

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
Posts & Telecommunications Institute of Technology



Mạng Máy Tính

KHOA CÔNG NGHỆ THÔNG TIN 1

GV: Nguyễn Thanh Thủy

Thông tin môn học

➤ Trang web:

- <http://tinyurl.com/mmt2016>

➤ Tài liệu:

- J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet (6nd edition)*, Addison-Wesley, 2012.
- *Slide* (giảng viên cung cấp)
- *Bài giảng Mạng máy tính*, Học viện CN BCVT (2015).
- Nguyễn Thúc Hải, *Mạng máy tính và các hệ thống mở*, Nhà xuất bản giáo dục, 1999

Mục tiêu

- Lý thuyết
 - Kiến trúc và các phần tử của mạng Internet
 - Các tầng trong mô hình TCP/IP và giao thức tương ứng.
 - Hiệu năng mạng
- Thực hành
 - Thực hành các phần mềm tiện ích mạng đơn giản
 - Hiểu về TCP/IP và các thuật ngữ mạng
- Cách tiếp cận
 - Từ trên xuống: từ tầng ứng dụng tới tầng liên kết
 - Sử dụng Internet làm ví dụ

Cách tổ chức

- Môn học bao gồm
 - Bài giảng + bài tập
 - Thực hành
 - Bài tập lớn
- Sinh viên
 - Báo cáo bài tập lớn và các bài tập được giao

Điểm thành phần

- Điểm thành phần như sau:
 - Chuyên cần + thái độ học tập trên lớp: 10%
(*điểm danh và kiểm tra trong quá trình học*)
 - Kiểm tra giữa kỳ (1-2 bài), bài tập lớn, thực hành: 20%
 - Thi cuối kì: 70%

Mô tả nội dung môn học

Chương 1_1: Giới thiệu chung về mạng máy tính

- Lịch sử phát triển
- Định nghĩa mạng máy tính
- Đường truyền vật lý
- Kiến trúc mạng
- Phân loại mạng máy tính
- Kiến trúc phân tầng
- Mô hình OSI (Open Systems Interconnection)
- Hệ điều hành mạng

Mô tả nội dung môn học

Chương 1_2: Giới thiệu về mạng Internet

- 1.1 Internet là gì?
- 1.2 Phần cạnh của mạng
 - Hệ thống đầu cuối, mạng truy nhập, liên kết
- 1.3 Phần lõi của mạng
 - Chuyển mạch gói, chuyển mạch kênh, cấu trúc mạng
- 1.4 Trễ, mất mát, thông lượng trong mạng
- 1.5 Các tầng giao thức, các mô hình dịch vụ
- 1.6 Các mạng bị tấn công: vấn đề an ninh mạng
- 1.7 Lịch sử phát triển mạng Internet

7

Mô tả nội dung môn học

Chương 2: Tầng ứng dụng

- 2.1 Nguyên lý của ứng dụng mạng
 - Kiến trúc của ứng dụng
 - Các yêu cầu của ứng dụng
- 2.2 Web và HTTP
- 2.3 FTP
- 2.4 Thư điện tử
 - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 Ứng dụng P2P
- 2.7 Lập trình socket với UDP và TCP

8

Mô tả nội dung môn học

Chương 3: Tầng giao vận

- 3.1 Các dịch vụ tầng giao vận
- 3.2 Ghép kênh và phân kênh
- 3.3 Vận chuyển không kết nối: UDP
- 3.4 Các nguyên lý truyền dữ liệu tin cậy
- 3.5 Vận chuyển hướng kết nối: TCP
 - Cấu trúc đoạn dữ liệu (segment)
 - Truyền dữ liệu tin cậy
 - Điều khiển luồng
 - Quản lý kết nối
- 3.6 Các nguyên lý điều khiển tắc nghẽn
- 3.7 Điều khiển tắc nghẽn TCP

← MIDTERM
EXAM (approx)

9

Mô tả nội dung môn học

Chương 4: Tầng mạng

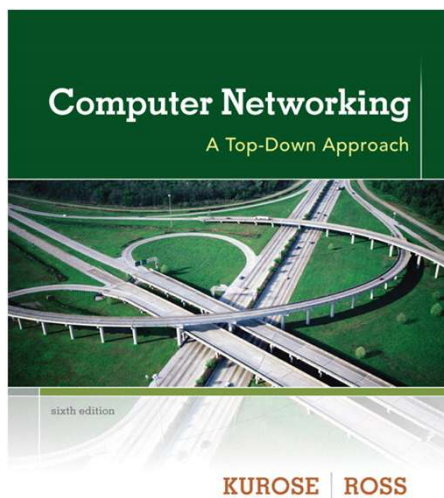
- 4.1 Giới thiệu
- 4.2 Các mạng mạch ảo và mạng chuyển gói
- 4.3 Kiến trúc của bộ định tuyến
- 4.4 IP: Internet Protocol
 - Định dạng gói tin
 - Định địa chỉ IPv4
 - ICMP
 - IPv6
- 4.5 Các giải thuật định tuyến: Link state, Distance vector, Hierarchical routing
- 4.6 Định tuyến trong mạng Internet: RIP, OSPF, BGP

10

Mô tả nội dung môn học

Chương 5: Tầng liên kết dữ liệu và mạng LAN

- 5.1 Giới thiệu, các dịch vụ
- 5.2 Phát hiện và sửa lỗi
- 5.3 Các giao thức đa truy nhập
- 5.4 Các mạng LAN
 - Định địa chỉ, ARP
 - Ethernet
 - Các switch
 - Các VLAN
- 5.5 Chuyển mạch nhãn đa giao thức (MPLS)
- 5.6 Mạng trung tâm dữ liệu
- 5.7 Vòng đời của một yêu cầu web



Chương 1 Giới thiệu

Phần 1: Giới thiệu
mạng máy tính

J.F Kurose and K.W. Ross, All Rights Reserved

1

CÁC KHÁI NIỆM CƠ BẢN VỀ MẠNG MÁY TÍNH

2

CÁC KHÁI NIỆM CƠ BẢN

- Lịch sử phát triển
- Định nghĩa mạng máy tính
- Đường truyền vật lý
- Kiến trúc mạng
- Phân loại mạng máy tính
- Kiến trúc phân tầng
- Mô hình OSI (Open Systems Interconnection)
- Hệ điều hành mạng

3

LỊCH SỬ PHÁT TRIỂN

- 60s: *Mạng xử lý với bộ tiền xử lý*
 - 70s: Các máy tính được nối với nhau trực tiếp.
Xuất hiện khái niệm *Mạng truyền thông (Communication network)*: các nút mạng là các *bộ chuyển mạch*, hướng thông tin tới đích.
- ⇒ Các máy tính được nối thành mạng máy tính nhằm đạt tới các mục tiêu chính sau:
- Làm cho các tài nguyên có giá trị cao trở nên khả dụng đối với bất kỳ người sử dụng nào trên mạng.
 - Tăng độ tin cậy của hệ thống (do có khả năng thay thế khi xảy ra sự cố đối với một máy tính nào đó).

4

ĐỊNH NGHĨA MẠNG MÁY TÍNH

Mạng máy tính: Là một tập hợp các máy tính được nối với nhau bởi các *đường truyền vật lý* theo một *kiến trúc* nào đó.

5

ĐƯỜNG TRUYỀN VẬT LÝ

- Được dùng để chuyển các tín hiệu điện tử (các giá trị dữ liệu dưới dạng các xung nhị phân) giữa các máy tính.
- Tất cả các tín hiệu này đều thuộc một dạng sóng điện từ nào đó, trải từ tần số radio tới sóng cực ngắn (viba) và tia hồng ngoại.
- Tùy theo tần số của sóng điện từ có thể dùng các đường truyền vật lý khác nhau để truyền các tín hiệu.

6

ĐƯỜNG TRUYỀN VẬT LÝ

- Các tần số radio: Có thể truyền bằng cáp điện (giây đôi xoắn hoặc cáp đồng trục) hoặc bằng phương tiện quảng bá (radio broadcasting).
- Sóng cực ngắn (viba): Thường được dùng để truyền giữa các trạm mặt đất và các trạm vệ tinh hoặc truyền các tín hiệu quảng bá từ một trạm phát tới nhiều trạm thu.
- Tia hồng ngoại: Có thể được truyền giữa 2 điểm hoặc quảng bá từ một điểm đến nhiều máy thu. Tia hồng ngoại và các tần số cao hơn của ánh sáng có thể được truyền qua các loại cáp sợi quang.

7

ĐƯỜNG TRUYỀN VẬT LÝ

Các đặc trưng cơ bản của đường truyền vật lý:

- *Giải thông (bandwidth)*: Là độ đo phạm vi tần số mà đường truyền có thể đáp ứng được.
Ví dụ: Giải thông của đường điện thoại là 400-4000Hz.
Giải thông của cáp phụ thuộc vào độ dài cáp
=> khi thiết kế cáp cho mạng phải chỉ rõ độ dài chạy cáp tối đa.
- *Thông lượng (throughput)*: Là tốc độ truyền dữ liệu trên đường truyền, thường được tính bằng số lượng bit được truyền đi trong một giây (bps).
Thông lượng còn được đo bằng một đơn vị khác là *baud*: biểu thị số lượng thay đổi tín hiệu trong một giây.

ĐƯỜNG TRUYỀN VẬT LÝ

Các đặc trưng cơ bản của đường truyền vật lý:

- *Độ suy hao*: Là độ đo sự yếu đi của tín hiệu trên đường truyền.
- *Độ nhiễu điện từ (EMI-Electromagnetic Interference)*: Gây ra bởi tiếng ồn điện từ bên ngoài làm ảnh hưởng đến tín hiệu đường truyền.

9

ĐƯỜNG TRUYỀN VẬT LÝ

Phân loại:

- Đường truyền hữu tuyến (cable):
 - Cáp đồng trục (Coaxial cable).
 - Cáp đôi xoắn (twisted-pair cable), gồm 2 loại: có bọc kim (Shielded) và không bọc kim (Unshielded).
 - Cáp sợi quang (fiber-optic cable).
- Đường truyền vô tuyến (wireless)
 - Radio
 - Sóng cực ngắn (Viba) (Microwave)
 - Tia hồng ngoại (infrared).

10

KIẾN TRÚC MẠNG

- **Kiến trúc mạng máy tính (Network Architecture):** Thể hiện cách nối các máy tính với nhau ra sao và tập hợp các quy tắc, quy ước mà tất cả các thực thể tham gia truyền thông trên mạng phải tuân theo để đảm bảo cho mạng hoạt động tốt.
- Cách nối các máy tính được gọi là *hình trạng (topology)* của mạng hay gọi tắt là *topo* của mạng.
- Tập hợp các quy tắc, quy ước truyền thông được gọi là *giao thức (protocol)* của mạng.

11

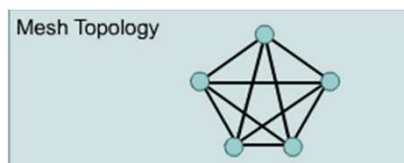
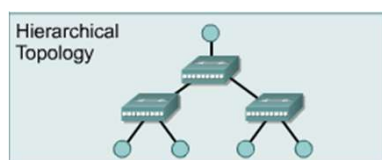
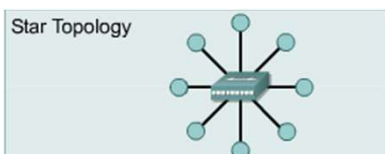
KIẾN TRÚC MẠNG

- **Topo mạng:** 2 kiểu chủ yếu:
 - **Điểm – điểm (point-to-point):** các đường truyền nối từng cặp nút với nhau. Mỗi nút có trách nhiệm lưu trữ tạm thời, sau đó chuyển tiếp dữ liệu đi tới đích
=> Mạng này còn được gọi là mạng “lưu và chuyển tiếp” (store-and-forward).
 - **Quảng bá (broadcast hay point-to-multipoint):** Tất cả các nút phân chia chung một đường truyền vật lý. Dữ liệu được gửi đi từ một nút nào đó sẽ có thể được tiếp nhận bởi tất cả các nút còn lại => chỉ cần chỉ ra địa chỉ đích của dữ liệu để mỗi nút tự kiểm tra xem dữ liệu đó có phải gửi cho mình hay không.

12

KIẾN TRÚC MẠNG

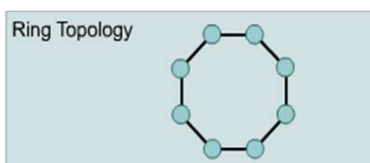
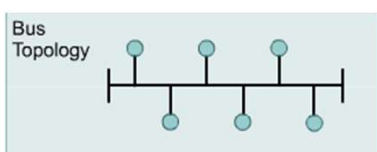
Một số dạng topo của mạng kiểu điểm – điểm:



13

KIẾN TRÚC MẠNG

Một số dạng topo của mạng kiểu quảng bá:



⇒ Cần có cơ chế để giải quyết xung đột khi nhiều nút muốn truyền tin một lúc. Có 2 dạng cấp phát:

- “Tĩnh”: Phân chia theo khoảng thời gian định trước.
- “Động”: Cấp phát theo yêu cầu => hạn chế được thời gian “chết” vô ích của đường truyền.

14

KIẾN TRÚC MẠNG

- Giao thức mạng:

- Khi truyền tín hiệu trên mạng, cần phải có các quy tắc, quy ước về nhiều mặt, từ khuôn dạng (cú pháp, ngữ nghĩa) của dữ liệu đến các thủ tục gửi, nhận dữ liệu, kiểm soát hiệu quả và chất lượng truyền tin, và xử lý các lỗi và sự cố.
- Các mạng có thể sử dụng các giao thức khác nhau tùy sự lựa chọn của người thiết kế.

15

PHÂN LOẠI MẠNG MÁY TÍNH

- Dựa vào khoảng cách địa lý:

- Mạng cục bộ (LAN – Local Area Networks): Được cài đặt trong phạm vi tương đối nhỏ (ví dụ, trong một tòa nhà, khu trường học, ...)
- Mạng đô thị (MAN – Metropolitan Area Networks): Là mạng được cài đặt trong phạm vi một đô thị hoặc một trung tâm kinh tế - xã hội.
- Mạng diện rộng (WAN – Wide Area Networks): Phạm vi của mạng có thể vượt qua biên giới quốc gia.
- Mạng toàn cầu (GAN – Global Area Networks): Phạm vi của mạng trải rộng khắp lục địa.

16

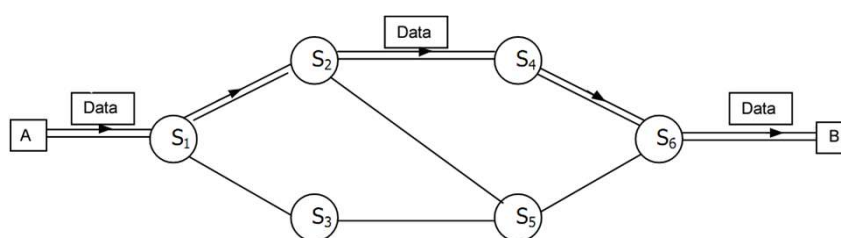
PHÂN LOẠI MẠNG MÁY TÍNH

- Dựa vào kỹ thuật chuyển mạch:
 - Mạng chuyển mạch kênh (circuit – switched networks)
 - Mạng chuyển mạch thông báo (message – switched networks)
 - Mạng chuyển mạch gói (package – switched networks)

17

PHÂN LOẠI MẠNG MÁY TÍNH

- Mạng chuyển mạch kênh (circuit – switched networks):



18

PHÂN LOẠI MẠNG MÁY TÍNH

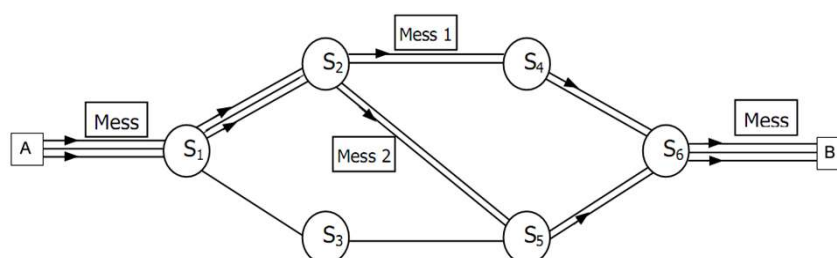
- Mạng chuyển mạch kênh (circuit – switched networks):

- Hai thực thể thiết lập một “kênh” (circuit) cố định để trao đổi thông tin. Kênh này được duy trì đến khi một trong hai bên ngắt liên lạc.
- Nhược điểm:
 - ✓ Tiêu tốn thời gian để thiết lập kênh cố định giữa hai thực thể.
 - ✓ Hiệu suất đường truyền không cao vì có lúc kênh bị bỏ không do hai bên đều hết thông tin cần truyền trong khi các thực thể khác không được phép sử dụng kênh này.

19

PHÂN LOẠI MẠNG MÁY TÍNH

- Mạng chuyển mạch thông báo (message – switched networks):



20

PHÂN LOẠI MẠNG MÁY TÍNH

- **Mạng chuyển mạch thông báo (message – switched networks):**
 - Thông báo (message) là một đơn vị thông tin của người dùng có khuôn dạng được quy định trước. Mỗi thông báo đều có chứa vùng thông tin điều khiển trong đó chỉ rõ đích của thông báo.
 - Mỗi nút đều phải lưu trữ tạm thời thông báo để “đọc” thông tin điều khiển trên thông báo. Sau đó mới quyết định chuyển tiếp thông báo đi hay không.
 - Tùy thuộc vào điều kiện mạng, các thông báo khác nhau có thể được gửi đi theo những đường khác nhau.

21

PHÂN LOẠI MẠNG MÁY TÍNH

- **Mạng chuyển mạch thông báo:**
 - Ưu điểm so với chuyển mạch kênh:**
 - Hiệu suất sử dụng đường truyền cao vì không bị chiếm dụng độc quyền mà được phân chia giữa nhiều thực thể.
 - Mỗi nút mạng có thể lưu trữ thông báo cho tới khi kênh truyền rồi mới gửi thông báo đi => giảm được tình trạng *tắc nghẽn* mạng.
 - Có thể điều khiển việc truyền tin bằng cách sắp xếp độ ưu tiên cho các thông báo.
 - Tăng hiệu suất sử dụng giải thông của mạng bằng cách gán địa chỉ quảng bá để gửi thông báo tới nhiều đích.

PHÂN LOẠI MẠNG MÁY TÍNH

- Mạng chuyển mạch thông báo:

Nhược điểm:

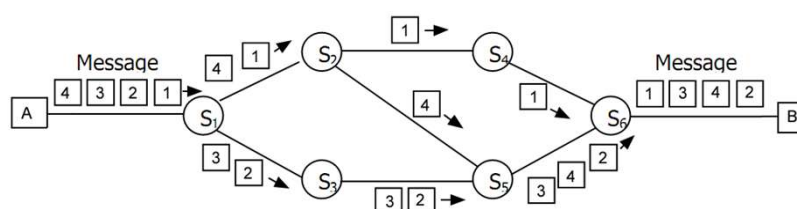
- Do không hạn chế kích thước của thông báo có thể dẫn đến phí tổn lưu trữ tạm thời cao => ảnh hưởng đến thời gian đáp ứng và chất lượng truyền.

=> Mạng chuyển mạch thông báo thích hợp với các dịch vụ thông tin kiểu thư điện tử (email) hơn là đối với các ứng dụng có tính thời gian thực do tồn tại độ trễ nhất định bởi quá trình lưu trữ và xử lý thông tin điều khiển tại mỗi nút.

23

PHÂN LOẠI MẠNG MÁY TÍNH

- Mạng chuyển mạch gói (package – switched networks):



24

PHÂN LOẠI MẠNG MÁY TÍNH

- Mạng chuyển mạch gói (package – switched networks):

- Mỗi thông báo được chia thành nhiều phần nhỏ hơn gọi là các *gói tin* (package) có khuôn dạng định trước.
- Mỗi gói tin chứa các thông tin điều khiển, trong đó có địa chỉ nguồn và đích của gói tin.
- Các gói tin thuộc về một thông báo nào đó có thể được gửi đi qua mạng để tới đích bằng nhiều đường khác nhau.

⇒ 2 phương pháp chuyển mạch thông báo và chuyển mạch gói là gần giống nhau.

25

PHÂN LOẠI MẠNG MÁY TÍNH

- Mạng chuyển mạch gói khác với chuyển mạch thông báo:

- Các gói tin được giới hạn kích thước tối đa sao cho các nút mạng có thể xử lý toàn bộ gói tin trong bộ nhớ mà không phải lưu trữ tạm thời trên đĩa.

⇒ Mạng chuyển mạch gói truyền các gói tin qua mạng nhanh hơn và hiệu quả hơn so với mạng chuyển mạch thông báo.

26

PHÂN LOẠI MẠNG MÁY TÍNH

- Mạng chuyển mạch gói:

Ưu điểm:

- Mềm dẻo, hiệu suất cao => được dùng phổ biến hơn các mạng chuyển mạch thông báo.

Nhược điểm:

- Khó khăn khi tập hợp các gói tin để tạo lại thông báo ban đầu của người dùng, đặc biệt trong trường hợp các gói tin được truyền theo nhiều đường khác nhau.

⇒ Cần phải cài đặt các cơ chế “đánh dấu” gói tin và phục hồi các gói tin bị thất lạc hoặc truyền bị lỗi cho các nút mạng.

27

PHÂN LOẠI MẠNG MÁY TÍNH

- Ngoài ra, còn có thể phân loại mạng theo kiến trúc mạng (topo và giao thức sử dụng).

- Ví dụ:

- Mạng ISO (theo kiến trúc chuẩn Quốc tế),
- Mạng TCP/IP,
- Mạng SNA của IBM,
- ...

28

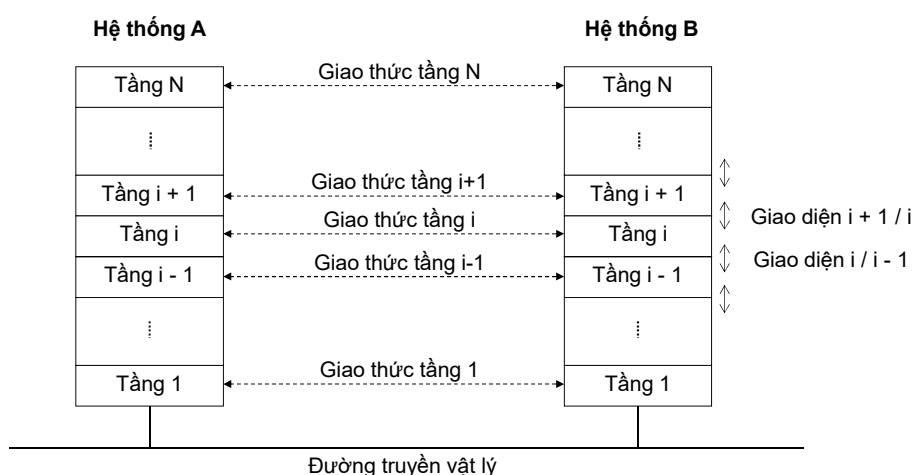
KIẾN TRÚC PHÂN TẦNG

- Mục đích của việc *phân tầng (layering)*: Giảm độ phức tạp của việc thiết kế và cài đặt mạng.
- Mỗi hệ thống mạng là một cấu trúc đa tầng, trong đó mỗi tầng được xây dựng trên tầng trước nó.
- Số lượng tầng, tên và chức năng của mỗi tầng tùy thuộc vào người thiết kế.
- Mỗi tầng cung cấp một số *dịch vụ (services)* nhất định cho tầng cao hơn.

29

KIẾN TRÚC PHÂN TẦNG

Minh họa kiến trúc phân tầng tổng quát:



30

KIẾN TRÚC PHÂN TẦNG

Nguyên tắc của kiến trúc mạng phân tầng:

- Các hệ thống trong cùng mạng có cấu trúc tầng như nhau (số lượng tầng, chức năng của mỗi tầng).
- Hai hệ thống kết nối với nhau: Định nghĩa mối quan hệ (giao diện) giữa 2 tầng kề nhau, và mối quan hệ (giao diện) giữa 2 tầng đồng mức.
- Cách truyền và nhận dữ liệu: Dữ liệu bên *hệ thống gửi (sender)* từ các tầng trên được chuyển xuống tầng dưới cùng, qua đường truyền vật lý truyền sang *hệ thống nhận (receiver)* và cứ thế đi ngược lên các tầng trên.

31

KIẾN TRÚC PHÂN TẦNG

Nguyên tắc của kiến trúc mạng phân tầng (tiếp):

- ⇒ Giữa hai hệ thống kết nối với nhau, chỉ có tầng thấp nhất mới có *liên kết vật lý*, còn ở các tầng cao hơn chỉ là những *liên kết logic (liên kết ảo)*.
- ⇒ Ưu điểm của phân tầng:
 - Cho phép xác định cụ thể quan hệ giữa các thành phần.
 - Bảo trì, nâng cấp dễ dàng.
- ⇒ Nếu không phân tầng: Khi có công nghệ mạng mới phải viết lại các ứng dụng => rất tốn kém.

32

MÔ HÌNH OSI (Open Systems Interconnection)

- Người thiết kế tự do lựa chọn kiến trúc mạng, dẫn đến tình trạng không tương thích giữa các mạng, về phương pháp truy nhập đường truyền, các họ giao thức,... => Gây khó khăn cho những tương tác của người dùng mạng khác nhau.
- ⇒ Cần có một khung chuẩn về kiến trúc mạng để làm căn cứ cho người thiết kế và chế tạo các sản phẩm về mạng.

33

MÔ HÌNH OSI (Open Systems Interconnection)

- ⇒ Tổ chức tiêu chuẩn hóa quốc tế (*ISO – International Standard Organization*), xuất phát từ kiến trúc phân tầng, đã xây dựng *Mô hình tham chiếu cho việc kết nối các hệ thống mở (Reference Model for Open Systems Interconnection*, gọi tắt là *OSI Reference Model*), phục vụ cho các ứng dụng phân tán.

34

MÔ HÌNH OSI (Open Systems Interconnection)

Nguyên tắc xây dựng:

1. Để đơn giản, cần hạn chế số lượng các tầng.
2. Tạo ranh giới các tầng sao cho các tương tác và mô tả các dịch vụ là tối thiểu.
3. Chia các tầng theo các chức năng và các loại công nghệ tách biệt nhau.
4. Các chức năng giống nhau được đặt vào một tầng.
5. Chọn ranh giới các tầng theo kinh nghiệm đã được chứng minh là thành công.

35

MÔ HÌNH OSI (Open Systems Interconnection)

Nguyên tắc xây dựng (tiếp):

6. Các chức năng được định vị sao cho có thể thiết kế lại tầng mà ảnh hưởng ít nhất đến các tầng kề nó.
7. Tạo ranh giới các tầng sao cho có thể chuẩn hóa giao diện tương ứng.
8. Tạo một tầng khi dữ liệu được xử lý một cách khác biệt.
9. Cho phép thay đổi chức năng hoặc giao thức trong một tầng không làm ảnh hưởng đến các tầng khác.

36

MÔ HÌNH OSI (Open Systems Interconnection)

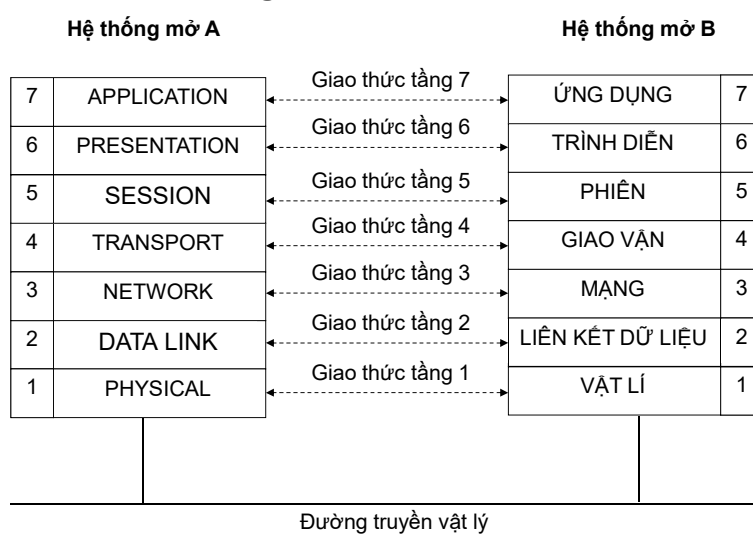
Nguyên tắc xây dựng (tiếp):

10. Mỗi tầng chỉ có các ranh giới (giao diện) với các tầng trên và dưới nó.
11. Có thể chia một tầng thành các tầng con khi cần thiết.
12. Tạo các tầng con để cho phép giao diện với các tầng kế cận.
13. Cho phép hủy bỏ các tầng con nếu thấy không cần thiết.

37

MÔ HÌNH OSI (Open Systems Interconnection)

Mô hình OSI 7 tầng:



38

MÔ HÌNH OSI (Open Systems Interconnection)

Chức năng các tầng trong mô hình OSI:

1. **Tầng vật lý:** Truyền dòng bit không có cấu trúc qua đường truyền vật lý, truy nhập đường truyền vật lý nhờ các phương tiện cơ, điện, hàm, thủ tục.
2. **Tầng liên kết dữ liệu:** Cung cấp phương tiện truyền thông tin qua liên kết vật lý đảm bảo tin cậy; gửi các khối dữ liệu (frame) với các cơ chế đồng bộ hóa, kiểm soát lỗi và kiểm soát luồng dữ liệu cần thiết.
3. **Tầng mạng:** Chọn đường và chuyển tiếp thông tin với công nghệ chuyển mạch thích hợp, thực hiện kiểm soát luồng dữ liệu và cắt/hợp dữ liệu nếu cần.

39

MÔ HÌNH OSI (Open Systems Interconnection)

Chức năng các tầng trong mô hình OSI (tiếp):

4. **Tầng giao vận:** Truyền dữ liệu, kiểm soát lỗi và luồng dữ liệu giữa hai đầu end-to-end. Có thể ghép kênh (multiplexing), cắt/hợp dữ liệu nếu cần.
5. **Tầng phiên:** Cung cấp phương tiện quản lý truyền thông giữa các ứng dụng. Thiết lập, duy trì, đồng bộ hóa và hủy bỏ các phiên truyền thông giữa các ứng dụng.

40

MÔ HÌNH OSI (Open Systems Interconnection)

Chức năng các tầng trong mô hình OSI (tiếp):

6. **Tầng trình diễn:** Chuyển đổi cú pháp dữ liệu để đáp ứng yêu cầu truyền dữ liệu của các ứng dụng qua môi trường OSI.
7. **Tầng ứng dụng:** Cung cấp các phương tiện để người dùng có thể truy nhập được vào môi trường OSI, đồng thời cung cấp các dịch vụ thông tin phân tán.

41

MÔ HÌNH OSI (Open Systems Interconnection)

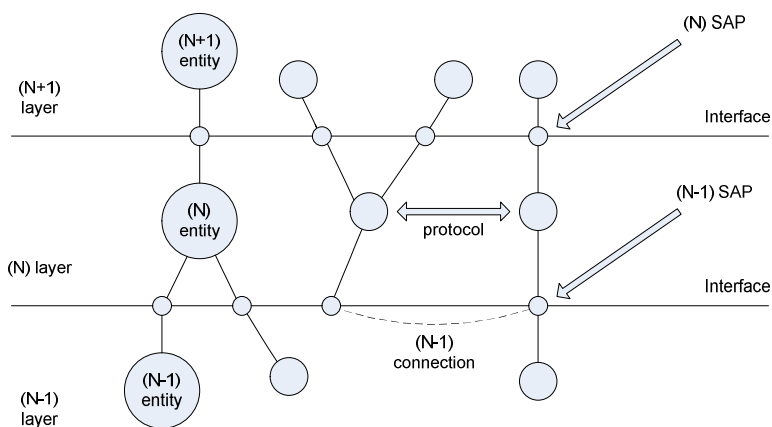
Quan niệm về tầng theo tiếp cận OSI:

- Mỗi tầng có một hoặc nhiều *thực thể (entity)* hoạt động. Một (N)entity (thực thể của tầng N) cài đặt các chức năng của tầng N và giao thức truyền thông với các (N)entity trong các hệ thống khác.
- Mỗi thực thể truyền thông với các thực thể ở tầng trên và tầng dưới nó qua một *giao diện (interface)*, gồm một hoặc nhiều *điểm truy nhập dịch vụ (SAP – Service Access Point)*.

42

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI (tiếp):



43

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI (tiếp):

- (N-1)entity cung cấp dịch vụ cho một N(entity) thông qua việc gọi các *hàm nguyên thủy (primitive)*. Hàm này chỉ rõ chức năng cần thực hiện và được dùng để truyền dữ liệu và thông tin điều khiển.
- Có 4 kiểu hàm nguyên thủy được sử dụng:
 - Request (yêu cầu)
 - Indication (chỉ báo)
 - Response (trả lời)
 - Confirm (xác nhận)

44

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI (tiếp):

Bốn kiểu hàm nguyên thủy:

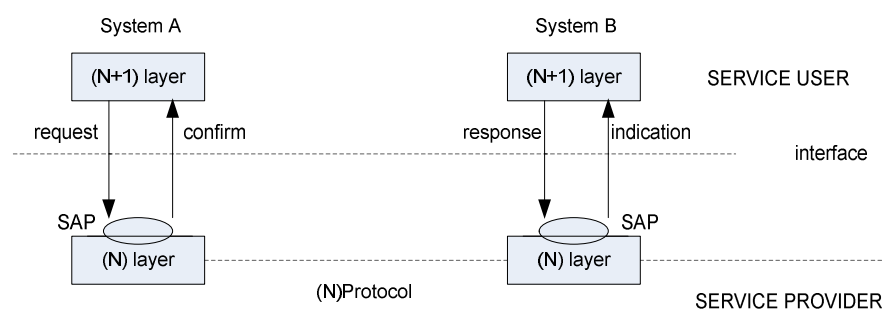
- **Request:** Service User dùng để gọi một chức năng.
- **Indication:** Service Provider dùng để: (1) Gọi một chức năng; (2) Chỉ báo một chức năng đã được gọi ở một điểm truy nhập dịch vụ (SAP).
- **Response:** Service User dùng để hoàn tất một chức năng đã được gọi từ trước bởi một hàm Indication ở SAP đó.
- **Confirm:** Service Provider dùng để hoàn tất một chức năng đã được gọi từ trước bởi một hàm Request tại SAP đó.

45

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI:

Sơ đồ nguyên lý hoạt động của các hàm nguyên thủy:



46

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI (tiếp):

Nguyên lý hoạt động của các hàm nguyên thủy:

- Tầng (N+1) của A gửi xuống tầng (N) kèm dưới một hàm Request.
- Tầng (N) của A tạo một đơn vị dữ liệu để gửi yêu cầu đó sang tầng (N) của B theo giao thức tầng N đã xác định.
- Nhận được yêu cầu, tầng (N) của B chỉ báo lên tầng (N+1) kèm nó bằng hàm Indication.

47

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI (tiếp):

Nguyên lý hoạt động của các hàm nguyên thủy:

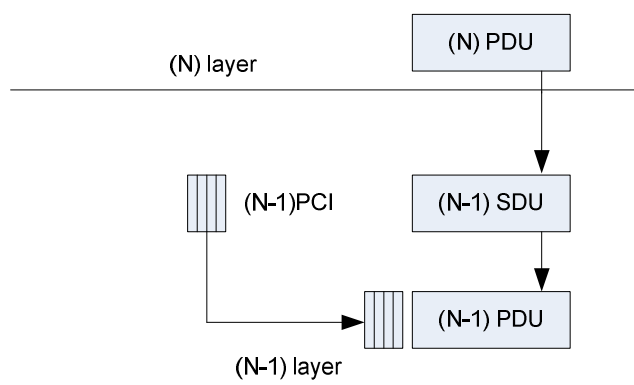
- Tầng (N+1) của B trả lời bằng hàm Response gửi xuống tầng (N) kèm dưới nó.
- Tầng (N) của B tạo một đơn vị dữ liệu để gửi trả lời đó về tầng (N) của A theo giao thức tầng N đã xác định.
- Nhận được trả lời, tầng (N) của A xác nhận với tầng (N+1) kèm trên nó bằng hàm Confirm => kết thúc một giao tác giữa 2 hệ thống.

48

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI (tiếp):

Quan hệ giữa các đơn vị dữ liệu ở các tầng kề nhau:



49

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI (tiếp):

Quan hệ giữa các đơn vị dữ liệu ở các tầng kề nhau:

- Một thực thể ở tầng (N) không thể truyền dữ liệu trực tiếp tới một thực thể tầng (N) ở một hệ thống khác mà phải chuyển xuống dưới để truyền qua bằng tầng thấp nhất (tầng vật lý).
- Khi xuống đến tầng (N-1), dữ liệu được chuyển từ tầng (N) được xem như một đơn vị dữ liệu cho dịch vụ (SDU) của tầng (N-1).
- Phần thông tin điều khiển của tầng (N-1) (gọi là (N-1)PCI) được thêm vào đầu của (N-1)SDU để tạo thành (N-1)PDU.

50

MÔ HÌNH OSI (Open Systems Interconnection)

Quan niệm về tầng theo tiếp cận OSI (tiếp):

Quan hệ giữa các đơn vị dữ liệu ở các tầng kề nhau:

- Nếu (N-1)SDU quá dài thì nó được cắt nhỏ thành nhiều đoạn, mỗi đoạn được bổ sung phần (N-1)PCI ở đầu và tạo thành nhiều (N-1)PDU.
- Quá trình tiếp diễn cho đến tầng vật lý, sau đó dữ liệu sẽ được truyền qua đường truyền vật lý.
- Bên hệ thống nhận, trình tự sẽ diễn ra ngược lại. Qua mỗi tầng PCI tương ứng sẽ được phân tích và sau đó cắt bỏ khỏi các PDU trước khi gửi lên tầng trên.

51

MÔ HÌNH OSI (Open Systems Interconnection)

Phương thức hoạt động trong mô hình OSI:

Gồm 2 phương thức hoạt động chính:

- Có liên kết (connection-oriented)
- Không liên kết (connectionless)

52

MÔ HÌNH OSI (Open Systems Interconnection)

Phương thức hoạt động trong mô hình OSI (tiếp):

Phương thức hoạt động có liên kết:

Trước khi truyền dữ liệu cần thiết lập một liên kết logic giữa các thực thể đồng mức. Quá trình truyền thông bao gồm 3 giai đoạn:

1. **Thiết lập liên kết (logic):** Hai thực thể đồng mức ở hai hệ thống sẽ thương lượng với nhau về tập các tham số sẽ sử dụng trong giai đoạn sau (truyền dữ liệu).
2. **Truyền dữ liệu:** dữ liệu được truyền với các cơ chế kiểm soát và quản lý kèm theo (như kiểm soát lỗi, kiểm soát luồng dữ liệu, cắt/hợp dữ liệu, ...) để tăng cường độ tin cậy và hiệu quả của việc truyền dữ liệu.

MÔ HÌNH OSI (Open Systems Interconnection)

Phương thức hoạt động trong mô hình OSI (tiếp):

Phương thức hoạt động có liên kết:

3. **Hủy bỏ liên kết (logic):** giải phóng các tài nguyên hệ thống đã được cấp phát cho liên kết để dùng cho các liên kết khác.

Phương thức hoạt động không liên kết: Chỉ có duy nhất một giai đoạn **truyền dữ liệu**.

MÔ HÌNH OSI (Open Systems Interconnection)

Phương thức hoạt động trong mô hình OSI (tiếp):

So sánh 2 phương thức hoạt động:

- **Phương thức có liên kết:** cho phép truyền dữ liệu tin cậy, do được kiểm soát và quản lý chặt chẽ theo từng liên kết logic. Tuy nhiên nhược điểm là cài đặt khá phức tạp.
- **Phương thức hoạt động không liên kết:** cho phép các PDU được truyền đi theo nhiều đường khác nhau tới đích => thích nghi với sự thay đổi trạng thái của mạng. Khó khăn là việc tập hợp lại các PDU để chuyển đến người dùng.

55

MÔ HÌNH OSI (Open Systems Interconnection)

Phương thức hoạt động trong mô hình OSI (tiếp):

⇒ Việc lựa chọn phương thức hoạt động cho mỗi tầng phụ thuộc vào yêu cầu tổng hợp về chất lượng, hiệu quả, độ tin cậy,... của việc truyền thông.

Không nhất thiết 2 tầng kề nhau phải sử dụng chung một phương thức hoạt động.

56

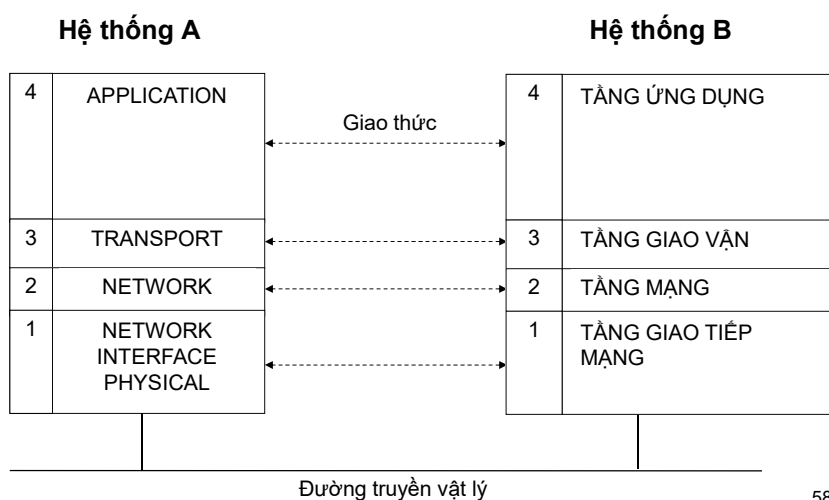
MÔ HÌNH TCP/IP

- TCP/IP (Transmission Control Protocol/Internet Protocol – Giao thức điều khiển truyền/Giao thức liên mạng) là bộ giao thức cùng làm việc với nhau để cung cấp phương tiện truyền liên mạng.
- TCP/IP được phát triển từ thời kỳ đầu của Internet vào năm 1974.
- Mô hình TCP/IP được thiết kế dựa trên họ giao thức TCP/IP.

57

MÔ HÌNH TCP/IP

- Mô hình TCP/IP gồm có 4 tầng:



58

MÔ HÌNH TCP/IP

- Mỗi tầng giải quyết một vấn đề liên quan đến việc truyền dữ liệu, và tầng dưới cung cấp các dịch vụ cho tầng trên nó (tương tự OSI).
- Để đảm bảo tương thích giữa các mạng và sự tin cậy của việc truyền thông tin trên mạng, bộ giao thức TCP/IP được chia thành 2 phần riêng biệt:
 - Giao thức IP sử dụng cho việc kết nối mạng.
 - Giao thức TCP để đảm bảo việc truyền dữ liệu một cách tin cậy.

59

MÔ HÌNH TCP/IP

Chức năng từng tầng trong mô hình TCP/IP:

1. Tầng giao tiếp mạng:

- Là tầng thấp nhất của mô hình TCP/IP, có trách nhiệm nhận các IP datagram và truyền chúng trên một mạng nhất định.
- Chia tầng giao tiếp mạng thành 2 tầng con:
 - Tầng vật lý: Làm việc với các thiết bị vật lý, truyền dòng bit 0, 1 từ nơi gửi đến nơi nhận.
 - Tầng liên kết dữ liệu: Dữ liệu được tổ chức thành các khung (frame). Phần đầu khung chứa địa chỉ và thông tin điều khiển, phần cuối khung dành cho việc phát hiện lỗi.

60

MÔ HÌNH TCP/IP

2. Tầng mạng (Tầng Internet):

- Đảm nhiệm việc chọn lựa đường đi tốt nhất cho các gói tin. Giao thức được sử dụng chính ở tầng này là IP (Internet Protocol).
 - Nhận yêu cầu để gửi gói dữ liệu từ tầng giao vận, cùng với một định danh của máy mà gói dữ liệu cần được gửi đến.
 - Thực hiện đóng gói segment vào trong một packet, gắn vào phần tiêu đề của packet, sau đó sử dụng các giao thức định tuyến để chuyển gói tin đến đích hoặc trạm kế tiếp.
 - Tại nơi nhận sẽ kiểm tra tính hợp lệ, và sử dụng các giao thức định tuyến để xử lý gói tin.
 - Cuối cùng, tầng mạng gửi và nhận các thông điệp kiểm soát và xử lý lỗi ICMP (Internet Control Message Protocol).

61

MÔ HÌNH TCP/IP

3. Tầng giao vận:

- Nhiệm vụ cơ bản của tầng giao vận là cung cấp phương tiện liên lạc từ chương trình ứng dụng này đến chương trình ứng dụng khác, gọi là end-to-end.
- Có thể điều khiển luồng dữ liệu.
- Có thể cung cấp giao vận có độ tin cậy, bảo đảm dữ liệu đến nơi mà không có lỗi và theo đúng thứ tự, bằng cách sử dụng giao thức TCP.
- Trong những môi trường truyền dẫn tốt (ví dụ, cáp quang) thì việc xảy ra lỗi là rất nhỏ. Tầng giao vận cung cấp một giao thức khác, là UDP (User Datagram Protocol).

62

MÔ HÌNH TCP/IP

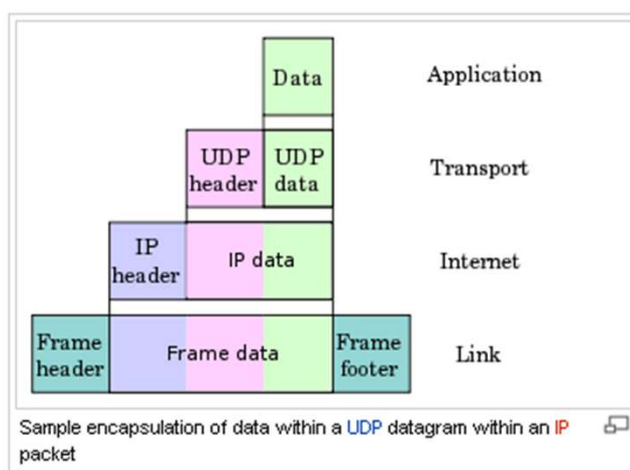
4. Tầng ứng dụng:

- Là tầng cao nhất, trong đó người dùng thực hiện các chương trình ứng dụng truy xuất đến các dịch vụ trên TCP/IP Internet.
- Một ứng dụng tương tác với một trong những giao thức ở tầng giao vận (transport) để gửi hoặc nhận dữ liệu.
- Quản lý các giao thức, hỗ trợ việc trình bày, mã hóa và quản lý cuộc gọi.
- Hỗ trợ nhiều ứng dụng như : FTP (File Transfer Protocol), HTTP (Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), DNS (Domain Name System), ...

63

MÔ HÌNH TCP/IP

Đóng gói dữ liệu trong TCP/IP



64

MÔ HÌNH TCP/IP

So sánh mô hình OSI và TCP/IP

OSI Model	TCP/IP (Internet)
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Network Interface
Physical	Physical

65

MÔ HÌNH TCP/IP

So sánh mô hình OSI và TCP/IP:

Giống nhau:

- Đều phân tầng chức năng.
- Đều có tầng ứng dụng, mặc dù các dịch vụ ở mỗi tầng khác nhau.
- Đều có tầng giao vận và tầng mạng.
- Sử dụng kỹ thuật chuyển mạch gói
- Mối quan hệ giữa các tầng trên dưới và các tầng đồng mức giống nhau.

66

MÔ HÌNH TCP/IP

Khác nhau:

- TCP/IP đơn giản hơn.
- OSI không có khái niệm truyền thiếu tin cậy ở tầng giao vận như giao thức UDP của mô hình TCP/IP.
- Ứng dụng khác nhau:
 - Internet được phát triển dựa trên các chuẩn của họ giao thức TCP/IP, do đó mô hình TCP/IP được tin tưởng tin nhiệm bởi các giao thức cụ thể của nó. (Hiện nay đã được chuẩn hóa và sử dụng phổ biến trên toàn thế giới).
 - Mô hình OSI không định ra một giao thức cụ thể nào và nó chỉ đóng vai trò như một khung tham chiếu (hướng dẫn) để hiểu và tạo ra một quá trình truyền thông.

67

HỆ ĐIỀU HÀNH MẠNG (NOS – Network Operating Systems)

- Chức năng: Quản lý dữ liệu và tính toán, xử lý một cách thống nhất trên phạm vi toàn mạng.
- Để cài đặt một hệ điều hành, có 2 cách tiếp cận khác nhau:
 1. Giữ nguyên các hệ điều hành đã có sẵn trên các máy tính của mạng => Hệ điều hành mạng được cài đặt như một tập các chương trình tiện ích chạy trên các máy khác nhau của mạng
=> Dễ cài đặt và không vô hiệu hóa các phần mềm đã có.

68

HỆ ĐIỀU HÀNH MẠNG (NOS – Network Operating Systems)

2. Cài một hệ điều hành thuần nhất trên toàn bộ mạng, gọi là *hệ điều hành phân tán (distributed operating system)* => Độ phức tạp của công việc lớn hơn nhiều.

=> Tùy từng trường hợp cụ thể của mạng để chọn giải pháp nào cho phù hợp.

69

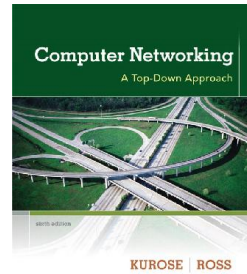
YÊU CẦU TỰ ĐỌC THÊM

- Các tổ chức thực hiện việc chuẩn hóa mạng máy tính (ISO, CCITT, ECMA, ANSI, IEEE, ...).
- Kiến trúc mạng riêng của một số công ty (kiến trúc mạng SNA của IBM, kiến trúc mạng DNA của DEC, ...).
- Cách tiếp cận của việc nối kết các mạng máy tính với nhau. Các giao diện kết nối: Gateway, bridge, router.

70

Chương 1

Giới thiệu



Người dịch: Nguyễn Thanh Thủy

Tài liệu được dịch cho mục đích giảng dạy (được sự đồng ý của tác giả).

© All material copyright 1996-2012
J.F Kurose and K.W. Ross, All Rights Reserved

*Computer
Networking: A Top
Down Approach*
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Giới thiệu 1-1

Chương 1: Giới thiệu

Mục đích:

- ❖ Hiểu được các thuật ngữ
- ❖ Hiểu sâu, chi tiết hơn trong các phần sau của khóa học
- ❖ Cách tiếp cận:
 - Dùng Internet làm ví dụ

Tổng quan:

- ❖ Internet là gì?
- ❖ Giao thức là gì?
- ❖ Phần cạnh của mạng; các hệ thống đầu cuối (hosts), truy cập mạng, đường truyền vật lý
- ❖ Phần lõi của mạng: chuyển mạch gói/chuyển mạch kênh (packet/circuit switching), cấu trúc mạng Internet.
- ❖ Hiệu năng mạng: mất mát, trễ, thông lượng.
- ❖ An ninh mạng
- ❖ Các tầng giao thức, các mô hình dịch vụ
- ❖ Lịch sử phát triển

Giới thiệu 1-2

Chương 1: Nội dung

1.1 Internet là gì?

1.2 Phần cạnh của mạng

- Hệ thống đầu cuối, mạng truy nhập, liên kết

1.3 Phần lõi của mạng

- Chuyển mạch gói, chuyển mạch kênh, cấu trúc mạng

1.4 Trễ, mất mát, thông lượng trong mạng

1.5 Các tầng giao thức, các mô hình dịch vụ

1.6 Các mạng bị tấn công: vấn đề an ninh mạng

1.7 Lịch sử phát triển mạng Internet

Giới thiệu 1-3

Internet là gì?



Các thiết bị Internet "thú vị"



Khung ảnh IP
<http://www.ceiva.com/>



Lò nướng bánh mì
được kích hoạt bằng Web
+ máy dự báo thời tiết



Tweet-a-watt: Giám sát
sử dụng năng lượng



Tủ lạnh Internet



Slingbox: đồng hồ,
TV cáp điều khiển từ xa



Điện thoại Internet

Giới thiệu 1-5

Internet là gì?

❖ **Internet: "mạng của các mạng"**

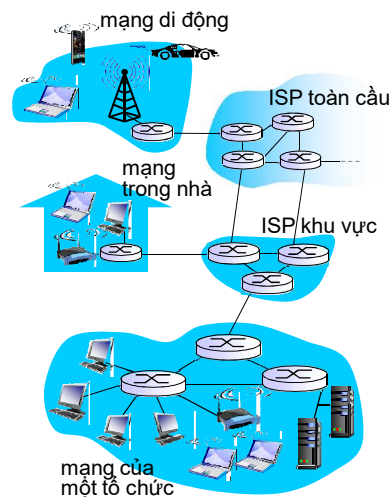
- Các ISP được kết nối với nhau

❖ **Giao thức** điều khiển việc gửi và nhận các thông điệp

- Ví dụ: TCP, IP, HTTP, Skype, 802.11

❖ **Các chuẩn Internet**

- RFC: Request for comments
- IETF: Internet Engineering Task Force



Giới thiệu 1-6

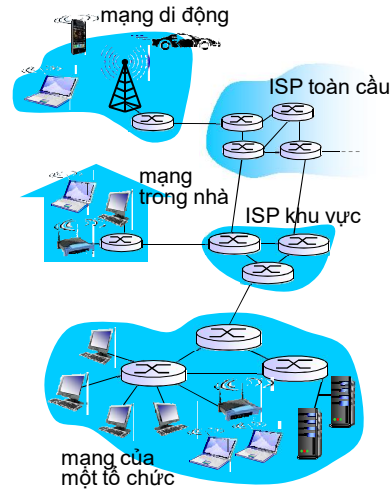
Internet là gì?

❖ Cơ sở hạ tầng cung cấp các dịch vụ cho các ứng dụng:

- Web, VoIP, thư điện tử, games, thương mại điện tử, mạng xã hội,...

❖ Cung cấp giao diện lập trình cho các ứng dụng

- Cho phép chương trình ứng dụng "kết nối" được với mạng Internet
- Cung cấp các tùy chọn dịch vụ



Giới thiệu 1-7

Giao thức là gì?

Giao thức của con người:

- ❖ "Máy giờ rồi?"
- ❖ "Tôi có một câu hỏi"
- ❖ Giới thiệu

... xác định các thông điệp được gửi

... xác định các hành động sẽ thực hiện khi nhận được các thông điệp, hoặc các sự kiện khác.

Giao thức mạng:

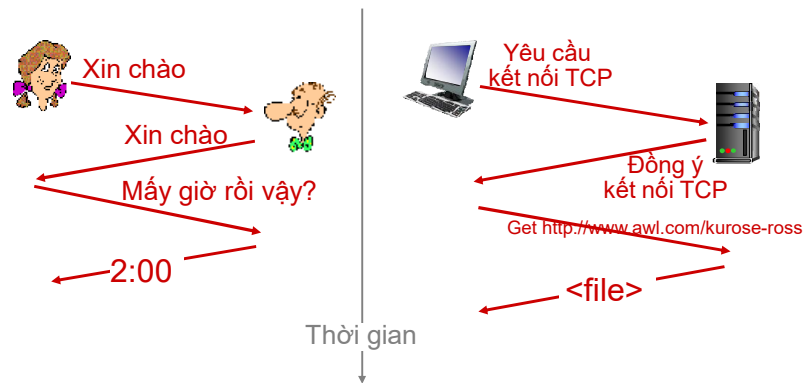
- ❖ Giữa các máy tính chứ không phải con người
- ❖ Tất cả các hoạt động truyền thông trong mạng Internet đều được quản lý bởi các giao thức.

Giao thức định nghĩa định dạng, thứ tự của các thông điệp gửi và nhận giữa các thực thể mạng, và các hành động được thực hiện trong quá trình truyền và nhận thông điệp.

Giới thiệu 1-8

Giao thức là gì?

So sánh giữa giao thức của con người và giao thức mạng máy tính:



Hỏi: các giao thức khác của con người?

Giới thiệu 1-9

Chương 1: Nội dung

1.1 Internet là gì?

1.2 Phần cạnh của mạng

- Hệ thống đầu cuối, mạng truy nhập, liên kết

1.3 Phần lõi của mạng

- Chuyển mạch gói, chuyển mạch kênh, cấu trúc mạng

1.4 Trễ, mất mát, thông lượng trong mạng

1.5 Các tầng giao thức, các mô hình dịch vụ

1.6 Các mạng bị tấn công: vấn đề an ninh mạng

1.7 Lịch sử phát triển mạng Internet

Giới thiệu 1-10

Cấu trúc của mạng

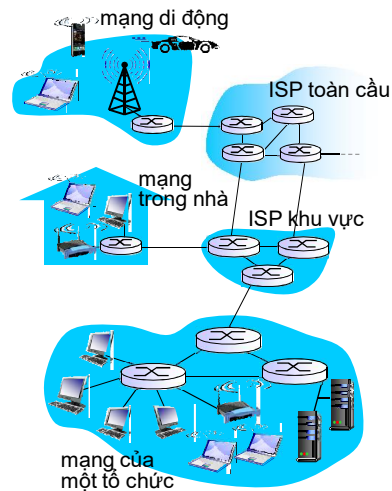
❖ *Phần cạnh của mạng:*

- hosts: clients (máy khách) và servers (máy chủ)
- servers thường có trong các trung tâm dữ liệu

❖ *Các mạng truy nhập, đường truyền vật lý:* các kết nối truyền thông có dây (hữu tuyến), không dây (vô tuyến)

❖ *Phần lõi của mạng:*

- Các bộ định tuyến được kết nối với nhau
- Mạng của các mạng



Giới thiệu 1-11

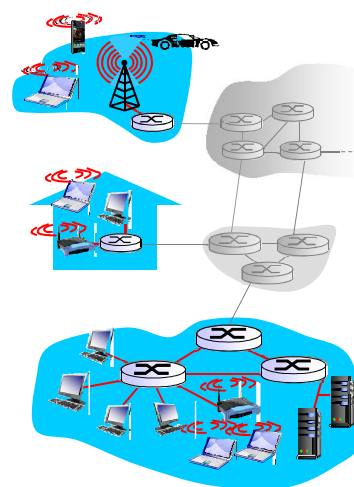
Mạng truy nhập và đường truyền vật lý

Hỏi: Làm thế nào để kết nối các hệ thống đầu cuối với bộ định tuyến cạnh?

- ❖ Các mạng truy nhập thuộc khu dân cư
- ❖ Các mạng truy nhập của các tổ chức (trường học, công ty)
- ❖ Các mạng truy nhập di động

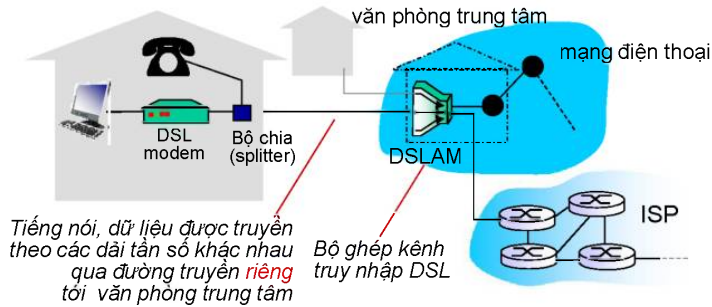
Lưu ý:

- ❖ Băng thông (bps – bits per second) của mạng truy nhập?
- ❖ Đường truyền dùng chung hay đường truyền riêng?



Giới thiệu 1-12

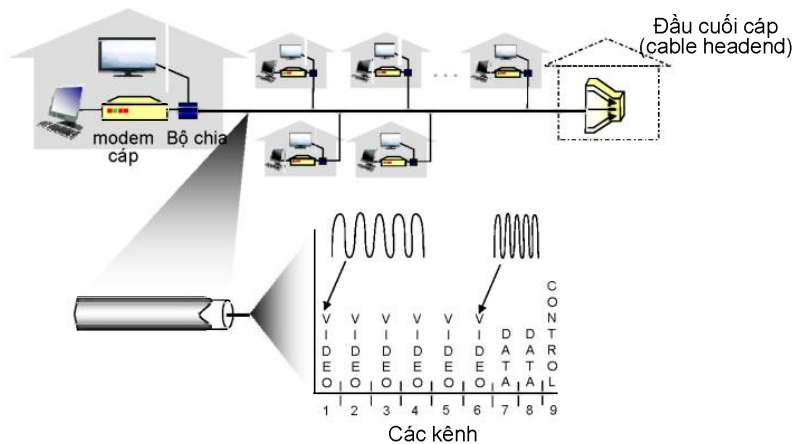
Mạng truy nhập: đường thuê bao số (digital subscriber line - DSL)



- ❖ Sử dụng đường điện thoại **có sẵn** để đi đến văn phòng trung tâm DSLAM
 - Dữ liệu qua đường điện thoại DSL đi ra Internet
 - Tiếng nói qua đường điện thoại DSL đi tới mạng điện thoại
- ❖ Tốc độ tải lên < 2.5 Mbps (thường < 1 Mbps)
- ❖ Tốc độ tải về < 24 Mbps (thường < 10 Mbps)

Giới thiệu 1-13

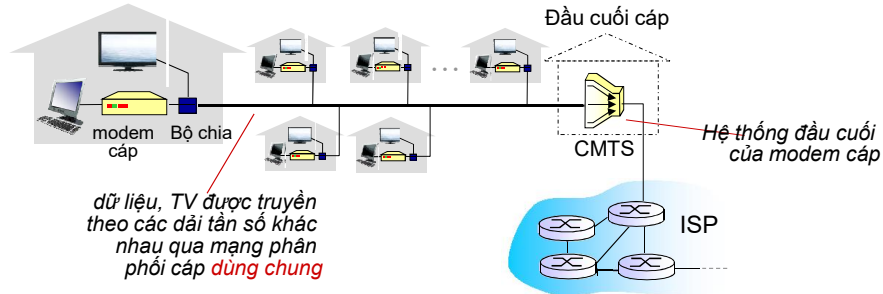
Mạng truy nhập: mạng cáp



Ghép kênh phân chia theo tần số (FDM - frequency division multiplexing):
các kênh khác nhau truyền theo các dải tần số khác nhau

Giới thiệu 1-14

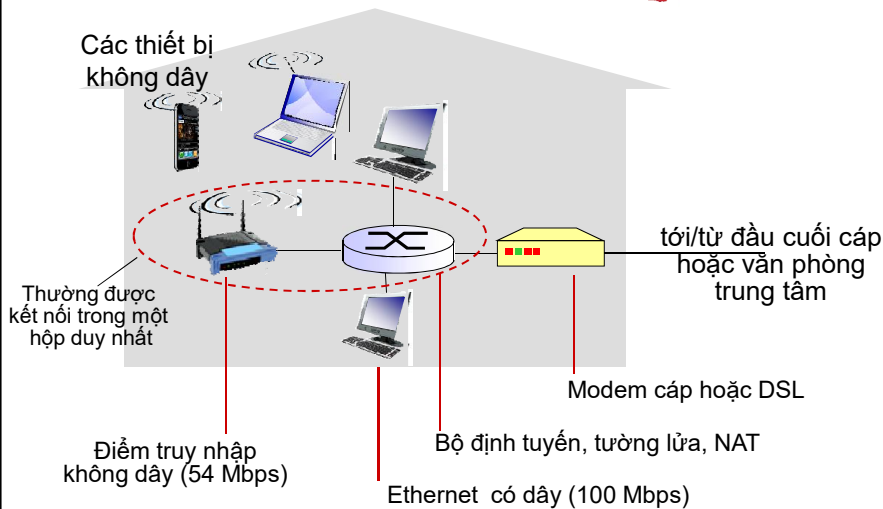
Mạng truy nhập: mạng cáp



- ❖ HFC: ghép lai cáp sợi quang-đồng trục (hybrid fiber coax)
 - Bất đối xứng: tốc độ tải xuống là 30Mbps, tốc độ tải lên là 2 Mbps
- ❖ Mạng cáp, cáp quang được nối từ nhà tới bộ định tuyến ISP
 - Các nhà *dùng chung mạng truy nhập* tới đầu cuối cáp
 - Không giống như DSL (dùng đường truy nhập riêng tới văn phòng trung tâm)

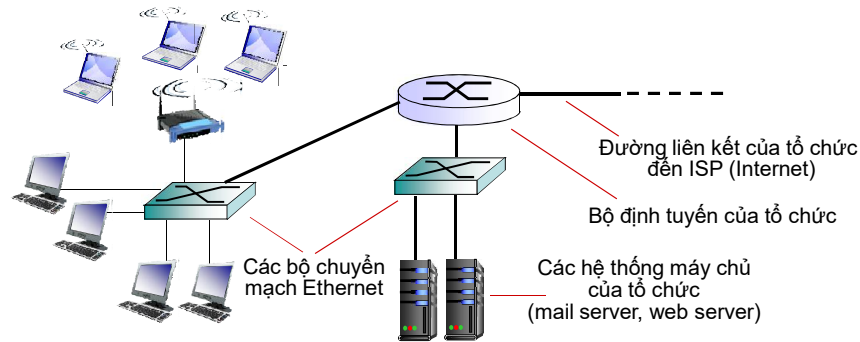
Giới thiệu 1-15

Mạng truy nhập: mạng ở nhà



Giới thiệu 1-16

Mạng truy nhập của tổ chức (Ethernet)



- ❖ Thường dùng trong các công ty, trường học,...
- ❖ Tốc độ truyền 10 Mbps, 100Mbps, 1Gbps, 10Gbps
- ❖ Ngày nay, các hệ thống đầu cuối thường được kết nối vào bộ chuyển mạch (switch) Ethernet

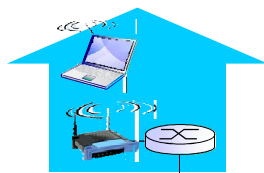
Giới thiệu 1-17

Các mạng truy nhập không dây

- ❖ Mạng truy nhập không dây chia sẻ (dùng chung) kết nối hệ thống đầu cuối tới bộ định tuyến
 - Qua trạm cơ sở: điểm truy nhập (access point)

Mạng LAN không dây (wireless LANs):

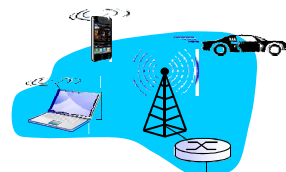
- Dùng bên trong tòa nhà
- 802.11b/g (WiFi): tốc độ truyền 11 Mbps, 54 Mbps



tới Internet

Mạng truy nhập không dây diện rộng

- Được cung cấp bởi các nhà điều hành viễn thông (di động) (10 km)
- Tốc độ truyền từ 1 đến 10 Mbps
- 3G, 4G: LTE



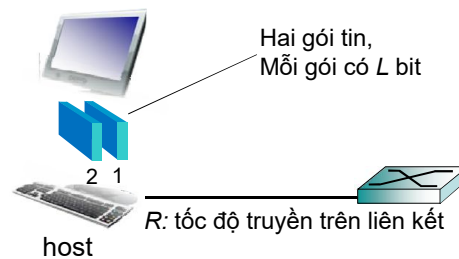
tới Internet

Giới thiệu 1-18

Host: gửi các gói dữ liệu

Chức năng gửi của host:

- ❖ Truyền các thông điệp ứng dụng
- ❖ Chia dữ liệu thành các đoạn nhỏ hơn, được gọi là gói tin (*packet*), có độ dài L bit.
- ❖ Truyền gói tin trên mạng truy nhập với **tốc độ R**
 - Tốc độ truyền trên liên kết, còn được gọi là **khả năng của liên kết**, hay **băng thông của liên kết**.



$$\text{Trễ truyền gói tin} = \text{Thời gian cần để truyền gói tin } L\text{-bit trên liên kết} = \frac{L \text{ (bit)}}{R \text{ (bit/sec)}}$$

1-19

Đường truyền vật lý

- ❖ **bit**: lan truyền giữa cặp thiết bị truyền/thiết bị nhận (máy phát/máy thu)
- ❖ **Liên kết vật lý**: là phần nằm giữa thiết bị truyền và thiết bị nhận
- ❖ **Đường truyền có dây**:
 - Tín hiệu lan truyền trong môi trường rắn: dây đồng, cáp quang, cáp đồng trục
- ❖ **Đường truyền không dây**:
 - Tín hiệu lan truyền tự do, ví dụ sóng radio

Cáp xoắn đôi (TP)

- ❖ Hai dây đồng cách điện
 - Loại 3: 10 Mbps Ethernet
 - Loại 5: 100 Mbps, 1 Gbps Ethernet
 - Loại 6: 10Gbps



Giới thiệu 1-20

Đường truyền vật lý: cáp đồng trục, cáp quang

Cáp đồng trục:

- ❖ Hai dây dẫn bằng đồng cùng tâm
- ❖ Tín hiệu truyền hai chiều
- ❖ Băng tần rộng:
 - nhiều kênh trên cáp
 - HFC



Cáp quang:

- ❖ Sợi thủy tinh mang dao động ánh sáng, mỗi dao động là 1 bit.
- ❖ Hoạt động tốc độ cao:
 - Truyền điểm-nối-điểm tốc độ cao (10-100 Gpbs)
- ❖ Tỷ lệ lỗi thấp
 - Truyền được những khoảng cách rất xa
 - Không bị ảnh hưởng bởi nhiễu điện từ



Giới thiệu 1-21

Đường truyền vật lý: sóng radio

- ❖ Tín hiệu được mang dưới dạng sóng điện từ
- ❖ Không có dây dẫn vật lý
- ❖ Truyền tín hiệu hai chiều
- ❖ Những ảnh hưởng trong môi trường truyền:
 - Bị phản xạ
 - Bị các chướng ngại vật cản trở
 - Bị nhiễu

Các loại liên kết radio:

- ❖ **Vi sóng mặt đất**
 - Các kênh truyền với tốc độ lên tới 45 Mbps
- ❖ **LAN** (ví dụ WiFi)
 - 11Mbps, 54 Mbps
- ❖ **Mạng diện rộng** (ví dụ cellular)
 - 3G cellular: ~ vài Mbps
- ❖ **Sóng vệ tinh**
 - Kênh từ Kbps tới 45Mbps (hoặc chia nhiều kênh nhỏ hơn)
 - Độ trễ 270 msec giữa hai đầu cuối
 - Giữ khoảng cách cố định so với mặt đất (độ cao, thấp)

Giới thiệu 1-22

Chương 1: Nội dung

1.1 Internet là gì?

1.2 Phần cạnh của mạng

- Hệ thống đầu cuối, mạng truy nhập, liên kết

1.3 Phần lõi của mạng

- Chuyển mạch gói, chuyển mạch kênh, cấu trúc mạng

1.4 Trễ, mất mát, thông lượng trong mạng

1.5 Các tầng giao thức, các mô hình dịch vụ

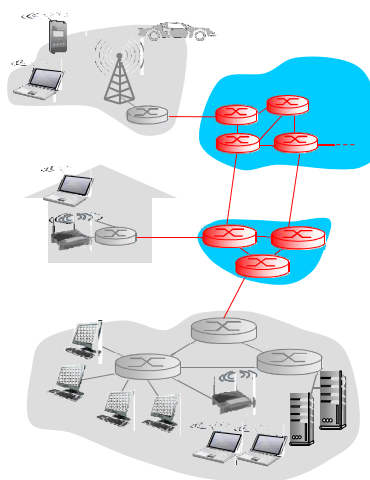
1.6 Các mạng bị tấn công: vấn đề an ninh mạng

1.7 Lịch sử phát triển mạng Internet

Giới thiệu 1-23

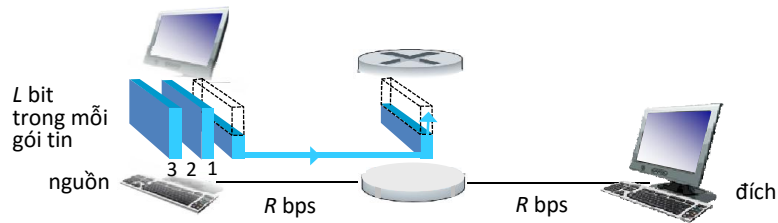
Phần lõi của mạng

- ❖ Lưới các bộ định tuyến được kết nối với nhau.
- ❖ Chuyển mạch gói: host chia các thông điệp ứng dụng thành các gói tin (*packet*)
 - Chuyển tiếp các gói tin từ một bộ định tuyến đến bộ định tuyến tiếp theo, qua các liên kết trên đường đi từ nguồn đến đích.
 - Mỗi gói tin được truyền đi với toàn bộ khả năng của liên kết.



Giới thiệu 1-24

Chuyển mạch gói: lưu và chuyển tiếp



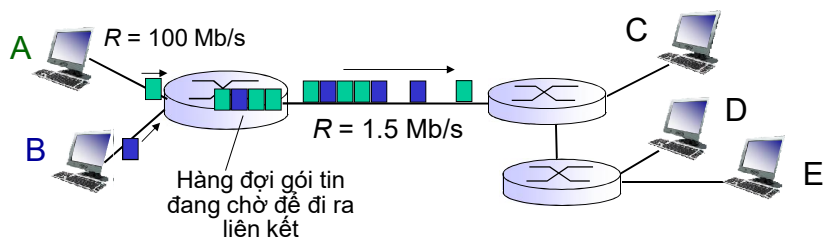
- ❖ Cần L/R giây để truyền (đẩy qua) gói có L -bit trên liên kết có tốc độ R bps
- ❖ **Lưu và chuyển tiếp:** toàn bộ gói phải đến bộ định tuyến trước khi nó có thể được truyền sang liên kết kế tiếp.
- ❖ Trễ đầu cuối-đầu cuối = $2L/R$ (giả sử trễ truyền bằng 0)

Ví dụ trên một hop:

- $L = 7.5$ Mb
- $R = 1.5$ Mbps
- Trễ truyền trên một hop = 5 giây

Giới thiệu 1-25

Chuyển mạch gói: trễ hàng đợi, mất mát



Hàng đợi và mất mát:

- ❖ Nếu tốc độ đi đến (tính theo bit) liên kết vượt quá tốc độ truyền của liên kết trong một khoảng thời gian, thì:
 - Các gói tin sẽ phải xếp hàng, chờ đợi để được truyền trên liên kết.
 - Các gói tin có thể bị mất nếu bộ nhớ (đệm) bị đầy.

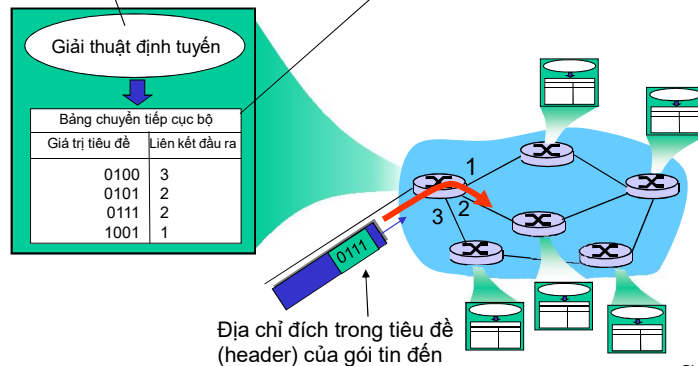
Giới thiệu 1-26

Hai chức năng chính trong phần lõi của mạng

Định tuyến: xác định tuyến đường đi để chuyển gói tin từ nguồn đến đích.

- Các giải thuật tìm đường

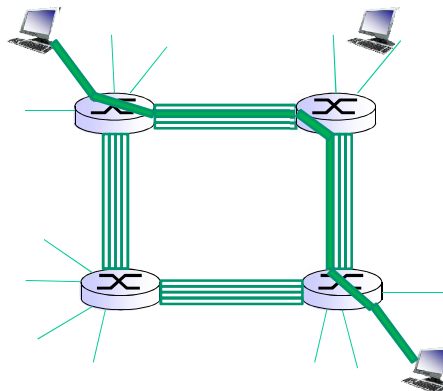
Chuyển tiếp: chuyển gói tin từ đầu vào tới đầu ra phù hợp của bộ định tuyến



Chuyển mạch kênh

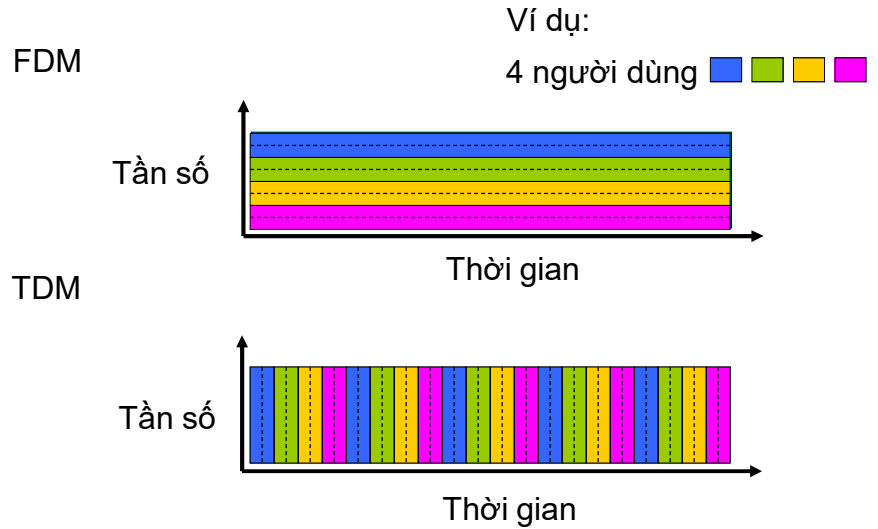
Tài nguyên giữa hai đầu cuối được xác định và dành riêng cho “cuộc gọi” giữa nguồn và đích:

- ❖ Trong sơ đồ, mỗi liên kết có 4 kênh.
 - Cuộc gọi dùng kênh số 2 trong liên kết phía trên và kênh số 1 trong liên kết bên phải.
- ❖ Tài nguyên dành riêng: không chia sẻ.
 - Hiệu năng được đảm bảo
- ❖ Đoạn kênh rỗi nếu không được sử dụng bởi cuộc gọi (*không chia sẻ*)
- ❖ Thường được dùng trong mạng điện thoại truyền thống



Giới thiệu 1-28

Chuyển mạch kênh: FDM và TDM



Giới thiệu 1-29

So sánh chuyển mạch kênh và chuyển mạch gói

Chuyển mạch gói cho phép nhiều người dùng sử dụng mạng hơn!

Ví dụ:

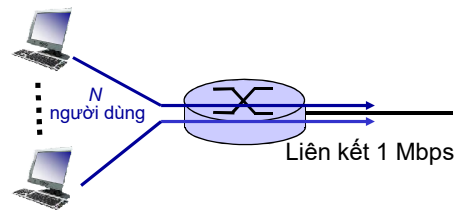
- Liên kết 1 Mb/s
- Mỗi người dùng:
 - 100 kb/s khi "kích hoạt"
 - Chiếm 10% thời gian

❖ **Chuyển mạch kênh:**

- 10 người dùng

❖ **Chuyển mạch gói:**

- Với 35 người dùng, xác suất > 10 người kích hoạt (dùng) tại cùng một thời điểm là nhỏ hơn .0004



Giới thiệu 1-30

So sánh chuyển mạch kênh và chuyển mạch gói

Ưu điểm của chuyển mạch gói:

- ❖ Rất tốt trong trường hợp bùng nổ dữ liệu
 - Chia sẻ tài nguyên
 - Đơn giản hơn, không cần thiết lập cuộc gọi
- ❖ Trong trường hợp tắc nghẽn quá mức: các gói tin bị trễ hoặc bị mất
 - Cần có các giao thức cho việc truyền dữ liệu tin cậy, điều khiển tắc nghẽn

Ưu điểm của chuyển mạch kênh

- ❖ Đảm bảo băng thông yêu cầu cho các ứng dụng audio/video

Giới thiệu 1-31

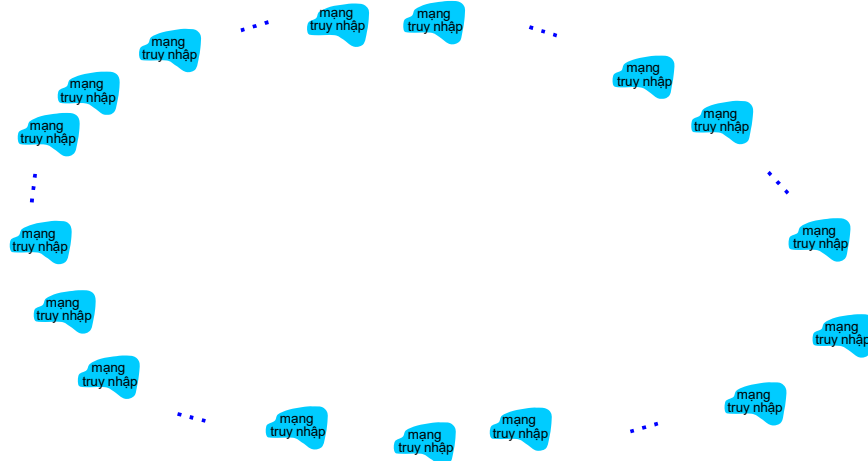
Cấu trúc của Internet: mạng của các mạng

- ❖ Các hệ thống đầu cuối kết nối tới Internet qua **mạng truy nhập của các nhà cung cấp dịch vụ Internet (ISP - Internet Service Providers)**
 - Mạng truy nhập của khu dân cư, công ty và trường học
- ❖ Các ISP lần lượt được kết nối với nhau
 - Để cho bất kỳ 2 host nào cũng có thể gửi các gói tin đến nhau
- ❖ Kết quả là có được hệ thống mạng của các mạng rất phức tạp
 - Sự phát triển được thúc đẩy bởi **kinh tế** và **chính sách quốc gia**
- ❖ Phần sau, theo cách tiếp cận từng bước sẽ mô tả cấu trúc của Internet hiện tại.

Giới thiệu 1-32

Cấu trúc của Internet: mạng của các mạng

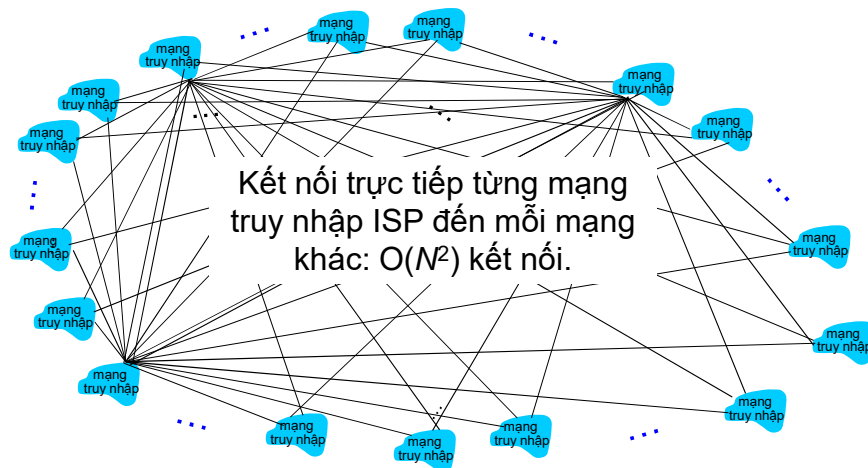
Hỏi: Có hàng triệu ISP truy nhập, làm thế nào có thể kết nối được chúng lại với nhau?



Giới thiệu 1-33

Cấu trúc của Internet: mạng của các mạng

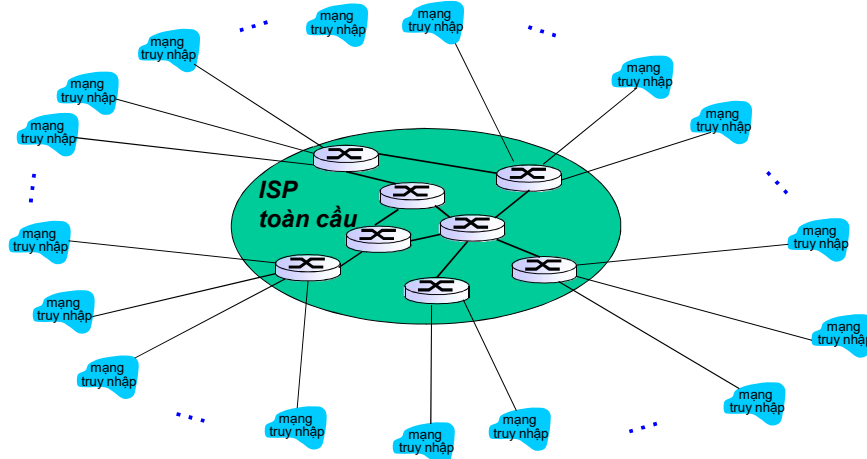
Lựa chọn: kết nối từng mạng truy nhập ISP đến tất cả các mạng truy nhập ISP khác?



Giới thiệu 1-34

Cấu trúc của Internet: mạng của các mạng

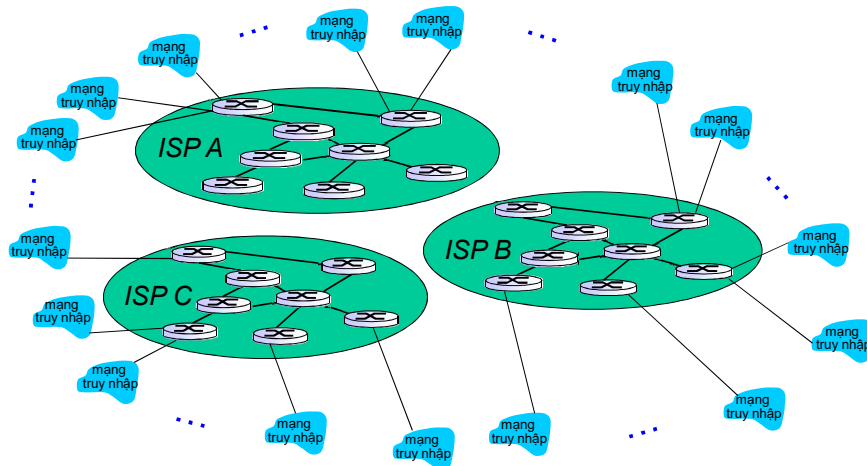
Lựa chọn: kết nối từng mạng truy nhập ISP tới một ISP chuyển tiếp toàn cầu? *Khách hàng và nhà cung cấp* ISP có thỏa thuận kinh tế.



Giới thiệu 1-35

Cấu trúc của Internet: mạng của các mạng

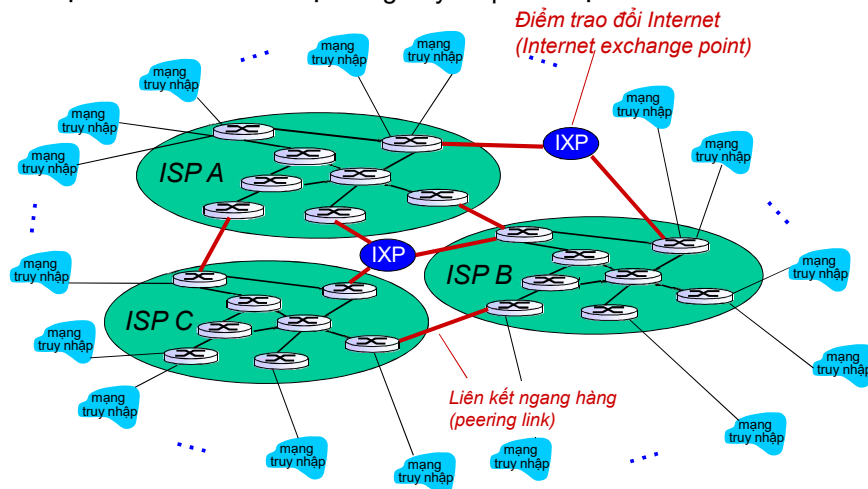
Nhưng nếu chỉ có một ISP toàn cầu kinh doanh khả thi, thì sẽ có các đối thủ cạnh tranh...



Giới thiệu 1-36

Cấu trúc của Internet: mạng của các mạng

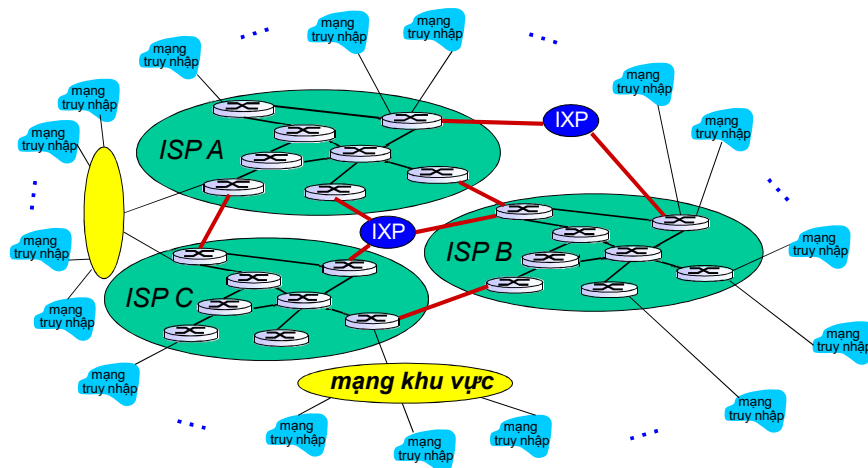
Nhưng nếu chỉ có một ISP toàn cầu kinh doanh khả thi, thì sẽ có các đối thủ cạnh tranh... và các hệ thống này sẽ phải được kết nối với nhau.



Giới thiệu 1-37

Cấu trúc của Internet: mạng của các mạng

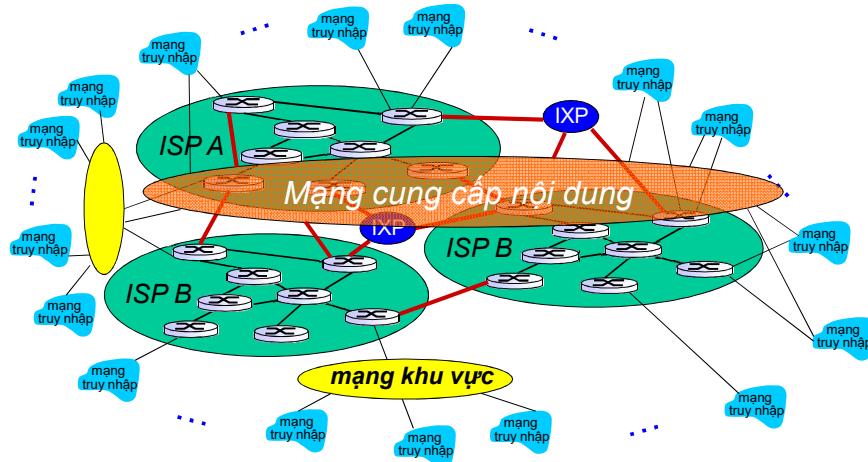
... và các mạng khu vực có thể phát sinh để kết nối các mạng truy nhập tới các ISP.



Giới thiệu 1-38

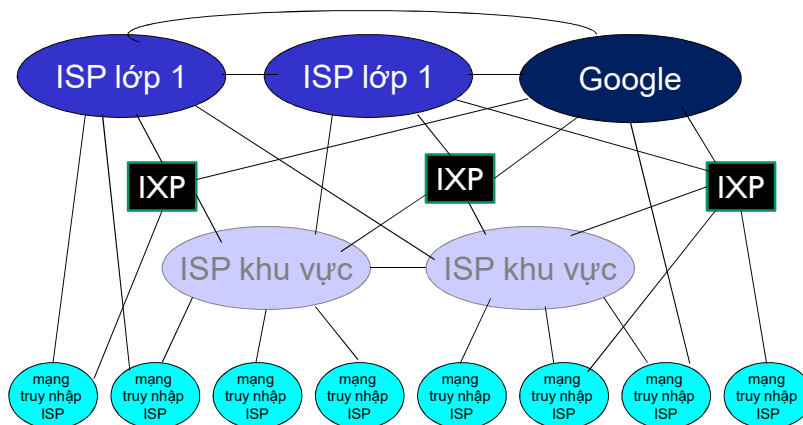
Cấu trúc của Internet: mạng của các mạng

... và các mạng cung cấp nội dung (Content provider networks) (ví dụ, Google, Microsoft, Akamai) có thể chạy mạng riêng của họ, để đưa các dịch vụ, nội dung đến gần với người dùng cuối.



Giới thiệu 1-39

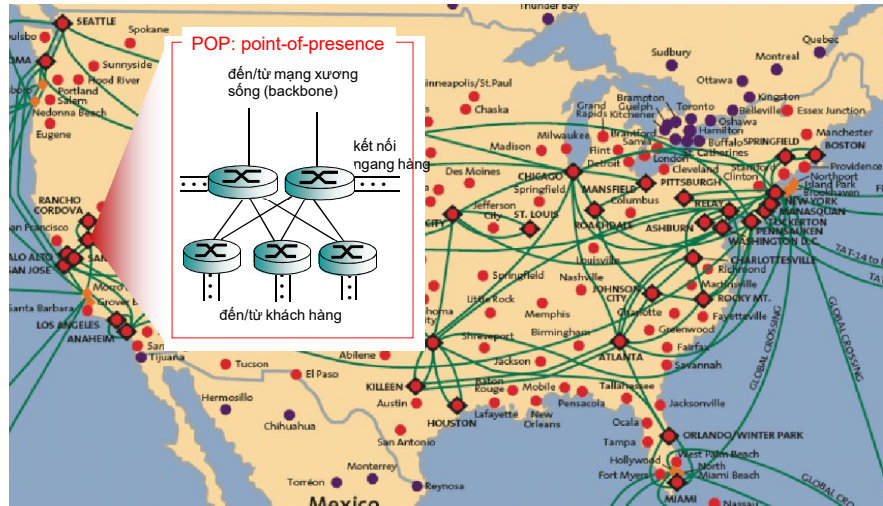
Cấu trúc của Internet: mạng của các mạng



- ❖ Tại trung tâm: số lượng nhỏ các mạng lớn được kết nối với nhau.
 - "Lớp-1" các ISP thương mại (ví dụ, Level 3, Sprint, AT&T, NTT), bao trùm các quốc gia và toàn thế giới
 - Mạng cung cấp nội dung (ví dụ, Google): mạng riêng kết nối nó với trung tâm dữ liệu Internet, thường bỏ qua lớp-1 và các ISP khu vực

Giới thiệu 1-40

ISP lớp-1: ví dụ, Sprint



Giới thiệu 1-41

Chương 1: Nội dung

1.1 Internet là gì?

1.2 Phần cạnh của mạng

- Hệ thống đầu cuối, mạng truy nhập, liên kết

1.3 Phần lõi của mạng

- Chuyển mạch gói, chuyển mạch kênh, cấu trúc mạng

1.4 Trễ, mất mát, thông lượng trong mạng

1.5 Các tầng giao thức, các mô hình dịch vụ

1.6 Các mạng bị tấn công: vấn đề an ninh mạng

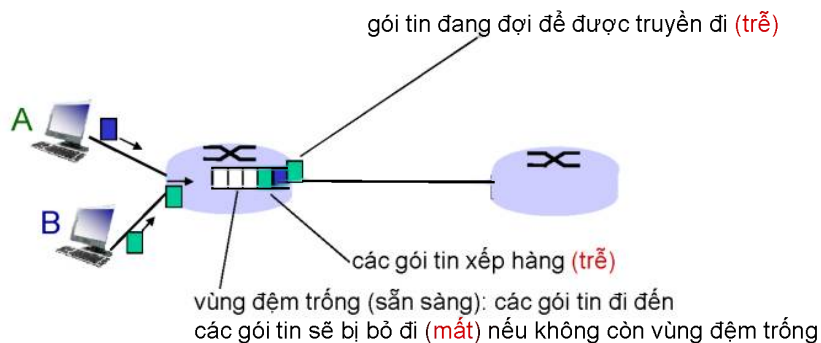
1.7 Lịch sử phát triển mạng Internet

Giới thiệu 1-42

Trễ và mất mát xảy ra như thế nào?

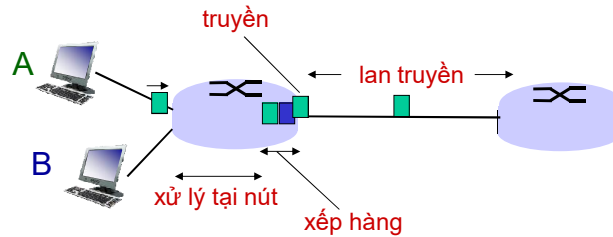
Các gói tin xếp hàng trong bộ đệm của bộ định tuyến

- ❖ Tỷ lệ các gói tin đi đến liên kết (tạm thời) vượt quá khả năng truyền đi của liên kết
- ❖ Các gói tin xếp hàng, chờ đến lượt được truyền đi



Giới thiệu 1-43

Bốn nguyên nhân của trễ



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

d_{proc} : trễ xử lý tại nút

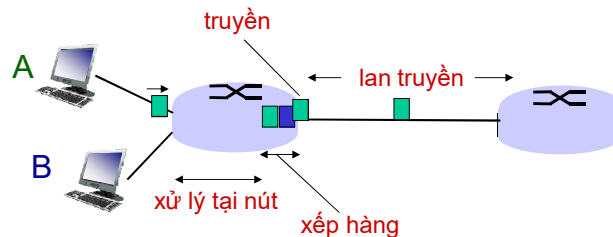
- Kiểm tra lỗi bit
- Xác định liên kết ra
- Thường < msec

d_{queue} : trễ xếp hàng

- Thời gian chờ tại đầu ra của liên kết để truyền đi
- Phụ thuộc vào mức độ tắc nghẽn của bộ định tuyến

Giới thiệu 1-44

Bốn nguyên nhân của trễ



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

d_{trans} : trễ truyền

- L : chiều dài của gói tin (bit)
- R : băng thông của liên kết (bps)
- $d_{\text{trans}} = L/R$

d_{prop} : trễ lan truyền

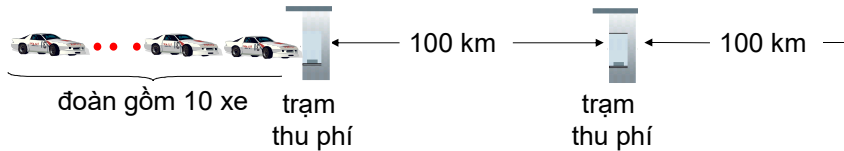
- d : chiều dài của liên kết vật lý
- s : tốc độ lan truyền trong môi trường ($\sim 2 \times 10^8$ m/s)

$$d_{\text{prop}} = d/s$$

d_{trans} và d_{prop}
rất khác nhau

Giới thiệu 1-45

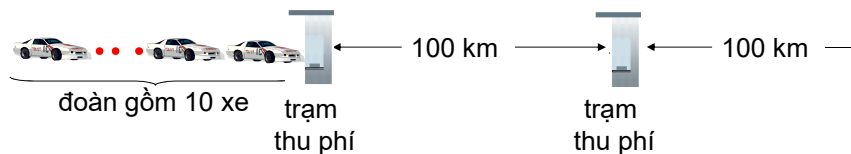
So sánh với việc di chuyển của đoàn xe



- ❖ Các xe “di chuyển” với tốc độ 100 km/giờ
- ❖ Thời gian đóng lệ phí tại trạm thu phí là 12 giây (thời gian di chuyển từng xe)
- ❖ Mỗi xe ~ 1 bit; đoàn xe ~ gói tin
- ❖ **Hỏi:** Mất bao lâu để cả đoàn xe đi đến được trạm thu phí thứ hai?
- Thời gian để “đẩy” cả đoàn xe qua trạm thu phí trên đường cao tốc = $12 \times 10 = 120$ giây
- Thời gian để mỗi xe di chuyển từ trạm thu phí thứ nhất đến trạm thứ hai: $100 \text{ km} / (100 \text{ km/giờ}) = 1$ giờ
- **Trả lời:** 62 phút

Giới thiệu 1-46

So sánh với việc di chuyển của đoàn xe

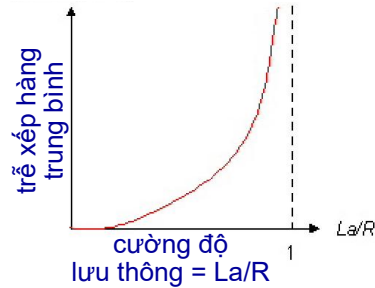


- ❖ Giả sử bây giờ các xe “di chuyển” với tốc độ 1000km/giờ
- ❖ Và thời gian đóng lệ phí tại trạm thu phí là 1 phút 1 xe
- ❖ **Hỏi:** Liệu có xe nào đến được trạm thu phí thứ hai trước khi cả đoàn xe hoàn thành việc đóng lệ phí tại trạm thứ nhất?
 - **Trả lời: Có!** Sau 7 phút, xe thứ nhất đến được trạm thu phí thứ hai; trong khi vẫn còn 3 xe đang dừng ở trạm thu phí thứ nhất.

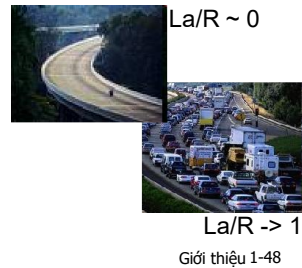
Giới thiệu 1-47

Trễ xếp hàng

- ❖ R : băng thông của liên kết (bps)
- ❖ L : chiều dài gói tin (bit)
- ❖ a : tốc độ đến của gói tin trung bình

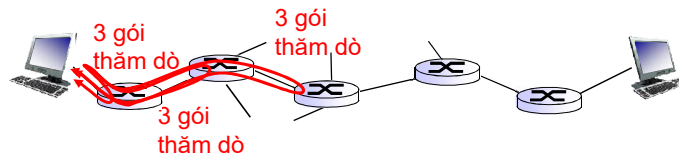


- ❖ $La/R \sim 0$: trễ xếp hàng trung bình nhỏ
- ❖ $La/R \rightarrow 1$: trễ xếp hàng trung bình lớn
- ❖ $La/R > 1$: lưu lượng đến vượt quá khả năng phục vụ, trễ trung bình có giá trị vô cùng!



Trễ và định tuyến “thực tế” trên mạng Internet

- ❖ Trễ và mất mát trên mạng Internet thực tế như thế nào?
- ❖ Chương trình **traceroute**: giúp đo độ trễ từ nguồn đến các bộ định tuyến dọc theo đường đi đến đích trên mạng Internet. Với tất cả i :
 - Gửi 3 gói tin đi đến bộ định tuyến i trên đường hướng tới đích
 - Bộ định tuyến i sẽ trả các gói tin về phía máy gửi



Giới thiệu 1-49

Trễ và định tuyến “thực tế” trên mạng Internet

traceroute: gaia.cs.umass.edu tới www.eurecom.fr

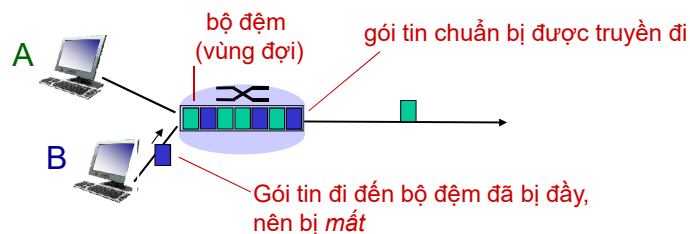
3 giá trị trễ từ
gaia.cs.umass.edu tới cs-gw.cs.umass.edu

1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms
6 abilene-vbns.abilene.ucaid.edu (198.32.11.9) 22 ms 18 ms 22 ms
7 nycm-wash.abilene.ucaid.edu (198.32.8.46) 22 ms 22 ms 22 ms
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms ← liên kết qua đại dương
9 de2-1.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms
10 de.fr1.fr.geant.net (62.40.96.50) 113 ms 121 ms 114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms
12 nio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms
13 nice.cssi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms
14 r3t2-nice.cssi.renater.fr (195.220.98.110) 126 ms 126 ms 124 ms
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54) 135 ms 128 ms 133 ms
16 194.214.2.11.25 (194.214.2.11.25) 126 ms 128 ms 126 ms
17 *** ← * không có phản hồi (mất gói thăm dò, bộ định tuyến không trả lời)
18 ***
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms

Giới thiệu 1-50

Mất mát gói tin

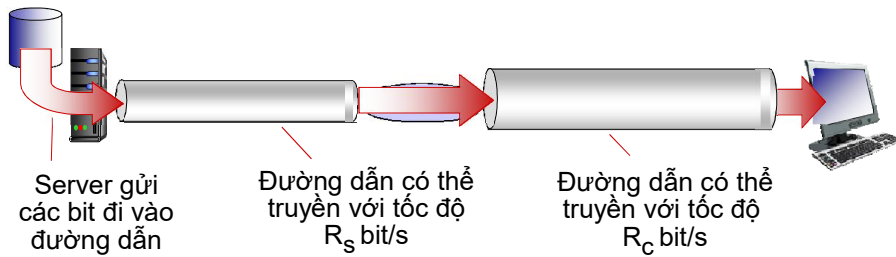
- ❖ Hàng đợi (bộ đệm) trước liên kết trong vùng nhớ đệm có dung lượng hữu hạn.
- ❖ Khi các gói tin đến hàng đợi đã bị đầy thì nó sẽ bị bỏ qua (nghĩa là bị làm mất)
- ❖ Gói tin bị mất có thể được truyền lại bởi nút mạng phía trước, hoặc hệ thống đầu cuối nguồn, hoặc không được truyền lại.



Giới thiệu 1-51

Thông lượng

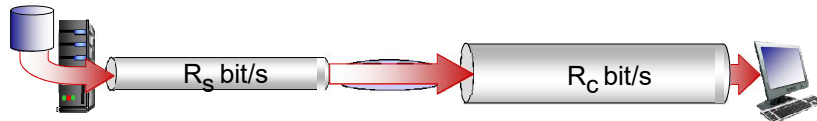
- ❖ **Thông lượng:** tốc độ (số bit/đơn vị thời gian) mà các bit được truyền đi giữa bên gửi/bên nhận
 - **Thông lượng tức thời:** tốc độ tại thời điểm đưa ra
 - **Thông lượng trung bình:** tốc độ đo trong một khoảng thời gian



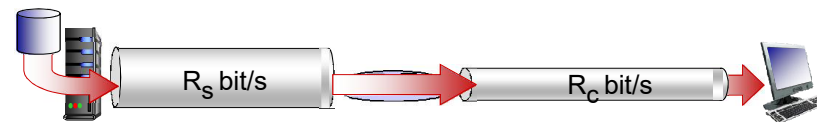
Giới thiệu 1-52

Thông lượng

- ❖ $R_s < R_c$ Thông lượng trung bình giữa hai đầu cuối sẽ như thế nào?



- ❖ $R_s > R_c$ Thông lượng trung bình giữa hai đầu cuối sẽ như thế nào?



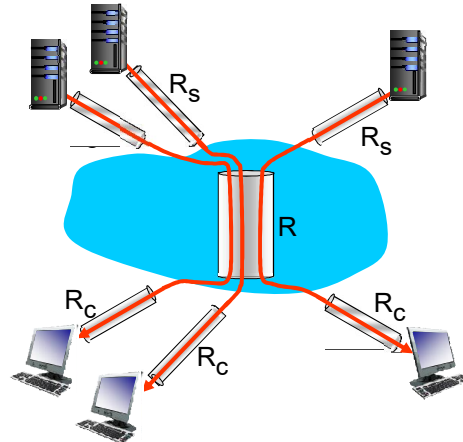
Liên kết nút cổ chai

Liên kết trên đường giữa hai đầu cuối mà làm giới hạn thông lượng giữa hai đầu cuối đó.

Giới thiệu 1-53

Thông lượng: Kịch bản trên mạng Internet

- ❖ Thông lượng giữa hai đầu cuối trên mỗi kết nối: $\min(R_c, R_s, R/10)$
- ❖ Trên thực tế: R_c hoặc R_s thường là nút cổ chai



10 kết nối chia sẻ liên kết nút cổ chai trên mạng xương sống có tốc độ R bit/s

Giới thiệu 1-54

Chương 1: Nội dung

- 1.1 Internet là gì?
- 1.2 Phần cạnh của mạng
 - Hệ thống đầu cuối, mạng truy nhập, liên kết
- 1.3 Phần lõi của mạng
 - Chuyển mạch gói, chuyển mạch kênh, cấu trúc mạng
- 1.4 Trễ, mất mát, thông lượng trong mạng
- 1.5 Các tầng giao thức, các mô hình dịch vụ
- 1.6 Các mạng bị tấn công: vấn đề an ninh mạng
- 1.7 Lịch sử phát triển mạng Internet

Giới thiệu 1-55

“Các tầng” giao thức

Các mạng rất phức tạp với nhiều “phần”:

- Các trạm (host)
- Các bộ định tuyến (router)
- Các liên kết với nhiều loại đường truyền khác nhau
- Các ứng dụng
- Các giao thức
- Phần cứng, phần mềm

Hỏi:

Liệu có cách nào để tổ chức cấu trúc của mạng không?

Giới thiệu 1-56

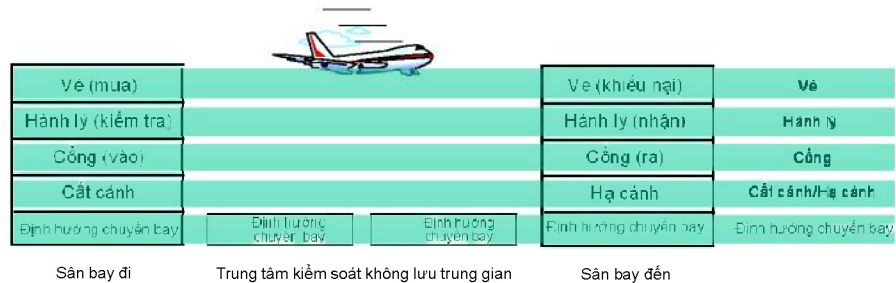
Tổ chức theo kiểu vận chuyển hàng không



❖ Một chuỗi các bước

Giới thiệu 1-57

Các tầng chức năng của vận chuyển hàng không



Các tầng: mỗi tầng thực hiện một dịch vụ

- thông qua các hoạt động của tầng bên trong nội bộ của nó
- dựa vào các dịch vụ được cung cấp bởi tầng dưới

Giới thiệu 1-58

Tại sao lại phân tầng

Nhằm xử lý các hệ thống phức tạp:

- ❖ Cấu trúc rõ ràng cho phép xác định các phần và mối quan hệ giữa chúng trong hệ thống phức tạp.
 - Thảo luận việc phân tầng trong *mô hình tham chiếu*
- ❖ Việc mô-đun hóa làm dễ dàng cho việc bảo trì, cập nhật hệ thống.
 - Việc thay đổi thực hiện dịch vụ của một tầng là trong suốt đối với phần còn lại của hệ thống
 - Ví dụ: thay đổi thủ tục kiểm tra tại cổng không ảnh hưởng tới phần còn lại của hệ thống
- ❖ Việc phân tầng có hại gì không?

Giới thiệu 1-59

Chồng giao thức của Internet

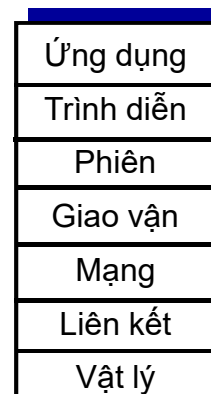
- ❖ **Tầng ứng dụng (application):** hỗ trợ các ứng dụng mạng
 - FTP, SMTP, HTTP
- ❖ **Tầng giao vận (transport):** truyền dữ liệu giữa các tiến trình
 - TCP, UDP
- ❖ **Tầng mạng (network):** định tuyến các gói tin đi từ nguồn đến đích
 - IP, các giao thức định tuyến
- ❖ **Tầng liên kết (link):** truyền dữ liệu giữa các phần tử mạng kề nhau (hàng xóm)
 - Ethernet, 802.11 (WiFi), PPP
- ❖ **Tầng vật lý (physical):** các bit "trên đường truyền"



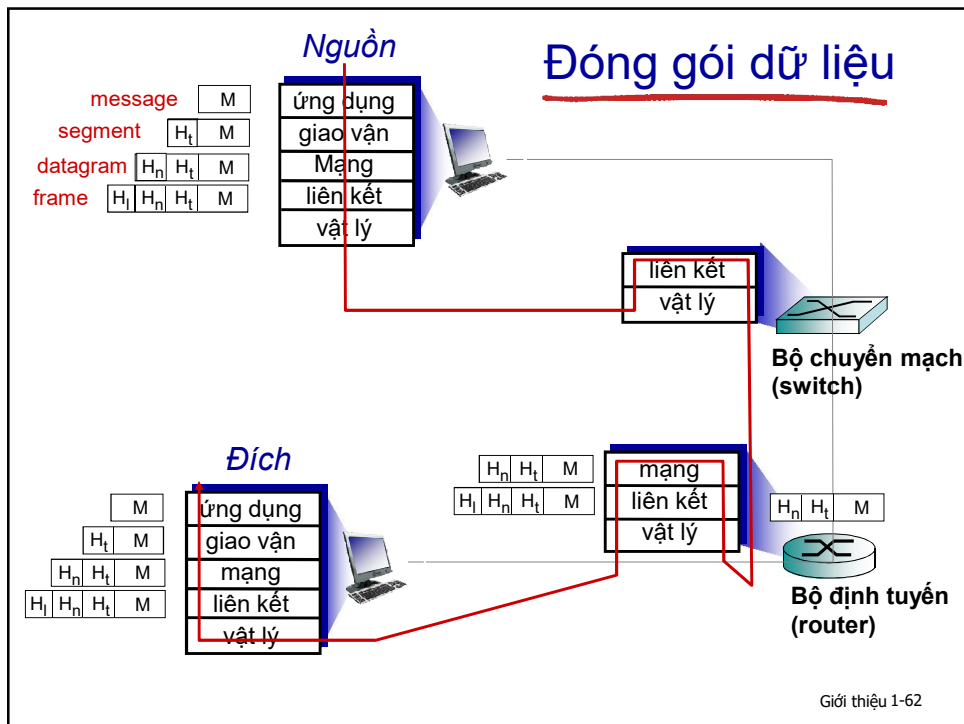
Giới thiệu 1-60

Mô hình tham chiếu ISO/OSI

- ❖ **Tầng trình diễn (presentation):** cho phép các ứng dụng diễn dịch ý nghĩa của dữ liệu, ví dụ: mã hóa, nén, định dạng của từng máy cụ thể
- ❖ **Tầng phiên (session):** đồng bộ hóa, quản lý phiên của ứng dụng
- ❖ Trong mạng Internet không có các tầng này!
 - Các dịch vụ này, nếu cần sẽ được cài đặt trong ứng dụng



Giới thiệu 1-61



Chương 1: Nội dung

1.1 Internet là gì?

1.2 Phần cạnh của mạng

- Hệ thống đầu cuối, mạng truy nhập, liên kết

1.3 Phần lõi của mạng

- Chuyển mạch gói, chuyển mạch kênh, cấu trúc mạng

1.4 Trễ, mất mát, thông lượng trong mạng

1.5 Các tầng giao thức, các mô hình dịch vụ

1.6 Các mạng bị tấn công: vấn đề an ninh mạng

1.7 Lịch sử phát triển mạng Internet

Giới thiệu 1-63

An ninh mạng

❖ Các lĩnh vực của an ninh mạng:

- Kẻ xấu có thể tấn công mạng máy tính như thế nào
- Chúng ta có thể bảo vệ mạng chống lại các tấn công như thế nào
- Có thể thiết kế kiến trúc mạng như thế nào để không bị tấn công

❖ Internet được thiết kế ban đầu không quan tâm nhiều đến vấn đề an ninh mạng

- *Cách nhìn ban đầu*: "một nhóm người dùng tin tưởng lẫn nhau được gắn với một hệ thống mạng trong suốt" ☺
- Các nhà thiết kế giao thức Internet chọn phương pháp "catch-up"
- Xem xét an ninh trong tất cả các tầng!

Giới thiệu 1-64

Kẻ xấu: đặt phần mềm độc hại vào các host qua mạng Internet

❖ Phần mềm độc hại (malware) có thể đi vào máy chủ từ:

- *Vi rút*: lây nhiễm theo cách tự sao qua đối tượng nhận/thực thi (ví dụ: tệp đính kèm trong thư điện tử)
- *Sâu mạng (worm)*: lây nhiễm theo cách tự sao qua đối tượng nhận thụ động mà có thể được tự thực thi

❖ Phần mềm độc hại gián điệp (spyware malware) có thể ghi lại thao tác bàn phím, các trang web truy cập, và tải thông tin lên cho trang thu thập.

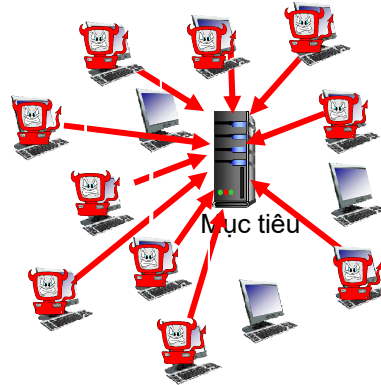
❖ Các host bị lây nhiễm có thể được ghi vào trong botnet, và được dùng để spam trong các cuộc tấn công DDoS.

Giới thiệu 1-65

Kẻ xấu: tấn công server, cơ sở hạ tầng mạng

Tấn công từ chối dịch vụ (Denial of Service - DoS): kẻ tấn công làm cho các nguồn tài nguyên (máy chủ, băng thông) không còn có sẵn để phục vụ cho các lưu lượng hợp pháp bằng cách sử dụng áp đảo tài nguyên với những lưu lượng không có thật.

1. Lựa chọn mục tiêu
2. Đột nhập vào host trên toàn mạng (xem botnet)
3. Gửi các gói tin tới mục tiêu từ các host đã bị xâm nhập

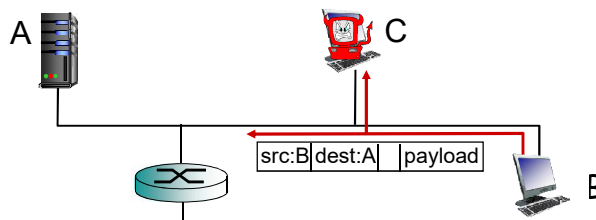


Giới thiệu 1-66

Kẻ xấu có thể bắt các gói tin

Bắt gói tin (packet "sniffing"):

- Đường truyền chung (quảng bá) (ethernet, wireless chia sẻ)
- Đọc/ghi lại tất cả các gói tin qua giao diện mạng ngẫu nhiên nào đó (ví dụ: bao gồm mật khẩu)

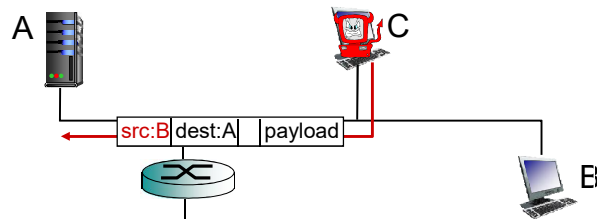


- ❖ Phần mềm wireshark dùng trong thực hành môn học có thể bắt gói tin (đây là phần mềm miễn phí).

Giới thiệu 1-67

Kẻ xấu có thể giả mạo địa chỉ

Giả mạo địa chỉ IP (IP spoofing): gửi gói tin với địa chỉ nguồn sai



... có rất nhiều vấn đề về an ninh mạng (xem thêm trong tài liệu)

Giới thiệu 1-68

Chương 1: Nội dung

1.1 Internet là gì?

1.2 Phần cạnh của mạng

- Hệ thống đầu cuối, mạng truy nhập, liên kết

1.3 Phần lõi của mạng

- Chuyển mạch gói, chuyển mạch kênh, cấu trúc mạng

1.4 Trễ, mất mát, thông lượng trong mạng

1.5 Các tầng giao thức, các mô hình dịch vụ

1.6 Các mạng bị tấn công: vấn đề an ninh mạng

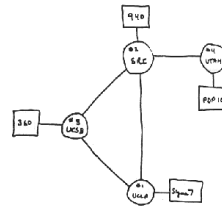
1.7 Lịch sử phát triển mạng Internet

Giới thiệu 1-69

Lịch sử phát triển Internet

1961-1972: Thời kỳ đầu của nguyên lý chuyển mạch gói

- ❖ 1961: Kleinrock – lý thuyết hàng đợi cho thấy tính hiệu quả của chuyển mạch gói
- ❖ 1964: Baran – chuyển mạch gói trong các mạng quân đội
- ❖ 1967: ARPANet được hình thành từ Advanced Research Projects Agency
- ❖ 1969: Nút ARPANet đầu tiên hoạt động
- ❖ 1972:
 - ARPANet được công bố
 - NCP (Network Control Protocol) là giao thức quản lý mạng đầu tiên
 - Chương trình đầu tiên là thư điện tử
 - ARPANet có 15 nút mạng



THE ARPANET NETWORK

Giới thiệu 1-70

Lịch sử phát triển Internet

1972-1980: Liên mạng, các mạng riêng và mới

- ❖ 1970: mạng vệ tinh ALOHAnet ở Hawaii
- ❖ 1974: Cerf and Kahn – kiến trúc cho hệ thống mạng toàn cầu
- ❖ 1976: Ethernet tại Xerox PARC
- ❖ Những năm 70: các mạng kiến trúc riêng: DECnet, SNA, XNA
- ❖ Cuối những năm 70: chuyển mạch cho các gói tin có độ dài cố định (tiền thân của ATM)
- ❖ 1979: ARPANet có 200 nút mạng

Nguyên lý mạng toàn cầu của Cerf and Kahn's :

- Yêu cầu tính tối thiểu, tự chủ - không thay đổi bên trong để kết nối các hệ thống mạng lại với nhau
- Mô hình dịch vụ tốt nhất
- Các bộ định tuyến phi trạng thái
- Điều khiển tập trung

Định nghĩa kiến trúc mạng Internet ngày nay!

Giới thiệu 1-71

Lịch sử phát triển Internet

1980-1990: giao thức mới, sự phát triển của các mạng

- ❖ 1983: triển khai TCP/IP
- ❖ 1982: định nghĩa giao thức SMTP cho e-mail
- ❖ 1983: DNS được định nghĩa cho chuyển đổi tên miền – IP
- ❖ 1985: định nghĩa giao thức FTP
- ❖ 1988: Giao thức điều khiển tắc nghẽn TCP
- ❖ Các mạng quốc gia mới: Cernet, BITnet, NSFnet, Minitel
- ❖ 100.000 host được kết nối vài liên minh các mạng

Giới thiệu 1-72

Lịch sử phát triển Internet

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

- ❖ Đầu những năm 1990: ARPANet ngừng hoạt động
- ❖ 1991: NSF chấm dứt những hạn chế trong thương mại do dùng NSFnet (ngừng hoạt động năm 1995)
- ❖ Những năm đầu 1990: Web
 - Siêu văn bản [Bush 1945, Nelson trong những năm 1960]
 - HTML, HTTP: Berners-Lee
 - 1994: Mosaic, sau đó Netscape
 - Cuối những năm 1990: thương mại hóa trên Web
- ❖ Cuối những năm 1990–2000:
 - ❖ Nhiều ứng dụng mới: tin nhắn nhanh, chia sẻ file P2P
 - ❖ An ninh mạng được đặt lên hàng đầu
 - ❖ Ước tính có khoảng 50 triệu host, hơn 100 triệu người dùng
 - ❖ Liên kết xương sống chạy với tốc độ Gbps

Giới thiệu 1-73

Lịch sử phát triển Internet

2005-hiện tại

- ❖ ~750 triệu host
 - Smartphones và tablets
- ❖ Triển khai mạnh truy nhập băng rộng
- ❖ Tăng độ phủ của truy nhập không dây tốc độ cao
- ❖ Sự xuất hiện của các mạng xã hội trực tuyến:
 - Facebook: sớm lên đến một tỷ người dùng
- ❖ Các nhà cung cấp dịch vụ (Google, Microsoft) tạo ra các mạng riêng của mình
 - Bypass Internet, cung cấp "tức thời" truy nhập để tìm kiếm, email,...
- ❖ Thương mại điện tử, các trường đại học, các doanh nghiệp chạy các dịch vụ của họ trong "đám mây" ("cloud")(ví dụ, Amazon EC2)

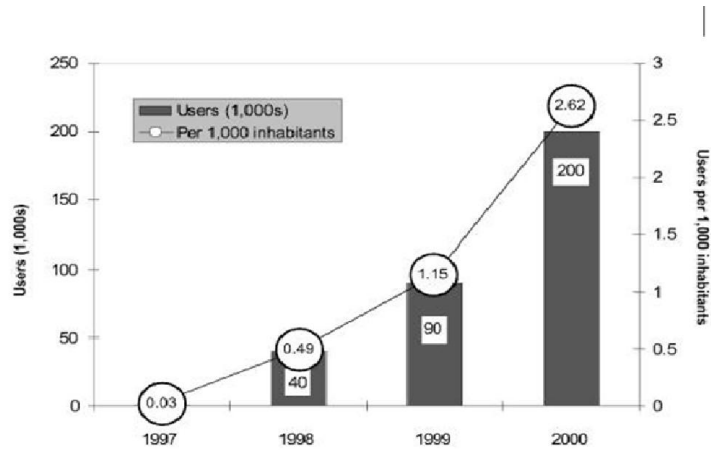
Giới thiệu 1-74

Lịch sử Internet Việt Nam

- ❖ 1991: Nỗ lực kết nối Internet không thành.
- ❖ 1996: Giải quyết các cản trở, chuẩn bị hạ tầng Internet
 - ISP: VNPT
 - Tốc độ 64kbps. Một đường kết nối quốc tế. Có một số người dùng.
- ❖ 1997: Việt Nam chính thức kết nối Internet.
 - 1 IXP: VNPT
 - 4 ISP: VNPT, Netnam (IOT), FPT, SPT
- ❖ 2007: "Mười năm Internet Việt Nam"
 - 20 ISP, 4 IXP
 - 19 triệu người dùng, chiếm 22,04% dân số

Giới thiệu 1-75

Lịch sử Internet Việt Nam

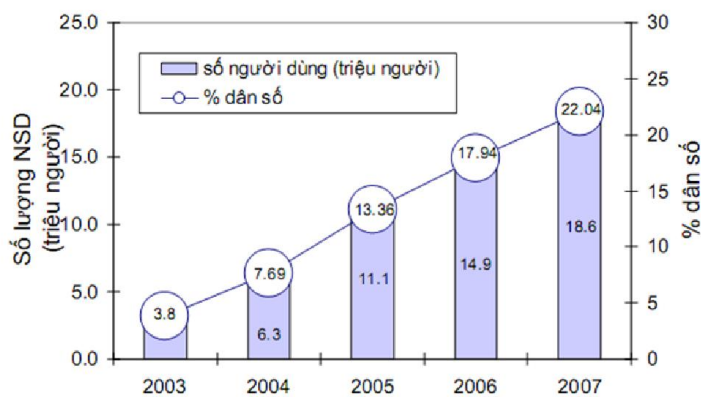


Ước tính số người dùng bằng hai lần số thuê bao

Source: Vietnam Internet Case Study, <http://www.itu.int/asean2001/reports/material/VNM%20CS.1>

Giới thiệu 1-76

Lịch sử Internet Việt Nam



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

Giới thiệu 1-77

Tổng kết

Cần nắm vững các nội dung:

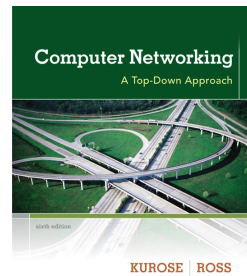
- ❖ Khái quát về Internet
- ❖ Giao thức là gì?
- ❖ Phần cạnh, phần lõi của mạng, mạng truy nhập
 - So sánh chuyển mạch gói và chuyển mạch kênh
 - Cấu trúc mạng Internet
- ❖ Hiệu năng: mất mát, trễ, thông lượng
- ❖ Phân tầng, các mô hình dịch vụ
- ❖ An ninh mạng
- ❖ Lịch sử phát triển mạng

Kiến thức thu được:

- ❖ Bối cảnh, khái quát, “cảm nhận” về mạng
- ❖ Để hiểu sâu hơn, chi tiết trong *các phần sau!*

Giới thiệu 1-78

Chương 2 Tầng ứng dụng



*Computer
Networking: A
Top Down
Approach*
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Người dịch: Nguyễn Thanh Thủy

Tài liệu được dịch cho mục đích giảng dạy (được sự đồng ý của tác giả).

All material copyright 1996-2012
© J.F Kurose and K.W. Ross, All Rights Reserved

Tầng ứng dụng 2-1

Chương 2: Tầng ứng dụng

Mục tiêu:

- ❖ Khái niệm, các vấn đề cài đặt giao thức ứng dụng mạng
 - Các mô hình dịch vụ tầng giao vận
 - Mô hình khách-chủ (client-server)
 - Mô hình điểm-điểm (peer-to-peer)
- ❖ Nghiên cứu một số giao thức tầng ứng dụng
 - HTTP
 - FTP
 - SMTP / POP3 / IMAP
 - DNS
- ❖ Tạo một ứng dụng mạng
 - socket API

Tầng ứng dụng 2-2

Chương 2: Nội dung

2.1 Nguyên lý của ứng dụng mạng

- Kiến trúc của ứng dụng
- Các yêu cầu của ứng dụng

2.2 Web và HTTP

2.3 FTP

2.4 Thư điện tử

- SMTP, POP3, IMAP

2.5 DNS

2.6 Ứng dụng P2P

2.7 Lập trình socket với UDP và TCP

Tầng ứng dụng 2-3

Một số ứng dụng mạng

- ❖ Thư điện tử (e-mail)
- ❖ web
- ❖ Tin nhắn văn bản (text messaging)
- ❖ Truy nhập từ xa (remote login)
- ❖ Chia sẻ file P2P
- ❖ Trò chơi nhiều người trên mạng
- ❖ streaming video (YouTube, Hulu, Netflix)
- ❖ Điện thoại Internet (voice over IP) (ví dụ Skype)
- ❖ Hội thảo video thời gian thực
- ❖ Mạng xã hội
- ❖ Các ứng dụng tìm kiếm
- ❖ ...
- ❖ ...

Tầng ứng dụng 2-4

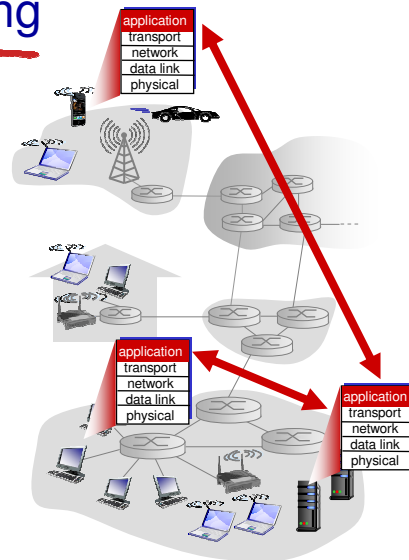
Tạo một ứng dụng mạng

Viết chương trình để:

- ❖ Chạy trên các *hệ thống đầu cuối* khác nhau
- ❖ Truyền thông qua mạng
- ❖ Ví dụ: phần mềm máy chủ web (web server) truyền thông với phần mềm trình duyệt (browser software)

Không cần viết chương trình ứng dụng cho các thiết bị trong phần lõi của mạng

- ❖ Các thiết bị trong phần lõi của mạng không chạy các ứng dụng người dùng.
- ❖ Các ứng dụng chạy trên thiết bị đầu cuối cho phép phát triển và phổ biến ứng dụng một cách nhanh chóng.



Tầng ứng dụng 2-5

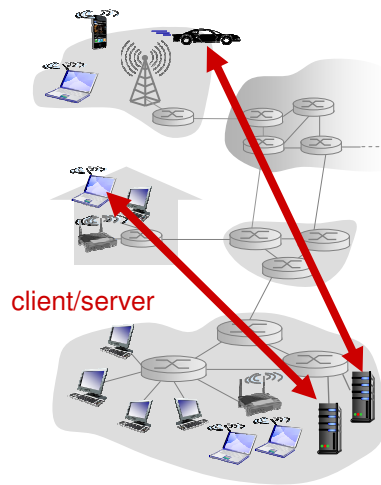
Kiến trúc của ứng dụng

Các ứng dụng có thể có kiến trúc dạng:

- ❖ Client-server (khách-chủ)
- ❖ Peer-to-peer (P2P, ngang hàng)

Tầng ứng dụng 2-6

Kiến trúc client-server



server:

- ❖ Là host luôn hoạt động
- ❖ Có địa chỉ IP cố định
- ❖ Các trung tâm dữ liệu

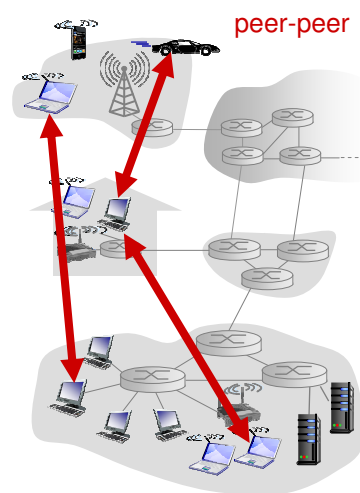
client:

- ❖ Truyền thông với server
- ❖ Có thể được kết nối liên tục vào mạng hoặc không
- ❖ Có thể có địa chỉ IP thay đổi
- ❖ Không truyền thông trực tiếp với client khác

Tầng ứng dụng 2-7

Kiến trúc P2P

- ❖ Không có server luôn hoạt động
- ❖ Các hệ thống đầu cuối (peer) truyền thông trực tiếp với nhau
- ❖ Mỗi peer yêu cầu dịch vụ từ một peer nào đó, và cung cấp dịch vụ lại cho các peer khác.
 - Có khả năng tự mở rộng – peer mới mang lại khả năng dịch vụ mới, cũng như có những yêu cầu dịch vụ mới.
- ❖ Các peer không kết nối liên tục và có thể thay đổi địa chỉ IP
 - Quản lý phức tạp



Tầng ứng dụng 2-8

Tiến trình truyền thông

Tiến trình: chương trình chạy trên một host

- ❖ Trên cùng một host, hai tiến trình truyền thông với nhau qua **truyền thông tiến trình nội bộ** (được xác định bởi hệ điều hành)
- ❖ Các tiến trình trên các host khác nhau truyền thông với nhau bằng cách trao đổi các **thông điệp**

client, server

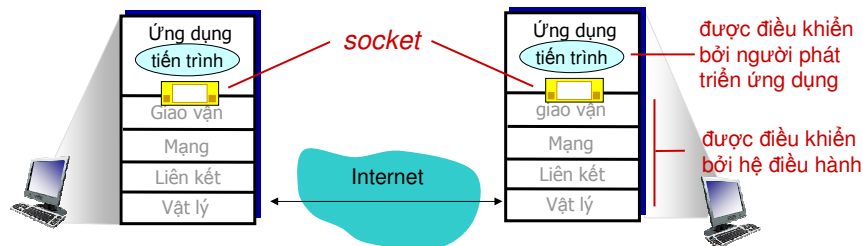
Tiến trình client: tiến trình khởi tạo truyền thông

Tiến trình server: tiến trình chờ được tiếp xúc

- ❖ Lưu ý: ứng dụng theo kiến trúc P2P có cả tiến trình client & tiến trình server

Socket

- ❖ Các tiến trình gửi/nhận các thông điệp đến/từ **socket** của nó
- ❖ Socket tương tự như một cửa vào/ra
 - Tiến trình gửi đẩy thông điệp ra bên ngoài cửa
 - Tiến trình gửi dựa trên cơ sở hạ tầng tầng giao vận ở phía bên kia cửa để truyền thông điệp đến socket của tiến trình nhận



Định địa chỉ cho tiến trình

- ❖ Để nhận các thông điệp, tiến trình phải có **định danh (identifier)**
- ❖ Thiết bị host phải có địa chỉ IP 32-bit duy nhất
- ❖ **Hỏi:** Địa chỉ IP của host mà trên đó tiến trình chạy có đủ để xác định định danh của tiến trình không?
 - **Trả lời:** Không, do nhiều tiến trình có thể cùng chạy trên một host.
- ❖ **Định danh** bao gồm cả **địa chỉ IP** và **số hiệu cổng** được kết hợp với tiến trình trên host.
- ❖ Ví dụ các số hiệu cổng:
 - HTTP server: 80
 - mail server: 25
- ❖ Để gửi thông điệp HTTP tới web server `gaia.cs.umass.edu`:
 - **Địa chỉ IP:** 128.119.245.12
 - **Số hiệu cổng:** 80

Tầng ứng dụng 2-11

Các giao thức tầng ứng dụng

- ❖ **Các loại thông điệp được trao đổi**
 - Ví dụ: thông điệp yêu cầu (request), thông điệp đáp ứng (response)
- ❖ **Cú pháp của thông điệp:**
 - Các trường trong thông điệp & mô tả
- ❖ **Ngữ nghĩa của thông điệp**
 - Ý nghĩa thông tin của các trường
- ❖ **Quy tắc** về thời gian và cách thức các tiến trình gửi và đáp ứng thông điệp
- Các giao thức mở (công khai):**
 - ❖ Được định nghĩa trong RFCs
 - ❖ Cho phép tương tác
 - ❖ Ví dụ: HTTP, SMTP
- Các giao thức riêng (độc quyền):**
 - ❖ Ví dụ: Skype

Tầng ứng dụng 2-12

Một ứng dụng cần những dịch vụ giao vận nào?

Toàn vẹn dữ liệu

- ❖ Một số ứng dụng (ví dụ: truyền file, web) yêu cầu truyền tin cậy 100%
- ❖ Các ứng dụng khác (ví dụ: audio) có thể chịu một số mất mát

Thời gian thực

- ❖ Một số ứng dụng (ví dụ: điện thoại Internet, trò chơi tương tác) yêu cầu độ trễ thấp mới “hiệu quả”

Băng thông

- ❖ Một số ứng dụng (ví dụ: đa phương tiện) yêu cầu băng thông tối thiểu để đảm bảo “hiệu quả”.
- ❖ Các ứng dụng khác (“các ứng dụng mềm dẻo”) có thể dùng băng thông nào cũng được

An toàn bảo mật

- ❖ Mã hóa, đảm bảo toàn vẹn dữ liệu, ...

Tầng ứng dụng 2-13

Một số yêu cầu dịch vụ giao vận với các ứng dụng phổ biến

Ứng dụng	Mất mát dữ liệu	Thông lượng	Thời gian thực
Truyền file	không	mềm dẻo	không
Thư điện tử	không	mềm dẻo	không
Web	không	mềm dẻo	không
Audio/video thời gian thực	chịu lỗi	audio: 5kbps-1Mbps video: 10kbps-5Mbps	có, 100 msec
Lưu trữ audio/video	chịu lỗi	như trên	có, vài giây
Trò chơi tương tác	chịu lỗi	một vài kbps	có, 100 msec
Tin nhắn nhanh	không	mềm dẻo	có và không

Tầng ứng dụng 2-14

Dịch vụ giao thức tầng giao vận của Internet

Dịch vụ TCP:

- ❖ **Truyền tin cậy** giữa tiến trình gửi và tiến trình nhận
- ❖ **Điều khiển luồng**: bên gửi không lấn át bên nhận
- ❖ **Điều khiển tắc nghẽn**: điều tiết bên gửi khi mạng bị quá tải
- ❖ **Không cung cấp**: thời gian thực, đảm bảo băng thông tối thiểu, an toàn bảo mật
- ❖ **Hướng kết nối**: yêu cầu thiết lập kết nối giữa tiến trình client và tiến trình server

Dịch vụ UDP:

- ❖ **Truyền dữ liệu không tin cậy** giữa tiến trình gửi và tiến trình nhận
- ❖ **Không cung cấp**: truyền tin cậy, điều khiển luồng, điều khiển tắc nghẽn, thời gian thực, đảm bảo băng thông, an toàn bảo mật, hoặc thiết lập kết nối

Hỏi: Tại sao lại dùng cả hai dịch vụ? Dùng UDP để làm gì?

Tầng ứng dụng 2-15

Các ứng dụng Internet: các giao thức tầng ứng dụng và giao vận

Ứng dụng	Giao thức tầng ứng dụng	Giao thức tầng giao vận
Thư điện tử	SMTP [RFC 2821]	TCP
Truy nhập từ xa	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Truyền file	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (ví dụ: YouTube), RTP [RFC 1889]	TCP hoặc UDP
Điện thoại Internet	SIP, RTP, giao thức độc quyền (ví dụ: Skype)	TCP hoặc UDP

Tầng ứng dụng 2-16

Bảo mật TCP

TCP & UDP

- ❖ Không mã hóa
- ❖ Mật khẩu dạng bản rõ được gửi vào socket để truyền trên Internet theo dạng bản rõ

SSL (Secure Sockets Layer)

- ❖ Hỗ trợ kết nối TCP được mã hóa
- ❖ Toàn vẹn dữ liệu
- ❖ Xác thực thiết bị đầu cuối

SSL là bảo mật tại tầng ứng dụng

- ❖ Ứng dụng dùng các thư viện của SSL để “nói” với TCP

Socket API của SSL

- ❖ Mật khẩu dạng bản rõ được gửi vào socket để truyền trên Internet theo dạng đã được mã hóa

Tầng ứng dụng 2-17

Chương 2: Nội dung

2.1 Nguyên lý của ứng dụng mạng

- Kiến trúc của ứng dụng
- Các yêu cầu của ứng dụng

2.2 Web và HTTP

2.3 FTP

2.4 Thư điện tử

- SMTP, POP3, IMAP

2.5 DNS

2.6 Ứng dụng P2P

2.7 Lập trình socket với UDP và TCP

Tầng ứng dụng 2-18

Web và HTTP

Một số thuật ngữ:

- ❖ *Trang web* bao gồm một số *đối tượng (object)*
- ❖ Đối tượng có thể là tệp HTML, ảnh JPEG, Java applet, tệp audio,...
- ❖ Trang web bao gồm *tệp HTML cơ bản* chứa *một số đối tượng được tham chiếu*
- ❖ Mỗi đối tượng được định địa chỉ bởi một *URL*, ví dụ:

`www.someschool.edu/someDept/pic.gif`

Tên host

Tên đường dẫn

Tầng ứng dụng 2-19

Khái quát về HTTP

HTTP: hypertext transfer protocol

- ❖ Giao thức tầng ứng dụng của Web
- ❖ Mô hình client/server
 - *client*: trình duyệt (browser) yêu cầu, nhận (sử dụng giao thức HTTP), và “hiển thị” các đối tượng Web
 - *server*: Máy chủ web (web server) gửi (sử dụng giao thức HTTP) các đối tượng đáp ứng cho yêu cầu.



Tầng ứng dụng 2-20

Khái quát về HTTP

Dùng TCP:

- ❖ Client khởi tạo kết nối TCP (tạo socket) tới server, cổng 80
- ❖ Server chấp nhận kết nối TCP từ client
- ❖ Thông điệp HTTP (thông điệp giao thức tầng ứng dụng) được trao đổi giữa trình duyệt (HTTP client) và máy chủ Web (HTTP server)
- ❖ Đóng kết nối TCP

HTTP là “không trạng thái”

- ❖ Server không lưu giữ thông tin về những yêu cầu trước đó của client

Vấn đề liên quan

Giao thức lưu giữ “trạng thái” khá phức tạp!

- ❖ Lịch sử quá khứ (trạng thái) cần phải được lưu giữ
- ❖ Nếu server/client bị sự cố, thì quan điểm về “trạng thái” của chúng có thể không nhất quán, cần phải được hòa giải

Tầng ứng dụng 2-21

Kết nối HTTP

HTTP không bền vững

- ❖ Chỉ có tối đa một đối tượng được gửi qua một kết nối TCP
 - Kết nối sau đó sẽ được đóng lại
- ❖ Việc tải về nhiều đối tượng sẽ yêu cầu nhiều kết nối

HTTP bền vững

- ❖ Nhiều đối tượng có thể được gửi qua một kết nối TCP duy nhất giữa client và server

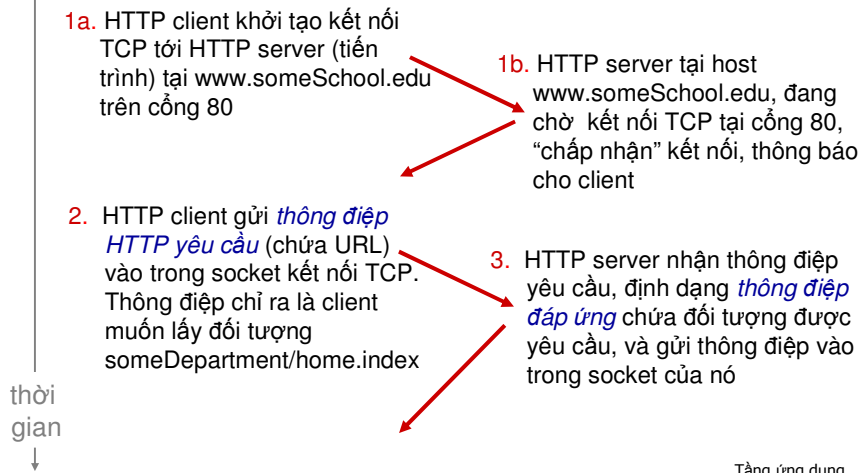
Tầng ứng dụng 2-22

HTTP không bền vững

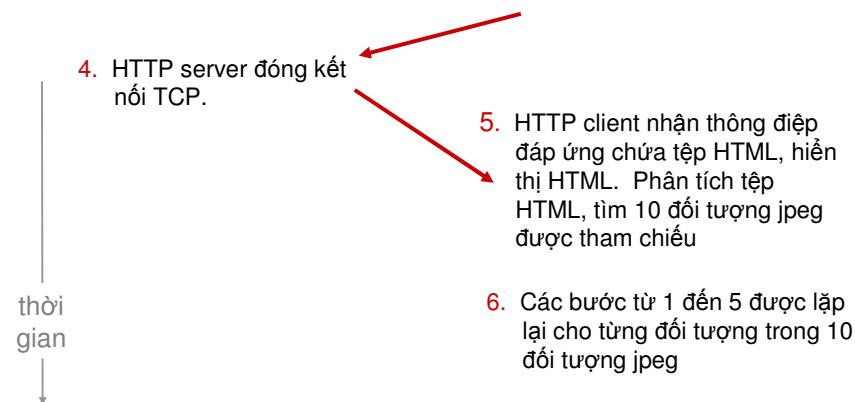
Giả sử người dùng gõ URL:

`www.someSchool.edu/someDepartment/home.index`

(chứa văn bản,
tham chiếu tới 10
ảnh jpeg)



HTTP không bền vững

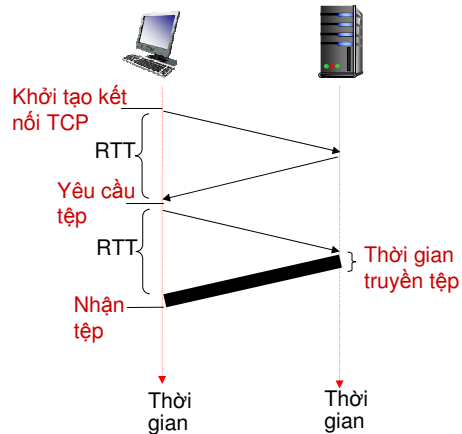


HTTP không bền vững: thời gian đáp ứng

RTT (định nghĩa): thời gian để một gói tin nhỏ đi từ client đến server và quay lại.

Thời gian đáp ứng HTTP:

- ❖ Một RTT để khởi tạo kết nối TCP
- ❖ Một RTT để gửi HTTP yêu cầu và một vài byte HTTP đáp ứng được trả về
- ❖ Thời gian truyền tệp
- ❖ Thời gian đáp ứng của HTTP không bền vững = $2RTT +$ thời gian truyền tệp



Tầng ứng dụng 2-25

HTTP bền vững

Vấn đề với HTTP không bền vững:

- ❖ Cần 2 RTT cho mỗi đối tượng
- ❖ Hệ điều hành liên quan đến *mỗi* kết nối TCP
- ❖ Các trình duyệt thường mở nhiều kết nối TCP song song để lấy các đối tượng được tham chiếu

HTTP bền vững:

- ❖ Server để mở kết nối sau khi gửi đáp ứng
- ❖ Chuỗi các thông điệp HTTP tiếp theo giữa cùng client/server sẽ được gửi thông qua kết nối mở này
- ❖ Client gửi yêu cầu ngay sau khi gặp một đối tượng được tham chiếu
- ❖ Ít nhất là một RTT cho tất cả các đối tượng được tham chiếu

Tầng ứng dụng 2-26

Thông điệp yêu cầu HTTP

- ❖ Có hai loại thông điệp HTTP: *yêu cầu và đáp ứng*
- ❖ Thông điệp HTTP yêu cầu:
 - ASCII (định dạng con người có thể đọc được)

Dòng yêu cầu (các lệnh GET, POST, HEAD) → GET /index.html HTTP/1.1\r\n

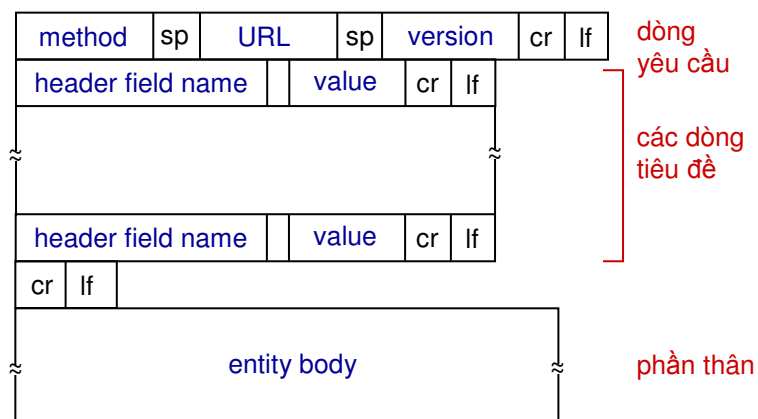
Các dòng tiêu đề (header) → Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n

Ký tự trở về đầu dòng, xuống dòng ở đầu một dòng sẽ cho biết điểm cuối của dòng tiêu đề → \r\n

ký tự trở về đầu dòng
ký tự xuống dòng

Tầng ứng dụng 2-27

Thông điệp HTTP yêu cầu: định dạng tổng quát



Tầng ứng dụng 2-28

Tải lên form input

Phương pháp POST:

- ❖ Trang web thường chứa form input
- ❖ Đầu vào (input) được tải lên server trong phần thân thực thể (entity body)

Phương pháp URL:

- ❖ Dùng phương thức GET
- ❖ Đầu vào được tải lên trong trường URL của dòng yêu cầu:

`www.somesite.com/animalsearch?monkeys&banana`

Các loại phương thức

HTTP/1.0:

- ❖ GET
- ❖ POST
- ❖ HEAD
 - Yêu cầu server đòi đối tượng được yêu cầu ra khỏi đáp ứng

HTTP/1.1:

- ❖ GET, POST, HEAD
- ❖ PUT
 - Tải tệp lên trong phần thân thực thể tới đường dẫn được xác định trong trường URL
- ❖ DELETE
 - Xóa tệp được xác định trong trường URL

Thông điệp đáp ứng HTTP

Dòng trạng thái
(Giao thức
mã trạng thái,
cụm từ trạng
thái)

Các dòng
tiêu đề

Dữ liệu,
ví dụ
tệp HTML
yêu cầu

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
1\r\n
\r\n
data data data data data ...
```

Tầng ứng dụng 2-31

Các mã trạng thái đáp ứng HTTP

- ❖ Mã trạng thái xuất hiện ngay trong dòng đầu tiên của thông điệp đáp ứng từ server đến client.
- ❖ Ví dụ một số mã:

200 OK

- Yêu cầu thành công, đối tượng yêu cầu nằm ở phía sau thông điệp này

301 Moved Permanently

- Đối tượng yêu cầu đã bị di chuyển, vị trí mới được xác định ở phía sau thông điệp này (Location:)

400 Bad Request

- Server không hiểu thông điệp yêu cầu

404 Not Found

- Tài liệu yêu cầu không có trong server này

505 HTTP Version Not Supported

Tầng ứng dụng 2-32

Tự kiểm tra HTTP (phía client)

1. Telnet đến Web server ưa thích của bạn:

```
telnet cis.poly.edu 80
```

Mở kết nối TCP ở cổng 80
(cổng mặc định của HTTP server) tại cis.poly.edu.
Nhập yêu cầu gì đó và gửi tới cổng 80 tại
cis.poly.edu

2. Nhập yêu cầu trong lệnh GET HTTP:

```
GET /~ross/ HTTP/1.1  
Host: cis.poly.edu
```

Bằng cách gõ lệnh này (nhấn enter
2 lần), bạn đã gửi yêu cầu GET tối
thiểu (nhưng đầy đủ) tới HTTP server

3. Xem thông điệp đáp ứng được gửi về từ HTTP server!

(hoặc dùng Wireshark để xem thông điệp yêu cầu/đáp ứng HTTP bất được)

Tầng ứng dụng 2-33

Trạng thái User-server: các cookie

Nhiều trang Web sử dụng các
cookie

Bốn thành phần:

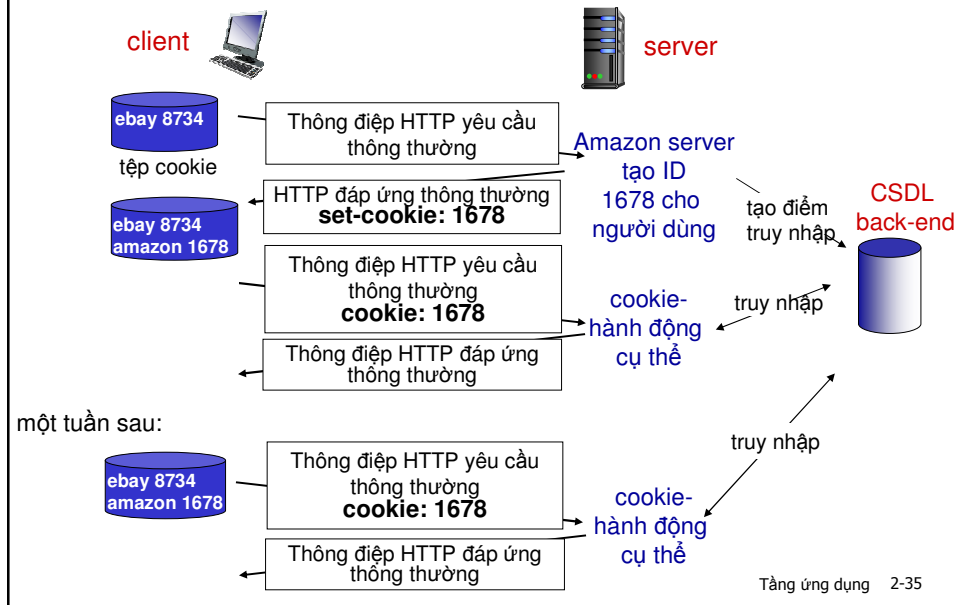
- 1) Dòng tiêu đề cookie của
thông điệp HTTP *đáp
ứng*
- 2) Dòng tiêu đề cookie
trong thông điệp HTTP
yêu cầu tiếp theo
- 3) Tập cookie được lưu giữ
trên host của người
dùng, được quản lý bởi
trình duyệt của người
dùng.
- 4) Cơ sở dữ liệu back-end
tại Web site

Ví dụ:

- ❖ Susan thường truy nhập
Internet từ một PC
- ❖ Cô ấy vào một trang thương
mại điện tử lần đầu tiên
- ❖ Khi yêu cầu HTTP khởi tạo đi
đến trang, trang sẽ tạo ra:
 - Một ID duy nhất
 - Điểm truy nhập vào cơ sở
dữ liệu back-end cho ID

Tầng ứng dụng 2-34

Các cookie: lưu giữ “trạng thái” (tiếp)



Các cookie (tiếp)

Các cookie có thể được dùng cho những việc gì?

- ❖ Cấp phép
- ❖ Giỏ mua hàng
- ❖ Khuyến nghị
- ❖ Trạng thái phiên làm việc của người dùng (Web e-mail)

Ngoài ra *cookie và sự riêng tư:*

- ❖ Cookie cho phép các site biết nhiều hơn về người dùng
- ❖ Người dùng có thể cung cấp tên và e-mail cho các site

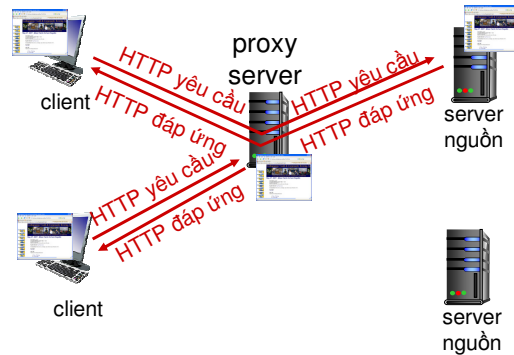
Làm cách nào để lưu giữ “trạng thái”:

- ❖ Các điểm cuối giao thức: duy trì trạng thái tại bên gửi/bên nhận thông qua nhiều giao dịch
- ❖ Các cookie: các thông điệp HTTP mang trạng thái

Web caches (proxy server)

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

- ❖ Người dùng thiết lập trình duyệt: truy nhập Web qua vùng nhớ đệm (cache)
- ❖ Trình duyệt gửi tất cả các yêu cầu HTTP tới cache
 - Nếu đối tượng có trong cache: cache sẽ trả về đối tượng
 - Ngược lại, cache sẽ yêu cầu đối tượng từ server nguồn, sau đó sẽ trả đối tượng về cho client



Tầng ứng dụng 2-37

Web caching

- ❖ Bộ nhớ cache hoạt động ở cả client và server
 - Server đáp ứng cho yêu cầu đầu tiên từ một client
 - Client gửi yêu cầu tới server gốc
- ❖ Cache thường được cài đặt bởi ISP (trường học, công ty, khu dân cư)

Tại sao lại cần Web caching?

- ❖ Làm giảm thời gian đáp ứng cho yêu cầu của client
- ❖ Làm giảm lưu lượng trên một liên kết truy nhập của tổ chức
- ❖ Nếu Internet thực hiện cache nhiều thì sẽ làm cho các nhà cung cấp nội dung “nghèo nàn” phân phối nhiều nội dung đó (cũng tương tự như chia sẻ file P2P).

Tầng ứng dụng 2-38

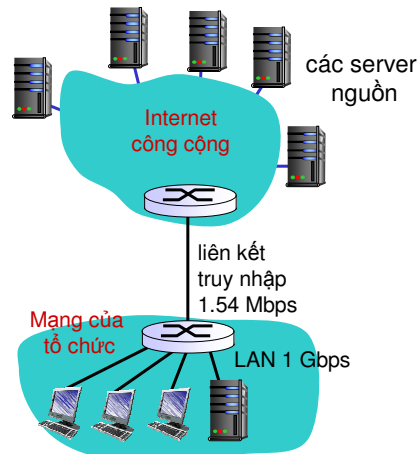
Ví dụ caching

Giả thiết:

- ❖ Kích thước của đối tượng trung bình: 100K bits
- ❖ Tốc độ yêu cầu trung bình từ các trình duyệt tới server nguồn: 15/giây
- ❖ Tốc độ dữ liệu trung bình tới trình duyệt: 1.50 Mbps
- ❖ RTT từ bộ định tuyến của tổ chức đến server nguồn bất kỳ: 2 giây
- ❖ Tốc độ của liên kết truy nhập: 1.54 Mbps

Kết quả:

- ❖ Độ khả dụng của LAN: 15%
- ❖ Độ khả dụng của liên kết truy nhập = **99%** *Vấn đề!*
- ❖ Trễ tổng = trễ của Internet + trễ truy nhập + trễ của LAN
= 2 giây + một số phút + (~ 1 giây)



Tầng ứng dụng 2-39

Ví dụ caching: tăng tốc độ của liên kết truy nhập lên

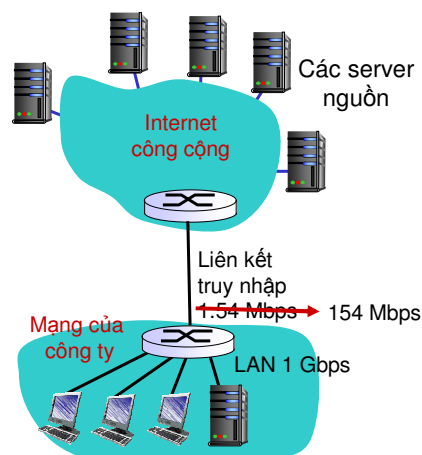
Giả thiết:

- ❖ Kích thước của đối tượng trung bình: 100K bits
- ❖ Tốc độ yêu cầu trung bình từ các trình duyệt tới server nguồn: 15/giây
- ❖ Tốc độ dữ liệu trung bình tới trình duyệt: 1.50 Mbps
- ❖ RTT từ bộ định tuyến của tổ chức đến server nguồn bất kỳ: 2 giây
- ❖ Tốc độ của liên kết truy nhập:
~~1.54 Mbps~~ → **154 Mbps**

Kết quả:

- ❖ Độ khả dụng của LAN: 15%
- ❖ Độ khả dụng của liên kết truy nhập = ~~99%~~ → **9.9%**
- ❖ Tổng trễ = trễ của Internet + trễ truy nhập + trễ của LAN
= 2 giây + ~~một số phút~~ → **Một số giây** + (~ 1 giây)

Chi phí: để làm tăng tốc độ liên kết truy nhập (không hề rẻ!)



Tầng ứng dụng 2-40

Ví dụ caching: cài đặt cache cục bộ

Giả thiết:

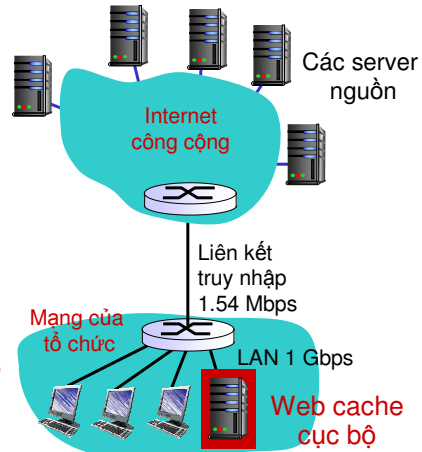
- ❖ Kích thước của đối tượng trung bình: 100K bits
- ❖ Tốc độ yêu cầu trung bình từ các trình duyệt tới server nguồn: 15/giây
- ❖ Tốc độ dữ liệu trung bình tới trình duyệt: 1.50 Mbps
- ❖ RTT từ bộ định tuyến của tổ chức đến server nguồn bất kỳ: 2 giây
- ❖ Tốc độ của liên kết truy nhập: 1.54 Mbps

Kết quả:

- ❖ Độ khả dụng của LAN: 15%
- ❖ Độ khả dụng của liên kết truy nhập = ?
- ❖ Tổng trễ = ?

Tính độ khả dụng của liên kết truy nhập và trễ như thế nào?

Chi phí: web cache (rất rẻ!)

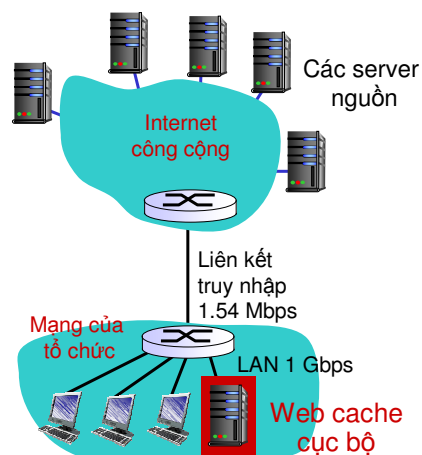


Tầng ứng dụng 2-41

Ví dụ caching: cài đặt cache cục bộ

Tính toán độ khả dụng của liên kết truy nhập, độ trễ khi dùng cache:

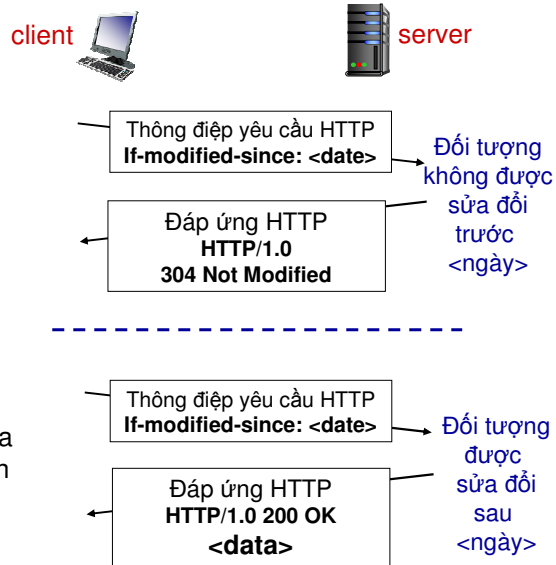
- ❖ Giả sử tỷ lệ hỗ trợ của cache là 0.4
 - 40% yêu cầu được thỏa mãn tại cache, 60% yêu cầu được thỏa mãn tại nguồn
- ❖ Độ khả dụng của liên kết:
 - 60% yêu cầu dùng liên kết truy nhập
- ❖ Tốc độ dữ liệu tới các trình duyệt qua liên kết truy nhập = $0.6 \cdot 1.50 \text{ Mbps} = 0.9 \text{ Mbps}$
 - Độ khả dụng = $0.9 / 1.54 = 0.58$
- ❖ Tổng trễ:
 - = $0.6 \cdot (\text{trễ từ các server nguồn}) + 0.4 \cdot (\text{trễ khi được thỏa mãn tại cache})$
= $0.6 \cdot (2.01\text{s}) + 0.4 \cdot (0.01\text{s})$
= 1.21s
 - Ít hơn so với liên kết 154 Mbps (và chi phí rẻ hơn rất nhiều!)



Tầng ứng dụng 2-42

GET có điều kiện

- ❖ **Mục tiêu:** không gửi đối tượng nếu cache đã cập nhật
 - Không có trễ truyền đối tượng
 - Độ khả dụng của liên kết thấp hơn
- ❖ **cache:** xác định ngày của bản sao cache trong yêu cầu HTTP
If-modified-since: <date>
- ❖ **server:** đáp ứng không chứa đối tượng nếu bản sao cache đã được cập nhật:
HTTP/1.0 304 Not Modified



Chương 2: Nội dung

2.1 Nguyên lý của ứng dụng mạng

- Kiến trúc của ứng dụng
- Các yêu cầu của ứng dụng

2.2 Web và HTTP

2.3 FTP

2.4 Thư điện tử

- SMTP, POP3, IMAP

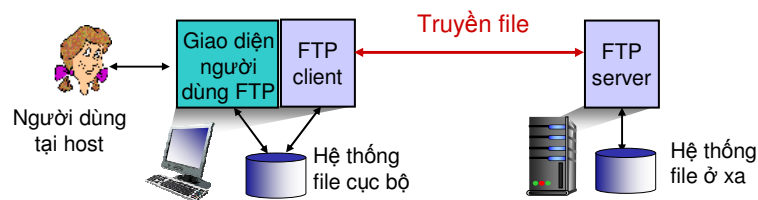
2.5 DNS

2.6 Ứng dụng P2P

2.7 Lập trình socket với UDP và TCP

Tầng ứng dụng 2-44

FTP: giao thức truyền file (file transfer protocol)

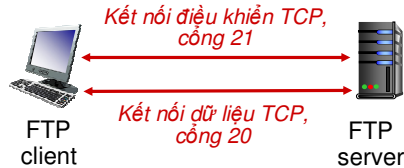


- ❖ Truyền file đến/từ host ở xa
- ❖ Mô hình client/server
 - *client*: phía khởi tạo việc truyền (đến/từ host ở xa)
 - *server*: host ở xa
- ❖ ftp: RFC 959
- ❖ ftp server: cổng 21

Tầng ứng dụng 2-45

FTP: kết nối dữ liệu và kết nối điều khiển riêng biệt nhau

- ❖ FTP client tiếp xúc với FTP server tại cổng số 21, sử dụng TCP
- ❖ client được cấp phép qua kết nối điều khiển
- ❖ client duyệt thư mục ở xa, gửi lệnh qua kết nối điều khiển
- ❖ Khi server nhận lệnh truyền file, **server** mở kết nối TCP thứ 2 (kết nối dữ liệu) (cho file) *tới* client
- ❖ Sau khi truyền một file, server đóng kết nối dữ liệu.
- ❖ server mở một kết nối dữ liệu khác để truyền một file khác.
- ❖ Kết nối điều khiển: “*out of band*”
- ❖ FTP server duy trì “trạng thái”: thư mục hiện hành, sự cấp phép trước đó.



Tầng ứng dụng 2-46

Các lệnh và đáp ứng của FTP

Ví dụ các lệnh:

- ❖ Được gửi như các văn bản dạng mã ASCII qua kênh điều khiển
- ❖ **USER *username***
- ❖ **PASS *password***
- ❖ **LIST** trả về danh sách các file trong thư mục hiện hành
- ❖ **RETR *filename*** trích chọn (lấy) file
- ❖ **STOR *filename*** lưu (đặt) file vào host ở xa

Ví dụ các mã lệnh trả về

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

Tầng ứng dụng 2-47

Chương 2: Nội dung

2.1 Nguyên lý của ứng dụng mạng

- Kiến trúc của ứng dụng
- Các yêu cầu của ứng dụng

2.2 Web và HTTP

2.3 FTP

2.4 Thư điện tử

- SMTP, POP3, IMAP

2.5 DNS

2.6 Ứng dụng P2P

2.7 Lập trình socket với UDP và TCP

Tầng ứng dụng 2-48

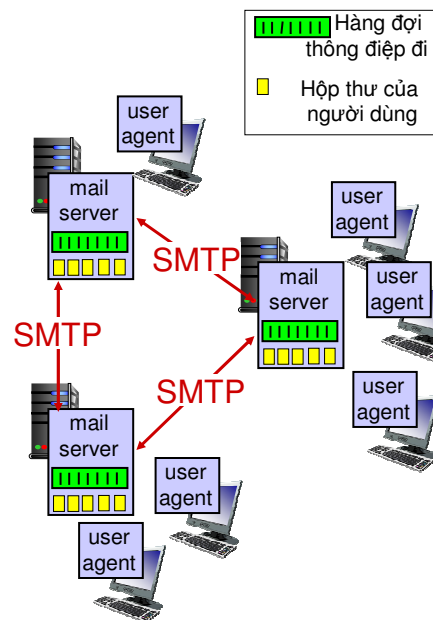
Thư điện tử

Ba thành phần chính:

- ❖ user agent
- ❖ mail server
- ❖ Giao thức truyền thư đơn giản (simple mail transfer protocol): SMTP

User Agent

- ❖ Còn được gọi là “mail reader”
- ❖ Soạn thảo, sửa, đọc các thông điệp thư
- ❖ Ví dụ: Outlook, Thunderbird, iPhone mail client
- ❖ Các thông điệp đi/đến được lưu trên server



Tầng ứng dụng 2-49

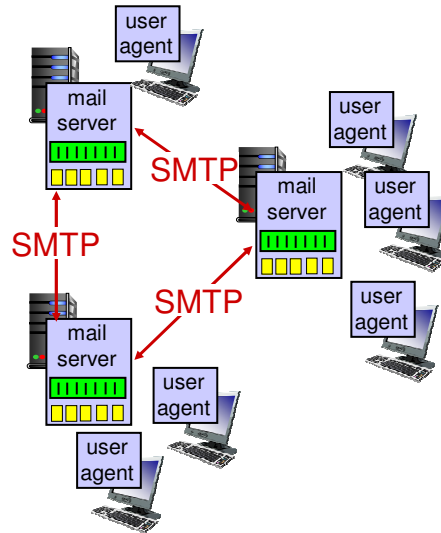
Thư điện tử: mail server

mail server:

- ❖ *Hộp thư (mailbox)* chứa các thông điệp thư đi đến người dùng
- ❖ *Hàng đợi thông điệp (message queue)* của các thông điệp thư đi ra ngoài (chuẩn bị được gửi đi)

Giao thức SMTP giữa các mail server thực hiện gửi các thông điệp thư

- ❖ client: gửi đến mail server
- ❖ server: mail server nhận

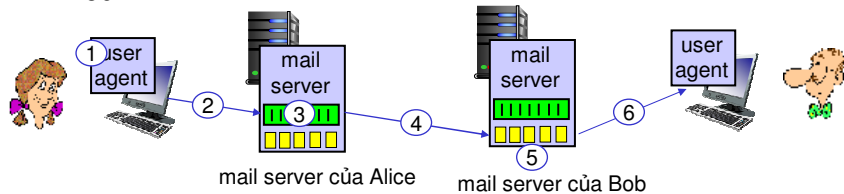


Thư điện tử: SMTP [RFC 2821]

- ❖ Sử dụng TCP để truyền tin cậy thông điệp thư điện tử từ client đến server, trên cổng số 25
- ❖ Truyền trực tiếp: từ server gửi tới server nhận
- ❖ Truyền theo 3 bước
 - Bắt tay (chào hỏi)
 - Truyền thông điệp
 - Đóng
- ❖ Tương tác lệnh/đáp ứng (như HTTP, FTP)
 - **Lệnh:** văn bản ASCII
 - **Đáp ứng:** mã trạng thái và các cụm từ trạng thái
- ❖ Các thông điệp phải ở dạng mã ASCII 7-bit

Kịch bản: Alice gửi thông điệp tới Bob

- 1) Alice dùng UA soạn thảo thông điệp để "gửi tới" bob@someschool.edu
- 2) UA của Alice gửi thông điệp tới mail server của cô ấy; thông điệp được đặt trong hàng đợi thông điệp
- 3) Phía client của SMTP mở kết nối TCP tới mail server của Bob.
- 4) SMTP client gửi thông điệp của Alice qua kết nối TCP
- 5) mail server của Bob đặt thông điệp trong hộp thư của Bob
- 6) Bob kích hoạt user agent để đọc thông điệp thư



Tầng ứng dụng 2-52

Ví dụ tương tác SMTP

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

Tầng ứng dụng 2-53

Tự thử nghiệm tương tác SMTP:

- ❖ `telnet servername 25`
- ❖ Thấy 220 trả lời từ server
- ❖ Nhập các lệnh HELO, MAIL FROM, RCPT TO, DATA, QUIT

Các lệnh trên cho phép gửi email mà không cần dùng email client (reader)

SMTP

- ❖ SMTP sử dụng kết nối bền vững
 - ❖ SMTP yêu cầu thông điệp (phần tiêu đề & phần thân) ở dạng mã ASCII 7-bit
 - ❖ SMTP server sử dụng CRLF . CRLF để xác định điểm kết thúc của thông điệp
- So sánh với HTTP:*
- ❖ HTTP: kéo
 - ❖ SMTP: đẩy
 - ❖ Cả hai đều sử dụng tương tác lệnh/đáp ứng, các mã trạng thái dạng bảng mã ASCII
 - ❖ HTTP: mỗi đối tượng được đóng gói trong thông điệp đáp ứng của nó
 - ❖ SMTP: nhiều đối tượng được gửi trong thông điệp nhiều phần

Định dạng thông điệp thư

SMTP: giao thức trao đổi các thông điệp thư

RFC 822: chuẩn cho định dạng thông điệp văn bản:

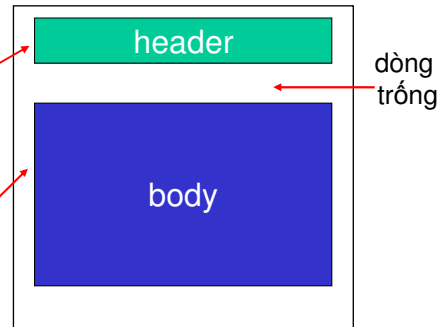
❖ Dòng tiêu đề, ví dụ:

- To:
- From:
- Subject:

Khác với các lệnh SMTP
MAIL FROM, RCPT TO!

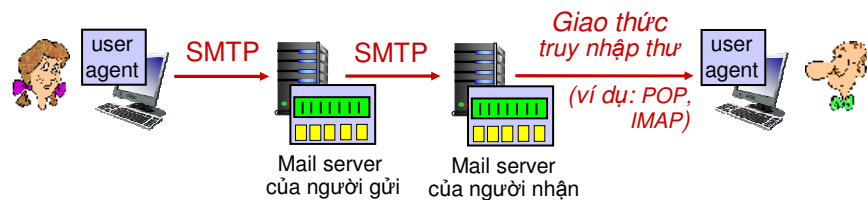
❖ Phân thân: “thông điệp”

- Chỉ dùng các ký tự mã ASCII



Tầng ứng dụng 2-56

Giao thức truy nhập thư



- ❖ **SMTP**: phân phối/lưu trữ thư tới/tại server của người nhận
- ❖ Giao thức truy nhập thư: trích xuất thư từ server
 - **POP**: Post Office Protocol [RFC 1939]: cấp phép, tải thư
 - **IMAP**: Internet Mail Access Protocol [RFC 1730]: có nhiều đặc tính hơn, bao gồm cả những thao tác với các thông điệp được lưu trữ trên server
 - **HTTP**: gmail, Hotmail, Yahoo! Mail,...

Tầng ứng dụng 2-57

Giao thức POP3

Giai đoạn cấp phép

- ❖ Các lệnh của client:
 - **user**: khai báo tên người dùng
 - **pass**: mật khẩu
- ❖ Các đáp ứng của server
 - **+OK**
 - **-ERR**

Giai đoạn giao dịch, client:

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

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

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

Tầng ứng dụng 2-58

POP3 (tiếp) và IMAP

POP3

- ❖ Ví dụ trước dùng POP3 với chế độ “tải và xóa”
 - Bob không thể đọc lại thư nếu thay đổi client
- ❖ POP3 với chế độ “tải và lưu giữ”: sao các thông điệp lên các client khác nhau
- ❖ POP3 không giữ trạng thái của các phiên làm việc

IMAP

- ❖ Lưu giữ tất cả các thông điệp tại một nơi là server
- ❖ Cho phép người dùng tổ chức các thông điệp theo dạng các thư mục
- ❖ Lưu giữ trạng thái của người dùng qua các phiên làm việc:
 - Đặt tên thư mục và ánh xạ giữa các ID của thông điệp với tên thư mục

Tầng ứng dụng 2-59

Chương 2: Nội dung

2.1 Nguyên lý của ứng dụng mạng

- Kiến trúc của ứng dụng
- Các yêu cầu của ứng dụng

2.2 Web và HTTP

2.3 FTP

2.4 Thư điện tử

- SMTP, POP3, IMAP

2.5 DNS

2.6 Ứng dụng P2P

2.7 Lập trình socket với UDP và TCP

Tầng ứng dụng 2-60

DNS: Hệ thống tên miền (domain name system)

Con người: có nhiều định danh:

- CMT, tên, số hộ chiếu

Các host và router trên Internet:

- Địa chỉ IP (32 bit) – được dùng để gán địa chỉ cho các gói tin
- “tên miền”, ví dụ: www.yahoo.com – được con người sử dụng

Hỏi: Làm cách nào để ánh xạ giữa địa chỉ IP và tên miền, và ngược lại?

Hệ thống tên miền:

- ❖ *Cơ sở dữ liệu phân tán* được cài đặt phân cấp với nhiều *server tên miền*
- ❖ *Giao thức tầng ứng dụng:* để các host, các server tên miền truyền thông được thì cần phải *phân giải* các tên miền (diễn dịch địa chỉ/tên miền)
 - Chú ý: chức năng lõi của Internet, được cài đặt như giao thức tầng ứng dụng
 - Phức tạp tại “phần cạnh” của mạng

Tầng ứng dụng 2-61

DNS: các dịch vụ và cấu trúc

Các dịch vụ của DNS

- ❖ Dịch tên host thành địa chỉ IP
- ❖ Bí danh của host
 - Các tên đúng chuẩn, các tên là bí danh
- ❖ Bí danh mail server
- ❖ Phân tán tải
 - Nhân rộng các máy chủ Web: nhiều địa chỉ IP tương ứng với một tên

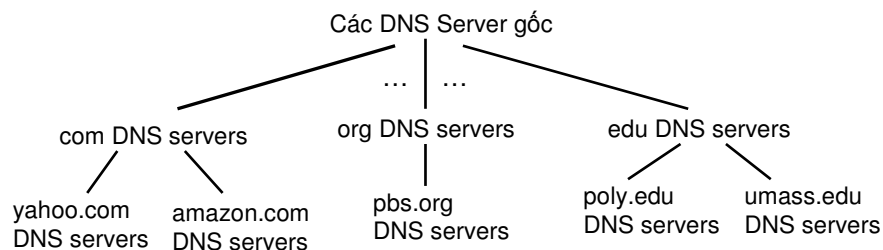
Tại sao không tập trung hóa trong DNS?

- ❖ Một điểm chịu lỗi
- ❖ Lưu lượng
- ❖ Cơ sở dữ liệu tập trung ở xa
- ❖ Bảo trì

Trả lời: Không thể thực hiện với quy mô lớn!

Tầng ứng dụng 2-62

DNS: cơ sở dữ liệu phân tán và phân cấp



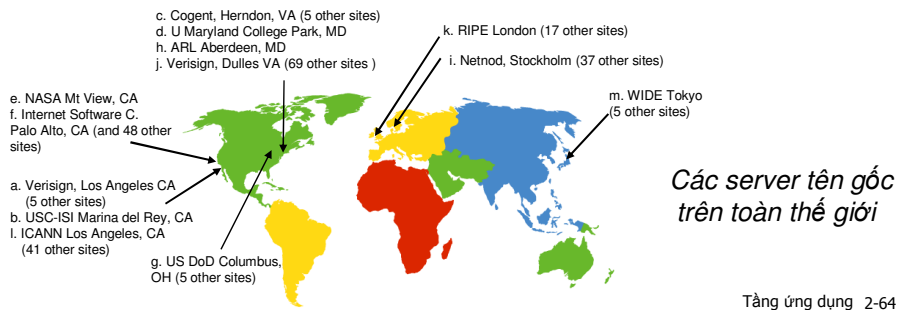
client muốn IP cho www.amazon.com:

- ❖ client truy vấn server gốc để tìm com DNS server
- ❖ client truy vấn .com DNS server để lấy amazon.com DNS server
- ❖ client truy vấn amazon.com DNS server để lấy địa chỉ IP của www.amazon.com

Tầng ứng dụng 2-63

DNS: Các server tên gốc

- ❖ Được tiếp xúc bởi server tên cục bộ mà server này không có khả năng phân giải tên
- ❖ Server tên gốc:
 - Liên lạc với server tên có thẩm quyền nếu việc ánh xạ tên không được xác định
 - Lấy ánh xạ
 - Trả lại ánh xạ cho server tên cục bộ



Các server TLD và server có thẩm quyền

Các server top-level domain (TLD):

- Chịu trách nhiệm cho tên miền com, org, net, edu, aero, jobs, museums, và tất cả các tên miền cấp cao nhất thuộc quốc gia như: uk, fr, ca, jp
- Network Solutions duy trì các server cho .com TLD
- Educause cho .edu TLD

Các server DNS có thẩm quyền:

- Các DNS server của tổ chức, cung cấp các tên host có thẩm quyền cho các ánh xạ IP với các host của tổ chức (ví dụ web và mail).
- Có thể được duy trì bởi các tổ chức hoặc nhà cung cấp dịch vụ

Server tên DNS cục bộ

- ❖ Không hoàn toàn theo cấu trúc phân cấp
- ❖ Mỗi ISP (ISP khu dân cư, công ty, trường học) có một server cục bộ
 - Cũng được gọi là “server tên mặc định”
- ❖ Khi một host tạo một truy vấn DNS, truy vấn này sẽ được gửi tới server DNS cục bộ
 - Có một vùng đệm cục bộ dịch chuyển cặp tên-địa chỉ gần nhất (tuy nhiên vẫn có thể đã bị quá hạn, chưa được cập nhật!)
 - Hoạt động giống như proxy, chuyển tiếp truy vấn vào hệ thống phân cấp

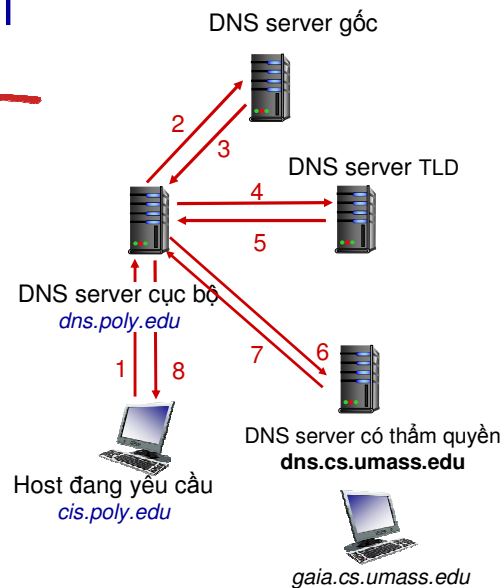
Tầng ứng dụng 2-66

Ví dụ phân giải tên DNS

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

Truy vấn lặp:

- ❖ Server được hỏi sẽ trả lời với tên của server sẽ có thể hỏi được tiếp
- ❖ “Tôi không biết tên đó, nhưng có thể hỏi server này”

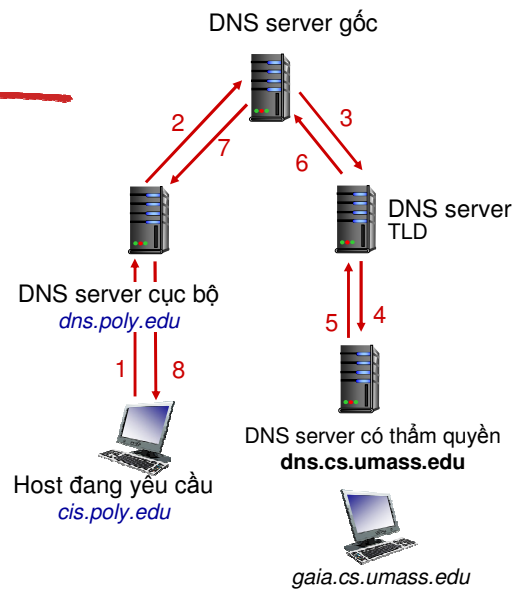


Tầng ứng dụng 2-67

Ví dụ phân giải tên DNS

Truy vấn đệ quy:

- ❖ Đặt trách nhiệm phân giải tên lên server tên được hỏi
- ❖ Tải nặng tại các cấp cao hơn trong hệ thống phân cấp?



Tầng ứng dụng 2-68

DNS: caching và cập nhật các bản ghi

- ❖ Khi server tên học cách ánh xạ, nó sẽ *cache* ánh xạ
 - Mục cache sẽ bị quá hạn (bị xóa) sau một vài lần (TTL)
 - Các server TLD điển hình thường cache trong các server tên cục bộ
 - Do đó, các server tên gốc sẽ không thường xuyên được truy cập
- ❖ Các mục cache có thể bị *quá hạn*
 - Nếu host thay đổi địa chỉ IP, thì Internet sẽ có thể không biết được cho đến khi TTL hết hạn
- ❖ Cơ chế cập nhật, thông báo được đề xuất bởi chuẩn IETF
 - RFC 2136

Tầng ứng dụng 2-69

Bản ghi DNS (DNS records)

DNS: cơ sở dữ liệu phân tán lưu trữ các bản ghi nguồn (resource records - RR)

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

type=A

- **name** là tên máy
- **Value** là địa chỉ IP

type=NS

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

type=CNAME

- **name** là tên bí danh của tên "chuẩn" (tên thật)
- **www.ibm.com** tên thật là **servereast.backup2.ibm.com**
- **value** là tên chuẩn

type=MX

- **value** là tên của mail server được kết hợp với **name**

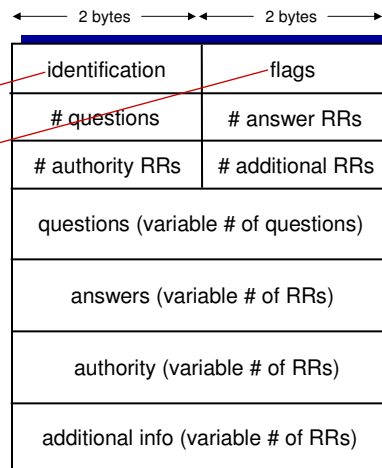
Tầng ứng dụng 2-70

Giao thức, thông điệp DNS

- ❖ Các thông điệp *truy vấn* và *trả lời* đều có cùng *định dạng thông điệp*

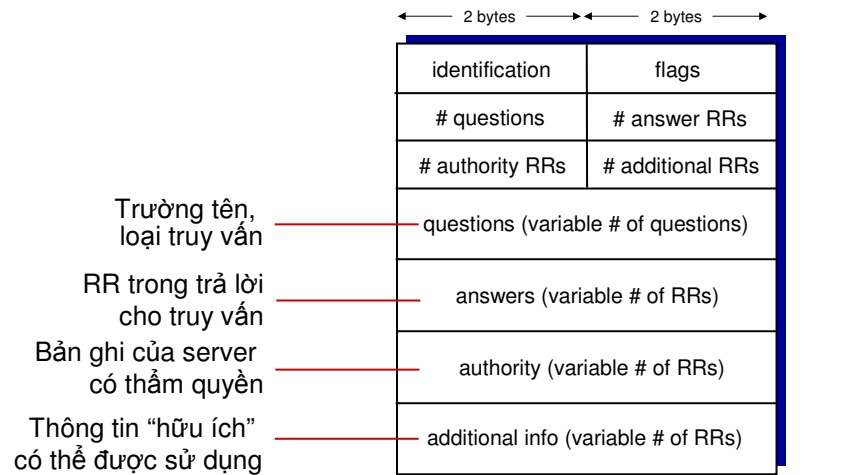
Tiêu đề thông điệp

- ❖ **identification**: 16 bit cho truy vấn/trả lời
- ❖ **flags**:
 - Truy vấn hoặc trả lời
 - Độ quy chờ
 - Độ quy sẵn sàng
 - Trả lời có thẩm quyền



Tầng ứng dụng 2-71

Giao thức, thông điệp DNS



Chèn thêm bản ghi vào trong DNS

- ❖ Ví dụ: Tạo mới "Network Utopia"
- ❖ Đăng ký tên miền networkutopia.com tại *DNS registrar* (Ví dụ: Network Solutions)
 - Cung cấp tên, địa chỉ IP của server tên có thẩm quyền (sơ cấp và thứ cấp) của bạn
 - Registrar sẽ chèn hai bản ghi RR vào trong .com TLD server:
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
- ❖ Tạo bản ghi loại A vào trong server có thẩm quyền cho www.networkutopia.com; bản ghi loại MX record cho networkutopia.com

Tấn công DNS

Tấn công DDoS

- ❖ Tấn công server gốc với lưu lượng lớn
 - Không thành công cho đến nay
 - Lọc lưu lượng
 - DNS server cục bộ cache IP của TLD server, cho phép bỏ qua server gốc
 - Tấn công TLD servers
 - Tiềm tàng nhiều nguy hiểm hơn

Chuyển hướng tấn công

- ❖ Man-in-middle
 - Truy vấn đánh chặn
- ❖ Đầu độc DNS
 - Gửi trả lời giả mạo đến DNS server, mà các server này có thể cache lại

Khai thác DNS cho DDoS

- ❖ Gửi truy vấn với địa chỉ nguồn giả mạo: IP đích
- ❖ Yêu cầu khuếch đại

Tầng ứng dụng 2-74

Chương 2: Nội dung

2.1 Nguyên lý của ứng dụng mạng

- Kiến trúc của ứng dụng
- Các yêu cầu của ứng dụng

2.2 Web và HTTP

2.3 FTP

2.4 Thư điện tử

- SMTP, POP3, IMAP

2.5 DNS

2.6 Ứng dụng P2P

2.7 Lập trình socket với UDP và TCP

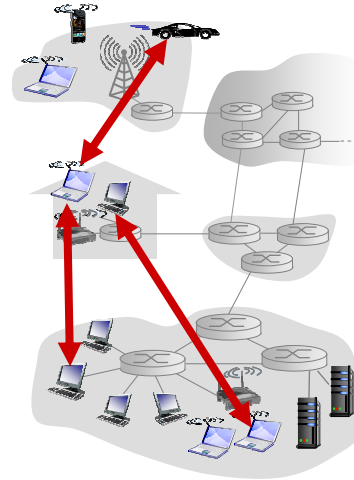
Tầng ứng dụng 2-75

Kiến trúc mạng P2P thuần túy

- ❖ Không cần phải có server luôn hoạt động
- ❖ Các hệ thống cuối tùy ý kết nối trực tiếp
- ❖ Các peer (các nút mạng) không cần kết nối liên tục vào hệ thống mạng và có thể thay đổi địa chỉ IP.

Ví dụ:

- Chia sẻ file (BitTorrent)
- Streaming (KanKan)
- VoIP (Skype)

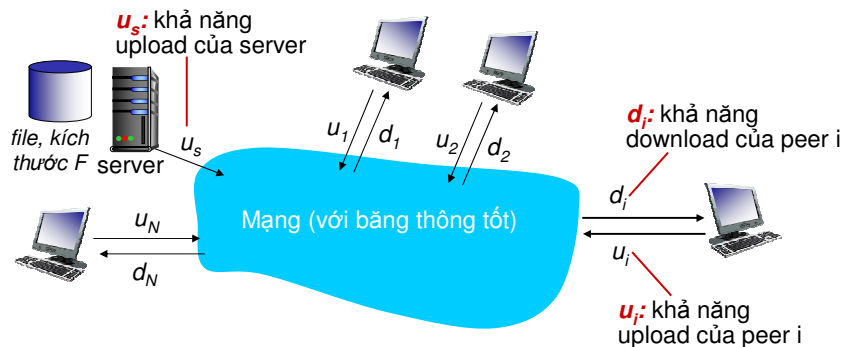


Tầng ứng dụng 2-76

Chia sẻ file: client-server và P2P

Hỏi: Cần mất bao lâu để truyền (phân phối) file (có kích thước F) từ một server đến N peer?

- Khả năng upload/download (tải lên/tải về) của peer bị giới hạn bởi tài nguyên.

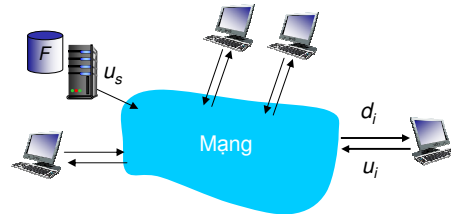


Tầng ứng dụng 2-77

Thời gian truyền file: client-server

❖ **Việc truyền của server:** phải gửi tuần tự (upload) N bản sao của file:

- Thời gian gửi một bản sao: F/u_s
- Thời gian gửi N bản sao: NF/u_s



❖ **Client:** Mỗi client phải download bản sao của file

- d_{\min} = tốc độ download nhỏ nhất của client
- Thời gian download (ứng với d_{\min}) của client: F/d_{\min}

Thời gian để phân phối F tới N client sử dụng cách tiếp cận client-server

$$D_{c-s} \geq \max\{NF/u_s, F/d_{\min}\}$$

Tăng tuyến tính theo N

Tầng ứng dụng 2-78

Thời gian truyền file: P2P

❖ **Việc truyền của server:** phải upload ít nhất một bản sao của file:

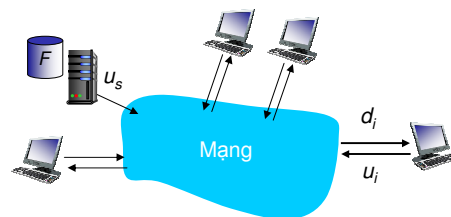
- Thời gian để gửi một bản: F/u_s

❖ **Client:** mỗi client phải download bản sao của file

- Thời gian download (ứng với d_{\min}) của client: F/d_{\min}

❖ **Các client:** phải download NF bits

- Tốc độ upload cao nhất (giới hạn) là $u_s + \sum u_i$



Thời gian để phân phối F tới N client sử dụng cách tiếp cận P2P

$$D_{P2P} \geq \max\{F/u_s, F/d_{\min}, NF/(u_s + \sum u_i)\}$$

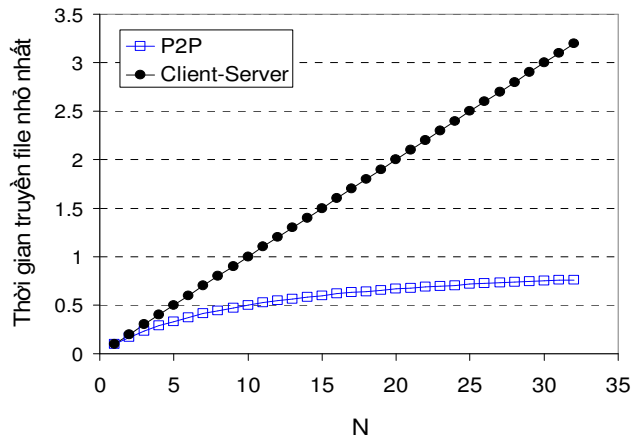
Tăng tuyến tính theo N ...

... nhưng để thực hiện việc này, mỗi peer phải có khả năng phục vụ

Tầng ứng dụng 2-79

Ví dụ so sánh client-server với P2P

Tốc độ upload của client = u , $F/u = 1$ giờ, $u_s = 10u$, $d_{min} \geq u_s$



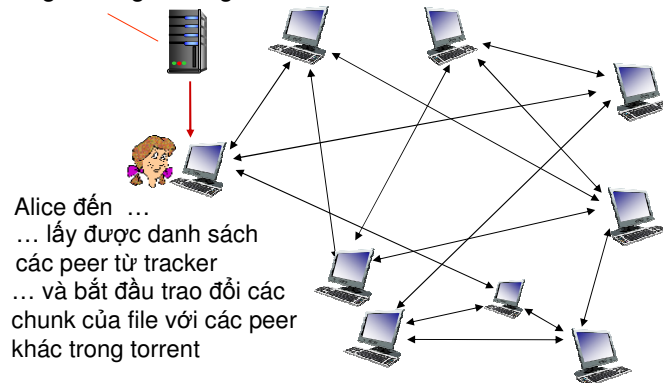
Tầng ứng dụng 2-80

Truyền file P2P: BitTorrent

- ❖ File được chia thành các chunk (phần) có kích thước là 256Kb
- ❖ Các peer trong torrent gửi/nhận các chunk của file

tracker: kiểm tra các peer đang tham gia trong torrent

torrent: nhóm các peer trao đổi các chunk của một file

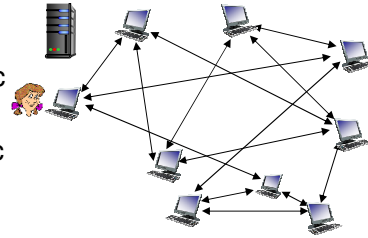


Tầng ứng dụng 2-81

Truyền file P2P: BitTorrent

❖ peer tham gia vào torrent:

- Không có các chunk, nhưng sẽ tích lũy chúng qua thời gian từ các peer khác
- Đăng ký với tracker để nhận được danh sách các peer, kết nối với tập con của các peer (“các hàng xóm”)



- ❖ Trong khi download, peer sẽ upload các chunk tới các peer khác
- ❖ peer có thể thay đổi các peer mà nó sẽ trao đổi các chunk
- ❖ **churn**: các peer có thể đến hoặc đi
- ❖ Khi peer có được toàn bộ file, nó có thể (ích kỷ) rời đi hoặc (vị tha) ở lại trong torrent

Tầng ứng dụng 2-82

BitTorrent: yêu cầu lấy, gửi các chunk của file

Yêu cầu lấy chunk:

- ❖ Tại bất kỳ thời điểm nào, các peer khác nhau đều có các tập con các chunk khác nhau của file.
- ❖ Theo định kỳ, Alice sẽ hỏi mỗi peer về danh sách các chunk mà họ có
- ❖ Alice sẽ yêu cầu các chunk còn thiếu từ các peer, phần hiếm nhất trước

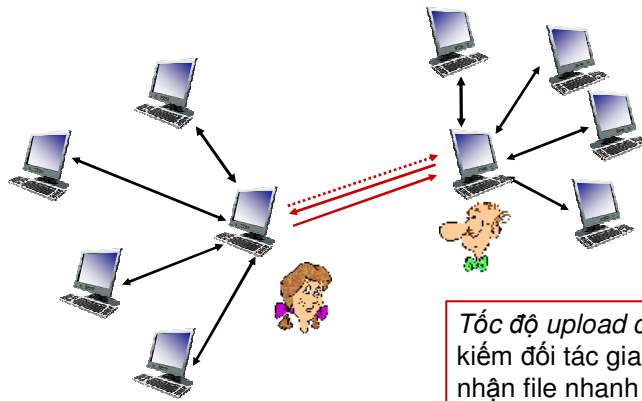
Gửi chunk: tit-for-tat

- ❖ Alice gửi các chunk tới 4 peer hiện đang gửi chunk của cô ấy với **tốc độ cao nhất**
 - Các peer khác đang bị chặn bởi Alice (không nhận được chunk từ cô ấy)
 - Đánh giá lại top 4 sau mỗi 10 giây
- ❖ Cứ mỗi 30 giây: chọn các peer khác một cách ngẫu nhiên, và bắt đầu gửi chunk
 - “Tối ưu hóa không chặn” peer này
 - Peer được chọn mới sẽ được gia nhập vào top 4

Tầng ứng dụng 2-83

BitTorrent: tit-for-tat

- (1) Alice “tối ưu hóa không chặn” Bob
- (2) Alice trở thành một nhà cung cấp trong top 4 của Bob;
- (3) Ngược lại, Bob trở thành một nhà cung cấp trong top 4 của Alice



Tăng ứng dụng 2-84

Bảng băm phân tán - Distributed Hash Table (DHT)

- ❖ DHT: là một *cơ sở dữ liệu P2P phân tán*
- ❖ Cơ sở dữ liệu chứa các cặp (khóa, giá trị) (key, value); Ví dụ:
 - Khóa: Số thứ tự; Giá trị: tên người
 - Khóa: tiêu đề bộ phim; Giá trị: địa chỉ IP
- ❖ Phân tán các cặp (key, value) qua (hàng triệu) các peer
- ❖ Mỗi peer **truy vấn** DHT theo khóa
 - DHT trả lại các giá trị tương ứng với khóa
- ❖ Các peer cũng có thể **chèn thêm (insert)** các cặp (khóa, giá trị)

Tăng ứng dụng 2-85

Hỏi: Làm cách nào để gán khóa cho các peer?

- ❖ Vấn đề chính:
 - Gán các cặp (khóa, giá trị) cho các peer.
- ❖ Ý tưởng cơ bản:
 - Chuyển mỗi khóa thành một số kiểu số nguyên (integer)
 - Gán số nguyên cho từng peer
 - Đặt cặp (khóa, giá trị) trong một peer mà nó là **gần nhất** với khóa

Tầng ứng dụng 2-86

Định danh DHT

- ❖ Gán định danh số nguyên cho mỗi peer trong dãy $[0, 2^n - 1]$ cho một số n .
 - Mỗi định danh được biểu diễn bởi n bit.
- ❖ Yêu cầu mỗi khóa là một số nguyên trong cùng dãy
- ❖ Để nhận được khóa số nguyên, cần băm khóa gốc
 - Ví dụ: khóa = **hash**("Led Zeppelin IV")
 - Đây là lý do tại sao nó được gọi là một **bảng "băm" phân tán**

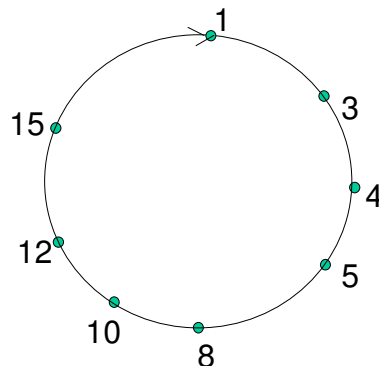
Tầng ứng dụng 2-87

Gán khóa cho các peer

- ❖ Quy tắc: gán khóa cho peer mà có ID *gần nhất*.
- ❖ Trong bài giảng: gần nhất là *giá trị kế liền ngay* với khóa.
- ❖ Ví dụ: $n=4$; các peer: 1,3,4,5,8,10,12,14;
 - khóa = 13, thì peer kế liền với nó là 14
 - khóa = 15, thì peer kế liền là 1

Tầng ứng dụng 2-88

DHT vòng (1)

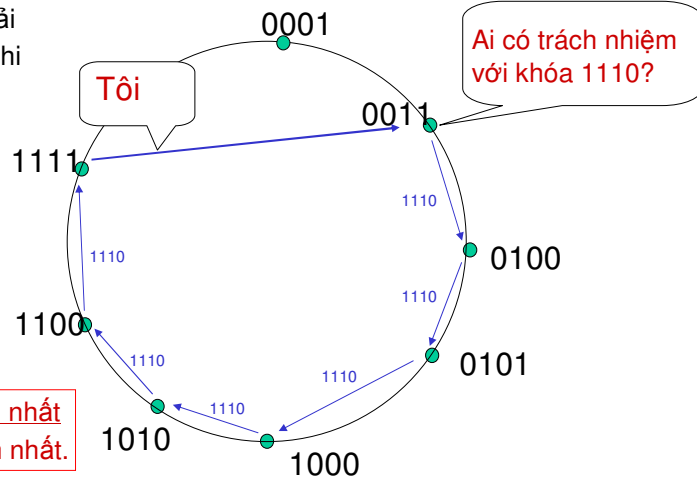


- ❖ Mỗi peer *chỉ biết* về peer kế tiếp và peer tiền nhiệm ngay trước nó
- ❖ Mạng che phủ (“overlay network”)

Tầng ứng dụng 2-89

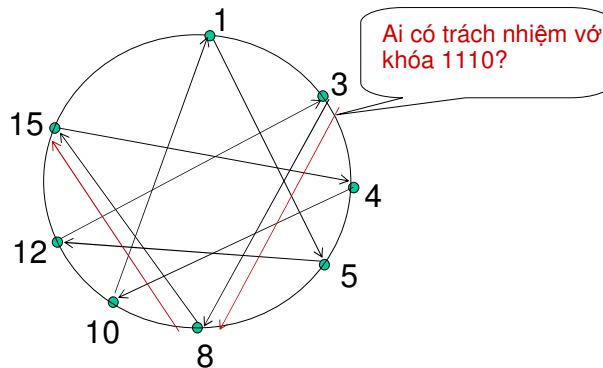
DHT vòng (1)

Trung bình cần $O(N)$ thông điệp để giải quyết truy vấn, khi có N peer



Tầng ứng dụng 2-90

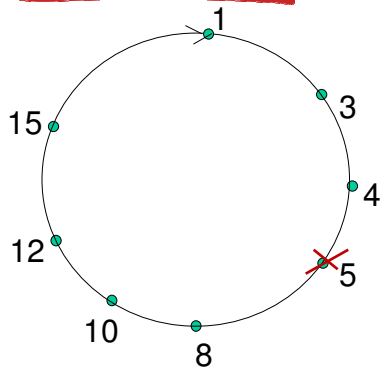
DHT vòng với peer tắt (shortcut)



- ❖ Mỗi peer giữ theo dõi địa chỉ IP của peer kế tiếp, tiền nhiệm và các peer tắt.
- ❖ Sẽ giảm từ 6 xuống còn 2 thông điệp.
- ❖ Có thể thiết kế các peer tắt để có $O(\log N)$ hàng xóm, thì sẽ có $O(\log N)$ thông điệp trong truy vấn

Tầng ứng dụng 2-91

Peer churn



Ví dụ: peer 5 đột ngột rời đi

- ❖ peer 4 phát hiện ra peer 5 rời đi, nó sẽ làm việc với peer 8 kế liền ngay với nó; hỏi peer 8 xem peer nào sẽ làm peer kế liền ngay của nó được; sau đó sẽ làm việc tiếp với peer kế liền ngay với peer 8 này (kế liền ngay thứ cấp).
- ❖ peer 13 muốn kết nối với ai?

Xử lý peer churn:

- ❖ Các peer có thể đến hoặc đi (churn)
- ❖ Mỗi peer biết được địa chỉ của 2 peer kế của nó
- ❖ Định kỳ mỗi peer sẽ ping đến 2 peer kế của nó để kiểm tra sự tồn tại
- ❖ Nếu peer kế liền ngay nó rời đi, thì nó sẽ chọn peer kế tiếp để làm peer kế liền ngay của nó

Chương 2: Nội dung

2.1 Nguyên lý của ứng dụng mạng

- Kiến trúc của ứng dụng
- Các yêu cầu của ứng dụng

2.2 Web và HTTP

2.3 FTP

2.4 Thư điện tử

- SMTP, POP3, IMAP

2.5 DNS

2.6 Ứng dụng P2P

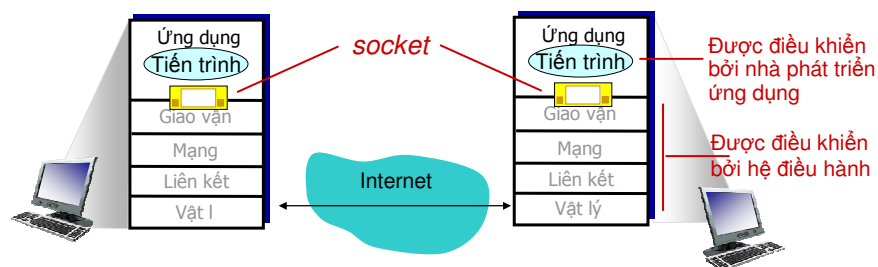
2.7 Lập trình socket với UDP và TCP

Tầng ứng dụng 2-93

Lập trình Socket

Mục đích: hiểu được cách xây dựng ứng dụng truyền thông client/server dùng socket

socket: là cánh cửa giữa các tiến trình ứng dụng và giao thức giao vận end-to-end



Tầng ứng dụng 2-94

Lập trình Socket

Hai loại socket cho hai dịch vụ tầng giao vận:

- **UDP:** truyền các gói tin không tin cậy
- **TCP:** truyền tin cậy, truyền dòng byte có hướng

Ví dụ ứng dụng:

1. Client đọc vào một dòng ký tự (dữ liệu) từ bàn phím và gửi dữ liệu đến server.
2. Server nhận dữ liệu và chuyển các ký tự thành dạng ký tự viết hoa.
3. Server gửi dữ liệu đã được chuyển thành dạng viết hoa về cho client.
4. Client nhận dữ liệu và hiển thị dòng ký tự lên màn hình.

Lập trình mạng trên Java

- ❖ Gói java.net
 - InetAddress
 - ServerSocket
 - Socket
 - URL
 - URLConnection
 - DatagramSocket

Lập trình mạng trên Java

❖ InetAddress class

- Class mô tả về địa chỉ IP
- Các phương thức `getLocalHost`, `getByName`, hay `getAllByName` để tạo một `InetAddress` instance:

```
public static InetAddress InetAddress.getByName(String hostname)
public static InetAddress [] InetAddress.getAllByName(String hostname)
public static InetAddress InetAddress.getLocalHost()
```
- Để lấy địa chỉ IP hay tên dùng các phương thức:

```
getHostAddress()
getHostName()
```

Tầng ứng dụng 2-97

Lập trình Socket với UDP

UDP: không có “kết nối” giữa client & server

- ❖ Không bắt tay trước khi gửi dữ liệu
- ❖ Bên gửi gắn địa chỉ IP và số hiệu cổng đích vào trong mỗi gói tin
- ❖ Bên nhận sẽ trích địa chỉ IP và số hiệu cổng của bên gửi từ gói tin nhận được

UDP: dữ liệu được truyền có thể bị mất hoặc không đúng trình tự khi nhận

Quan điểm ứng dụng:

- ❖ UDP cung cấp truyền không tin cậy theo các nhóm byte (“các gói tin”) giữa client và server

Tầng ứng dụng 2-98

Tương tác client/server socket: UDP

server (chạy trên serverIP)

Tạo socket, port= x:
`serverSocket =
socket(AF_INET,SOCK_DGRAM)`

↓
Đọc datagram từ
`serverSocket`

↓
Ghi trả lời vào
`serverSocket`
địa chỉ client,
số hiệu cổng
cụ thể

client

Tạo socket:
`clientSocket =
socket(AF_INET,SOCK_DGRAM)`

↓
Tạo datagram với IP của server
và port=x; gửi datagram qua
`clientSocket`

↓
Đọc datagram từ
`clientSocket`

↓
Đóng
`clientSocket`

Tầng ứng dụng 2-99

Ví dụ: UDP client (Java client)

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

class UDPClient {
    public static void main(String args[]) throws Exception
    {
        tạo
        input stream → BufferedReader inFromUser =  
                    new BufferedReader(new InputStreamReader(System.in));

        tạo
        client socket → DatagramSocket clientSocket = new DatagramSocket();

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

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

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

Tầng ứng dụng 2-100

Ví dụ: UDP client (Java client)

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

    clientSocket.send(sendPacket);

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

    clientSocket.receive(receivePacket);

    String modifiedSentence =
        new String(receivePacket.getData());

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

tạo dữ liệu gửi datagram, độ dài, địa chỉ IP, port →

gửi datagram đến server →

đọc datagram từ server →

Tầng ứng dụng 2-101

Ví dụ: UDP server (Java server)

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

class UDPServer {
    public static void main(String args[]) throws Exception
    {
        DatagramSocket serverSocket = new DatagramSocket(9876);

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

        while(true)
        {
            DatagramPacket receivePacket =
                new DatagramPacket(receiveData, receiveData.length);

            serverSocket.receive(receivePacket);

            Tạo không gian để nhận datagram →
            Nhận datagram →
        }
    }
}
```

tạo datagram socket tại port 9876 →

Tầng ứng dụng 2-102

Ví dụ: UDP server (Java server)

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

    lấy địa chỉ IP, số hiệu cổng của người gửi → InetAddress IPAddress = receivePacket.getAddress();
    int port = receivePacket.getPort();

String capitalizedSentence = sentence.toUpperCase();

sendData = capitalizedSentence.getBytes();

    tạo datagram để gửi tới client → DatagramPacket sendPacket =
        new DatagramPacket(sendData, sendData.length, IPAddress,
            port);

    ghi datagram vào socket → serverSocket.send(sendPacket);
}
}
}
    kết thúc vòng lặp while, quay lại và chờ datagram khác
```

Tầng ứng dụng 2-103

Lập trình socket với TCP

client phải tiếp xúc với server

- ❖ Tiến trình server phải chạy trước
- ❖ server phải tạo socket (cửa) để đón client tiếp xúc

client tiếp xúc với server bằng cách:

- ❖ Tạo TCP socket, xác định địa chỉ IP, số hiệu cổng của tiến trình server
- ❖ *Khi client tạo socket:* TCP client sẽ thiết lập kết nối tới TCP server

❖ Khi được tiếp xúc bởi client, *TCP server sẽ tạo socket mới* cho tiến trình server để truyền thông với client

- Cho phép server “nói chuyện” với nhiều client
- Các số hiệu cổng nguồn được dùng để phân biệt các client (xem thêm trong Chương 3)

Quan điểm ứng dụng:

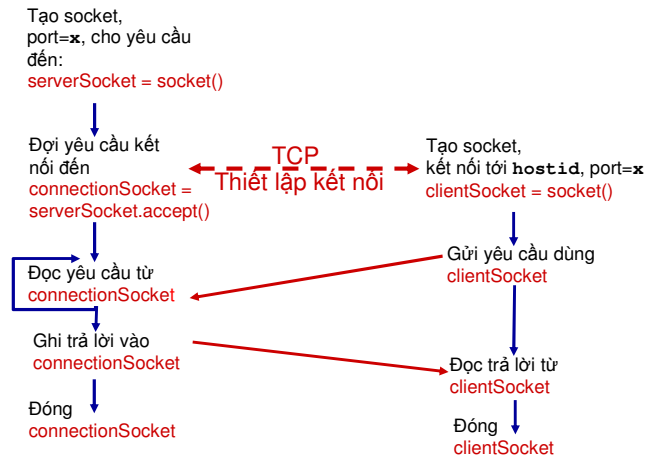
TCP cung cấp truyền tin cậy, truyền dòng byte theo đúng trình tự giữa client và server.

Tầng ứng dụng 2-104

Tương tác client/server socket: TCP

server (chạy trên `host.id`)

client



Tầng ứng dụng 2-105

Ví dụ: TCP client (Java client)

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

    public static void main(String argv[]) throws Exception
    {
        String sentence;
        String modifiedSentence;

        tạo input stream ] BufferedReader inFromUser =
            new BufferedReader(new InputStreamReader(System.in));

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

        tạo output stream gắn vào socket ] DataOutputStream outToServer =
            new DataOutputStream(clientSocket.getOutputStream());
```

Tầng ứng dụng 2-106

Ví dụ: TCP client (Java client)

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

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

        modifiedSentence = inFromServer.readLine();

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

        clientSocket.close();

    }
}
```

tạo input stream
gắn vào socket

gửi dòng
đến server

đọc dòng
từ server

Tầng ứng dụng 2-107

Ví dụ: TCP server (Java server)

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

class TCPServer {

    public static void main(String argv[]) throws Exception
    {
        String clientSentence;
        String capitalizedSentence;

        ServerSocket welcomeSocket = new ServerSocket(6789);

        while(true) {

            Socket connectionSocket = welcomeSocket.accept();

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

            clientSentence = inFromClient.readLine();
            capitalizedSentence = capitalize(clientSentence);

            PrintWriter outToClient =
            new PrintWriter(connectionSocket.getOutputStream(), true);
            outToClient.println(capitalizedSentence);

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

tạo socket đón
tiếp xúc tại
port 6789

chờ client tiếp
xúc với server

tạo input stream,
gắn vào socket

Tầng ứng dụng 2-108

Ví dụ: TCP server (Java server)

```
    tạo output stream,
    gắn vào socket } DataOutputStream outToClient =
                    new DataOutputStream(connectionSocket.getOutputStream());

    đọc dòng
    từ socket } clientSentence = inFromClient.readLine();

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

    ghi dòng ra
    từ socket } outToClient.writeBytes(capitalizedSentence);
              }
            }
          }
        }
      }
    }
```

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

Tầng ứng dụng 2-109

Xây dựng một Web server đơn giản

- ❖ Quản lý một yêu cầu HTTP
- ❖ Chấp nhận yêu cầu
- ❖ Phân tích cú pháp phần tiêu đề (header)
- ❖ Lấy file được yêu cầu từ hệ thống file của server
- ❖ Tạo thông điệp đáp ứng HTTP:
 - các dòng header + file
- ❖ Gửi đáp ứng đến client
- ❖ Sau khi tạo server, có thể yêu cầu file dùng trình duyệt (ví dụ: IE)
- ❖ Xem giáo trình để biết thêm chi tiết

Tầng ứng dụng 2-110

Chương 2: Tổng kết

Trình bày các vấn đề liên quan đến ứng dụng mạng!

- ❖ Kiến trúc của ứng dụng
 - client-server
 - P2P
- ❖ Các yêu cầu dịch vụ của ứng dụng:
 - Truyền tin cậy, băng thông, trễ
- ❖ Mô hình dịch vụ giao vận của Internet
 - Hướng kết nối, truyền tin cậy: TCP
 - Truyền không tin cậy, truyền gói tin: UDP
- ❖ Các giao thức cụ thể:
 - HTTP
 - FTP
 - SMTP, POP, IMAP
 - DNS
 - P2P: BitTorrent, DHT
- ❖ Lập trình socket: TCP socket, UDP socket

Tầng ứng dụng 2-111

Chương 2: Tổng kết

Quan trọng hơn: được học về các giao thức!

- ❖ Trao đổi giữa các thông điệp yêu cầu/đáp ứng:
 - client yêu cầu thông tin hoặc dịch vụ
 - server đáp ứng với dữ liệu, hoặc mã trạng thái
- ❖ Định dạng thông điệp:
 - Phần tiêu đề (header): các trường với thông tin về dữ liệu
 - Dữ liệu: thông tin được truyền thông
- Các vấn đề quan trọng:***
 - ❖ Thông điệp điều khiển và thông điệp dữ liệu
 - in-band, out-of-band
 - ❖ Tập trung hóa và không tập trung hóa
 - ❖ Không trạng thái và có trạng thái
 - ❖ Truyền thông điệp tin cậy và truyền thông điệp không tin cậy
 - ❖ “Sự phức tạp tại phần cạnh của mạng”

Tầng ứng dụng 2-112

Chapter 2: outline

2.1 principles of network applications

- app architectures
- app requirements

2.2 Web and HTTP

2.3 FTP

2.4 electronic mail

- SMTP, POP3, IMAP

2.5 DNS

2.6 P2P applications

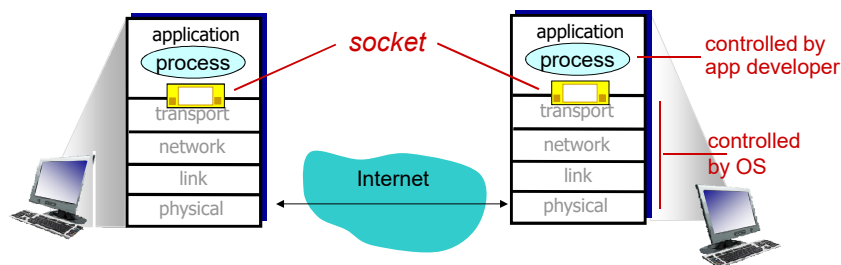
2.7 socket programming with UDP and TCP

Application Layer 2-93

Socket programming

goal: learn how to build client/server applications that communicate using sockets

socket: door between application process and end-end-transport protocol



Application Layer 2-94

Socket programming

Two socket types for two transport services:

- **UDP:** unreliable datagram
- **TCP:** reliable, byte stream-oriented

Application Example:

1. Client reads a line of characters (data) from its keyboard and sends the data to the server.
2. The server receives the data and converts characters to uppercase.
3. The server sends the modified data to the client.
4. The client receives the modified data and displays the line on its screen.

Application Layer 2-95

Socket programming *with UDP*

UDP: no “connection” between client & server

- ❖ no handshaking before sending data
- ❖ sender explicitly attaches IP destination address and port # to each packet
- ❖ rcvr extracts sender IP address and port# from received packet

UDP: transmitted data may be lost or received out-of-order

Application viewpoint:

- ❖ UDP provides *unreliable* transfer of groups of bytes (“datagrams”) between client and server

Application Layer 2-96

Client/server socket interaction: UDP

server (running on serverIP)

create socket, port= x:
serverSocket =
socket(AF_INET,SOCK_DGRAM)

read datagram from
serverSocket

write reply to
serverSocket
specifying
client address,
port number

client

create socket:
clientSocket =
socket(AF_INET,SOCK_DGRAM)

Create datagram with server IP and
port=x; send datagram via
clientSocket

read datagram from
clientSocket

close
clientSocket

Application 2-97

Example app: UDP client

Python UDPClient

```
include Python's socket library → from socket import *
serverName = 'hostname'
serverPort = 12000

create UDP socket for server → clientSocket = socket(socket.AF_INET,
                                                    socket.SOCK_DGRAM)

get user keyboard input → message = raw_input('Input lowercase sentence:')
Attach server name, port to message; send into socket → clientSocket.sendto(message,(serverName, serverPort))

read reply characters from socket into string → modifiedMessage, serverAddress =
                                                    clientSocket.recvfrom(2048)

print out received string and close socket → print modifiedMessage
                                                    clientSocket.close()
```

Application Layer 2-98

Example app: UDP server

Python UDPServer

```
from socket import *
serverPort = 12000
create UDP socket → serverSocket = socket(AF_INET, SOCK_DGRAM)
bind socket to local port number 12000 → serverSocket.bind(("", serverPort))
print "The server is ready to receive"
loop forever → while 1:
  Read from UDP socket into message, getting client's address (client IP and port) → message, clientAddress = serverSocket.recvfrom(2048)
  modifiedMessage = message.upper()
  send upper case string back to this client → serverSocket.sendto(modifiedMessage, clientAddress)
```

Application Layer 2-99

Socket programming *with TCP*

client must contact server

- ❖ server process must first be running
- ❖ server must have created socket (door) that welcomes client's contact

client contacts server by:

- ❖ Creating TCP socket, specifying IP address, port number of server process
- ❖ *when client creates socket:* client TCP establishes connection to server TCP

- ❖ when contacted by client, *server TCP creates new socket* for server process to communicate with that particular client
 - allows server to talk with multiple clients
 - source port numbers used to distinguish clients (more in Chap 3)

application viewpoint:

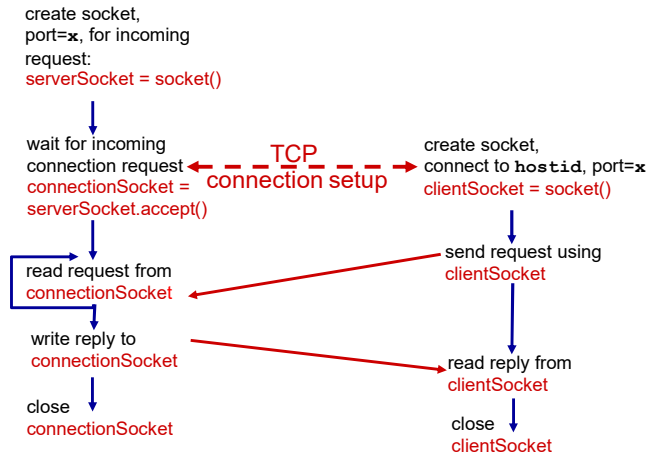
TCP provides reliable, in-order byte-stream transfer ("pipe") between client and server

Application Layer 2-100

Client/server socket interaction: TCP

server (running on `hostid`)

client



Application Layer 2-101

Example app:TCP client

Python TCPClient

```
from socket import *
serverName = 'servername'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = raw_input('Input lowercase sentence:')
clientSocket.send(sentence)
modifiedSentence = clientSocket.recv(1024)
print 'From Server:', modifiedSentence
clientSocket.close()
```

create TCP socket for server, remote port 12000 →

No need to attach server name, port →

Application Layer 2-102

Example app: TCP server

Python TCP Server

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', serverPort))
serverSocket.listen(1)
print 'The server is ready to receive'

while 1:
    connectionSocket, addr = serverSocket.accept()

    sentence = connectionSocket.recv(1024)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence)
    connectionSocket.close()
```

create TCP welcoming socket →

server begins listening for incoming TCP requests →

loop forever →

server waits on accept() for incoming requests, new socket created on return →

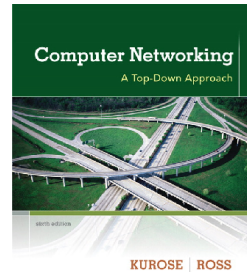
read bytes from socket (but not address as in UDP) →

close connection to this client (but *not* welcoming socket) →

Application Layer 2-103

Chương 3

Tầng giao vận



*Computer
Networking: A Top
Down Approach*
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Người dịch: Nguyễn Thanh Thủy

Tài liệu được dịch cho mục đích giảng dạy (được sự đồng ý của tác giả).

© All material copyright 1996-2012.
J.F. Kurose and K.W. Ross, All Rights Reserved

Tầng giao vận 3-1

Chương 3: Tầng giao vận

Mục đích:

- ❖ Hiểu được các nguyên lý đằng sau các dịch vụ tầng giao vận:
 - Ghép kênh/phân kênh (multiplexing, demultiplexing)
 - Truyền dữ liệu tin cậy
 - Điều khiển luồng
 - Điều khiển tắc nghẽn
- ❖ Nghiên cứu về các giao thức tầng giao vận trong mạng Internet:
 - UDP: vận chuyển không kết nối
 - TCP: Vận chuyển tin cậy, hướng kết nối
 - Điều khiển tắc nghẽn trong TCP

Tầng giao vận 3-2

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)
- Truyền dữ liệu tin cậy
- Điều khiển luồng
- Quản lý kết nối

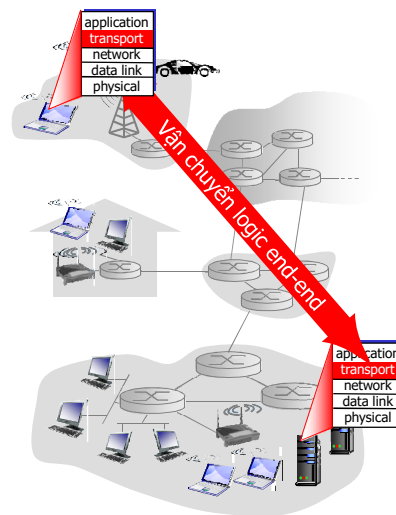
3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-3

Các dịch vụ và giao thức tầng giao vận

- ❖ Cung cấp **truyền thông logic** giữa các tiến trình ứng dụng chạy trên các host khác nhau.
- ❖ Giao thức tầng giao vận chạy trên các hệ thống đầu cuối
 - Phía gửi: cắt các thông điệp ứng dụng thành các **đoạn (segment)**, chuyển xuống tầng mạng
 - Phía nhận: Tập hợp lại các đoạn thành các thông điệp, chuyển lên tầng ứng dụng.
- ❖ Có nhiều hơn một giao thức tầng giao vận dành cho các ứng dụng
 - Internet: TCP và UDP



Tầng giao vận 3-4

Tầng giao vận và tầng mạng

- ❖ **Tầng mạng:** truyền thông logic giữa các host
- ❖ **Tầng giao vận:** truyền thông logic giữa các tiến trình
 - Dựa vào và nâng cao các dịch vụ tầng mạng

Tình huống tương tự:

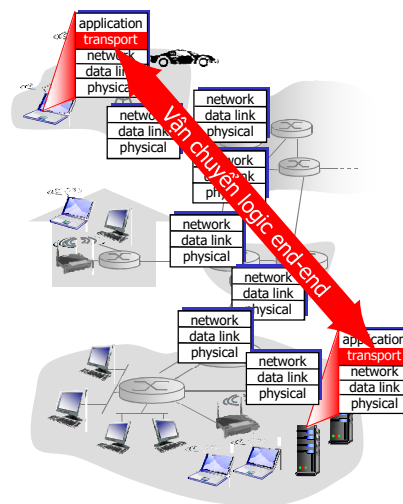
12 em bé nhà Ann gửi thư đến 12 em bé nhà Bill:

- ❖ Các host = Các ngôi nhà
- ❖ Các tiến trình = các em bé
- ❖ Thông điệp ứng dụng = Nội dung bức thư (trong bì thư)
- ❖ Giao thức giao vận = Quy ước giữa các em bé nhà Ann và nhà Bill
- ❖ Giao thức tầng mạng = Dịch vụ bưu điện

Tầng giao vận 3-5

Các giao thức tầng giao vận trên Internet

- ❖ Truyền tin cậy, theo thứ tự: TCP
 - Điều khiển tắc nghẽn
 - Điều khiển luồng
 - Thiết lập kết nối
- ❖ Truyền không tin cậy, không theo thứ tự: UDP
 - Mở rộng của giao thức IP
- ❖ Không có các dịch vụ:
 - Đảm bảo trễ
 - Đảm bảo băng thông



Tầng giao vận 3-6

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)
- Truyền dữ liệu tin cậy
- Điều khiển luồng
- Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-7

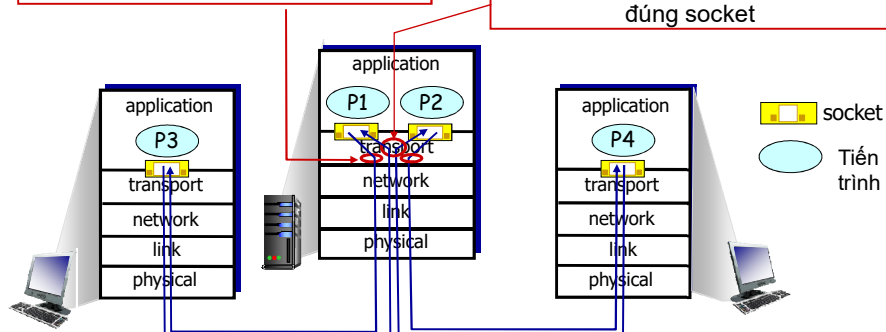
Ghép kênh/Phân kênh

Ghép kênh tại phía gửi:

Xử lý dữ liệu từ nhiều socket, thêm phần tiêu đề tầng giao vận (sau này dùng cho việc phân kênh)

Phân kênh tại phía nhận:

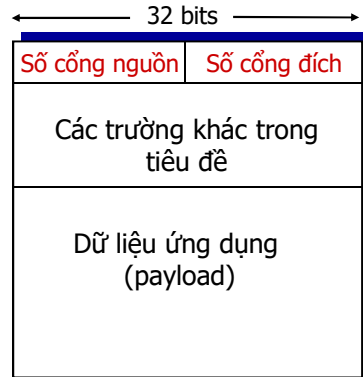
Sử dụng thông tin trong phần tiêu đề để phân phối các đoạn dữ liệu (segment) đã nhận được đến đúng socket



Tầng giao vận 3-8

Việc phân kênh được thực hiện như thế nào?

- ❖ Host nhận các IP datagram
 - Mỗi datagram có địa chỉ nguồn IP và địa chỉ IP đích
 - Mỗi datagram mang một đoạn dữ liệu của tầng giao vận
 - Mỗi segment có số hiệu cổng nguồn và số hiệu cổng đích
- ❖ Host sử dụng **địa chỉ IP & số hiệu cổng** để định hướng đoạn đến socket phù hợp



Định dạng TCP/UDP segment

Tầng giao vận 3-9

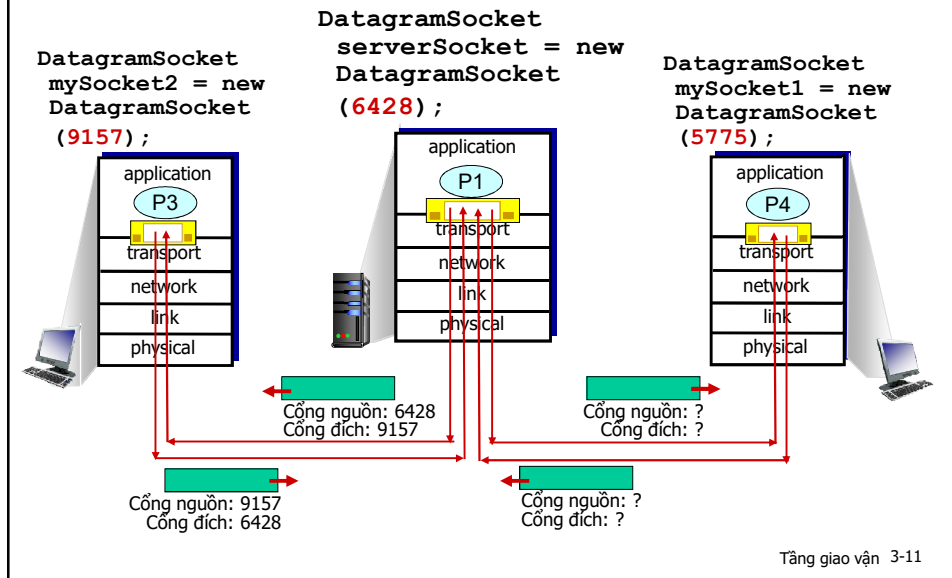
Phân kênh hướng không kết nối

- ❖ Tạo các socket có số hiệu cổng cục bộ của host:

```
DatagramSocket mySocket1  
= new DatagramSocket(12534);
```
 - ❖ Khi tạo datagram để gửi vào trong UDP socket, cần phải xác định:
 - Địa chỉ IP đích
 - Số hiệu cổng đích
 - ❖ Khi host nhận UDP segment:
 - Kiểm tra số hiệu cổng đích trong segment
 - Định hướng UDP segment tới socket tương ứng với số hiệu cổng đó
- Các IP datagram với **cùng số hiệu cổng đích**, nhưng có địa chỉ IP nguồn và/hoặc các số hiệu cổng nguồn khác nhau sẽ được định hướng tới **cùng socket** tại đích

Tầng giao vận 3-10

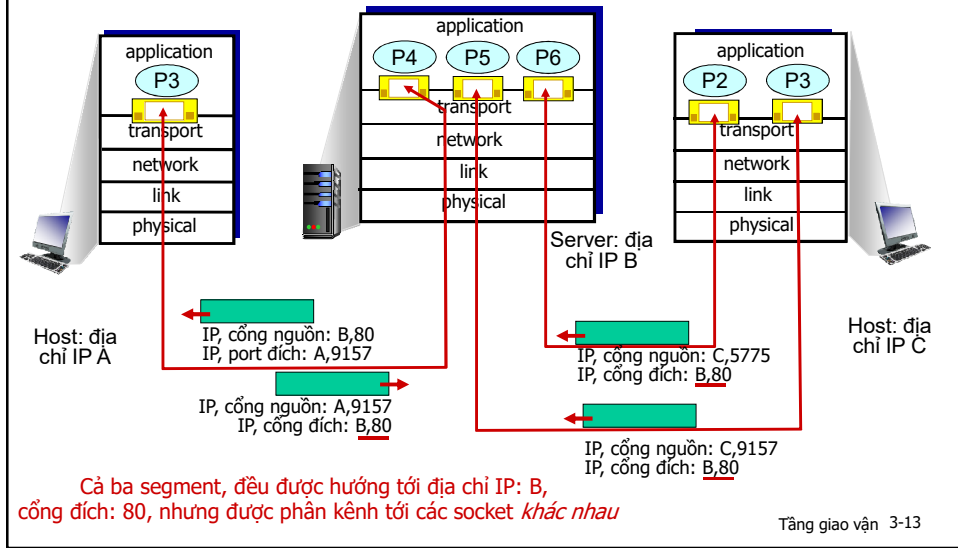
Ví dụ phân kênh hướng không kết nối



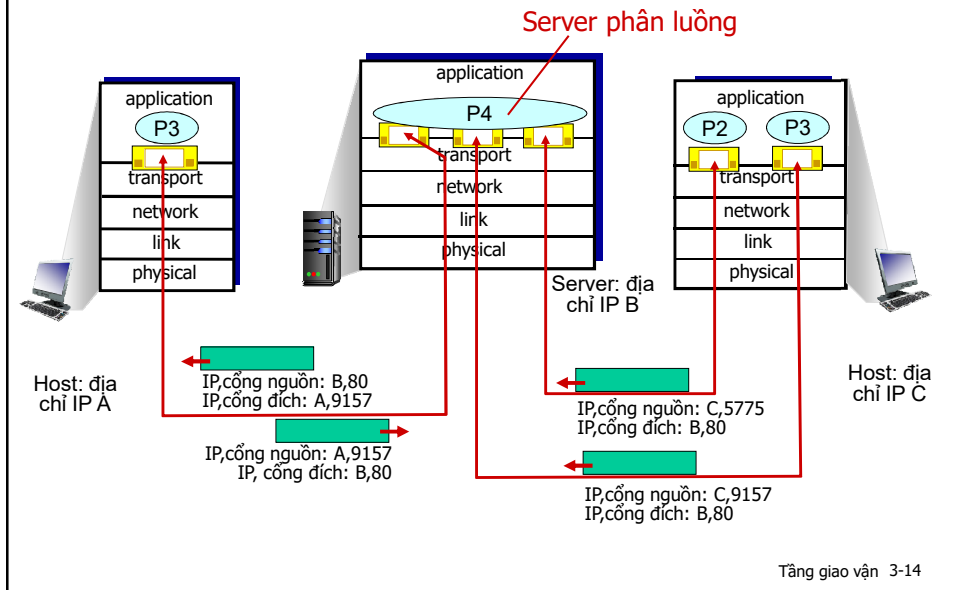
Phân kênh hướng kết nối

- ❖ TCP socket được xác định bởi bộ-4 giá trị:
 - Địa chỉ IP nguồn
 - Số hiệu cổng nguồn
 - Địa chỉ IP đích
 - Số hiệu cổng đích
- ❖ Phân kênh: Phía nhận sử dụng cả bốn giá trị này để định hướng segment tới socket phù hợp
- ❖ Host server có thể hỗ trợ nhiều TCP socket đồng thời:
 - Mỗi socket được xác định bởi bộ-4 giá trị của nó
- ❖ Web server có các socket khác nhau cho mỗi kết nối từ client
 - Kết nối HTTP không bền vững sẽ có các socket khác nhau cho mỗi yêu cầu.

Ví dụ phân kênh hướng kết nối



Ví dụ phân kênh hướng kết nối



Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)
- Truyền dữ liệu tin cậy
- Điều khiển luồng
- Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

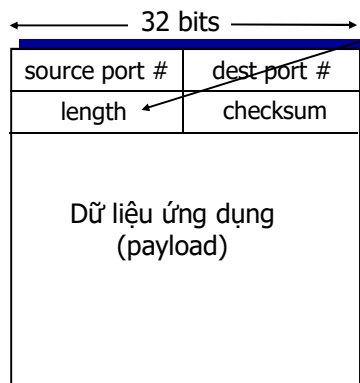
Tầng giao vận 3-15

UDP: User Datagram Protocol [RFC 768]

- ❖ Là giao thức tầng giao vận của mạng Internet
- ❖ Dịch vụ “best effort”, các UDP segment có thể:
 - Bị mất
 - Được vận chuyển không đúng thứ tự tới ứng dụng
- ❖ **Hướng không kết nối:**
 - Không có giai đoạn bắt tay giữa bên gửi và bên nhận của UDP
 - Mỗi UDP segment được xử lý độc lập với các segment khác
- ❖ UDP được dùng trong:
 - Các ứng dụng streaming multimedia (chịu mất mát dữ liệu, bị ảnh hưởng bởi tốc độ)
 - DNS
 - SNMP
- ❖ Truyền tin cậy trên UDP:
 - Bổ sung đặc tính tin cậy vào tầng ứng dụng
 - Khôi phục lỗi cụ thể của ứng dụng

Tầng giao vận 3-16

UDP: Tiêu đề segment



Định dạng UDP segment

Chiều dài, được tính theo số byte của UDP segment, bao gồm cả phần tiêu đề

Tại sao lại dùng UDP?

- ❖ Không cần thiết lập kết nối (vì việc này có thể làm tăng độ trễ)
- ❖ Đơn giản: không lưu trạng thái kết nối tại bên gửi, bên nhận
- ❖ Kích thước tiêu đề nhỏ
- ❖ Không điều khiển tắc nghẽn: UDP có thể gửi nhanh theo mong muốn

Tầng giao vận 3-17

UDP checksum

Mục tiêu: Phát hiện các “lỗi” (ví dụ: các bit bị bật lên) trong các segment được truyền đến

Bên gửi:

- ❖ Xử lý nội dung các đoạn, bao gồm cả các trường trong tiêu đề, như là chuỗi các số nguyên 16-bit
- ❖ checksum: bổ sung thêm (tổng bù của 1) vào nội dung segment
- ❖ Bên gửi đặt giá trị checksum vào trong trường checksum của UDP

Bên nhận:

- ❖ Tính toán checksum của segment đã nhận được
- ❖ Kiểm tra xem checksum đã tính có bằng giá trị của trường checksum hay không:
 - KHÔNG – phát hiện có lỗi
 - CÓ – không phát hiện lỗi. *Nhưng vẫn có thể có lỗi mà chưa được phát hiện? Xem thêm phần sau*

Tầng giao vận 3-18

Ví dụ: checksum trên Internet

Ví dụ: Cộng hai số nguyên 16-bit

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
<hr/>																
Bit dư	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
<hr/>																
Tổng	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

Chú ý: Khi cộng các số nguyên, một bit nhớ ở phía cao nhất cần phải được thêm vào kết quả

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)
- Truyền dữ liệu tin cậy
- Điều khiển luồng
- Quản lý kết nối

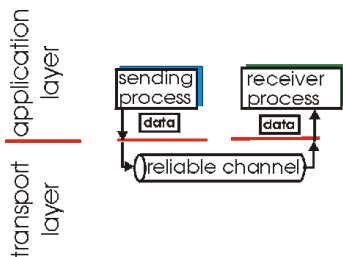
3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-20

Các nguyên lý của truyền dữ liệu tin cậy

- ❖ Quan trọng trong các tầng ứng dụng, giao vận và liên kết
 - Thuộc danh sách 10 vấn đề quan trọng nhất của mạng!



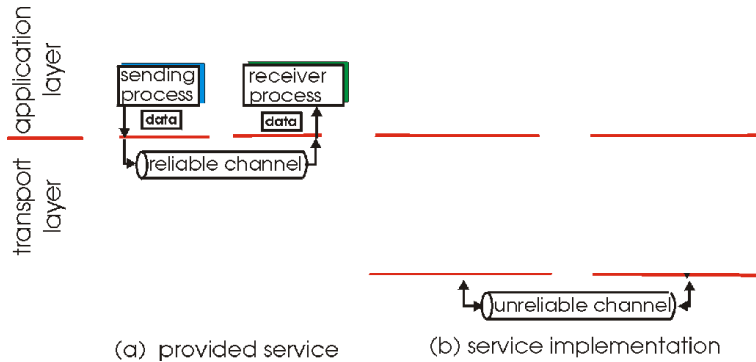
(a) provided service

- ❖ Các đặc tính của kênh truyền không tin cậy sẽ xác định sự phức tạp của giao thức truyền dữ liệu tin cậy (reliable data transfer protocol – rdt)

Tầng giao vận 3-21

Các nguyên lý của truyền dữ liệu tin cậy

- ❖ Quan trọng trong các tầng ứng dụng, giao vận và liên kết
 - Thuộc danh sách 10 vấn đề quan trọng nhất của mạng!

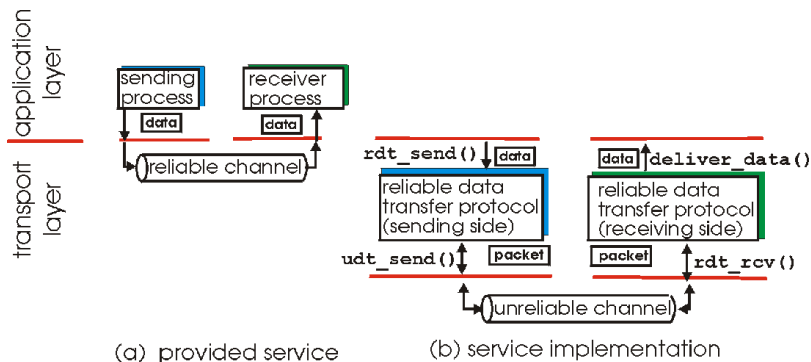


- ❖ Các đặc tính của kênh truyền không tin cậy sẽ xác định sự phức tạp của giao thức truyền dữ liệu tin cậy (reliable data transfer protocol – rdt)

Tầng giao vận 3-22

Các nguyên lý của truyền dữ liệu tin cậy

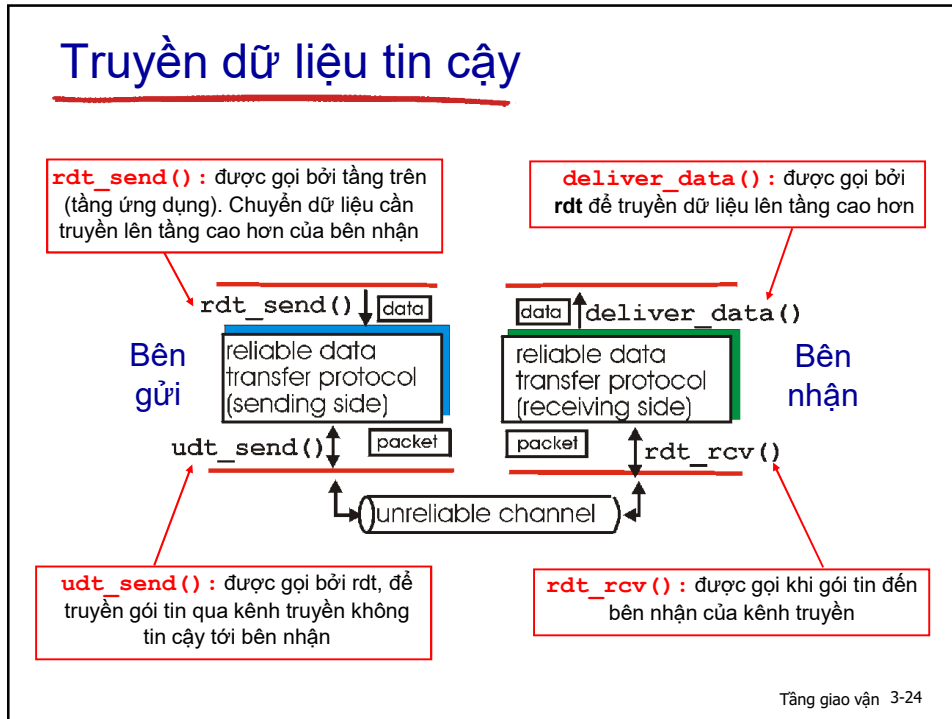
- ❖ Quan trọng trong các tầng ứng dụng, giao vận và liên kết
 - Thuộc danh sách 10 vấn đề quan trọng nhất của mạng!



- ❖ Các đặc tính của kênh truyền không tin cậy sẽ xác định sự phức tạp của giao thức truyền dữ liệu tin cậy (reliable data transfer protocol – rdt)

Tầng giao vận 3-23

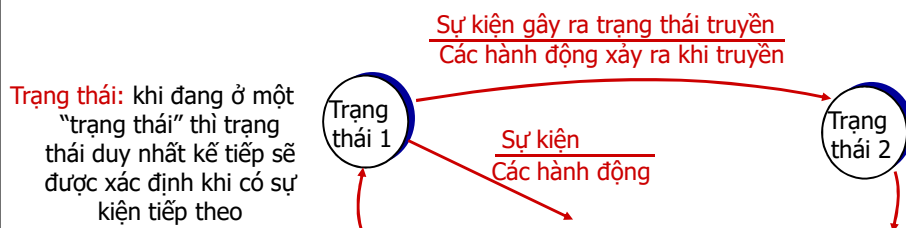
Truyền dữ liệu tin cậy



Truyền dữ liệu tin cậy

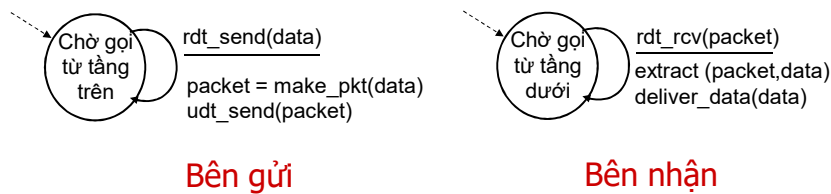
Việc cần làm:

- ❖ Phát triển dần giao thức truyền dữ liệu tin cậy (reliable data transfer protocol - rdt) cho cả bên gửi và bên nhận
- ❖ Chỉ xem xét truyền dữ liệu theo một hướng
 - Nhưng thông tin điều khiển vẫn được truyền theo cả hai hướng
- ❖ Dùng máy trạng thái hữu hạn (finite state machines - FSM) để xác định bên gửi, bên nhận



rdt1.0: truyền dữ liệu tin cậy qua một kênh truyền tin cậy

- ❖ Kênh truyền cơ bản hoàn toàn tin cậy
 - Không có lỗi bit
 - Không có mất mát gói tin
- ❖ Phân biệt các FSM cho bên gửi, bên nhận:
 - Bên gửi gửi dữ liệu vào kênh truyền cơ bản
 - Bên nhận đọc dữ liệu từ kênh truyền cơ bản



Tầng giao vận 3-26

rdt2.0: Kênh truyền có lỗi bit

- ❖ Kênh cơ bản có thể bật một vài bit trong gói tin
 - Kiểm tra (checksum) để phát hiện các lỗi bit
- ❖ Câu hỏi: Làm thế nào để khôi phục lại các lỗi?

Làm thế nào con người khôi phục được "lỗi" trong suốt quá trình thực hiện cuộc hội thoại?

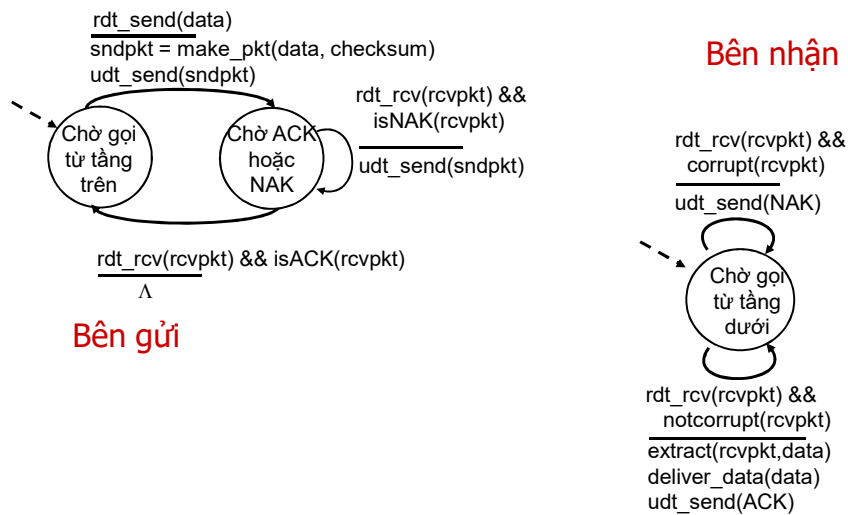
Tầng giao vận 3-27

rdt2.0: Kênh truyền có lỗi bit

- ❖ Kênh truyền cơ bản có thể bật một vài bit trong gói tin
 - Kiểm tra (checksum) để phát hiện các lỗi bit
- ❖ Câu hỏi: Làm thế nào để khôi phục lại các lỗi?
 - **Báo nhận ACK (acknowledgement):** bên nhận thông báo rõ cho bên gửi là gói tin nhận được tốt
 - **Báo nhận NAK (negative acknowledgement):** bên nhận thông báo rõ cho bên gửi là gói tin nhận được có lỗi
 - Bên gửi truyền lại gói tin có báo nhận là NAK
- ❖ Các cơ chế mới trong rdt2.0 (ngoài rdt1.0):
 - Phát hiện lỗi
 - Phản hồi: các thông điệp điều khiển (ACK,NAK) từ bên nhận gửi về bên gửi

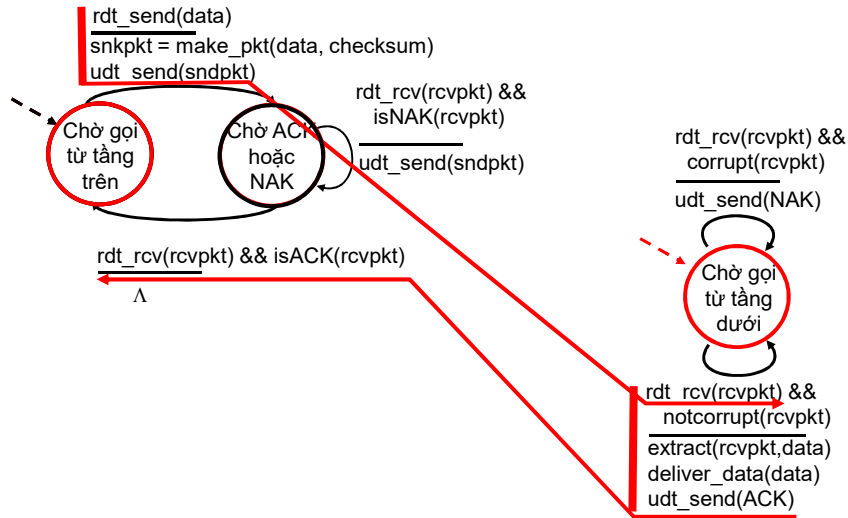
Tầng giao vận 3-28

rdt2.0: Đặc tả FSM



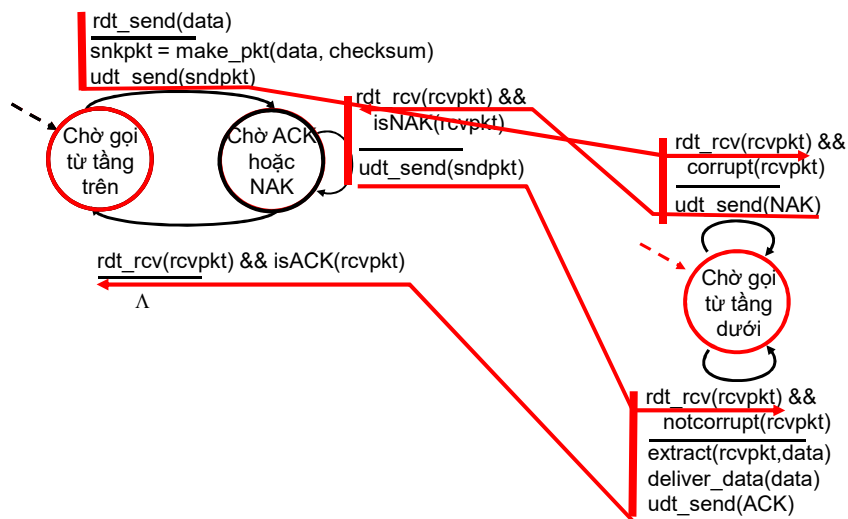
Tầng giao vận 3-29

rdt2.0: Hoạt động khi không có lỗi



Tầng giao vận 3-30

rdt2.0: Kịch bản khi có lỗi



Tầng giao vận 3-31

rdt2.0 có lỗi hỏng nghiêm trọng!

Điều gì xảy ra khi ACK/NAK bị hỏng?

- ❖ Bên gửi không biết được điều gì đã xảy ra tại bên nhận!
- ❖ Không thể đơn phương truyền lại: có thể bị trùng lặp

Xử lý trùng lặp:

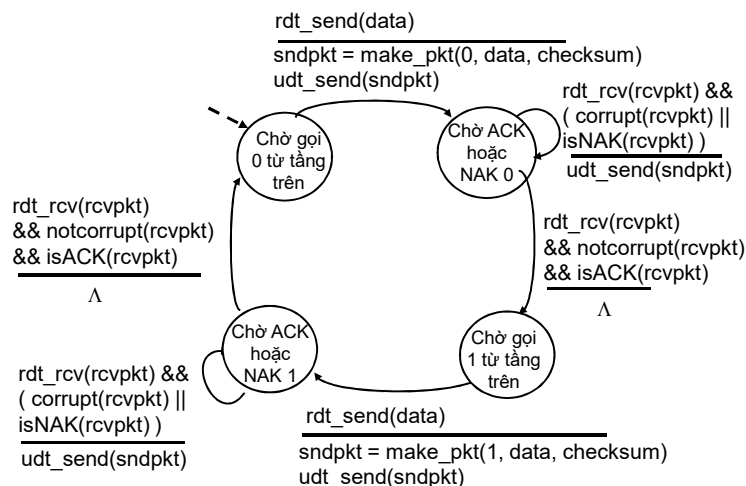
- ❖ Bên gửi truyền lại gói tin hiện tại nếu ACK/NAK bị hỏng
- ❖ Bên gửi thêm *số thứ tự* vào trong mỗi gói tin
- ❖ Bên nhận bỏ qua (không nhận) gói bị trùng lặp

Dừng và chờ

Bên gửi gửi một gói tin,
sau đó dừng lại chờ bên
nhận phản hồi

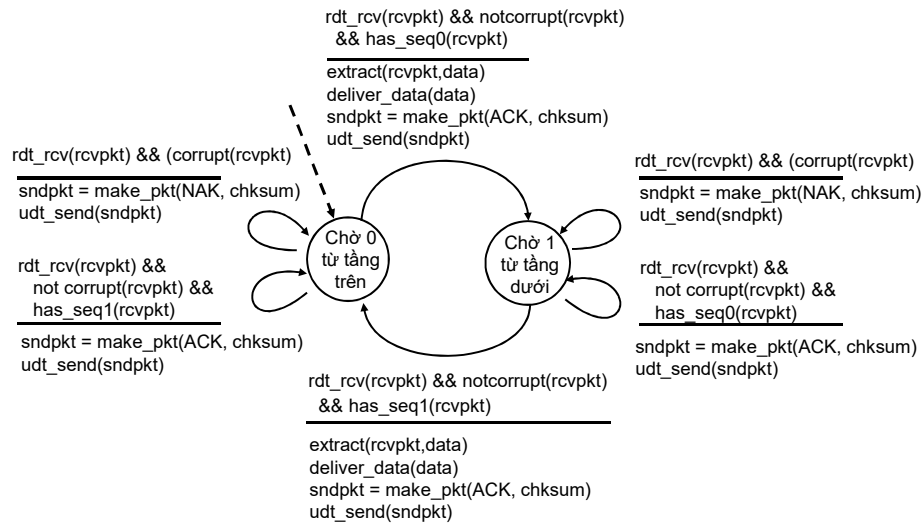
Tầng giao vận 3-32

rdt2.1: Bên gửi xử lý các ACK/NAK bị hỏng



Tầng giao vận 3-33

rdt2.1: Bên nhận xử lý các ACK/NAK bị hỏng



Tầng giao vận 3-34

rdt2.1: Thảo luận

Bên gửi:

- ❖ Số thứ tự được bổ sung vào gói tin
- ❖ Chỉ cần hai số thứ tự (0,1) là đủ. Vì sao?
- ❖ Phải kiểm tra lại nếu việc nhận ACK/NAK bị hỏng
- ❖ Số trạng thái tăng lên 2 lần
 - Trạng thái phải “nhỏ” xem gói tin đang “dự kiến” đến sẽ có số thứ tự là 0 hay 1

Bên nhận:

- ❖ Phải kiểm tra xem gói tin nhận được có bị trùng lặp hay không
 - Trạng thái chỉ rõ gói tin đang chờ đến có số thứ tự là 0 hay 1
- ❖ Chú ý: bên nhận *không thể* biết được ACK/NAK cuối cùng gửi đi có được nhận tốt hay không tại bên gửi

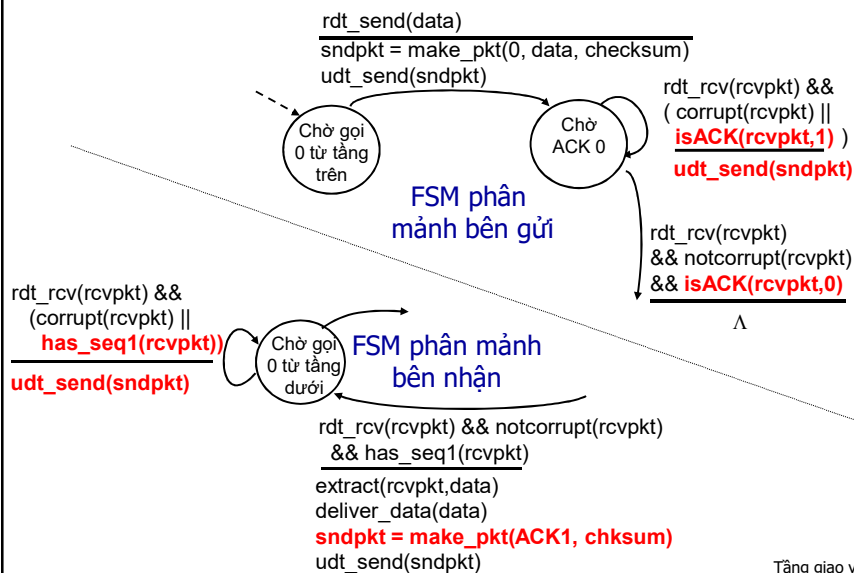
Tầng giao vận 3-35

rdt2.2: Một giao thức không cần NAK

- ❖ Chức năng giống như trong rdt2.1, nhưng chỉ dùng báo nhận ACK
- ❖ Thay vì sử dụng NAK, bên nhận sẽ gửi ACK cho gói tin cuối cùng nhận tốt
 - Bên nhận phải thêm số thứ tự của gói tin đang được báo nhận
- ❖ ACK bị trùng lặp tại bên gửi sẽ dẫn đến cùng hành động như NAK: *truyền lại gói tin hiện tại*

Tầng giao vận 3-36

rdt2.2: Phân mảnh tại bên gửi, bên nhận



Tầng giao vận 3-37

rdt3.0: Kênh truyền có lỗi và mất mát

Giải thiết mới: Kênh cơ bản cũng có thể làm mất các gói tin (dữ liệu, ACK)

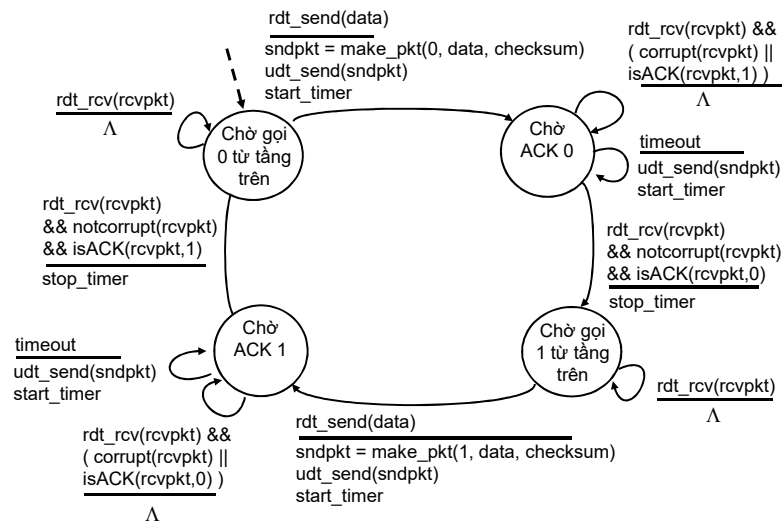
- checksum, số thứ tự, báo nhận ACK, truyền lại sẽ hỗ trợ... nhưng chưa đủ

Tiếp cận: Bên gửi chờ ACK trong khoảng thời gian “chấp nhận được”

- ❖ Truyền lại nếu không nhận được ACK trong khoảng thời gian này
- ❖ Nếu gói tin (hoặc ACK) chỉ đến trễ (chứ không bị mất):
 - Việc truyền lại sẽ gây trùng lặp, nhưng số thứ tự sẽ xử lý việc này
 - Bên nhận phải chỉ rõ số thứ tự của gói tin đang được báo nhận
- ❖ Cần bộ định thời đếm ngược

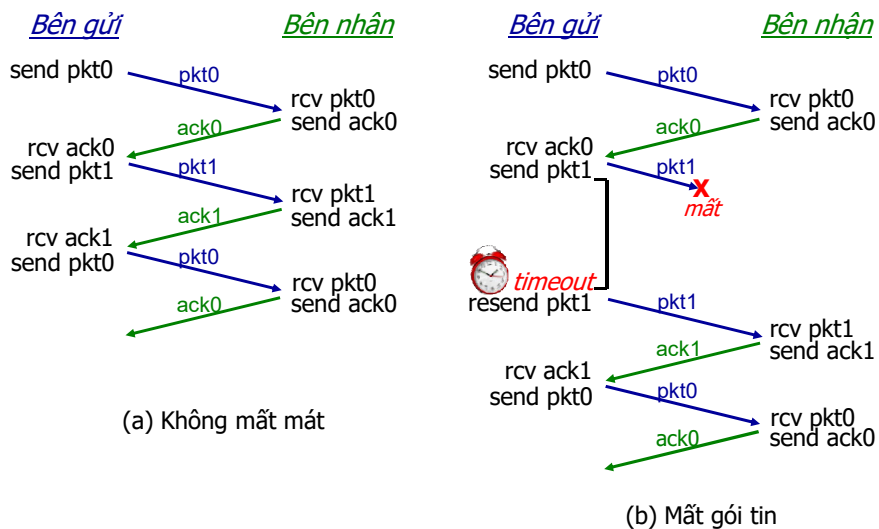
Tầng giao vận 3-38

rdt3.0 bên gửi



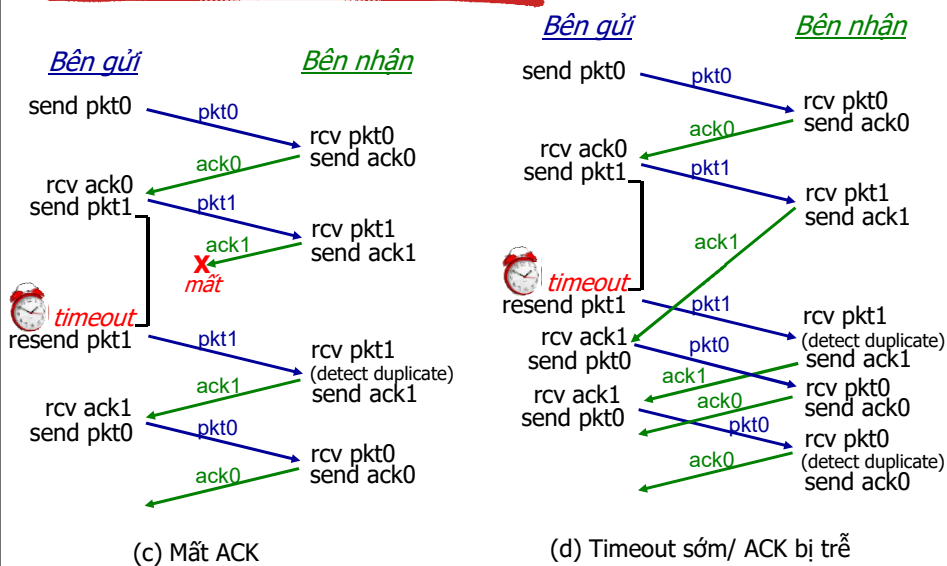
Tầng giao vận 3-39

Hoạt động của rdt3.0



Tầng giao vận 3-40

Hoạt động của rdt3.0



Tầng giao vận 3-41

Hiệu suất của rdt3.0

- ❖ rdt3.0 hoạt động tốt, nhưng không hiệu quả
- ❖ Ví dụ: Liên kết 1 Gbps, trễ lan truyền 15 ms, gói tin 8000 bit:

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

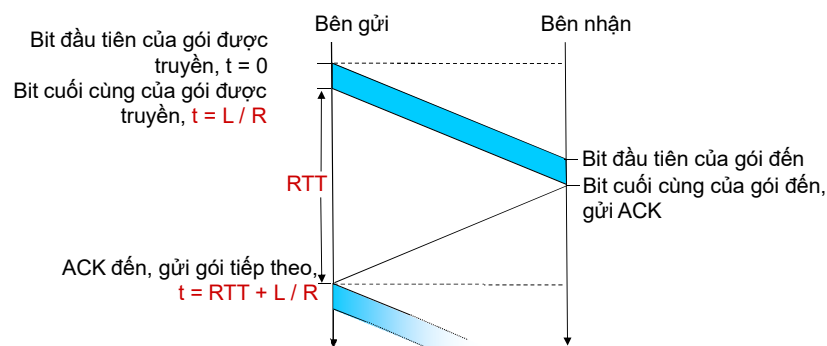
- U_{sender} : **độ khả dụng** – tỷ lệ về mặt thời gian bên gửi liên tục phải gửi

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

- Nếu $RTT=30$ msec, gói tin 1KB được truyền sau mỗi 30 msec: thông lượng trên liên kết 1 Gbps là 33kB/sec
- ❖ Giao thức mạng giới hạn việc sử dụng các tài nguyên vật lý!

Tăng giao vận 3-42

rdt3.0: Hoạt động dừng-và-chờ



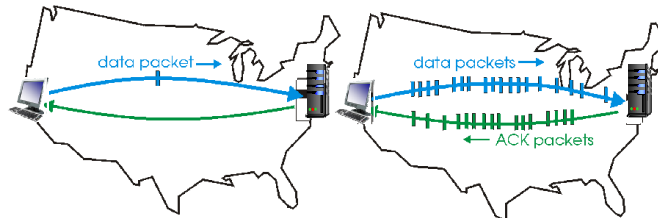
$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

Tăng giao vận 3-43

Các giao thức Pipeline

Pipelining: bên gửi cho phép gửi nhiều gói “đồng thời”,
mà không cần chờ gói báo nhận

- Dãy các số thứ tự sẽ được tăng dần
- Cần có bộ đệm tại bên gửi và/hoặc bên nhận



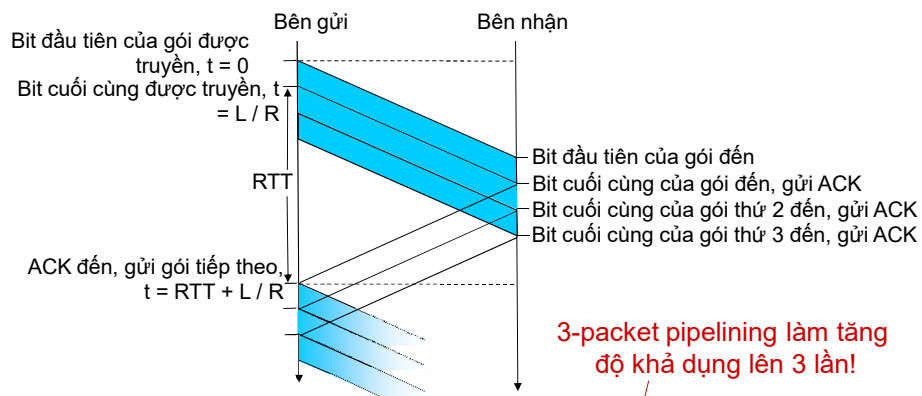
(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

❖ Hai dạng thức chung của các giao thức pipeline : **go-Back-N, lặp có lựa chọn (selective repeat)**

Tăng giao vận 3-44

Pipelining: tăng độ khả dụng



$$U_{\text{sender}} = \frac{3L/R}{RTT + L/R} = \frac{.0024}{30.008} = 0.00081$$

Tăng giao vận 3-45

Các giao thức pipeline

Go-back-N:

- ❖ Bên gửi có thể có đến N gói chưa được báo nhận trong pipeline
- ❖ Bên nhận chỉ gửi *ack tích lũy*
 - Không báo nhận cho gói tin cho đến khi có một khoảng trống
- ❖ Bên gửi có bộ định thời cho các gói tin gửi đi mà chưa được báo nhận
 - Khi bộ định thời hết hạn, truyền lại tất cả các gói tin chưa được báo nhận

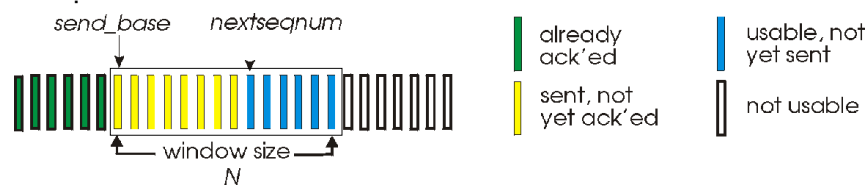
Lắp có lựa chọn:

- ❖ Bên gửi có thể có đến N gói chưa được báo nhận trong pipeline
- ❖ Bên nhận gửi *ack riêng* cho mỗi gói tin
- ❖ Bên gửi duy trì bộ định thời cho mỗi gói tin chưa được báo nhận
 - Khi bộ định thời hết hạn, chỉ truyền lại gói tin chưa được báo nhận

Tầng giao vận 3-46

Go-Back-N: bên gửi

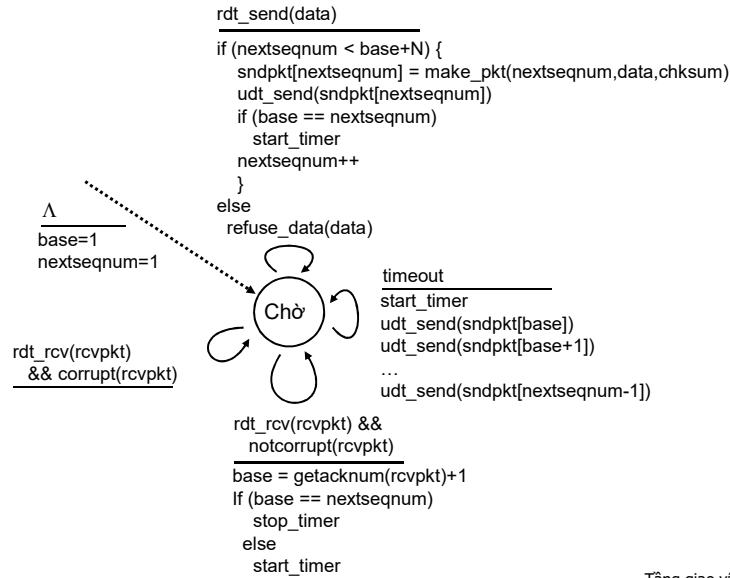
- ❖ k-bit số thứ tự trong phần tiêu đề của gói tin
- ❖ “Cửa sổ” tăng lên đến N, cho phép gửi gói liên tục không cần báo nhận



- ❖ ACK(n): báo nhận ACK cho tất cả các gói đến, chứa số thứ tự n-
“*ACK tích lũy*”
 - Có thể nhận được ACK trùng lặp (xem bên nhận)
- ❖ Đặt bộ định thời cho các gói tin truyền đi
- ❖ *timeout(n)*: truyền lại gói n và tất cả các gói có số thứ tự lớn hơn trong cửa sổ

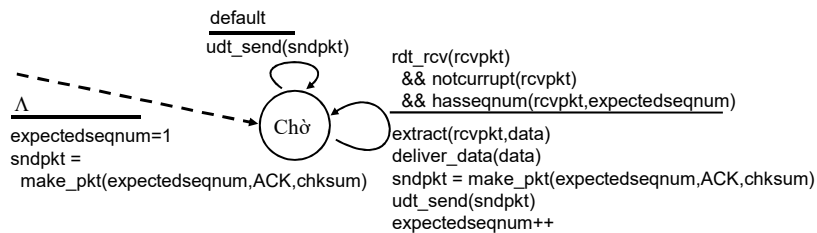
Tầng giao vận 3-47

GBN: FSM mở rộng tại bên gửi



Tăng giao vận 3-48

GBN: FSM mở rộng tại bên nhận



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

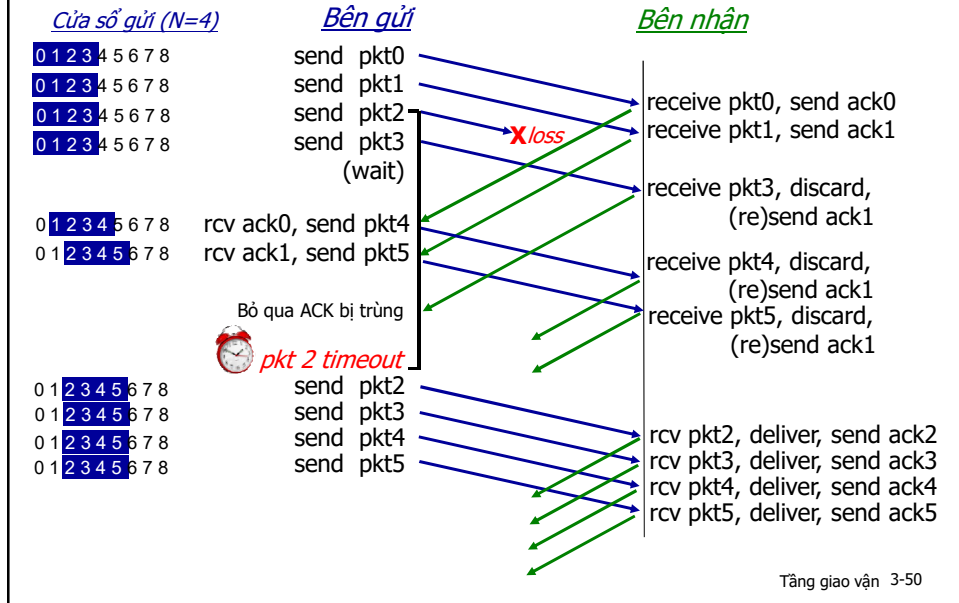
- Có thể sinh ra ACK trùng nhau
- Chỉ cần nhớ số thứ tự của gói dự kiến đến (**expectedseqnum**)

❖ Gói không theo đúng thứ tự:

- Hủy: **không nhận vào vùng đệm!**
- Gửi lại ACK với số thứ tự (xếp hạng) cao nhất

Tăng giao vận 3-49

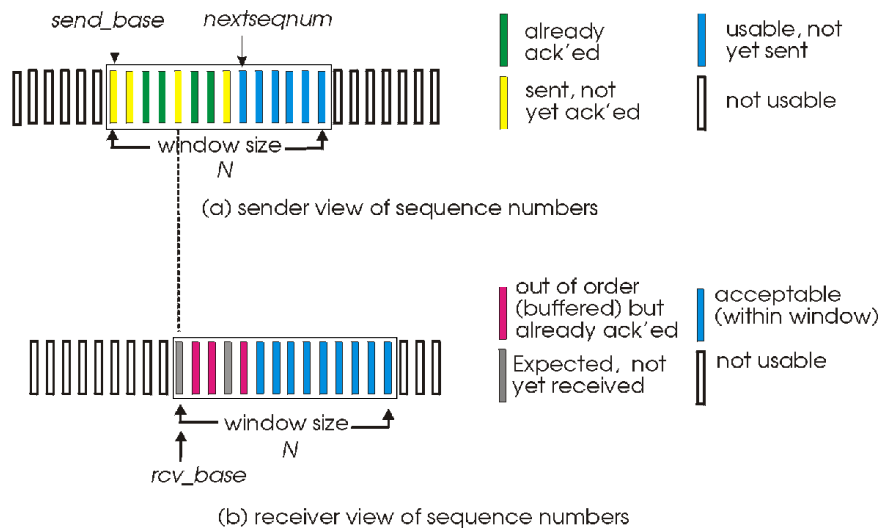
Hoạt động của GBN



Lặp có lựa chọn

- ❖ Bên nhận báo nhận *riêng* cho tất cả các gói tin đã nhận đúng.
 - Đặt các gói vào bộ đệm (nếu cần), cho đúng thứ tự để chuyển lên tầng cao hơn
- ❖ Bên gửi chỉ gửi lại các gói tin nào mà không nhận được ACK
 - Có bộ định thời bên gửi cho mỗi gói tin không gửi ACK
- ❖ Cửa sổ bên gửi
 - N số thứ tự liên tục
 - Hạn chế số thứ tự các gói không gửi ACK

Lựa chọn có lựa chọn: cửa sổ bên gửi, bên nhận



Tầng giao vận 3-52

Lựa chọn có lựa chọn

Bên gửi

Dữ liệu từ tầng trên:

- ❖ Nếu số thứ tự kế tiếp sẵn sàng trong cửa sổ, thì gửi gói tin

timeout(n):

- ❖ Gửi lại gói n, khởi tạo lại bộ định thời

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

- ❖ Đánh dấu gói n là đã nhận
- ❖ Nếu gói có số thứ tự n thấp nhất mà chưa được ACK, thì dịch chuyển cửa sổ cơ sở đến số thứ tự kế tiếp chưa được ACK.

Bên nhận

Gói n trong [rcvbase, rcvbase+N-1]

- ❖ Gửi ACK(n)
- ❖ Không đúng thứ tự: đệm
- ❖ Đúng thứ tự: truyền (cũng truyền các gói đã đệm, đúng thứ tự), dịch chuyển cửa sổ đến gói chưa nhận được kế tiếp

Gói n trong [rcvbase-N, rcvbase-1]

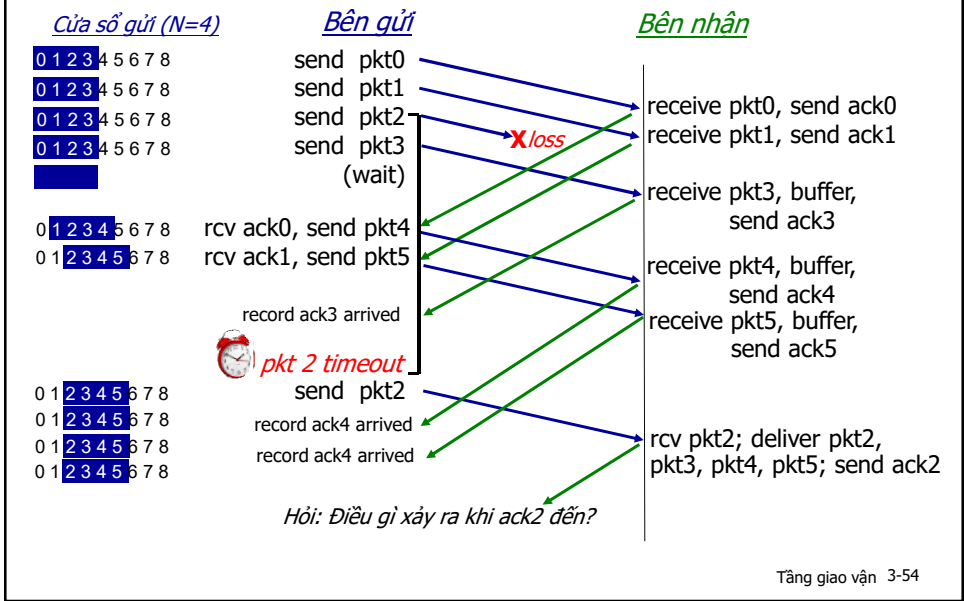
- ❖ ACK(n)

Ngược lại:

- ❖ Bỏ qua

Tầng giao vận 3-53

Hoạt động trong lập có lựa chọn

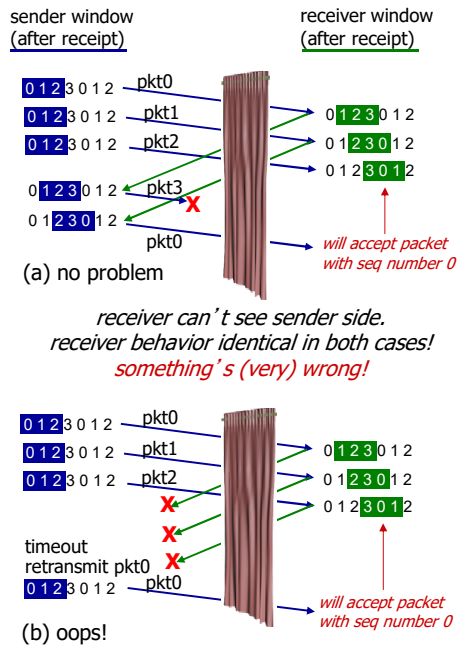


Lập có lựa chọn: tình trạng khó giải quyết

Ví dụ:

- ❖ Các số thứ tự: 0, 1, 2, 3
- ❖ Kích thước cửa sổ = 3
- ❖ Bên nhận không nhận ra sự khác biệt giữa 2 kịch bản!
- ❖ Chấp nhận dữ liệu bị trùng lặp như là dữ liệu mới trong (b)

Hỏi: Quan hệ giữa kích thước số thứ tự và kích thước cửa sổ như thế nào để tránh vấn đề như trong (b)?



Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)

- Truyền dữ liệu tin cậy

- Điều khiển luồng

- Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-56

Khái quát TCP RFCs: 793,1122,1323, 2018, 2581

- ❖ **Điểm-tới-điểm:**

- Một bên gửi, một bên nhận

- ❖ **Truyền *dòng byte* theo đúng thứ tự và truyền tin cậy:**

- Không có “ranh giới thông điệp”

- ❖ **pipeline:**

- Điều khiển tắc nghẽn và điều khiển luồng TCP thiết lập kích thước cửa sổ

- ❖ **Truyền dữ liệu song công (full duplex):**

- Luồng dữ liệu đi theo 2 hướng trên cùng một kết nối

- MSS: maximum segment size (kích thước đoạn lớn nhất)

- ❖ **Hướng kết nối:**

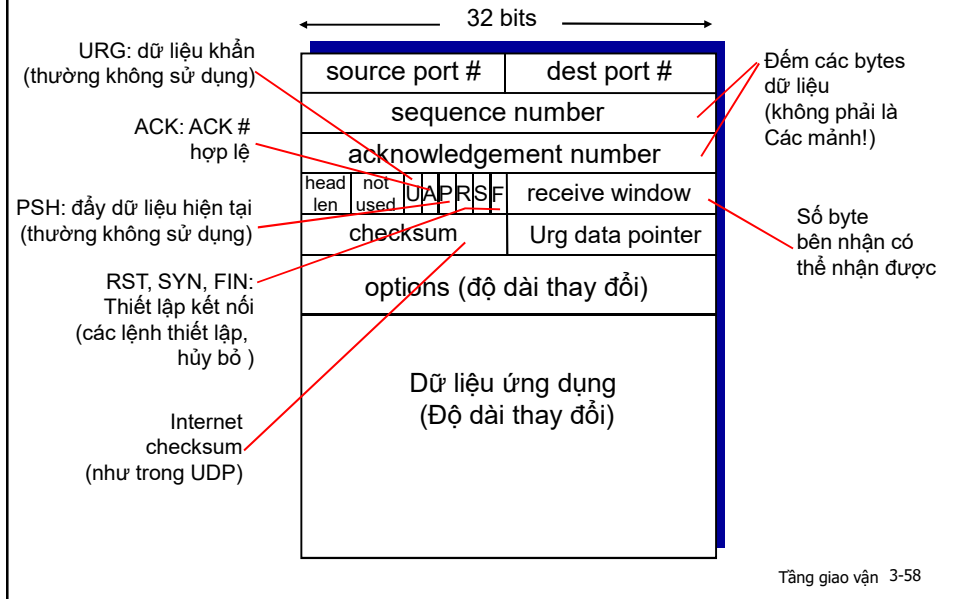
- Bất tay (trao đổi các thông điệp điều khiển) khởi tạo trạng thái cho bên gửi và bên nhận trước khi trao đổi dữ liệu

- ❖ **Điều khiển luồng:**

- Bên gửi không lấn át bên nhận

Tầng giao vận 3-57

Cấu trúc TCP segment



Số thứ tự và báo nhận ACK trong TCP

Số thứ tự:

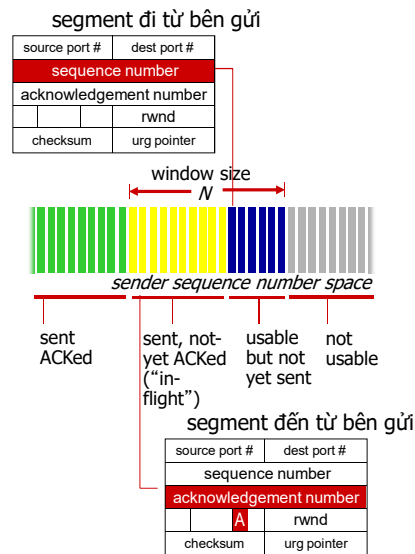
- “Số” dòng byte của byte đầu tiên trong đoạn (segment) dữ liệu

Báo nhận:

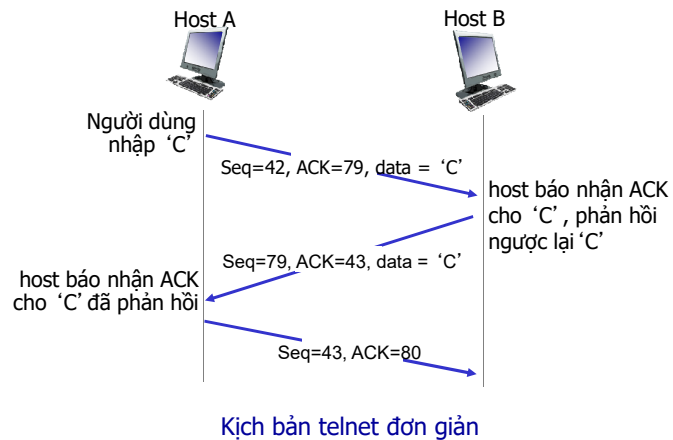
- Số thứ tự của byte tiếp theo được mong đợi từ phía bên kia
- ACK tích lũy

Hỏi: Làm thế nào bên nhận xử lý được các segment không đúng thứ tự?

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



Số thứ tự và báo nhận ACK trong TCP



Tầng giao vận 3-60

TCP round trip time và timeout

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

- ❖ Dài hơn RTT
 - nhưng RTT thay đổi
- ❖ *Quá ngắn:* timeout sớm, không cần truyền lại
- ❖ *Quá dài:* phản ứng chậm với các segment bị mất

Hỏi: Ước lượng RTT như nào?

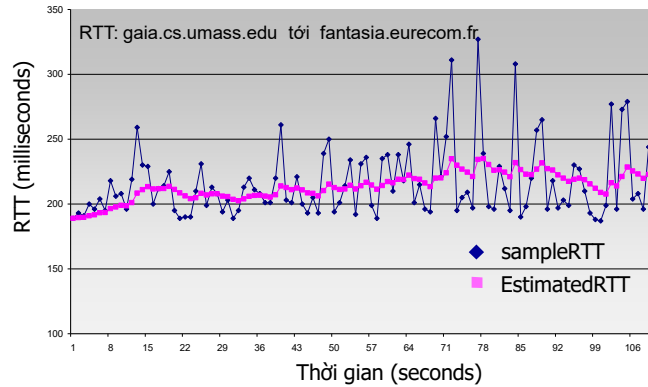
- ❖ **SampleRTT:** thời gian đo được từ khi truyền segment đến khi nhận được ACK
 - Bỏ qua việc truyền lại
- ❖ **SampleRTT** có thể thay đổi, cần giá trị RTT ước lượng “mượt hơn”
 - Tính trung bình một vài độ đo gần đây, không chỉ **SampleRTT** hiện tại

Tầng giao vận 3-61

TCP round trip time và timeout

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

- ❖ Giá trị đặc trưng: $\alpha = 0.125$



Tăng giao vận 3-62

TCP round trip time và timeout

- ❖ Khoảng thời gian timeout: **EstimatedRTT** cộng với “hệ số dự trữ an toàn”
 - Nếu có biến thiên lớn trong **EstimatedRTT**, thì hệ số dự trữ an toàn phải lớn hơn
- ❖ Ước lượng sự biến thiên của SampleRTT từ EstimatedRTT:

$$\text{DevRTT} = (1 - \beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$$

(Giá trị đặc trưng: $\beta = 0.25$)

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



↑ RTT ước lượng ↑ “hệ số dự trữ an toàn”

Tăng giao vận 3-63

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)

- Truyền dữ liệu tin cậy

- Điều khiển luồng

- Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-64

Truyền dữ liệu tin cậy trong TCP

❖ TCP tạo dịch vụ rdt trên dịch vụ không tin cậy của IP

- Truyền segment theo kiểu pipelining
- ACK tích lũy
- Dùng bộ định thời cho việc truyền lại

❖ Việc truyền lại được kích hoạt bởi:

- Các sự kiện timeout
- ACK bị trùng lặp

Hãy bắt đầu xem xét bên gửi TCP theo cách đơn giản:

- Bỏ qua trùng lặp ACK
- Bỏ qua điều khiển luồng, điều khiển tắc nghẽn

Tầng giao vận 3-65

Các sự kiện của TCP bên gửi:

Dữ liệu nhận từ ứng dụng: Timeout:

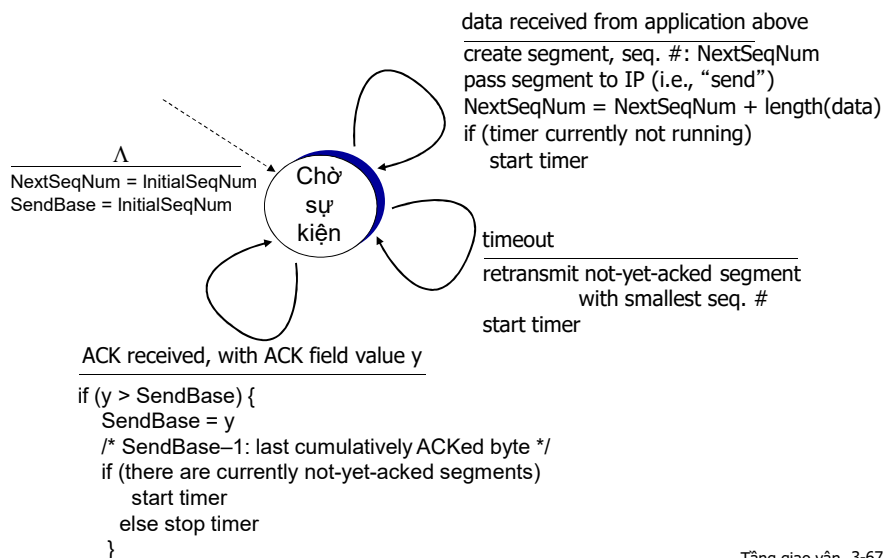
- ❖ Tạo segment với số thứ tự
- ❖ Số thứ tự là số dòng byte của byte dữ liệu đầu tiên trong segment
- ❖ Khởi tạo bộ định thời nếu chưa chạy:
 - Chú ý bộ định thời của segment chưa được báo nhận muộn nhất
 - Hết thời hạn: **TimeOutInterval**
- ❖ Truyền lại segment bị timeout
- ❖ Khởi tạo lại bộ định thời

ACK đã nhận:

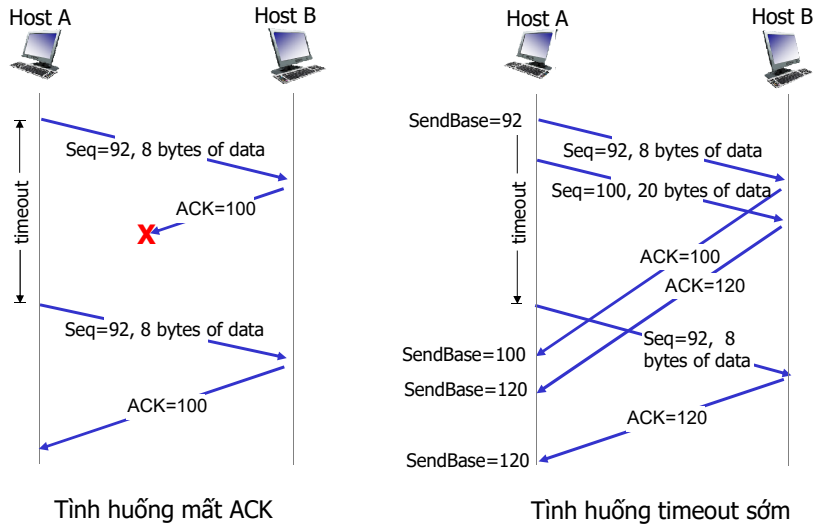
- ❖ Nếu ACK báo nhận cho các segment chưa được báo nhận trước đó, thì:
 - Cập nhật lại các segment đã được báo nhận
 - Khởi tạo bộ định thời nếu vẫn còn các segment chưa được báo nhận

Tầng giao vận 3-66

TCP bên gửi (đơn giản hóa)

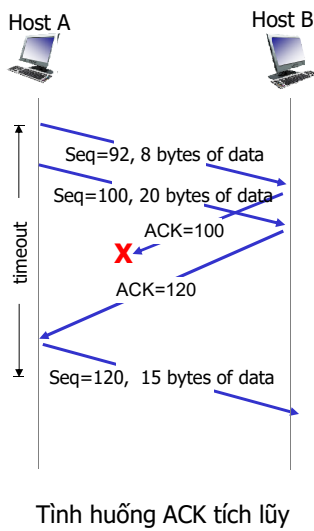


TCP: Các tình huống phải truyền lại



Tầng giao vận 3-68

TCP: Các tình huống phải truyền lại



Tầng giao vận 3-69

Tạo ACK trong TCP [RFC 1122, RFC 2581]

<i>Sự kiện tại bên nhận</i>	<i>Hành động của TCP tại bên nhận</i>
Segment đến đúng thứ tự với số thứ tự mong muốn. Tất cả dữ liệu đến đã được báo nhận	ACK bị trễ. Chờ 500ms cho segment tiếp theo. Nếu không có segment tiếp theo thì gửi ACK
Segment đến đúng thứ tự với số thứ tự mong muốn. Một segment khác đang chờ ACK	Gửi ngay một ACK tích lũy, báo nhận ACK cho cả hai segment đến đúng thứ tự
Segment đến không đúng số thứ tự, số thứ tự lớn hơn mong đợi. Phát hiện có khoảng trống	Gửi ngay ACK trùng lặp , chỉ ra số thứ tự của byte mong đợi tiếp theo
Segment đến lấp đầy hoặc một phần khoảng trống	Gửi ngay ACK, với điều kiện là segment bắt đầu ngay tại điểm có khoảng trống

Tầng giao vận 3-70

Truyền lại nhanh trong TCP

- ❖ Chu kỳ time-out thường tương đối dài:
 - Trễ dài trước khi gửi lại gói tin đã bị mất
- ❖ Phát hiện các segment bị mất qua các ACK bị trùng lặp.
 - Bên gửi thường gửi nhiều segment song song
 - Nếu segment, có thể sẽ có nhiều ACK bị trùng lặp.

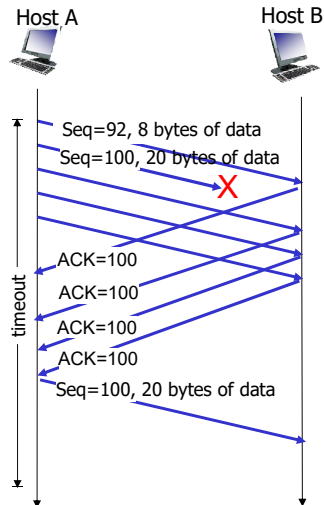
Truyền lại nhanh trong TCP

Nếu bên gửi nhận được 3 ACK trùng lặp cho cùng một dữ liệu (“Ba ACK trùng lặp”), thì sẽ gửi lại segment chưa được báo nhận có số thứ tự nhỏ nhất

- Có thể là đã bị mất segment chưa được báo nhận, nên không cần phải đợi đến timeout

Tầng giao vận 3-71

Truyền lại nhanh trong TCP



Truyền lại nhanh sau khi bên gửi nhận được 3 ACK trùng lặp

Tầng giao vận 3-72

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)

- Truyền dữ liệu tin cậy

- Điều khiển luồng

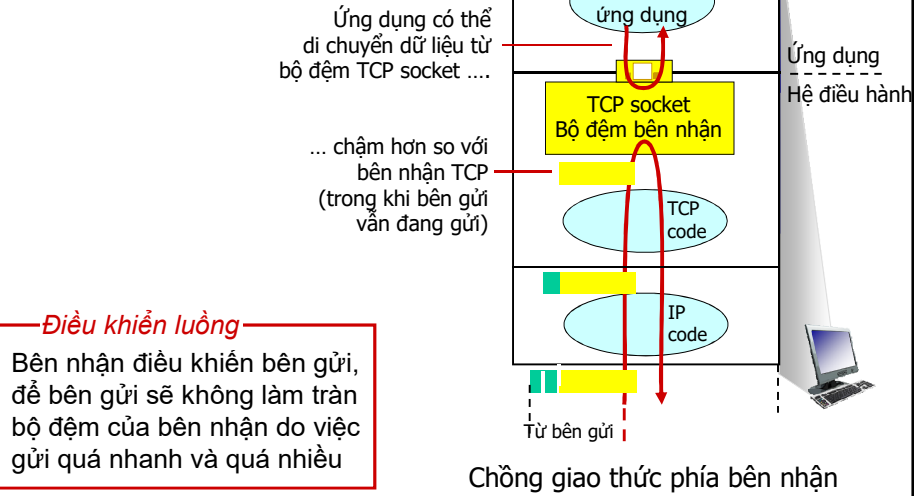
- Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

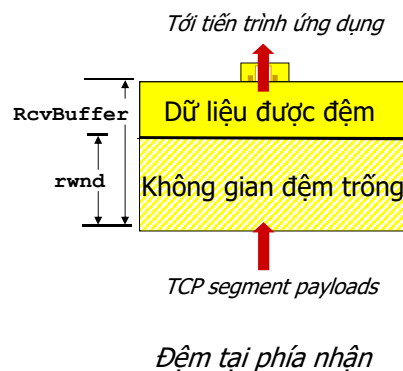
Tầng giao vận 3-73

Điều khiển luồng trong TCP



Điều khiển luồng trong TCP

- ❖ Bên nhận “thông báo” không gian đệm còn trống bởi giá trị **rwnd** trong TCP header của các segment gửi-nhận
 - Kích thước **RcvBuffer** được thiết lập qua tùy chọn (option) của socket (thường mặc định là 4096 byte)
 - Nhiều hệ điều hành tự động điều chỉnh **RcvBuffer**
- ❖ Bên gửi giới hạn tổng số dữ liệu chưa được báo nhận đến bên nhận theo giá trị **rwnd**
- ❖ Đảm bảo vùng đệm nhận không bị tràn



Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)
- Truyền dữ liệu tin cậy
- Điều khiển luồng
- Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

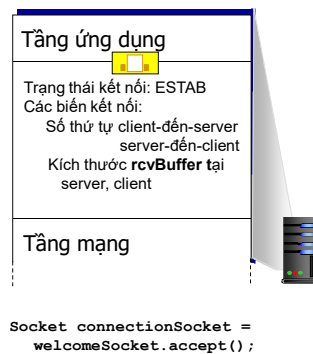
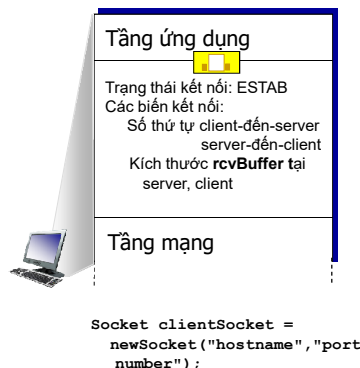
3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-76

Quản lý kết nối

Trước khi trao đổi dữ liệu, bên gửi/bên nhận “bắt tay”:

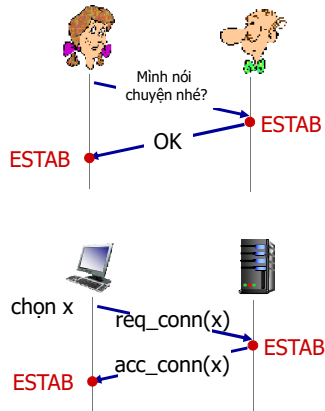
- ❖ Đồng ý thiết lập kết nối
- ❖ Đồng ý các tham số kết nối



Tầng giao vận 3-77

Đồng ý thiết lập kết nối

Bắt tay hai bước



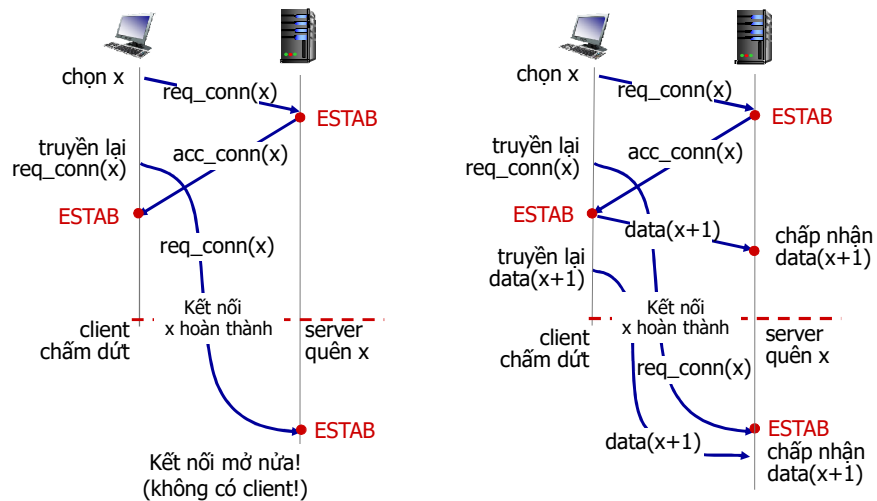
Hỏi: Bắt tay hai bước có luôn được thực hiện trên mạng?

- ❖ Chậm trễ liên quan tới các biến
- ❖ Các thông điệp được truyền lại (ví dụ, req_conn(x)) là do mất thông điệp
- ❖ Sắp xếp lại thông điệp
- ❖ Không thể “nhìn thấy” phía bên kia

Tầng giao vận 3-78

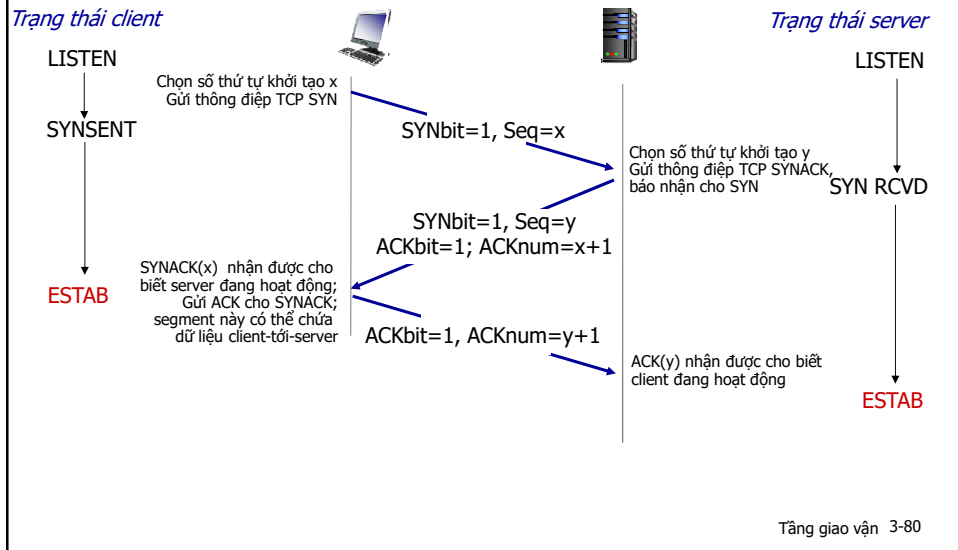
Đồng ý thiết lập kết nối

Các kịch bản có lỗi khi thực hiện bắt tay 2 bước:

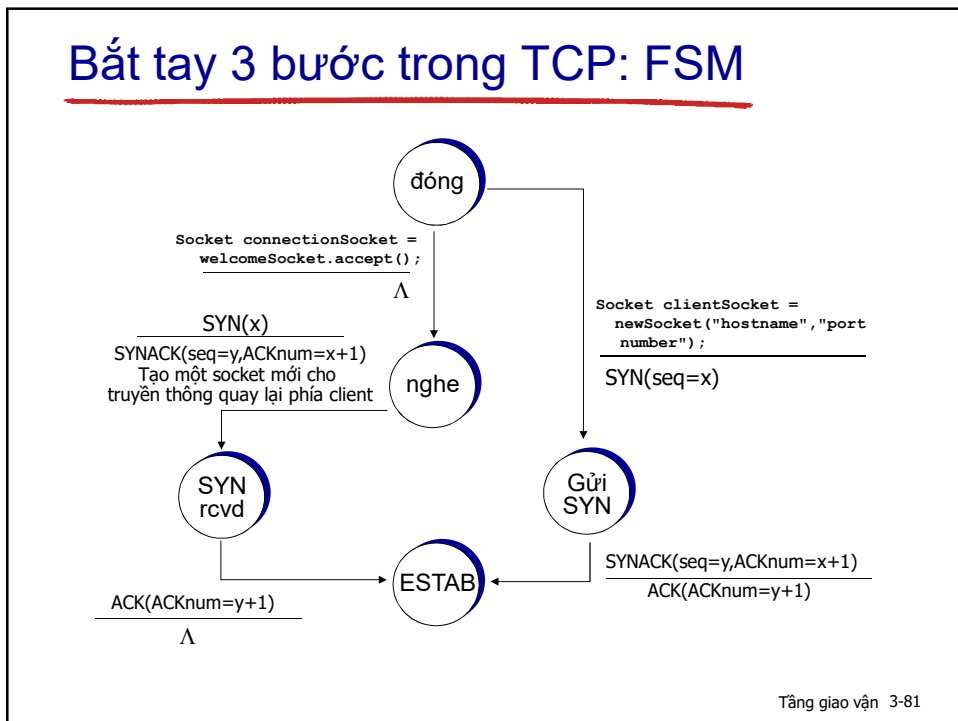


Tầng giao vận 3-79

Bắt tay 3 bước trong TCP



Bắt tay 3 bước trong TCP: FSM

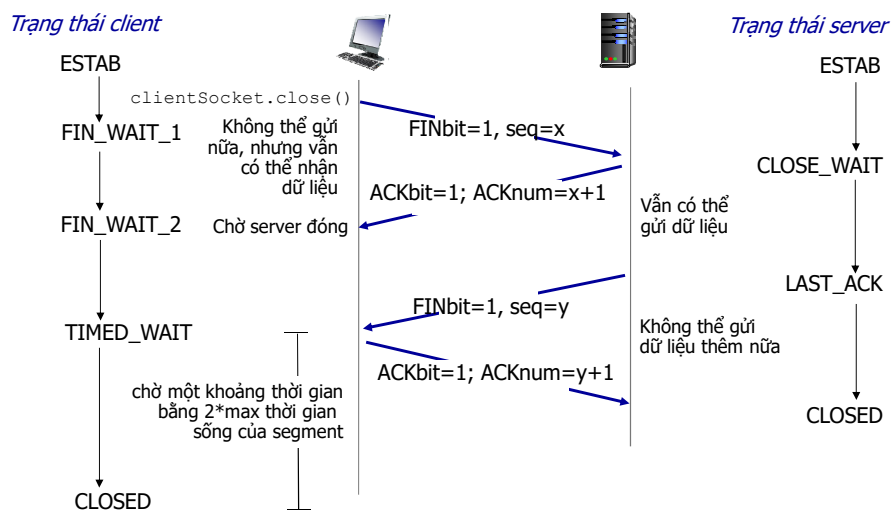


TCP: đóng kết nối

- ❖ Mỗi bên client và server thực hiện đóng kết nối
 - Gửi TCP segment với bit FIN = 1
- ❖ Đáp ứng lại FIN nhận được bằng ACK
 - FIN, ACK đang nhận có thể được kết nối với FIN của nó
- ❖ Có thể thực hiện đồng bộ trao đổi FIN

Tầng giao vận 3-82

TCP: đóng kết nối



Tầng giao vận 3-83

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)
- Truyền dữ liệu tin cậy
- Điều khiển luồng
- Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-84

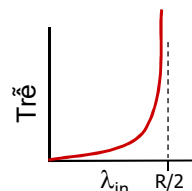
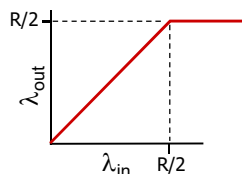
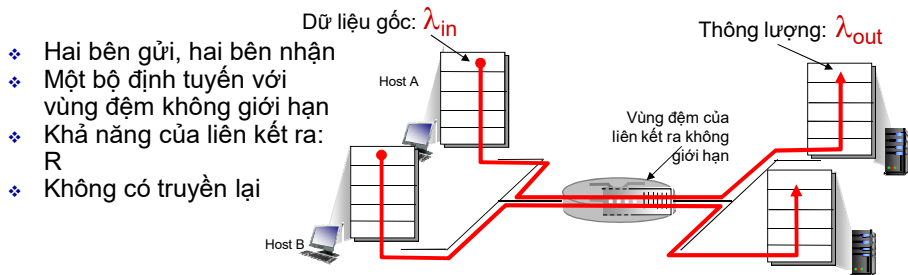
Các nguyên lý điều khiển tắc nghẽn

Tắc nghẽn:

- ❖ Có thể hiểu là: “quá nhiều nguồn cùng gửi quá nhiều dữ liệu với tốc độ quá nhanh tới *mạng*”
- ❖ Khác điều khiển luồng dữ liệu!
- ❖ Các biểu hiện chính:
 - Mất các gói tin (tràn bộ đệm tại các bộ định tuyến)
 - Trễ quá lâu (hàng đợi dài trong vùng đệm của bộ định tuyến)
- ❖ Là một trong mười vấn đề nan giải nhất của mạng!

Tầng giao vận 3-85

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

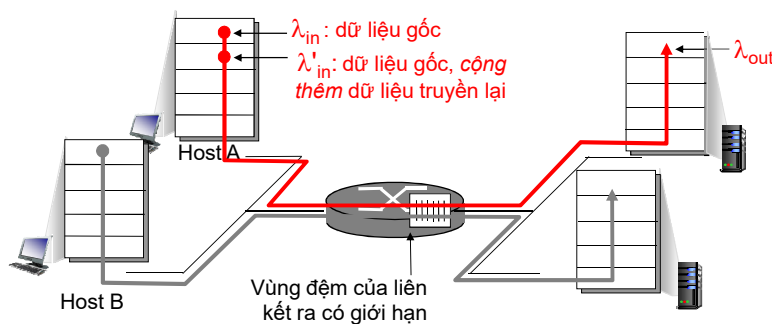


- ❖ Thông lượng lớn nhất trên mỗi kết nối: $R/2$
- ❖ Trễ lớn do tốc độ đến, λ_{in} , tiệm cận đến thông lượng tối đa

Tầng giao vận 3-86

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

- ❖ Một bộ định tuyến, vùng đệm có *giới hạn*
- ❖ Bên gửi truyền lại gói tin bị timeout
 - Đầu vào tầng ứng dụng = đầu ra tầng ứng dụng: $\lambda_{in} = \lambda_{out}$
 - Đầu vào tầng giao vận bao gồm việc truyền lại $\lambda'_{in} \geq \lambda_{in}$

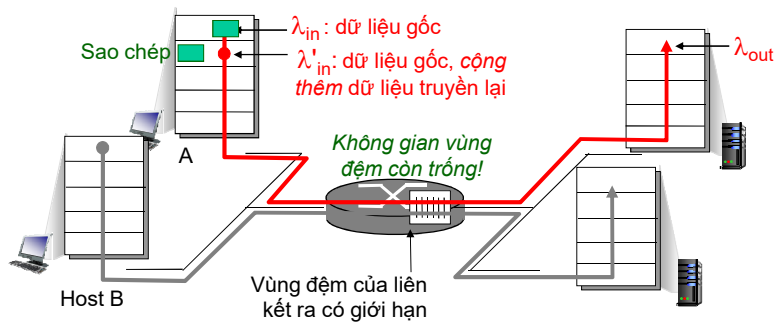
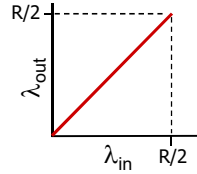


Tầng giao vận 3-87

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

Lý tưởng hóa: hiểu biết hoàn hảo

- ❖ Bên gửi chỉ gửi khi vùng đệm của bộ định tuyến sẵn sàng.

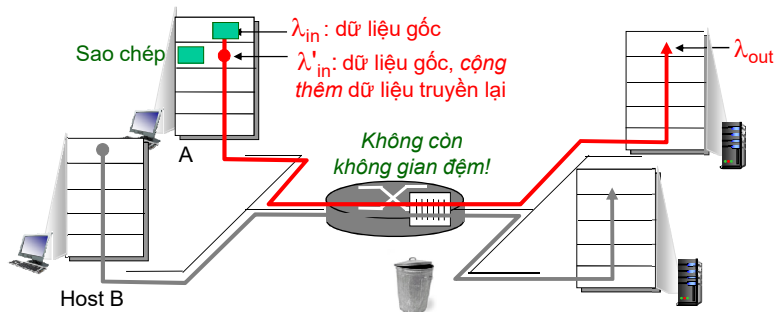


Tăng giao vận 3-88

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

Lý tưởng hóa: *biết về sự mất mát*

- ❖ Các gói tin có thể bị mất, bị bỏ rơi tại bộ định tuyến nếu vùng đệm của nó bị đầy
- ❖ Bên gửi chỉ gửi lại nếu gói tin được biết là đã bị mất

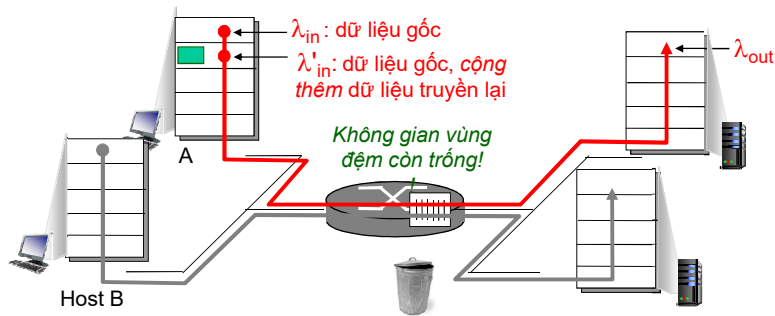
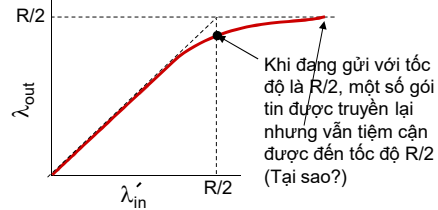


Tăng giao vận 3-89

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

Lý tưởng hóa: biết về sự mất mát

- ❖ Các gói tin có thể bị mất, bị bỏ rơi tại bộ định tuyến nếu vùng đệm của nó bị đầy
- ❖ Bên gửi chỉ gửi lại nếu gói tin được biết là đã bị mất

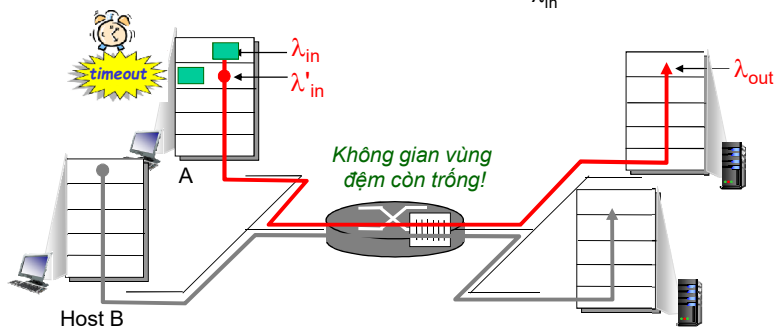
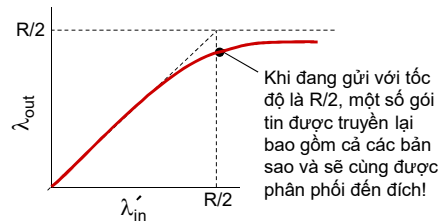


Tăng giao vận 3-90

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

Thực tế: các bản sao

- ❖ Các gói tin có thể bị mất, bị bỏ rơi tại bộ định tuyến nếu vùng đệm của nó bị đầy
- ❖ Nếu bên gửi timeout sớm, thì sẽ gửi đi hai bản sao của gói tin, và cả hai đều được phân phối đến đích

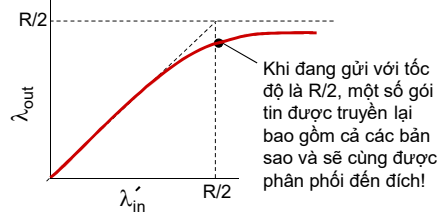


Tăng giao vận 3-91

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

Thực tế: các bản sao

- ❖ Các gói tin có thể bị mất, bị bỏ rơi tại bộ định tuyến nếu vùng đệm của nó bị đầy
- ❖ Nếu bên gửi timeout sớm, thì sẽ gửi đi *hai* bản sao của gói tin, và cả hai đều được phân phối đến đích



“Chi phí” của tắc nghẽn:

- ❖ Nhiều việc (truyền lại), với lưu lượng xác định
- ❖ Không cần thiết phải truyền lại: liên kết mang nhiều bản sao của gói tin
 - Làm giảm lưu lượng

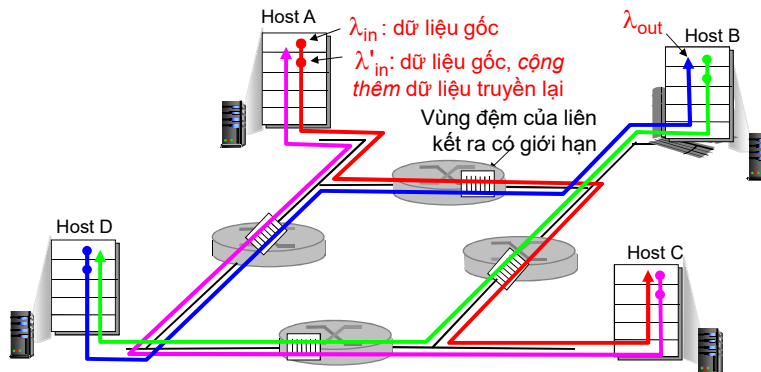
Tăng giao vận 3-92

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

- ❖ Bốn bên gửi
- ❖ Nhiều đường đến đích
- ❖ timeout/truyền lại

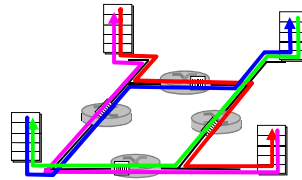
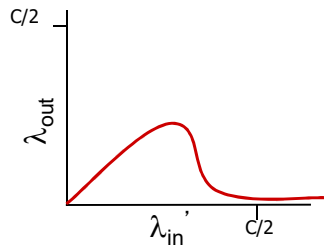
Hỏi: Điều gì sẽ xảy ra khi λ_{in} và λ_{in}' tăng lên?

Trả lời: Nếu λ_{in}' (đỏ) tăng lên, thì tất cả gói tin màu xanh nước biển đang đến tại hàng đợi phía trên sẽ bị bỏ rơi, thông lượng màu xanh nước biển sẽ tiến đến 0



Tăng giao vận 3-93

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



“Chi phí” khác của tắc nghẽn:

- ❖ Khi gói tin bị bỏ rơi, thì bất kỳ luồng lưu lượng truyền nào cho gói tin đều là lãng phí!

Tăng giao vận 3-94

Phương pháp tiếp cận hướng tới điều khiển tắc nghẽn

Hai cách tiếp cận chính hướng tới điều khiển tắc nghẽn:

Điều khiển tắc nghẽn end-end:

- ❖ Không có phản hồi rõ ràng từ mạng
- ❖ Tắc nghẽn được suy ra từ hiện tượng mất mát hoặc trễ quan sát được tại hệ thống đầu cuối
- ❖ Cách tiếp cận này được thực hiện bởi TCP

Điều khiển tắc nghẽn có hỗ trợ từ mạng:

- ❖ Các bộ định tuyến cung cấp phản hồi tới các hệ thống đầu cuối.
 - bit đơn chỉ thị tắc nghẽn (SNA, DECbit, TCP/IP ECN, ATM)
 - Tốc độ gửi được xác định rõ ràng

Tăng giao vận 3-95

Case study: điều khiển tắc nghẽn trong ATM ABR

ABR: tốc độ bit có sẵn:

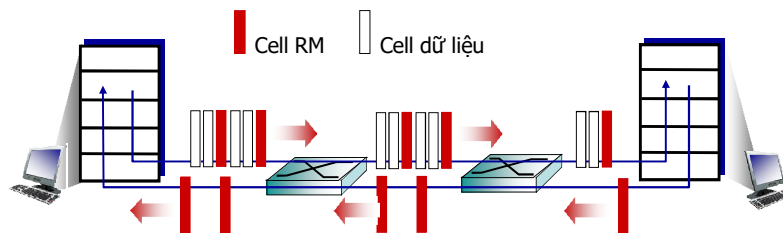
- ❖ “Dịch vụ mềm dẻo”
- ❖ Nếu đường dẫn phía bên gửi “dưới tải” thì:
 - Bên gửi nên dùng băng thông có sẵn
- ❖ Nếu đường dẫn bên gửi bị tắc nghẽn thì:
 - Bên gửi nên giảm để đảm bảo tốc độ là tối thiểu

Các cell RM (quản lý tài nguyên):

- ❖ Được gửi bởi bên gửi, xen kẽ với các cell dữ liệu
- ❖ Các bit trong cell RM được thiết lập bởi các switch (“có hỗ trợ từ mạng”)
 - *bit NI*: không tăng theo tốc độ (tắc nghẽn nhẹ)
 - *bit CI*: xác định tắc nghẽn
- ❖ Các cell RM được trả lại bên gửi từ bên nhận, với các bit còn nguyên vẹn

Tầng giao vận 3-96

Case study: điều khiển tắc nghẽn trong ATM ABR



- ❖ Hai byte trường ER (explicit rate) trong cell RM
 - Switch bị tắc nghẽn có thể có giá trị ER thấp hơn trong cell
 - Bên gửi gửi với tốc độ được hỗ trợ lớn nhất trên đường truyền
- ❖ Bit EFCI trong các cell dữ liệu: được thiết lập là 1 trong switch bị tắc nghẽn
 - Nếu cell dữ liệu trước cell RM có EFCI được thiết lập, thì bên nhận thiết lập bit CI trong cell RM được trả về

Tầng giao vận 3-97

Chương 3: Nội dung

3.1 Các dịch vụ tầng giao vận

3.2 Ghép kênh và phân kênh

3.3 Vận chuyển không kết nối: UDP

3.4 Các nguyên lý truyền dữ liệu tin cậy

3.5 Vận chuyển hướng kết nối: TCP

- Cấu trúc đoạn dữ liệu (segment)

- Truyền dữ liệu tin cậy

- Điều khiển luồng

- Quản lý kết nối

3.6 Các nguyên lý điều khiển tắc nghẽn

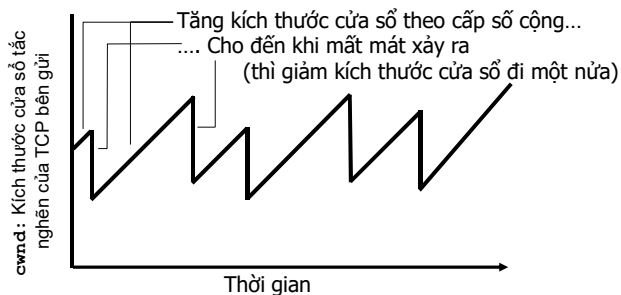
3.7 Điều khiển tắc nghẽn TCP

Tầng giao vận 3-98

Điều khiển tắc nghẽn trong TCP: Tăng theo cấp số cộng Giảm theo cấp số nhân

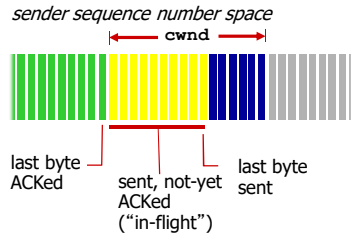
- ❖ **Cách tiếp cận:** Bên gửi tăng tốc độ truyền (kích thước cửa sổ), thăm dò bằng thông sử dụng, cho đến khi có mất mát xảy ra
 - **Tăng theo cấp số cộng:** tăng **cwnd** theo 1 MSS mỗi RTT cho đến khi phát hiện mất mát
 - **Giảm theo cấp số nhân:** giảm **cwnd** đi một nửa sau khi phát hiện có mất mát

Thăm dò bằng thông



Tầng giao vận 3-99

Chi tiết điều khiển tắc nghẽn trong TCP



Tốc độ gửi của TCP:

- Được hiểu là: gửi cwnd byte, chờ một RTT cho ACK, sau đó gửi nhiều byte hơn

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

- Bên gửi giới hạn việc truyền:

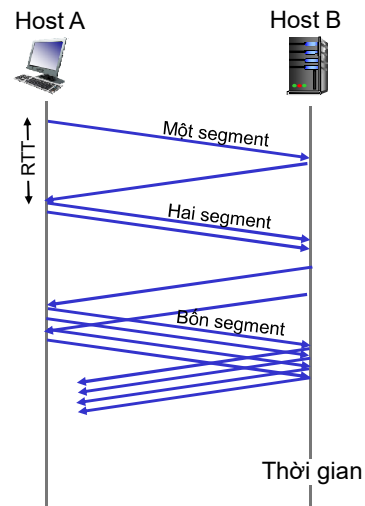
$$\text{LastByteSent} - \text{LastByteAcked} \leq \text{cwnd}$$

- cwnd thay đổi, có chức năng nhận biết tắc nghẽn trên mạng

Tầng giao vận 3-100

TCP khởi động chậm

- Khi kết nối bắt đầu, tăng tốc độ lên theo cấp số nhân cho đến khi có sự kiện mất mát đầu tiên xảy ra:
 - Khởi tạo **cwnd** = 1 MSS
 - Tăng gấp đôi **cwnd** cho mỗi RTT
 - Thực hiện tăng **cwnd** cho mỗi ACK nhận được
- Tổng kết:** tốc độ khởi đầu là chậm nhưng sau đó tăng lên theo cấp số nhân



Tầng giao vận 3-101

Phát hiện và phản ứng lại khi có mất mát

- ❖ Mất mát được xác định khi bị timeout:
 - **cwnd** được thiết lập lại là 1 MSS;
 - Cửa sổ sau đó sẽ tăng theo cấp số nhân (như trong khởi động chậm) tới ngưỡng, thì sẽ tăng tuyến tính
- ❖ Mất mát được xác định khi thấy 3 ACK trùng lặp: TCP RENO
 - Các ACK trùng lặp xác định khả năng truyền các segment của mạng
 - **cwnd** giảm đi một nửa kích thước cửa sổ, sau đó tăng tuyến tính
- ❖ TCP Tahoe luôn đặt **cwnd** là 1 (khi có timeout hoặc 3 ACK trùng lặp)

Tăng giao vận 3-102

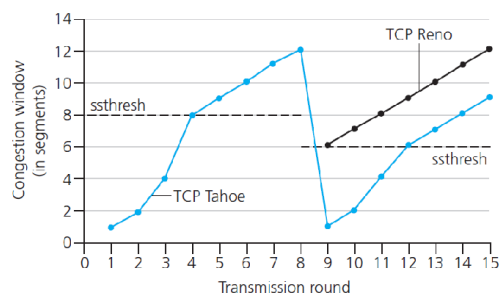
Hiện thực trong TCP

Hỏi: Khi nào nên chuyển từ tăng theo cấp số nhân sang tăng tuyến tính?

Trả lời: khi **cwnd** đạt đến 1/2 giá trị của nó trước khi timeout.

Cài đặt:

- ❖ Biến **ssthresh**
- ❖ Với mỗi sự kiện mất mát, **ssthresh** sẽ được đặt bằng 1/2 **cwnd** ngay trước khi có mất mát xảy ra



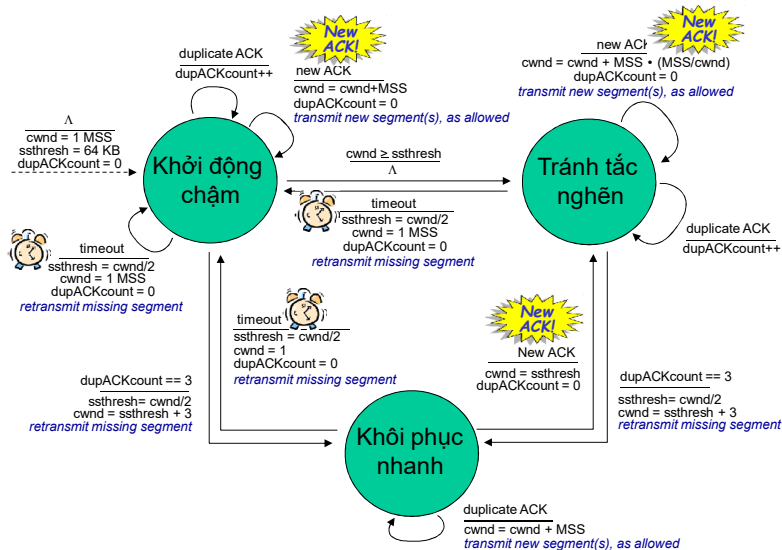
Tăng giao vận 3-103

Tổng kết điều khiển tắc nghẽn trong TCP

- ❖ Khi **cwnd** dưới **ssthresh**, bên gửi đang trong giai đoạn **khởi động chậm**, kích thước cửa sổ tăng nhanh theo cấp số nhân.
- ❖ Khi **cwnd** trên **ssthresh**, bên gửi đang trong giai đoạn **tránh tắc nghẽn**, kích thước cửa sổ tăng nhanh theo cấp tuyến tính.
- ❖ Khi có **3 ACK trùng lặp** xảy ra, **ssthresh = cwnd/2** và **cwnd = ssthresh**.
- ❖ Khi **timeout** xảy ra, **ssthresh = cwnd/2** và **cwnd=1 MSS**.

Tăng giao vận 3-104

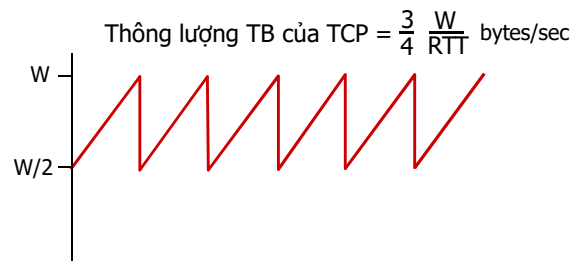
Tổng kết điều khiển tắc nghẽn trong TCP



Tăng giao vận 3-105

Thông lượng của TCP

- ❖ Thông lượng trung bình của TCP được xác định qua kích thước cửa sổ và RTT như thế nào?
 - Bỏ qua khởi động chậm, giả sử dữ liệu luôn luôn được gửi
- ❖ W : kích thước cửa sổ (được tính bằng byte) khi có mất mát xảy ra
 - Kích thước cửa sổ trung bình (số byte trong lưu lượng) là $\frac{3}{4} W$
 - Thông lượng trung bình là $\frac{3}{4} W$ trên RTT



Tăng giao vận 3-106

TCP trong tương lai: TCP qua “đường truyền rộng và dài”

- ❖ Ví dụ: Các segment dài 1500 byte, RTT là 100ms, muốn đạt được thông lượng là 10 Gbps
- ❖ Yêu cầu lưu lượng với kích thước cửa sổ là $W = 83,333$ segment
- ❖ Thông lượng của xác suất mất đoạn là L [Mathis 1997]:

$$\text{Thông lượng TCP} = \frac{1.22 \cdot MSS}{RTT \sqrt{L}}$$

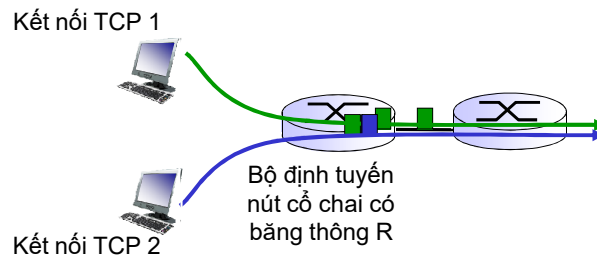
→ Để có được thông lượng là 10 Gbps, cần tỷ lệ mất mát là $L = 2 \cdot 10^{-10}$ – *một tỷ lệ mất mát rất nhỏ!*

- ❖ Các phiên bản mới của TCP dành cho tốc độ cao

Tăng giao vận 3-107

Tính công bằng trong TCP

Mục tiêu: Nếu K phiên làm việc trong TCP chia sẻ cùng liên kết nút cổ chai có băng thông là R, thì mỗi phiên nên có tốc độ trung bình là R/K

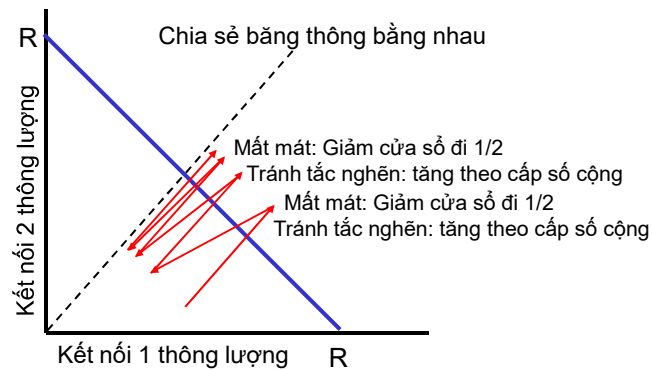


Tăng giao vận 3-108

Tại sao TCP là công bằng?

Hai phiên làm việc cạnh tranh nhau:

- ❖ Tăng theo cấp số cộng làm tăng lưu lượng liên tục
- ❖ Giảm theo cấp số nhân làm giảm lưu lượng tương ứng



Tăng giao vận 3-109

Tính công bằng (tiếp)

Tính công bằng và UDP

- ❖ Các ứng dụng đa phương tiện thường không dùng TCP
 - Không muốn tốc độ bị chặn do điều khiển tắc nghẽn
- ❖ Thay bằng dùng UDP:
 - Gửi audio/video với tốc độ ổn định, chịu mất mát gói tin

Tính công bằng và kết nối song song trong TCP

- ❖ Ứng dụng có thể mở nhiều kết nối song song giữa hai host
- ❖ Các trình duyệt web làm theo cách này
- ❖ Ví dụ: liên kết có tốc độ R hỗ trợ 9 kết nối:
 - Ứng dụng mới yêu cầu 1 TCP, có tốc độ R/10
 - Ứng dụng mới yêu cầu 11 TCP, có tốc độ R/2

Tầng giao vận 3-110

Chương 3: Tổng kết

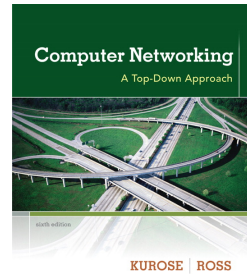
- ❖ Các nguyên lý của các dịch vụ tầng giao vận:
 - Ghép kênh, phân kênh
 - Truyền dữ liệu tin cậy
 - Điều khiển luồng
 - Điều khiển tắc nghẽn
- ❖ Hiện thực trên mạng Internet:
 - UDP
 - TCP

Tiếp theo:

- ❖ Kết thúc các vấn đề liên quan đến “phần cạnh” của mạng (tầng ứng dụng và tầng giao vận)
- ❖ Chuẩn bị đi vào “phần lõi” của mạng

Tầng giao vận 3-111

Chương 4 Tầng mạng



*Computer
Networking: A Top
Down Approach*
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Người dịch: Nguyễn Thanh Thủy

Tài liệu được dịch cho mục đích giảng dạy (được sự đồng ý của tác giả).

© All material copyright 1996-2012.
J.F Kurose and K.W. Ross, All Rights Reserved

Tầng mạng 4-1

Chương 4: Tầng mạng

Mục tiêu:

- ❖ Hiểu được nguyên lý của các dịch vụ tầng mạng:
 - Các mô hình dịch vụ tầng mạng
 - Chuyển tiếp (forwarding) và định tuyến (routing)
 - Bộ định tuyến làm việc như thế nào
 - Định tuyến (chọn đường)
- ❖ Cài đặt hiện thực trong mạng Internet

Tầng mạng 4-2

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

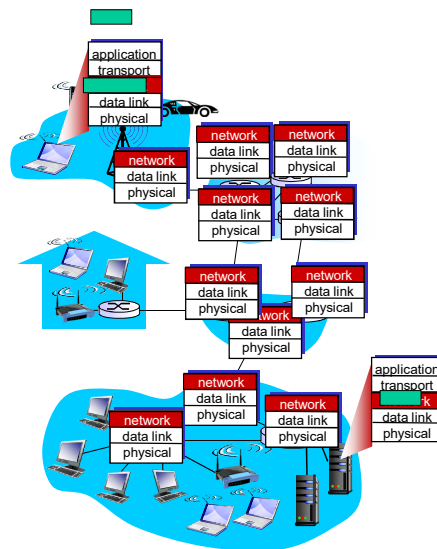
4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tầng mạng 4-3

Tầng mạng

- ❖ Chuyển các segment từ host gửi sang host nhận
- ❖ Bên gửi sẽ đóng gói các segment vào trong các datagram
- ❖ Bên nhận sẽ phân phối các segment đến tầng giao vận
- ❖ Các giao thức tầng mạng được cài đặt trong *mỗi* host và router
- ❖ Router kiểm tra các trường trong tiêu đề của tất cả các gói tin IP datagram để chuyển nó đi tiếp



Tầng mạng 4-4

Hai chức năng chính của tầng mạng

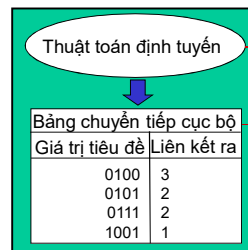
- ❖ **Chuyển tiếp (forwarding):** chuyển các gói tin từ đầu vào tới đầu ra phù hợp của router
- ❖ **Định tuyến (routing):** xác định tuyến đường đi cho các gói tin từ nguồn đến đích.
 - Các thuật toán định tuyến

Tương tự:

- ❖ **Định tuyến:** tiến trình lập kế hoạch chuyển đi từ nguồn đến đích
- ❖ **Chuyển tiếp:** tiến trình vận chuyển qua một giao điểm (nút)

Tầng mạng 4-5

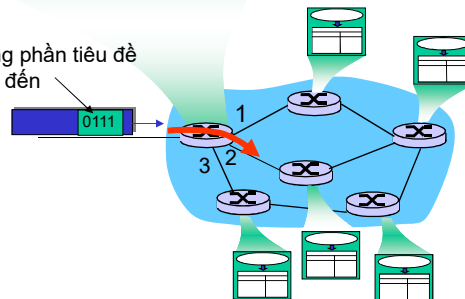
Tác động qua lại giữa định tuyến và chuyển tiếp



Thuật toán định tuyến xác định đường đi từ đầu cuối này đến đầu cuối kia trên mạng

Bảng chuyển tiếp xác định việc chuyển tiếp cục bộ tại bộ định tuyến này

Giá trị trong phần tiêu đề của gói tin đến



Tầng mạng 4-6

Thiết lập kết nối

- ❖ Chức năng quan trọng thứ 3 trong một số kiến trúc mạng:
 - ATM, frame relay, X.25
- ❖ Trước khi các datagram chuyển đi, hai host đầu cuối và các router trung gian thiết lập kết nối ảo
 - Các router cũng liên quan
- ❖ Dịch vụ kết nối tầng mạng và tầng giao vận:
 - **Tầng mạng**: giữa hai host (cũng có thể chứa các router trung gian trong trường hợp kết nối ảo)
 - **Tầng giao vận**: giữa hai tiến trình

Tầng mạng 4-7

Mô hình dịch vụ tầng mạng

Hỏi: *Mô hình dịch vụ* nào cho “kênh” vận chuyển các datagram từ bên gửi đến bên nhận?

Ví dụ các dịch vụ cho các datagram riêng:

- ❖ Giao nhận đảm bảo
- ❖ Giao nhận đảm bảo với trễ nhỏ hơn 40 msec

Ví dụ các dịch vụ cho một luồng datagram:

- ❖ Giao nhận datagram theo đúng thứ tự
- ❖ Đảm bảo băng thông tối thiểu cho luồng
- ❖ Hạn chế những thay đổi trong khoảng trống giữa các gói tin

Tầng mạng 4-8

Các mô hình dịch vụ tầng mạng

Kiến trúc mạng	Mô hình dịch vụ	Bảo đảm?				Phản hồi tắc nghẽn
		Băng thông	Mất mát	Đúng thứ tự	Thời gian thực	
Internet	best effort	Không	Không	Không	Không	Không (phát hiện thông qua mất mát)
ATM	CBR	Tốc độ ổn định	Có	Có	Có	Không tắc nghẽn
ATM	VBR	Đảm bảo tốc độ	Có	Có	Có	Không tắc nghẽn
ATM	ABR	Bảo đảm tối thiểu	Không	Có	Không	Có
ATM	UBR	Không	Không	Có	Không	Có

Tầng mạng 4-9

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tầng mạng 4-10

Dịch vụ hướng kết nối và không kết nối

- ❖ Mạng *chuyển gói (datagram network)* cung cấp dịch vụ *hướng không kết nối* tầng mạng
- ❖ Mạng *mạch ảo (virtual-circuit network)* cung cấp dịch vụ *hướng kết nối* tầng mạng
- ❖ Tương tự với các dịch vụ hướng kết nối/không kết nối TCP/UDP tầng giao vận, nhưng:
 - **Dịch vụ:** host-to-host
 - **Không lựa chọn:** tầng mạng chỉ cung cấp hoặc dịch vụ này, hoặc dịch vụ kia
 - **Cài đặt:** bên trong phần lõi của mạng

Tầng mạng 4-11

Mạch ảo (Virtual circuit - VC)

“Cách xử lý đường từ nguồn đến đích giống như mạch điện thoại”

- Hiệu suất tốt
 - Mạng hoạt động theo đường từ nguồn đến đích
- ❖ Thiết lập cuộc gọi, chia nhỏ *mỗi* cuộc gọi trước khi dữ liệu được truyền đi
 - ❖ Mỗi gói tin mang định danh mạch ảo (không phải là địa chỉ của host đích)
 - ❖ *Mỗi* router trên đường đi từ nguồn đến đích duy trì “trạng thái” cho mỗi kết nối qua.
 - ❖ Kết nối, các tài nguyên router (bảng thông, đệm) có thể được *cấp phát* cho mạch ảo (Các tài nguyên dành riêng = dịch vụ dự đoán trước được).

Tầng mạng 4-12

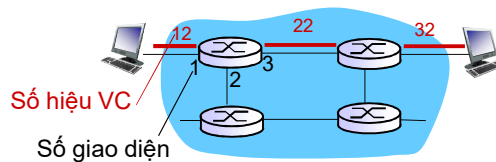
Cài đặt mạch ảo

Một mạch ảo bao gồm:

1. *Đường* từ nguồn đến đích
 2. *Số hiệu mạch ảo*, mỗi số dành cho một liên kết dọc theo đường
 3. *Các điểm đăng ký vào các bảng chuyển tiếp* trong các router dọc theo đường
- ❖ Gói thuộc về mạch ảo sẽ mang số hiệu của mạch ảo (không phải là địa chỉ đích)
 - ❖ Số hiệu mạch ảo có thể được thay đổi trên mỗi liên kết
 - Số hiệu mạch ảo mới được cung cấp từ bảng chuyển tiếp

Tầng mạng 4-13

Bảng chuyển tiếp mạch ảo



Ví dụ bảng chuyển tiếp của router:

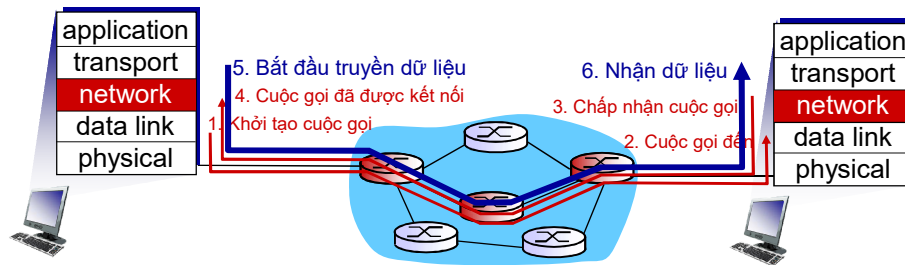
Giao diện đến	Số hiệu VC đến	Giao diện đi	Số hiệu VC đi
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Các router mạch ảo duy trì thông tin trạng thái kết nối!

Tầng mạng 4-14

Mạch ảo: Các giao thức báo hiệu

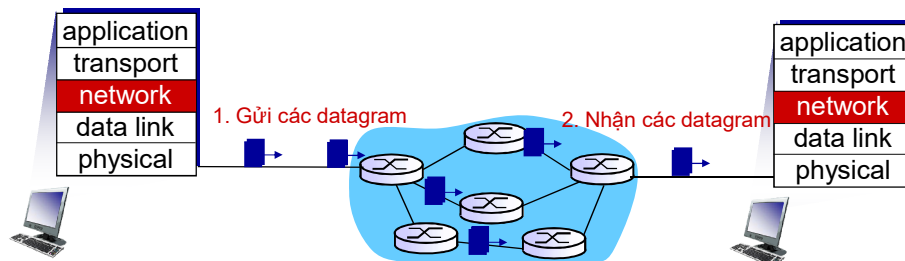
- ❖ Được sử dụng để thiết lập, duy trì phân mạch ảo
- ❖ Được dùng trong ATM, frame-relay, X.25
- ❖ Không được dùng trong mạng Internet ngày nay



Tầng mạng 4-15

Mạng chuyển gói

- ❖ Không có thiết lập cuộc gọi tại tầng mạng
- ❖ Các router: Không lưu giữ trạng thái về các kết nối giữa các đầu cuối (end-to-end)
 - Không có khái niệm “kết nối” mức mạng
- ❖ Các gói tin được chuyển tiếp bằng cách sử dụng địa chỉ host đích



Tầng mạng 4-16

Bảng chuyển tiếp datagram

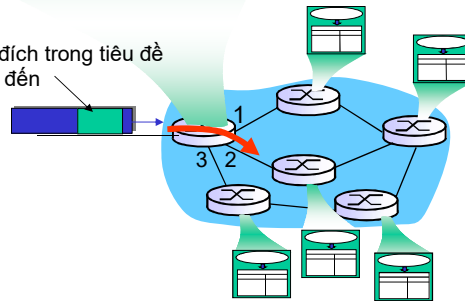
Thuật toán định tuyến

Bảng chuyển tiếp cục bộ

Địa chỉ đích	Liên kết ra
Dãy địa chỉ 1	3
Dãy địa chỉ 2	2
Dãy địa chỉ 3	2
Dãy địa chỉ 4	1

Có 4 tỷ địa chỉ IP, do vậy, nên thay danh sách địa chỉ đích riêng bằng danh sách **đầy** các địa chỉ (bảng các mục tổng hợp)

Địa chỉ IP đích trong tiêu đề của gói tin đến



Tăng mạng 4-17

Bảng chuyển tiếp datagram

Dãy địa chỉ đích	Giao diện liên kết
11001000 00010111 00010000 00000000 đến 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 đến 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 đến 11001000 00010111 00011111 11111111	2
khác	3

Hỏi: Nhưng điều gì sẽ xảy ra khi các dãy không được phân chia hợp lý?

Tăng mạng 4-18

So khớp tiền tố dài nhất

So khớp tiền tố dài nhất

Khi tìm kiếm mục vào trong bảng chuyển tiếp cho một địa chỉ đích xác định, dùng tiền tố địa chỉ *dài nhất* giống với địa chỉ đích.

Dãy địa chỉ đích	Giao diện liên kết
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
khác	3

Ví dụ:

DA: 11001000 00010111 00010110 10100001

Giao diện nào?

DA: 11001000 00010111 00011000 10101010

Giao diện nào?

Tăng mạng 4-19

Chuyển mạch gói hay chuyển mạch ảo: Tại sao?

Internet (datagram)

- ❖ Dữ liệu trao đổi giữa các máy tính
 - Dịch vụ “mềm dẻo”, không giới hạn yêu cầu thời gian
- ❖ Nhiều loại liên kết
 - Các đặc tính khác nhau
 - Khó khăn khi đồng nhất dịch vụ
- ❖ Các hệ thống đầu cuối “thông minh” (máy tính)
 - Có thể đáp ứng, điều thực thi khiển, khôi phục lỗi
 - **Mạng bên trong đơn giản, sự phức tạp nằm ở “phần cạnh”**

ATM (VC)

- ❖ Phát triển từ hệ thống điện thoại
- ❖ Hội thoại của con người:
 - Giới hạn thời gian, yêu cầu độ tin cậy
 - Cần dịch vụ đảm bảo
- ❖ Các hệ thống đầu cuối “ít thông minh”
 - Máy điện thoại
 - **Sự phức tạp ở bên trong mạng**

Tăng mạng 4-20

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

4.6 Định tuyến trong mạng Internet

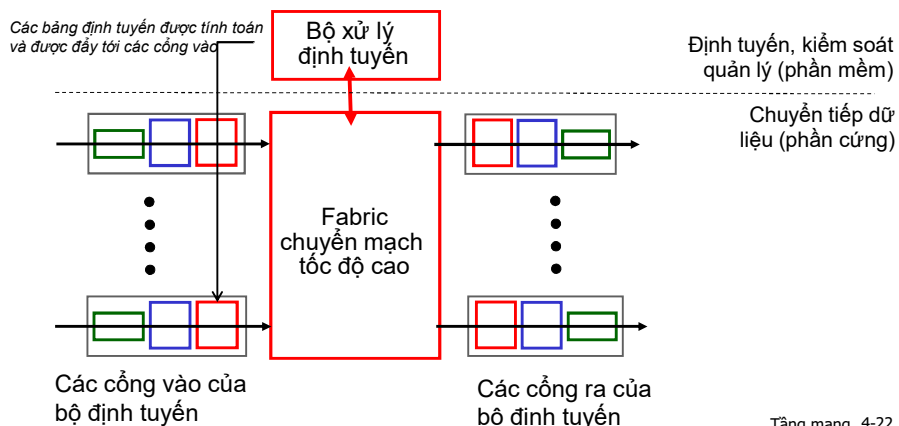
- RIP
- OSPF
- BGP

Tầng mạng 4-21

Khái quát kiến trúc của bộ định tuyến

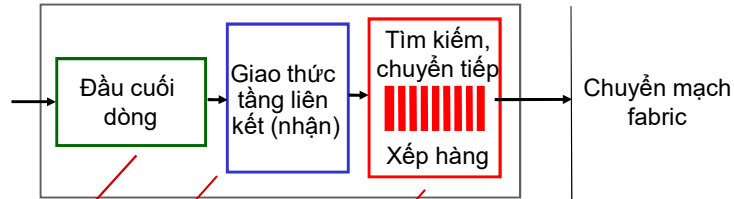
Hai chức năng chính của bộ định tuyến:

- ❖ Chạy các giải thuật/giao thức định tuyến (RIP, OSPF, BGP)
- ❖ *Chuyển tiếp* các datagram từ liên kết vào tới liên kết ra



Tầng mạng 4-22

Các chức năng của cổng vào



Tầng vật lý:
Tiếp nhận mức bit

Tầng liên kết
dữ liệu:

Ví dụ: Ethernet
(xem chương 5)

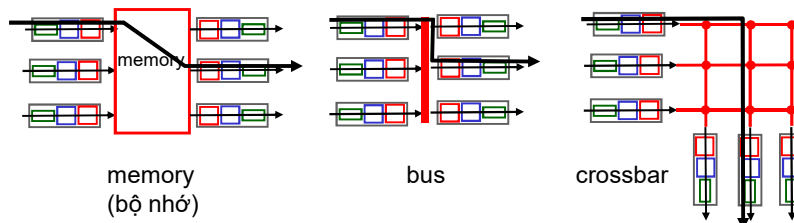
Chuyển mạch không tập trung:

- ❖ Với đích của datagram đã biết, tìm cổng ra bằng bảng chuyển tiếp trong bộ nhớ cổng vào
- ❖ Mục đích: hoàn thành xử lý cổng vào theo “tốc độ dòng”
- ❖ Xếp hàng: nếu các datagram đến nhanh hơn tốc độ chuyển tiếp trong bộ chuyển mạch fabric

Tầng mạng 4-23

Chuyển mạch fabric

- ❖ Chuyển gói tin từ vùng đệm vào đến vùng đệm ra phù hợp
- ❖ Tốc độ chuyển mạch: là tốc độ mà các gói tin có thể được chuyển từ các đầu vào tới các đầu ra
 - Thường được đo như là bội số của tốc độ dòng vào/dòng ra
 - N đầu vào: tốc độ chuyển mạch bằng N lần tốc độ dòng mong muốn
- ❖ Có 3 loại chuyển mạch fabric:

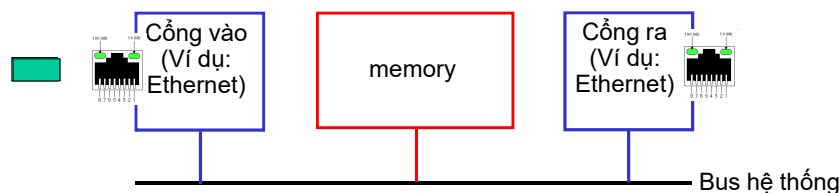


Tầng mạng 4-24

Chuyển mạch qua memory

Các bộ định tuyến thế hệ đầu tiên:

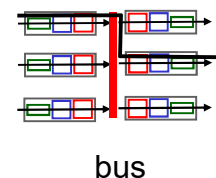
- ❖ Các máy tính truyền thống với các bộ chuyển mạch được điều khiển trực tiếp bởi CPU
- ❖ Gói tin được sao chép vào trong bộ nhớ của hệ thống
- ❖ Tốc độ bị giới hạn bởi băng thông bộ nhớ



Tăng mạng 4-25

Chuyển mạch qua bus

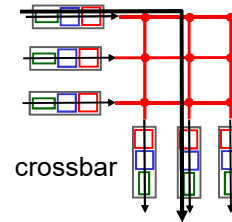
- ❖ Datagram từ bộ nhớ cổng vào tới bộ nhớ cổng ra thông qua một bus chung
- ❖ **Tranh chấp bus:** tốc độ chuyển mạch bị giới hạn bởi băng thông của bus
- ❖ 32 Gbps bus, Cisco 5600: tốc độ đủ cho các router truy nhập và các router của tổ chức.



Tăng mạng 4-26

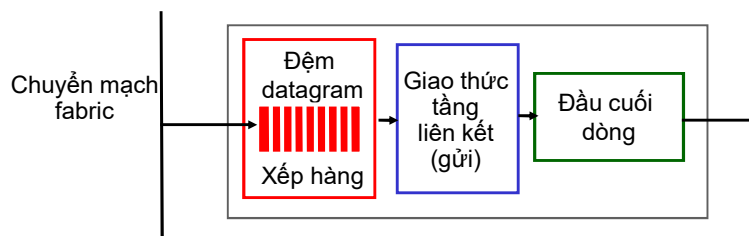
Chuyển mạch thông qua mạng kết nối nội bộ

- ❖ Vượt qua các giới hạn về băng thông của bus
- ❖ Các mạng ban đầu được phát triển để kết nối các bộ vi xử lý thành một bộ đa xử lý
- ❖ Thiết kế nâng cao: phân mảnh datagram thành các cell có độ dài cố định, chuyển mạch các cell qua fabric.
- ❖ Cisco 12000: chuyển mạch 60 Gbps qua mạng kết nối nội bộ



Tầng mạng 4-27

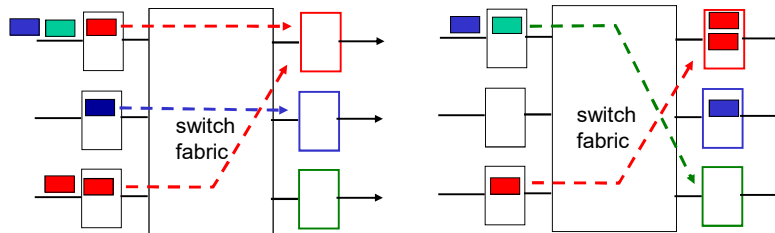
Các cổng ra



- ❖ **Việc đệm** được yêu cầu khi các datagram đến từ fabric nhanh hơn tốc độ truyền đi
- ❖ **Lịch truyền** sẽ lựa chọn các datagram trong hàng đợi để truyền

Tầng mạng 4-28

Xếp hàng tại cổng ra



Tại t , các gói tin từ cổng vào đi đến một cổng ra nhiều hơn

Mỗi lần một gói

- ❖ Việc đệm khi tốc độ đến qua chuyển mạch vượt quá tốc độ dòng ra
- ❖ *Xếp hàng (trễ) và mất mát là do tràn bộ đệm cổng ra!*

Tăng mạng 4-29

Cần bao nhiêu cho bộ đệm?

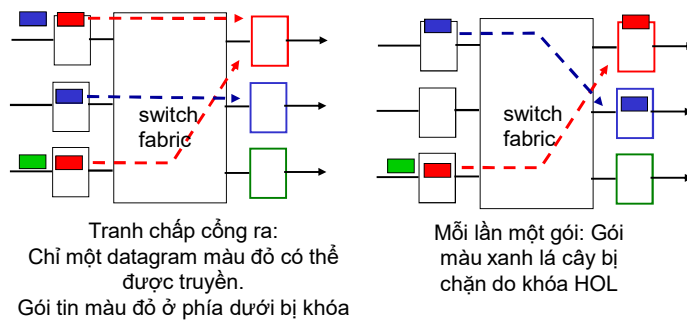
- ❖ Chuẩn RFC 3439: đệm trung bình bằng một RTT “điển hình” (là 250msec) nhân với tốc độ C của liên kết.
 - Ví dụ: liên kết $C = 10$ Gpbs thì đệm là 2.5 Gbit
- ❖ Khuyến nghị hiện tại: với N luồng, đệm được tính bằng:

$$\frac{RTT \cdot C}{\sqrt{N}}$$

Tăng mạng 4-30

Xếp hàng tại cổng vào

- ❖ Nếu fabric chậm hơn so với các cổng vào được kết nối, thì cần phải xếp hàng tại hàng đợi vào.
 - *Trễ xếp hàng và mất mát là do tràn bộ đệm vào!*
- ❖ **Khóa đầu hàng (Head-of-the-Line - HOL):** datagram đã được xếp hàng tại phía trước của hàng đợi ngăn cản các datagram khác trong hàng di chuyển về phía trước



Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

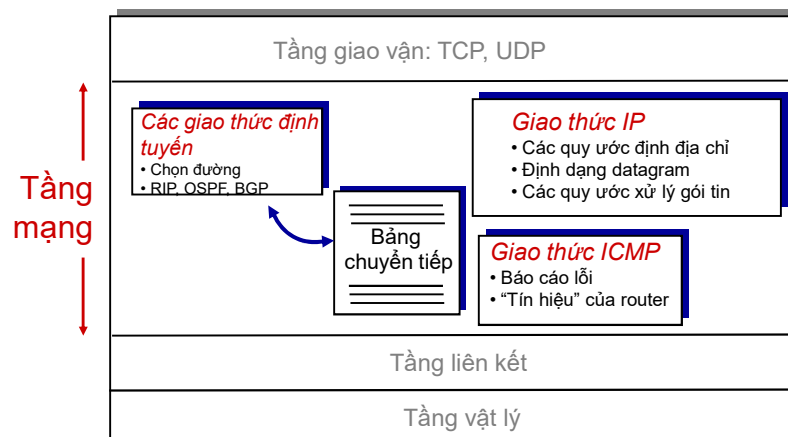
4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tầng mạng 4-32

Tầng mạng trong mạng Internet

Chức năng của tầng mạng tại bộ định tuyến và host:



Tầng mạng 4-33

Định dạng IP datagram

Phiên bản giao thức IP

Chiều dài phần tiêu đề (byte)

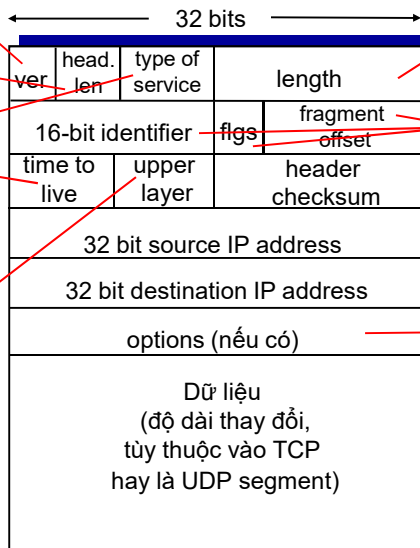
“Loại” dữ liệu

Số hop tối đa còn lại (được giảm đi tại mỗi router)

Giao thức tầng cao hơn thực hiện phân phối payload đi

Bao nhiêu byte cho phần tiêu đề gói tin?

- ❖ 20 byte trong TCP
- ❖ 20 byte trong IP
- ❖ = 40 byte + phần tiêu đề của tầng ứng dụng



Tổng chiều dài datagram (byte)

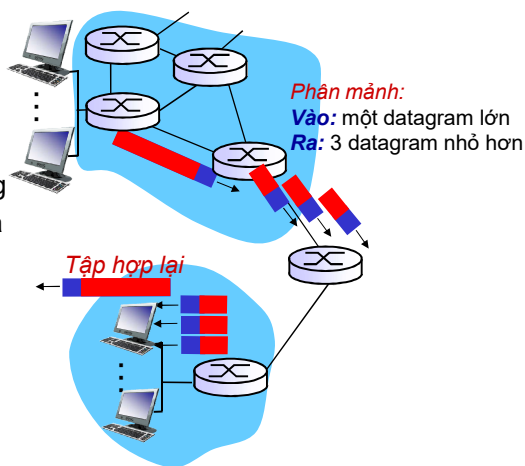
Dùng cho việc phân mảnh/tập hợp lại

Ví dụ: trường timestamp, ghi nhận đường đi, cụ thể là danh sách các router đi qua.

Tầng mạng 4-34

Phân mảnh và tập hợp lại gói tin IP

- ❖ Các liên kết mạng có MTU (max.transfer size) – frame mức liên kết lớn nhất có thể.
 - Các loại liên kết khác nhau sẽ có MTU khác nhau
- ❖ IP datagram lớn sẽ được chia (“phân mảnh”) bên trong mạng
 - Một datagram sẽ được chia thành một số datagram
 - Chúng sẽ được “tập hợp lại” tại đích cuối cùng
 - Các bit trong tiêu đề IP được dùng để xác định thứ tự liên quan đến các mảnh



Tầng mạng 4-35

Phân mảnh và tập hợp lại gói tin IP

Ví dụ:

- ❖ Datagram 4000 byte
- ❖ MTU = 1500 byte

length	ID	fragflag	offset
=4000	=x	=0	=0

Một datagram lớn được chia thành một số datagram nhỏ hơn

1480 byte trong trường dữ liệu

offset = 1480/8

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

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

length	ID	fragflag	offset
=1040	=x	=0	=370

Tầng mạng 4-36

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

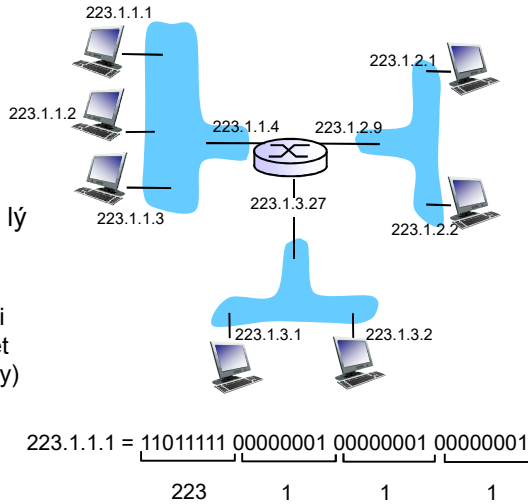
4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tầng mạng 4-37

Định địa chỉ IP: giới thiệu

- ❖ **Địa chỉ IP:** 32-bit định danh cho *giao diện* (*interface*) của host và router
- ❖ **Giao diện:** kết nối giữa host/router với liên kết vật lý
 - Một router thường có nhiều giao diện
 - Một host có một hoặc hai giao diện (Ví dụ: Ethernet có dây, 802.11 không dây)
- ❖ **Địa chỉ IP được gắn với từng giao diện**



Tầng mạng 4-38

Định địa chỉ IP: giới thiệu

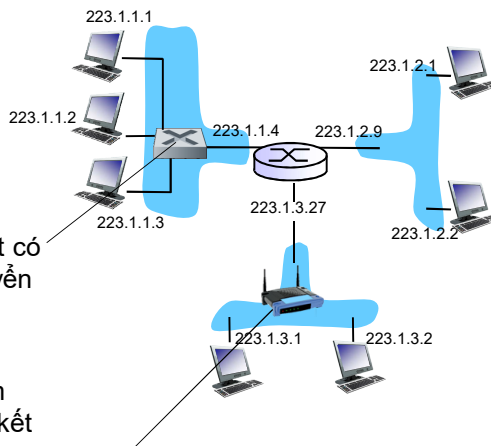
Hỏi: Thực tế các giao diện được kết nối như thế nào?

Trả lời: Sẽ học trong các chương sau (5,6).

Trả lời: Các giao diện Ethernet có dây được kết nối bởi các chuyển mạch Ethernet

Hiện tại: Không cần quan tâm đến việc các giao diện được kết nối với nhau như thế nào (mà không có sự can thiệp của router)

Trả lời: Các giao diện WiFi không dây được kết nối bởi trạm cơ sở WiFi



Tầng mạng 4-39

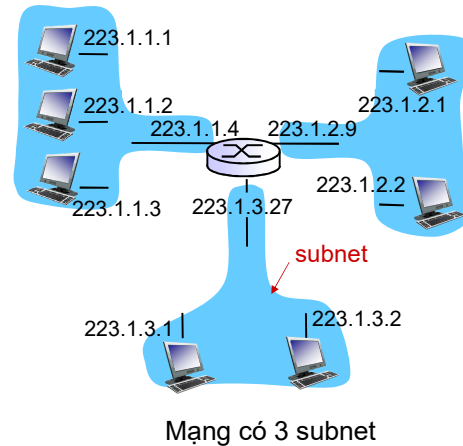
Subnet (Mạng con)

❖ Địa chỉ IP:

- Phần subnet – các bit cao (bên trái)
- Phần host – các bit thấp (bên phải)

❖ Subnet là gì?

- Các giao diện của thiết bị có cùng phần subnet của địa chỉ IP
- Có thể tìm thấy nhau mà *không cần sự can thiệp của router*

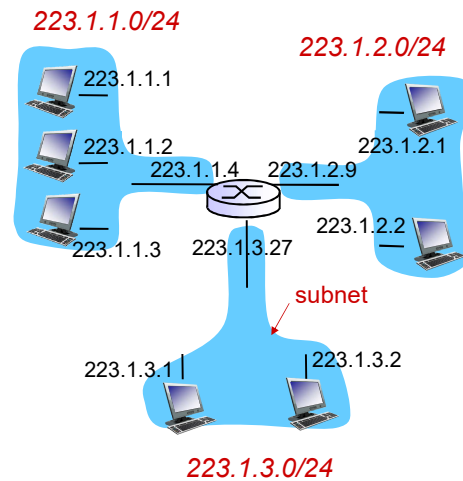


Tầng mạng 4-40

Subnet

Phương pháp

- ❖ Để xác định các subnet, tách mỗi giao diện từ host hoặc router, tạo thành các vùng mạng độc lập
- ❖ Mỗi mạng độc lập được gọi là một *subnet*

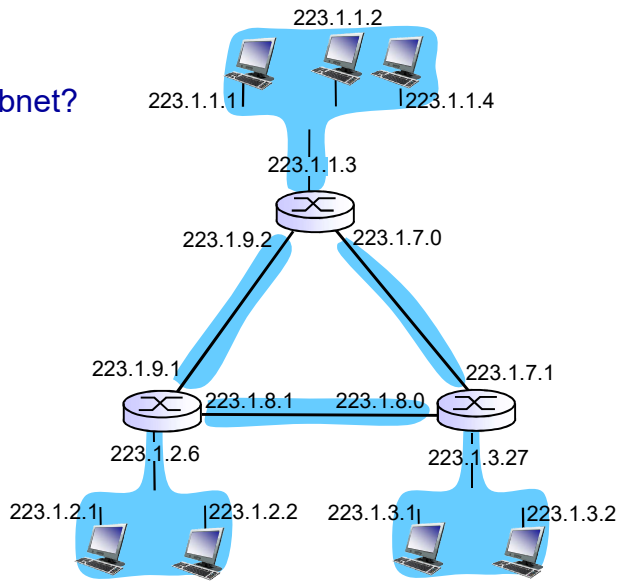


Mặt nạ mạng con (subnet mask): /24

Tầng mạng 4-41

Subnet

Có bao nhiêu subnet?



Tầng mạng 4-42

Định địa chỉ IP: Phân lớp địa chỉ IPv4

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

	# of network	# of hosts
Class A	128	2^{24}
Class B	16384	65536
Class C	2^{21}	256

Hạn chế: lãng phí không gian địa chỉ

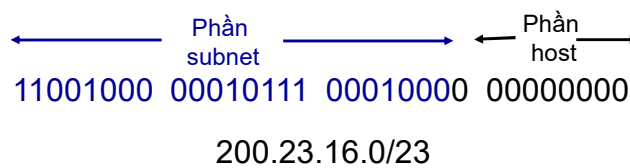
- Việc phân chia cứng thành các lớp (A, B, C, D, E) làm hạn chế việc sử dụng toàn bộ không gian địa chỉ

Tầng mạng 4-43

Định địa chỉ IP: CIDR

CIDR: Classless Inter Domain Routing

- Phần địa chỉ của subnet có độ dài tùy ý
- Định dạng địa chỉ: **a.b.c.d/x**, với x là số bit trong phần subnet của địa chỉ



Tầng mạng 4-44

Làm thế nào để có được một địa chỉ IP?

Hỏi: Làm thế nào để một *host* lấy được một địa chỉ IP?

- ❖ Mã hóa cứng trong một tệp bởi người quản trị hệ thống
 - Windows: control-panel->network->configuration->tcp/ip->properties
 - UNIX: /etc/rc.config
- ❖ **DHCP: Dynamic Host Configuration Protocol:** tự động lấy địa chỉ từ server
 - “plug-and-play”

Tầng mạng 4-45

DHCP: Dynamic Host Configuration Protocol

Mục đích: cho phép host có được địa chỉ IP một cách tự động từ server mạng khi kết nối vào mạng

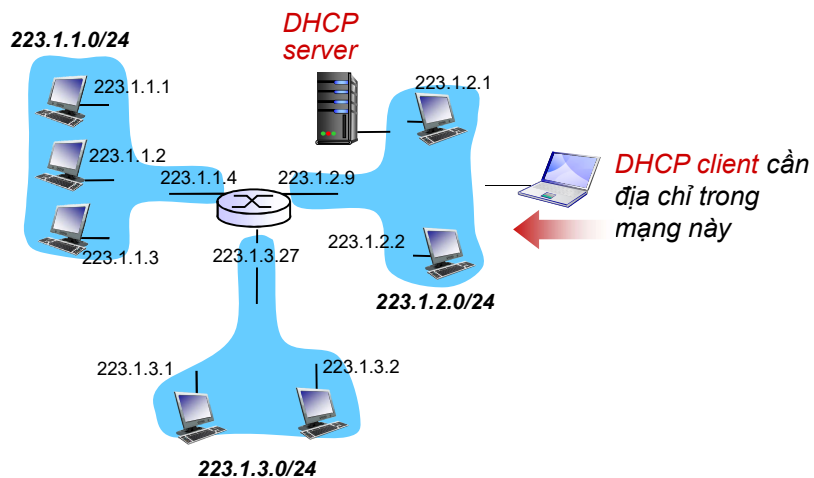
- Có thể làm mới địa chỉ đang dùng
- Cho phép dùng lại địa chỉ (chỉ giữ địa chỉ khi đang kết nối)
- Hỗ trợ cho người dùng di động khi muốn kết nối vào mạng

Khái quát DHCP:

- Host gửi thông điệp quảng bá “DHCP discover” [optional]
- DHCP server trả lời bằng thông điệp “DHCP offer” [optional]
- Host yêu cầu địa chỉ IP bằng thông điệp “DHCP request”
- DHCP server gửi địa chỉ bằng thông điệp “DHCP ack”

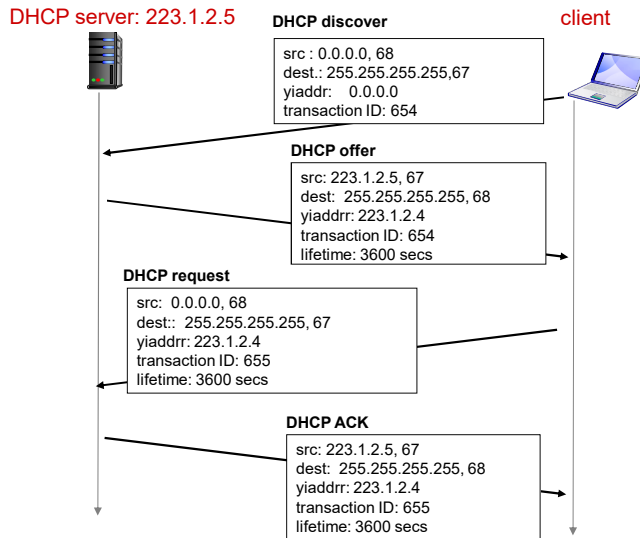
Tầng mạng 4-46

Kịch bản DHCP client-server



Tầng mạng 4-47

Kịch bản DHCP client-server



Tăng mạng 4-48

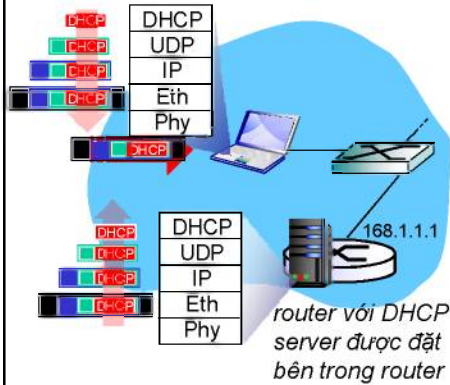
DHCP: có nhiều địa chỉ IP hơn

DHCP có thể cho phép có nhiều địa chỉ IP hơn số địa chỉ IP được phân bổ cho subnet:

- Địa chỉ của router của hop đầu tiên cho client
- Tên và địa chỉ IP của DNS sever
- Mặt nạ mạng (chỉ ra phần host và phần mạng của một địa chỉ)

Tăng mạng 4-49

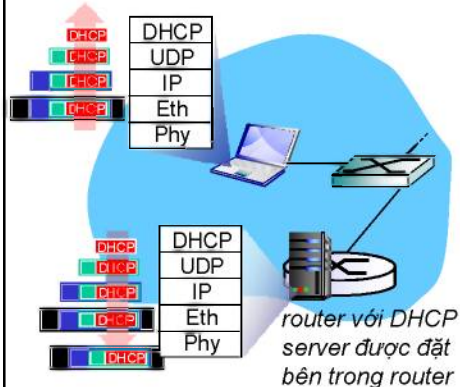
DHCP: Ví dụ



- ❖ Laptop đang kết nối cần một địa chỉ IP, địa chỉ IP của router của hop đầu tiên, địa chỉ của DNS server: dùng DHCP
- ❖ DHCP yêu cầu sẽ được đóng gói trong UDP, UDP được đóng gói trong IP, và IP được đóng gói trong 802.1 Ethernet
- ❖ Gửi quảng bá khung Ethernet (đích: FFFFFFFF) trên mạng LAN, được router đang chạy DHCP server nhận
- ❖ Ethernet được cắt bỏ phần tiêu đề thành IP, IP được cắt bỏ phần tiêu đề thành UDP, UDP được cắt bỏ phần tiêu đề thành DHCP.

Tầng mạng 4-50

DHCP: Ví dụ



- ❖ DHCP server định dạng DHCP ACK bao gồm địa chỉ IP của client, địa chỉ IP của router của hop đầu tiên cho client, tên và địa chỉ IP của DNS server
- ❖ Sau khi được đóng gói ở DHCP server, frame được chuyển tiếp cho client, việc cắt bỏ các phần tiêu đề để thành thông điệp DHCP được thực hiện tại client
- ❖ Lúc này, client biết được địa chỉ IP của nó, tên và địa chỉ IP của DNS server, và địa chỉ IP của router của hop đầu tiên của nó.

Tầng mạng 4-51

DHCP: đầu ra trong Wireshark (LAN ở nhà)

Message type: **Boot Request (1)**
 Hardware type: Ethernet
 Hardware address length: 6
 Hops: 0
Transaction ID: 0x6b3a11b7
 Seconds elapsed: 0
 Bootp flags: 0x0000 (Unicast)
 Client IP address: 0.0.0.0 (0.0.0.0)
 Your (client) IP address: 0.0.0.0 (0.0.0.0)
 Next server IP address: 0.0.0.0 (0.0.0.0)
 Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
 Server host name not given
 Boot file name not given
 Magic cookie: (OK)
 Option: (t=53,l=1) DHCP Message Type = DHCP Request
 Option: (61) Client identifier
 Length: 7; Value: 010016D323688A;
 Hardware type: Ethernet
 Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
 Option: (t=50,l=4) Requested IP Address = 192.168.1.101
 Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
 Length: 11; Value: 010F03062C2E2F1F21F92B
 1 = Subnet Mask; 15 = Domain Name
 3 = Router; 6 = Domain Name Server
 44 = NetBIOS over TCP/IP Name Server

Yêu cầu

Message type: **Boot Reply (2)**
 Hardware type: Ethernet
 Hardware address length: 6
 Hops: 0
Transaction ID: 0x6b3a11b7
 Seconds elapsed: 0
 Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
 Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
 Relay agent IP address: 0.0.0.0 (0.0.0.0)
 Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
 Server host name not given
 Boot file name not given
 Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
 Length: 12; Value: 445747E2445749F244574092;
 IP Address: 68.87.71.226;
 IP Address: 68.87.73.242;
 IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."

Đáp ứng

Tầng mạng 4-52

Làm thế nào có được một địa chỉ IP?

Hỏi: Làm thế nào để mạng có được phần subnet của địa chỉ IP?

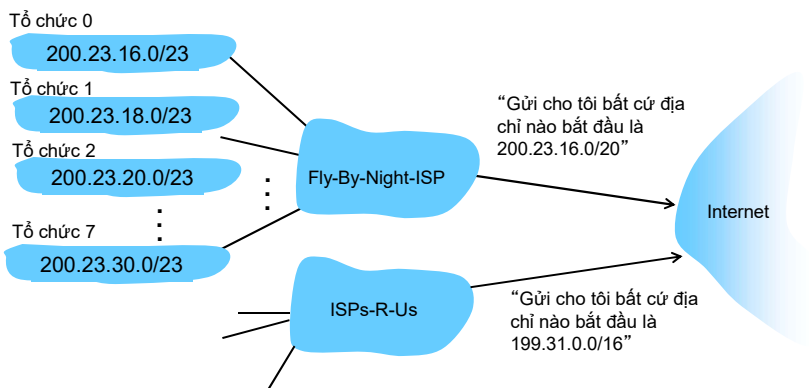
Trả lời: Lấy theo phần được phân bổ từ không gian địa chỉ của nhà cung cấp ISP.

Khối của ISP	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/20
Tổ chức 0	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/23
Tổ chức 1	<u>11001000</u>	<u>00010111</u>	<u>00010010</u>	00000000	200.23.18.0/23
Tổ chức 2	<u>11001000</u>	<u>00010111</u>	<u>00010100</u>	00000000	200.23.20.0/23
...
Tổ chức 7	<u>11001000</u>	<u>00010111</u>	<u>00011110</u>	00000000	200.23.30.0/23

Tầng mạng 4-53

Định địa chỉ phân cấp: tích hợp định tuyến

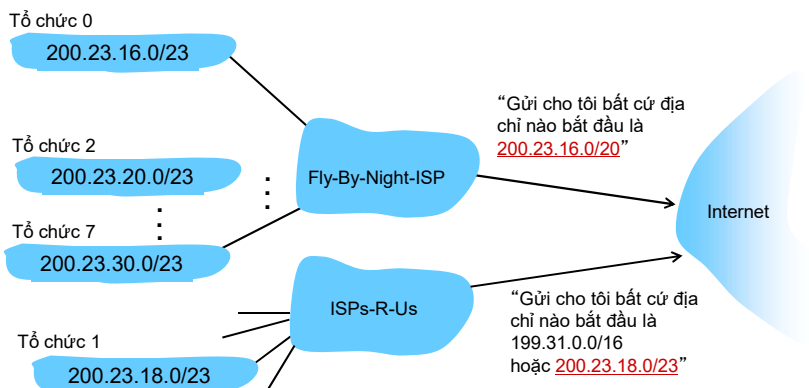
Định địa chỉ phân cấp cho phép quảng bá hiệu quả thông tin định tuyến:



Tầng mạng 4-54

Định địa chỉ phân cấp: định tuyến cụ thể hơn

ISPs-R-Us có nhiều cách định tuyến cụ thể hơn đến Tổ chức 1



Tầng mạng 4-55

Định địa chỉ IP...

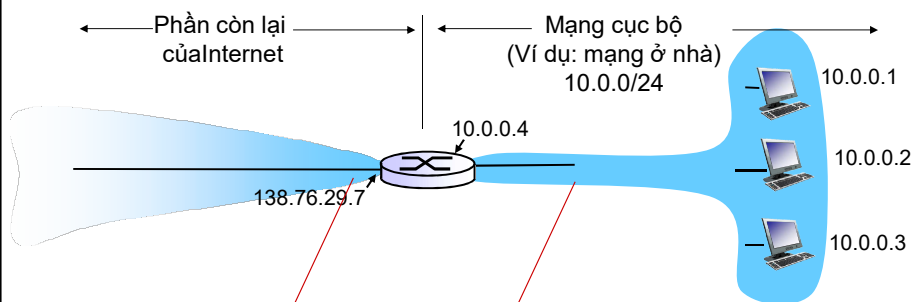
Hỏi: Làm thế nào một ISP có thể lấy được khối địa chỉ?

Trả lời: ICANN: Internet Corporation for Assigned Names and Numbers <http://www.icann.org/>

- Phân bổ địa chỉ
- Quản lý DNS
- Gán các tên miền, giải quyết tranh chấp

Tầng mạng 4-56

NAT: network address translation (chuyển đổi địa chỉ mạng)



Tất cả các datagram **đi ra khỏi** mạng cục bộ đều có **cùng** địa chỉ IP NAT nguồn duy nhất là 138.76.29.7 với các số hiệu cổng nguồn khác nhau

Các datagram với nguồn và đích khác nhau trong mạng này có địa chỉ 10.0.0/24 cho nguồn và đích

Tầng mạng 4-57

NAT: network address translation

Lý do: Mạng cục bộ chỉ dùng một địa chỉ IP đối với hệ thống mạng bên ngoài:

- Không cần thiết sử dụng cả dãy địa chỉ từ một ISP: chỉ cần một địa chỉ cho tất cả các dịch vụ
- Có thể thay đổi địa chỉ của dịch vụ trong mạng cục bộ mà không cần thông báo với hệ thống mạng bên ngoài.
- Có thể thay đổi ISP mà không cần thay đổi địa chỉ của các dịch vụ bên trong mạng cục bộ
- Hệ thống mạng bên ngoài không nhìn thấy, cũng không biết được địa chỉ rõ ràng của các thiết bị bên trong mạng cục bộ (tăng tính bảo mật)

Tăng mạng 4-58

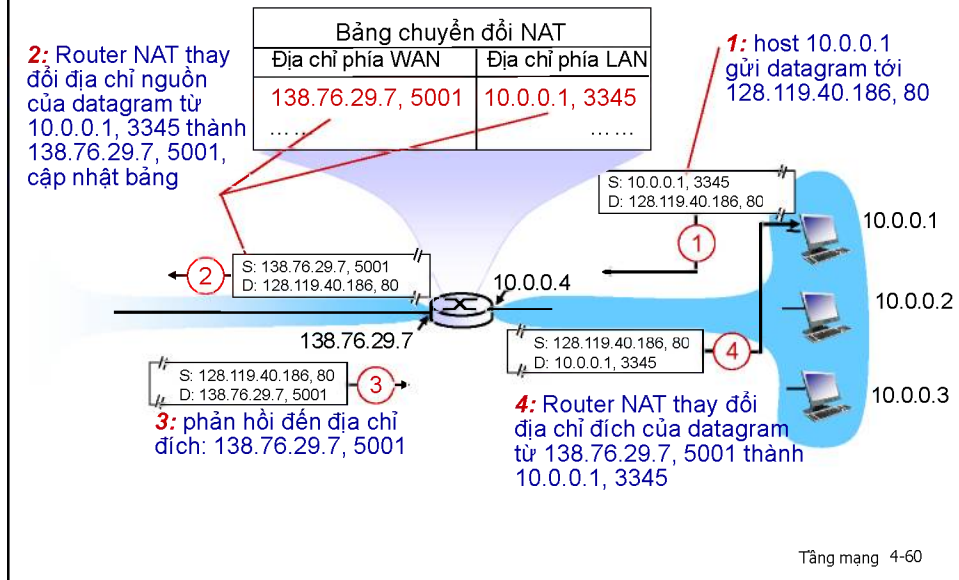
NAT: network address translation

Cài đặt: Router NAT phải:

- **Các datagram đi ra: thay thế** (địa chỉ IP nguồn, số cổng) của mỗi datagram đi ra ngoài thành (địa chỉ IP NAT, số cổng mới) . . . các client/server ở xa sẽ dùng (địa chỉ IP NAT, số cổng mới) như là địa chỉ đích
- **Ghi nhớ (trong bảng chuyển đổi NAT)** mọi cặp chuyển đổi (địa chỉ IP nguồn, số cổng) thành (địa chỉ IP NAT, số cổng mới)
- **Các datagram đi đến: thay thế** (địa chỉ IP NAT, số cổng mới) trong trường địa chỉ đích của mọi datagram đi đến thành (địa chỉ IP nguồn, số cổng) tương ứng được lưu trong bảng NAT.

Tăng mạng 4-59

NAT: network address translation



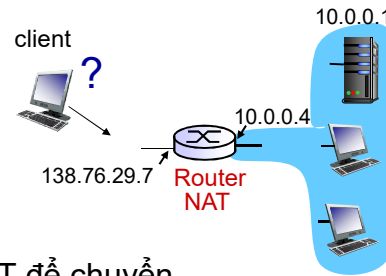
NAT: network address translation

- ❖ Trường số hiệu cổng gồm 16-bit :
 - 60.000 kết nối đồng thời chỉ với một địa chỉ phía LAN!
- ❖ NAT hiện vẫn còn đang gây tranh cãi
 - Các router chỉ nên xử lý đến tầng 3
 - Vi phạm thỏa thuận end-to-end
 - Các nhà thiết kế ứng dụng phải xem xét đến khả năng NAT, ví dụ ứng dụng P2P
 - Việc thiếu địa chỉ nên được thay bằng cách giải quyết là dùng IPv6

Vấn đề đi qua NAT

❖ Client muốn kết nối tới server có địa chỉ 10.0.0.1

- Địa chỉ 10.0.0.1 của server được đặt trong mạng LAN (client không thể sử dụng địa chỉ này là địa chỉ đích)
- Từ bên ngoài, client chỉ nhìn thấy địa chỉ NAT là 138.76.29.7



❖ **Giải pháp 1:** Cấu hình tĩnh NAT để chuyển tiếp các yêu cầu kết nối đến tới cổng đã xác định của server

- Ví dụ: (138.76.29.7, cổng 2500) sẽ luôn được chuyển tiếp tới (10.0.0.1, cổng 25000)

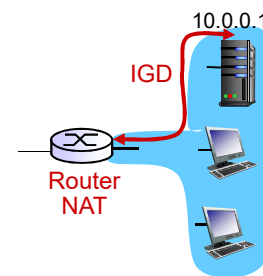
Tầng mạng 4-62

Vấn đề đi qua NAT

❖ **Giải pháp 2:** Dùng giao thức Universal Plug and Play (UPnP) Internet Gateway Device (IGD), cho phép chuyển đổi NAT:

- ❖ Ghi nhớ địa chỉ IP công khai (138.76.29.7)
- ❖ Thêm/xóa các ánh xạ cổng (trong khoảng thời gian cho phép)

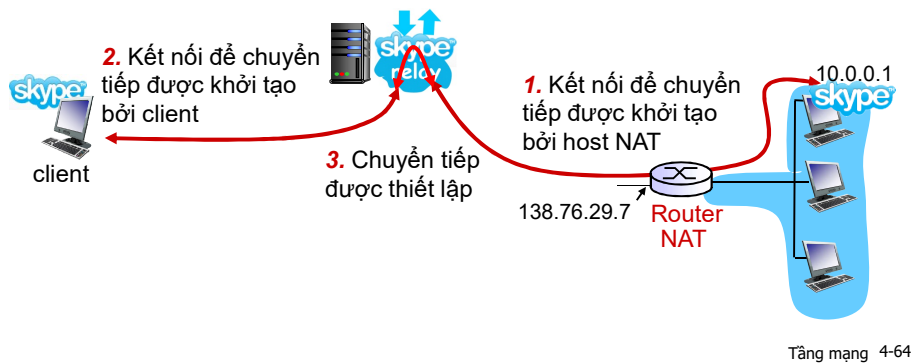
Ví dụ: Cấu hình ánh xạ cổng NAT tĩnh tự động



Tầng mạng 4-63

Vấn đề đi qua NAT

- ❖ **Giải pháp 3:** chuyển tiếp (được dùng trong Skype)
 - Client NAT thiết lập kết nối để chuyển tiếp
 - Client bên ngoài kết nối để chuyển tiếp
 - Chuyển tiếp giữa các gói tin của các cầu để kết nối



Chương 4: Nội dung

- 4.1 Giới thiệu
- 4.2 Các mạng mạch ảo và mạng chuyển gói
- 4.3 Kiến trúc của bộ định tuyến
- 4.4 IP: Internet Protocol
 - Định dạng gói tin
 - Định địa chỉ IPv4
 - ICMP
 - IPv6
- 4.5 Các giải thuật định tuyến
 - Link state
 - Distance vector
 - Hierarchical routing
- 4.6 Định tuyến trong mạng Internet
 - RIP
 - OSPF
 - BGP

ICMP: internet control message protocol

- Được sử dụng bởi các host & các router để truyền thông tin tầng mạng

- Báo cáo lỗi: không tìm được host, mạng, cổng, giao thức
- Phản hồi yêu cầu/đáp ứng (được dùng bởi ping)

- “Ở phía trên” trong tầng mạng:

- Các thông điệp ICMP được mang trong các IP datagram

- Thông điệp ICMP: type, code và 8 byte đầu tiên của IP datagram mô tả nguyên nhân lỗi

Type	Code	description (mô tả)
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Tầng mạng 4-66

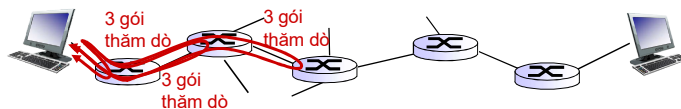
Traceroute và ICMP

- Nguồn gửi chuỗi UDP segments tới đích
 - Segment đầu tiên được thiết lập TTL=1
 - Segment thứ hai TTL=2, ...
 - Không giống với số hiệu cổng
- Khi datagram thứ n tới router n :
 - Router bỏ qua các datagram
 - Và gửi đến nguồn thông điệp ICMP (type 11, code 0)
 - Thông điệp ICMP có chứa tên của router & địa chỉ IP

- Khi thông điệp ICMP đến, nguồn tính toán các RTT

Điều kiện dừng:

- UDP segment cuối cùng đến được host đích.
- Đích trả lại thông điệp ICMP “port unreachable” (type 3, code 3) → cổng không có
- Nguồn dừng lại



Tầng mạng 4-67

IPv6: Lý do

- ❖ **Động lực thúc đẩy ban đầu:** không gian địa chỉ 32-bit sắp được cấp phát hết.
- ❖ Động lực bổ sung:
 - Định dạng tiêu đề (header) giúp tăng tốc độ xử lý/chuyển tiếp
 - Tiêu đề thay đổi giúp tạo điều kiện cho QoS

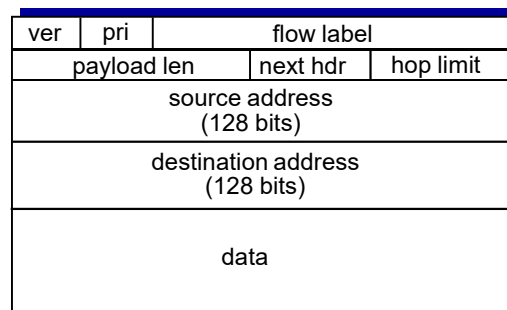
Định dạng IPv6 datagram:

- Phần tiêu đề có chiều dài cố định 40 byte
- Không cho phép phân mảnh gói tin

Tầng mạng 4-68

Định dạng IPv6 datagram

- ❖ **Priority (ưu tiên):** xác định ưu tiên giữa các datagram trong luồng
- ❖ **Flow Label (nhãn luồng):** xác định các datagram trong cùng một “luồng”.
- ❖ **Next header:** xác định giao thức tầng cao hơn cho dữ liệu



← 32 bits →

Tầng mạng 4-69

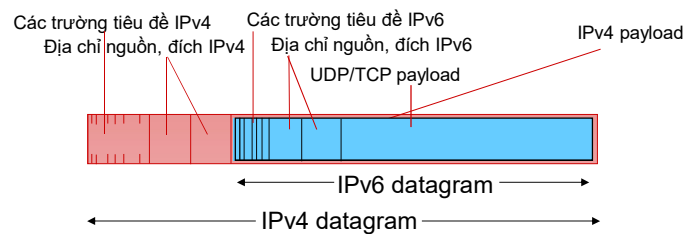
Những thay đổi của IPv6 so với IPv4

- ❖ **Checksum:** bỏ hoàn toàn, nhằm giảm thời gian xử lý tại mỗi hop
- ❖ **Options:** được phép, nhưng nằm ngoài phần tiêu đề, được xác định trong trường “Next Header”
- ❖ **ICMPv6:** phiên bản mới của ICMP
 - Các loại thông điệp bổ sung, ví dụ: “Packet Too Big”
 - Các chức năng quản lý nhóm multicast

Tầng mạng 4-70

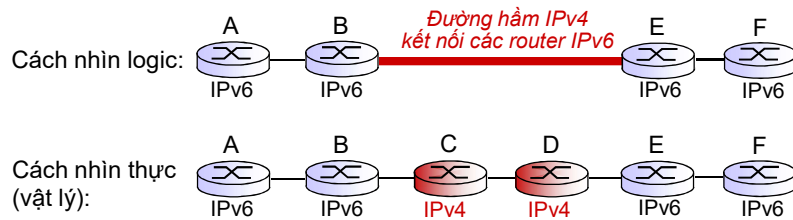
Chuyển đổi từ IPv4 sang IPv6

- ❖ Không phải tất cả các router đều có thể được nâng cấp đồng thời
 - Không có ngày dành riêng cho việc chuyển đổi (flag days)
 - Mạng sẽ hoạt động như thế nào với việc sử dụng đồng thời các router IPv4 và IPv6?
- ❖ **Tunneling (đường hầm):** Payload của IPv6 datagram được mang trong IPv4 datagram giữa các router IPv4



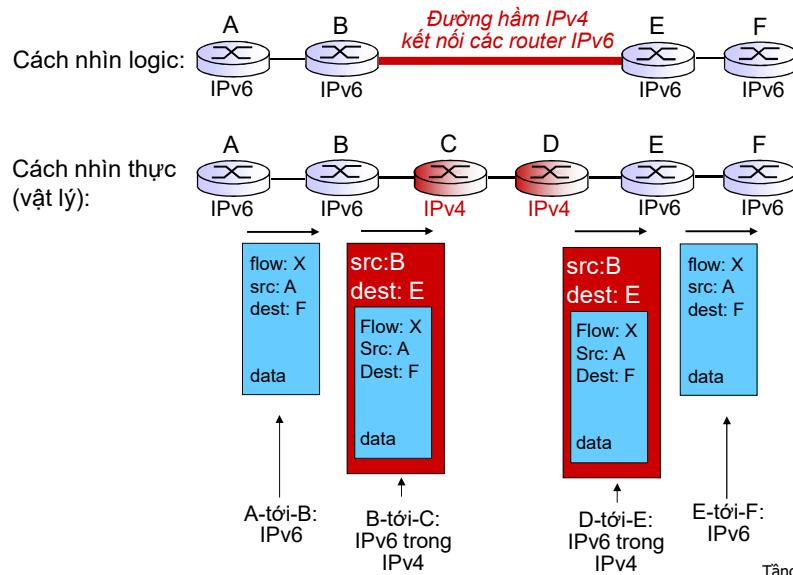
Tầng mạng 4-71

Tunneling



Tầng mạng 4-72

Tunneling



Tầng mạng 4-73

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tăng mạng 4-74

Tác động qua lại giữa định tuyến và chuyển tiếp

Thuật toán định tuyến

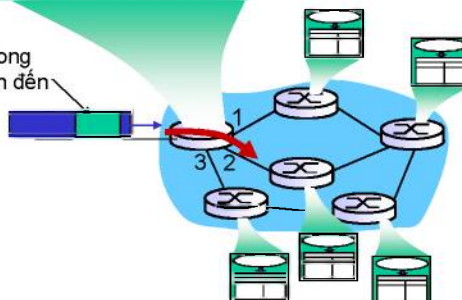
Bảng chuyển tiếp cục bộ

Địa chỉ đích	Liên kết ra
Dãy địa chỉ 1	3
Dãy địa chỉ 2	2
Dãy địa chỉ 3	2
Dãy địa chỉ 4	1

Thuật toán định tuyến xác định đường đi từ đầu cuối này đến đầu cuối kia trên mạng

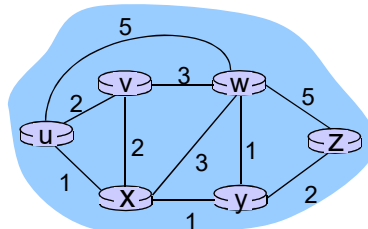
Bảng chuyển tiếp xác định việc chuyển tiếp cục bộ tại bộ định tuyến này

Địa chỉ IP đích trong tiêu đề của gói tin đến



Tăng mạng 4-75

Mô hình đồ thị



Đồ thị: $G = (N, E)$

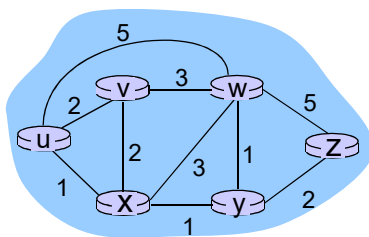
$N =$ Tập các router $= \{ u, v, w, x, y, z \}$

$E =$ Tập các liên kết $= \{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

Chú ý: Mô hình đồ thị cũng được dùng trong các ngữ cảnh mạng khác, như P2P, trong đó N là tập các peer và E là tập các kết nối TCP.

Tăng mạng 4-76

Mô hình đồ thị: Các chi phí



$c(x, x') =$ chi phí của kết nối (x, x')
ví dụ: $c(w, z) = 5$

Chi phí có thể luôn bằng 1, hoặc có thể liên quan đến băng thông, hoặc liên quan đến tắc nghẽn

Chi phí của đường đi $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

Hỏi: Chi phí thấp nhất đường đi từ u đến z là bao nhiêu?

Giải thuật định tuyến: giải thuật tìm đường đi có chi phí thấp nhất

Tăng mạng 4-77

Phân loại giải thuật định tuyến

Hỏi: Thông tin là tập trung hay không tập trung?

Tập trung:

- ❖ Tất cả các router đều có thông tin đầy đủ về cấu trúc mạng và chi phí của các liên kết
- ❖ Giải thuật "link state" (trạng thái kết nối)

Không tập trung:

- ❖ Router biết về các hàng xóm có kết nối vật lý với nó và chi phí liên kết tới các hàng xóm này.
- ❖ Lặp lại quá trình tính toán, trao đổi thông tin với các hàng xóm
- ❖ Giải thuật "distance vector" (véc-tơ khoảng cách)

Hỏi: Động hay tĩnh?

Tĩnh:

- ❖ Việc định tuyến thay đổi chậm theo thời gian

Động:

- ❖ Việc định tuyến thay đổi nhanh hơn:
 - Cập nhật định kỳ
 - Phản ứng với những thay đổi chi phí liên kết

Tăng mạng 4-78

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tăng mạng 4-79

Một giải thuật định tuyến Link-State

Giải thuật Dijkstra

- ❖ Tất cả các nút đều biết được cấu trúc mạng và chi phí của các liên kết trên mạng
 - Được thực hiện bằng cách “quảng bá trạng thái liên kết”
 - Tất cả các nút có thông tin giống nhau
- ❖ Tính toán chi phí thấp nhất đường đi từ một nút (“nguồn”) đến tất cả các nút khác.
 - Cho *bảng chuyển tiếp* của nút đó
- ❖ Lặp: sau k lần duyệt, sẽ biết được chi phí thấp nhất tới k đích

Ký hiệu:

- ❖ $c(x,y)$: chi phí liên kết từ nút x tới y ; $= \infty$ nếu không có kết nối trực tiếp đến nút lân cận
- ❖ $D(v)$: giá trị hiện tại của chi phí đường đi từ nguồn đến đích
- ❖ $p(v)$: nút trước nằm trên đường đi từ nguồn đến v
- ❖ N' : tập các nút mà chi phí đường đi thấp nhất đã được xác định

Tăng mạng 4-80

Giải thuật Dijkstra

1 Khởi tạo:

- 2 $N' = \{u\}$
- 3 for tất cả các nút v
- 4 if v kề với u
- 5 then $D(v) = c(u,v)$
- 6 else $D(v) = \infty$
- 7

8 Lặp

- 9 tìm w không thuộc N' mà $D(w)$ là nhỏ nhất
- 10 thêm w vào N'
- 11 Cập nhật $D(v)$ cho tất cả v kề với w và không thuộc N' :
- 12 $D(v) = \min(D(v), D(w) + c(w,v))$
- 13 /* chi phí mới đến v là chính giá trị cũ hoặc chi phí đường đi ngắn nhất đã tính tới w cộng với chi phí từ w tới v */
- 15 **cho đến khi tất cả các nút nằm trong N'**

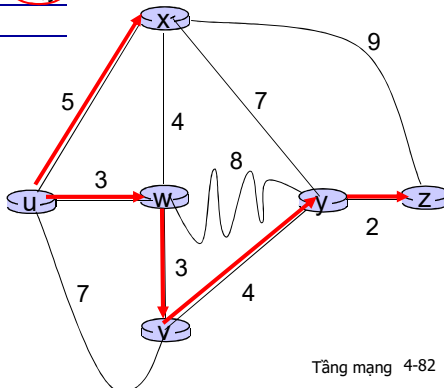
Tăng mạng 4-81

Giải thuật Dijkstra: Ví dụ

Bước	N'	D(v)	D(w)	D(x)	D(y)	D(z)
		p(v)	p(w)	p(x)	p(y)	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	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					

Chú ý:

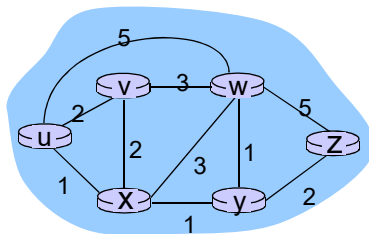
- ❖ Xây dựng cây đường đi ngắn nhất bằng cách đi tìm các nút trước.
- ❖ Các mối quan hệ có thể tồn tại (có thể được chia tùy ý)



Tăng mạng 4-82

Giải thuật Dijkstra: Một ví dụ khác

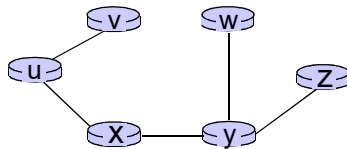
Bước	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Tăng mạng 4-83

Giải thuật Dijkstra: Ví dụ (2)

Kết quả cây đường đi ngắn nhất từ u:



Kết quả bảng chuyển tiếp trong u:

Đích	Liên kết
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Tăng mạng 4-84

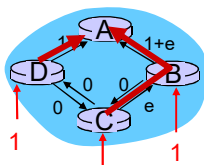
Giải thuật Dijkstra: Thảo luận

Độ phức tạp của giải thuật: n nút

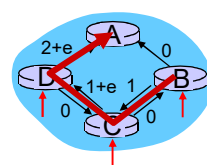
- ❖ Mỗi lần duyệt: cần kiểm tra lại tất cả các nút w không thuộc N
- ❖ $n(n+1)/2$ phép so sánh: $O(n^2)$
- ❖ Có thể cài đặt hiệu quả hơn: $O(n \log n)$

Có thể dao động:

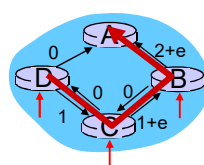
- ❖ Ví dụ: chi phí kết nối bằng tổng lưu lượng mạng:



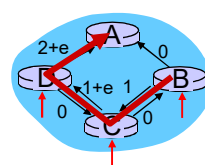
Khởi tạo



Với chi phí đã biết, tính toán định tuyến mới.... kết quả có chi phí mới



Với chi phí đã biết, tính toán định tuyến mới.... kết quả có chi phí mới



Với chi phí đã biết, tính toán định tuyến mới.... kết quả có chi phí mới

Tăng mạng 4-85

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tăng mạng 4-86

Giải thuật distance vector

Công thức Bellman-Ford

Cho

$d_x(y)$:= chi phí thấp nhất đường đi từ x đến y
thì

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

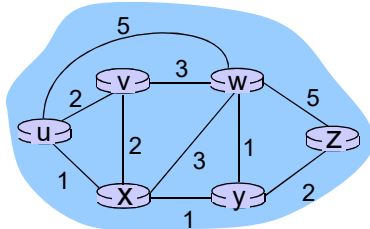
\min được tính trên tất cả các lân cận v của x

Chi phí tới lân cận v

Chi phí từ lân cận v tới đích y

Tăng mạng 4-87

Ví dụ Bellman-Ford



Có: $d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

Công thức B-F cho:

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

Giá trị nhỏ nhất đạt được của nút là hop kế tiếp trong đường đi ngắn nhất, được dùng trong bảng chuyển tiếp.

Tăng mạng 4-88

Giải thuật distance vector

- ❖ $D_x(y)$ = ước lượng chi phí thấp nhất từ x đến y
 - x duy trì véc-tơ khoảng cách $D_x = [D_x(y): y \in N]$
- ❖ Nút x:
 - Biết chi phí đến mỗi nút lân cận v: $c(x,v)$
 - Duy trì véc-tơ khoảng cách của các nút lân cận của. Với mỗi nút lân cận v, x duy trì $D_v = [D_v(y): y \in N]$

Tăng mạng 4-89

Giải thuật distance vector

Ý tưởng chính:

- ❖ Mỗi nút định kỳ gửi ước lượng véc-tơ khoảng cách của nó đến các nút lân cận
- ❖ Khi x nhận được ước lượng DV mới từ nút lân cận, nó sẽ cập nhật DV của nó bằng cách dùng công thức B-F:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ với mỗi nút } y \in N$$

- ❖ Dưới các điều kiện tự nhiên, ước lượng $D_x(y)$ sẽ hội tụ về giá trị chi phí thấp nhất thực tế $d_x(y)$

Tăng mạng 4-90

Giải thuật distance vector

Lặp, không đồng bộ: mỗi lần lặp cục bộ có nguyên nhân từ:

- ❖ Thay đổi chi phí liên kết cục bộ
- ❖ Thông báo cập nhật DV từ nút lân cận

Phân bố:

- ❖ Mỗi nút *chỉ* thông báo đến các nút lân cận khi DV của nó thay đổi
 - Các lân cận sau đó sẽ thông báo đến các lân cận khác của nó nếu cần

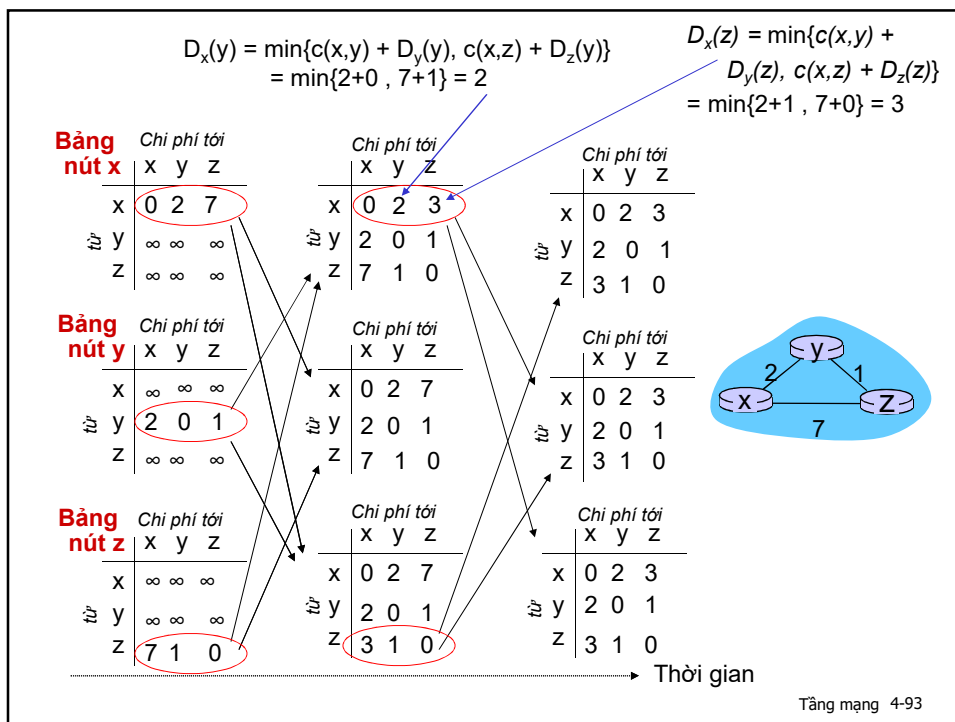
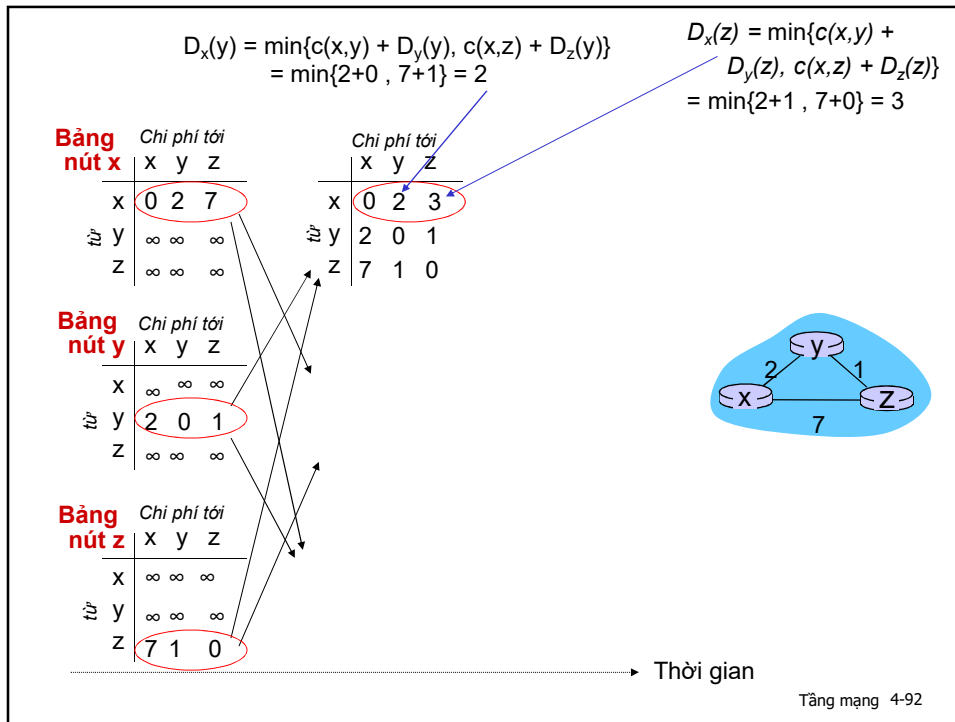
Mỗi nút:

Chờ cho (thay đổi trong chi phí liên kết cục bộ hoặc thông báo từ nút lân cận)

Tính toán lại các ước lượng

Nếu DV tới bất kỳ đích nào có thay đổi, thì **thông báo** cho các nút lân cận

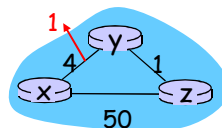
Tăng mạng 4-91



Distance vector: Chi phí kết nối thay đổi

Chi phí kết nối thay đổi:

- ❖ Nút kiểm tra thay đổi chi phí kết nối cục bộ
- ❖ Cập nhật thông tin tìm đường, tính toán lại véc-tơ khoảng cách
- ❖ Nếu DV thay đổi, thông báo đến các lân cận



“thông tin tốt truyền đi nhanh”

t_0 : y kiểm tra thay đổi chi phí kết nối, cập nhật lại DV của nó, và thông báo cho các lân cận.

t_1 : z nhận thông tin cập nhật từ y, cập nhật lại bảng của nó, tính toán chi phí thấp nhất mới tới x, và gửi DV của nó tới các lân cận.

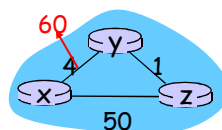
t_2 : y nhận cập nhật của z, cập nhật lại bảng khoảng cách của nó. Chi phí thấp nhất của y *không* thay đổi, nên y *không cần* gửi thông báo tới z.

Tăng mạng 4-94

Distance vector: Chi phí kết nối thay đổi

Chi phí kết nối thay đổi:

- ❖ Nút kiểm tra thay đổi chi phí cục bộ
- ❖ **Tin xấu truyền đi chậm** – vấn đề “đếm vô hạn”!
- ❖ 44 lần duyệt trước khi thuật toán ổn định: (xem thêm trong tài liệu)



poisoned reverse:

- ❖ Nếu Z định tuyến Y đi thẳng tới X :
 - Z thông báo cho Y khoảng cách (của Z) tới X là vô hạn (để Y không đi tới X qua Z)
- ❖ Sẽ giải quyết triệt để vấn đề đếm vô hạn?

Tăng mạng 4-95

So sánh các giải thuật LS và DV

Độ phức tạp thông báo

- ❖ **LS:** với n nút, E liên kết, $O(nE)$ thông báo được gửi đi
- ❖ **DV:** chỉ trao đổi giữa các nút lân cận
 - Thời gian hội tụ thay đổi

Tốc độ hội tụ

- ❖ **LS:** Thuật toán $O(n^2)$ yêu cầu $O(nE)$ thông báo
 - Có thể có dao động
- ❖ **DV:** Thời gian hội tụ thay đổi
 - Có thể lập định tuyến
 - Vấn đề đếm vô hạn

Sự linh hoạt: điều gì sẽ xảy ra nếu router hoạt động sai chức năng?

LS:

- Nút có thể thông báo chi phí *kết nối* không chính xác
- Mỗi nút chỉ tính toán bảng *riêng* của nó

DV:

- Nút có thể thông báo chi phí *đường đi* không chính xác
- Mỗi bảng của nút được dùng bởi các nút khác
 - Lỗi lan truyền thông qua mạng

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tầng mạng 4-97

Hierarchical routing (Định tuyến phân cấp)

Những vấn đề định tuyến được học cho đến lúc này là với môi trường lý tưởng hóa

- ❖ Tất cả các bộ định tuyến là đồng nhất
 - ❖ Mạng “phẳng”
- ... *không* đúng trong thực tế!

Quy mô: với 600 triệu

đích:

- ❖ Không thể lưu tất cả các đích trong các bảng định tuyến!
- ❖ Việc trao đổi bảng định tuyến sẽ làm tràn các liên kết!

Tự quản

- ❖ Internet = mạng của các mạng
- ❖ Mỗi nhà quản trị mạng có thể muốn điều hành định tuyến riêng trong mạng của họ

Tầng mạng 4-98

Hierarchical routing

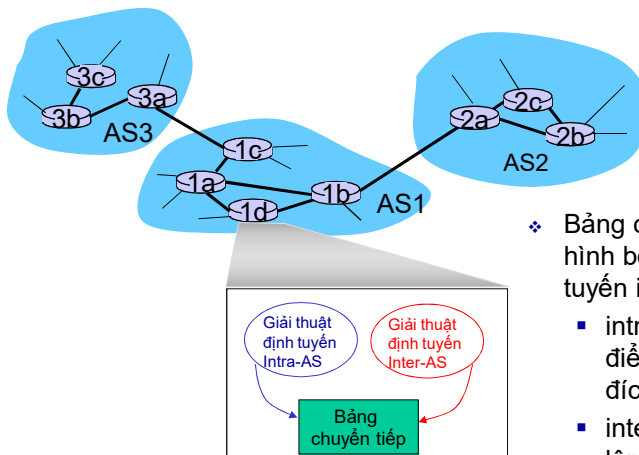
- ❖ Các router được tập hợp lại thành các vùng, “**hệ thống tự trị**” (autonomous systems - AS)
- ❖ Các router trong cùng AS sẽ chạy cùng giao thức định tuyến
 - Giao thức định tuyến “**nội vùng-AS**” (intra-AS)
 - Các router trong các AS khác nhau có thể chạy các giao thức định tuyến intra-AS khác nhau

Gateway router:

- ❖ Tại “**cạnh**” của AS riêng của nó
- ❖ Có liên kết tới router trong AS khác

Tầng mạng 4-99

Kết nối các AS



- ❖ Bảng chuyển tiếp được cấu hình bởi cả giải thuật định tuyến intra- và inter-AS
 - intra-AS thiết lập các điểm đăng nhập cho các đích nội mạng
 - inter-AS & intra-AS thiết lập các điểm đăng nhập cho các đích ngoại mạng

Tầng mạng 4-100

Nhiệm vụ của Inter-AS

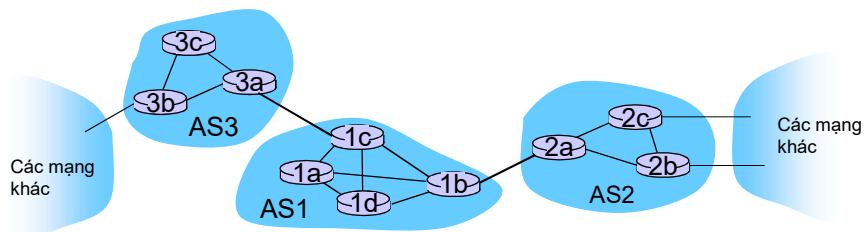
- ❖ Giả sử router trong AS1 nhận datagram có đích ở bên ngoài AS1:

- Router nên chuyển tiếp gói tin đến gateway router, nhưng mà là cái nào?

AS1 phải:

1. Học xem có thể đến được đích nào qua AS2, và AS3
2. Lan truyền thông tin này đến tất cả các router trong AS1

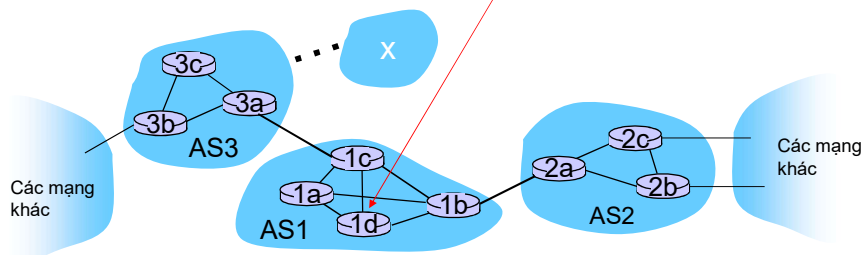
Đây là nhiệm vụ của định tuyến inter-AS!



Tầng mạng 4-101

Ví dụ: thiết lập bảng chuyển tiếp trong router 1d

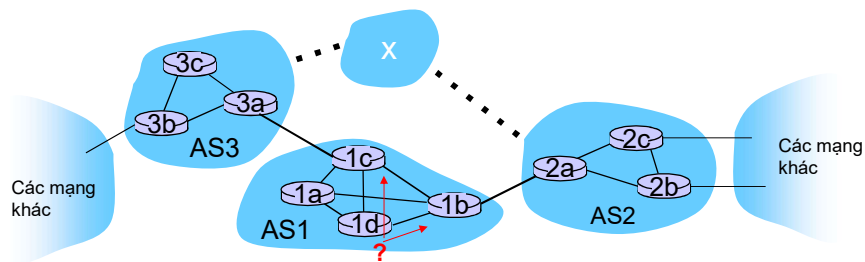
- ❖ Giả sử AS1 học (qua giao thức inter-AS) được là subnet x có thể đến được qua AS3 (gateway 1c), nhưng không qua AS2
 - Giao thức inter-AS lan truyền thông tin đi được cho tất cả các router nội mạng
- ❖ Router 1d biết được từ thông tin định tuyến intra-AS là giao diện l của nó thuộc đường đi có chi phí thấp nhất tới 1c
 - Đưa giá trị (x, l) vào bảng chuyển tiếp



Tầng mạng 4-102

Ví dụ: lựa chọn giữa nhiều AS

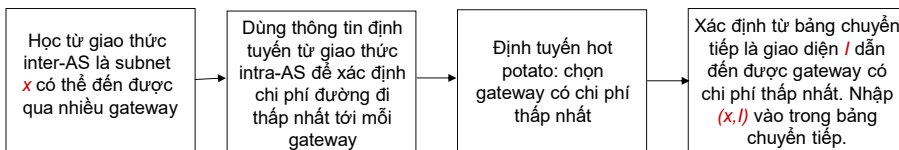
- ❖ Bây giờ, giả sử AS1 học từ giao thức inter-AS là subnet **x** có thể đến được từ AS3 và từ AS2.
- ❖ Để cấu hình bảng chuyển tiếp, router 1d cần phải xác định gateway nào mà nó nên chuyển tiếp các gói tin đến để tới được đích **x**
 - Đây là nhiệm vụ của giao thức định tuyến inter-AS!



Tầng mạng 4-103

Ví dụ: lựa chọn giữa nhiều AS

- ❖ Bây giờ, giả sử AS1 học từ giao thức inter-AS là subnet **x** có thể đến được từ AS3 và từ AS2.
- ❖ Để cấu hình bảng chuyển tiếp, router 1d cần phải xác định gateway nào mà nó nên chuyển tiếp các gói tin đến để tới được đích **x**
 - Đây là nhiệm vụ của giao thức định tuyến inter-AS!
- ❖ **Định tuyến hot potato: gửi** gói tin đến router gần nhất trong hai router



Tầng mạng 4-104

Chương 4: Nội dung

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của bộ định tuyến

4.4 IP: Internet Protocol

- Định dạng gói tin
- Định địa chỉ IPv4
- ICMP
- IPv6

4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

Tầng mạng 4-105

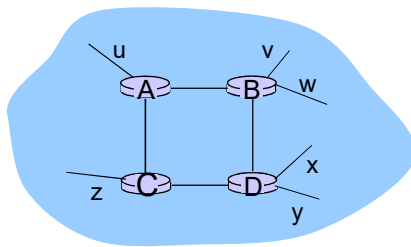
Định tuyến Intra-AS

- ❖ Còn được gọi là các giao thức cổng nội mạng (*interior gateway protocols - IGP*)
- ❖ Các giao thức định tuyến intra-AS phổ biến nhất:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IGRP: Interior Gateway Routing Protocol (Cisco độc quyền)

Tầng mạng 4-106

RIP (Routing Information Protocol)

- ❖ Được công bố trong BSD-UNIX distribution năm 1982
- ❖ Giải thuật distance vector
 - Độ đo khoảng cách: số hop (lớn nhất = 15 hop), mỗi liên kết có chi phí là 1
 - Các DV được trao đổi giữa các điểm lân cận sau mỗi 30s bằng một thông điệp đáp ứng (còn được gọi là **thông báo (advertisement)**)
 - Mỗi thông báo: danh sách lên đến 25 **subnet đích**

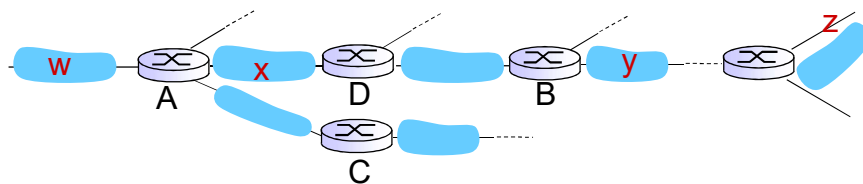


Từ router A đến các **subnet** đích:

subnet	hop
u	1
v	2
w	2
x	3
y	3
z	2

Tăng mạng 4-107

RIP: Ví dụ

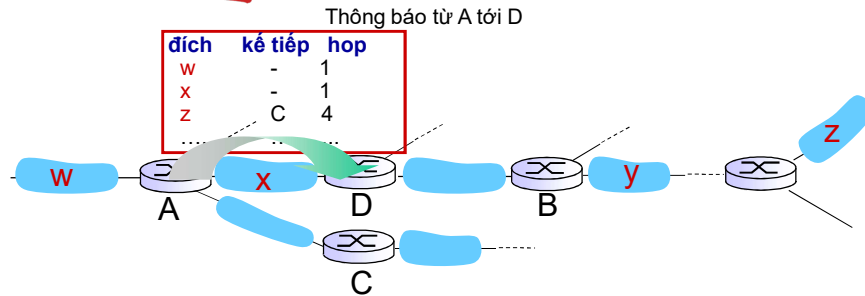


Bảng định tuyến trong router D

Subnet đích	Router kế tiếp	Số hop đến đích
w	A	2
y	B	2
z	B	7
x	--	1
....

Tăng mạng 4-108

RIP: Ví dụ



Bảng định tuyến trong router D

Subnet đích	Router kế tiếp	Số hop đến đích
w	A	2
y	B	2
z	B A	7 5
x	--	1
....

Tăng mạng 4-109

RIP: Lỗi liên kết và khôi phục

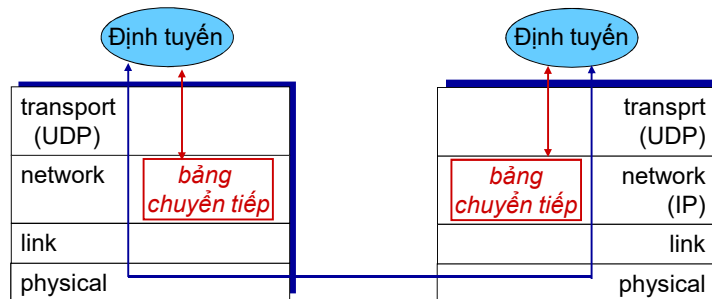
Nếu không thấy có thông báo sau khoảng 180s thì lân cận/liên kết được coi là “đã chết”.

- Các tuyến đường qua lân cận là không còn dùng được
- Các thông báo mới được gửi tới các lân cận
- Các lân cận tiếp tục gửi các thông báo mới (nếu các bảng bị thay đổi)
- Thông báo lỗi liên kết lan truyền nhanh chóng (?) trên toàn bộ mạng
- *Poison reverse* được dùng để ngăn chặn các vòng lặp ping-pong (khoảng cách vô hạn = 16 hop)

Tăng mạng 4-110

Xử lý bảng RIP

- ❖ Các bảng định tuyến RIP được quản lý bởi tiến trình *tầng ứng dụng* được gọi là route-d (daemon)
- ❖ Các thông báo được gửi trong các gói tin UDP, lặp lại định kỳ



Tầng mạng 4-111

OSPF (Open Shortest Path First)

- ❖ “Mở”: sẵn sàng công khai
- ❖ Dùng giải thuật link state
 - Phân phối gói LS
 - Bản đồ cấu trúc mạng tại mỗi nút
 - Tính toán đường đi dùng giải thuật Dijkstra
- ❖ Thông báo OSPF mang một điểm truy nhập vào mỗi lân cận
- ❖ Các thông báo được phân phối đến *toàn bộ* AS (qua cơ chế flooding)
 - Các thông điệp OSPF được mang trực tiếp trên IP (chứ không phải là TCP hay UDP)
- ❖ *Giao thức định tuyến IS-IS*: gần giống với OSPF

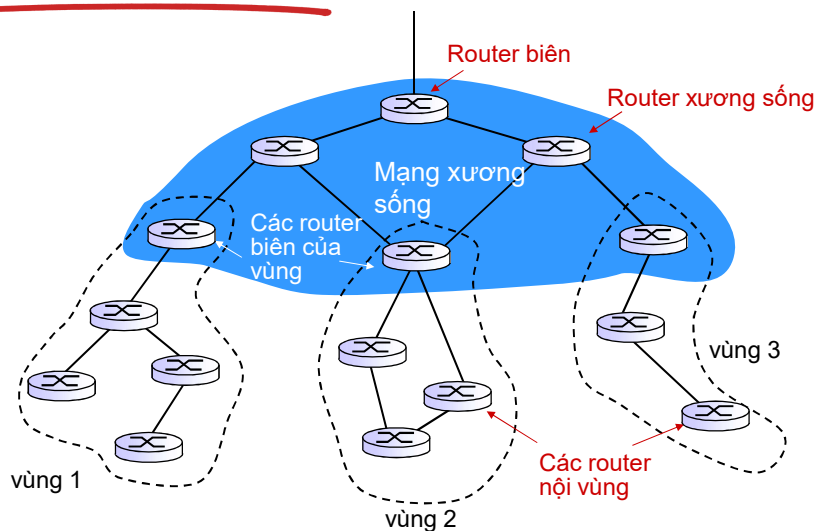
Tầng mạng 4-112

Các đặc tính “cải tiến” trong OSPF (không có trong RIP)

- ❖ **Bảo mật:** Tất cả các thông điệp OSPF đều được chứng thực (để ngăn chặn những xâm nhập xấu)
- ❖ Cho phép có **nhiều tuyến đường đi với cùng chi phí** (trong RIP chỉ có một)
- ❖ Với mỗi liên kết, có nhiều độ đo chi phí cho các **TOS** khác nhau. (Ví dụ: chi phí liên kết vệ tinh được thiết lập “thấp” để đạt hiệu quả tốt; “cao” cho thời gian thực).
- ❖ Hỗ trợ tích hợp uni- và **multicast**:
 - Multicast OSPF (MOSPF) dùng cơ sở dữ liệu cùng cấu trúc như OSPF
- ❖ OSPF **phân cấp** trong các miền lớn.

Tăng mạng 4-113

OSPF phân cấp



Tăng mạng 4-114

OSPF phân cấp

- ❖ **Phân cấp 2 mức:** vùng cục bộ, vùng xương sống.
 - Chỉ dùng thông báo link-state bên trong vùng
 - Mỗi nút có cấu trúc vùng chi tiết; chỉ biết hướng (đường đi ngắn nhất) đến các mạng trong các vùng khác.
- ❖ **Các router biên của vùng:** “tổng hợp” khoảng cách đến các mạng trong vùng của nó, thông báo tới các router biên của vùng khác.
- ❖ **Các router xương sống:** chạy định tuyến OSPF hạn chế đến mạng xương sống
- ❖ **Các router biên:** kết nối tới các router biên của các AS khác.

Tăng mạng 4-115

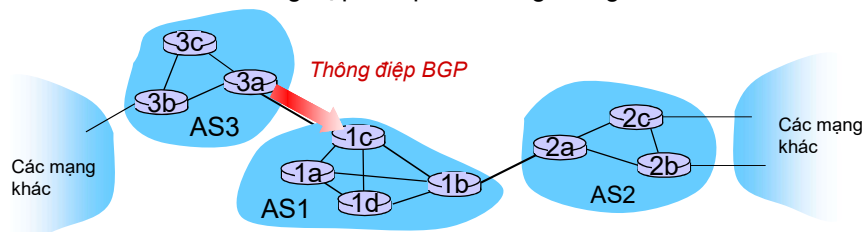
Định tuyến inter-AS trên Internet: BGP

- ❖ **BGP (Border Gateway Protocol):** Giao thức định tuyến liên miền thực tế
 - “gắn kết mọi người lại với nhau trên Internet”
- ❖ BGP cung cấp cho mỗi AS:
 - **eBGP:** lấy thông tin đi đến subnet từ các AS lân cận.
 - **iBGP:** lan truyền thông tin đến tất cả các router bên trong AS.
 - Xác định đường đi “tốt” tới các mạng khác dựa trên thông tin đường đi và chính sách
- ❖ Cho phép subnet thông báo sự tồn tại của nó đến phần còn lại của Internet.

Tăng mạng 4-116

Các cơ sở của BGP

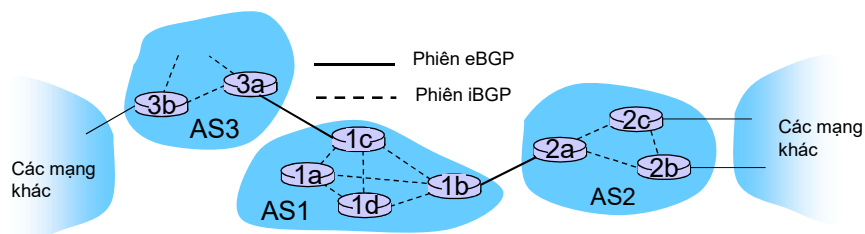
- ❖ **Phiên BGP:** Hai router BGP (“các peer”) trao đổi các thông điệp BGP:
 - Thông báo **đường đi** tới các tiền tố (prefix) mạng đích khác nhau (giao thức “path vector” (véc-tơ đường))
 - Được trao đổi qua các kết nối TCP bán bền vững
- ❖ Khi AS3 thông báo một prefix đến AS1:
 - AS3 **hứa hẹn** nó sẽ chuyển tiếp các datagram hướng tới prefix đó
 - AS3 có thể tổng hợp các prefix trong thông báo của mình



Tầng mạng 4-117

Các cơ sở của BGP: phân phối thông tin đường đi

- ❖ Dùng phiên eBGP giữa 3a và 1c, AS3 gửi thông tin đường đi (prefix) cho AS1.
 - Tiếp theo, 1c có thể dùng iBGP để phân phối thông tin prefix mới cho tất cả các router trong AS1
 - Sau đó, 1b có thể thông báo thông tin đường đi mới tới AS2 qua phiên eBGP từ 1b-đến-2a.
- ❖ Khi router học được prefix mới, nó sẽ tạo ra điểm truy nhập cho prefix trong bảng chuyển tiếp của nó.



Tầng mạng 4-118

Các thuộc tính đường và định tuyến BGP

- ❖ Prefix được thông báo chứa các thuộc tính BGP
 - Prefix + các thuộc tính = “định tuyến”
- ❖ Hai thuộc tính quan trọng:
 - **AS-PATH**: chứa các AS qua đó thông báo prefix nào được truyền. Ví dụ: AS 67, AS 17
 - **NEXT-HOP**: xác định router AS nội vùng nào là AS kế tiếp. (Có thể có nhiều liên kết từ AS hiện tại tới AS kế tiếp).
- ❖ Gateway router nhận thông báo định tuyến bằng cách dùng **import policy (chính sách nhập)** để chấp nhận/từ chối
 - Ví dụ: không bao giờ định tuyến qua AS x
 - Định tuyến *dựa trên chính sách*.

Tăng mạng 4-119

Chọn định tuyến BGP

- ❖ Router có thể học được nhiều đường đi đến AS đích, và việc chọn tuyến đường được dựa trên:
 1. Thuộc tính giá trị ưu tiên cục bộ: quyết định chính sách
 2. AS-PATH ngắn nhất
 3. Router NEXT-HOP gần nhất: định tuyến hot potato
 4. Tiêu chuẩn bổ sung

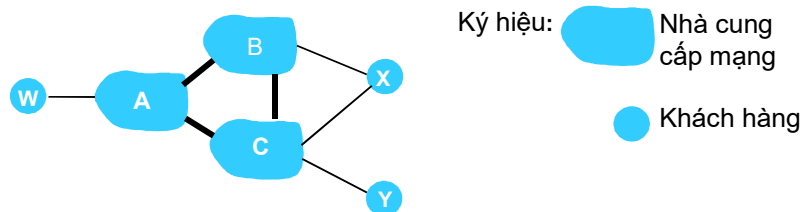
Tăng mạng 4-120

Các thông điệp BGP

- ❖ Các thông điệp BGP được trao đổi giữa các peer qua kết nối TCP
- ❖ Các thông điệp BGP:
 - **OPEN**: Mở kết nối TCP tới peer và xác thực bên gửi
 - **UPDATE**: thông báo đường đi mới (hoặc xóa bỏ đường cũ)
 - **KEEPALIVE**: giữ kết nối tồn tại khi UPDATES thiếu; cũng có thể yêu cầu ACKs OPEN
 - **NOTIFICATION**: báo cáo lỗi trong thông điệp trước; cũng được dùng để đóng kết nối.

Tăng mạng 4-121

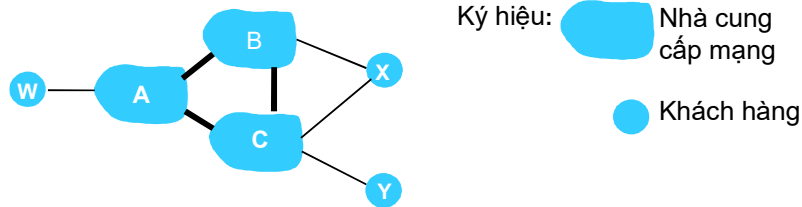
Chính sách định tuyến BGP



- ❖ A, B, C là *các nhà cung cấp mạng*
- ❖ X, W, Y là khách hàng (của nhà cung cấp mạng)
- ❖ X là *dual-homed*: được gắn vào hai mạng
 - X không muốn định tuyến từ B đến C qua X
 - ... do vậy, X sẽ không thông báo tới B về đường đi đến C

Tăng mạng 4-122

Chính sách định tuyến BGP



- ❖ A thông báo đường đi AW đến B
- ❖ B thông báo đường đi BAW đến X
- ❖ B sẽ thông báo đường đi BAW đến C?
 - Không! B không nhận “thu thập” cho định tuyến CBAW vì W và C đều không phải là khách hàng của B
 - B muốn buộc C phải định tuyến tới w qua A
 - B *chỉ* muốn định tuyến từ/tới khách hàng của nó!

Tăng mạng 4-123

Tại sao định tuyến Intra-, Inter-AS khác nhau?

Chính sách:

- ❖ Inter-AS: nhà quản trị muốn điều hành định tuyến lưu lượng và ai định tuyến qua mạng của họ.
- ❖ Intra-AS: Quản trị riêng, vì vậy không cần các quyết định chính sách

Quy mô:

- ❖ Định tuyến phân cấp tiết kiệm kích thước bảng, giảm lưu lượng cập nhật

Hiệu năng:

- ❖ Intra-AS: có thể tập trung vào hiệu năng
- ❖ Inter-AS: chính sách quan trọng hơn hiệu suất

Tăng mạng 4-124

Chương 4: Hoàn thành!

4.1 Giới thiệu

4.2 Các mạng mạch ảo và mạng chuyển gói

4.3 Kiến trúc của router

4.4 IP: Internet Protocol

- Định dạng datagram, định địa chỉ IPv4, ICMP, IPv6

❖ Hiểu được các nguyên lý bên trong các dịch vụ tầng mạng:

- Các mô hình dịch vụ tầng mạng, tác động qua lại giữa định tuyến và chuyển tiếp, cách router hoạt động, định tuyến (chọn đường).

❖ Cài đặt, hiện thực trên mạng Internet

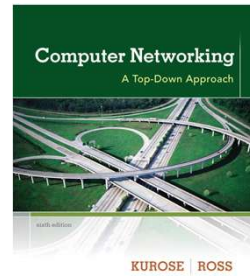
4.5 Các giải thuật định tuyến

- link state, distance vector, hierarchical routing

4.6 Định tuyến trên Internet

- RIP, OSPF, BGP

Chương 5 Tầng liên kết



*Computer
Networking: A
Top Down
Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012*

Người dịch: Nguyễn Thanh Thủy

Tài liệu được dịch cho mục đích giảng dạy (được sự đồng ý của tác giả).

© All material copyright 1996-2012
J.F. Kurose and K.W. Ross, All Rights Reserved

Tầng liên kết 5-1

Chương 5: Tầng liên kết

Mục tiêu:

- ❖ Hiểu được các nguyên lý của các dịch vụ tầng liên kết
 - Phát hiện và sửa lỗi
 - Chia sẻ kênh truyền chung (broadcast channel): đa truy nhập
 - Định địa chỉ tầng liên kết
 - Các mạng cục bộ: Ethernet, VLANs
- ❖ Cài đặt và hiện thực các công nghệ tầng mạng khác nhau

Tầng liên kết 5-2

Tầng liên kết và các mạng LAN: Nội dung

5.1 Giới thiệu, các dịch vụ

5.2 Phát hiện và sửa lỗi

5.3 Các giao thức đa truy nhập

5.4 Các mạng LAN

- Định địa chỉ, ARP
- Ethernet
- Các switch
- Các VLAN

5.5 Chuyển mạch nhãn đa giao thức (MPLS)

5.6 Mạng trung tâm dữ liệu

5.7 Vòng đời của một yêu cầu web

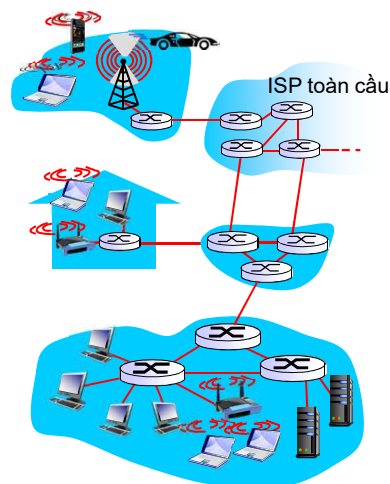
Tầng liên kết 5-3

Tầng liên kết: giới thiệu

Thuật ngữ:

- ❖ Các host và các router: **các nút mạng (node)**
- ❖ Các kênh truyền thông kết nối giữa các nút lân cận theo đường truyền thông: gọi là các liên kết (hay các kết nối, **link**)
 - Các liên kết có dây
 - Các liên kết không dây
 - Các LAN
- ❖ Gói tin tầng 2: **khung (frame)**, đóng gói datagram

Tầng liên kết dữ liệu có trách nhiệm truyền datagram từ một nút đến nút **vật lý lân cận** qua một liên kết



Tầng liên kết 5-4

Tầng liên kết: ngữ cảnh

- ❖ Datagram được truyền bởi các giao thức liên kết khác nhau qua các liên kết khác nhau:
 - Ví dụ: Ethernet trên liên kết thứ nhất, frame relay trên các liên kết trung gian, 802.11 trên liên kết cuối cùng.
 - ❖ Mỗi giao thức liên kết cung cấp các dịch vụ khác nhau.
 - Ví dụ: có thể hoặc không cung cấp truyền tin cậy (rdt) qua liên kết
- Tương tự giao thông:**
- ❖ Chuyển đi từ Princeton tới Lausanne
 - Ô tô: Princeton tới JFK
 - Máy bay: JFK tới Geneva
 - Tàu điện: Geneva tới Lausanne
 - ❖ Khách du lịch = **datagram**
 - ❖ Đoạn đường đi = **liên kết truyền thông**
 - ❖ Kiểu vận chuyển = **Giao thức tầng giao vận**
 - ❖ Đại lý du lịch = **Giải thuật định tuyến**

Tầng liên kết 5-5

Các dịch vụ tầng liên kết

- ❖ **Tạo khung dữ liệu, truy nhập liên kết**
 - Đóng gói datagram vào trong frame, thêm phần tiêu đề (header), phần đuôi (trailer)
 - Truy nhập kênh truyền nếu được chia sẻ
 - Các địa chỉ “MAC” được sử dụng trong các tiêu đề của khung để xác định địa chỉ nguồn, đích
 - Khác với địa chỉ IP!
- ❖ **Truyền tin cậy giữa các nút lân cận**
 - Đã được học (trong chương 3)!
 - Ít khi được dùng trên liên kết có tỷ lệ lỗi thấp (cáp quang, một số loại cáp xoắn)
 - Các liên kết không dây: tỷ lệ lỗi cao
 - **Hỏi:** Tại sao cần truyền tin cậy trên cả mức liên kết và mức đầu cuối-đến-đầu cuối?

Tầng liên kết 5-6

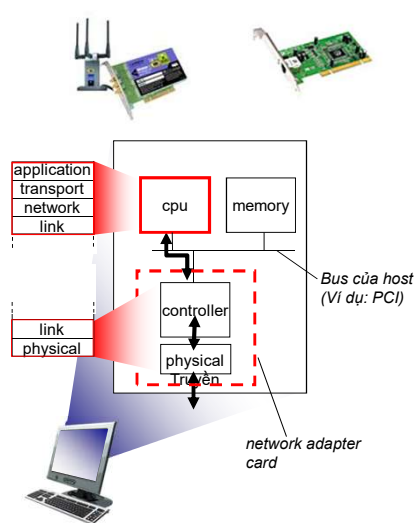
Các dịch vụ tầng liên kết (tiếp)

- ❖ **Điều khiển luồng**
 - Điều khiển tốc độ giữa các nút gửi và nhận kề nhau
- ❖ **Phát hiện lỗi**
 - Lỗi là do suy giảm tín hiệu, nhiễu
 - Bên nhận phát hiện ra sự xuất hiện của các lỗi:
 - Thông báo cho bên gửi truyền lại hoặc loại bỏ frame đó
- ❖ **Sửa lỗi**
 - Bên nhận xác định **và sửa** các lỗi bit mà không cần phải yêu cầu truyền lại
- ❖ **Bán song công (half-duplex) và song công (full-duplex)**
 - Với bán song công, cả hai đầu cuối của liên kết đều có thể truyền, nhưng không được truyền tại cùng một thời điểm.

Tầng liên kết 5-7

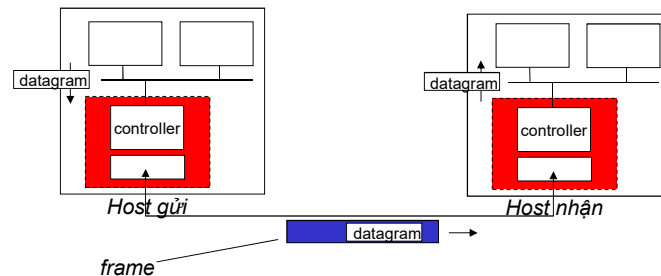
Tầng liên kết được cài đặt ở đâu?

- ❖ Tại tất cả các host
- ❖ Tầng liên kết được cài đặt tại “adaptor” (còn được gọi là **thẻ giao diện mạng (network interface card - NIC)** hoặc trên chip
 - Ethernet card, 802.11 card; Ethernet chipset
 - Cài đặt tầng liên kết và tầng vật lý
- ❖ Gắn vào bên trong các bus hệ thống của host
- ❖ Kết hợp phần cứng, phần mềm, phần sụn (firmware)



Tầng liên kết 5-8

Các adaptor truyền thông



- ❖ Phía gửi:
 - Đóng gói datagram trong frame
 - Bổ sung kiểm tra lỗi bit, rdt, điều khiển luồng,...
- ❖ Phía nhận:
 - Kiểm tra lỗi, rdt, điều khiển luồng,...
 - Trích xuất datagram, chuyển lên tầng cao hơn tại phía nhận

Tầng liên kết 5-9

Tầng liên kết và các mạng LAN: Nội dung

- 5.1 Giới thiệu, các dịch vụ
- 5.2 Phát hiện và sửa lỗi
- 5.3 Các giao thức đa truy nhập
- 5.4 Các mạng LAN
 - Định địa chỉ, ARP
 - Ethernet
 - Các switch
 - Các VLAN
- 5.5 Chuyển mạch nhãn đa giao thức (MPLS)
- 5.6 Mạng trung tâm dữ liệu
- 5.7 Vòng đời của một yêu cầu web

Tầng liên kết 5-10

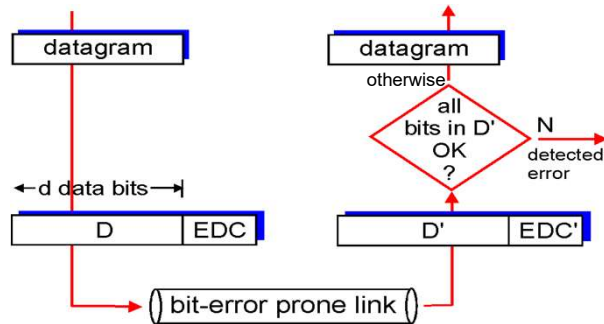
Phát hiện lỗi

EDC= Các bit dùng để phát hiện và sửa lỗi (Error Detection and Correction bits) (dư thừa)

D = Dữ liệu được bảo vệ bằng cách kiểm tra lỗi, có thể bao gồm cả các trường trong phần tiêu đề.

Phát hiện lỗi không thể đảm bảo tin cậy 100%!

- Giao thức có thể bỏ lỡ một vài lỗi, nhưng rất hiếm khi
- Trường EDC càng lớn thì càng tốt hơn cho việc phát hiện và sửa lỗi.

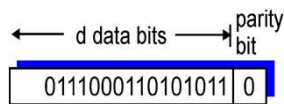


Tăng liên kết 5-11

Kiểm tra Parity

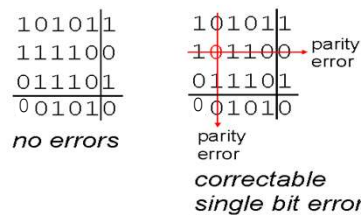
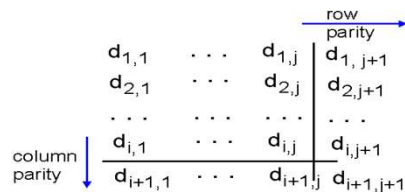
Bit parity đơn:

- ❖ Phát hiện các lỗi bit đơn



Bit parity hai chiều:

- ❖ Phát hiện và sửa các lỗi bit đơn



Tăng liên kết 5-12

Internet checksum (xem lại)

Mục tiêu: phát hiện “các lỗi” (ví dụ: các bit bị đảo ngược) trong gói tin được truyền (chú ý: *chỉ* được dùng tại tầng giao vận).

Bên gửi:

- ❖ Xử lý các nội dung segment như là chuỗi các số nguyên 16-bit
- ❖ checksum: bổ sung (tổng bù 1) vào nội dung của segment
- ❖ Bên gửi đặt giá trị checksum vào trong trường UDP checksum

Bên nhận:

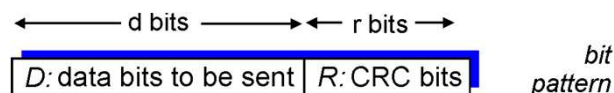
- ❖ Tính toán checksum của segment nhận được
- ❖ Kiểm tra xem checksum đã được tính có bằng giá trị của trường checksum hay không:
 - KHÔNG – có phát hiện lỗi
 - Có – không phát hiện ra lỗi. *Nhưng có thể có những lỗi khác?*

Tầng liên kết 5-13

Kiểm tra dư thừa theo chu kỳ

(Cyclic redundancy check - CRC)

- ❖ Có nhiều tiềm năng phát hiện lỗi hơn
- ❖ Coi các bit dữ liệu **D** như là số nhị phân
- ❖ Chọn mẫu **G** có $r+1$ bit
- ❖ Mục tiêu: chọn r bit CRC, **R**, như sau:
 - $\langle D, R \rangle$ chia hết cho G (theo mô đun 2)
 - Bên nhận biết G , chia $\langle D, R \rangle$ cho G . Nếu số dư khác 0: phát hiện lỗi!
 - Có thể phát hiện tất cả các lỗi nhỏ hơn $r+1$ bit
- ❖ Được sử dụng phổ biến trong thực tế (Ethernet, 802.11 WiFi, ATM)



$$D * 2^r \text{ XOR } R \quad \text{mathematical formula}$$

Tầng liên kết 5-14

Ví dụ CRC

Muốn:

$$D \cdot 2^r \text{ XOR } R = nG$$

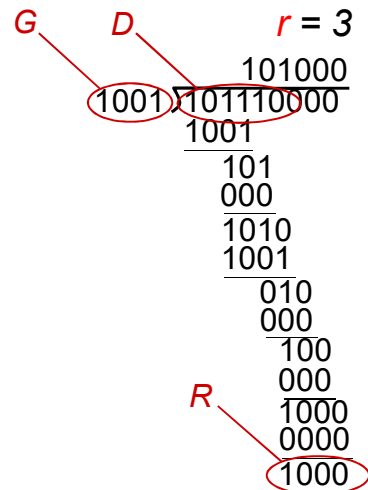
Tương đương:

$$D \cdot 2^r = nG \text{ XOR } R$$

Tương đương:

Nếu lấy G chia cho $D \cdot 2^r$,
muốn phần còn lại R
thỏa mãn:

$$R = \text{phần dư của } \left[\frac{D \cdot 2^r}{G} \right]$$



Tầng liên kết 5-15

Tầng liên kết và các mạng LAN: Nội dung

5.1 Giới thiệu, các dịch vụ

5.2 Phát hiện và sửa lỗi

5.3 Các giao thức đa truy
nhập

5.4 Các mạng LAN

- Định địa chỉ, ARP
- Ethernet
- Các switch
- Các VLAN

5.5 Chuyển mạch nhãn đa
giao thức (MPLS)

5.6 Mạng trung tâm dữ liệu

5.7 Vòng đời của một yêu
cầu web

Tầng liên kết 5-16

Tầng liên kết và các mạng LAN: Nội dung

5.1 Giới thiệu, các dịch vụ

5.2 Phát hiện và sửa lỗi

5.3 Các giao thức đa truy nhập

5.4 Các mạng LAN

- Định địa chỉ, ARP
- Ethernet
- Các switch
- Các VLAN

5.5 Chuyển mạch nhãn đa giao thức (MPLS)

5.6 Mạng trung tâm dữ liệu

5.7 Vòng đời của một yêu cầu web

Tầng liên kết 5-16

Các giao thức và các liên kết đa truy nhập

Có hai loại “liên kết”:

❖ Điểm-nối-điểm (Point-to-point)

- PPP cho truy nhập dial-up
- Liên kết point-to-point giữa các host, switch Ethernet

❖ **Quảng bá (broadcast) (chia sẻ đường truyền chung)**

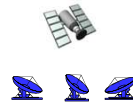
- Ethernet mô hình cũ
- upstream HFC
- 802.11 wireless LAN (LAN không dây)



Chia sẻ đường truyền
(Ví dụ: cabled Ethernet)



Chia sẻ RF
(Ví dụ: 802.11 WiFi)



Chia sẻ RF
(vệ tinh)



Con người tại bữa tiệc cocktail (chia sẻ không khí, âm thanh)

Tầng liên kết 5-17

Các giao thức đa truy nhập

- ❖ Kênh quảng bá (broadcast) được chia sẻ
- ❖ Hai hoặc nhiều nút muốn truyền: giao thoa
 - *Tranh chấp (đụng độ, collision)* xảy ra khi nút nhận được hai hay nhiều tín hiệu tại cùng một thời điểm.

Giao thức đa truy nhập

- ❖ Giải thuật phân quyền xác định cách các nút chia sẻ kênh truyền, ví dụ: xác định khi nào nút có thể được truyền.
- ❖ Truyền thông về chia sẻ kênh phải dùng chính kênh đó!
 - Không có kênh riêng để điều phối

Tăng liên kết 5-18

Một giao thức đa truy nhập lý tưởng

Cho: Kênh quảng bá có tốc độ R bps

Mong muốn:

1. Khi một nút muốn truyền, nó có thể gửi đi với tốc độ R .
2. Khi M nút muốn truyền, mỗi nút có thể gửi đi với tốc độ trung bình là R/M .
3. Phân quyền hoàn toàn:
 - Không có nút đặc biệt cho việc điều phối truyền
 - Không có các khe (slot) hay đồng hồ đồng bộ
4. Đơn giản

Tăng liên kết 5-19

Các giao thức MAC: Phân loại

Gồm 3 loại chính:

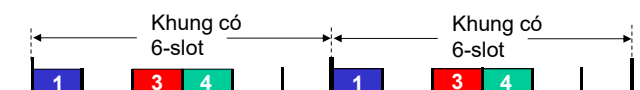
- ❖ **Phân chia kênh**
 - Chia kênh thành các “phần” nhỏ hơn (khe thời gian, tần số, mã)
 - Cấp phát các phần cho các nút để dùng riêng
- ❖ **Truy nhập ngẫu nhiên**
 - Kênh không được phân chia, cho phép tranh chấp
 - “Giải quyết” các tranh chấp
- ❖ **“Xoay vòng”**
 - Các nút lần lượt xoay vòng, nhưng nút gửi nhiều hơn được nắm quyền truyền lâu hơn.

Tăng liên kết 5-20

Giao thức MAC phân chia kênh: TDMA

TDMA: Đa truy nhập phân chia theo thời gian (time division multiple access)

- ❖ Truy nhập tới kênh theo “các vòng”
- ❖ Mỗi trạm có một khe thời gian (slot) có độ dài cố định (độ dài = thời gian truyền gói) trong mỗi vòng
- ❖ Không được dùng các khe đang “rảnh” (không hoạt động - idle)
- ❖ Ví dụ: LAN có 6 trạm, các trạm 1,3,4 có các gói tin, các khe 2,5,6 đang rảnh.

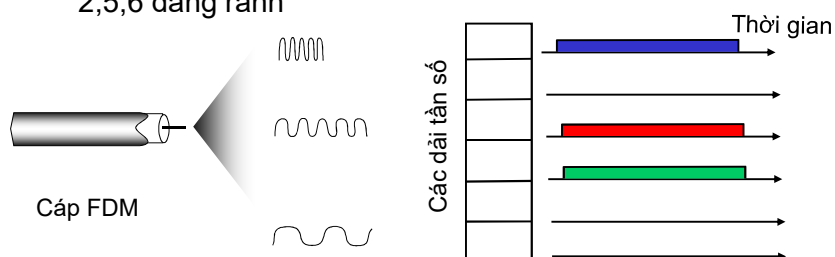


Tăng liên kết 5-21

Giao thức MAC phân chia kênh: FDMA

FDMA: Đa truy nhập phân chia theo tần số (frequency division multiple access)

- ❖ Phổ kênh truyền được chia theo các dải tần số
- ❖ Mỗi trạm được gán một dải tần số cố định
- ❖ Trong lúc truyền không được dùng các dải tần số “rảnh” khác.
- ❖ Ví dụ: LAN có 6 trạm, các trạm 1,3,4 có gói tin, các dải tần 2,5,6 đang rảnh



Tăng liên kết 5-22

Giao thức truy nhập ngẫu nhiên

- ❖ Khi nút có gói cần gửi đi
 - Truyền dữ liệu với tốc độ của kênh truyền R.
 - Không có điều phối *ưu tiên* giữa các nút
- ❖ Khi hai hoặc nhiều nút truyền → “tranh chấp”,
- ❖ **Giao thức MAC truy nhập ngẫu nhiên** xác định:
 - Cách phát hiện ra tranh chấp
 - Cách giải quyết các tranh chấp (Ví dụ: thông qua truyền lại)
- ❖ Các ví dụ của giao thức MAC truy nhập ngẫu nhiên:
 - Chia slot ALOHA
 - ALOHA thuần nhất
 - CSMA, CSMA/CD, CSMA/CA

Tăng liên kết 5-23

Chia Slot ALOHA

Giả thiết:

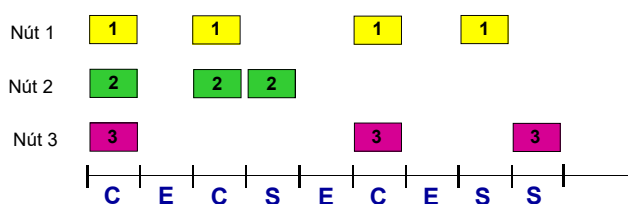
- ❖ Tất cả các frame có cùng kích thước
- ❖ Thời gian được chia thành các slot có kích thước bằng nhau (thời gian đủ để truyền 1 frame)
- ❖ Các nút bắt đầu truyền chỉ khi slot bắt đầu
- ❖ Các nút được đồng bộ hóa
- ❖ Nếu có 2 hoặc nhiều nút truyền trong một slot, thì tất cả các nút đều phát hiện có tranh chấp.

Hoạt động:

- ❖ Khi nút có một khung mới, nó được phép truyền trong slot tiếp theo.
 - *Nếu không có tranh chấp:* nút có thể gửi khung mới trong slot kế tiếp
 - *Nếu có tranh chấp:* nút truyền lại frame trong mỗi slot kế tiếp với xác suất bằng p cho đến khi thành công

Tăng liên kết 5-24

Chia Slot ALOHA



Ưu điểm:

- ❖ Nút kích hoạt có thể truyền liên tục với tốc độ tối đa của kênh.
- ❖ Phân quyền cao: chỉ các slot trong các nút cần được đồng bộ
- ❖ Đơn giản

Nhược điểm:

- ❖ Có tranh chấp,
- ❖ Lãng phí các slot không hoạt động
- ❖ Các nút có thể phát hiện tranh chấp với thời gian ít hơn truyền gói
- ❖ Cần đồng hồ đồng bộ hóa

Tăng liên kết 5-25

Chia slot ALOHA: Hiệu suất

Hiệu suất: là phần slot truyền thành công trong số nhiều frame cần truyền của nhiều nút.

- ❖ *Giả sử:* N nút với nhiều frame cần truyền, mỗi cuộc truyền trong slot có xác suất là p .
- ❖ Xác suất để một nút truyền trong một slot thành công là $p(1-p)^{N-1}$
- ❖ Xác suất để một nút *bất kỳ* truyền thành công là $Np(1-p)^{N-1}$

- ❖ Để tìm hiệu suất tối đa: tìm p^* để $Np(1-p)^{N-1}$ đạt giá trị lớn nhất
- ❖ Với nhiều nút, tìm giới hạn của $Np^*(1-p^*)^{N-1}$ khi N tiến đến vô cùng:

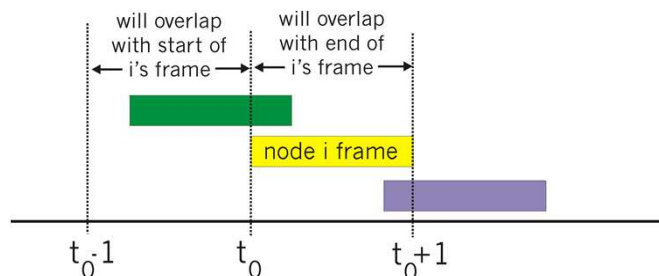
Hiệu suất tối đa = $1/e = .37$

Tốt nhất: kênh hữu dụng truyền trong khoảng 37% thời gian! **!**

Tăng liên kết 5-26

ALOHA thuần nhất (không chia slot)

- ❖ Aloha không chia slot: đơn giản hơn, không đồng bộ
- ❖ Khi frame đầu tiên đi đến
 - Truyền đi ngay lập tức
- ❖ Khả năng tranh chấp tăng lên:
 - Frame gửi tại thời điểm t_0 xung đột với các frame khác được gửi trong thời điểm $[t_0-1, t_0+1]$



Tăng liên kết 5-27

Hiệu suất của ALOHA thuần nhất

$P(\text{thành công của một nút}) = P(\text{nút truyền}) \cdot P(\text{không có nút nào khác truyền trong } [t_0-1, t_0])$

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$
$$= p \cdot (1-p)^{2(N-1)}$$

... chọn p tối ưu sau đó cho $n \rightarrow \infty$

$$= 1/(2e) = .18$$

Thậm chí hiệu suất còn kém hơn chia slot Aloha!

Tăng liên kết 5-28

Đa truy nhập sóng mang CSMA (carrier sense multiple access)

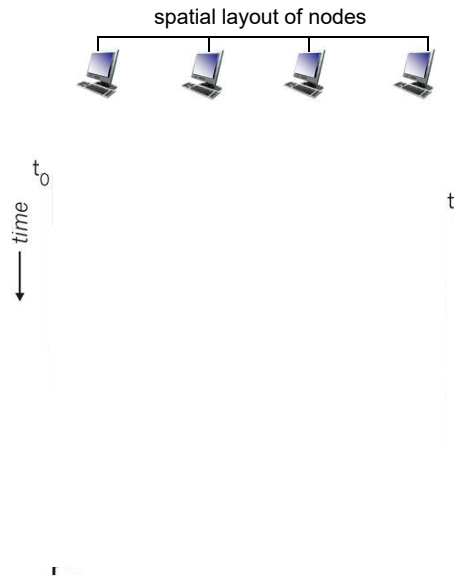
CSMA: nghe trước khi truyền:

- ❖ Nếu kênh truyền đang rảnh: truyền toàn bộ frame
- ❖ Nếu kênh truyền đang bận, trì hoãn việc truyền
- ❖ Tương tự với giao tiếp của con người: không ngắt lời khi người khác đang nói!

Tăng liên kết 5-29

Các tranh chấp trong CSMA

- ❖ **Tranh chấp vẫn có thể xảy ra:** do trễ lan truyền, nghĩa là hai nút không “nghe” thấy việc truyền của nhau.
- ❖ **Tranh chấp:** toàn bộ thời gian truyền gói tin bị lãng phí
 - Chú ý vai trò của khoảng cách và trễ lan truyền trong việc xác định khả năng có tranh chấp.



Tăng liên kết 5-30

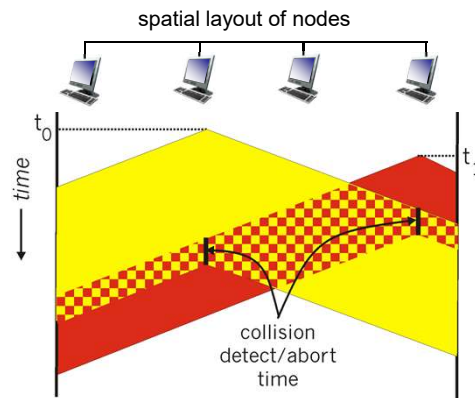
Đa truy nhập sóng mang có phát hiện tranh chấp CSMA/CD (collision detection)

CSMA/CD: trì hoãn như trong CSMA

- Tranh chấp *được phát hiện* trong thời gian ngắn
- Tranh chấp đường truyền được bỏ qua, giảm sự lãng phí kênh truyền
- ❖ Phát hiện tranh chấp:
 - Dễ dàng trong các mạng LAN không dây: đo cường độ tín hiệu, so sánh với các tín hiệu đã truyền, đã nhận.
 - Khó khăn trong các mạng LAN có dây: cường độ tín hiệu nhận được bị áp đảo bởi cường độ truyền cục bộ
- ❖ Tương tự với con người: đàm thoại lịch sự

Tăng liên kết 5-31

Đa truy nhập sóng mang có phát hiện tranh chấp CSMA/CD (collision detection)



Tăng liên kết 5-32

Giải thuật CSMA/CD trong Ethernet

1. NIC nhận datagram từ tầng mạng, tạo ra frame
2. Nếu NIC nhận thấy kênh truyền đang rảnh, sẽ bắt đầu truyền frame. Nếu NIC nhận thấy kênh truyền đang bận, sẽ đợi cho đến khi kênh truyền rảnh thì mới truyền.
3. Nếu NIC truyền toàn bộ frame mà không phát hiện thấy bất kỳ cuộc truyền nào khác, thì NIC sẽ hoàn thành việc truyền frame!
4. Nếu NIC phát hiện thấy có cuộc truyền khác trong khi đang truyền thì sẽ hủy bỏ và không gửi tín hiệu nữa
5. Sau khi hủy bỏ, NIC thực hiện **quay trở lại theo cơ chế (mũ) nhị phân:**
 - Sau tranh chấp thứ m , NIC chọn K ngẫu nhiên trong $\{0, 1, 2, \dots, 2^m - 1\}$. NIC chờ trong khoảng thời gian $K \cdot 512$ bit, quay trở lại bước 2
 - Khoảng thời gian chờ quay trở lại sẽ lâu hơn nếu có nhiều tranh chấp hơn.

Tăng liên kết 5-33

Hiệu suất CSMA/CD

- ❖ T_{prop} = trễ lan truyền lớn nhất giữa 2 nút trong LAN
- ❖ t_{trans} = thời gian truyền frame có kích thước lớn nhất

$$HiệuSuất = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- ❖ Hiệu suất sẽ tiến đến 1
 - khi t_{prop} tiến đến 0
 - khi t_{trans} tiến đến vô cùng
- ❖ Hiệu suất tốt hơn ALOHA: và đơn giản, chi phí thấp và phân quyền!

Tăng liên kết 5-34

Giao thức MAC “xoay vòng”

Các giao thức MAC phân chia kênh:

- Chia sẻ kênh *hiệu quả* và *công bằng* với tải cao
- Không hiệu quả với tải thấp: trễ trong việc tiếp cận kênh, băng thông được cấp phát bằng $1/N$ ngay cả khi chỉ có 1 nút cần truyền!

Các giao thức MAC truy nhập ngẫu nhiên:

- Hiệu quả với tải thấp: một nút có thể dùng hết khả năng của kênh
- Với tải cao: có tranh chấp.

Các giao thức MAC “xoay vòng”

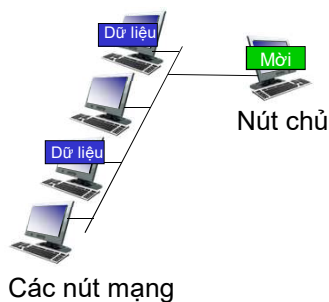
Tìm kiếm giải pháp tốt nhất!

Tăng liên kết 5-35

Giao thức MAC “xoay vòng”

Mời tuần tự:

- ❖ Nút chủ “mời” các nút truyền theo lượt tuần tự.
- ❖ Liên quan:
 - Việc mời truyền
 - Độ trễ
 - Một điểm chịu lỗi (Nút chủ)

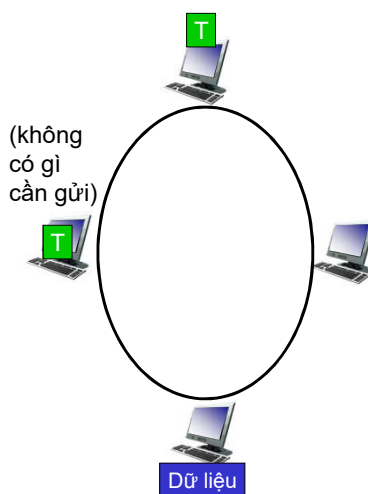


Tăng liên kết 5-36

Giao thức MAC “xoay vòng”

Chuyển thẻ bài (token):

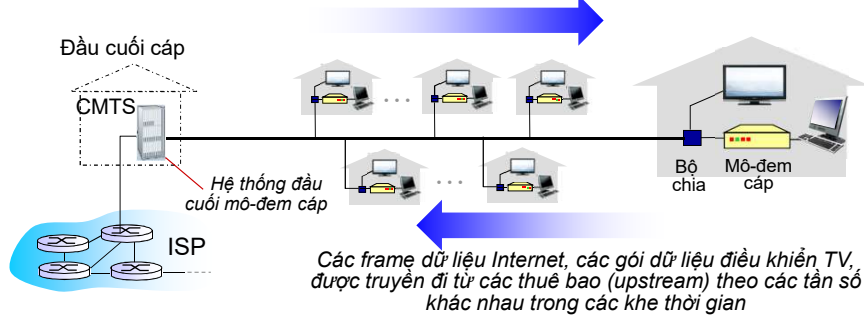
- ❖ Điều khiển *thẻ bài* chuyển tuần tự từ một nút đến nút kế tiếp.
- ❖ Thông điệp thẻ bài
- ❖ Liên quan:
 - Thẻ bài
 - Độ trễ
 - Một điểm chịu lỗi (thẻ bài)



Tăng liên kết 5-37

Mạng truy nhập cáp

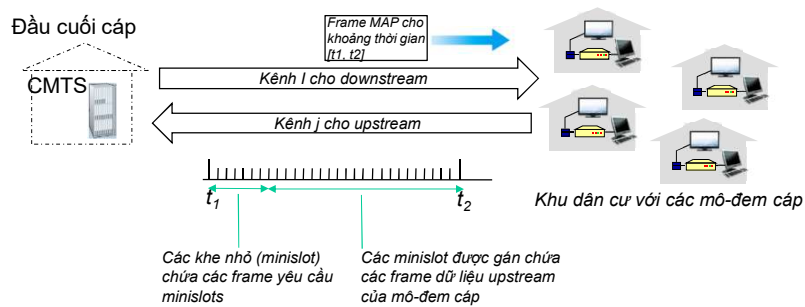
Các frame Internet, các kênh TV hay gói dữ liệu điều khiển được truyền đến thuê bao (downstream) theo các tần số khác nhau



- ❖ **Nhiều** kênh downstream (broadcast) 40 Mbps
 - Từ CMTS đơn truyền vào trong các kênh
- ❖ **Nhiều** kênh upstream 30 Mbps
 - **Đa truy nhập:** Tất cả người dùng đều có thể tranh kênh upstream nào đó trong các khe thời gian (mà những người khác đã được gán).

Tăng liên kết 5-38

Mạng truy nhập cáp



DOCSIS: chuẩn dữ liệu tốc độ cao cho hệ thống cáp (data over cable service interface specifications).

- ❖ FDM trên các kênh tần số upstream, downstream
- ❖ TDM upstream: một số slot được gán, một số có tranh chấp
 - Các khung MAP downstream: gán các slot upstream
 - Yêu cầu cho các slot upstream (và dữ liệu) được truyền truy cập ngẫu nhiên trong các khe thời gian đã được chọn

Tăng liên kết 5-39

Tổng kết về các giao thức MAC

- ❖ *Phân chia kênh*, theo thời gian, tần số hoặc mã
 - TDMA, FDMA
- ❖ *Truy nhập ngẫu nhiên* (động),
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - Sóng mang: dễ dàng trong một số công nghệ (có dây), khó khăn trong một số khác (không dây)
 - CSMA/CD được dùng trong Ethernet
 - CSMA/CA được dùng trong 802.11
- ❖ *Xoay vòng*
 - Trạm trung tâm mời các trạm truyền, chuyển thẻ bài
 - Bluetooth, FDDI, token ring

Tăng liên kết 5-40

Tầng liên kết và các mạng LAN: Nội dung

5.1 Giới thiệu, các dịch vụ

5.2 Phát hiện và sửa lỗi

5.3 Các giao thức đa truy
nhập

5.4 Các mạng LAN

- Định địa chỉ, ARP
- Ethernet
- Các switch
- Các VLAN

5.5 Chuyển mạch nhãn đa
giao thức (MPLS)

5.6 Mạng trung tâm dữ liệu

5.7 Vòng đời của một yêu
cầu web

Tầng liên kết 5-41

Địa chỉ MAC và ARP

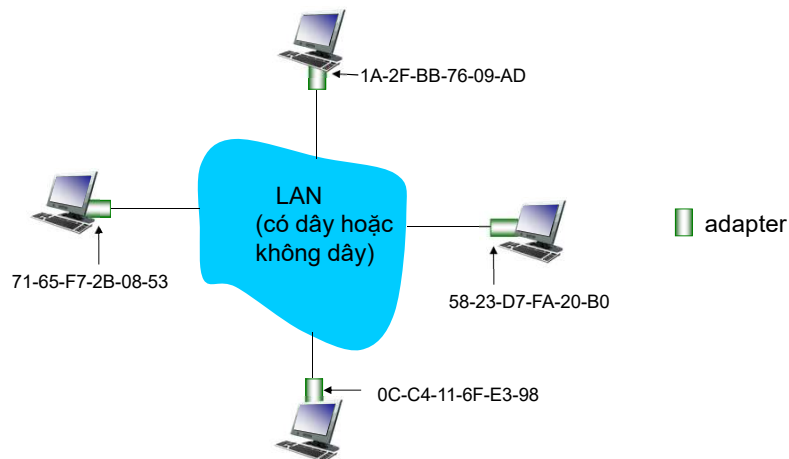
- ❖ Địa chỉ IP 32-bit:
 - Địa chỉ *tầng mạng* cho giao diện
 - Được dùng cho việc chuyển tiếp gói tin tại tầng 3 (tầng mạng)
- ❖ Địa chỉ MAC (hoặc LAN/vật lý/Ethernet):
 - Chức năng: *được dùng “cục bộ” để lấy frame từ một giao diện với một giao diện được kết nối vật lý khác (cùng mạng)*
 - Địa chỉ MAC có 48 bit (cho hầu hết các LAN) được ghi sẵn trong bộ nhớ ROM của NIC, (đôi khi cũng được thiết lập bởi phần mềm)
 - Ví dụ: 1A-2F-BB-76-09-AD

Ký hiệu trong hệ cơ số 16
(mỗi “số” biểu diễn cho 4 bit)

Tầng liên kết 5-42

Địa chỉ LAN và ARP

Mỗi adapter trên LAN có duy nhất một địa chỉ **LAN**



Tăng liên kết 5-43

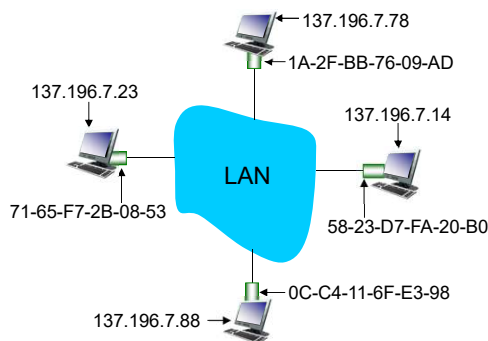
Địa chỉ LAN (tiếp)

- ❖ Việc cấp phát địa chỉ MAC được quản lý bởi IEEE
- ❖ Nhà sản xuất mua phần không gian địa chỉ MAC (để đảm bảo là duy nhất)
- ❖ So sánh:
 - Địa chỉ MAC: như số chứng minh nhân dân
 - Địa chỉ IP: như số điện thoại
- ❖ Địa chỉ MAC phẳng → có thể di chuyển
 - Có thể chuyển card từ LAN này sang LAN khác
- ❖ Địa chỉ phân cấp IP *không thể* di chuyển
 - Địa chỉ IP phụ thuộc vào IP subnet mà nút được gắn vào

Tăng liên kết 5-44

ARP: address resolution protocol

Hỏi: Làm thế nào để xác định địa chỉ MAC của một giao diện khi biết địa chỉ IP?



Bảng ARP: mỗi nút IP (host, router) trên LAN có một bảng ARP.

- Ánh xạ địa chỉ IP/MAC cho một số nút LAN:
< địa chỉ IP; địa chỉ MAC; TTL >
- TTL (Time To Live): thời gian sau đó ánh xạ địa chỉ sẽ bị hủy (thường là 20 phút)

Tăng liên kết 5-45

Giao thức ARP: cùng LAN

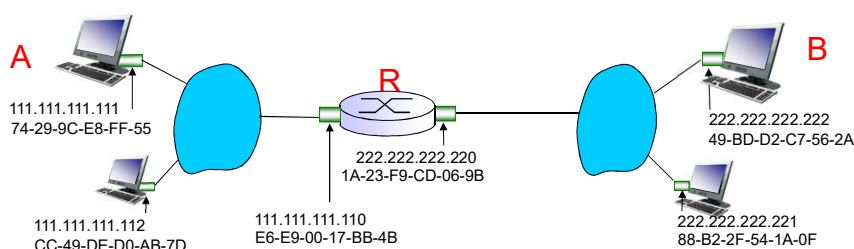
- ❖ A muốn gửi datagram tới B
 - Địa chỉ MAC của B không có trong bảng ARP của A.
- ❖ A **quảng bá (broadcasts)** gói tin truy vấn ARP, chứa địa chỉ IP của B
 - Địa chỉ MAC đích = FF-FF-FF-FF-FF-FF
 - Tất cả các nút trên LAN đều nhận truy vấn ARP
- ❖ B nhận được gói tin ARP, sẽ trả lời A với địa chỉ MAC của mình.
 - Frame được gửi tới địa chỉ MAC của A (unicast)
- ❖ A ghi lại cặp địa chỉ IP-to-MAC trong bảng ARP của nó cho đến khi thông tin bị timeout.
 - Trạng thái mềm: thông tin này sẽ bị timeout trừ khi được làm mới lại.
- ❖ ARP là “plug-and-play”:
 - Các nút tạo ra bảng ARP của nó *mà không cần bất kỳ sự can thiệp nào từ nhà quản trị mạng.*

Tăng liên kết 5-46

Định địa chỉ: định tuyến tới LAN khác

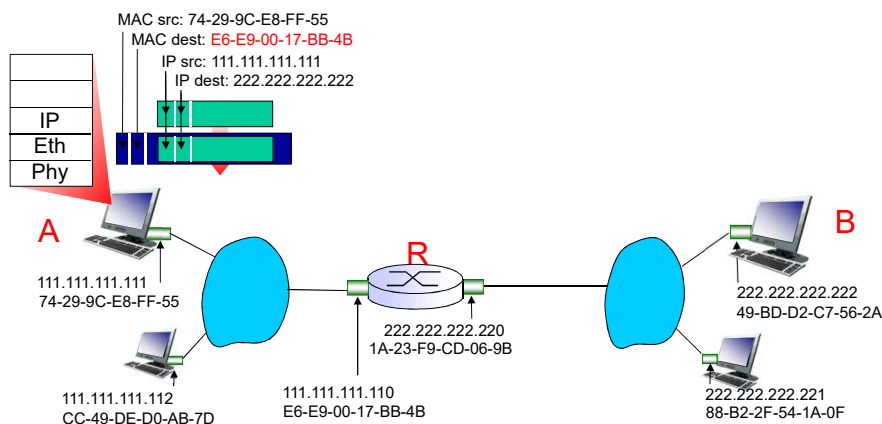
Tình huống: **gửi datagram từ A tới B qua R**

- Tập trung vào định địa chỉ – tại IP (datagram) và tầng MAC (frame)
- Giả thiết A biết địa chỉ IP của B
- Giả thiết A biết địa chỉ IP của router hop đầu tiên, là R (thì như thế nào?)
- Giả thiết A biết địa chỉ MAC của R (thì như thế nào?)



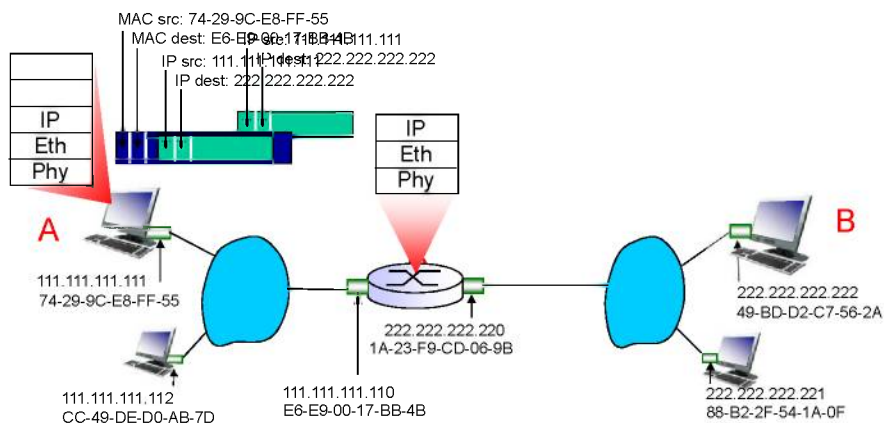
Định địa chỉ: định tuyến tới LAN khác

- ❖ A tạo IP datagram với IP nguồn A, đích B
- ❖ A tạo frame tầng liên kết với địa chỉ MAC của R là đích, frame chứa IP datagram từ A-tới-B



Định địa chỉ: định tuyến tới LAN khác

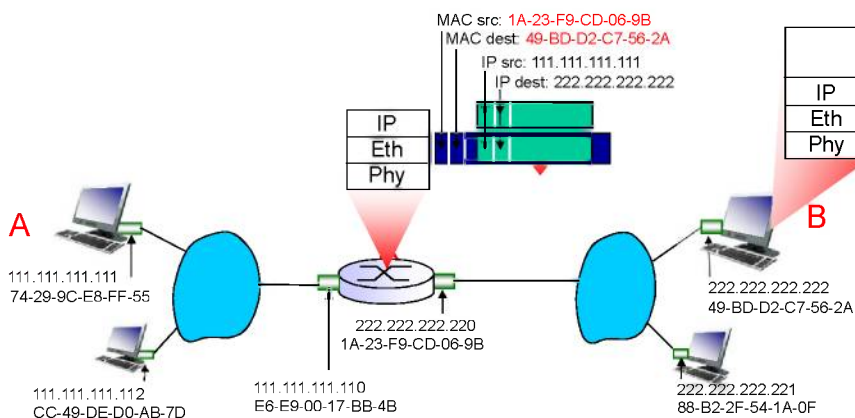
- ❖ Frame được gửi từ A tới R
- ❖ Frame được nhận tại R, datagram được chuyển lên tầng IP



Tầng liên kết 5-49

Định địa chỉ: định tuyến tới LAN khác

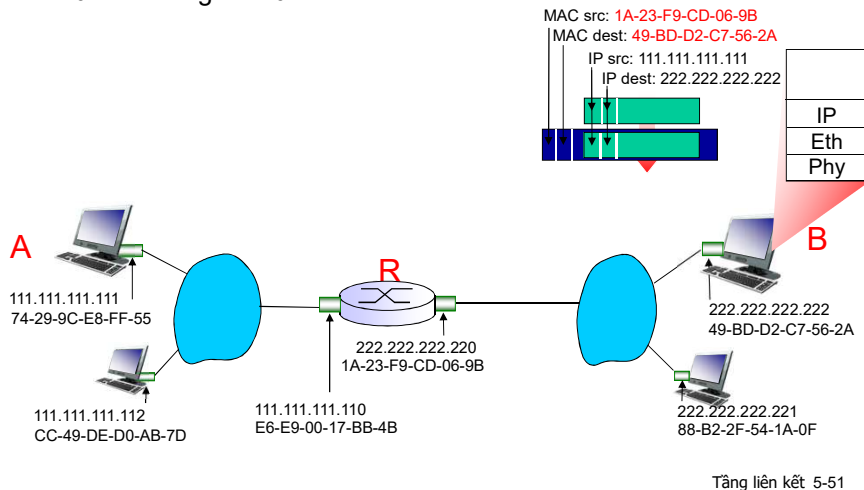
- ❖ R chuyển tiếp datagram với địa chỉ IP nguồn A, đích B
- ❖ R tạo frame tầng liên kết với địa chỉ MAC của B là đích, frame chứa IP datagram từ A-tới-B



Tầng liên kết 5-50

Định địa chỉ: định tuyến tới LAN khác

- ❖ R chuyển tiếp datagram với địa chỉ IP nguồn A, đích B
- ❖ R tạo frame tầng liên kết với địa chỉ MAC của B là đích, frame chứa IP datagram từ A-tới-B



Tầng liên kết và các mạng LAN: Nội dung

5.1 Giới thiệu, các dịch vụ

5.2 Phát hiện và sửa lỗi

5.3 Các giao thức đa truy nhập

5.4 Các mạng LAN

- Định địa chỉ, ARP
- Ethernet
- Các switch
- Các VLAN

5.5 Chuyển mạch nhãn đa giao thức (MPLS)

5.6 Mạng trung tâm dữ liệu

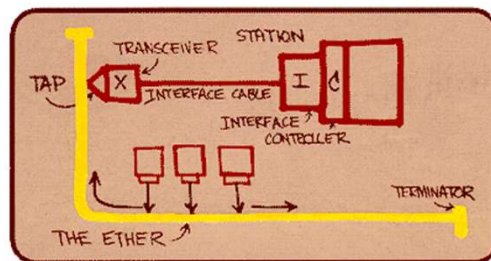
5.7 Vòng đời của một yêu cầu web

Tầng liên kết 5-52

Ethernet

“Thống trị” công nghệ mạng LAN có dây:

- ❖ Rẻ hơn \$20 cho NIC
- ❖ Công nghệ LAN được sử dụng phổ biến đầu tiên
- ❖ Đơn giản, rẻ hơn so với token LAN và ATM
- ❖ Giữ tốc độ trung bình từ: 10 Mbps – 10 Gbps

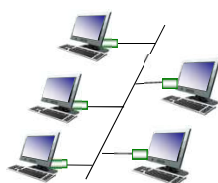


Phác họa Ethernet của Metcalfe

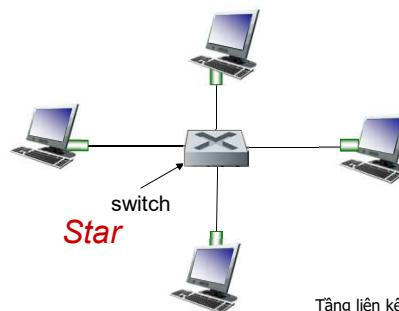
Tăng liên kết 5-53

Ethernet: cấu trúc vật lý

- ❖ **Bus**: phổ biến cho đến giữa thập niên 90
 - Tất cả các nút đều nằm trong vùng tranh chấp (có thể tranh chấp với các nút khác)
- ❖ **Star (hình sao)**: chiếm ưu thế hiện nay
 - **Switch** hoạt động ở trung tâm
 - Mỗi “chi nhánh” (*văn phòng, spoke*) chạy một giao thức Ethernet (riêng) (các nút không tranh chấp với nút khác)



Bus: cáp đồng trục



Star

Tăng liên kết 5-54

Cấu trúc Frame của Ethernet

Gửi IP datagram (hoặc gói giao thức tầng mạng khác) đã được đóng gói trong **frame** của **Ethernet**



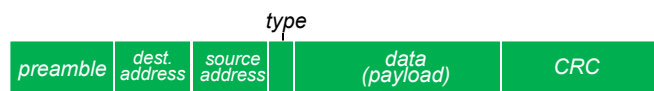
Trường **preamble**:

- ❖ 7 byte với mẫu 10101010 được theo sau bởi một byte với mẫu 10101011.
- ❖ Được dùng để đồng bộ tốc độ của bên nhận, bên gửi.

Tăng liên kết 5-55

Cấu trúc Frame của Ethernet (tiếp)

- ❖ Các trường **địa chỉ (nguồn và đích)**: 6 byte địa chỉ MAC nguồn và đích
 - Nếu adapter nhận frame với địa chỉ đích phù hợp, hoặc địa chỉ quảng bá (ví dụ: gói ARP), thì nó sẽ chuyển dữ liệu trong frame tới giao thức tầng mạng.
 - Ngược lại, adapter sẽ bỏ qua frame
- ❖ Trường **type**: chỉ ra giao thức tầng cao hơn (thường là IP nhưng cũng có thể là giao thức khác, ví dụ như Novell IPX, AppleTalk)
- ❖ Trường **CRC**: kiểm tra mã vòng dư thừa tại phía nhận
 - Phát hiện có lỗi: hủy bỏ frame



Tăng liên kết 5-56

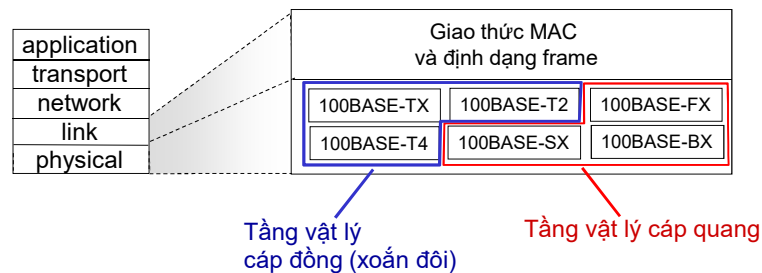
Ethernet: truyền không tin cậy, không hướng kết nối

- ❖ **Không hướng kết nối**: không có bắt tay giữa bên NIC gửi và bên NIC nhận
- ❖ **Không tin cậy**: NIC nhận không gửi báo nhận (ACK hoặc NACK) cho NIC gửi
 - Dữ liệu trong các frame đã bị hủy chỉ được khôi phục lại khi bên gửi khởi tạo việc dùng giao thức truyền tin cậy (rdt) ở tầng trên (ví dụ: TCP), còn không thì dữ liệu đó sẽ bị mất.
- ❖ Giao thức MAC của Ethernet: **CSMA/CD**

Tầng liên kết 5-57

Chuẩn Ethernet 802.3: tầng liên kết và tầng vật lý

- ❖ **Có nhiều** chuẩn Ethernet khác nhau
 - Giao thức MAC và định dạng frame chung
 - Tốc độ khác nhau: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
 - Phương tiện tầng vật lý khác nhau: cáp quang, cáp



Tầng liên kết 5-58

Tầng liên kết và các mạng LAN: Nội dung

- 5.1 Giới thiệu, các dịch vụ
- 5.2 Phát hiện và sửa lỗi
- 5.3 Các giao thức đa truy nhập
- 5.4 Các mạng LAN
 - Định địa chỉ, ARP
 - Ethernet
 - Các switch
 - Các VLAN
- 5.5 Chuyển mạch nhãn đa giao thức (MPLS)
- 5.6 Mạng trung tâm dữ liệu
- 5.7 Vòng đời của một yêu cầu web

Tầng liên kết 5-59

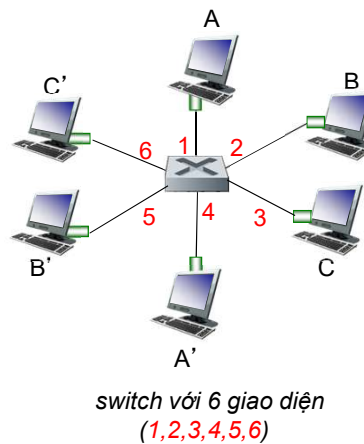
Switch Ethernet

- ❖ **Thiết bị tầng liên kết: có nhiệm vụ**
 - Lưu và chuyển tiếp các frame Ethernet
 - Kiểm tra địa chỉ MAC của frame đến, **chọn** và chuyển tiếp frame tới một hoặc nhiều liên kết ra
 - Khi frame được chuyển tiếp trên segment, dùng CSMA/CD để truy nhập segment
- ❖ **Trong suốt**
 - Các host không cần biết đến sự có mặt của các switch
- ❖ **Plug-and-play, tự học**
 - Các switch không cần được cấu hình

Tầng liên kết 5-60

Switch: Đa truyền đồng thời

- ❖ Các host trực tiếp kết nối với switch
- ❖ Các gói tin đệm trong switch
- ❖ Giao thức Ethernet được dùng trên *mỗi* liên kết đến, nhưng không tranh chấp; truyền song công.
 - Mỗi liên kết là vùng tranh chấp của riêng nó.
- ❖ **Chuyển mạch:** A-tới-A' và B-tới-B' có thể truyền đồng thời, mà không bị tranh chấp.



Tăng liên kết 5-61

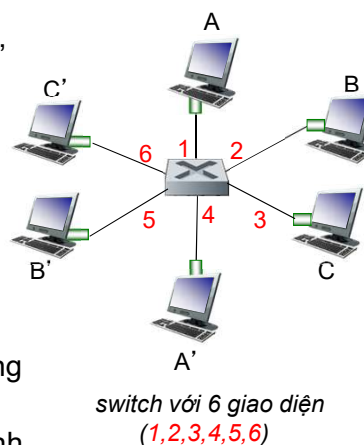
Bảng chuyển tiếp trong Switch

Hỏi: Làm thế nào switch biết được A' có thể truy cập thông qua giao diện 4, B' có thể truy cập được thông qua giao diện 5?

- ❖ **Trả lời:** Mỗi switch có một **bảng chuyển mạch**, mỗi mục:
 - (Địa chỉ MAC của host, giao diện tới host, nhãn thời gian)
 - Nhìn giống như bảng định tuyến!

Hỏi: Cách tạo các mục và duy trì trong bảng chuyển mạch như thế nào?

- Một số giống như giao thức định tuyến?

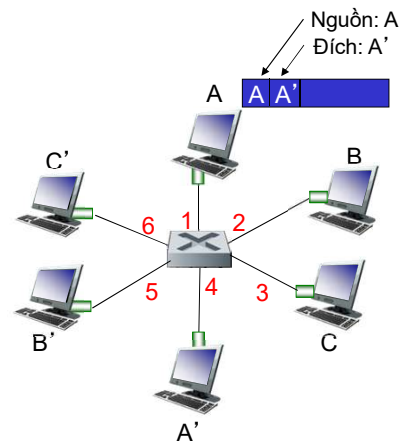


Tăng liên kết 5-62

Switch: tự học

❖ Switch *học* để biết những host nào có thể được truy nhập đến thông qua những giao diện nào

- Khi nhận được frame, switch “học” vị trí của bên gửi: segment LAN đi đến
- Ghi cập bên gửi/vị trí vào trong bảng chuyển mạch



Địa chỉ MAC	Giao diện	TTL
A	1	60

Bảng chuyển mạch
(khởi tạo rỗng)

Tăng liên kết 5-63

Switch: lọc/chuyển tiếp frame

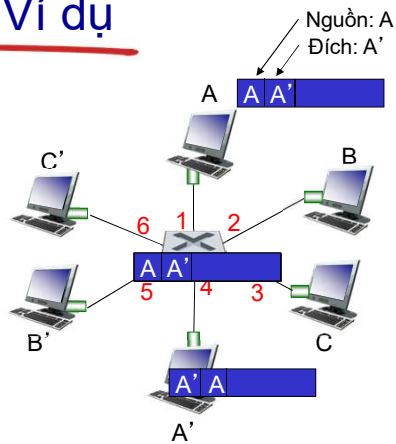
Khi switch nhận được frame:

1. Ghi lại liên kết đến, địa chỉ MAC của host gửi
2. Đánh chỉ mục bảng chuyển mạch dùng địa chỉ MAC đích
3. if mục được tìm thấy cho đích
then {
if đích trên segment mà từ đó frame đến
then bỏ qua frame
else chuyển tiếp frame trên giao diện được xác định bởi mục
}
else ngập tràn /* chuyển tiếp trên tất cả các giao diện ngoại trừ giao diện đến*/

Tăng liên kết 5-64

Tự học, chuyển tiếp: Ví dụ

- ❖ Đích frame, A', vị trí không được biết: *tràn ngập*
- ❖ Vị trí đích A được biết:
Lựa chọn gửi chỉ trên một liên kết



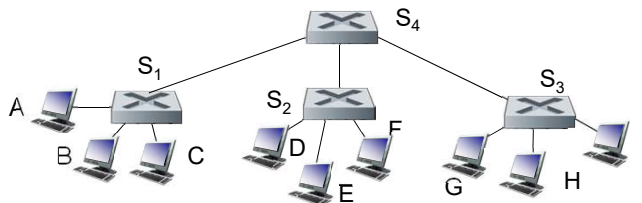
Địa chỉ MAC	Giao diện	TTL
A	1	60
A'	4	60

Bảng chuyển mạch
(khởi tạo rỗng)

Tăng liên kết 5-65

Kết nối các switch

- ❖ Các switch có thể được kết nối với nhau



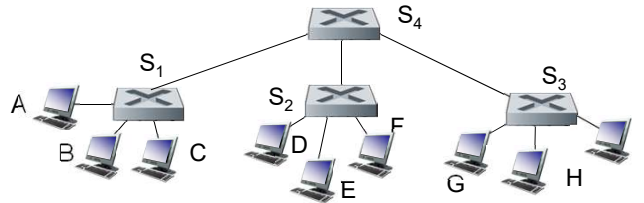
Hỏi: gửi từ A đến G – Làm thế nào S₁ biết cách chuyển tiếp frame hướng đích G qua S₄ và S₃?

- ❖ **Trả lời:** tự học! (làm theo đúng cách trong trường hợp switch đơn!)

Tăng liên kết 5-66

Ví dụ tự học nhiều switch

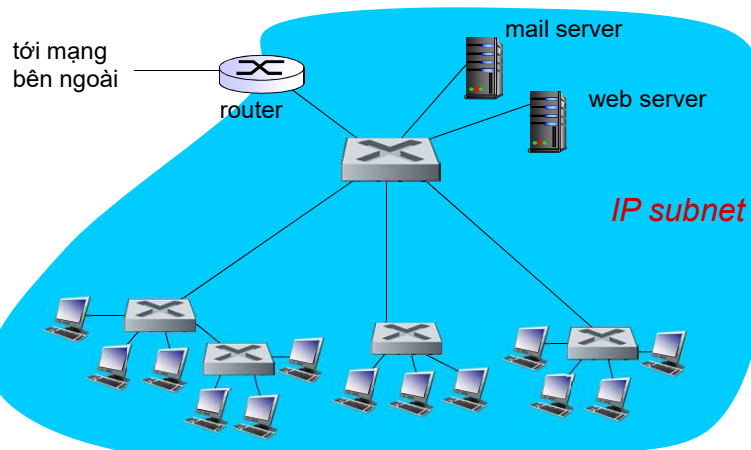
Giả sử C gửi frame tới I, I trả lời lại C



- ❖ **Hỏi:** Đưa ra các bảng chuyển mạch và chuyển tiếp gói tin trong S₁, S₂, S₃, S₄

Tăng liên kết 5-67

Mạng nội bộ trong một tổ chức



Tăng liên kết 5-68

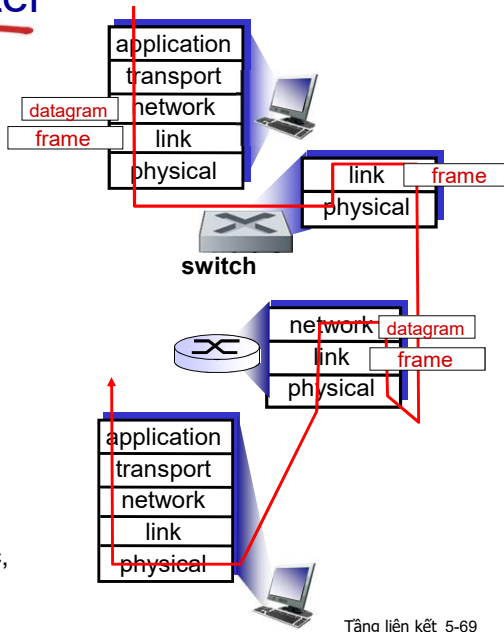
Các switch và router

Cả hai đều có chức năng lưu và chuyển tiếp (store-and-forward):

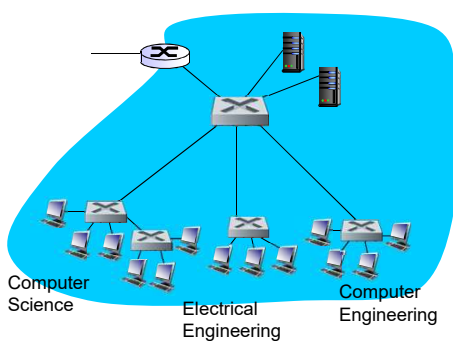
- **router**: thiết bị tầng mạng (kiểm tra phần tiêu đề tầng mạng)
- **switch**: thiết bị tầng liên kết (kiểm tra phần tiêu đề tầng liên kết)

Cả hai đều có bảng chuyển tiếp:

- **router**: tính toán bảng chuyển tiếp dùng các giải thuật định tuyến và địa chỉ IP
- **switch**: học bảng chuyển tiếp dùng kỹ thuật ngập lụt, tự học, và địa chỉ MAC



VLAN: Động lực



Xem xét:

- ❖ Người dùng CS chuyển văn phòng tới EE, nhưng muốn kết nối với switch của CS?
- ❖ Miền quảng bá đơn:
 - Tất cả lưu lượng quảng bá tầng 2 (ARP, DHCP, không biết vị trí của địa chỉ MAC đích) đều phải đi qua toàn bộ LAN.
 - Các vấn đề an toàn/sự riêng tư và hiệu suất.

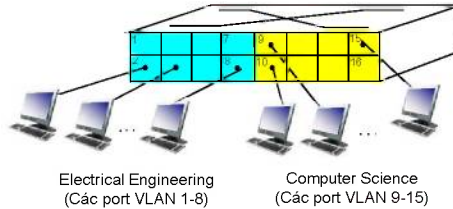
Tầng liên kết 5-70

VLAN

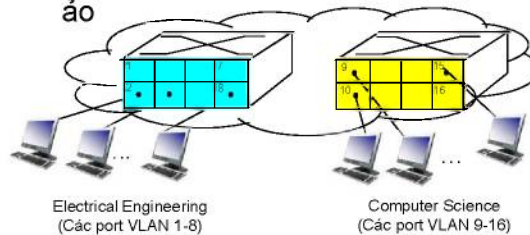
Virtual Local Area Network

Các switch hỗ trợ khả năng VLAN có thể được cấu hình để xác định nhiều mạng LAN **ảo** trên cơ sở hạ tầng mạng LAN vật lý duy nhất.

VLAN dựa trên cổng: Các port switch được nhóm lại (bởi phần mềm quản lý switch) để thành một switch vật lý **duy nhất**



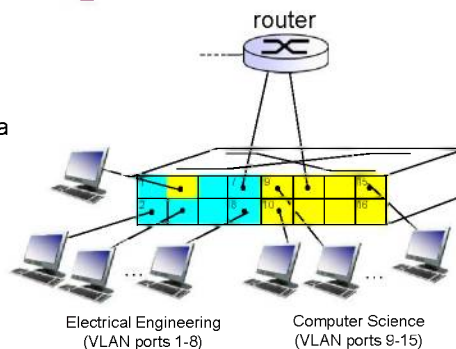
... hoạt động giống như là **nhiều** switch ảo



Tăng liên kết 5-71

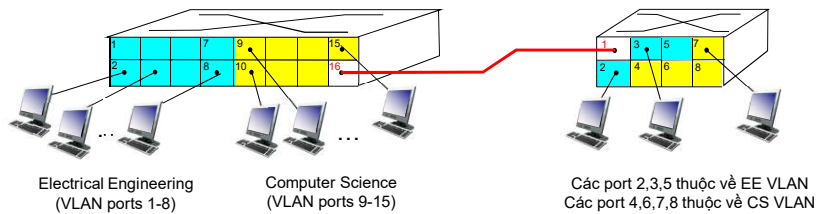
VLAN dựa trên cổng

- ❖ **Cô lập lưu lượng:** các frame tới/từ các port 1-8 **chỉ** có thể tới các port 1-8
 - Cũng có thể xác định VLAN dựa trên địa chỉ MAC của các điểm cuối (endpoint), thay vì các port của switch
- ❖ **Thành viên động:** các port có thể được gán động giữa các VLAN
- ❖ **Chuyển tiếp giữa các VLAN:** được thực hiện thông qua định tuyến (chỉ như là các switch riêng)
 - Thực tế các nhà cung cấp bán các switch được kết hợp với các router



Tăng liên kết 5-72

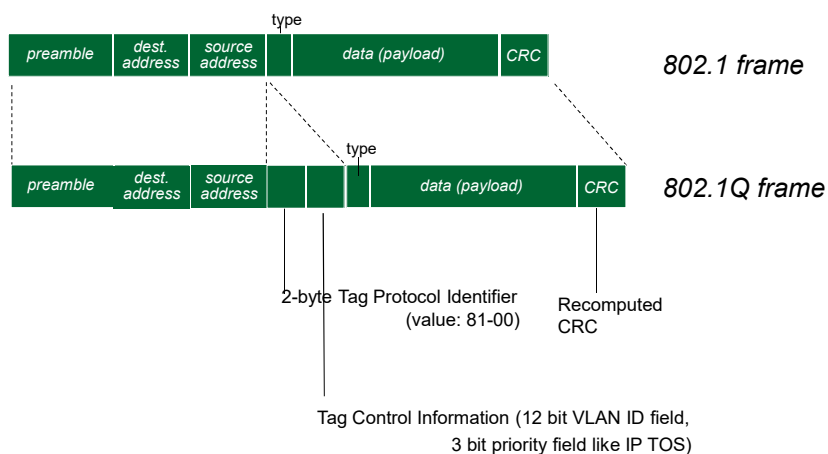
VLAN mở rộng qua nhiều switch



- ❖ **Trunk port:** mang các frame giữa các VLAN được xác định qua nhiều switch vật lý.
 - Các frame được chuyển tiếp bên trong VLAN giữa các switch cần mang thông tin ID của VLAN.
 - Giao thức 802.1q thêm/xóa các trường tiêu đề bổ sung của frame được chuyển tiếp giữa các trunk port.

Tăng liên kết 5-73

Định dạng frame 802.1Q VLAN



Tăng liên kết 5-74

Tầng liên kết và các mạng LAN: Nội dung

5.1 Giới thiệu, các dịch vụ

5.2 Phát hiện và sửa lỗi

5.3 Các giao thức đa truy nhập

5.4 Các mạng LAN

- Định địa chỉ, ARP
- Ethernet
- Các switch
- Các VLAN

5.5 Chuyển mạch nhãn đa giao thức (MPLS)

5.6 Mạng trung tâm dữ liệu

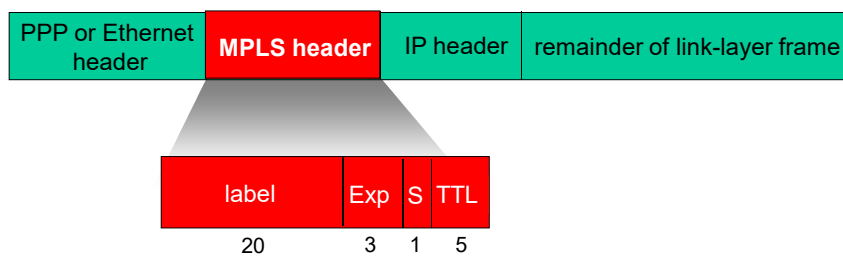
5.7 Vòng đời của một yêu cầu web

Tầng liên kết 5-75

Chuyển mạch nhãn đa giao thức

(Multiprotocol label switching - MPLS)

- ❖ Mục tiêu ban đầu: chuyển mạch IP tốc độ cao sử dụng nhãn chiều dài cố định (thay vì dùng địa chỉ IP)
 - Tìm kiếm nhanh sử dụng định danh chiều dài cố định (thay vì so khớp prefix ngắn nhất)
 - Dựa theo cách tiếp cận mạch ảo (VC)
 - Nhưng IP datagram vẫn giữ địa chỉ IP!



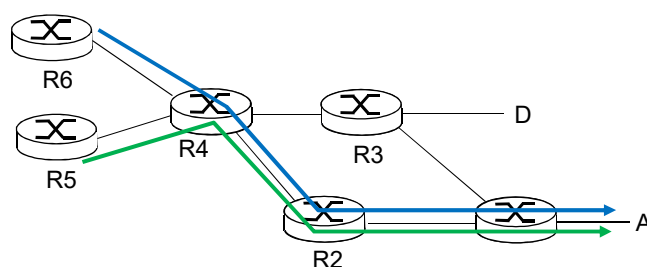
Tầng liên kết 5-76

Các router có khả năng MPLS

- ❖ Còn được gọi là các router chuyển mạch nhãn
- ❖ Chuyển tiếp các gói tin tới giao diện ra chỉ dựa trên giá trị nhãn (không *kiểm tra* địa chỉ IP)
 - Bảng chuyển tiếp MPLS khác với bảng chuyển tiếp IP
- ❖ **Tính mềm dẻo:** các quyết định chuyển tiếp MPLS có thể khác với các quyết định chuyển tiếp của IP
 - Dùng địa chỉ đích và nguồn để định hướng luồng tới cùng đích theo các cách khác nhau (kỹ thuật luồng)
 - Định tuyến lại luồng nhanh chóng nếu liên kết bị lỗi: có các đường đi dự phòng được tính toán từ trước (hữu dụng cho VoIP).

Tăng liên kết 5-77

So sánh đường đi của MPLS với IP

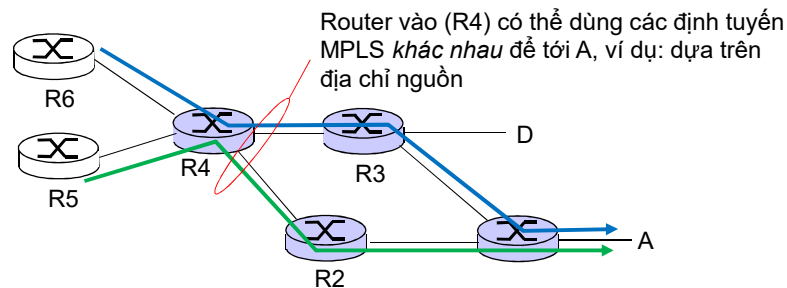


- ❖ **Định tuyến IP:** đường đi tới đích chỉ được xác định bởi địa chỉ đích

 Router IP

Tăng liên kết 5-78

So sánh đường đi của MPLS với IP



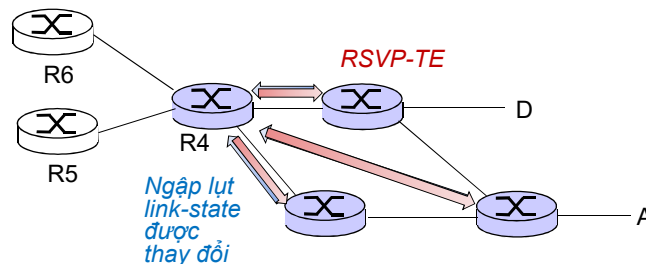
- ❖ **Định tuyến IP:** đường đi tới đích chỉ được xác định bởi địa chỉ đích
- ❖ **Định tuyến MPLS:** đường đi tới đích có thể được xác định dựa trên địa chỉ nguồn và địa chỉ đích
 - **Định tuyến lại nhanh:** có đường đi dự phòng trong trường hợp liên kết bị lỗi



Tăng liên kết 5-79

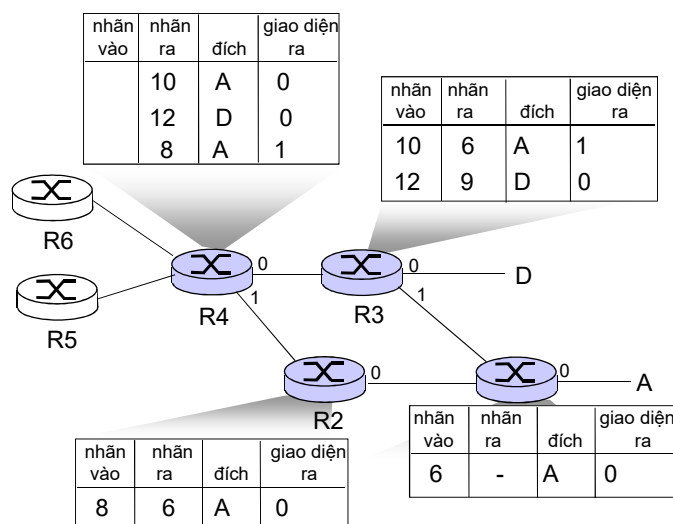
Báo hiệu trong MPLS

- ❖ Thay đổi OSPF, các giao thức ngập lụt link-state IS-IS để mang thông tin cho định tuyến MPLS
 - Ví dụ: bảng thông liên kết, tổng bảng thông của liên kết “dự phòng”
- ❖ Mục router MPLS sử dụng giao thức báo hiệu RSVP-TE để thiết lập chuyển tiếp MPLS tại dòng downstream của các router.



Tăng liên kết 5-80

Các bảng chuyển tiếp MPLS



Tầng liên kết 5-81

Tầng liên kết và các mạng LAN: Nội dung

5.1 Giới thiệu, các dịch vụ

5.2 Phát hiện và sửa lỗi

5.3 Các giao thức đa truy nhập

5.4 Các mạng LAN

- Định địa chỉ, ARP
- Ethernet
- Các switch
- Các VLAN

5.5 Chuyển mạch nhãn đa giao thức (MPLS)

5.6 Mạng trung tâm dữ liệu

5.7 Vòng đời của một yêu cầu web

Tầng liên kết 5-82

Mạng trung tâm dữ liệu (Data center networks)

- ❖ 10 đến 100 nghìn host, thường được kết hợp chặt chẽ:
 - Thương mại điện tử (e-business) (ví dụ: Amazon)
 - Các máy chủ nội dung (content-server) (Ví dụ: YouTube, Akamai, Apple, Microsoft)
 - search engines, khai phá dữ liệu (Ví dụ: Google)
- ❖ Thách thức:
 - Nhiều ứng dụng, mỗi ứng dụng phục vụ một số lượng rất lớn các client
 - Quản lý/cân bằng tải, xử lý các vấn đề về mạng, tắc nghẽn dữ liệu.



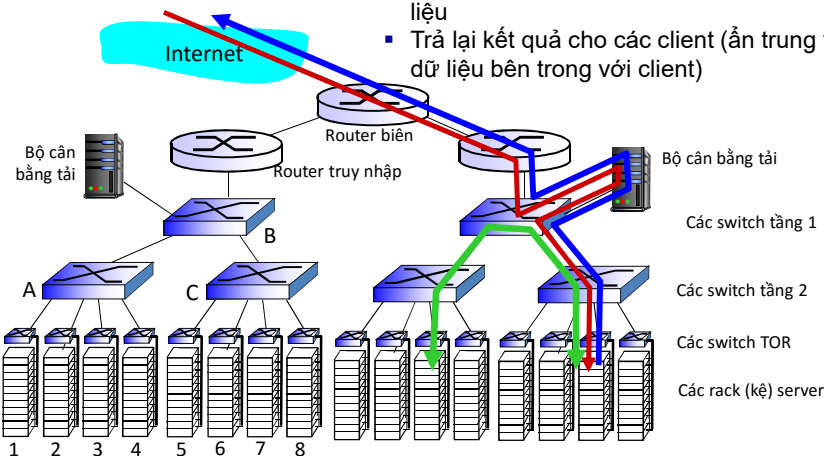
Bên trong một container 40-ft của Microsoft, tại trung tâm dữ liệu ở Chicago

Tầng liên kết 5-83

Mạng trung tâm dữ liệu

Bộ cân bằng tải: định tuyến tầng ứng dụng

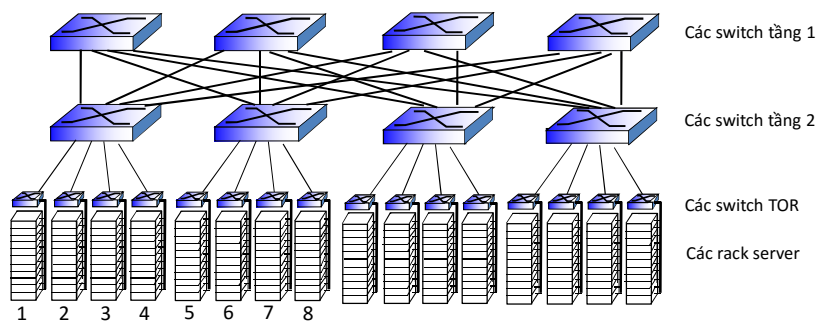
- Nhận các yêu cầu từ client ở phía ngoài
- Chỉ đạo công việc bên trong trung tâm dữ liệu
- Trả lại kết quả cho các client (ẩn trung tâm dữ liệu bên trong với client)



Tầng liên kết 5-84

Mạng trung tâm dữ liệu

- ❖ Kết nối rất nhiều các switch và các kệ (rack):
 - Tăng thông lượng giữa các kệ (có thể có nhiều đường đi)
 - Tăng độ tin cậy thông qua dự phòng.



Tầng liên kết 5-85

Tầng liên kết và các mạng LAN: Nội dung

- 5.1 Giới thiệu, các dịch vụ
- 5.2 Phát hiện và sửa lỗi
- 5.3 Các giao thức đa truy nhập
- 5.4 Các mạng LAN
 - Định địa chỉ, ARP
 - Ethernet
 - Các switch
 - Các VLAN
- 5.5 Chuyển mạch nhãn đa giao thức (MPLS)
- 5.6 Mạng trung tâm dữ liệu
- 5.7 Vòng đời của một yêu cầu web

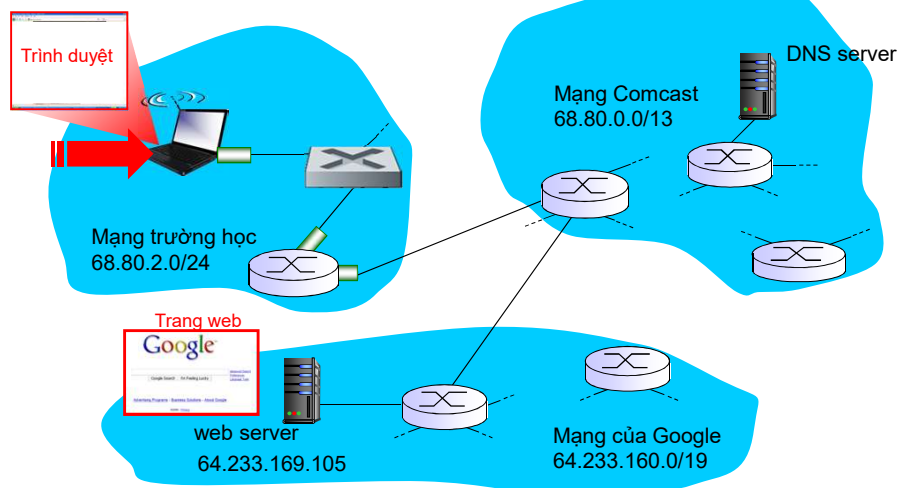
Tầng liên kết 5-86

Tổng hợp: Vòng đời của một yêu cầu web

- ❖ Hành trình đi xuống chồng giao thức đã hoàn thành!
 - Tầng ứng dụng, tầng giao vận, tầng mạng, tầng liên kết.
- ❖ Đặt tất cả mọi thứ lại cùng nhau: tổng hợp!
 - **Mục đích:** xác định, xem xét, hiểu các giao thức (tại tất cả các tầng) liên quan trong một kịch bản khá đơn giản: yêu cầu một trang web.
 - **Kịch bản:** Sinh viên thực hiện yêu cầu/đáp ứng từ hệ thống mạng trong trường: www.google.com

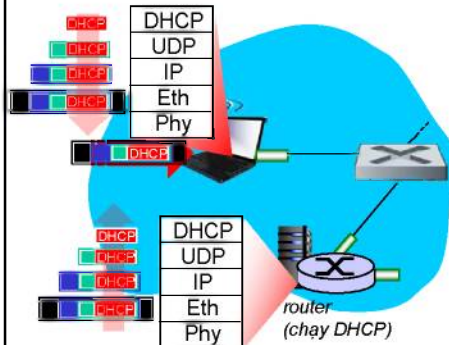
Tầng liên kết 5-87

Kịch bản: Vòng đời của một yêu cầu web



Tầng liên kết 5-88

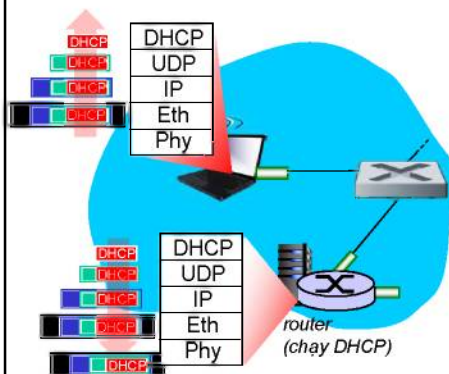
Kết nối đến Internet



- ❖ Việc kết nối đến máy tính cần có địa chỉ IP của máy, địa chỉ của router hop đầu tiên, địa chỉ của DNS server: dùng **DHCP**
- ❖ DHCP request được đóng gói trong **UDP**, được đóng gói trong **IP**, được đóng gói trong **802.3** Ethernet
- ❖ Ethernet frame **quảng bá** (dest: FFFFFFFF) trên LAN, được nhận tại router đang chạy **DHCP** server
- ❖ Ethernet **mở gói** thành IP, IP mở gói thành UDP, UDP mở gói thành DHCP

Tăng liên kết 5-89

Kết nối đến Internet

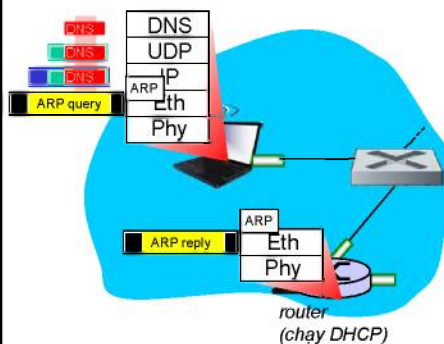


- ❖ DHCP server định dạng **DHCP ACK** chứa địa chỉ IP của client, địa chỉ IP của router hop đầu tiên cho client, tên và địa chỉ IP của DNS server
- ❖ Đóng gói tại DHCP server, frame được chuyển tiếp (**học chuyển mạch**) qua LAN, việc mở gói tại client
- ❖ DHCP client nhận trả lời DHCP ACK

Bây giờ, client có địa chỉ IP, biết được tên và địa chỉ của DNS server, địa chỉ IP của router của hop đầu tiên của nó.

Tăng liên kết 5-90

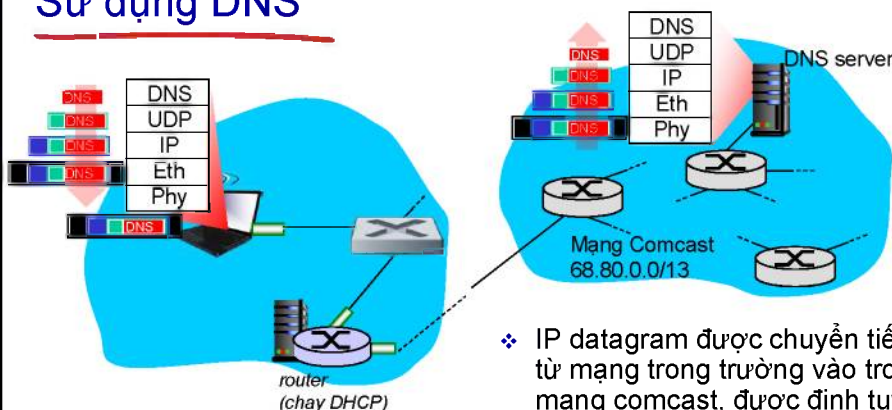
ARP (trước DNS, trước HTTP)



- ❖ Trước khi gửi yêu cầu *HTTP*, cần địa chỉ IP của `www.google.com`: *DNS*
- ❖ Truy vấn DNS được tạo ra, được đóng gói trong UDP, được đóng gói trong IP, được đóng gói trong Ethernet. Để gửi frame tới router, cần địa chỉ MAC của giao diện router: *ARP*
- ❖ *ARP query* quảng bá, được nhận bởi router mà sẽ trả lời với *ARP reply*, cho biết địa chỉ MAC của giao diện router.
- ❖ Lúc này, client biết địa chỉ MAC của router hop đầu tiên, do đó có thể gửi frame chứa truy vấn DNS

Tăng liên kết 5-91

Sử dụng DNS

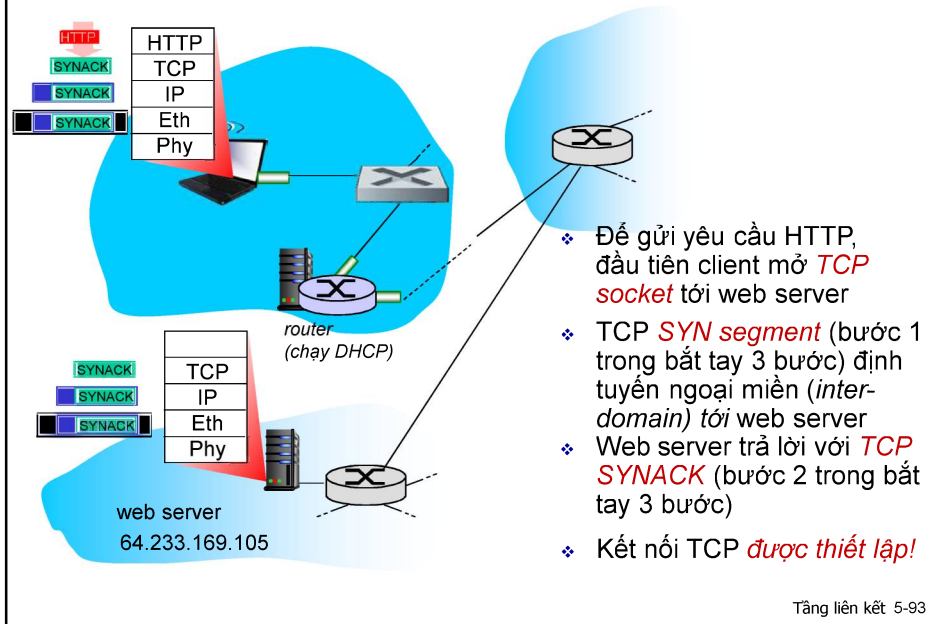


- ❖ IP datagram chứa truy vấn DNS được chuyển tiếp qua switch LAN từ client tới router hop đầu tiên

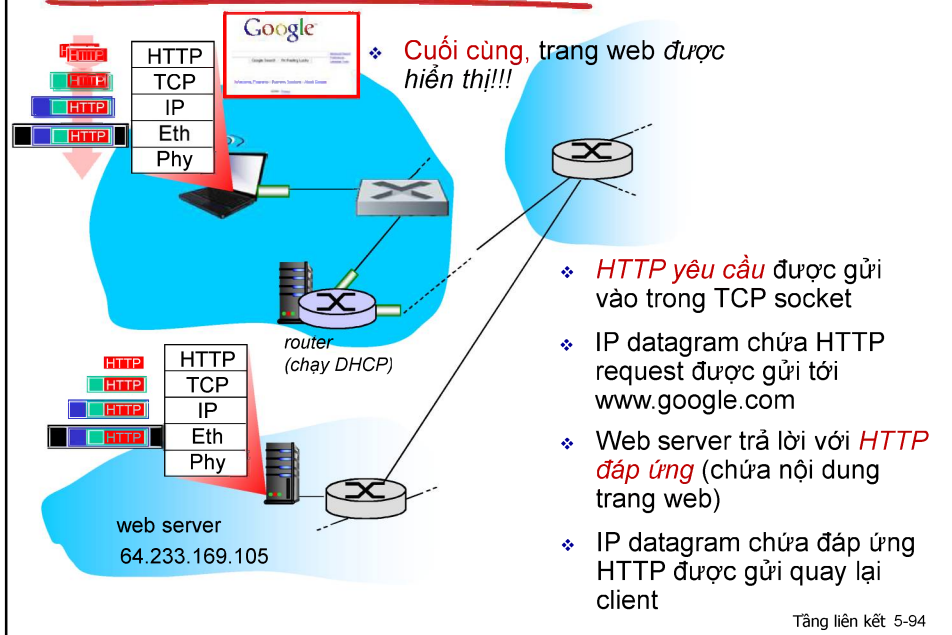
- ❖ IP datagram được chuyển tiếp từ mạng trong trường vào trong mạng comcast, được định tuyến (các bảng được tạo ra bởi các giao thức định tuyến *RIP*, *OSPF*, *IS-IS* và/hoặc *BGP*) tới DNS server
- ❖ DNS server trả lời lại client với địa chỉ IP của `www.google.com`

Tăng liên kết 5-92

Kết nối TCP mang thông điệp HTTP



HTTP yêu cầu/đáp ứng



Chương 5: Tổng kết

- ❖ Nguyên lý các dịch vụ bên trong tầng liên kết dữ liệu:
 - Phát hiện và sửa lỗi
 - Chia sẻ các kênh truyền chung: đa truy nhập
 - Định địa chỉ tầng liên kết
- ❖ Hiện thực và cài đặt một số công nghệ tầng liên kết
 - Ethernet
 - switched LANS, VLANs
 - MPLS
- ❖ Tổng hợp: Vòng đời của một yêu cầu web

Tầng liên kết 5-95

Chương 5: Tổng kết

- ❖ Còn *rất nhiều* chủ đề thú vị đáng quan tâm!
 - Mạng không dây
 - Đa phương tiện (multimedia)
 - An ninh mạng
 - Quản trị mạng
 - ...

Tầng liên kết 5-96