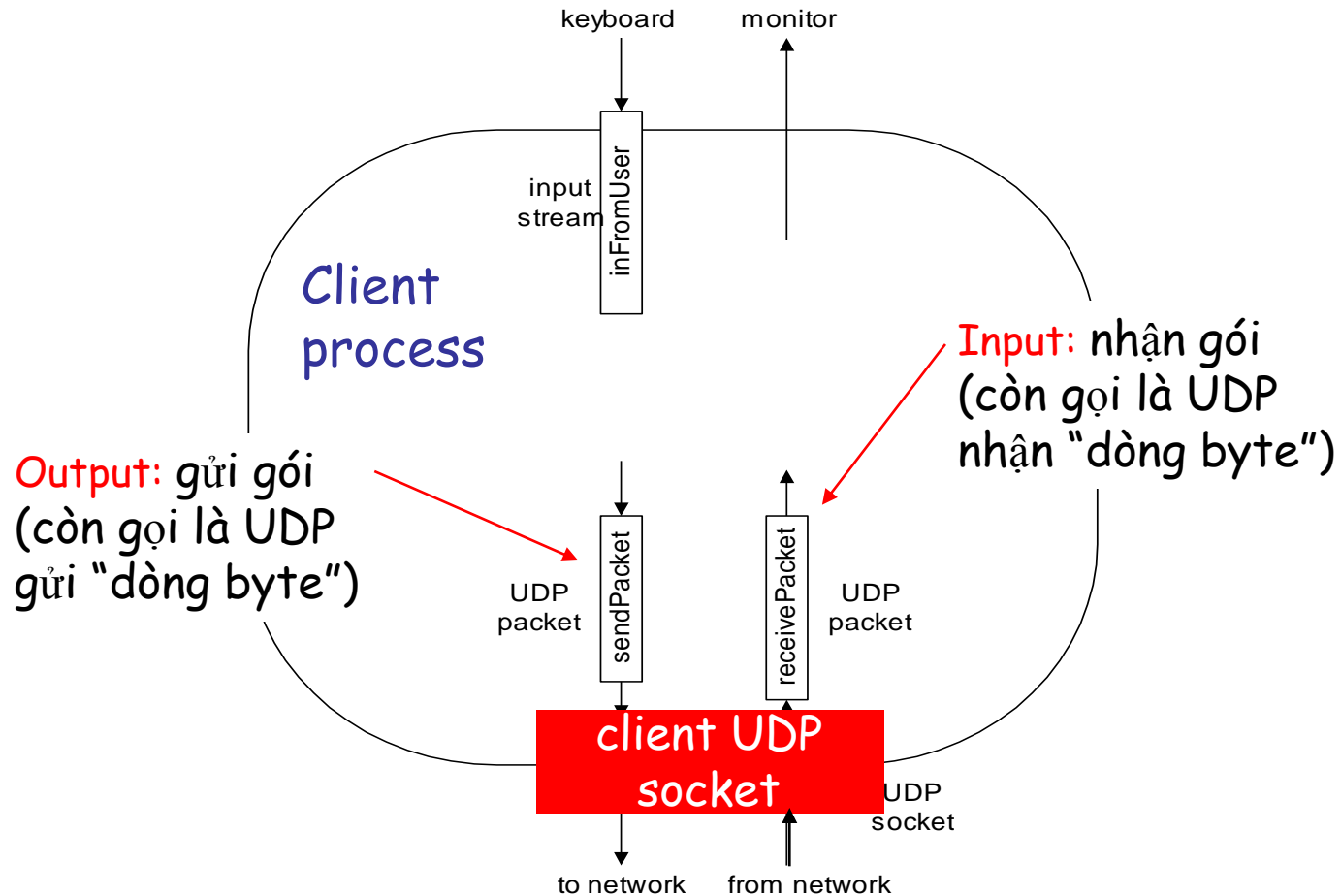
# Giao tiếp socket Client/server: UDP

## Server (chạy trên `hostid`)

## Client



# Ví dụ: Java client (UDP)



# Ví dụ: Java client (UDP)

```
import java.io.*;
import java.net.*;
```

```
class UDPClient {
    public static void main(String args[]) throws Exception
    {
```

input stream <sup>tao</sup> → BufferedReader inFromUser =  
new BufferedReader(new InputStreamReader(System.in));

client socket <sup>tao</sup> → DatagramSocket clientSocket = new DatagramSocket();

hostname thành <sup>dịch</sup> địa chỉ IP dùng DNS → InetAddress IPAddress = InetAddress.getByName("hostname");

```
byte[] sendData = new byte[1024];
byte[] receiveData = new byte[1024];
```

```
String sentence = inFromUser.readLine();
sendData = sentence.getBytes();
```

# Ví dụ: Java client (UDP).

tạo datagram với  
dữ liệu để gửi,  
độ dài, địa chỉ IP, port → 9876);

```
DatagramPacket sendPacket =  
    new DatagramPacket(sendData, sendData.length, IPAddress,  
                        9876);
```

gửi datagram  
đến server →

```
clientSocket.send(sendPacket);
```

đọc datagram  
từ server →

```
DatagramPacket receivePacket =  
    new DatagramPacket(receiveData, receiveData.length);  
  
clientSocket.receive(receivePacket);  
  
String modifiedSentence =  
    new String(receivePacket.getData());  
  
System.out.println("FROM SERVER:" + modifiedSentence);  
clientSocket.close();  
}  
}
```



# Ví dụ: Java server (UDP)

```
import java.io.*;  
import java.net.*;
```

```
class UDPServer {  
    public static void main(String args[]) throws Exception  
    {
```

Tạo  
datagram socket  
tại port 9876

```
        DatagramSocket serverSocket = new DatagramSocket(9876);
```

```
        byte[] receiveData = new byte[1024];  
        byte[] sendData = new byte[1024];
```

```
        while(true)  
        {
```

Tạo không gian để  
nhận datagram

```
            DatagramPacket receivePacket =  
                new DatagramPacket(receiveData, receiveData.length);
```

nhận  
datagram

```
            serverSocket.receive(receivePacket);
```

# Ví dụ: Java server (UDP)

```
String sentence = new String(receivePacket.getData());
```

lấy địa chỉ IP  
port #, của  
người gửi

```
InetAddress IPAddress = receivePacket.getAddress();
```

```
int port = receivePacket.getPort();
```

```
String capitalizedSentence = sentence.toUpperCase();
```

```
sendData = capitalizedSentence.getBytes();
```

tạo datagram  
để gửi tới client

```
DatagramPacket sendPacket =  
    new DatagramPacket(sendData, sendData.length, IPAddress,  
        port);
```

viết  
datagram  
vào socket

```
serverSocket.send(sendPacket);
```

```
}
```

```
}
```

```
}
```

kết thúc vòng lặp while,  
quay lại và chờ  
datagram khác

# Bài tập lập trình mạng

- ❑ Bài 1: Hãy lập trình socket TCP & UDP thực hiện:
  - Client: Nhập một mảng số nguyên và gửi đến Server
  - Server: Nhận mảng số nguyên và in ra màn hình. Sau đó sắp xếp theo thứ tự tăng dần rồi gửi lại cho client.
  - Client nhận mảng số nguyên sau khi sắp xếp và in ra màn hình cho người sử dụng.
- ❑ Bài 2: Sửa lại chương trình trên cho phép người dùng có thể lặp lại quá trình gửi dãy số nguyên đến Server.
- ❑ Bài 3: Viết chương trình cho phép 2 người chat với nhau (simpleChat)
- ❑ Bài 4: Viết chương trình cho phép nhiều người chat (MultiChat)



Hết Chương 2

COMPUTER FIFTH EDITION  
NETWORKING

*A Top-Down Approach*



KUROSE • ROSS

## Chương 3 TẦNG GIAO VẬN Transport Layer

Nhập môn mạng máy tính

# Chương 3:

## Mục tiêu:

□ hiểu các nguyên tắc đằng sau các dịch vụ :

- multiplexing/demultiplexing
- truyền dữ liệu tin cậy
- điều khiển lưu lượng
- điều khiển tắc nghẽn

□ nghiên cứu về các giao thức trên Internet:

- UDP: vận chuyển không kết nối (connectionless)
- TCP: vận chuyển hướng kết nối (connection-oriented)
- điều khiển tắc nghẽn TCP

# Chương 3: Nội dung trình bày

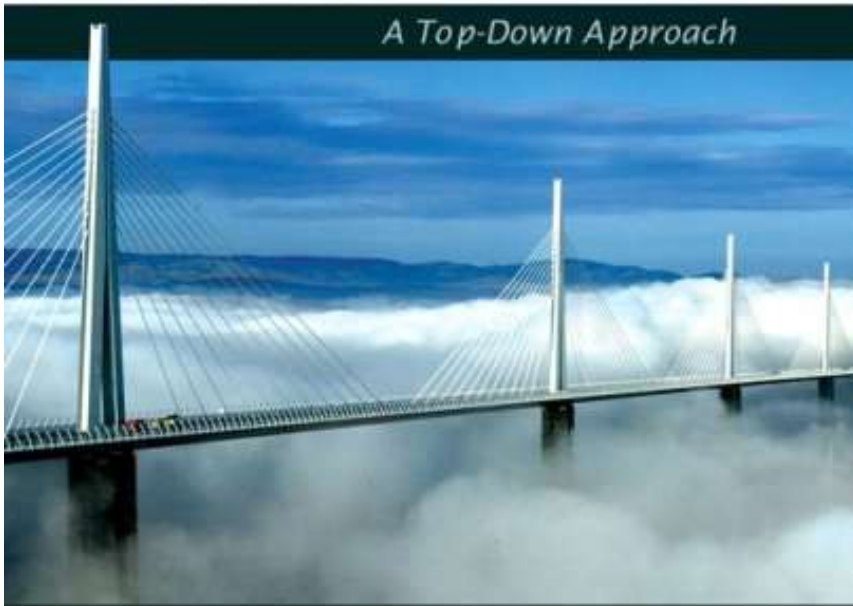
- 3.1 Các dịch vụ
- 3.2 Multiplexing và demultiplexing
- 3.3 Vận chuyển không kết nối: UDP
- 3.4 Các nguyên lý của việc truyền dữ liệu tin cậy
- 3.5 Vận chuyển hướng kết nối: TCP
  - cấu trúc phân đoạn
  - truyền dữ liệu tin cậy
  - điều khiển lưu lượng
  - quản lý kết nối
- 3.6 Các nguyên lý của điều khiển tắc nghẽn
- 3.7 Điều khiển tắc nghẽn TCP

## 3.1 Các dịch vụ

# COMPUTER NETWORKING

FIFTH EDITION

*A Top-Down Approach*

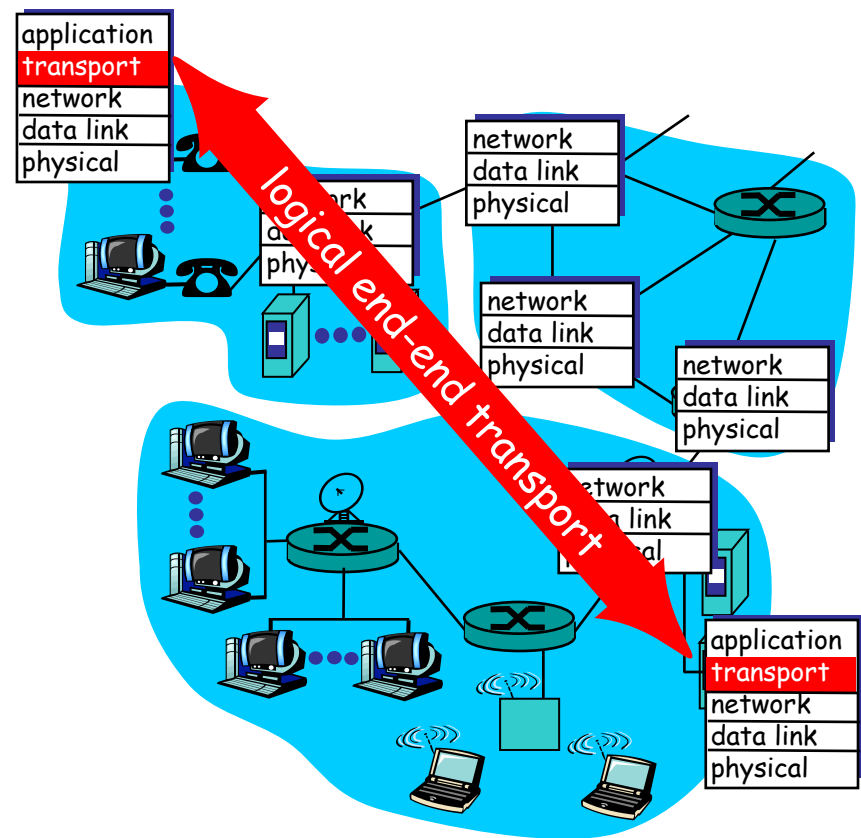


KUROSE • ROSS



# Các dịch vụ và giao thức Transport

- ❑ cung cấp *truyền thông logic* chạy trên các host khác nhau
- ❑ các giao thức transport chạy trên các hệ thống đầu cuối
  - phía gửi: cắt các thông điệp ứng dụng thành các **đoạn**, chuyển cho lớp network
  - phía nhận: tái kết hợp các đoạn thành các thông điệp, chuyển cho lớp application
- ❑ Tầng giao vận của Internet có 2 giao thức giao vận: TCP và UDP



# với lớp network

- ❑ *lớp network*: truyền thông logic giữa các host
- ❑ *Lớp transport*: truyền thông logic giữa các tiến trình
  - Thêm, phát triển dịch vụ mới từ dịch vụ tầng mạng

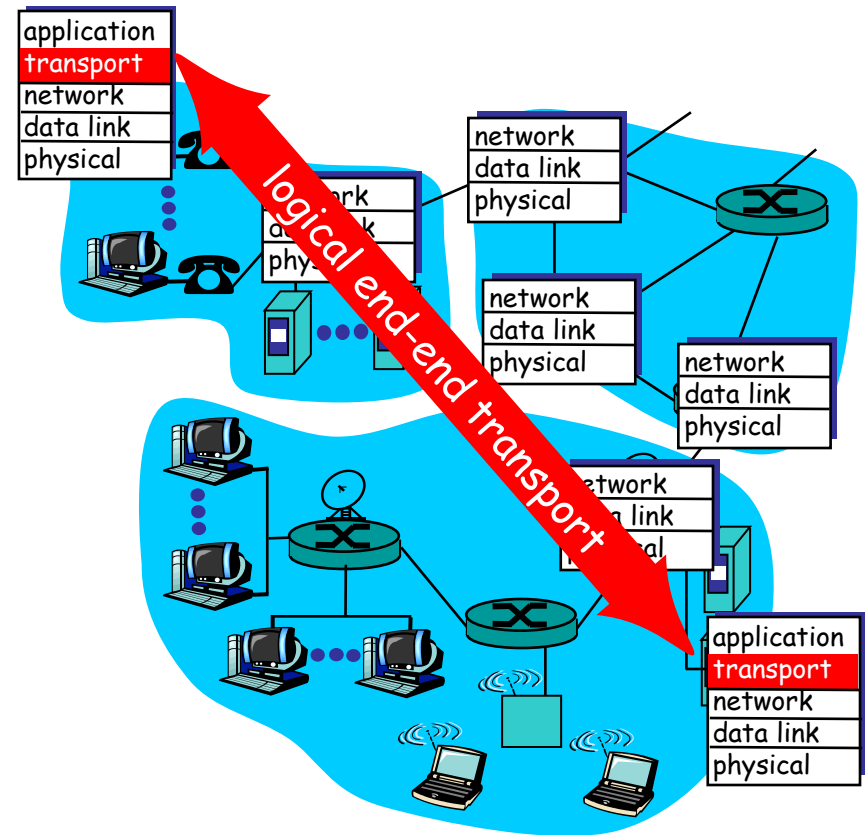
## Tình huống tự nhiên tương tự:

*12 đĩa trẻ gửi thư đến 12 đĩa trẻ khác*

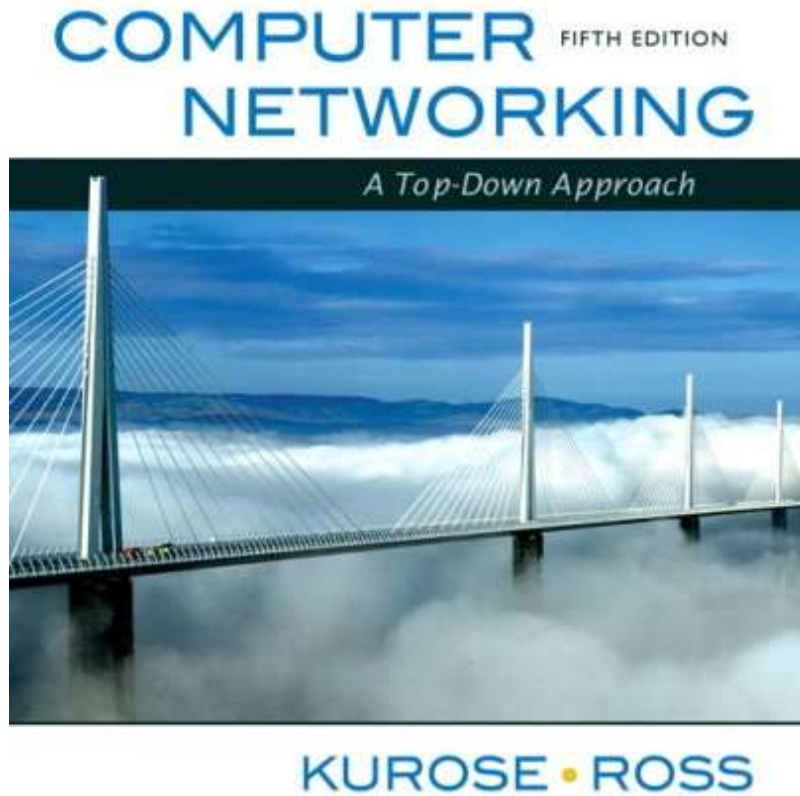
- ❑ các tiến trình = các đĩa trẻ
- ❑ các thông điệp = thư trong bao thư
- ❑ các host = các gia đình
- ❑ giao thức transport = An và Bình
- ❑ giao thức lớp network = dịch vụ bưu điện

# Các giao thức trên Internet

- ❑ tin cậy, truyền theo thứ tự (TCP)
  - điều khiển tắc nghẽn
  - điều khiển lưu lượng
  - thiết lập kết nối
- ❑ không tin cậy, truyền không theo thứ tự: UDP
  - mở rộng của giao thức IP
- ❑ không có các dịch vụ:
  - bảo đảm trễ
  - bảo đảm bandwidth



## 3.2 Multiplexing và demultiplexing




# Multiplexing/demultiplexing


## Demultiplexing tại host nhận:

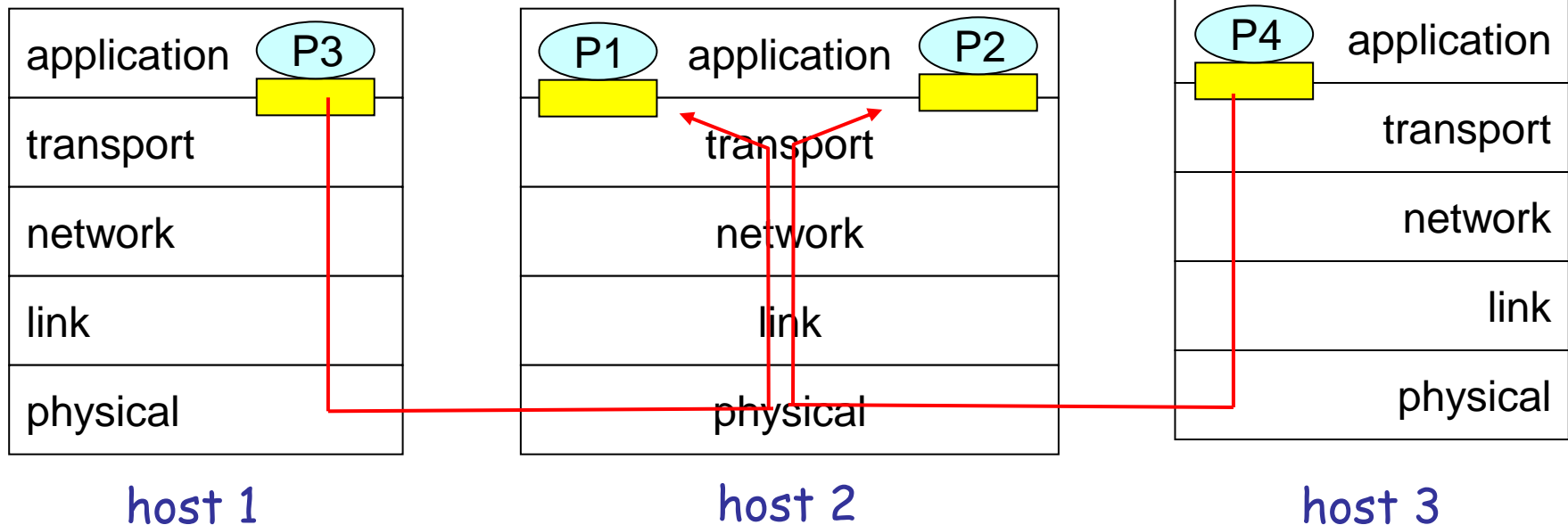
vận chuyển các đoạn đã nhận được đến đúng socket

## Multiplexing tại host gửi:

thu nhận dữ liệu từ nhiều socket, đóng gói dữ liệu với header (sẽ dùng sau đó cho demultiplexing)

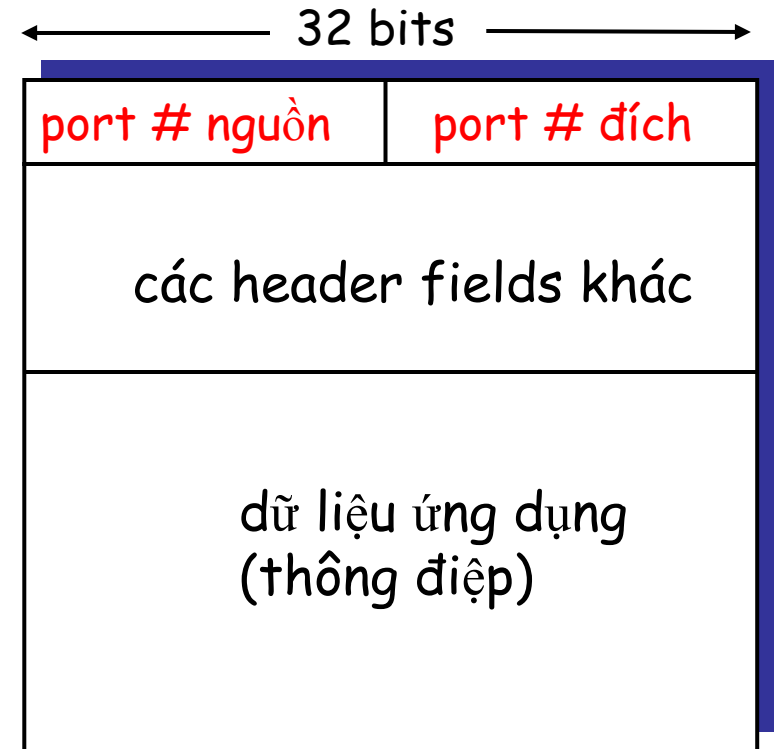
 = socket

 = tiến trình



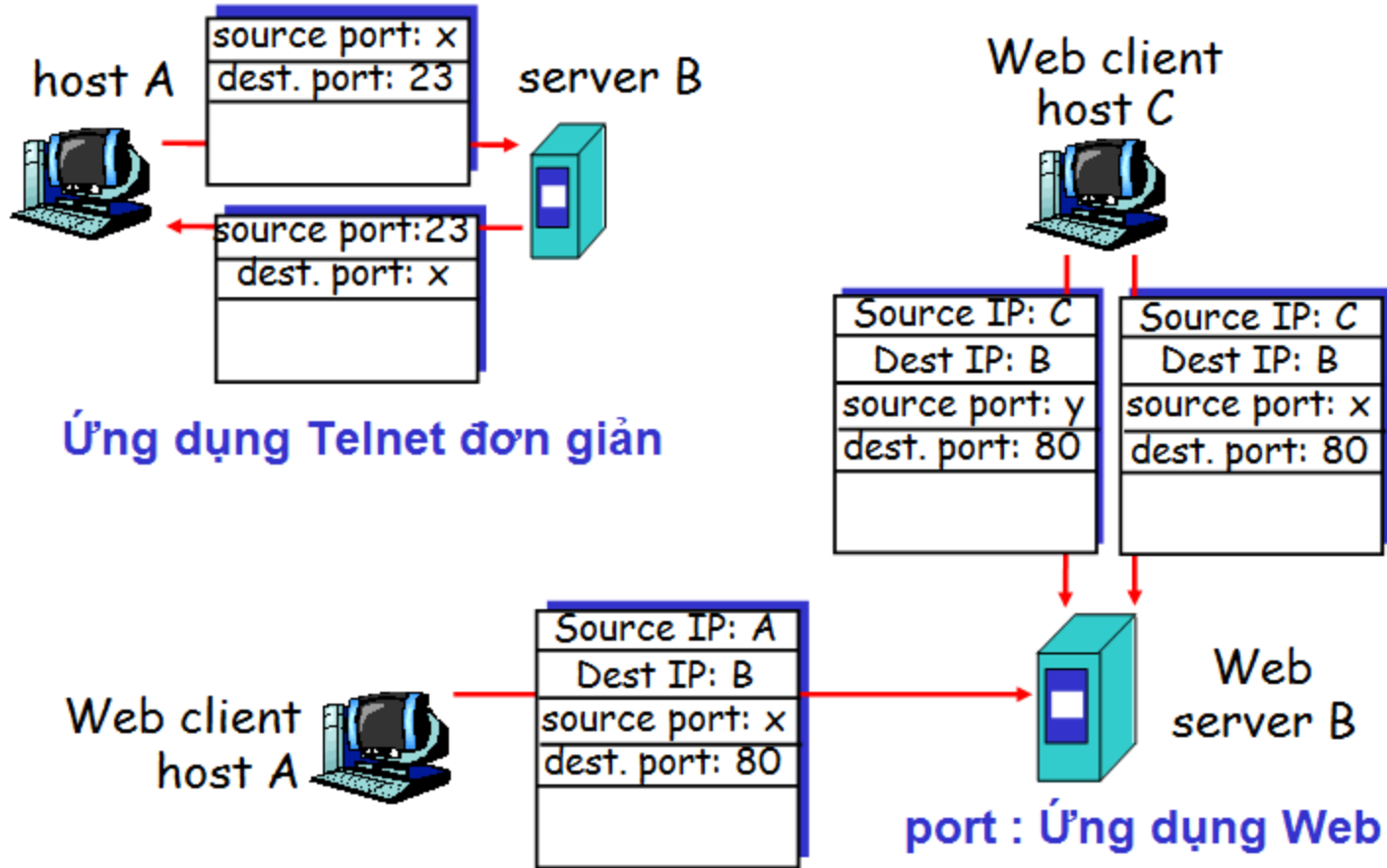
# Demultiplexing làm việc như thế nào

- host nhận các IP datagrams
  - mỗi datagram có địa chỉ IP nguồn và IP đích
  - mỗi datagram mang 1 đoạn của tầng giao vận
  - mỗi đoạn có số port nguồn và đích
- host dùng địa chỉ IP & số port để điều hướng đoạn đến socket thích hợp

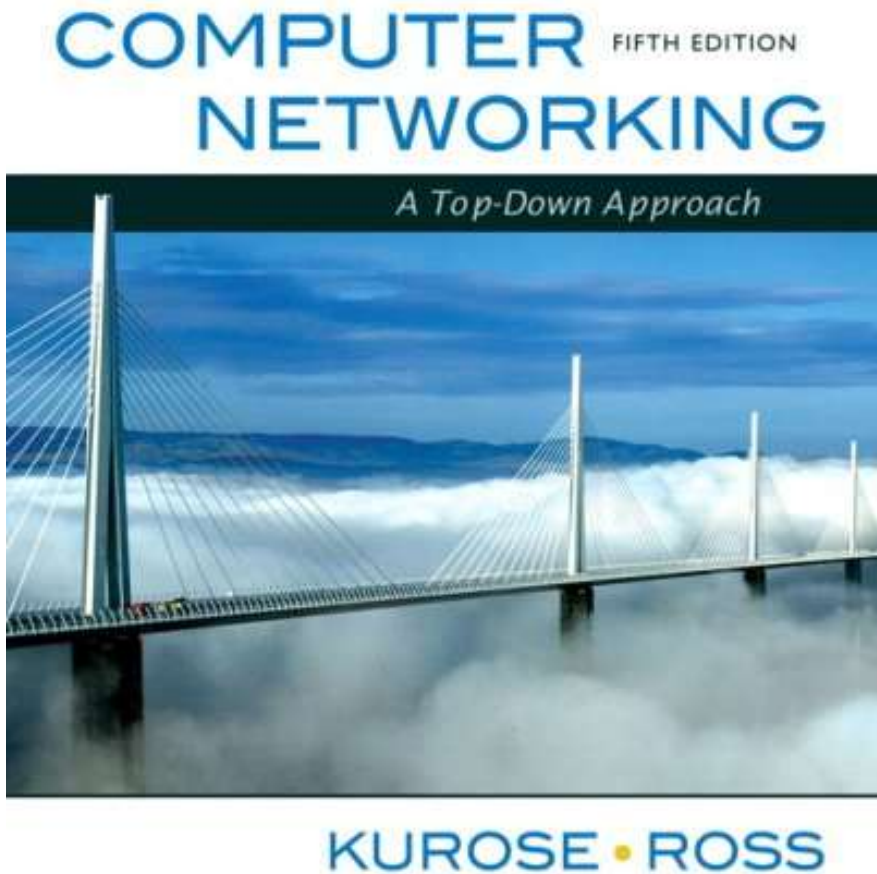


dạng thức đoạn TCP/UDP

# Ví dụ Multiplexing/demultiplexing



## 3.3 Vận chuyển không kết nối: UDP





# UDP: User Datagram Protocol [RFC 768]

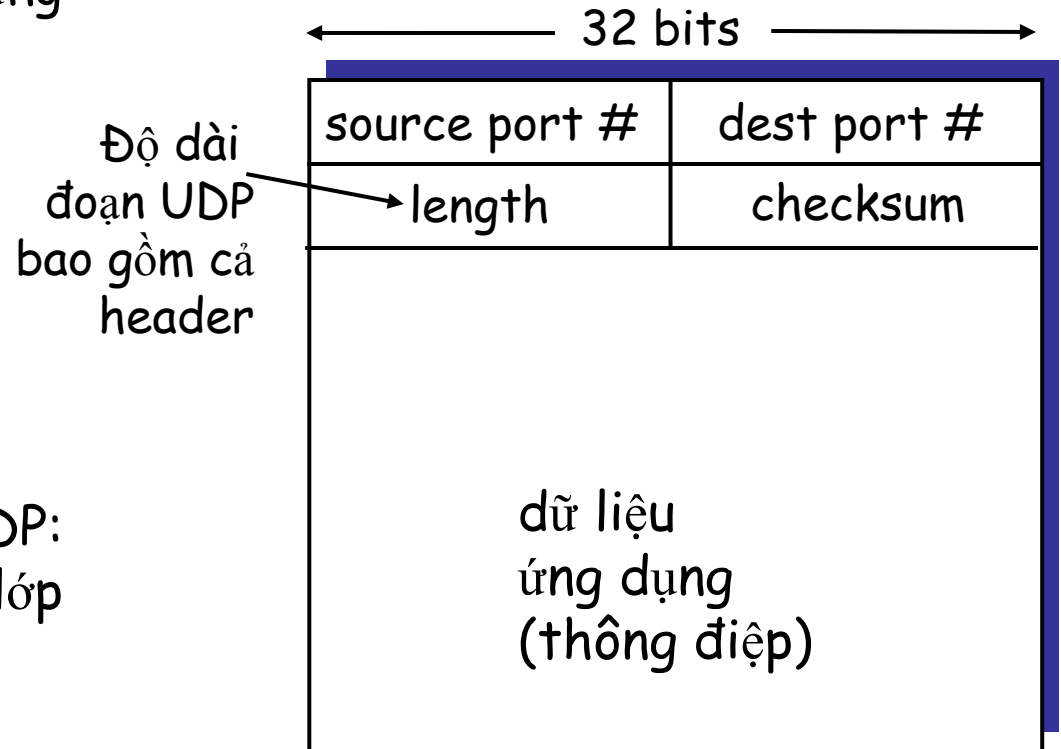
- ❑ giao thức Internet transport "đơn giản hóa"
- ❑ dịch vụ "best effort", các đoạn UDP có thể:
  - mất mát
  - vận chuyển không thứ tự đến ứng dụng
- ❑ *connectionless (không kết nối)*:
  - không bắt tay giữa bên nhận và bên gửi UDP
  - mỗi đoạn UDP được quản lý độc lập

## Có UDP để làm gì?

- ❑ không thiết lập kết nối (giúp có thể thêm delay)
- ❑ đơn giản: không trạng thái kết nối tại nơi gửi, nơi nhận
- ❑ header của đoạn nhỏ
- ❑ không điều khiển tắc nghẽn: UDP có thể gửi nhanh nhất theo mong muốn

# UDP: (tt)

- thường dùng cho các ứng dụng streaming multimedia
  - chịu mất mát
  - cảm nhận tốc độ
- ngoài ra, UDP dùng
  - DNS
  - SNMP
- truyền tin cậy trên UDP: thêm khả năng này tại lớp application
  - sửa lỗi



dạng thức đoạn UDP

# UDP checksum

Mục tiêu: kiểm tra các “lỗi” (các bit cờ đã bật lên) trong các đoạn đã truyền

## bên gửi:

- ❑ đối xử các nội dung đoạn như một chuỗi các số nguyên 16-bit
- ❑ checksum: bổ sung(tổng bù 1) của các nội dung đoạn
- ❑ đặt giá trị checksum vào trường UDP checksum

## bên nhận:

- ❑ tính toán checksum của đoạn đã nhận
- ❑ kiểm tra giá trị trên có bằng với giá trị trong trường checksum:
  - NO - có lỗi xảy ra
  - YES - không có lỗi.
  - *Nhưng có thể còn lỗi khác nữa? Xem tiếp phần sau ....*

# Ví dụ Checksum

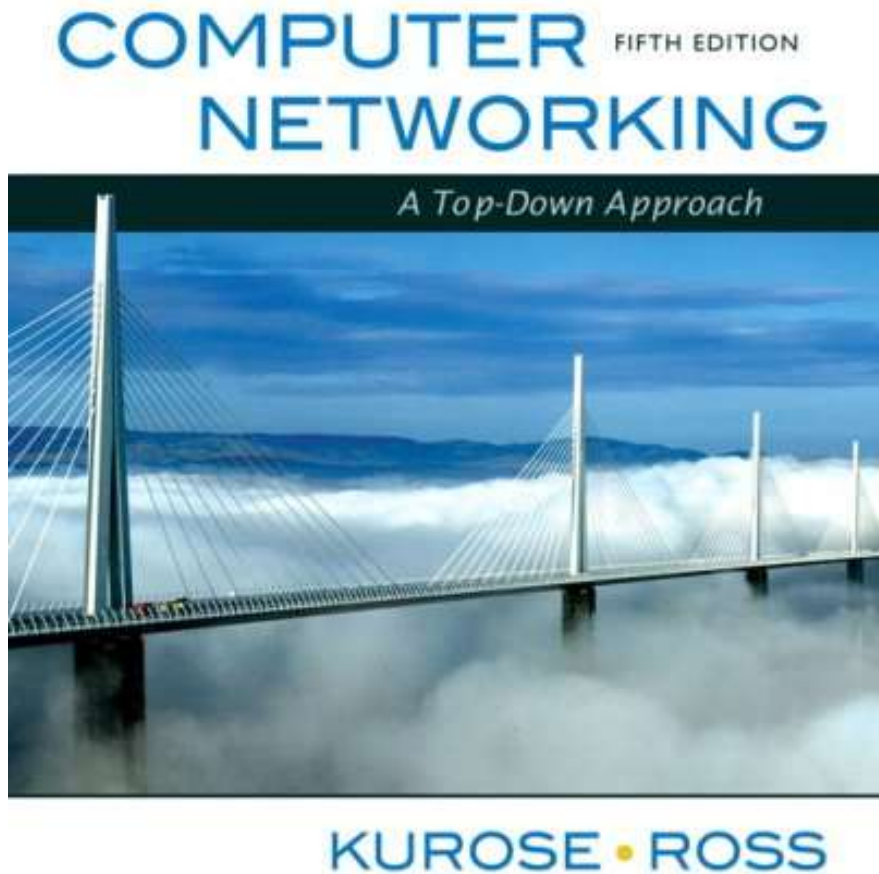
## □ Lưu ý

- Khi cộng các số, một bit nhớ ở phía cao nhất có thể sẽ phải thêm vào kết quả

## □ Ví dụ: cộng hai số nguyên 16-bit

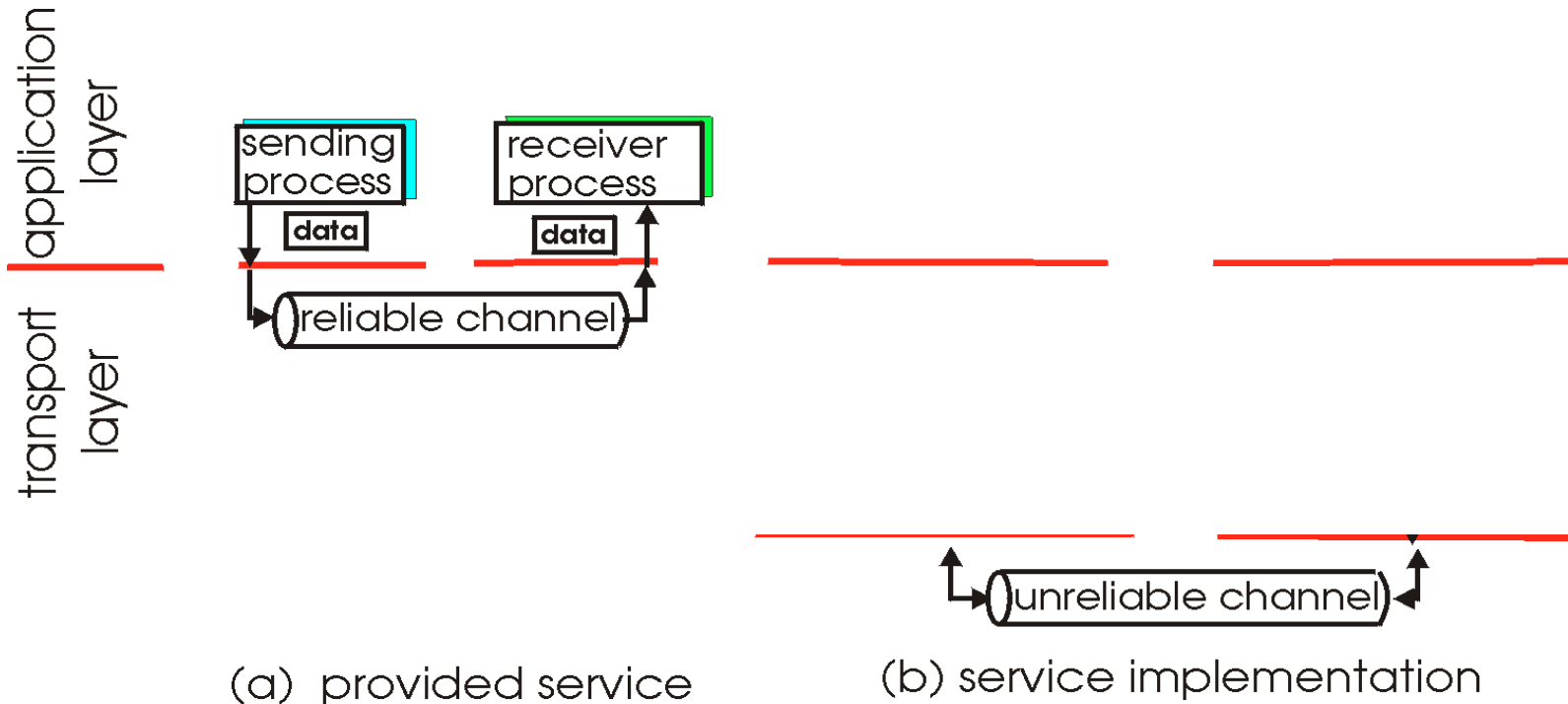
	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
	<hr/>																
bit dư	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
	<hr/>																
tổng	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0	
checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1	

## 3.4 Các nguyên lý của việc truyền dữ liệu tin cậy



# Các nguyên lý truyền dữ liệu tin cậy

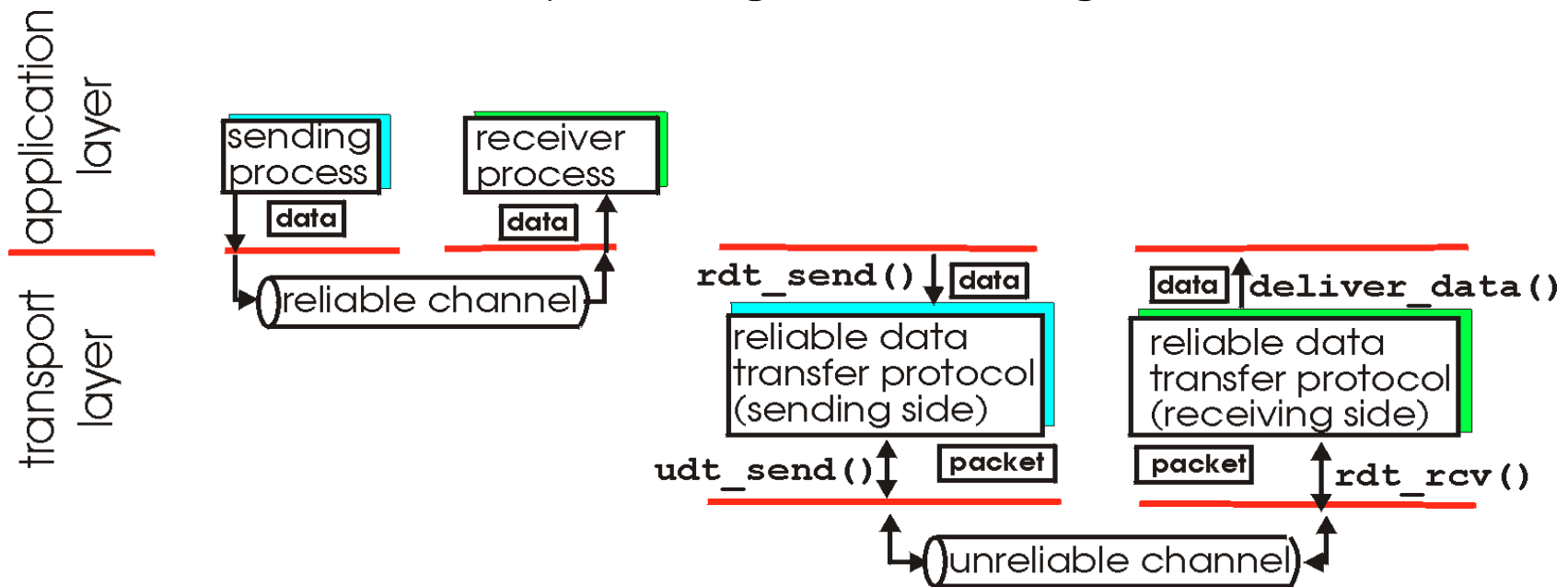
- ❑ quan trọng trong các lớp application, transport, link
- ❑ là danh sách 10 vấn đề quan trọng nhất của mạng



- ❑ các đặc thù của kênh truyền không tin cậy sẽ xác định sự phức tạp của giao thức truyền dữ liệu data transfer protocol (rdt)

# Các nguyên lý truyền dữ liệu tin cậy

- ❑ quan trọng trong các lớp application, transport, link
- ❑ là danh sách 10 vấn đề quan trọng nhất của mạng



(a) provided service

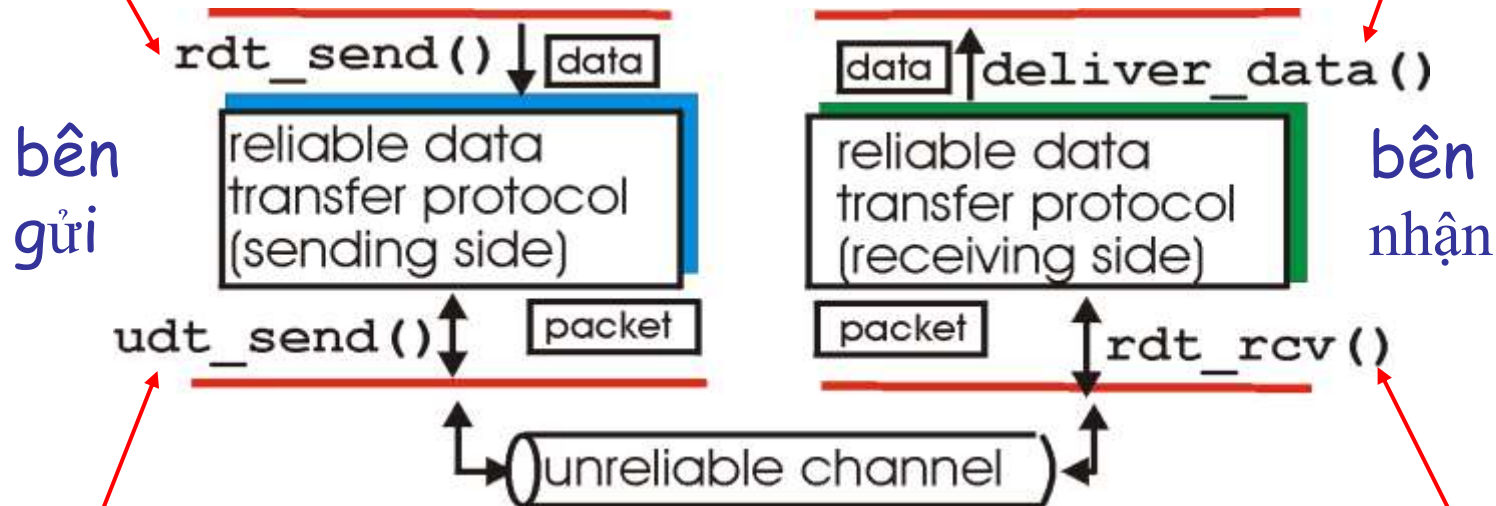
(b) service implementation

- ❑ các đặc thù của kênh truyền không tin cậy sẽ xác định sự phức tạp của giao thức truyền dữ liệu data transfer protocol (rdt)

# Truyền dữ liệu tin cậy

**rdt\_send()** : được gọi bởi lớp app.  
Chuyển dữ liệu cần truyền đến lớp cao hơn bên nhận

**deliver\_data()** : được gọi bởi rdt để truyền dữ liệu đến lớp cao hơn



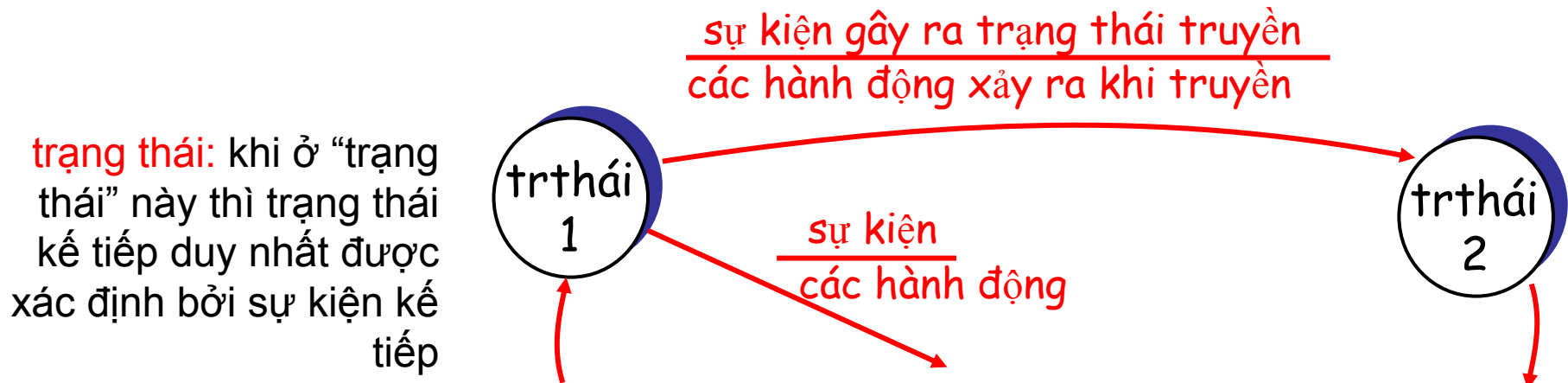
**udt\_send()** : được gọi bởi rdt,  
để truyền các gói trên kênh  
không tin cậy đến nơi nhận

**rdt\_rcv()** : được gọi khi gói đến  
kênh bên nhận



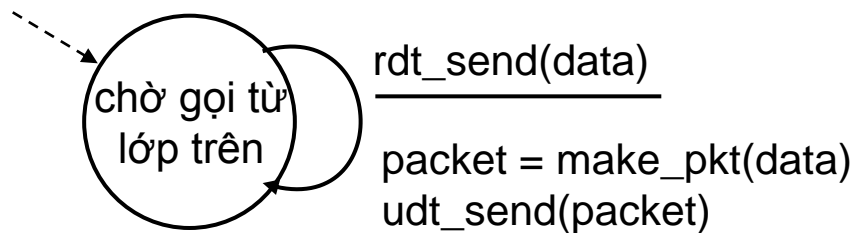
# Truyền dữ liệu tin cậy

- ❑ chỉ xem xét truyền dữ liệu theo 1 hướng duy nhất
  - nhưng điều khiển lưu lượng thông tin sẽ theo cả 2 chiều!
- ❑ dùng máy trạng thái hữu hạn (finite state machines-FSM) để xác định bên gửi, bên nhận

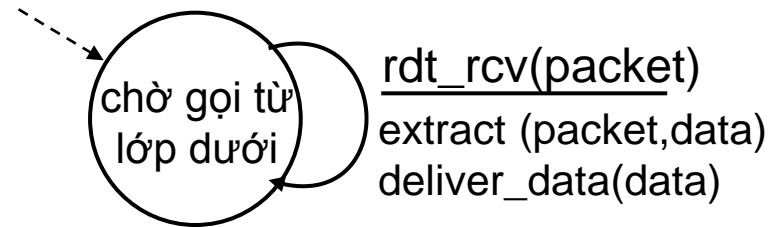


# Rdt1.0: truyền dữ liệu tin cậy trên 1 kênh truyền tin cậy

- ❑ kênh truyền tin cậy hoàn toàn
  - không có các lỗi
  - không mất mát các gói
- ❑ các FSM phân biệt cho bên gửi, bên nhận:
  - bên gửi gửi dữ liệu vào kênh tin cậy
  - bên nhận nhận dữ liệu từ kênh tin cậy



bên gửi

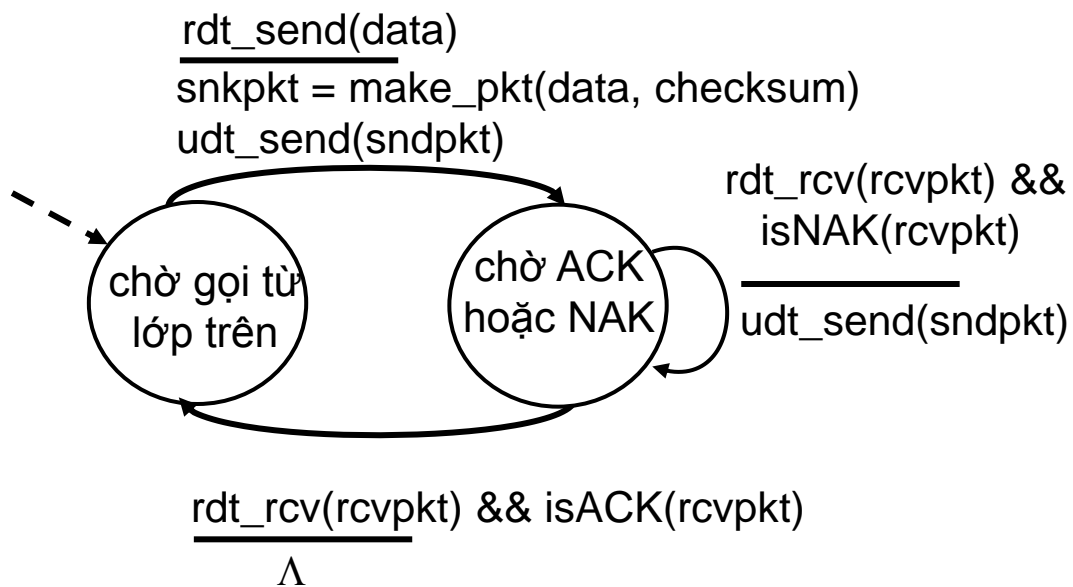


bên nhận

# Rdt2.0: kênh truyền có lỗi

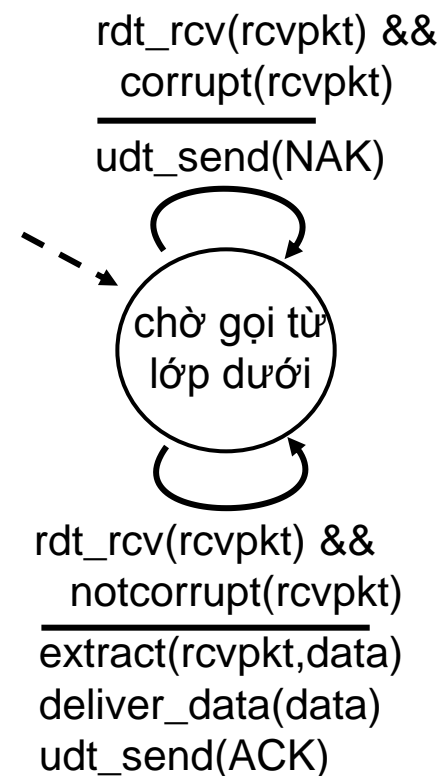
- ❑ Kênh truyền bên dưới có lỗi bit
  - checksum để kiểm tra các lỗi
- ❑ *Hỏi*: làm sao khôi phục các lỗi?
  - *các acknowledgements (ACK)*: bên nhận rõ ràng thông báo cho bên gửi rằng quá trình nhận gói tốt
  - *các negative acknowledgements (NAK)*: bên nhận rõ ràng thông báo cho bên gửi rằng quá trình nhận gói có lỗi
  - bên gửi gửi lại gói nào được xác nhận là NAK
- ❑ **các cơ chế mới trong rdt2.0 (sau rdt1.0)**:
  - kiểm tra lỗi
  - nhận phản hồi: các thông điệp điều khiển (ACK,NAK) bên nhận → bên gửi

# rdt2.0: đặc tả FSM

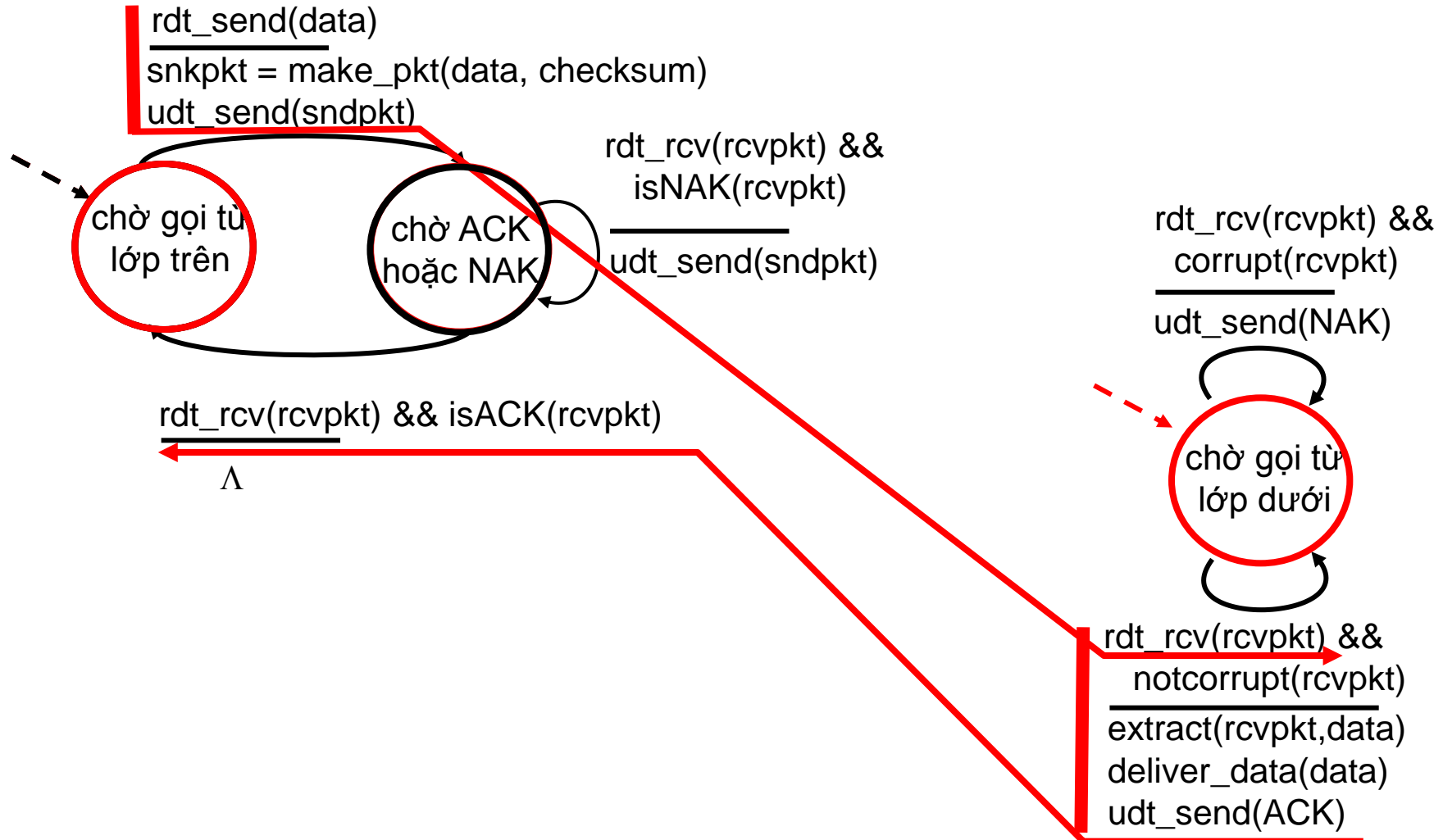


**bên gửi**

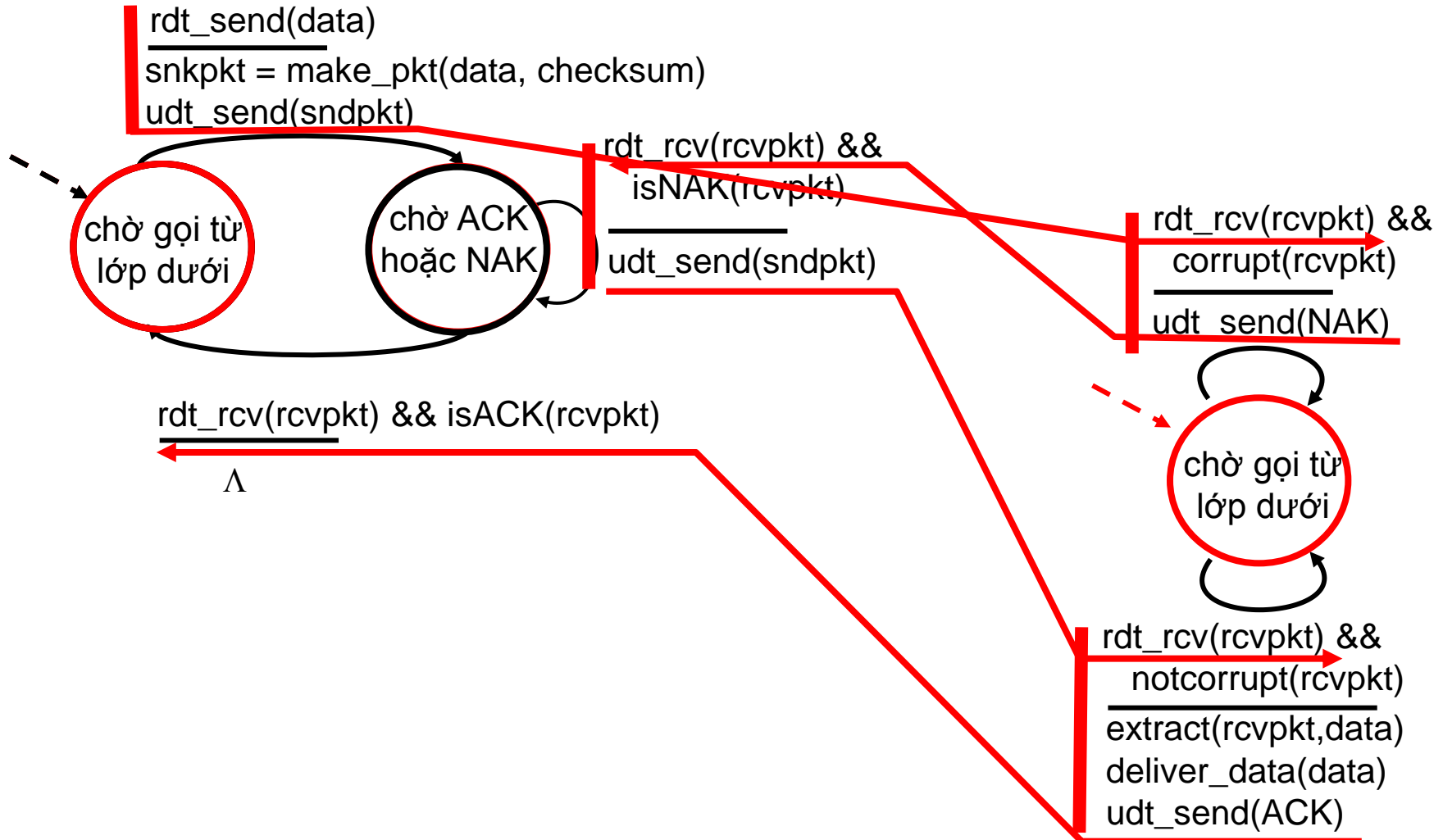
**bên nhận**



# rdt2.0: hoạt động khi không lỗi



# rdt2.0: hoạt động khi có lỗi



# rdt2.0 có lỗi hỏng nghiêm trọng!

## Điều gì xảy ra nếu ACK/NAK bị hỏng?

- ❑ bên gửi không biết điều gì đã xảy ra tại bên nhận!
- ❑ không thể đơn phương truyền lại: khả năng trùng lặp

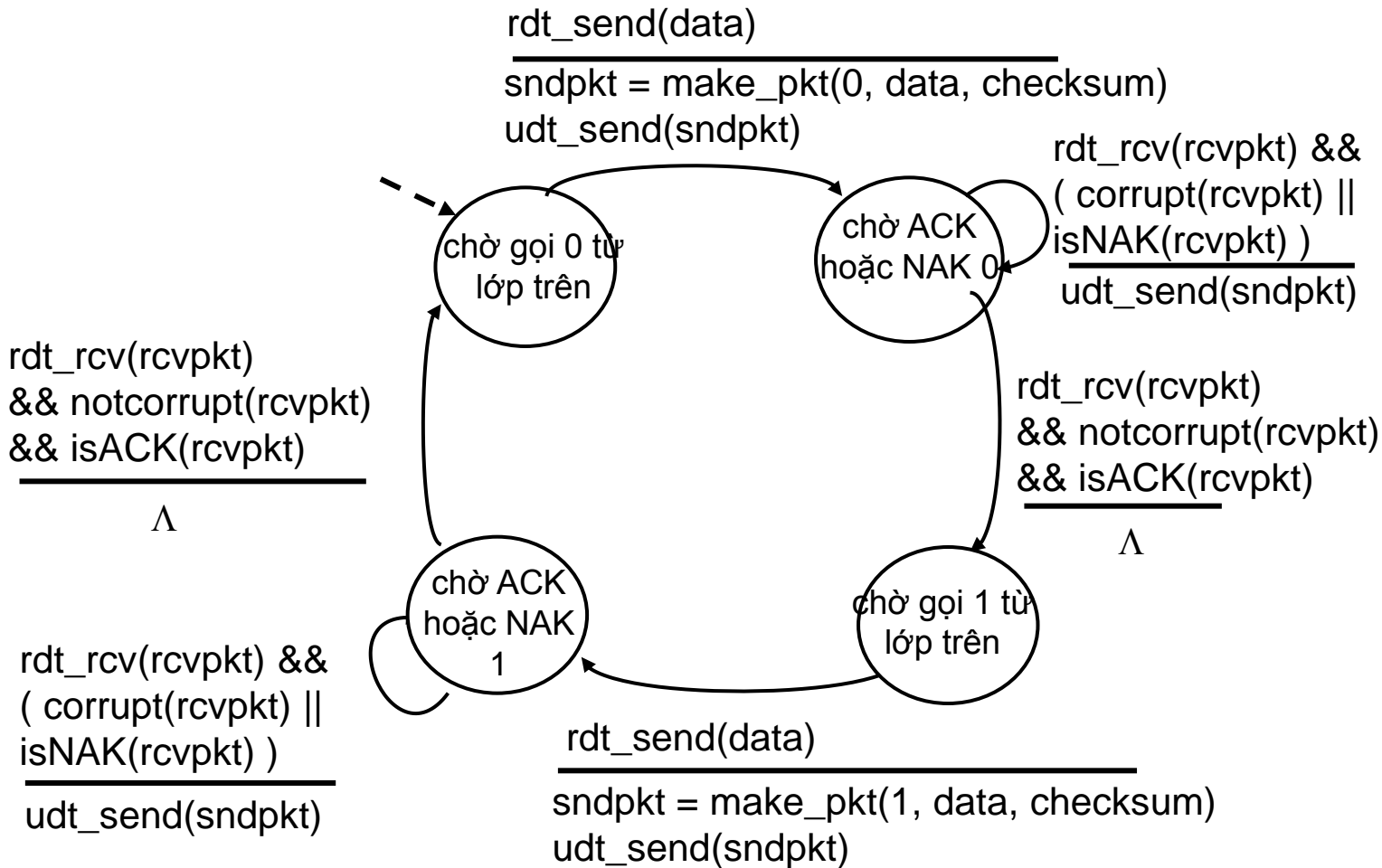
## Quản lý trùng lặp:

- ❑ Bên gửi truyền lại gói hiện tại nếu ACK/NAK bị hỏng
- ❑ bên gửi *thêm số thứ tự* vào mỗi gói
- ❑ bên nhận hủy (không nhận) gói trùng lặp

### dừng và chờ

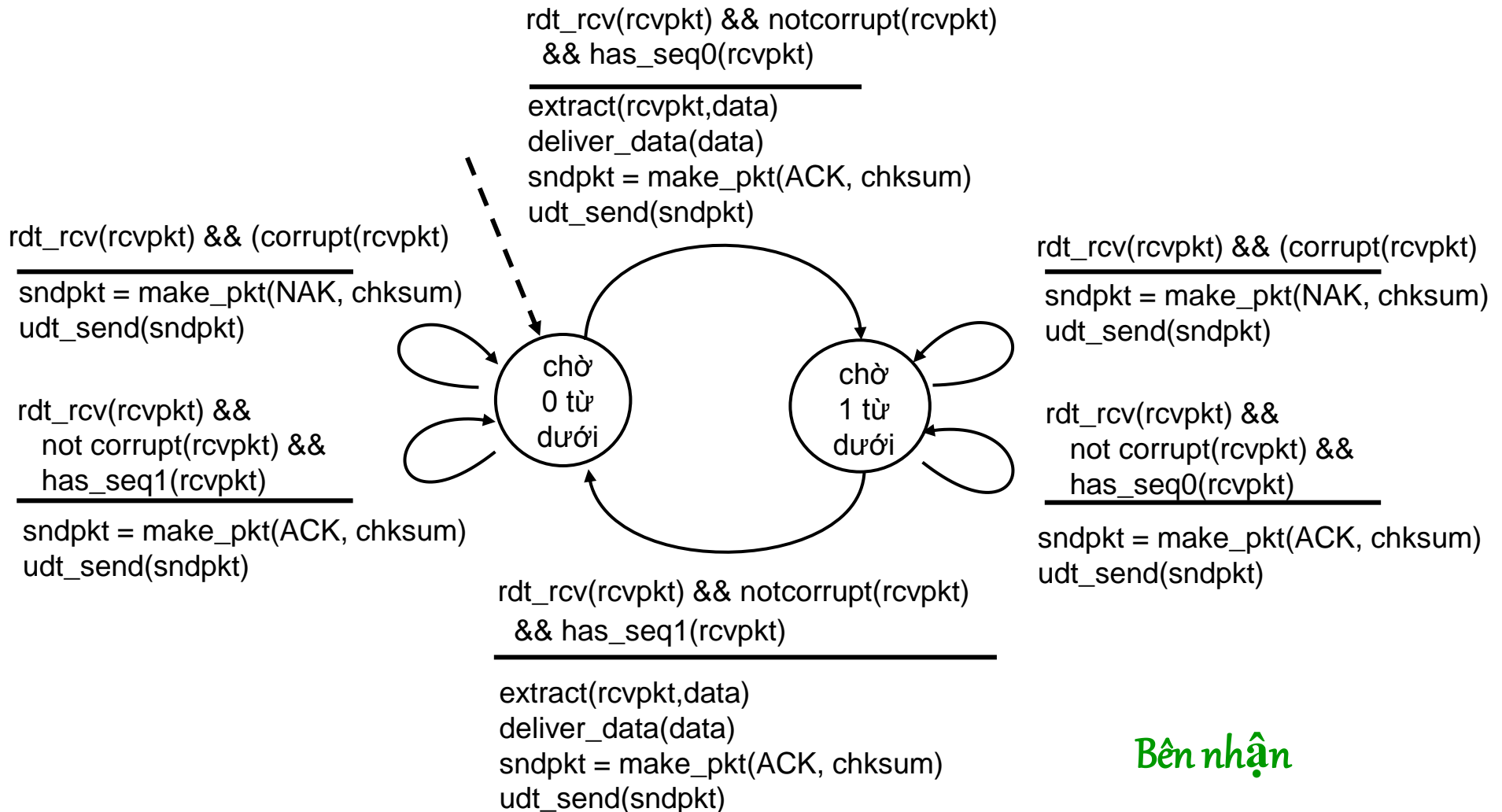
bên gửi gửi 1 gói,  
sau đó dừng lại chờ phản hồi  
từ bên nhận

# rdt2.1: bên gửi quản lý các ACK/NAK bị hỏng





# rdt2.1: bên gửi quản lý các ACK/NAK bị hỏng



Bên nhận

# rdt2.1: thảo luận

## bên gửi:

- ❑ số thứ tự  $\#$  thêm vào gói
- ❑ chỉ cần hai số thứ tự (0,1) là đủ. Tại sao?
- ❑ phải kiểm tra nếu việc nhận ACK/NAK bị hỏng
- ❑ số trạng thái tăng lên 2 lần
  - trạng thái phải "nhớ" gói "hiện tại" có số thứ tự là 0 hay 1

## bên nhận:

- ❑ phải kiểm tra có nhận trùng gói không
  - trạng thái chỉ rõ có hay không mong chờ số thứ tự 0 hoặc 1
- ❑ chú ý: bên nhận không biết ACK/NAK vừa rồi của nó có được bên gửi nhận tốt hay không

# rdt2.2: một giao thức không cần NAK

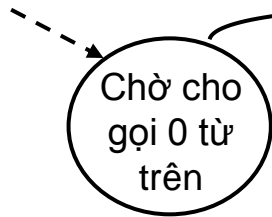
- ❑ chức năng giống như rdt2.1, chỉ dùng các ACK
- ❑ thay cho NAK, bên nhận gửi ACK cho gói vừa rồi đã nhận tốt
  - bên nhận phải *rõ ràng* chèn số thứ tự của gói vừa ACK
- ❑ trùng ACK tại bên gửi hậu quả giống như hành động của NAK: *truyền lại gói vừa rồi*

# rdt2.2: gửi, nhận các mảnh

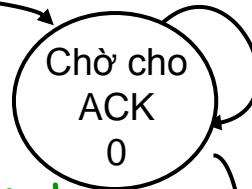
rdt\_send(data)

sndpkt = make\_pkt(0, data, checksum)

udt\_send(sndpkt)



**gửi phân mảnh**  
FSM

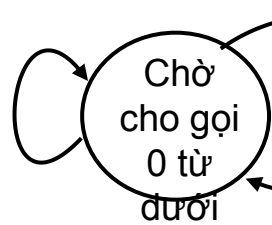


rdt\_rcv(rcvpkt) &&  
( corrupt(rcvpkt) ||  
**isACK(rcvpkt,1)** )  
**udt\_send(sndpkt)**

rdt\_rcv(rcvpkt)  
&& notcorrupt(rcvpkt)  
&& **isACK(rcvpkt,0)**

Λ

rdt\_rcv(rcvpkt) &&  
( corrupt(rcvpkt) ||  
**has\_seq1(rcvpkt)** )  
**udt\_send(sndpkt)**



**nhận phân mảnh**  
FSM

rdt\_rcv(rcvpkt) && notcorrupt(rcvpkt)  
&& has\_seq1(rcvpkt)

extract(rcvpkt,data)  
deliver\_data(data)  
**sndpkt = make\_pkt(ACK1, chksum)**  
udt\_send(sndpkt)

# rdt3.0: các kênh với lỗi và mất mát

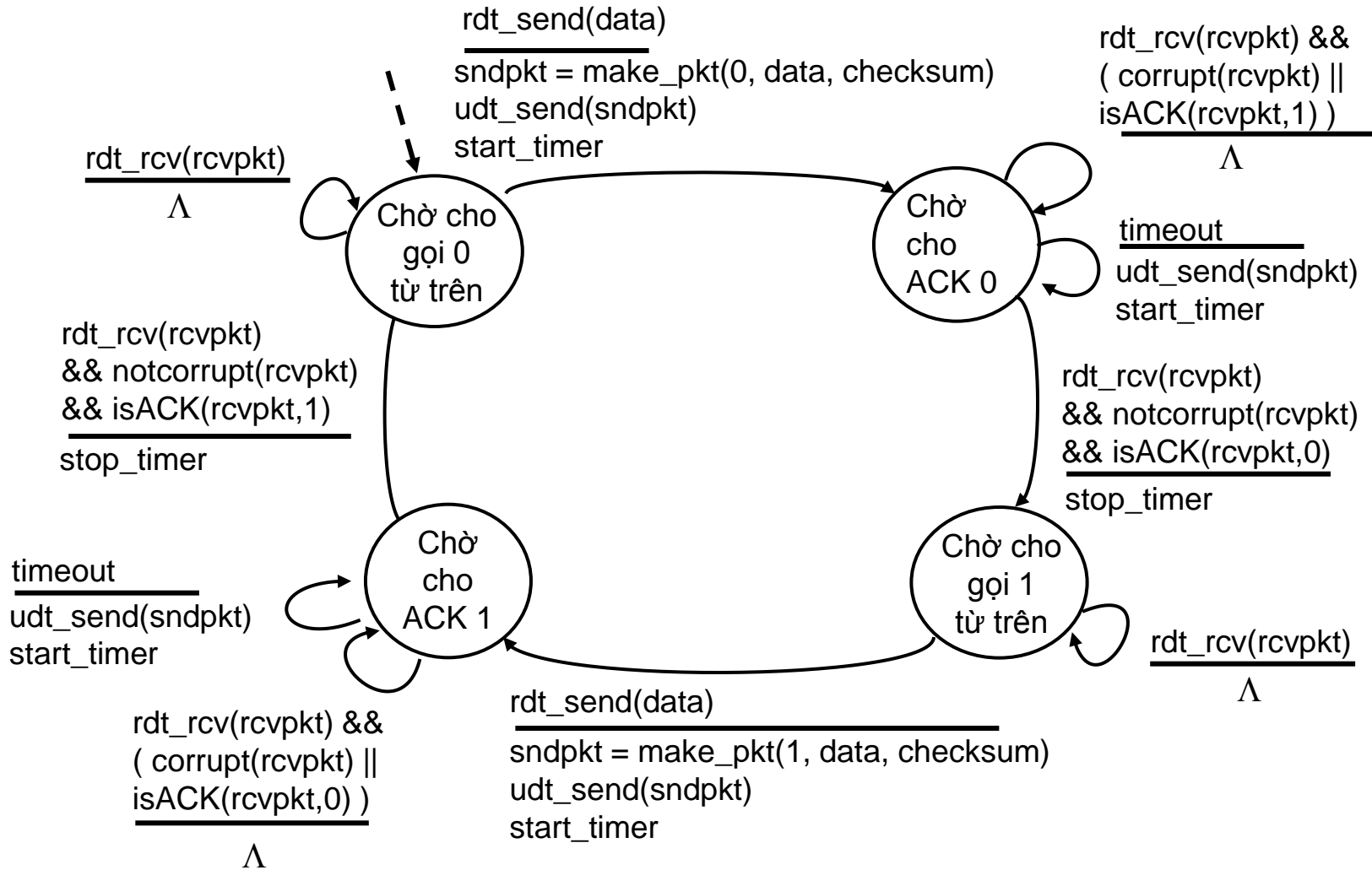
Giả định mới: kênh truyền cũng có thể làm mất các gói (dữ liệu hoặc các ACK)

- checksum, số thứ tự, các ACK, các việc truyền lại sẽ hỗ trợ nhưng không đủ

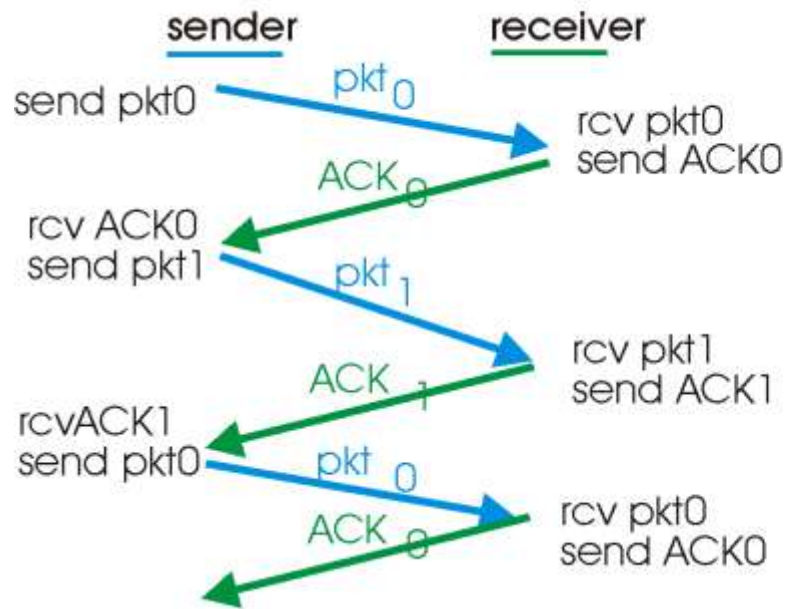
Cách tiếp cận: bên gửi chờ ACK trong khoảng thời gian "chấp nhận được"

- ❑ truyền lại nếu không nhận ACK trong khoảng thời gian này
- ❑ nếu gói (hoặc ACK) chỉ trễ (không mất):
  - truyền lại sẽ gây trùng, nhưng dùng số thứ tự sẽ giải quyết được
  - bên nhận phải xác định số thứ tự của gói vừa gửi ACK
- ❑ cần bộ định thì đếm lùi

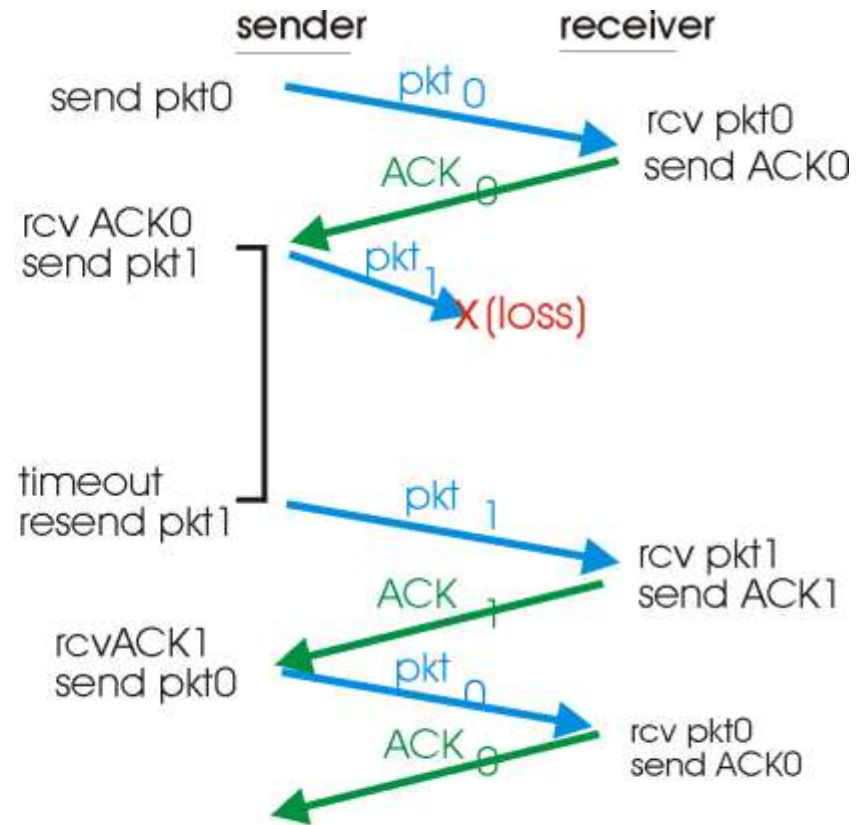
# rdt3.0 gửi



# hành động của rdt3.0

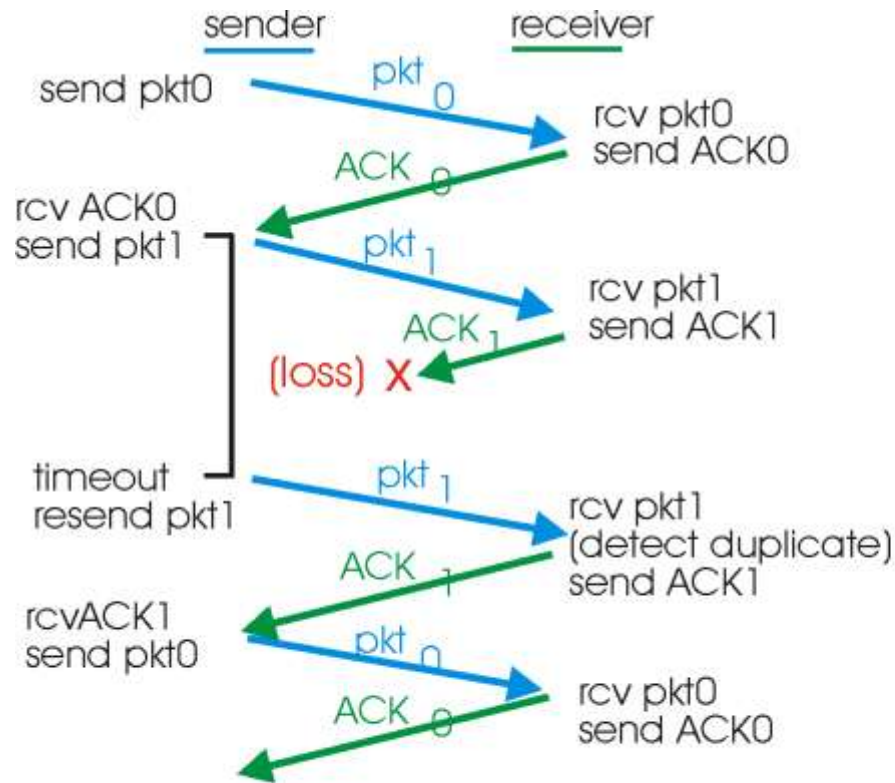


(a) operation with no loss

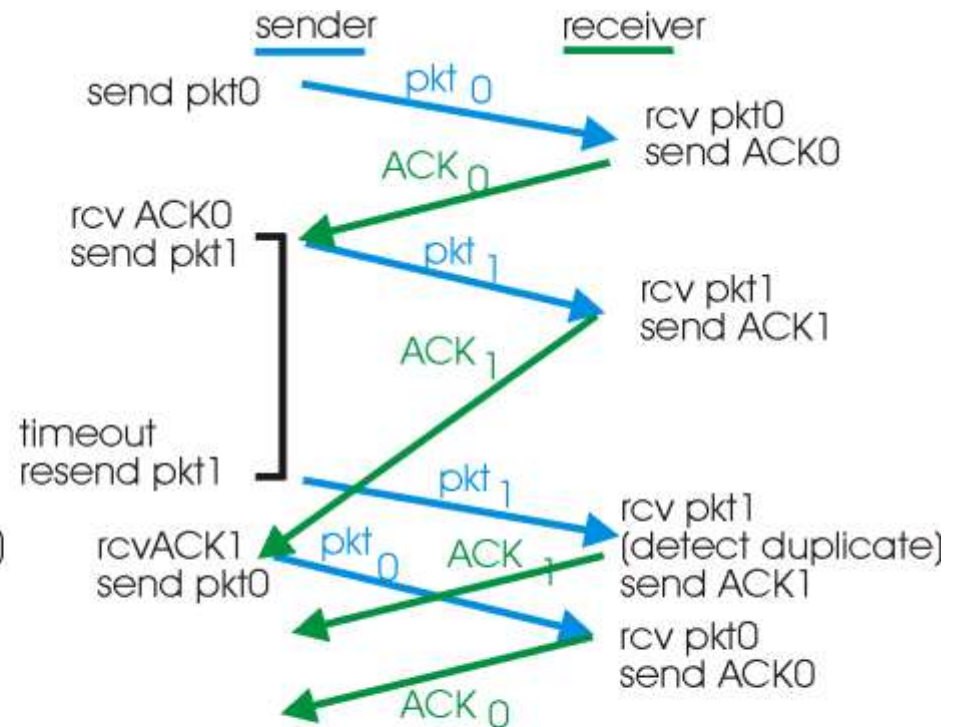


(b) lost packet

# hành động của rdt3.0



(c) lost ACK



(d) premature timeout



# Hiệu suất của rdt3.0

- ❑ rdt3.0 làm việc được, nhưng hiệu suất thấp
- ❑ ví dụ: liên kết 1 Gbps, trễ lan truyền giữa hai đầu cuối là 15 ms, gói 1KB:

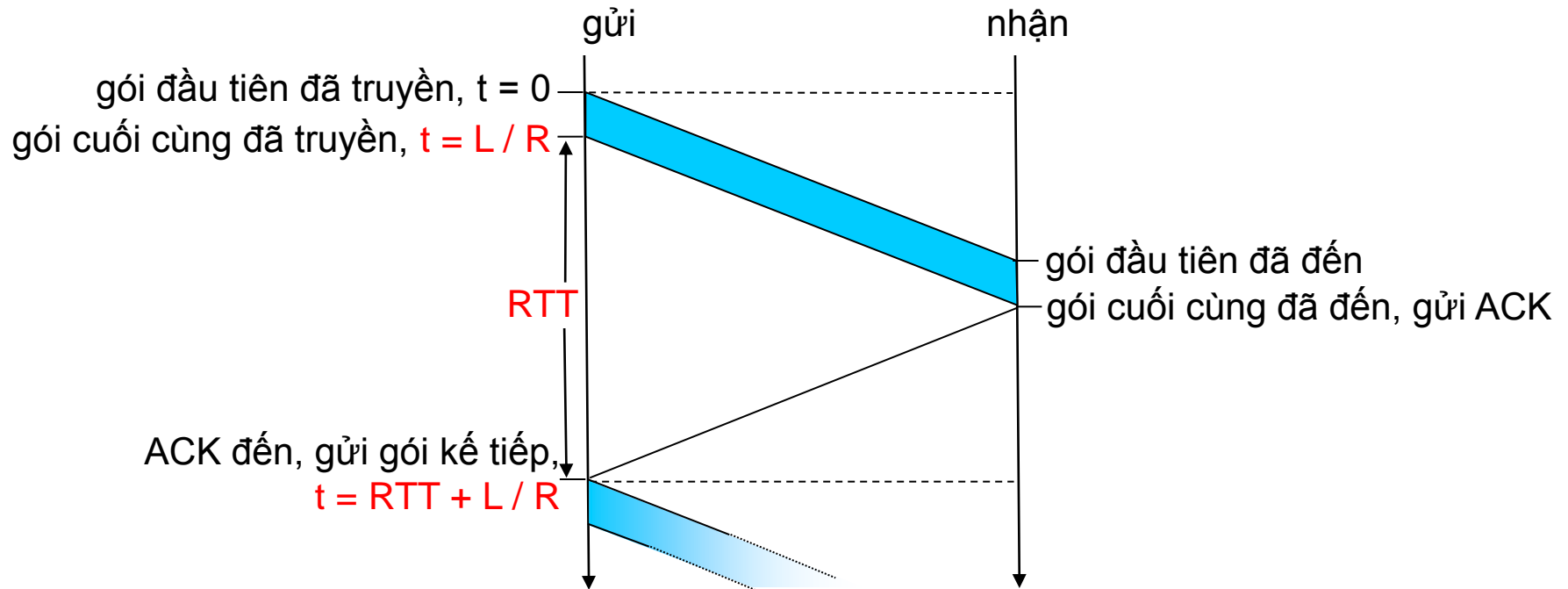
$$T_{\text{truyền}} = \frac{L \text{ (độ dài gói tính bằng bits)}}{R \text{ (tốc độ truyền, bps)}} = \frac{8 \times 10^3 \text{ b/pkt}}{10^9 \text{ b/sec}} = 8 \text{ microsec}$$

- $U_{\text{sender}}$ : **độ khả dụng** – Hiệu suất sử dụng đường truyền

$$U_{\text{sender}} = \frac{L / R}{RTT + L / R} = \frac{.008}{30.008} = 0.00027$$

- gói 1KB mỗi 30 msec -> 33kB/s trên đường truyền 1 Gbps
- giao thức network hạn chế việc dùng các tài nguyên vật lý!

# rdt3.0: hoạt động dừng-và-chờ

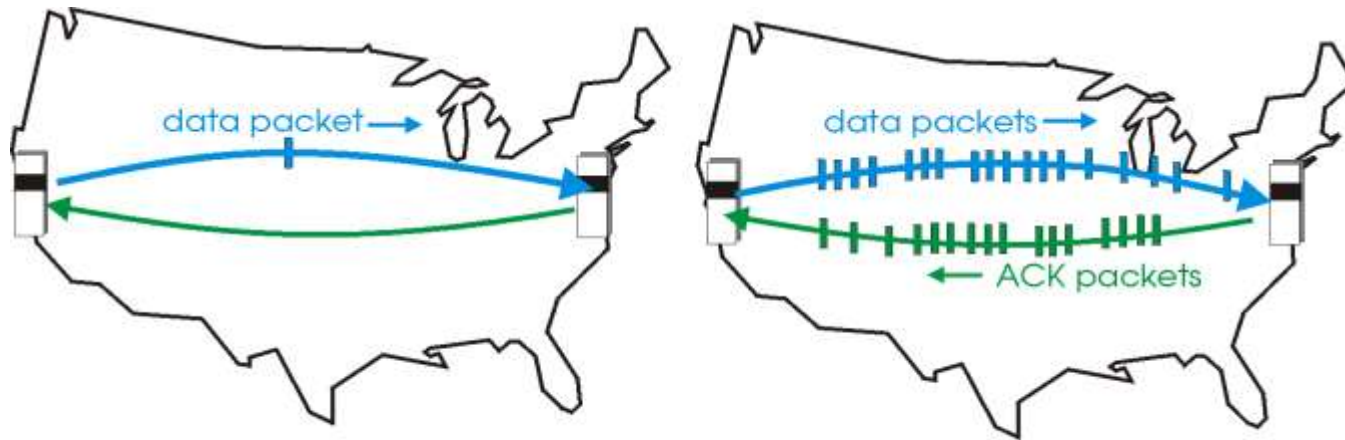


$$U_{\text{sender}} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

# Các giao thức Pipelined

**Pipelining:** bên gửi cho phép gửi nhiều gói đồng thời, không cần chờ báo nhận được

- nhóm các số thứ tự phải tăng dần
- phải có bộ nhớ đệm tại nơi gửi và/hoặc nơi nhận

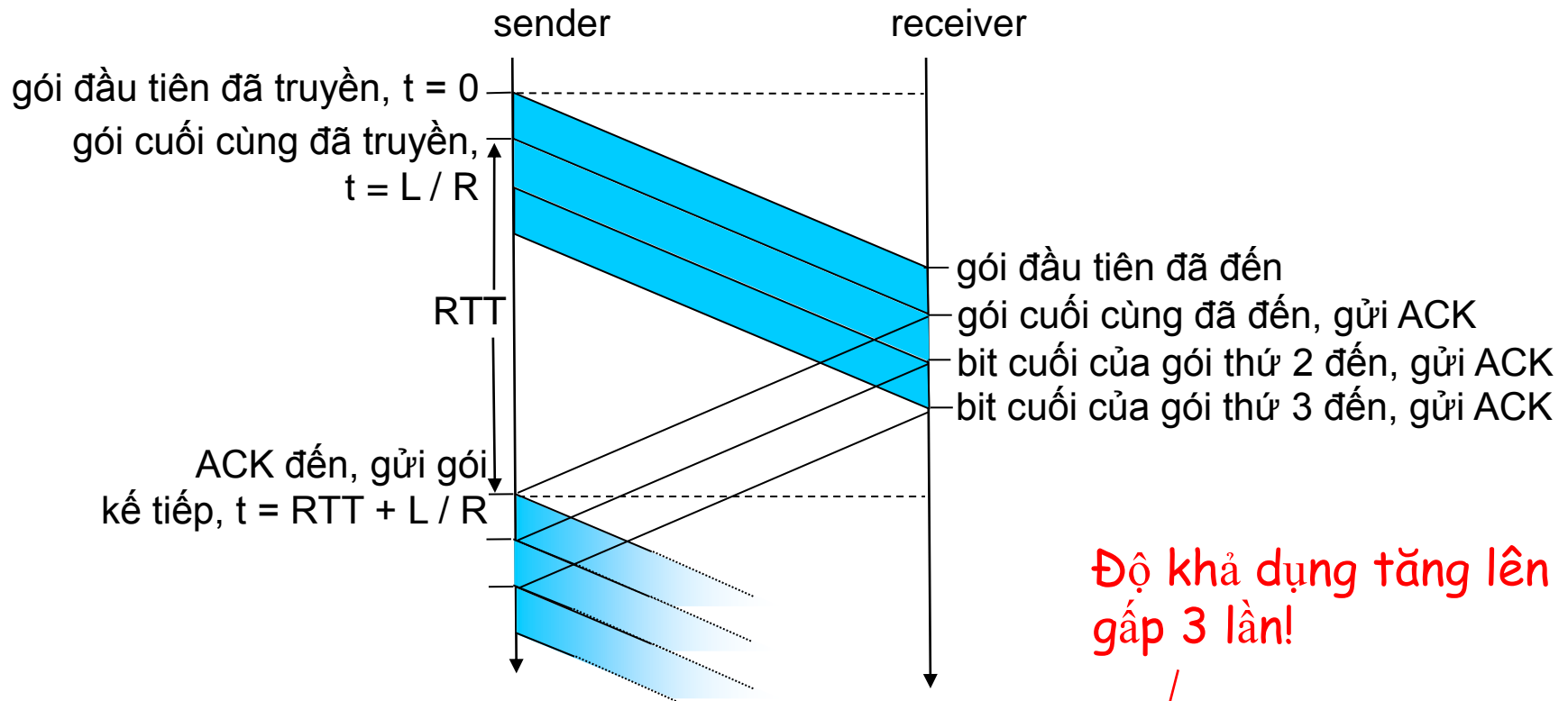


(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

- hai dạng phổ biến của các giao thức pipelined: *go-Back-N*, *Selective Repeat*

# Pipelining: độ khả dụng tăng



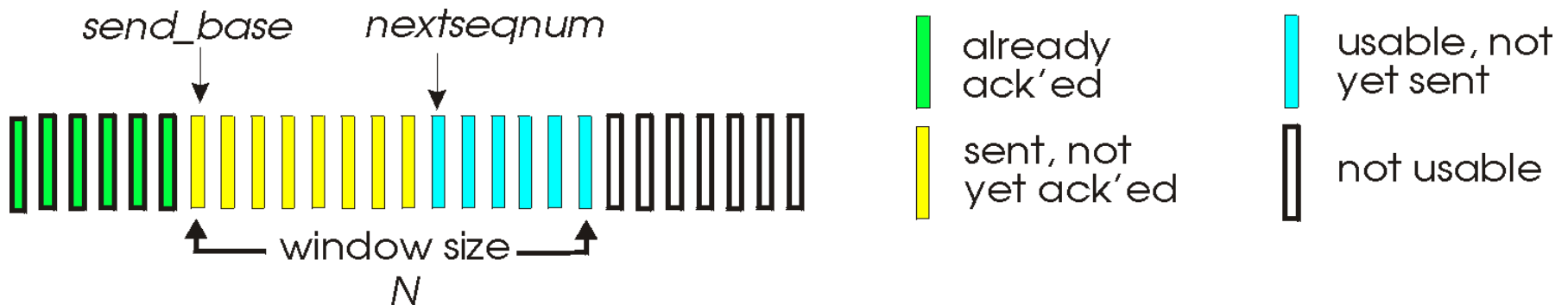
Độ khả dụng tăng lên  
gấp 3 lần!

$$U_{\text{sender}} = \frac{3 * L / R}{RTT + L / R} = \frac{.024}{30.008} = 0.0008$$

# Go-Back-N

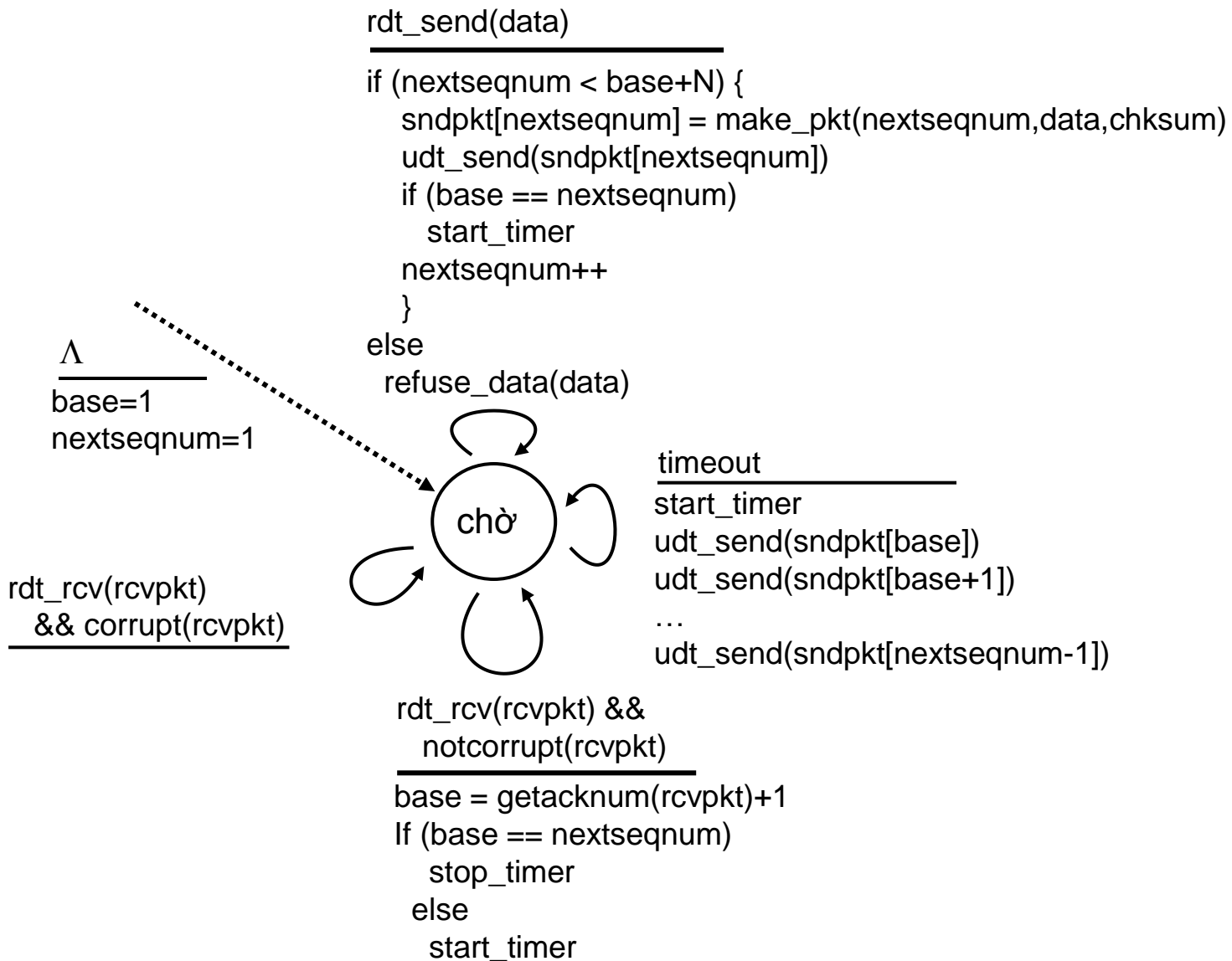
## Bên gửi:

- ❑ k-bit số thứ tự trong header của gói
- ❑ “cửa sổ” tăng lên đến N, cho phép gửi các gói liên tục không cần ACK

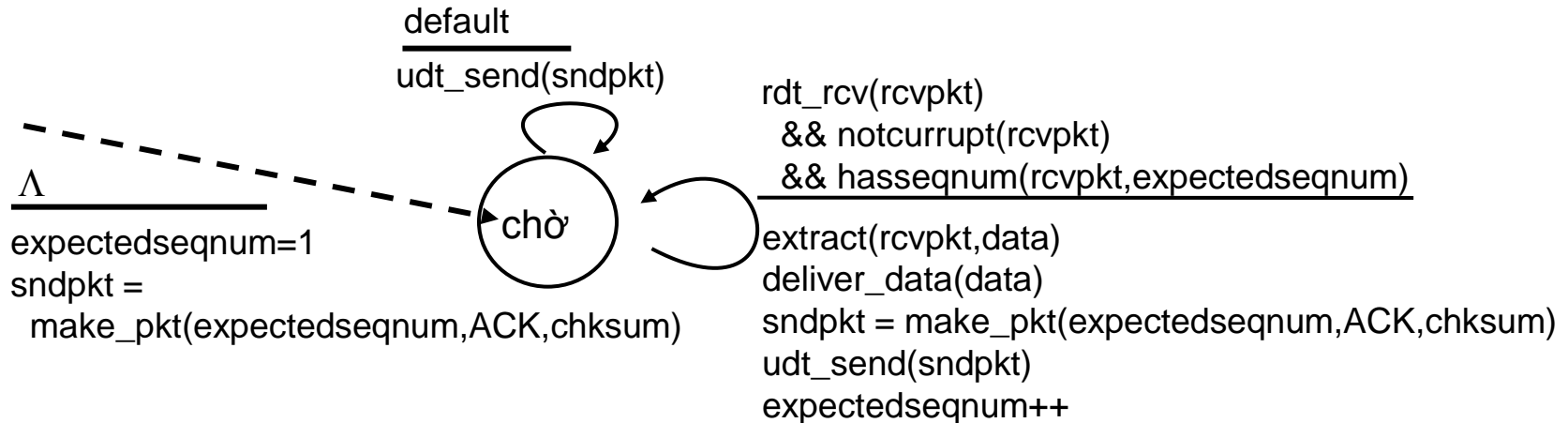


- ❑ ACK(n): ACKs tất cả các gói đến, chứa số thứ tự n – “ACK tích lũy”
  - có thể nhận các ACK trùng lặp (xem bên nhận)
- ❑ định thì cho mỗi gói trên đường truyền
- ❑ *timeout(n)*: gửi lại gói n và tất cả các gói có số thứ tự cao hơn trong cửa sổ

# GBN: bên gửi mở rộng FSM



# GBN: bên nhận mở rộng FSM



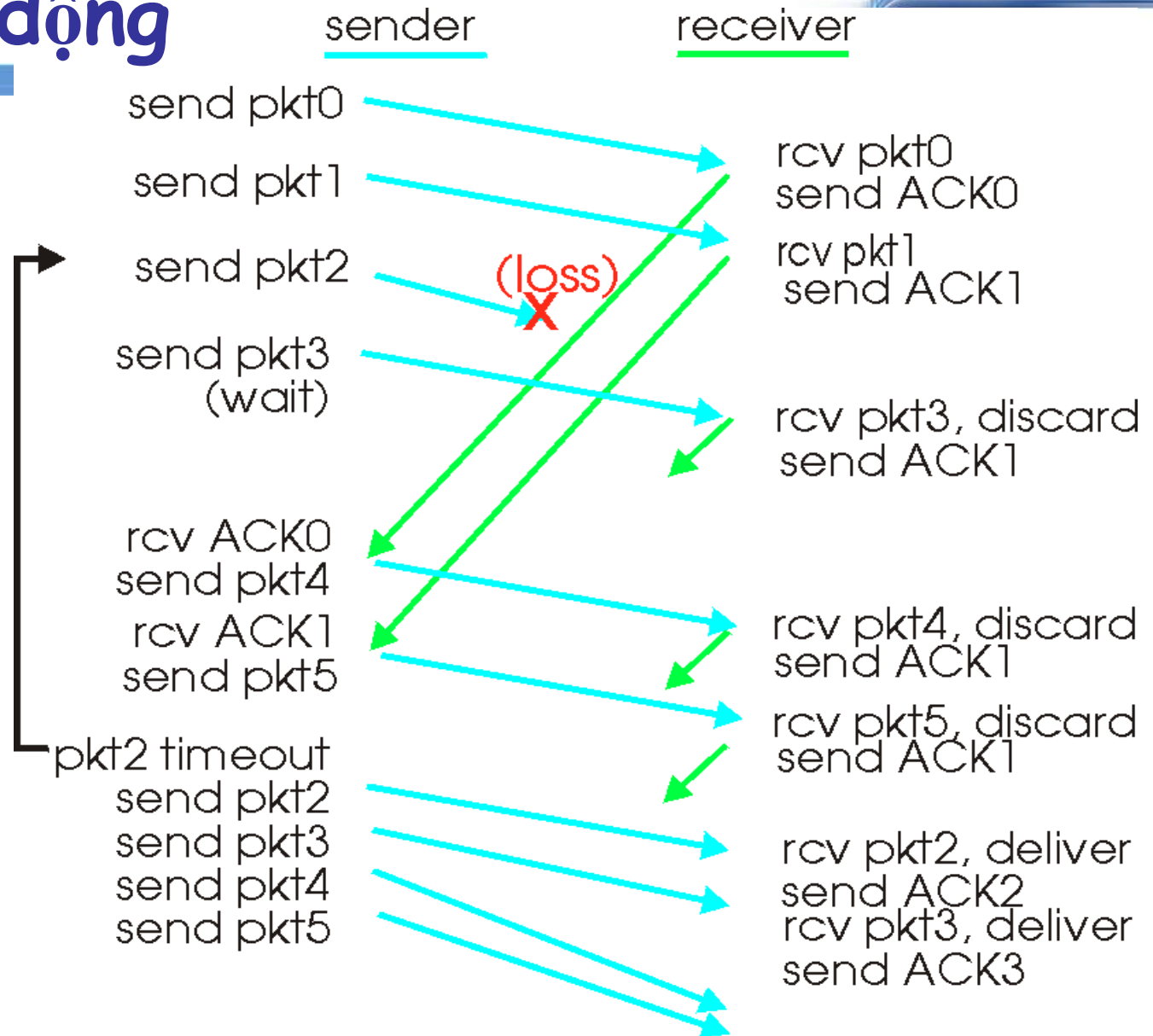
**ACK-duy nhất:** luôn luôn gửi ACK cho gói đã nhận đúng, với số thứ tự xếp hạng cao nhất

- có thể sinh ra các ACK trùng nhau
- chỉ cần nhớ **expectedseqnum**

□ gói không theo thứ tự:

- hủy -> **không nhận vào bộ đệm!**
- gửi lại ACK của gói đúng thứ tự lớn nhất

# GBN hoạt động

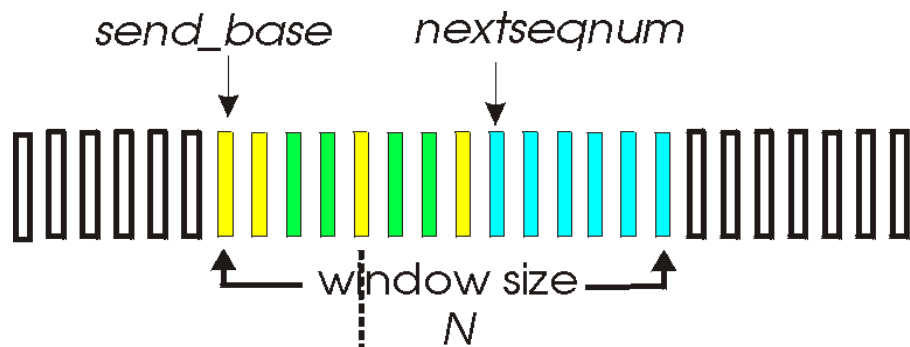




# Lặp có lựa chọn

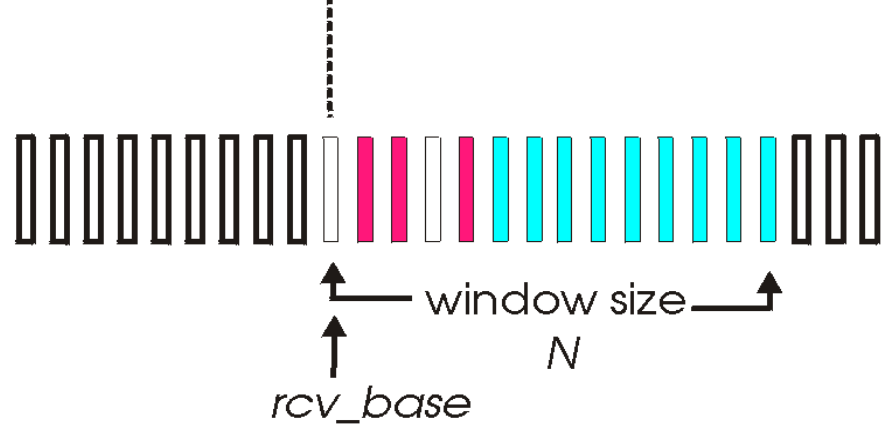
- ❑ bên nhận thông báo đã nhận đúng tất cả từng gói một
  - đệm (buffer) các gói nếu cần thiết
- ❑ bên gửi chỉ gửi lại các gói nào không nhận được ACK
  - bên gửi định thì đối với mỗi gói không gửi ACK
- ❑ cửa sổ bên gửi
  - N số thứ tự liên tục
  - hạn chế số thứ tự các gói không gửi ACK

# Lặp có lựa chọn: các cửa sổ gửi, nhận



- Green box: already ack'ed
- Yellow box: sent, not yet ack'ed
- Cyan box: usable, not yet sent
- Black box: not usable

(a) sender view of sequence numbers



- Pink box: out of order (buffered) but already ack'ed
- White box: Expected, not yet received
- Cyan box: acceptable (within window)
- Black box: not usable

(b) receiver view of sequence numbers

# Lắp có lựa chọn

## Gửi

### Gửi dữ liệu từ lớp trên:

- nếu số thứ tự kế tiếp sẵn sàng trong cửa sổ, gửi gói

### timeout(n):

- gửi lại gói n, tái khởi tạo bộ định thì

### ACK(n) trong [sendbase, sendbase+N]:

- đánh dấu gói n là đã nhận
- Nếu n là STT bé nhất chưa biên nhận, dịch chuyển cửa sổ lên STT gói tin bé nhất chưa biên nhận kế tiếp

## Nhận

### gói n trong [rcvbase, rcvbase+N-1]

- gửi ACK(n)
- không thứ tự: lưu vào bộ đệm (buffer)
- Đúng thứ tự: chuyển tất cả dữ liệu đã nhận đúng thứ tự lên tầng bên trên

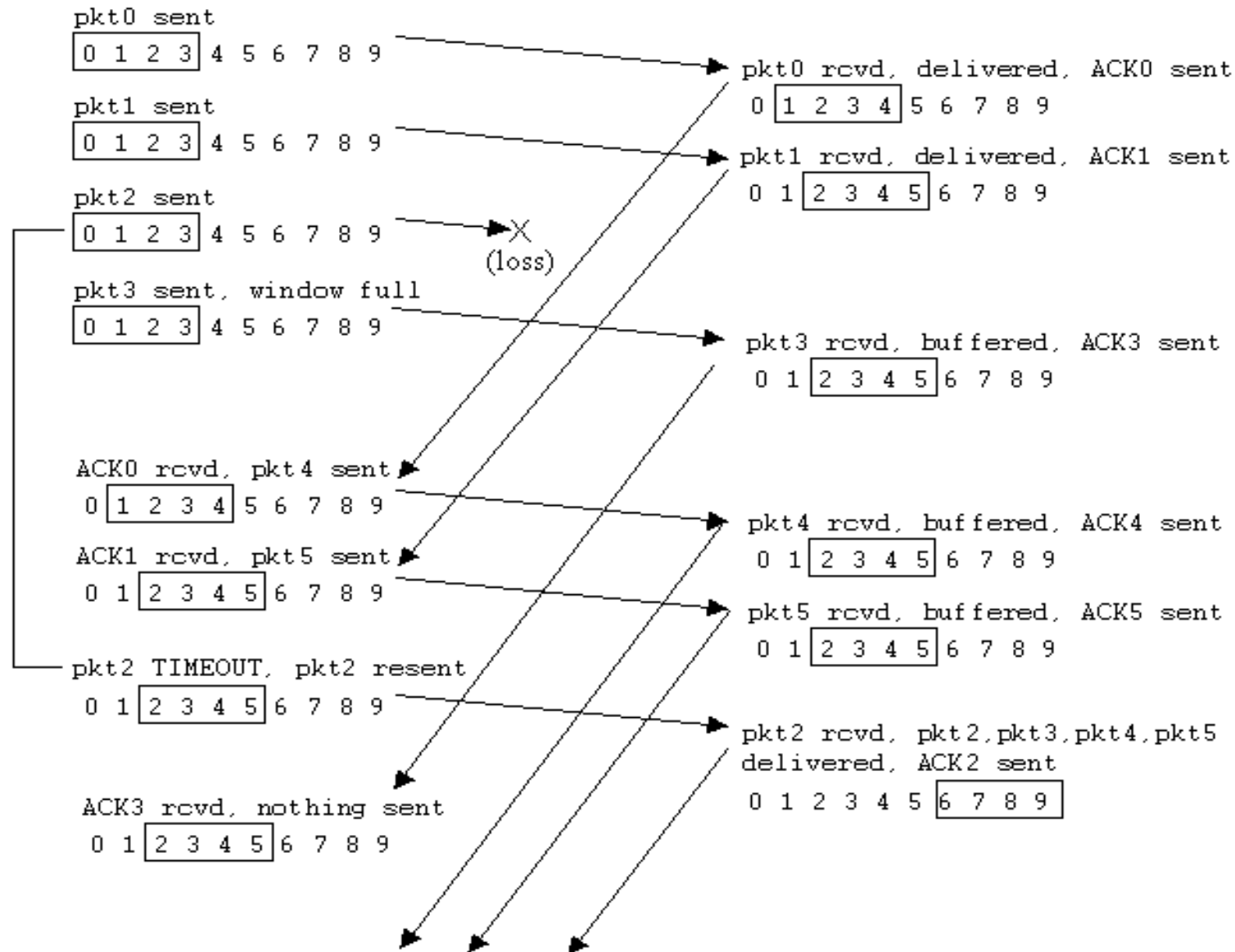
### gói n trong [rcvbase-N, rcvbase-1]

- ACK(n)

### ngược lại:

- lờ đi

# Hoạt động của lặp có lựa chọn



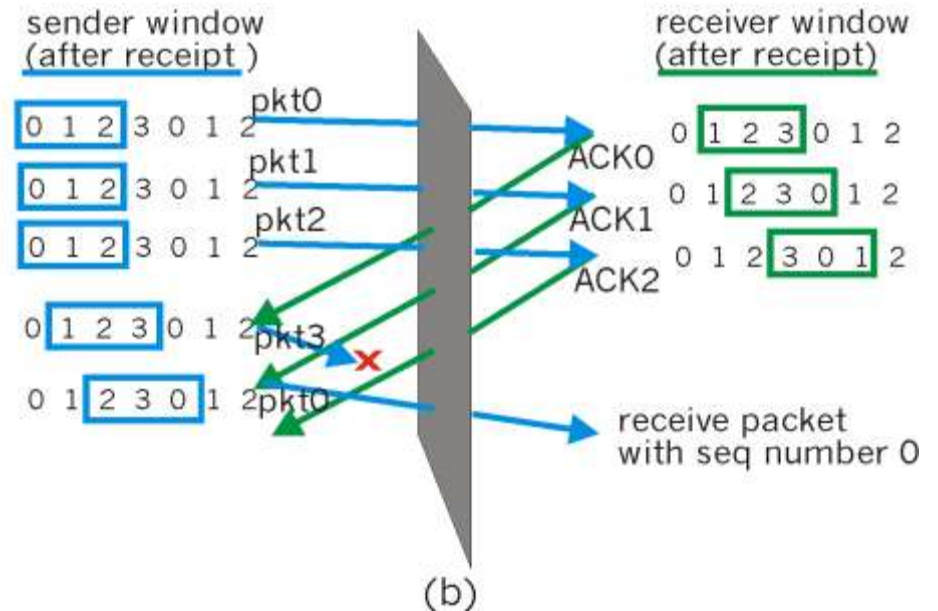
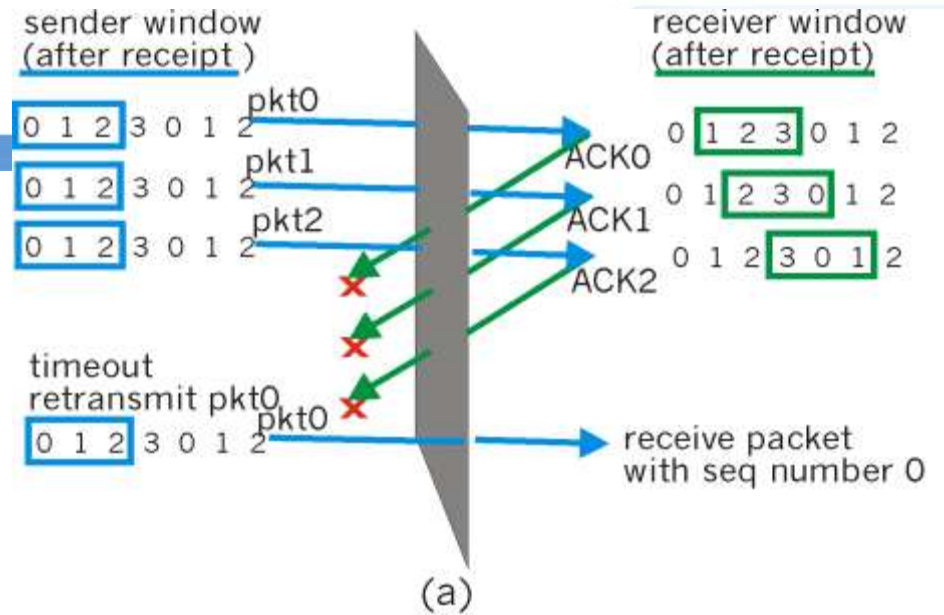
# Lắp có lựa chọn: tình hợp khó giải quyết

Ví dụ:

- ❑ Số thứ tự: 0, 1, 2, 3
- ❑ Kích thước cửa sổ = 3

- ❑ bên nhận không thấy sự khác nhau trong 2 tình huống
- ❑ chuyển không chính xác dữ liệu trùng lặp như dữ liệu mới trong (a)

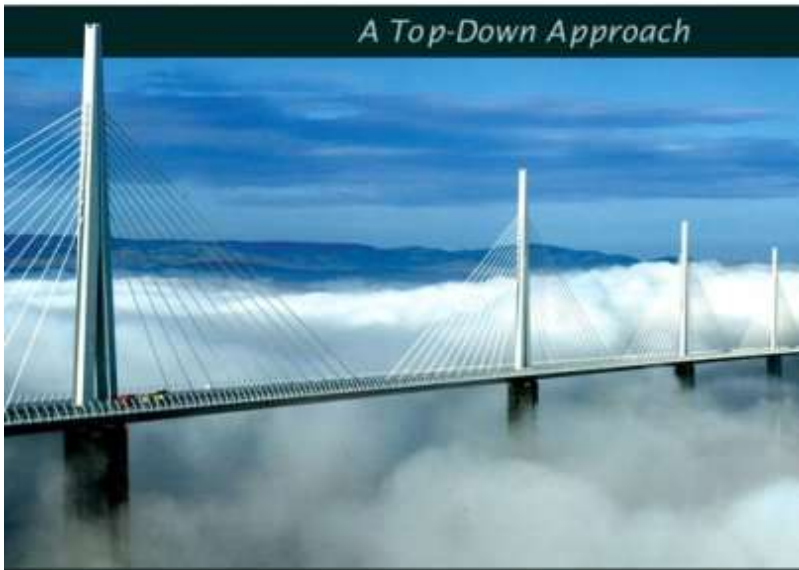
**Hỏi:** quan hệ giữa dãy số thứ tự và kích thước cửa sổ?



## 3.5 Vận chuyển hướng kết nối: TCP

### COMPUTER NETWORKING FIFTH EDITION

*A Top-Down Approach*



KUROSE • ROSS

# TCP: Tổng quan RFCs: 793, 1122, 1323, 2018, 2581

## □ point-to-point:

- một bên gửi, một bên nhận

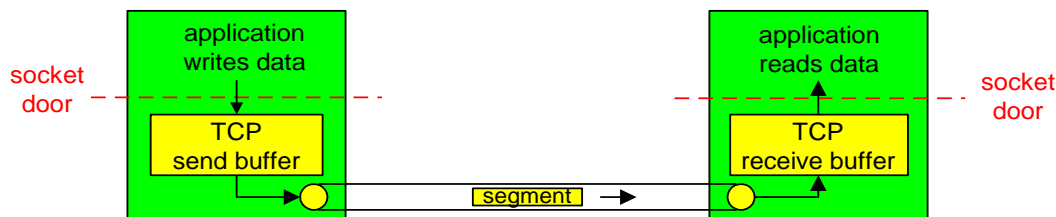
## □ tin cậy, dòng byte có thứ tự:

- không "ranh giới thông điệp"

## □ Đường ống - pipelining:

- TCP điều khiển lưu lượng và tắc nghẽn, thiết lập kích thước cửa sổ

## □ các bộ đệm gửi & nhận



## □ dữ liệu full duplex:

- luồng dữ liệu đi 2 chiều trong cùng một kết nối
- MSS: maximum segment size - kích thước đoạn tối đa

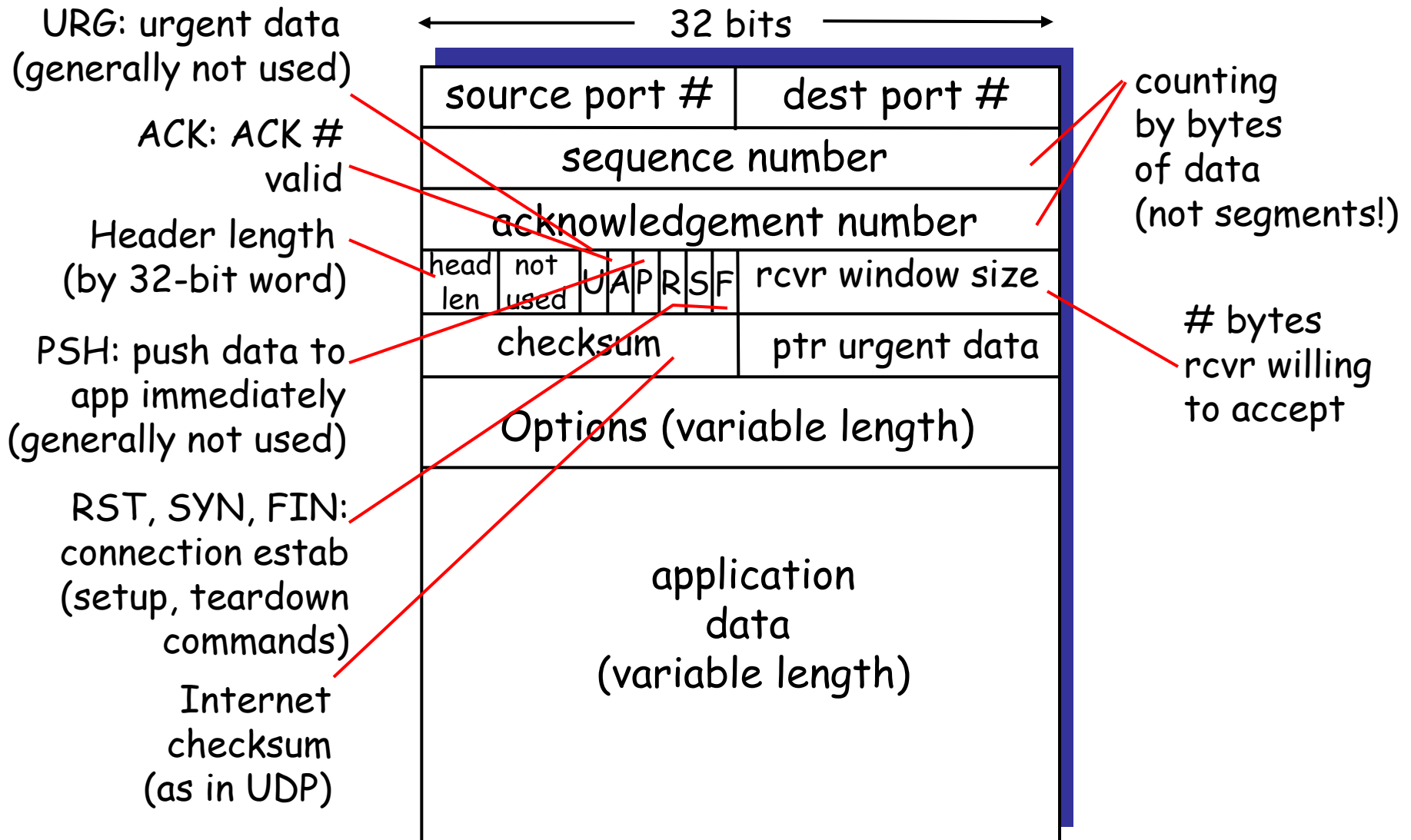
## □ hướng kết nối:

- bắt tay (trao đổi các thông điệp điều khiển) trạng thái bên gửi, bên nhận trước khi trao đổi dữ liệu

## □ điều khiển lưu lượng:

- bên gửi sẽ không lấn át bên nhận

# TCP: cấu trúc đoạn





# Các số thứ tự TCP và ACK

## Các số thứ tự:

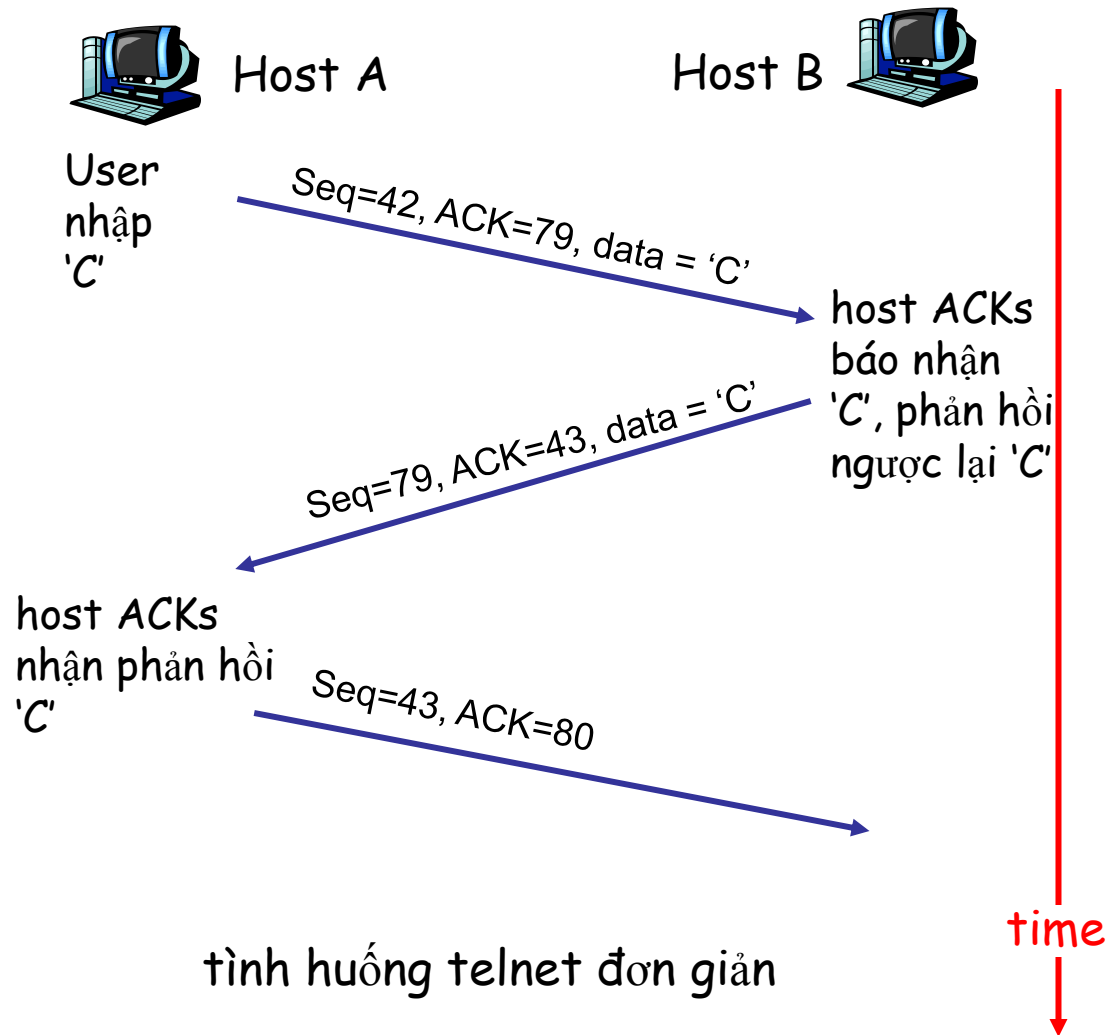
- dòng byte "đánh số" byte đầu tiên trong dữ liệu của đoạn

## các ACK:

- số thứ tự của byte kế tiếp được chờ đợi từ phía bên kia
- ACK tích lũy

**Hỏi:** làm thế nào bên nhận quản lý các đoạn không thứ tự

- Trả lời: TCP không đề cập, tùy thuộc người hiện thực



# TCP Round Trip Time và Timeout

Hỏi: Làm thế nào để thiết lập giá trị TCP timeout?

- ❑ dài hơn RTT
  - khác với RTT
- ❑ quá ngắn: timeout sớm
  - truyền lại không cần thiết
- ❑ quá dài: phản ứng chậm đối với việc mất mát gói

Hỏi: Làm thế nào để thiết lập RTT?

- ❑ **SampleRTT**: thời gian được đo từ khi truyền đoạn đến khi báo nhận ACK
  - lờ đi việc truyền lại
- ❑ **SampleRTT** sẽ thay đổi mạnh nên cần ước lượng RTT "mượt hơn"
  - tính trung bình một số giá trị đo được gần đó, không chỉ **SampleRTT** hiện tại

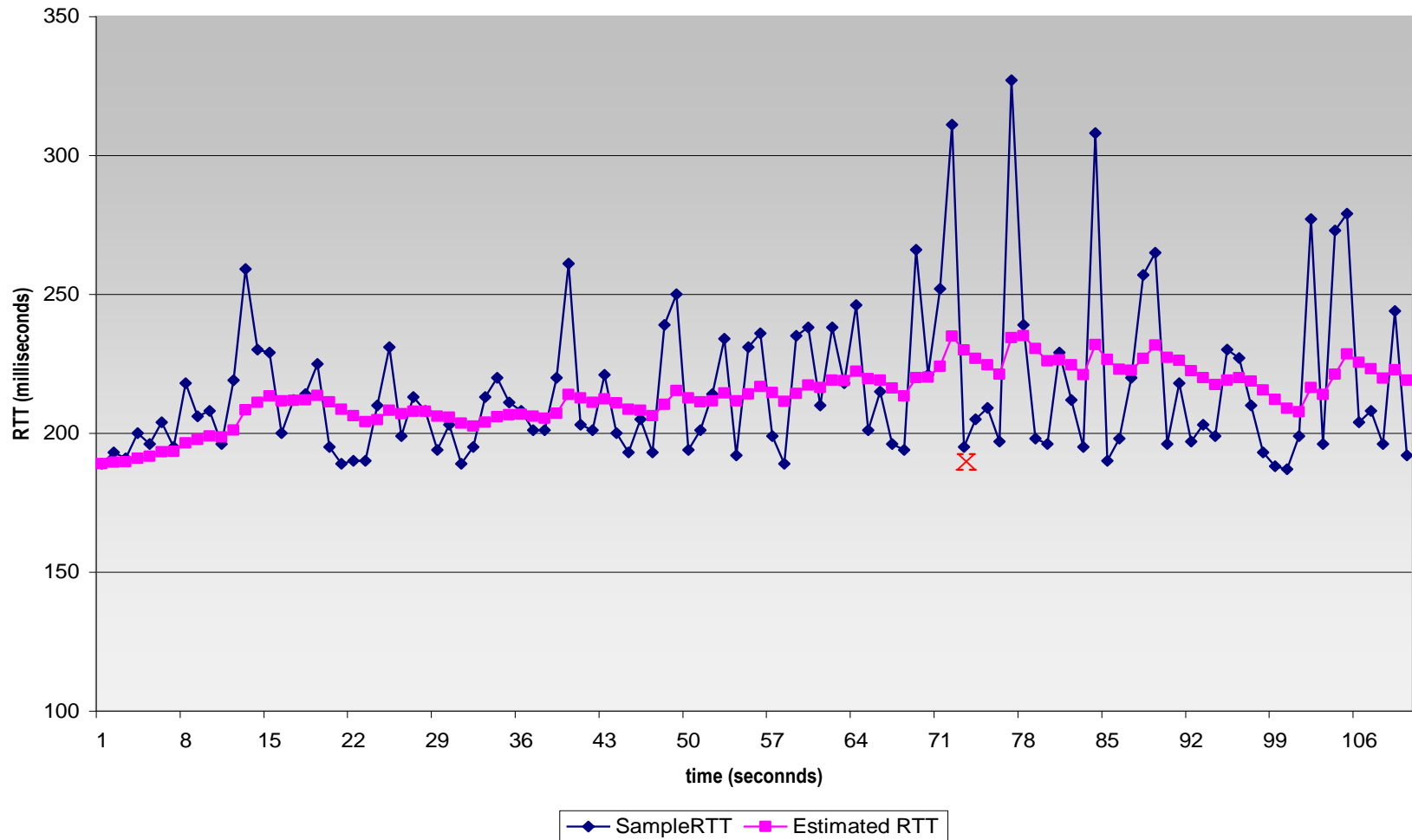
# TCP Round Trip Time và Timeout

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

□ giá trị đặc trưng:  $\alpha = 0.125$

# Ví dụ đánh giá RTT:

RTT: gaia.cs.umass.edu to fantasia.eurecom.fr



# TCP Round Trip Time và Timeout

## Thiết lập timeout

- ❑ EstimatedRTT cộng "hệ số dự trữ an toàn"
  - > sự biến thiên lớn trong EstimatedRTT -> hệ số dự trữ an toàn lớn hơn

- ❑ ước lượng đầu tiên về sự biến thiên của SampleRTT từ EstimatedRTT:

$$\text{DevRTT} = (\beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$$

(tiêu biểu  $\beta = 0.25$ )

Sau đó thiết lập timeout interval:

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$

# TCP: truyền dữ liệu tin cậy

- ❑ TCP tạo dịch vụ rdt trên dịch vụ không tin cậy IP

## Sử dụng các cơ chế

- ❑ các đoạn Pipelined
- ❑ các ACK tích lũy
- ❑ TCP dùng bộ định thời để truyền lại%

- ❑ Truyền lại được kích hoạt bởi:

- các sự kiện timeout
- các ack trùng lặp

- ❑ lúc đầu khảo sát các bên gửi TCP đơn giản:

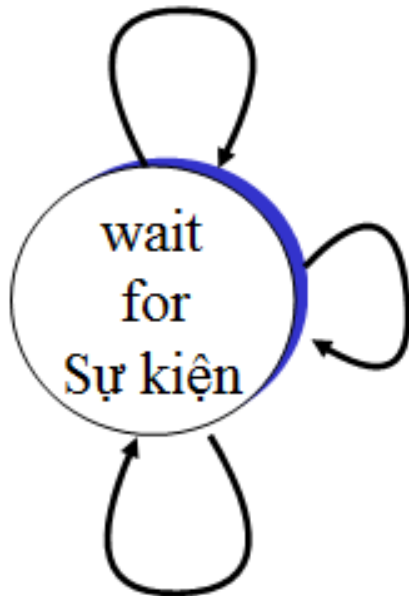
- lờ đi các ack trùng lặp
- lờ đi điều khiển lưu lượng, điều khiển tắc nghẽn

# Các sự kiện TCP tại bên gửi

Sự kiện: nhận dữ liệu  
từ phía bên trên

---

Tạo và Gửi segment



Sự kiện: timer timeout cho  
segment có STT là y

---

Gửi lại segment

Sự kiện: Nhận biên nhận cho  
gói có STT là y

---

Xử lý ACK

NextSeqNum = InitialSeqNum

SendBase = InitialSeqNum

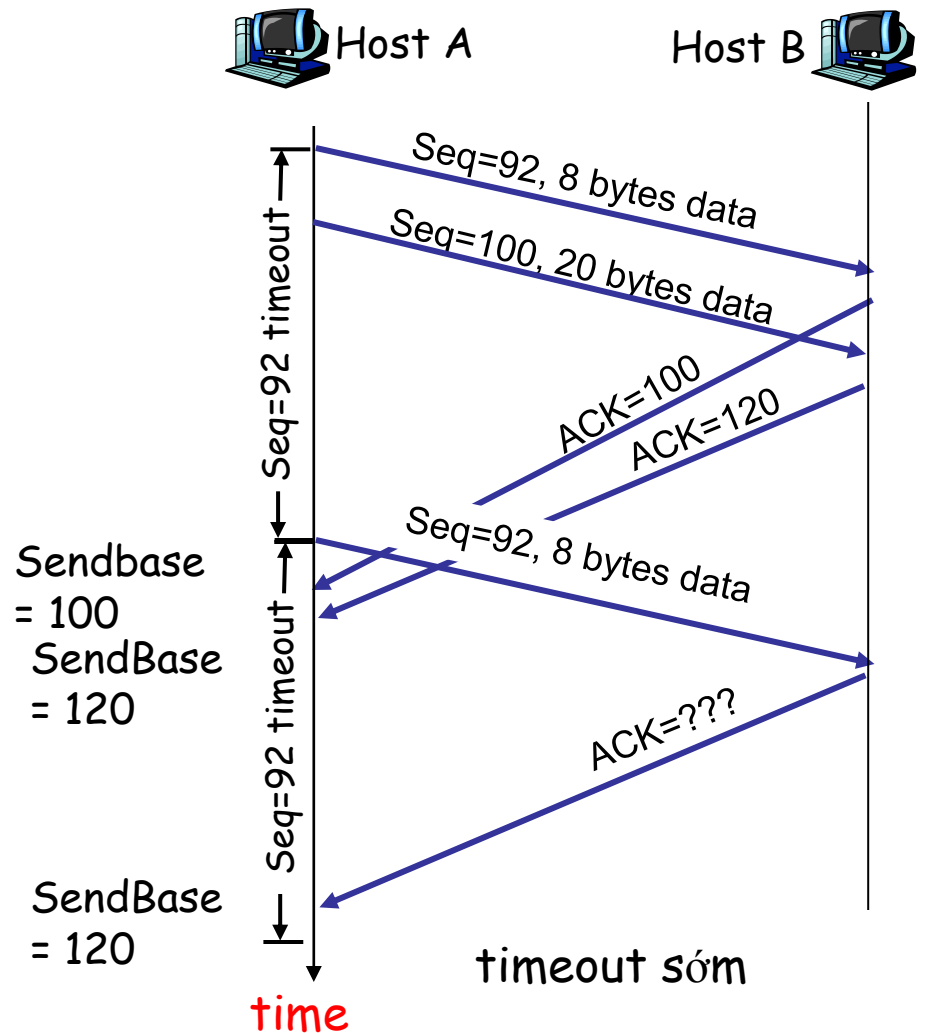
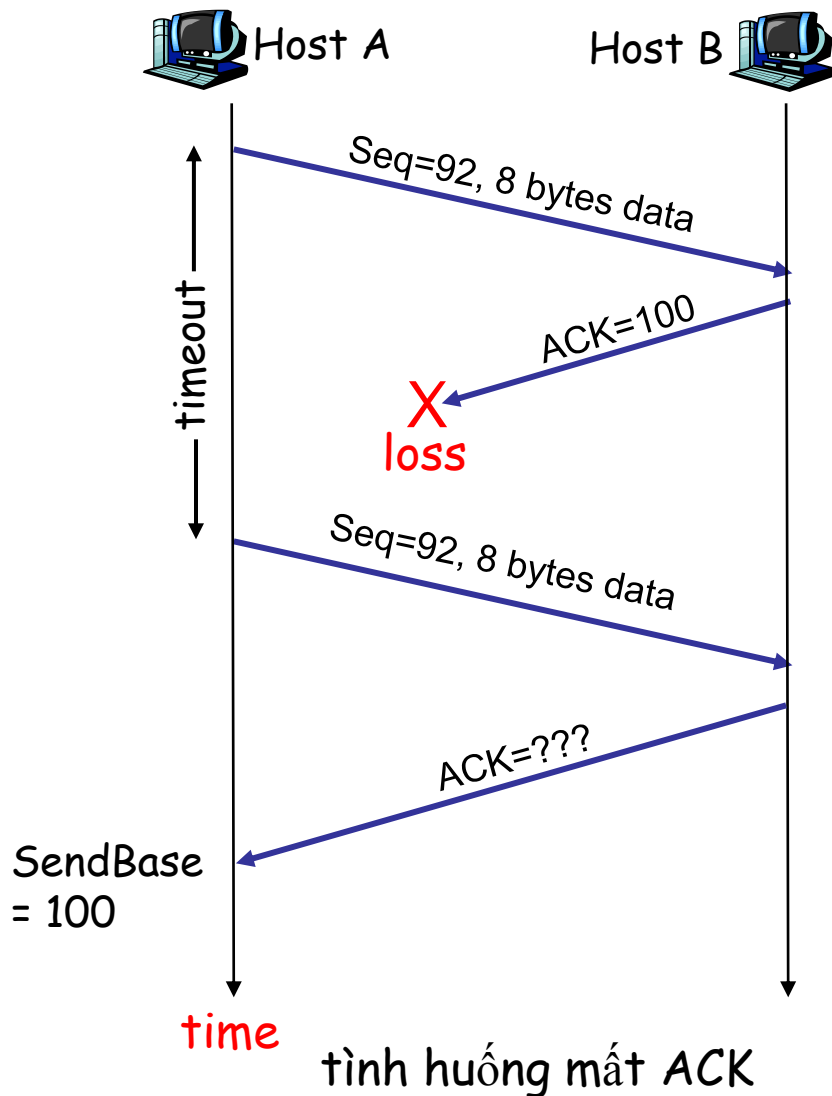
# TCP

## bên gửi (đơn giản)

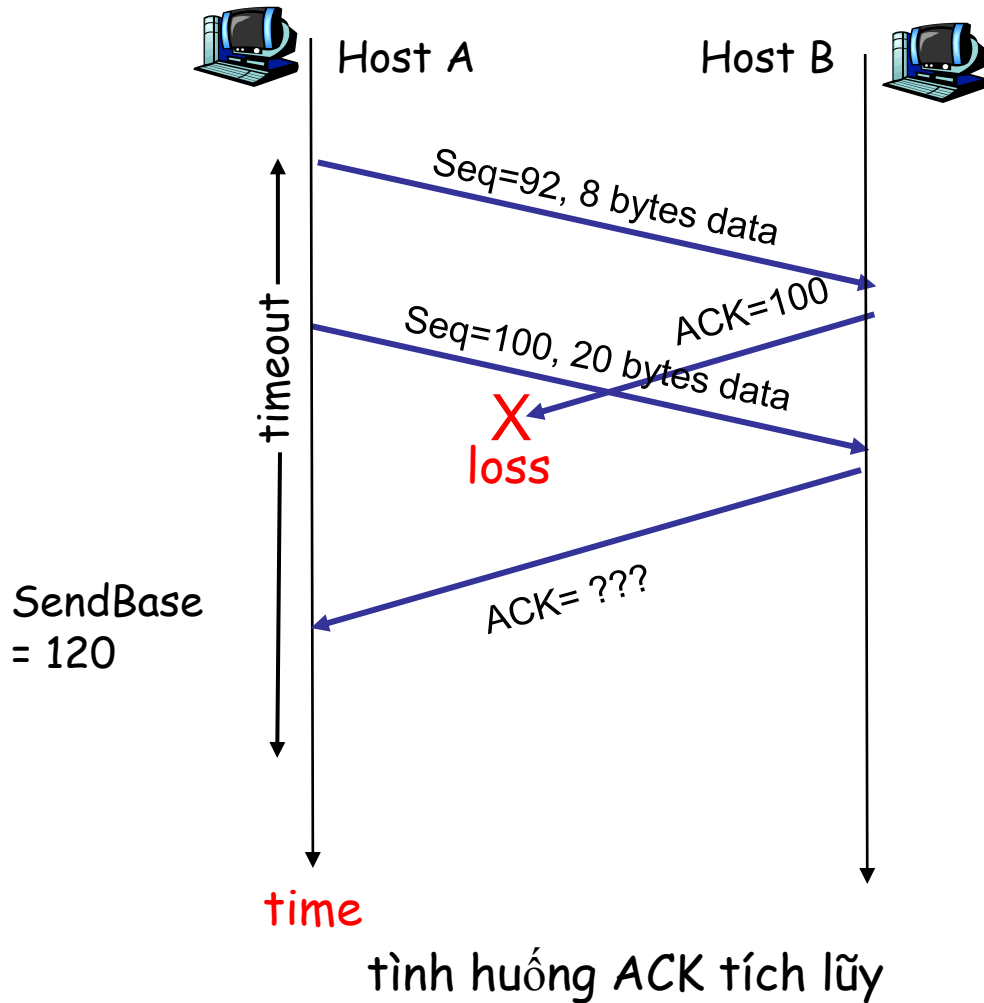
```
loop (forever) {  
  switch(event)  
  event: data received from application above  
    create TCP segment with sequence number NextSeqNum  
    if (timer currently not running)  
      start timer  
    pass segment to IP  
    NextSeqNum = NextSeqNum + length(data)  
  event: timer timeout  
    retransmit not-yet-acknowledged segment with  
      smallest sequence number  
    start timer  
  event: ACK received, with ACK field value of y  
    if (y > SendBase) {  
      SendBase = y  
      cancel the timer for sendbase  
      if (there are currently not-yet-acknowledged segments)  
        start timer  
    }  
} /* end of loop forever */
```



# TCP: các tình huống truyền lại



# TCP: các tình huống truyền lại (tt)



# TCP tại bên nhận [RFC 1122, RFC 2581]

## Sự kiện tại bên nhận

## TCP bên nhận hành động

Đoạn đến với đúng số thứ tự mong muốn.

ACK trễ. Chờ đến 500ms cho đoạn kế tiếp. Nếu không có đoạn kế tiếp, gửi ACK

Đoạn đến với đúng số thứ tự mong muốn. Một đoạn khác đang chờ ACK

Gửi ngay một ACK tích lũy


Các đoạn đến không thứ tự lớn hơn số thứ tự đoạn mong muốn  
Có khoảng trống


Gửi ngay **ACK trùng lặp**, chỉ thị số thứ tự đoạn của byte kế tiếp đang mong chờ

Đoạn đến điền vào vị trí khoảng trống

Gửi ngay ACK tích lũy, là gói tin có số lớn nhất chưa được nhận


 Gói tin đến

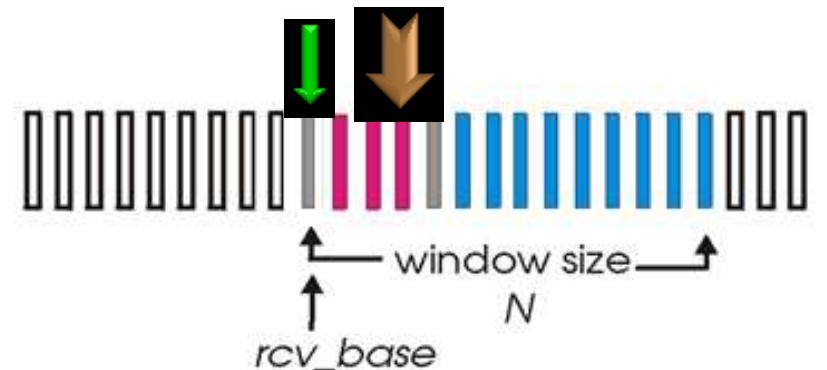
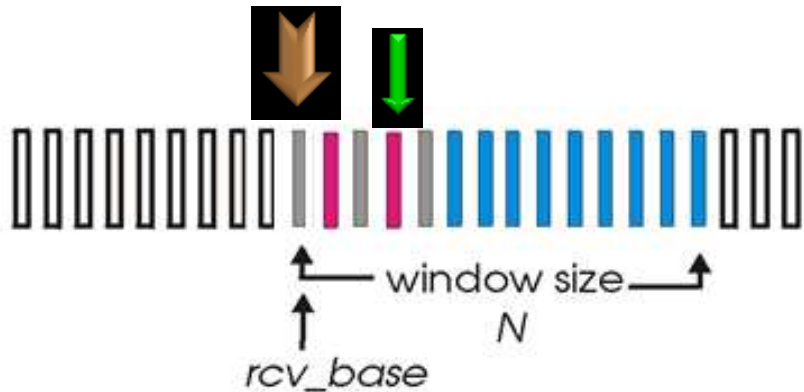
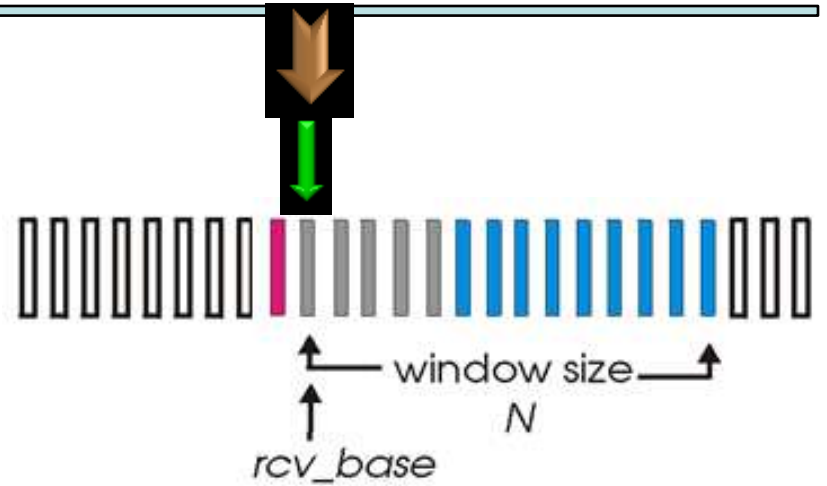
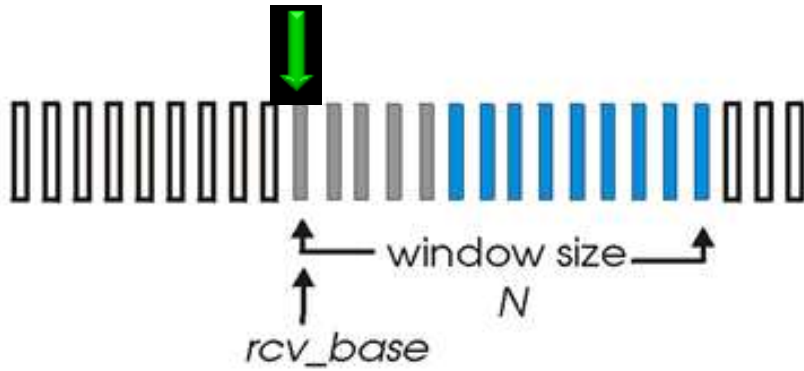
 Ack tích lũy

 out of order  
(buffered) but  
already ack'ed

 acceptable  
(within window)

 Expected, not  
yet received

 not usable



# Truyền lại nhanh

- ❑ Chu kỳ Time-out thường tương đối dài:
  - độ trễ dài trước khi gửi lại gói đã mất
- ❑ Xác nhận các đoạn đã mất bằng các ACK trùng lặp.
  - bên gửi thường gửi nhiều đoạn song song
  - Nếu đoạn bị mất, sẽ xảy ra tình trạng giống như nhiều ACK trùng nhau
- ❑ Nếu bên gửi nhận 3 ACK của cùng một dữ liệu, nó cho là đoạn sau dữ liệu đã ACK bị mất:
  - Truyền lại nhanh: gửi lại đoạn trước khi bộ định thì hết hạn

# Giải thuật truyền lại nhanh:

**sự kiện:** ACK đã nhận, với trường là y

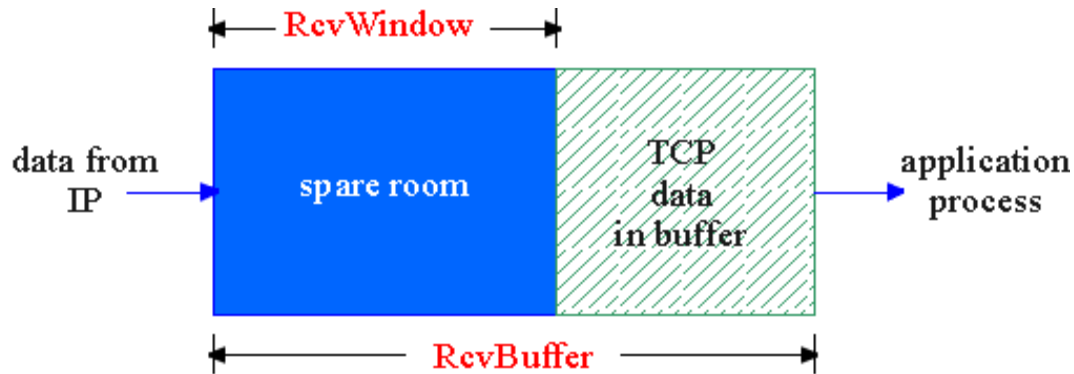
```
if (y > SendBase) {  
    SendBase = y  
    if (there are currently not-yet-acknowledged segments)  
        start timer  
}  
else {  
    increment count of dup ACKs received for y  
    if (count of dup ACKs received for y = 3) {  
        resend segment with sequence number y  
    }  
}
```

một ACK trùng lặp cho  
đoạn đã được ACK

Truyền lại nhanh

# TCP điều khiển lưu lượng

- bên nhận của kết nối TCP có một bộ đệm nhận:



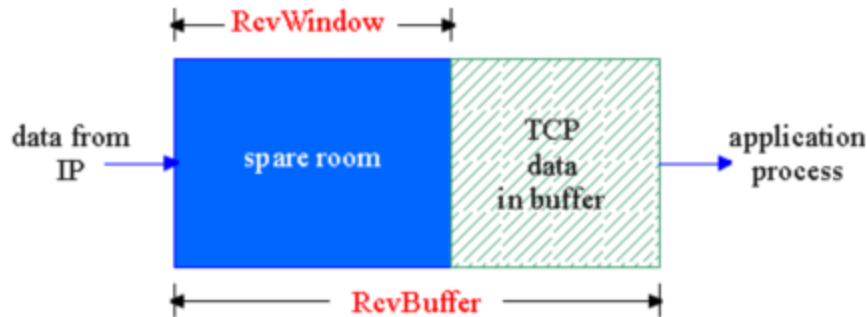
- tiến trình ứng dụng có thể chậm tại lúc đọc bộ đệm

## điều khiển lưu lượng

bên gửi sẽ không làm tràn bộ đệm vì truyền quá nhiều và quá nhanh

- dịch vụ so trùng tốc độ: so trùng tốc độ gửi với tốc độ nhận của ứng dụng

# TCP điều khiển lưu lượng



r Chỗ trống trong Bộ đệm  
= **RcvWindow**

source port #		dest port #	
sequence number			
acknowledgement number			
head len	not used	U	A
		P	R
		S	F
checksum		ptr urgent data	
Options (variable length)			
application data (variable length)			



# TCP quản lý kết nối

Chú ý: Bên gửi và bên nhận TCP thiết lập "kết nối" trước khi trao đổi dữ liệu

- khởi tạo các biến TCP:
  - các số thứ tự đoạn
  - thông tin các bộ đệm, điều khiển lưu lượng (như RcvWindow)

- *client*: Khởi tạo, yêu cầu kết nối

```
Socket clientSocket = new  
Socket("hostname", "port  
number");
```

- *server*: chấp nhận kết nối khi client có yêu cầu

```
Socket connectionSocket =  
welcomeSocket.accept();
```

## phương pháp bắt tay 3 bước:

Bước 1: client host gửi đoạn TCP SYN đến server

- xác định số thứ tự khởi đầu
- không phải dữ liệu

Bước 2: server host nhận SYN, trả lời với đoạn SYNACK

- server cấp phát các bộ đệm
- xác định số thứ tự khởi đầu

Bước 3: client nhận SYNACK, trả lời với đoạn ACK (có thể chứa dữ liệu)

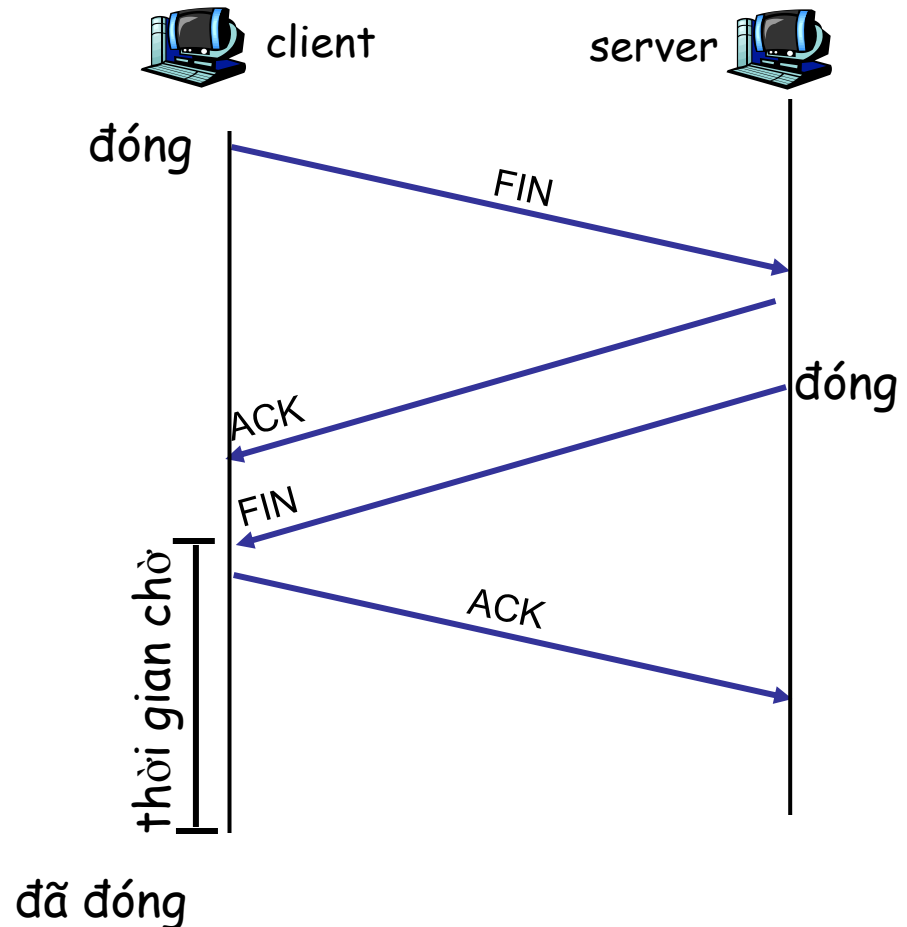
# TCP quản lý kết nối (tt)

## Đóng một kết nối:

client                  đóng                  socket:  
`clientSocket.close();`

**Bước 1:** client gửi đoạn điều khiển TCP FIN đến server

**Bước 2:** server nhận FIN, trả lời với ACK. Đóng kết nối, gửi FIN.



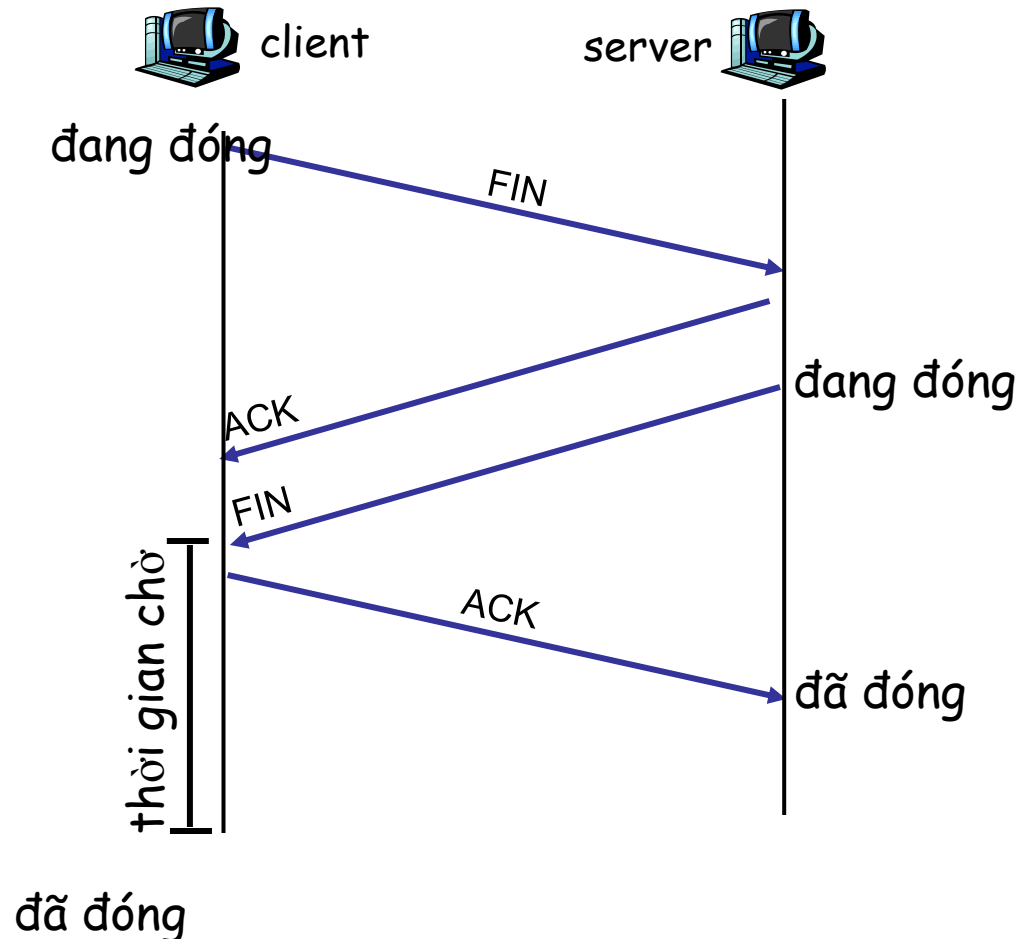
# TCP quản lý kết nối (tt)

**Bước 3:** client nhận FIN, trả lời với ACK.

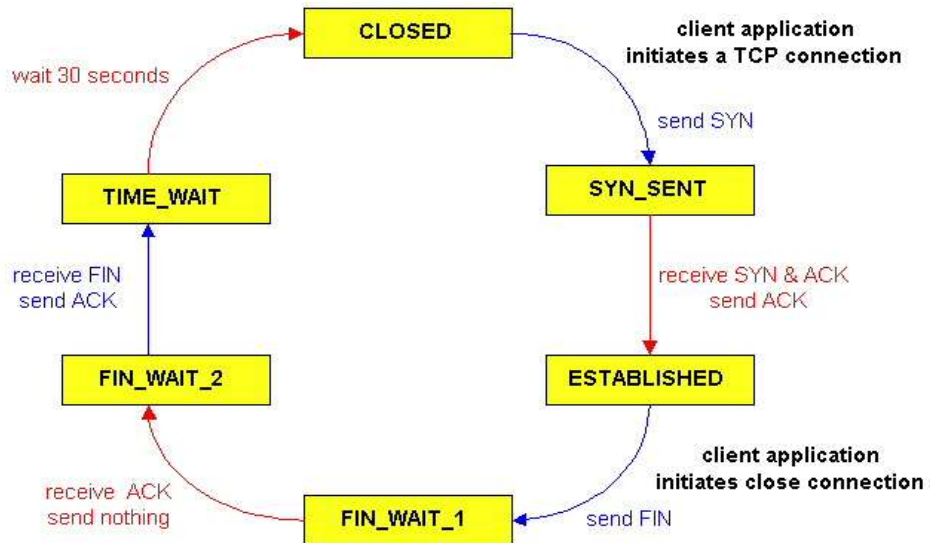
- Trong khoảng "thời gian chờ" - sẽ phản hồi với ACK để nhận các FIN

**Bước 4:** server, nhận ACK thì kết nối đã đóng thực sự đóng.

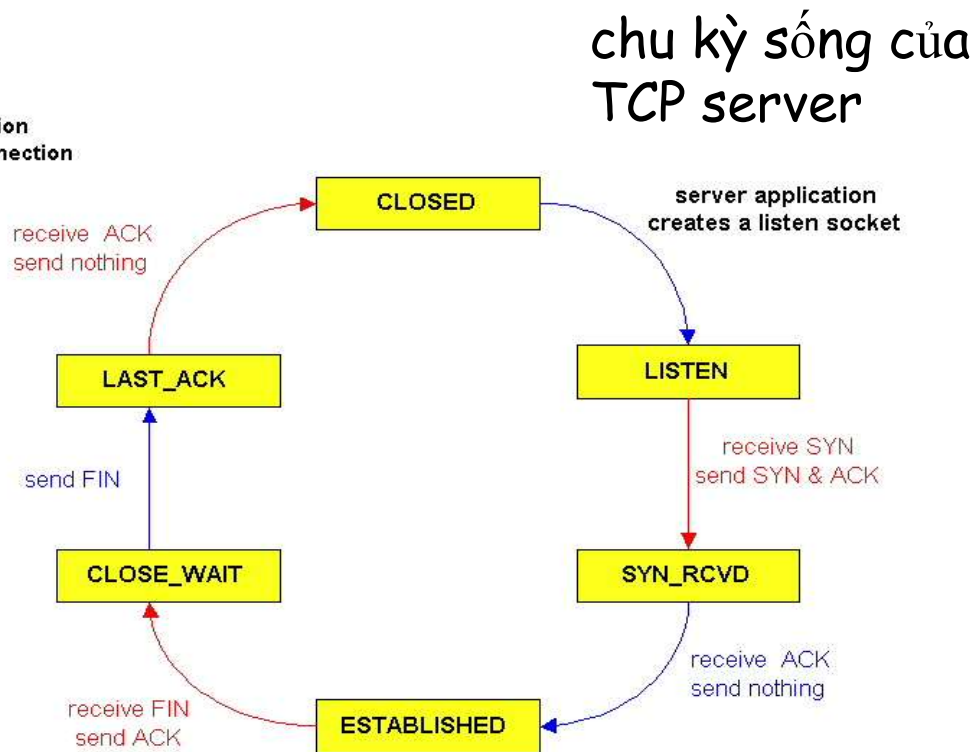
**Chú ý:** với một sửa đổi nhỏ, có thể quản lý nhiều FIN đồng thời.



# TCP quản lý kết nối (tt)

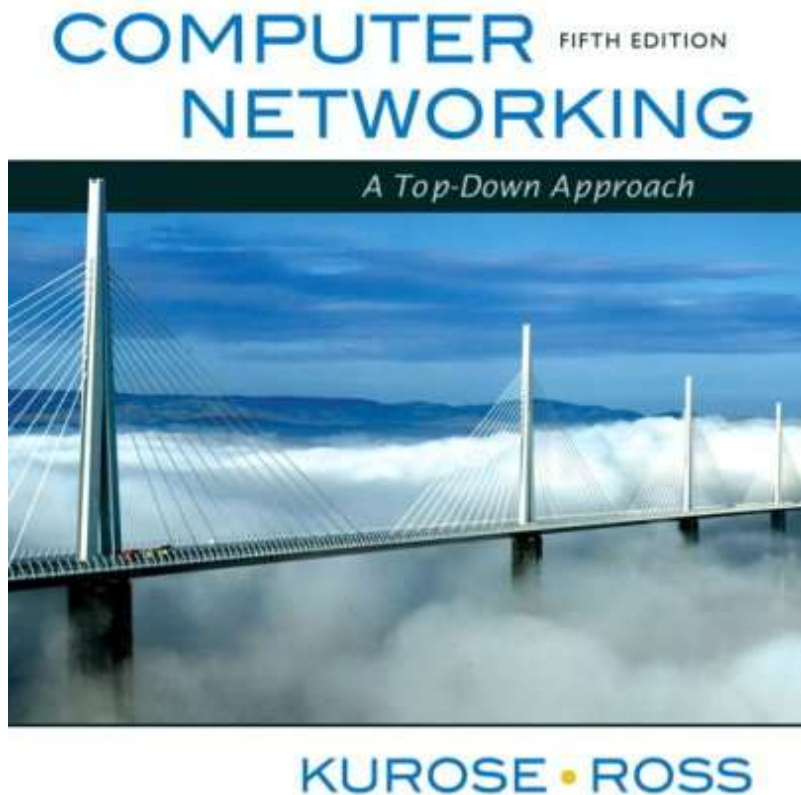


chu kỳ sống của TCP client



chu kỳ sống của TCP server

## 3.6 Các nguyên lý của điều khiển tắc nghẽn



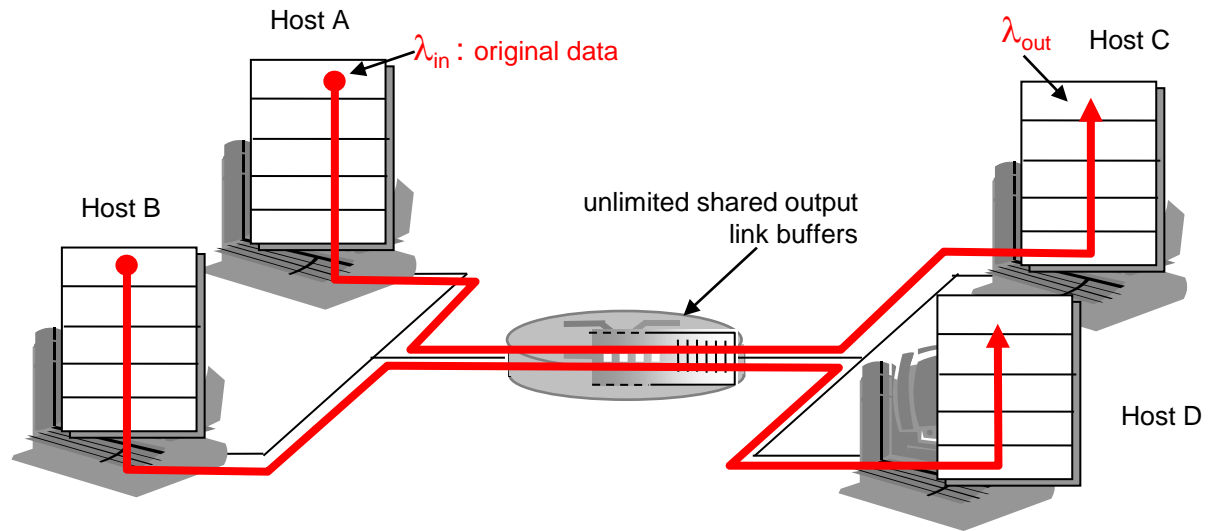
# Các nguyên lý điều khiển tắc nghẽn

## Tắc nghẽn:

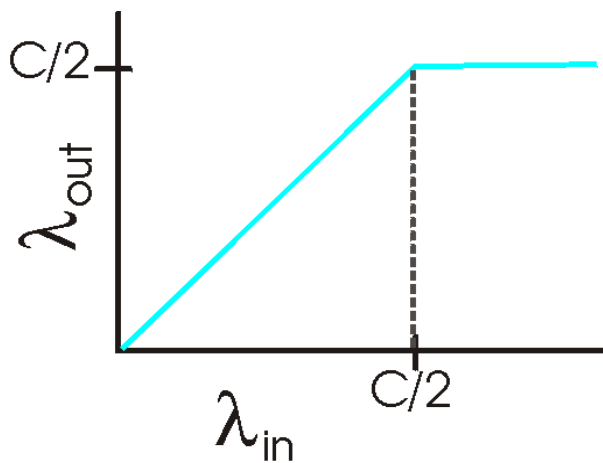
- ❑ “quá nhiều nguồn gửi quá nhanh và quá nhiều dữ liệu đến *mạng*”
- ❑ khác với điều khiển lưu lượng!
- ❑ các biểu hiện:
  - các gói bị mất (tràn bộ đệm tại các router)
  - các độ trễ quá dài (xếp hàng trong bộ đệm của router)
- ❑ là 1 trong 10 vấn đề nan giải nhất!

# Các nguyên nhân/chi phí của tắc nghẽn: tình huống 1

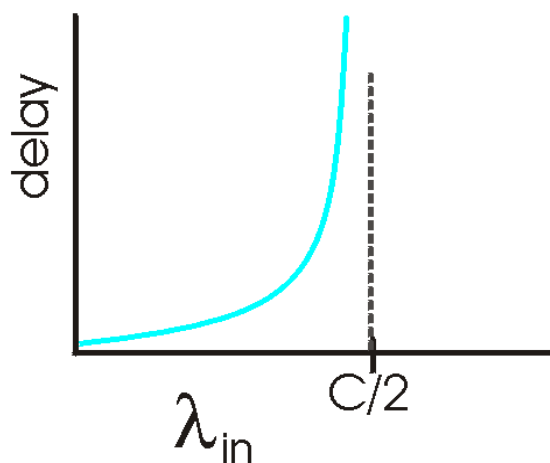
- 2 gửi, 2 nhận
- 1 router, các bộ đệm không giới hạn
- không có truyền lại



Hình A



Hình B

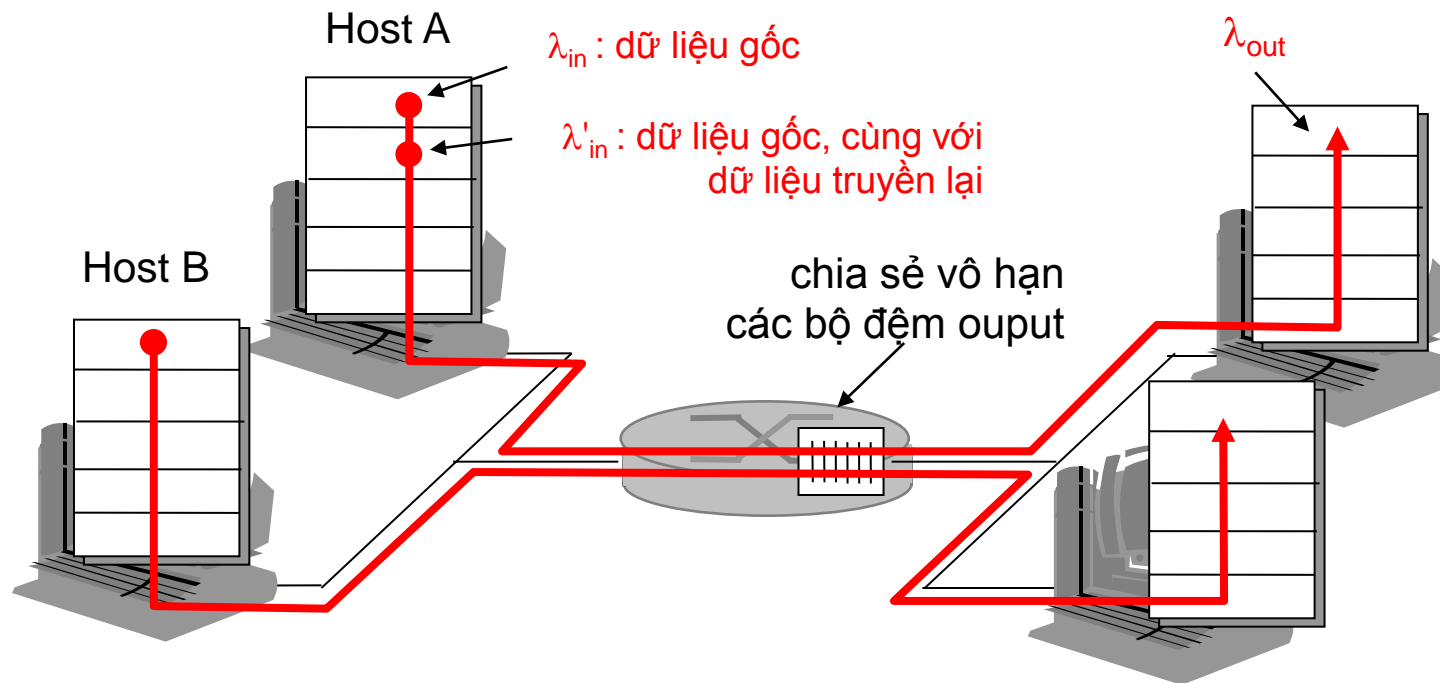


Hình C

- các độ trễ lớn hơn khi tắc nghẽn
- năng suất có thể đạt tối đa

# Các nguyên nhân/chi phí của tắc nghẽn: tình huống 2

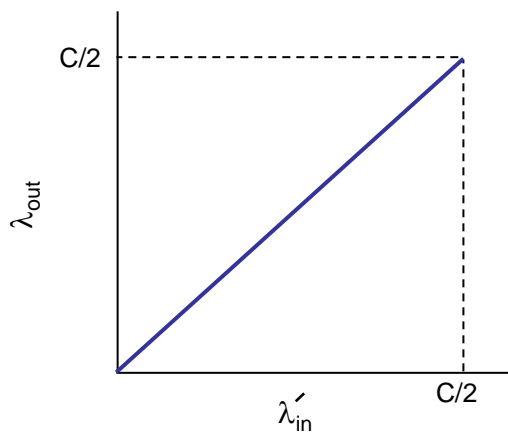
- ❑ 1 router, các bộ đệm *có giới hạn*
- ❑ bên gửi truyền lại các gói đã mất



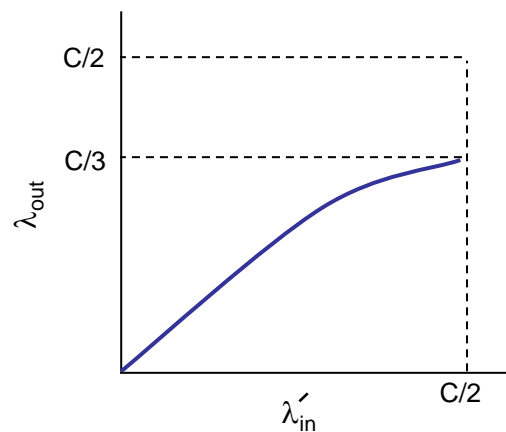


# Các nguyên nhân/chi phí của tắc nghẽn: tình huống 2

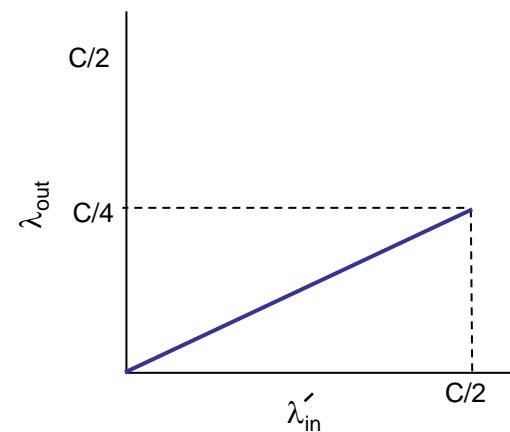
- Trường hợp lý tưởng  $\lambda_{in} = \lambda_{out}$
- truyền lại khi mất gói tin:  $\lambda'_{in} > \lambda_{out}$
- truyền lại vì trễ (không mất) làm cho  $\lambda'_{in} > \lambda_{out}$



a.



b.



c.

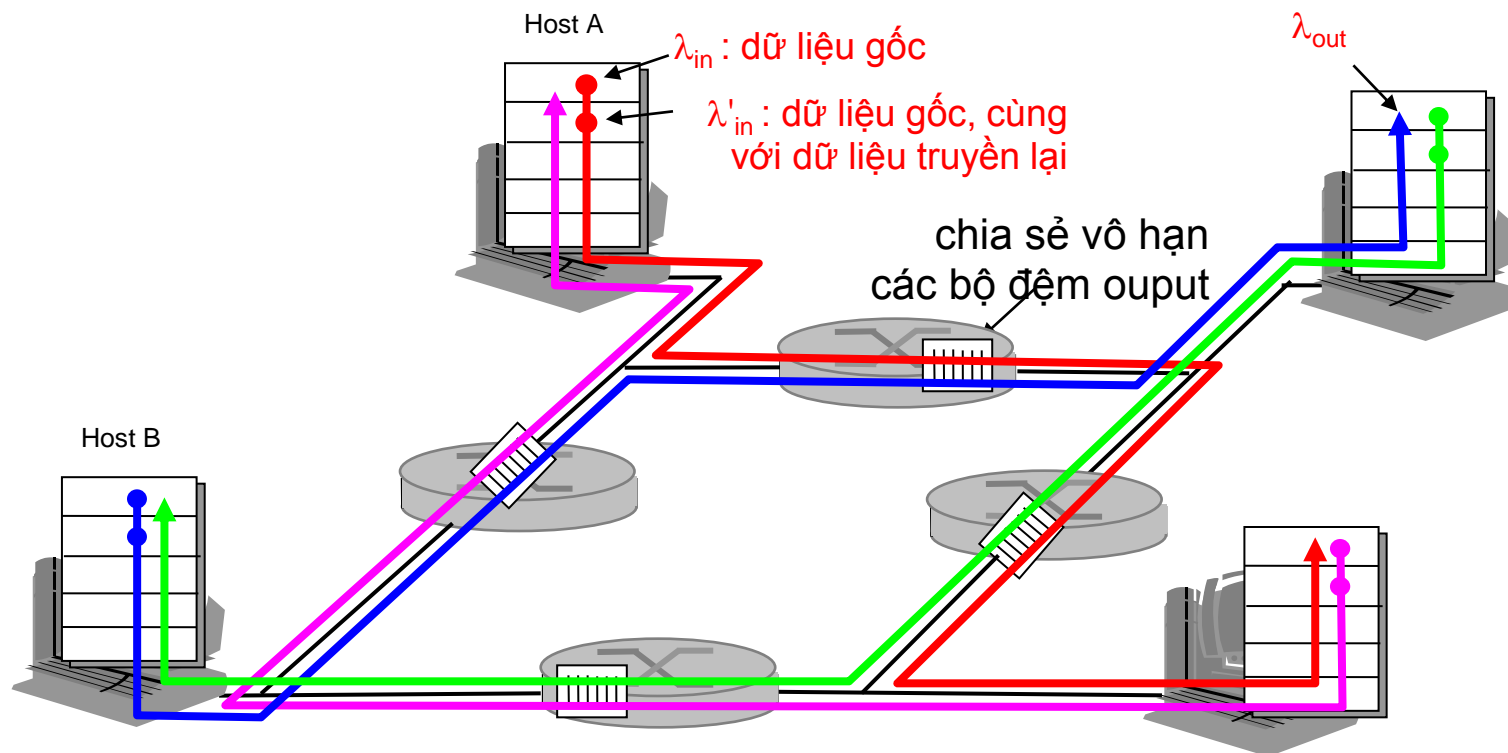
“các chi phí” của tắc nghẽn:

- nhiều việc (truyền lại)
- các truyền lại không cần thiết: liên kết nhiều bản sao của gói

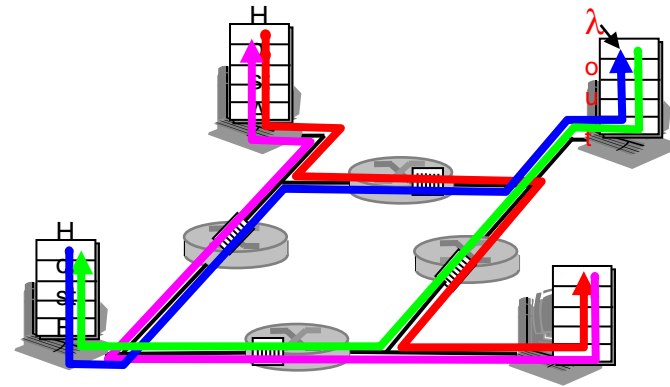
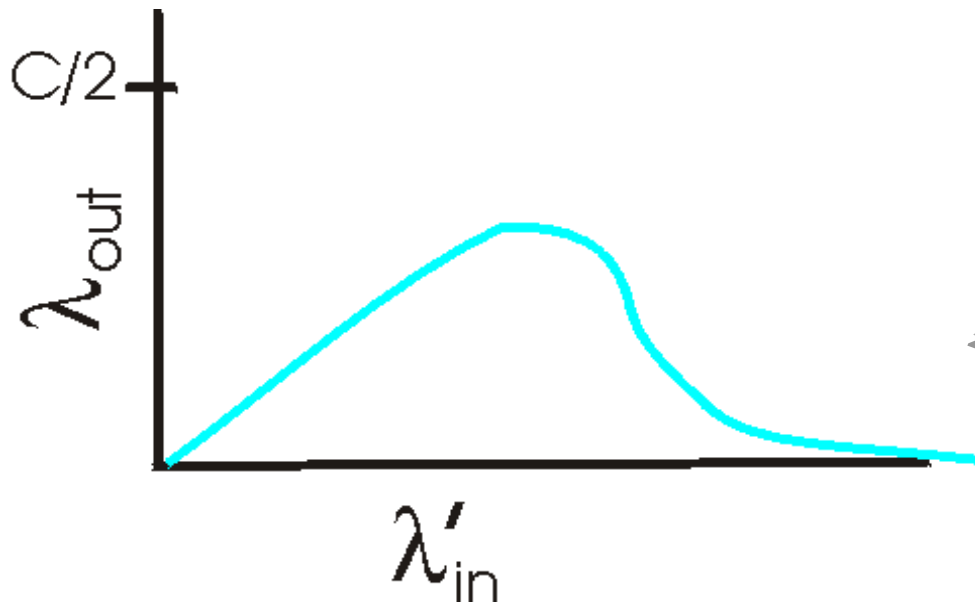
# Các nguyên nhân/chi phí của tắc nghẽn: tình huống 3

- ❑ 4 người gửi
- ❑ các đường qua nhiều hop
- ❑ timeout/truyền lại

**Hỏi:** điều gì xảy ra nếu  $\lambda_{in}$  và  $\lambda'_{in}$  tăng lên?



# Các nguyên nhân/chi phí của tắc nghẽn: tình huống 3



# Các cách tiếp cận đối với điều khiển tắc nghẽn

2 cách:

## điều khiển tắc nghẽn đầu - cuối

- ❑ không có phản hồi rõ ràng từ mạng
- ❑ tắc nghẽn được suy ra từ việc quan sát các hệ thống đầu cuối có mất mát, trễ
- ❑ tiếp cận được quản lý bởi TCP

## điều khiển tắc nghẽn có sự hỗ trợ của mạng:

- ❑ các router cung cấp phản hồi về các hệ thống đầu cuối
  - 1 bit duy nhất chỉ thị tắc nghẽn
  - tốc độ sẽ gửi được xác định rõ ràng

# Ví dụ: điều khiển tắc nghẽn ATM ABR

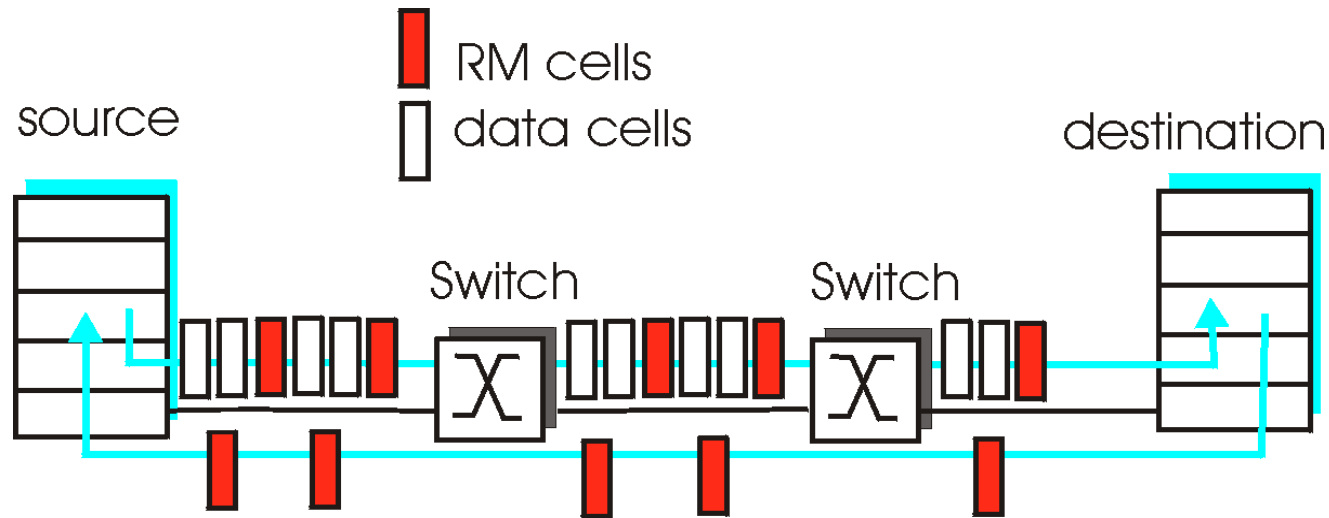
## ABR: available bit rate:

- ❑ "dịch vụ mềm dẻo"
- ❑ nếu đường truyền "tốt":
  - bên gửi sẽ dùng tăng tốc
- ❑ nếu đường gửi tắc nghẽn:
  - bên gửi điều tiết với tốc độ tối thiểu

## RM (resource management):

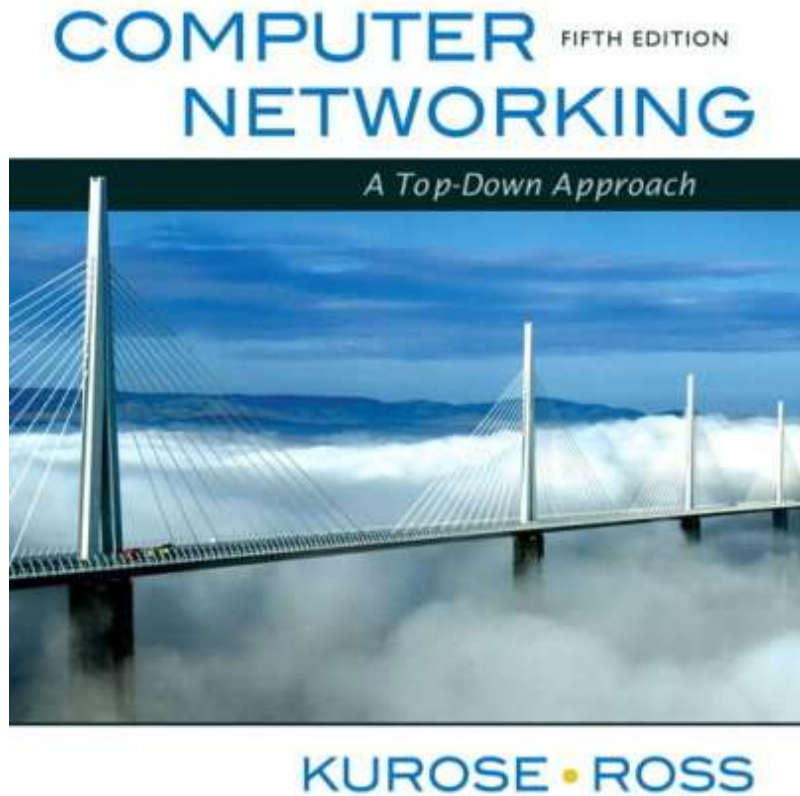
- ❑ gửi bởi bên gửi, rải rác với các ô dữ liệu
- ❑ các bit trong ô thiết lập bởi các switch
  - bit NI : không tăng tốc độ (tắc nghẽn nhẹ)
  - bit CI : tắc nghẽn rõ rệt
- ❑ Các ô RM được trả về bên gửi từ bên nhận với nguyên vẹn các bit

# Ví dụ: điều khiển tắc nghẽn ATM ABR



- trường 2-byte ER (explicit rate) trong tế bào RM
  - switch đã tắc nghẽn có thể gán giá trị ER thấp hơn
  - tốc độ gửi do đó có thể được hỗ trợ tối đa trên đường
- EFCI bit trong các ô dữ liệu: được cài giá trị 1 trong switch đã tắc nghẽn
  - nếu tế bào dữ liệu đứng trước tế bào RM có cài EFCI, bên nhận sẽ cài bit CI trong tế bào RM và trả lại cho bên gửi

## 3.7 Điều khiển tắc nghẽn TCP



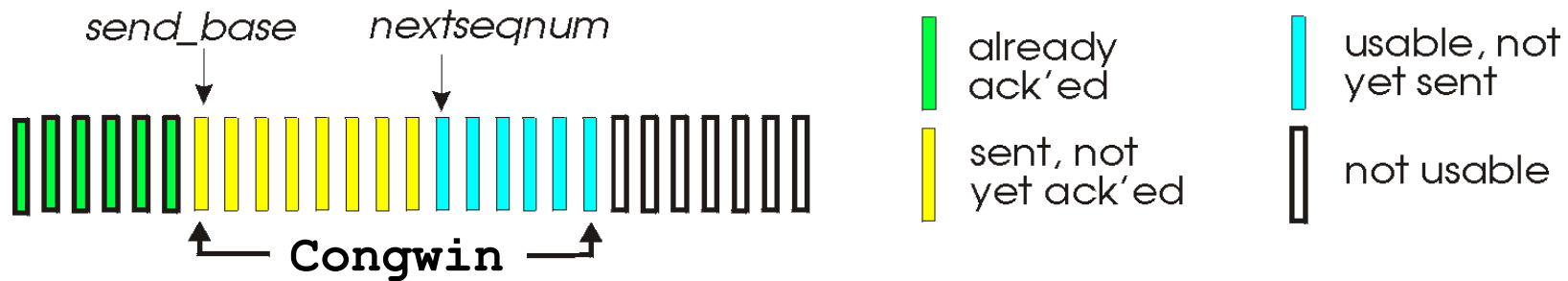
# Cơ chế điều khiển tắc nghẽn

- Các chiến lược (thuật toán) điều khiển tắc nghẽn trong TCP:
  - Slow Start (SS): bắt đầu chậm
  - Tính thời gian khứ hồi một cách thông minh + Rút lui theo hàm mũ
  - Congestion Avoidance (CA): Tránh tắc nghẽn
  - Fast Retransmit (FRTX): Phát lại nhanh
  - Fast Recovery (FRCV): Khôi phục nhanh
- Các thuật toán trên là độc lập, nhưng thường được kết hợp



# TCP Congestion Control (điều khiển tắc nghẽn)

- end-end control (no network assistance)
- Tốc độ truyền bị giới hạn bởi kích thước cửa sổ tắc nghẽn  $\text{congwin}$ , được tính là số segment



- $w$  segments, each with  $MSS$  bytes sent in one  $RTT$ :

$$\text{throughput} = \frac{w * MSS}{RTT} \text{ Bytes/sec}$$

# TCP congestion control:

- ❑ Sử dụng kỹ thuật thăm dò “**probing**” kênh truyền
  - Khi không có tắc nghẽn, thì truyền với tốc độ nhanh nhất có thể
  - *Tăng* Congwin cho đến khi mất gói tin (congestion)
  - Khi mất gói tin: *giảm* Congwin, sau đó lại thăm dò để tăng trở lại

- ❑ Có 2 giai đoạn (“phases”)
  - Khởi động chậm (slow start)
  - Tránh tắc nghẽn (congestion avoidance)

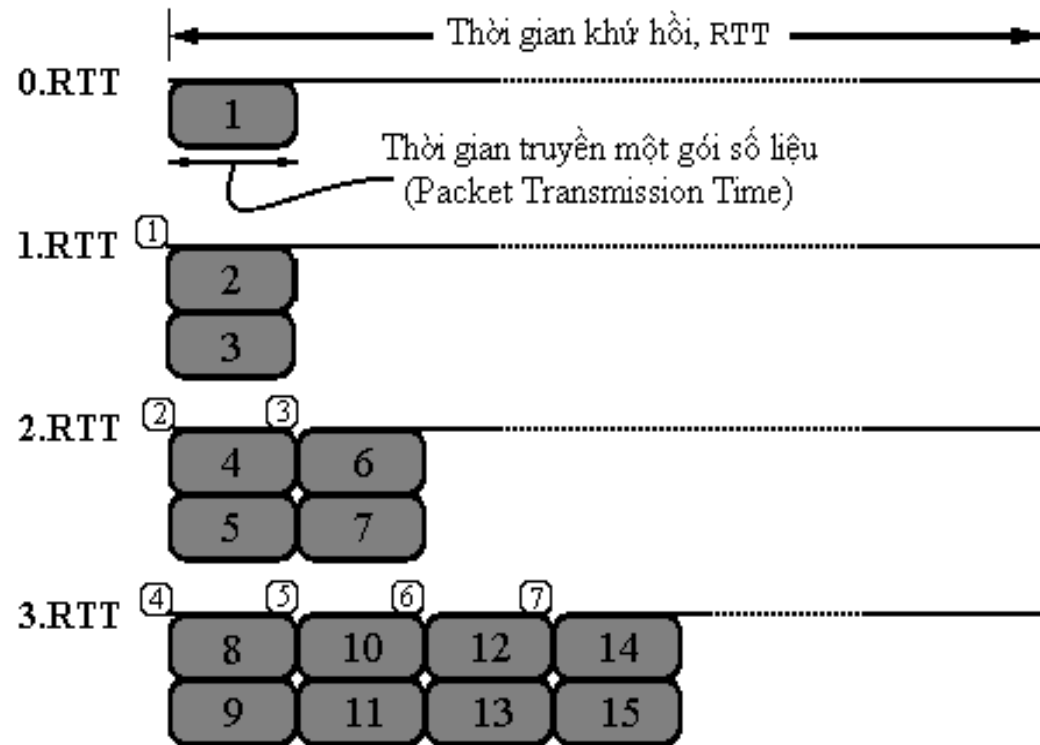
# TCP khởi động chậm - Thuật toán Slow Start

□ Thực thể phát sử dụng thêm biến:

- Congwin (congestion window) - kích thước của sổ phát
- threshold - giới hạn trên của cwnd, nếu vượt qua → tắc nghẽn

## Slowstart algorithm

```
initialize: Congwin = 1
for (each segment ACKed)
  Congwin++
until (loss event OR
      CongWin > threshold)
```



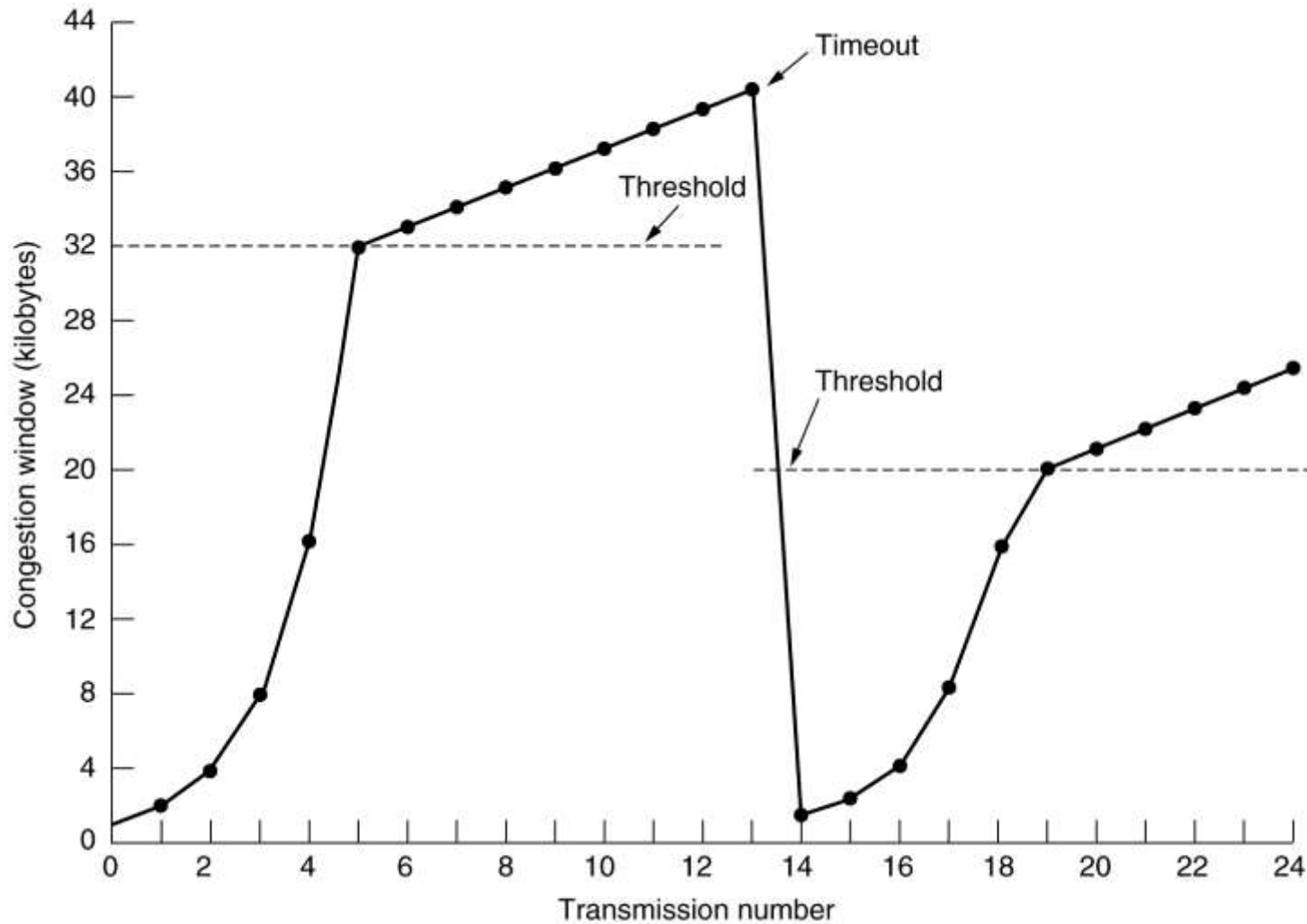
□ Lưu ý

- Congwin: tăng theo hàm mũ
- Loss: khi timeout hoặc 3 ACK trùng lặp

# Tránh tắc nghẽn - Thuật toán CA:

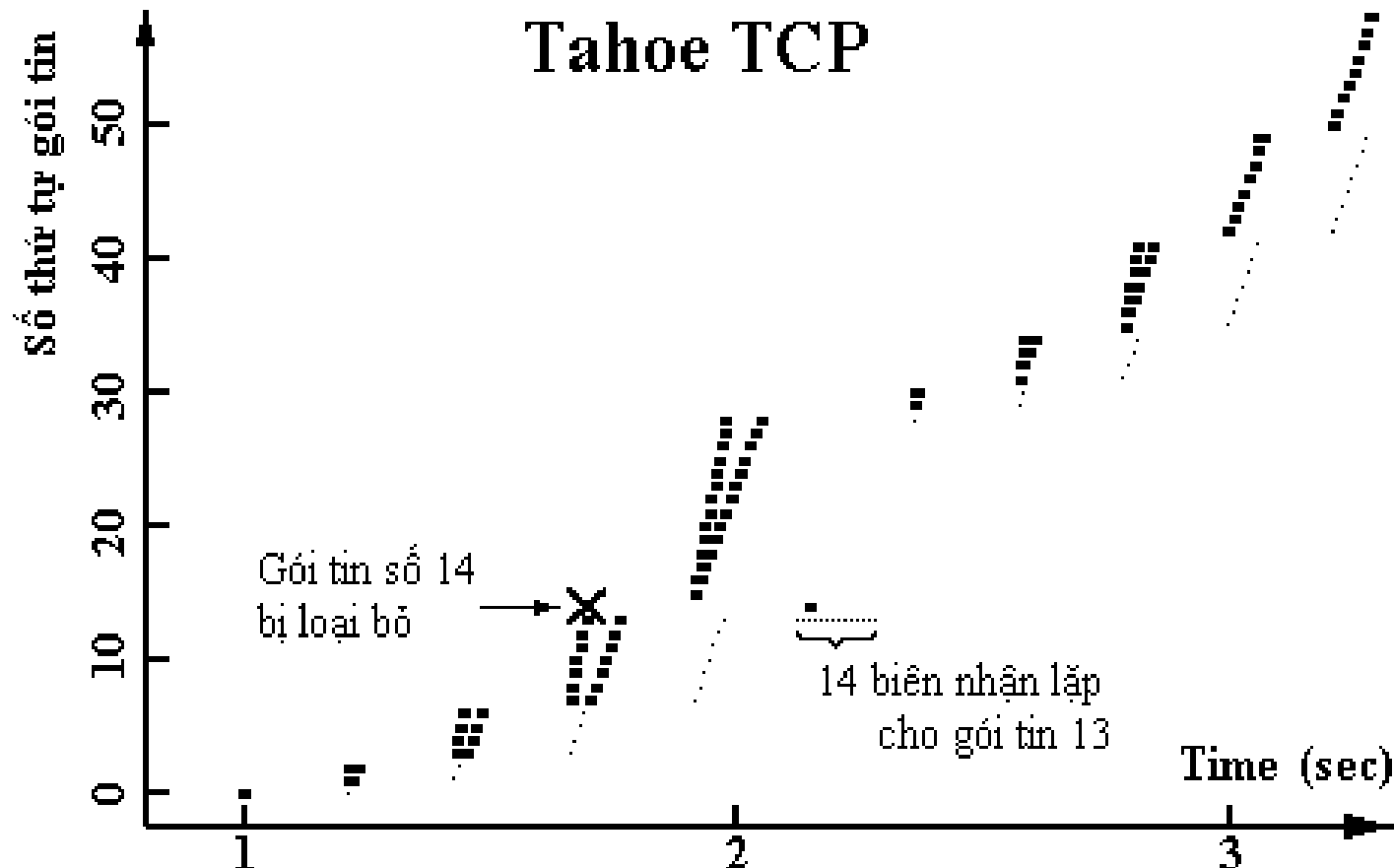
- Congwin := Congwin + 1 với mỗi ack.
- Khi phát hiện dấu hiệu tắc nghẽn:
  - $\text{threshold} := \text{Congwin}/2$ ,  $\text{Congwin} := 1$
  - $\text{RTO} = \text{RTO} * 2$  (Exponential backoff)
  - $\rightarrow \text{SS}$
- Nhận xét:
  1. Trong giai đoạn CA, cwnd tăng tuyến tính:
    - Đảm bảo tận dụng băng thông có thể sử dụng được
    - Vẫn thăm dò tiếp khả năng sử dụng băng thông nhiều hơn
  2. Congwin bị giảm theo cấp số nhân (Multiplicative Decreased)

# Minh họa thuật toán SS và CA:



# Fast Retransmit (FRTX)

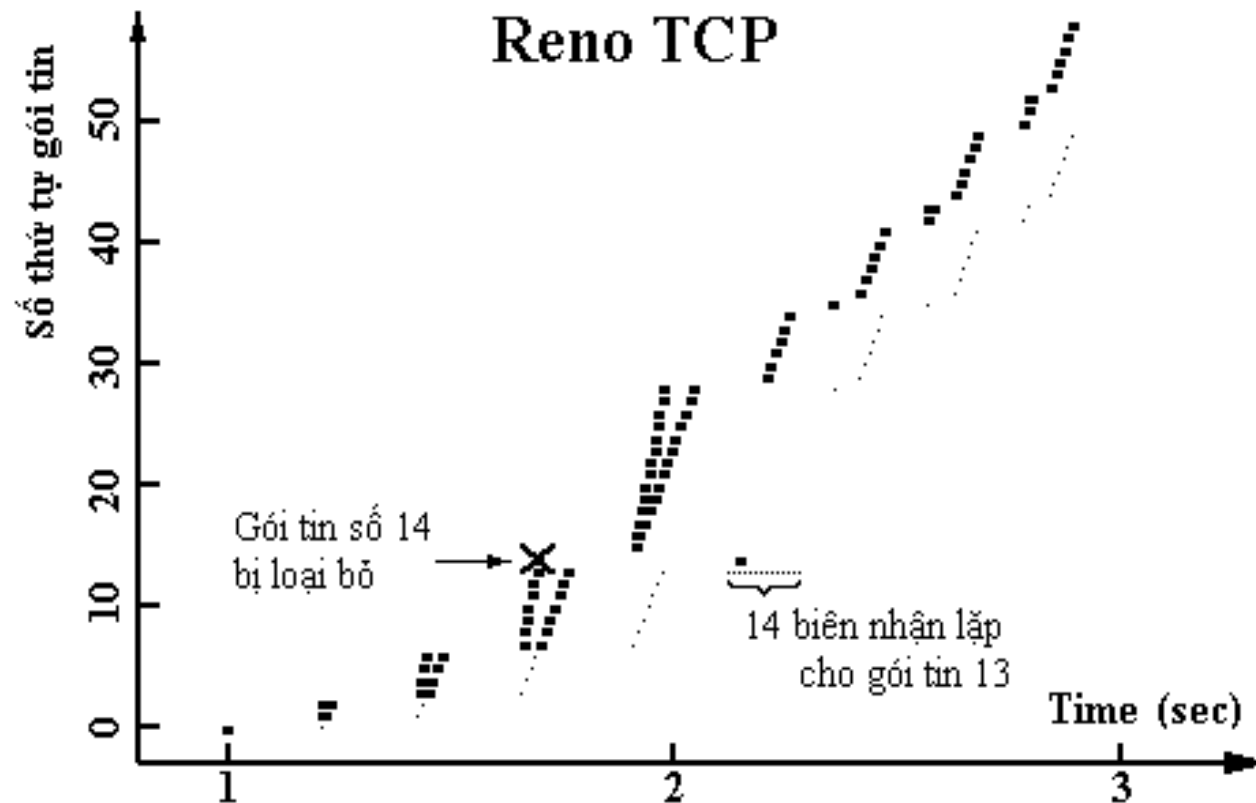
- Sau khi nhận được Dupack ( $\geq 3$ ), TCP thực hiện phát lại nhanh, không chờ bị Timeout, sau đó chuyển ngay về SS.
- Đây là một cách “dự đoán thông minh” rằng, gói tin đã bị mất.



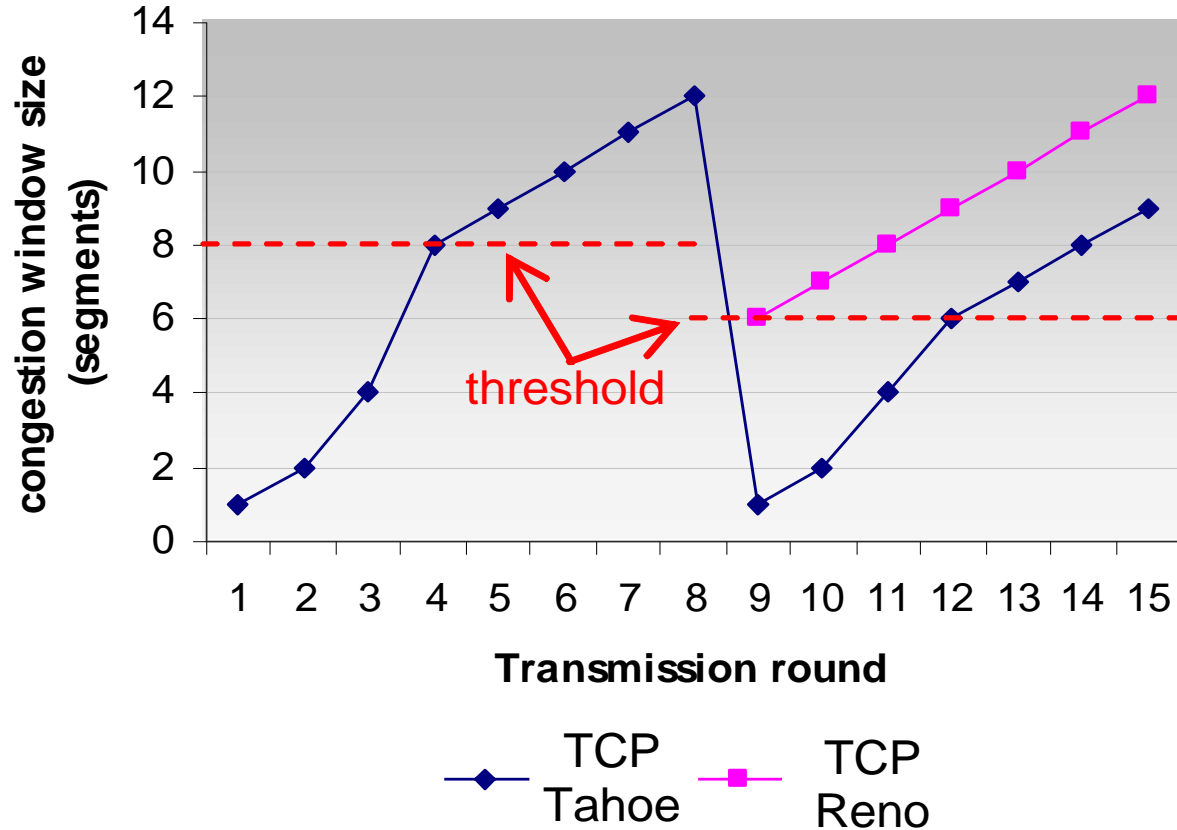
# Fast Recovery (FRCV):

Cải tiến FRTX: thực hiện FRTX xong về CA chứ không về SS:

- $ssthresh := Congwin / 2$ , nhưng không nhỏ hơn 2 (gói tin)
- $Congwin := Congwin + 3$ . Bên gửi “đoán”: 3 dupack ứng với 3 gói tin đã được nhận đúng.
- Với mỗi dupack nhận được thêm, tăng  $Congwin := Congwin + 1$



# TCP Reno và TCP Tahoe:



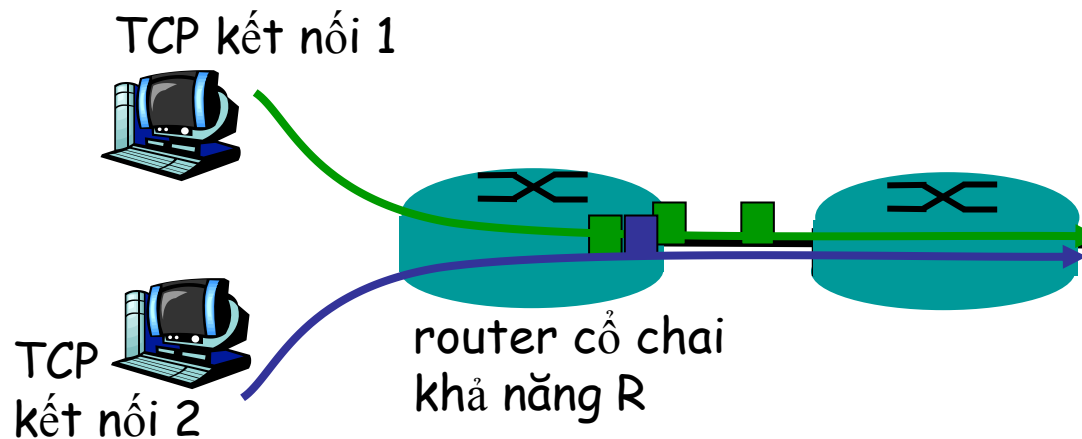


## *Chú ý:*

- Trong các thuật toán trên, không bao giờ tăng cửa sổ phát quá giá trị mà bên nhận loan báo - rcnd (Receiver Adverticed Window). Nói cách khác, cửa sổ phát = Congwin, rcnd}

# TCP: tính công bằng

**Mục tiêu:** nếu K phiên làm việc TCP chia sẻ kết nối cổ chai của băng thông là R, mỗi phiên có tốc độ trung bình là  $R/K$



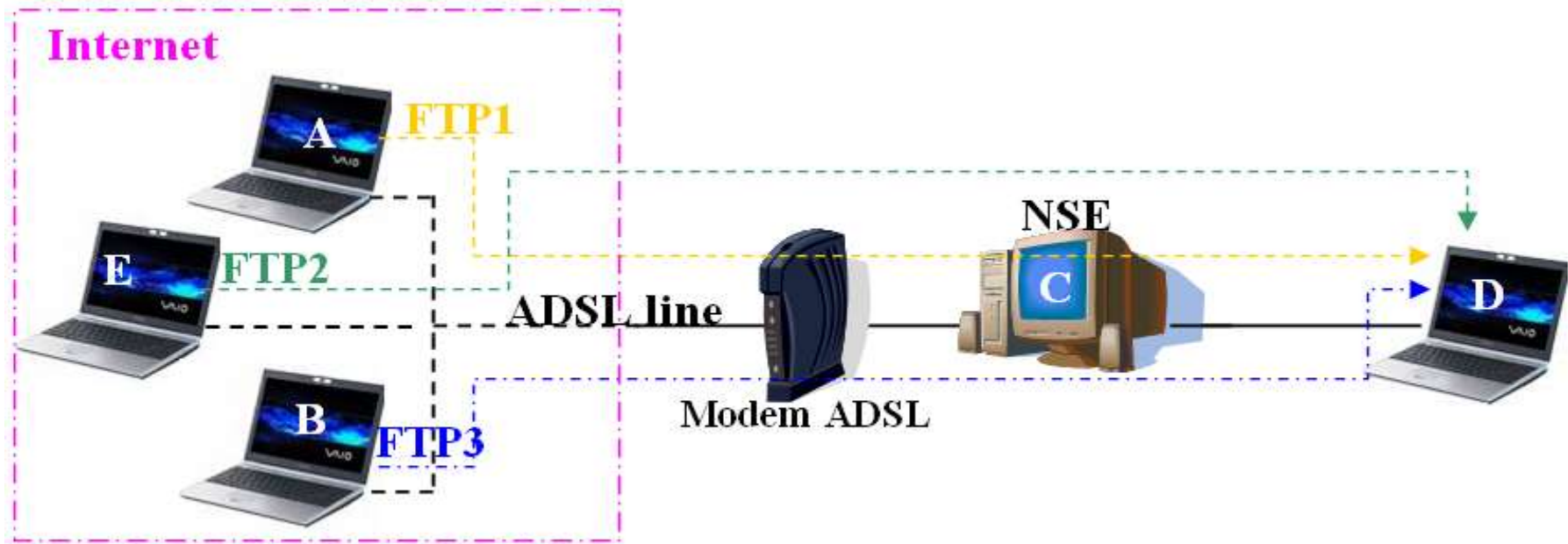
# TCP: tính công bằng (tt)

## Tính công bằng & UDP

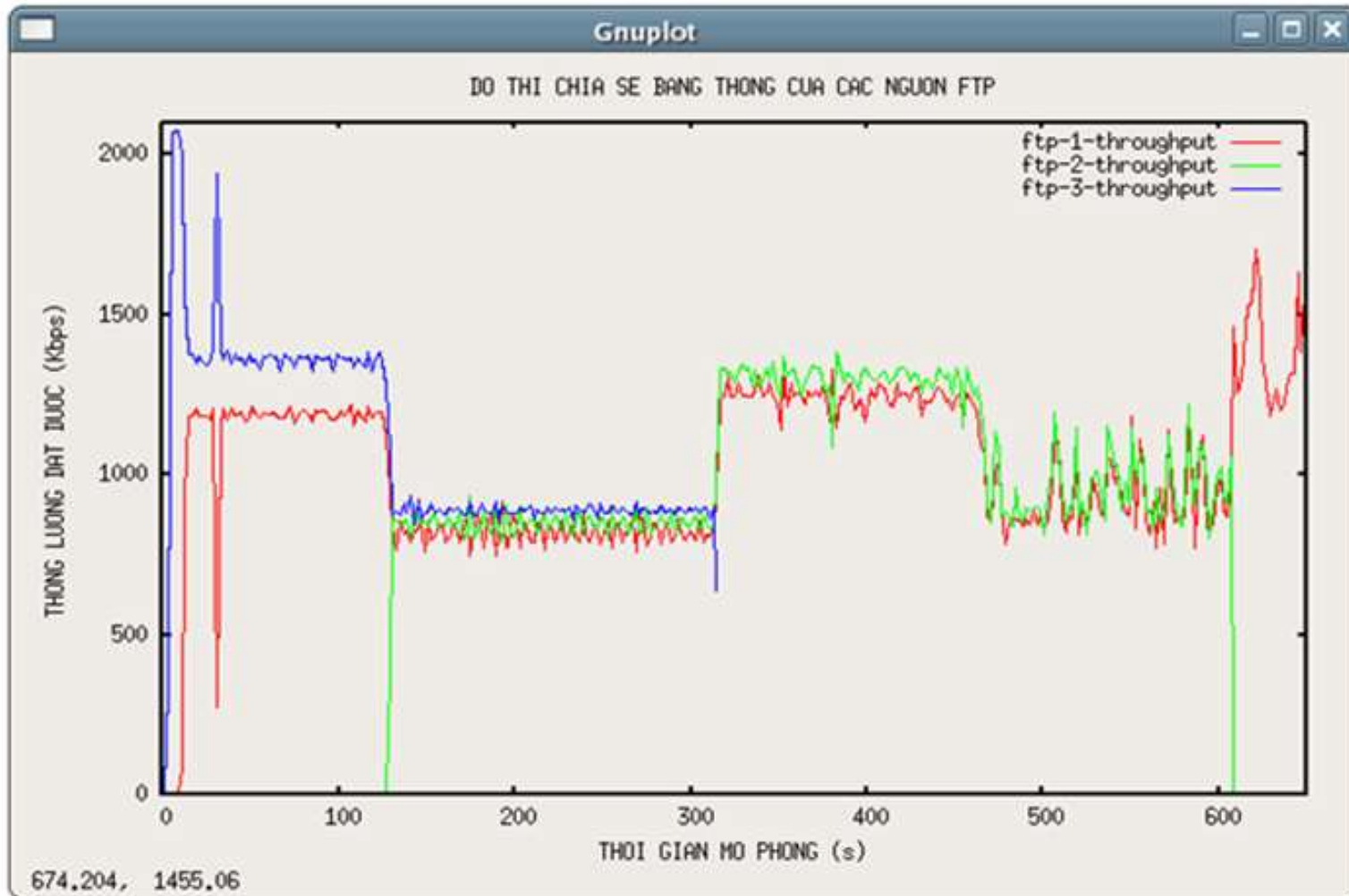
- ❑ nhiều ứng dụng thường không dùng TCP
  - không muốn tốc độ bị điều tiết do điều khiển tắc nghẽn
- ❑ Thay bằng dùng UDP:
  - truyền audio/video với tốc độ ổn định, chịu được mất mát
  
- ❑ Làm thế nào để xây dựng một ứng dụng truyền file (tin cậy) mà không phải chia sẻ đường truyền???

# Ví dụ Thực nghiệm 1

- Mục tiêu thực nghiệm
- Topo mạng mô phỏng

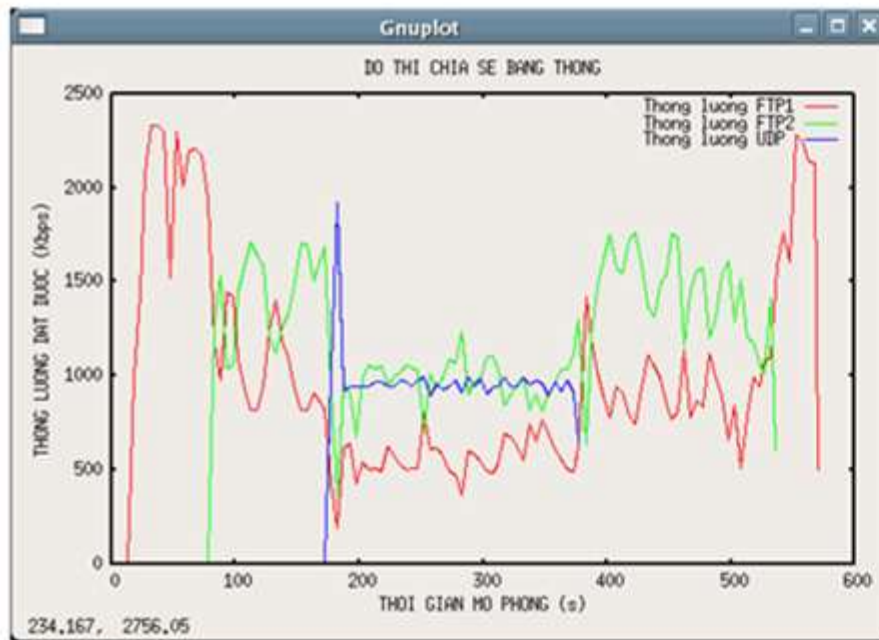


# Kết quả thực nghiệm 1

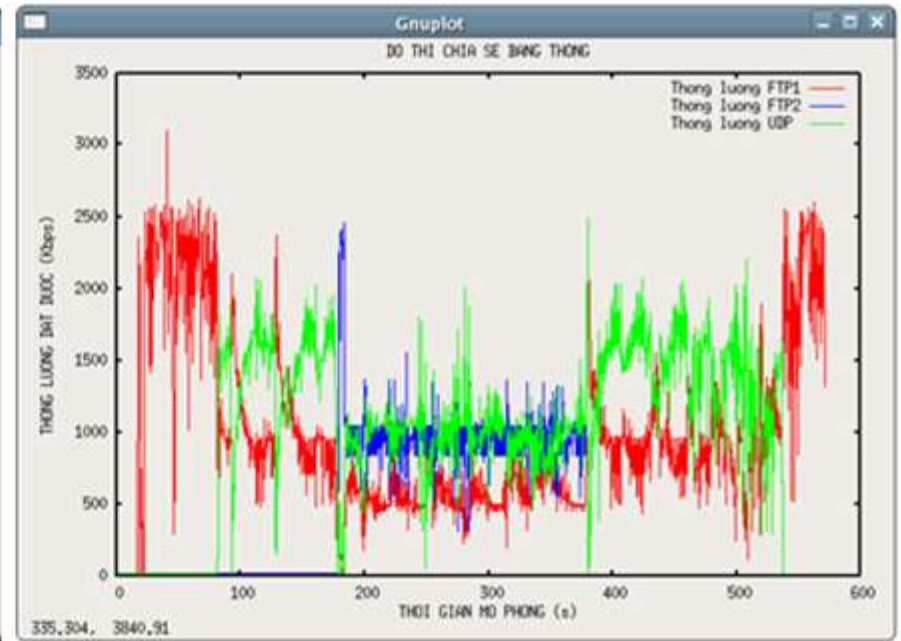


Đồ thị thông lượng của ba kết nối ftp khi chia sẻ băng thông

# Thực nghiệm 2



Hình (a) Kết quả mô phỏng lần 1



Hình (b) Kết quả mô phỏng lần 1 với thời gian tính TLTB là 0.5s



Hết Chương 3

# NHẬP MÔN MẠNG MÁY TÍNH

## Chương 1

### GIỚI THIỆU

### MẠNG MÁY TÍNH





# Nội dung chương 1

I. Định nghĩa mạng máy tính

II. Các mô hình mạng máy tính

III. Kiến trúc mạng máy tính

IV. Môi trường truyền vật lý mạng cục bộ

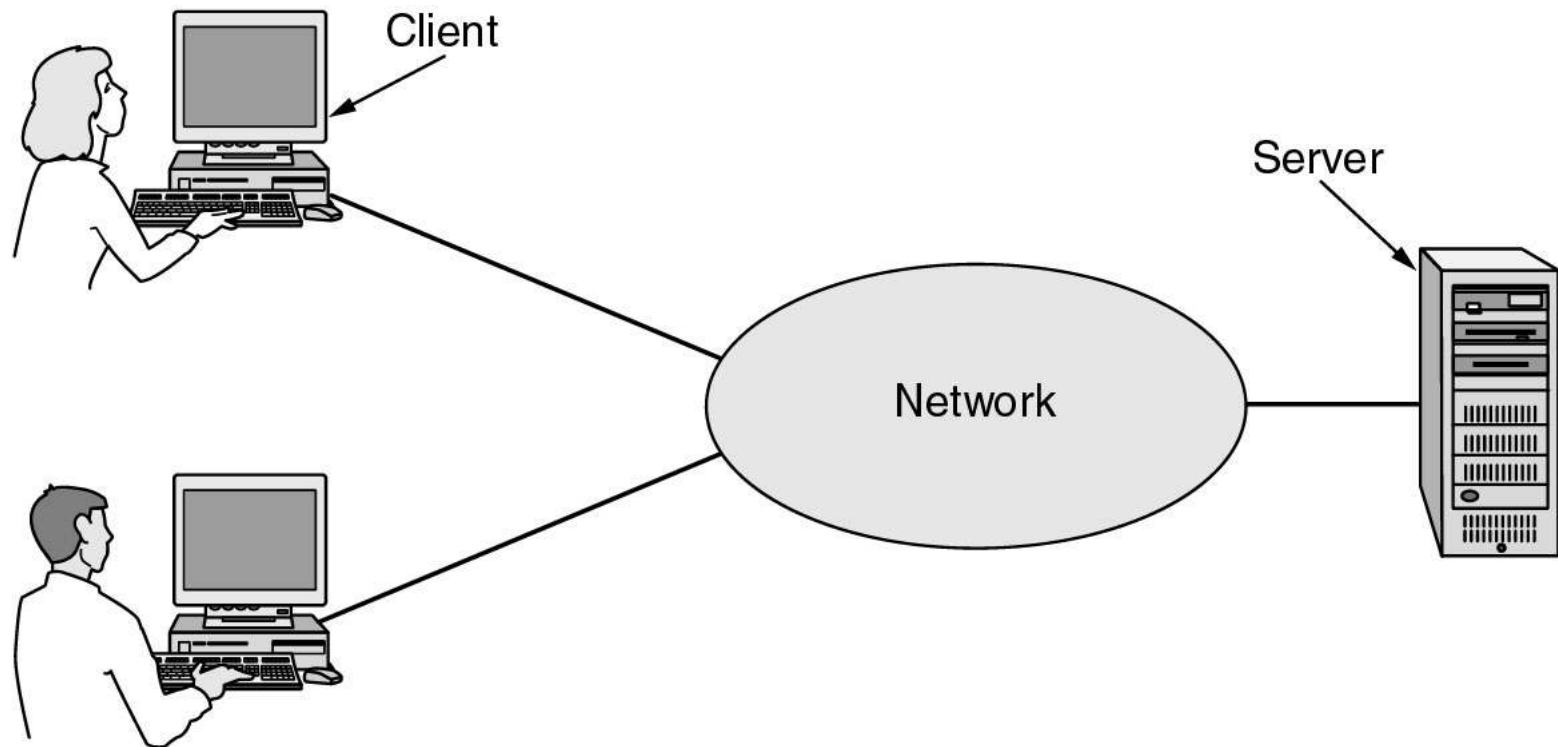
# I. Định nghĩa mạng máy tính

Mạng máy tính:

- bao gồm các máy tính độc lập,
- được kết nối với nhau trên mạng
- nhằm chia sẻ tài nguyên  
và trao đổi dữ liệu

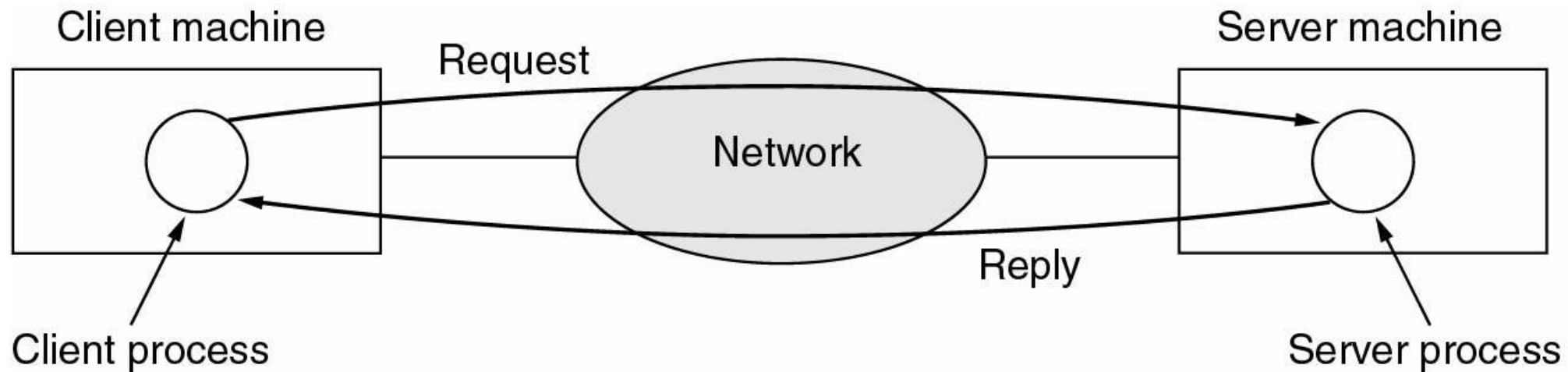
Host: máy tính trên mạng

# Ví dụ 1: mô hình client-server

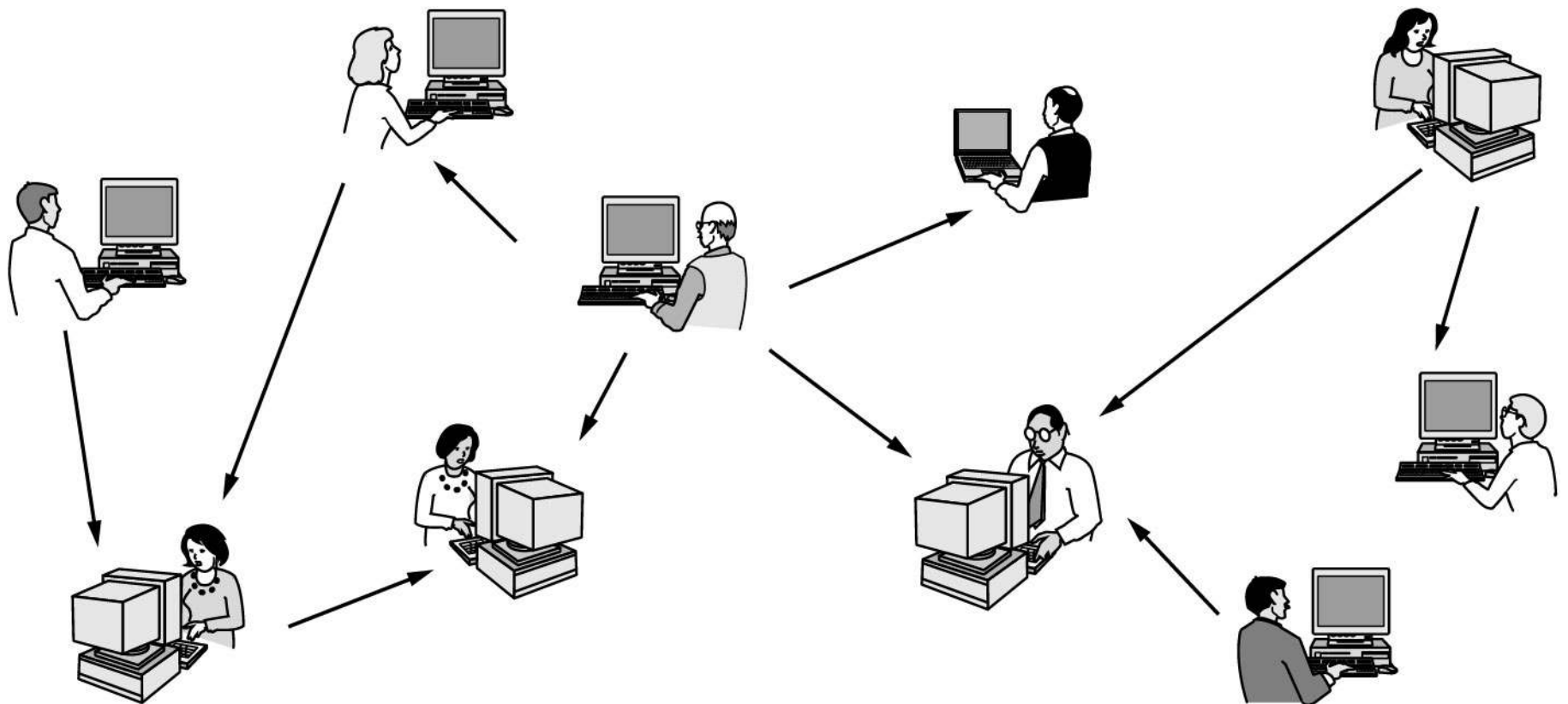


Mạng máy tính với 2 Client và 1 Server

# Mô hình ứng dụng mạng Client-Server



## Ví dụ 2: mô hình peer-to-peer



Mạng ngang hàng (peer-to-peer network)



# Các ứng dụng của mạng máy tính

- Ứng dụng trong cơ quan, doanh nghiệp
- Ứng dụng trong gia đình, cá nhân
- Ứng dụng trên thiết bị di động

## Ví dụ: e-commerce – thương mại điện tử

	Dạng đầy đủ	Ví dụ
B2C	Business-to-consumer	Đặt mua hàng trên mạng
B2B	Business-to-business	Nhà sản xuất đặt hàng
G2C	Government-to-consumer	Chính phủ phát hành biểu mẫu
C2C	Consumer-to-consumer	Đấu giá trên mạng
P2P	Peer-to-peer	Chia sẻ file

Một số dạng thương mại điện tử



## II. Các mô hình mạng máy tính

II.1 Các kỹ thuật truyền dữ liệu

II.2 Phân loại mạng máy tính

II.3 Phần cứng mạng máy tính

II.4 Phần mềm mạng máy tính





## II.1 Các kỹ thuật truyền dữ liệu

Hai dạng truyền dữ liệu cơ bản:

- Broadcast (quảng bá)
- Point-to-point (giữa hai điểm)

# Truyền dữ liệu dạng broadcast

- Dùng 1 kênh truyền chung cho tất cả các máy trên mạng
- Dữ liệu (packet) gửi từ 1 máy sẽ đến tất cả các máy khác
- Có địa chỉ máy nhận cùng với dữ liệu

Multicast: 1 máy gửi dữ liệu và một nhóm máy nhận

# Truyền dữ liệu dạng point-to-point

- Tồn tại một kênh truyền riêng giữa hai máy
- Kênh truyền này có thể qua các máy trung gian khác trên mạng
- Còn được gọi là dạng unicast

## II.2 Phân loại mạng máy tính

Khoảng cách	Loại mạng
10m – 1km	Local Area Network (LAN)
10km-100km	Metropolitan Area Network (MAN)
100km-1.000km	Wide Area Network (WAN)
10.000km	Internet

Phân loại mạng máy tính theo khoảng cách

# Các dạng mạng cục bộ (LAN)

## ■ Mạng ngang hàng (workgroup)

- Các máy tương đương nối mạng để chia sẻ tài nguyên

## ■ Mạng client/server

- Có một hoặc nhiều máy dùng làm server để quản lý user, cài đặt các ứng dụng, lưu trữ dữ liệu ...
- Các máy khác kết nối đến server để truy xuất có kiểm soát các tài nguyên

## II.3 Phần cứng mạng máy tính

- Local Area Network

Mạng cục bộ

- Wide Area Network

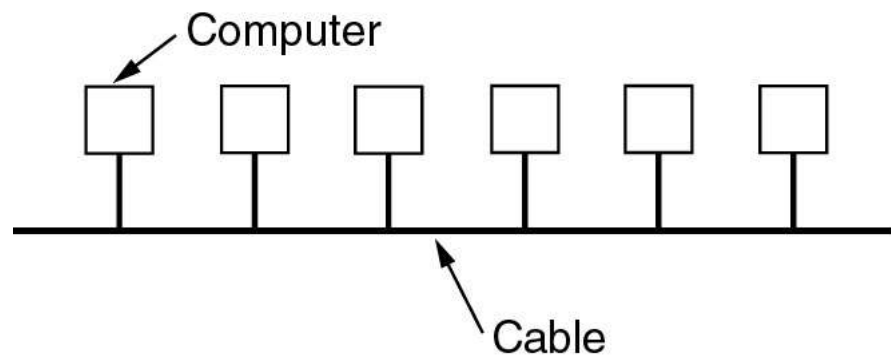
Mạng miền rộng/Mạng diện rộng

- Wireless Network

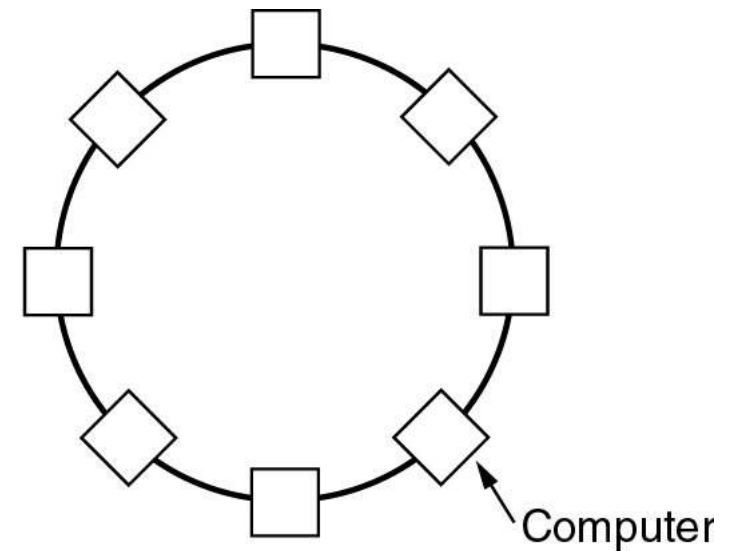
Mạng cục bộ không dây (ví dụ Wi-Fi)

Mạng miền rộng không dây (ví dụ WiMax)

# Mạng cục bộ - LAN



(a)



(b)

Hai dạng mạng cục bộ

a. Bus      b. Ring

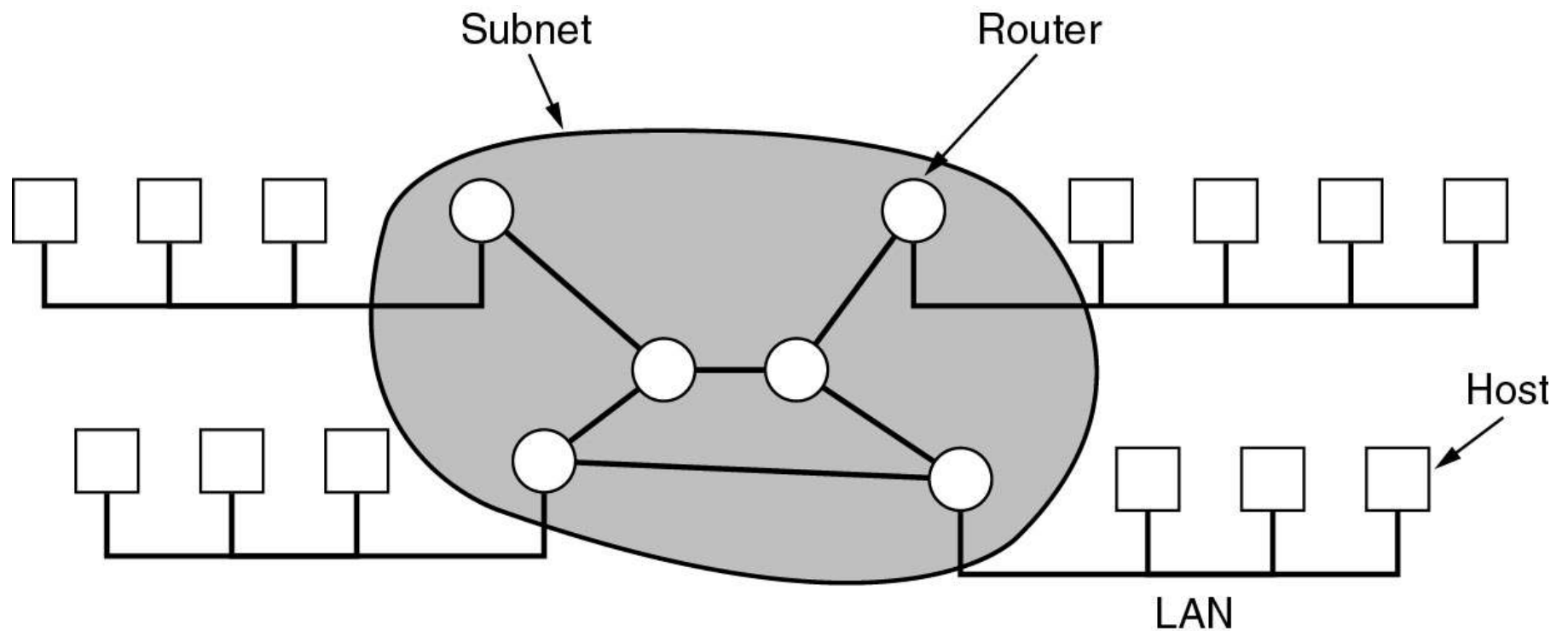


# Các thành phần kết nối LAN

- Card mạng – Network Interface Card (NIC)
- Dây mạng – Cable
- Các thiết bị kết nối: Hub, Switch, ...



# Mạng miền rộng - WAN



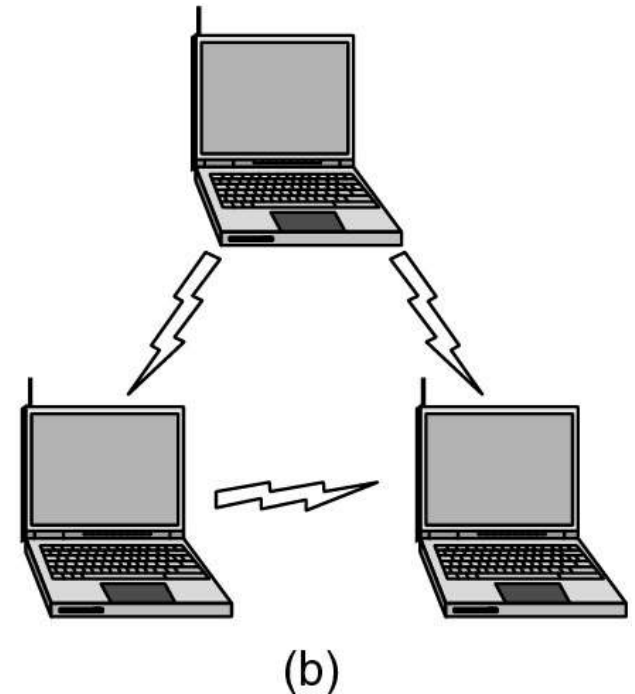
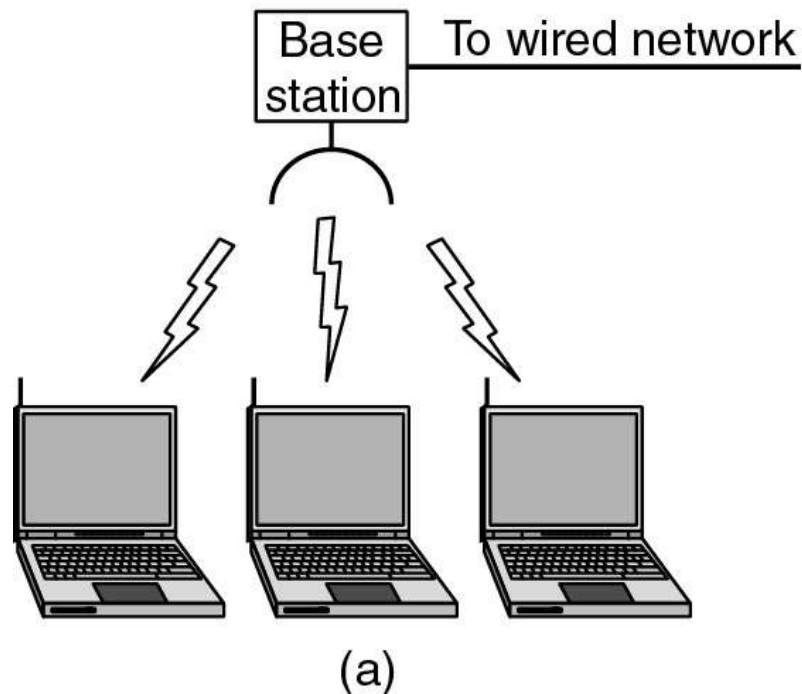
WAN gồm các LANs và phần kết nối (subnet)

# Subnet

Phần kết nối mạng miền rộng gồm 2 phần:

- Các đường truyền (transmission lines)  
dây đồng, cáp quang, sóng điện từ, ...
- Các phần tử chuyển mạch (switching elements), thường được gọi là router
  - Kết nối với nhiều đường truyền
  - Nhận dữ liệu và chọn đường truyền để chuyển sang mạng khác

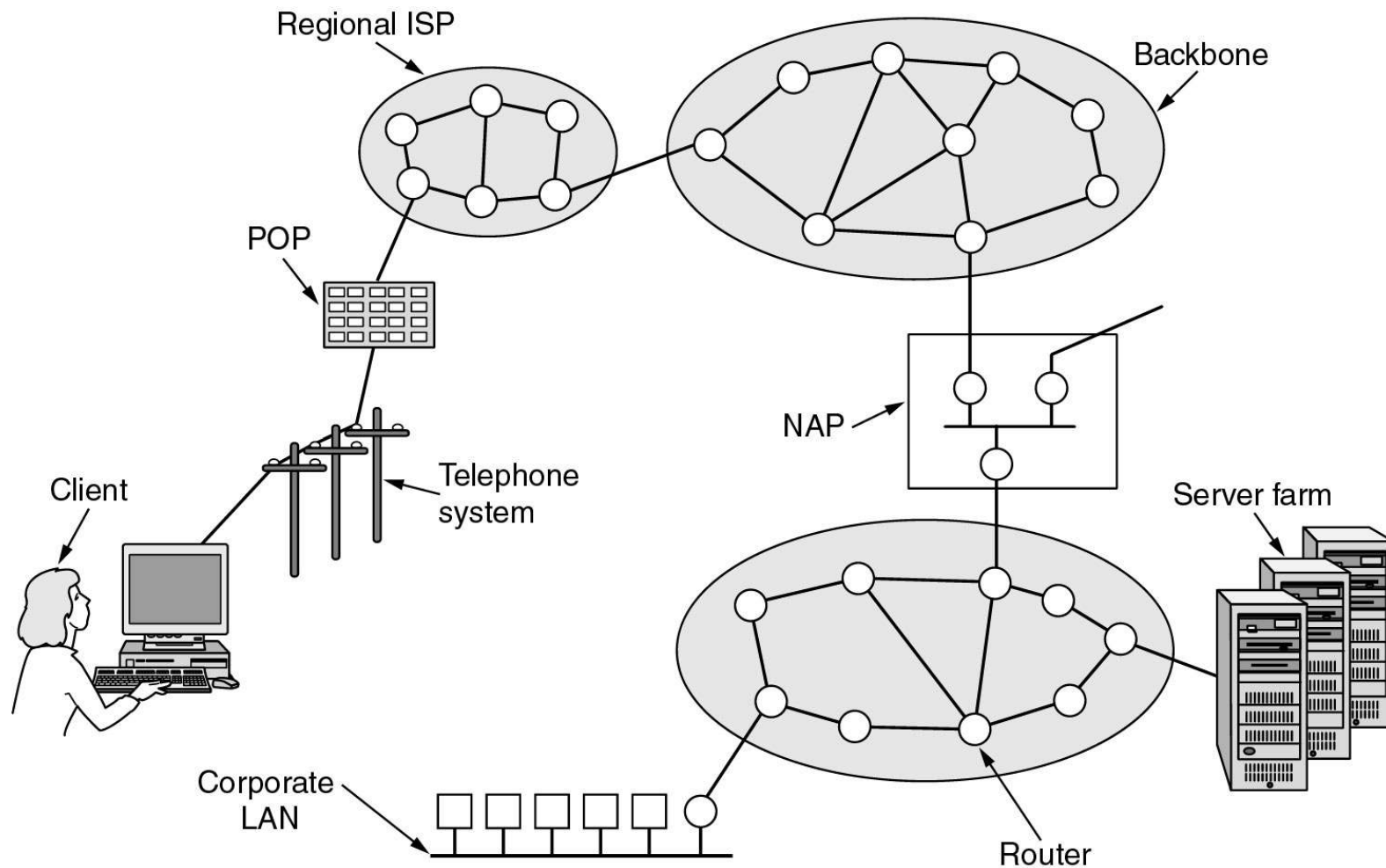
# Mạng không dây – Wireless LAN



Hai dạng kết nối mạng không dây

- Có dùng base station, còn gọi là access point
- Các máy gửi nhận trực tiếp, ad hoc networking

# Mạng Internet



Tổng quan mạng Internet

# Các thành phần chính trên mạng Internet

- Trục chính – Backbone
- Các nhà cung cấp dịch vụ - ISPs  
(Internet Service Provider)
  - POP (Point of Presence): nơi nhận tín hiệu từ mạng điện thoại và đưa vào mạng của ISP
- NAP (Network Access Point)
- Các server
- Client từ máy lẻ, các LANs

## II.4 Phần mềm mạng máy tính

- Hệ điều hành mạng
- Phần mềm phía server
- Phần mềm phía client



# III. Kiến trúc mạng máy tính

III.1 Tổ chức thứ bậc của các giao thức

III.2 Các tiêu chuẩn mạng

- ISO OSI

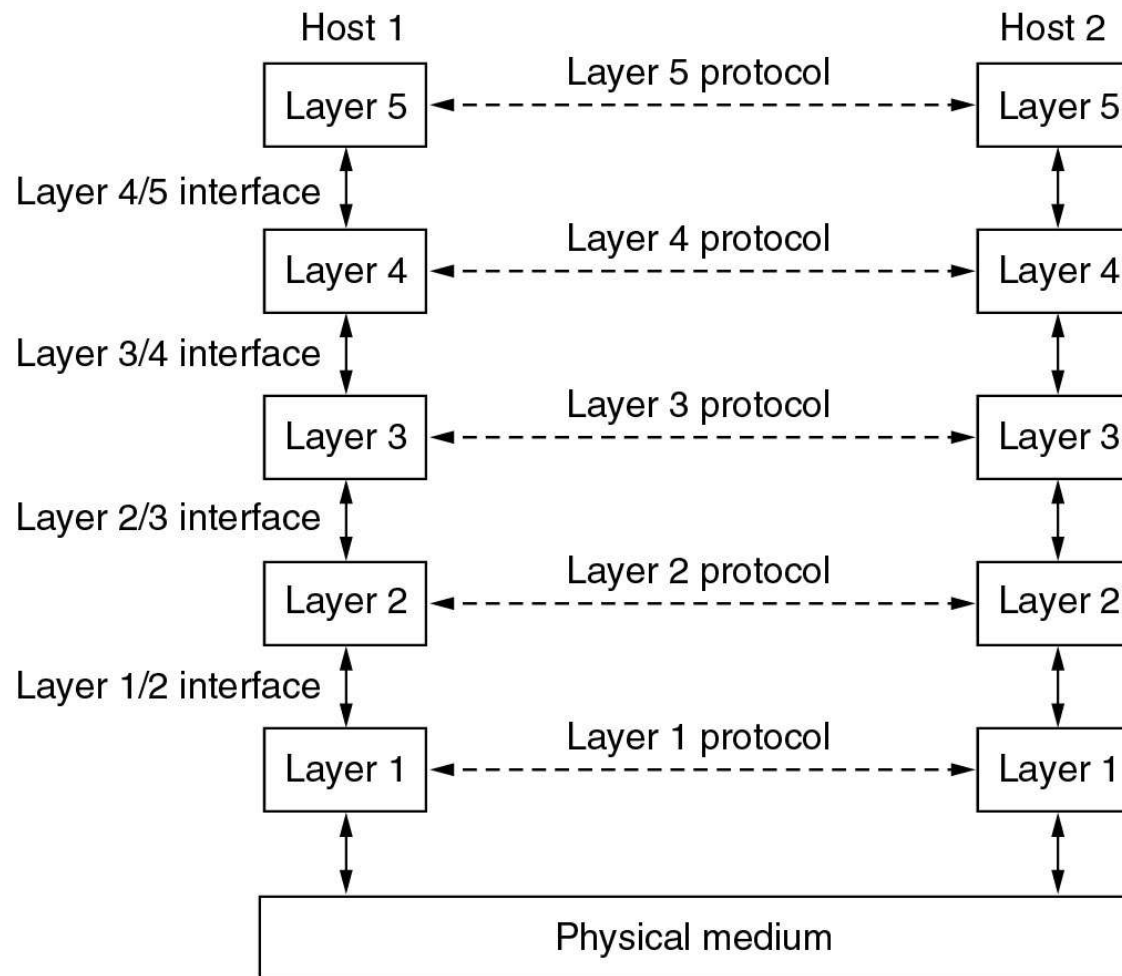
- TCP/IP

## III.1 Tổ chức thứ bậc của các giao thức

- Tổ chức luận lý mạng máy tính: gồm các lớp (layers/levels)
  - Số lớp, chức năng mỗi lớp phụ thuộc loại mạng.
- Giao thức (protocol): tập hợp các luật và thủ tục thực hiện việc truyền thông giữa hai bên truyền thông.
- Giao diện (Interface): định nghĩa các thao tác cơ sở của lớp dưới cung cấp cho lớp trên



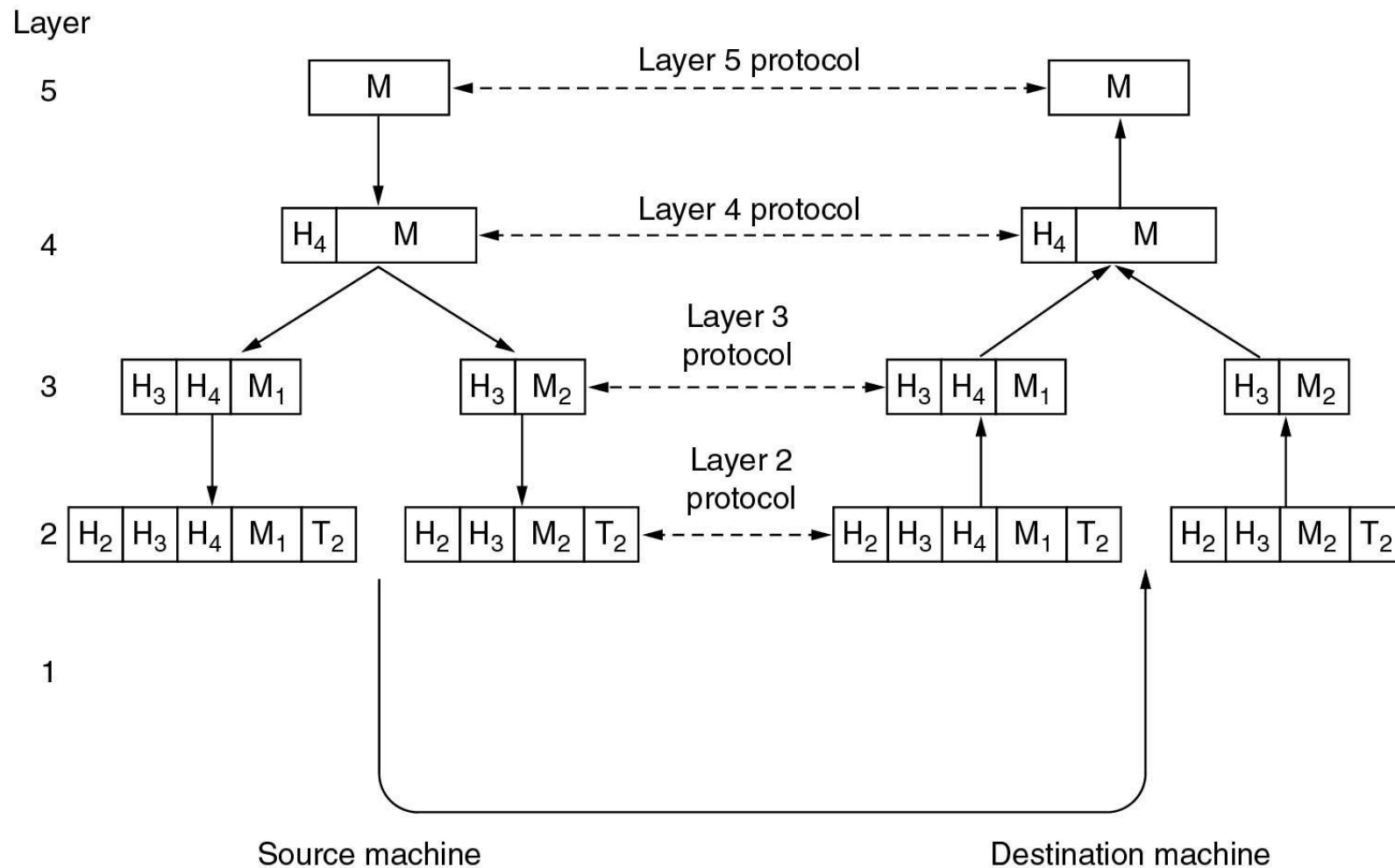
# Ví dụ: tổ chức mạng có 5 lớp



# Mục đích của tổ chức mạng theo lớp

- Giảm sự phức tạp khi thiết kế
- Mô tả chi tiết quá trình truyền dữ liệu từ một máy đến một máy khác

# Ví dụ: truyền dữ liệu M giữa 2 máy



H: Header, T: Trailer

# Kiến trúc mạng máy tính

- Kiến trúc mạng máy tính:

Tập hợp các lớp và giao thức.

- Bộ giao thức (protocol stack / protocol suite): Danh sách các giao thức được sử dụng cho từng lớp trên một hệ thống xác định.

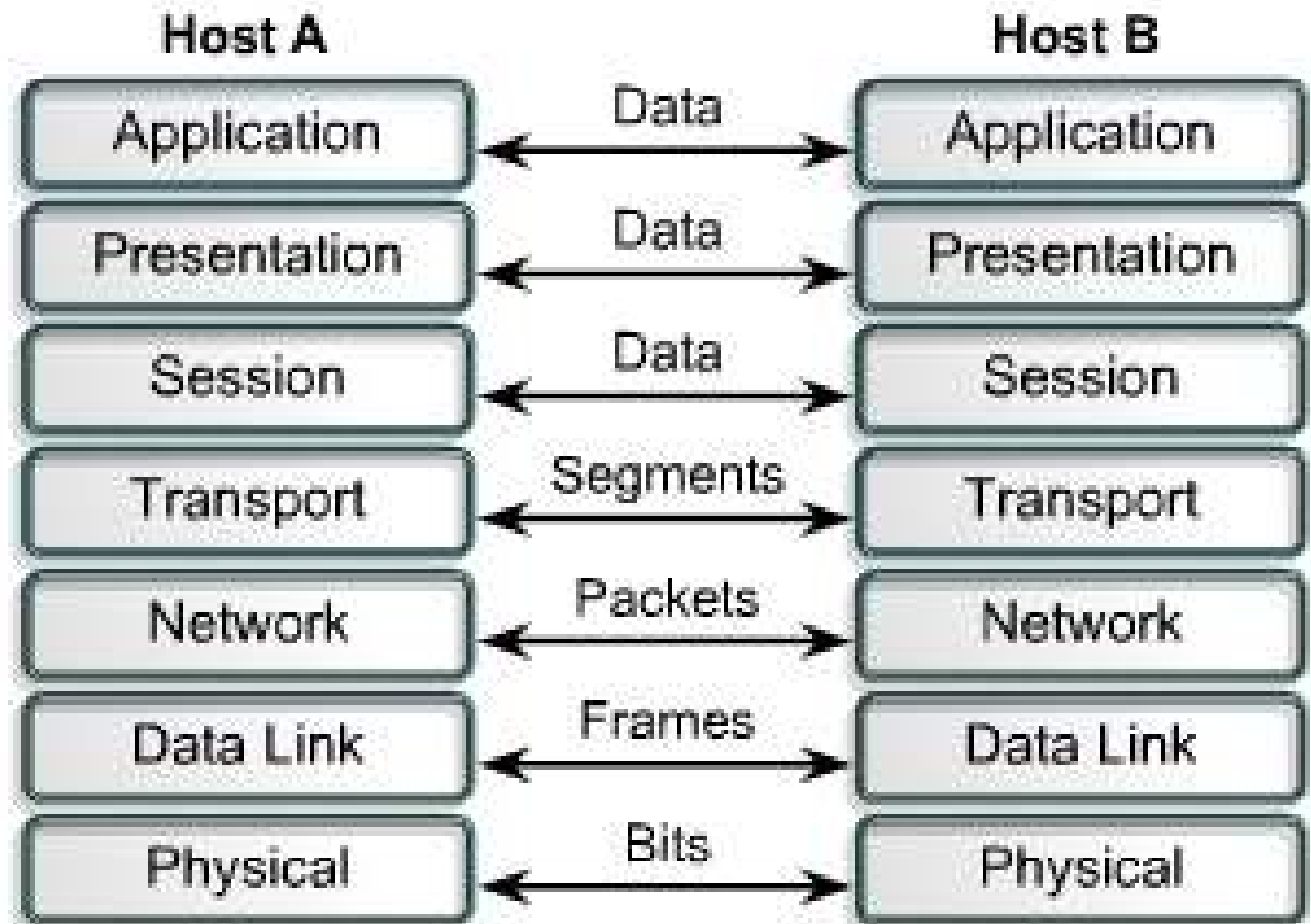
## III.2 Các tiêu chuẩn mạng

- Hai mô hình kiến trúc mạng quan trọng:  
OSI (Open Systems Interconnection)  
TCP/IP (Transmission Control Protocol/  
Internet Protocol)
- Các bộ giao thức khác:
  - IPX/SPX (Internetwork Packet Exchange/  
Sequenced Packet Exchange)
  - NetBEUI (NetBIOS Extended User Interface)
  - AppleTalk

# OSI



a. Mô hình OSI



b. Truyền thông giữa 2 máy

# Sơ lược chức năng các lớp mô hình OSI

## Lớp vật lý - Physical

- Truyền chuỗi bit trên kênh truyền
- Quy định về môi trường truyền vật lý, tín hiệu điện, cơ khí.

## Lớp liên kết dữ liệu – Data Link

- Truyền dữ liệu có cấu trúc (frame) tin cậy giữa hai máy trên môi trường vật lý.
- Quy định về địa chỉ thiết bị, kiểm soát lỗi

# Sơ lược chức năng các lớp mô hình OSI (tt)

## Lớp mạng – Network

- Xác định con đường (route) từ máy gửi đến máy nhận, quản lý các vấn đề lưu thông trên mạng
- Quy định về địa chỉ mạng

## Lớp giao vận - Transport

- Chia dữ liệu thành các đơn vị nhỏ hơn nếu cần và ghép lại tại nơi nhận.
- Thực hiện kiểm soát lỗi



# Sơ lược chức năng các lớp mô hình OSI (tt)

## Lớp phiên – Session

- Thiết lập, quản lý, kết thúc các phiên làm việc giữa các ứng dụng

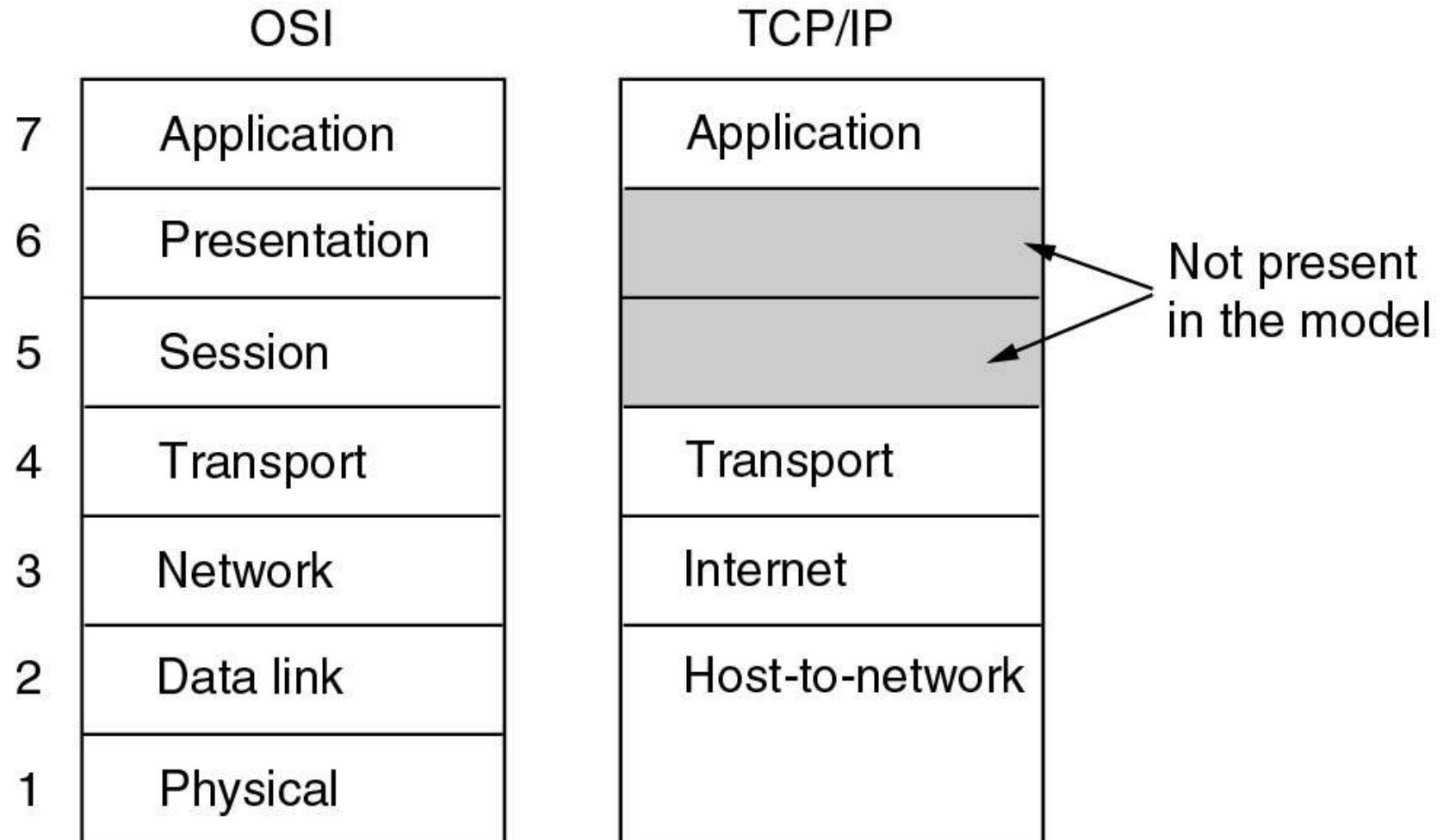
## Lớp trình diễn – Presentation

- Quy định về khuôn dạng, cú pháp, ngữ nghĩa của dữ liệu khi truyền thông  
→ data representation

## Lớp ứng dụng – Application

- Bao gồm các giao thức của các dịch vụ mạng

# OSI và TCP/IP

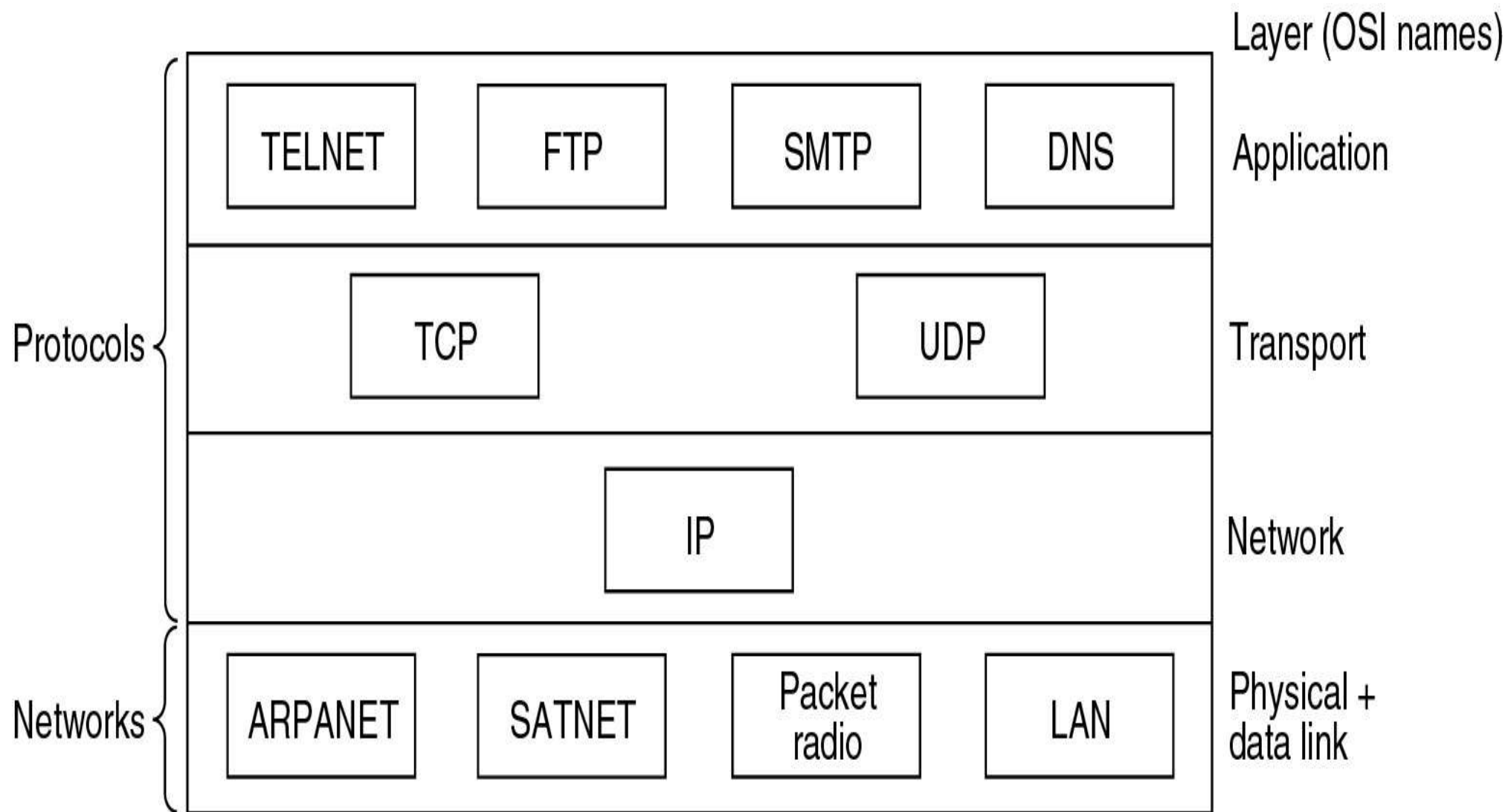


# TCP/IP

Có 4 lớp, so với mô hình OSI:

- Lớp ứng dụng (application) bao gồm lớp presentation và lớp session của mô hình OSI
- Lớp giao vận giải quyết vấn đề chất lượng dịch vụ (quality of service) như độ tin cậy, kiểm soát lỗi, kiểm soát lưu lượng
- Lớp internet chia dữ liệu từ lớp transport thành các gói (packet)
- Lớp host-to-network thực hiện tạo kết nối vật lý, bao gồm các lớp Physical và Data Link của mô hình OSI

# Một phần của bộ giao thức TCP/IP





## IV. Môi trường truyền vật lý mạng cục bộ

IV.1 Card mạng

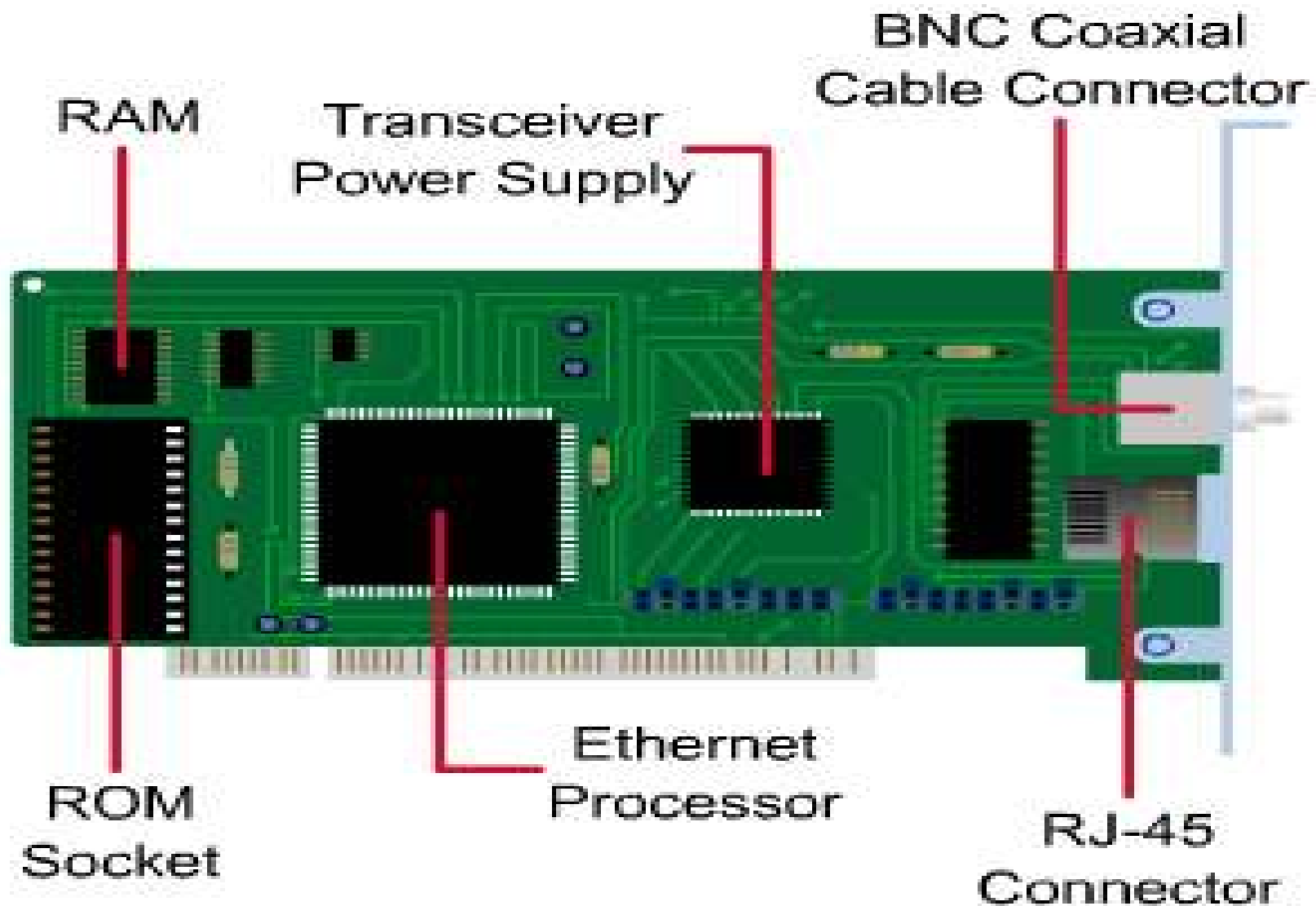
IV.2 Dây mạng

IV.3 Một số thiết bị kết nối

## IV.1 Card mạng



# Các thành phần trên card mạng



# Card mạng không dây



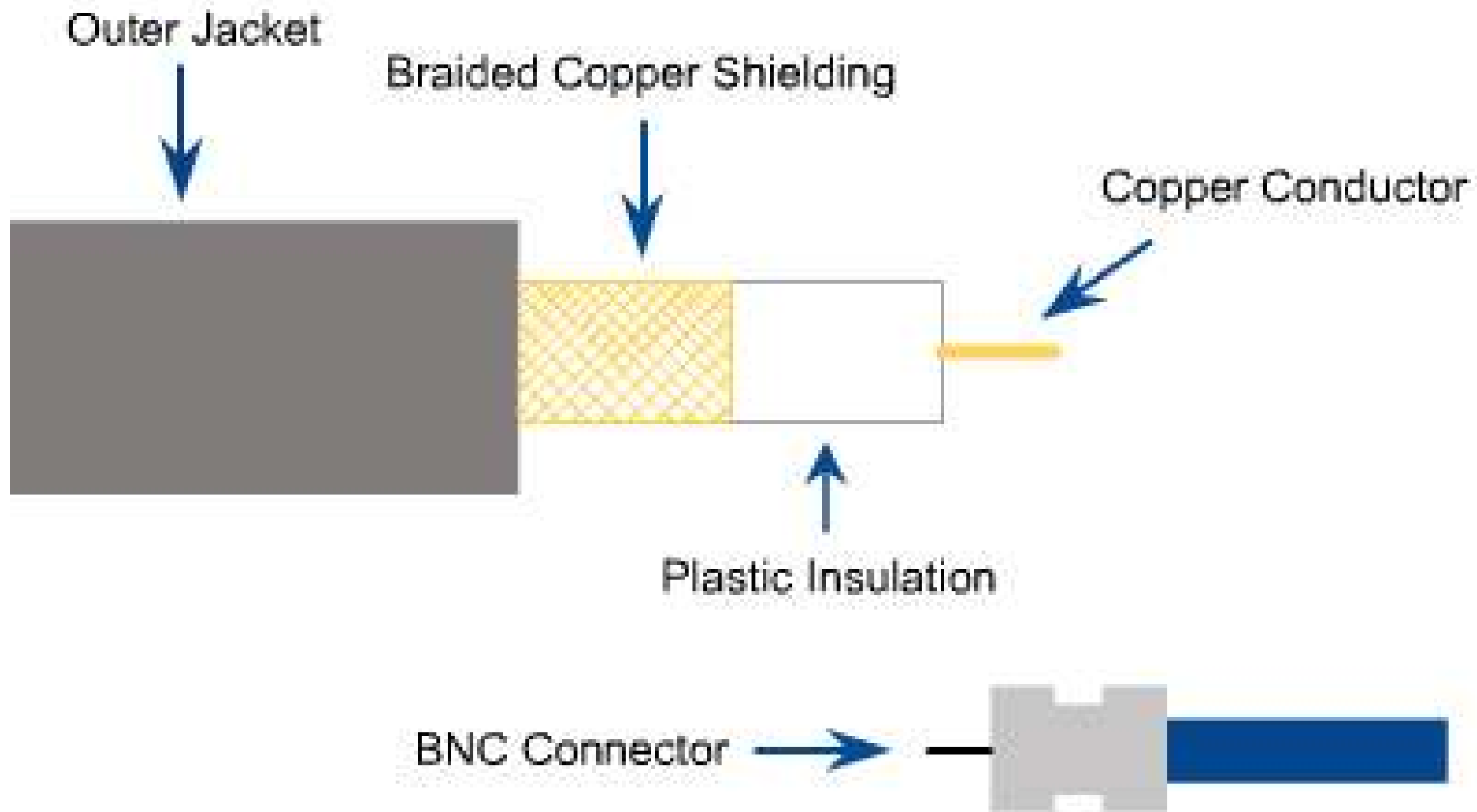




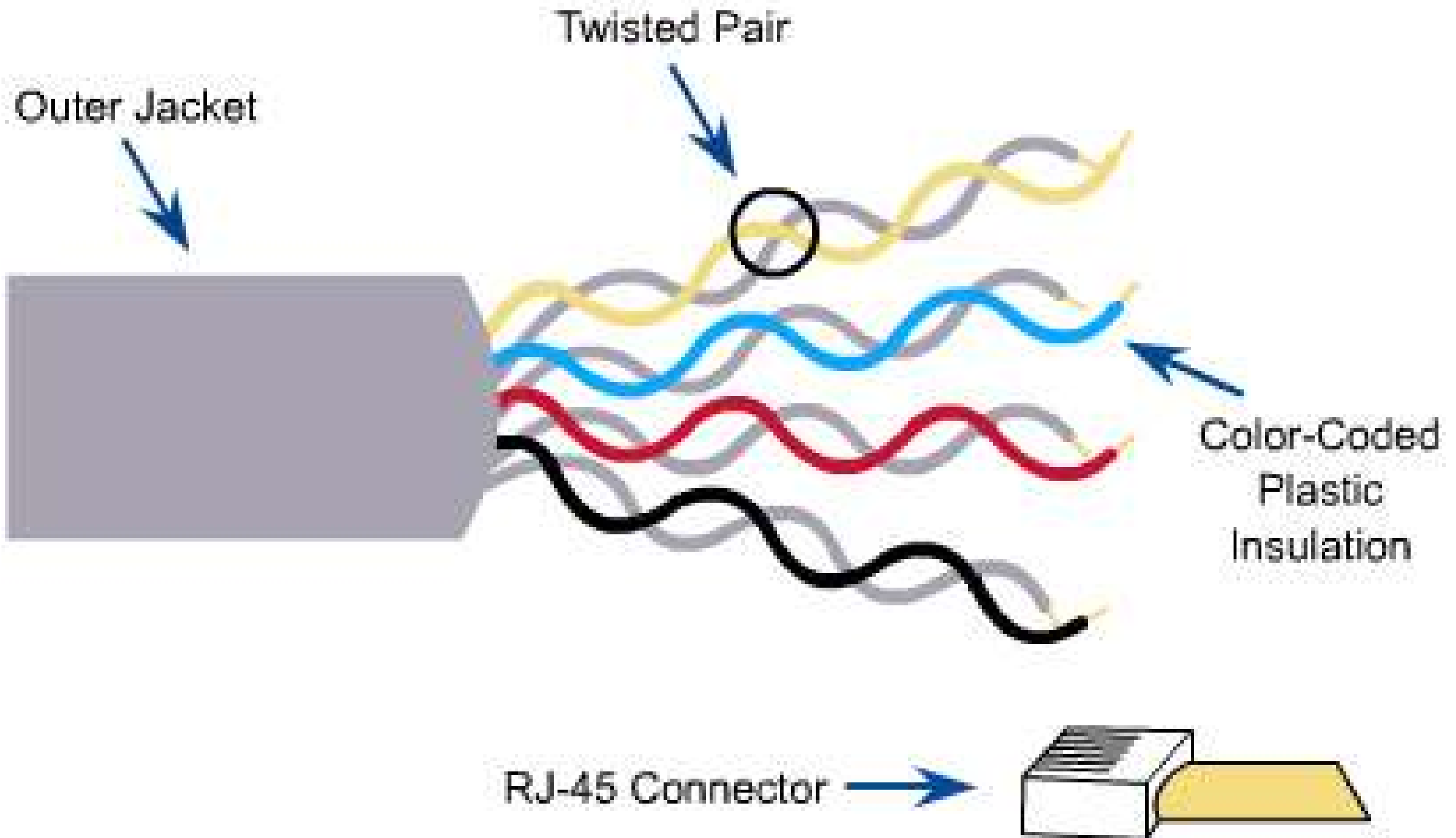
## IV.2 Dây mạng

- Cáp đồng trục – Coaxial cable
- Các đôi dây xoắn – Twisted pairs
  - UTP – Unshielded Twisted - Pair
  - STP – Shielded Twisted - Pair
- Cáp quang – Fiber optic

# Cáp đồng trục



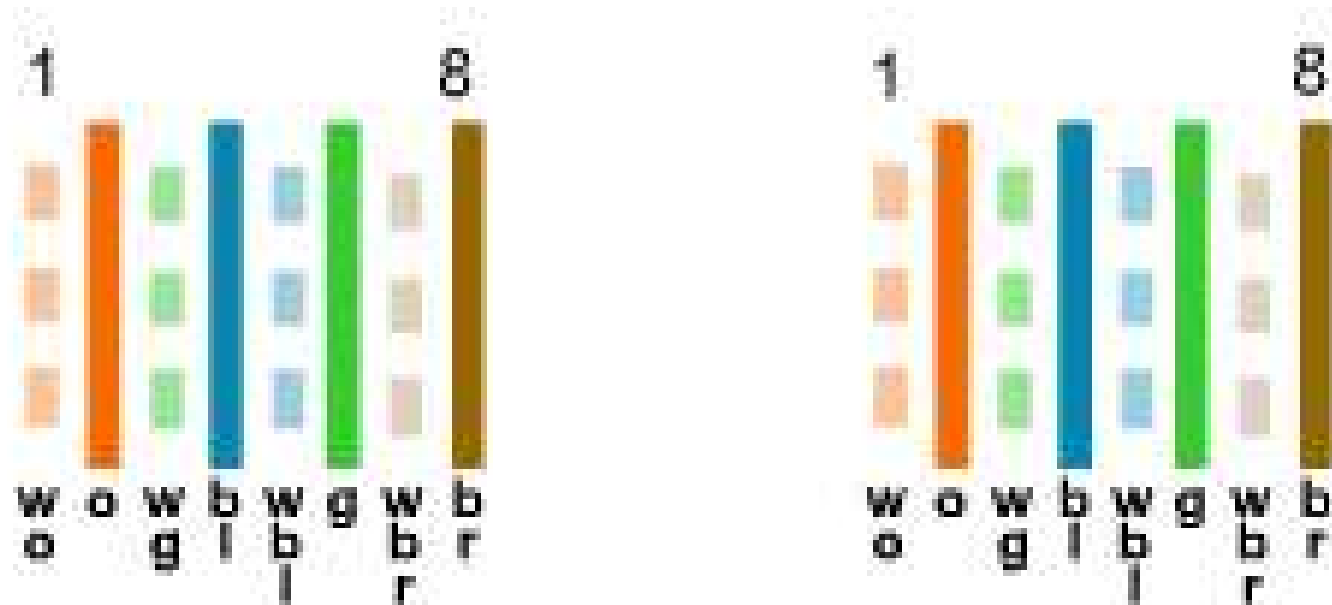
# UTP – Unshielded Twisted-Pair



# Dạng nối thẳng – Straight-Through

## Pin Label

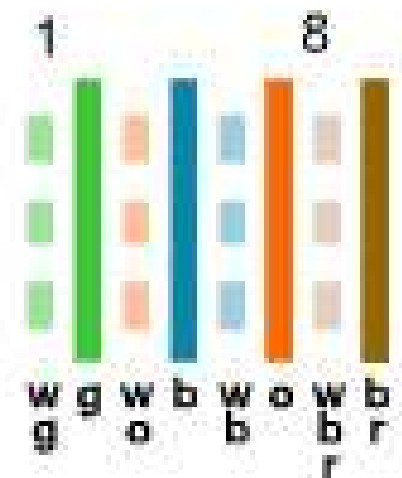
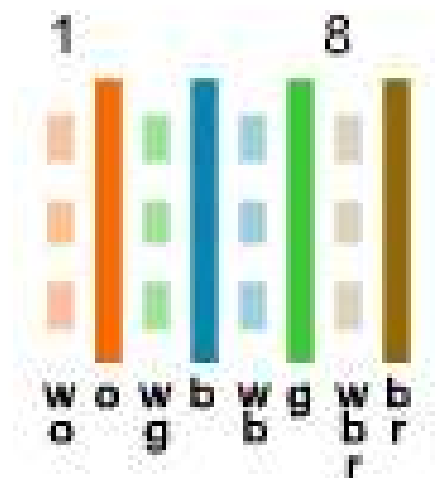
1	TD+
2	TD-
3	RD+
4	NC
5	NC
6	RD-
7	NC
8	NC



Wires on cable ends  
are in same order.

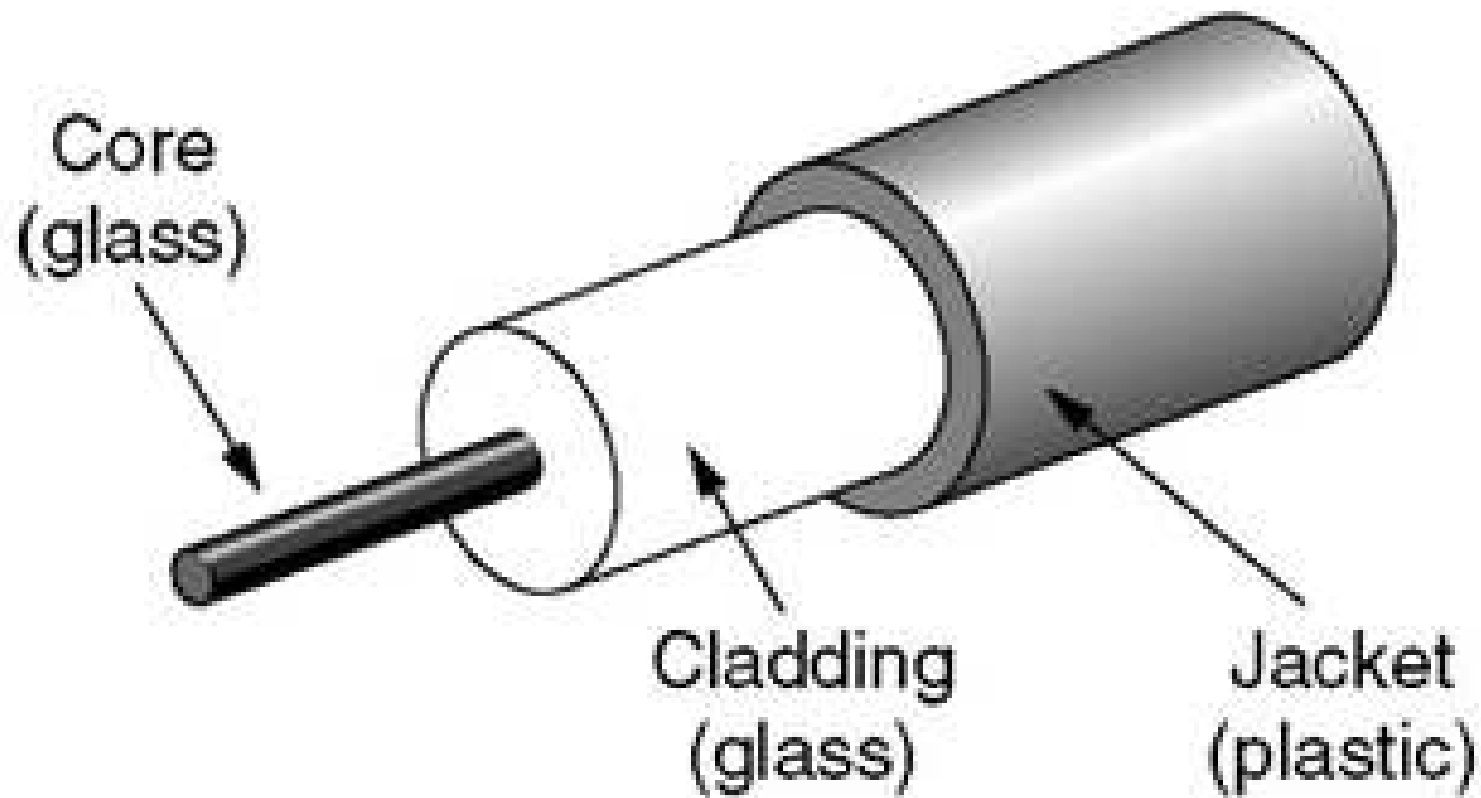
# Dạng nối chéo – Crossover

Pin Label		Pin Label	
1	TD+	1	TD+
2	RD-	2	RD-
3	RD+	3	RD+
4	NC	4	NC
5	NC	5	NC
6	TD+	6	TD-
7	NC	7	NC
8	NC	8	NC

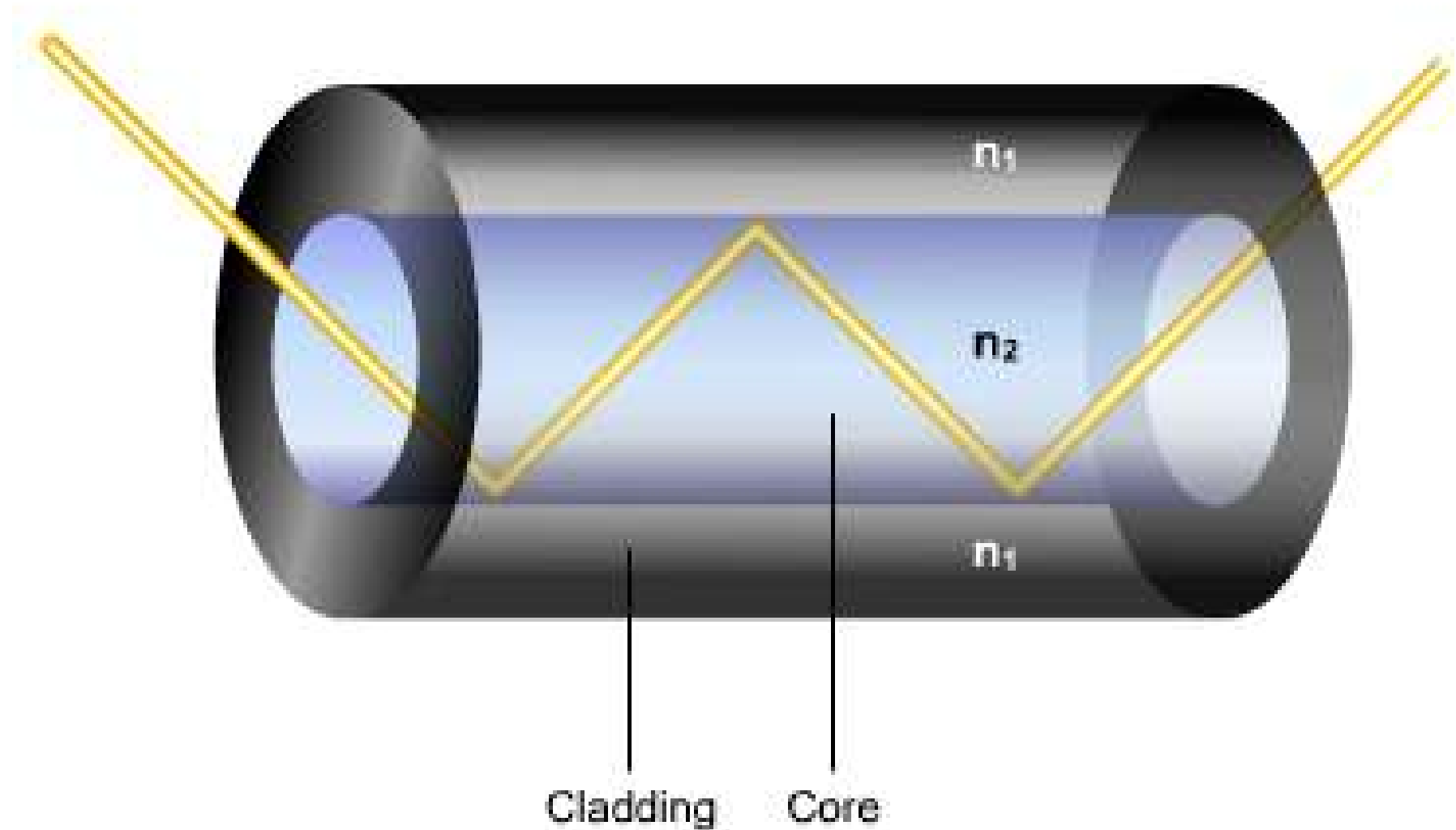


The orange wire pair and the green wire pair switch places on one end of the cable.

# Cáp quang



# Nguyên tắc phản xạ toàn phần trong cáp quang



# Đầu nối cáp quang





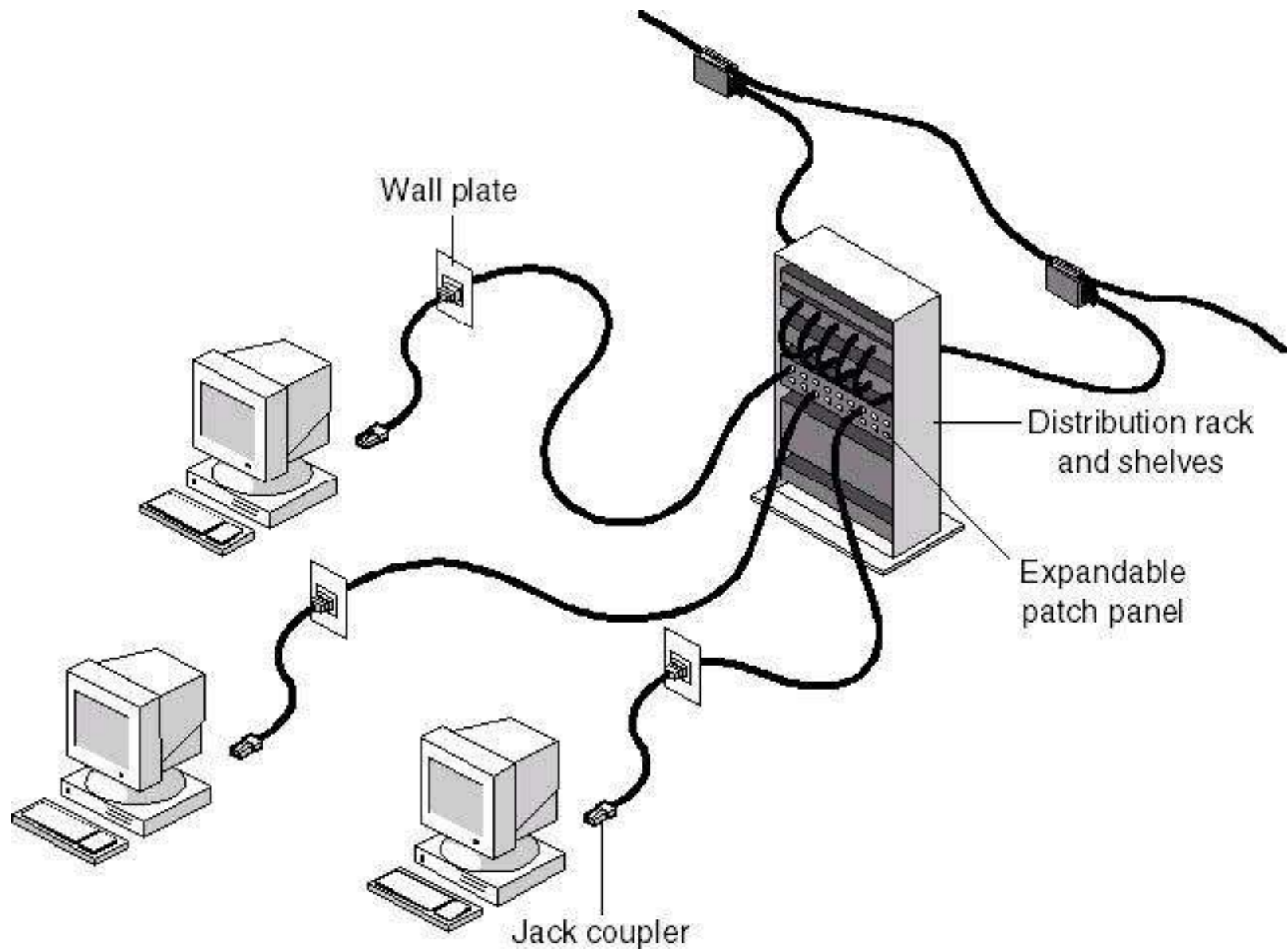
## IV.3 Các thiết bị kết nối

- Phụ thuộc loại mạng, sơ đồ kết nối
- Ví dụ:
  - Hub: điểm nối dây trên mạng cục bộ dạng Ethernet
  - Access Point trên mạng không dây

# Hub



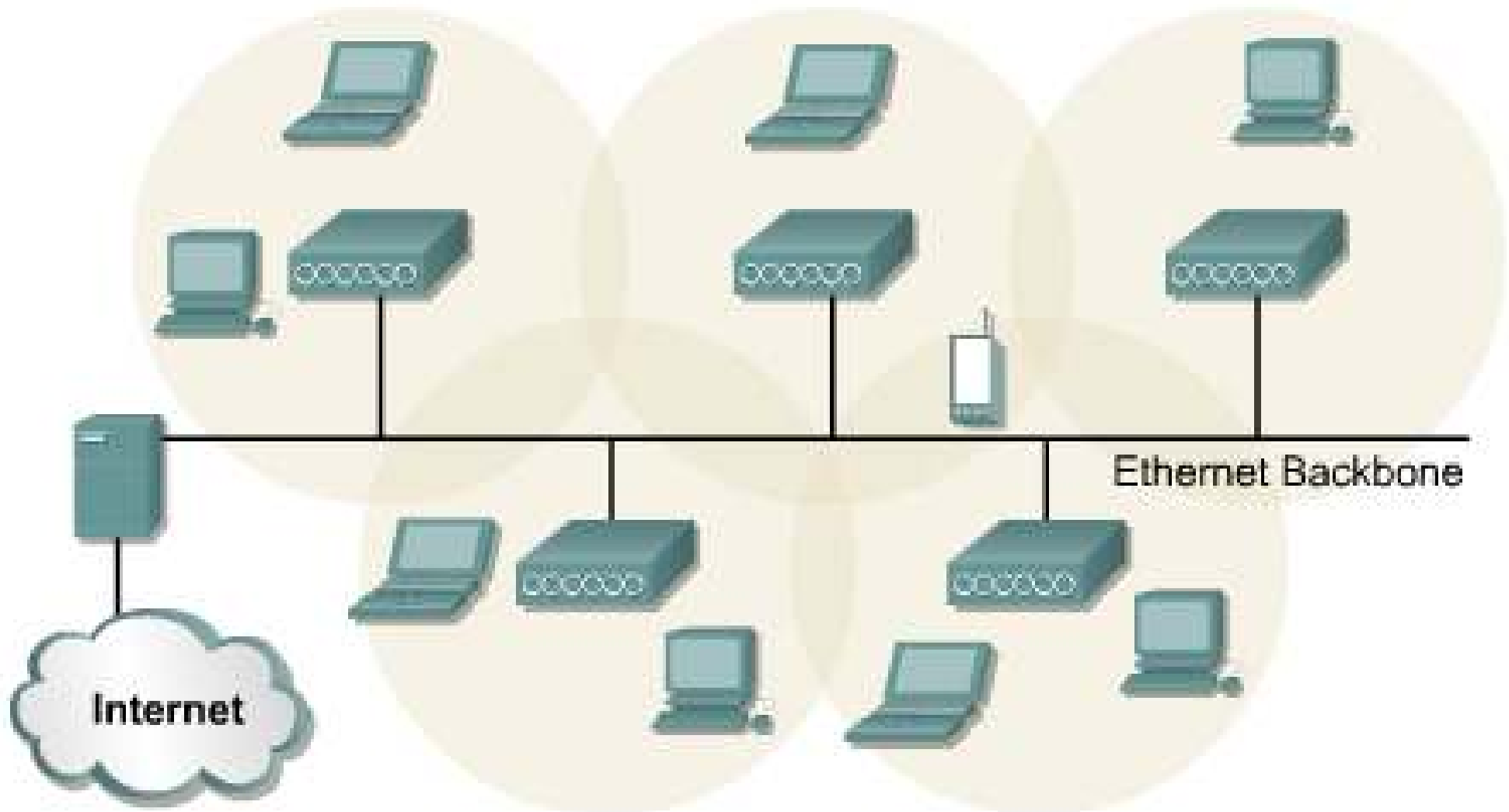
# Kết nối mạng dùng dây UTP



# Access point



# Kết nối mạng không dây



# **NHẬP MÔN MẠNG MÁY TÍNH**

## **Chương 2**

### **LỚP DATA LINK**

**(LỚP LIÊN KẾT DỮ LIỆU)**



# Nội dung chương 2

I. Các vấn đề thiết kế lớp data link

II. Các giao thức gửi nhận frame cơ bản

III. Các kỹ thuật kết nối mạng miền rộng

IV. Ví dụ giao thức lớp data link

Giao thức PPP



# I. Các vấn đề thiết kế lớp data link

I.1 Nhiệm vụ lớp data link

I.2 Các dịch vụ cung cấp cho lớp network

I.3 Các phương pháp tạo frame

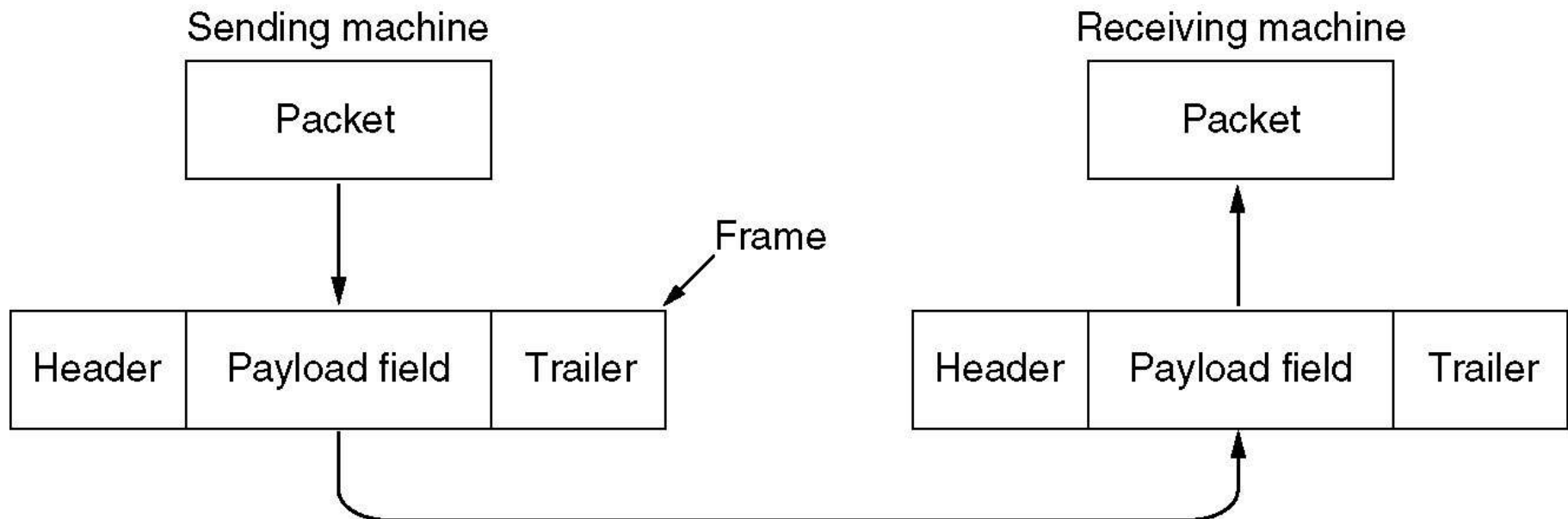
I.4 Các kỹ thuật kiểm soát lỗi

I.5 Kiểm soát lưu lượng



## I.1 Nhiệm vụ lớp data link

- Cung cấp dịch vụ gửi nhận dữ liệu (frame) tin cậy giữa hai máy láng giềng  
*hai máy láng giềng: hai máy có kết nối vật lý*
- Kiểm soát lỗi và kiểm soát lưu lượng



## I.2 Các dịch vụ cung cấp cho lớp network

- Gởi nhận không kiểm soát  
Unacknowledged connectionless service
- Gởi nhận có xác nhận của máy nhận  
Acknowledged connectionless service
- Gởi nhận có kết nối  
Acknowledged connection-oriented service



# Gửi nhận không kiểm soát

Máy gửi tạo frame và gửi cho máy nhận

## Gửi nhận có xác nhận của máy nhận

- Máy gửi tạo frame (data frame) và gửi cho máy nhận
- Máy nhận gửi trả frame khác (acknowledge frame, ACK) để xác nhận đã nhận được data frame.

## Gởi nhận có kết nối

- Máy gởi và máy nhận thiết lập kết nối (connection) trước khi trao đổi dữ liệu
- Mỗi frame được gởi trên kết nối có số thứ tự → không sai, không mất, không đảo lộn thứ tự.

Có ba giai đoạn trong gởi nhận frame:

- Thiết lập kết nối → khởi động biến, ...
- Gởi nhận frame
- Hủy kết nối → giải phóng bộ nhớ, ...

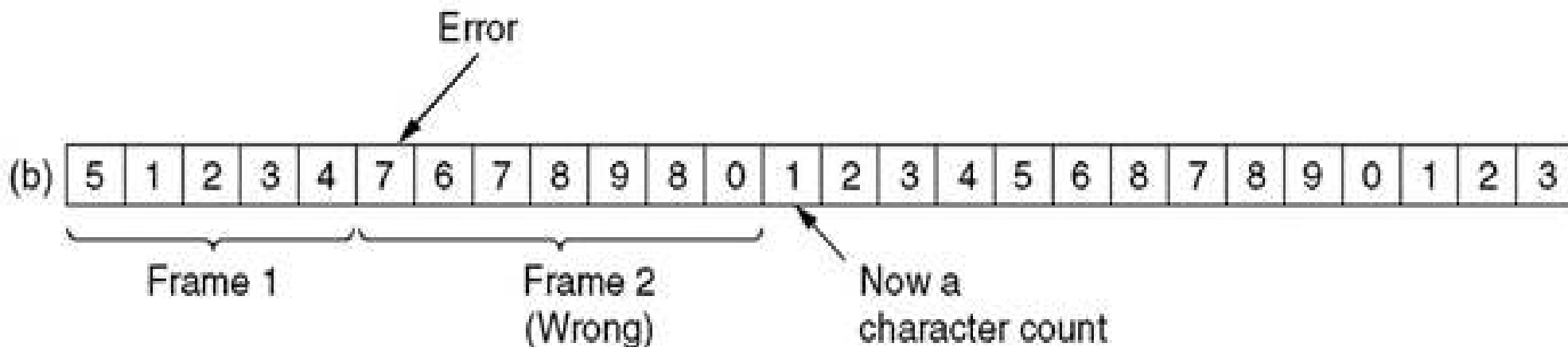
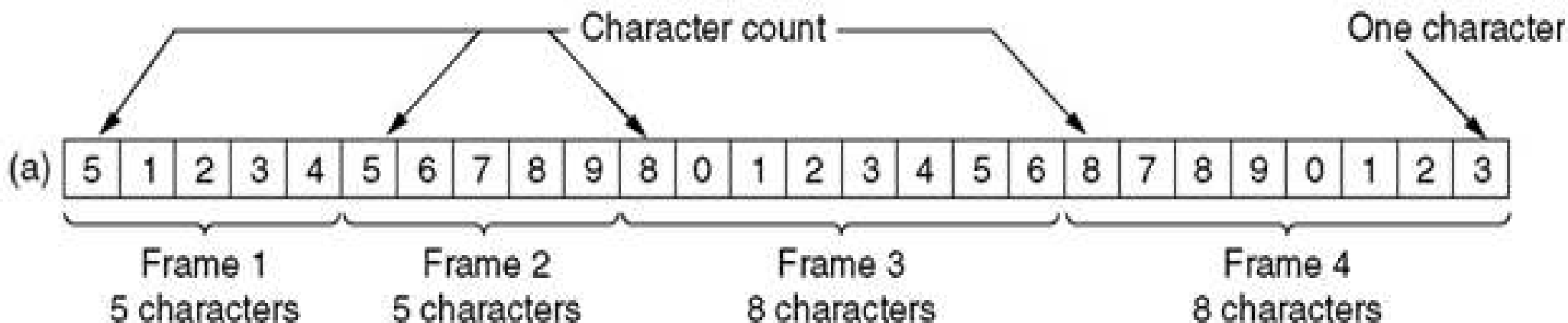
## I.3 Các phương pháp tạo frame

Có hai kỹ thuật cơ bản:

- Đếm ký tự trong frame
- Dùng các ký tự đặc biệt đánh dấu frame

Thực tế: dùng kết hợp hai kỹ thuật

# Tạo frame bằng cách đếm ký tự



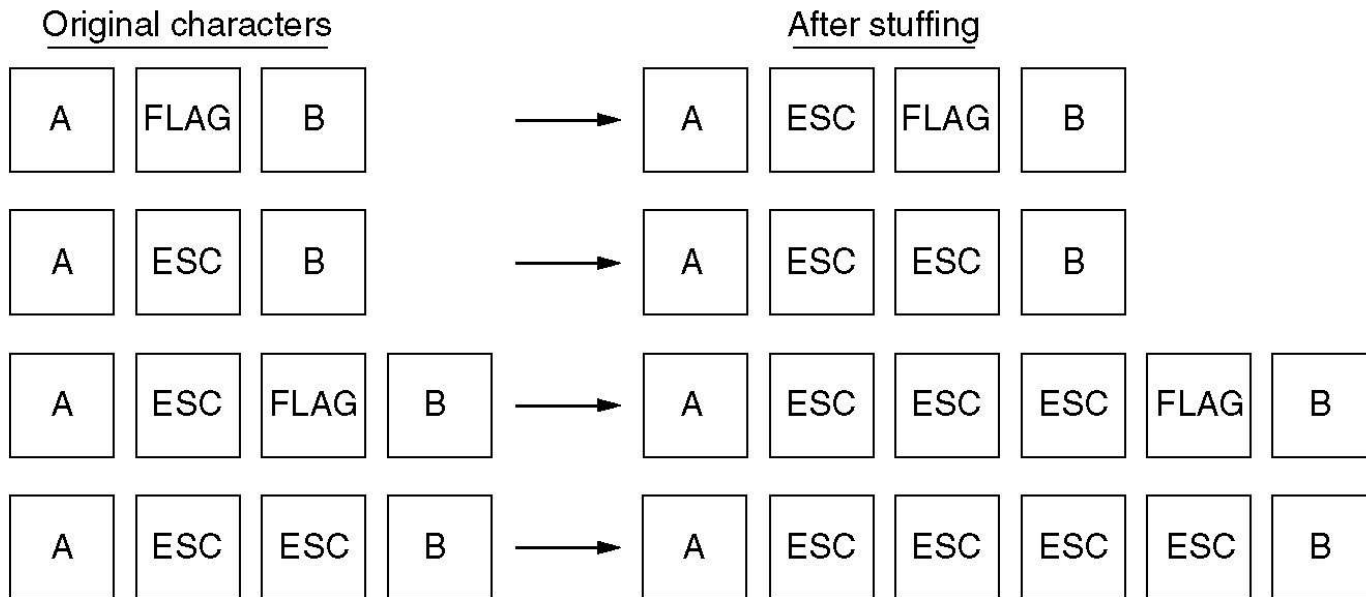
a. Không có lỗi

b. Có lỗi tại counter frame 2

# Tạo frame dùng ký tự đánh dấu (FLAG)



(a)



(b)

- Frame được đánh dấu bằng flag
- Ví dụ kỹ thuật chèn ký tự (character stuffing)



## I.4 Các kỹ thuật kiểm soát lỗi

- Dùng checksum
- Có xác nhận của máy nhận (ACK)
- Định thời (timer)
- Số thứ tự trình tự (sequence number)

Kiểm soát lỗi bảo đảm việc gửi nhận frame:

- không sai
- không mất
- không đảo lộn thứ tự.

# Checksum

- Máy gửi tạo frame và tính checksum
- Máy gửi sẽ gửi frame có checksum
- Nhờ vùng checksum máy nhận xác định frame không có lỗi

Với checksum → không sai

# ACK

- Khi nhận một frame không có lỗi thì máy nhận sẽ gửi một frame điều khiển (ACK) cho máy gửi để xác nhận
- Nếu không có ACK thì máy gửi sẽ gửi lại frame

Với ACK → không mất

# Timer

- Sau khi gửi frame, máy gửi khởi động một bộ định thời (timer)
- Nếu hết thời gian (timeout) mà không có ACK từ máy nhận thì máy gửi sẽ gửi lại frame

# Sequence number

- ACK từ máy nhận có thể không đến máy gửi, và máy gửi sẽ gửi lại frame
  - Máy nhận có thể nhận cùng một frame nhiều lần
  - Để tránh nhận trùng frame, mỗi frame có một số thứ tự
  - Số thứ tự frame thuộc về một khoảng giá trị xác định → số thứ tự trình tự
- Ví dụ: dùng số thứ tự 3 bit → có số thứ tự từ 0 đến 7

## I.5 Kiểm soát lưu lượng

- Mục đích: máy gửi không nhanh hơn máy nhận
- Hai kỹ thuật cơ bản:
  - Máy gửi chờ ACK từ máy nhận
  - Máy gửi hoạt động theo tốc độ giới hạn

## II. Các giao thức gửi nhận frame cơ bản

- Giao thức đơn giản trên đường truyền 1 chiều lý tưởng
- Giao thức stop-and-wait
- Giao thức trên đường truyền 1 chiều thực tế
- Các giao thức dạng sliding window

## ■ Giao thức đơn giản

trên đường truyền 1 chiều lý tưởng

Đường truyền lý tưởng:

- Không có lỗi → không cần kiểm soát lỗi
- Máy nhận tốc độ vô hạn → không cần kiểm soát lưu lượng

Đường truyền 1 chiều:

- dữ liệu 1 chiều từ máy gửi đến máy nhận
- simplex

*Máy gửi tạo frame và gửi cho máy nhận*



# Giao thức stop-and-wait

Đường truyền: không có lỗi và máy nhận tốc độ hữu hạn

- *Máy gửi tạo frame gửi đến máy nhận*
- *Máy gửi chờ ACK từ máy nhận*
- *Máy gửi gửi frame tiếp theo*

Đường truyền 1 chiều dữ liệu nhưng có chiều truyền ACK

→ 2 chiều không đồng thời: half-duplex

## ■ Giao thức đơn giản

trên đường truyền 1 chiều thực tế

Đường truyền thực tế:

- Có thể có lỗi
- Máy nhận tốc độ hữu hạn
- *Máy gửi tạo frame, tính checksum, ghi số thứ tự frame, khởi động timer, gửi đến máy nhận*
- *Nếu có ACK thì gửi frame tiếp theo*
- *Nếu không có ACK thì gửi lại frame*

# Các giao thức dạng sliding window

Mục tiêu:

- Sử dụng đường truyền với 2 chiều dữ liệu  
→ full-duplex
- Gởi nhận theo nhóm frame

Khái niệm cơ bản:

- Piggybacking: chờ gởi kèm ACK với frame dữ liệu tiếp theo
- Sliding Window

## Sliding window – cửa sổ trượt

Một máy sẽ gửi một nhóm frame trước khi chờ ACK.

Danh sách số thứ tự các frame đã gửi chưa có ACK thuộc sending window

Tương tự, danh sách số thứ tự các frame chờ nhận thuộc về receiving window

Ví dụ: sequence number 3 bit

Gửi các frame 1 đến 4      0 1 2 3 4 5 6 7

Nhận ACK frame 1      0 1 2 3 4 5 6 7

Gửi tiếp frame 5      0 1 2 3 4 5 6 7

## Sliding window (tt)

Kỹ thuật sliding window còn được dùng trên giao thức TCP (chương 5)

Máy gửi truyền mỗi chuỗi dữ liệu không cần chờ ACK của từng dữ liệu

Máy nhận có thể nhận dữ liệu chưa đúng thứ tự và sắp xếp lại trong khi chờ dữ liệu khác

Nếu không có ACK sau 1 khoảng thời gian thì dữ liệu sẽ được gửi lại

# Các giao thức dạng sliding window cơ bản

Có hai dạng cơ bản với cách xử lý frame có lỗi (mất, checksum error) khác nhau

## ■ Go back n

- Máy gởi sẽ gởi lại tất cả các frame từ frame có lỗi

## ■ Selective Repeat

- Máy gởi chỉ gởi lại frame có lỗi
- Máy nhận phải lưu lại các frame tốt sau frame có lỗi

### III. Các kỹ thuật kết nối mạng miền rộng

- Dùng đường dây điện thoại
- Kết nối trực tiếp dùng cable
- ISDN  
(Integrated Services Digital Network)
- Kết nối không dây
- Kết nối qua vệ tinh



# Kết nối dùng đường dây điện thoại

- Dạng quay số (Dial-up)
- DSL (Digital Subscriber Line)



# Dạng quay số

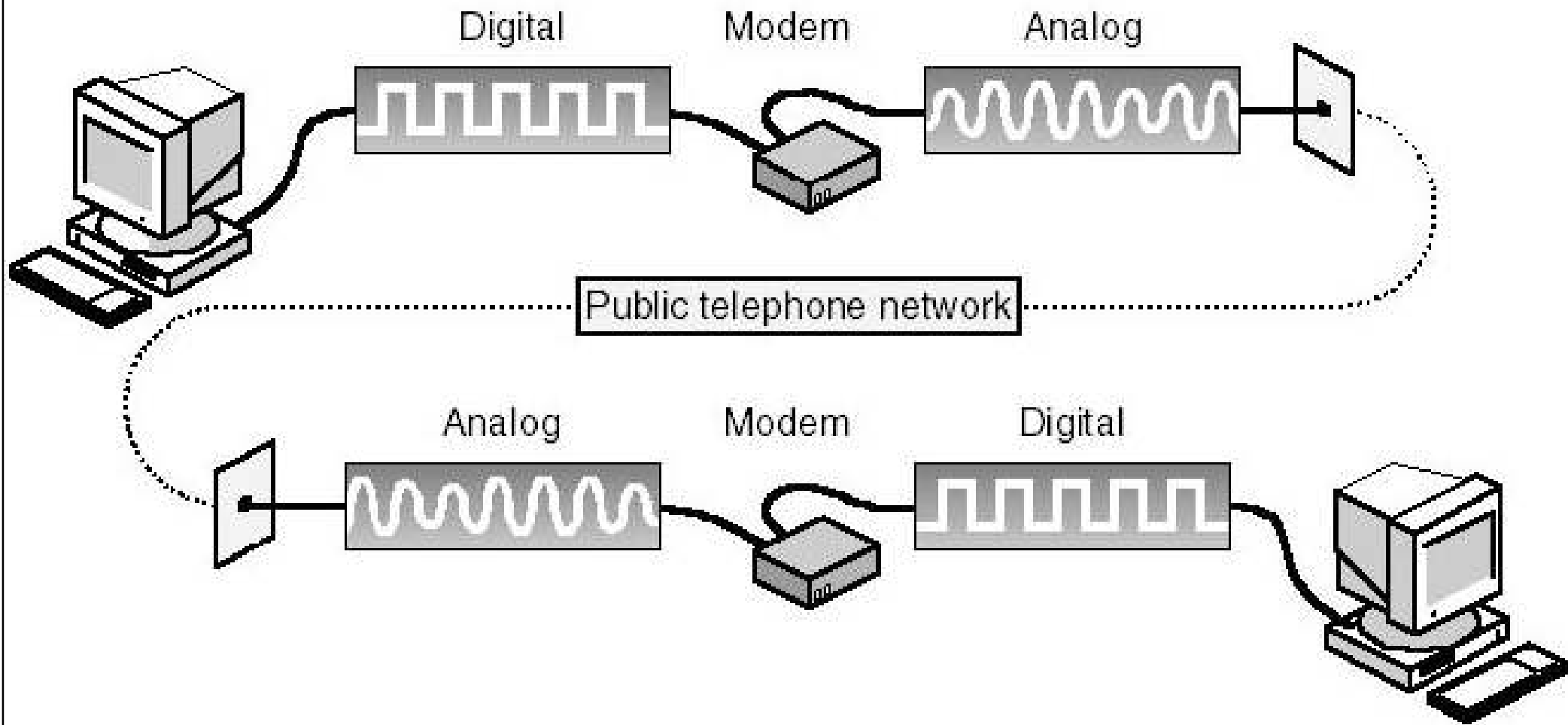
Thiết bị:

- Người sử dụng: modem
- Nhà cung cấp dịch vụ: modem

Giao thức thông dụng: PPP  
(Point-to-Point Protocol)

Phần mềm: tích hợp trên các hệ điều hành

# Modem



Modem thực hiện điều chế (M**O**dulation) khi gửi và giải điều chế (DE**M**odulation) khi nhận

# Modem (tt)

## Các dạng modem:

- Internal – mạch điều khiển gắn trong máy
  - Kết nối với I/O bus, ví dụ PCI
  - Tích hợp trên mainboard
- External – Thiết bị đặt ngoài kết nối qua cổng COM hay USB

Một số tiêu chuẩn modem theo ITU:

V34 – tốc độ 28.800 bps (bits per second)

V90 – tốc độ 56.600 bps

# Digital Subscriber Line - DSL

- Dùng chung kết nối mạng trên đường dây điện thoại
- Không có quay số → kết nối thường trực
- Tốc độ cao hơn so với dùng modem

## DSL (tt)

Có các dạng:

- ADSL-Asymmetric DSL: thông dụng  
Tốc độ download: 384Kbps → 8Mbps  
Tốc độ upload: 64Kbps → 1 Mbps  
Có giới hạn về khoảng cách ~ 5.500 mét
- SDSL-Symmetric DSL  
Tốc độ download và upload đến 3Mbps
- VDSL-Very High Data Rate DSL  
Tốc độ download và upload đến 52Mbps

# ADSL

## Thiết bị

- Người sử dụng:

ADSL modem/ ADSL router

- Nhà cung cấp dịch vụ: Access Multiplexer

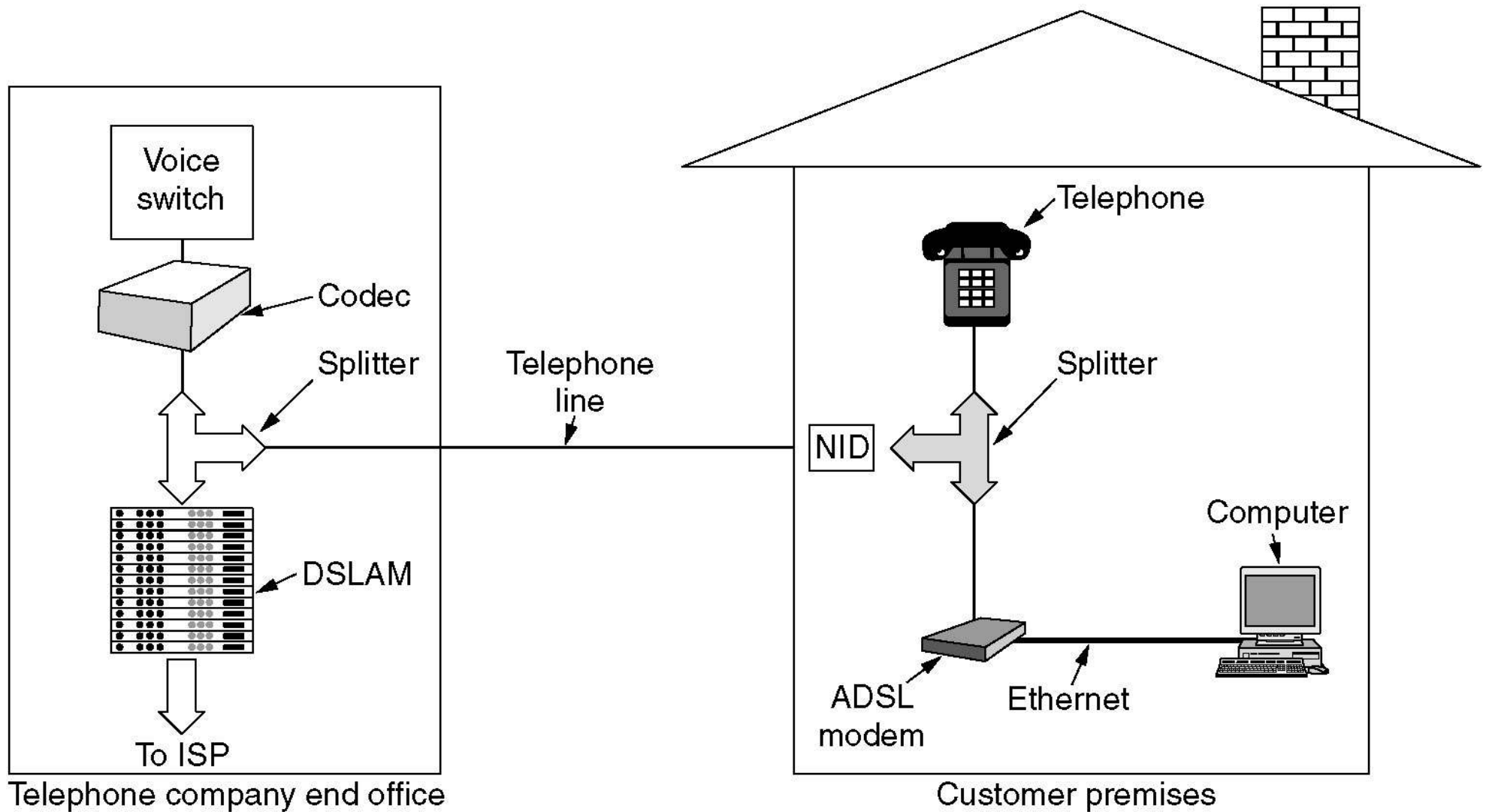
## Phần mềm:

- Người sử dụng dùng phần mềm do nhà sản xuất thiết bị cung cấp

- Nhà cung cấp dịch vụ thường dùng kỹ thuật ATM (Asynchronous Transfer Mode)

ADSL là tiêu chuẩn của lớp vật lý

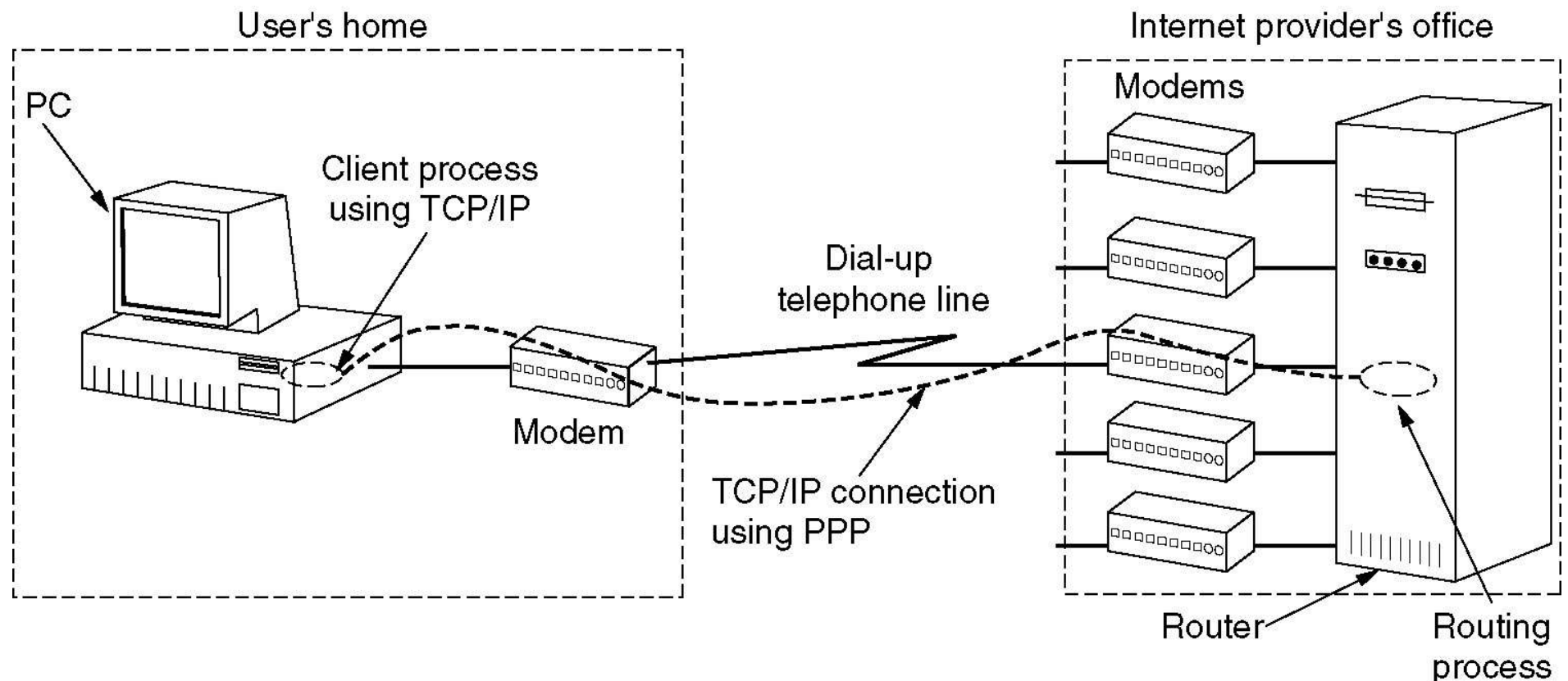
# ADSL (tt)



Cấu hình cơ bản dùng ADSL

# IV. Giao thức PPP (Point-to-Point Protocol)

Giao thức PPP dùng trong kết nối giữa máy tính cá nhân (PC) với nhà cung cấp dịch vụ Internet (ISP) qua đường điện thoại quay số.





# Các đặc điểm của giao thức PPP

- Tạo frame theo giao thức HDLC (High-level Data Link Control), dùng kỹ thuật chèn ký tự, có kiểm soát lỗi
- Dùng giao thức LCP (Link Control Protocol) để kiểm soát kết nối, thoả thuận tham số...
- Dùng giao thức NCP (Network Control Protocol) để thiết lập tham số cho lớp Network, dùng được với nhiều loại mạng như TCP/IP, IPX/SPX, NetBEUI, Apple Talk

## Các bước máy PC kết nối Internet

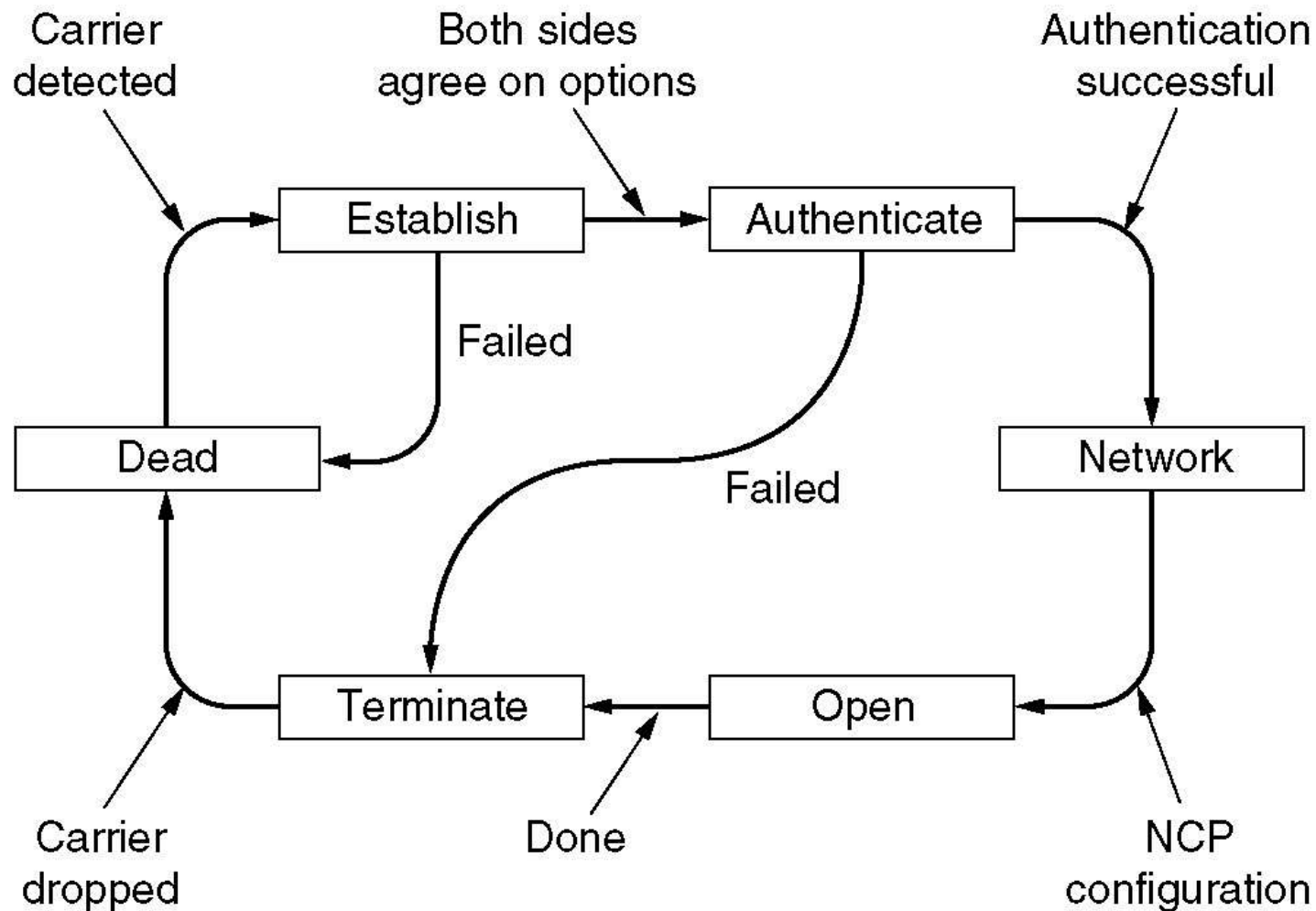
- Máy PC thiết lập kết nối vật lý với ISP bằng cách quay số qua modem
- Máy PC gửi một chuỗi LCP packet trên các PPP frame để thoả thuận tham số
- Máy PC gửi một chuỗi NCP packet trên các PPP frame để thiết lập cấu hình hoạt động lớp network
  - máy PC được cấp một địa chỉ IP động và trở thành Internet host, có thể gửi nhận dữ liệu theo các IP packet.

## Các bước máy PC kết nối Internet (tt)

Khi kết thúc phiên làm việc:

- Dùng các NCP packet để hủy kết nối lớp network và trả lại địa chỉ IP
- Dùng các LCP packet hủy kết nối lớp data link
- Hủy kết nối vật lý bằng lệnh ngắt modem ra khỏi đường dây điện thoại

# Các bước máy PC kết nối Internet(tt)



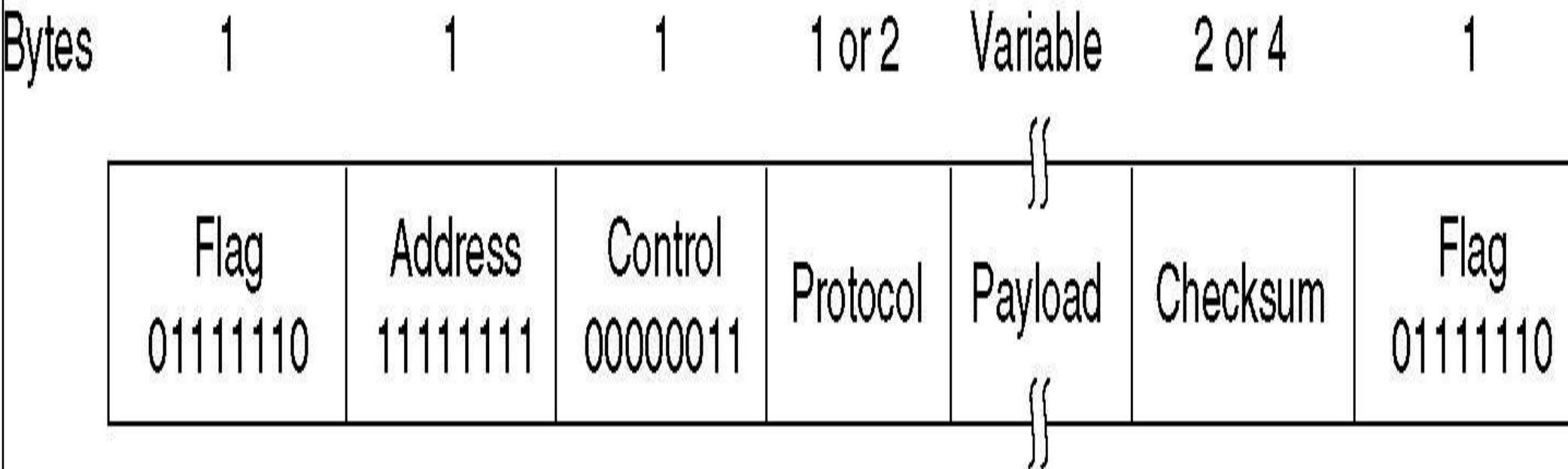
Các giai đoạn hoạt động theo giao thức PPP

## Ví dụ PPP frame

Dạng frame điều khiển không có số thứ tự

- Flag: đánh dấu đầu và cuối frame
- Address và Control: hằng số đối với control frame
- Protocol: xác định loại dữ liệu trong vùng payload
- Payload: dữ liệu, kích thước do thoả thuận, mặc định là 1500 bytes
- Checksum: dùng kiểm soát lỗi

# PPP Frame (tt)



# NHẬP MÔN MẠNG MÁY TÍNH

## Chương 3

### LỚP MAC

(LỚP CON ĐIỀU KHIỂN

TRUY CẬP MÔI TRƯỜNG)



# Nội dung chương 3

- I. Khái niệm lớp MAC
- II. Vấn đề cấp phát kênh truyền
- III. Giao thức CSMA/CD
- IV. Giới thiệu các tiêu chuẩn IEEE 802.x
- V. Giới thiệu về Bridge, Switch



# I. Khái niệm lớp MAC

Lớp Physical và Data link (mô hình OSI):  
giải quyết vấn đề các máy đồng thời truy cập đường truyền dạng broadcast (quảng bá)

Dự án IEEE 802: các đặc tả của 2 lớp này trên mạng cục bộ

→ tiêu chuẩn mạng cục bộ

# Lớp Data Link trong IEEE 802

7. Application layer

6. Presentation layer

5. Session layer

4. Transport layer

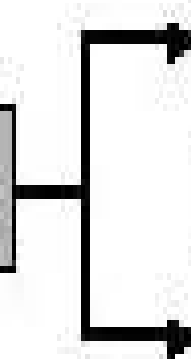
3. Network layer

2. Data-link layer

1. Physical layer

Logical Link Control (LLC)

Media Access Control (MAC)

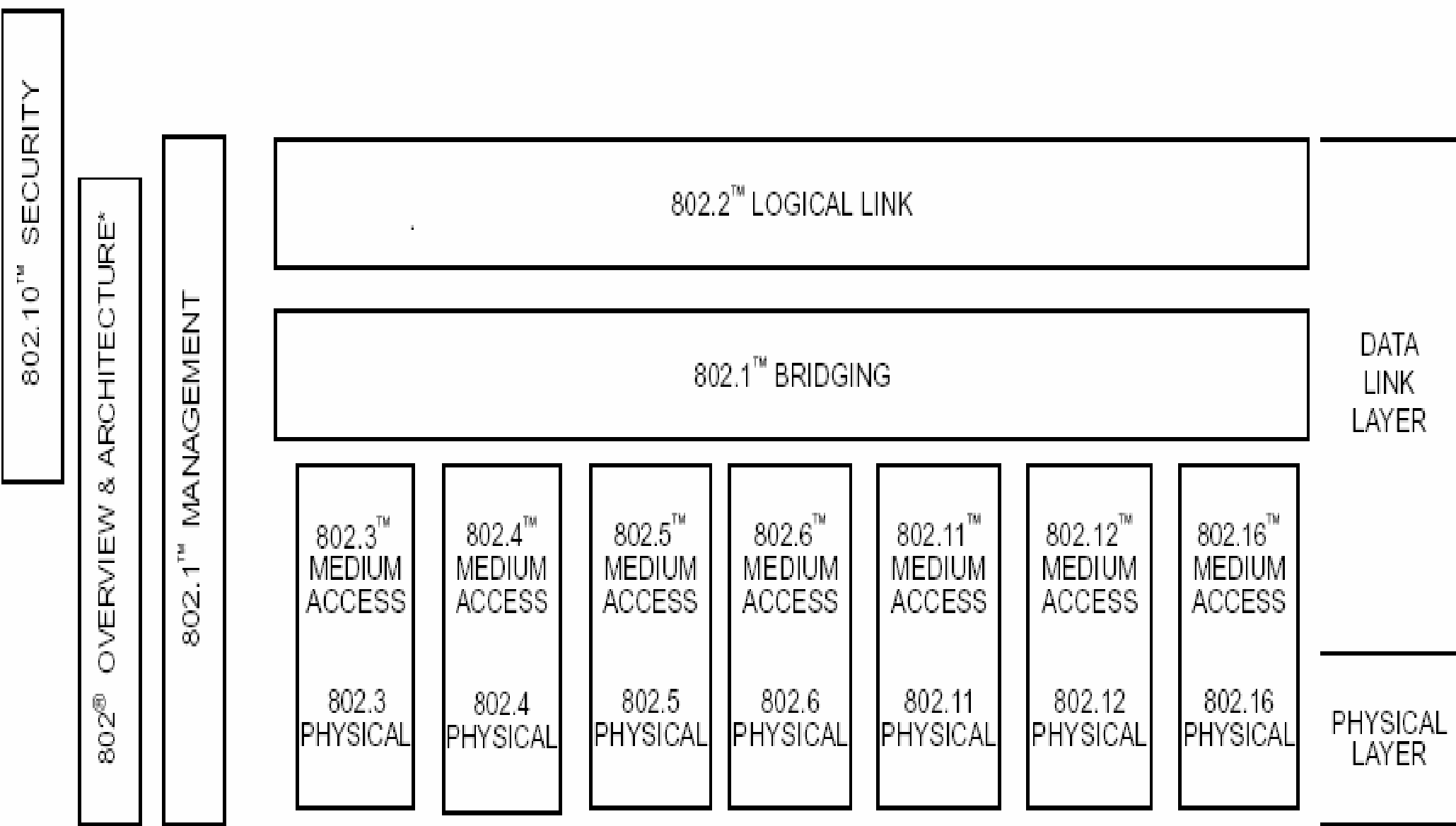


## Lớp Data Link trong IEEE 802 (tt)

Gồm 2 lớp con (sublayer):

- Logical Link Control (LLC): thiết lập và kết thúc liên kết, quản lý truyền frame
- Medium Access Control (MAC): quản lý truy cập đường truyền, tạo frame, kiểm soát lỗi, xác định địa chỉ

# Các tiêu chuẩn IEEE 802.x



\* Formerly IEEE Std 802.1A.

## Các tiêu chuẩn IEEE 802.x chính

- 802.2 - *Logical Link Control*
- 802.3 - *CSMA/CD Access Method and Physical Layer Specifications*
- 802.5 - *Token Ring Access Method and Physical Layer Specifications*
- 802.11 - *Wireless LAN Medium Access Control (MAC) Sublayer and Specifications*
- 802.16 - *Standard Air Interface for Fixed Broadband Wireless Access Systems*



## II. Vấn đề cấp phát kênh truyền

Mục đích: cấp phát một kênh truyền dạng quảng bá cho nhiều máy cùng sử dụng

## Một số thuật ngữ

- Đường truyền (Transmission line): vật lý
- Kênh truyền (Communication channel): luận lý
- Baseband: một kênh truyền trên đường truyền
- Broadband: nhiều kênh truyền trên đường truyền
- Multiplexing: ghép kênh tại nơi gửi
- Demultiplexing: tách kênh tại nơi nhận

# Các kỹ thuật cấp phát kênh truyền

- Cấp phát tĩnh: số kênh truyền cố định
- Cấp phát động: số kênh truyền thay đổi
  - một máy truy cập đường truyền không làm ảnh hưởng các máy khác



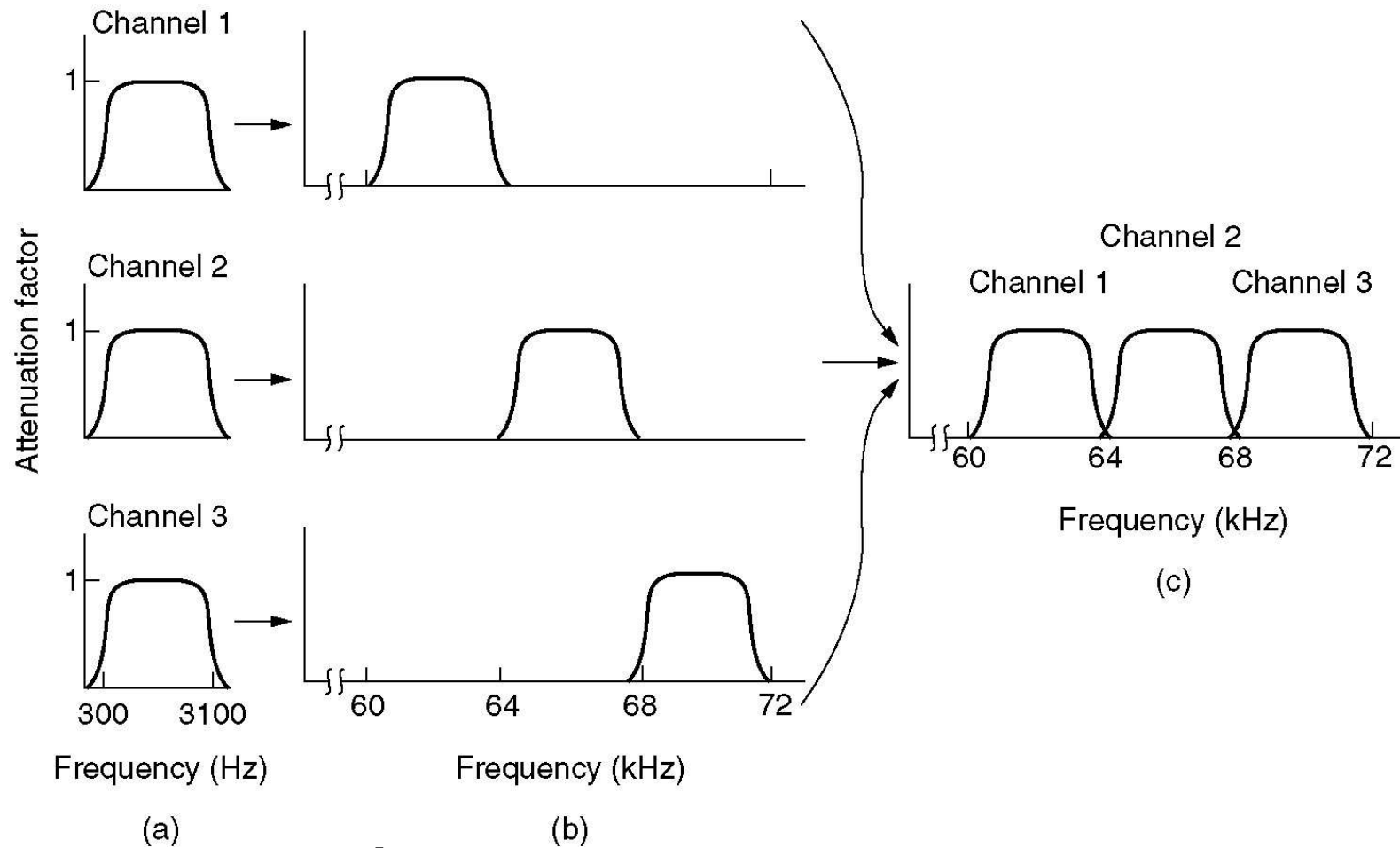
# Cấp phát tĩnh kênh truyền

Hai kỹ thuật thông dụng:

- FDM – Frequency Division Multiplexing  
(Ghép kênh phân chia theo tần số)
- TDM – Time Division Multiplexing  
(Ghép kênh phân chia theo thời gian)

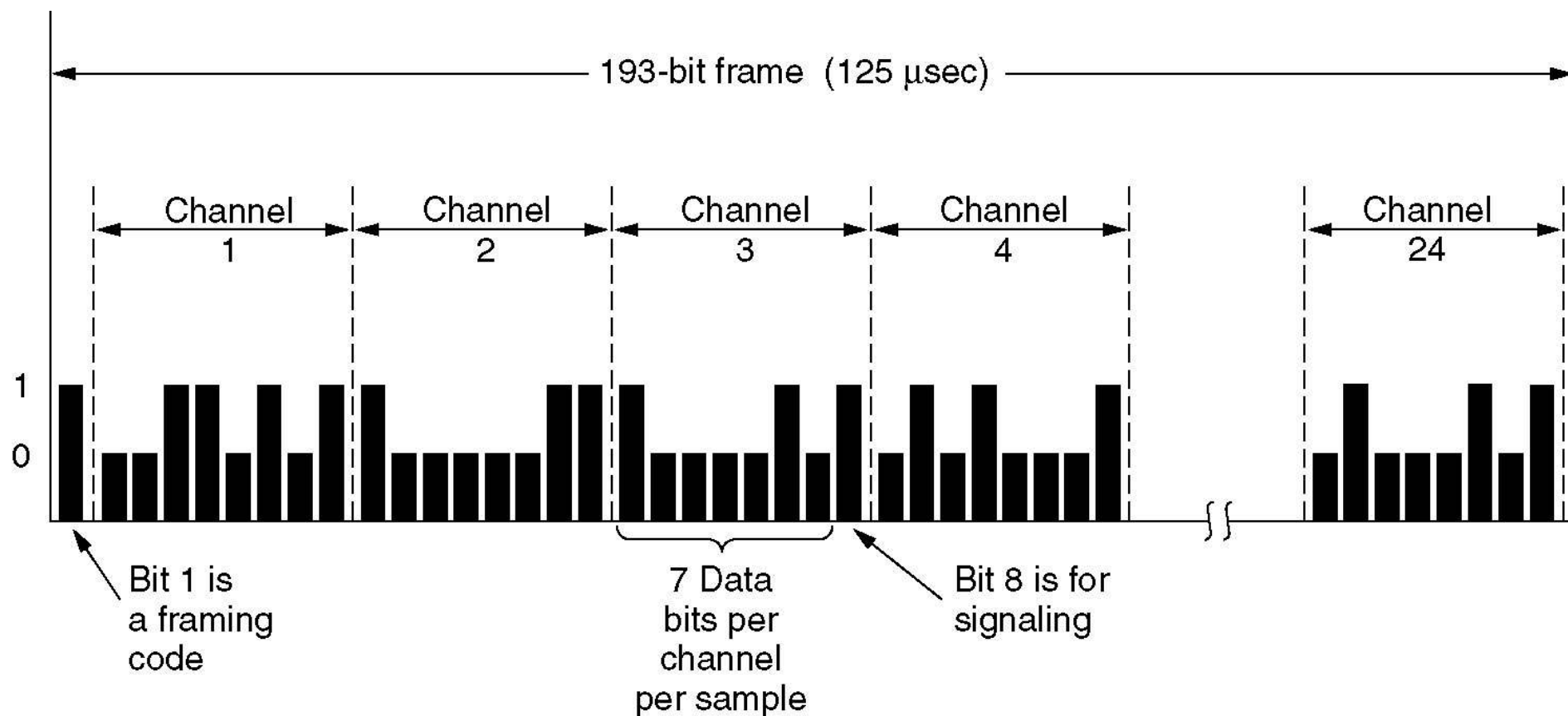
Ứng dụng: mạng điện thoại cố định

# Ví dụ FDM



- Băng thông gốc
- Băng thông được nâng tần số
- Kênh sau khi ghép

# Ví dụ TDM



Ghép 24 kênh thoại trong 1 kênh T1

# Cấp phát động kênh truyền

- Ứng dụng trong mạng máy tính, mạng điện thoại
- Có nhiều giao thức: ALOHA, CSMA, WDMA, ...

# Môi trường cấp phát động kênh truyền

- Mô hình trạm (station model)
  - Có N trạm (máy tính, điện thoại) có thể tạo và truyền frame
- Kênh truyền đơn (single channel)
  - Các trạm dùng chung 1 đường truyền
- Xung đột (collision)
  - Nếu 2 trạm truyền frame đồng thời
  - Tất cả trạm có thể phát hiện xung đột
  - Không có kết quả

# Môi trường cấp phát động kênh truyền (tt)

- Thời gian liên tục – Continuous time
  - Truyền frame tại thời điểm bất kỳ
- Thời gian được phân khe – Slotted time
  - Thời gian được chia thành các khe (slot)
  - Truyền frame tại thời điểm bắt đầu một khe thời gian
- Cảm nhận truyền tải – Carrier sense
  - Các trạm có thể xác định kênh truyền đang được sử dụng

# Môi trường cấp phát động kênh truyền (tt)

- Không cảm nhận truyền tải – No carrier sense
  - Các trạm không thể xác định kênh truyền đang được sử dụng



## III. Giao thức CSMA/CD

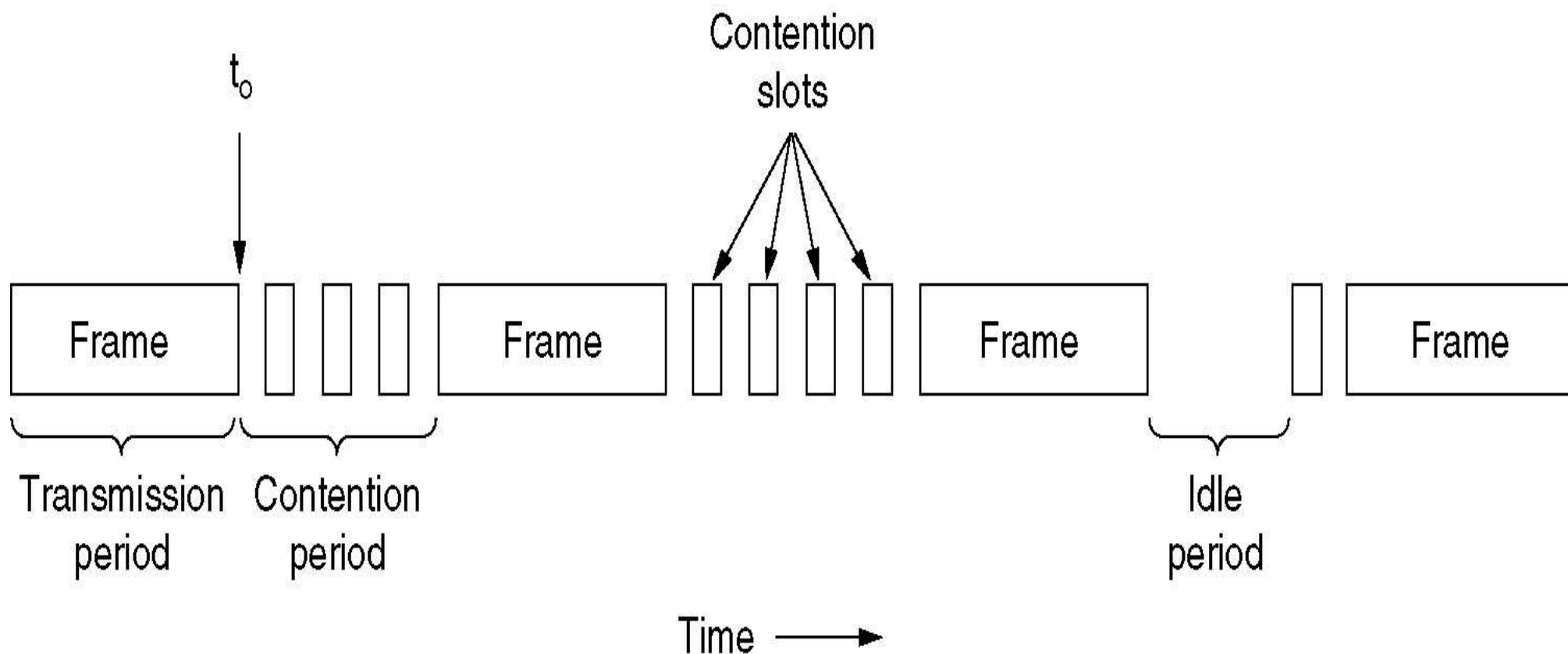
Carrier Sense Multiple Access with  
Collision Detection

(Đa truy cập cảm nhận truyền tải có phát  
hiện xung đột)

Dùng trong tiêu chuẩn mạng IEEE 802.3



# CSMA/CD (tt)



Ba trạng thái của đường truyền: Transmission (truyền), Contention (tranh chấp), Idle (ngủ)<sup>19</sup>

## Hoạt động khi cần truyền frame

- Kiểm tra trạng thái đường truyền (cảm nhận truyền tải)
- Nếu đường truyền rảnh thì truyền frame

# Xung đột và xử lý xung đột

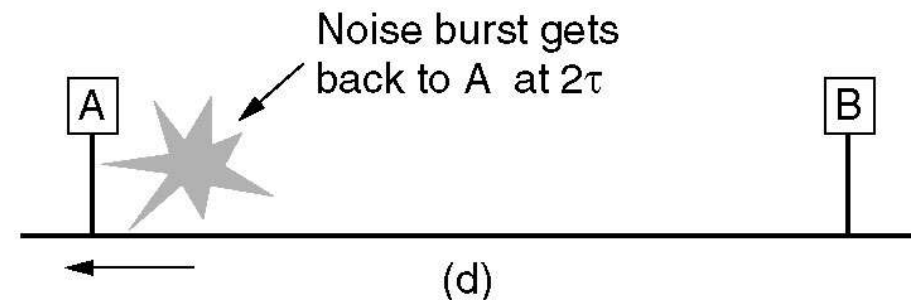
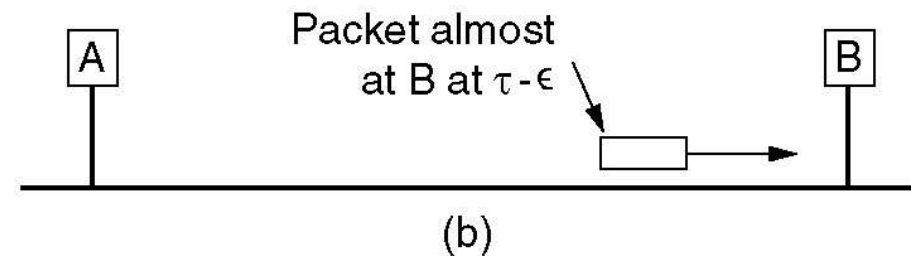
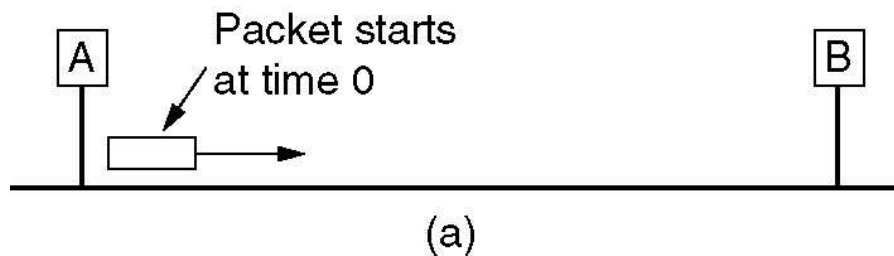
## Xung đột:

- Nếu có 2 máy truyền đồng thời thì tạo xung đột
- Xung đột được phát hiện bởi phần cứng

## Xử lý xung đột:

- Hủy frame đã truyền
- Chờ một khoảng thời gian ngẫu nhiên
- Kiểm tra đường truyền, nếu rảnh thì truyền lại

# Thời gian để phát hiện xung đột



A, B: 2 máy xa nhau nhất trên mạng

Tau ( $\tau$ ): thời gian truyền giữa A, B

→ A phải truyền frame trong thời gian  $\geq 2\tau$

## IV. Giới thiệu các tiêu chuẩn IEEE 802

1. Mạng Ethernet – 802.3
2. Mạng Fast Ethernet
3. Mạng Gigabit Ethernet
4. Mạng Token Ring – 802.5
5. Mạng Wireless Lan – 802.11
6. IEEE 802.2 – Logical Link Control  
(LLC, Điều khiển liên kết luận lý)



# 1. Mạng Ethernet – 802.3

- a. Giới thiệu mạng Ethernet
- b. Nối cáp
- c. Mã hoá bit
- d. Giao thức lớp MAC
- e. Giải quyết xung đột

## a. Giới thiệu mạng Ethernet

- Xuất phát từ mạng LAN dạng CSMA/CD 2.94 Mbps của Xerox, mạng Ethernet
- 1978, DEC, Intel, Xerox thiết lập tiêu chuẩn mạng Ethernet 10 Mbps, chuẩn DIX
- 1983, chuẩn DIX trở thành IEEE 802.3
- Mạng Ethernet tiếp tục phát triển với các tốc độ cao hơn 100 Mbps, 1000 Mbps, ...

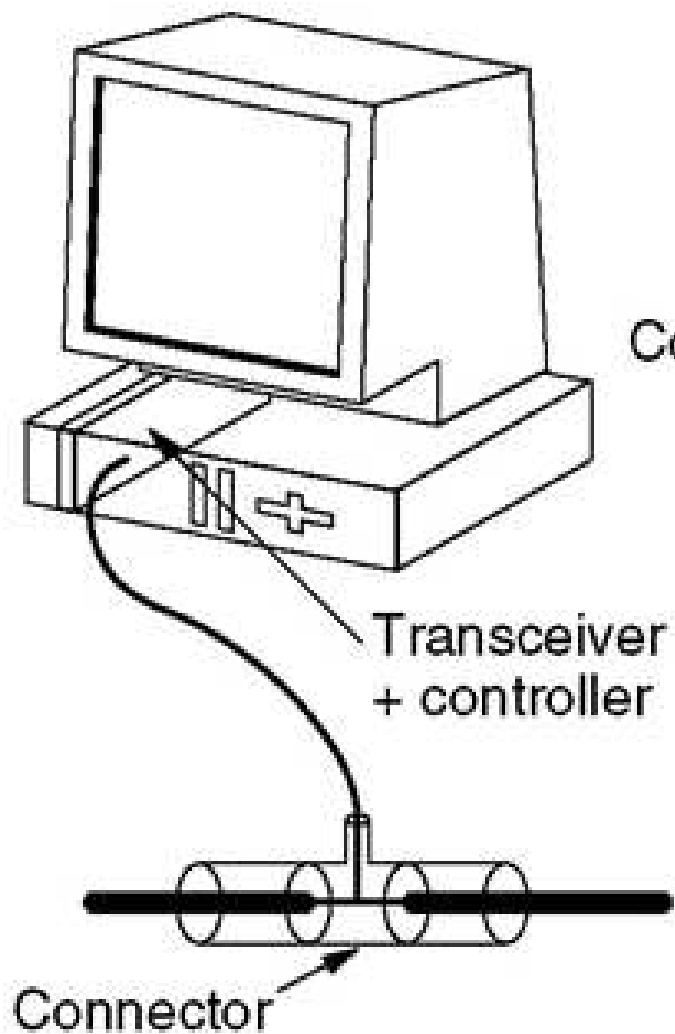
## b. Nối cáp

Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings

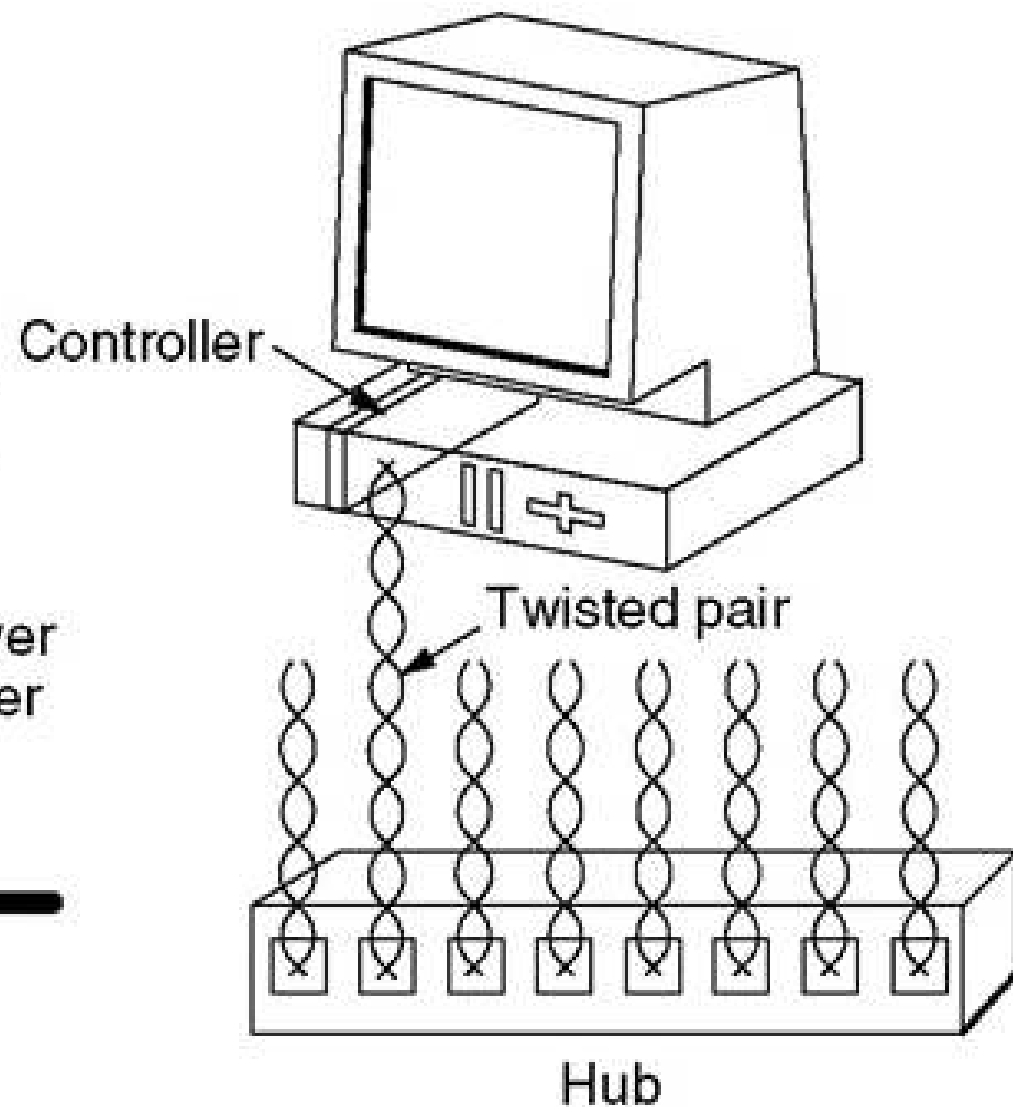
Các loại dây cáp thông dụng của Ethernet  
(Cáp đồng trục dày, cáp đồng trục mỏng,  
đôi dây xoắn, cáp quang)



# Một số dạng nối cáp

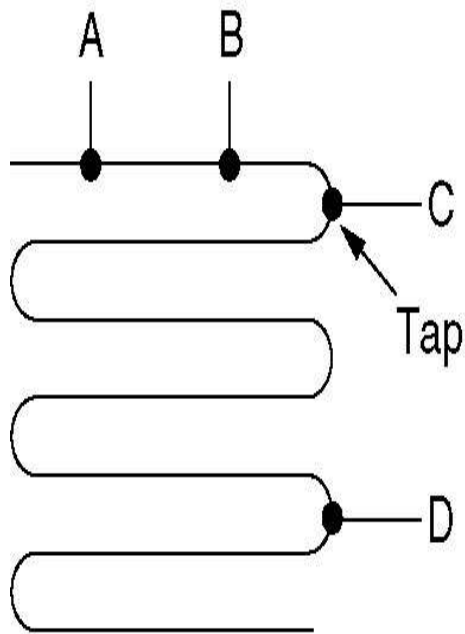


a. 10BASE2



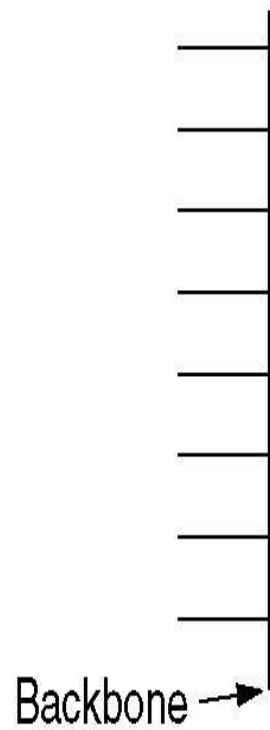
b. 10BASE-T

# Các dạng hình học của cáp



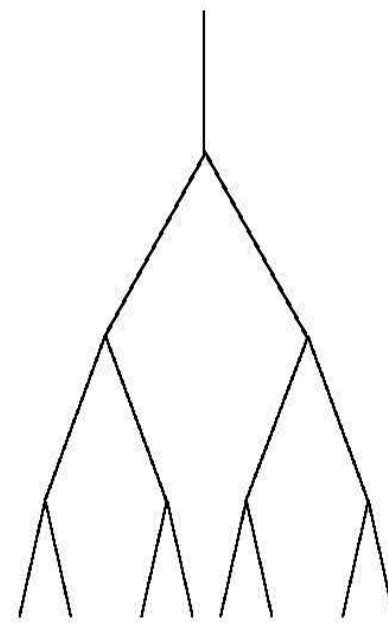
(a)

a. Tuyến tính



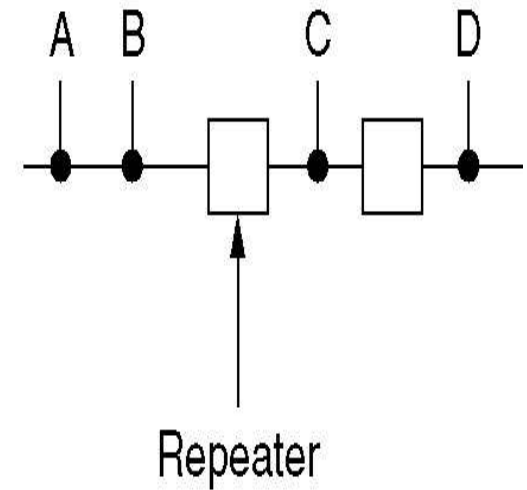
(b)

b. Đường trục



(c)

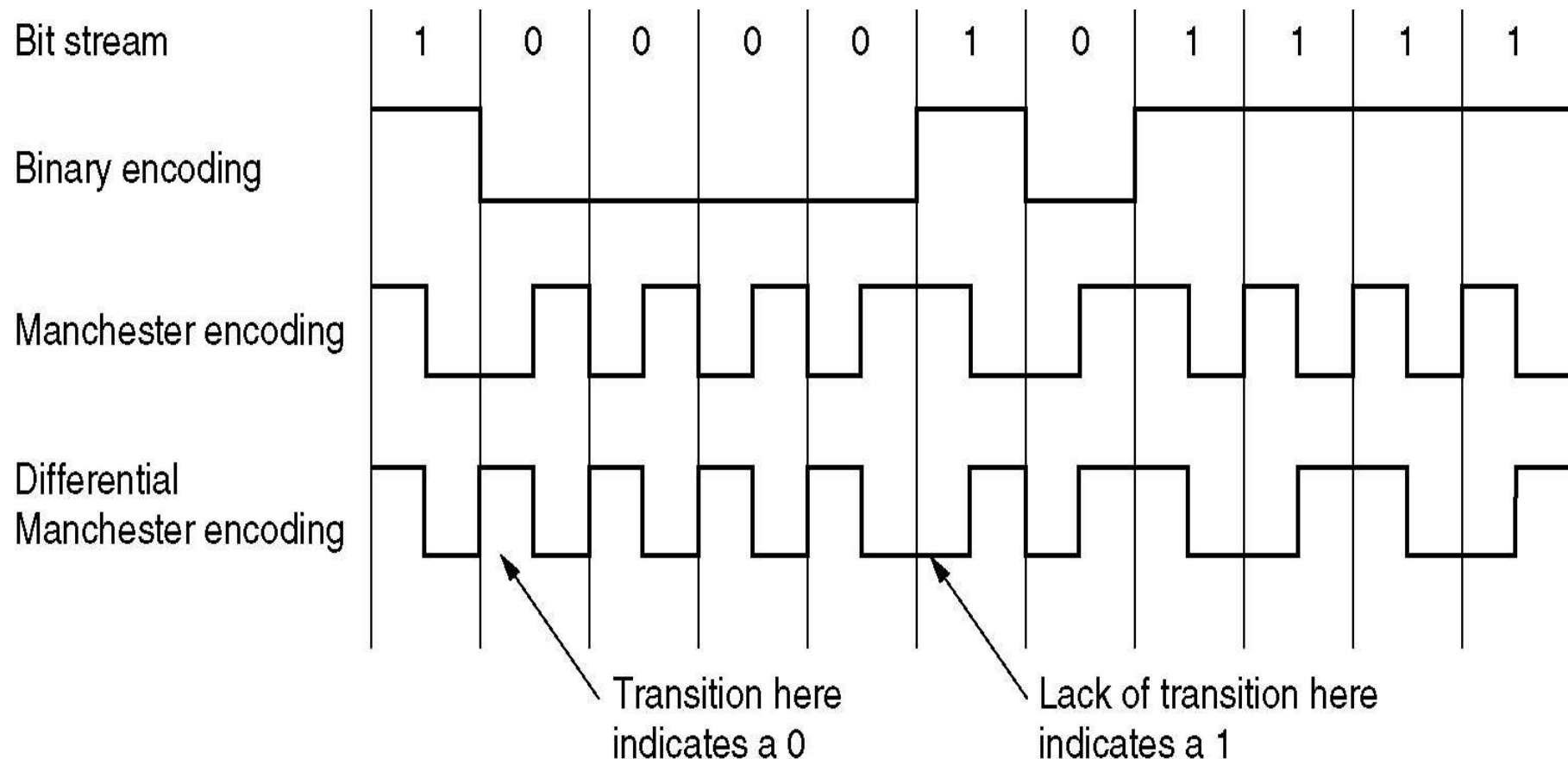
c. Cây



(d)

d. Phân đoạn

## c. Mã hoá bit



Mã hoá nhị phân (binary encoding)

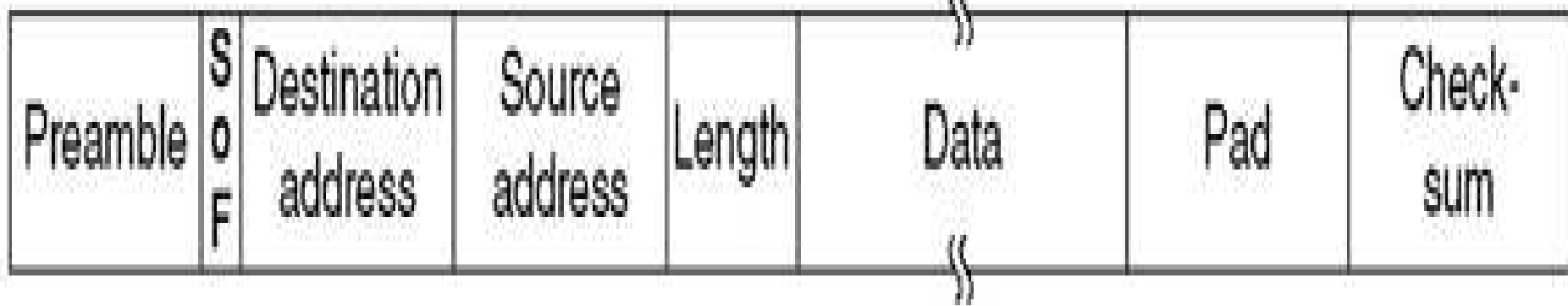
Mã hoá Manchester (Manchester encoding)

Mã hoá Manchester vi phân (Differential Manchester encoding)

## d. Giao thức lớp MAC

- CSMA/CD
- Cấu trúc frame theo IEEE 802.3

Bytes            8            6            6            2            0-1500            0-46            4



# Các trường trong Ethernet frame

- Preamble – Mở đầu: 7 bytes 10101010  
SOF – Start Of Frame: 10101011  
đánh dấu bắt đầu frame
- Destination address – Địa chỉ MAC máy nhận (6 bytes địa chỉ card mạng)
- Source address – Địa chỉ MAC máy gửi (6 bytes địa chỉ card mạng)
- Length/Type: kích thước/loại frame

## Các trường trong Ethernet frame (tt)

- Data: dữ liệu
- Pad: cần thêm vào để  $\text{frame} \geq 64$  bytes, từ yêu cầu phần cứng phát hiện xung đột
- Checksum: dùng trong phát hiện lỗi

## e. Giải quyết xung đột

- Theo giao thức CSMA/CD
- Thời gian chờ ngẫu nhiên theo giải thuật dạng hàm mũ nhị phân (binary exponent backoff)

đơn vị tính là  $\text{slotTime} = 512 \text{ bit times}$

mạng 10 Mbps,  $1 \text{ bit time} = 100 \text{ nanosec}$

## Giải quyết xung đột (tt)

- Nếu có xung đột, mỗi máy chờ ngẫu nhiên trong thời gian  $0 \rightarrow 1 \text{ slotTime}$
- Nếu có xung đột lần 2, mỗi máy chờ ngẫu nhiên trong thời gian  $0 \rightarrow 3 \text{ slotTime}$
- Nếu có xung đột lần  $i$ , mỗi máy chờ ngẫu nhiên trong thời gian  $0 \rightarrow 2^i - 1 \text{ slotTime}$
- Từ xung đột lần 10, mỗi máy chờ ngẫu nhiên trong thời gian  $0 \rightarrow 1023 \text{ slotTime}$
- Nếu xung đột đến lần 16 thì báo lỗi



## 2. Mạng Fast Ethernet

- Còn gọi là chuẩn IEEE 802.3u
- Giữ nguyên cấu trúc frame mạng Ethernet, giao thức CSMA/CD, tăng tốc độ 100 Mbps.  
1 bit time = 10 nanosec
- Không dùng cáp đồng trục

# Một số loại cáp mạng Fast Ethernet

Name	Cable	Max. segment	Advantages
100Base-T4	Twisted pair	100 m	Uses category 3 UTP
100Base-TX	Twisted pair	100 m	Full duplex at 100 Mbps
100Base-FX	Fiber optics	2000 m	Full duplex at 100 Mbps; long runs

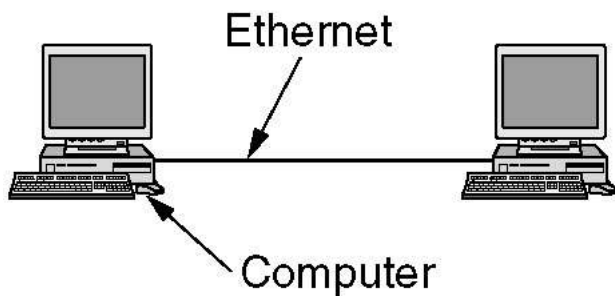
### 3. Mạng Gigabit Ethernet

- Còn gọi là chuẩn IEEE 802.3z
- Mở rộng mạng dạng Ethernet lên tốc độ 1000 Mbps
- Giữ cấu trúc frame, giao thức CSMA/CD

# Một số loại cáp mạng Gigabit Ethernet

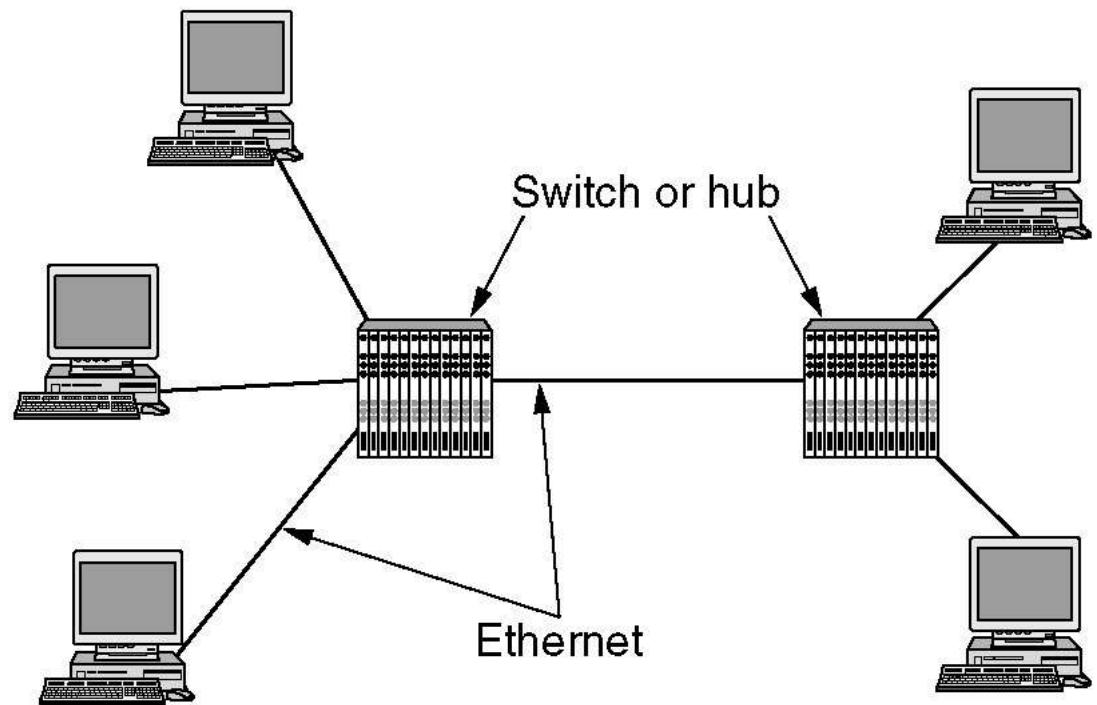
Name	Cable	Max. segment	Advantages
1000Base-SX	Fiber optics	550 m	Multimode fiber (50, 62.5 microns)
1000Base-LX	Fiber optics	5000 m	Single (10 $\mu$ ) or multimode (50, 62.5 $\mu$ )
1000Base-CX	2 Pairs of STP	25 m	Shielded twisted pair
1000Base-T	4 Pairs of UTP	100 m	Standard category 5 UTP

# Hai dạng kết nối mạng Gigabit Ethernet



(a)

a. Hai trạm



(b)

b. Nhiều trạm

# Nhận xét về các loại mạng Ethernet

## ■ Đơn giản

- Giá thành rẻ
- Tin cậy
- Dễ bảo trì

## ■ Hoạt động tốt với bộ giao thức TCP/IP

## ■ Tiếp tục phát triển



## 4. Mạng Token Ring – 802.5

- a. Giới thiệu mạng Token Ring
- b. Kết nối
- c. Sơ lược hoạt động

## a. Giới thiệu mạng Token Ring

- Xuất phát từ mạng Token Ring của IBM  
→ SNA (System Networks Architecture)
- Bao gồm các dạng máy tính IBM: PC, Midrange, Mainframe
- Tiêu chuẩn IEEE 802.5

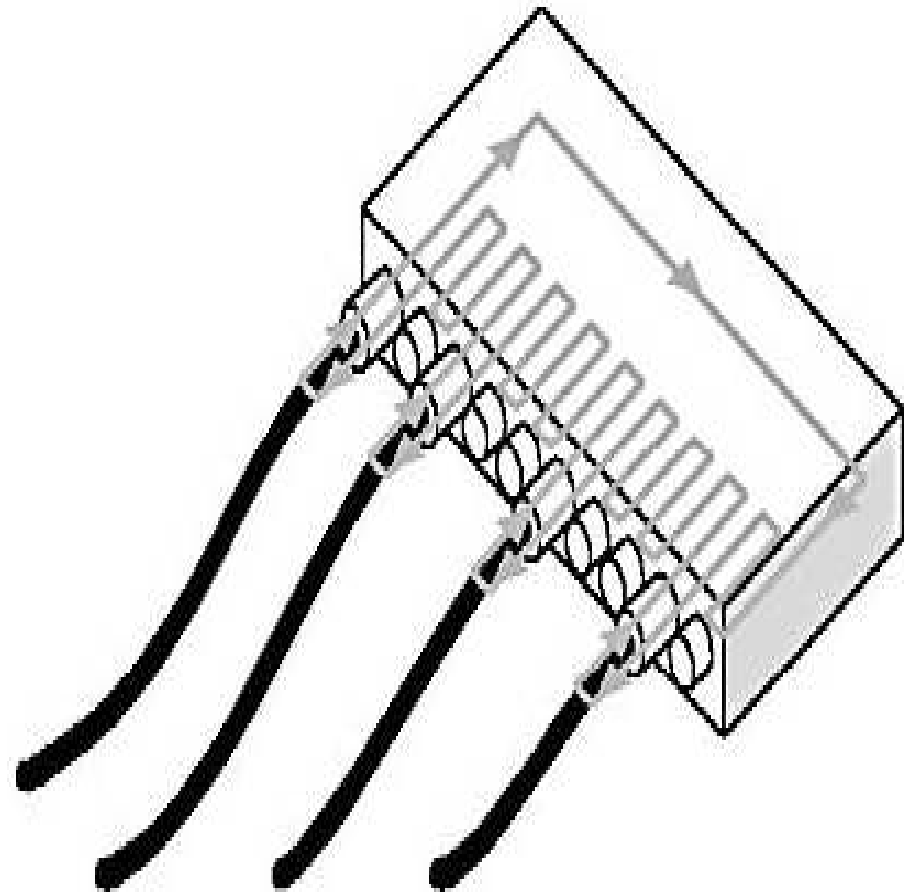
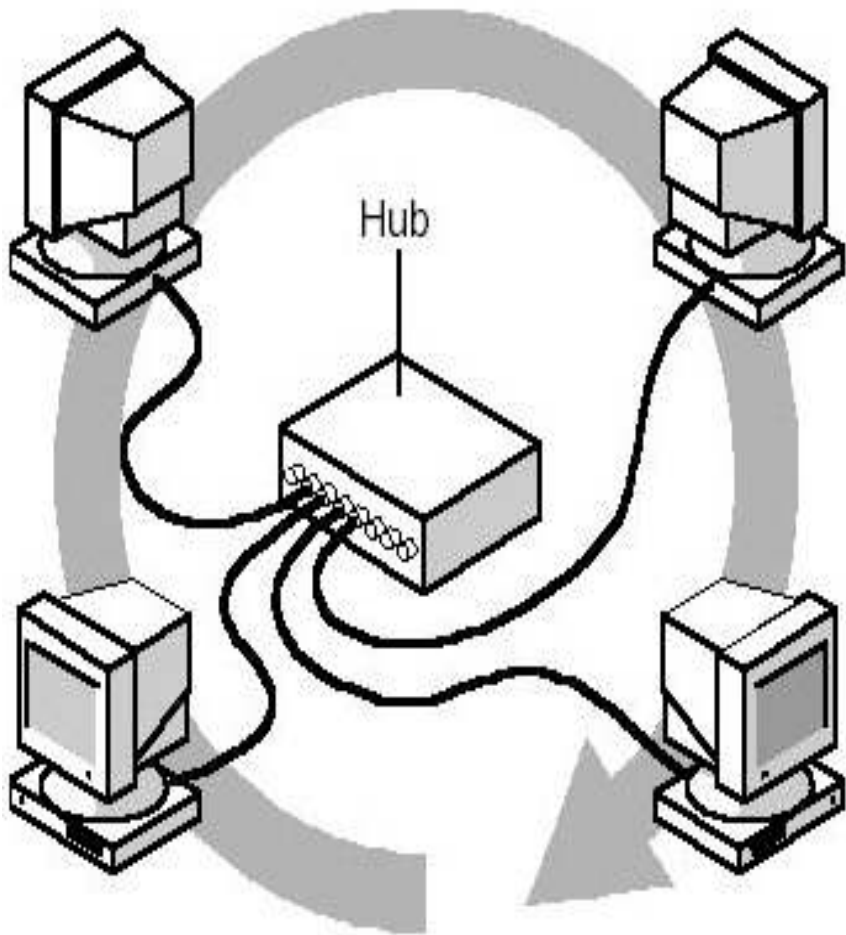
Tốc độ	4/16 Mbps	802.5
	100 Mbps	802.5t
	1000 Mbps	802.5v



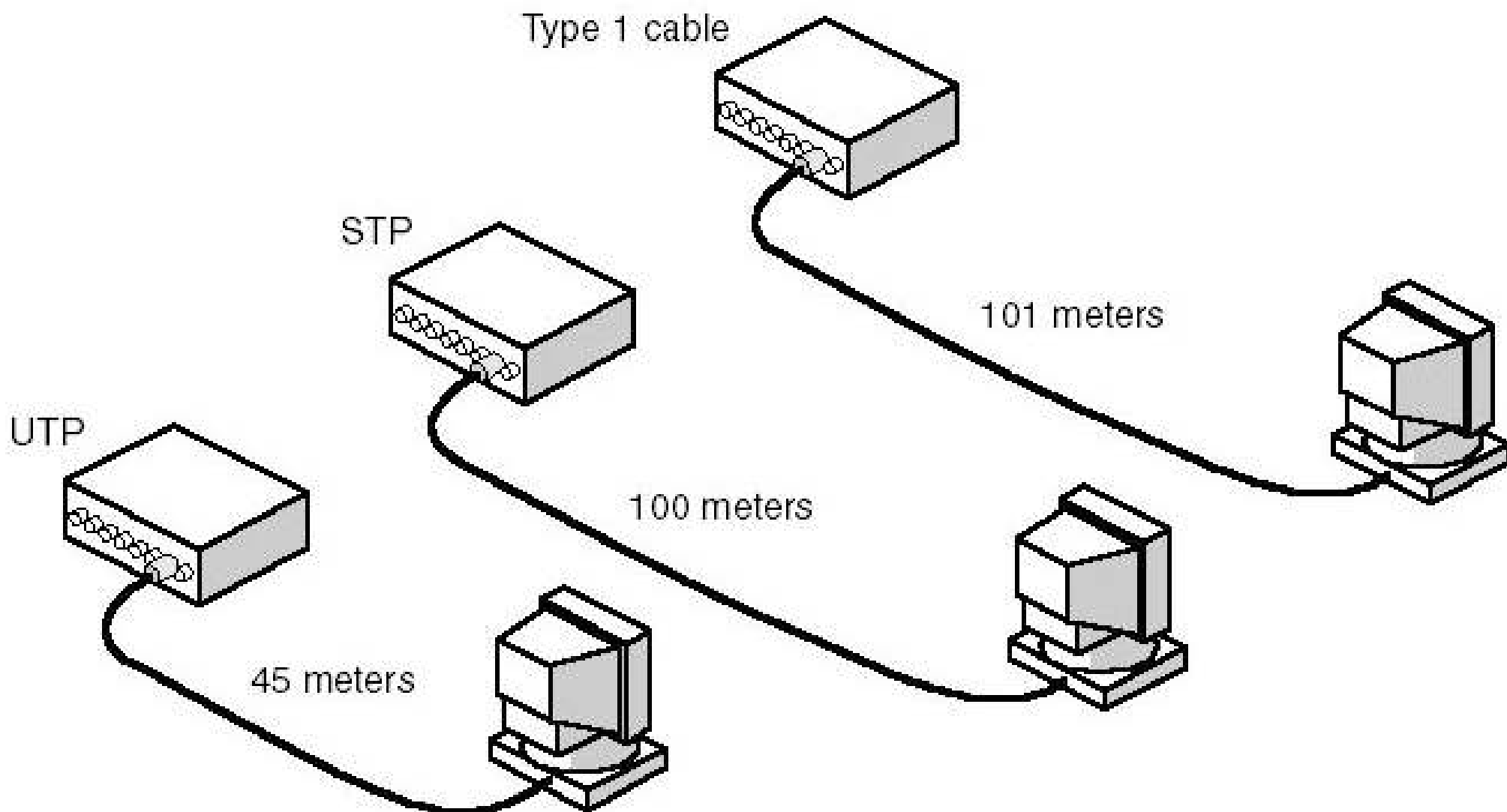
## b. Kết nối mạng Token Ring

- Dùng Hub, còn gọi là Wire center, MAU (Multistation Access Unit) tạo vòng vật lý
- Token Ring NIC
- UTP, STP với RJ-45

# Vòng vật lý dùng Hub



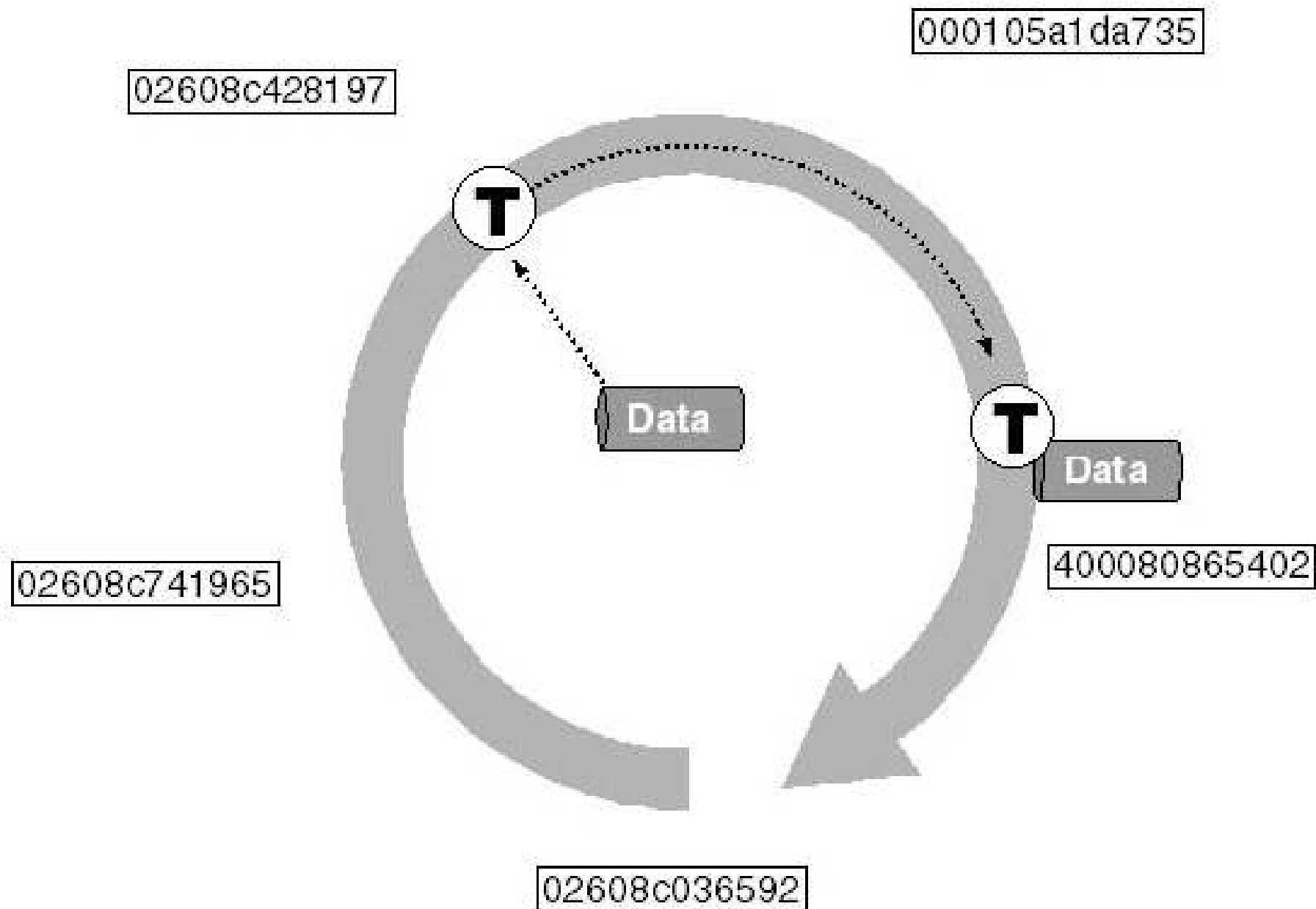
# Một số dạng cáp



## c. Sơ lược hoạt động mạng Token Ring

- Có 1 frame đặc biệt (token) truyền trên vòng
- Một máy cần gửi frame:
  - Chờ token
  - Truyền data frame
  - Data frame theo vòng đến máy nhận
  - Máy nhận xác nhận trên frame
  - Data frame theo vòng trở về máy gửi
  - Máy gửi hủy frame, gửi lại token

# Token xoay tròn





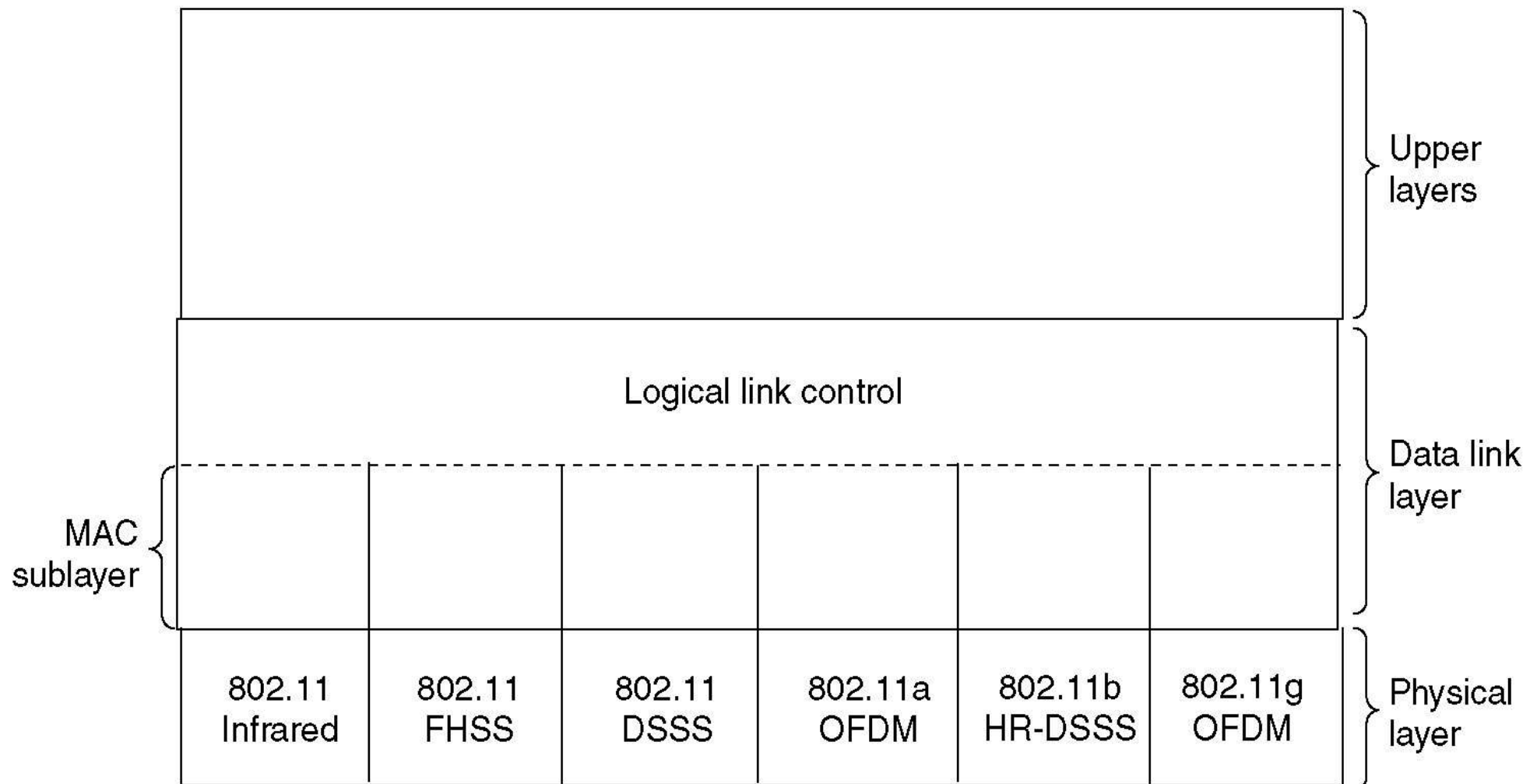
## 5. Mạng Wireless Ethernet – 802.11

- a. Giới thiệu tiêu chuẩn IEEE 802.11
- b. Kết nối mạng 802.11
- c. Sơ lược hoạt động mạng 802.11

## a. Giới thiệu tiêu chuẩn IEEE 802.11

- Là tiêu chuẩn cho mạng cục bộ không dây (Wireless LAN)
- Dùng sóng điện từ với nhiều kỹ thuật cho lớp vật lý
- Các dạng tốc độ
  - 1 → 2 Mbps : 802.11
  - 1 → 11 Mbps : 802.11b (Wi-Fi)
  - 5 Ghz band (~ 54 Mbps): 802.11a
  - 802.11g : tương đương 802.11a

# Một phần các giao thức theo chuẩn 802.11

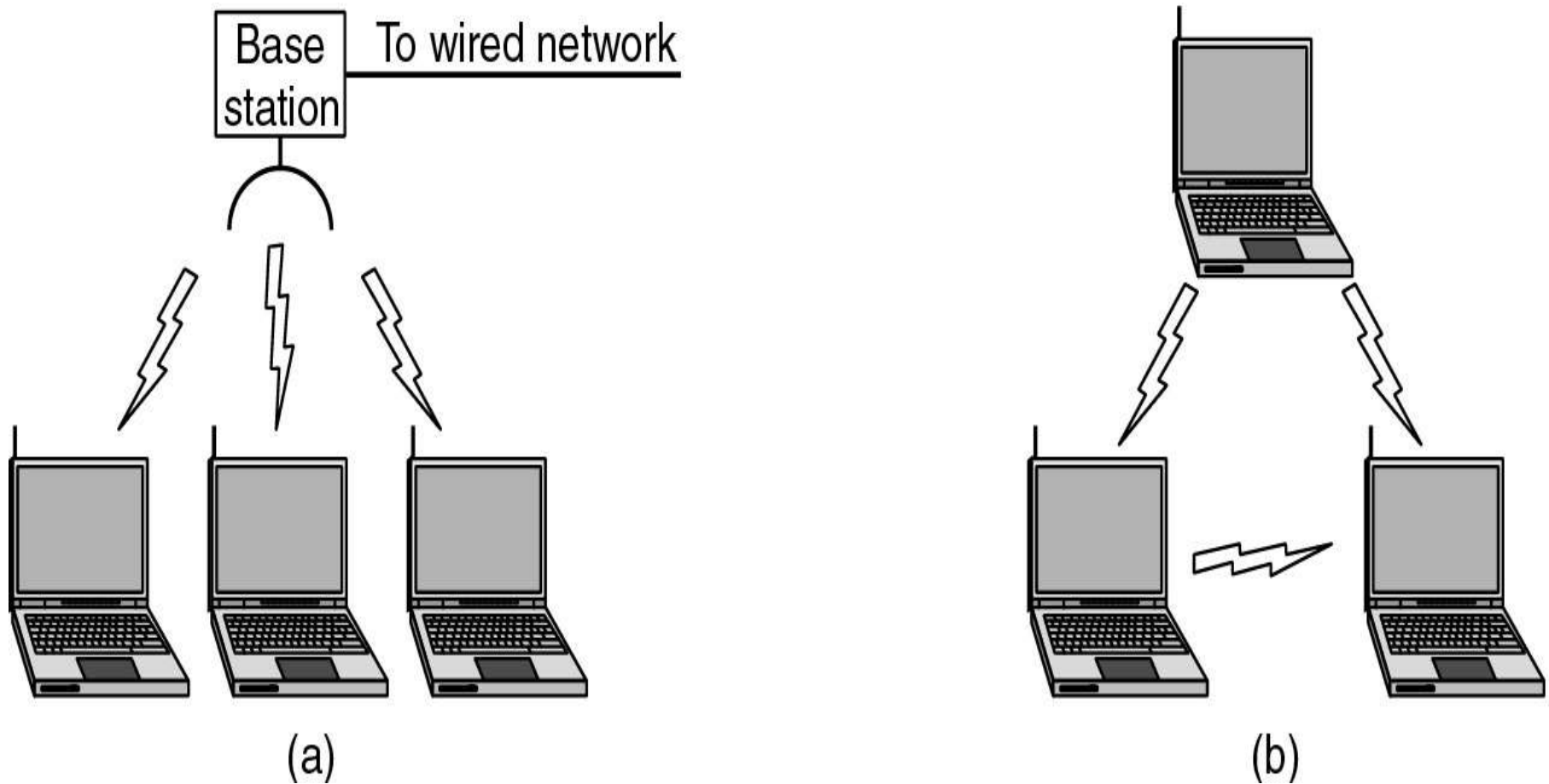




## b. Kết nối mạng 802.11

- Card mạng không dây (Wireless NIC)
- Kết nối:
  - Có trạm nền (base station/access point)
  - Ngang hàng (peer nodes / ad hoc)

# Hai dạng kết nối mạng không dây



- a. Có dùng base station, còn gọi là access point
- b. Các máy gửi nhận trực tiếp, ad hoc networking

## c. Sơ lược hoạt động mạng 802.11

- Máy trạm phải liên kết (associate) để kết nối vào mạng (access point/peer)
- Máy trạm có thể tách (disassociate) khỏi trạm nền, hay thay đổi trạm nền khác (reassociate)
- Máy trạm có thể cần đăng nhập (authenticate) trước khi trao đổi dữ liệu

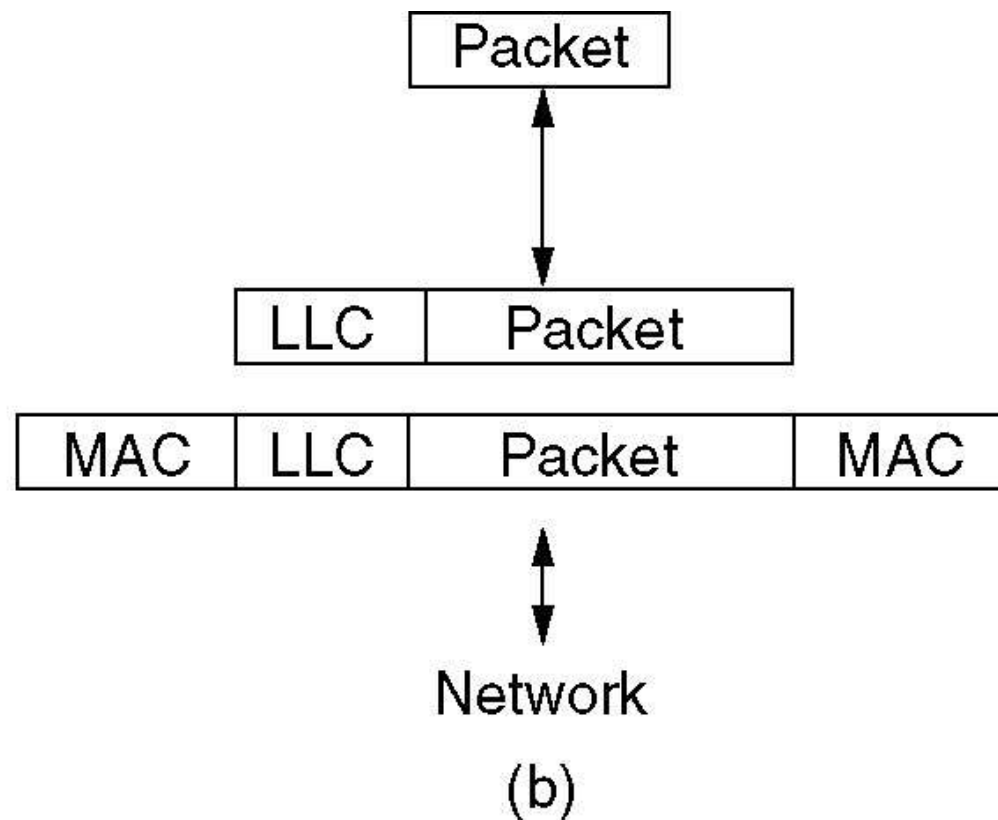
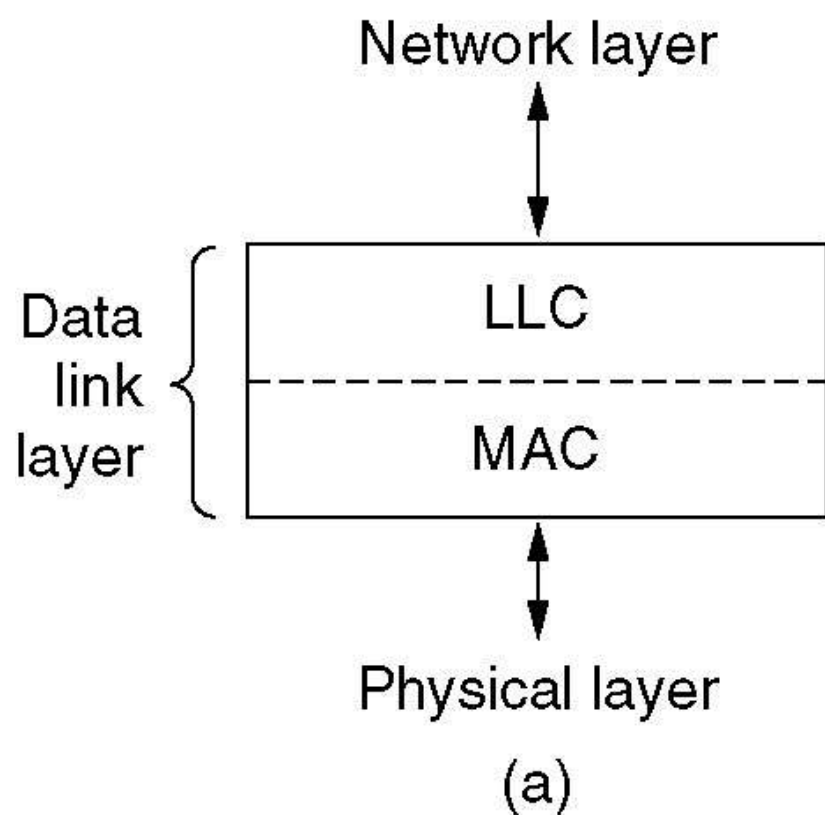
## Sơ lược hoạt động mạng 802.11 (tt)

- Sau khi thiết lập kết nối với mạng, mỗi máy có thể gửi frame theo tiêu chuẩn 802.11
- Dùng giao thức CSMA/CA  
(Carrier Sense Multiple Access/Collision Avoidance)
- Khi máy gửi truyền frame, máy nhận gửi ACK

## 6. Điều khiển liên kết luận lý (LLC)

- Chuẩn IEEE 802.2
- Giao thức LLC ở trên các giao MAC:
  - Che dấu những khác biệt, tạo khuôn dạng và giao diện chung đối với lớp mạng
  - Thực hiện kiểm soát lỗi, kiểm soát lưu lượng nếu cần thiết

# Quan hệ giữa các lớp



a. Vị trí lớp con LLC

b. Quan hệ về dữ liệu

# Các dịch vụ của lớp LLC

## ■ Unacknowledged connectionless-mode

Gửi nhận không kiểm soát

- Có các dạng point-to-point, multicast, broadcast

## ■ Acknowledged connectionless-mode

Gửi nhận có xác nhận của máy nhận

- dạng point-to-point

## ■ Connection-mode

Gửi nhận có thiết lập kết nối



## V. Giới thiệu về Bridge, Switch

1. Bridge (cầu nối)
2. Switch (chuyển mạch)



# 1. Bridge

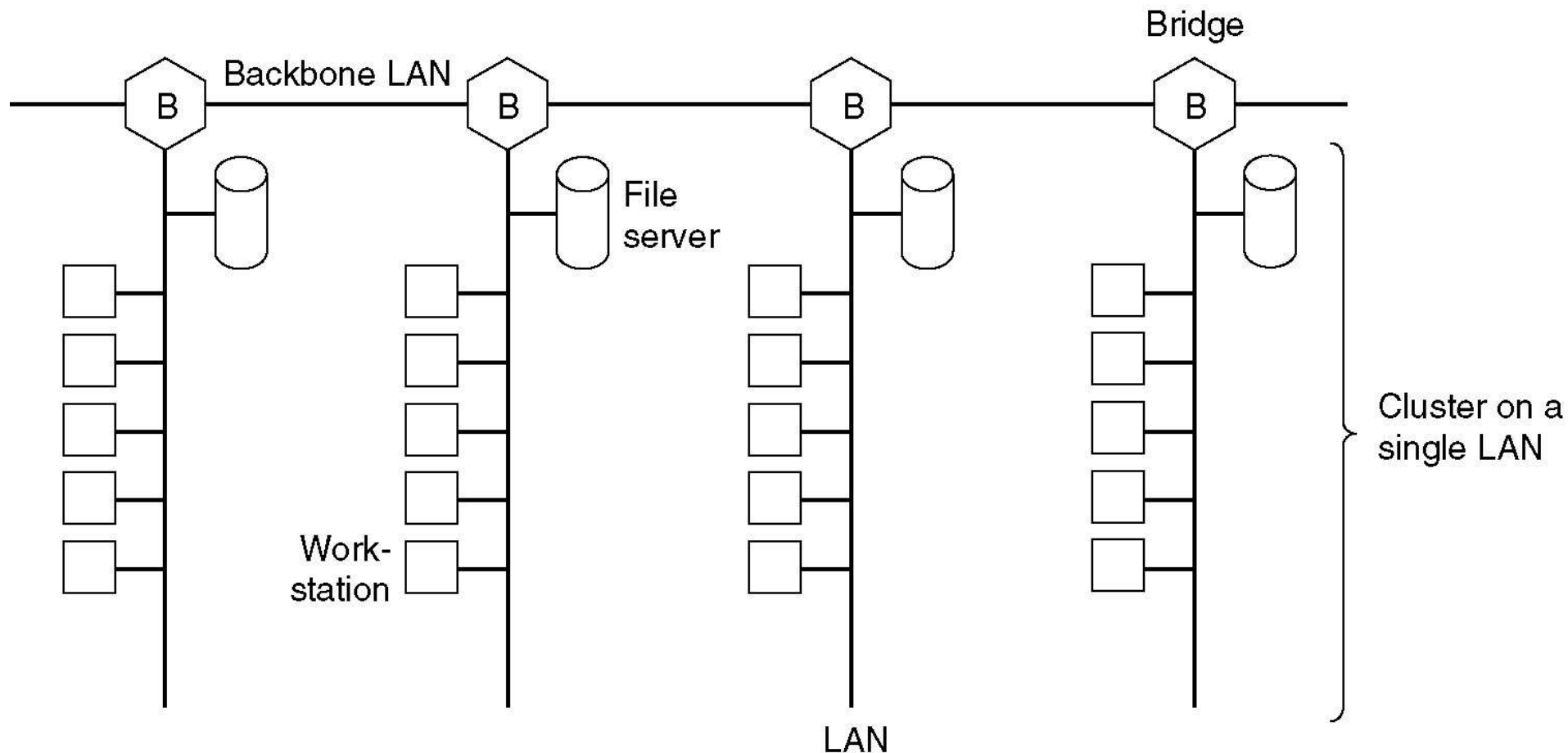
Mục đích:

- Kết nối các mạng LAN khác loại
- Mở rộng khoảng cách giữa các máy
- Chia mạng lớn thành các mạng nhỏ

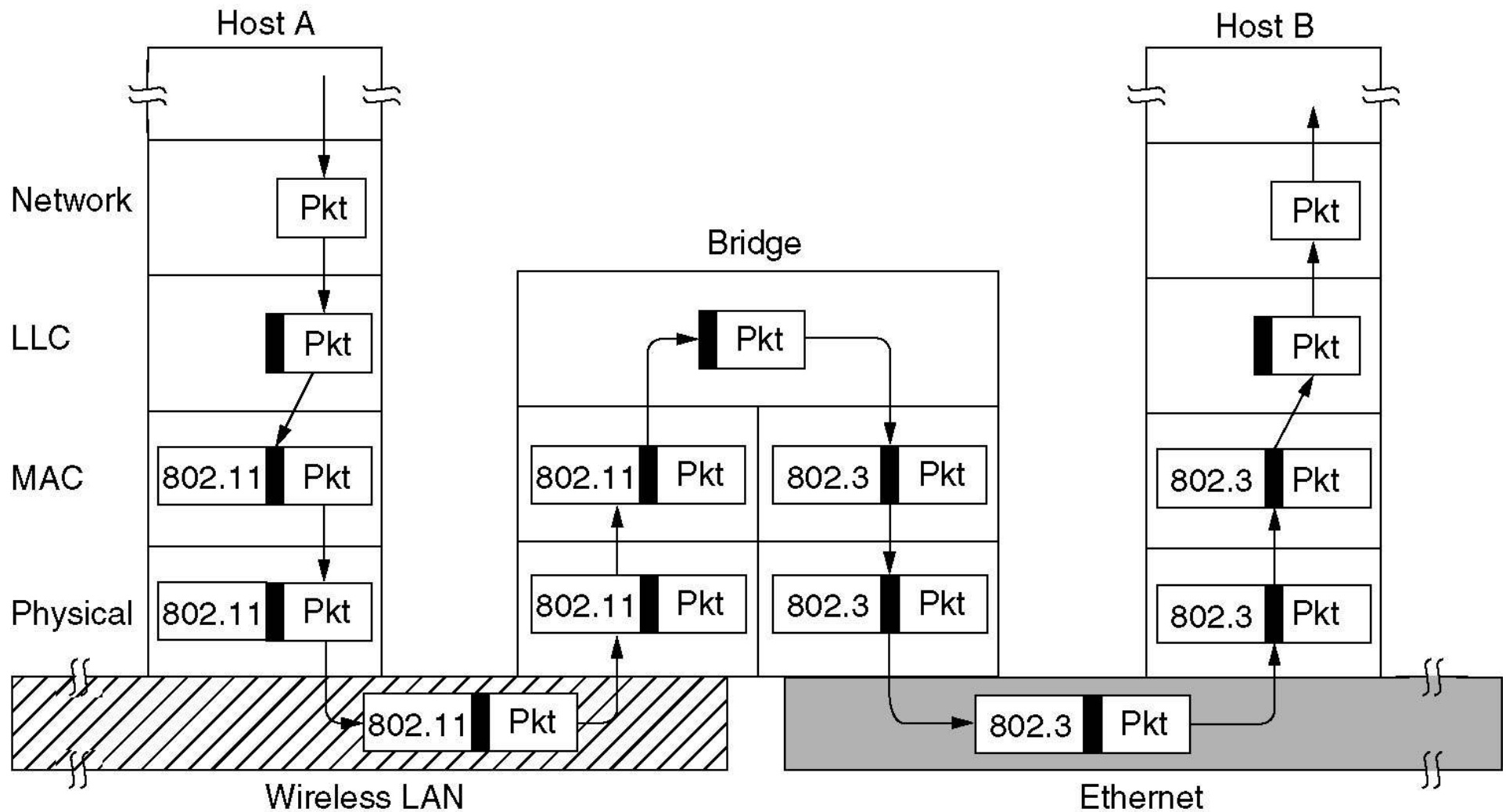
Hoạt động: *dạng store-and-forward*

- Nhận frame từ mạng nguồn
- Thực hiện các xử lý cần thiết
- Chuyển frame đến mạng đích

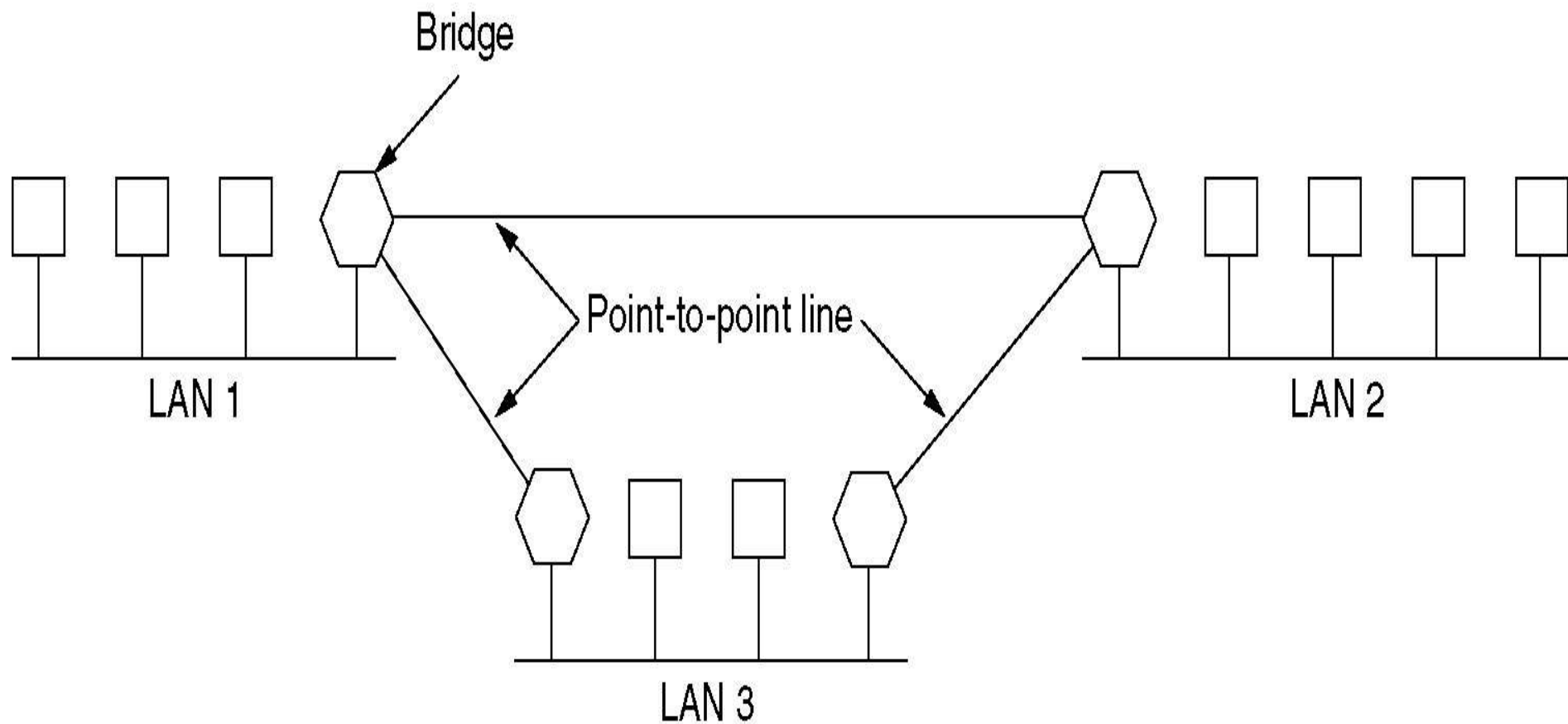
# Nhiều LAN kết dùng các bridge



# Hoạt động của bridge từ 802.11 sang 802.3



# Kết nối các LAN từ xa dùng bridge



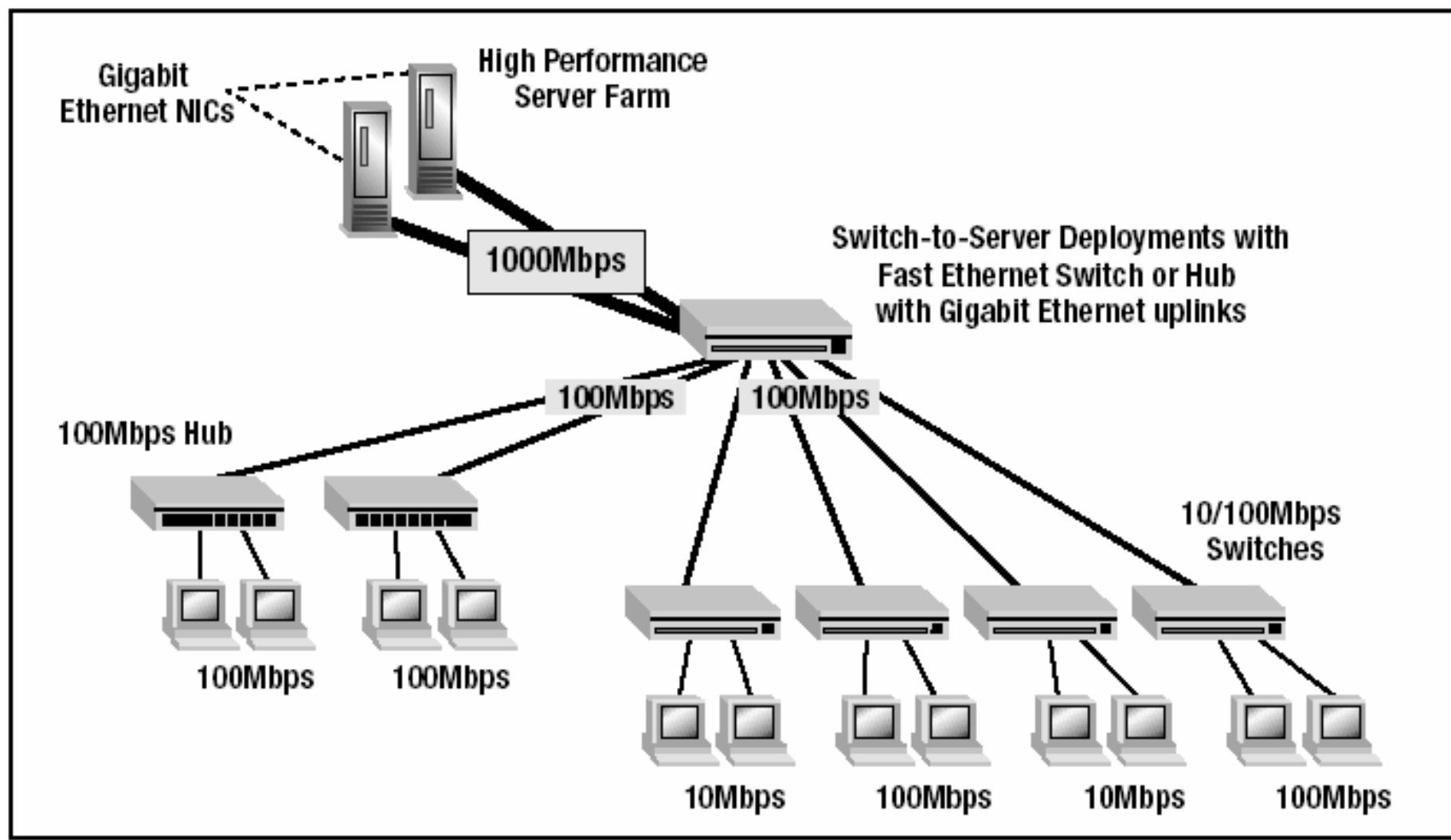
## 2. Switch

Switch: bridge nhiều port, tốc độ cao

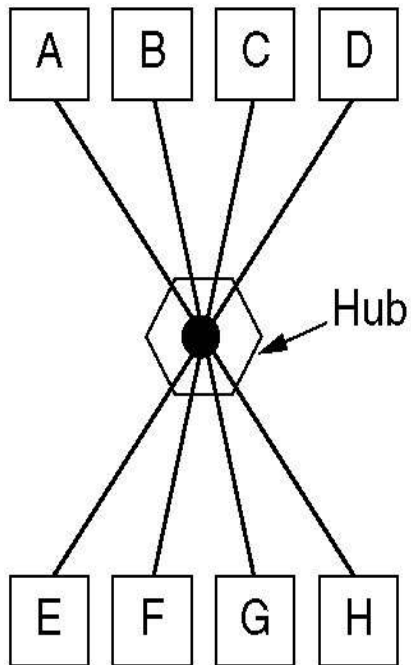
Đặc điểm:

- Tốc độ cao
- Giảm xung đột → chỉ xung đột giữa máy và switch port
- Hoạt động ở chế độ full-duplex  
→ không xung đột
- Có khả năng kiểm tra checksum của frame

# Ví dụ:

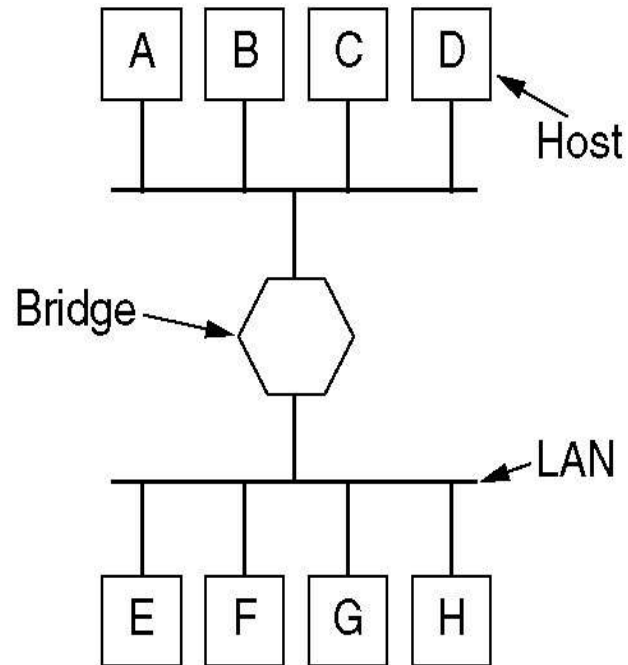


# So sánh Hub, Bridge, Switch



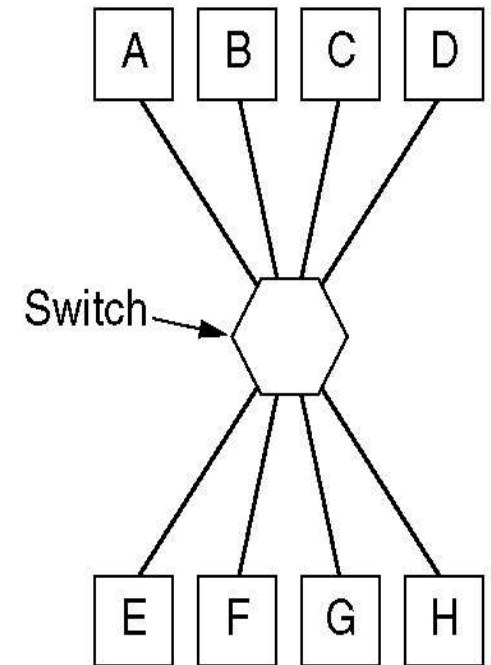
(a)

a. Hub



(b)

b. Bridge



(c)

c. Switch

# NHẬP MÔN MẠNG MÁY TÍNH

## Chương 4

### LỚP NETWORK (LỚP MẠNG)





# Nội dung chương 4

- I. Các vấn đề thiết kế lớp network
- II. Giới thiệu về định tuyến
- III. Các vấn đề liên mạng
- IV. Lớp network trên mạng TCP/IP
- V. Giới thiệu IPv6



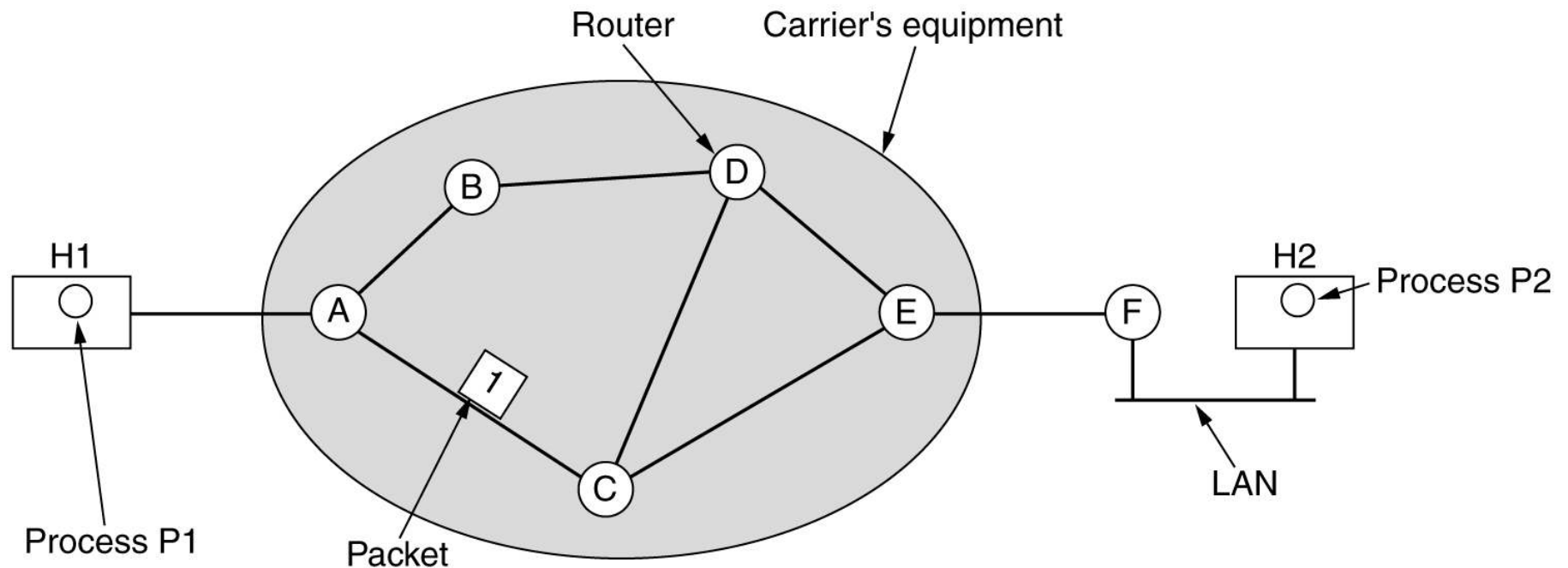
# I. Các vấn đề thiết kế lớp network

1. Nhiệm vụ lớp Network
2. Các dịch vụ cung cấp cho lớp transport

# 1. Nhiệm vụ lớp network

- Cung cấp dịch vụ gửi nhận dữ liệu (packet) giữa hai máy bất kỳ  
*hai máy bất kỳ có thể trên các mạng khác nhau*
- Giải quyết vấn đề định tuyến, liên mạng, định địa chỉ mạng

# Môi trường hoạt động lớp network



- Host gửi packet đến router gần nhất
- Các router truyền các packet theo dạng store-and-forward



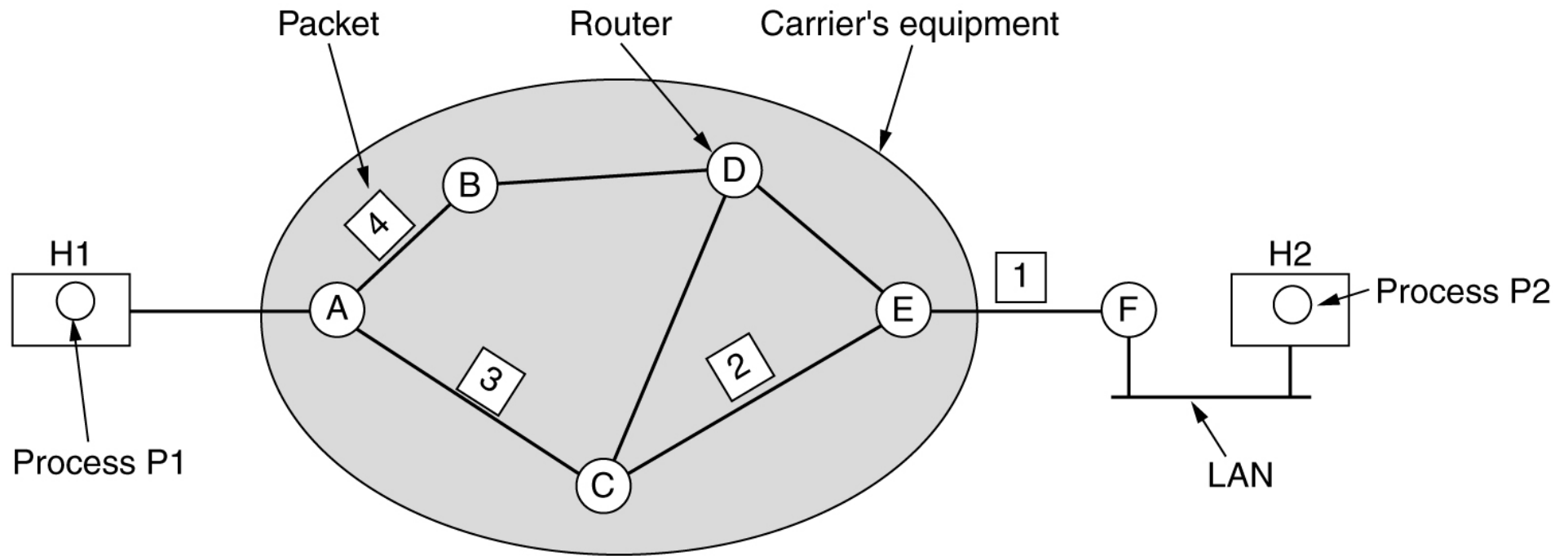
## 2. Các dịch vụ cung cấp cho lớp transport

- Dịch vụ không kết nối (connectionless)
- Dịch vụ có kết nối (connection-oriented)

# Các đặc điểm hai dạng dịch vụ

Vấn đề	Dịch vụ không kết nối	Dịch vụ có kết nối
Thiết lập kết nối	Không cần	Cần → mạch ảo
Định địa chỉ	Mỗi packet chứa địa chỉ nguồn và địa chỉ đích	Mỗi packet chứa thông tin về mạch ảo
Định tuyến	Mỗi packet được định tuyến độc lập	Tuyến được chọn khi thiết lập mạch ảo. Tất cả packet truyền trên tuyến.

# Ví dụ: định tuyến dạng không kết nối



A's table

initially	later
A   -	A   -
B   B	B   B
C   C	C   C
D   B	D   B
E   C	E   B
F   C	F   B

Dest. Line

C's table

A   A
B   A
C   -
D   D
E   E
F   E

E's table

A   C
B   D
C   C
D   D
E   -
F   F



## II. Giới thiệu về định tuyến

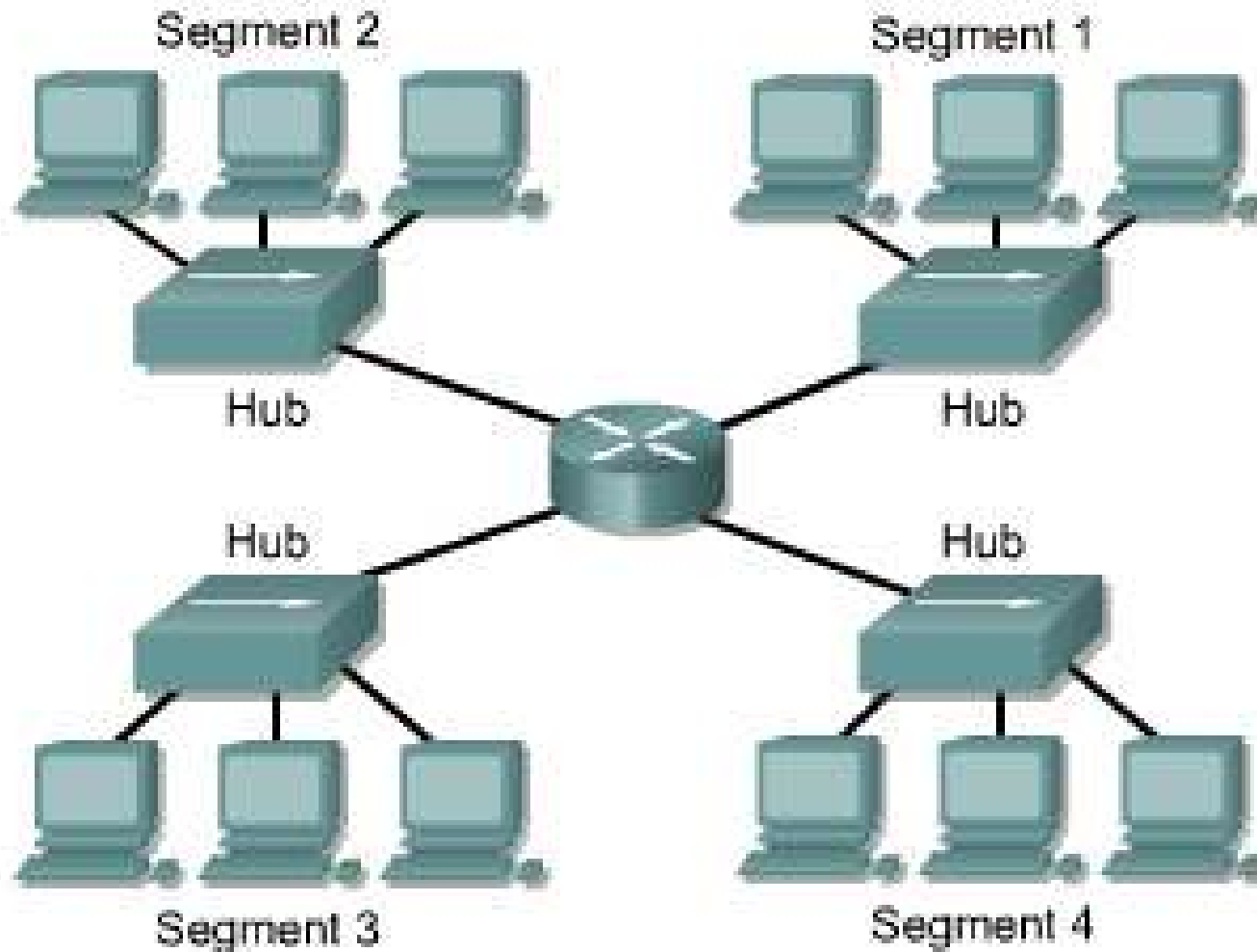
1. Khái niệm định tuyến
2. Định tuyến tĩnh
3. Định tuyến động



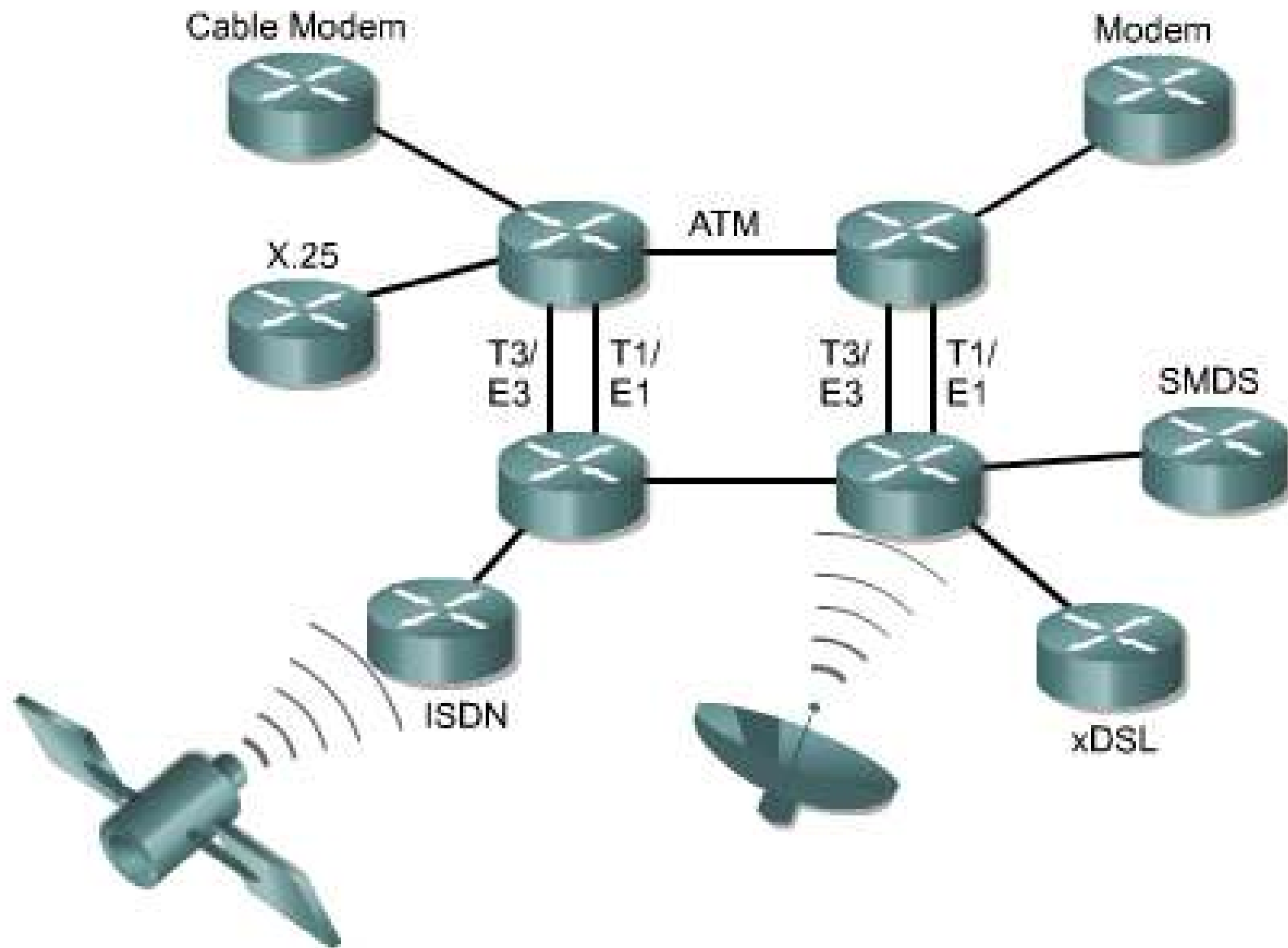
# 1. Khái niệm định tuyến

- Định tuyến (routing): xác định con đường (tuyến, route) chuyển tiếp dữ liệu từ mạng này sang mạng khác
- Định tuyến là chức năng của lớp network
- Định tuyến được thực hiện tại bộ định tuyến (router)
- Router là thiết bị (hay phần mềm trên một máy tính) kết nối giữa các mạng

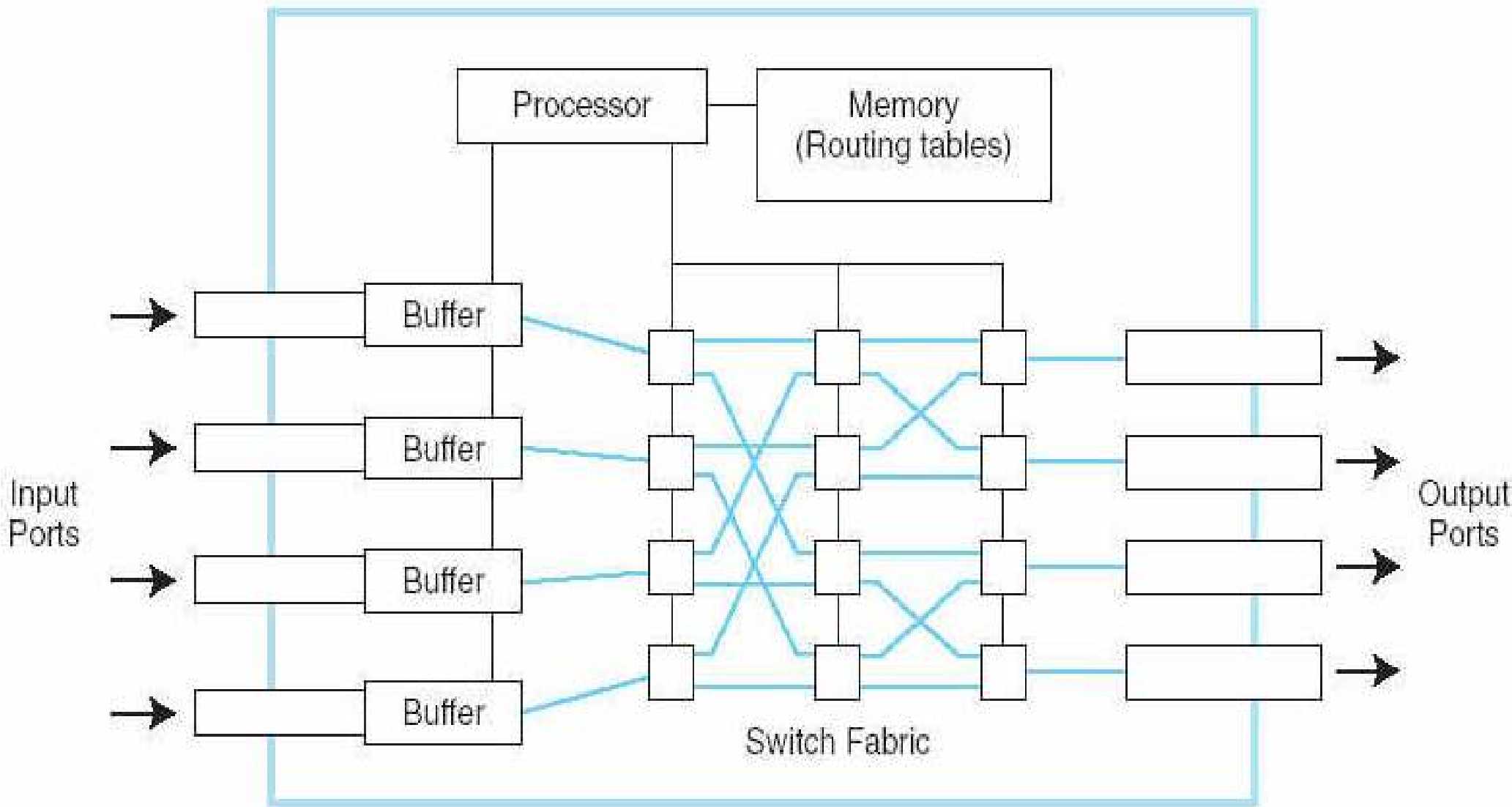
# Router kết nối các mạng cục bộ



# Router trên mạng miền rộng



# Cấu trúc cơ bản router



# Chức năng router

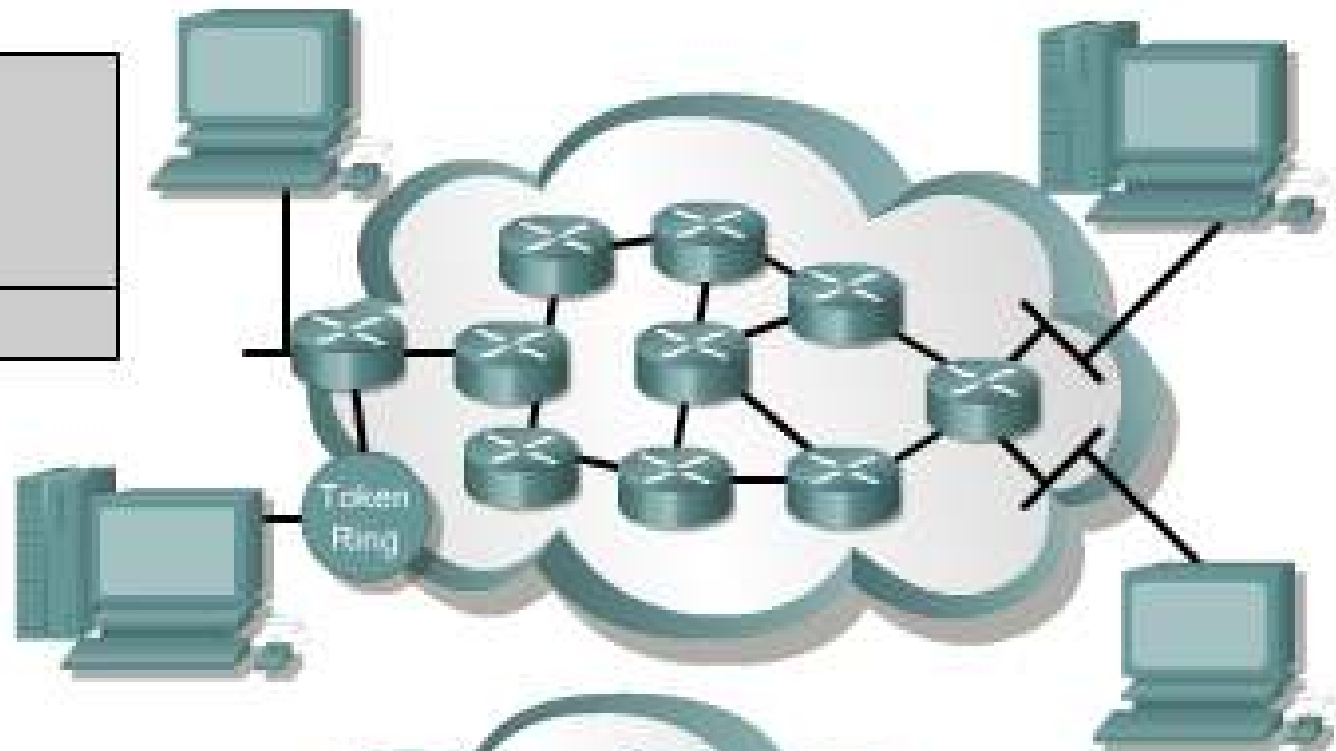
- Duy trì các bảng định tuyến (routing tables), được xây dựng theo các giao thức định tuyến (routing protocol)
- Khi nhận dữ liệu thì dùng bảng định tuyến để xác định ngõ ra

# Giao thức định tuyến (routed protocol)

## Giao thức định tuyến (routing protocol)

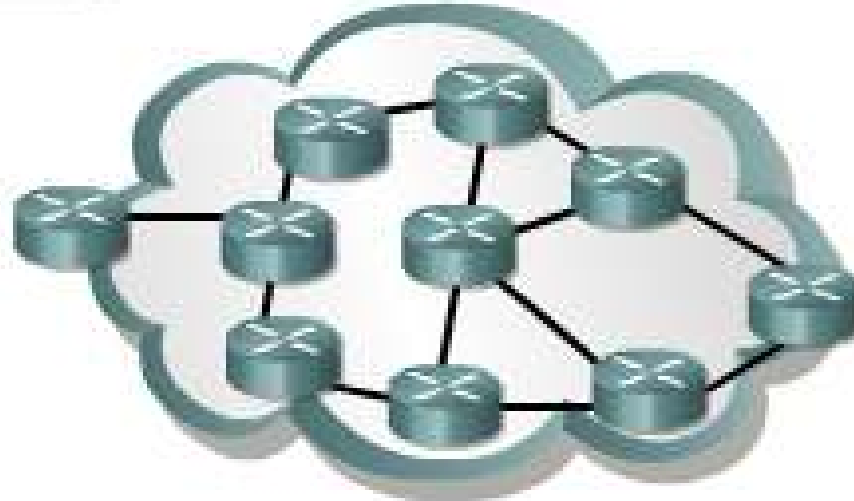
Routed protocol  
used between  
routers to direct  
user traffic

Examples: IP and IPX



Routing protocol  
used between  
routers to maintain  
tables

Examples: RIP, IGRP, OSPF



# Ví dụ: Node 1 cần gửi dữ liệu cho Node 2

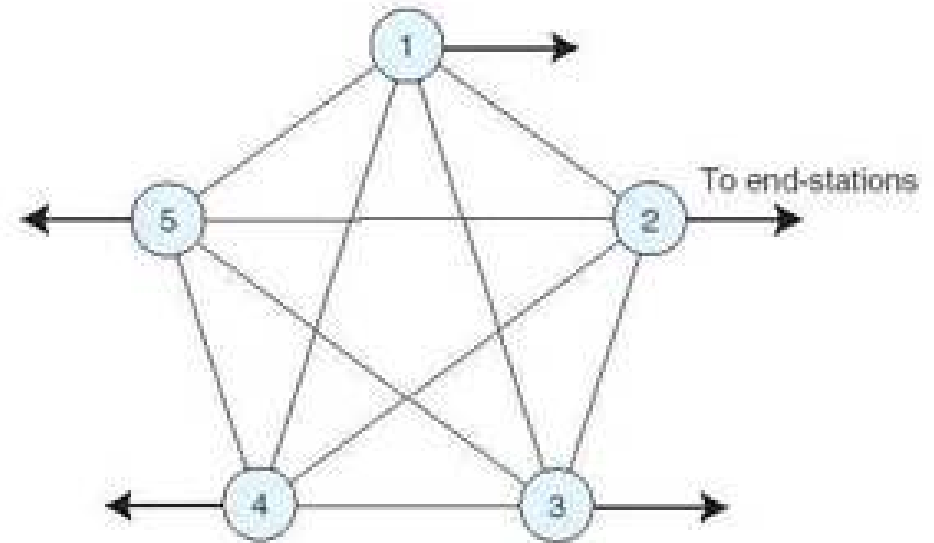
- 1 route với 1 hop

$1 \rightarrow 2$

- 3 routes với 2 hops

$1 \rightarrow 3 \rightarrow 2$        $1 \rightarrow 4 \rightarrow 2$

$1 \rightarrow 5 \rightarrow 2$



- 6 routes với 3 hops:

$1 \rightarrow 3 \rightarrow 4 \rightarrow 2$        $1 \rightarrow 3 \rightarrow 5 \rightarrow 2$        $1 \rightarrow 5 \rightarrow 4 \rightarrow 2$

$1 \rightarrow 4 \rightarrow 3 \rightarrow 2$        $1 \rightarrow 5 \rightarrow 3 \rightarrow 2$        $1 \rightarrow 4 \rightarrow 5 \rightarrow 2$

- 6 routes với hops:

$1 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 2$        $1 \rightarrow 4 \rightarrow 3 \rightarrow 5 \rightarrow 2$

$1 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2$        $1 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 2$

$1 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 2$        $1 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 2$

# Các dạng định tuyến

## ■ Định tuyến tĩnh

- Tuyến do người quản trị mạng thiết lập

## ■ Định tuyến động

- Tuyến do các router thiết lập động theo các giao thức định tuyến



## 2. Định tuyến tĩnh

Gồm 3 giai đoạn:

- Người quản trị thiết lập các tuyến
- Tuyến được cài đặt trên router dưới dạng bảng định tuyến
- Các packet được định tuyến theo các tuyến cố định

## Định tuyến tĩnh (tt)

- Khi mạng thay đổi, phải xác định lại các tuyến
- Chỉ dùng cho mạng cố định, quy mô nhỏ
- Ví dụ giải thuật định tuyến tĩnh:  
Giải thuật đường dẫn ngắn nhất  
(Shortest Path Routing)  
theo Dijkstra, Moore, ...

### 3. Định tuyến động

- Tuyến được thiết lập tự động đáp ứng sự thay đổi của mạng
- Tuyến có dạng tối ưu
- Giao thức định tuyến là cố định, dữ liệu (bảng định tuyến) thay đổi thông qua việc trao đổi giữa các router

# Giải thuật định tuyến

Gồm 2 dạng:

- Distance Vector Routing

Định tuyến vector khoảng cách

- Link State Routing

Định tuyến trạng thái liên kết

# Định tuyến vector khoảng cách

- Còn gọi là giải thuật Bellman-Ford
- Nguyên tắc:
  - Mỗi router lưu bảng định tuyến cung cấp:
    - Khoảng cách tốt nhất đến đích
    - Đường đi đến đích
  - Các router định kỳ trao đổi bảng định tuyến với các router láng giềng, cập nhật bảng định tuyến

## Định tuyến vector khoảng cách (tt)

- Khoảng cách: số router trên tuyến
  - Hop count
- Ưu điểm
  - Đơn giản
- Khuyết điểm
  - Thời gian xây dựng bảng định tuyến lớn khi mạng quy mô lớn
  - Dữ liệu trao đổi trên mạng lớn
  - Các tuyến không còn sử dụng có thể tồn tại trên bảng định tuyến

# Ví dụ định tuyến vector khoảng cách

- Dùng trên mạng ARPANET/Internet đến 1979 dưới tên RIP  
(Routing Information Protocol)
- Đặc điểm RIP
  - Dạng định tuyến vector khoảng cách
  - Khoảng cách: số hop
  - Packet bị hủy khi hop  $> 15$
  - Định kỳ cập nhật bảng định tuyến: 30 giây

# Định tuyến trạng thái liên kết

Công việc của router:

- Tìm các router láng giềng và học địa chỉ mạng của các router láng giềng
- Xác định thời gian trì hoãn, chi phí truyền dữ liệu đến từng láng giềng
- Xây dựng 1 gói cho biết các thông tin trên (link state packet)
- Truyền gói này đến các router khác
- Tính đường dẫn ngắn nhất đến mỗi router khác



# Định tuyến trạng thái liên kết(tt)

## ■ Đặc điểm so với định tuyến vector khoảng cách:

- Đáp ứng nhanh với sự thay đổi của mạng
- Duy trì cơ sở dữ liệu phức tạp về hình học của toàn mạng
- Router cần nhiều bộ nhớ hơn, xử lý nhiều hơn
- Cập nhật thông tin khi có biến cố trên mạng  
→ sử dụng ít băng thông hơn

## Ví dụ định tuyến trạng thái liên kết

### Giao thức OSPF (Open Shortest Path First)

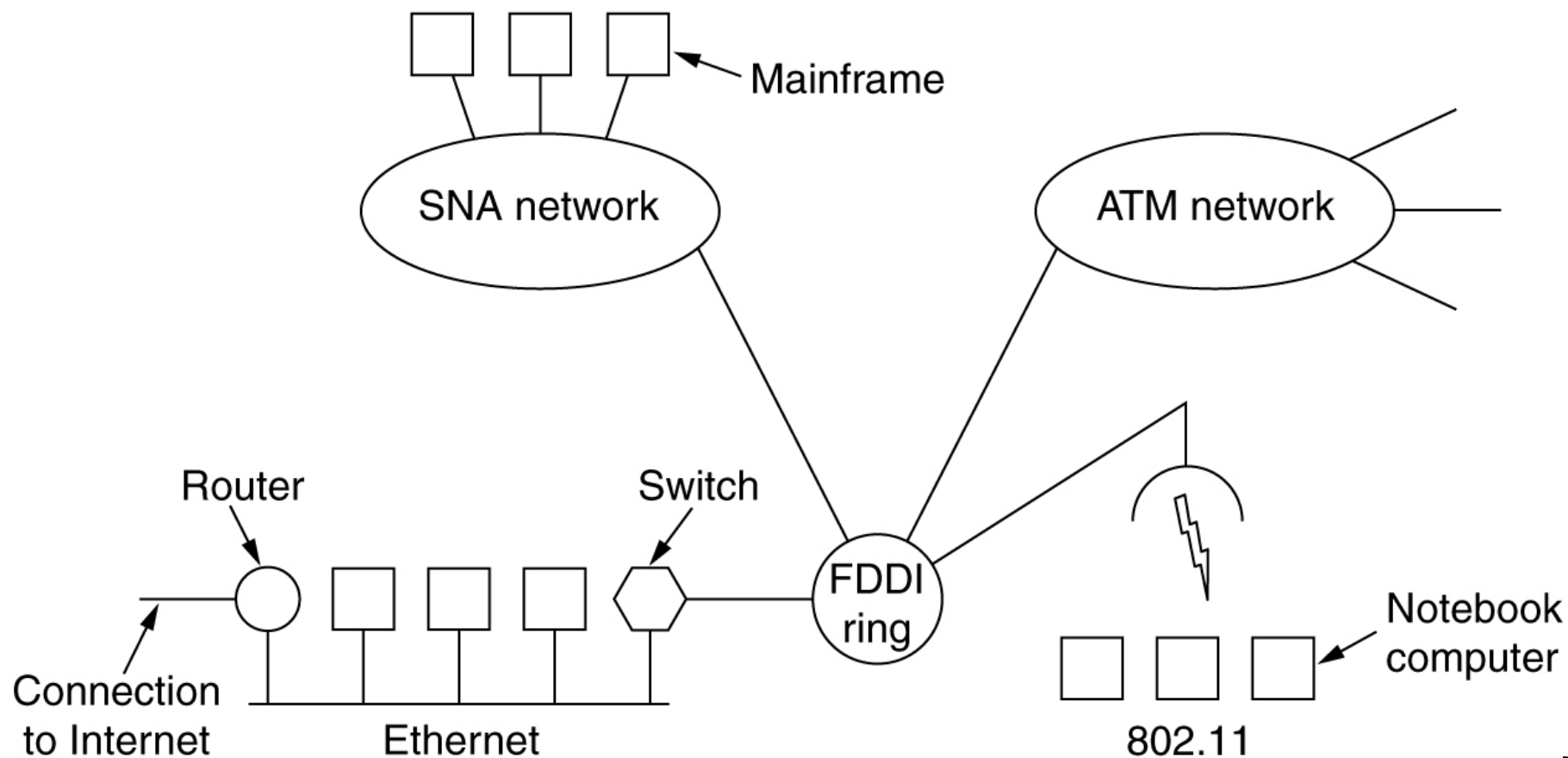
- Dạng định tuyến trạng thái liên kết
- Dùng giải thuật đường dẫn ngắn nhất để xác định các tuyến
- Khi mạng thay đổi thì thông tin trạng thái được gửi tràn ngập (flooding) đến các router láng giềng

## III. Các vấn đề liên mạng

1. Khái niệm liên mạng
2. Một số thiết bị liên mạng
3. Khái niệm về tunneling
4. Khái niệm về firewall
5. Khái niệm về mạng riêng ảo

# 1. Khái niệm liên mạng

- Liên mạng (internetwork): sự kết nối của nhiều mạng



# Sự khác nhau của các loại mạng

Thông số	Các khả năng
Dạng dịch vụ	Có kết nối, không kết nối
Các giao thức	IP, IPX, ...
Định địa chỉ	Phẳng (IEEE 802), có thứ bậc (IP)
Kích thước gói	Mỗi mạng có max riêng
Kiểm soát lỗi	Truyền tin cậy, có/không có số thứ tự
*****	*****

## 2. Một số thiết bị liên mạng

- Repeater (bộ lặp lại): hoạt động tại lớp physical
- Bridge (cầu nối): hoạt động tại lớp data link
- Switch (bộ chuyển mạch): hoạt động tại lớp data link
- Router (bộ định tuyến): hoạt động tại lớp network

## Một số thiết bị liên mạng (tt)

- Gateway (cổng nối): tên gọi tổng quát thiết bị liên mạng
  - Hoạt động tại một lớp  
Router: gateway tại lớp network
  - Hoạt động trên nhiều lớp

### 3. Khái niệm về tunneling (tạo đường hầm)

- Xử lý liên mạng tổng quát rất phức tạp

- Trường hợp đơn giản:

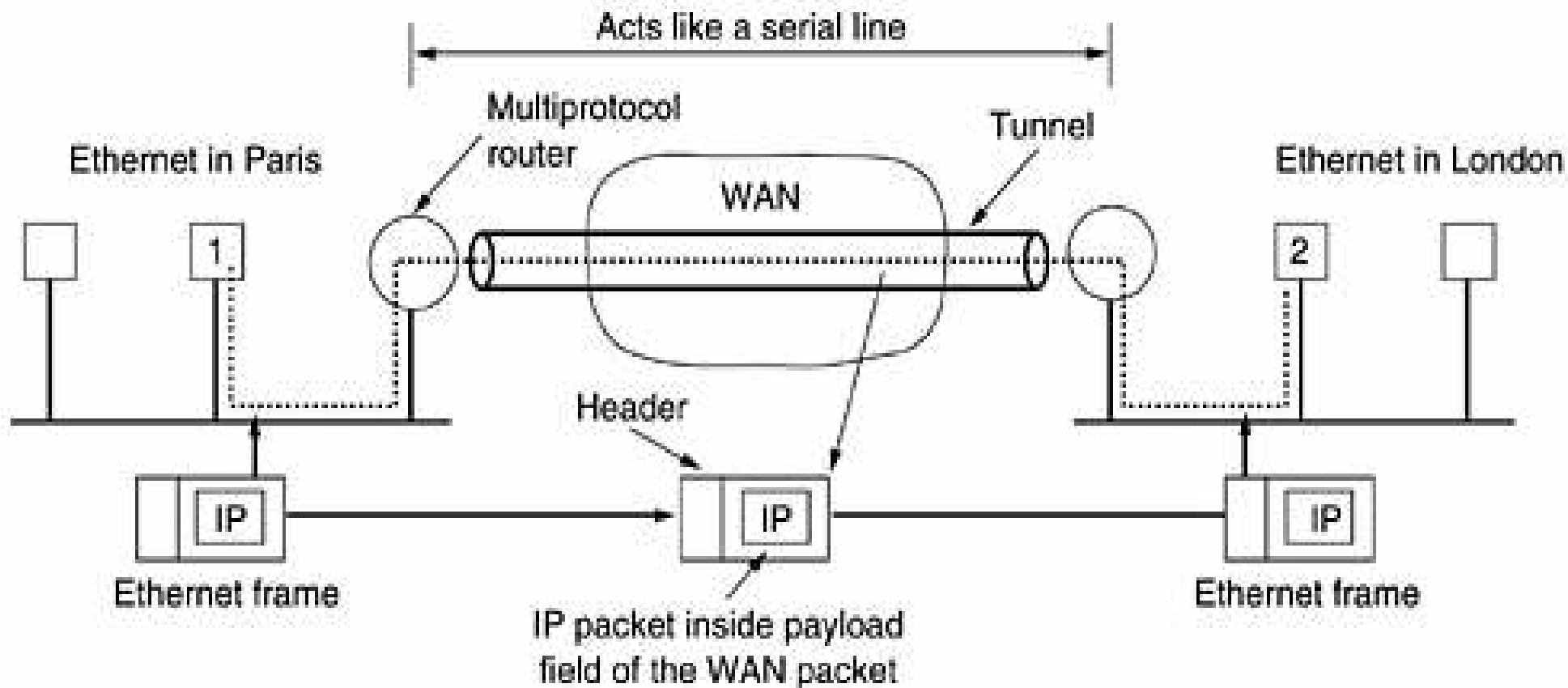
Máy gửi và máy nhận trên hai mạng cùng loại được kết nối bởi một mạng khác loại

ví dụ: dạng LAN-WAN-LAN

→ sử dụng kỹ thuật tạo đường hầm



# Ví dụ tunnel

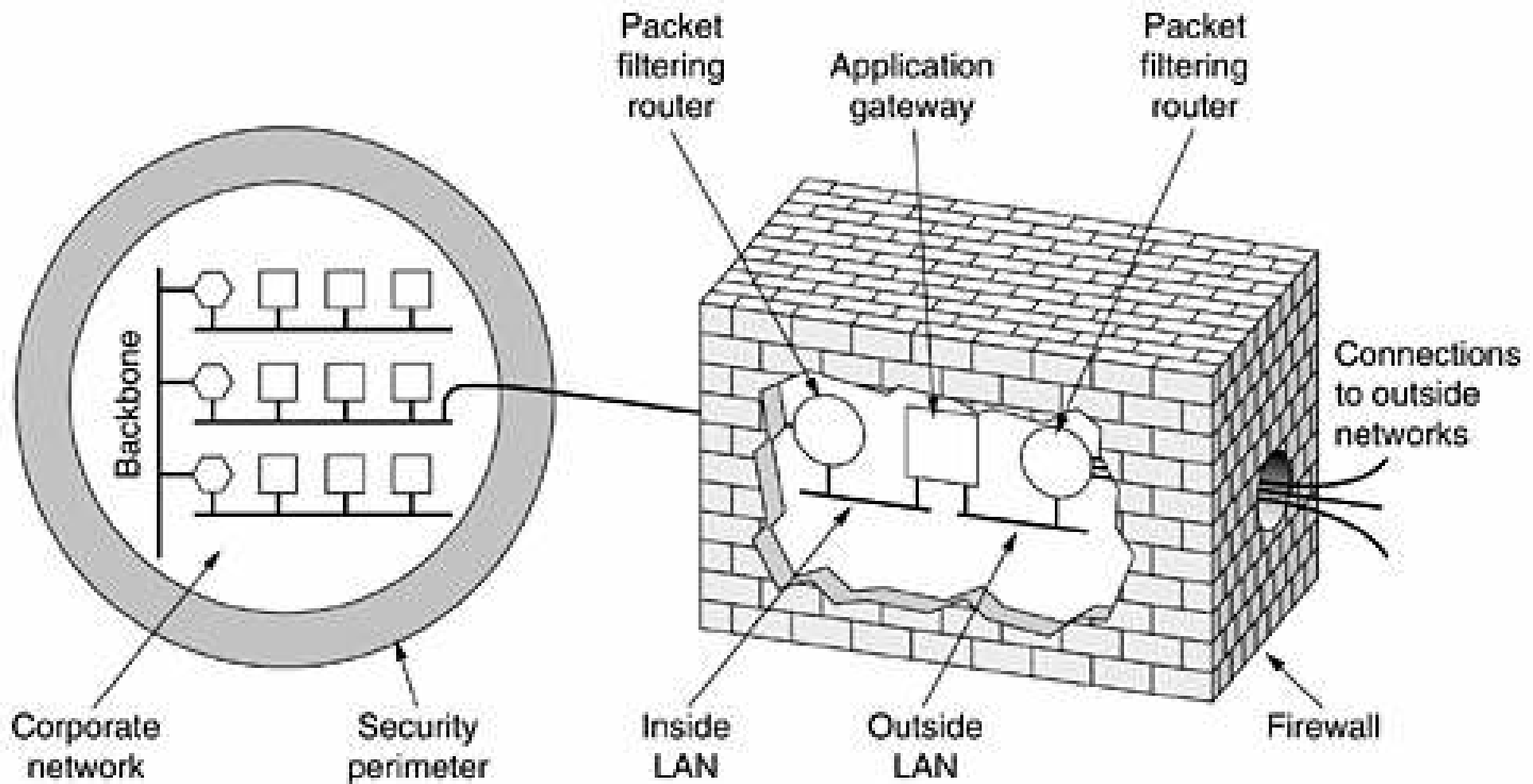


Hai router và mạng WAN đóng vai trò như đường hầm (tunnel) giữa hai mạng Ethernet

## 4. Khái niệm về firewall

- Là thiết bị liên mạng
- Mục đích: kiểm soát việc trao đổi dữ liệu
- Cấu tạo cơ bản:
  - Router lọc dữ liệu (packet filtering router)
    - Loại bỏ packet theo điều kiện xác định
  - Cổng nối ứng dụng (application gateway)
    - Hoạt động tại lớp ứng dụng, ví dụ Mail gateway
    - Kiểm tra nội dung dữ liệu

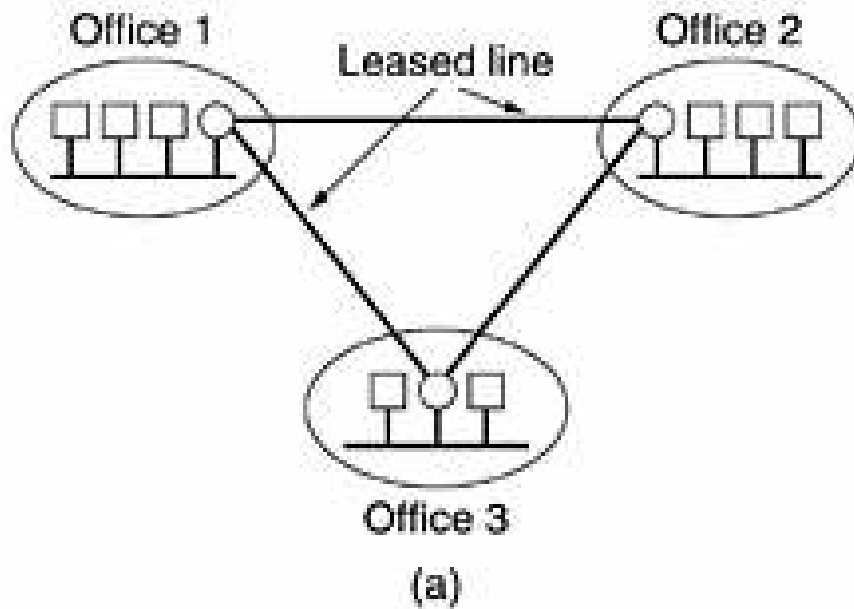
# Cấu trúc firewall



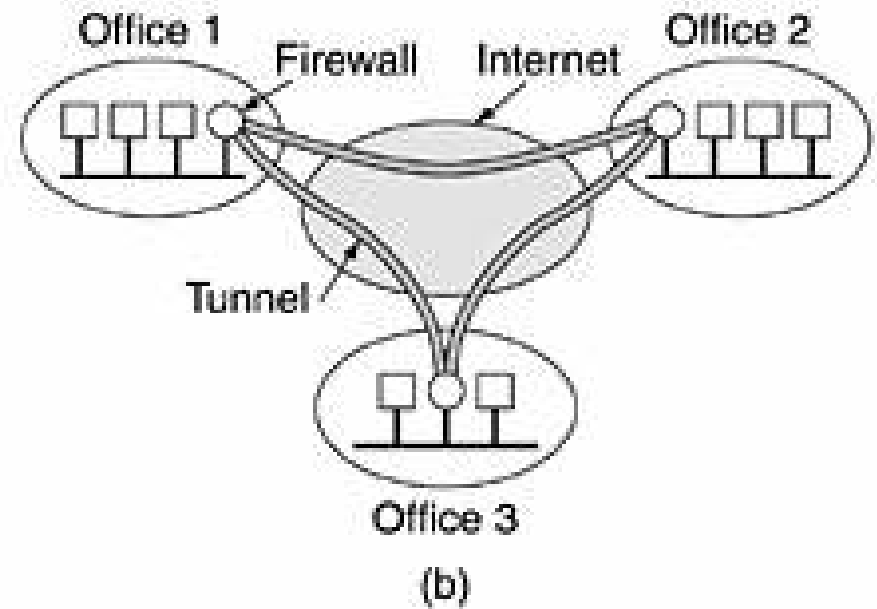
## 5. Khái niệm VPN (mạng riêng ảo)

- Mạng riêng (Private Network): mạng dùng riêng của một tổ chức
- Mạng riêng ảo (VPN, Virtual Private Network) là mạng riêng thiết lập trên nền tảng mạng công cộng với kỹ thuật tunneling và firewall

# Mạng riêng ảo



a. Mạng riêng  
(Private Network)



b. Mạng riêng ảo  
(VPN)

## IV. Lớp network trên mạng TCP/IP

1. Giới thiệu
2. Giao thức IP
3. Địa chỉ IP
4. Các giao thức điều khiển
5. Định tuyến trên Internet

# 1. Giới thiệu

- Tại lớp network, mạng Internet là sự kết nối của các mạng độc lập
- Lớp network trên mạng TCP/IP gọi là lớp Internet
- Nhiệm vụ lớp Internet: chọn tuyến để truyền dữ liệu (packet) giữa hai máy bất kỳ

# Các giao thức trên lớp Internet

## ■ IP (Internet Protocol)

- Truyền các gói dữ liệu dạng không kết nối

## ■ ARP (Address Resolution Protocol)

- Chuyển đổi địa chỉ IP thành địa chỉ lớp data link (địa chỉ MAC)

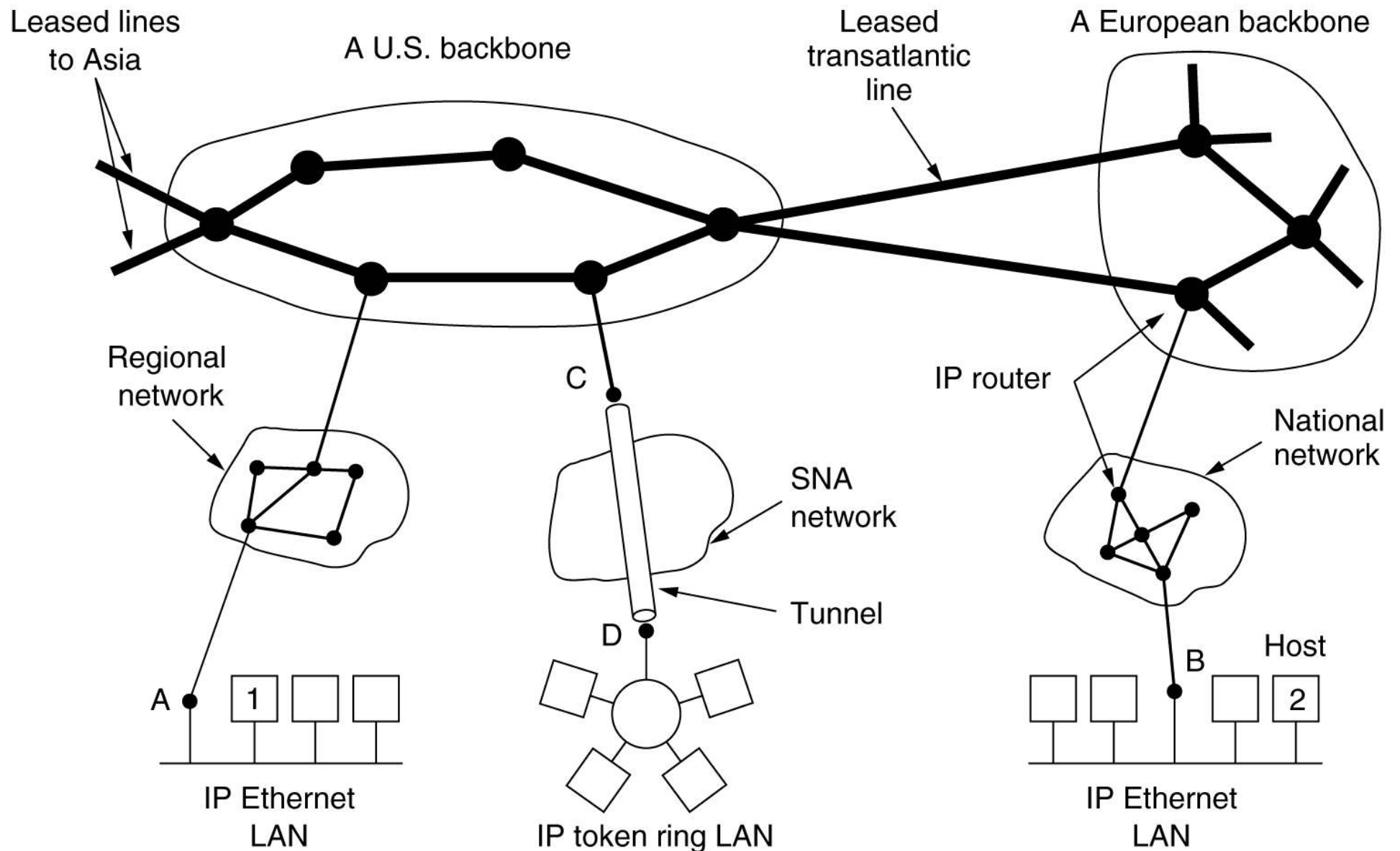
## ■ ICMP (Internet Control Message Protocol)

- Truyền các thông tin trạng thái, các thông điệp điều khiển

■ .....



# Mạng Internet: sự kết nối các mạng



# Hoạt động mạng Internet

- Lớp transport nhận dữ liệu từ lớp application, chia thành các gói dữ liệu, giao cho lớp network
- Lớp network truyền các gói dữ liệu đến máy nhận, các gói ban đầu có thể được chia thành các gói nhỏ hơn
- Khi tất cả các gói dữ liệu đến máy nhận, lớp network tạo lại các gói ban đầu, đưa cho lớp transport và đến lớp application

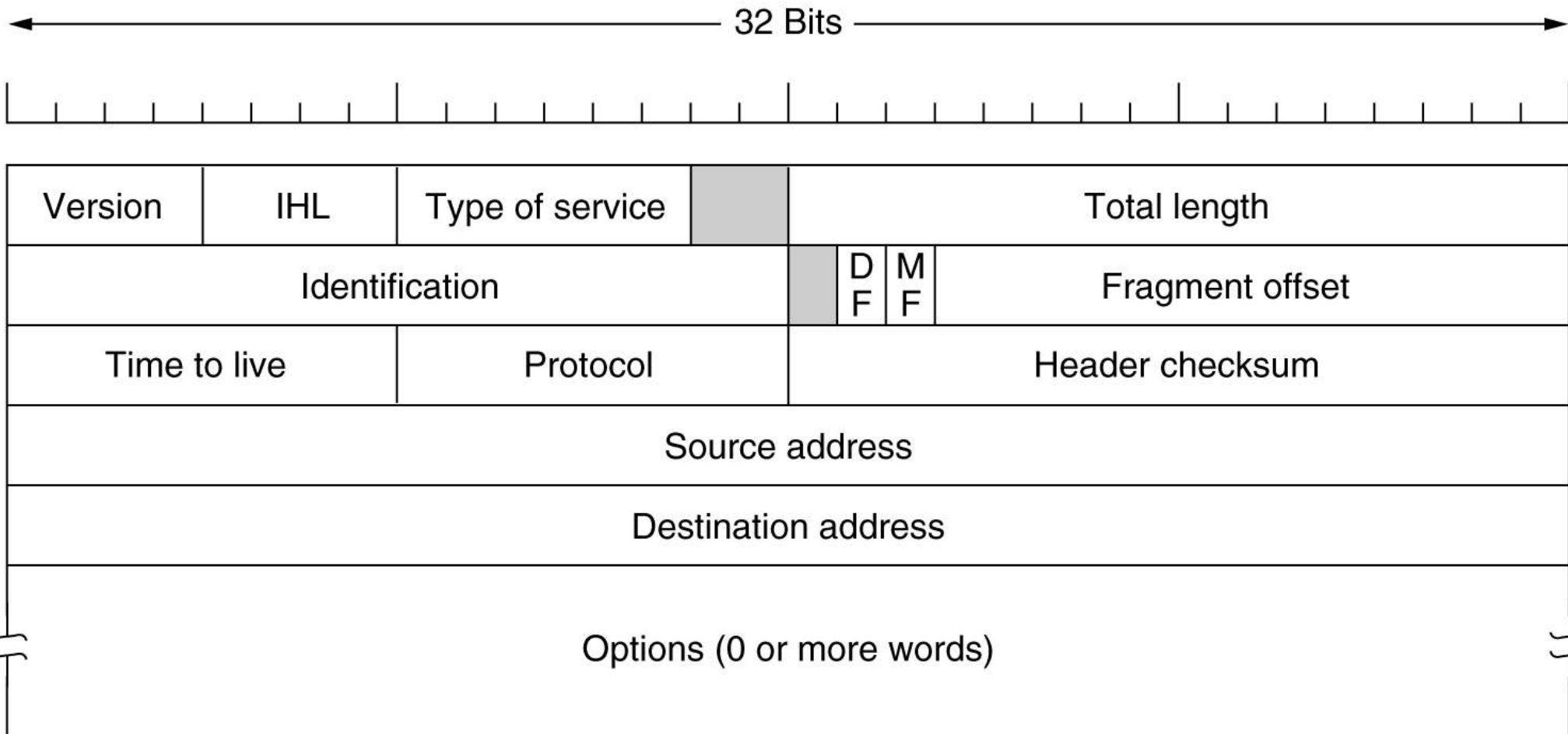
## 2. Giao thức IP

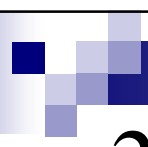
- Truyền dữ liệu dạng không kết nối
- Đơn vị dữ liệu: gói IP (IP packet)
  - IP Header  $\geq 20$  bytes
  - IP Data



- Khi chuyển sang mạng khác, gói IP có thể bị chia thành các gói nhỏ hơn

# IP header





## 3. Địa chỉ IP

- a. Khái niệm
- b. Các lớp địa chỉ IP
- c. Địa chỉ dành riêng, địa chỉ riêng
- d. Subnet
- e. CIDR (Classless InterDomain Routing)
- f. Đặt địa chỉ IP
- g. Dùng chung kết nối Internet

## a. Khái niệm

- Mỗi máy, bộ định tuyến có một địa chỉ luận lý lớp network, địa chỉ IP (IP address)
- Hai máy không thể có cùng địa chỉ IP
- Một máy có thể có nhiều địa chỉ IP nếu kết nối vào nhiều mạng

# Địa chỉ IP

- Giá trị nhị phân 32 bit, viết dưới dạng dotted-decimal

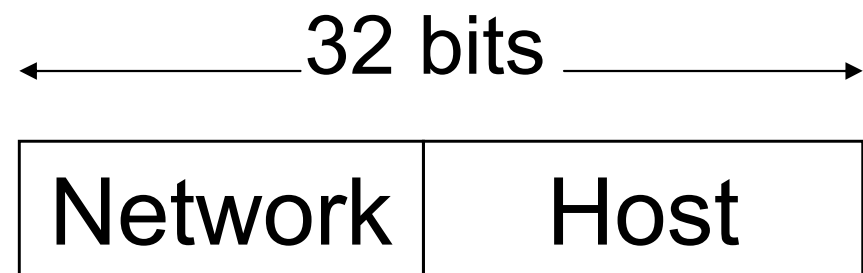
- Ví dụ:

11000000.10101000.00000001.00001000

192.168.1.8

- Gồm 2 phần

- Network address
- Host address



## Địa chỉ IP (tt)

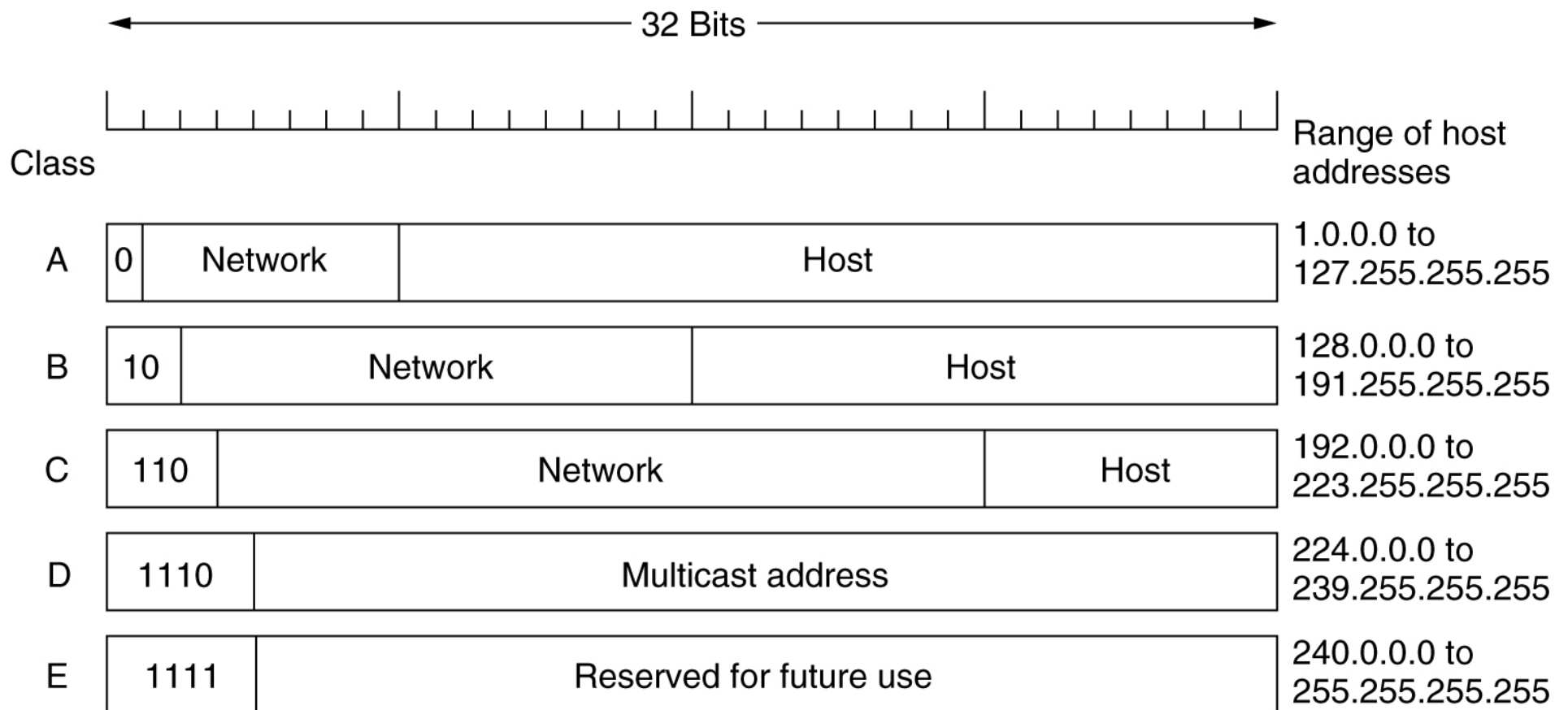
- Network addresses do ICANN cấp phát để tránh trùng địa chỉ  
(Internet Corporation for Assigned Names and Numbers)
- ICANN phân quyền cho các vùng, quốc gia, ví dụ VNNIC (VN Network Information Center), và ISPs



# Các dạng địa chỉ IP

- Theo lớp (classful addressing)
  - các lớp địa chỉ IP
  - không còn sử dụng
- Không theo lớp (classless addressing)
  - dạng CIDR  
(Classless InterDomain Routing)

## b. Các lớp địa chỉ IP



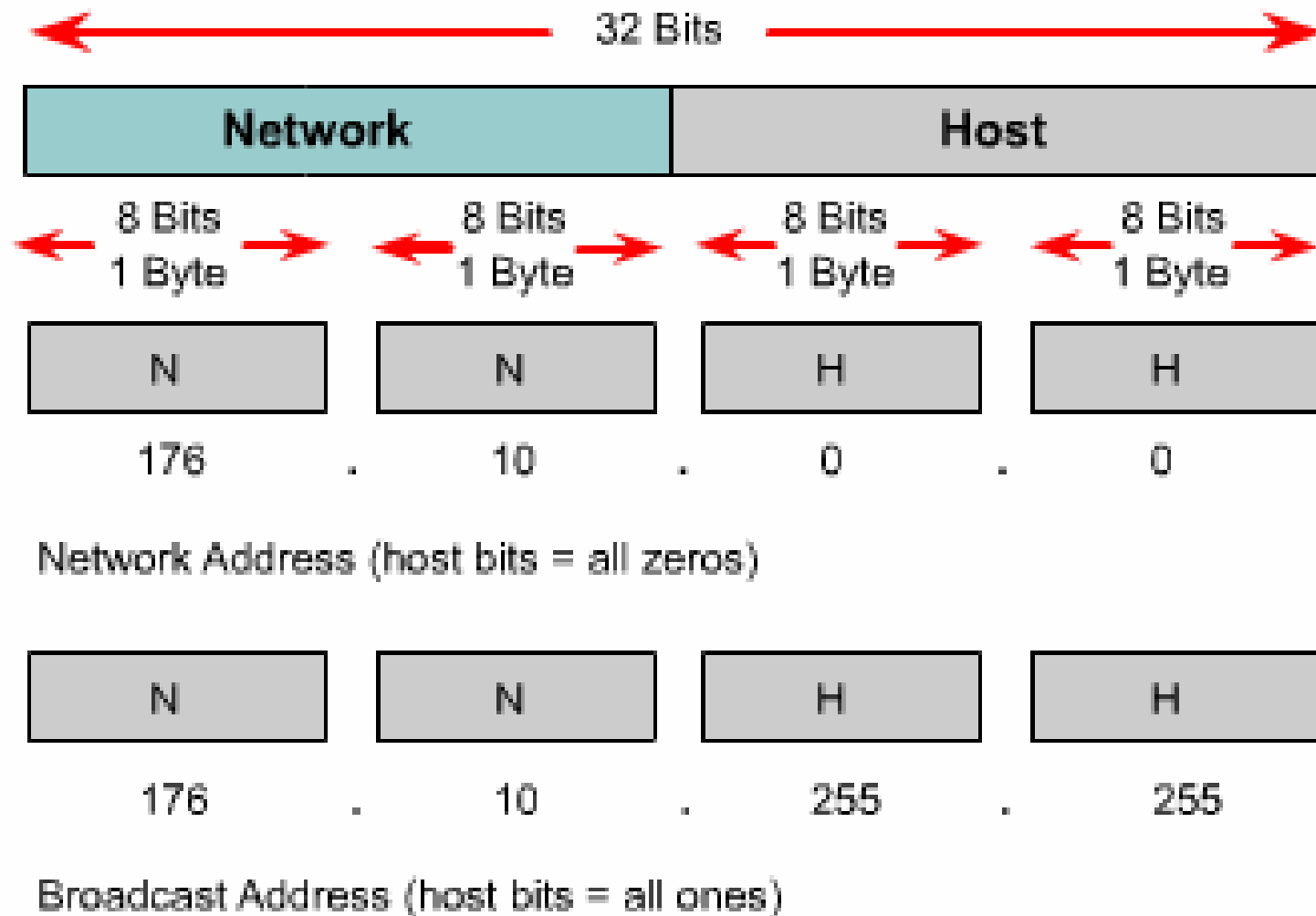
## c. Địa chỉ dành riêng, địa chỉ riêng

- Địa chỉ dành riêng (reserved addresses):  
không dùng làm địa chỉ máy
- Địa chỉ riêng (private addresses)  
dùng trên mạng riêng, không cấp phát  
trên Internet

# Địa chỉ dành riêng

- Địa chỉ mạng – Network address
  - Dùng xác định mạng
  - Vùng host toàn bit 0
- Địa chỉ quảng bá – Broadcast address
  - Dùng để gửi packet đến tất cả các máy trên một mạng
  - Vùng host toàn bit 1
- Địa chỉ vòng – Loopback
  - Dùng để kiểm tra
  - 127.x.y.z, giá trị thông dụng 127.0.0.1

# Ví dụ



Địa chỉ mạng, địa chỉ quảng bá của một mạng lớp B

## Địa chỉ riêng

Lớp A: 10.0.0.0 → 10.255.255.255

Lớp B: 172.16.0.0 → 172.31.255.255

Lớp C: 192.168.0.0 → 192.168.255.255

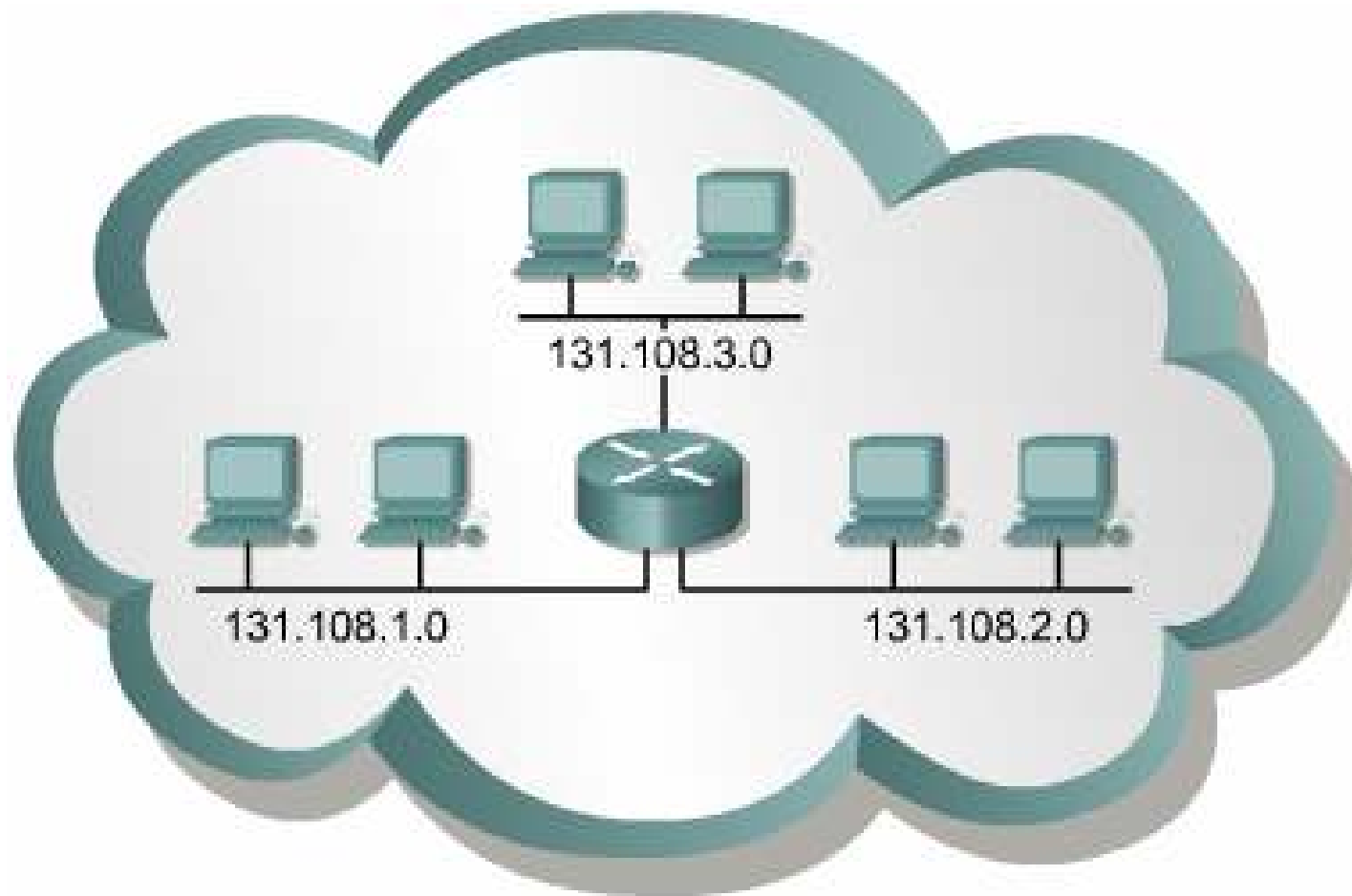
Dùng cho các máy:

- Trên mạng intranet
- Mạng dùng riêng

## d. Subnet

- Địa chỉ mạng trong địa chỉ IP là mạng luận lý
- Các máy trên cùng một mạng phải có cùng phần địa chỉ mạng (network) trong địa chỉ IP
- Mạng luận lý không tương ứng với một mạng cục bộ
- Subnetting là kỹ thuật chia mạng nhiều máy thành các mạng nhỏ hơn (subnet)

# Ví dụ



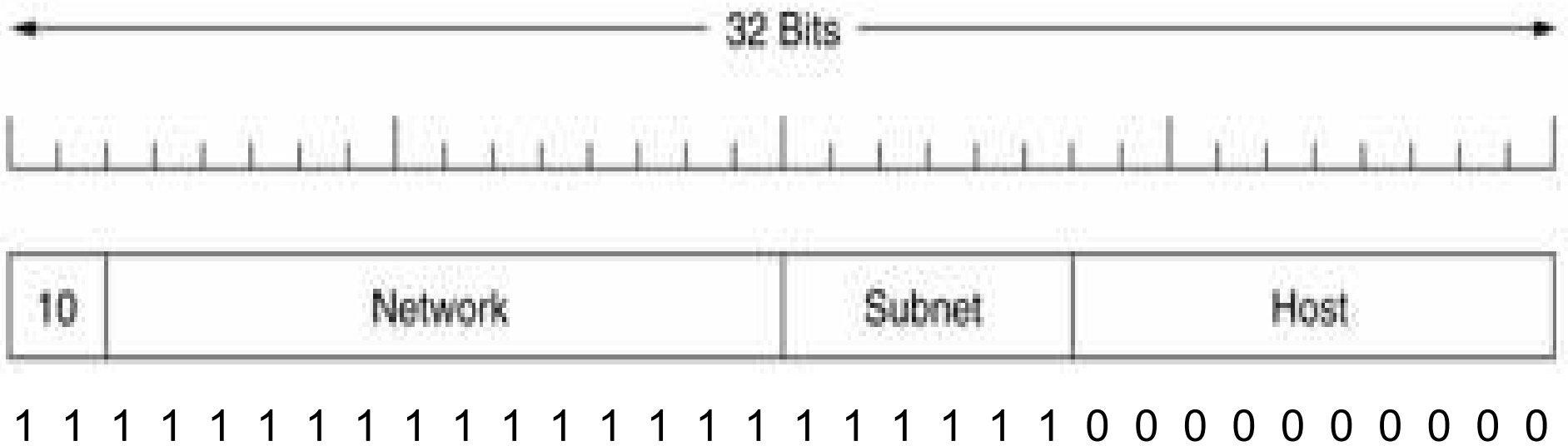
Các subnet 131.108.1.0, 131.109.2.0, 131.108.3.0  
trong network 131.108.0.0



# Subnet mask

- Trong địa chỉ IP cần có thêm vùng subnet được lấy từ vùng host
- Subnet mask là giá trị xác định số bit của vùng network + vùng subnet trong địa chỉ IP
- Hình thức subnet mask:
  - Dotted-decimal, ví dụ 255.255.252.0
  - Slash: /n, với n là số bit network+subnet ví dụ /22

# Ví dụ



Một mạng lớp B được chia thành 64 mạng nhỏ

Subnet mask : 255.255.252.0 /22

# Xác định giá trị subnet

## từ địa chỉ IP và subnet mask

### ■ Dùng hàm AND

### ■ Ví dụ:

• Địa chỉ IP: 130.50.15.6

10000010.00110010.00001111.00000110

• Subnet mask: 255.255.252.0 /22

11111111.11111111.11111100.00000000

→ Subnet: 130.50.12.0

10000010.00110010.00001100.00000000



## e. CIDR

Cấp phát các khối địa chỉ IP:

- có kích thước thay đổi
- không theo lớp địa chỉ
- tồn tại như một mạng trên Internet

# Ví dụ

University	First address	Last address	How many	Written as
Cambridge	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
(Available)	194.24.12.0	194.24.15.255	1024	194.24.12/22
Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20

C: 11000010.00011000.00000000.00000000

mask 11111111.11111111.11111000.00000000

E: 11000010.00011000.00001000.00000000

mask 11111111.11111111.11111100.00000000

O: 11000010.00011000.00010000.00000000

mask 11111111.11111111.11110000.00000000

## Ví dụ (tt)

- Xét địa chỉ 194.24.17.4

11000010.00011000.00010001.00000100

- Thực hiện AND với các mask của 3 mạng trên  
→ 194.24.17.4 thuộc mạng Oxford

# Tác dụng của CIDR

- Sử dụng hiệu quả không gian địa chỉ IP
  - Giảm số lượng mạng
    - Nhiều mạng lớp C tồn tại như một mạng
    - Có thể kết hợp nhiều mạng thành một mạng
- Ví dụ: 3 mạng trong ví dụ trên có thể được kết hợp thành một mạng 194.24.0.0/19

## f. Đặt địa chỉ IP

### ■ Địa chỉ tĩnh

- Do administrator đặt

### ■ Địa chỉ động

- Do DHCP server cấp phát

### ■ Các thành phần

- IP address
- Subnet mask
- Default gateway address, .....



# Kiểm tra địa chỉ IP

Các công cụ:

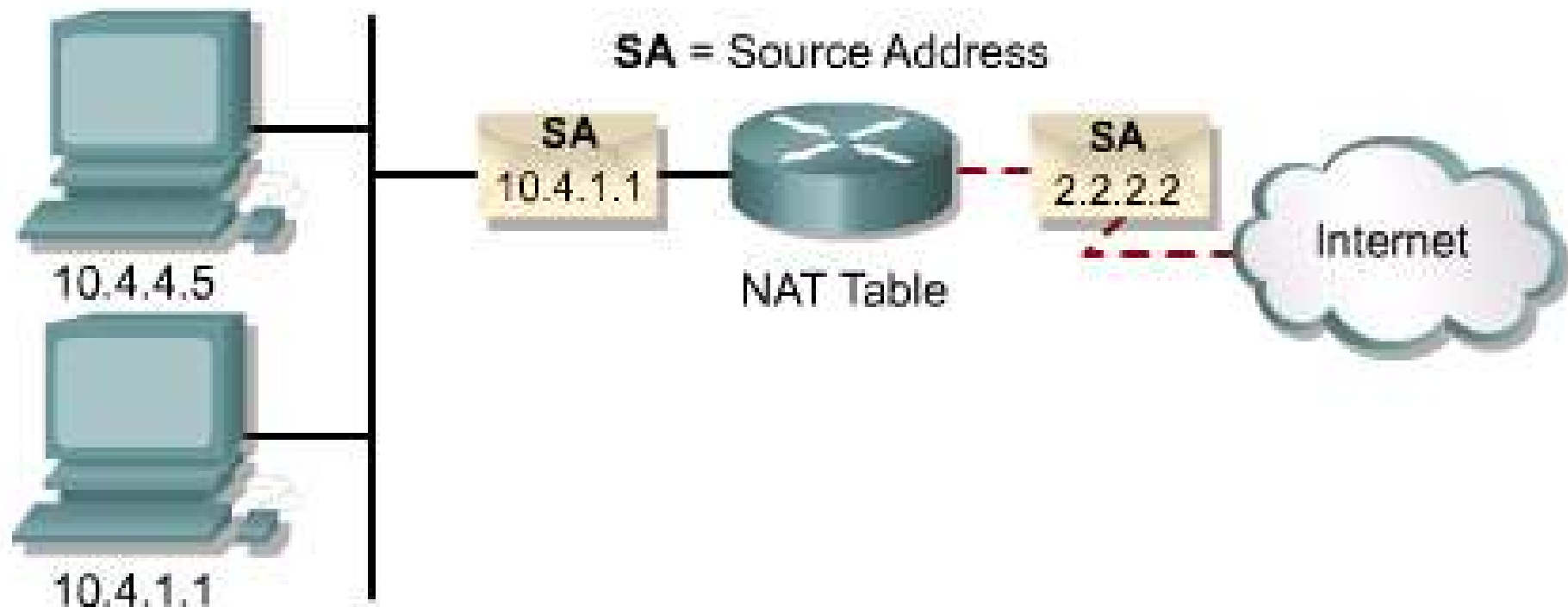
- ipconfig, winipcfg (windows 9x)  
cung cấp các thông tin ip address, subnet mask, default gateway, ...
- ping  
kiểm tra kết nối theo IP

## ■ g. Dùng chung kết nối Internet

### (Internet Connection Sharing)

- Các máy trên một LAN, sử dụng địa chỉ IP riêng
- Có một kết nối Internet, sử dụng địa chỉ IP toàn cục
- Cần khôi chuyển đổi địa chỉ NAT (Network Address Translation), có thể là:
  - Thiết bị
  - Phần mềm

# Ví dụ



Địa chỉ địa phương: 10.4.4.5, 10.4.1.1

Địa chỉ toàn cục: 2.2.2.2

# Hoạt động của khối NAT

- Khi một máy X gửi dữ liệu ra ngoài mạng thì gửi đến khối NAT
- Khối NAT thay thế địa chỉ máy gửi trên gói IP bằng địa chỉ toàn cục
- Khi có đáp ứng từ bên ngoài, khối NAT:
  - Nhận dữ liệu
  - Kiểm tra bảng chuyển đổi địa chỉ
  - Thay thế địa chỉ máy nhận trên gói IP bằng địa chỉ máy X



## 4. Các giao thức điều khiển

### a. DHCP

(Dynamic Host Configuration Protocol)

### b. ARP

(Address Resolution Protocol)

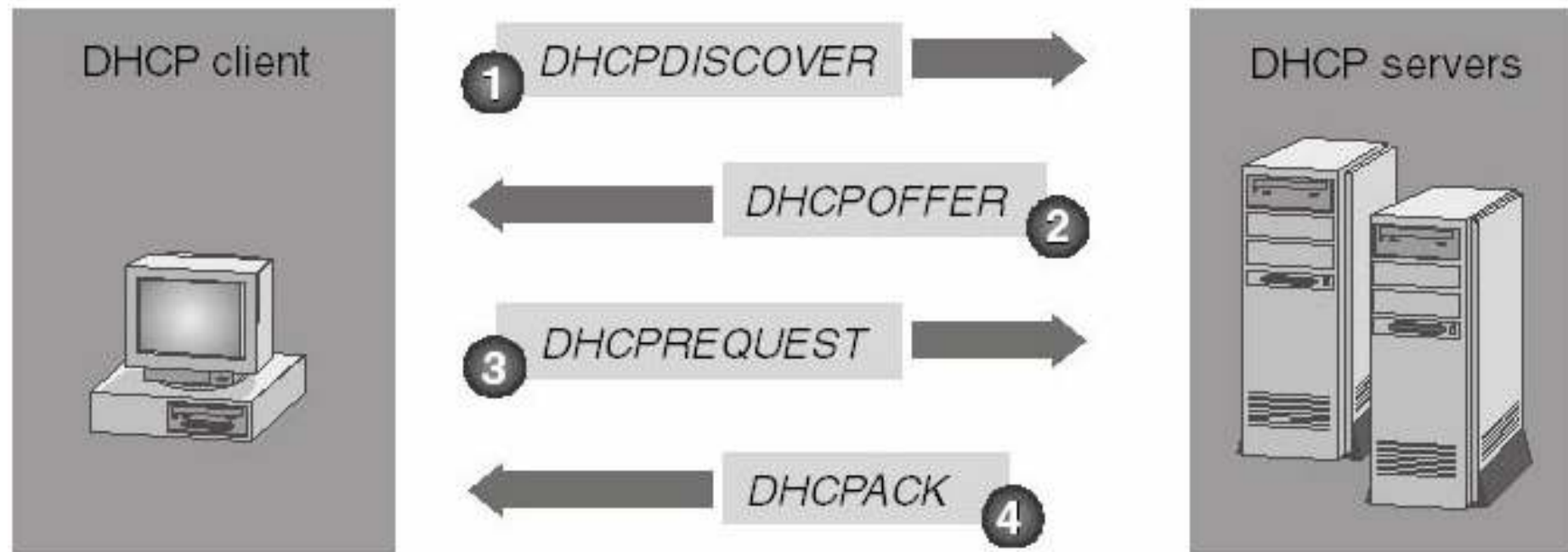
### c. ICMP

(Internet Control Message Protocol)

## a. DHCP

- DHCP server cấp các thông số địa chỉ IP cho DHCP Client:
  - IP address
  - Subnet mask
  - Options: gateway address, DNS Server, ...
- Mục đích:
  - Đơn giản công việc quản trị mạng
  - Sử dụng hiệu quả địa chỉ IP

# Các giai đoạn cấp địa chỉ IP động



**DHCPDISCOVER**: client tìm server

**DHCPOFFER**: server cung cấp thông số IP

**DHCPREQUEST**: client thông báo đã nhận

**DHCPACK**: server chấp nhận

## b. ARP

- Chuyển đổi địa chỉ IP thành địa chỉ MAC để truyền thông bên trong một mạng
- Cần khôi thực hiện giao thức ARP
- Khôi ARP xây dựng và duy trì một bảng chứa các phần tử (IP address – MAC address)



## c. ICMP

- Giao thức IP dùng để gửi dữ liệu
- Giao thức ICMP dùng để gửi các thông báo lỗi và các thông tin điều khiển
- Ví dụ:
  - Thông báo không đến được máy nhận
  - Kiểm tra một máy có tồn tại
- Thông điệp ICMP được gửi trên gói IP

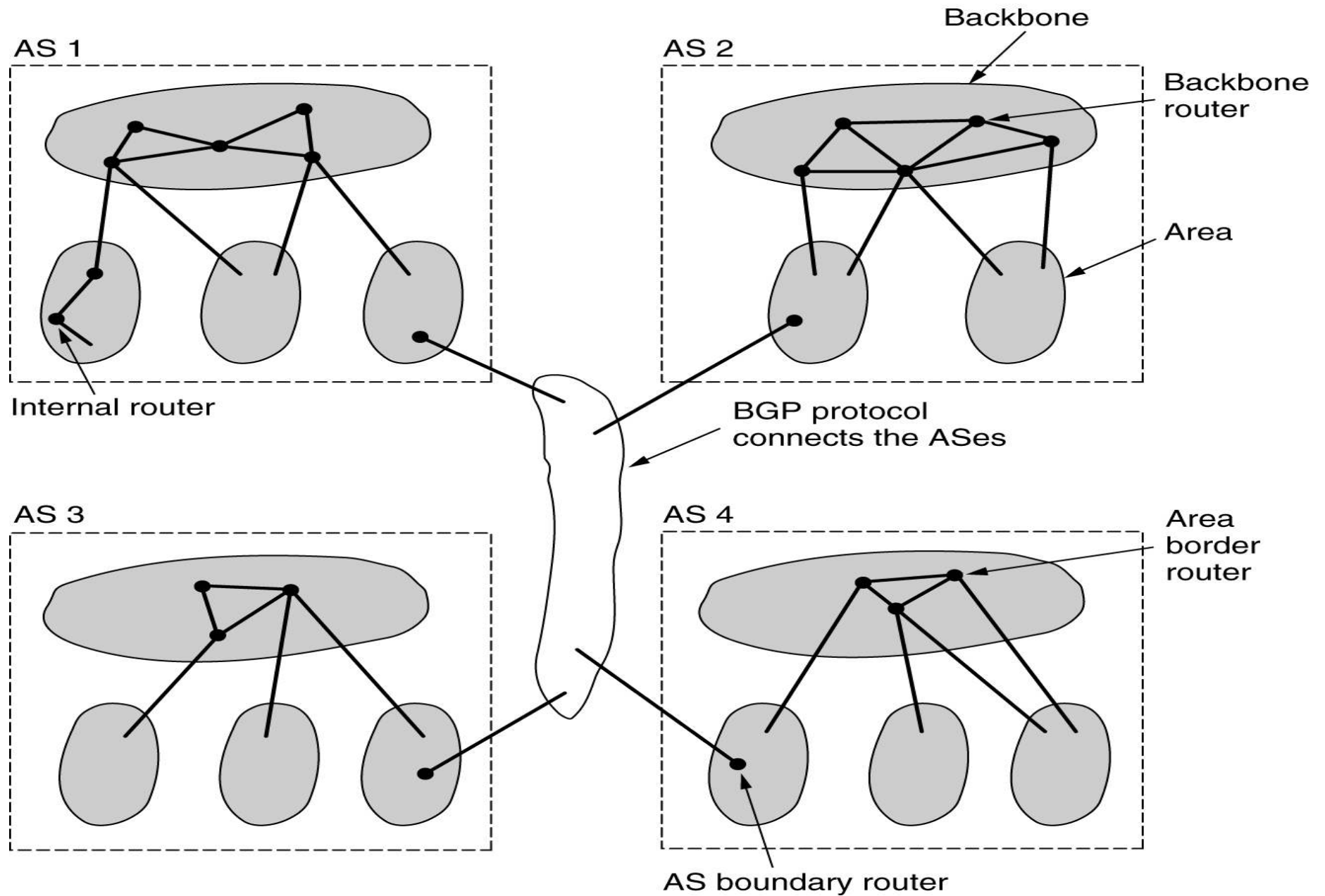
# Một số dạng thông điệp ICMP

<b>Message type</b>	<b>Description</b>
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

## 5. Định tuyến trên Internet

- Tại lớp Network, mạng Internet là tập hợp các mạng độc lập (Autonomous System)
- Có 2 dạng giao thức định tuyến:
  - Interior Gateway Protocol  
thực hiện bên trong AS, ví dụ OSPF (Open Shortest Path First)
  - Exterior Gateway Protocol  
thực hiện giữa các AS, ví dụ BGP (Border Gateway Protocol)

# Ví dụ



## V. Giới thiệu IPv6

- Dùng 128 bit địa chỉ, viết dưới dạng colon-hexadecimal
- Các đặc điểm chính, so với IPv4:
  - Không gian địa chỉ lớn ( $\sim 3.4 * 10^{38}$ )
  - Phần header đơn giản hơn
  - Hỗ trợ tốt hơn các tùy chọn (options)
  - Bảo mật
  - Chất lượng dịch vụ tốt hơn

# NHẬP MÔN MẠNG MÁY TÍNH

## Chương 5

### LỚP TRANSPORT (LỚP GIAO VẬN)



# Nội dung chương 5

- I. Các vấn đề thiết kế lớp transport
- II. Lớp transport trên mạng TCP/IP
- III. Giới thiệu giao diện lập trình mạng socket



# I. Các vấn đề thiết kế lớp transport

1. Nhiệm vụ lớp transport
2. Dịch vụ lớp transport



# 1. Nhiệm vụ lớp transport

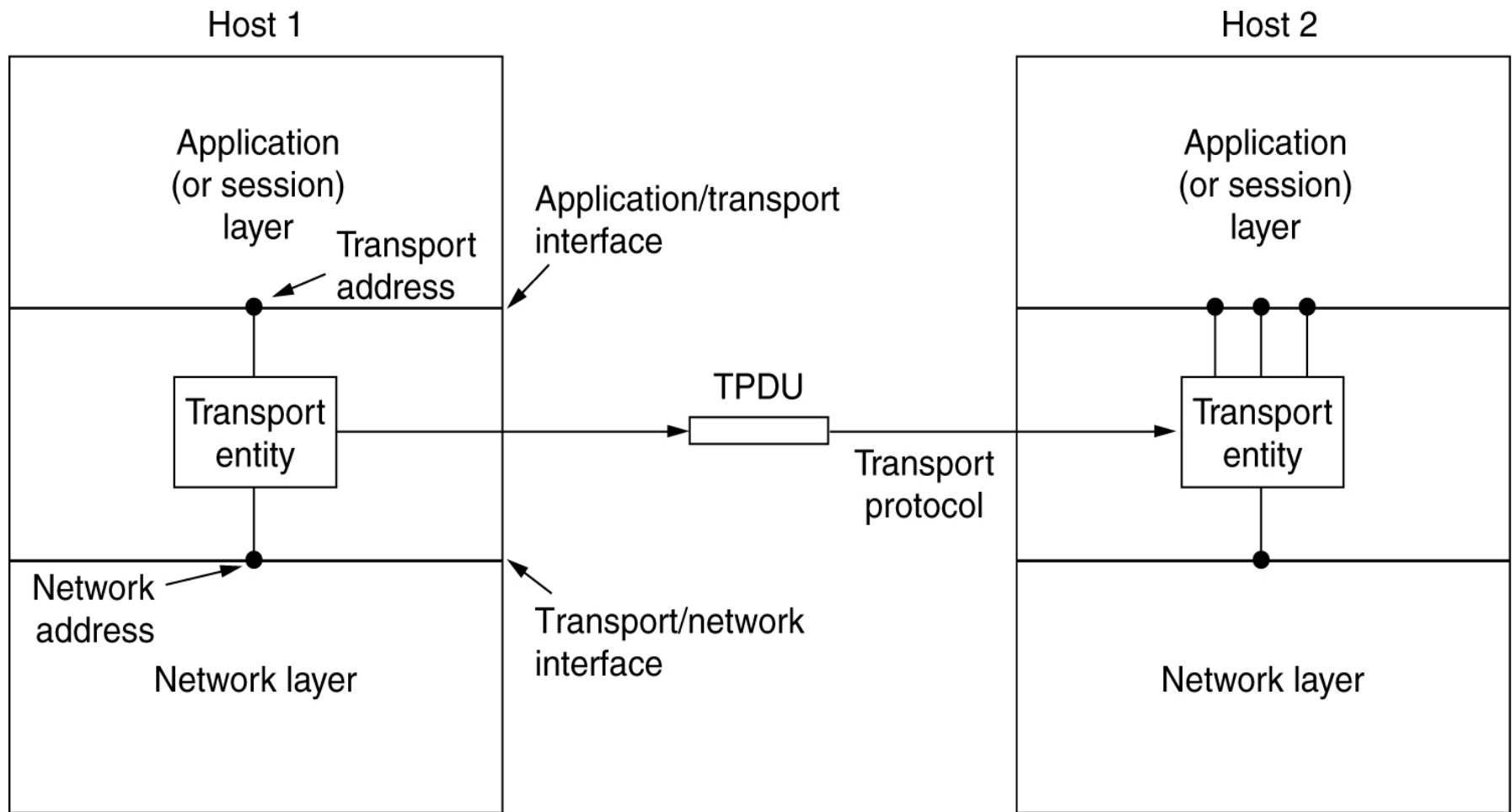
- Cung cấp dịch vụ gửi nhận dữ liệu tin cậy giữa các chương trình trên hai máy bất kỳ
- Thực hiện:
  - Chia và ghép dữ liệu từ lớp application
  - Kiểm soát lỗi, kiểm soát lưu lượng
- Lớp transport có vai trò quan trọng trên kiến trúc mạng nhiều lớp



## 2. Dịch vụ lớp transport

- a. Dịch vụ lớp transport
- b. Các thao tác cơ sở

# a. Dịch vụ lớp transport



Quan hệ giữa các lớp

# Các thuật ngữ

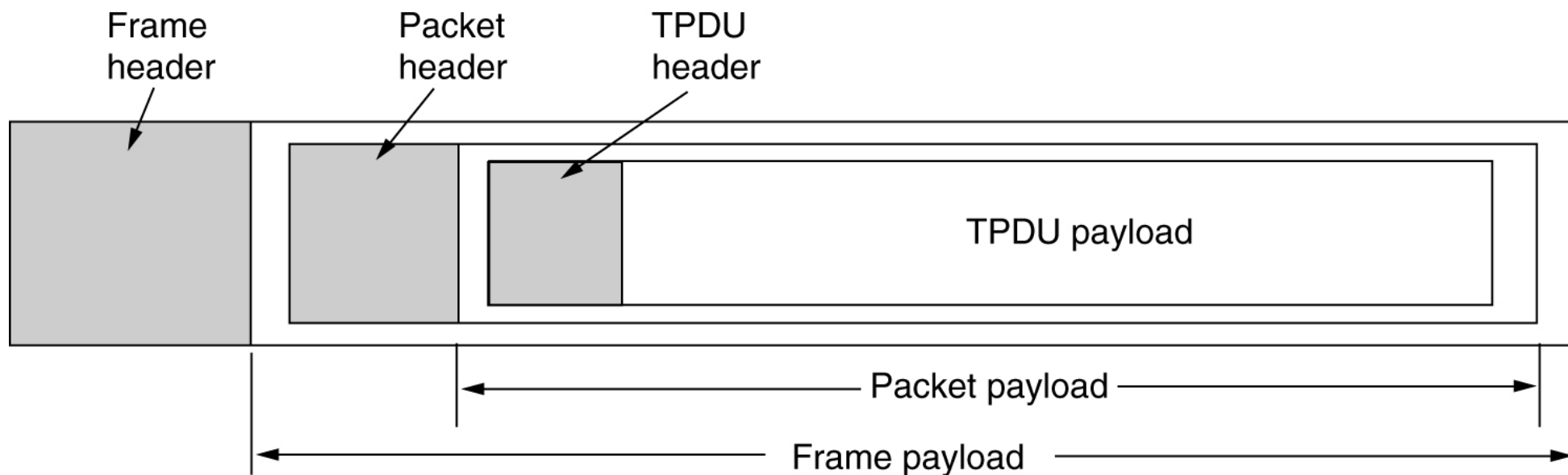
- Transport entity: thực thể lớp transport
- TPDU (Transport Protocol Data Unit): đơn vị dữ liệu giao thức lớp transport
- Transport address: địa chỉ lớp transport
  - Transport Service Access Point
  - Port (mạng TCP/IP)
- Network address: địa chỉ lớp network
  - Địa chỉ IP (mạng TCP/IP)



## Các dạng dịch vụ

- Có kết nối (connection-oriented service)
- Không kết nối (connectionless service)

# Đơn vị dữ liệu giao thức lớp transport



TPDU trong packet và frame



## b. Các thao tác cơ sở

(Transport service primitives)

- Các thao tác cơ sở của dịch vụ đơn giản

- Ví dụ:

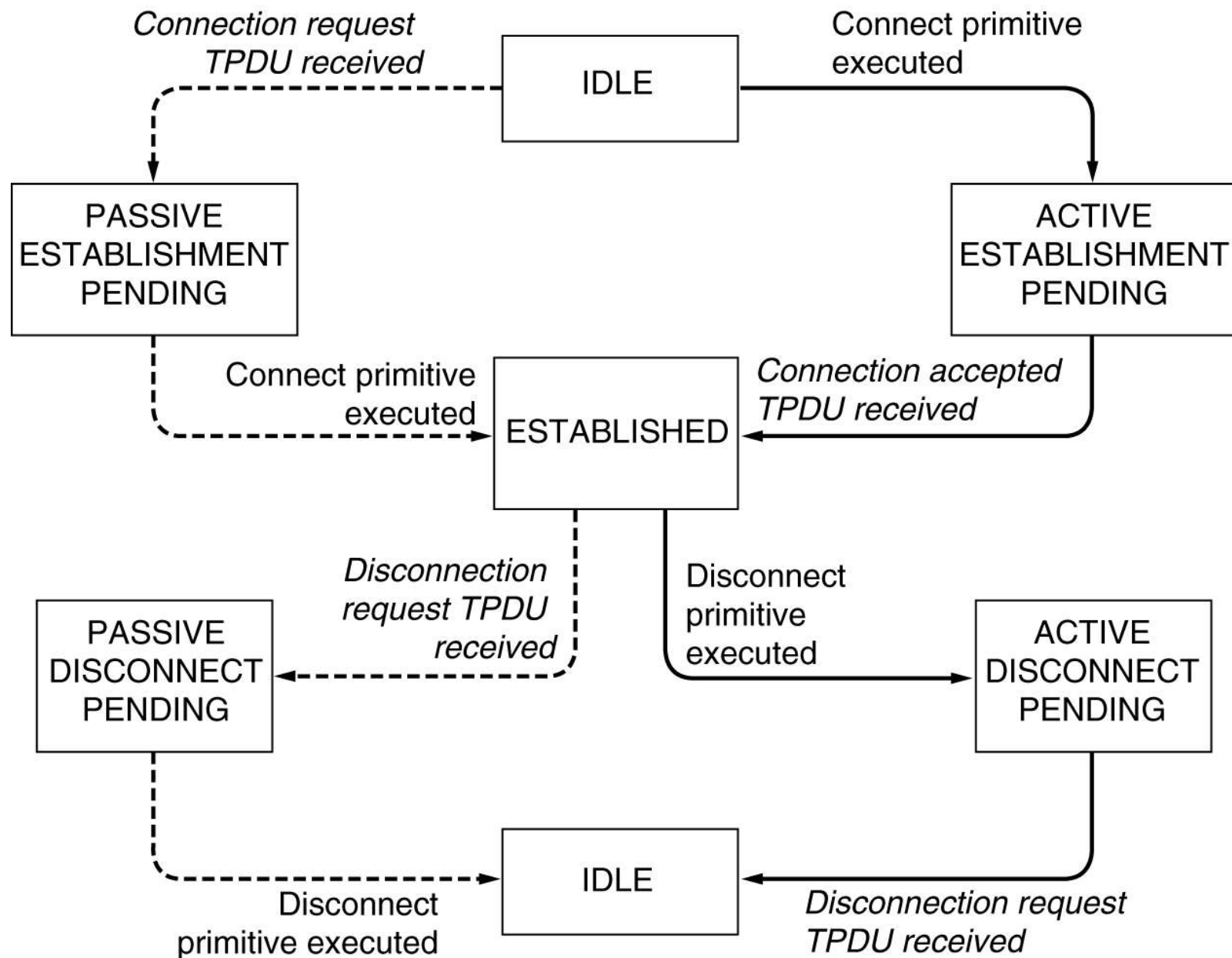
Mô hình client-server dạng có kết nối

# Các thao tác cơ sở của dịch vụ đơn giản

Primitive	Dữ liệu gửi	Ý nghĩa
LISTEN	Không có	Chờ process khác kết nối
CONNECT	CONNECTION REQUEST	Thiết lập kết nối
SEND	DATA	Gửi dữ liệu
RECEIVE	Không có	Chờ nhận dữ liệu
DISCONNECT	DISCONNECTION REQUEST	Yêu cầu hủy kết nối



# Mô hình Client-Server dạng có kết nối





## II. Lớp transport trên mạng TCP/IP

1. Giao thức TCP  
(Transmission Control Protocol)
2. Giao thức UDP  
(User Datagram Protocol)



# 1. Giao thức TCP

- a. Giới thiệu TCP
- b. Mô hình dịch vụ TCP
- c. Giao thức TCP
- d. TCP segment header
- e. Thiết lập kết nối TCP

## a. Giới thiệu TCP

- Cung cấp dịch vụ gửi nhận chuỗi byte tin cậy giữa hai chương trình trên mạng có thể không tin cậy
- Thực thể TCP:
  - Thư viện
  - User process
  - Kernel
- Chia dữ liệu từ process ứng dụng, gửi trên các gói IP

## b. Mô hình dịch vụ TCP

- Dịch vụ TCP thực hiện trên kết nối TCP (TCP connection)
- Kết nối TCP bao gồm hai đầu cuối (end-point), được gọi là socket
- Socket number (socket address):
  - Địa chỉ IP – 32 bit
  - Port – 16 bit

# Port

- Khái niệm trừu tượng → nhiều ứng dụng TCP trên một máy
- Well-known ports: dùng cho các dịch vụ chuẩn, ví dụ:

Port	21: FTP	- File Transfer Protocol
	25: SMTP	- Email
	80: HTTP	- Web



# Kết nối TCP

- Full-duplex
- Point-to-point
- Byte stream

## c. Giao thức TCP

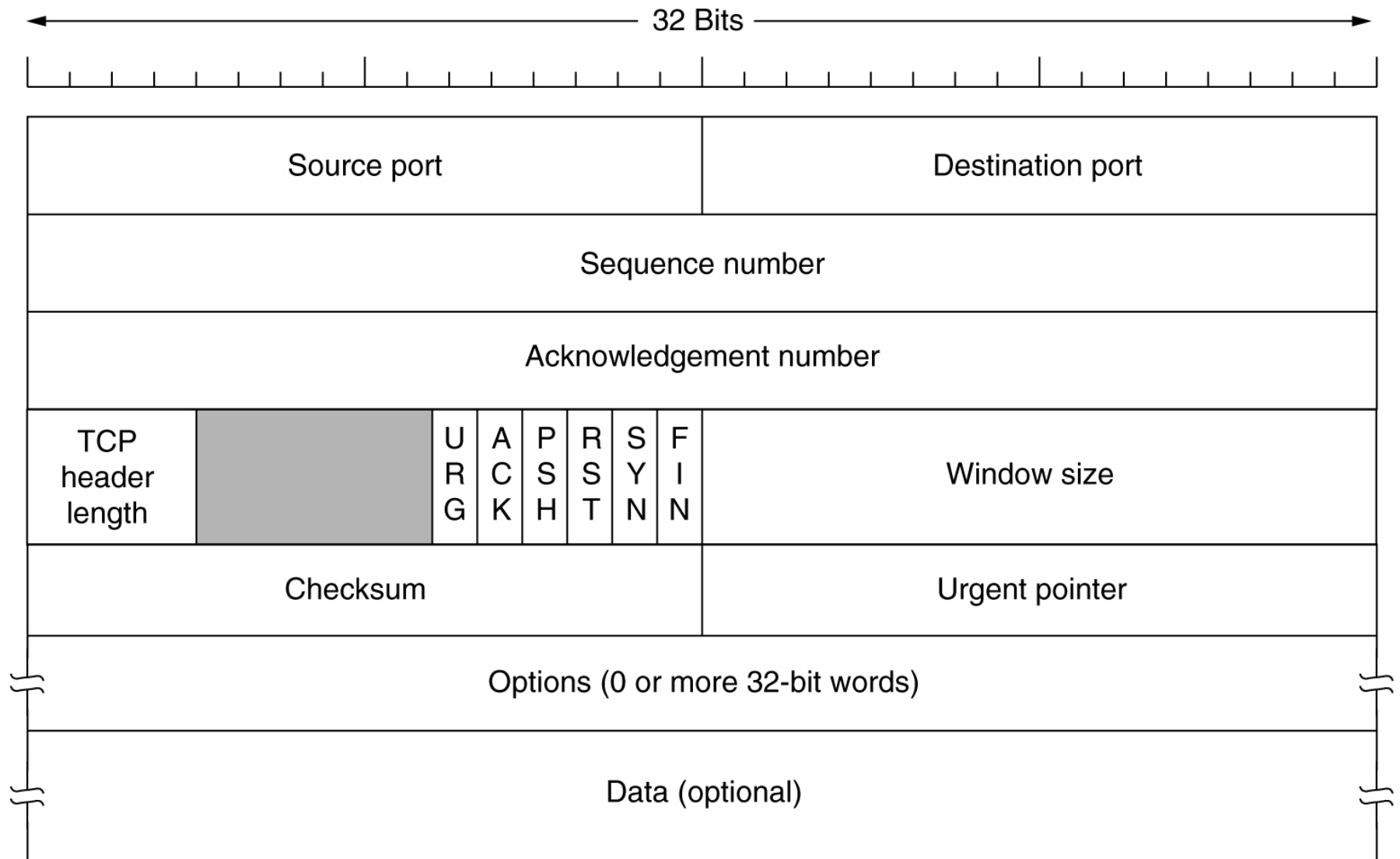
- Đơn vị dữ liệu: TCP segment
  - TCP header  $\geq 20$  bytes
  - TCP data  $\geq 0$  bytes
- Kích thước TCP segment bị giới hạn bởi:
  - IP payload (65515 bytes)
  - MTU (Maximum Transfer Unit)  
Ví dụ: MTU mạng Ethernet  $\sim 1500$  bytes



## Giao thức TCP (tt)

- Mỗi byte truyền trên kết nối TCP có số thứ tự trình tự (sequence number) 32 bit
- Giao thức cơ bản: sliding window
  - Sender gửi segment, khởi động timer
  - Receiver gửi segment có kèm ACK number là số thứ tự byte chờ nhận tiếp theo
  - Sender sẽ gửi lại nếu không có ACK khi hết thời gian

# d. TCP segment header



## e. Thiết lập kết nối TCP

Thiết lập kết nối giữa

Host 1 (Client) và Host 2 (Server)  
(Three-way handshake)

■ Host 1 → Host 2:

$seq=x, ack=0, SYN=1, ACK=0$

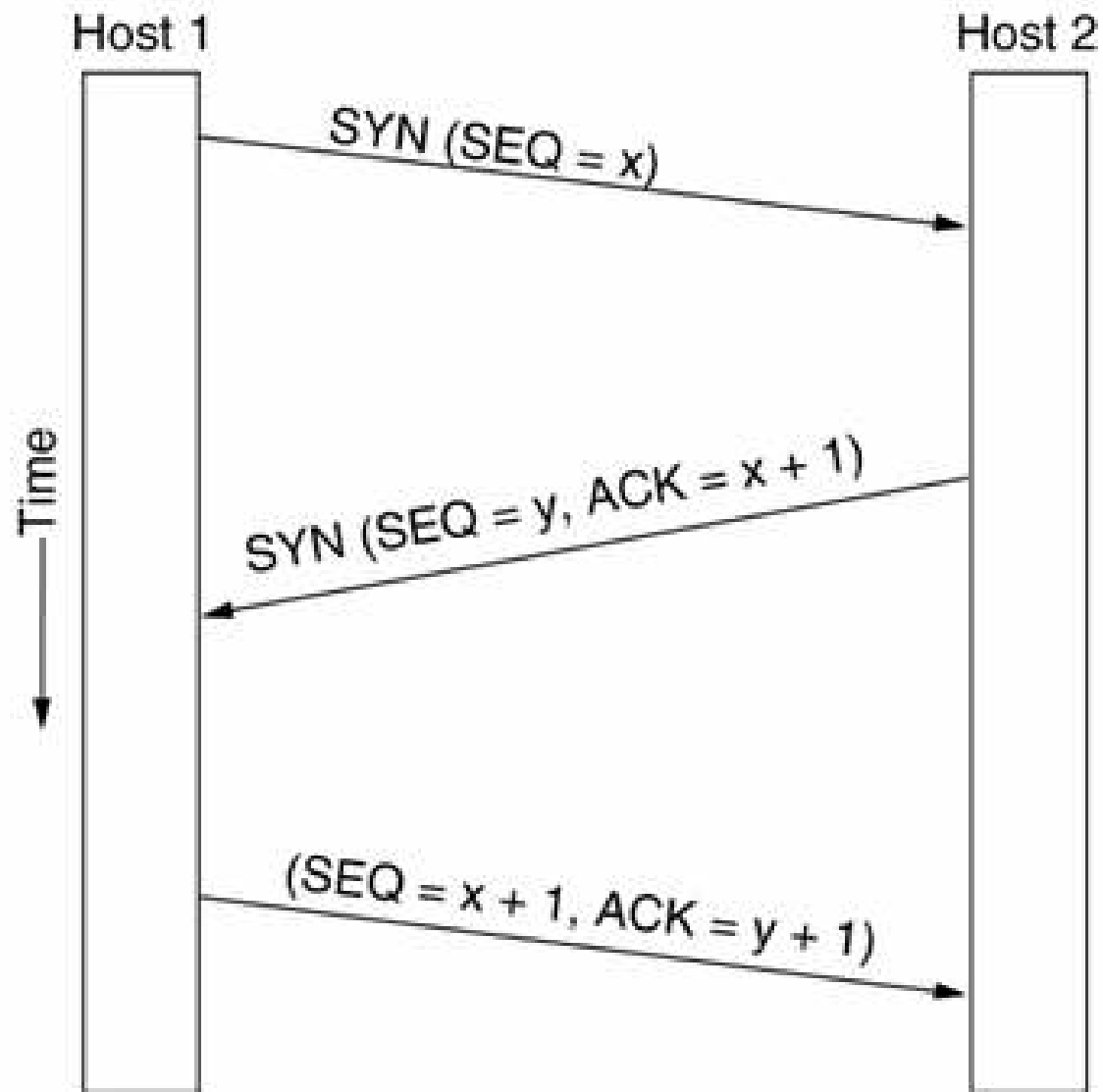
■ Host 2 → Host 1:

$seq=y, ack=x+1, SYN=1, ACK=1$

■ Host 1 → Host 2:

$seq=x+1, ack=y+1, SYN=0, ACK=1$

# Sơ đồ thiết lập kết nối TCP



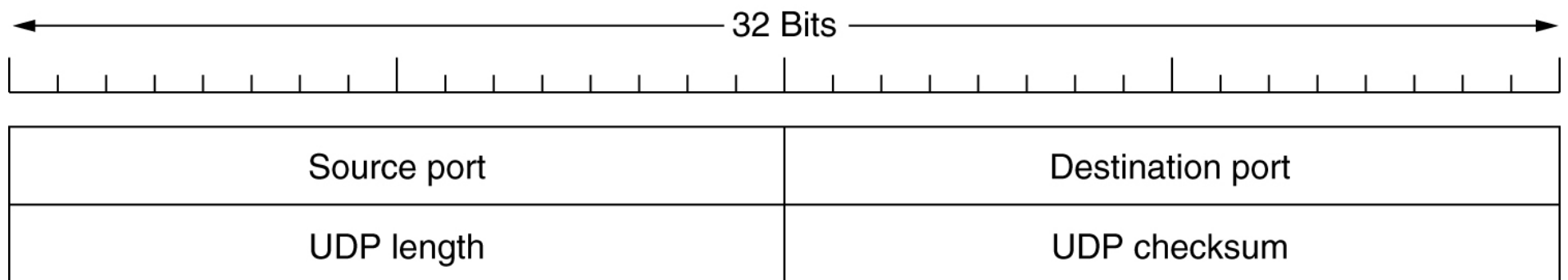
# Hủy bỏ kết nối TCP

- Gởi TCP segment với FIN=1
- Cần một FIN segment và một ACK segment cho một bên truyền thông

## 2. Giao thức UDP

- Giao thức dạng không kết nối
- Không có kiểm soát lỗi
  - nếu cần thì thực hiện trên lớp application
- Đơn vị dữ liệu: UDP datagram/segment
  - UDP header: 8 bytes
  - UDP data
- Sử dụng khái niệm port tương tự TCP

# UDP header



Source port: địa chỉ port chương trình gửi

Destination port: địa chỉ port chương trình nhận

UDP length: kích thước header+data

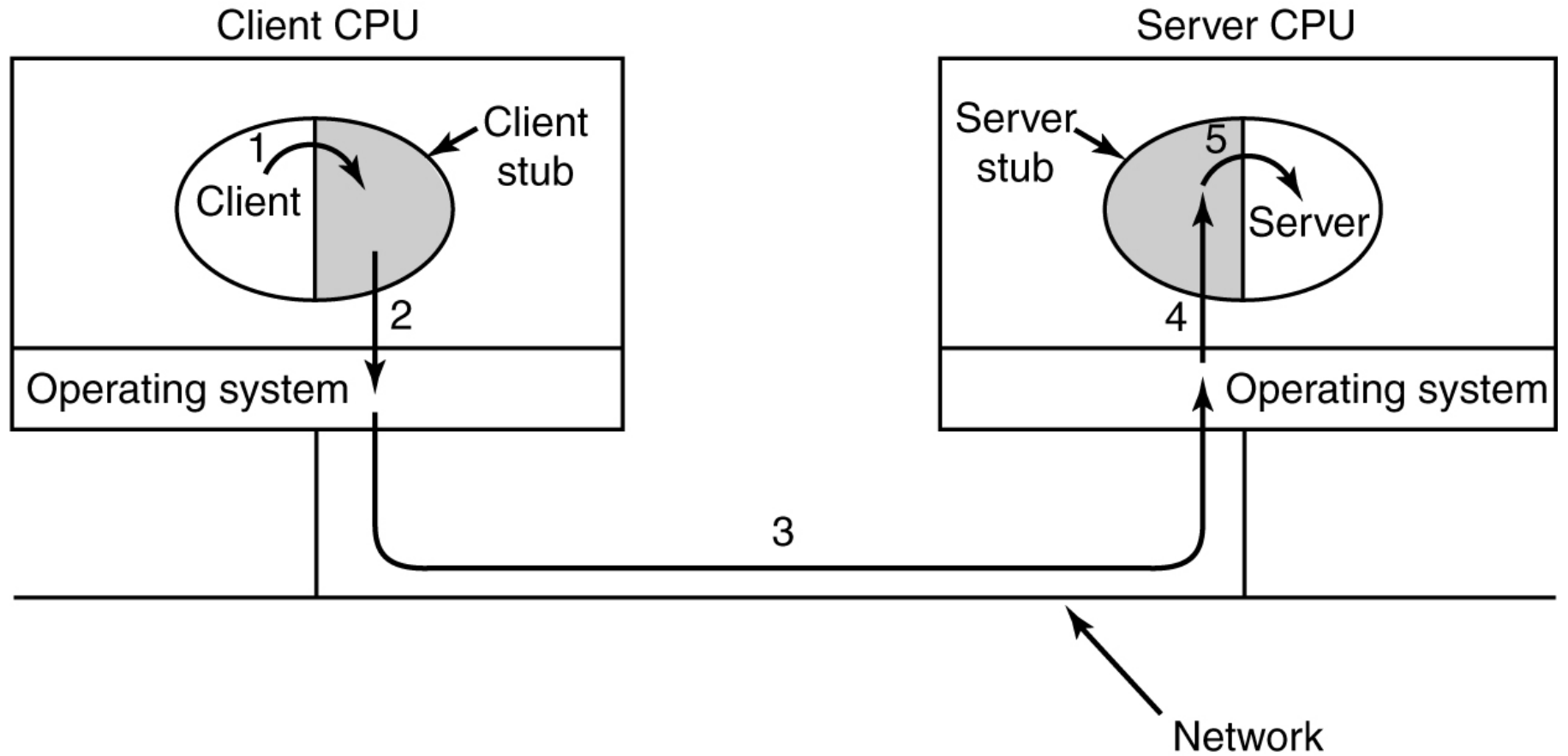
UDP checksum: phát hiện lỗi cho header+data

# Khái niệm RPC (Remote Procedure Call)

- Mô hình hoạt động thông dụng:
  - Một chương trình gửi thông điệp yêu cầu (request) đến chương trình trên máy khác
  - Chờ thông điệp trả lời (reply)
- Trừu tượng hoá mô hình request-reply thành dạng gọi thủ tục từ xa (RPC)
  - Client: nơi gọi thủ tục
  - Server: nơi thực thi thủ tục



# Mô hình RPC



## III. Giới thiệu giao diện lập trình mạng socket

1. Khái niệm Socket API
2. Giới thiệu Windows Sockets (WinSock)

# 1. Khái niệm Socket API

- API (Application Programming Interface)  
Giao diện lập trình ứng dụng: tập hợp các hàm cung cấp cho chương trình ứng dụng
- Socket APIs trừu tượng hoá việc truyền thông dạng client/server trên bộ giao thức TCP/IP với mô hình socket
- Socket API có thể sử dụng cho các bộ giao thức khác như IPX/SPX, DECNet, ...)

# Hai dạng Socket APIs

- Berkeley Sockets (BSD Sockets)

cung cấp các thao tác cơ sở (primitives)  
dùng trên UNIX

- Windows Sockets (WinSock)

Có các mở rộng hỗ trợ cơ chế message-driven của Windows

# Ví dụ

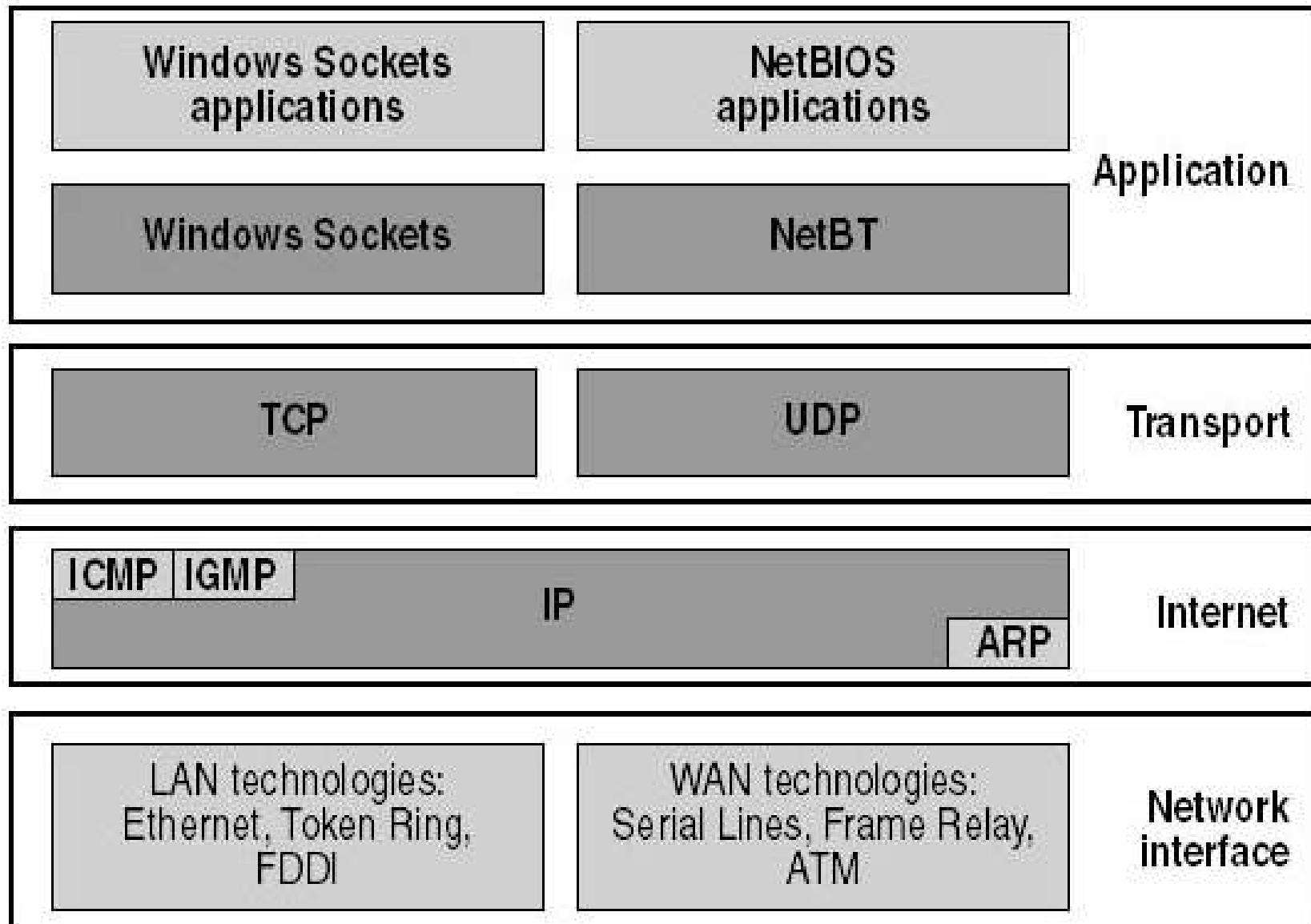
Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Các thao tác cơ sở trên TCP của BSD sockets

## 2. Giới thiệu WinSock

- WinSock: giao diện lập trình mạng dùng trên hệ điều hành Windows trên mô hình socket
- Chương trình sử dụng WinSock API, liên kết với thư viện WinSock

# Kiến trúc TCP/IP trên Microsoft Windows



# Dịch vụ WinSock

## ■ Các thao tác cơ sở

- Liên kết chương trình ứng dụng với socket
- Khởi tạo, chấp nhận kết nối
- Gửi nhận dữ liệu
- Đóng kết nối

## ■ Các hàm bất đồng bộ

## ■ Các hàm chuyển đổi dữ liệu



# Các dạng socket

## ■ Stream socket

- Trao đổi dữ liệu tin cậy 2 chiều dùng TCP

## ■ Datagram socket

- Trao đổi dữ liệu 2 chiều dùng UDP

Socket được định nghĩa theo:

## ■ Giao thức sử dụng

## ■ Địa chỉ

# NHẬP MÔN MẠNG MÁY TÍNH

## Chương 6

### LỚP APPLICATION (LỚP ỨNG DỤNG)



# Nội dung chương 6

- I. Giới thiệu
- II. Domain Name System (DNS)
- III. Telnet
- IV. File Transfer Protocol (FTP)
- V. E-Mail
- VI. World Wide Web (WWW)

# I. Giới thiệu

- Chương trình ứng dụng thực hiện các dịch vụ mạng
- Dịch vụ được đặc tả bởi giao thức
- Các dịch vụ chuẩn trên mạng TCP/IP:
  - DNS
  - FTP
  - SMTP
  - HTTP
  - .....



## II. Domain Name System (DNS)

1. Giới thiệu DNS
2. Không gian tên DNS
3. Dữ liệu DNS
4. Name servers

# 1. Giới thiệu DNS

- DNS là sơ đồ đặt tên:
  - Dạng text
  - Có thứ bậc
  - Cơ sở dữ liệu tên được quản lý phân bố
- Dùng để ánh xạ tên máy với địa chỉ IP, có thể dùng cho mục đích khác
- Được định nghĩa trong RFC 1034, 1035

## Hoạt động dạng đơn giản

Chương trình ứng dụng cần địa chỉ IP của một tên máy:

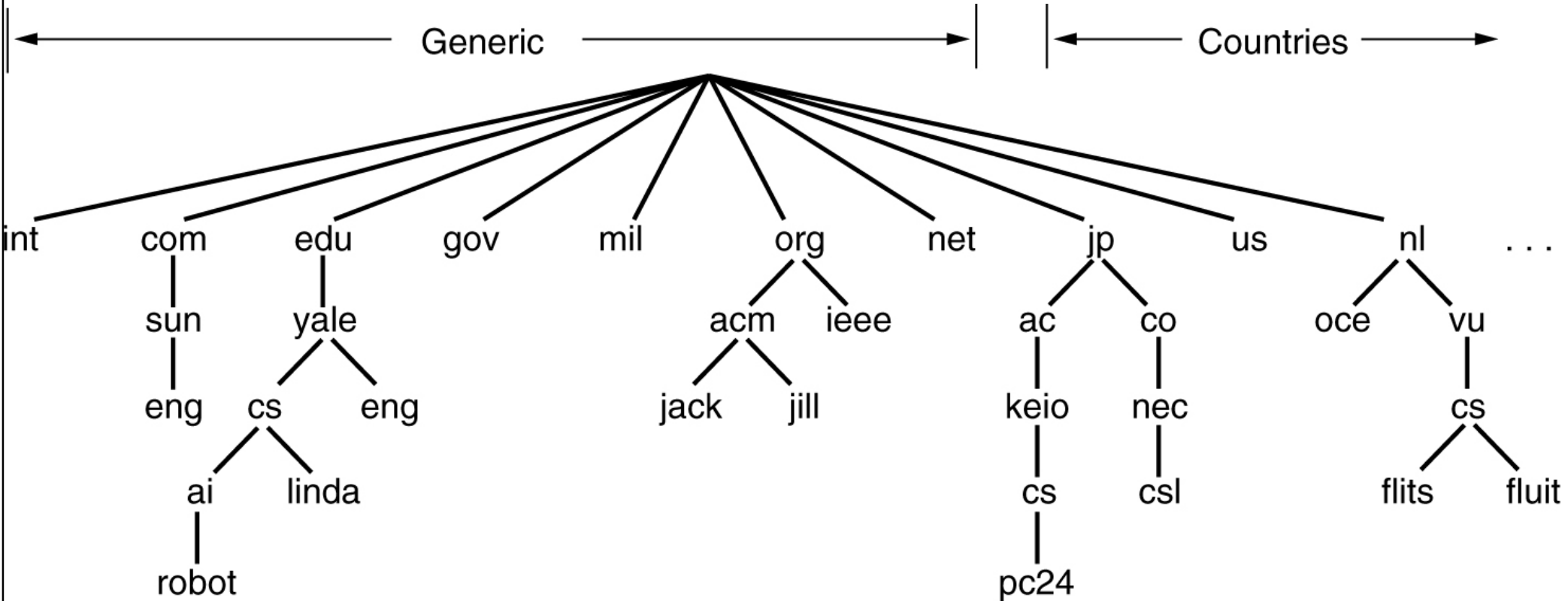
- Gọi hàm thư viện resolver (DNS client), tham số là tên máy
- Resolver gửi yêu cầu đến DNS server
- DNS server trả địa chỉ IP cho resolver
- Resolver trả địa chỉ IP cho chương trình ứng dụng

## 2. Không gian tên DNS

- Cấu trúc cây
- Có các top-level domain
- Trong top-level domain chia thành các subdomain
- Trong subdomain có thể chia thành các domain cấp thấp hơn



# Một phần không gian tên DNS



Ví dụ:            eng.sun.com  
                  robot.ai.cs.yale.edu  
                  www.vnn.vn

# Top-level domain (tên miền cấp 1)

Gồm 2 phần:

- Tên miền quốc gia

(Country code top-level domains)

- Theo ISO 3166
- Ví dụ: .vn, .fr, ...

- Tên miền chung

(Generic top-level domains)

- Do ICANN/IANA quy định

(Internet Assigned Numbers Authority)



# Tên miền chung

- com (commercial)
- edu (educational institutions)
- gov (US government)
- int (international organizations)
- mil (US armed forces)
- net (network providers)
- org (nonprofit organizations)

## Tên miền chung (tt)

- biz (businesses)
- info (information)
- name (people's name)
- pro (professions)

## Tên miền dành riêng

- aero (aerospace industry)
- coop (co-operatives)
- museum (museums)

### 3. Dữ liệu DNS

- Bao gồm các mẫu tin (resource record)
- Cấu trúc mẫu tin: có 5 trường
  - Domain\_name: tên miền
  - Time\_to\_live: thời gian ổn định của mẫu tin
  - Class: luôn là IN (Internet)
  - Type: loại mẫu tin
  - Value: giá trị

## Dữ liệu DNS (tt)

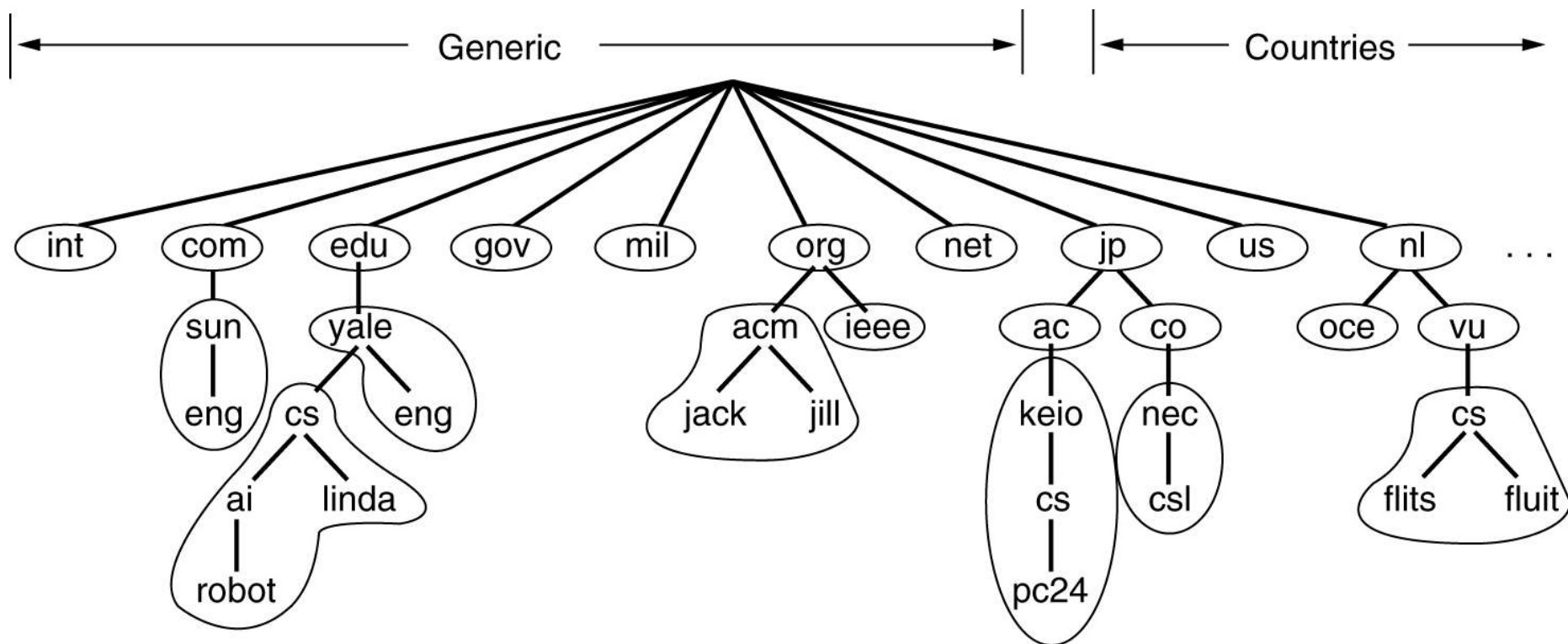
Loại (Type)	Ý nghĩa	Giá trị (Value)
SOA	Start of Authority	Các thông số của vùng
NS	Name Server	Tên của Name Server
A	IP address	Số nguyên 32 bit

Một số loại mẫu tin dữ liệu DNS

## 4. Name servers

- Không gian tên DNS được chia thành các vùng (zones) rời nhau
- Mỗi vùng được quản lý bởi các name server:
  - Primary name server
  - Các secondary name server

# Ví dụ các vùng



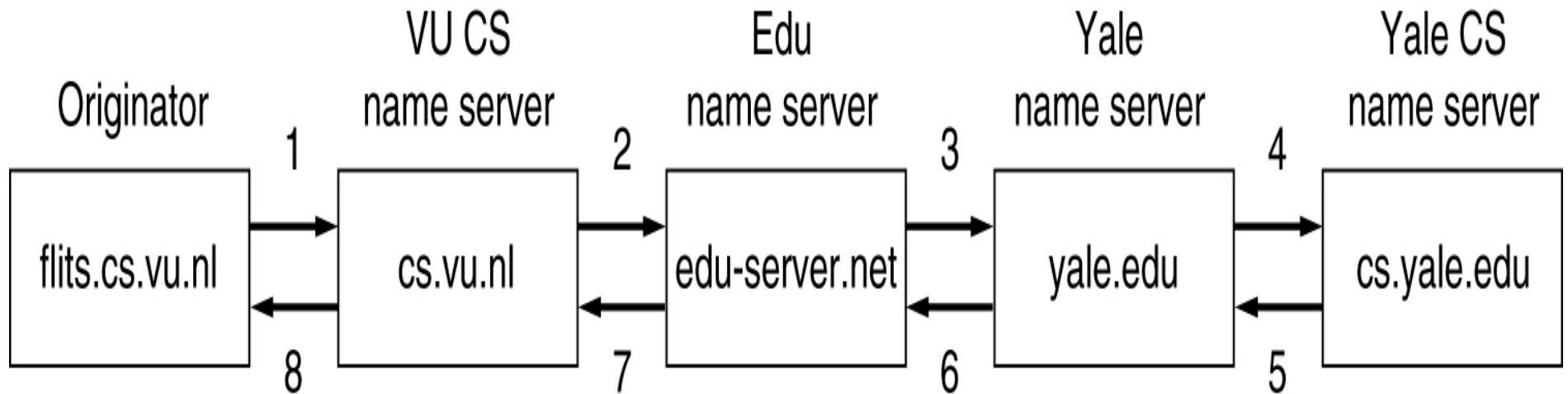


# Hoạt động của DNS

Resolver cần địa chỉ IP của một tên máy:

- Resolver gửi yêu cầu đến local name server
- Nếu có thông tin, local name server cung cấp mẫu tin cho resolver
- Nếu không có thông tin, local name server gửi yêu cầu đến top-level name server tương ứng, để có thông tin từ name server lưu mẫu tin cần tìm

# Ví dụ



Máy `flits.cs.vu.nl` cần địa chỉ IP của máy `linda.cs.yale.edu`

## III. Telnet

- Là ứng dụng chuẩn dạng có kết nối trên mạng TCP/IP
- Cho phép Telnet client (local host) đăng nhập vào Telnet server (remote host) tại port 23 và thực thi các lệnh trên dòng lệnh
  - sử dụng Network Virtual Terminal (NVT)
  - client system có thể khác server system
- Được định nghĩa trong RFC 854, 855

# Hoạt động telnet

- Truyền các phím ấn từ local host đến remote host
- Xử lý trên remote host
- Truyền màn hình kết quả cho local host



# Các dạng tương tự telnet

- VNC (Virtual Network Computer)
- Terminal Service
- Remote shell (RSH)
- Remote execution (REXEC)



## IV. File Transfer Protocol (FTP)

1. Khái niệm
2. Mô hình FTP

# 1. Khái niệm

- FTP là dịch vụ cho phép FTP client kết nối với FTP server để truyền và quản lý file
- Các tính chất:
  - Truy xuất dạng tương tác
  - Có 2 chế độ truyền: nhị phân và văn bản
  - Client phải cung cấp username, password  
anonymous user: không cần password
- Được định nghĩa trong RFC 959

# FTP Client

## ■ Có 2 dạng:

- Văn bản: dùng các lệnh FTP tại dòng lệnh  
ftp, get, close, quit, ...
- Đồ hoạ: thao tác trên file như chương trình quản lý file





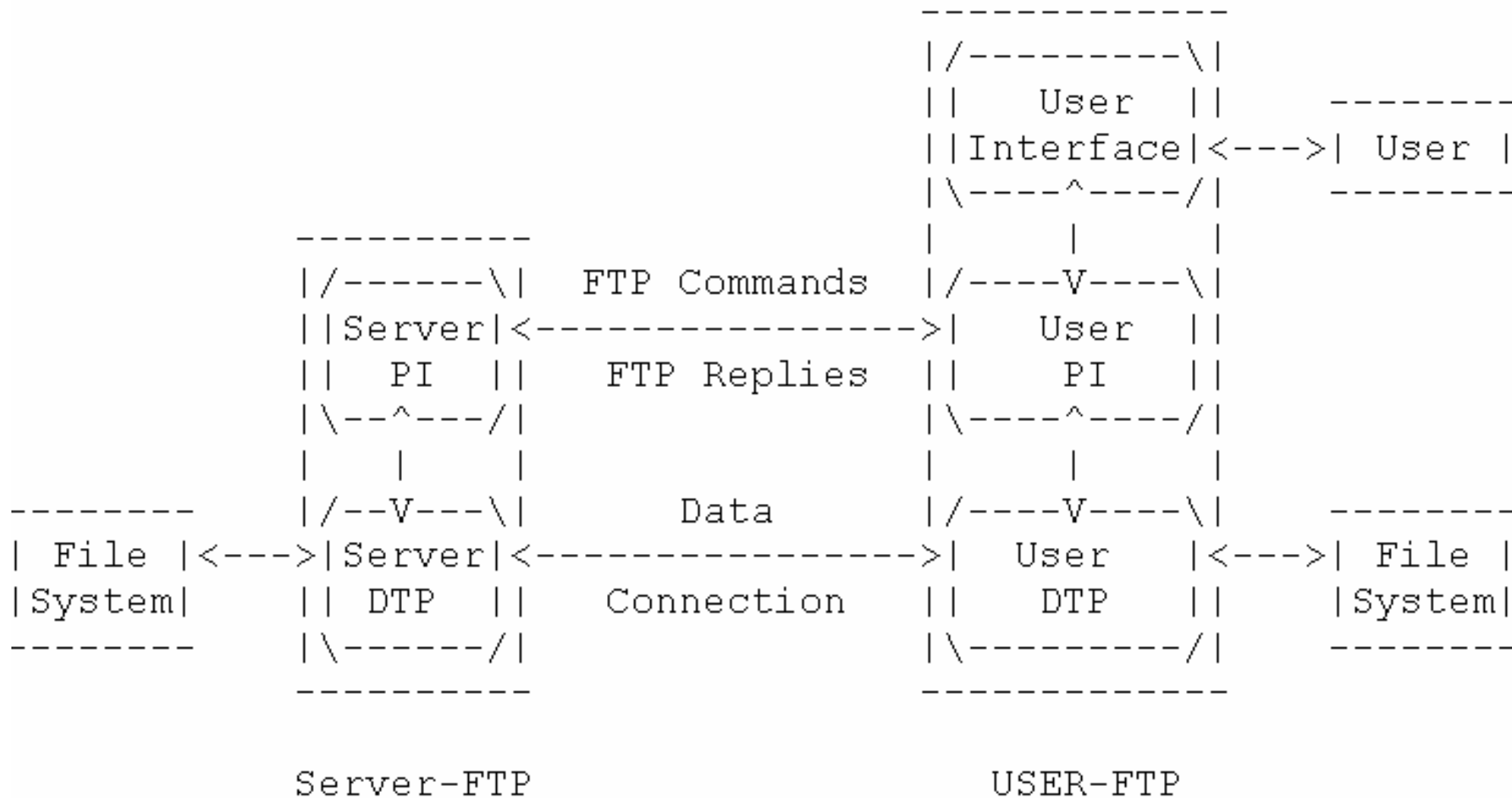
# TFTP (Trivial FTP)

- Dạng không kết nối (dùng UDP)
- Tốc độ cao hơn FTP
- Không tin cậy
- Ít chức năng hơn FTP

# Các dịch vụ tương tự FTP

- Web Browser có thể thực hiện các chức năng của FTP Client
- Gopher
  - Truyền file dạng phân bố
  - Giao diện menu
  - Kết hợp với các dịch vụ tìm kiếm

## 2. Mô hình FTP



PI: Protocol Interface, DTP: Data Transfer Process

# Mô hình FTP (tt)

Gồm 2 loại kết nối:

## ■ FTP control

- Server port 21, client port ( $>1023$ )
- Được thiết lập và duy trì trong phiên làm việc FTP

## ■ FTP data

- Server port 20, client port như FTP control
- Được thiết lập khi có truyền file, và kết thúc tự động



## V. Electronic Mail

1. Khái niệm
2. Kiến trúc hệ thống mail
3. Khuôn dạng mail
4. Các giao thức truyền mail
5. Webmail

# 1. Khái niệm

## ■ Hệ thống mail:

- Cho phép gửi nhận thông tin bất đồng bộ giữa hai người hay hai nhóm người
- Cung cấp phương tiện tạo, truyền, xử lý các thông tin

## ■ Có nhiều hệ thống mail

## ■ Internet mail:

- Khuôn dạng mail theo RFC 822, 2822
- Giao thức truyền mail theo RFC 821, 2821

## 2. Kiến trúc hệ thống mail

Gồm 2 thành phần

### ■ User Agents – UA

- Chương trình địa phương phía user  
(Local program)
- Cung cấp các phương tiện tương tác với hệ thống mail

### ■ Message Transfer Agents –MTA

- Chương trình thường trú phía server  
(System daemon)
- Thực hiện việc truyền mail

# Các chức năng cơ bản của hệ thống mail

- Tạo mail – Composition
- Truyền mail – Transfer
- Thông báo kết quả cho người gửi – Reporting
- Thông báo trạng thái cho người nhận – Displaying
- Xử lý mail đã nhận – Disposition



# Tổ chức hệ thống mail

Có các mailbox:

- Inbox (Incoming mailbox)
- Outbox (Outgoing mailbox)

Các dạng mailbox đặc biệt:

- Bulk
  - Chứa thư rác – spam/junk e-mail
- Sent
- ...

# Tổ chức hệ thống mail (tt)

Hỗ trợ về địa chỉ mail:

- Mailing list
- Address book

Địa chỉ mail – Mail account

- local-part@domain
- mailbox@host

## 3. Khuôn dạng mail

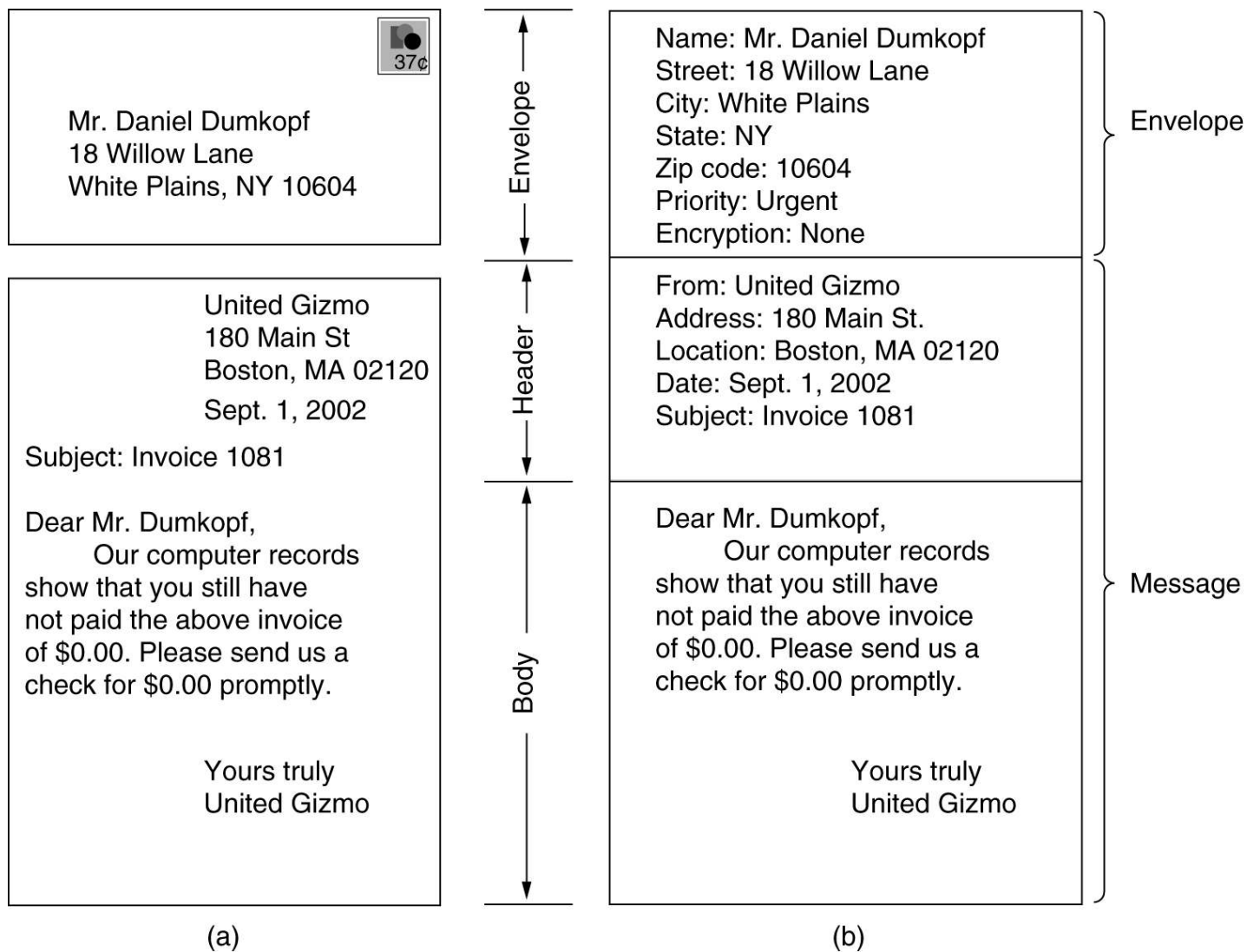
Gồm các phần:

- Envelope – Bao thư/phong bì
- Message – Thông điệp
  - Header: các thông tin điều khiển
  - Body: nội dung

Tiêu chuẩn khuôn dạng mail

- Internet message format - RFC 822/2822
- MIME – RFC 2045-2049
  - Multipurpose Internet Mail Extensions

# Ví dụ



a. Thư trên giấy

b. Thư điện tử

# Khuôn dạng mail theo RFC 822

- Không phân biệt phần envelope và phần header, gọi chung là header
- Phần body là tùy ý

# Các thành phần chính trên RFC 822 header

To:	Địa chỉ mail các người nhận chính
Cc:	Carbon copy Địa chỉ mail các người nhận phụ
Bcc:	Blind carbon copy Địa chỉ mail các người nhận ẩn
From:	Tên người tạo mail
Sender:	Địa chỉ mail người gửi
Subject	Nội dung tóm tắt
.....	.....

# MIME

- Mở rộng khuôn dạng thông điệp theo RFC 822:
  - Nội dung thông điệp với các bộ ký tự khác ASCII
  - Nội dung thông điệp không là ký tự (hình ảnh, âm thanh, ...)
  - Thông điệp có nhiều phần (multi-part)
  - Phần header với ký tự khác ASCII

# Mở rộng phần header

Header	Ý nghĩa
MIME-Version	Phiên bản MIME
Content-Description	Mô tả nội dung
Content-ID	Số thứ tự
Content-Transfer-Encoding	Dạng mã hoá của nội dung
Content-Type	Loại và khuôn dạng của nội dung



# Một số loại dữ liệu theo MIME

Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	Rfc822	A MIME RFC 822 message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message

# Ví dụ

From: elinor@abcd.com  
To: carolyn@xyz.com  
MIME-Version: 1.0  
Message-Id: <0704760941.AA00747@abcd.com>  
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm  
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm  
Content-Type: text/enriched

Happy birthday to you  
Happy birthday to you  
Happy birthday dear <bold> Carolyn </bold>  
Happy birthday to you

--qwertyuiopasdfghjklzxcvbnm  
Content-Type: message/external-body;  
    access-type="anon-ftp";  
    site="bicycle.abcd.com";  
    directory="pub";  
    name="birthday.snd"

content-type: audio/basic  
content-transfer-encoding: base64  
--qwertyuiopasdfghjklzxcvbnm--

## 4. Các giao thức truyền mail

- SMTP - Simple Mail Transfer Protocol

MTA → MTA, UA → MTA

- POP3 - Post Office Protocol version 3

MTA → UA

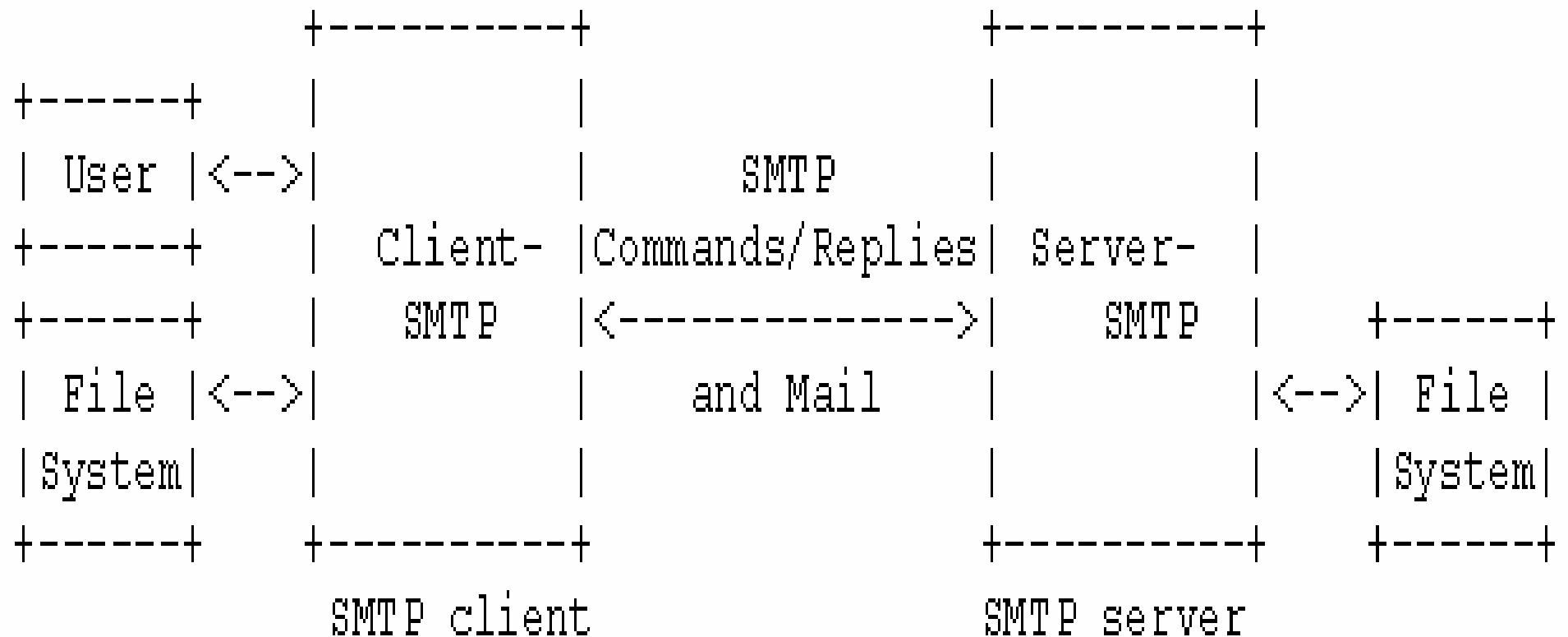
- IMAP-Internet Message Access Protocol

MTA → UA

# SMTP

- Được định nghĩa trong RFC 821, 2821
- Dạng client-server
- SMTP client thiết lập kết nối TCP với SMTP server tại port 25
- Nếu SMTP server đồng ý nhận mail:
  - SMTP client gửi địa chỉ người gửi, người nhận
  - SMTP client gửi mail
  - SMTP server gửi ACK
  - Hủy kết nối

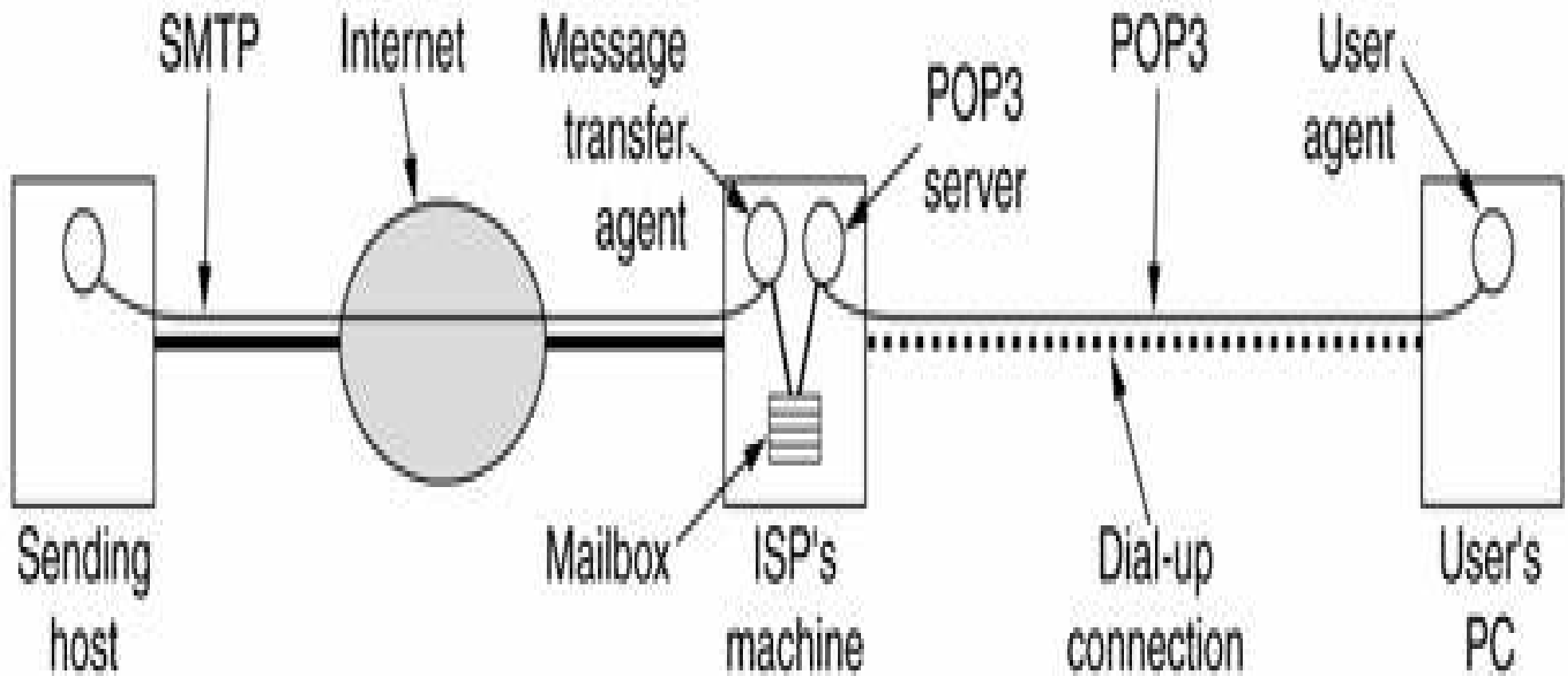
# Mô hình SMTP



# POP3

- Được định nghĩa trong RFC 1939, 2449
- Dùng lấy mail từ remote mailbox về máy địa phương
- Client thiết lập kết nối TCP với server tại port 110

# Mô hình POP3



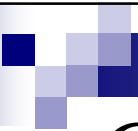
# Các giai đoạn hoạt động POP3

- Authorization – Cho phép
  - Client gửi username, password
- Transaction – Giao dịch
  - Client yêu cầu nội dung mail và xoá mail tại server
  - Server gửi các mail
- Update – Cập nhật
  - Client gửi lệnh thoát (quit)
  - Server xoá các mail, hủy kết nối



# IMAP

- Được định nghĩa trong RFC 2060
- Quản lý mail tập trung tại server, không di chuyển về máy địa phương như POP3
  - có thể truy xuất từ nhiều máy
- Client thiết lập kết nối TCP với server tại port 143



## Các đặc điểm của IMAP

- Cho phép tạo, xoá, xử lý nhiều mailbox tại server
- Có thể truy xuất từng phần của mail
- Có thể truy xuất mail theo thuộc tính

## 5. Webmail

- Web site cung cấp dịch vụ mail
- Có MTA tại port 25, nhận các kết nối SMTP
- User dùng các form trên trang web để tương tác với hệ thống:
  - Đăng nhập - Login
  - Liệt kê các mail box
  - Đọc, xoá, soạn thảo, ...mail



## VI. World Wide Web

1. Khái niệm
2. Kiến trúc hệ thống Web
3. Trang web tĩnh
4. Trang web động
5. Giao thức HTTP
6. Web không dây

# 1. Khái niệm

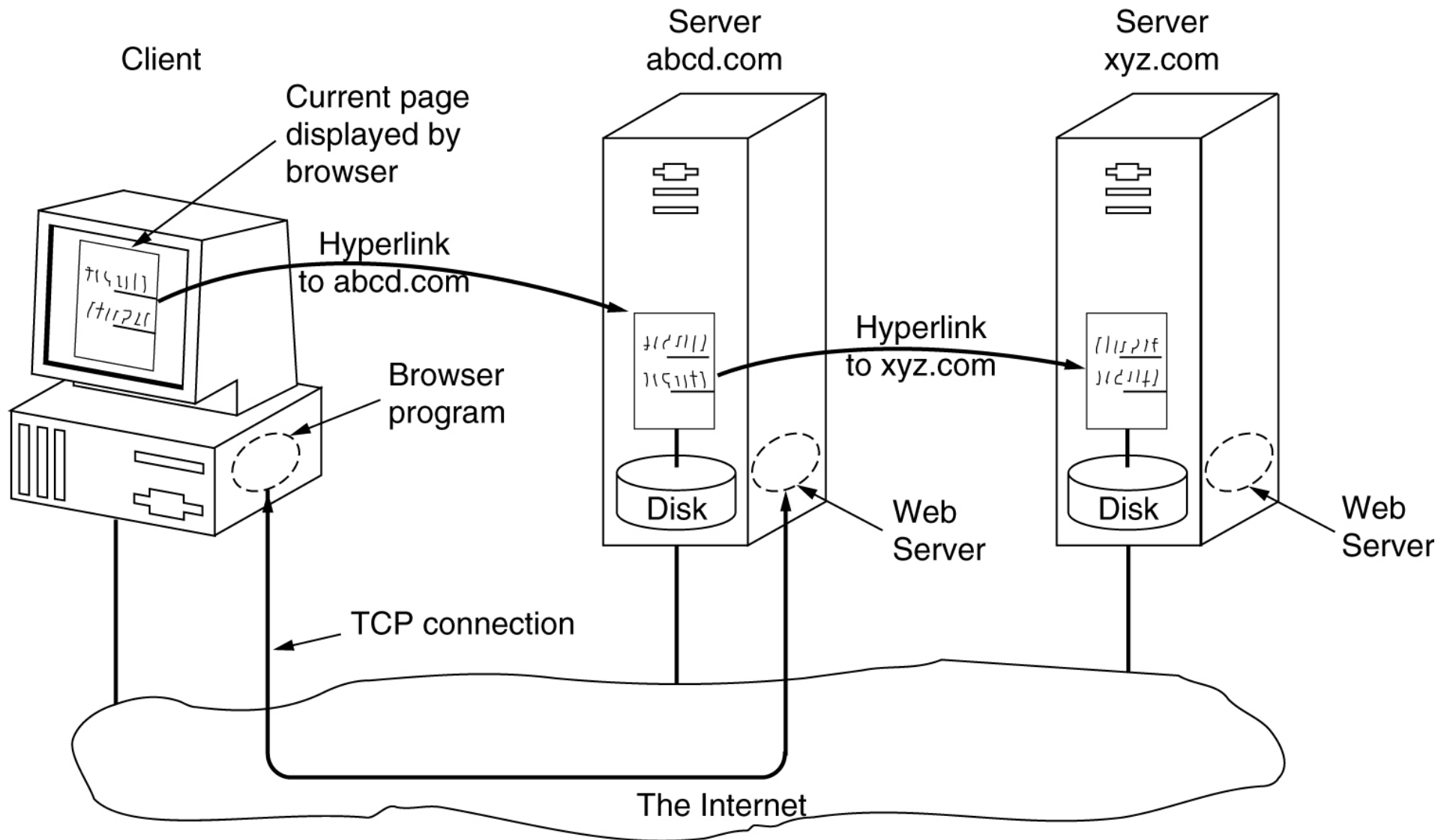
- Web là dịch vụ truy xuất các văn bản có liên kết, trang web, từ các máy trên mạng Internet
- Do Tim Berners-Lee thiết kế năm 1989 tại CERN (trung tâm nghiên cứu hạt nhân châu Âu)
- Năm 1994, CERN và MIT thành lập tổ chức World Wide Web Consortium ([www.w3c.org](http://www.w3c.org)) để phát triển Web



## 2. Kiến trúc hệ thống Web

- a. Hoạt động phía client
- b. Hoạt động phía server
- c. Tên trang web
- d. Cookies

# Mô hình dịch vụ Web



## a. Hoạt động phía Client

- Web browser: chương trình hiển thị các trang web phía client
- Hoạt động web browser:
  - Lấy trang web được yêu cầu
  - Thông dịch nội dung trang web
  - Hiển thị trên màn hình
- Tên trang web có dạng URL  
(Uniform Resource Locator)

Ví dụ: <http://www.itu.org/home/index.html>



# ■ Ví dụ: web browser lấy và hiển thị trang web <http://www.itu.org/home/index.html>

1. Browser xác định URL
2. Browser yêu cầu DNS cung cấp địa chỉ IP máy [www.itu.org](http://www.itu.org)
3. DNS trả lời 156.106.192.32
4. Browser thiết lập kết nối TCP port 80 với máy 156.106.192.32
5. Browser gửi yêu cầu file /home/index.html
6. Server [www.itu.org](http://www.itu.org) gửi file /home/index.html
7. Hủy kết nối TCP
8. Browser hiển thị phần text trong file index.html
9. Browser lấy và hiển thị các hình ảnh trong file (nếu có)

# Các chức năng của browser


- Duyệt các trang web: back, forward, history, favorites/bookmarks
- Lưu trang web thành file, in
- Cache các trang web trên đĩa địa phương  
→ hoạt động offline

# Plug-in

- Mở rộng khả năng của browser
- Đoạn chương trình lưu trong thư mục plug-in
- Được browser gọi khi cần hiển thị các loại dữ liệu không là html, ví dụ PDF

## b. Hoạt động phía server

- Web server chờ kết nối TCP tại port 80
- Hoạt động web server:
  - Chấp nhận kết nối từ client (web browser)
  - Nhận tên file được yêu cầu
  - Lấy file (từ đĩa)
  - Gởi file cho client
  - Hủy kết nối

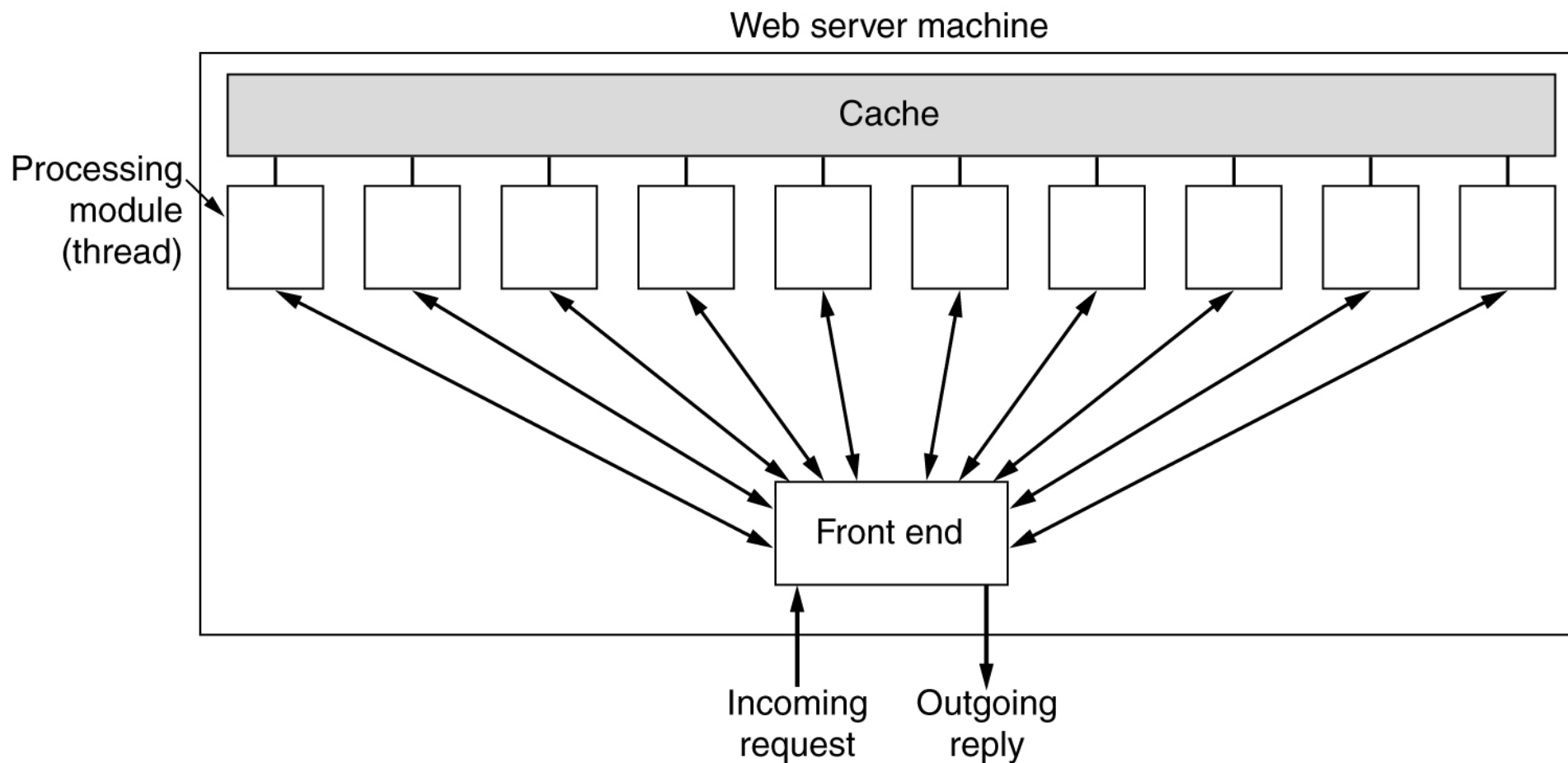


# Tăng tốc độ web server

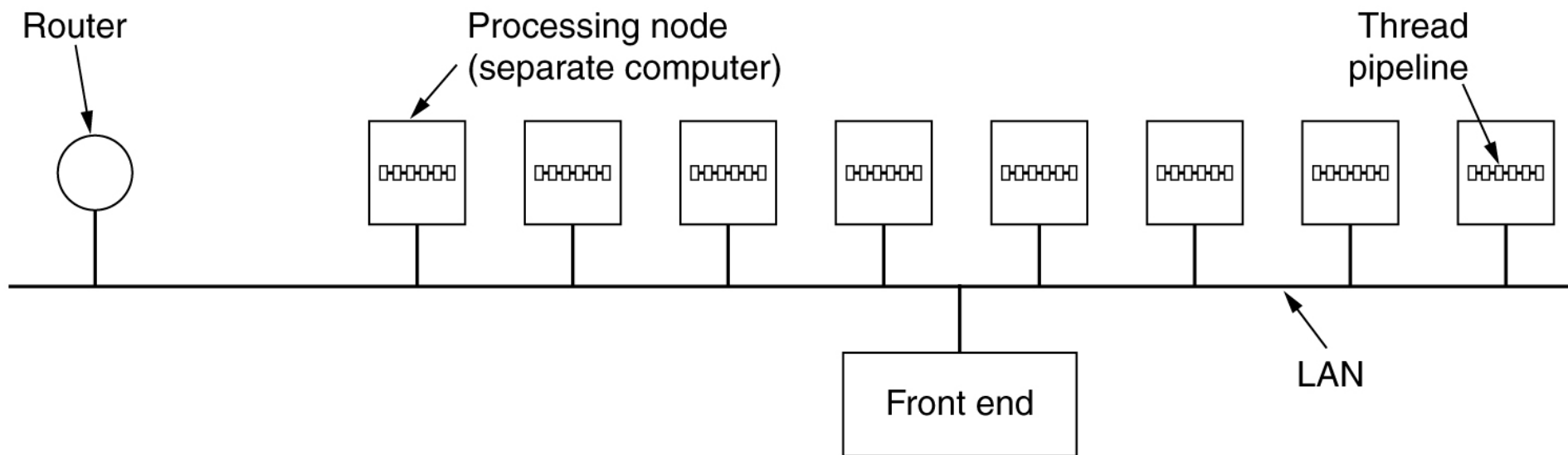
Hai kỹ thuật:

- Dùng cache và server đa luồng
- Dùng nhiều máy làm web server (server farm)

# Web server dạng đa luồng



# Nhiều máy làm web server



## c. Tên trang web

### ■ Theo URL (Uniform Resource Locator)

Tên\_giao\_thức://tên\_máy/tên\_file

- Tên\_file: tên file địa phương
- Tên\_máy: theo DNS
- Tên\_giao\_thức: có nhiều loại giao thức

### ■ Web browser có thể dùng cho nhiều dịch vụ với URL



# Thành phần tên\_giao\_thức trong URL

<b>Name</b>	<b>Used for</b>	<b>Example</b>
http	Hypertext (HTML)	<a href="http://www.cs.vu.nl/~ast/">http://www.cs.vu.nl/~ast/</a>
ftp	FTP	<a href="ftp://ftp.cs.vu.nl/pub/minix/README">ftp://ftp.cs.vu.nl/pub/minix/README</a>
file	Local file	<a href="file:///usr/suzanne/prog.c">file:///usr/suzanne/prog.c</a>
news	Newsgroup	<a href="news:comp.os.minix">news:comp.os.minix</a>
news	News article	<a href="news:AA0134223112@cs.utah.edu">news:AA0134223112@cs.utah.edu</a>
gopher	Gopher	<a href="gopher://gopher.tc.umn.edu/11/Libraries">gopher://gopher.tc.umn.edu/11/Libraries</a>
mailto	Sending e-mail	<a href="mailto:JohnUser@acm.org">mailto:JohnUser@acm.org</a>
telnet	Remote login	<a href="telnet://www.w3.org:80">telnet://www.w3.org:80</a>

## d. Cookies

- Chứa thông tin trạng thái của phiên làm việc giữa web server và web browser
- Là chuỗi ký tự lưu thành file tại máy client
- Khi gửi trang web cho client, server có thể gửi kèm cookies để lưu các thông tin trạng thái
- Khi gửi yêu cầu đến server, browser sẽ gửi kèm cookies (nếu có)

# Ví dụ cookies

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No



## 3. Trang web tĩnh

- a. HTML
- b. Forms
- c. XML

## a. HTML (HyperText Markup Language)

- Trang web được tạo theo ngôn ngữ HTML (ngôn ngữ đánh dấu siêu văn bản)
- Nội dung trang web có thể bao gồm:
  - Văn bản
  - Hình ảnh
  - Âm thanh, hình ảnh động
  - Các siêu liên kết (hyperlink)

# Đặc điểm HTML

- Là ứng dụng SGML  
(Standard Generalized Markup Language)
- Bao gồm các lệnh định dạng, gọi là tag  
Ví dụ: `<b> boldface </b>`
- Các browser có thể định dạng lại cho phù hợp với môi trường
- Có các tiêu chuẩn HTML 1.0, 2.0, ..., 4.0  
XHTML (eXtended HTML)

# Một số lệnh định dạng

Tag	Description
<code>&lt;html&gt; ... &lt;/html&gt;</code>	Declares the Web page to be written in HTML
<code>&lt;head&gt; ... &lt;/head&gt;</code>	Delimits the page's head
<code>&lt;title&gt; ... &lt;/title&gt;</code>	Defines the title (not displayed on the page)
<code>&lt;body&gt; ... &lt;/body&gt;</code>	Delimits the page's body
<code>&lt;h n&gt; ... &lt;/hn&gt;</code>	Delimits a level <i>n</i> heading
<code>&lt;b&gt; ... &lt;/b&gt;</code>	Set ... in boldface
<code>&lt;i&gt; ... &lt;/i&gt;</code>	Set ... in italics
<code>&lt;center&gt; ... &lt;/center&gt;</code>	Center ... on the page horizontally
<code>&lt;ul&gt; ... &lt;/ul&gt;</code>	Brackets an unordered (bulleted) list
<code>&lt;ol&gt; ... &lt;/ol&gt;</code>	Brackets a numbered list
<code>&lt;li&gt;</code>	Starts a list item (there is no <code>&lt;/li&gt;</code> )
<code>&lt;br&gt;</code>	Forces a line break here
<code>&lt;p&gt;</code>	Starts a paragraph
<code>&lt;hr&gt;</code>	Inserts a Horizontal rule
<code>&lt;img src="..."&gt;</code>	Displays an image here
<code>&lt;a href="..."&gt; ... &lt;/a&gt;</code>	Defines a hyperlink



## b. Forms

- Có từ HTML 2.0
- Form bao gồm các nút ấn (button), hộp (boxes) cho phép user nhập thông tin, lựa chọn
- Dữ liệu trên form được gửi lại server dưới dạng string



# Ví dụ form

## Widget Order Form

Name

Street address

City

State

Country

Credit card #

Expires

M/C

Visa

Widget size

Big

Little

Ship by express courier

Submit order

Thank you for ordering an AWI widget, the best widget money can buy!

## c. XML (eXtensible Markup Language)

### ■ Mục đích:

- Thể hiện cấu trúc trang web
- Mô tả thông tin
- Có thể dùng cho các loại ứng dụng khác

### ■ Cần cơ chế hiển thị thông tin XML trên browser dạng HTML, ví dụ XSL (eXtensible Style Language)

# Ví dụ văn bản XML

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="b5.xsl"?>
<book_list>
  <book>
    <title> Computer Networks, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2003 </year>
  </book>
  <book>
    <title> Modern Operating Systems, 2/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2001 </year>
  </book>
  <book>
    <title> Structured Computer Organization, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 1999 </year>
  </book>
</book_list>
```

## 4. Trang web động

### ■ Trang web tĩnh:

- Client gửi yêu cầu là tên file
- Server gửi file đã có

### ■ Trang web động

- Nội dung trang web được tạo theo yêu cầu, thay vì đã có trên đĩa

### ■ Có 2 dạng:

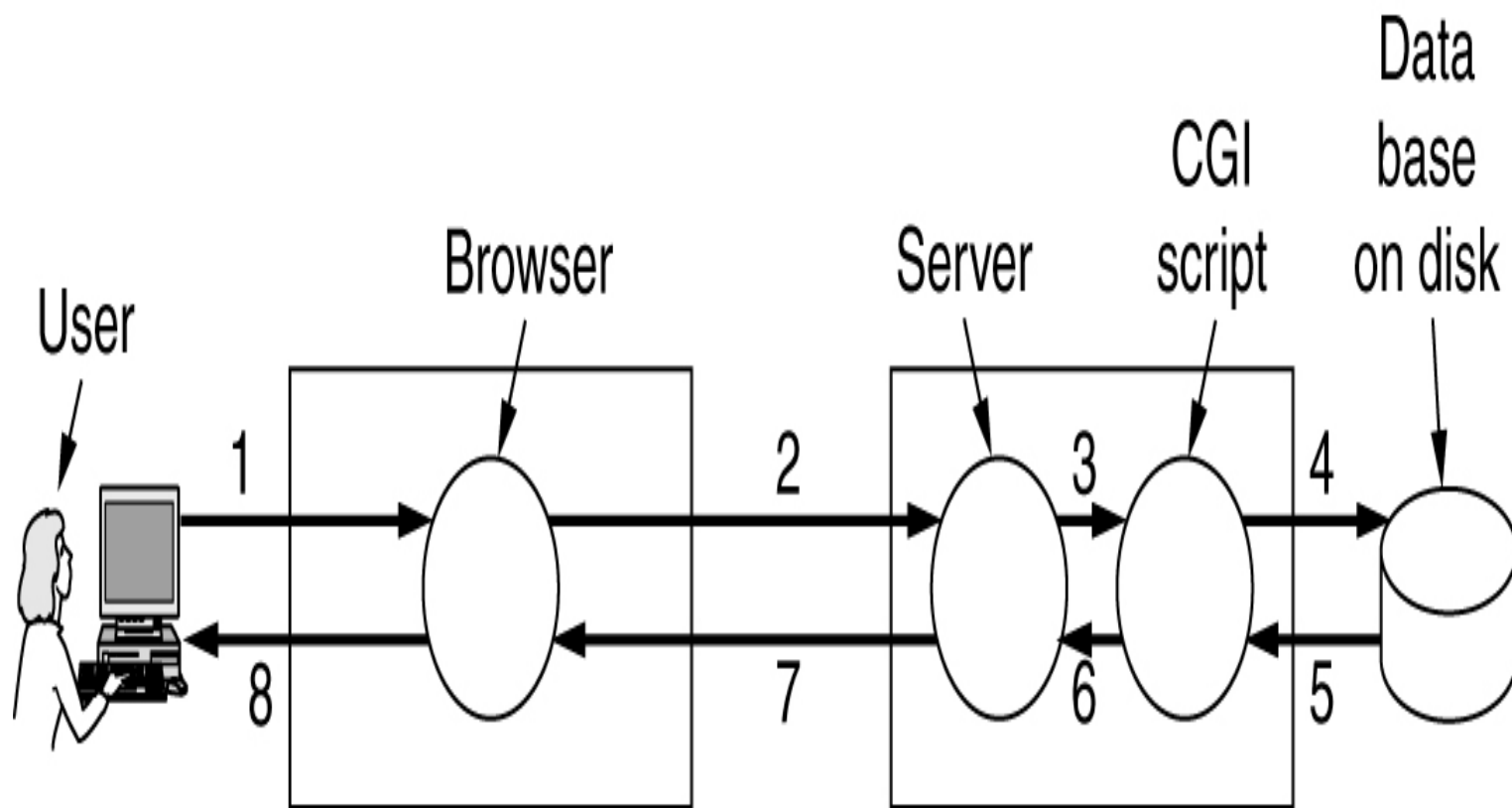
- Tạo trang web động tại server
- Tạo trang web động tại client

# Tạo trang web động tại server

Có các dạng:

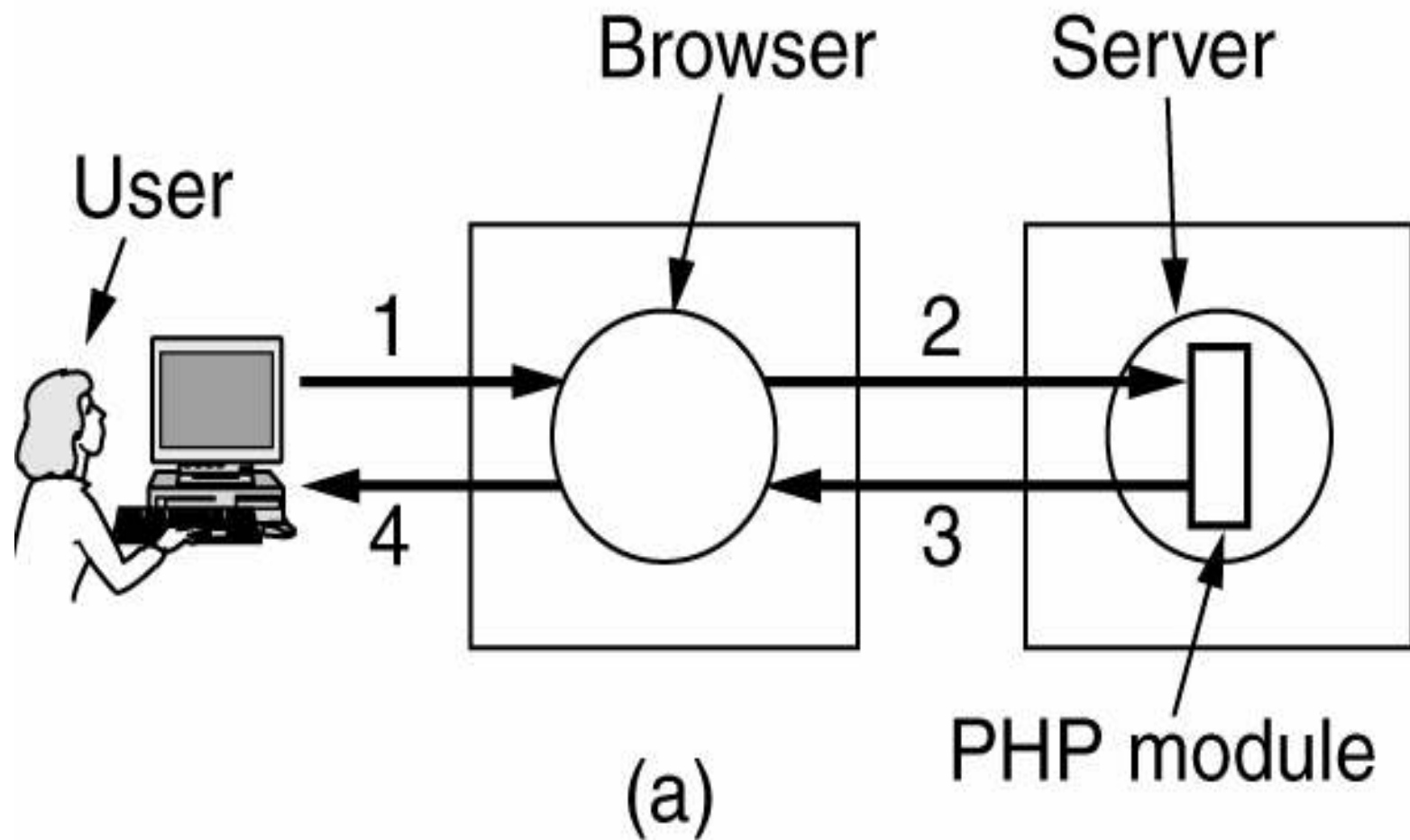
- CGI (Common Gateway Interface)  
với các script, ví dụ Perl, Python, ...
- Dùng các dạng script trong trang web  
(HTML-embedded scripting language)
  - PHP (PHP: Hypertext Preprocessor)
  - JSP (JavaServer Pages)
  - ASP (Active Server Pages)

# Ví dụ: các bước xử lý form dùng CGI



1. User fills in form
2. Form sent back
3. Handed to CGI
4. CGI queries DB
5. Record found
6. CGI builds page
7. Page returned
8. Page displayed

# Ví dụ: tạo trang web động với PHP



# Tạo trang web động tại client

- Dùng các script trong trang web, thực hiện tại máy client để tương tác trực tiếp với user
- Các công nghệ thông dụng
  - Javascript: client-side scripting language
  - JavaApplets
  - Microsoft ActiveX control



## ■ 5. Giao thức HTTP

### (HyperText Transfer Protocol)

- Được định nghĩa trong RFC 2616
- Quy định các dạng thông điệp trao đổi giữa web browser và web server
- Mỗi tương tác bao gồm:
  - Yêu cầu từ browser dạng ASCII
  - Đáp ứng từ server dạng tương tự MIME
- Yêu cầu (request) còn gọi là lệnh (command, method) và có đáp ứng (response)

# Một số dạng yêu cầu HTTP

<b>Method</b>	<b>Description</b>
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

# Đáp ứng HTTP

Bao gồm:

- Dòng trạng thái
- Thông tin (1 phần hay toàn bộ trang web)

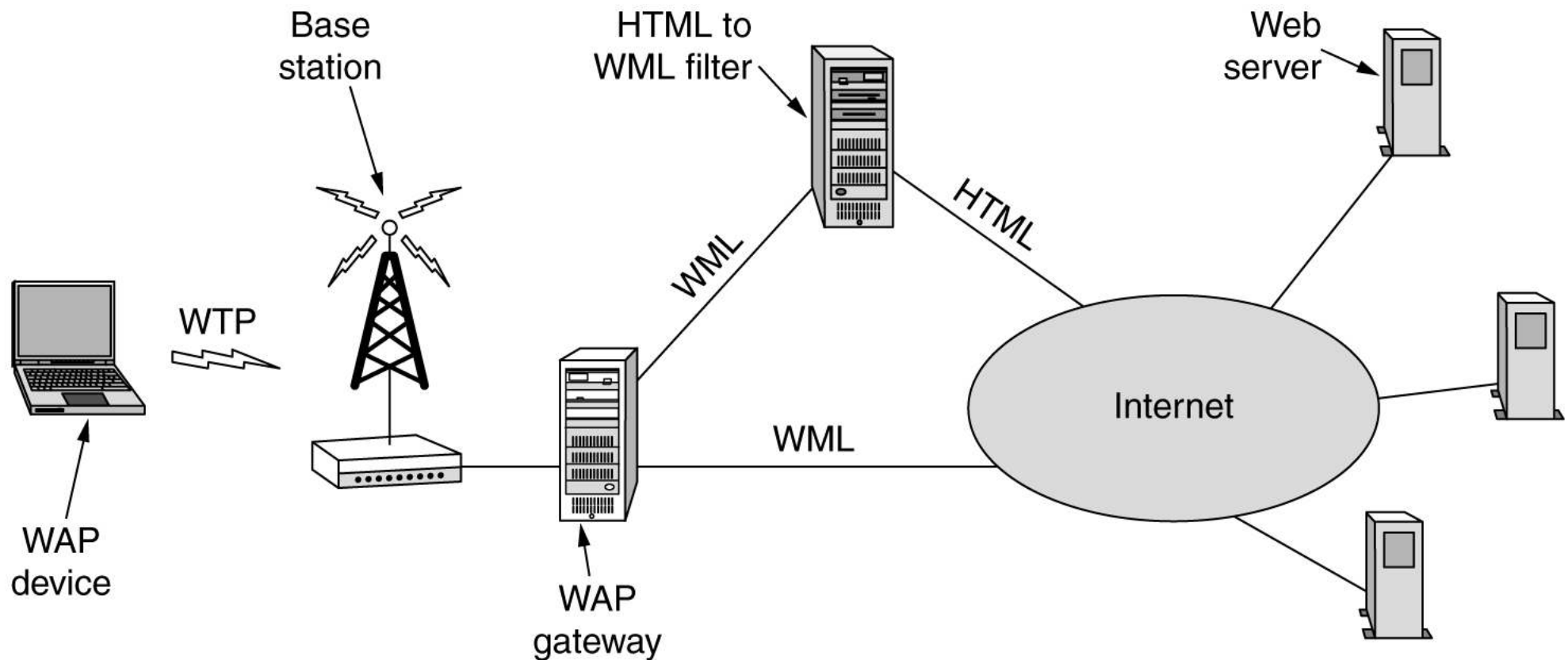
Một số mã trạng thái:

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

## 6. Web không dây (Wireless web)

- Cung cấp dịch vụ truy cập web dạng không dây cho điện thoại di động, PDA (Personal Digital Assistant), máy tính xách tay
- Đặc điểm:
  - Tốc độ truyền thấp
  - Bộ nhớ ít
  - Màn hình kích thước nhỏ
- Tiêu chuẩn thông dụng
  - WAP (Wireless Application Protocol)

# Kiến trúc hệ thống WAP



WTP: Wireless Transaction Protocol

WML: Wireless Markup Language

# Bộ giao thức WAP 2.0

XHTML	
WSP	HTTP
WTP	TLS
WTLS	TCP
WDP	IP
Bearer layer	Bearer layer

WAP 1.0 protocol  
stack

WAP 2.0 protocol  
stack

WSP: Wireless Session Protocol

WTP: Wireless Transaction Protocol

WTLS: Wireless Transport Layer Security

WDP: Wireless Datagram Protocol

# **NHẬP MÔN MẠNG MÁY TÍNH**

## **Chương 7**

**GIỚI THIỆU QUẢN TRỊ MẠNG**

**VÀ**

**AN TOÀN THÔNG TIN MẠNG**



# Nội dung chương 7

- I. Giới thiệu về quản trị mạng
- II. Giới thiệu về an toàn thông tin mạng





# I. Giới thiệu về quản trị mạng

1. Khái niệm
2. Các bước thiết lập mạng cục bộ
3. Ví dụ Windows 2000 Server

# 1. Khái niệm

- Các loại mạng: WAN, LAN
- Các loại LAN:
  - Peer-to-peer
  - Server-based
  - Dạng tổ hợp
- Quản trị mạng có tính chất động:
  - Quy mô mạng thay đổi
  - Công dụng mạng thay đổi



# Công dụng mạng

- Chia sẻ tài nguyên
- Truy xuất có kiểm soát tài nguyên
- Môi trường truyền thông
- Quản lý các hệ thống máy tính tốt hơn



## 2. Các bước thiết lập mạng cục bộ

- a. Lập kế hoạch
- b. Hiện thực mạng
- c. Quản trị mạng

## a. Lập kế hoạch

Các bước lập kế hoạch:

- Thu thập dữ liệu cần thiết
- Khảo sát các khả năng hiện thực
- Chọn giải pháp tốt nhất về giá cả và hiệu suất



# Thông số mạng LAN

- Loại mạng
- Kiến trúc mạng
- Môi trường truyền vật lý
- Giao thức mạng
- Phần mềm mạng
- An toàn dữ liệu



## b. Hiện thực mạng

- Cài đặt
- Kiểm tra
- Tập huấn, đào tạo

# Cài đặt

- Cài đặt phần cứng
- Cài đặt hệ điều hành mạng
  - Hệ điều hành mạng độc lập
  - Phần mềm mạng thêm vào hệ điều hành đã có
- Cài đặt các dịch vụ mạng
- Cài đặt các ứng dụng
  - Ứng dụng mạng dạng multiuser
  - Ứng dụng dùng chung



# Kiểm tra

Kiểm tra các thành phần bằng cách cô lập và kiểm tra:

- Các máy tính server
- Các máy tính Client/Workstation
- Các thiết bị ngoại vi
- Môi trường truyền vật lý
- Phần mềm client, phần mềm server

# Tập huấn, đào tạo

Mục đích:

- Sử dụng mạng hiệu quả
- Hoạt động ổn định

Đối tượng tập huấn, đào tạo:

- Administrators – Người quản trị
- Users – Người sử dụng

## c. Quản trị mạng

- Quản trị user
  - Tạo và duy trì các tài khoản user
- Quản lý tài nguyên
  - Hiện thực, hỗ trợ sử dụng tài nguyên
- Quản lý cấu hình
  - Bảo trì, mở rộng thông tin cấu hình
- Quản trị hiệu suất
  - Kiểm tra hoạt động mạng, tăng hiệu suất
- Bảo trì
  - Ngăn chặn, phát hiện, giải quyết lỗi



## 3. Ví dụ Windows 2003 Server

- a. Các dạng Windows 2003
- b. Cài đặt Windows 2003 Server
- c. Quản lý users, groups
- d. Thiết lập cấu hình các giao thức mạng
- e. Thiết lập cấu hình các dịch vụ ứng dụng



## a. Các dạng Windows 2003 Server

- Windows Server 2003 Standard Edition
- Windows Server 2003 Enterprise Edition
- Windows Server 2003 Datacenter Edition
- Windows Server 2003 Web Edition



# Windows Server 2003 Standard Edition

- Dùng cho doanh nghiệp nhỏ, trung bình
- Hỗ trợ đến 4 CPU, 4GB RAM
- Có thể là File/Print/Web/Application Servers
- Có thể là Domain Controller

# Windows Server 2003 Enterprise Edition

- Có các khả năng như bản Standard
- Hỗ trợ đến 8 CPU, 32GB RAM – 32 bit  
Hỗ trợ đến 8 CPU, 64GB RAM – 64 bit
- Hỗ trợ ghép cụm (clustering) 8 nodes

# Windows Server 2003 Datacenter Edition

- Phát hành với phần cứng
- Có các khả năng như bản Enterprise
- Hỗ trợ đến 32 CPU, 64GB RAM – 32 bit  
Hỗ trợ đến 32 CPU, 512GB RAM – 64 bit



# Windows Server 2003 Web Edition

- Sử dụng cho:
  - Web applications
  - Web pages
  - XML services
- Hỗ trợ ASP.NET, .NET framework
- Hỗ trợ đến 2 CPU, 2GB RAM
- Không có một số chức năng như Domain Controller, Remote Installation Service, DNS services, ...

# Vai trò của Windows 2003 Server

- Member Server: thành viên của domain, không lưu trữ thông tin directory
- Domain Controller: lưu trữ thông tin domain, cung cấp dịch vụ kiểm chứng
- Stand-alone Server: không tham gia domain, tự kiểm chứng các yêu cầu đăng nhập

## b. Cài đặt Windows 2003 Server

Các yêu cầu trước khi cài đặt:

- Yêu cầu về phần cứng, tương thích phần cứng
- Phân vùng đĩa, chọn hệ thống file
- Bản quyền CAL (Client Access License)
  - Per-server: CAL cấp cho server
  - Per-seat: mỗi máy truy xuất Server có CAL
- Chọn vai trò của server



## Các dạng cài đặt

- Cài đặt từ các đĩa mềm khởi động
- Cài đặt từ CDROM có thể khởi động
- Cài đặt từ đĩa cứng, từ mạng

## c. Quản lý users, groups

- Sử dụng các công cụ Active Directory
- User accounts – User credentials
  - Cho phép user đăng nhập vào máy tính hay đăng nhập vào mạng
- Group account
  - Là tập hợp các user accounts
  - Giúp đơn giản hoá các thao tác quản trị
  - Không thể đăng nhập bằng group account

## d. Thiết lập cấu hình các giao thức mạng

Windows 2003 hỗ trợ nhiều bộ giao thức:

- TCP/IP
- IPX/SPX
- Apple Talk

# Thiết lập cấu hình TCP/IP

## ■ Đặt địa chỉ IP

- IP tĩnh
- IP động

## ■ DHCP Server

## ■ DNS Server, DNS Client

## e. Thiết lập cấu hình dịch vụ ứng dụng

- Các dịch vụ ứng dụng mở rộng chức năng của hệ điều hành mạng
- Windows 2003 có các dịch vụ:
  - IIS (Internet Information Services) với Web server, FTP server
  - Terminal service
  - ...





## II. Giới thiệu về an toàn thông tin mạng

1. Khái niệm
2. Các dạng tấn công
3. Các kỹ thuật an toàn

# 1. Khái niệm

Sự an toàn (security) liên quan 2 vấn đề:

## ■ Mất dữ liệu

- Lỗi thiết bị
- Lỗi phần mềm
- Lỗi do con người

## ■ Xâm nhập trái phép

- Truy xuất không hợp lệ
- Giả mạo
- .....

# Các mục tiêu của an toàn mạng

## ■ Tính bí mật – Confidentiality

- Bảo vệ dữ liệu trước các truy xuất không hợp lệ

## ■ Tính toàn vẹn – Integrity

- Không thay đổi hay mất dữ liệu do truy xuất không hợp lệ

## ■ Tính sẵn sàng – Availability

- Hệ thống hoạt động liên tục



## 2. Các dạng tấn công

- a. Các nhược điểm về an toàn  
(Vulnerability / Weaknesses)
- b. Một số dạng tấn công

## a. Các nhược điểm về an toàn

### ■ Nhược điểm công nghệ

- Bộ giao thức TCP/IP không an toàn
- Các hệ điều hành không hoàn thiện
- Các thiết bị mạng như router, firewall, ... không tuyệt đối an toàn

### ■ Nhược điểm về cấu hình

- Administrator thiết lập cấu hình mạng không an toàn

### ■ Nhược điểm trong quản trị, sử dụng mạng

## b. Một số dạng tấn công

### ■ Reconnaissance – Do thám

- Thu thập thông tin hệ thống bất hợp lệ

### ■ Eavesdropping – Nghe trộm

- Nghe trộm dữ liệu trên mạng phục vụ các dạng tấn công khác

### ■ Password Attacks

- Truy xuất tài nguyên không được phép (vì không có password hợp lệ)

# Một số dạng tấn công (tt)

## ■ Masquerade/IP spoofing

- Giả dạng user hợp lệ

## ■ Back doors

- Tạo đường vào hệ thống không hợp lệ phục vụ các dạng tấn công khác

## ■ Session replay

- Lưu chuỗi packet hay chuỗi lệnh ứng dụng và sử dụng lại để truy xuất không hợp lệ

# Một số dạng tấn công (tt)

## ■ Denial of Service (DoS)

- Ngăn cản người sử dụng hợp lệ bằng cách tiêu thụ tất cả tài nguyên hệ thống
- Ví dụ: Ping of death, SYN flood, E-mail bombs, malicious applets, ...

## ■ Distributed Denial of Service (DDoS)

- Làm nghẽn mạch đường truyền bằng các thông tin giả mạo

■ .....



### 3. Các kỹ thuật an toàn

- An toàn mạng là công việc thường xuyên thực hiện dựa trên các chính sách an toàn
- Các kỹ thuật liên quan phần cứng
  - An toàn vật lý cho thiết bị
  - Dùng thiết bị dự phòng (RAID, Mirror, ...)
  - Dùng các máy trạm không đĩa (diskless workstation)
- Các kỹ thuật cơ sở cho giải pháp phần mềm
  - Mã hoá dữ liệu (data encryption)
  - Kiểm chứng (authentication)

# Các giai đoạn trong chính sách an toàn

- Bảo vệ hệ thống – Secure
  - Kiểm chứng, cấp quyền truy xuất
  - Firewalls
  - Sửa chữa các nhược điểm
- Giám sát hệ thống – Monitor
  - Phát hiện các xâm nhập không hợp lệ
- Kiểm tra hoạt động – Test
- Cải tiến các kỹ thuật bảo vệ hệ thống – Improve