



Chương 01

Tổng quan

MẠNG MÁY TÍNH

Tháng 09/2011

Nội dung



1. Mạng máy tính là gì?
2. Lịch sử MMT
3. Các khái niệm cơ bản
4. Các thành phần trong mạng máy tính
5. Đồ hình mạng
6. Các ứng dụng mạng

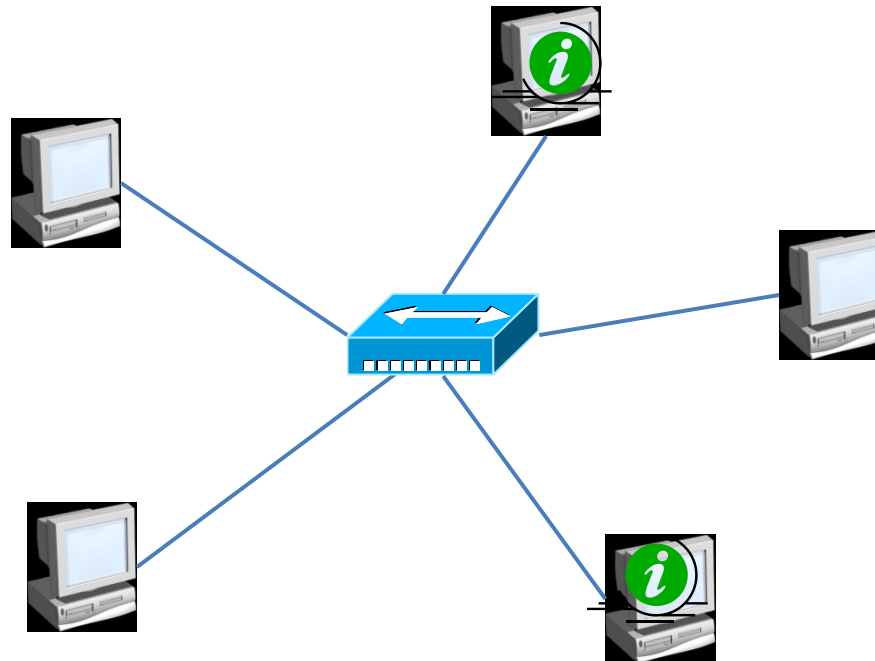


Mạng máy tính là gì?



□ Mạng máy tính (computer network):

- Nhiều *máy tính* kết nối với nhau bằng phương tiện truyền dẫn
- Liên lạc và chia sẻ tài nguyên

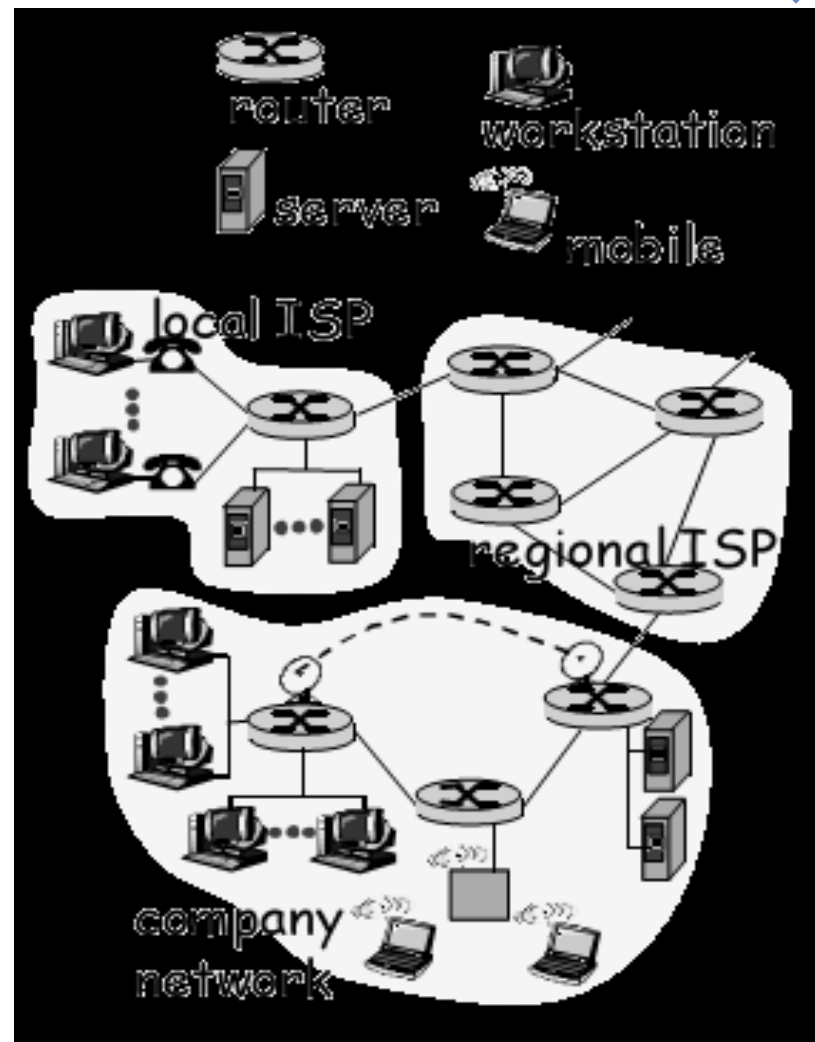


Internet?



□ Internet:

- Mạng của mạng
- Có khả năng truy cập toàn cầu



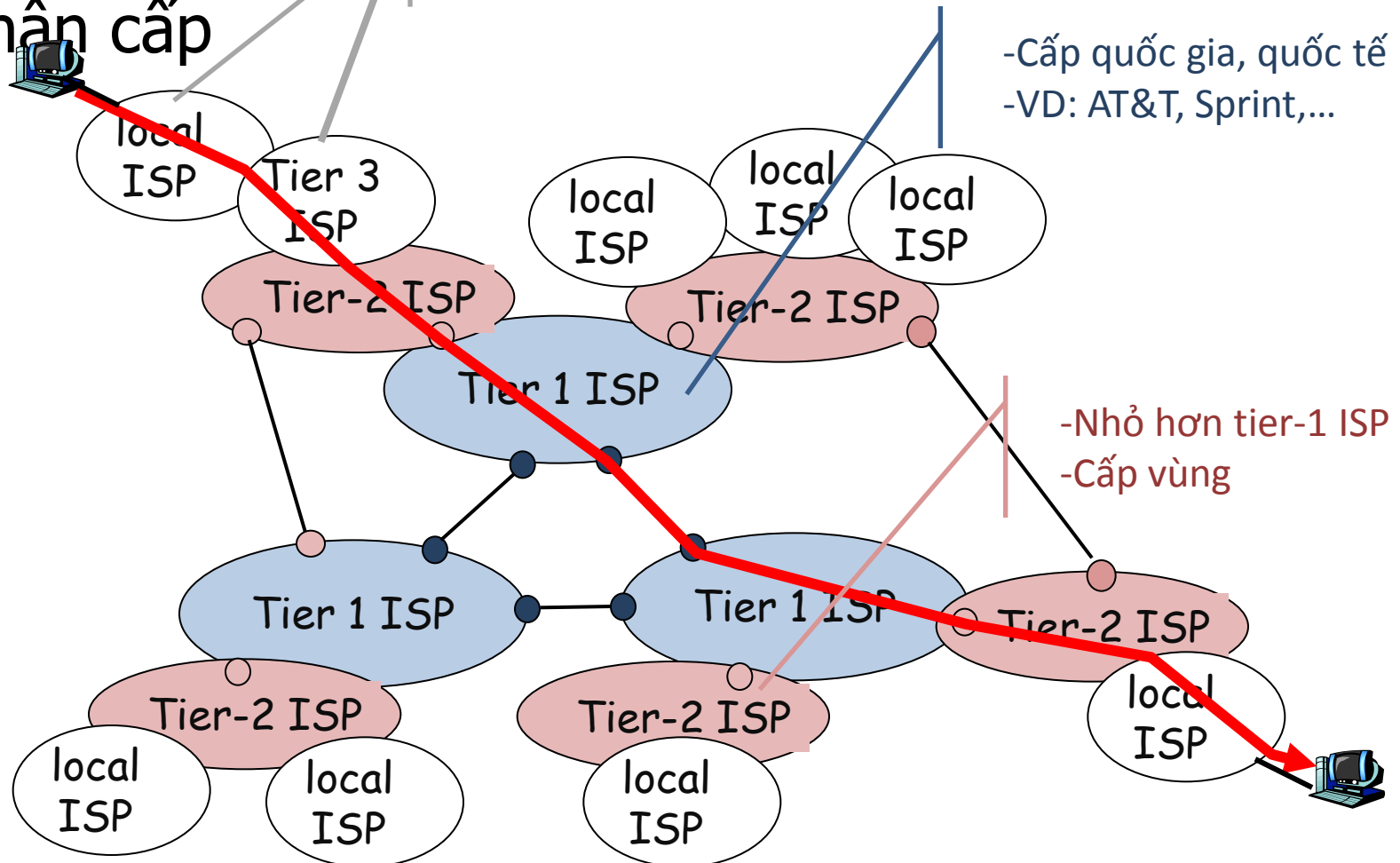
- ❑ Hạ tầng truyền thông (communication infrastructure): ứng dụng phân tán
 - Web
 - VoIP
 - Games
 - ...
- ❑ Dịch vụ truyền thông (communication services) cho các ứng dụng
 - Truyền dữ liệu đáng tin cậy
 - Truyền dữ liệu không đáng tin cậy

Cấu trúc Internet

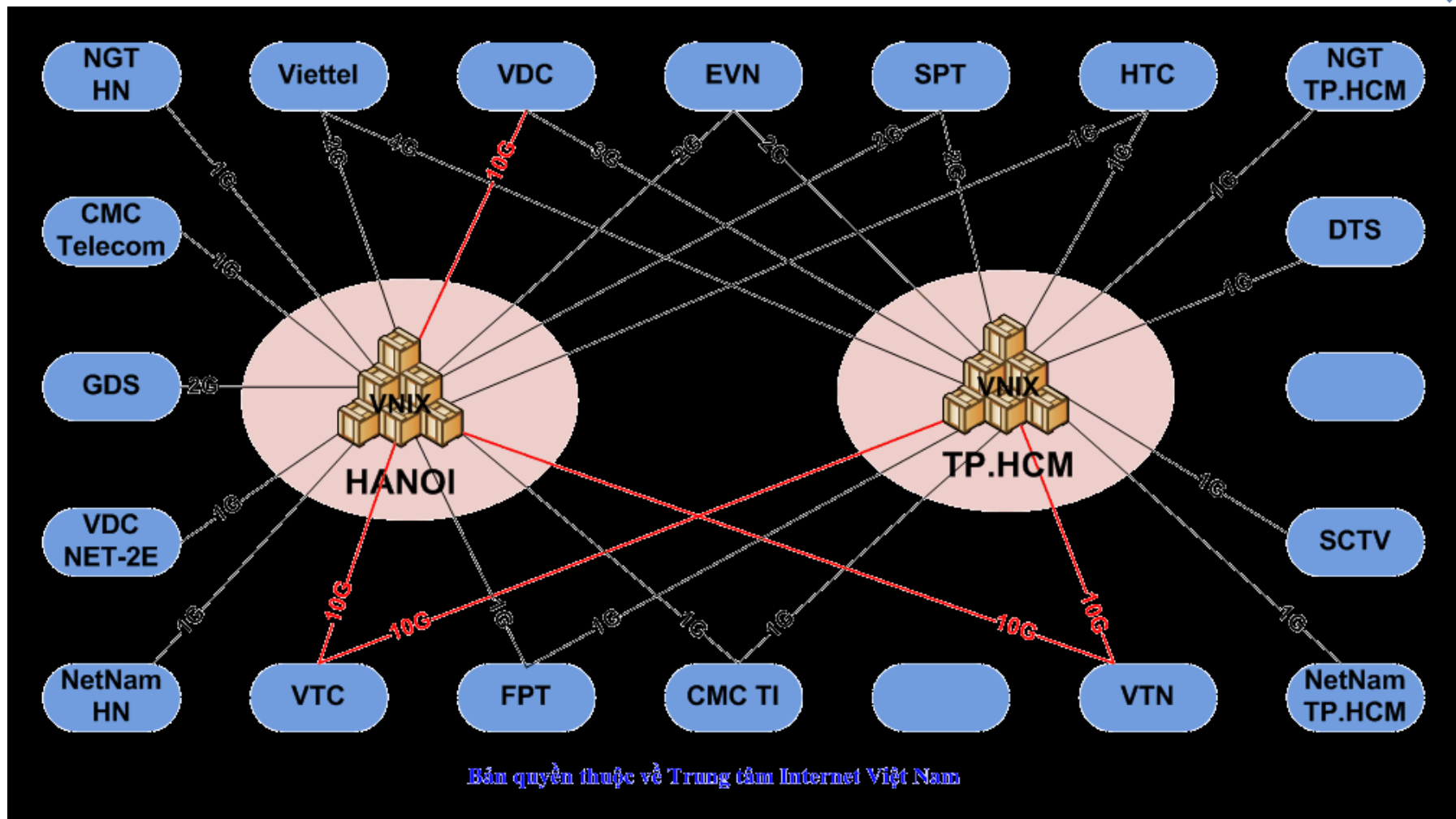


-Nhà cung cấp kết nối cho người dùng cuối
-VD: Viettel, FPT,...

□ Phân cấp



Sơ đồ kết nối các isp ở việt nam



Bản quyền thuộc về Trung tâm Internet Việt Nam



Phân loại mạng - 1



□ Theo địa hình:

- **Mạng cục bộ (LAN – Local Area Network)**
 - Kích thước nhỏ (toà nhà, phòng máy, công ty, ..)
 - Thuộc 1 đơn vị, 1 tổ chức
 - Tốc độ cao, ít lỗi
 - Rẻ tiền
- **Mạng đô thị (MAN - Metropolitan Area Network)**
 - Nhiều mạng LAN kết hợp lại
 - Có phạm vi trong 1 quận, huyện, thành phố
 - Thuộc 1 đơn vị, 1 tổ chức
 - Chậm, nhiều lỗi, chi phí cao hơn LAN
- **Mạng diện rộng (WAN - Wide Area Network)**
 - Nhiều LAN, MAN kết hợp với nhau
 - Phạm vi quốc gia, châu lục, quốc tế
 - Thuộc nhiều đơn vị, 1 tổ chức
 - Chậm, nhiều lỗi, chi phí cao hơn LAN, MAN



Phân loại mạng - 2



□ Theo phạm vi hoạt động:

- intranet
 - Nội bộ trong 1 đơn vị
- extranet
 - Intranet
 - Cho phép bên ngoài truy cập vào thông qua chứng thực
- internet
 - Cho phép bên ngoài truy cập



Phân loại mạng – 4



□ Theo phương tiện truyền dẫn:

- Có dây
- Không dây
 - Infrastructure
 - Ad-hoc



Nội dung



1. Mạng máy tính là gì?
2. Lịch sử MMT
3. Các khái niệm cơ bản
4. Các thành phần trong mạng máy tính
5. Đồ hình mạng
6. Các ứng dụng mạng



Lịch sử MMT



- Ý tưởng mầm móng đầu tiên là của [J.C.R. Licklider](#) (MIT)

"a network of such [computers], connected to one another by wide-band communication lines" which provided "the functions of present-day libraries together with anticipated advances in information storage and retrieval and [other] symbiotic functions." - J.C.R. Licklider



Lịch sử MMT (tt)



- Khởi đầu là mạng ARPANET năm 1969
 - Xuất phát từ việc phát minh ra công nghệ chuyển mạch mạch gói của [Leonard Kleinrock](#) (MIT)
 - J.C.R. Licklider và Lawrence Robert
 - 21/11/1969, mạng ARPANET đầu tiên đã kết nối 2 nơi: Trường ĐH California, Los Angeles và Viện nghiên cứu Stanford



Lịch sử MMT (tt)



□ Từ 1970s đến 1980s:

- ALOHAnet
- Telenet
- CyclabITNET
- CSNET
- NSFNET
-



Lịch sử MMT (tt)



□ 1990s: năm bùng nổ của Internet

- 1990, ARPANET đóng
- 1995, NSFNET đóng
- Rất nhiều ứng dụng ra đời
 - Email
 - Web
 - Instant message, ICQ
 - Peer-to-peer file sharing

□ 2000s: P2P, wireless, sensor, grid computing, VoIP,

...



Nội dung



1. Mạng máy tính là gì?
2. Lịch sử MMT
3. Các khái niệm cơ bản
4. Các thành phần trong mạng máy tính
5. Đồ hình mạng
6. Các ứng dụng mạng

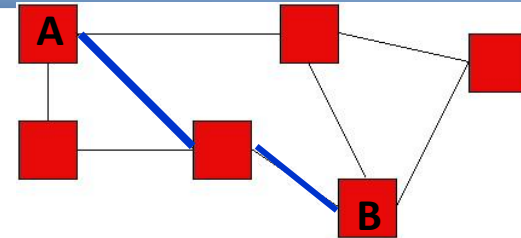


Kiểu truyền



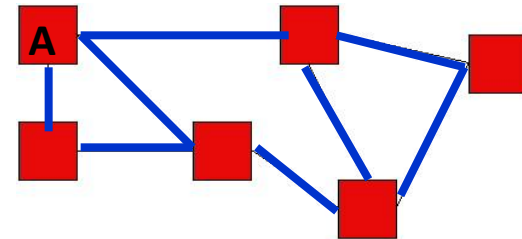
❑ Unicast

- Từ 1 node đến 1 node



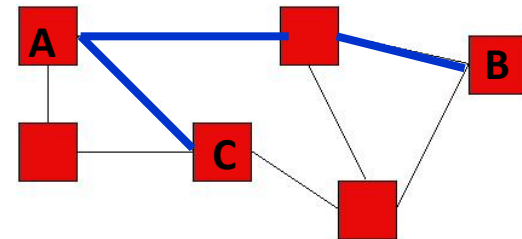
❑ Broadcast

- Từ 1 node đến tất cả các node trong một vùng mạng



❑ Multicast

- Từ 1 node đến 1 nhóm



❑ Anycast

- Từ 1 node đến 1 node bất kỳ trong một nhóm



Giao thức - 1



□ Giao thức:

- Hiểu: như là một *"thống nhất"* giữa các *"đối tượng"* khi trao đổi thông tin
- qui định, qui tắc để trao đổi dữ liệu giữa các đối tượng trên mạng
 - Định dạng dữ liệu trao đổi (syntax, semantic)
 - Thứ tự thông tin truyền nhận giữa các thực thể trên mạng
 - Các hành động cụ thể sau mỗi sự kiện nhận/gửi hay 1 sự kiện nào đó xảy ra
- VD: HTTP, TCP, IP, PPP, ...

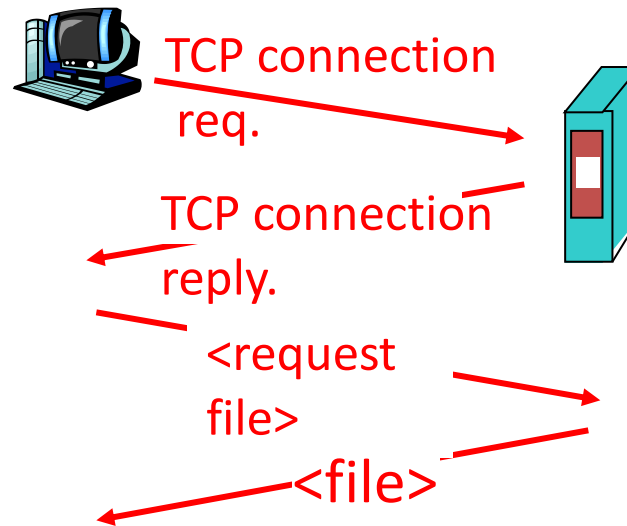
□ Do các tổ chức và hiệp hội xây dựng: IEEE, ANSI, TIA, EIA, ITU-T



Giao thức - VD



- o Giao thức TCP



□ Băng thông (bandwidth):

- Lượng thông tin có thể truyền đi trên 1 kết nối mạng trong 1 khoảng thời gian
- Lý tưởng
- Đơn vị tính: bit/s (bps), Mbps, Gbps, ...

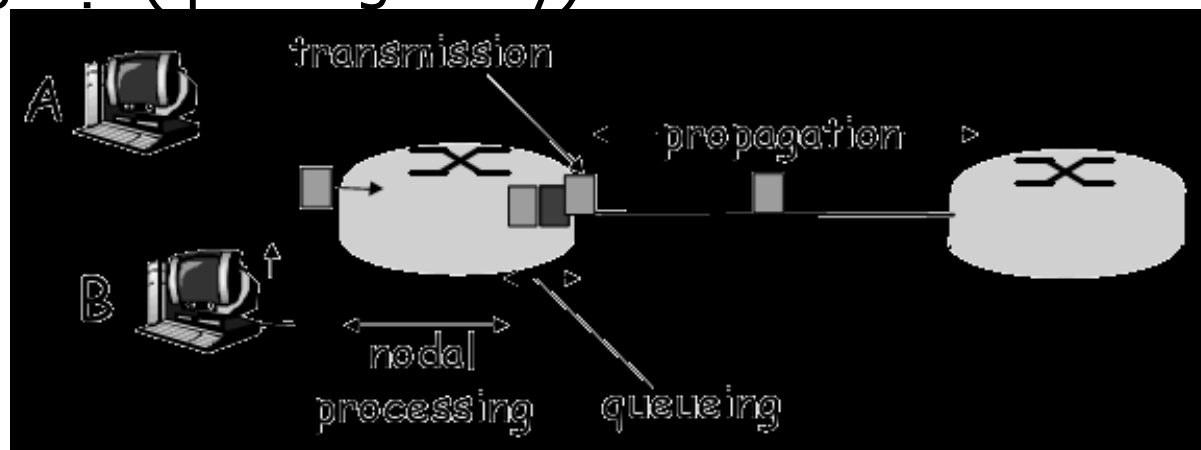
□ Thông lượng (throughput):

- Băng thông thực tế
- Nhỏ hơn nhiều so với băng thông lý thuyết
- Các yếu tố ảnh hưởng:
 - Thiết bị liên mạng
 - Topology mạng
 - Số lượng user trên mạng
 - Máy tính của user, server
 - ...

Độ trễ - 1



- Là thời gian trễ của 1 gói tin
- Các nguyên nhân gây ra trễ:
 - Trễ do tốc độ truyền (transmission delay)
 - Trễ trên đường truyền (propagation delay)
 - Xử lí tại nút (nodal processing)
 - Hàng đợi (queuing delay)



□ Trễ do tốc độ truyền (transmission delay):

- Là thời gian cần thiết để chuyển mạch hết gói tin lên đường truyền
- $D_{\text{trans}} = L/R$ (s)
 - R = băng thông của đường truyền (bps)
 - L = chiều dài gói tin (bit)
- Ví dụ: gói tin có chiều dài $L = 100$ bytes. Đường truyền có băng thông $R = 10$ Mbps

$$\rightarrow D_{\text{trans}} = 100 * 8 / (10 * 1000^2) \text{ s}$$



- ❑ Trễ trên đường truyền (propagation delay)
 - Thời gian truyền 1 bit từ nơi gửi đến nơi nhận
 - $D_{prop} = d/c$
 - d = chiều dài đường truyền
 - c = tốc độ truyền ($\sim 2 \times 10^8$ m/sec - 3×10^8 m/sec)
- ❑ Xử lý tại nút (nodal processing): D_{proc}
 - Là thời gian xử lý header của 1 gói tin và quyết định chuyển mạch gói tin theo hướng nào
 - Kiểm lỗi bit
 - Xác định đầu ra (vd dựa trên địa chỉ đến.)
 - Thường rất nhỏ

□ Hàng đợi: D_{queue}

- Là thời gian gói tin chờ trong hàng đợi để được đưa lên đường truyền
- Phụ thuộc: số lượng gói tin đến trước nó

□ Tổng độ trễ khi truyền 1 gói tin:

$$D = D_{proc} + D_{queue} + D_{trans} + D_{prop}$$



□ Ví dụ 1:

- Khoảng cách từ A đến B: 100km
- tốc độ đường truyền 360.000km/h
- Trung bình mỗi gói tin có kích thước 1000 bytes
- Băng thông của đường truyền: 100Mbps
- Mỗi gói tin cần 0.01s để xử lý

Cho biết:

- Thời gian để gửi 1 gói tin. Giả sử, tại thời điểm đang xét, hàng đợi của A là rỗng
- Tại thời điểm $t = 0.1s$, bit đầu tiên của gói tin đang ở vị trí nào?
- Tính thời gian cần thiết để gửi 5 gói tin, giả sử 5 gói tin đã nằm trong hàng đợi

Độ trễ - 6



- ❑ Các lệnh dùng để kiểm tra thời gian trễ
 - Ping
 - Tracert
 - Pathping



Độ trễ - 7



```
C:\Users\ttmtrang>ping 172.29.2.2

Pinging 172.29.2.2 with 32 bytes of data:

Reply from 172.29.2.2: bytes=32 time=1ms TTL=62
Reply from 172.29.2.2: bytes=32 time=1ms TTL=62
Reply from 172.29.2.2: bytes=32 time=1ms TTL=62
Reply from 172.29.2.2: bytes=32 time=1ms TTL=62

Ping statistics for 172.29.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

```
C:\Users\ttmtrang>tracert 172.29.2.2

Tracing route to Maillog.hcmuns.edu.vn [172.29.2.2]
over a maximum of 30 hops:

  0  1 ms  1 ms  1 ms  172.29.70.1
  1  1 ms  3 ms  4 ms  172.29.90.1
  2  2 ms  1 ms  1 ms  Maillog.hcmuns.edu.vn [172.29.2.2]

Trace complete.
```



Độ trễ - 8



```
C:\Users\ttmtrang>pathping 172.29.2.2

Tracing route to Maillog.hcmuns.edu.vn [172.29.2.2]
over a maximum of 30 hops:
 0 ttmtrang-PC [172.29.70.95]
 1 172.29.70.1
 2 172.29.90.1
 3 Maillog.hcmuns.edu.vn [172.29.2.2]

Computing statistics for 75 seconds...

Hop  RTT      Source to Here   This Node/Link   Address
 0           Lost/Sent = Pct  Lost/Sent = Pct  ttmtrang-PC [172.29.70.95]
 1    2ms      0/ 100 = 0%      0/ 100 = 0%      172.29.70.1
 2    5ms      0/ 100 = 0%      0/ 100 = 0%      172.29.90.1
 3    4ms      0/ 100 = 0%      0/ 100 = 0%      Maillog.hcmuns.edu.vn [172.29.2.2]

Trace complete.
```



Firewall



❑ Bức tường lửa (Firewall):

- Bảo vệ hệ thống
- Kiểm soát luồng dữ liệu
 - từ mạng bên trong đi ra ngoài
 - Từ bên ngoài đi vào mạng bên trong
- Phần mềm/phần cứng

Streaming Media

SMTP

External Network

DNS Intrusion

Firewall

Internal Network

Access Policy

Allow

HTTP

All Destinations

Rules Applied

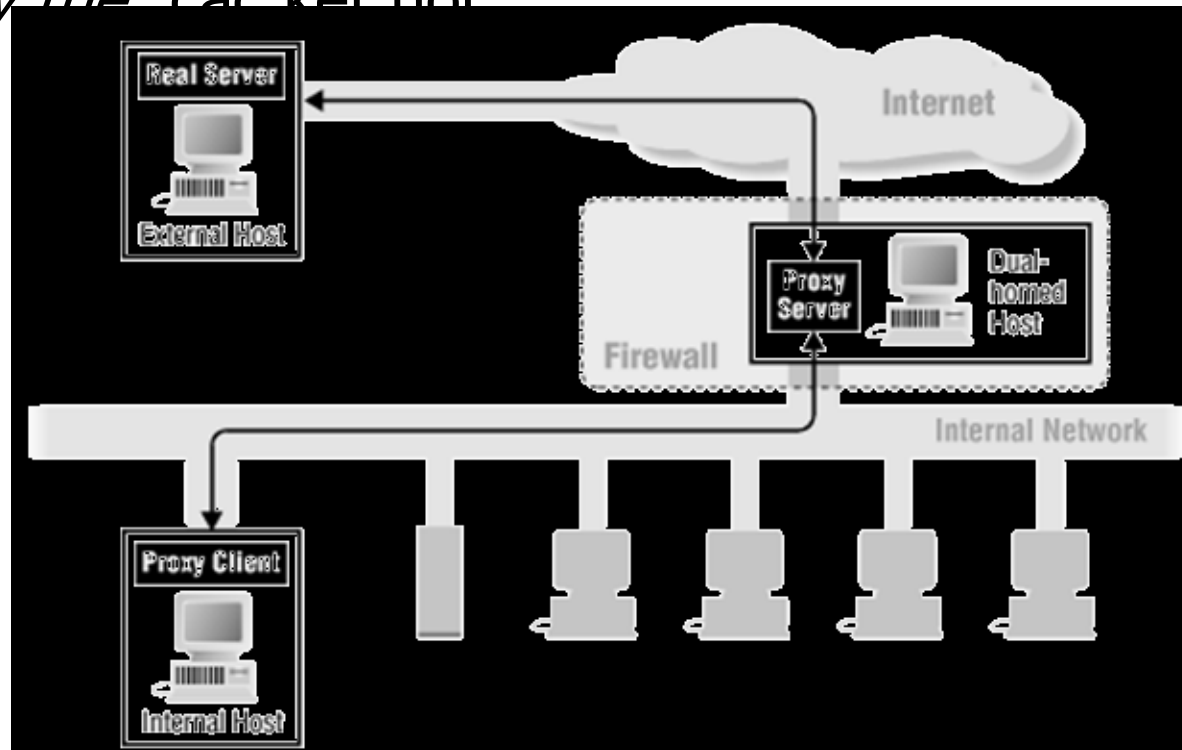


Proxy



□ Proxy

- Là 1 ứng dụng đặc biệt
- "*Thay thế*" các kết nối



Nội dung



1. Mạng máy tính là gì?
2. Lịch sử MMT
3. Các khái niệm cơ bản
4. Các thành phần trong mạng máy tính
5. Đồ hình mạng
6. Các ứng dụng mạng



Thành phần mạng – bên ngoài



PC

□ *hosts = end systems*



server

❖ chạy ứng dụng mạng



wireless laptop



cellular handheld

□ *Phương tiện kết nối*



wireless

❖ cáp, sóng vô tuyến



wired links

❖ Tốc độ truyền = *bandwidth*



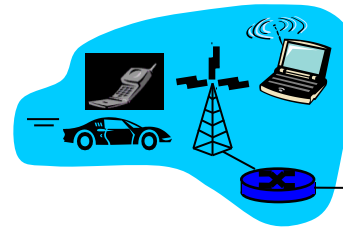
router

□ *Thiết bị liên mạng*

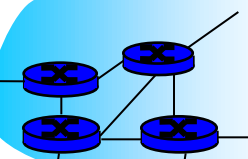
❖ Routers, switch,..

❖ Chuyển tiếp dữ liệu

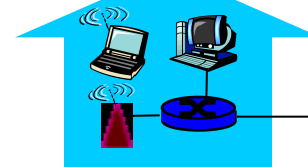
Mobile network



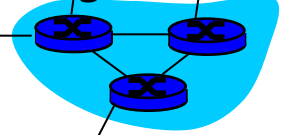
Global ISP



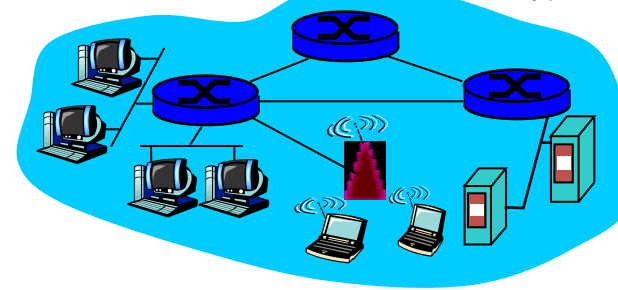
Home network



Regional ISP



Institutional network



Thành phần mạng – bên trong



Dịch vụ mạng

- Web, Mail, FTP,...

Giao thức

- HTTP, FTP, TCP, IP, PPP,...

Phương thức truyền dữ liệu:

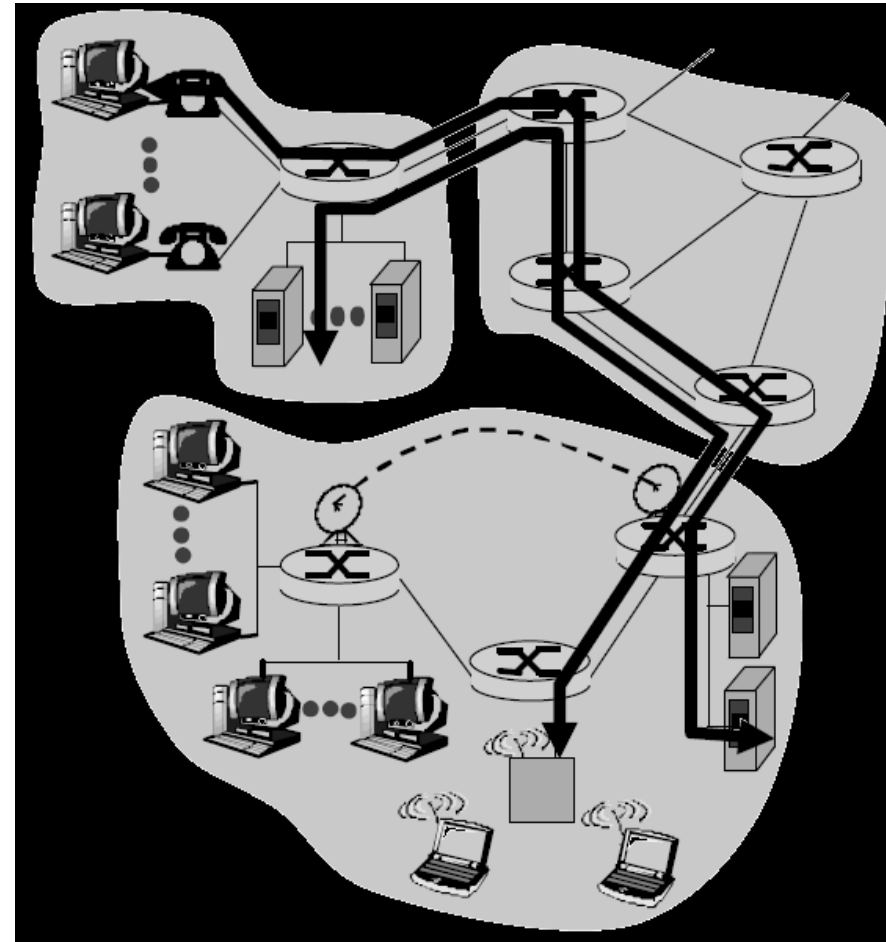
- chuyển mạch mạch (circuit-switching):
 - mỗi cuộc gọi chiếm dụng hết vùng băng thông được cấp.
 - VD: mạng điện thoại PSTN
- chuyển mạch gói (packet-switching)
 - dữ liệu chuyển mạch trên mạng rời rạc theo từng khúc, gọi là gói (packet)
 - VD: mạng máy tính



Chuyển mạch mạch - 1



- ❑ Mỗi “cuộc gọi” chiếm 1 tài nguyên nhất định
 - Yêu cầu thiết lập đường dẫn trước
 - Chiếm giữ tài nguyên suốt “cuộc gọi”
- ❑ Đảm bảo không bị nghẽn mạch
- ❑ Sử dụng băng thông không hiệu quả nếu dữ liệu rời rạc (hoặc dày đặc nhưng bit rate không đều)
- ❑ Khắc phục:
 - Nhập nhiều “cuộc gọi” trên cùng 1 đường truyền
 - kỹ thuật: FDMA, TDMA



Chuyển mạch mạch - 2



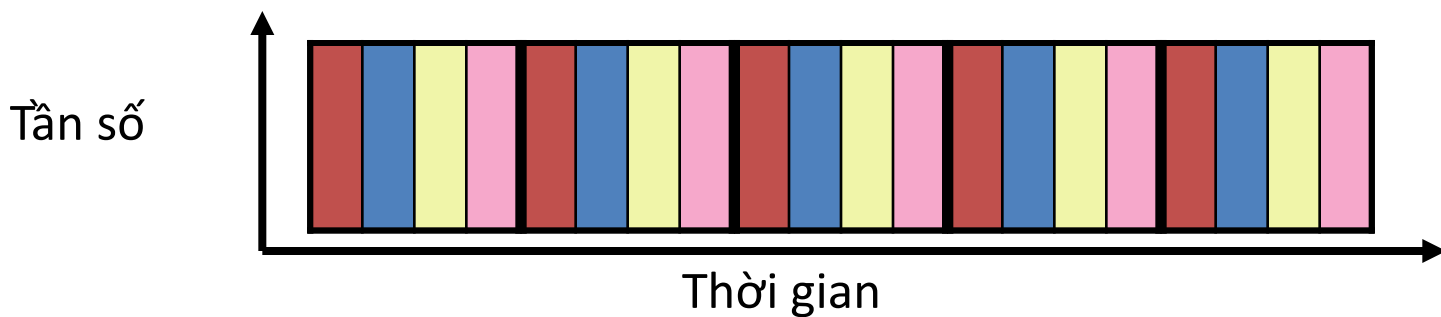
- 4 cuộc gọi



○ FDMA



○ TDMA



Chuyên mạch gói



- ❑ Mỗi luồng dữ liệu được chia nhỏ thành các gói
- ❑ Các gói tin chia sẻ chung tài nguyên mạng
 - Mỗi gói sử dụng toàn bộ băng thông
 - cho phép nhiều người dùng đồng thời hơn
 - Bị tắc nghẽn
- ❑ Yêu cầu tài nguyên sử dụng vừa đủ
- ❑ Mỗi gói có thêm phần "header" làm tăng kích thước dữ liệu truyền
- ❑ Phù hợp cho dữ liệu tức thời

Băng thông bị chia nhỏ
Cấp phát tài nguyên
Đăng kí trước tài nguyên



Nội dung



1. Mạng máy tính là gì?
2. Lịch sử MMT
3. Các khái niệm cơ bản
4. Các thành phần trong mạng máy tính
5. Đồ hình mạng
6. Các ứng dụng mạng



Đồ hình mạng - 1



□ Đồ hình mạng (network topology):

- cách thức bố trí đường truyền để nối kết các nút mạng

□ Phân loại:

- Đồ hình vật lí : Mô tả cách bố trí đường truyền thật sự
- Đồ hình logic: Mô tả con đường mà dữ liệu thật sự di chuyển.

□ Các kiểu đồ hình mạng:

- Bus: các thiết bị nối trực tiếp vào một **đường mạng chung**
- Star: các thiết bị nối trực tiếp vào một **thiết bị chung**
- Ring: các thiết bị nối với nhau tạo thành **vòng tròn**
- Mesh: 2 thiết bị bất kì được **nối trực tiếp với nhau**

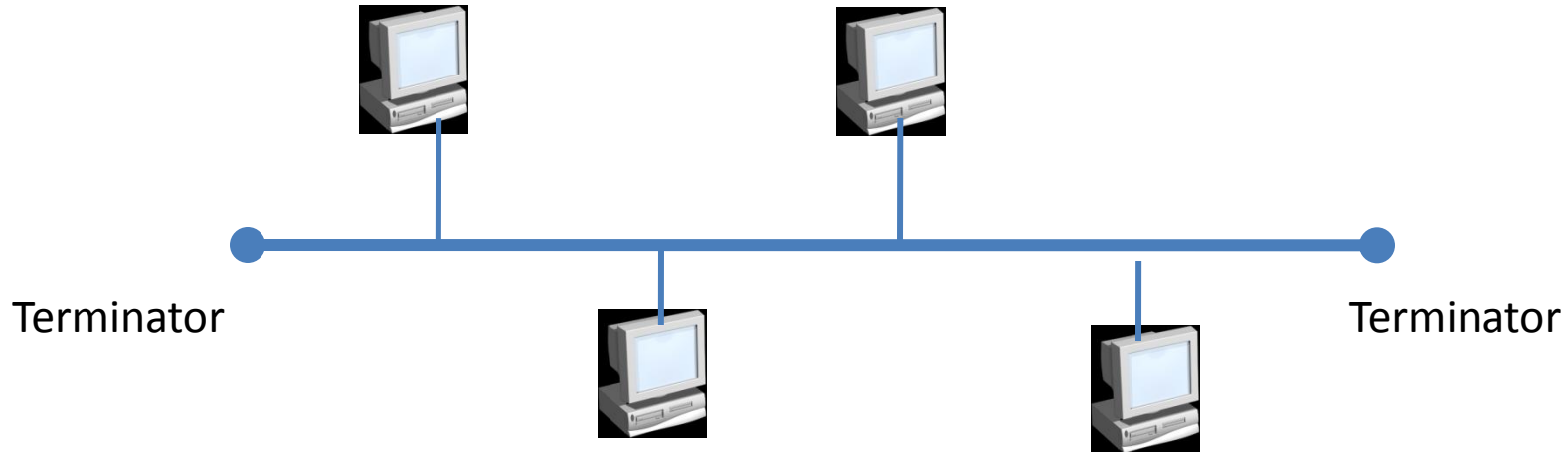


Đồ hình mạng - 2



□ Dạng bus:

- các node chia sẻ chung 1 đường truyền

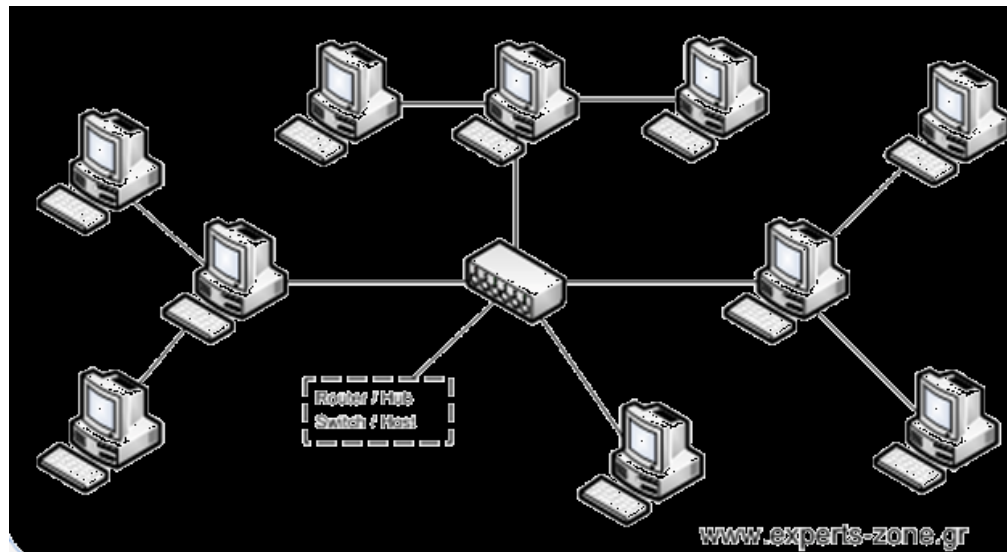
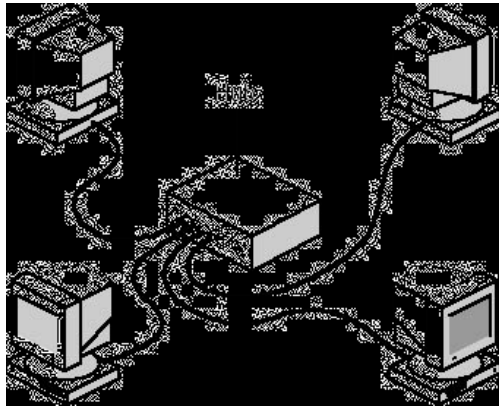


Đồ hình mạng - 3



□ Dạng sao (star)

- Các node liên kết với nhau qua 1 node trung tâm

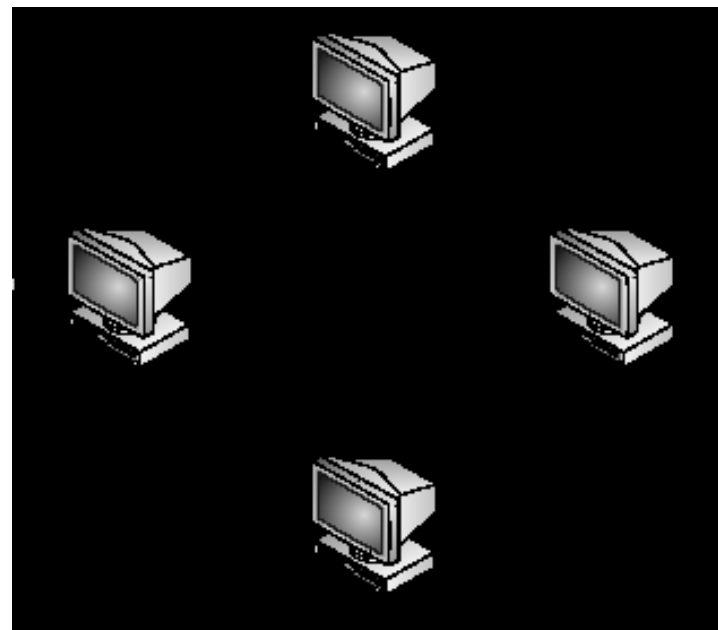
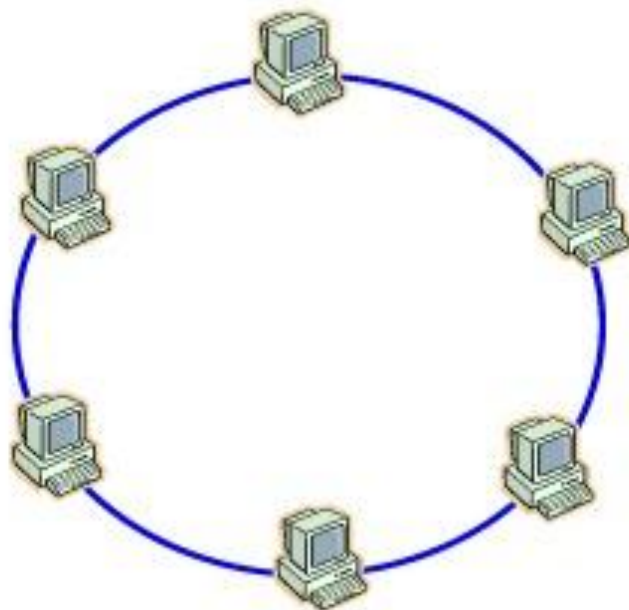


Đồ hình mạng - 4



□ Dạng vòng (ring)

- Các node nối với nhau thành 1 vòng khép kín

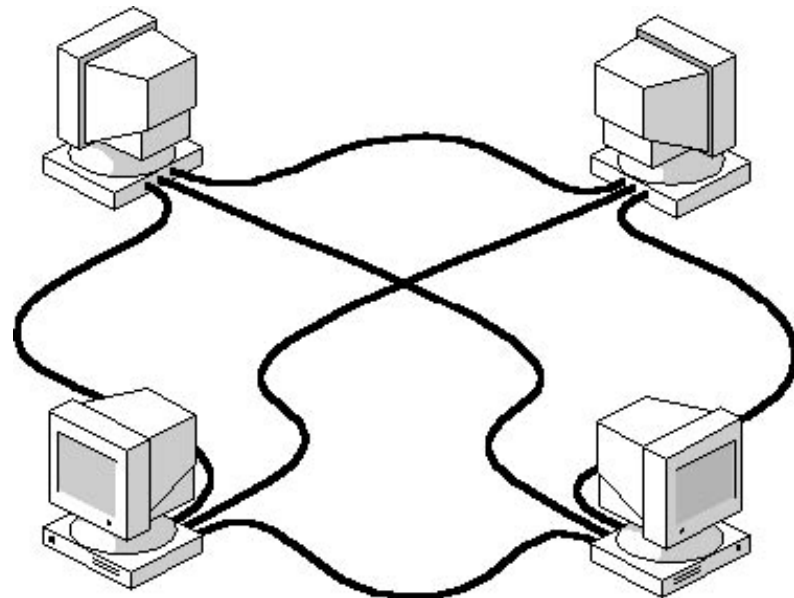


Đồ hình mạng - 5



□ Dạng lưới (mesh)

- Một node nối với nhiều node
- Gia tăng độ tin cậy của hệ thống
- Có 2 loại:
 - mesh 1 phần (bán phần)
 - mesh toàn phần



<p>Bus</p>	<p>Use of cable is economical. Media is inexpensive & easy to work with.</p> <p>System is simple and reliable.</p> <p>Bus is easy to extend.</p>	<p>Network can slow down in heavy traffic.</p> <p>Problems are difficult to isolate.</p> <p>Cable break can affect many users</p>
<p>Ring</p>	<p>System provides equal access for all computers. Performance is even despite many users.</p>	<p>Failure of one computer can impact the rest of the network. Problems are hard to isolate. Network reconfiguration disrupts operation.</p>
<p>Star</p>	<p>Modifying system and adding new computers is easy. Centralized monitoring and management are possible. Failure of one computer does not affect the rest of the network.</p>	<p>If the centralized point fails, the network fails.</p>
<p>Mesh</p>	<p>System provides increased redundancy and reliability as well as ease of troubleshooting.</p>	<p>System is expensive to install because it uses a lot of cabling.</p>

Nội dung



1. Mạng máy tính là gì?
2. Lịch sử MMT
3. Các khái niệm cơ bản
4. Các thành phần trong mạng máy tính
5. Đồ hình mạng
6. Các ứng dụng mạng



Ứng dụng mạng



- ❑ Ứng dụng trong gia đình
 - Chat
 - Mail
 - Instant message
- ❑ Ứng dụng trong thương mại
 - Trang web mua bán online
- ❑ Ứng dụng trong xã hội
 - Social network
 - Báo online
- ❑ Ứng dụng hỗ trợ cho người dùng di động
 - Wireless network

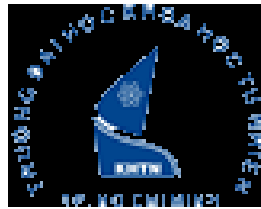


Vấn đề phát sinh



- Virus, trojan, spyware...
- Bảo mật
 - Tấn công
 - Nghe lén thông tin
 - ...
- Tội phạm





Chương 02

Địa chỉ IP và chia subnet

MẠNG MÁY TÍNH

Tháng 09/2011

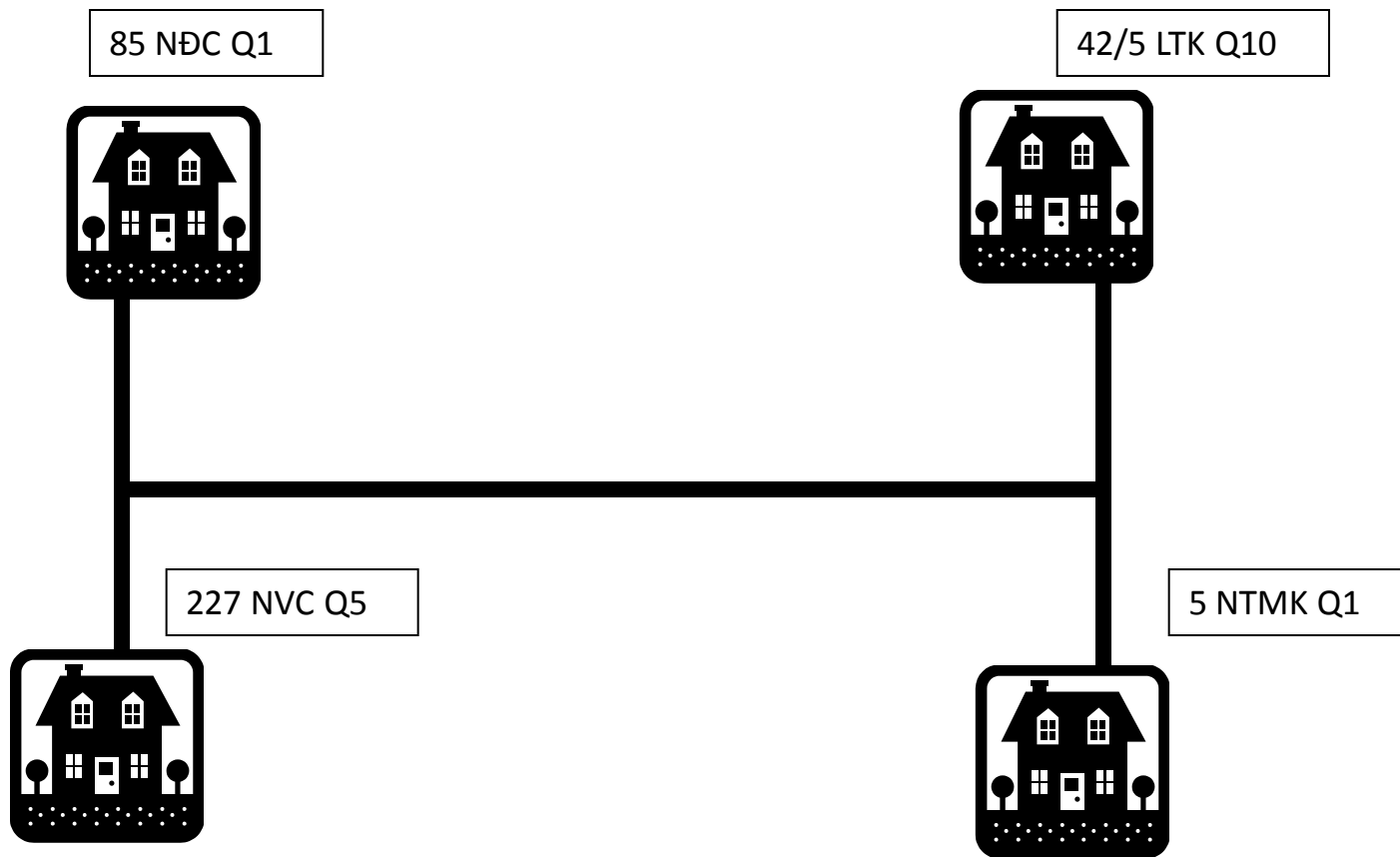
Nội dung



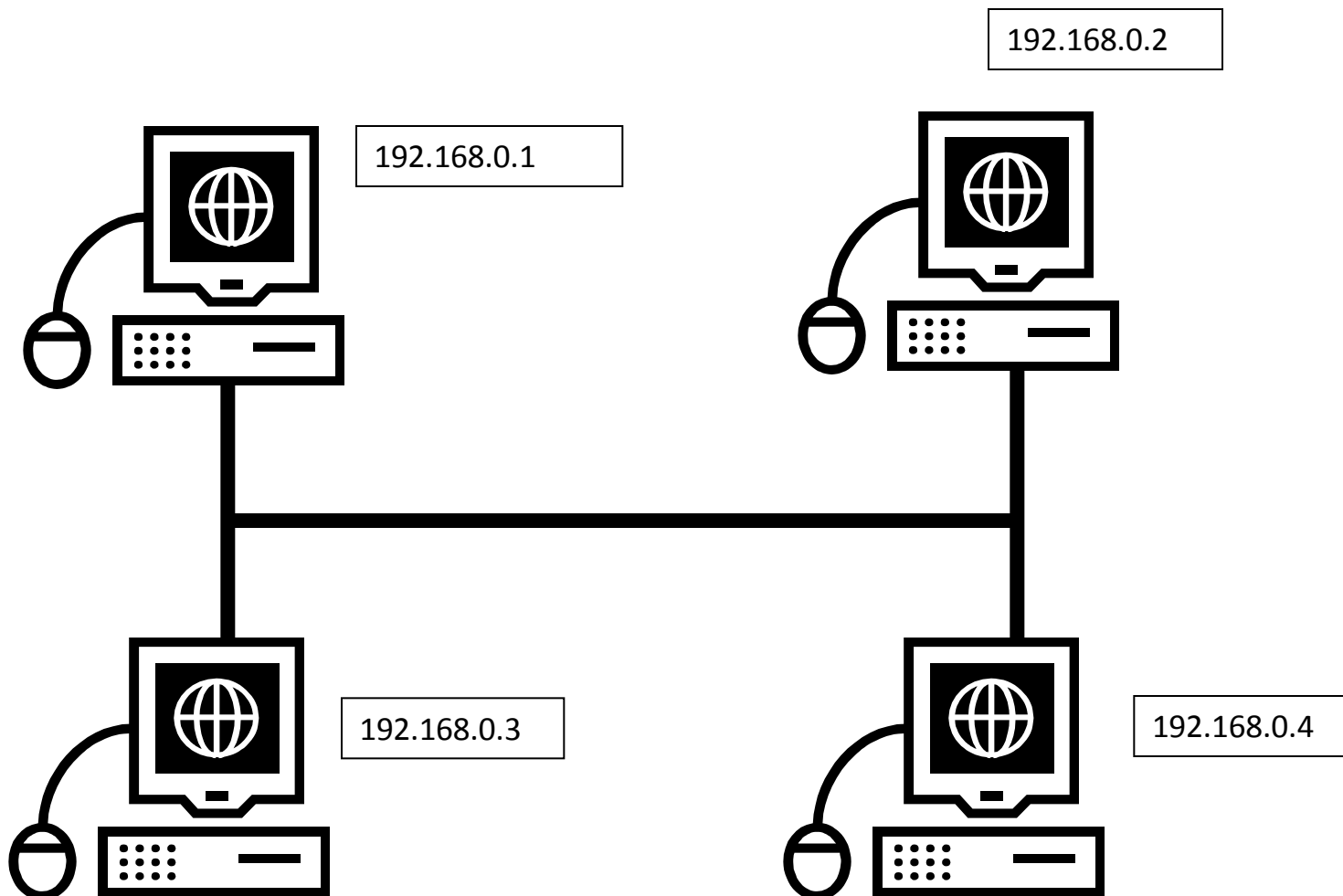
1. Giới thiệu
2. Địa chỉ IP
3. Chia subnet



Giới thiệu - 1



Giới thiệu - 2





□ Địa chỉ mạng (identifier):

- định danh của 1 node mạng

□ Phân loại:

- Địa chỉ vật lý
 - do nhà sản xuất ấn định trên sản phẩm
 - VD: địa chỉ MAC (Media Access Control)
- Địa chỉ logic
 - do người dùng ấn định
 - VD: địa chỉ IP (Internet Protocol)



Nội dung



1. Giới thiệu
2. Địa chỉ IP
3. Chia subnet



Địa chỉ IP



□ Tầng 3 trong mô hình OSI

□ Version:

- IPv4
- IPv5 ([RFC 1819](#))
- IPv6



Địa chỉ IPv4 - 1



❑ Kích thước: 4 bytes (32 bits)

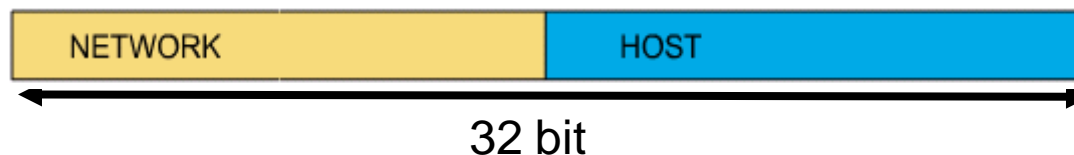
❑ Định dạng:

- Mỗi byte được biểu diễn bằng số thập phân, gọi là một octet
- hai octet được viết cách nhau bằng 1 dấu chấm “.”

VD: 10101100 00011101 00000001 00001010
172.29.1.10

❑ Chia thành 2 phần:

- Network ID (NetID)
- Host ID



Địa chỉ IPv4 - 2



172.29.1.10

NetID???
HostID???

SUBNET MASK



Địa chỉ IPv4 - 3



□ Subnet mask

- Dùng phân định phần NetID và HostID trong địa chỉ IPv4
- kích thước 4 bytes
 - các bit thuộc NetID có giá trị là 1
 - các bit thuộc HostID có giá trị là 0
- VD: 172.29.5.128/255.255.192.0
(hoặc 172.29.5.128/18)

HostIP	1010 1100	0001 1101	0000 0101	1000 0000
SubnetMask	1111 1111	1111 1111	1100 0000	0000 0000



Địa chỉ IPv4 - 5



❑ Địa chỉ đường mạng (Net Addr)

- Các bit thuộc NetID: giữ nguyên
- Các bit thuộc Host ID: xoá về 0

❑ Địa chỉ broadcast

- Các bit thuộc NetID: giữ nguyên
- Các bit thuộc Host ID: bật lên 1

VD: 192.168.1.2/24 → Net Addr: 192.168.1.0

→ đc broadcast: 192.168.1.255

HostIP	1100 0000	1010 1000	0000 0001	0000 0010
SubnetMask	1111 1111	1111 1111	1111 1111	0000 0000
Net Addr	1100 0000	1010 1000	0000 0001	0000 0000
Broadcast	1100 0000	1010 1000	0000 0001	1111 1111

Địa chỉ IPv4 - 6



□ Hai node có cùng địa chỉ đường mạng thì thuộc cùng 1 đường mạng

192.168.1.2 và 192.168.1.200 → cùng 1 đường mạng

192.168.1.2 và 192.168.2.1 → khác đường mạng

□ Số địa chỉ host hợp lệ trong 1 đường mạng

▪ $2^m - 2$

• m là số bit trong phần HostID

VD: 172.29.1.1/16

→ $m = 32 - 16 = 16$

→ Số host trong 1 network = $2^{16} - 2$



Địa chỉ IPv4 – 7



Phân lớp

Bits	1	8	9	16	17	24	25	32	
Class A	0NNNNNNN		Host				Host		Host
Range (1-255)									
Bits	1	8	9	16	17	24	25	32	
Class B	10NNNNNN		Network			Host		Host	
Range (128-191)									
Bits	1	8	9	16	17	24	25	32	
Class C	110NNNNN		Network			Network		Host	
Range (192-223)									
Bits	1	8	9	16	17	24	25	32	
Class D	1110MMMM		Multicast Address						
Range (224-239)									
Bits:	1	8	9	16	17	24	25	32	
Class E	1111MMMM		Reserved For Future Use						

Range (240-255)
Khoa Công nghệ Thông tin - Đại học Khoa học tự nhiên TP Hồ Chí Minh



Địa chỉ IPv4 - 8



□ Subnet mask mặc định:

- Lớp A: 255.0.0.0 (/8)
- Lớp B: 255.255.0.0 (/16)
- Lớp C: 255.255.255.0 (/24)

□ VD:

15.19.18.29
↓
00001111

- Lớp A
- Subnet mask mặc định: 255.0.0.0



Địa chỉ IPv4 – 9



□ Cho địa chỉ IP: 172.29.7.10

- Lớp:
- Net Addr :
- Số host trong cùng network:
- Các địa chỉ của host:
- Địa chỉ broadcast:



Địa chỉ IPv4 – 10



□ Cho địa chỉ IP: 172.29.7.10

- Lớp: B
- Net Addr : 172.29.0.0
- Số host trong cùng network: $2^{16}-2$
- Các địa chỉ: 172.29.0.1 – 172.29.255.254
- Địa chỉ broadcast: 172.29.255.255



Địa chỉ IPv4 - 11



□ Phân loại:

- Địa chỉ public:
 - dùng để trao đổi trên Internet
 - Địa chỉ thật
- Địa chỉ private
 - Dùng để đánh địa chỉ cho các mạng LAN bên trong 1 tổ chức
 - Địa chỉ ảo
- Địa chỉ loopback: 127.0.0.0 – 127.255.255.255

Table 2-4 Private IP Address Information

Class	Address (range)	Networks	Total Private Hosts
Class A	10.0.0.0	1	16,777,214
Class B	172.16.0.0–172.31.0.0	16	1,048,544
Class C	192.168.0.0–192.168.255.0	256	65,024



Nội dung



1. Giới thiệu
2. Địa chỉ IP
3. Chia subnet



Chia subnet - 1



□ Mục tiêu:

- giảm số lượng node → Tăng thông lượng mạng
- Tăng tính bảo mật
- Dễ quản trị
- Dễ bảo trì
- Tránh lãng phí địa chỉ IP



Chia subnet - 2



□ Qui tắc:

- Mượn các **bit đầu** trong HostID làm NetID
- Số subnet = 2^n (n: số bit vay mượn phần HostID)

□ Lên kế hoạch:

- Số subnet cần chia
- Số node trong mỗi subnet



Chia subnet – 3.1: Ví dụ 1



- Công ty A được cấp đc đường mạng là: **172.29.0.0/16**. Công ty muốn chia thành **10** subnet trong đó có 3 subnet có 100 PCs, 4 subnet có 255 PCs, 3 subnet có **500** PCs

Dùng 4 bit chia subnet



Chia subnet – 3.2: Ví dụ 1



□ Các subnet:

Subnet	Net Addr	HostIP	Broadcast
0000 0000	172.29.0.0	172.29.0.1-172.29.15.254	172.29.15.255
0001 0000	172.29.16.0	172.29.16.1-172.29.31.254	172.29.31.255
0010 0000	172.29.32.0	172.29.32.1-172.29.47.254	172.29.47.255
0011 0000	172.29.48.0	172.29.48.1-172.29.63.254	172.29.63.255
0100 0000	172.29.64.0	172.29.64.1-172.29.79.254	172.29.79.255
0101 0000	172.29.80.0	172.29.80.1-172.29.95.254	172.29.95.255
...



Chia subnet – 3.3: Ví dụ 1



Net Addr		Subnet Mask
172.29.0.0	1111 0000	255.255.240.0
172.29.16.0	1111 0000	255.255.240.0
172.29.32.0	1111 0000	255.255.240.0
172.29.48.0	1111 0000	255.255.240.0
172.29.64.0	1111 0000	255.255.240.0
172.29.80.0	1111 0000	255.255.240.0
...



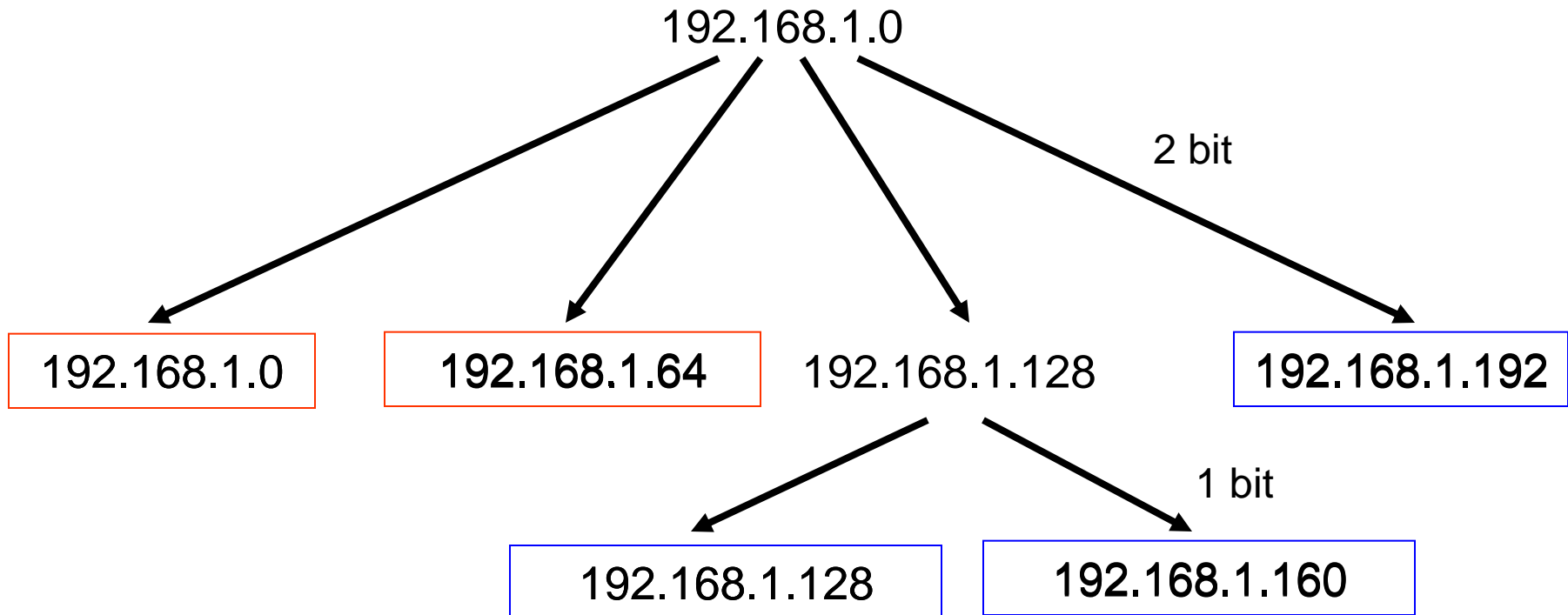
Chia subnet – 4.1: Ví dụ 2



- Ví dụ
mạng
thành
30



Chia subnet – 4.2: Ví dụ 2



Chia subnet – 4.3: Ví dụ 2

Net Addr		Subnet Mask
192.168.1.0	1100 000	255.255.255.192
192.168.1.64	1100 000	255.255.255.192
192.168.1.172	1100 000	255.255.255.192
192.168.1.128	1110 000	255.255.255.224
192.168.1.160	1110 000	255.255.255.224

Chia subnet - 5



□ Giá trị các subnetmask:

128	64	32	16	8	4	2	1	
↓	↓	↓	↓	↓	↓	↓	↓	
0	0	0	0	0	0	0	0	≡ 0
1	0	0	0	0	0	0	0	≡ 128
1	1	0	0	0	0	0	0	≡ 192
1	1	1	0	0	0	0	0	≡ 224
1	1	1	1	0	0	0	0	≡ 240
1	1	1	1	1	0	0	0	≡ 248
1	1	1	1	1	1	0	0	≡ 252
1	1	1	1	1	1	1	0	≡ 254
1	1	1	1	1	1	1	1	≡ 255



Cho 172.100.112.4/19. Hãy cho biết:

1. Địa chỉ trên thuộc về đường mạng nào?
2. Số IP hợp lệ có thể dùng trong đường mạng đó. Và hãy cho biết gồm những địa chỉ nào?
3. Địa chỉ broadcast của đường mạng đó
4. Với địa chỉ đường mạng trên, hãy chia thành 5 subnet như sau: 2 subnet có 1000 host, 2 subnet có 500 host, 1 subnet có 100 host
5. Với địa chỉ đường mạng trên, hãy chia thành 17 subnet như sau: 4 subnet có 1000 host, 6 subnet có 500 host, 7 subnet có 100 host



Chương 03

Mô hình OSI và TCP/IP

MẠNG MÁY TÍNH

Tháng 09/2011

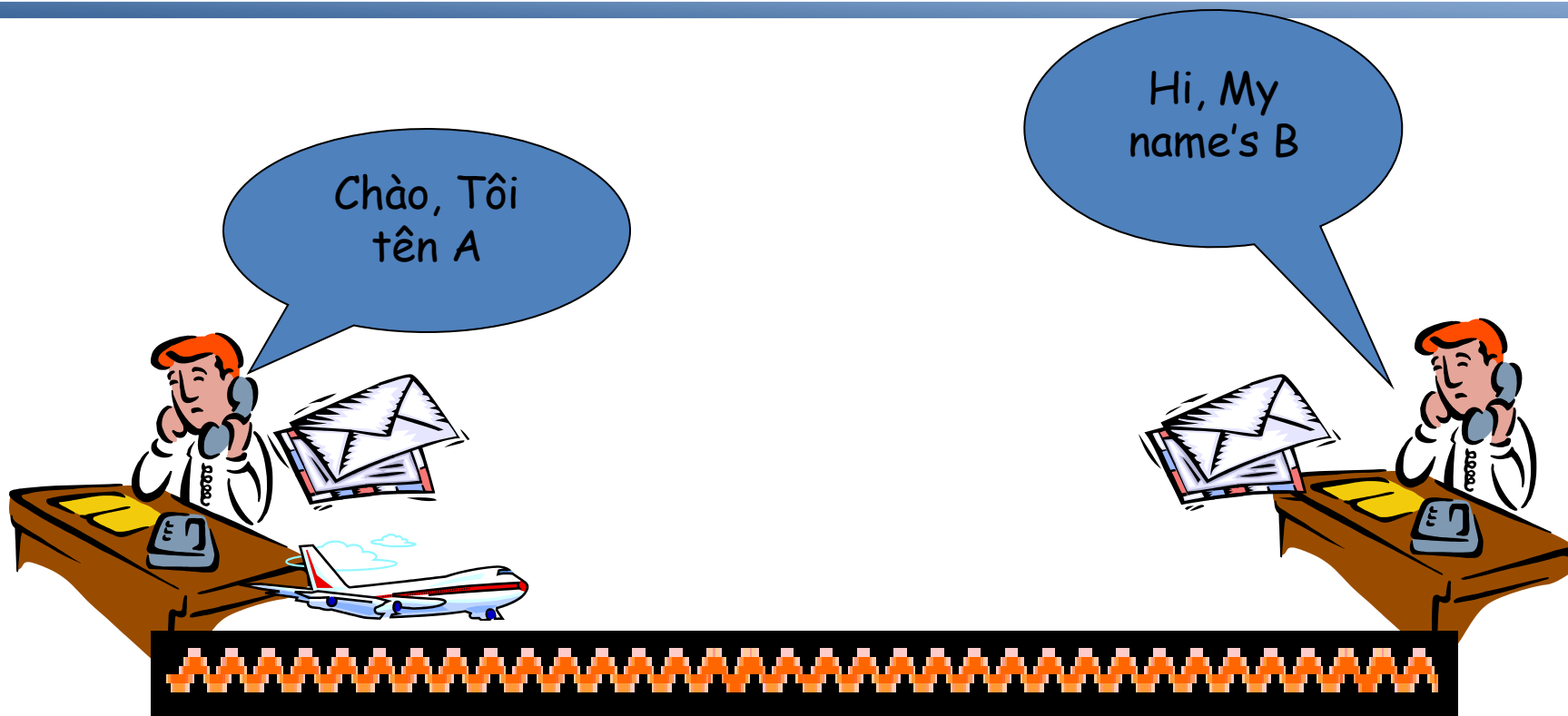
Nội dung



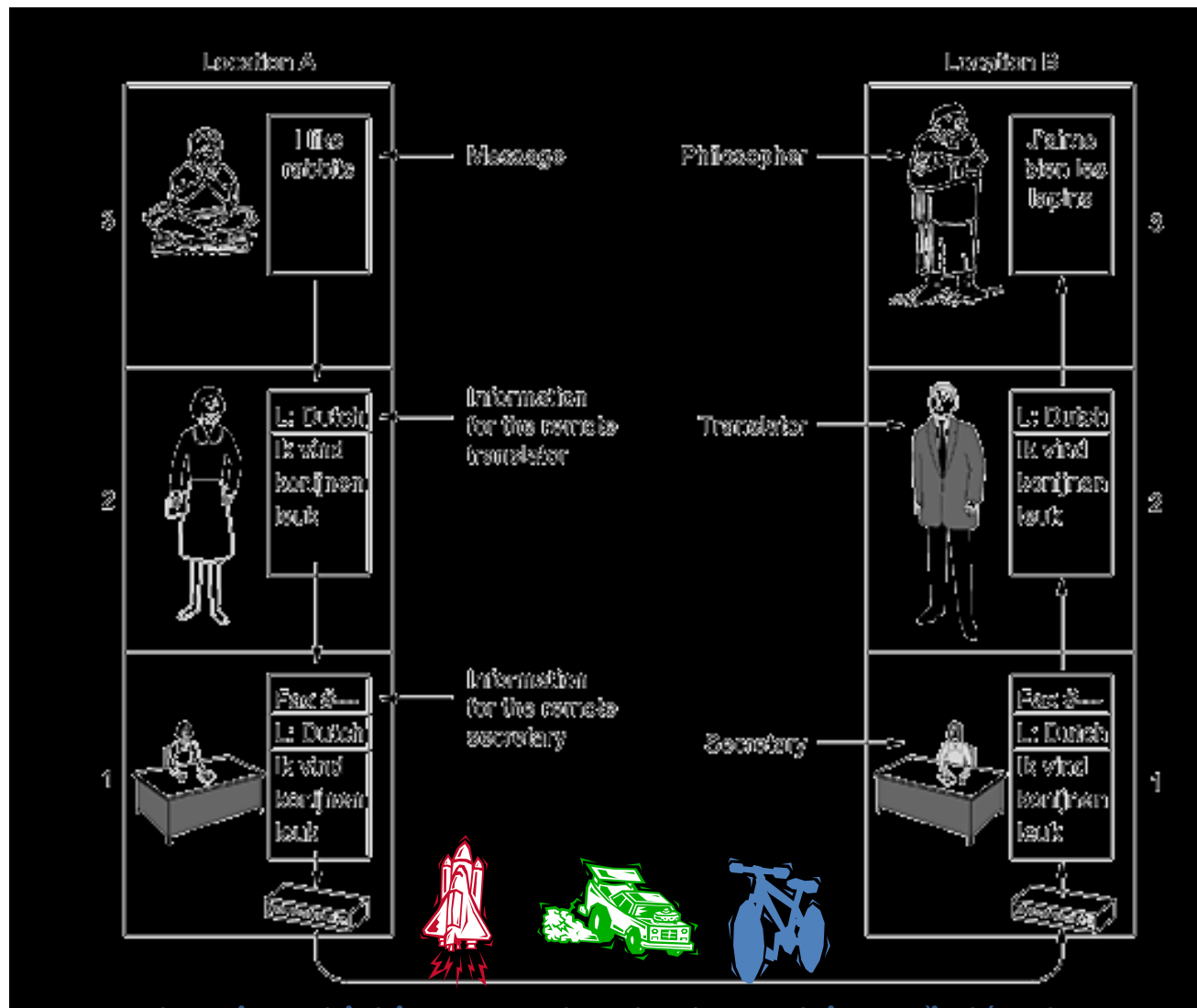
1. Giới thiệu
2. Mô hình OSI
3. Mô hình TCP/IP
4. Đóng gói dữ liệu



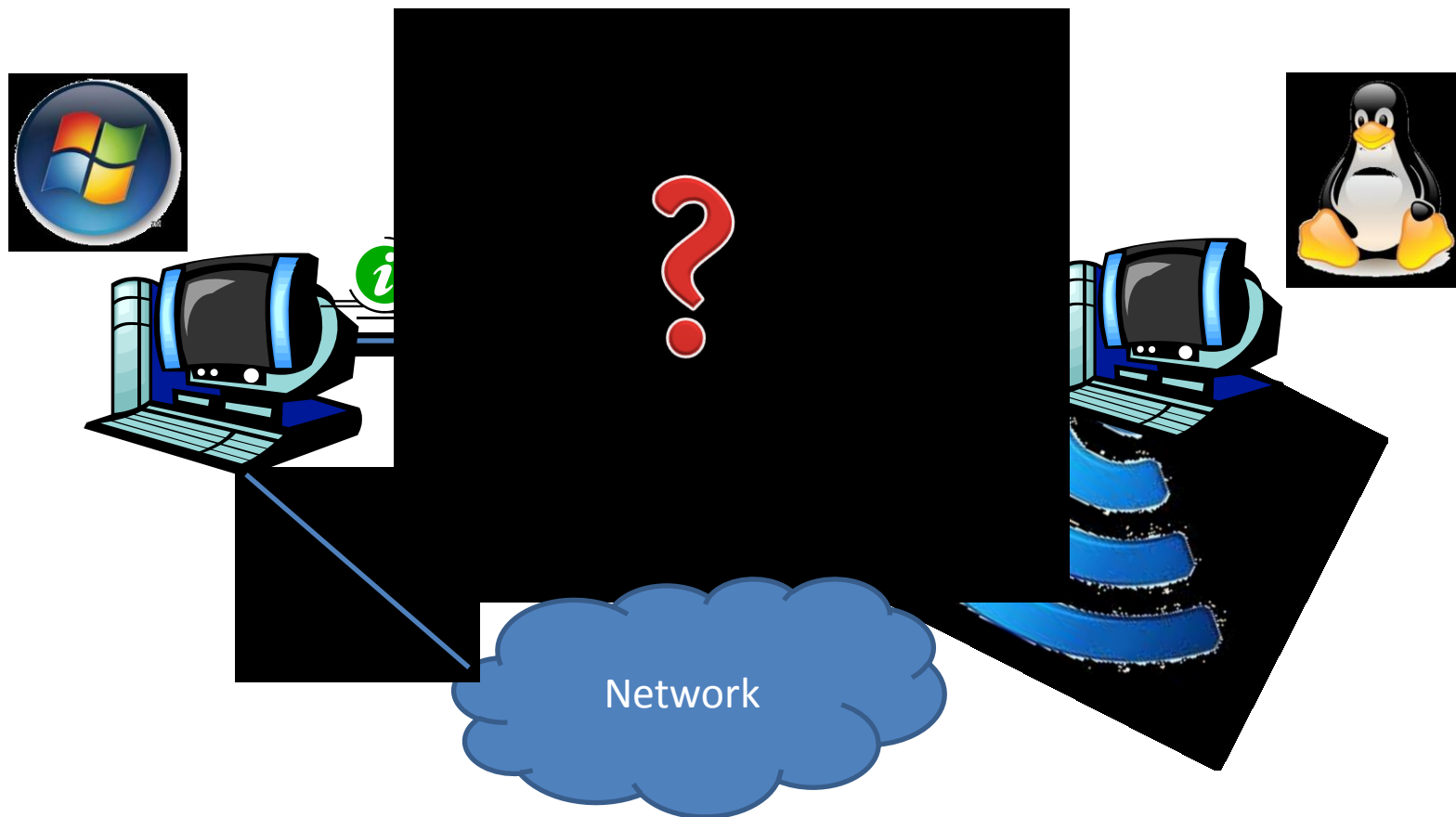
Giới thiệu - 1



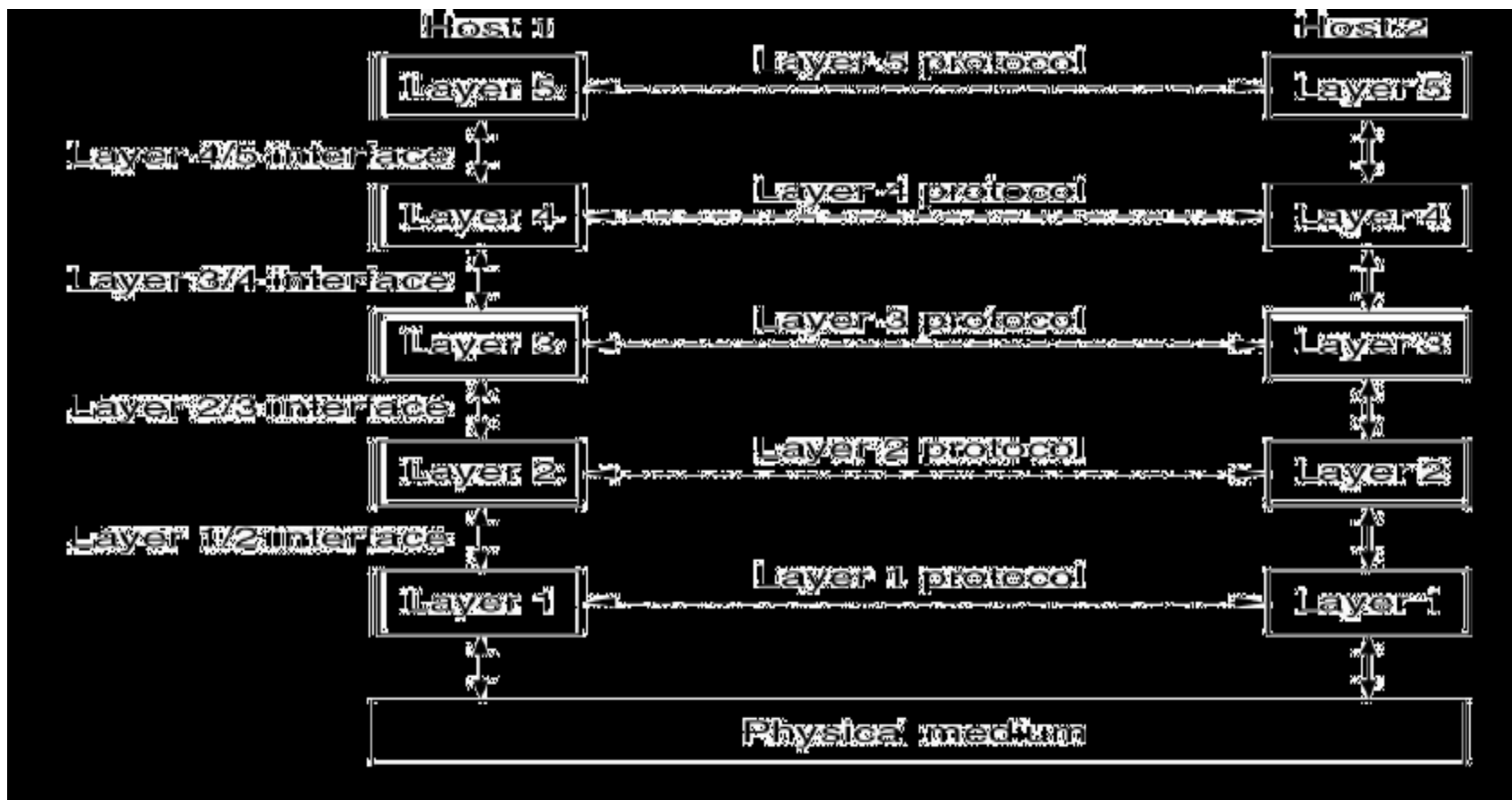
Giới thiệu - 2



Giới thiệu - 3



Giới thiệu - 4



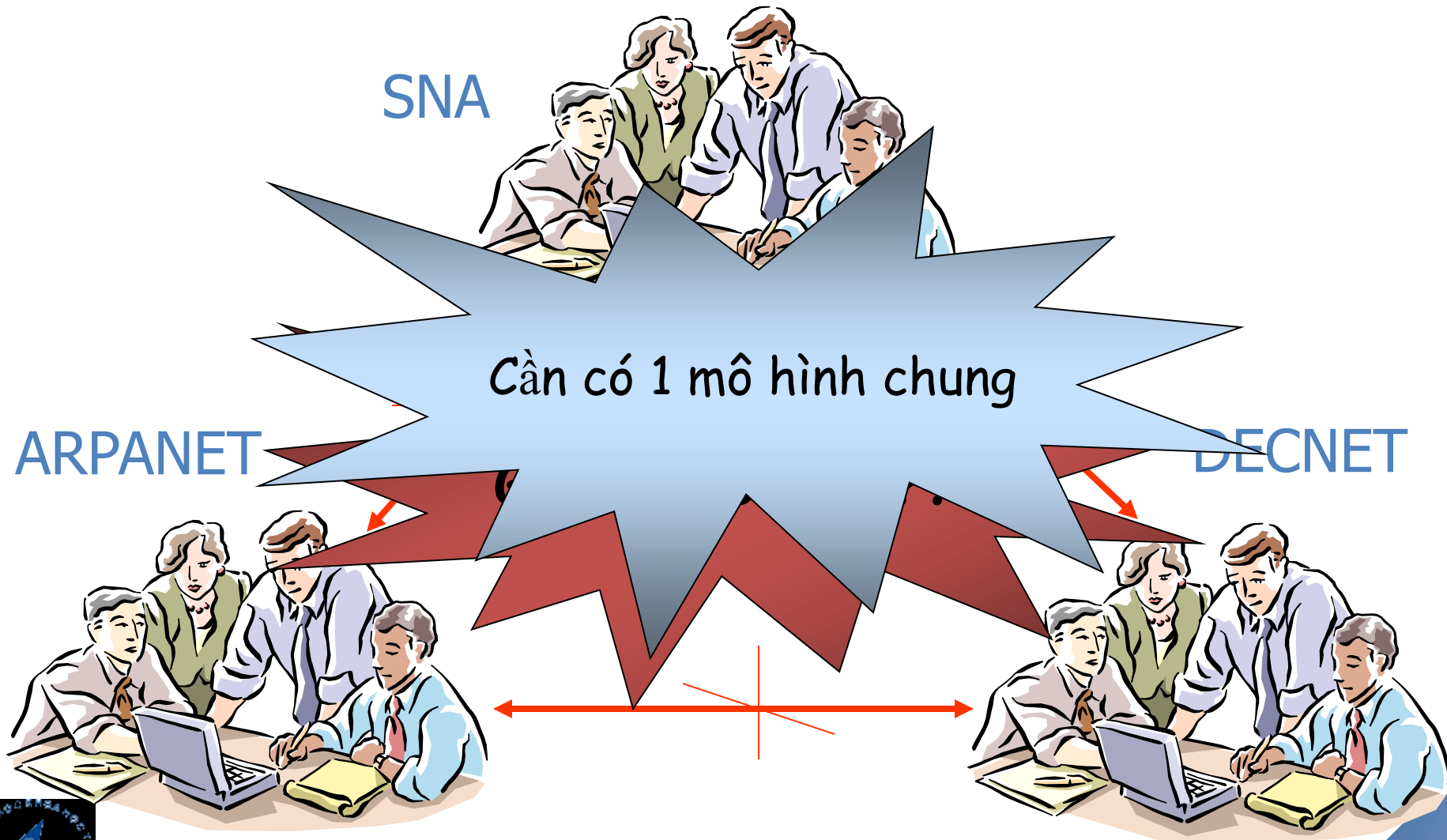
Giới thiệu - 5



- ❑ Xem mạng như là 1 chồng các lớp (layer)
- ❑ Lớp N cung cấp các dịch vụ cho lớp N+1
- ❑ Mỗi lớp trao đổi với nhau theo 1 giao thức
- ❑ Ưu điểm:
 - Mỗi lớp có 1 chức năng riêng, lớp N sử dụng dịch vụ do lớp N-1 cung cấp
 - Giảm độ phức tạp khi xử lý dữ liệu
 - Dễ quản lý
 - Dễ mở rộng, dễ phát triển
 - Đơn giản



Giới thiệu - 5



Nội dung



1. Giới thiệu
2. Mô hình OSI
3. Mô hình TCP/IP
4. Đóng gói dữ liệu



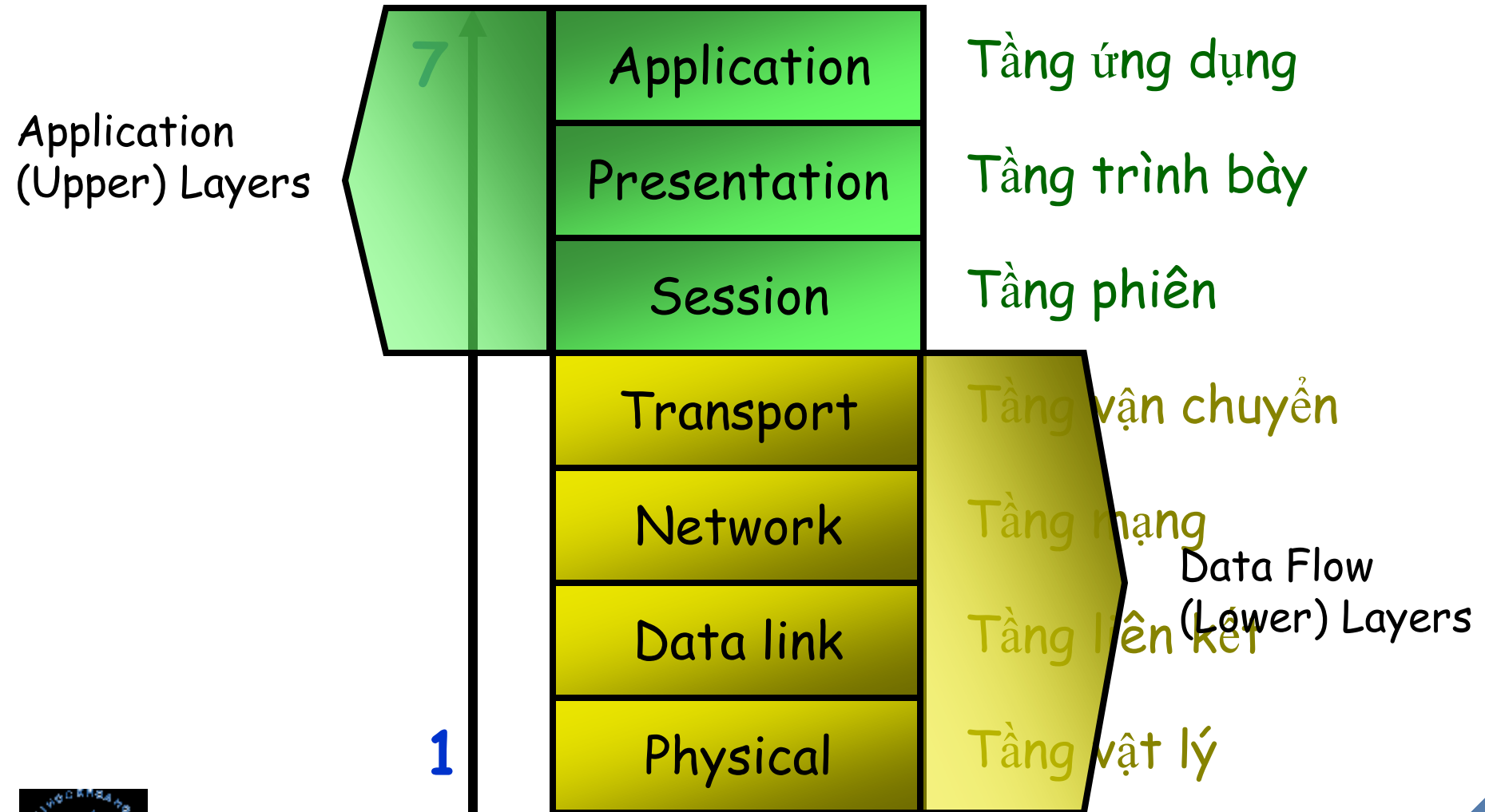
OSI Model - 1



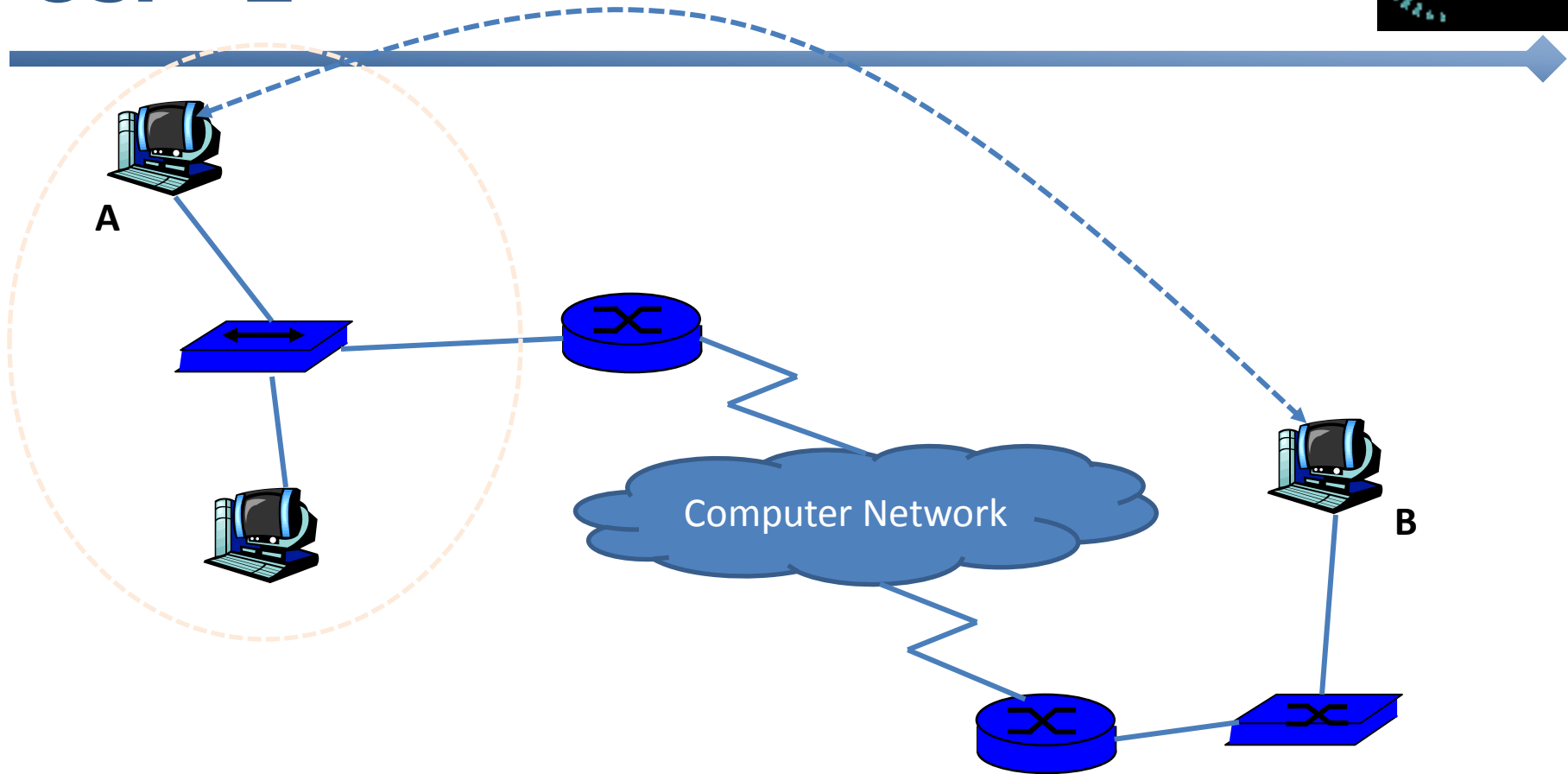
- ❑ Mô hình OSI (Open Systems Interconnection):
 - do tổ chức ISO (International Organization for Standardization) đề xuất từ 1977
 - công bố lần đầu vào 1984
 - Là **khung sườn** biểu diễn cách thông tin di chuyển trên mạng như thế nào



OSI Model - 2



Osi - 2



OSI Model - 4



Application	Cung cấp các dịch vụ mạng
Presentation	Cách biểu diễn dữ liệu
Session	Quản lý các phiên của ứng dụng



OSI Model - 5



Transport	Truyền dữ liệu end-to-end
Network	Truyền dữ liệu host-host
Data link	Truyền dữ liệu link-link
Physical	Truyền dữ liệu nhị phân



Nội dung



1. Giới thiệu
2. Mô hình OSI
3. Mô hình TCP/IP
4. Đóng gói dữ liệu



Mô hình TCP/IP - 1



- TCP/IP - Transmission Control Protocol/Internet Protocol
- Do Cerf và Kahn định nghĩa vào năm 1974
- Đặc tả chồng giao thức



Mô hình TCP/IP - 2

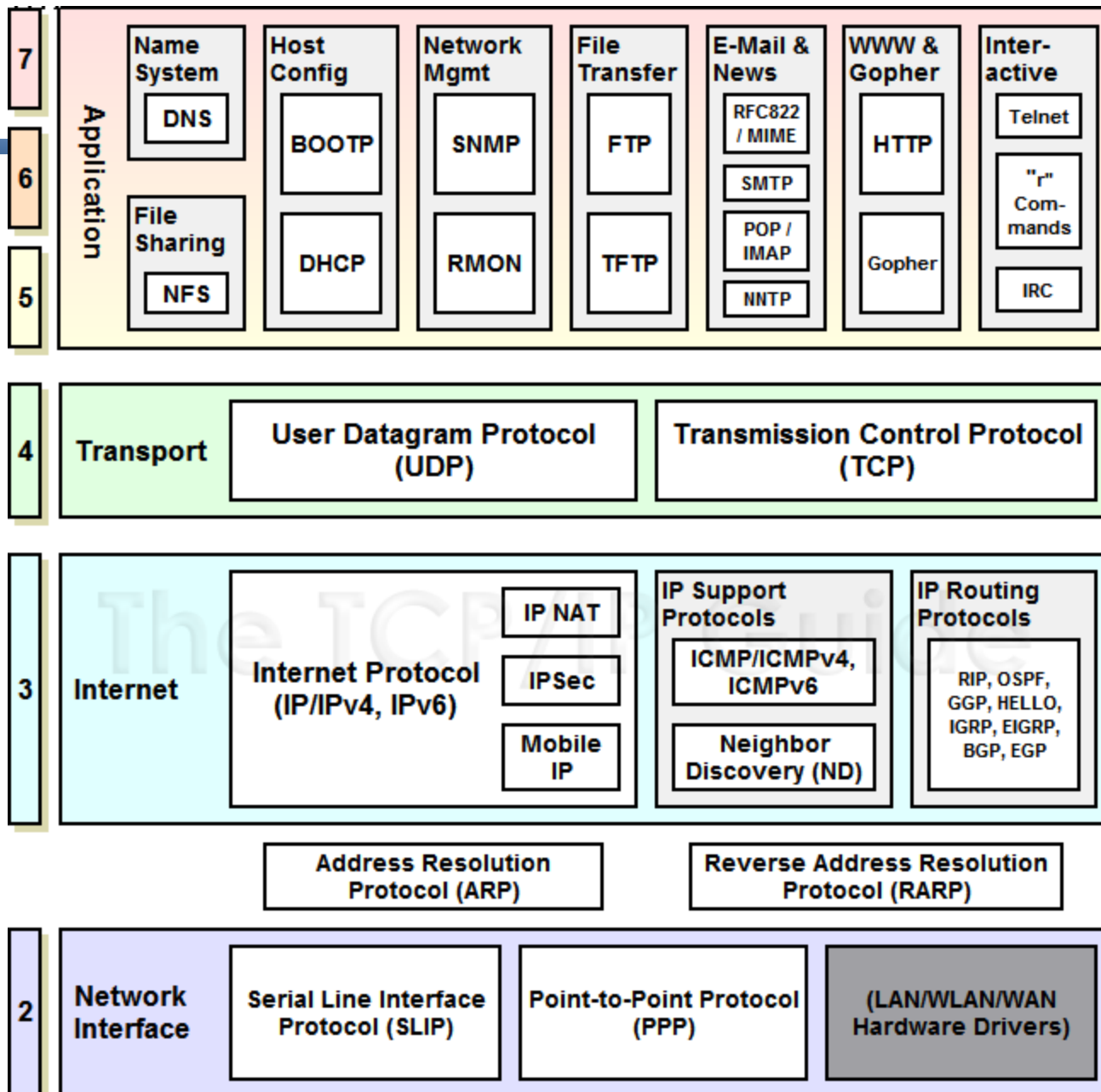


OSI

TCP/IP

Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data link	Host to Network (Network Interface)
Physical	





Nội dung



1. Giới thiệu
2. Mô hình OSI
3. Mô hình TCP/IP
4. Đóng gói dữ liệu



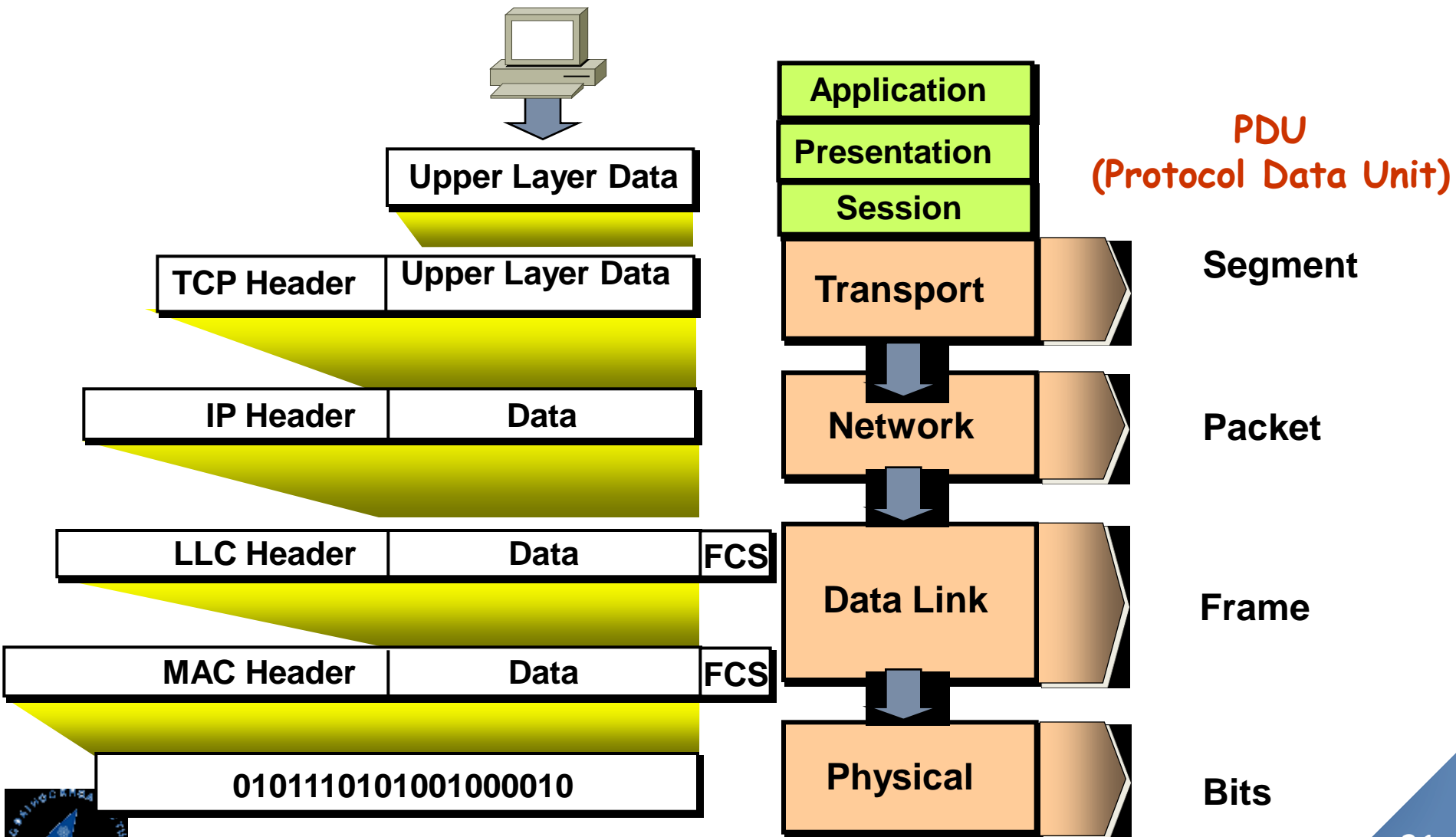
Đóng gói dữ liệu - 1



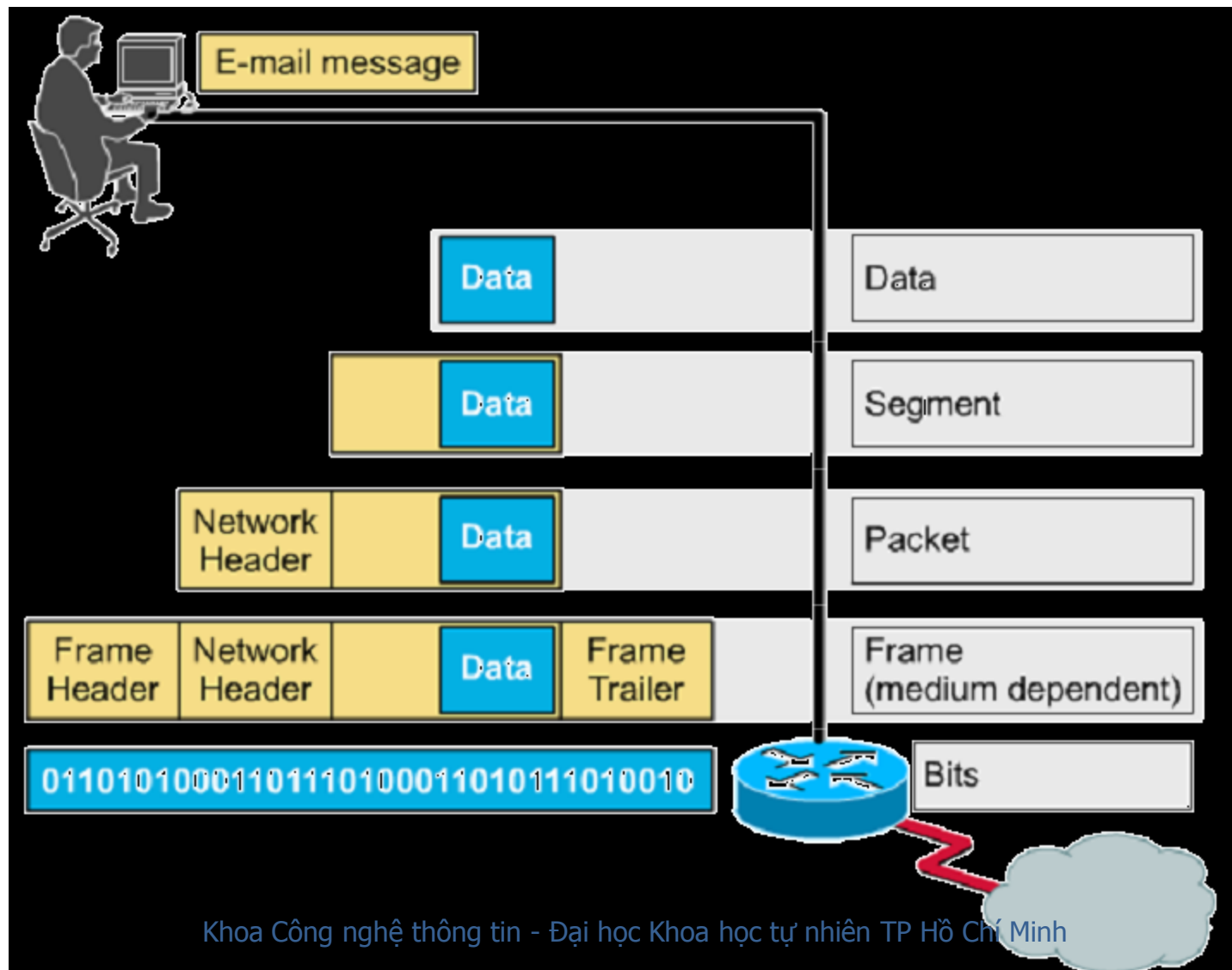
- ❑ Đóng gói dữ liệu = Encapsulation
- ❑ Là quá trình đóng gói dữ liệu với các thông tin của giao thức trước khi chuyển đi



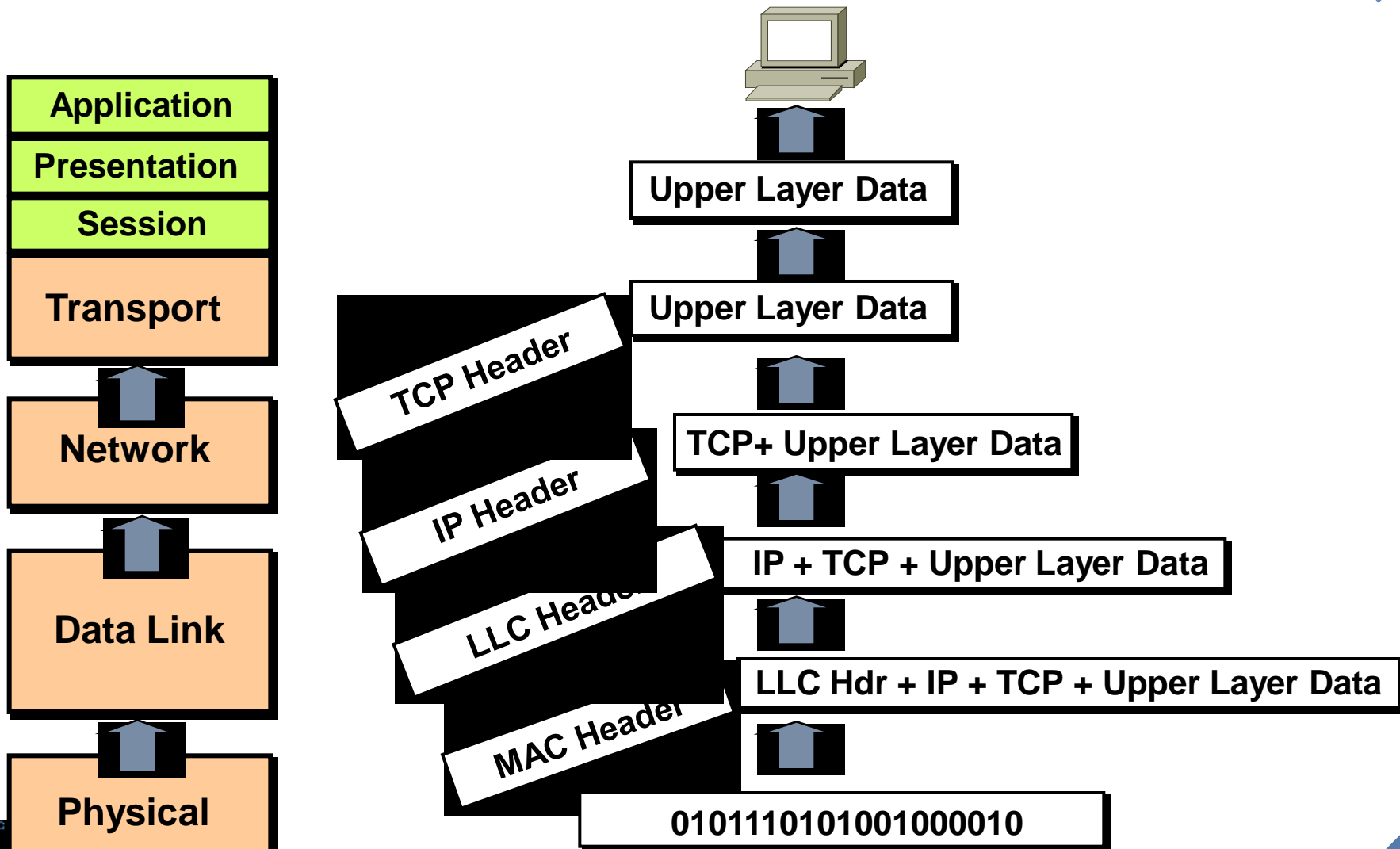
Đóng gói dữ liệu - 2



Đóng gói dữ liệu - 3



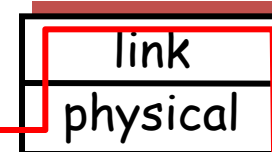
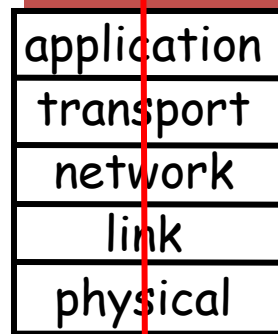
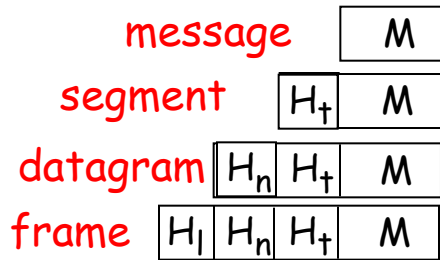
Phân rã



Minh họa

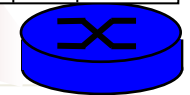
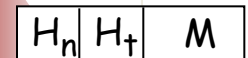
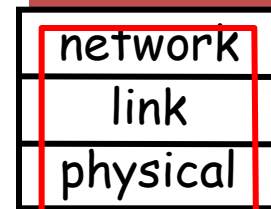
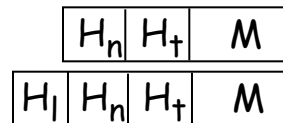
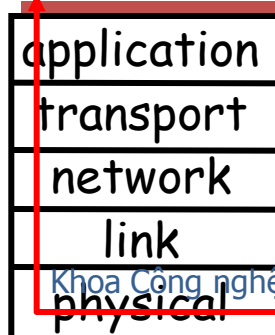
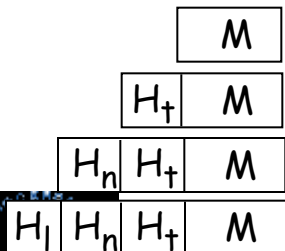


source



switch

destination



router

Tài liệu tham khảo



- ❑ Slide bài giảng của J.F Kurose and K.W. Ross về Computer Networking: A Top Down Approach





Chương 03

Tầng ứng dụng

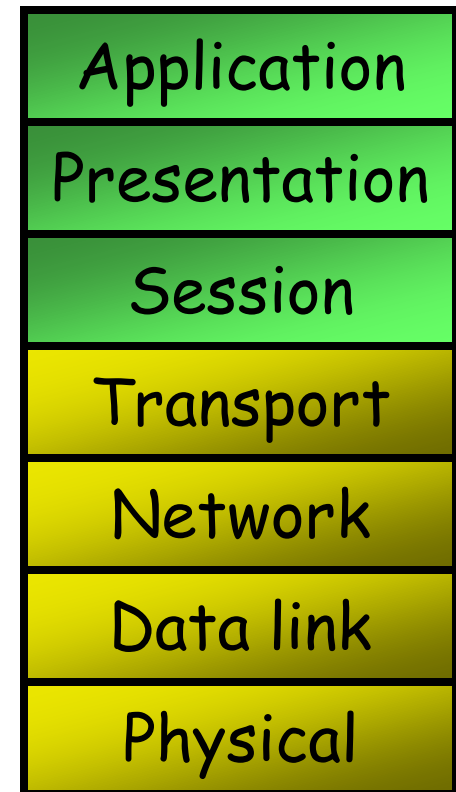
MẠNG MÁY TÍNH

Tháng 09/2011

Mục tiêu



- ❑ Cung cấp các dịch vụ mạng cho người dùng cuối
- ❑ Các ứng dụng mạng phổ biến:
 - E-mail
 - Web
 - Instant Message
 - Telnet, SSH
 - FTP, P2P file sharing
 - Networked Games
 - Video conference



Nội dung



- Các khái niệm
- Một số dịch vụ mạng
- Lập trình ứng dụng



Process - 1

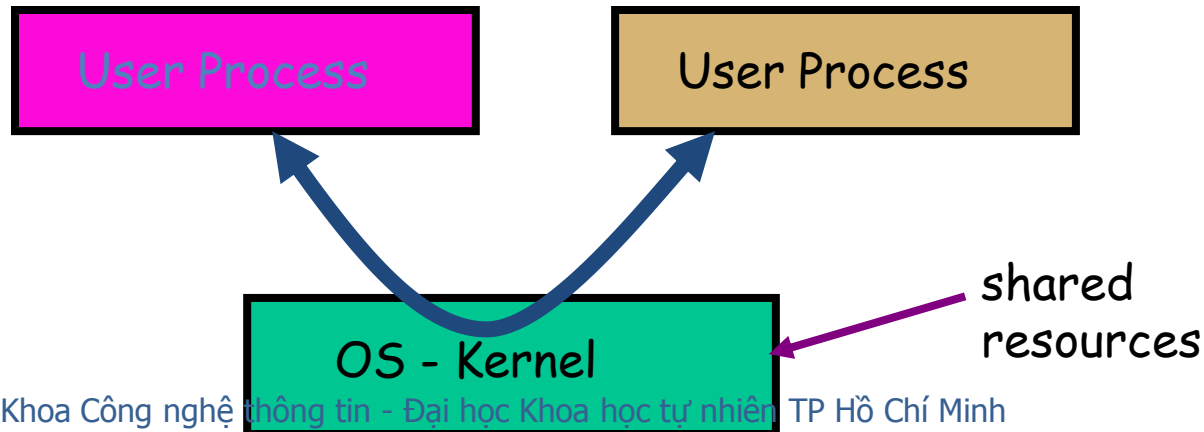


□ Process = tiến trình

- chương trình chạy trên máy
- Nhiều luồng công việc (thread – tiểu trình)

□ Liên lạc giữa các tiến trình:

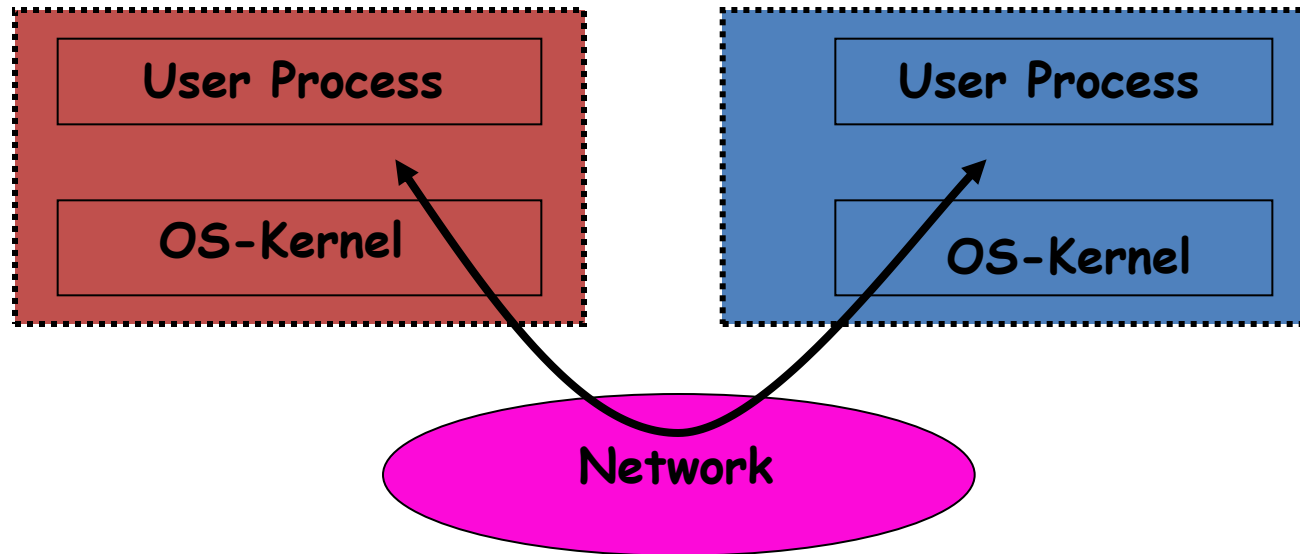
- Trên cùng 1 máy:
 - hệ điều hành
 - Chia sẻ bộ nhớ
 - Truyền thông điệp giữa các tiến trình



Process - 2



- Trên 2 máy khác nhau:
 - truyền dữ liệu qua đường mạng
 - VD: socket, name pipe, ...



Ứng dụng mạng



- ❑ Chạy trên các end-system
- ❑ Liên lạc với nhau qua mạng
- ❑ Kiến trúc:
 - Server-client
 - Peer-to-peer



Server - client



❑ Server:

- Luôn luôn "*sống*"
- Chạy trên 1 địa chỉ cố định
- Nhận và xử lý yêu cầu từ client

❑ Client:

- Liên lạc và gửi yêu cầu cho Server
- Có thể dùng IP "*động*"
- 2 client không thể liên lạc trực tiếp với nhau

❑ VD:

- Web: WebServer (IIS, Apache, ...), web browser (IE, FireFox, ...)
- FTP: FTP Server (ServerU), FTP Client



Peer-to-peer



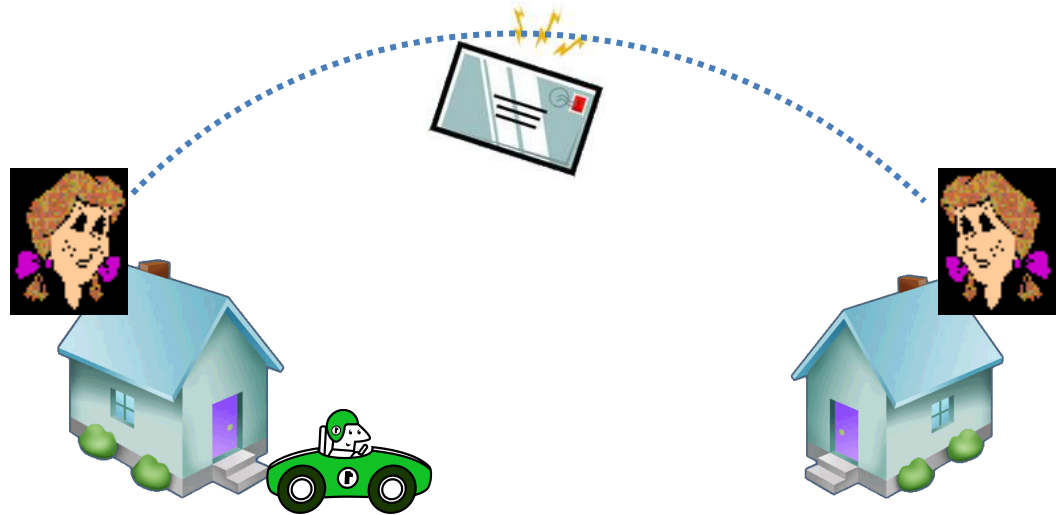
- ứng dụng có cả hai chức năng của server và client
- Các client liên lạc trực tiếp
- Dùng địa chỉ "*động*"
- Quản lý khó
- VD: Skype, Bittorrent



Process - 3

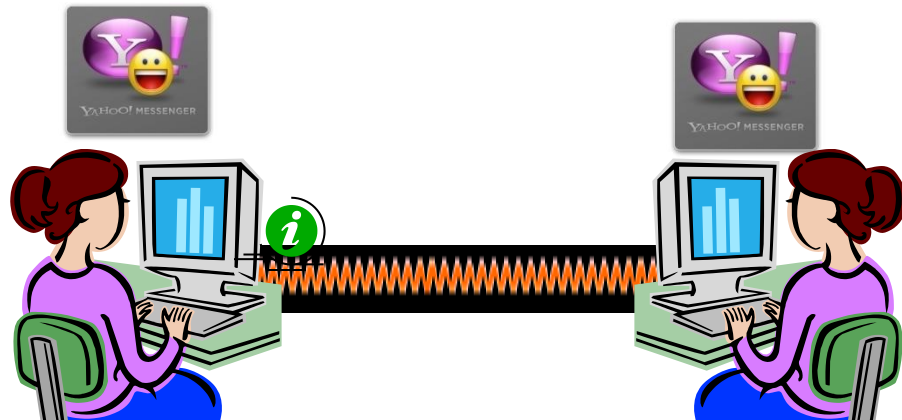
- “Địa chỉ” của một người:

- Địa chỉ nhà
- Tên người



- “Địa chỉ” của tiến trình:

- Địa chỉ IP
- Port:
 - 0..1023: port chuẩn
 - 1024..49151: port cố định, đăng ký trước
 - ...: port linh động



Một số khái niệm khác - 1



- ❑ Giao thức tầng ứng dụng:
 - Do người cài đặt ứng dụng xây dựng
 - VD: HTTP, FTP, ...
- ❑ Những yêu cầu dịch vụ của tầng ứng dụng:
 - Truyền dữ liệu đáng tin cậy
 - Thời gian
 - Băng thông
 - Bảo mật dữ liệu



Một số khái niệm khác - 2

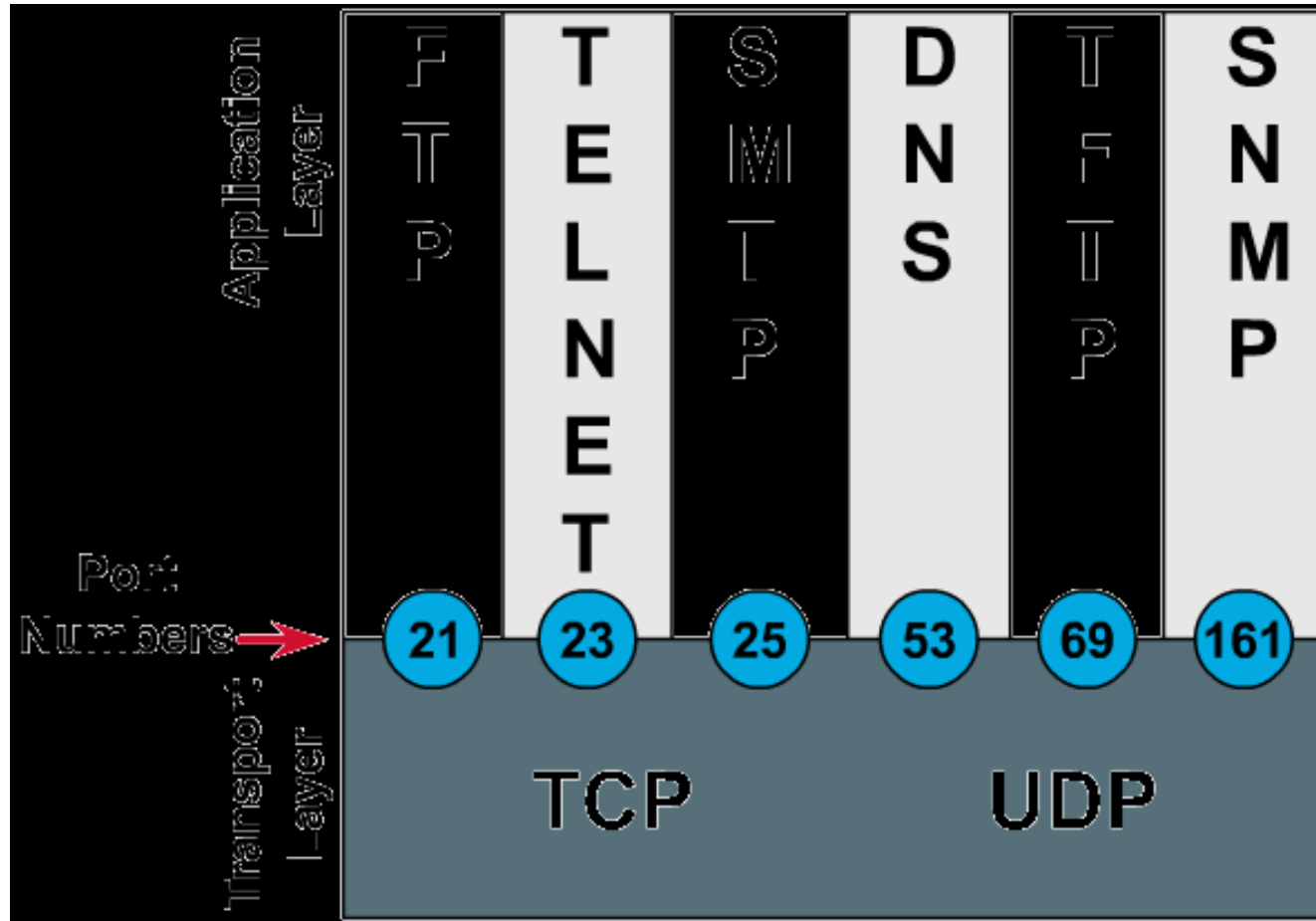


□ Các dịch vụ tầng transport cung cấp:

- TCP service
 - Dịch vụ hướng kết nối
 - truyền dữ liệu đáng tin cậy
- UDP service
 - Dịch vụ hướng không kết nối
 - dữ liệu truyền không đáng tin cậy
 - nhanh hơn TCP



Một số khái niệm khác -3



Tham khảo thêm: <http://www.iana.org/assignments/port-numbers>

<http://www.bctes.com/network-applications-tcp-udp-port-numbers.html>



Nội dung



- Các khái niệm
- Một số dịch vụ mạng
- Lập trình ứng dụng



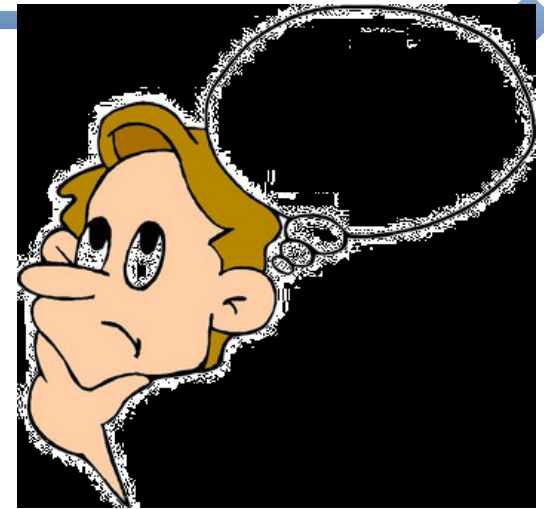
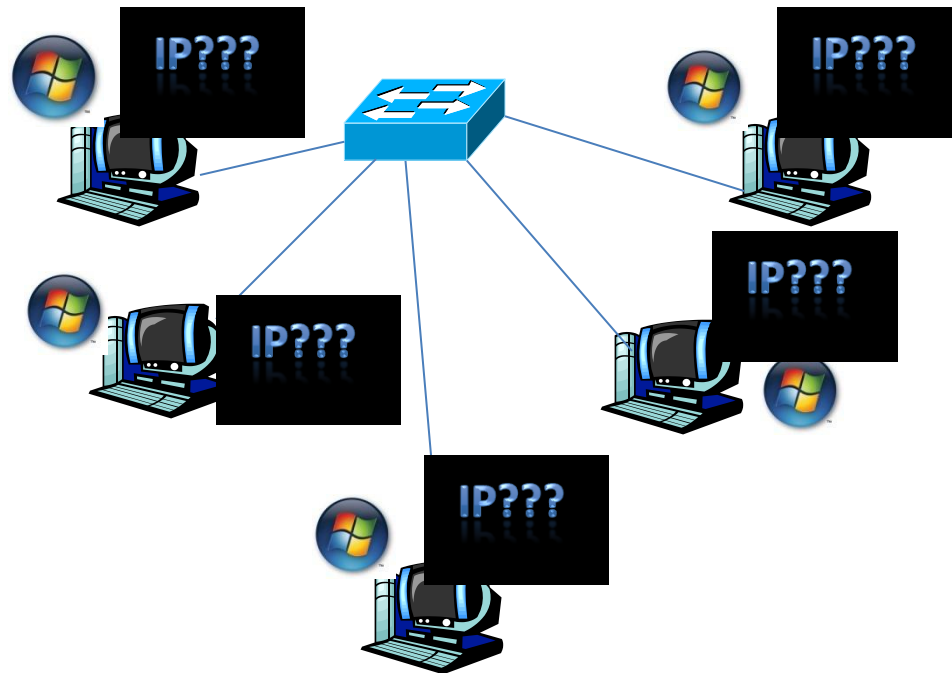
Nội dung



- Các khái niệm
- Một số dịch vụ mạng
 - **DHCP**
 - DNS
- Lập trình ứng dụng



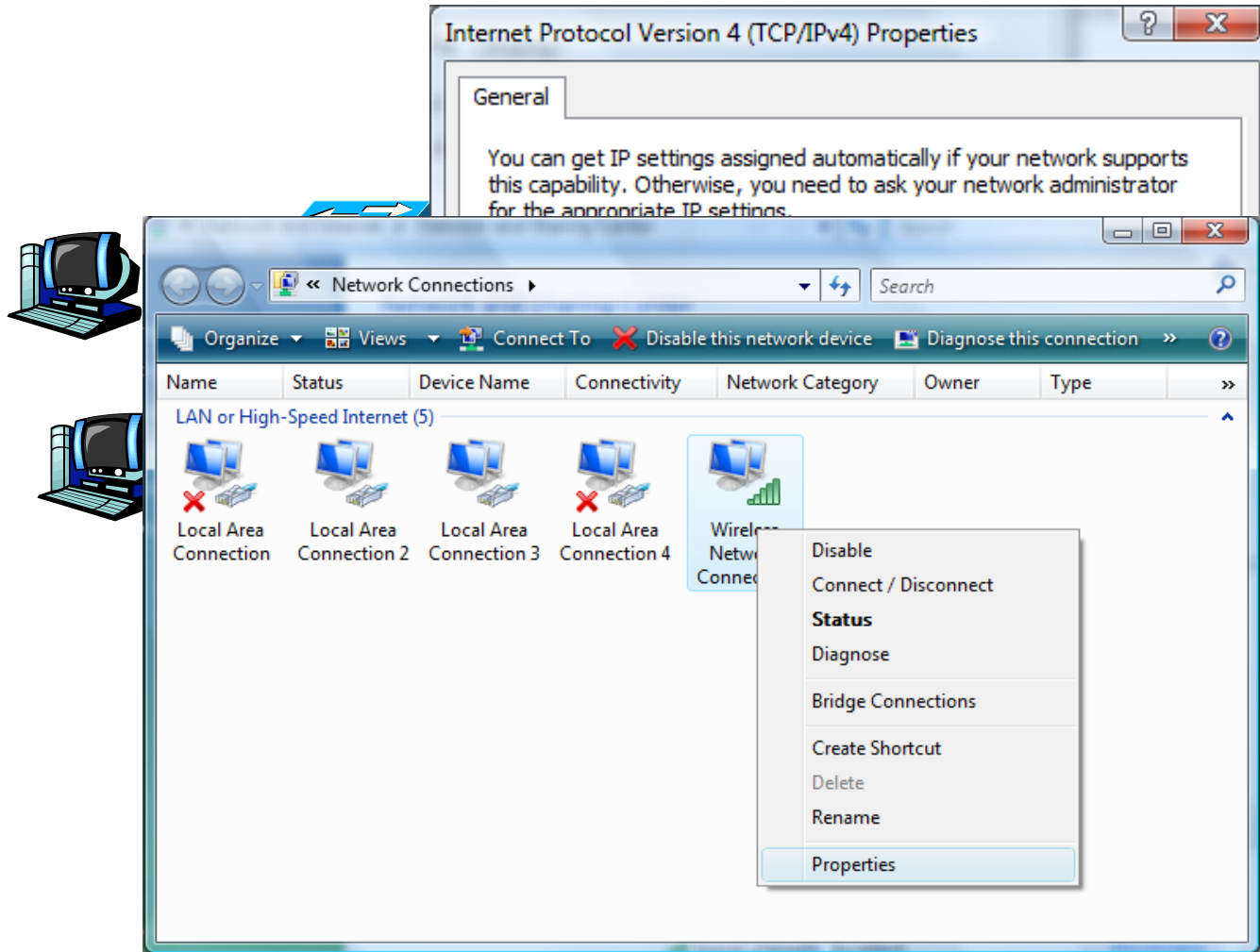
DHCP - Đặt vấn đề



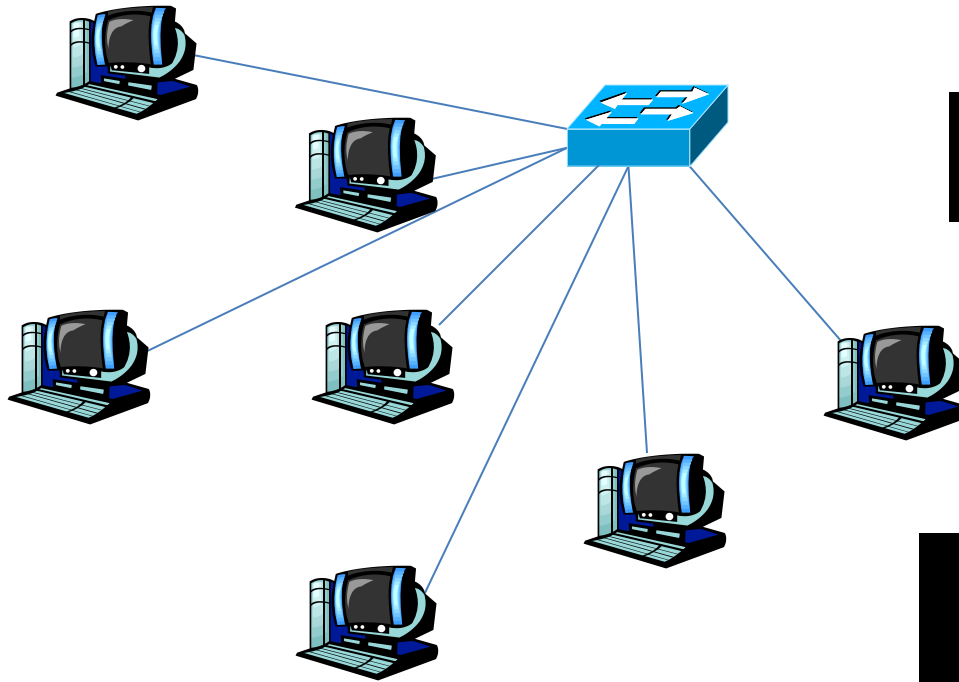
192.168.1.0/24



DHCP - Đặt vấn đề



DHCP - Đặt vấn đề



Số lượng máy lớn?
Không rõ thông tin mạng?



DHCP – giới thiệu



- ❑ DHCP = Dynamic Host Configuration Protocol
 - RFC 1533, 1534, 1541, 1542, 2131
- ❑ Tiền thân: BOOTP
- ❑ Hoạt động tầng ứng dụng
 - Giao thức hoạt động ở tầng Transport: UDP
- ❑ Chức năng: cấp phát địa chỉ IP động
- ❑ Mô hình Client - Server:
 - Server
 - Port: 67
 - Cung cấp địa thông tin cấu hình TCP/IP cho các client
 - Client:
 - Port: 68
 - Yêu cầu server cấp thông tin cấu hình TCP/IP



DHCP - Mô hình hoạt động - 1



MAC đã biết
IP chưa xác định

IP Address, subnet mas
Lease time
DHCP sever IP Address

Default Gateway
DNS Server

khảo: <http://www.ietf.org/rfc/rfc2131.txt>

Khoa Công nghệ thông tin - Đại học Khoa học tự nhiên TP Hồ Chí Minh

DHCP - Mô hình hoạt động - 2



□ Xin cấp mới:

- **Discover:** client tìm DHCP Server
- **Offer:** DHCP gợi ý một địa chỉ IP
- **Request:** Client yêu cầu cấp 1 địa chỉ IP
- **Ack:** Server xác nhận đồng ý và giải phóng địa chỉ IP
- **Nak:** Server từ chối địa chỉ IP mà client yêu cầu

□ Xin cấp lại:

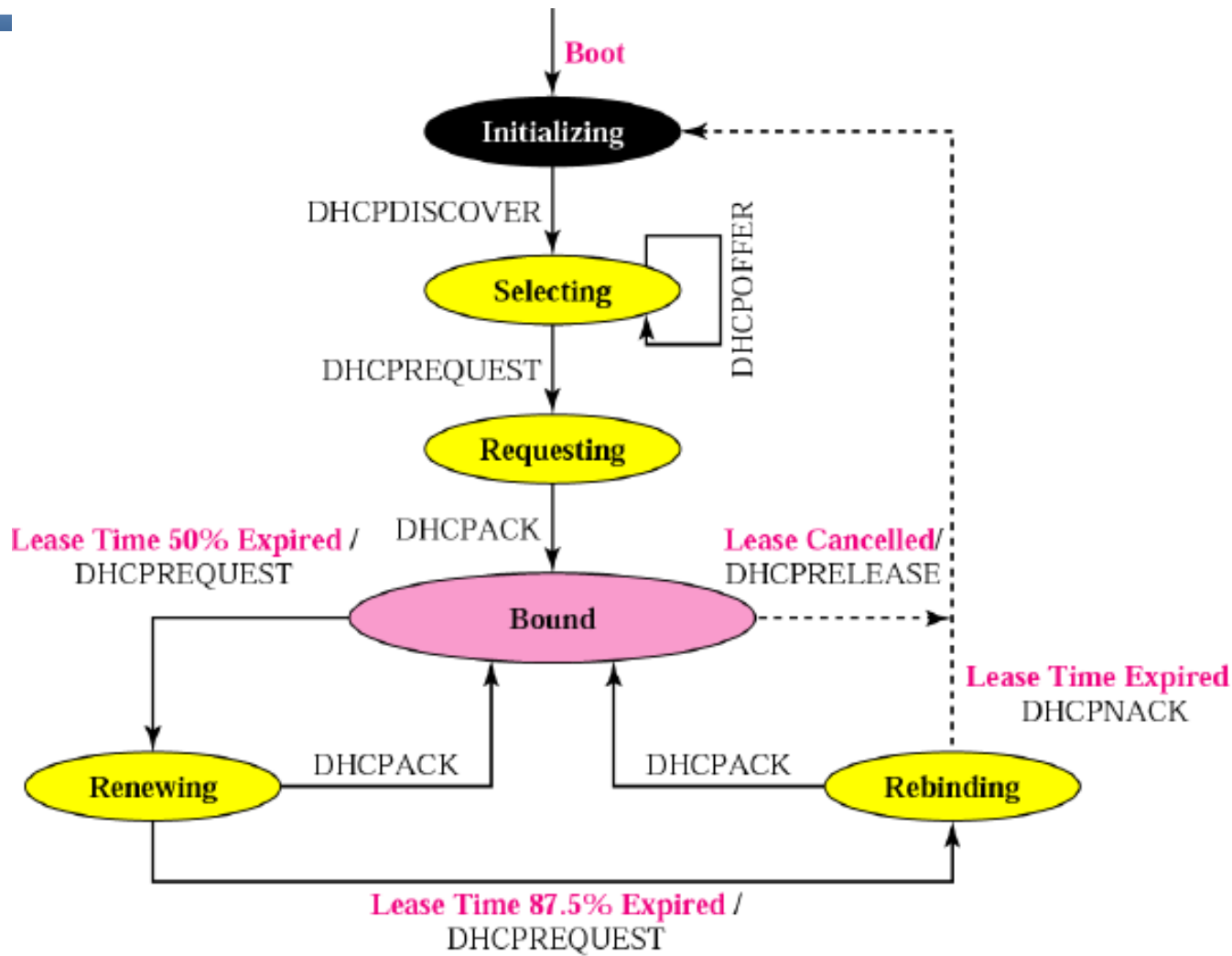
- **Request**
- **Ack/ Nak**

□ Huỷ thông tin được cấp:

- **Release**



DHCP – sơ đồ hoạt động



DHCP – Format message



op (1)	htype (1)	hlen (1)	hops (1)
xid (4)			
secs (2)		flags (2)	
ciaddr (4)			
yiaddr (4)			
siaddr (4)			
giaddr (4)			
chaddr (16)			
sname (64)			
file (128)			
options (312)			



DHCP – Format message



Op	1	Loại thông điệp
Htype	1	Loại địa chỉ vật lý
Hlen	1	Kích thước 1 địa chỉ vật lý
Hops	1	Client set là 0, dùng cho relay agent
Xid	4	Transaction ID,
Secs	2	Được điền bởi client
Flags	1	Cho biết gói tin này có phải là gói tin broadcast không
Ciaddr	4	Địa chỉ của Client, dùng trong các trường hợp renew,..
Yiaddr	4	Địa chỉ của “đối tác”
Siaddr	4	Địa chỉ của next server để “mời” và được dùng trong các thông điệp DHCPOFFER, DHCPACK
Giaddr	4	Địa chỉ của relay agent
Chaddr	16	Địa chỉ vật lý của client



DHCP – lệnh console



- Kiểm tra thông tin cấu hình IP: `Ipconfig /all`
- Xin cấp một IP mới: `Ipconfig /renew`
- Trả địa chỉ IP đang dùng: `Ipconfig /release`



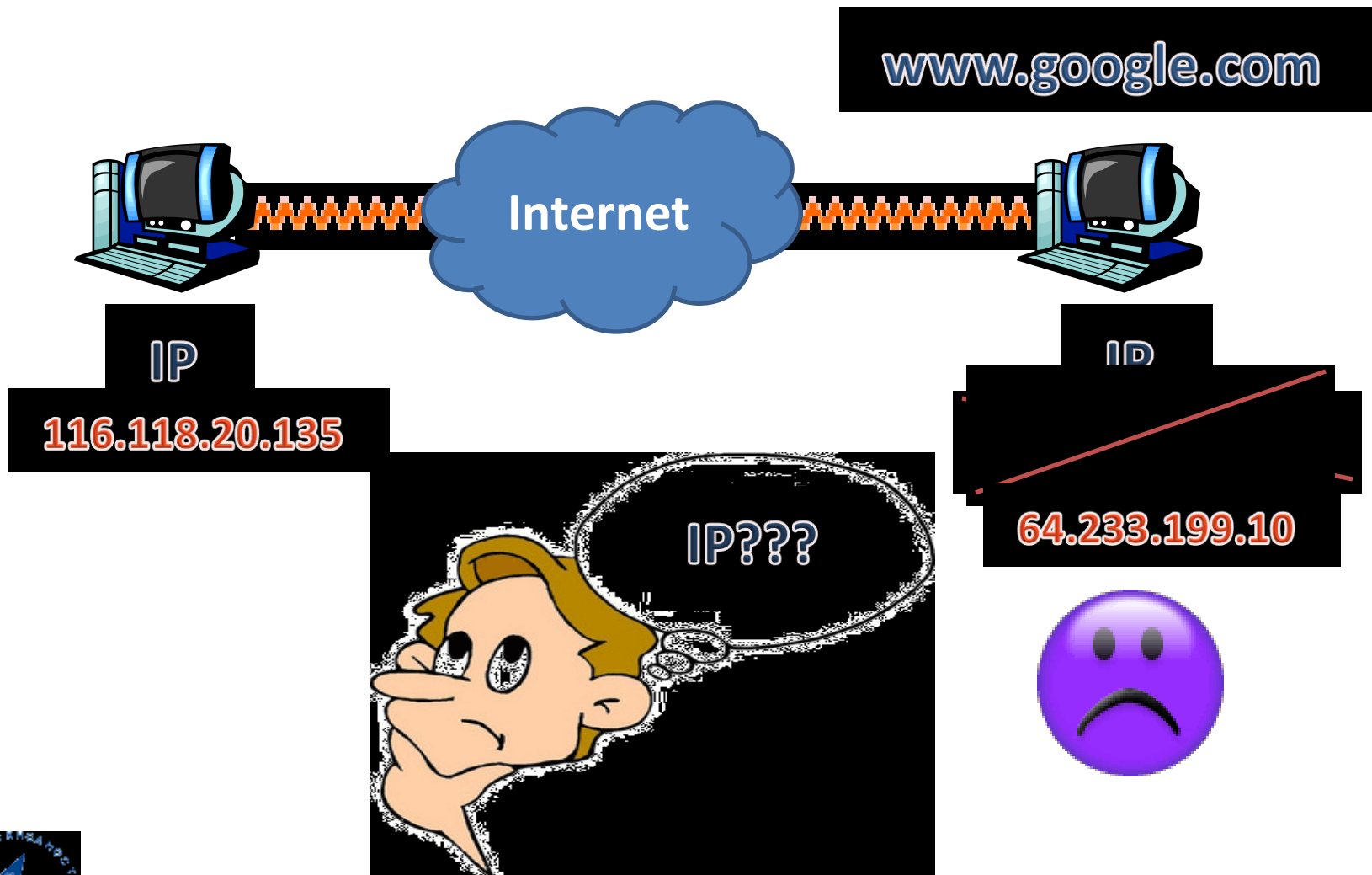
Nội dung



- Các khái niệm
- Một số dịch vụ mạng
 - DHCP
 - **DNS**
- Lập trình ứng dụng



DNS - Đặt vấn đề - 1



DNS - Đặt vấn đề - 2



192.31.7.130 CISCO.COM

204.71.177.35 MICROSOFT.COM

152.162.249. MICROSOFT.COM

198. MICROSOFT.COM

207.46.131. MICROSOFT.COM

192.233.80.9 NOVELL.COM

Domain Name ⇔ IP



DNS – giới thiệu



- ❑ DNS = Domain Name System
- ❑ Rfc 1034, 1035
- ❑ Chức năng
 - Dùng phân giải tên miền thành địa chỉ IP và ngược lại
- ❑ Lịch sử:
 - Đầu tiên, lưu trữ bằng file hosts
 - Ngày nay, lưu trữ bằng 1 CSDL phân tán
 - Tạo thành cây domain
 - Mỗi node
 - Name Server (NS)
 - Có một tên miền (domain name)
 - Có thể có sub domain



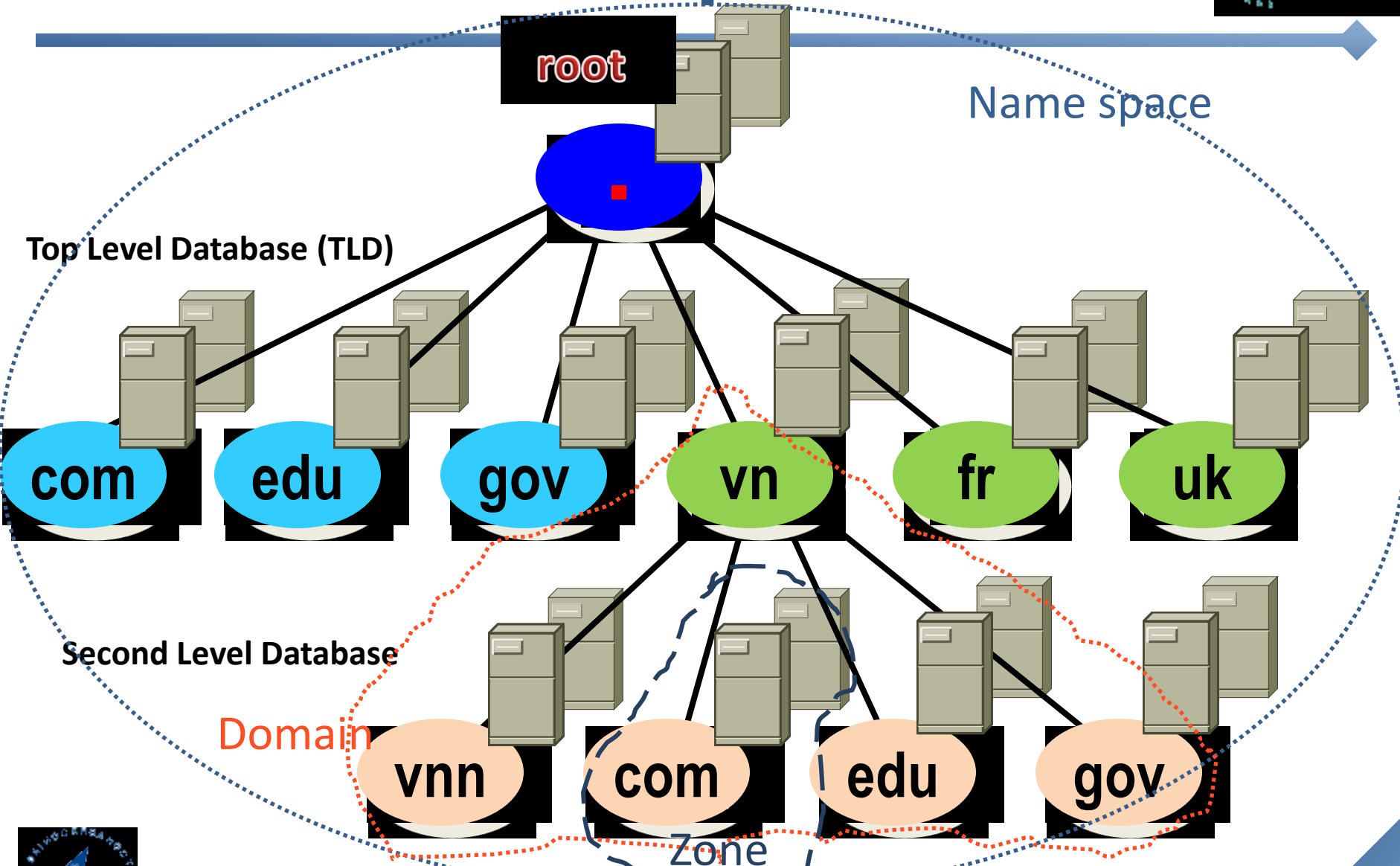
DNS – mô hình hoạt động



- ❑ Hoạt động tầng Application
- ❑ Tầng Transport:
 - UDP: truy vấn (query) – port 53
 - TCP: cập nhật thông tin (zone transfer)
- ❑ Mô hình Client – Server
 - Server:
 - Primary NS
 - Chứa thông tin về một zone
 - Secondary NS
 - Backup của primary NS
 - Client - Resolver
 - Port linh động



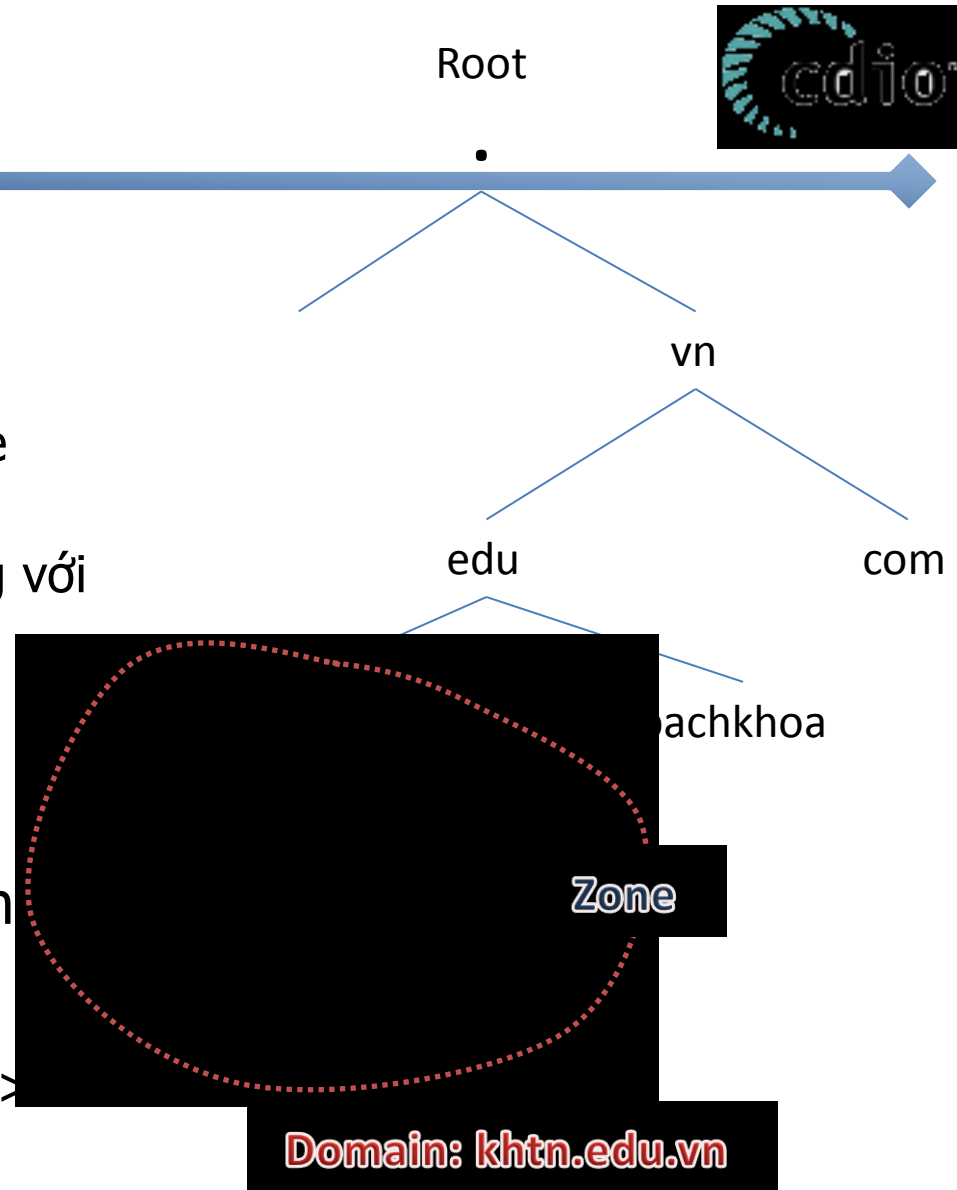
Internet domain space



DNS – thuật ngữ



- ❑ Name space:
 - Tập các tên miền trên Internet
- ❑ Domain:
 - là một nhánh trong name space
- ❑ Zone:
 - Là một vùng domain tương ứng với “biên” quản lý trong DNS
 - Phân loại:
 - Primary zone
 - Secondary zone
 - Stub zone
- ❑ FQDN (Fully Qualified Domain Name)
 - Tên đầy đủ
 - <hostname> + <domain name> + “.”
 - VD: www.cntt.khtn.edu.vn.



DNS – Internet domain space – thống kê

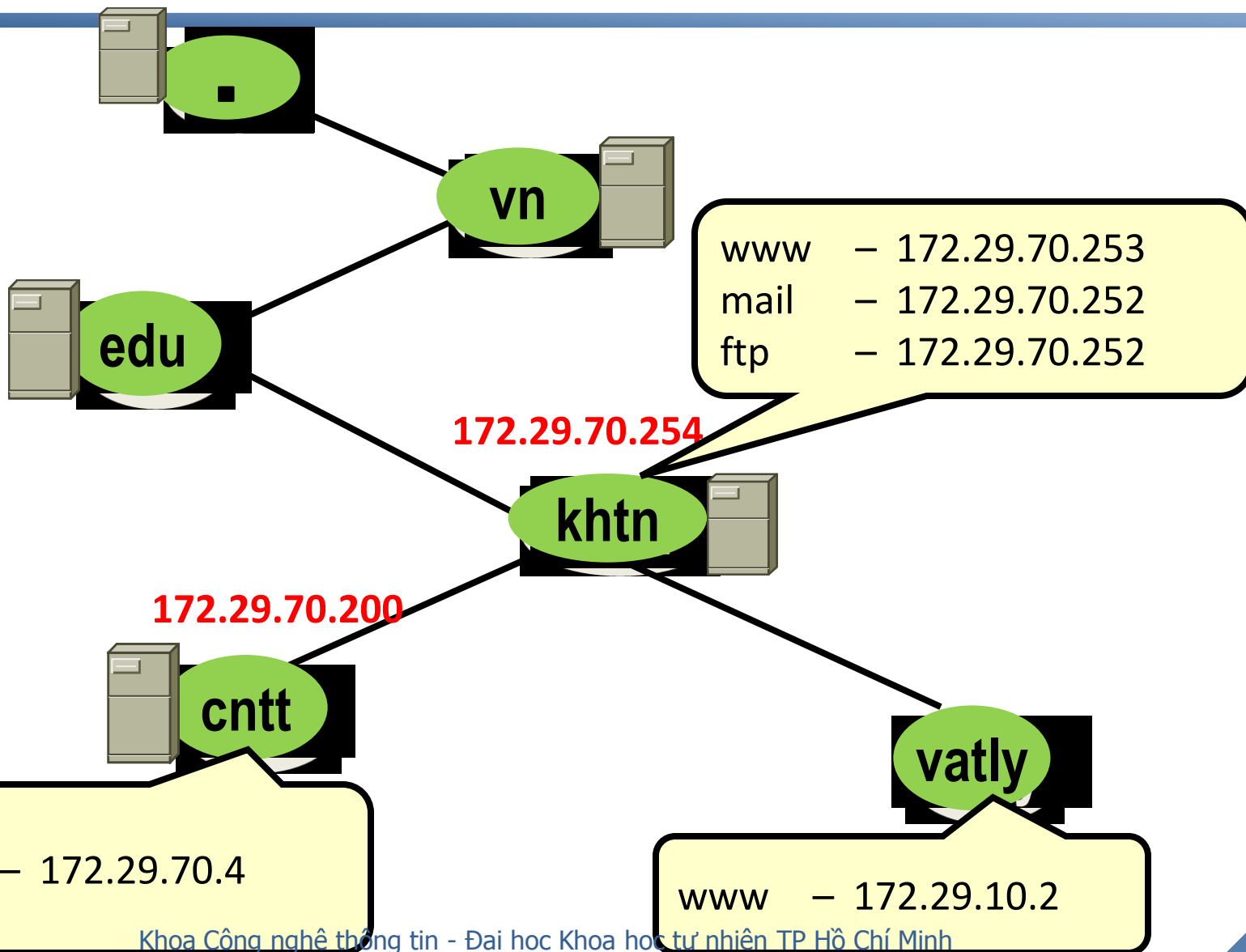


		Daily Changes(last 24hrs)			
Active	Deleted	New	Expired	Transferred	TLD
81,325,776	303,600,735	60,995	54,069	90,327	.Com
12,348,084	31,822,042	8,847	8,647	9,640	.Net
7,710,052	20,100,327	4,128	4,424	4,555	.Org
5,232,582	8,307,651	7,874	8,528	7,326	.Info
2,044,242	1,801,091	896	1,485	1,223	.Biz
1,641,930	1,406,153	1,979	1,451	1,668	.Us
110,434,500	367,699,576	84,719	78,604	114,739	Total

Last Updated : 2009-07-12



DNS – ví dụ



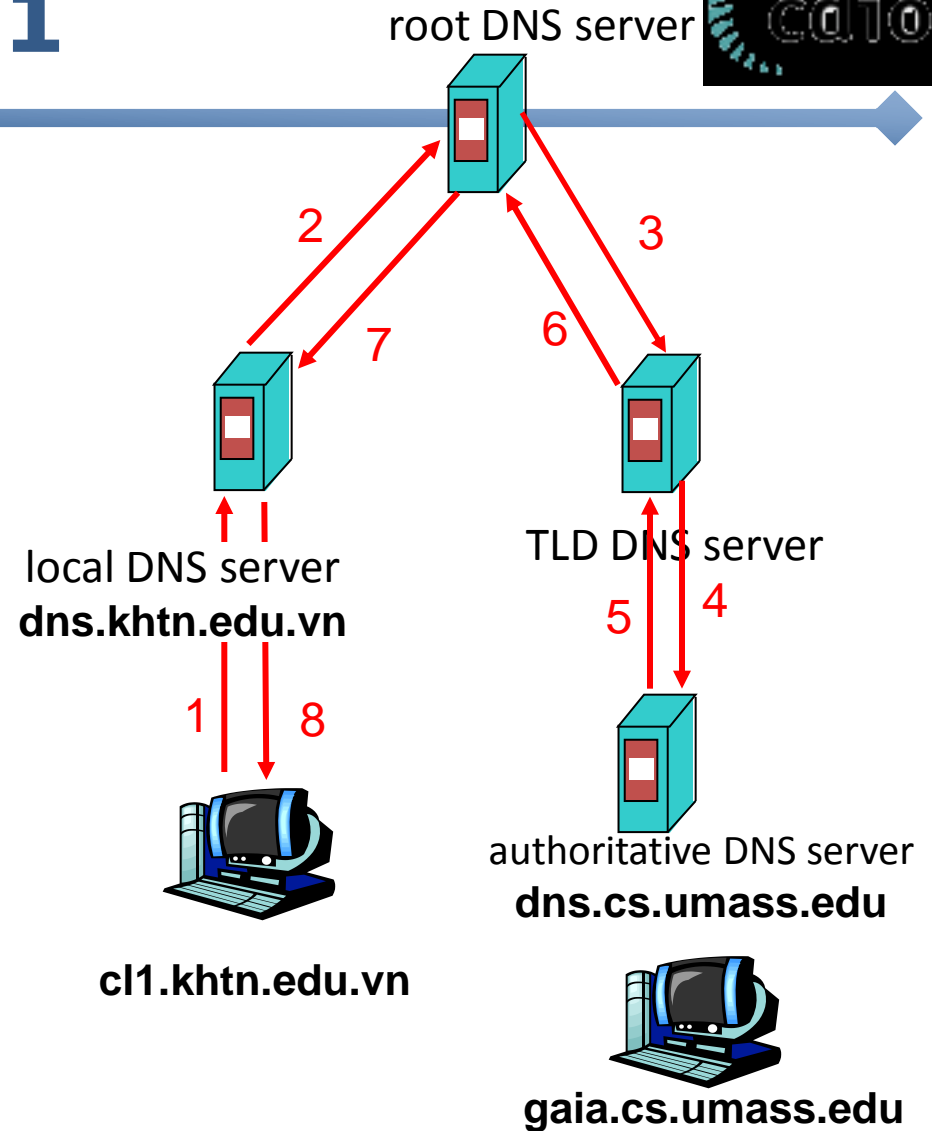
- Lưu dữ liệu dưới dạng các resource record – RR (name, value, type, ttl)
 - SOA: thông tin cho toàn bộ 1 zone
 - MX: thông tin của server nhận mail của miền
 - NS: thông tin các name server quản lý zone
 - Name: tên miền
 - Value: địa chỉ NS của miền
 - A: dùng để phân giải tên máy thành địa chỉ IP
 - ❖ Name: hostname
 - ❖ Value: IP address
 - CNAME: lưu tên phụ của 1 máy
 - Name: tên alias
 - Value: tên thật
 - PTR: dùng để phân giải địa chỉ IP thành tên máy
 - Name: IP addr.
 - Value: hostname

DNS – phân giải - 1



□ Đệ qui (recursive query)

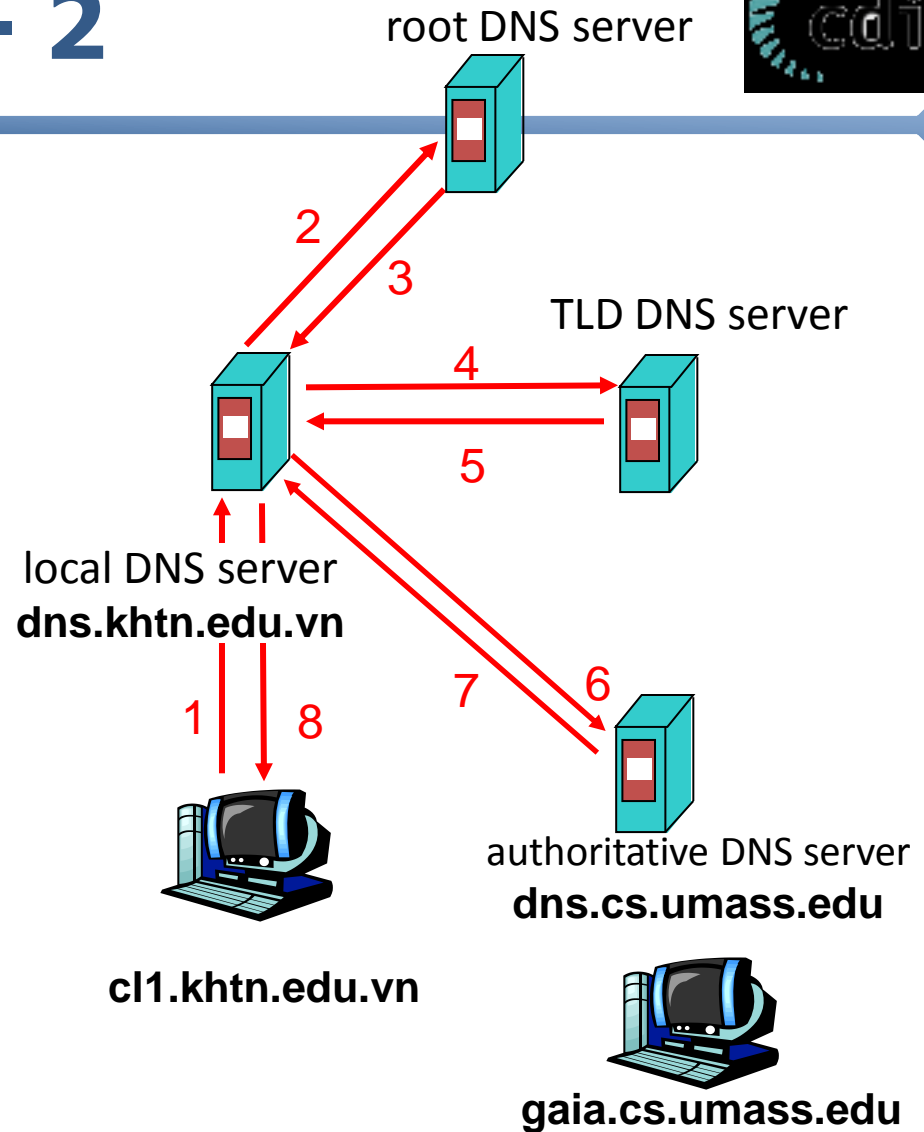
- Server nhận câu truy vấn phải trả lời thông tin cuối cùng: phân giải được hay không?
- VD: máy `cl1.khtn.edu.vn` truy vấn tên miền `gaia.cs.umass.edu`



DNS – phân giải - 2



- ❑ Tuần tự (iterated query)
 - Server nhận câu truy vấn không phải trả lời thông tin cuối cùng, chỉ trả lời thông tin về NS “gần nhất”
 - VD: máy `cl1.khntn.edu.vn` truy vấn tên miền `gaia.cs.umass.edu`

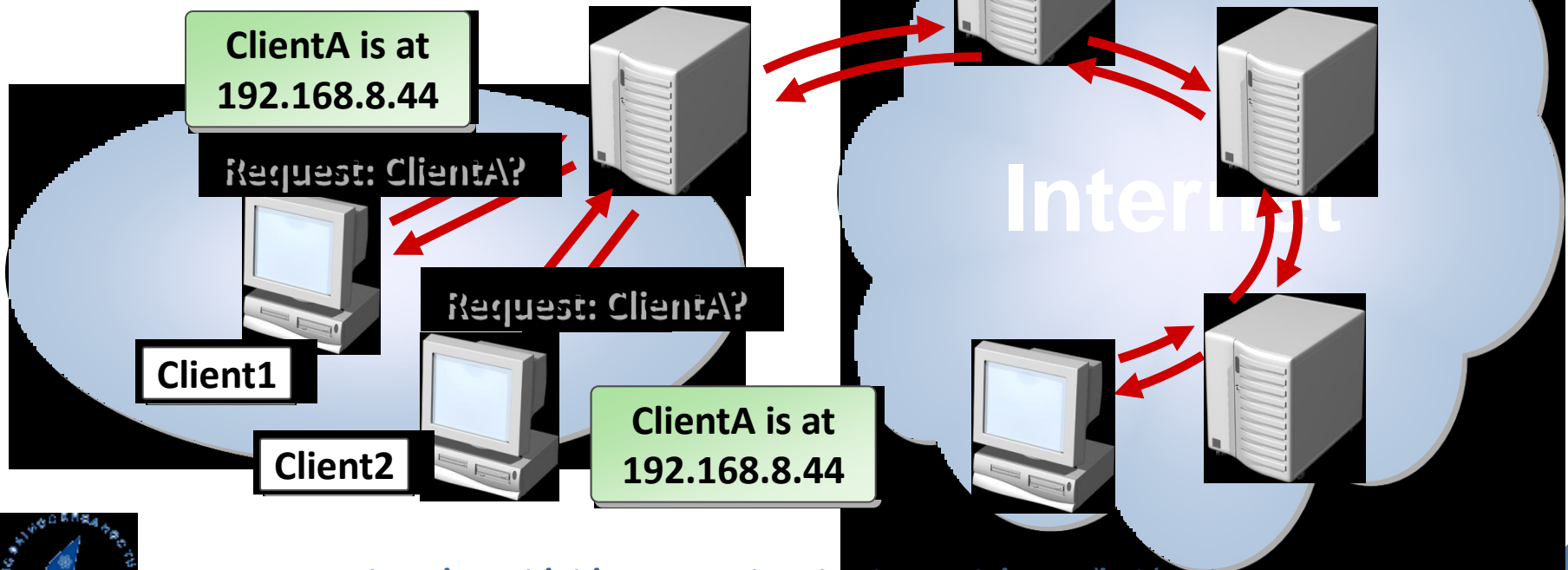


DNS – Phân giải - caching

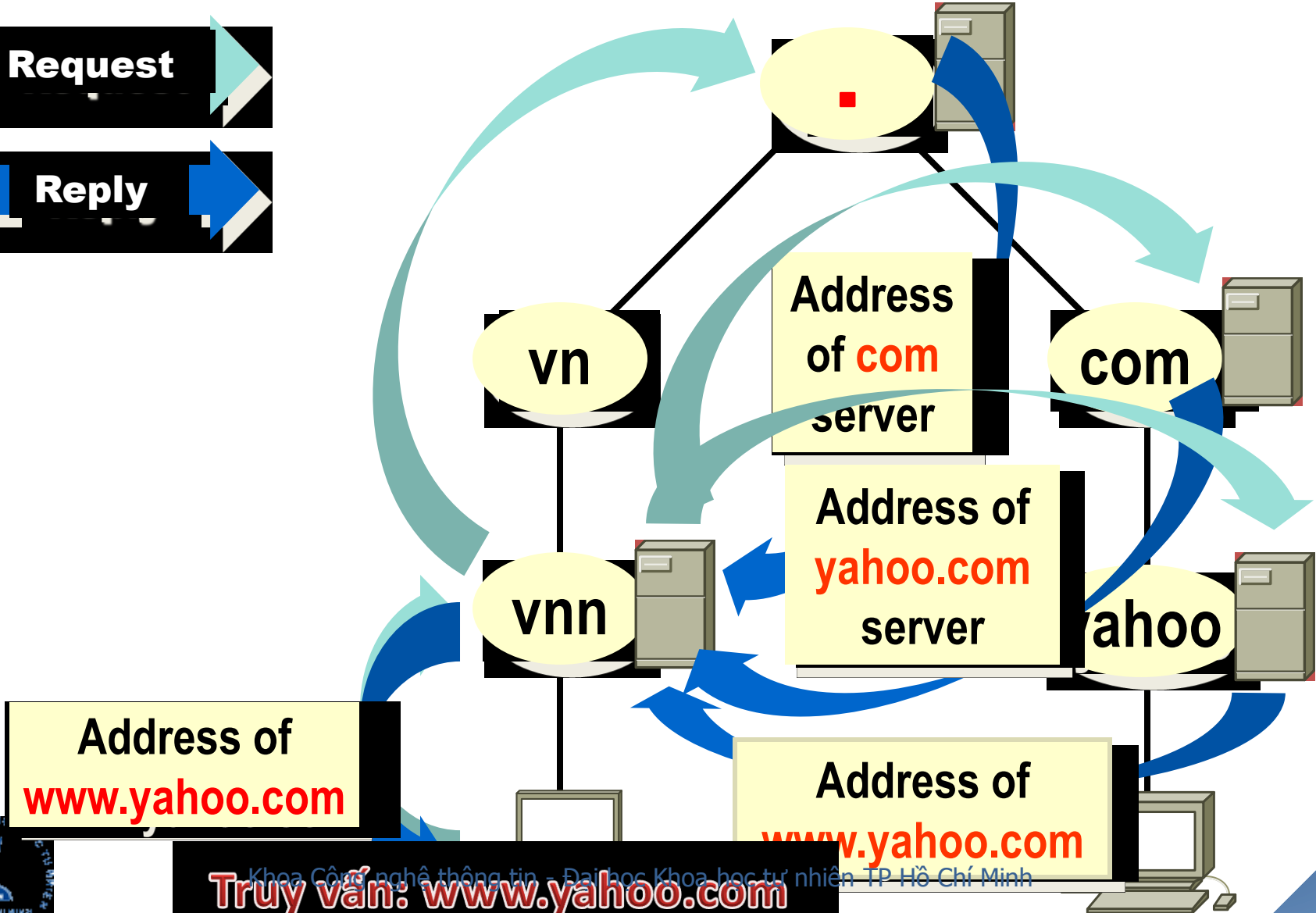
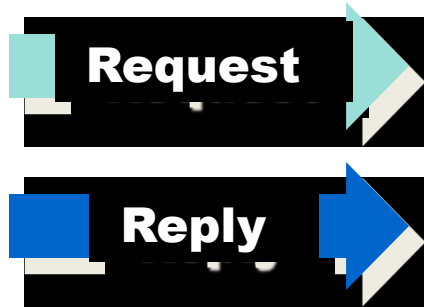


- Lưu tạm kết quả đã truy vấn trong cache

Caching Table		
Host Name	IP Address	TTL
clientA.contoso.msft.	192.168.8.44	28 seconds



DNS – phân giải - ví dụ



Nội dung



- ❑ Các khái niệm
- ❑ Một số dịch vụ mạng
- ❑ Lập trình ứng dụng
 - TCP
 - UDP



□ Socket

- “Cánh cửa” giữa ứng dụng và giao thức tầng transport (TCP, UDP)
- cung cấp interface để lập trình mạng tại tầng Transport
- Một socket là một end-point của một liên kết giữa hai ứng dụng

□ Windows Socket Application Programming Interface (Winsock API)

- thư viện các hàm socket
- xây dựng các ứng dụng mạng trên nền TCP/IP

Lập trình ứng dụng mạng



1. Xác định kiến trúc mạng: Client – Server, Peer-to-Peer
2. Giao thức sử dụng tầng Transport: TCP, UDP
3. Các port sử dụng ở Server và Client
4. Giao thức tầng ứng dụng khi trao đổi dữ liệu giữa hai end-host
5. Lập trình



Lập trình ứng dụng – TCP - 1



- **Giai đoạn 1**: Server tạo Socket và lắng nghe yêu cầu kết nối tại PORT

SERVER

CLIENT

Tạo socket để lắng nghe kết nối

`socket()`



Đăng ký tên cho socket

`bind()`

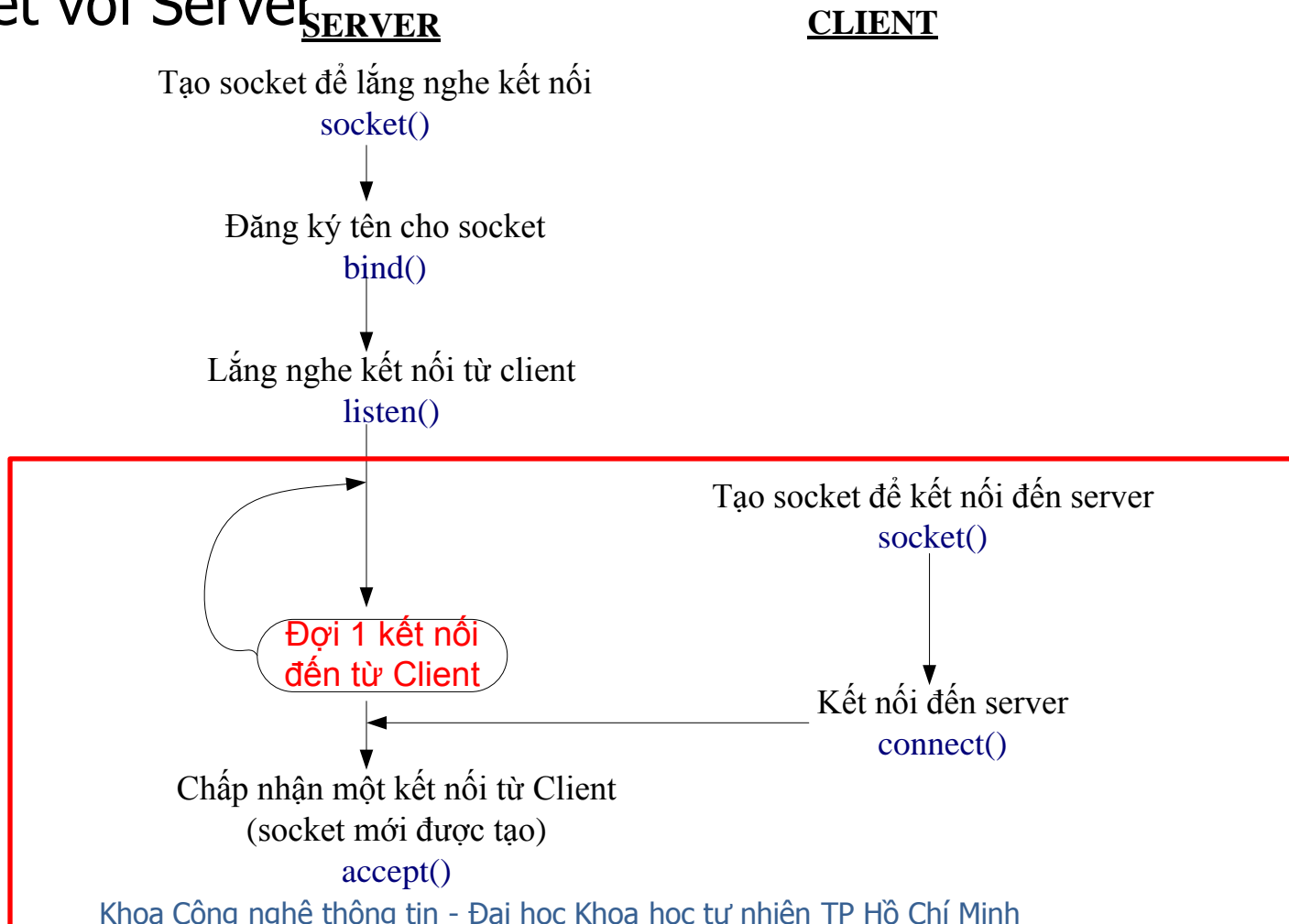


Lắng nghe kết nối từ client

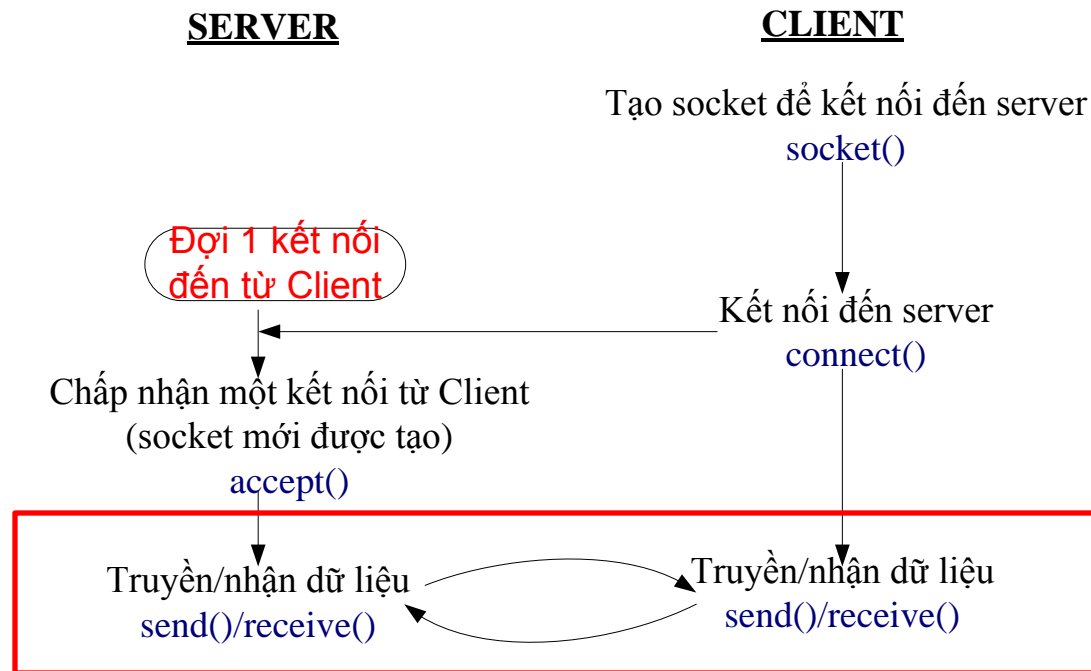
`listen()`



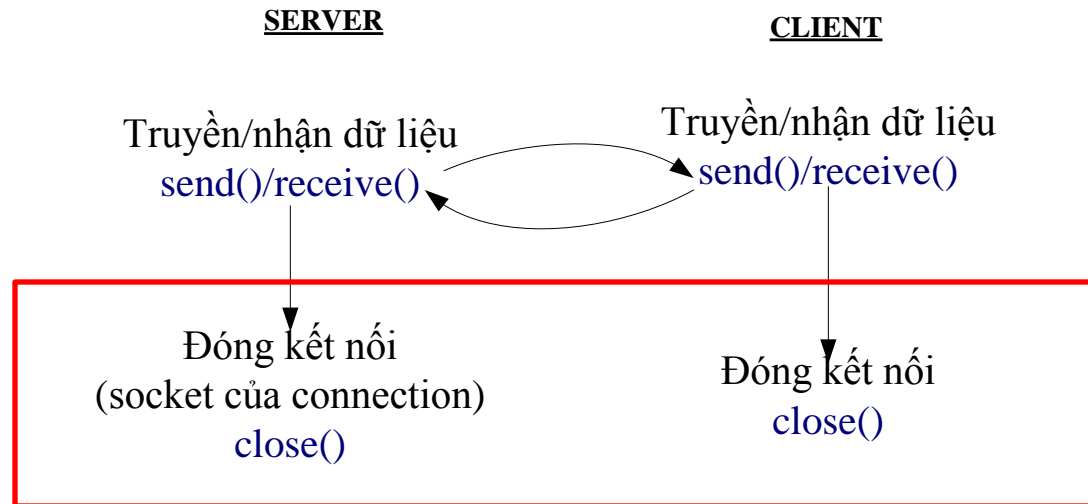
- **Giai đoạn 2**: Client tạo Socket, yêu cầu thiết lập một nối kết với Server



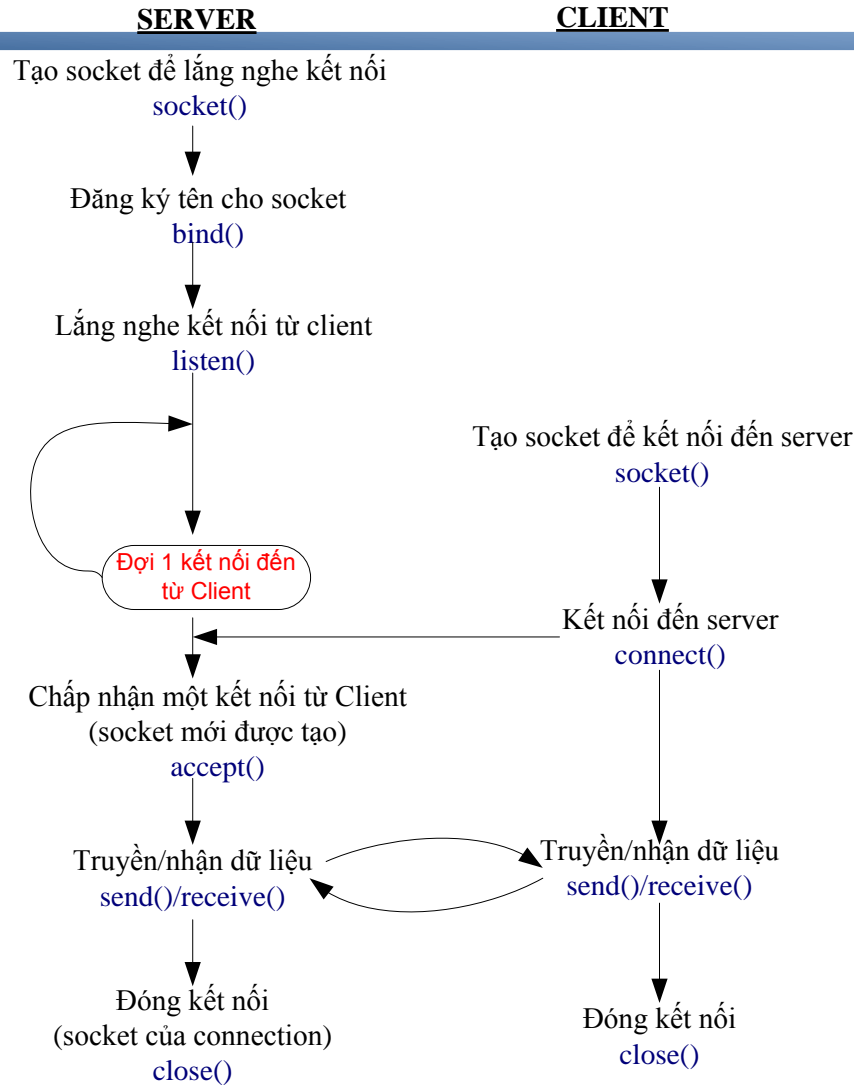
- **Giai đoạn 3**: Trao đổi thông tin giữa Client và Server



- **Giai đoạn 4**: Kết thúc phiên làm việc



TCP - 5

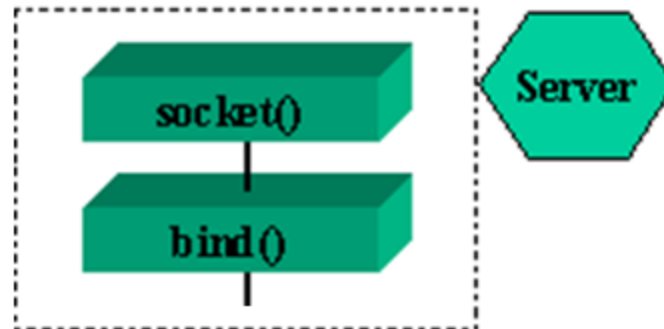


Lập trình ứng dụng – UDP - 1



□ Mô hình UDP

- **Giai đoạn 1**: Server tạo Socket tại PORT



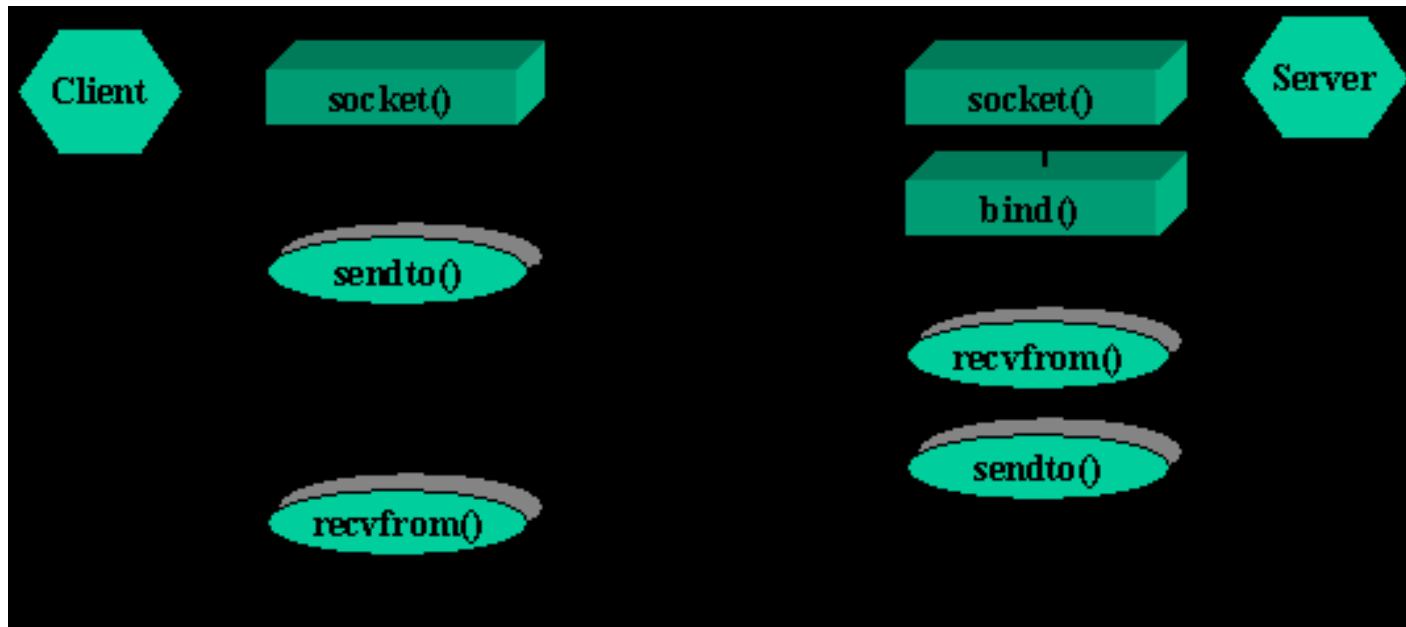
- **Giai đoạn 2**: Client tạo Socket



UDP - 3



- **Giai đoạn 3**: Trao đổi thông tin giữa Client và Server



Tài liệu tham khảo



- ❑ Slide của J.F Kurose and K.W. Ross về Computer Networking: A Top Down Approach





Chương 03

Tăng vận chuyển

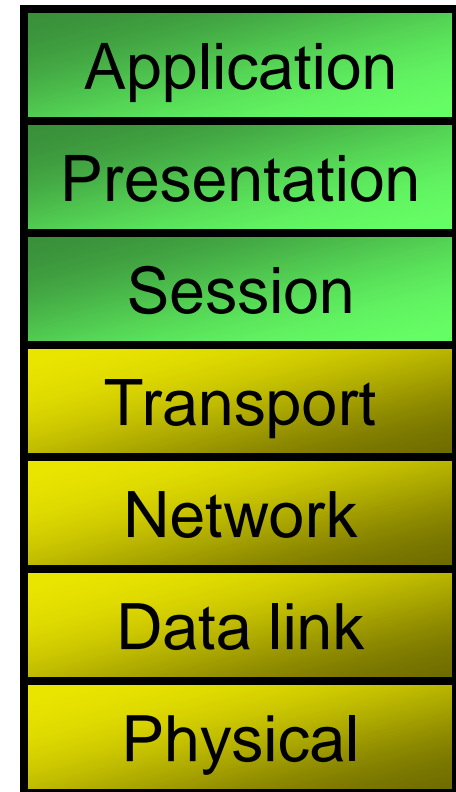
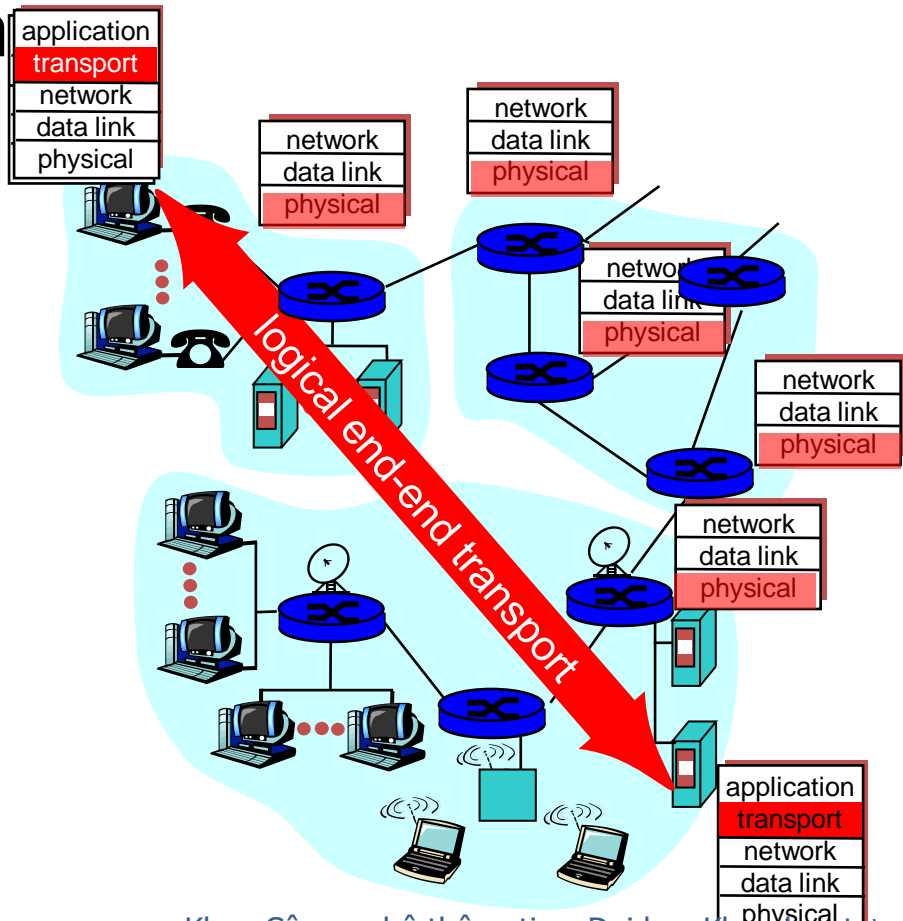
MẠNG MÁY TÍNH

Tháng 09/2011

Chức năng - 1



- Cung cấp kênh truyền dữ liệu ở mức logic giữa 2 tiến trình trên 2 máy



Nội dung



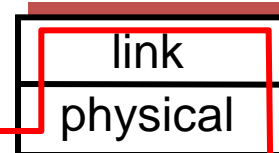
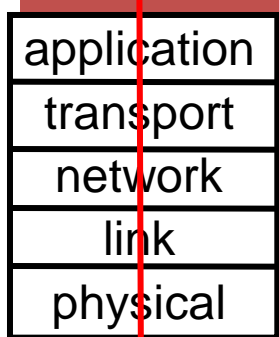
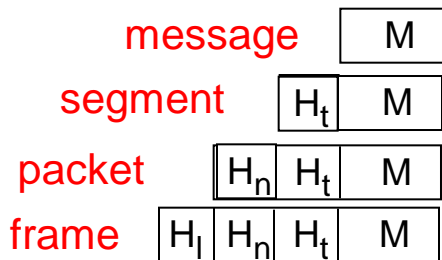
- Giới thiệu
- Nguyên tắc truyền dữ liệu đáng tin cậy
- Giao thức TCP
- Giao thức UDP



Nhắc lại

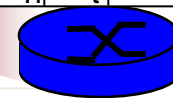
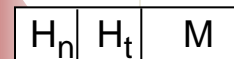
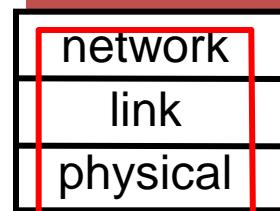
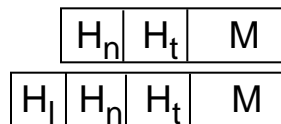
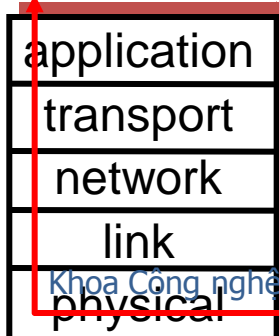
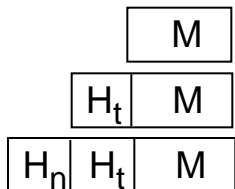


source



switch

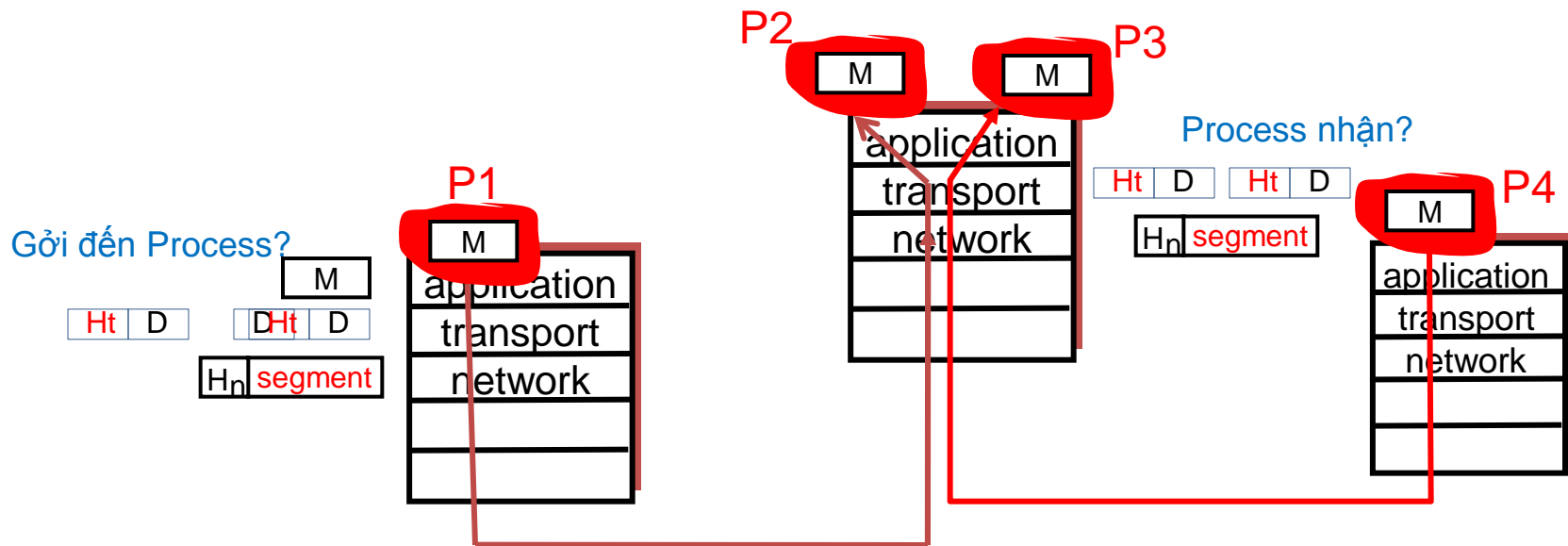
destination



router



tầng vận chuyển - 1



tầng vận chuyển - 2



- ❑ Thực thi ở end-system
- ❑ Bên gửi: thực hiện **Dồn kênh**
 - Nhận dữ liệu từ tầng ứng dụng (từ các socket)
 - Phân đoạn thông điệp ở tầng ứng dụng thành các **segment**
 - Dán nhãn dữ liệu: đóng gói theo giao thức tại tầng Transport
 - Chuyển các segment xuống tầng mạng (network layer)
- ❑ Bên nhận: thực hiện **Phân kênh**
 - Nhận các segment từ tầng mạng
 - Phân rã các segment thành thông điệp tầng ứng dụng
 - Chuyển thông điệp lên tầng ứng dụng (đến socket tương ứng)



Hỗ trợ

- Truyền dữ liệu đáng tin cậy
 - Điều khiển luồng
 - Điều khiển tắc nghẽn
 - Thiết lập và duy trì kết nối
- Truyền dữ liệu không đáng tin cậy
 - Nỗ lực gửi dữ liệu hiệu quả nhất

Không hỗ trợ

- Đảm bảo thời gian trễ
- Đảm bảo băng thông

Dồn kênh – Phân kênh - 1



❑ Dồn kênh (Multiplexing):

- Thực hiện tại bên gửi
- Thu thập dữ liệu từ các socket
- dán nhãn dữ liệu với 1 header

❑ Phân kênh (Demultiplexing):

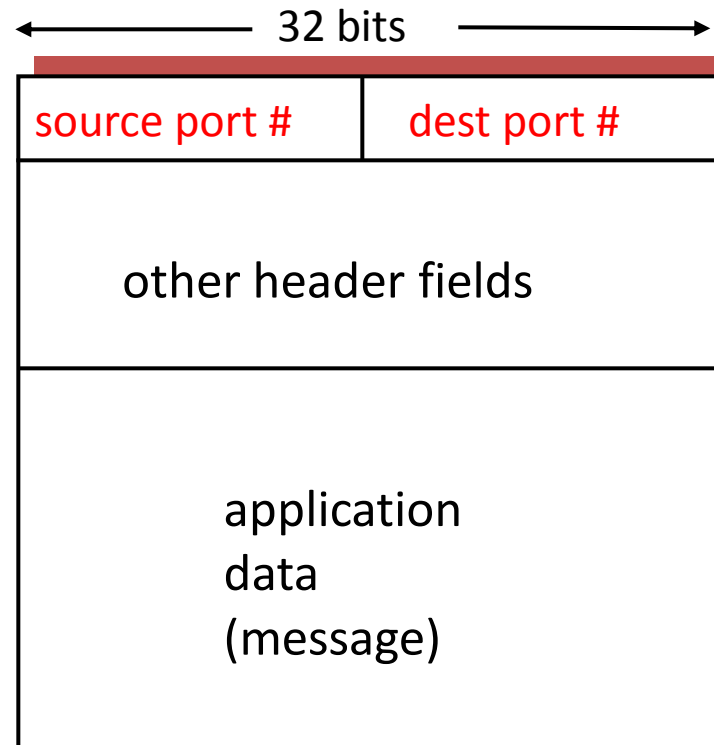
- Thực hiện tại bên nhận
- phân phối các segment nhận được cho socket tương ứng

❑ Khi đóng gói dữ liệu ở tầng transport, header sẽ thêm vào:

- Source port
- Destination port



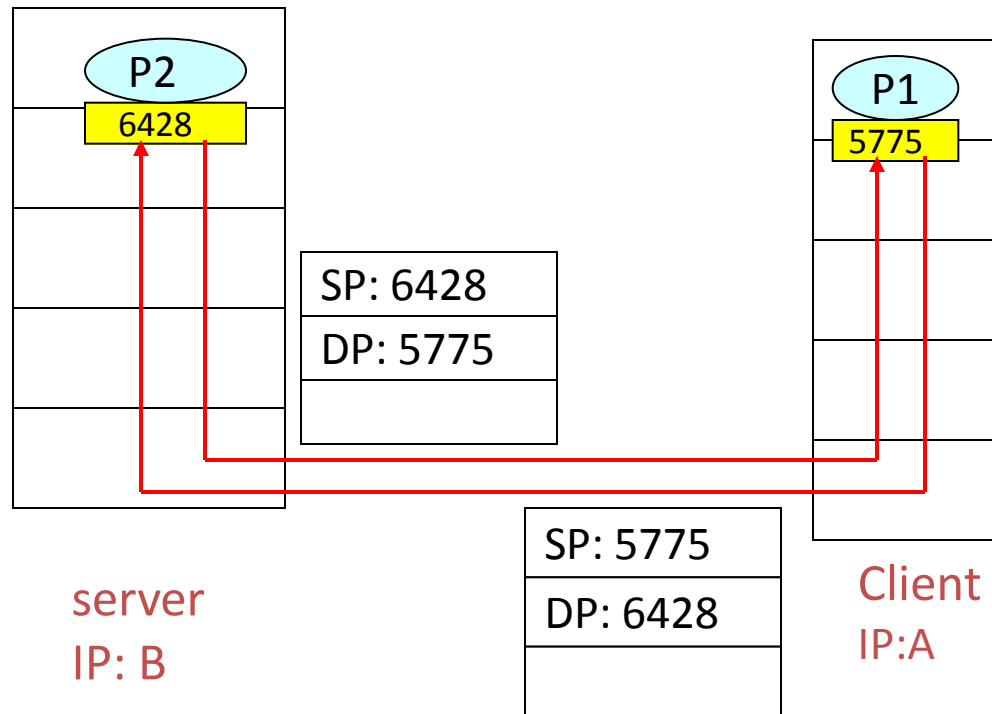
Dồn kênh – Phân kênh - 2



Cấu trúc của một segment



Dồn kênh – Phân kênh - 3



Nội dung



- Giới thiệu
- Giao thức UDP
- Nguyên tắc truyền dữ liệu đáng tin cậy
- Giao thức TCP



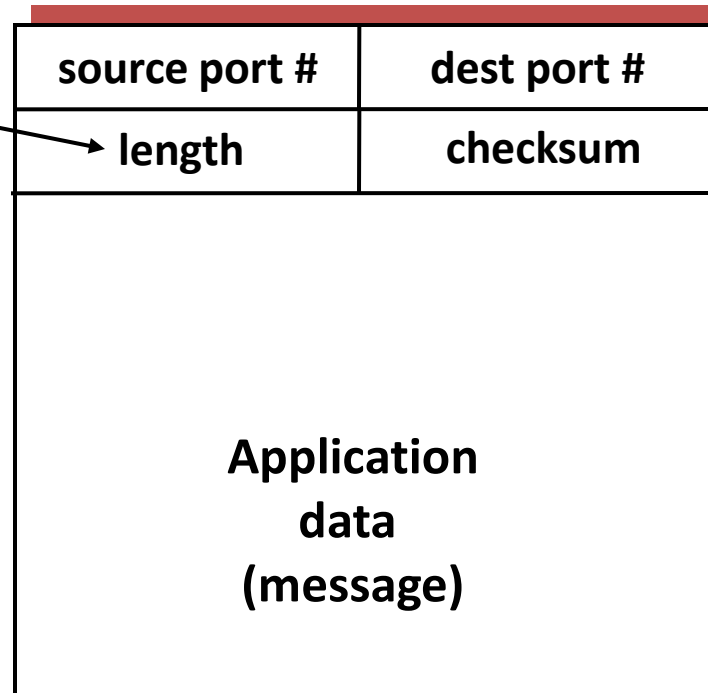
□ UDP: User Datagram Protocol [rfc768]

- Dịch vụ “nỗ lực” để truyền nhanh
- Gói tin UDP có thể:
 - Mất
 - Không đúng thứ tự
- Không kết nối:
 - Không có handshaking giữa bên gửi và nhận
 - Mỗi gói tin UDP được xử lý độc lập
 - Không có trạng thái kết nối

UDP - 2



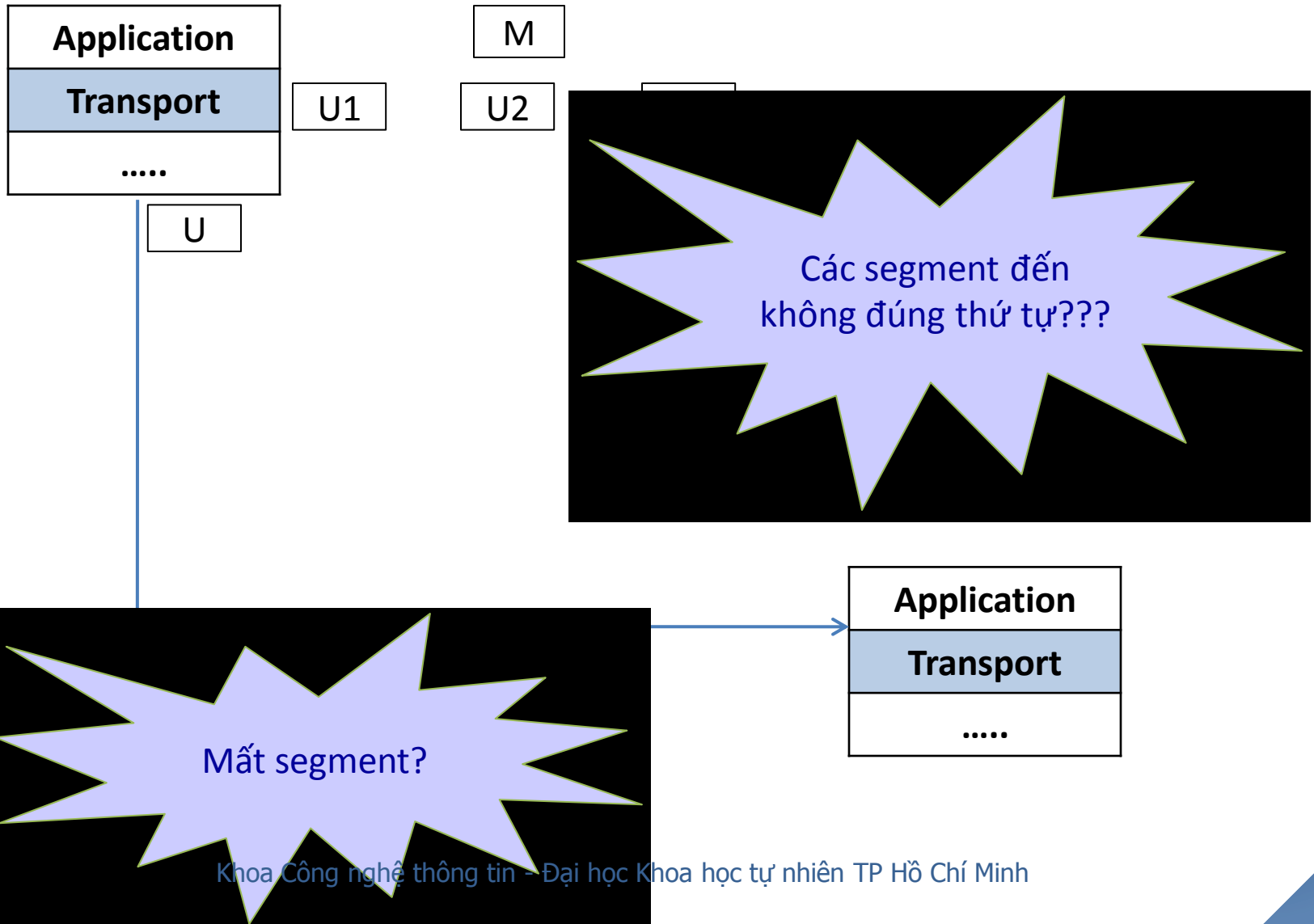
Chiều dài gói tin
(tính cả header)



UDP segment format



UDP - 3



□ Tại sao lại sử dụng UDP?

- Không thiết lập kết nối
- Đơn giản:
 - không quản lý trạng thái nối kết
 - Không kiểm soát luồng
- Header nhỏ
- Nhanh

□ Truyền thông tin cậy qua UDP

- Tầng application phát hiện và phục hồi lỗi

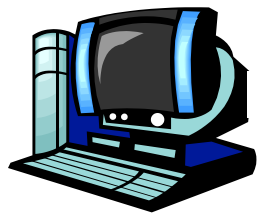
- ❑ Thường sử dụng cho các ứng dụng multimedia
 - Chịu lỗi
 - Yêu cầu tốc độ
- ❑ Một số ứng dụng sử dụng UDP
 - DNS
 - SNMP
 - TFTP
 - ...

Nội dung



- Giới thiệu
- Nguyên tắc truyền dữ liệu đáng tin cậy
- Giao thức TCP
- Giao thức UDP



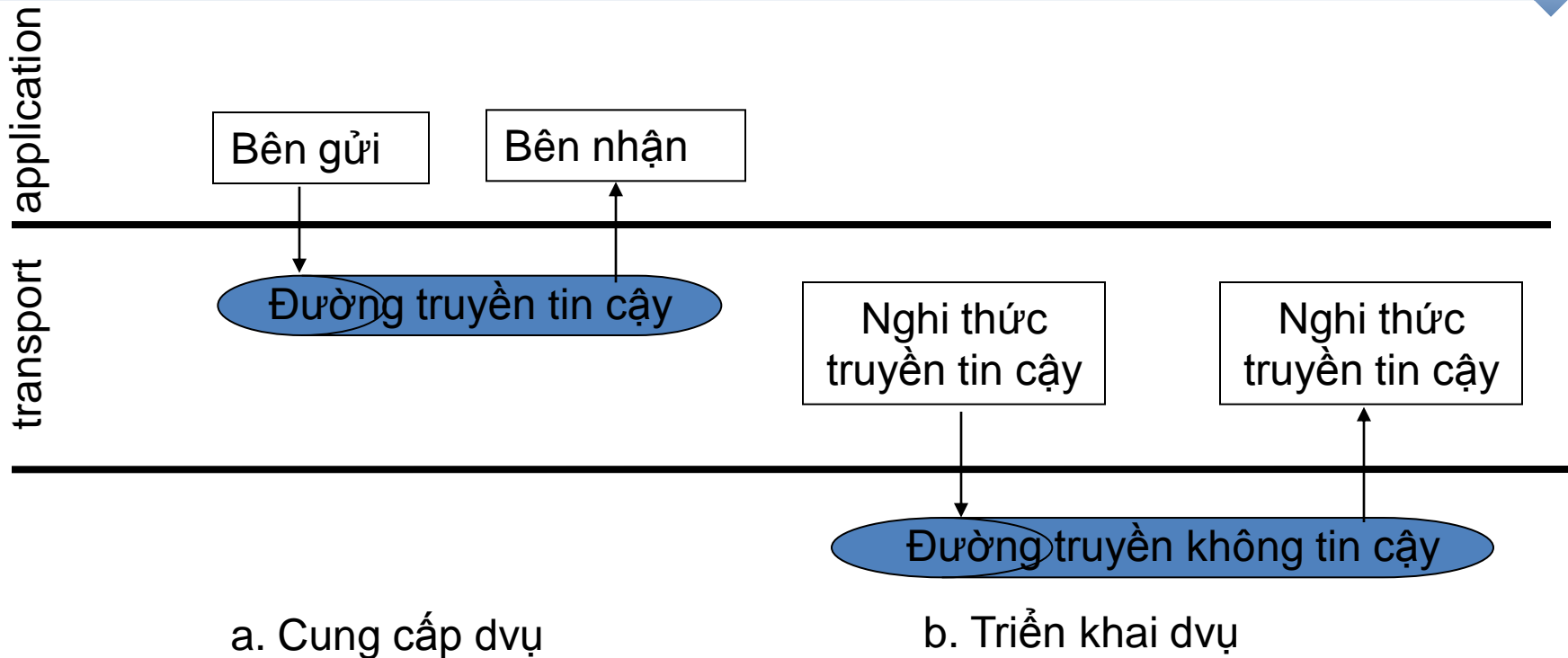


Lỗi bit???
Mất gói???



Làm sao để truyền
đáng tin cậy???

Nguyên lý truyền dữ liệu đáng tin cậy



Đặc tính của đường truyền không tin cậy quyết định độ phức tạp của nghi thức truyền tin cậy



Nghi thức truyền dữ liệu đáng tin cậy

- RDT 1.0
- RDT 2.0, RDT 2.1, RDT 2.2
- RDT 3.0

Pipeline

- Go-back-N
- Gửi lại có chọn

Giải quyết lỗi bit



□ Bên gửi

- Gửi kèm theo thông tin kiểm tra lỗi
- Sử dụng các phương pháp kiểm tra lỗi
 - Checksum, parity checkbit, CRC,...

□ Bên nhận

- Kiểm tra có xảy ra lỗi bit?
- Hành động khi xảy ra lỗi bit?
 - Báo về bên gửi



Giải quyết mất gói



Bên nhận

- Gửi tín hiệu báo
 - Gửi gói tin báo hiệu ACK, NAK

Bên gửi

- Định nghĩa trường hợp mất gói
- Chờ nhận tín hiệu báo
- Hành động khi phát hiện mất gói



Giao thức RDT



□ RDT = Reliable Data Transfer

□ Nguyên tắc: dừng và chờ

▪ Bên gửi

• Gửi gói tin kèm theo thông tin kiểm tra lỗi

• **Dừng và chờ** đến khi nào gói tin vừa gửi đến được bên nhận **an toàn**: nhận được gói tin ACK

• Gửi lại khi có lỗi xảy ra: lỗi bit, mất gói

▪ Bên nhận:

• Kiểm tra lỗi, trùng lặp dữ liệu

• Gửi gói tin phản hồi

□ Phiên bản:

▪ RDT 1.0

▪ RDT 2.0, RDT 2.1, RDT 2.2

▪ RDT 3.0



Nguyên lý pipe line



- ❑ Cho phép gửi nhiều gói tin khi chưa nhận ACK



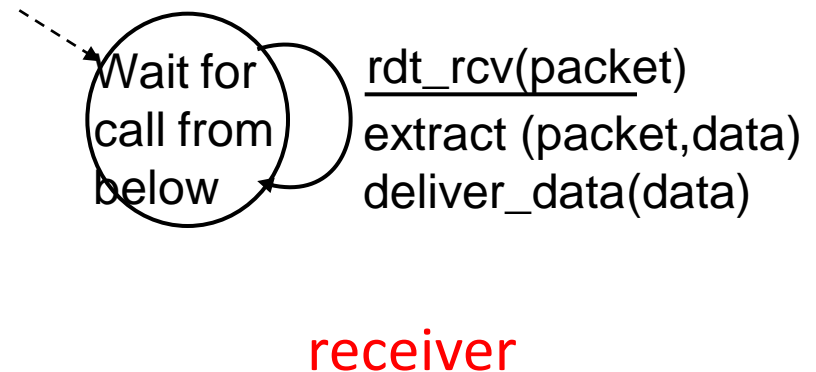
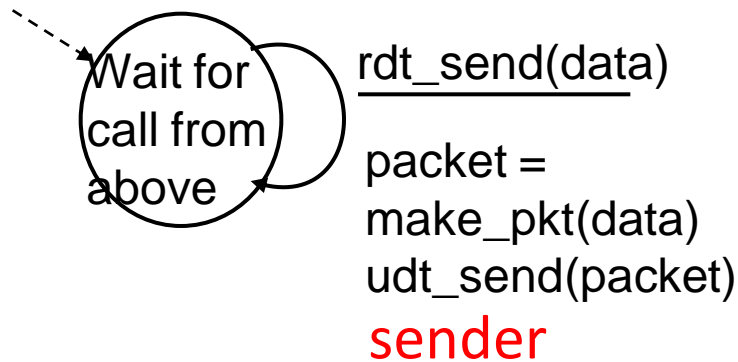
- ❑ Sử dụng buffer để lưu các gói tin
 - Bên gửi: lưu gói tin đã gửi nhưng chưa ack
 - Bên nhận: lưu gói tin đã nhận đúng nhưng chưa đúng thứ tự
- ❑ Giải quyết mất gói
 - Go back N
 - Selective Repeat (gửi lại có chọn)



Rdt1.0 : đường truyền lý tưởng



- ❑ Giả thiết: kênh truyền bên dưới tuyệt đối
 - Không lỗi bit
 - Không mất gói tin
- ❑ FSM (finite state machine) cho bên gửi và nhận
 - Bên gửi chuyển dữ liệu xuống kênh bên dưới
 - Bên nhận đọc dữ liệu từ kênh truyền bên dưới



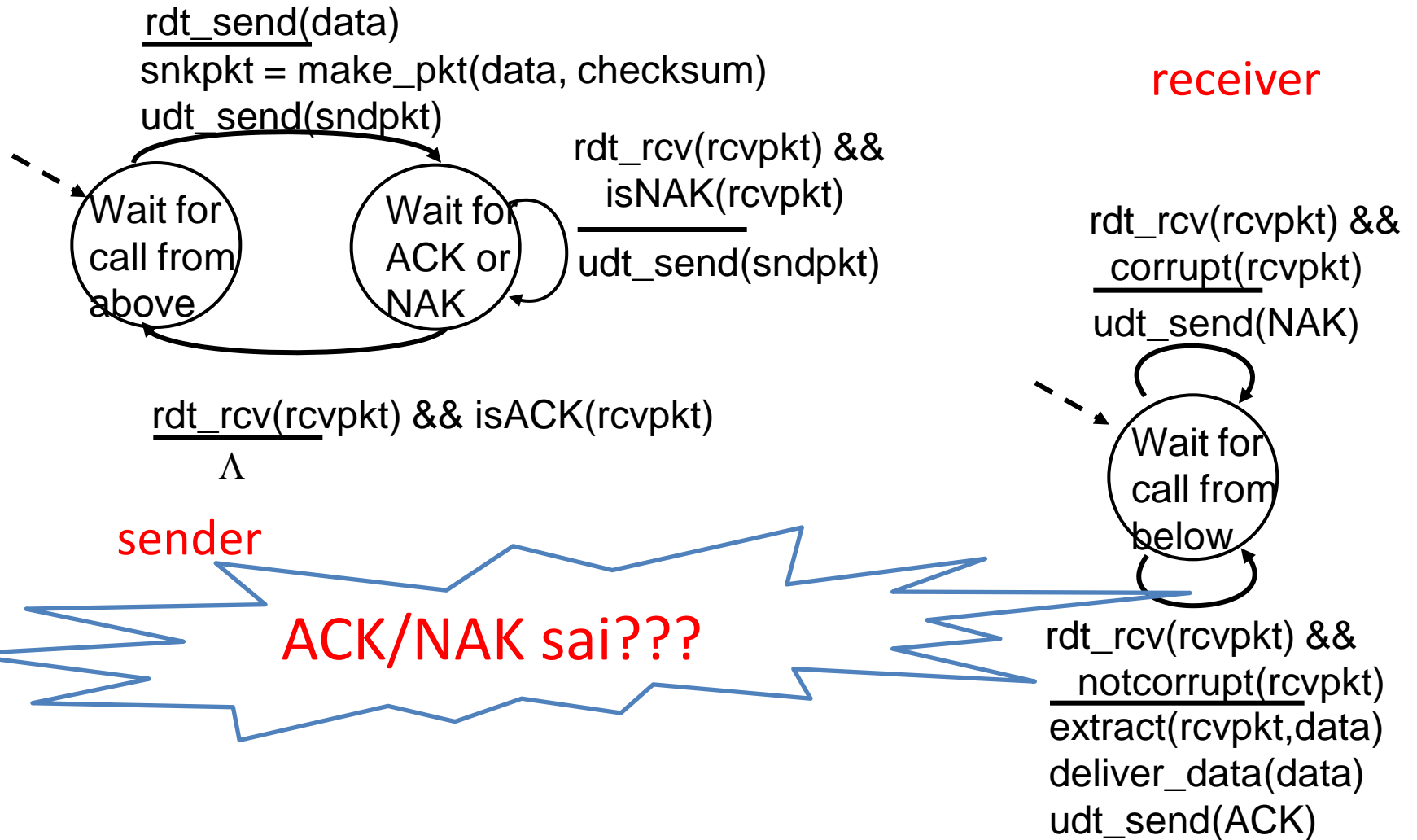
Rdt2.0 kênh truyền có lỗi bit - 1



- ❑ Giả thiết: kênh truyền có thể xảy ra lỗi bit
 - Sử dụng các cơ chế kiểm tra lỗi
 - checksum
- ❑ Làm sao để khắc phục khi nhận ra lỗi?
 - **Acknowledgement(ACKs)**: bên nhận báo cho bên gửi đã nhận được dữ liệu
 - **Negative acknowledgement(NAKs)**: bên nhận báo gói tin bị lỗi
 - Bên gửi sẽ gửi lại gói tin khi nhận NAK
- ❑ So với rdt1.0, rdt2.0:
 - Nhận dạng lỗi
 - Cơ chế phản hồi: ACK, NAK



Rdt2.0 FSM - 2



Rdt2.0 - 3



❑ Giải quyết:

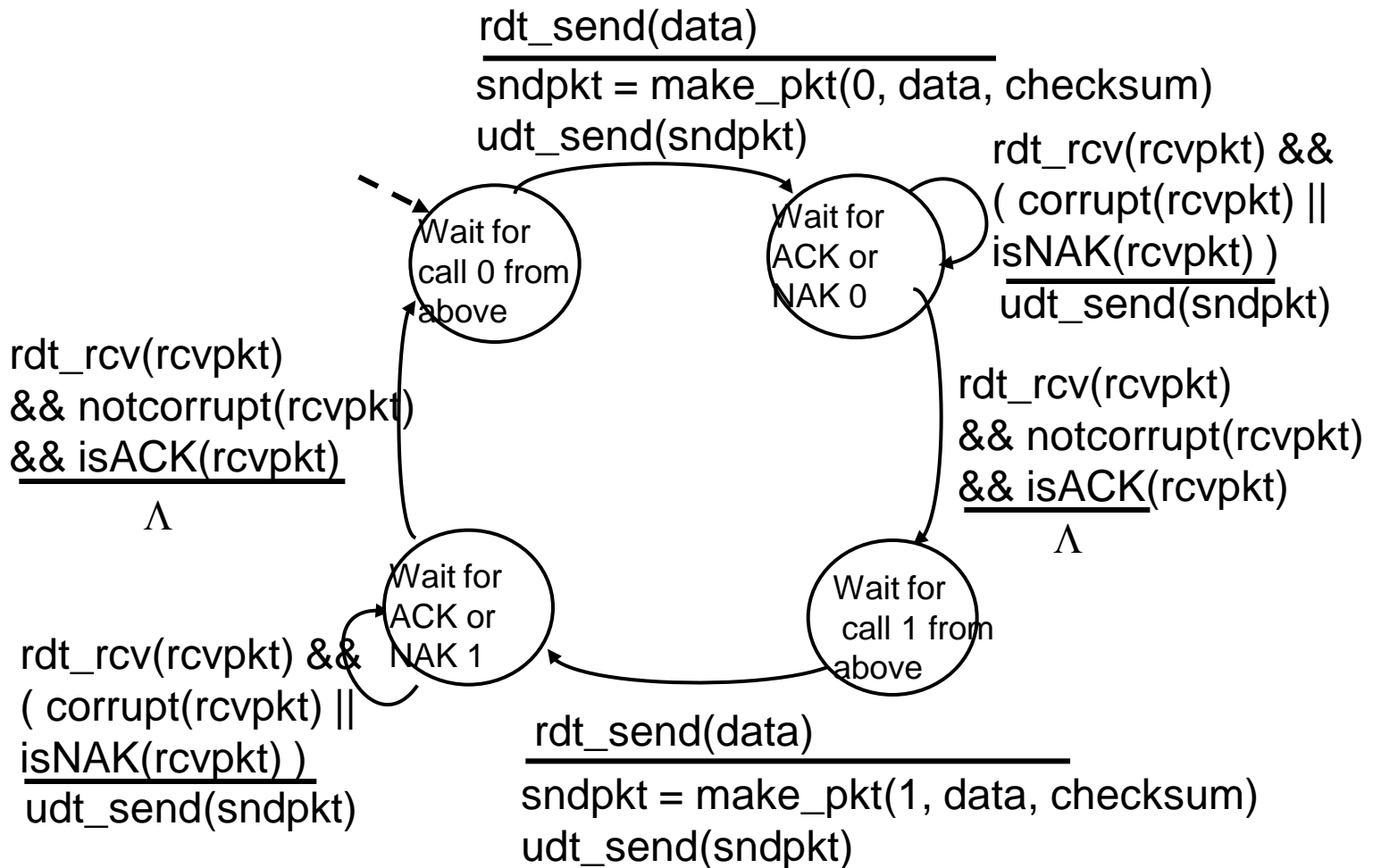
- Bên gửi gửi lại gói tin khi nhận ACK/NAK sai
- Bên gửi đánh **số thứ tự** cho mỗi gói tin
- Bên nhận sẽ loại bỏ gói tin trùng.

❑ Dừng và đợi

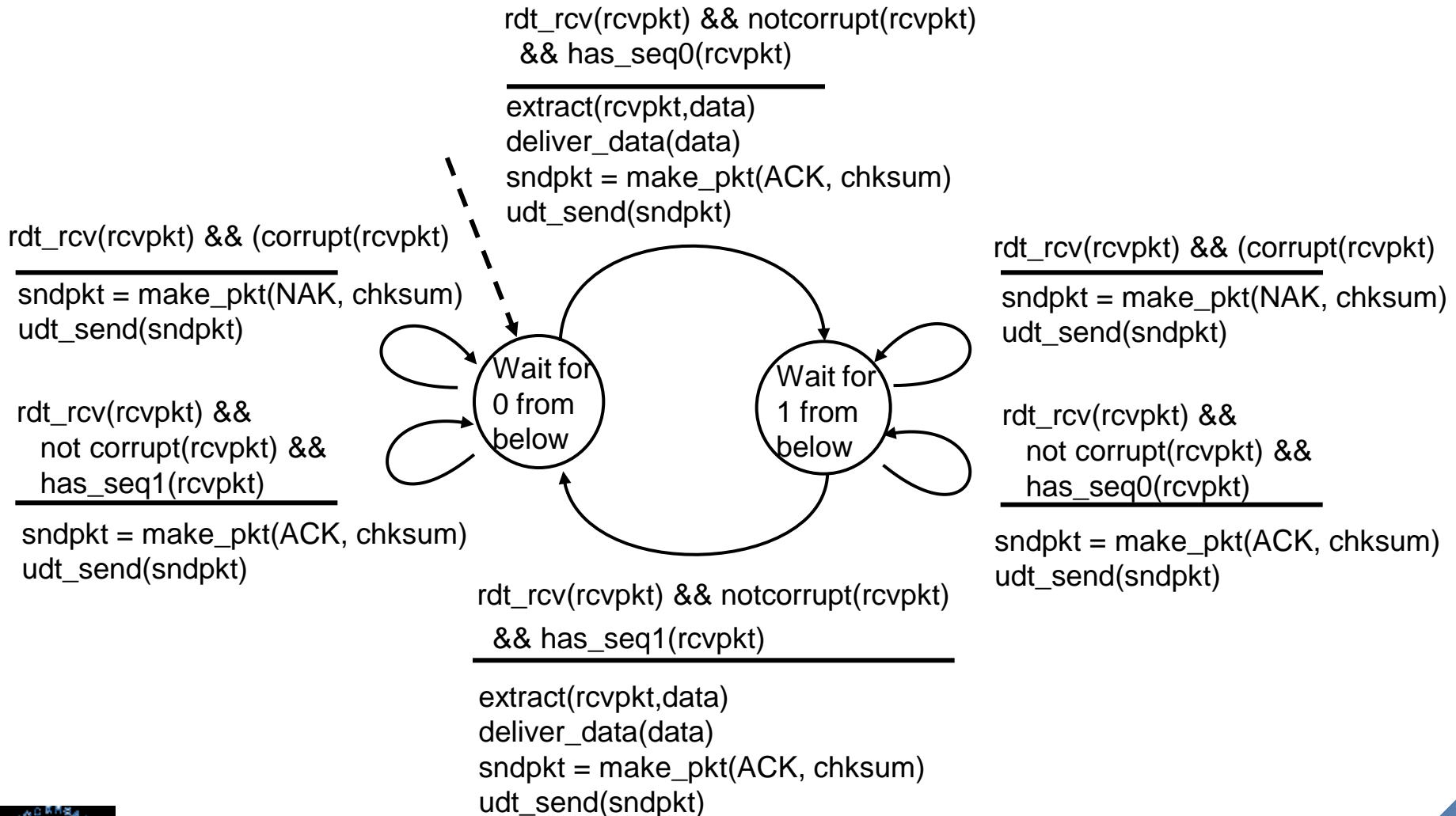
- Bên gửi gửi một gói tin và chờ phản hồi từ bên nhận



Rdt2.1 bên gửi xử lí lỗi ACK/NAK



Rdt2.1 bên nhận xử lí lỗi ACK/NAK



Rdt2.1 thảo luận



Bên gửi

- ❑ Thêm số thứ tự vào gói tin
 - 0 và 1???
- ❑ Phải kiểm tra: ACK/NAK sai không
- ❑ Phải nhớ gói tin hiện thời có thứ tự 0 hay 1

Bên nhận

- Phải kiểm tra nếu nhận trùng
 - So sánh trạng thái đang chờ (0 hay 1) với trạng thái gói tin nhận được
- Bên nhận không biết ACK/NAK cuối cùng có chuyển tới bên gửi an toàn không?



Cơ chế truyền đáng tin cậy - RDT



□ Cơ chế:

- Checksum: kiểm tra có lỗi xảy ra không?
- ACK: bên nhận nhận đúng gói tin
- NAK: bên nhận nhận sai gói tin
- Sequence Number (1 bit = 0 hoặc 1)



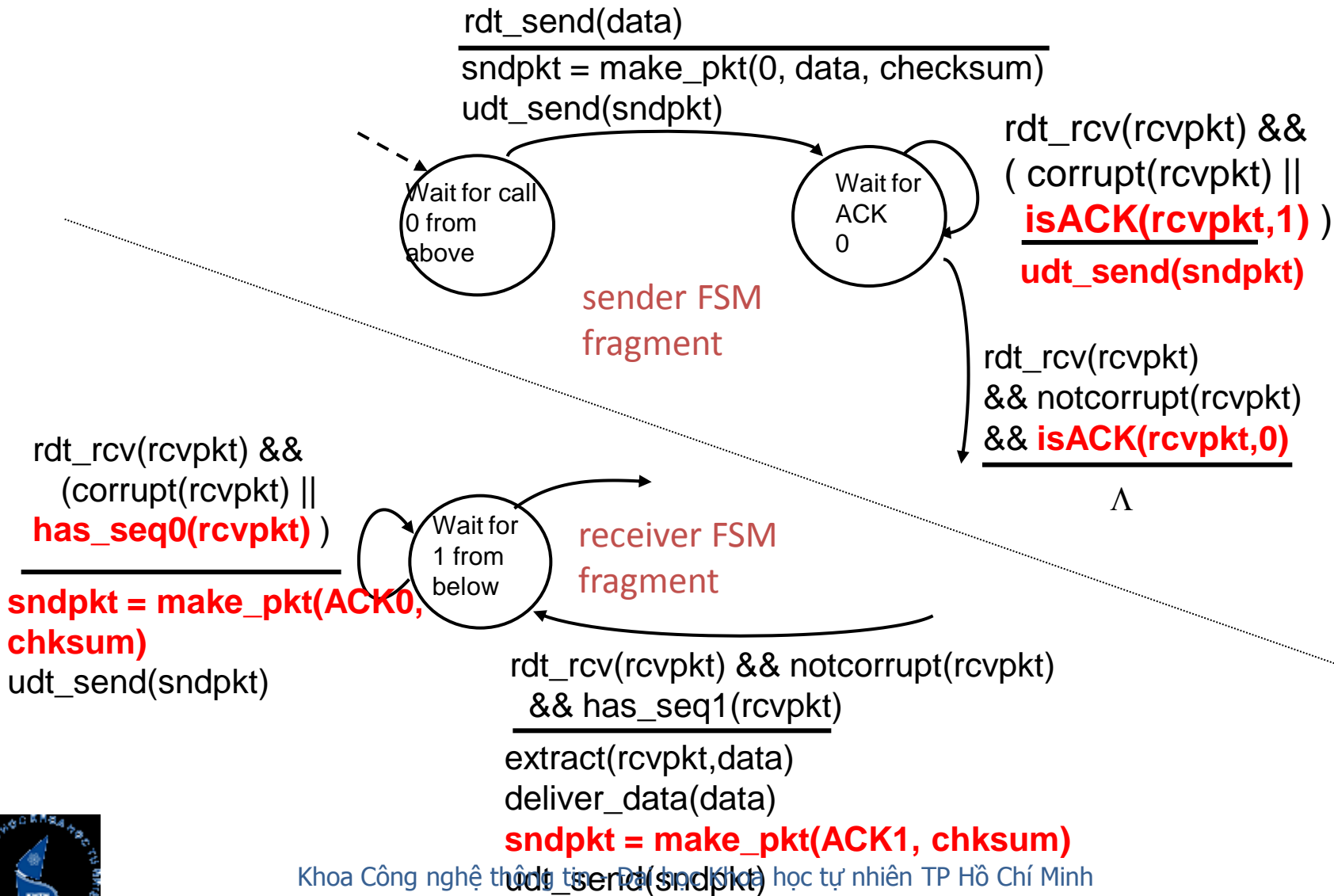
Rdt2.2 không sử dụng NAK



- ❑ Hoạt động giống rdt2.1, nhưng không dùng NAK
 - ❑ Bên nhận gửi ACK cho gói tin không lỗi nhận được cuối cùng.
 - Bên nhận phải thêm số thứ tự vào gói tin ACK
 - ❑ Bên gửi nhận trùng gói tin ACK xem như gói tin NAK
- ➔ gửi lại gói vừa gửi vì gói này chưa nhận được ACK



Rdt2.2: bên gửi và bên nhận



Rdt3.0 kênh truyền có lỗi và mất - 1



❑ Giả thiết:

- Lỗi bit
- mất gói
- Checksum, số thứ tự, ACKs, truyền lại vẫn chưa đủ

❑ Xử lý?

Giải pháp:

- bên gửi đợi một khoảng thời gian hợp lí cho ACK
- Gửi lại nếu không nhận đc ACK trong khoảng thời gian này
- Nếu gói tin (hay ACK) bị trễ (không mất)
 - Gửi lại có thể trùng, phải đánh số thứ tự
 - Bên nhận phải xác định thứ tự của gói tin đã ACK
- Yêu cầu đếm thời gian



Rdt3.0 bên gửi - 2



rdt_rcv(rcvpkt) &&
(corrupt(rcvpkt) ||
isACK(rcvpkt,1))

Λ

rdt_send(data)

sndpkt = make_pkt(0, data, checksum)
udt_send(sndpkt)
start_timer

Wait for
call 0 from
above

Wait
for
ACK0

timeout
udt_send(sndpkt)
start_timer

rdt_rcv(rcvpkt)
&& notcorrupt(rcvpkt)
&& isACK(rcvpkt,1)
stop_timer

rdt_rcv(rcvpkt)
&& notcorrupt(rcvpkt)
&& isACK(rcvpkt,0)
stop_timer

timeout
udt_send(sndpkt)
start_timer

Wait
for
ACK1

Wait for
call 1 from
above

rdt_rcv(rcvpkt)

Λ

rdt_rcv(rcvpkt) &&
(corrupt(rcvpkt) ||
isACK(rcvpkt,0))

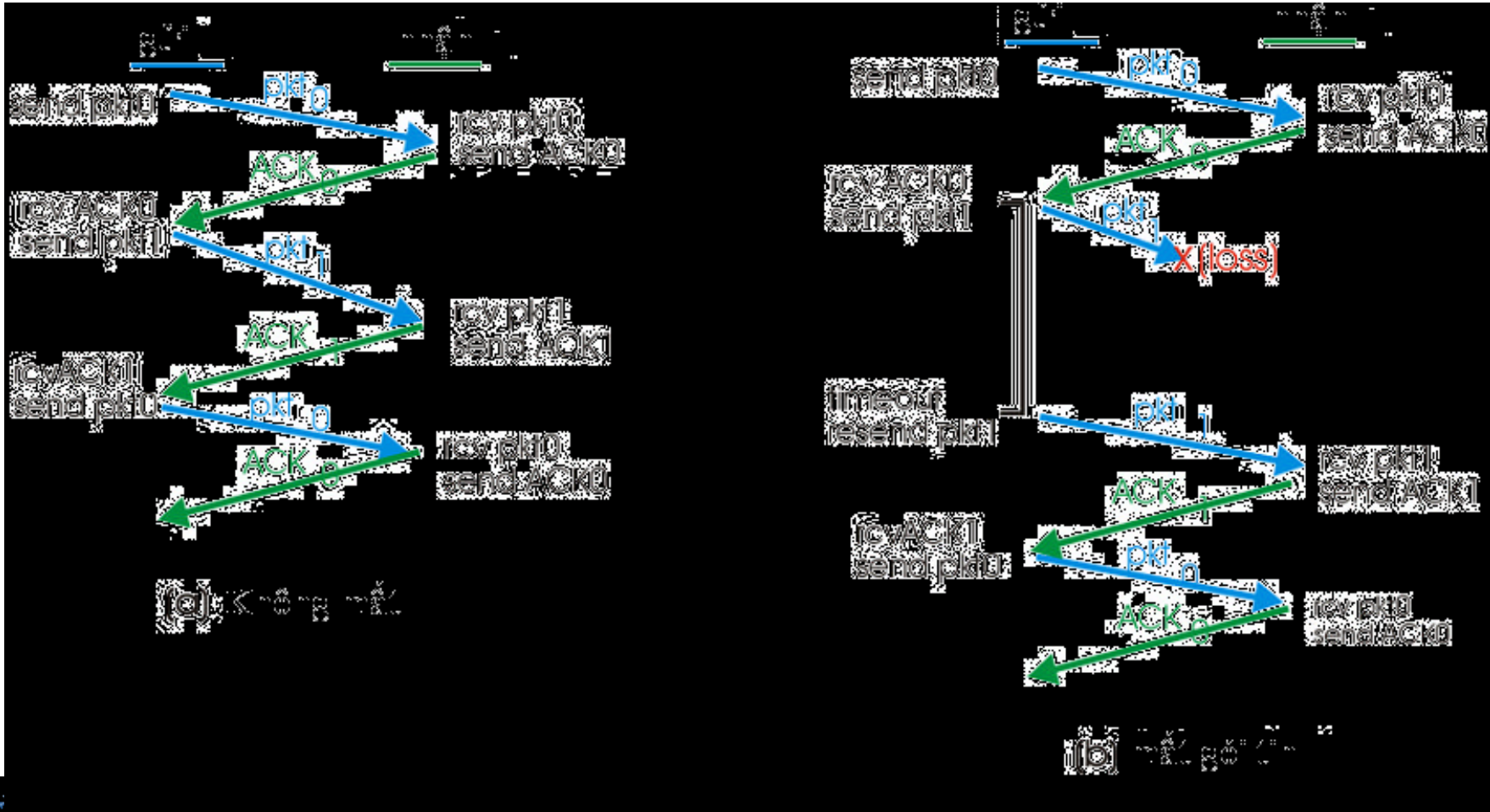
Λ

rdt_send(data)

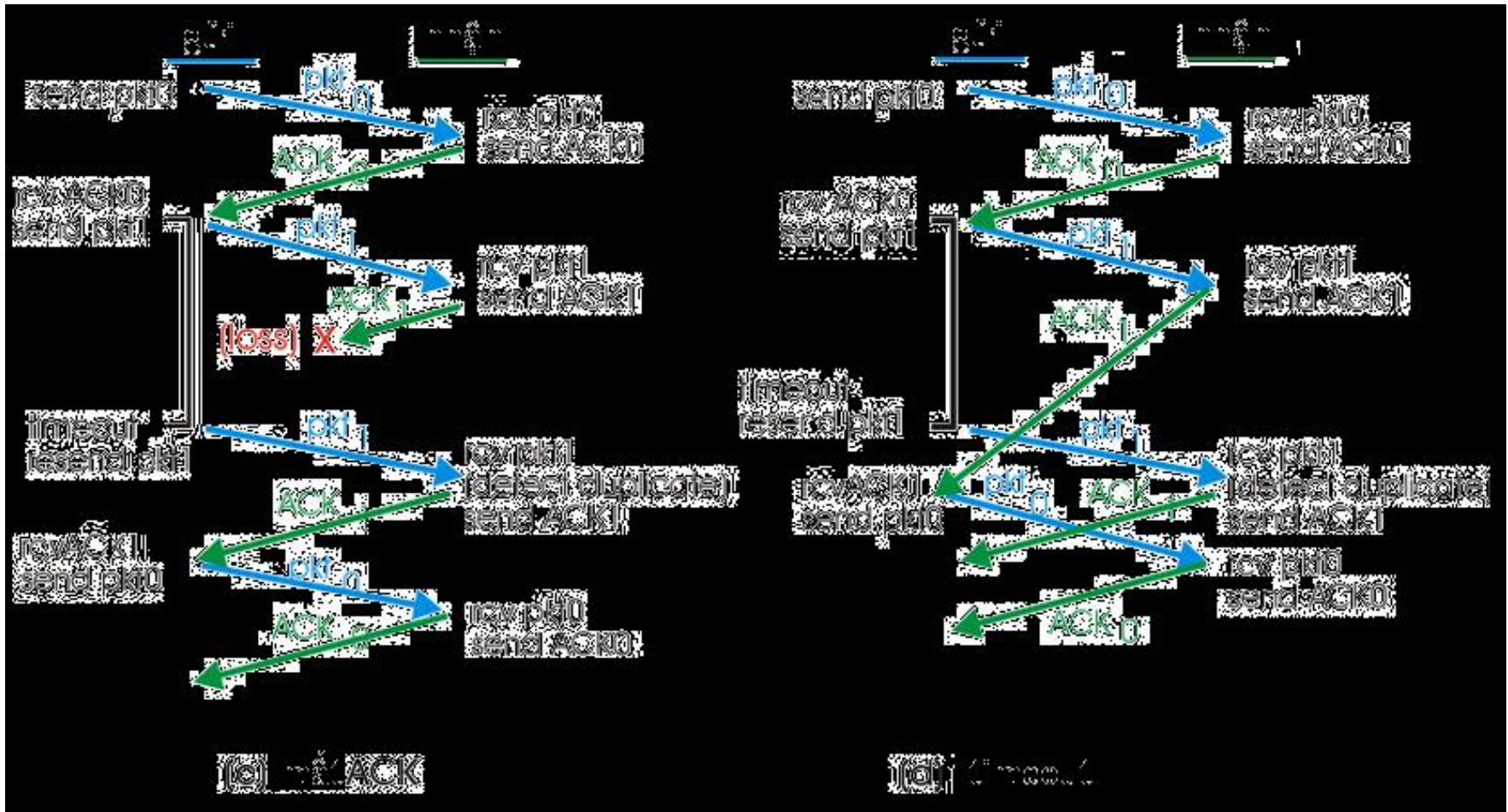
sndpkt = make_pkt(1, data, checksum)
udt_send(sndpkt)
start_timer



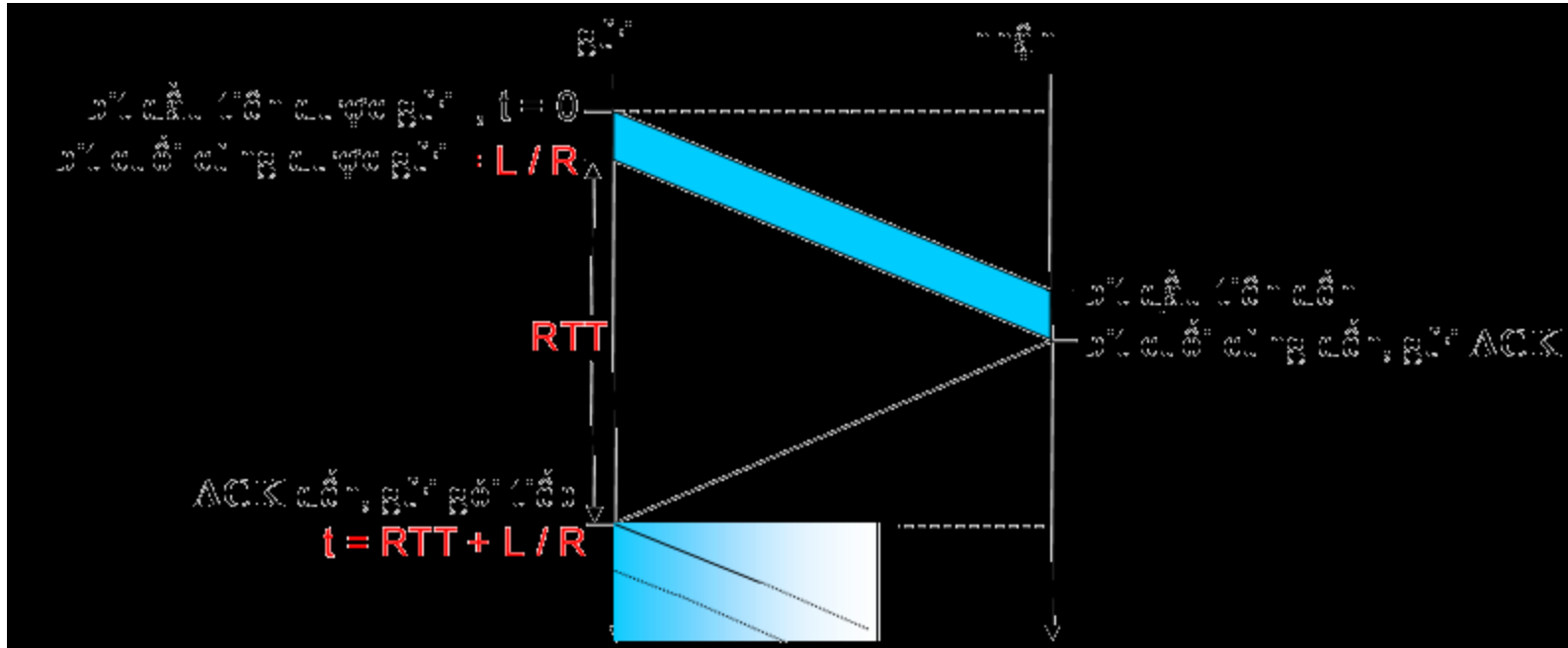
Rdt3.0 - 3



Rdt3.0 - 4



Rdt3.0 dừng và đợi - 5



Rdt3.0 – Hiệu quả - 6



- ❑ Rdt3.0 làm việc, nhưng không hiệu quả
- ❑ Vd: băng thông 1Gbps, 15ms end2end delay, gói tin 8Kb

$$T_{\text{transmit}} = \frac{L \text{ (packet length in bits)}}{R \text{ (transmission rate, bps)}} = \frac{8\text{kb/pkt}}{10^{**}9 \text{ b/sec}} = 8 \text{ microsec}$$

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

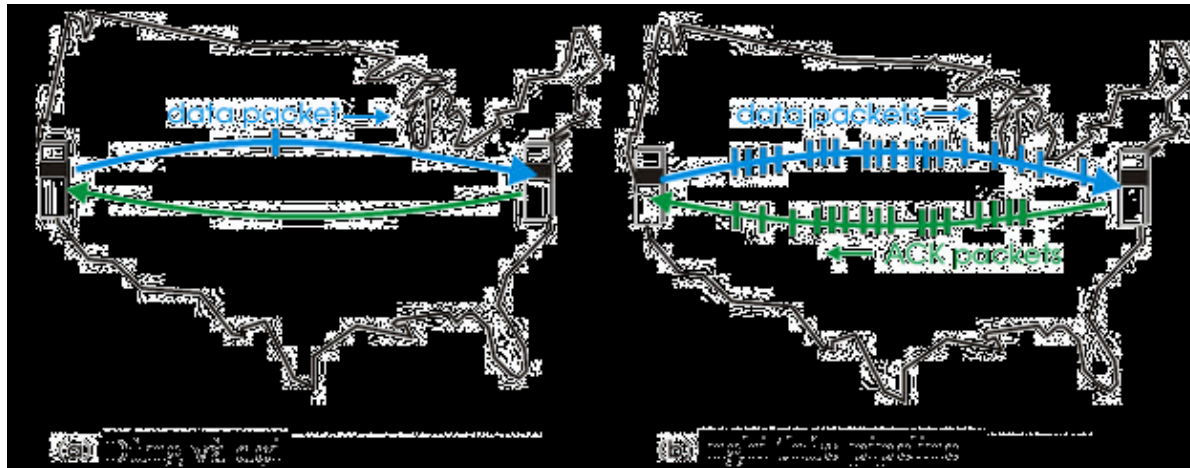
- U_{sender} : tỉ lệ thời gian bên gửi gửi gói tin
- Nghi thức đã hạn chế việc sử dụng tài nguyên mạng



Nghi thức pipeline - 1



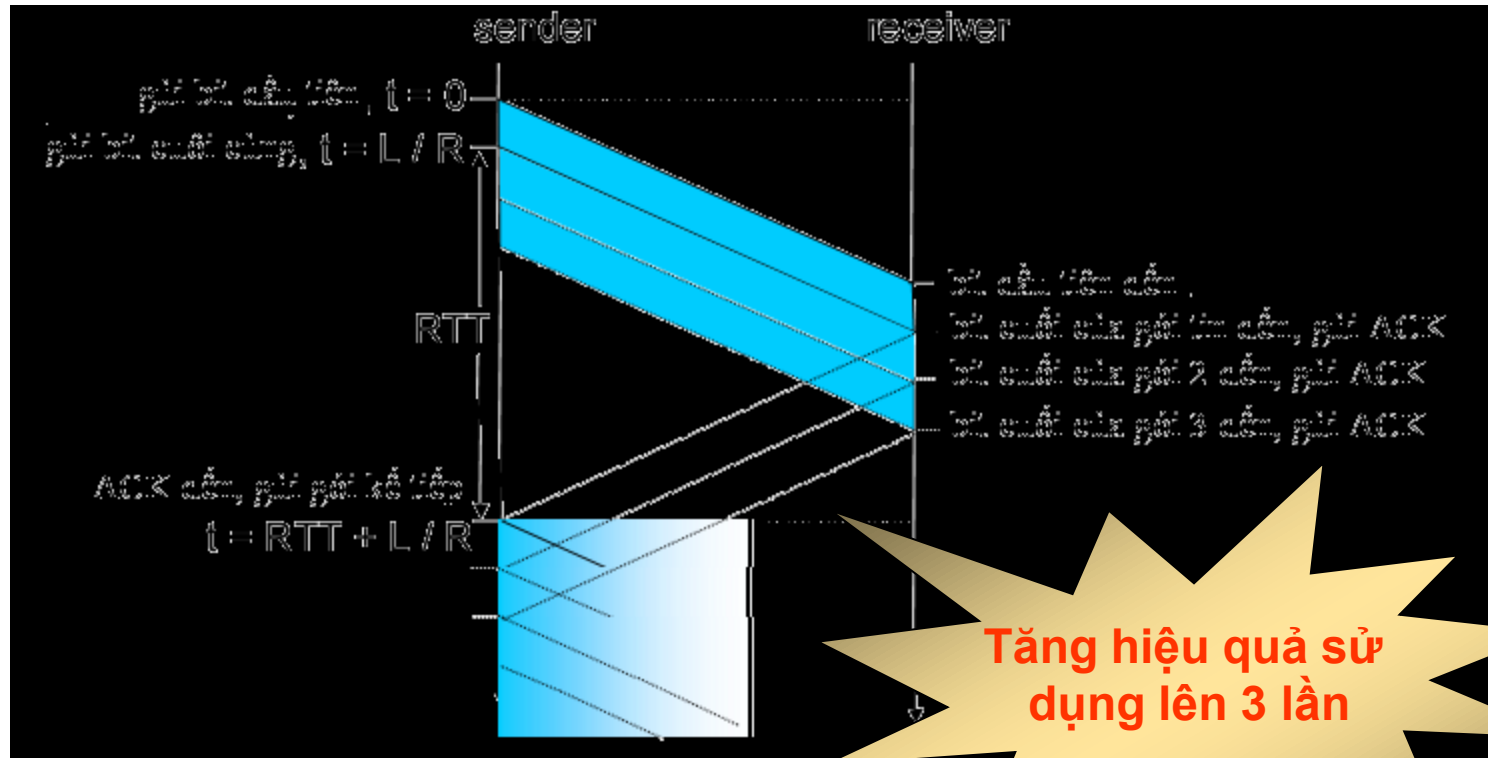
- ❑ Pipelining: bên gửi cho phép gửi nhiều gói tin khi chưa được báo nhận (ACK)
 - Gói tin: sắp theo thứ tự tăng dần
 - Dùng bộ đệm ở bên gửi hoặc/và bên nhận: "Sliding window"



- Có hai giải pháp chính của nghi thức pipeline:
 - go-Back-N
 - gửi lại có chọn.



Nghi thức pipeline - 2



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



Go-Back-N – 1



- Số thứ tự: k-bit
- “window” = N → số gói tin được gửi liên tục không ACK



- ACK(seq#): nhận đúng đến seq#



Go-Back-N: bên nhận - 2



❑ Bên gửi:

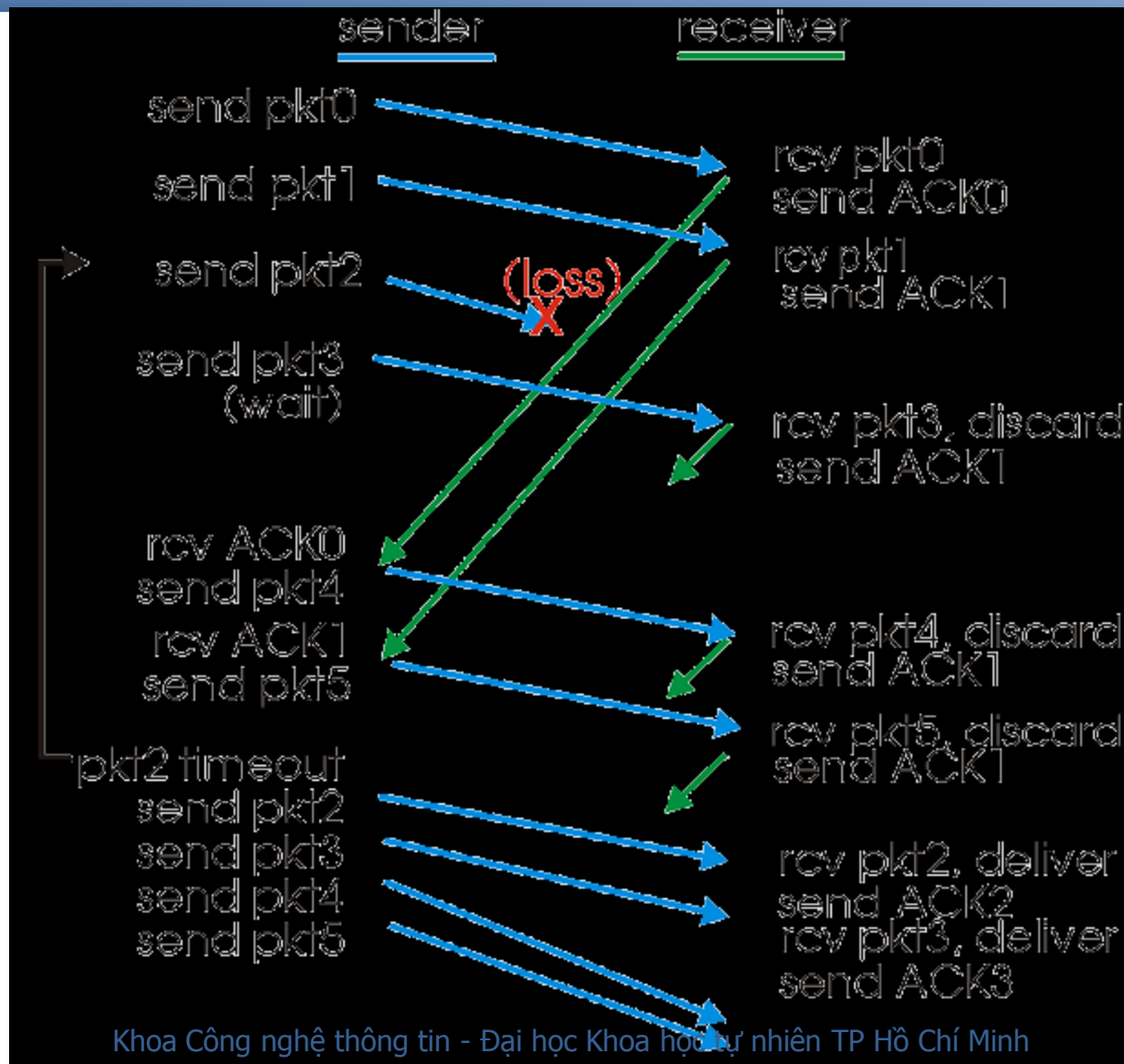
- Sử dụng buffer (“window”) để lưu các gói tin đã gửi nhưng chưa nhận được ACK
- Gửi nếu gói tin có thể đưa vào “window”
- Thiết lập đồng hồ cho gói tin cũ nhất (gói tin ở đầu “window”)
- Timeout: gửi lại tất cả các gói tin chưa ACK trong window

❑ Bên nhận:

- Chỉ gửi ACK cho gói tin đã nhận đúng với số thứ tự cao nhất
 - Có thể phát sinh trùng ACK
- Chỉ cần nhớ số thứ tự đang đợi
- Gói tin không theo thứ tự:
 - Loại bỏ: không có bộ đệm
 - Gửi lại ACK với số thứ tự lớn nhất



Go-Back-N – ví dụ - 3



Gửi lại có chọn - 1



□ Bên nhận:

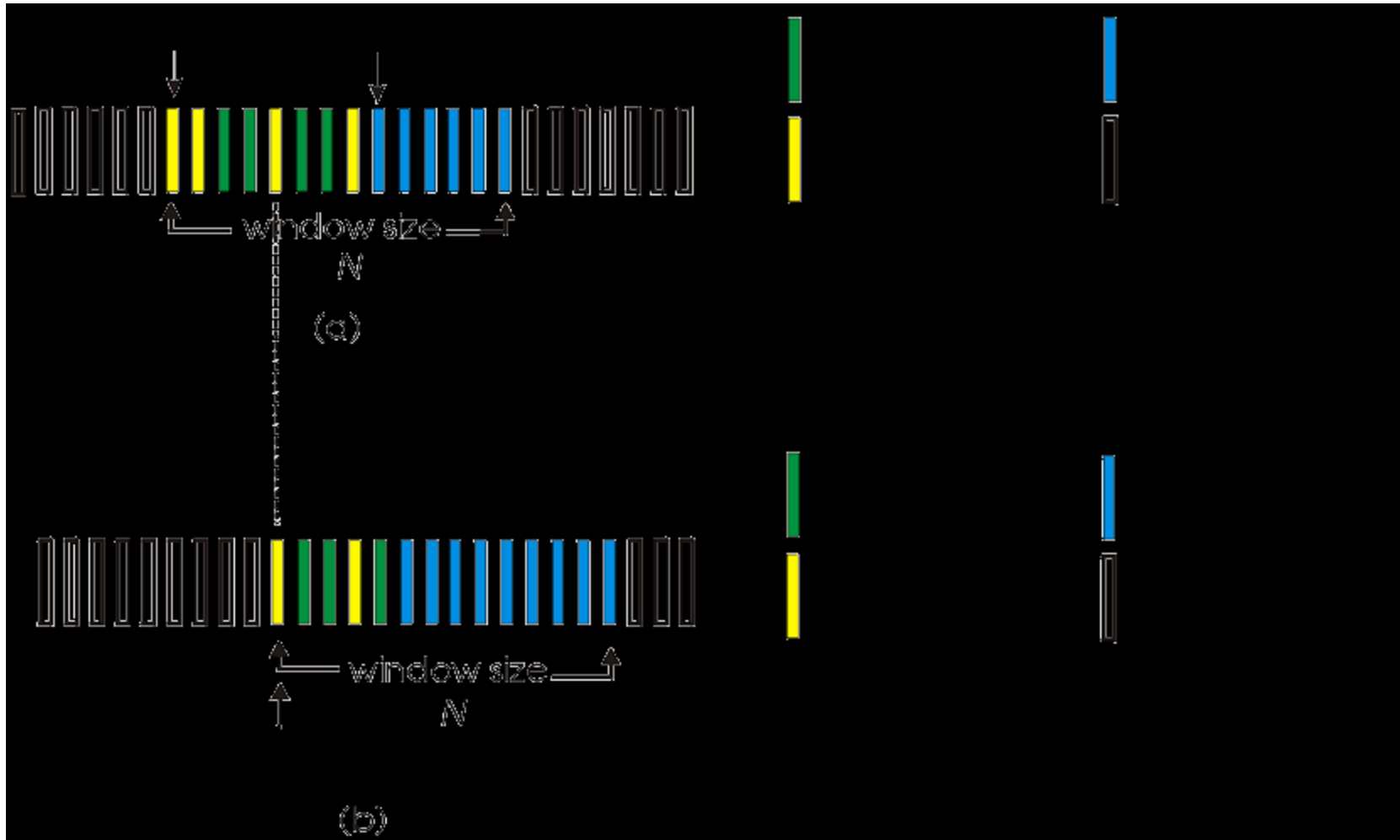
- Báo nhận riêng lẻ từng gói tin nhận đúng
 - ACK(seq#): đã nhận đúng gói tin seq#
- dùng bộ đệm để lưu các gói tin không đúng thứ tự
- Nhận 1 gói tin không đúng thứ tự
 - Đưa vào bộ đệm nếu còn chỗ
 - Hủy gói tin

□ Bên gửi:

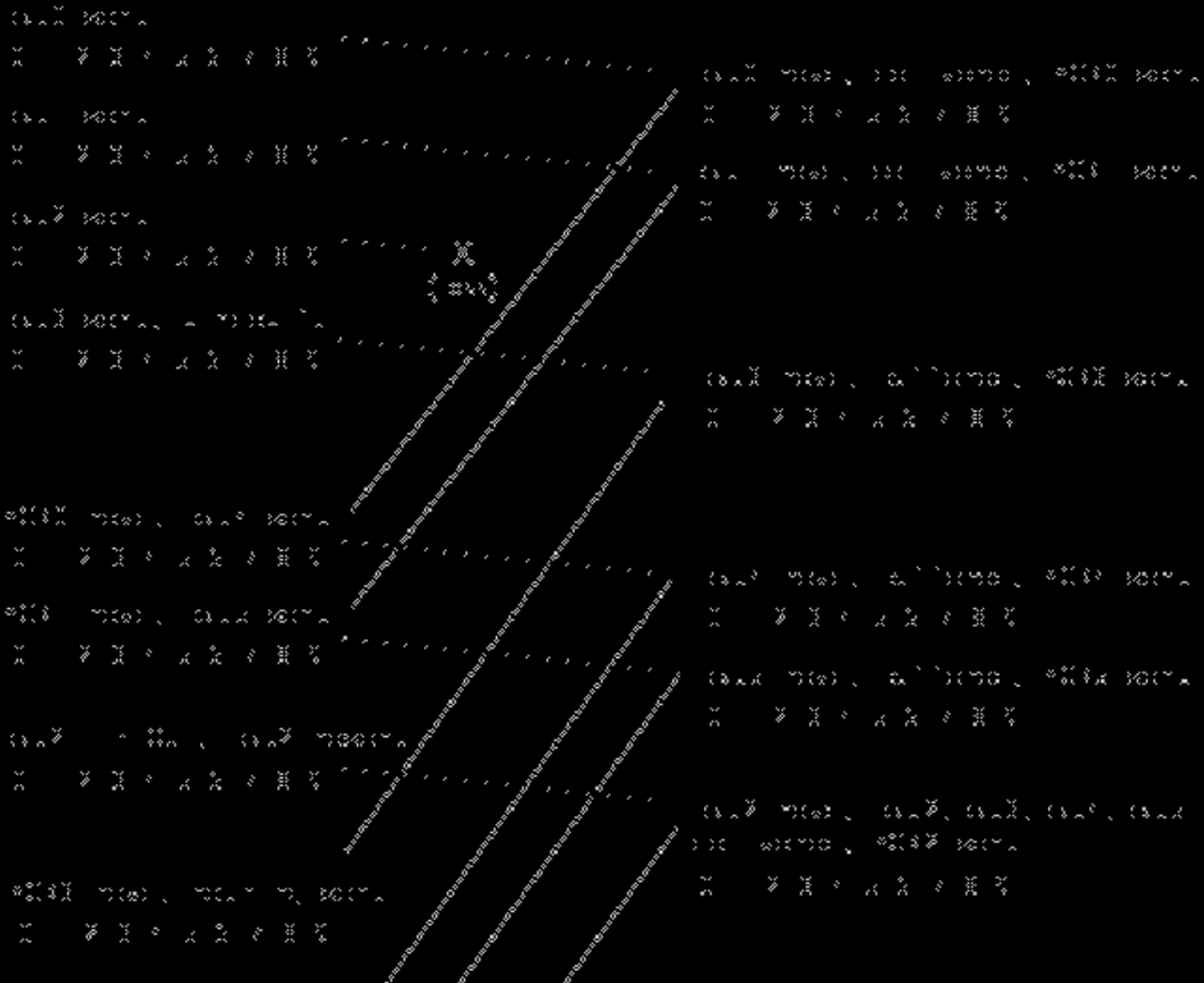
- Có đồng hồ cho mỗi gói tin chưa nhận đc ACK
- Time out: chỉ gửi những gói tin không nhận được ACK



Gửi lại có chọn - 2



Gửi lại có chọn - 4



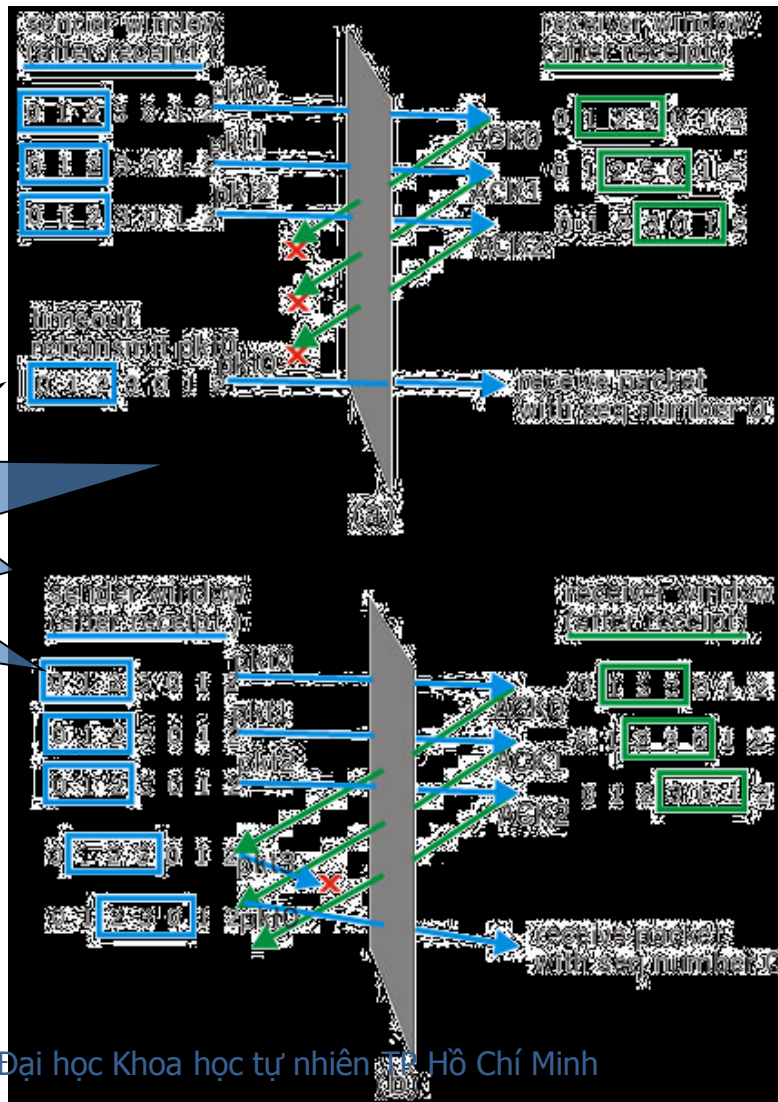
Gửi lại có chọn - 5



□ Vd:

- Số thứ tự: 0,1,2,3
- Window size: 3

Mối quan hệ giữa số thứ tự và window size???



Nội dung



- Giới thiệu
- Nguyên tắc truyền dữ liệu đáng tin cậy
- Giao thức TCP
- Giao thức UDP



- Giới thiệu
- Nguyên tắc hoạt động
- Quản lý kết nối
- Điều khiển luồng
- Điều khiển tắt nghẽn

TCP – giới thiệu - 1



□ TCP = Transport Control Protocol

- rfc: 793,1122,1323,2018,2581
- Point – to – point
 - 1 người gửi và 1 người nhận
- Full-duplex
 - Dữ liệu truyền 2 chiều trên cùng kết nối
 - MSS: maximum segment size
- Hướng kết nối
 - Handshaking trước khi gửi dữ liệu

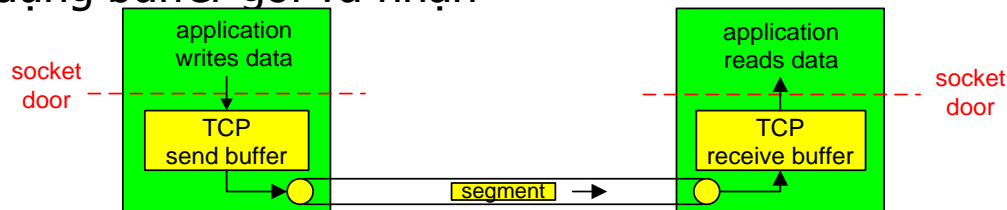


TCP - giới thiệu - 2



□ TCP = Transport Control Protocol

- TCP cung cấp kết nối theo kiểu dòng (**stream-of-bytes**)
 - Không có ranh giới giữa các gói tin
 - Sử dụng buffer gửi và nhận



- Tin cậy, theo thứ tự
- Pipeline
- Kiểm soát luồng
- Kiểm soát tắc nghẽn



TCP – cấu trúc gói tin



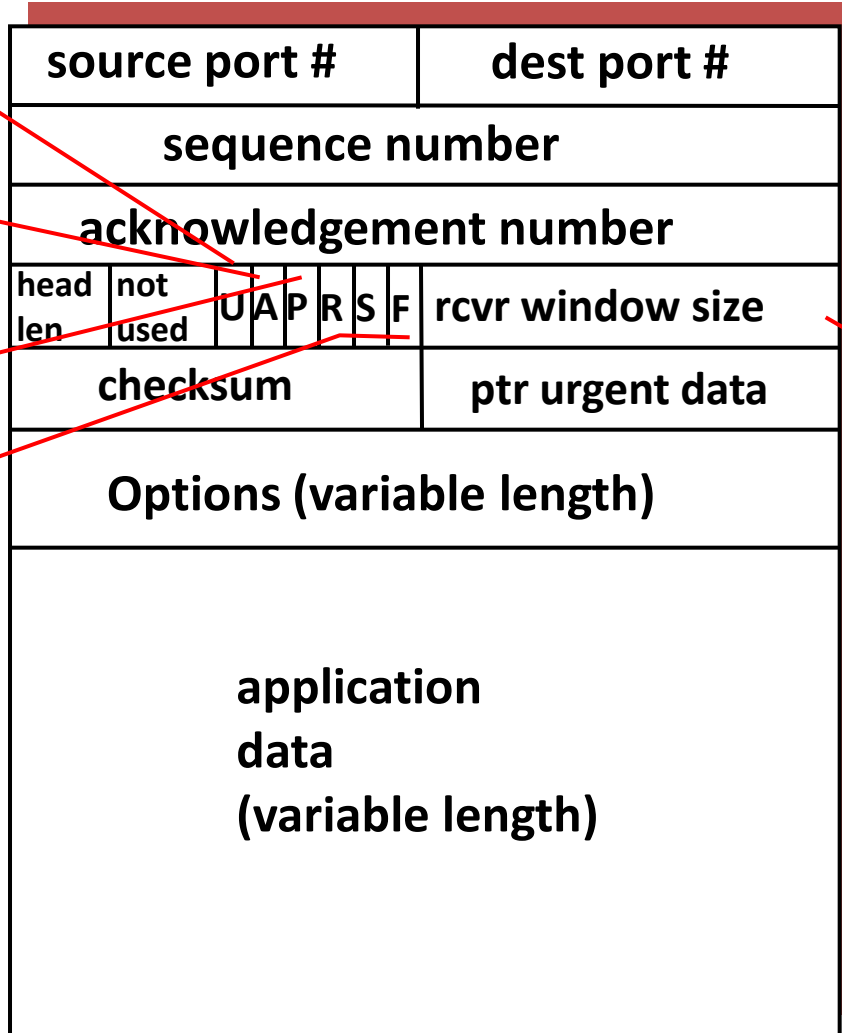
URG: urgent data
(generally not used)

ACK: ACK #
valid

PSH: push data now
(generally not used)

RST, SYN, FIN:
connection estab
(setup, teardown
commands)

← 32 bits →



bytes
rcvr willing
to accept



TCP – định nghĩa các trường - 1



- Source & destination port
 - Port của nơi gửi và nơi nhận
- Sequence number
 - Số thứ tự của byte đầu tiên trong phần data của gói tin
- Acknowledgment number
 - Số thứ tự của byte đang mong chờ nhận tiếp theo
- Window size
 - Thông báo có thể nhận bao nhiêu byte sau byte cuối cùng được xác nhận đã nhận



TCP – định nghĩa các trường - 2



❑ Checksum

- Checksum TCP header

❑ Urgent pointer

- Chỉ đến dữ liệu khẩn trong trường dữ liệu

❑ Cờ:

- URG = trường urgent pointer valid
- ACK = trường Acknowledge number valid
- PSH = dữ liệu cần phân phối ngay
- RST = chỉ định nối kết cần thiết lập lại (reset)
- SYN = sử dụng để thiết lập kết nối
- FIN = sử dụng để đóng kết nối

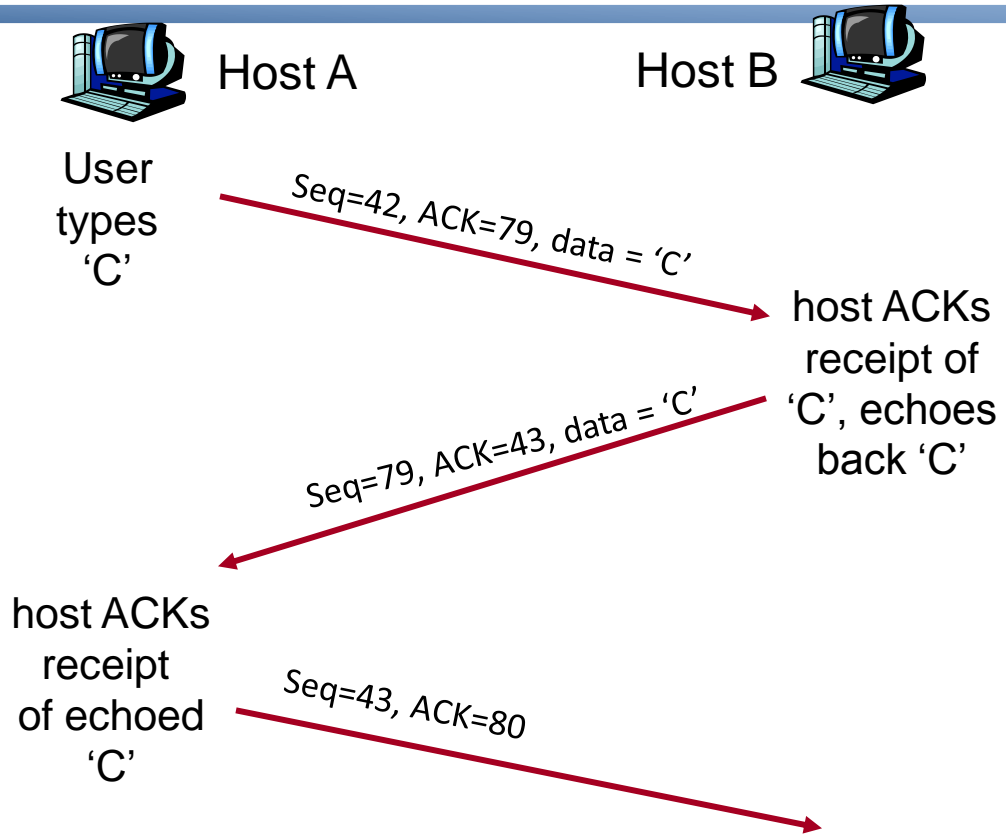


TCP – ví dụ



Seq: số thứ tự của byte
đầu tiên trong vùng data

ACK: số thứ tự của byte
chờ nhận tiếp theo



simple telnet scenario



□ Nguyên tắc: dùng pipeline

- Bên gửi đính kèm thông tin kiểm tra lỗi trong mỗi gói tin
- Sử dụng ACK để báo nhận
- Thiết lập thời gian timeout khi cho gói tin ở đầu buffer
- Gửi lại toàn bộ dữ liệu trong buffer khi hết time out



TCP – bên gửi



- Nhận dữ liệu từ tầng ứng dụng
 - Tạo các segment
 - Bật đồng hồ (nếu chưa bật)
 - Thiết lập thời gian chờ, timeout
- Nhận gói tin ACK
 - Nếu trước đó chưa nhận: trượt “cửa sổ”
 - Thiết lập lại thời gian của đồng hồ
- Hết time out
 - Gửi lại dữ liệu còn trong buffer
 - Reset đồng hồ



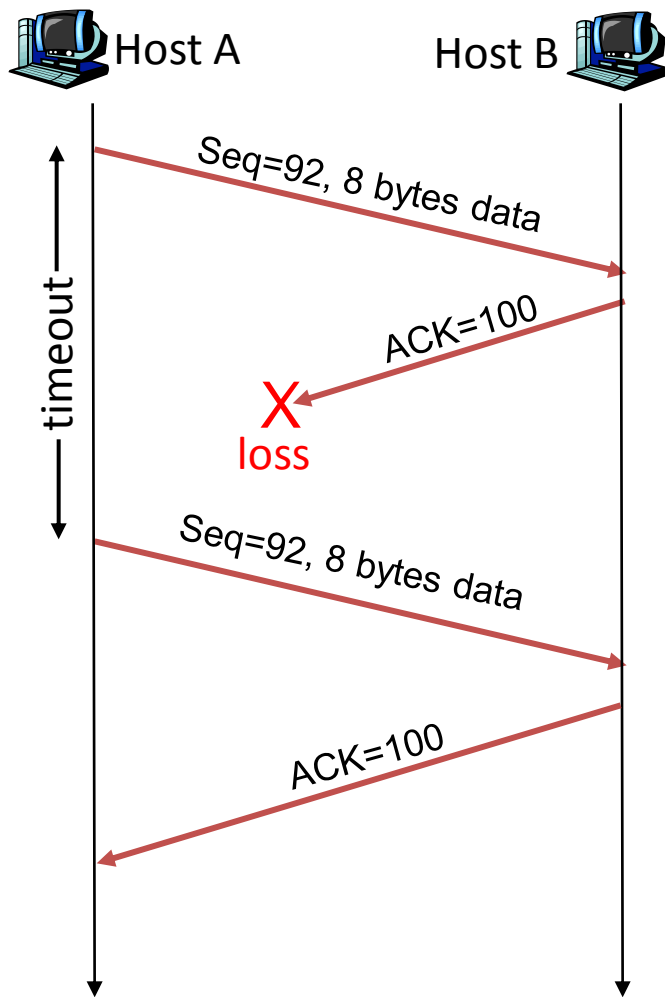
TCP – bên nhận



- ❑ Nhận gói tin đúng thứ tự
 - Chấp nhận
 - Gửi ACK về cho bên gửi
- ❑ Nhận gói tin không đúng thứ tự
 - Phát hiện “khoảng trống dữ liệu (GAP)”
 - Gửi ACK trùng

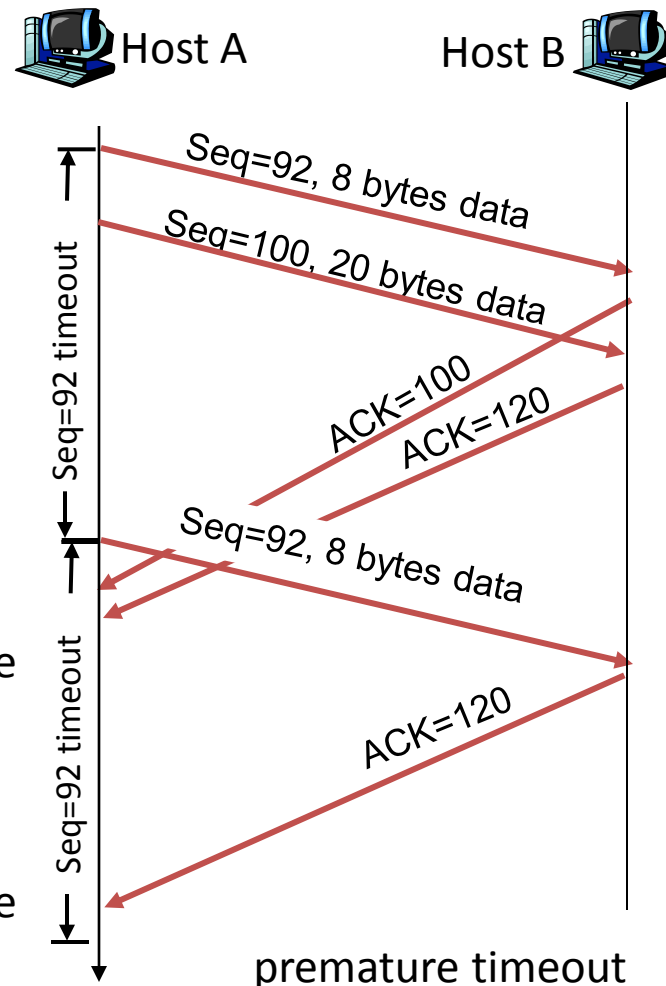


TCP – ví dụ



time

lost ACK scenario



Sendbase = 100
SendBase = 120

SendBase = 120

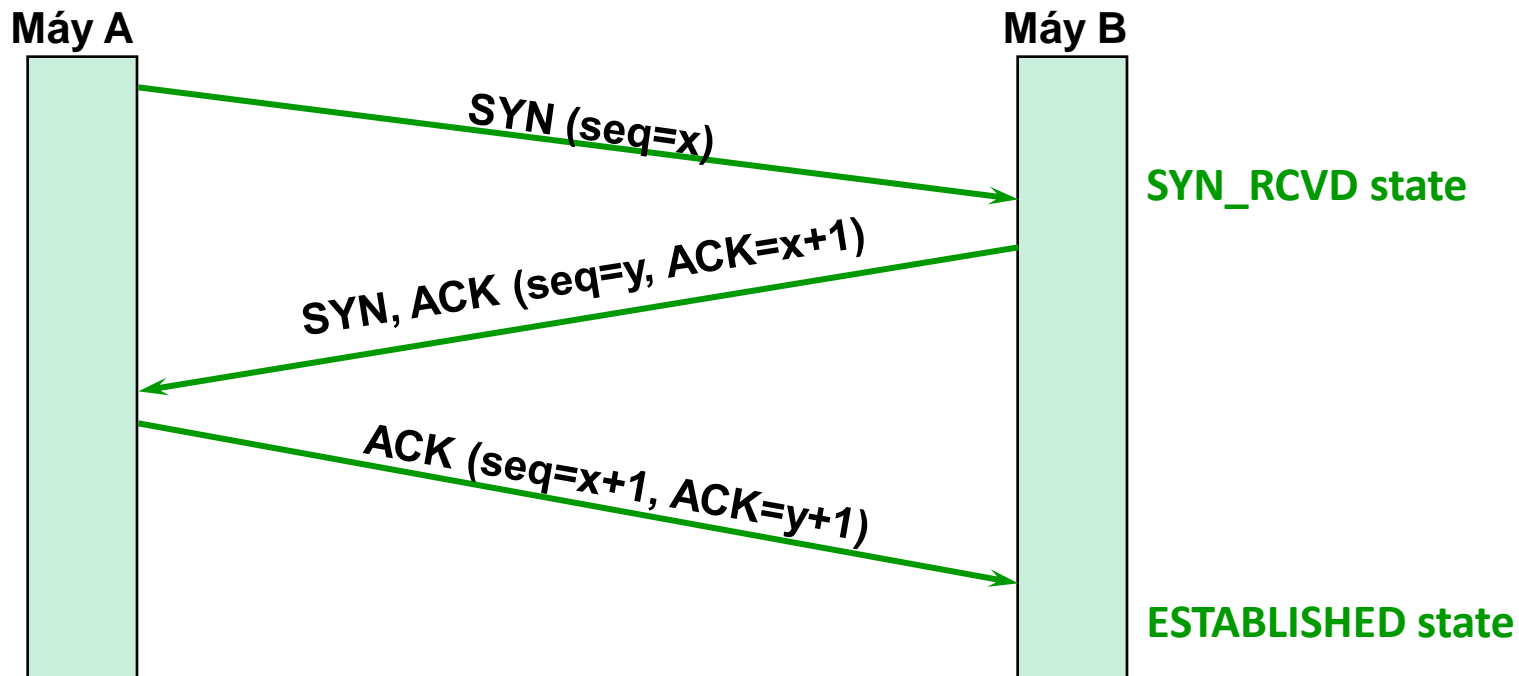
time



TCP – thiết lập kết nối



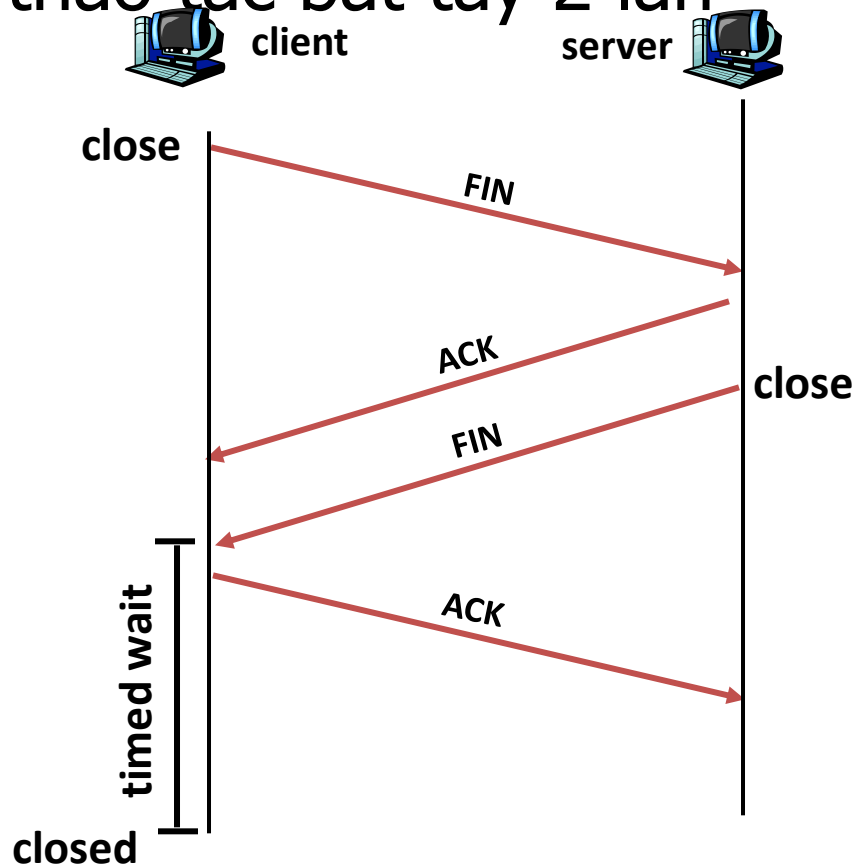
- ❑ Thực hiện thao tác bắt tay 3 lần (Three way handshake)



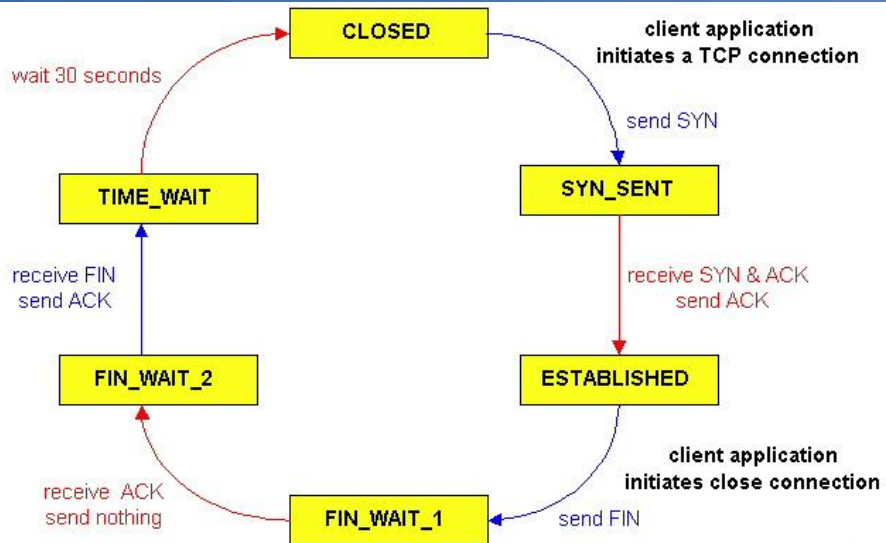
TCP – đóng kết nối



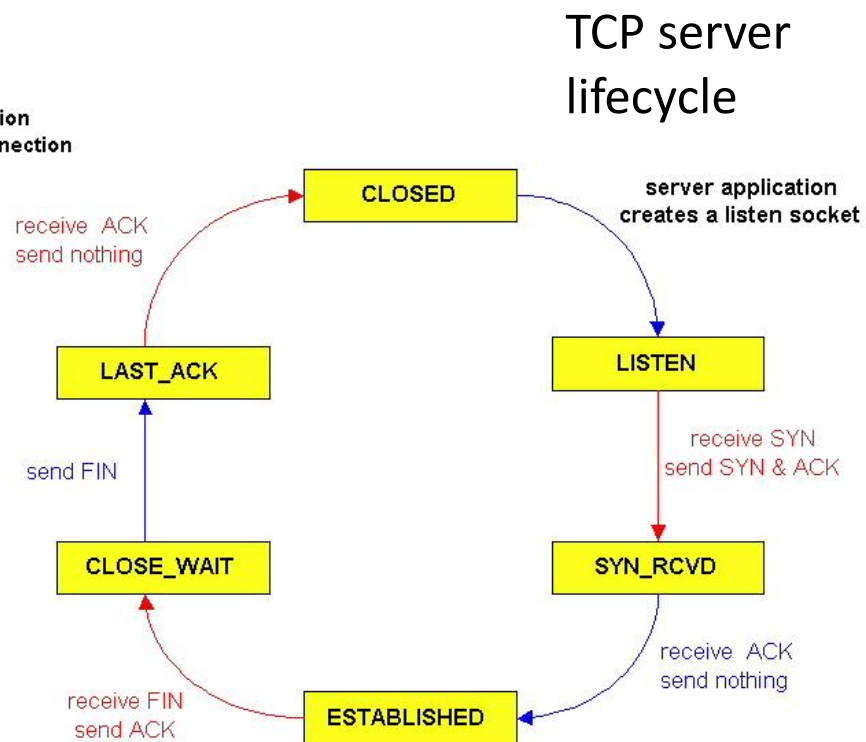
❑ Thực hiện thao tác bắt tay 2 lần



TCP – quản lý kết nối



TCP client lifecycle



TCP server lifecycle



TCP - Điều khiển luồng - 1

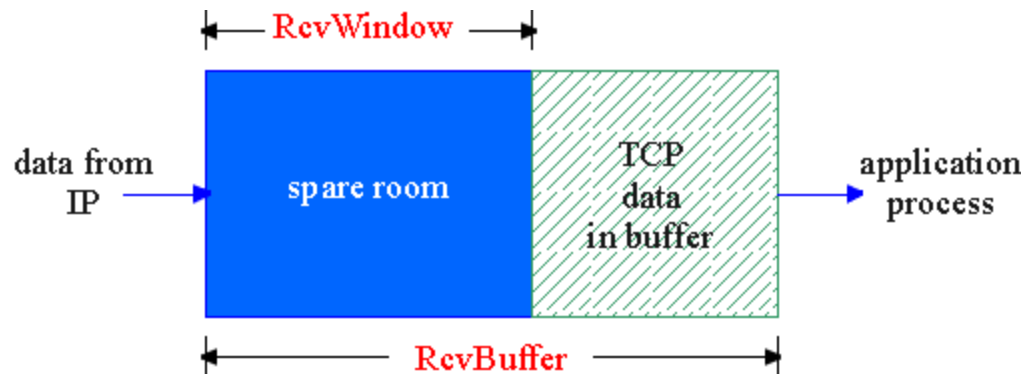


□ Nguyên nhân:

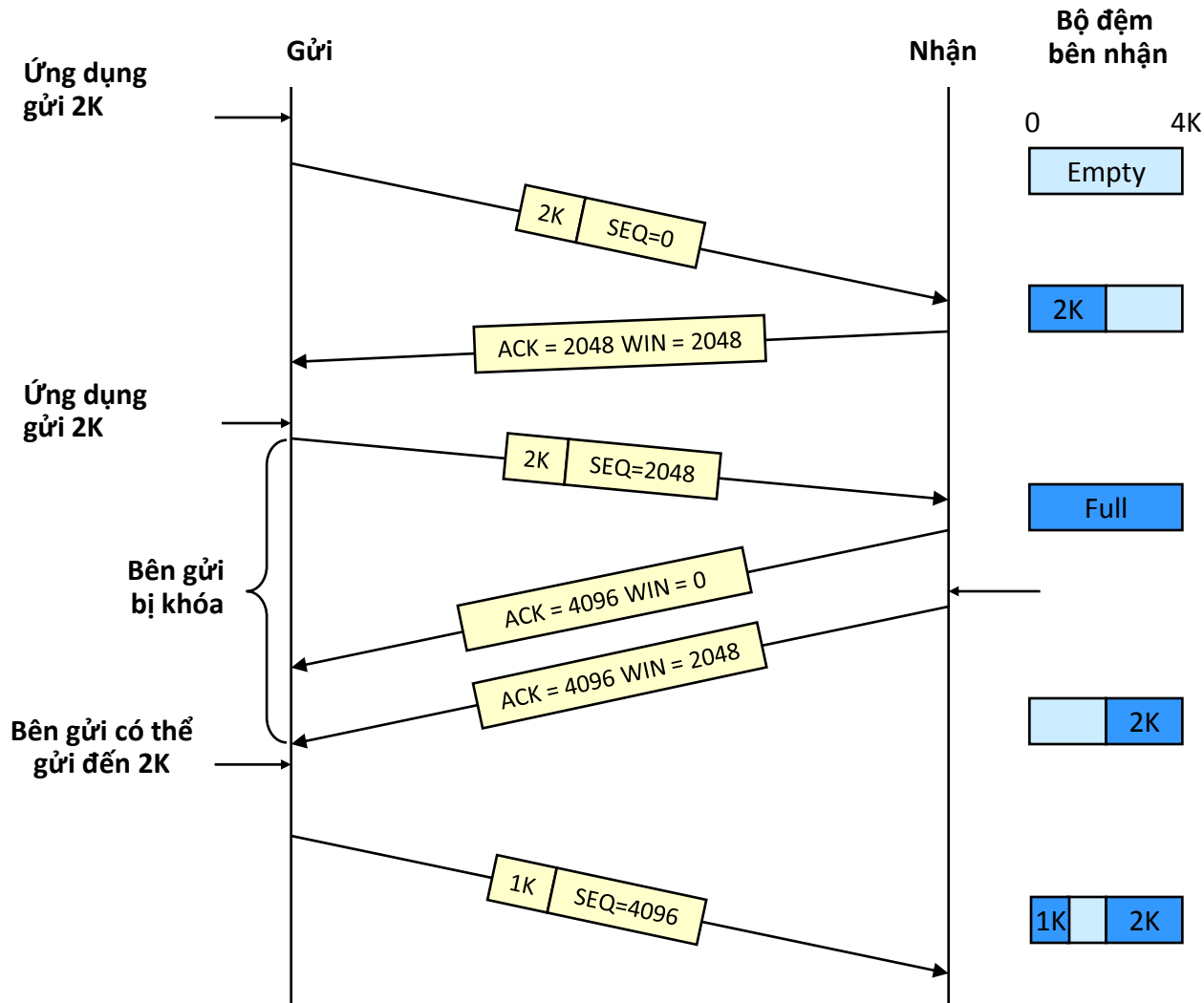
- Bên gửi làm tràn bộ đệm của bên nhận khi gửi quá nhiều dữ liệu hoặc gửi quá nhanh

□ Sử dụng trường "window size"

- Window size: lượng DL có thể đưa vào buffer



TCP - Điều khiển luồng - 2



Kiểm soát tắc nghẽn - 1



❑ Vấn đề: 1 node có thể nhận dữ liệu từ nhiều nguồn

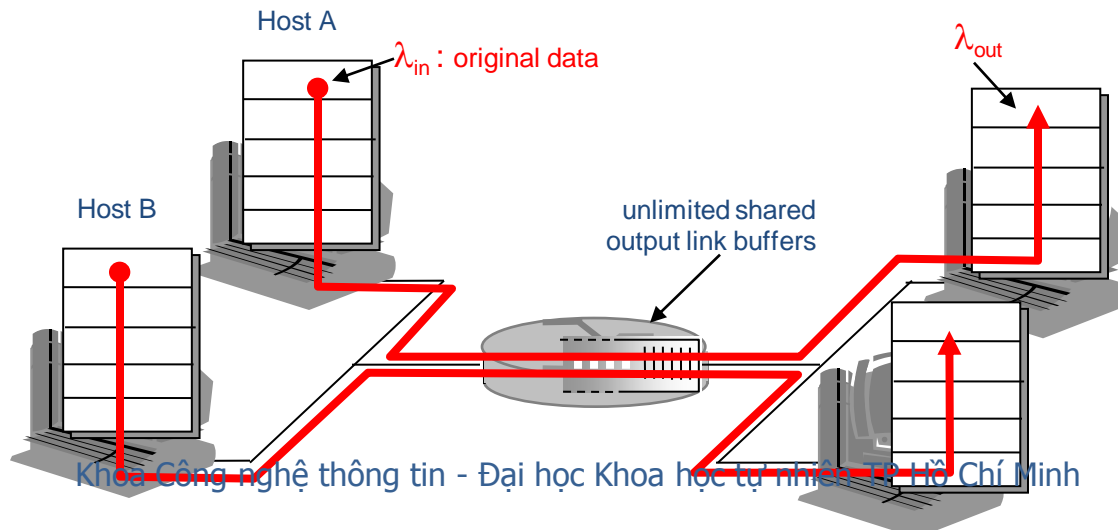
- Buffer: giới hạn
- gói tin: đến ồ ạt

➔ xử lý không kịp ➔ tắc nghẽn

❑ Hiện tượng:

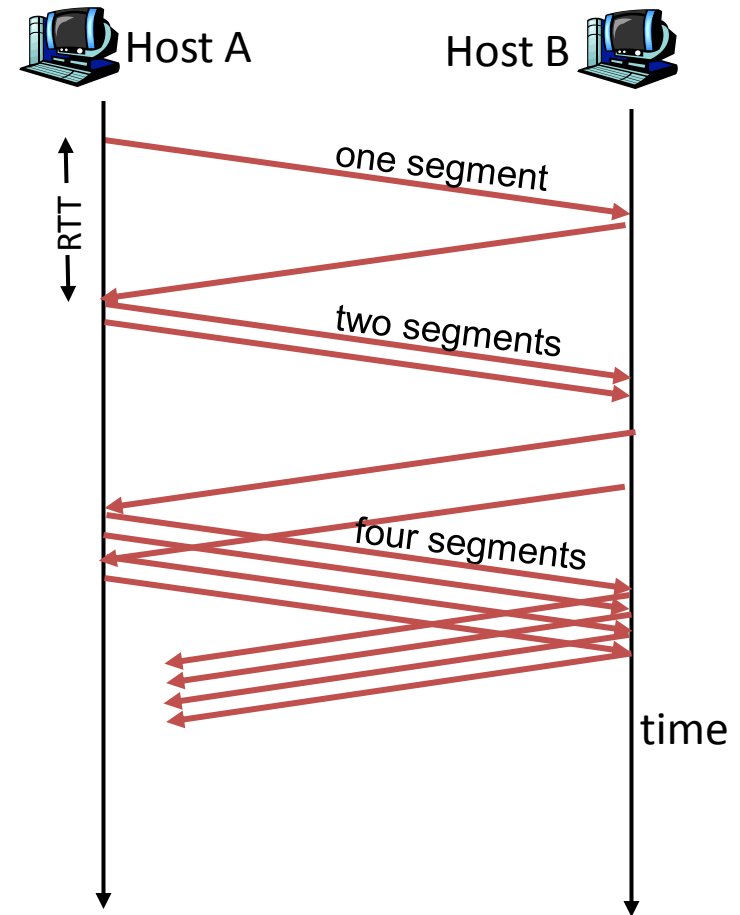
- Mất gói
- Delay cao

➔ Sử dụng đường truyền không hiệu quả



□ Giải quyết trong TCP:

- Bên gửi:
 - Thiết lập tốc độ gửi dựa trên phản hồi từ bên nhận
 - Nhận ACK
 - Mất gói
 - Độ trễ gói tin
- Tốc độ gửi: có 2 pha
 - Slow-Start
 - Congestion Avoidance



Tài liệu tham khảo



- ❑ Bài giảng của J.F Kurose and K.W. Ross về Computer Networking: A Top Down Approach





Chương 03

Tầng mạng

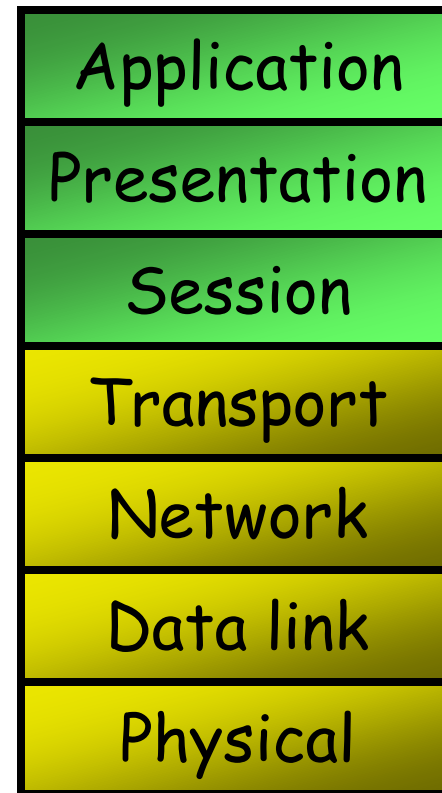
MẠNG MÁY TÍNH

Tháng 09/2011

Mục tiêu



- ❑ Thiết lập kết nối giữa 2 host để truyền dữ liệu từ host - host



Tầng mạng vs tầng vận chuyển



- ❑ *Tầng mạng*: cung cấp kết nối logic giữa các host
- ❑ *Tầng vận chuyển*: cung cấp kết nối logic giữa các tiến trình
 - Dựa trên, mở rộng dịch vụ của tầng mạng

Ví dụ:

A gửi B 1 bức thư qua đường bưu điện

- processes = A, B
- app messages = bức thư
- hosts = nhà của A, nhà của B
- transport protocol ???
- network-layer protocol???



Nội dung



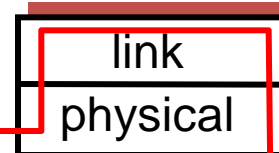
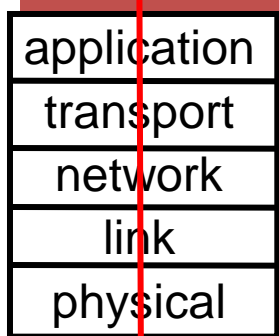
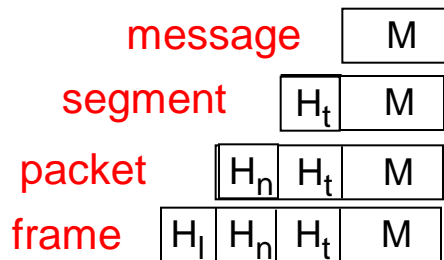
- Giới thiệu
- Định tuyến – chuyển tiếp
- Giao thức IP
- Giao thức ICMP
- Giao thức NAT



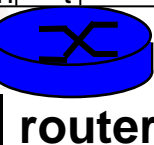
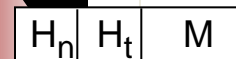
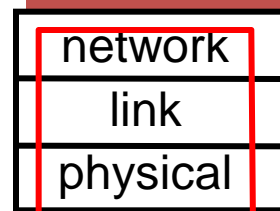
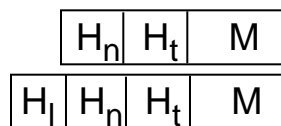
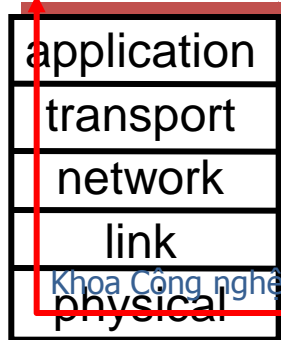
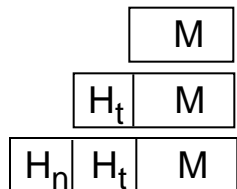
Nhắc lại



source



destination



giới thiệu - 1



- ❑ Thực hiện chuyển các segment từ host gửi đến host nhận
- ❑ Tại host gửi:
 - Nhận các segment từ transport layer
 - Đóng gói thành các packet
- ❑ Tại host nhận:
 - Nhận các packet từ data link layer
 - Chuyển các segment lên transport layer
- ❑ Tại các router:
 - Dựa vào **thông tin đích đến** để chuyển các packet đến host nhận
 - Định tuyến: quyết định gói tin đi đường nào
 - Chuyển tiếp: chuyển gói tin từ interface nhận ra interface gửi



- Tầng mạng cung cấp 2 loại dịch vụ
 - Hướng kết nối (Connection)
 - Virtual Circuit
 - Trước khi truyền dữ liệu, 2 host phải thiết lập kết nối
 - Hướng không kết nối (Connectionless)
 - Datagram Network
 - Không cần thiết lập kết nối trước khi gửi
- Trong 1 kiến trúc mạng: chỉ hỗ trợ duy nhất 1 loại dịch vụ

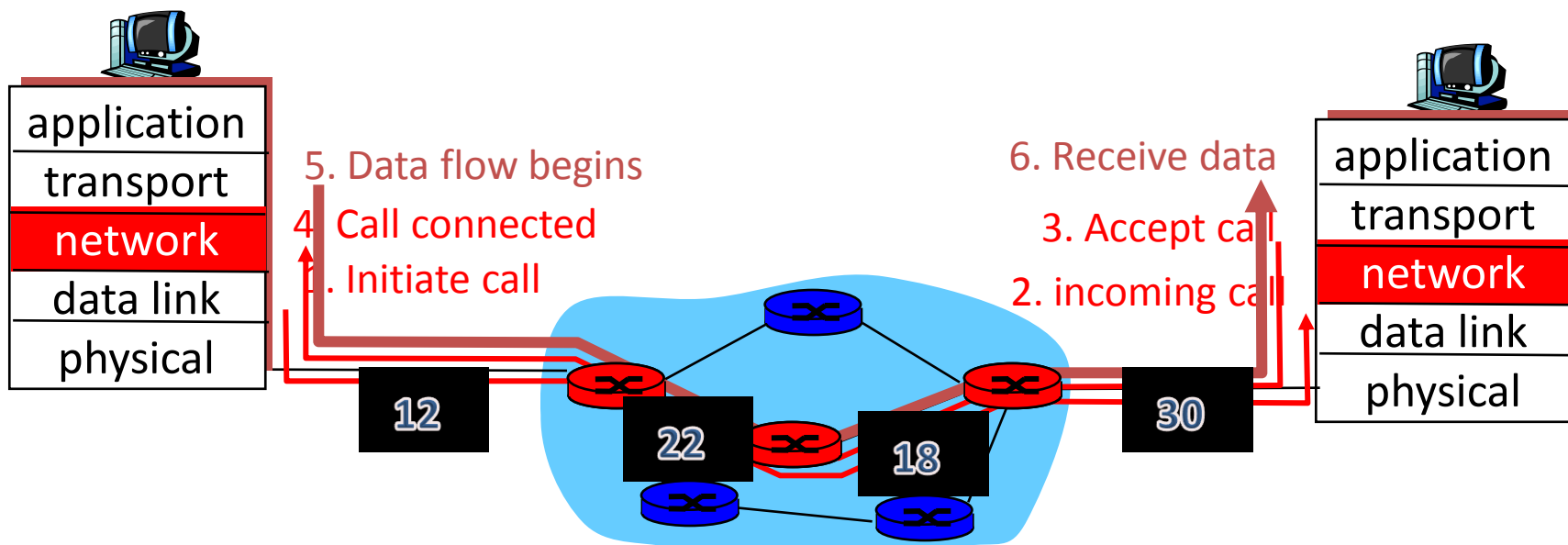
Virtual circuit (VC) network - 1



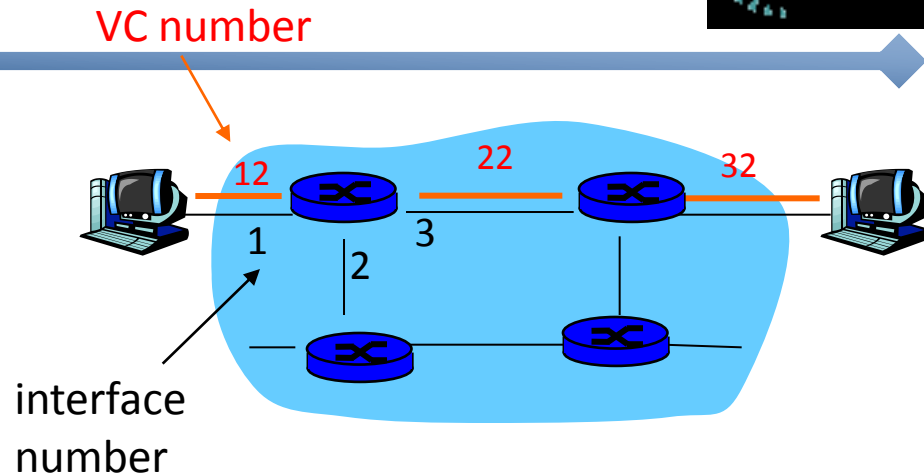
- ❑ Thiết lập, quản lý, duy trì mỗi kết nối khi truyền dữ liệu
 - 1 đường đi ảo khi truyền dữ liệu
 - Số hiệu VC (VC number)
 - Khác nhau trên mỗi link
 - Mỗi gói tin có một virtual circuit identifier (VC ID)
 - Các router duy trì trạng thái kết nối đi qua
 - bảng chuyển đổi VC ID
 - Thay thế thông tin VD ID của gói tin đi ngang qua router
- ❑ Thông tin định tuyến: Virtual Circuit number (VC ID)
- ❑ Dùng trong ATM, X.25, Frame-Relay,...



Virtual circuit (VC) network - 2



Virtual circuit network - 3



Cổng vào	VC# vào	Cổng ra	VC# ra
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Routers duy trì thông tin về trạng thái kết nối!



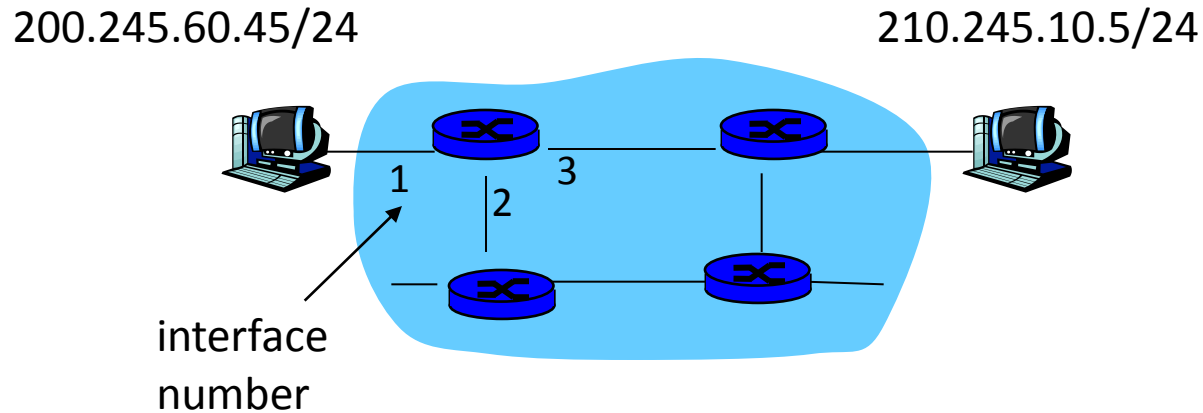
Datagram network - 1



- ❑ Không thiết lập kết nối trước khi truyền dữ liệu
 - Router không cần quản lý trạng thái kết nối
- ❑ Thông tin định tuyến: địa chỉ đích đến
 - Mỗi router duy trì một bảng định tuyến
- ❑ Dùng trong Internet



Datagram network - 2



Destination Network	Subnetmask	Out Interface	Next hop
210.245.10.0	255.255.255.0	3
210.245.15.0	255.255.255.0	1
210.245.15.192	255.255.255.192	2
...



Nội dung



- Giới thiệu
- Định tuyến – chuyển tiếp
- Giao thức IP
- Giao thức ICMP
- Giao thức NAT



Định tuyến - Chuyển tiếp - 1



□ Định tuyến:

- Quyết định “lộ trình” mà gói tin di chuyển từ host nguồn đến host đích đến
- Sử dụng thông tin toàn cục

□ Chuyển tiếp:

- Di chuyển gói tin từ cổng vào đến cổng ra
- Sử dụng thông tin cục bộ



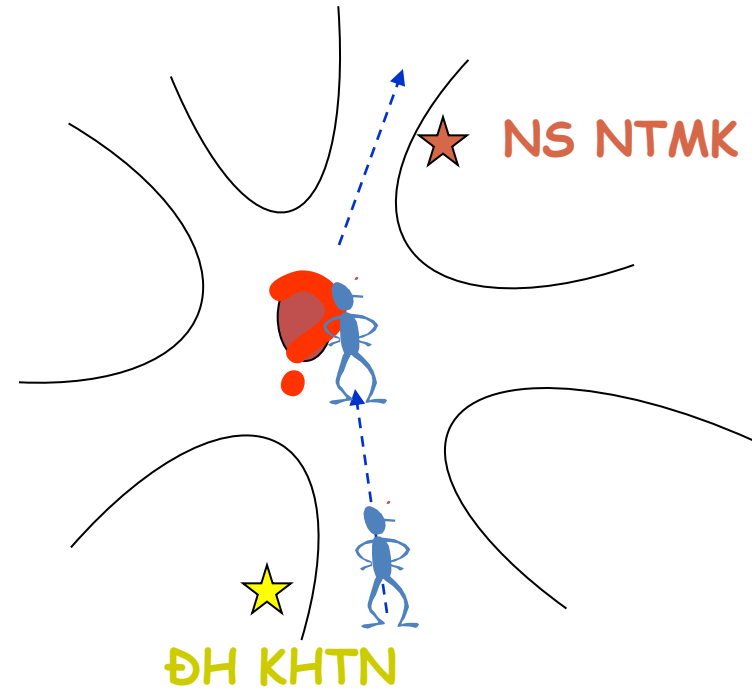
Định tuyến - Chuyển tiếp - 2



ĐH KHTN



NS NVCừ



Vạch ra lộ trình đi: NVCừ → NTMKhai



Định tuyến - 1



- ❑ Được thực hiện bởi các bộ định tuyến.
 - VD: router
- ❑ Dùng bảng định tuyến (routing/forwarding table)
 - destination/subnetmask
 - Out interface
 - next hop
 - chi phí
 - Hop count
 - Delay
 - Bandwidth
 - ...

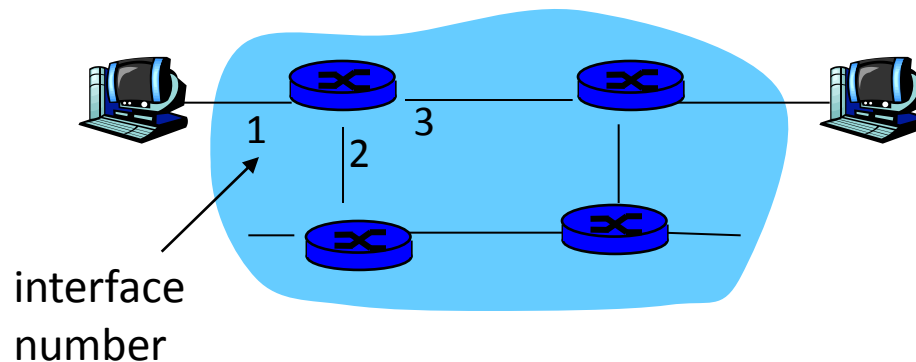


Ví dụ - định tuyến



200.245.60.45/24

210.245.10.5/24



Destination Network	Subnet mask	Next hop	Out Interface
210.245.10.0	255.255.255.0	192.168.3.2	3
210.245.15.0	255.255.255.0	192.168.1.2	1
210.245.15.192	255.255.255.192	192.168.2.2	2
...



Định tuyến - 2



□ Router định tuyến một gói tin như thế nào?

- Dùng địa chỉ đích đến và bảng định tuyến
- Thực hiện:
 - Tìm record thích hợp trong bảng định tuyến
 - Tính địa chỉ đường mạng giữa địa chỉ đích đến với subnetmask của từng record
 - So sánh destination network với địa chỉ đường mạng vừa tính
 - Gửi gói tin theo thông tin của record tìm được

□ VD: R1 nhận gói tin có destination 210.245.10.5

- 255.255.255.192
 - Net: 210.245.10.0 → không có record thoả
- 255.255.255.0
 - Net: 210.245.10.0 → record số 1 thoả
 - gói tin chuyển ra interface số 3 và nơi nhận gói tin tiếp theo là 192.168.3.2



Bảng định tuyến



□ Xây dựng bảng định tuyến:

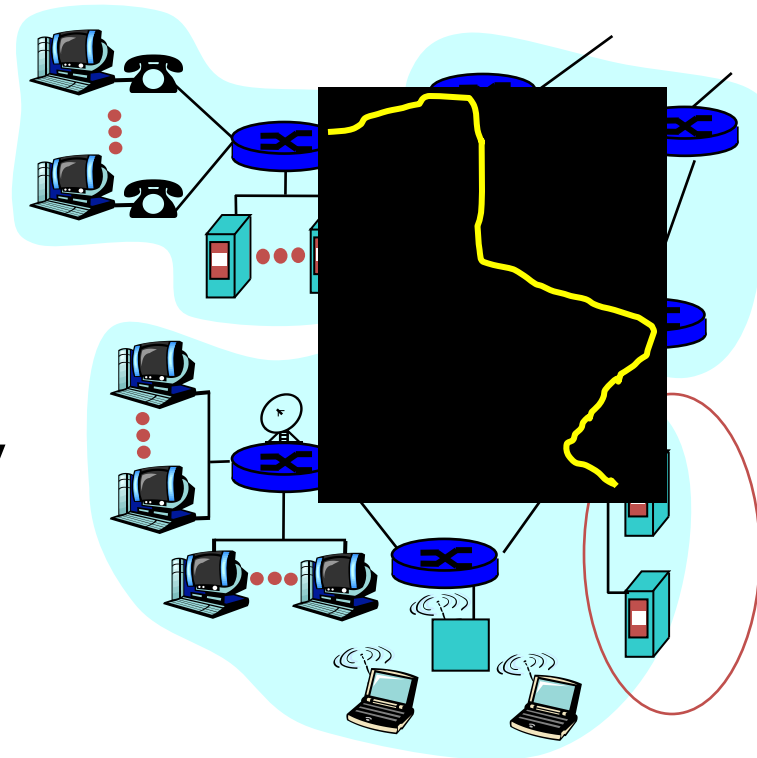
- Tĩnh (static): con người tự thiết lập
- Động (dynamic): học
 - Distance Vector:
 - Gửi theo định kỳ
 - Gửi toàn bộ bảng định tuyến
 - VD: RIP, IGRP, ...
 - Link State:
 - Gửi khi có thay đổi
 - Gửi tình trạng kết nối
 - VD: OSPF, ISIS, ...



Static route



- ❑ Biết: Sơ đồ mạng
- ❑ Xây dựng:
 - Vẽ "đường đi" *tối ưu*
- ❑ Khi có thay đổi:
 - Tự cập nhật bằng tay



Dynamic route



❑ Biết: không

❑ Xây dựng:

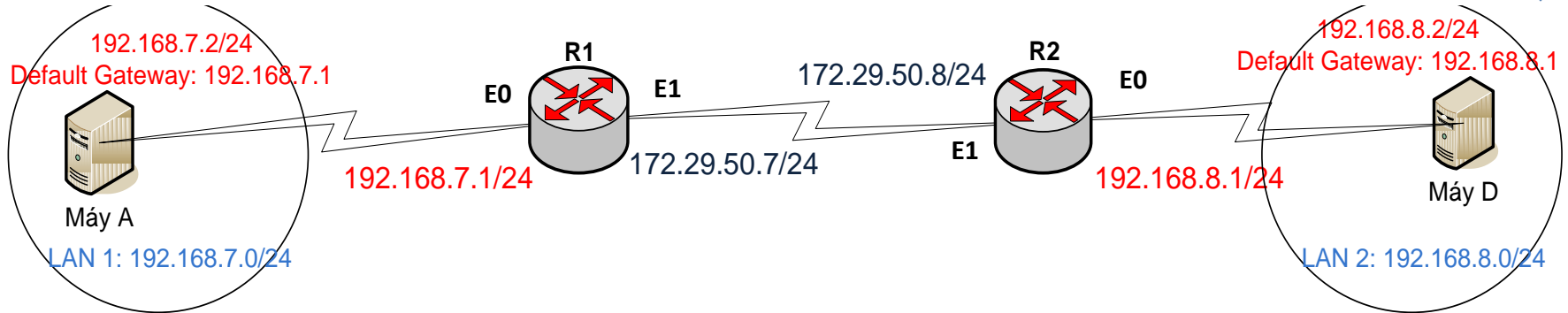
- Sử dụng các giao thức định tuyến
 - Thông qua các gói tin “thu thập” thông tin
 - Thành phần:
 - Gửi và nhận thông tin từ các router khác
 - Tính đường đi tối ưu
 - Phản ứng khi có thay đổi

❑ Khi thay đổi

- Cập nhật tự động



Static route - Ví dụ - 1



Tại router R1:

Destination network	Out interface	Next hop
192.168.8.0/24	E1	172.29.50.8

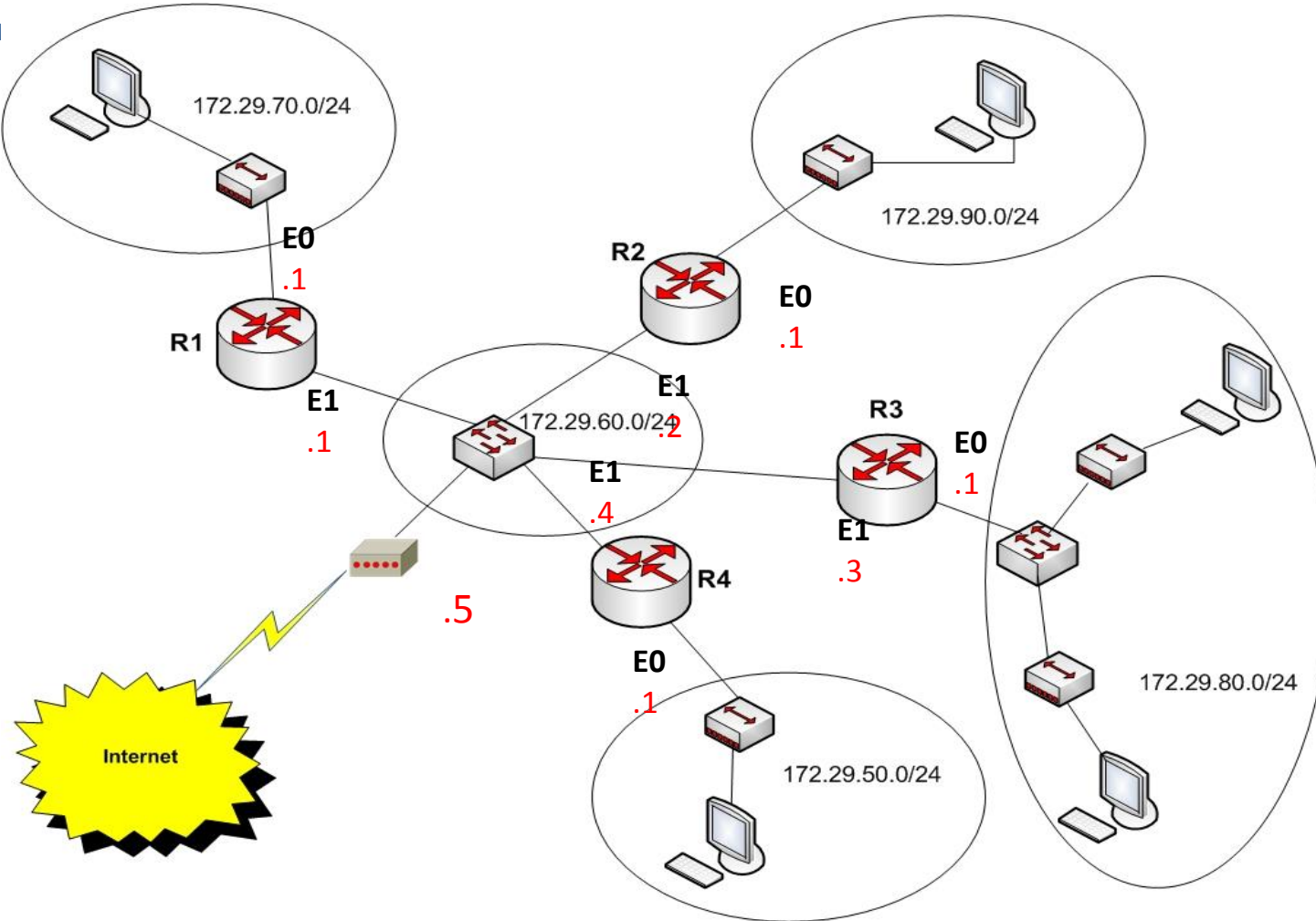
Tại router R2:

Destination network	Out interface	Next hop
192.168.7.0/24	E1	172.29.50.7

Khoa Công nghệ thông tin - Đại học Khoa học tự nhiên TP Hồ Chí Minh



Static route – ví dụ 2



Static route – ví dụ 2



Tại router R1:

Destination network	Out interface	Next hop
172.29.90.0/24	E1	172.29.60.2
172.29.80.0/24	E1	172.29.60.3
172.29.50.0/24	E1	172.29.60.4
0.0.0.0/0	E1	172.29.60.5

Tại router R2:

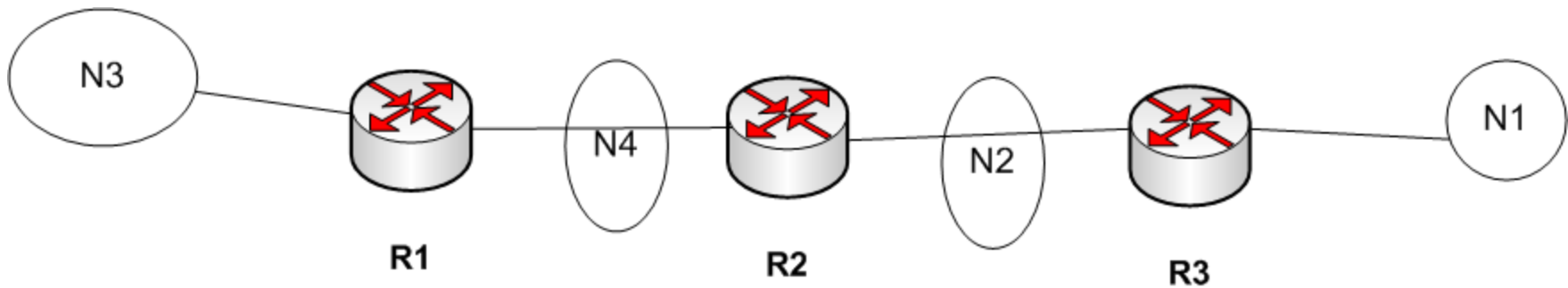
Destination network	Out interface	Next hop
172.29.70.0/24	E1	172.29.60.1
172.29.80.0/24	E1	172.29.60.3
172.29.50.0/24	E1	172.29.60.4
0.0.0.0/0	E1	172.29.60.5



Dynamic route – ví dụ



R2	
N2, N4	0 hop

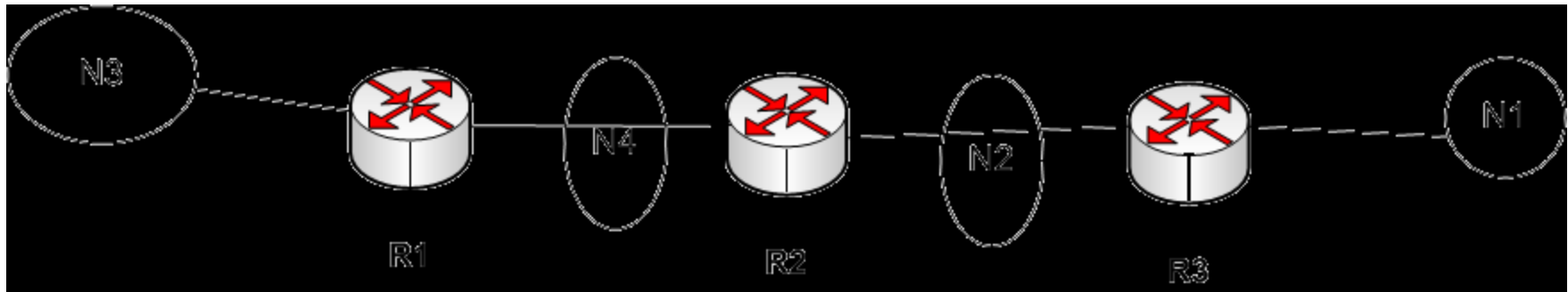


R1	
N3, N4	0 hop

R3	
N1, N2	0 hop



Dynamic route – ví dụ

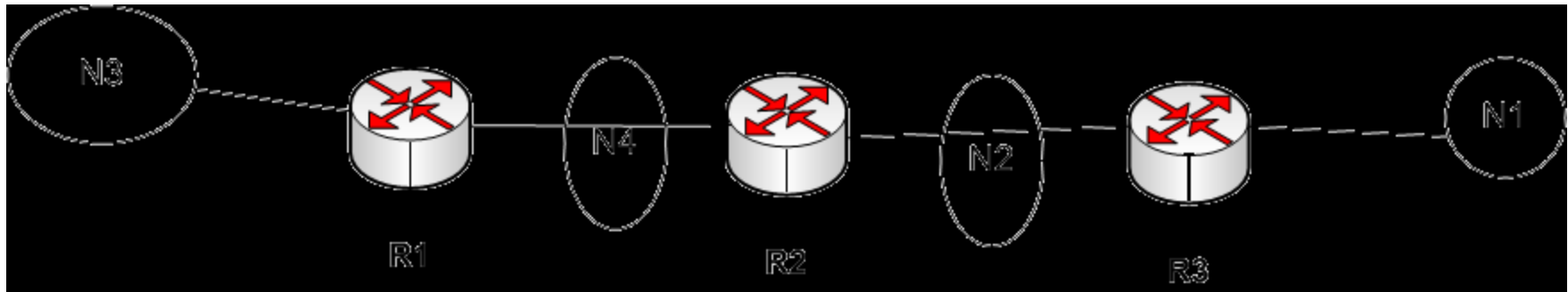


R1: N3, **N4** – 0 hop

R2	
N2, N4	0 hop
N3	1 hop



Dynamic route – ví dụ



R2: N2, N4 – 0 hop
N3 – 1 hop

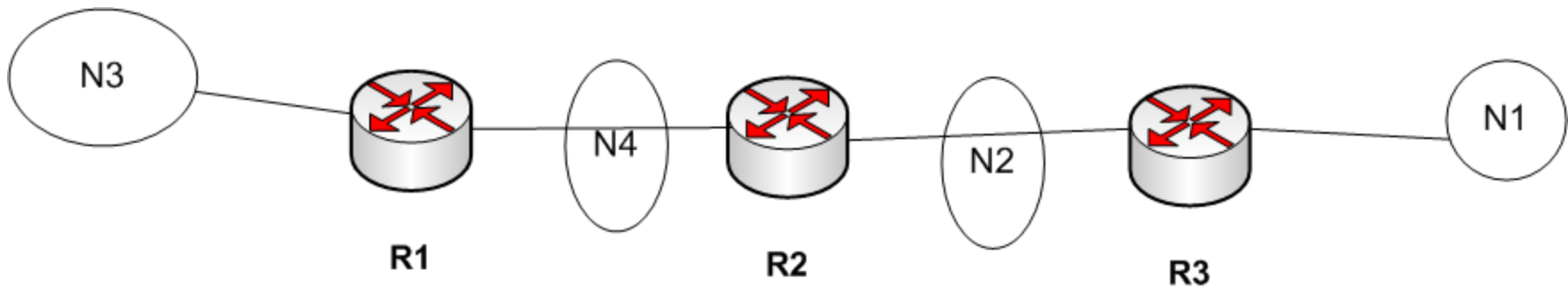
R3	
N1, N2	0 hop
N4	1 hop
N3	2 hops



Dynamic route – ví dụ



R2	
N2, N4	0 hop
N3, N1	1 hop

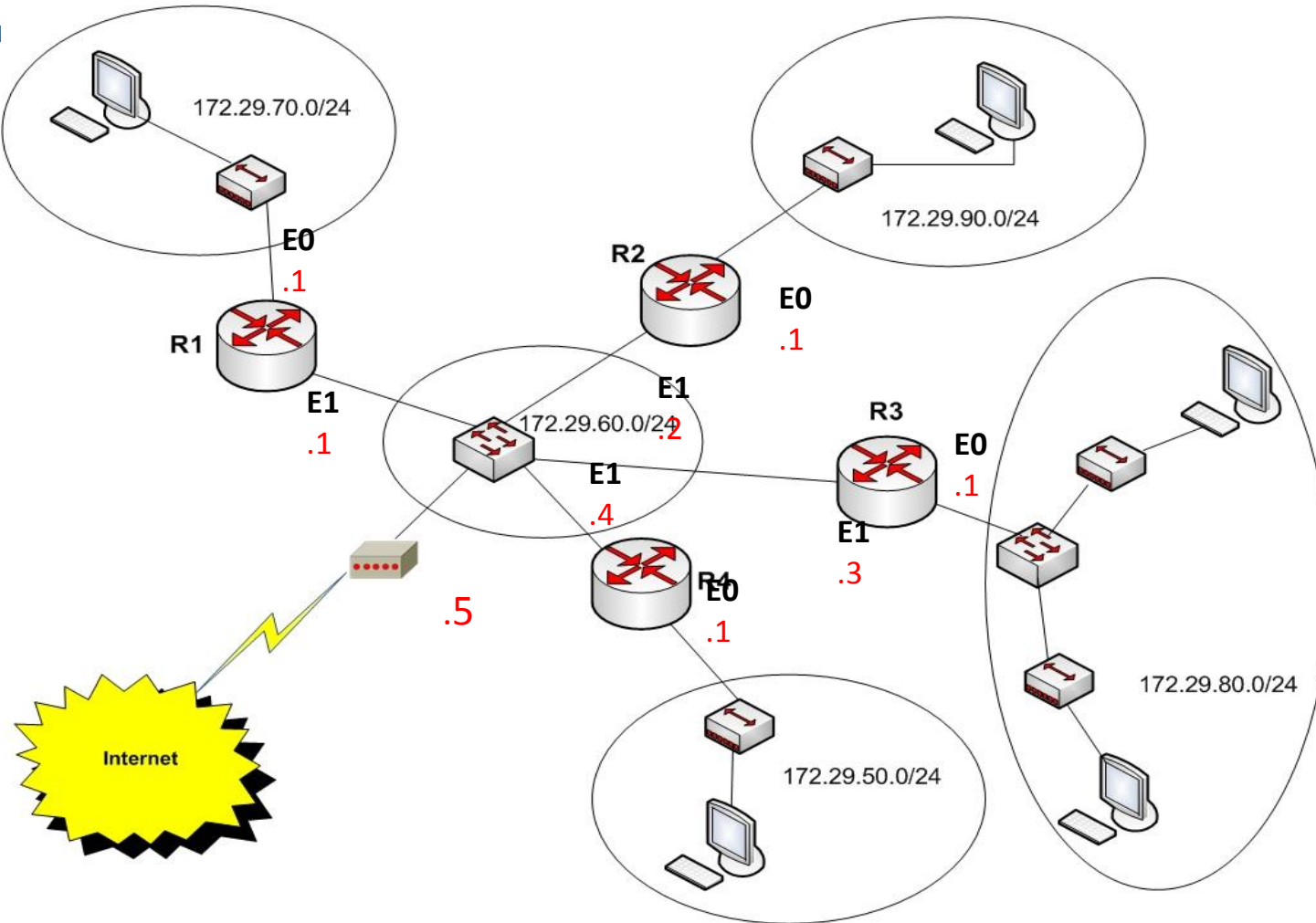


R1	
N3, N4	0 hop
N2	1 hop
N1	2 hops

R3	
N1, N2	0 hop
N4	1 hop
N3	2 hops



Dynamic route – ví dụ



Dynamic route – ví dụ



```
RI#show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP  
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area  
* - candidate default, U - per-user static route, o - ODR  
P - periodic downloaded static route
```

```
Gateway of last resort is not set
```

```
172.29.0.0/24 is subnetted, 5 subnets  
O      172.29.50.0 [110/2] via 172.29.60.4, 00:00:15, FastEthernet0/0  
C      172.29.60.0 is directly connected, FastEthernet0/0  
C      172.29.70.0 is directly connected, FastEthernet0/1  
O      172.29.80.0 [110/2] via 172.29.60.3, 00:00:15, FastEthernet0/0  
O      172.29.90.0 [110/2] via 172.29.60.2, 00:00:46, FastEthernet0/0
```



Nội dung



- Giới thiệu
- Định tuyến – chuyển tiếp
- Giao thức IP
- Giao thức ICMP
- Giao thức NAT



Routed protocol - 1



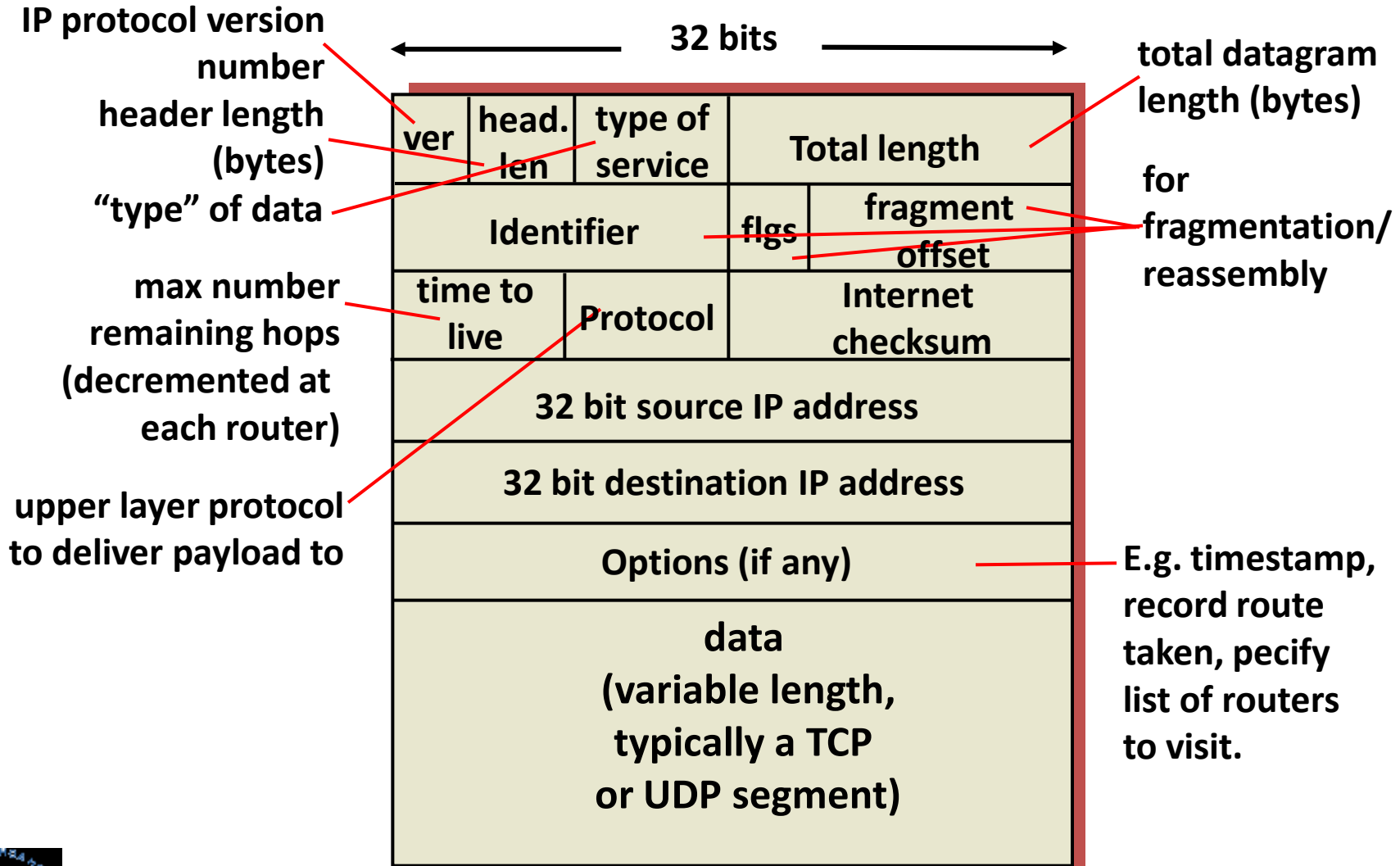
□ Giao thức được định tuyến (routed protocol):

- qui định cách thức đóng gói dữ liệu truyền trên đường truyền
- VD: IP (IPv4, IPv6), IPSec,...

Routing protocol	Routed protocol
Tạo bảng định tuyến	Đóng gói tin tại tầng mạng



Routed protocol - 2



Routed protocol - 3



- ❑ Version (4)
 - version của IP
- ❑ Header Length (4):
 - Chiều dài IP header (byte)
- ❑ Type of service (8)
 - Chứa định thông tin ưu tiên
 - Ít sử dụng
- ❑ Total length (16)
 - Tổng chiều dài của datagram (tính cả header) (byte)
- ❑ Identifier (16):
 - Khi một gói tin IP bị chia nhỏ ra thành nhiều đoạn, thì mỗi đoạn được gán cùng số ID
 - Dùng khi tổng hợp



Routed protocol - 4



❑ Flag (3)



▪ DF

- Don't fragment, không chia nhỏ

▪ MF

- More fragment, còn gói tin nhỏ tiếp
- Khi 1 gói tin bị chia nhỏ, tất cả các gói nhỏ (trừ gói tin cuối cùng), bit này được bật lên

❑ Fragment offset (13)

- Vị trí gói nhỏ trong gói tin ban đầu

❑ Time to live – TTL (8)

- Thời gian sống của gói tin (hop count)
- Giảm mỗi khi gói tin đến 1 router mới
 - Khi hop count = 0 thì gói tin bị loại bỏ



Routed protocol - 5



- ❑ Protocol (8)
 - Chỉ ra nghi thức nào ở tầng transport mà gói tin đang sử dụng
 - VD: TCP = 6, UDP = 17
- ❑ Internet (Header) checksum (16)
 - Kiểm tra tính đúng đắn nội dung của IP header
 - Không theo cách kiểm tra tuần tự
- ❑ Source and destination addr (32)
 - Địa chỉ IP của bên gửi và bên nhận
- ❑ Options (32)
 - Có thể dài đến 40 bytes
 - Dùng cho các tính năng mở rộng của IP
 - Vd: source routing, security, record route, ...
- ❑ Data:
 - Dữ liệu ở tầng transport gửi xuống



Nội dung



- Giới thiệu
- Định tuyến – chuyển tiếp
- Giao thức IP
- Giao thức ICMP
- NAT



Giao thức ICMP



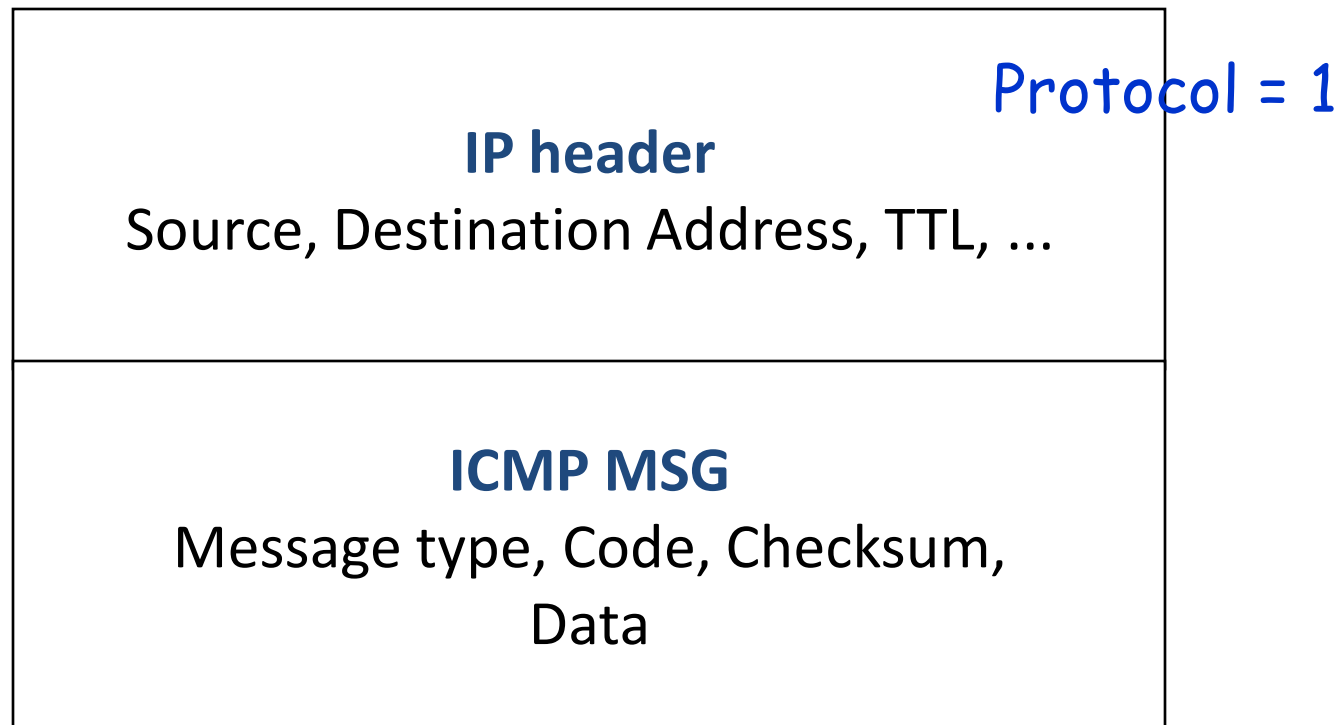
- ❑ ICMP (Internet Control Message Protocol)
- ❑ Được sử dụng bởi các host và router để trao đổi thông tin ở tầng mạng
 - Báo lỗi:
 - Mạng, host, protocol, port ... không vươn đến được
 - Báo mạng bị tắt nghẽn
 - Báo timeout
 - Echo request/reply (ping)



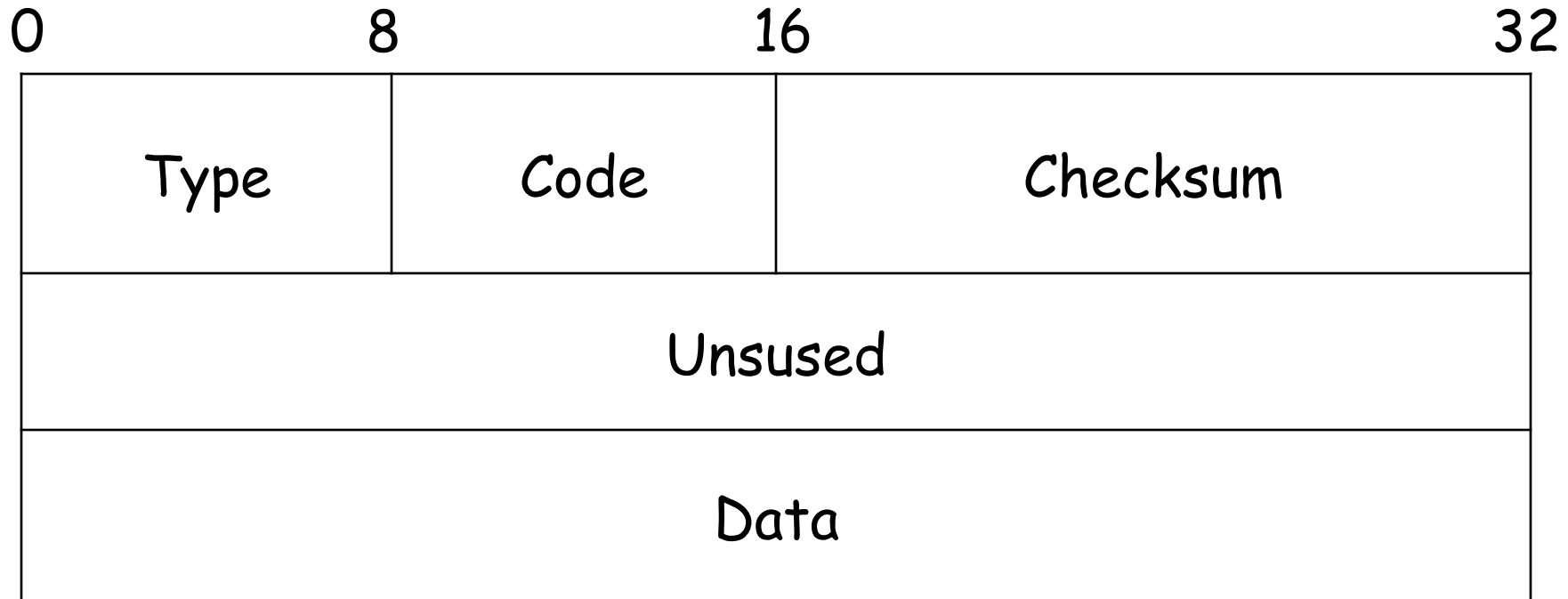
Gói tin ICMP



- Thông điệp ICMP được đóng gói trong gói tin IP



Cấu trúc thông điệp icmp - 1



Cấu trúc thông điệp icmp - 2



ICMP Type	Code	Description
0	0	echo reply
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad



Cấu trúc thông điệp icmp - 3



□ Không đến được đích:

- Nguyên nhân: liên kết mạng bị đứt, đích đến không tìm thấy, ...
- Type = 3
- Code:
 - 0: unreachable network
 - 1: unreachable host
 - 2: unreachable protocol
 - 3: unreachable port
 - 4: không được phép fragment
 - 5: source route bị sai



Cấu trúc thông điệp icmp - 4



□ Quá hạn:

- Nguyên nhân:
 - TTL = 0 trước khi đến đích
 - Quá hạn thời gian tái lắp ghép các fragment
- Type = 11
- Code:
 - 0: TTL
 - 1: hết thời gian tái lắp ghép



Giao thức ICMP



- ❑ Các trường hợp GỬI ICMP msg:
 - Datagram không đạt đến đích
 - Time out
 - Error xuất hiện trong header
 - Router/host bị tắt nghẽn
- ❑ Các trường hợp KHÔNG gửi ICMP msg:
 - Bản thân ICMP msg có lỗi
 - Broadcast, multicast (gói DL định tuyến)
 - Những fragment khác với fragment đầu tiên



Nội dung



- Giới thiệu
- Định tuyến – chuyển tiếp
- Giao thức IP
- Giao thức ICMP
- NAT



Nhắc lại



□ Địa chỉ IP:

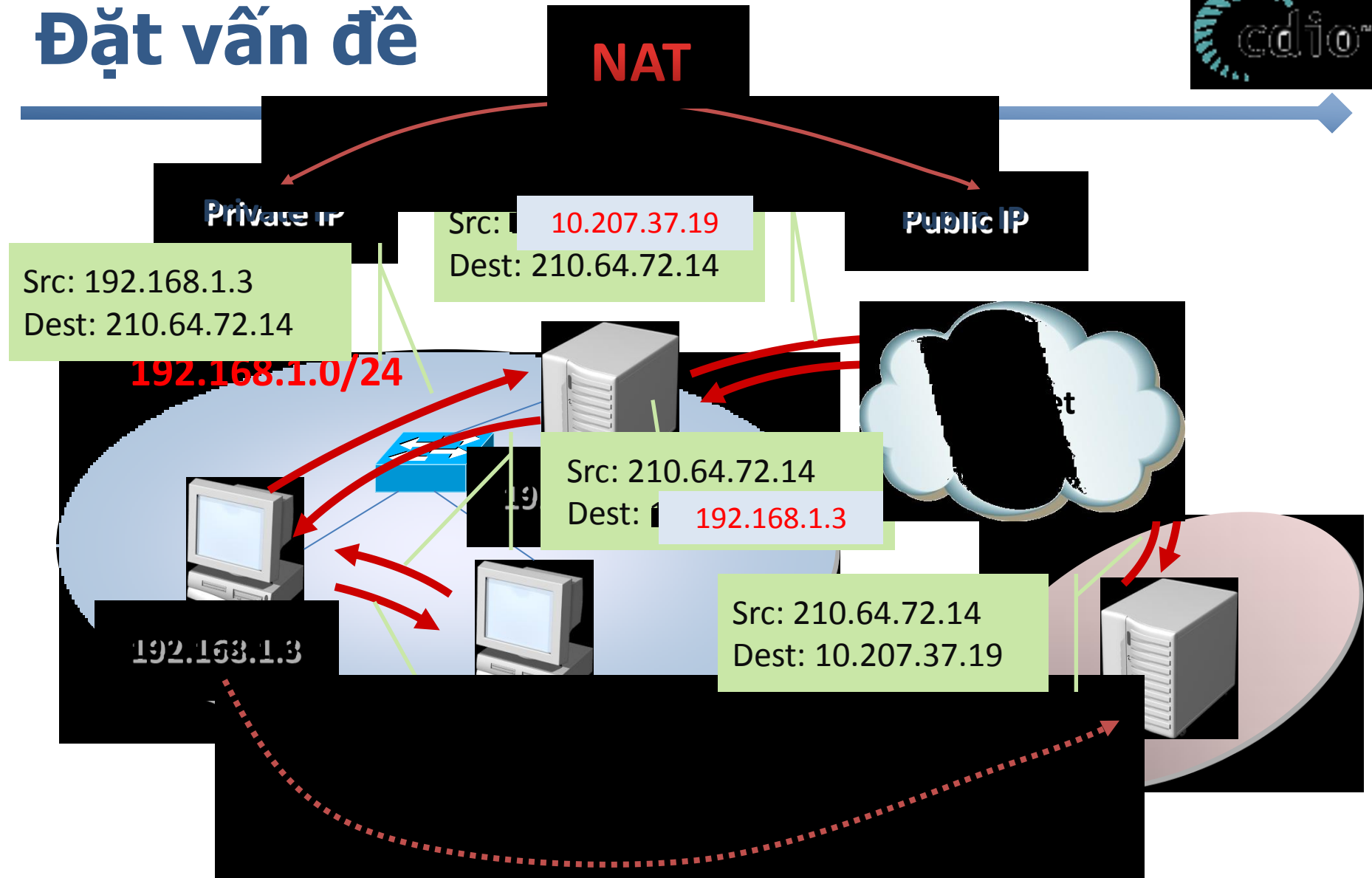
- Kích thước: 32 bits → không gian: 2^{32} địa chỉ
 - 0.x.x.x/8, 127.0.0.0/8, lớp D, lớp E; không dùng
 - Số lượng node trên Internet “khổng lồ”
- Giải quyết:
 - dùng địa chỉ private trong mạng LAN
 - Dùng địa chỉ public khi giao tiếp bên ngoài Internet

□ Gửi dữ liệu giữa 2 host

- Địa chỉ host gửi
- Địa chỉ host nhận



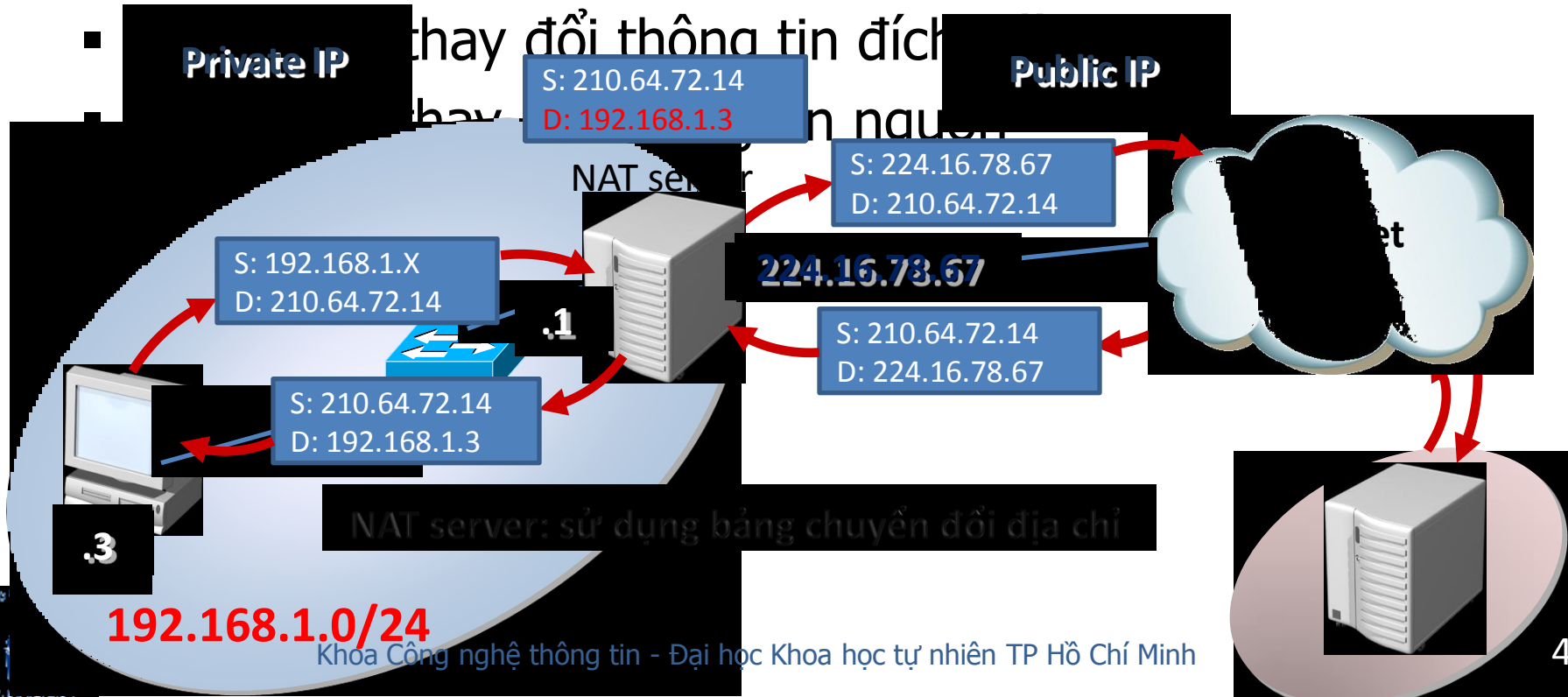
Đặt vấn đề



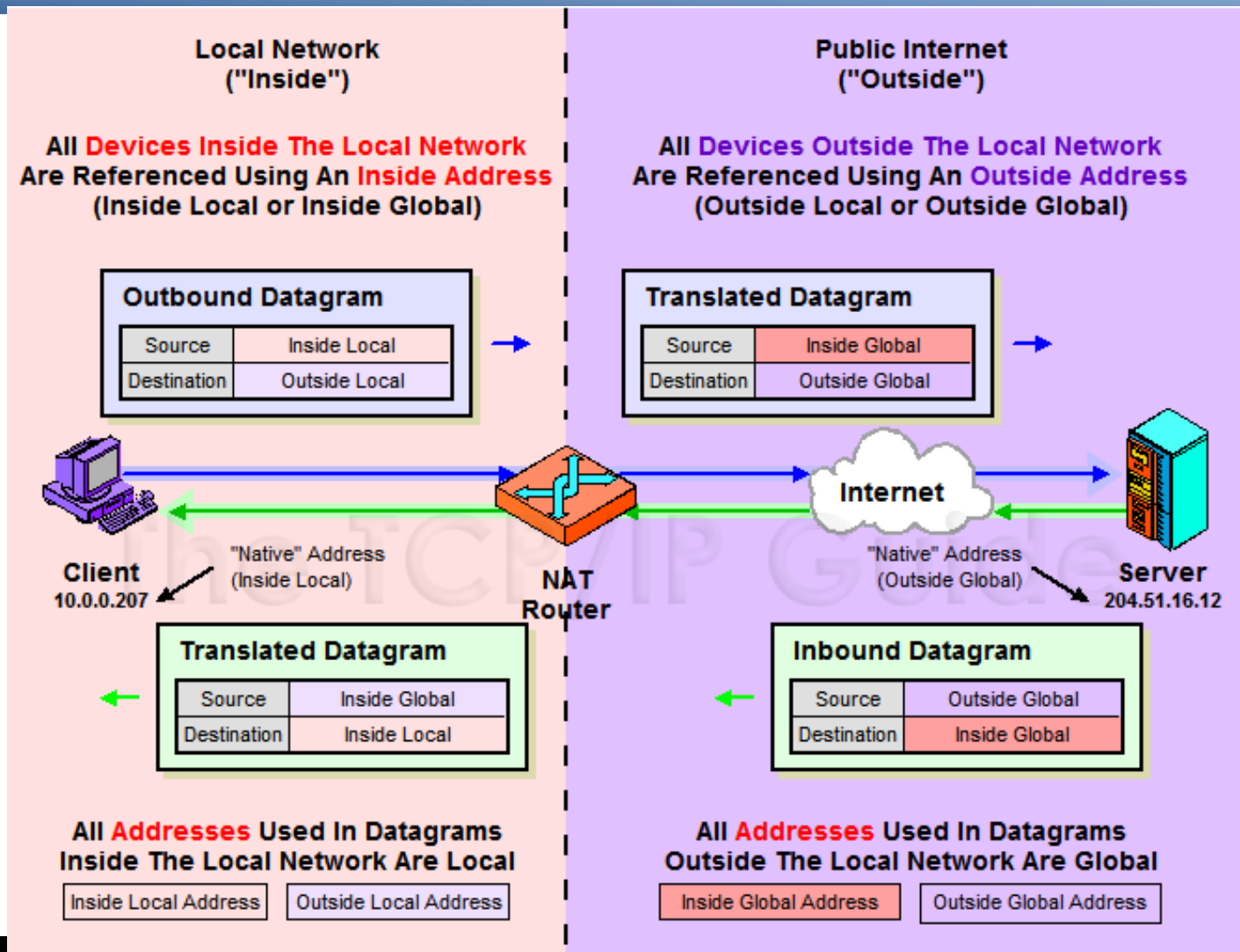
NAT – giới thiệu



- NAT = Network Address Translation
- RFC 1631, 1918, 2663
- Chức năng: “thay đổi” địa chỉ



NAT – thuật ngữ



NAT – bảng chuyển đổi địa chỉ



- ❑ Dùng chuyển đổi global <-> local
 - Thông tin cục bộ bên trong (Inside local)
 - Thông tin toàn cục bên trong (Inside global)
- ❑ Thông tin trong bảng chuyển đổi
 - Static
 - dynamic



Nat – phân loại



Static

- Cố định: 1 local IP \Leftrightarrow 1 global IP

Dynamic

- n local IP \Leftrightarrow m global IP
- NAT: chọn 1 global IP còn rảnh để NAT

Overloading

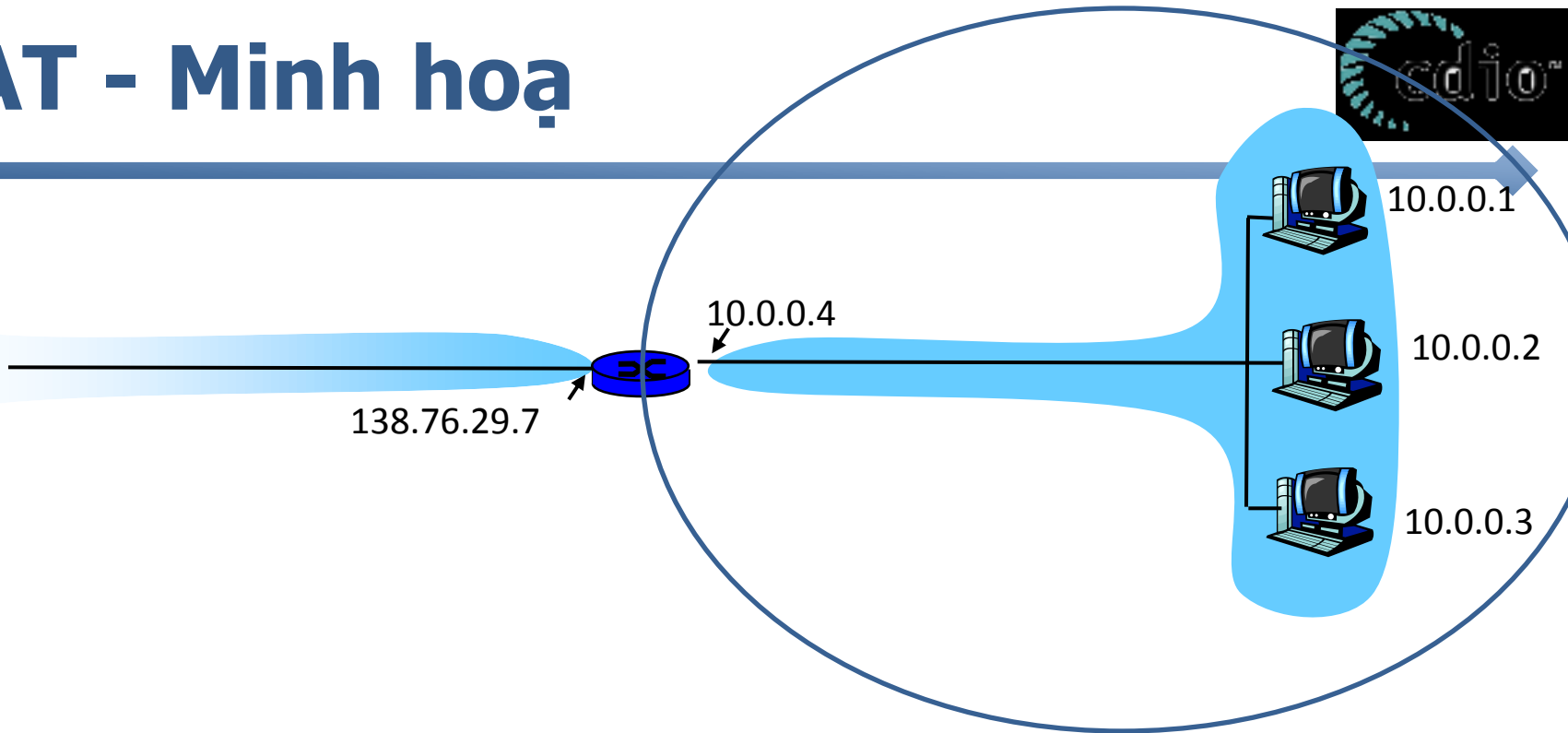
- n local IP \Leftrightarrow 1 global IP
- NAT: $\langle \text{local IP, local port} \rangle \Leftrightarrow \langle \text{global IP, global port} \rangle$

Overlapping

- Cố định: $\langle \text{local IP, } \textit{port} \rangle \Leftrightarrow \langle \text{global IP, } \textit{port} \rangle$



NAT - Minh họa



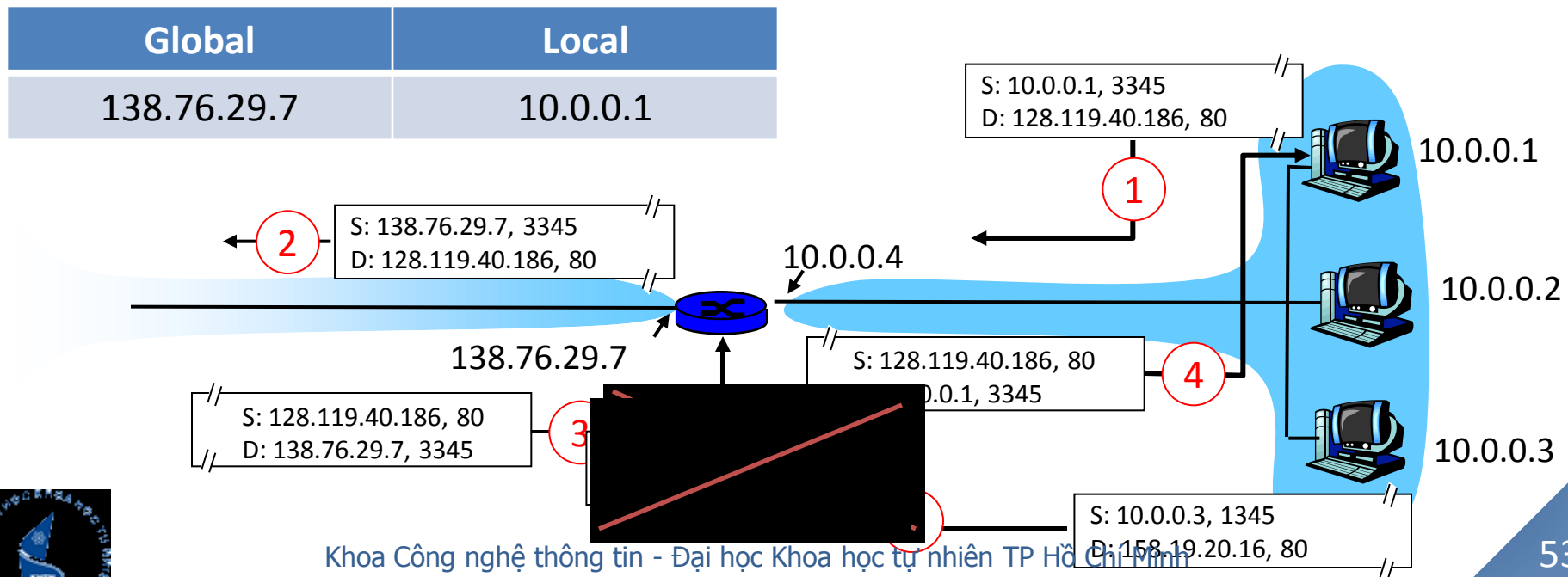
□ Thứ tự gửi các gói tin như sau:

- Máy 10.0.0.1 gửi 1 gói tin đến 128.119.40.186, 80 từ ứng dụng 3345
- Ứng dụng <128.119.40.186, 80> gửi lại gói tin phản hồi
- Máy 10.0.0.3 gửi 1 gói tin đến 158.19.20.16, 80 từ ứng dụng 1234
- Ứng dụng <120.11.40.18, 3345> gửi gói tin truy cập dịch vụ web tại máy 10.0.0.1

Static NAT



- Cấu hình **cố định**: 1 local IP \Leftrightarrow 1 global IP
 - Số máy kết nối ra ngoài bằng với số địa chỉ IP global
 - Bên ngoài (outside) có thể chủ động tạo kết nối với bên trong (inside)



Dynamic NAT

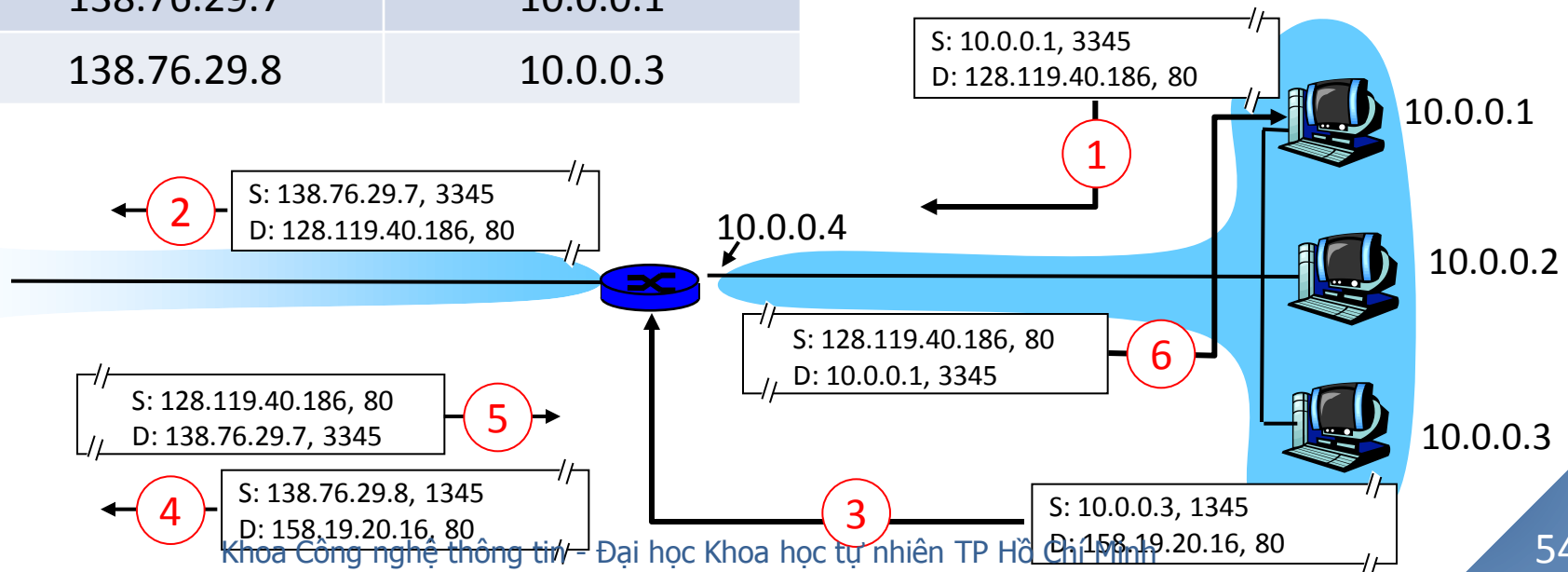


❑ Cấu hình: n local IP ⇔ m global IP

- Có m kết nối đồng thời
- Bên ngoài (outside) **không** thể chủ động tạo kết nối với bên trong (inside)

❑ Ví dụ: 10.0.0.0/24 ⇒ 138.76.29.7 và 138.76.29.8

Global	Local
138.76.29.7	10.0.0.1
138.76.29.8	10.0.0.3



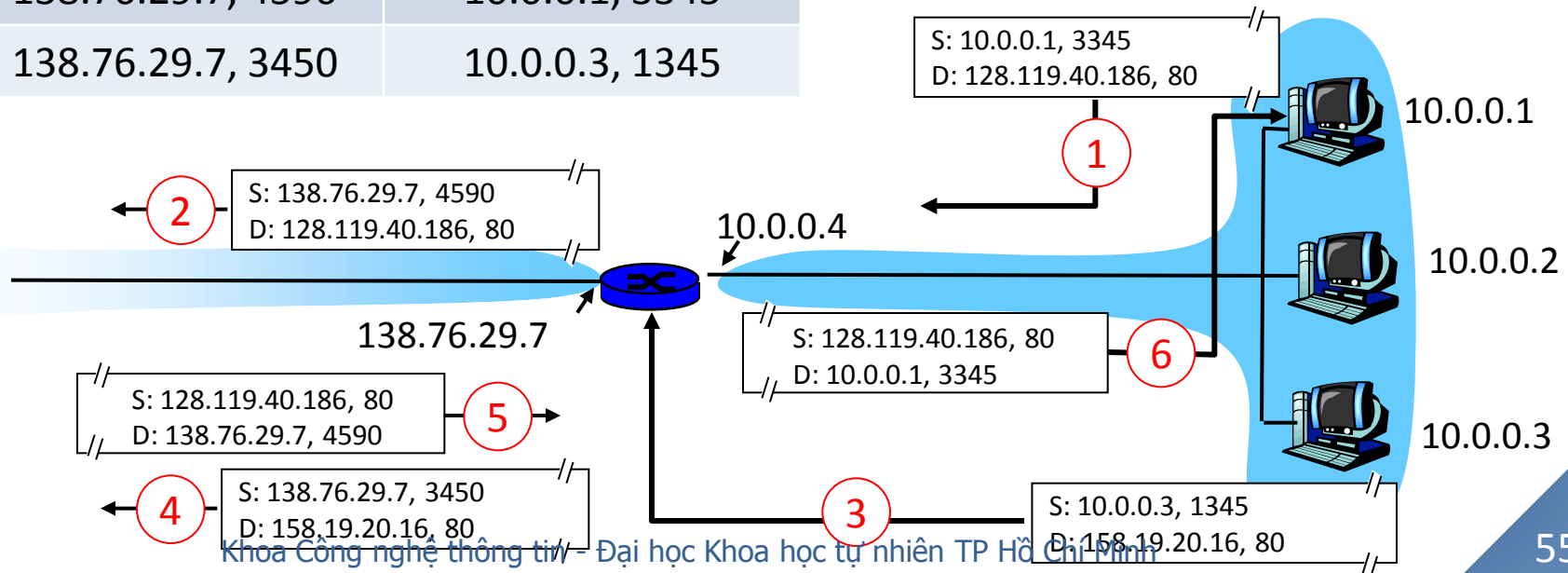
Overloading NAT



□ Cấu hình: n local IP \Leftrightarrow 1 global IP

- NAT: $\langle \text{local IP, local port} \rangle \Leftrightarrow \langle \text{global IP, global port} \rangle$
- Có n kết nối đồng thời
- Bên ngoài (outside) **không** thể chủ động tạo kết nối với bên trong (inside)

Global	Local
138.76.29.7, 4590	10.0.0.1, 3345
138.76.29.7, 3450	10.0.0.3, 1345

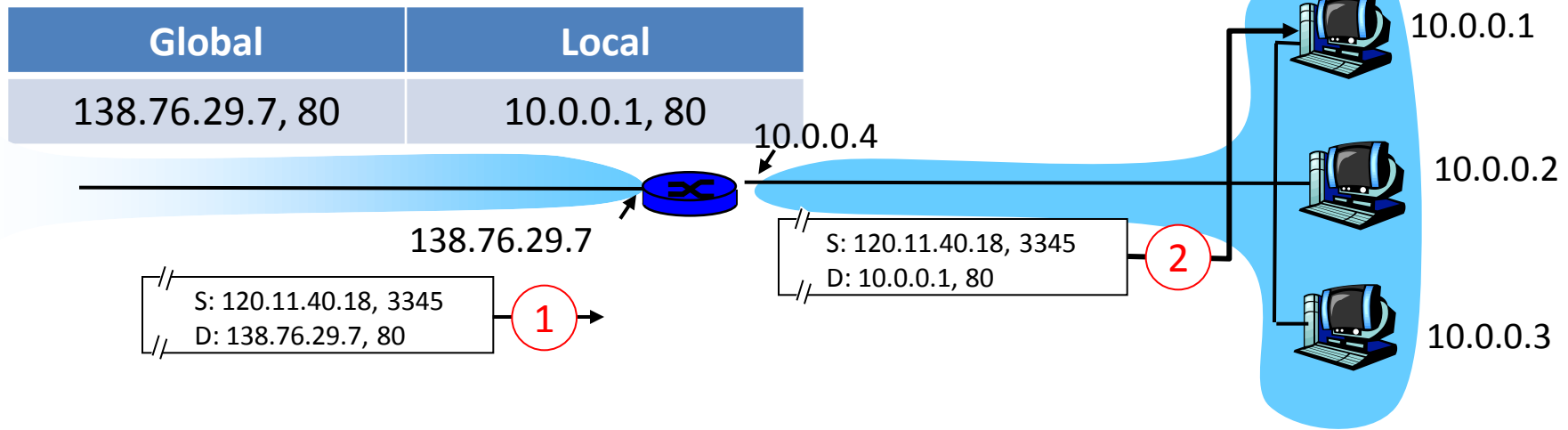


Overlapping NAT

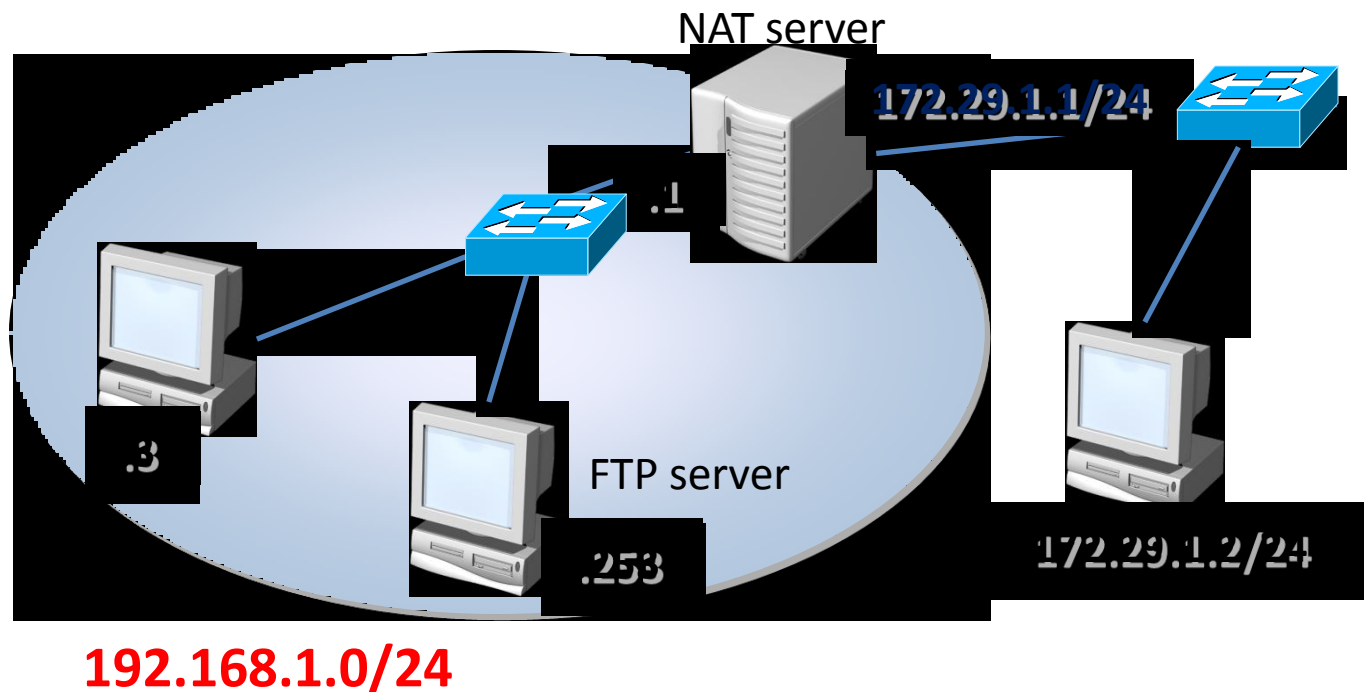


□ Cấu hình **cố định**: <local IP, port> ↔ <global IP, port>

- Bên ngoài (outside) có thể chủ động tạo kết nối với bên trong (inside)
- Dùng để publish một dịch vụ ra *ngoài*



NAT – mô tả bài toán



□ Yêu cầu:

- Các máy tính trong LAN: 192.168.1.0/24 có thể truy cập ra ngoài bằng IP: 172.29.1.1
- Bên ngoài có thể truy cập dịch vụ FTP trên máy 192.168.3.253



NAT – cấu hình trên wins 2k3



- Chọn card public và private
 - Private: 192.168.1.1
 - Public: 172.29.1.1
- Chọn dịch vụ để publish (nếu có): Web
 - Local IP: 192.168.1.253
 - Incoming port: 80
 - Outgoing port: 80



Tài liệu tham khảo



- ❑ Slide của J.F Kurose and K.W. Ross về Computer Networking: A Top Down Approach





Chương 03

Tăng liên kết dữ liệu

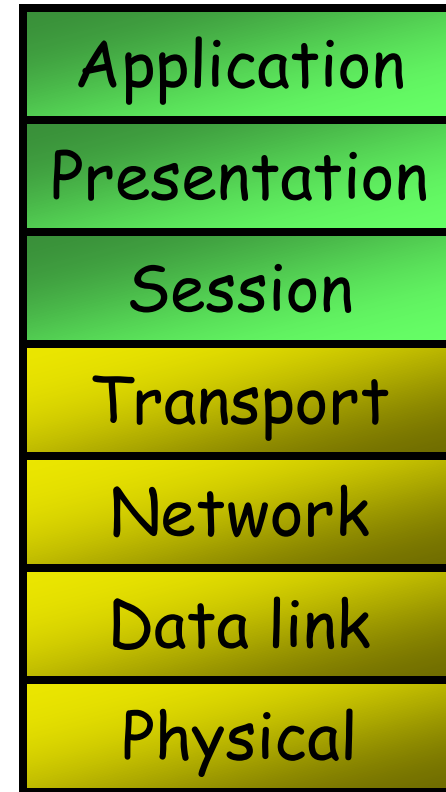
MẠNG MÁY TÍNH

Tháng 09/2011

Mục tiêu



- Điều khiển truy cập đường truyền
- Điều khiển liên kết



Nội dung



- Giới thiệu
- Kỹ thuật phát hiện và sửa lỗi
- Điều khiển truy cập đường truyền
- ARP
- Ethernet



Giới thiệu - 1

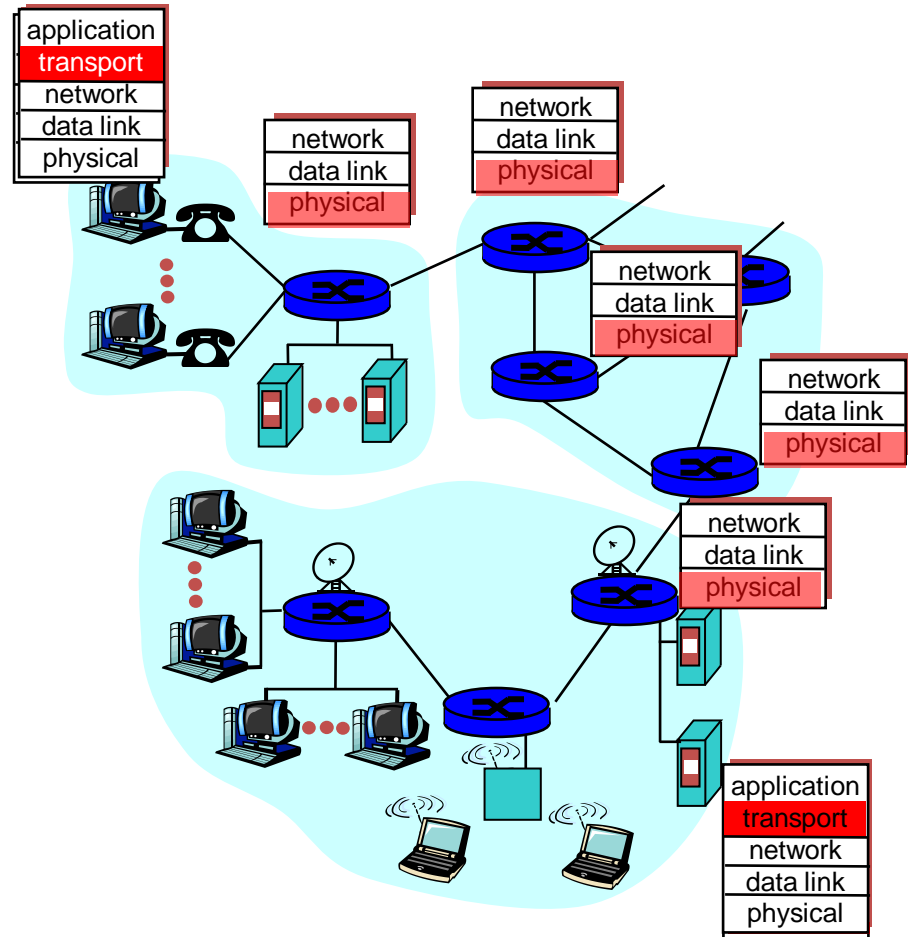


□ Link: “kết nối/liên kết” giữa các nodes kề nhau

- Wired
- Wireless

□ Data link layer: chuyển gói tin (frame) từ một node đến node kề qua 1 link

- Mỗi link có thể dùng giao thức khác nhau để truyền tải frame



Giới thiệu - 2



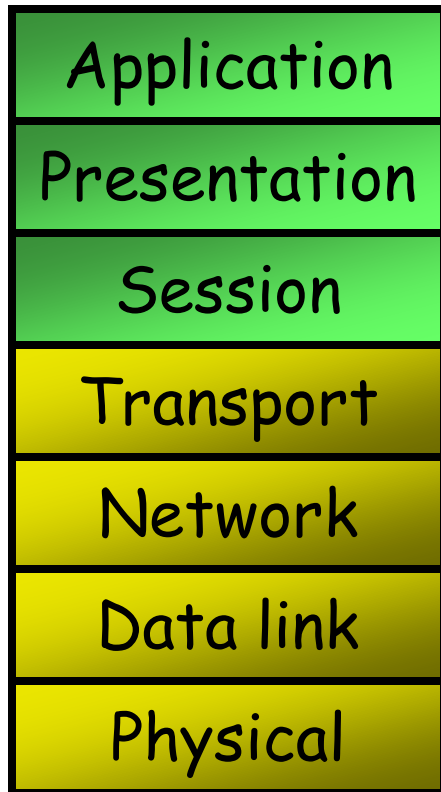
□ Tại nơi gửi:

- Nhận các packet từ tầng network → đóng gói thành các frame
- Truy cập đường truyền (nếu dùng đường truyền chung)

□ Tại nơi nhận:

- Nhận các frame dữ liệu từ tầng physical
- Kiểm tra lỗi
- Chuyển cho tầng network





❑ LLC (Logical Link Control)

- Điều khiển luồng
- Kiểm tra lỗi
- Báo nhận

❑ MAC (Media Access Control)

- Truy cập đường truyền

Logical Link Control

Media Access Control

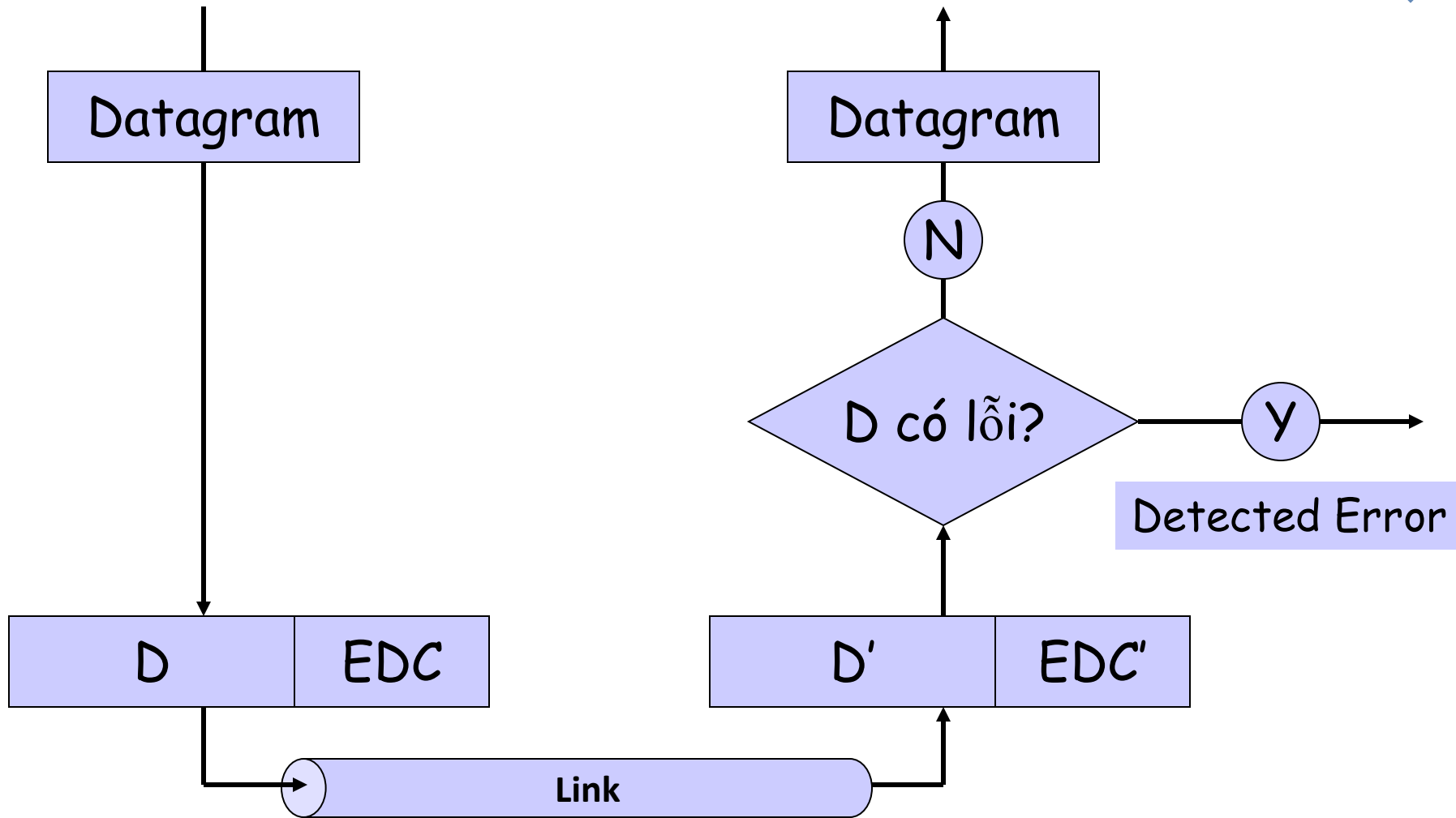
Nội dung



- Giới thiệu
- Kỹ thuật phát hiện và sửa lỗi
- Điều khiển truy cập đường truyền
- ARP
- Ethernet



Kỹ thuật phát hiện và sửa lỗi - 1



Kỹ thuật phát hiện và sửa lỗi - 2



□ Các phương pháp:

- Parity Check (bit chẵn lẻ)
- Checksum
- Cyclic Redundancy Check (CRC)



Parity Check



❑ Dùng thêm một số bit để đánh dấu tính chẵn lẻ

- Dựa trên số bit 1 trong dữ liệu
- Phân loại:
 - Even Parity: số bit 1 phải là một số chẵn
 - Odd Parity: số bit 1 phải là một số lẻ

❑ Các phương pháp:

- Parity 1 chiều
- Parity 2 chiều
- Hamming code



Parity 1 chiều - 1



- Số bit parity: 1 bit
- Chiều dài của dữ liệu cần gửi đi: d bit
→ DL gửi đi sẽ có $(d+1)$ bit
- Bên gửi:

- Thêm 1 bit parity vào dữ liệu cần gửi đi
 - Mô hình chẵn (Even parity)
 - số bit 1 trong $d+1$ bit là một số chẵn
 - Mô hình lẻ (Odd Parity)
 - số bit 1 trong $d+1$ bit là một số lẻ

$\overbrace{0111000110101011}^{d \text{ bits}} \quad \overbrace{1}^{\text{Parity bit}} \quad (\text{mô hình chẵn})$
 $\overbrace{0111000110101011}^{d \text{ bits}} \quad \overbrace{0}^{\text{Parity bit}} \quad (\text{mô hình lẻ})$



Parity 1 chiều - 2



□ Bên nhận:

- Nhận D' có $(d+1)$ bits
- Đếm số bit 1 trong $(d+1)$ bits = x
- Mô hình chẵn: nếu x lẻ \rightarrow error
- Mô hình lẻ: nếu x chẵn \rightarrow error

□ Ví dụ: nhận 0111000110101011

- Parity chẵn: sai
- Parity lẻ: đúng
 - Dữ liệu thật: 011100011010101

□ Đặc điểm:

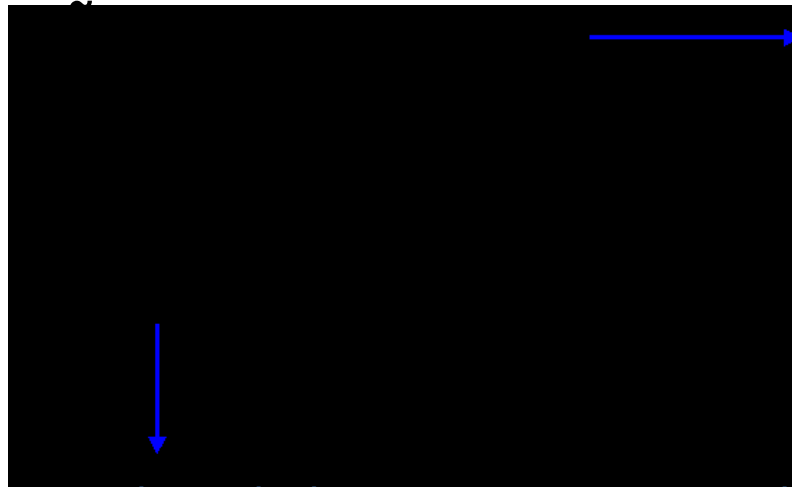
- Phát hiện được lỗi khi số bit lỗi trong dữ liệu là số lẻ
- Không sửa được lỗi



Parity 2 chiều - 1



- ❑ Dữ liệu gửi đi được biểu diễn thành ma trận $N \times M$
- ❑ Số bit parity: $(N + M + 1)$ bit
- ❑ Đặc điểm:
 - Phát hiện và sửa được 1 bit lỗi
- ❑ Bên gửi
 - Biểu diễn dữ liệu thành ma trận $N \times M$
 - Tính toán parity theo hàng, từng cột



Parity 2 chiều - 2



□ Ví dụ:

- Dùng parity chẵn
- $N = 3, M = 5$

- Dữ liệu cần gửi đi: 10101 11110 01110

10101	1
11110	0
01110	1
<hr/>	
00101	0



Parity 2 chiều - 1



□ Bên nhận:

- Biểu diễn dữ liệu nhận thành ma trận $(N+1) \times (M+1)$
- Kiểm tra tính đúng đắn của từng dòng/cột
- Đánh dấu các dòng/cột dữ liệu bị lỗi
- Bit lỗi: bit tại vị trí giao giữa dòng và cột bị lỗi



Parity 2 chiều - 2



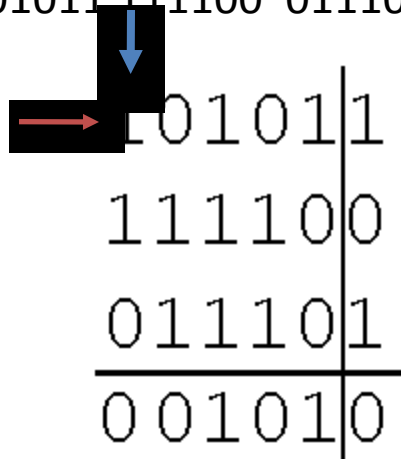
□ Ví dụ:

- Dùng parity chẵn

▪ $N = 3, M = 5$

Dữ liệu nhận:

101011 111100 011101 001010

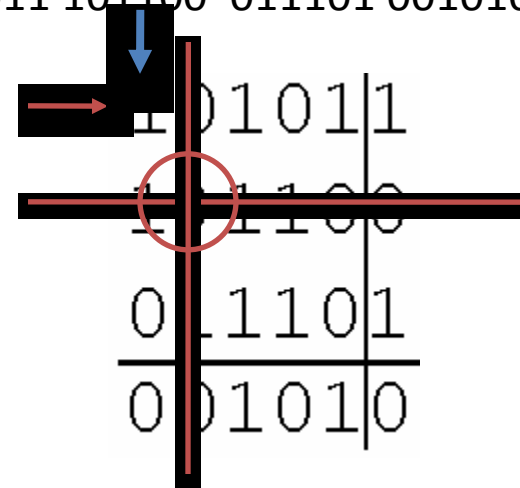


Không có lỗi

Dữ liệu thật: 10101 11110 01110

Dữ liệu nhận:

101011 101100 011101 001010



Có lỗi

Dữ liệu thật: 10101 11110 01110



Hamming code - 1



□ Mỗi hamming code

- có M bit, đánh số từ 1 đến M
- Bit parity: $\log_2 M$ bits, tại các vị trí lũy thừa của 2
- Dữ liệu thật được đặt tại các vị trí không là lũy thừa của 2
- VD: $M = 7$
 - $\log_2 7 = 3$: dùng 3 bits làm bit parity (1, 2, 4)
 - Có 4 vị trí có thể đặt dữ liệu (3, 5, 6, 7)

□ Đặc điểm:

- sửa lỗi 1 bit
- nhận dạng được 2 bit lỗi
- Sửa lỗi nhanh hơn Parity code 2 chiều



Hamming code - 2



□ Bên gởi:

- Chia dữ liệu cần gởi đi thành các khối dữ liệu (với số bit là số vị trí có thể đặt vào Hamming Code)
- Với mỗi khối dữ liệu → tạo 1 Hamming Code
 - Đặt các bit dữ liệu vào các vị trí không phải là lũy thừa của 2 trong Hamming Code
 - lưu ý: vị trí được đánh số từ 1 đến M
 - Tính check bits
 - Tính giá trị của các bit parity

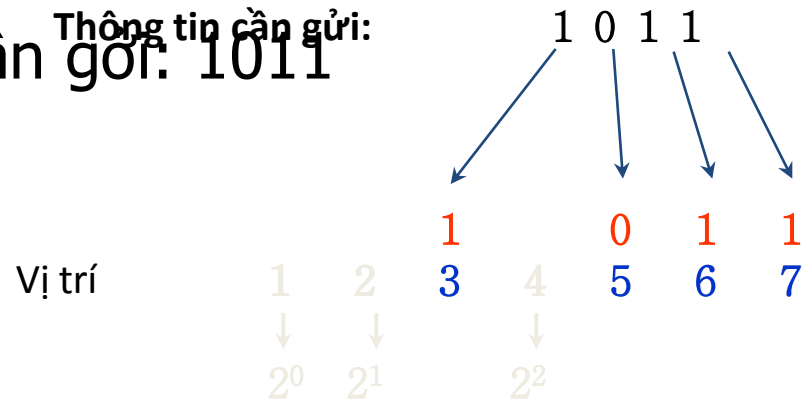


Hamming code – 3



□ Ví dụ:

- $M = 7$
- Dùng parity lẻ
- Thông tin cần gửi: 1011



Tính check bits:

$$\begin{aligned} 3 &= 2^1 + 2^0 = 0 \ 1 \ 1 \\ 5 &= 2^2 + 2^0 = 1 \ 0 \ 1 \\ 6 &= 2^2 + 2^1 + 2^0 = 1 \ 1 \ 0 \\ 7 &= 2^2 + 2^1 + 2^0 = 1 \ 1 \ 1 \end{aligned}$$



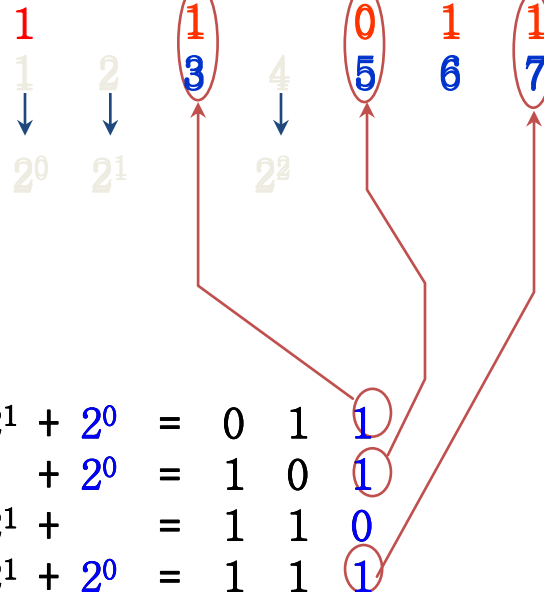
Hamming code - 4



Thông tin cần gửi:

1 0 1 1

Vị trí



Check bits:

$$\begin{aligned} 3 &= 2^1 + 2^0 = 0 \ 1 \ 1 \\ 5 &= 2^2 + 2^0 = 1 \ 0 \ 1 \\ 6 &= 2^2 + 2^1 + 2^0 = 1 \ 1 \ 0 \\ 7 &= 2^2 + 2^1 + 2^0 = 1 \ 1 \ 1 \end{aligned}$$

Vị trí 2^0 :

- Xét cột 2^0 trong check bit → các vị trí có bit 1
- Lấy các bit DL tại các vị trí có bit 1 trong check bit → tính bit parity cho các bit dữ liệu này



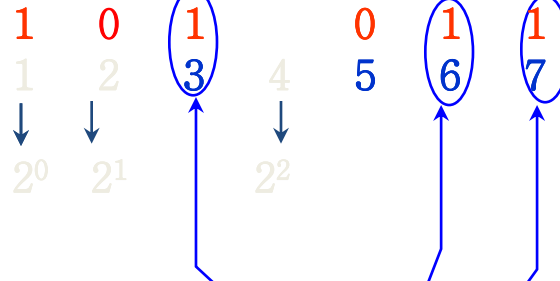
Hamming code - 5



Thông tin cần gửi:

1 0 1 1

Vị trí

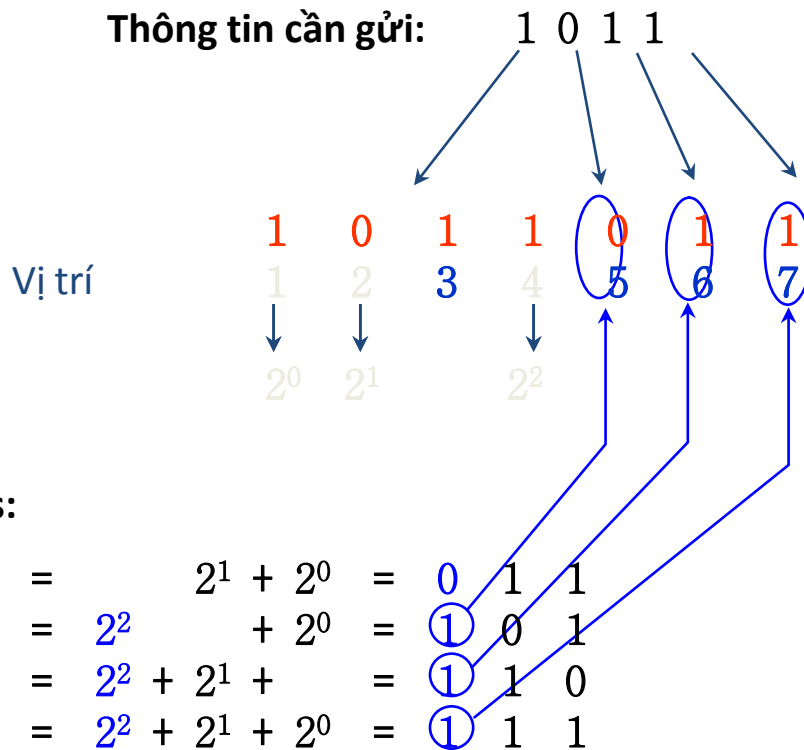


Check bits:

$$\begin{array}{rcl}
 3 & = & 2^1 + 2^0 = 0 \quad \textcircled{1} \quad 1 \\
 5 & = & 2^2 + 2^0 = 1 \quad 0 \quad 1 \\
 6 & = & 2^2 + 2^1 + 2^0 = 1 \quad \textcircled{1} \quad 0 \\
 7 & = & 2^2 + 2^1 + 2^0 = 1 \quad \textcircled{1} \quad 1
 \end{array}$$



Hamming code - 6



Hamming code - 7



- Dữ liệu cần gửi: 1011
- Dữ liệu gửi: 1011011



Hamming code - 8



□ Bên nhận: với mỗi Hamming Code

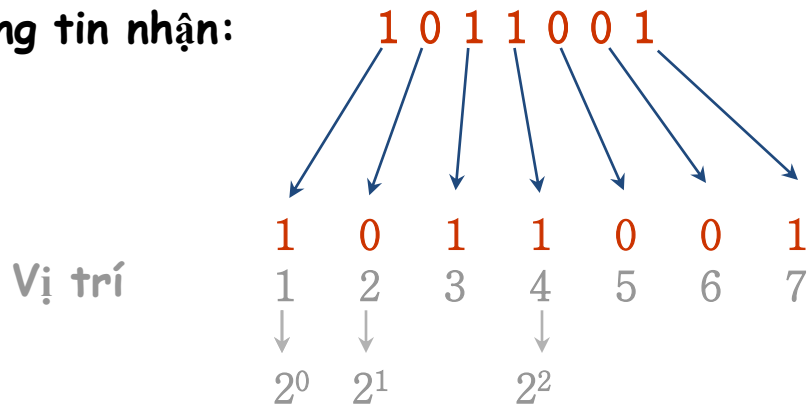
- Điền các bit Hamming Code nhận vào các vị trí từ 1 đến M
- Tính check bit
- Kiểm tra các bit parity
 - Nếu tại bit 2^i phát hiện sai \rightarrow đánh dấu Error, hệ số $k_i = 1$
 - Ngược lại, đánh dấu No Error = 0, hệ số $k_i = 0$
- Vị trí bit lỗi: $pos = \sum 2^i * k_i$



Hamming code – 9



Thông tin nhận:



Tính check bits:

$$\begin{aligned} 3 &= 2^1 + 2^0 = 0 \ 1 \ 1 \\ 5 &= 2^2 + 2^0 = 1 \ 0 \ 1 \\ 6 &= 2^2 + 2^1 = 1 \ 1 \ 0 \\ 7 &= 2^2 + 2^1 + 2^0 = 1 \ 1 \ 1 \end{aligned}$$



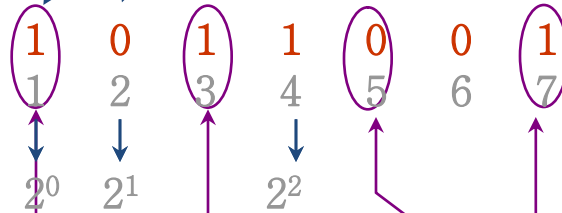
Hamming code – 10



Thông tin nhận:

1 0 1 1 0 0 1

Vị trí



Tính check bits:

$$\begin{array}{rcl}
 3 & = & 2^1 + 2^0 = 0 \ 1 \ 1 \\
 5 & = & 2^2 + 2^0 = 1 \ 0 \ 1 \\
 6 & = & 2^2 + 2^1 = 1 \ 1 \ 0 \\
 7 & = & 2^2 + 2^1 + 2^0 = 1 \ 1 \ 1
 \end{array}$$

Odd parity: Không có lỗi



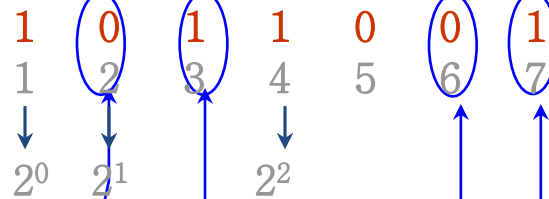
Hamming code – 11



Thông tin nhận:

1 0 1 1 0 0 1

Vị trí



Tính check bits:

$$\begin{aligned}
 3 &= 2^1 + 2^0 = 0 \text{ } \textcircled{1} \text{ } 1 \\
 5 &= 2^2 + 2^0 = 1 \text{ } 0 \text{ } 1 \\
 6 &= 2^2 + 2^1 = 1 \text{ } \textcircled{1} \text{ } 0 \\
 7 &= 2^2 + 2^1 + 2^0 = 1 \text{ } \textcircled{1} \text{ } 1
 \end{aligned}$$

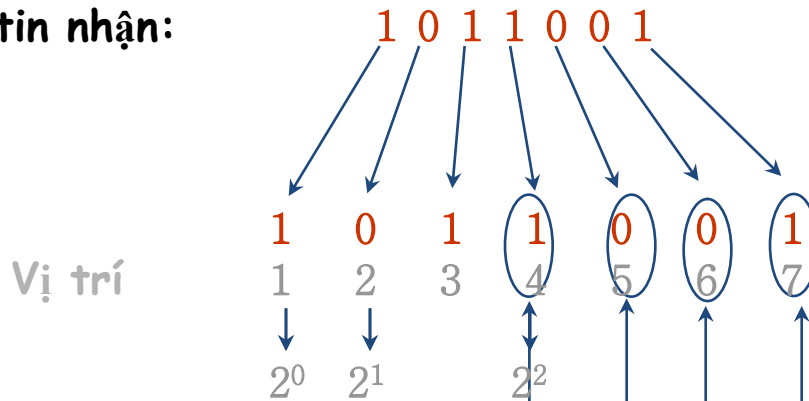
Odd parity: LỖI



Hamming code – 12



Thông tin nhận:



Tính check bits:

$$\begin{array}{rcl}
 3 & = & 2^1 + 2^0 = 0 \ 1 \ 1 \\
 5 & = & 2^2 + 2^0 = 1 \ 0 \ 1 \\
 6 & = & 2^2 + 2^1 = 1 \ 1 \ 0 \\
 7 & = & 2^2 + 2^1 + 2^0 = 1 \ 1 \ 1
 \end{array}$$

Odd parity: LỖI



Hamming code – 13



$$\begin{array}{rcll} & & 2^0 & 2^1 & 2^2 \\ 3 & = & 2^1 + 2^0 & = & 0 & 1 & 1 \\ 5 & = & 2^2 & + & 2^0 & = & 1 & 0 & 1 \\ 6 & = & 2^2 + 2^1 & = & 1 & 1 & 0 \\ 7 & = & 2^2 + 2^1 + 2^0 & = & \underline{1} & \underline{1} & \underline{1} \end{array}$$

E = error in column
NE = no error in column

$$\begin{array}{ccc} \text{E} & \text{E} & \text{NE} \\ \downarrow & \downarrow & \downarrow \\ \mathbf{1} & \mathbf{1} & \mathbf{0} \end{array} = 6$$

→ Lỗi bit thứ 6 trong Hamming Code

Dữ liệu nhận đúng: 10**11011**

Dữ liệu thật: 1011



Check sum - 1

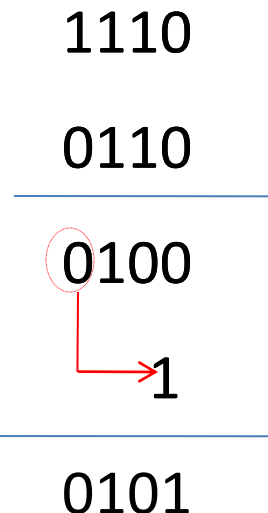


□ Bên gửi

- d bits trong DL gửi đi được xem như gồm N số k bits: x_1, x_2, \dots, x_N
- Tính tổng $X = x_1 + x_2 + \dots + x_N$
- Tính **bù 1** của X \rightarrow giá trị checksum

□ VD: Dữ liệu cần gửi: 1110 0110 0110 0110, k = 4

- 1110, 0110, 0110, 0110
- 0101, 0110, 0110
-
- Sum = 0010
- Checksum = 1101



Check sum - 1



□ Bên nhận:

- tính tổng cho tất cả giá trị nhận được (kể cả giá trị checksum).
- Nếu tất cả các bit là 1, thì dữ liệu nhận được là đúng; ngược lại: có lỗi xảy ra

□ VD:

- nhận: 1110 0110 0110 0110 1101
 - Sum = 1111
 - đúng
- Nhận: 1010 0110 0110 0110 1101
 - Sum = 1011
 - sai



Nội dung



- Giới thiệu
- Kỹ thuật phát hiện và sửa lỗi
- Điều khiển truy cập đường truyền
- ARP
- Ethernet

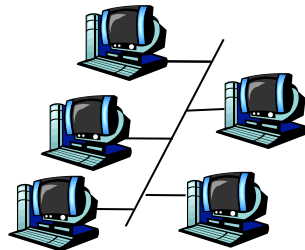


Điều khiển truy cập đường truyền

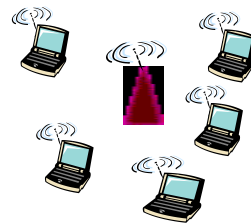


□ Loại liên kết (link)

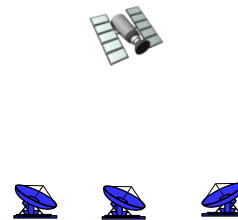
- Điểm đến điểm (Point-to-point)
 - Dialup
 - Nối trực tiếp giữa: host - host, host – SW
- Chia sẻ (Shared)



shared wire (e.g.,
cabled Ethernet)



shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)



Điều khiển truy cập đường truyền

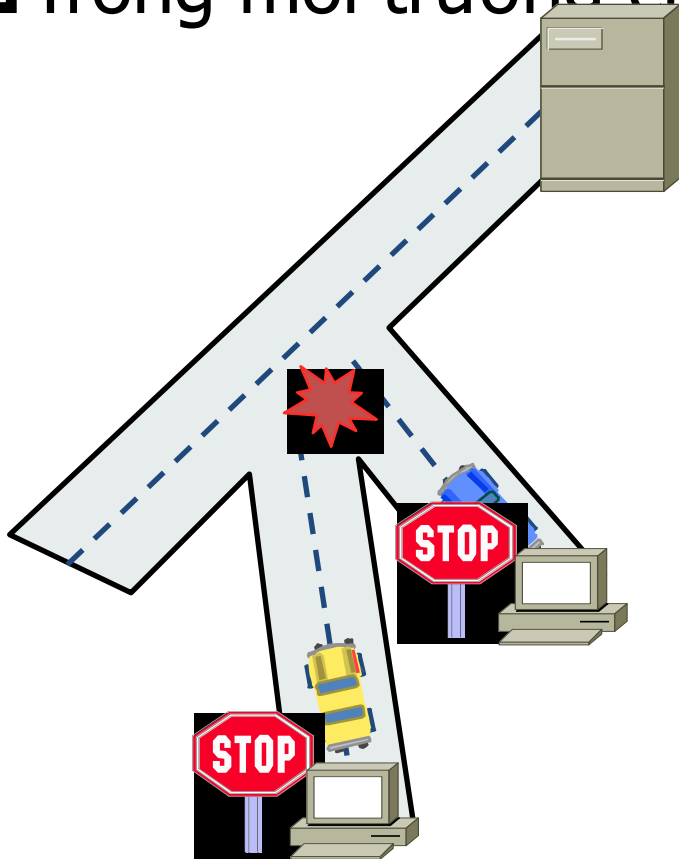


□ Trong môi trường chia sẻ

Hạn chế xảy ra collision

→ Giao thức tầng Data link:
Quyết định cơ chế để các
node sử dụng môi trường
chia sẻ

- khi nào được phép gửi DL xuống đường truyền
- Làm sao phát hiện xảy ra Collision
-



Điều khiển truy cập đường truyền



□ Các phương pháp:

- Phân chia kênh truyền (Channel partition protocols)
- Tranh chấp (Random access protocols)
- Luân phiên (Taking-turns protocols)



Phân chia kênh truyền

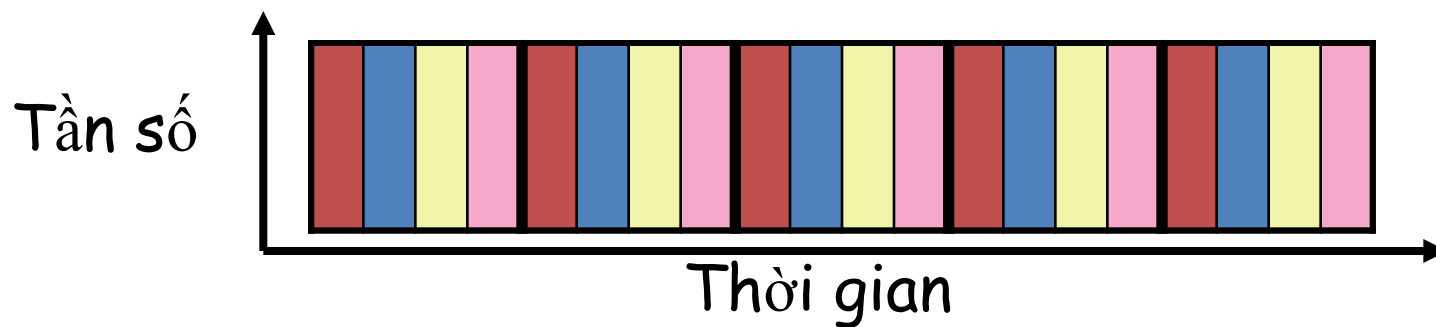


- TDM (Time Division Multiplexing)
- FDM (Frequency Division Multiplexing)
- CDMA (Code Division Multiple Access)



□ Ý tưởng:

- Chia kênh truyền thành các khe thời gian
 - Mỗi khe thời gian chia thành N khe nhỏ
 - Mỗi khe nhỏ dành cho 1 node trong mạng
- Mỗi node có băng thông: R/N



□ Ý tưởng:

- Chia kênh truyền thành N kênh truyền nhỏ
 - Mỗi kênh truyền dành cho 1 node
- Mỗi node có băng thông: R/N



CDMA - 1

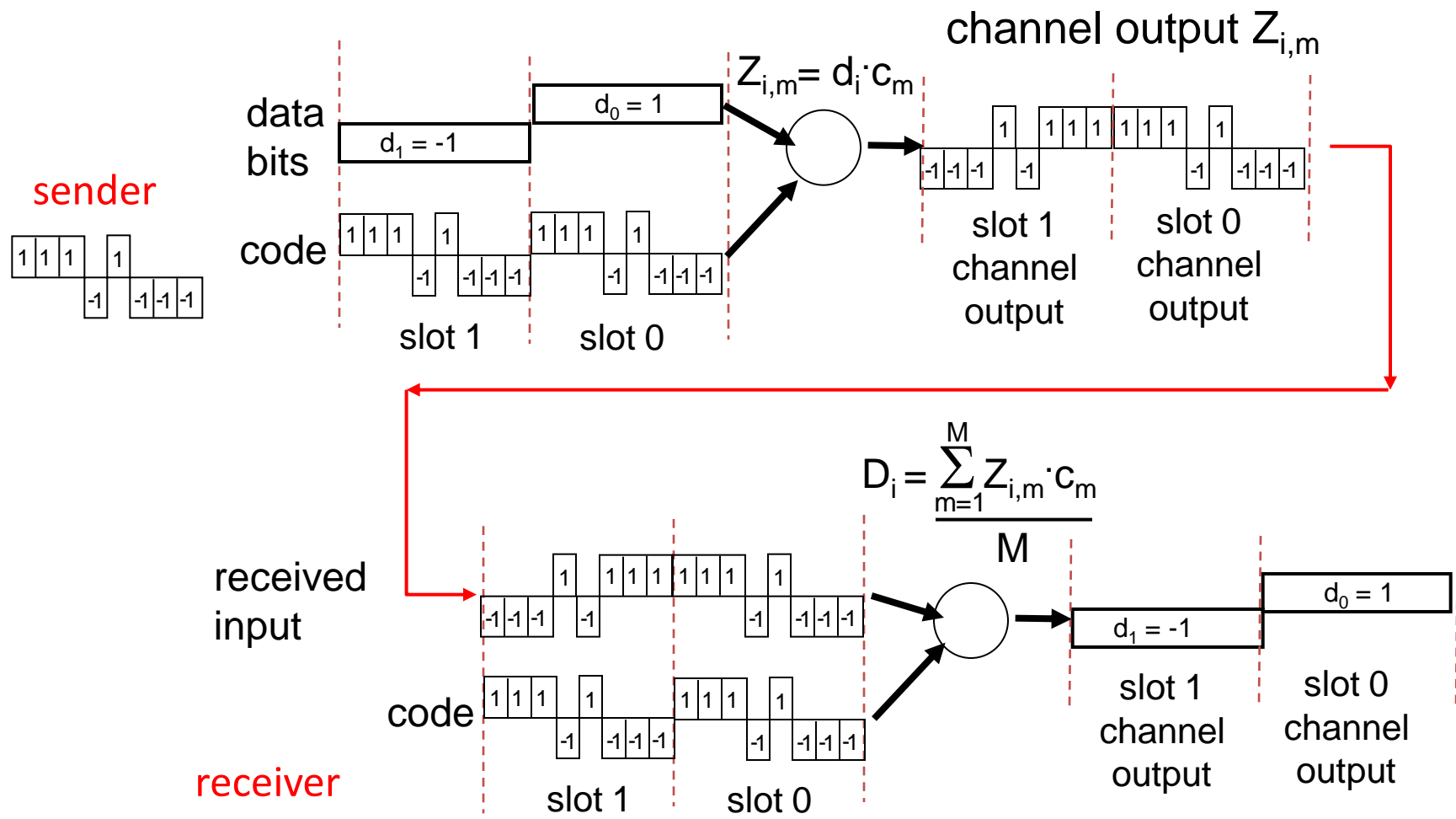


□ Ý tưởng:

- Mỗi node có 1 code riêng
- Bên gửi: mã hoá dữ liệu trước khi gửi bằng code của mình và bên nhận phải biết code của người gửi
- 1 bit DL được mã hoá thành M bits
- Kênh truyền: chia thành từng các khe thời gian, mỗi bit truyền trong 1 khe



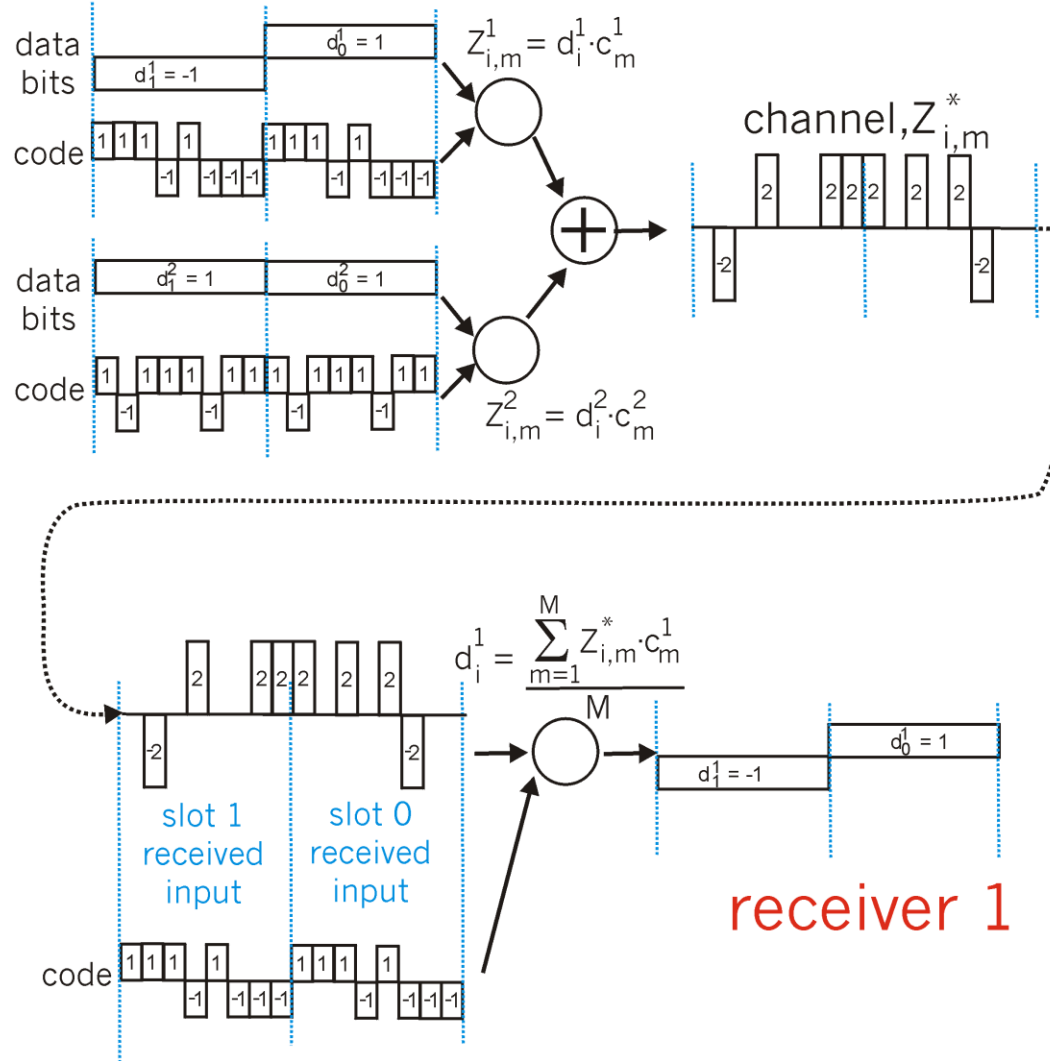
CDMA - 2



CDMA - 3



senders



Tranh chấp



- ❑ Các node chiếm trọn băng thông khi truyền
- ❑ Lắng nghe đặng độ sau khi truyền
- ❑ Một số phương pháp:
 - ALOHA (Slotted, Pure)
 - CSMA (Carrier Sense Multiple Access)



Pure ALOHA



- ❑ Mỗi node có thể bắt đầu truyền dữ liệu bất cứ khi nào node có nhu cầu
- ❑ Nếu phát hiện xung đột → chờ 1 khoảng thời gian rồi truyền lại



Slotted ALOHA



- ❑ Giả thiết:
 - Các frame có kích thước tối đa là L bits
- ❑ Kênh truyền: chia thành các khe thời gian có kích thước L/R (s)
- ❑ Khi 1 node có nhu cầu truyền dữ liệu: phải chờ đến thời điểm bắt đầu của 1 khe mới được truyền
 - cần đồng bộ thời gian giữa các node
- ❑ Nếu đụng độ xảy ra: truyền lại với xác suất là p



CSMA - 1



- ❑ Lắng nghe đường truyền trước khi truyền:
 - Đường truyền rảnh: truyền dữ liệu
 - Đường truyền bận: chờ
- ❑ Lắng nghe đường truyền sau khi truyền
 - Nếu đụng độ xảy ra:
 - dừng truyền
 - đợi 1 khoảng thời gian và truyền lại



❑ Đánh giá:

- Các node có quyền ngang nhau
- Chi phí cao
- Tốc độ: chấp nhận được nếu số lượng node ít
- Không ấn định độ ưu tiên cho thiết bị đặc biệt

❑ Cải tiến:

- CSMA/CD (Carrier Sense Multiple Access / Collision Detection)
- CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance)

□ Ý tưởng:

- Thiết bị lắng nghe đường truyền
- Nếu đường truyền rảnh, thiết bị truyền DL của mình lên đường truyền
- Sau khi truyền, lắng nghe đụng độ?
- Nếu có, thiết bị gửi tín hiệu cảnh báo các thiết bị khác
- Tạm dừng 1 khoảng thời gian ngẫu nhiên rồi gửi DL
- Nếu tiếp tục xảy ra đụng độ, tạm dừng khoảng thời gian gấp đôi.

□ Dùng trong mạng Ethernet



Luân phiên



- Dùng thẻ bài (Token Passing)
- Lò chọn (Polling)



Token Passing



□ Ý tưởng:

- Dùng 1 thẻ bài (token) di chuyển qua các node
- Thiết bị muốn truyền DL thì phải chiếm được thẻ bài

□ Đánh giá:

- Thích hợp cho các mạng có tải nặng
- Thiết lập được độ ưu tiên cho thiết bị đặc biệt
- Chậm hơn CSMA trong mạng có tải nhẹ
- Thiết bị mạng đắt tiền

□ Dùng trong mạng Token Ring



Polling



□ Ý tưởng:

- Có 1 node đóng vai trò điều phối
- Node điều phối kiểm tra nhu cầu gửi DL của các node thứ cấp và xếp vào hàng đợi theo thứ tự và độ ưu tiên
- Thiết bị truyền DL khi đến lượt

□ Đánh giá:

- Có thể thiết lập độ ưu tiên
- Tồn chi phí
- Việc truyền DL của 1 thiết bị tùy thuộc vào thiết bị dò chọn



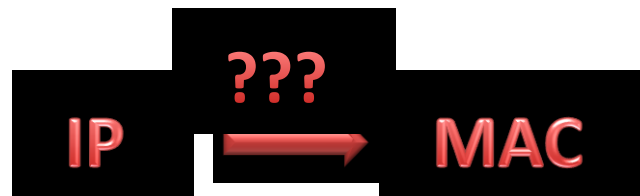
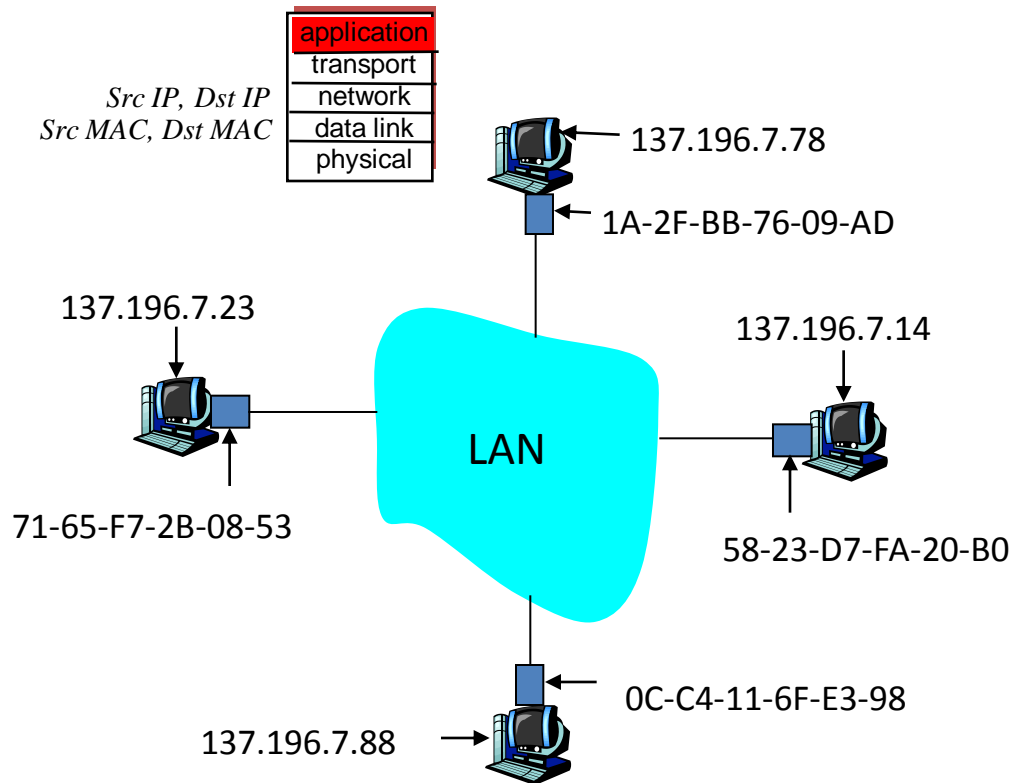
Nội dung



- Giới thiệu
- Kỹ thuật phát hiện và sửa lỗi
- Điều khiển truy cập đường truyền
- ARP
- Ethernet



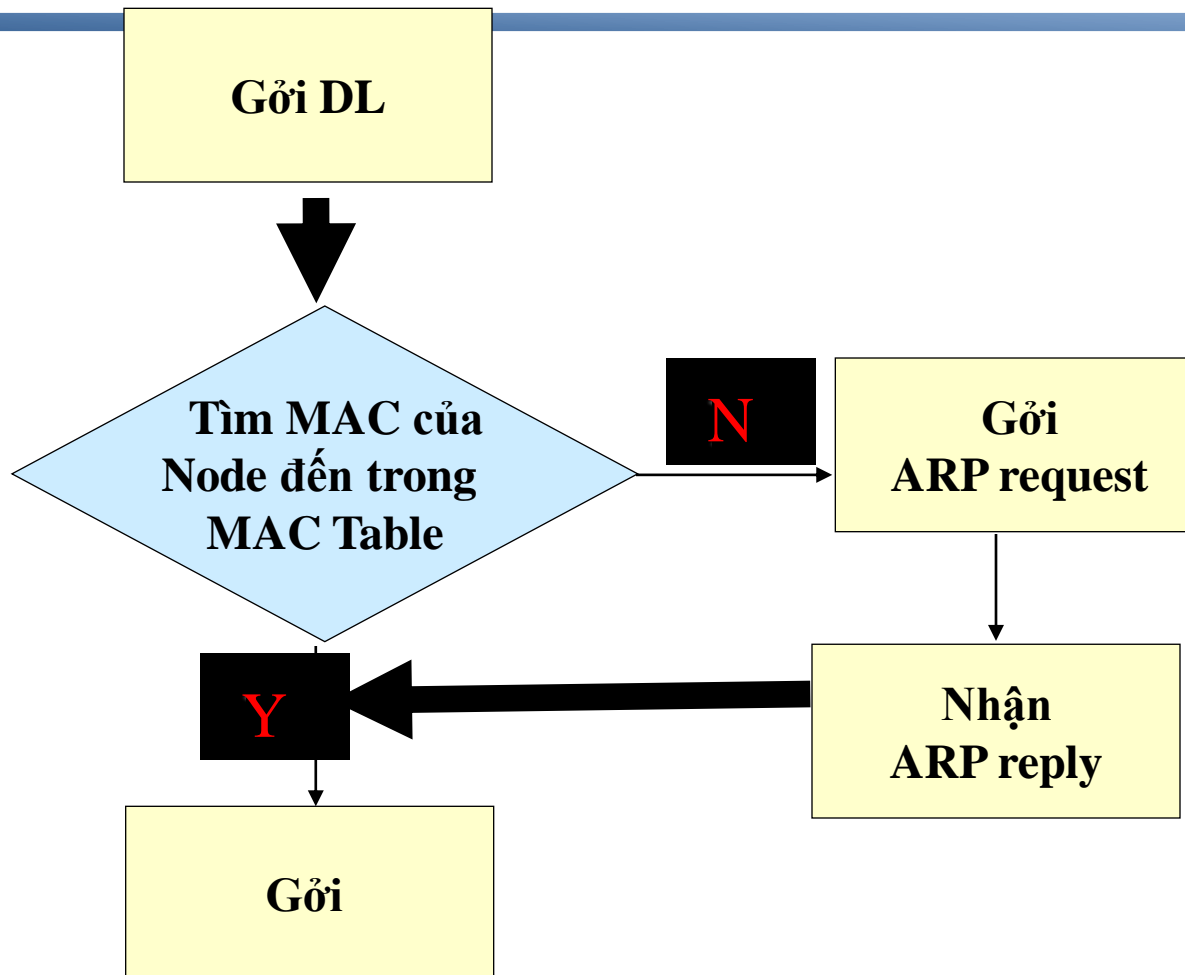
ARP - 1



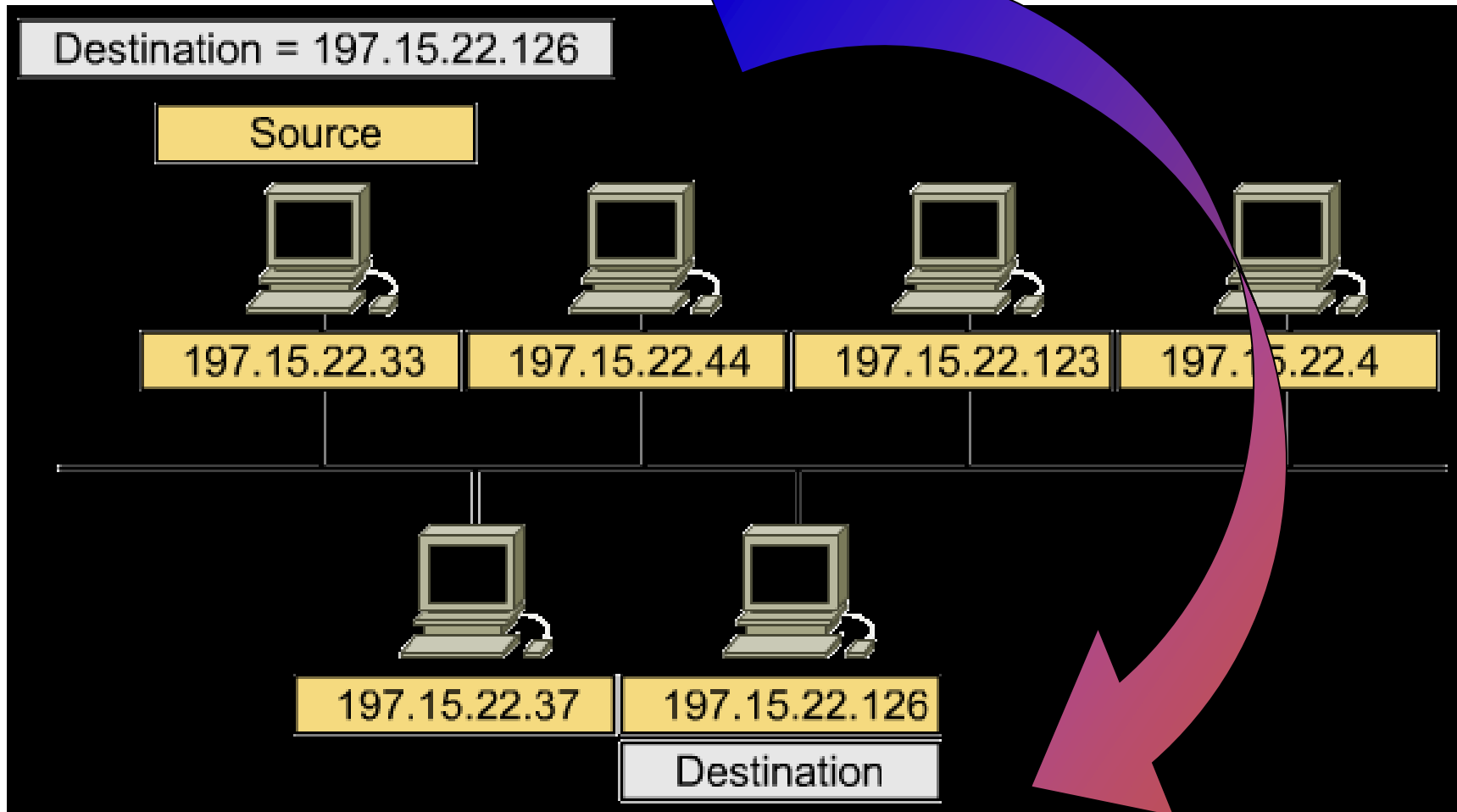
□ ARP (Address Resolution Protocol)

- Phân giải từ địa chỉ IP thành địa chỉ MAC
- Chỉ phân giải trong cùng đường mạng
- Sử dụng ARP table:
 - IP
 - MAC
 - TTL :thời gian sống của record
 - Lưu trong RAM

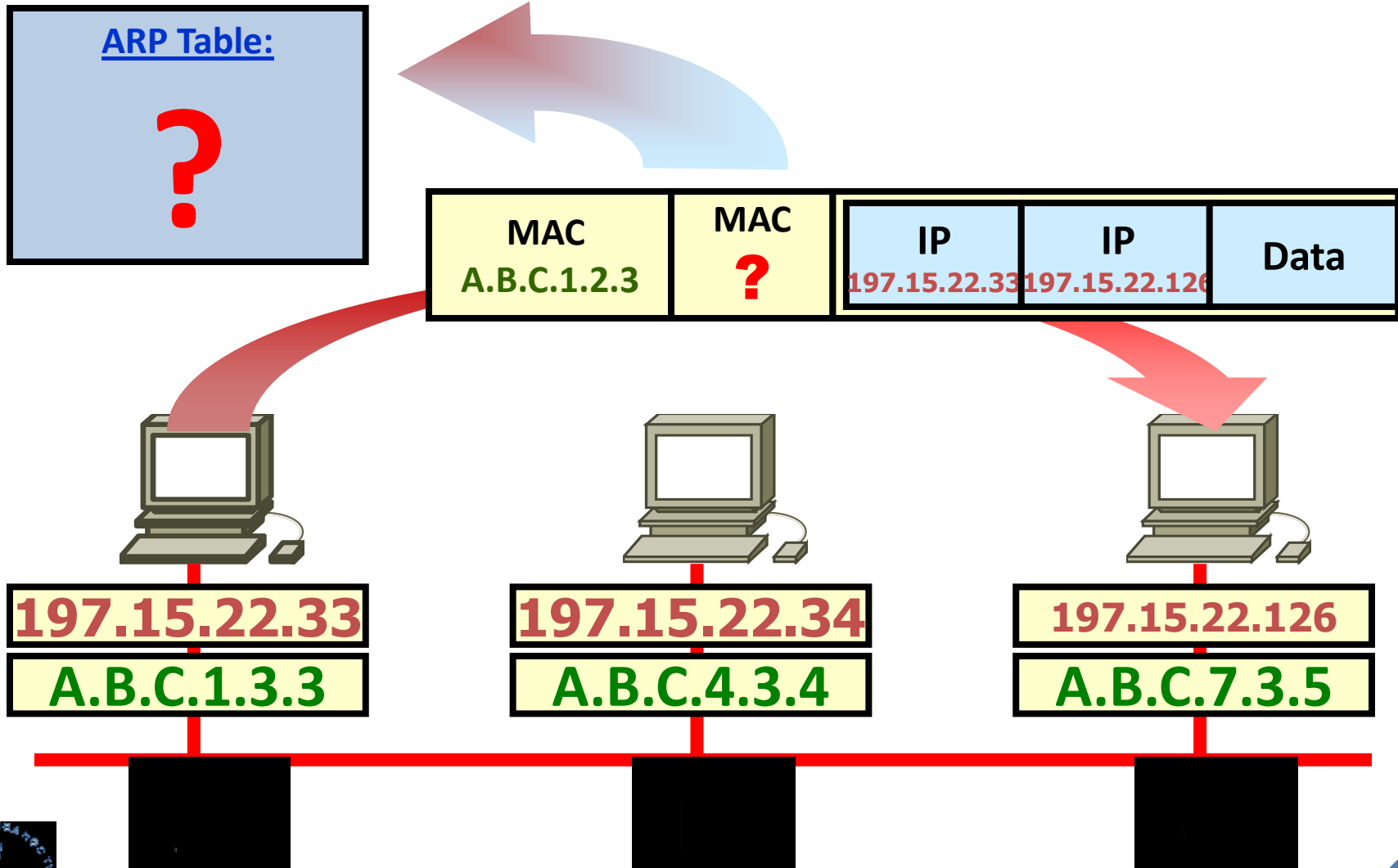
ARP – cơ chế hoạt động



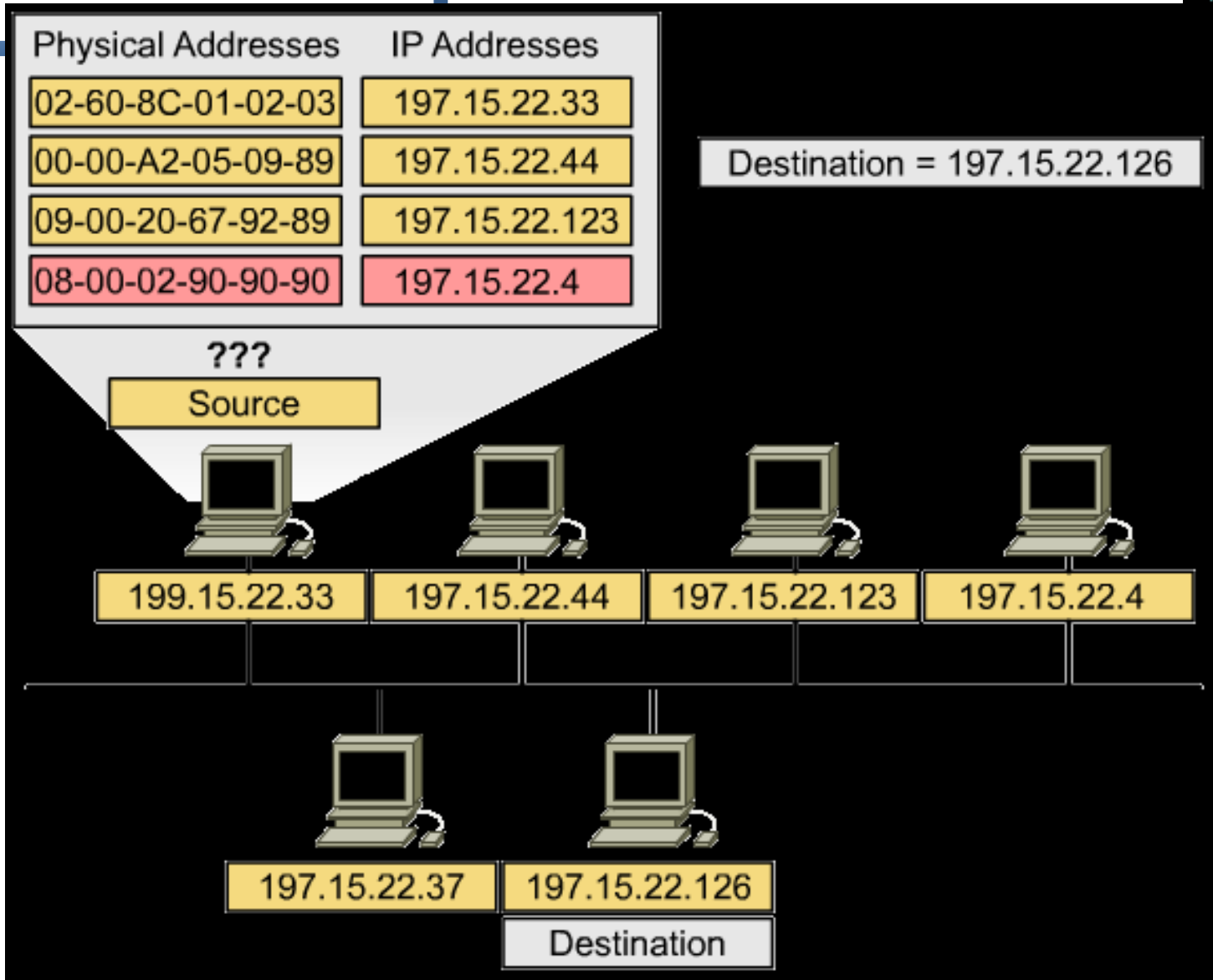
ARP – minh họa - 1



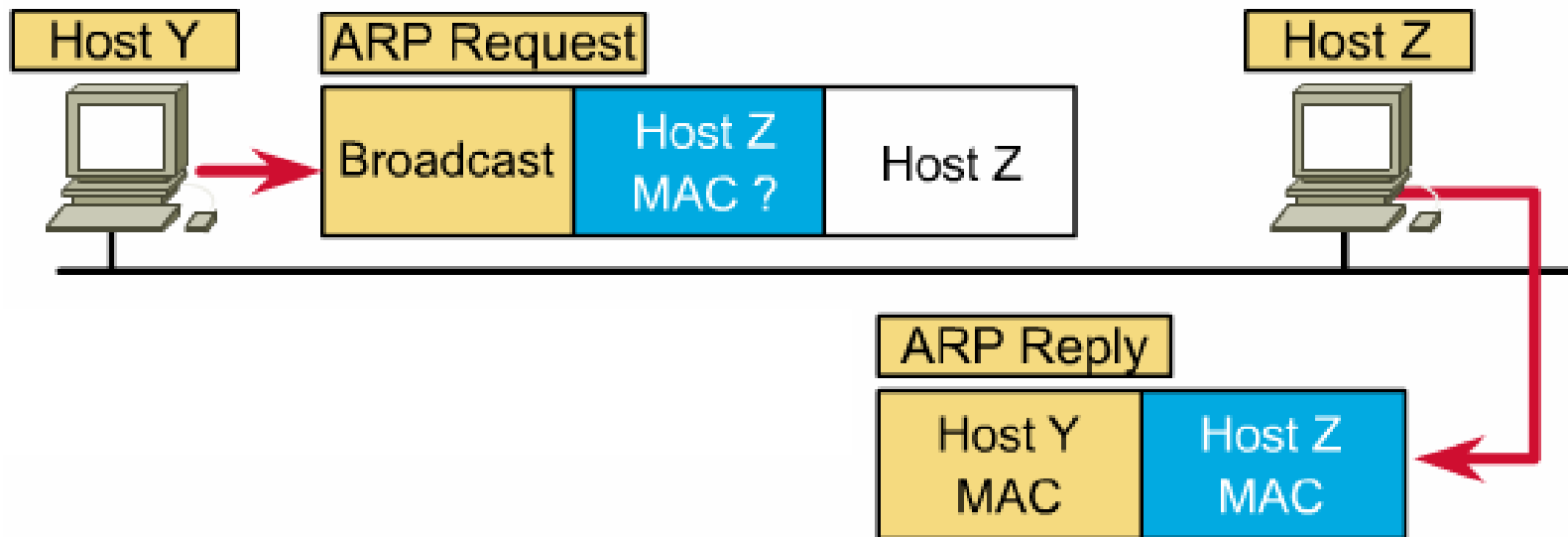
ARP – minh họa - 2



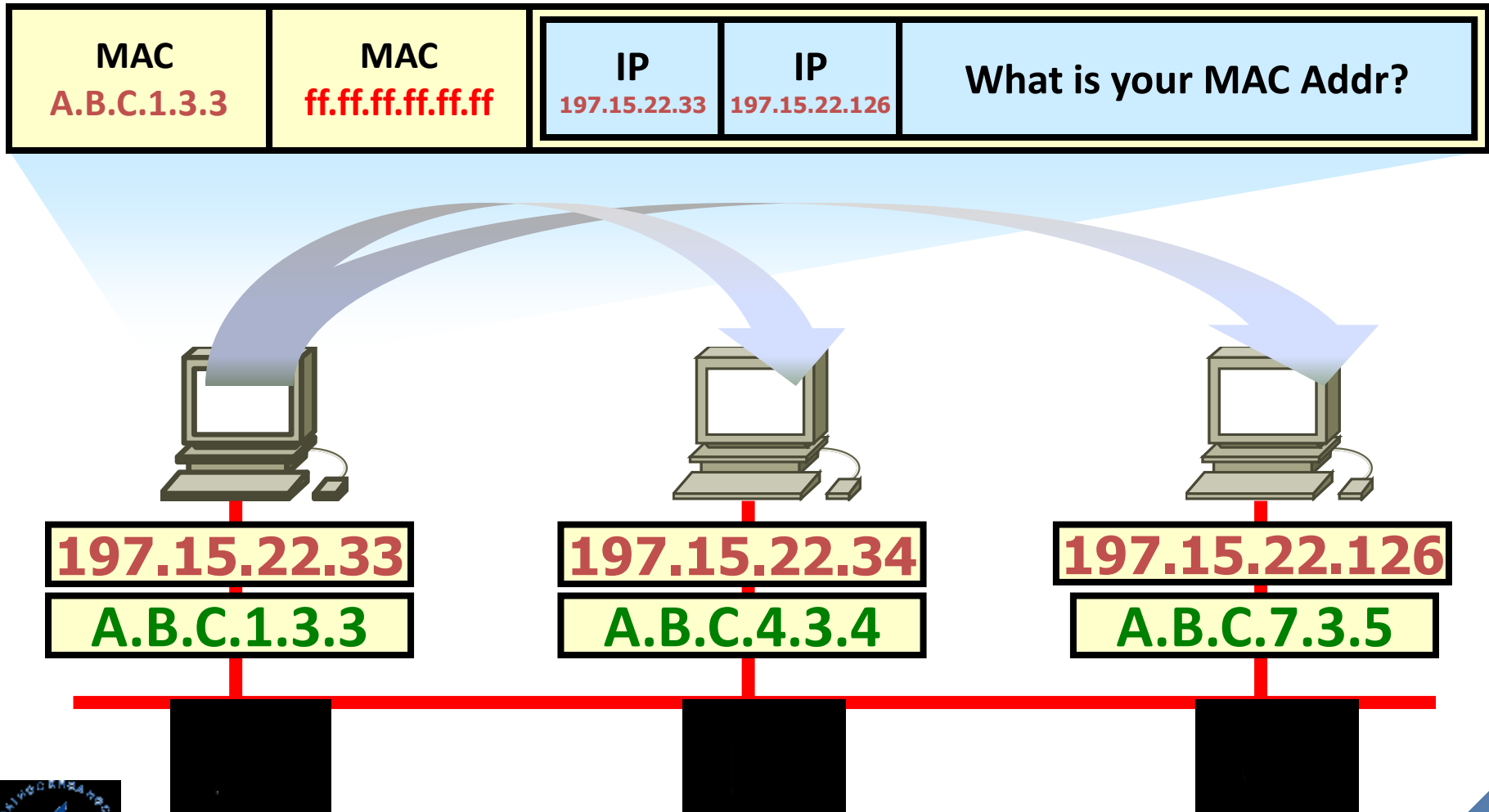
ARP – minh họa - 3



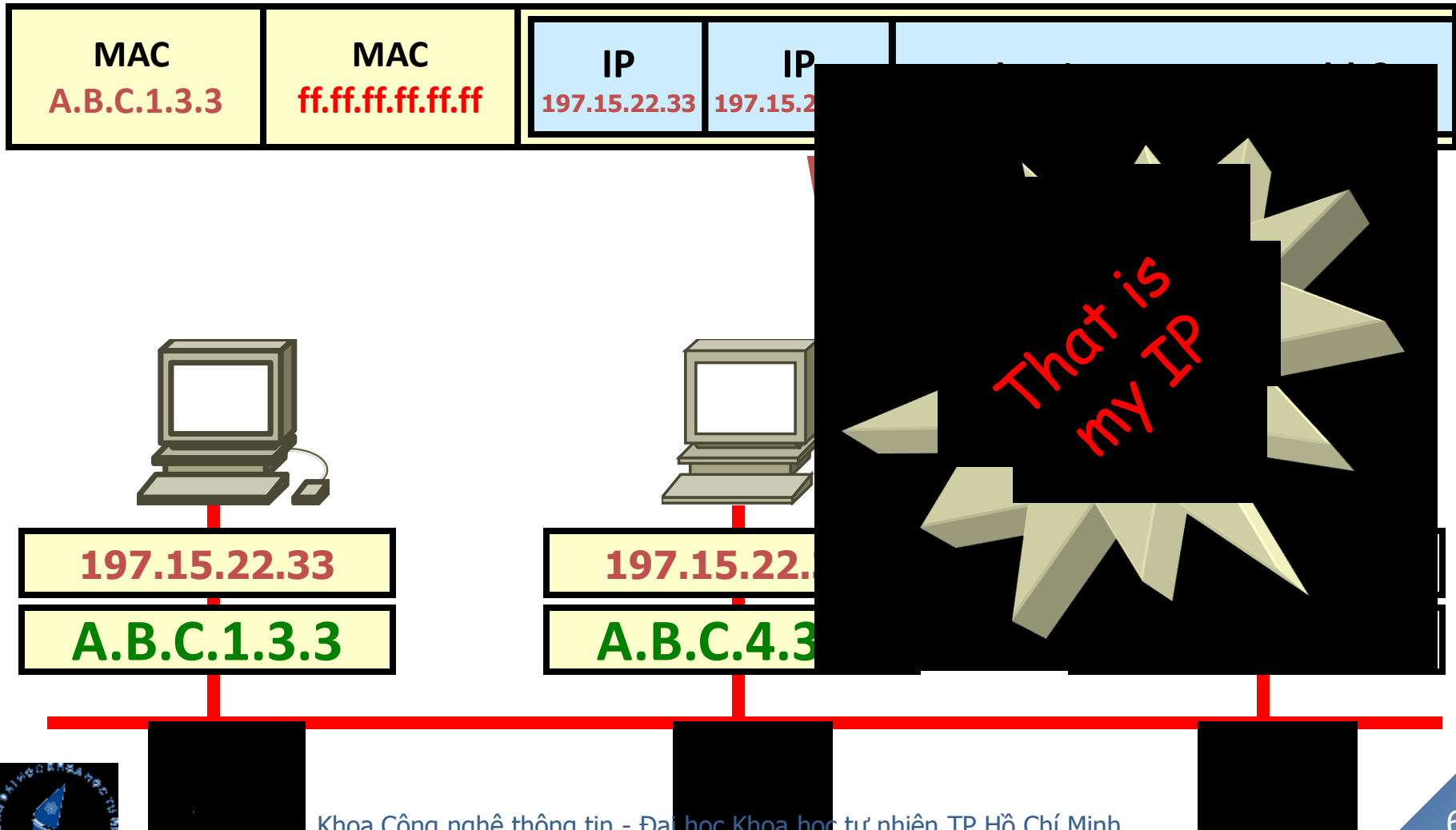
ARP – minh họa - 4



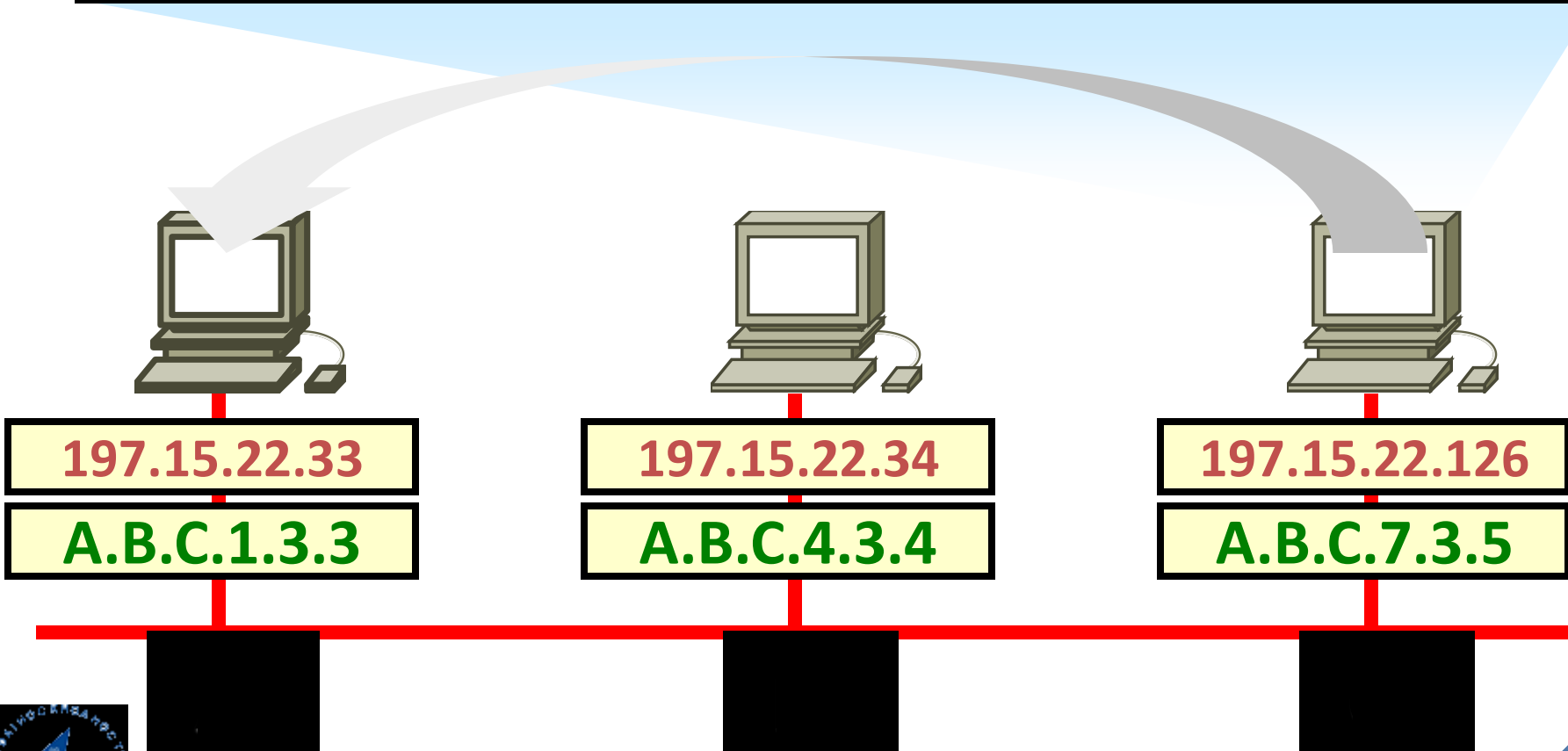
ARP – Request



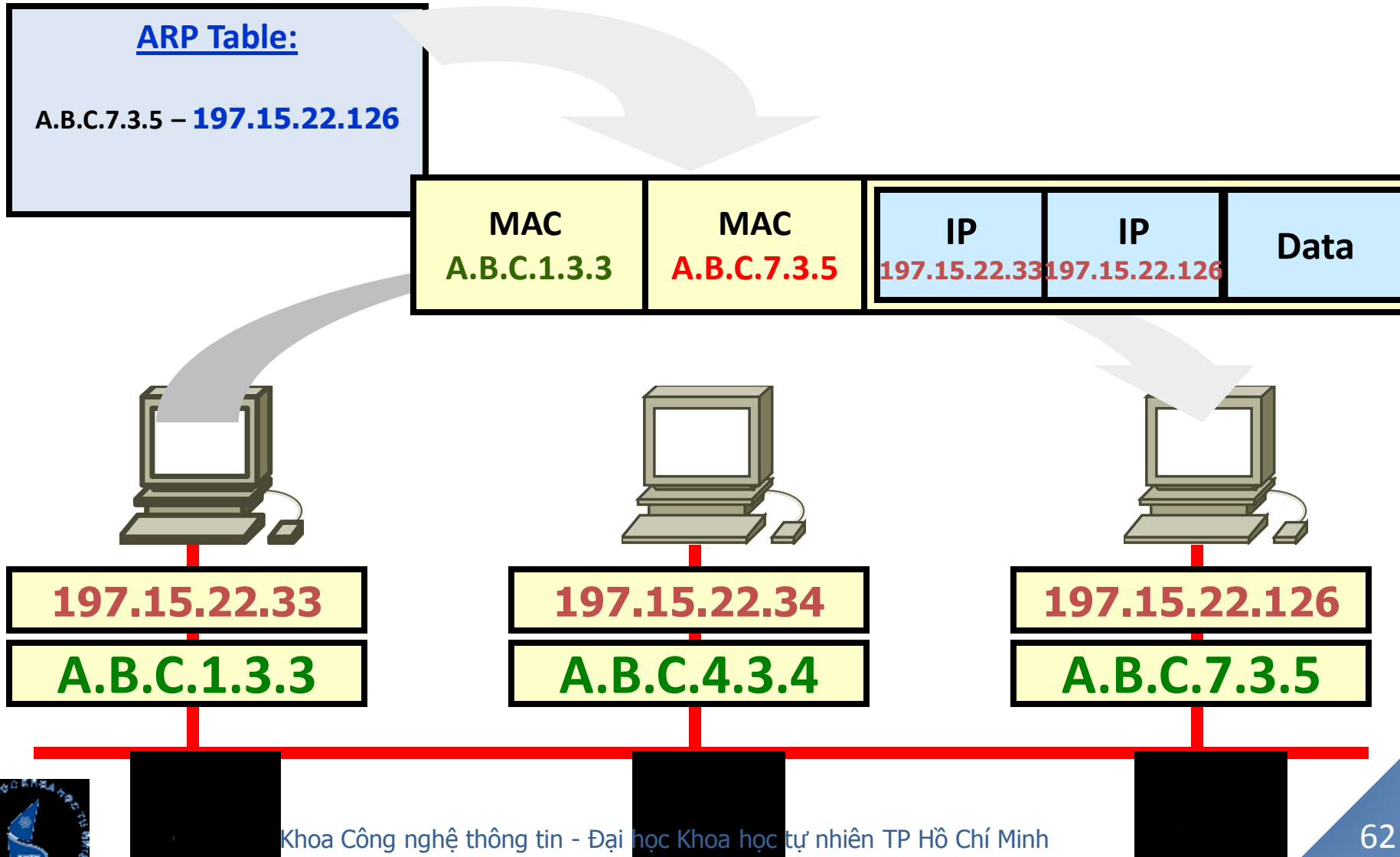
ARP - Checking



ARP - Reply



ARP - Caching



Nội dung



- Giới thiệu
- Kỹ thuật phát hiện và sửa lỗi
- Điều khiển truy cập đường truyền
- ARP
- Ethernet



Ethernet - 1



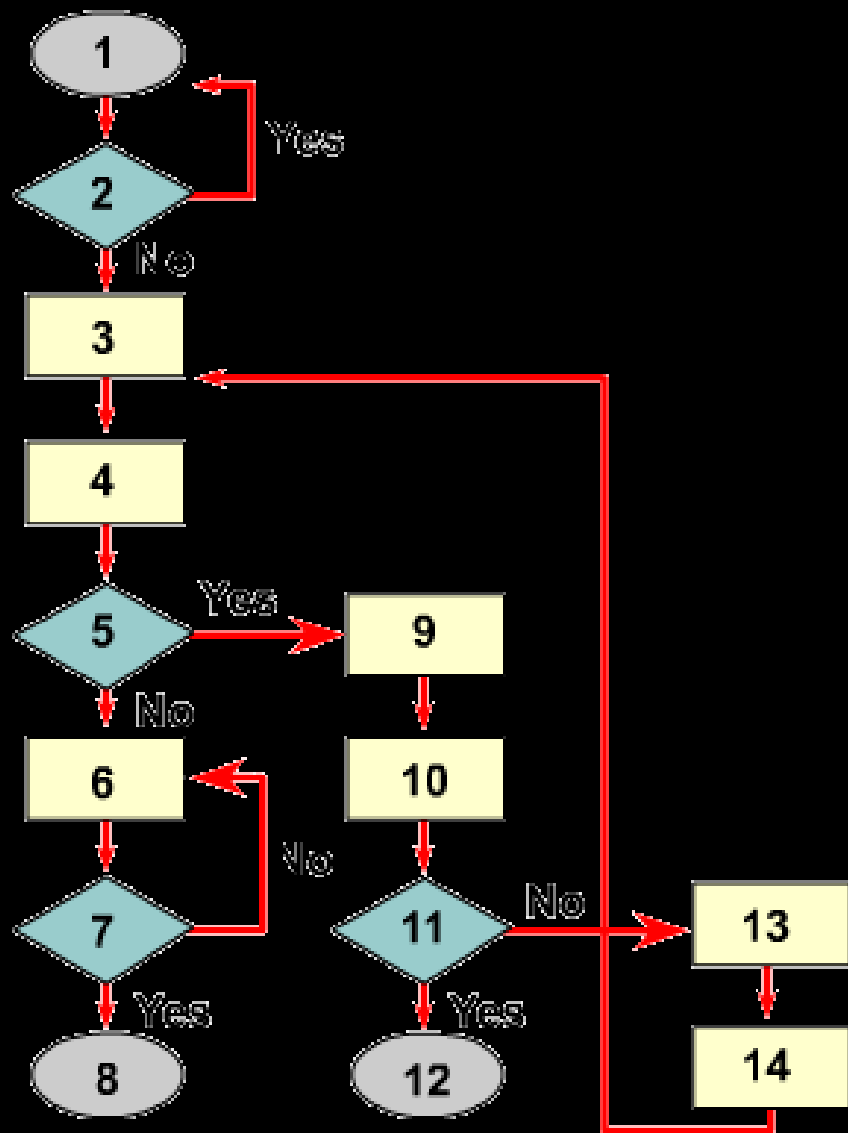
- Là 1 kỹ thuật (technology) mạng LAN có dây
 - Là 1 kỹ thuật mạng LAN đầu tiên
 - Chuẩn 802.3
 - Hoạt động tầng Data Link và Physical
 - Tốc độ: 10 Mbps – 10 Gbps
 - Đồ hình mạng:
 - Bus
 - Star
 - Giao thức tầng MAC: CSMA/CD
 - Đơn giản và rẻ hơn mạng Token Ring LAN, ATM



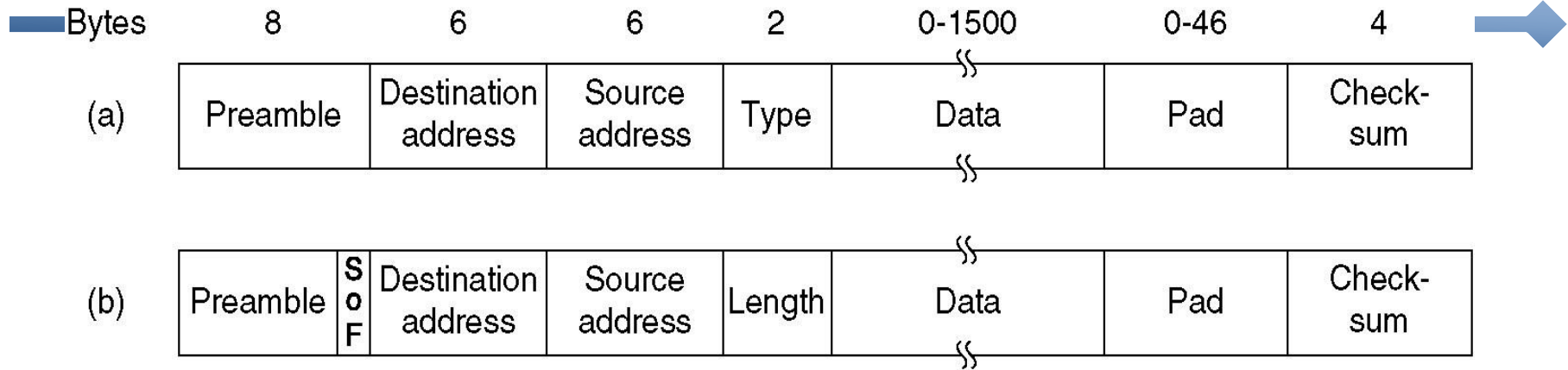
CSMA/CD – quá trình truyền dữ liệu



1. Host wants to transmit
2. Is carrier sensed?
3. Assemble frame
4. Start transmitting
5. Is a collision detected?
6. Keep transmitting
7. Is the transmission done?
8. Transmission completed
9. Broadcast jam signal
10. Attempts = Attempts + 1
11. Attempts > Too many?
12. Too many collisions; abort transmission
13. Algorithm calculates backoff
14. Wait for t microseconds



Ethernet – cấu trúc frame



a) earlier Ethernet frames - b) 802.3 frames

- ❑ Preamble (8 bytes)
 - Đồng bộ đồng hồ bên gửi và bên nhận (10101010)
 - Start of Frame (SOF): báo hiệu bắt đầu frame (101010**11**)
- ❑ Dest. Addr (6 bytes)
 - địa chỉ MAC của card mạng nhận gói tin tiếp theo
- ❑ Src. Addr (6 bytes)
 - địa chỉ MAC của card mạng gửi gói tin
- ❑ Type (2 bytes)
 - Giao thức sử dụng ở tầng trên
- ❑ CRC: dùng để kiểm tra lỗi



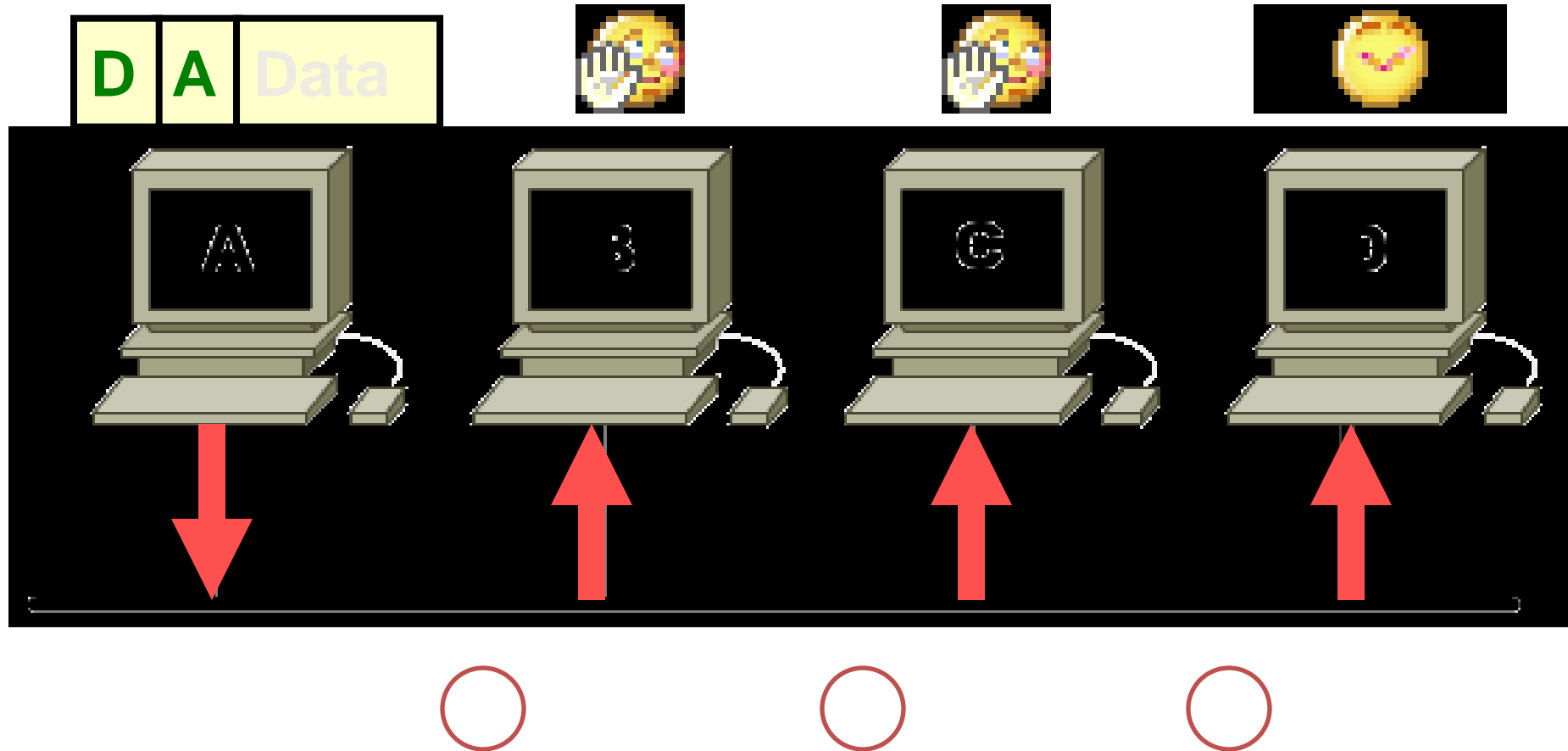
Ethernet – trường type



EtherType	Protocol
0x0800	Internet Protocol, Version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)
0x8035	Reverse Address Resolution Protocol (RARP)
0x809b	AppleTalk (Ethertalk)
0x80f3	AppleTalk Address Resolution Protocol (AARP)
0x8100	IEEE 802.1Q-tagged frame
0x8137	Novell IPX (alt)
0x8138	Novell
0x86DD	Internet Protocol, Version 6 (IPv6)
0x8847	MPLS unicast
0x8848	MPLS multicast



Ethernet – minh hoạ



Ethernet – các công nghệ mạng



- 10Base2
- 10Base5
- 10BaseT
- 100BaseTX
- 100BaseFX
- Gigabit Ethernet

10 Base T

Tốc độ mạng

Loại cáp

Kiểu truyền dữ liệu



Ethernet – chuẩn 10Mbps



Standard	Topology	Medium	Maximum cable length	Transport
10BASE5	Bus	Thick coaxial cable	500m	Half-duplex
10BASE2	Bus	Thin coaxial cable	185m	Half-duplex
10BASE-T	Star	CAT3 UTP	100m	Half or Full-duplex



Ethernet – chuẩn 100Mbps



Standard	Medium	Maximum cable length
100BASE-TX	CAT5 UTP	100m
100BASE-FX	Multi-mode fibre (MMF) 62.5/125	412m



Ethernet – chuẩn gigabit



Standard	Medium	Maximum cable length
1000BASE-SX	Fiber optics	550 m
1000BASE-LX	Fiber optics	5000 m
1000BASE-CX	STP	25 m
1000BASE-T	Cat 5 UTP	100 m



Tài liệu tham khảo



- ❑ Slide của J.F Kurose and K.W. Ross về Computer Networking: A Top Down Approach
- ❑ Slide CCNA, version 3.0, Cisco





Chương 04

Phương tiện truyền dẫn

MẠNG MÁY TÍNH

Tháng 09/2011

Nội dung



1. Giới thiệu
2. PTTD hữu tuyến
3. PTTD vô tuyến



Giới thiệu - 1



- ❑ Phương tiện truyền dẫn: là môi trường dùng để truyền tín hiệu từ nơi này đến nơi khác
- ❑ Phân loại:
 - Hữu tuyến: cáp đồng trục, cáp xoắn, cáp quang
 - Vô tuyến: sóng vô tuyến (wireless)



Giới thiệu - 2



□ Các vấn đề liên quan:

- Chi phí
- Tốc độ
- Suy giảm (suy dần) tín hiệu
- Nhiễu
- An toàn



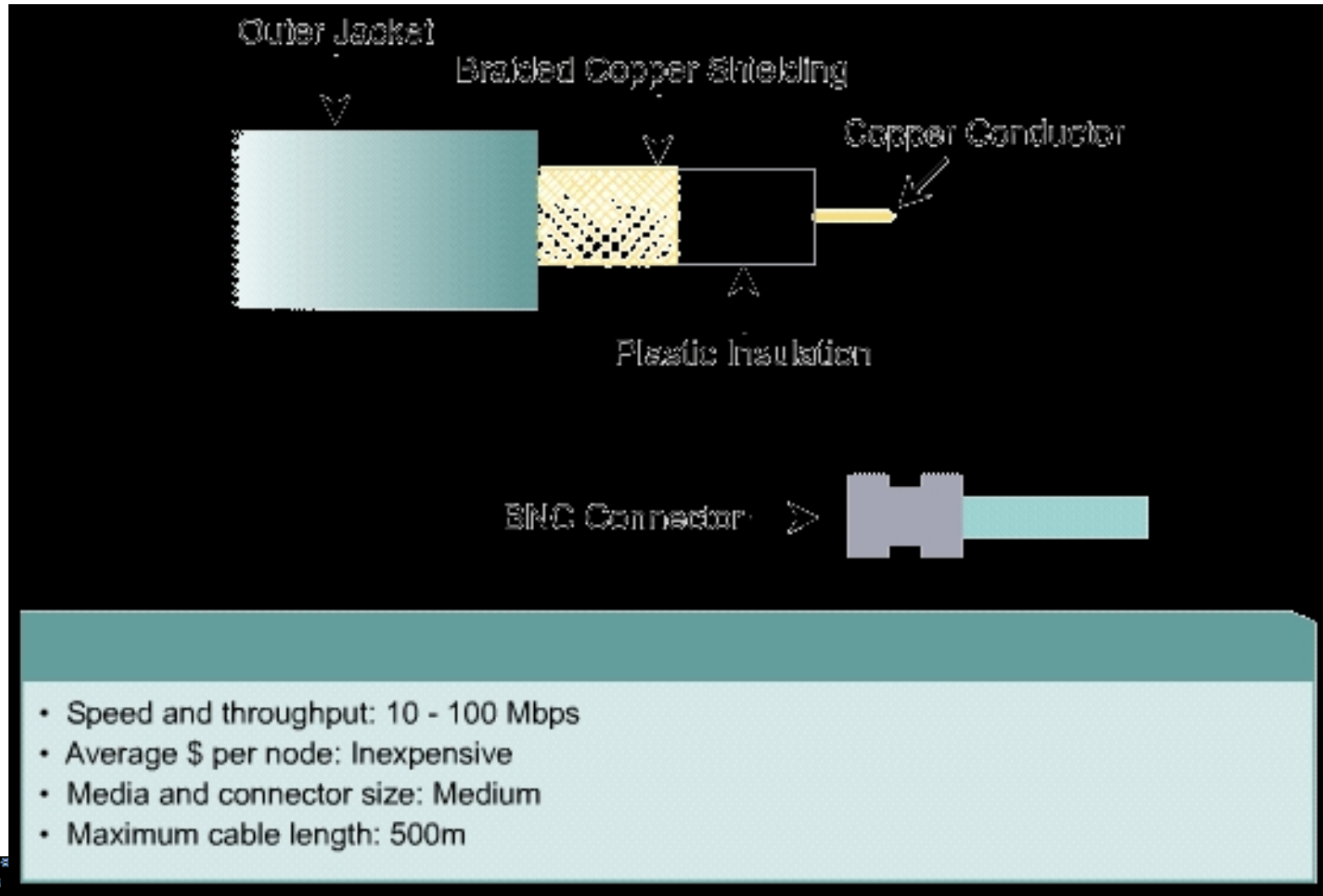
Nội dung



1. Đặc tính của một loại PTTD
2. PTTD hữu tuyến
3. PTTD vô tuyến



Cáp đồng trục (Coax cable) - 1



- Speed and throughput: 10 - 100 Mbps
- Average \$ per node: Inexpensive
- Media and connector size: Medium
- Maximum cable length: 500m

Cáp đồng trục – 2



□ Cấu tạo:

- Hai dây dẫn quấn quanh một trục chung
- Dây dẫn trung tâm: dây đồng hoặc dây đồng bện
- Dây dẫn ngoài: dây đồng bện hoặc lá → bảo vệ dây dẫn trung tâm khỏi nhiễu điện từ và được nối đất để thoát nhiễu.
- Giữa 2 dây dẫn là một lớp vỏ cách điện
- Ngoài cùng là lớp vỏ plastic dùng để bảo vệ cáp



Cáp đồng trục - 3



□ Phân loại:

- Cáp mỏng (thin cable/ ThinNet – 10BASE2)
 - đường kính: 6mm
 - chiều dài cáp tối đa: 185m
- Cáp dày (thick cable/ ThickNet – 10BASE5)
 - đường kính: 13mm
 - chiều dài cáp tối đa: 500m



Cáp thinnet – cách kết nối - 1



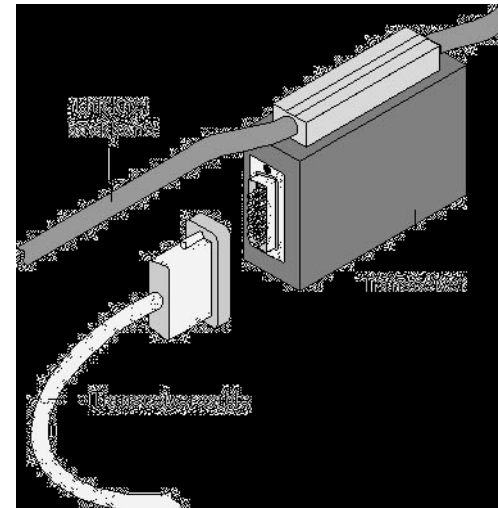
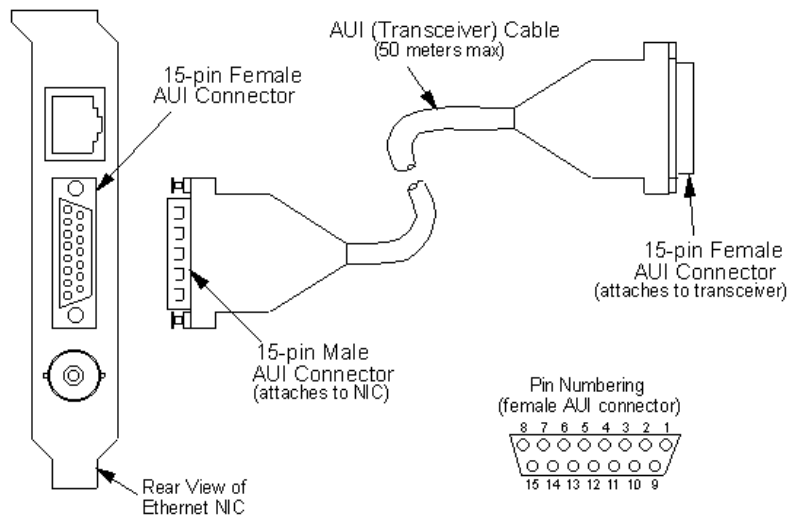
Cáp thinnet – cách kết nối - 2



Cáp thicknet – cách kết nối - 1



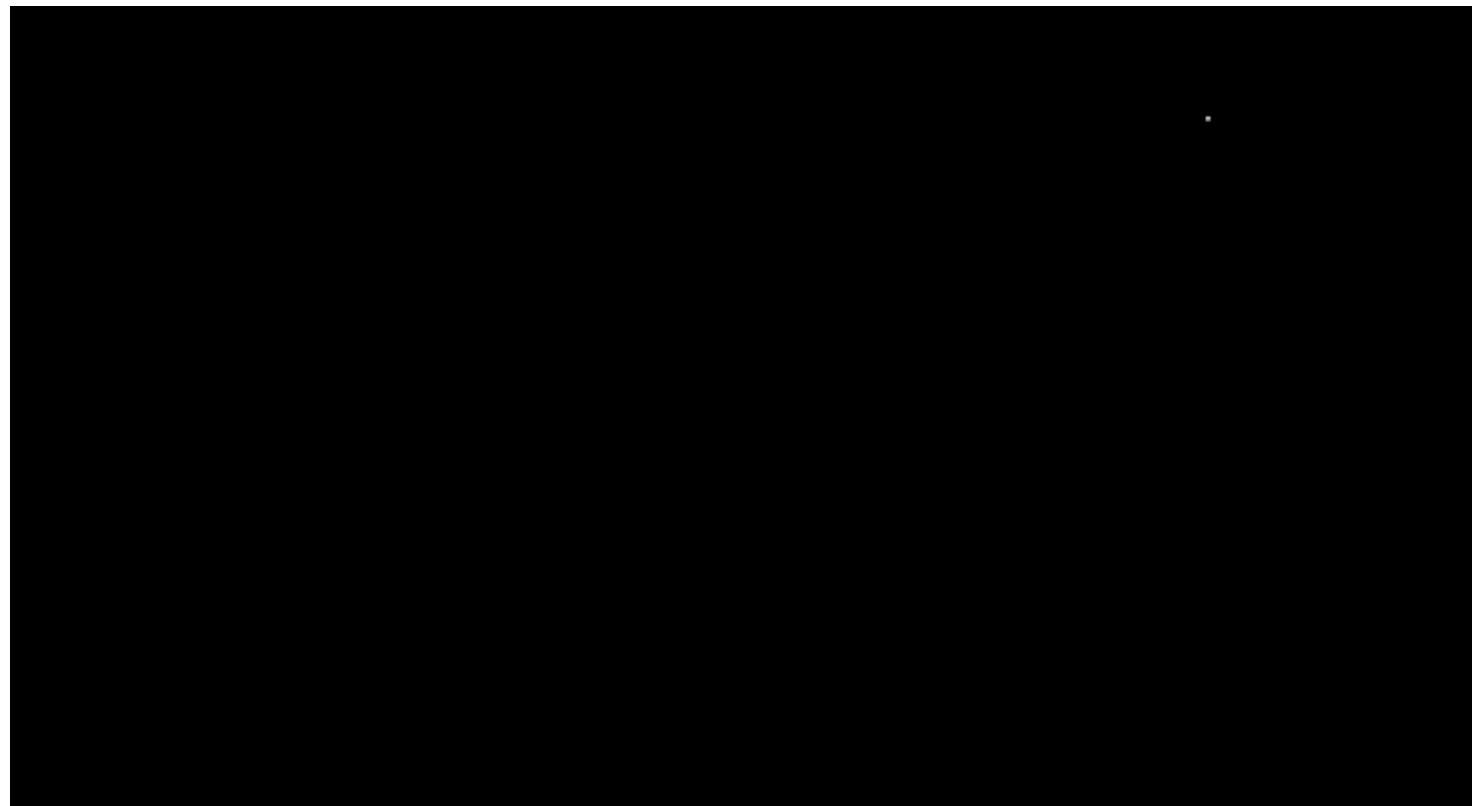
AUI Cable and Connectors



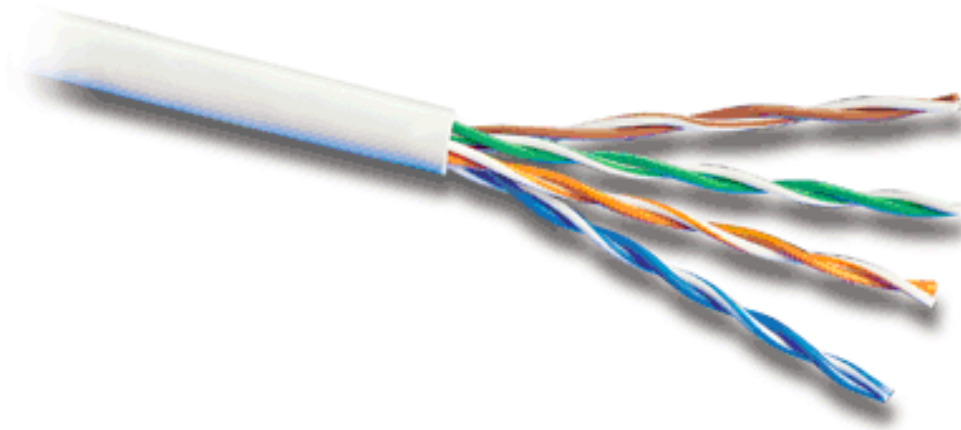
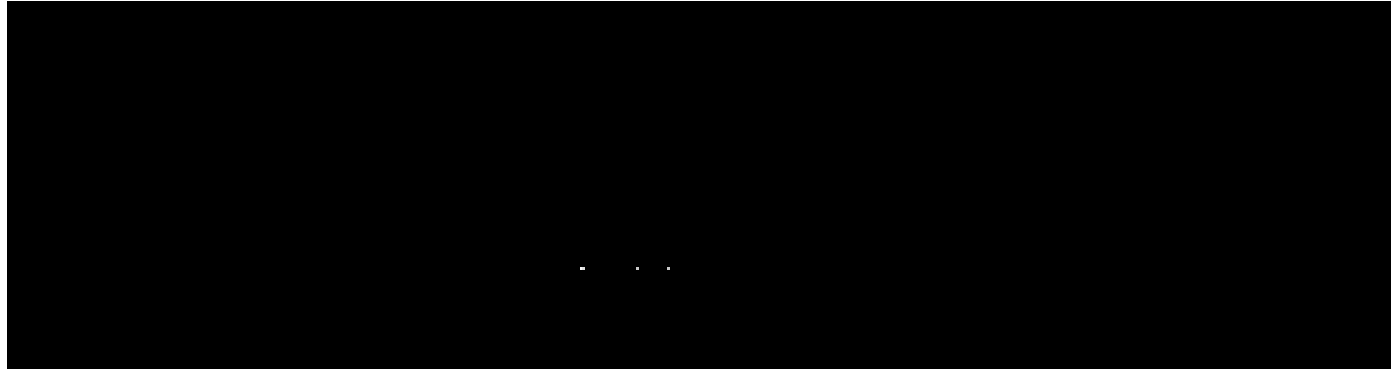
Copyright 1999 TechFest.com All rights reserved.



Cáp thicknet – cách kết nối - 2



Cáp xoắn (Twisted pair) - 1



Cáp xoắn - 2



□ Cấu tạo:

- Hai dây dẫn được xoắn lại thành một cặp
 - chống nhiễu từ bên ngoài và nhiễu từ dây dẫn kế cận (crosstalk)
- Mức độ xoắn (trên 1m dây) càng cao thì khả năng chống nhiễu crosstalk càng cao

□ Phân loại:

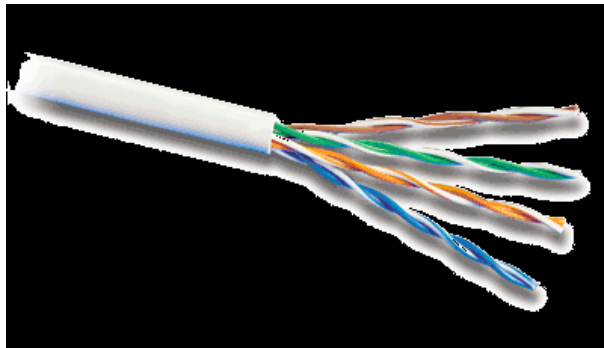
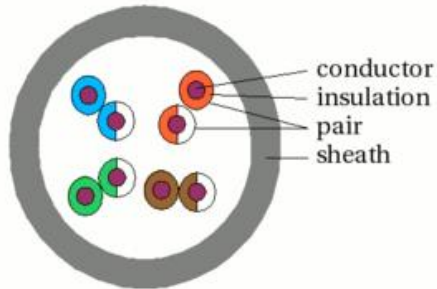
- STP (**Shielded Twisted Pair**)
- S/STP (**Screened Shielded Twisted Pair**)
- UTP (**Unshielded Twisted Pair**)
- S/UTP - FTP (**Screened Unshielded Twisted Pair**)



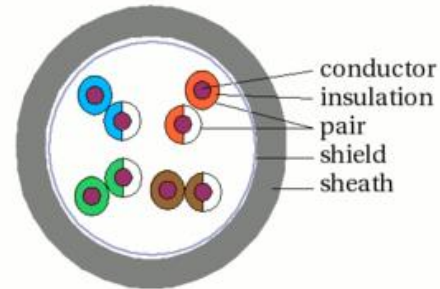
UTP – S/UTP - 1



UTP



S/UTP - FTP - S/FTP



- Speed and throughput: 10 - 100 - 1000 Mbps (depending on the quality/category of cable)
- Average \$ per node: Least Expensive
- Media and connector size: Small
- Maximum cable length: 100m



UTP – S/UTP - 2



- Chi phí: rẻ nhất
- Độ suy dần: lớn
- chiều dài tối đa : 100m
- EMI: dễ bị nhiễu
- Đầu nối: RJ-45



UTP – 3



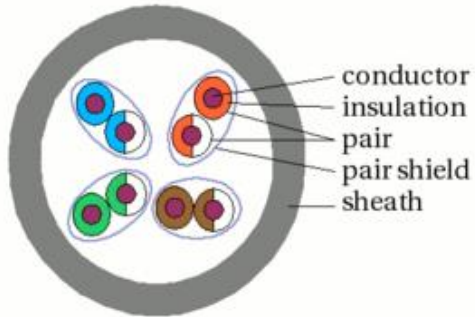
Type	Use
Category 1	Voice Only (Telephone Wire)
Category 2	Data to 4 Mbps (LocalTalk)
Category 3	Data to 10 Mbps (Ethernet)
Category 4	Data to 20 Mbps (16 Mbps Token Ring)
Category 5	Data to 100 Mbps (Fast Ethernet)
Category 5e, 6	Data to 1Gbps (Gigabit Ethernet)



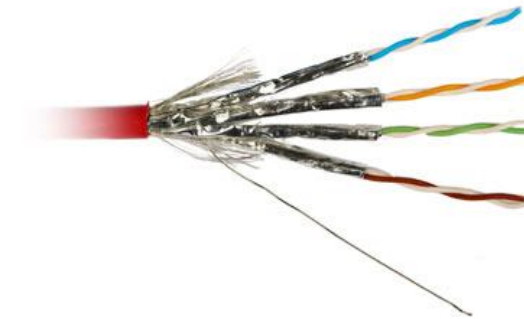
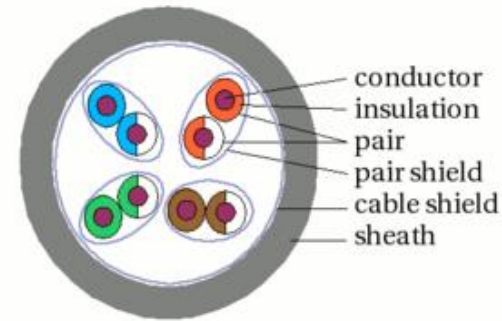
STP – S/STP - 1



STP



S/STP



- Speed and throughput: 10 - 100 Mbps
- Average \$ per node: Moderately Expensive
- Media and connector size: Medium to Large
- Maximum cable length: 100m



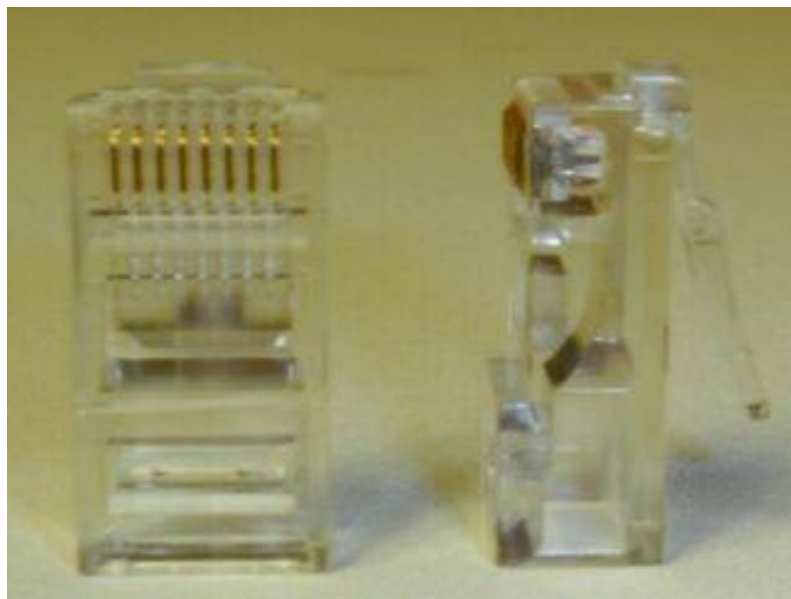
STP – S/STP - 2



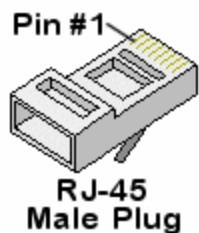
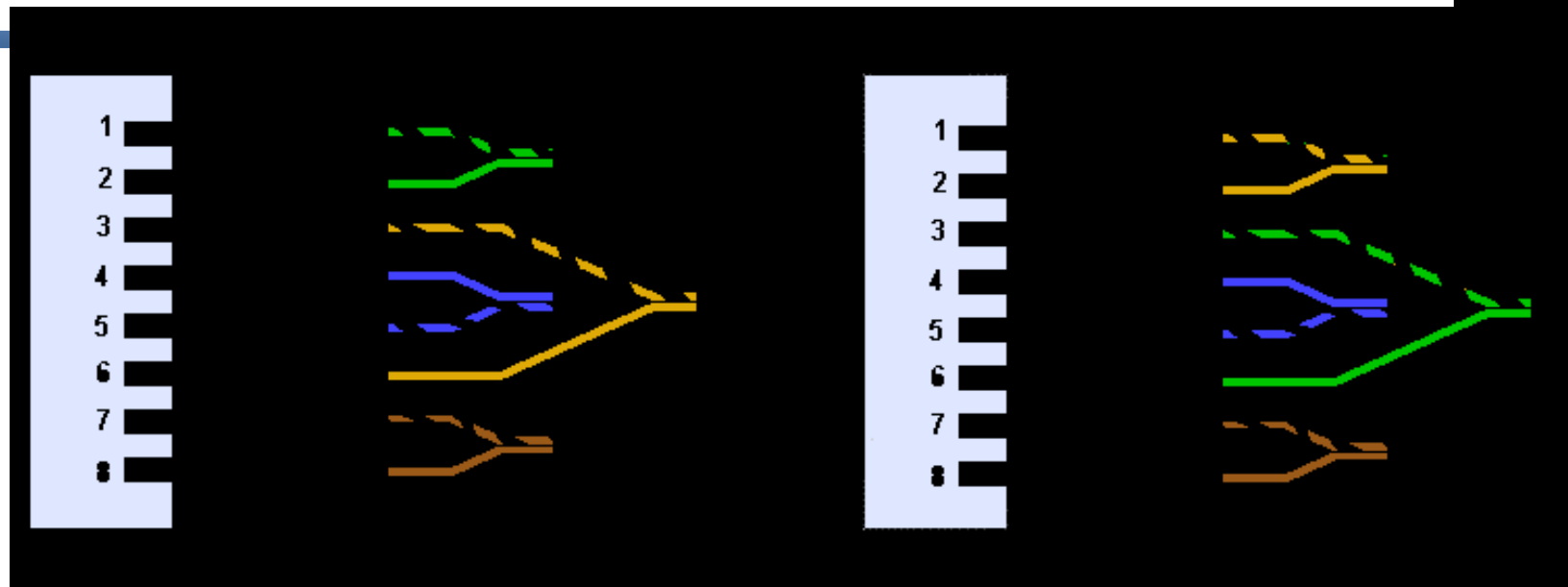
- ❑ Chi phí:
 - Đắt hơn ThinNet và UTP
 - nhưng rẻ hơn ThickNet và cáp quang
- ❑ Tốc độ: 10 – 100Mbps
- ❑ Độ suy dần (*attenuation*): cao
- ❑ Nhiễu: chống nhiễu tốt
- ❑ Độ dài tối đa: 100m
- ❑ Đầu nối: đầu nối DIN (DB-9), RJ-45



Đầu bấm rj-45

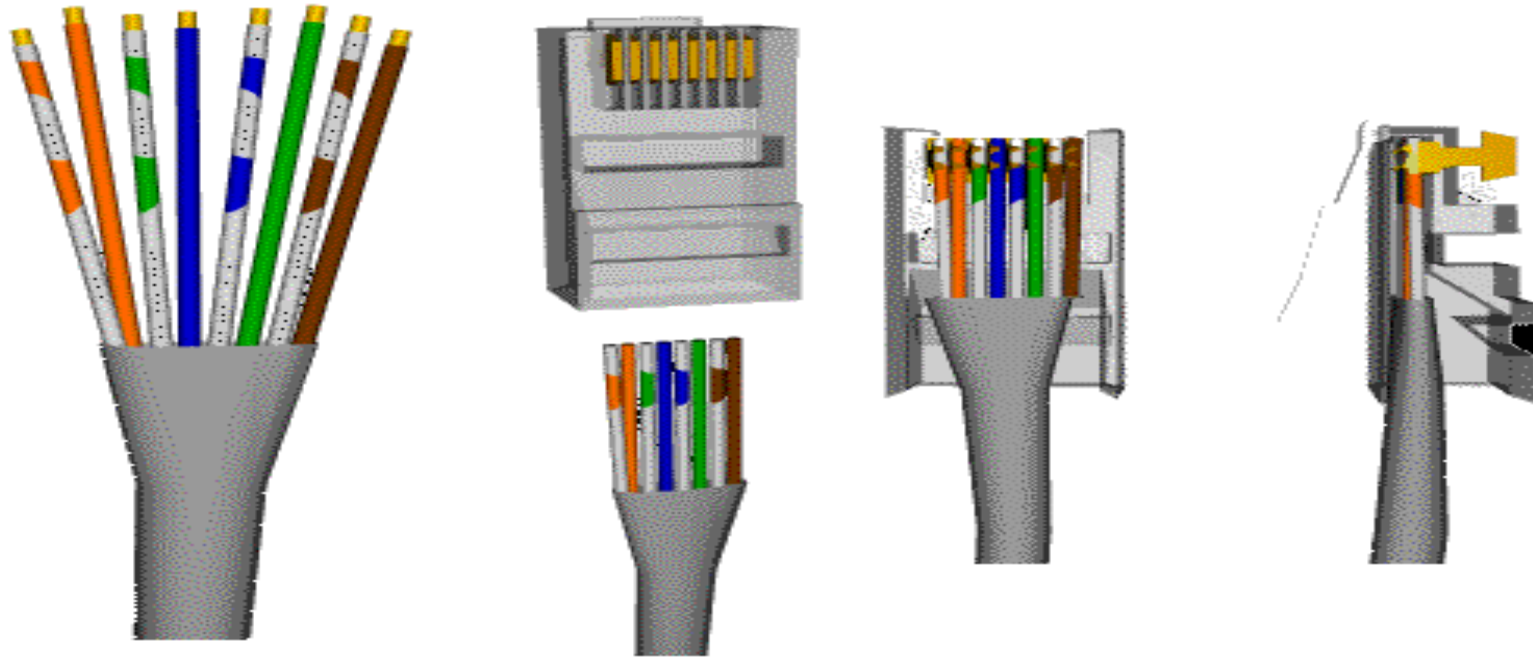


Chuẩn bấm cáp với đầu bấm rj-45



Pin	Name	Description	TIA/EIA 568A	TIA/EIA 568B
1	TX_D1+	Tranceive Data+	white and green	white and orange
2	TX_D1-	Tranceive Data-	green	orange
3	RX_D2+	Receive Data+	white and orange	white and green
4	BI_D3+	Bi-directional Data+	blue	blue
5	BI_D3-	Bi-directional Data-	white and blue	white and blue
6	RX_D2-	Receive Data-	orange	green
7	BI_D4+	Bi-directional Data+	white and brown	white and brown
8	BI_D4-	Bi-directional Data-	brown	brown

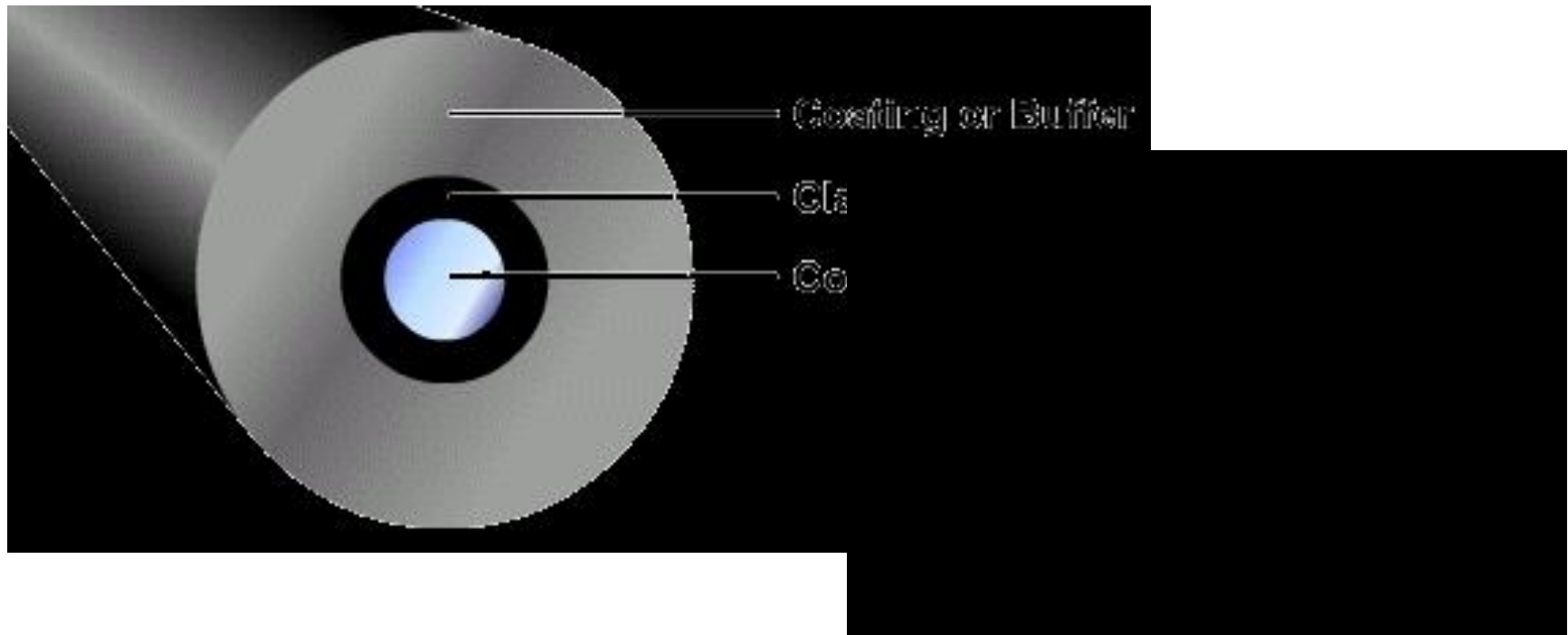
CÁCH BẮM CÁP XOẼN



Bấm cáp xoắn với đầu bấm RJ-45



Cáp quang (Fiber optic) – mô tả



Cáp quang – mô tả



- Dùng sóng ánh sáng để truyền
 - Sự khúc xạ
 - Sự phản xạ
- Không bị nhiễu
- Độ suy dần: thấp
- Chiều dài cáp: rất lớn, đến vài Km
- Chi phí: rất đắt tiền
- Khó lắp đặt

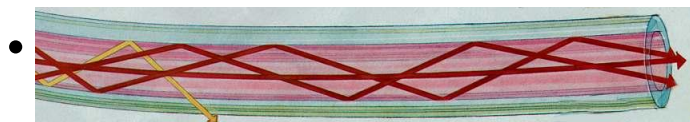


Cáp quang – phân loại



□ Mode: đường đi của ánh sáng khi vào trong lõi của cáp quang

□ Phân loại:

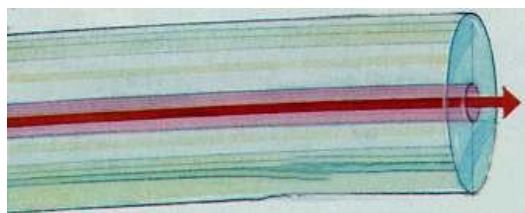


step-index multimode















graded-index multimode

- Đơn mode (single



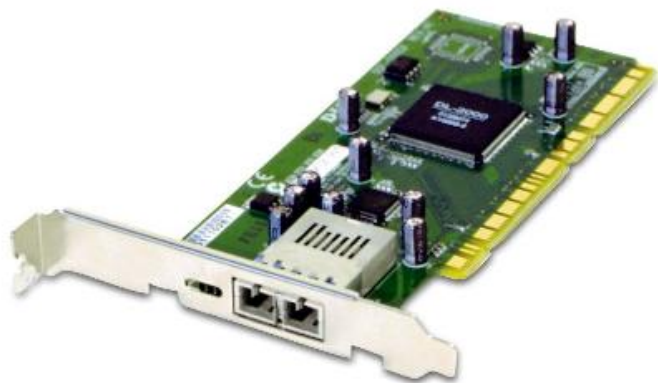
Cáp quang - connector



Fiber Connector Styles			
 <p>ST Connector A slotted bayonet type connector. This connector is one of the most popular styles.</p>	 <p>SC Connector A push/pull type connector. This connector has emerged as one of the most popular styles.</p>	 <p>FC Connector A slotted screw-on type connector. This connector is popular in singlemode applications.</p>	 <p>SMA Connector A screw-on type connector. This connector is waning in popularity.</p>
 <p>FDDI Connector A push/pull type dual connector. This connector is one of the more popular styles.</p>	 <p>Mini-BNC Connector A bayonet style connector using the traditional BNC connection method.</p>	 <p>Biconic Connector A screw-on style connector. This connector is almost obsolete.</p>	 <p>MT-RJ Connector A new RJ style housing fiber connector with two fiber capability.</p>
 <p>ST Feedthru A slotted bayonet type feedthru. ST connectors are one of the most popular styles.</p>	 <p>SC Feedthru A push/pull type feedthru. SC connectors are one of the most popular styles.</p>	 <p>FDDI Feedthru A push/pull type feedthru. FDDI connectors are popular in both singlemode and multimode applications.</p>	 <p>FC Feedthru A slotted screw-on type feedthru. FC connectors are popular in singlemode applications.</p>



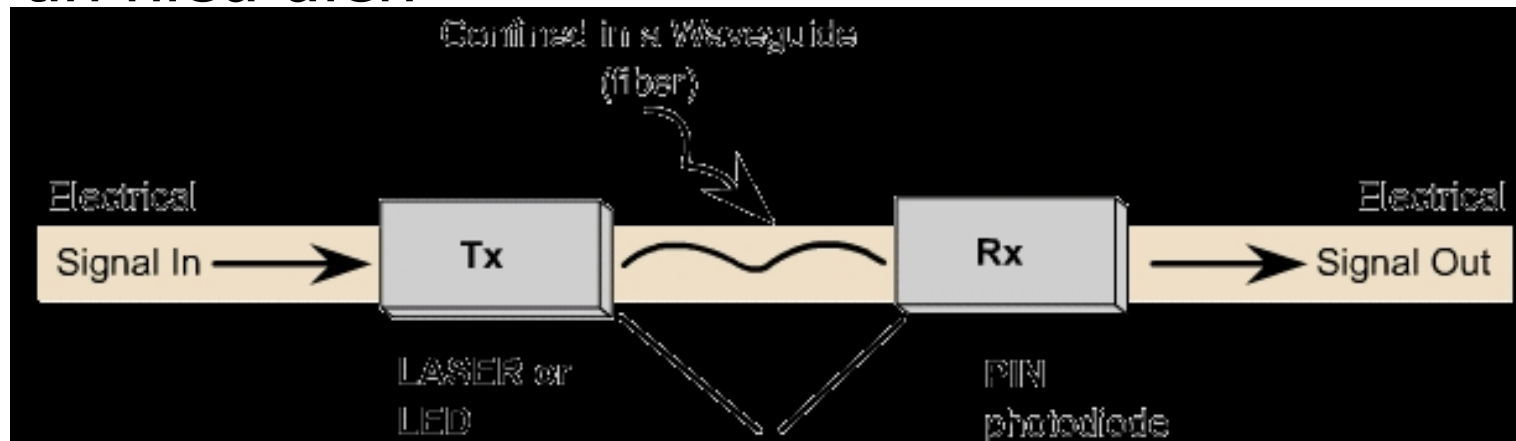
Cáp quang – cách kết nối



Cáp quang – thành phần



- ❑ Tx: biến đổi tín hiệu điện thành xung ánh sáng
 - LED: dùng cho đa mode
 - LASER: dùng cho đơn mode
- ❑ Rx (PIN photodiode): chuyển xung ánh sáng thành tín hiệu điện



Nội dung



1. Đặc tính của một loại PTTD
2. PTTD hữu tuyến
3. PTTD vô tuyến



PTTD vô tuyến



- ❑ Là loại đường truyền sử dụng không khí làm vật mang tín hiệu thay cho cáp.
- ❑ Các loại đường truyền vô tuyến:
 - Radio
 - Viba
 - Tia hồng ngoại
 - Laser
 - Vệ tinh (satellites)
 - ...



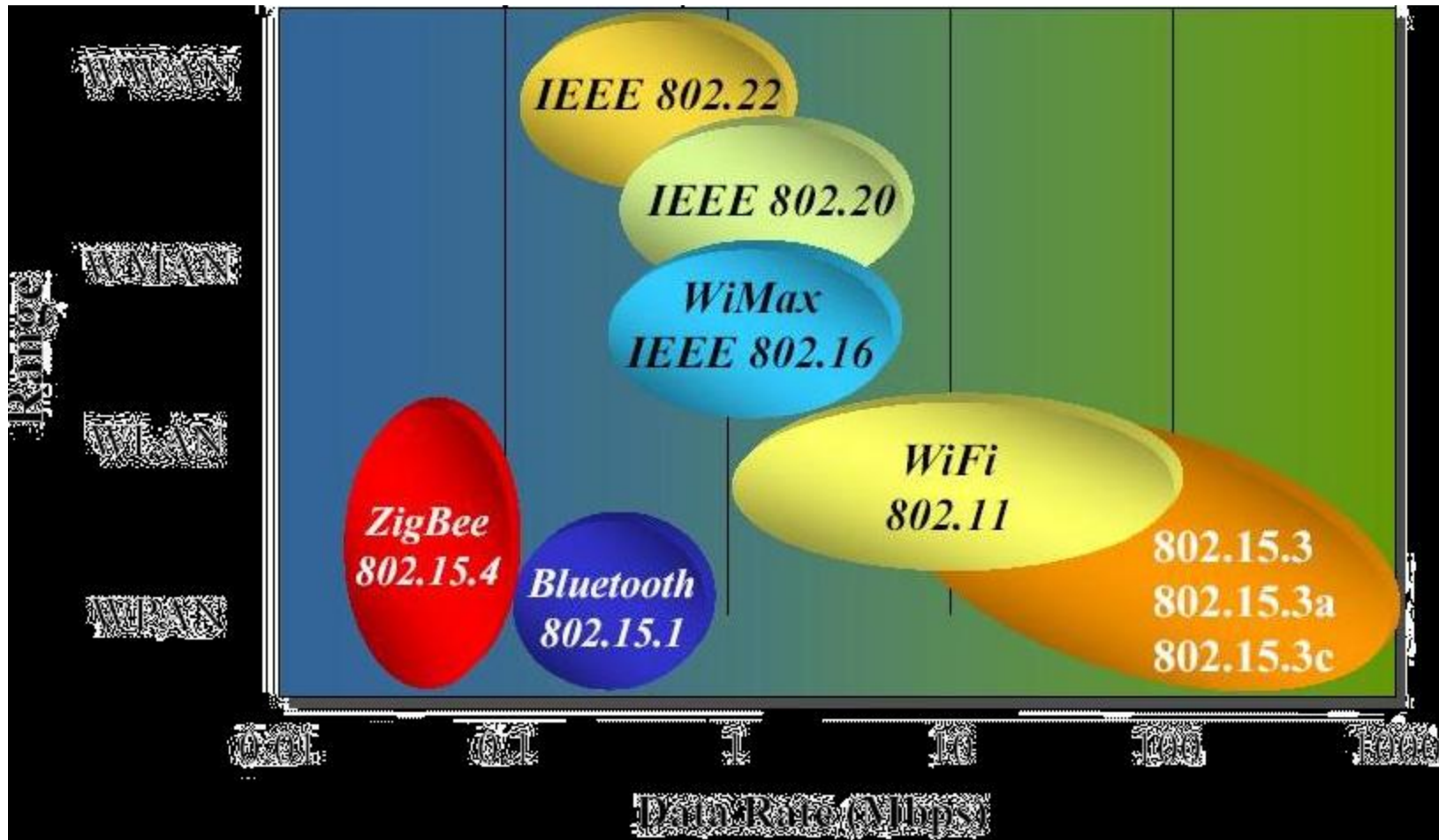
Tại sao dùng PTTD vô tuyến?



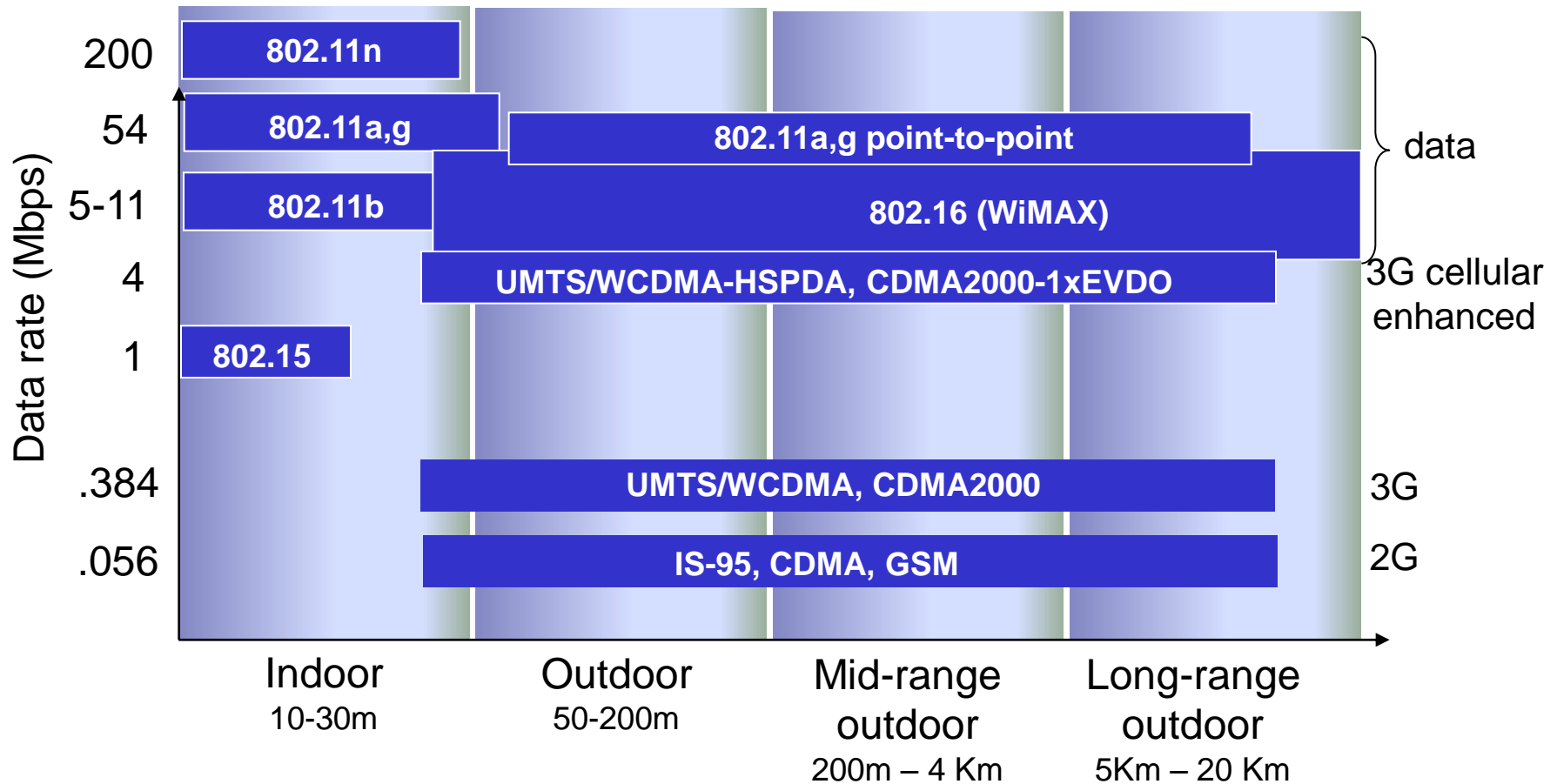
- Loại bỏ các ràng buộc vật lý
 - Không thể đi cáp qua những địa hình phức tạp
 - Sử dụng các thiết bị di động
- Thiết lập đường truyền tạm thời
- Bất lợi: security



Phân bố các chuẩn wireless - 1



Phân bố các chuẩn wireless - 2



So sánh các loại PTTD



Medium	Cost	Speed	Attenu	Interference	Security
UTP ³	Low	1-100M	High	High	Low
STP ³	Medium	1-100M	High	Medium	Low
Coax	Medium	1M-1G	Medium	Medium	Low
Fiber	High	10M-2G	Low	Low	High
Radio	Medium	1-10M	Varies	High	Low
Microwav	High	1M-10G	Varies	High	Medium
Satellite	High	1M-10G	Varies	High	Medium
Cellular	High	9.6-19.2K	Low	Medium	Low



Chương 05

Thiết bị mạng

MẠNG MÁY TÍNH

Tháng 09/2011

Nội dung



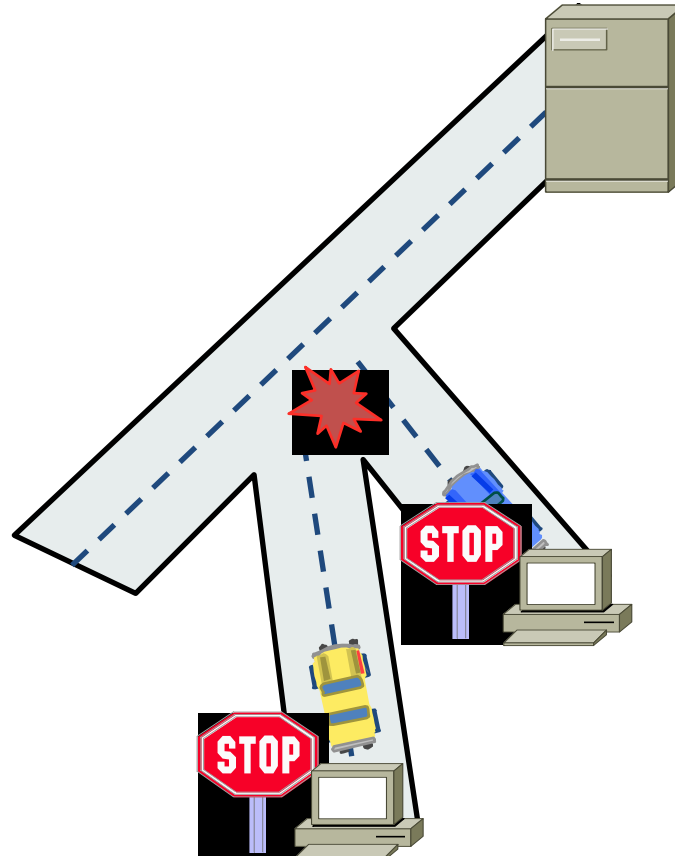
- Các thiết bị mạng
- Collision domain & Broadcast domain



Collision



- ❑ Collision (đụng độ): khi có hai hay nhiều node cùng gửi DL lên đường truyền chia sẻ cùng lúc



☐ Chức năng

- Hỗ trợ truy cập mạng
 - NIC
- Dụng để phân tách mạng hoặc mở rộng mạng
 - Router
 - Switch, Bridge, hub, repeater, gateway
- Dụng để truy cập từ xa
 - Modem, ADSL modem

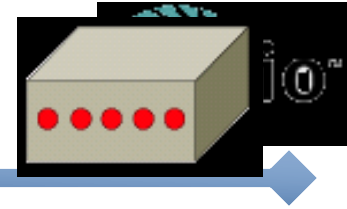
Các thiết bị mạng



- Tầng 1: modem, repeater, hub
- Tầng 2: bridge, switch
- Tầng 3: router
- Khác: NIC, access point

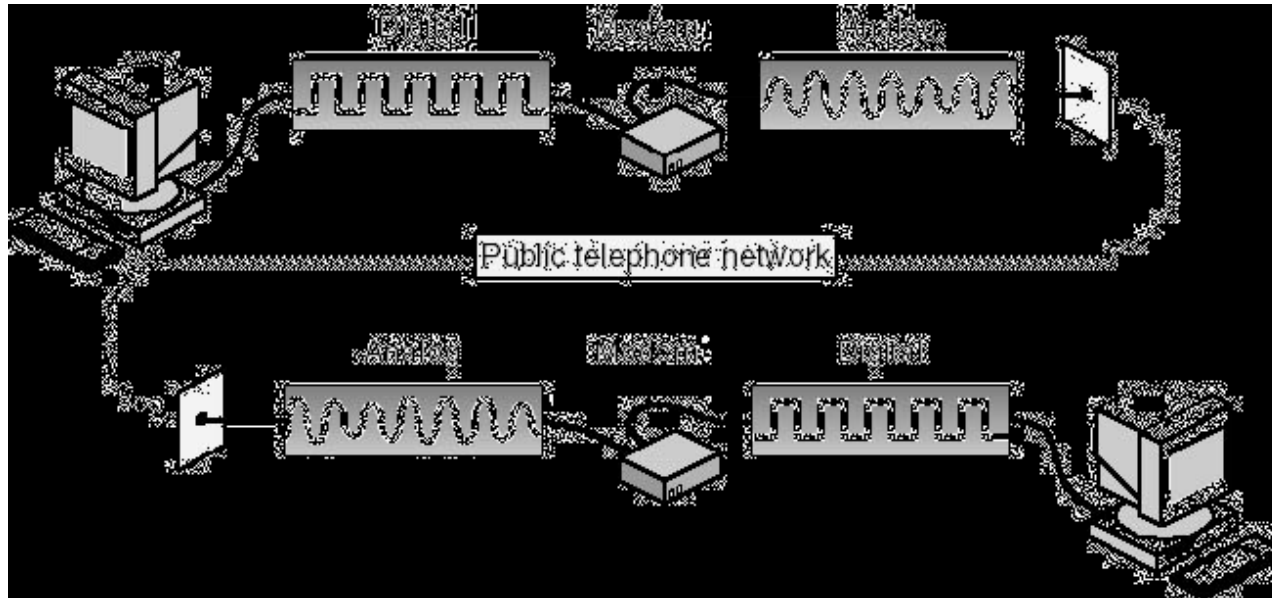


Modem - 1

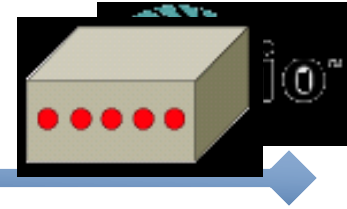


❑ **MODEM = MODulate and DEModulate**

❑ Là thiết bị cho phép các máy tính truyền thông với nhau qua mạng điện thoại



Modem - 2



□ Chức năng:

- Điều chế [**M**odulate]: chuyển đổi tín hiệu số (digital) trên máy tính thành tín hiệu tương tự (analog) trên điện thoại.
- Giải điều chế [**D**emodulate]: chuyển đổi tín hiệu tương tự trên điện thoại thành tín hiệu số trên máy tính



Repeater - 1

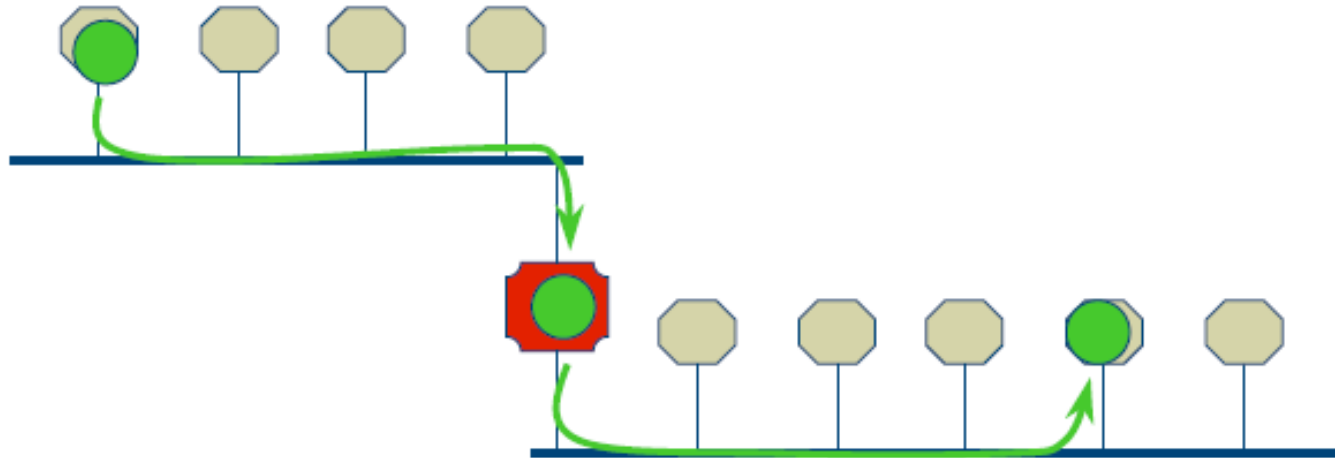


- ❑ Repeater là thiết bị mạng nối kết **2** nhánh mạng
 - nhận tín hiệu ở một nhánh mạng
 - khuếch đại tín hiệu (không xử lý nội dung)
 - truyền đi tiếp vào nhánh mạng còn lại

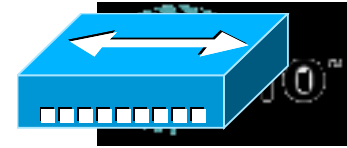
❑ S



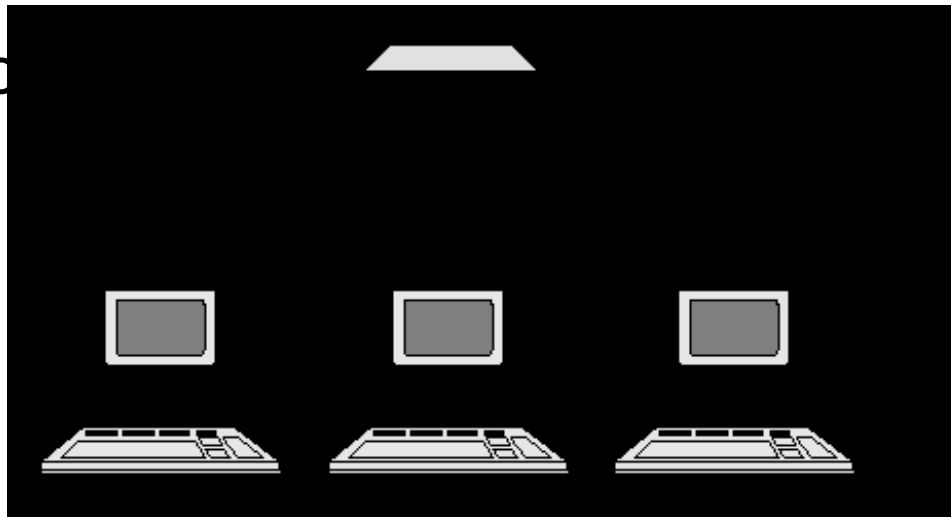
Repeater – minh họa tín hiệu mạng



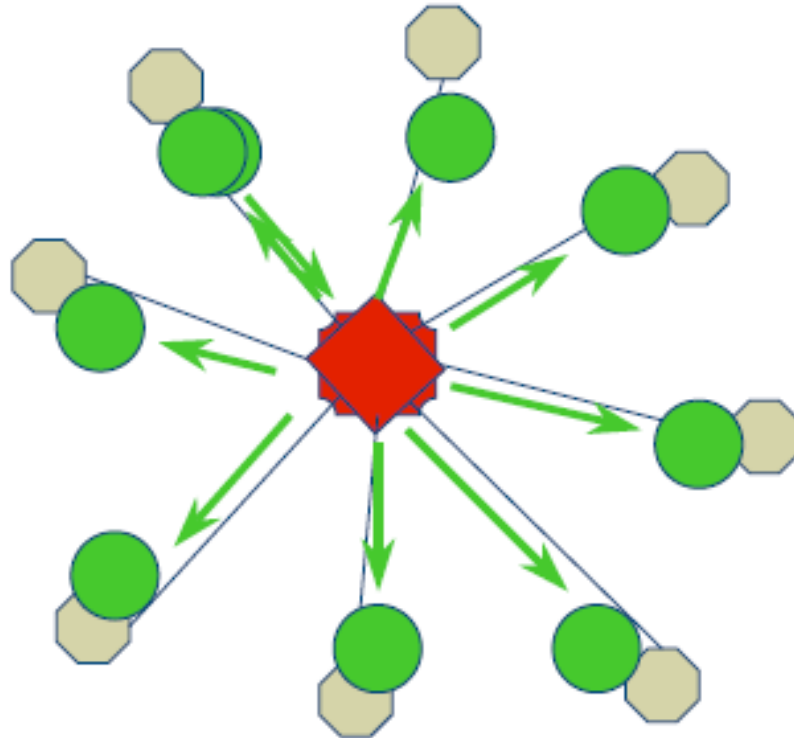
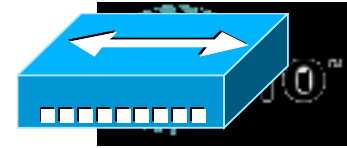
Hub



- ❑ Là thiết bị mạng cho phép tập kết dây dẫn mạng
- ❑ Tín hiệu vào 1 port của Hub sẽ được chuyển ra tất cả các port
 - Mỗi port



Hub – minh họa tín hiệu mạng



Hub – phân loại



❑ Passive hub:

- Không khuếch đại tín hiệu

❑ Active Hub

- Khuếch đại tín hiệu
- Như 1 repeater nhiều cổng

❑ Intelligent Hub

- Là 1 active hub
- Chuyển mạch (switching): chuyển tín hiệu đến đúng port của máy nhận



Repeater & hub



- ❑ Chức năng: Tái sinh tín hiệu mạng và chuyển tín hiệu mạng đến các segment mạng còn lại
- ❑ Đặc điểm:
 - Không thể liên kết các segment khác nhau
 - Khác đường mạng
 - Khác phương pháp truy cập đường truyền
 - dùng phương tiện truyền dẫn khác nhau
 - Không thể “nhận dạng” packet
 - Không cho phép giảm tải mạng
 - Cho phép mở rộng mạng dễ dàng



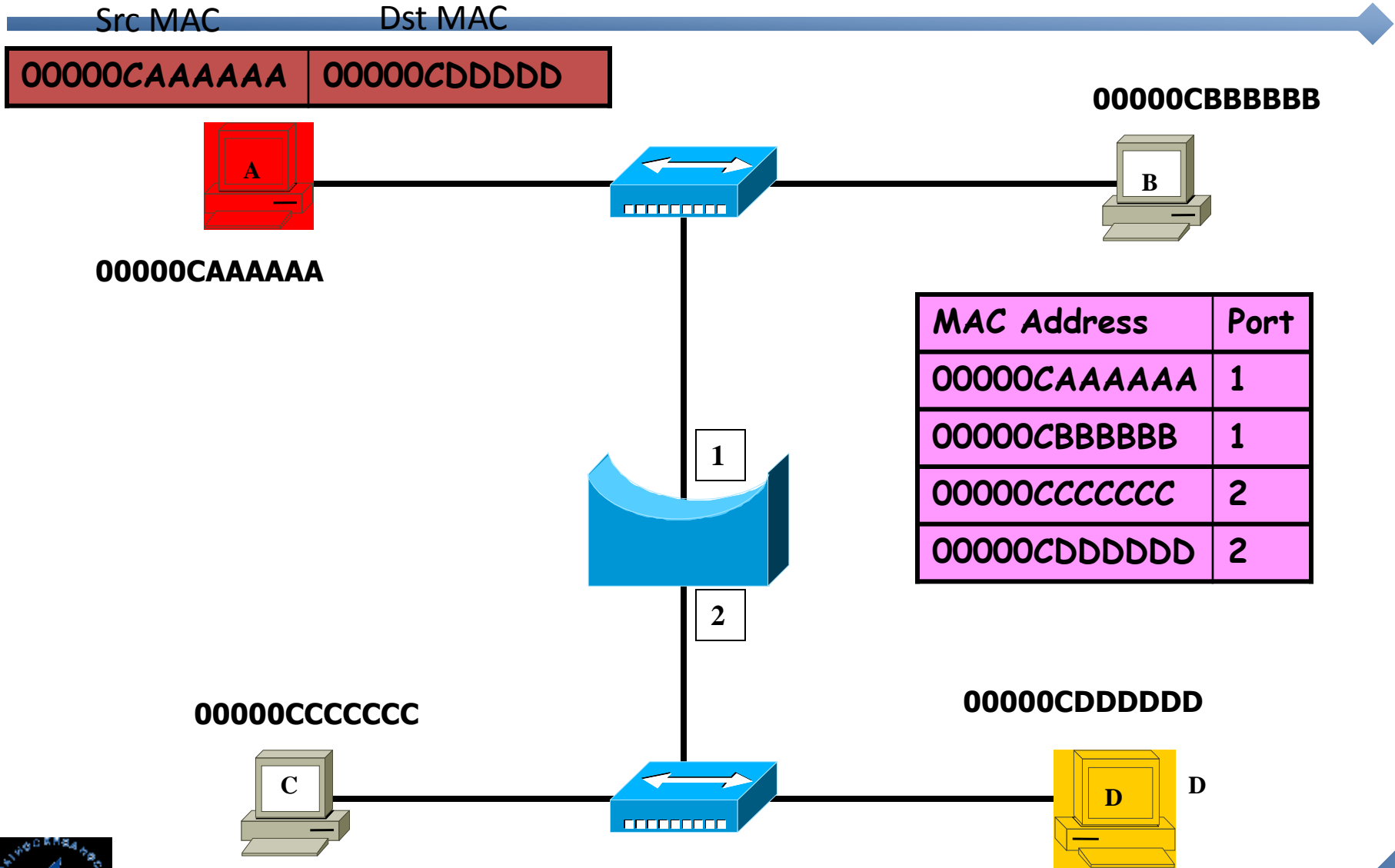
Bridge - 1



- ❑ Là thiết bị mạng cho phép nối kết **2** nhánh mạng vật lý
- ❑ Chức năng: *chuyển có chọn lọc* các gói tin đến nhánh mạng chứa trạm nhận gói tin.
 - Duy trì bảng địa chỉ
 - MAC – Port
 - khởi tạo và duy trì tự động hoặc thủ công
 - Nếu trạm nhận cùng segment với trạm gửi, hủy gói tin; ngược lại chuyển gói tin đến segment đích



Bridge – minh họa tín hiệu mạng



Bridge – minh họa tín hiệu mạng

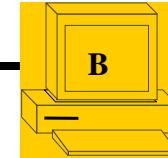
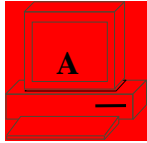


Src MAC

Dst MAC

00000CAAAAAA 00000CBBBBBB

00000CBBBBBB



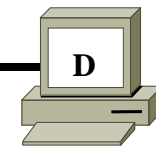
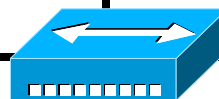
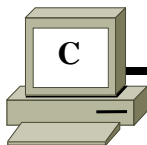
00000CAAAAAA

MAC Address	Port
00000CAAAAAA	1
00000CBBBBBB	1
00000CCCCCCC	2
00000CDDDDDD	2

1

2

00000CCCCCCC



00000CDDDDDD

D



Bridge - 3



□ Đặc điểm:

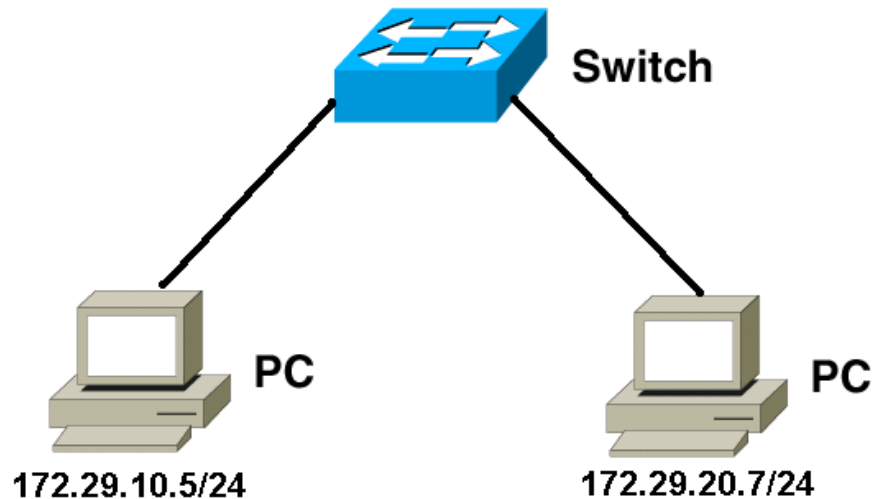
- Cho phép mở rộng cùng một mạng logic với nhiều kiểu chạy cáp khác nhau
- Tách một mạng thành nhiều phần nhằm giảm lưu lượng mạng.
- Chậm hơn repeater do phải xử lý các gói tin
- Không có khả năng tìm đường đi tối ưu trong trường hợp có nhiều đường đi.
- Đắt tiền hơn repeater



Switch - 1



- ❑ Là 1 bridge nhiều port
- ❑ Hỗ trợ full-duplex
- ❑ Duy trì bảng CAM (Content Addressable Memory)
 - MAC – P



Switch - 2



- ❑ Chức năng:
 - Học địa chỉ MAC (self –learning)
 - Filtering/Forwarding
 - Tránh loop

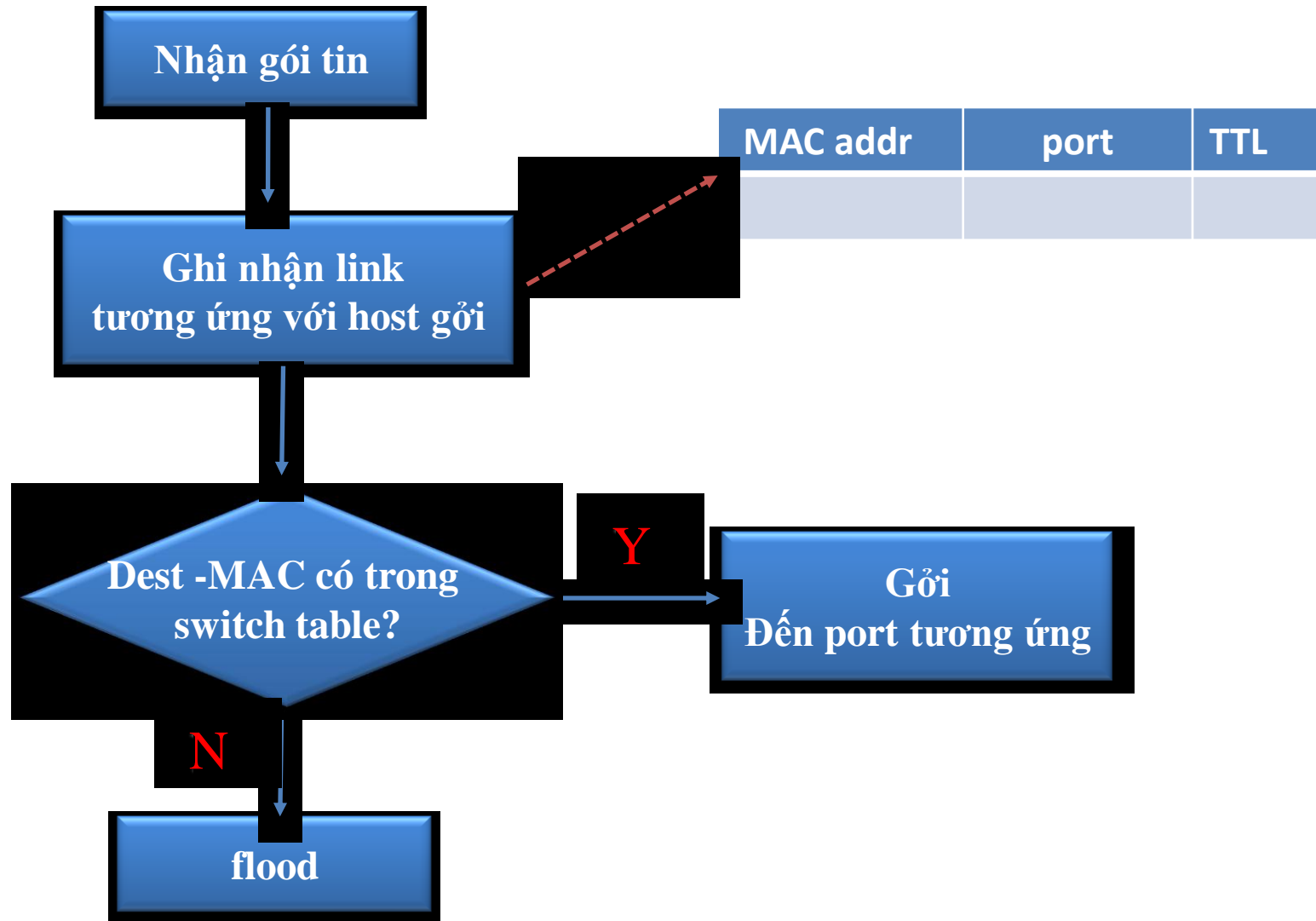
❑ Các chế độ chuyển mạch:

- Store-and-forward
 - Đọc hết nội dung gói tin
 - Đảm bảo chính xác
- Cut-through
 - Đọc 14 bytes đầu tiên
 - Không phát hiện được gói tin bị lỗi
- Fragment-free
 - Đọc 1 phần gói tin

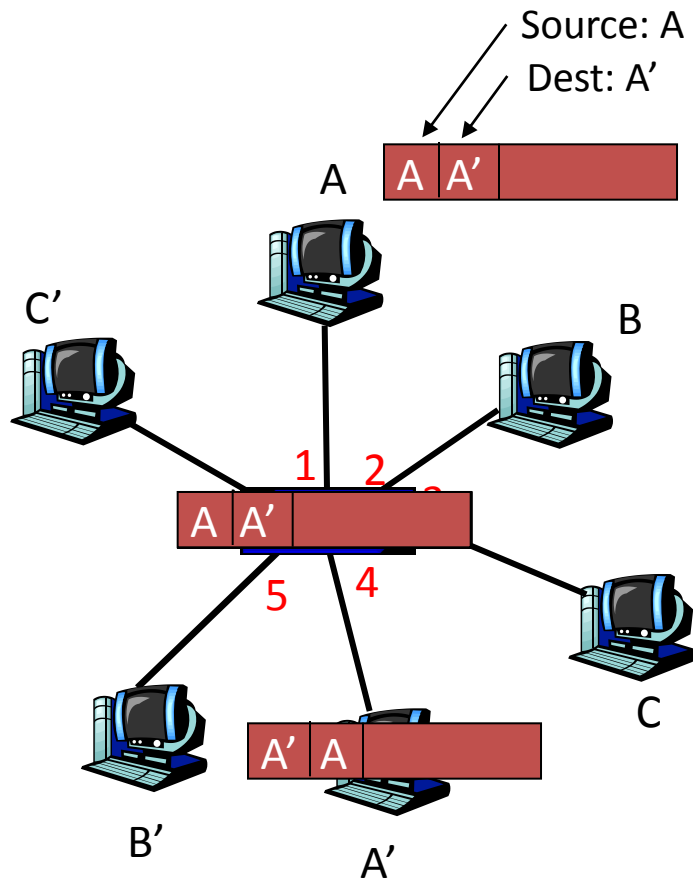
Preamble	Des. Add	Sour. Add	Length	Data	FCS	
7 Bytes	1 Byte	2/6 Bytes	2/6 Bytes	2 Bytes	46 - 1500 Bytes	4 Bytes



Switch – học địa chỉ mac - 1



Switch – học địa chỉ mac - 2

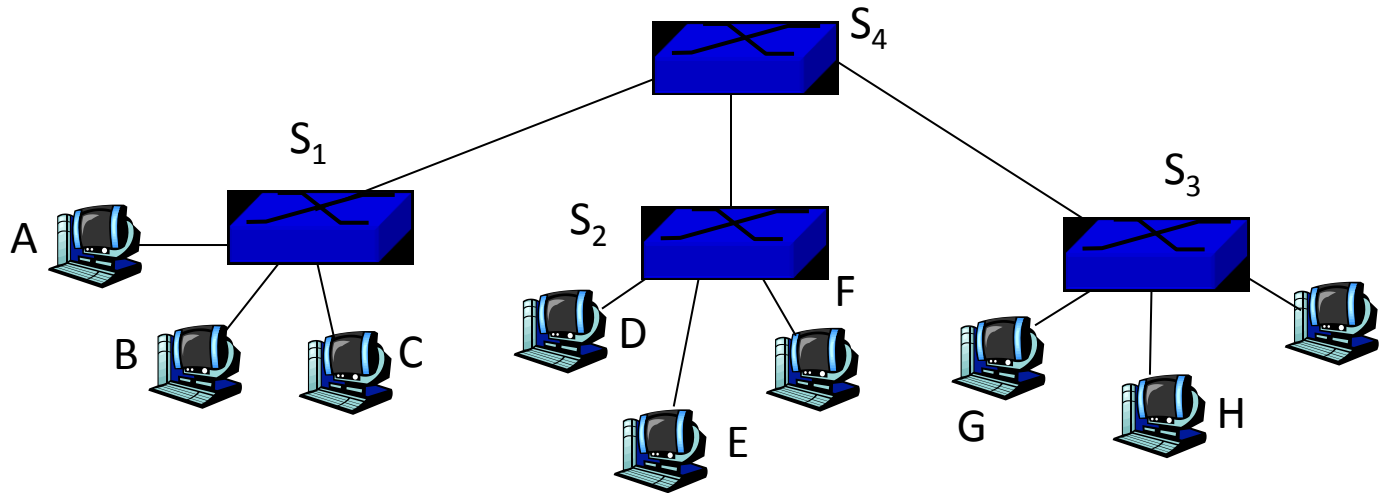


Switch table
(giả sử ban đầu rỗng)

MAC addr	port	TTL
A	1	60
A'	4	60



Switch – học địa chỉ mac - 3



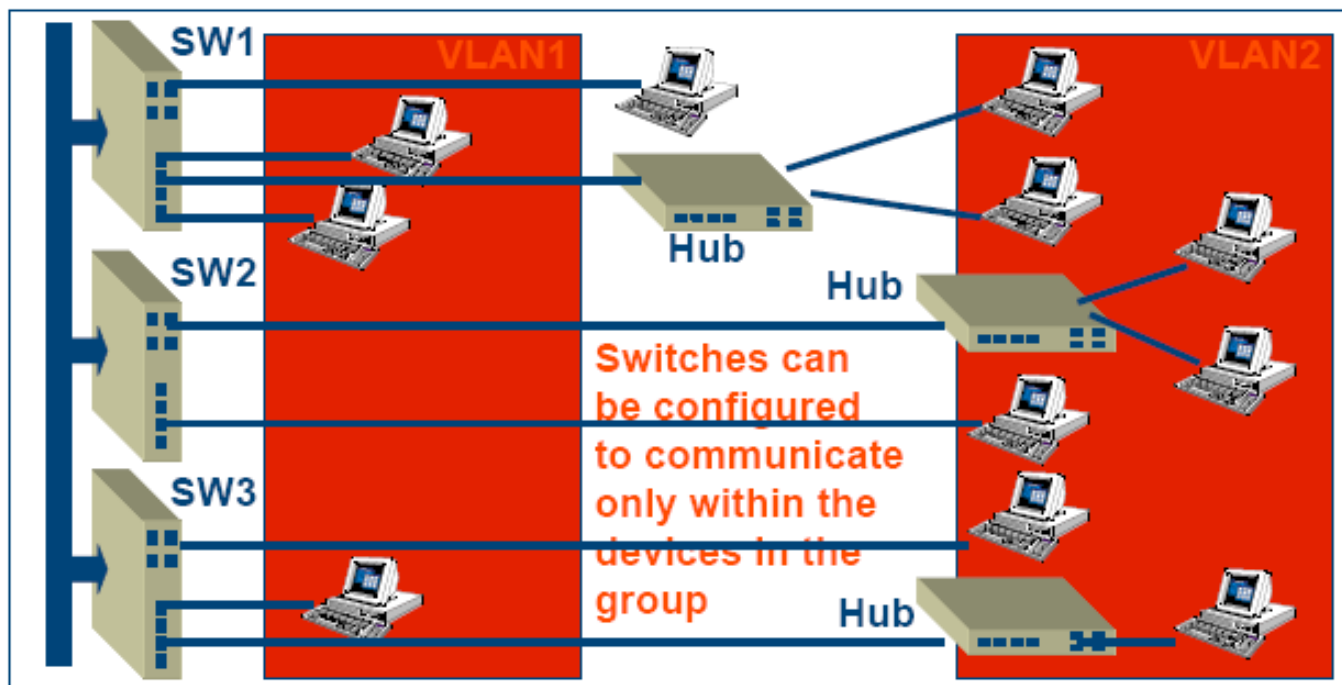
□ Hỏi: Khi A gửi cho F 1 gói tin???



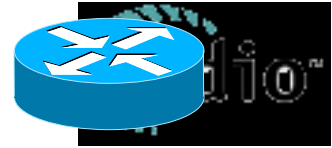
Switch – VLAN



- VLAN = Virtual LAN
- Nhóm một số port thành 1 mạng LAN ảo



Router



☐ Chức năng:

- Nối kết các mạng logic khác nhau.
- Sử dụng địa chỉ logic (IP) để xử lý gói tin
- Định tuyến (Routing)
 - Chạy các thuật toán định tuyến (OSPF, RIP, BGP,...) → tạo ra bảng định tuyến
- Chuyển tiếp (Forwarding)
 - Chuyển gói tin từ cổng vào (incoming port) ra cổng ra (outcoming port)



NIC



- ❑ NIC = Network Interface Card
- ❑ Là thiết bị chuyển đổi tín hiệu máy tính thành tín hiệu trên phương tiện truyền dẫn và ngược lại
- ❑ Cung cấp kết nối vật lý đến phương tiện truyền dẫn



Access Point



- ❑ Là thiết bị cho phép thiết bị truy cập mạng không dây
- ❑ Đóng vai trò như 1 hub
- ❑ Thành phần:
 - Bộ thu: thu tín hiệu radio và chuyển thành tín hiệu mạng
 - Bộ phát: chuyển tín hiệu mạng thành tín hiệu radio
- ❑ Ngày nay, một số AP còn tích hợp chức năng của 1 Router



THIẾT BỊ MẠNG



Nhu cầu	Thiết bị
Kết nối nhiều máy tính trong 1 Net	SW, Hub, Bridge
Kết nối nhiều Net	Router
Truyền qua điện thoại	Modem
Kéo dài dây cáp	Repeater
Thiết lập mạng không dây	AP



Nội dung



- Các thiết bị mạng
- Collision domain & Broadcast domain



Collision domain - Broadcast domain



- ❑ Collision domain (miền đụng độ): là miền có khả năng xảy ra đụng độ
 - Là miền dùng chung (chia sẻ)
 - Hai segment thuộc cùng 1 collision domain nếu chúng gây ra collision khi đồng thời gửi dữ liệu xuống đường truyền

- ❑ Broadcast domain (miền broadcast): là miền nhận được gói tin broadcast
 - Gồm nhiều collision domain (1 – n)
 - Collision domain A và B thuộc cùng 1 broadcast domain nếu các node mạng trong collision domain B nhận được gói tin broadcast từ 1 node trong collision domain A



Các thiết bị mạng



❑ Thiết bị mở rộng collision domain:

- Repeater
- Hub
- ...

❑ Thiết bị phân tách collision domain

- Switch
- Bridge

❑ Thiết bị phân tách broadcast domain

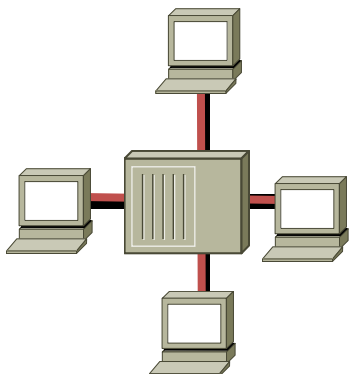
- Router
- Switch (VLAN)



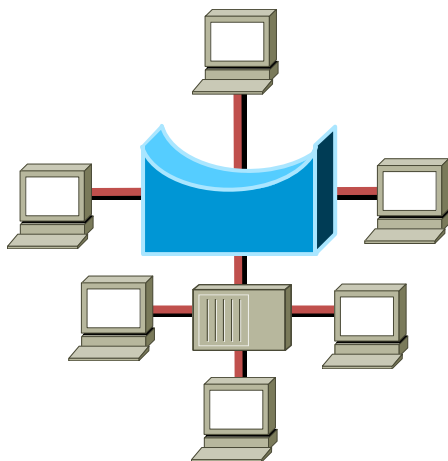
Ví dụ 1



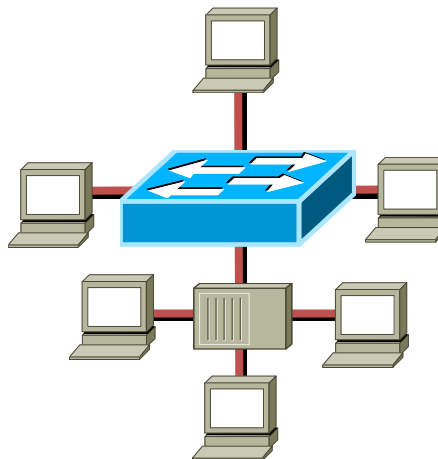
Hub



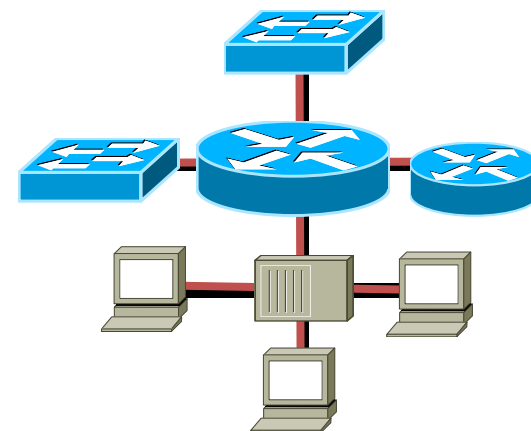
Bridge



Switch



Router



Collision Domains:

1

4

4

4

Broadcast Domains:

1

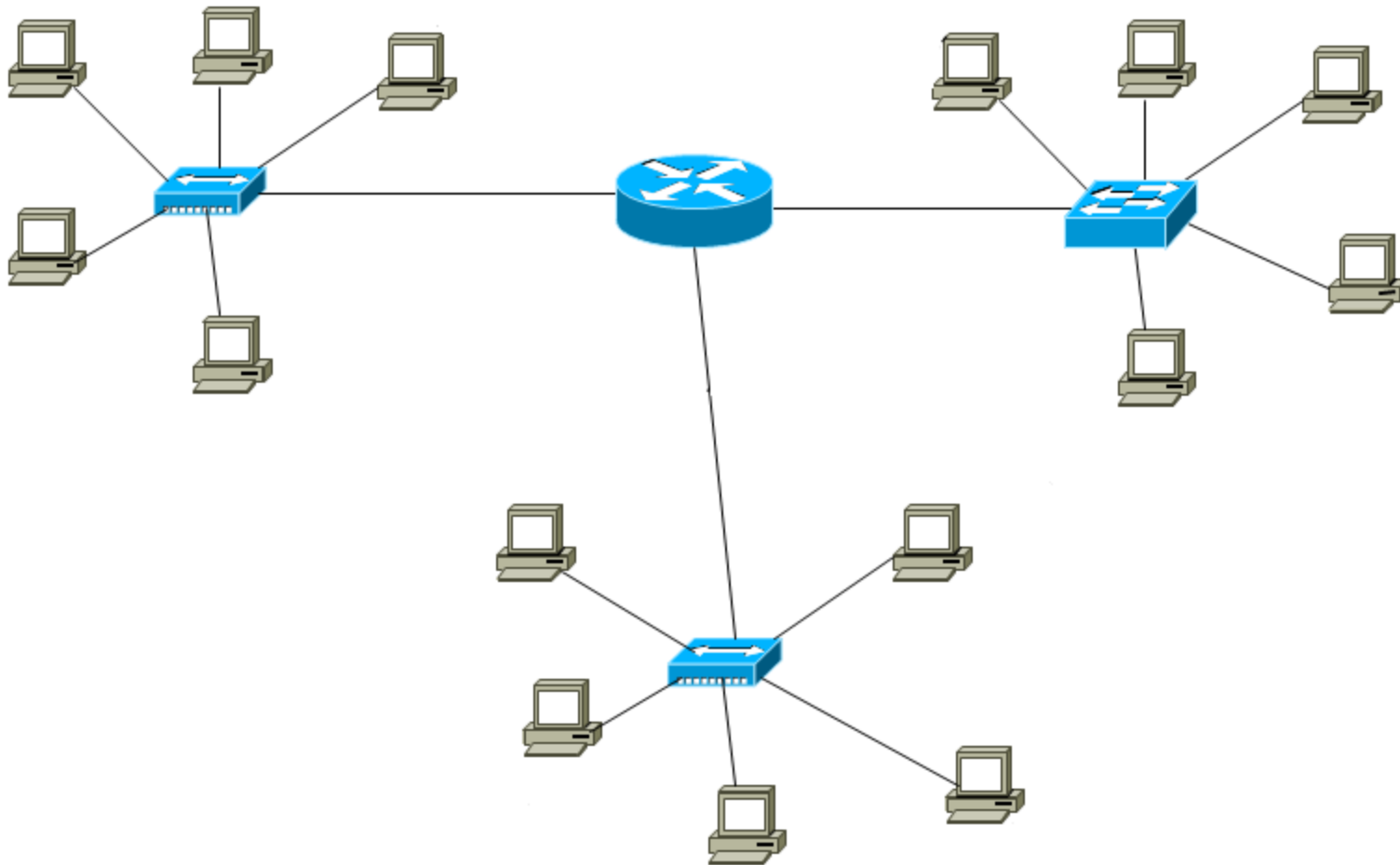
1

1

4



Ví dụ 2



Tài liệu tham khảo



- ❑ Slide của J.F Kurose and K.W. Ross về Computer Networking: A Top Down Approach
- ❑ <http://www.eie.polyu.edu.hk/~ensmall/eng224>

