



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Mạng máy tính

Chương 1: Tổng quan

Về môn học

- Mã HP: IT3080
- Tên học phần: Mạng máy tính
- Khối lượng: 3(3-0-1-6)
- Đánh giá:
 - Quá trình (50%):
 - ❖ Thực hành (30%)
 - ❖ Kiểm tra giữa kỳ (20%)
 - Cuối kỳ (50%): thi tại phòng máy
- Điểm danh:
 - Đi học đầy đủ, làm đầy đủ bài tập: +1 điểm
 - Vắng mặt 3-4 buổi: trừ 1 điểm
 - Vắng mặt 5 buổi trở lên: trừ 2 điểm

Tài liệu tham khảo

1. **Computer Network**, 5th Edition, *Andrew Tannenbaum*, Pearson Education 2011
2. **Networking: a top-down approach featuring the Internet**, 6th Edition, *James F. Kurose, Keith W. Ross*, Addison Wesley 2012
3. **TCP/IP tutorial and technical overview**, *Lydia Parziale, David T.Britt*, IBM Redbooks 2006
4. **Data and Computer Communications**, 8th Edition *William Stallings*, Pearson Prentice Hall 2007

Giảng viên

Tống Văn Vạn

Trung tâm An toàn An ninh Thông tin (BKCS)

Trường Công nghệ Thông tin và Truyền thông

Đại học Bách khoa Hà Nội

Email : vantv@soict.hust.edu.vn

Phòng làm việc: phòng 405 – Nhà B1 - ĐHBKHN

1. Cơ bản về mạng máy tính

- Internet & lịch sử phát triển
- Các khái niệm mạng máy tính
- Kiến trúc mạng

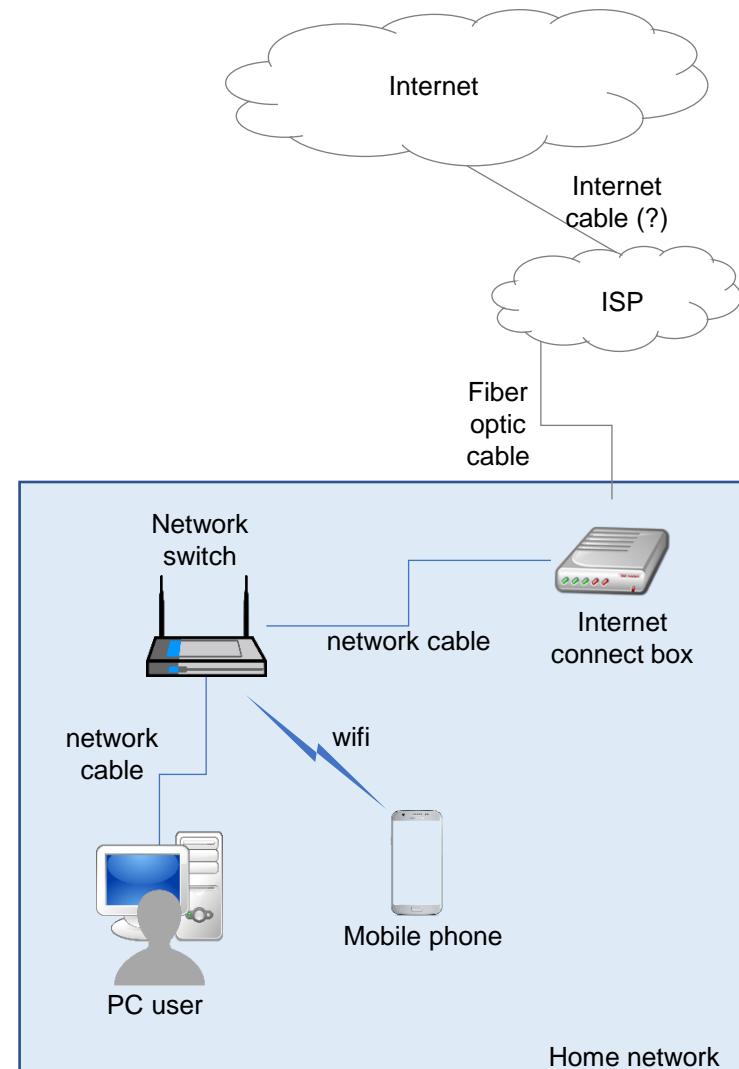
Mạng Internet “đơn giản”

- Internet “đơn giản”:

- Kết nối Internet từ nhà
- Duyệt web, gửi email
- Gọi điện với bạn bè

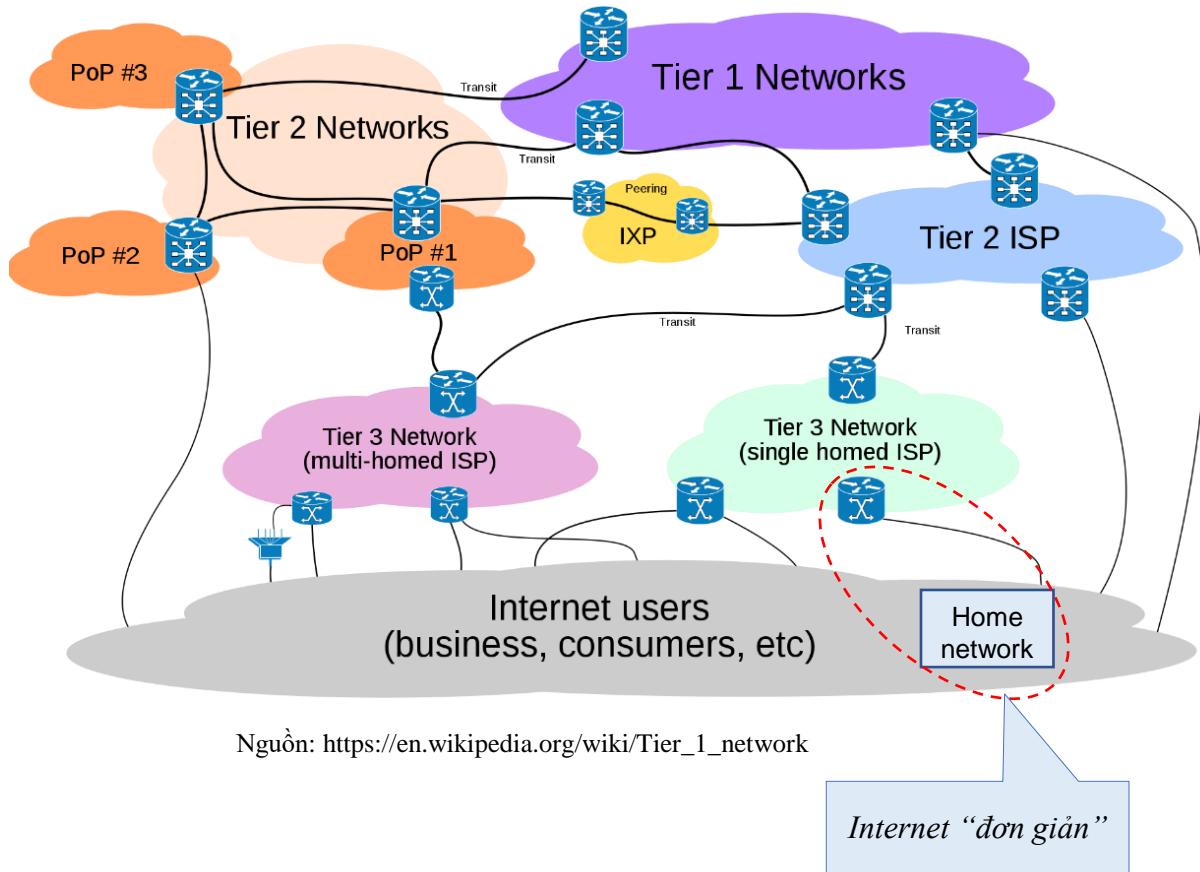
- Các thành phần:

- Trạm làm việc: PC, mobile phone
- Đường truyền: có dây, không dây
- Phần mềm sử dụng: PC web, phone voice chat, v.v...
- Thiết bị kết nối mạng: switch, Internet connect box, v.v..
- Đơn vị cung cấp kết nối Internet (ISP)
- Đám mây Internet



Mạng Internet đầy đủ

- Internet: hệ sinh thái toàn cầu, sản phẩm nhân tạo lớn nhất
- Mạng xương sống (backbone)
- Mạng ISP
- Mạng home/office
- Trạm làm việc
- Phần mềm ứng dụng



Nguồn: https://en.wikipedia.org/wiki/Tier_1_network

Nguồn gốc Internet: ARPANET

- Bắt đầu từ một thí nghiệm của dự án của Advanced Research Project Agency (ARPA)¹ – Bộ quốc phòng Mỹ
- Một liên kết giữa hai nút (IMP tại UCLA và IMP tại SRI) → ARPANET
- Hợp tác giữa Bob Kahn² tại DARPA và Vint Cerf³ tại đại học Stanford

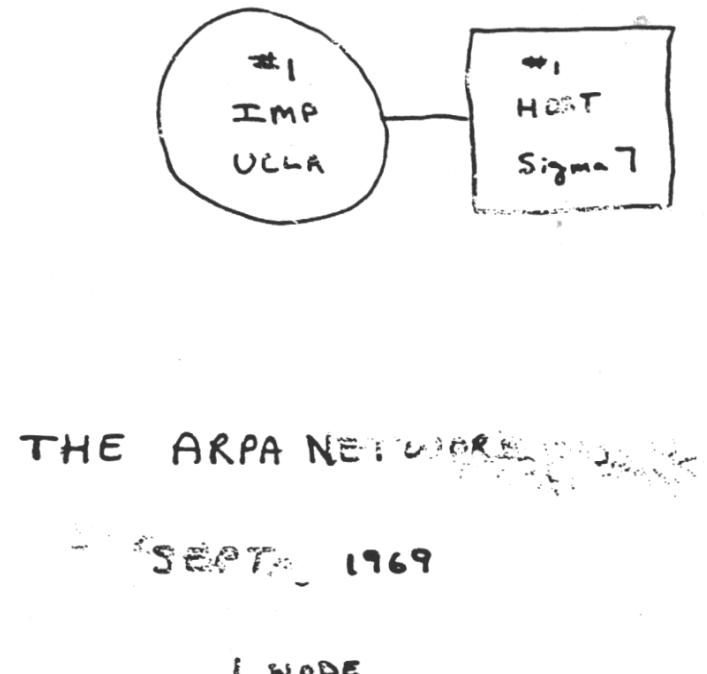


FIGURE 6.1 Drawing of September 1969
(Courtesy of Alex McKenzie)

Source: <http://www.cybergeography.org/atlas/historical.html>

¹ <https://en.wikipedia.org/wiki/DARPA>

² https://en.wikipedia.org/wiki/Bob_Kahn

³ https://en.wikipedia.org/wiki/Vint_Cerf

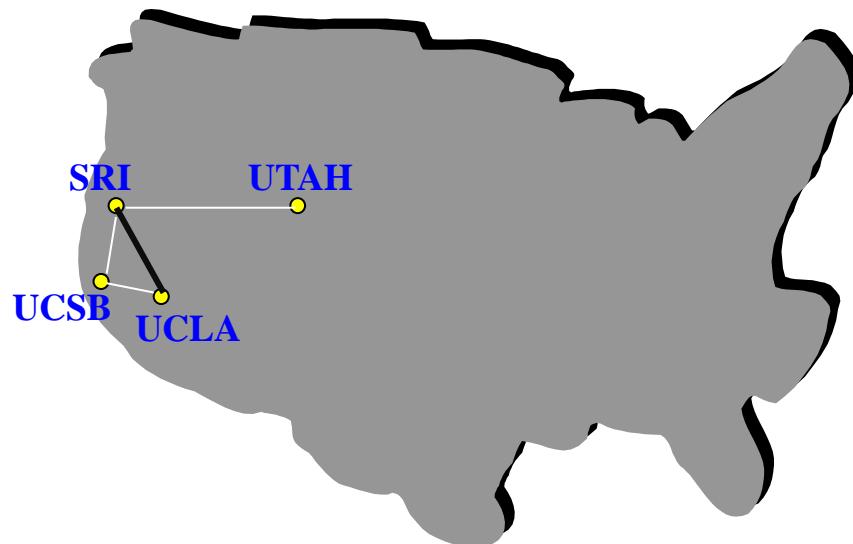
ARPA: Advanced Research Project Agency

UCLA: University California Los Angeles

SRI: Stanford Research Institute

IMP: Interface Message Processor

3 tháng sau, 12/1969



Một mạng hoàn chỉnh với 4 nút, 56kbps

UCSB: University of California, Santa Barbara

UTAH: University of Utah

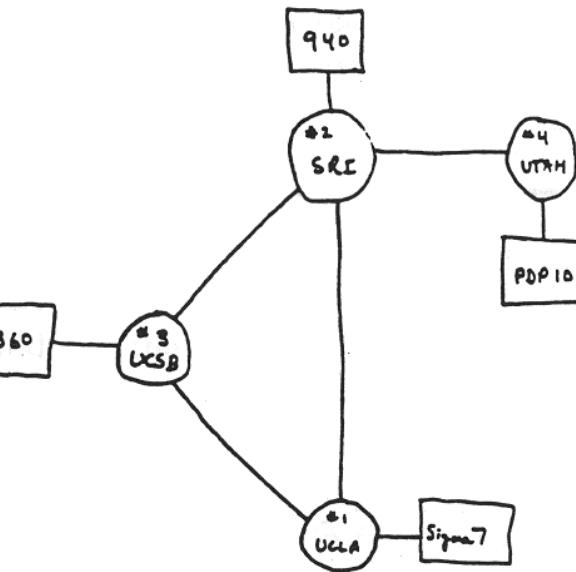
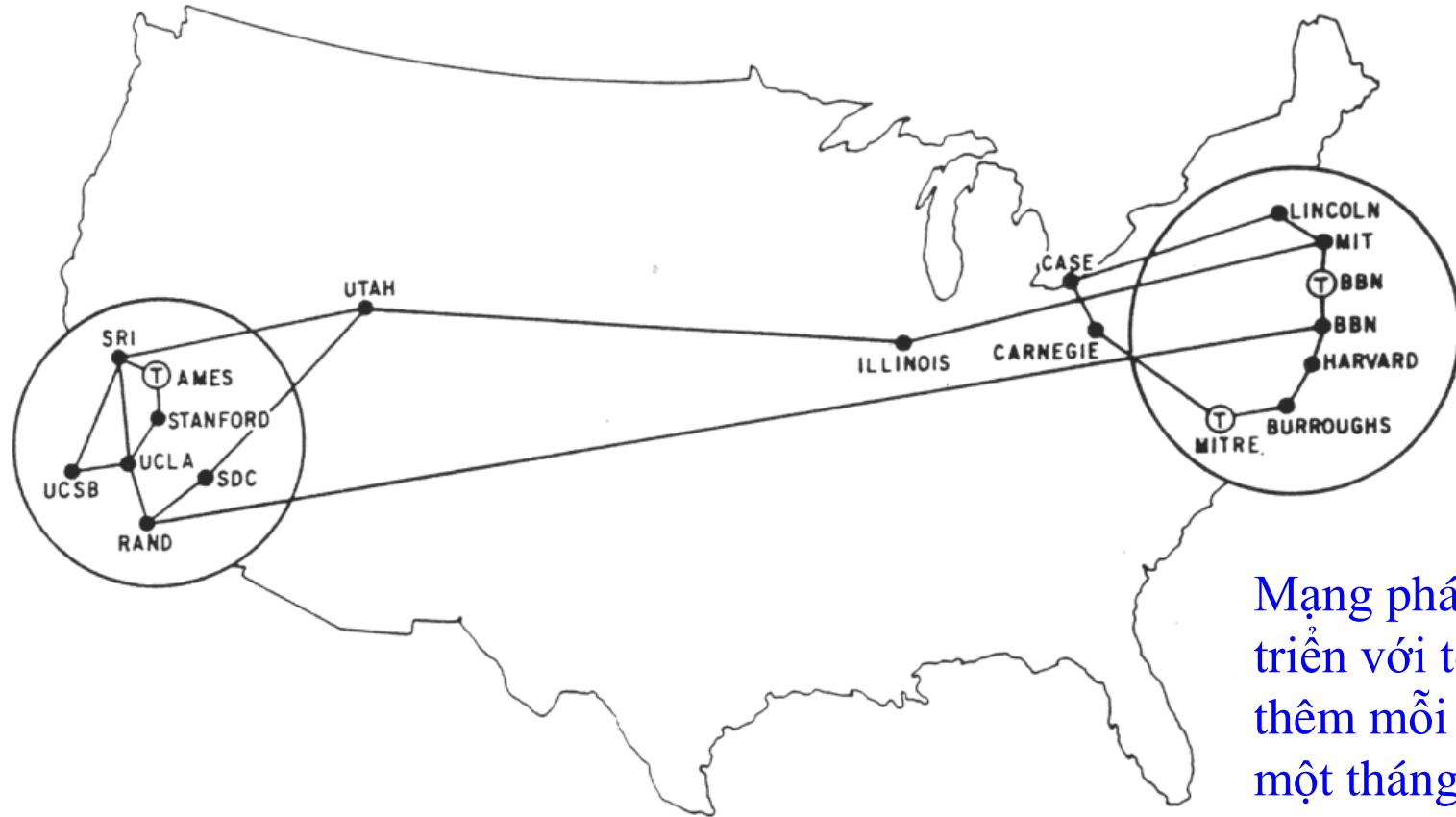


FIGURE 6.2 Drawing of 4 Node Network
(Courtesy of Alex McKenzie)

source: <http://www.cybergeography.org/atlas/historical.html>

ARPANET thời kỳ đầu, 1971



MAP 4 September 1971

Source: <http://www.cybergeography.org/atlas/historical.html>

ARPANET thập niên 70

- Kết nối liên mạng
- Kiến trúc mạng mới
- Kết nối các mạng riêng

Sự mở rộng của ARPANET, 1974

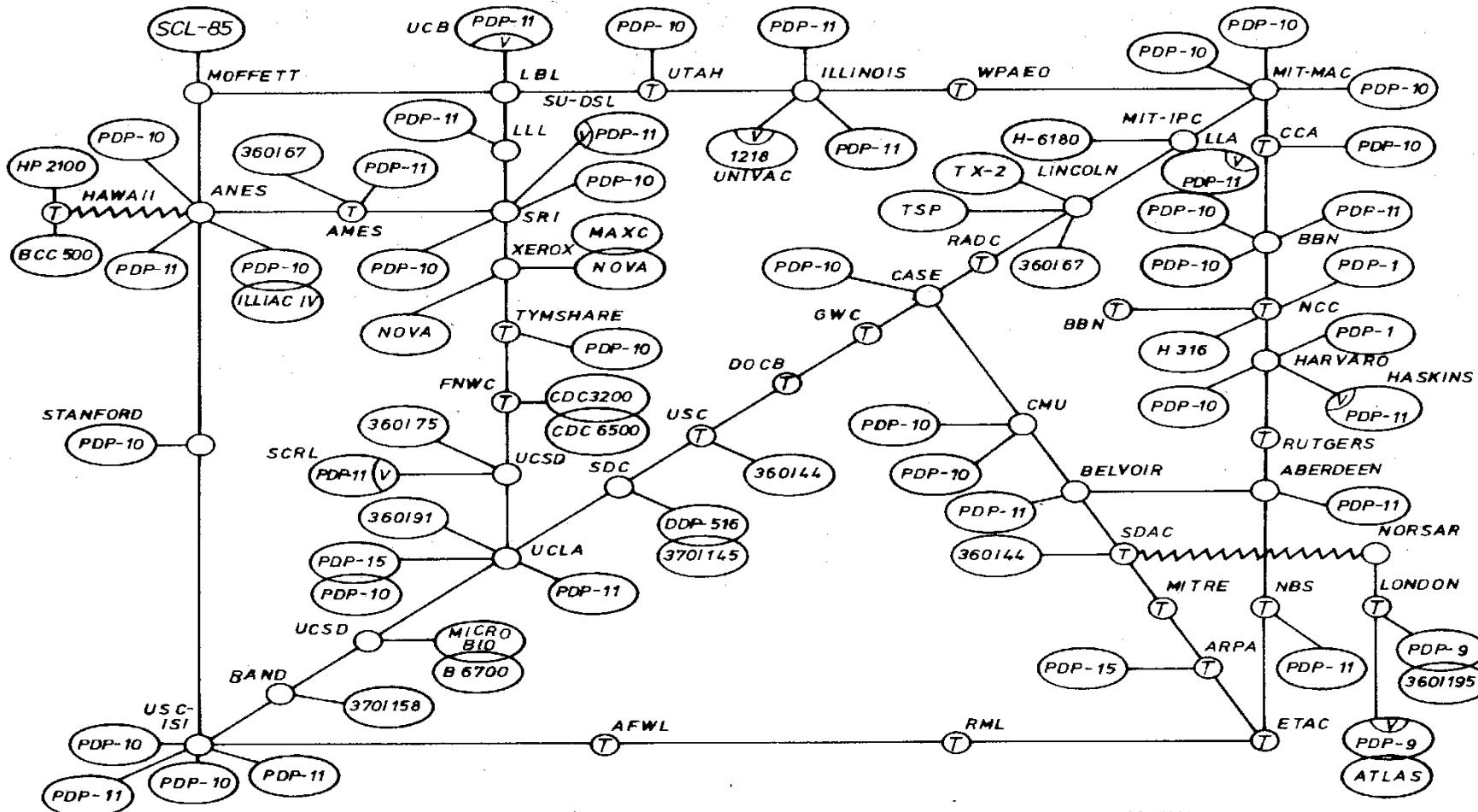
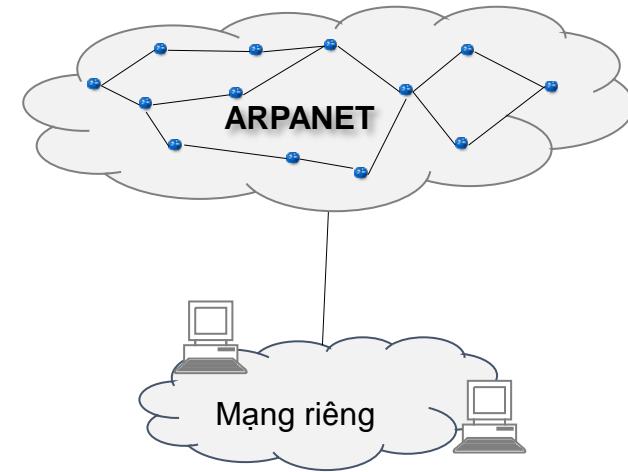


Abb. 4 ARPA NETwork, topologische Karte. Stand Juni 1974.

source:
[http://www.cybergeography.org/
atlas/historical.html](http://www.cybergeography.org/atlas/historical.html)

Kết nối liên mạng thập niên 70

- Đầu 1970 xuất hiện các mạng riêng:
 - ALOHAnet¹ tại Hawaii
 - DECnet²
 - IBM Systems Network Architecture (SNA)³
- kết nối mạng riêng với ARPANET?
- 1974: Mô hình kết nối các hệ thống mở
 - Cerf & Kahn nghiên cứu sự khác nhau giữa các mạng riêng & phương pháp kết nối chúng vào mạng ARPANET
 - Giao thức IP & TCP ra đời
- Turing Awards
- 1976: Ethernet⁴ by Xerox PARC, tiền thân của mạng cục bộ (LAN) sau này
- Cuối 1970: ATM



¹ <https://en.wikipedia.org/wiki/ALOHAnet>

² <https://en.wikipedia.org/wiki/DECnet>

³ https://en.wikipedia.org/wiki/Systems_Network_Architecture

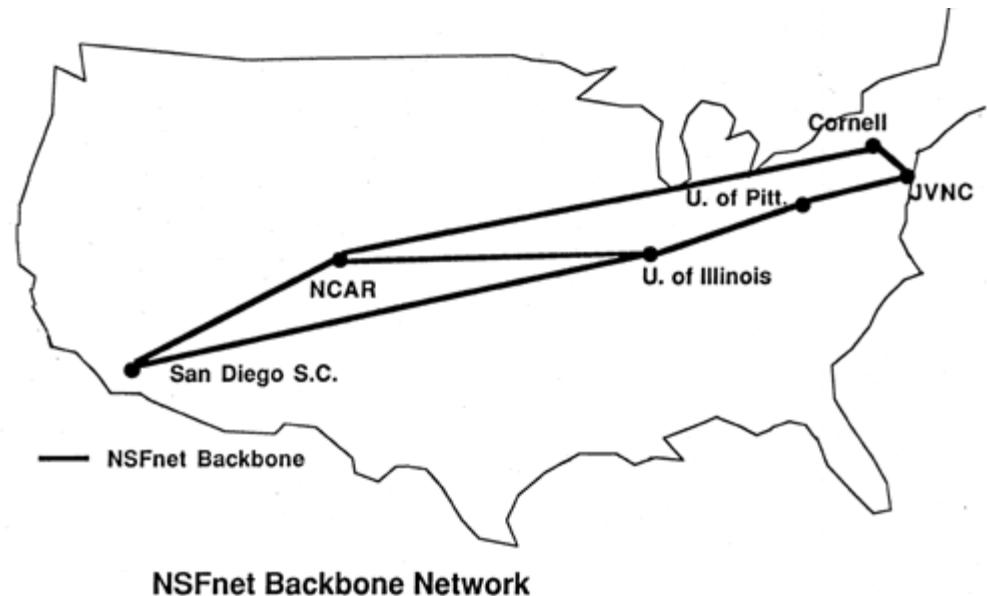
⁴ <https://en.wikipedia.org/wiki/Ethernet>

Internet thập niên 80

- Mạng NSFNET & thay thế sứ mệnh ARPANET
- Unix & mạng USENET
- USENET chuyển từ ARPANET sang NSFNET
- Các giao thức mới & kết nối mạng mới
- NSFNET → Internet backbone
- Chuẩn hóa Internet: IETF

Mạng NSFNET

- 1980: National Science Foundation¹ (NSF) thành lập các “supercomputing center” tại một số trường đại học
- 1986: NSF triển khai dự án kết nối các “supercomputing center” → NSFNET²
- Họ giao thức TCP/IP được sử dụng cho NSFNET
- Các đường truyền backbone dần được hình thành và nâng cấp: 56Kbps ban đầu, T1: 1.5Mbps (1988), T3 – 45Mbps (1991)
- 1990: ARPANET (Bộ quốc phòng Mỹ) kết thúc sứ mệnh, chuyển vai trò kết nối các mạng riêng cho NSFNET



Các “supercomputing center” kết nối vào NSFNET:

- JVNC: John von Neumann Center at Princeton University
- Cornell Theory Center at Cornell University
- San Diego Supercomputer Center (SDSC) University of California
- National Center for Supercomputing Applications (NCSA), University of Illinois
- Pittsburgh Supercomputing Center (PSC), a joint effort of Carnegie Mellon University, the University of Pittsburgh, and Westinghouse

¹ https://en.wikipedia.org/wiki/National_Science_Foundation

² https://en.wikipedia.org/wiki/National_Science_Foundation_Network

Unix & mạng USENET

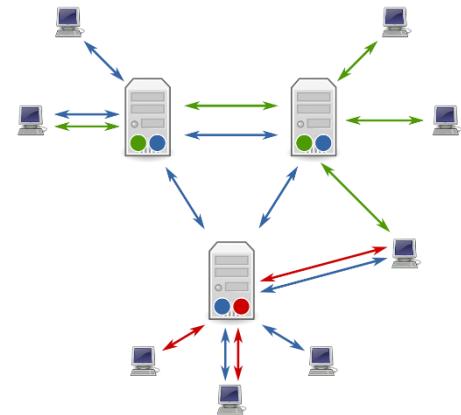
- Unix:

- Hệ điều hành máy tính lâu đời nhất vẫn phát triển đến nay
- Ra đời năm 1970 tại phòng thí nghiệm Bell¹
- Phát triển theo rất nhiều dòng sản phẩm, cài đặt trên các loại máy tính lớn mainframe, mini và cả các dòng máy cá nhân
- Linux: hệ điều hành “clone” từ Unix

- USENET:

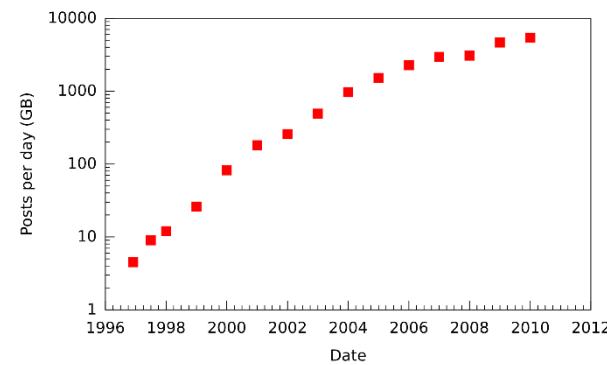
- Ra đời năm 1980 và vẫn được sử dụng đến gần đây
- Kết nối các máy tính chạy hệ điều hành Unix
- Sử dụng ARPANET làm đường truyền (sau này dùng Internet)
- Unix-to-Unix Copy (UUCP) network architecture
- Hướng đến unix users với các dịch vụ unix như email, file transfer, telnet
- Newsgroup: dịch vụ cung cấp thông tin rất phổ biến (trước khi có Web)

Usenet servers and clients:



Source: Wikipedia

Usenet Traffic Per Day:



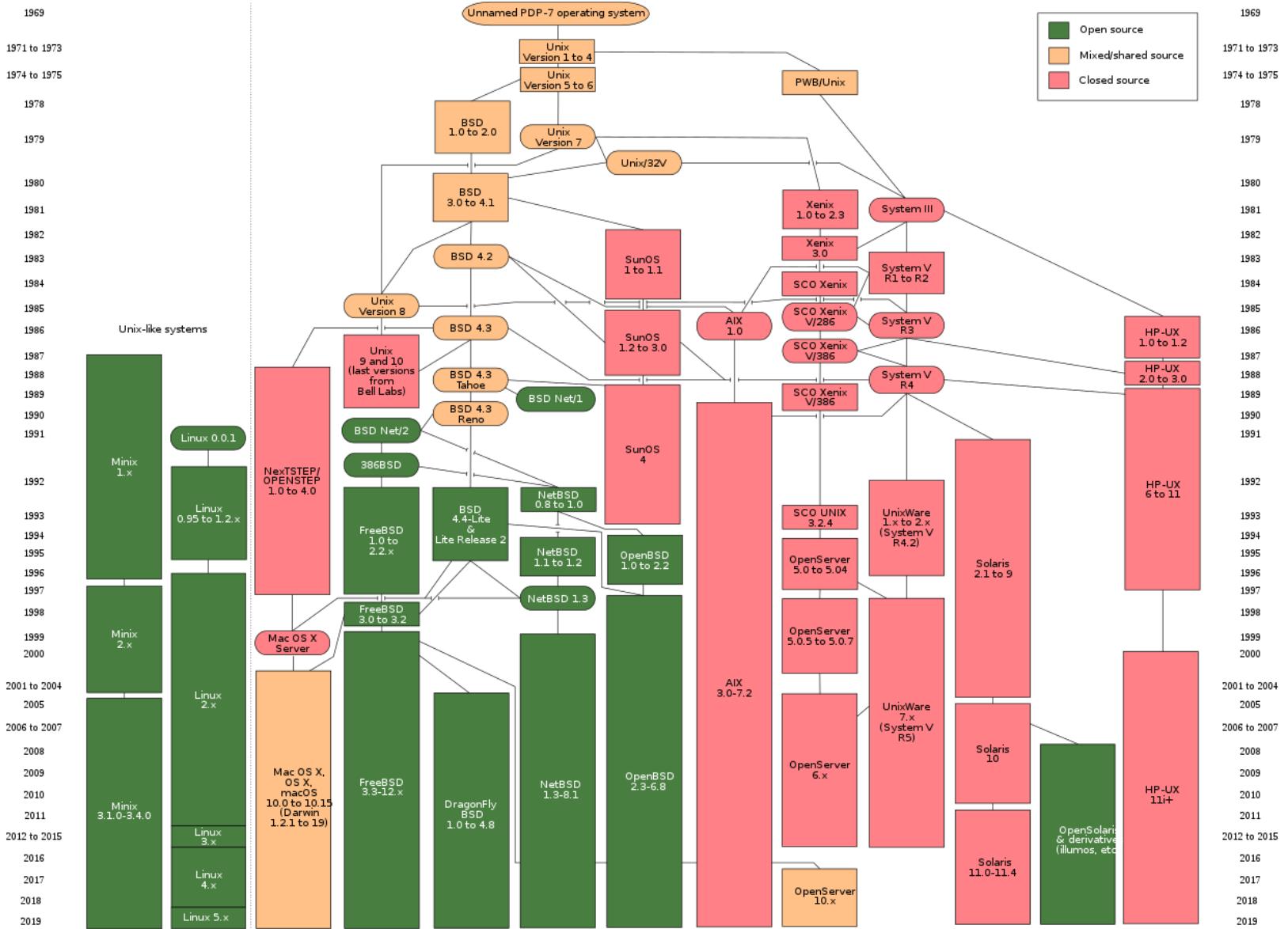
Source: Wikipedia

¹ https://en.wikipedia.org/wiki/Bell_Labs

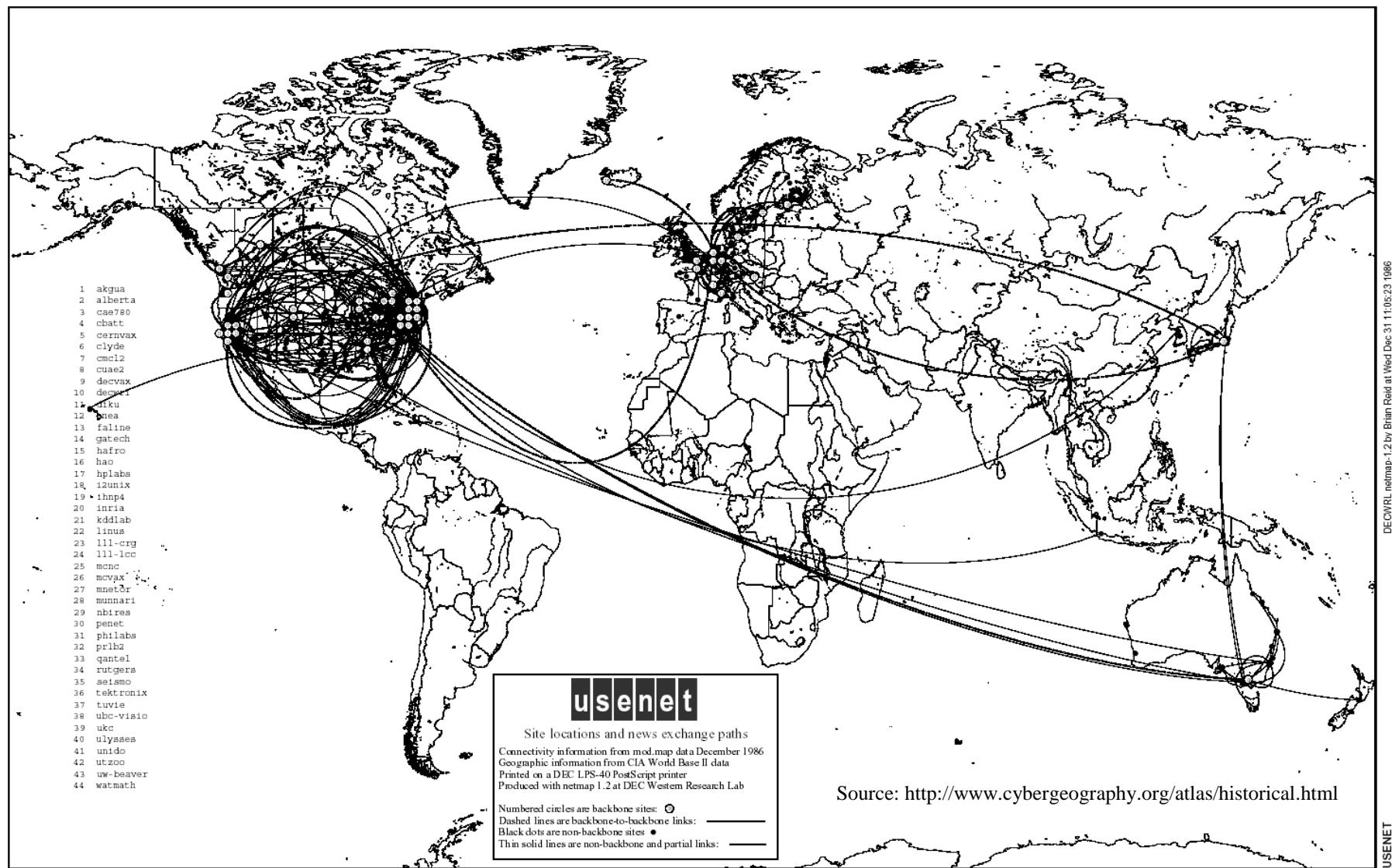
² https://en.wikipedia.org/wiki/National_Science_Foundation_Network

Phân hạch Unix

Source: Wikipedia
<https://en.wikipedia.org/wiki/Unix>



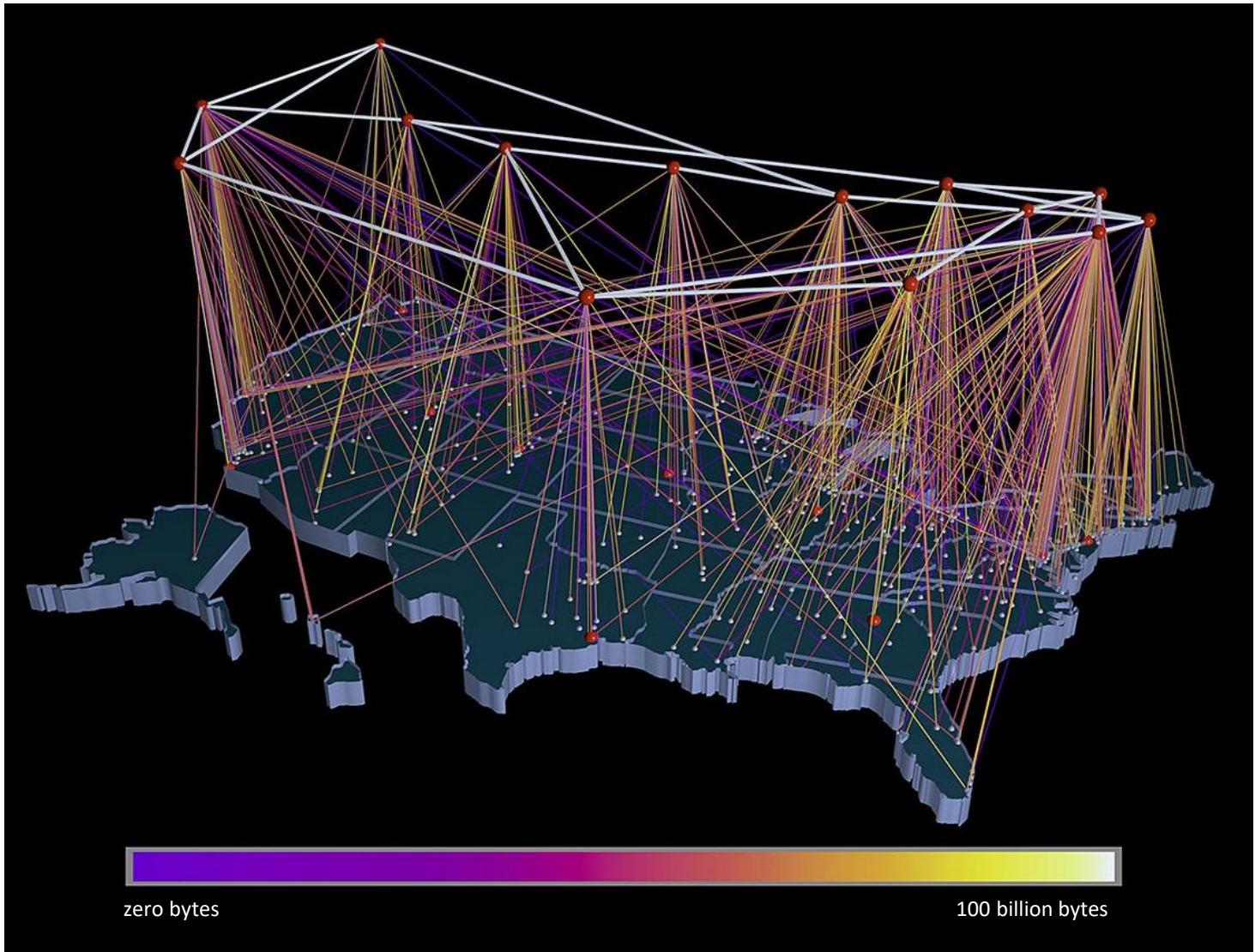
1986: Nối kết USENET & NSFNET



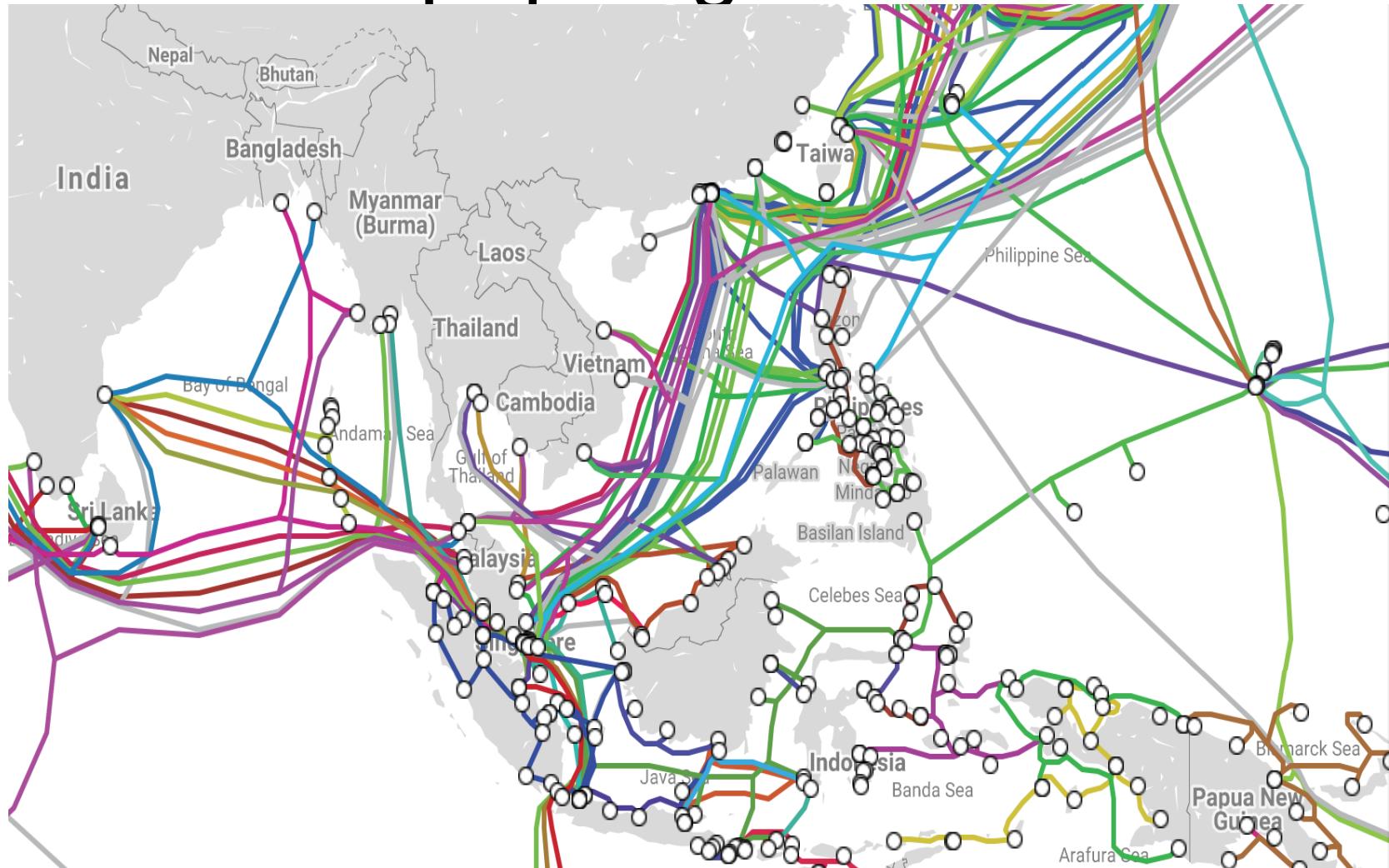
Giao thức mới, mạng mới & chuẩn hóa

- Bộ giao thức mạng TCP/IP:
 - “Ngôn ngữ” của ARPANET & NFSNET
 - 1985: tích hợp vào Unix của IBM, AT&T
 - 1989: Berkeley tích hợp TCP/IP vào FreeBSD Unix
 - Dịch vụ trên TCP/IP xuất hiện bên cạnh Unix Newsgroups: FTP, DNS, Mail, v.v..
 - 1995: Microsoft tích hợp TCP/IP vào hệ điều hành Windows
- Internet Engineering Task Force (IETF) & Internet:
 - 1986 – 1988: các mạng riêng mới kết nối vào NSFNET (MFENET, HEPNET - Dept. Energy, SPAN - NASA, BITnet, CSnet, NSFnet, Minitel, v.v..)
 - NSFNET trở thành mạng backbone kết nối các mạng khác với nhau → thuật ngữ Internet được hình thành
 - 1986: cuộc họp đầu tiên theo hình thức mở cửa công cộng (public open) với các thành viên tham gia tự nguyện, tự đặt sứ mệnh chuẩn hóa TCP/IP
 - 1987: Ed Krol, giám đốc trung tâm kết nối mạng trường đại học Illinois, mệt mỏi vì phải lặp lại việc trả lời các câu hỏi liên quan đến Internet, đã cho xuất bản tài liệu text online có tên “*Hitchhiker's Guide to the Internet*” (hitchhiker nghĩa là người đi nhò xe). Hai năm sau, tài liệu được IETF xuất bản với mã số RFC1118, trở thành bản hướng dẫn “Internet manual” đầu tiên.

Kết nối Internet năm 1991

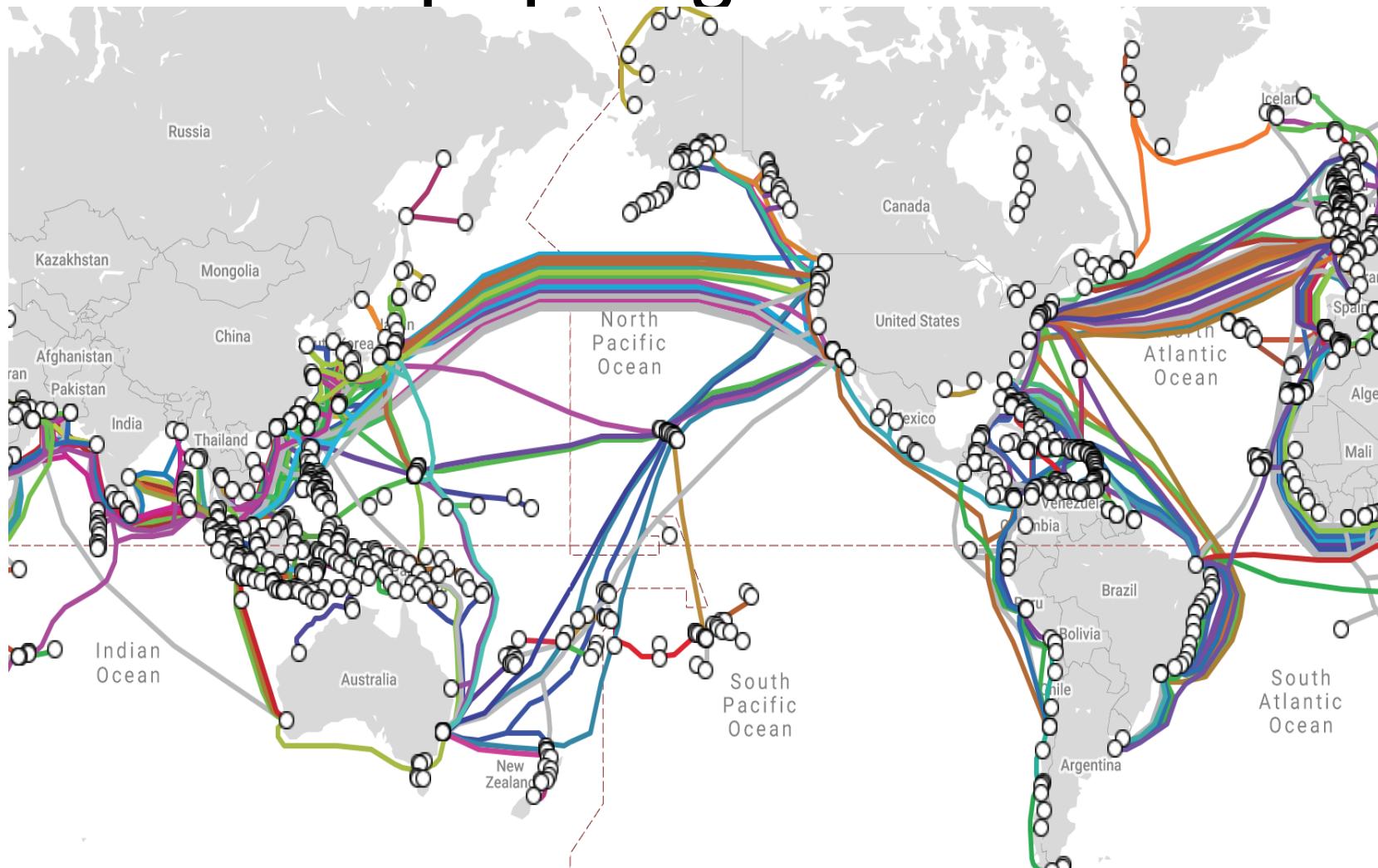


Kết nối cáp quang dưới biển



Nguồn: <https://www.submarinecablemap.com>

Kết nối cáp quang dưới biển



Nguồn: <https://www.submarinecablemap.com>

Internet thập niên 90

- Thương mại hóa Internet
- Web & Web & Web

Internet thập niên 90

- Đầu 90: ARPAnet chỉ là một phần của Internet
- Đầu 90: Web
 - HTML, HTTP: Berners-Lee
 - 1994: Mosaic, Netscape
- Cuối 90: Thương mại hóa Internet

Cuối 1990's – 2000's:

- Nhiều ứng dụng mới: chat, chia sẻ file P2P...
- E-commerce, Yahoo, Amazon, Google...
- > 50 triệu máy trạm, > 100 triệu NSD
- Vấn đề an toàn an ninh thông tin!
 - Internet dành cho tất cả mọi người
 - Tất cả các dịch vụ phải quan tâm tới vấn đề này

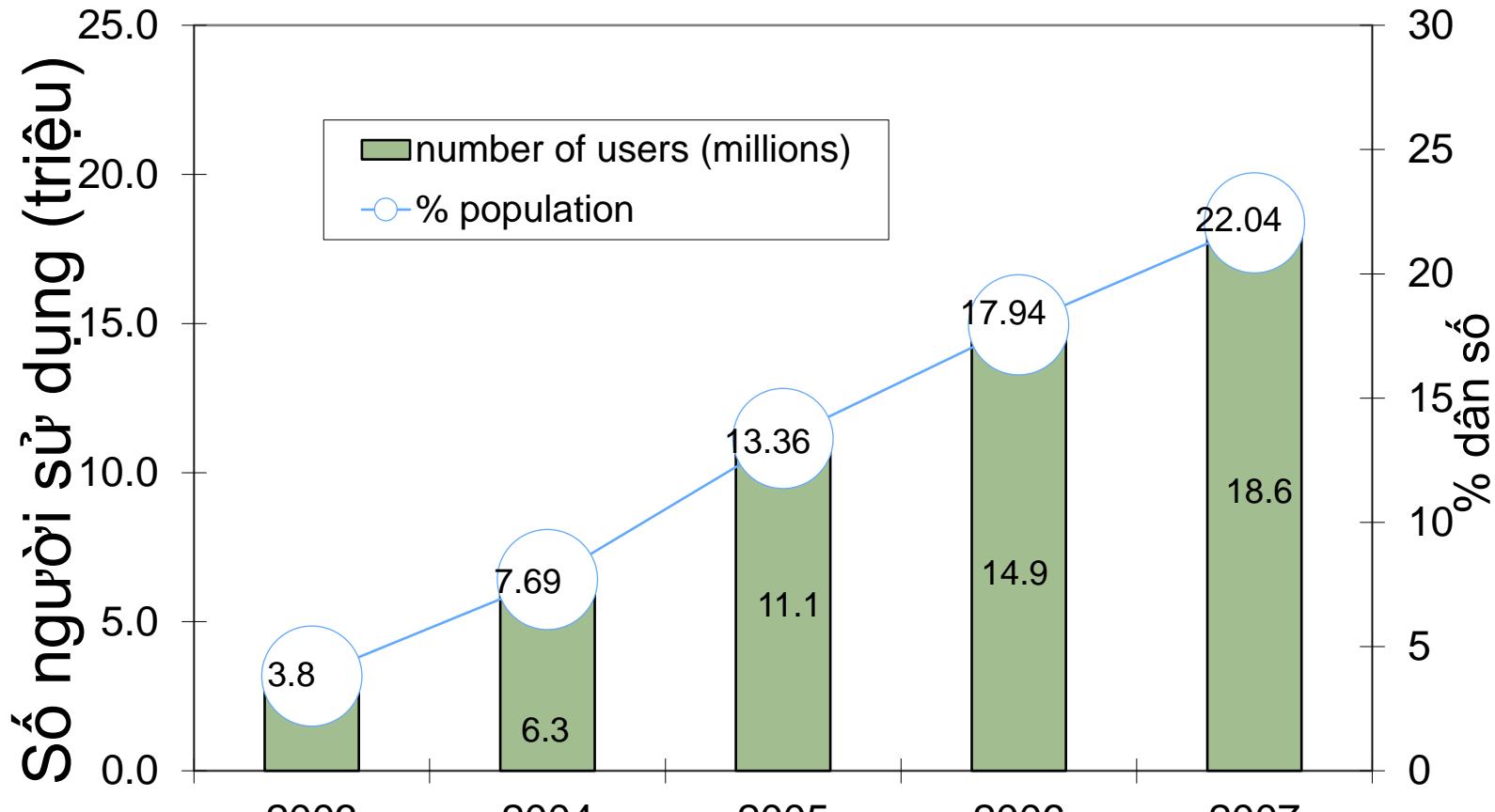
Internet Việt Nam

- 1991: Kết nối đầu tiên¹
- 1996: Chuẩn bị hạ tầng kết nối Internet
 - ISP: VNPT
 - 64kbps, 1 đường kết nối quốc tế
- 1997: Việt Nam chính thức kết nối Internet
 - 1 IXP: VNPT
 - 4 ISP: VNPT, Netnam (IoIT), FPT, SPT
- Quản lý Internet tại Việt Nam:
 - Bộ Thông tin & Truyền thông với Trung tam Internet VN² (VNNIC)
 - Các ISP: VNPT, FPT, Viettel, CMC, VDC, Netnam, v.v..

¹ https://vi.wikipedia.org/wiki/Internet_t%E1%BA%A1i_Vi%E1%BB%87t_Nam

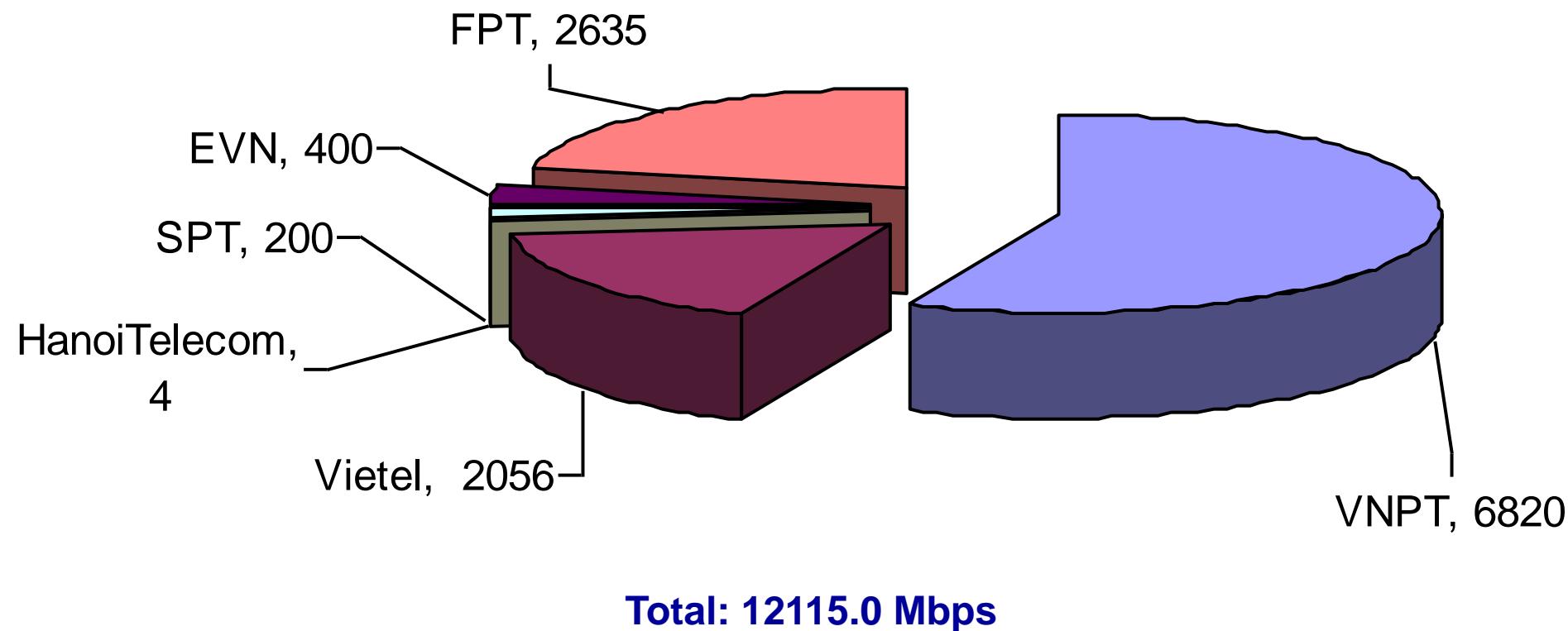
² https://vi.wikipedia.org/wiki/Trung_t%C3%A2m_Internet_Vi%E1%BB%87t_Nam

Thống kê đến năm 2007



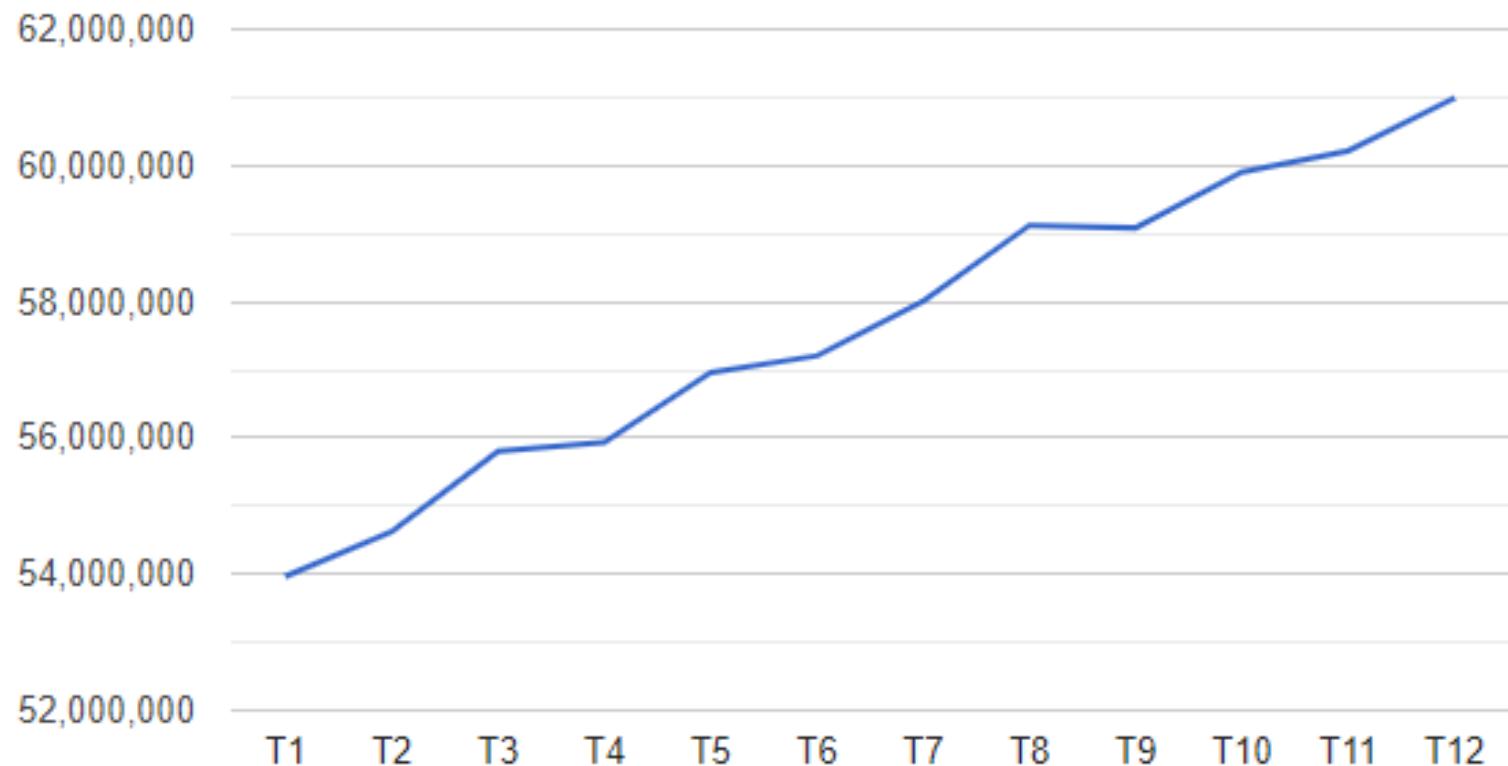
Nguồn: Vnnic, <http://www.thongkeinternet.vn>

Bảng thông (Mbps) đến quý 3 - 2007



Thống kê gần đây

Số người sử dụng di động năm 2019



Nguồn: Vnnic, <http://www.thongkeinternet.vn>

Số thuê bao Internet, 2019

Tình hình phát triển thuê bao băng rộng cố định tháng 5/2019

Số thuê bao truy nhập Internet qua hình thức xDSL:	182,853
Số thuê bao truy nhập Internet qua kênh thuê riêng:	22,929
Số thuê bao truy nhập Internet qua hệ thống cáp truyền hình (CATV):	868,039
Số thuê bao truy nhập Internet qua hệ thống cáp quang tới nhà thuê bao (FTTH):	12,606,506
Tổng số thuê bao băng rộng cố định:	13,680,327

Cung cấp bởi cục Viễn thông, Bộ thông tin và truyền thông
<http://vnta.gov.vn/thongke/Trang/dulieuthongke.aspx>

Số thuê bao di động năm 2019

Tình hình phát triển thuê bao điện thoại di động tháng 5/2019

‣ Tổng số thuê bao điện thoại di động có phát sinh lưu lượng:	133,877,535
‣ Tổng số thuê bao điện thoại di động đang hoạt động chỉ sử dụng thoại, tin nhắn:	75,216,569
▪ Thuê bao trả trước:	70,448,710
▪ Thuê bao trả sau:	4,767,859
‣ Tổng số thuê bao điện thoại di động đang hoạt động có sử dụng dữ liệu:	58,660,966
▪ Thuê bao trả trước:	54,158,129
▪ Thuê bao trả sau:	4,502,837

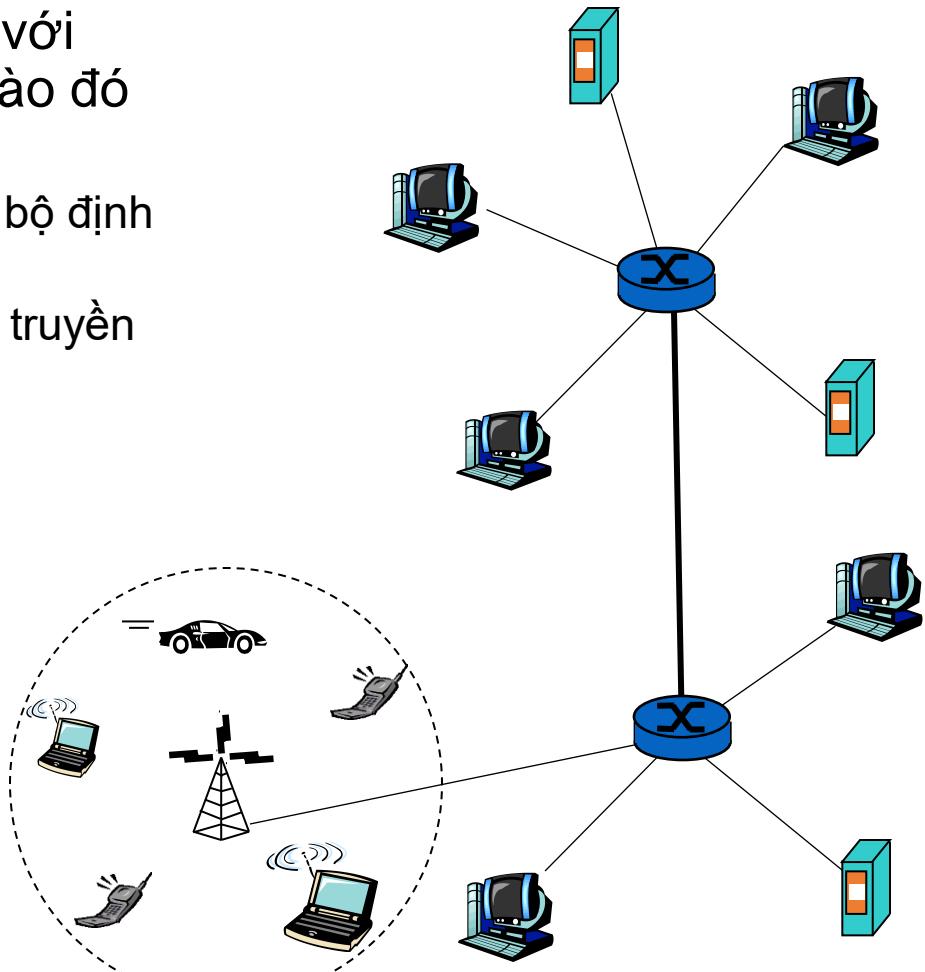
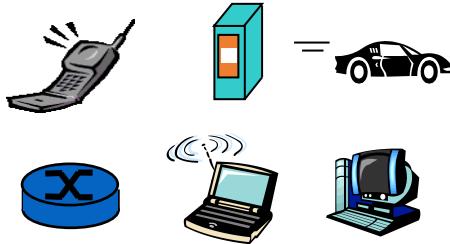
Cung cấp bởi cục Viễn thông, Bộ thông tin và truyền thông
<http://vnta.gov.vn/thongke/Trang/dulieuthongke.aspx>

1. Cơ bản về mạng máy tính

- Internet & lịch sử phát triển
- Các khái niệm mạng máy tính
- Kiến trúc mạng

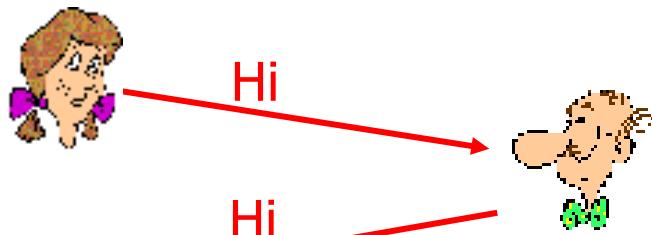
Mạng máy tính là gì?

- Tập hợp các máy tính kết nối với nhau dựa trên một kiến trúc nào đó để có thể trao đổi dữ liệu
 - Máy tính: máy trạm, máy chủ, bộ định tuyến
 - Kết nối bằng một phương tiện truyền
 - Theo một kiến trúc mạng
- Các dạng “máy tính” (trạm làm việc):



Giao thức là gì?

Giao thức người-người



Hi
Hi
Anh cho hỏi
mấy giờ rồi ạ?
→
← 2:00

Thời gian
↓

Giao thức máy-máy



Giao thức mạng

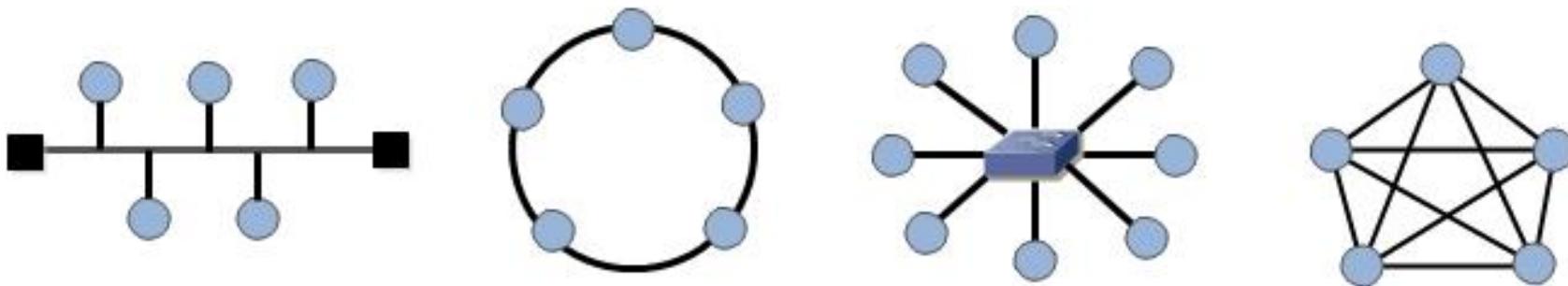
- Protocol: Quy tắc để truyền thông
 - Gửi một yêu cầu hoặc thông tin
 - Nhận một thông tin hoặc yêu cầu hành động
 - Các yêu cầu, thông tin được gửi dưới dạng các thông điệp
- Định nghĩa
 - Khuôn dạng dữ liệu, thông điệp
 - Thứ tự truyền, nhận thông điệp giữa các thực thể trên mạng
 - Các hành động tương ứng khi nhận được thông điệp
- Ví dụ về giao thức mạng: TCP, UDP, IP, HTTP, Telnet, SSH, Ethernet, ...

Đường truyền vật lý

- Là các phương tiện vật lý có khả năng truyền dẫn tín hiệu
- Phân loại:
 - Hữu tuyến: cáp xoắn, cáp đồng trục, cáp quang,...
 - Vô tuyến: sóng radio, viba, sóng hồng ngoại,...
- Một số thông số đặc trưng:
 - Băng tần: Độ rộng tần số tín hiệu có thể truyền đi
 - ❖ f_{\min} : tần số nhỏ nhất, f_{\max} : tần số lớn nhất
 - ❖ $Băng tần = f_{\max} - f_{\min}$
 - Tỉ lệ lỗi bit khi truyền (BER – Bit Error Rate/Ratio)
 - Độ suy hao: mức suy giảm tín hiệu khi truyền

Kiến trúc mạng

- Các nút mạng kết nối với nhau như thế nào? (Hình trạng – Topology)
 - Topology vật lý: hình trạng dựa trên cáp kết nối



Bus

Ring (Vòng)

Star (Sao)

Mesh (Lưới)

- Topology logic: hình trạng dựa trên cách thức truyền tín hiệu: điểm-điểm, điểm-đa điểm (quảng bá)
- ...và trao đổi dữ liệu với nhau như thế nào? (Giao thức – Protocol)

Một vài ví dụ

- Mạng Internet
- Mạng nội bộ cơ quan, trường học
- Mạng gia đình
- Hệ thống ATM của ngân hàng
- Mạng điện thoại
- ...

Phân loại mạng máy tính

- Mạng cá nhân (PAN – Personal Area Network)
 - Phạm vi kết nối: vài chục mét
 - Số lượng người dùng: một vài người dùng
 - Thường phục vụ cho cá nhân
- Mạng cục bộ (LAN – Local Area Network):
 - Phạm vi kết nối: vài ki-lô-mét
 - Số lượng người dùng: một vài đến hàng trăm nghìn
 - Thường phục vụ cho cá nhân, hộ gia đình, tổ chức

Phân loại mạng máy tính

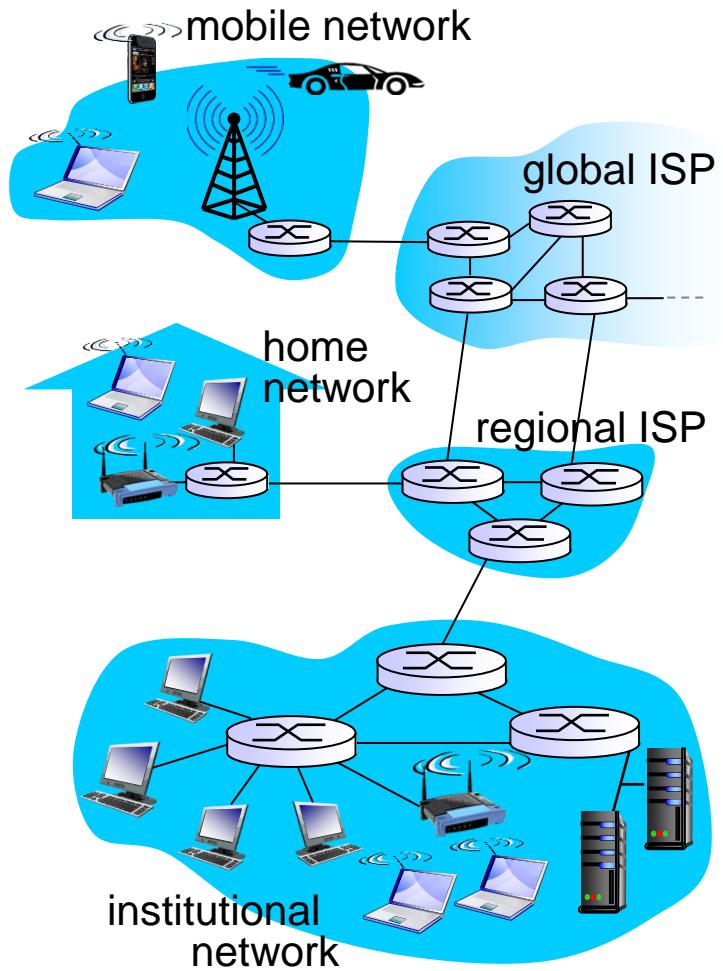
- Mạng đô thị (MAN – Metropolitan Area Network)
 - Phạm vi kết nối: hàng trăm ki-lô-mét
 - Số lượng người dùng: hàng triệu
 - Phục vụ cho thành phố, khu vực
- Mạng diện rộng (WAN – Wide Area Network)
 - Phạm vi kết nối: vài nghìn ki-lô-mét
 - Số lượng người dùng: hàng tỉ
 - GAN – Global Area Network: phạm vi toàn cầu (Ví dụ: Internet)

1. Cơ bản về mạng máy tính

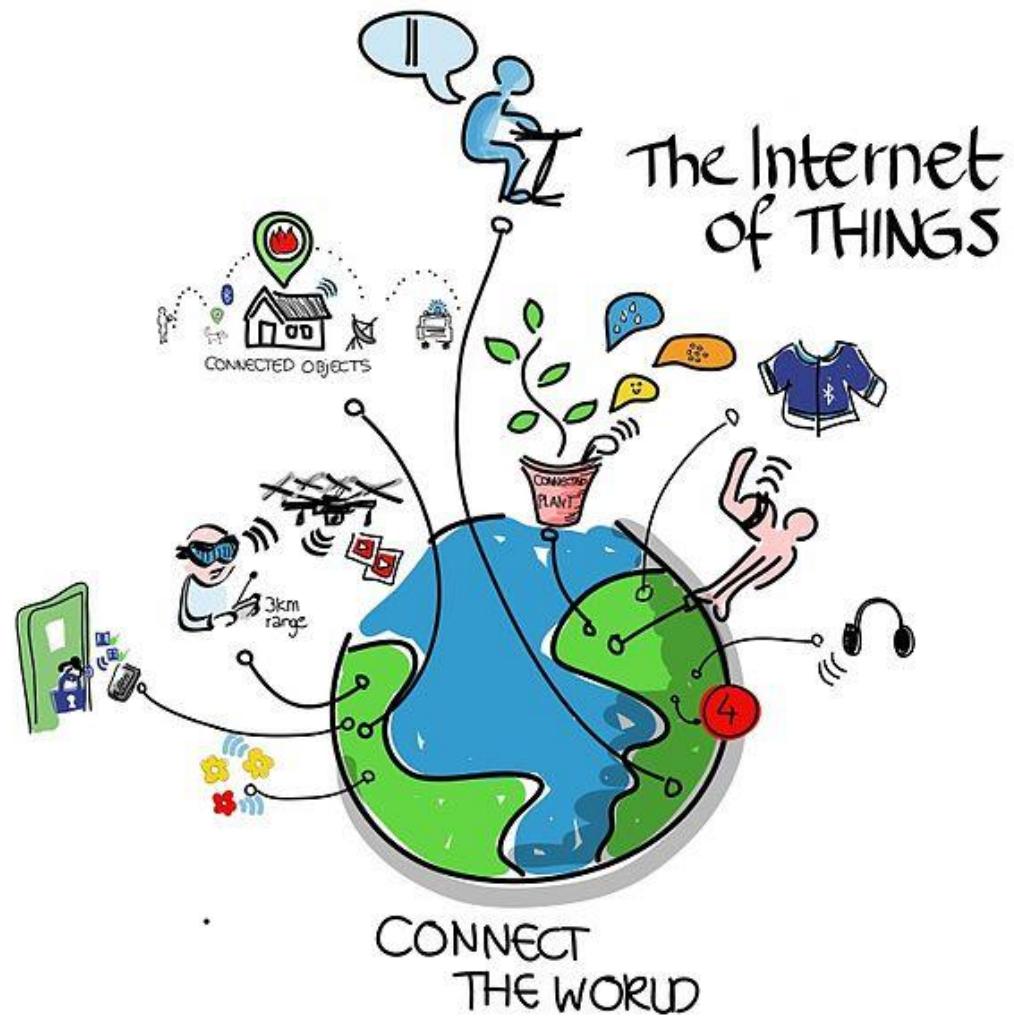
- Internet & lịch sử phát triển
- Các khái niệm mạng máy tính
- Kiến trúc mạng

Mạng Internet

- Trên 5 tỉ thiết bị kết nối
- 3.2 tỉ người dùng(40%)
- Đường truyền: cáp quang, cáp đồng, Wimax, 3G...
- Truyền tải $\sim 3 \times 10^9$ GB mỗi ngày
- Dịch vụ: Web, email, mạng xã hội, Skype...

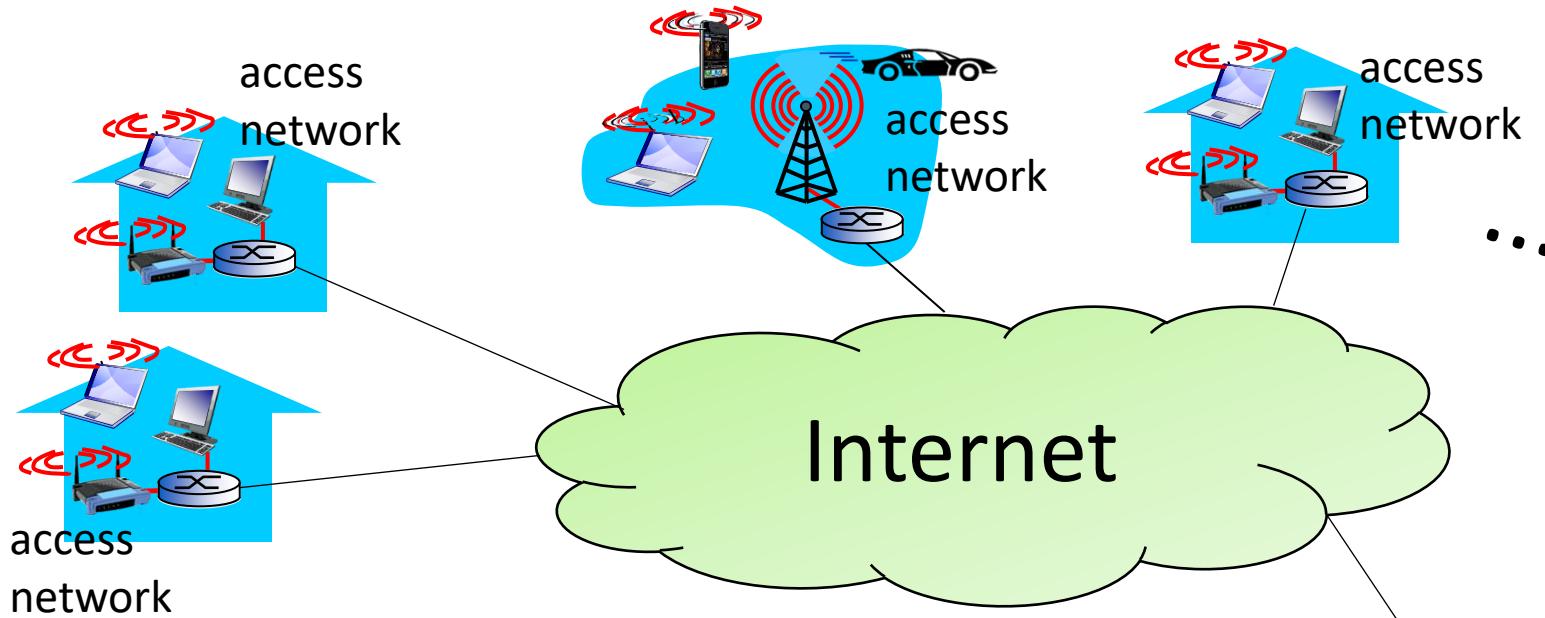


Internet of things

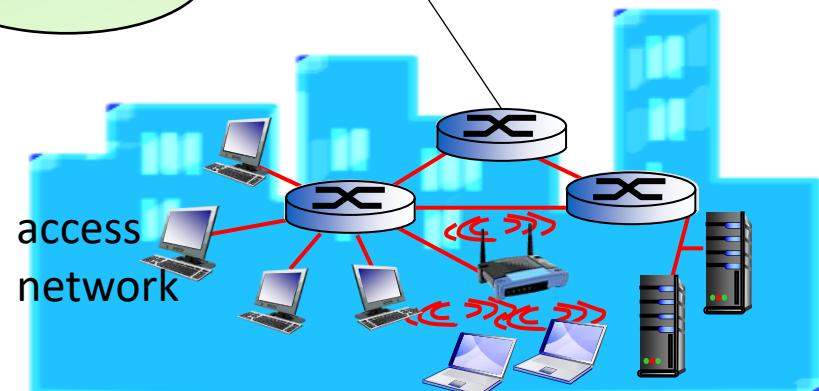


Kiến trúc Internet

- Mạng của các mạng (Network of networks)

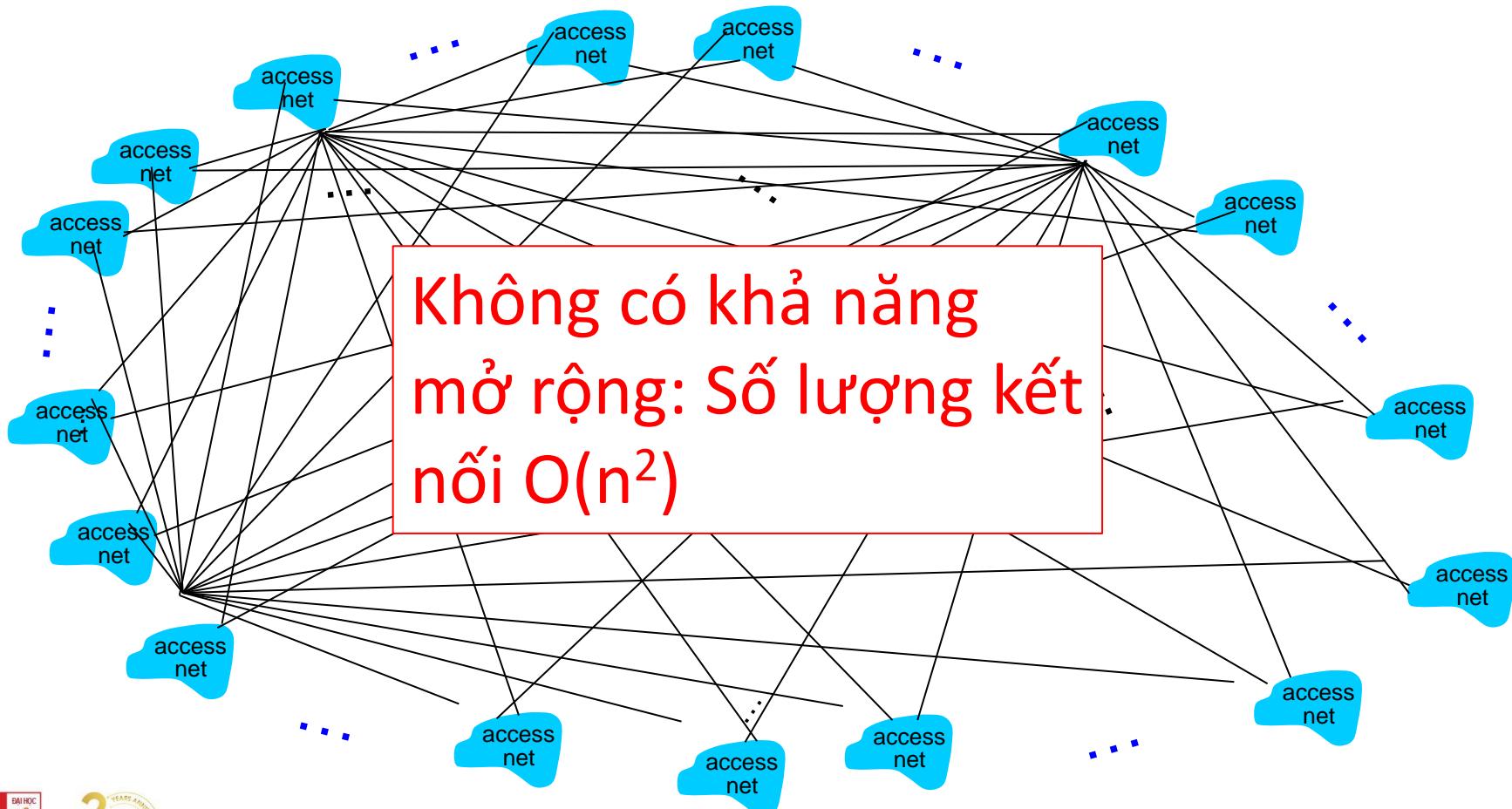


Làm thế nào để kết nối hàng
triệu hệ thống mạng với nhau?



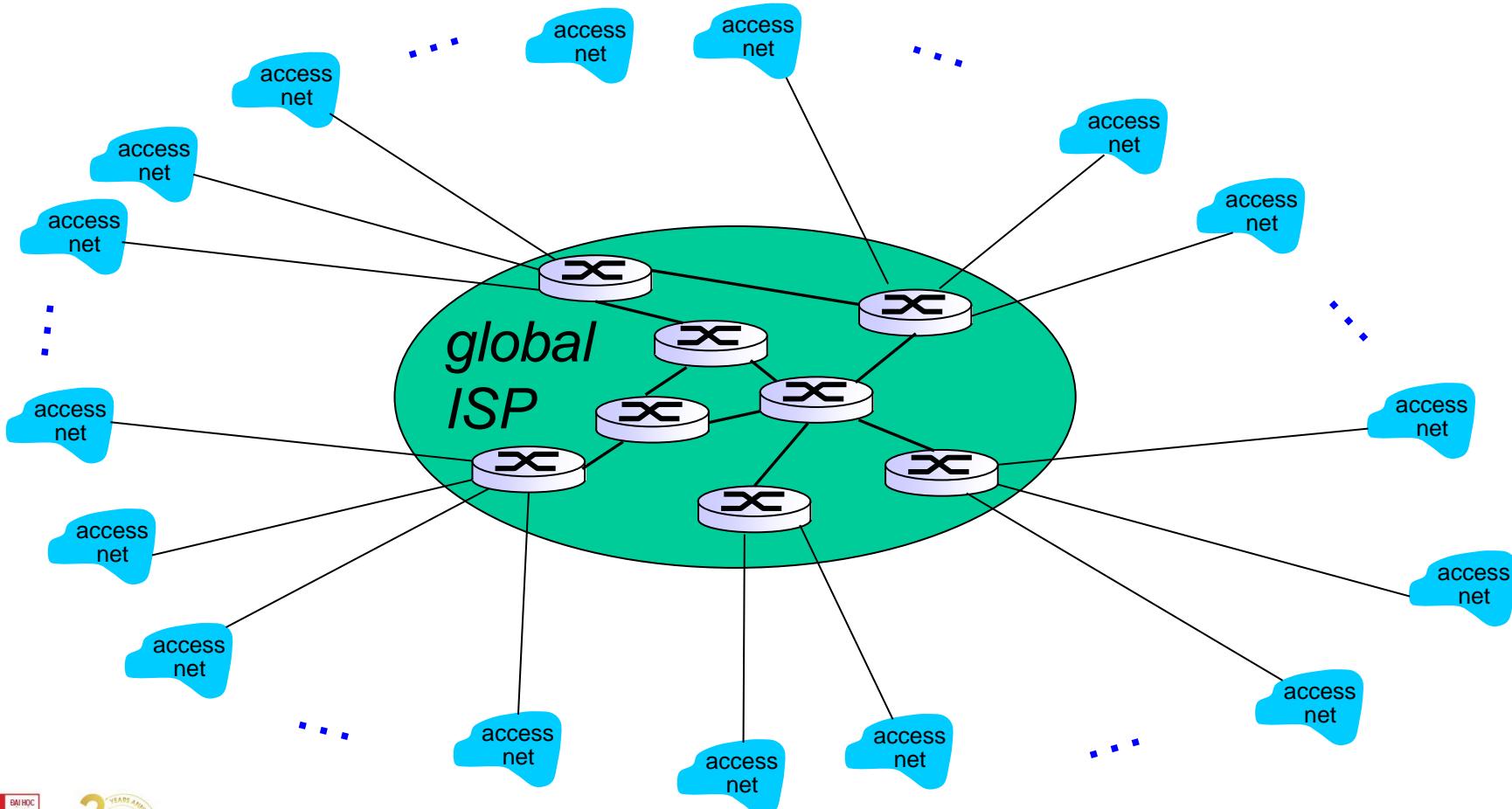
Kiến trúc Internet: Mạng của các mạng

- Kết nối một mạng với tất cả các mạng khác?



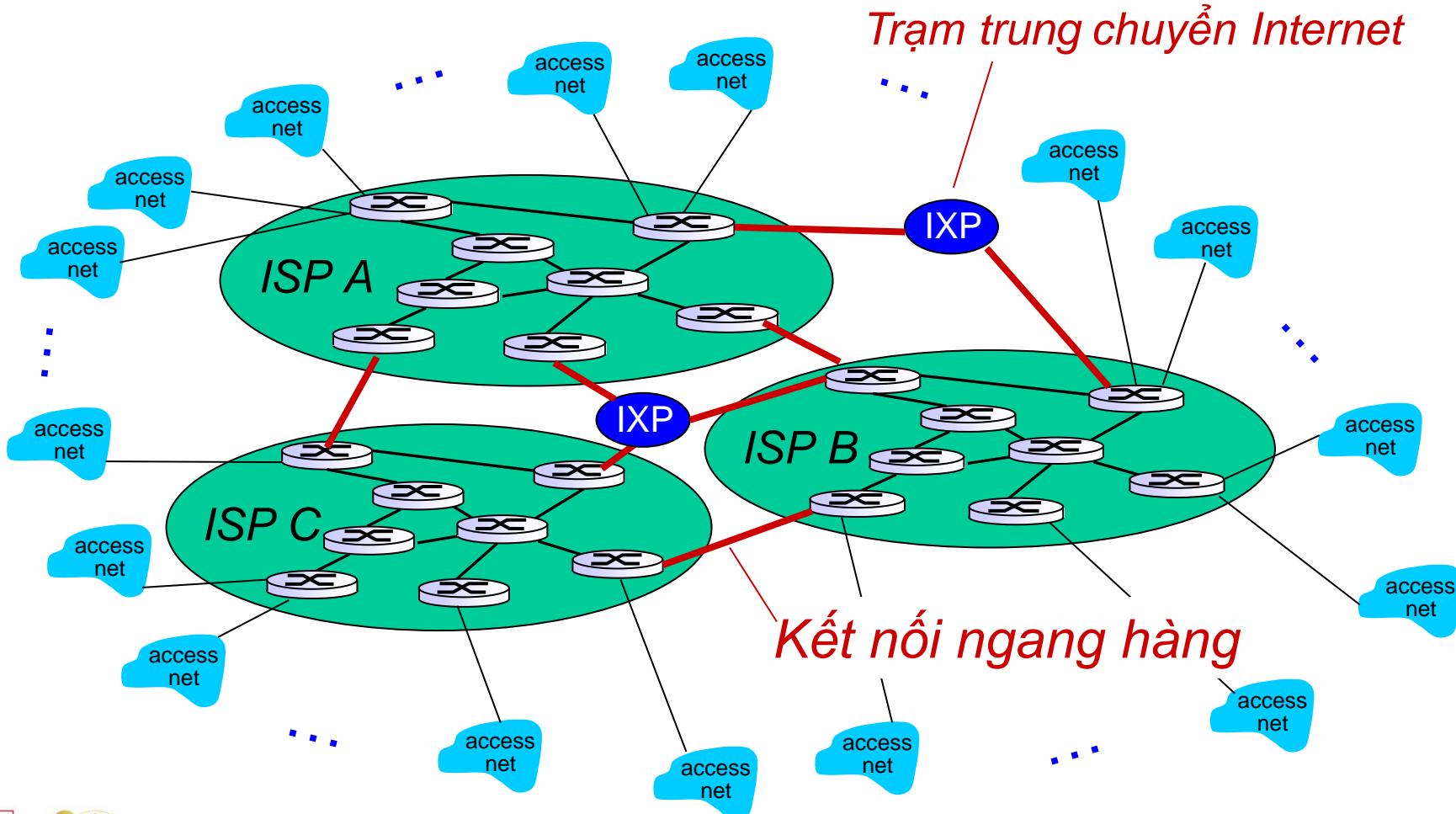
Kiến trúc Internet: Mạng của các mạng (2)

- Kết nối mỗi mạng vào một trạm chuyển tiếp của một nhà cung cấp toàn cầu (global ISP)*



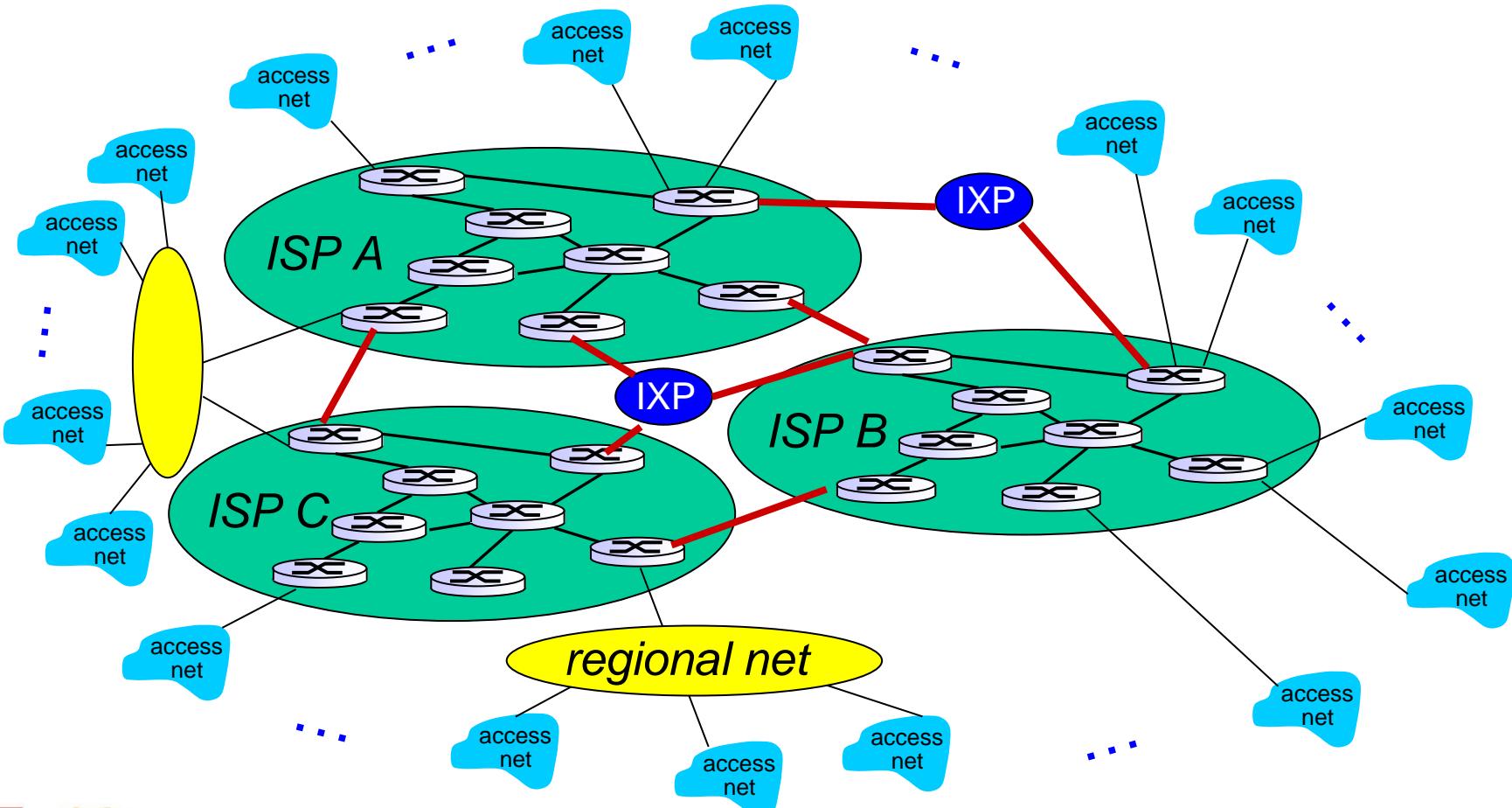
Kiến trúc Internet: Mạng của các mạng (3)

- Thêm nhiều ISP...



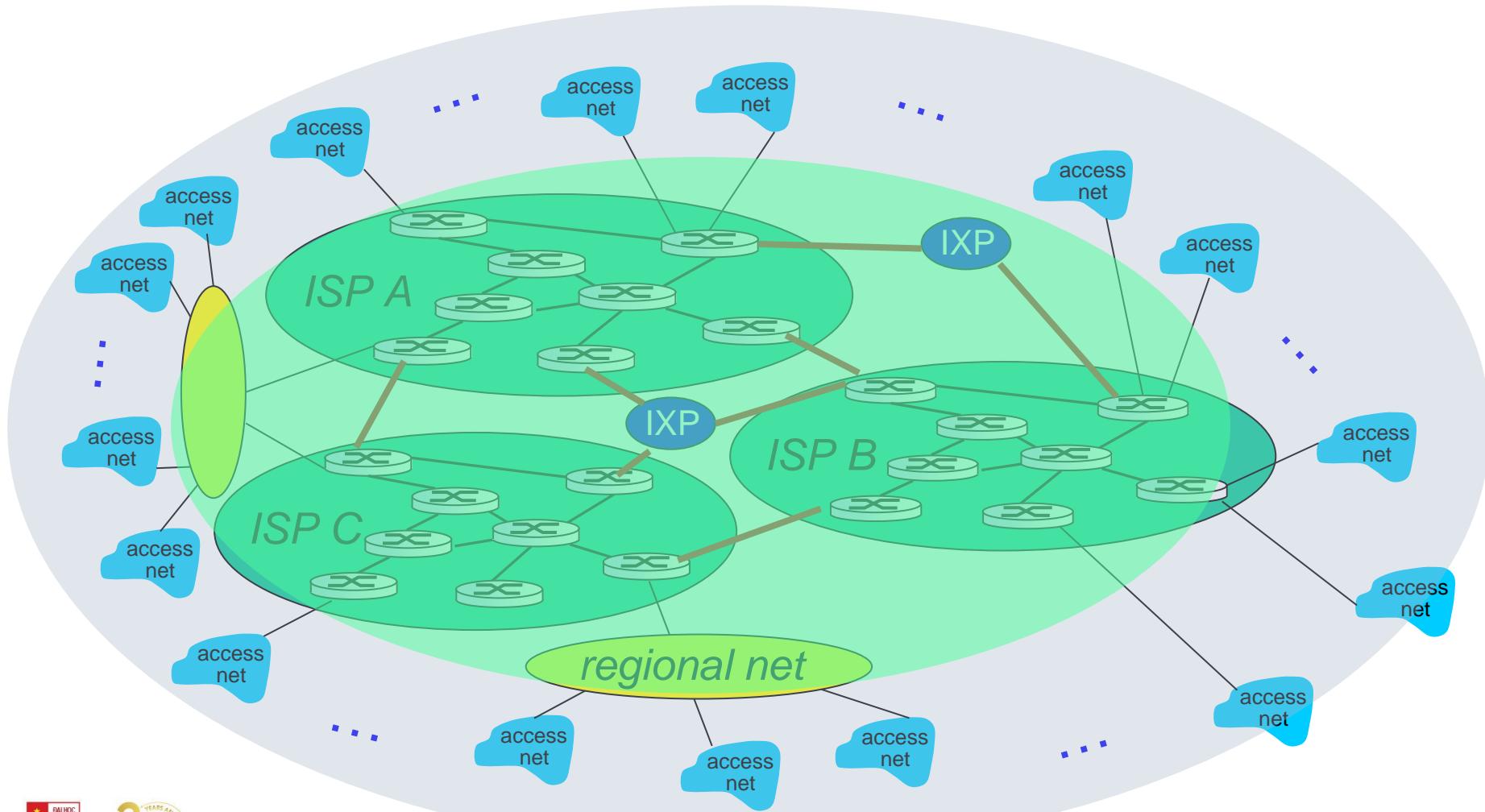
Kiến trúc Internet: Mạng của các mạng (4)

- Thêm các mạng khu vực (*regional network*)...



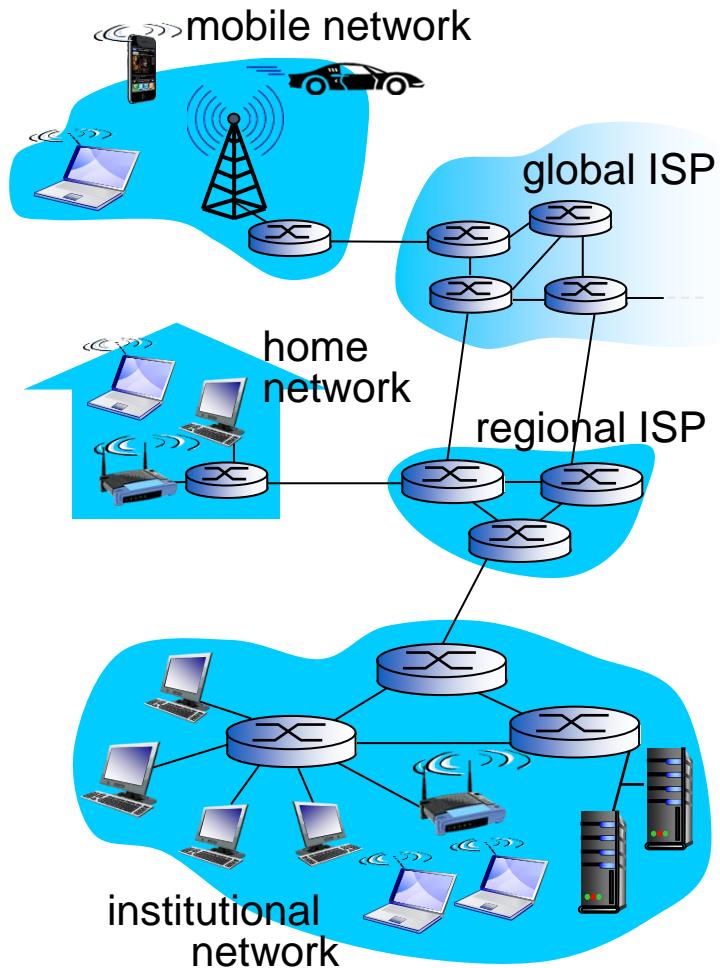
Kiến trúc Internet: Mạng của các mạng (5)

- *Mạng lõi và mạng biên*



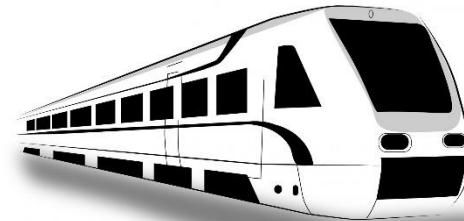
Kiến trúc mạng

- Mạng biên (network edge):
 - Nút mạng đầu cuối (end-system, host): PC, điện thoại, máy chủ, máy tính nhúng...
 - Mạng truy nhập (access network): đường truyền, thiết bị kết nối (router, switch, hub, tổng đài...)
- Mạng lõi (network core):
đường truyền, thiết bị kết nối
 - Mạng của các mạng
- **Mới chỉ đề cập đến khía cạnh “Kết nối như thế nào?”**



2. Chuyển mạch gói & chuyển mạch kênh

- Chuyển tiếp dữ liệu qua các kết nối mạng như thế nào?



2. Chuyển mạch gói & chuyển mạch kênh

- Đặt vấn đề
- Chuyển mạch kênh
- Chuyển mạch gói

2.1. Đặt vấn đề

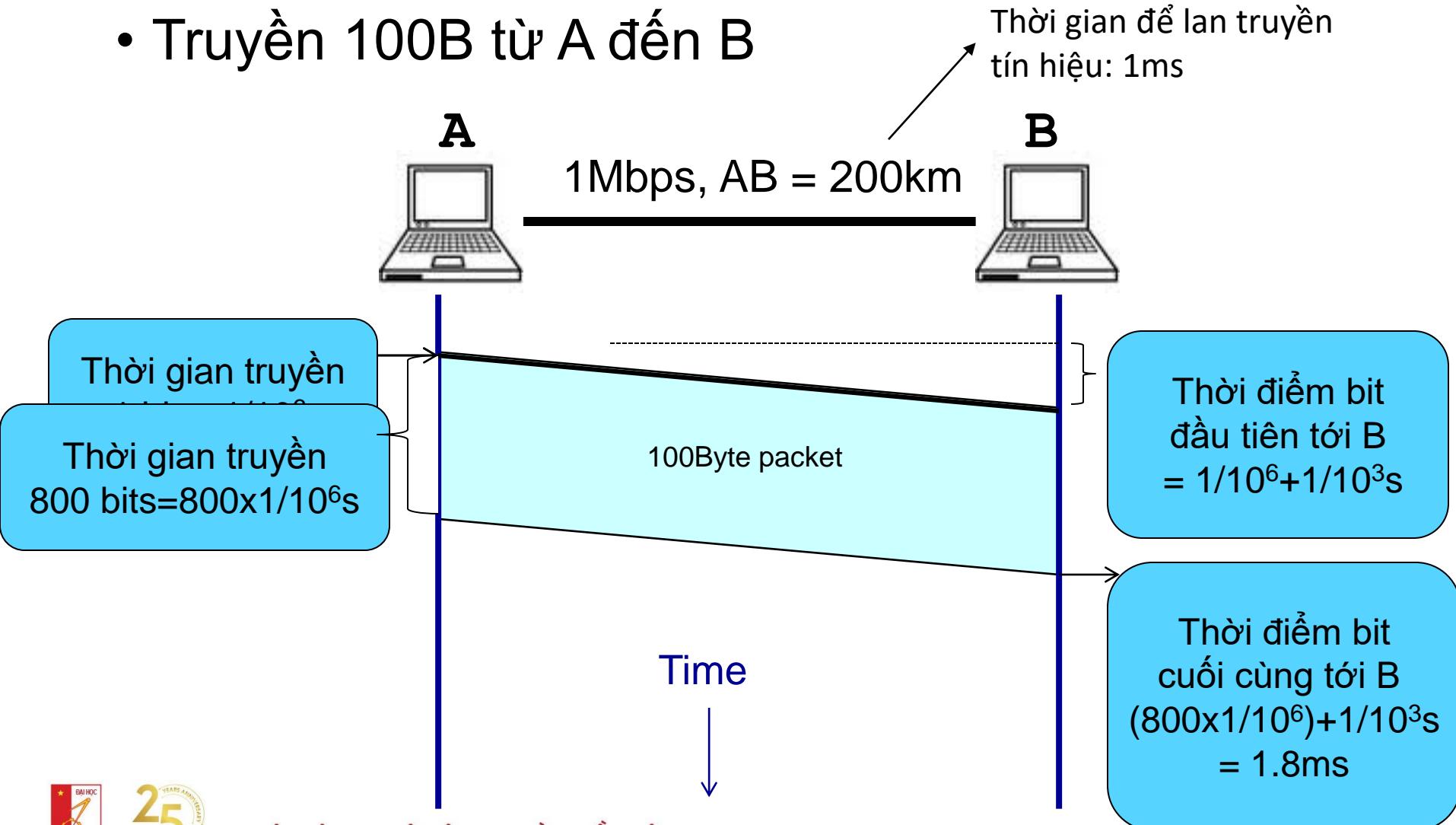
- Kết nối điểm-điểm giữa 2 host



- Thông số của kết nối:
 - Băng thông (bandwidth - R): lượng dữ liệu truyền tối đa trong một đơn vị thời gian (bps – bit per second). Ví dụ: đường cáp quang băng thông (tốc độ truyền) 100Mbps.
 - Trễ (Latency): thời gian truyền liệu từ A đến B
 - Trễ truyền dẫn (propagation delay): Độ dài liên kết / tốc độ tín hiệu. Ví dụ cáp quang dài 10km, truyền tốc độ gần ánh sáng ($3 \times 10^8 \text{m/s}$) $\rightarrow 10 \times 10^3 / (3 \times 10^8) \sim 3.333 \times 10^{-5} = 0.03333 \text{ ms}$
 - Trễ truyền tin: Kích thước dữ liệu / Băng thông

Kết nối điểm-điểm giữa 2 host

- Truyền 100B từ A đến B



Bài tập tại lớp

Câu 3. Các gói dữ liệu có độ dài 1000 bytes được truyền qua đường dây cáp quang với tốc độ 100Mbps. Độ dài của đường dây là 100km.

- a. Tính thời gian phát hết một gói.

Trả lời: (0.25 điểm)

- b. Tính thời gian để truyền 1 bit từ nguồn đến đích (tốc độ lan truyền là 200000km/s).

Trả lời: (0.25 điểm)

- c. Tính số gói dữ liệu cùng xuất hiện trên đường truyền.

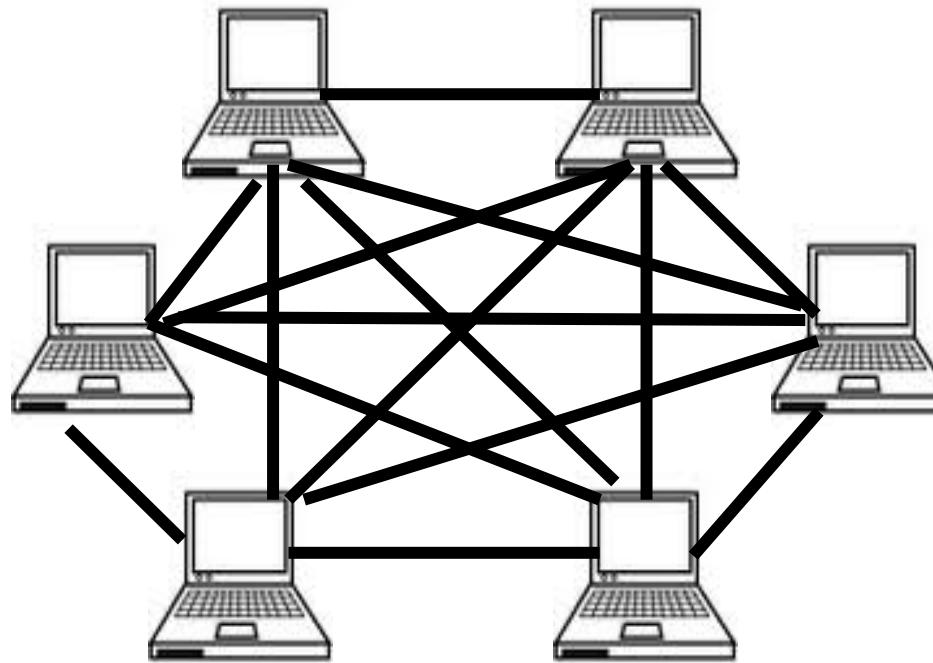
Trả lời: (0.25 điểm)

- d. Xác định tốc độ truyền thực tế, để mỗi bit của một gói phát ra phía bên phát cần được báo nhận bởi một bit phía bên thu.

Trả lời: (0.25 điểm)

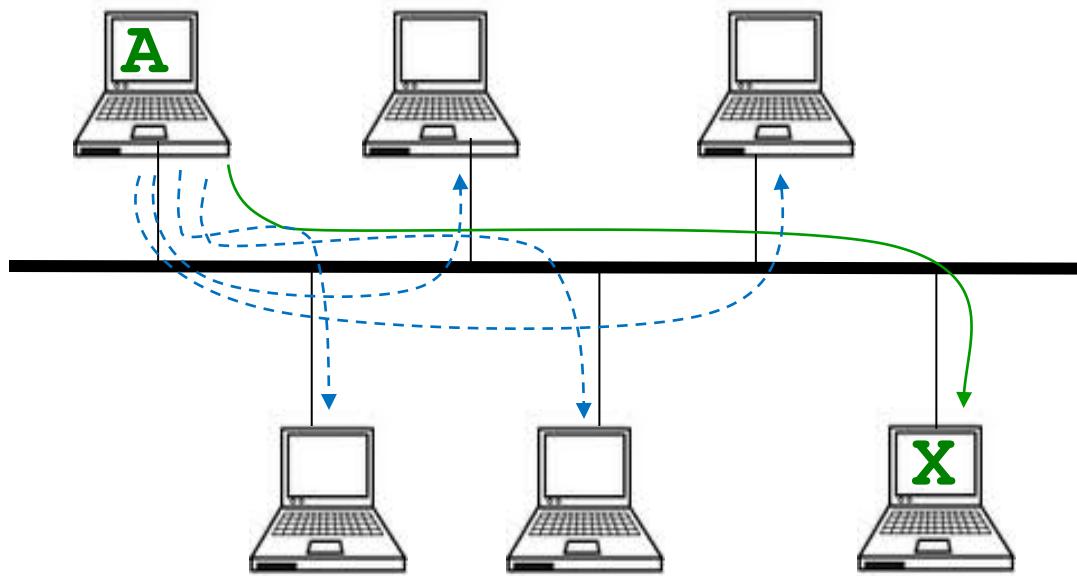
Kết nối giữa nhiều host

- Điểm-điểm giữa mọi cặp
- Hạn chế?



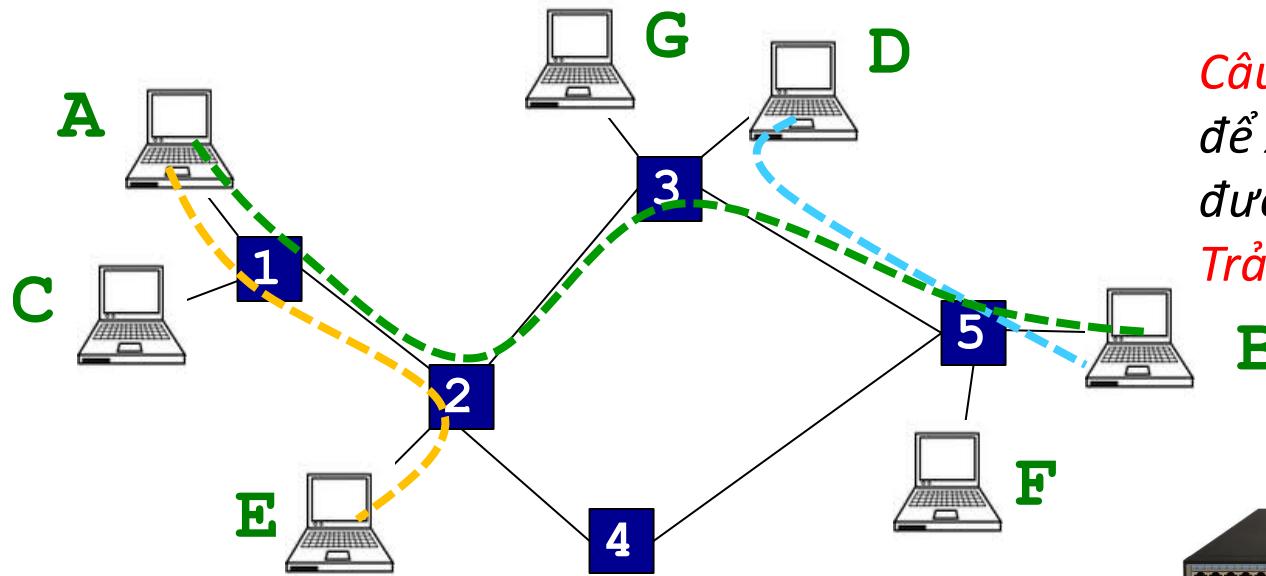
Kết nối giữa nhiều nút mạng

- Điểm-đa điểm: Sử dụng 1 đường truyền chung cho tất cả → truyền “quảng bá” (broadcast)
- Hạn chế?



Kết nối giữa nhiều nút mạng

- Giải pháp: mạng chuyển mạch (switch)
 - Mỗi host kết nối với 1 thiết bị chuyển mạch
 - Các thiết bị chuyển mạch kết nối điểm-điểm và thực hiện chuyển tiếp dữ liệu tới đích
 - Chia sẻ tài nguyên đường truyền



Câu hỏi: Làm thế nào
để xác định được tuyến
đường?

Trả lời: Định tuyến

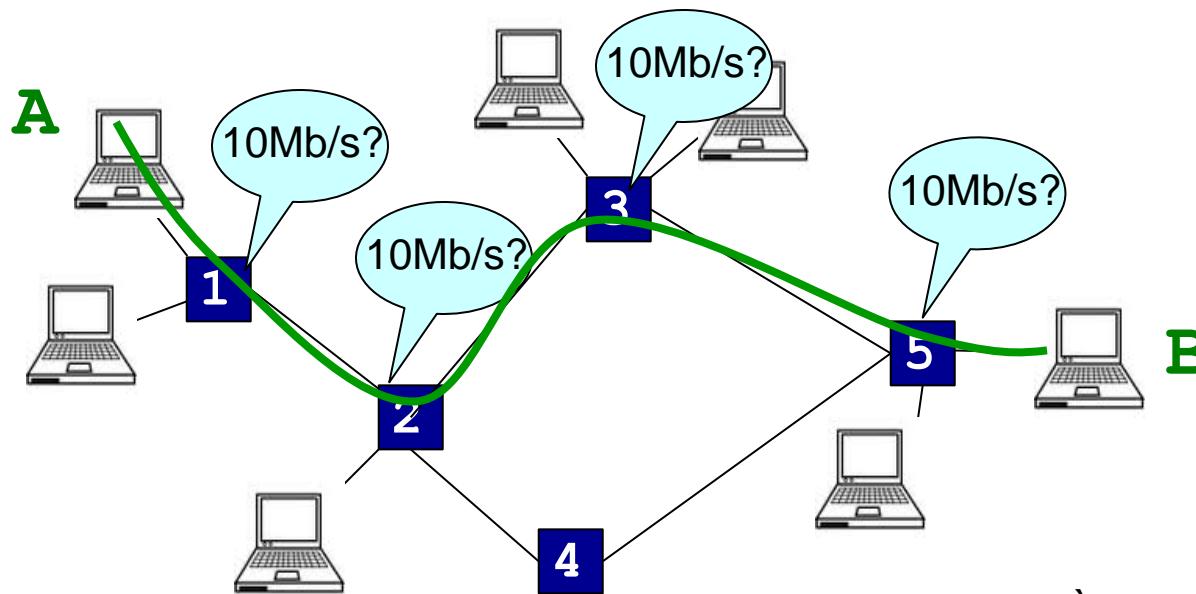


2. Chuyển mạch gói & chuyển mạch kênh

- Đặt vấn đề
- Chuyển mạch kênh
- Chuyển mạch gói

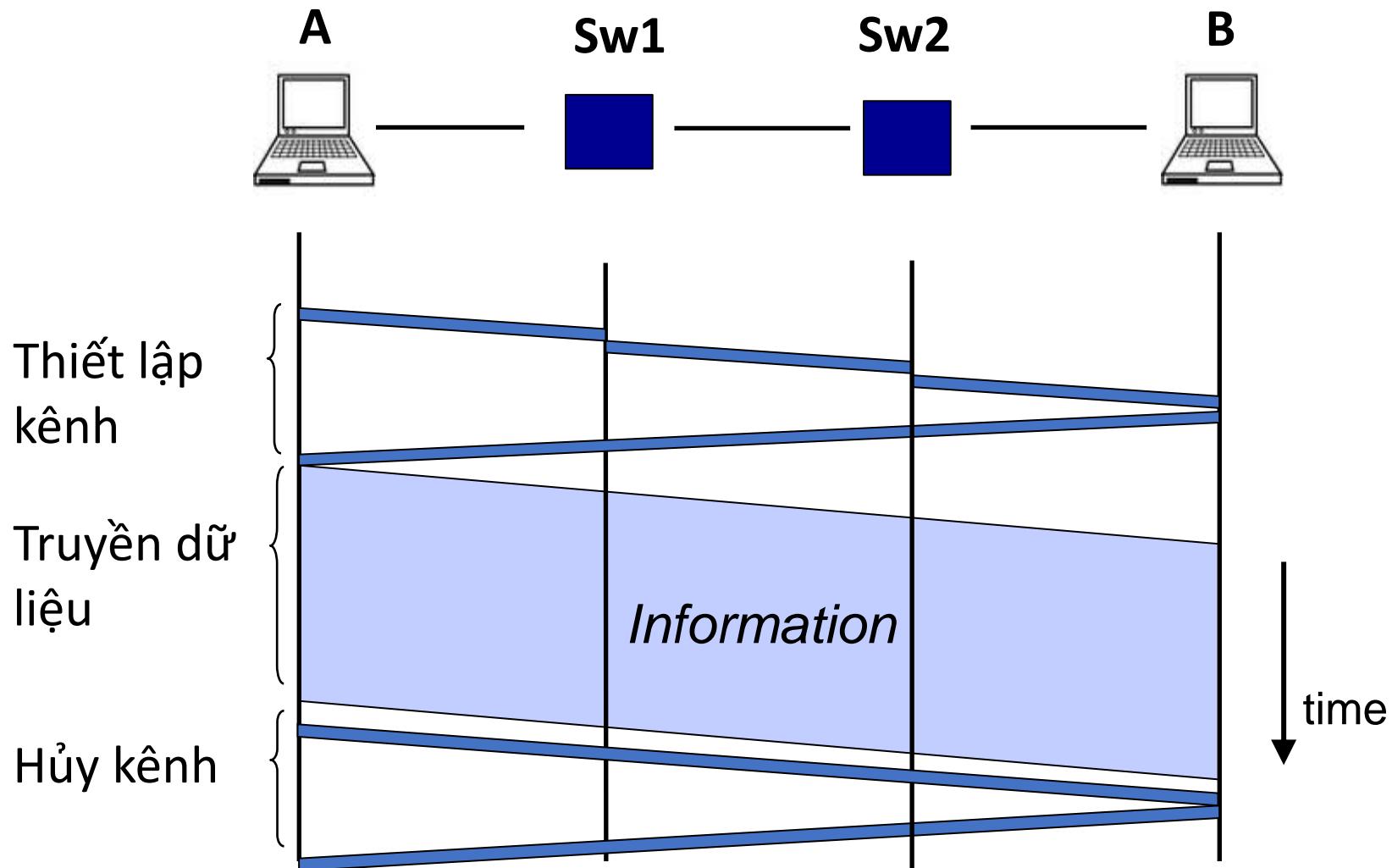
Chuyển mạch kênh

- Circuit switching network: cấp phát tài nguyên đường truyền (kênh) dành riêng cho từng kết nối logic giữa 2 nút mạng



- (1) A phát yêu cầu xin thiết lập kênh
- (2) Các thiết bị chuyển mạch thiết lập kênh
- (3) A bắt đầu truyền dữ liệu
- (4) A truyền xong: phát yêu cầu hủy kênh

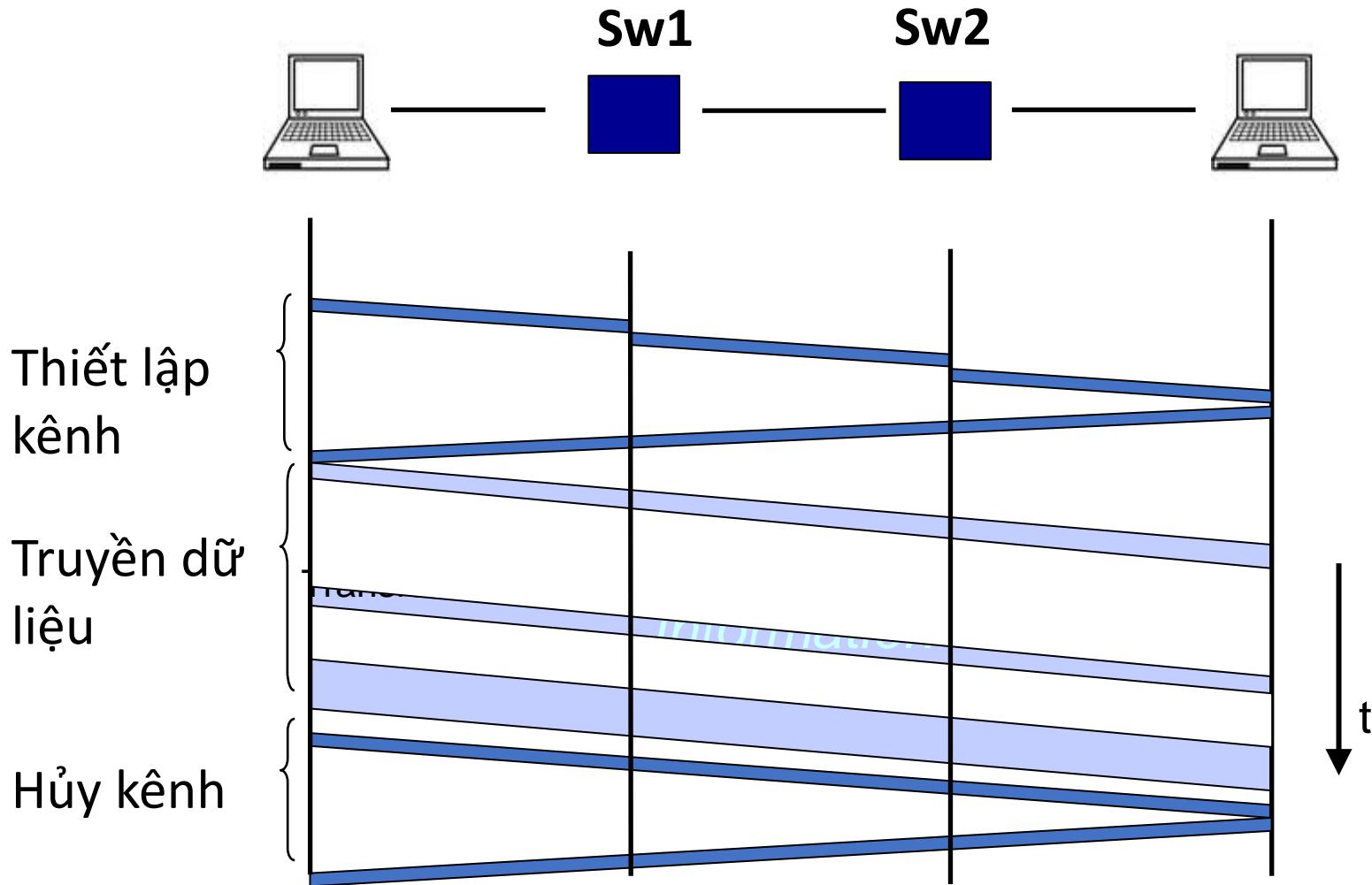
Giản đồ thời gian



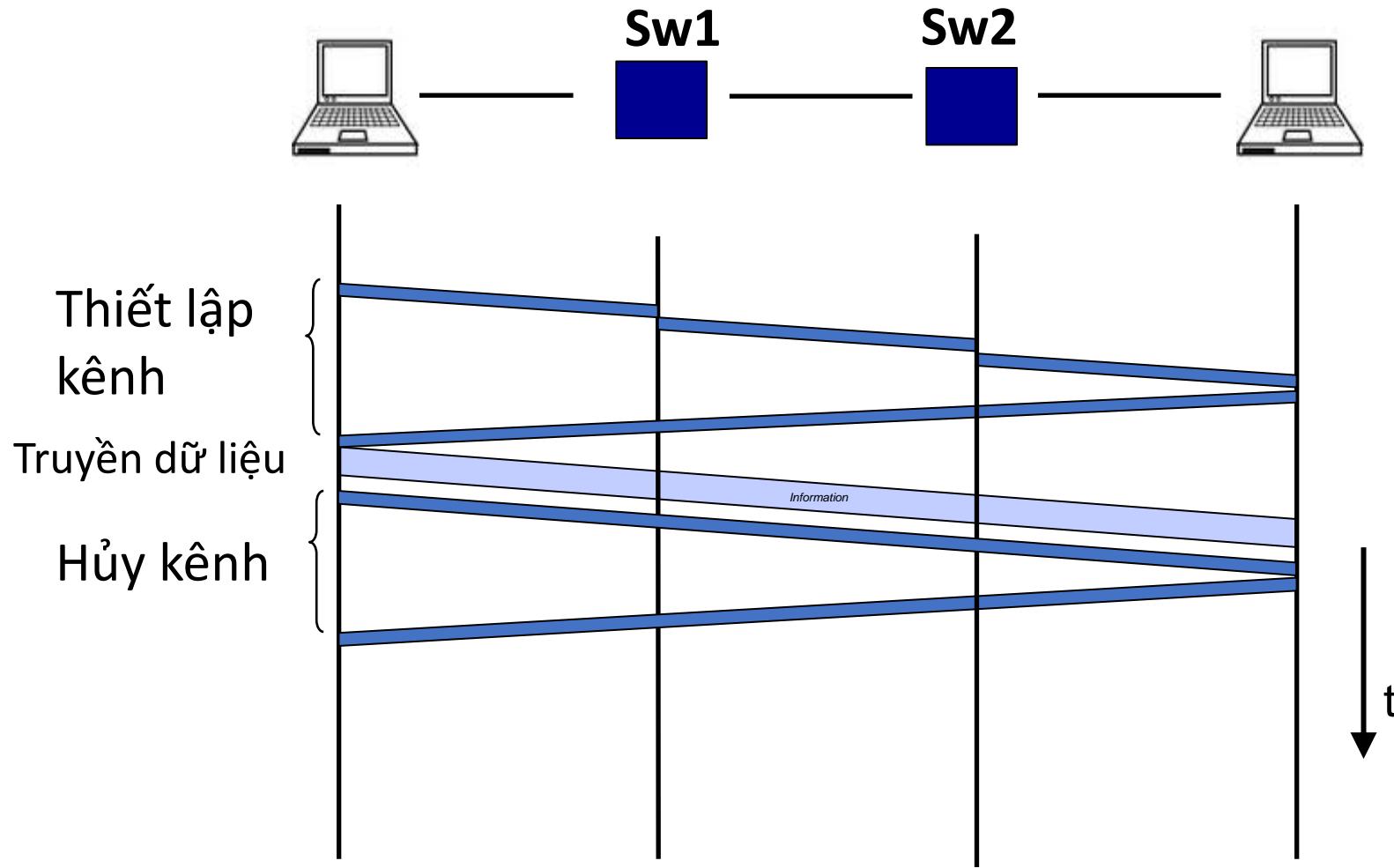
Ưu điểm và nhược điểm

- **Ưu điểm:**
 - Kênh được thiết lập sẵn → Trễ khi chuyển mạch rất thấp
 - Tài nguyên dành riêng cho kênh và không đổi trong quá trình truyền → đảm bảo chất lượng dịch vụ
- **Nhược điểm?**

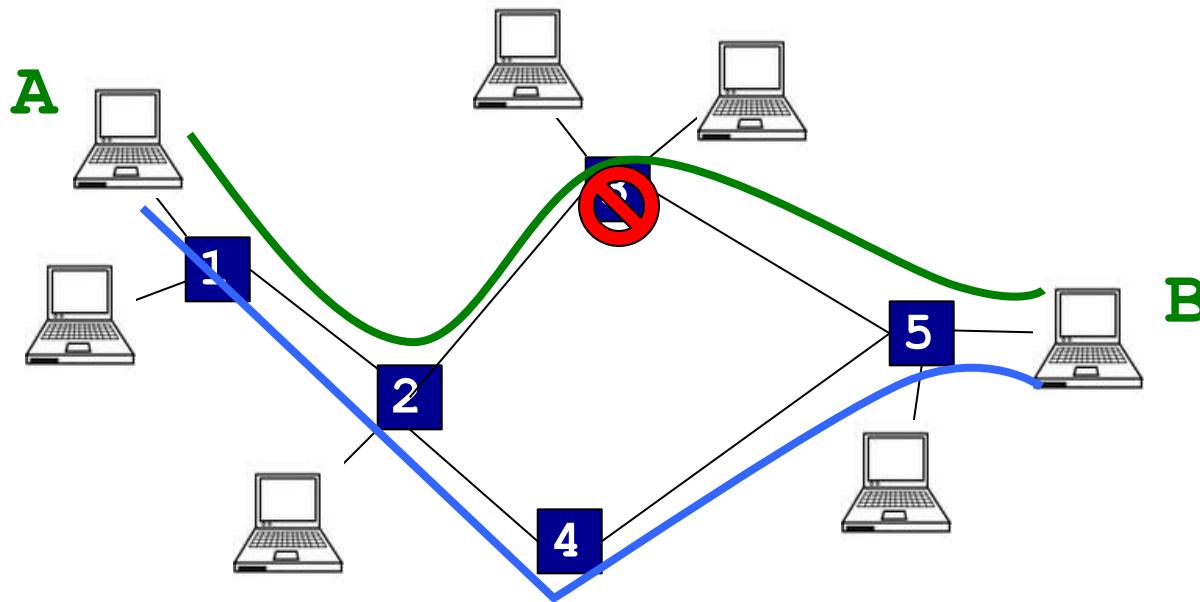
Nhược điểm: kênh “trắng”



Nhược điểm: kênh “bé”



Nhược điểm: lỗi kênh truyền



- Bắt đầu lại toàn bộ quá trình thiết lập kênh truyền nếu lỗi xảy ra trên thiết bị chuyển mạch khi truyền

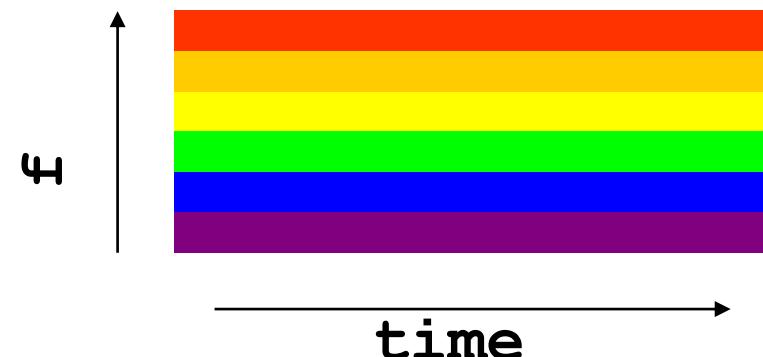
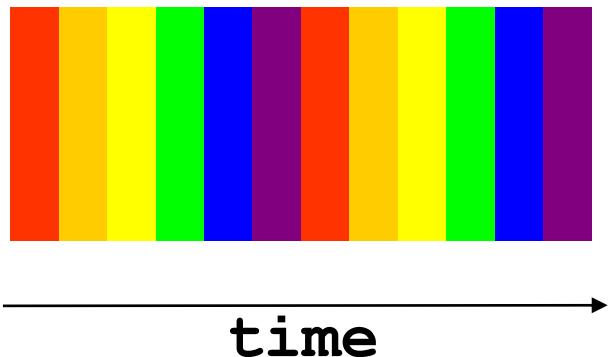
Ghép kênh/Phân kênh

- Mong muốn: nhiều kênh truyền dữ liệu trên 1 kênh liên kết vật lý
- Ghép kênh (Multiplexing): gửi dữ liệu của nhiều kênh khác nhau trên cùng một liên kết vật lý
- Phân kênh (Demultiplexing): phân dữ liệu nhận được trên liên kết vật lý vào các kênh tương ứng và chuyển đến đúng đích



Một số kỹ thuật ghép kênh

- Ghép kênh theo thời gian(TDM): mỗi kết nối sử dụng tài nguyên trong khe thời gian được phân
- Ghép kênh theo tần số(FDM): mỗi kết nối sử dụng một băng tần tín hiệu riêng



2. Chuyển mạch gói & chuyển mạch kênh

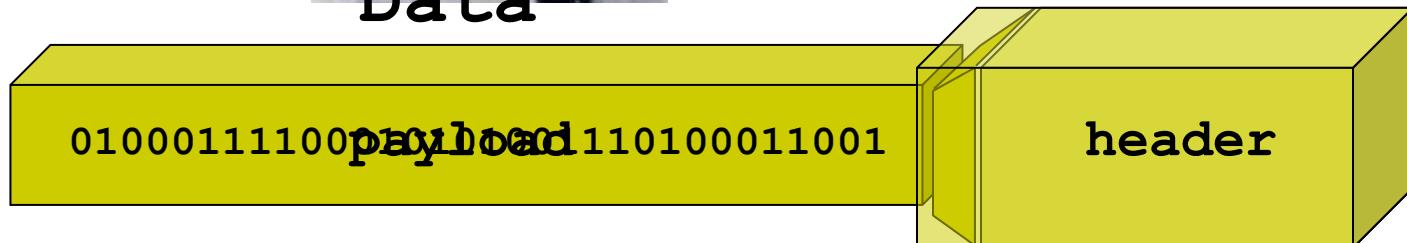
- Đặt vấn đề
- Chuyển mạch kênh
- Chuyển mạch gói

2.3. Chuyển mạch gói

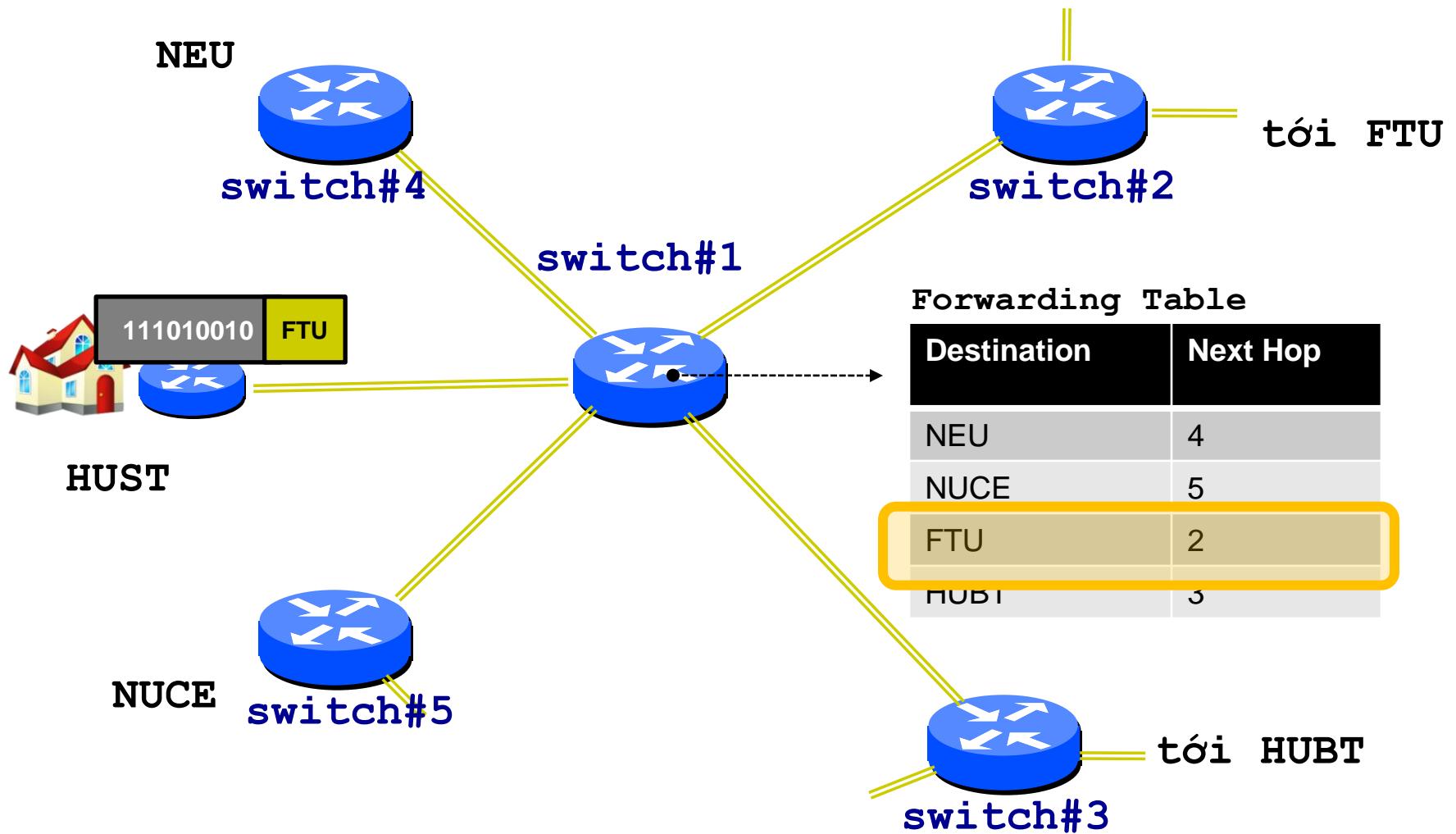
- Dữ liệu được chia thành các gói tin (packet)
 - Phần tiêu đề (header): địa chỉ, số thứ tự
 - Phần dữ liệu (payload data)
- Thiết bị chuyển mạch chuyển tiếp gói tin dựa trên tiêu đề



Data

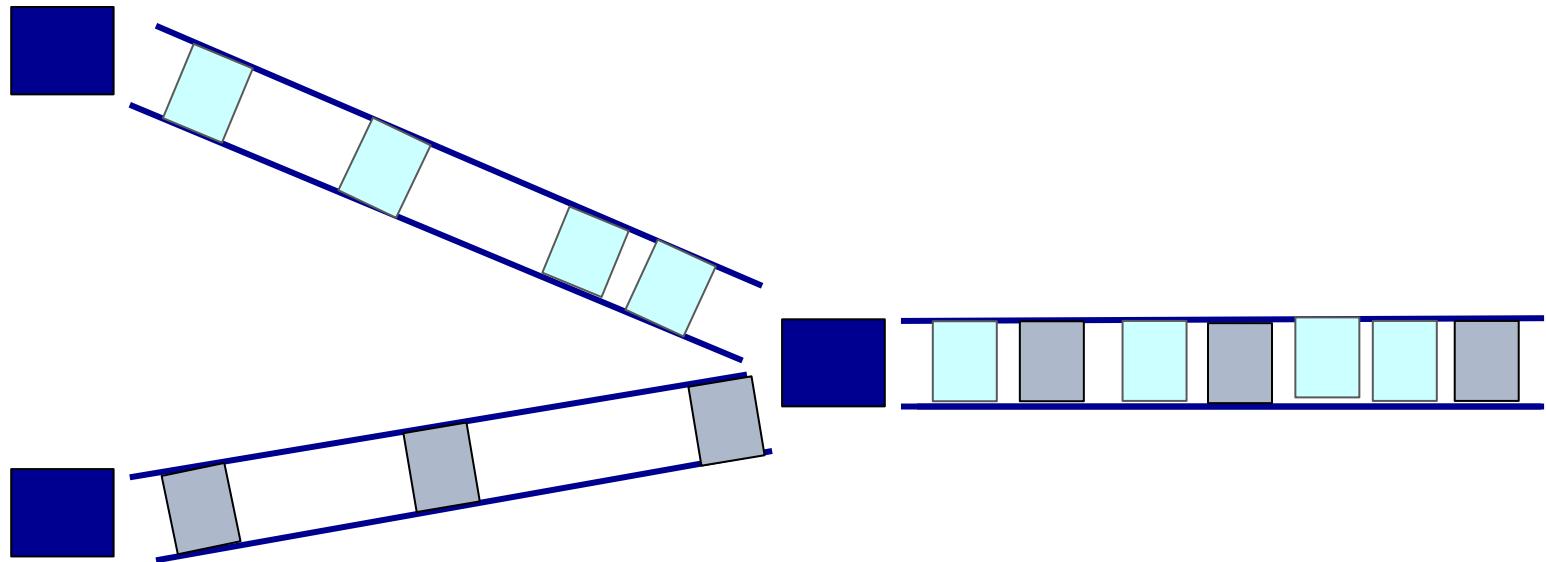


Chuyển tiếp gói tin



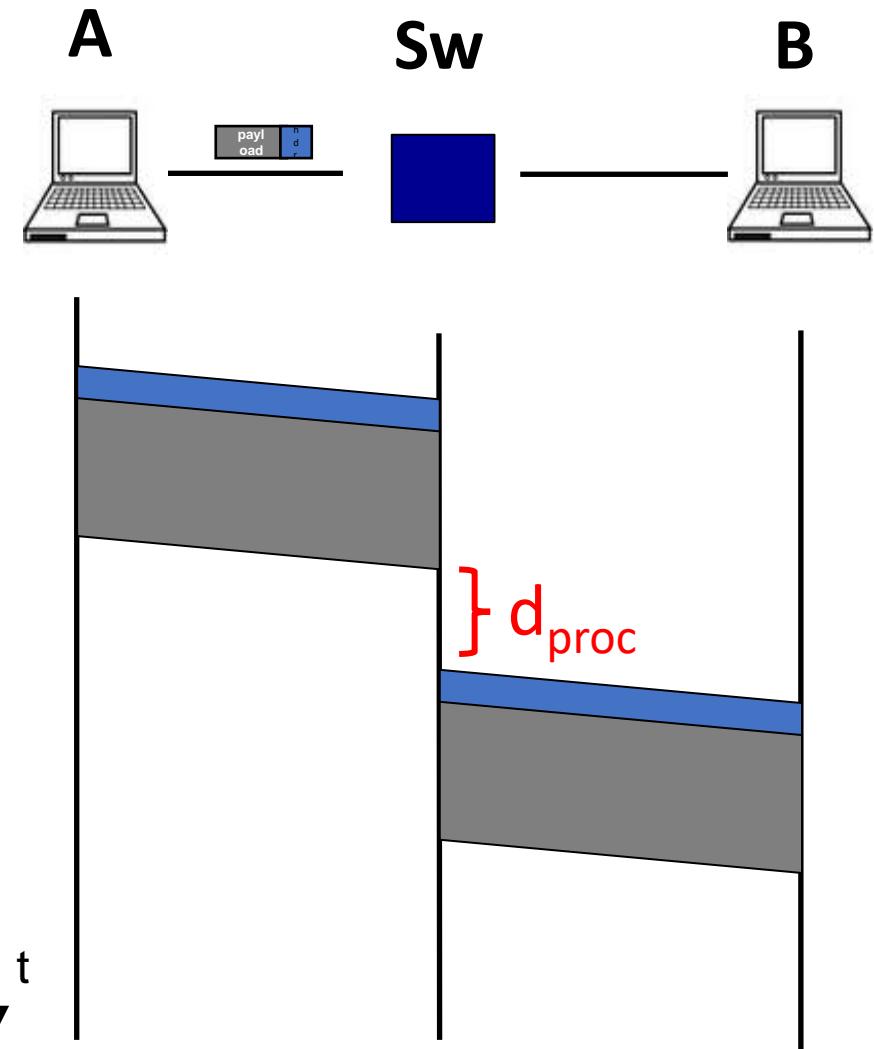
Chuyển mạch gói

- Mỗi gói tin có thể được xử lý độc lập
 - Các gói tin có thể tới đích theo các đường khác nhau, không còn đúng thứ tự
- Tài nguyên dùng chung cho tất cả các kết nối
 - Nếu còn tài nguyên, bất kỳ nút nào cũng có thể sử dụng



Giản đồ thời gian

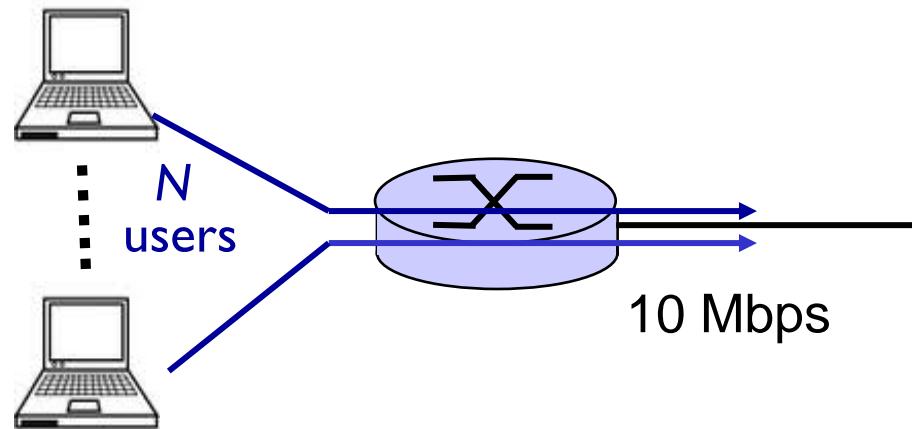
- Thiết bị chuyển mạch chỉ chuyển tiếp khi nhận được đầy đủ gói tin (**store and forward**)
- Thiết bị chuyển mạch cần thời gian để xử lý gói tin (d_{proc}):
 - Kiểm tra lỗi trên gói tin
 - Quyết định gói tin gửi đến đâu
 - Thường rất nhỏ so với trễ truyền tin



Chuyển mạch gói vs Chuyển mạch kênh

Ví dụ:

- Băng thông đi 10 Mb/s
- Mỗi kết nối của người dùng tới:
 - Được cấp phát 1 Mb/s
 - Thời gian sử dụng để truyền dữ liệu: 10% tổng thời gian



❖ Mạng chuyển mạch kênh:

- Tối đa 10 người dùng đồng thời xin cấp phát

❖ Mạng chuyển mạch gói:

- Giả sử có 30 người dùng sử dụng chung
- Xác suất để >10 người dùng đồng thời truyền dữ liệu là bao nhiêu? (~0.00036)

- *Phân phối nhị thức:*
 $P(x = k) = C_n^k p^k (1-p)^{n-k}$
- *Nếu số người dùng tăng lên?*

Tính hiệu suất chuyển mạch

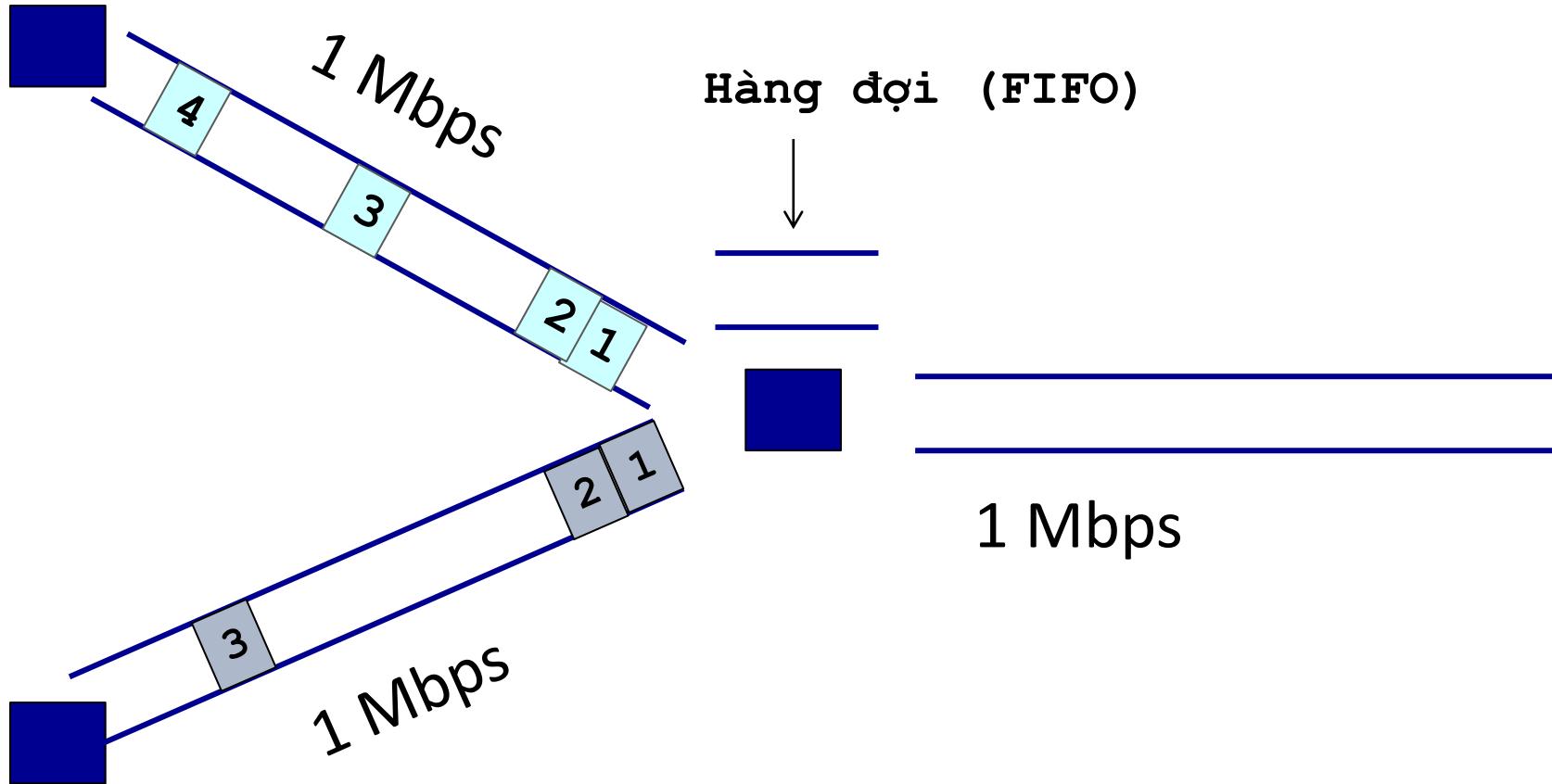
- Mạng chuyển mạch kênh: Xác suất cả 10 máy của người dùng truyền dữ liệu:

$$P(k = 10) = C_{10}^{10} \times 0.1^{10} \times 0.9^0 = 10^{-10}$$

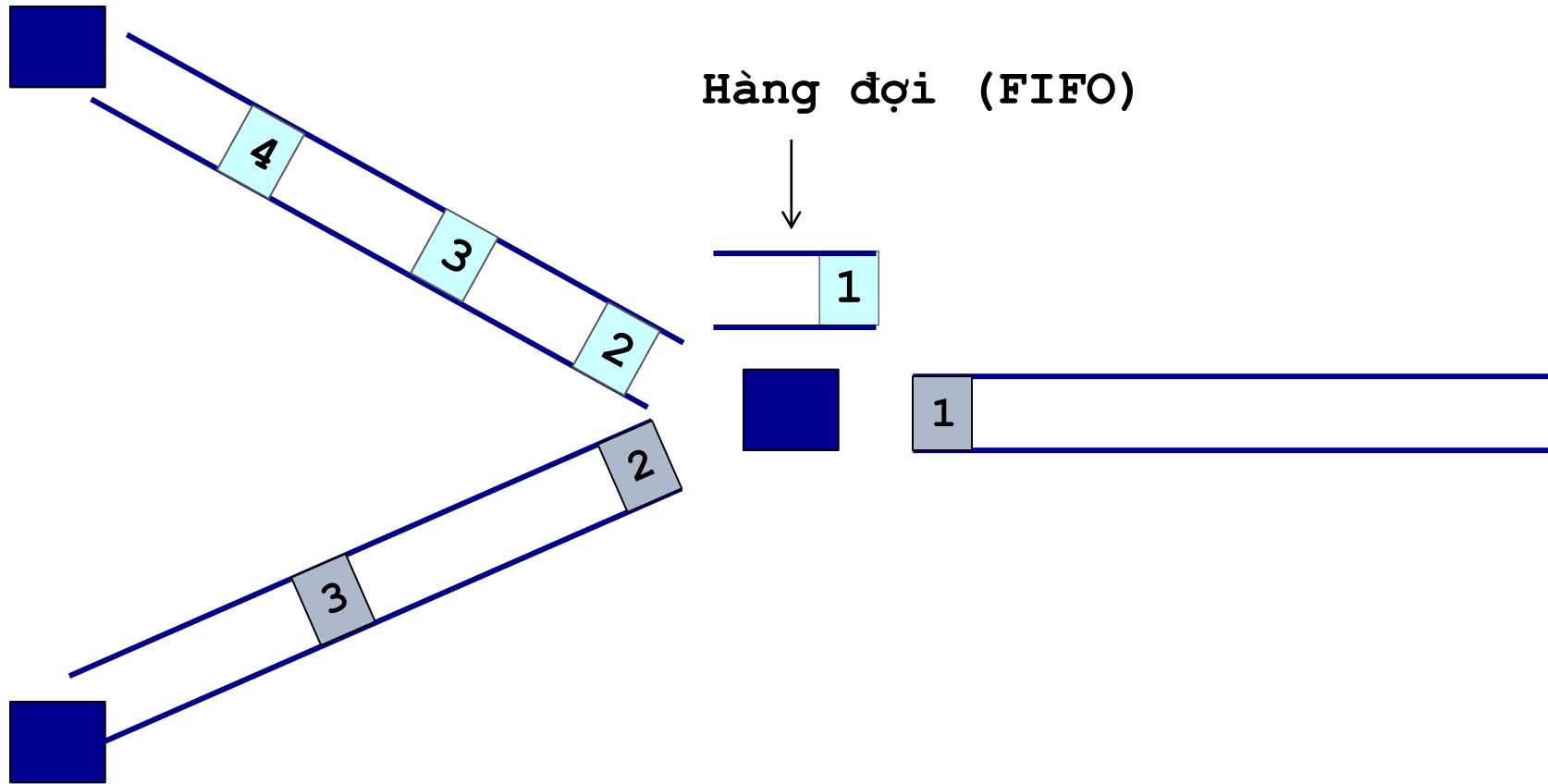
- Mạng chuyển mạch gói: Xác suất cả 10 máy truyền dữ liệu (với 30 máy)

$$P(k = 10) = C_{30}^{10} \times 0.1^{10} \times 0.9^{20} = 0.00037$$

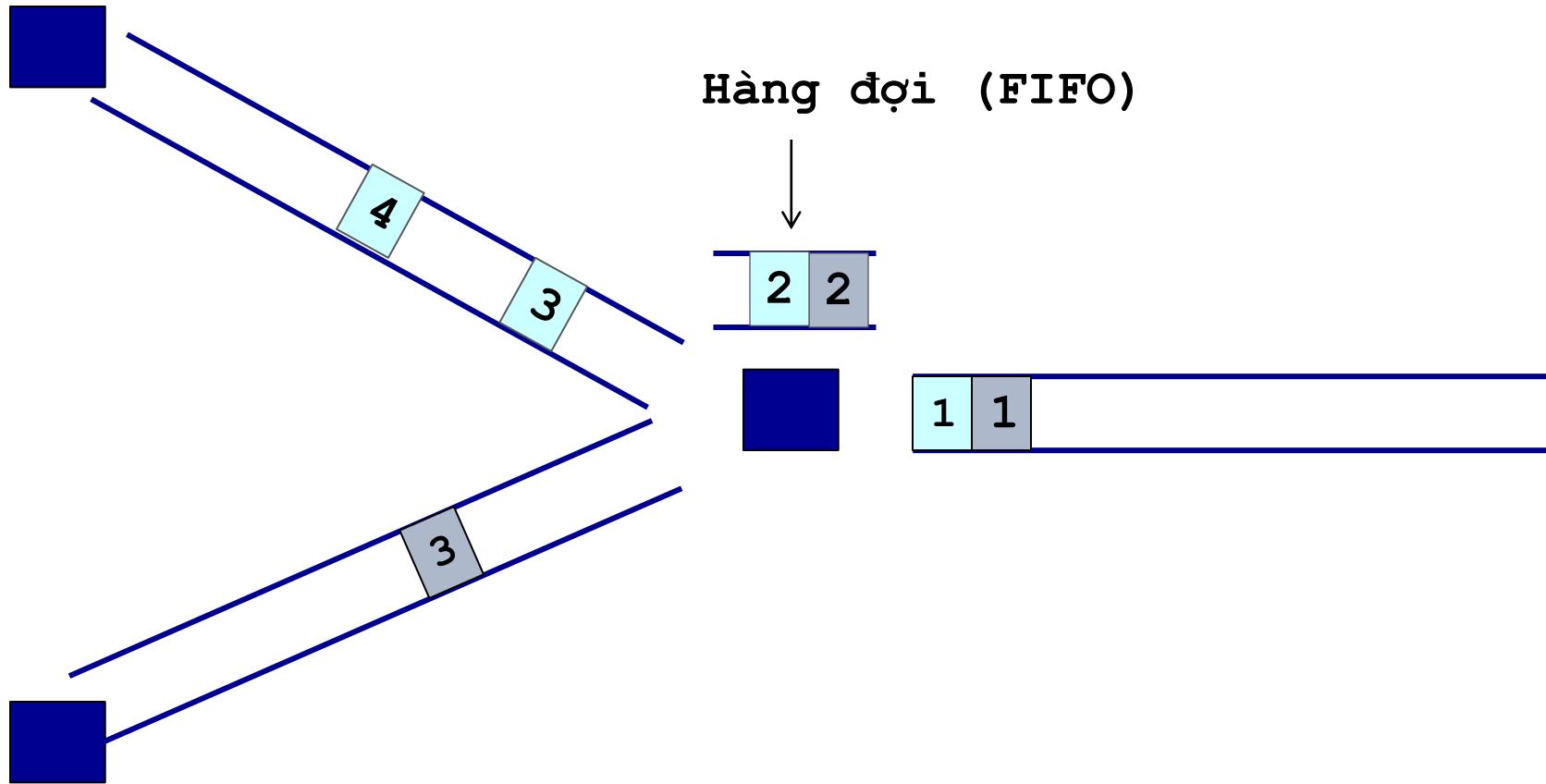
Hàng đợi



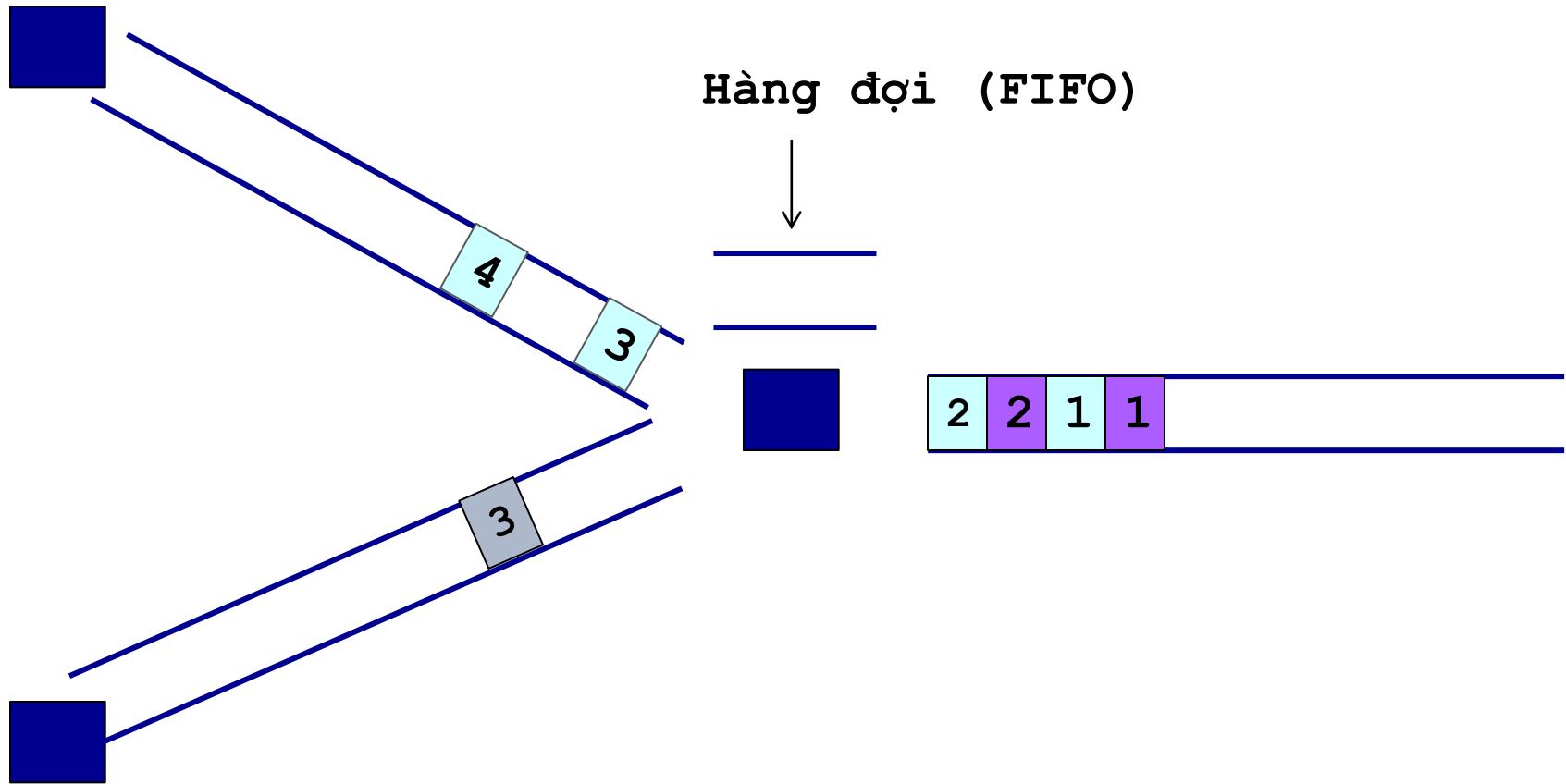
Hàng đợi



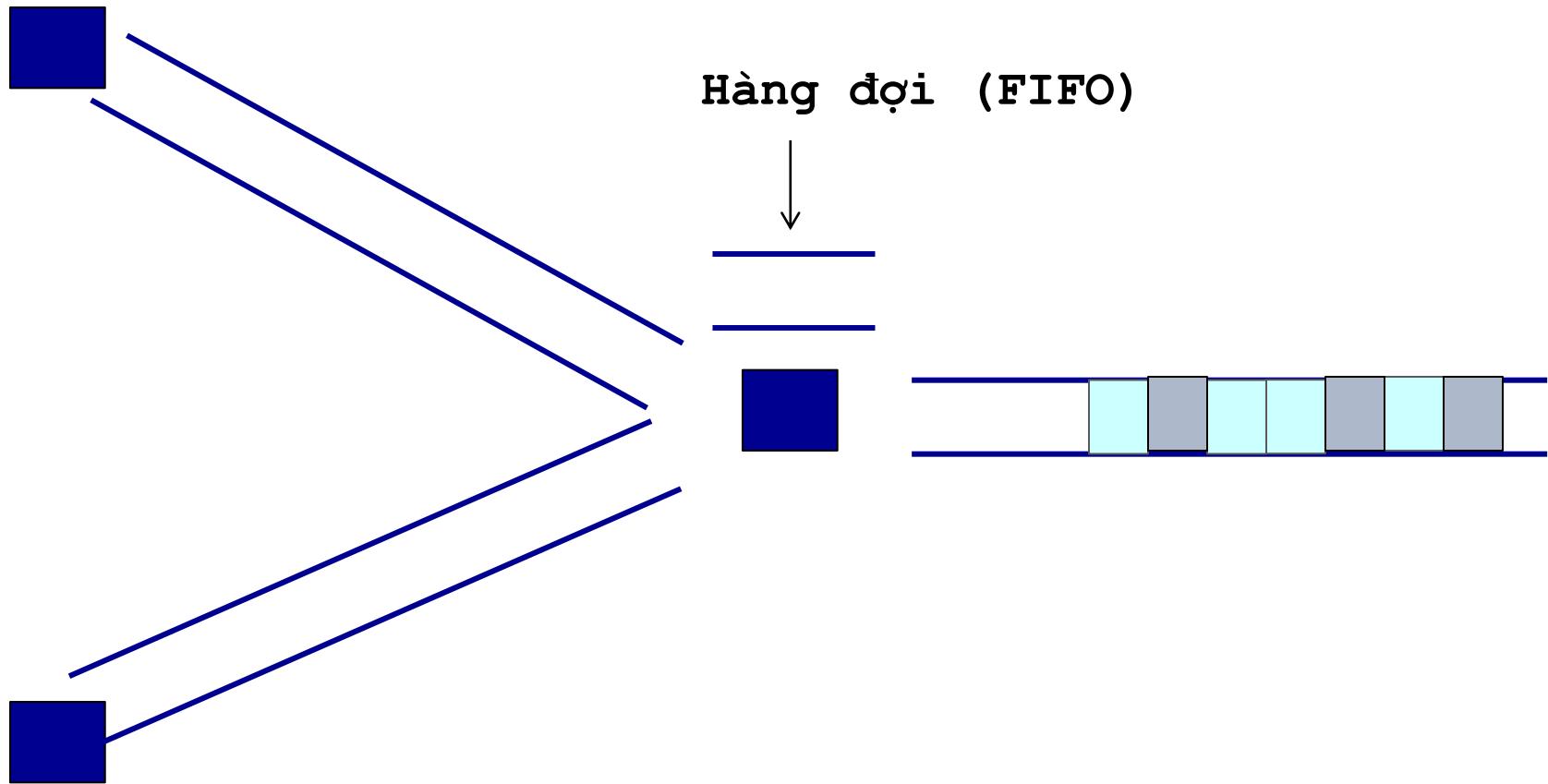
Hàng đợi



Hàng đợi

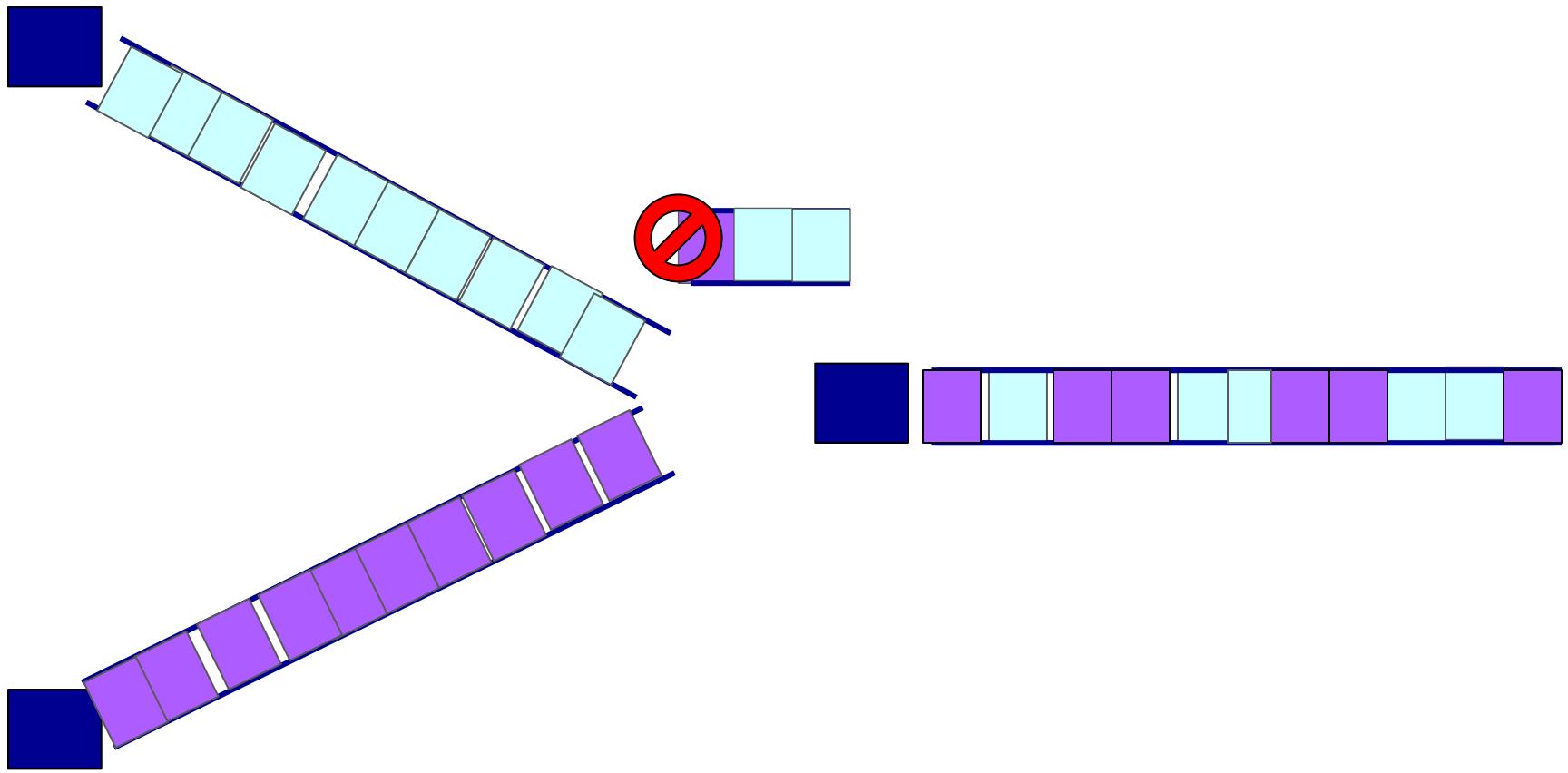


Hàng đợi



Mất gói tin

- Kích thước hàng đợi có hạn
- Gói tin tới khi hàng đợi đã đầy sẽ bị mất



3. Một số thông số cơ bản trong mạng

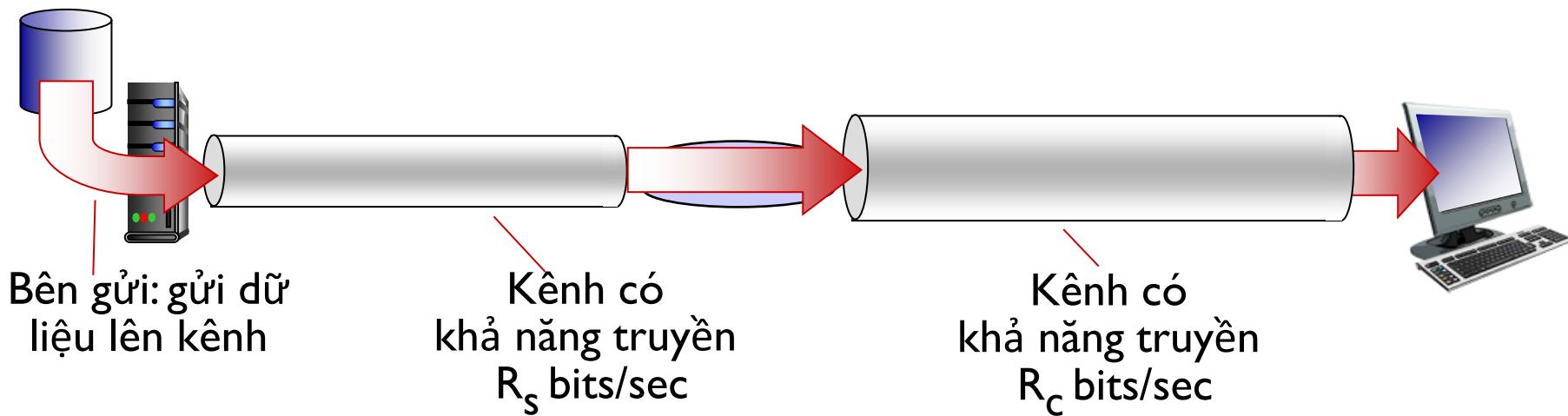
- Băng thông
- Thông lượng
- MTU
- Độ trễ
- Độ mất gói tin

Các thông số cơ bản

- Băng thông ≡ Tốc độ truyền tin ≡ Dung lượng
- Thông lượng
- MTU(Maximum Transmission Unit): kích thước lớn nhất của gói tin
- Độ trễ
 - Trễ trên thiết bị đầu cuối
 - Trễ trên thiết bị trung gian
 - Trễ truyền tin
 - Trễ lan truyền
- Độ mất gói tin

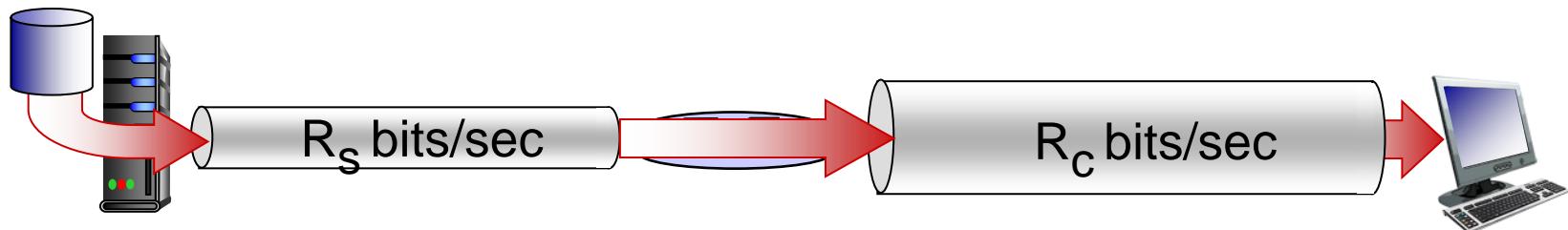
Thông lượng (throughput)

- ❖ *Thông lượng*: tốc độ (bits/sec) truyền tin tại một điểm nào đó trong mạng
 - *Tức thời*: thông lượng tại một thời điểm
 - *Trung bình*: thông lượng tính trung bình trong một khoảng thời gian

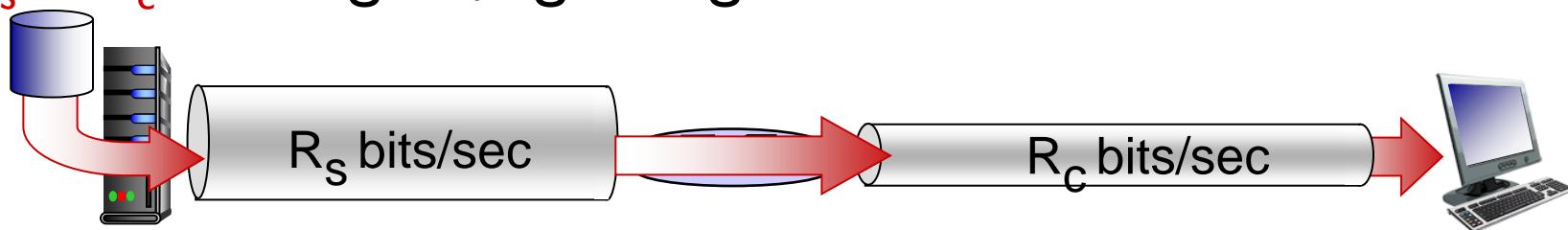


Thông lượng (tiếp)

- ❖ $R_s < R_c$ Thông lượng trung bình là bao nhiêu?



- ❖ $R_s > R_c$ Thông lượng trung bình là bao nhiêu?

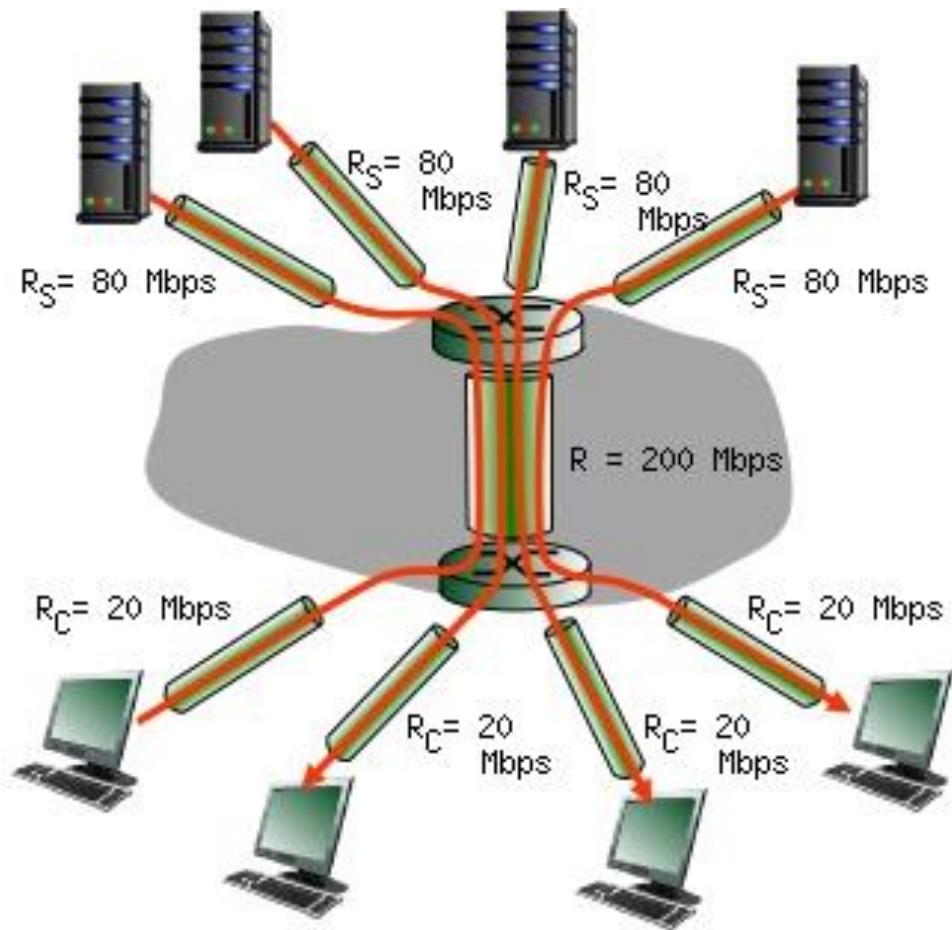


Nút thắt cổ chai (bottleneck)

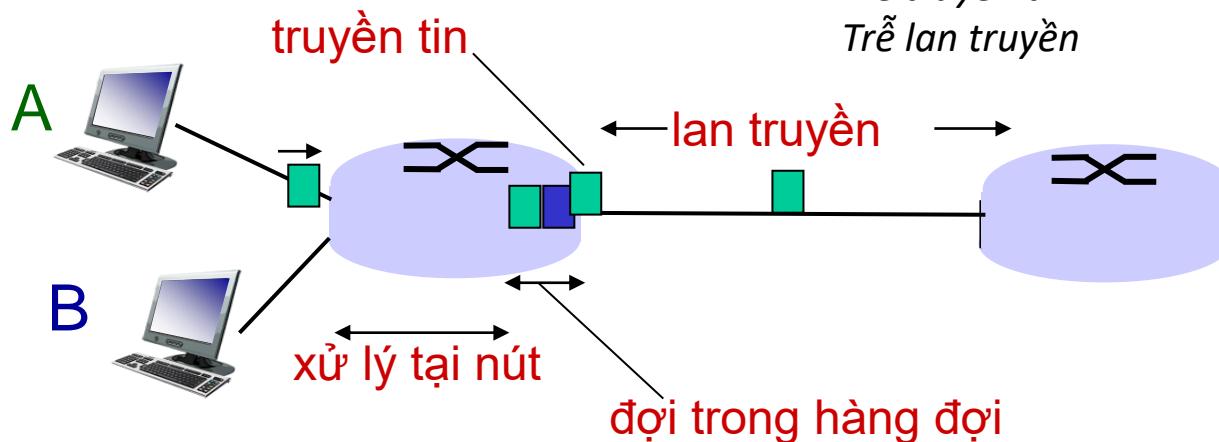
Là điểm tại đó làm giới hạn thông lượng trên đường truyền

Nút thắt cổ chai

- Xác định nút thắt cổ chai?



Độ trễ



Trễ trên thiết bị đầu cuối
Trễ trên thiết bị trung gian
Trễ truyền tin
Trễ lan truyền

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

d_{trans} : trễ truyền tin:

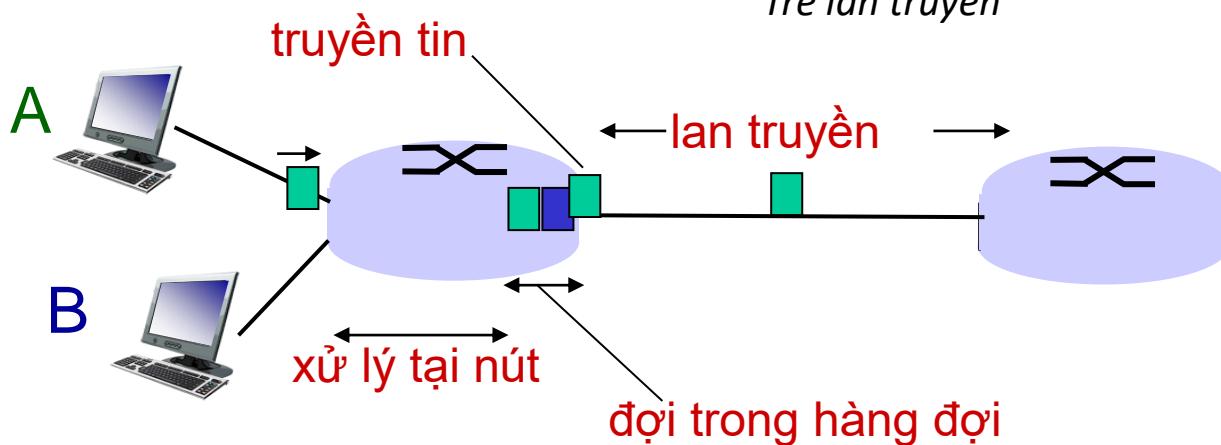
- L : kích thước dữ liệu (bits)
- R : băng thông (bps)
- $d_{\text{trans}} = L/R$

d_{prop} : trễ lan truyền (truyền dẫn)

- d : độ dài đường truyền
- s : tốc độ lan truyền tín hiệu ($\sim 2 \times 10^8$ m/sec)
- $d_{\text{prop}} = d/s$

Độ trễ (tiếp)

Trễ trên thiết bị đầu cuối
Trễ trên thiết bị trung gian
Trễ truyền tin
Trễ lan truyền



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

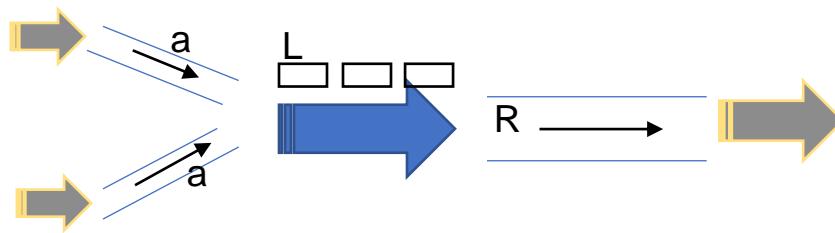
d_{proc} : trễ xử lý

- Kiểm tra lỗi bit
- Xác định liên kết ra
- Thường < μsec

d_{queue} : trễ hàng đợi

- Phụ thuộc vào số lượng dữ liệu trong hàng đợi

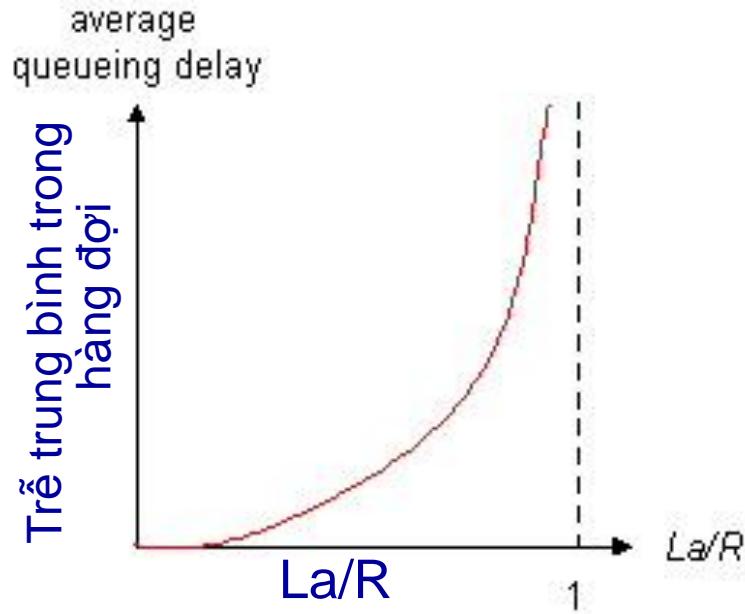
Trễ hàng đợi



- ❖ R : băng thông (bps)
- ❖ L : kích thước gói tin (bits)
- ❖ a : tốc độ đến của gói tin

- ❖ $La/R \sim 0$: trễ hàng đợi nhỏ
- ❖ $La/R \rightarrow 1$: trễ hàng đợi lớn
- ❖ $La/R > 1$: trễ vô cùng (mất gói tin)

➔ Bài toán điều phối tốc độ a trên kênh truyền điểm cuối đến điểm cuối (end-to-end)



$La/R \sim 0$

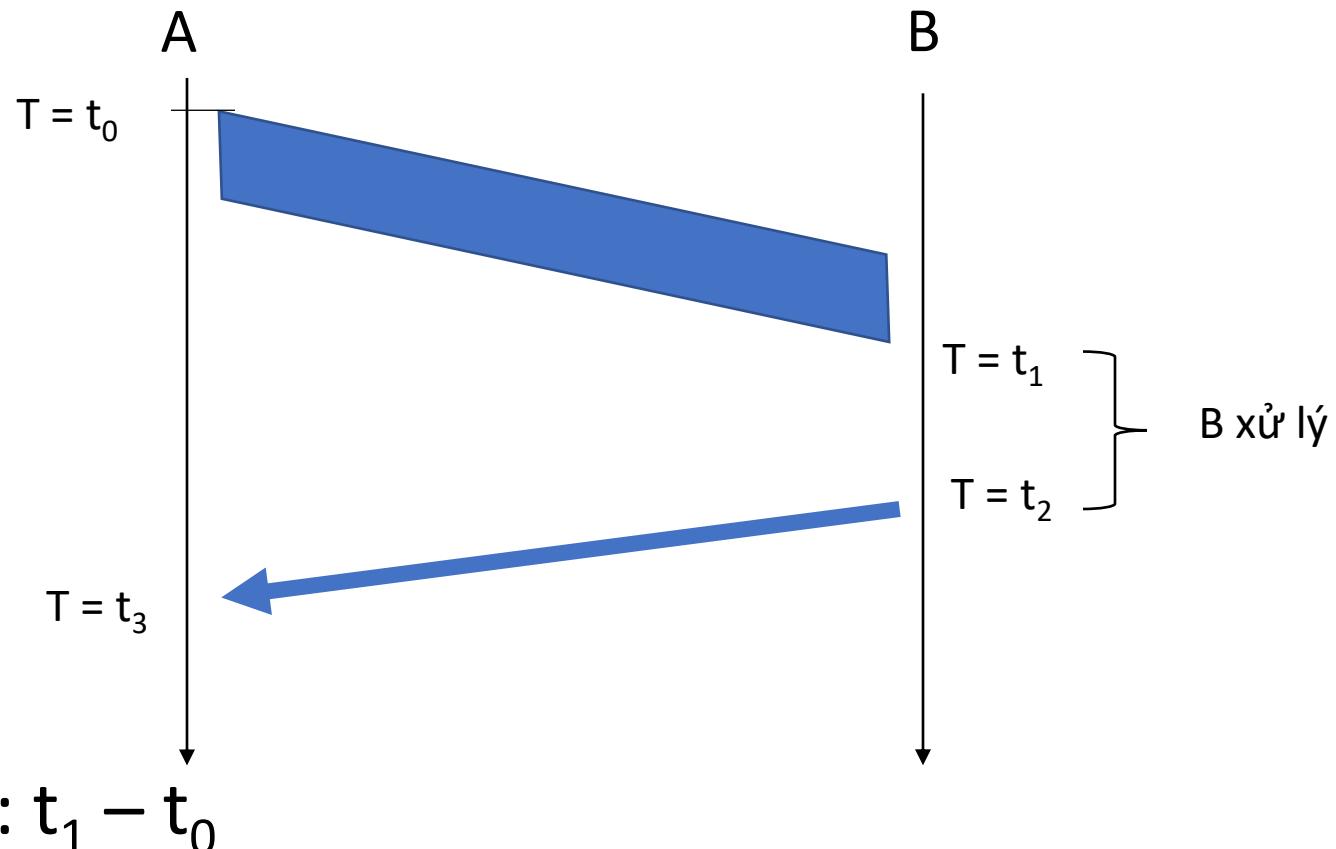


$La/R \rightarrow 1$



Round Trip Time

- RTT: (two-way) = $t_3 - t_0$



- One-way: $t_1 - t_0$

MTU

- Maximum Transmission Unit: Kích thước tối đa của gói tin có thể truyền trên đường truyền
- Ví dụ: Mạng Ethernet có MTU = 1526 byte
- Tại sao?
- Lý do 1: Giảm tỉ lệ gói tin bị lỗi bit
 - BER = Số bit lỗi / Tổng số bit truyền → thường là hằng số
 - Ví dụ: $BER = 10^{-3}$ → truyền 1000 bit sẽ lỗi 1 bit
 - Nếu gói tin kích thước $L = 1000$ bit → xác suất gói tin có lỗi bit?
 - Nếu $L = 100$ bit → xác suất gói tin có lỗi bit?
- Lý do 2: Giảm xác suất (kích thước dữ liệu) phải truyền lại do mất gói tin
 - Kích thước hàng đợi: N byte
 - Nếu gói tin có kích thước $L = 1000$ byte; hàng đợi đã đầy → gói tin bị mất → truyền lại gói tin → kích thước dữ liệu cần truyền lại: 1000 byte
 - Nếu gói tin có kích thước $L = 100$ byte: hàng đợi đầy → ?
- Kết luận: MTU làm giảm kích thước dữ liệu phải truyền lại

Tại sao MTU không nên quá nhỏ?

- MTU quá nhỏ làm giảm hiệu suất truyền
- Giải thích:
 - Gói tin gồm: tiêu đề (header) + phần thân (payload)
 - Kích thước header: hằng số
 - Hiệu suất truyền:

$$H = \frac{payload}{header+payload}$$

4. Kiến trúc phân tầng

- Mô hình thiết kế phân tầng
- Truyền thông trong kiến trúc phân tầng
- Mô hình OSI và TCP/IP
- Định danh trong TCP/IP
- Tổng kết về phân tầng & chồng giao thức

Tiếp tục với chủ đề “Làm thế nào để các nút mạng trao đổi thông tin?”

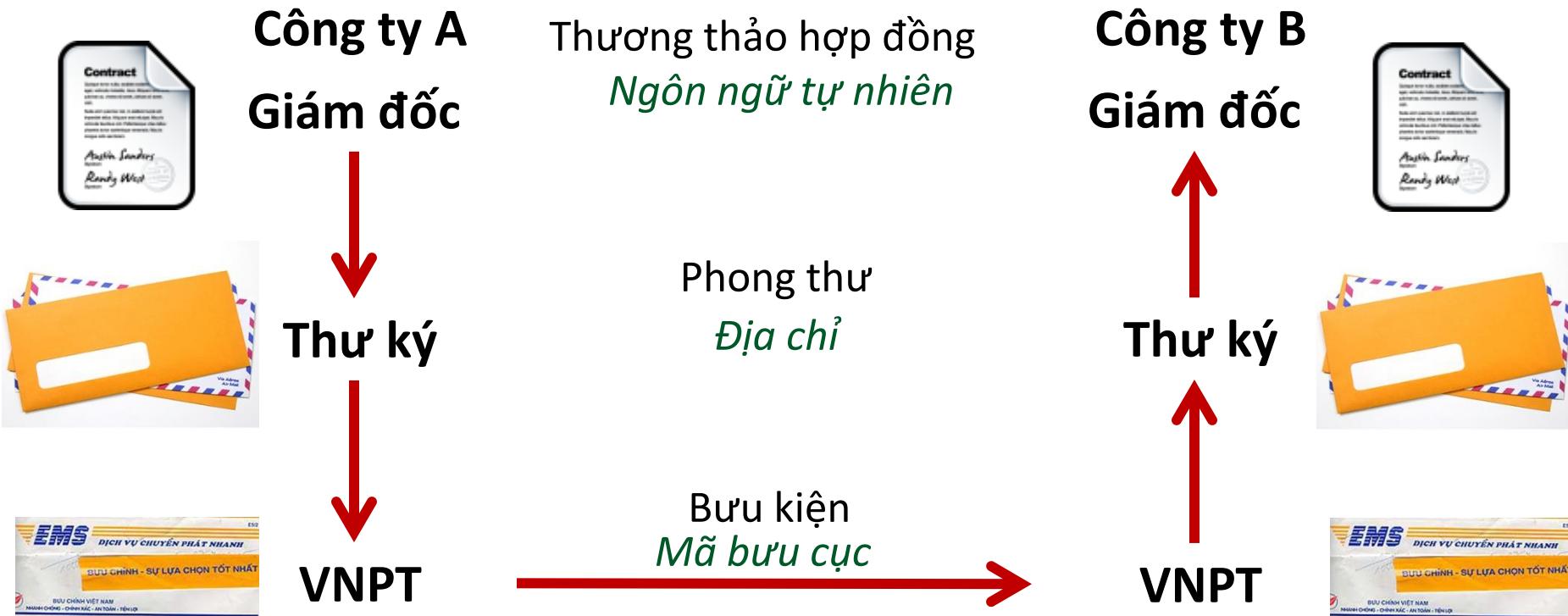


Nguyên tắc “chia để trị”

- Xác định các nhiệm vụ cần thực hiện
- Tổ chức, điều phối thứ tự thực hiện các nhiệm vụ
- Phân định ai làm nhiệm vụ gì
- Ví dụ: Giám đốc công ty A & Giám đốc công ty B trao đổi thương thuyết hợp đồng
 - **Giám đốc A:** chỉnh nội dung hợp đồng & chuyển thư ký
 - **Thư ký:**
 - ❖ Format hợp đồng, cho vào bì thư, điền tên & địa chỉ công ty B
 - ❖ Đem đến bưu điện VNPT
 - **Bưu điện VNPT:**
 - ❖ Đóng gói bưu kiện
 - ❖ Ghi địa chỉ bưu cục nhận
 - ❖ Chuyển bưu kiện lên xe thư
 - ❖ Đưa bưu kiện đến bưu cục nhận

Bức thư được gửi và nhận như thế nào?

- Các bộ phận đồng cấp: Phương tiện và cách thức trao đổi thông tin giống nhau



Trao đổi thông tin giữa các nút mạng

- Dữ liệu được tổ chức như thế nào?
 - Định danh – đánh địa chỉ: Phân biệt các máy với nhau trên mạng?
 - Tìm đường đi cho dữ liệu qua hệ thống mạng như thế nào?
 - Làm thế nào để phát hiện lỗi dữ liệu (và sửa)?
 - Làm thế nào để dữ liệu gửi đi không làm quá tải đường truyền, quá tải máy nhận?
 - Làm thế nào để chuyển dữ liệu thành tín hiệu?
 - Làm thế nào để biết dữ liệu đã tới đích?...
- Phân chia nhiệm vụ cho các thành phần, tổ chức các thành phần thành các tầng (layer)

Phân tầng

- **Mỗi tầng:**

- Có thể có một hoặc nhiều chức năng
- Triển khai dịch vụ để thực hiện các chức năng
 - ❖ Cung cấp dịch vụ cho tầng trên
 - ❖ Sử dụng dịch vụ tầng dưới
 - ❖ Độc lập với các tầng còn lại
- Mỗi dịch vụ có thể có một hoặc nhiều cách triển khai khác nhau, cho phép tầng trên lựa chọn dịch vụ phù hợp

- **Lợi ích:**

- Dễ dàng thiết kế, triển khai
- Dễ dàng tái sử dụng
- Dễ dàng nâng cấp



Không phân tầng



Phân tầng

Điểm truy cập dịch vụ

- Service Access Point: là một khái niệm trừu tượng, tại đó tầng trên sử dụng dịch vụ tầng dưới cung cấp
 - Tầng trên chỉ cần quan tâm cách sử dụng dịch vụ tầng dưới
 - ...không quan tâm tới cách thức thực hiện
- Quan điểm lập trình: cung cấp API (Application Programming Interface)
 - Tên hàm và các thức truyền đối số không đổi
 - Nội dung hàm có thể thay đổi



```
function doMyWork () {  
    //do anything  
    lowerService(parameters) ;  
    //do anything  
}
```

4. Kiến trúc phân tầng

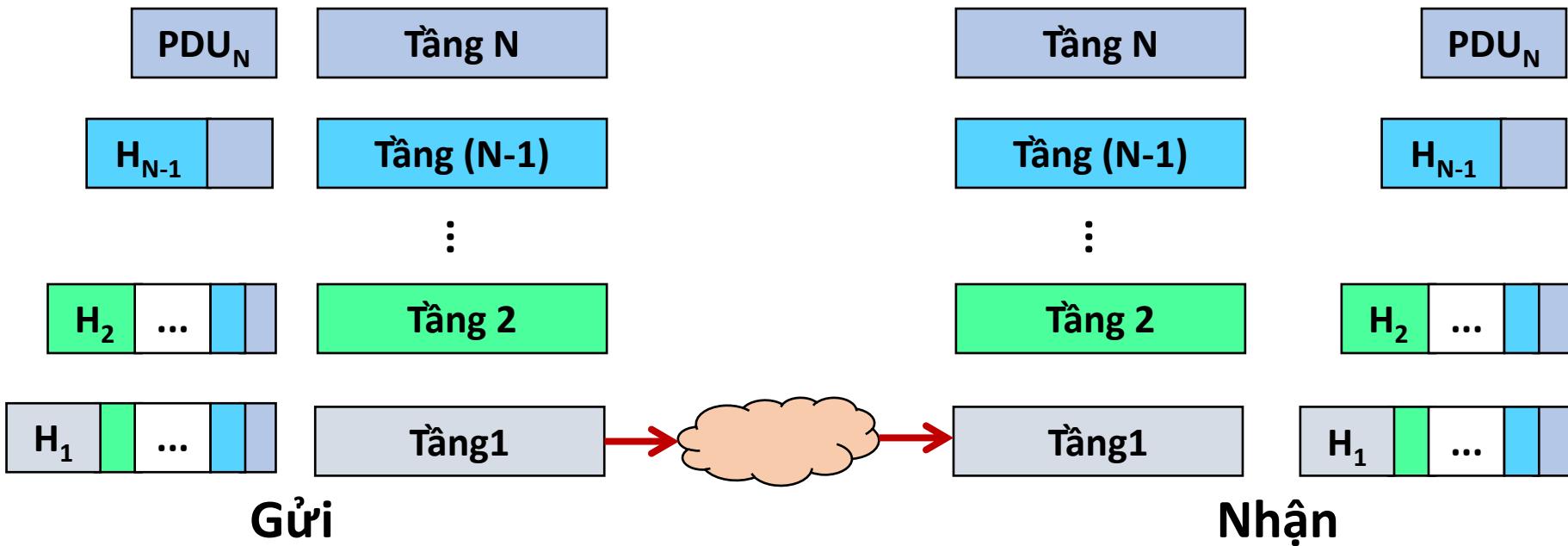
- Mô hình thiết kế phân tầng
- Truyền thông trong kiến trúc phân tầng
- Mô hình OSI và TCP/IP
- Định danh trong TCP/IP
- Tổng kết về phân tầng & chồng giao thức

Truyền thông trong kiến trúc phân tầng

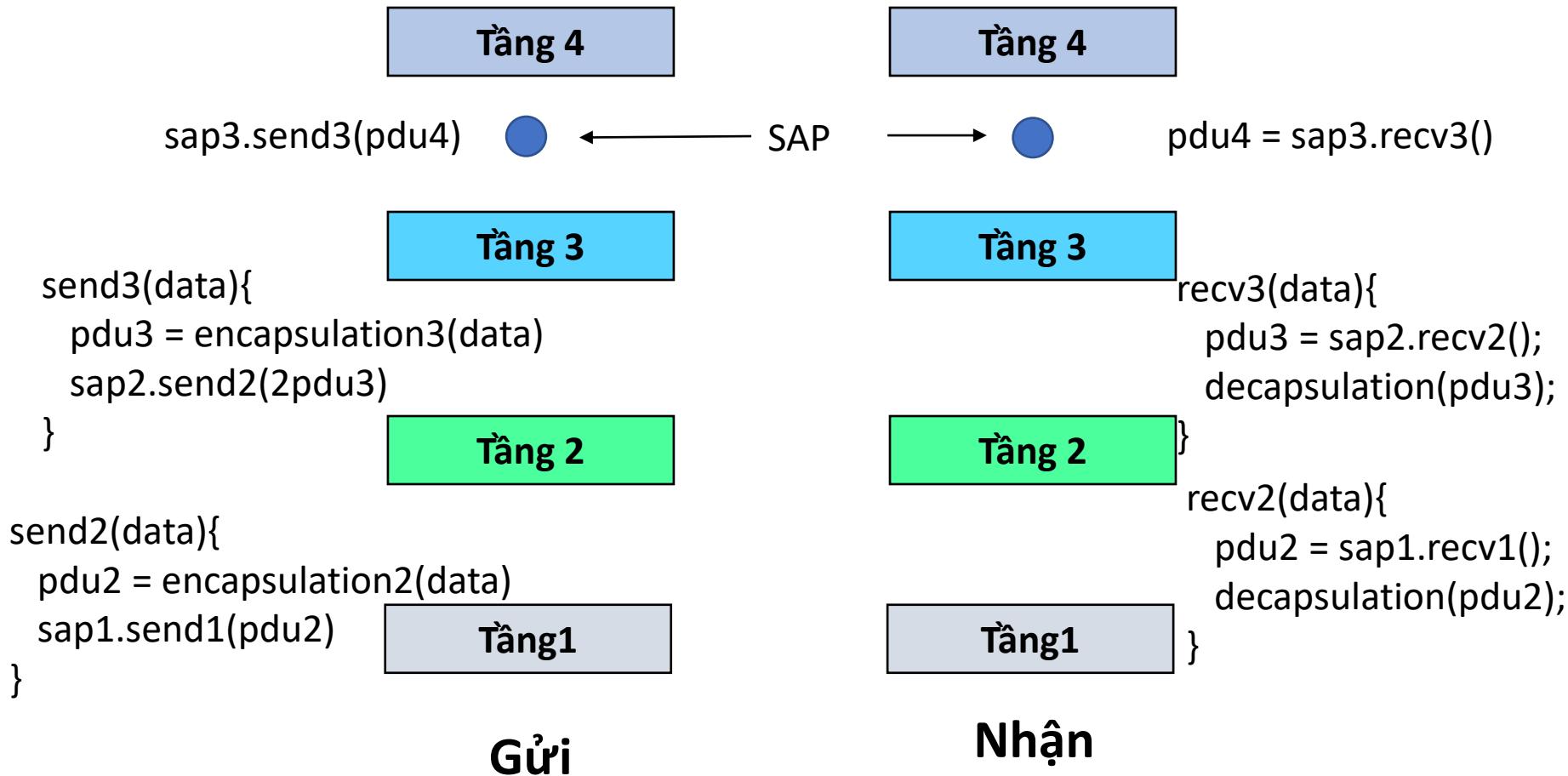
- Các nguyên lý chung:
 - Tầng trên sử dụng dịch vụ tầng dưới
 - Các tầng ngang hàng sử dụng chung “ngôn ngữ” và phương tiện trao đổi dữ liệu
- Dữ liệu được xử lý tại mỗi tầng như thế nào?
 - Chia thành các đơn vị dữ liệu **giao thức** - PDU (**Protocol Data Unit**) gồm có
 - Header: chứa địa chỉ, thông tin khác để hệ thống mạng xử lý
 - Payload: dữ liệu cần truyền tải
 - Chức năng mỗi tầng khác nhau, cách thức xử lý dữ liệu khác nhau → cần phối hợp chức năng giữa các tầng trong quá trình truyền tải

Truyền thông trong kiến trúc phân tầng

- Bên gửi: thêm tiêu đề chứa thông tin phục vụ cho việc xử lý dữ liệu tại tầng tương ứng và chuyển cho tầng dưới (Đóng gói dữ liệu – Encapsulation)
- Bên nhận: xử lý dữ liệu theo thông tin trong phần tiêu đề, tách tiêu đề và chuyển dữ liệu cho tầng trên



SAP Invocation & Implementation



Truyền thông trong kiến trúc phân tầng (tiếp)

- Nhận xét:

- PDU tại các tầng đồng cấp của hai bên giống nhau → truyền thông giữa các tầng ngang hàng (truyền thông logic)
 - Phía nhận phải hiểu nội dung PDU của phía gửi
 - Phía nhận xử lý PDU nhận được với các tham số là thông tin trong tiêu đề mà phía gửi đã thiết lập
 - Phía nhận trả lời/không trả lời cho phía gửi
 - Các PDU phải truyền đúng theo thứ tự
- cần có bộ quy tắc cho hai bên

Giao thức (Network protocol)

Là tập hợp các quy tắc quy định khuôn dạng, ngữ nghĩa, thứ tự các thông điệp được gửi và nhận giữa các nút mạng và các hành vi khi trao đổi các thông điệp đó

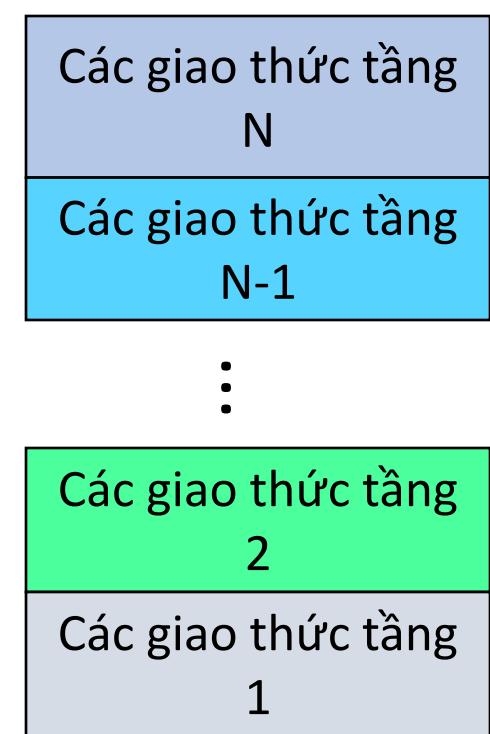
Chồng giao thức (Protocol stack)

- Các chức năng được phân chia cho các tầng
- Mỗi tầng có nhiều cách thức để thực hiện các chức năng → sinh ra các giao thức khác nhau

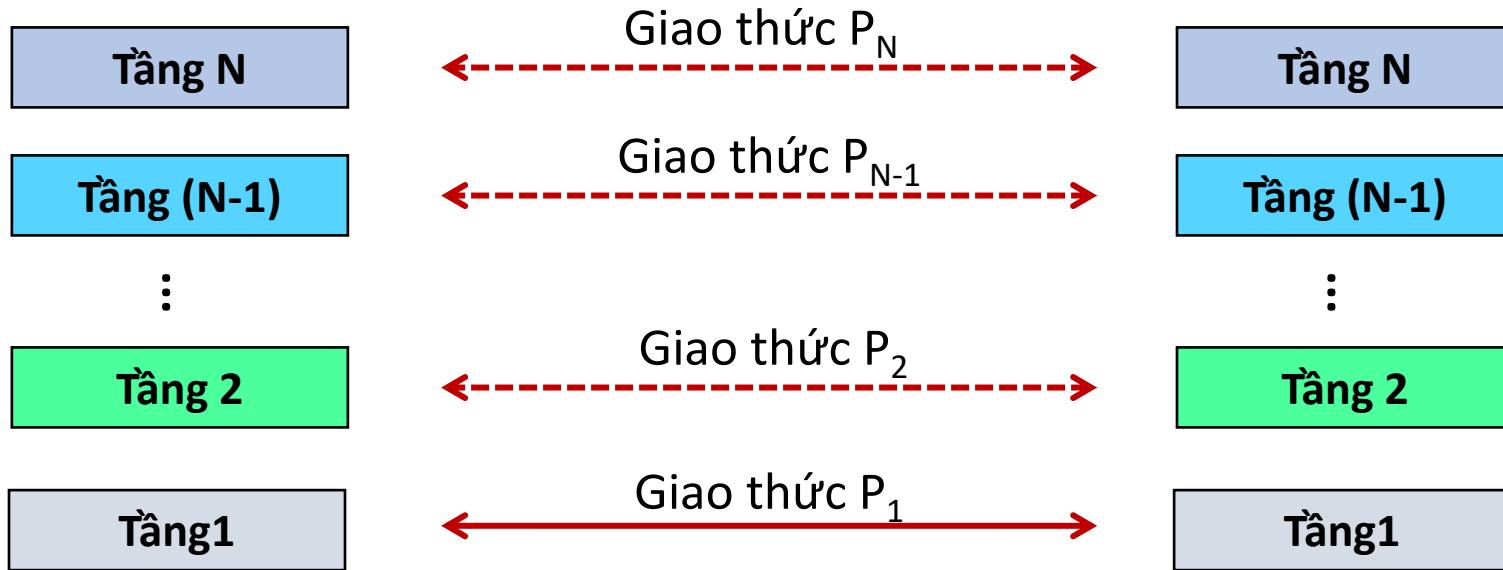
→chồng giao thức: ngăn xếp các giao thức truyền thông trên kiến trúc phân tầng

→Giao thức mỗi tầng bao gồm:

- Gọi dịch vụ nào của giao thức tầng dưới
- Và cung cấp dịch vụ cho giao thức tầng trên như thế nào



Truyền thông trong kiến trúc phân tầng (tiếp)



- Các tầng đồng cấp ở mỗi bên sử dụng chung giao thức để điều khiển quá trình truyền thông logic giữa chúng
 - 2 cách thức để giao thức điều khiển truyền thông logic giữa các tầng đồng cấp: hướng liên kết hoặc hướng không liên kết

Truyền thông hướng liên kết vs Truyền thông hướng không liên kết

- Truyền thông hướng liên kết (connection oriented):
 - Dữ liệu được truyền qua một liên kết đã được thiết lập
 - Ba giai đoạn: Thiết lập liên kết, Truyền dữ liệu, Hủy liên kết
 - Tin cậy
- Truyền thông hướng không liên kết (conectionless)
 - Không thiết lập liên kết, chỉ có giai đoạn truyền dữ liệu
 - Không tin cậy
 - “Best effort”: truyền ngay với khả năng tối đa

Giao thức Unicast, Multicast, Broadcast

- Unicast: giao thức điều khiển truyền dữ liệu tới 1 đích
- Multicast: giao thức điều khiển truyền dữ liệu tới nhiều đích
- Broadcast: giao thức điều khiển truyền dữ liệu tới mọi đích

4. Kiến trúc phân tầng

- Mô hình thiết kế phân tầng
- Truyền thông trong kiến trúc phân tầng
- Mô hình OSI và TCP/IP
- Định danh trong TCP/IP
- Tổng kết về phân tầng & chồng giao thức

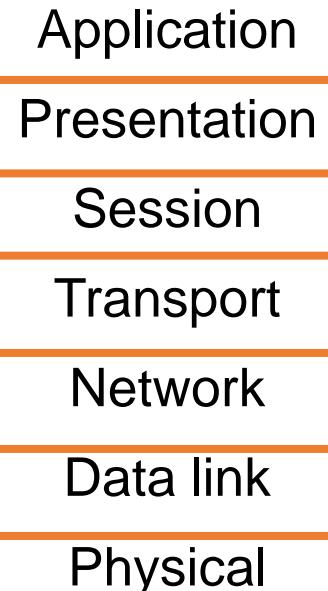
Kiến trúc phân tầng trên thực tế (Bao nhiêu tầng? Chức năng cụ thể?...)

Kiến trúc phân tầng triển khai trên các nút mạng như thế nào?

Mô hình OSI/ISO

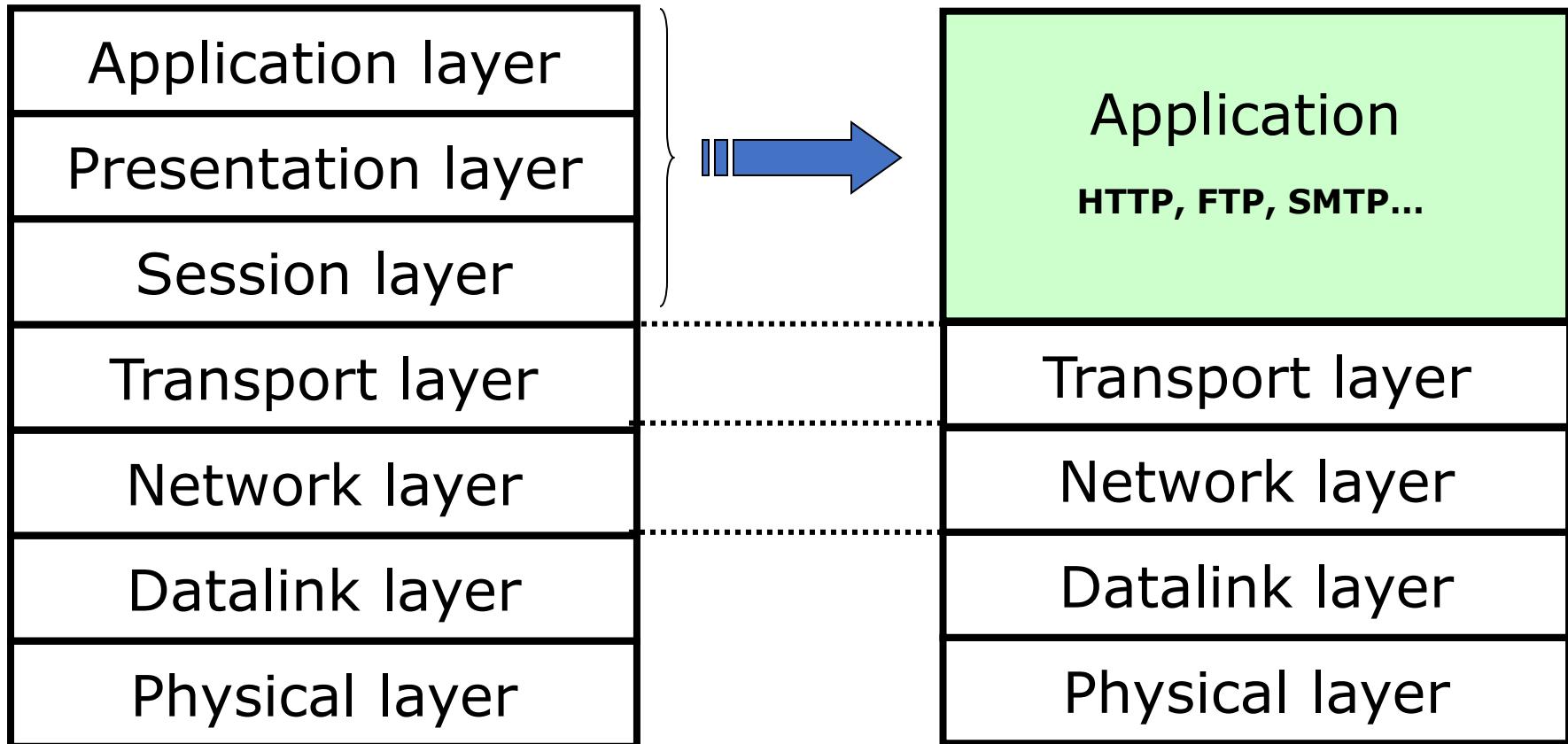
Open Systems Interconnection by ISO

- **Tầng Ứng dụng (Application):** cung cấp các ứng dụng trên mạng (web, email, truyền file...)
- **Tầng Trình diễn (Presentation):** biểu diễn dữ liệu của ứng dụng, e.g., mã hóa, nén, chuyển đổi...
- **Tầng Phiên(Session):** quản lý phiên làm việc, đồng bộ hóa phiên, khôi phục quá trình trao đổi dữ liệu
- **Tầng Giao vận (Transport):** Xử lý việc truyền-nhận dữ liệu cho các ứng dụng chạy trên nút mạng đầu-cuối
- **Tầng Mạng (Network):** Chọn đường (định tuyến), chuyển tiếp gói tin từ nguồn đến đích
- **Tầng Liên kết dữ liệu (Data link):** Truyền dữ liệu trên các liên kết vật lý giữa các nút mạng kế tiếp nhau
- **Tầng Vật lý (Physical):** Chuyển dữ liệu (bit) thành tín hiệu và truyền



Mô hình OSI và TCP/IP

Trong mô hình TCP/IP (Internet), chức năng 3 tầng trên được phân định cho một tầng duy nhất



Mô hình OSI và TCP/IP (tiếp)

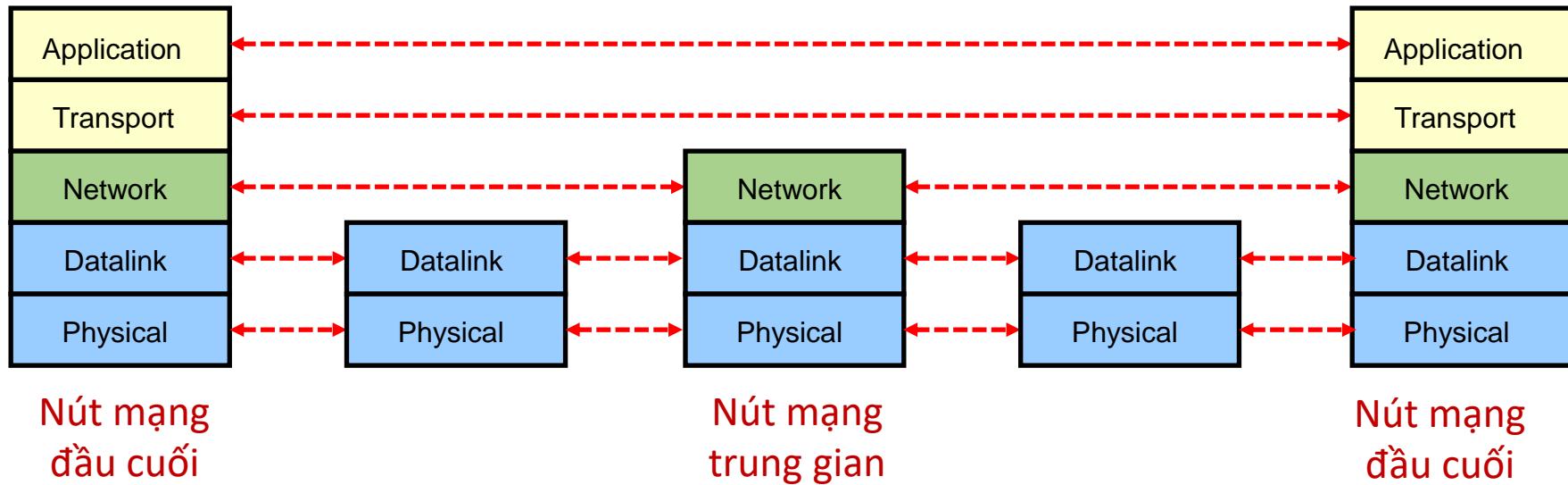
- Mô hình OSI:

- Mô hình **tham chiếu** chức năng: Các mô hình khác phải tham chiếu từ mô hình OSI
 - ❖ Cung cấp đầy đủ các chức năng mô hình OSI đã chỉ ra
 - ❖ Đảm bảo thứ tự các tầng chức năng
- Có ý nghĩa lớn về mặt cơ sở lý thuyết
- Không sử dụng trên thực tế

- Mô hình TCP/IP: mô hình Internet

- Sử dụng trên hầu hết các hệ thống mạng

Triển khai kiến trúc phân tầng

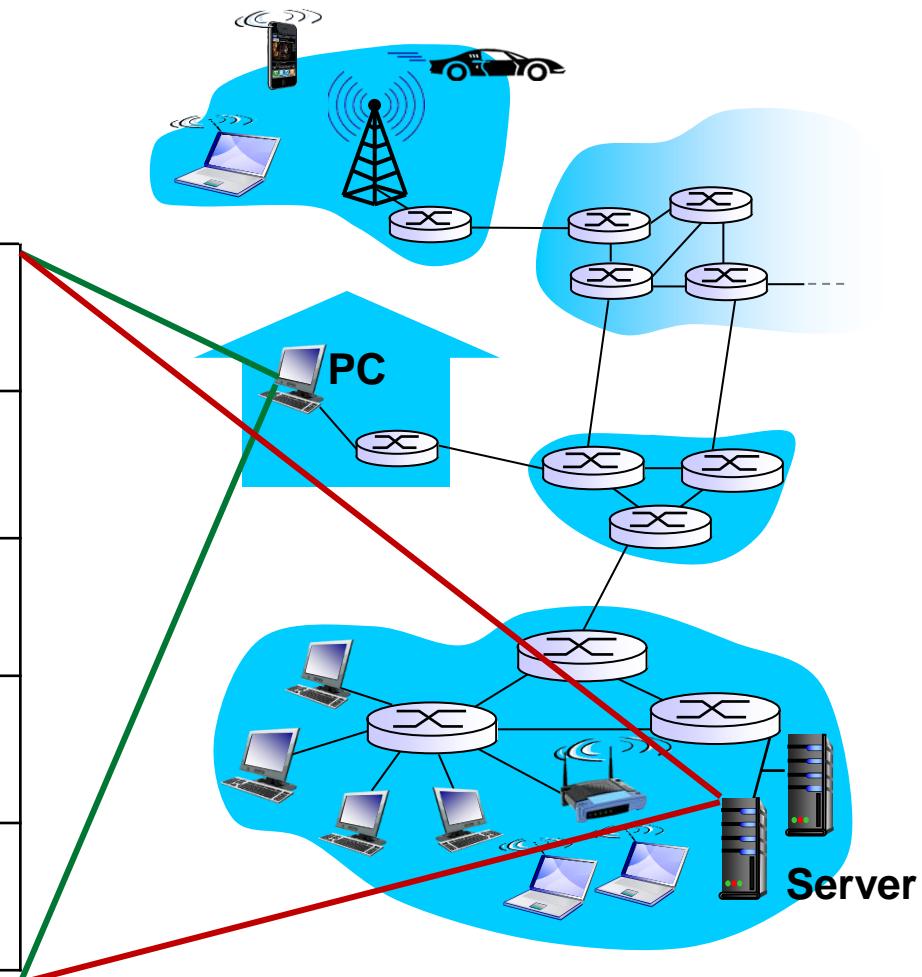


- Nút mạng đầu cuối (end-system): PC, server, smartphone...
- Nút mạng trung gian: các thiết bị mạng chuyển tiếp dữ liệu (Hub, Switch, Router)

Triển khai kiến trúc phân tầng

Nút mạng đầu cuối (server, PC, smartphone...)

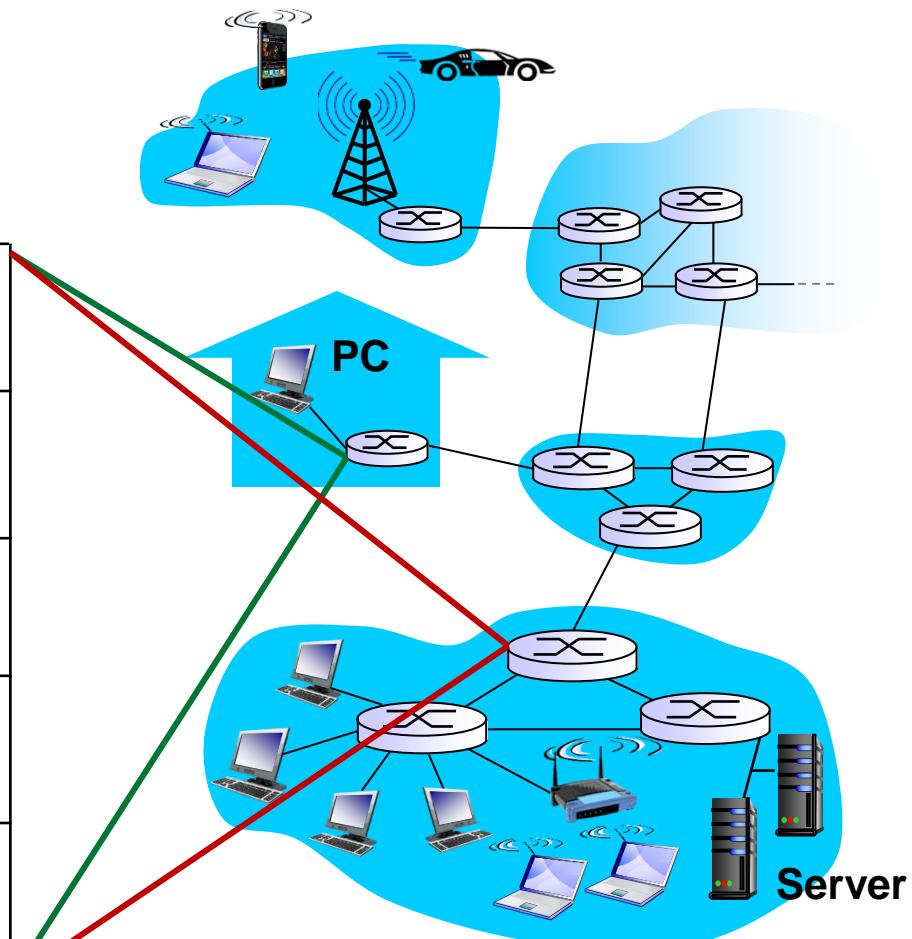
	Ứng dụng mạng cung cấp dịch vụ cho người dùng
	Điều khiển truyền dữ liệu giữa các ứng dụng
	Chọn đường, chuyển tiếp dữ liệu
	Điều khiển truyền dữ liệu trên các liên kết vật lý
	Chuyển dữ liệu thành tín hiệu và truyền đi



Triển khai kiến trúc phân tầng

Nút router trung gian

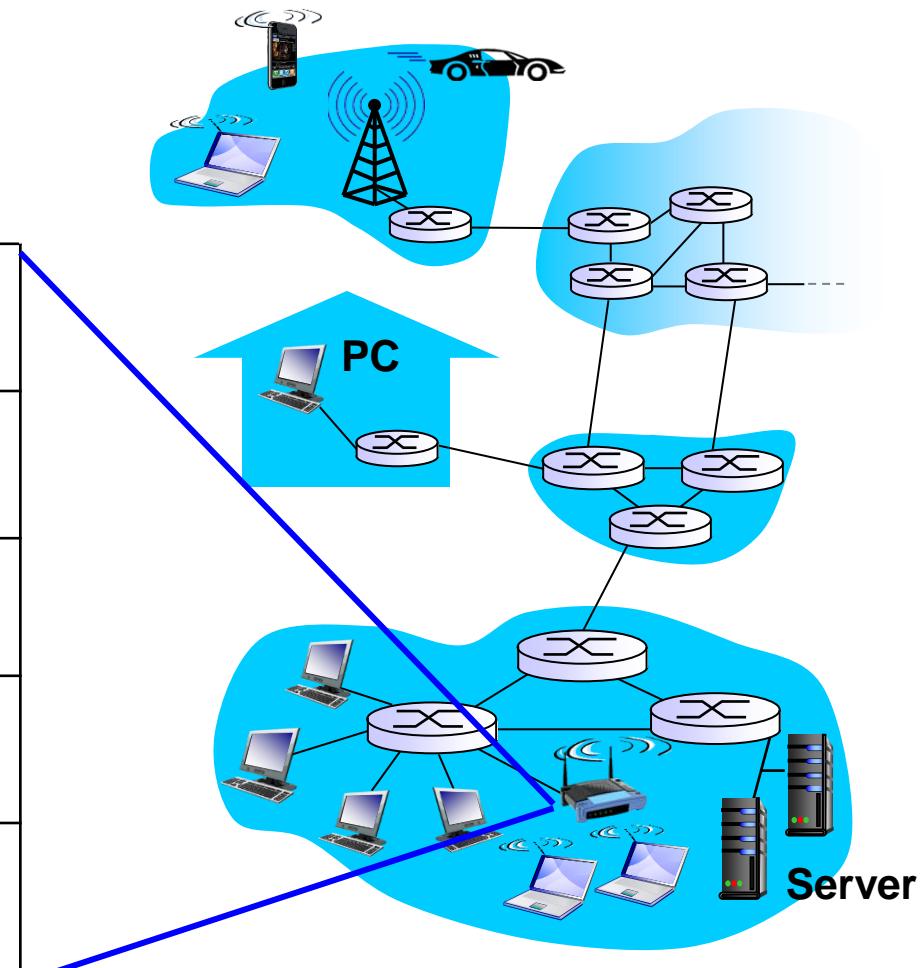
	Ứng dụng mạng cung cấp dịch vụ cho người dùng
	Điều khiển truyền dữ liệu giữa các ứng dụng
	Chọn đường, chuyển tiếp dữ liệu
	Điều khiển truyền dữ liệu trên các liên kết vật lý
	Chuyển dữ liệu thành tín hiệu và truyền đi



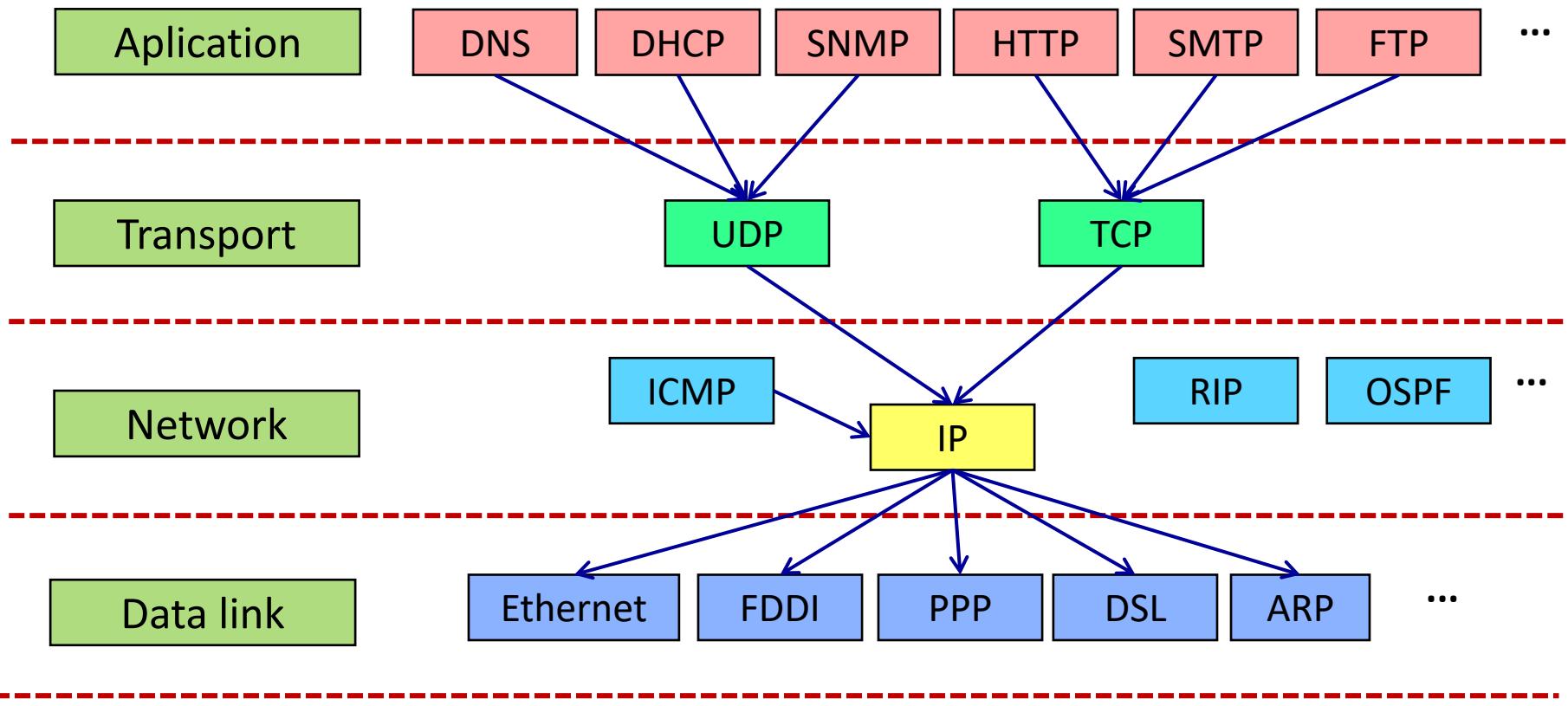
Triển khai kiến trúc phân tầng

Nút switch trung gian

	Ứng dụng mạng cung cấp dịch vụ cho người dùng
	Điều khiển truyền dữ liệu giữa các ứng dụng
	Chọn đường, chuyển tiếp dữ liệu
	Điều khiển truyền dữ liệu trên các liên kết vật lý
	Chuyển dữ liệu thành tín hiệu và truyền đi



Chồng giao thức TCP/IP



Nhớ lại “Internet & lịch sử phát triển”:
Sử dụng duy nhất một giao thức liên mạng là IP

Chồng giao thức TCP/IP

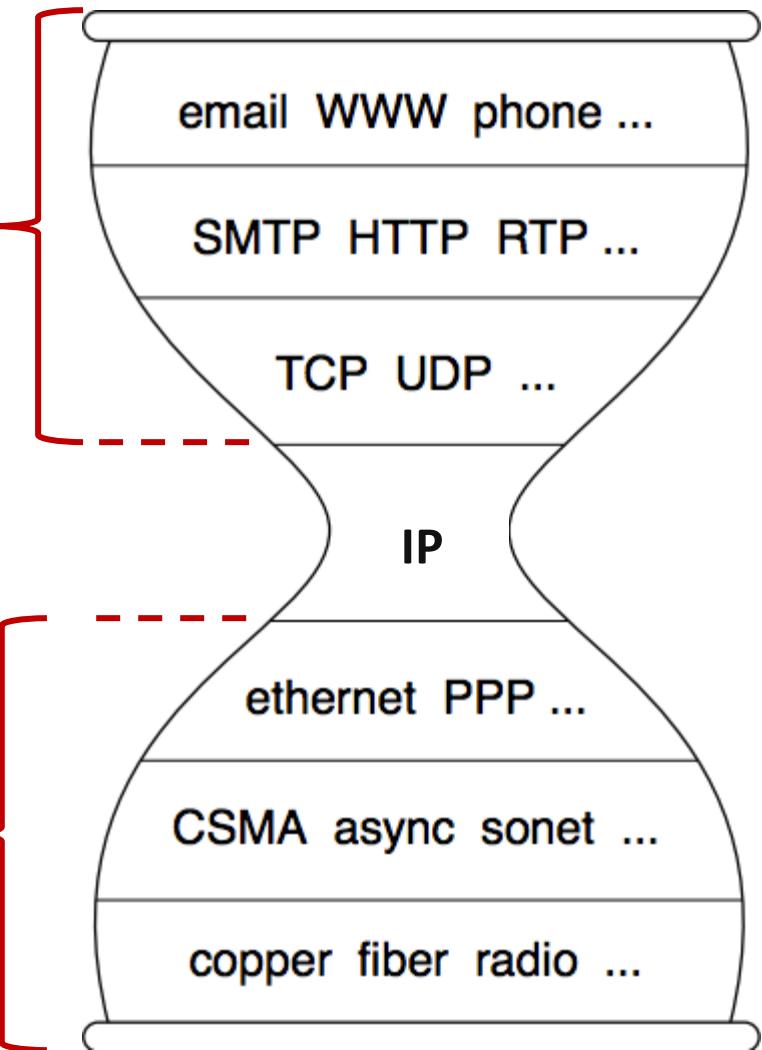
- Dạng “đồng hồ cát”: sử dụng duy nhất một giao thức liên mạng (IP – Internet Protocol) tại tầng mạng:
 - Cho phép một hệ thống mạng mới sử dụng công nghệ truyền dẫn bất kỳ kết nối với hệ thống mạng hiện tại
 - Tách rời phát triển ứng dụng ở tầng cao với công nghệ truyền dẫn các tầng thấp
 - IP-based application: Ứng dụng trên nền tảng IP (VoIP...)
 - Hỗ trợ thay đổi song song các công nghệ ở trên và dưới IP
- Tuy nhiên, rất khó để nâng cấp bản thân giao thức IP (vấn đề chuyển đổi IPv4 sang IPv6)

Cài đặt TCP/IP trên hệ thống mạng

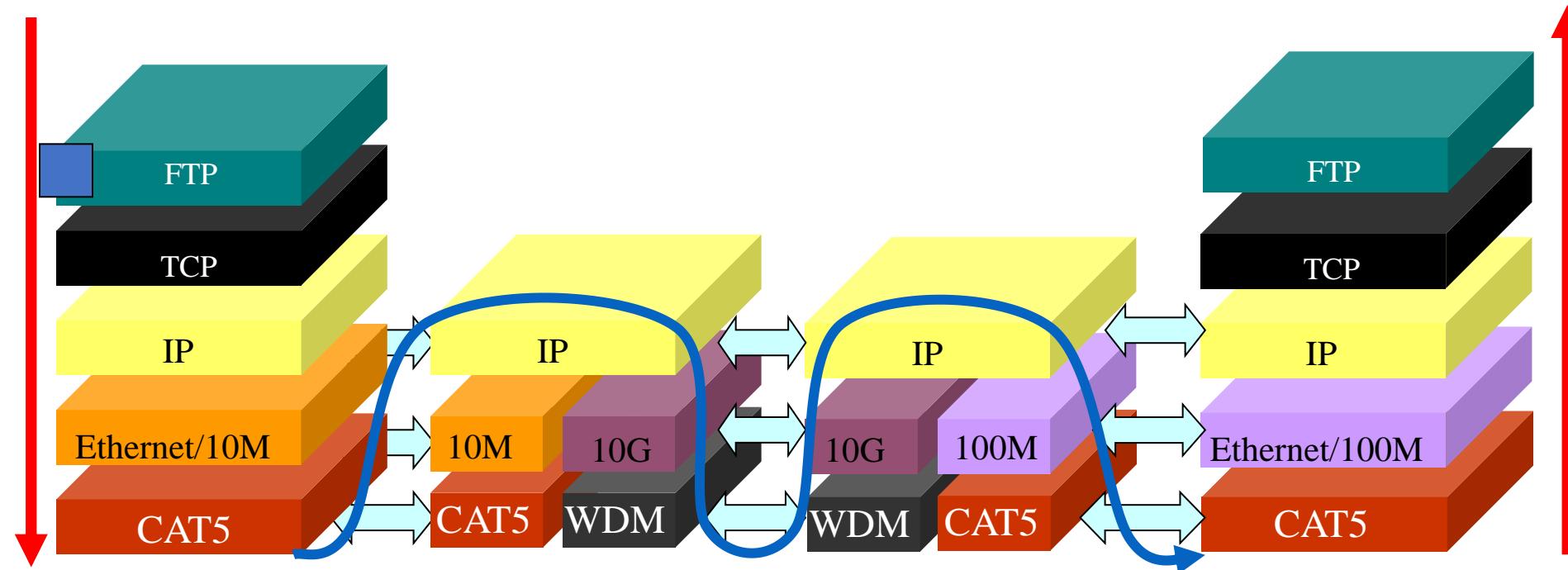
- Trên các hệ thống đầu cuối
- Khác nhau trên các ứng dụng khác nhau

- Như nhau trên mọi nút

- Trên mọi nút
- Khác nhau trên các liên kết khác nhau



Đóng gói trên chồng giao thức TCP/IP



Nút mạng
đầu cuối

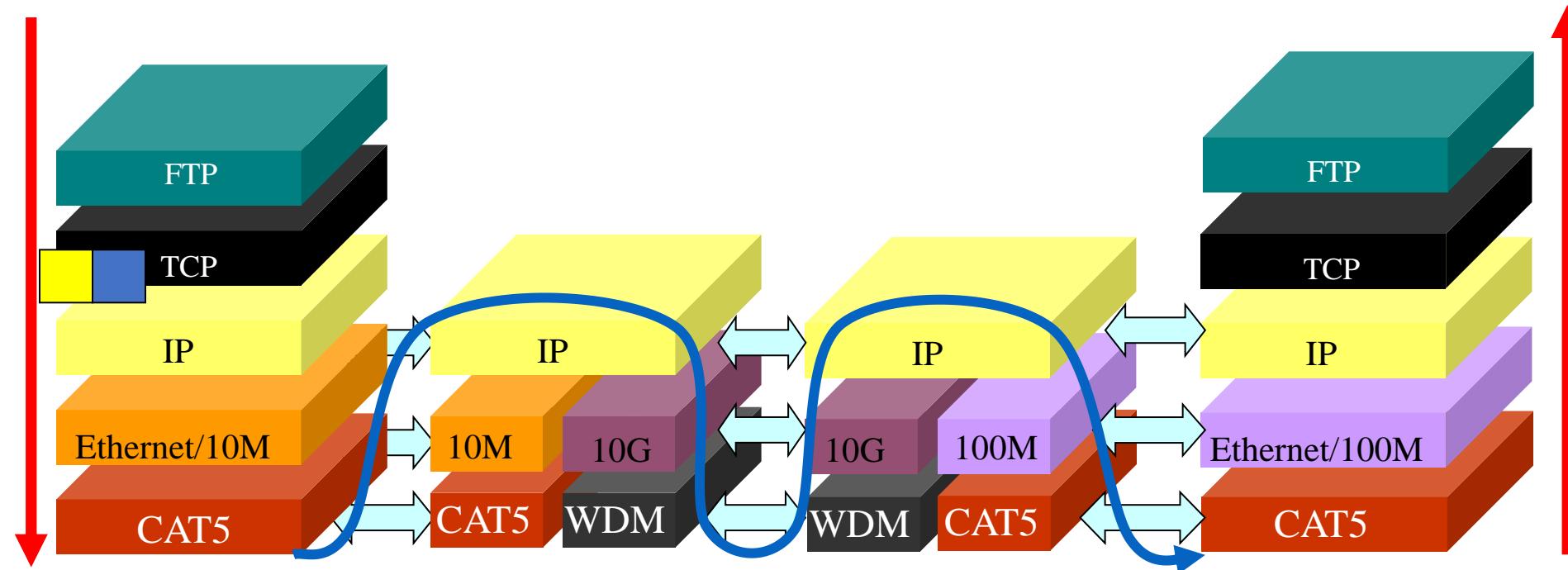
Các nút mạng trung gian

Nút mạng
đầu cuối

Dữ liệu - payload

CAT5: https://en.wikipedia.org/wiki/Category_5_cable
WDM: https://en.wikipedia.org/wiki/Wavelength-division_multiplexing

Đóng gói trên ch่อง giao thức TCP/IP



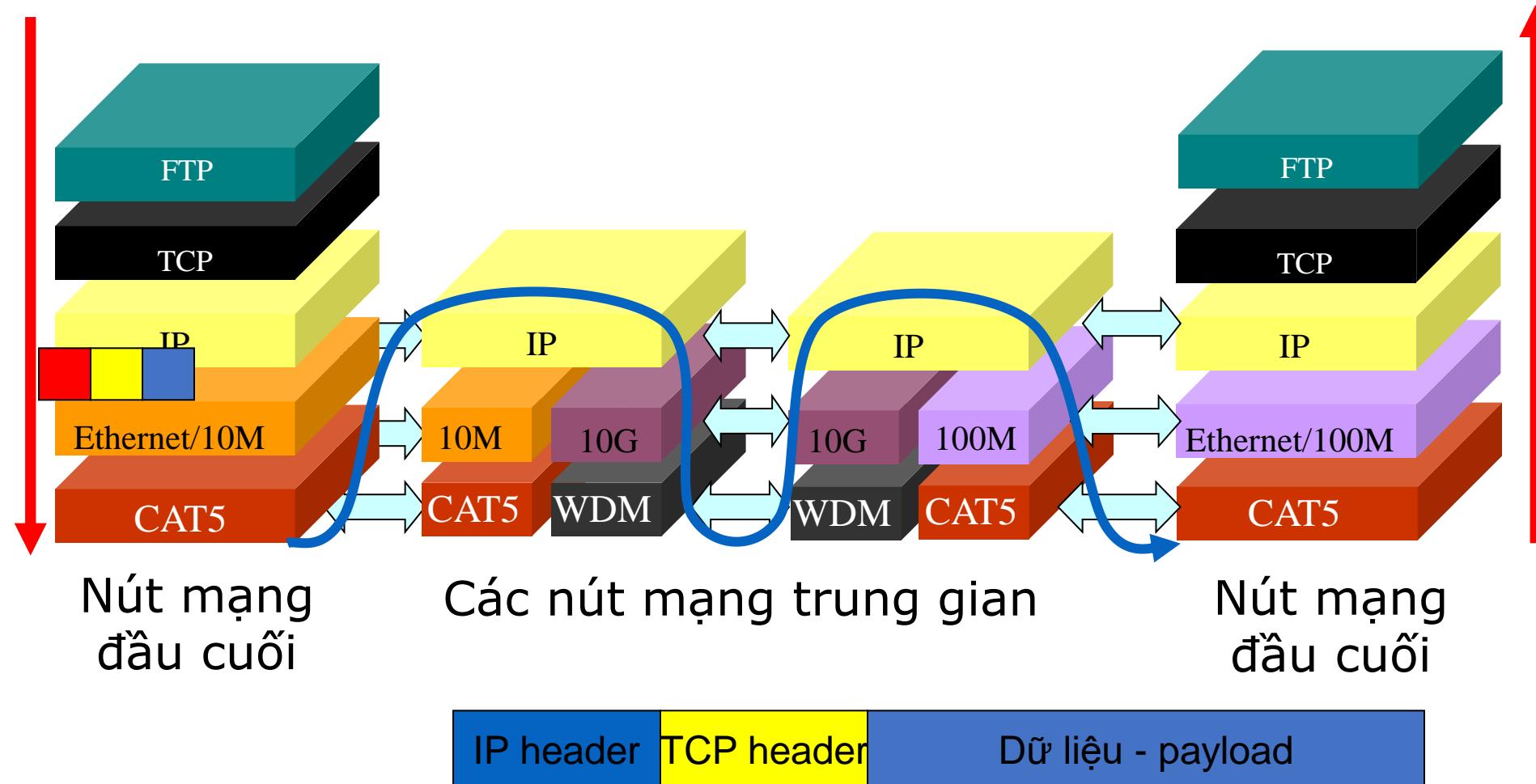
Nút mạng
đầu cuối

Các nút mạng trung gian

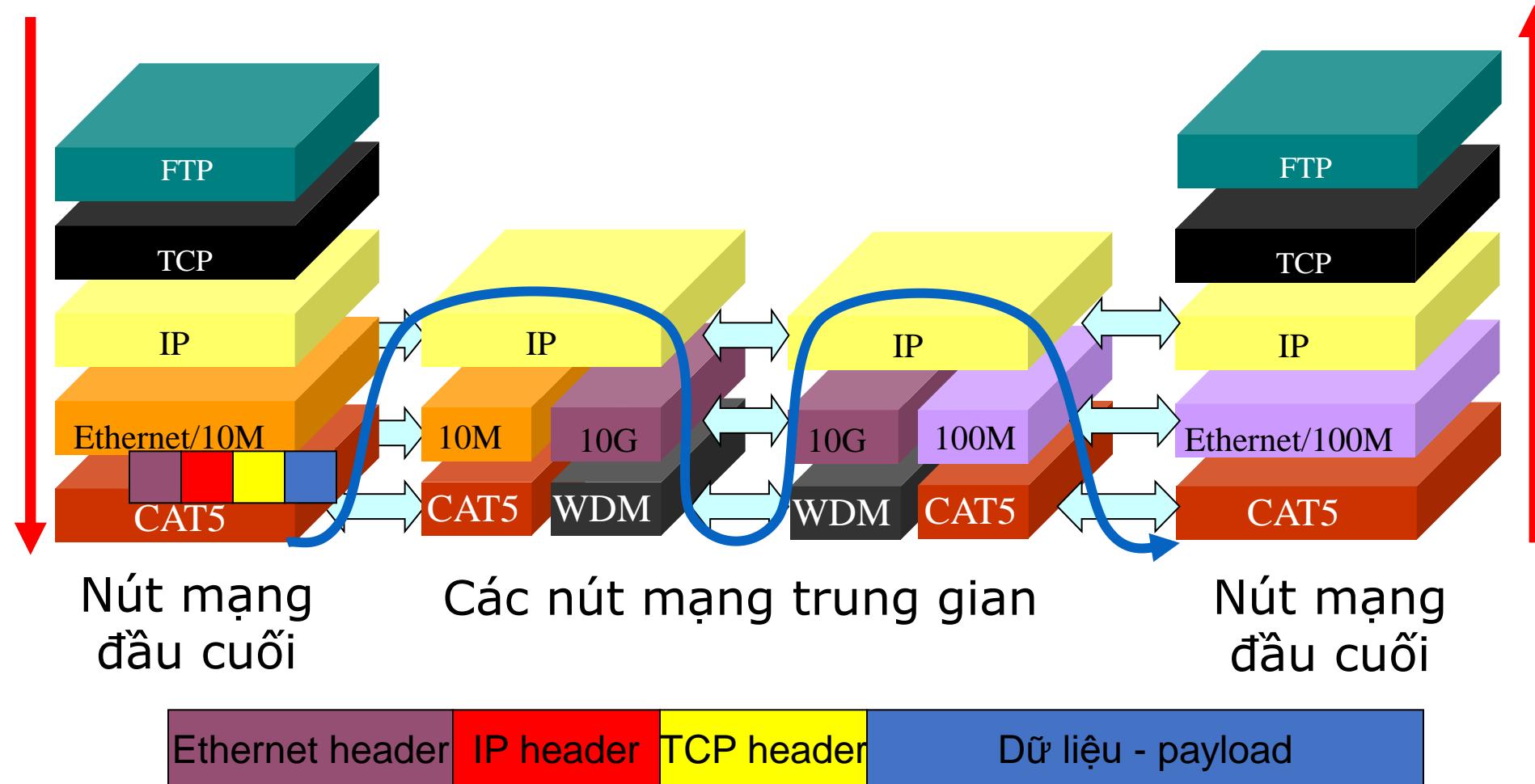
Nút mạng
đầu cuối



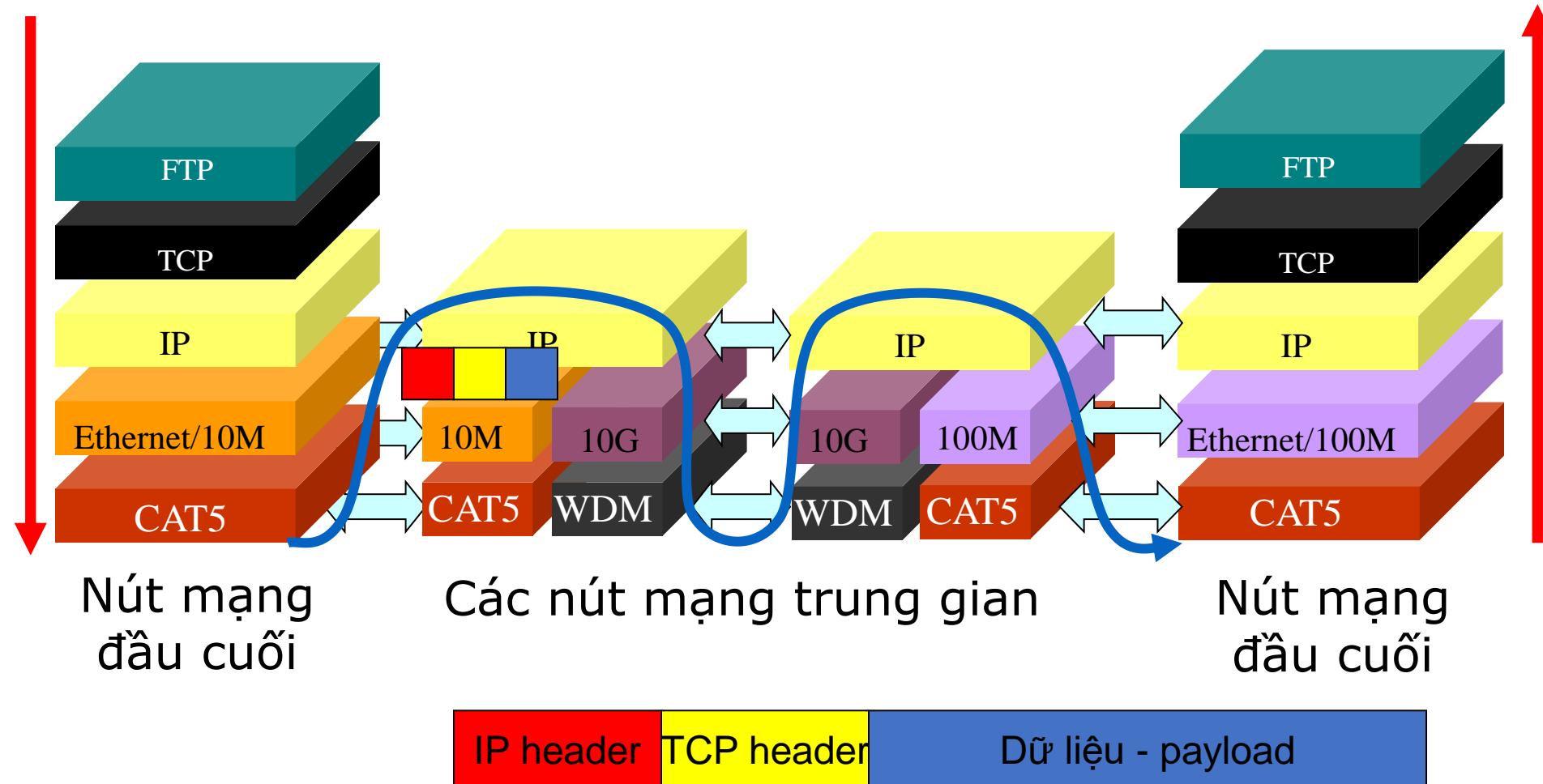
Đóng gói trên ch่อง giao thức TCP/IP



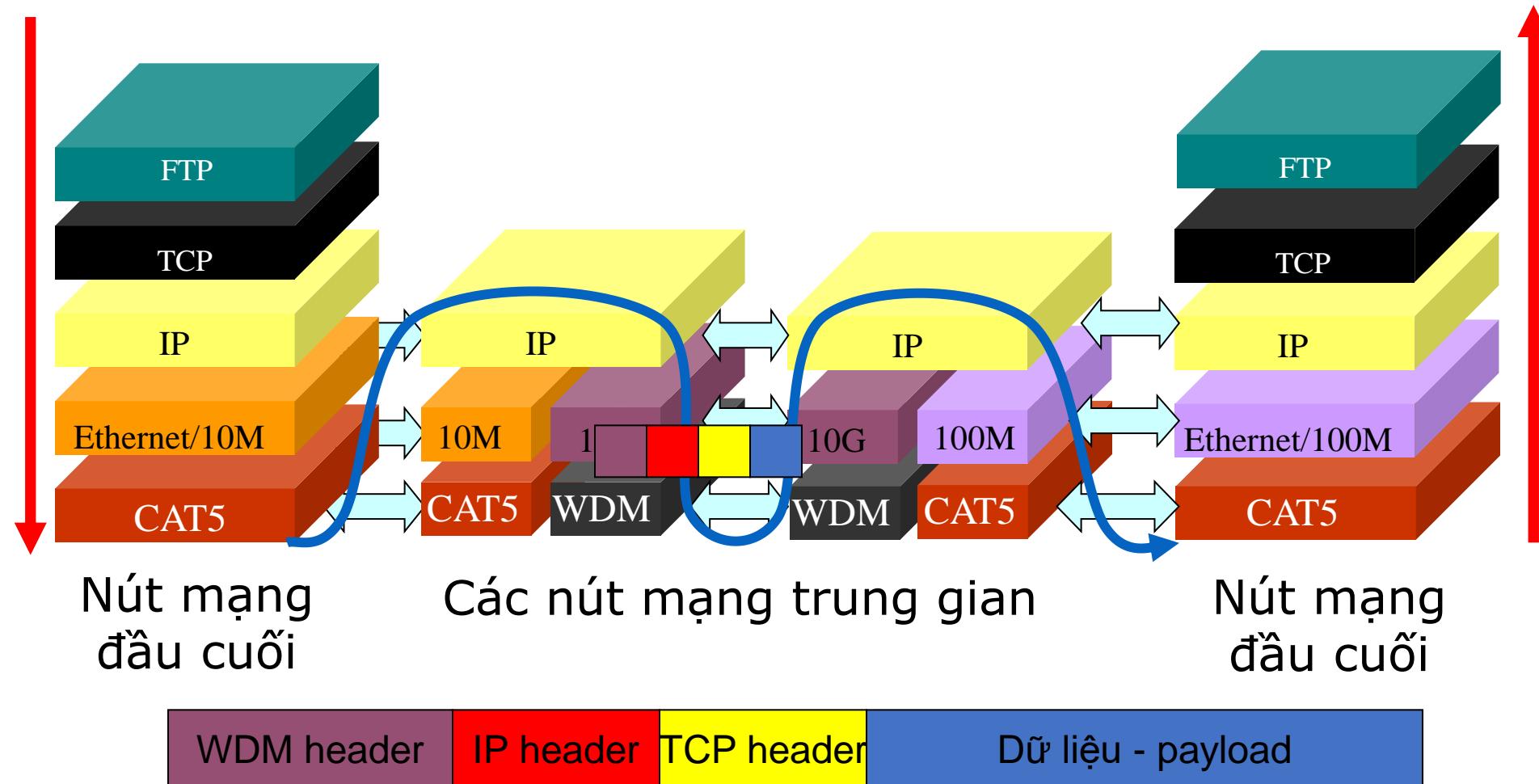
Đóng gói trên chồng giao thức TCP/IP



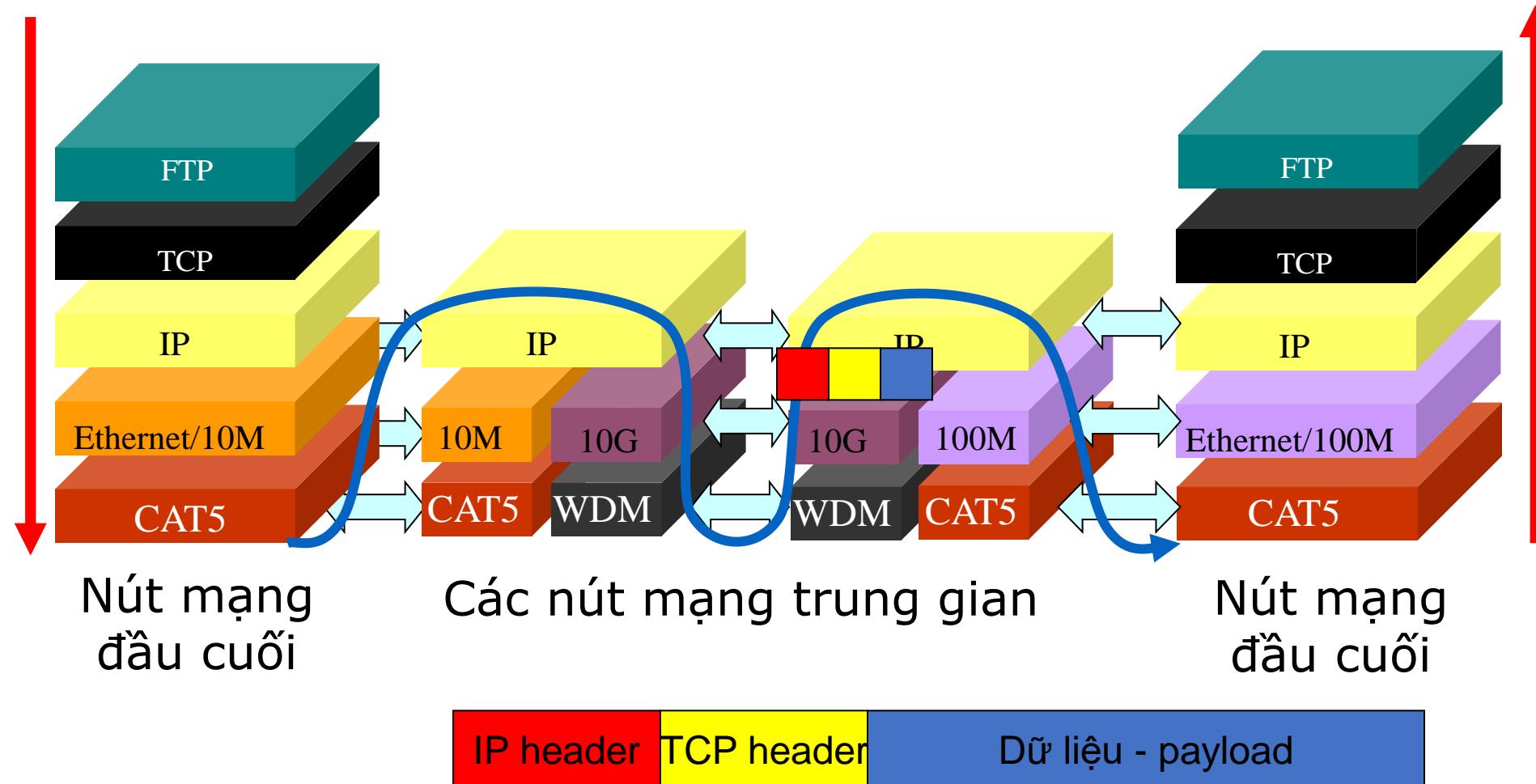
Đóng gói trên ch่อง giao thức TCP/IP



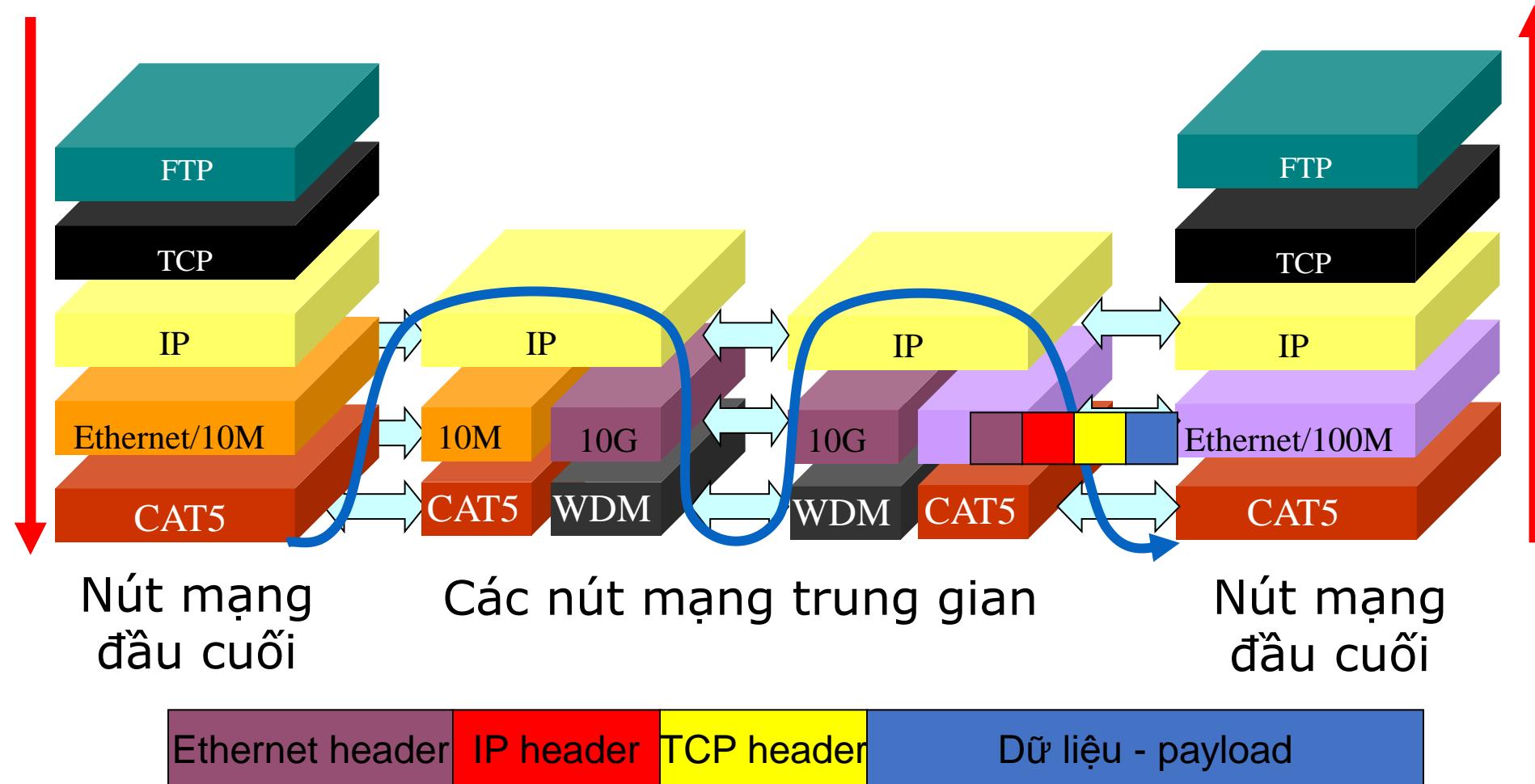
Đóng gói trên chồng giao thức TCP/IP



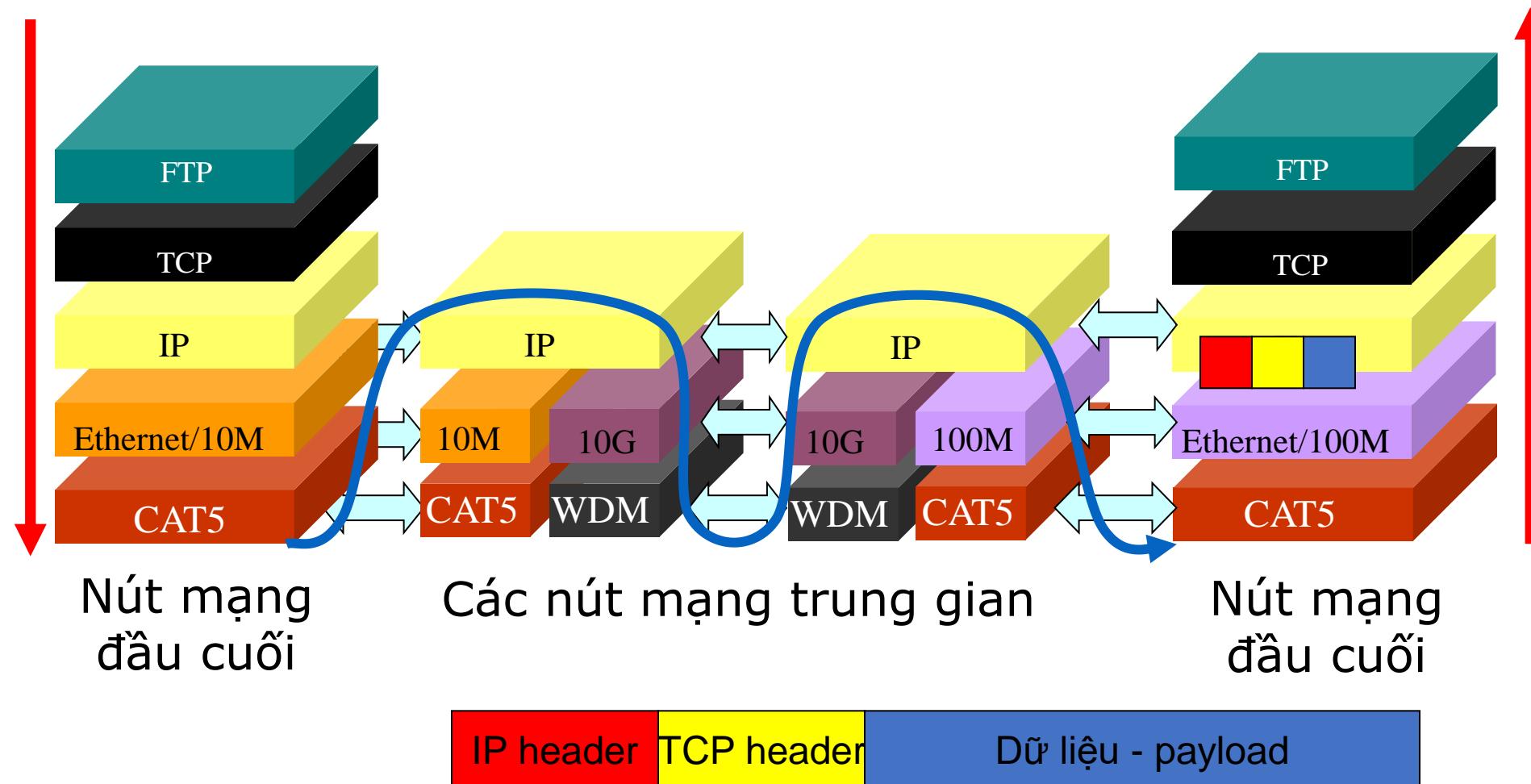
Đóng gói trên ch่อง giao thức TCP/IP



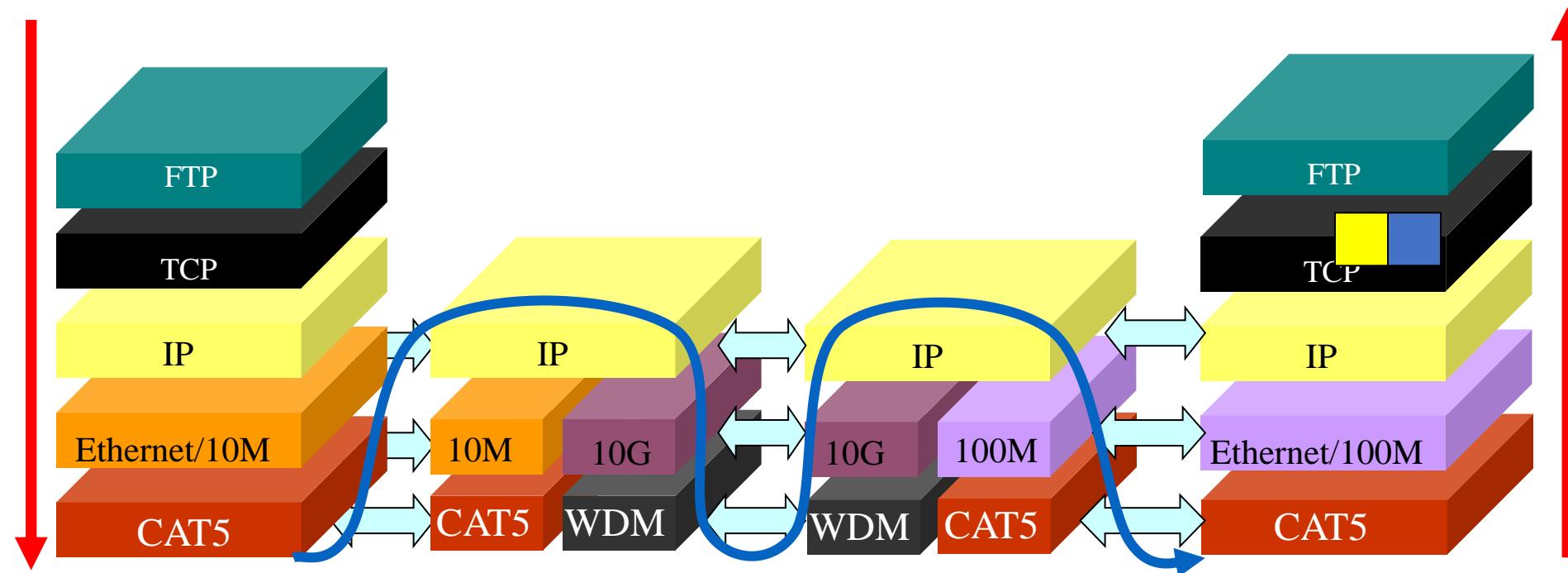
Đóng gói trên chồng giao thức TCP/IP



Đóng gói trên ch่อง giao thức TCP/IP



Đóng gói trên ch่อง giao thức TCP/IP



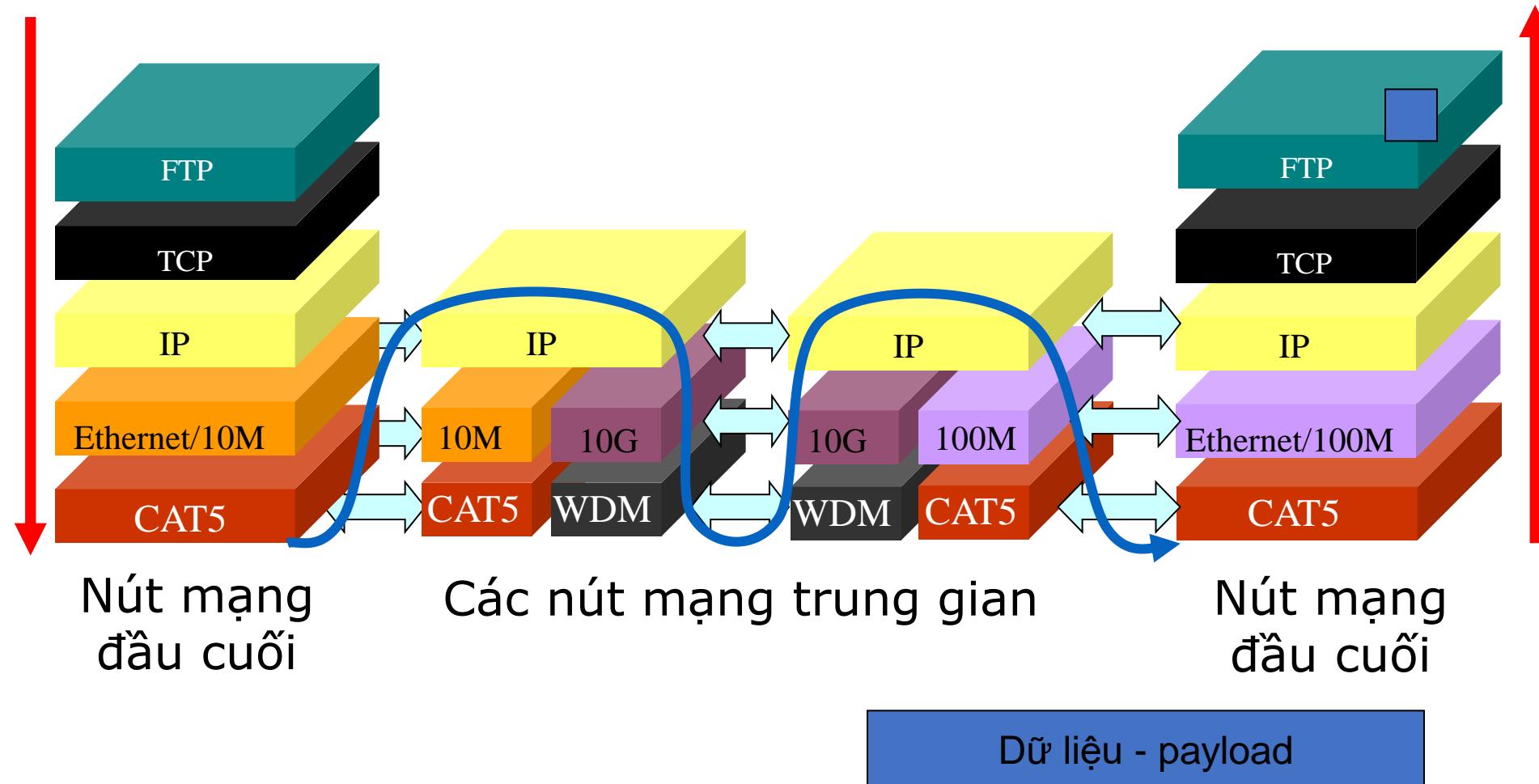
Nút mạng
đầu cuối

Các nút mạng trung gian

Nút mạng
đầu cuối



Đóng gói trên chồng giao thức TCP/IP



4. Kiến trúc phân tầng

- Mô hình thiết kế phân tầng
- Truyền thông trong kiến trúc phân tầng
- Mô hình OSI và TCP/IP
- **Định danh trong TCP/IP**
- Tổng kết về phân tầng & chồng giao thức

Tên miền

Số hiệu cổng

Địa chỉ IP

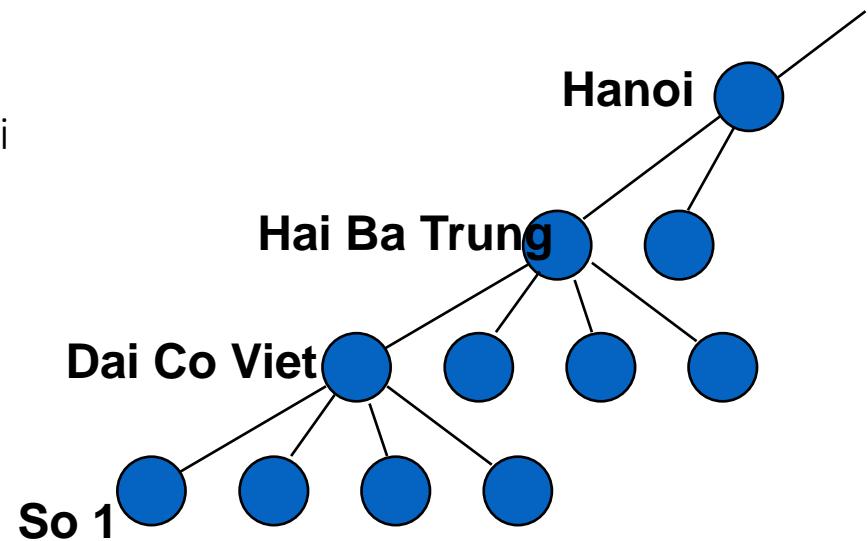
Địa chỉ MAC

Định danh

- Giá trị cho phép xác định một người hay một đối tượng
 - Tên
 - ❖ Bui Trong Tung
 - Địa chỉ
 - ❖ 1 Dai Co Viet, Hai Ba Trung, Ha Noi
 - Số điện thoại
 - ❖ 8680896
 - Email
 - ❖ tungbt@soict.hust.edu.vn

Định danh và cây phân cấp

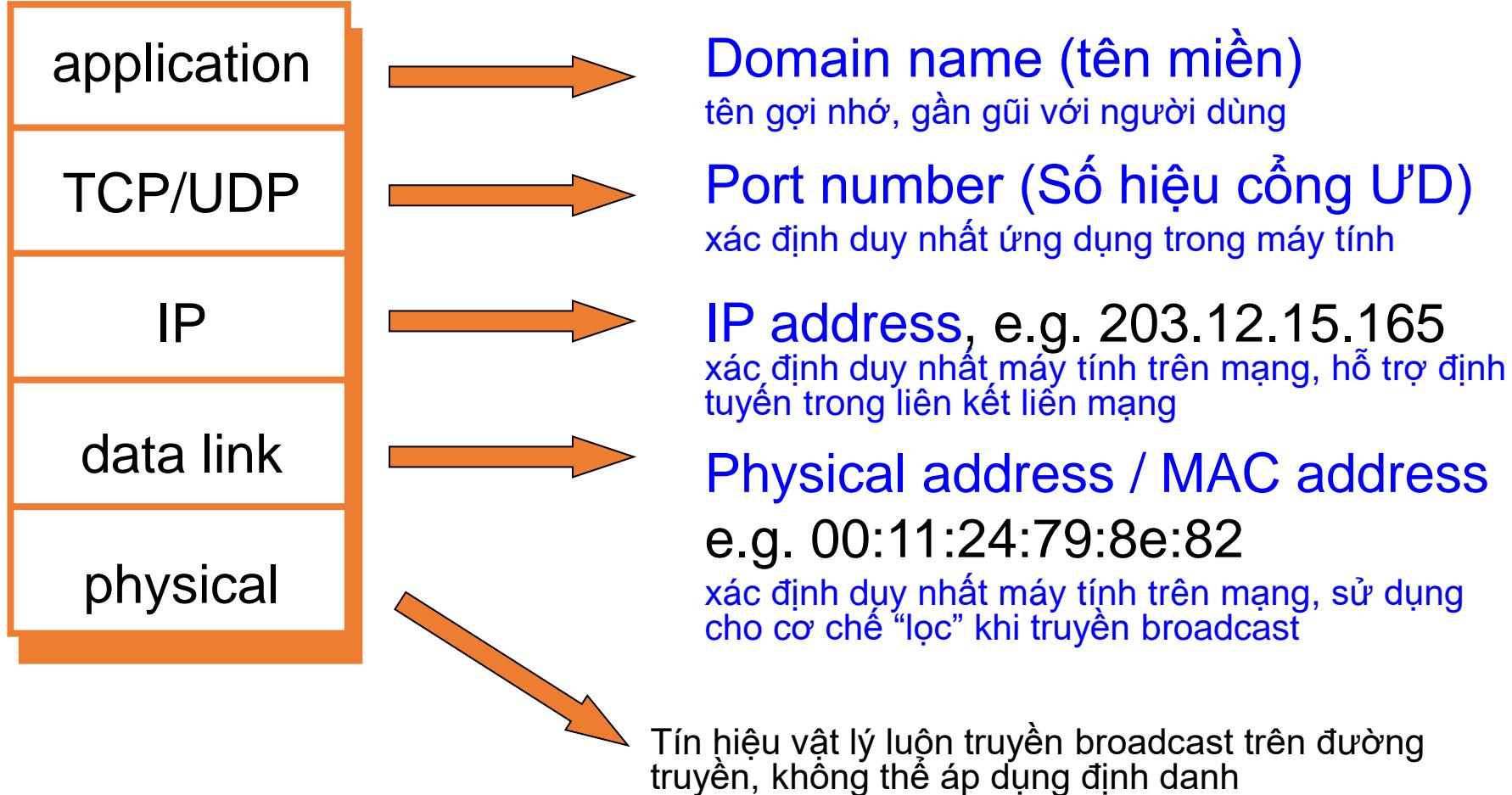
- Các định danh xác định địa chỉ có tính phân cấp
 - Cho phép quản lý một cách logic và hiệu quả một không gian địa chỉ khổng lồ
 - Tính mở rộng
- Ví dụ về tính phân cấp
 - Địa chỉ
 - 1 Dai Co Viet, Hai Ba Trung, Ha Noi
 - Số điện thoại
 - +84-(4) 868-08-96



Định danh trên kiến trúc phân tầng

- Định danh trong hệ thống mạng máy tính: gán cho mỗi đối tượng (dịch vụ, máy trạm, thiết bị mạng) một giá trị riêng.
- Tại sao phải định danh?
 - Phân biệt các đối tượng trong hệ thống
 - Xác định dữ liệu xuất phát từ đâu
 - Xác định dữ liệu đi đến đâu
- Mỗi tầng có nhiệm vụ khác nhau để điều khiển việc truyền thông tin giữa những đối tượng khác nhau → mỗi tầng có cơ chế định danh khác nhau
 - Cùng một đối tượng có thể mang nhiều định danh → có thể cần cơ chế “phân giải” để tìm kiếm một định danh của đối tượng trên tầng này khi biết định danh của đối tượng đó ở tầng khác

Định danh trên TCP/IP



Tên miền (Domain name)

- Định danh sử dụng trên tầng ứng dụng
- Là một chuỗi ký tự gợi nhớ
- Do người dùng sử dụng khi truy cập dịch vụ trên tầng ứng dụng
- Không có ý nghĩa khi truyền dữ liệu giữa các nút mạng
- Phân cấp
- Ví dụ: soict.hust.edu.vn

Số hiệu cổng ứng dụng

- Định danh sử dụng trên tầng giao vận
 - 16 bit
 - Một chỉ số phụ, dùng kèm theo địa chỉ IP
 - Các ứng dụng được định danh bởi một địa chỉ IP và một số hiệu cổng
 - Tương tự như số phòng trong một tòa nhà
 - ❖ Địa chỉ nhà : Nhà C1, 1 Dai Co Viet, Ha Noi => Địa chỉ IP
 - ❖ Phòng số 325 => Số hiệu cổng
- E.g. HTTP cổng 80, FTP cổng 20, 21 ...

Địa chỉ IP

- Định danh dùng trên tầng mạng
- Dùng trong giao thức IP - **Internet Protocol** (tầng mạng)
- Giá trị phụ thuộc từng mạng, mỗi card mạng được gán một địa chỉ IP
- Sử dụng để định danh một máy tính trong một mạng IP, ví dụ:
 - ❖ 133.113.215.10 (ipv4)
 - ❖ 2001:200:0:8803::53 (ipv6)

Địa chỉ dùng trên tầng liên kết dữ liệu

- 48 bit
- Địa chỉ vật lý / địa chỉ MAC
 - Sử dụng trong **tầng liên kết dữ liệu**
 - Cố định trên card mạng NIC (Network Interface Card)
 - Sử dụng để định danh máy tính trong mạng cục bộ

HEX

00:11:24:79:8e:82

BIN

00000000 00010001 00100100 01111001 10001110 10000010

OUI

Gán bởi nhà sản xuất

OUI (Organizationally Unique Identifier): Mã nhà sản xuất
Mỗi nhà sản xuất có các giá trị OUI riêng
Mỗi nhà sản xuất có thể có nhiều OUI

4. Kiến trúc phân tầng

- Mô hình thiết kế phân tầng
- Truyền thông trong kiến trúc phân tầng
- Mô hình OSI và TCP/IP
- Định danh trong TCP/IP
- Tổng kết về phân tầng & chòng giao thức

Khả năng cộng tác

- Rất nhiều công nghệ được triển khai theo nhiều cách rất khác nhau trên các nút mạng:
 - Phần cứng của những NSX khác nhau: IBM, Dell, Fujitsu, Apple...
 - HĐH khác nhau: Linux, Windows, MacOS, Android, iOS...
 - Người dùng sử dụng các ứng dụng khác nhau: Firefox, Chrome, Cốc Cốc...
 - Thiết bị mạng của những NSX khác nhau: Cisco, TP-Link...
- Và luôn luôn thay đổi
- Phew!

Nhưng tất cả đều có thể nói chuyện với nhau vì
chúng sử dụng chung giao thức

Trùu tượng và tái sử dụng

- Mỗi tầng có nhiều lựa chọn giao thức để sử dụng:
 - Tầng vật lý: cáp quang, ADSL, 3G, LTE...
 - Tầng liên kết dữ liệu: Ethernet, Token Ring, SONET, FDDI...
 - Giao vận: TCP, UDP
- Nhưng ở góc nhìn của tầng ứng dụng: Mozilla (và tất cả NSX khác) không phải viết trình duyệt Firefox (và tất cả ứng dụng khác) với 1 phiên bản cho mạng LAN, 1 cho mạng cáp quang, 1 cho mạng WiFi... ☺
 - Các giao thức cung cấp API chuẩn để phát triển ứng dụng
 - Các tầng thấp “trong suốt” với tầng ứng dụng

Trong suốt

- Công nghệ trên mỗi tầng thực hiện các phương thức truyền thông khác nhau
- Thay thế công nghệ ở các tầng có thể thực hiện song song
 - Miễn là giữ nguyên điểm truy cập dịch vụ SAP
- Thay thế công nghệ ở một tầng không ảnh hưởng đến các tầng khác

Hạn chế

- Một số thông tin ở tầng dưới bị “ẩn” (do tính trong suốt) đối với tầng trên có thể làm giảm hiệu năng hoạt động của tầng trên (và do đó làm giảm hiệu năng hoạt động của mạng)
 - Ví dụ: TCP phải kiểm soát tắc nghẽn trên đường truyền
- Phần tiêu đề có kích thước đáng kể trong gói tin
- Một số công nghệ tầng dưới có thể làm giao thức tầng trên thực hiện khó khăn hơn:
 - Ví dụ: TCP trên mạng không dây
- **TCP/IP không có các cơ chế an toàn bảo mật thông tin**

Tài liệu tham khảo

- “Computer Networking: A Top Down Approach”, J.Kurose
- “Computer Networks”, A.Tanenbaum
- “Data and Computer Communications”, W.Stallings



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chương Tầng vật lý

Đọc trước: Chapter 2 – The Physical Layer ,
Computer Networks, Tanenbaum

Tổng quan

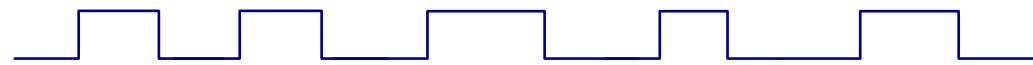
- Đảm nhận việc truyền dòng bit
 - đặt dòng bit từ máy trạm lên đường truyền
 - lấy dòng bit từ đường truyền vào máy trạm
- Một số vấn đề
 - Phương tiện truyền
 - Mã hóa - Điều chế
 - Dồn kênh...

Tùy tín hiệu tới gói tin

Analog Signal



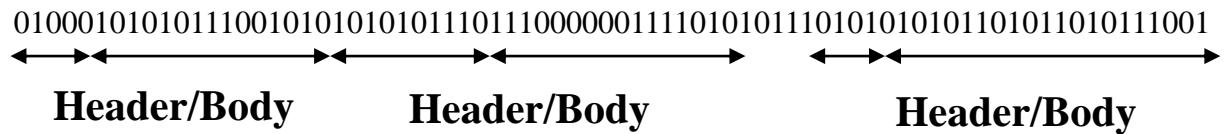
“Digital” Signal



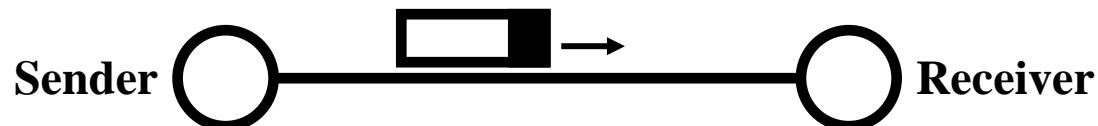
Bit Stream

0 0 1 0 1 1 1 0 0 0 1

Packets

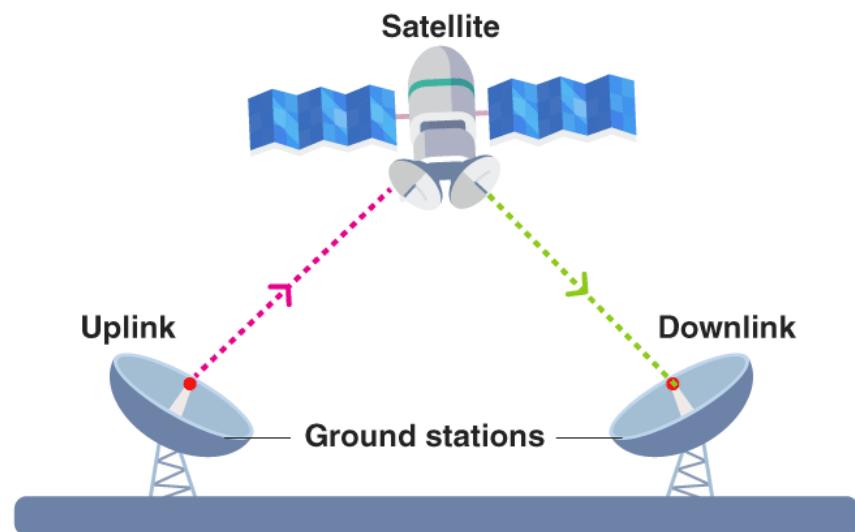


Packet
Transmission

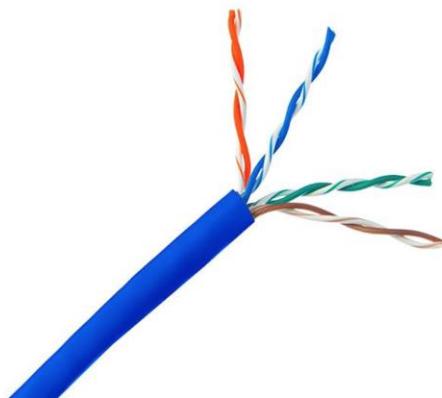


Đường truyền

- Hữu tuyến
 - Twisted Pair
 - Coaxial Cable
 - Fiber Optics
- Vô tuyến, không dây
 - Radio
 - Hồng ngoại
 - Ánh sáng
 - ...



Cáp xoắn đôi Twisted Pair



(a)



(b)

- (a) Category 3 UTP (10 Mbps, máy trạm, máy in, v.v).
- (b) Category 5 UTP (100-1000 Mbps, hạ tầng mạng).

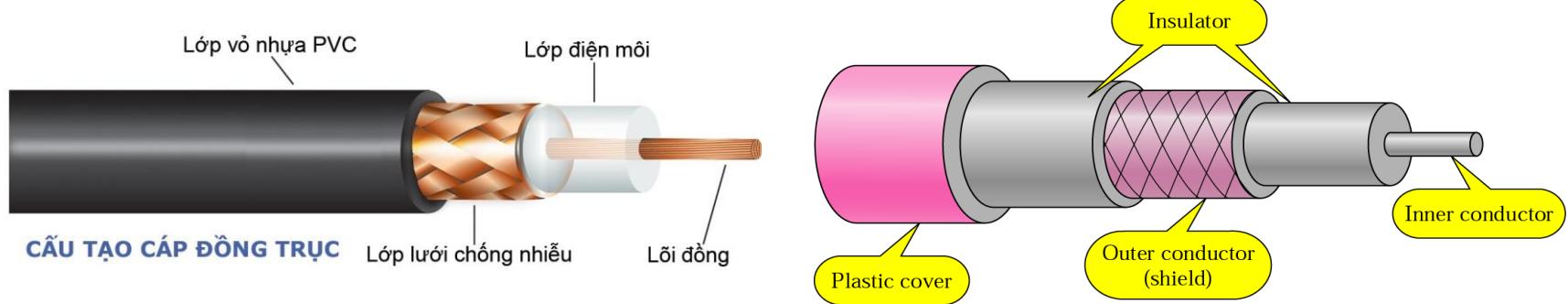
Cáp xoắn đôi

- **Cấu tạo:** Gồm nhiều cặp dây đồng xoắn với nhau.
- Có 2 loại:
 - Có bọc kim chống nhiễu (STP-Shielded Twisted Pair): ít phồng biến
 - Không bọc kim chống nhiễu (UTP-Unshielded Twisted Pair): phồng biến

Cáp xoắn đôi- Đánh giá

- Đơn giản
- Rẻ tiền
- Được dùng rộng rãi
- Khả năng chống nhiễu kém
(STP chống nhiễu tốt hơn UTP)
- Khoảng cách hạn chế 100m khi triển khai mạng Ethernet
- Giải thông hạn chế (x1MHz)
- Tốc độ hạn chế (100MHz)

Cáp đồng trục Coaxial Cable



Category	Impedance	Use
RG-59	75 W	Cable TV
RG-58	50 W	Thin Ethernet
RG-11	50 W	Thick Ethernet

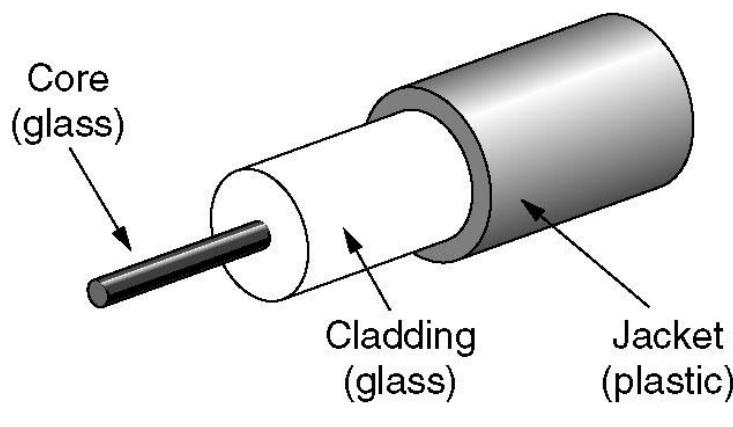
Cáp đồng trục

- Cấu tạo:
 - Lõi dẫn điện được bọc bởi một lớp điện môi không dẫn điện
 - Quấn thêm một lớp bện kim loại
 - Ngoài cùng có vỏ bọc cách điện

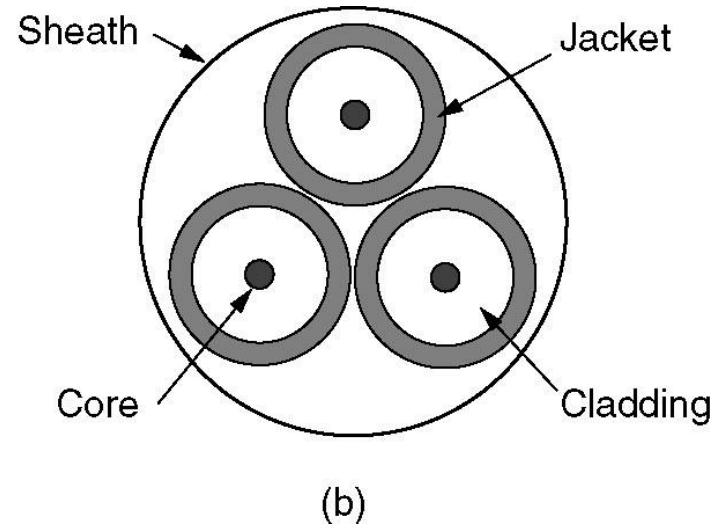
Ứng dụng

- Truyền bá TV
- Truyền các cuộc gọi điện thoại đường dài
 - 10,000 cuộc gọi cùng lúc
 - Đang bị thay thế bởi cáp quang
- Liên kết các máy tính khoảng cách ngắn
- Mạng cục bộ 10BaseT, 100BaseT, ...
- Khoảng cách triển khai thực tế ~500m.

Cáp sợi quang Optical fiber



(a)

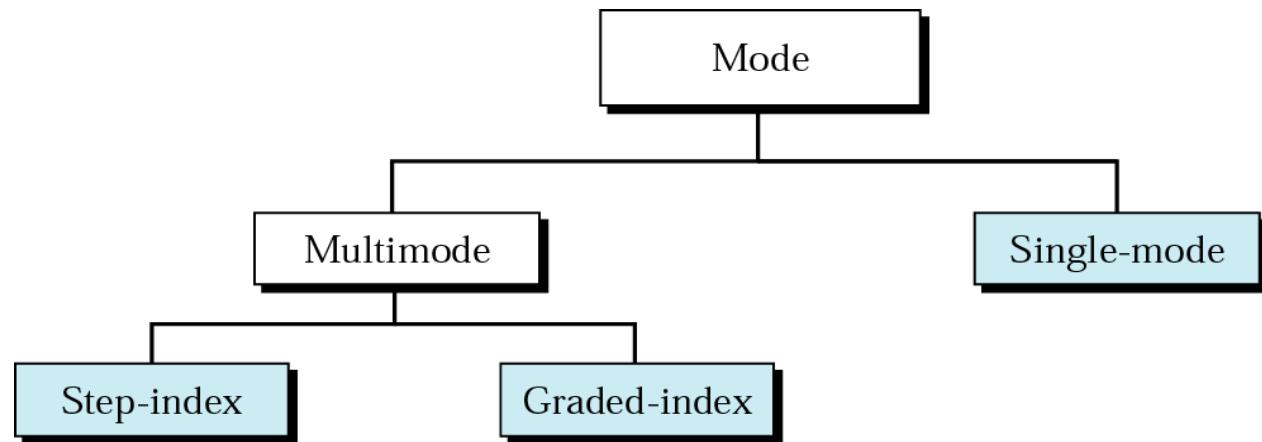
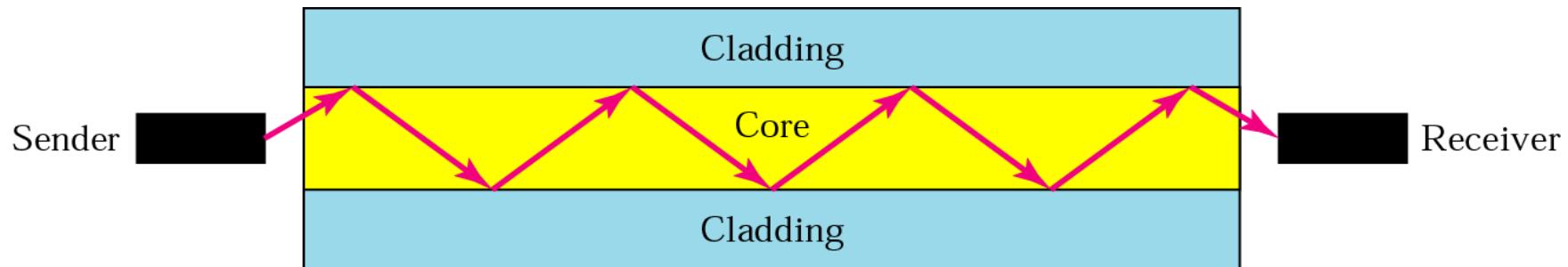


(b)

(a) Một sợi cáp

(b) Một đường cáp với 3 lõi

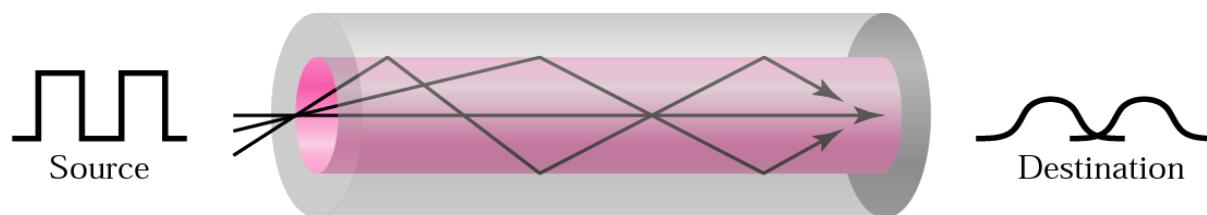
Cáp quang



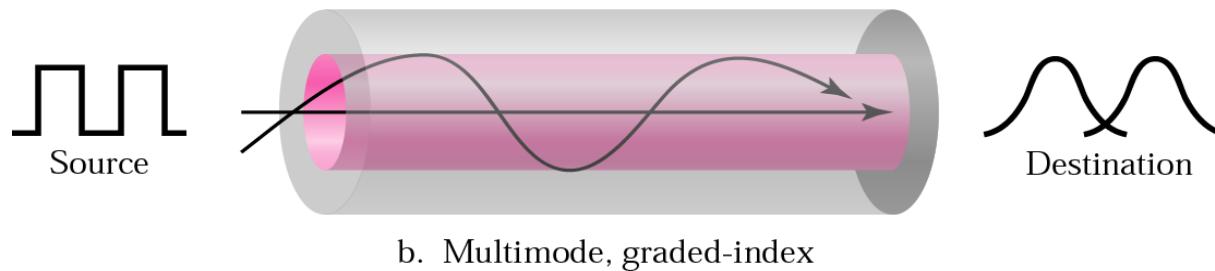
Cáp sợi quang

- Cấu tạo
 - Core: lớp lõi là sợi thủy tinh hoặc sợi plastic để truyền tín hiệu ánh sáng
 - Cladding: Vật chất quang bên ngoài bao bọc lõi mà phản xạ ánh sáng trở lại vào lõi.
 - Jacket: vỏ bọc bên ngoài
- Các sợi quang có thể được bó với nhau trong một đường cáp quang, có thêm sợi dây gia cường làm cứng cáp

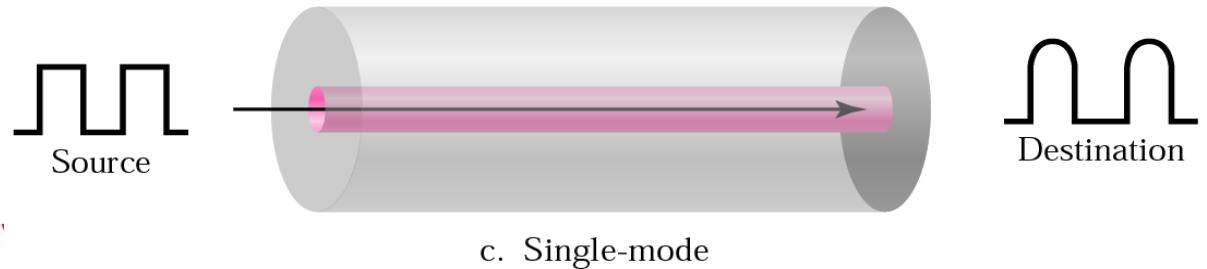
Cáp sợi quang



a. Multimode, step-index



b. Multimode, graded-index



c. Single-mode

Cáp sợi quang

- Multimode stepped index (chiết suất liên tục):
 - nhiều tia sáng đi theo nhiều đường
 - tại điểm đến sẽ nhận các chùm tia riêng lẻ,
 - xung dễ bị méo dạng.
- Multimode graded index (chiết suất bước):
 - Chỉ số khúc xạ của lõi giảm dần từ trong ra ngoài cladding.
 - Các tia gần trực truyền chậm hơn các tia gần cladding. Các tia truyền theo đường cong.
 - xung ít bị méo dạng.

Cáp sợi quang

- Single mode:
 - hệ số khúc xạ thay đổi từ lõi ra cladding ít hơn multimode.
 - Các tia truyền theo phương song song trực.
 - Xung nhậm được hội tụ tốt, ít méo dạng.

Ứng dụng

- Đường truyền khoảng cách xa
- Đường truyền trong thành phố
- Đường truyền giữa các router của Cty viễn thông
- Xương sống của LAN
- Thông lượng cao hơn
- Nhỏ, nhẹ hơn
- Suy hao ít hơn
- Cách ly điện từ tốt
- Khoảng cách phải lặp tín hiệu lớn hơn (10km)

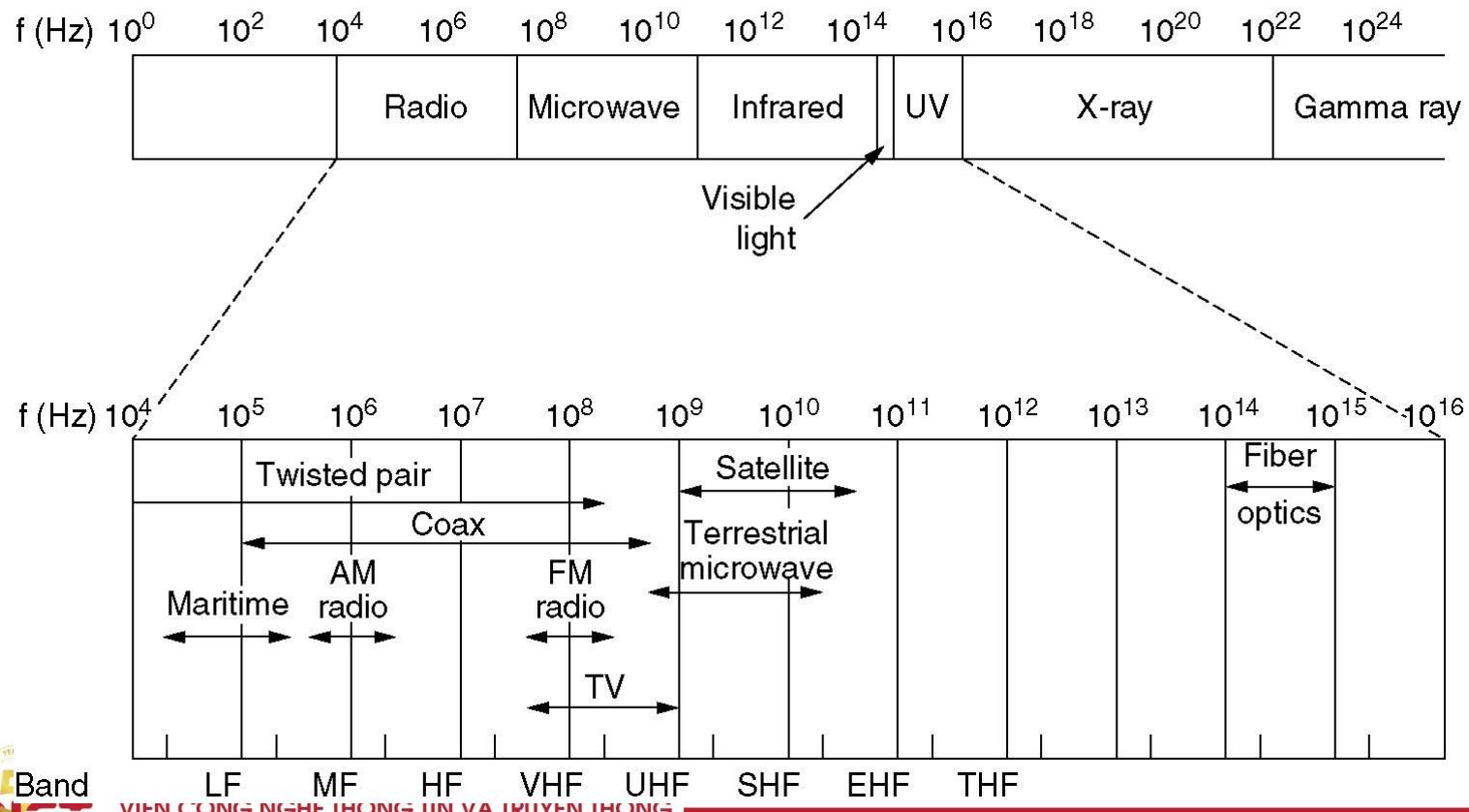
Phương tiện truyền không dây

- Truyền thông tin trên các dải tần khác nhau của sóng điện từ
- Không sử dụng dây nối
- Broadcast, bán song công: tại một thời điểm chỉ gửi hoặc nhận
- Ảnh hưởng của môi trường gây ra các hiện tượng:
 - ✓ Phản xạ
 - ✓ Nhiễu/giao thoa
 - ✓ Tán xạ do vật cản

Một vài khái niệm

- Sóng radio:
 - Bước sóng: 1mm – 100.000km
 - Tần số: 3 Hz – 300 GHz
 - VD: Bluetooth, WIFI
- Sóng vi ba (microwave):
 - Bước sóng: 1mm-1 m
 - tần số: 300 MHz-300 GHz
 - Vi ba mặt đất: Kết nối nội thị, hệ thống điện thoại di động
 - Vi ba vệ tinh: TV, điện thoại đường dài
- Hồng ngoại:
 - Bước sóng 700 nm- 1 mm
 - Tần số: 300 GHz-430 THz
 - Phạm vi nhỏ, không xuyên tường
 - VD: sóng dùng cho các bộ điều khiển từ xa
- Free Space Optics
 - Bước sóng dùng phổ biến: 850nm, 1300nm, 1550 nm.

Dải tần của các kênh truyền thông

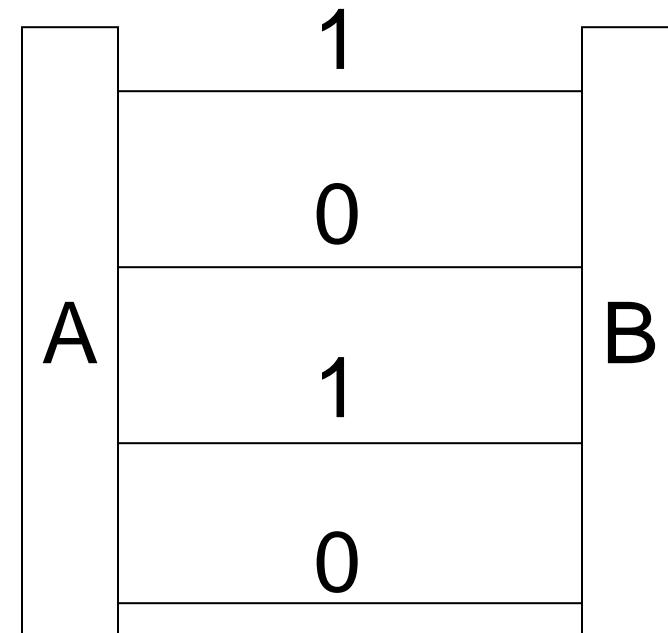
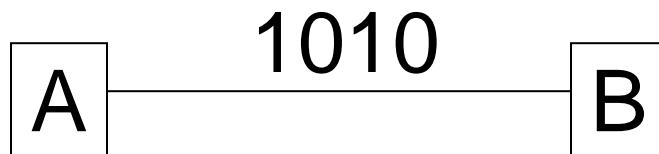


Phương thức truyền

- Đơn công – Simplex: Dữ liệu chỉ được truyền theo 1 chiều
- Song công – (Full) Duplex: Dữ liệu có thể được truyền theo cả 2 chiều tại cùng 1 thời điểm
- Bán song công – Half duplex: Dữ liệu có thể truyền theo cả 2 chiều nhưng tại 1 thời điểm thì chỉ có thể truyền theo 1 chiều

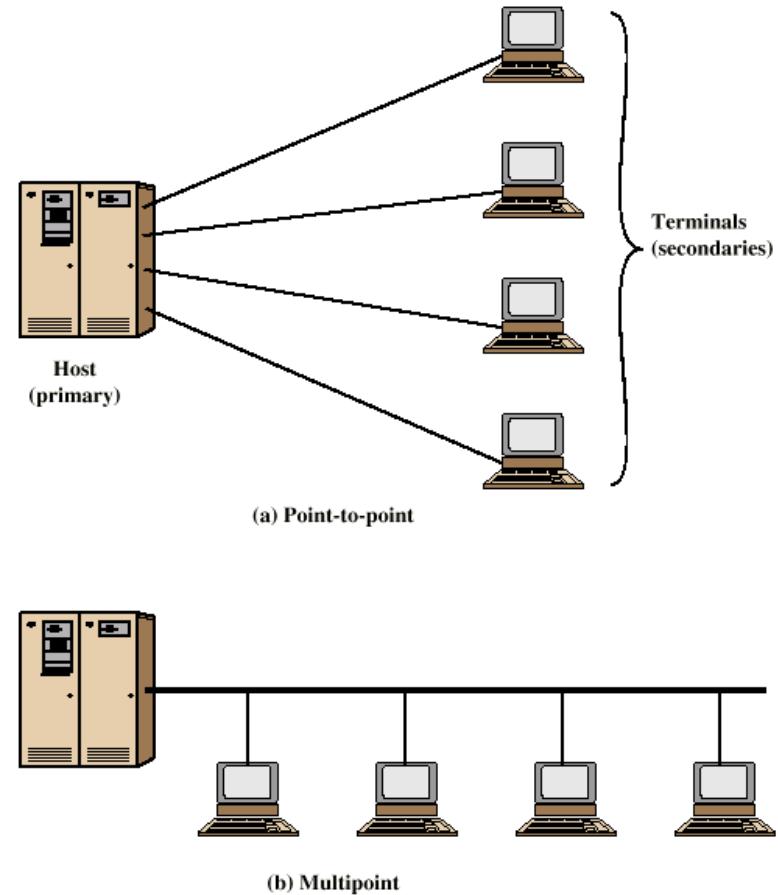
Hình thức truyền

- Truyền nối tiếp: Truyền 1 bit tại 1 thời điểm (trên 1 dây dẫn)
- Truyền song song: Truyền đồng thời nhiều bit tại cùng 1 thời điểm (trên nhiều dây dẫn)

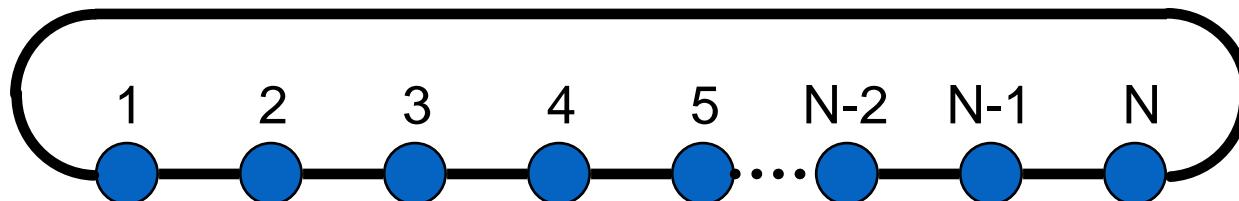
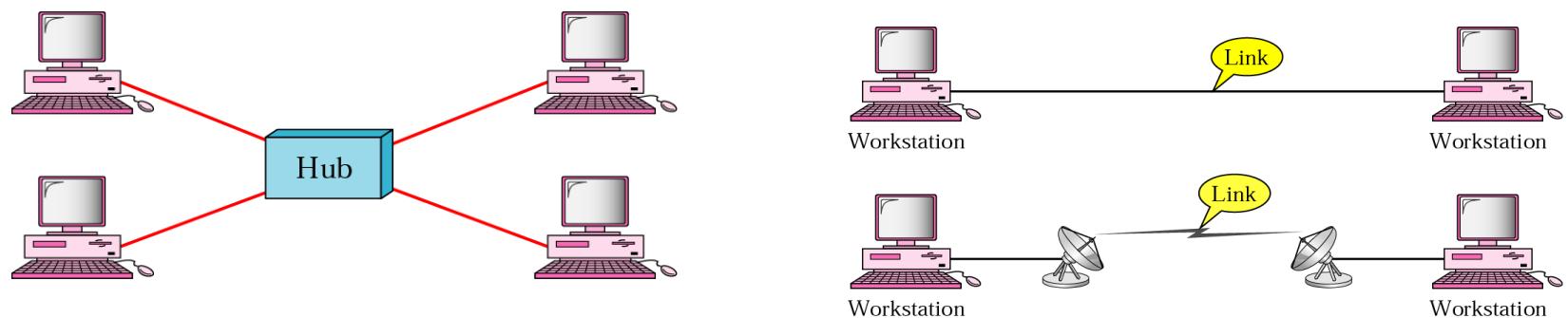
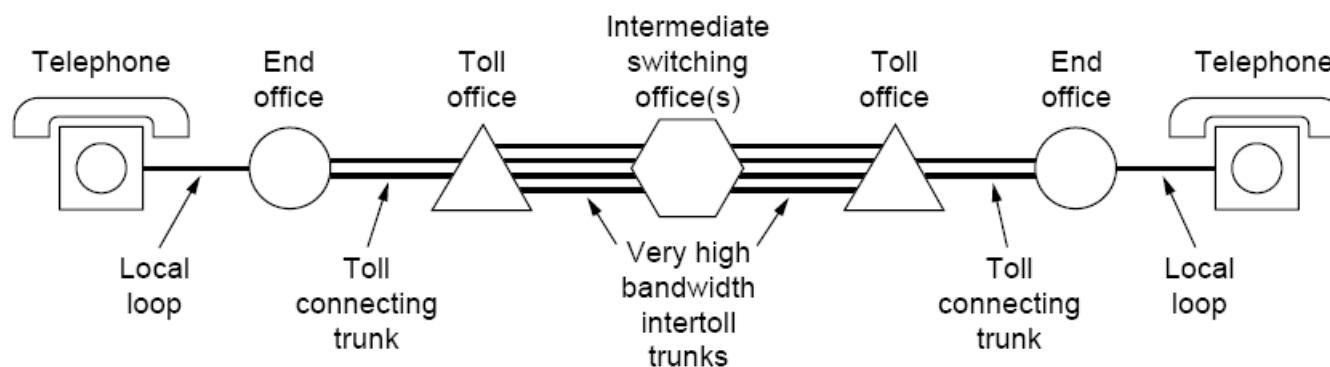


Topology

- Điểm điểm
 - Hình sao
 - Vòng
 - Đồ thị
- Điểm nhiều điểm
 - Trục
 - Vòng
 - Vệ tinh



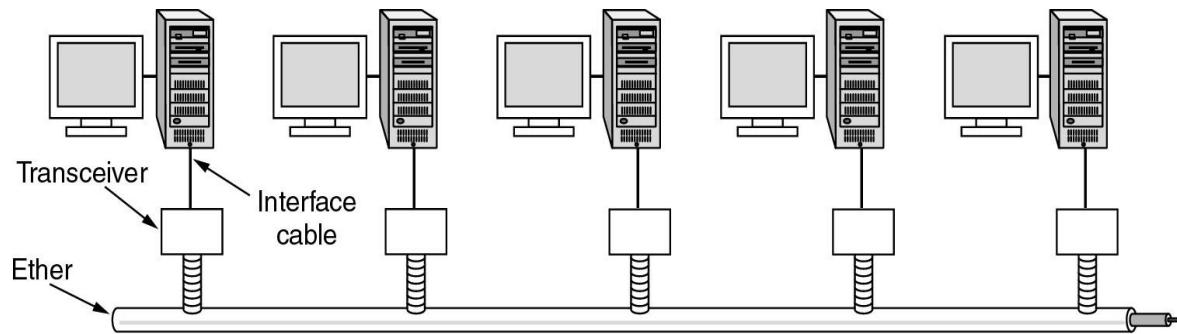
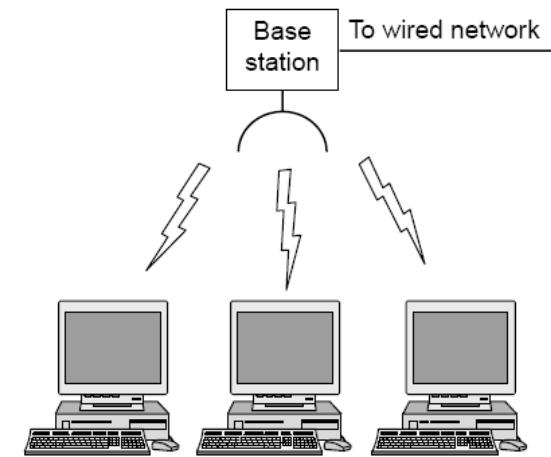
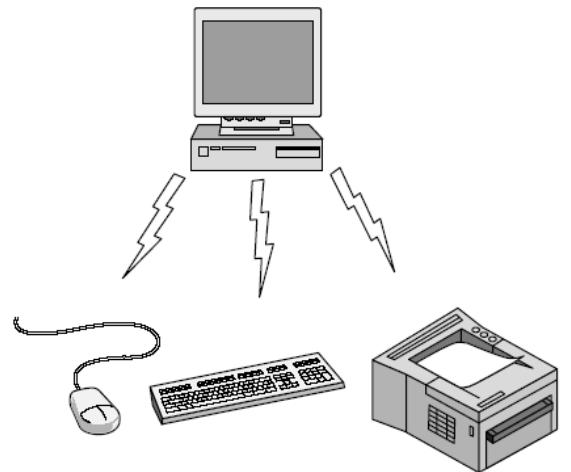
Điểm-điểm



Điểm-điểm

- Một đường truyền chỉ kết nối 2 thiết bị
- Kết nối giữa 2 thiết bị:
 - 1 đường truyền (bán song công) hoặc
 - 2 đường truyền (song công)
- *Trường hợp bán song công có thể có xung đột xảy ra khi 2 thiết bị trên một kết nối cùng truyền dữ liệu*

Điểm-nhiều điểm



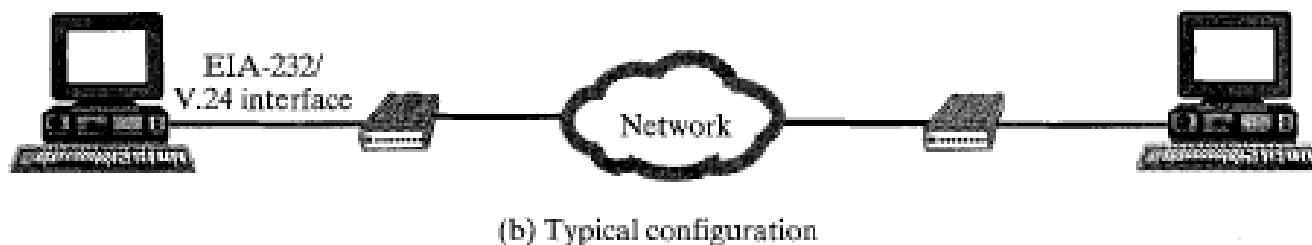
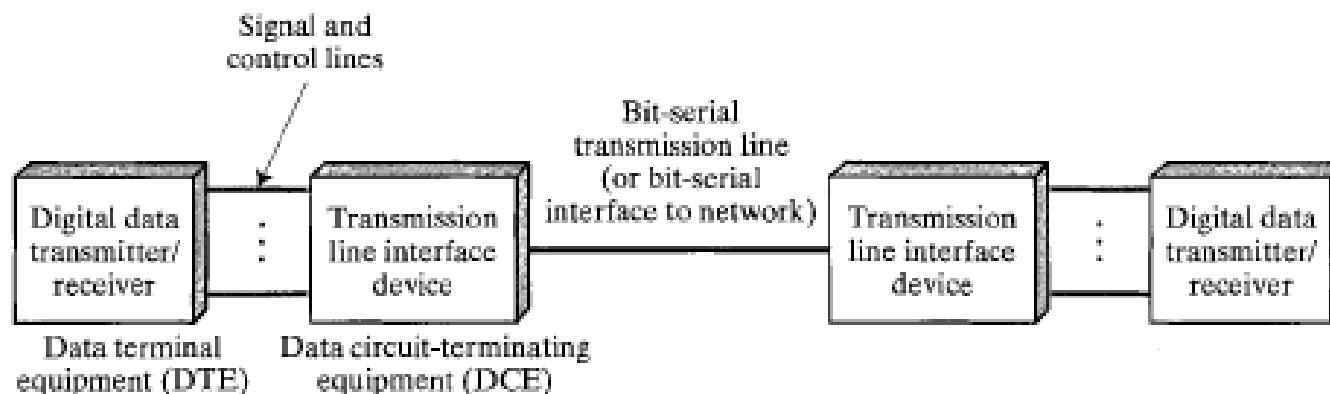
Điểm-nhiều điểm

- Đặc trưng chung của topo điểm – nhiều điểm là một đường truyền duy nhất kết nối nhiều thiết bị đầu cuối với nhau
- Dữ liệu được quảng bá (broadcast)
- Xung đột khi hai trạm cùng phát tín hiệu → Hai tín hiệu ngược chiều nhau gắp nhau trên đường truyền
- Cần có các phương pháp điều khiển *đa truy cập (multiple access)* → sẽ xem ở tầng Liên kết dữ liệu.

Giao diện đường truyền

- Thiết bị đầu cuối dữ liệu (data terminal equipment, DTE)
 - Không có các tính năng truyền thông
 - Cần có các thiết bị bổ sung để truy cập đường truyền
- Thiết bị cuối kênh dữ liệu (data circuit terminating equipment, DCE)
 - Truyền các bít trên đường truyền
 - Trao đổi dữ liệu và các thông tin điều khiển với DCE qua các dây nối
- Cần các giao diện chuẩn, rõ ràng giữa DTE, DCE

DTE-DCE



Giao diện đường truyền

- Cơ
 - Hình dạng giắc cắm, số lượng chân, đảm bảo cắm được lắn nhau
- Điện
 - Mức điện áp sử dụng
 - Chiều dài xung (tần số xung nhịp)
 - Phương pháp mã hóa
- Chức năng
 - Dây dẫn nào dùng làm gì
 - Có 4 nhóm: dữ liệu, điều khiển, đồng bộ, nối đất
- Thủ tục
 - Các thủ tục, chuỗi các sự kiện để thực hiện việc truyền tin

Ví dụ: EIA-232-E (RS-232)

- Chuẩn định nghĩa cho cổng nối tiếp
- Cơ: ISO 2110
- Điện: V. 28
- Chức năng: V. 24
- Thủ tục: V. 24

Ví dụ: V.24 /EIA-232-E

- Cơ:
 - 25 chân hoặc 15 chân
 - Khoảng cách 15m
- Điện
 - Tín hiệu số
 - 1=-3v, 0=+3v (NRZ-L)
 - Tốc độ truyền tin 20kbps
 - Khoảng cách < 15m

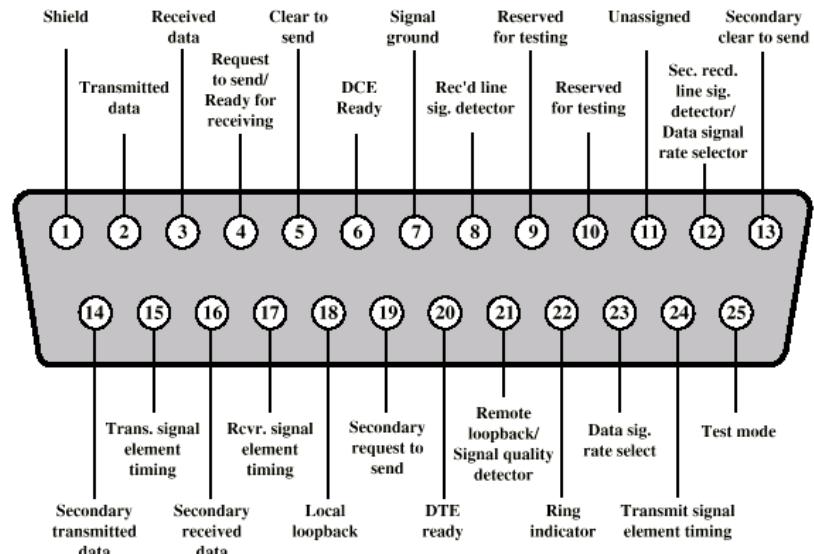


Figure 6.5 Pin Assignments for V.24/EIA-232 (DTE Connector Face)

Mã hóa thông tin

- Biến đổi bít nhị phân thành dạng tín hiệu vật lý thích hợp để truyền trên đường truyền vật lý.
- Mã hóa: Sử dụng các tín hiệu rời rạc, điện áp khác nhau để biểu diễn các bít 0 và 1.
- Điều chế: Sử dụng tín hiệu tương tự (sóng) để biểu diễn các bit 0,1.
- Việc truyền phải được đồng bộ giữa hai bên
- Có thể mã hóa theo từng bit hoặc một khối các bit, e.g., 4 hay 8 bits.
- Có nhiều cách biểu diễn khác nhau → các phương pháp mã hóa

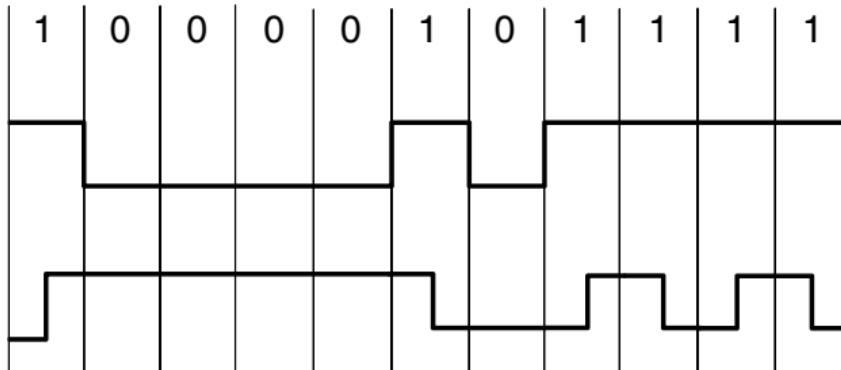
Các phương pháp mã hóa dữ liệu số - tín hiệu số

- NRZ
 - NRZ-L,NRZI
- Nhị phân đa mức
 - Đa cực AMI
 - pseudoternary
- Hai pha
 - Manchester

NRZ-L Non Return to Zero Level

- Trong thời gian của một bít, tín hiệu không trở về mức 0
- Không có chuyển mức trong khoảng thời gian của một bít
- NRZ-L Non return to zero level
 - Bít 1 tương ứng mức tín hiệu cao/thấp
 - Bít 0 tương ứng với mức tín hiệu thấp/cao

(a) Bit stream



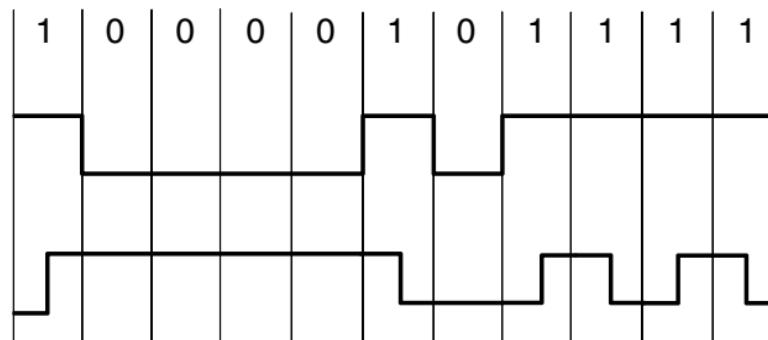
(b) Non-Return to Zero (NRZ)

(c) NRZ Invert (NRZI)

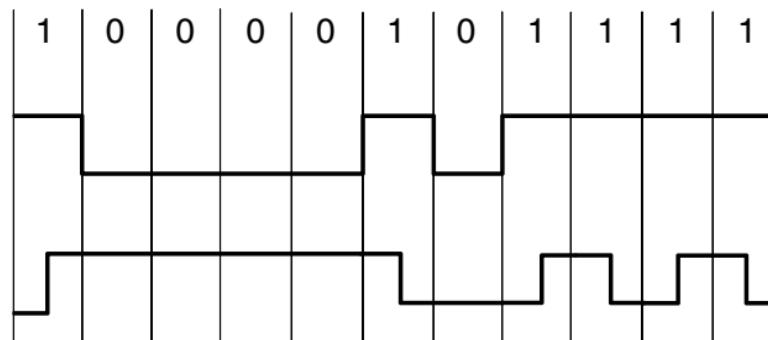
NRZ-I Non return to zero invert

- Bít 0 tương ứng với không chuyển mức ở đầu thời gian bít
- Bít 1 tương ứng với chuyển mức ở đầu thời gian bít
- Là một phương pháp điều chế vi sai:
 - 0 và 1 tương ứng với chuyển mức, không phải với mức giá trị
 - Tin cậy/Đơn giản hơn điều chế theo mức
 - Không phụ thuộc vào cực của tín hiệu

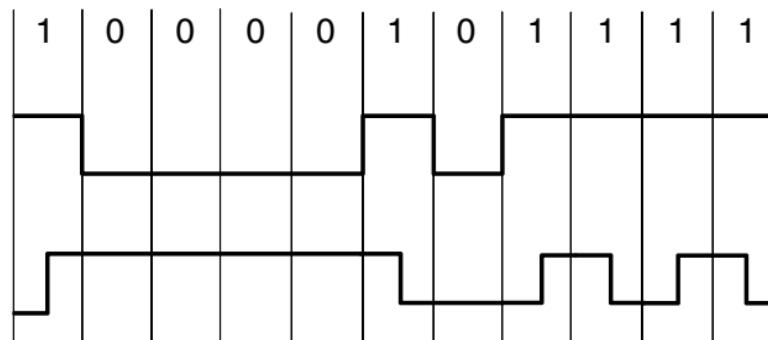
(a) Bit stream



(b) Non-Return to Zero (NRZ)

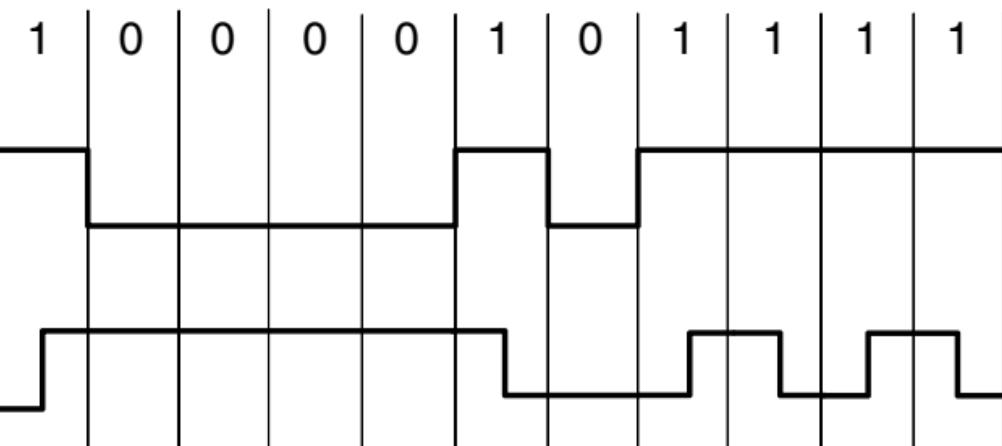


(c) NRZ Invert (NRZI)



NRZ-L và NRZ-I

(a) Bit stream



(c) NRZ Invert (NRZI)

Một số yếu tố cần xem xét để đánh giá một mã

- Đồng bộ đồng hồ bên gửi và bên nhận:
 - Nếu đồng hồ bên gửi và bên nhận không được đồng bộ, bên nhận có thể xác định sau thời gian một bit
 - → giải mã sai → tăng tỷ lệ lỗi dữ liệu
 - Một số mã có yếu tố giúp đồng bộ trong mã
- Thành phần một chiều trong tín hiệu:
 - Thành phần một chiều xuất hiện khi tín hiệu ở mức dương quá lâu hoặc âm quá lâu
 - làm cho bên nhận xác định sai mức tín hiệu cơ sở
 - → giải mã sai dữ liệu
 - Phương pháp mã hóa cần tránh thành phần một chiều bằng cách duy trì giá trị trung bình của tín hiệu ở mức 0.

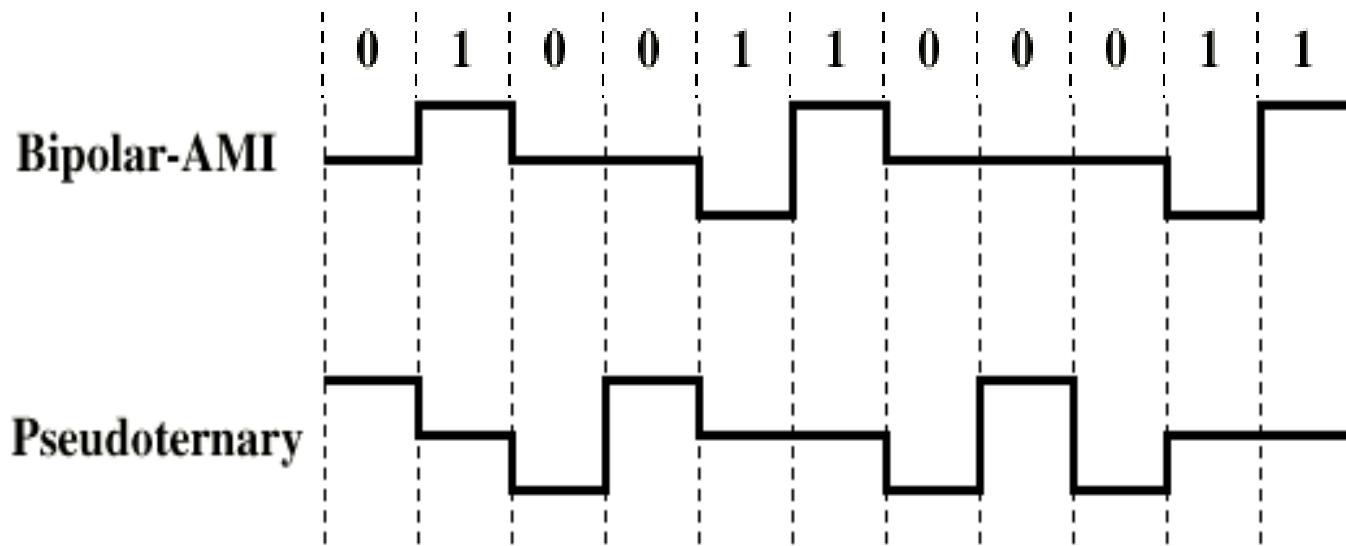
Đánh giá NRZ

- **Ưu điểm**
 - Đơn giản, sử dụng tối đa đường truyền
 - Frequency range from 0 - $\frac{1}{2}$ data speed
 - Example: 9600bps -> 4800khz
- **Nhược điểm**
 - Khó đồng bộ bằng tín hiệu
 - Vd với NRZ-L khi có nhiều 0 hoặc 1 liên tiếp, tín hiệu giữ một mức trong khoảng thời gian dài, dễ mất đồng bộ. Với NRZ-I, một chuỗi 0 cũng gây ra tình trạng như vậy
 - Có thành phần một chiều khi truyền toàn 1.
- **Ứng dụng**
 - Lưu trữ dữ liệu trên các vật liệu từ tính
 - Ít dùng trong truyền số liệu

Mã nhị phân đa mức

- Bipolar alternate mark inversion
- Sử dụng nhiều hơn 2 mức tín hiệu cho một bít
- Lưỡng cực đảo mức 1
 - 0 Tương ứng với không có tín hiệu
 - 1 tương ứng với có tín hiệu. Tín hiệu đảo cực giữa hai bít 1 liên tiếp
- Giả tam phân (pseudoternary)
 - 1 Tương ứng với không có tín hiệu
 - 0 tương ứng với có tín hiệu. Tín hiệu đảo cực giữa hai bít 0 liên tiếp

Mã nhị phân đa mức



Mã nhị phân đa mức

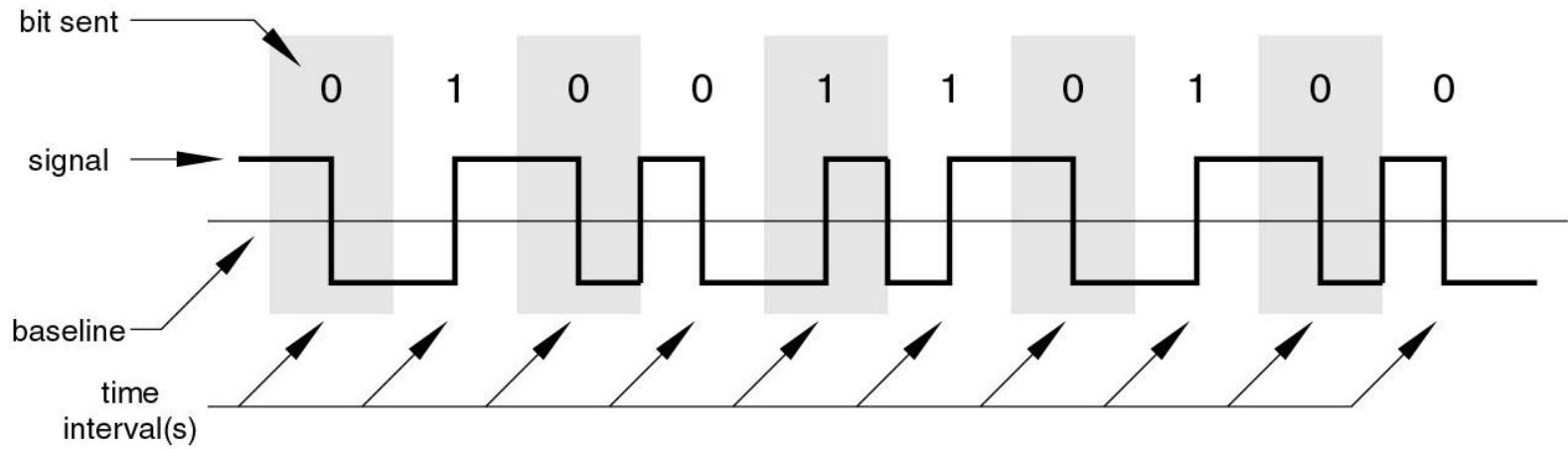
- Thành phần một chiều=0
- Có khả năng phát hiện lỗi
- Đồng bộ khi có nhiều bít 1(0), không đồng bộ khi có nhiều bít 0(1)
- Giải thông thấp hơn
- 3 mức tín hiệu cho một bít:
 - Không sử dụng tối ưu đường truyền
 - Tăng tỉ lệ lỗi (đích cần phân biệt 3 mức tín hiệu)

Mã hai pha: Manchester

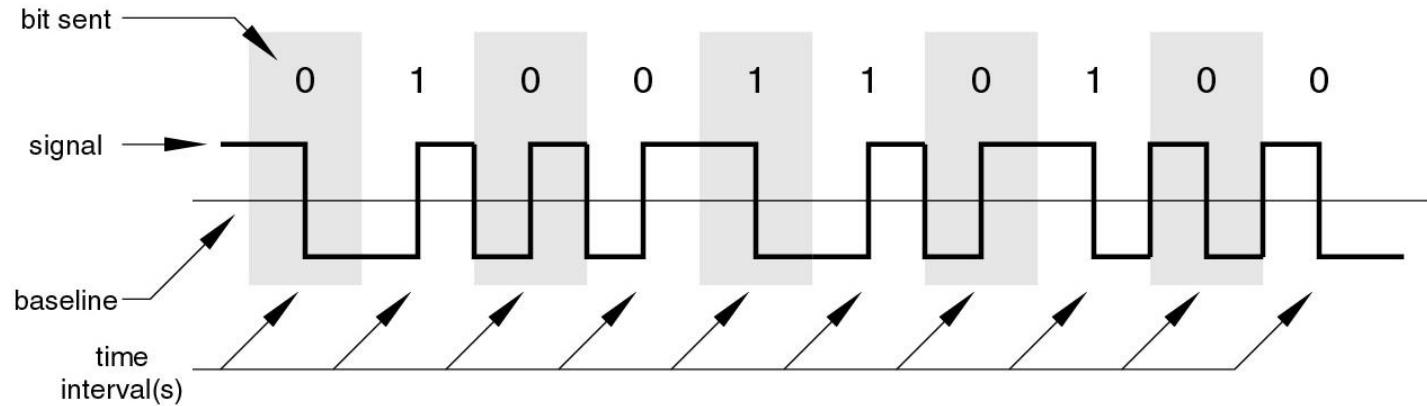
- Luôn luôn có chuyển mức ở giữa thời gian của một bít
 - Thấp lên cao: 0, cao xuống thấp 1
 - Chuyển mức cung cấp cơ chế đồng bộ
- Manchester
 - Luôn có chuyển mức ở giữa bít
 - Bít 0: sườn âm (chuyển từ +V sang –V)
 - Bít 1: sườn dương (chuyển từ –V sang +V)
 - Dùng trong mạng Ethernet.
- Manchester visai
 - 0: có chuyển mức ở đầu bít, 1 không có chuyển mức
 - Chuyển mức ở giữa bít chỉ phục vụ cho đồng bộ
 - Luôn có chuyển mức tín hiệu ở giữa bit

Mã Manchester

Manchester Encoding



Differential Manchester Encoding

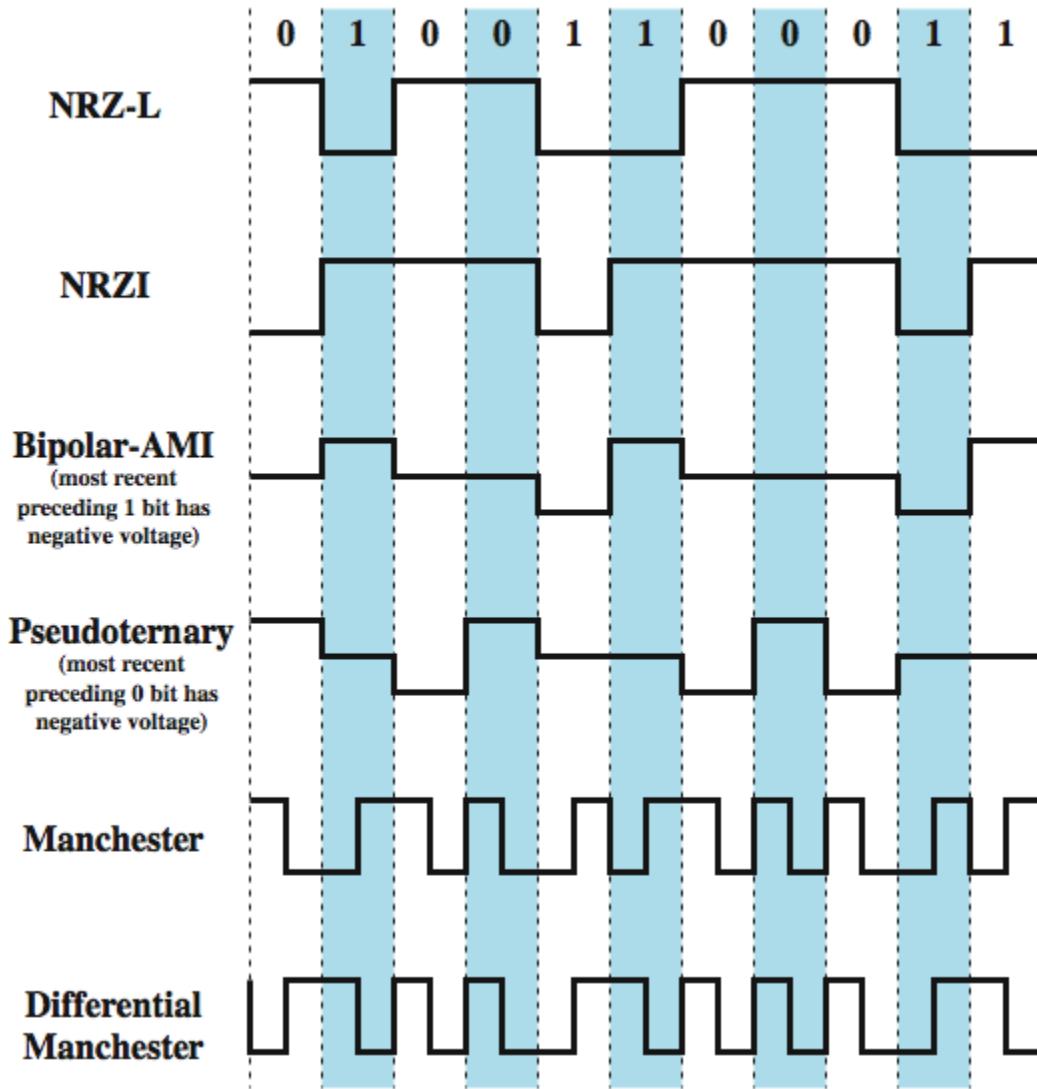


Tốc độ tín hiệu/Tốc độ điều chế

<i>Thuật ngữ</i>	<i>Đơn vị</i>	<i>Định nghĩa</i>
Đơn vị dữ liệu	bit	Một bít đơn, giá trị 0 hoặc 1
Tốc độ dữ liệu	bit/s	Tốc độ truyền các đơn vị dữ liệu
Đơn vị tín hiệu	xung chữ nhật xung hình sin	Phần tín hiệu tương ứng với thời gian nhỏ nhất của 1 ký hiệu
Tốc độ tín hiệu Tốc độ điều chế	Đơn vị tín hiệu/s (baud)	Tốc độ truyền các đơn vị tín hiệu

- Có bao nhiêu tín hiệu dùng cho 1 bít trong mã hóa NRZ, Manchester?

Tổng hợp các phương pháp mã hóa



Bài tập-01

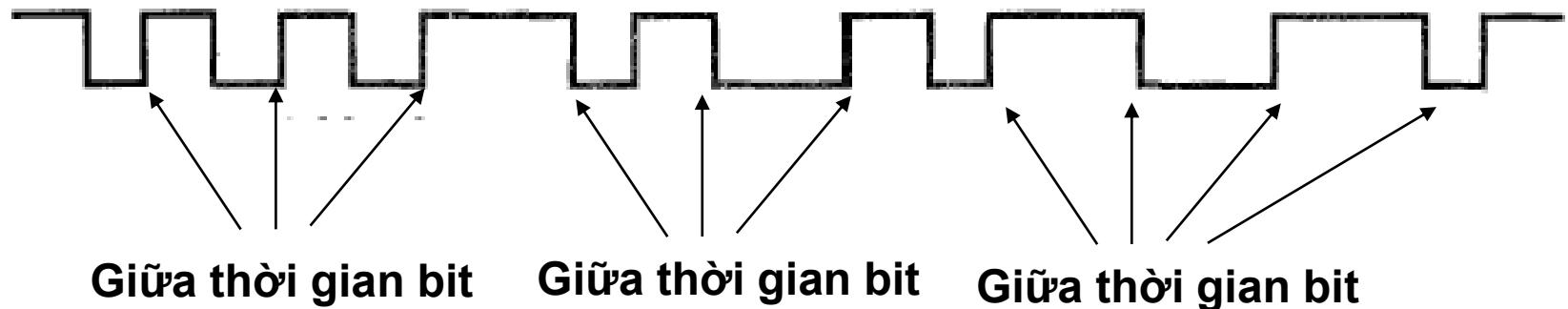
- Biểu diễn các tín hiệu mã hóa chuỗi dữ liệu sau đây bằng các phương pháp mã hóa đã học
 - 11000000 00000010 11001101 01010101

Bài tập-02

- Dữ liệu mã hóa bằng mã manchester (không vi sai) cho tín hiệu
 - Xác định thời gian của từng bít
 - Xác định dữ liệu ban đầu



Đáp án bài 2



- Chuỗi bit ban đầu: 1110011010

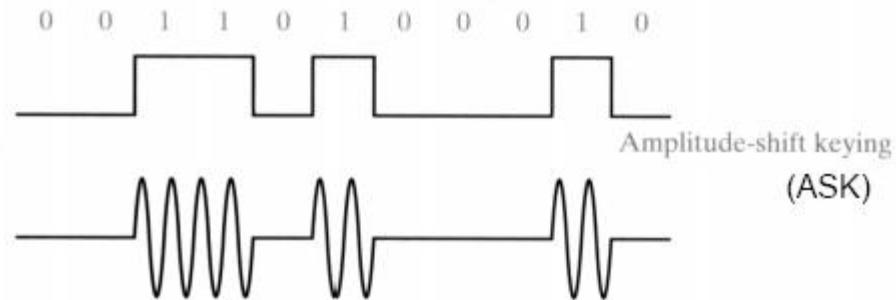
2. Điều chế dữ liệu số-tín hiệu liên tục

- Ví dụ: truyền số liệu thông qua hệ thống điện thoại
 - Hệ thống điện thoại truyền, chuyển tiếp tín hiệu điện có tần số 300Hz đến 3400Hz
 - Tại nguồn và đích, dữ liệu số cần được điều chế thành tín hiệu liên tục để truyền trên đường điện thoại
- Căn cứ vào tính chất của tín hiệu, chúng ta có 3 kỹ thuật điều chế
 - Điều chế khóa dịch biên độ
 - Điều chế khóa dịch pha
 - Điều chế khóa dịch tần số
 - $A \cos(2\pi f t + \phi)$

Điều chế khóa dịch biên độ (ASK)

- Biên độ của sóng mang biến đổi theo thông tin cần truyền.
- 0 và 1 tương ứng với hai biên độ tín hiệu, thông thường một trong hai biên độ=0
- Dễ bị ảnh hưởng bởi nhiễu (1200bps cho đường thoại)
- Khó đồng bộ
- Thường được dùng trong cáp quang (LED hoặc laser)

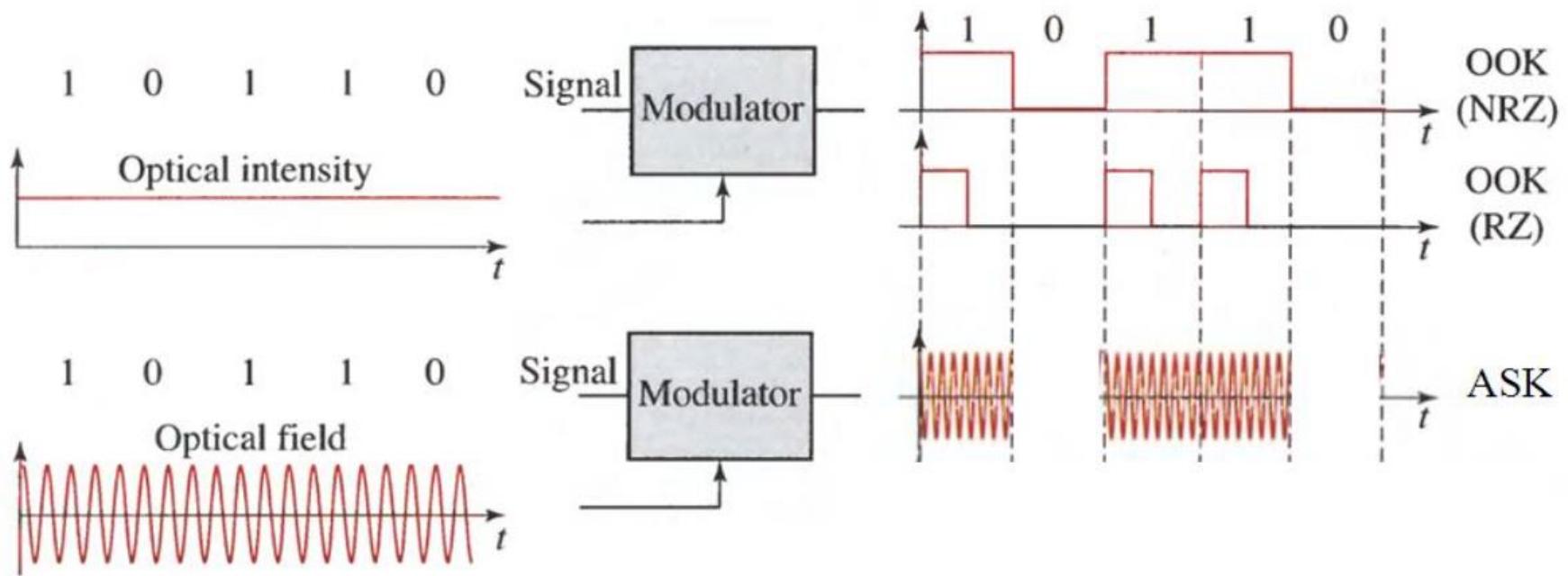
$$s(t) = \begin{cases} A\cos(2\pi ft) & \text{cho } 1 \\ 0 & \text{cho } 0 \end{cases}$$



Mã On-Off Keying (OOK)

- Dùng trong cáp quang
- Là một loại điều chế dịch biên độ.
 - 1: có xung ánh sáng trong thời gian bit (bật nguồn sáng).
 - 0: không có xung ánh sáng trong thời gian bit (tắt nguồn sáng).
- OOK có thể dùng nhiều định dạng tín hiệu khác nhau:
 - NRZ: xung ánh sáng chiếm toàn bộ độ dài bit 1.
 - RZ (return-to-zero): chỉ phát xung ánh sáng trong một phần thời gian của bit 1.

Mã On-Off Keying (OOK)

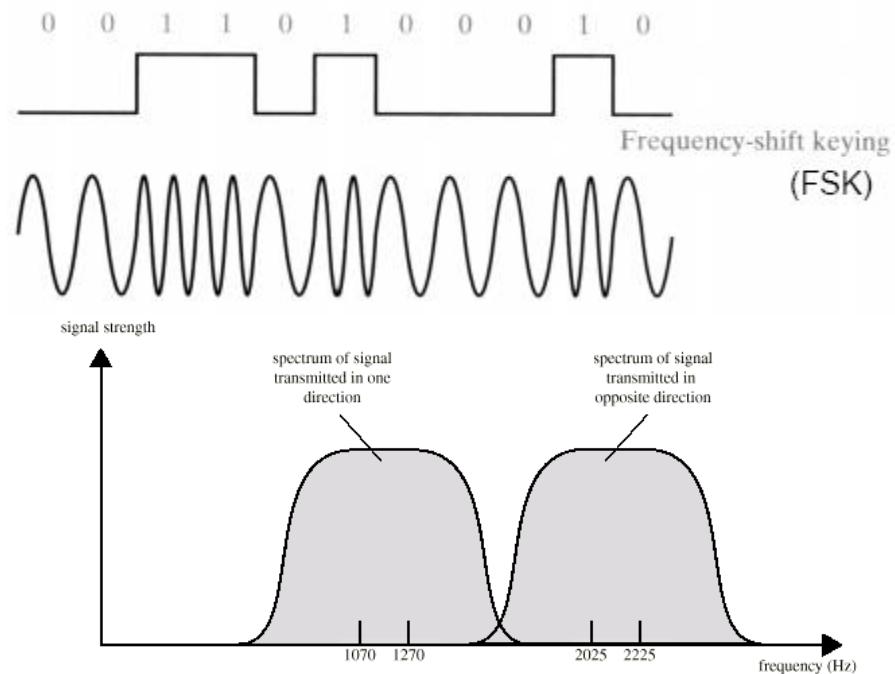


On off key nhìn từ phương diện cường độ sáng (hình trên)
và tín hiệu quang học (hình dưới)

Điều chế khóa dịch tần số (FSK)

- Hai giá trị nhị phân được biểu diễn bởi hai tín hiệu tần số khác nhau
- Ví dụ về điều tần song công
- Tỷ suất lỗi thấp hơn
- Dùng trong truyền số liệu qua đường điện thoại (tần số thấp), hoặc trong mạng không dây (tần số cao)

$$s(t) = \begin{cases} A \cos(2\pi f_1 t) \\ A \cos(2\pi f_2 t) \end{cases}$$

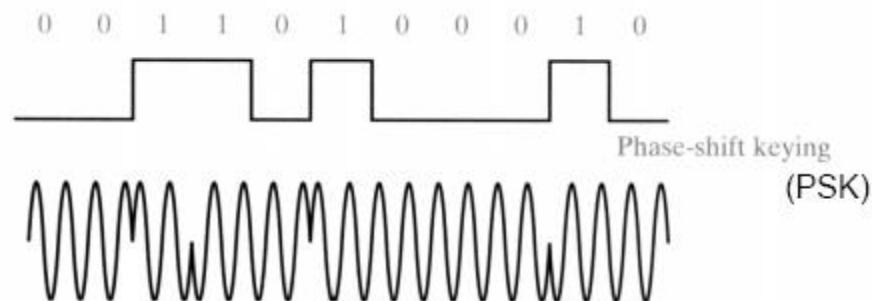


Điều chế khóa dịch pha (PSK)

- 0,1 tương ứng với hai độ lệch pha khác nhau
- 0,1 tương ứng với chuyển pha (vi sai)
- Có thể sử dụng giải thông một cách hiệu quả hơn khi mã hóa cùng lúc nhiều bít
- Có thể kết hợp với điều biên
- Nếu tốc độ dữ liệu là 9600 bps, tốc độ điều chế là ?

$$s(t) = \begin{cases} A \cos(2\pi f_c t + \pi) & \text{binary 1} \\ A \cos(2\pi f_c t) & \text{binary 0} \end{cases}$$

$$s(t) = \begin{cases} A \cos(2\pi f_c t + 45^\circ) & 11 \\ A \cos(2\pi f_c t + 135^\circ) & 10 \\ A \cos(2\pi f_c t + 225^\circ) & 00 \\ A \cos(2\pi f_c t + 315^\circ) & 01 \end{cases}$$

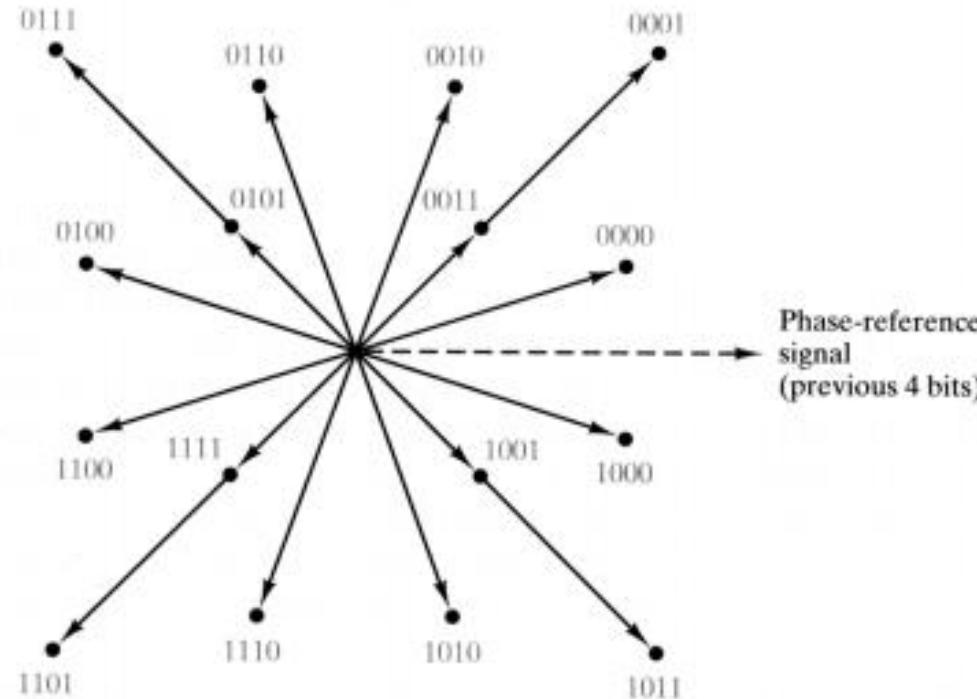


Kết hợp với điều biến

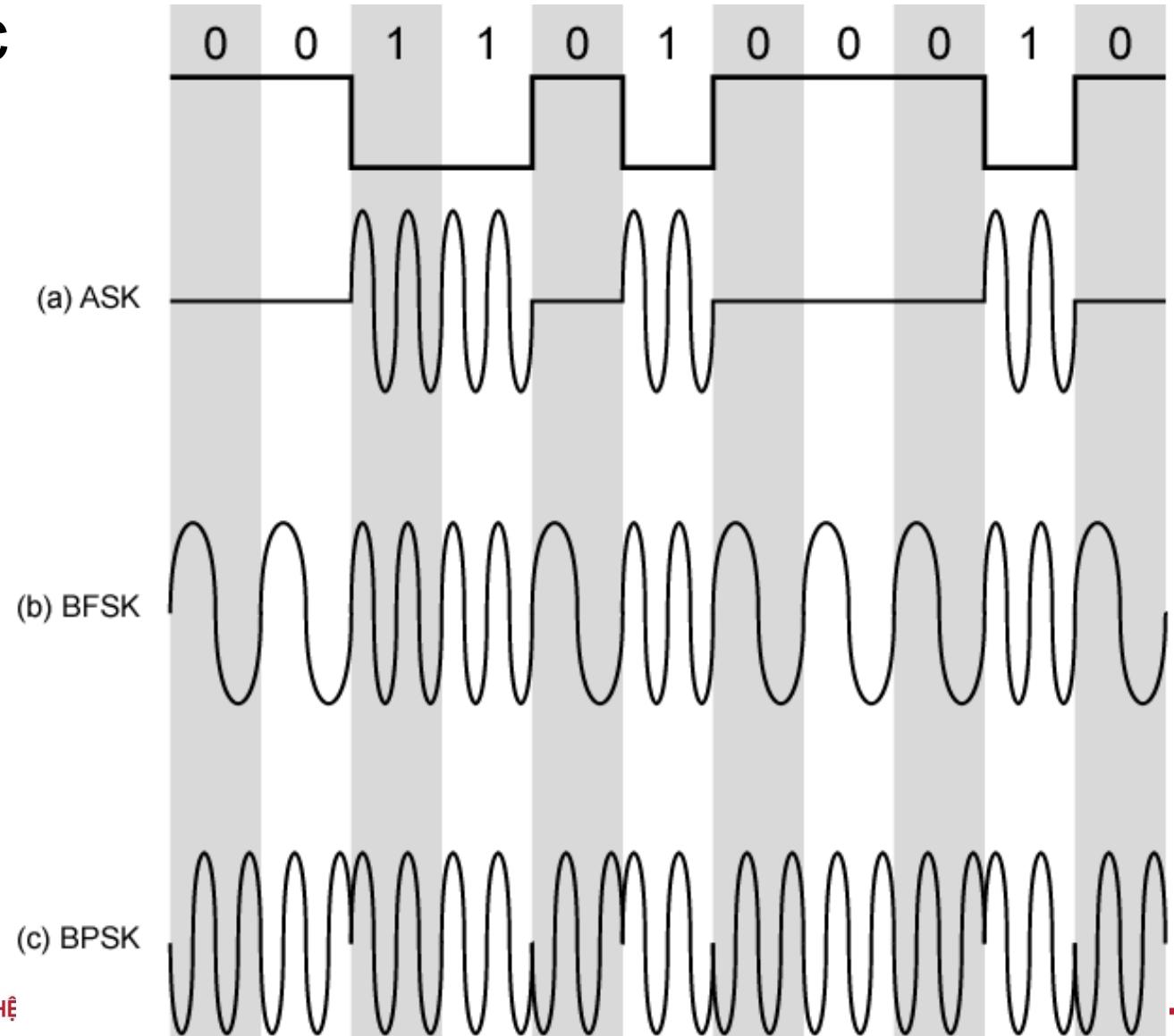
Các tổ hợp được biểu diễn trên bản đồ sao, mỗi tia ứng với 1 mã.

- Độ dài tia ứng với biên độ.
- Góc lệch với pha tham chiếu ứng với góc pha

– 9,600 bps modem (2,400 baud × 4)



Tổng hợp điều chế số/liên tục

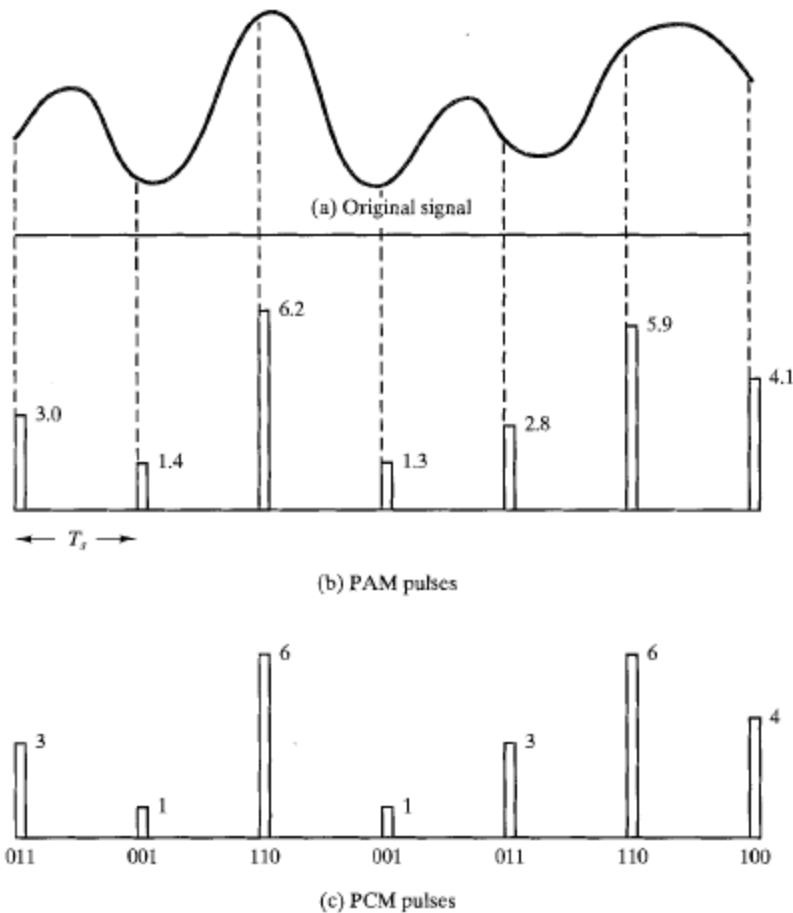


3. Điều chế dữ liệu liên tục- số

- Điều chế dữ liệu liên tục thành dữ liệu số, sau đó
 - Điều chế thành tín hiệu số
 - Mã hóa trực tiếp bằng NRZ-L
 - Sử dụng phương pháp mã hóa tín hiệu số khác
 - Điều chế thành tín hiệu liên tục
 - Sử dụng các biện pháp điều chế số-liên tục đã học
- Có hai phương pháp chính điều chế dữ liệu liên tục thành dữ liệu số
 - Điều chế mã xung
 - Điều chế Delta

Điều chế mã xung (PCM)

- Pulse Code Modulation
- Lấy mẫu tín hiệu dựa trên định luật lấy mẫu của Shannon
 - Nếu tần số lấy mẫu ≥ 2 lần tần số (có ý nghĩa) cao nhất của tín hiệu, phép lấy mẫu bảo toàn thông tin của tín hiệu
 - Vd: Tiếng nói tần số tối đa 4300Hz, cần lấy mẫu với tần số min 8600Hz
- Tiến hành theo hai bước
 - Lấy mẫu (PAM)
 - Lượng tử hóa

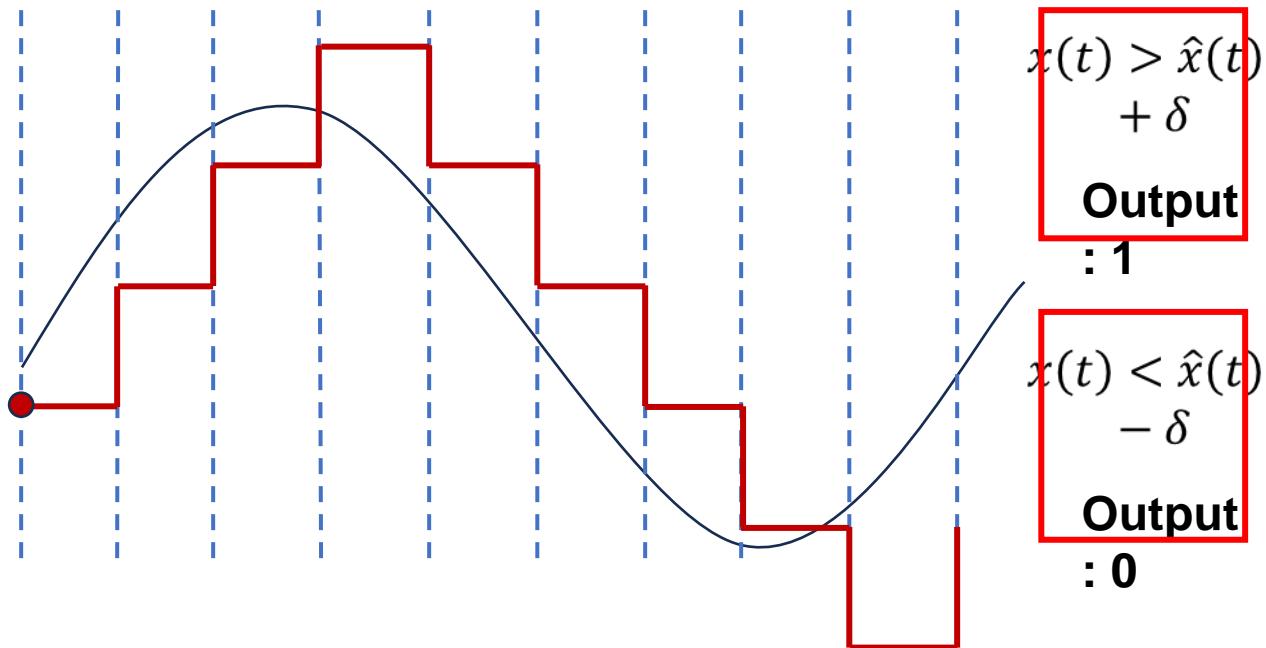


3.2. Điều chế delta (Delta Modulation)

- Vấn đề của PCM: tốn băng thông → các giải pháp khác: Điều chế Delta
- Ưu điểm của Điều chế Delta: Chỉ sử dụng 1 bit cho mỗi mẫu
- Ý tưởng: Không truyền đi giá trị thật của mẫu $x(t)$, mà truyền đi thông tin thay đổi (lớn hơn hay nhỏ hơn) của giá trị mẫu hiện tại với *giá trị xấp xỉ* $\hat{x}(t)$.
 - Nếu lớn hơn: bit 1
 - Nếu nhỏ hơn: bit 0

3.2. Điều chế delta (Delta Modulation)

- Nếu $x(t) > \hat{x}(t) \rightarrow \hat{x}(t) := \hat{x}(t) + \delta$
 - Output = 1
- Nếu $x(t) < \hat{x}(t) \rightarrow \hat{x}(t) := \hat{x}(t) - \delta$
 - Output = 0

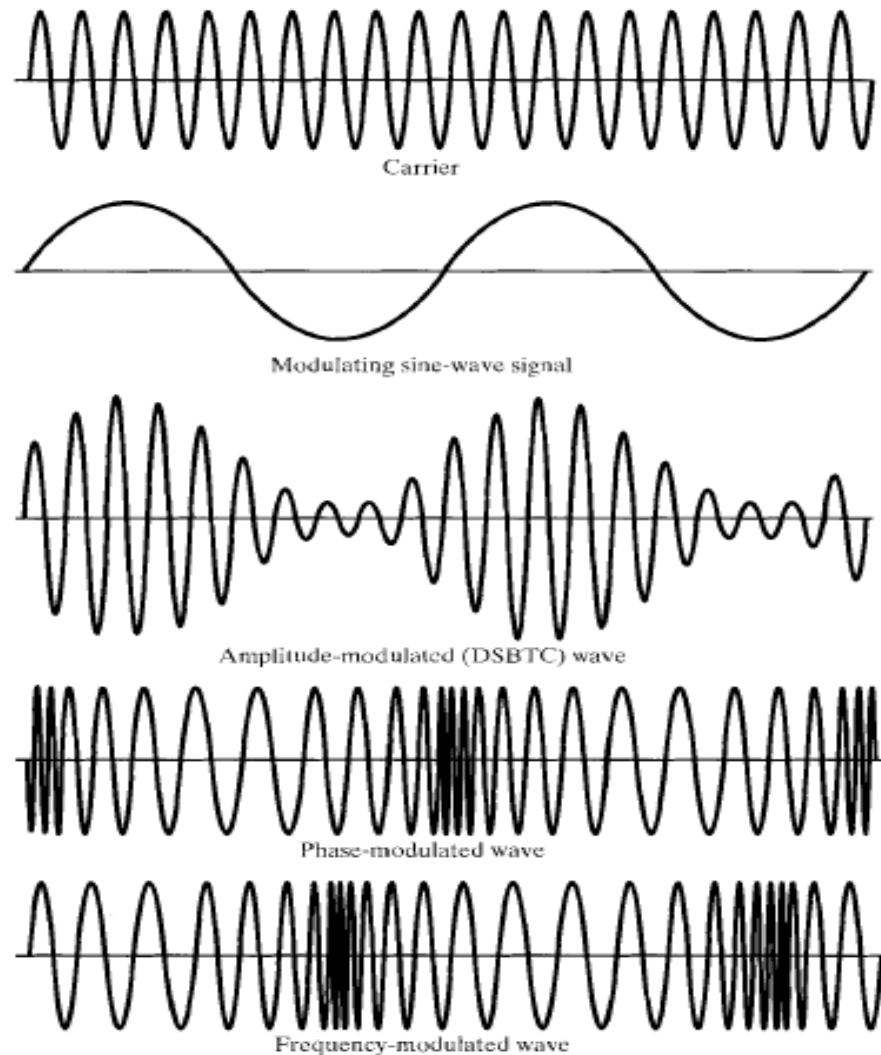


Transmitting
signal:

1 1 1 0 0 0 0 0 1 1

Dữ liệu liên tục tín hiệu liên tục

- Kết hợp tín hiệu $m(t)$ và sóng mang có tần số F_c thành một tín hiệu tập trung xung quanh F_c
- Cho phép chuyển tín hiệu trên một tần số khác phù hợp với kênh truyền
- Cho phép dồn kênh bằng các tần số sóng mang khác nhau
- 3 phương pháp chính dựa vào đặc điểm của tín hiệu
 - Điều biên
 - Điều tần
 - Điều chế góc pha



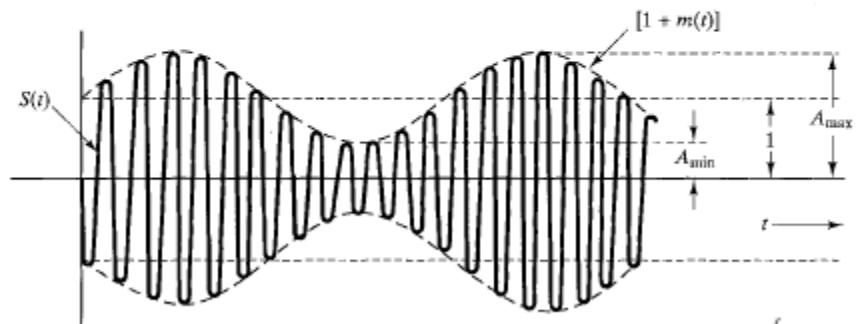
Điều biến

- Biến đổi biên độ sóng mang theo đầu vào
- Nếu đầu vào cũng là hình sin
 - Tín hiệu đầu ra sẽ có hai thành phần lệch với tần số sóng mang một khoảng bằng tần số đầu vào
 - $N_a < 1$ điều biến hợp lệ
 - $N_a > 1$ mất thông tin
- Giải thông=2 lần giải thông đầu vào
- Điều biến một chiều: 1 lần giải thông

$$s(t) = [1 + m(t)] \cos 2\pi f_c t$$

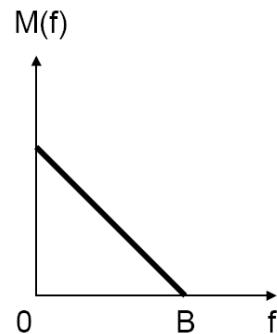


(a) Sinusoidal modulating wave

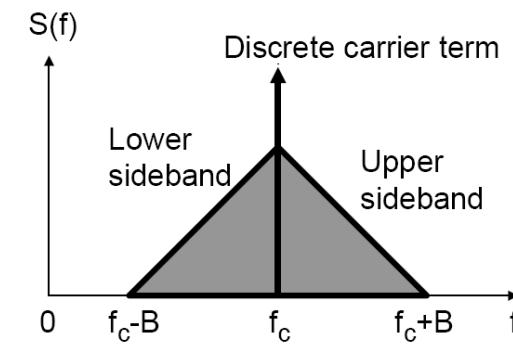


(b) Resulting AM signal

$$P_t = P_c \left(1 + \frac{n_a^2}{2}\right)$$



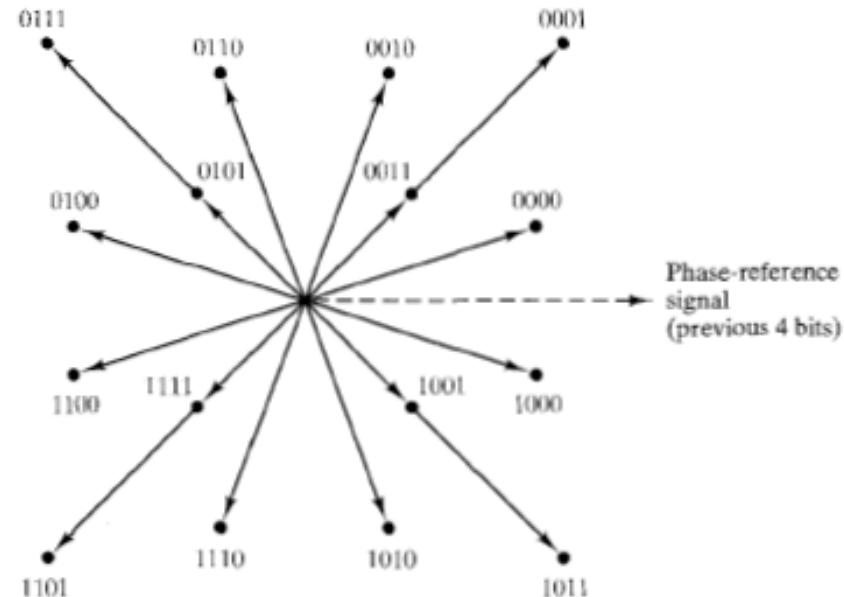
M(f)



S(f)

Bài tập-04

- Biểu diễn phương pháp điều chế pha-biên độ sau bằng công thức
- Tốc độ ký hiệu là 2400 baud. Tốc độ dữ liệu là bao nhiêu?





ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chương 3. Tầng liên kết dữ liệu

Nội dung

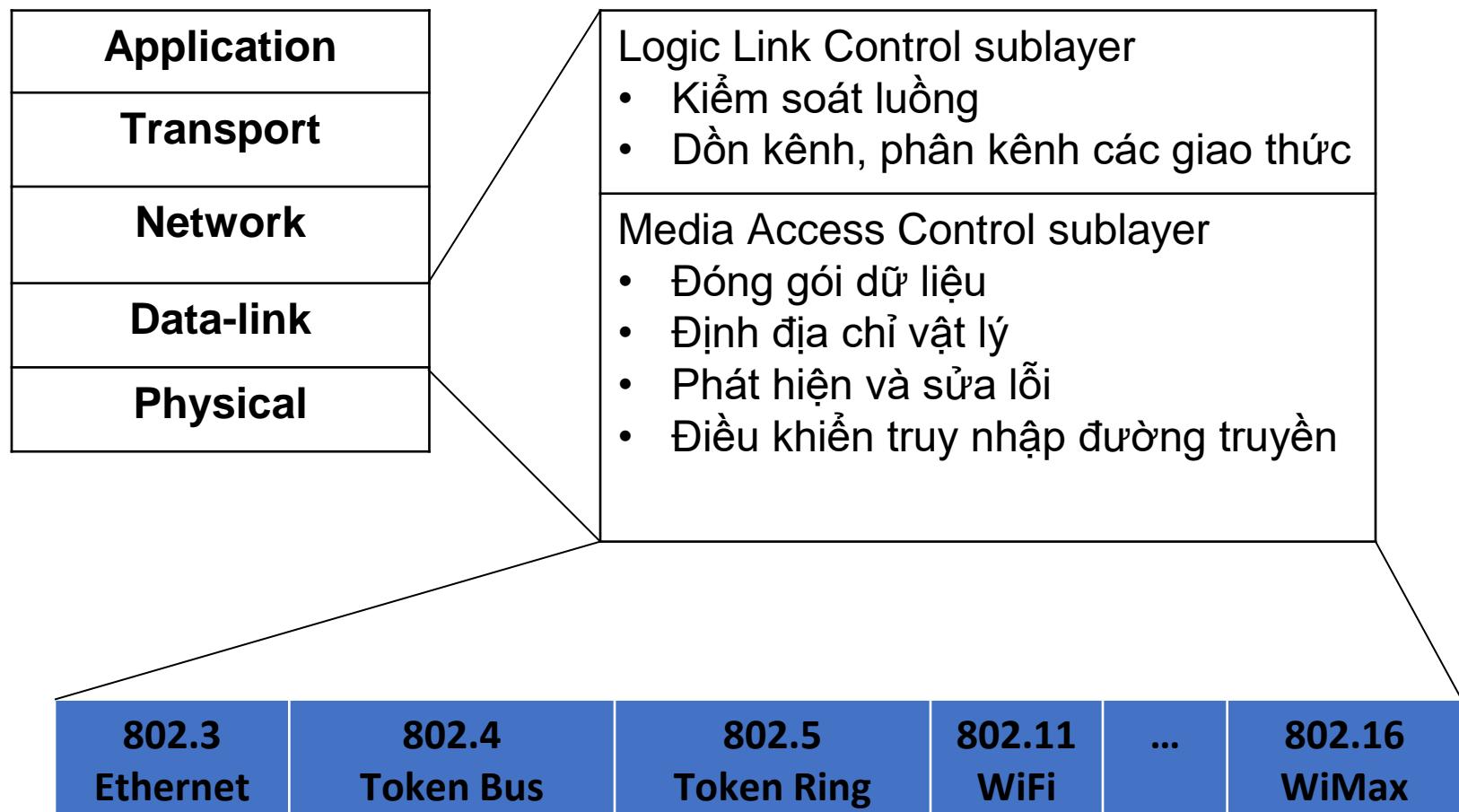
1. Tổng quát về tầng liên kết dữ liệu
2. Kiểm soát lỗi
3. Điều khiển truy nhập đường truyền
4. Chuyển tiếp dữ liệu
5. Mạng cục bộ (LAN)
6. Mạng diện rộng (WAN)



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

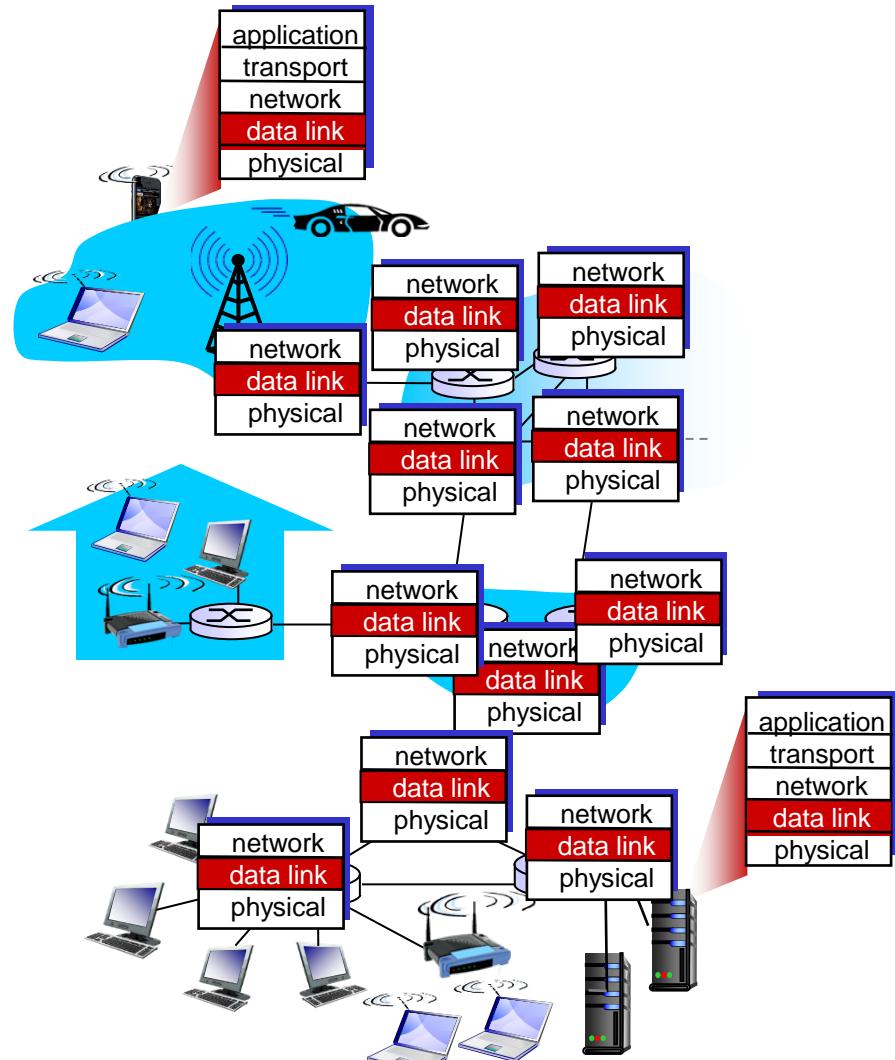
1. Tổng quan

Tầng liên kết dữ liệu trên mô hình TCP/IP



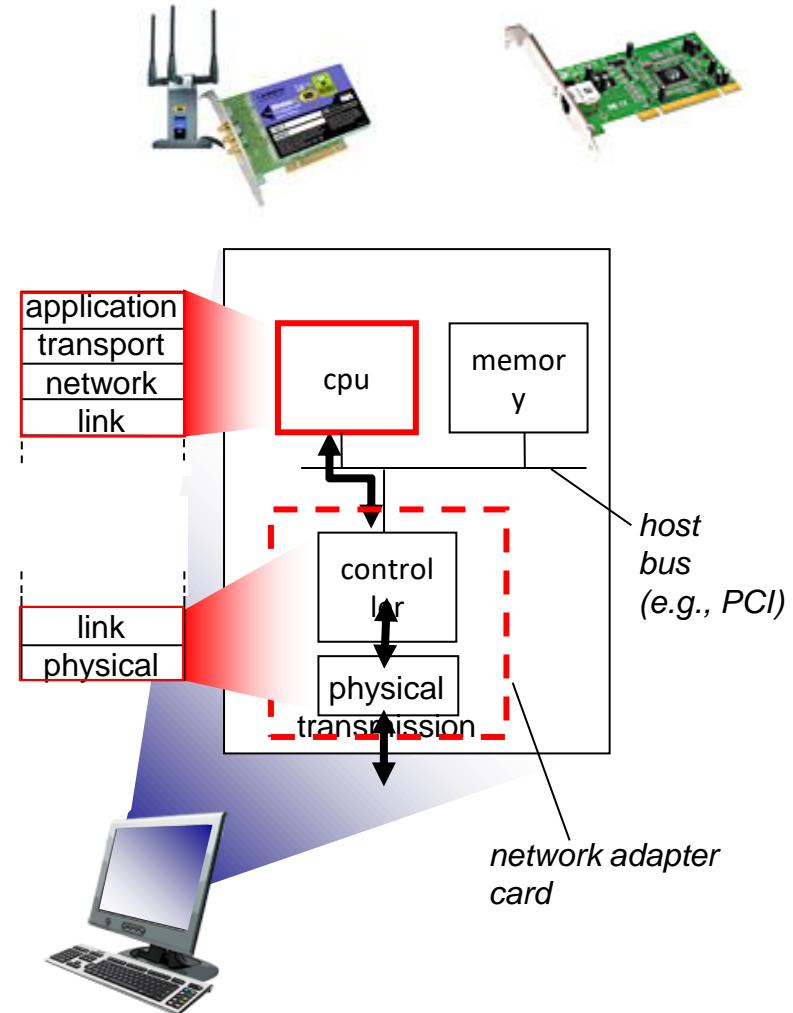
Triển khai trên hệ thống mạng

- Điều khiển truyền dữ liệu trên liên kết vật lý giữa 2 nút mạng kế tiếp
- Triển khai trên mọi nút mạng
- Các thức triển khai và cung cấp dịch vụ phụ thuộc vào đường truyền(WiFi, Wimax, 3G, cáp quang, cáp đồng...)
- Truyền thông tin cậy (cơ chế giống TCP nhưng đơn giản hơn) hoặc không
- Đơn vị truyền: frame (khung tin)



Triển khai trên các nút mạng

- Tầng liên kết dữ liệu được đặt trên các mạng (NIC-Network Interface Card) hoặc trên chip tích hợp
- Cùng với tầng vật lý
- NIC được kết nối với hệ thống bus



Các chức năng chính

- Đóng gói:
 - Đơn vị dữ liệu: khung tin (frame)
 - Bên gửi: thêm phần đầu cho gói tin nhận được từ tầng mạng
 - Bên nhận: bỏ phần đầu, chuyển lên tầng mạng
- Địa chỉ hóa: sử dụng địa chỉ MAC
- Điều khiển truy nhập đường truyền: nếu mạng đa truy nhập, cần có giao thức điều khiển đa truy nhập
- Kiểm soát luồng: đảm bảo bên nhận không bị quá tải
- Kiểm soát lỗi: phát hiện và sửa lỗi bit trong các khung tin
- Chế độ truyền: simplex, half-duplex, full-duplex

Chế độ truyền

- Simplex (Đơn công): Mỗi thiết bị trên liên kết vật lý chỉ thực hiện 1 chức năng (phát tín hiệu hoặc thu tín hiệu). Ví dụ: Hệ thống truyền hình tương tự, truyền thanh
- Half-duplex (Bán song công): Mỗi thiết bị trên liên kết vật lý có 2 chức năng phát và thu, nhưng chỉ thực hiện 1 chức năng tại mỗi thời điểm. Ví dụ: truyền thông bằng bộ đàm
- Full-duplex (Song công đầy đủ): Mỗi thiết bị trên liên kết vật lý có thể phát và thu đồng thời. Ví dụ: truyền thông trong hệ thống mạng, hệ thống điện thoại

Định danh: địa chỉ MAC

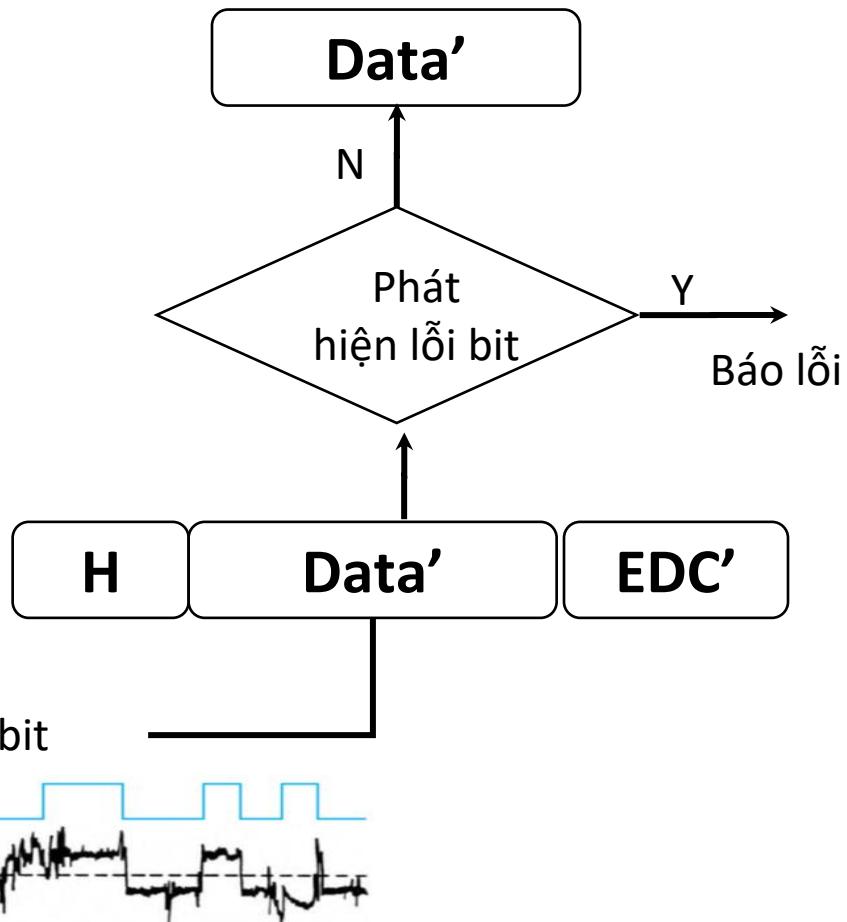
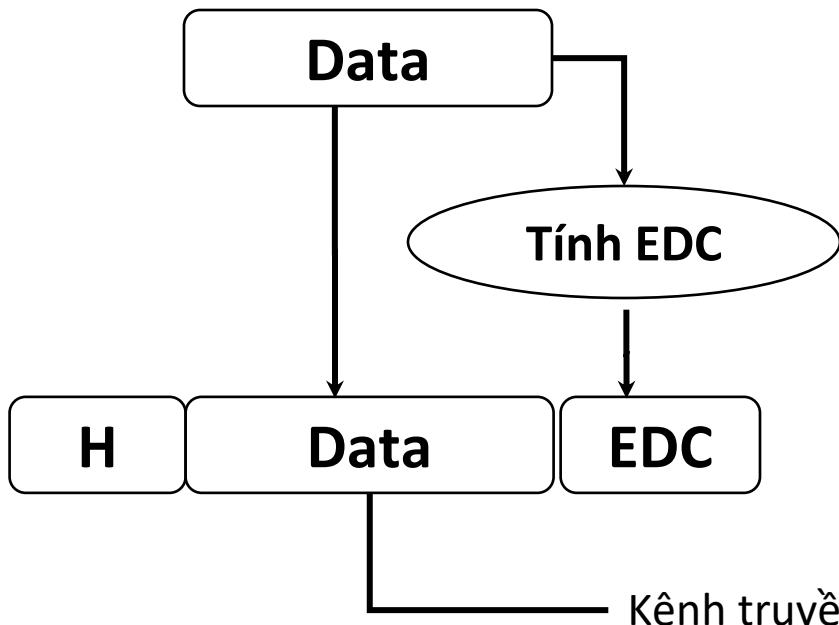
- Địa chỉ MAC: 48 bit, được quản lý bởi IEEE
- Mỗi cổng mạng được gán một MAC
 - Không thể thay đổi địa chỉ vật lý
- Không phân cấp, có tính di động
 - Không cần thay đổi địa chỉ MAC khi host chuyển sang mạng khác
- Địa chỉ quảng bá trong mạng LAN:
FF-FF-FF-FF-FF-FF



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

2. Kiểm soát lỗi

Nguyên lý phát hiện lỗi



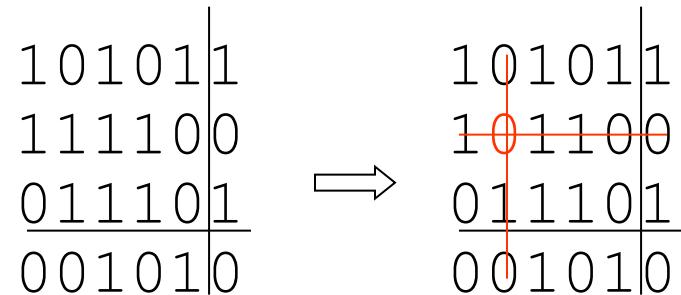
EDC: Error Detection Code

- Mã parity
- Mã checksum
- Mã vòng CRC (được sử dụng chủ yếu trong các giao thức trên tầng liên kết dữ liệu)

Mã chẵn lẻ (parity)

- Mã đơn:
 - Phát hiện lỗi bít đơn
 - Không phát hiện nếu nhiều hơn 1 bit lỗi
- Mã hai chiều:
 - Phát hiện và sửa lỗi bít đơn
- Phát hiện lỗi trên nhiều bit?

0111000110101011	0
------------------	---



$$f_{edc}(P_{gửi}) \neq f_{edc}(P_{nhận}) \quad \forall P_{gửi} \neq P_{nhận}$$

(khái niệm hàm băm!!!)

Bài tập mã chẵn lẻ

- Tìm bit parity chẵn của các chuỗi dữ liệu sau
 - 0100 1100
 - 0011 0111
 - 1100 0011
- Tìm bit parity lẻ của các chuỗi dữ liệu sau
 - 1110 1100
 - 0010 0111
 - 1101 0010

Bài tập mã chẵn lẻ

- Tìm bit parity chẵn của các chuỗi dữ liệu sau
 - 0100 1100 → 1
 - 0011 0111 → 1
 - 1100 0011 → 0
- Tìm bit parity lẻ của các chuỗi dữ liệu sau
 - 1110 1100 → 0
 - 0010 0111 → 1
 - 1101 0010 → 1

Mã Checksum

- Gửi:
 - Chia dữ liệu thành các phần có kích thước n bit
 - Tính tổng các phần. Nếu kết quả tràn quá n bit, cộng các bit tràn vào phần kết quả
 - Đảo bit kết quả cuối cùng được checksum
 - Truyền checksum kèm theo dữ liệu
- Nhận:
 - Tách dữ liệu và checksum
 - Chia dữ liệu thành các phần có kích thước n bit
 - Tính tổng các phần và checksum. Nếu kết quả tràn quá n bit, cộng các bit tràn vào phần kết quả
 - Nếu kết quả cuối xuất hiện bit 0 → dữ liệu bị lỗi

Mã checksum: Ví dụ

Dữ liệu: 0011 0110 1000

Tính checksum 4 bit:

$$\begin{array}{r} 0011 \\ + 0110 \\ \hline 1000 \\ \hline \text{Bit} & 10001 \\ \text{tràn} & \curvearrowright 1 \\ \hline 0010 \end{array}$$

Đảo bit → mã checksum: 1101

Chuỗi bit truyền: 0011 0110 1000 **1101**

Mã checksum: Xử lý nhận

Nhận chuỗi bit: 0011 0110 1000 **1101**

Kiểm tra:

$$\begin{array}{r} 0011 \\ 0110 \\ + \quad 1000 \\ \hline 1101 \\ \hline 11110 \\ \text{Bít} \quad \rightarrow 1 \\ \text{tràn} \end{array}$$

1111 → Không có lỗi bit

Bài tập checksum

- Tìm checksum 4-bit của các chuỗi dữ liệu sau
 - 1100 0001 0101 → 1100
 - 1011 1001 1001 → 0001
 - 0100 0010 1100 → 1100

Bài tập checksum

$$\begin{array}{r} 1100 \\ + 0001 \\ \hline 0101 \\ \hline 10010 \\ + \quad \swarrow \\ \hline 0011 \end{array}$$

Đảo bit → 1100

$$\begin{array}{r} 1011 \\ + 1001 \\ \hline 1001 \\ \hline 11101 \\ + \quad \swarrow \\ \hline 1110 \end{array}$$

Đảo bit → 0001

Mã vòng CRC (Cyclic Redundancy Check)

- Phía gửi
 - Chọn 1 đa thức sinh bậc k
 - Biểu diễn đa thức dưới dạng chuỗi bit P
 - Thêm k bit 0 vào frame dữ liệu F được F_k
 - Chia F_k cho P, lấy phần dư R (có kích thước k bit)
 - Ghép phần dư vào chuỗi dữ liệu được FR
- Phía nhận : lấy FR chia cho P
 - Nếu chia hết → truyền đúng
 - Nếu chia có dư, căn cứ vào số dư (syndrom) để phát hiện và sửa lỗi (nếu được)

Mã CRC – Ví dụ

Frame : 1101011011

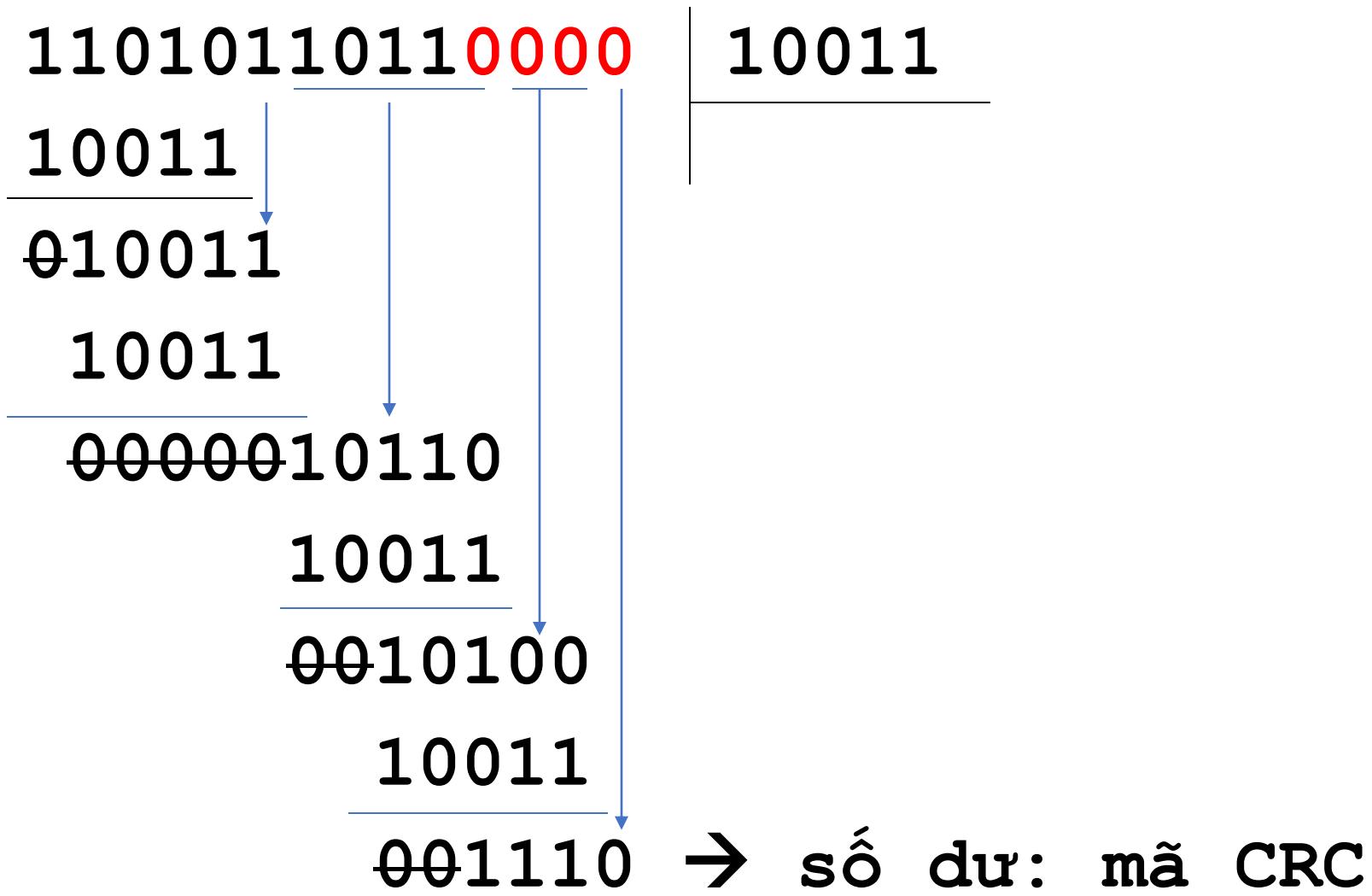
Generator : $G(x) = x^4 + x + 1 \rightarrow P = 10011$

Dividend : $F_k = 1101011011\textcolor{red}{0000}$

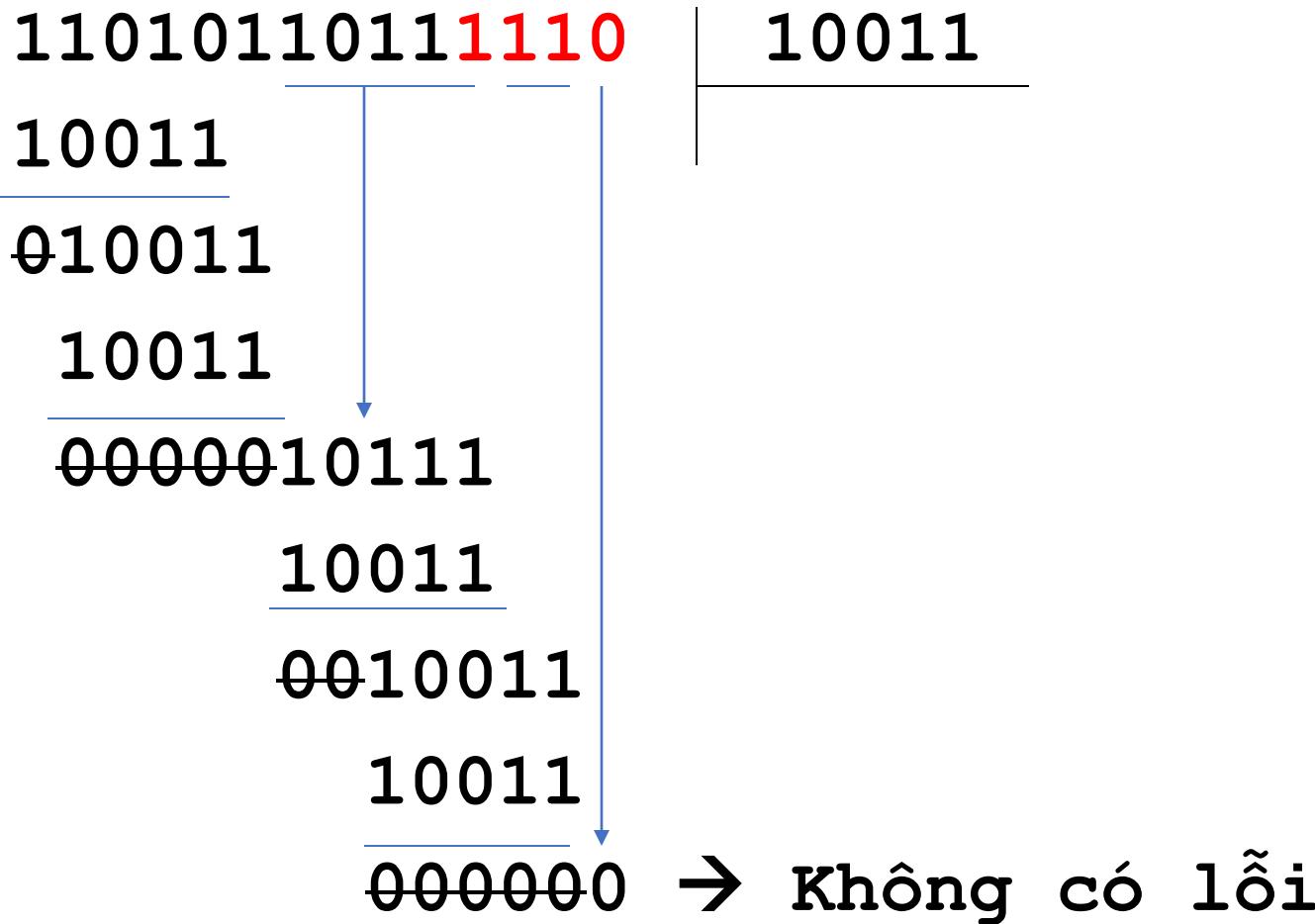
$R = F_k \bmod P = 1110$

Send : 1101011011 $\textcolor{red}{1110}$

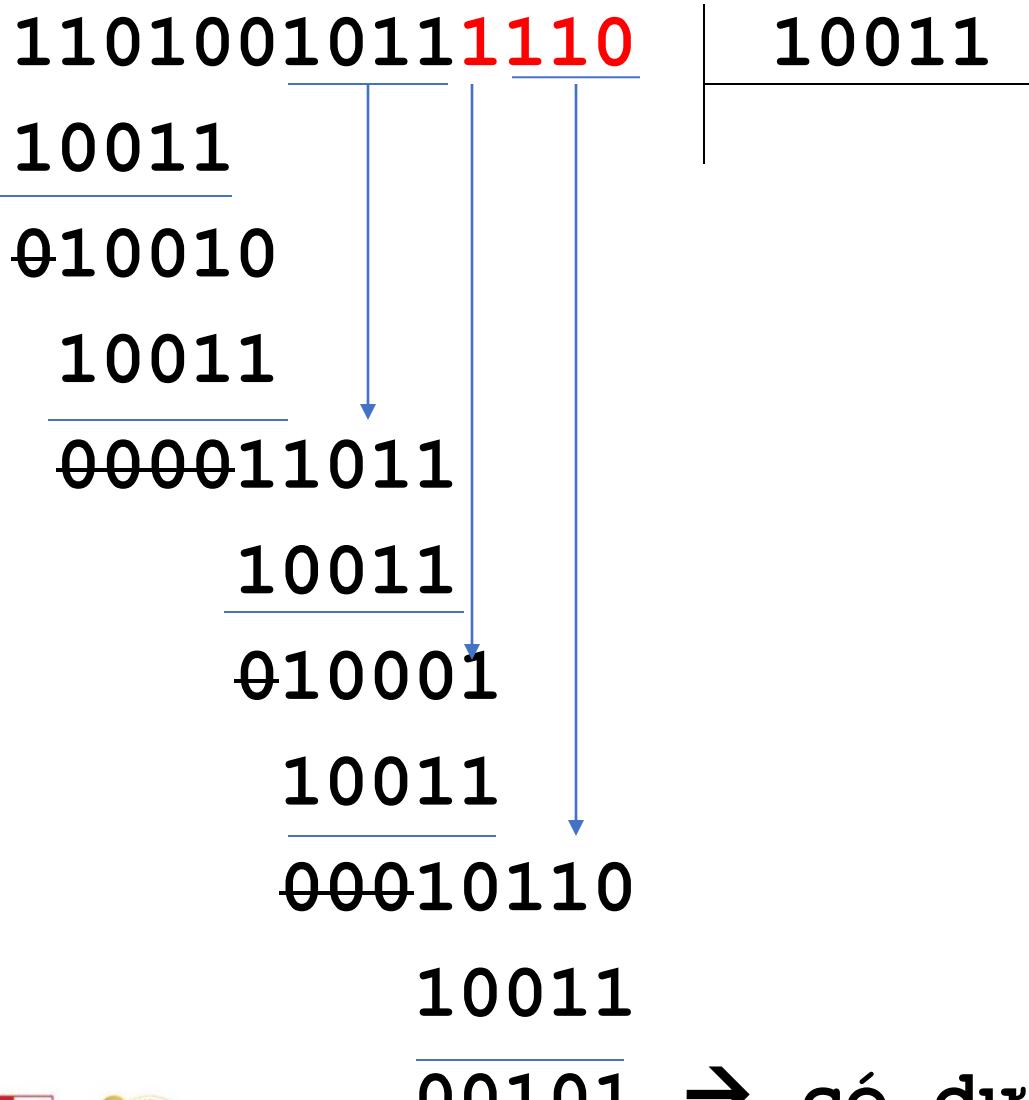
Mã CRC – Ví dụ



Mã CRC - Kiểm tra 11010110111110



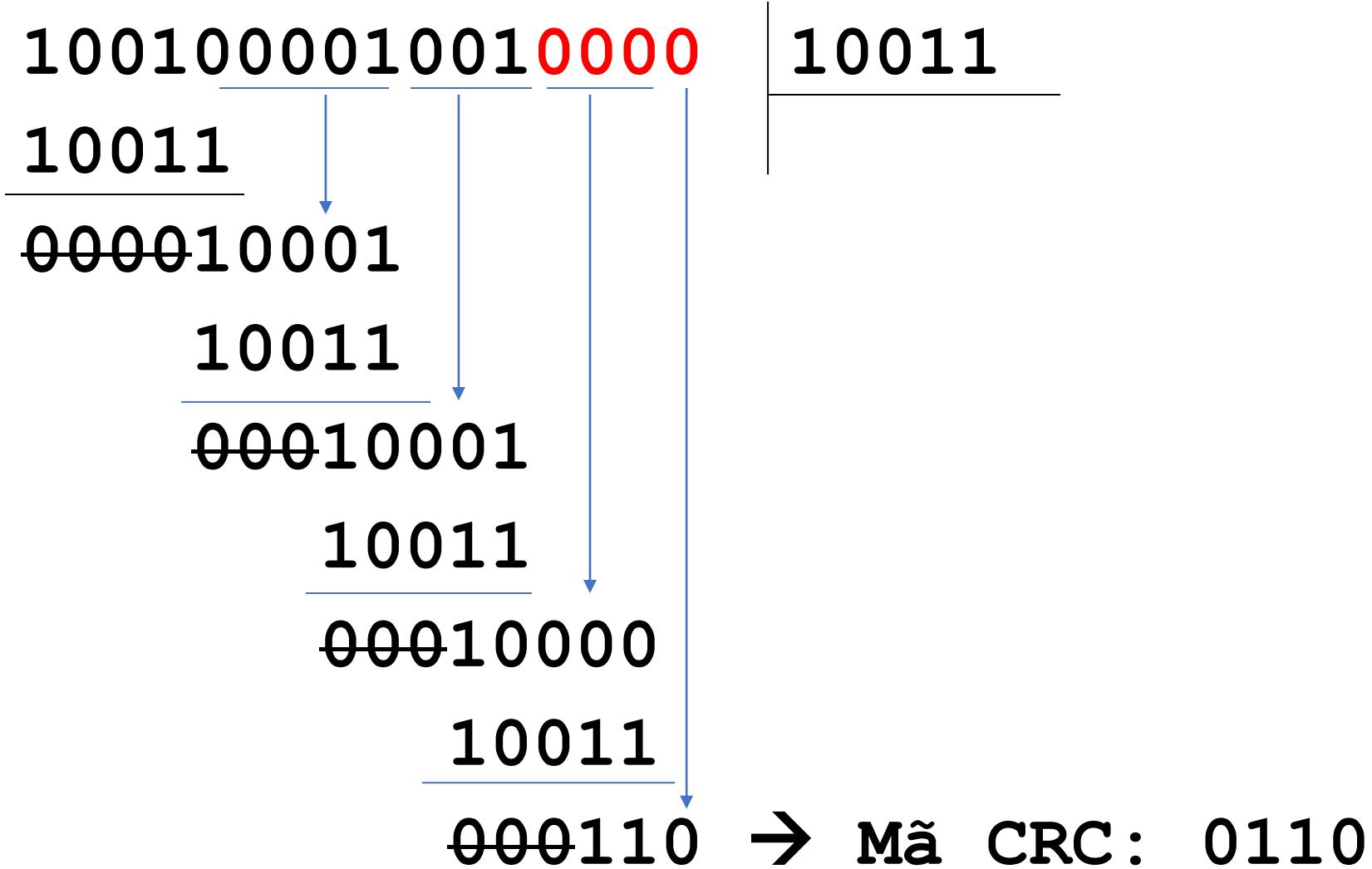
Mã CRC - Kiểm tra 11010010111110



Bài tập mã CRC

- Tìm mã CRC với các điều kiện sau
 - Chuỗi dữ liệu là 1001 0000 1001 và $G = 10011 \rightarrow CRC = 0010 / 1111$
 - Chuỗi dữ liệu là 1000 1100 1101 và đa thức $G = x^4 + 1 \rightarrow CRC = 1001 / 1000$

Bài tập mã CRC (1)



Làm gì sau khi phát hiện lỗi

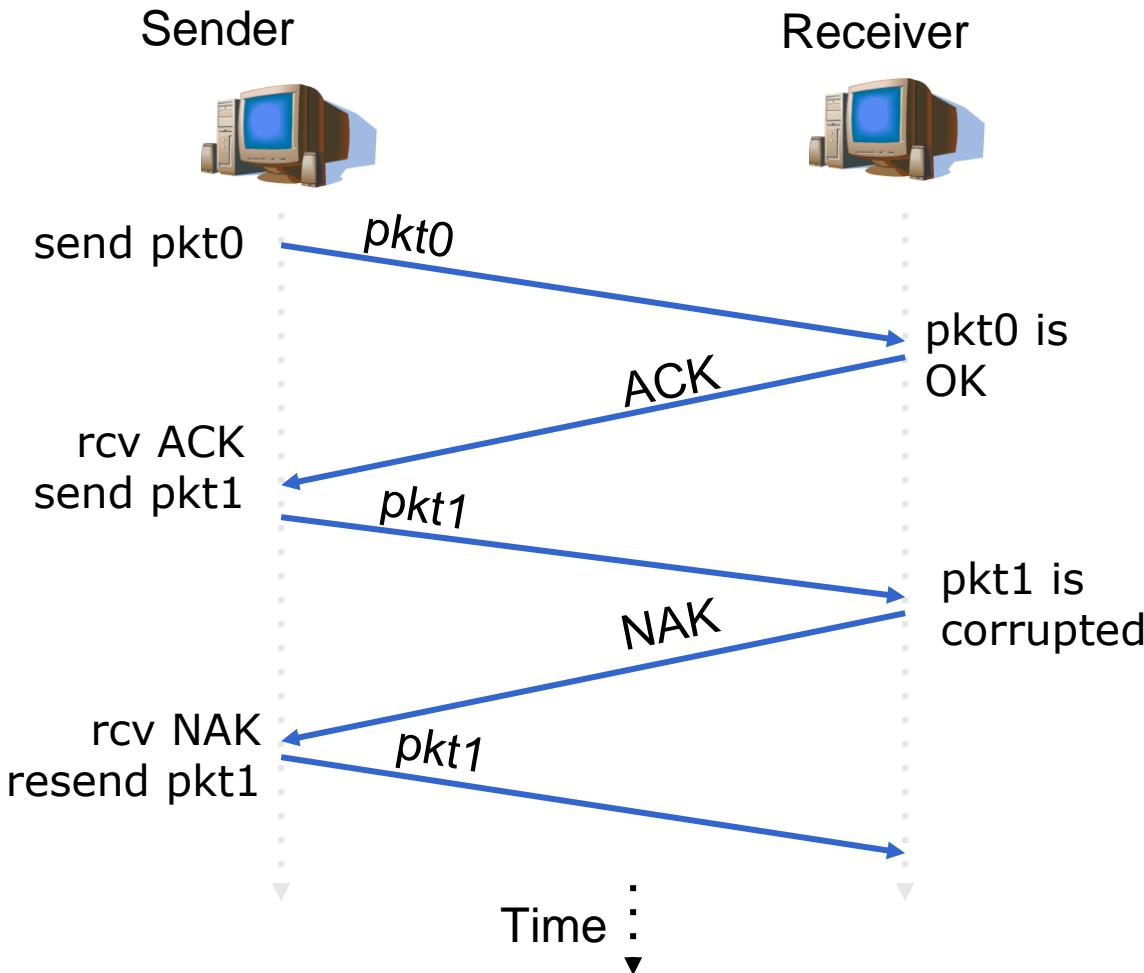
- Các loại lỗi
 - Mất khung dữ liệu
 - Khung dữ liệu bị lỗi
 - Thông báo lỗi bị mất
- Làm thế nào để báo cho bên gửi?
 - Báo nhận: ACK (*acknowledgements*):
 - Báo lỗi: NAK (*negative acknowledgements*), báo cho người gửi về một gói tin bị lỗi
- Phản ứng của bên gửi?
 - Truyền lại nếu là NAK
 - Nếu không nhận được cả ACK/NAK? Timeout

Các kỹ thuật báo nhận, phát lại

- **ARQ: automatic repeat request.**
- Có 3 phiên bản chuẩn hóa
 - Dừng và chờ (Stop and Wait) ARQ
 - Chỉ phát tiếp khi nhận được ACK của gói đã phát
 - Hoặc nếu gặp timeout mà chưa nhận được ACK thì phát lại gói vừa phát.
 - Quay lại N (Go Back N) ARQ
 - Sẽ xem ở tầng 4
 - Loại bỏ chọn lọc (Selective Reject) ARQ
 - Sẽ xem ở tầng 4

Stop-and-wait ARQ

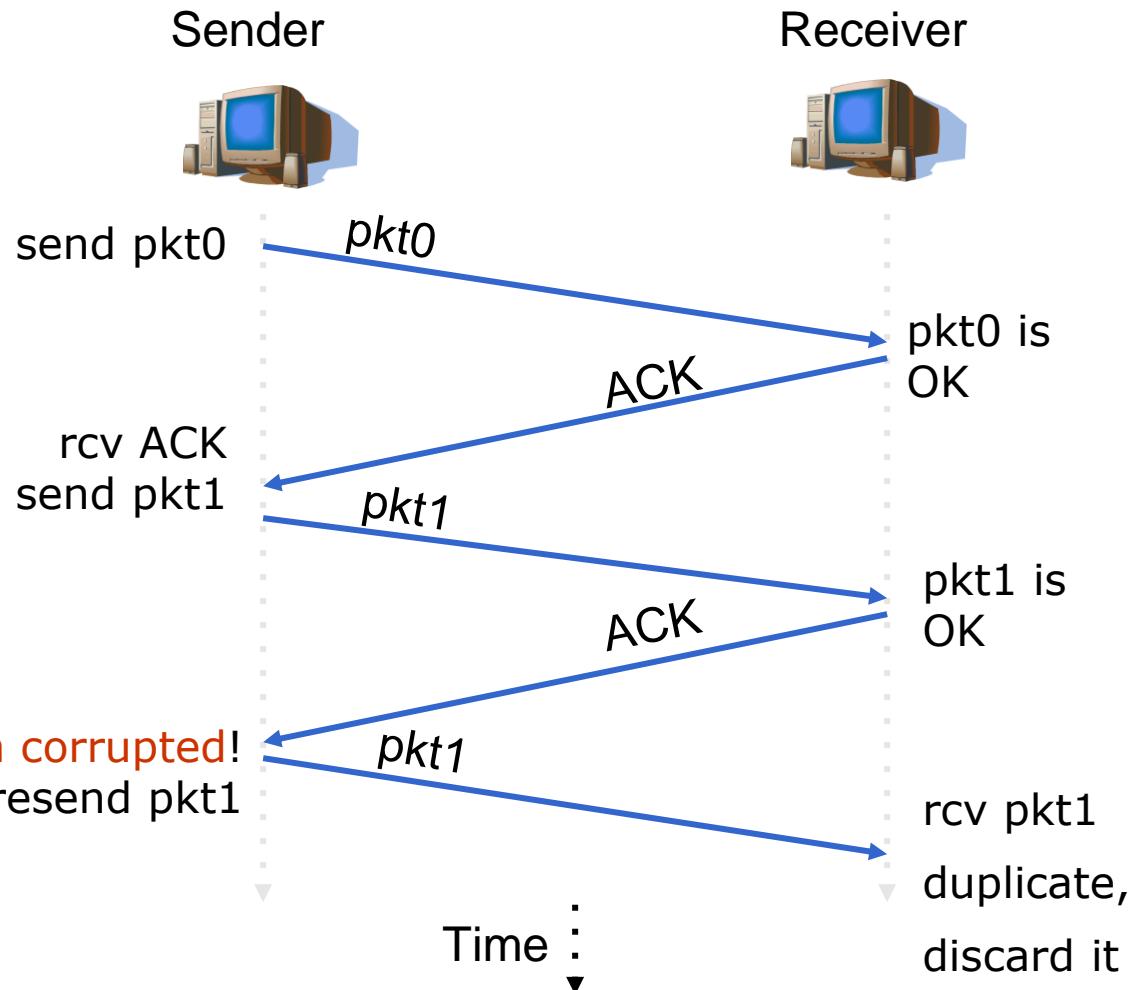
Trường hợp thông thường



Stop-and-wait ARQ

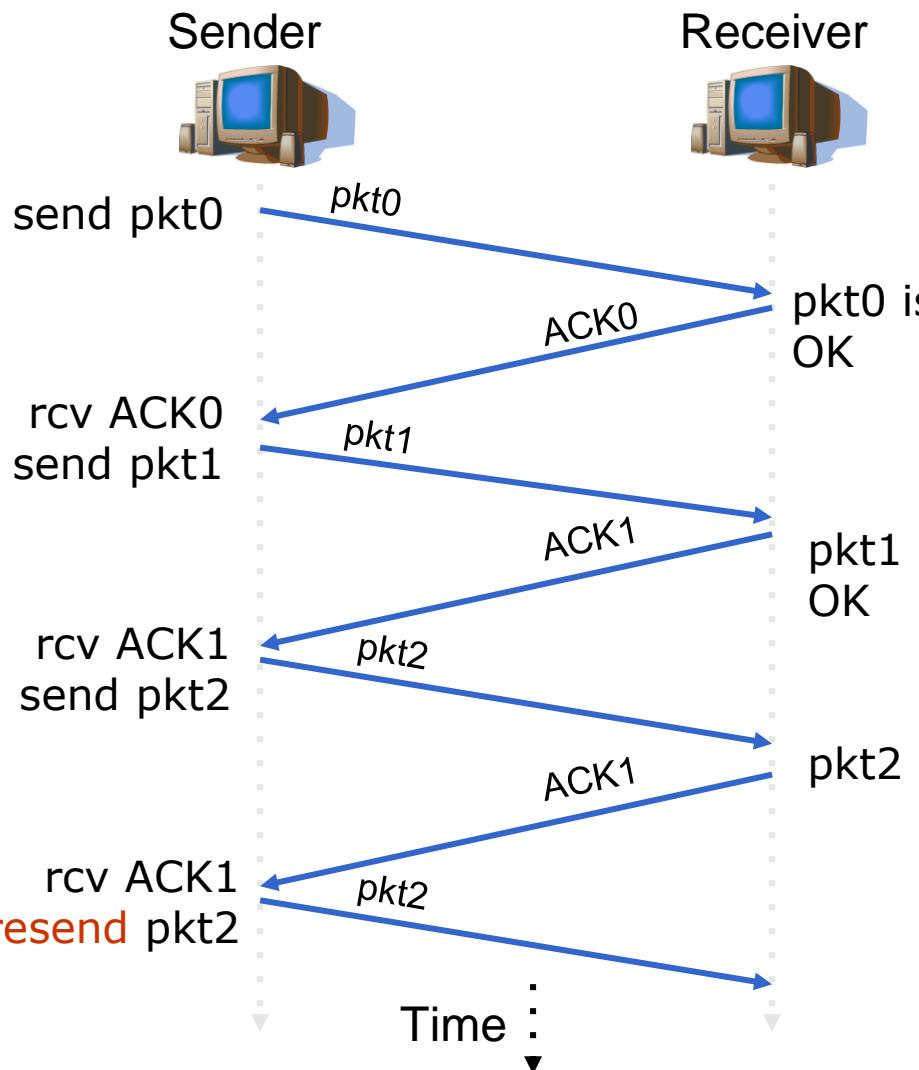
Trường hợp lỗi ACK/NAK

- Cần truyền lại
- Xử lý việc lặp gói tin ntn?
- Thêm Seq.#



Stop-and-wait ARQ

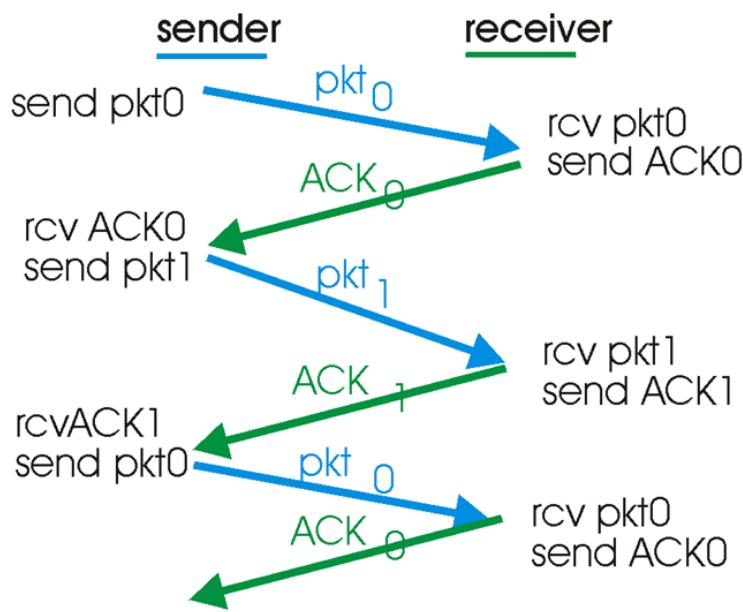
Giai pháp không dùng NAK



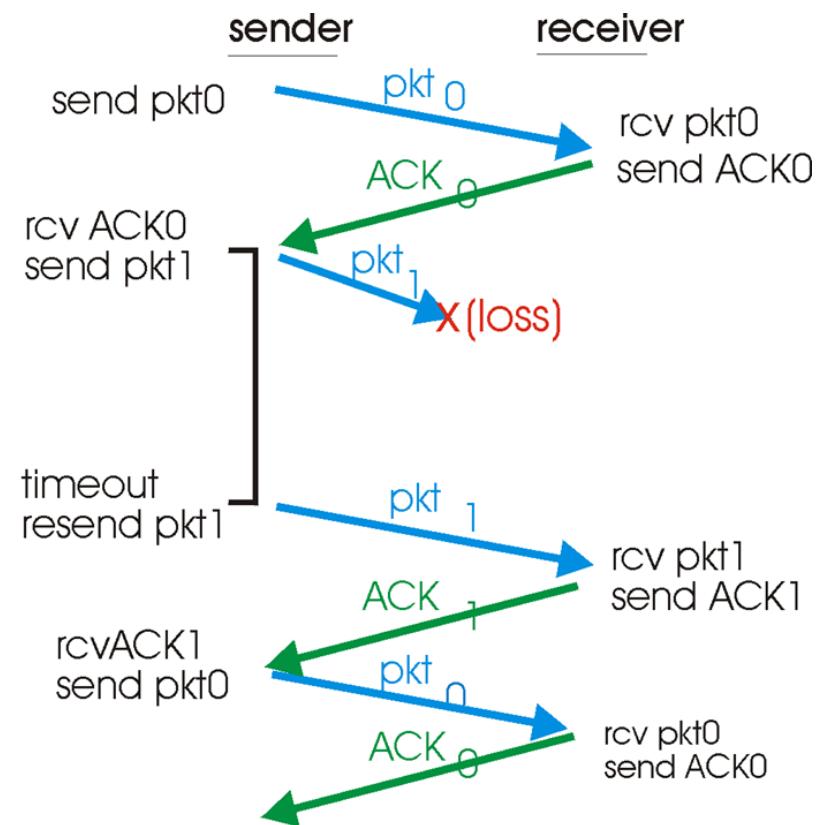
Stop-and-wait ARQ: Khi ACK bị mất

- Dữ liệu và ACK có thể bị mất
 - Nếu không nhận được ACK?
 - Truyền lại như thế nào?
- Cơ chế
 - Các gói được đánh số và bên gửi tự động truyền lại cho đến khi nhận được ACK của gói.
 - Chờ hết Timeout mới truyền lại
- Timeout nên dài bao lâu?
 - Ít nhất là 1 RTT (Round Trip Time)
 - Mỗi gói tin gửi đi cần 1 timer
- Nếu gói tin vẫn đến đích, chỉ ACK bị mất?
 - Phía nhận lọc gói tin dự căn cứ số hiệu gói tin trùng lặp

Minh họa ARQ

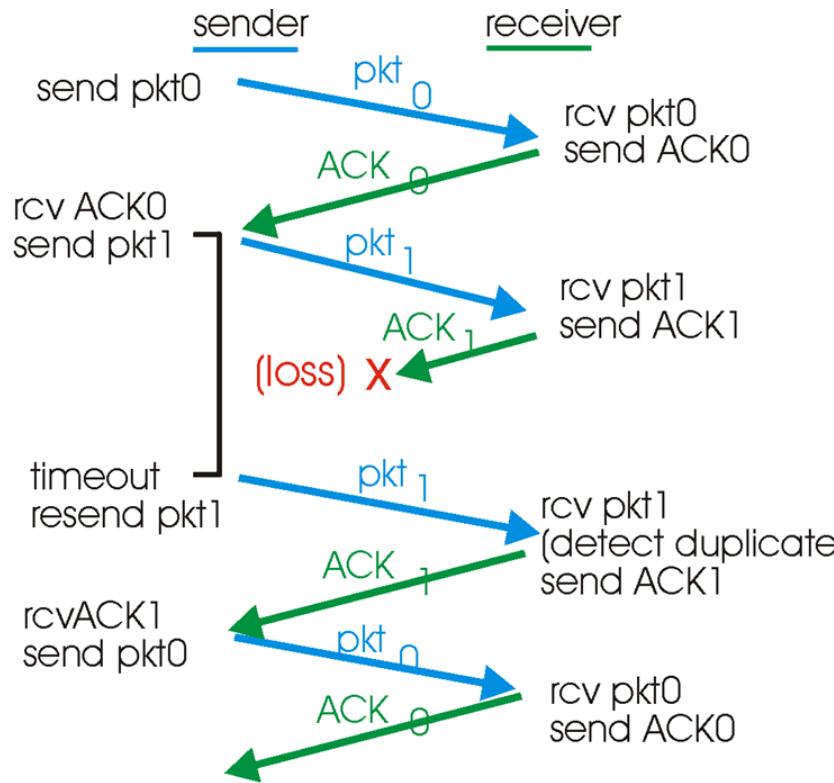


(a) operation with no loss

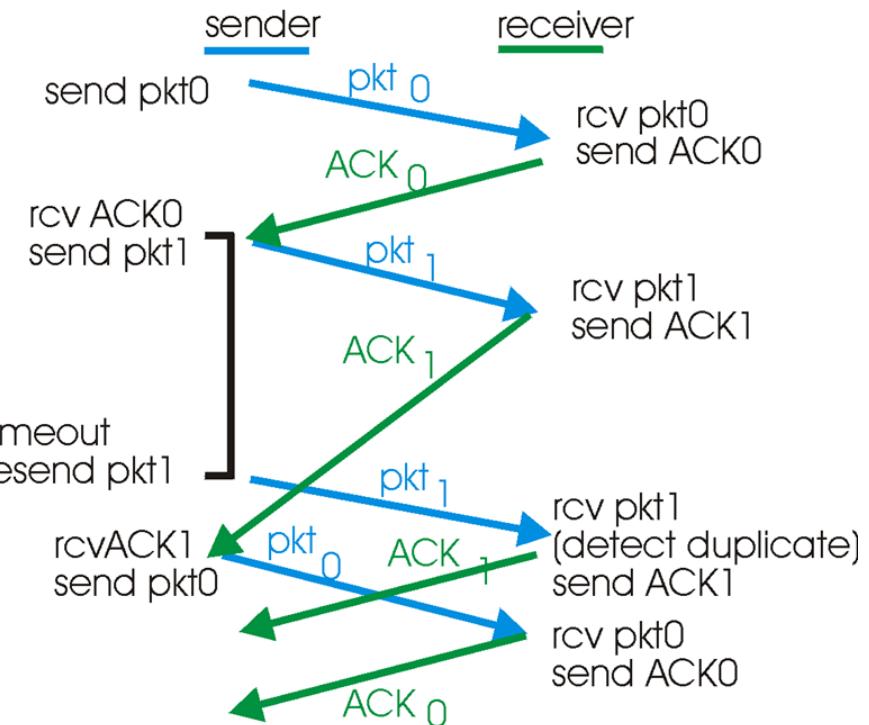


(b) lost packet

Minh họa ARQ (tiếp)



(c) lost ACK



(d) premature timeout

3. Kiểm soát luồng

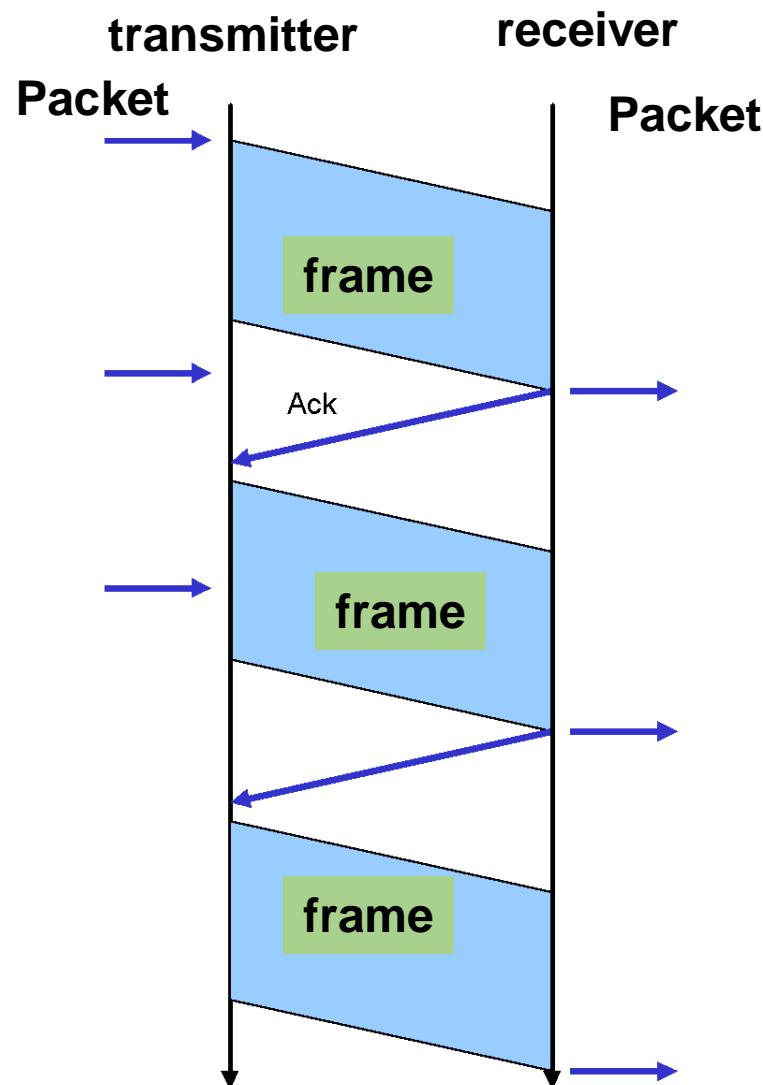
Kiểm soát luồng dữ liệu

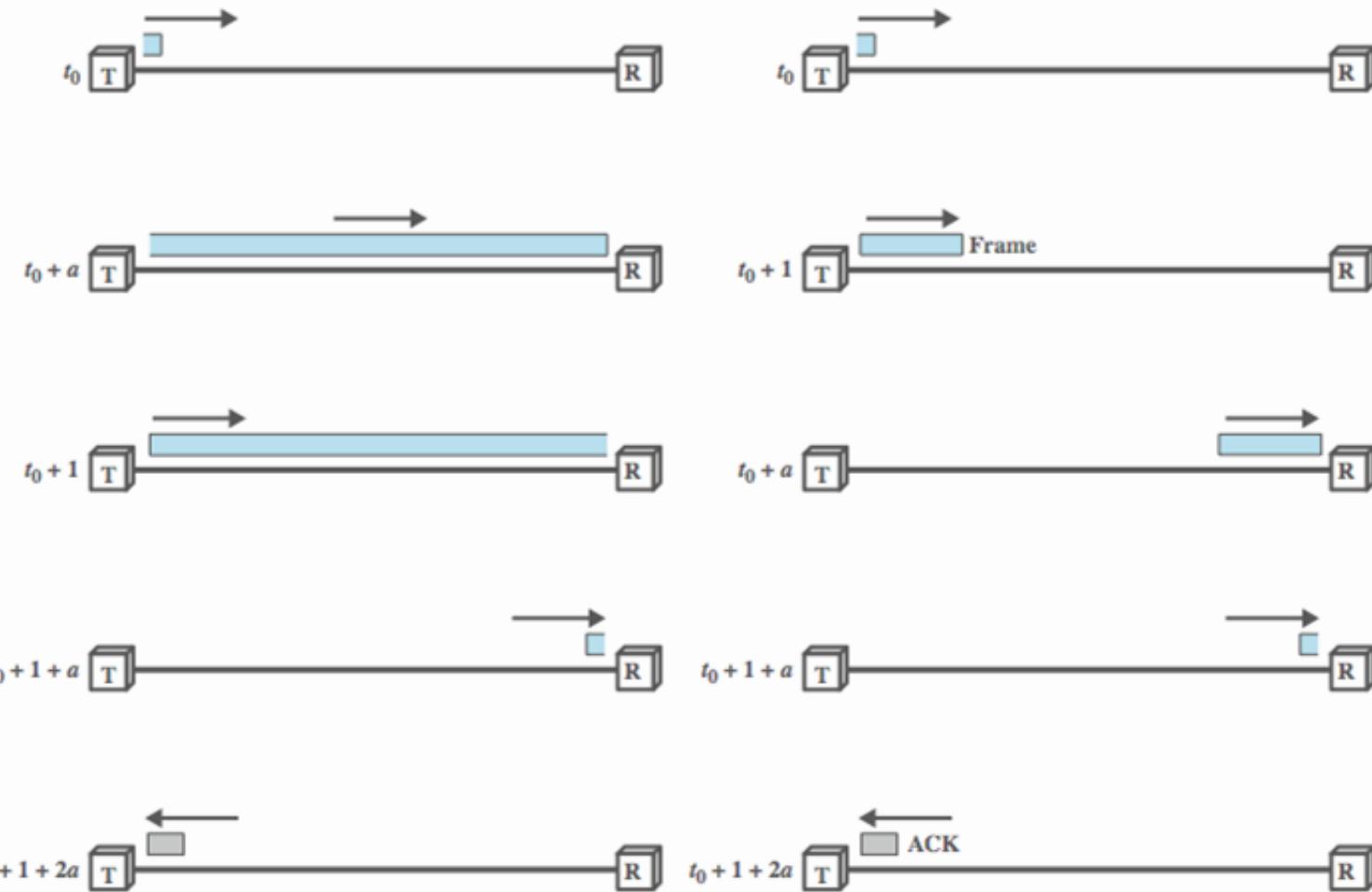
- Mục tiêu: Đảm bảo trạm gửi không làm quá tải trạm nhận
- Trạm đích
 - Lưu trữ các khung dữ liệu trong bộ nhớ đệm
 - Thực hiện một số thao tác trước khi chuyển dữ liệu lên tầng trên
 - Bộ nhớ đệm có thể bị đầy, dẫn tới mất khung dữ liệu
- Không đặt vấn đề lỗi truyền tin
 - Các khung dữ liệu luôn luôn được truyền chính xác
 - Độ trễ truyền tin không đáng kể
- Giải pháp
 - Cơ chế dừng và chờ
 - Cơ chế cửa sổ trượt (sẽ xem ở tầng 4)

Cơ chế dùng và chờ

- Nguyên tắc
 - Nguồn gửi một khung dữ liệu
 - Đích nhận khung dữ liệu, xử lý, sau đó thông báo sẵn sàng nhận các khung dữ liệu tiếp theo bằng một thông báo báo nhận (acknowledgement)
 - Nguồn phải chờ đến khi nhận được báo nhận mới truyền tiếp khung dữ liệu tiếp theo

Cơ chế dừng và chờ





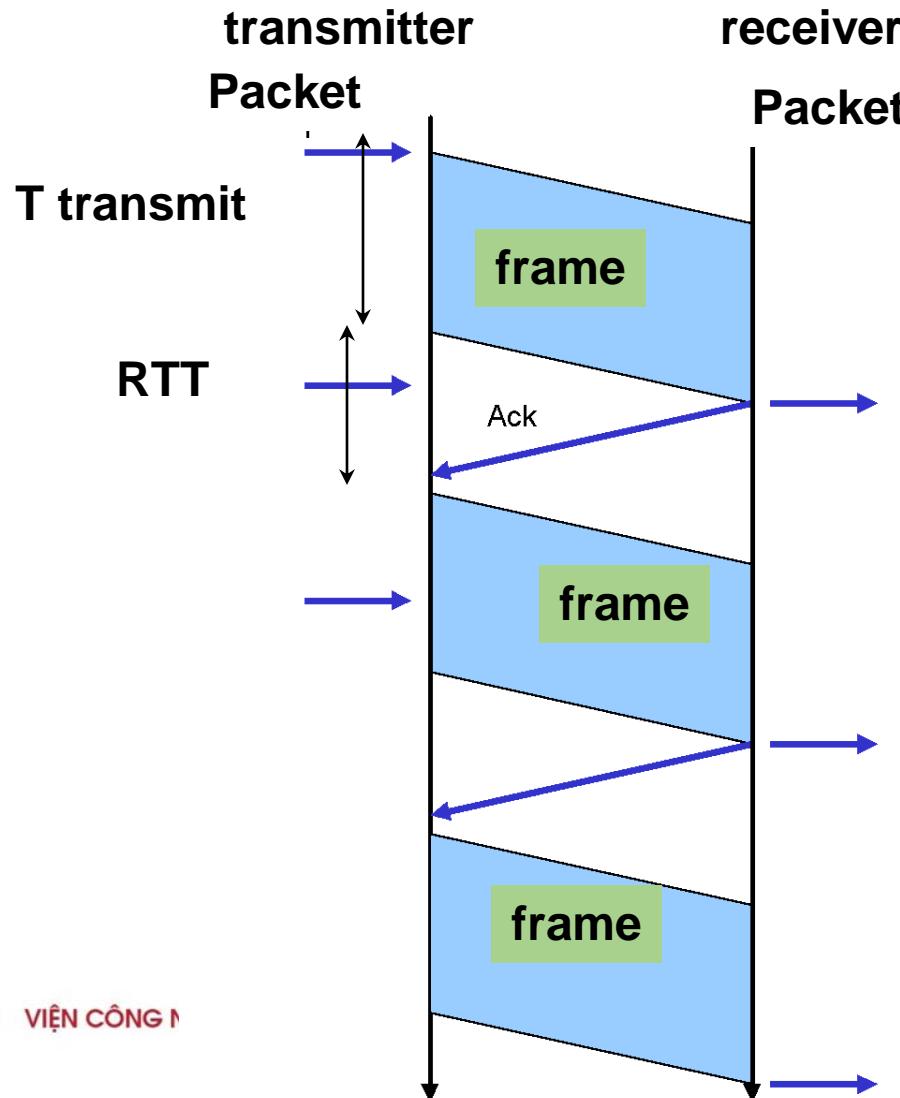
Cơ chế dừng và chờ

- **Ưu điểm**
 - Đơn giản, đặc biệt thích hợp với các khung dữ liệu lớn
- **Nhược điểm**
 - Với các khung dữ liệu nhỏ, thời gian sử dụng đường truyền bị lãng phí
 - Không thể sử dụng các khung dữ liệu lớn một cách phổ biến
 - Bộ nhớ đệm có hạn
 - Khung dữ liệu dài khả năng lõi lớn
 - Trong môi trường truyền tin chia sẻ, không cho phép trạm nào chiếm dụng lâu đường truyền

Bài tập

- Cho một liên kết có tốc độ $R=100\text{Mbps}$
- Cần truyền 1 dữ liệu có kích thước tổng tại tầng liên kết dữ liệu là $L=100\text{KB}$
- Giả sử kích thước một khung liên kết dữ liệu là: 1KB
- RTT giữa 2 nút của liên kết là 3ms
- Hỏi thời gian truyền khi áp dụng phương pháp kiểm soát luồng Stop-and-wait.

Thời gian phát với cơ chế dừng và chờ



Thời gian phát với cơ chế dừng và chờ

- $T_{\text{tổng}} = \text{Số frame} * (T_{\text{transmit}} + RTT)$
 - $T_{\text{transmit}} (F) = L(\text{Frame}) / R$
 - $\text{Số frame} = L / L(\text{frame})$
-
- Với Số liệu đã cho
 - Số frame = $100 \text{ KB} / 1 \text{ KB} = 100$
 - $T_{\text{transmit}} (F) = 1 \text{ KB} / 100 \text{ Mbps} = 10^3 \cdot 8 / 10^8$
 $= 8 \cdot 10^{-5} (\text{s}) = 0.08 (\text{ms})$

4. Điều khiển truy nhập đường truyền

Các dạng liên kết

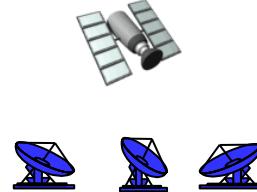
- Điểm-điểm (point-to-point): ADSL, Telephone modem, Leased line...
- Điểm-đa điểm (point-to-multipoint):
 - Mạng LAN có dạng bus, mạng LAN hình sao dùng hub
 - Mạng không dây
 - Cần giao thức điều khiển truy nhập để tránh xung đột



shared wire (e.g., cabled Ethernet)



shared RF (e.g., 802.11 WiFi)



shared RF (satellite)



humans at a cocktail party (shared air, acoustical)

Phân loại các giao thức đa truy nhập

- Phân hoạch tài nguyên sử dụng kỹ thuật chia kênh:
 - Chia tài nguyên của đường truyền thành nhiều phần nhỏ (Thời gian - TDMA, Tần số - FDMA, Mã - CDMA)
 - Chia từng phần nhỏ đó cho các nút mạng
- Truy nhập ngẫu nhiên:
 - Kênh không được chia, cho phép đồng thời truy nhập, chấp nhận là có xung đột
 - Cần có cơ chế để phát hiện và tránh xung đột
 - e.g. Pure Aloha, Slotted Aloha, CSMA/CD, CSMA/CA...
- Lần lượt:
 - Theo hình thức quay vòng
 - Token Ring, Token Bus....

Các phương pháp chia kênh

- FDMA: frequency division multiple access
- TDMA: time division multiple access
- CDMA: code division multiple access

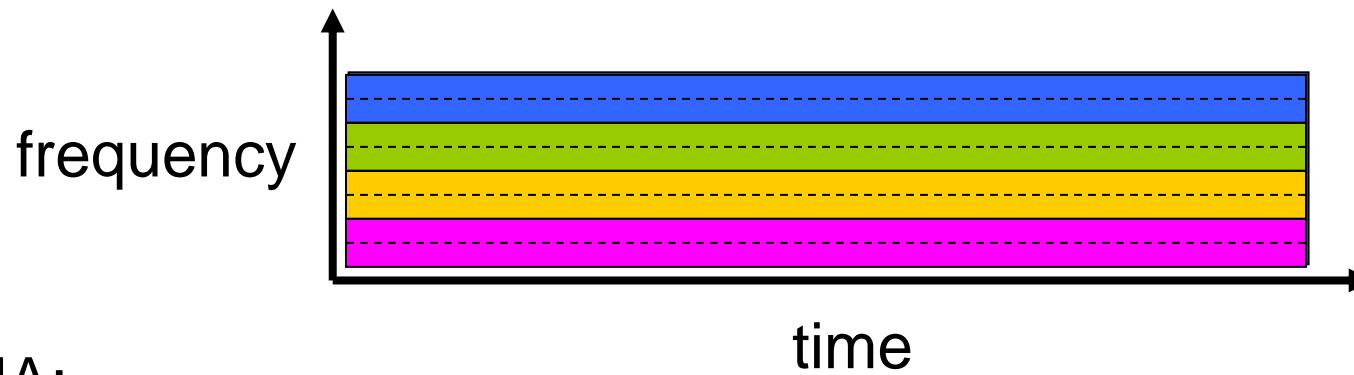
TDMA và FDMA

Ví dụ:

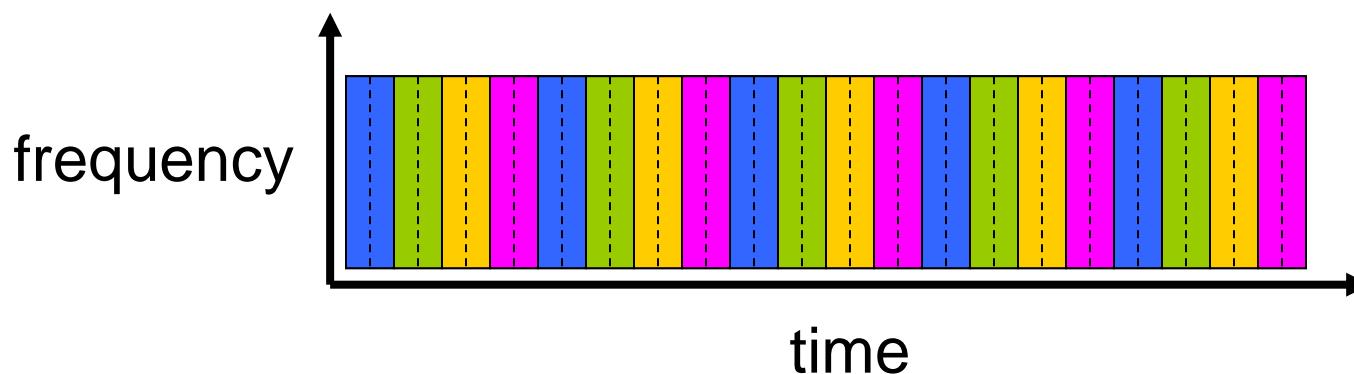
4 máy



FDMA

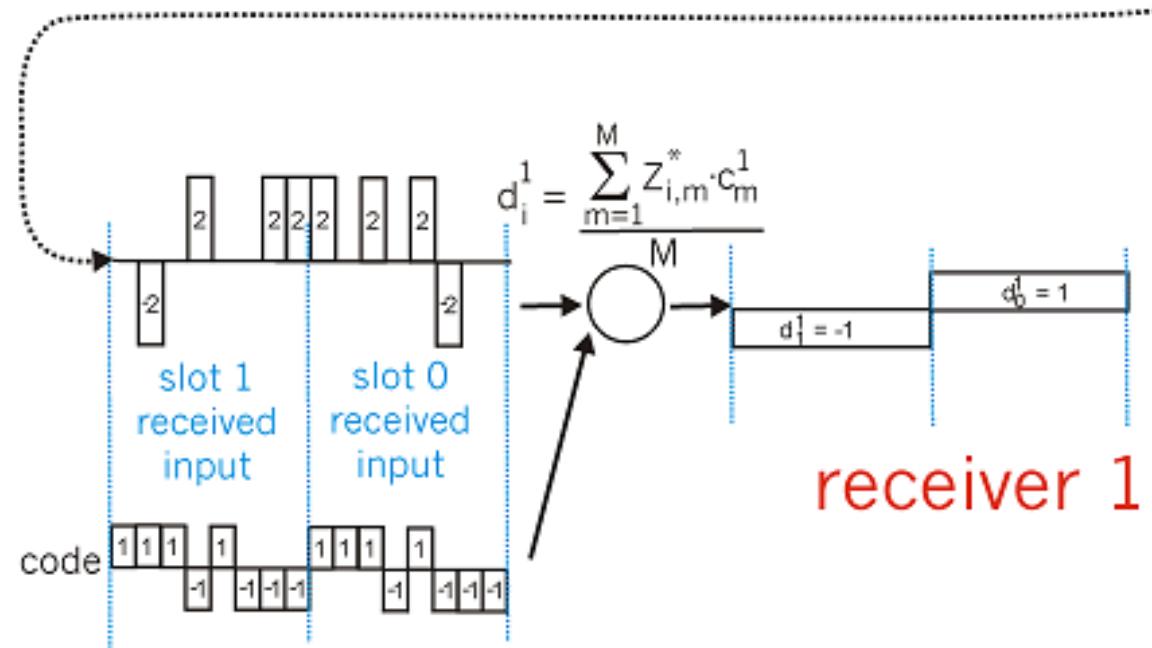
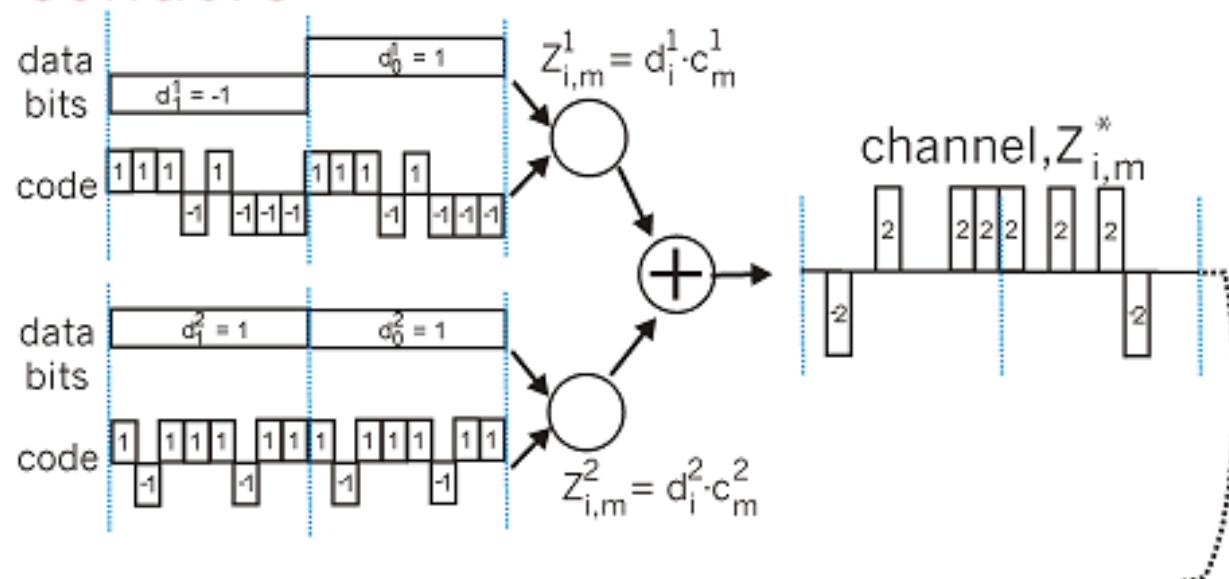


TDMA:



CDMA

senders



CDMA: đảm bảo toán học

- Các trạm truyền dữ liệu trên n tần số. Mỗi bit dữ liệu được truyền đồng thời bằng n tín hiệu (gọi là một *chips*) trên n tần số
- Mỗi trạm # m được gán một mã code **đôi một trực giao**:

$$\begin{aligned} C_m &= (c_{m1}, c_{m2}, \dots, c_{mn}) & c_{mi} &= -1, 1 \\ C_m \times C_k &= \sum (c_{mi} \times c_{ki}) = 0 & \text{với } m \neq k \\ &= n & \text{với } m = k (\text{đương nhiên}) \end{aligned}$$

- Khi truyền dãy bit d_m (-1 hoặc 1), trạm # m sử dụng mã của mình để điều chế mỗi bit d_{mi} thành n chips và truyền đồng thời trên n tần số:

$$d_m = (d_{m1}, d_{m2}, \dots) \quad \square \quad \begin{bmatrix} d_{m1} \cdot c_{m1} \\ d_{m1} \cdot c_{m2} \\ \dots \\ d_{m1} \cdot c_{mn} \\ \text{chips } d_{m1} \end{bmatrix} \quad \begin{bmatrix} d_{m2} \cdot c_{m1} \\ d_{m2} \cdot c_{m2} \\ \dots \\ d_{m2} \cdot c_{mn} \\ \text{chips } d_{m2} \end{bmatrix} \quad \dots \quad \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{bmatrix} \quad d_{mi} = -1, 1$$

- Trạm nhận (thường là BTS) nhận được tín hiệu của nhiều trạm phát đồng thời trên n tần số. Ví dụ trên tần số k , tại thời điểm *chip#1*:

$$R_k = d_{11} \cdot c_{1k} + d_{21} \cdot c_{2k} + d_{31} \cdot c_{3k} + \dots$$

- Do tính chất đôi một trực giao, giá trị $R_k \times C_m$ tổng hợp trên toàn bộ n tần số ($k=1,2..n$) cho phép xác định bit d_{mi} do trạm # m đã phát:

$$\begin{aligned} \sum (R_k \times C_m) &= \sum (d_{11} \cdot c_{1k} \cdot c_{mk} + d_{21} \cdot c_{2k} \cdot c_{mk} + d_{31} \cdot c_{3k} \cdot c_{mk} + \dots) \\ &= d_{11} \underbrace{\sum (c_{1k} \cdot c_{mk})}_{= 0 (m \neq 1)} + d_{21} \underbrace{\sum (c_{2k} \cdot c_{mk})}_{= 0 (m \neq 2)} + d_{31} \underbrace{\sum (c_{3k} \cdot c_{mk})}_{= n (m = 2)} + \dots = d_{m1} \times n \end{aligned}$$

CDMA: ví dụ

- Code trực giao, truyền trên 8 tần số:

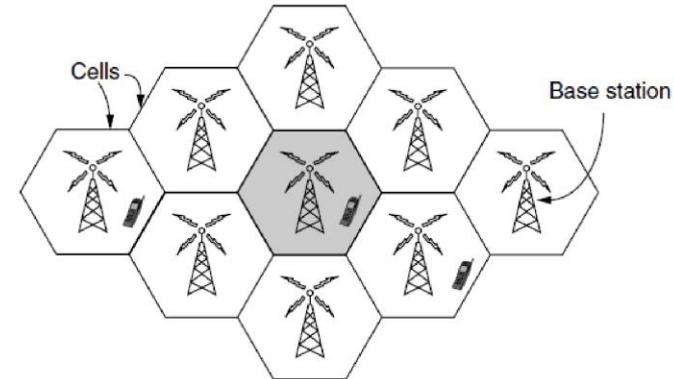
- C1 = (-1 -1 -1 +1 +1 -1 +1 +1)
- C2 = (-1 -1 +1 -1 +1 +1 +1 -1)
- C3 = (-1 +1 -1 +1 +1 +1 -1 -1)
- C4 = (-1 +1 -1 -1 -1 -1 +1 -1)

- Dữ liệu truyền:

- C1: {1,1,...} □ {(-1 -1 -1 +1 +1 -1 +1 +1), (-1 -1 -1 +1 +1 -1 +1 +1), ...}
- C2: {-1,1,...} □ {(+1 +1 -1 +1 -1 -1 -1 +1), (-1 -1 +1 -1 +1 +1 +1 -1), ...}
- C3: {1,-1,...} □ {(-1 +1 -1 +1 +1 +1 -1 -1), (+1 -1 +1 -1 -1 -1 +1 +1), ...}
- #4 không truyền

- Dữ liệu nhận tại BTS:

- R (không giải mã) = {(-1 +1 -3 +3 +1 -1 -1 +1), (-1 -3 +1 -1 +1 -1 +3 +1), ...}
- R (giải mã) x C₁ = {(+1 -1 +3 +3 +1 +1 -1 +1), (+1 +3 -1 -1 +1 +1 +3 +1), ...}
- R (giải mã) x C₂ = {(+1 -1 -3 -3 +1 -1 -1 -1), (+1 +3 +1 +1 +1 -1 +3 -1), ...}
- R (giải mã) x C₃ = {(+1 +1 +3 +3 +1 -1 +1 -1), (+1 -3 -1 -1 -1 +1 -3 -1), ...}
- R (giải mã) x C₄ = {(+1 +1 +3 -3 -1 +1 -1 -1), (+1 -3 -1 +1 -1 +1 +3 -1), ...}

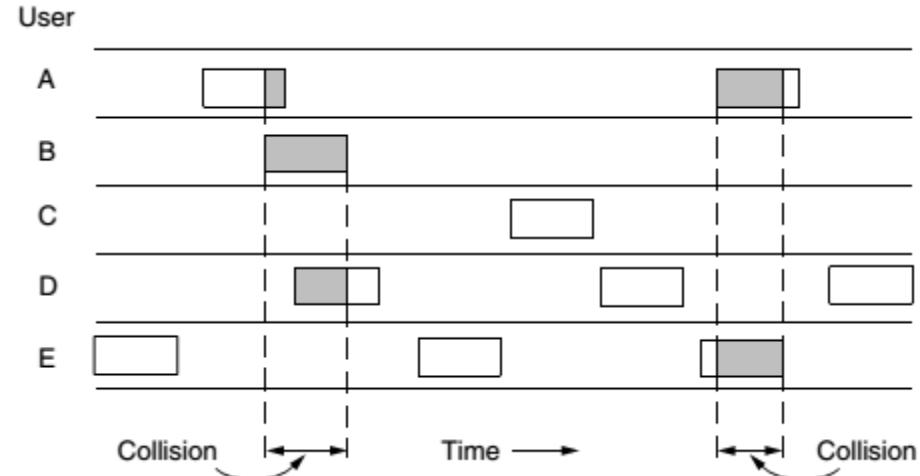


Điều khiển truy nhập ngẫu nhiên

- Aloha (Pure Aloha)

- Frame-time: thời gian để truyền hết một frame có kích thước lớn nhất ($= \text{MTU}/R$)
- Khi một nút mạng cần truyền dữ liệu:
 - Frame đầu tiên: truyền ngay. Nếu có đụng độ thì truyền lại với xác suất p
 - Các frame sau: truyền với xác suất là p
 - Trong 1 frame-time chỉ được truyền 1 frame

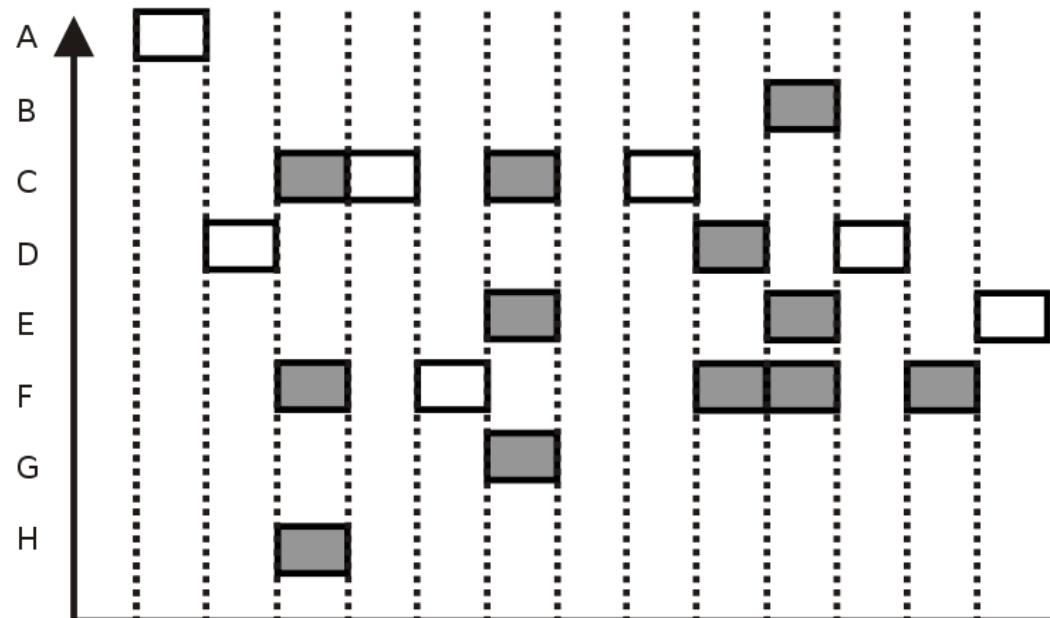
- Xác suất truyền thành công là $\sim 18.4\%$



Slotted Aloha

- Hoạt động như Aloha với các yêu cầu:

- Frame-time là nhau nhau với mọi nút
- Tất cả các nút phải đồng bộ về thời gian
- Các trạm đồng bộ time slot & Frame chỉ được truyền vào đầu slot
→ giảm nguy cơ collision
- Xác suất truyền thành công: 36.8%



Slotted ALOHA protocol (shaded slots indicate collision)

CSMA

- Đa truy nhập sử dụng sóng mang
- CSMA:
 - Carrier Sense Multiple Access
 - Cảm nhận sóng mang để quyết định đường truyền có bận hay không?
 - Nghe trước khi nói
 - Đụng độ xảy ra do trễ trên đường truyền
- CSMA/CA: CSMA with Collision Avoidance
- CSMA/CD: CSMA with Collision Detection
 - Phát hiện đụng độ : nghe trong khi nói
 - Giải quyết đụng độ với backoff (thuật toán quay lui)



Nhắc lại khái niệm sóng mang (Carrier)

- Tham khảo: DATA AND COMPUTER COMMUNICATIONS – Chương 5
- Sóng mang có tần số lớn (năng lượng lớn) để truyền được xa (cả môi trường có dây & không dây)
- Sóng mang là giao động tuần hoàn (không mang thông tin)

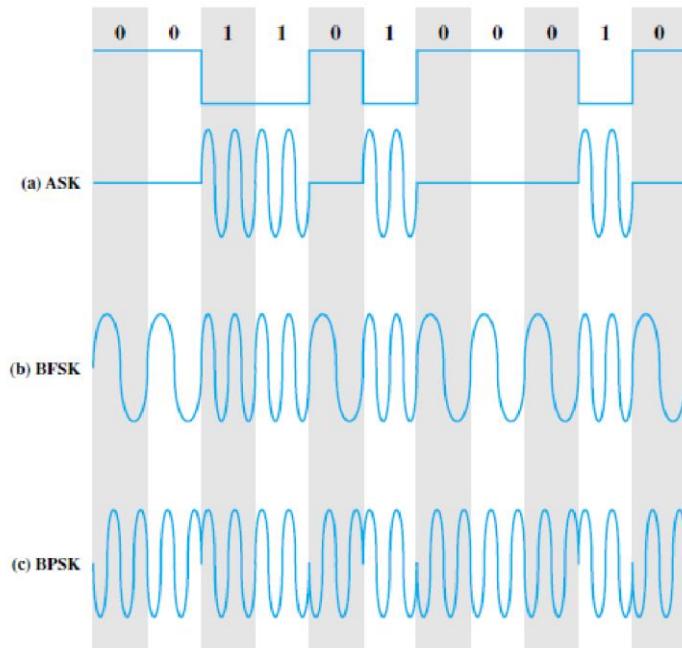


Figure 5.7 Modulation of Analog Signals for Digital Data

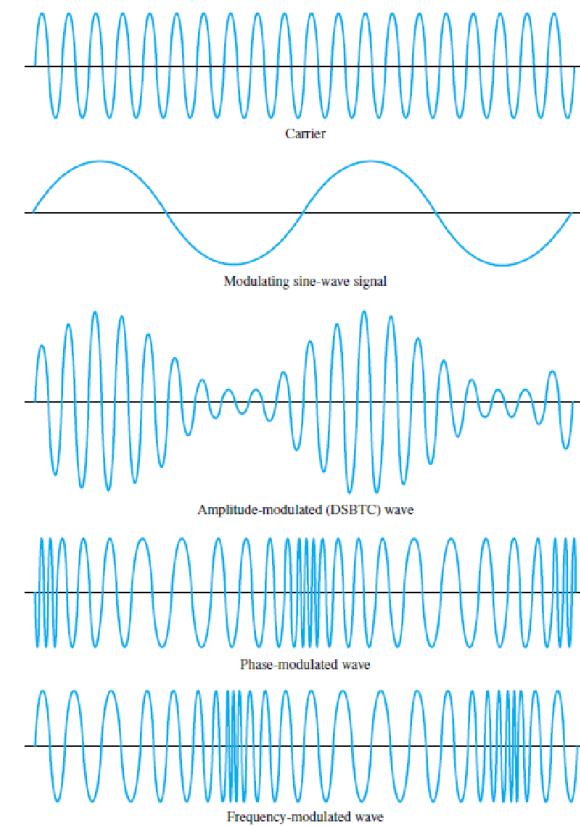
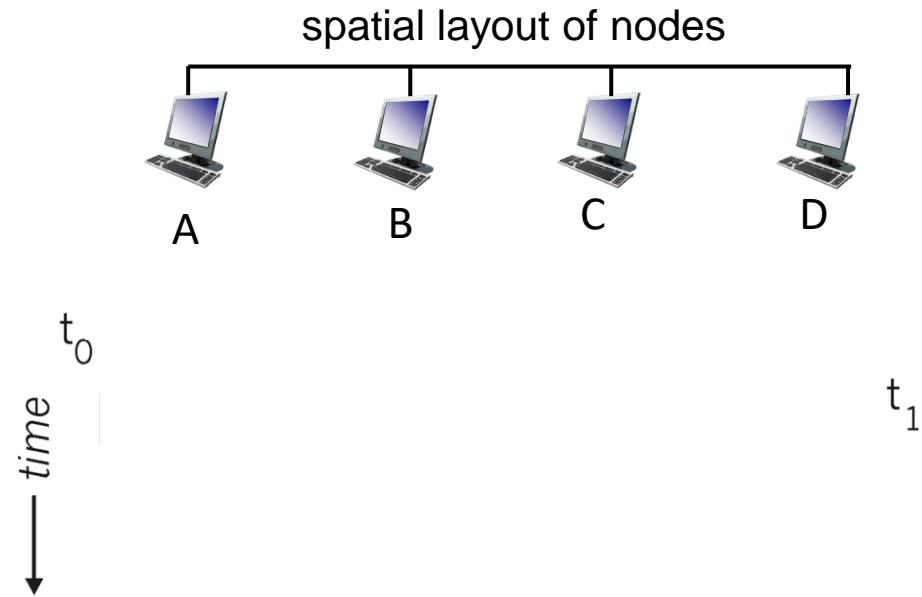


Figure 5.24 Amplitude, Phase, and Frequency Modulation of a Sine-Wave Carrier by a Sine-Wave Signal

- Tín hiệu/dữ liệu (mang thông tin) cần được truyền đi xa. Thường có tần số rất thấp (năng lượng yếu) ↗ dùng sóng mang “cõng” tín hiệu/dữ liệu đầu vào
- Phương pháp AM, FM (tín hiệu đầu vào analog)
- Các phương pháp Coding (dữ liệu đầu vào digital)

Độ truy cập trong CSMA

- Giả sử kênh truyền có 4 nút
- Tín hiệu điện tử lan truyền từ nút này đến nút kia mất một thời gian nhất định (trễ lan truyền)
- Ví dụ:



CSMA/CA (Collision Avoidance)

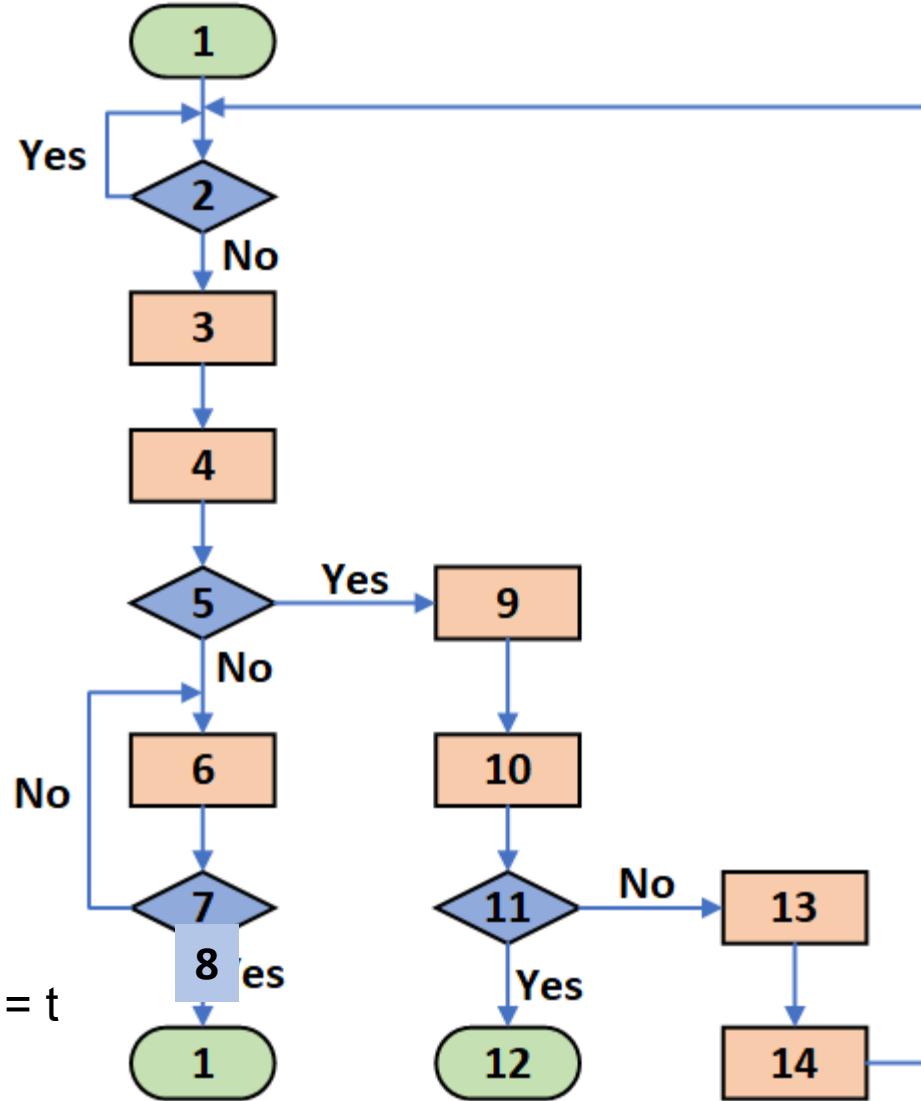
- Dùng trong mạng WIFI 802.11
- Nếu 2 hay nhiều trạm cùng phát hiện đường truyền bận và cùng chờ thì có khả năng chúng sẽ cùng truyền lại một lúc.
 - → xung đột,
- Giải pháp CSMA/CA.
 - Mỗi trạm chờ một khoảng thời gian được tính ngẫu nhiên → giảm xác xuất đụng độ.

CSMA/CD (Collision detection)

- Dùng trong mạng Ethernet
- CSMA có phát hiện xung đột: Listen while talk.
- Nếu đường truyền rõ, truyền
- Nếu bận, chờ rồi truyền với xác suất p
- Nghe đường truyền trong khi truyền, nếu có xung đột chỉ phát 1 tín hiệu báo xung đột ngắn và dừng ngay
- Không tiếp tục truyền như CSMA
- Thủ truyền lại sau một khoảng thời gian ngẫu nhiên.

Thuật toán CSMA

- 1: Yêu cầu truyền dữ liệu
- 2: Đường truyền bận ?
- 3: Tổ chức data thành Frame
- 4: Truyền Frame
- 5: Có dung độ ?
- 6: Tiếp tục truyền
- 7: Hết dữ liệu cần truyền ?
- 8: Kết thúc
- 9: Truyền tín hiệu JAM
- 10: Inc(attempts)
- 11: attempts > Max Attempts
- 12: Error !!!
- 13: Tính toán khoảng thời gian backoff = t
- 14: Delay(t)



CSMA/CD: Tóm tắt

- Máy trạm nghe trước khi muốn truyền
 - BẬN: Chờ, tiếp tục nghe
 - RỖI: Bắt đầu truyền, vừa truyền vừa “nghe ngóng” xem có xung đột hay không
 - Nghe trong thời gian bao lâu?
 - Nếu phát hiện thấy xung đột: Hủy bỏ quá trình truyền và quay lại trạng thái chờ, nghe
- Một số biến thể của CSMA
 - CSMA kiên nhẫn (persistance)
 - CSMA không kiên nhẫn
 - CSMA với xác suất p nào đó

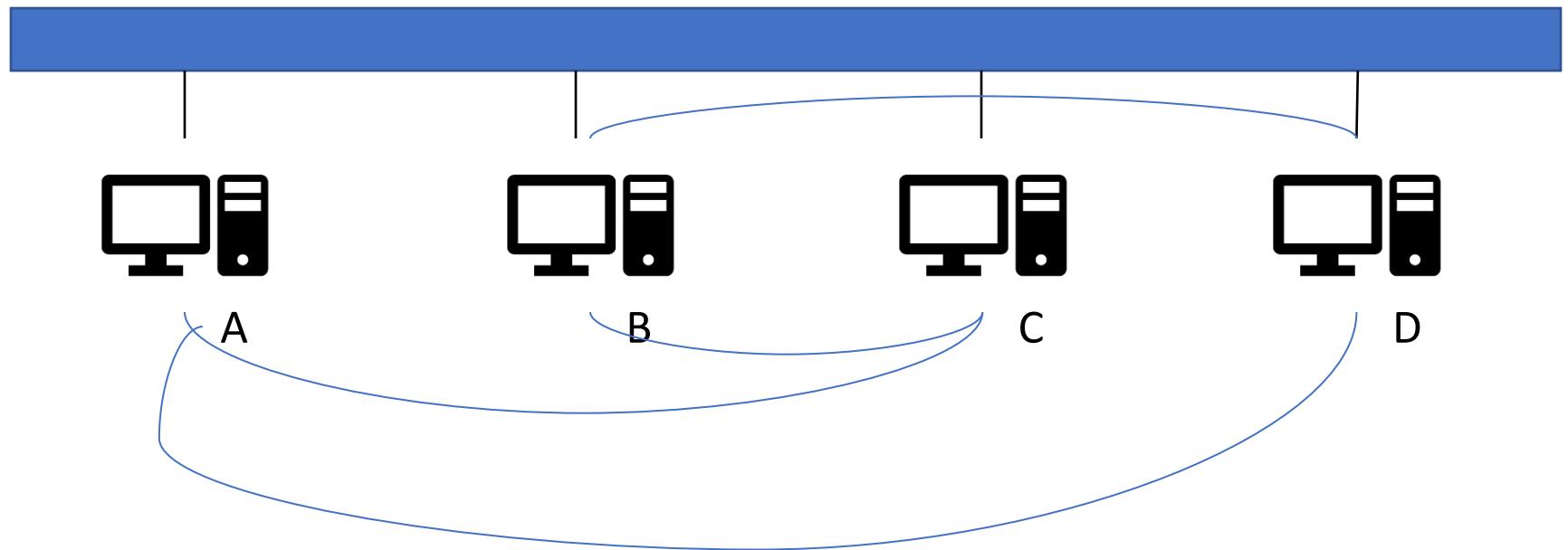
So sánh chia kênh & truy nhập ngẫu nhiên

- Chia kênh
 - Hiệu quả, công bằng cho đường truyền với lưu lượng lớn
 - Lãng phí nếu chúng ta cấp kênh con cho một nút chỉ cần lưu lượng nhỏ
- Truy nhập ngẫu nhiên
 - Khi tải nhỏ: Hiệu quả vì mỗi nút có thể sử dụng toàn bộ kênh truyền
 - Tải lớn: Xung đột tăng lên
- Phương pháp quay vòng: Có thể dung hòa ưu điểm của hai phương pháp trên

Token passing

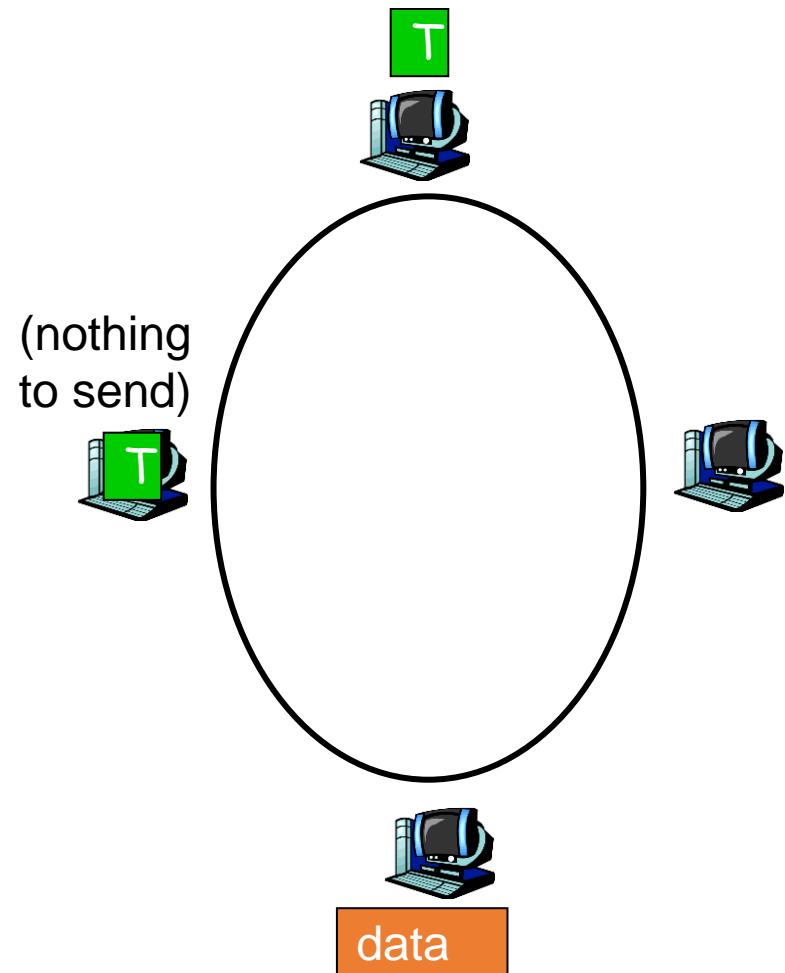
- Bit trạng thái: rõ hay bận
- Nút mạng nhận được thẻ bài rõ, không mang dữ liệu : được phép truyền dữ liệu
 - Thiết lập trạng thái thẻ bài về trạng thái bận
 - Tổ chức dữ liệu để truyền, thẻ bài trở thành tiêu đề của frame
 - Sau khi truyền xong dữ liệu: thiết lập trạng thái thẻ bài là rõ
- Nút đích: sao chép dữ liệu trên frame và trả lại frame cho nút nguồn
- Token Ring: vòng luân chuyển thẻ bài là vòng vật lý
- Token Bus: vòng luân chuyển thẻ bài là vòng logic
- Hạn chế

Token bus



Token Ring – Mạng vòng dùng thẻ bài

- Một “thẻ bài” luân chuyển lần lượt qua từng nút mạng
- Nút nào giữ thẻ bài sẽ được gửi dữ liệu
- Gửi xong phải chuyển thẻ bài đi
- Một số vấn đề
 - Tốn thời gian chuyển thẻ
 - Trễ
 - Mất thẻ bài....



Tổng kết các phương pháp kiểm soát đa truy nhập

- Chia kênh
- Truy nhập ngẫu nhiên
- Quay vòng
- Phân tích ưu, nhược điểm

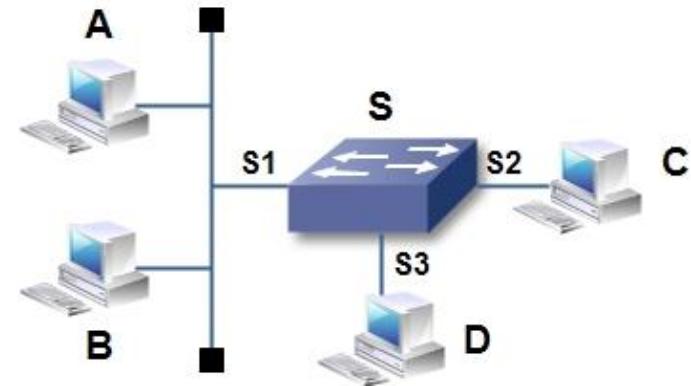


ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

5. Chuyển tiếp dữ liệu

Chuyển tiếp dữ liệu tầng 2

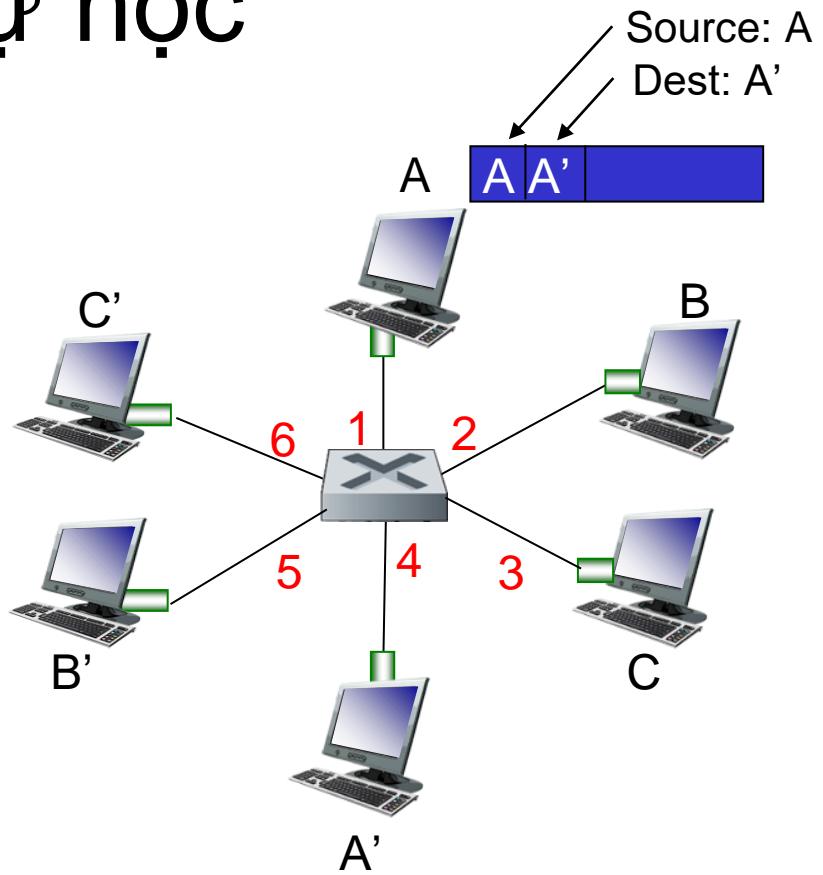
- Bảng địa chỉ MAC
 - Địa chỉ MAC của host
 - Cổng kết nối với host
 - TTL: thời gian giữ lại thông tin trong bảng
- Cơ chế tự học
- Chuyển mạch
- Quảng bá: địa chỉ MAC là FF:FF:FF:FF:FF:FF



MAC Addr.	Interface	TTL
A	S1	
B	S1	
C	S2	

Switch: Cơ chế tự học

- Cập nhật địa chỉ MAC nguồn và cổng nhận gói tin vào bảng MAC Table nếu:
 - Địa chỉ nguồn chưa có trong bảng MAC Table, hoặc
 - Địa chỉ nguồn đã có nhưng nhận được gói tin trên cổng khác



MAC addr	interface	TTL
A	1	60

MAC Table
(ban đầu rỗng)

Switch: Cơ chế chuyển tiếp

Khi nhận được 1 frame

1. Tìm đ/c cổng vào (tự học)
2. Tìm địa chỉ cổng ra dùng bảng chuyển tiếp
3. **if** tìm thấy cổng ra
then {
 if cổng ra == cổng vào
 then hủy bỏ frame
 else chuyển tiếp frame đến cổng ra
 }
 else quảng bá frame đến tất cả các cổng

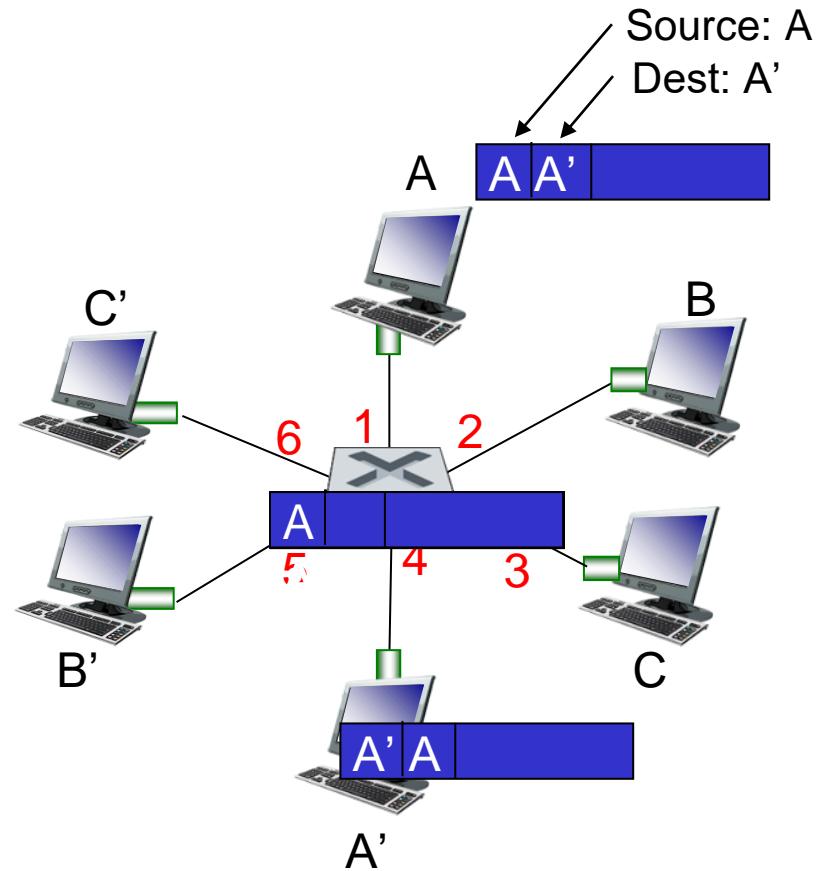
Ví dụ

Không có cổng ra:

→ *Quảng bá*

Đã biết địa chỉ A:

→ *Chuyển trực tiếp*

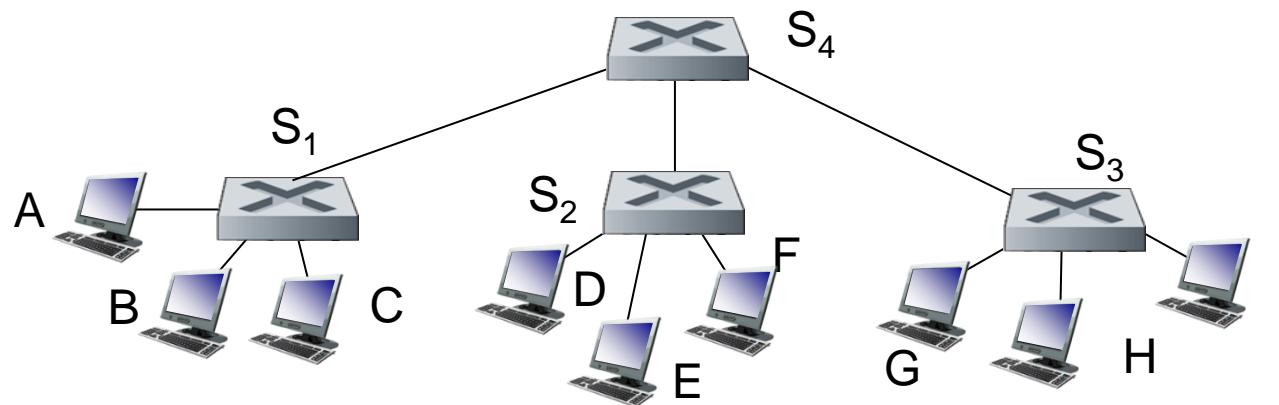


*MAC Table
(ban đầu rỗng)*

MAC addr	interface	TTL
A	1	60
A'	4	60

Nối chồng các switch với nhau

- Các switch có thể được nối với nhau



- Cũng dùng cơ chế tự học:
 - A → I, S1 học A, quảng bá: B, C, S4
 - S4: học A từ S1, quảng bá: S2, S3
 - S3: học A từ S4, quảng bá: G, H, I



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

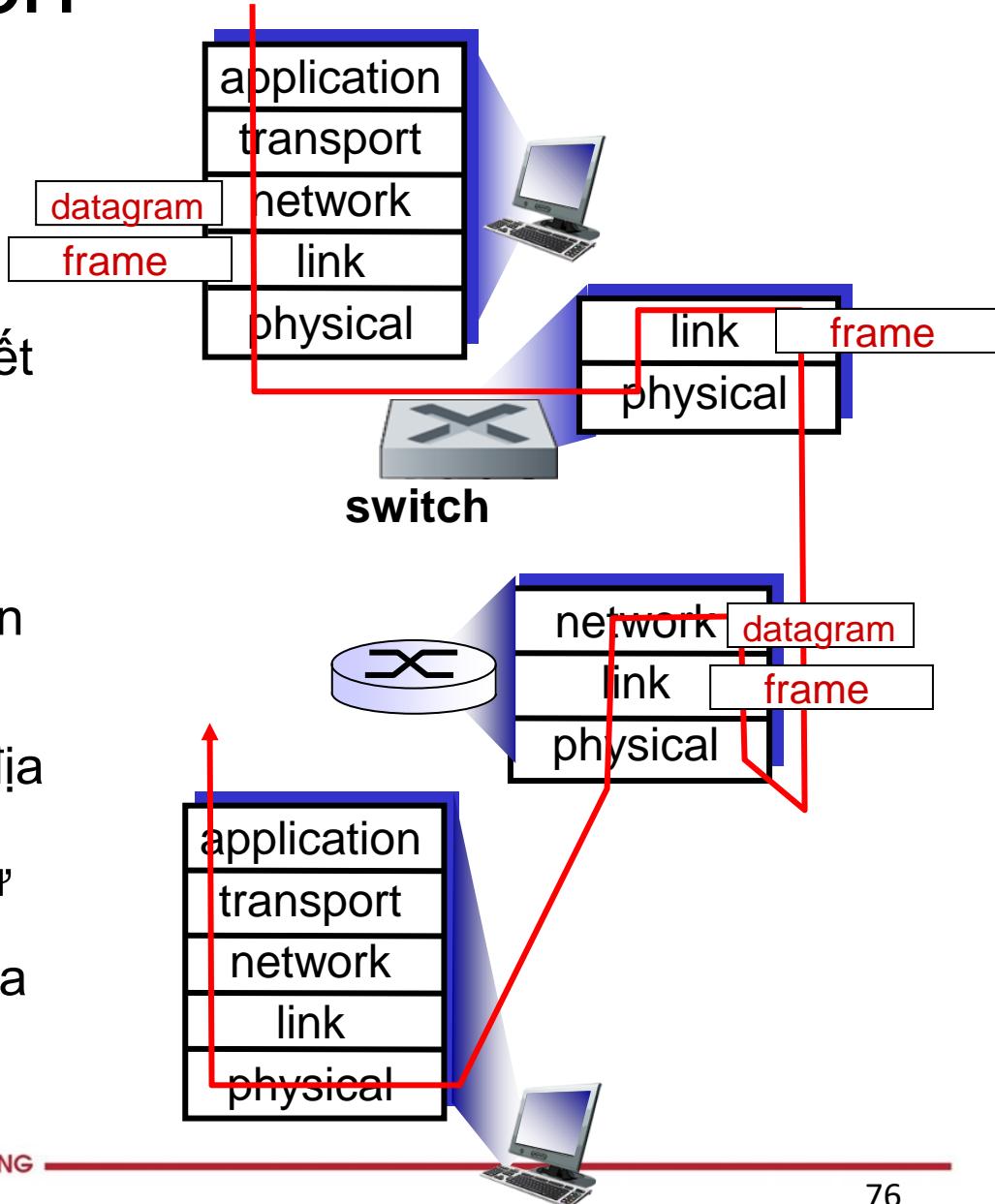
6. Mạng cục bộ (LAN)

Các thiết bị kết nối trong mạng LAN

- Repeater (bộ lặp), Hub (bộ chia)
 - Đảm nhiệm chức năng tầng 1
 - Tăng cường tín hiệu → mở rộng phạm vi kết nối (broadcast zone)
 - ≤ 4 repeater / 1 đoạn mạng (đường truyền kết nối 2 nút mạng)
- Bridge (Cầu), Switch (Bộ chuyển mạch)
 - Đảm nhiệm chức năng tầng 1 và 2
 - Cho phép kết nối các loại đường truyền vật lý khác nhau
 - Chia nhỏ miền đụng độ
 - Chuyển mạch cho khung tin dựa trên địa chỉ MAC
- Router (Bộ định tuyến)

Router vs Switch

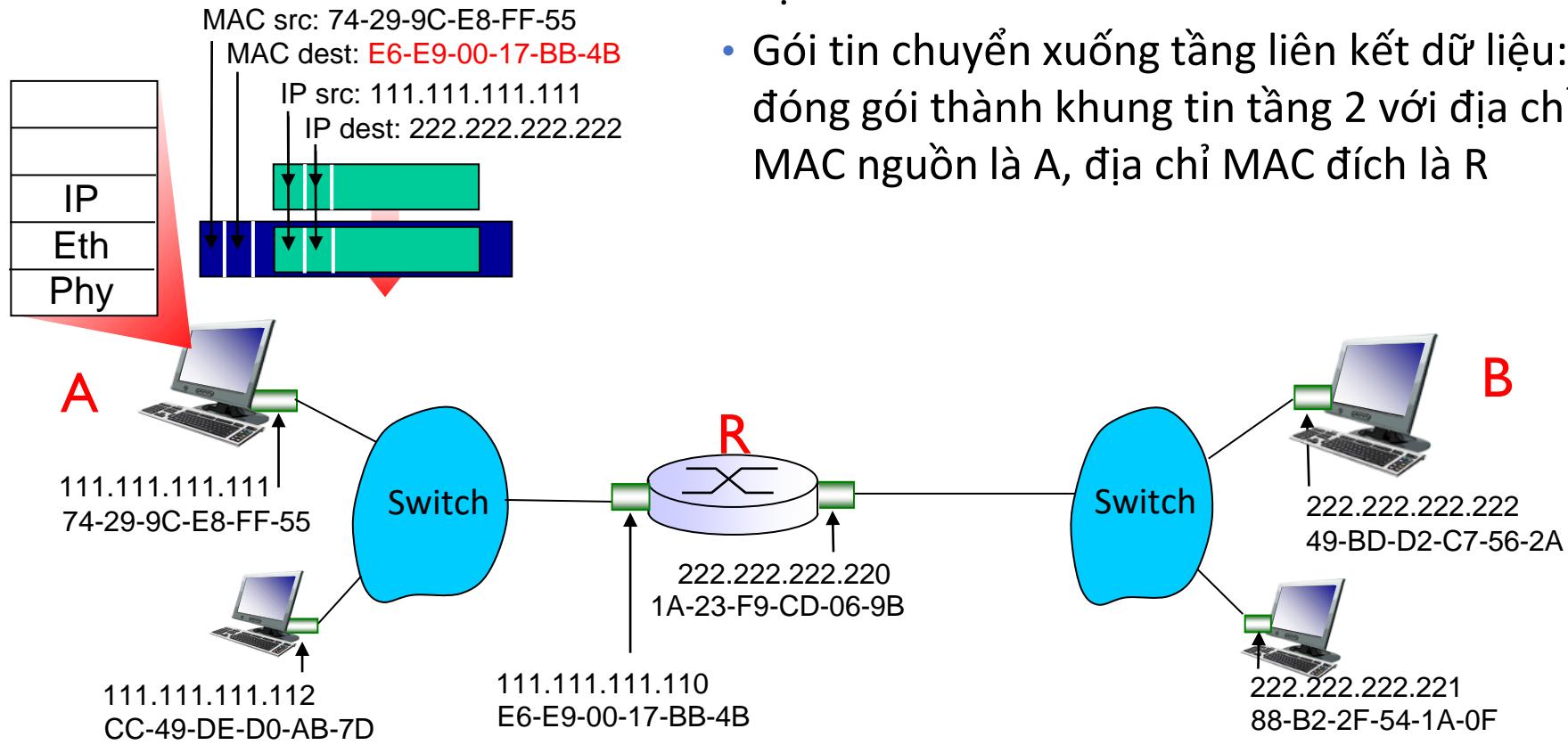
- Xử lý gói tin: lưu và chuyển tiếp (store-and-forward)
 - Router: thiết bị tầng mạng
 - Switch: thiết bị tầng liên kết dữ liệu
- Chuyển tiếp gói tin:
 - Router: sử dụng thuật toán định tuyến tính toán bảng chuyển tiếp (Forwarding Table), chuyển tiếp theo địa chỉ IP đích
 - Switch: sử dụng cơ chế tự học tính toán bảng MAC Table, chuyển tiếp theo địa chỉ MAC đích



Router kết nối LAN

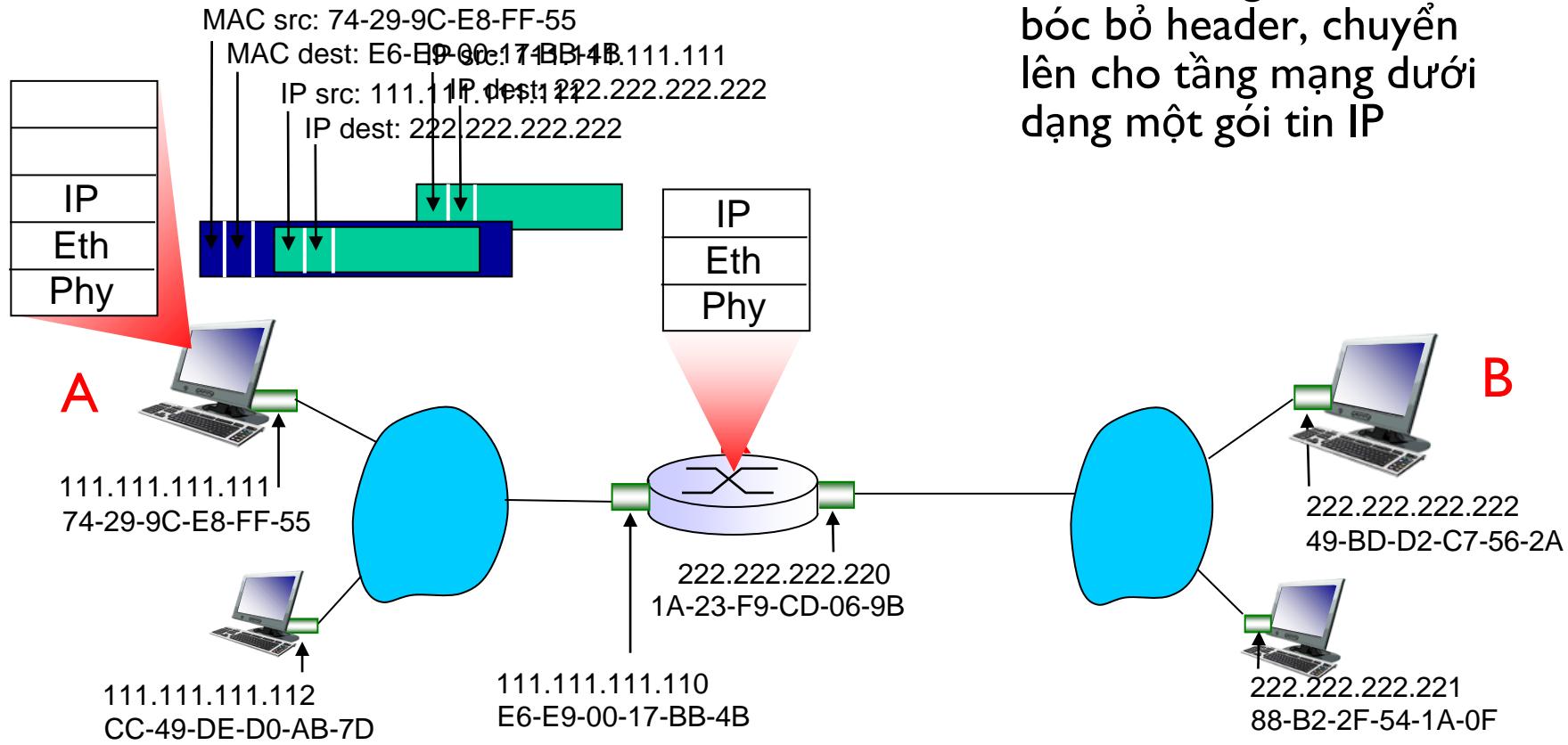
Ví dụ: Gửi dữ liệu từ A tới B qua router R

- A soạn một *gói tin IP* với địa chỉ nguồn là A, địa chỉ đích là B
- Gói tin chuyển xuống tầng liên kết dữ liệu: đóng gói thành khung tin tầng 2 với địa chỉ MAC nguồn là A, địa chỉ MAC đích là R



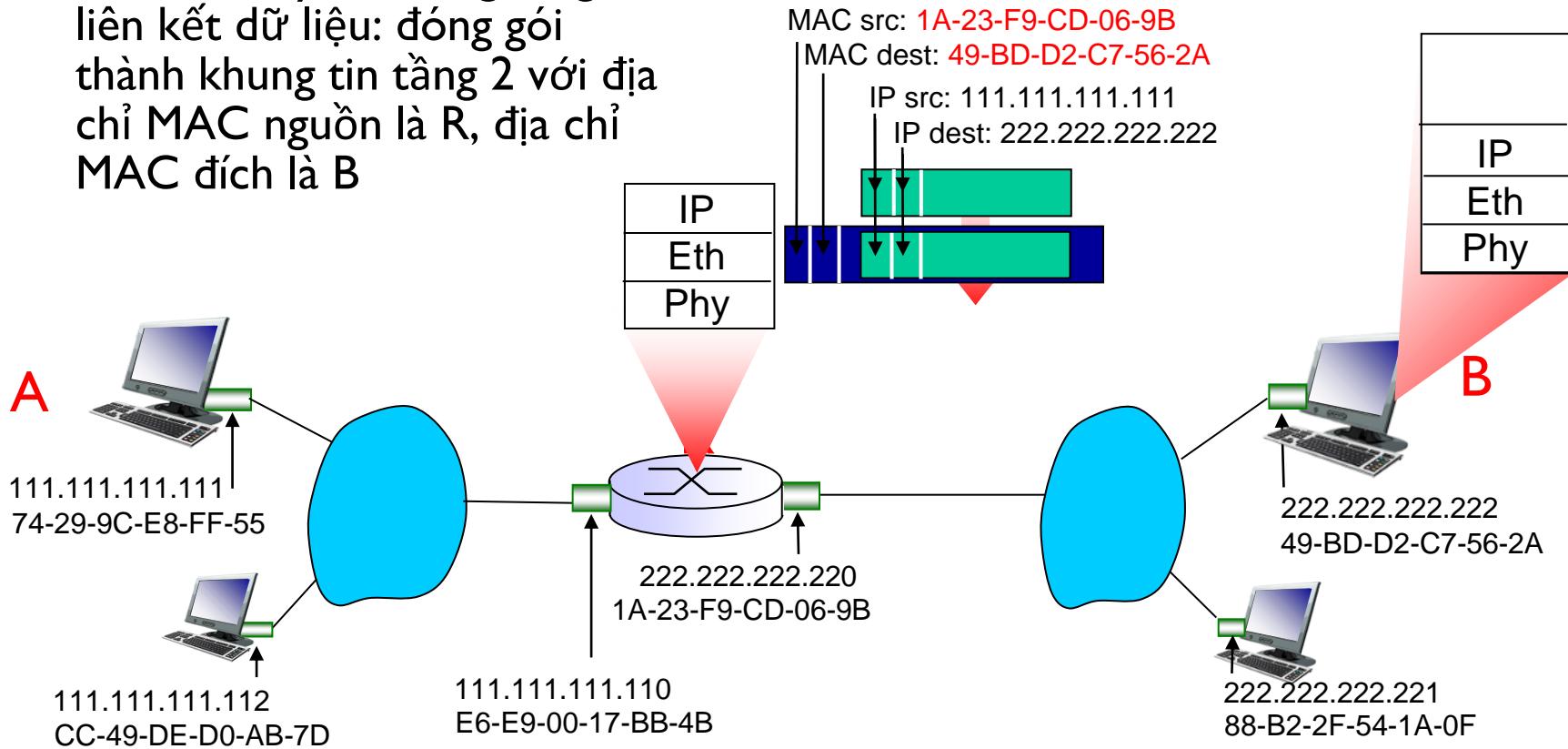
Chuyển tiếp dữ liệu tới LAN khác

- ❖ Khung tin được chuyển từ A tới R
- ❖ Tại R: khung tin được bóc bỏ header, chuyển lên cho tầng mạng dưới dạng một gói tin IP



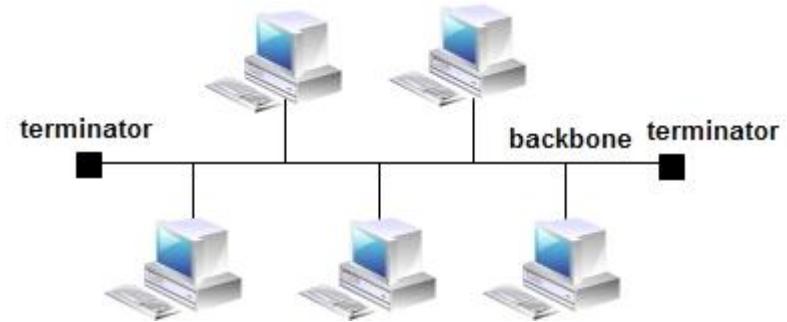
Chuyển tiếp gói tin trong LAN đích

- ❖ R chuyển tiếp gói tin với địa chỉ IP nguồn là A, IP đích là B
- ❖ Gói tin chuyển xuống tầng liên kết dữ liệu: đóng gói thành khung tin tầng 2 với địa chỉ MAC nguồn là R, địa chỉ MAC đích là B



Hình trạng LAN: bus (mạng trực)

- Tất cả các nút mạng sử dụng chung đường truyền – trực (backbone)
- Mỗi nút mạng kết nối vào trực bằng đầu nối chữ T (T-connector)
- Terminator hai đầu
- Phương thức truyền: điểm – đa điểm (point-to-multipoint)
 - Dữ liệu truyền theo 2 hướng
 - Nút nhận: kiểm tra địa chỉ đích của dữ liệu
- Ưu điểm
- Nhược điểm



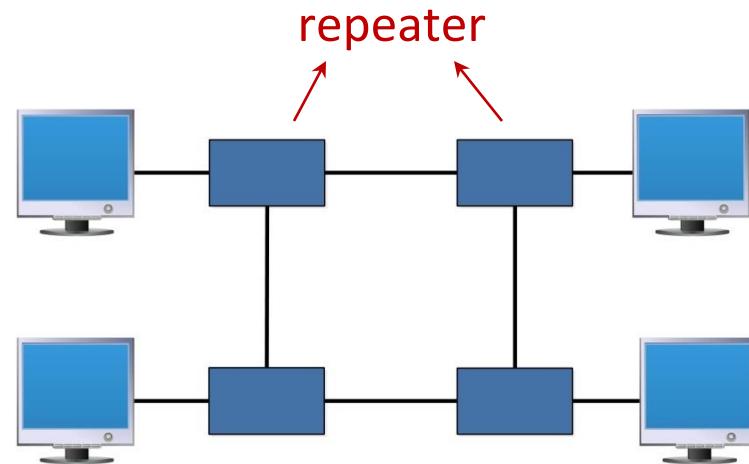
Hình trạng LAN: star (hình sao)

- Một nút mạng đóng vai trò thiết bị trung tâm
 - Hub
 - Switch
 - Router
- Các nút mạng khác kết nối trực tiếp với thiết bị trung tâm
- Phương thức truyền
 - Điểm – điểm: switch, router
 - Điểm – đa điểm: hub
- Ưu điểm
- Nhược điểm

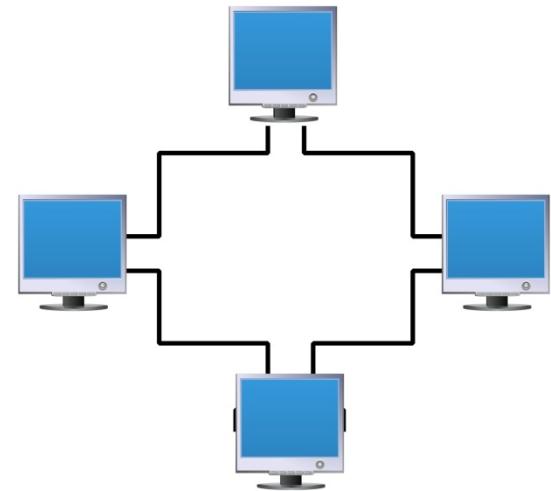


Hình trạng LAN: ring (hình vòng)

- Các nút mạng chung đường truyền khép kín
- Phương thức truyền: điểm – điểm (point-to-point) hoặc điểm-đa điểm
- Dự phòng: FDDI vòng kép, thường sử dụng cho khu vực mạng xương sống
- Ưu điểm
- Nhược điểm



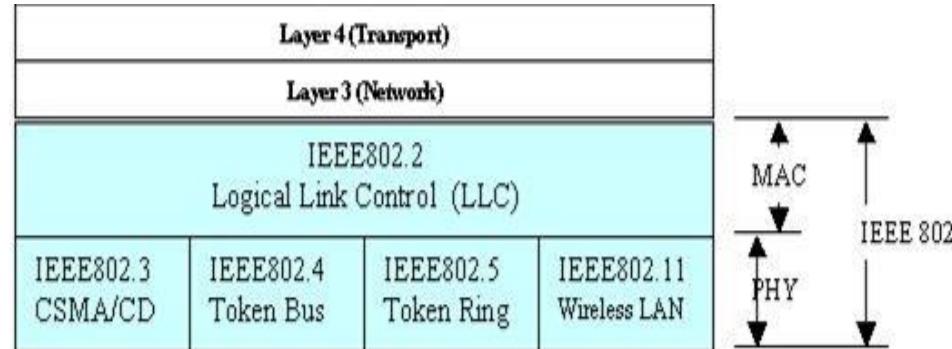
điểm – đa điểm



điểm – điểm

Chuẩn hóa LAN: IEEE 802.x

- IEEE 802.1 Network Management
- IEEE 802.2 Logical link control
- IEEE 802.3 Ethernet (CSMA/CD)
- IEEE 802.4 Token bus
- IEEE 802.5 Token Ring
- IEEE 802.6 Metropolitan Area Networks
- IEEE 802.7 Broadband LAN using Coaxial Cable
- IEEE 802.8 Fiber Optic TAG
- IEEE 802.9 Integrated Services LAN
- IEEE 802.10 Interoperable LAN Security
- IEEE 802.11 Wireless LAN



- IEEE 802.12 demand priority
- IEEE 802.14 Cable modems
- IEEE 802.15 Wireless PAN
- IEEE 802.15.1 (Bluetooth)
- IEEE 802.15.4 (ZigBee)
- IEEE 802.16 WiMAX
- V.v...

LLC: IEEE802.2

802.2 LLC Header			Information
DSAP address	SSAP address	Control	
8 bits	8 bits	8 or 16 bits	multiple of 8 bits

- Vai trò:
 - kết nối các giao thức tầng Network: IPX, DCE, **IP**, v.v..
 - với các loại tầng vật lý khác nhau: có dây, không dây, quang
- Chức năng:
 - phân kênh/dồn kênh (multiplexing)
 - kiểm soát luồng (flow control) theo 3 chế độ làm việc:
 - Unacknowledged connectionless
 - Acknowledged connectionless
 - Connection mode
- Cấu trúc gói tin:
 - DSAP & SSAP: Destination/Source SAP, phục vụ phân kênh/dồn kênh với tầng trên (cho biết thực thể nào trên tầng Network đang thực hiện truyền/nhận gói tin LLC)
 - Control: định nghĩa phân loại PDU để truyền tin & kiểm soát:
 - U-frame: gửi/nhận trong chế độ connectionless (U: Unnumbered)
 - I-frame: frame có thông báo (I: Information), sử dụng trong chế độ acknowledged
 - S-frame: dùng để kiểm soát (S: Supervisor)

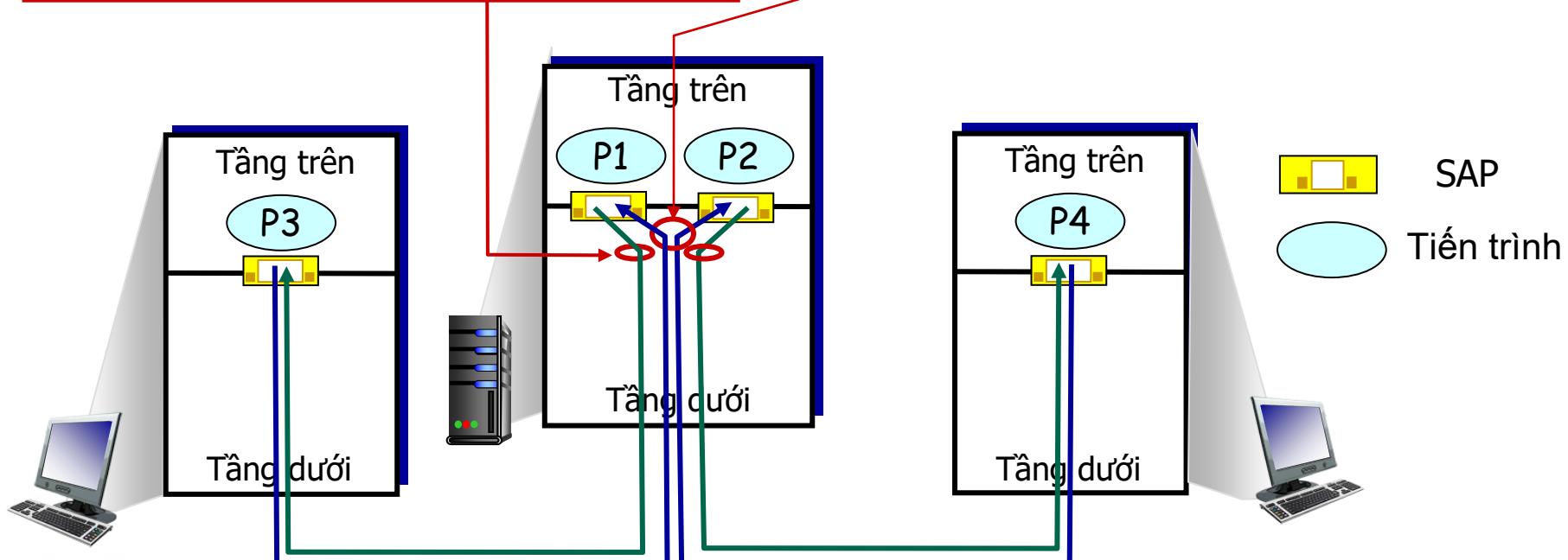
Nhắc lại SAP & cơ chế phân kênh/dồn kênh

Gửi: Dồn kênh

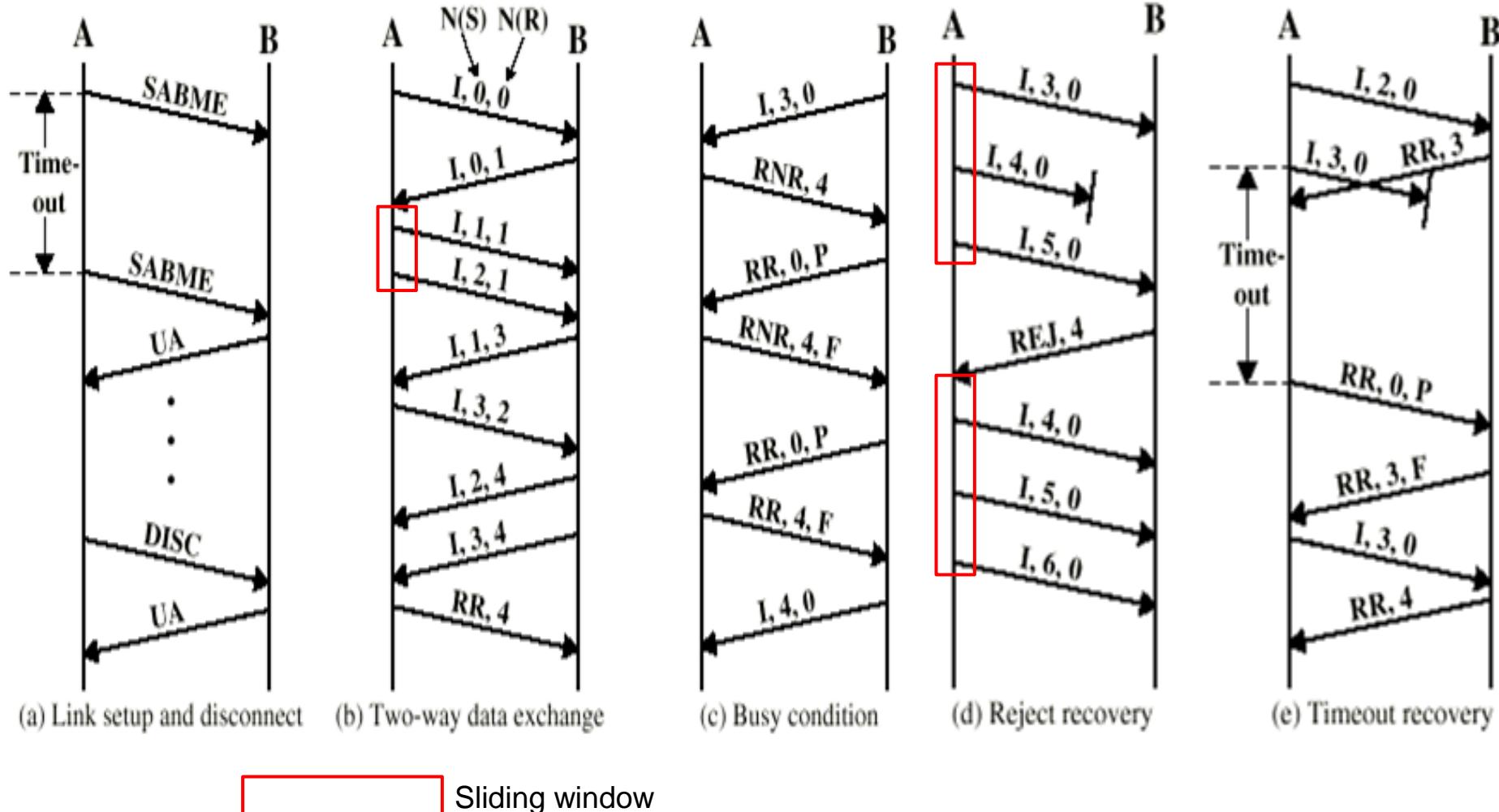
Nhận data từ tiến trình tầng trên, đóng gói PDU với SAP bên gửi (S.SAP) và SAP bên nhận (D.SAP) rồi chuyển PDU lên “đường truyền”

Nhận: Phân kênh

Nhận PDU từ “đường truyền”, đọc thông tin trường D.SAP để xác định tiến trình tầng trên & gửi data lên



Hoạt động: stop & wait, sliding window

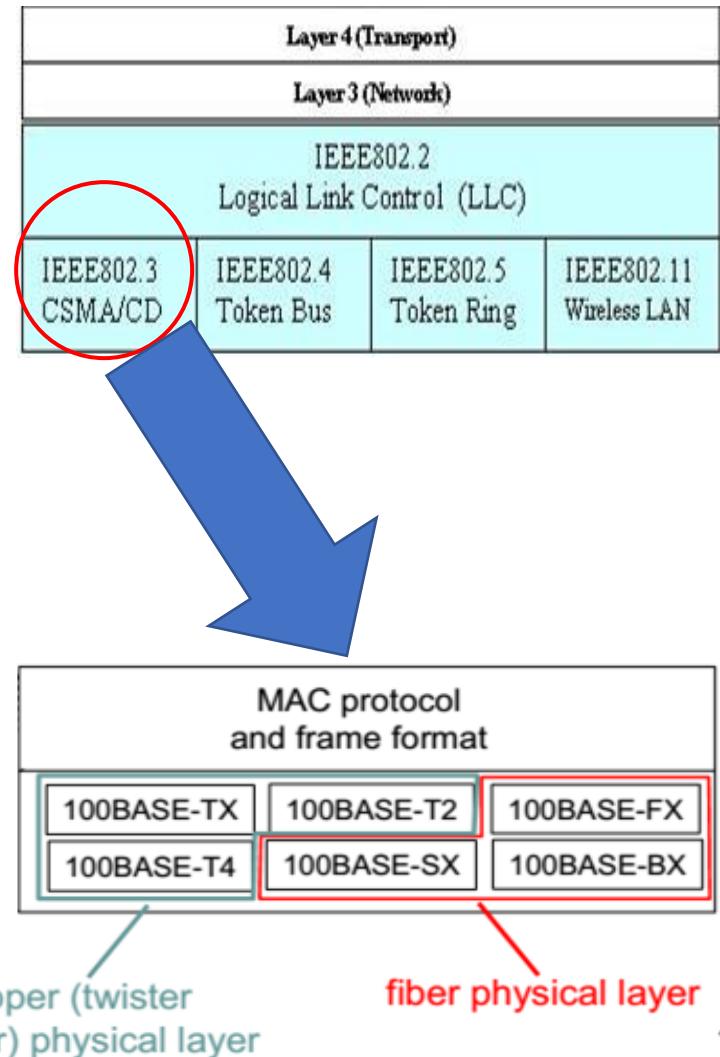


Thực tế sử dụng LLC

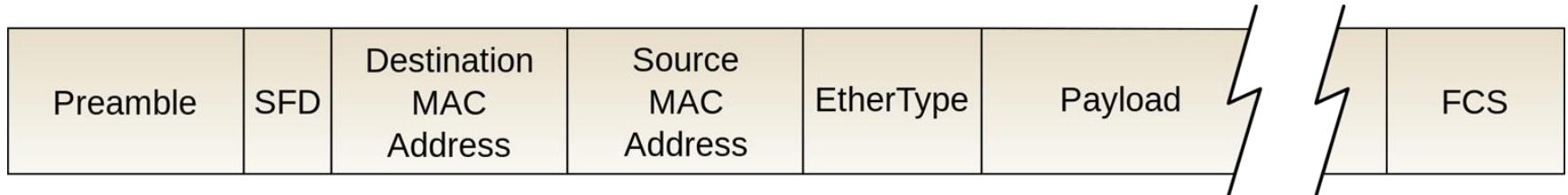
- Chức năng kiểm soát lỗi & luồng (với I-frame & S-frame) được một số giao thức tầng trên sử dụng (NetBIOS).
- U-frame đóng gói PDU kiểu không có chỉ số (unnumbered) và như vậy không hỗ trợ kiểm soát lỗi cũng như kiểm soát luồng.
- Hầu hết các chòng giao thức bên trên LLC (TCP/IP) đã hỗ trợ kiểm soát lỗi & luồng → sử dụng LLC chỉ với riêng chức năng phân kênh/dồn kênh ở chế độ “Unacknowledged connectionless” với U-frame.

Ethernet: IEEE802.3

- Chức năng:
 - Kiểm soát truy nhập đường truyền (Data-link)
 - Mã hóa tín hiệu đường truyền có dây (Physical)
- Cài đặt CSMA/CD
- Hỗ trợ đường truyền:
 - Cáp đồng trục
 - 10BASE-TX: cáp xoắn đôi tốc độ 10Mbps
 - 100BASE-TX (fast ethernet): cáp xoắn đôi tốc độ 100Mbps
 - Giga Ethernet FX: cáp quang tốc độ Gbps



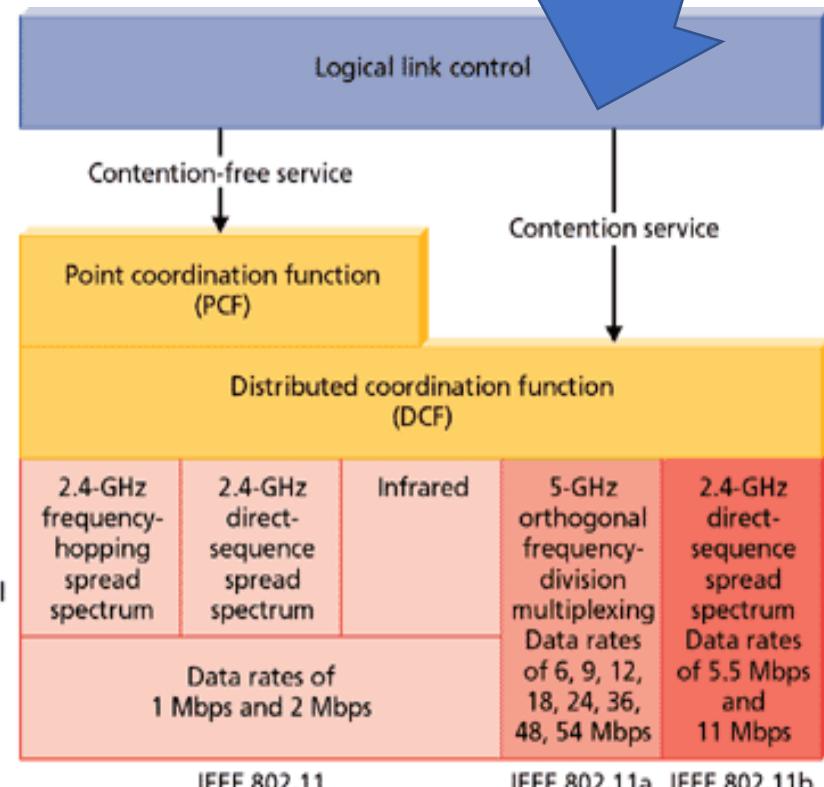
Cấu trúc gói tin Ethernet



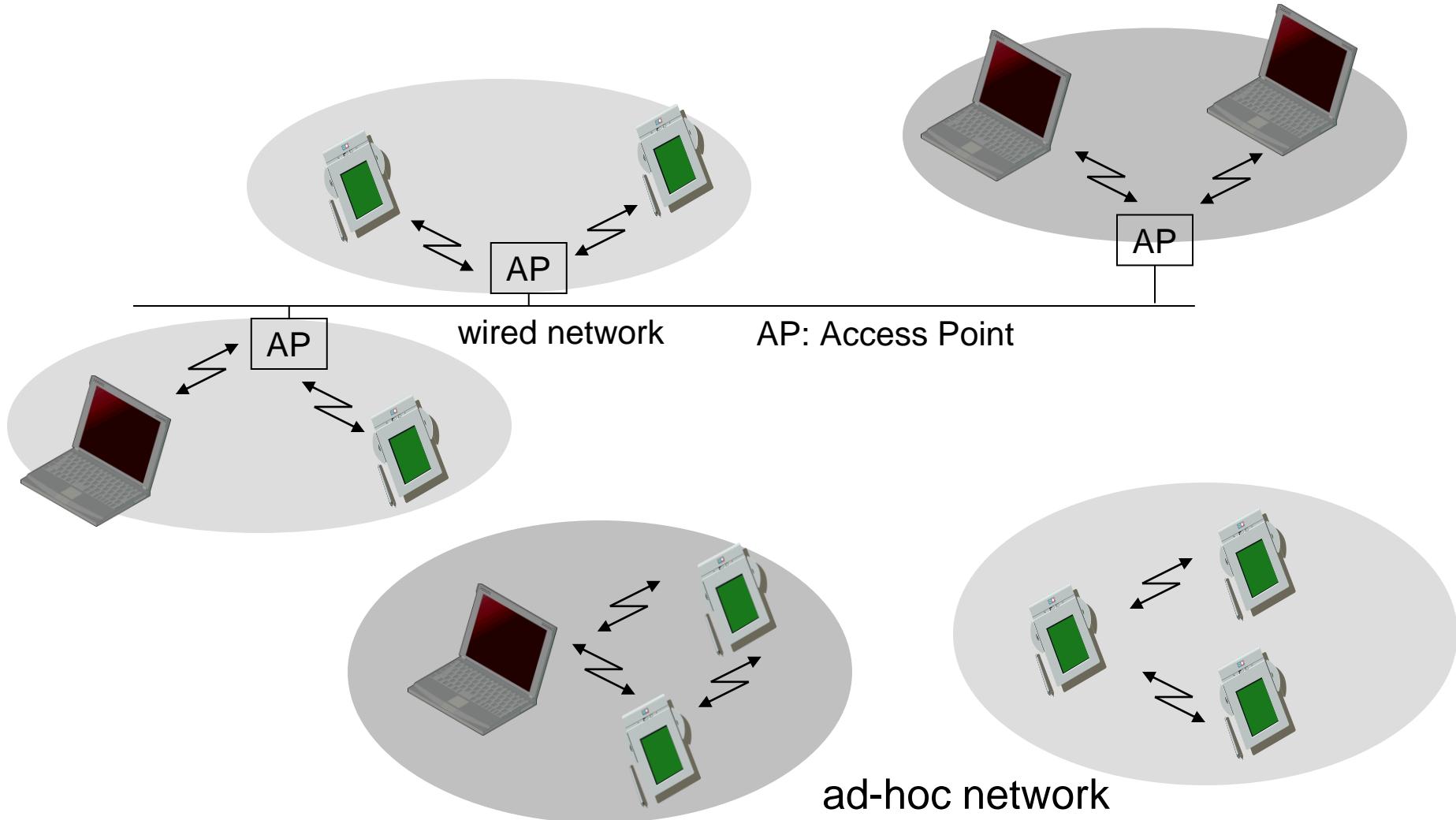
- Preamble (7 byte): 101010... đánh dấu bắt đầu một Ethernet frame
- SFD (1 byte): Start frame delimiter = 10101011. Ngăn cách Preamble với frame
- MAC Address (6 byte): địa chỉ vật lý các trạm nguồn & đích
- EtherType (2 byte): xác định giao thức tầng trên (IP, Novell IPX, AppleTalk, ...)
- FCS (4 byte): Frame check sequence. Mã kiểm soát lỗi theo cơ chế CRC. Phối hợp với LLC (i-frame & s-frame) để xử lý báo lỗi & truyền lại. Trường hợp sử dụng LLC với u-frame, FCS chỉ để kiểm tra lỗi nhận frame và hủy frame bên nhận (không thông báo, không phát lại)

Wireless LAN: IEEE802.11

- Kiểm soát truy nhập đường truyền (Data-link):
 - PCF: dựa vào một thiết bị trung tâm (Access Point – AP) để điều phối truy nhập đường truyền của tất cả các trạm
 - DCF: điều phối truy nhập phân tán
- Mã hóa tín hiệu đường truyền không dây (Physical):
 - Tia hồng ngoại: 902-928MHz
 - Dải tần 2.4GHz
 - Dải tần 5GHz
- Cài đặt CSMA/CA

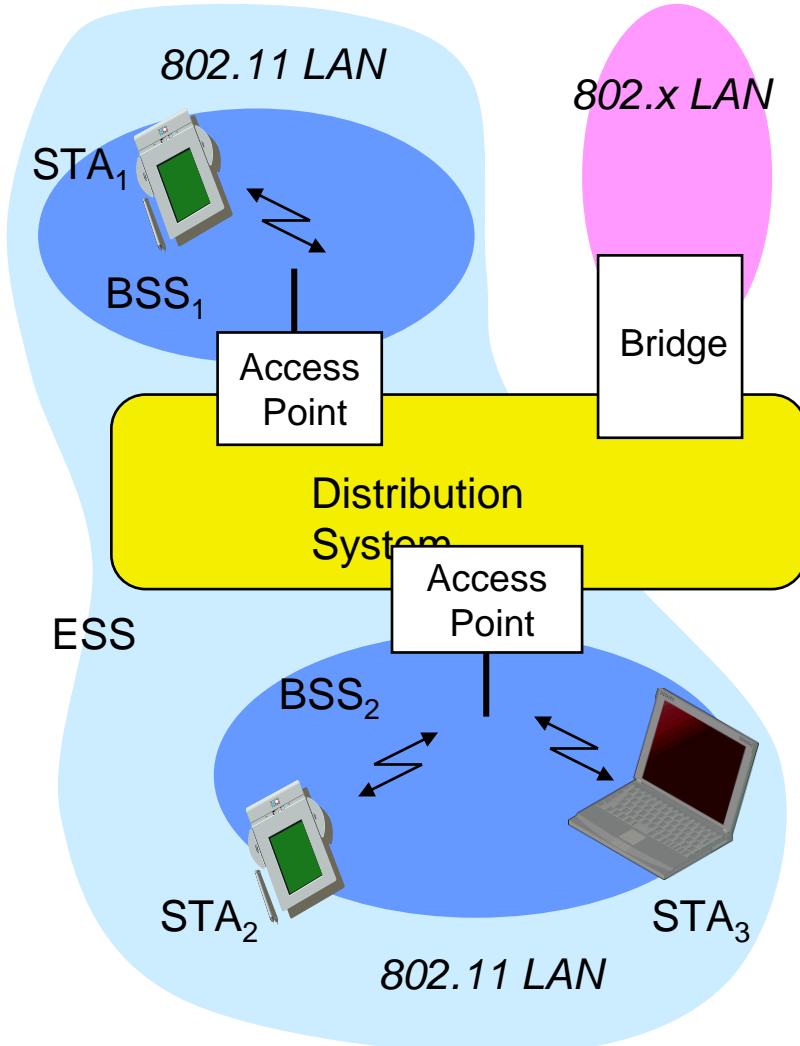


Mô hình triển khai WLAN



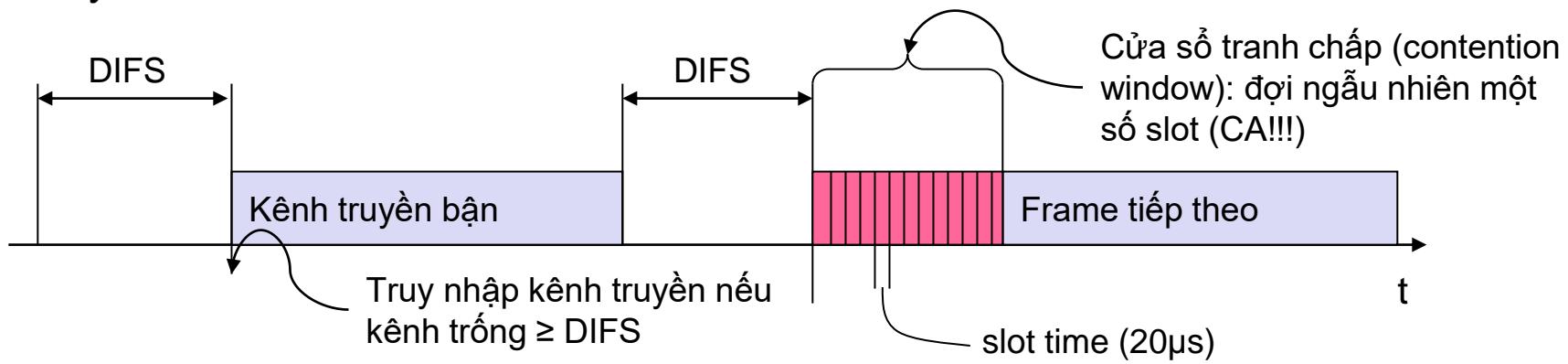
Kiến trúc thê thống

- Station (STA): Trạm làm việc, với kết nối không dây đến Access Point
- Access Point: Điểm điều phối các STA, đồng thời liên kết với hệ thống điều phối phân tán (Distributed System)
- Basic Service Set (BSS): nhóm các trạm làm việc chung trên một dải tần (với một AP hoặc nhiều AP)
- Bridge: cầu kết nối với hệ thống có dây (Ethernet, Bus, v.v..)
- Distribution System:
 - Liên kết nhiều BSS để tạo nên một ESS (Extended Service Set)
 - Không được chuẩn hóa trong IEEE802.11, hoạt động tùy theo nhà sản xuất

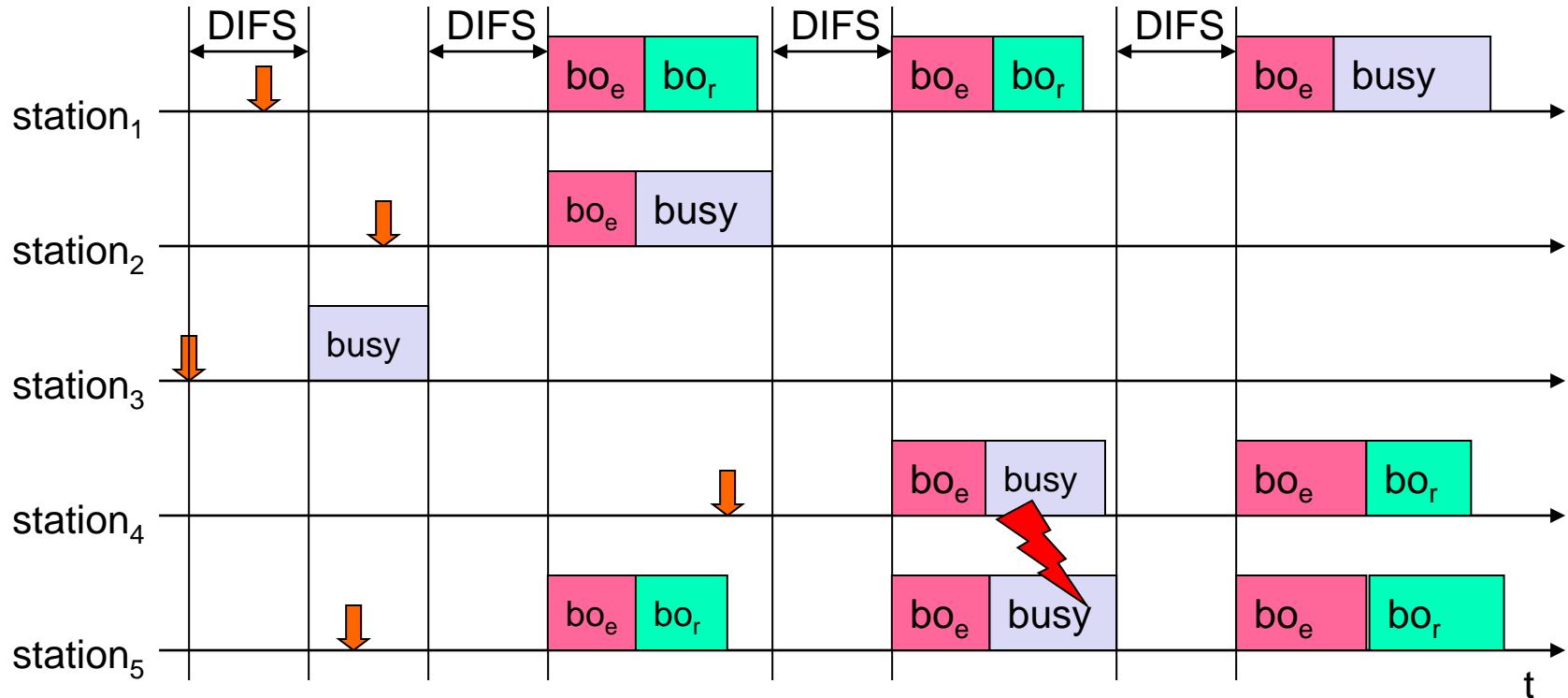


Cài đặt CSMA/CA trên 802.11

- STA muốn truyền dữ liệu, nghe kênh truyền xem có bận không?
- Nếu kênh truyền trống với một khoảng thời gian lớn hơn hoặc bằng Distributed Inter-Frame Space (DIFS), STA bắt đầu truyền
- Nếu kênh truyền bận, STA phải đợi để tìm được một DIFS trống, sau đó đợi thêm một khoảng ngẫu nhiên số slot time (gọi là khoảng back-off time) để chống đụng độ nếu có STA khác cũng đang muốn truyền (collision avoidance)
- Nếu có STA khác chiếm dụng đường truyền trong khoảng back-off time, trạm STA hiện tại hủy cơ chế back-off timer và đợi DIFS trống khác để truyền



Ví dụ CSMA/CA trên 802.11



busy

Chiếm dụng đường truyền



Có yêu cầu truyền gói tin

bo_e

backoff time hợp lý (elapsed)

bo_r

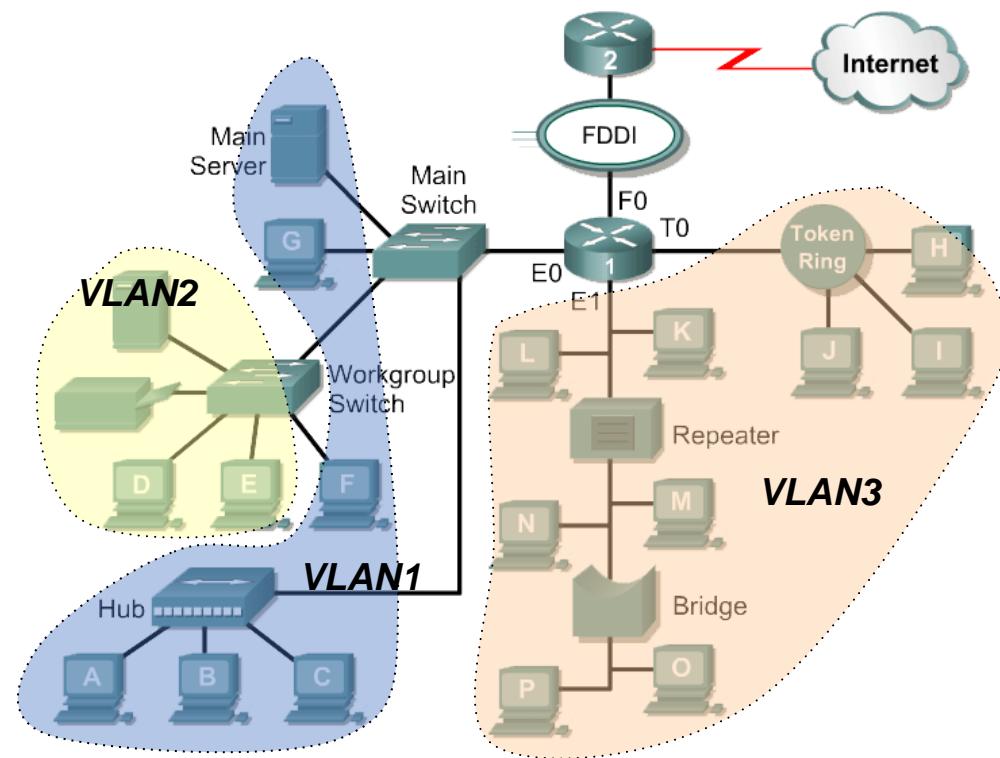
backoff time dư (residual)

Tổng kết Wireless LAN

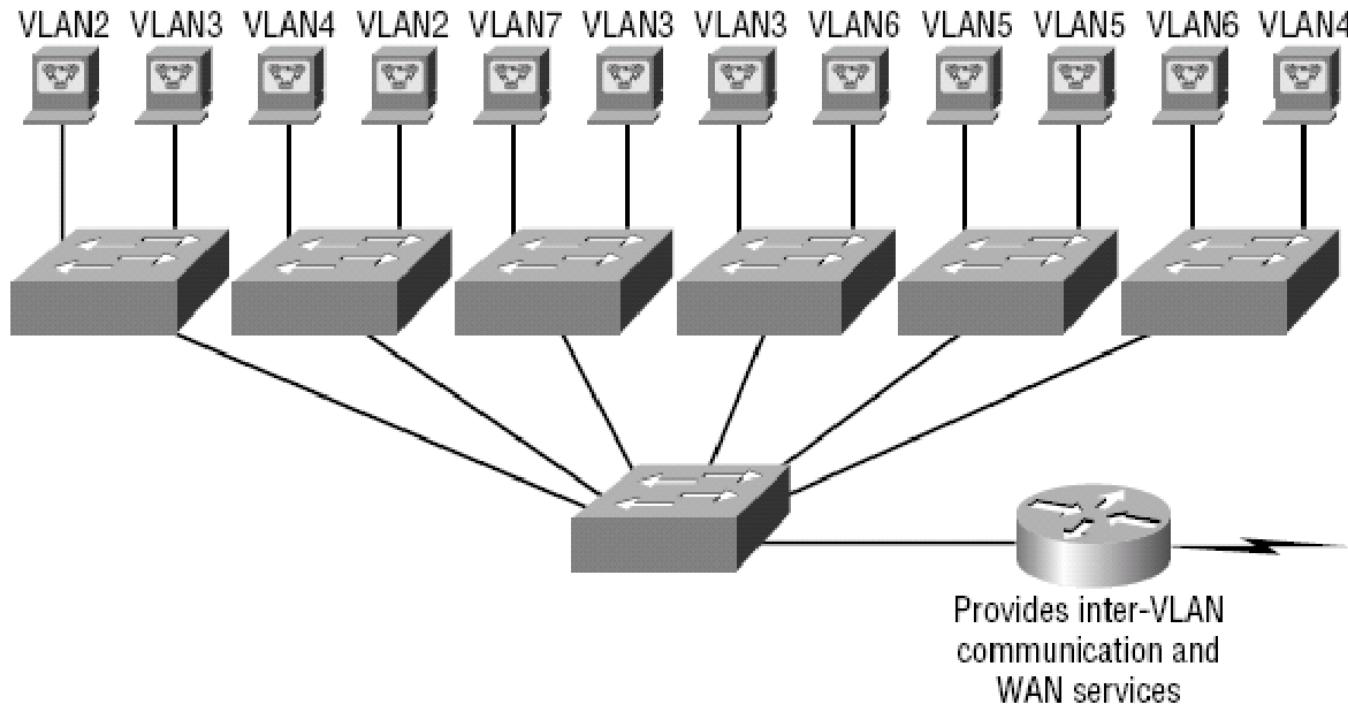
- **Ưu điểm:**
 - Khả năng di động
 - Triển khai dễ dàng
 - Khả năng mở rộng
- **Nhược điểm:**
 - Bảo mật
 - Phạm vi
 - Độ tin cậy
 - Tốc độ
- **Đọc thêm:**
 - Roaming: STA thay đổi AP bên trong một BSS hoặc giữa các BSS (trong một ESS)
 - Wired Equivalent Privacy (WEP): Hỗ trợ bảo mật tương đương mạng có dây.

Mạng LAN ảo – VLAN (Virtual LAN)

- Yêu cầu thực tế
 - Chia sẻ tài nguyên (file, máy in, v.v..) giữa các trạm trong LAN (broadcast zone)
 - Vấn đề trạm “xa nhau”
 - Bảo mật thông tin nội bộ giữa các phòng ban
- Giải pháp mạng LAN ảo
 - Nhóm các trạm vào các broadcast zone (LAN ảo)
 - Broadcast zone không bị ràng buộc về mặt địa lý của các trạm
 - Broadcast zone độc lập với các ứng dụng mạng



Vai trò của Switch

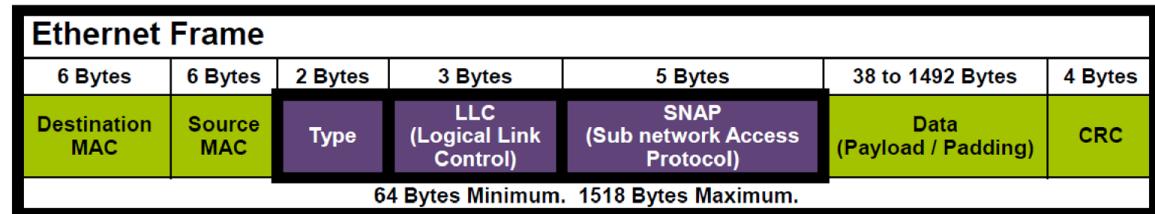
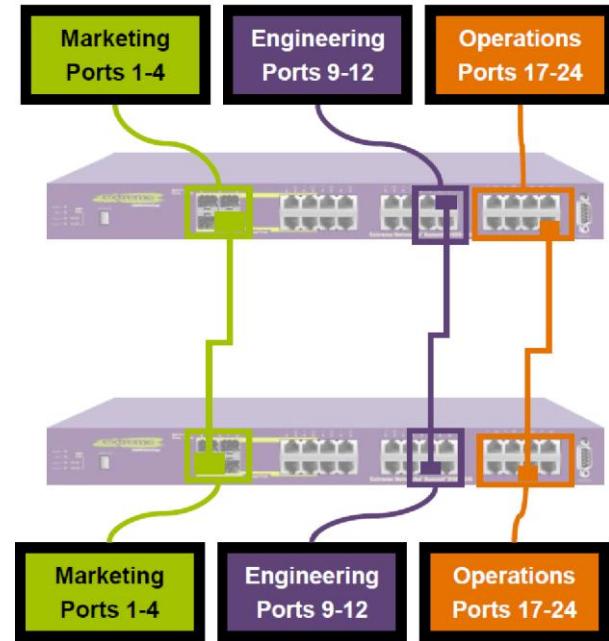


Marketing	VLAN2	172.16.20.0/24
Shipping	VLAN3	172.16.30.0/24
Engineering	VLAN4	172.16.40.0/24
Finance	VLAN5	172.16.50.0/24
Management	VLAN6	172.16.60.0/24
Sales	VLAN7	172.16.70.0/24

- Cấu hình một switch hoặc liên kết các switch để thiết lập broadcast zone (tương ứng VLAN)
- Một switch có thể chứa một hoặc nhiều VLAN

Các phương pháp chia VLAN

- Port-based: chia theo cổng trên switch –
VLAN tĩnh (Static VLAN): tất cả các thiết bị
gắn với cổng đó phải cùng VLAN
- MAC-based: chia theo địa chỉ MAC của
thiết bị - VLAN động (Dynamic VLAN): linh
hoạt
- Protocol-based: chia theo giao thức. Định
nghĩa filter trên các switch, dựa trên các
trường của gói tin (hay sử dụng Type,
LLC, SNAP) để xác định nó thuộc VLAN
nào.



Chuẩn hóa: IEEE802.1Q

- Phương pháp thiết lập VLAN theo cổng switch, địa chỉ MAC hoặc ánh xạ từ giao thức là dựa trên cơ chế hoạt động đặc thù của từng switch.
- Để các switch (của các hãng sản xuất khác nhau) có thể cùng phối hợp hoạt động VLAN, IEEE chuẩn hóa 802.1Q
- 802.1Q bổ sung một số trường dữ liệu vào gói tin Ethernet để switch xác định VLAN cho gói tin này:
 - Tag Protocol ID (TPID)
 - User Priority
 - Canonical Format Indicator (CFI)
 - VLAN Identifier (VID)

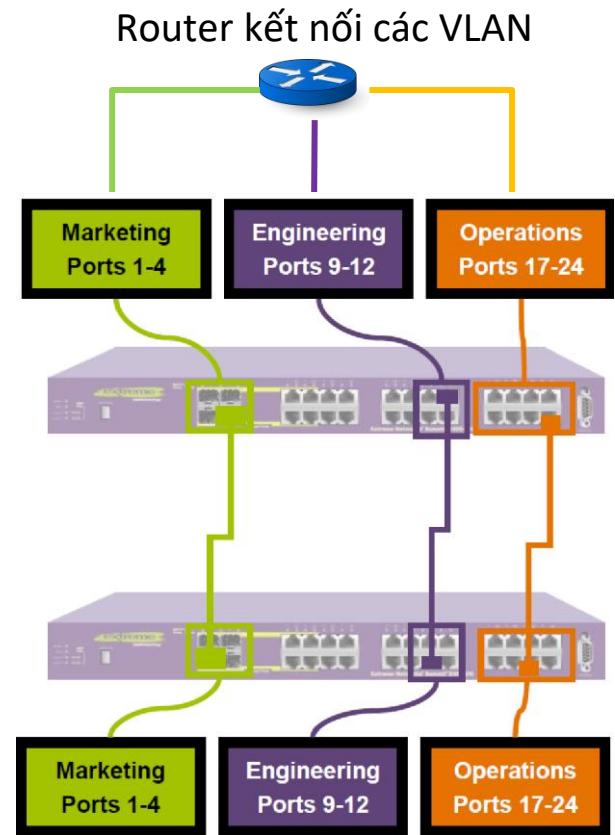
802.1Q Ethernet Frame								
6 Bytes	6 Bytes	2 Bytes	3 bits	1 bit	12 bits	2 Bytes	42 to 1500 Bytes	4 Bytes
Destination MAC	Source MAC	TPID (0x8100)	802.1p	CFI	VLAN ID	Type / Length	Data (Payload / Padding)	CRC
64 Bytes Minimum. 1522 Bytes Maximum.								

Kết nối Switch hỗ trợ VLAN

- Access link:
 - Thuộc về một VLAN đơn lẻ, thường nối trực tiếp từ 1 cổng đến 1 máy trạm.
 - Switch gỡ bỏ các thông tin VLAN trong frame trước khi chuyển tiếp đến cổng chứa access link.
 - Các thiết bị nối với access link không thể truyền thông trực tiếp với thiết bị khác VLAN
- Trunk link:
 - Dùng chung cho nhiều VLAN khác nhau, thường nối giữa switch với nhau hoặc giữa switch với router.
 - Trunk link cho phép 1 cổng thuộc về nhiều VLAN tại cùng một thời điểm để kết nối đến server hoặc với các switch khác

Kết nối các VLAN

- Giống như kết nối các LAN
- Sử dụng router





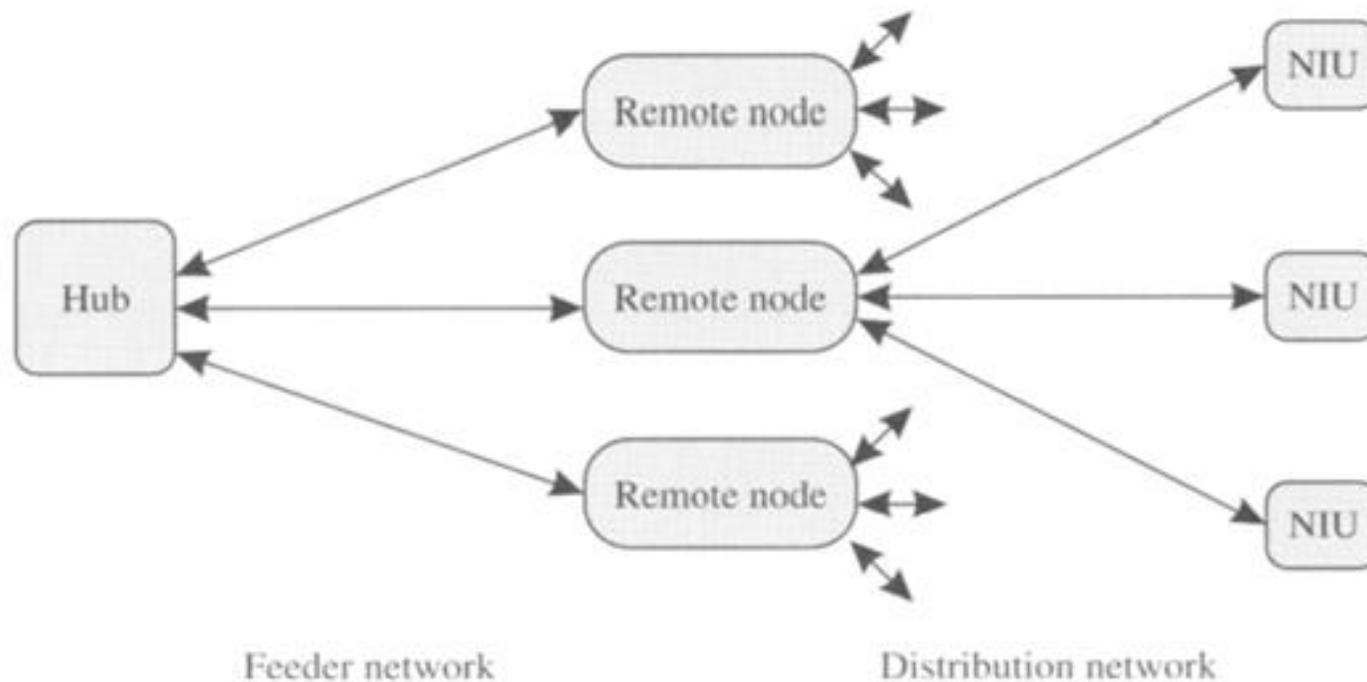
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

7. Mạng truy nhập sử dụng cáp quang

Mạng truy nhập

- Mạng truy nhập thu thập dữ liệu từ phía người dùng và cung cấp cho mạng lõi
- Các dịch vụ phổ biến từ phía người dùng
 - Điện thoại
 - Mạng truyền hình cáp
 - Truyền dữ liệu. Ví dụ trên nền đường truyền điện thoại (xDSL) hoặc cáp quang (FTTH).

Kiến trúc của mạng truy nhập



Kiến trúc bao gồm một hub phía nhà cung cấp dịch vụ, các Remote Node đóng vai trò là nút phân phối ở ngoài trời, các NIU (Network Interface Unit) là các nút mạng phía người dùng
Ref: R. Ramaswami , K. Sivarajan, G. Sasaki, "Optical Networks: A Practical Perspective", 3rd Edition

Kiến trúc mạng truy nhập

- Hub
 - Nằm phía nhà cung cấp
- NIU: Network Interface Unit
 - Nằm phía người sử dụng
 - Nối với 1 người dùng hoặc 1 doanh nghiệp
- Remote Node
 - Trong mạng broadcast, RN phân phối dữ liệu từ Hub đến mọi NIU
 - Trong mạng switched, RN nhận dữ liệu từ Hub và phân phối các luồng khác nhau đến các NIU

Mạng truy nhập quang: FTTx

- Dữ liệu được truyền trên cáp quang trong mạng phân phối (distribution network) cho đến ONU (Optical Network Unit)
 - Mong muốn: Cáp quang đến gần thuê bao nhất
- **FTTCab** (*Fiber To The Cabinet*): Cáp quang kết thúc ở một cabinet, dưới 1km cuối đến thuê bao dùng mạng phân phối cáp đồng.
- **FTTC** (*Fiber To The Curb*) / **FTTB**(*Fiber To The Building*); ONU phục vụ một số thuê bao (8 to 64); từ ONU đến NIU dùng cáp đồng (dưới 100m)
- **FTTH** (*Fiber To The Home*); ONUs thực hiện chức năng của NIUs;
 - ONU: có thể là modem quang.

Mạng truy nhập quang: FTTx

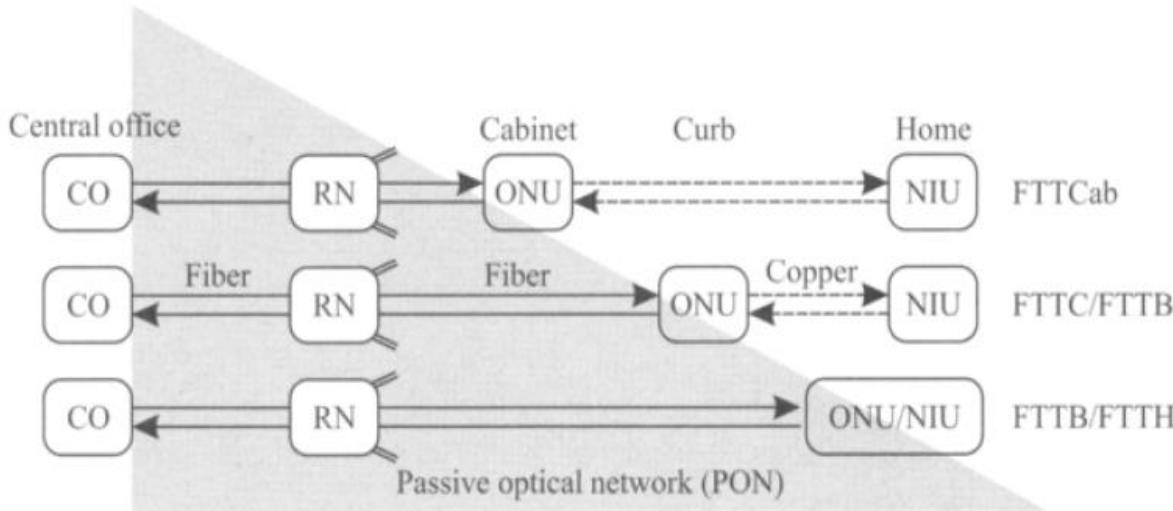
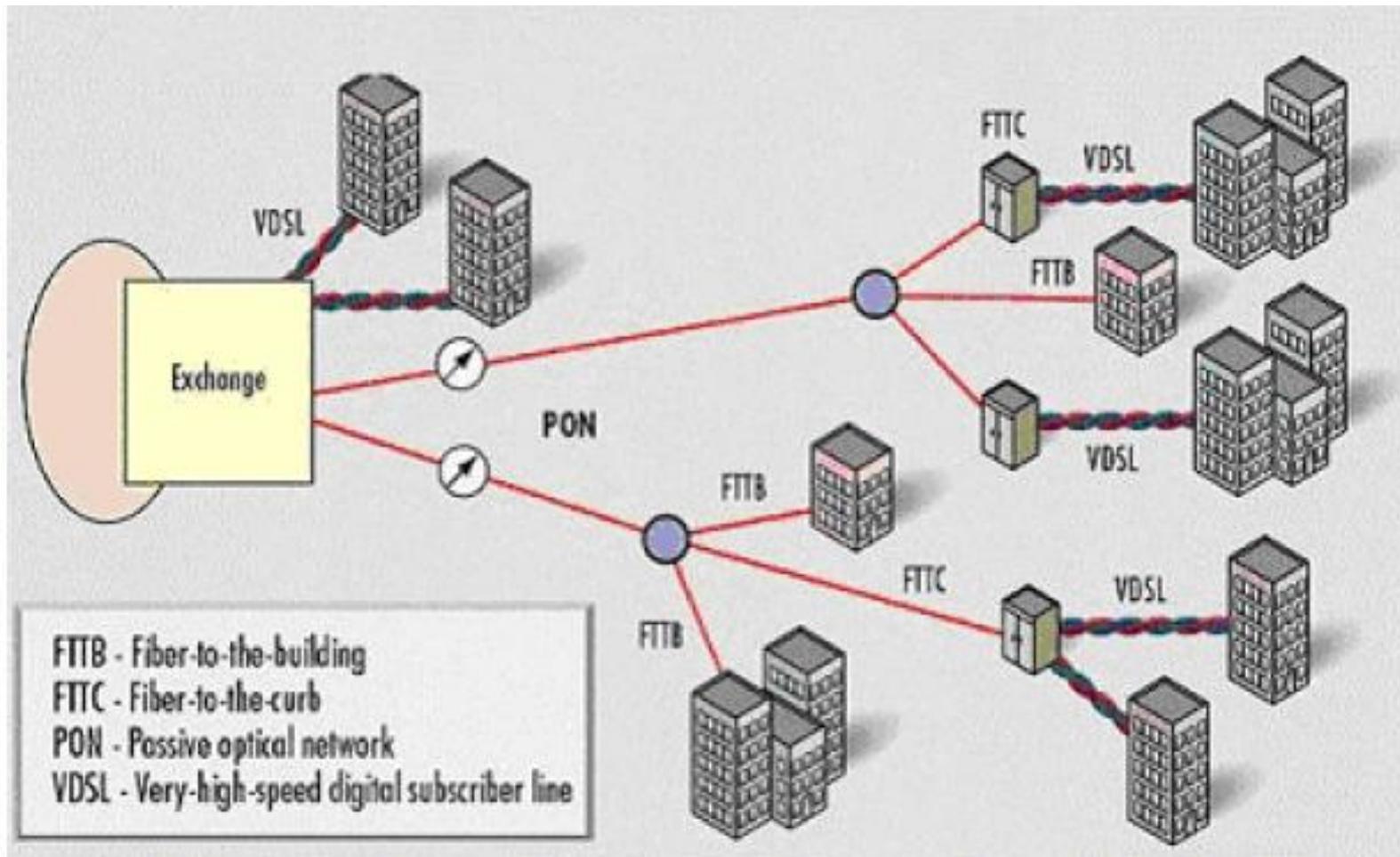


Figure 11.5 Different types of fiber access networks, based on how close the fiber gets to the end user. In many cases, the remote node may be located at the central office itself. The ONUs terminate the fiber signal, and the links between the ONUs and the NIUs are copper based.

Ref: R.Ramaswami , K. Sivarajan, G. Sasaki, “Optical Networks: A Practical Perspective”, 3rd Edition

Mạng truy cập FTTx



AON vs. PON

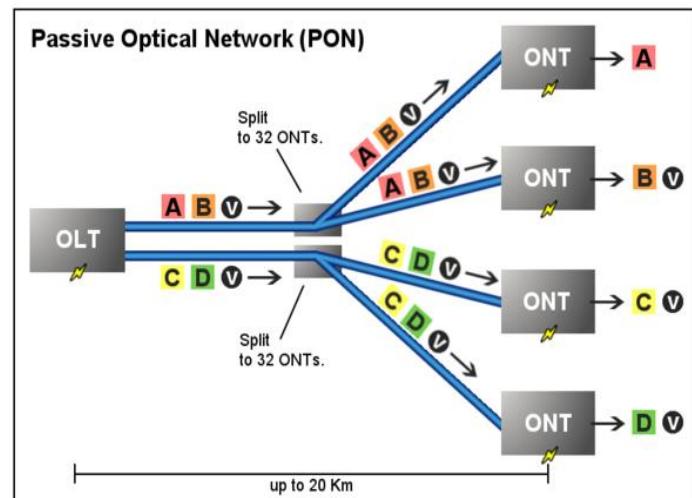
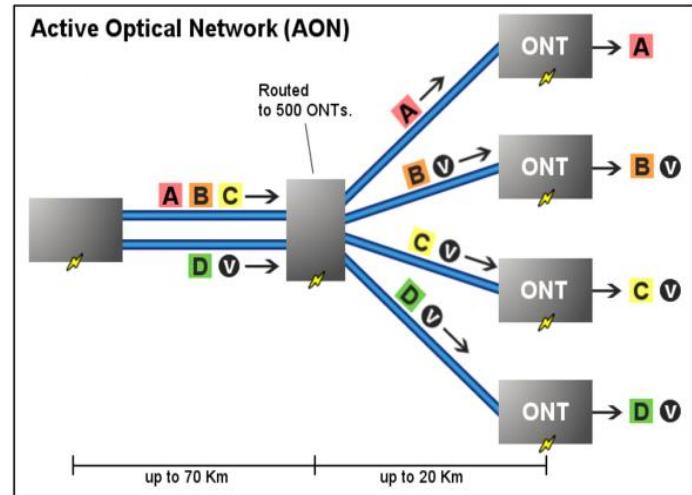
Remote Note (Distribution nodes) chia dữ liệu về các đích.

AON: Active Optical Network

- là mạng sử dụng công nghệ chủ động (Remote Node tiêu thụ điện)
- Remote node phân tích và định tuyến riêng các gói tin theo địa chỉ đích
- Khoảng chạy cáp có thể dài đến 100km

PON: Passive Optical Network

- Là mạng sử dụng công nghệ thụ động. (Remote Node không tiêu thụ điện)
- Remote node (Splitter) không phân tích mà chỉ lặp tín hiệu trên tất cả các cổng ra
- Upstream: MUX từ các nguồn khác nhau bằng TDM (TDM PON) hoặc WDM (WDM PON)
- Khoảng chạy cáp giới hạn 20km



Key: A - Data or voice for a single customer. V - Video for multiple customers.

EPON: Ethernet PON

- EPON: PON vận chuyển dữ liệu là các frame Ethernet
- Chiều xuống (down stream)
 - Quảng bá dữ liệu chung

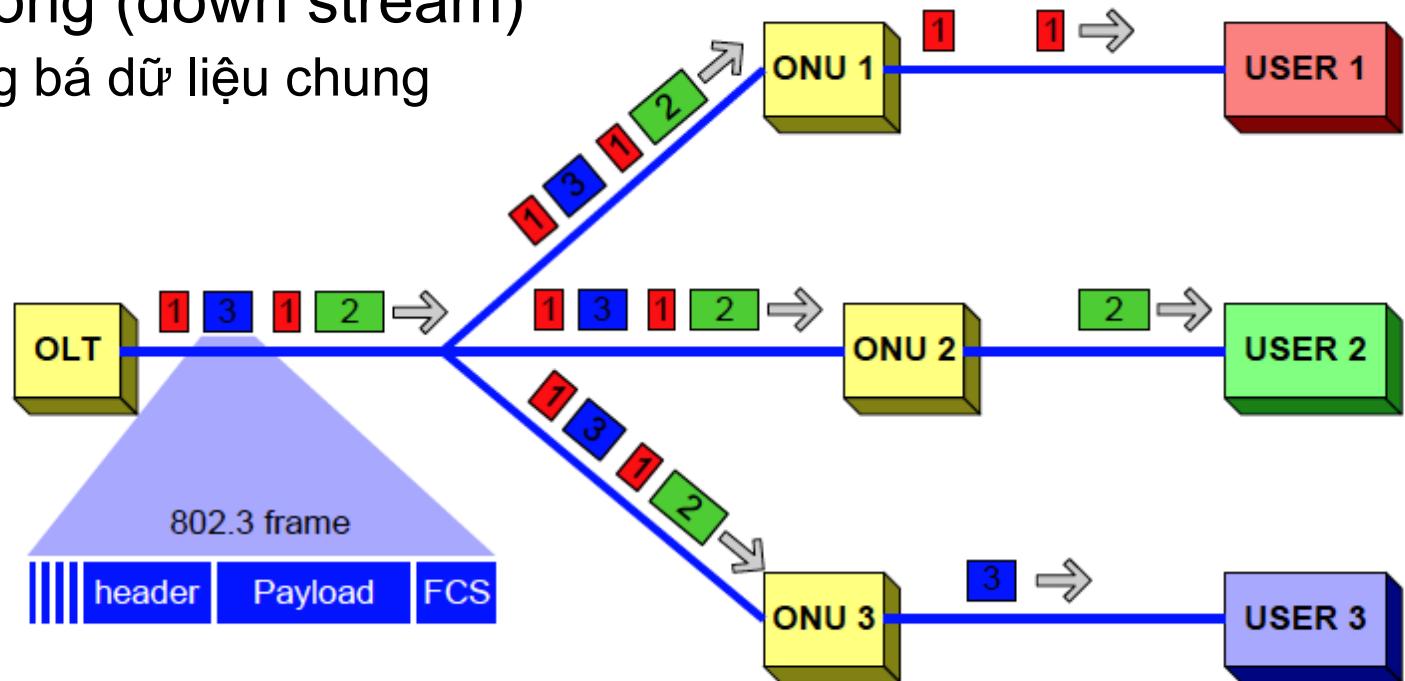


Figure 8-6. Downstream traffic in EPON.

EPON

- Chiều lên (Upstream): dồn kênh theo thời gian (TDM) trực tiếp các gói Ethernet của người dùng từ các nguồn khác nhau vào kết nối chung OLT-RN
- EPON thuộc loại TDM PON

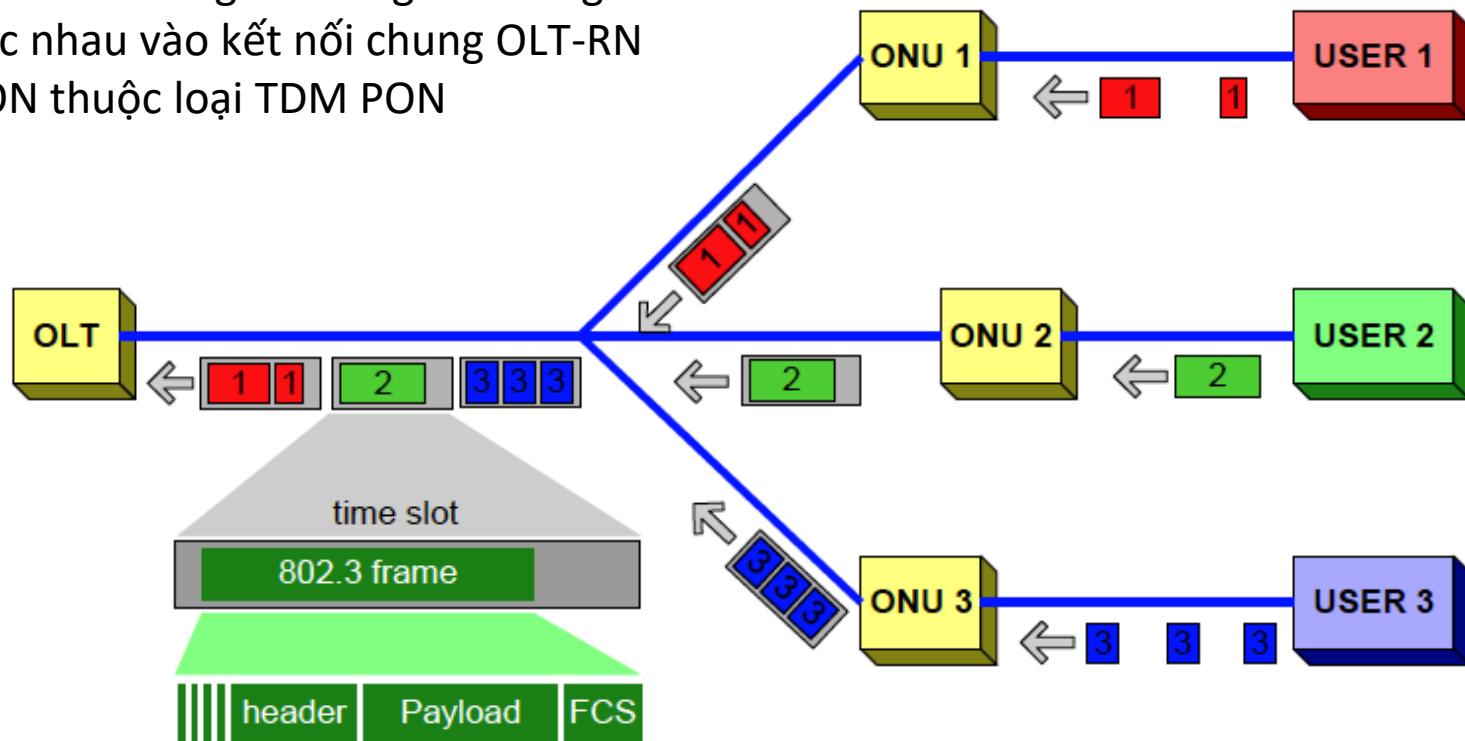


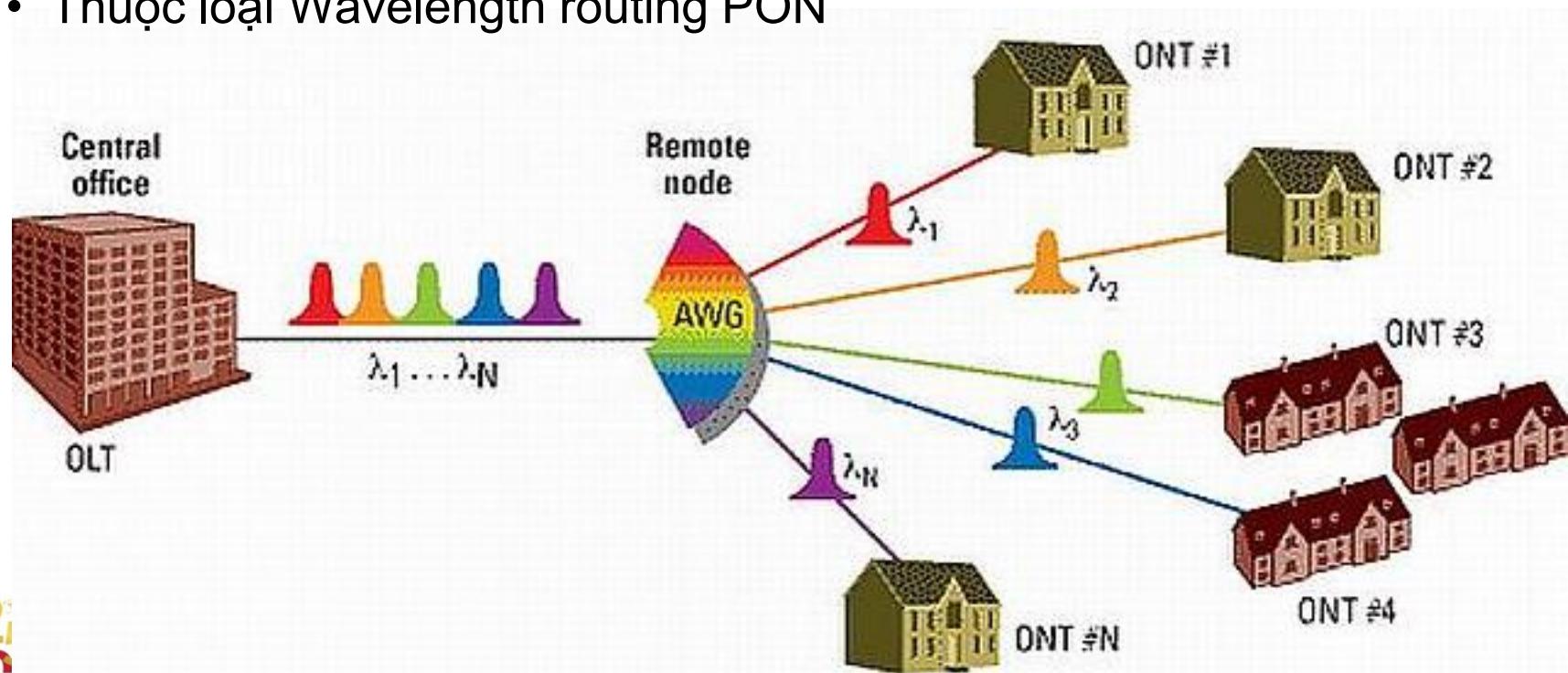
Figure 8-7. Upstream traffic in EPON.

GPON: Gigabit Capable PON

- GPON có thể dùng để tải nhiều dữ liệu khác nhau: Ethernet, ATM, voice ...
- Dữ liệu từ OLT đến người dùng chia sẻ kênh chung giữa OLT và RN
 - Downstream broadcast
 - Upstream TDM
 - Các gói được đóng trong khung dữ liệu GPON có trường định danh người nhận (chiều downstream), người gửi (chiều upstream)

WPON (WDM PON)

- Được phát triển bởi các công ty, chưa chuẩn hóa
- Mỗi ONT sử dụng một bước sóng để truyền dữ liệu
- Remote node là AWG thiết bị có khả năng tách ghép các bước sóng, thực hiện MUX/DEMUX theo bước sóng chiều xuống và lên.
- Thuộc loại Wavelength routing PON





ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chương 4

Tầng liên mạng



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

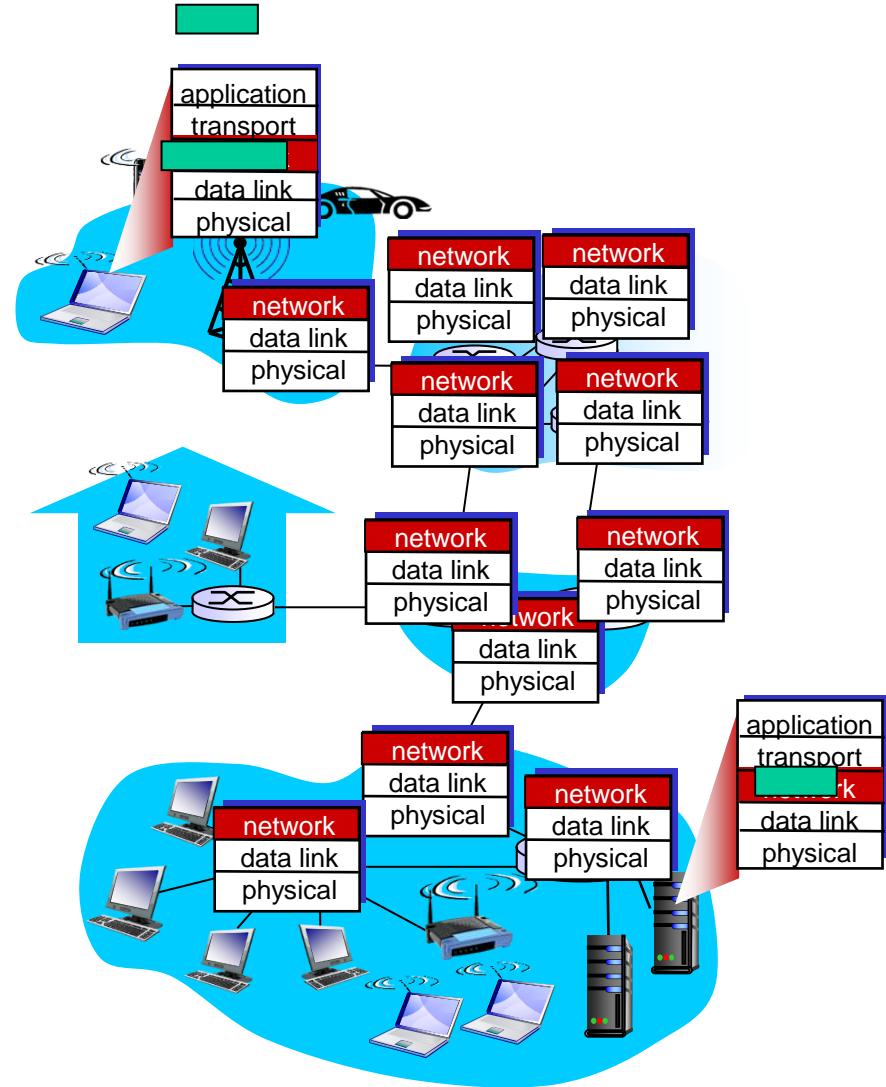
4.1. Tổng quan về tầng liên mạng

Tầng liên mạng trên kiến trúc TCP/IP

Application (HTTP, Mail, ...)	Hỗ trợ các ứng dụng trên mạng
Transport (UDP, TCP ...)	Điều khiển truyền dữ liệu giữa các tiến trình của tầng ứng dụng
Network (IP, ICMP...)	Điều khiển truyền dữ liệu sang mạng khác
Datalink (Ethernet, ADSL...)	Hỗ trợ việc truyền thông cho các thành phần kế tiếp trên cùng 1 mạng
Physical (bits...)	Truyền và nhận dòng bit trên đường truyền vật lý

Tầng mạng

- Truyền dữ liệu từ host-host
- Cài đặt trên mọi hệ thống cuối và bộ định tuyến
- Đơn vị truyền: datagram
- Bên gửi: nhận dữ liệu từ tầng giao vận, đóng gói
- Bên nhận: mở gói, chuyển phần dữ liệu trong payload cho tầng giao vận
- Bộ định tuyến(router): định tuyến và chuyển tiếp gói tin

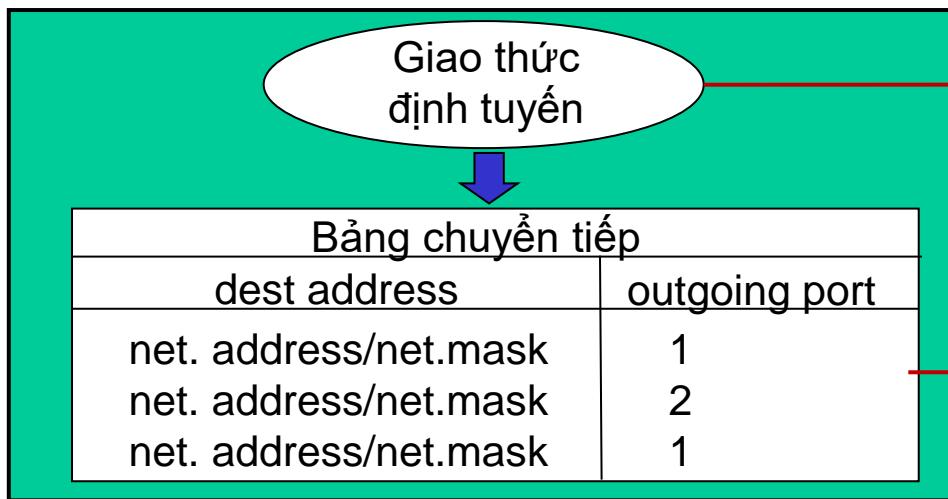


Hình ảnh từ: "Computer Networking: A Top Down Approach", Jim Kurose

Chức năng chính

- **Định tuyến (Routing)**: Tìm tuyến đường để gửi dữ liệu từ nguồn tới đích
- **Chuyển tiếp (Forwarding)**: Chuyển gói tin tới cổng ra theo tuyến đường đã có
- **Định địa chỉ (Addressing)**: Định danh cho các nút mạng
- **Đóng gói dữ liệu (Encapsulating)**
- **Đảm bảo chất lượng dịch vụ (QoS)**: đảm bảo các thông số phù hợp của đường truyền theo từng dịch vụ

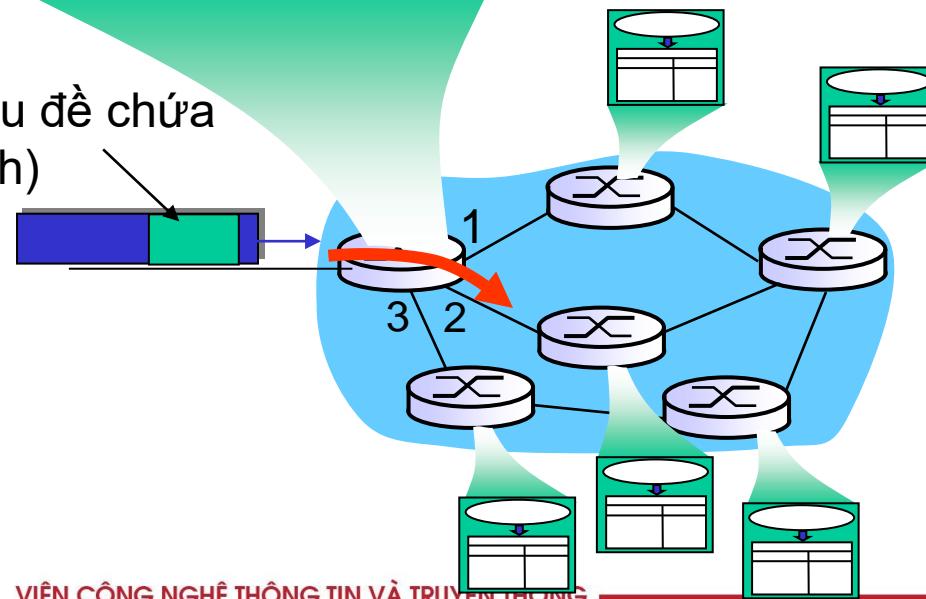
Định tuyến và chuyển tiếp



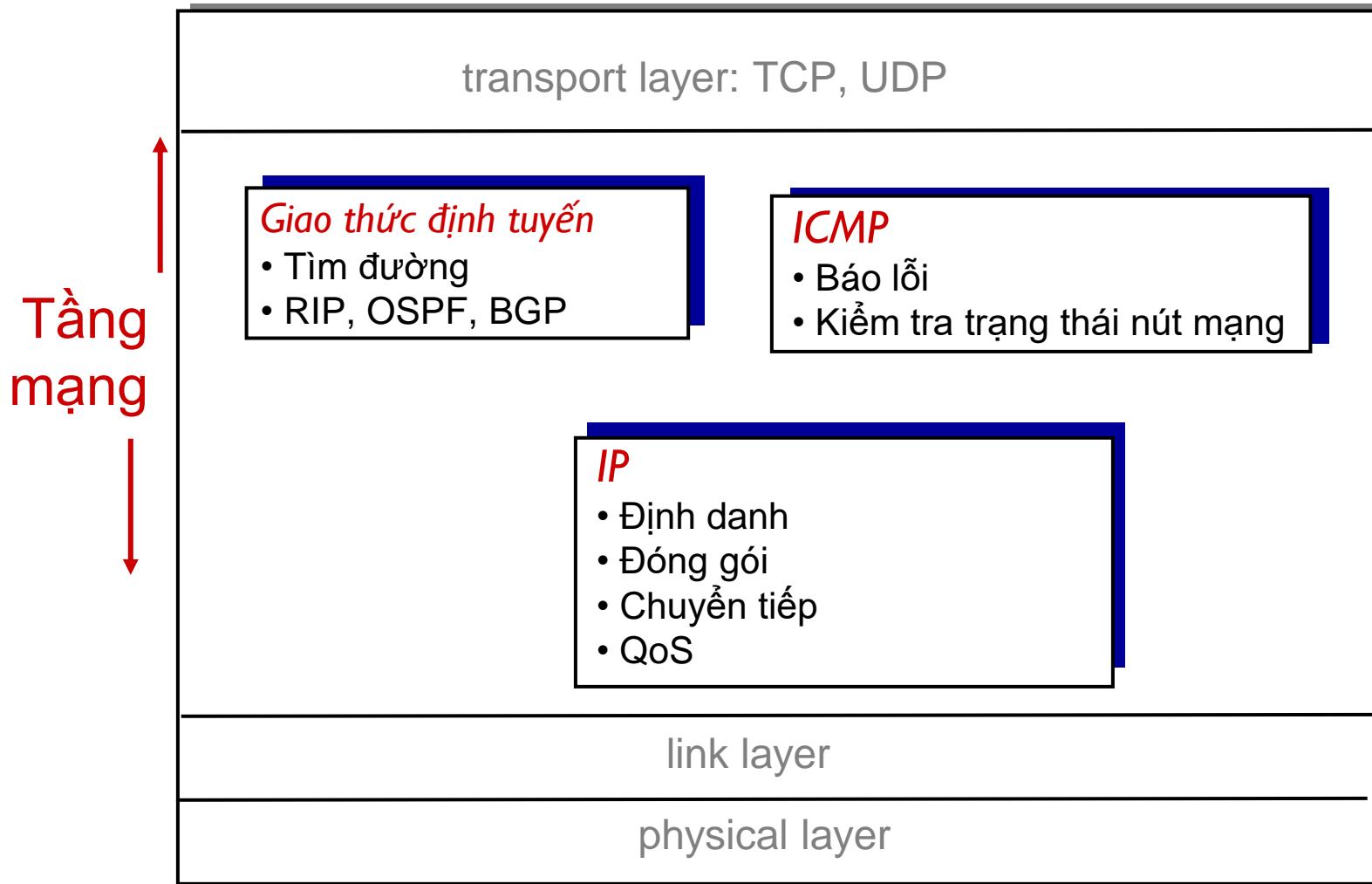
Giao thức định tuyến xác định đường đi ngắn nhất giữa 2 bên truyền tin

Bảng chuyển tiếp xác định cổng ra (outgoing port) để chuyển dữ liệu tới đích

Gói tin (tiêu đề chứa địa chỉ đích)



Các giao thức tầng mạng





ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4.6. Giao thức IPv4

Đặc điểm cơ bản của giao thức

- Giao thức hướng không liên kết
- Không tin cậy / nhanh
 - Truyền dữ liệu theo phương thức “best effort”
 - IP không có cơ chế phục hồi nếu có lỗi
 - Khi cần, ứng dụng sẽ sử dụng dịch vụ tầng trên để đảm bảo độ tin cậy (TCP)
- Là giao thức được định tuyến (**routed protocol**)
 - Đòi hỏi phải có các giao thức định tuyến để xác định trước đường đi cho dữ liệu.
- Hiện nay có 2 phiên bản: IPv4 và IPv6

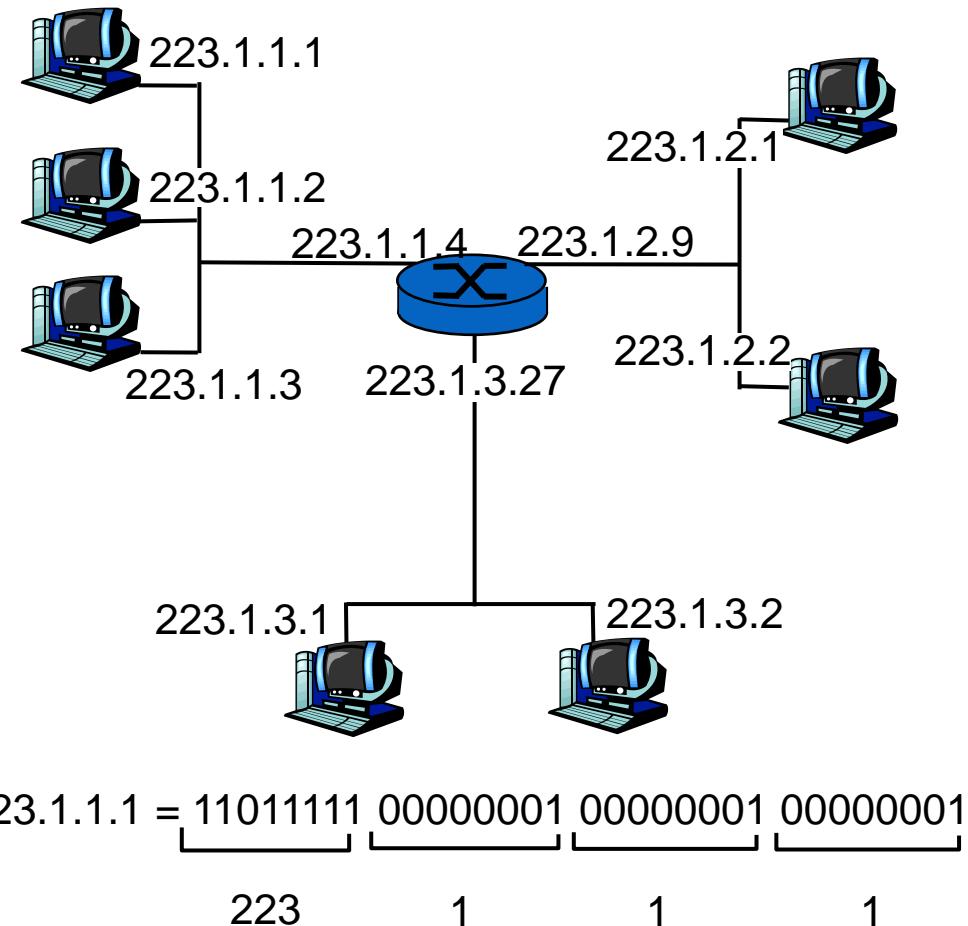


ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Địa chỉ IPv4

Địa chỉ IP (IPv4)

- **Địa chỉ IPv4:** Một số 32-bit để định danh cổng kết nối mạng trên nút đầu cuối (PC, server, smart phone), bộ định tuyến
- Mỗi địa chỉ IP được gán cho một cổng duy nhất
- Địa chỉ IP có tính duy nhất trong mạng

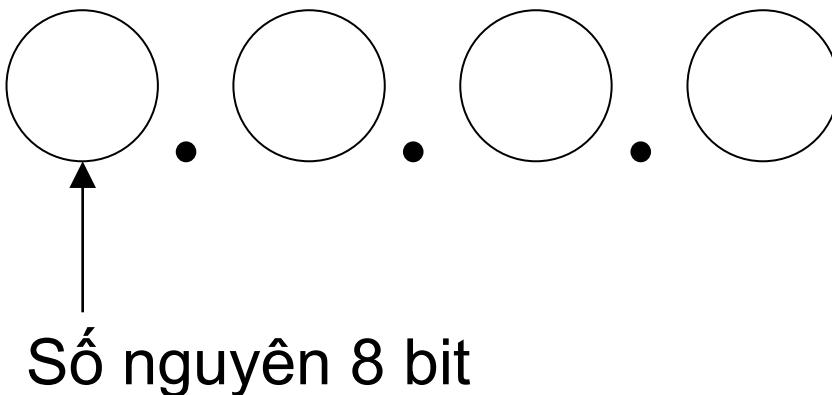


Cấp phát địa chỉ IPv4

- Cấp phát cố định(Static IP):
 - Windows: Control Panel → Network → Configuration → TCP/IP → Properties
 - Linux: /etc/network/interfaces
- Cấp phát tự động: **DHCP- Dynamic Host Configuration Protocol**

Biểu diễn địa chỉ IPv4

- Biểu diễn dạng thập phân có chấm



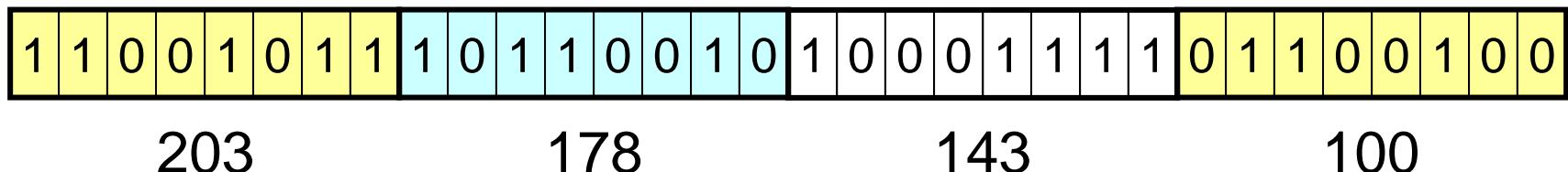
Ví dụ:

203.178.136.63 o

259.12.49.192 x

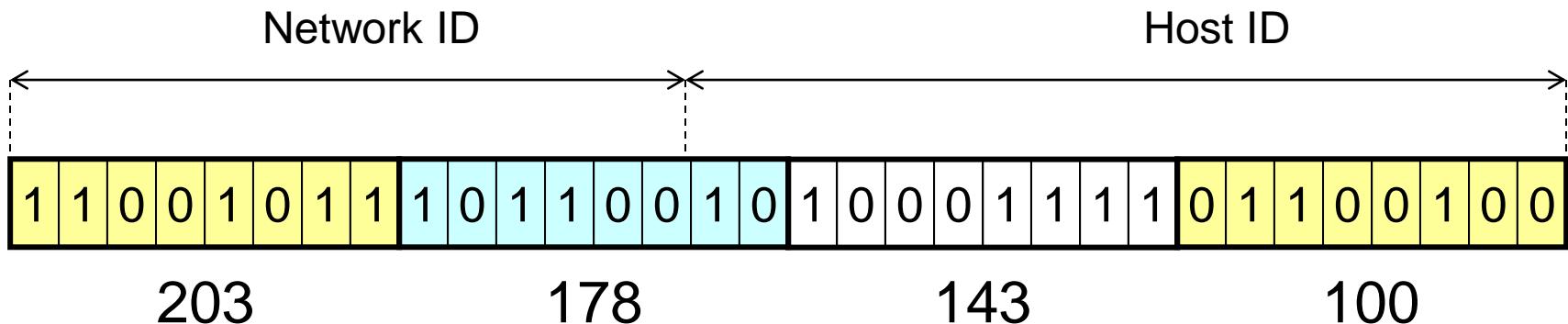
133.27.4.27 o

Sử dụng 4 phần 8 bit để miêu tả một địa chỉ 32 bits



Địa chỉ IPv4

- Địa chỉ IP có hai phần
 - Host ID – phần địa chỉ máy trạm: (32-n) bit cuối
 - Network ID – phần địa chỉ mạng: n bit đầu

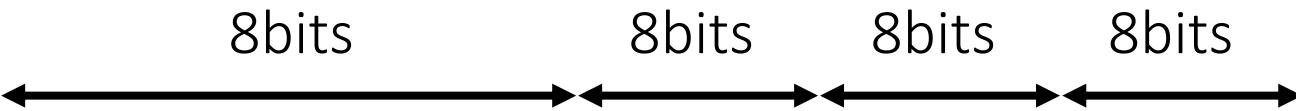


- Làm thế nào biết được phần nào là cho máy trạm, phần nào cho mạng?
 - Phân lớp địa chỉ
 - Không phân lớp – CIDR

Các dạng địa chỉ

- Địa chỉ mạng (Network Address):
 - Định danh cho một mạng
 - Tất cả các bit phần HostID là 0
 - 0.0.0.0: Địa chỉ toàn mạng, đại diện cho 1 mạng bất kỳ
- Địa chỉ quảng bá (Broadcast Address)
 - Địa chỉ dùng để gửi dữ liệu cho tất cả các máy trạm trong mạng
 - Tất cả các bit phần HostID là 1
 - 255.255.255.255: Địa chỉ quảng bá trong 1 mạng
- Địa chỉ máy trạm (Unicast Address)
 - Gán cho một cổng mạng
- Địa chỉ nhóm (Multicast address): định danh cho nhóm

Phân lớp địa chỉ IP(Classful Addressing)



		8bits	8bits	8bits	8bits
Class A	0	7 bit	H	H	H
Class B	1 0	6 bit	N	H	H
Class C	1 1 0	5 bit	N	N	H
Class D	1 1 1 0	Multicast			
Class E	1 1 1 1	Reserve for future use			

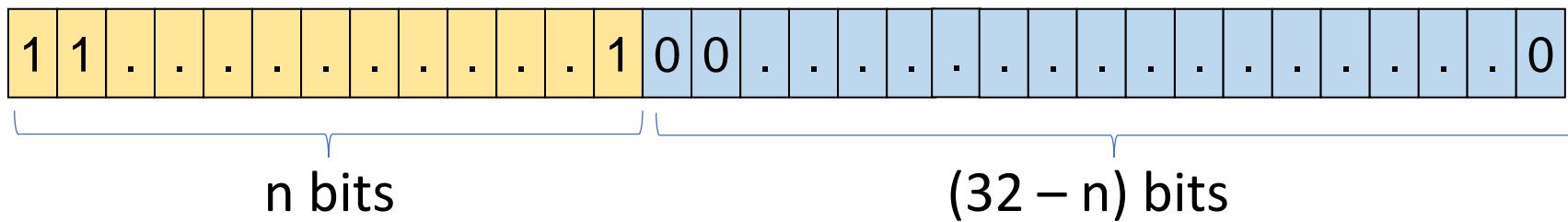
	# of network	# of hosts/1 net
Class A	128	$2^{24} - 2$
Class B	16384	65534
Class C	2^{21}	254

Hạn chế của việc phân lớp địa chỉ

- Lãng phí không gian địa chỉ
 - Việc phân chia thành các lớp (A, B, C, D, E) làm hạn chế việc sử dụng toàn bộ không gian địa chỉ
- Phương pháp CIDR: **Classless Inter Domain Routing**
 - Classless addressing
 - Phần địa chỉ mạng (NetworkID) sẽ có độ dài bất kỳ
 - Dạng địa chỉ: $m_1.m_2.m_3.m_4/n$, trong đó n (mặt nạ mạng) là số bit trong phần ứng với địa chỉ mạng (NetworkID)

Mặt nạ mạng

- Mặt nạ mạng xác định 2 phần trong địa chỉ IPv4
 - Phần ứng với máy trạm (HostID)
 - Phần ứng với mạng (NetworkID)
- Biểu diễn:
 - Prefix: /n. Ví dụ: /27
 - n: Số bit của NetworkID
 - Nhị phân:



- Ví dụ: /27 = 11111111 11111111 11111111 11100000
- Thập phân có chấm: 255.255.255.224

Địa chỉ IP và mặt nạ mạng

- Địa chỉ nào là địa chỉ máy trạm, địa chỉ mạng, địa chỉ quảng bá?
 - (1) 203.178.142.128 /25
 - (2) 203.178.142.128 /24
 - (3) 203.178.142.127 /25
 - (4) 203.178.142.127 /24
- Lưu ý: Với cách địa chỉ hóa theo CIDR, địa chỉ IP và mặt nạ mạng luôn phải đi cùng nhau

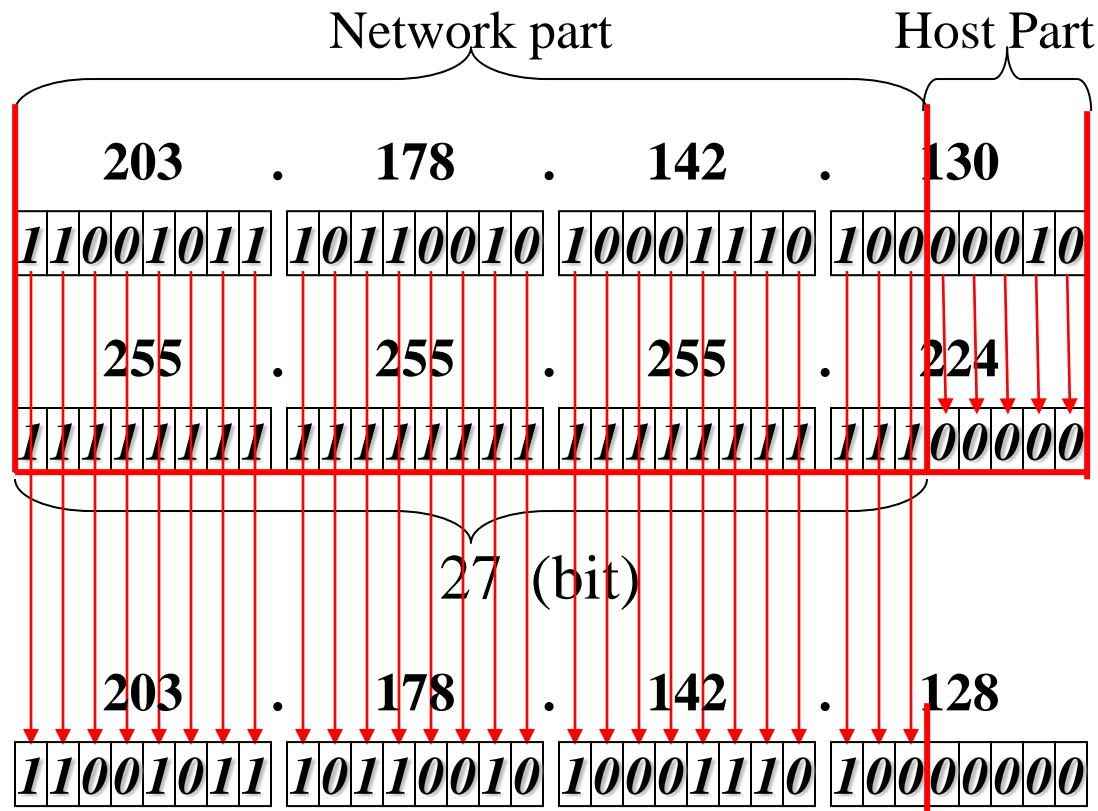
Cách tính địa chỉ mạng

IP Address

Netmask (/27)

AND

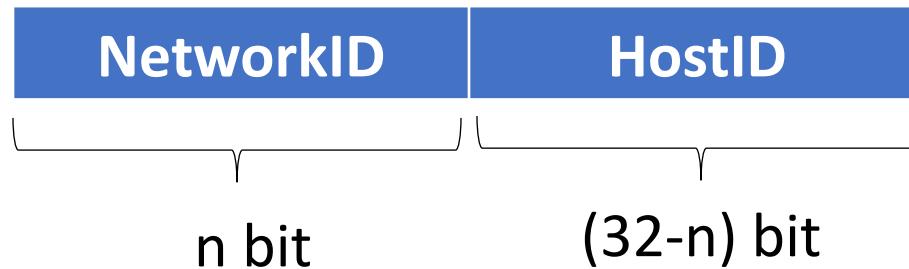
Network address



203.178.142.128 /27

Mặt nạ mạng và kích thước mạng

- Mặt nạ mạng: /n
- Kích thước mạng: Số nút mạng tối đa có trong mạng
- Địa chỉ IP:

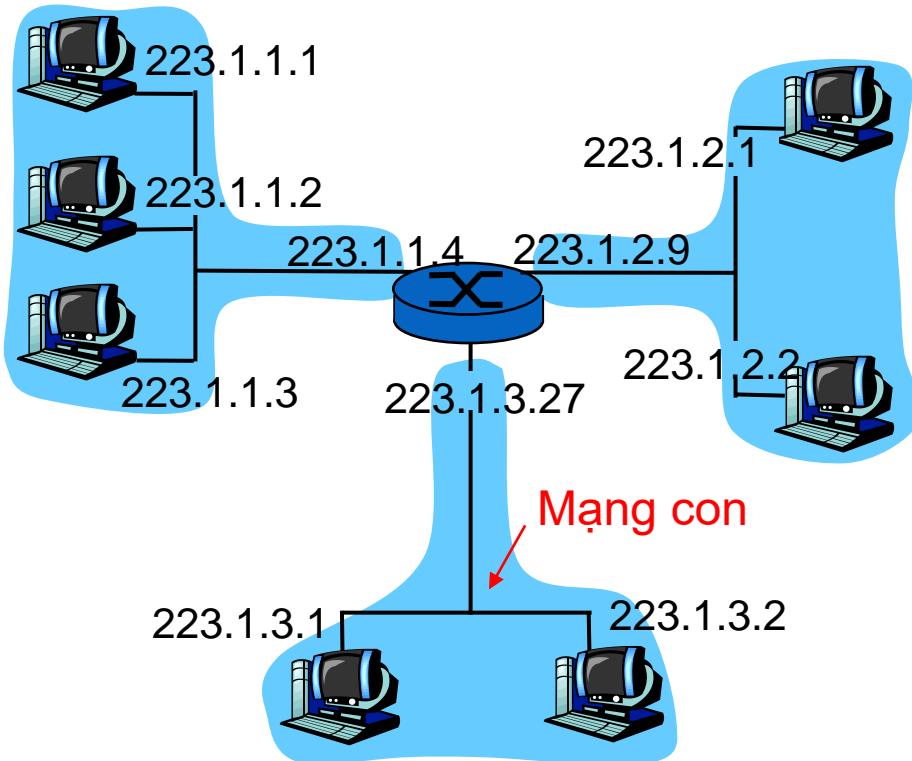


- Số lượng địa chỉ máy trạm (Unicast Address):

$$2^{32-n} - 2$$

Mạng con - subnet

- Là một phần của một mạng nào đó
 - ISP thường được gán một khối địa chỉ IP
 - Một vài mạng con sẽ được tạo ra
- Tạo subnet như thế nào
 - Sử dụng một mặt nạ mạng lớn hơn



Mạng với 3 mạng con

Ví dụ: Chia làm 2 subnets

11001000 00010111 00010000 00000000
200. 23. 16. 0 /24



SubnetID

11001000	00010111	00010000	00000000	
200.	23.	16.	0	/25
11001000	00010111	00010000	10000000	
200.	23.	16.	128	/25

Mặt nạ mạng con = Mặt nạ mạng cũ + Số bit SubnetID
Chia thành 4 mạng con? → Mặt nạ mạng con: /26

Bài tập: Chia mạng con

- Cho một mạng có địa chỉ 200.23.16.0/24

1) Cần chia mạng này thành 8 mạng con

- Địa chỉ mỗi mạng con là gì? Mặt nạ mạng? Số máy tối đa mỗi mạng

2) Câu hỏi chung: Cần tạo N mạng con? Mạng con cần một số lượng địa chỉ nhất định? Địa chỉ mạng mỗi mạng con? Mặt nạ mạng?

- Mỗi mạng con cần 14 máy /28
- Mỗi mạng con cần 30 máy → /27
- Mỗi mạng con cần 31 máy /26
- Mỗi mạng con cần 70 máy /25

Đáp án

- 200.23.16.0 /27 → **0000 0000**
- 200.23.16.32 /27 → **0010 0000**
- 200.23.16.64 /27 → **0100 0000**
- 200.23.16.96 /27 → **0110 0000**
- 200.23.16.128 /27 → **1000 0000**
- 200.23.16.160 /27 → **1010 0000**
- 200.23.16.192 /27 → **1100 0000**
- 200.23.16.224 /27 → **1110 0000**

Không gian địa chỉ IPv4

- Theo lý thuyết
 - Có thể là 0.0.0.0 ~ 255.255.255.255
 - Một số địa chỉ đặc biệt
- Địa chỉ IP đặc biệt ([RFC1918](#))

Private address	10.0.0.0/8 172.16.0.0/16 → 172.31.0.0/16 192.168.0.0/24 → 192.168.255.0 /24
Loopback address	127.0.0.0 /8
Multicast address	224.0.0.0 ~239.255.255.255

- Địa chỉ liên kết nội bộ: 169.254.0.0/16 (tự động cấu hình)

Quản lý địa chỉ IP công cộng

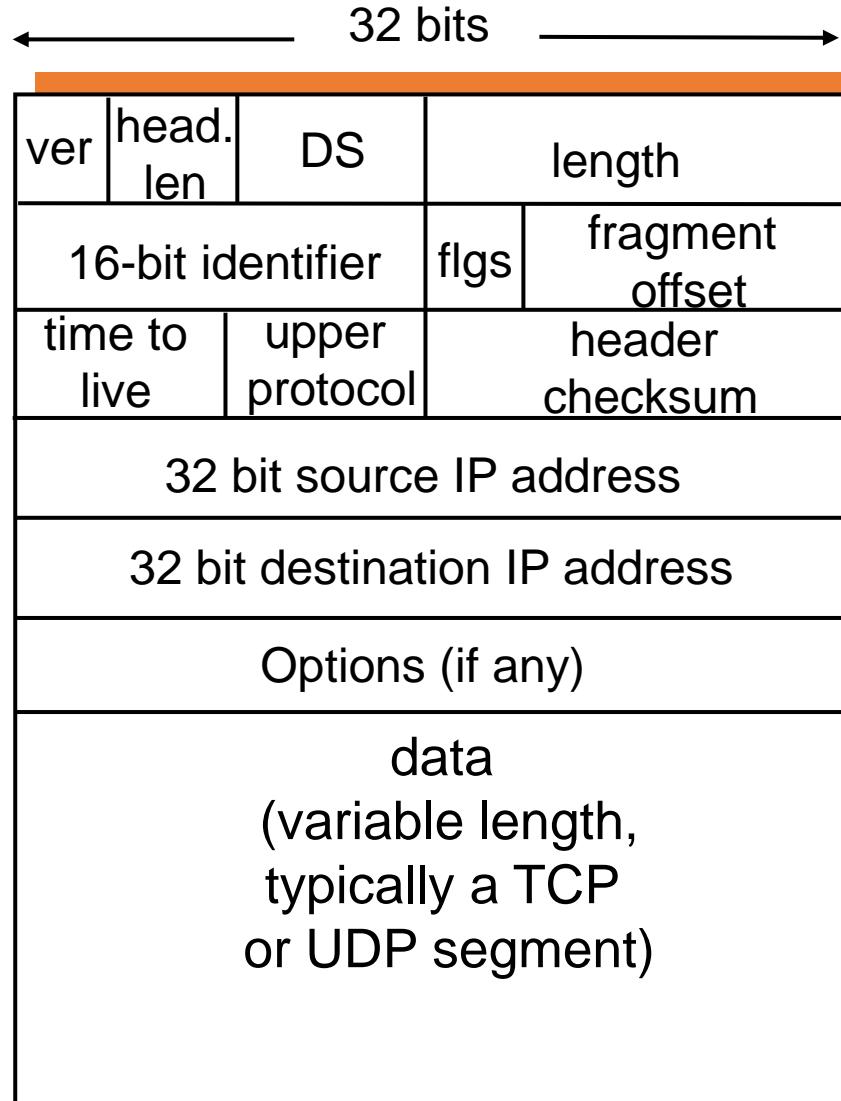
- Internet Corporation for Assigned Names and Numbers (ICANN): quản lý toàn bộ tài nguyên địa chỉ IP
- Regional Internet Registries: quản lý địa chỉ IP theo vùng (châu Á-Thái Bình Dương, châu Âu và Trung Đông, châu Phi, Bắc Mỹ, Nam Mỹ)
- Cơ quan quản lý quốc gia
 - Việt Nam: VNNIC
- Nhà cung cấp dịch vụ (ISP)
- Cơ quan, tổ chức
- Ví dụ: ICANN → APNIC → VNNIC → HUST



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Khuôn dạng gói tin IP

Tiêu đề gói tin IP



Tiêu đề gói tin IP (1)

- Version: Phiên bản giao thức (4 bits)
 - IPv4
 - IPv6
- Header length: Độ dài phần đầu: 4bits
 - Tính theo từ (4 bytes)
 - Min: 5×4 (byte)
 - Max: 15×4 (byte)
- DS (Differentiated Service : 8bits)
 - Tên cũ: Type of Service
 - Hiện tại được sử dụng trong quản lý QoS

Tiêu đề gói tin IP (2)

- Length: Độ dài toàn bộ, tính cả phần đầu (16 bits)
 - Theo bytes
 - Max: 65536
- Identifier – Số hiệu gói tin (16 bit)
 - Dùng để xác định một chuỗi các fragments của một gói tin bị phân mảnh
- Flag – Cờ báo phân mảnh(3 bit)
- Fragmentation offset – Vị trí gói tin phân mảnh trong gói tin ban đầu (13 bit)

Tiêu đề gói tin IP (3)

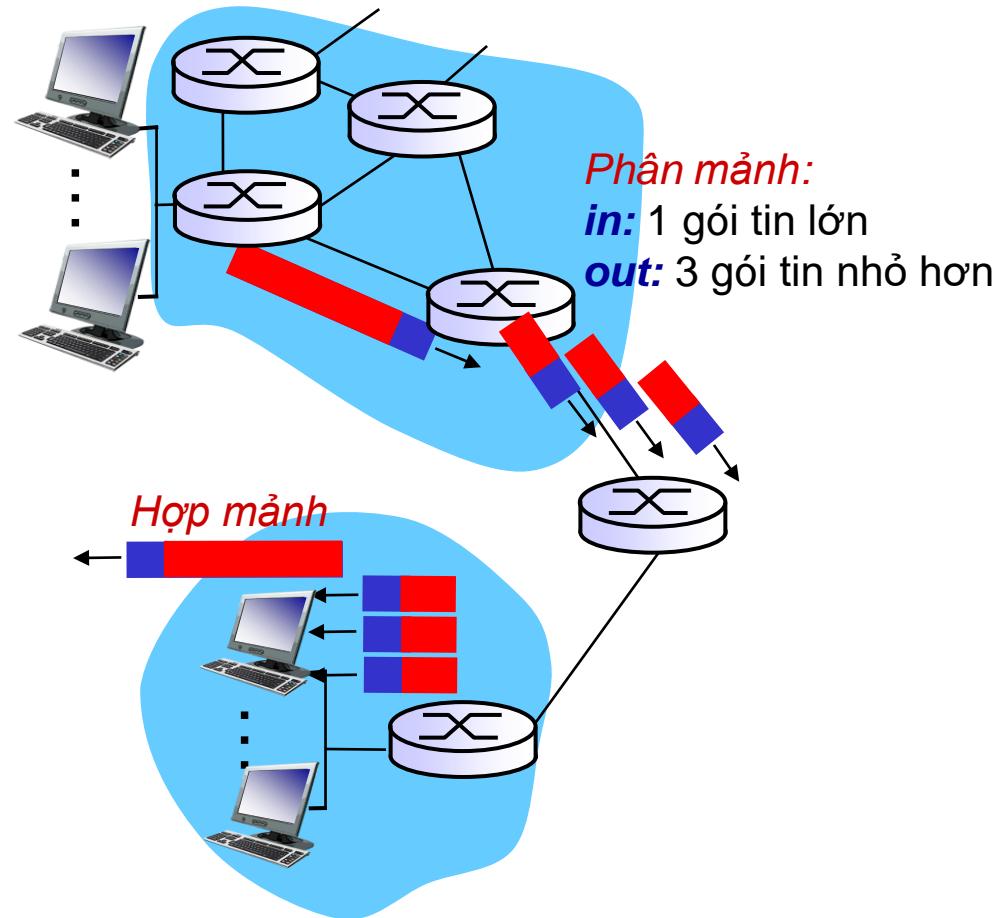
- TTL, 8 bits – Thời gian sống
 - Độ dài đường đi gói tin có thể đi qua
 - Max: 255
 - Router giảm TTL đi 1 đơn vị khi xử lý
 - Gói tin bị hủy nếu TTL bằng 0
- Upper protocol – giao thức tầng trên
 - Giao thức giao vận phía trên (TCP, UDP,...)
 - Các giao thức tầng mạng khác (ICMP, IGMP, OSPF) cũng có trường này
 - Sử dụng để dồn kênh/phân kênh

Tiêu đề gói tin IP (4)

- Checksum – Mã kiểm soát lỗi
- Địa chỉ IP nguồn
 - 32 bit, địa chỉ của trạm gửi
- Địa chỉ IP đích
 - 32 bit, địa chỉ của trạm đích

Phân mảng gói tin (1)

- Đường truyền có một giá trị MTU (Kích thước đơn vị dữ liệu tối đa)
- Các đường truyền khác nhau có MTU khác nhau
- Một gói tin IP có kích thước lớn quá MTU sẽ bị
 - Chia làm nhiều gói tin nhỏ hơn
 - Được tập hợp lại tại trạm đích



Phân mảnh (2)

- Trường Identification
 - ID được sử dụng để tìm các phần của gói tin
- Flags – cờ (3 bits)
 - 1st bit: Dự phòng
 - 2nd bit: Không được phép phân mảnh
 - 3rd bit: Còn phân mảnh
- Độ lệch - Offset
 - Vị trí của gói tin phân mảnh trong gói tin ban đầu
 - Theo đơn vị 8 bytes

Phân mảnh (3)

Ví dụ:

- ❖ Gói tin: 4020 byte
(header: 20 byte)
- ❖ MTU = 1500 bytes

1480 bytes in
data field

	length =4020	ID =x	fragflag =0	offset =0	
--	-----------------	----------	----------------	--------------	--

offset =
 $1480/8$

	length =1500	ID =x	fragflag =1	offset =0	
--	-----------------	----------	----------------	--------------	--

	length =1500	ID =x	fragflag =1	offset =185	
--	-----------------	----------	----------------	----------------	--

	length =1060	ID =x	fragflag =0	offset =370	
--	-----------------	----------	----------------	----------------	--



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4.3. Chuyển đổi địa chỉ (NAT)

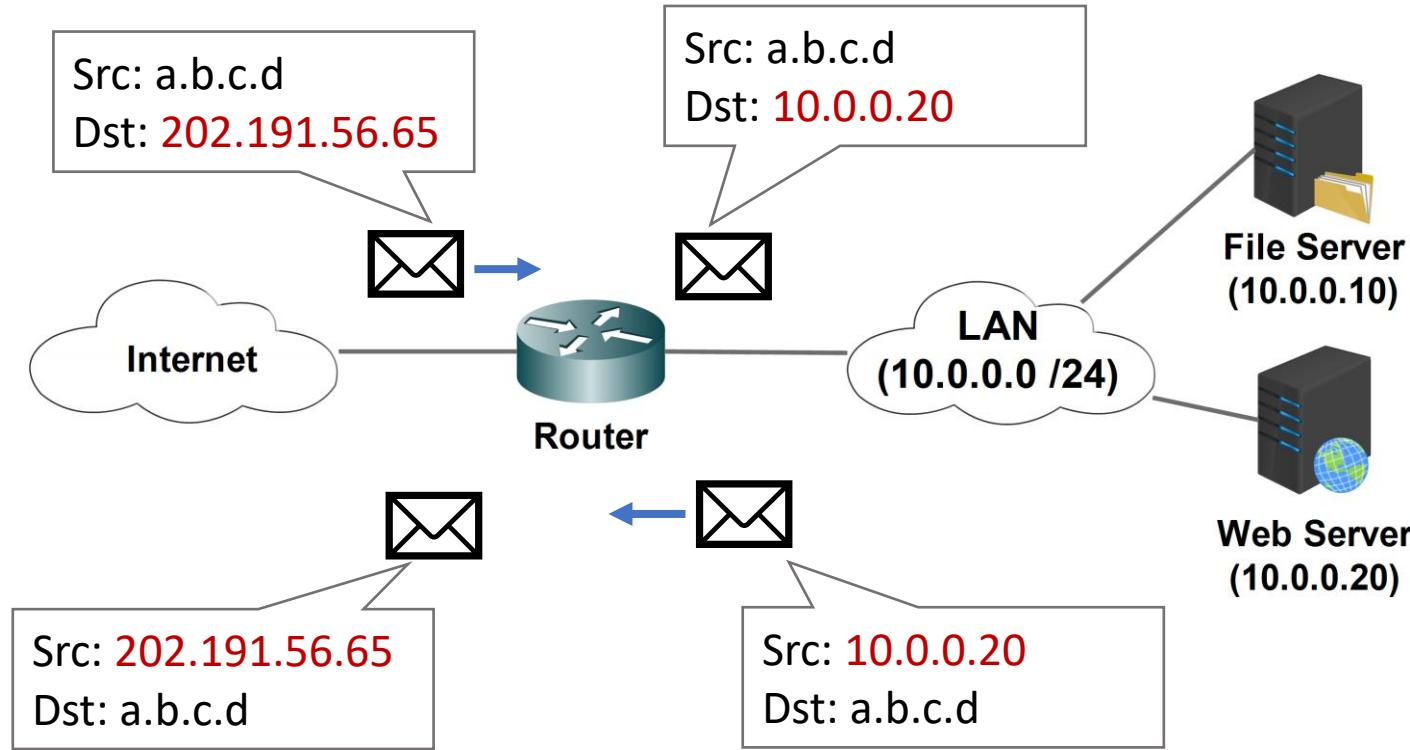
Khái niệm cơ bản NAT

- Dữ liệu chuyển tiếp từ mạng LAN(sử dụng địa chỉ cục bộ) sang mạng Internet(sử dụng địa chỉ công cộng) và ngược lại cần được chuyển đổi địa chỉ
→ NAT(Network Address Translation) thực hiện trên bộ định tuyến
- Trên thực tế, có thể sử dụng NAT để chuyển đổi địa chỉ IP từ mạng LAN này sang mạng LAN khác
- Lợi ích:
 - Tiết kiệm địa chỉ IP công cộng
 - Che giấu địa chỉ riêng
 - Giảm chi phí cấu hình khi thay đổi ISP

Static NAT

- Là dạng NAT đơn giản nhất
- Một địa chỉ IP private được gán cho một địa chỉ IP public xác định
- NAT router sẽ lưu một bảng chuyển đổi trong bộ nhớ
 - Bảng này ánh xạ các địa chỉ nội bộ bên trong với địa chỉ bên ngoài Internet
- Thường sử dụng cho các server nằm trong mạng LAN để cung cấp dịch vụ cho mạng công cộng

Static NAT - Ví dụ



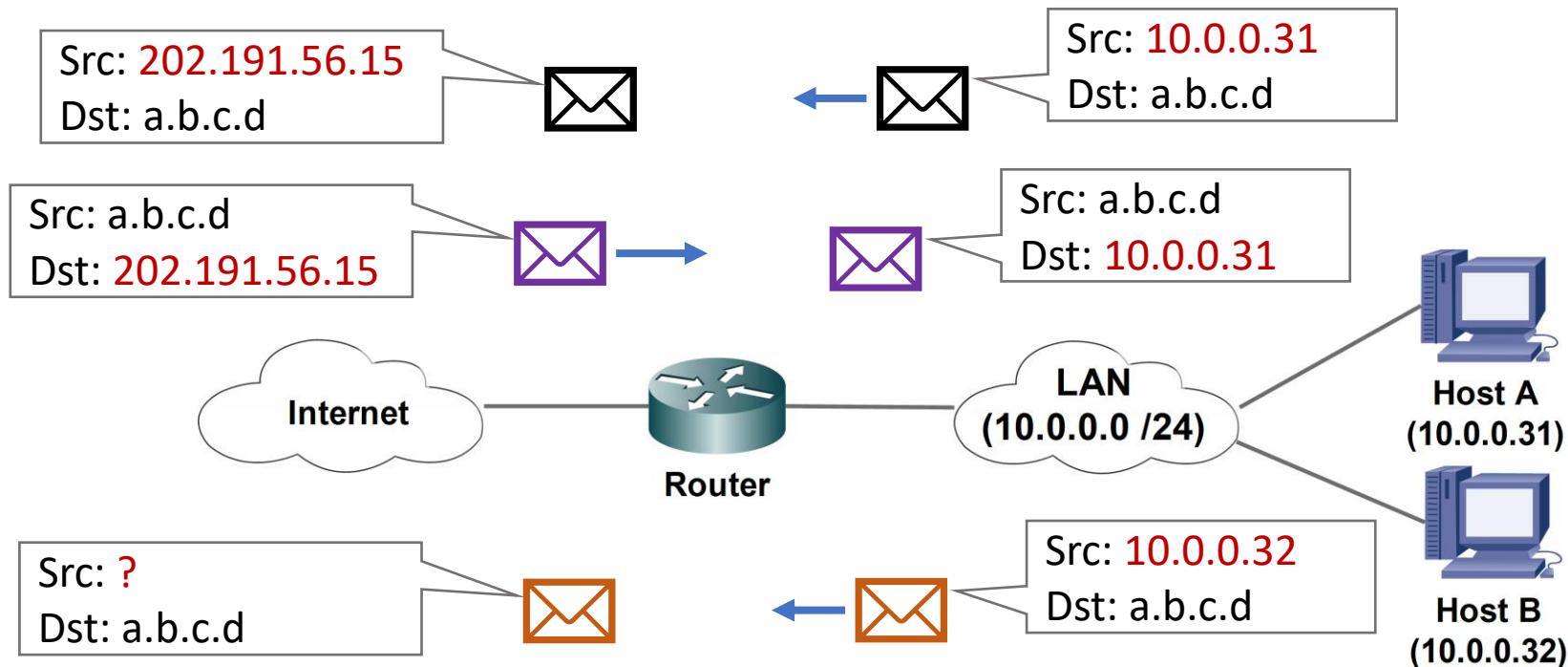
NAT Table

Inside IP	Outside IP
10.0.0.10	202.191.56.11
10.0.0.20	202.191.56.65

Dynamic NAT

- NAT router tự động ánh xạ một dải các địa chỉ IP private với một dải các địa chỉ public để các máy tính có thể đi ra ngoài Internet khi cần thiết
- Trái với static NAT, người quản trị sẽ không cần cấu hình cụ thể 1 địa chỉ IP private được gán với 1 địa chỉ IP public nào
- Bất kỳ địa chỉ IP private nào sẽ được gán và dịch tự động sang 1 trong các địa chỉ IP public đang sẵn có (đang rồi) ở trong pool địa chỉ public bởi NAT router

Dynamic NAT – Ví dụ

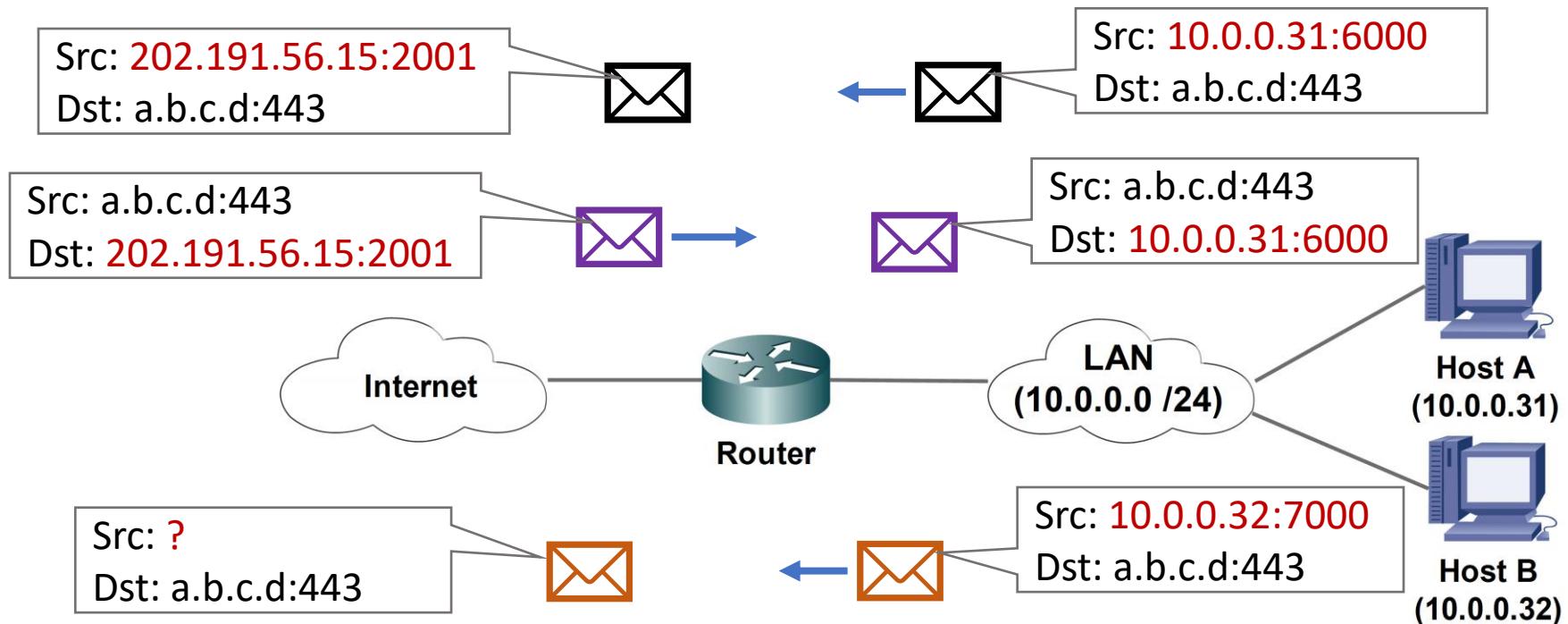


NAT Table	Inside IP	Outside IP
	10.0.0.31	202.191.56.15
	10.0.0.32	202.191.56.16

Port Address Translation

- Còn được biết là NAT **overloading**
- Là một dạng đặc biệt của dynamic NAT để tận dụng hơn nữa không gian địa chỉ IPv4
- Cho phép nhiều địa chỉ IP private ánh xạ với một địa chỉ IP public đã đăng ký kèm các số hiệu cổng khác nhau
- Như vậy PAT sử dụng các sockets duy nhất để ánh xạ với từng máy

Dynamic NAT – Ví dụ



NAT Table

Inside IP	Outside IP
10.0.0.31:6000	202.191.56.15:2001
10.0.0.32:7000	202.191.56.15: 2002



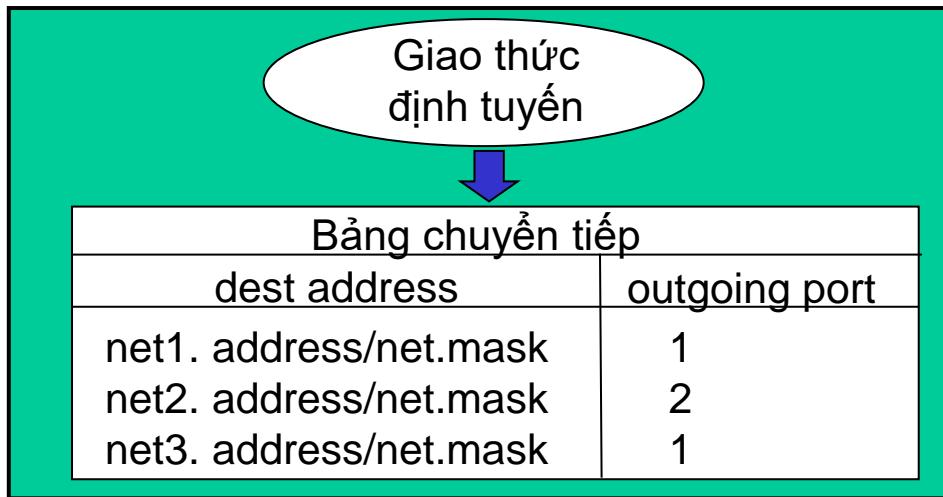
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4.4. Chuyển tiếp gói tin

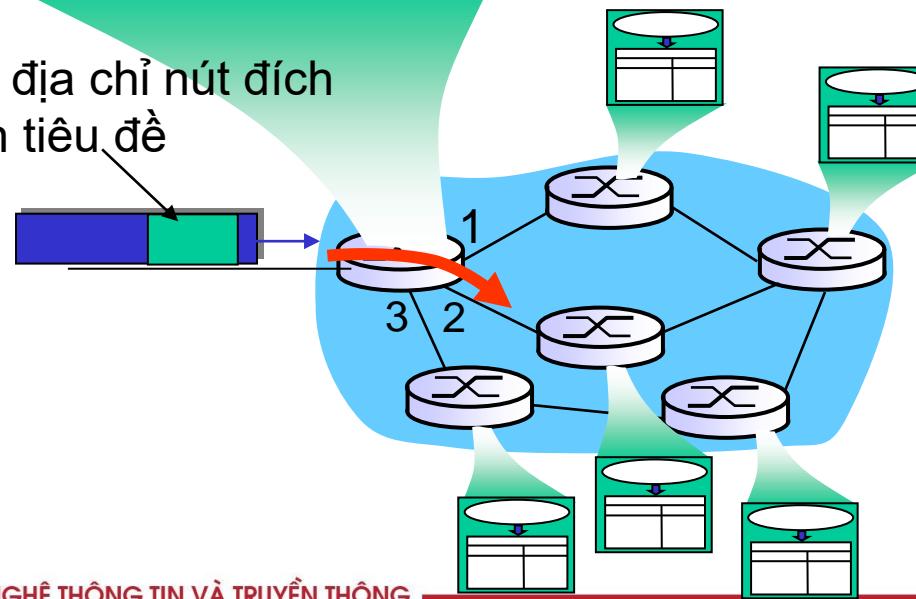
Chuyển tiếp gói tin IP

- Mỗi nút mạng sử dụng bảng chuyển tiếp (Forwarding Table)
 - Là một phần của bảng định tuyến (Routing Table)
- Các thông tin:
 - Đích đến (Destination): Địa chỉ mạng/Mặt nạ (/n)
 - Lối ra mặc định: Sử dụng địa chỉ 0.0.0.0/0 đại diện cho một đích bất kỳ chưa biết
 - Cổng ra (Outgoing port): địa chỉ của cổng ra trên router để chuyển tới nút kế tiếp trong đường đi

Bảng chuyển tiếp



Gói tin với địa chỉ nút đích
trong phần tiêu đề



Ví dụ - Bảng chuyển tiếp trên máy trạm

C:\Documents and Settings\vantv>netstat -r

Route Table

```
=====
Interface List
0x1 ... Destination ....MS TCP Loopback interface
0x2 ... 08 00 11 b2 a1 a3 ..... Outgoing port Realtek RTL8159 Family PCI Fast Ethernet NIC -
=====
```

Network	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.34	20
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.1.0	255.255.255.0	192.168.1.34	192.168.1.34	20
192.168.1.34	255.255.255.255	127.0.0.1	127.0.0.1	20
192.168.1.255	255.255.255.255	192.168.1.34	192.168.1.34	20
224.0.0.0	240.0.0.0	192.168.1.34	192.168.1.34	20
255.255.255.255	255.255.255.255	192.168.1.34	192.168.1.34	1

Default Gateway: 192.168.1.1

Ví dụ - Bảng chuyển tiếp trên router (rút gọn)

Router# **show ip route**

Destination

Outgoing port

- O 203.238.37.0/24 via 203.178.136.14, FastEthernet0/1
- O 203.238.37.96/27 via 203.178.136.26, Serial0/0/0
- C 203.238.37.128/27 is directly connected, Serial0/0/0
- O 192.68.132.0/24 via 203.178.136.14, FastEthernet0/1
- C 203.254.52.0/24 is directly connected, FastEthernet0/1
- C 202.171.96.0/24 is directly connected, Serial0/0/1

Nguyên tắc tìm kiếm

- So sánh n bit đầu tiên trên địa chỉ đích gói tin với các bit tương ứng trên địa chỉ mạng đích
 - /n: Mặt nạ mạng đích
 - Nếu có mạng đích khớp chuyển ra cổng tương ứng
 - Nếu không có mạng đích nào khớp, chuyển ra cổng mặc định (nếu có)
- Quy tắc “longest matching”: nếu có nhiều mạng đích thỏa mãn, chuyển tiếp tới mạng đích có mặt nạ lớn nhất

Địa chỉ đích của gói tin:
11.1.2.10

Destination	Outgoing Port
11.0.0.0 /8	Se0/1
11.1.0.0 /16	Se0/2
11.1.2.0/24	Se0/3

Quy tắc “Longest matching”

Địa chỉ đích:

11.1.2.5 = 00001011.00000001.00000010.00000101

11.1.3.6 = 00001011.00000001.00000011.00000110

Đường đi 1:

11.1.2.0/24 = 00001011.00000001.00000010.00000000

Đường đi 2:

11.1.0.0/16 = 00001011.00000001.00000000.00000000

Đường đi 3:

11.0.0.0/8 = 00001011.00000000.00000000.00000000

Tại sao phải cần quy tắc này?

Chuyển tiếp gói tin trên router

- B1 : Nếu TTL = 1(hoặc TTL = 0), hủy gói tin và báo lỗi. Kết thúc.
- B2 : Nếu TTL >1, lấy địa chỉ đích DA của gói tin. Mật nạ hóa địa chỉ đích của gói tin với các mặt nạ của mạng đích
- B3 : So sánh kết quả mặt nạ hóa với địa chỉ mạng đích tương ứng. Nếu có mạng đích khớp chuyển tới cổng ra tương ứng, giảm TTL.
- B4 : Nếu không có mạng đích khớp, kiểm tra cổng ra mặc định (tương ứng với đích 0.0.0.0 /0)
 - Có cổng mặc định : chuyển gói tin tới cổng mặc định, giảm TTL
 - Không có : hủy gói tin, báo lỗi.



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4.6. Giao thức ICMP

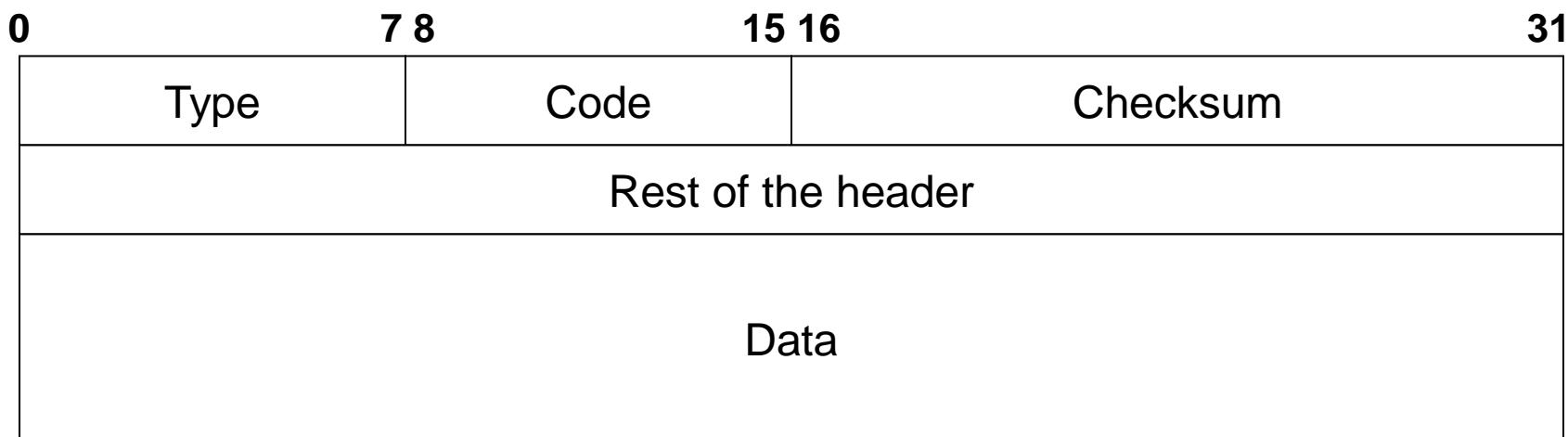
Giới thiệu chung về ICMP

- Internet Control Message Protocol
- IP là giao thức không tin cậy, hướng không liên kết
 - Thiếu các cơ chế hỗ trợ và kiểm soát lỗi → Phía gửi không biết gói tin có được truyền thành công
- ICMP được sử dụng ở tầng mạng để trao đổi thông tin:
2 chức năng chính
 - Báo lỗi khi chuyển tiếp một gói tin IP không thành công
 - Truy vấn thông tin trạng thái
- Gói tin ICMP được đóng gói vào trong gói tin IP:



Khuôn dạng gói tin ICMP

- Type: dạng gói tin ICMP
- Code: Nguyên nhân gây lỗi
- Checksum
- Rest of header: Mỗi dạng có phần còn lại tương ứng, phụ thuộc vào Type và Code ()



Một số dạng gói tin ICMP

ICMP Message Type	Error-reporting messages	3	Destination Unreachable
		4	Source quench
		5	Redirection
		11	Time exceeded
		12	Parameter problem
	Query messages	8 or 0	Echo request or reply
		13 or 14	Time stamp request or reply
		17 or 18	Address mask request or reply
		9 or 10	Router advertisement or solicitation

ICMP và các công cụ debug

- ICMP luôn hoạt động song trong suốt với người dùng
- Có thể sử dụng ICMP thông qua các công cụ debug. Ví dụ:
 - ping
 - traceroute

ping

- Tác dụng: Kiểm tra trạng thái kết nối với một nút mạng
- Cú pháp: ping *host_address*
 - *host_address* là địa chỉ IP hoặc tên miền
- Hoạt động
 - Bên gửi: Gửi gói tin ICMP Echo Request (Type = 8)
 - Bên nhận: Trả gói tin ICMP Echo Reply (Type = 0)
- Trường dữ liệu chứa thời gian gửi gói tin
 - Khi nhận được phản hồi thì tính được trễ khứ hồi (RTT - Round trip time)

Ping: Ví dụ

```
C:\Documents and Settings\admin>ping www.yahoo.co.uk
```

Pinging www.euro.yahoo-eu1.akadns.net [217.12.3.11] with 32 bytes of data:

Reply from 217.12.3.11: bytes=32 time=600ms TTL=237

Reply from 217.12.3.11: bytes=32 time=564ms TTL=237

Reply from 217.12.3.11: bytes=32 time=529ms TTL=237

Reply from 217.12.3.11: bytes=32 time=534ms TTL=237

Ping statistics for 217.12.3.11:

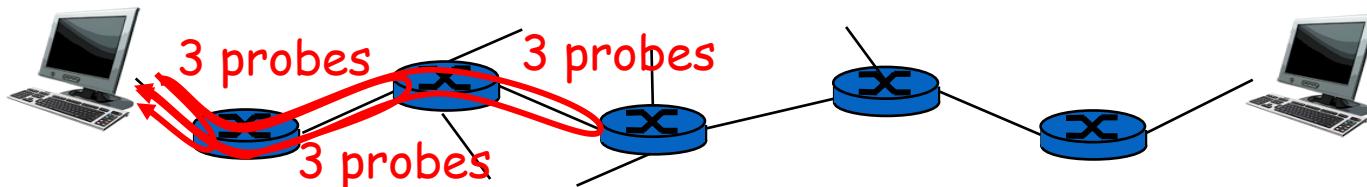
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 529ms, Maximum = 600ms, Average = 556ms

traceroute

- Cú pháp:
 - Windows: tracert *host_address*
 - Linux: traceroute *host_address*
- Bên gửi truyền gói tin UDP cho bên nhận
 - Lượt thứ n có TTL = n
- Khi gói tin lượt gửi thứ n đến router thứ n:
 - Router hủy gói tin
 - Gửi một gói tin ICMP báo lỗi “Time exceeded” (Type = 11)
 - Dữ liệu có chứa tên và địa chỉ IP của router
- Khi nhận được gói tin trả lời, bên gửi sẽ tính ra RTT



traceroute

Điều kiện kết thúc

- Gói tin đến được đích
- Đích trả về gói tin ICMP báo lỗi “Time exceeded”
- Khi nguồn nhận được gói tin ICMP này sẽ dừng lại

Traceroute: Ví dụ

```
C:\Documents and Settings\admin>tracert www.jaist.ac.jp
```

Tracing route to www.jaist.ac.jp [150.65.5.208] over a maximum of 30 hops:

```
1  1 ms  <1 ms  <1 ms  192.168.1.1
2  15 ms  14 ms  13 ms  210.245.0.42
3  13 ms  13 ms  13 ms  210.245.0.97
4  14 ms  13 ms  14 ms  210.245.1.1
5  207 ms  230 ms  94 ms  pos8-2.br01.hkg04.pccwbtn.net [63.218.115.45]
6  *        403 ms  393 ms  0.so-0-1-0.XT1.SCL2.ALTER.NET [152.63.57.50]
7  338 ms  393 ms  370 ms  0.so-7-0-0.XL1.SJC1.ALTER.NET [152.63.55.106]
8  402 ms  404 ms  329 ms  POS1-0.XR1.SJC1.ALTER.NET [152.63.55.113]
9  272 ms  288 ms  310 ms  193.ATM7-0.GW3.SJC1.ALTER.NET [152.63.49.29]
10 395 ms  399 ms  417 ms  ve-4.foundry3.nezu.wide.ad.jp [203.178.138.244]
11 355 ms  356 ms  378 ms  ve-3705.cisco2.komatsu.wide.ad.jp [203.178.136.193]
12 388 ms  398 ms  414 ms  c76.jaist.ac.jp [203.178.138.174]
13 438 ms  377 ms  435 ms  www.jaist.ac.jp [150.65.5.208]
```

Trace complete.



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4.7. Giao thức ARP

Địa chỉ MAC và ARP

- Address Resolution Protocol
- Tìm địa chỉ MAC (định danh tầng liên kết dữ liệu) của một nút mạng khi đã biết địa chỉ IP
- Tại sao cần ARP?
 - Truyền tin trên tầng mạng dùng địa chỉ IP
 - Truyền tin trên tầng liên kết dữ liệu dùng địa chỉ MAC
 - Khi gửi: dữ liệu chuyển từ tầng mạng xuống tầng liên kết dữ liệu.
 - Dữ liệu gửi trong mạng LAN: Máy nguồn cần phải biết địa chỉ MAC của máy đích
 - Dữ liệu gửi ra ngoài mạng LAN: Máy nguồn phải biết địa chỉ MAC của bộ định tuyến mặc định

Hoạt động của ARP

- Mỗi nút trong sử dụng bảng ARP Table:
 - Ánh xạ <Địa chỉ IP, Địa chỉ MAC, TTL>
 - TTL: Thời gian giữ ánh xạ trong bảng
- Khi tìm địa chỉ MAC của một nút khác, nút mạng gửi quảng bá gói tin ARP Request lên trên mạng.
 - Gói tin ARP Request chứa địa chỉ IP của nút cần tìm kiếm
- Nút mạng mang địa chỉ IP được hỏi sẽ gửi ARP Reply trả lời

Hoạt động của ARP – Ví dụ

ARP Request

(broadcast)

Who has 10.0.0.2?



Host A: 10.0.0.1

MAC: 11-29-9C-E8-FF-11



Host B: 10.0.0.2

MAC: 12-29-9C-E8-FF-12



Host C: 10.0.0.3

MAC: 13-29-9C-E8-FF-13

ARP Request

(broadcast)

Who has 10.0.0.2?



Host A: 10.0.0.1

MAC: 11-29-9C-E8-FF-11



Host B: 10.0.0.2

MAC: 12-29-9C-E8-FF-12



Host C: 10.0.0.3

MAC: 13-29-9C-E8-FF-13

ARP Request

(broadcast)

Who has 10.0.0.2?



Host A: 10.0.0.1

MAC: 11-29-9C-E8-FF-11

ARP Reply (unicast)

10.0.0.2 is at 12-29-9C-E8-FF-12



Host B: 10.0.0.2

MAC: 12-29-9C-E8-FF-12



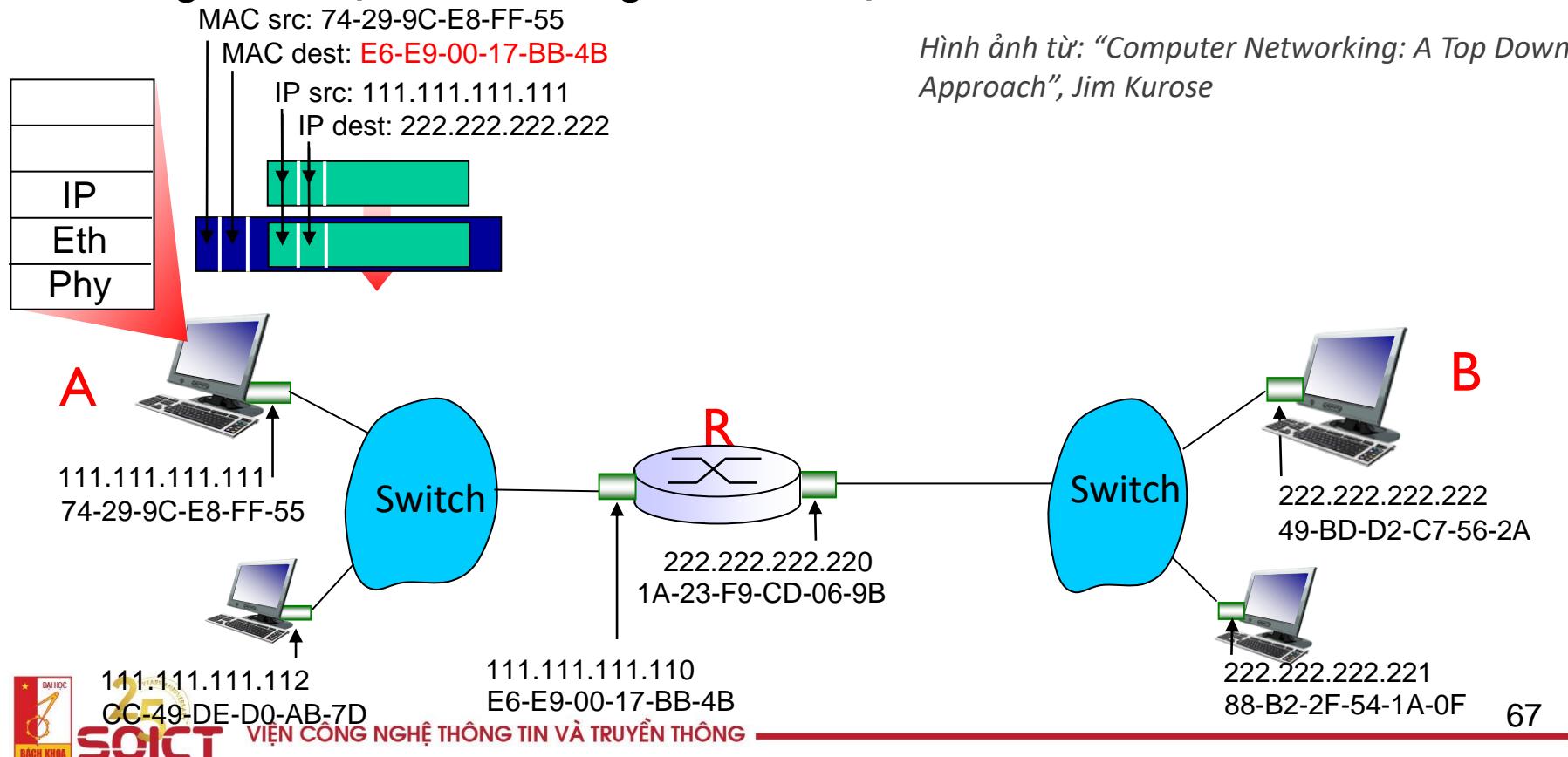
Host C: 10.0.0.3

MAC: 13-29-9C-E8-FF-13

Chuyển tiếp dữ liệu tới LAN khác

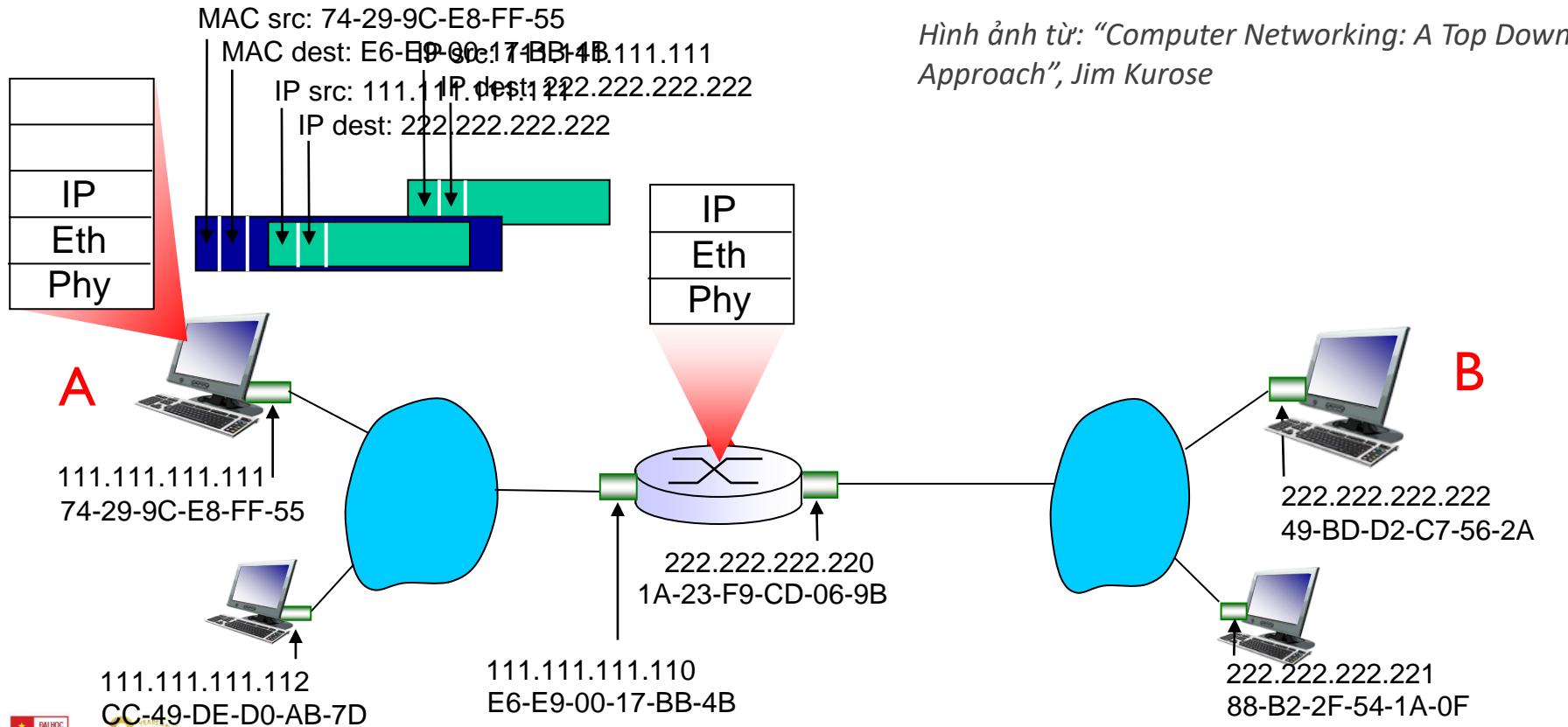
Ví dụ: **Gửi dữ liệu từ A tới B qua router R**

- A soạn một gói tin IP với địa chỉ nguồn là A, địa chỉ đích là B
- Gói tin chuyển xuống tầng liên kết dữ liệu: đóng gói thành khung tin tầng 2 với địa chỉ MAC nguồn là A, địa chỉ MAC đích là R



Chuyển tiếp dữ liệu tới LAN khác

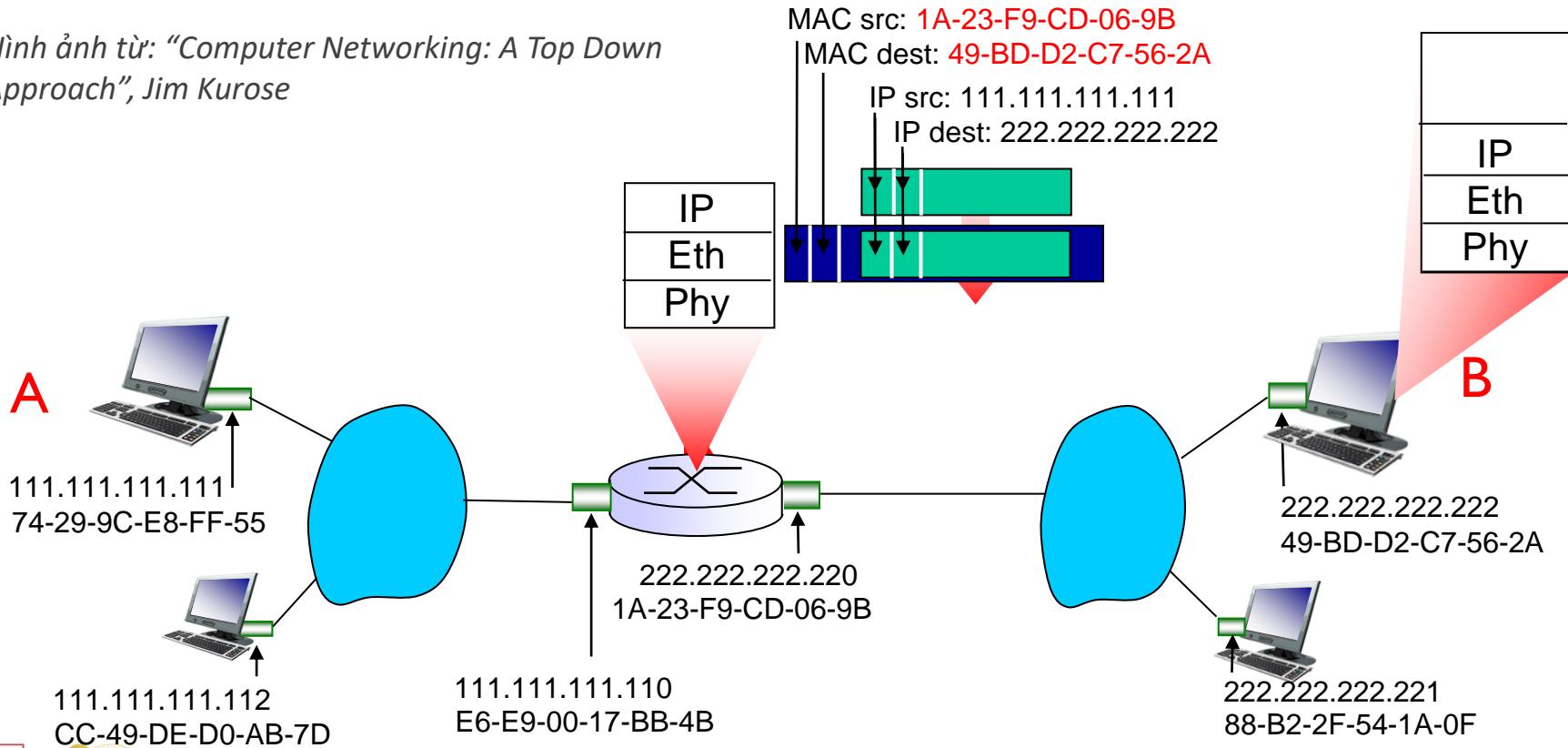
- ❖ Khung tin được chuyển từ A tới R
- ❖ Tại R: khung tin được bóc bỏ header, chuyển lên cho tầng mạng dưới dạng một gói tin IP



Chuyển tiếp dữ liệu tới LAN khác

- ❖ R chuyển tiếp gói tin với địa chỉ IP nguồn là A, IP đích là B
- ❖ Gói tin chuyển xuống tầng liên kết dữ liệu: đóng gói thành khung tin tầng 2 với địa chỉ MAC nguồn là R, địa chỉ MAC đích là B

Hình ảnh từ: "Computer Networking: A Top Down Approach", Jim Kurose





ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4.8. DHCP

Giới thiệu chung

- Dynamic Host Configuration Protocol
- Dịch vụ tầng ứng dụng cung cấp cấu hình địa chỉ IP cho các nút mạng:
 - Địa chỉ IP
 - Mặt nạ mạng
 - Địa chỉ bộ định tuyến mặc định (default router, default gateway)
 - Có thể địa chỉ máy chủ DNS phân giải tên miền
- Hoạt động theo mô hình client/server: client sử dụng các thông số địa chỉ IP do server cấp phát

Các thông điệp DHCP

Client

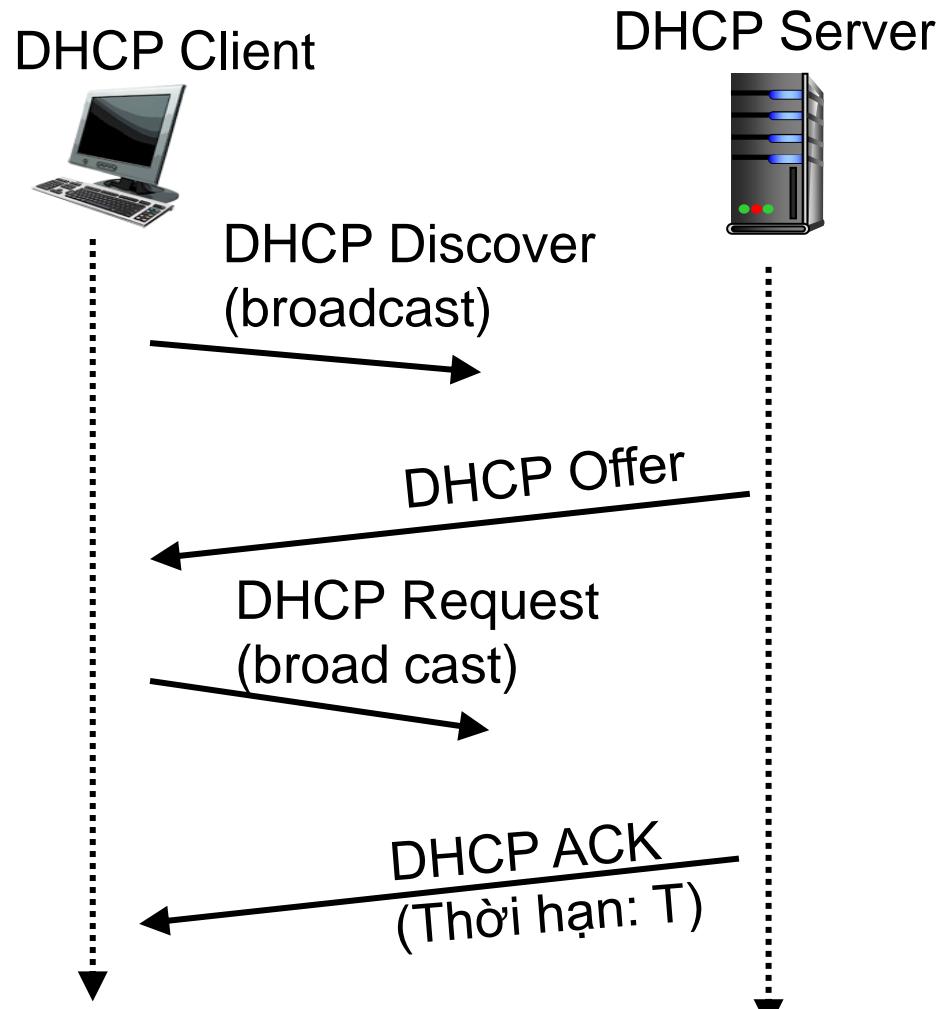
- DHCP Discover: tìm kiếm DHCP Server
- DHCP Request: đăng ký sử dụng địa chỉ IP
- DHCP Release: giải phóng địa chỉ IP đang sử dụng
- DHCP Decline: Từ chối địa chỉ IP được cấp phát

Server

- DHCP Offer: Cung cấp thông số cấu hình địa chỉ IP
- DHCP ACK: Chấp nhận đăng ký
- DHCP NAK: Từ chối đăng ký

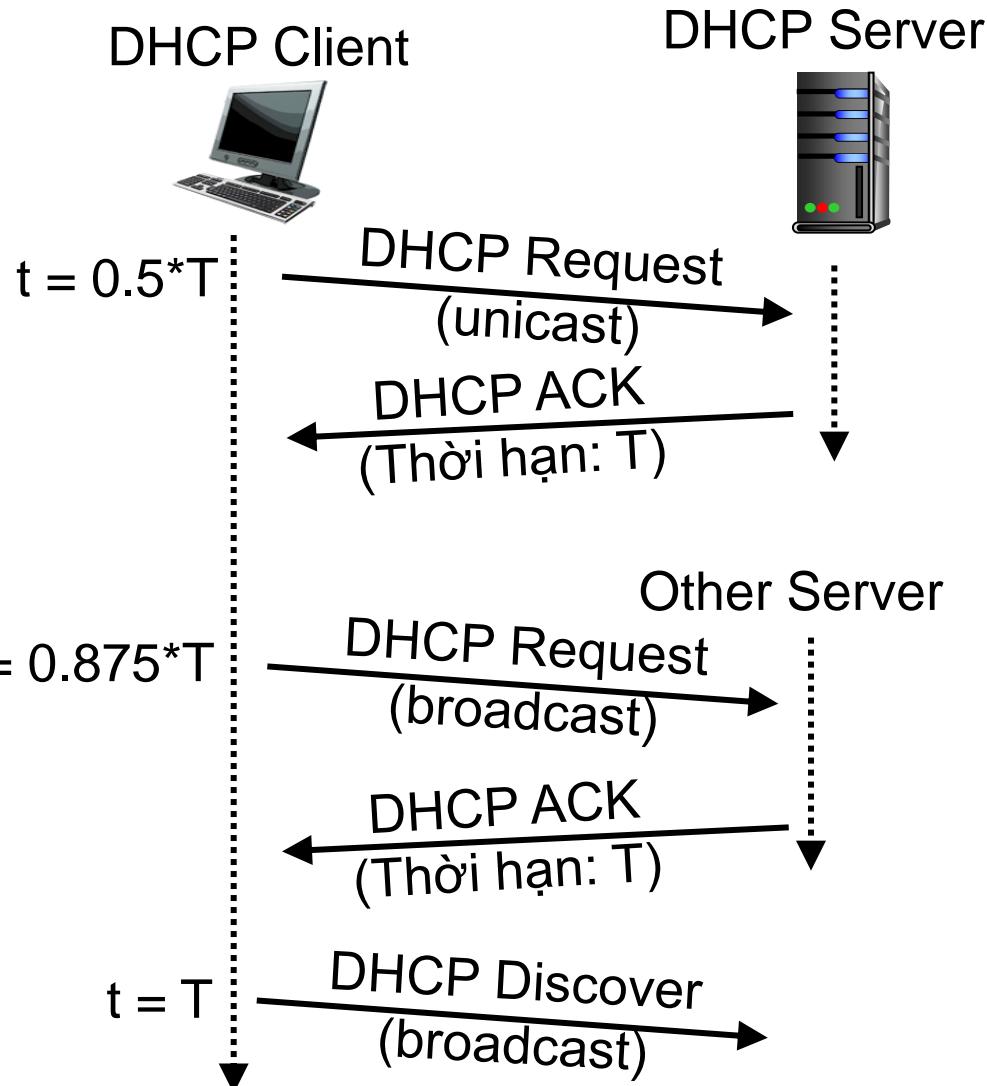
Cấp phát cấu hình mới

- B1: Client gửi thông điệp DHCP Discover quảng bá để tìm kiếm DHCP Server
- B2: Nếu có DHCP Server trong mạng, server gửi thông điệp DHCP Offer chứa các thông số địa chỉ
- B3: Client chọn một cấu hình từ các DHCP Offer nhận được và gửi thông điệp DHCP Request để đăng ký
- B4: DHCP Server gửi thông điệp DHCP ACK chấp nhận



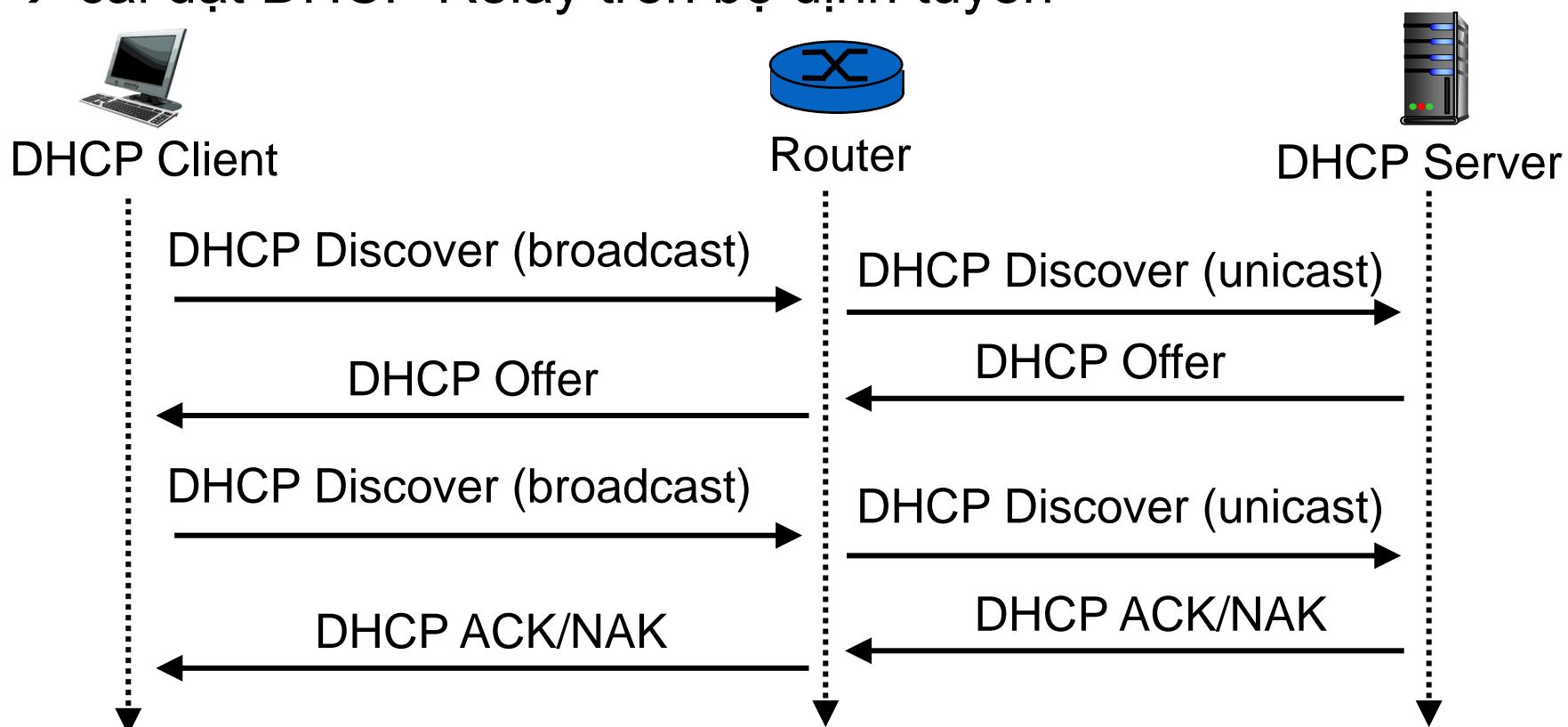
Gia hạn sử dụng

- Mỗi cấu hình được sử dụng trong khoảng thời gian $T \rightarrow$ client cần gia hạn thời gian sử dụng
- Khi $t = 0.5*T$, client gửi DHCP Request tới DHCP Server để yêu cầu gia hạn
- Nếu không có DHCP ACK, khi $t = 0.875*T$, client gửi DHCP Request quảng bá
- Nếu không có DHCP ACK, khi $t = T$, client gửi DHCP Discover



DHCP Relay

- DHCP Server nằm trên một mạng khác với client → các gói tin quảng bá không được bộ định tuyến chuyển tiếp
→ cài đặt DHCP Relay trên bộ định tuyến





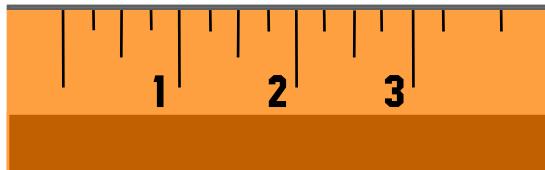
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

4.8. Giới thiệu về IPv6

Giao thức IPv6

- Xuất phát từ nhu cầu thực tế: địa chỉ IPv4 cạn kiệt, không đủ để cấp phát
- Cải tiến trên IPv6:
 - Mở rộng không gian địa chỉ
 - Sử dụng địa chỉ có độ dài 128 bit
 - Phân vùng địa chỉ(scope)

Không gian địa chỉ IPv4



1mm

Không gian địa chỉ IPv6



84.000 lần đường kính của thiên hà

Cải tiến trên IPv6(tiếp)

- Tăng tốc độ xử lý
 - Khuôn dạng header đơn giản hơn
 - Không phân mảnh gói tin
- Hỗ trợ QoS tốt hơn
- Về an toàn an ninh
 - Sử dụng IPSec như một chuẩn
- Tự động cấu hình
 - Chuẩn hóa cơ chế tự động cấu hình



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Địa chỉ IPv6

Cách thức biểu diễn

- 128 bit, biểu diễn bởi số hệ 16
- Phân cách “:” giữa các nhóm gồm 4 số hexa
 - 3ffe:501:100c:e320:2e0:18ff:fe98:936d
- Bỏ qua chuỗi liên tiếp các số 0
 - 3ffe:501:100c:e320:0:0:0:1 →
3ffe:501:100c:e320:::1
- Sử dụng ký hiệu mặt nạ mạng /n

Cấu trúc địa chỉ IPv6

- 2 phần: Network prefix và Interface ID
- Network prefix: 64 bit
- Host ID: 64 bit
 - Chuẩn EUI-64 (*extended unique identifier*)
 - Trong trường hợp mạng Ethernet, Host ID được xác định từ địa chỉ MAC

Phân vùng địa chỉ

- Địa chỉ toàn cục: Network prefix có
 - Global routing prefix: 48 bit với 3 bit đầu là 001
 - SubnetID: 16 bit
- Địa chỉ liên kết nội bộ: Network prefix có
 - 10 bit đầu là 1111 1110 10
 - 54 bit còn lại mang giá trị 0
- Địa chỉ cục bộ: Network prefix có
 - 10 bit đầu là 1111 1110 11
 - 38 bit kế tiếp mang giá trị 0
 - SubnetID: 16 bit

Các dạng địa chỉ IPv6

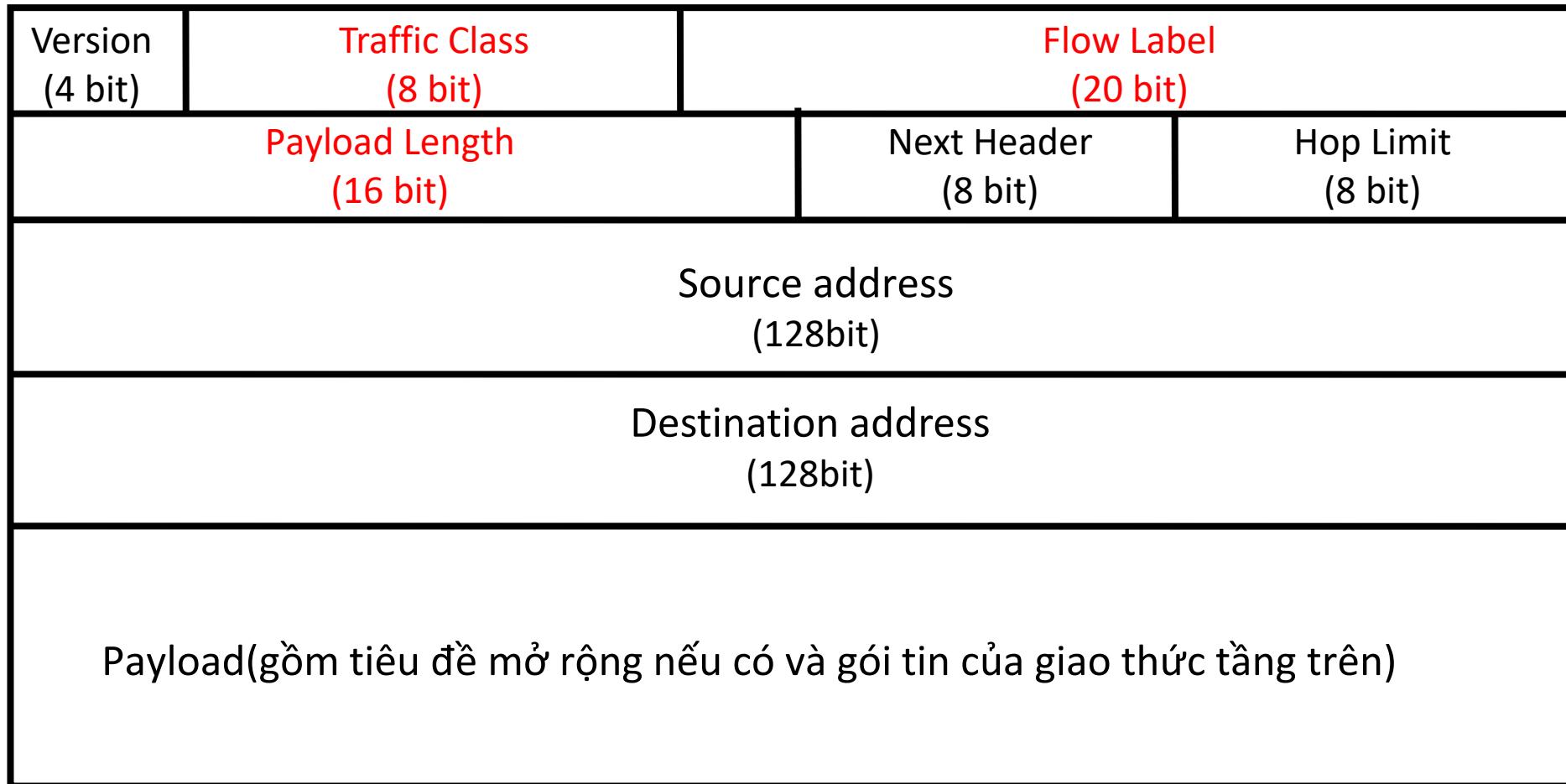
- Địa chỉ Unicast Address: gán cho một cổng giao tiếp mạng
- Địa chỉ Anycast Address: gán cho một nhóm cổng giao tiếp mạng
 - Gói tin gửi tới địa chỉ anycast addr. được chuyển tiếp cho nút gần nhất xác định bởi giao thức định tuyến
- Địa chỉ Multicast Address: gán cho một nhóm cổng giao tiếp mạng trong một scope
 - Bắt đầu bởi 1111 1111
 - Gói tin gửi tới địa chỉ multicast addr. được chuyển tới tất cả các nút trong nhóm



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Khuôn dạng gói tin IPv6

Khuôn dạng gói tin IPv6



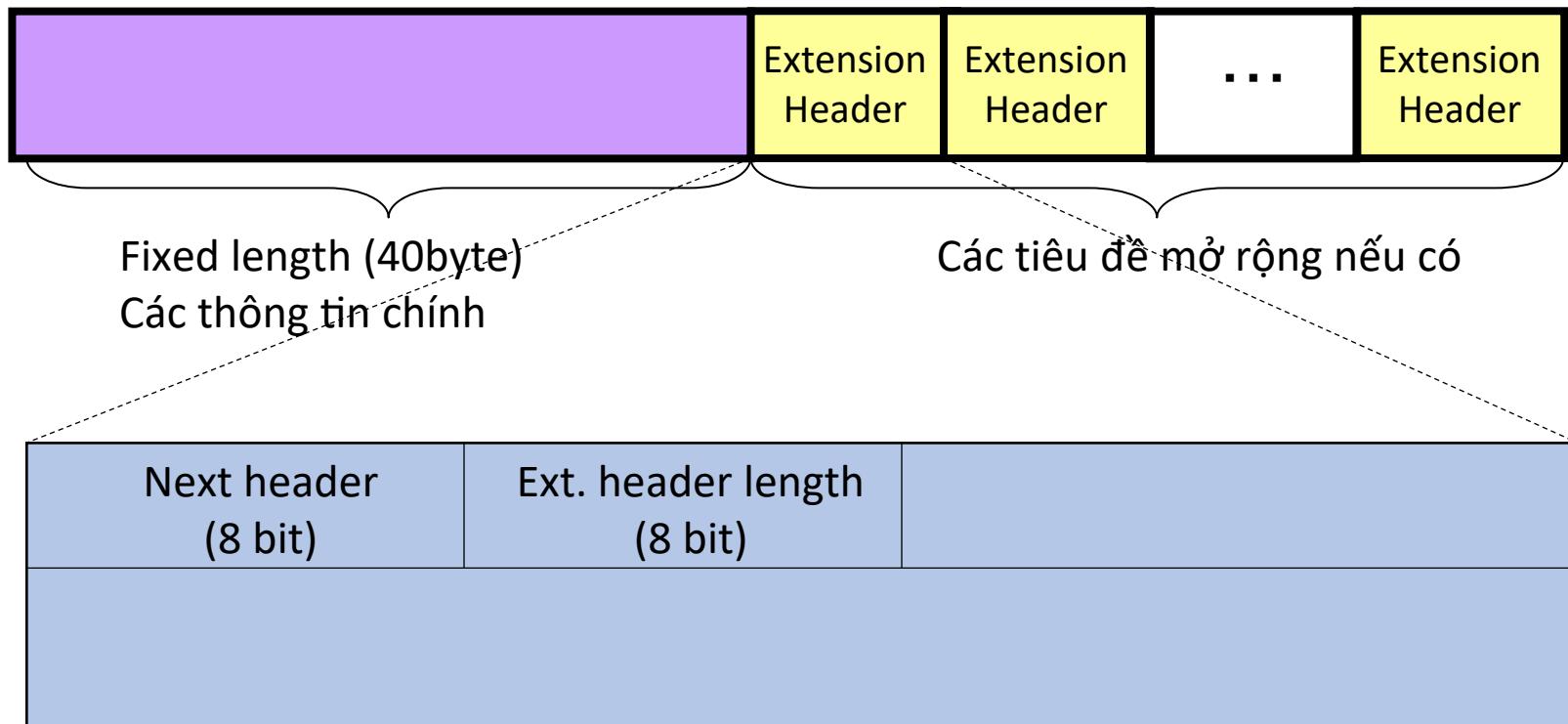
Các trường đổi tên từ IPv4

Tiêu đề gói tin IPv6 (1)

- Version: Phiên bản giao thức(=110)
- Traffic Class: số hiệu giao thức tầng trên
- Flow Label: Điều khiển QoS
- Payload Length: kích thước phần dữ liệu
- Next header: Tiêu đề tiếp theo mở rộng tiếp theo
- Hop limit: tương tự TTL

Tiêu đề gói tin IPv6 (2)

- Trong IPv6, tiêu đề mở rộng đặt trong phần payload



Chọn đường là gì?

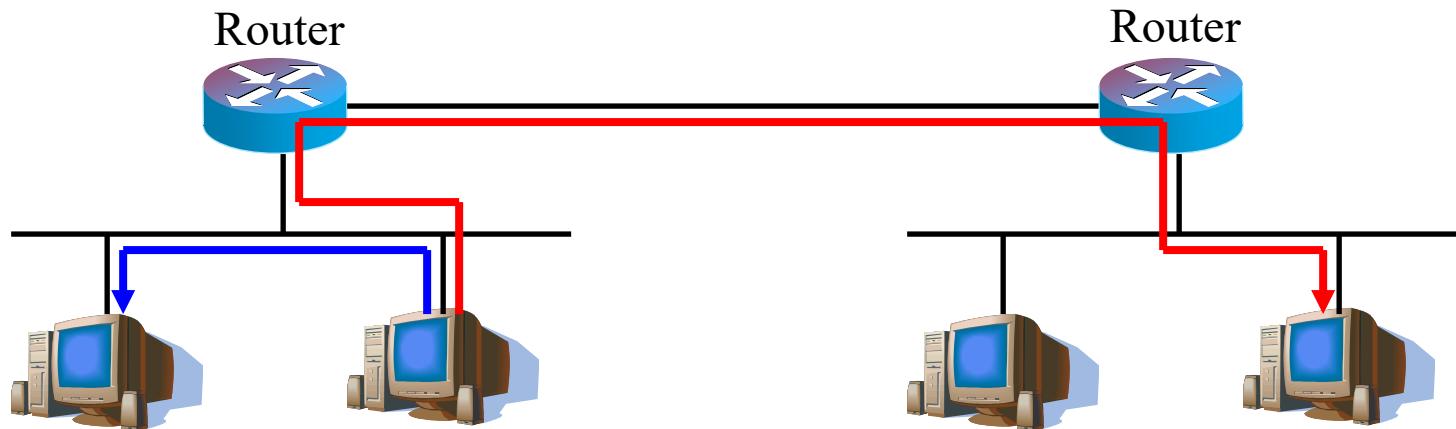
Các nguyên lý chọn đường

Cơ chế chuyển tiếp gói tin

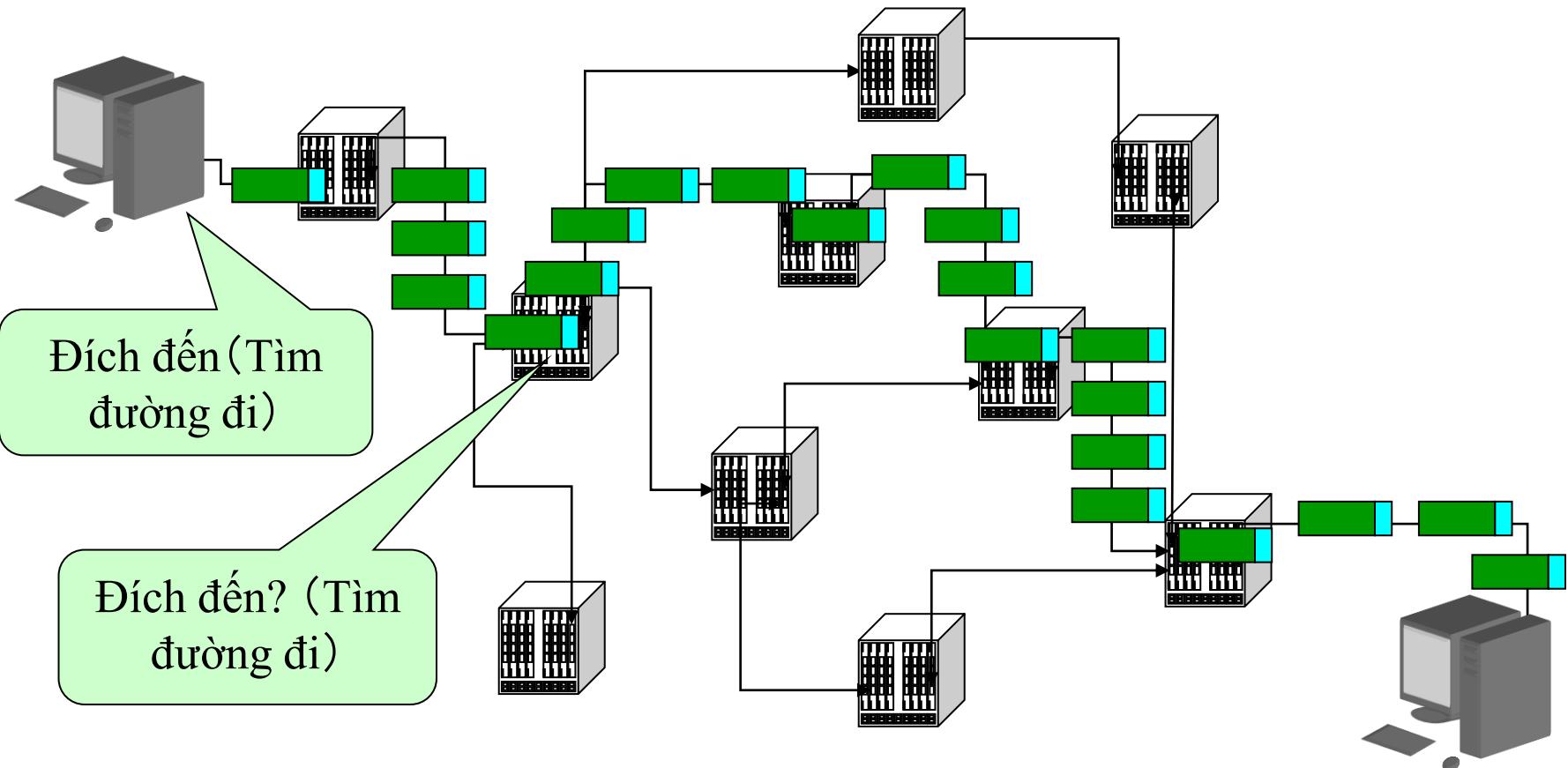
Quy tắc “Longest matching”

Cơ bản về chọn đường (1)

- Khi một máy trạm gửi một gói tin IP tới một máy khác
 - Nếu địa chỉ đích nằm trên cùng một đường truyền vật lý: Chuyển trực tiếp
 - Nếu địa chỉ đích nằm trên một mạng khác: Chuyển gián tiếp qua bộ định tuyến (chọn đường)



Cơ bản về chọn đường (2)



Chọn đường là gì?

- Cơ chế để máy trạm hay bộ định tuyến chuyên tiếp gói tin từ nguồn đến đích
- Các thành phần của chọn đường
 - Bảng chọn đường
 - Thông tin chọn đường
 - Giải thuật, giao thức chọn đường

Bộ định tuyến?

- Thiết bị chuyển tiếp các gói tin giữa các mạng
 - Là một máy tính, với các phần cứng chuyên dụng
 - Kết nối nhiều mạng với nhau
 - Chuyển tiếp gói tin dựa trên bảng chọn đường
- Có nhiều giao diện
- Phù hợp với nhiều dạng lưu lượng và phạm vi của mạng

Một số ví dụ...



BUFFALO
BHR-4RV



PLANEX
GW-AP54SAG



YAMAHA
RTX-1500



Cisco 2600



Cisco CRS-1



Hitachi
GR2000-1B



Juniper M10



Foundry Networks
Netiron 800



Cisco 3700

Router cõi trung

<http://www.cisco.com.vn>

<http://www.juniper.net/>

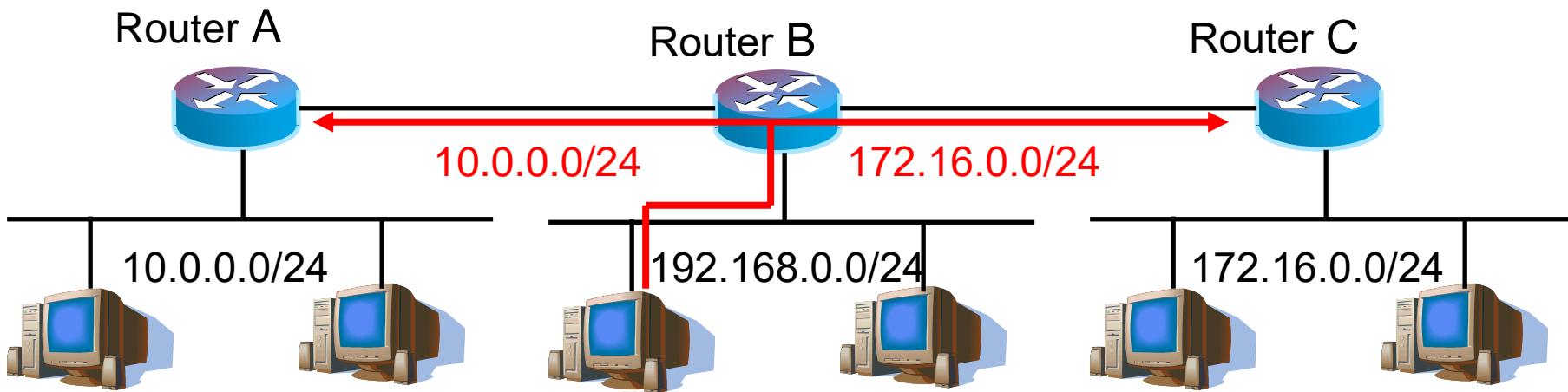
<http://www.buffalotech.com>

Bảng chọn đường

- Chỉ ra danh sách các đường đi có thể, được lưu trong bộ nhớ của router
- Các thành phần chính của bảng chọn đường
 - Địa chỉ đích/mặt nạ mạng
 - Router kế tiếp

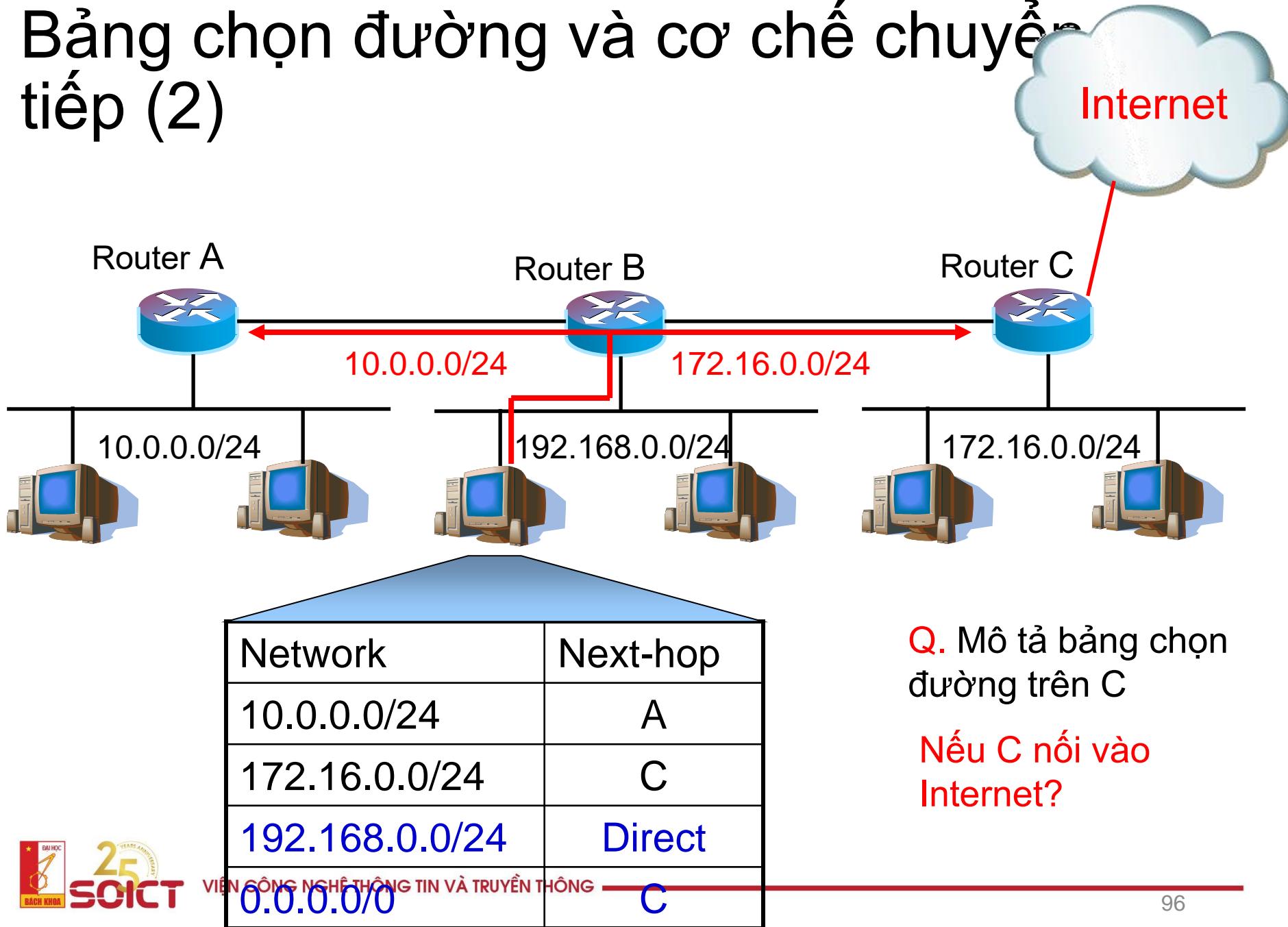
Bảng chọn đường và cơ chế chuyển tiếp (1)

Network	Next-hop
10.0.0.0/24	A
172.16.0.0/24	C
192.168.0.0/24	Direct



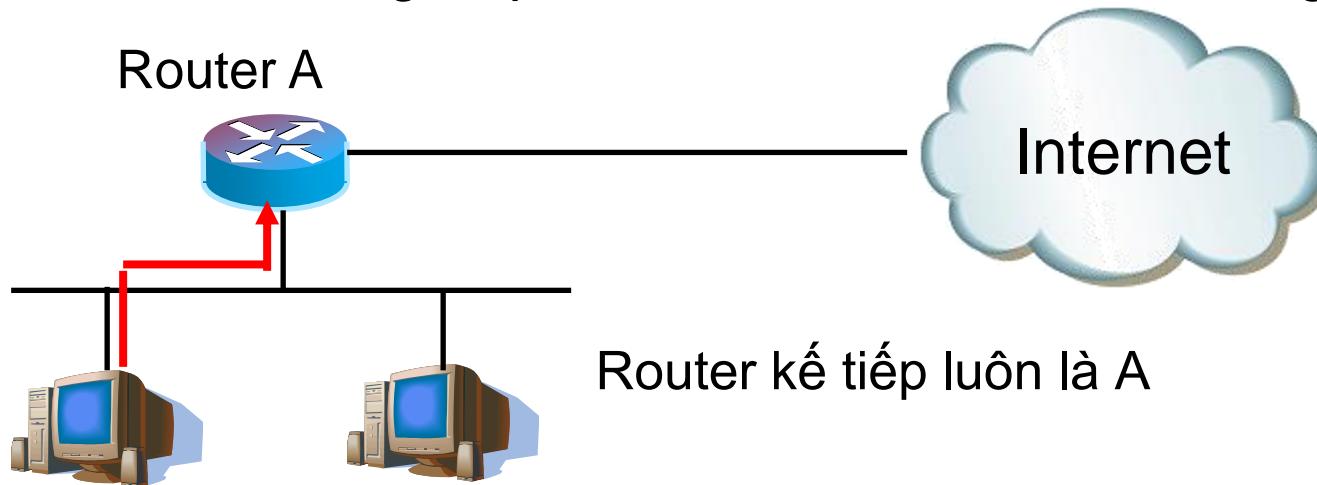
Lưu ý quy tắc: **No routes, no reachability!**

Bảng chọn đường và cơ chế chuyển tiếp (2)



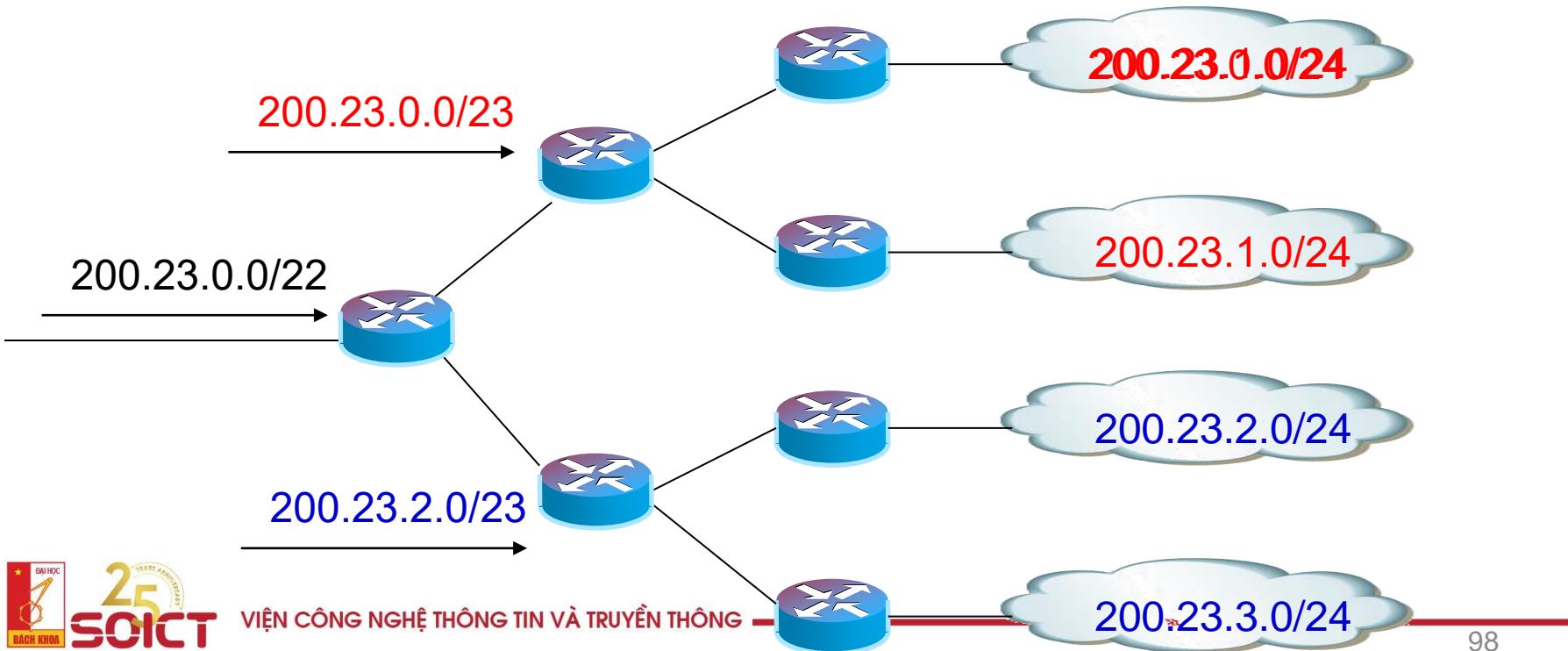
Đường đi mặc định

- Nếu đường đi không tìm thấy trong bảng chọn đường
 - Đường đi mặc định trỏ đến một router kế tiếp
 - Trong nhiều trường hợp, đây là đường đi duy nhất
- 0.0.0.0/0
 - Là một trường hợp đặc biệt, chỉ tất cả các đường đi



Kết hợp đường đi (Routing aggregation)

- Có bao nhiêu mạng con trên mạng Internet?
- Sẽ có rất nhiều mục trong bảng chọn đường?
- Các mạng con kế tiếp với cùng địa chỉ đích có thể được tổng hợp lại để làm giảm số mục trong bảng chọn đường.



Kết hợp đường đi (2)

- Ví dụ về Viettel
 - Không gian địa chỉ IP: khá lớn
 - 203.113.128.0 - 203.113.191.255
 - Để kết nối đến một mạng con của Viettel (khách hàng): Chỉ cần chỉ ra đường đi đến mạng Viettel
- Đường đi mặc định chính là một dạng của việc kết hợp đường
 - 0.0.0.0/0

Ví dụ - Bảng chuyển tiếp trên máy trạm

C:\Documents and Settings\tungbt>netstat -r

Route Table

```
=====
Interface List
0x1 ..... MS TCP Loopback interface
0x2 ...08 00 1f b2 a1 a3 ..... Realtek RTL8139 Family PCI Fast Ethernet NIC -
=====
```

Network	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.34	20
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.1.0	255.255.255.0	192.168.1.34	192.168.1.34	20
192.168.1.34	255.255.255.255	127.0.0.1	127.0.0.1	20
192.168.1.255	255.255.255.255	192.168.1.34	192.168.1.34	20
224.0.0.0	240.0.0.0	192.168.1.34	192.168.1.34	20
255.255.255.255	255.255.255.255	192.168.1.34	192.168.1.34	1

Default Gateway: 192.168.1.1

Ví dụ - Bảng chuyển tiếp trên router (rút gọn)

Router# **show ip route**

- O 203.238.37.0/24 via 203.178.136.14, FastEthernet0/1
- O 203.238.37.96/27 via 203.178.136.26, Serial0/0/0
- C 203.238.37.128/27 is directly connected, Serial0/0/0
- O 192.68.132.0/24 via 203.178.136.14, FastEthernet0/1
- C 203.254.52.0/24 is directly connected, FastEthernet0/1
- C 202.171.96.0/24 is directly connected, Serial0/0/1

Chọn đường tĩnh và chọn đường động

Chọn đường tĩnh

Chọn đường động

Ưu điểm – nhược điểm

Vấn đề cập nhật bảng chọn đường

- Sự thay đổi cấu trúc mạng: thêm mạng mới, một nút mạng bị mất điện
- Sự cần thiết phải cập nhật bảng chọn đường
 - Cho tất cả các nút mạng (về lý thuyết)
 - Thực tế, chỉ một số nút mạng phải cập nhật

Network	Next-hop
192.168.0.0/24	B
172.16.0.0/24	B

172.16.1.0/24

B

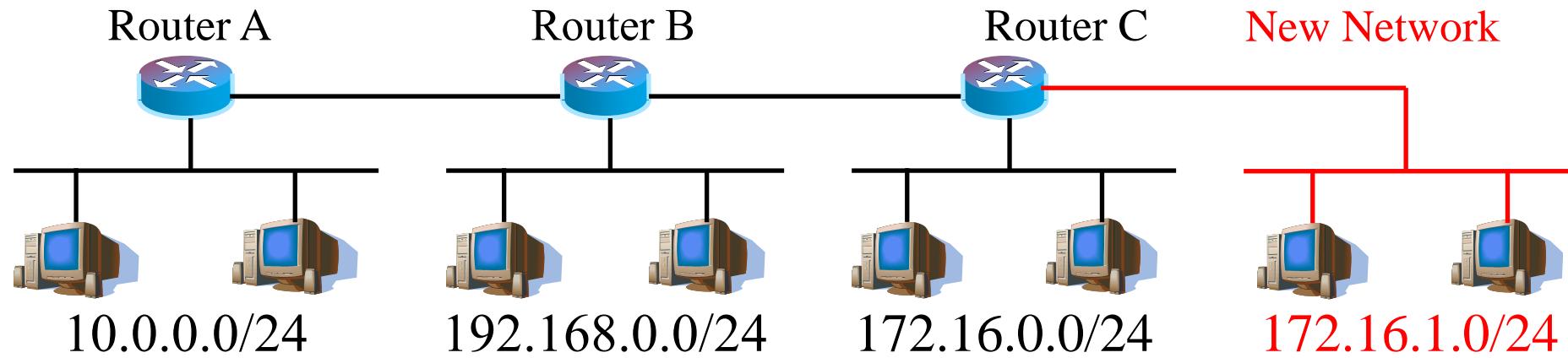
Network	Next-hop
10.0.0.0/24	A
172.16.0.0/24	C

172.16.1.0/24

C

Network	Next-hop
10.0.0.0/24	B
192.168.0.0/24	B

New Network

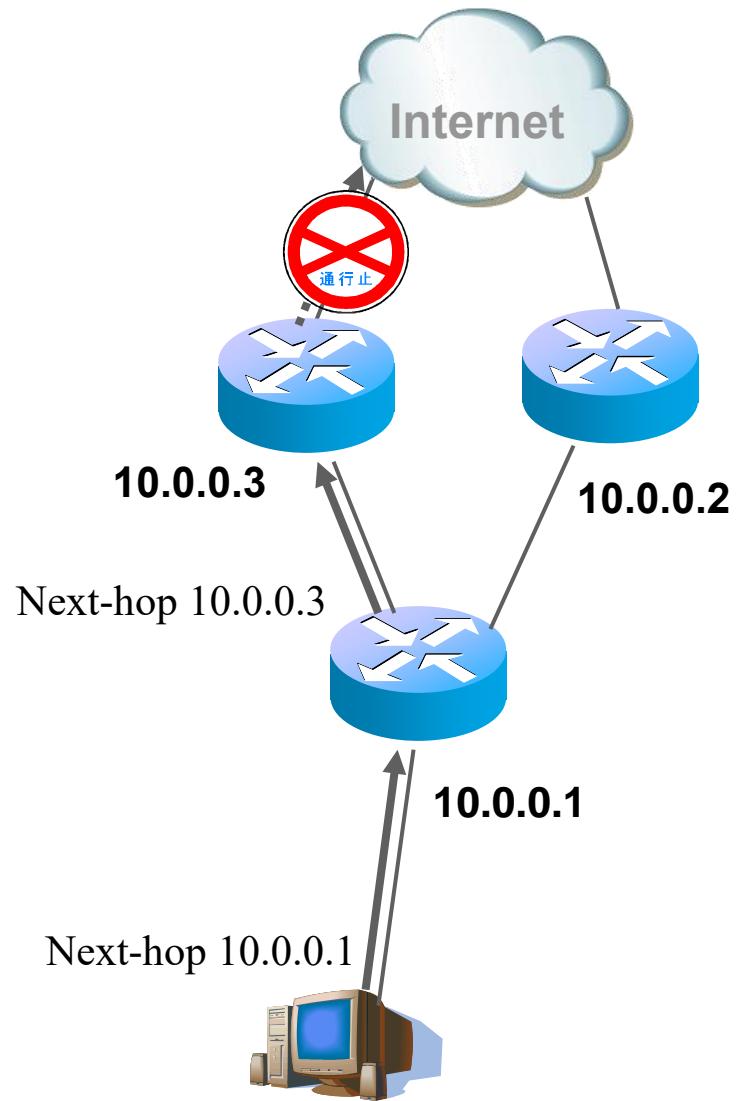


Làm thế nào để cập nhật?

- Chọn đường tĩnh
 - Các mục trong bảng chọn đường được sửa đổi thủ công bởi người quản trị
- Chọn đường động
 - Tự động cập nhật bảng chọn đường
 - Bằng các giao thức chọn đường

Chọn đường tĩnh

- Khi có sự cố:
 - Không thể nối vào Internet kể cả khi có tồn tại đường đi dự phòng
 - Người quản trị mạng cần thay đổi



Bảng chọn đường của 10.0.0.1 (1 phần)

Prefix	Next-hop
0.0.0.0/0	10.0.0.3

Kết nối bị lỗi

Chọn đường động

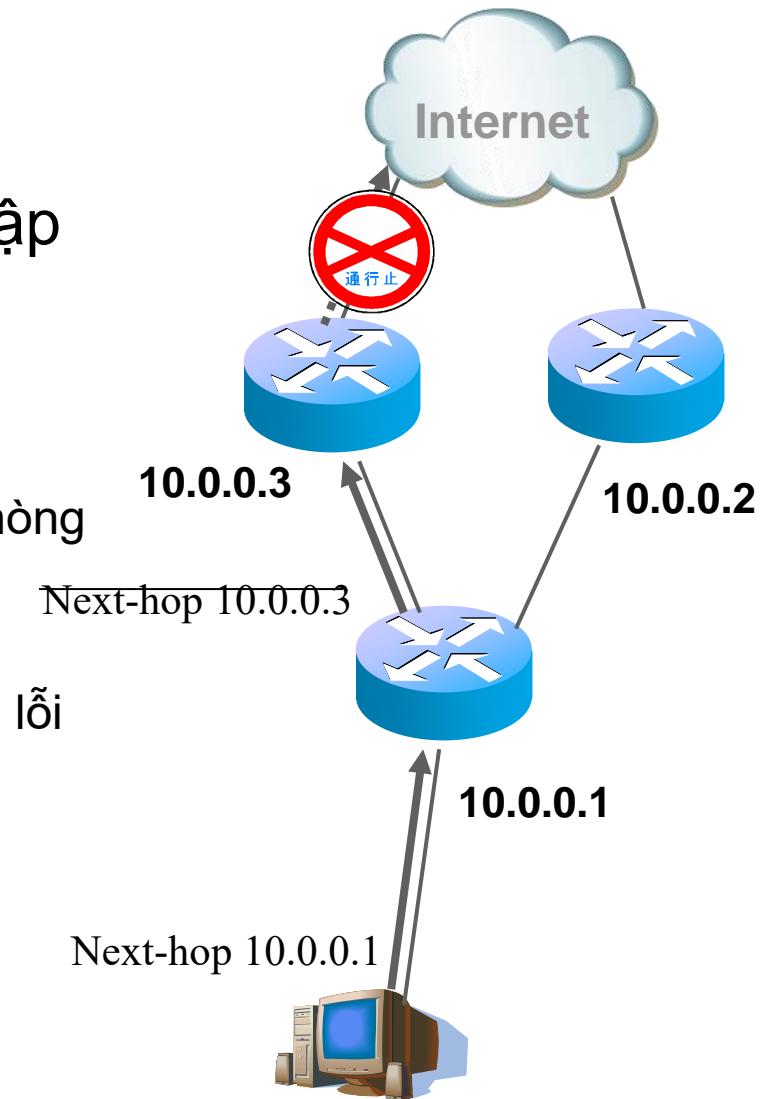
- Khi có sự cố:
 - Đường đi thay thế được cập nhật một cách tự động

Bảng chọn đường của 10.0.0.1 (1 phần)

Prefix	Next-hop
0.0.0.0/0	10.0.0.2
0.0.0.0/0	10.0.0.3

Kết nối dự phòng

Kết nối bị lỗi



Đặc điểm của chọn đường tĩnh

- **Ưu điểm**
 - Ôn định
 - An toàn
 - Không bị ảnh hưởng bởi các yếu tố tác động
- **Nhược điểm**
 - Cứng nhắc
 - Không thể sử dụng tự động kết nối dự phòng
 - Khó quản lý

Chọn đường động

- **Ưu**
 - Dễ quản lý
 - Tự động sử dụng kết nối dự phòng
- **Nhược**
 - Tính an toàn
 - Các giao thức chọn đường phức tạp và khó hiểu
 - Khó quản lý

Các giải thuật và giao thức chọn đường

Giải thuật Dijkstra và Bellman-Ford

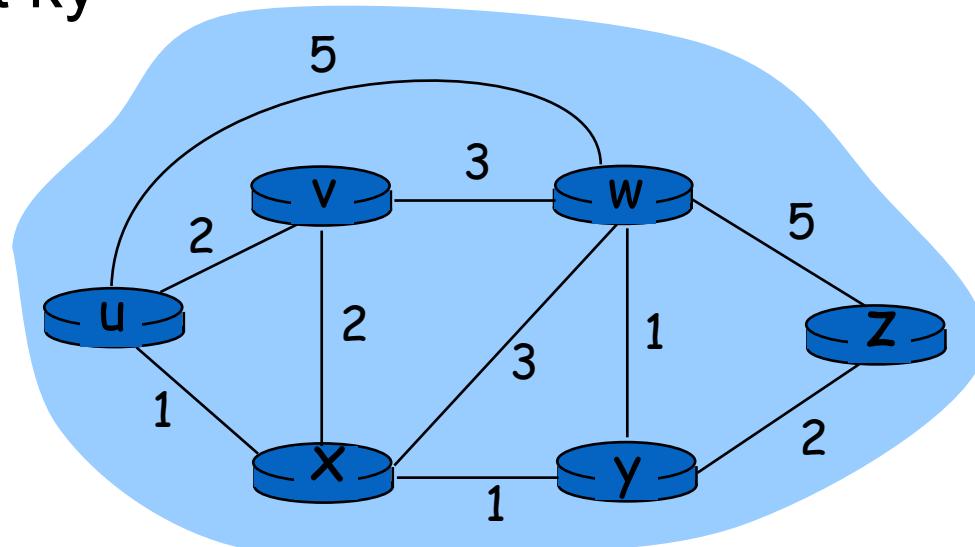
Giao thức dạng link-state và dạng
distance-vector

Vai trò của giải thuật và giao thức chọn đường

- Cần thiết phải xây dựng bảng chọn đường một cách tự động tại các router
- Giải thuật chọn đường thực hiện tính toán đường đi
- Trong hoặc trước quá trình tính toán cần thu thập topo hoặc trao đổi thông tin giữa các nút mạng
 - → giao thức chọn đường thực hiện việc trao đổi này.
 - Giao thức chọn đường còn quy định các quy trình duy trì bảng chọn đường.

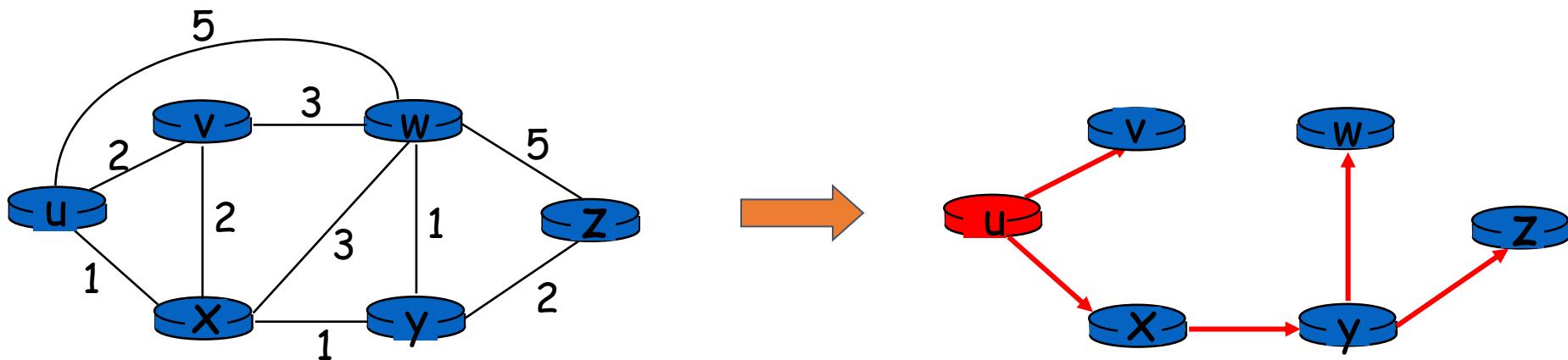
Biểu diễn mạng bởi đồ thị

- Đồ thị với các nút (bộ định tuyến) và các cạnh (liên kết)
- Chi phí cho việc sử dụng mỗi liên kết $c(x,y)$
 - Băng thông, độ trễ, chi phí, mức độ tắc nghẽn...
- Giả thuật chọn đường: Xác định đường đi ngắn nhất giữa hai nút bất kỳ



Cây đường đi ngắn nhất - SPT

- SPT – Shortest Path Tree
- Các cạnh xuất phát từ nút gốc và tới các lá
- Đường đi duy nhất từ nút gốc tới nút v, là đường đi ngắn nhất giữa nút gốc và nút v
- Mỗi nút sẽ có một SPT của riêng nút đó



Tập trung hay phân tán

- Thông tin chọn đường là cần thiết để xây dựng bảng chọn đường
- Tập trung hay phân tán?
 - Tập trung:
 - Mỗi router có thông tin đầy đủ về trạng thái của mạng
 - Giao thức dạng “link state”
 - Phân tán:
 - Các nút chỉ biết được trạng thái của liên kết vật lý tới nút kế bên
 - Liên tục lặp lại việc tính toán và trao đổi thông tin với nút kế bên
 - Giao thức dạng “distance vector”
 - “Bạn của bạn cũng là bạn”

Giải thuật Dijkstra

Ý tưởng

- Mỗi nút đều có sơ đồ và chi phí mỗi link kề với nó
 - Quảng bá “Link-state”
 - Mỗi nút có cùng thông tin
- Tìm đường đi chi phí nhỏ nhất từ một nút ('nguồn') tới tất cả các nút khác
 - dùng để xây dựng bảng chọn đường
- Còn được gọi là giải thuật dạng link-state.

Giải thuật Dijkstra: Pseudo code

Ký hiệu

- $G = (V, E)$: Đồ thị với tập đỉnh V và tập cạnh E
- $c(x, y)$: chi phí của liên kết x tới y ; $= \infty$ nếu không phải 2 nút kế nhau
- $d(v)$: chi phí hiện thời của đường đi từ nút nguồn tới nút đích. v
- $p(v)$: nút ngay trước nút v trên đường đi từ nguồn tới đích
- T : Tập các nút mà đường đi ngắn nhất đã được xác định

Giải thuật Dijkstra: Pseudo code (1)

- **Init()**:

Với mỗi nút v , $d[v] = \infty$, $p[v] = \text{NIL}$

$$d[s] = 0$$

- **Improve(u, v)**, trong đó (u, v) u, v là một cạnh nào đó của G

if $d[v] > d[u] + c(u, v)$ then

$$d[v] = d[u] + c(u, v)$$

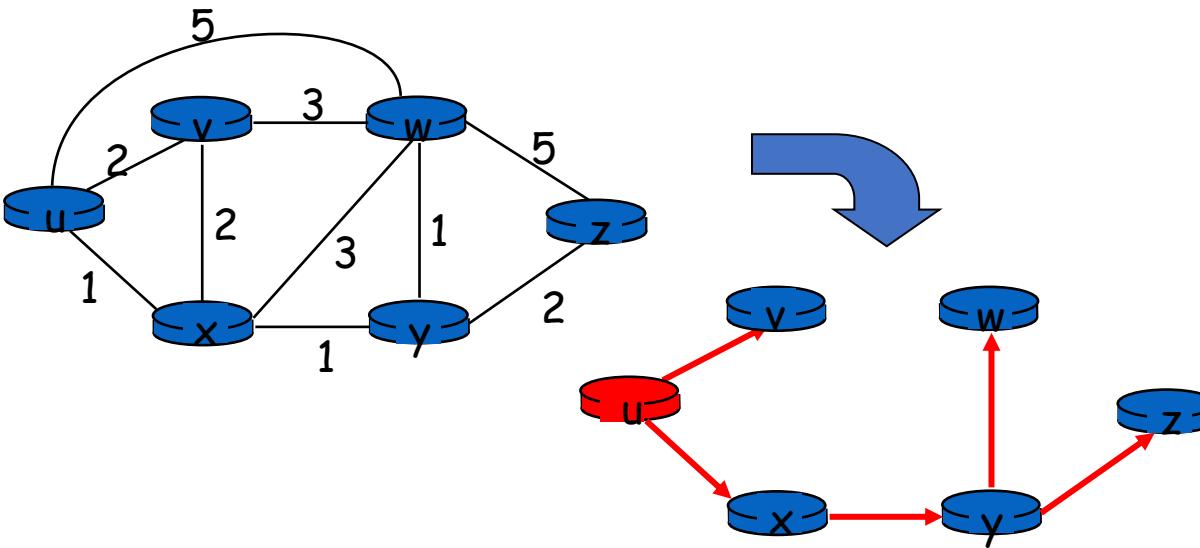
$$p[v] = u$$

Giải thuật Dijkstra: Pseudo code (2)

1. **Init()** ;
2. $T = \emptyset$;
3. **Repeat**
4. $u: u \notin T \mid d(u)$ là bé nhất ;
5. $T = T \cup \{u\}$;
6. for all $v \in \text{neighbor}(u)$ và $v \notin T$
7. **update**(u, v) ;
8. **Until** $T = V$

Giải thuật Dijkstra: Ví dụ

Step	T	$d(v), p(v)$	$d(w), p(w)$	$d(x), p(x)$	$d(y), p(y)$	$d(z), p(z)$
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



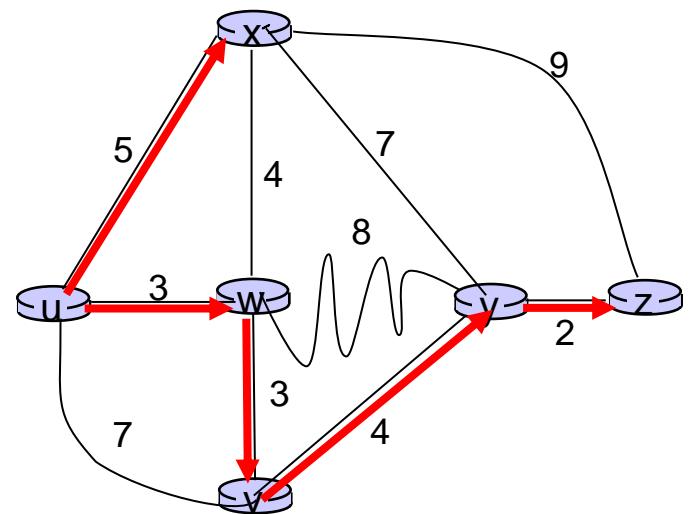
Bảng chọn đường của u:

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

SPT của u:

Giải thuật Dijkstra: Ví dụ 2

Step	N'	$D(v), p(v)$	$D(w), p(w)$	$D(x), p(x)$	$D(y), p(y)$	$D(z), p(z)$
0	u	$7, u$	$3, u$	$5, u$	∞	∞
1	uw	$6, w$	$5, u$	$11, w$	∞	
2	uwx	$6, w$		$11, w$	$14, x$	
3	uwvx			$10, v$	$14, x$	
4	uwxvy				$12, y$	
5	uwxvyz					



Chú ý:

- Tạo thành cây đường đi ngắn nhất bằng cách quay ngược nút trước đó
- Có thể có đường đi bằng nhau (lựa chọn ngẫu nhiên)

Giải thuật Bellman-ford

- Mỗi nút lưu trữ một vector khoảng cách tới mọi đích → distance vector
- Tính toán đường đi dựa trên việc thử đi qua hàng xóm → giải thuật Distance-vector.

Phương trình Bellman-Ford (quy hoạch động)

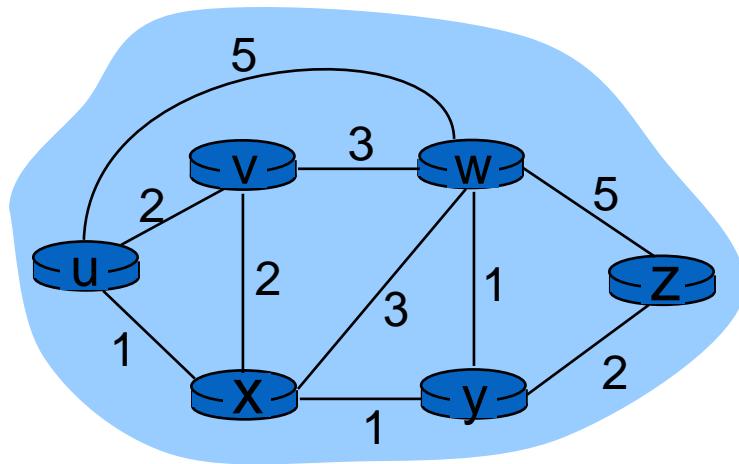
$d_x(y) :=$ chi phí của đường đi ngắn nhất từ x tới y

Ta có
$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

cho tất cả các v là hàng xóm của x

Giải thuật Bellman-ford: Ví dụ

Dễ thấy, $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$



B-F eq. cho ta biết:

$$\begin{aligned}d_u(z) &= \min \{ c(u,v) + d_v(z), \\&\quad c(u,x) + d_x(z), \\&\quad c(u,w) + d_w(z) \} \\&= \min \{ 2 + 5, \\&\quad 1 + 3, \\&\quad 5 + 3 \} = 4\end{aligned}$$

Nút nào làm giá trị trên nhỏ nhất → Lựa chọn là nút kế tiếp trong bảng chọn đường

Giải thuật Bellman-ford

Ý tưởng cơ bản:

- DV: Vector khoảng cách, tạm coi là đường đi ngắn nhất của từ một nút tới những nút khác
- Mỗi nút định kỳ gửi DV của nó tới các nút bên cạnh
- Khi nút x nhận được 1 DV, nó sẽ cập nhật DV của nó qua pt Bellman-ford
- Với một số điều kiện, ước lượng $D_x(y)$ sẽ *hội tụ dần đến giá trị nhỏ nhất* $d_x(y)$

Mỗi nút:

Chờ (Thay đổi trong DV của nút bên cạnh)

Tính lại ước lượng DV

Nếu DV thay đổi, *Báo* cho nút bên cạnh

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

nút x

chi phí tới

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

nút y

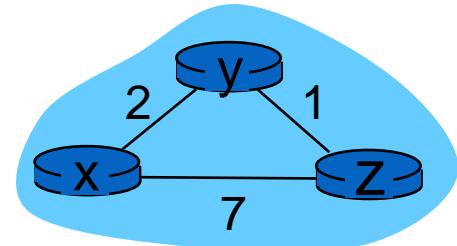
chi phí tới

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

nút z

chi phí tới

	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



thời gian

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(z)\}$$

$$= \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

nút x

chi phí tới

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

từ

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

từ

chi phí tới

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

từ

chi phí tới

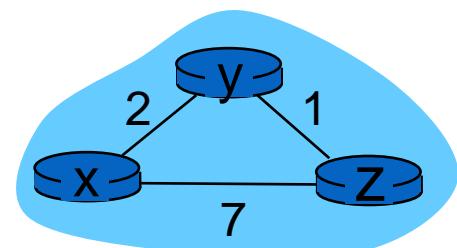
	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

từ

chi phí tới

	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

từ



$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(z)\}$$

$$= \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

nút x

chi phí tới

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

từ

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

từ

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

nút y

chi phí tới

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

từ

chi phí tới

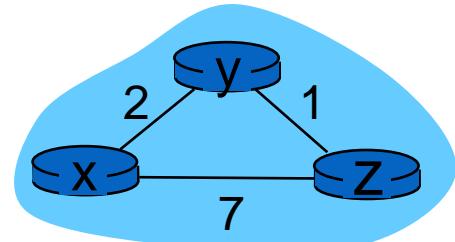
	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

từ

chi phí tới

	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

từ



time

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(z)\}$$

$$= \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

nút x

chi phí tới

	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

từ

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

nút y

chi phí tới

	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

từ

chi phí tới

	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

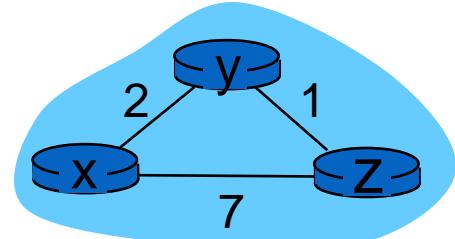
nút z

chi phí tới

	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

chi phí tới

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



thời gian

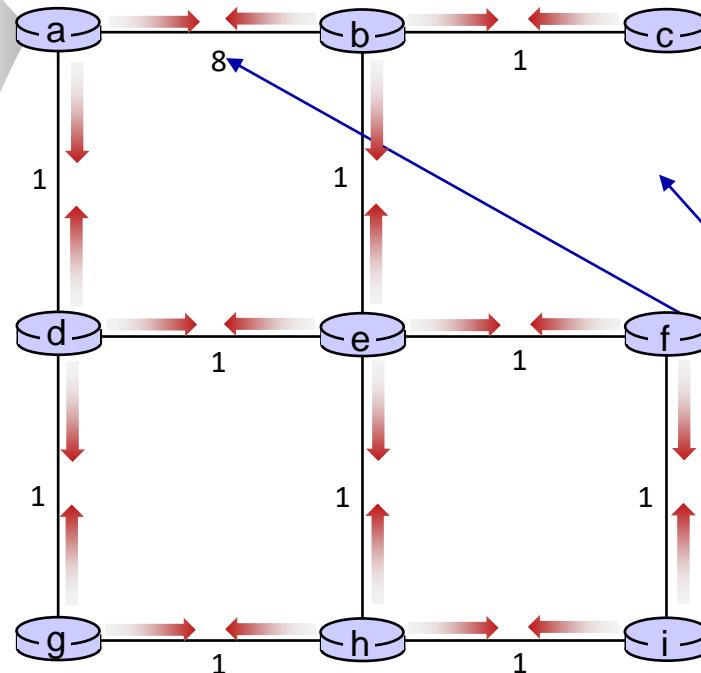
Distance vector: ví dụ



$t=0$

DV in :
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$

- Mọi nút chỉ có khoảng cách đến nút lân cận
- Mọi nút gửi thông tin khoảng cách đến các nút lân cận



Vài đường liên kết:

- Không có
- Giá cao

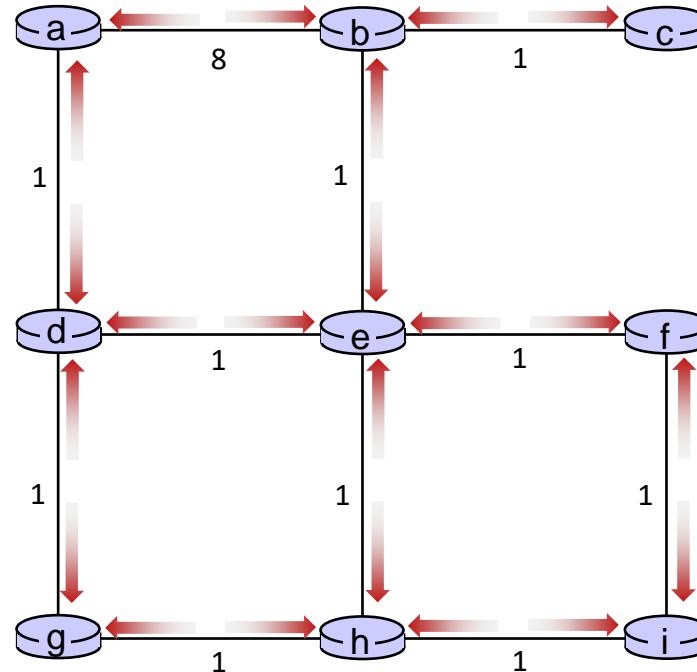
Distance vector : lặp



$t=1$

Mọi nút:

- Nhận thông tin khoảng cách vector từ hàng xóm
- Tính toán lại khoảng cách vector mới
- Gửi thông tin khoảng cách vector mới cho hàng xóm



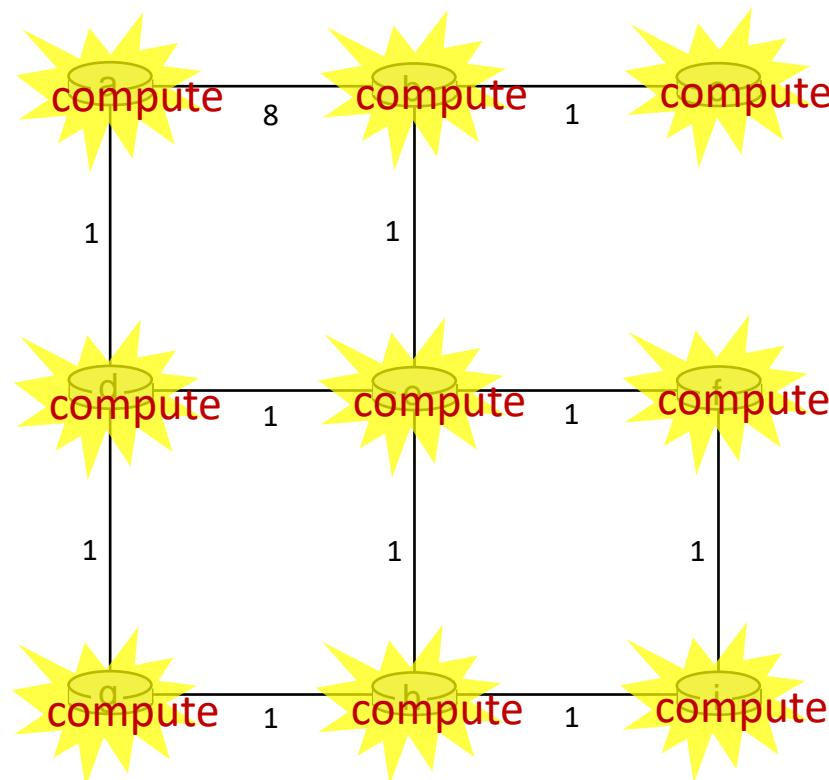
Distance vector : lặp



t=1

All nodes:

- Nhận thông tin khoảng cách vector từ hàng xóm
- Tính toán lại khoảng cách vector mới
- Gửi thông tin khoảng cách vector mới cho hàng xóm



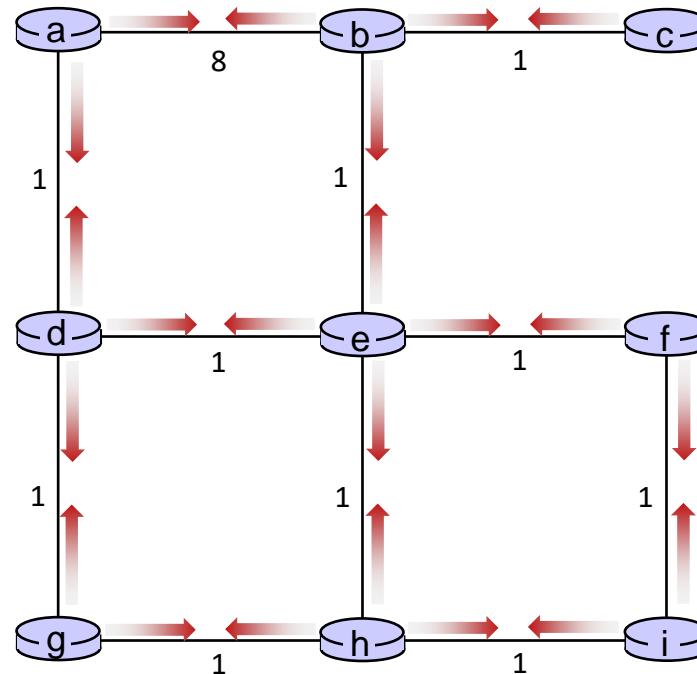
Distance vector : lặp



t=1

All nodes:

- Nhận thông tin khoảng cách vector từ hàng xóm
- Tính toán lại khoảng cách vector mới
- Gửi thông tin khoảng cách vector mới cho hàng xóm



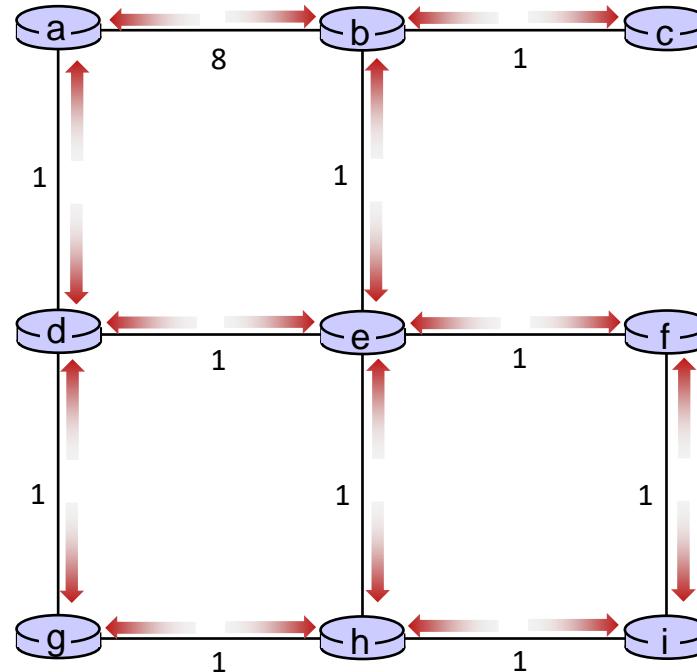
Distance vector : lặp



$t=2$

Mọi nút:

- Nhận thông tin khoảng cách vector từ hàng xóm
- Tính toán lại khoảng cách vector mới
- Gửi thông tin khoảng cách vector mới cho hàng xóm



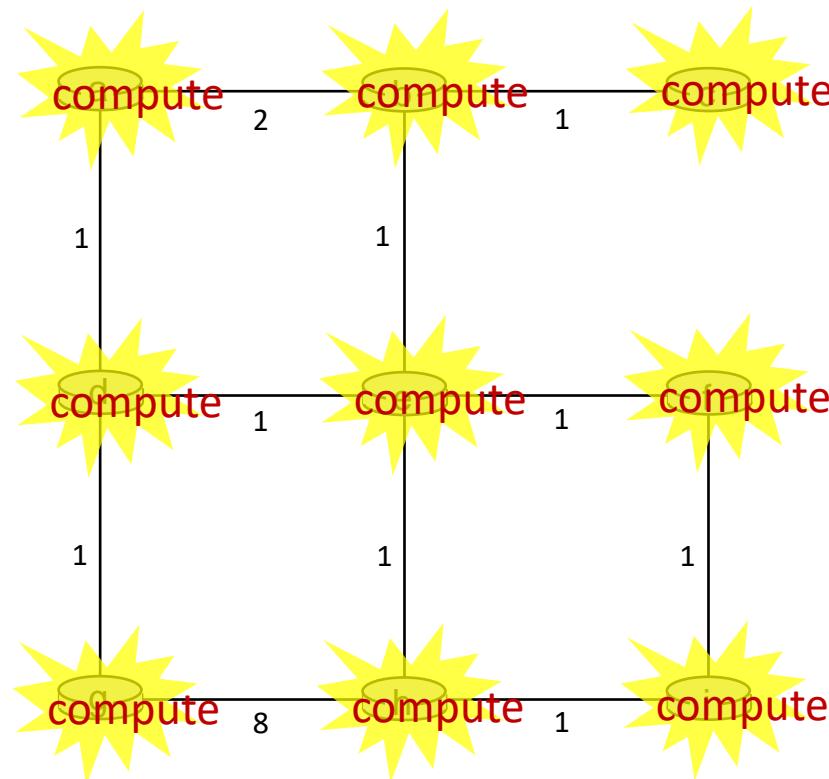
Distance vector : lặp



t=2

All nodes:

- Nhận thông tin khoảng cách vector từ hàng xóm
- Tính toán lại khoảng cách vector mới
- Gửi thông tin khoảng cách vector mới cho hàng xóm



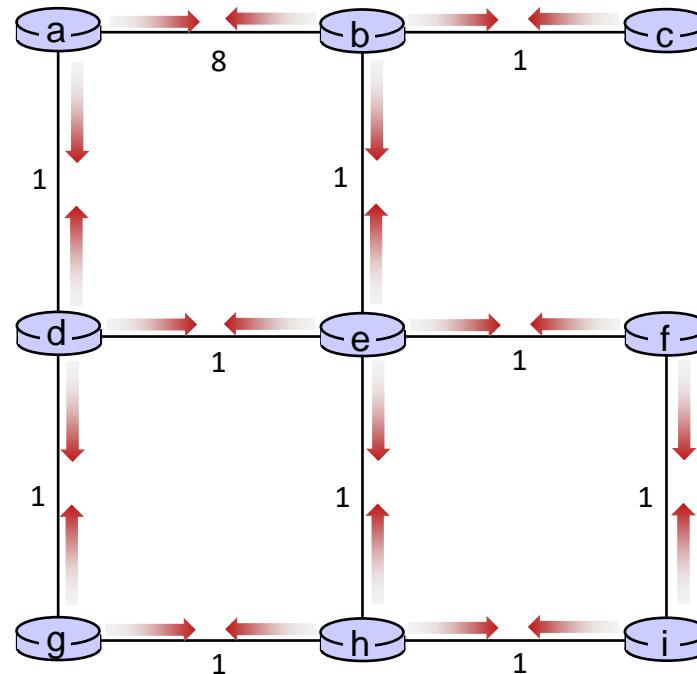
Distance vector : lặp



t=2

All nodes:

- Nhận thông tin khoảng cách vector từ hàng xóm
- Tính toán lại khoảng cách vector mới
- Gửi thông tin khoảng cách vector mới cho hàng xóm



Distance vector : lặp

.... Và tiếp tục như vậy

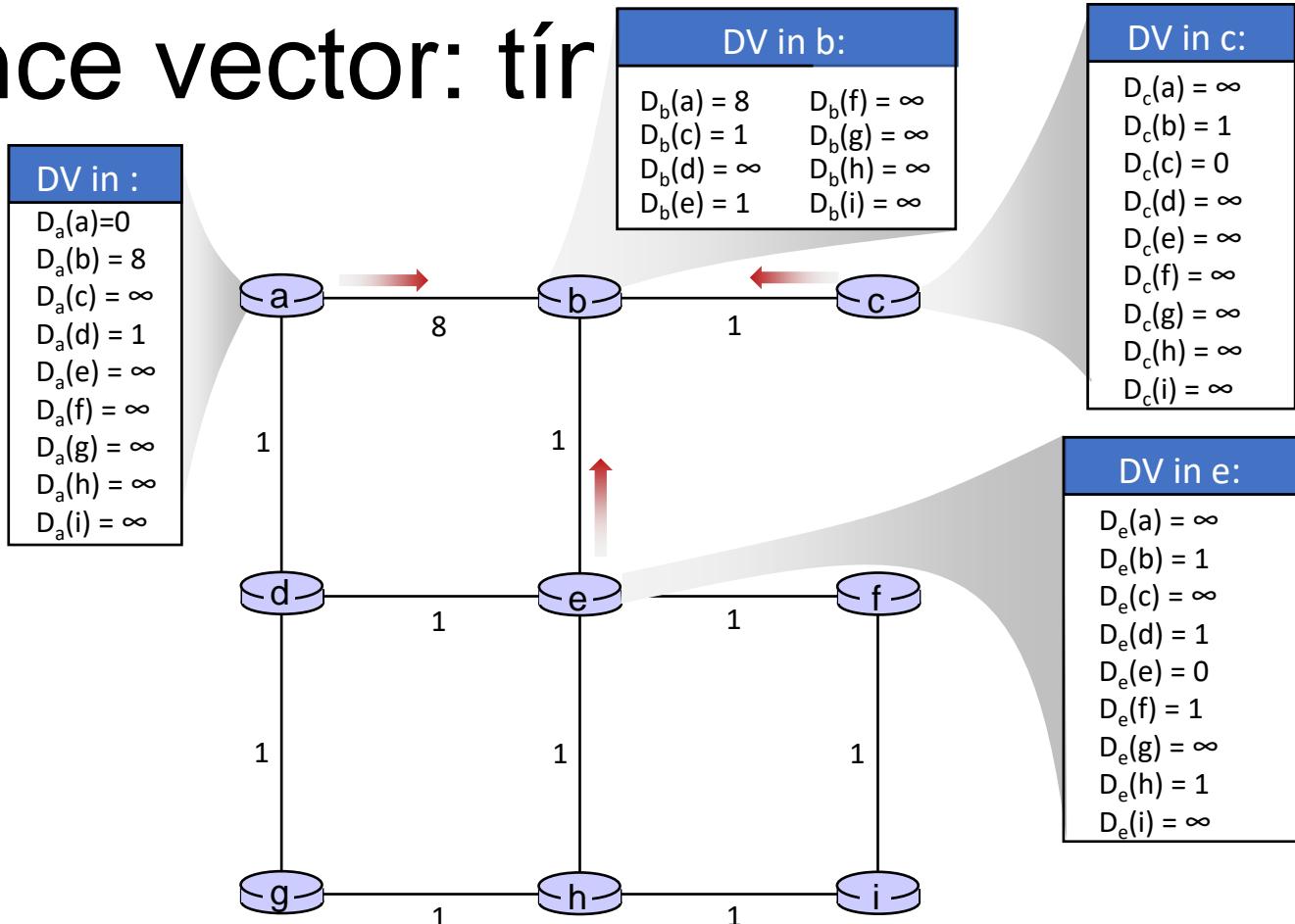
Giờ hãy thử xem việc tính toán trong một lần lặp tại các nút

Distance vector: tóm



$t=1$

- b nhận DVs từ a, c, e



Distance vector: tóm

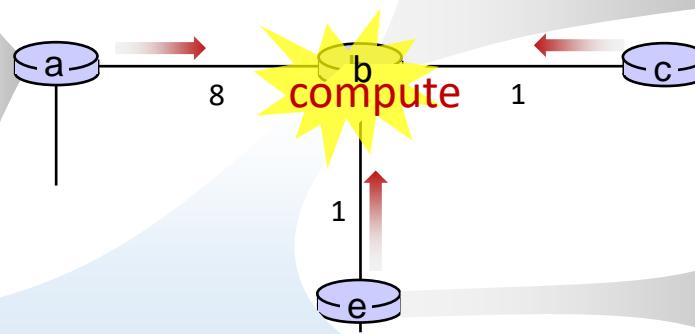


$t=1$

- b nhận DVs từ a, c, e, và tính toán:

$$\begin{aligned}
 D_b(a) &= \min\{c_{b,a}+D_a(a), c_{b,c}+D_c(a), c_{b,e}+D_e(a)\} = \min\{8, \infty, \infty\} = 8 \\
 D_b(c) &= \min\{c_{b,a}+D_a(c), c_{b,c}+D_c(c), c_{b,e}+D_e(c)\} = \min\{\infty, 1, \infty\} = 1 \\
 D_b(d) &= \min\{c_{b,a}+D_a(d), c_{b,c}+D_c(d), c_{b,e}+D_e(d)\} = \min\{9, 2, \infty\} = 2 \\
 D_b(e) &= \min\{c_{b,a}+D_a(e), c_{b,c}+D_c(e), c_{b,e}+D_e(e)\} = \min\{\infty, \infty, 1\} = 1 \\
 D_b(f) &= \min\{c_{b,a}+D_a(f), c_{b,c}+D_c(f), c_{b,e}+D_e(f)\} = \min\{\infty, \infty, 2\} = 2 \\
 D_b(g) &= \min\{c_{b,a}+D_a(g), c_{b,c}+D_c(g), c_{b,e}+D_e(g)\} = \min\{\infty, \infty, \infty\} = \infty \\
 D_b(h) &= \min\{c_{b,a}+D_a(h), c_{b,c}+D_c(h), c_{b,e}+D_e(h)\} = \min\{\infty, \infty, 2\} = 2 \\
 D_b(i) &= \min\{c_{b,a}+D_a(i), c_{b,c}+D_c(i), c_{b,e}+D_e(i)\} = \min\{\infty, \infty, \infty\} = \infty
 \end{aligned}$$

DV in :
$D_a(a)=0$
$D_a(b)=8$
$D_a(c)=\infty$
$D_a(d)=1$
$D_a(e)=\infty$
$D_a(f)=\infty$
$D_a(g)=\infty$
$D_a(h)=\infty$
$D_a(i)=\infty$



DV in b:
$D_b(a) = 8$
$D_b(c) = 1$
$D_b(d) = \infty$
$D_b(e) = 1$
$D_b(f) = \infty$
$D_b(g) = \infty$
$D_b(h) = \infty$
$D_b(i) = \infty$

DV in c:
$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$

DV in e:
$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

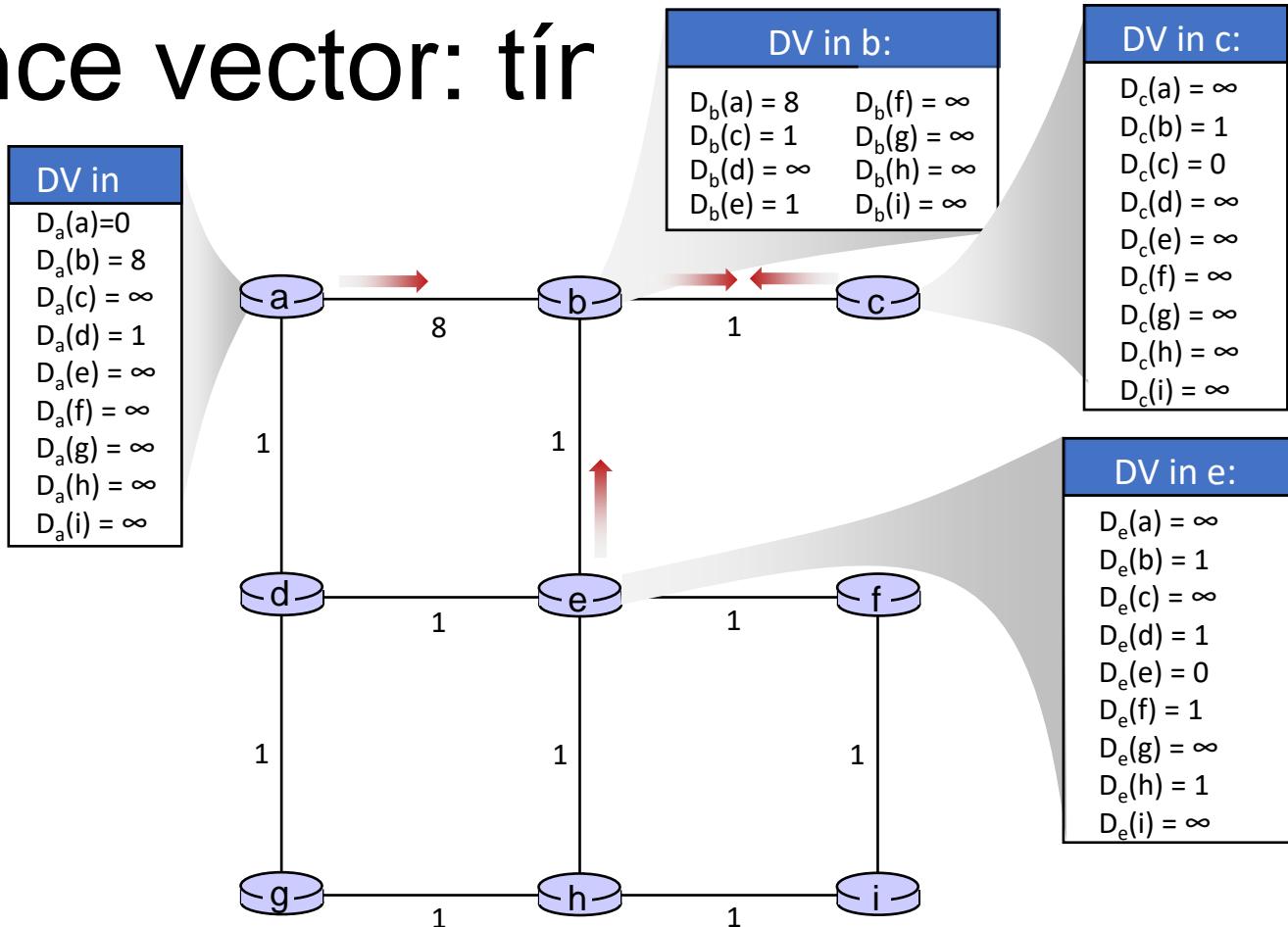
DV in b:
$D_b(a) = 8$
$D_b(c) = 1$
$D_b(d) = 2$
$D_b(e) = 1$
$D_b(f) = 2$
$D_b(g) = \infty$
$D_b(h) = 2$
$D_b(i) = \infty$

Distance vector: tóm



$t=1$

- c nhận DVs từ b



Distance vector: tóm



t=1

- c nhận DVs từ b và tính toán:

$$\begin{aligned}D_c(a) &= \min\{c_{c,b} + D_b(a)\} = 1 + 8 = 9 \\D_c(b) &= \min\{c_{c,b} + D_b(b)\} = 1 + 0 = 1 \\D_c(d) &= \min\{c_{c,b} + D_b(d)\} = 1 + \infty = \infty \\D_c(e) &= \min\{c_{c,b} + D_b(e)\} = 1 + 1 = 2 \\D_c(f) &= \min\{c_{c,b} + D_b(f)\} = 1 + \infty = \infty \\D_c(g) &= \min\{c_{c,b} + D_b(g)\} = 1 + \infty = \infty \\D_c(h) &= \min\{c_{c,b} + D_b(h)\} = 1 + \infty = \infty \\D_c(i) &= \min\{c_{c,b} + D_b(i)\} = 1 + \infty = \infty\end{aligned}$$

DV in b:	
$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$



DV in c:	
$D_c(a) = \infty$	
$D_c(b) = 1$	
$D_c(c) = 0$	
$D_c(d) = \infty$	
$D_c(e) = \infty$	
$D_c(f) = \infty$	
$D_c(g) = \infty$	
$D_c(h) = \infty$	
$D_c(i) = \infty$	

DV in c:	
$D_c(a) = 9$	
$D_c(b) = 1$	
$D_c(c) = 0$	
$D_c(d) = 2$	
$D_c(e) = \infty$	
$D_c(f) = \infty$	
$D_c(g) = \infty$	
$D_c(h) = \infty$	
$D_c(i) = \infty$	

Distance vector: tóm



$t=1$

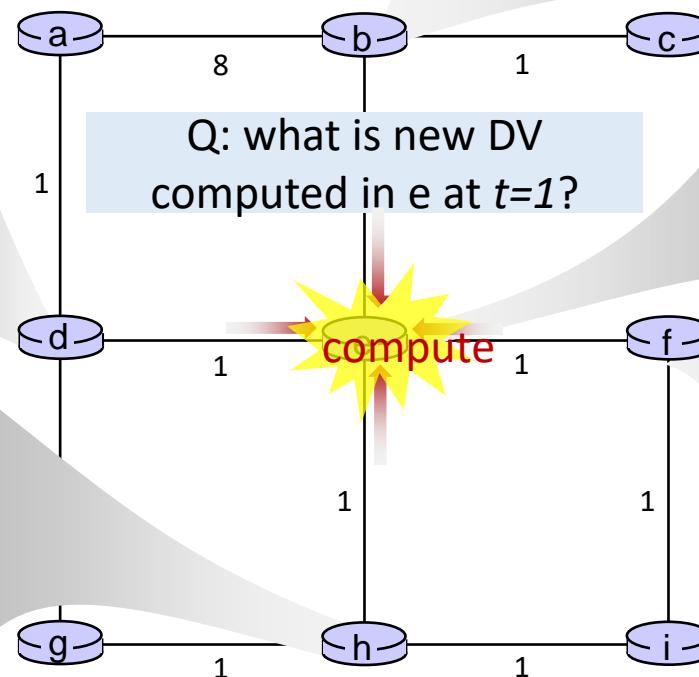
- e nhận DVs từ b, d, f, h

DV in d:
$D_c(a) = 1$
$D_c(b) = \infty$
$D_c(c) = \infty$
$D_c(d) = 0$
$D_c(e) = 1$
$D_c(f) = \infty$
$D_c(g) = 1$
$D_c(h) = \infty$
$D_c(i) = \infty$

DV in b:
$D_b(a) = 8$
$D_b(c) = 1$
$D_b(d) = \infty$
$D_b(e) = 1$
$D_b(f) = \infty$
$D_b(g) = \infty$
$D_b(h) = \infty$
$D_b(i) = \infty$

DV in e:
$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

DV in f:
$D_c(a) = \infty$
$D_c(b) = \infty$
$D_c(c) = \infty$
$D_c(d) = \infty$
$D_c(e) = 1$
$D_c(f) = 0$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = 1$

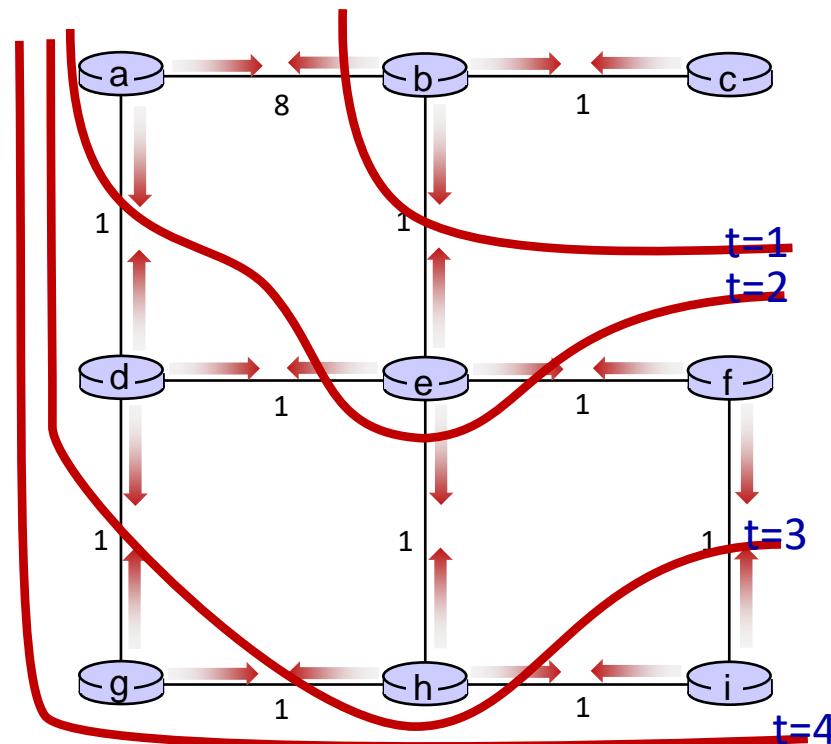


DV in h:
$D_c(a) = \infty$
$D_c(b) = \infty$
$D_c(c) = \infty$
$D_c(d) = \infty$
$D_c(e) = 1$
$D_c(f) = \infty$
$D_c(g) = 1$
$D_c(h) = 0$
$D_c(i) = 1$

Distance vector: lan truyền thông tin

Thông qua việc truyền thông tin, các bước tính toán lan truyền thông tin trong mạng

-  t=0 Trạng thái c's t=0 chỉ tại c
-  t=1 Trạng thái c's t=0 lan truyền đến b và có thể ảnh hưởng đến việc tính toán khoảng cách vector mới khoảng cách 1 hop
-  t=2 Trạng thái c's t=0 lan truyền và có thể ảnh hưởng đến việc tính toán khoảng cách vector mới khoảng cách 2 hop
-  t=3 Trạng thái c's t=0 lan truyền và có thể ảnh hưởng đến việc tính toán khoảng cách vector mới khoảng cách 3 hop
-  t=4 Trạng thái c's t=0 lan truyền và có thể ảnh hưởng đến việc tính toán khoảng cách vector mới khoảng cách 4 hop



So sánh các giải thuật

Thông điệp trao đổi

- LS: n nút, E cạnh, $O(nE)$ thông điệp
- DV: Chỉ trao đổi giữa các hàng xóm
 - thay đổi

Tốc độ hội tụ

- LS: Thuật toán: $O(n^2)$ cần $O(nE)$ thông điệp
- DV: Thay đổi

Sự chắc chắn: Giải sử một router hoạt động sai

LS:

- nút gửi các chi phí sai
- Mỗi nút tính riêng bảng chọn đường -> có vẻ chắc chắn hơn

DV:

- DV có thể bị gửi sai
- Mỗi nút tính toán dựa trên các nút khác
 - Lỗi bị lan truyền trong mạng



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Các giao thức chọn đường

Đọc trước: Chapter 5- Computer Networks, Tanenbaum

Phân cấp trong chọn đường

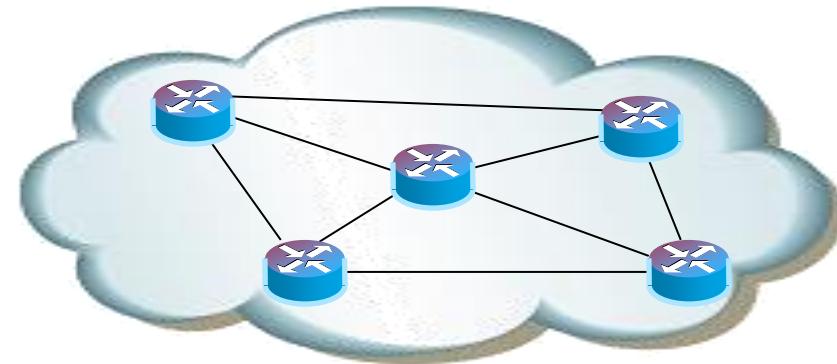
Các hệ tự trị

Chọn đường nội vùng

Chọn đường liên vùng

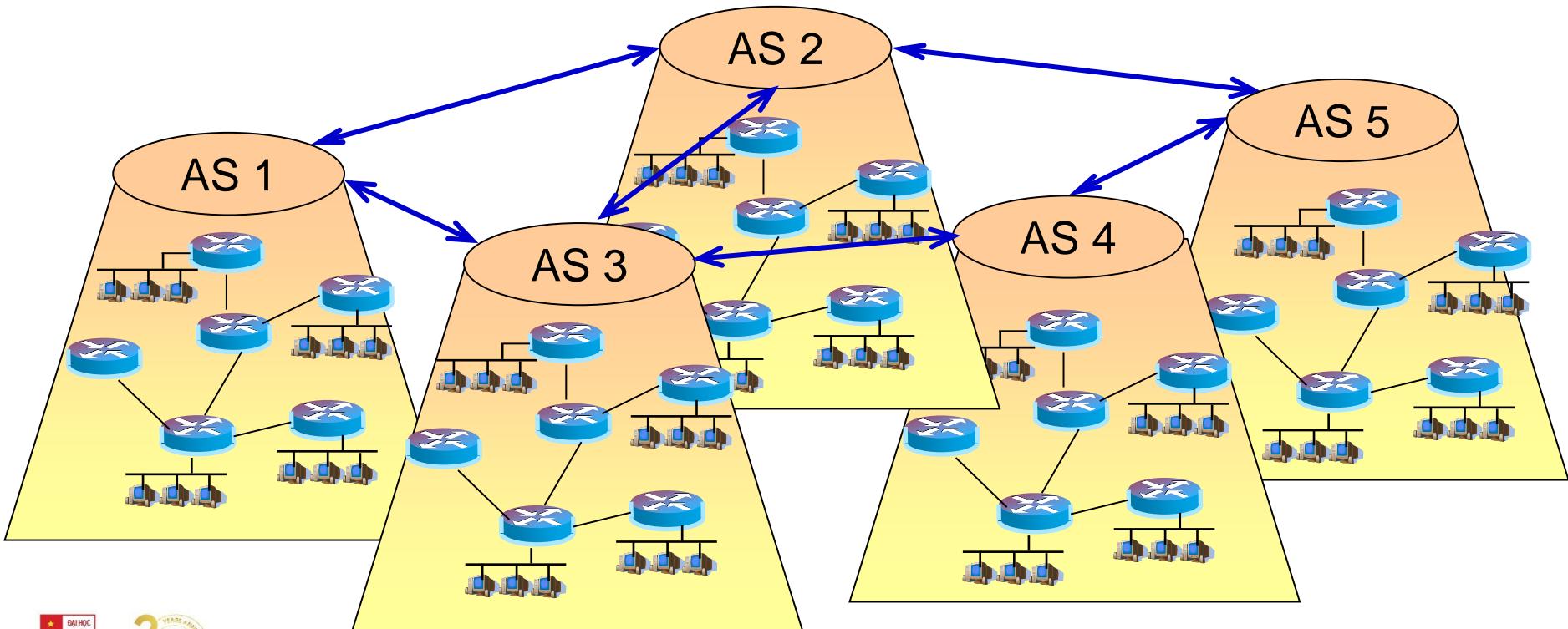
Tổng quan

- Vấn đề chọn đường đã học được xem xét trong điều kiện lý tưởng
 - Các nút mạng có vai trò như nhau
 - Chỉ có một mạng duy nhất, mạng “phẳng”
- Thực tế không giống như vậy
- Tính mở rộng: Internet có hàng triệu (tỷ) máy trạm, chọn đường bằng LS hay DV?
 - LS: Quá tải thông tin chọn đường
 - DV: Có hội tụ được không?
- Quyền tự trị (administrative autonomy)
 - Mỗi mạng có chính sách riêng



Kiến trúc phân cấp của Internet

- Internet = Mạng của các mạng
- Mỗi mạng có thể lựa chọn riêng cho mình một chiến lược chọn đường riêng.
- Mỗi mạng như vậy có thể gọi là một hệ tự trị - Autonomous System (AS)



Khái niệm hệ tự trị - AS

- Tập hợp các nút mạng có cùng chính sách chọn đường (Giao thức, quy ước chi phí...)
- Các ASes được nối kết thông qua các router hay gateway
- Mỗi hệ tự trị có một số hiệu riêng – AS number (ASN - 16 bits hay 32 bits).

[2914](#) NTT-COMMUNICATIONS-2914 - NTT America, Inc.

[3491](#) BTN-ASN - Beyond The Network America, Inc.

[4134](#) CHINANET-BACKBONE No.31,Jin-rong Street

[6453](#) GLOBEINTERNET Teleglobe America Inc.

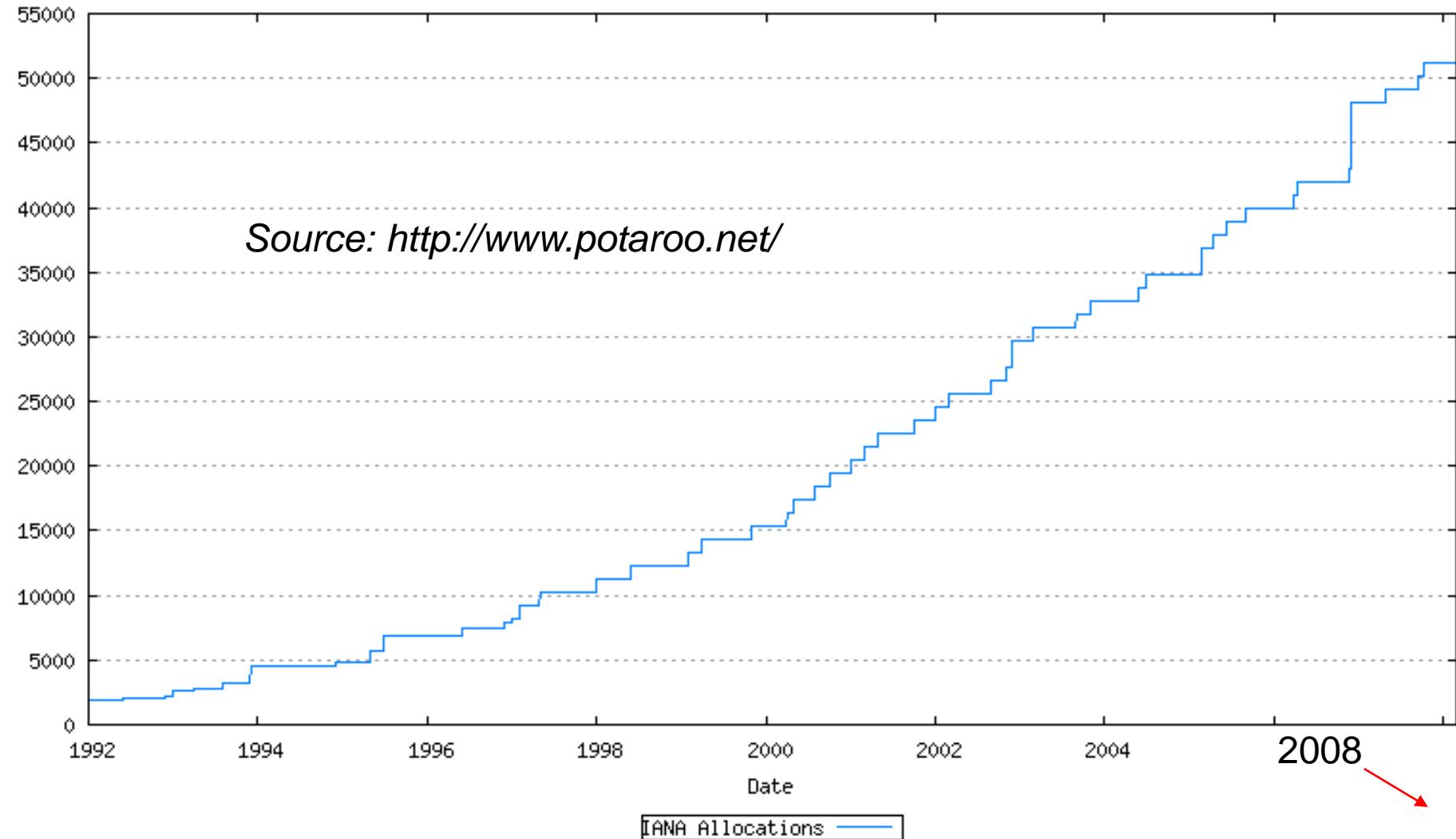
[24087](#) VNGT-AS-AP Vietnam New Generation Telecom

[24066](#) VNNIC-AS-VN Vietnam Internet Network Information Center

[17981](#) CAMBOTECH-KH-AS ISP Cambodia

Số lượng ASN cấp phát bởi IANA

Time Series of IANA AS Allocations



Cách tiếp cận của Internet để định tuyến

- Các router phân thành từng khu vực gọi là “autonomous systems” (AS) (“domains”) – Hệ tự trị

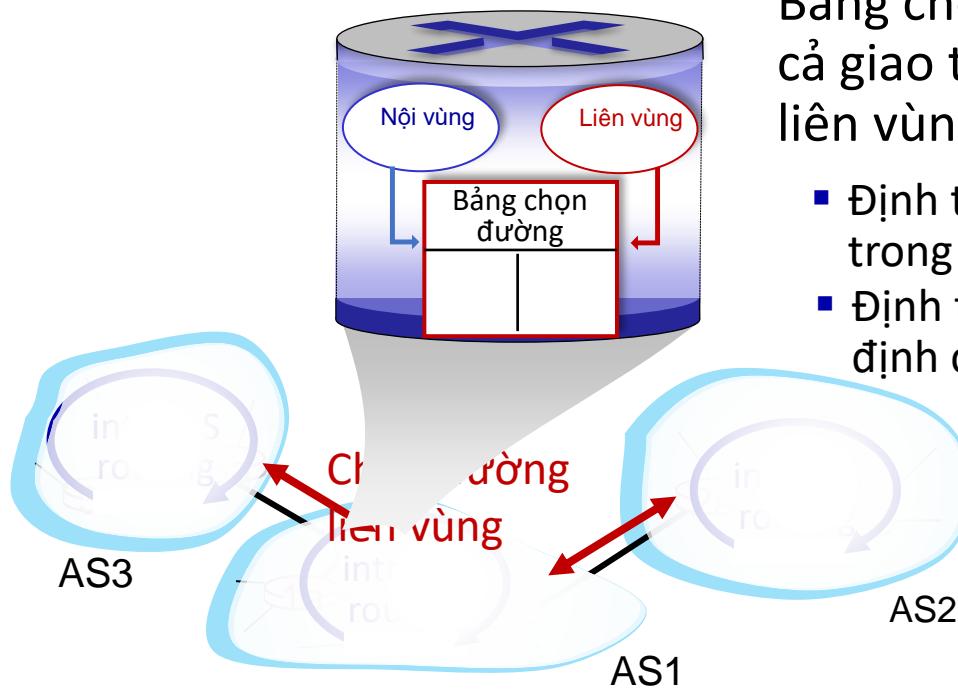
intra-AS (“intra-domain”): định tuyến giữa những router cùng miền (“network”) – *Giao thức chọn đường nội vùng*

- Tất cả các router cùng miền dùng chung 1 giải thuật định tuyến
- Router khác miền có thể chạy các thuật toán intra-AS khác nhau
- **gateway router:** các router tại vùng “biên” giữa các miền/vùng, có liên kết với router khác vùng

inter-AS (“inter-domain”): chọn đường giữa các AS (hệ tự trị) – *Giao thức chọn đường liên vùng*

- Gateway router thực hiện các giao thức chọn đường liên vùng (cùng như nội vùng)

Kết nối các miền tự trị ASes



Bảng chọn đường được cấu hình bởi cả giao thức định tuyến nội vùng và liên vùng

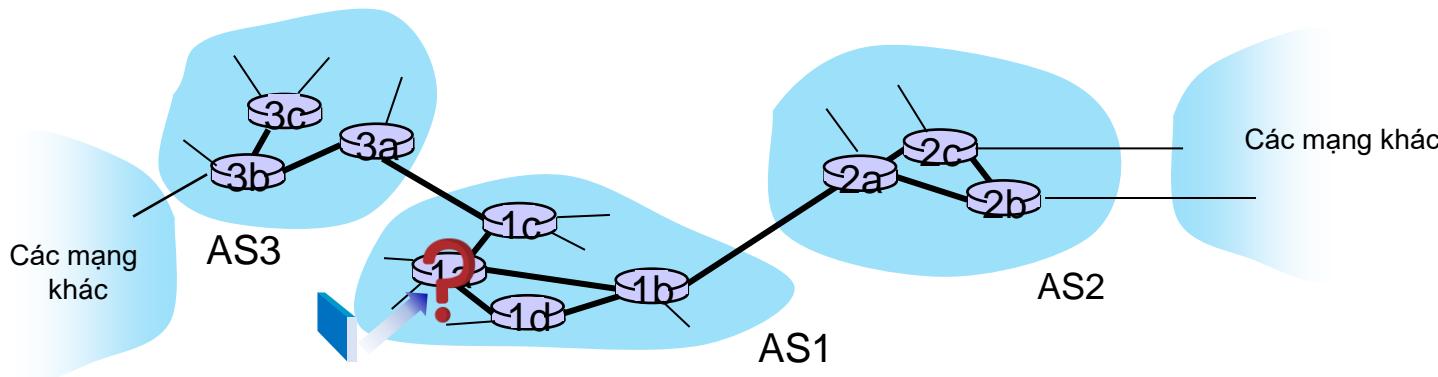
- Định tuyến nội vùng xác định các dòng trong một vùng
- Định tuyến nội vùng và liên vùng xác định các dòng địa chỉ bên ngoài vùng

Định tuyến liên vùng: vai trò trong chọn đường nội vùng

- Giả sử router trong AS1 nhận được gói tin với địa chỉ đích ở ngoài AS1
 - Router nên chuyển gói tin đến gateway router trong AS1, nhưng router nào?

Chọn đường liên vùng của AS1 phải:

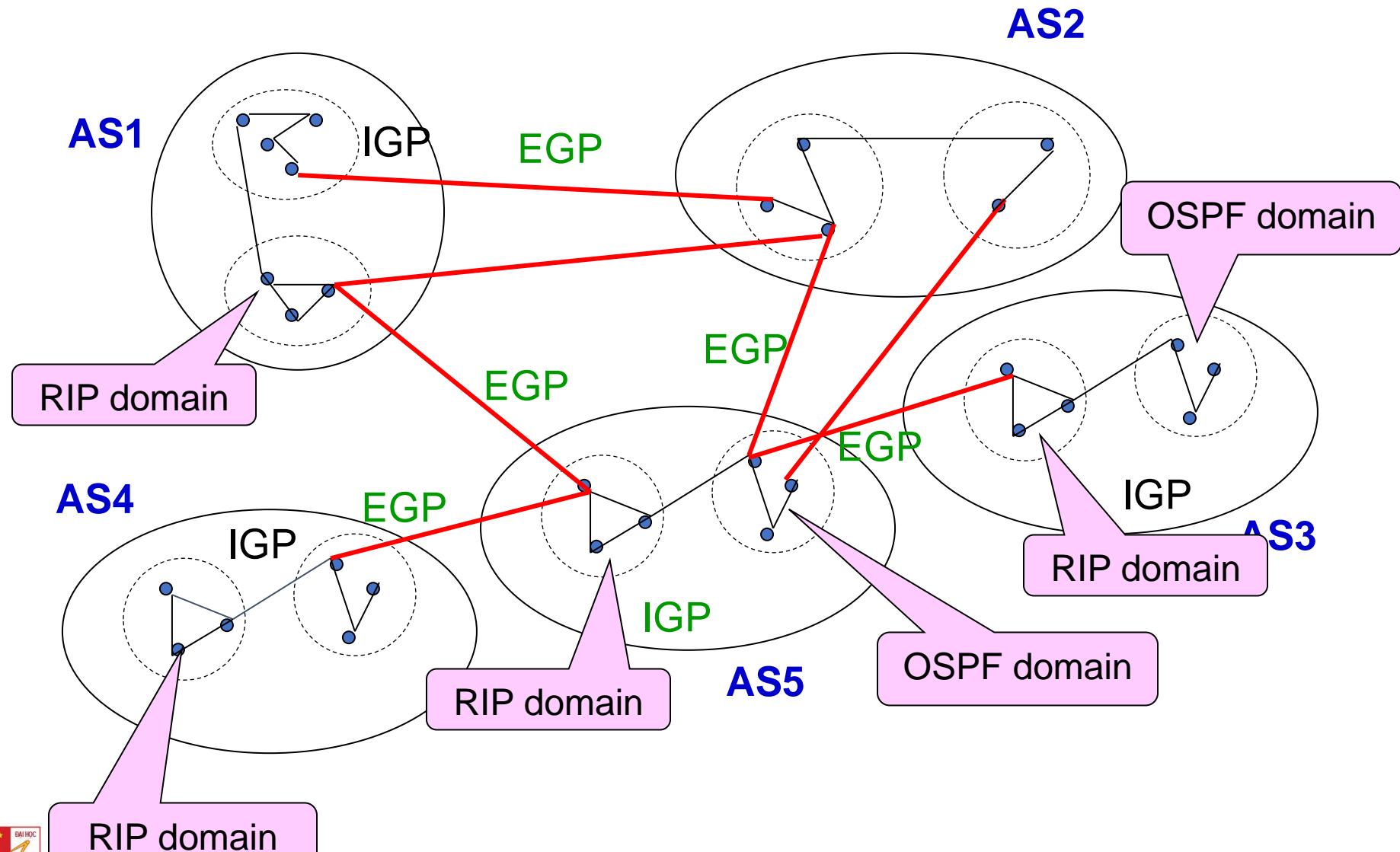
- Biết được địa chỉ đích nào đến được qua AS2, cái nào đến được qua AS3
- Lan truyền thông tin này đến tất cả router trong AS1



Phân cấp giao thức chọn đường

- Trong một hệ tự trị: Giao thức chọn đường nội vùng
 - IGP: *Interior Gateway Protocol*
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IS-IS, IGRP, EIGRP (Cisco)...
- Giữa các hệ tự trị: Giao thức chọn đường liên vùng
 - EGP: *Exterior Gateway Protocol*
 - BGP (v4): Border Gateway Protocol

Chọn đường nội vùng và liên vùng





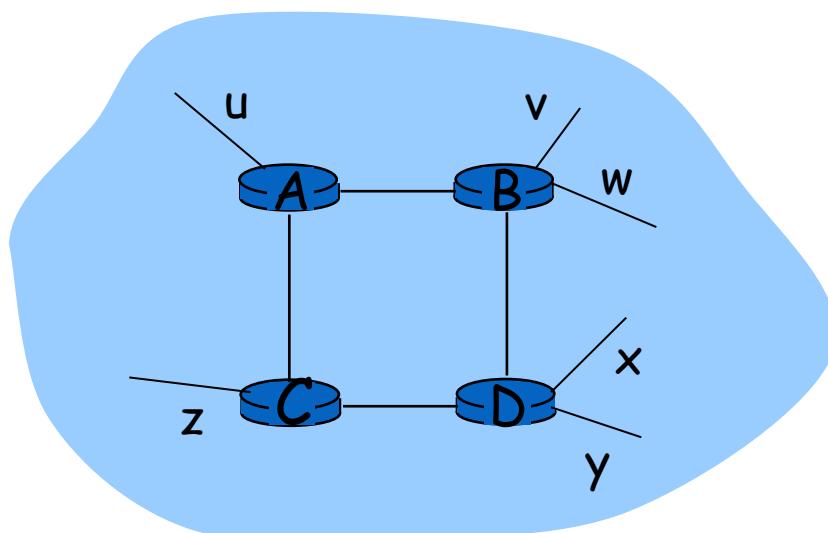
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chọn đường nội vùng

RIP
OSPF

RIP (Routing Information Protocol)

- RIP v.1, phiên bản mới RIP v.2
- Giao thức dạng Distance vector (vector khoảng cách)
- Chọn đường đi theo số nút mạng đi qua (# of hops, max = 15 hops)

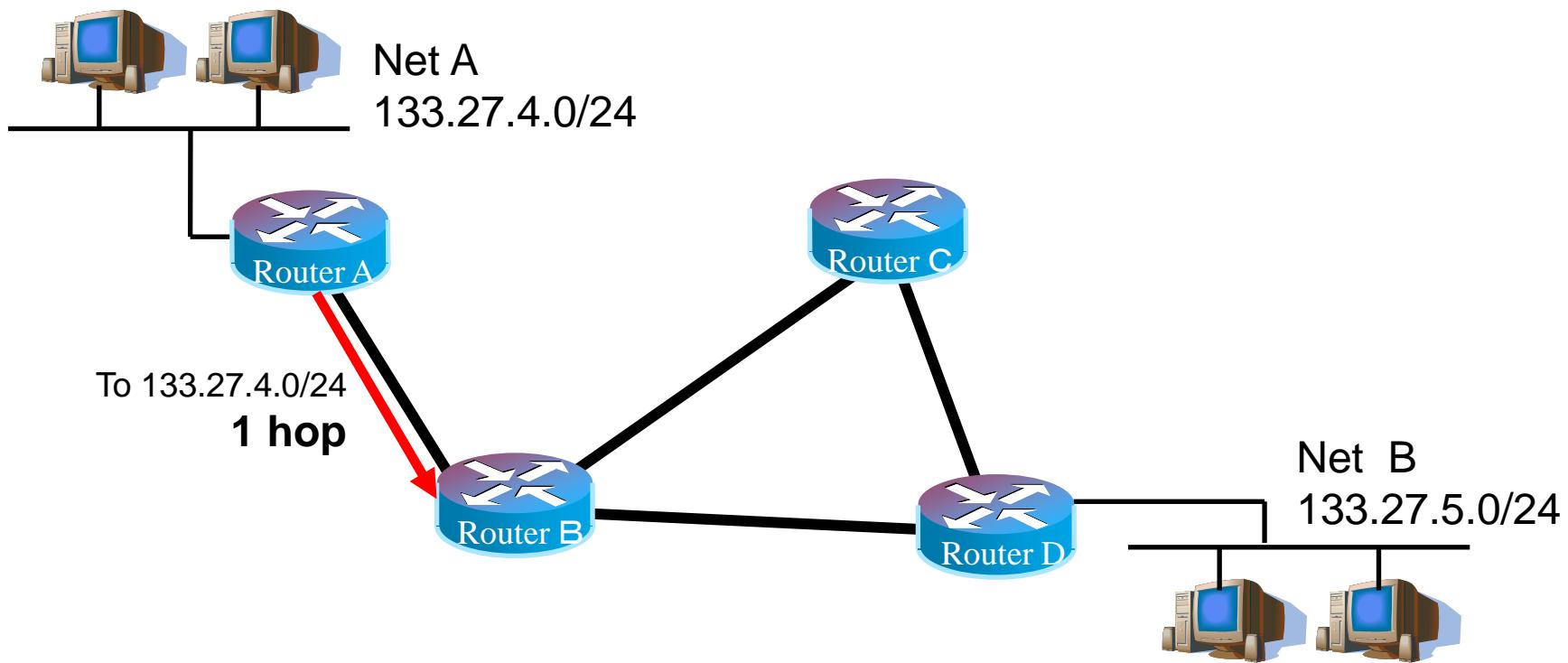


Từ nút A:

<u>Đích đến</u>	<u>số nút</u>
u	1
v	2
w	2
x	3
y	3
z	2

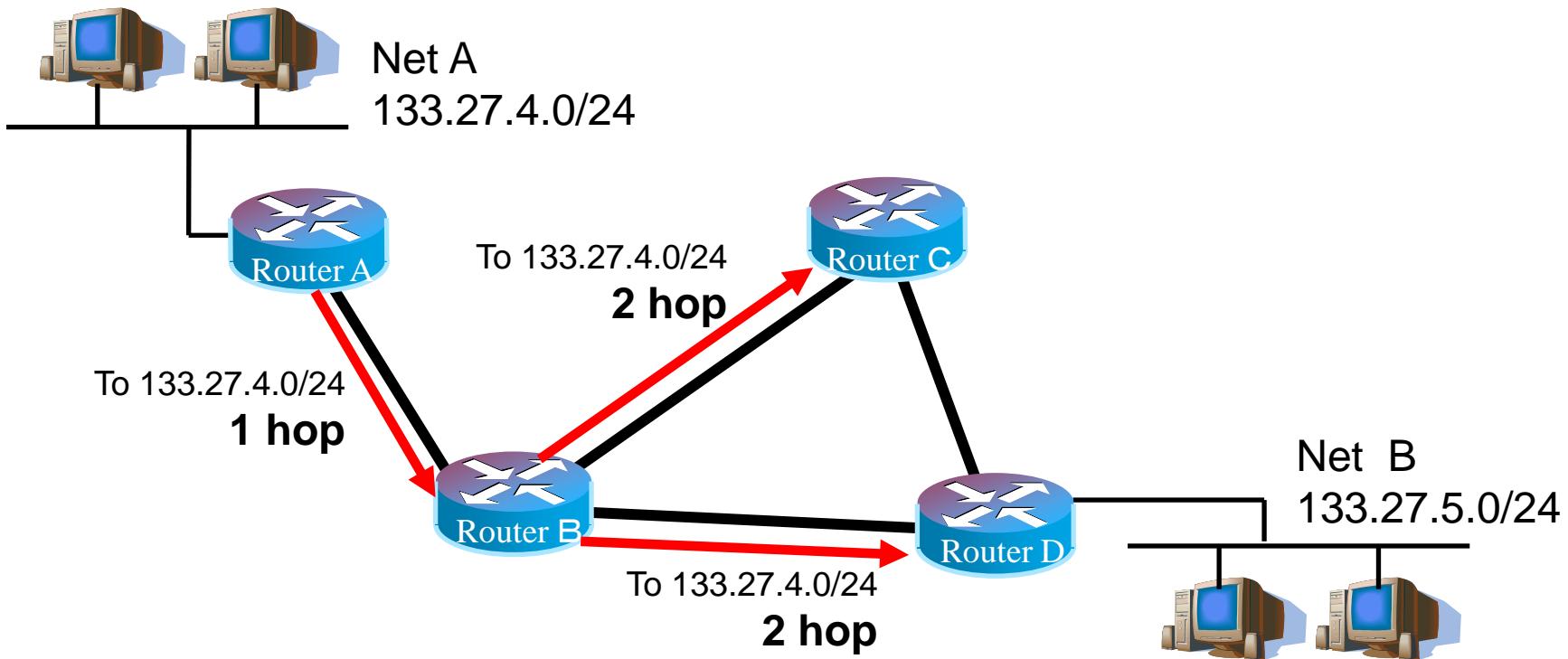
Nhắc lại: Chọn đường dạng DV (1)

- Bạn của bạn là bạn



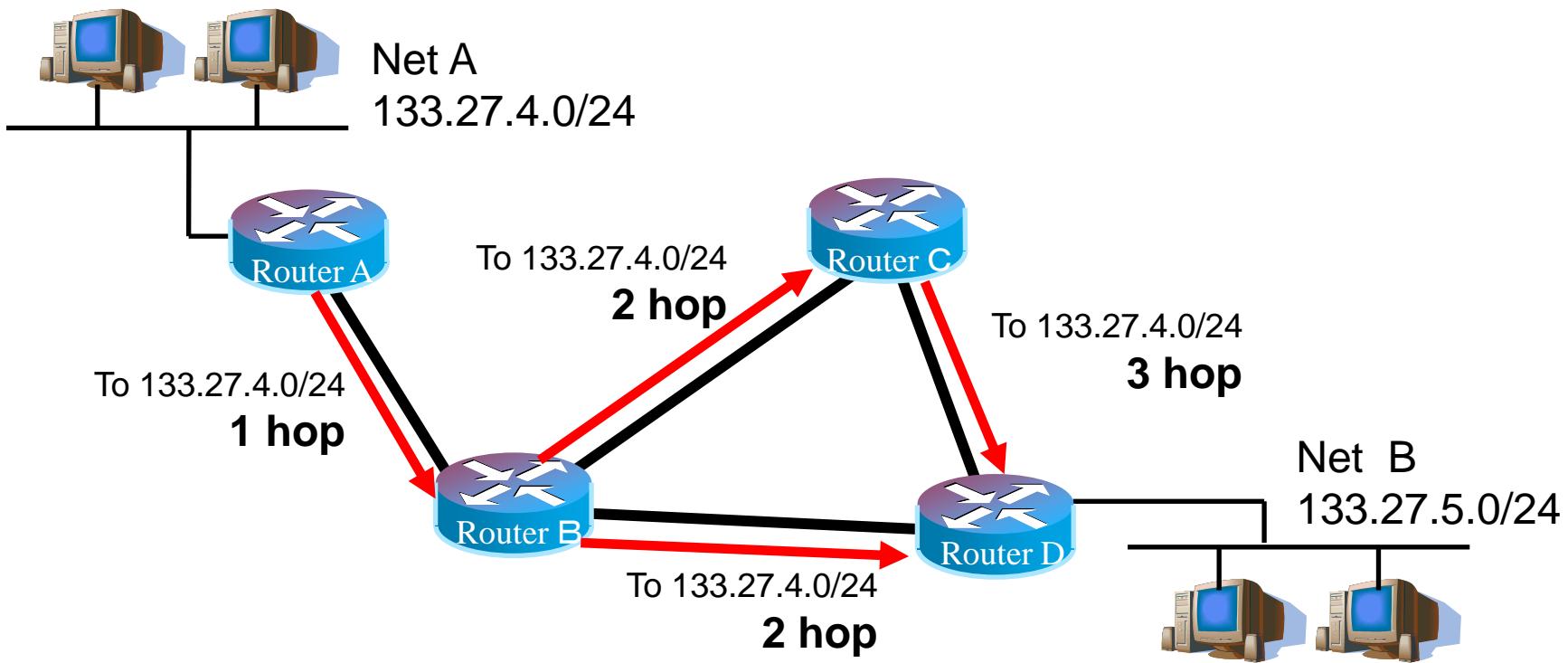
Nhắc lại: Chọn đường dẫn DV (2)

- Bạn của bạn là bạn



Nhắc lại: Chọn đường dẫn DV (3)

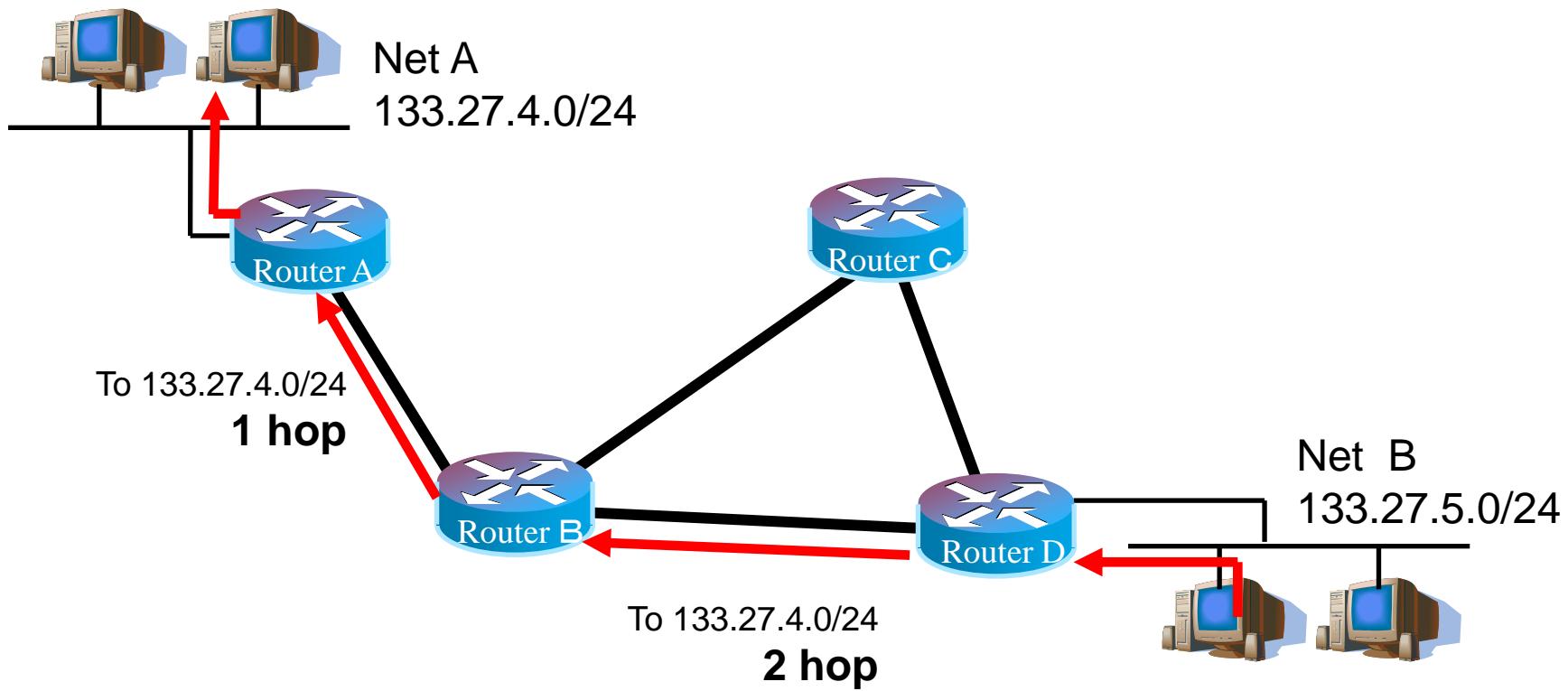
- Bạn của bạn là bạn



Nhắc lại: Chọn đường dạng DV (4)

- Bạn của bạn là bạn

Lưu ý: Tên của router



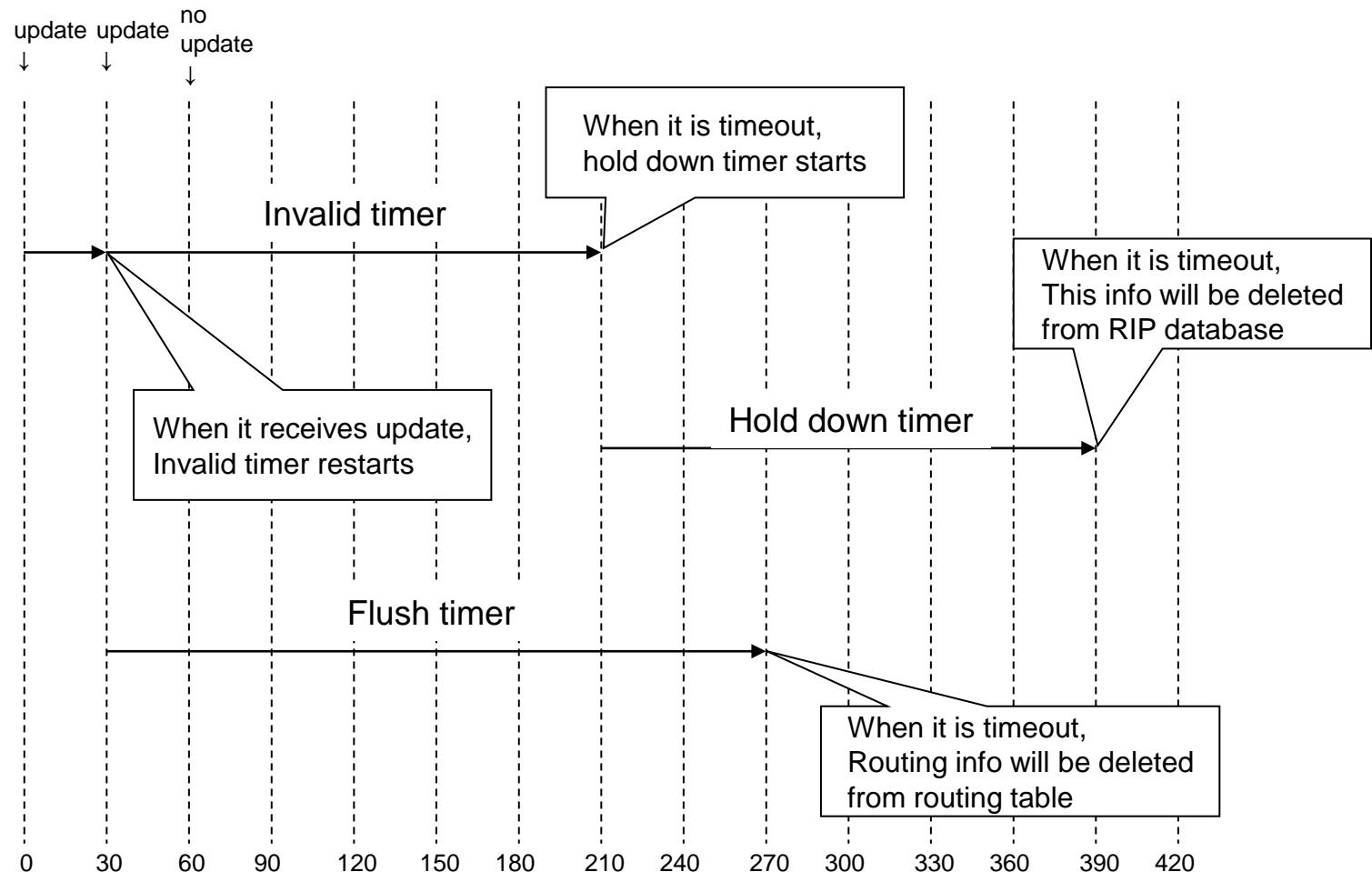
RIP: Trao đổi thông tin

- Trao đổi bằng vector khoảng cách
- Định kỳ
 - Các vector khoảng cách được trao đổi định kỳ - 30s
 - Mỗi thông điệp chứa tối đa 25 mục
 - Trong thực tế, nhiều thông điệp được sử dụng
- Sự kiện
 - Gửi thông điệp cho nút hàng xóm mỗi khi có thay đổi
 - Nút hàng xóm sẽ cập nhật bảng chọn đường của nó

Các bộ đếm thời gian - RIP timer (1)

- Update timer
 - Dùng để trao đổi thông tin cứ 30s
- Invalid timer
 - Khởi tạo lại mỗi khi nhận được thông tin chọn đường
 - Nếu sau 180s không nhận được thông tin -> trạng thái hold-down
- Hold down timer
 - Giữ trạng thái hold-down trong 180s
 - Chuyển sang trạng thái down
- Flush timer
 - Khởi tạo lại mỗi khi nhận được thông tin chọn đường
 - Sau 120s, xóa mục tương ứng trong bảng chọn đường

RIP timer (2)

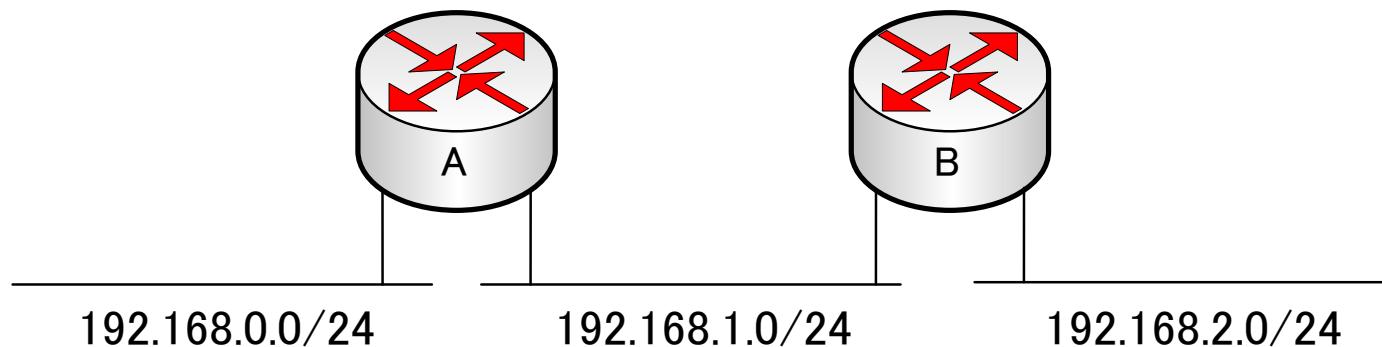


Lỗi lặp vô hạn (Ping-pong failure)

- Nếu 192.168.0.0/24 bị lỗi...
 - B cập nhật thông tin về 192.168.0.0 cho A
 - Các gói tin đến 192.168.0.0/24 sẽ bị quẩn
- A cập nhật thông tin về 192.168.0.0 cho B
 - Luẩn quẩn, vô hạn

192.168.0.0/24	conn
192.168.1.0/24	conn
192.168.2.0/24	B

192.168.1.0/24	conn
192.168.2.0/24	conn
192.168.0.0/24	A



RIP: Đã tranh lỗi lặp vô hạn

- Giới hạn số hop tối đa
 - 16
- “Split horizon”
 - Thông tin chọn đường không được quay về nút nguồn
- “Poison reverse”
 - Khi liên kết bị lỗi, gửi giá trị của chi phí là 16
 - Liên kết chuyển sang trạng thái hold-down

OSPF: Open Shortest Path First

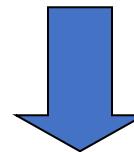
- *Open*: Chuẩn mở của IETF (phiên bản 3, định nghĩa trong [RFC 2740](#))
- *Shortest Path First*: Cài đặt giải thuật Dijkstra.
- Loại Link-state
- Thông tin về trạng thái liên kết - LSA (link state advertisement) được quảng bá “tràn ngập” trên toàn AS

Một số đặc điểm của OSPF

- **An toàn:** thông điệp OSPF được bảo vệ
- Với các AS lớn: OSPF **phân cấp**
- **Địa chỉ không phân lớp** (Variable-Length Subnet Masking -VLSM)
- Mỗi link sẽ có nhiều giá trị về **chi phí** khác nhau dựa trên TOS (tuy nhiên hơi phức tạp và chưa được sử dụng)

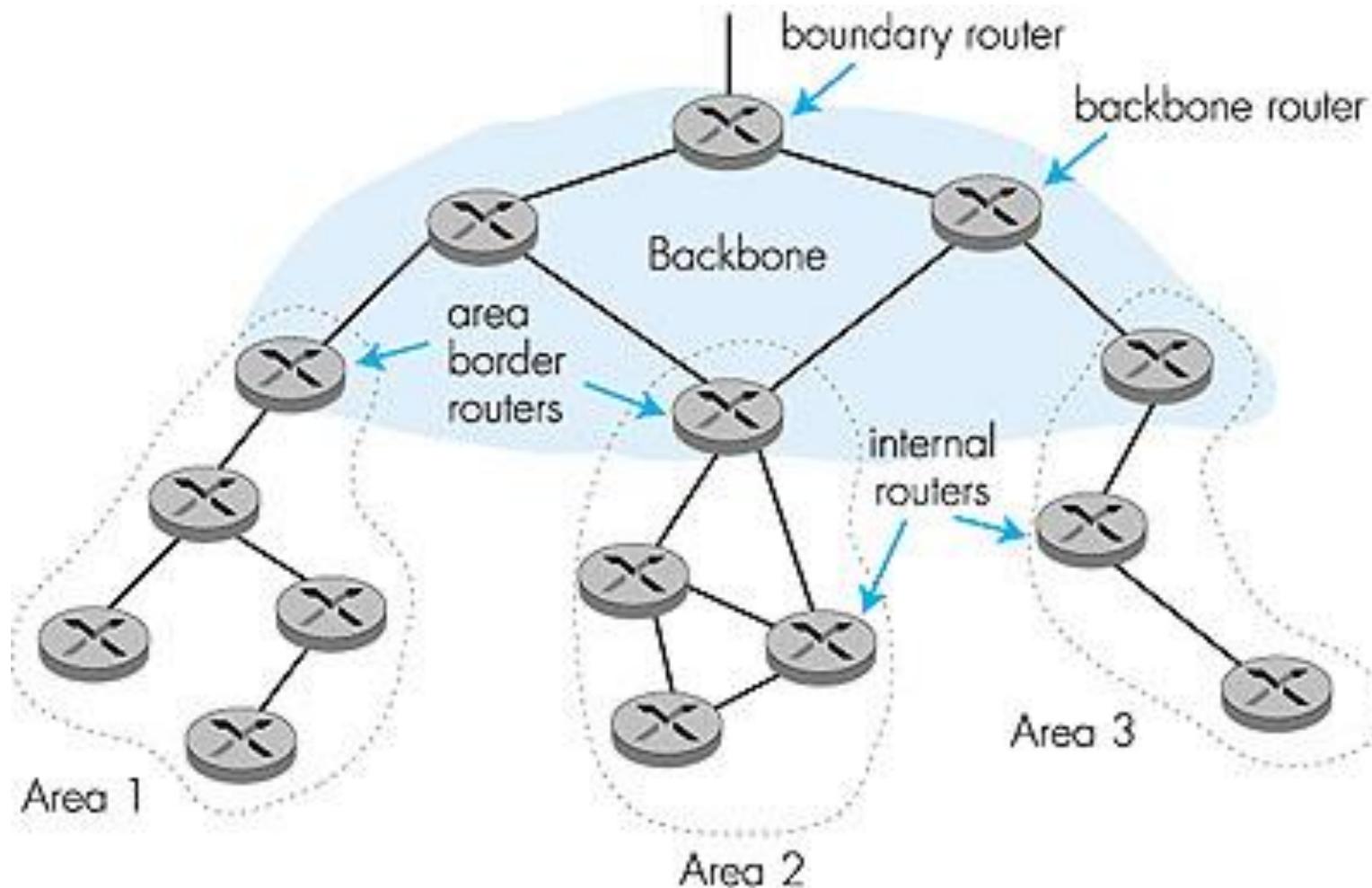
Phân cấp OSPF

- Trong việc chọn đường, tại sao phải chia mạng thành các vùng nhỏ hơn?
- Nếu có quá nhiều router
 - Thông tin trạng thái liên kết được truyền nhiều lần hơn
 - Phải liên tục tính toán lại
 - Cần nhiều bộ nhớ hơn, nhiều tài nguyên CPU hơn
 - Lượng thông tin phải trao đổi tăng lên
 - Bảng chọn đường lớn hơn



- Vùng
 - Nhóm các router có cùng thông tin LSA

Phân cấp OSPF

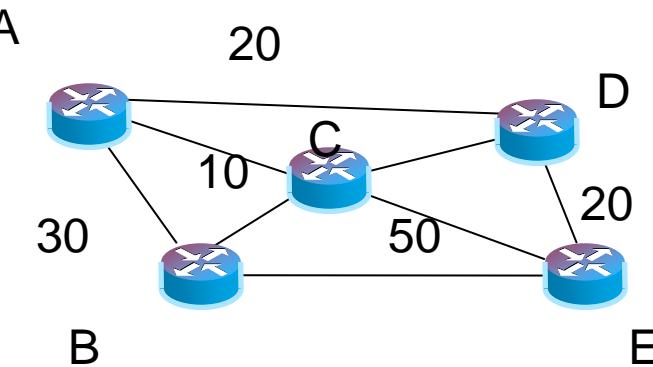


Các dạng router

- ABR - Area border routers: Quản lý 1 vùng và kết nối đến các vùng khác
- ASBR - Autonomous system boundary router: Nối đến các AS khác
- BR - backbone routers: thực hiện OSPF routing trong vùng backbone
- Internal Router – Thực hiện OSPF bên trong một vùng

Thông tin liên kết

- Link-State Advertisement (LSA): Chỉ ra một nút được nối tới nút nào (link) và chi phí (cost) tương ứng
- Ví dụ: nút A
 - link to B, cost 30
 - link to D, cost 20
 - link to C, cost 10
- Ví dụ: nút D
 - link to A, cost 20
 - link to E, cost 20
 - link to C, cost 50



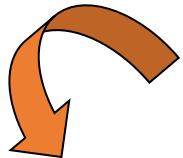
Chi phí trong giao thức OSPF - metric

- Giá trị mặc định
 - 100Mbps / bandwidth of interface
 - Hiện nay người quản trị có thể gán giá trị này
- Khi tính toán bảng chọn đường
 - Chọn đường đi chi phí nhỏ nhất
- Chi phí bằng nhau
 - Có thể thực hiện cân bằng tải

Chi phí mặc định của OSPF

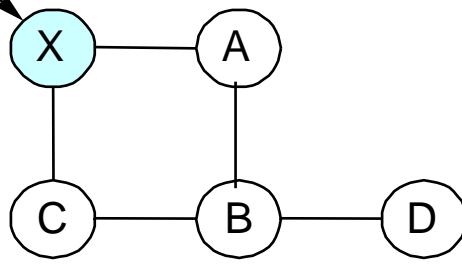
Link Bandwidth	Default OSPF cost
56Kbps serial link	1785
64Kbps serial link	1562
T1 (1.544Mbps) serial link	65
E1 (2.048Mbps) serial link	48
4Mbps Token Ring	25
Ethernet	10
16Mbps Token Ring	6
FDDI or Fast Ethernet	1
Gigabit Ethernet / 10G network	1

Quảng bá thông tin LSA



LSA_X

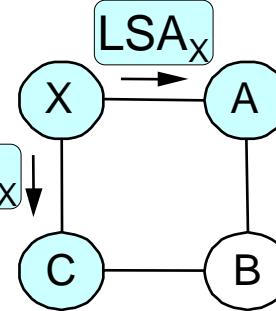
X has link to A, cost 10
X has link to C, cost 20



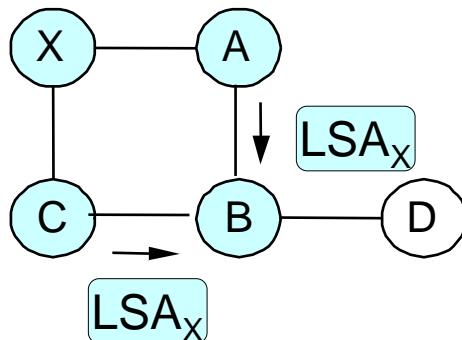
(a)

LSA_X

LSA_X



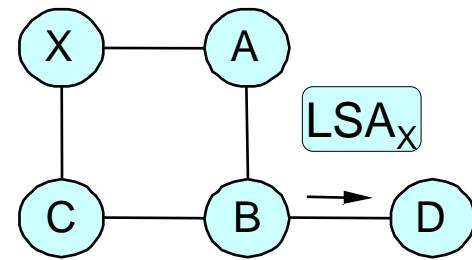
(b)



(c)

LSA_X

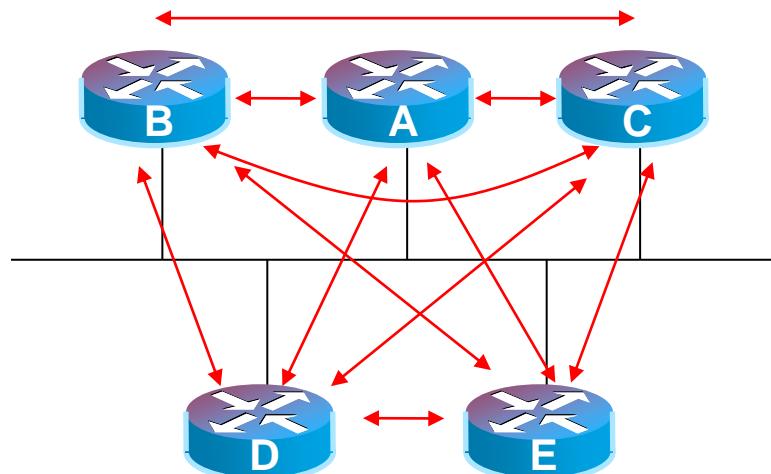
LSA_X



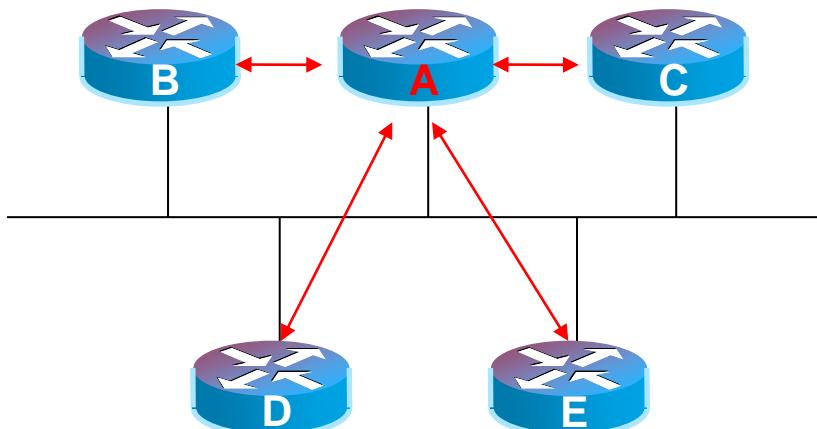
(d)

Router đại diện - DR

- Để tăng hiệu quả của việc quảng bá LSA
- Mỗi router phải lập quan hệ với router đại diện - designated router (DR)
 - Trao đổi thông tin thông qua DR
 - DR dự phòng
- Chọn DR và DR dự phòng?



Không có DR



Có DR

Neighbor & Adjacency

- “Neighbor” và “adjacency” là các khái niệm khác nhau!
 - Adjacency: có trao đổi thông tin
 - Neighbor: có đường nối trực tiếp
- Mạng quảng bá đa truy cập (e.g Ethernet)
 - Neighbor != Adjacency
- Mạng điểm-nối-điểm
 - Neighbor == Adjacency

RIP vs. OSPF

	RIP	OSPF
Đặc điểm	<ul style="list-style-type: none">Router bình đẳngCấu hình dễ dàngMạng cỡ nhỏ	<ul style="list-style-type: none">Phân cấpCấu hình phức tạpMạng cỡ vừa và lớn
Khả năng mở rộng	X	O
Độ phức tạp tính toán	Nhỏ	Lớn
Hội tụ	Chậm	Nhanh
Trao đổi thông tin	Bảng vector khoang cách	Trạng thái liên kết
Giải thuật	Distant vector	Link-state
Cập nhật hàng xóm	30s	10s (Hello packet)
Đơn vị chi phí	Số nút mạng	Băng thông



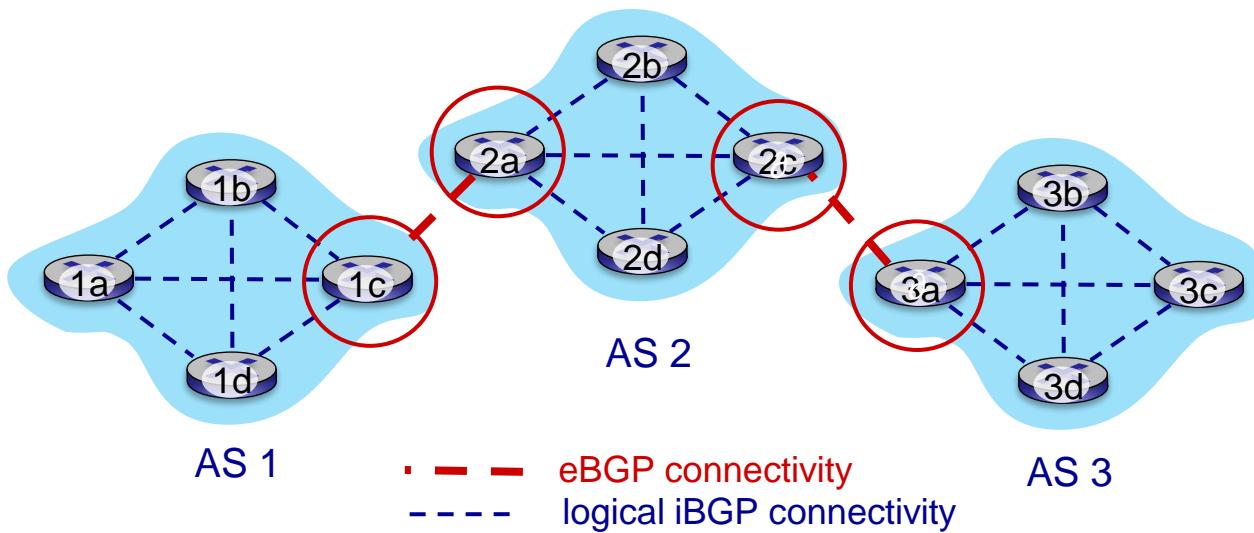
ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Giao thức chọn đường liên vùng

Giao thức chọn đường liên vùng: BGP

- **BGP (Border Gateway Protocol):** Giao thức chọn đường liên vùng
 - “chất keo gắn kết Internet với nhau” – “glue that holds the Internet together”
- Cho phép các subnet quảng bá sự tồn tại của nó và những mạng nó có thể đến được tới toàn bộ Internet: “*I am here, here is who I can reach, and how*”
- BGP cung cấp cho mỗi AS thông tin:
 - **eBGP:** lấy thông tin những subnet có thể đến được từ các AS lân cận
 - **iBGP:** lan truyền thông tin đến được đến các router nội vùng.
 - Xác định đường đi “tốt” tới các mạng khác dựa vào thông tin đến được và **chính sách**

Kết nối eBGP, iBGP

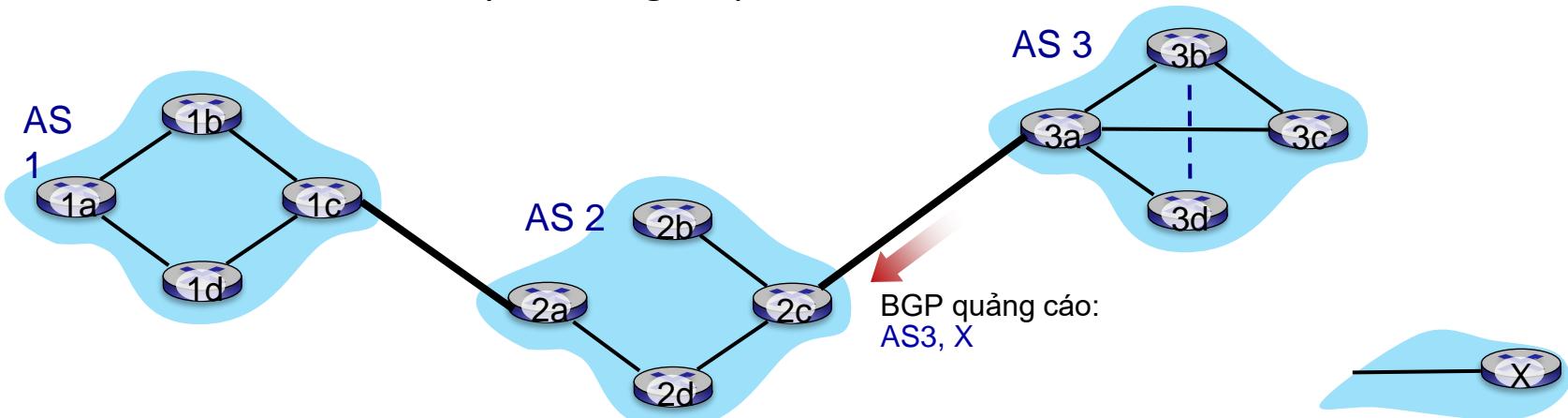


Những gateway routers chạy cả eBGP và iBGP



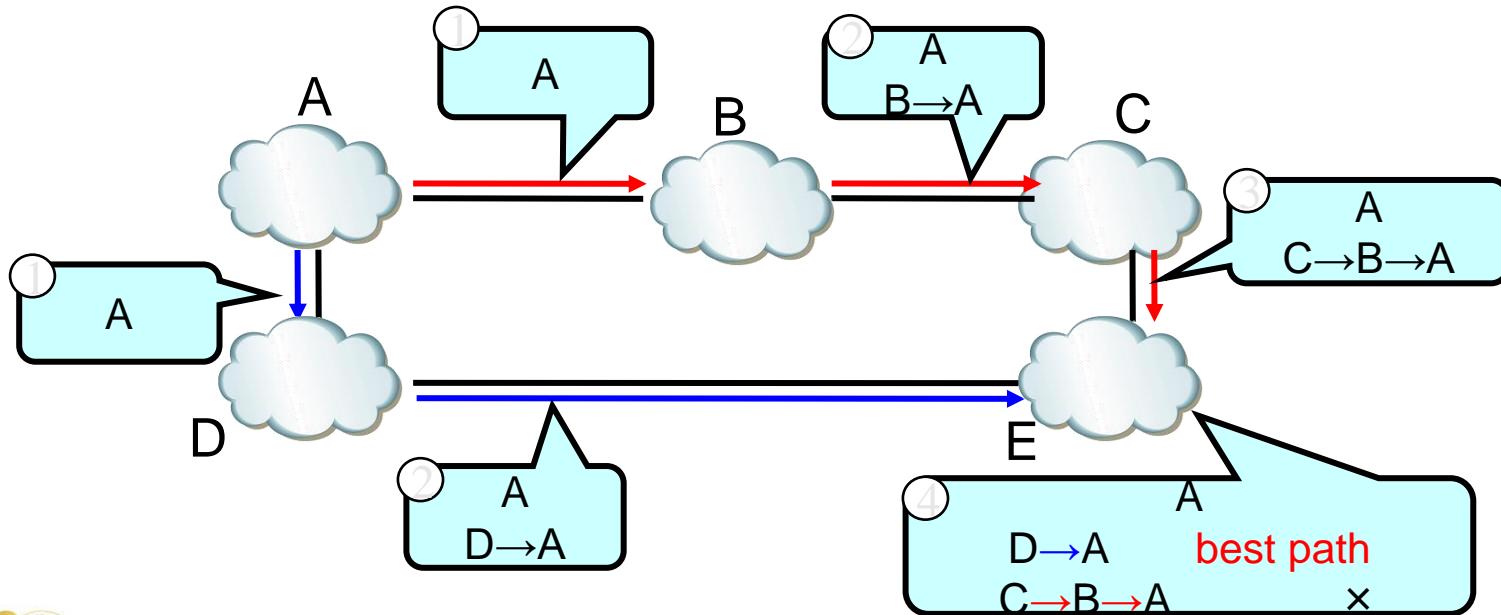
BGP cơ bản

- **BGP phiên:** hai BGP router (“peers”) trao đổi thông điệp BGP thông qua một kết nối TCP bán vĩnh cửu :
 - Quảng bá “con đường” tới những địa chỉ mạng khác dựa vào prefix (BGP là một giao thức vector)
- khi AS3 gateway 3a quảng bá **đường AS3,X** tới AS2 gateway 2c:
 - AS3 *hứa* với AS2 nó chuyển thông điệp tới X



BGP: Path vector routing

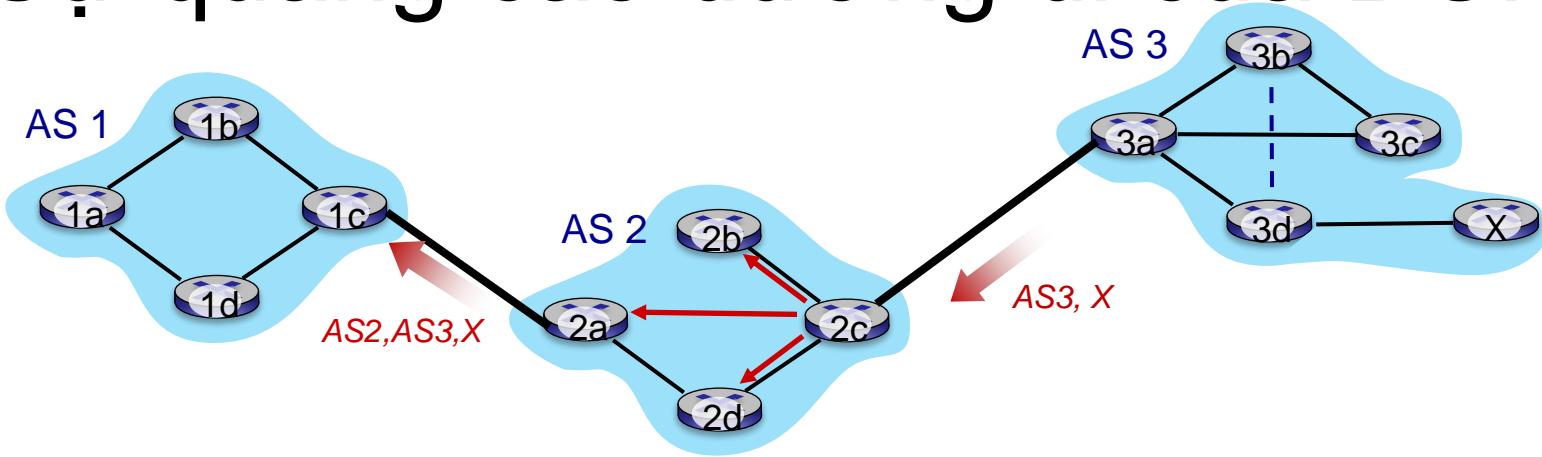
- Giữa các AS nên dùng giao thức nào?
 - Khó có một chính sách và đơn vị chi phí chung
 - LS: Chi phí không đồng nhất, CSDL quá lớn
 - DV: Mạng quá rộng, khó hội tụ
- Giải pháp: Chọn đường theo path-vector



Thuộc tính “đường đi” và đường đi BGP

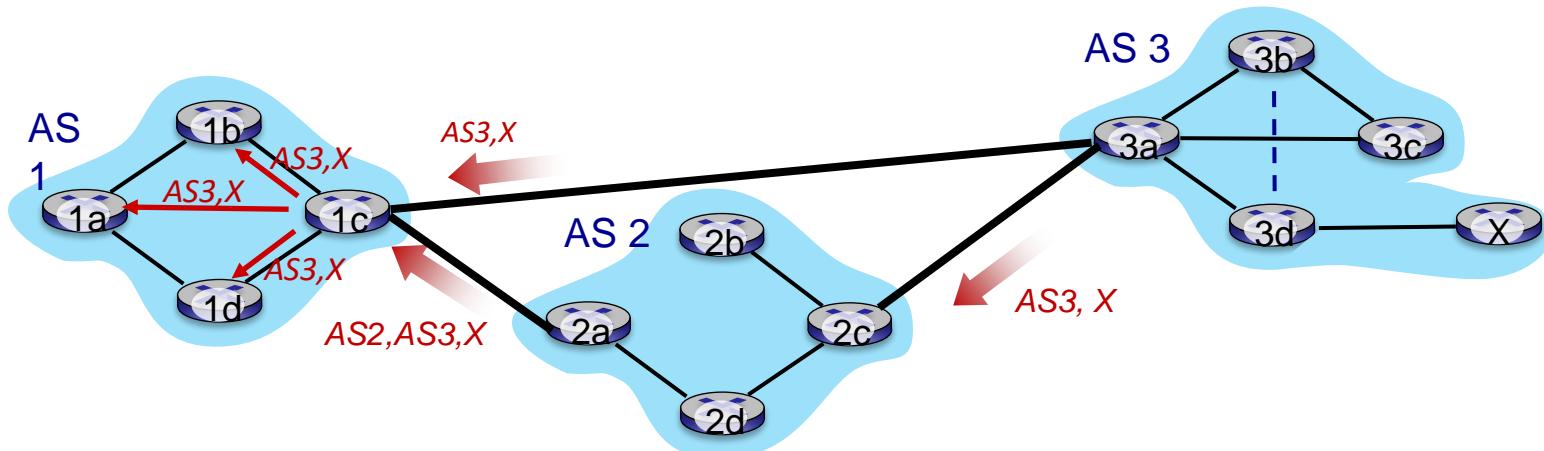
- Đường đi “quảng cáo” của BGP: tiền tố (prefix) + thuộc tính
 - prefix: đích đến được quảng cáo
 - Hai thuộc tính quan trọng:
 - AS-PATH: danh sách các AS mà prefix đã đi qua
 - NEXT-HOP: xác định router nội vùng cụ thể tới AS tiếp theo.
- Định tuyến theo chính sách:
 - Gateway router nhận thông tin quảng bá con đường dùng “chính sách nhập khẩu” (import policy) để chấp nhận hay từ chối đường đi đấy (Ví dụ: không bao giờ đi qua AS Y)
 - Chính sách AS cũng xác định việc quảng bá con đường đến các AS hàng xóm hay không

Sự quảng cáo đường đi của BGP



- Router 2c nhận sự quảng cáo đường đi **AS3,X** (via eBGP) từ AS3 router 3a
- Dựa trên chính sách của AS2, router 2c chấp nhận đường đi AS3,X, lan truyền (thông qua iBGP) tới tất cả các router của AS2
- Dựa trên chính sách của AS2, 2a quảng cáo (qua eBGP) con đường **AS2, AS3, X** tới router 1c của AS1

Sự quảng cáo đường đi của BGP

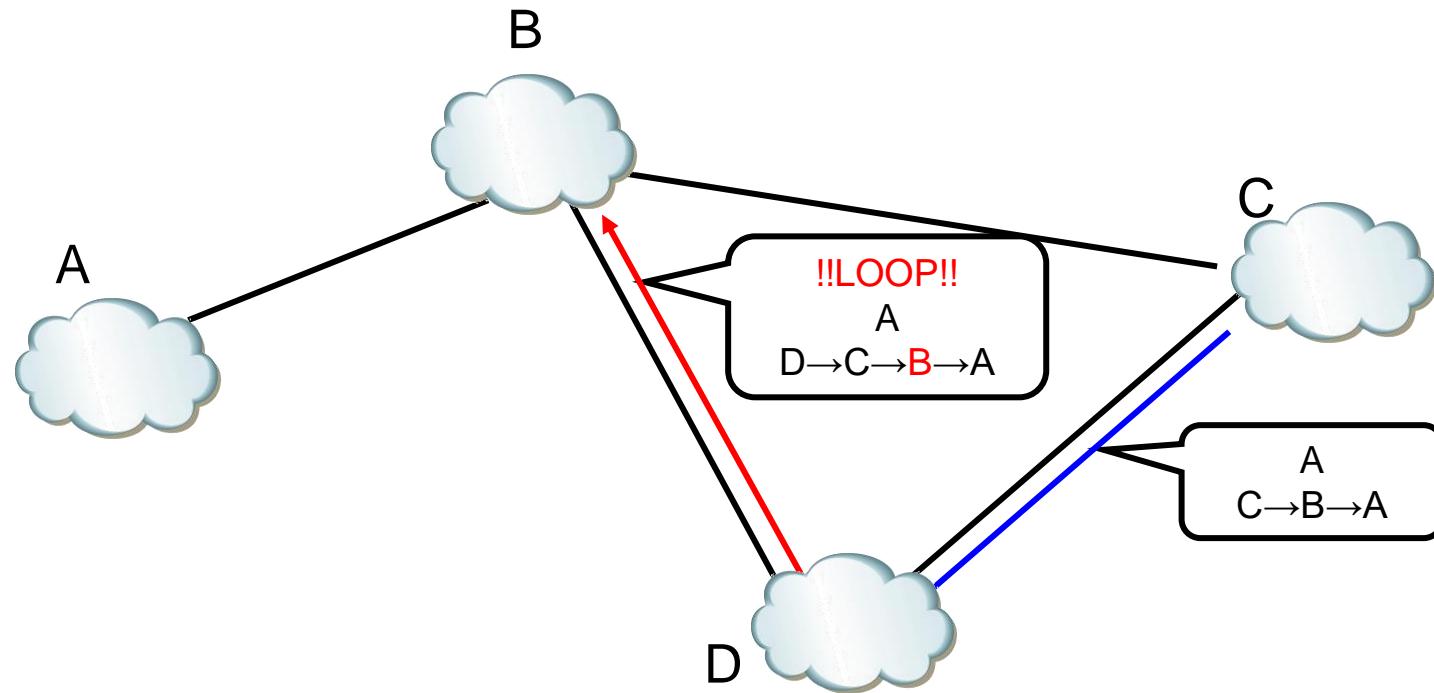


gateway router có thể học **nhiều** con đường đến đích

- Gateway router 1c của AS1 học con đường *AS2,AS3,X* từ 2a
- Gateway router 1c của AS1 học con đường *AS3,X* từ 3a
- Theo *chính sách*, gateway router 1c của AS1 chọn con đường *AS3,X* và quảng cáo con đường đó đến tất cả các router trong AS1 theo iBGP

Cơ chế tránh vòng lặp

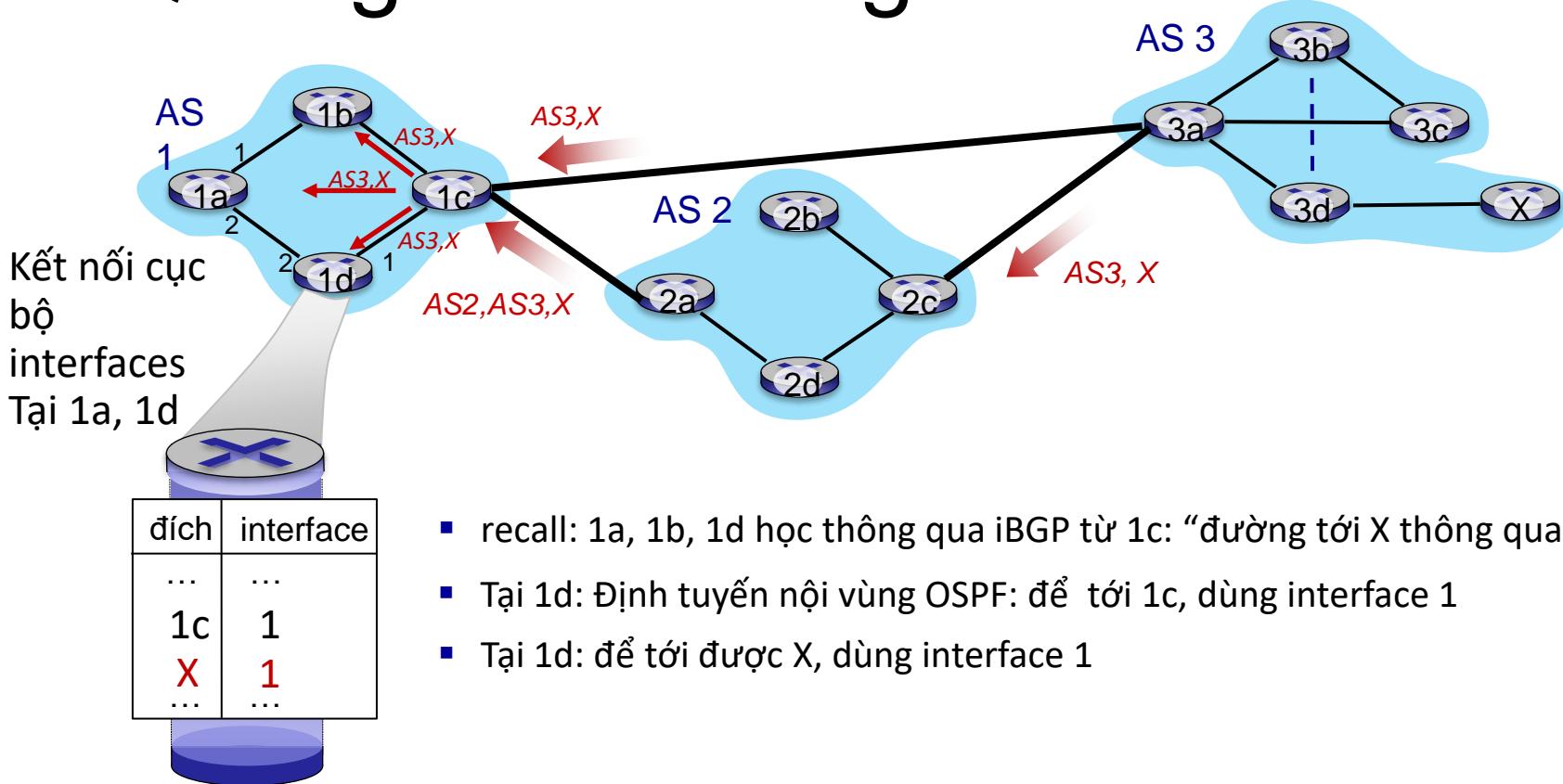
- Dò lại xem router đã có trên path-vector hay chưa
 - B hủy đường đi tới A



Thông điệp BGP

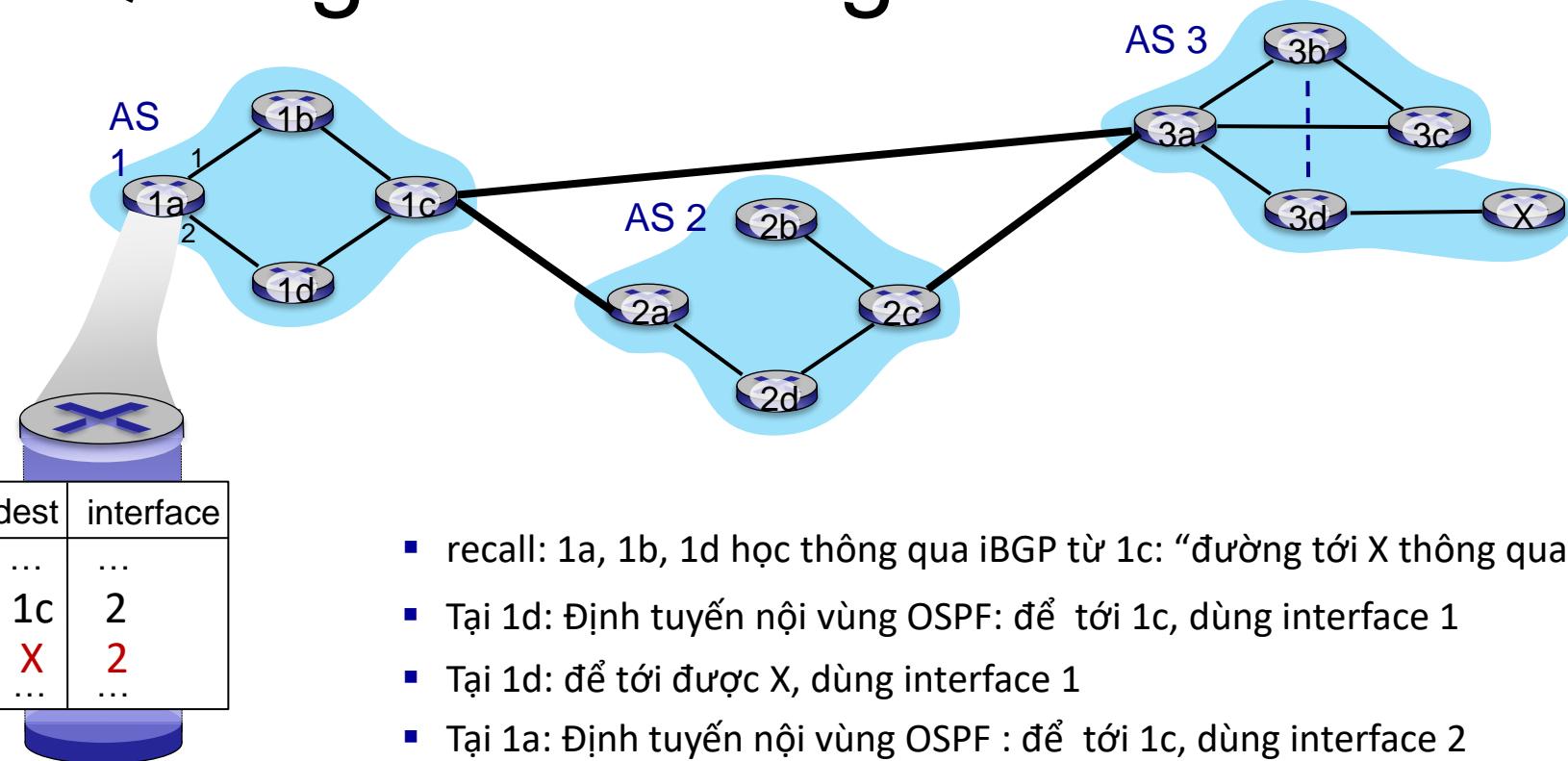
- Thông điệp BGP trao đổi giữa các peer thông qua kết nối TCP
- Thông điệp BGP:
 - **OPEN**: mở kết nối TCP để kết nối đến BGP peer và xác thực peer gửi thông tin
 - **UPDATE**: quảng cáo đường đi mới (hoặc bỏ đường đi cũ)
 - **KEEPALIVE**: duy trì kết nối trong trường hợp không có thông điệp UPDATE, cũng là ACK thông điệp OPEN
 - **NOTIFICATION**: báo cáo lỗi trong thông điệp trước đó, cũng được dùng để đóng kết nối

Quảng cáo đường đi của BGP

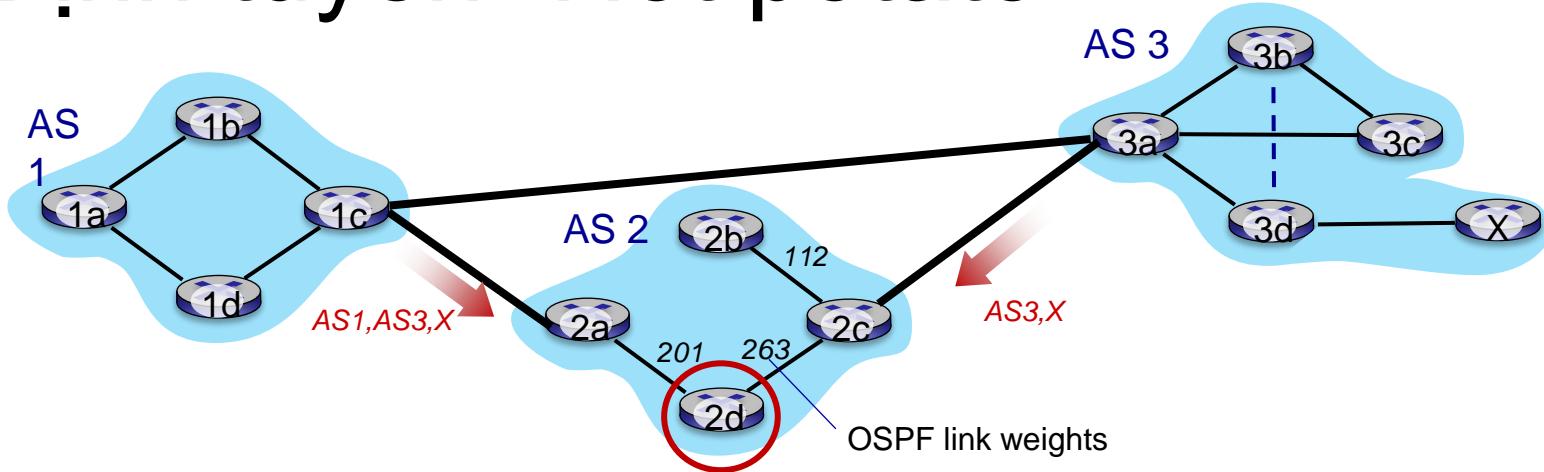


- recall: 1a, 1b, 1d học thông qua iBGP từ 1c: “đường tới X thông qua 1c”
- Tại 1d: Định tuyến nội vùng OSPF: để tới 1c, dùng interface 1
- Tại 1d: để tới được X, dùng interface 1

Quảng cáo đường đi của BGP



Định tuyến “Hot potato”



- 2d học (qua iBGP) rằng nó có thể định tuyến đến X qua 2a hoặc 2c
- **Định tuyến “Hot potato”**: chọn router gateway có giá nội vùng nhỏ nhất (Ví dụ: 2d chọn 2a dù phải đi qua nhiều AS hơn để đến đích) và không cần quan tâm đến giá liên miền.

Tại sao dùng định tuyến nội vùng và liên vùng?

Chính sách:

- inter-AS: người quản trị muốn điều khiển việc định tuyến gói tin, ai được phép định tuyến qua mạng
- intra-AS: cùng người quản trị, ít vấn đề chính sách hơn

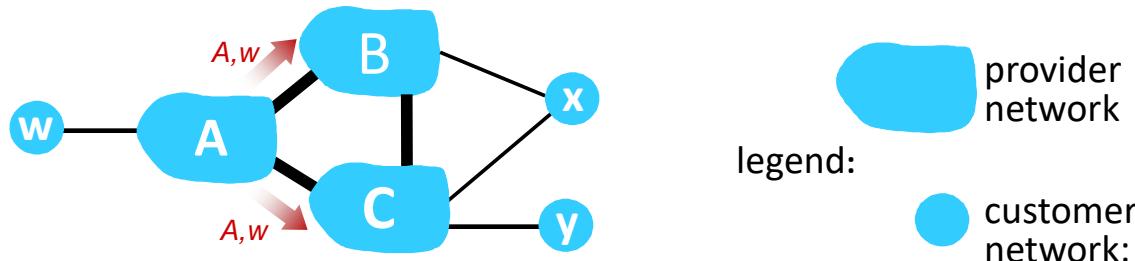
Kích thước:

- Định tuyến phân cấp giảm kích thước bảng định tuyến, giảm việc cập nhập

Hiệu năng:

- intra-AS: tập trung vào hiệu năng
- inter-AS: chính sách quan trọng hơn hiệu năng

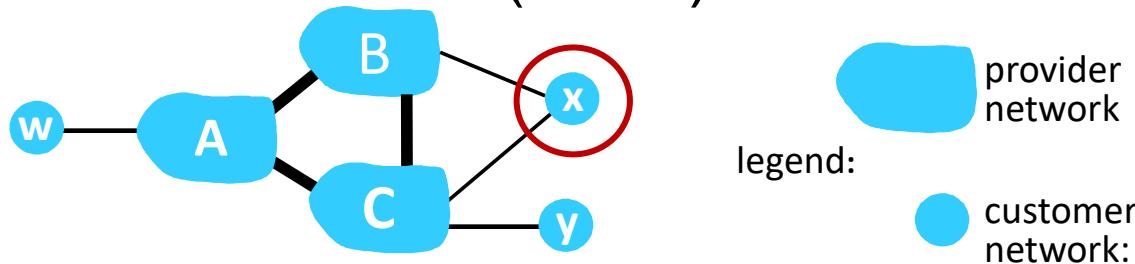
BGP: achieving policy via advertisements



ISP chỉ muốn định tuyến tới/từ các mạng khách hàng bản thân (không muốn tải lưu lượng từ khách hàng các ISP khác – thực tế)

- A quảng cáo đường đi Aw tới B và tới C
- B *lựa chọn không quảng cáo* Baw tới C
- B không nhận được “doanh thu” cho đường đi CBAw, vì cả C,A và w không phải là khách hàng của B
 - C không học được con đường CBAw
- C sẽ định tuyến qua Caw (không dùng B) để tới w

BGP: achieving policy via advertisements (more)



ISP chỉ muốn định tuyến tới/từ các mạng khách hàng bản thân (không muốn tải lưu lượng từ khách hàng các ISP khác – thực tế)

- A,B,C là *những người cung cấp mạng* (Ex: ISP)
- x,w,y là **khách hàng**
- x là **khách hàng kép (dual-homed)**: kết nối với 2 mạng
- **Thực thi chính sách**: x không muốn định tuyến từ B tới C qua x
 - .. nên x không quảng cáo tới B con đường tới C

BGP có thể cài đặt các policy

- Khi các router gửi và nhận thông tin chọn đường router BGP có thể áp đặt các chính sách
 - Cho đường vào: chọn lọc đường đi sẽ tiếp nhận
 - Cho đường ra: chọn lọc đường đi sẽ quảng bá

Các thuộc tính của đường đi

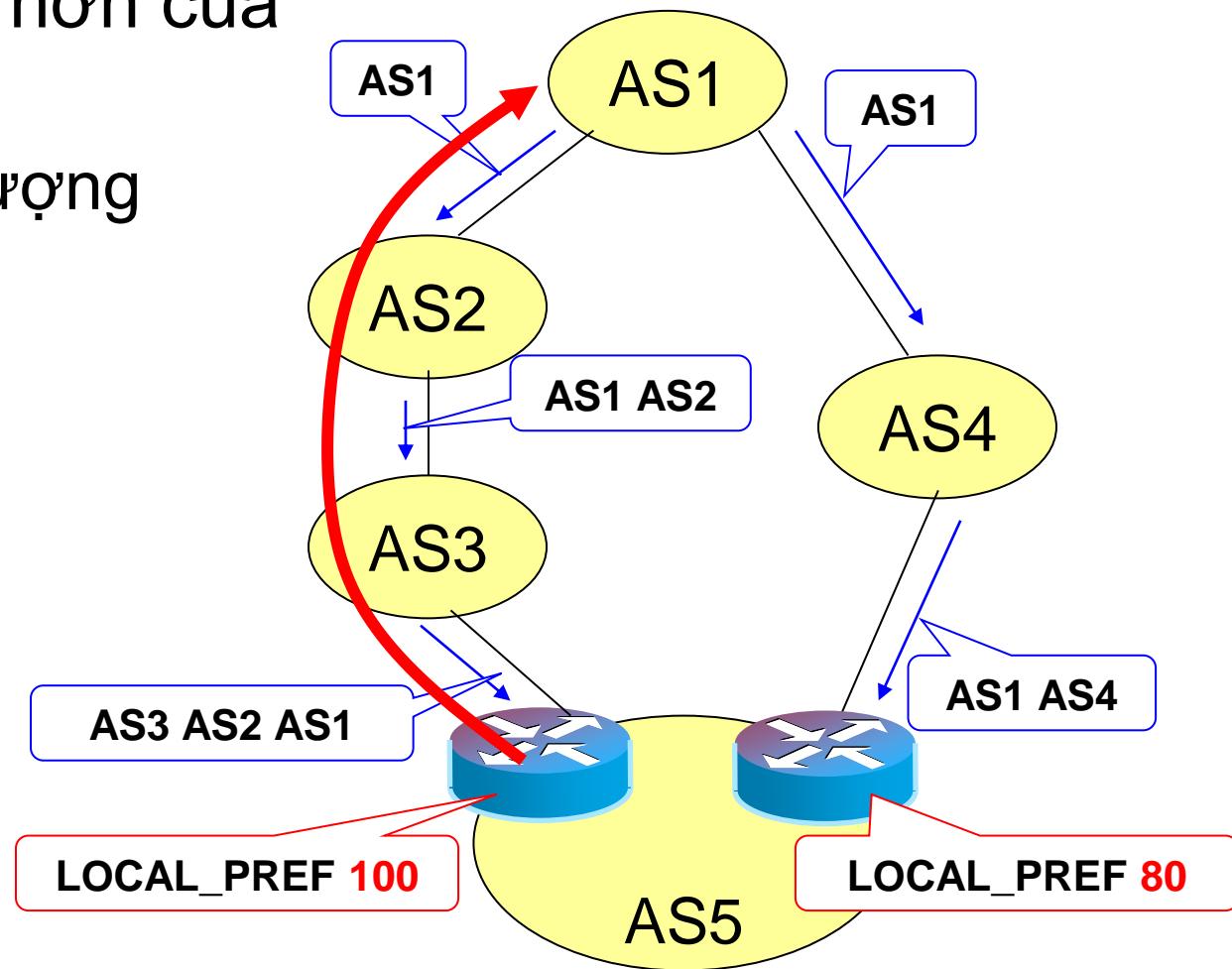
- ORIGIN
 - Nguồn của thông tin (IGP/EGP/incomplete)
- AS_PATH
- NEXT_HOP
- MED (MULTI_EXIT_DISCRIMINATOR)
- LOCAL_PREF
- ATOMIC_AGGREGATE
- AGGREGATOR
- COMMUNITY

Các bước chọn đường đi

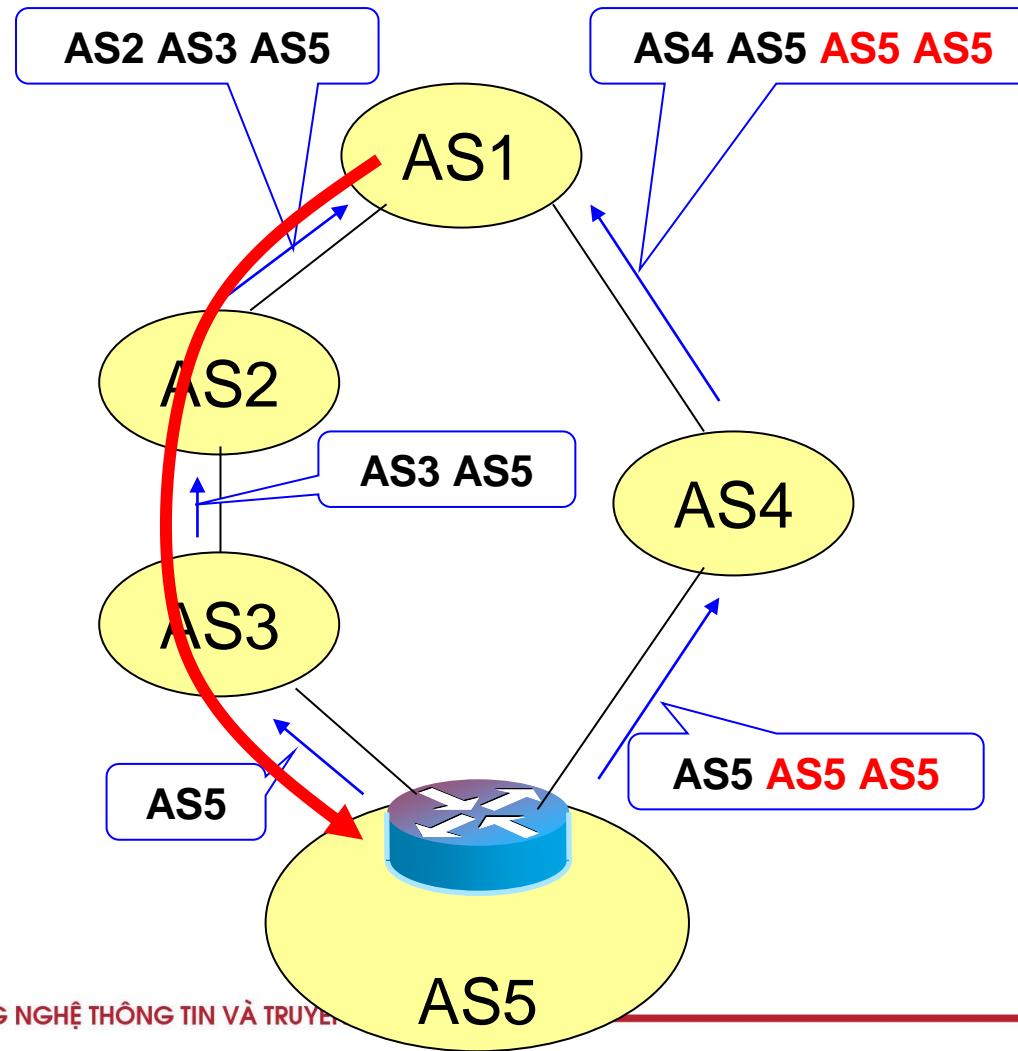
- Bước 1: So sánh LOCAL_PREF
- Bước 2: So sánh độ dài AS_PATH
- Bước 3: So sánh ORIGIN
- Bước 4: So sánh MED
- Bước 5: So sánh EBGP/IBGP
- Bước 6: So sánh chi phí tới NEXT_HOP
- Bước 7: So sánh Router ID

Sử dụng LOCAL_PREF

- Chọn giá trị lớn hơn của LOCAL_PREF
- Điều khiển lưu lượng upbound



Chọn đường với AS_PATH Prepend



Ví dụ về AS PATH

Network	Next Hop	Metric	LocPrf	Weight	Path
4.79.201.0/26	203.178.136.29	700	500	0	7660 22388 11537 10886 40220
	203.178.136.29	700	500	0	7660 22388 11537 10886 40220
	203.178.136.29	700	500	0	7660 22388 11537 10886 40220
6.1.0.0/16	203.178.136.29	700	500	0	7660 22388 11537 668
	203.178.136.29	700	500	0	7660 22388 11537 668
	203.178.136.29	700	500	0	7660 22388 11537 668
6.2.0.0/22	203.178.136.29	700	500	0	7660 22388 11537 668
	203.178.136.29	700	500	0	7660 22388 11537 668

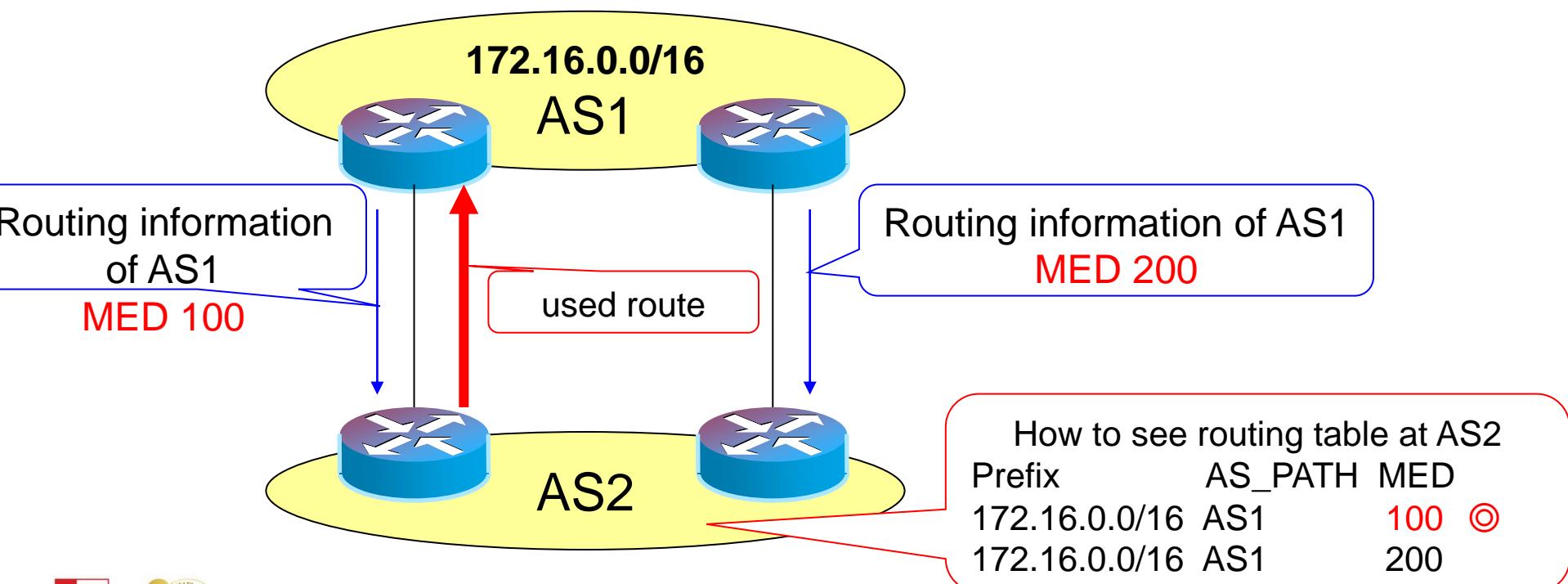
Ví dụ về AS PATH prepend

Network	Next Hop	Metric	LocPrf	Weight	Path
8.5.192.0/22	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
8.5.196.0/24	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
8.5.200.0/22	203.178.136.14	100	0	2516	209 13989 13989 13989 13989
	203.178.136.14	100	0	2516	209 13989 13989 13989 13989

Một số AS được lặp lại trên đường đi để làm đường
đi dài hơn và sẽ không được ưu tiên chọn

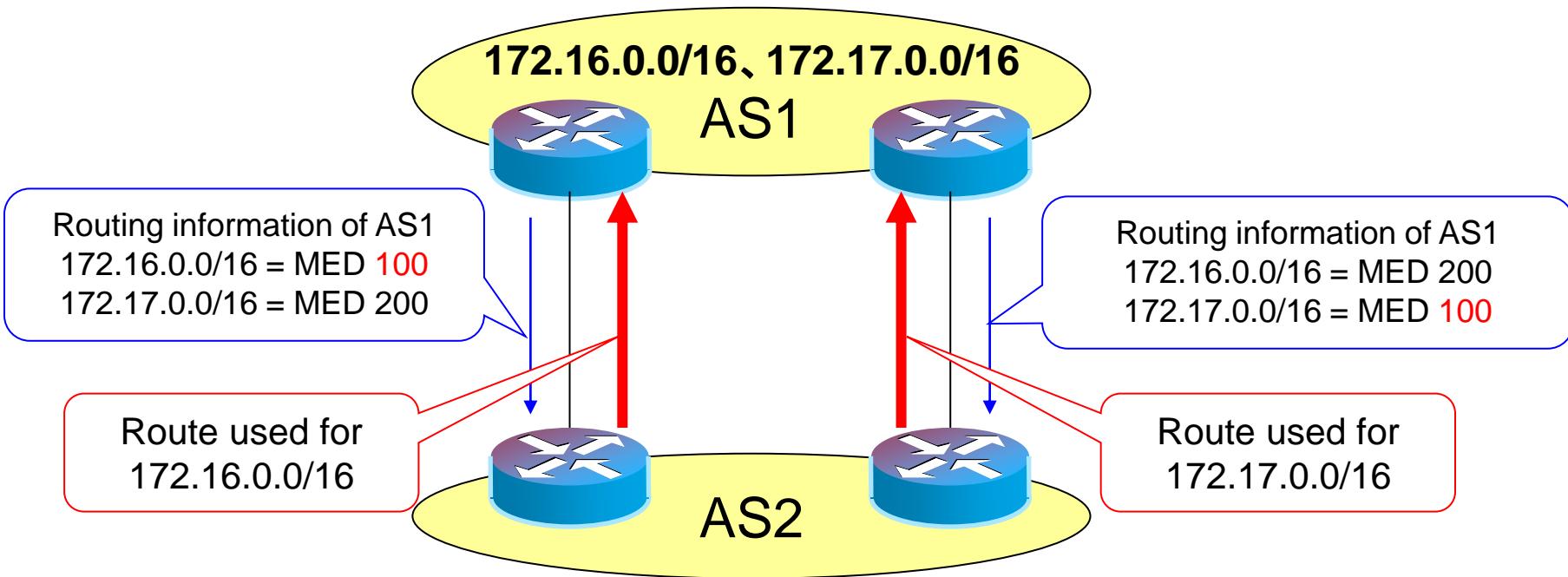
Chọn đường với MED

- Trong trường hợp 2 AS với nhiều link
- Chọn MED nhỏ hơn
- Áp dụng trong điều khiển lưu lượng



Phân tải với MED

- Đặt giá trị MED khác nhau cho mỗi đường
- Cũng điều khiển lưu lượng





25
YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you for
your attentions!**





ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chương 5: Tầng giao vận

Đọc trước: Chapter 5-Computer Networks,
Tanenbaum

Tổng quan

- Các tuần trước : Giao thức IP
 - Địa chỉ, gói tin IP
 - ICMP
 - Chọn đường
- Hôm nay: Tầng giao vận
 - Nguyên lý tầng giao vận
 - Giao thức UDP
 - Giao thức TCP



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Các khái niệm cơ bản

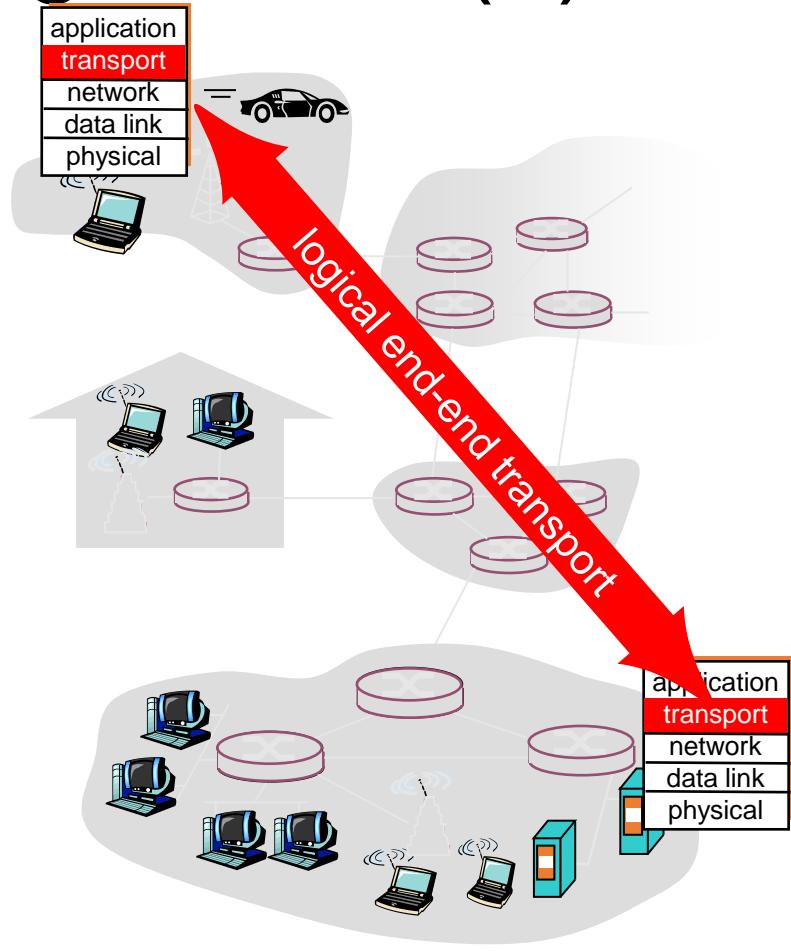
Vị trí trong kiến trúc phân tầng
Hướng liên kết vs. Không liên kết
UDP & TCP

Vị trí trong kiến trúc phân tầng

Application (HTTP, Mail, ...)	Hỗ trợ các ứng dụng trên mạng
Transport (UDP, TCP ...)	Truyền dữ liệu giữa các ứng dụng
Network (IP, ICMP ...)	Chọn đường và chuyển tiếp gói tin giữa các máy, các mạng
Datalink (Ethernet, ADSL...)	Hỗ trợ việc truyền thông cho các thành phần kế tiếp trên cùng 1 mạng
Physical (bits...)	Truyền và nhận dòng bit trên đường truyền vật lý

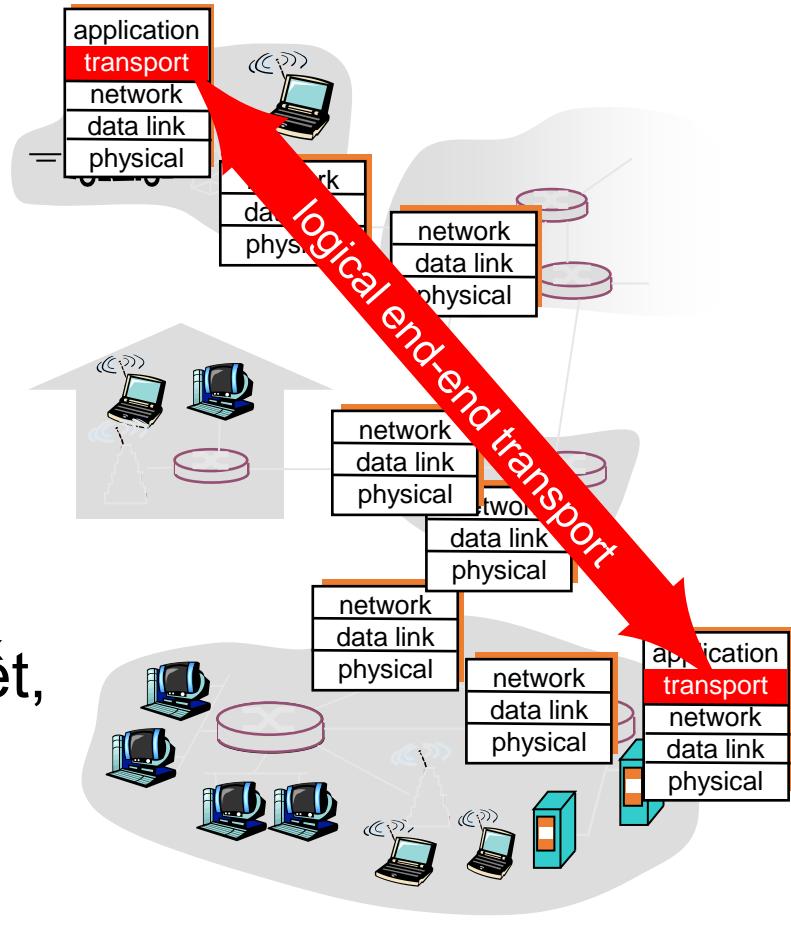
Tổng quan về tầng giao vận (1)

- Cung cấp phương tiện truyền giữa các ứng dụng cuối
 - Các ứng dụng là các tiến trình chạy trên các máy.
- Bên gửi:
 - Nhận dữ liệu từ ứng dụng
 - Đặt dữ liệu vào các đoạn tin và chuyển cho tầng mạng
 - Nếu dữ liệu quá lớn, nó sẽ được chia làm nhiều phần và đặt vào nhiều đoạn tin khác nhau
- Bên nhận:
 - Nhận các đoạn tin từ tầng mạng
 - Tập hợp dữ liệu và chuyển lên cho ứng dụng



Tổng quan về tầng giao vận (2)

- Được cài đặt trên các hệ thống cuối
 - Không cài đặt trên các routers, switches...
- Hai dạng dịch vụ giao vận
 - Tin cậy, hướng liên kết, e.g TCP
 - Không tin cậy, không liên kết, e.g. UDP



Tại sao lại cần 2 loại dịch vụ?

- Các yêu cầu đến từ tầng ứng dụng là đa dạng
- Các ứng dụng cần dịch vụ với 100% độ tin cậy như mail, web...
 - Sử dụng dịch vụ của TCP
- Các ứng dụng cần chuyển dữ liệu nhanh, có khả năng chịu lỗi, e.g. VoIP, Video Streaming
 - Sử dụng dịch vụ của UDP

Ứng dụng và dịch vụ giao vận

Ứng dụng	Giao thức ứng dụng	Giao thức giao vận
e-mail	SMTP	TCP
remote terminal access	Telnet	TCP
Web	HTTP	TCP
file transfer	FTP	TCP
streaming multimedia	giao thức riêng (e.g. RealNetworks)	TCP or UDP
Internet telephony	giao thức riêng (e.g., Vonage, Dialpad)	thường là UDP

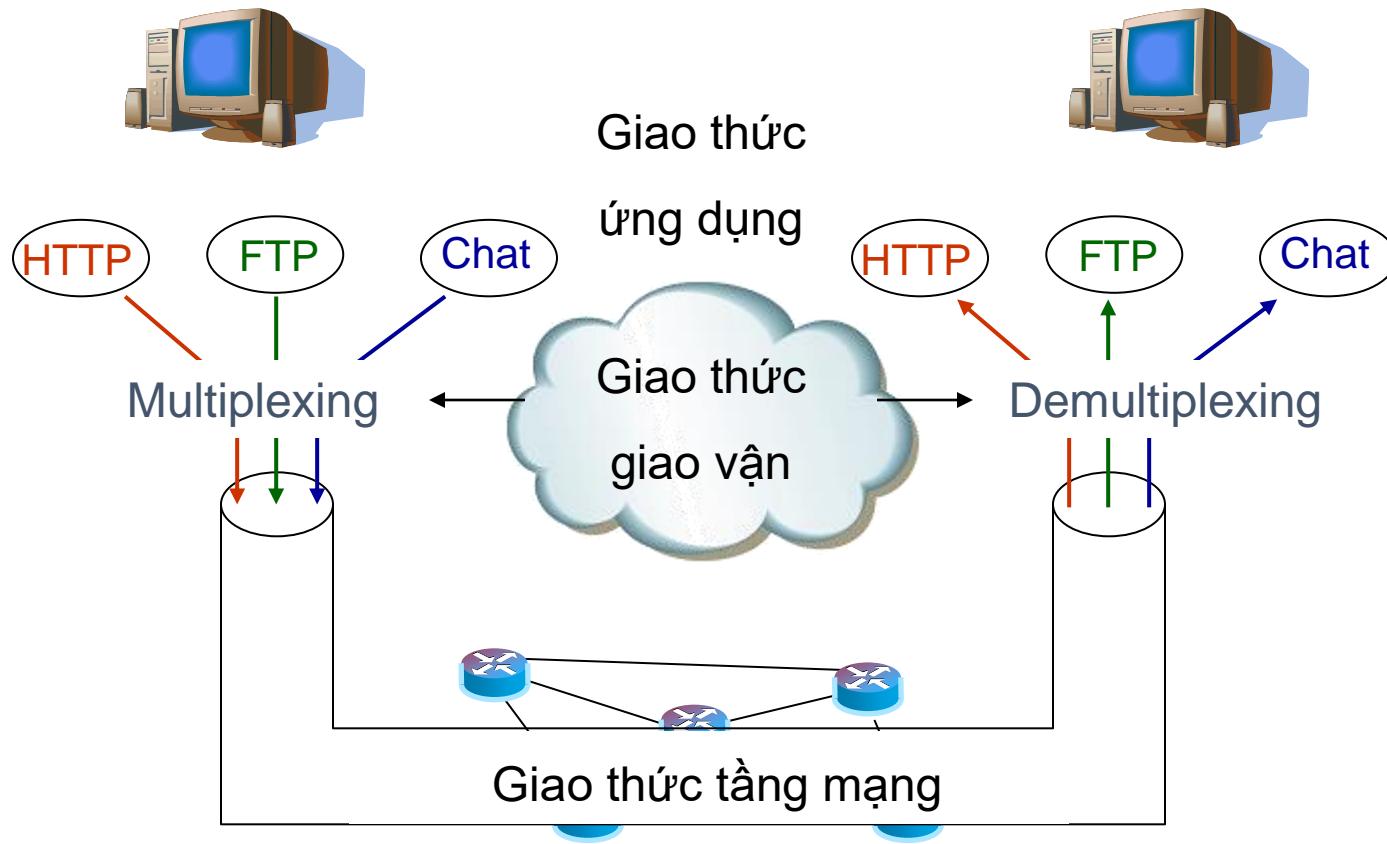


ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Các chức năng chung

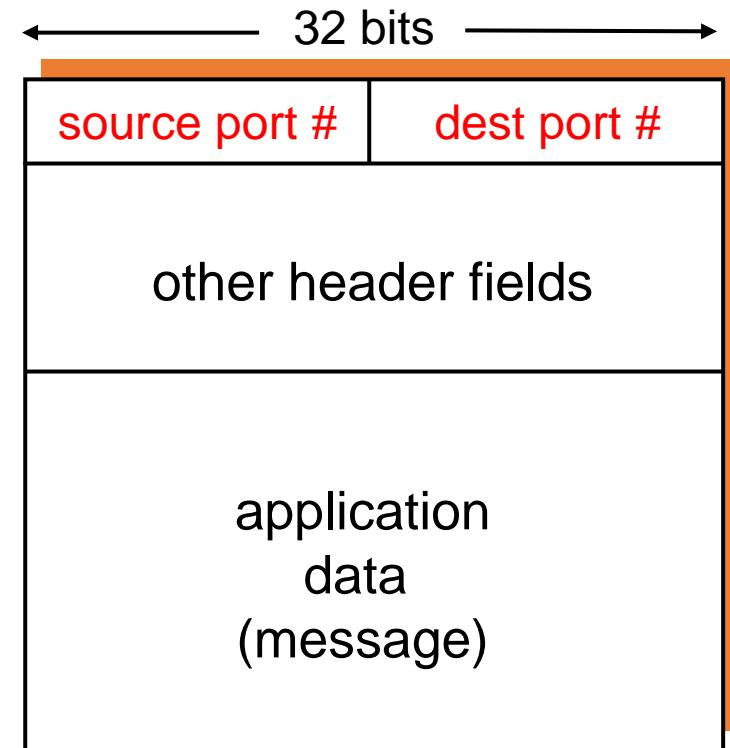
Dồn kênh/phân kênh
Kiểm soát lỗi

Dồn kênh/phân kênh - Mux/Demux



Mux/Demux hoạt động ntn?

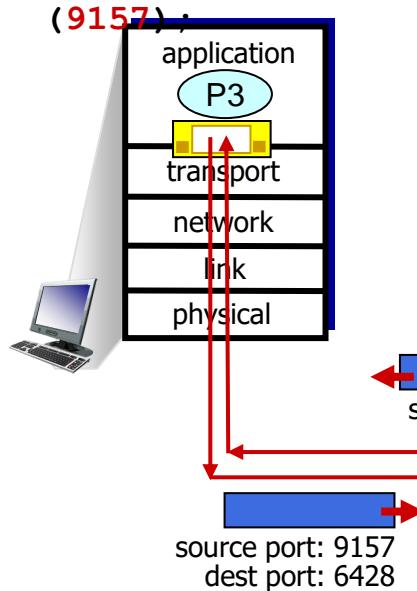
- Tại tầng mạng, gói tin IP được định danh bởi địa chỉ IP
 - Để xác định máy trạm
- Làm thế nào để phân biệt các ứng dụng trên cùng một máy?
 - Sử dụng số hiệu cổng (16 bits)
 - Mỗi trình duyệt ứng dụng được gán 1 cổng
- **Socket:** Một cặp địa chỉ IP và số hiệu cổng



TCP/UDP segment format

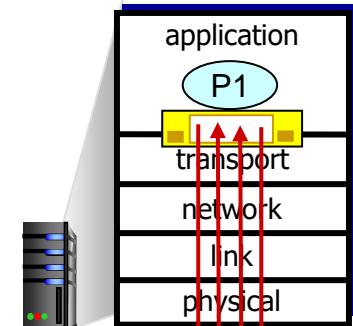
Demultiplexing không kết nối: ví dụ

```
DatagramSocket mySocket2  
= new DatagramSocket  
(9157);
```

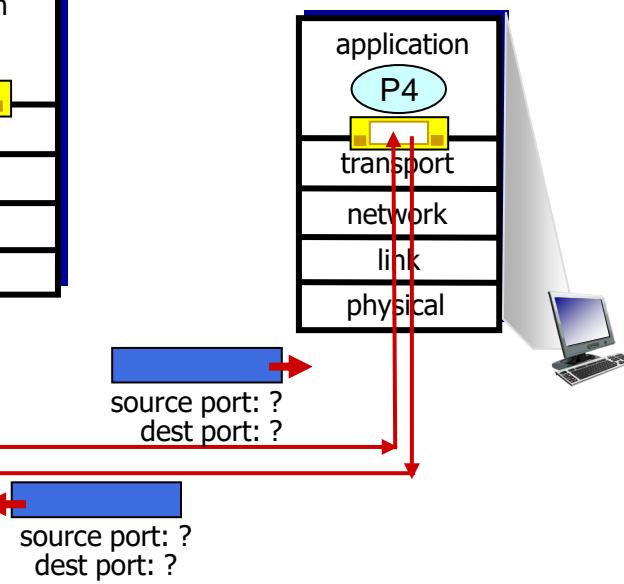


DatagramSocket

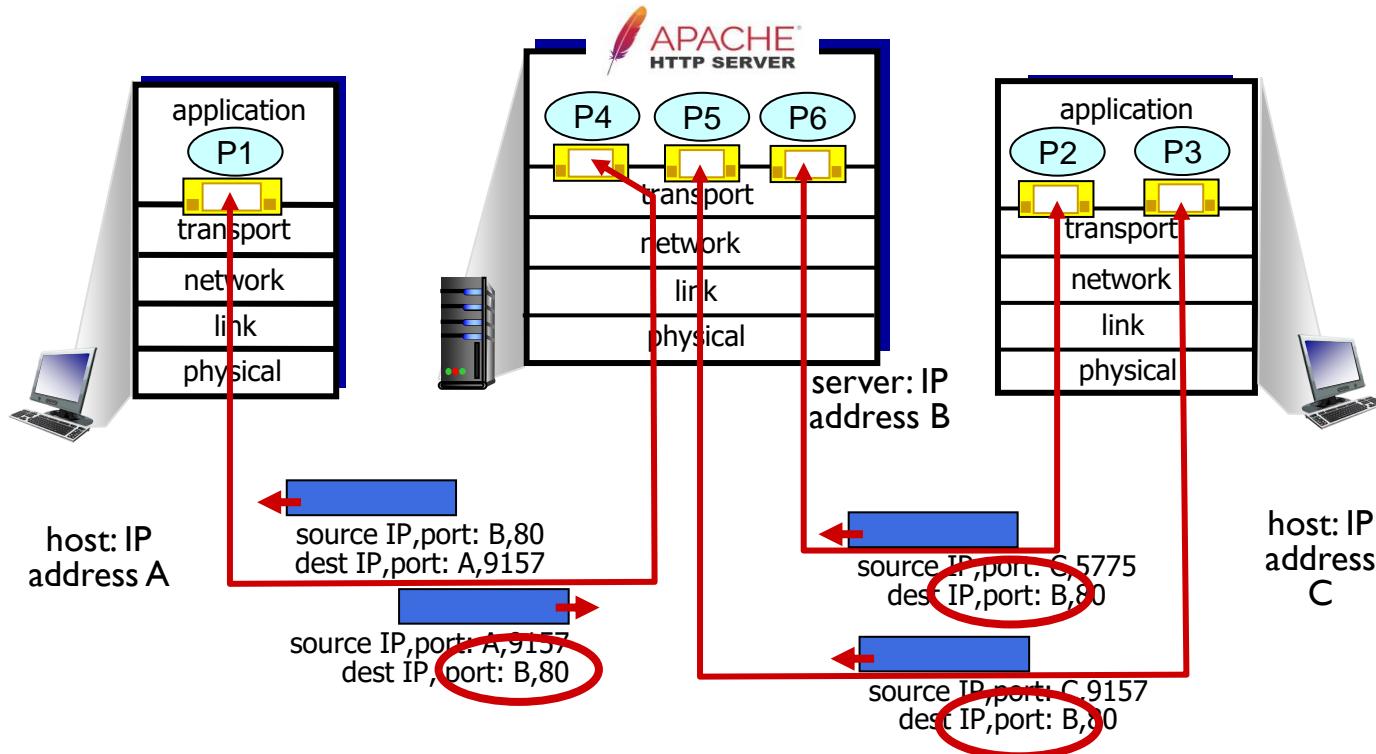
```
serverSocket = new  
DatagramSocket  
(6428);
```



```
DatagramSocket mySocket1 =  
new DatagramSocket (5775);
```



Demultiplexing hướng kết nối: ví dụ



Ba gói tin, cùng tới địa chỉ IP B, cổng 80
được chia thành các socket khác nhau

Kiểm soát lỗi

- Sử dụng CRC hoặc Checksum
- Checksum
 - Phát hiện lỗi bit trong các đoạn tin/gói tin
 - Nguyên lý giống như checksum (16 bits) của giao thức IP
- Nguyên lý checksum
 - Dữ liệu cần gửi được chia thành các đoạn bằng nhau
 - Các đoạn được tính tổng với nhau, nếu có nhó thì cộng giá trị nhó vào tổng
 - Đảo tổng thu được checksum

Checksum: Ví dụ

Partial Sum:	$1\ 01100111$
	$+ 1$
	<hr/>
	01110111
Frame 3:	$+ 11110000$
	<hr/>
Partial Sum:	$1\ 01100111$
	$+ 1$
	<hr/>
	01101000
Frame 4:	$+ 11000011$
	<hr/>
Partial Sum:	$1\ 00101011$
	$+ 1$
	<hr/>
Sum:	00101100
	<hr/>
Checksum:	11010011

Partial Sum:	$1\ 01100111$
	$+ 1$
	<hr/>
	01110111
Frame 3:	$+ 11110000$
	<hr/>
Partial Sum:	$1\ 01100111$
	$+ 1$
	<hr/>
	01101000
Frame 4:	$+ 11000011$
	<hr/>
Partial Sum:	$1\ 00101011$
	$+ 1$
	<hr/>
Sum:	00101100
	<hr/>
Checksum:	11010011



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Truyền thông tin cậy tại tầng giao vận

ARQ

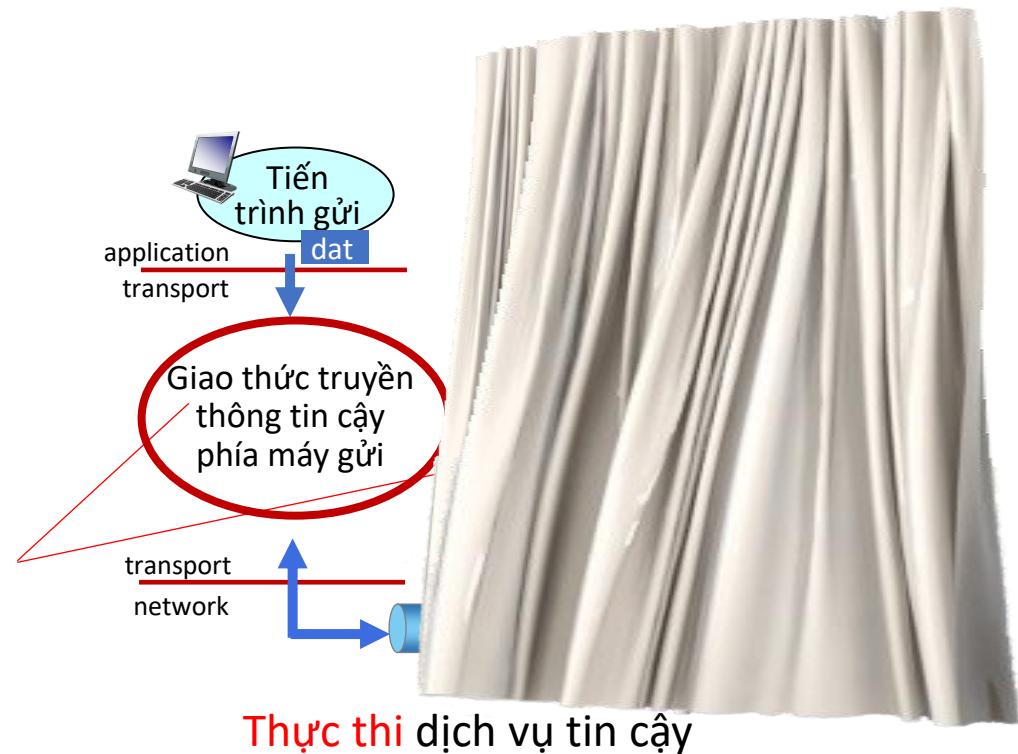
Stop-and-wait

Sliding windows

Nguyên lý của truyền thông tin cậy

Máy gửi và nhận không biết “trạng thái” của nhau (Vd: một gói tin đã nhận được chưa)

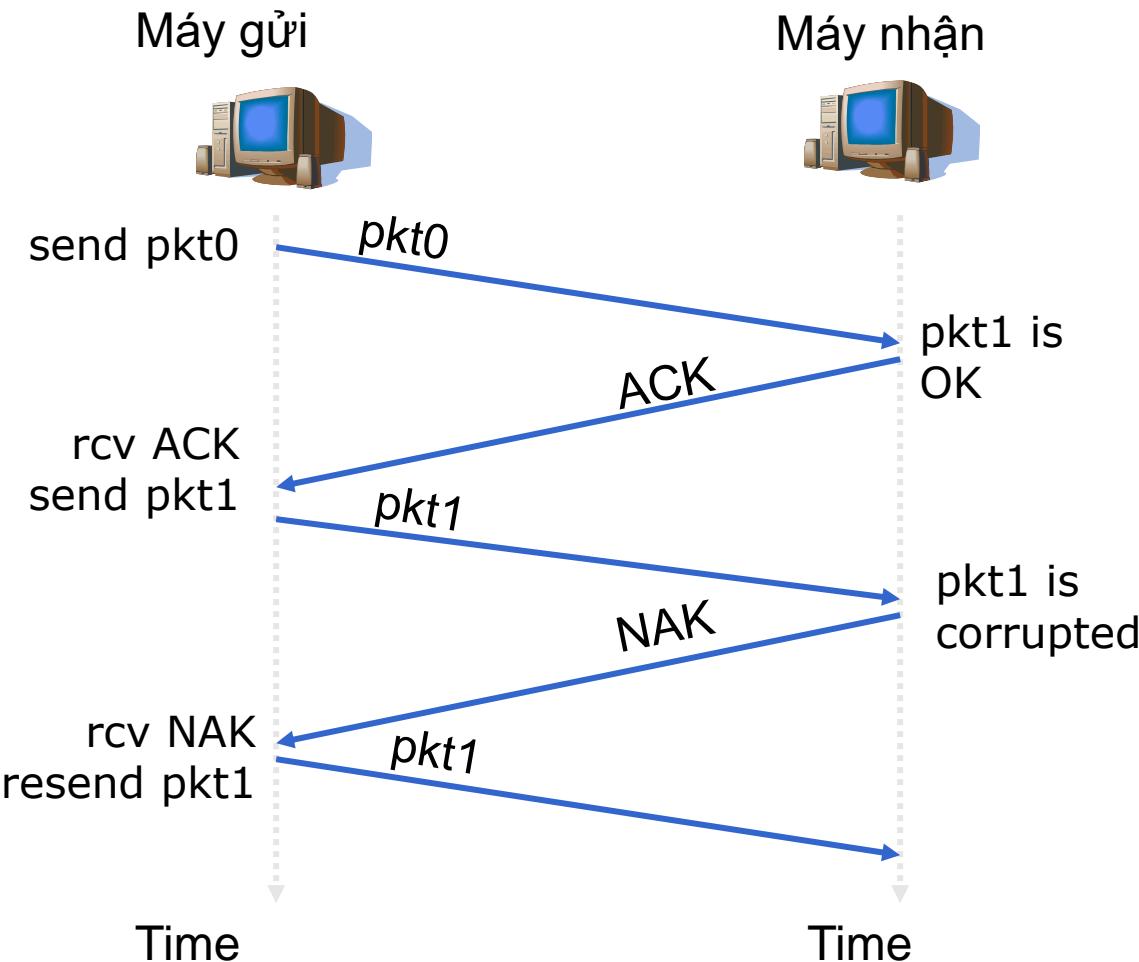
- Trừ khi giao tiếp qua 1 thông điệp



Truyền thông tin cây

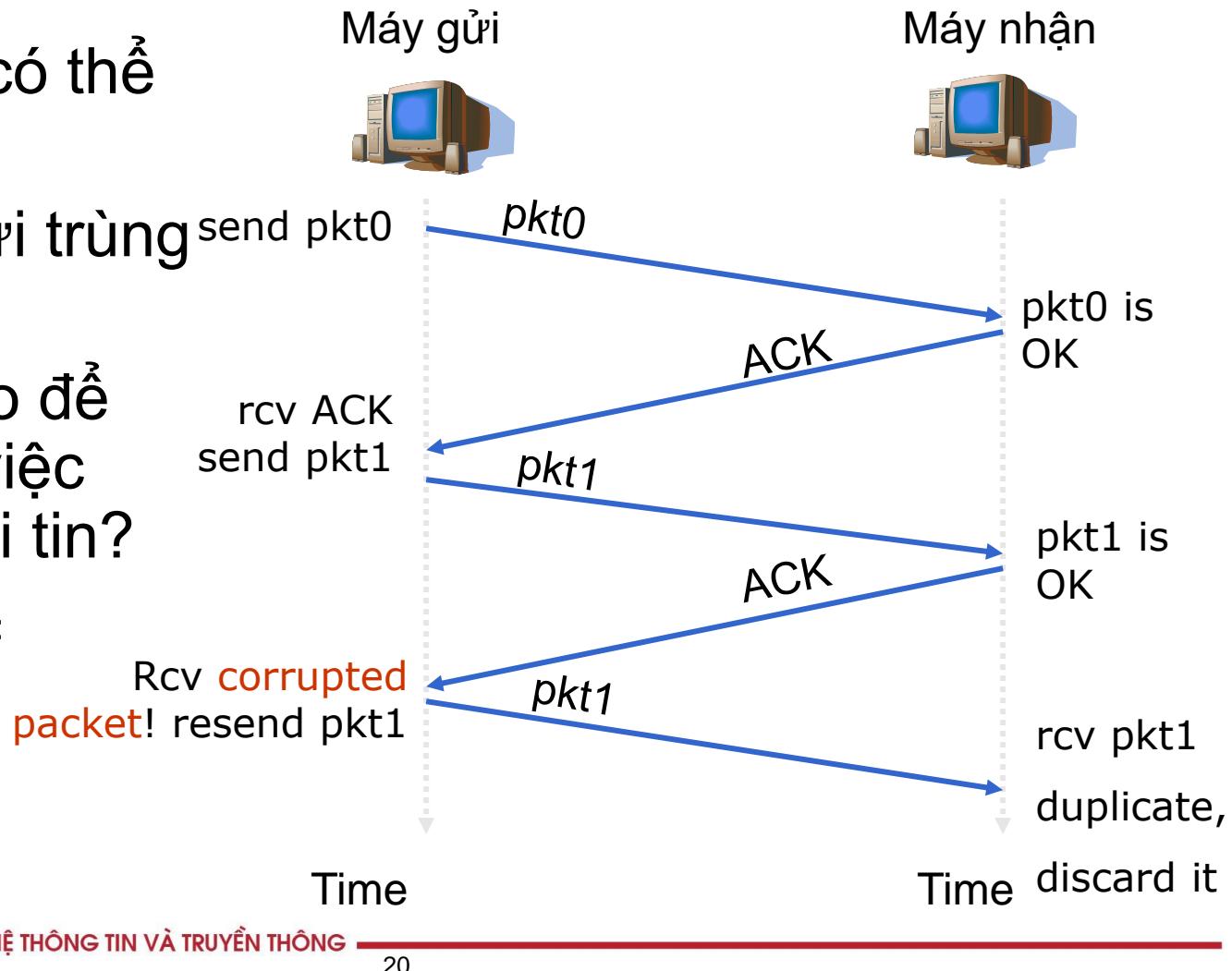
- Làm sao phát hiện lỗi?
 - Checksum
- Làm sao báo cho bên gửi?
 - ACK (*acknowledgements*):
 - NAK (*negative acknowledgements*)
- Phản ứng của bên gửi?
 - Gửi lại gói tin bị lỗi khi nhận được NAK

Điều khiển lỗi

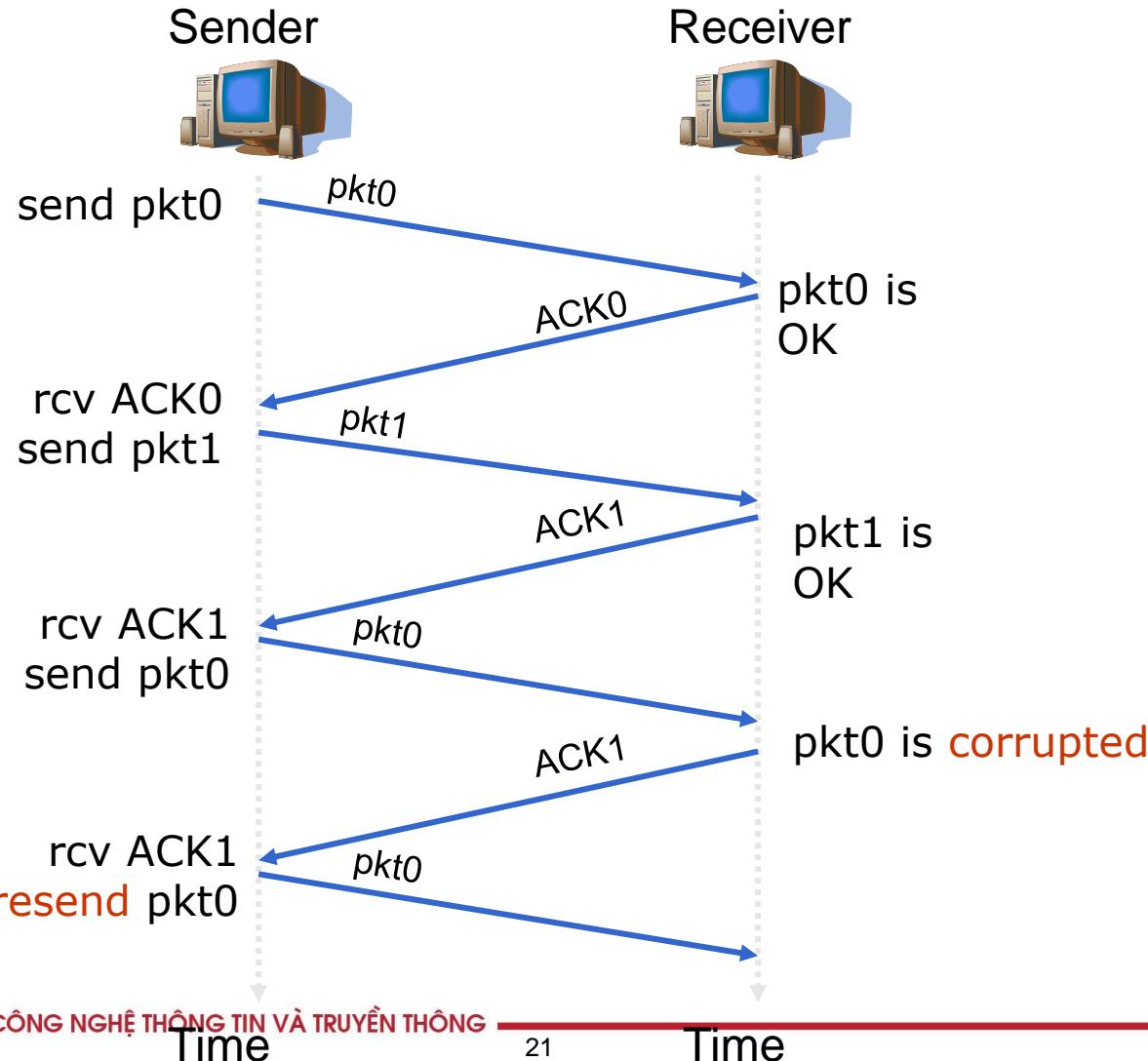


Gói tin ACK/NAK bị lỗi

- ACK/ NAK có thể bị lỗi
- Gói tin bị gửi trùng lặp
- Làm thế nào để khắc phục việc trùng lặp gói tin?
- Dùng Seq.#



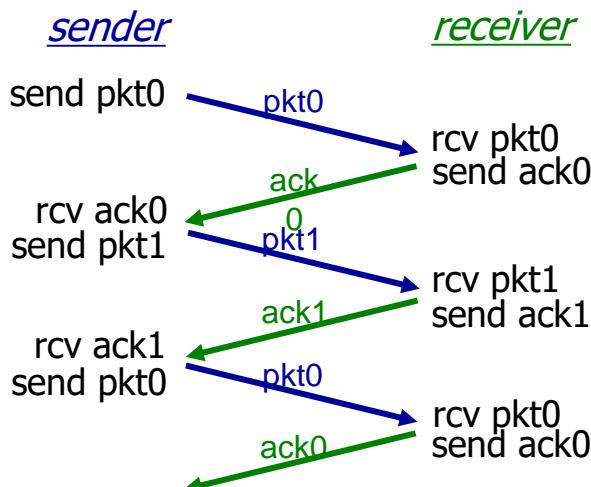
Điều khiển lỗi không cần NAK



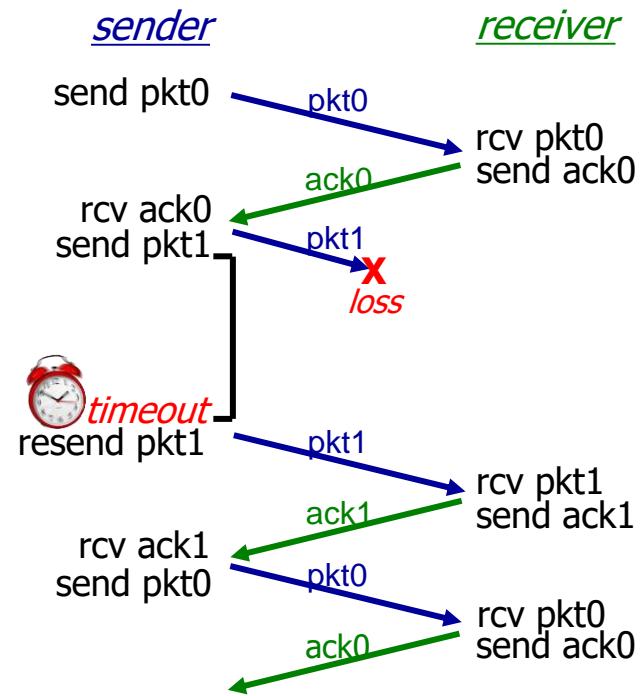
Chanel with error and packet lost

- Dữ liệu và gói tin ACK có thể bị mất trên đường truyền
 - Không nhận được ACK, làm sao máy gửi biết và gửi lại gói tin?
 - Máy gửi đợi ACK trong 1 khoảng thời gian nhất định.
Timeout
- Nên đợi trong bao lâu?
 - Ít nhất 1 RTT (Round Trip Time)
 - Cần bắt đầu bộ đếm thời gian khi gửi 1 gói tin
- Nếu gói tin đã đến và ACK bị mất?
 - Đánh số gói tin

Mô tả các kịch bản truyền thông

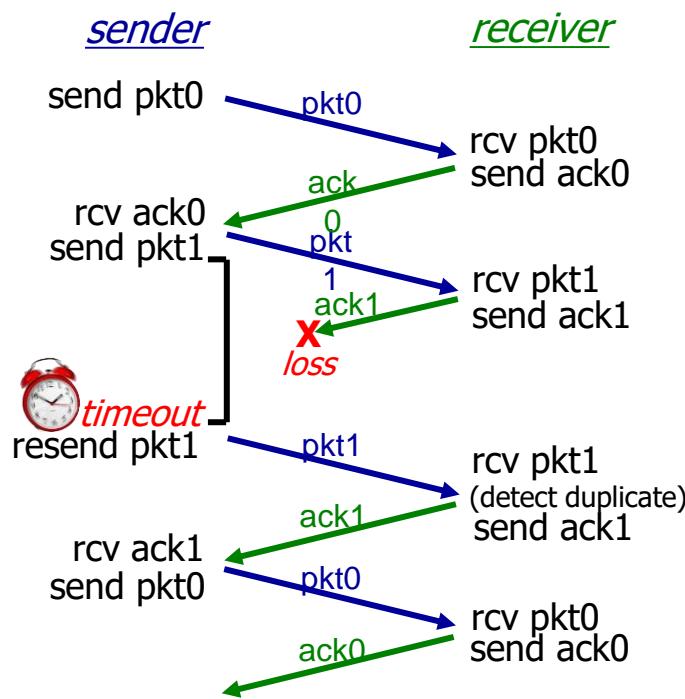


(a) Không mất gói

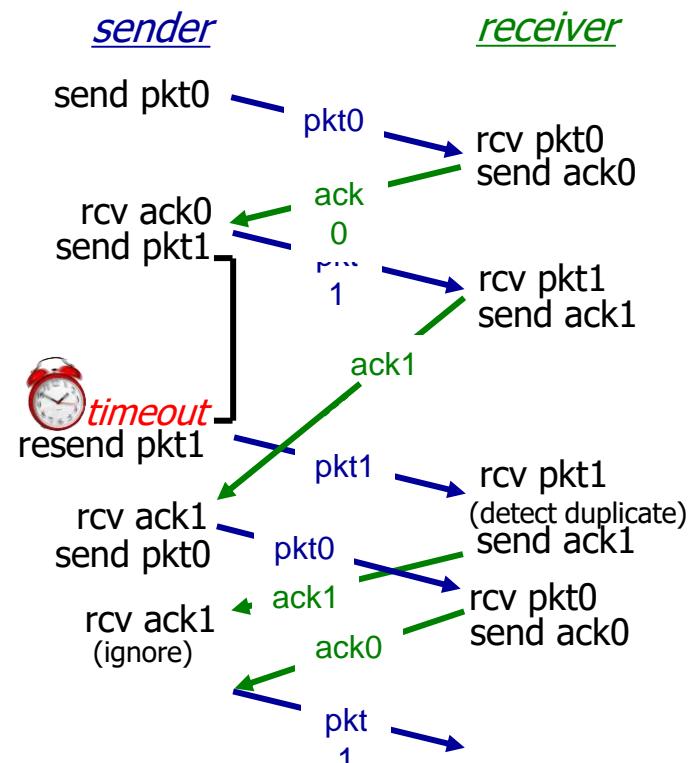


(b) mất gói

Mô tả các kịch bản truyền thông



(c) Mất ACK



(d) ACK bị trễ

Hiệu năng của truyền thông tin cậy (stop-and-wait)

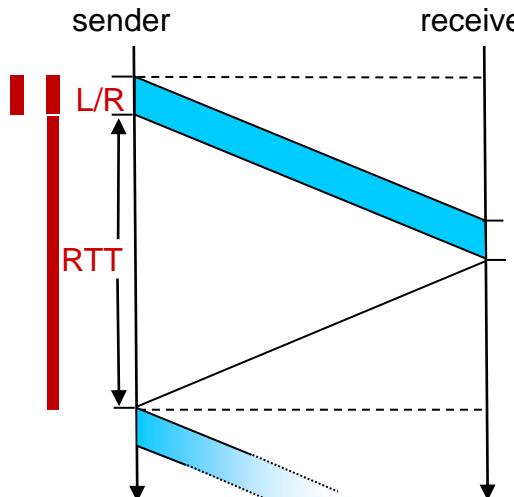
- U_{sender} : *utilization* – fraction of time sender busy sending
- example: 1 Gbps link, 15 ms prop. delay, 8000 bit packet
 - time to transmit packet into channel:

$$D_{trans} = \frac{L}{R} = \frac{8000 \text{ bits}}{10^9 \text{ bits/sec}} = 8 \text{ microsecs}$$

Hiệu năng của truyền thông tin cây (stop-and-wait)

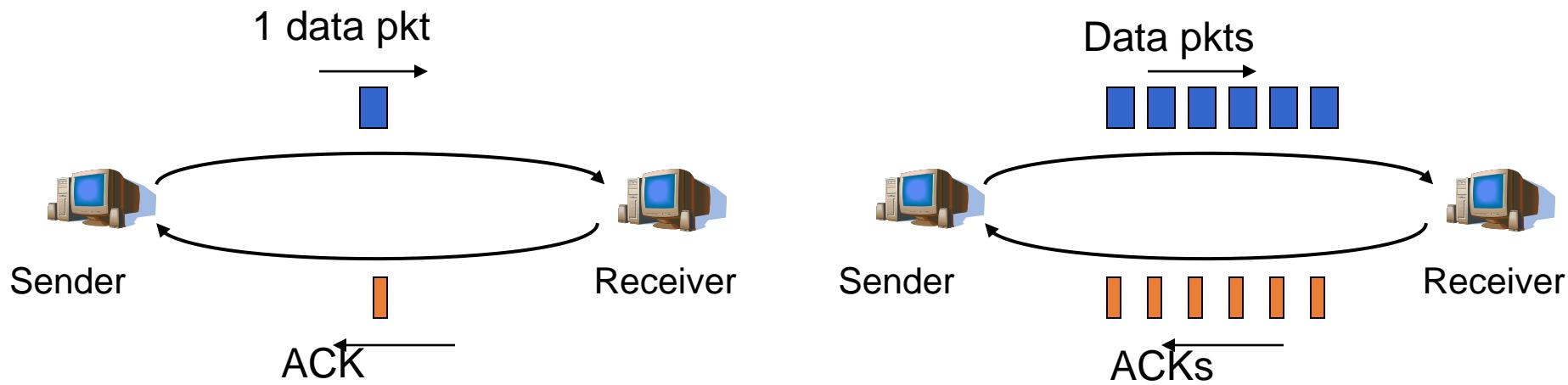
- Ví dụ: Liên kết 1 Gbps, 15 ms trễ lan truyền, gói tin 8000 bit
 - Để đẩy gói tin lên đường truyền: $D_{trans} = \frac{L}{R} = \frac{8000 \text{ bits}}{10^9 \text{ bits/sec}} = 8 \text{ microsecs}$

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



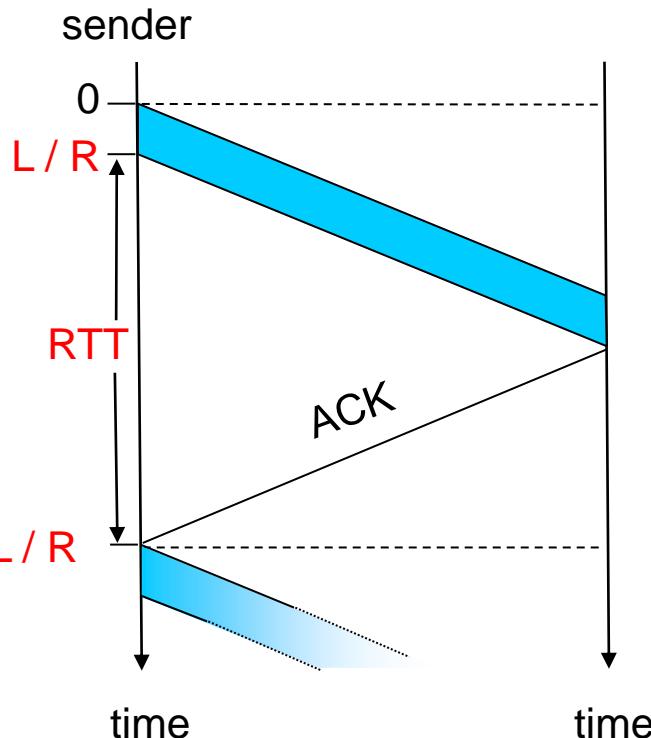
- Hiệu năng rất kém

Truyền thông kiểu đường ống



So sánh hiệu quả

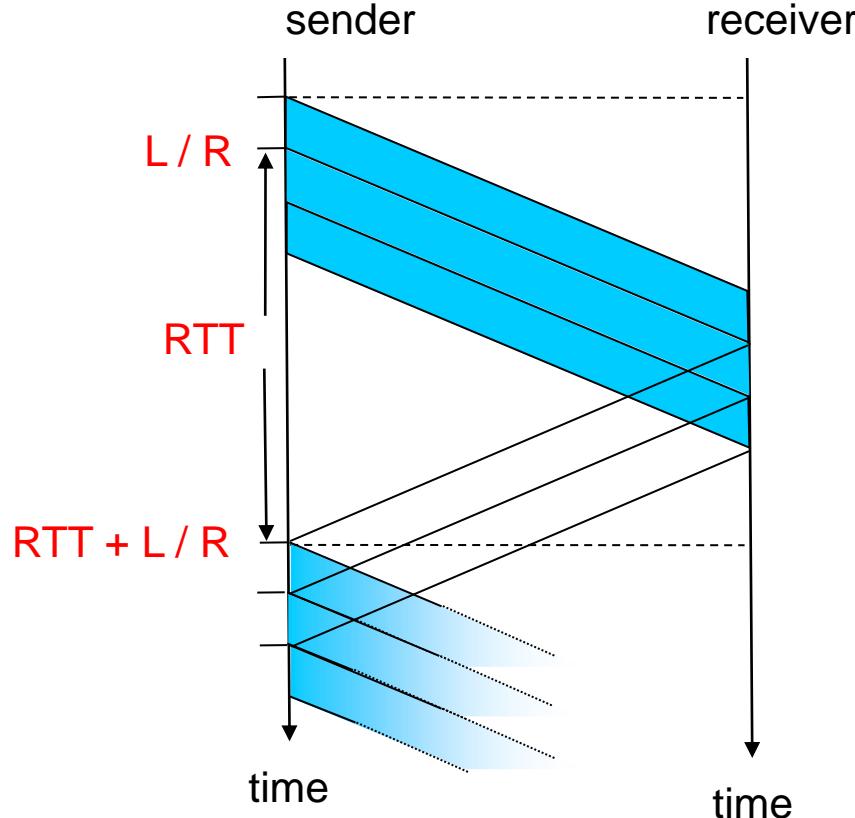
stop-and-wait



L: Size of data pkt
 R: Link bandwidth
 RTT: Round trip time

$$\text{Performance} = \frac{L/R}{RTT + L/R}$$

Pipeline



$$\text{Performance} = \frac{3 * L/R}{RTT + L/R}$$

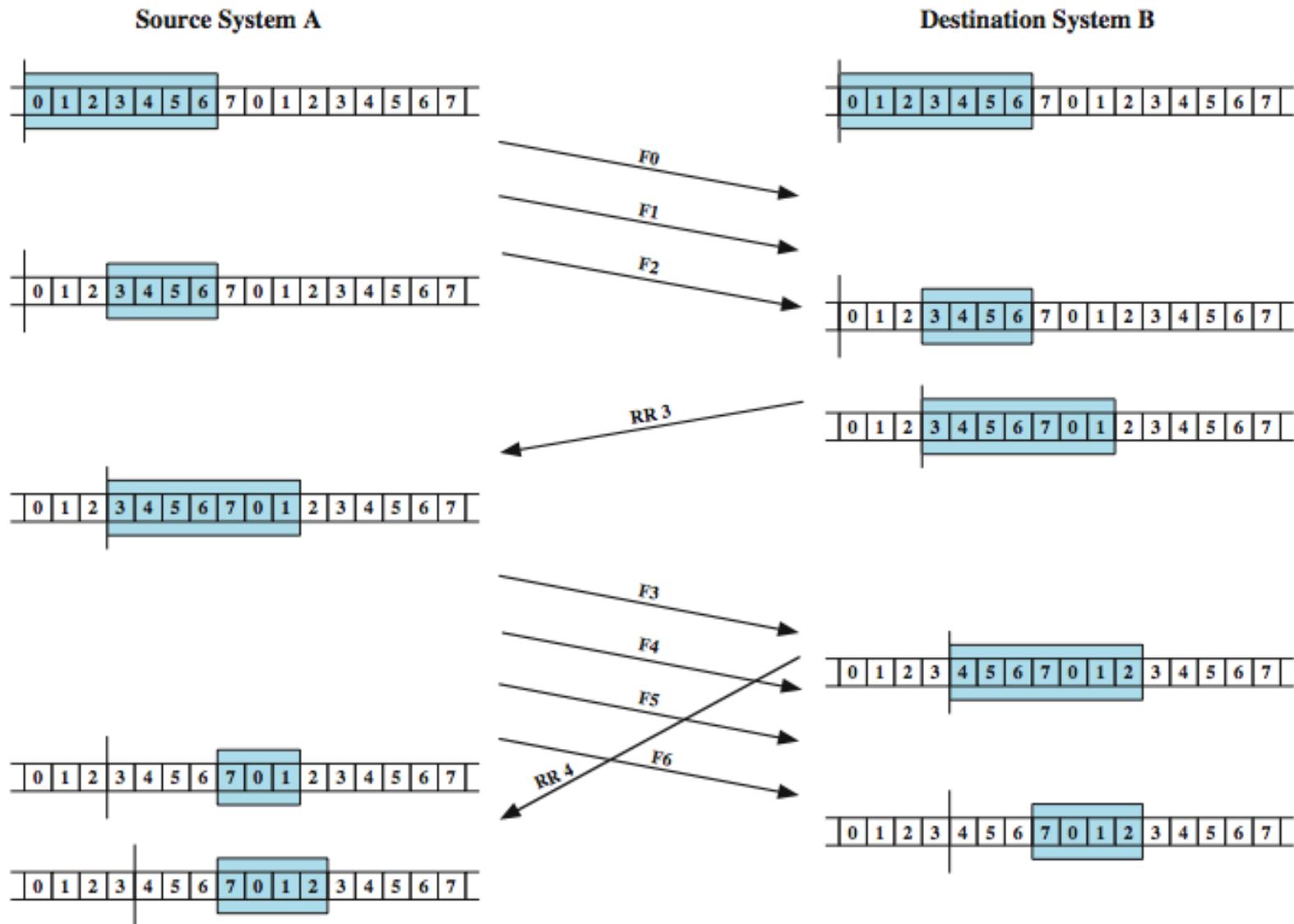
Cơ chế của sổ trượt: nguyên tắc

- Gửi nhiều khung dữ liệu để giảm thời gian chờ.
- Các khung đã gửi đi chưa báo nhận được lưu trữ tạm thời trong bộ nhớ đệm.
- Số khung được truyền đi phụ thuộc vào bộ nhớ đệm.
- Khi nhận được báo nhận
 - giải phóng khung dữ liệu đã truyền thành công khỏi bộ nhớ đệm
 - Truyền tiếp các khung bằng với số khung đã truyền thành công

Cơ chế của sổ trượt: Nguyên tắc

- Xét hai trạm A, B kết nối bằng một đường truyền song công
 - B có bộ nhớ đệm n khung dữ liệu
 - Như vậy B có thể nhận cùng một lúc n khung dữ liệu mà không cần báo nhận
- Báo nhận
 - Để ‘nhớ’ các khung dữ liệu đã báo nhận, cần đánh số các khung dữ liệu
 - B báo nhận một khung bằng cách báo số khung dữ liệu mà B đang chờ nhận, ngầm định đã nhận tất cả các khung trước đó
 - Một báo nhận có thể dùng cho nhiều khung dữ liệu

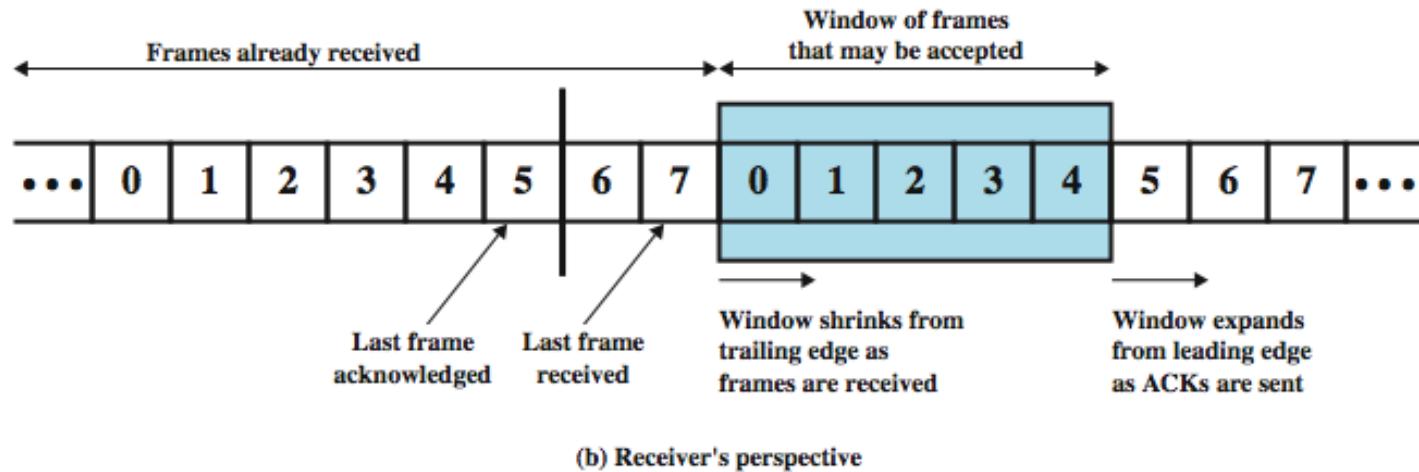
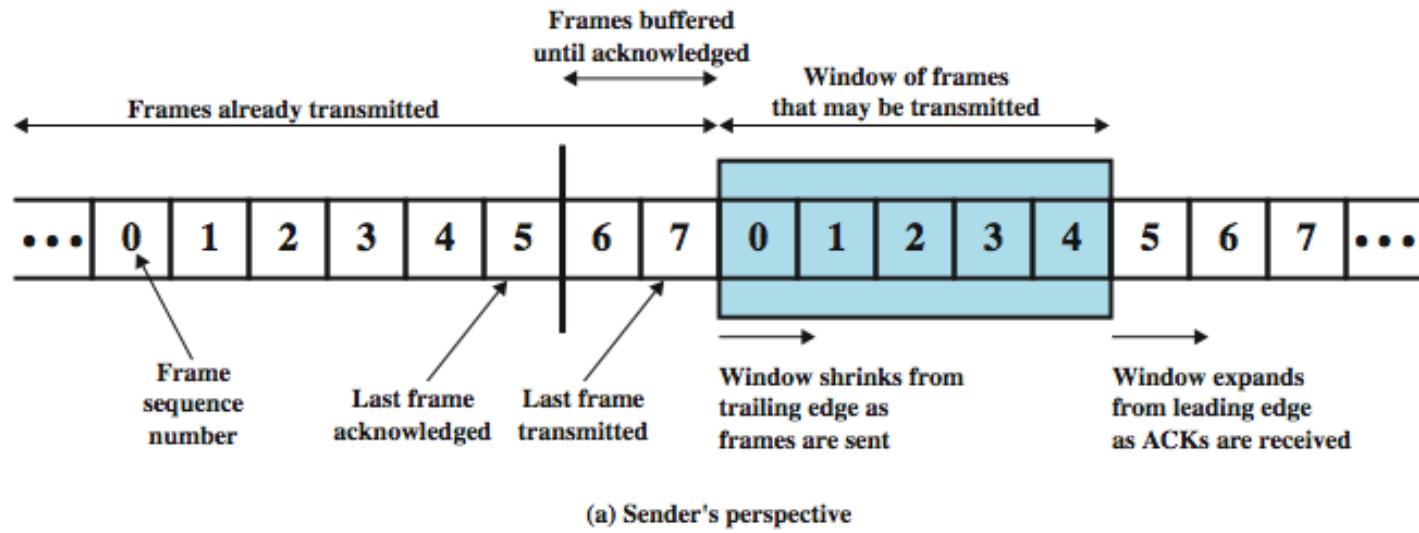
Cơ chế cửa sổ trượt



Trong cửa sổ là các khung sẽ phát

Trong cửa sổ là các khung chờ nhận

Cơ chế cửa sổ trượt



Cơ chế cửa sổ trượt

- Các khung dữ liệu đang được gửi đi được đánh số. Số thứ tự phải lớn hơn hoặc bằng kích thước cửa sổ
- Các khung dữ liệu được báo nhận bằng thông báo có đánh số
- Được báo gộp. Nếu 1,2,3,4 được nhận thành công, chỉ gửi báo nhận 4
- Khi đã nhận được thông báo nhận được khung k, có nghĩa là tất cả các khung k-1, k-2.. đã nhận được

Cơ chế cửa sổ trượt

- Nguồn quản lý
 - Các khung đã gửi đi thành công
 - Các khung đã gửi đi chưa báo nhận
 - Các khung có thể gửi đi ngay
 - Các khung chưa thể gửi đi ngay
- Đích quản lý
 - Các khung đã nhận được
 - Các khung đang chờ nhận

Các kỹ thuật báo nhận, phát lại

- **Kỹ thuật tự động truyền lại (ARQ automatic repeat request).**
- Có 3 phiên bản chuẩn hóa
 - Dừng và chờ (Stop and Wait) ARQ
 - Đã trình bày
 - Quay lại N (Go Back N) ARQ
 - Loại bỏ chọn lọc (Selective Reject) ARQ

Kỹ thuật tự động truyền lại ARQ

- Quay lại N (Go Back N) ARQ
 - Phiên bản đơn giản của Sliding windows.
 - Phía gửi phát liên tục các gói tin (max N gói)
 - Phía nhận phát liên tục các ACK nếu gói được nhận tốt
 - Phía gửi đặt timeout phải nhận ACK cho từng gói tin
 - Khi gặp timeout phía gửi **tự động phát lại tất cả các gói tin bắt đầu từ gói đầu tiên không nhận được ACK**
- Loại bỏ chọn lọc (Selective Reject) ARQ
 - Phía gửi phát liên tục các gói tin (max N gói)
 - Phía nhận phát liên tục các ACK nếu gói được nhận tốt
 - Phía gửi đặt timeout phải nhận ACK cho từng gói tin
 - Phía gửi **tự động phát lại chỉ gói tin không nhận được ACK sau timeout**

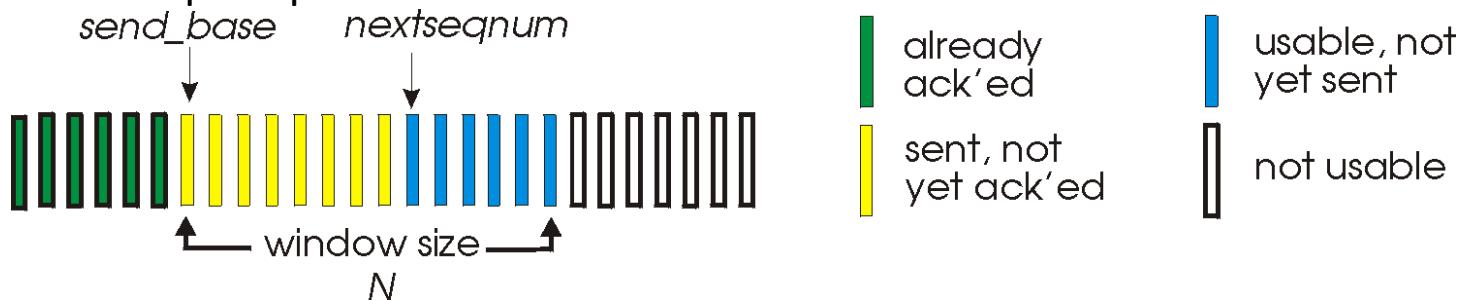
Kỹ thuật tự động truyền lại ARQ

- Các kỹ thuật ARQ được sử dụng trong cả kiểm soát luồng và kiểm soát lỗi
 - Khi gói tin đến đích bị lỗi cần phát lại
 - Khi gói tin đến đích không đúng thứ tự cần phát lại
 - Khi gói tin bị mất hoặc ACK bị mất.

Go-Back-N: máy gửi

- Máy gửi: “cửa sổ” tối đa N, các gói tin đã được gửi nhưng chưa nhận được ACK

- k-bit seq # in pkt header

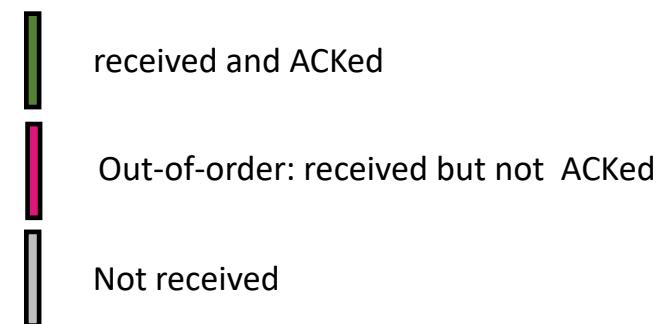
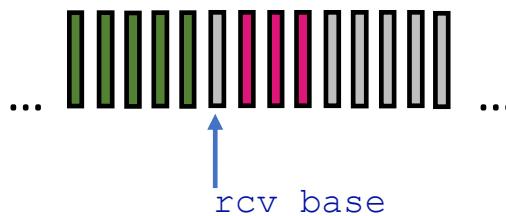


- ACK tích lũy:** ACK(n): ACKs tới gói tin có số thứ tự seq # n
- Khi nhận được ACK(n): dịch cửa sổ lên phía trước 1 vị trí tại $n+1$
 - Bộ đếm thời gian cho gói tin được gửi sớm nhất
- timeout(n):** gửi lại gói tin n và những gói khác với seq# lớn hơn trong cửa sổ

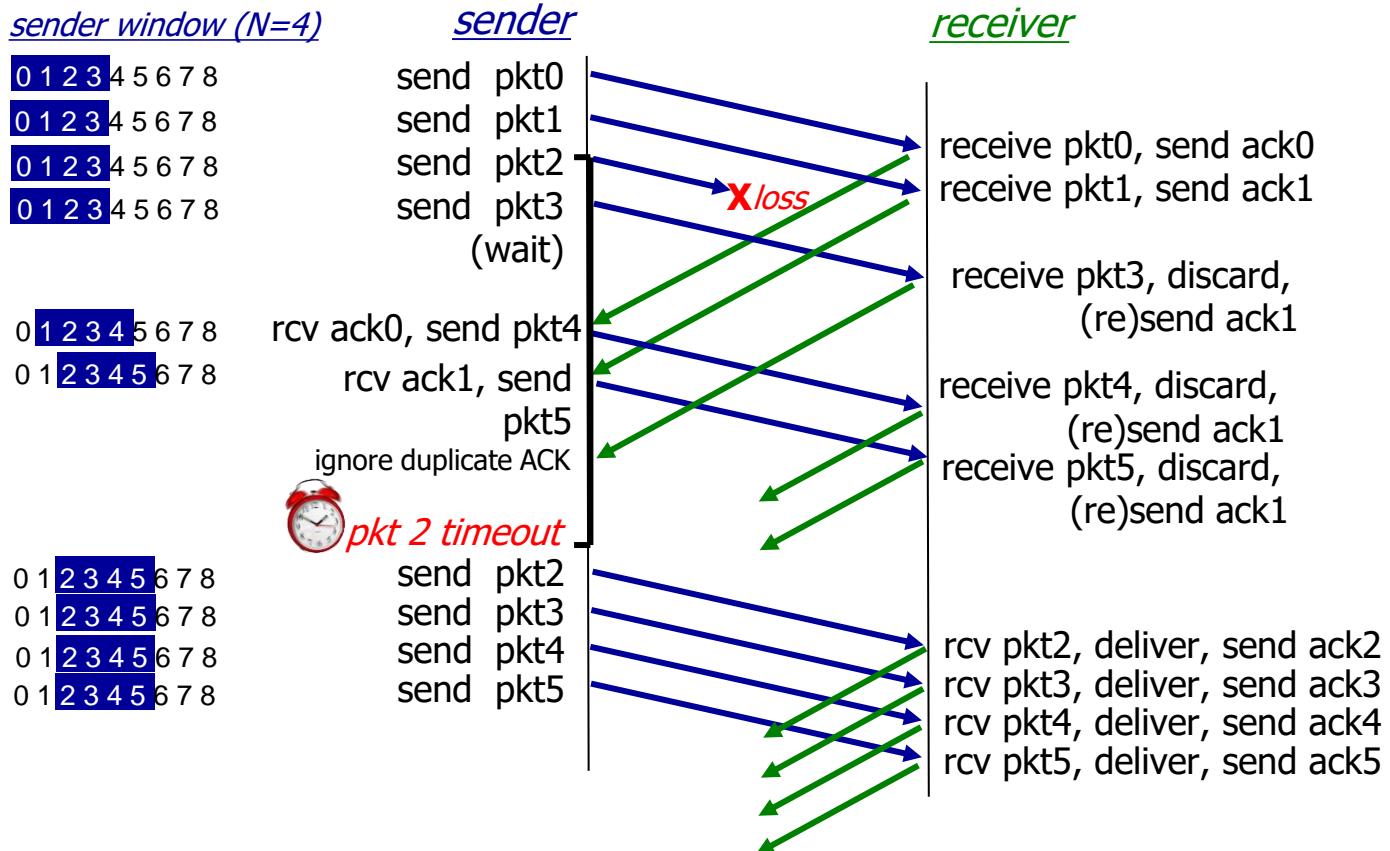
Go-Back-N: máy nhận

- Chỉ-ACK: luôn gửi ACK cho những gói đến đúng thứ tự, với seq# lớn nhất **theo thứ tự**
 - Có thể tạo ACK lặp
 - Chỉ cần nhớ `rcv_base`
- Khi nhận được gói tin không đúng thứ tự:
 - Loại bỏ (không dùng buffer) hoặc buffer: tùy khi thực thi
 - Gửi lại ACK với thứ tự seq# lớn nhất đúng thứ tự

Receiver view of sequence number space:



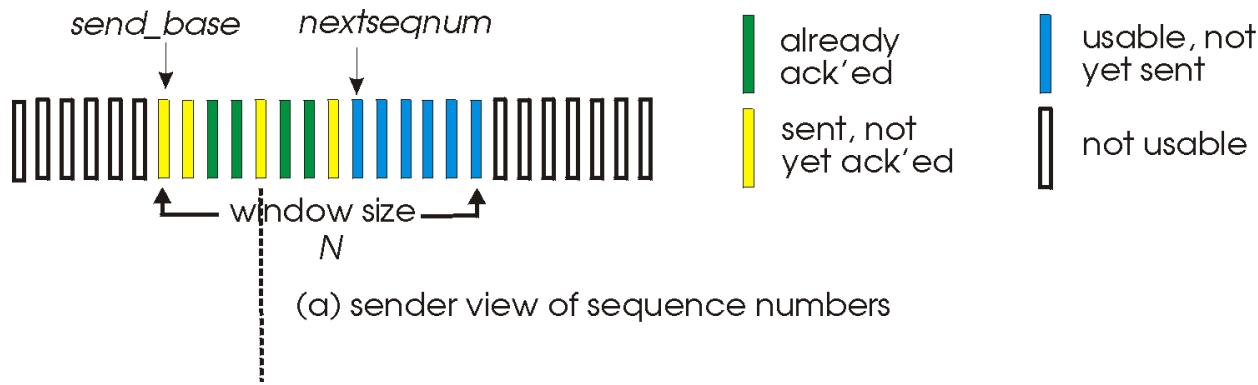
Go-Back-N khi thực hiện



Selective repeat

- Máy nhận đánh dấu từng gói tin nhận được thành công
- Máy nhận đánh dấu và gửi (ACK) từng gói tin nhận được thành công
 - Lưu gói tin bằng buffer để có thể chuyển lên tầng trên theo đúng thứ tự
- Máy gửi gửi lại những gói tin chưa có ACK đơn lẻ khi timeout
 - Máy gửi cần duy trì bộ đếm cho từng gói tin đã gửi (nhưng chưa nhận được ACK)
- Cửa sổ máy gửi
 - *N* gói tin liên tiếp
 - Giới hạn số gói tin được gửi mà chưa nhận được ACK

Selective repeat: cửa sổ máy gửi và nhận



Selective repeat: máy gửi và nhận

sender

data từ tầng trên:

- Nếu seq# tiếp theo trong cửa sổ sẵn sàng, gửi gói tin

timeout(n):

- Gửi lại gói tin n , reset bộ đếm

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

- Đánh dấu gói tin n là “đã nhận”
- Nếu n là gói tin ACK “nhỏ nhất”, dịch cửa sổ 1 vị trí tới seq# tiếp theo (chưa nhận được ACK)

receiver

Gói tin n trong [rcvbase, rcvbase+N-1]

- Gửi ACK(n)
- Không đúng thứ tự: buffer
- Đúng thứ tự: chuyển toàn bộ gói tin (bao gồm cả buffer) lên tầng trên, dịch cửa sổ 1 vị trí

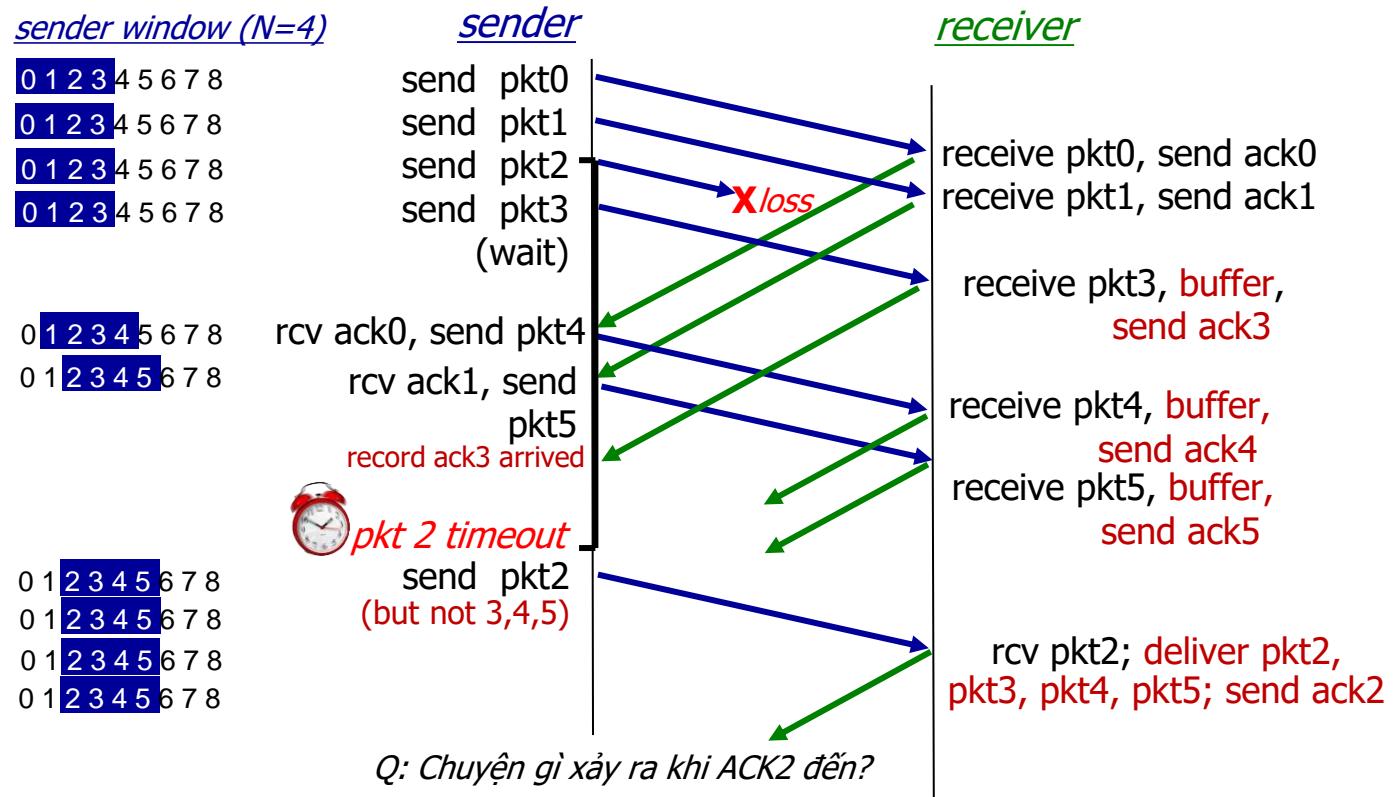
Gói tin n trong [rcvbase-N, rcvbase-1]

- ACK(n)

Trái lại:

- Bỏ qua

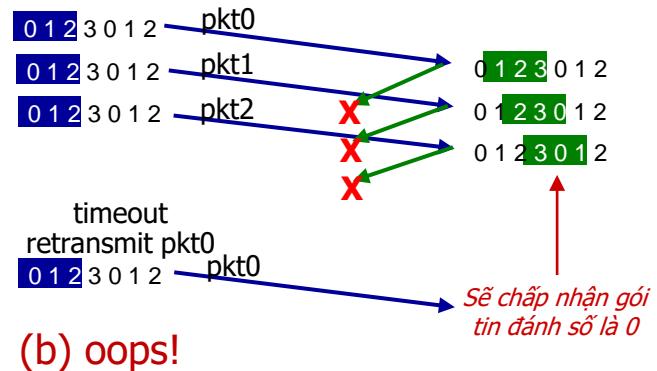
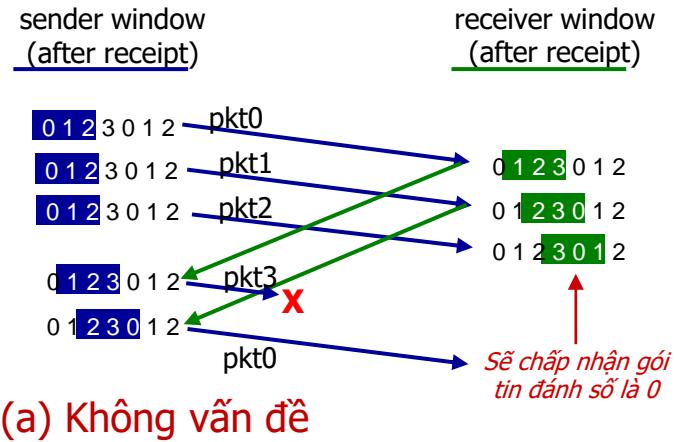
Selective Repeat khi thực hiện



Selective repeat: vẫn đê!

Ví dụ:

- seq #s: 0, 1, 2, 3 (bộ đếm cơ số 4)
- Kích thước cửa sổ=3

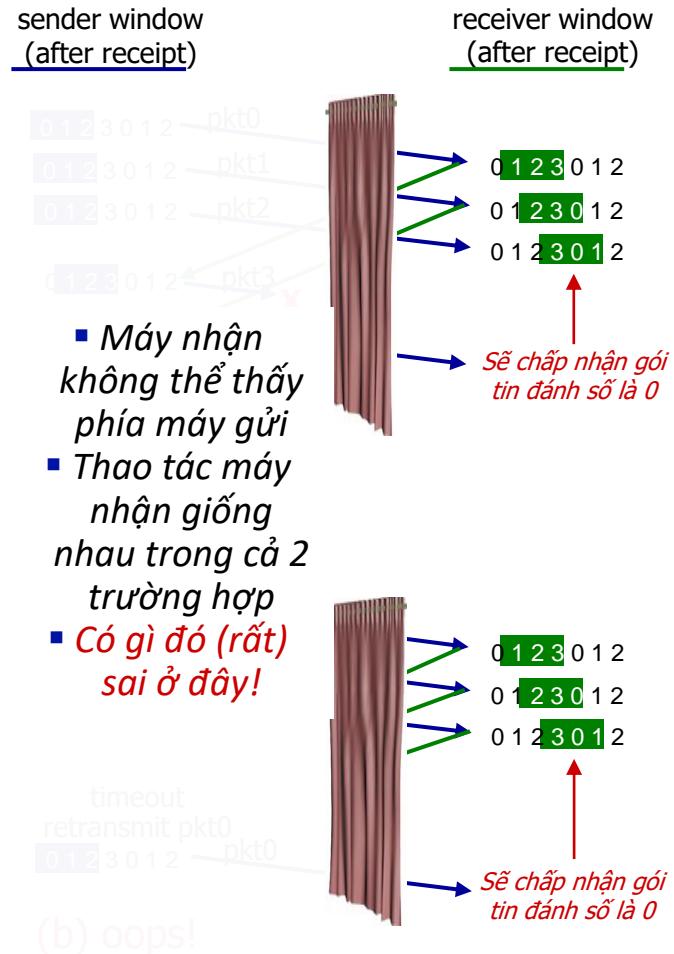


Selective repeat: vẫn đê!

Ví dụ:

- seq #: 0, 1, 2, 3 (bộ đếm cơ số 4)
- Kích thước cửa sổ=3

Q: Mỗi quan hệ giữa kích thước seq# và kích thước cửa sổ là bao nhiêu để tránh kịch bản này?





ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

UDP User Datagram Protocol

Tổng quan

Khuôn dạng gói tin

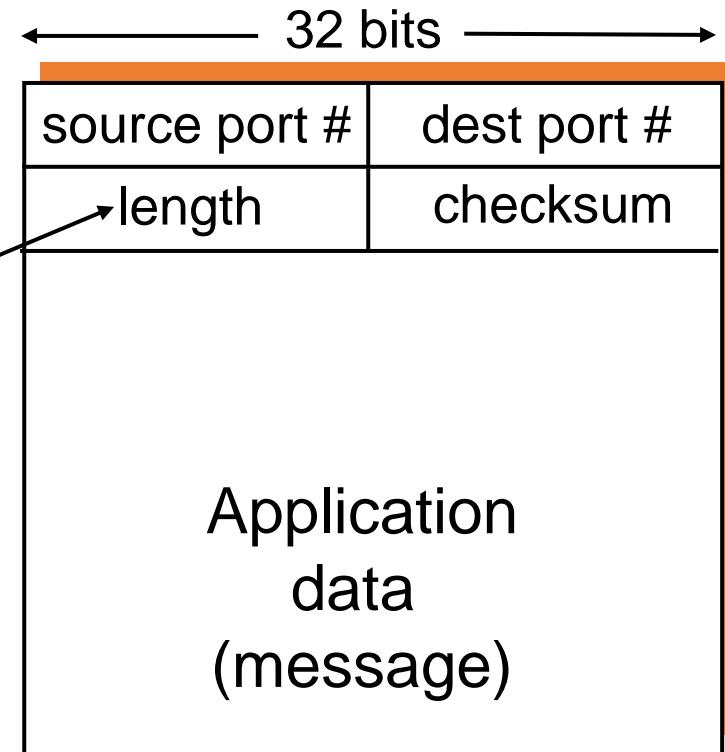
Giao thức dạng “Best effort”

- Dùng UDP khi nào?
 - Không cần thiết lập liên kết (tăng độ trễ)
 - Đơn giản: Không cần lưu lại trạng thái liên kết ở bên gửi và bên nhận
 - Phàn đầu đoạn tin nhỏ
 - Không có quản lý tắc nghẽn: UDP cứ gửi dữ liệu nhanh nhất, nhiều nhất nếu có thể
- UDP có những chức năng cơ bản gì?
 - Dồn kênh/phân kênh
 - Phát hiện lỗi bit bằng checksum

Khuôn dạng bức tin (datagram)

- UDP sử dụng đơn vị dữ liệu gọi là –datagram (bức tin)

Độ dài toàn bộ bức tin tính theo byte



Khuôn dạng đơn vị dữ liệu của UDP

Các vấn đề của UDP

- Không có kiểm soát tắc nghẽn
 - Làm Internet bị quá tải
- Không bảo đảm được độ tin cậy
 - Các ứng dụng phải cài đặt cơ chế tự kiểm soát độ tin cậy
 - Việc phát triển ứng dụng sẽ phức tạp hơn



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TCP

Transmission Control Protocol

Cấu trúc đoạn tin TCP

Quản lý liên kết

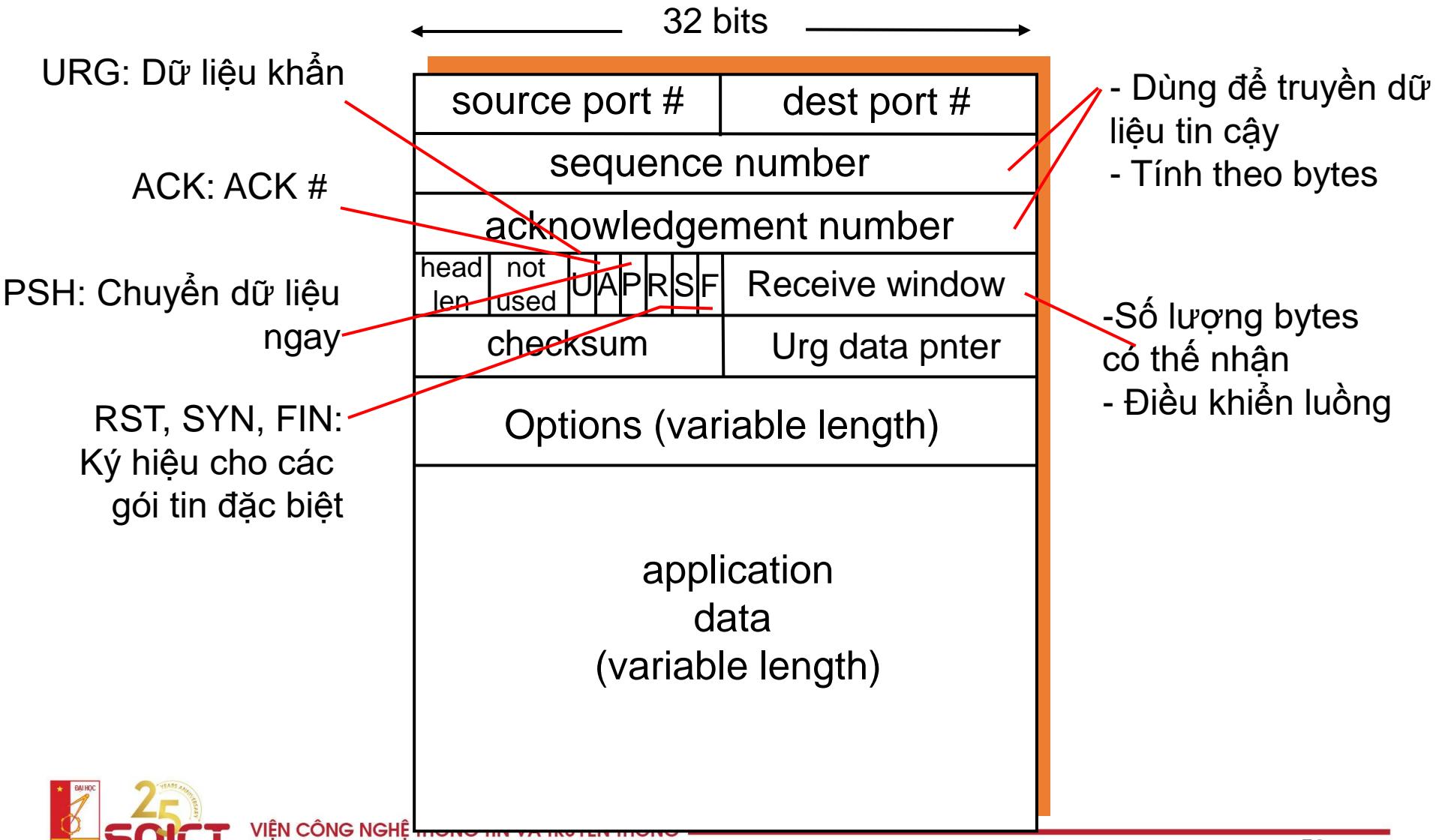
Kiểm soát luồng

Kiểm soát tắc nghẽn

Tổng quan về TCP

- Giao thức hướng liên kết
 - Bắt tay ba bước
- Giao thức truyền dữ liệu theo dòng byte, tin cậy
 - Sử dụng vùng đệm
- Truyền theo kiểu pipeline, sử dụng sliding windows.
 - Tăng hiệu quả
- Kiểm soát luồng
 - Bên gửi không làm quá tải bên nhận (thực tế: quá tải)
- Kiểm soát tắc nghẽn
 - Việc truyền dữ liệu không nên làm tắc nghẽn mạng (thực tế: luôn có tắc nghẽn)

Khuôn dạng đoạn tin - TCP segment



TCP cung cấp dịch vụ tin cậy ntn?

- Kiểm soát dữ liệu đã được nhận chưa:
 - Seq. #
 - Ack
- Chu trình làm việc của TCP:
 - Thiết lập liên kết
 - Bắt tay ba bước
 - Truyền/nhận dữ liệu
 - Đóng liên kết

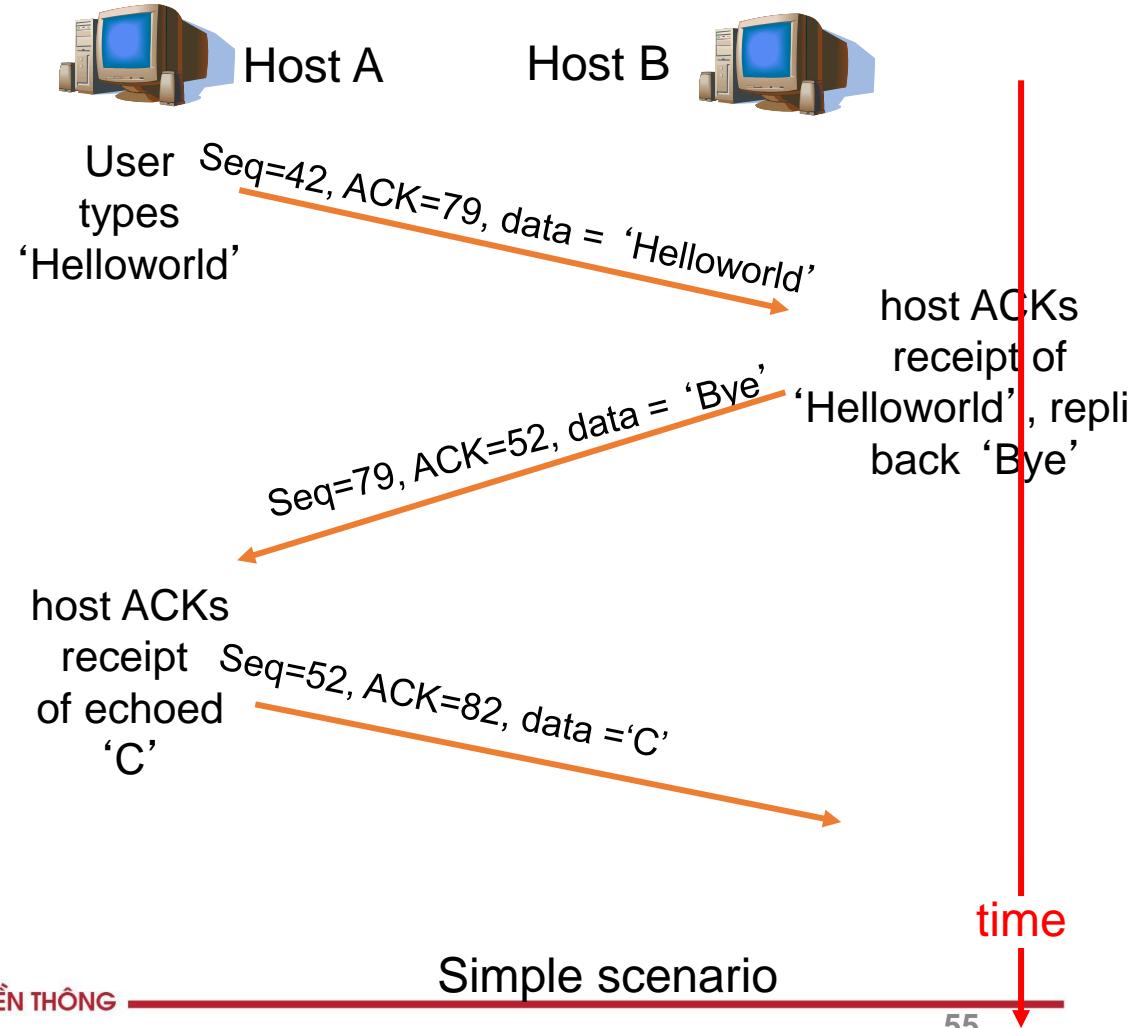
Cơ chế báo nhận trong TCP

Seq. #:

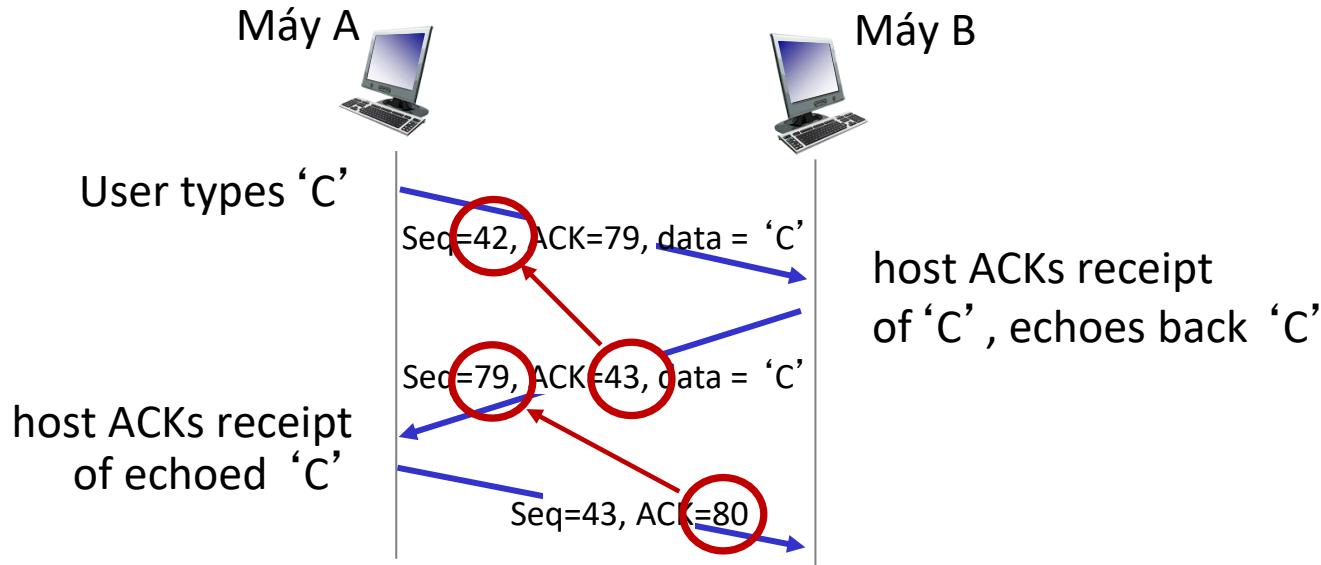
- Số hiệu của byte đầu tiên của đoạn tin trong dòng dữ liệu

ACK:

- Số hiệu byte mong muốn nhận từ đối tác
- Ngầm xác nhận đã nhận tốt các byte trước đó.

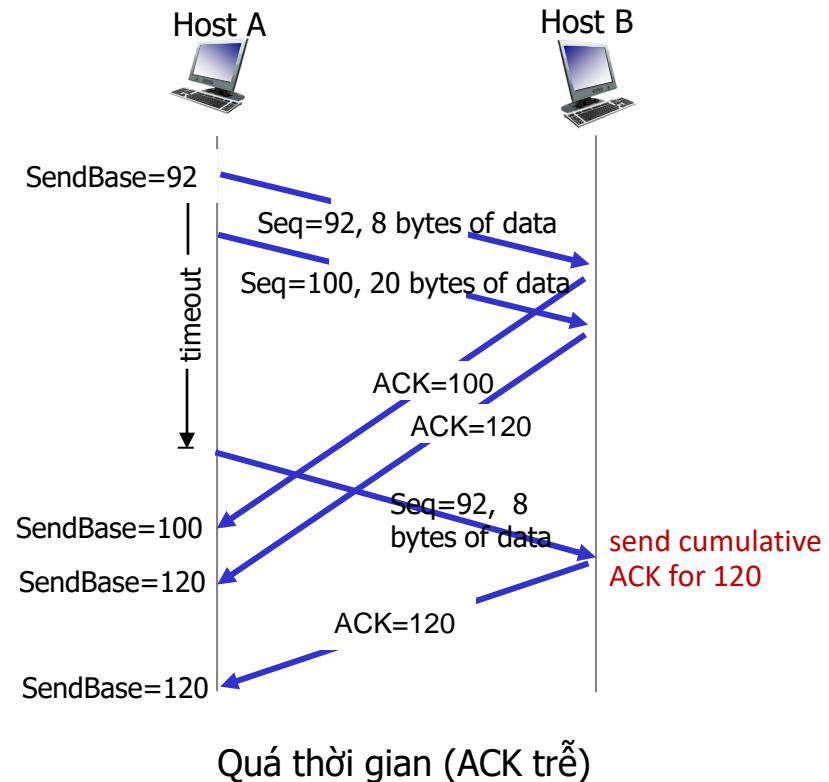
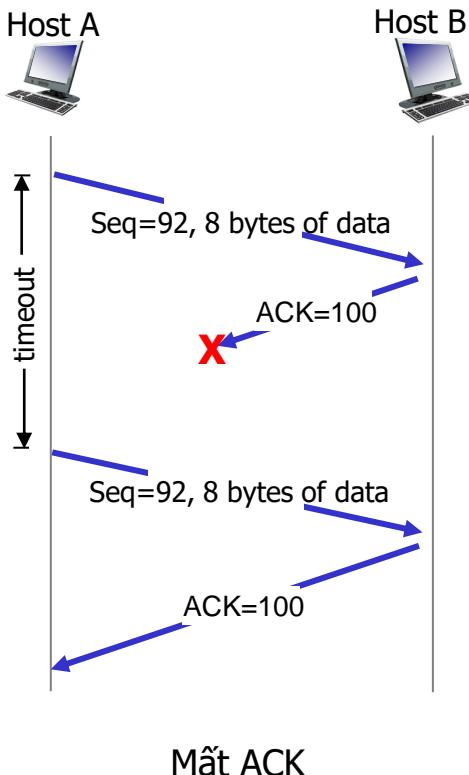


Cơ chế báo nhận trong TCP (ACK)

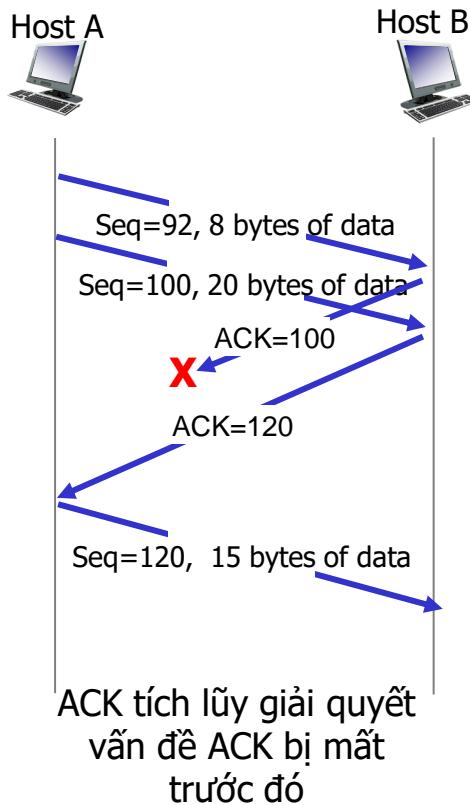


Kịch bản telnet đơn giản

TCP: kịch bản gửi lại



TCP: kịch bản gửi lại



Gửi lại nhanh trong TCP

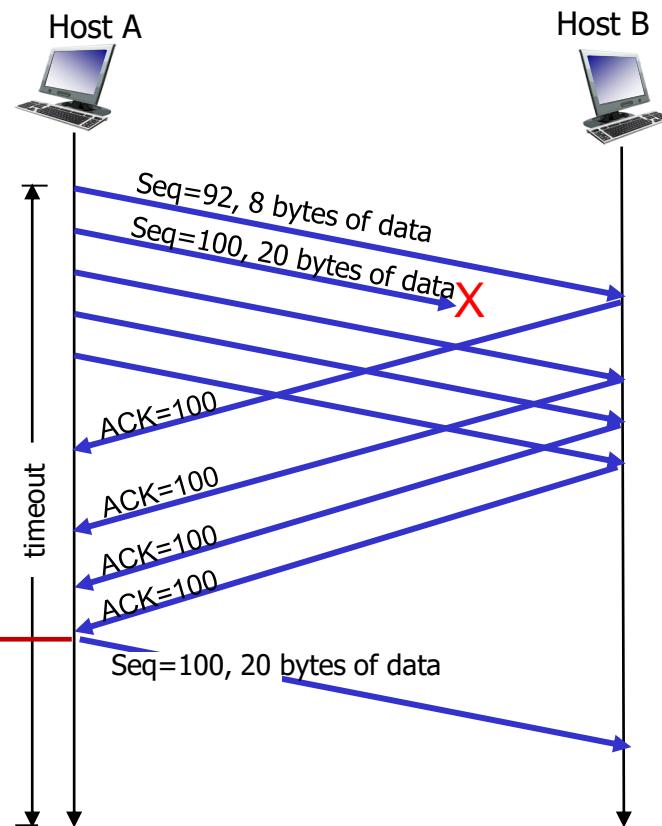
Gửi lại nhanh trong TCP

Nếu máy gửi nhận được 3 gói tin ACK cùng dữ liệu, thì gửi lại gói tin (chưa ACK) với seq# nhỏ nhất

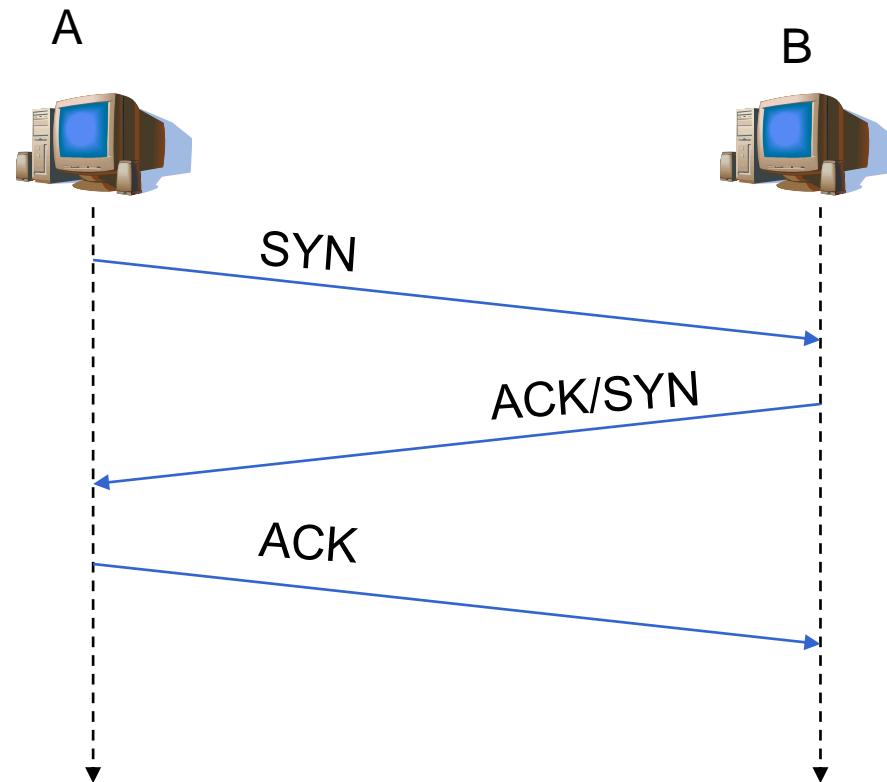
- Nhiều khả năng gói tin đó đã mất, không cần đợi timeout



Nhận 3 ACK giống nhau,
đoán gói tin đã mất, gửi lại
ngay lập tức



Thiết lập liên kết TCP : Giao thức bắt tay 3 bước

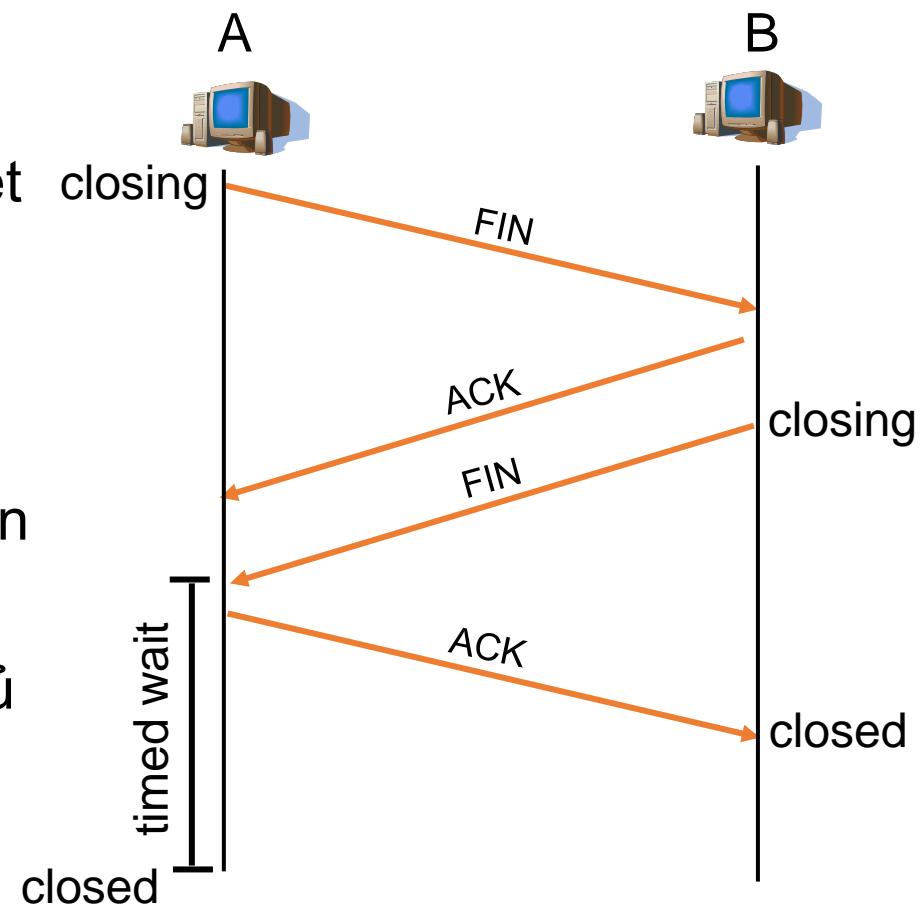


- **Bước 1:** A gửi SYN cho B
 - chỉ ra giá trị khởi tạo seq # của A
 - không có dữ liệu
- **Bước 2:** B nhận SYN, trả lời bằng SYNACK
 - B khởi tạo vùng đệm
 - chỉ ra giá trị khởi tạo seq. # của B
- **Bước 3:** A nhận SYNACK, trả lời ACK, có thể kèm theo dữ liệu

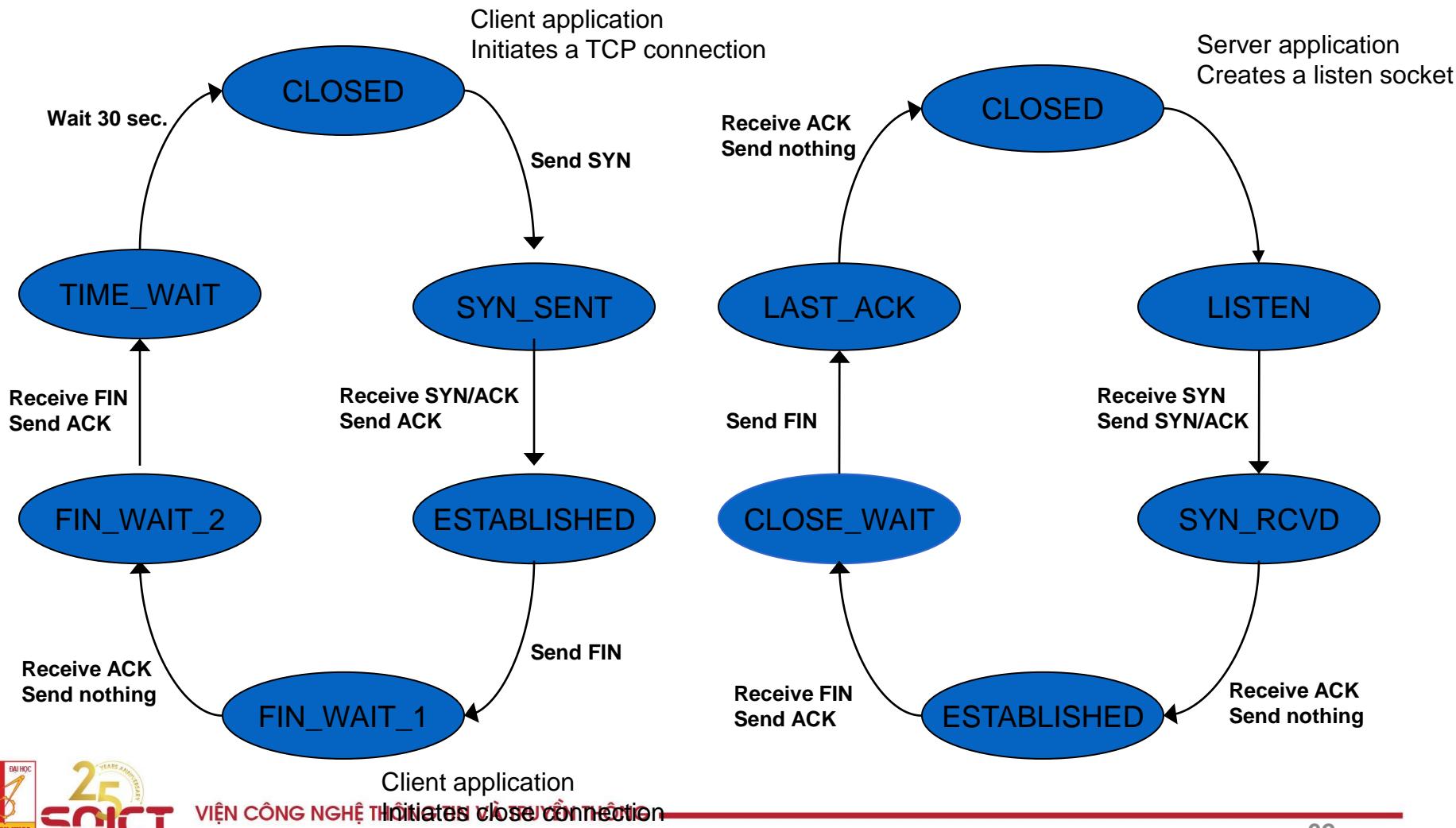
Ví dụ về việc đóng liên kết

- Bước 1: Gửi FIN cho B
- Bước 2: B nhận được FIN, trả lời ACK, đồng thời đóng liên kết closing và gửi FIN.
- Bước 3: A nhận FIN, trả lời ACK, vào trạng thái “chờ”.
- Bước 4: B nhận ACK. đóng liên kết.

Lưu ý: Cả hai bên đều có thể chủ động đóng liên kết



Chu trình sống của TCP (đơn giản hóa)





ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

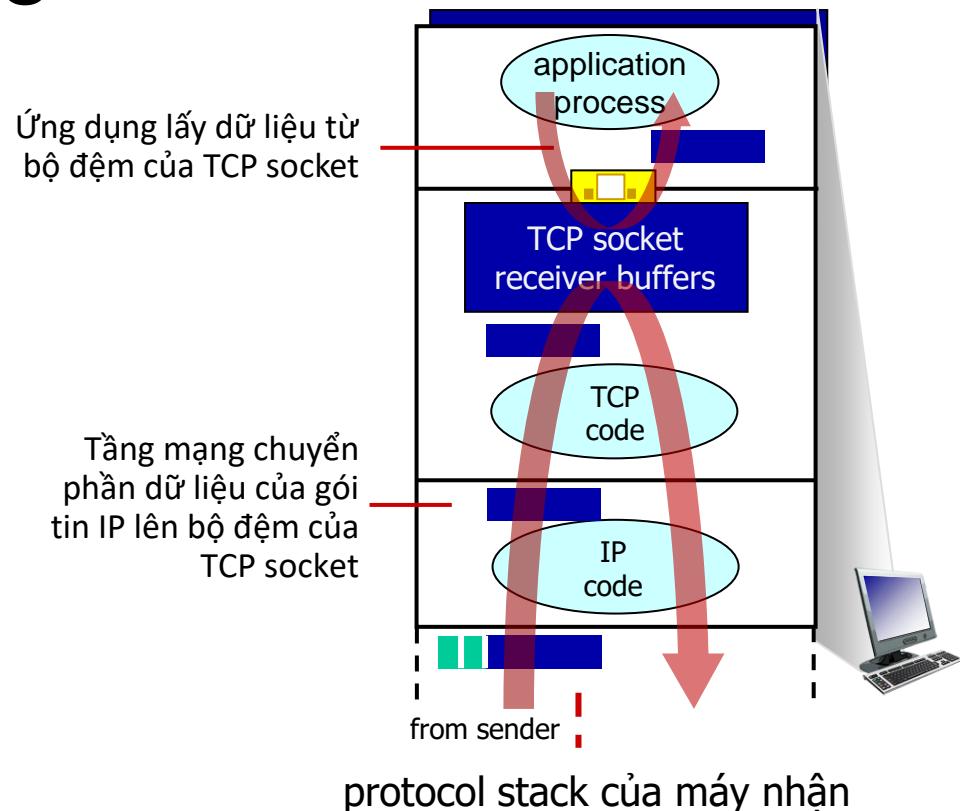
Kiểm soát luồng trong TCP

Kiểm soát luồng trong TCP

- Điều khiển lượng dữ liệu được gửi đi
 - Bảo đảm rằng hiệu quả là tốt
 - Không làm quá tải các bên
- Các bên sẽ có cửa sổ kiểm soát
 - Rwnd: Cửa sổ nhận
 - CWnd: Cửa sổ kiểm soát tắc nghẽn
- Lượng dữ liệu gửi đi phải nhỏ hơn min(Rwnd, Cwnd)

Điều khiển luồng của TCP

Q: Chuyện gì xảy ra nếu tầng mạng chuyển dữ liệu lớn hơn khả năng lấy dữ liệu của tầng ứng dụng?



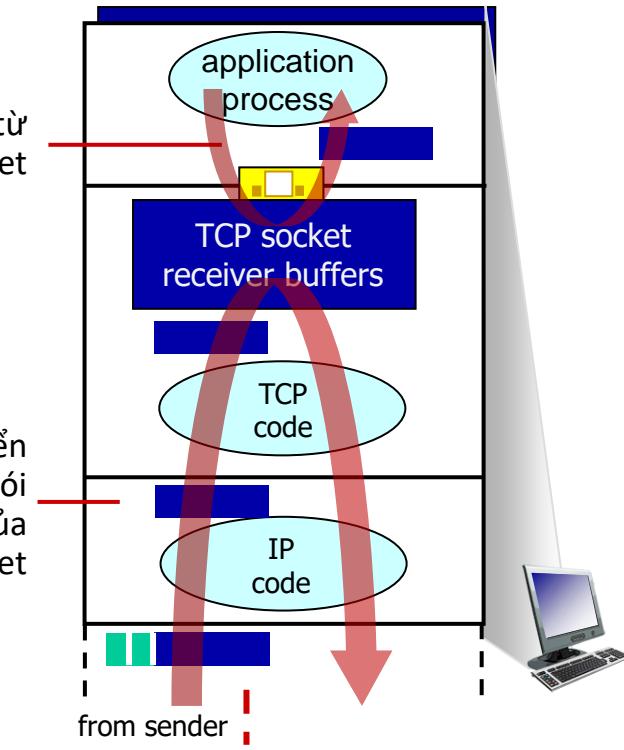
Điều khiển luồng của TCP

Q: Chuyện gì xảy ra nếu tầng mạng chuyển dữ liệu lớn hơn khả năng lấy dữ liệu của tầng ứng dụng?



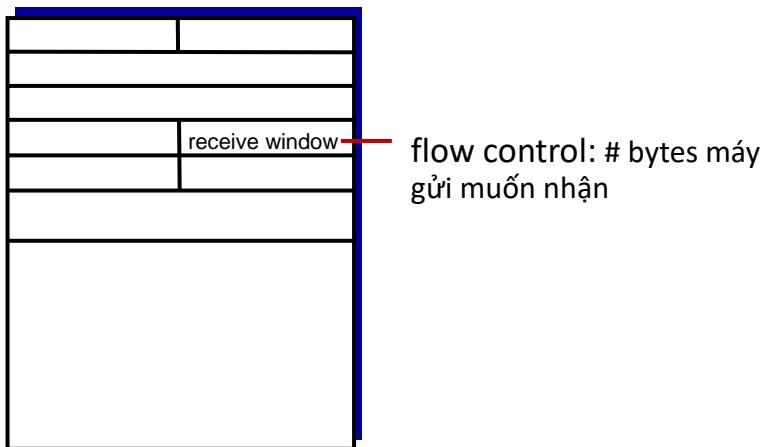
Ứng dụng lấy dữ liệu từ bộ đệm của TCP socket

Tầng mạng chuyển phần dữ liệu của gói tin IP lên bộ đệm của TCP socket

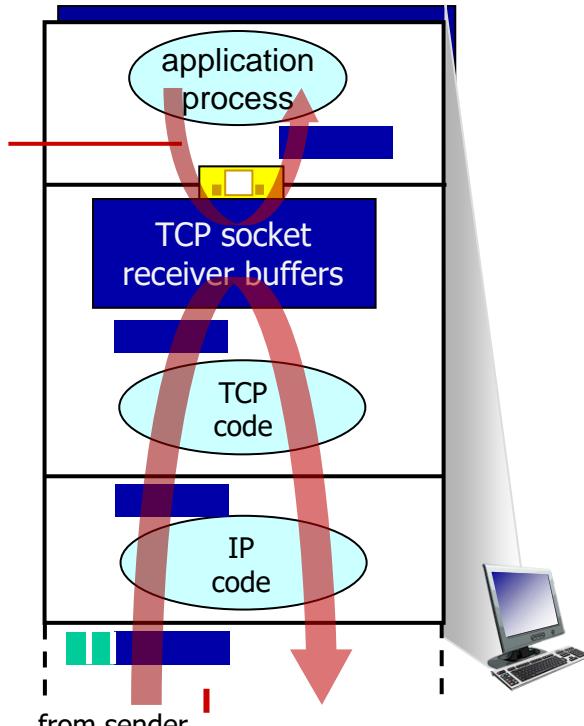


Điều khiển luồng của TCP

Q: Chuyện gì xảy ra nếu tầng mạng chuyển dữ liệu lớn hơn khả năng lấy dữ liệu của tầng ứng dụng?



Ứng dụng lấy dữ liệu từ bộ đệm của TCP socket



protocol stack của máy nhận

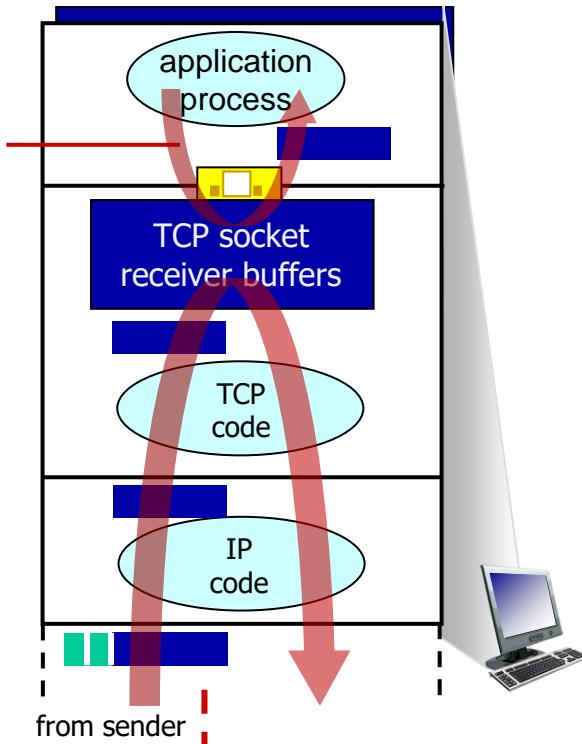
TCP flow control

Q: Chuyện gì xảy ra nếu tầng mạng chuyển dữ liệu lớn hơn khả năng lấy dữ liệu của tầng ứng dụng?

Điều khiển luồng

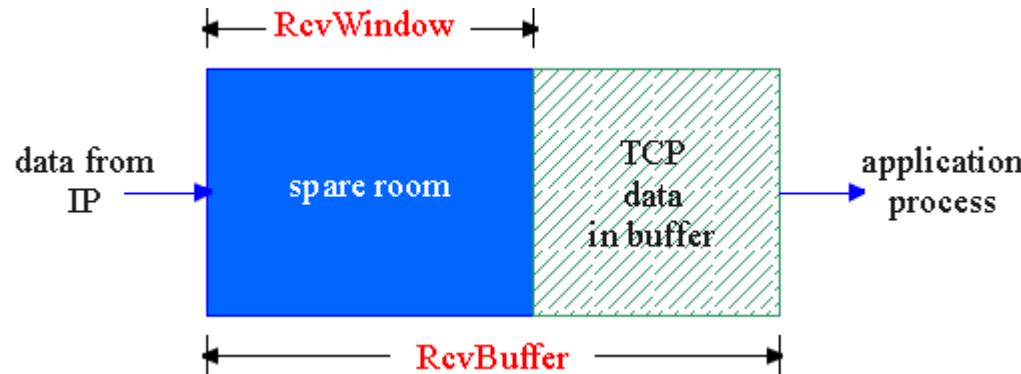
Máy nhận điều khiển máy gửi, để máy gửi không gửi quá khả năng nhận của máy nhận

Ứng dụng lấy dữ liệu từ bộ đệm của TCP socket



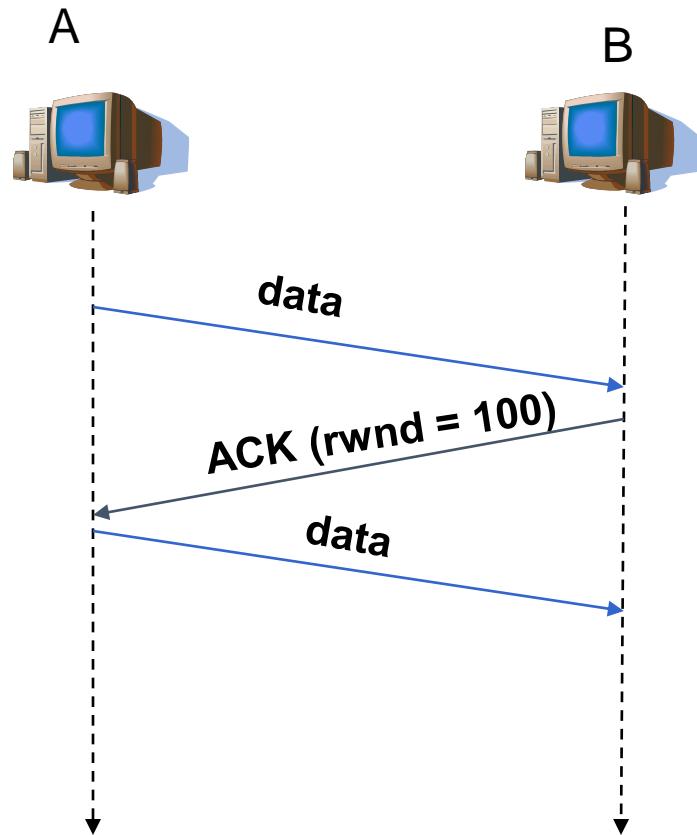
protocol stack của máy nhận

Kiểm soát luồng trong TCP



- Kích thước vùng đệm trống
 - = Rwnd
 - = RcvBuffer - [LastByteRcvd - LastByteRead]

Trao đổi thông tin về Rwnd



- Bên nhận sẽ báo cho bên gửi biết Rwnd trong các đoạn tin
- Bên gửi đặt kích thước cửa sổ gửi theo Rwnd



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Điều khiển tắc nghẽn trong TCP

Nguyên lý kiểm soát tắc nghẽn

Tắc nghẽn:

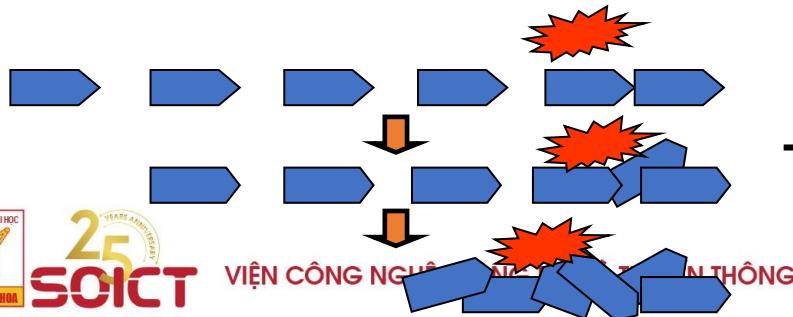
- Không chính thức: “quá nhiều nguồn gửi quá nhiều dữ liệu quá nhanh để **mạng** có thể xử lý”
- Hậu quả:
 - Thông lượng giảm, độ trễ tăng (hàng đợi ở router)
 - Mất gói (quá khả năng bộ đệm của router)
- Vấn đề khác kiểm soát luồng!
- top-10 vấn đề mạng!



Tắc nghẽn: quá
nhiều máy gửi, gửi quá
nhanh



Kiểm soát luồng: một
máy gửi cho 1 máy nhận



**Tắc nghẽn
xảy ra**

Điều khiển tắc nghẽn của TCP: AIMD

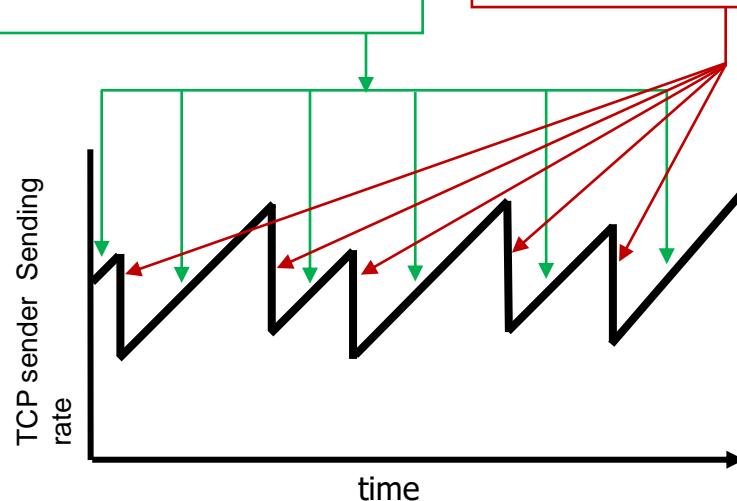
- **Cách tiếp cận:** máy gửi tăng tốc độ gửi cho đến khi mất gói xảy ra (tắc nghẽn), sau đó giảm tốc độ gửi khi xảy ra tắc nghẽn

Additive Increase

Tăng tốc độ gửi 1 gói tin với mỗi RTT cho đến khi mất gói được phát hiện

Multiplicative Decrease

Giảm tốc độ gửi còn một nửa khi mất gói xảy ra



AIMD hình
răng cưa: *thăm
dò băng thông*

TCP AIMD (tiếp)

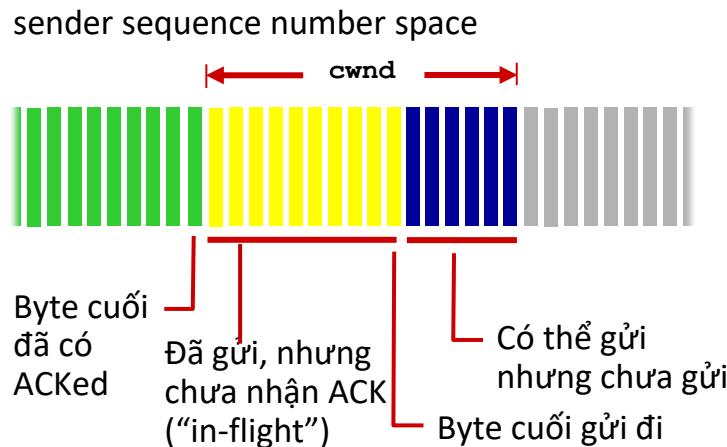
Chi tiết Giảm cấp số nhân (Multiplicative decrease) : tốc độ gửi sẽ

- Giảm một nửa khi nhận được 3 gói ACK trùng lặp (TCP Reno)
- Giảm về 1 MSS (maximum segment size) khi timeout (TCP Tahoe)

Tại sao dùng AIMD?

- AIMD – thuật toán phân tán, bất đồng bộ – giúp :
 - Tối ưu tốc độ luồng trên diện rộng
 - Có những thuộc tính ổn định

Kiểm soát tắc nghẽn TCP: chi tiết



Hành vi gửi TCP:

- Gửi cwnd bytes, chờ RTT các ACK, rồi gửi nhiều byte hơn

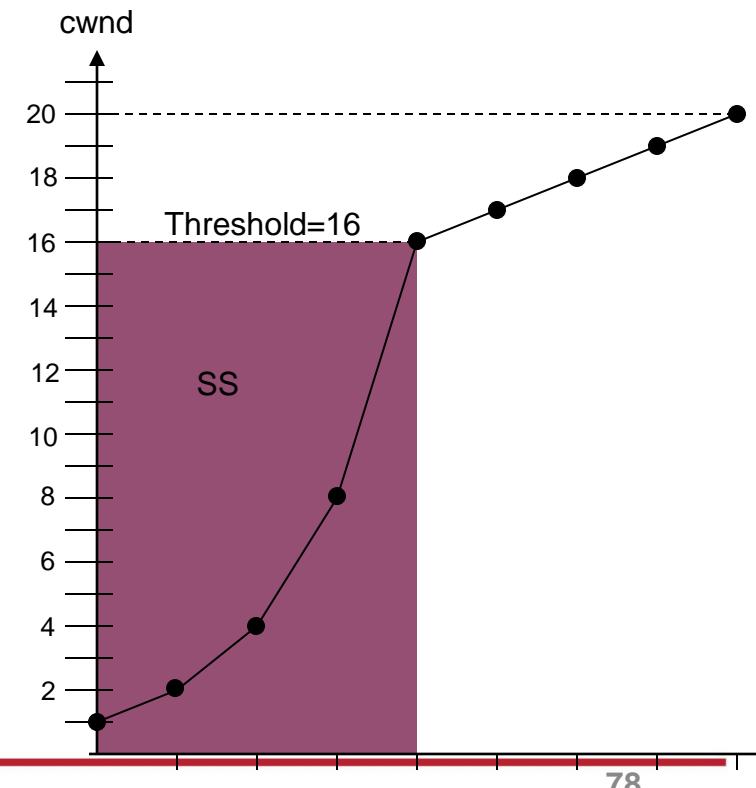
$$\text{Tốc độ TCP} \approx \frac{\text{cwnd}}{\text{RTT}} \text{ bytes/s}$$

- Giới hạn gửi của TCP:
- cwnd thay đổi động tùy thuộc vào mức độ tắc nghẽn của mạng (thực thi điều khiển tắc nghẽn TCP)

$$\text{LastByteSent} - \text{LastByteAcked} \leq \text{cwnd}$$

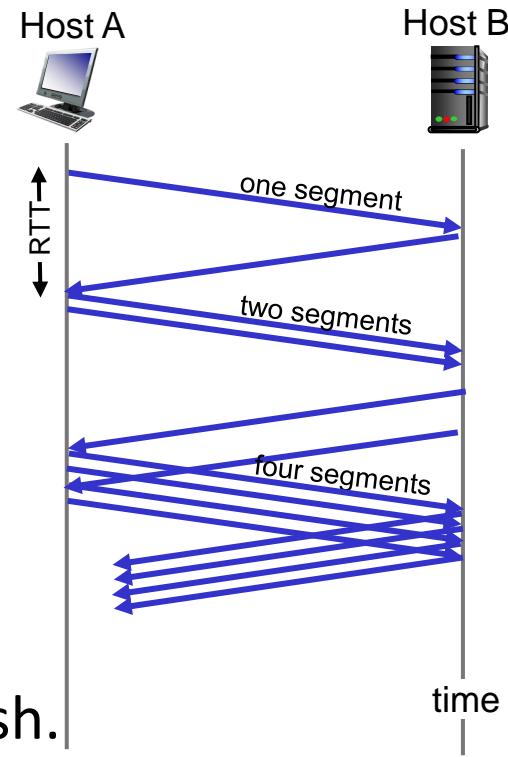
Nguyên lý kiểm soát tắc nghẽn

- Slow-start
 - Tăng tốc độ theo hàm số mũ
 - Tiếp tục tăng đến một ngưỡng nào đó
- Tránh tắc nghẽn
 - Tăng dần tốc độ theo hàm tuyến tính cho đến khi phát hiện tắc nghẽn
- Phát hiện tắc nghẽn
 - Nếu gói tin bị mất



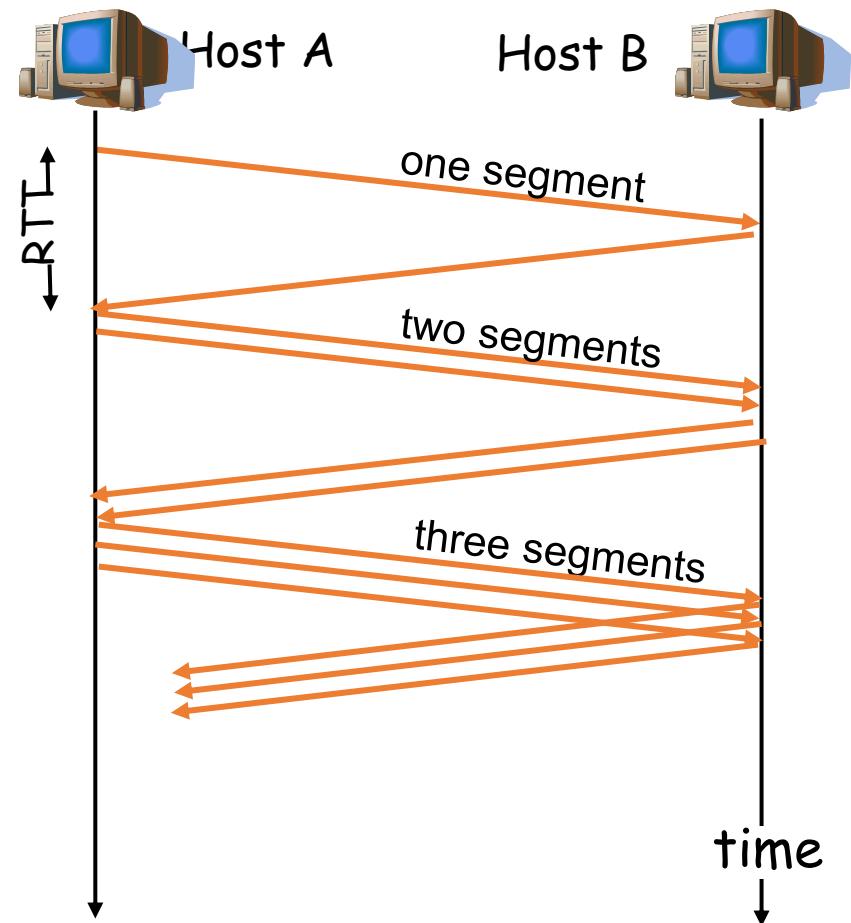
TCP slow start (khởi động chậm)

- Khi bắt đầu kết nối, tăng tốc độ gửi theo cấp số nhân đến khi mất gói hay đến ngưỡng :
 - Bắt đầu **cwnd** = 1 MSS
 - Gấp đôi **cwnd** mỗi RTT
 - Thực hiện tăng cwnd cho mỗi ACK nhận được
- **Tổng kết:** bắt đầu chậm nhưng tăng tốc nhanh
- Tăng cho đến một ngưỡng: ssthresh. Sau đó, TCP chuyển sang trạng thái tránh tắc nghẽn



Tránh tắc nghẽn - Congestion avoidance

- Sau mỗi RTT tăng cwnd thêm 1 MSS
- ➔ Tăng cwnd theo cấp số cộng sau khi nó đạt tới ssthresh



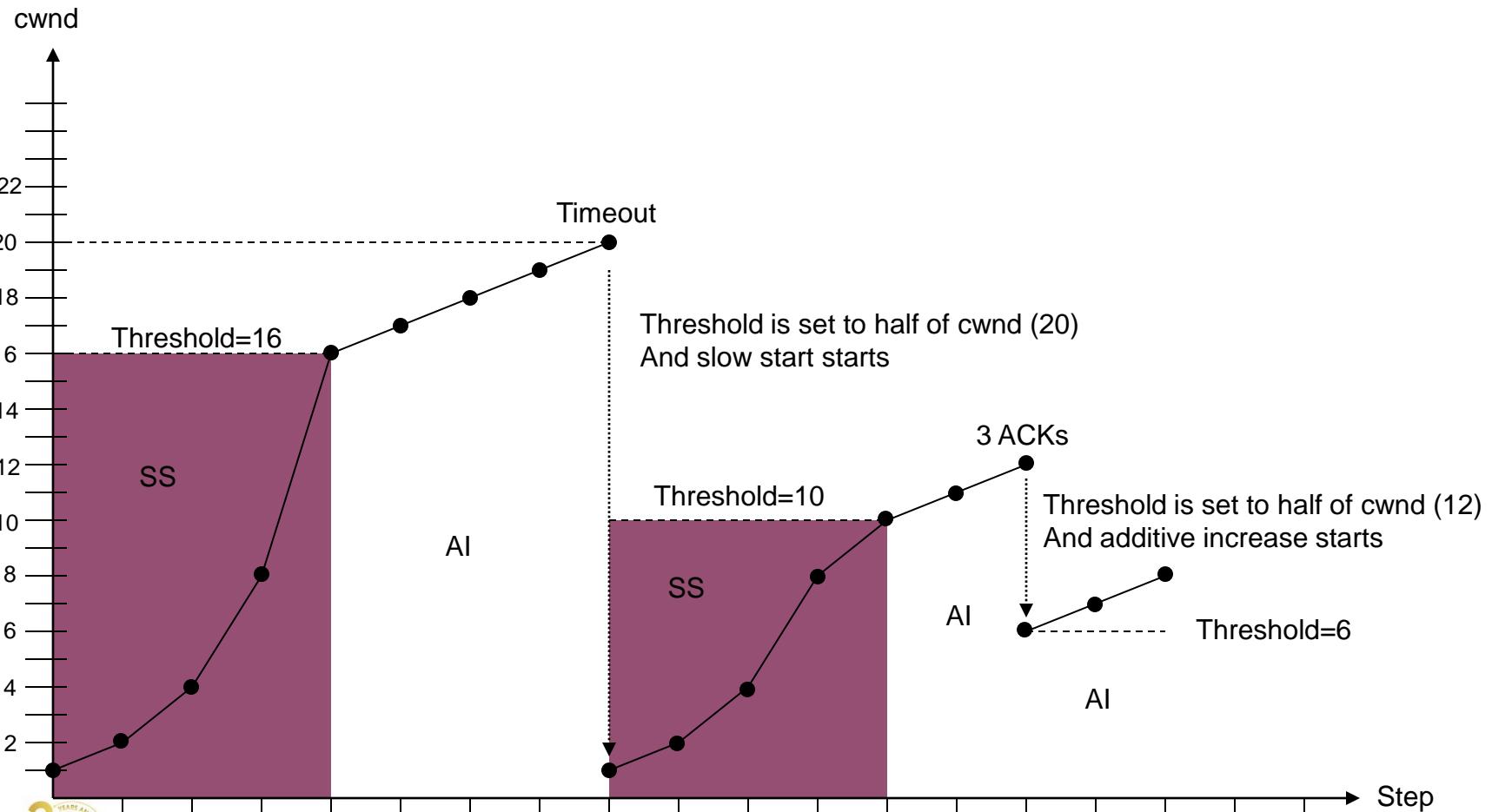
Phản ứng của TCP (1)

- Giảm tốc độ gửi
- Phát hiện tắc nghẽn?
 - Nếu như phải truyền lại
 - Có thể suy ra là mạng “tắc nghẽn”
- Khi nào thì phải truyền lại?
 - Timeout!
 - Cùng một gói tin số hiệu gói tin trong ACK

Phản ứng của TCP (2)

- Khi có timeout của bên gửi
 - TCP đặt ngưỡng ssthresh xuống còn một nửa giá trị hiện tại của cwnd
 - TCP đặt cwnd về 1 MSS
 - TCP chuyển về slow start
- Nếu nhận được 3 ACK giống nhau
 - TCP đặt ngưỡng xuống còn một nửa giá trị hiện tại của cwnd
 - TCP chuyển trạng thái “congestion avoidance”

Kiểm soát tắc nghẽn – minh họa



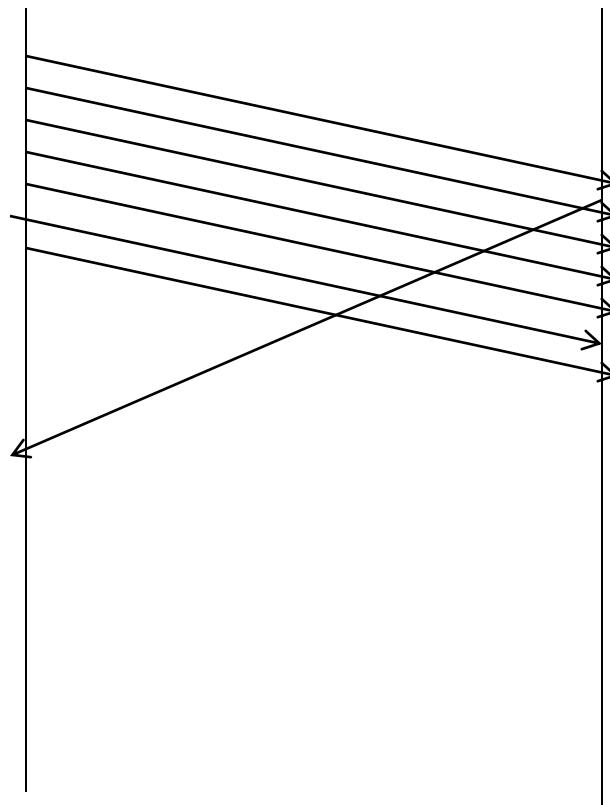
Tổng kết

- Còn rất nhiều chi tiết về TCP!
- Có hai dạng giao thức giao vận
 - UDP và TCP
 - Best effort vs. reliable transport protocol
- Các cơ chế bảo đảm độ tin cậy
 - Báo nhận
 - Truyền lại
 - Kiểm soát luồng và kiểm soát tắc nghẽn

Bài tập

- Giả sử cần truyền 1 file
 - Kích thước $O = 100\text{KB}$ trên kết nối TCP
 - S là kích thước mỗi gói TCP, $S = 536 \text{ byte}$
 - $RTT = 3 \text{ ms.}$
- Giả sử cửa sổ nhận của TCP là cố định với kích thước W .
 - Thời gian truyền với phương pháp Stop-and-wait
 - Thời gian truyền với phương pháp cửa sổ trượt với kích thước cửa sổ =7
 - Hỏi kích thước cửa sổ nhận là bao nhiêu để thời gian chờ đợi ít nhất? Nếu tốc độ đường truyền là
 - $R = 20 \text{ Mbit/s}; R= 100 \text{ Mbits/s.}$

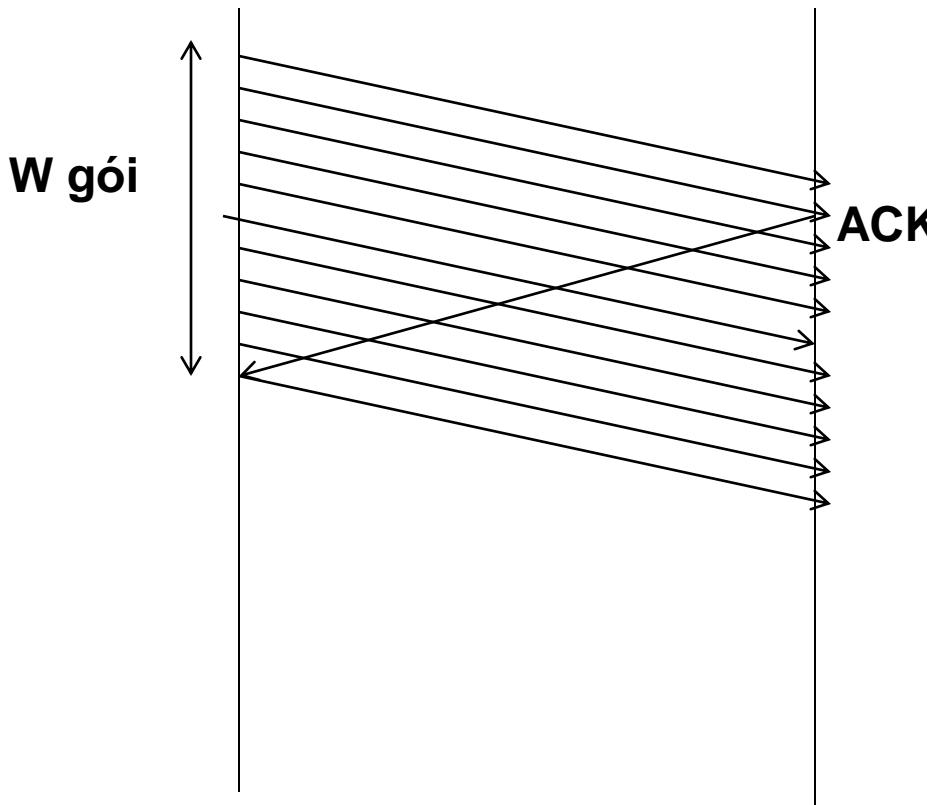
Cơ chế cửa sổ trượt



Thời gian truyền với cửa sổ 7

- T truyền nhanh nhất = (T phát 7 gói + chờ) *số lần.
- 1 lần Chờ = (T phát 1 gói + RTT) – T phát 7 gói
- Số lần chờ = số gói /7

Thời gian truyền nhanh nhất với cửa sổ trượt



- Truyền nhanh nhất đạt được khi nguồn phát xong gói cuối của cửa sổ thì đã nhận được ACK của gói đầu tiên.
- Kích thước cửa sổ W
- $T_{phát} (W \text{ gói}) \geq T_{phát \text{ gói đầu}} + RTT$

Thời gian truyền nhanh nhất với cửa sổ trượt (cont.)

- $T_{\text{phát}} (\text{W gói}) = W * S/R$
- Phát ngắn nhất khi không phải chờ:
- $\Rightarrow (W-1)*S/R \geq RTT$
- $\Rightarrow W \geq RTT * R/S + 1$
- Thời gian phát hết dữ liệu $L = L/R + RTT$
- $R=100 \text{ Mbps}$
 - $W \geq 3\text{ms} * 100 \text{ Mbps} / (536*8) + 1$

Tuần tới: Application Layer

- Application service model
 - Client-server vs. P2P
- Typical applications and protocols
 - HTTP
 - Mail
 - FTP
 - P2P file sharing
 -
 - and your applications?

Acknowledgment

- Bài giảng có sử dụng các hình vẽ từ
 - Tài liệu của trường đại học Keio và Ritsumeikan
 - Tài liệu “Computer Network, a top down approach” của J.F Kurose và K.W. Ross