

# Mạng máy tính

ThS. Phạm Liệu  
Email: lieu.pham@stu.edu.vn

2018



# NỘI DUNG CHÍNH



Internet là gì?



Mạng biên



Mạng lõi



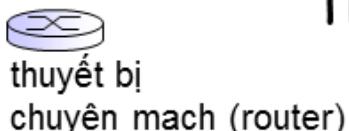
Độ trễ, sự mất mát, thông lượng trong mạng



Các lớp giao thức, các mô hình dịch vụ



# Internet là gì?



❖ Hàng triệu các thiết bị máy tính được kết nối:

- *hosts = hệ thống đầu cuối*
- chạy *ứng dụng mạng*

❖ *Các liên kết truyền thông*

- cáp quang, cáp đồng, radio, vệ tinh
- tốc độ truyền: *băng thông*

❖ *Chuyển mạch gói: chuyển tiếp gói tin (khối dữ liệu)*

- *Thiết bị định tuyến (routers) và thiết bị chuyển mạch (switches)*

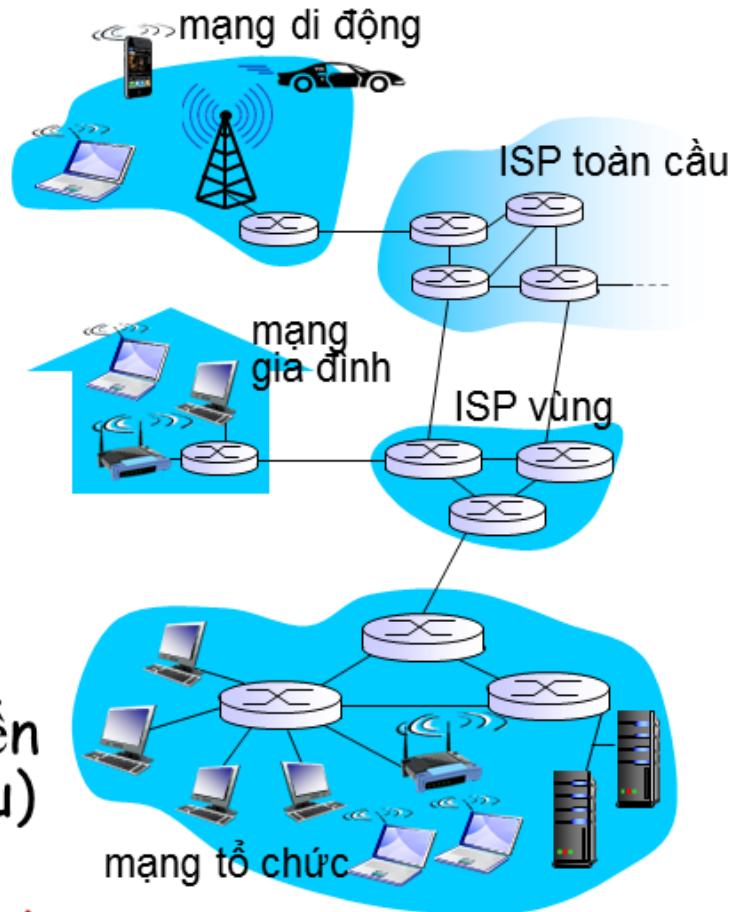
Internet là gì?

Mạng biển

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Các thiết bị internet “Giải trí”



**Khung hình IP**

<http://www.ceiva.com/>



**Tủ lạnh internet**



**Máy nướng bánh mì được tích hợp web+ dữ báo thời tiết**



**Tweet-a-watt:**

**Máy giám sát**

**sử dụng năng lượng**



**Điện thoại internet**



**Internet là gì?**

**Mạng biển**

**Mạng lõi**

**Độ trễ, sự mất mát, thông lượng trong mạng**

**Các lớp giao thức, các mô hình dịch vụ**



# Internet là gì?

- *Internet: “mạng của các mạng”*
  - Các nhà cung cấp dịch vụ mạng (ISPs) được kết nối với nhau.
- *Các giao thức* điều khiển gửi, nhận thông tin
  - Ví dụ: TCP, IP, HTTP, Skype, 802.11
- *Các chuẩn Internet*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force

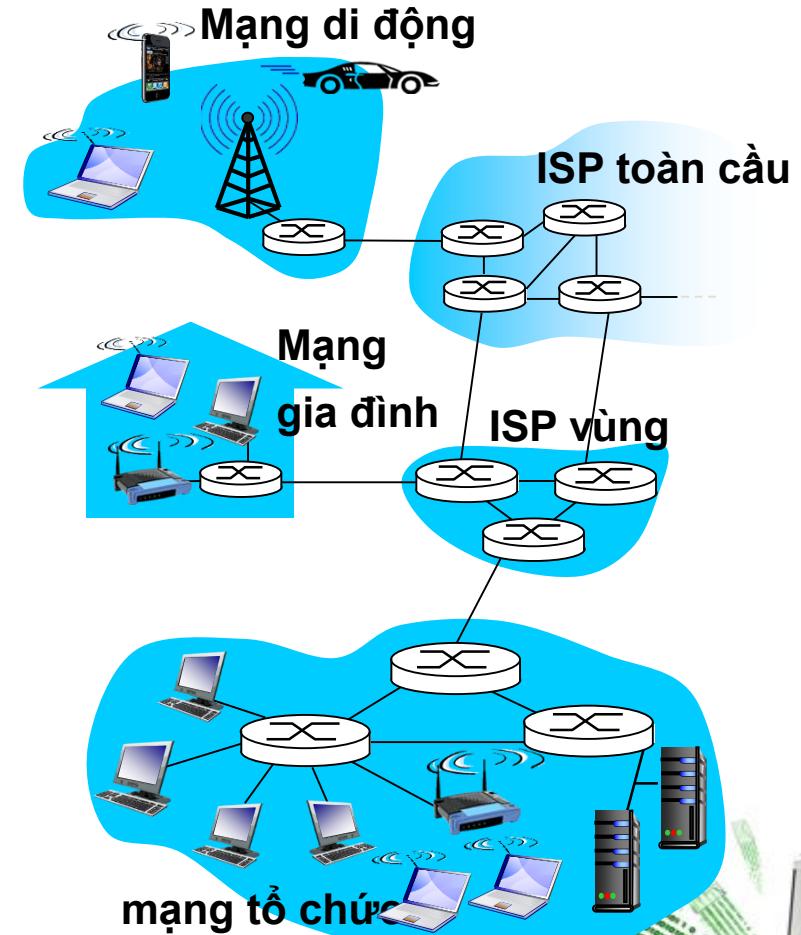
Internet là gì?

Mạng biển

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Internet là gì: cái nhìn về dịch vụ

- *Cơ sở hạ tầng cung cấp các dịch vụ cho các ứng dụng:*
  - Web, VoIP, email, games, thương mại điện tử, mạng xã hội, ...
- *Cung cấp giao diện lập trình cho các ứng dụng*
  - Cái móc cho phép gửi và nhận các chương trình ứng dụng để "kết nối" với Internet
  - cung cấp các lựa chọn dịch vụ, tương tự như dịch vụ bưu chính.

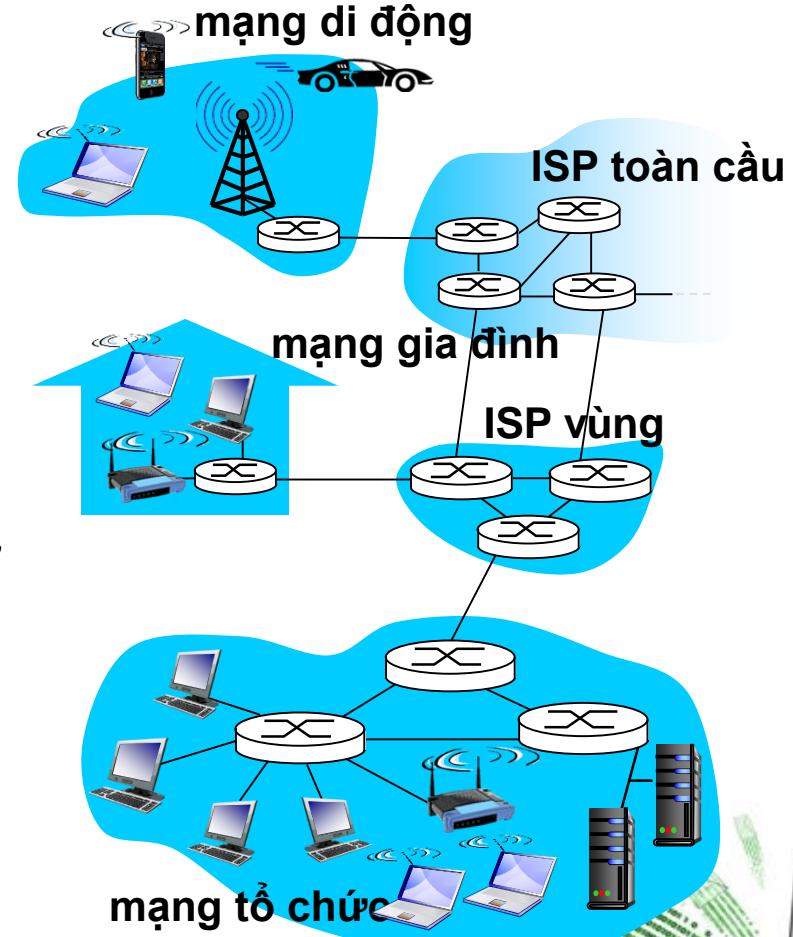
Internet là gì?

Mạng biển

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



## Giao thức là gì?

### Giao thức con người:

- “Bây giờ là mấy giờ?”
- “Tôi có một câu hỏi”
- Giới thiệu

... thông điệp cụ thể đã được gửi

... các hành động cụ thể được thực hiện khi các thông điệp được nhận, hoặc các sự kiện khác

### Giao thức mạng:

- Máy móc chứ không phải là con người
- tất cả các hoạt động truyền thông trên Internet bị chi phối bởi các giao thức.

**Giao thức định nghĩa định dạng, thứ tự các thông điệp được gửi và nhận giữa các thực thể mạng, và các hành động được thực hiện trên việc truyền và nhận thông điệp**

Internet là gì?

Mạng biển

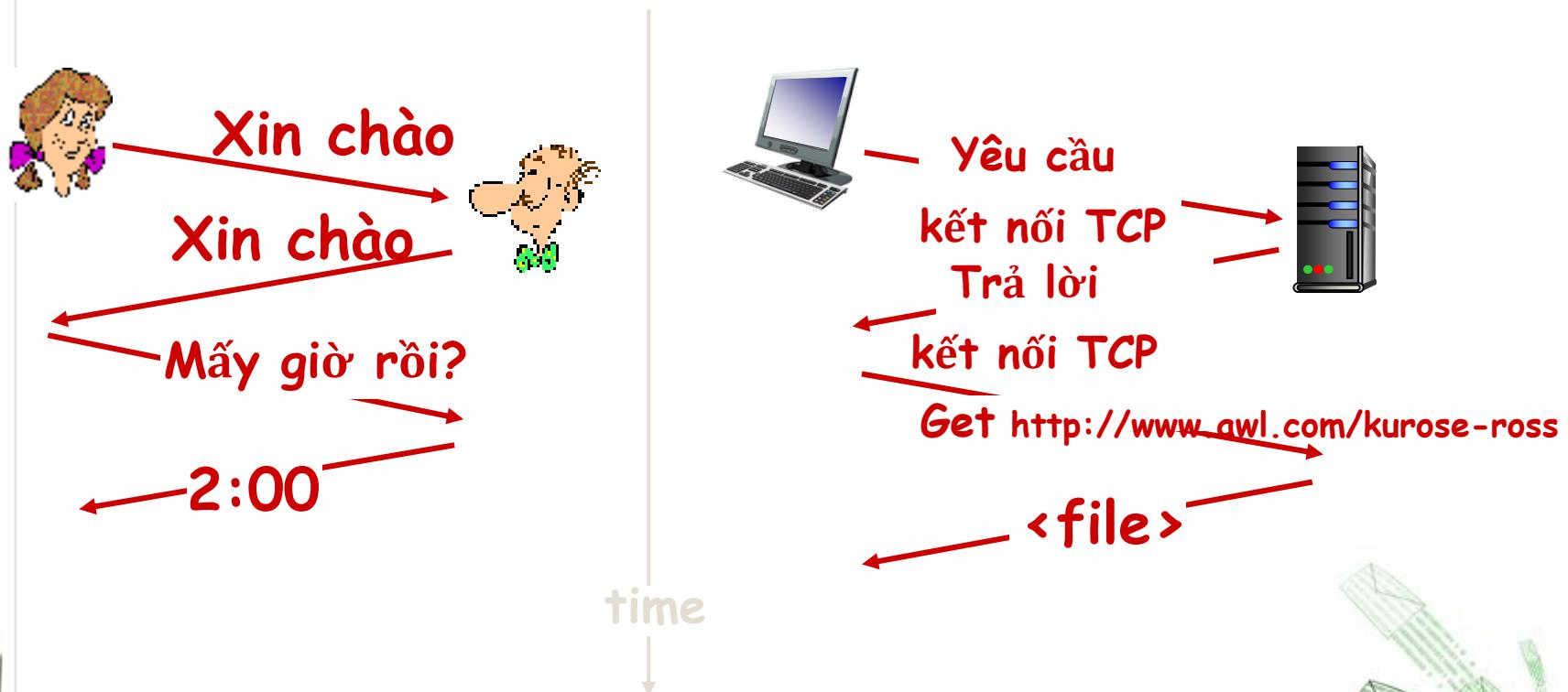
Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

## Giao thức là gì?

Giao thức con người và giao thức mạng máy tính:



Hỏi: các giao thức khác của con người?

Internet là gì?

Mạng biển

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Chương 1: Nội dung

## 1.1 Internet là gì?

## 1.2 Mạng biên

- hệ thống đầu cuối, mạng truy nhập, các liên kết

## 1.3 Mạng lõi

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

## 1.4 Độ trễ, sự mất mát, thông lượng trong mạng

## 1.5 Các lớp giao thức, mô hình dịch vụ

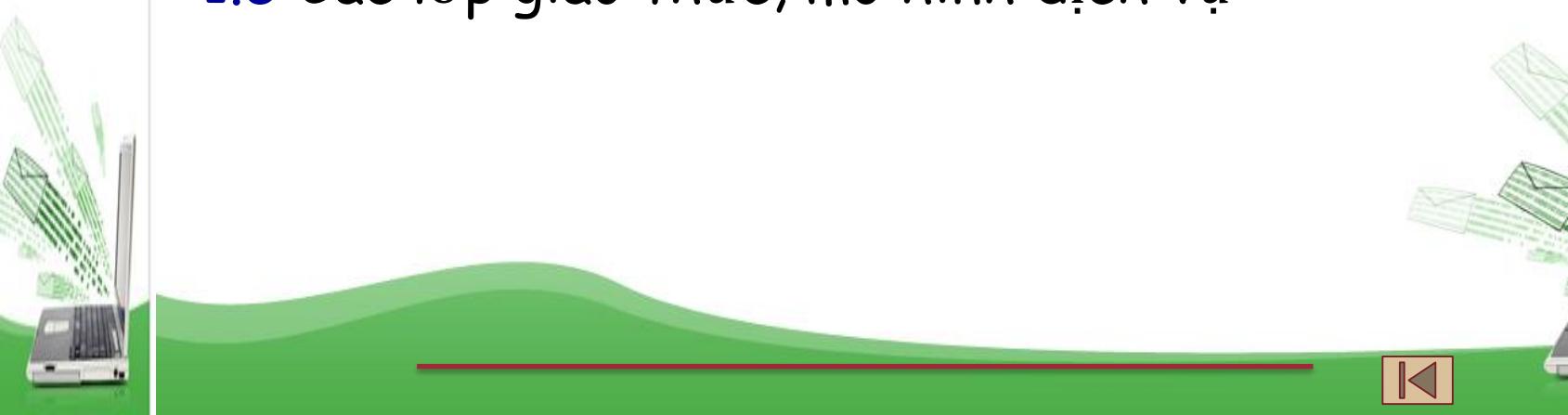
Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Cái nhìn gần hơn về cấu trúc mạng:

## Mạng biên:

- hosts: máy khách và máy chủ
- Máy chủ thường đặt trong trung tâm dữ liệu

❖ **Mạng truy nhập, phương tiện truyền thông vật lý:** kết nối truyền thông có dây, và không dây

## Mạng lõi:

- các thiết bị định tuyến được liên kết
- mạng của các mạng

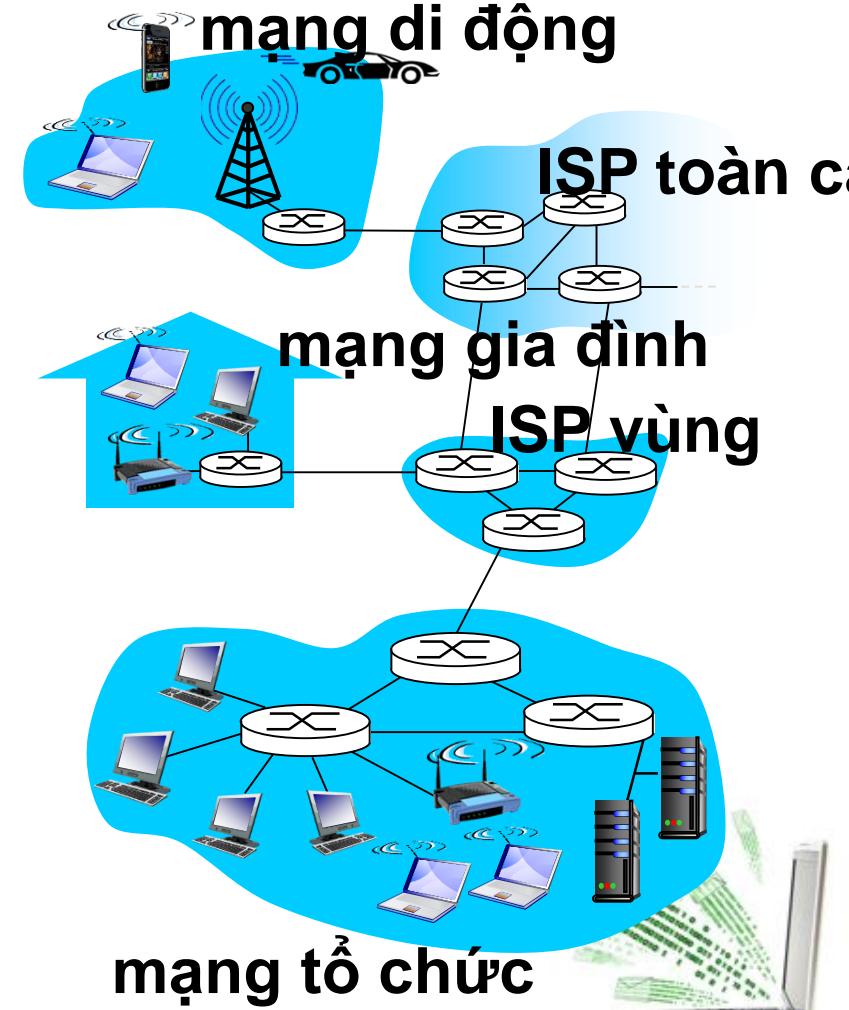
Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Mạng truy cập và phương tiện truyền thông vật lý

**Q: Làm thế nào để kết nối các hệ thống đầu cuối với thiết bị định tuyến biên?**

- mạng truy cập khu dân cư
- mạng truy cập tổ chức (trường học, công ty)
- mạng truy cập di động

**Ghi nhớ:**

- băng thông (bits per second) của mạng truy cập?

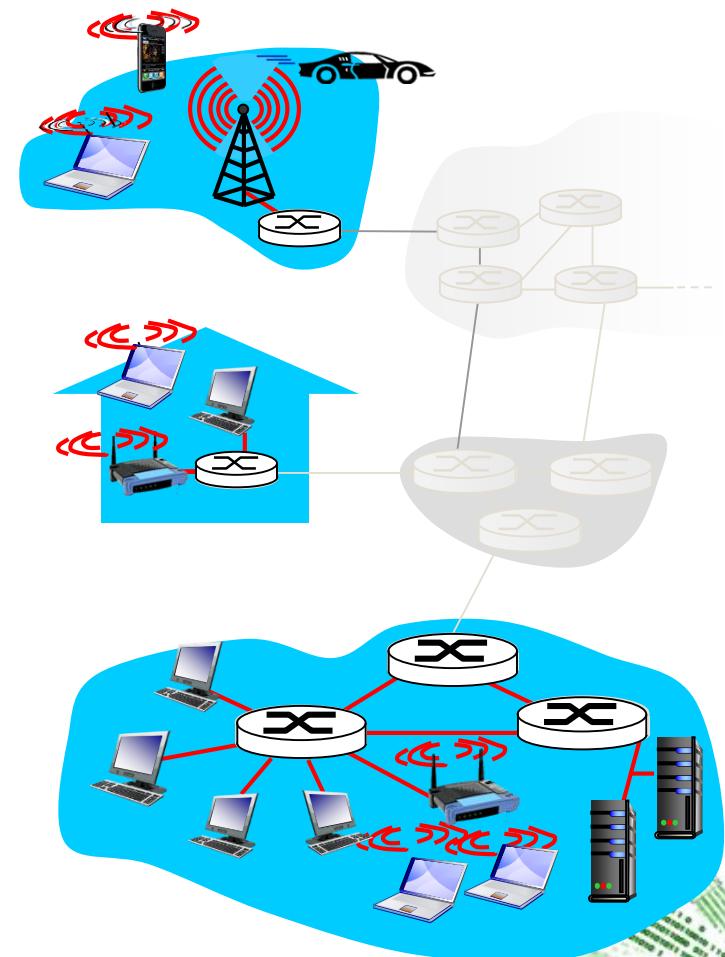
Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



Chia sẻ hoặc dành riêng?

# Mạng truy cập: mạng gia đình

Internet là gì?

Mạng biển

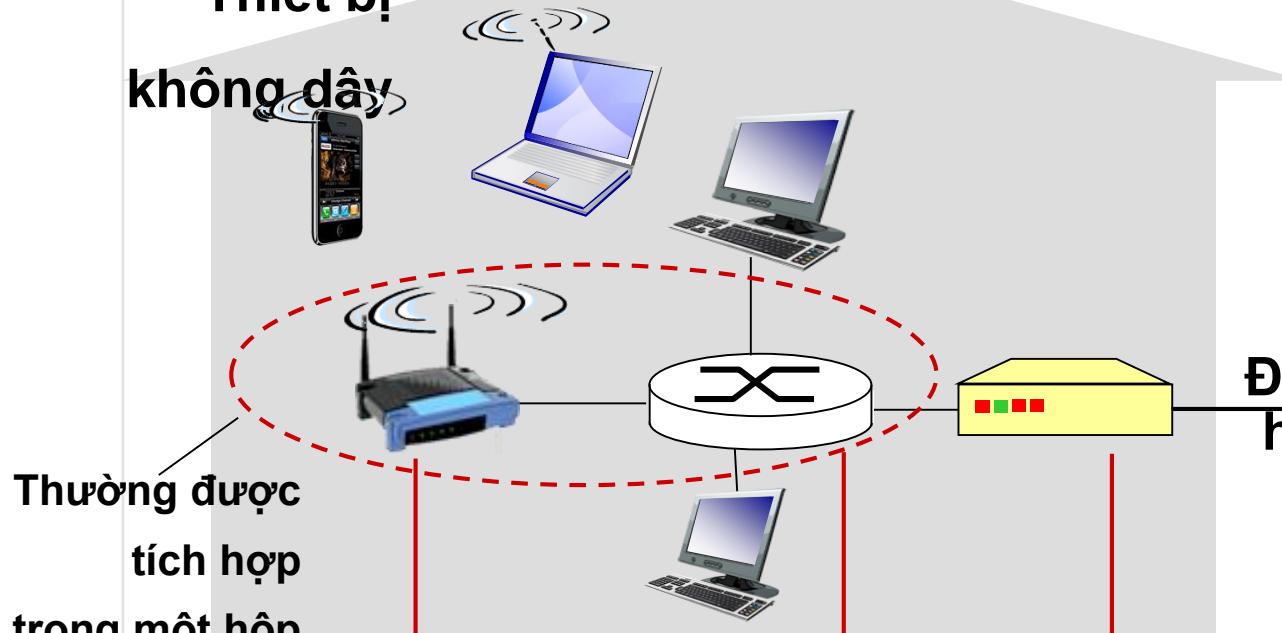
Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

## Thiết bị

### không dây



Thường được  
tích hợp  
trong một hộp  
duy nhất

wireless access  
point (54 Mbps)

wired Ethernet (100  
Mbps)

Đến/từ thiết bị đầu cuối

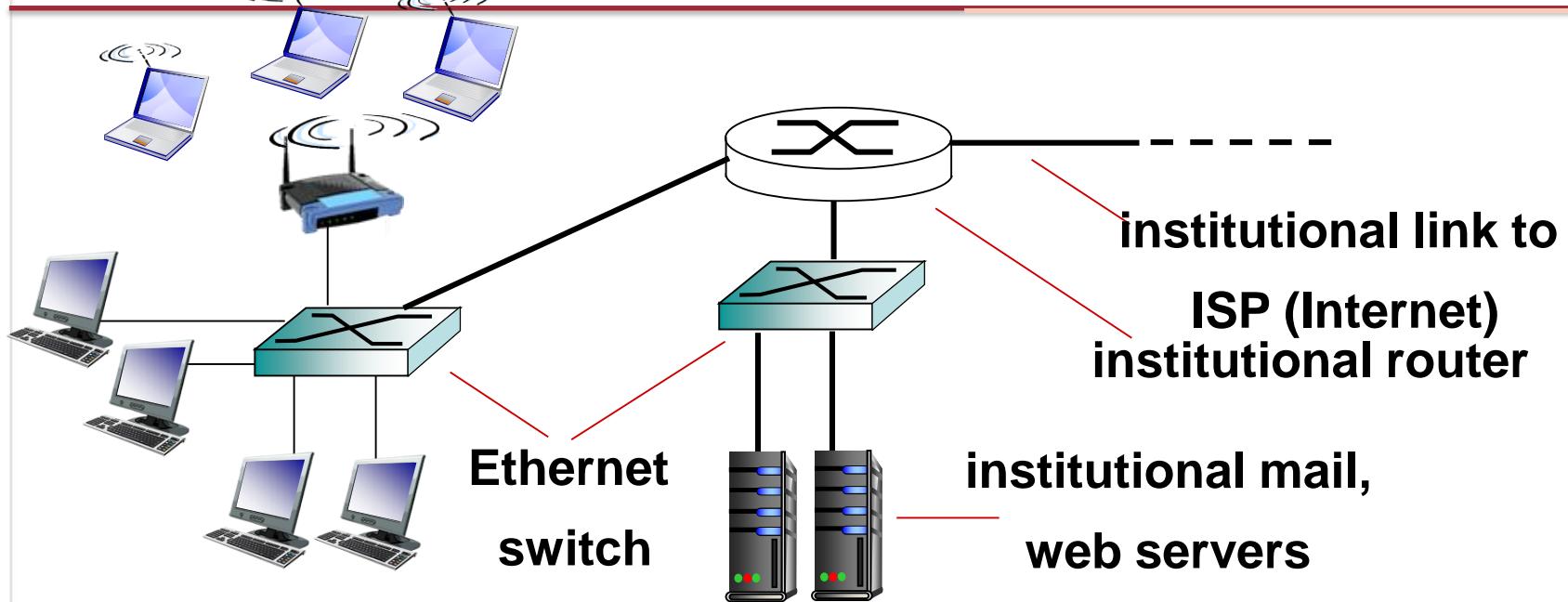
hoặc văn phòng trung  
tâm

cable or DSL modem

router, firewall, NAT



# Mạng truy cập Enterprise (Ethernet)



- Thường được sử dụng trong công ty, trường đại học...
- ❖ Tốc độ truyền 10 Mbps, 100Mbps, 1Gbps, 10Gbps
- ❖ Ngày nay, các hệ thống đầu cuối thường kết nối đến thiết bị chuyển mạch Ethernet (Ethernet switch)



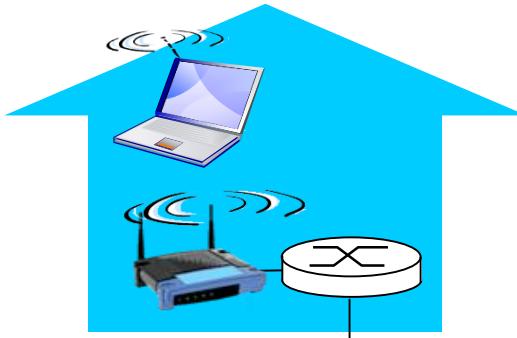
## Mạng truy cập không dây

- Mạng truy cập không dây được chia sẻ kết nối hệ thống đầu cuối đến thiết bị định tuyến

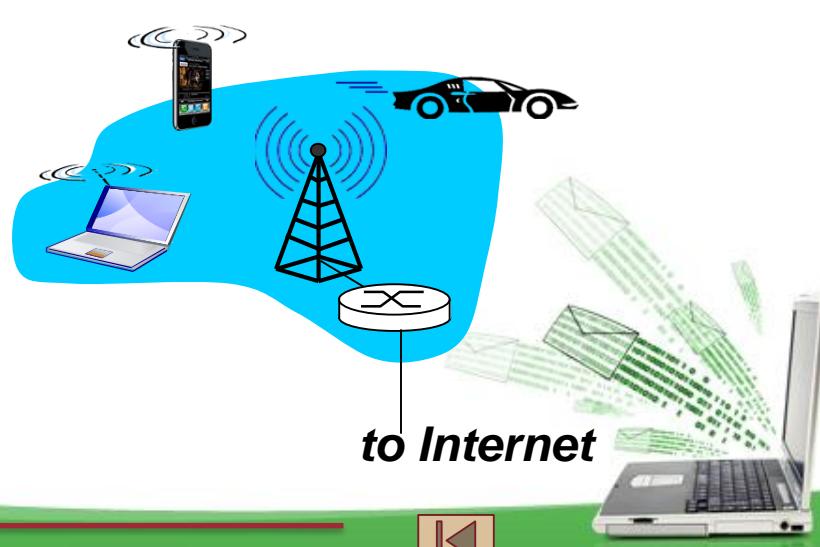
- Thông qua base station còn được gọi là “access point”

### Mạng cục bộ không dây: **wide-area wireless access**

- Trong tòa nhà (100 ft)
- 802.11b/g (WiFi): tốc độ truyền 11, 54 Mbps



**to Internet**



**to Internet**

Internet là gì?

Mạng biển

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

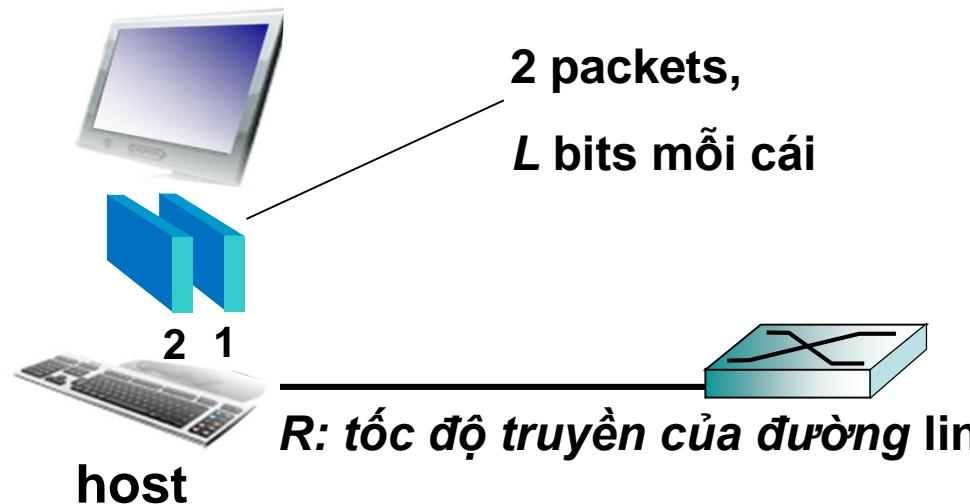
Các lớp giao thức, các mô hình dịch vụ



## Host: gửi các gói dữ liệu

Chức năng host sending:

- Lấy thông tin tầng ứng dụng
- Chia nhỏ thành những phần nhỏ hơn, được biết như là **packets**, chiều dài  **$L$**  bits
- Truyền packet trong mạng truy cập với tốc độ truyền  **$R$** 
  - Tốc độ truyền của đường link, còn được gọi là khả năng/công suất của đường link, **còn được gọi là băng thông của đường link**



$$\text{Độ trễ truyền gói} = \frac{\text{Thời gian cần để}}{\text{Truyền } L\text{-bit packet}} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$



# Đường truyền vật lý

- **bit:** lan truyền giữa các cặp truyền/nhận
- **Liên kết vật lý:** những gì nằm giữa thiết bị truyền và nhận
- **Phương tiện truyền thông được hướng dẫn:**
  - Các tín hiệu lan truyền trên các phương tiện truyền thông rắn: cáp đồng, cáp quang, cáp đồng trực
- **Phương tiện truyền thông không được hướng dẫn:**
  - Tín hiệu lan truyền tự do, ví dụ radio

Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

## Cặp xoắn (TP)

- Hai dây cáp đồng độc lập
  - Loại 5: 100 Mbps, 1 Gbps Ethernet
  - Loại 6: 10Gbps



# Đường truyền vật lý: cáp đồng trục, cáp quang

## Cáp đồng trục:

- Hai dây dẫn đồng đồng tâm
- Hai hướng
- Băng thông rộng:
  - nhiều kênh trên cáp



## Cáp quang:

- ❖ Sợi thủy tinh mang xung ánh sáng, mỗi xung là một bit
- ❖ Hoạt động tốc độ cao:
  - Truyền point-to-point với tốc độ cao (ví dụ., tốc độ truyền 10' s-100' s Gpbs)
- ❖ Tỷ lệ lỗi thấp:
  - Bộ lặp (repeaters) cách xa phần còn lại
  - Không bị nhiễu điện từ



Internet là gì?

Mạng biển

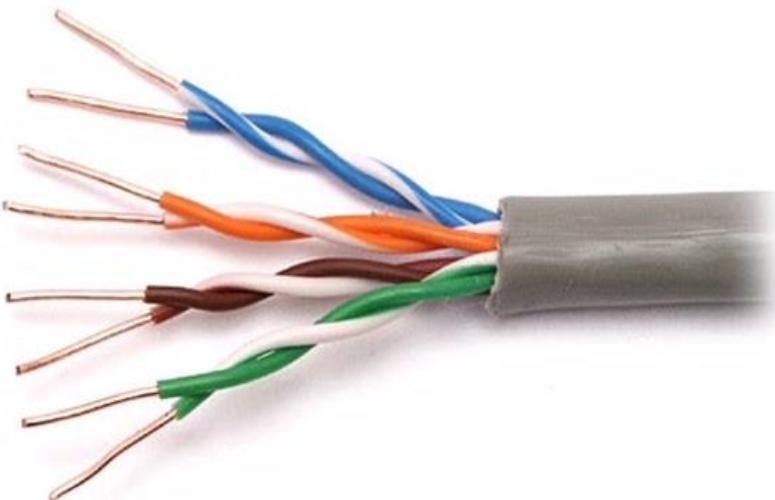
Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

# Đường truyền vật lý: cáp UTP, STP

Cáp UTP (Unshielded Twisted Pair) là loại cáp xoắn đôi không có vỏ bọc chống nhiễu. Cáp UTP gồm các cặp dây được xoắn lại với nhau. Đây là một trong những phương pháp cơ bản nhất được sử dụng để giúp ngăn nhiễu điện từ.



Cáp STP/FTP (Shielded Twisted or Foiled Twisted Pair) là loại cáp xoắn đôi có vỏ bọc chống nhiễu. STP và FTP cung cấp một lớp bảo vệ bổ sung với vỏ bọc quấn quanh các dây xoắn riêng lẻ, giúp chống lại EMI/FRI và xuyên âm.



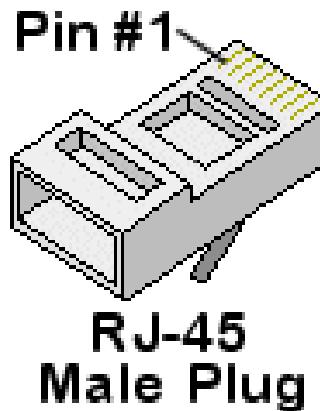
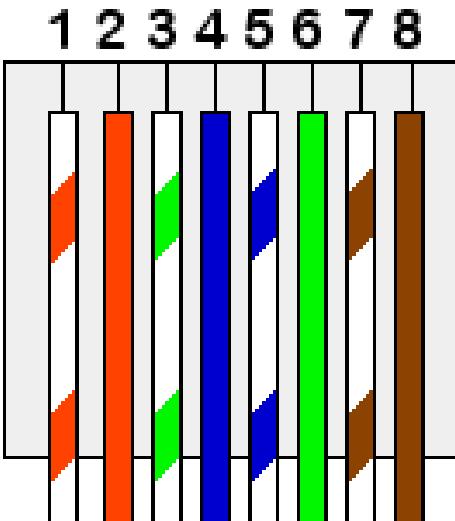
Cáp S/UTP và F/UTP (Shielded/ Screened or Foiled Unshielded Twisted Pair) là loại cáp gồm một vỏ bọc đặc bao quanh 4 cặp cáp xoắn đôi không có vỏ bọc chống nhiễu.



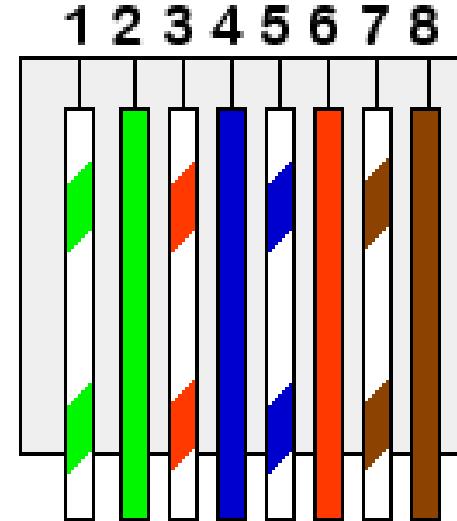
# Đường truyền vật lý: cáp UTP, STP

## RJ-45 Color Code

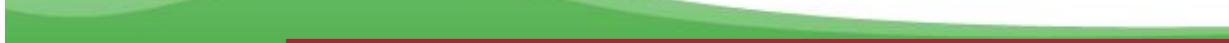
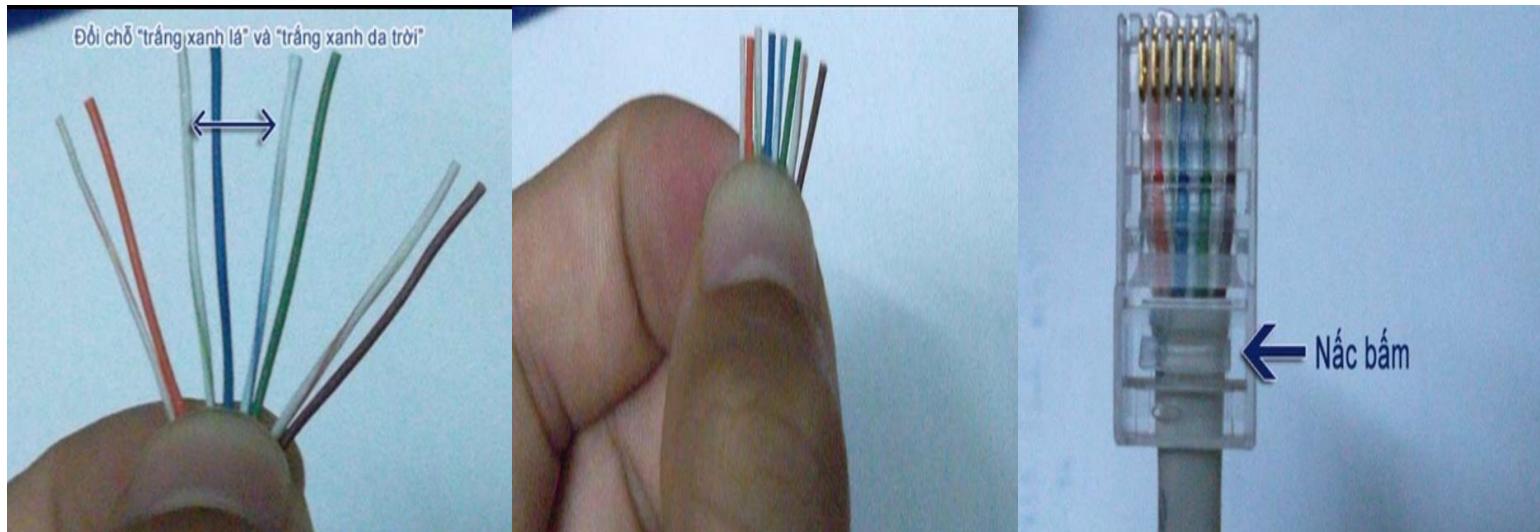
T-568B Standard



T-568A Standard



# Đường truyền vật lý: cáp UTP, STP



# Đường truyền vật lý: sóng radio

- Tín hiệu được mang dưới dạng phổ điện từ
- Không có dây dẫn vật lý
- Truyền hai chiều
- Các tác động môi trường lan truyền:
  - Phản xạ
  - Bị cản trở bởi vật cản
  - Bị nhiễu

## Các loại liên kết radio:

- ❖ **Vi sóng mặt đất**
  - Các kênh đến 45 Mbps
- ❖ **LAN** (như WiFi)
  - 11Mbps, 54 Mbps
- ❖ **Diện rộng** (như cellular)
  - 3G cellular: ~ vài Mbps
- ❖ **Vệ tinh**
  - Kênh từ Kbps đến 45Mbps (hoặc chia nhiều kênh nhỏ)
  - Độ trễ giữa 2 thiết bị đầu cuối 270 msec
- ❖ **giữ khoảng cách cố định với mặt đất, độ cao thấp**

Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Chương 1: nội dung

## 1.1 Internet là gì?

## 1.2 Mạng biên

- hệ thống đầu cuối, mạng truy cập, các liên kết

## 1.3 Mạng lõi

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

## 1.4 Độ trễ, sự mất mát, thông lượng trong mạng

## 1.5 Các lớp giao thức, các mô hình dịch vụ

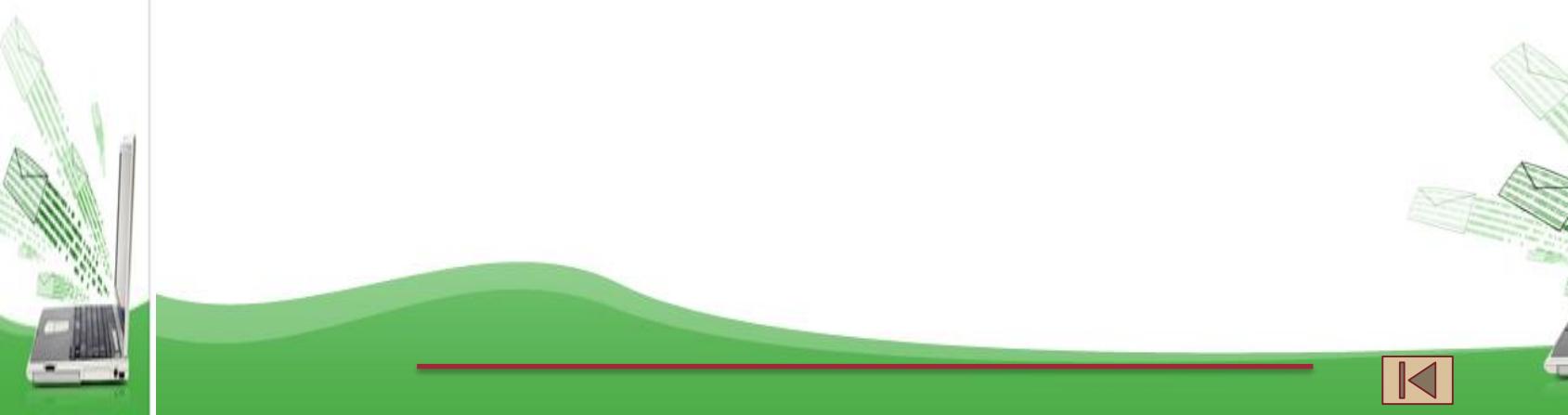
Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



## Mạng lõi

- Gồm các bộ định tuyến được kết nối với nhau
- **Chuyển mạch gói:** hosts chia nhỏ dữ liệu tần ứng dụng (application-layer messages) thành các packets
  - Chuyển tiếp các gói từ một bộ định tuyến này đến bộ định tuyến tiếp theo qua các đường link trên đường đi từ nguồn tới đích.
  - Mỗi packet được truyền tải với công suất lớn nhất của đường link

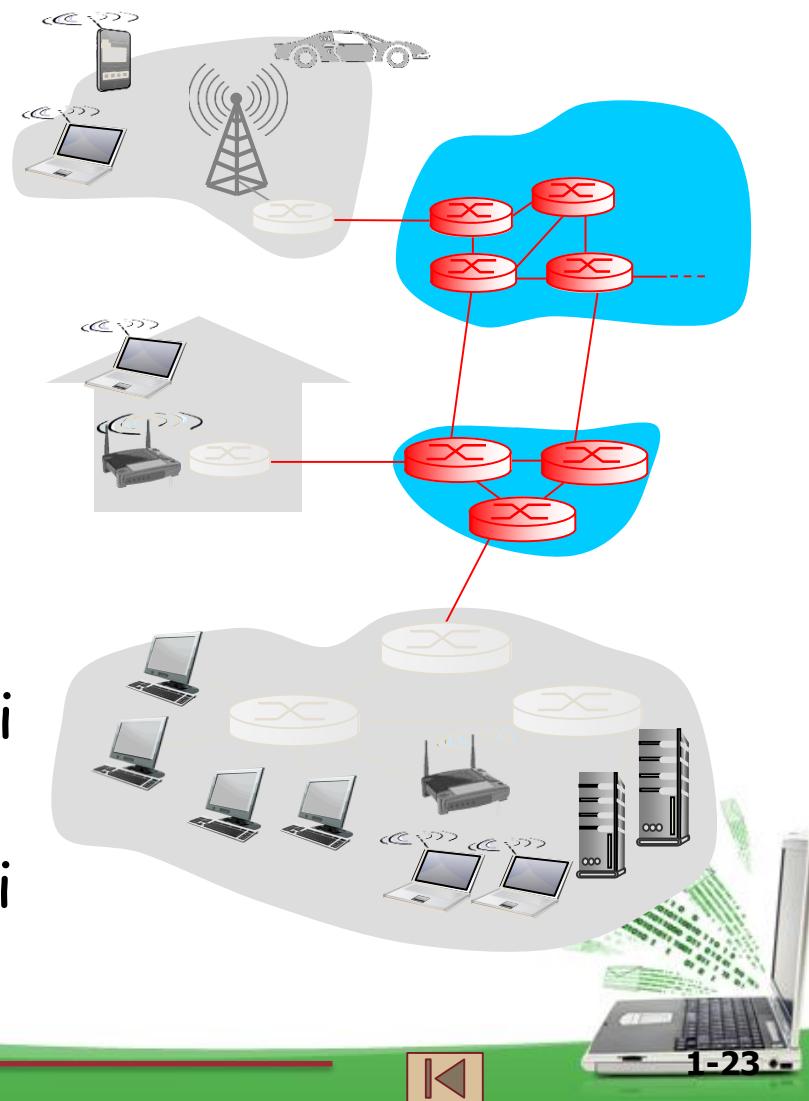
Internet là gì?

Mạng biên

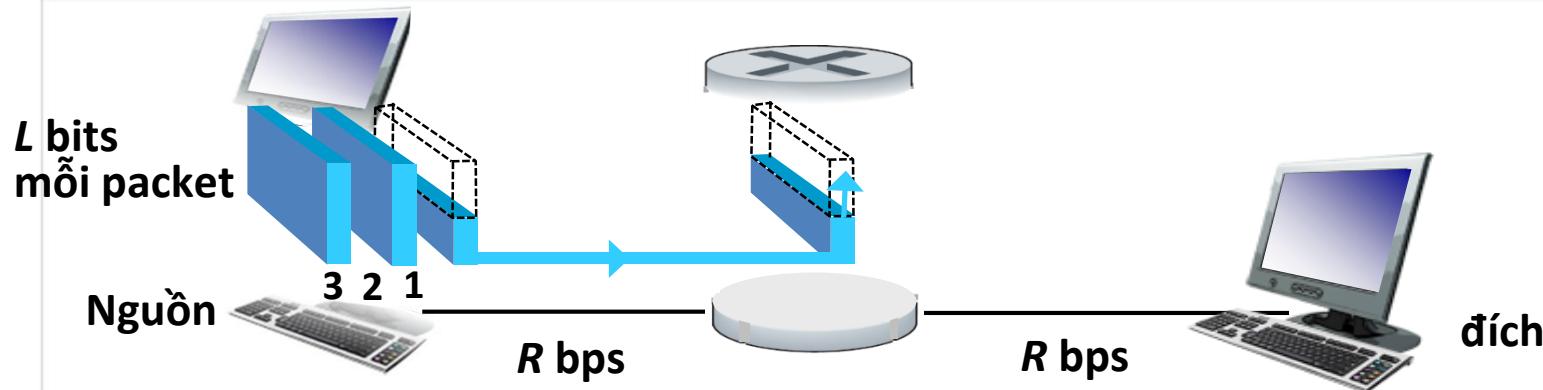
Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



## Chuyển mạch gói: store-and-forward



- Mất  $L/R$  giây để truyền tải  $L$ -bit packet trong đường link tại tốc độ  $R$  bps
- store and forward:** toàn bộ packet phải đến bộ định tuyến trước khi nó có thể được truyền tải trên đường link tiếp theo
- end-end delay =  $2L/R$  (giả sử không có độ trễ lan truyền)

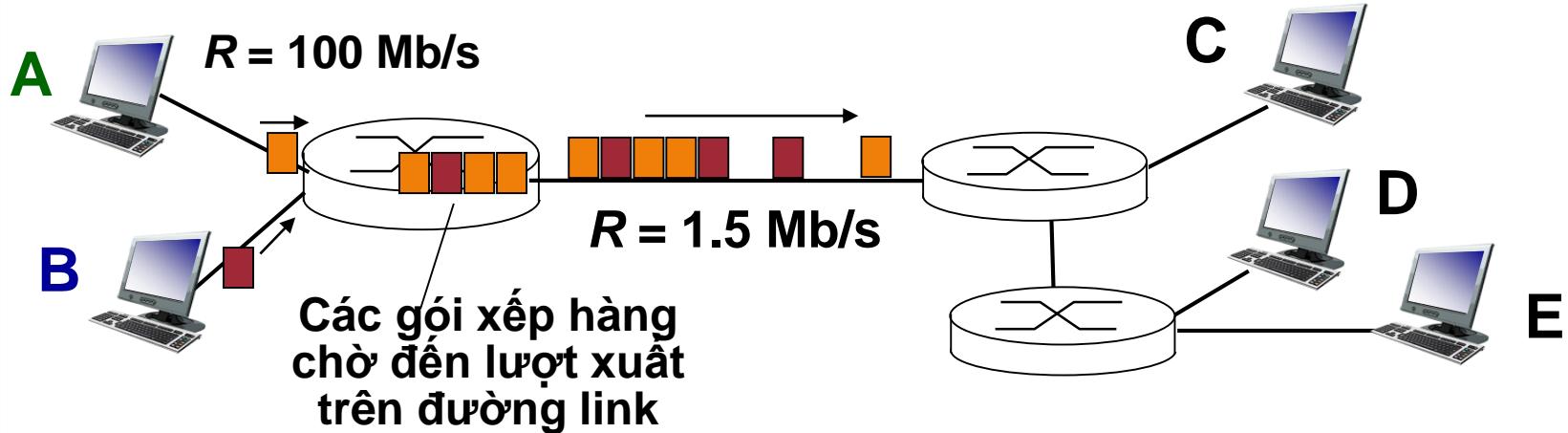
*Ví dụ số về one-hop :*

- $L = 7.5$  Mbits
- $R = 1.5$  Mbps
- Độ trễ truyền tải one-hop = 5 sec

} Thêm về độ trễ ngắn ...



# Chuyển mạch gói: sự mất mát, xếp hàng



## Xếp hàng và sự mất mát:

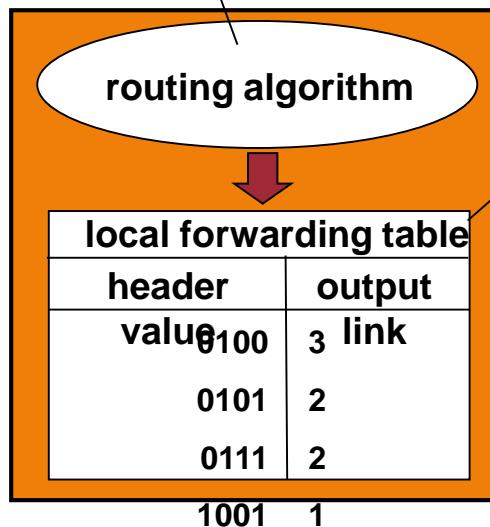
- ❖ Nếu tốc độ đến (theo bit) đến đường link vượt quá tốc độ truyền dẫn của đường link trong một khoảng thời gian:
  - Các packet sẽ xếp hàng và đợi để được truyền tải trên đường link
  - Các packet có thể bị bỏ (bị mất) nếu bộ nhớ (bộ đệm) bị đầy



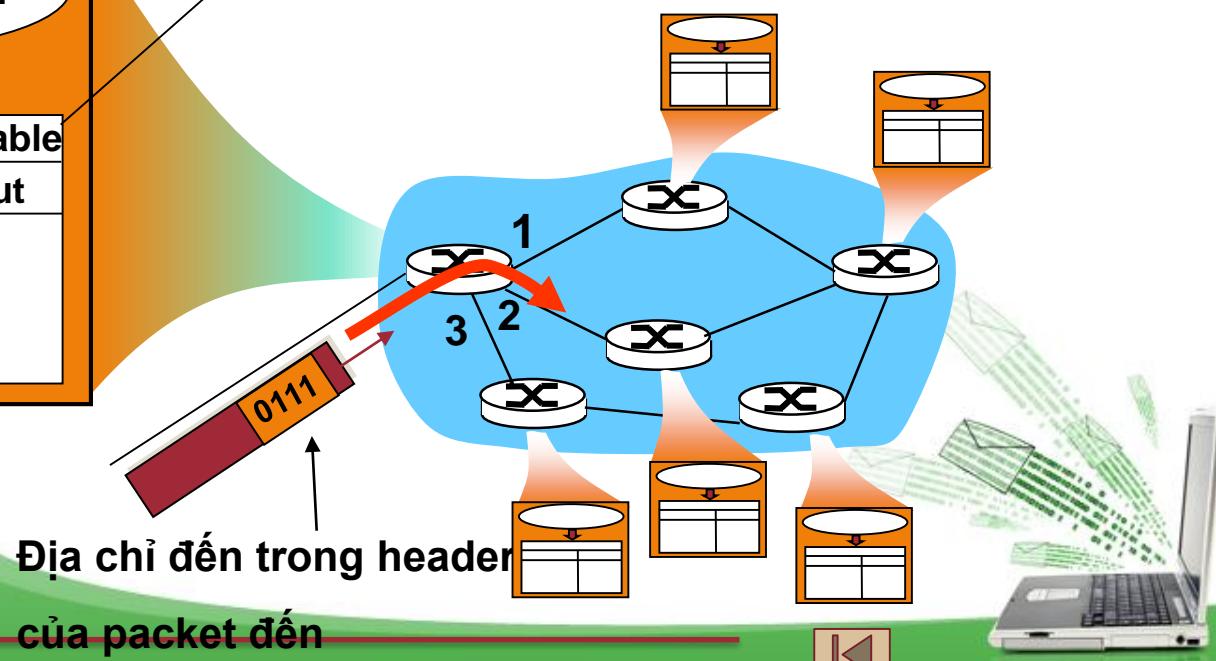
## Hai chức năng chính của mạng lõi

**routing:** xác định đường đi từ nguồn đến đích được thực hiện bởi các packet

- Thuật toán routing



**forwarding:** chuyển các packet từ đầu vào của bộ định tuyến đến đầu ra thích hợp của bộ định tuyến đó



## Alternative core: chuyển mạch kênh

Tài nguyên giữa 2 điểm cuối được phân bổ, được dành cho “cuộc gọi” giữa nguồn và đích:

- Trong sơ đồ, mỗi đường link có bốn kênh.
  - Cuộc gọi dùng kênh thứ 2<sup>nd</sup> trong đường link trên cùng và kênh thứ trong đường link bên phải.
- Tài nguyên được dành riêng : không chia sẻ
  - circuit-like (được đảm bảo) performance
- Mảnh kênh được cấp phát sẽ rảnh rỗi nếu không được sử dụng bởi cuộc gọi (*không chia sẻ*)
- Thường được sử dụng trong các mạng điện thoại truyền thống

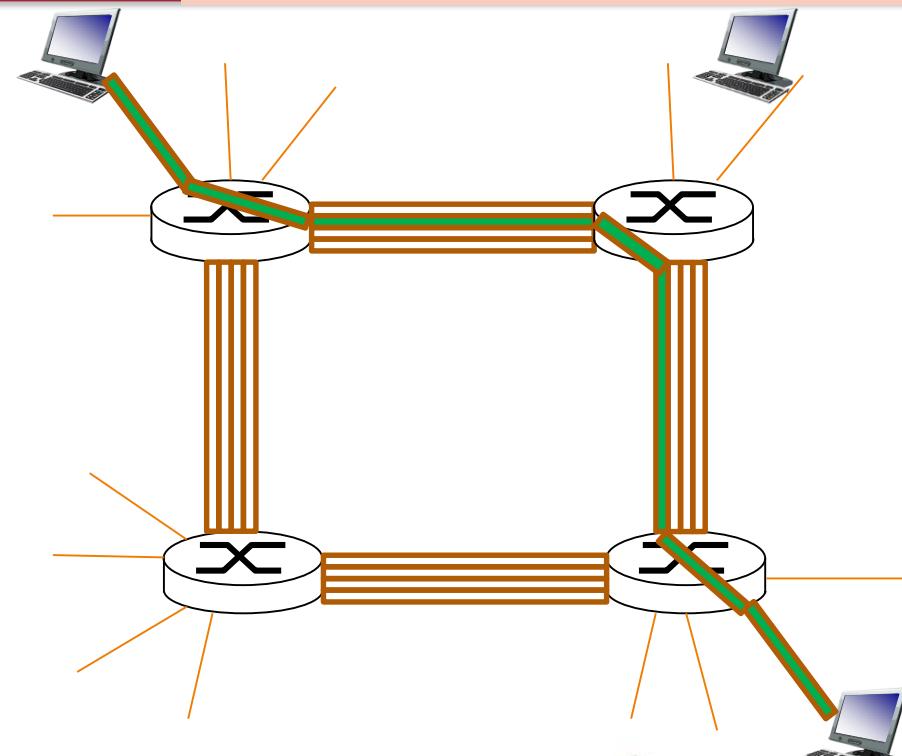
Internet là gì?

Mạng biên

Mạng lõi

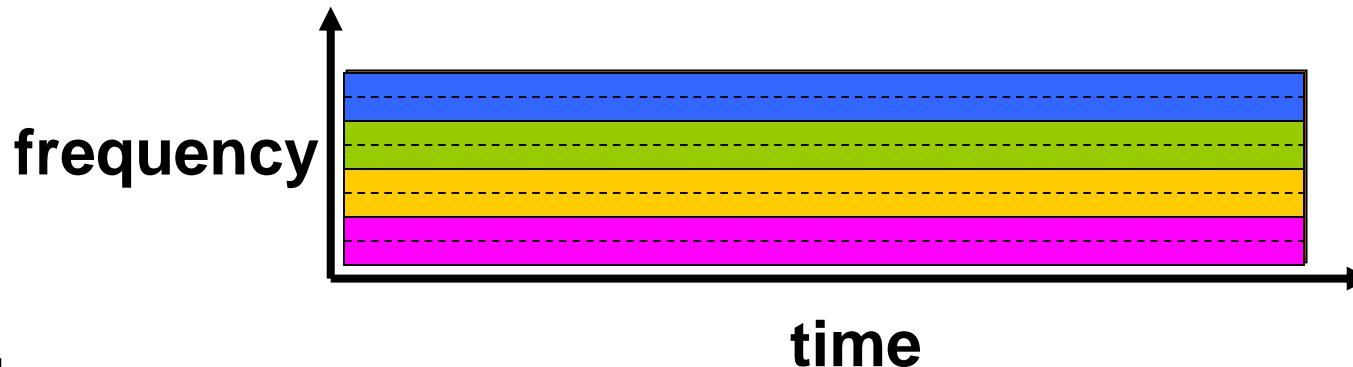
Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

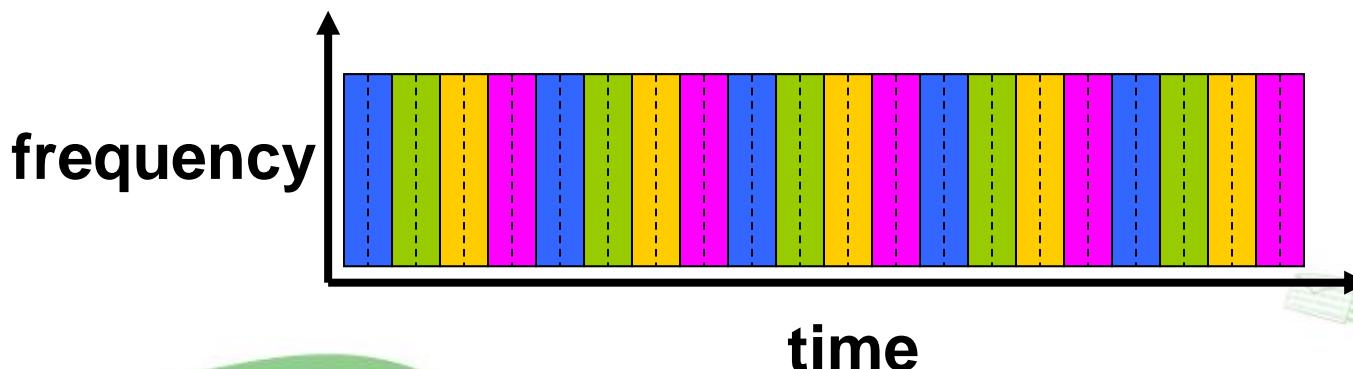


# Chuyển mạch kênh: FDM với TDM

FDM



TDM



Ví dụ:

4 users



Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



## So sánh chuyển mạch gói với chuyển mạch kẽm

Liệu chuyển mạch gói có phải là “kẻ chiến thắng tất cả”?

- Tốt cho trường hợp dữ liệu không được truyền ra với tốc độ đều
  - Chia sẻ tài nguyên
  - Đơn giản, không cần thiết lập cuộc gọi
- Trong trường hợp tắc nghẽn quá mức: packet bị trễ và thất lạc
  - Các giao thức cần thiết cho việc truyền dữ liệu một cách tin cậy và điều khiển tắc nghẽn



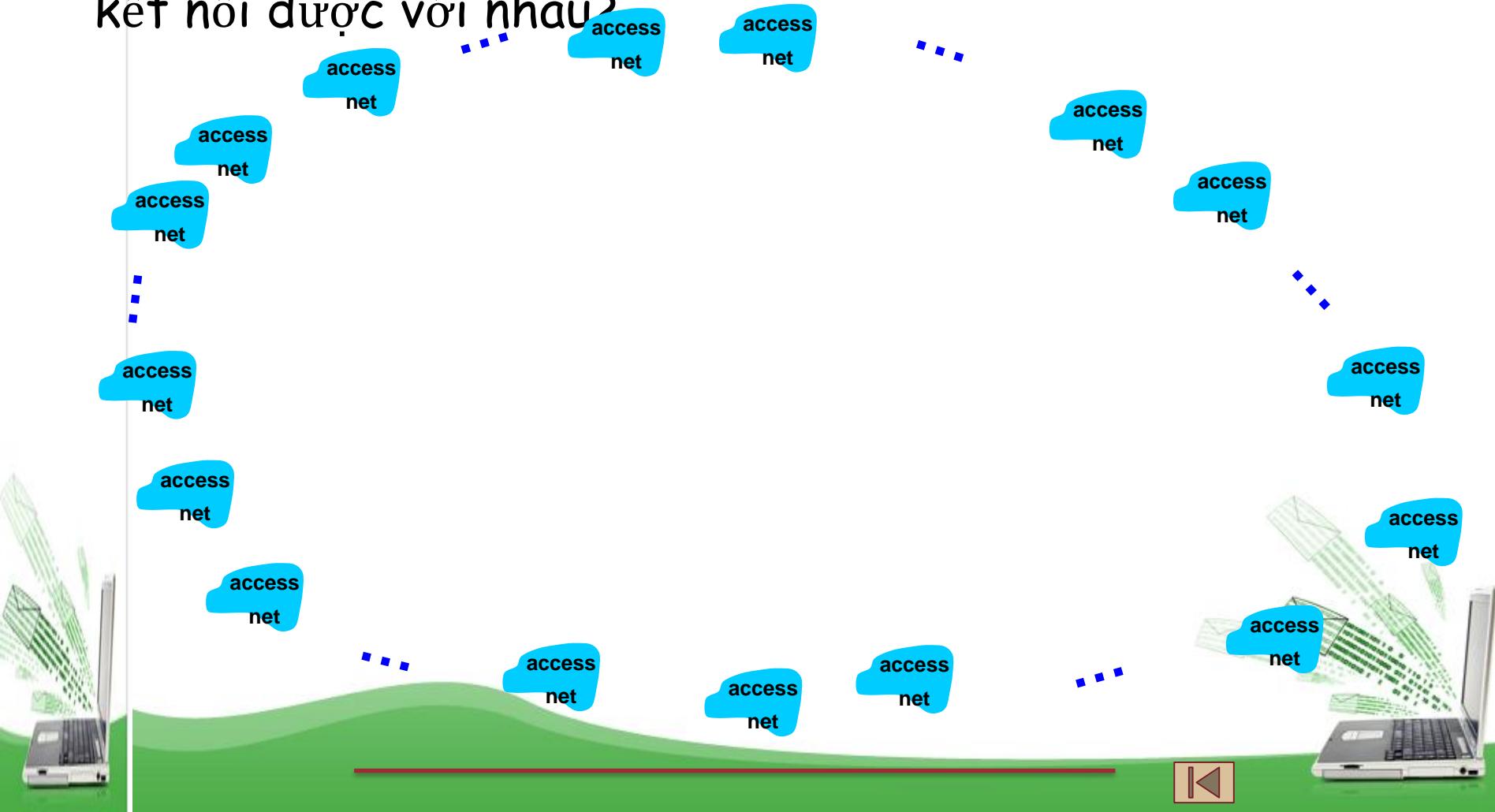
## Kiến trúc Internet: mạng của các mạng

- ❖ Các hệ thống đầu cuối kết nối đến Internet thông qua **access ISPs** (Internet Service Providers)
  - Các ISP khu dân cư, công ty và trường đại học
- ❖ Các Access ISP lần lượt phải được kết nối với nhau
  - ❖ Vì vậy, bất cứ hai host nào cũng có thể gửi được dữ liệu lẫn nhau.
- ❖ Vì vậy, mạng của các mạng rất là phức tạp
  - ❖ Sự phát triển được thúc đẩy bởi **kinh tế và chính sách quốc gia**
- ❖ Chúng ta hãy từng bước khám phá cấu trúc Internet hiện tại



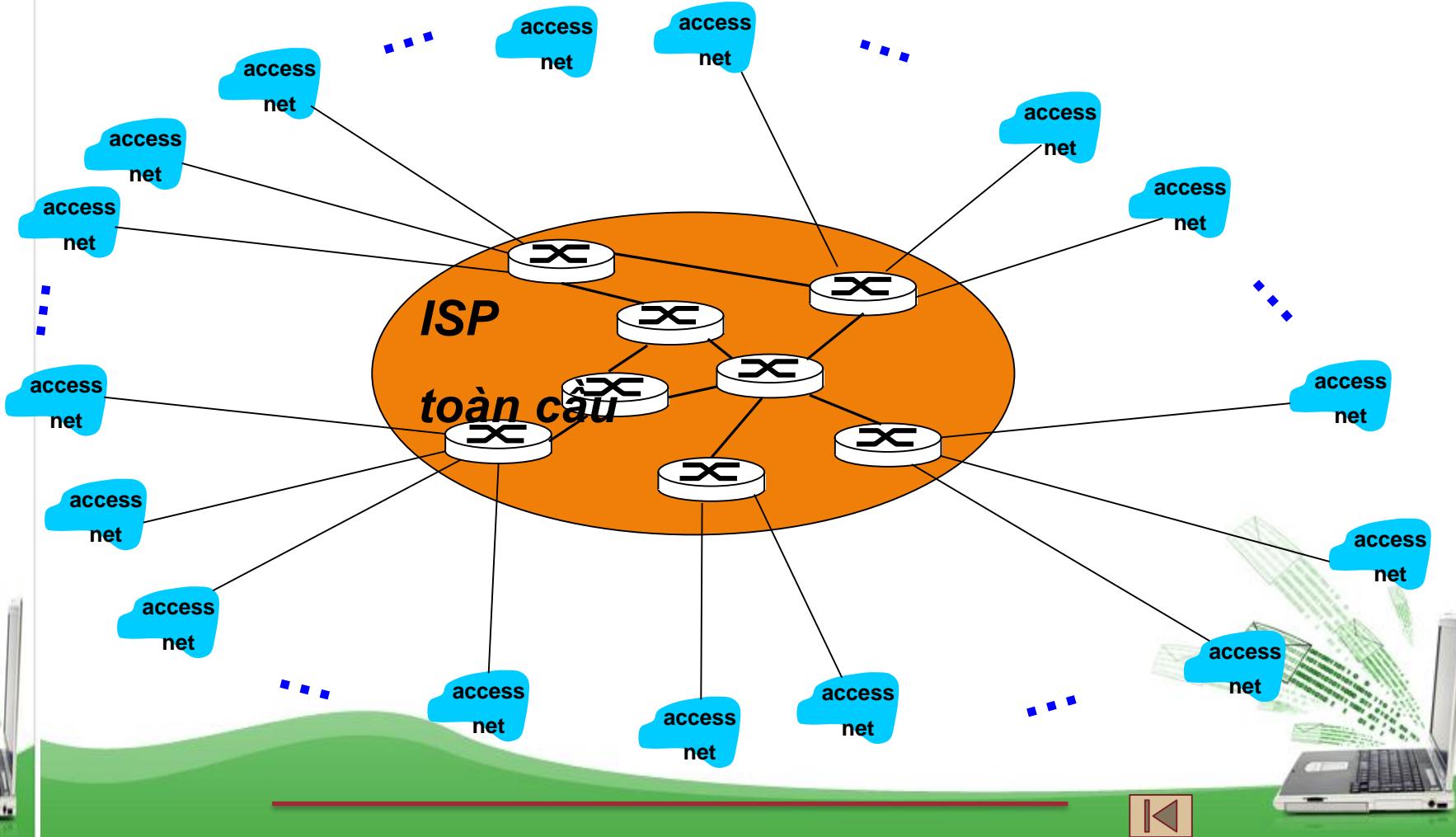
## Kiến trúc Internet: mạng của các mạng

*Câu hỏi:* có hàng triệu access ISPs, làm thế nào để chúng kết nối được với nhau?



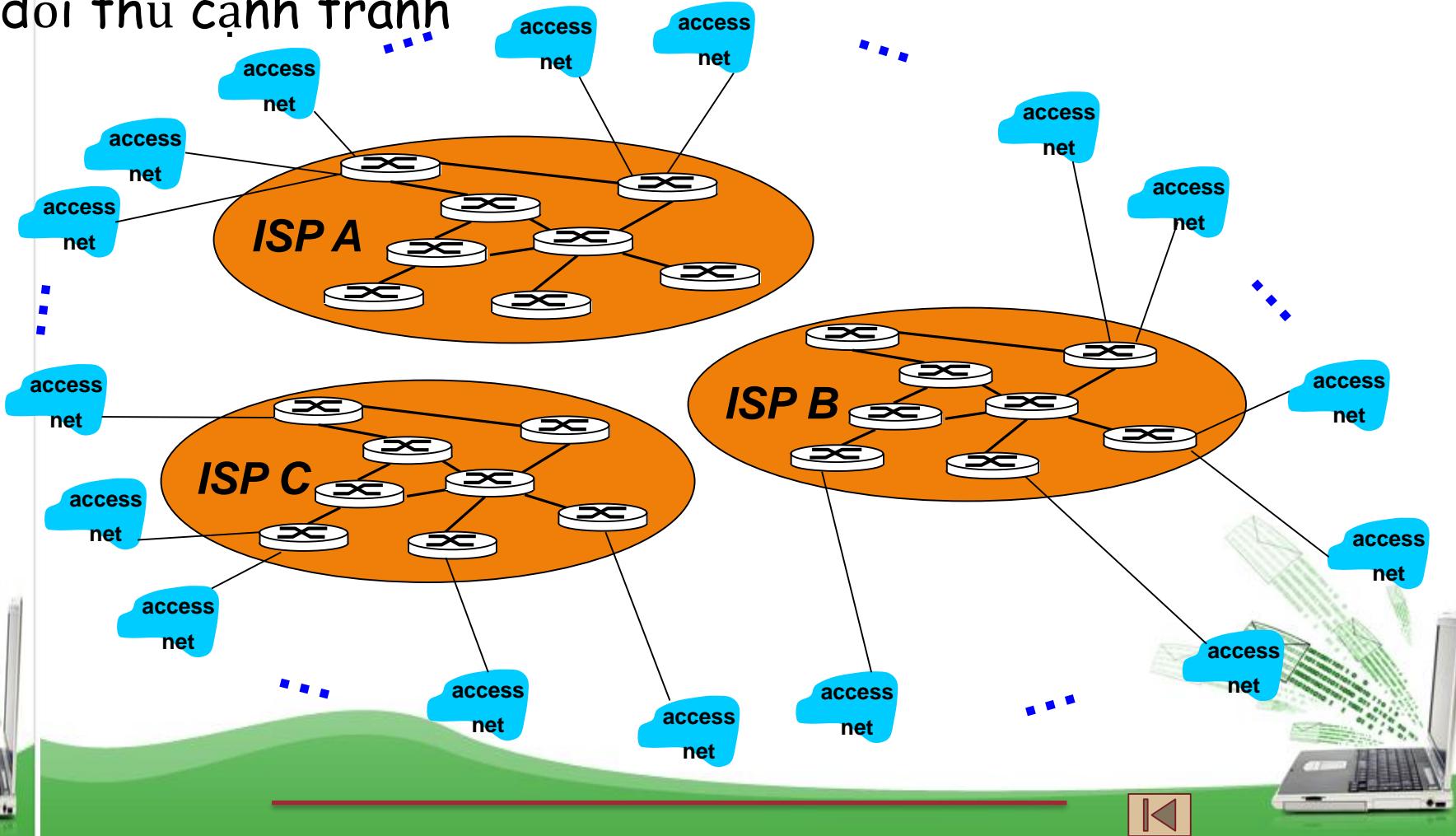
## Kiến trúc Internet: mạng của các mạng

Lựa chọn: kết nối từng access ISP đến một ISP chuyển tiếp toàn cầu?  
Khách hàng và nhà cung cấp dịch vụ ISP phải thoả thuận về kinh tế.



## Kiến trúc Internet: mạng của các mạng

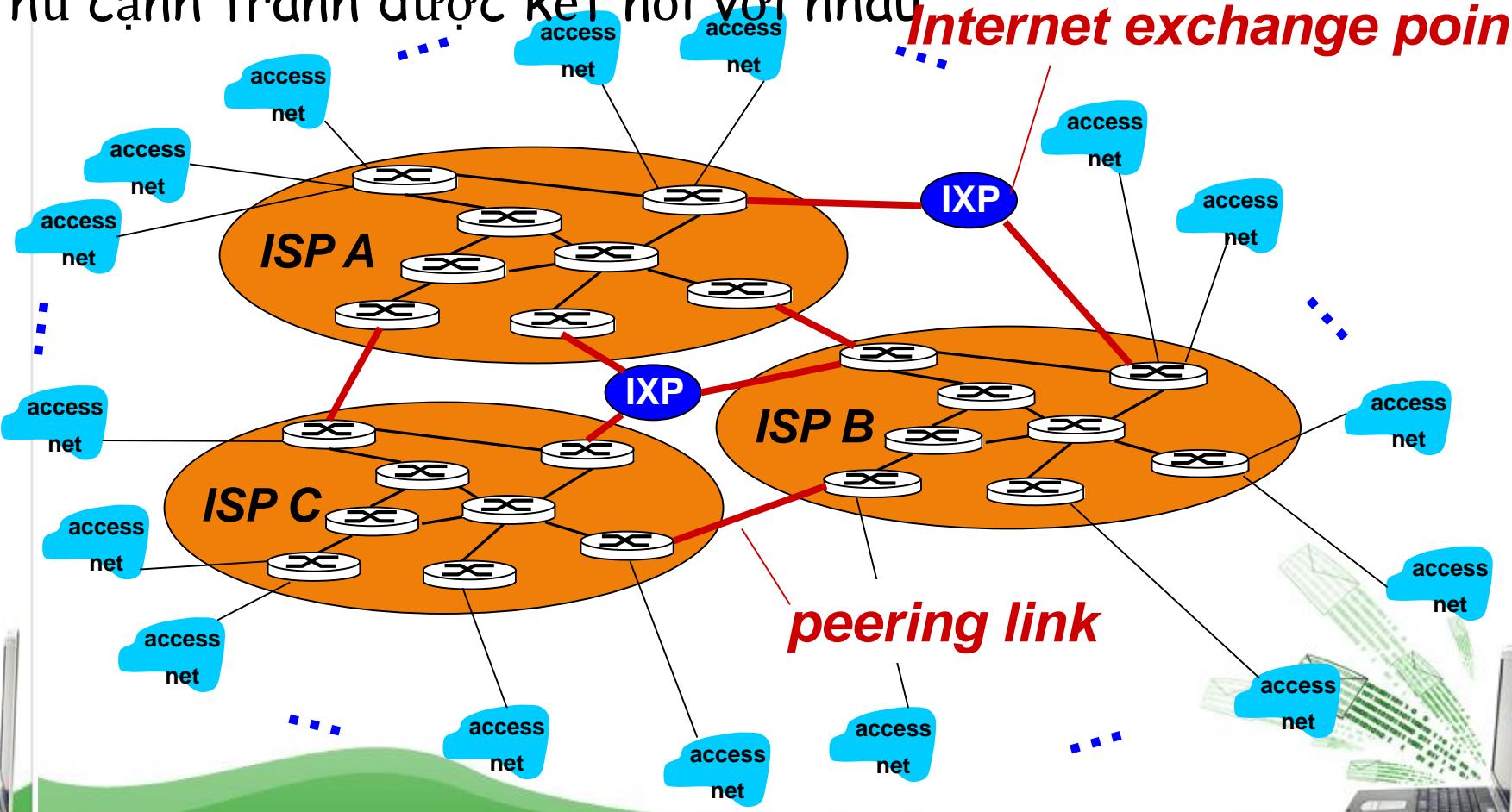
Nhưng nếu một ISP toàn cầu là khả thi, thì sẽ có nhiều đối thủ cạnh tranh



## Kiến trúc Internet: mạng của các mạng

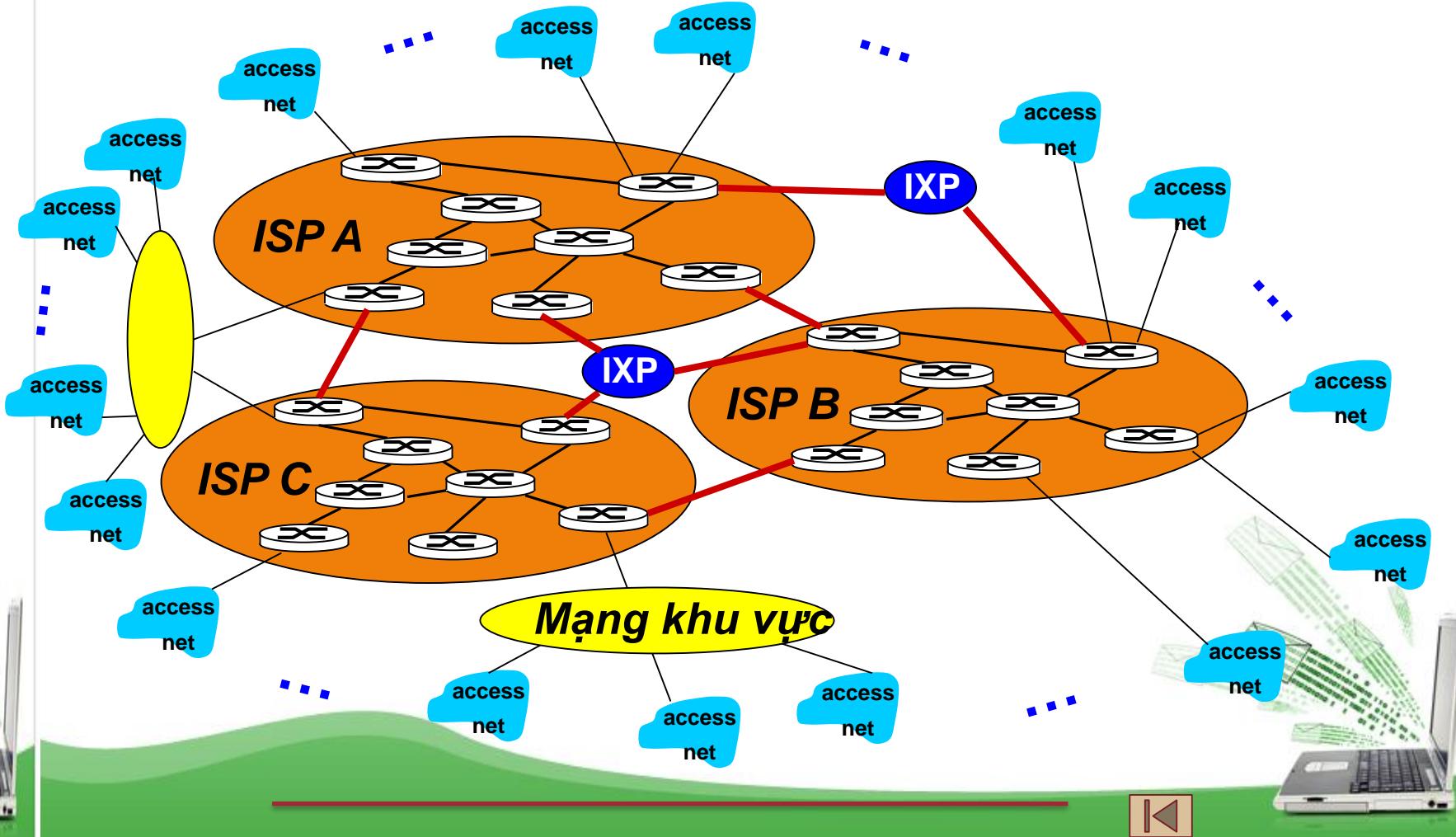
Nhưng nếu một ISP toàn cầu là khả thi, thì sẽ có các đối thủ cạnh tranh được kết nối với nhau.

**Internet exchange point**



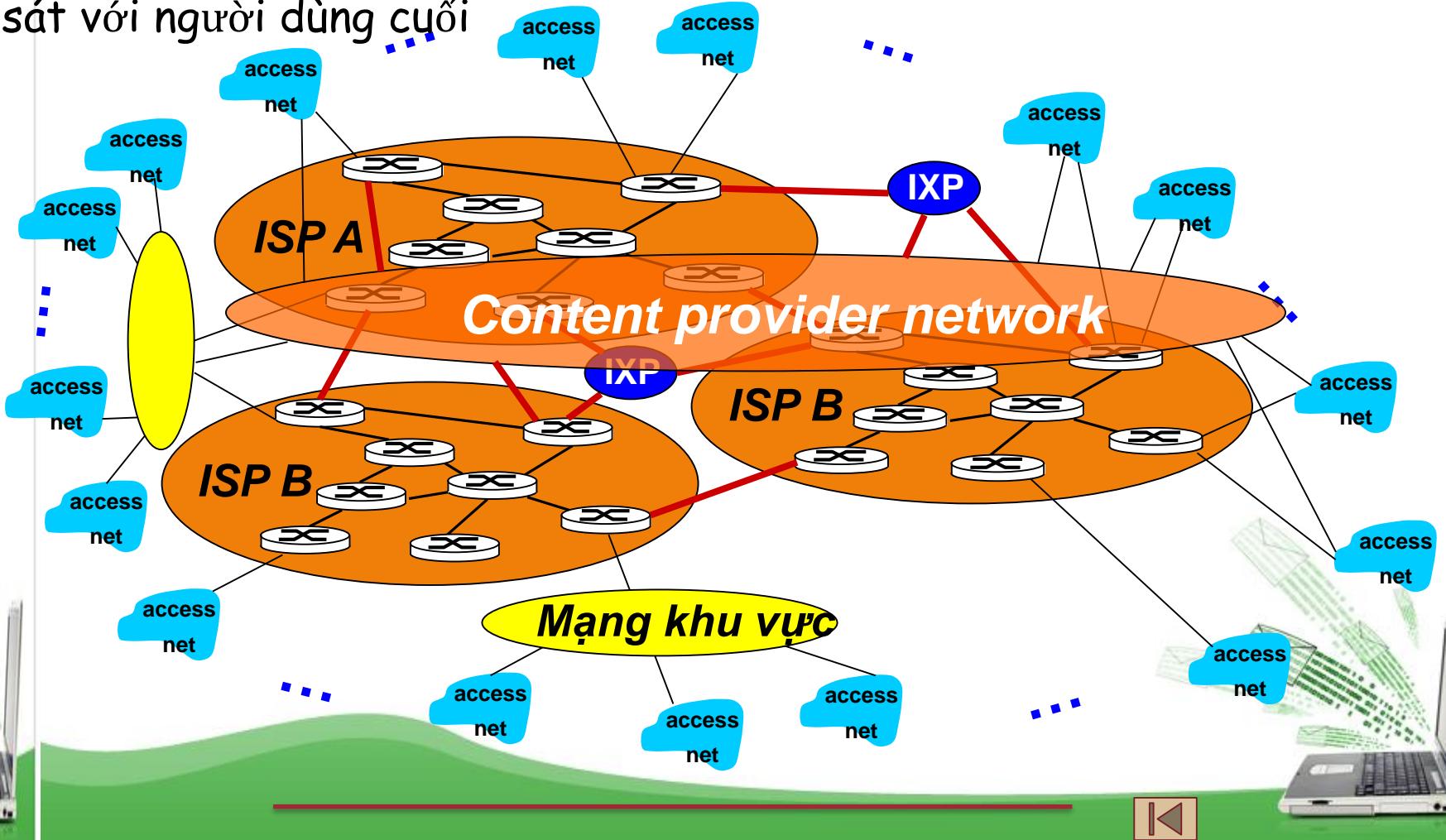
## Kiến trúc Internet: mạng của các mạng

... và các mạng khu vực có thể phát sinh để kết nối các mạng lưới truy cập này đến các ISP

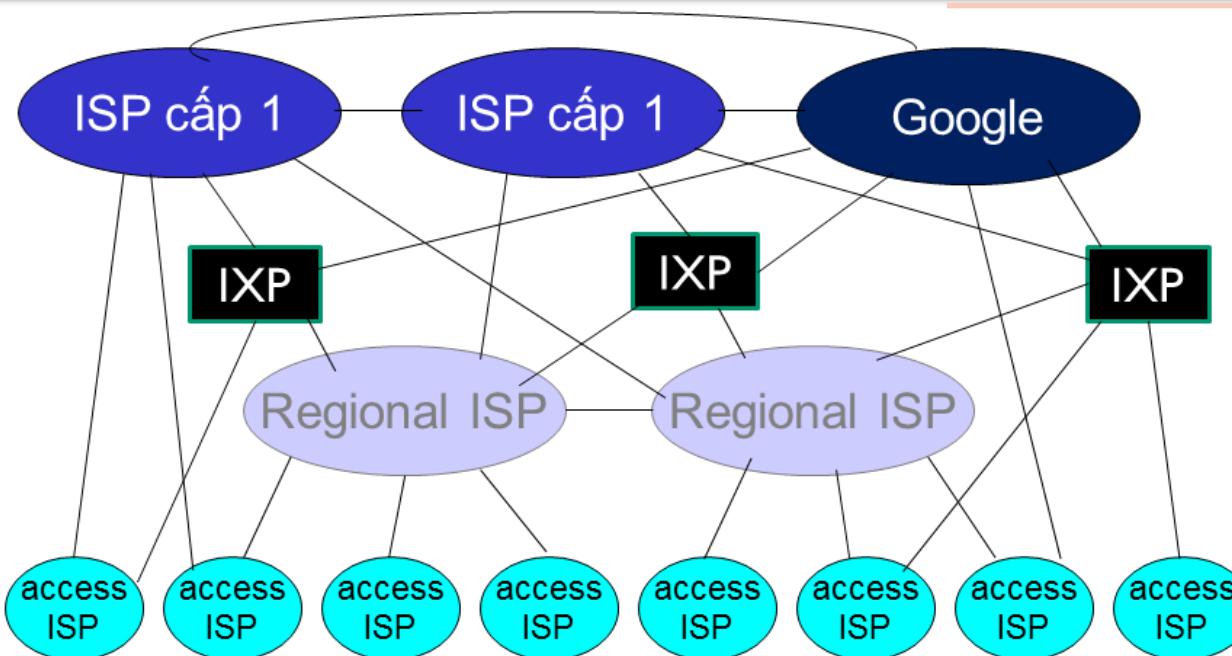


# Kiến trúc Internet: mạng của các mạng

... và các mạng cung cấp nội dung (như là Google, Microsoft, Akamai ) có thể chạy mạng riêng của họ để mang lại các dịch vụ và nội dung gần sát với người dùng cuối



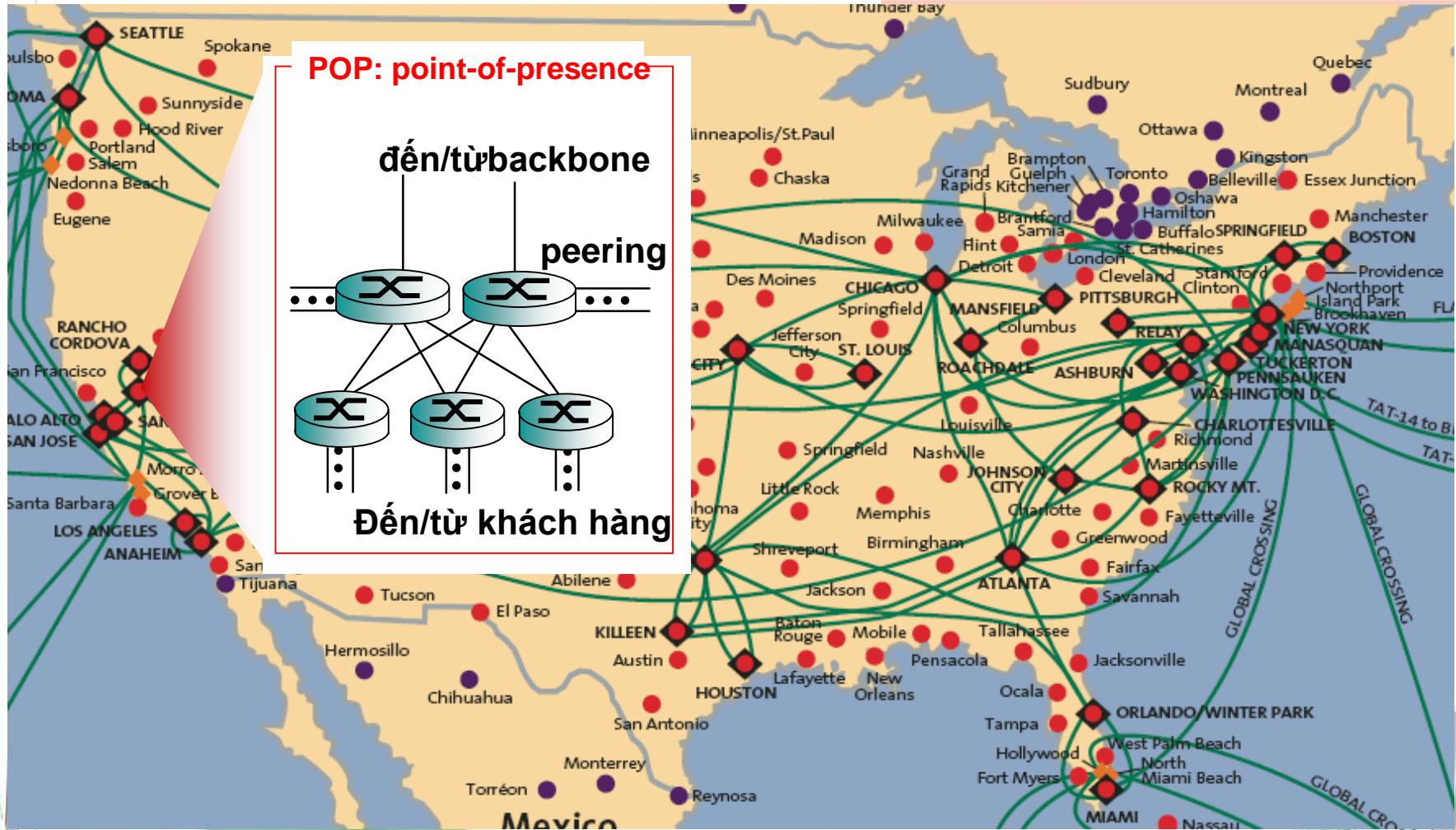
# Kiến trúc Internet: mạng của các mạng



- Tại trung tâm: một số lượng nhỏ các mạng lớn được kết nối với nhau
  - **ISPs thương mại “lớp-1”** (như là Level 3, Sprint, AT&T, NTT), bao trùm các quốc gia và toàn thế giới
  - **Content provider network** (như là Google): mạng riêng kết nối các trung tâm dữ liệu của nó với Internet, thường bỏ qua các IPS vùng và ISP cấp 1.



# ISP cấp 1: như là Sprint



Internet là gì?

Mạng biển

Mạng lõi

Độ trễ, sự mêt mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Chương 1: nội dung

1.1 Internet là gì?

1.2 Mạng biên

- Các hệ thống đầu cuối, mạng truy nhập, các liên kết

1.3 Mạng lõi

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

1.4 Độ trễ, sự mất mát, thông lượng trong mạng

1.5 Các lớp giao thức, các mô hình dịch vụ

Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

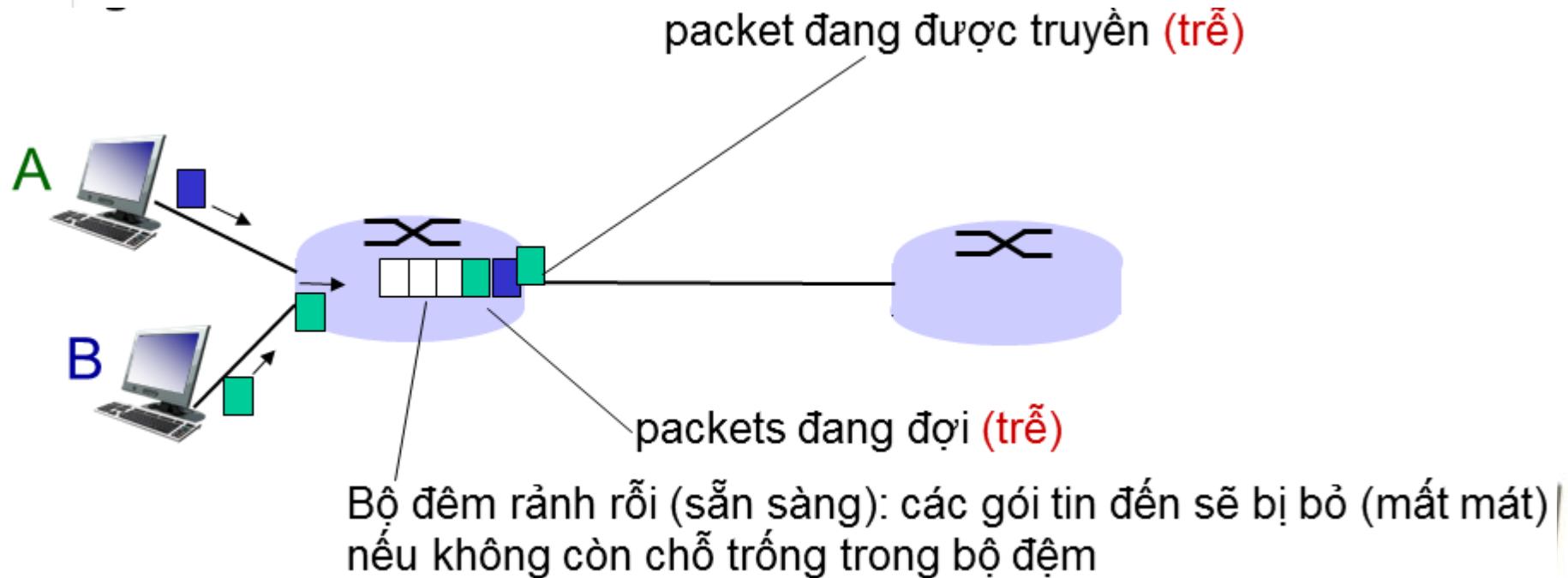
Các lớp giao thức, các mô hình dịch vụ



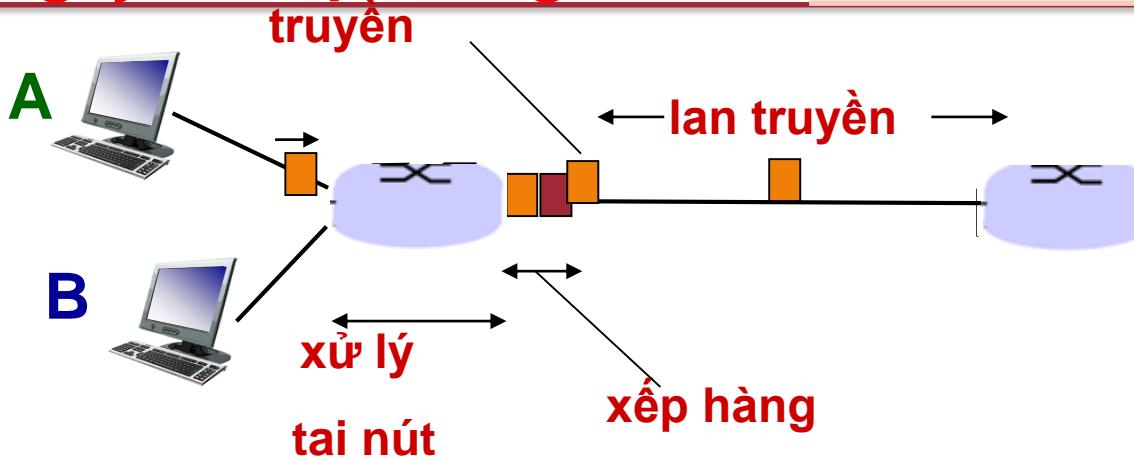
# Sự mất mát và độ trễ xảy ra như thế nào?

Các gói tin (packet) đợi trong bộ đệm của bộ định tuyến (router)

- Tốc độ đến của các gói tin đến đường link (tạm thời) vượt quá khả năng của đường liên kết đầu ra
- Các gói tin đợi và chờ đến lượt



## Bốn nguồn gây ra chậm trễ gói tin



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

$d_{\text{proc}}$ : xử lý tại nút

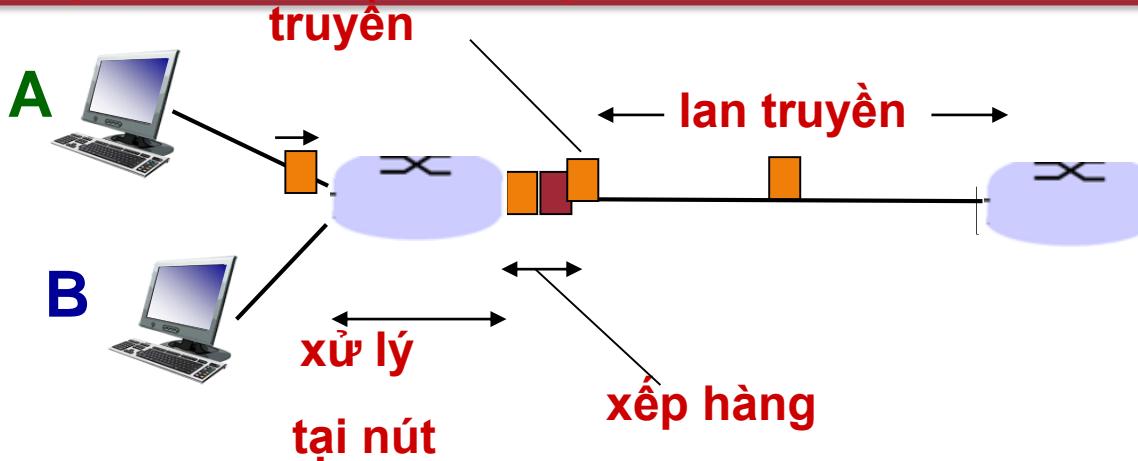
- Kiểm tra các bit lỗi
- Xác định đường ra
- Thông thường < msec

$d_{\text{queue}}$ : độ trễ xếp hàng

- Thời gian đợi tại cổng ra cho việc truyền dữ liệu
- Phụ thuộc vào mức độ tắc nghẽn của bộ định tuyến



# Bốn nguồn gây ra chậm trễ gói tin



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

$d_{\text{trans}}$ : trễ do truyền:

- $L$ : chiều dài gói (bits)
- $R$ : băng thông đường link (bps)
- $d_{\text{trans}} = L/R$

$d_{\text{trans}}$  and  $d_{\text{prop}}$

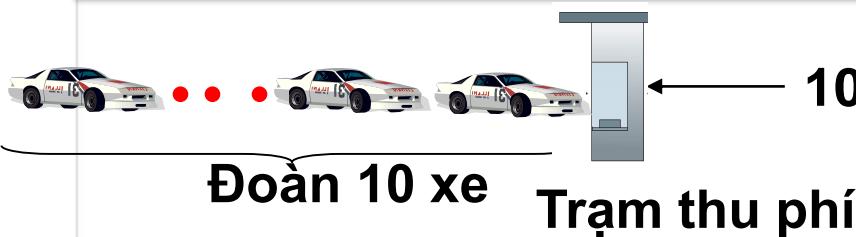
rất khác nhau

$d_{\text{prop}}$ : trễ do lan truyền:

- $d$ : độ dài của đường link vật lý
- $s$ : tốc độ lan truyền trong môi trường (thiết bị, dây dẫn) ( $\sim 2 \times 10^8$  m/sec)
- $d_{\text{prop}} = d/s$



## So sánh với đoàn xe



- Các xe “lan truyền” với tốc độ 100 km/hr
- Trạm thu phí xử lý mỗi xe là 12 giây ( thời gian truyền bit)
- xe~bit; đoàn xe~ packet
- **Q: Mất bao lâu thì đoàn xe tới trạm thu phí thứ 2?**

- Thời gian để “đẩy” toàn bộ đoàn xe qua trạm thu phí qua đường cao tốc =  $12*10 = 120 \text{ sec}$
- Thời gian để xe cuối cùng lan truyền từ trạm thu phí số 1 đến trạm thu phí số 2 là  $100\text{km}/(100\text{km/hr})= 1 \text{ hr}$
- **A: 62 phút**

Internet là gì?

Mạng biển

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

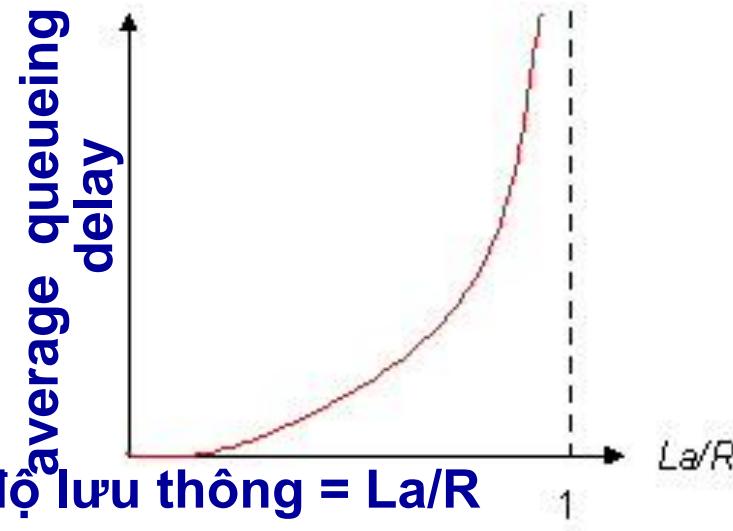
Các lớp giao thức, các mô hình dịch vụ

## Trễ do xếp hàng (nhắc lại)

- $R$ : băng thông đường link (bps)
- $L$ : độ dài gói tin (bits)
- $a$ : tỷ lệ trung bình gói tin đến

cường độ lưu thông =  $La/R$

- ❖  $La/R \sim 0$ : trễ trung bình nhỏ
- ❖  $La/R \rightarrow 1$ : trễ trung bình lớn
- ❖  $La/R > 1$ : nhiều “việc” đến hơn khả năng phục vụ, trễ trung bình vô hạn!



$La/R \sim 0$



$La/R \rightarrow 1$

Internet là gì?

Mạng biển

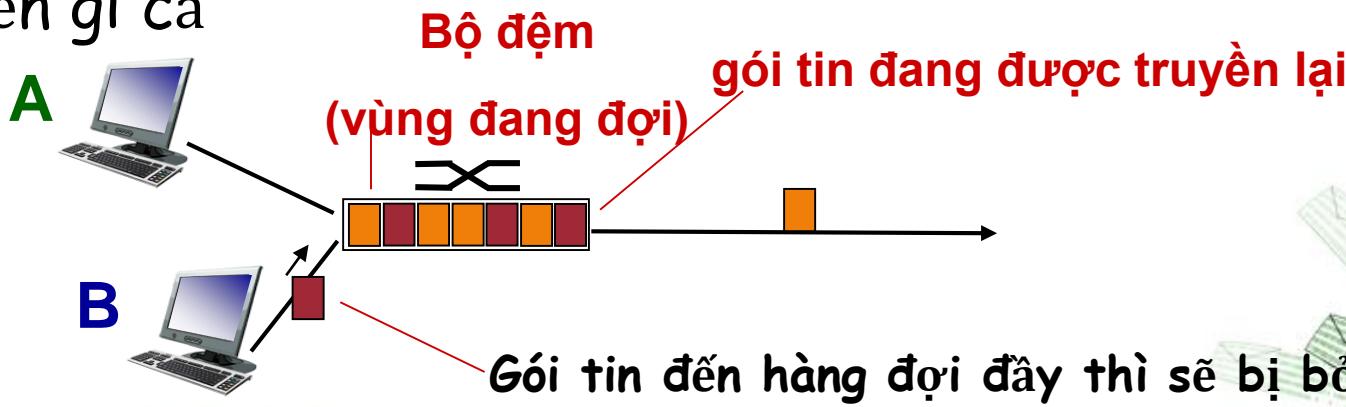
Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

# Sự mất gói

- Hàng đợi (còn gọi là bộ đệm) trước đường link trong bộ đệm có khả năng hữu hạn.
- Gói tin đến hàng đợi đầy thì sẽ bị bỏ (còn gọi là sự mất mát)
- Gói tin bị mất có thể được truyền lại bởi nút mạng trước đó, hay bởi hệ thống đầu cuối nguồn hoặc không truyền gì cả



Internet là gì?

Mạng biển

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

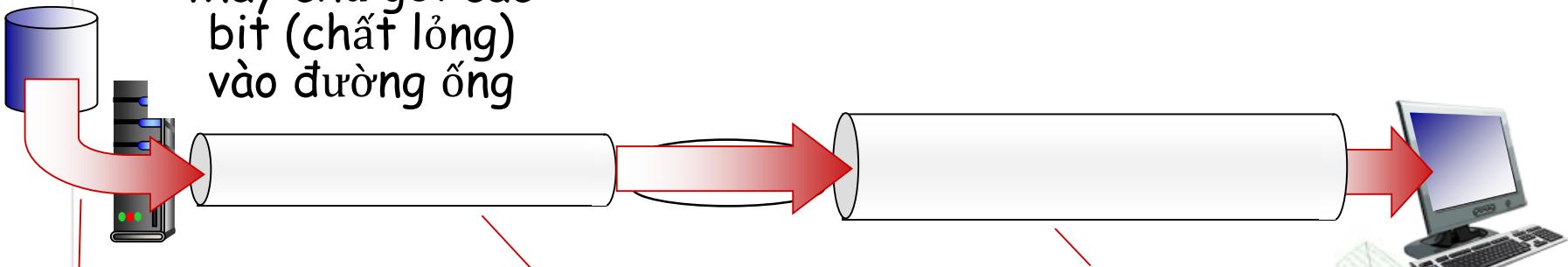


# Thông lượng

[Internet là gì?](#)[Mạng biển](#)[Mạng lõi](#)[Độ trễ, sự mất mát, thông lượng trong mạng](#)[Các lớp giao thức, các mô hình dịch vụ](#)

- **Thông lượng:** tốc độ(bits/time unit) mà các bit được truyền giữa người gửi và nhận
  - **tức thời:** tốc độ tại thời điểm được cho
  - **trung bình:** tốc độ trong thời gian dài hơn

Máy chủ gửi các bit (chất lỏng) vào đường ống



server, with file of F bits to send to client

đường ống có thể mang chất lỏng với tốc độ

$R_s$  bits/sec

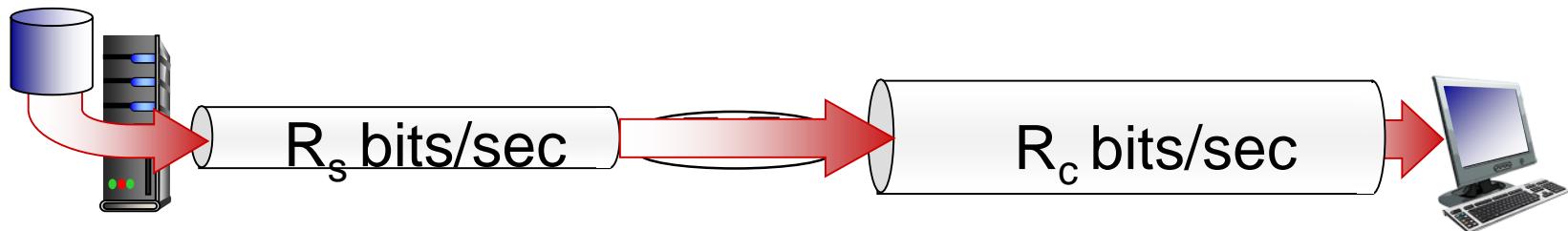
Đường ống có thể mang chất lỏng với tốc độ

$R_c$  bits/sec

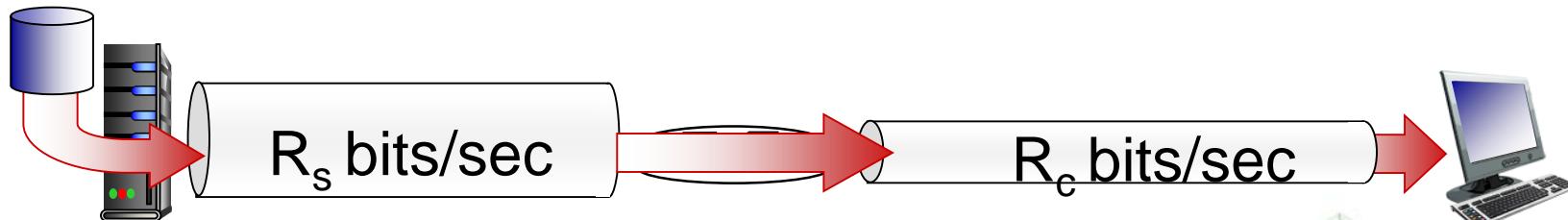


## Thông lượng (tt)

- $R_s < R_c$  thông lượng trung bình giữa 2 đầu cuối là gì?



$R_s > R_c$  thông lượng trung bình giữa 2 đầu cuối là gì?



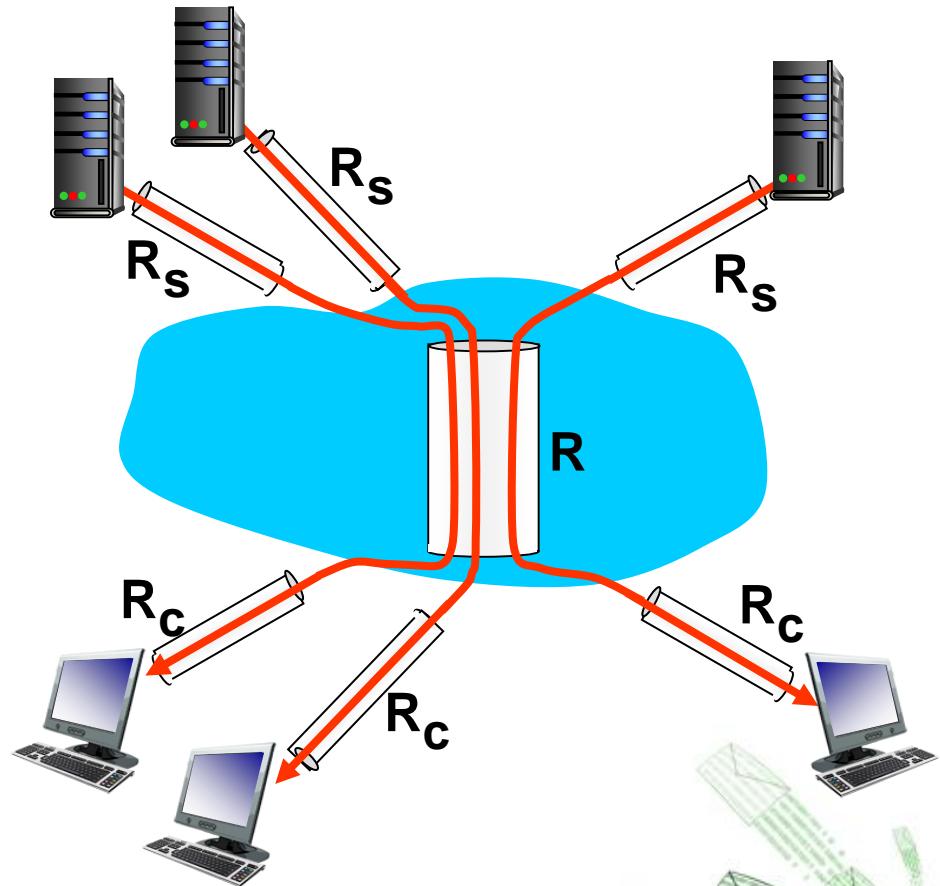
*Đường link nút cổ chai*

Đường link trên con đường từ điểm cuối này đến điểm cuối kia hạn chế thông lượng từ điểm cuối này đến điểm cuối kia



## Thông lượng: Ngữ cảnh Internet

- Thông lượng điểm cuối-cuối cho mỗi kết nối :  
 $\min(R_c, R_s, R/10)$
- Trong thực tế:  $R_c$  hoặc  $R_s$  thường bị thắt nút cổ chai



10 kết nối (công bằng) chia sẻ đường  
trục thắt nút cổ chai  $R$  bits/sec



# Chương 1: Nội dung

## 1.1 Internet là gì?

## 1.2 Mạng biên

- Các hệ thống đầu cuối, mạng truy nhập, các liên kết

## 1.3 Mạng lõi

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

## 1.4 Độ trễ, sự mất mát, thông lượng trong mạng

## 1.5 Các lớp giao thức, các mô hình dịch vụ

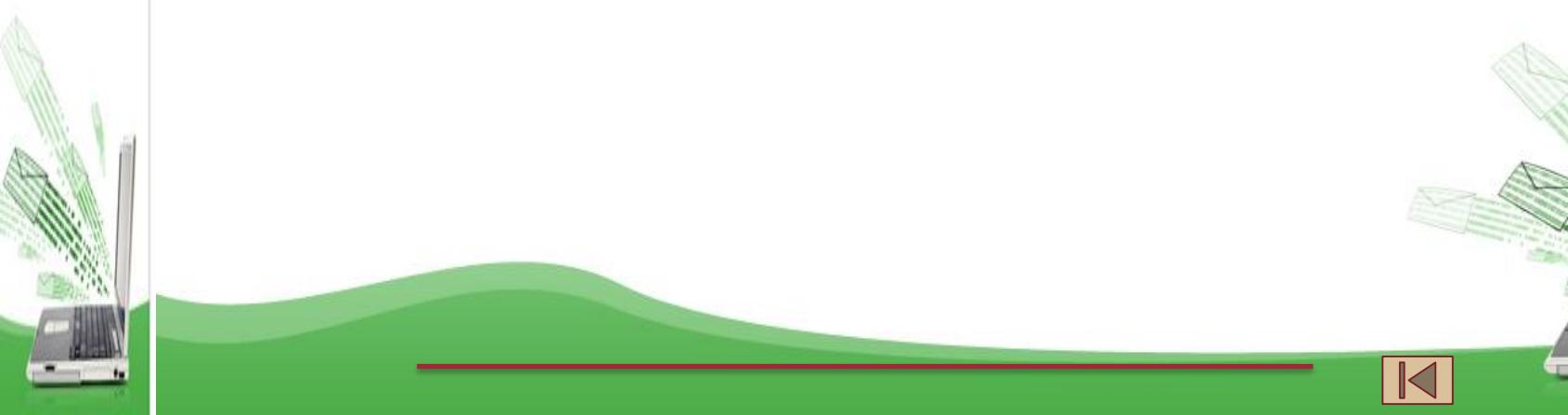
Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



## “Tầng” giao thức

Mạng rất phức tạp, với  
nhiều “miếng”

- Hosts
- Bộ định tuyến  
(routers)
- Các đường link của  
phương tiện  
truyền thông khác  
nhau
- Các ứng dụng
- Các giao thức
- Phần cứng, phần  
mềm

Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

*Câu hỏi:*

Có hy vọng nào để tổ chức  
cấu trúc của mạng hay  
không?

.... Hoặc là ít nhất chúng ta  
thảo luận về mạng



# Tổ chức du lịch hàng không

vé (mua)

hành lý (kiểm tra)

cỗng (tải)

đường băng cất cánh

lộ trình bay

lộ trình bay

Vé (khiếu nại)

hành lý (lấy lại)

cỗng (không tải)

đường băng hạ cánh

lộ trình bay

Internet là gì?

Mạng biên

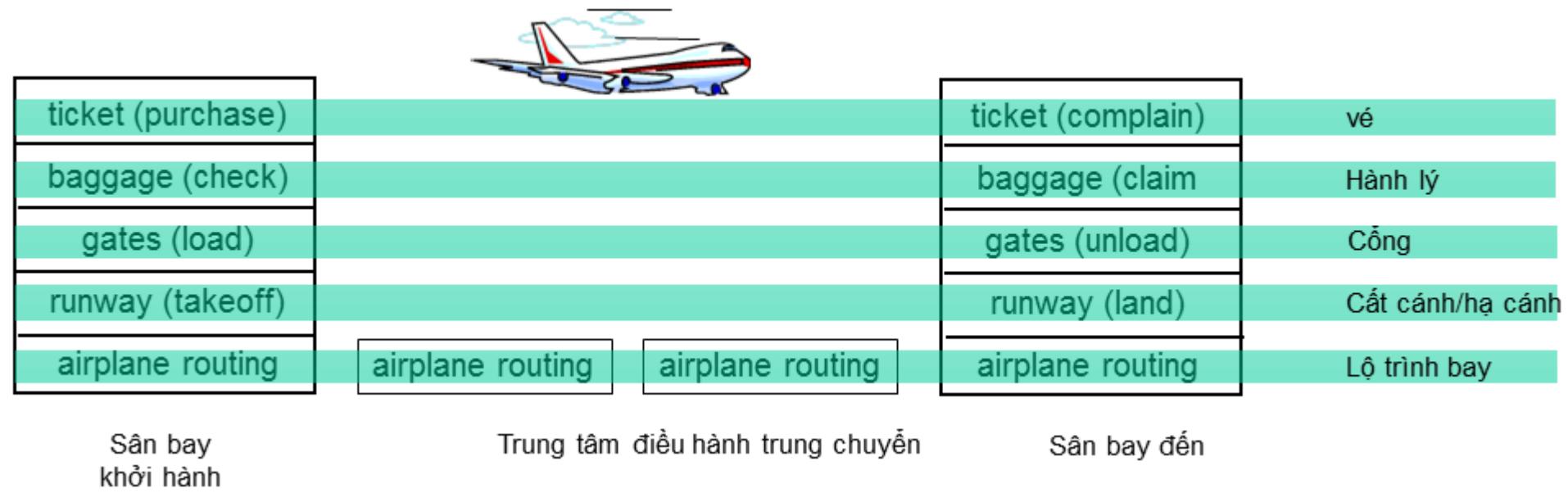
Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

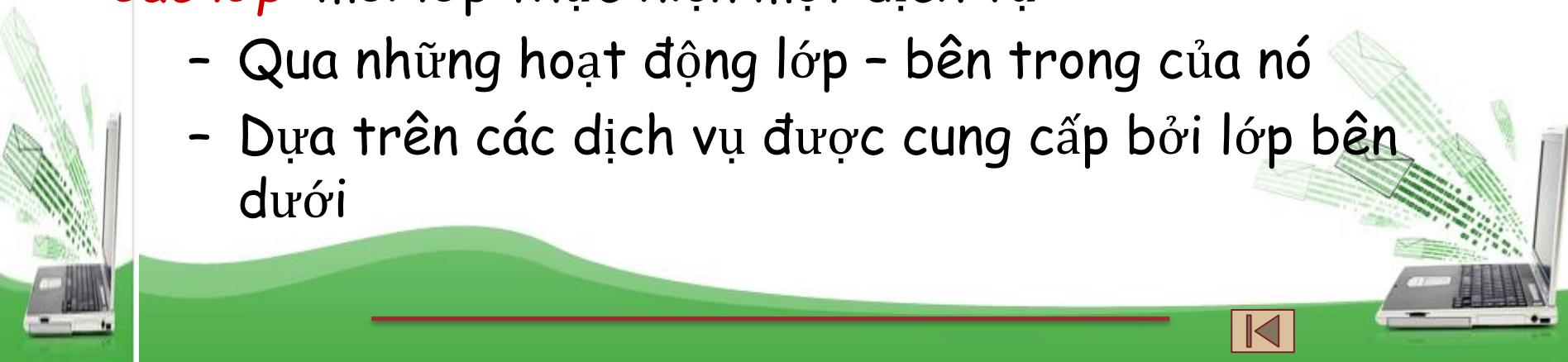


# Sự phân lớp của chức năng hàng không



**Các lớp:** mỗi lớp thực hiện một dịch vụ

- Qua những hoạt động lớp - bên trong của nó
  - Dựa trên các dịch vụ được cung cấp bởi lớp bên dưới



## Tại sao phải phân lớp?

Nhằm xử lý các hệ thống phức tạp

- Cấu trúc rõ ràng cho phép xác định quan hệ của các mảnh của hệ thống phức tạp
  - Thảo luận **mô hình tham chiếu** được phân
- Mô-dun hóa làm dễ dàng việc bảo trì và cập nhật hệ thống
  - Thay đổi việc thực hiện dịch vụ của các lớp là trong suốt với phần còn lại của hệ thống
  - Ví dụ: thay đổi thủ tục ở cổng sẽ không ảnh hưởng đến phần còn lại của hệ thống
- Xem xét những bất lợi của việc phân lớp

Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Mô hình tham chiếu OSI

- **Application:** hỗ trợ các ứng dụng mạng
  - FTP, SMTP, HTTP
- **Presentation:** cho phép các ứng dụng giải thích ý nghĩa của dữ liệu, ví dụ mã hóa, nén, những quy ước chuyên biệt
- **Session:** sự đồng bộ hóa, khả năng chịu lỗi, phục hồi sự trao đổi dữ liệu

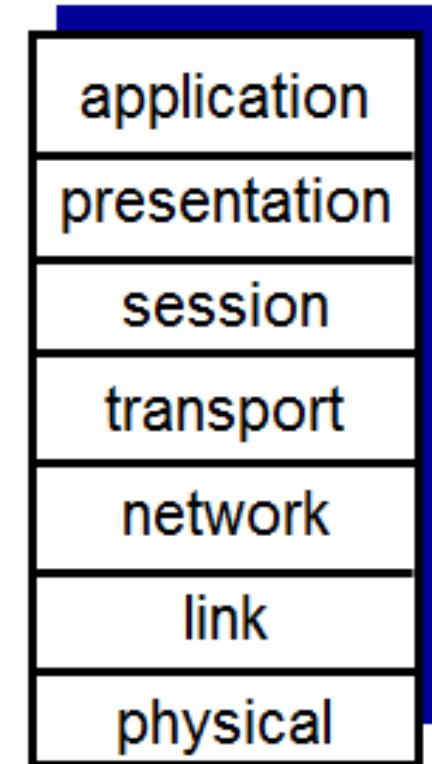
Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Chồng giao thức Internet (TCP/IP)

- **Application:** hỗ trợ các ứng dụng mạng
  - FTP, SMTP, HTTP
- **Transport:** chuyển dữ liệu từ tiến trình này đến tiến trình kia (process-process)
  - TCP, UDP
- **Network:** định tuyến những gói dữ liệu từ nguồn tới đích
  - IP, các giao thức định tuyến
- **Link:** chuyển dữ liệu giữa các thành phần mạng lân cận
  - Ethernet, 802.11 (WiFi), PPP
- **Physical:** bits “trên đường dây”

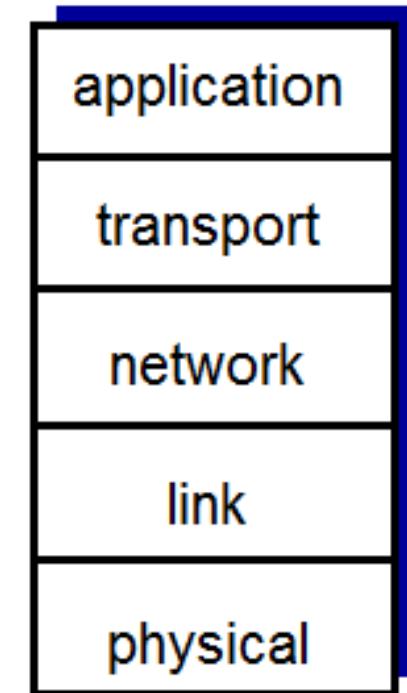
Internet là gì?

Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ



# Đóng gói (Encapsulation)

Internet là gì?

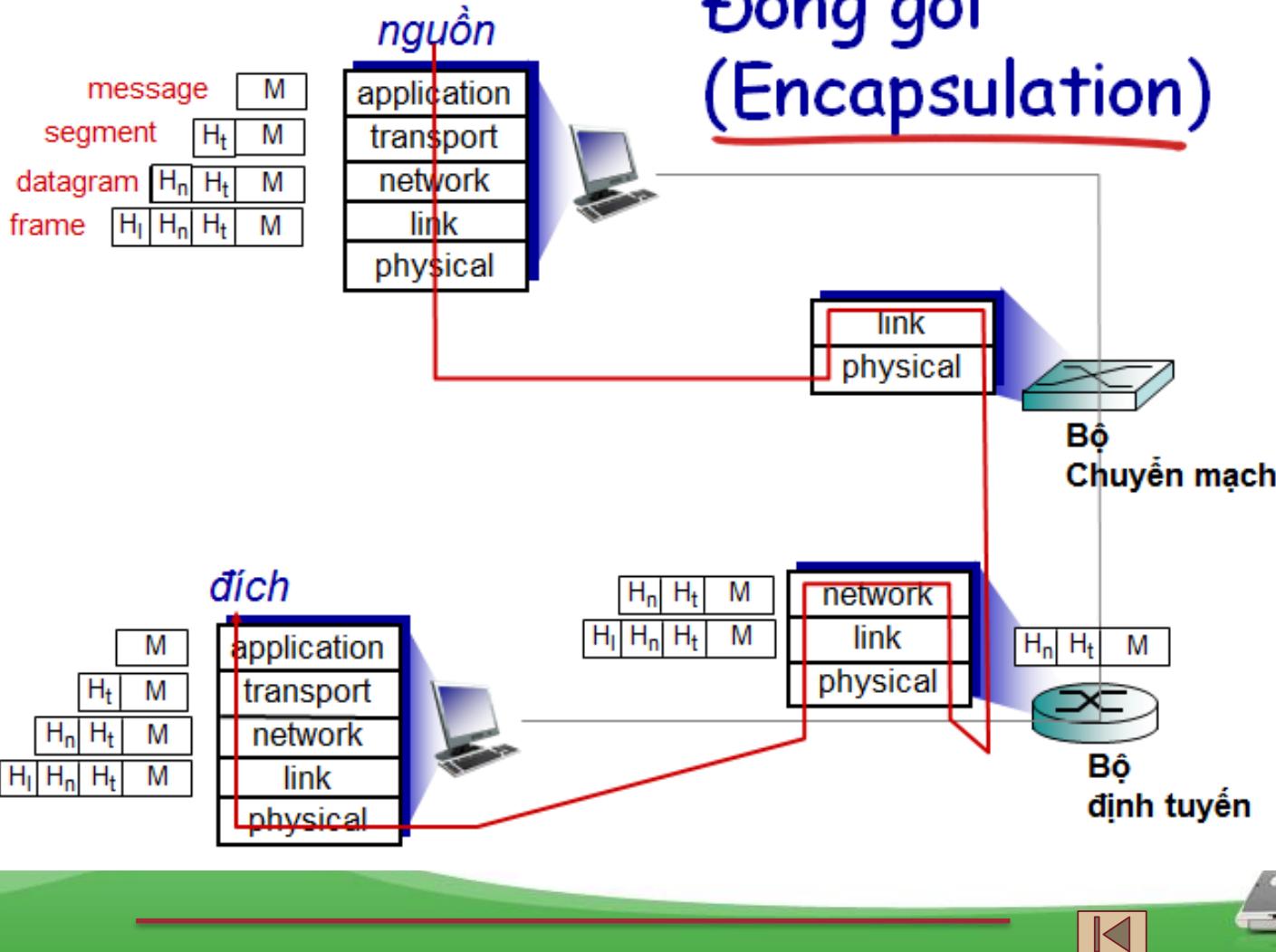
Mạng biên

Mạng lõi

Độ trễ, sự mất mát, thông lượng trong mạng

Các lớp giao thức, các mô hình dịch vụ

## Đóng gói (Encapsulation)



# Tham khảo

- Phần này chỉ những:
  - Sách/ chỉ mục bắt buộc học
  - Bài tập phải làm.
  - Sách/ chương đọc thêm.



# YÊU CẦU SỬ DỤNG FILE MẪU

- Slide số 2 (nội dung chính): khi click vào các nội dung chính 1,2,3... sẽ cho link tới các slide tương ứng.
- Tất cả các slide bên dưới có nút : khi click vào sẽ quay về slide số 2 (chứa nội dung chính)
- Phần nội dung chính của bài (để góc trên bên phải) khi soạn tới phần nào cho đổi màu nội dung phần đó. Nội dung này chính là slide số 2.
- Những slide có phần layout giống nhau nên sử dụng chức năng “Duplicate slide”.
- Phần hiệu ứng Thầy/Cô tùy chọn (nên đồng bộ cho tất cả slide)

**(Vui lòng xóa slide này khi đọc xong)**

# Mạng máy tính

ThS. Phạm Liệu  
*Email: lieu.pham@stu.edu.vn*

2018



# NỘI DUNG CHÍNH



Các nguyên lý của các ứng dụng mạng



Web và HTTP



FTP



Thư điện tử: SMTP, POP3, IMAP



DNS



# Tạo một ứng dụng mạng

**Viết chương trình:**

- Chạy trên các hệ thống đầu cuối (khác nhau)
- Liên lạc qua mạng
- Ví dụ: phần mềm web server giao tiếp với trình duyệt

**Không cần viết phần mềm cho các thiết bị trong lõi của mạng**

- Các thiết bị trong lõi mạng không chạy các ứng dụng của người dùng
- Các ứng dụng trên các hệ thống đầu cuối cho phép phát triển ứng dụng vi quảng bá nhanh chóng

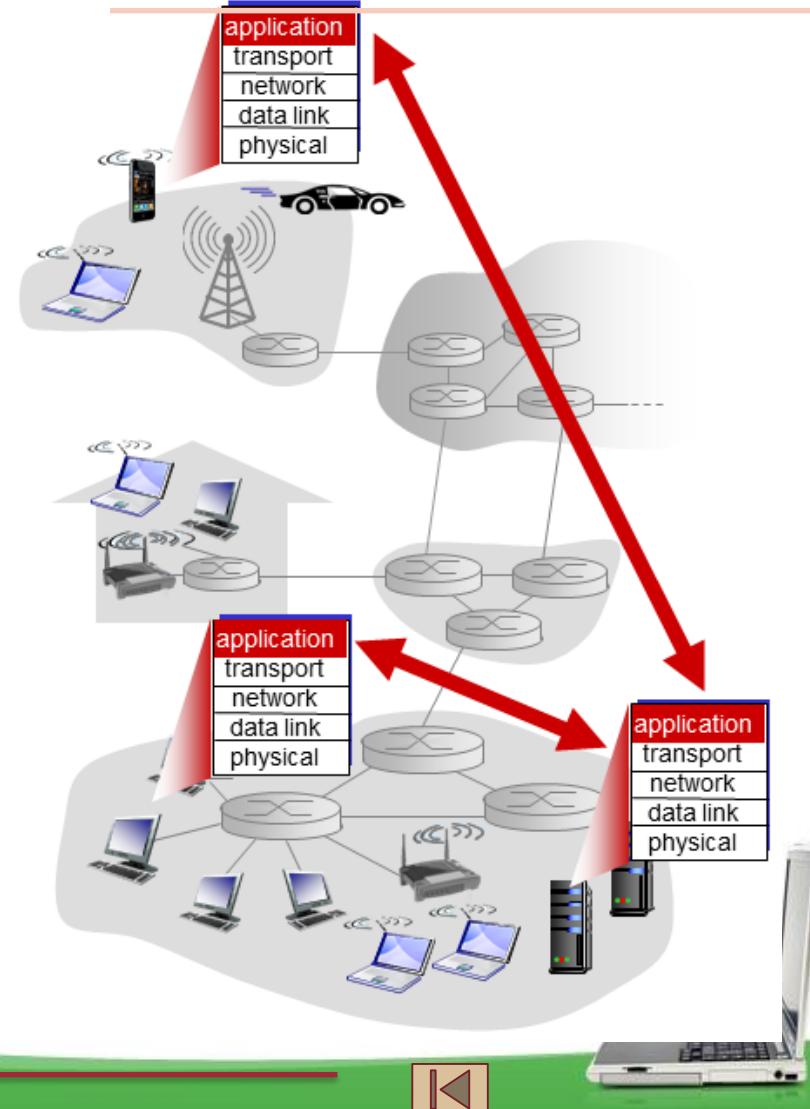
Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

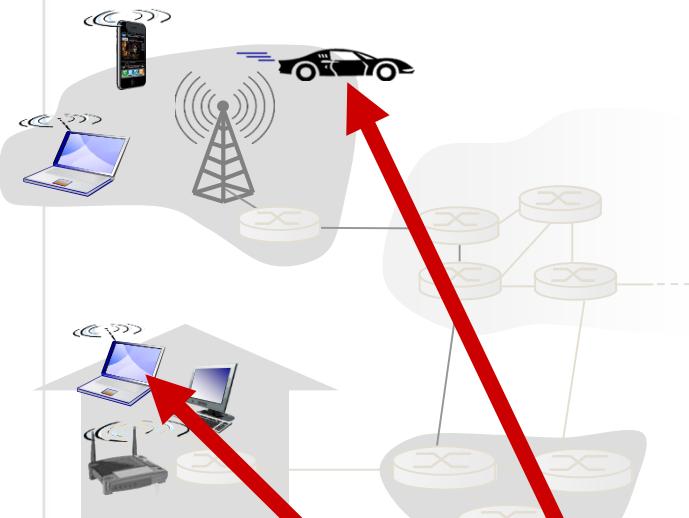


## Các kiến trúc ứng dụng

Kiến trúc phù hợp của các ứng dụng:

- Client-Server
- Peer-to-Peer (P2P) (Mạng ngang hàng)

# Kiến trúc Client-Server



## Client/Server



### Server:

- Máy luôn luôn hoạt động
- Địa chỉ IP cố định
- Tổ chức thành các trung tâm dữ liệu để mở rộng quy mô

### Clients:

- Giao tiếp với server
- Có thể kết nối không liên tục
- Có thể thay đổi địa chỉ IP
- Không giao tiếp trực tiếp với các client khác



## Kiến trúc P2P (ngang hàng)

- Không có server luôn luôn hoạt động
- Các hệ thống đầu cuối bất kỳ truyền thông trực tiếp với nhau
- Các peer yêu cầu dịch vụ từ các peer khác và cung cấp dịch vụ ngược lại cho các peer khác
  - Có khả năng tự mở rộng - các peer mới cung cấp thêm dịch vụ mới, cũng như có thêm nhu cầu mới về dịch vụ
- Các peer được kết nối không liên tục và có thể thay đổi địa chỉ IP
  - Quản lý phức tạp

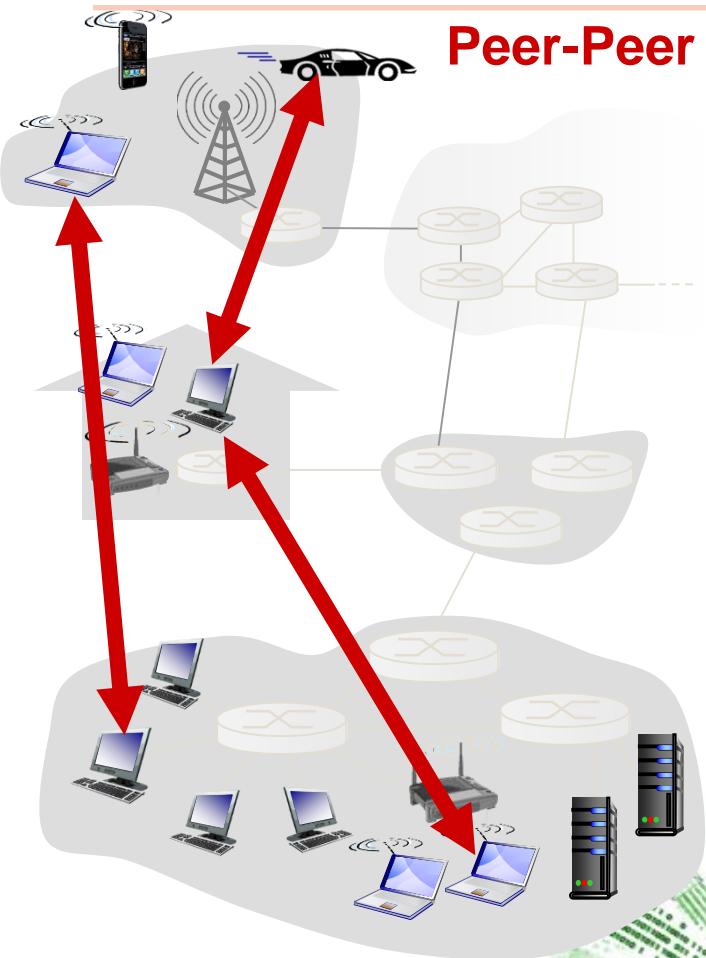
Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS



## Các tiến trình liên lạc

**Tiến trình (process):** chương trình đang chạy trong một máy

- Trong cùng một máy, hai tiến trình giao tiếp với nhau bằng cách sử dụng cơ chế truyền thông liên tiến trình (**inter-process communication**) (được định nghĩa bởi hệ điều hành)
- Các tiến trình trong các host khác nhau truyền thông với nhau bằng cách trao đổi **các thông điệp (message)**

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

## Clients, servers

**Tiến trình client:** tiến trình khởi tạo liên lạc

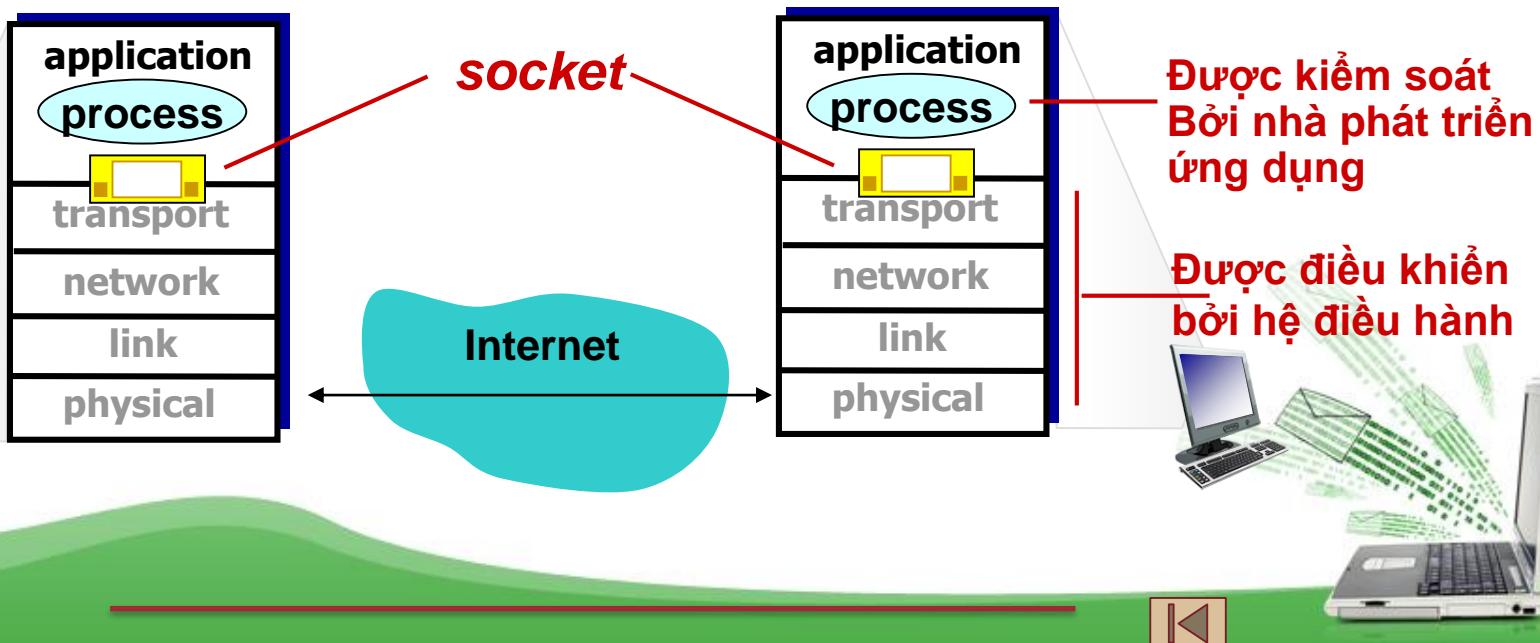
**Tiến trình server:** tiến trình chờ đợi để được liên lạc

- ❖ **Chú ý:** các ứng dụng với kiến trúc P2P có cả các tiến trình client và server



# Sockets

- Tiến trình gửi/nhận thông điệp đến/ từ **socket** của nó
- socket tương tự như cửa ra vào
  - Tiến trình gửi đẩy thông điệp ra khỏi cửa
  - Tiến trình gửi dựa trên hạ tầng vận chuyển bên kia của cánh cửa để phân phối thông điệp đến socket tại tiến trình nhận



## Xác định tiến trình

- Để nhận thông điệp, tiến trình phải có **định danh**
- Thiết bị host device có địa chỉ IP 32-bit duy nhất
- Q:** địa chỉ IP của host mà trên tiến trình đang chạy trên đó có đủ để xác định tiến trình đó hay không?
  - A:** không, có nhiều tiến trình có thể đang được chạy trên cùng một host

- Định danh (identifier)** bao gồm cả địa chỉ IP và **số cổng (port numbers)** được liên kết với tiến trình trên host.
- Ví dụ về số cổng:
  - HTTP server: 80
  - Mail server: 25
- Để gửi thông điệp HTTP đến web server gaia.cs.umass.edu :
  - **IP address:** 128.119.245.12
  - **port number:** 80
- Còn nữa...

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS



# Giao thức tầng ứng dụng định nghĩa

- Các loại thông điệp được trao đổi
  - e.g., yêu cầu (request), đáp ứng (response)
- Cú pháp thông điệp:
  - Các trường trong thông điệp và cách mà các trường được định nghĩa
- Ý nghĩa của thông điệp
  - Ý nghĩa của thông tin trong các trường
- Các quy tắc (rules) khi nào và cách mà các tiến trình gửi và đáp ứng các thông điệp

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

## Các giao thức mở:

- Được định nghĩa trong RFCs
- Cung cấp khả năng tương tác cho các ứng dụng thuộc các nhà phát triển khác nhau
- Vd: HTTP, SMTP

## Các giao thức độc quyền:

- Vd: Skype



# Dịch vụ vận chuyển nào mà ứng dụng cần?

## Toàn vẹn dữ liệu (data integrity)

- Một số ứng dụng (ví dụ truyền file, web transactions) yêu cầu độ tin cậy 100% khi truyền dữ liệu
- Các ứng dụng khác (ví dụ audio) có thể chịu được một số mất mát.

## Định thi (timing)

- Một số ứng dụng (ví dụ, thoại Internet, game tương tác) yêu cầu độ trễ thấp để đạt được "hiệu quả" thực thi

## Thông lượng (throughput)

- ❖ Một số ứng dụng (vd: đa phương tiện) yêu cầu thông lượng tối thiểu để đạt được "hiệu quả" thực thi
- ❖ Các ứng dụng khác ("ứng dụng mềm dẻo") có thể dùng bất kỳ thông lượng nào cũng được

## An ninh

- ❖ Mã hóa, toàn vẹn dữ liệu, ...



# Các yêu cầu dịch vụ vận chuyển: các ứng dụng phổ biến

<u>ứng dụng</u>	<u>mất dữ liệu</u>	<u>thông lượng</u>	<u>độ nhạy thời gian</u>
Truyền file	không	mềm dẻo	không
e-mail	không	mềm dẻo	không
Web documents	không	mềm dẻo	không
audio/video	chịu lỗi	audio: 5kbps-1Mbps	có, 100' s msec
thời gian thực		video:10kbps-5Mbps	
audio/video đã lưu	chịu lỗi	như trên	có, vài giây
Game tương tác	chịu lỗi		có, 100' s msec
nhắn tin	không		có và không
		Trên một vài kbps	
		mềm dẻo	



# Các dịch vụ thuộc giao thức vận chuyển trên Internet

## Dịch vụ TCP:

- Truyền tải có đảm bảo (reliable transport) giữa tiến trình gửi và nhận
- Điều khiển luồng thông tin (flow control): bên gửi sẽ không gửi vượt khả năng bên nhận
- Điều khiển tắc nghẽn (congestion control): điều tiết bên gửi khi mạng quá tải
- Không hỗ trợ: định thì, bảo đảm thông lượng tối thiểu, bảo mật
- Hướng kết nối (connection-oriented): yêu cầu thiết lập kết nối giữa tiến trình client và server trước khi truyền

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

## Dịch vụ UDP:

- Truyền dữ liệu không đảm bảo (unreliable data transfer) giữa tiến trình gửi và nhận
- Không hỗ trợ: độ tin cậy, điều khiển luồng, điều khiển tắc nghẽn, định thì, bảo đảm thông lượng, bảo mật, và thiết lập kết nối.

**Q:** Tại sao phải quan tâm?  
Tại sao có UDP?



# Ứng dụng Internet: Các giao thức tầng application, transport

Các nguyên lý của các ứng dụng mạng

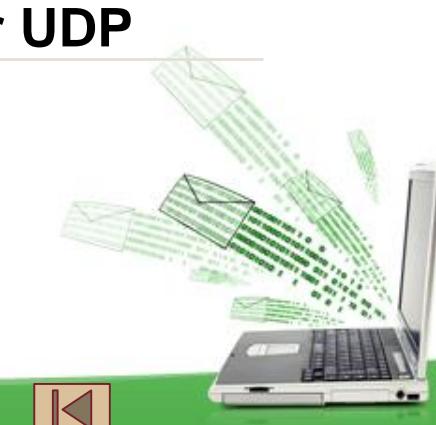
Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

	Giao thức tầng application	Giao thức dưới tầng transport
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Truyền file	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g., YouTube), RTP [RFC 1889]	TCP or UDP
Thoại Internet	SIP, RTP, độc quyền (e.g., Skype)	TCP or UDP



# Bảo mật TCP

## TCP & UDP

- Không mã hóa
- Mật mã *chưa* mã hóa được gởi đến socket để đi qua Internet trong dạng nguyên bản

## SSL

- Hỗ trợ kết nối TCP được mã hóa
- Toàn vẹn dữ liệu
- Chứng thực đầu cuối

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

## SSL là giao thức ở tầng Application

- Các ứng dụng dùng thư viện SSL, thông qua đó để "nói chuyện" với TCP

## SSL socket API

- ❖ Mật mã dạng không mã hóa được gởi vào trong socket và chuyển đi qua Internet ở dạng mã hóa
- ❖ Xem chương 7



## Web và HTTP

Ôn lại...

- *web page* bao gồm các đối tượng (*objects*)
- Đối tượng có thể là tập tin HTML, hình ảnh JPEG, Java applet, tập tin audio,...
- web page bao gồm tập tin HTML bao gồm một số đối tượng được tham chiếu
- Mỗi đối tượng có thể được xác định bởi một *URL*, ví dụ

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

tên máy

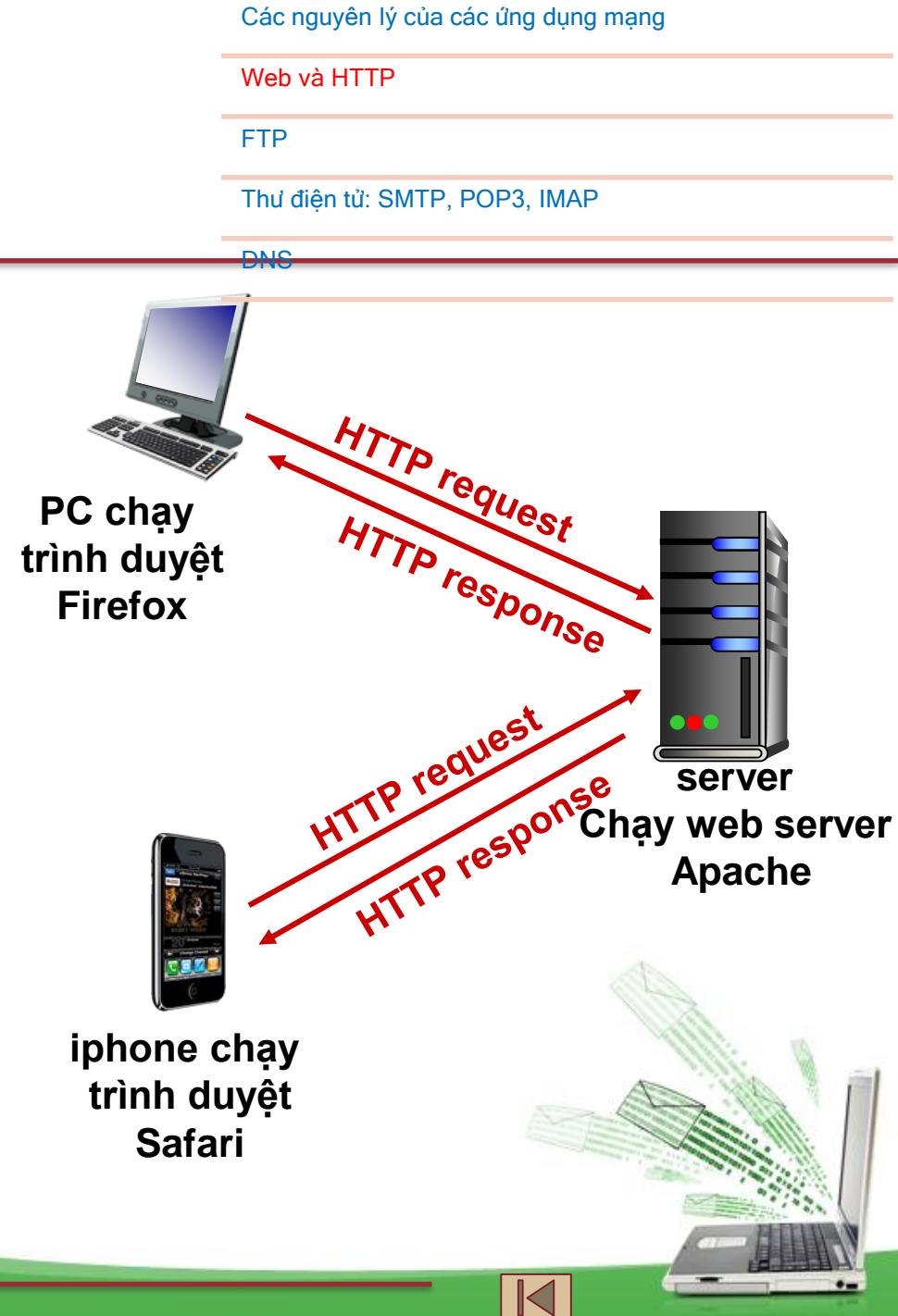
đường dẫn



# Tổng quan HTTP

## HTTP: hypertext transfer protocol

- Giao thức tầng Application của Web
- Mô hình client/server
  - **client:** trình duyệt gửi yêu cầu, nhận phản hồi (dùng giao thức HTTP) và "hiển thị" các đối tượng Web
  - **server:** Web server gửi (dùng giao thức HTTP) các đối tượng để trả lời yêu cầu



# Tổng quan HTTP (tt)

dùng TCP:

- Client khởi tạo kết nối TCP (tạo socket) đến cổng 80 server
- server chấp nhận kết nối TCP từ client
- Các thông điệp HTTP (thông điệp thuộc giao thức tầng application) được trao đổi giữa trình duyệt (HTTP client) và web server (HTTP server)
- Kết nối TCP được đóng

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

**HTTP “không lưu trạng thái”**

- server không duy trì thông tin về các yêu cầu trước đó của client

**ngoài lề**  
Các giao thức nào duy trì “trạng thái” thì phức tạp!

- ❖ Lịch sử trước đó (trạng thái) phải được duy trì
- ❖ Nếu server/client bị sự cố, cách nhìn về “trạng thái” của nó có thể bị mâu thuẫn, phải được điều chỉnh



## Các kết nối HTTP

### HTTP không bền vững

- Chỉ tối đa một đối tượng được gửi qua kết nối TCP
  - Kết nối sau đó sẽ bị đóng
- Tải nhiều đối tượng yêu cầu nhiều kết nối

### HTTP bền vững

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

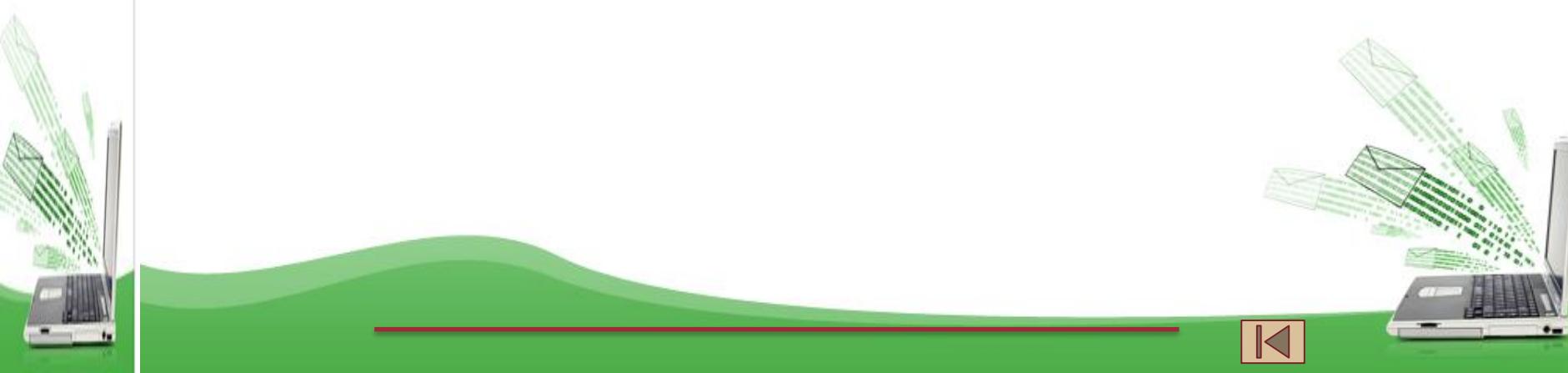
Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS



# HTTP không bền vững

Giả sử người dùng vào URL như sau:

`www.someSchool.edu/someDepartment/home.index` tham chiếu đến 10 hình jpeg)

1a. HTTP client khởi tạo kết nối

TCP đến HTTP server (tiến trình) tại `www.someSchool.edu` trên port 80

1b. HTTP server tại host `www.someSchool.edu` chờ kết nối TCP tại port 80. “chấp nhận” kết nối, thông báo cho client

2. HTTP client gửi **thông điệp yêu cầu** HTTP (chứa URL)

vào trong socket kết nối TCP. Thông điệp chỉ ra rằng client muốn đối tượng `someDepartment/home.index`

3. HTTP server nhận thông điệp yêu cầu, tạo **thông điệp phản hồi** chứa đối tượng được yêu cầu, và gửi thông điệp đến socket của nó



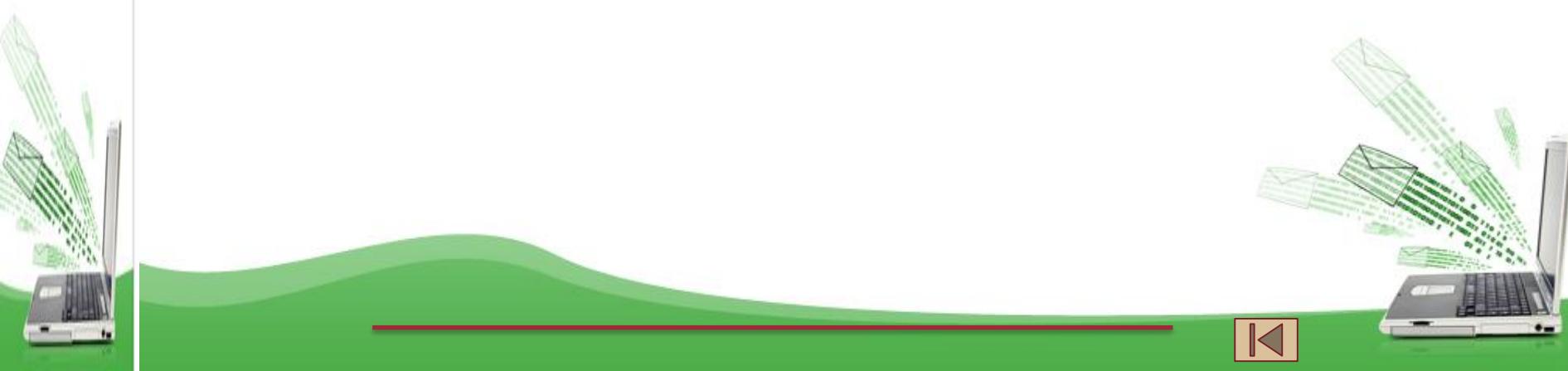
## HTTP không bền vững(tt)



### 4. HTTP server đóng kết nối TCP.

5. HTTP client nhận thông điệp phản hồi chứa file html, hiển thị html. Phân tích cú pháp file html, tìm ra các đối tượng jpeg được tham chiếu
6. Các bước 1-5 được lặp lại cho mỗi đối tượng trong 10 đối tượng jpeg

time  
↓



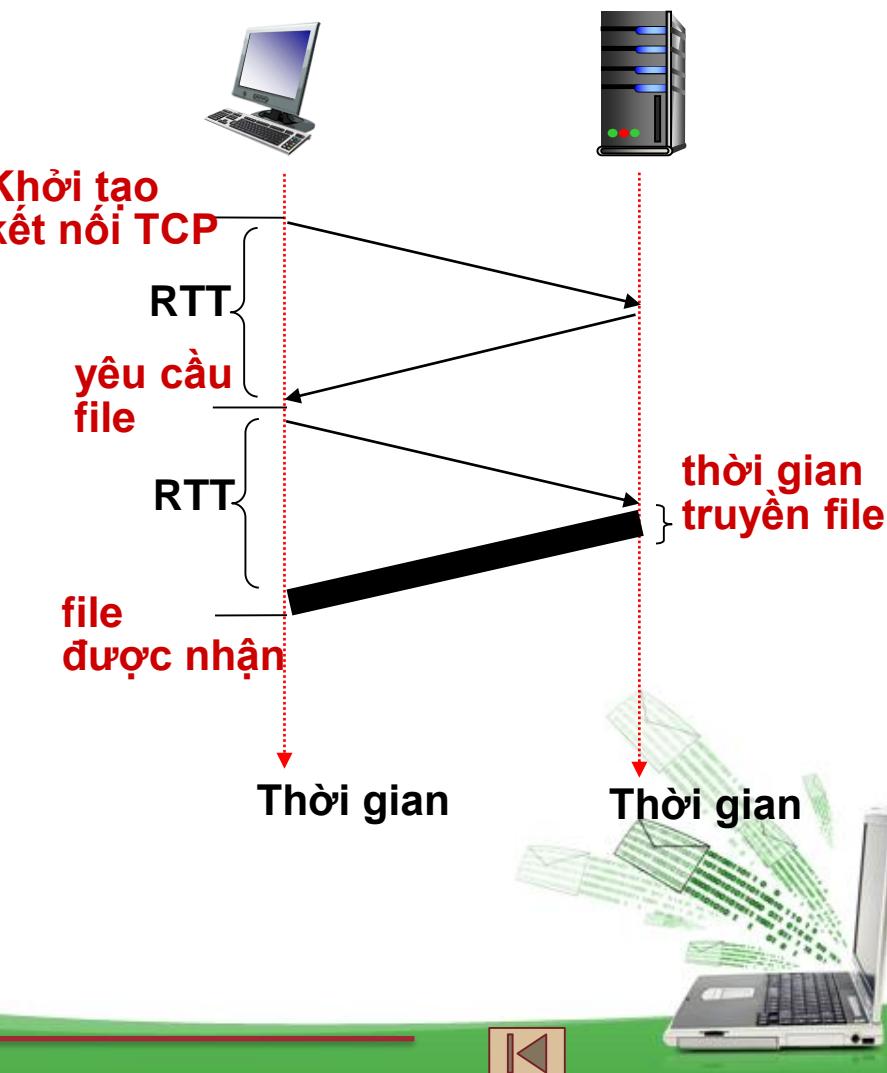
# HTTP không bền vững: thời gian đáp ứng

**RTT (round-trip time):** thời gian để cho một gói tin nhỏ đi từ client đến server và quay ngược lại

**Thời gian đáp ứng HTTP:**

- Một RTT để khởi tạo kết nối TCP
- Một RTT cho yêu cầu HTTP và vài byte đầu tiên của phản hồi HTTP được trả về
- Thời gian truyền file
- Thời gian đáp ứng HTTP không bền vững =

2RTT + thời gian truyền file



# HTTP bền vững

Vấn đề với HTTP không bền vững:

- Yêu cầu requires 2 RTTs cho mỗi đối tượng
- Tốn tài nguyên khi Hệ điều hành xử lý mỗi kết nối TCP
- Các trình duyệt thường mở các kết nối TCP song song để lấy các đối tượng được tham chiếu

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

**HTTP bền vững:**

- Server để kết nối mở sau khi gửi phản hồi
- Các thông điệp HTTP tiếp theo giữa cùng client/server được gửi trên kết nối đã mở ở trên
- Client gửi các yêu cầu ngay khi nó gặp một đối tượng tham chiếu
- Chỉ cần một RTT cho tất cả các đối tượng được tham chiếu



## Thông điệp yêu cầu HTTP

- hai loại thông điệp HTTP: yêu cầu (*request*), phản hồi (*response*)
- Thông điệp yêu cầu HTTP:
  - ASCII (dạng thức con người có thể đọc được)

Dòng yêu cầu

(các lệnh GET, POST, HEAD)

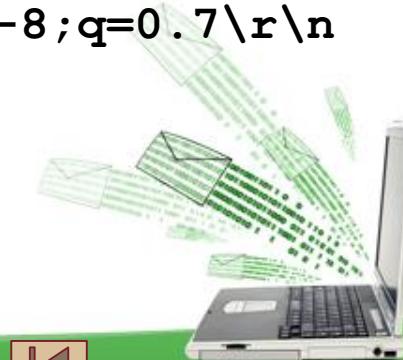
Các dòng header

ký tự xuống dòng,  
về đầu dòng mới chỉ  
điểm cuối cùng  
của thông điệp

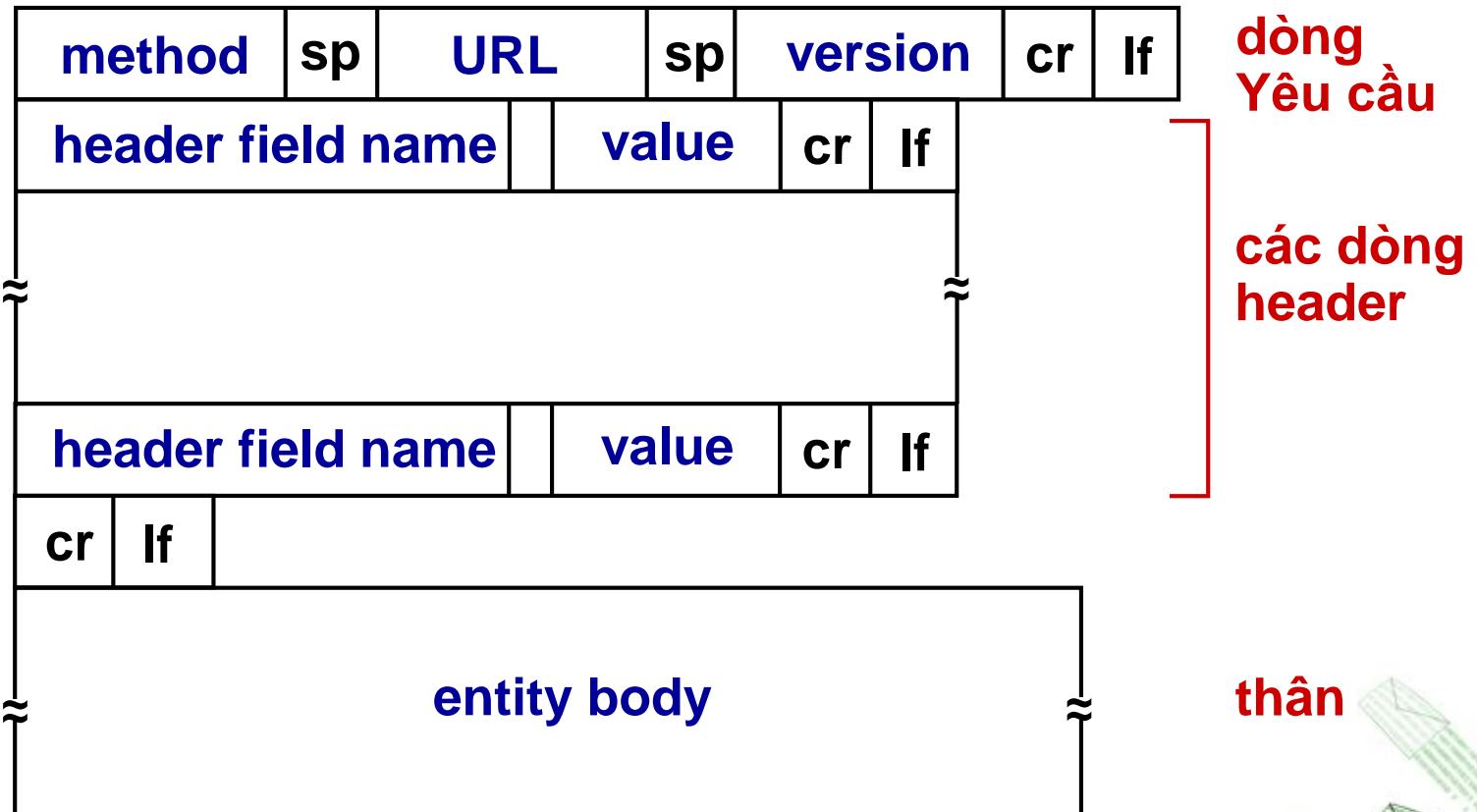
```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

Ký tự về đầu dòng

Ký tự xuống dòng



# Thông điệp yêu cầu HTTP: định dạng tổng quát



## Tải lên biểu mẫu nhập liệu

### Phương thức POST:

- Web page thường bao gồm form input
- dữ liệu nhập được tải lên server trong phần thân đối tượng HTML

### Phương thức URL:

- Dùng phương thức GET
- dữ liệu nhập được tải lên trong trường URL của dòng yêu cầu:

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



# Các phương thức

## HTTP/1.0:

- GET
- POST
- HEAD
  - Yêu cầu server loại bỏ đối tượng được yêu cầu ra khỏi thông điệp phản hồi

## HTTP/1.1:

- GET, POST, HEAD
- PUT
  - Tải file trong thân thực thể đến đường dẫn được xác định trong trường URL
- DELETE
  - Xóa file được chỉ định trong trường URL

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS



# Thông điệp phản hồi HTTP

dòng trạng thái

(giao thức)

mã trạng thái

cụm từ trạng thái)

các dòng header

```
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
```

Dữ liệu, ví dụ,  
file HTML  
được yêu cầu

data data data data data ...



## Các mã trạng thái phản hồi HTTP

- ❖ Mã trạng thái xuất hiện trong dòng đầu tiên trong thông điệp đáp ứng từ server tới client
- ❖ Một số mã mẫu:

### 200 OK

- Yêu cầu thành công, đối tượng được yêu cầu sau ở trong thông điệp này

### 301 Moved Permanently

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

### 400 Bad Request

- Server không hiểu thông điệp yêu cầu

### 404 Not Found

- Thông tin được yêu cầu không tìm thấy trên server này

### 505 HTTP Version Not Supported



## Thử kiểm tra HTTP (phía client)

### 1. Telnet đến Web server yêu thích của bạn:

telnet cis.poly.edu 80

Mở kết nối TCP ở port 80 (port server HTTP mặc định) tại cis.poly.edu. Mọi thứ nhập vào được gửi đến port 80 tại cis.poly.edu

### 2. Nhập vào yêu cầu trong lệnh GET HTTP:

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

Bằng cách gõ những dòng này (enter 2 lần), bạn đã gửi yêu cầu GET tối thiểu (nhưng đầy đủ) đến HTTP server

### 3. Xem thông điệp phản hồi được gửi bởi HTTP server!

(hoặc dùng Wireshark để xem thông điệp

yêu cầu và phản hồi của HTTP được bắt lại)



## Trạng thái User-server: cookies

Nhiều Web site dùng cookies

4 thành phần:

1) cookie header line của thông điệp phản hồi HTTP

2) cookie header line trong thông điệp yêu cầu HTTP kế tiếp

3) File cookie được lưu trữ trên máy người dùng, được quản lý bởi trình duyệt của người dùng

4) Cơ sở dữ liệu tại Web site

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

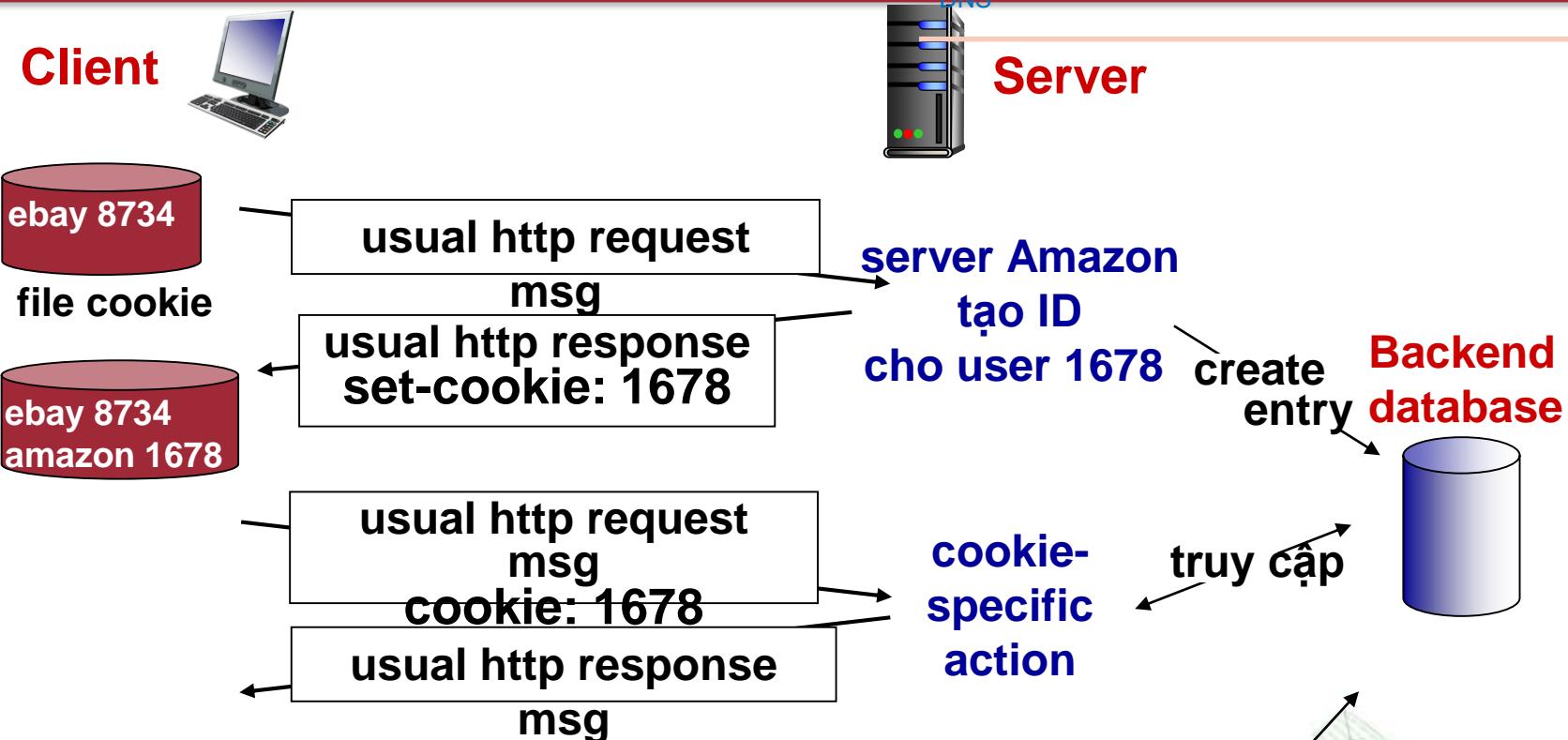
DNS

Ví dụ:

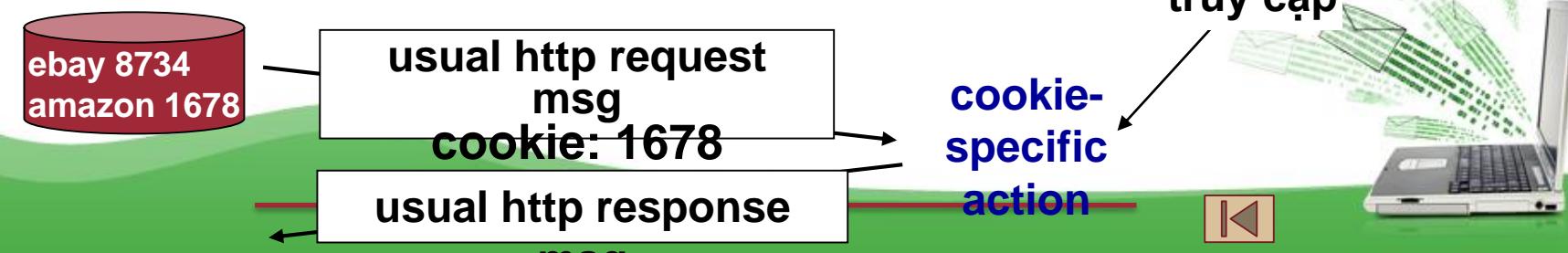
- Susan thường truy cập Internet từ một PC
- Vào trang thương mại điện tử lần đầu tiên
- Khi yêu cầu khởi tạo HTTP đến trang web đó, thì trang đó tạo:
  - ID duy nhất
  - Một bản ghi trong cơ sở dữ liệu cho ID đó



## Cookies: lưu trữ “trạng thái” (tt.)



một tuần sau:



## Cookies (tt)

*Cookie có thể được sử dụng cho:*

- Cấp phép
- Giỏ mua hàng
- Các khuyến cáo
- Trạng thái phiên làm việc của user (Web e-mail)

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

*ngoài ra cookies và sự riêng tư:*

- ❖ cookie cho phép các trang biết nhiều hơn về bạn
- ❖ Bạn có thể cung cấp tên và email cho các trang

*Làm thế nào để giữ “trạng thái”:*

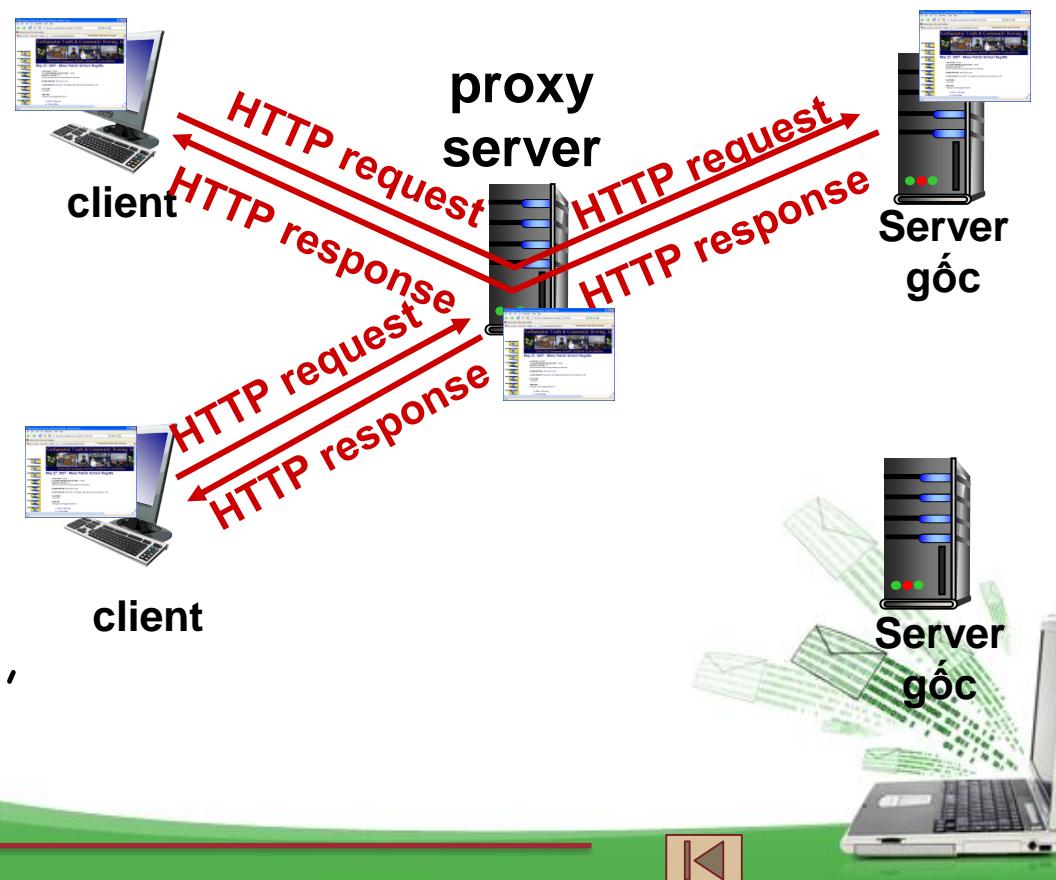
- ❖ các thời điểm kết thúc giao thức: duy trì trạng thái tại người gửi/nhận thông qua nhiều giao dịch
- ❖ cookies: các thông điệp http mang trạng thái



## Web caches (proxy server)

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

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



## Thông tin thêm về Web caching

- Cache hoạt động như là cả client và server
  - server đối với client yêu cầu thông tin
  - client đối với server cung cấp
- Thông thường cache được cài đặt bởi ISP (trường đại học, công ty, ISP riêng)

### Tại sao dùng Web caching?

- Giảm thời gian đáp ứng cho yêu cầu của client
- Giảm lưu lượng trên đường link truy cập ra Internet của một tổ chức
- Internet có rất nhiều caches: cho phép những nhà cung cấp nội dung với lượng tài nguyên "nghèo nàn" vẫn cung cấp nội dung một cách hiệu quả (chia sẻ file P2P cũng vậy)



## Ví dụ Caching:

*Giả sử:*

- ❖ Kích thước trung bình của đối tượng: 100K bits
- ❖ Số lượng yêu cầu trung bình từ trình duyệt đến server gốc: 15/sec
- ❖ Tốc độ truyền dữ liệu trung bình đến trình duyệt: 1.50 Mbps
- ❖ RTT từ bộ định tuyến của tổ chức đến bất kỳ server gốc: 2 giây
- ❖ Tốc độ link truy cập: 1.54 Mbps

*Vấn đề!*

*Kết quả:*

- ❖ Độ khả dụng của LAN: 15%
- ❖ Độ khả dụng của link truy cập = **99%**
- ❖ Tổng thời gian trễ = trễ Internet + trễ truy cập + trễ LAN  
 $= 2 \text{ giây} + \text{minutes} + \mu\text{secs}$

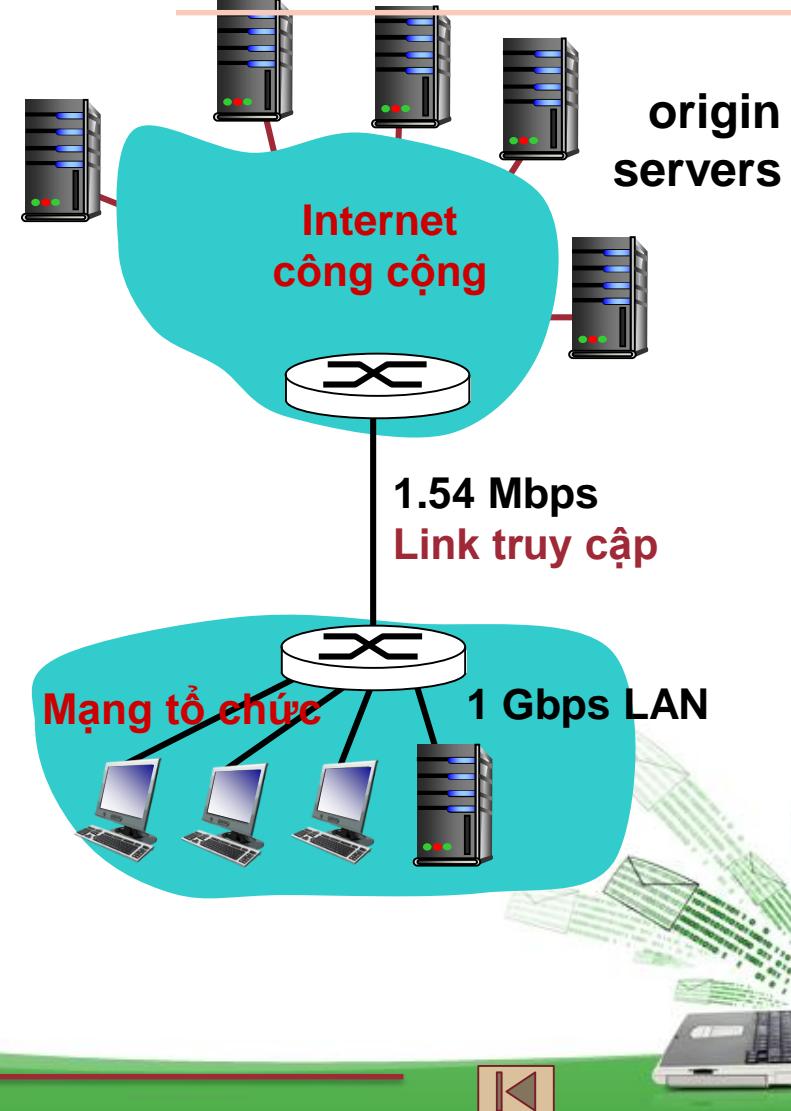
Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

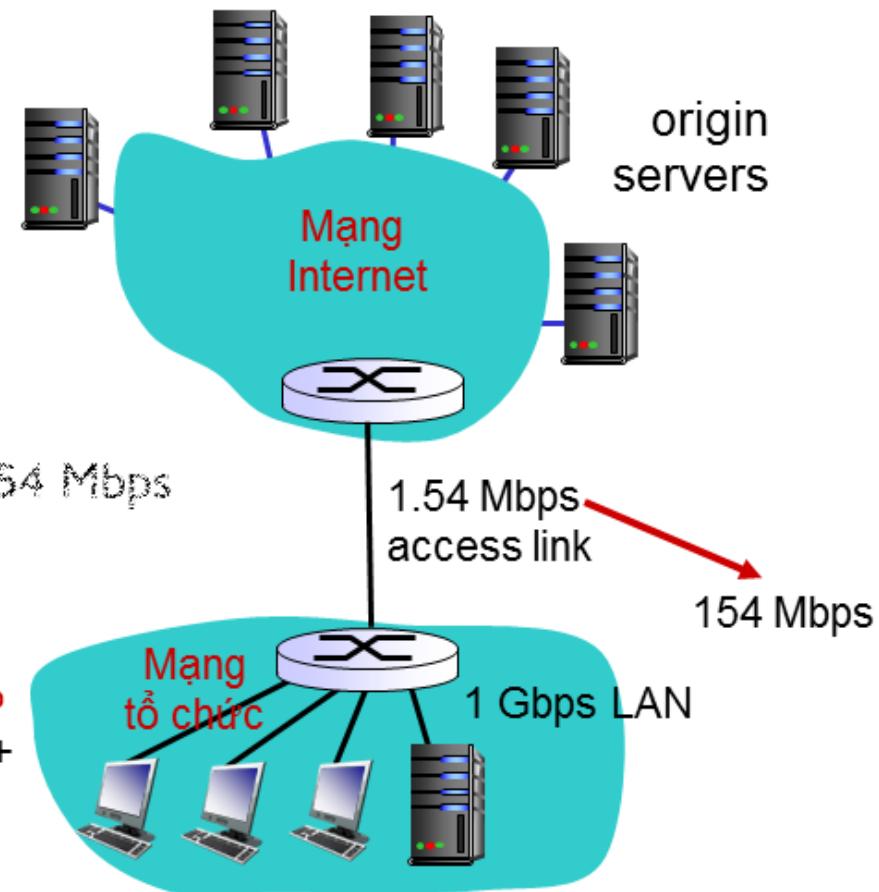
DNS



## Ví dụ Caching: đường link truy cập lớn hơn

*Giả sử:*

- ❖ Kích thước trung bình của đối tượng: 100K bits
- ❖ tốc độ trung bình yêu cầu từ trình duyệt đến server = 15/s
- ❖ Tốc độ trung bình dữ liệu đến trình duyệt: 1.50 Mbps
- ❖ RTT từ bộ định tuyến của tổ chức đến bất kỳ server gốc: 2 giây
- ❖ Tốc độ link truy cập: 1.54 Mbps



*Kết quả:*

- ❖ độ khả dụng của LAN = 15%
- ❖ độ khả dụng trên liên kết truy cập= 99%
- ❖ Tổng độ trễ= trễ Internet + trễ truy cập + trễ LAN  
= 2 sec + minutes + µsecs



## Ví dụ Caching: thiết lập cache

### Giả sử:

- ❖ Kích thước trung bình của đối tượng: 100K bits
- ❖ tốc độ trung bình yêu cầu từ trình duyệt đến server = 15/s
- ❖ Tốc độ trung bình dữ liệu đến trình duyệt: 1.50 Mbps
- ❖ RTT từ bộ định tuyến của tổ chức đến bất kỳ server gốc: 2 giây
- ❖ Tốc độ link truy cập: 1.54 Mbps

### Kết quả:

- ❖ Độ khả dụng LAN: 15%
- ❖ access link utilization =
- ❖ total delay =

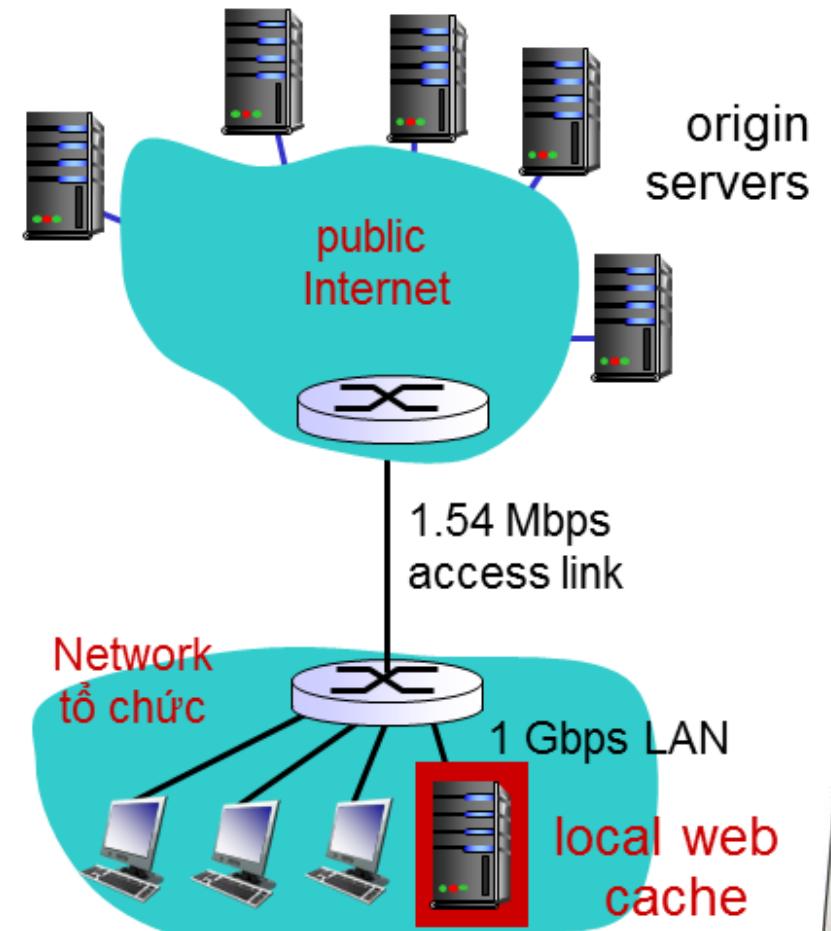
Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

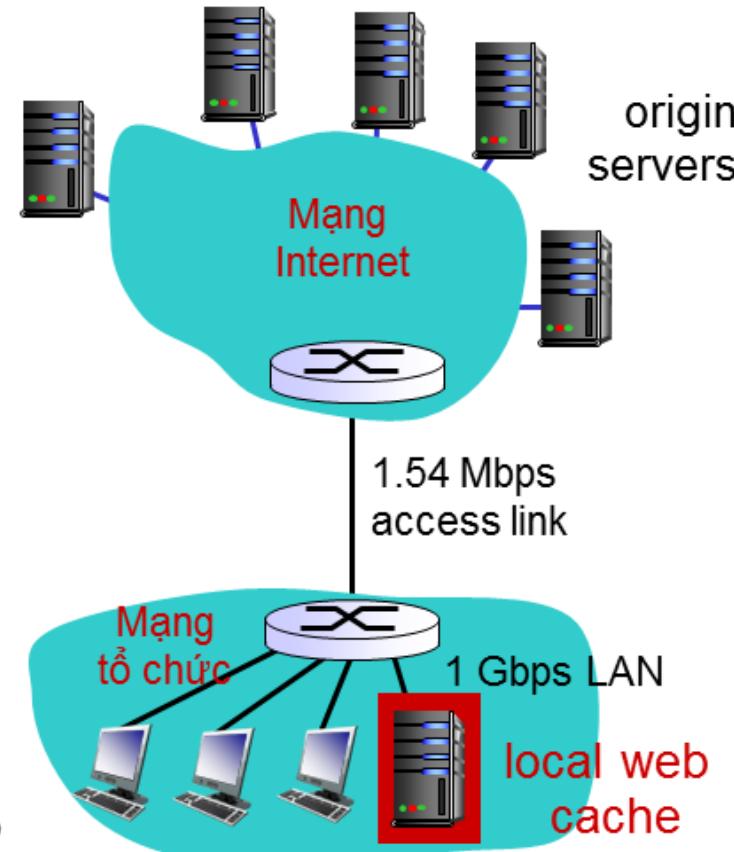
DNS



## Ví dụ Caching: thiết lập cache

### Tính độ khả dụng của đường link truy cập, độ trễ với cache:

- ❖ Giả sử khả năng đáp ứng của cache là 0.4
  - 40% yêu cầu được cache đáp ứng, 60% yêu cầu được server gốc đáp ứng
- ❖ Độ hiệu dụng của đường kết nối ra ngoài (access link):
  - 60% yêu cầu dùng access link
- ❖ Tốc độ truyền dữ liệu đến trình duyệt trên access link =  $0.6 * 1.50 \text{ Mbps} = 0.9 \text{ Mbps}$ 
  - Độ khả dụng =  $0.9 / 1.54 = 0.58$
- ❖ Tổng độ trễ
  - =  $0.6 * (\text{độ trễ từ server gốc}) + 0.4 * (\text{độ trễ khi được cache đáp ứng})$
  - =  $0.6 * (2.01) + 0.4 * (\sim \text{msecs})$
  - =  $\sim 1.2 \text{ secs}$
  - Ít hơn với link 154 Mbps (và cũng rẻ hơn!)



## GET có điều kiện

- Mục tiêu:** không gửi đối tượng nếu đối tượng trong cache đã được cập nhật
  - Không có độ trễ truyền dữ liệu
  - Mức độ sử dụng đường link thấp hơn
- cache:** xác định thời gian của bản sao được cache trong thông điệp yêu cầu HTTP  
**If-modified-since: <date>**
- server:** đáp ứng không chứa đối tượng nếu bản sao trong cache đã được cập nhật:  
**HTTP/1.0 304 Not Modified**

Client



Server



Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

**HTTP request msg**  
**If-modified-since: <date>**

**HTTP response**  
**HTTP/1.0**  
**304 Not Modified**

Đối tượng  
không được  
thay đổi  
trước  
<ngày>

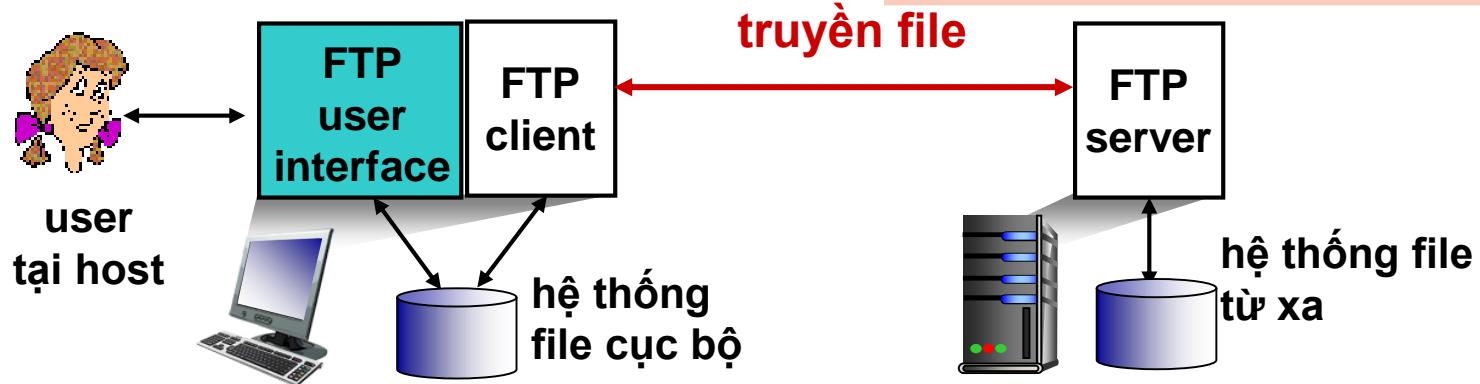
**HTTP request msg**  
**If-modified-since: <date>**

**HTTP response**  
**HTTP/1.0 200 OK**  
**<data>**

Đối tượng  
được thay  
đổi sau  
<ngày>



# FTP: giao thức truyền file



- ❖ Truyền file đến/từ máy ở xa
- ❖ Mô hình client/server
  - *client*: phía khởi tạo phiên truyền (đến/từ máy ở xa)
  - *server*: máy ở xa
- ❖ FTP: RFC 959
- ❖ FTP server: port 21



## FTP:

# kết nối điều khiển và kết nối dữ liệu riêng biệt

- FTP client liên hệ với FTP server tại port 21, dùng TCP
- client được cấp phép trên kết nối điều khiển
- client duyệt thư mục từ xa, bằng cách gửi các lệnh trên kết nối điều khiển
- Khi server nhận lệnh truyền file, **server** mở kết nối dữ liệu TCP thứ 2 (để truyền file) đến client
- Sau khi truyền một file, server đóng kết nối dữ liệu

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS



- ❖ **server** mở kết nối dữ liệu TCP khác để truyền file khác
- ❖ Kết nối điều khiển: “*out of band*” (*ngoại tuyến*)
- ❖ FTP server duy trì “trạng thái”: thư mục hiện tại, xác thực trước đó



# Các lệnh và phản hồi FTP

## Các lệnh mấu:

- Gởi văn bản ASCII trên kênh điều khiển
- USER *username*
- PASS *password*
- LIST trả về danh sách file trên thư mục hiện tại
- RETR *filename* lấy file
- STOR *filename* lưu trữ (đặt) file vào trong máy ở xa

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

## Ví dụ mã trả về

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



## Thư điện tử

# Thư điện tử

### Ba thành phần chính:

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

### User Agent

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

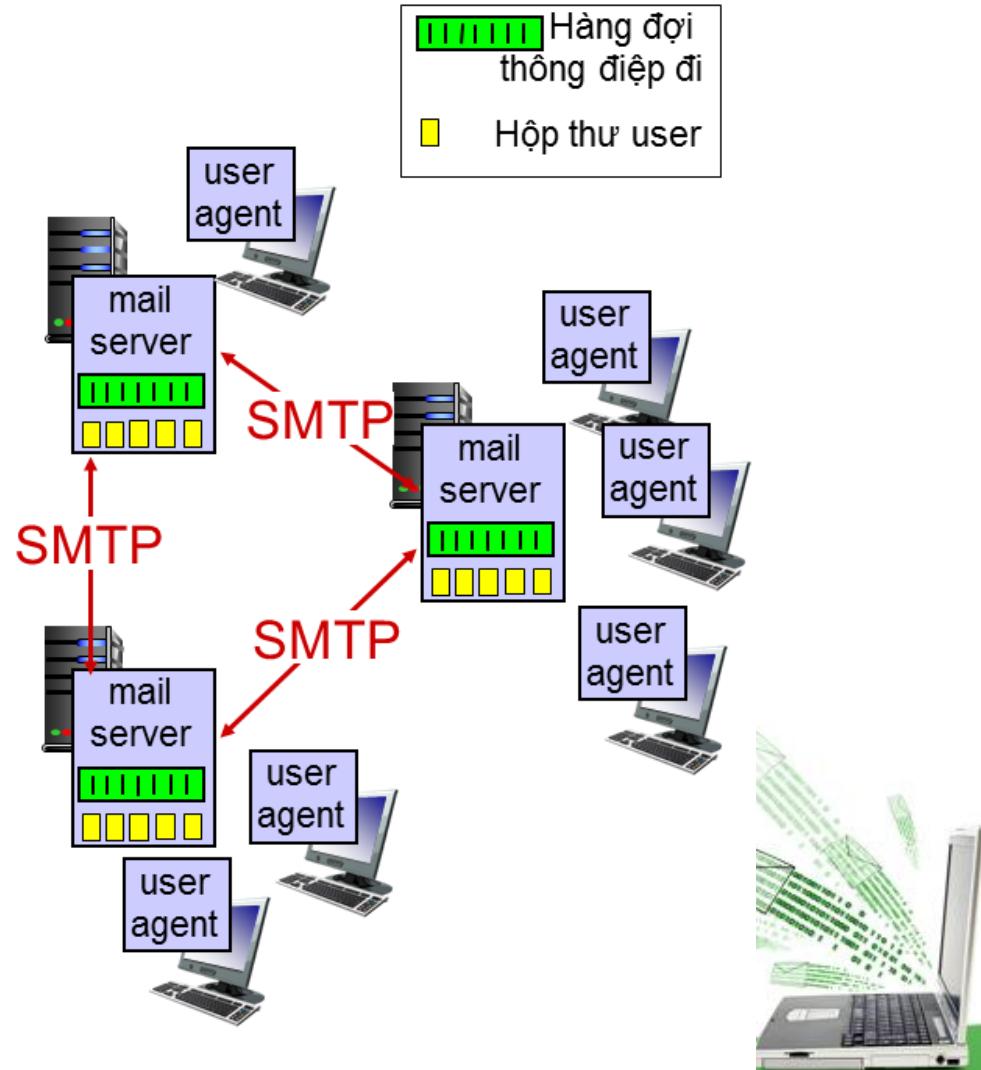
Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

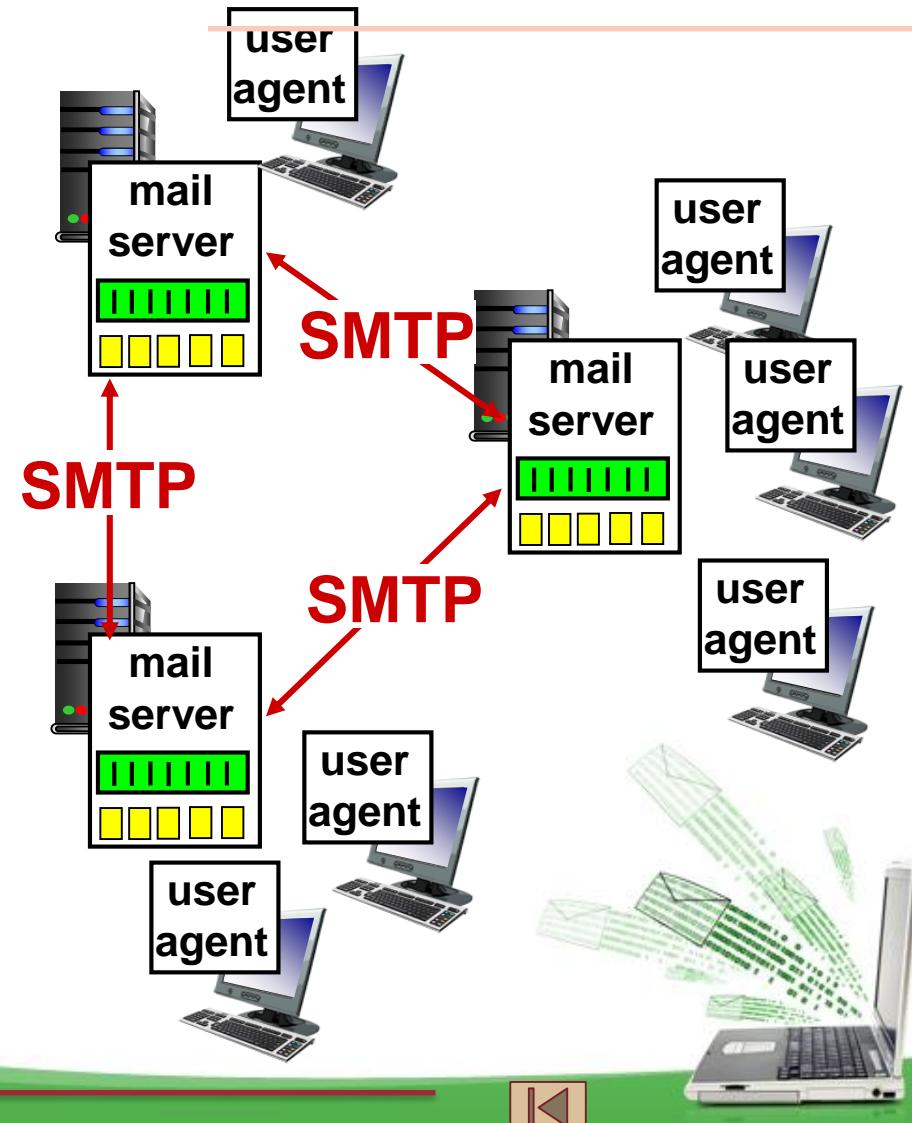
DNS



## Thư điện tử: mail servers

Mail servers:

- **Hộp thư (mailbox)** chứa thông điệp đến user
- **Hàng thông điệp (message queue)** của các thông điệp mail ra ngoài (chuẩn bị gửi)
- **Giao thức SMTP** giữa các mail server để gửi các thông điệp email
  - client: mail server gửi
  - “server”: mail server nhận



Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS



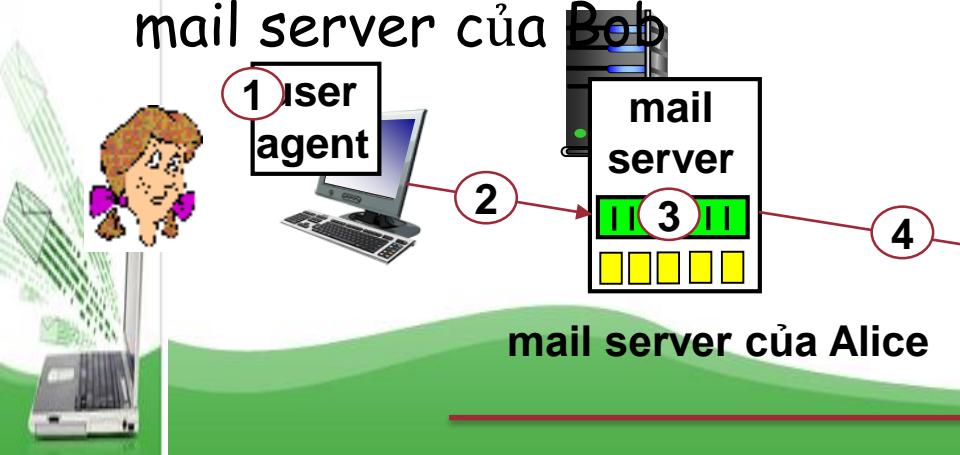
## Thư điện tử: SMTP [RFC 2821]

- Sử dụng TCP để truyền thông điệp email một cách tin cậy từ client đến cổng 25 server
- Truyền trực tiếp: server gửi đến server nhận
- 3 giai đoạn truyền
  - bắt tay (chào hỏi)
  - truyền thông điệp
  - đóng
- Tương tác lệnh/phản hồi (như HTTP, FTP)
  - Lệnh: văn bản ASCII
  - Phản hồi: mã và cụm trạng thái
- Thông điệp phải ở dạng mã ASCII 7 bit

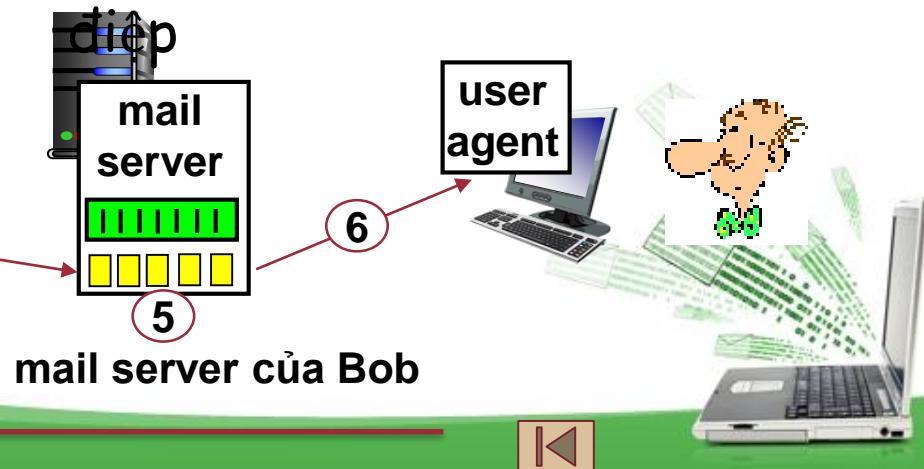


## Tình huống: Alice gửi thông điệp đến Bob

- 1) Alice dùng một UA để soạn thảo thông điệp "gởi đến" bob@someschool.edu
- 2) UA của Alice gửi thông điệp đến mail server của cô ta; thông điệp được đặt trong hàng đợi
- 3) Phần client của SMTP server mở kết nối TCP với mail server của Bob



- 4) Phần client của SMTP server gửi thông điệp của Alice trên kết nối TCP
- 5) Mail server của Bob đặt thông điệp đó trong hộp thư của Bob
- 6) Bob kích hoạt user agent của anh ta để đọc thông điệp



## Ví dụ tương tác SMTP

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



## Thử nghiệm tương tác SMNP:

- Telnet servername 25
- Xem trả lời 220 từ server
- Nhập các lệnh HELO, MAIL FROM, RCPT TO, DATA, QUIT

Lệnh ở trên cho phép bạn gửi email không cần dùng email client (reader)

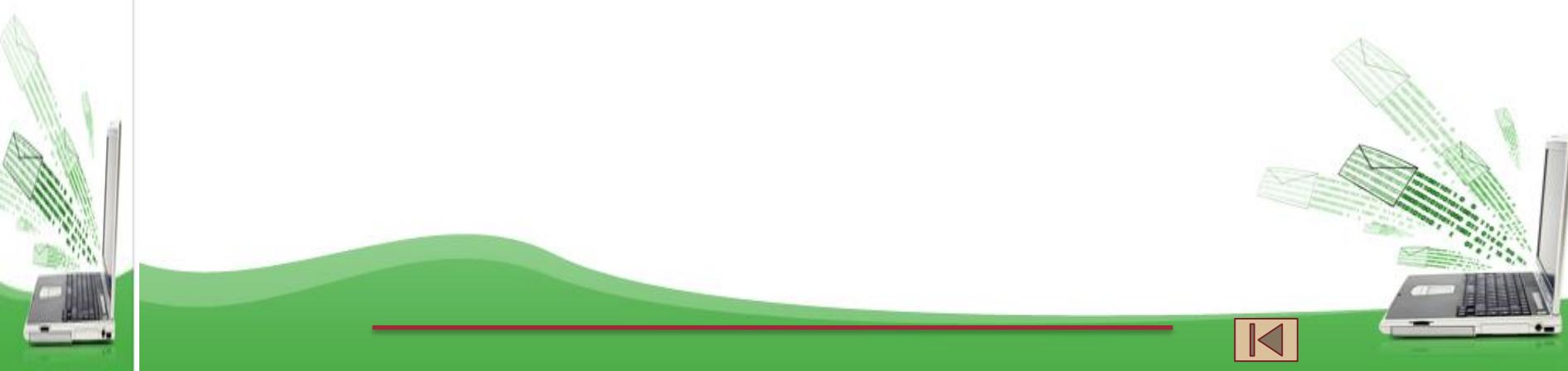
Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS



## SMTP: kết luận

- SMTP dùng kết nối bền vững
- SMTP yêu cầu thông điệp (header & body) phải ở dạng ASCII 7-bit
- SMTP server dùng uses CRLF.CRLF để xác định kết thúc thông điệp

### *So sánh với HTTP:*

- HTTP: pull (kéo)
- SMTP: push (đẩy)
- Cả hai đều có tương tác lệnh/phản hồi, các mã trạng thái dạng ASCII
- HTTP: mỗi đối tượng được đóng gói trong thông điệp phản hồi của nó
- SMTP: nhiều đối tượng được gửi trong thông điệp chứa nhiều phần

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

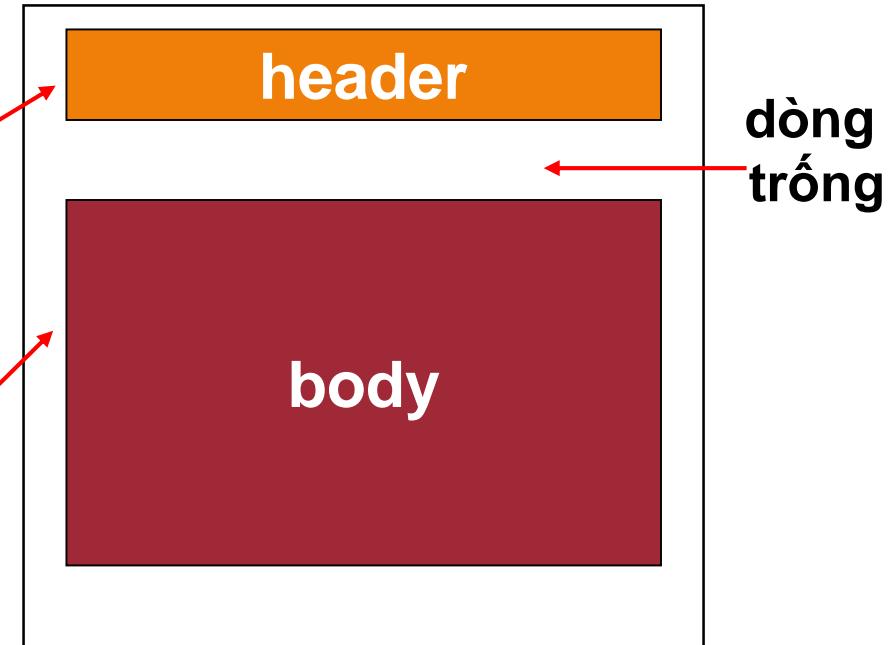


## Định dạng thông điệp Mail

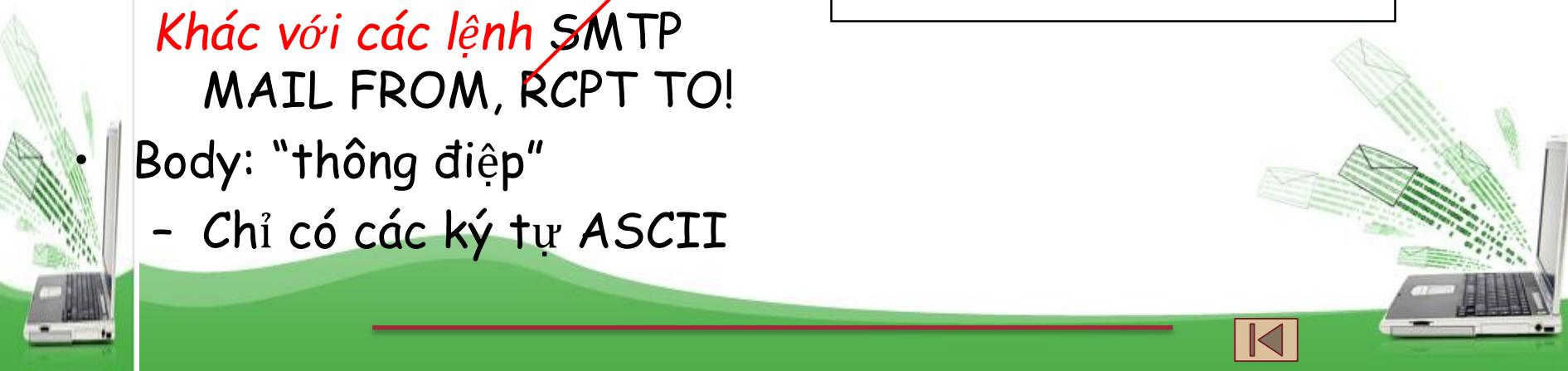
SMTP: giao thức dùng cho trao đổi thông điệp email

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

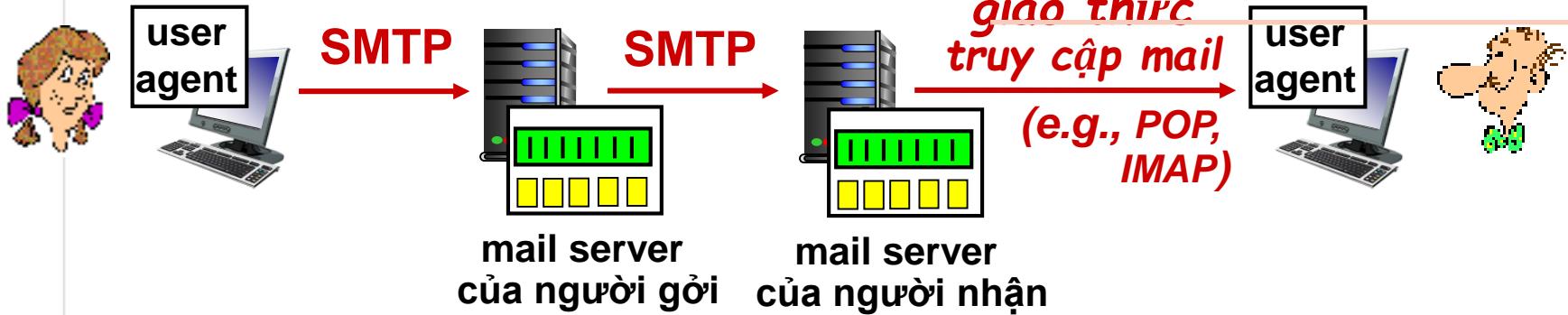
- Các dòng header, ví dụ
  - To:
  - From:
  - Subject:
- Khác với các lệnh SMTP MAIL FROM, RCPT TO!
- Body: "thông điệp"
  - Chỉ có các ký tự ASCII



dòng  
trống



## Các giao thức truy cập Mail



- **SMTP:** truyền dẫn/lưu trữ thư vào server của người nhận
- **Giao thức truy cập mail:** trích xuất từ server
  - **POP:** Post Office Protocol [RFC 1939]: xác thực, tải thư về
  - **IMAP:** Internet Mail Access Protocol [RFC 1730]: nhiều tính năng hơn, bao gồm cả các thao tác thay đổi các thông điệp đang được lưu trên server
  - **HTTP:** gmail, Hotmail, Yahoo! Mail...



# POP3 và IMAP

## Tìm hiểu thêm về POP3

- Ví dụ trên sử dụng chế độ “tải xuống và xóa” của POP3
  - Bob không thể đọc lại e-mail nếu anh ta thay đổi client
- Chế độ “tải xuống-và-giữ” của POP3: sao chép các thông điệp trên các client khác nhau
- POP3 không giữ trạng thái của các phiên làm việc

## IMAP

- Giữ tất cả các thông điệp ở một nơi: tại server
- Cho phép người dùng tổ chức, sắp xếp các thông điệp trong các thư mục
- Giữ trạng thái của người dùng trong suốt phiên làm việc:
  - Các tên của các thư mục và ánh xạ giữa các ID của thông điệp và tên của thư mục

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS



## DNS: domain name system

*Con người:* nhiều cách nhận dạng:

- Số an sinh xã hội, tên, số hộ chiếu

*Internet hosts, routers:*

- Địa chỉ IP (32 bit) - được dùng cho định địa chỉ gói tin
- “tên”, ví dụ www.yahoo.com - được dùng bởi con người

Q: làm cách nào để ánh xạ giữa địa chỉ IP và tên, và ngược lại?

**Domain Name System:**

- *Cơ sở dữ liệu phân tán* được thực hiện theo tổ chức phân cấp của nhiều *name server*
- *Giao thức tầng application:* các host, các name server trao đổi để *phân giải* tên (dịch địa chỉ  $\Leftrightarrow$  tên)
  - Lưu ý: chức năng trong phần lõi Internet, được thực hiện như là giao thức tầng application
  - Sự phức tạp ở “biên” của mạng”



# DNS: các dịch vụ, cấu trúc

## Các dịch vụ DNS

- Dịch tên máy ra địa chỉ IP
- Bí danh máy
  - Lưu các tên gốc, bí danh tương ứng
- Bí danh mail server
- Cân bằng tải
  - Các bản sao cho web server: nhiều địa chỉ IP tương ứng cho 1 tên

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

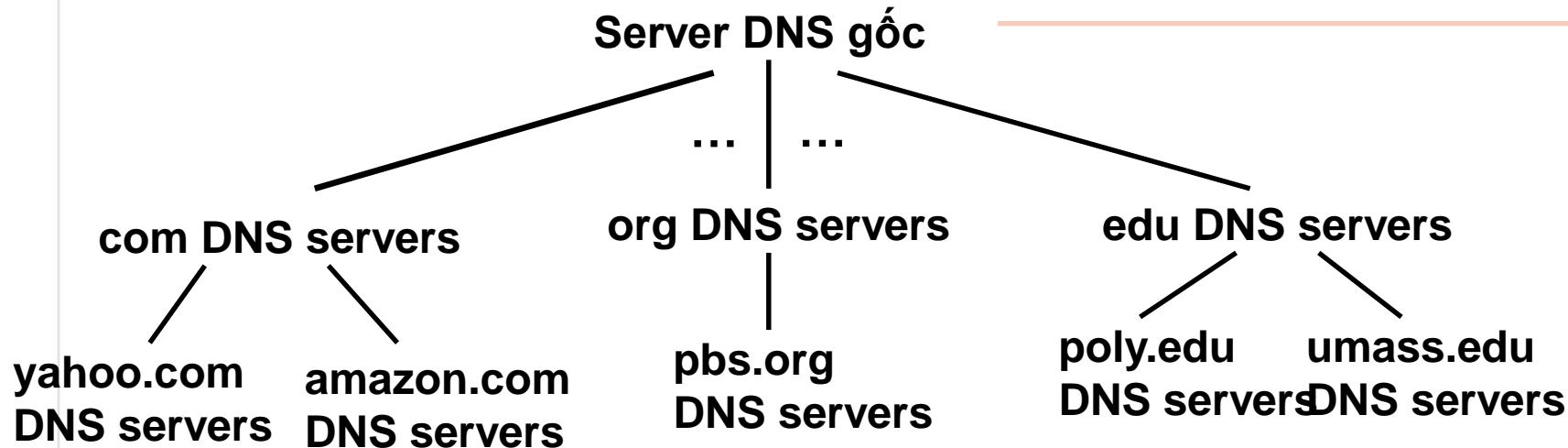
DNS

## Tại sao không tập trung hóa DNS?

- Một điểm chịu lỗi
- Lưu lượng
- Cơ sở dữ liệu tập trung cách xa nơi yêu cầu
- Bảo trì
  - A: *không biến đổi được quy mô!*



# DNS: cơ sở dữ liệu phân cấp, phân tán



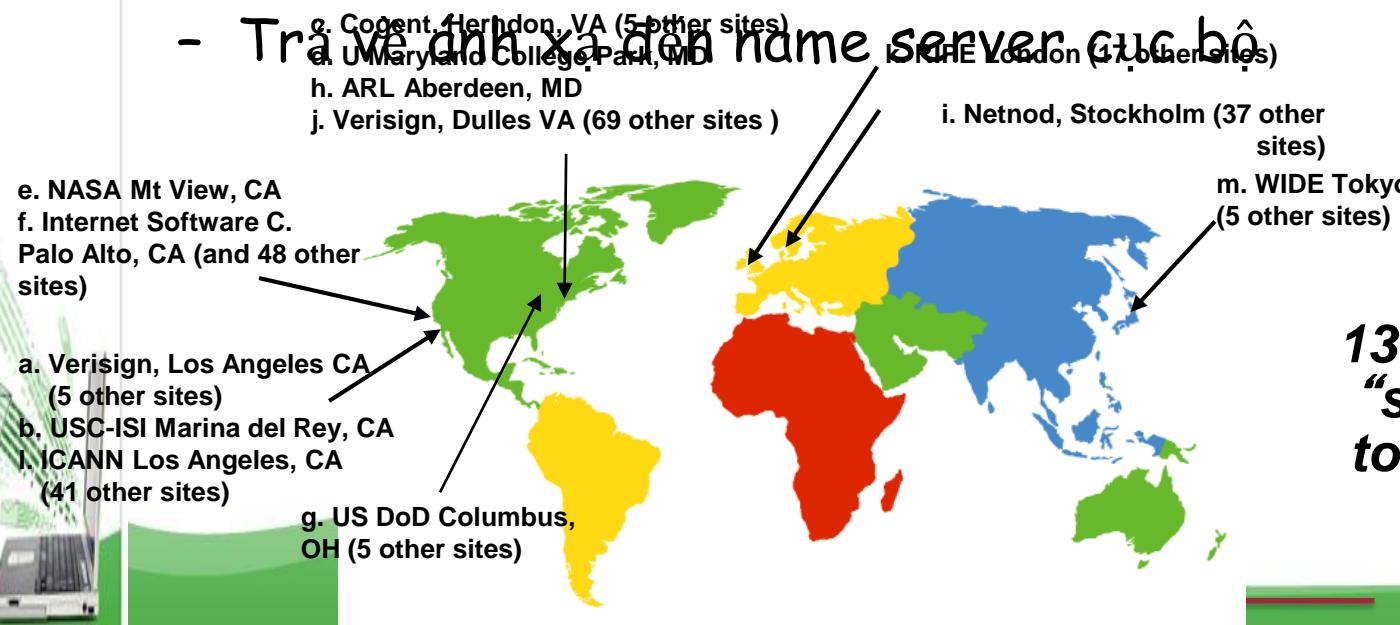
*client muốn địa chỉ IP của www.amazon.com:*

- client truy vấn server gốc (root) để tìm DNS server quản lý vùng ".com"
- client truy vấn DNS server ".com" tìm DNS server quản lý vùng "amazon.com"
- client truy vấn DNS server "amazon.com" để lấy địa chỉ IP của www.amazon.com



## DNS: các name server gốc

- Được name server cục bộ liên lạc để hỏi khi không thể phân giải tên
- name server gốc:
  - Liên lạc với name server có thẩm quyền (authoritative name server) nếu ánh xạ tên không xác định
  - Lấy ánh xạ
  - Trả về ánh xạ đến name server cục bộ



13 name  
“servers” gốc  
tổn cầu



## TLD, server có thẩm quyền

### Các top-level domain (TLD) server :

- Chịu trách nhiệm cho tên miền com, org, net, edu, aero, jobs, museums, và tất cả các tên miền cấp cao nhất của quốc gia, như là: uk, fr, ca, jp
- Công ty Network Solutions quản lý máy chủ chứa các thông tin của vùng .com TLD
- Tổ chức Educause quản lý .edu TLD

### Các DNS server có thẩm quyền:

- Các tổ chức sở hữu các DNS server riêng nhằm cung cấp các tên được cấp phép và ánh xạ địa chỉ IP cho các host được đặt tên của tổ chức đó
- Có thể được quản lý bởi tổ chức hoặc nhà cung cấp dịch vụ



## DNS name server cục bộ

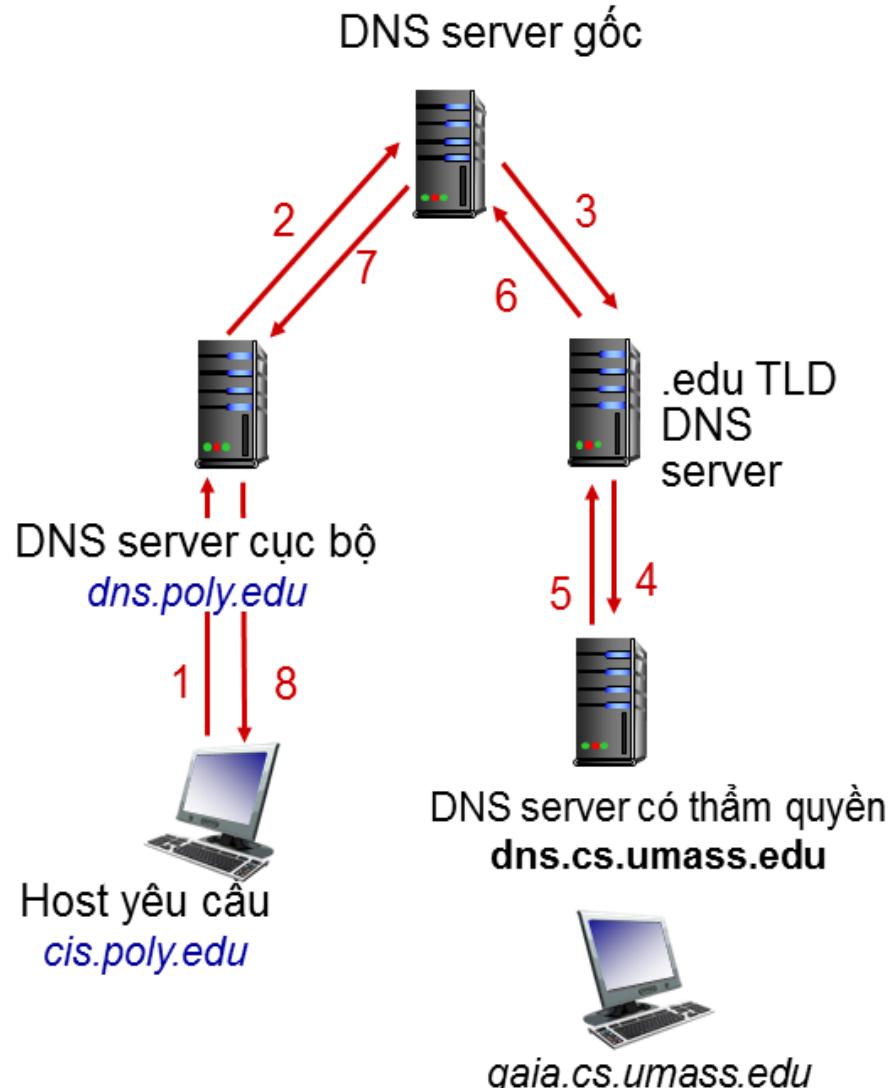
- Không hoàn toàn theo cấu trúc phân cấp
- Mỗi ISP (ISP, công ty, trường đại học) có một server cục bộ như vậy
  - Còn được gọi là “default name server”
- Khi một host tạo một truy vấn DNS, truy vấn được gửi đến DNS server cục bộ của nó
  - Có bộ nhớ đệm (cache) cục bộ chứa thông tin về các cặp tên-đến-địa chỉ gần đây (nhưng có thể hết hạn!)
  - Hoạt động như một proxy, chuyển truy vấn vào trong hệ thống DNS phân cấp



## Ví dụ phân giải tên miền DNS

### Truy vấn đệ quy:

- ❖ Đẩy trách nhiệm phân giải tên cho name server được hỏi
- ❖ Tải nặng tại các tầng trên của hệ thống phân cấp?



## DNS: caching, cập nhật các bản ghi

- Một khi name server biết về 1 ánh xạ tên-địa chỉ, nó sẽ **lưu tạm** ánh xạ đó
  - Các mục cache hết hạn (sẽ bị xóa) sau một khoảng thời gian (TTL)
  - Thông tin trong các TLD servers thường được lưu tạm trong các name server cục bộ
    - Do đó các name server gốc không bị truy cập thường xuyên
- Các mục được lưu tạm có thể hết hạn
  - Nếu cập thông tin tên-địa chỉ IP thay đổi, có thể các máy khác trên Internet không biết được cho đến khi tất cả TTL hết hạn.
- Cơ chế cập nhật/thông báo được đề xuất trong bộ chuẩn IETF
  - RFC 2136



## Các bản ghi DNS

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

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

### Type=A

- name là tên host
- value là địa chỉ IP

### Type=NS

- name là tên miền (e.g., foo.com)
- value là tên host của name server có thẩm quyền quản lý tên miền này

### Type=CNAME

- name là bí danh của một tên “gốc” (tên thực)
  - VD: www.ibm.com tên thực là servereast.backup2.ibm.com
- value là tên gốc

### Type=MX

- value là tên của mail server được liên kết với name

Các nguyên lý của các ứng dụng mạng  
Web và HTTP  
FTP  
Thư điện tử: SMTP, POP3, IMAP  
DNS



## Giao thức và các thông điệp DNS

- Các thông điệp truy vấn (query) và trả lời (reply) đều có cùng định dạng thông điệp

msg header

- Identification: số 16 bit xác định 1 truy vấn, hoặc trả lời cho truy vấn có cùng số này

- flags:
  - Truy vấn hoặc trả lời
  - Mong muốn đệ quy
  - Đệ quy sẵn sàng
  - Trả lời có thẩm quyền

identification	flags
# questions	# answer RRs
# authority RRs	# additional RRs
questions (variable # of questions)	
answers (variable # of RRs)	
authority (variable # of RRs)	
additional info (variable # of RRs)	



# Giao thức và các thông điệp DNS

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

← 2 bytes → ← 2 bytes →

identification	flags
# questions	# answer RRs
# authority RRs	# additional RRs
questions (variable # of questions)	
answers (variable # of RRs)	
authority (variable # of RRs)	
additional info (variable # of RRs)	

Các trường name,  
type cho một truy vấn

Các RRs để trả  
lời truy vấn

Các bản ghi thông tin về  
các server có thẩm quyền

thông tin “hữu ích”  
bổ sung có thể sẽ dùng



## Thêm các bản ghi vào trong DNS

- Ví dụ: khởi tạo mới công ty “Network Utopia”
- Đăng ký tên miền networkutopia.com tại một **DNS registrar**
  - **tổ chức nhận đăng ký tên miền** (như là Network Solutions)
    - Cung cấp tên, địa chỉ IP của name server có thẩm quyền quản lý tên miền này (primary và secondary)
    - Tổ chức quản lý tên miền thêm hai bản ghi vào trong server quản lý vùng .com:  
(networkutopia.com, dns1.networkutopia.com, NS)  
(dns1.networkutopia.com, 212.212.212.1, A)
- Tạo bản ghi type A trong server có thẩm quyền quản lý networkutopia.com cho www.networkutopia.com; và bản ghi type MX cho mail server thuộc networkutopia.com



# Tấn công DNS

## Tấn công DDoS

- Đẩy khối lượng dữ liệu lớn tới các server gốc
  - Không thành công cho đến nay
  - Lọc lưu lượng
  - Các DNS server cục bộ lưu tạm các địa chỉ IP của TLD servers, vì thế không cần truy cập server gốc
- Đẩy khối lượng dữ liệu lớn tới TLD server
  - Nguy hiểm hơn

Các nguyên lý của các ứng dụng mạng

Web và HTTP

FTP

Thư điện tử: SMTP, POP3, IMAP

DNS

## Tấn công chuyển hướng

- Man-in-middle
  - Ngăn chặn các truy vấn
- Đầu độc DNS
  - Gửi các trả lời giả tạo đến các DNS server, (sẽ được các server lưu lại)

## Khai thác DNS cho tấn công DDoS

- Gởi các truy vấn với địa chỉ nguồn giả mạo: địa chỉ IP mục tiêu
- Yêu cầu khuếch đại



# Tham khảo

- Phần này chỉ những:
  - Sách/ chỉ mục bắt buộc học
  - Bài tập phải làm.
  - Sách/ chương đọc thêm.



# YÊU CẦU SỬ DỤNG FILE MẪU

- Slide số 2 (nội dung chính): khi click vào các nội dung chính 1,2,3... sẽ cho link tới các slide tương ứng.
- Tất cả các slide bên dưới có nút : khi click vào sẽ quay về slide số 2 (chứa nội dung chính)
- Phần nội dung chính của bài (để góc trên bên phải) khi soạn tới phần nào cho đổi màu nội dung phần đó. Nội dung này chính là slide số 2.
- Những slide có phần layout giống nhau nên sử dụng chức năng “Duplicate slide”.
- Phần hiệu ứng Thầy/Cô tùy chọn (nên đồng bộ cho tất cả slide)

**(Vui lòng xóa slide này khi đọc xong)**

# Mạng máy tính

ThS. Phạm Liệu  
*Email: lieu.pham@stu.edu.vn*

2018



# NỘI DUNG CHÍNH



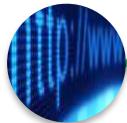
Các dịch vụ tầng vận chuyển



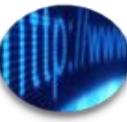
Vận chuyển phi kết nối: UDP



Các nguyên lý truyền dữ liệu tin cậy



Vận chuyển hướng kết nối: TCP



Các nguyên lý về điều khiển tắt nghẽn



Điều khiển tắt nghẽn TCP



# Các giao thức và dịch vụ tầng Vận chuyển

- Cung cấp *truyền thông logic* giữa các tiến trình ứng dụng đang chạy trên các host khác nhau
- Các giao thức (protocol) chạy trên các hệ thống đầu cuối
  - Phía gửi: chia nhỏ các thông điệp (message) ứng dụng thành các *segments*, sau đó chuyển các segments này cho tầng Mạng
  - Phía nhận: tái kết hợp các segments thành các thông điệp (message), các thông điệp này được chuyển lên tầng Ứng dụng
- Có nhiều hơn 1 giao thức tầng Vận chuyển dành cho các ứng dụng
  - Internet: TCP và UDP

Các dịch vụ tầng vận chuyển

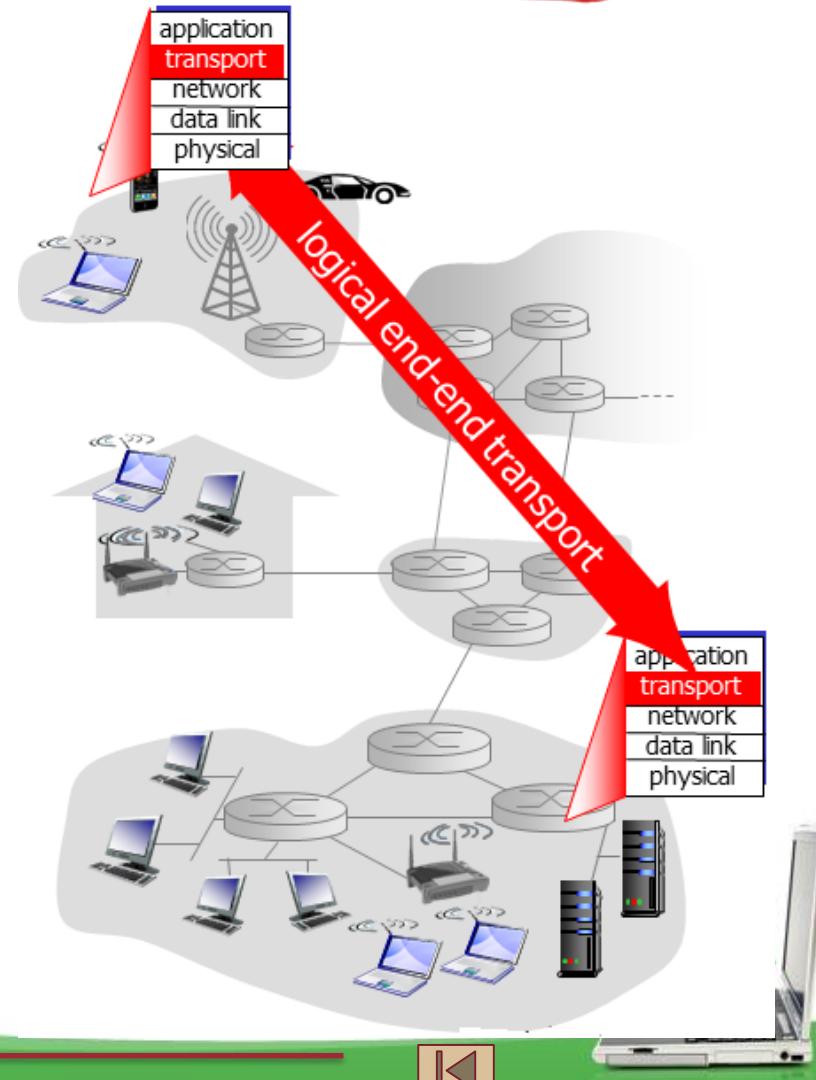
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



# Quan hệ giữa Tầng Vận chuyển và tầng Mạng

- **Tầng Mạng:** truyền thông logic giữa các host
- **Tầng Vận chuyển:** truyền thông logic giữa các tiến trình
  - Dựa vào và tăng cường các dịch vụ tầng Mạng

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

## Tình huống tương tự:

- 12 đứa trẻ ở nhà Ann gửi những bức thư đến 12 đứa trẻ ở nhà Bill:
- hosts = nhà
  - Các tiến trình (processes) = những đứa trẻ
  - Thông điệp tầng Ứng dụng = các bức thư trong các phong bì
  - Giao thức tầng Vận chuyển = Ann and Bill
  - Giao thức tầng Mạng = dịch vụ bưu điện



# Các giao thức tầng Vận chuyển trên Internet

- Tin cậy, truyền theo thứ tự (TCP)
  - Điều khiển tắc nghẽn
  - Điều khiển luồng
  - Thiết lập kết nối
- Không tin cậy, truyền không theo thứ tự: UDP
  - Không rườm rà, mở rộng "nỗ lực tốt nhất" của IP
- Không có các dịch vụ:
  - Bảo đảm độ trễ
  - Bảo đảm băng thông

Các dịch vụ tầng vận chuyển

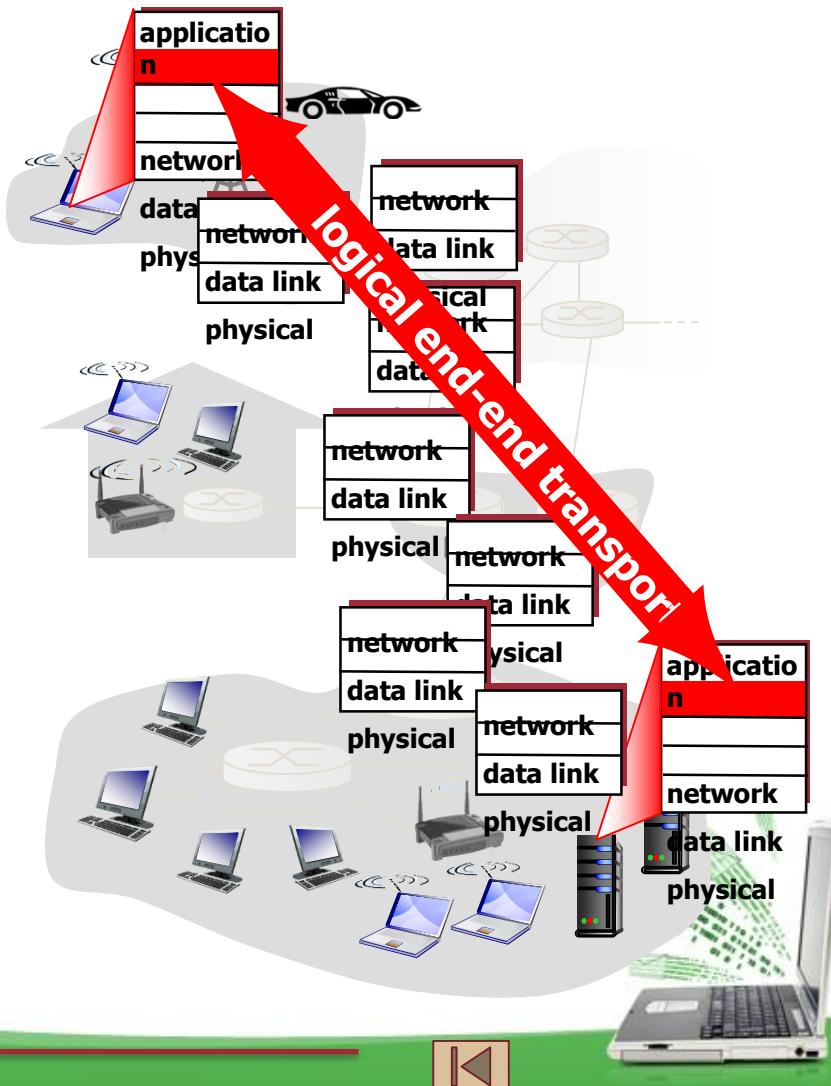
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP



# UDP: User Datagram Protocol

## [RFC 768]

- “đơn giản,” là giao thức thuộc tầng Vận chuyển
- Dịch vụ “best effort” (“nỗ lực tốt nhất”), các segment UDP có thể bị:
  - Mất mát
  - Vận chuyển không theo thứ tự đến ứng dụng đích
- *Connectionless (phi kết nối):*
  - Không bắt tay giữa bên nhận và gửi UDP
  - Mỗi segment UDP được xử lý độc lập

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

### ❖ Ứng dụng UDP:

- Các ứng dụng đa phương tiện trực tuyến (chiều mất mát loss tolerant), (cần tốc độ) (rate sensitive)
- DNS
- SNMP

### ❖ Truyền tin cậy trên UDP:

- Thêm độ tin cậy tại tầng Ứng dụng
- Phục hồi lỗi tại các ứng dụng cụ thể!



# UDP: segment header

Các dịch vụ tầng vận chuyển

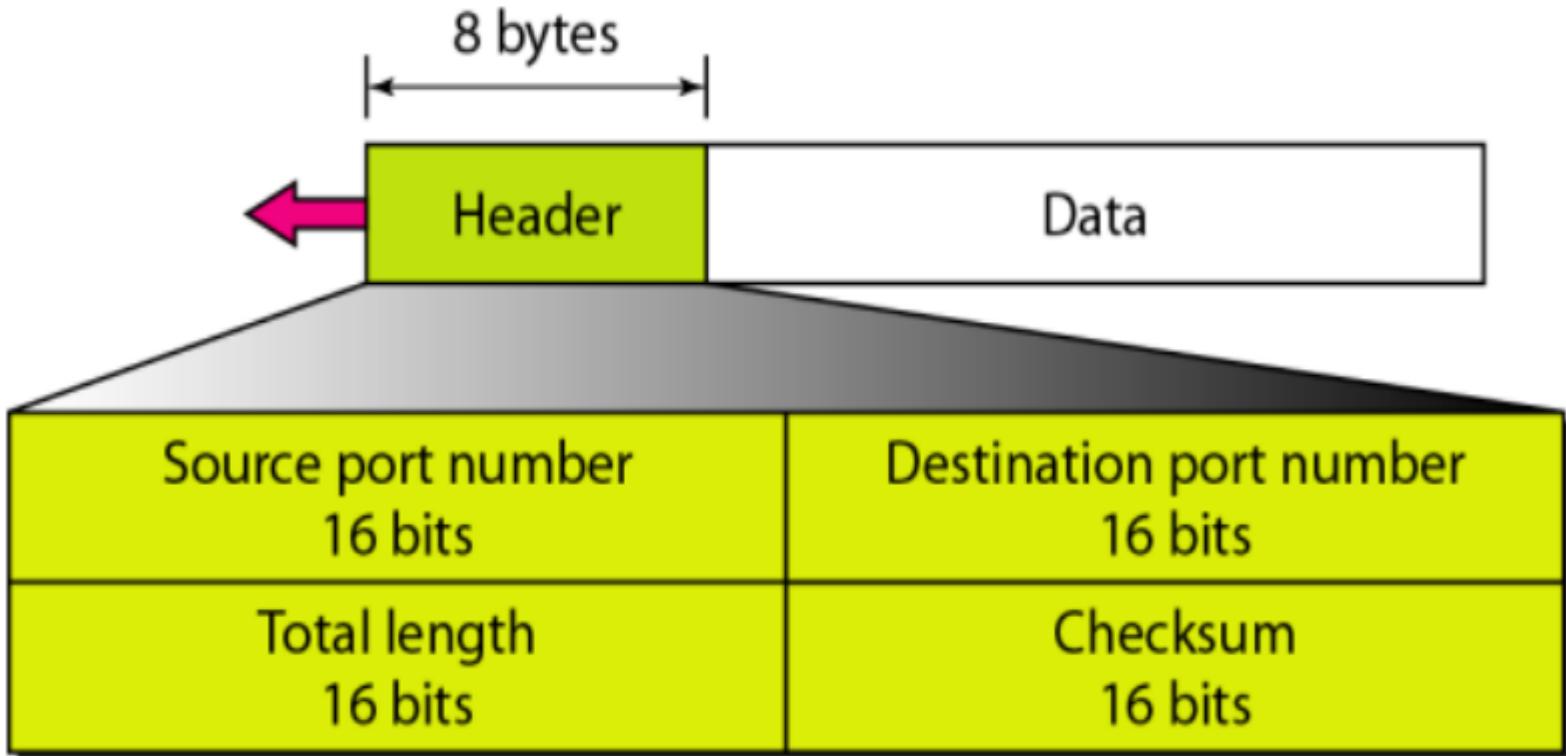
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



## Định dạng segment UDP



## UDP: segment header

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

## Tại sao có UDP?

- Không thiết lập kết nối (có thể gây ra độ trễ)
- Đơn giản: không trạng thái kết nối tại nơi gửi và nhận
- Kích thước header nhỏ
- Không điều khiển tắc nghẽn: UDP có thể gửi dữ liệu nhanh như mong muốn



## UDP checksum

**Mục tiêu:** dò tìm “các lỗi” (các bit cờ được bật) trong các segment đã được truyền

bên gửi:

- Xét nội dung của segment, bao gồm các trường của header, là chuỗi các số nguyên 16-bit
- checksum: tổng bù 1 của các chuỗi số 16 bit trong nội dung segment
- Bên gửi đặt giá trị checksum vào trường checksum UDP

bên nhận:

- Tính toán checksum của segment đã nhận
- Kiểm tra giá trị trên có bằng với giá trị trong trường checksum hay không:
  - NO - có lỗi xảy ra
  - YES - không có lỗi. Nhưng có thể còn lỗi khác nữa không? Xem phần sau....

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



## Internet checksum: ví dụ

Ví dụ: cộng 2 số nguyên 16 bit

$$\begin{array}{r} 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \\ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \\ \hline \end{array}$$

bit dư

tổng

checksum

1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1

1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0  
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

Lưu ý: khi cộng các số, bit nháy ở phía cao nhất cần được thêm vào kết quả

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

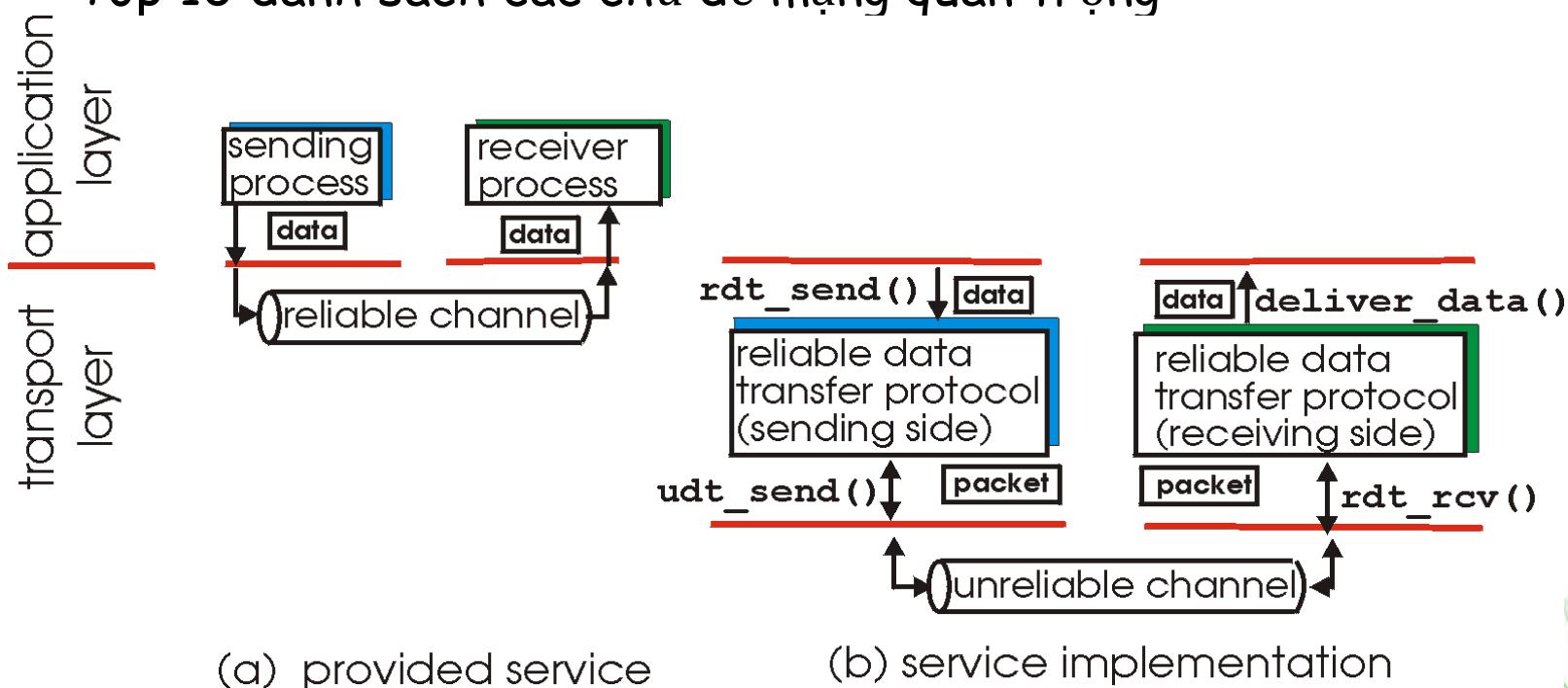
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

# Các nguyên lý truyền dữ liệu tin cậy

- Quan trọng trong các tầng Ứng dụng, Vận chuyển và Liên kết dữ liệu
  - Top 10 danh sách các chủ đề mạng quan trọng

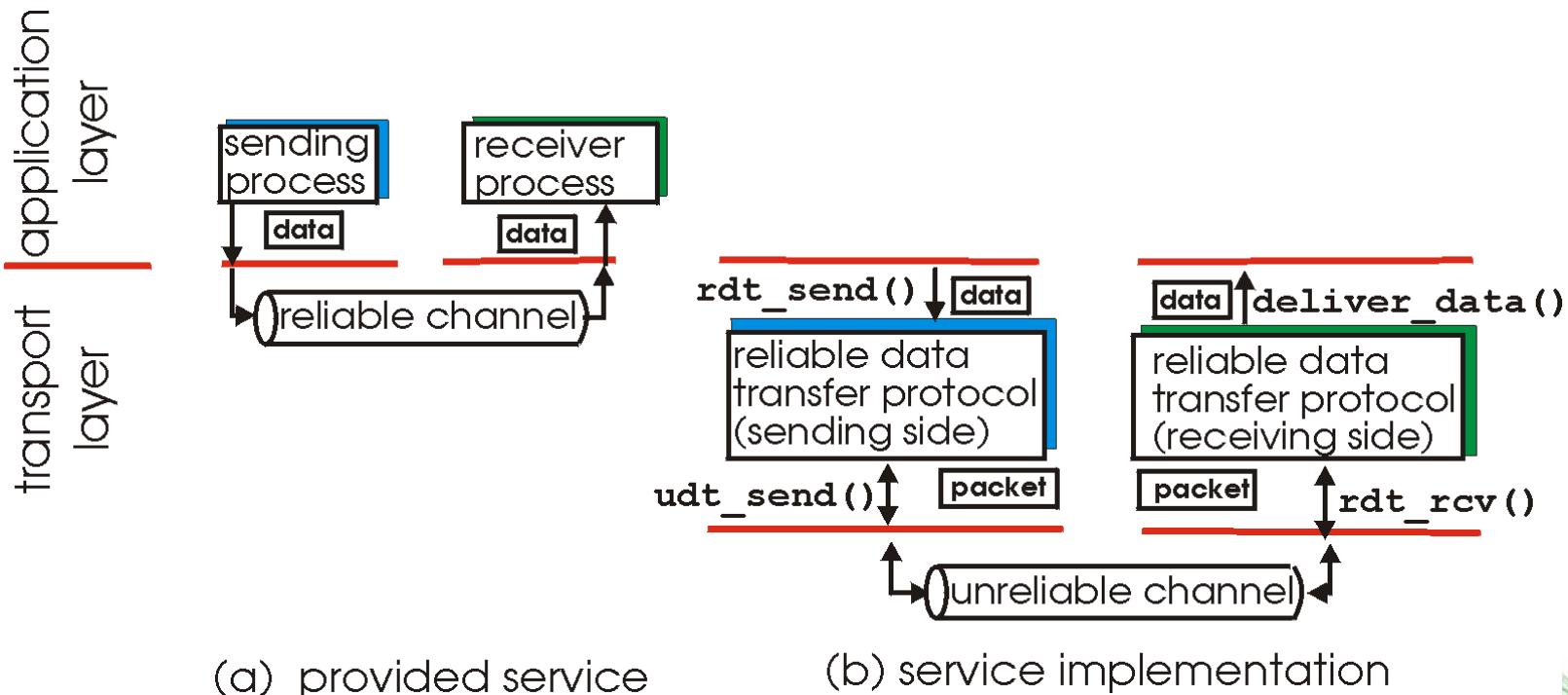


- Các đặc điểm của kênh truyền không tin cậy sẽ xác định sự phức tạp của giao thức truyền dữ liệu (data transfer protocol) (rdt)



# Các nguyên lý truyền dữ liệu tin cậy

- Quan trọng trong các tầng Ứng dụng, Vận chuyển và Liên kết dữ liệu
  - Top 10 danh sách các chủ đề mạng quan trọng



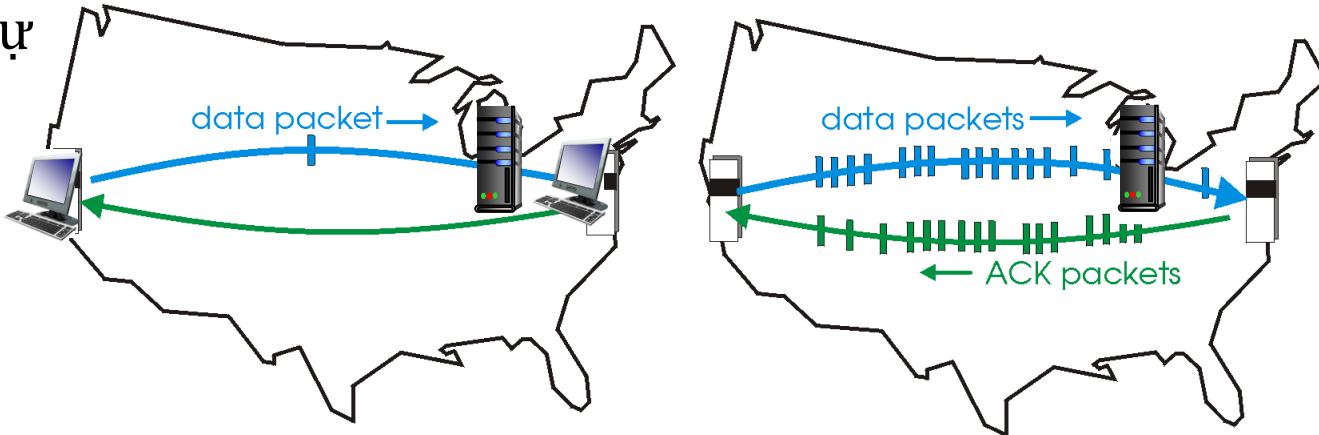
- Các đặc điểm của kênh truyền không tin cậy sẽ xác định sự phức tạp của giao thức truyền dữ liệu (data transfer protocol) (rdt)



## Các giao thức Pipelined

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

- Bên gửi: lưu gói tin đã gửi nhưng chưa ACK
- Bên nhận: lưu gói tin đã nhận đúng nhưng chưa đúng thứ tự



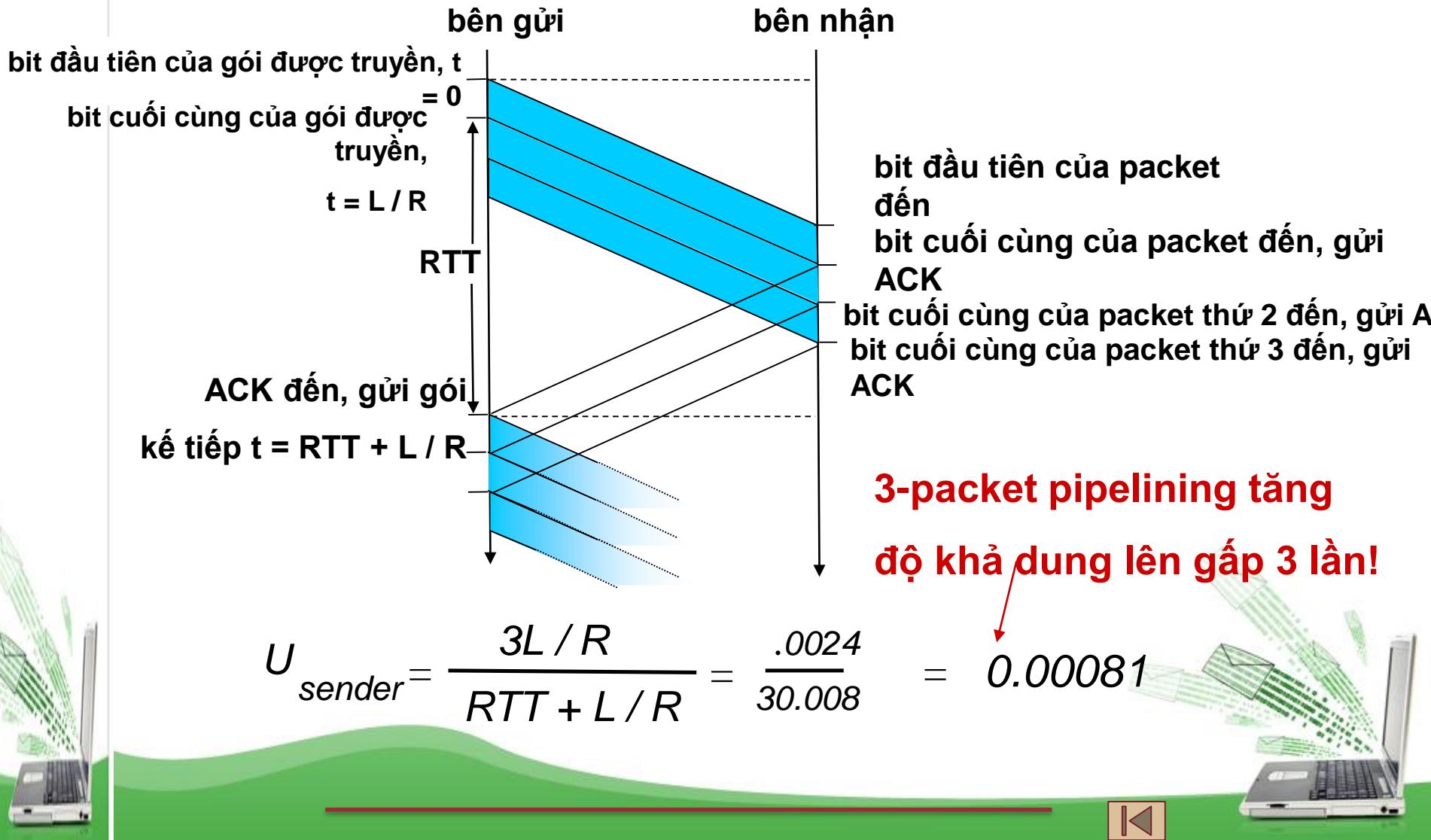
(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

- hai dạng phổ biến của các giao thức pipelined : *go-Back-N, selective repeat (lặp có lựa chọn)*



# Pipelining: độ khả dụng tăng



# Pipelined protocols: tổng quan

## Go-back-N:

- Bên gửi có thể có đến N packet không cần ACK trong đường ống (pipeline)
- Bên nhận chỉ gửi *cumulative ack* (*xác nhận tích lũy*)
  - Sẽ không thông báo nhận packet thành công nếu có gián đoạn
- Bên gửi có bộ định thì cho packet sớm nhất mà không cần ACK (*oldest unacked packet*)
  - Khi bộ định thì hết, truyền lại tất cả các packet mà không được ACK

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

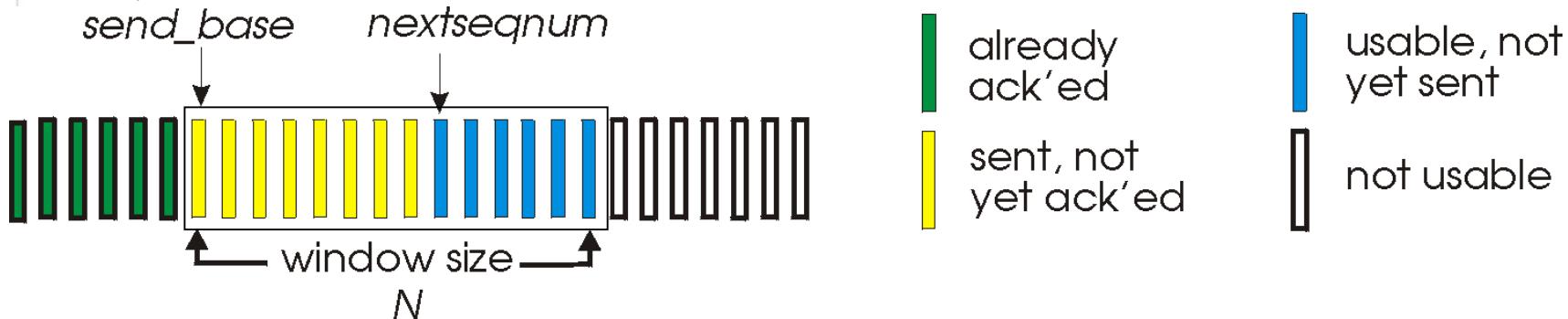
## Lắp có lựa chọn (Selective Repeat):

- Bên gửi có thể có đến N packet không cần ACK trong đường ống (pipeline)
- Bên nhận gửi rcvr ack riêng biệt (*individual ack*) cho mỗi packet
- Bên nhận duy trì bộ định thì cho mỗi packet không được ACK
  - Khi bộ định thì của packet nào hết hạn, thì chỉ truyền lại packet không được ACK đó



## Go-Back-N: bên gửi

- Số thứ tự k-bit trong header của packet
- "cửa sổ"("window") lên đến N gói, cho phép gửi liên tiếp không cần ACK



- ACK(n): thông báo nhận tất cả các packet lên đến n, bao gồm n số thứ tự - "**ACK tích lũy**" ("cumulative ACK")
  - Có thể nhận ACK trùng (xem bên nhận)
- Định thì cho packet sớm nhất đang trong tiến trình xử lý (oldest in-flight pkt)
- timeout(n): truyền lại packet n và tất cả các packet có số thứ tự cao hơn trong cửa sổ (window)

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

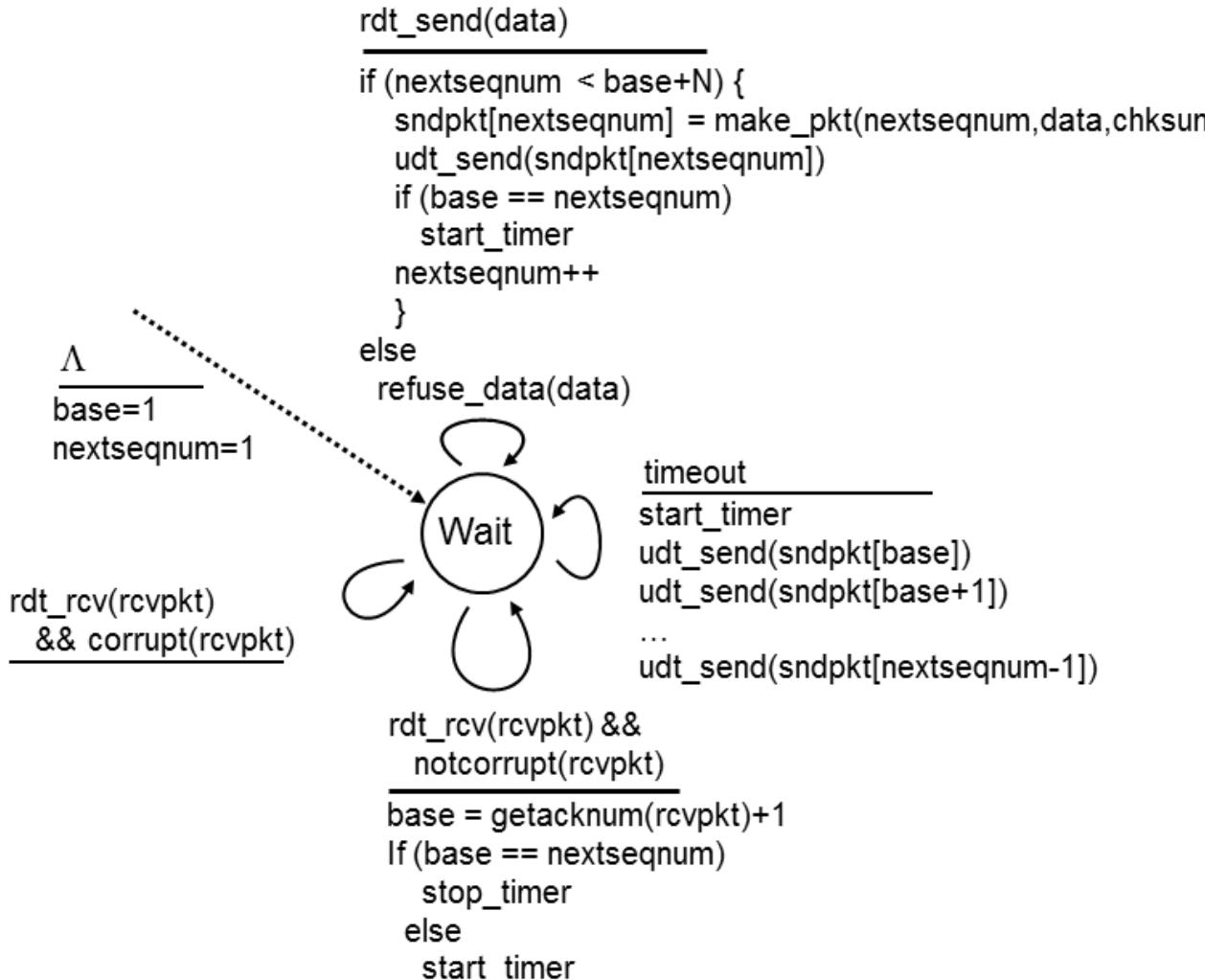
Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

# GBN: trạng thái mở rộng tại bên gửi



Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

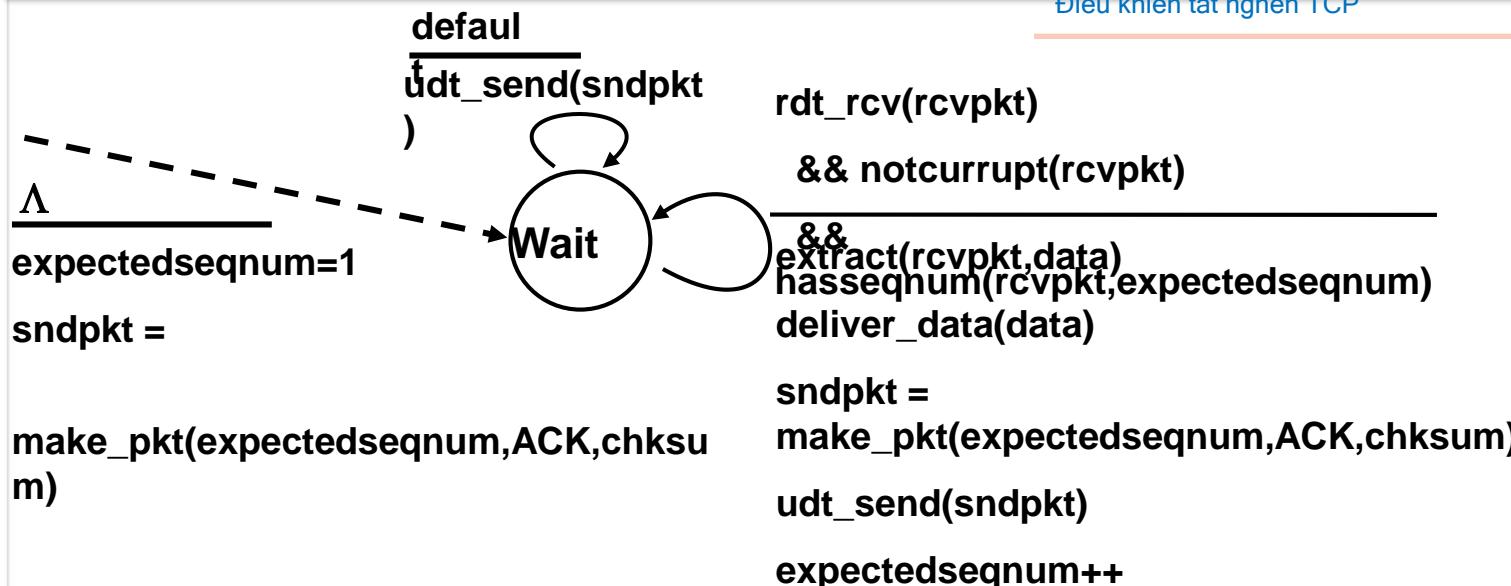
Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

# GBN: trạng thái mở rộng tại bên nhận



ACK- duy nhất: luôn luôn gửi ACK cho gói đã nhận chính xác, với số thứ tự xếp hạng cao nhất (highest *in-order* seq #)

- Có thể sinh ra các ACK trùng nhau
- Chỉ cần nhớ **expectedseqnum**
- Packet không theo thứ tự(out-of-order pkt):
  - hủy (discard): *không giữ trong bộ đệm bên nhận!*
  - Gửi lại ACK với số thứ tự xếp hạng cao nhất



# Hoạt động GBN

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

sender window (N=4)

0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8

bên gửi

send pkt0  
send pkt1  
send pkt2  
send pkt3  
(wait)

0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8

rcv ack0, send pkt4  
rcv ack1, send pkt5

bên nhận

receive pkt0, send ack0  
receive pkt1, send ack1

receive pkt3, discard,  
(re)send ack1

receive pkt4, discard,  
(re)send ack1  
receive pkt5, discard,  
(re)send ack1

0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8



*pkt 2 timeout*

send pkt2  
send pkt3  
send pkt4  
send pkt5

rcv pkt2, deliver, send ack2  
rcv pkt3, deliver, send ack3  
rcv pkt4, deliver, send ack4  
rcv pkt5, deliver, send ack5



## Lắp có lựa chọn (Selective repeat)

- Bên nhận thông báo đã nhận đúng tất cả từng gói một
  - Đếm các gói, khi cần thiết
- Bên gửi chỉ gửi lại các gói nào không nhận được ACK
  - Bên gửi đếm thời gian cho mỗi gói không có ACK
- Cửa sổ bên gửi (sender window)
  - N số thứ tự liên tục
  - Hạn chế số thứ tự các gói không có phản hồi ACK

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

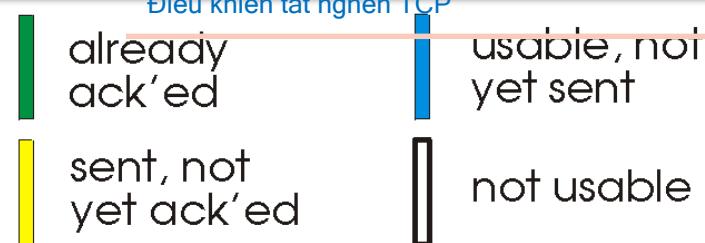
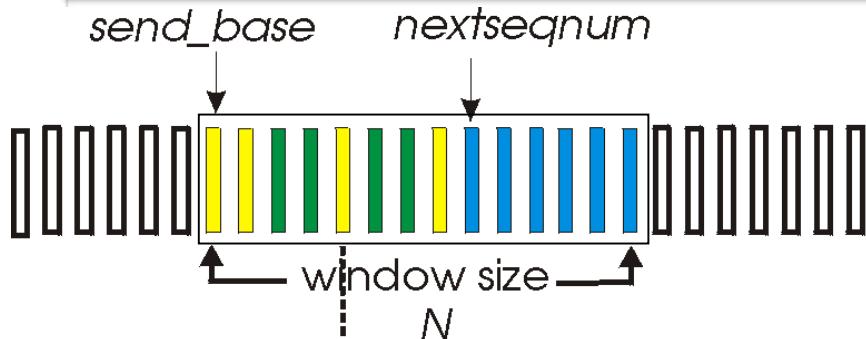
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

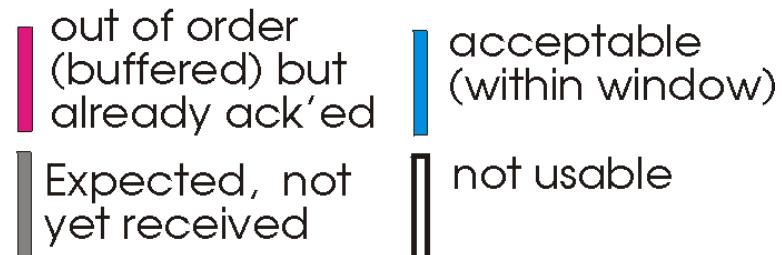
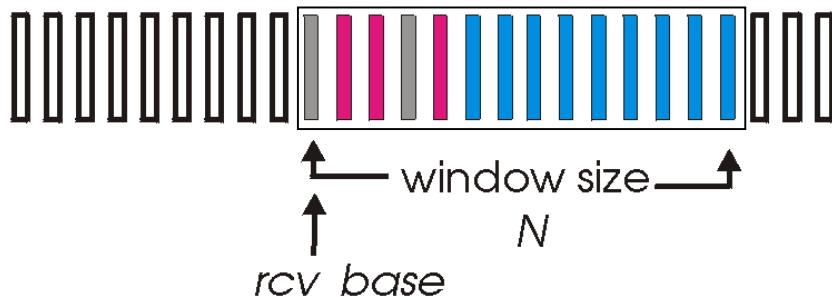
Điều khiển tắt nghẽn TCP



# Lắp có lựa chọn: cửa sổ bên gửi và nhận



(a) sender view of sequence numbers



(b) receiver view of sequence numbers

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



## Lắp có lựa chọn

### Bên gửi

Dữ liệu từ tầng trên:

- Nếu số thứ tự kế tiếp sẵn sàng trong cửa sổ, gửi gói  $\text{timeout}(n)$ :

- Gửi lại packet n, khởi động lại bộ đếm thời gian

$\text{ACK}(n)$  trong  $[\text{sendbase}, \text{sendbase}+N]$ :

- Đánh dấu packet n là đã được nhận
- Nếu gói chưa ACK có n nhỏ nhất, thì dịch chuyển cửa sổ base đến số thứ tự chưa ACK

Kế tiếp

### Bên nhận

Gói n trong  $[\text{rcvbase}, \text{rcvbase}+N-1]$

- Gửi  $\text{ACK}(n)$
- Không thứ tự: đệm
- Đúng thứ tự: chuyển dữ liệu lên tầng trên (cả các gói đã đệm, có thứ tự), dịch chuyển cửa sổ đến ô nhớ chờ gói chưa nhận kế tiếp

Packet n trong  $[\text{rcvbase}-N, \text{rcvbase}-1]$

- $\text{ACK}(n)$
- Ngược lại:
- Bỏ qua



## Hành động của lặp lại có lựa chọn

sender window (N=4)

0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8

0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8

nhận ack0, gửi pkt4

nhận ack1, gửi pkt5

Ghi nhận ack3 đã đến

**pkt 2 timeout**

gửi pkt2

Ghi nhận ack4 đã đến

Ghi nhận ack5 đã đến

**Bên gửi**

gửi pkt0

gửi pkt1

gửi pkt2

gửi pkt3

(đợi)

nhận ack0, gửi pkt4

nhận ack1, gửi pkt5

Ghi nhận ack3 đã đến

pkt 2 timeout

gửi pkt2

Ghi nhận ack4 đã đến

Ghi nhận ack5 đã đến

**Bên nhận**

nhận pkt0, gửi ack0

nhận pkt1, gửi ack1

nhận pkt3, buffer,

gửi ack3

nhận pkt4, buffer,

gửi ack4

nhận pkt5, buffer,

gửi ack5

nhận pkt2; chuyển pkt2,  
pkt3, pkt4, pkt5; gửi ack2

*Q: việc gì xảy ra khi ack2 đến?*



## Chương 3 Nội dung

3.1 các dịch vụ tầng  
Vận chuyển

3.2 3.3 vận chuyển phi  
kết nối: UDP

3.4 các nguyên lý  
truyền dữ liệu tin  
cây

3.5 vận chuyển hướng  
kết nối: TCP

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

3.6 các nguyên lý về điều  
khiển tắc nghẽn

3.7 điều khiển tắc nghẽn  
TCP



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

- **point-to-point:**
  - Một bên gửi, một bên nhận
- **Tin cậy, luồng byte theo thứ tự (in-order byte steam):**
  - Không "ranh giới thông điệp" ("message boundaries")
- **pipelined:**
  - Điều khiển luồng và tắc nghẽn của TCP thông qua việc thiết lập kích thước cửa sổ (window size)
- **Dữ liệu full duplex:**
  - Luồng dữ liệu đi 2 chiều trong cùng 1 kết nối
  - MSS: kích thước tối đa của gói tin (maximum segment size)
- **Hướng kết nối:**
  - Bắt tay (trao đổi các thông điệp điều khiển) khởi tạo trạng thái bên gửi và nhận trước khi trao đổi dữ liệu
- **Điều khiển luồng:**
  - Bên gửi sẽ không làm tràn bộ đệm bên nhận

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



# Cấu trúc segment TCP

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

URG: dữ liệu khẩn cấp  
(thường không dùng)

ACK: ACK #

hợp lệ

PSH: push data now  
(thường không dùng)

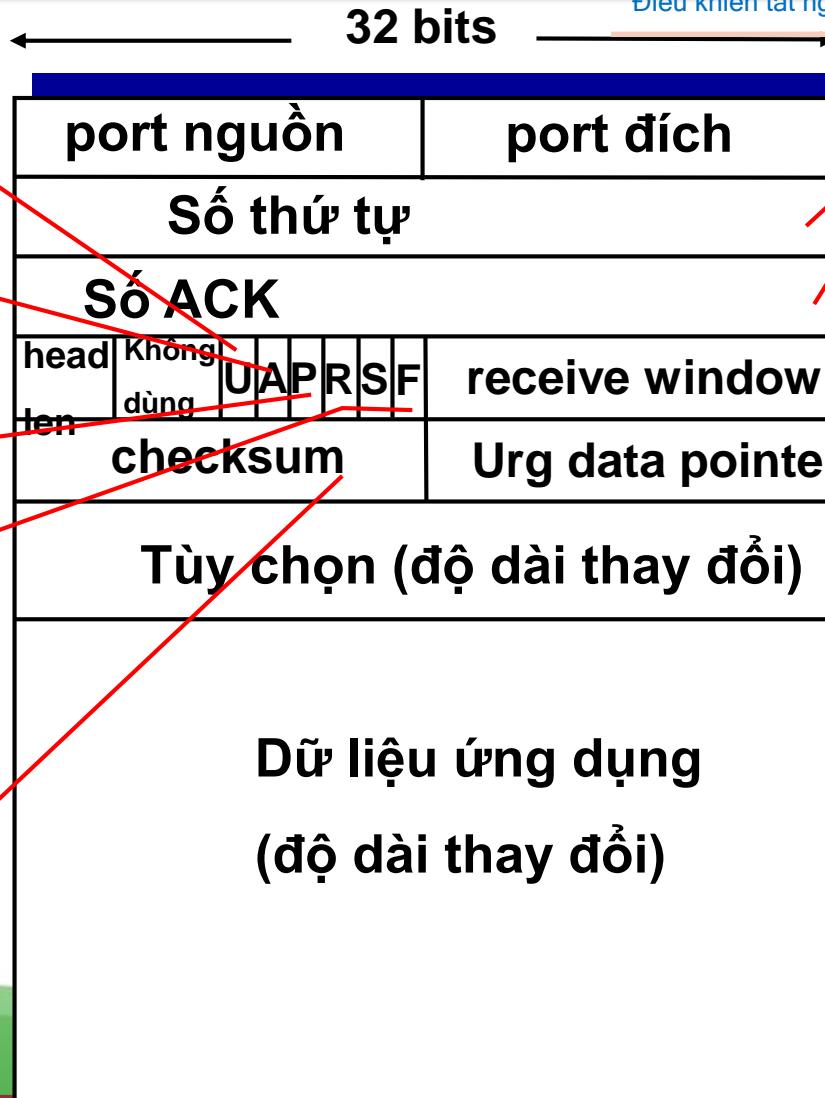
RST, SYN, FIN:

thiết lập kết nối  
(setup, teardown)

commands)  
Internet

checksum

(giống như UDP)



Đếm bằng bytes dữ liệu (không bằng segment!) Số byte bên nhận sẵn sàng chấp nhận



# Số thứ tự TCP và ACK

## Các số thứ tự:

- Dòng byte “đánh số” byte đầu tiên trong dữ liệu của segment

## Các ACK:

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

**Hỏi:** làm thế nào để bên nhận xử lý các segment không theo thứ tự

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

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

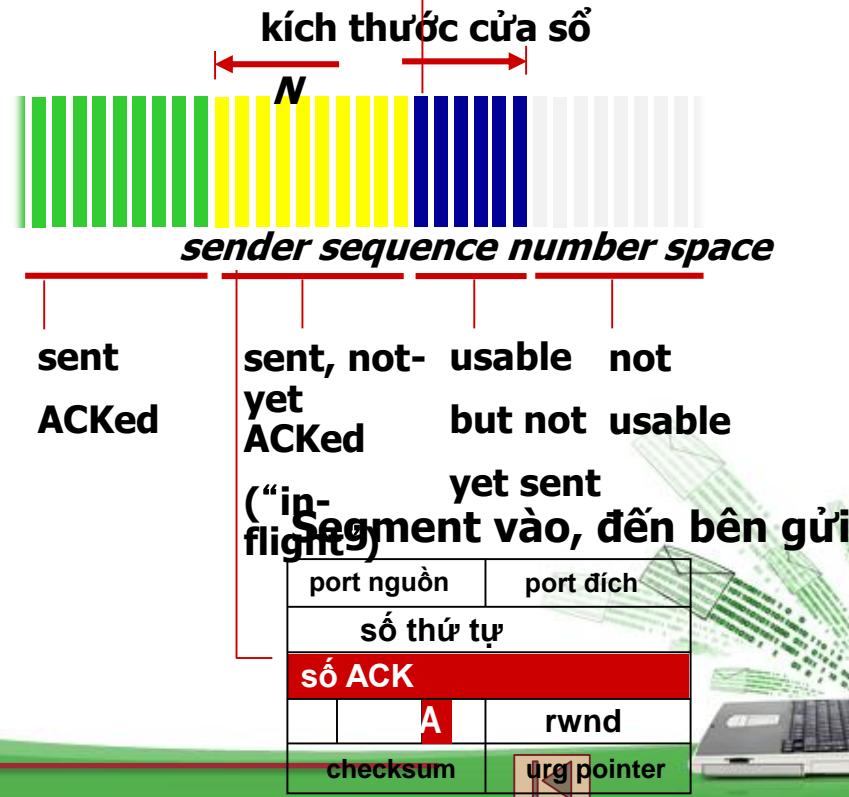
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

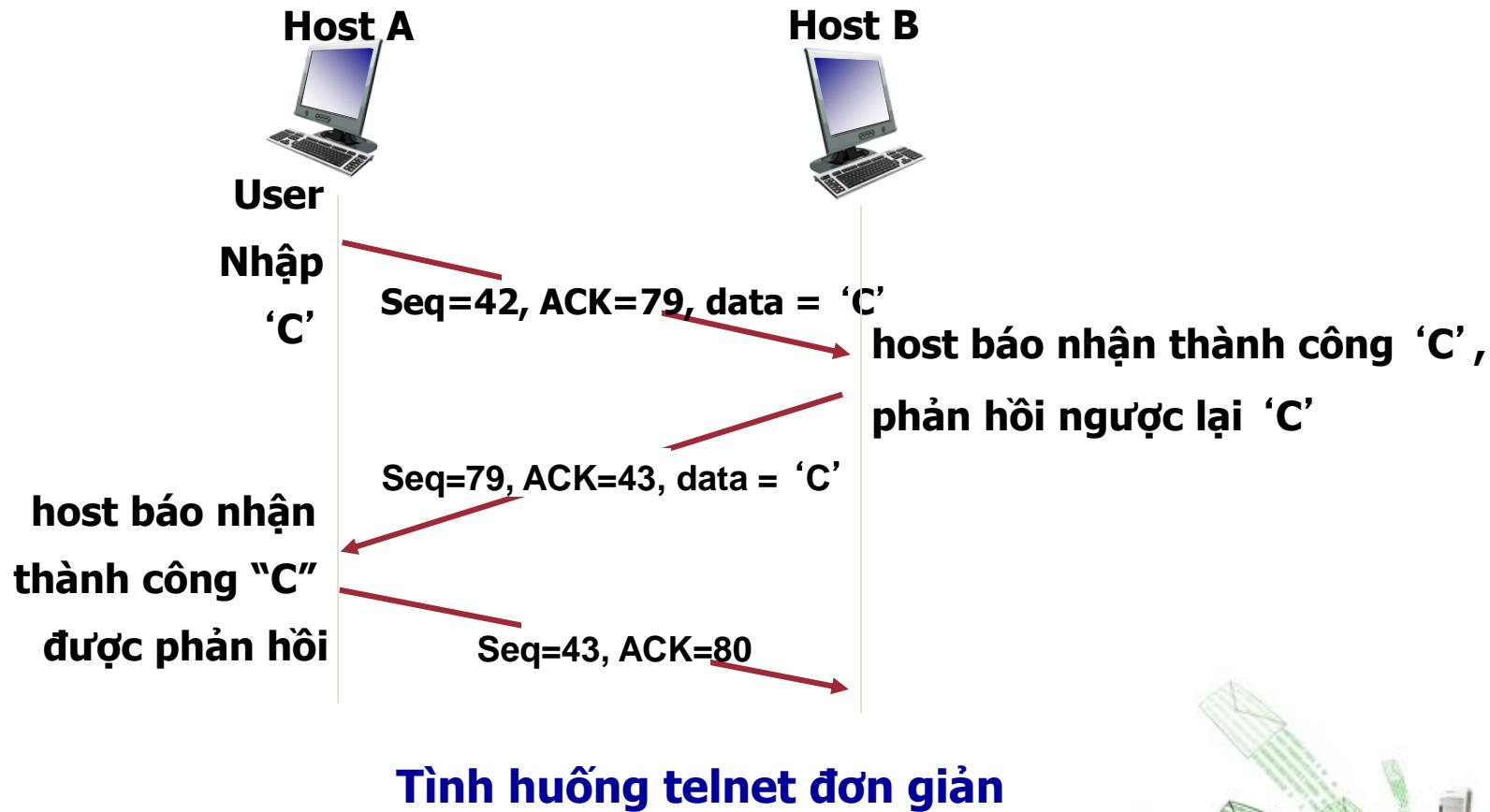
## Segment đi ra từ bên gửi

port nguồn	port đích
<b>số thứ tự</b>	
<b>số ACK</b>	
	rwnd
checksum	urg pointer



# Số thứ tự TCP và ACK

- Các dịch vụ tầng vận chuyển
- Vận chuyển phi kết nối: UDP
- Các nguyên lý truyền dữ liệu tin cậy
- Vận chuyển hướng kết nối: TCP
- Các nguyên lý điều khiển tắt nghẽn
- Điều khiển tắt nghẽn TCP



## TCP round trip time và timeout

Hỏi: làm cách nào để thiết lập giá trị TCP timeout?

- Dài hơn RTT
  - Nhưng RTT thay đổi
- Quá ngắn: timeout sớm, truyền lại không cần thiết
- Quá dài: phản ứng chậm đối với việc mất mât gói

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

Q: làm cách nào để ước lượng RTT?

- **SampleRTT:** thời gian được đo từ khi truyền segment đến khi báo nhận ACK
  - Lờ đi việc truyền lại
- **SampleRTT** sẽ thay đổi, muốn RTT được ước lượng “mượt hơn”
  - Đo lường trung bình của một số giá trị vừa xảy ra, không chỉ **SampleRTT** hiện tại



# TCP round trip time và timeout

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

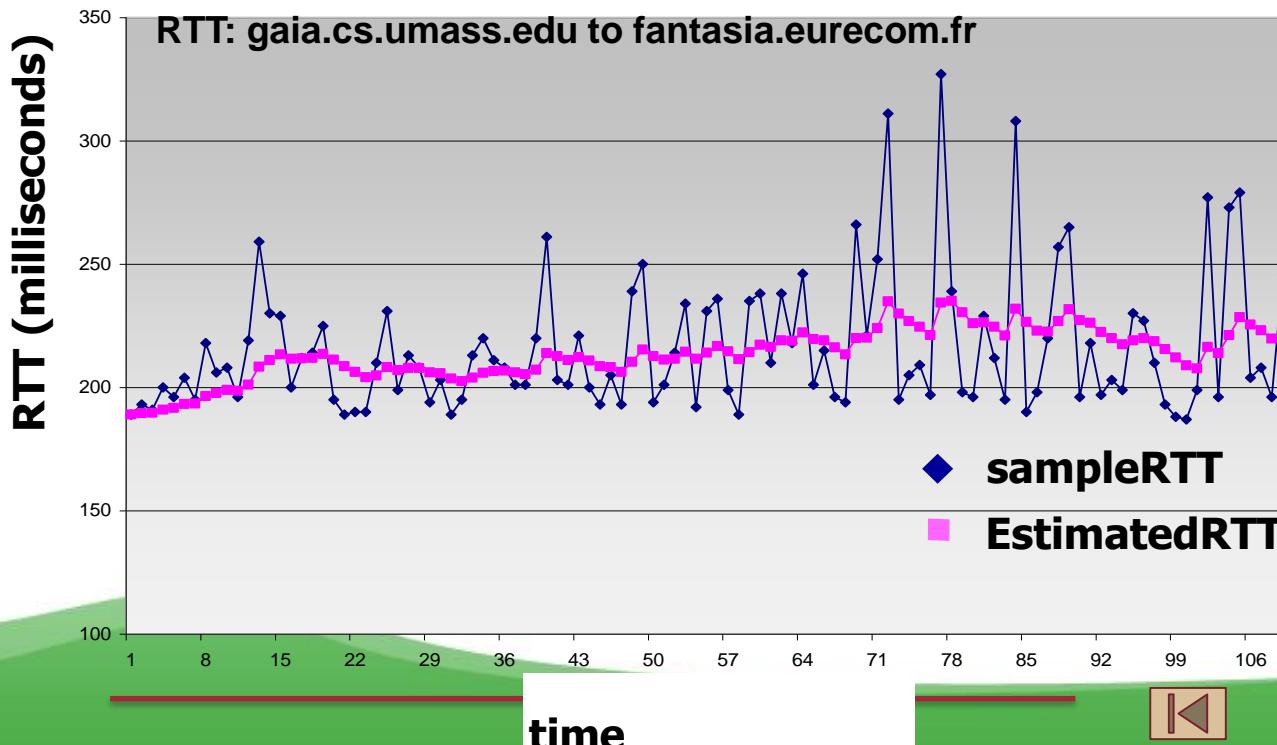
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

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

- ❖ Đường trung bình dịch chuyển hàm mũ (exponential weighted moving average)
- ❖ ảnh hưởng của mẫu đã xảy ra sẽ làm giảm tốc độ theo cấp số nhân
- ❖ typical value:  $\alpha = 0.125$



## TCP round trip time và timeout

- Khoảng thời gian timeout (timeout interval):  
**EstimatedRTT** cộng với “biên an toàn”
  - Sự thay đổi lớn trong **EstimatedRTT** -> an toàn biên lớn hơn
- Ước lượng độ lệch SampleRTT từ EstimatedRTT:

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

$$\beta * | \text{SampleRTT} - \text{EstimatedRTT} |$$

(typically,  $\beta = 0.25$ )

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



↑  
**estimated RTT**

↑  
“biên an toàn”



Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

## Chương 3 Nội dung

3.1 các dịch vụ tầng  
Vận chuyển

3.3 vận chuyển phi kết  
nối: UDP

3.4 các nguyên lý  
truyền dữ liệu tin  
cây

3.5 vận chuyển hướng  
kết nối: TCP

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

3.6 các nguyên lý về điều  
khiển tắc nghẽn

3.7 điều khiển tắc nghẽn  
TCP



# TCP truyền dữ liệu tin cậy

- TCP tạo dịch vụ rdt trên dịch vụ không tin cậy của IP
  - Các đoạn (*segment*) được truyền thông qua kiến trúc đường ống
  - Các ack tích lũy
  - TCP dùng một bộ đếm thời gian truyền lại
- Việc truyền lại được kích hoạt bởi:
  - Sự kiện timeout
  - Các ack bị trùng

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

Lúc đầu khảo sát TCP đơn giản ở bên gửi:

- Bỏ qua các ack bị trùng
- Bỏ qua điều khiển luồng và điều khiển tắc nghẽn



## TCP các sự kiện bên gửi:

Dữ liệu được nhận từ ứng dụng:

- Tạo segment với số thứ tự
- Số thứ tự là số thứ tự của byte dữ liệu đầu tiên trong segment
- Khởi động bộ đếm thời gian nếu chưa chạy
  - Xem bộ định thì như là đối với segment sớm nhất không được ACK
  - Khoảng thời gian hết hạn: TimeOutInterval

timeout:

- Gửi lại segment nào gây ra timeout
- Khởi động lại bộ đếm thời gian

nhận ack:

- Nếu xác nhận cho các segment không được xác nhận trước đó
  - Cập nhật những gì được biết là đã được nhận thành công
  - Khởi động lại bộ định thì nếu có các segment vẫn chưa được thông báo nhận thành công

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

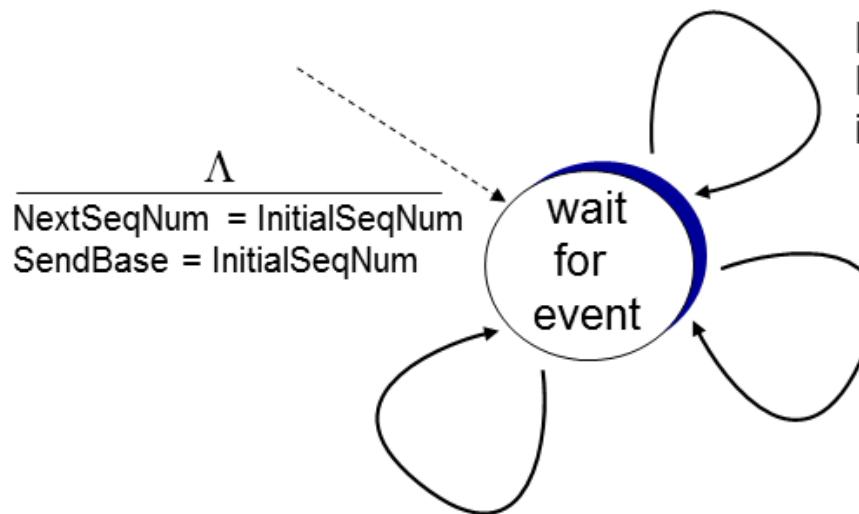
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



# TCP bên gửi (đơn giản)



$\Lambda$   
NextSeqNum = InitialSeqNum  
SendBase = InitialSeqNum

ACK received, with ACK field value y

```

if (y > SendBase) {
    SendBase = y
    /* SendBase-1: last cumulatively ACKed byte */
    if (there are currently not-yet-acked segments)
        start timer
    else stop timer
}
    
```

Dữ liệu được nhận từ tầng Ứng dụng trên  
create segment, seq. #: NextSeqNum  
pass segment to IP (i.e., “send”)  
NextSeqNum = NextSeqNum + length(data)  
if (bộ định thì hiện thời không chạy)  
    khởi động bộ định thì

timeout

Truyền lại segment nào chưa được  
báo đã nhận thành công với số  
thứ tự nhỏ nhất.  
Khởi động bộ định thì

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

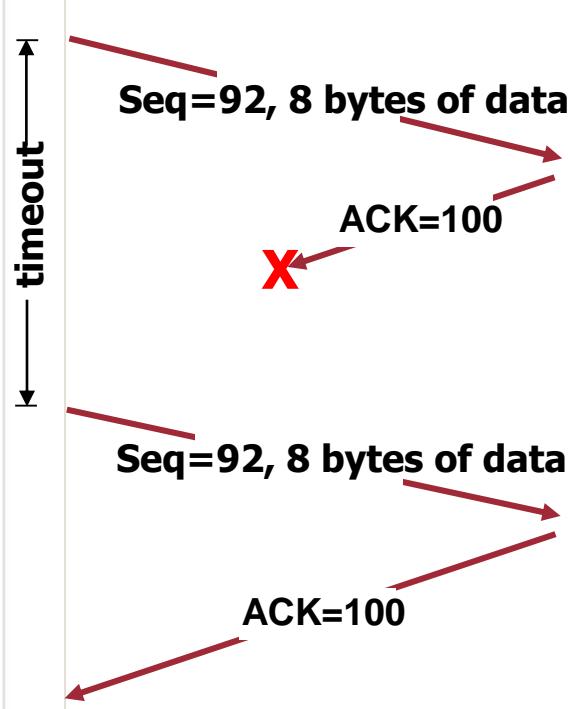
Điều khiển tắt nghẽn TCP

# TCP: tình huống truyền lại

Host A



Host B



Tình huống mất ACK

Host A



Host B



SendBase=92

SendBase=100

SendBase=120

Seq=92, 8 bytes of data

Seq=100, 20 bytes of data

ACK=100  
ACK=120

Seq=92, 8 bytes of data

ACK=120

↓

Timeout sớm



Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

# Sự phát sinh TCP ACK

## [RFC 1122, RFC 2581]

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

### Sự kiện tại bên nhận

### Hành động bên nhận TCP

segment đến theo thứ tự với số thứ tự được mong đợi. Tất cả dữ liệu đến đã được ACK

Hoãn gửi ACK. Đợi đến 500ms cho segment kế tiếp. Nếu không có segment kế tiếp, gửi ACK

segment đến theo thứ tự với số thứ tự mong muốn. 1 segment khác có ACK đang treo

Lập tức gởi lại một ACK tích lũy, thông báo nhận thành công cho cả segment theo thứ tự

Segment đến không theo thứ tự với số thứ tự lớn hơn số được mong đợi. Có khoảng trống

Lập tức gởi lại ACK trùng, chỉ ra số thứ tự của byte được mong đợi kế tiếp

segment đến lắp đầy từng phần hoặc toàn bộ khoảng trống

Lập tức gởi ACK, với điều kiện là segment đó bắt đầu ngay điểm có khoảng trống



## TCP truyền lại nhanh

- Chu kỳ time-out thường tương đối dài:
  - Độ trễ dài trước khi gửi lại gói bị mất
- Phát hiện các segment bị mất thông qua các ACKs trùng.
  - Bên gửi thường gửi nhiều segment song song
  - Nếu segment bị mất, thì sẽ có khả năng có nhiều ACK trùng.

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

### TCP truyền lại nhanh

Nếu bên gửi nhận 3 ACK của cùng 1 dữ liệu ("3 ACK trùng"), thì gửi lại segment chưa được ACK với số thứ tự nhỏ nhất

- Có khả năng segment không được ACK đã bị mất, vì thế không đợi đến thời gian timeout



# TCP truyền lại nhanh

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

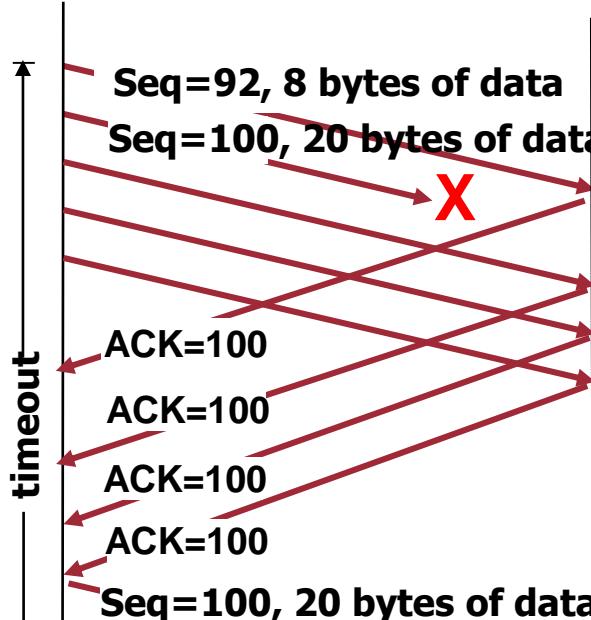
Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

Host A                          Host B



Truyền lại nhanh sau khi

bên gửi nhận 3 lần ACK bị trùng



## Chương 3 Nội dung

3.1 các dịch vụ tầng  
Vận chuyển

3.2 3.3 vận chuyển phi  
kết nối: UDP

3.4 các nguyên lý  
truyền dữ liệu tin  
cây

3.5 vận chuyển hướng  
kết nối: TCP

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

3.6 các nguyên lý về điều  
khiển tắc nghẽn

3.7 điều khiển tắc nghẽn  
TCP



# TCP điều khiển luồng

Ứng dụng có thể loại bỏ dữ liệu từ các bộ nhớ đệm socket TCP ....

... chậm hơn TCP  
bên nhận đang cung cấp  
(bên gửi đang gửi)

## Điều khiển luồng

bên nhận kiểm soát bên gửi, để bên gửi sẽ không làm tràn bộ nhớ đệm của bên nhận bởi truyền quá nhiều và quá nhanh

Các dịch vụ tầng vận chuyển

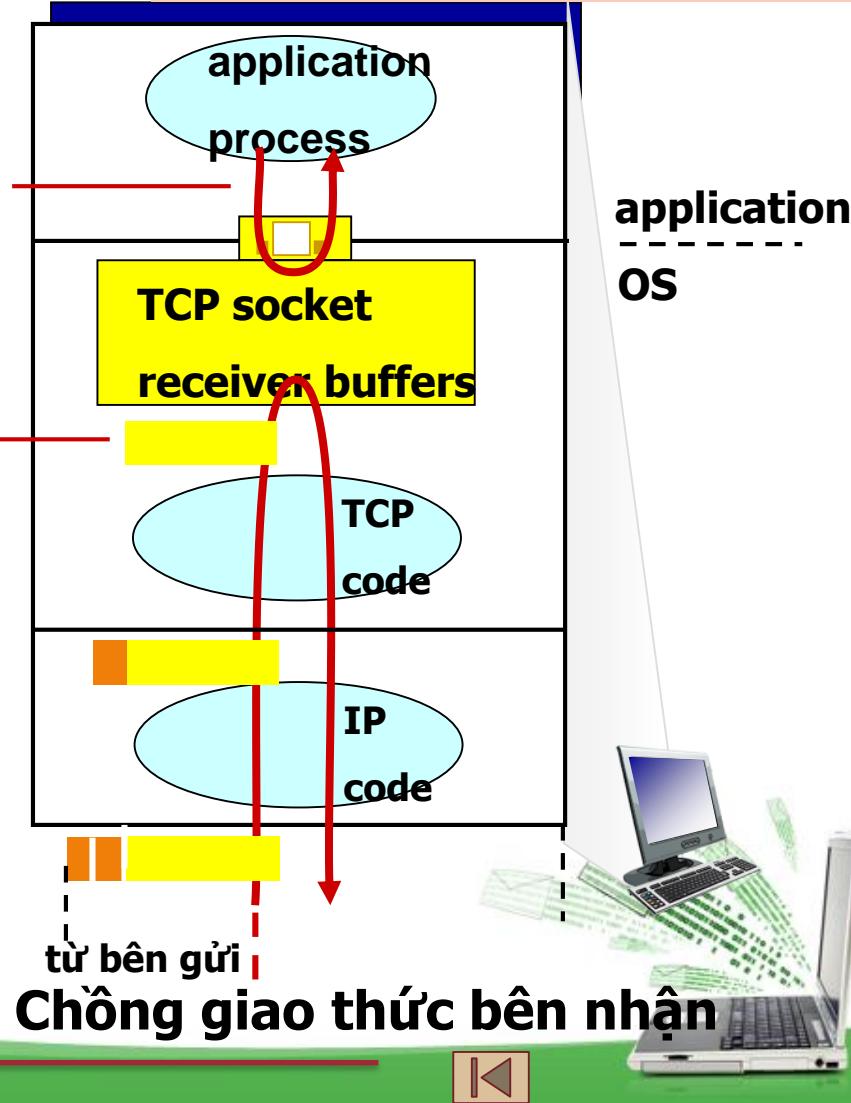
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

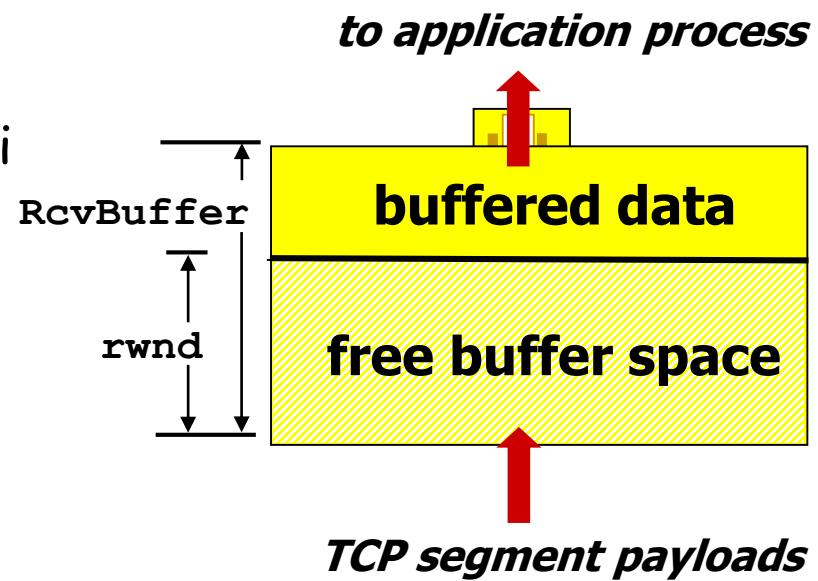
Điều khiển tắt nghẽn TCP



# TCP điều khiển luồng

- Bên nhận “thông báo” không gian bộ nhớ đệm còn trống bằng cách thêm giá trị **rwnd** trong TCP header của các segment từ bên nhận đến bên gửi
  - Kích thước của **RcvBuffer** được thiết đặt thông qua các tùy chọn của socket (thông thường mặc định là 4096 byte)
  - Nhiều hệ điều hành tự động điều chỉnh **RcvBuffer**
- Bên gửi giới hạn khối lượng dữ liệu gửi mà không cần ACK bằng giá trị **rwnd** của bên nhận
- Bảo đảm bộ đệm bên nhận sẽ không bị tràn

- Các dịch vụ tầng vận chuyển  
Vận chuyển phi kết nối: UDP  
Các nguyên lý truyền dữ liệu tin cậy  
Vận chuyển hướng kết nối: TCP  
Các nguyên lý điều khiển tắt nghẽn  
Điều khiển tắt nghẽn TCP



**Bộ đếm phía bên nhận**



## Chương 3 Nội dung

3.1 các dịch vụ tầng  
Vận chuyển

3.3 vận chuyển phi kết  
nối: UDP

3.4 các nguyên lý  
truyền dữ liệu tin  
cây

3.5 vận chuyển hướng  
kết nối: TCP

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

3.6 các nguyên lý về điều  
khiển tắc nghẽn

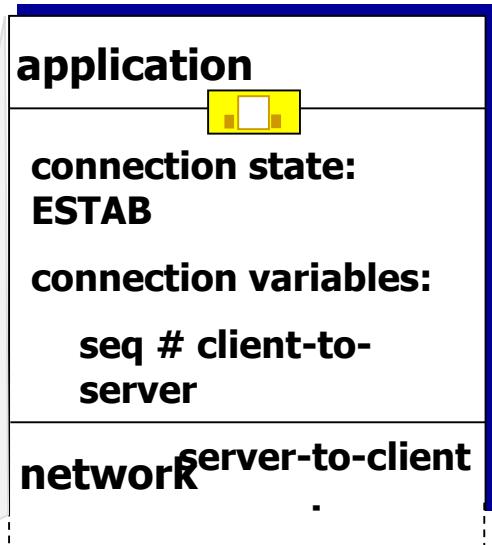
3.7 điều khiển tắc nghẽn  
TCP



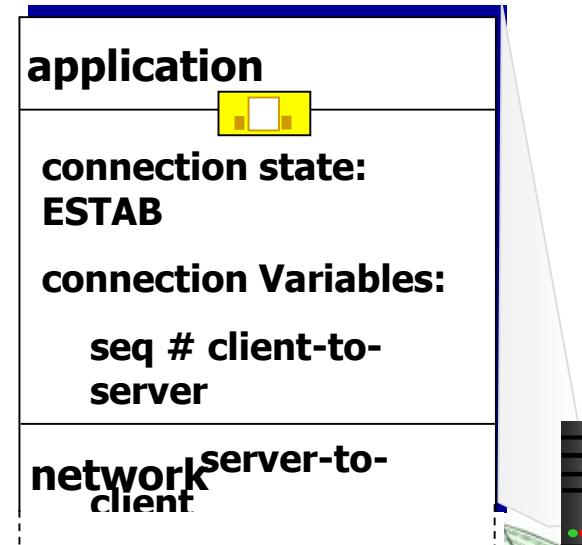
# Quản lý kết nối (Connection Management)

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

- Đồng ý thiết lập kết nối (mỗi bên biết bên kia sẵn sàng để thiết lập kết nối)
- Đồng ý các thông số kết nối



**at server,client**  
Socket clientSocket =  
newSocket("hostname", "port  
number");



**at server,client**  
Socket welcomeSocket =  
welcomeSocket.accept();

- Các dịch vụ tầng vận chuyển
- Vận chuyển phi kết nối: UDP
- Các nguyên lý truyền dữ liệu tin cậy
- Vận chuyển hướng kết nối: TCP
- Các nguyên lý điều khiển tắt nghẽn
- Điều khiển tắt nghẽn TCP



# TCP bắt tay 3 lần (3-way handshake)

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

## Trạng thái client

**LISTEN**

Chọn số thứ tự ban đầu, x

**SYNSENT**

Gửi TCP SYN msg

**ESTAB**

SYNACK(x) vừa được nhận

cho hay server vẫn còn sống;

send ACK for SYNACK;

Gói tin này có thể chứa

dữ liệu client gửi server



## Trạng thái server

**LISTEN**



**SYNbit=1, Seq=x**

Chọn số thứ tự ban đầu, y  
gửi TCP SYN ACK  
msg, xác nhận cho SYN

**SYNbit=1, Seq=y**

**ACKbit=1; ACKnum=x+1**

**ACKbit=1, ACKnum=y+1**

ACK(y) vừa được nhận  
cho hay client vẫn sống

**ESTAB**



# TCP bắt tay 3 lần (3-way handshake)

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

*Trạng thái client*



*Trạng thái server*



**SYN( Seq=1,Ack=0)**

**ACK(Seq=10,Ack=2)**

**ACK(Seq=2, ACK=10)**



# TCP bắt tay 3 lần: FSM

Các dịch vụ tầng vận chuyển

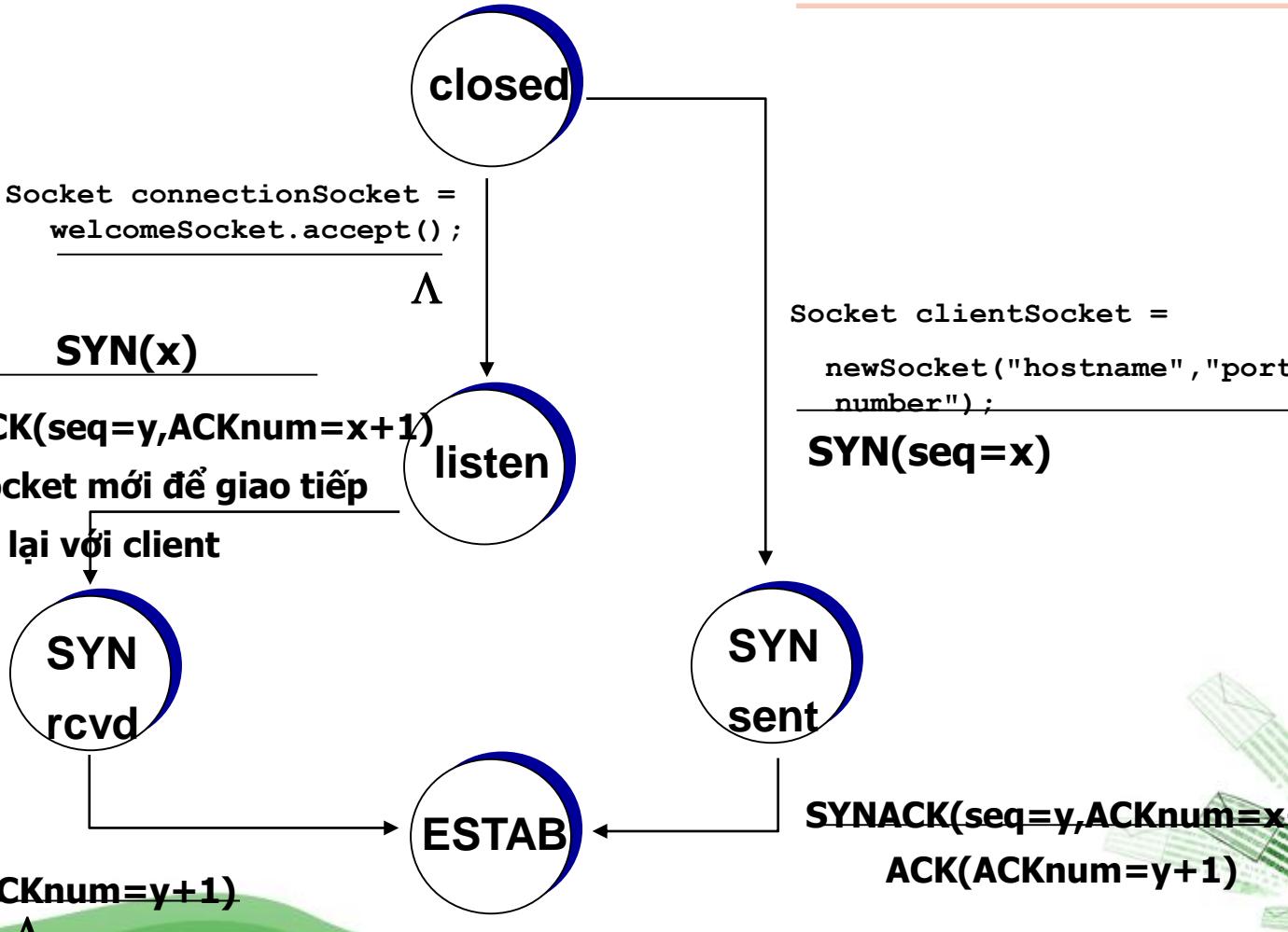
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



# TCP: đóng kết nối

- Mỗi bên client và server sẽ đóng kết nối bên phía của nó
  - Gởi TCP segment với FIN bit = 1
- Phản hồi bằng ACK cho FIN vừa được nhận
  - Khi nhận FIN, ACK có thể được kết hợp với FIN của nó
- Các trao đổi FIN đồng thời có thể được sử dụng

Các dịch vụ tầng vận chuyển

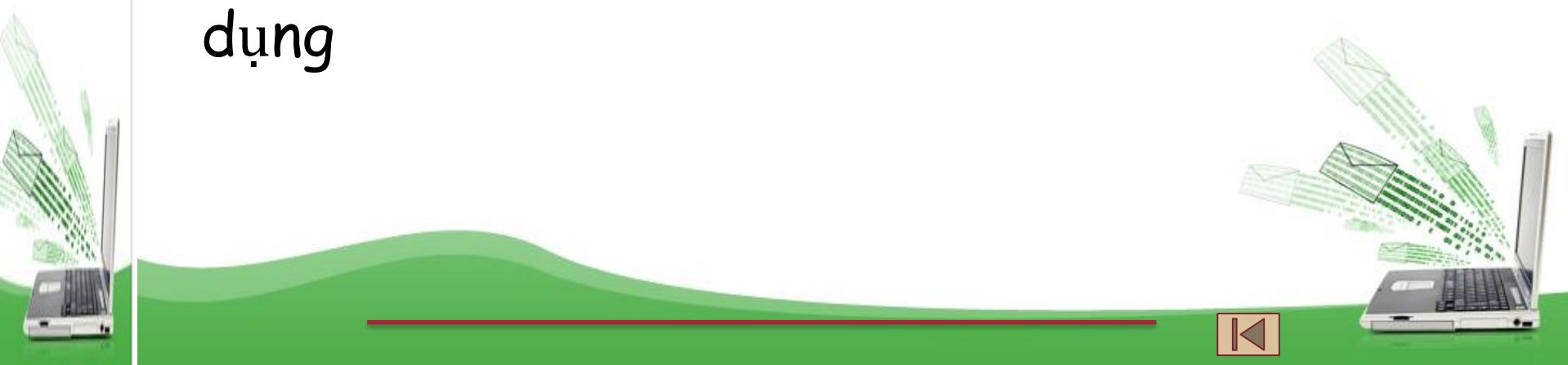
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



# TCP: đóng kết nối

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

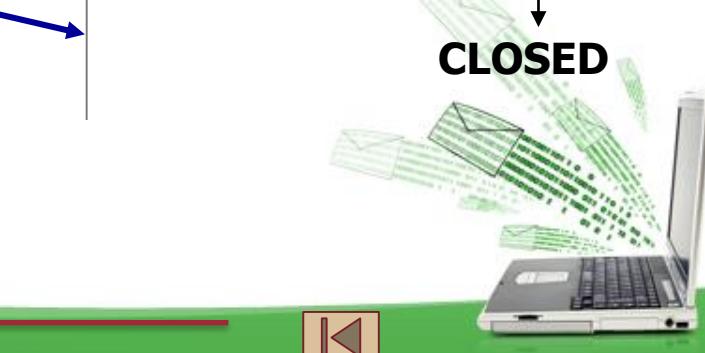
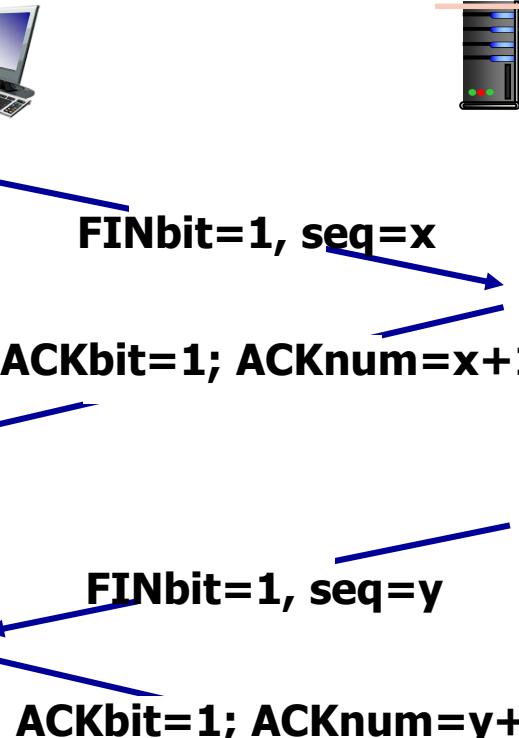
Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

## Trạng thái client



## Trạng thái server



# Các nguyên lý điều khiển tắc nghẽn (congestion control)

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

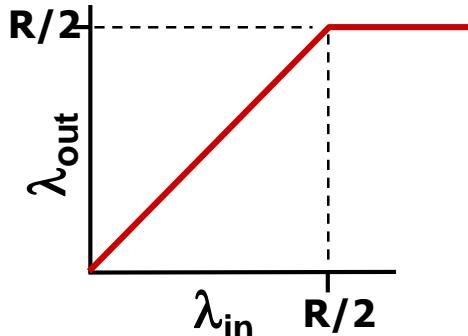
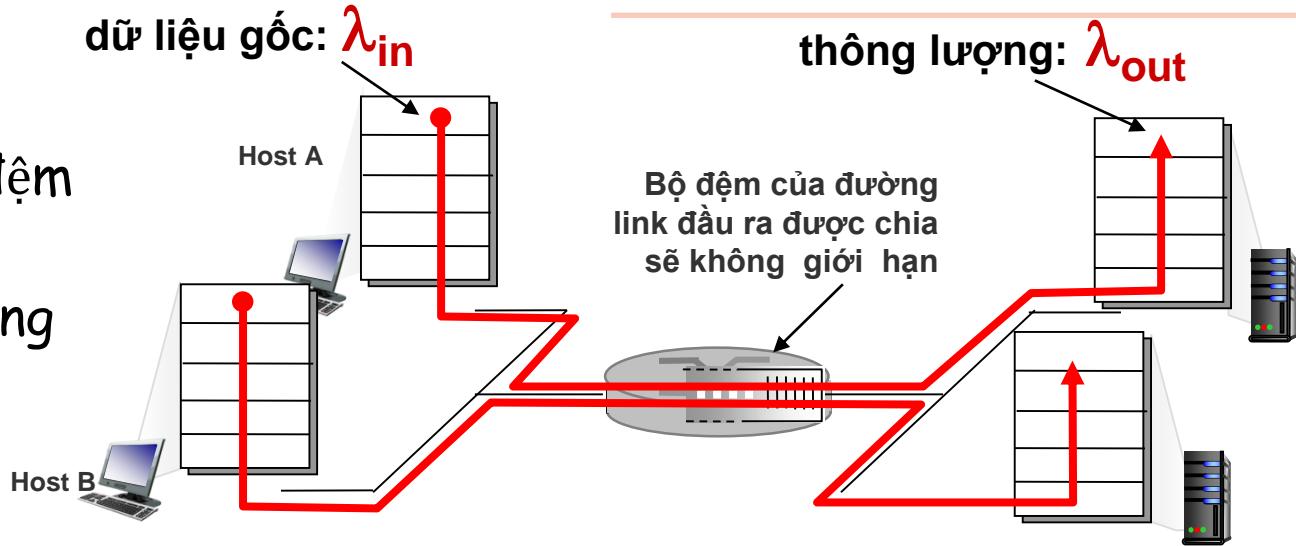
## Tắc nghẽn:

- “quá trình nguồn gửi quá nhiều dữ liệu với tốc độ quá nhanh vượt quá khả năng xử lý của *mạng*”
- Khác với điều khiển luồng (flow control)!
- Các biểu hiện:
  - Mất gói (tràn bộ đệm tại các router)
  - Độ trễ lớn (xếp hàng trong các bộ đệm của router)
- 1 trong 10 vấn đề khó khăn!

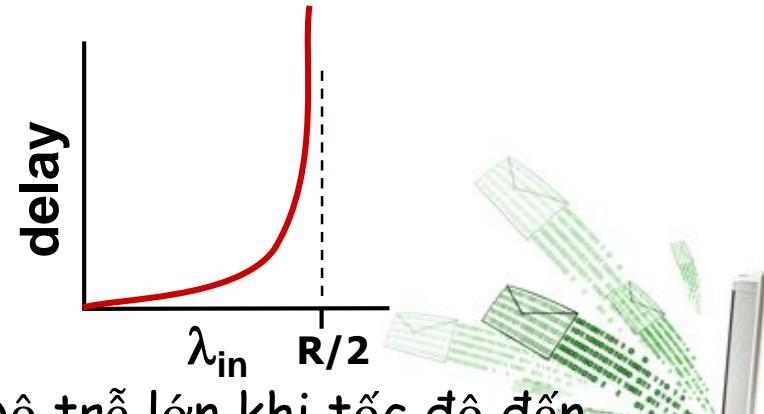


# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 1

- 2 gửi, 2 nhận
- 1 router, các bộ đệm không giới hạn
- Khả năng của đường link đầu ra:  $R$
- Không truyền lại



- Thông lượng lớn nhất của mỗi kết nối:  $R/2$



- ❖ Độ trễ lớn khi tốc độ đến,  $\lambda_{in}$ , vượt quá capacity

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

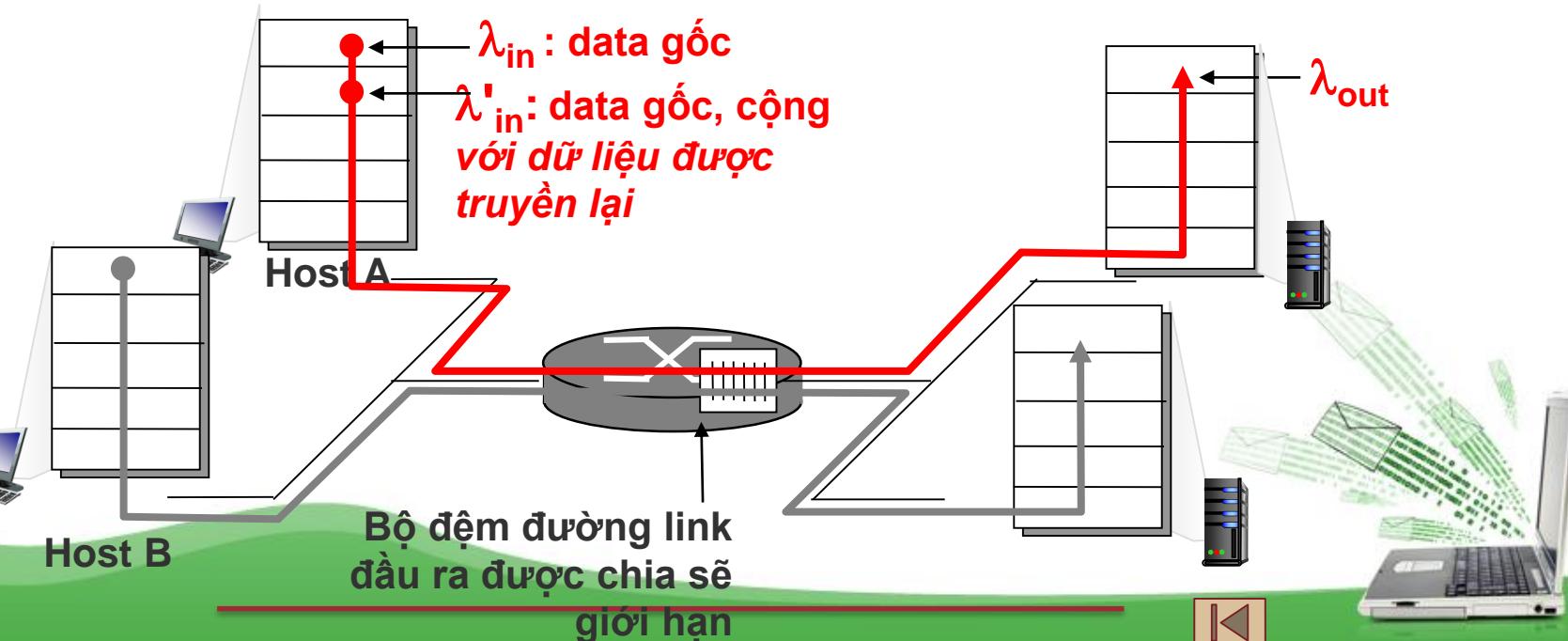
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 2

- 1 router, các bộ đệm có giới hạn
- Bên gửi truyền lại các packet bị hết thời gian chờ
  - input tầng Ứng dụng = output tầng Ứng dụng:  $I_{in} \leq I_{out}$
  - input tầng Vận chuyển bao gồm việc *truyền lại*:  $I_{in} > I_{in}$



Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

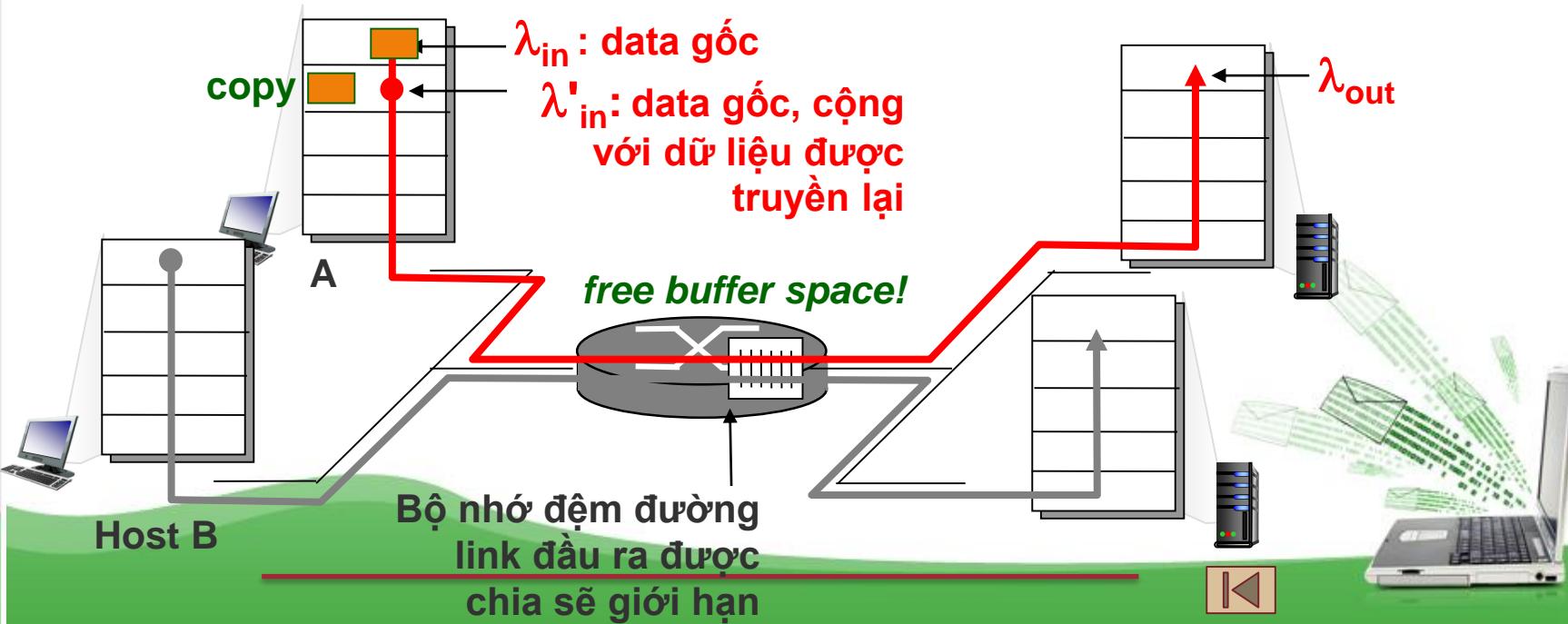
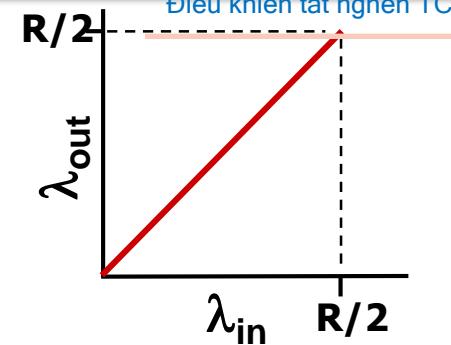
Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 2

Lý tưởng hóa: kiến thức hoàn hảo

- Bên gửi chỉ gửi khi bộ nhớ đệm của router sẵn sàng



Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

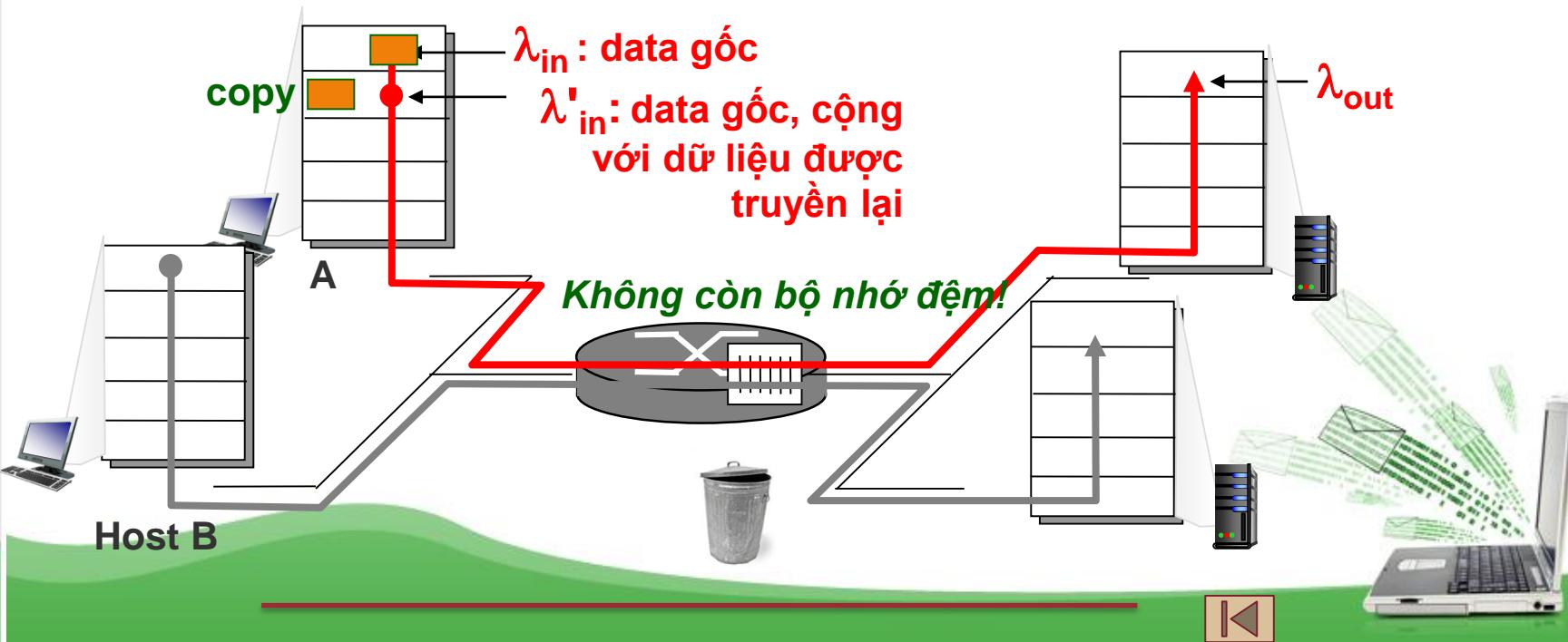
Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 2

Lý tưởng hóa: các packet có thể bị mất hoặc bị loại bỏ tại router bởi vì bộ nhớ đệm bị đầy

- Bên gửi chỉ gởi lại các gói đã bị mất



Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

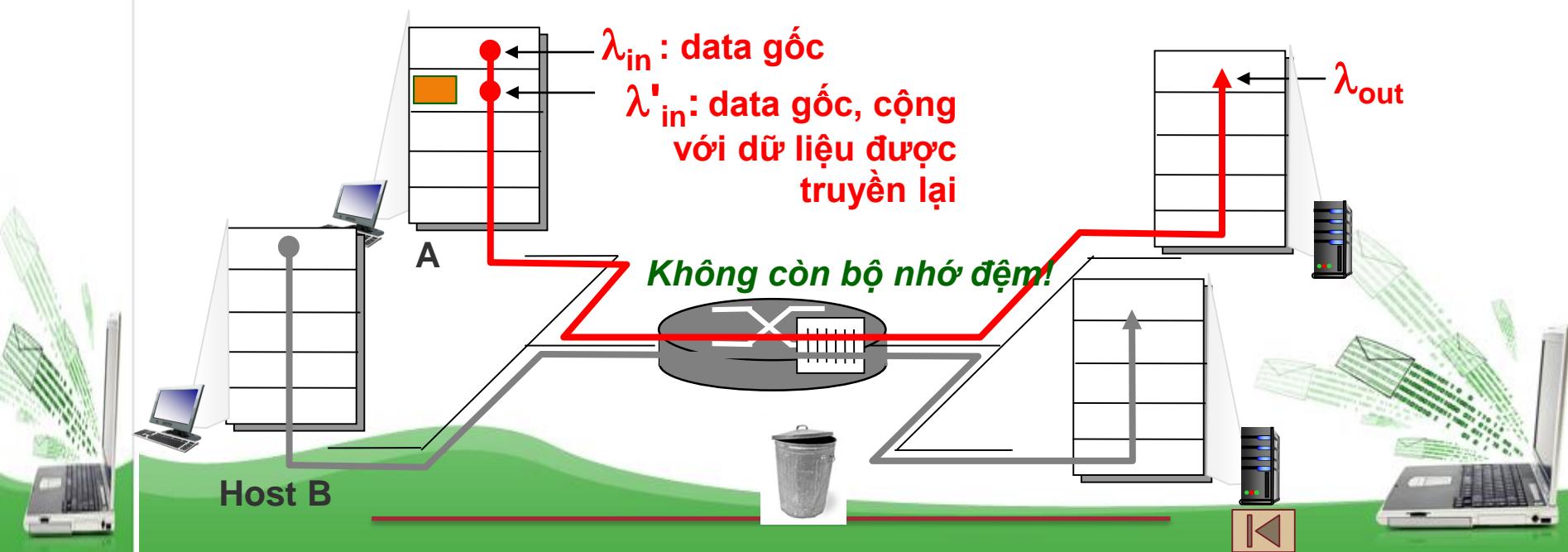
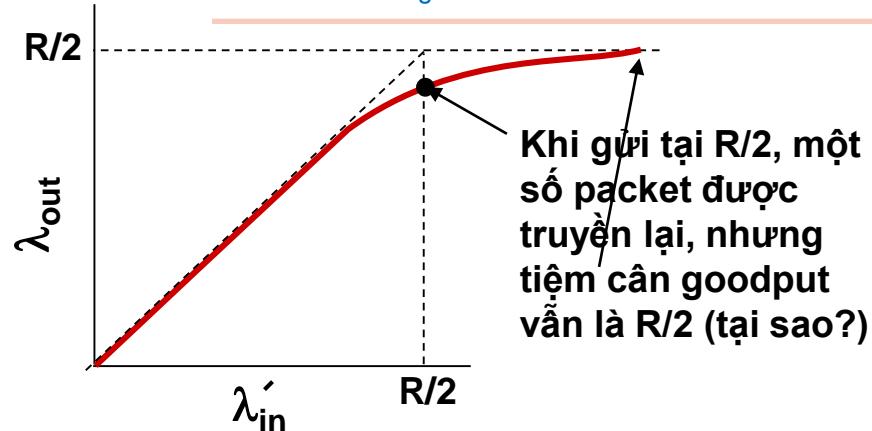
Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 2

**Lý tưởng hóa:** các packet có thể bị mất hoặc bị loại bỏ tại router bởi vì bộ nhớ đệm bị đầy

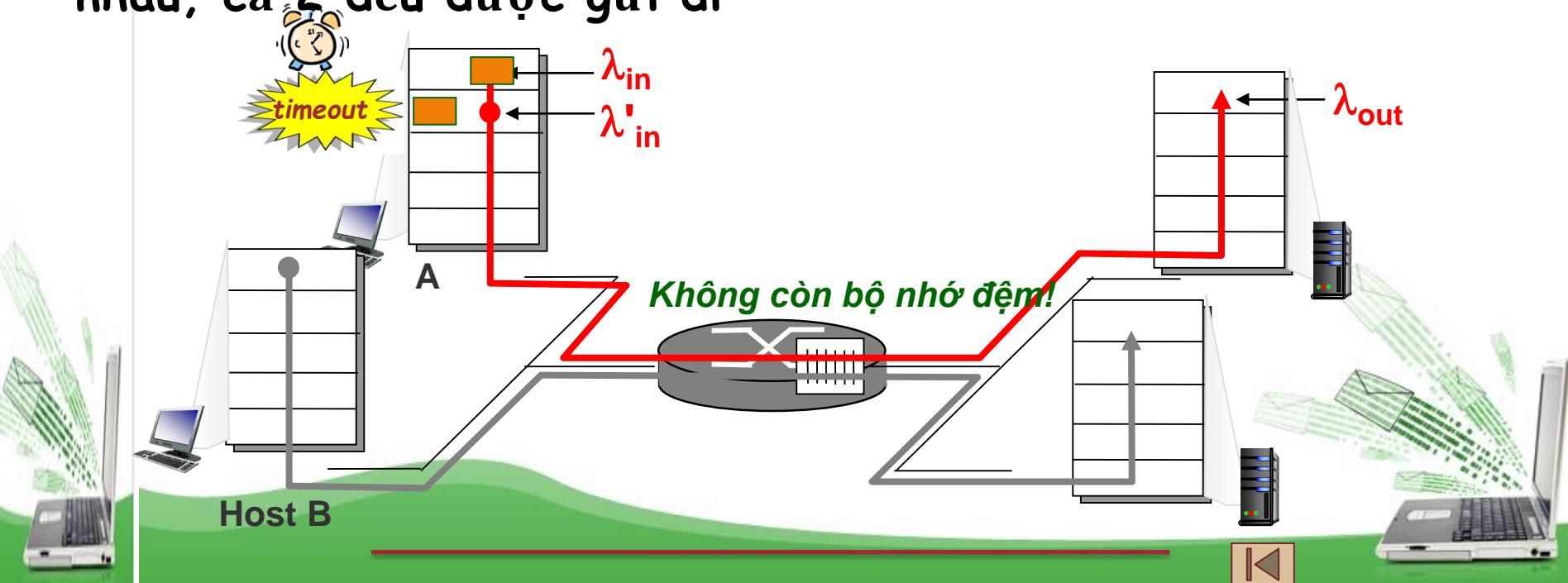
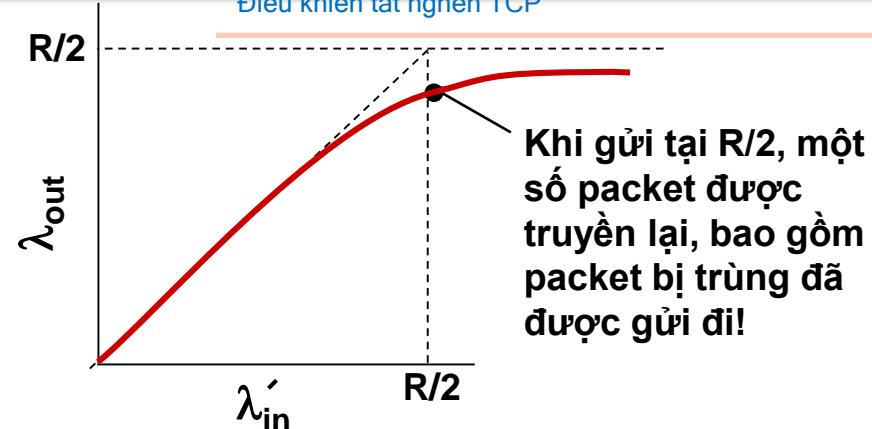
- Bên gửi chỉ gởi lại các gói đã bị mất



# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 2

## Thực tế: trùng lắp

- ❖ Các packet có thể bị mất, bị bỏ tại router bởi vì bộ nhớ đệm đầy
- ❖ Thời gian time out bên gửi hết sớm, gửi 2 bản giống nhau, cả 2 đều được gửi đi



Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

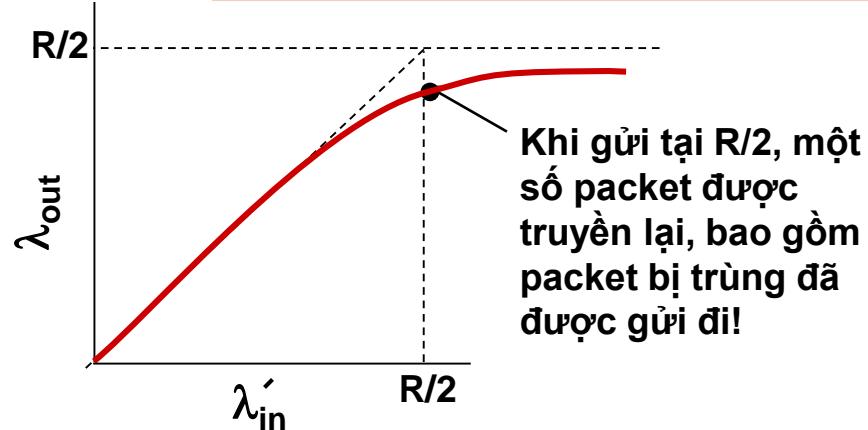
Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 2

## Thực tế: trùng lắp

- ❖ Các packet có thể bị mất, bị bỏ tại router bởi vì bộ nhớ đệm đầy
- ❖ Thời gian time out bên gửi hết sớm, gửi 2 bản giống nhau, cả 2 đều được gửi đi



## “chi phí” của tắc nghẽn:

- ❖ Nhiều việc hơn (truyền lại) cho “goodput”
- ❖ Truyền lại không cần thiết: đường truyền mang nhiều bản sao của gói
  - Giảm goodput

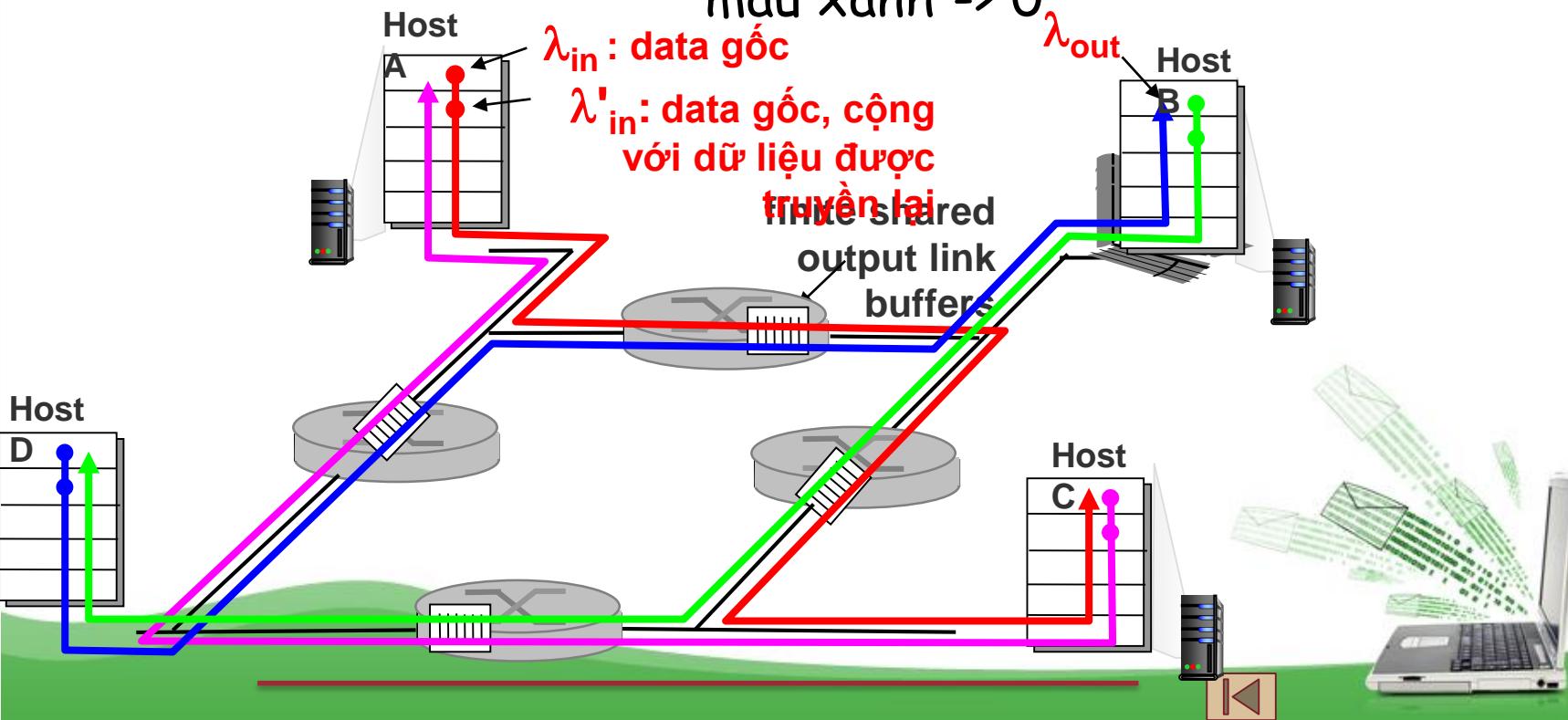


# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 3

- 4 người gởi
  - Các đường qua nhều hop
  - timeout/truyền lại

Hỏi: cái gì xảy ra khi  $|z_1|$  và  $|z_2|$  tăng?

TL: khi  $l_{in}$  màu đỏ tăng, tất cả packet màu xanh đến tại hàng đợi phía trên bị loại bỏ, thông lượng màu xanh  $\rightarrow 0$



# Nguyên nhân/Chi phí của tắc nghẽn: tình huống 3

Các dịch vụ tầng vận chuyển

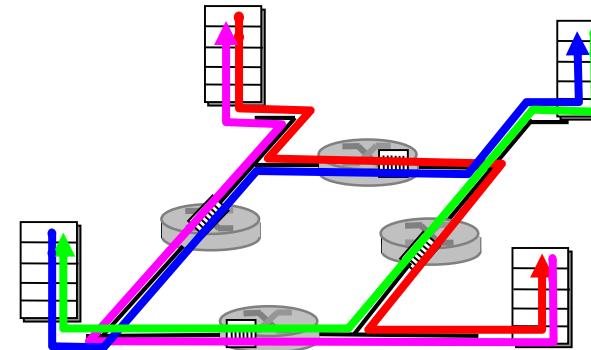
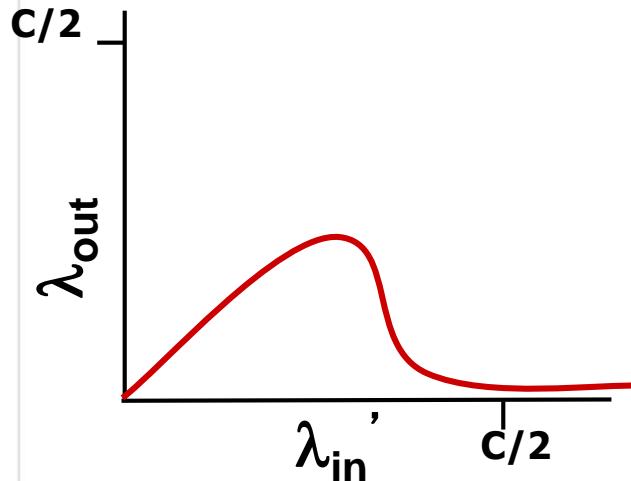
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP



## "Chi phí" khác của tắc nghẽn

- ❖ Khi gói bị loại bỏ, bất kỳ "lưu lượng tải lên (upstream) dùng cho gói đó sẽ bị lãng phí!"



# Các phương pháp tiếp cận đối với điều khiển tắc nghẽn

## 2 phương pháp tiếp cận:

### Điều khiển tắc nghẽn end-end :

- Không có phản hồi rõ ràng từ mạng
- Tắc nghẽn được suy ra từ việc quan sát hệ thống đầu cuối có mất mát hoặc bị trễ
- TCP được giao nhiệm vụ xử lý tắc nghẽn

### Điều khiển tắc nghẽn có sự hỗ trợ của mạng (network-assisted) :

- Các router cung cấp phản hồi đến các hệ thống đầu cuối
  - Bit đơn chỉ ra tắc nghẽn (SNA, DECbit, TCP/IP ECN, ATM)
  - Tốc độ sẽ gửi của người gửi được xác định rõ ràng

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



## Ví dụ: điều khiển tắc nghẽn ATM ABR

ABR: available bit rate:

- "dịch vụ mềm dẻo"
- Nếu đường gửi "dưới tải":
  - Bên gửi sẽ dùng băng thông trống
- Nếu đường gửi bị tắc nghẽn:
  - Bên gửi sẽ điều tiết với tốc độ tối thiểu được bảo đảm

Các gói RM (resource management cell):

- Được gửi bởi bên gửi, được xen kẽ với các gói dữ liệu
- Các bit trong RM cell được thiết lập bởi các switch
  - *NI bit*: không tăng tốc độ (tắc nghẽn nhẹ)
  - *CI bit*: tắc nghẽn rõ rệt
- Các RM cell được trả về bên gửi từ bên nhận với nguyên vẹn các bit trên

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

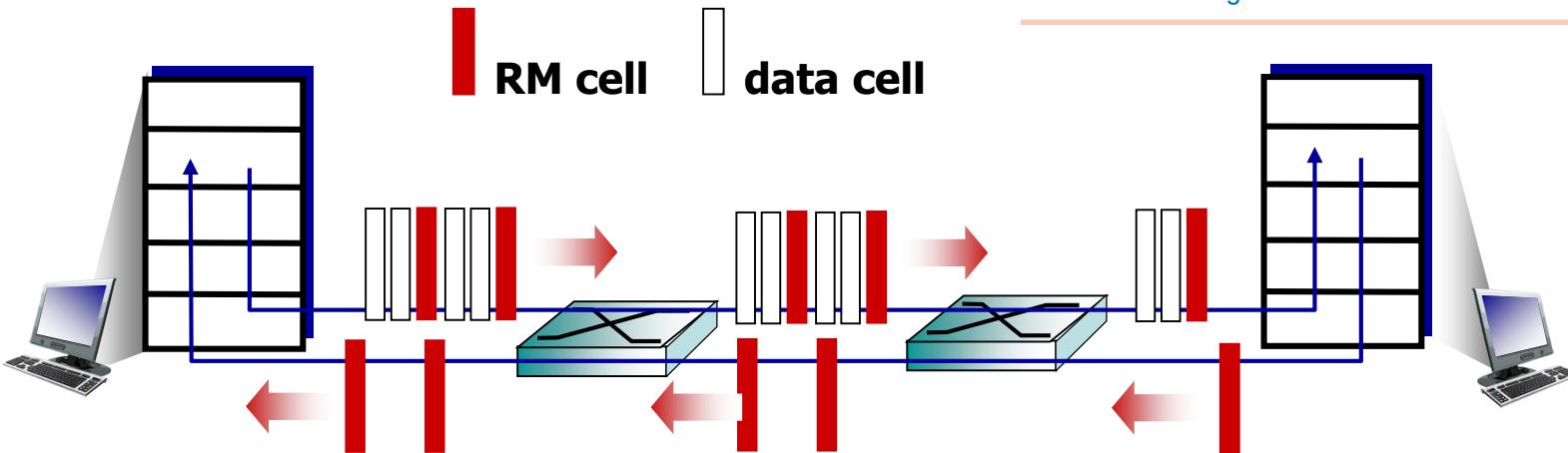
Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP



## Ví dụ: điều khiển tắc nghẽn ATM ABR

- Các dịch vụ tầng vận chuyển
- Vận chuyển phi kết nối: UDP
- Các nguyên lý truyền dữ liệu tin cậy
- Vận chuyển hướng kết nối: TCP
- Các nguyên lý điều khiển tắc nghẽn
- Điều khiển tắc nghẽn TCP



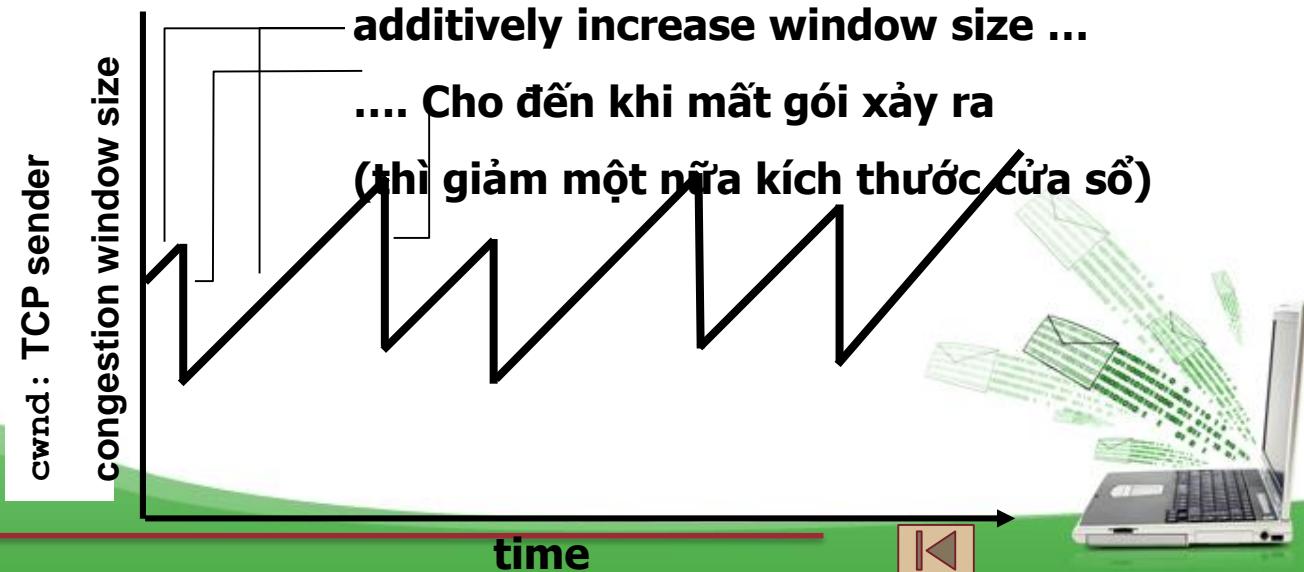
- Trường 2 byte ER (tốc độ tường minh) trong cell RM
  - Switch bị tắc nghẽn có thể giảm giá trị ER trong gói
  - tốc độ gửi do đó có thể được điều tiết cho phù hợp với tốc độ tối đa mà đường truyền hỗ trợ
- Bit EFCI bit trong cell dữ liệu: được thiết lập là 1 tại switch bị tắc nghẽn
  - Nếu gói dữ liệu đứng trước RM cell có bit EFCI bật lên, bên gửi sẽ bật bit CI trong RM cell trả về



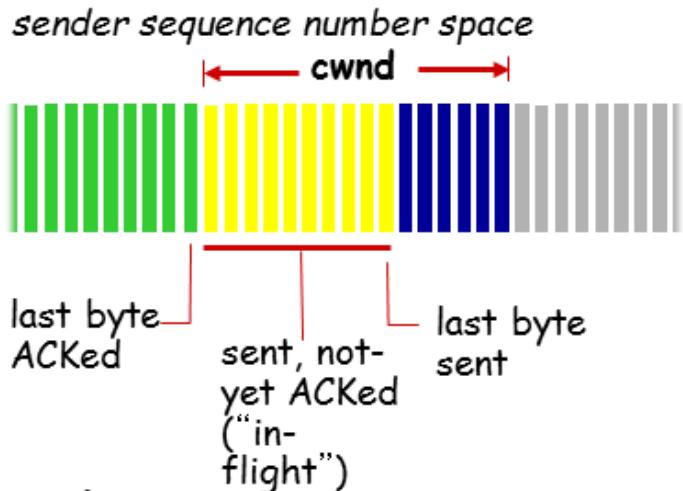
# TCP điều khiển tắc nghẽn: tăng theo cấp số cộng, giảm theo cấp số nhân

- ❖ **Hướng tiếp cận:** bên gửi tăng tốc độ truyền (kích thước cửa sổ), thăm dò băng thông có thể sử dụng, cho đến khi mất gói xảy ra
  - **tăng theo cấp số cộng (additive increase):** tăng cwnd (congestion window) lên 1 MSS sau mỗi RTT cho đến khi mất gói xảy ra
  - **giảm theo cấp số nhân (multiplicative decrease):** giảm một nửa cwnd sau khi mất gói xảy ra

**AIMD saw tooth behavior:** thăm dò băng thông



# TCP điều khiển tắc nghẽn: chi tiết



- ❖ Bên gửi giới hạn truyền tải:

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

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

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

## TCP tốc độ gửi:

- ❖ **Ước lượng**: khối lượng byte gửi (cwnd) đợi ACK trong khoảng thời gian RTT

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



## TCP Slow Start

- Khi kết nối bắt đầu, tăng tốc độ theo cấp số nhân cho đến sự kiện mất gói đầu tiên xảy ra:
  - initially  $cwnd = 1 MSS$
  - Gấp đôi  $cwnd$  mỗi RTT
  - Được thực hiện bằng cách tăng  $cwnd$  cho mỗi ACK nhận được
- Tóm lại: tốc độ ban đầu chậm, nhưng nó sẽ tăng lên theo cấp số nhân

Các dịch vụ tầng vận chuyển

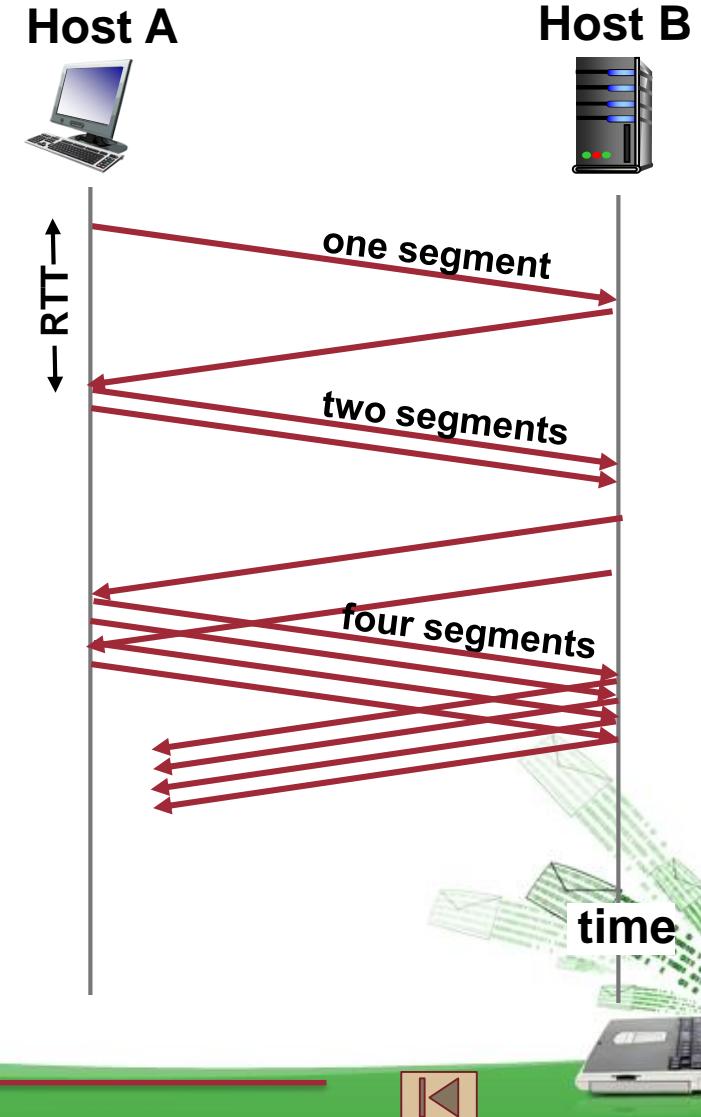
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



# TCP: phát hiện, phản ứng khi mất gói

- Mất gói được chỉ ra bởi timeout:
  - **cwnd** được thiết lập 1 MSS;
  - Sau đó kích thước cửa sổ sẽ tăng theo cấp số nhân (như trong slow start) đến ngưỡng, sau đó sẽ tăng tuyến tính
- Mất gói được xác định bởi 3 ACK trùng nhau: TCP RENO
  - Các ACK trùng lặp chỉ ra mạng vẫn có khả năng truyền
  - **cwnd** bị cắt một nửa sau đó tăng theo tuyến tính
- TCP TAHOE luôn luôn thiết lập **cwnd** bằng 1 (timeout hoặc 3 ack trùng nhau)



## TCP: chuyển từ slow start qua CA

**Hỏi:** khi nào tăng cấp lũy thừa nên chuyển qua tuyến tính?

**Trả lời:** khi **cwnd** được  $1/2$  giá trị của nó trước thời gian **timeout**.

### Thực hiện:

- **ssthresh** thay đổi
- Khi mất gói, **ssthresh** được thiết lập về chỉ  $1/2$  của **cwnd** trước khi mất gói

Các dịch vụ tầng vận chuyển

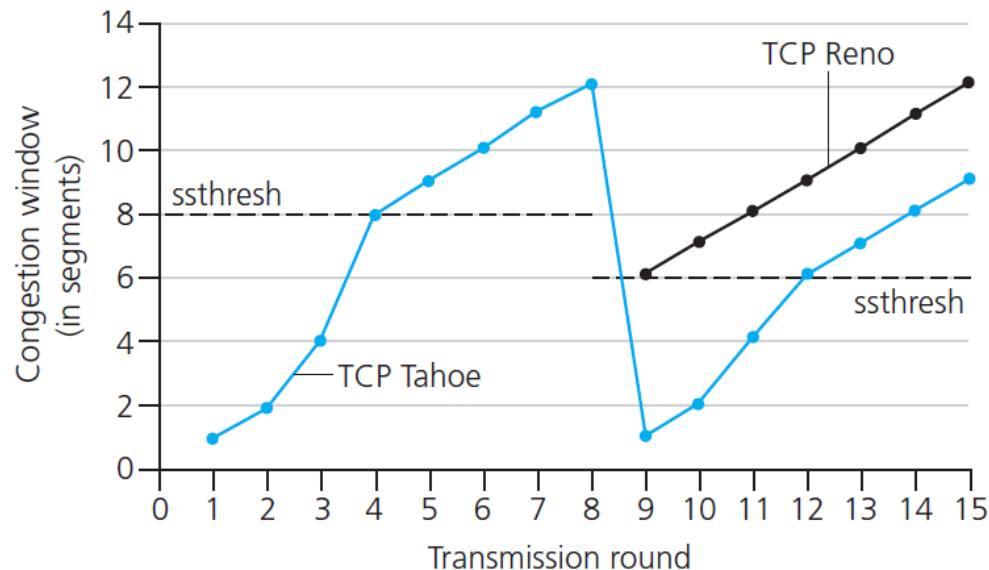
Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

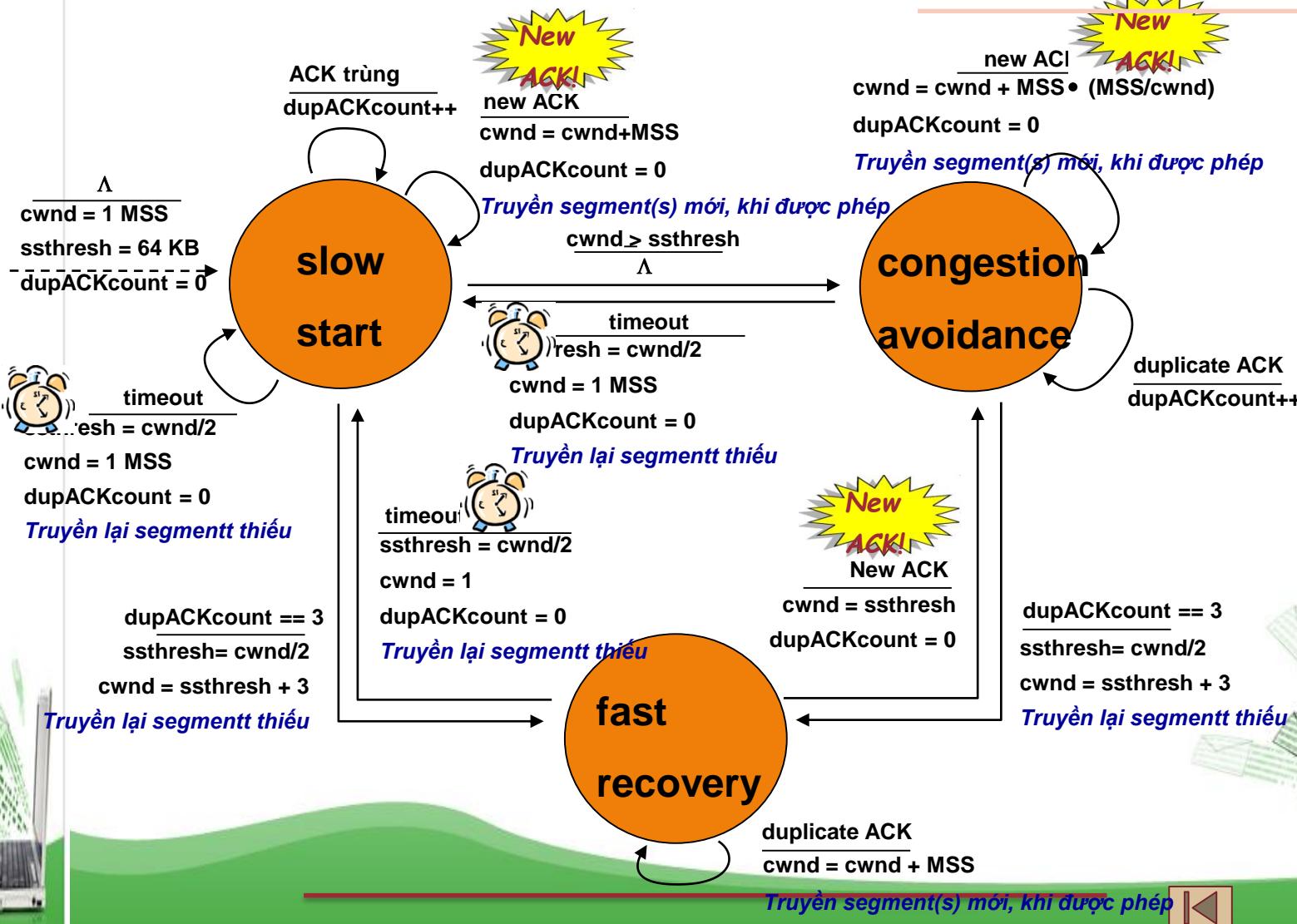
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP



# Tóm tắt: TCP điều khiển tắc nghẽn



Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắc nghẽn

Điều khiển tắc nghẽn TCP

New ACK!

$$\text{new ACK}$$

$$cwnd = cwnd + MSS$$

dupACKcount = 0

Truyền segment(s) mới, khi được phép

congestion avoidance

duplicate ACK

dupACKcount++

dupACKcount == 3

ssthresh= cwnd/2

cwnd = ssthresh + 3

Truyền lại segmentt thiếu

cwnd

= cwnd + MSS

Truyền segment(s) mới, khi được phép



## TCP thông lượng (throughput)

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

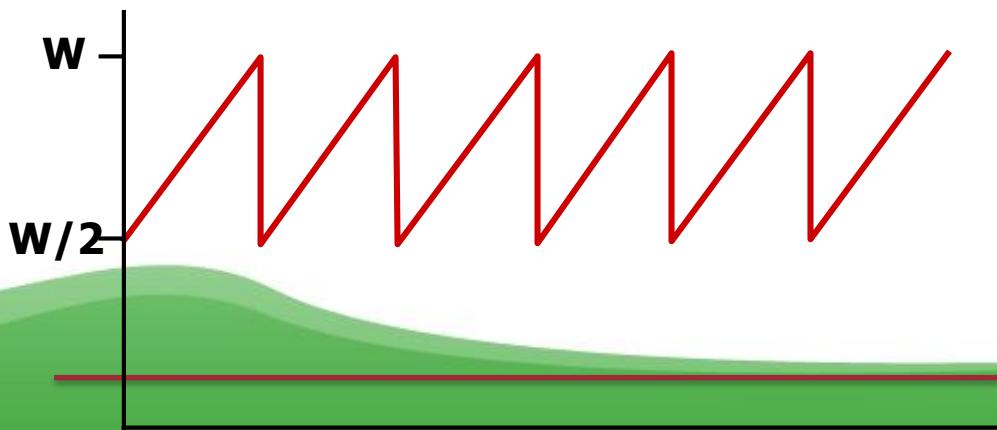
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

- Thông lượng trung bình của TCP như là chức năng của kích thước cửa sổ và RTT?
  - Bỏ qua slow start, giả sử dữ liệu luôn luôn được gởi
- W: kích thước cửa sổ (được đo bằng byte) khi mất gói xảy ra
  - Kích thước cửa sổ trung bình (# in-flight bytes) là  $\frac{3}{4} W$
  - Thông lượng trung bình là  $3/4W$  mỗi RTT

$$\text{avg TCP thruput } \frac{3}{4} \frac{W}{RTT} \text{ bytes/sec}$$



## TCP tương lai: TCP qua “ống lớn và dài”

- Ví dụ: segment 1500 byte, 100ms RTT, muốn thông lượng 10 Gbps
- Kích thước cửa sổ yêu cầu  $W = 83,333$  segment trên đường truyền
- Thông lượng trong các trường hợp mất gói, L [Mathis 1997]:

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

→ để đạt thông lượng 10 Gbps, cần thì lệ mất gói là  $L = 2 \cdot 10^{-10}$  - một tỷ lệ mất gói rất nhỏ!

- Phiên bản mới của TCP cho tốc độ cao



# TCP Công bằng

Các dịch vụ tầng vận chuyển

Vận chuyển phi kết nối: UDP

Các nguyên lý truyền dữ liệu tin cậy

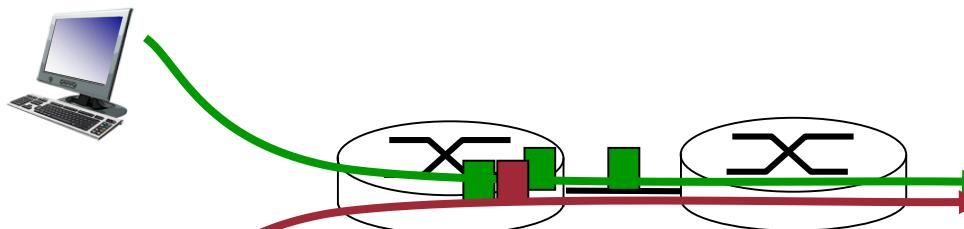
Vận chuyển hướng kết nối: TCP

Các nguyên lý điều khiển tắt nghẽn

Điều khiển tắt nghẽn TCP

*Mục tiêu công bằng:* nếu có K session TCP chia sẻ cùng đường link bị bóp cổ chai của băng thông R, thì mỗi phiên nên có tốc độ trung bình là  $R/K$

Kết nối TCP 1



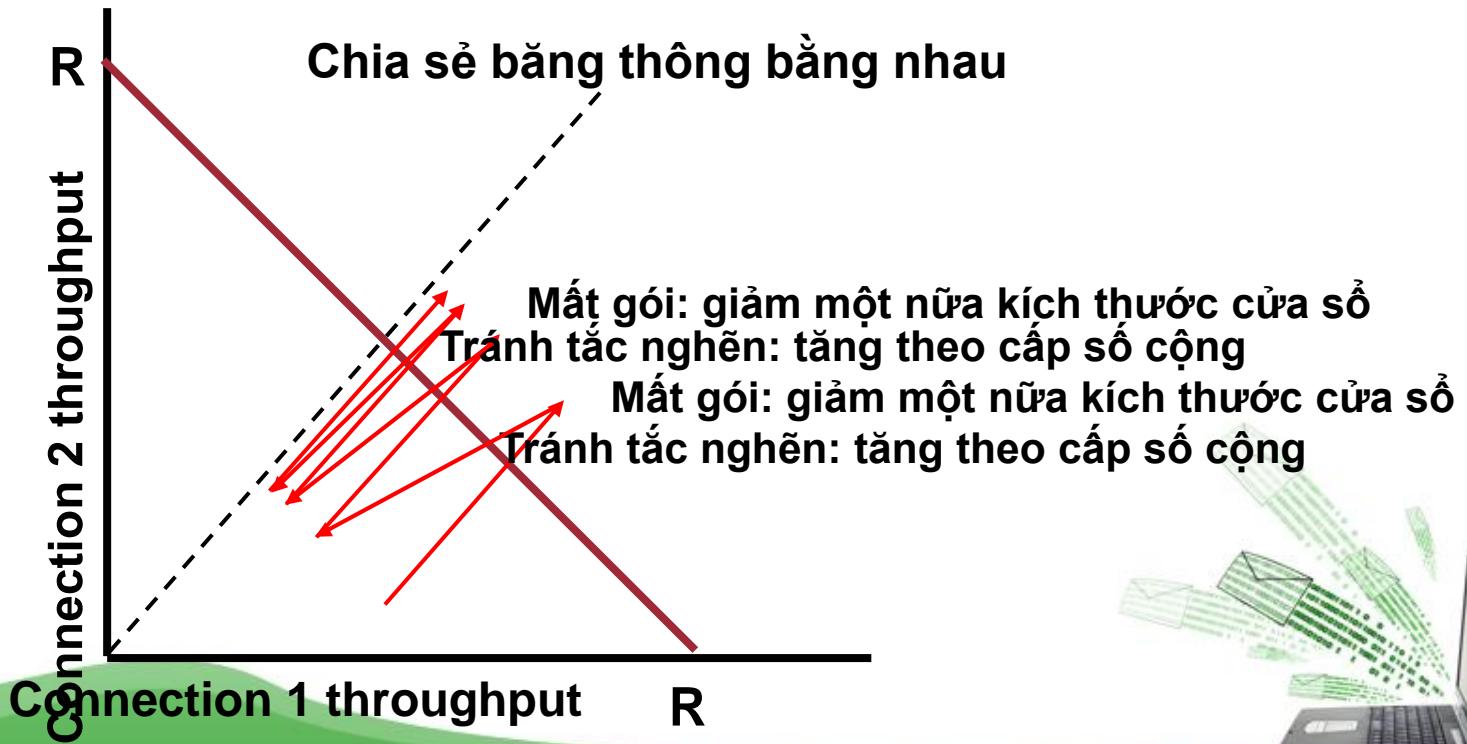
Kết nối TCP 2



# Tại sao TCP là công bằng?

2 session cạnh tranh nhau:

- Tăng theo cấp số cộng 1, khi thông lượng tăng
- Giảm lưu lượng theo cấp số nhân tương ứng



# Công bằng (tt)

## Công bằng và UDP

- Nhiều ứng dụng thường không dùng TCP
  - Không muốn tốc độ bị điều tiết do điều khiển tắc nghẽn
- Thay bằng dùng UDP:
  - Truyền audio/video với tốc độ ổn định, chịu được mất gói

## Công bằng, các kết nối TCP song song

- Ứng dụng có thể mở nhiều kết nối song song giữa 2 host
- Trình duyệt web làm điều này
- Ví dụ: đường link với tốc độ R đang có 9 kết nối:
  - Ứng dụng mới yêu cầu mở 1 kết nối TCP, có tốc độ R/10
  - Ứng dụng mới yêu cầu mở 11 kết nối TCP, có tốc độ R/2



# Tham khảo

- Phần này chỉ những:
  - Sách/ chỉ mục bắt buộc học
  - Bài tập phải làm.
  - Sách/ chương đọc thêm.



# Mạng máy tính

# ThS. Phạm Liệu

Email: lieu.pham@stu.edu.vn

2018



# NỘI DUNG CHÍNH



Giới Thiệu



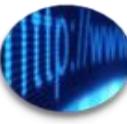
Cấu trúc trong Router



IP: Internet Protocol



Thuật Toán Routing



Routing trong Internet



Broadcast và multicast routing



# Tầng Mạng

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- Segment của tầng Vận chuyển từ host gửi đến host nhận
- Bên gửi: đóng gói (encapsulate) các segment vào trong các datagram
- Bên nhận: chuyển các segment lên tầng Vận chuyển
- Các giao thức tầng Mạng hoạt động trong *mọi* host, mọi router
- Router sẽ xem xét các trường của header trong tất cả các gói IP datagram đi qua nó



# Tầng Mạng

Giới thiệu

Virtual circuit network and diagram network

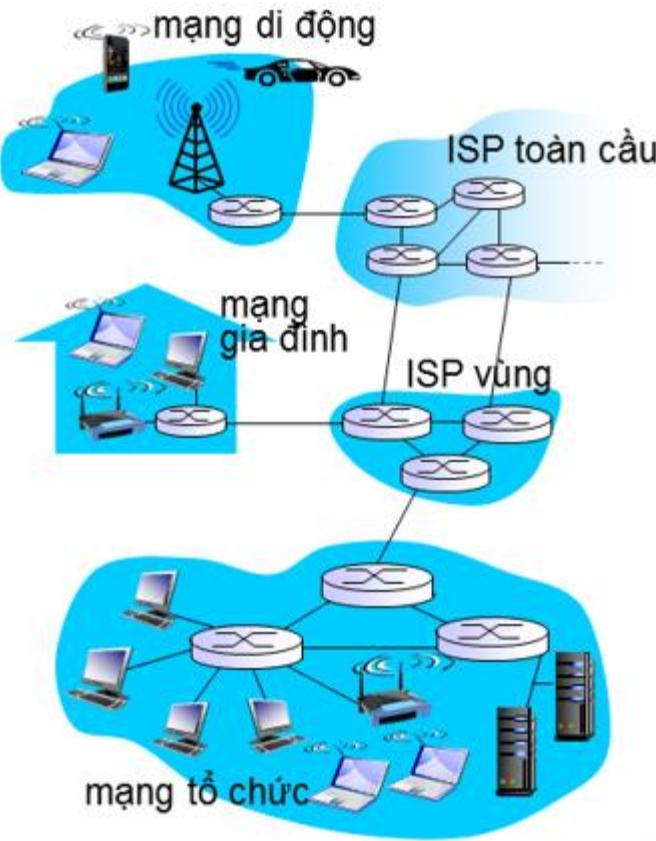
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Hai chức năng chính của tầng Mạng

- *Forwarding*: chuyển các gói tin (packet) từ đầu vào đến đầu ra thích hợp của router
- *Routing*: xác định đường đi cho các gói từ nguồn đến đích.
  - Các thuật toán định tuyến

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## Tương tự:

- ❖ *routing*: tiến trình lập kế hoạch cho chuyển đi của packet từ nguồn tới đích
- ❖ *forwarding*: tiến trình vận chuyển qua 1 điểm trung chuyển



# Tác động qua lại giữa routing và forwarding

Giới thiệu

Virtual circuit network and diagram network

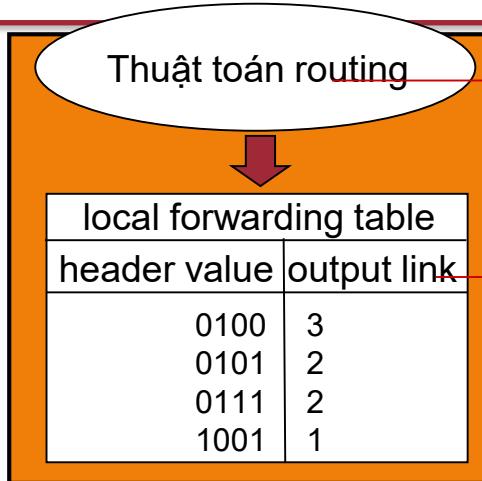
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

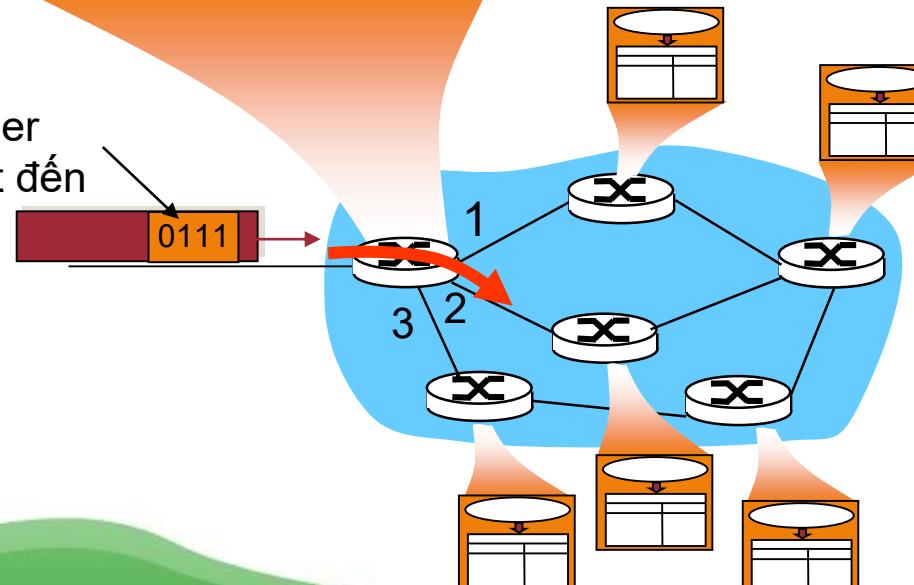
Routing trong internet



Thuật toán routing xác định đường đi của gói giữa 2 đầu cuối thông qua mạng

Bảng forwarding xác định việc chuyển gói bên trong một router

Giá trị  
trong header  
của packet đến



# Thiết lập kết nối

- Chức năng quan trọng thứ 3 trong một số kiến trúc mạng:
- Trước khi chuyển các datagram đi, 2 thiết bị đầu cuối và các router trung gian (intervening routers) thiết lập kết nối ảo (virtual connection)
  - Các router cũng tham gia
- So sánh giữa dịch vụ kết nối lớp Mạng và lớp Vận chuyển :
  - *Tầng Mạng*: giữa 2 hosts (cũng có thể bao gồm các router trung gian trong trường hợp kết nối ảo)
  - *Tầng Vận chuyển*: giữa 2 tiến trình

Giới thiệu

Virtual circuit network and diagram network

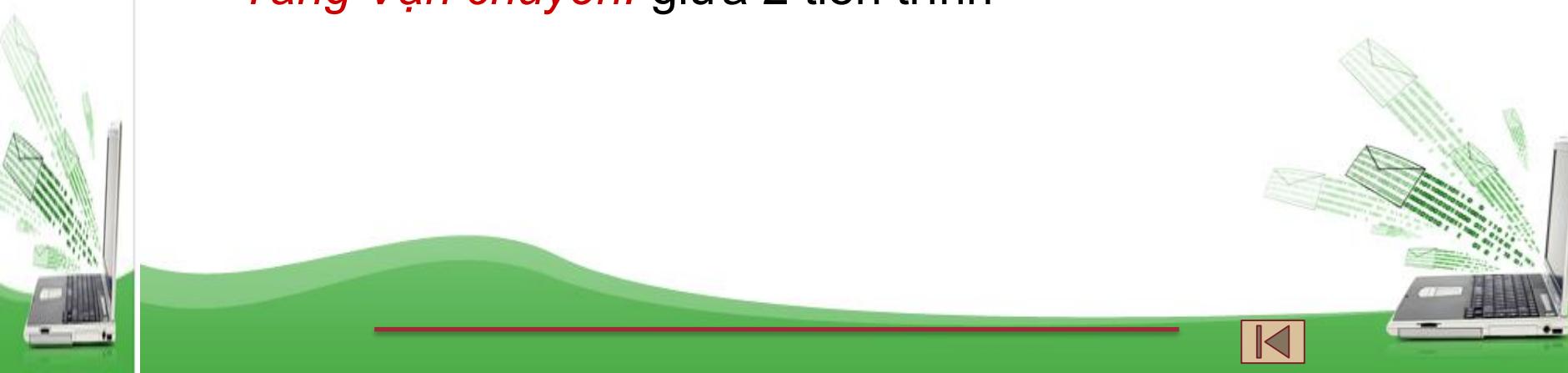
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Mô hình dịch vụ Mạng

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

*Hỏi:* mô hình dịch vụ nào cho việc tạo “kênh” truyền các datagram từ bên gửi đến bên nhận?

*Ví dụ các dịch vụ cho các datagram riêng biệt:*

- Giao nhận bảo đảm
- Giao nhận bảo đảm với độ trễ < 40ms

*Ví dụ các dịch vụ cho 1 luồng các datagram:*

- Giao nhận datagram theo thứ tự
- Băng thông được bảo đảm tối thiểu cho luồng
- Hạn chế các thay đổi trong khoảng trống giữa các gói tin



# Các mô hình dịch vụ tầng Mạng:

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Kiến trúc Network	Mô hình dịch vụ	Bảo đảm?					Phản hồi tắc nghẽn
		Băng thông	Mất mát	Thứ tự	Định thời gian		
Internet	Cố gắng tối đa	không	không	không	không	không	(phát hiện thông qua mất mát)
ATM	CBR	tốc độ không đổi	có	có	có	không	tắc nghẽn
ATM	VBR	tốc độ có bảo đảm	có	có	có	không	tắc nghẽn
ATM	ABR	bảo đảm tối thiểu	không	có	không	có	
ATM	UBR	không	không	có	không	không	



# Chương 4: Nội dung

4.1 Giới thiệu

4.3 Cấu trúc bên trong  
router

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

4.5 các thuật toán routing

- link state
- distance vector
- hierarchical routing

4.6 routing trong Internet

- RIP
- OSPF
- BGP

4.7 broadcast và multicast  
routing



# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## 4.5 các thuật toán routing

- link state
- distance vector
- hierarchical routing

## 4.6 routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 broadcast và multicast routing



# Tổng quan kiến trúc router

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

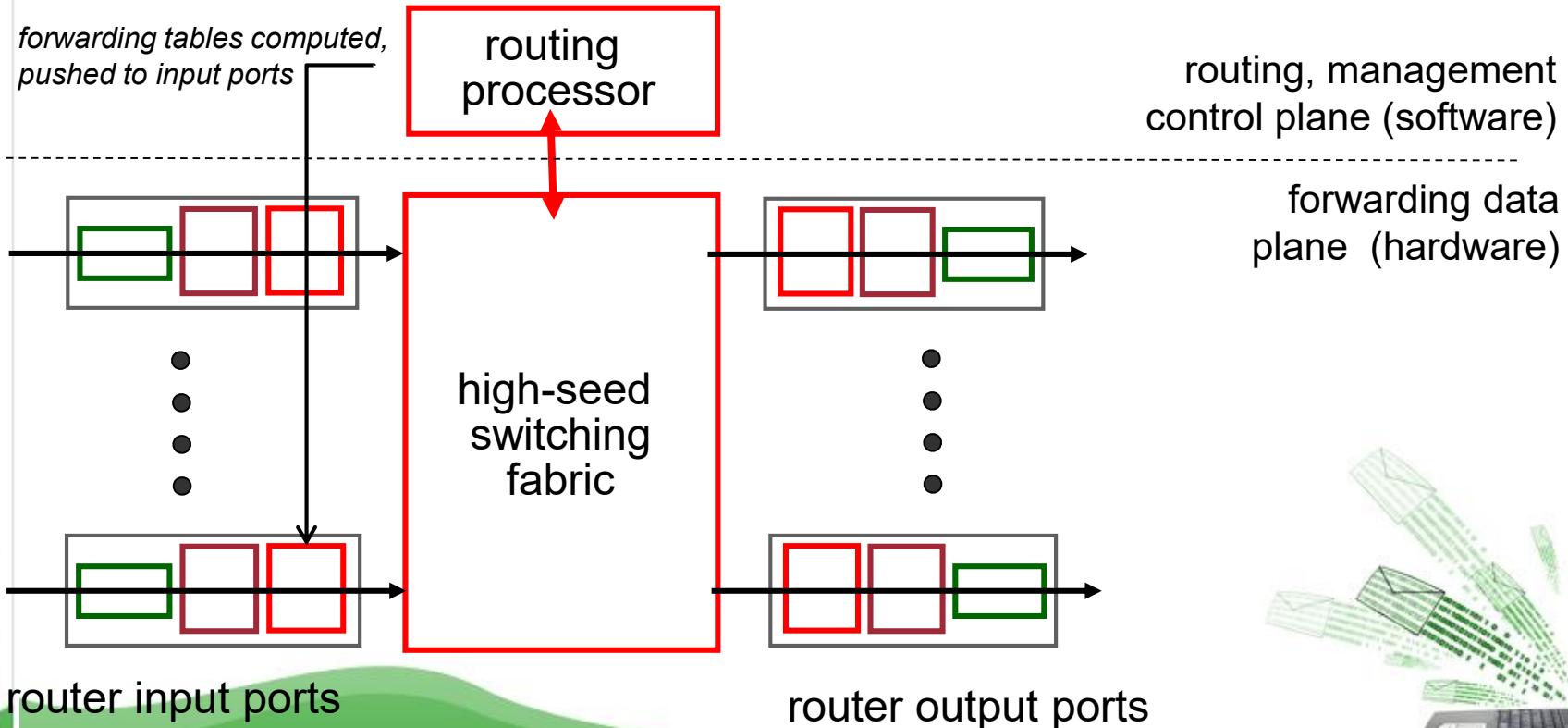
Thuật toán routing

Broadcast và multicast routing

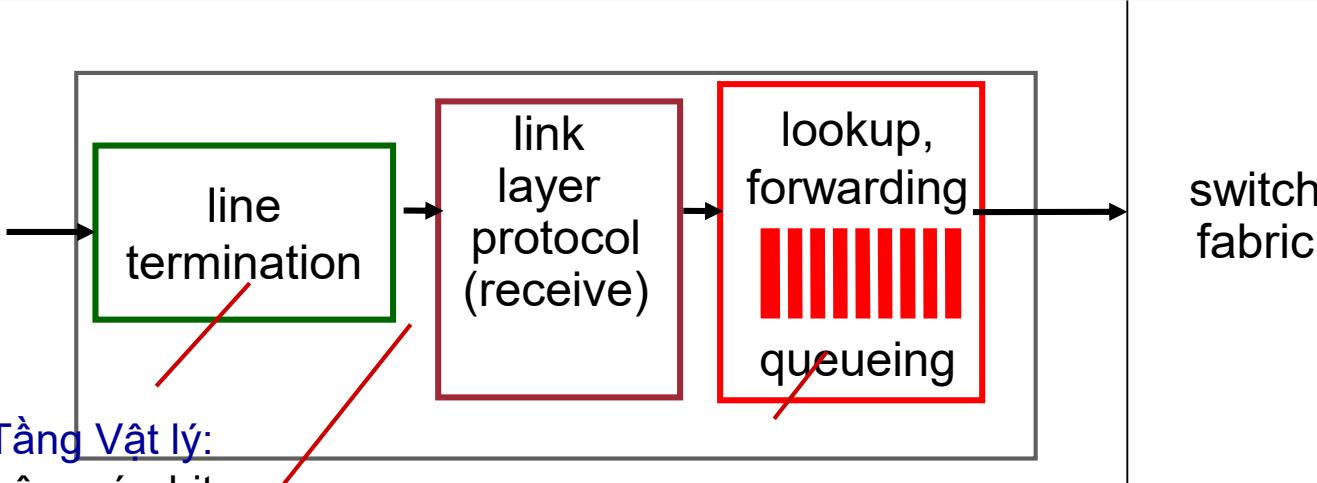
Routing trong internet

## 2 chức năng chính của router:

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



# Các chức năng của cổng vào (Input)



Tầng Vật lý:  
Tiếp nhận mức bit

Tầng Liên kết dữ liệu:

Như là Ethernet  
Xem chương 5

decentralized switching (chuyển mạch không tập trung):

- với địa chỉ đích của gói tin, tìm cổng ra dựa vào bảng forwarding trong bộ nhớ cổng vào
- mục tiêu: hoàn tất xử lý tại cổng vào kịp với “tốc độ đường truyền”
- xếp hàng: nếu gói tin đến nhanh hơn tốc độ chuyển gói bên trong mạch chuyển mạch (switch fabric)



# Mạch chuyển mạch (Switching fabrics)

- Truyền packet từ bộ nhớ đệm đầu vào đến bộ nhớ đệm đầu ra thích hợp
- Tốc độ switching: tốc độ mà các packet có thể được truyền từ đầu vào (inputs) đến đầu ra (outputs)
  - Thường được tính là tốc độ  $N \times$  (số luồng của đầu vào/đầu ra)
- 3 kiểu switching fabrics

Giới thiệu

Virtual circuit network and diagram network

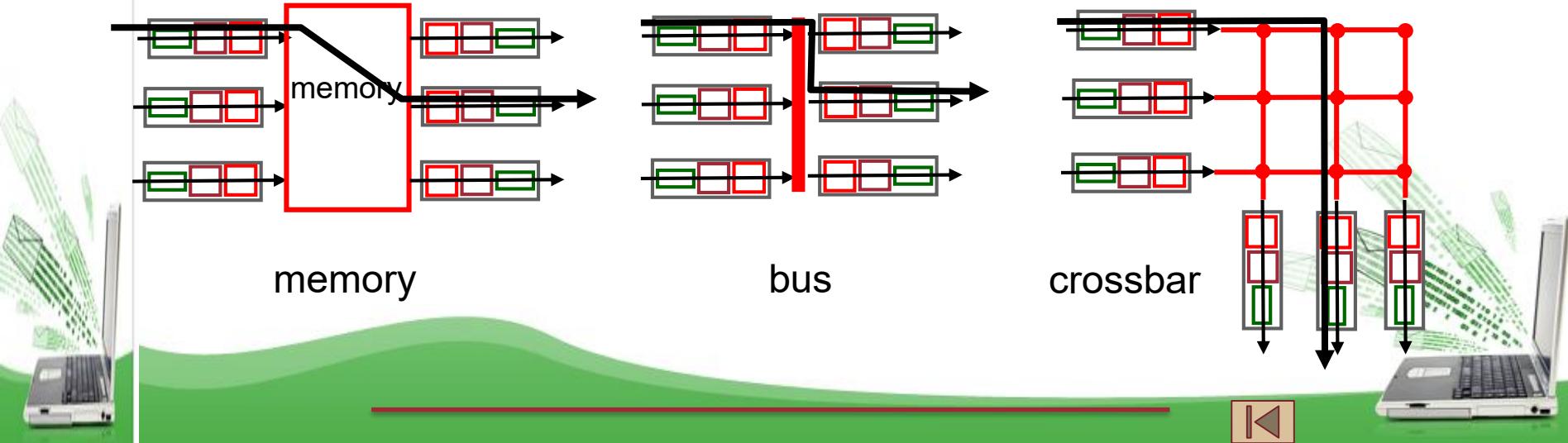
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Chuyển gói thông qua bộ nhớ (memory)

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

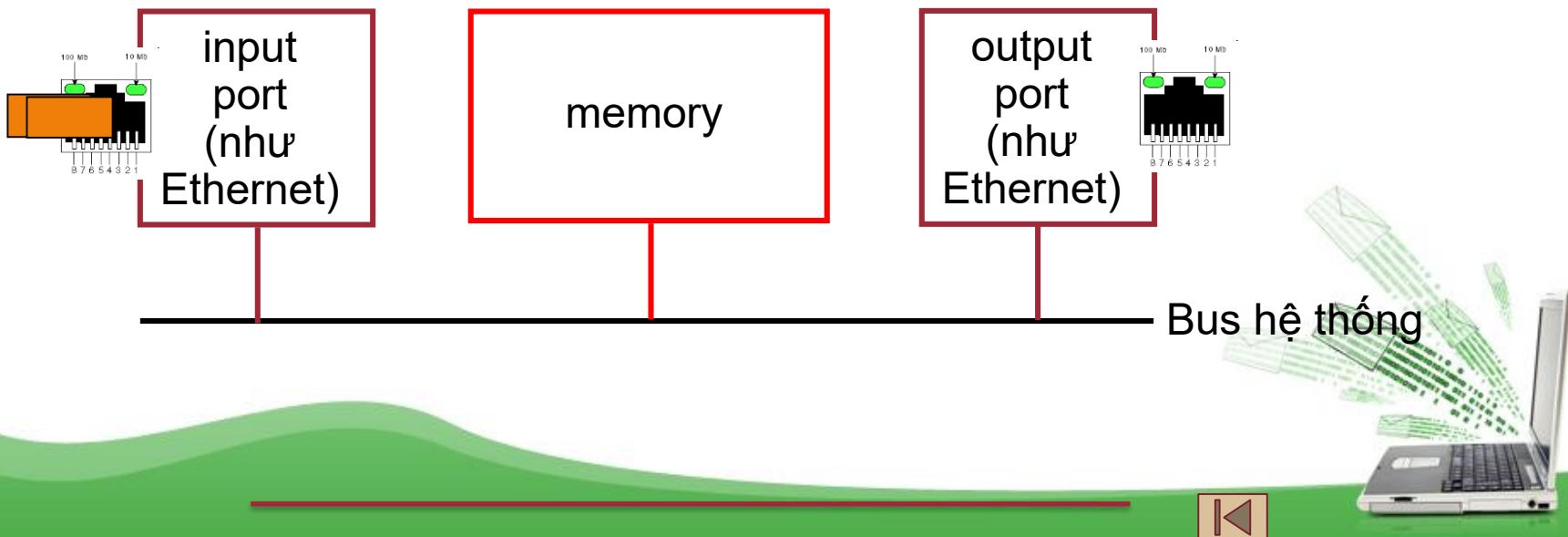
Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Các router thế hệ đầu tiên:

- Các máy tính cổ điển thực hiện chuyển gói dưới sự điều khiển của CPU
- Gói được sao chép đến bộ nhớ của hệ thống
- Tốc độ bị giới hạn bởi băng thông của bộ nhớ (gói tin phải đi qua bus hệ thống 2 lần)



# Chuyển gói thông qua bus

- ❖ Gói tin từ bộ nhớ cổng vào được chuyển đến bộ nhớ cổng ra thông qua một bus dùng chung
- ❖ **Tranh chấp bus:** tốc độ switch giới hạn bởi băng thông của bus
- ❖ 32 Gbps bus, Cisco 5600: tốc độ đủ cho truy cập và các enterprise router

Giới thiệu

Virtual circuit network and diagram network

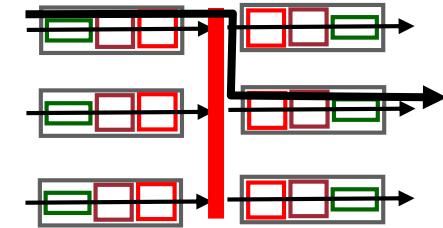
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



bus



# Chuyển mạch thông qua interconnection network

- ❖ Vượt qua các giới hạn của băng thông bus
- ❖ Các mạng kết nối nội bộ đầu tiên được phát triển để kết nối các bộ vi xử lý trong hệ thống đa xử lý
- ❖ Thiết kế nâng cao: chia nhỏ gói tin thành các gói có độ dài cố định, chuyển các gói thông qua bản mạch.
- ❖ Cisco 12000: chuyển 60 Gbps thông qua interconnection network

Giới thiệu

Virtual circuit network and diagram network

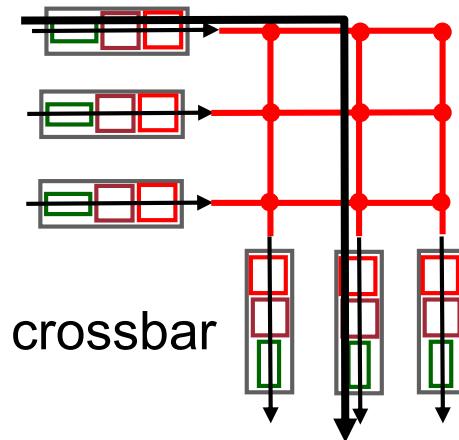
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Các cổng ra (Output)

Giới thiệu

Virtual circuit network and diagram network

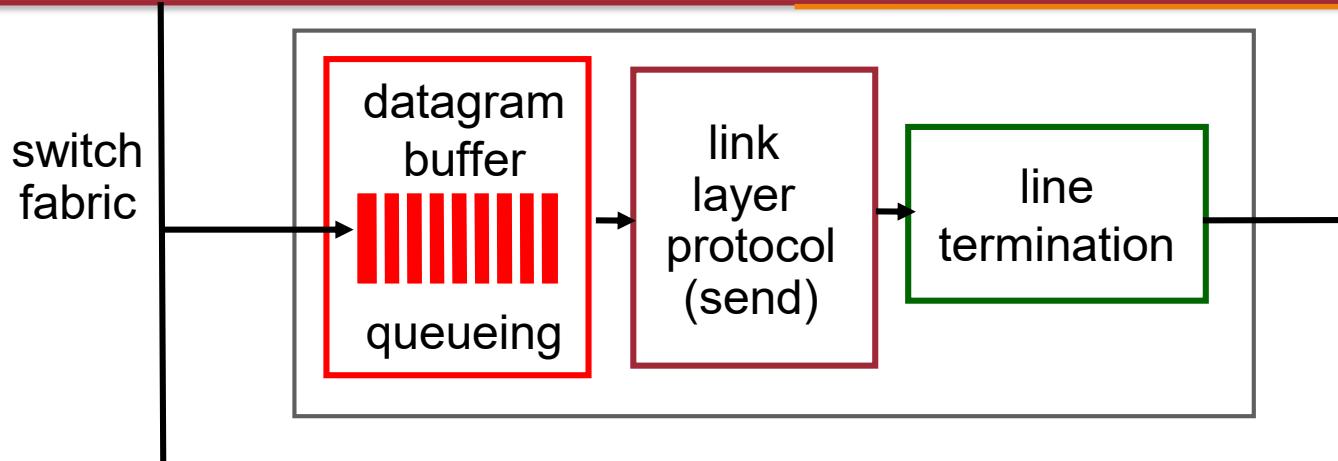
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



- *Xếp hàng trong bộ đệm* được yêu cầu khi tốc độ chuyển các gói tin đến từ mạch xử lý của switch nhanh hơn tốc độ truyền
- *Lập lịch (Scheduling discipline)* cách thức chọn trong số các gói tin đã xếp hàng để truyền



# Xếp hàng tại cổng ra

Giới thiệu

Virtual circuit network and diagram network

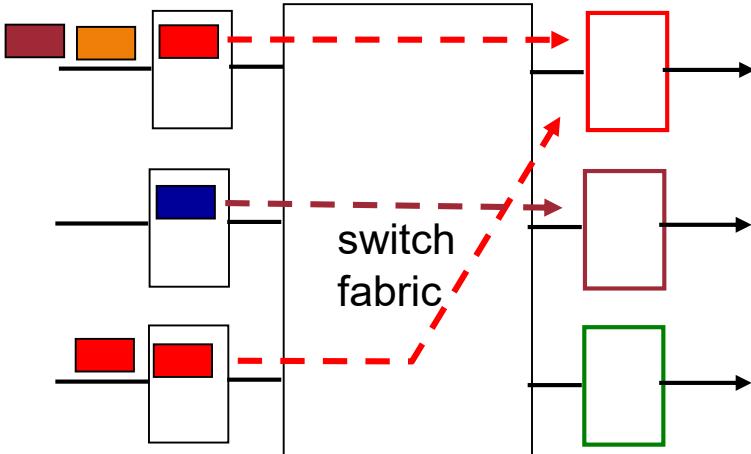
Cấu trúc trong router

Ip:internet protocol

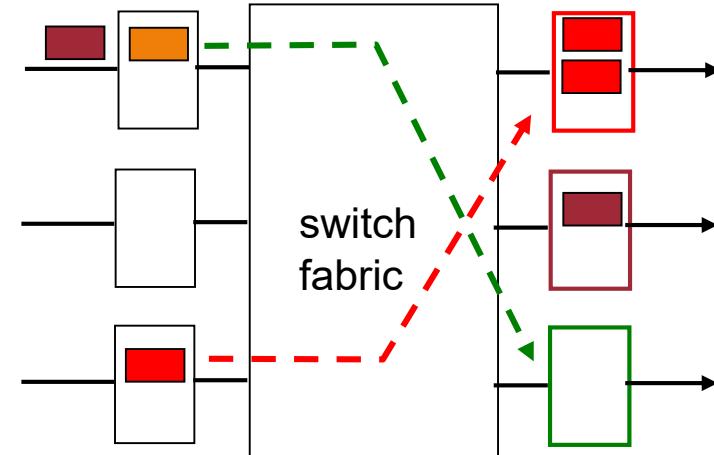
Thuật toán routing

Broadcast và multicast routing

Routing trong internet



at  $t$ , các packet nhiều hơn  
từ đầu vào đến đầu ra



one packet time later

- Lưu đệm khi tốc độ đến qua switch vượt quá tốc độ dòng ra (output line)
- Xếp hàng (trễ) và mất gói vì bộ nhớ đệm tại cổng ra bị tràn (overflow)!**



# Kích thước bộ đệm?

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- RFC 3439: kích thước đệm trung bình bằng với thời gian RTT “thông thường” (250 msec) nhân với dung lượng đường truyền C
  - Ví dụ  $C = 10 \text{ Gpbs link} \rightarrow 2.5 \text{ Gbit buffer}$
- Khuyến nghị mới nhất: với  $N$  luồng, đệm bằng với

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



# Sắp hàng tại cổng vào

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

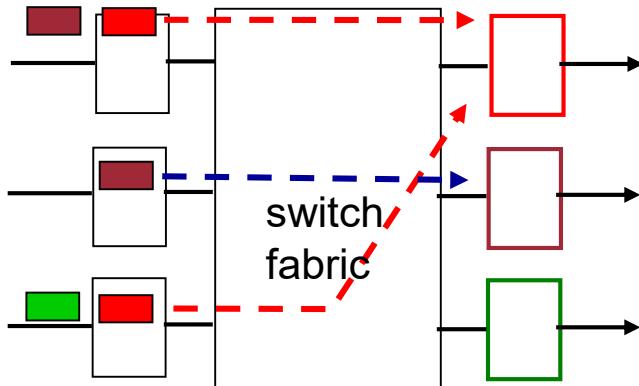
Ip:internet protocol

Thuật toán routing

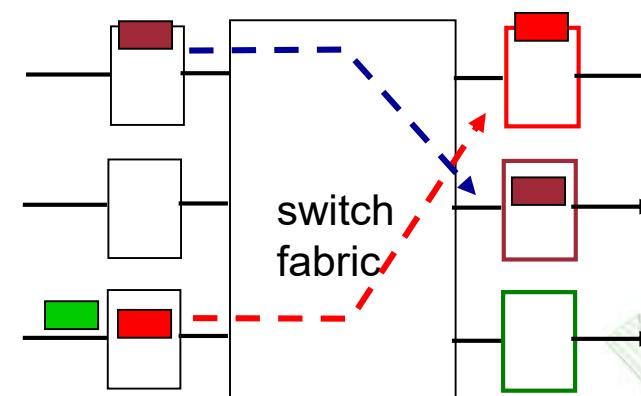
Broadcast và multicast routing

Routing trong internet

- Mạch xử lý chậm hơn tốc độ cổng vào cũng cùng dẫn đến việc phải xếp hàng tại các hàng đợi cổng vào
  - xếp hàng trễ và mất mát bởi vì bộ đệm tại cổng vào bị tràn!*
- Head-of-the-Line (HOL) blocking: gói tin được sắp hàng tại phía trước hàng đợi ngăn cản các gói khác trong hàng đợi di chuyển lên trước



Sự cạnh tranh tại cổng ra:  
Chỉ có một gói màu đỏ có thể  
được truyền.  
*packet màu đỏ thấp hơn bị  
chặn lại*



packet màu xanh lá  
bị chặn HOL  
blocking



# Chương 4: Nội dung

4.1 Giới thiệu

4.3 Cấu trúc bên trong router

**4.4 IP: Internet Protocol**

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

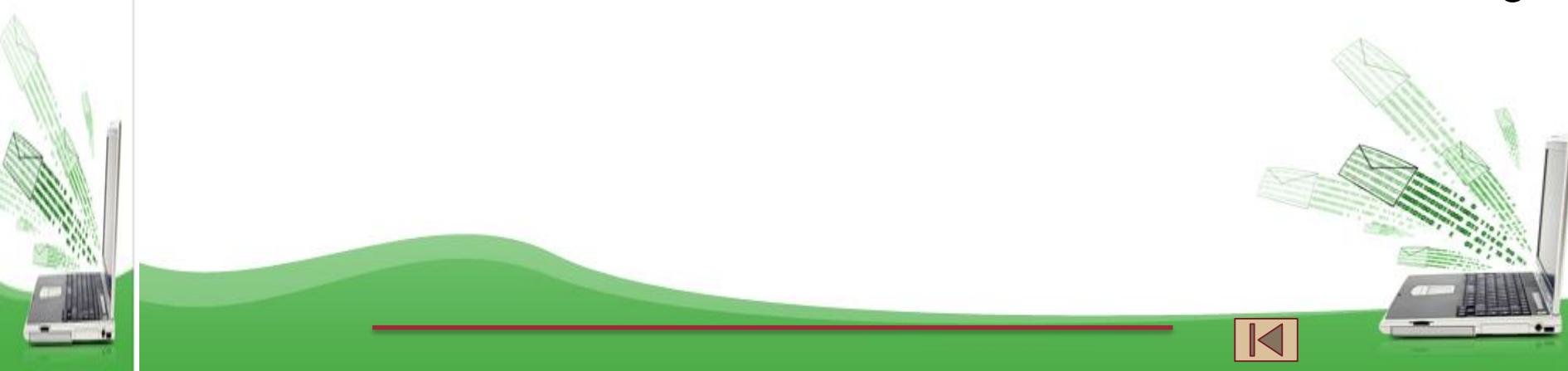
4.5 các thuật toán routing

- link state
- distance vector
- hierarchical routing

4.6 routing trong Internet

- RIP
- OSPF
- BGP

4.7 broadcast và multicast routing



# Tầng Mạng Internet

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

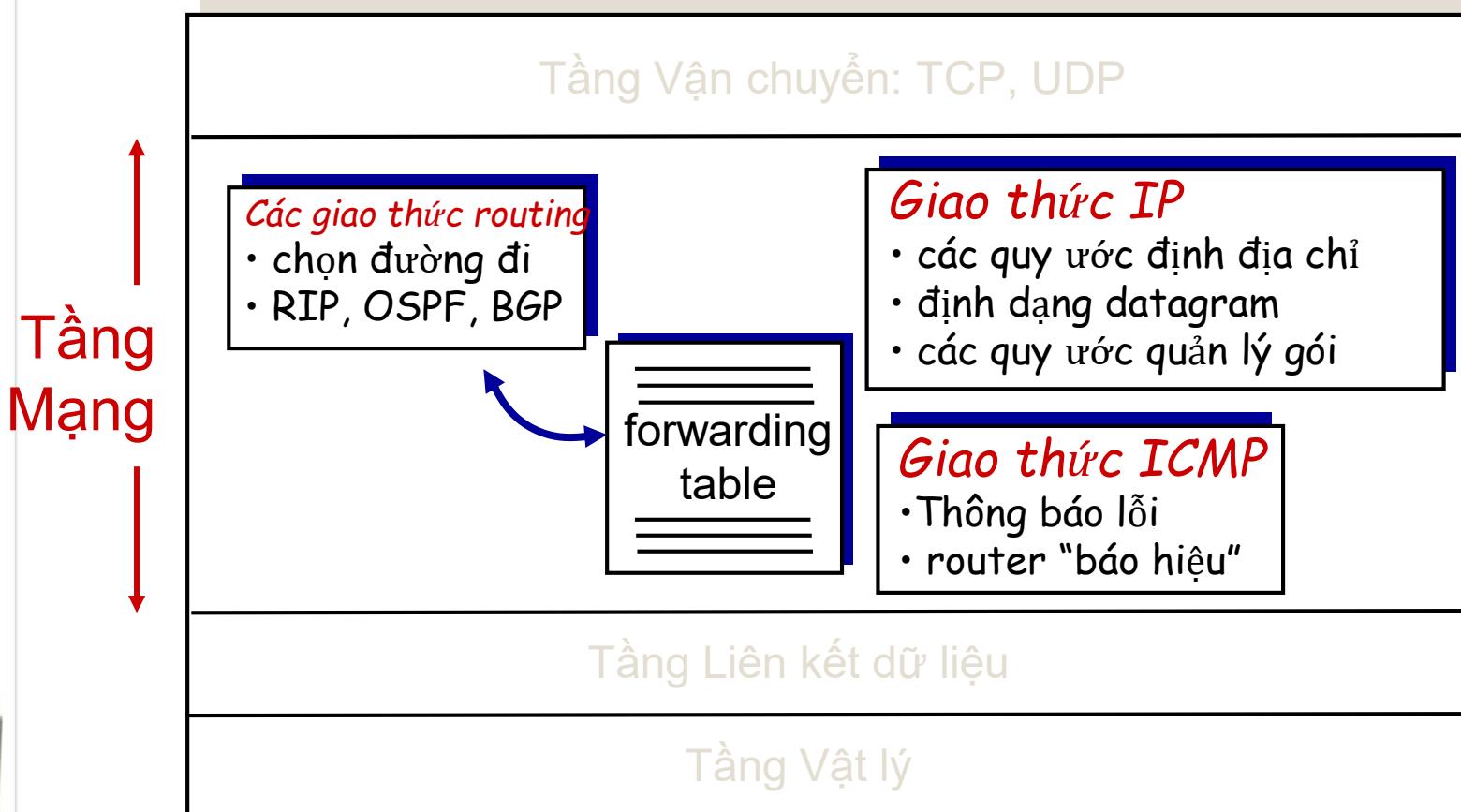
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Các chức năng tầng Mạng của host và router:



# Định dạng IP datagram

Giới thiệu

Virtual circuit network and diagram network

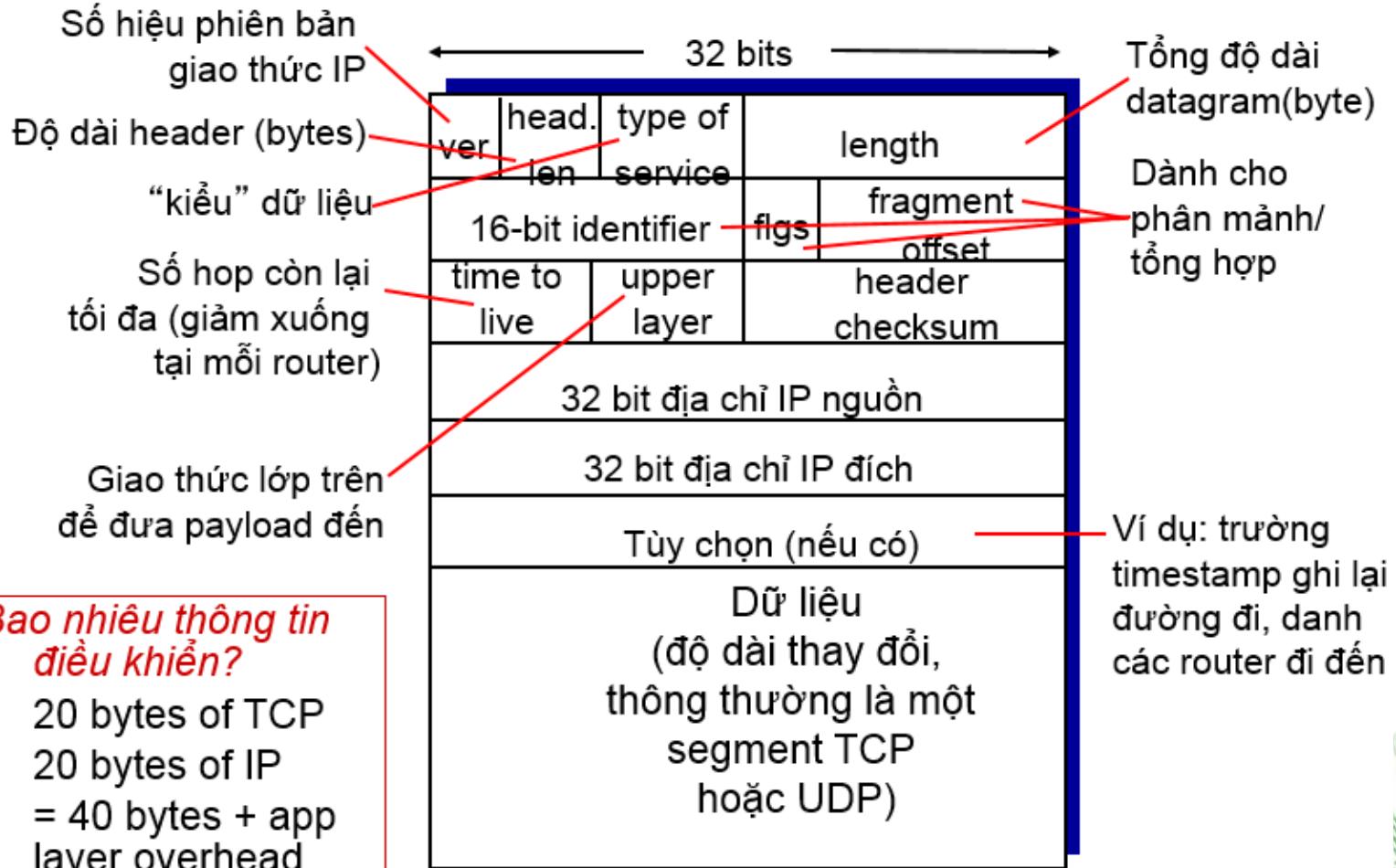
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Phân mảnh và tổng hợp IP

- Các đoạn kết nối mạng có MTU (max.transfer size) – frame lớn nhất có thể truyền trên kết nối
  - Các kiểu kết nối khác nhau có các MTU khác nhau
- Các gói IP datagram lớn được chia (“fragmented”) bên trong mạng
  - 1 datagram thành 1 vài datagram
  - “tổng hợp” chỉ được thực hiện ở đích cuối cùng
  - Các bit của IP header được sử dụng để xác định, xếp thứ tự các fragment liên quan

Giới thiệu

Virtual circuit network and diagram network

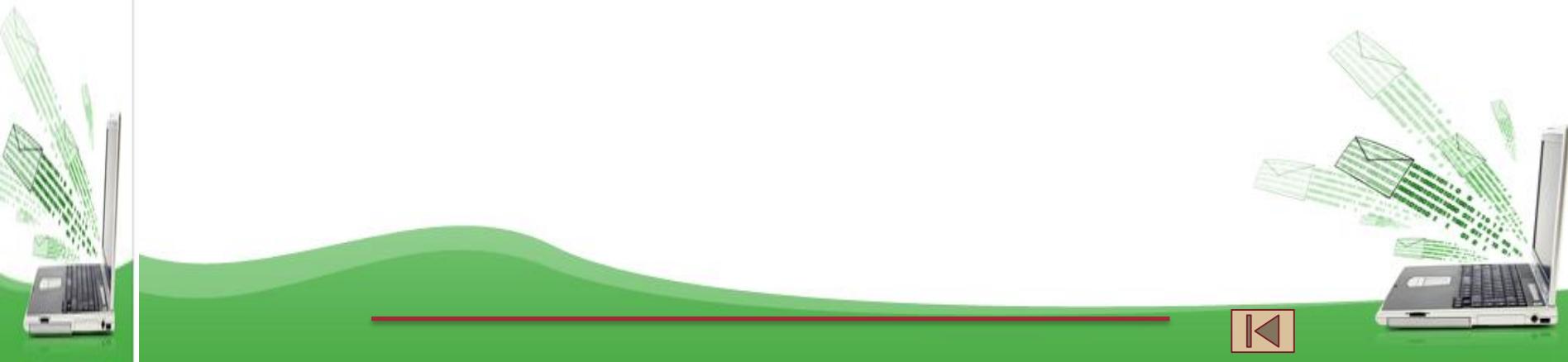
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Phân mảnh và tống hợp IP

Giới thiệu

Virtual circuit network and diagram network

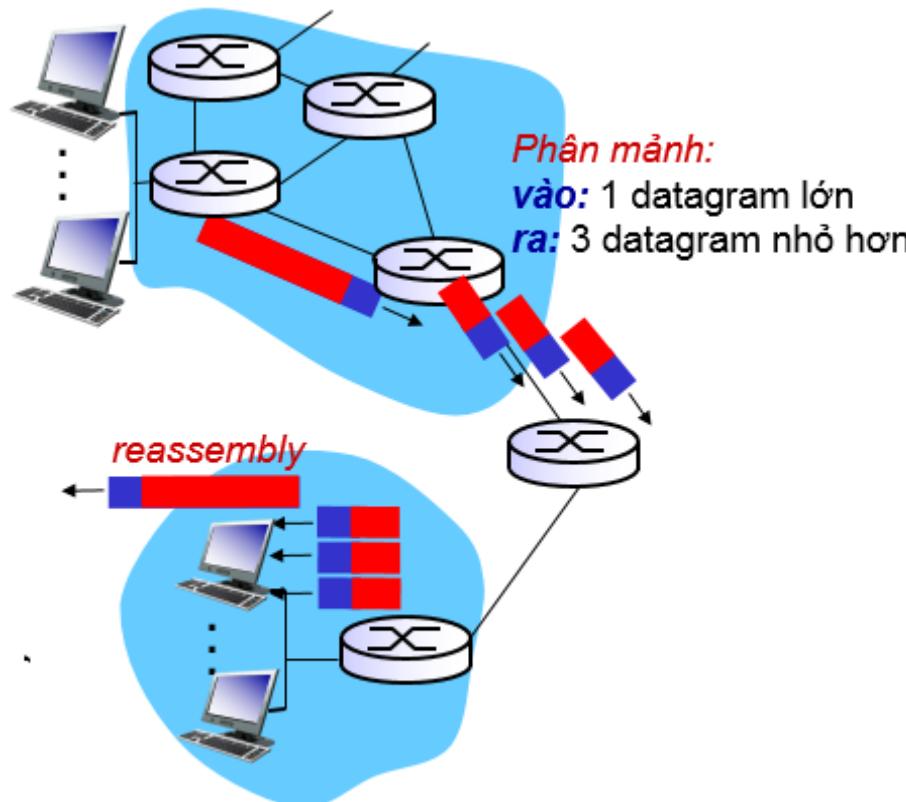
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Phân mảnh và tống hợp IP

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Ví dụ:

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

- ❖ 4000 byte datagram
- ❖ MTU = 1500 bytes

1480 bytes  
trong trường dữ liệu

$$\text{offset} = \frac{1480}{8}$$

1 datagram lớn thành vài datagram nhỏ hơn

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

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

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



# Chương 4: Nội dung

4.1 Giới thiệu

4.2 virtual circuit network  
(Mạng mạch ảo) và  
datagram network  
(Mạng chuyển gói)

4.3 Cấu trúc bên trong  
router

4.4 IP: Internet Protocol

- Định dạng datagram
- **IPv4 addressing**
- ICMP
- IPv6

4.5 các thuật toán routing

- link state
- distance vector
- hierarchical routing

4.6 routing trong Internet

- RIP
- OSPF
- BGP

4.7 broadcast và multicast  
routing

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Định địa chỉ IP: giới thiệu

- *Địa chỉ IP*: 32-bit nhận dạng cho host, router *interface*
- *interface*: kết nối giữa host/router và đường kết nối vật lý
  - Router thường có nhiều interface
  - host thường có 1 hoặc 2 interface (ví dụ wired Ethernet, wireless 802.11)
- *Mỗi địa chỉ IP được liên kết với mỗi interface*

Giới thiệu

Virtual circuit network and diagram network

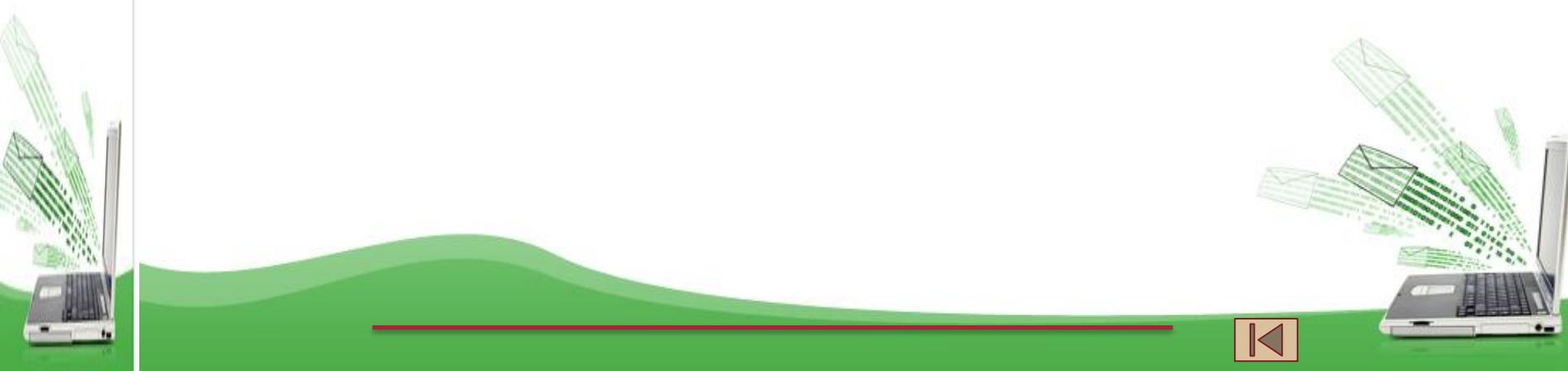
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Định địa chỉ IP: giới thiệu

Giới thiệu

Virtual circuit network and diagram network

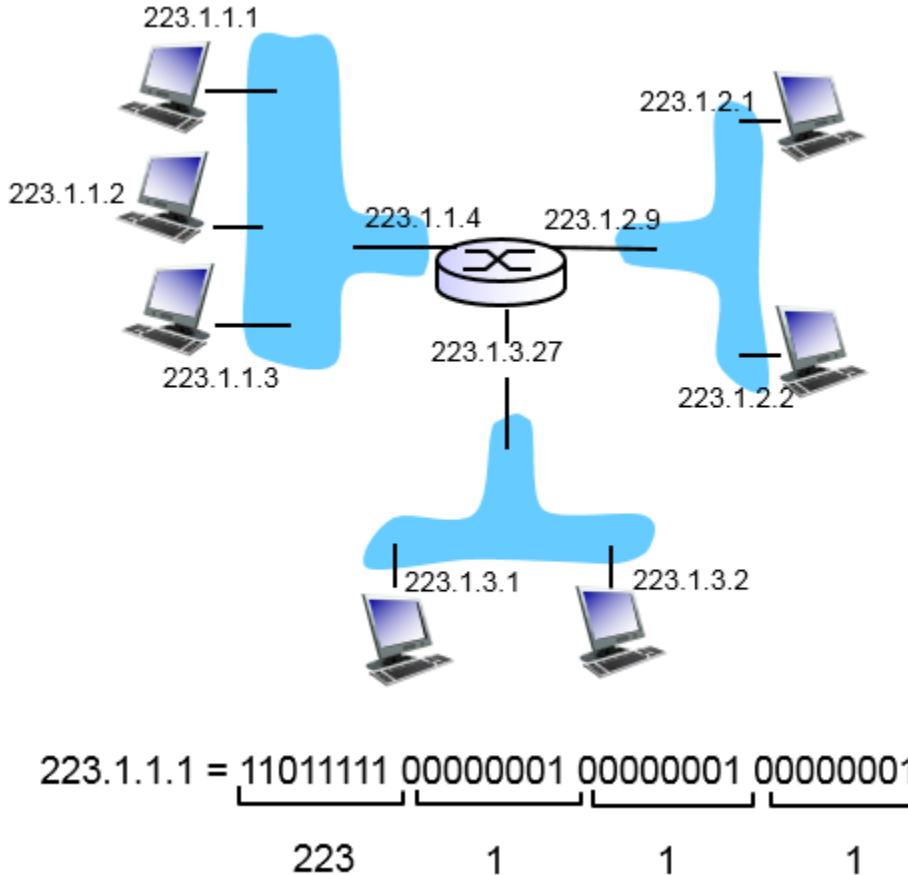
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Định địa chỉ IP: giới thiệu

*Hỏi: các interface thật sự được kết nối như thế nào?*

*Đáp: tìm hiểu kỹ hơn trong chương 5, 6.*

*Đáp:* các interface Ethernet có dây được kết nối bởi các switch Ethernet

*Bây giờ:* không cần lo lắng về cách mà 1 interface được kết nối với một interface khác (không có router trung gian))

Giới thiệu

Virtual circuit network and diagram network

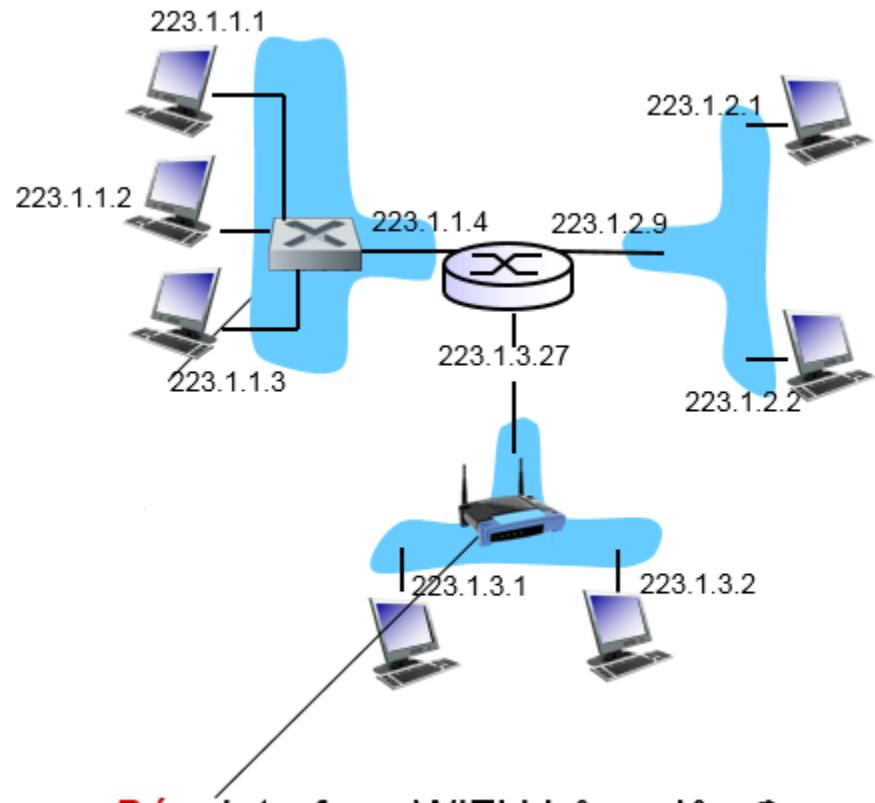
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



*Đáp:* interface WiFi không dây được kết nối thông qua WiFi base station



# Các Subnet (mạng con)

- Đại chỉ IP:
  - Phần subnet – các bit bên trái
  - Phần host – các bit bên phải
- Subnet là gì?*
  - Các interface của thiết bị có phần subnet của địa chỉ IP giống nhau
  - Có thể giao tiếp vật lý với nhau mà không cần **router trung gian can thiệp**

Giới thiệu

Virtual circuit network and diagram network

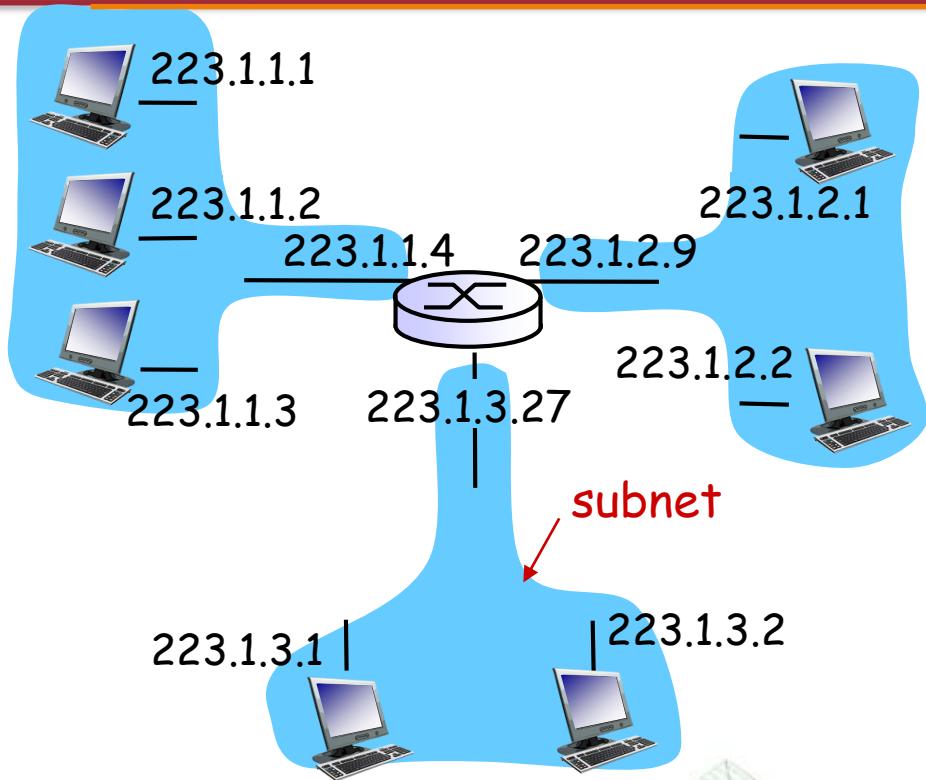
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



Mạng gồm 3 subnet



# Các mạng con (subnet)

## Phương pháp

- Để xác định các subnet, tách mỗi interface từ host hoặc router của nó, tạo vùng các mạng độc lập
- Mỗi mạng độc lập được gọi là một *subnet*

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

223.1.1.0/24

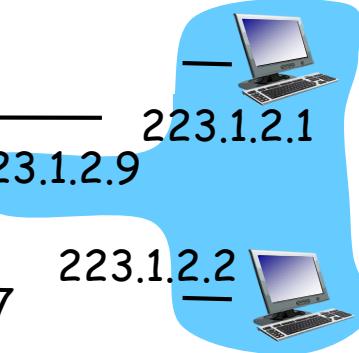


223.1.1.1

223.1.1.2

223.1.1.3

223.1.2.0/24



223.1.2.1

223.1.2.2

223.1.2.9

223.1.2.4

223.1.2.5

223.1.2.6

223.1.2.7

223.1.2.8

223.1.2.9

223.1.2.10

223.1.2.11

223.1.2.12

223.1.2.13

223.1.2.14

223.1.2.15

223.1.2.16

223.1.2.17

223.1.2.18

223.1.2.19

223.1.2.20

223.1.2.21

223.1.2.22

223.1.2.23

223.1.2.24

223.1.2.25

223.1.2.26

223.1.2.27

223.1.2.28

223.1.2.29

223.1.2.30

223.1.2.31

223.1.2.32

223.1.2.33

223.1.2.34

223.1.2.35

223.1.2.36

223.1.2.37

223.1.2.38

223.1.2.39

223.1.2.40

223.1.2.41

223.1.2.42

223.1.2.43

223.1.2.44

223.1.2.45

223.1.2.46

223.1.2.47

223.1.2.48

223.1.2.49

223.1.2.50

223.1.2.51

223.1.2.52

223.1.2.53

223.1.2.54

223.1.2.55

223.1.2.56

223.1.2.57

223.1.2.58

223.1.2.59

223.1.2.60

223.1.2.61

223.1.2.62

223.1.2.63

223.1.2.64

223.1.2.65

223.1.2.66

223.1.2.67

223.1.2.68

223.1.2.69

223.1.2.70

223.1.2.71

223.1.2.72

223.1.2.73

223.1.2.74

223.1.2.75

223.1.2.76

223.1.2.77

223.1.2.78

223.1.2.79

223.1.2.80

223.1.2.81

223.1.2.82

223.1.2.83

223.1.2.84

223.1.2.85

223.1.2.86

223.1.2.87

223.1.2.88

223.1.2.89

223.1.2.90

223.1.2.91

223.1.2.92

223.1.2.93

223.1.2.94

223.1.2.95

223.1.2.96

223.1.2.97

223.1.2.98

223.1.2.99

223.1.2.100

223.1.2.101

223.1.2.102

223.1.2.103

223.1.2.104

223.1.2.105

223.1.2.106

223.1.2.107

223.1.2.108

223.1.2.109

223.1.2.110

223.1.2.111

223.1.2.112

223.1.2.113

223.1.2.114

223.1.2.115

223.1.2.116

223.1.2.117

223.1.2.118

223.1.2.119

223.1.2.120

223.1.2.121

223.1.2.122

223.1.2.123

223.1.2.124

223.1.2.125

223.1.2.126

223.1.2.127

223.1.2.128

223.1.2.129

223.1.2.130

223.1.2.131

223.1.2.132

223.1.2.133

223.1.2.134

223.1.2.135

223.1.2.136

223.1.2.137

223.1.2.138

223.1.2.139

223.1.2.140

223.1.2.141

223.1.2.142

223.1.2.143

223.1.2.144

223.1.2.145

223.1.2.146

223.1.2.147

223.1.2.148

223.1.2.149

223.1.2.150

223.1.2.151

223.1.2.152

223.1.2.153

223.1.2.154

223.1.2.155

223.1.2.156

223.1.2.157

223.1.2.158

223.1.2.159

223.1.2.160

223.1.2.161

223.1.2.162

223.1.2.163

223.1.2.164

223.1.2.165

223.1.2.166

223.1.2.167

223.1.2.168

223.1.2.169

223.1.2.170

223.1.2.171

223.1.2.172

223.1.2.173

223.1.2.174

223.1.2.175

223.1.2.176

223.1.2.177

223.1.2.178

223.1.2.179

223.1.2.180

223.1.2.181

223.1.2.182

223.1.2.183

223.1.2.184

223.1.2.185

223.1.2.186

223.1.2.187

223.1.2.188

223.1.2.189

223.1.2.190

223.1.2.191

223.1.2.192

223.1.2.193

223.1.2.194

223.1.2.195

223.1.2.196

223.1.2.197

223.1.2.198

223.1.2.199

223.1.2.200

223.1.2.201

223.1.2.202

223.1.2.203

223.1.2.204

223.1.2.205

223.1.2.206

223.1.2.207

223.1.2.208

223.1.2.209

223.1.2.210

223.1.2.211

223.1.2.212

223.1.2.213

223.1.2.214

223.1.2.215

223.1.2.216

223.1.2.217

223.1.2.218

223.1.2.219

223.1.2.220

223.1.2.221

223.1.2.222

223.1.2.223

223.1.2.224

223.1.2.225

223.1.2.226

223.1.2.227

223.1.2.228

223.1.2.229

223.1.2.230

223.1.2.231

223.1.2.232

223.1.2.233

223.1.2.234

223.1.2.235

223.1.2.236

223.1.2.237

223.1.2.238

223.1.2.239

223.1.2.240

223.1.2.241

223.1.2.242

223.1.2.243

223.1.2.244

223.1.2.245

223.1.2.246

223.1.2.247

223.1.2.248

223.1.2.249

223.1.2.250

223.1.2.251

223.1.2.252

223.1.2.253

223.1.2.254

223.1.2.255

223.1.2.256

223.1.2.257

223.1.2.258

223.1.2.259

223.1.2.260

223.1.2.261

223.1.2.262

223.1.2.263

223.1.2.264

223.1.2.265

223.1.2.266

223.1.2.267

223.1.2.268

223.1.2.269

223.1.2.270

223.1.2.271

223.1.2.272

# Subnets

Có bao nhiêu?

Giới thiệu

Virtual circuit network and diagram network

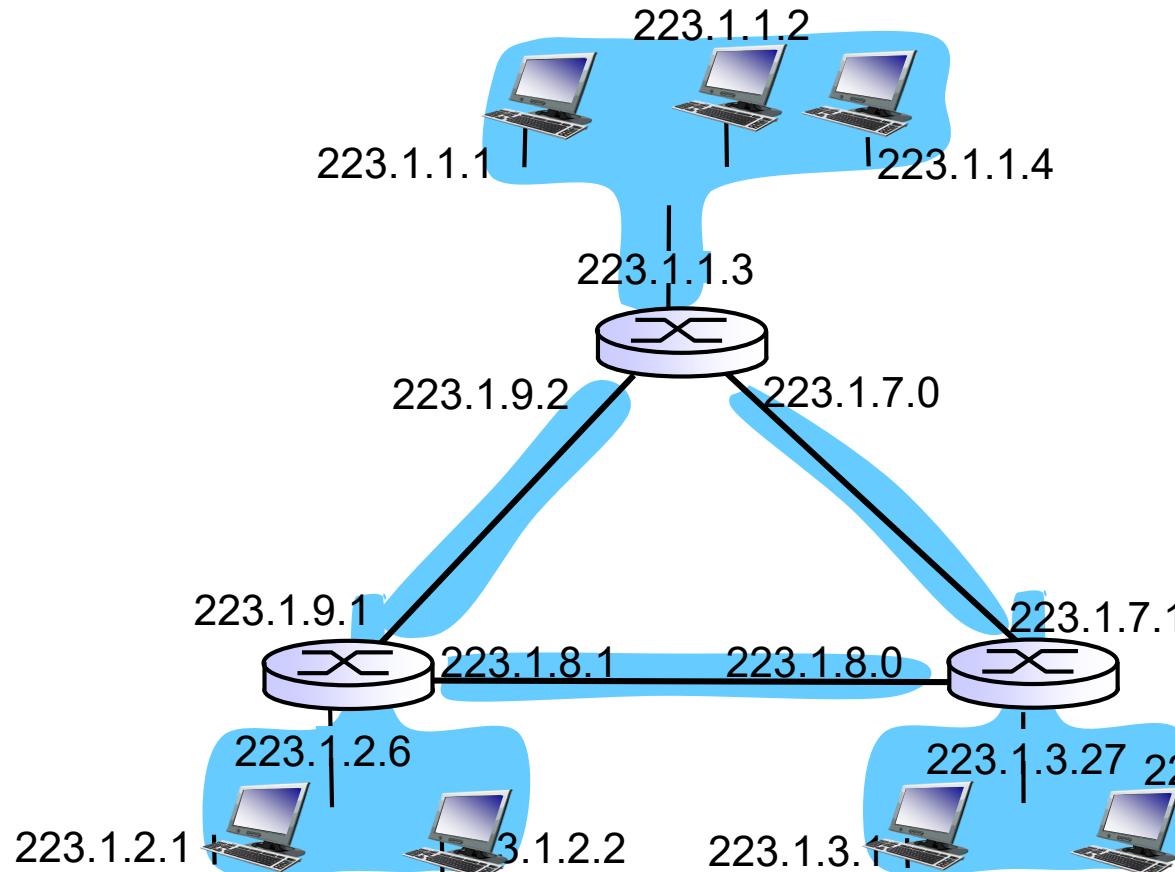
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Định địa chỉ IP: CIDR

- Giới thiệu
  - Virtual circuit network and diagram network
  - Cấu trúc trong router
  - Ip:internet protocol
  - Thuật toán routing
  - Broadcast và multicast routing
  - Routing trong internet

# CIDR: Classless InterDomain Routing

- Phần subnet của địa chỉ có độ dài bất kỳ
  - Định dạng địa chỉ:  $a.b.c.d/x$ , trong đó x là số các bits trong phần subnet của địa chỉ



# Địa chỉ IP: làm sao để lấy một địa chỉ?

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ipip network protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Hỏi: làm thế nào một host lấy được địa chỉ IP?

- người quản trị hệ thống lưu địa chỉ trong cấu hình hệ thống
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- **DHCP: Dynamic Host Configuration Protocol:** tự động lấy địa chỉ IP từ server
  - “plug-and-play”



# DHCP: Dynamic Host Configuration Protocol

**Mục tiêu:** cho phép host (máy) tự động lấy địa chỉ IP của nó từ server trong mạng khi host đó tham gia vào mạng

- Có thể gia hạn địa chỉ IP mà host đó vừa được cấp
- Cho phép tái sử dụng các địa chỉ IP (chỉ giữ địa chỉ trong khi được kết nối/"on")
- Hỗ trợ cho người dùng di động muốn tham gia vào mạng (trong thời gian ngắn)

**Tổng quan DHCP :**

- Host gửi quảng bá (broadcasts) thông điệp “**DHCP discover**” [tùy chọn] cho tất cả các máy trong mạng
- DHCP server nhận và trả lời bằng thông điệp “**DHCP offer**” [tùy chọn]
- host yêu cầu địa chỉ IP: “**DHCP request**” msg
- DHCP server gửi địa chỉ: “**DHCP ack**” msg

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Ngữ cảnh DHCP client-server

Giới thiệu

Virtual circuit network and diagram network

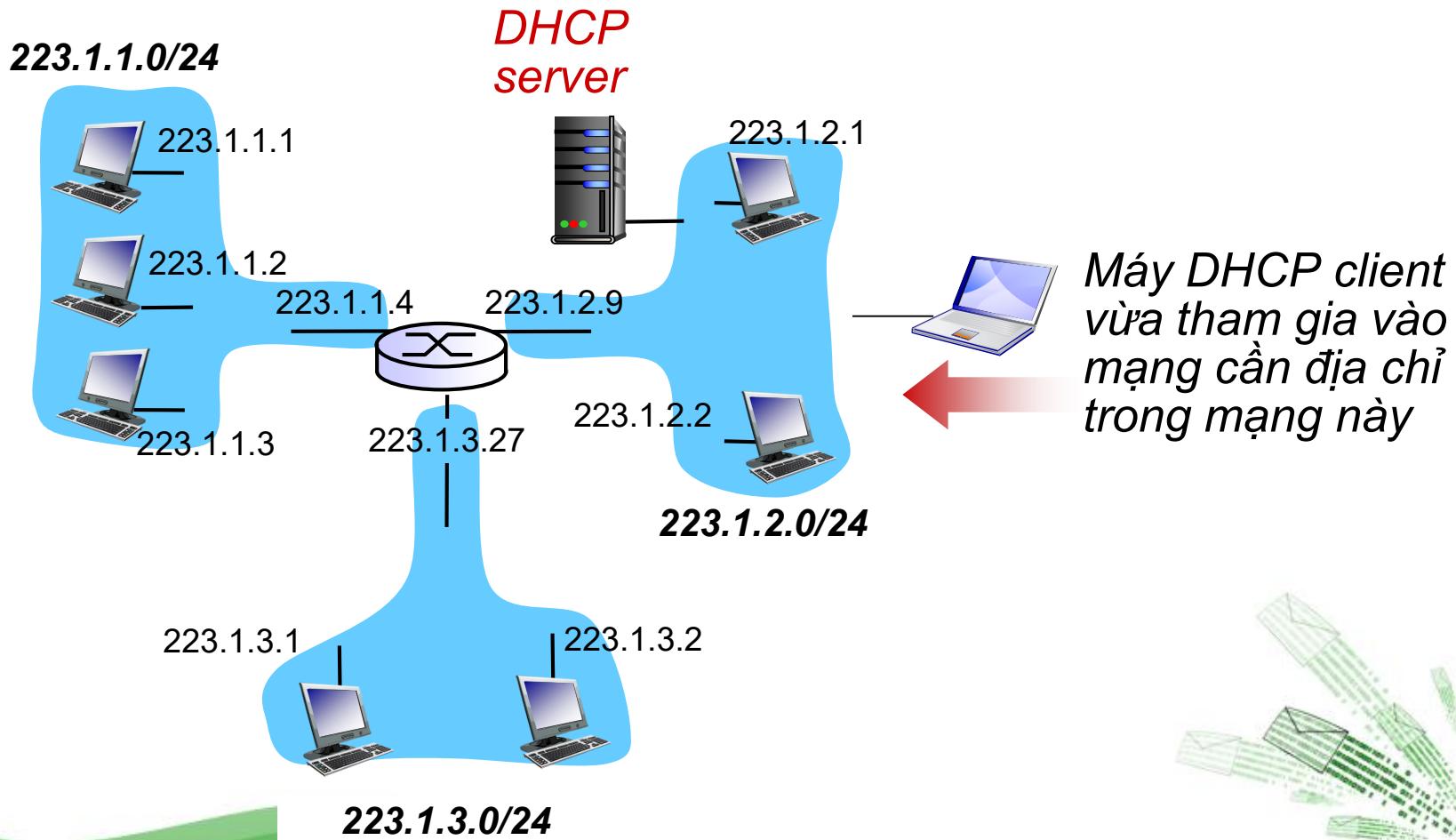
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Ngữ cảnh DHCP client-server

DHCP server: 223.1.2.5



DHCP discover

src : 0.0.0.0, 68  
dest.: 255.255.255.255,67  
yiaddr: 0.0.0.0  
transaction ID: 654

arriving client



DHCP offer

src: 223.1.2.5, 67  
dest: 255.255.255.255, 68  
yiaddr: 223.1.2.4  
transaction ID: 654  
lifetime: 3600 secs

DHCP request

src: 0.0.0.0, 68  
dest:: 255.255.255.255, 67  
yiaddr: 223.1.2.4  
transaction ID: 655  
lifetime: 3600 secs

DHCP ACK

src: 223.1.2.5, 67  
dest: 255.255.255.255, 68  
yiaddr: 223.1.2.4  
transaction ID: 655  
lifetime: 3600 secs

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# DHCP: cung cấp nhiều thông tin

DHCP không chỉ trả về địa chỉ IP được chỉ định trên subnet, mà nó còn có thể trả về nhiều thông tin như sau:

- Địa chỉ của router ở cửa ngõ kết nối ra ngoài mạng của client (default gateway)
- Tên và địa chỉ IP của DNS sever
- network mask (cho biết phần của mạng và phần host của địa chỉ IP)

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

# DHCP: ví dụ

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

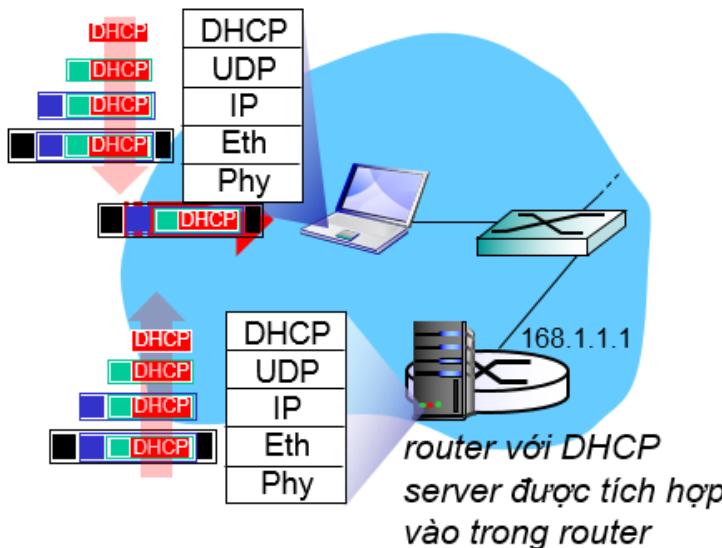
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

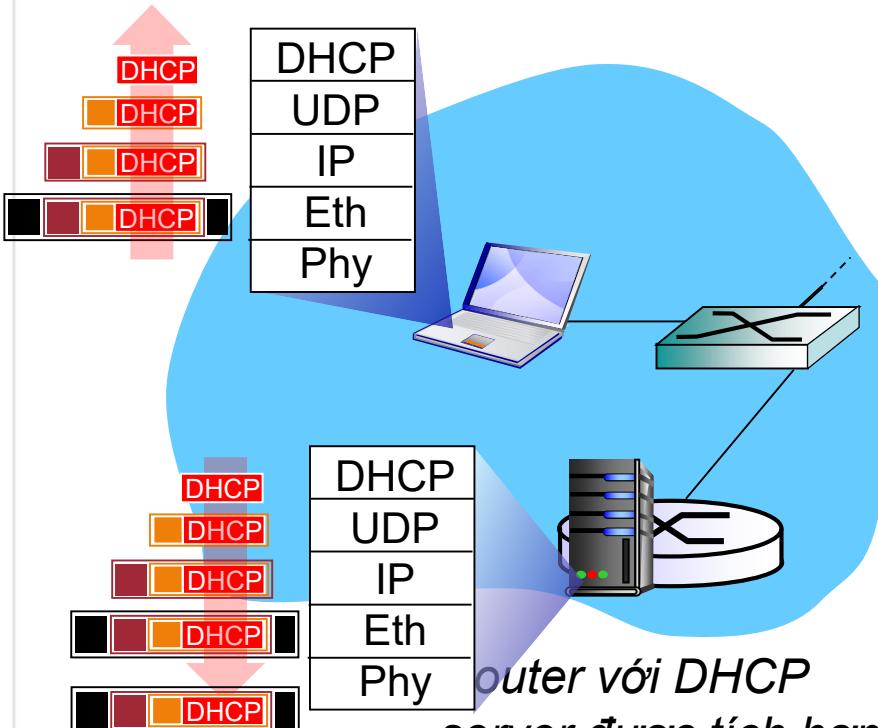
- Laptop tham gia vào mạng cần địa chỉ IP của nó, địa chỉ của default gateway, địa chỉ của DNS server: dùng DHCP



- DHCP request được đóng gói trong gói tin UDP, rồi được đóng gói trong gói IP, rồi được đóng gói trong gói 802.1 Ethernet
- Gói tin Ethernet frame được gửi quảng bá (đích: FFFFFFFFFFFF) trên LAN, được nhận tại router đang chạy DHCP server
- Ethernet gửi gói lên IP, rồi lên UDP, rồi lên DHCP



# DHCP: ví dụ



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- DHCP server tạo gói DHCP ACK chứa địa chỉ IP của client, địa chỉ IP của default gateway, tên và địa chỉ IP của DNS server
- ❖ Thông tin được đóng gói của DHCP server, là gói tin được chuyển cho DHCP client
- ❖ Bây giờ, client biết địa chỉ IP của nó, tên và địa chỉ IP của DNS server, địa chỉ IP của default gateway



# DHCP: Wireshark output (home LAN)

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

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

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."**

Trả lời



# Địa chỉ IP: làm sao để lấy được 1 địa chỉ IP?

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

**Hỏi:** làm sao mạng lấy được phần subnet của địa chỉ IP?

**Đáp:** lấy phần đã được cấp phát của không gian địa chỉ IP do ISP cung cấp

Dãy địa chỉ của ISP  
200.23.16.0/20

11001000 00010111 00010000 00000000

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



## Định địa chỉ phân cấp: định tuyến tích hợp

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Gán địa chỉ theo dạng phân cấp cho phép quảng bá hiệu quả thông tin định tuyến

Tổ chức 0

200.23.16.0/23

Tổ chức 1

200.23.18.0/23

Tổ chức 2

200.23.20.0/23

Tổ chức 7

200.23.30.0/23

Fly-By-Night-ISP

“gởi cho tôi bắt cứ thông tin gì  
với các địa chỉ bắt đầu  
200.23.16.0/20”

Internet

ISPs-R-Us

“gởi cho tôi bắt cứ thông  
tin gì với các địa chỉ bắt đầu  
199.31.0.0/16”



# Định địa chỉ phân cấp: các đường đi cụ thể hơn

ISPs-R-Us có 1 đường đi cụ thể hơn tới Tổ chức 1

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Tổ chức 0

200.23.16.0/23

Tổ chức 2

200.23.20.0/23

Tổ chức 7

200.23.30.0/23

Tổ chức 1

200.23.18.0/23

Fly-By-Night-ISP

ISPs-R-Us

“gởi cho tôi bất cứ thông tin gì  
với các địa chỉ bắt đầu  
200.23.16.0/20”

Internet

“gởi cho tôi bất cứ thông tin  
gi với các địa chỉ bắt đầu  
199.31.0.0/16  
hoặc 200.23.18.0/23”



# Định địa chỉ IP: lời cuối...

**Q:** làm cách nào mà một ISP lấy được khối địa chỉ?

**A:** ICANN: Internet Corporation for Assigned

Names and Numbers <http://www.icann.org/>

- Cấp phát địa chỉ
- Quản lý DNS
- Gán các tên miền, giải quyết tranh chấp

Giới thiệu

Virtual circuit network and diagram network

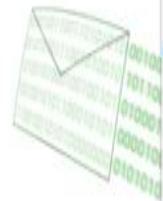
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# NAT: network address translation

Giới thiệu

Virtual circuit network and diagram network

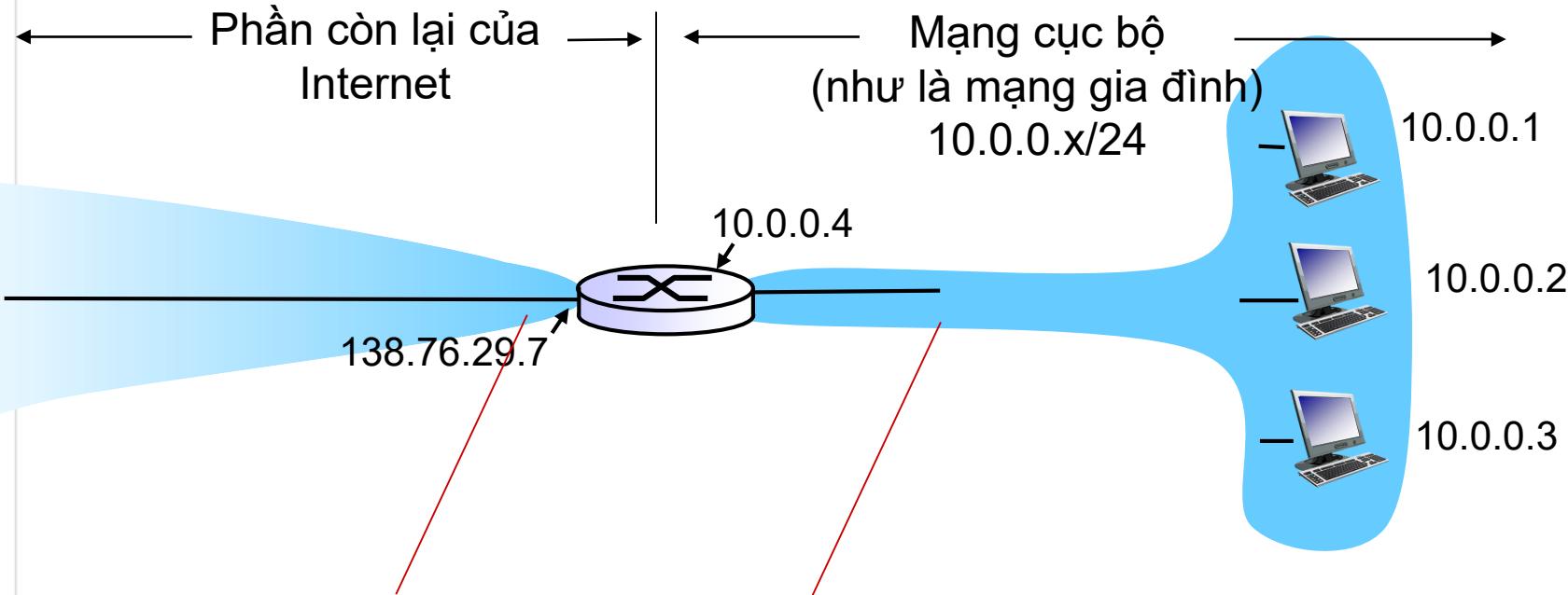
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



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

Các datagram với nguồn hoặc đích trong mạng này có địa chỉ 10.0.0.x/24 cho nguồn, đích (như thông thường)



# NAT: network address translation

**Động lực:** mạng cục bộ chỉ dùng 1 địa chỉ IP đối với thế giới bên ngoài:

- Không cần thiết dùng 1 vùng địa chỉ từ ISP: chỉ cần 1 địa chỉ IP cho tất cả các thiết bị
- Có thể thay đổi các địa chỉ IP của các thiết bị trong mạng cục bộ mà không cần thông báo cho thế giới bên ngoài
- Có thể thay đổi ISP mà không cần thay đổi địa chỉ IP của các thiết bị trong mạng nội bộ
- Bên ngoài không nhìn thấy và không biết địa chỉ rõ ràng của các thiết bị bên trong mạng cục bộ (tăng cường bảo mật)

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# NAT: network address translation

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

*Triển khai:* NAT router phải:

- *Với các datagram đi ra: thay thế* (địa chỉ IP nguồn, số hiệu cổng nguồn) của mọi datagram đi ra bên ngoài bằng (địa chỉ IP NAT, số hiệu port mới)  
... Các client/server ở xa sẽ dùng địa chỉ đó (địa chỉ IP NAT, số hiệu port mới) như là địa chỉ đích
- *Ghi nhớ (trong bảng chuyển đổi NAT)* mọi cặp chuyển đổi (địa chỉ IP nguồn, số hiệu port) sang (địa chỉ IP NAT, số hiệu port mới)
- *Với các datagram đi đến: thay thế* (địa chỉ IP NAT, số hiệu port mới) trong các trường đích của mọi datagram đến với giá trị tương ứng (địa chỉ IP và số hiệu cổng nguồn) trong bảng NAT



# NAT: network address translation

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

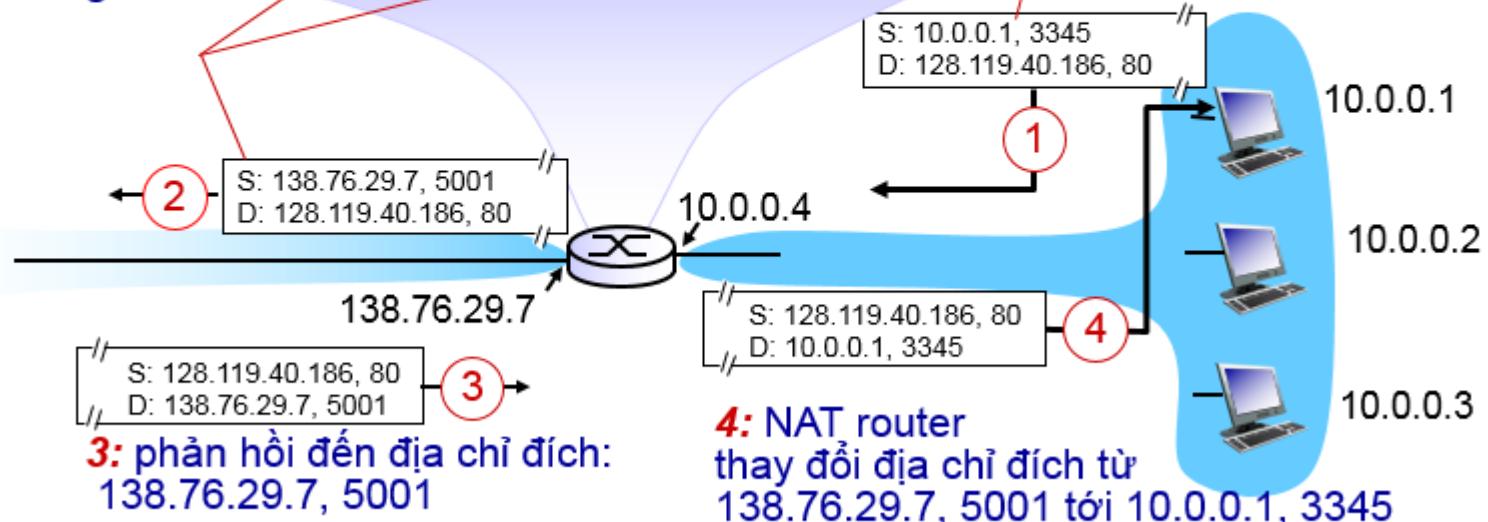
Thuật toán routing

Broadcast và multicast routing

Routing trong internet

**2:** NAT router  
thay đổi địa chỉ nguồn  
từ 10.0.0.1, 3345 đến  
138.76.29.7, 5001,  
cập nhật bảng

Bảng chuyển đổi NAT	
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
.....	.....



# NAT: network address translation

- Trường số hiệu port 16-bit:
  - Hỗ trợ hơn 60,000 kết nối đồng thời với một địa chỉ phía LAN!
- NAT gây ra tranh luận:
  - Các router chỉ nên xử lý đến tầng 3
  - Vi phạm thỏa thuận end-to-end
    - Những nhà thiết kế ứng dụng phải tính đến khả năng của NAT, ví dụ ứng dụng P2P
  - Việc thiếu địa chỉ IP sẽ được giải quyết khi dùng IPv6

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Vấn đề chuyển gói NAT

- Các client muốn kết nối tới server có địa chỉ 10.0.0.1
  - Địa chỉ 10.0.0.1 của server ở trong client mạng LAN (client không thể dùng nó như là địa chỉ IP đích)
  - Chỉ có 1 địa chỉ có thể được nhìn thấy từ bên ngoài địa chỉ được NAT: 138.76.29.7
- Giải pháp 1:* cấu hình NAT tĩnh để chuyển các yêu cầu kết nối đến tại port được cho trước tới server
  - Ví dụ (123.76.29.7, port 2500) luôn luôn được chuyển tới 10.0.0.1 port 25000

Giới thiệu

Virtual circuit network and diagram network

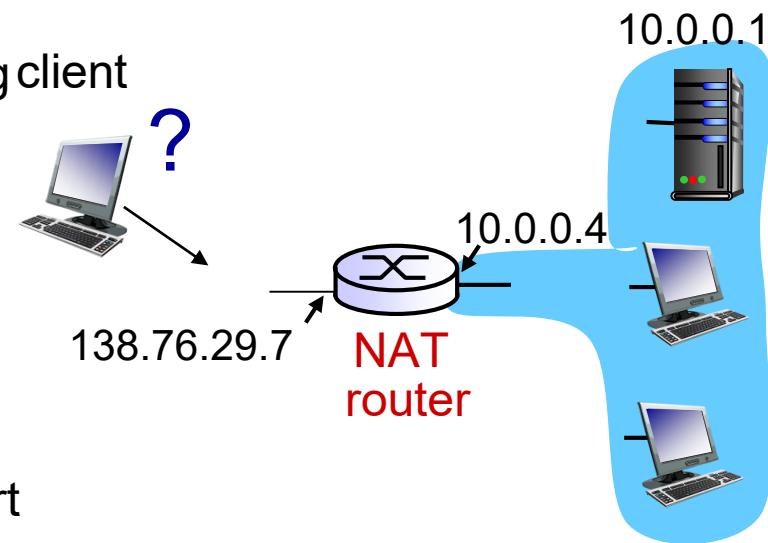
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Vấn đề chuyển gói NAT

- Giải pháp 2:* Giao thức Universal Plug and Play (UPnP) trên Internet Gateway Device (IGD). Cho phép host được NAT:
  - ❖ Học địa chỉ IP public (138.76.29.7)
  - ❖ thêm/gỡ bỏ các port mapping (thời gian thuê)

Giới thiệu

Virtual circuit network and diagram network

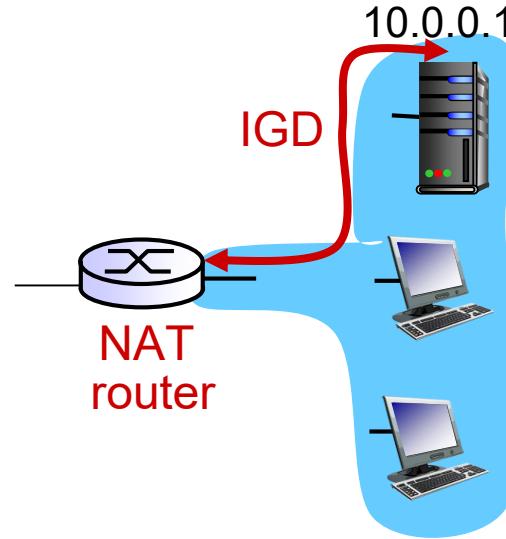
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

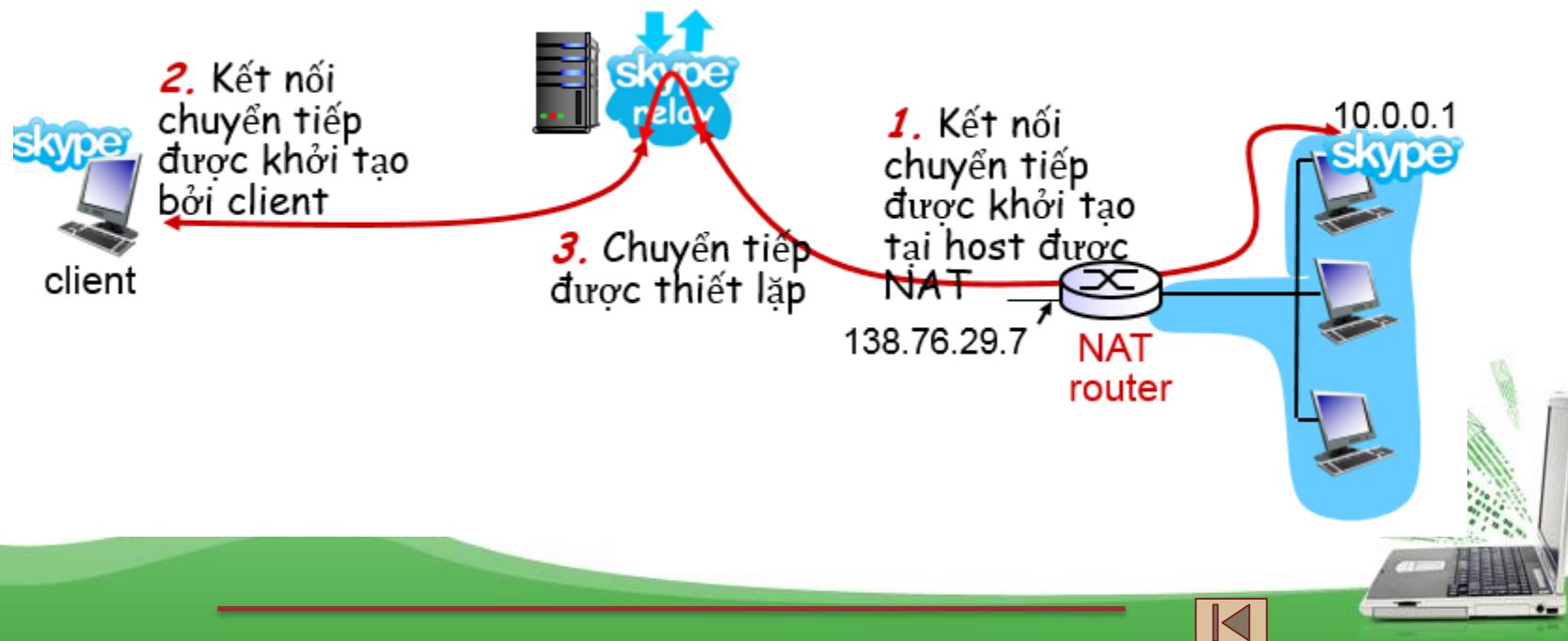
Broadcast và multicast routing

Routing trong internet



# Vấn đề chuyển gói NAT

- *Giải pháp 3:* chuyển tiếp (relaying) (được sử dụng trong Skype)
  - Client được NAT thiết lập kết nối để chuyển tiếp
  - client bên ngoài kết nối để chuyển tiếp
  - Sự chuyển tiếp này bắc cầu các packet giữa các kết nối



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 virtual circuit network (Mạng mạch ảo) và datagram network (Mạng chuyển gói)

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- **ICMP**
- **IPv6**

## Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

**Ip:internet protocol**

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## 4.5 các thuật toán routing

- link state
- distance vector
- hierarchical routing

## 4.6 routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 broadcast và multicast routing



# ICMP: Internet Control Message Protocol

- Được sử dụng bởi các host và router để truyền thông tin tầng Mạng
  - Thông báo: host, network, port, giao thức không có thực
  - Các dạng gói tin echo request/reply (được dùng bởi ping)
- Tầng Mạng “trên” IP:
  - Các thông điệp ICMP được gửi trong các IP datagram
- Thông điệp ICMP:** loại, mã cộng thêm 8 byte đầu tiên của IP datagram gây ra lỗi

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

<u>Loại</u>	<u>mã</u>	<u>Mô tả</u>
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



# Traceroute và ICMP

- Nguồn gửi một chuỗi các segment UDP đến đích
  - Cái đầu tiên có TTL =1
  - Cái thứ 2 có TTL=2, tương tự.
  - Không quan tâm số port
- Khi datagram thứ n đến router thứ n:
  - router hủy datagram
  - Và gửi đến nguồn một thông điệp ICMP (loại 11, mã 0)
  - Thông điệp ICMP bao gồm tên và địa chỉ IP của router

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

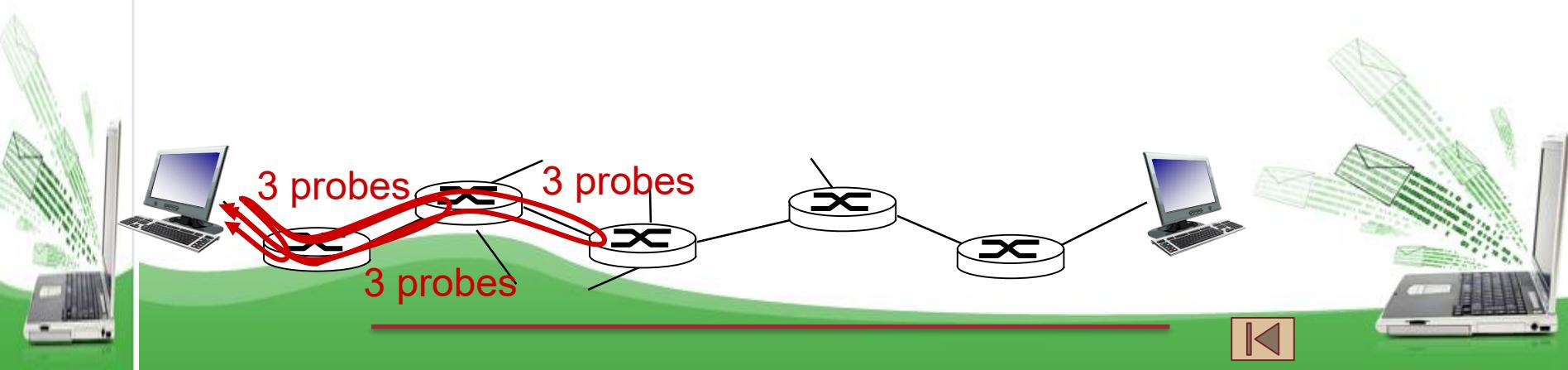
Broadcast và multicast routing

Routing trong internet

- ❖ Khi thông điệp ICMP đến, nguồn ghi lại RTTs

## Tiêu chuẩn dùng:

- ❖ Gói UDP đến host đích
- ❖ Đích trả về thông điệp ICMP “port không có thực”(loại 3, mã 3)
- ❖ Nguồn dùng



# IPv6: Động lực

- *Động lực thúc đẩy ban đầu:* không gian địa chỉ 32-bit sớm được cấp phát cạn kiệt.
- Động lực khác:
  - Cần định dạng của header giúp tăng tốc xử lý/chuyển gói
  - Cần thay đổi header để tạo điều kiện thuận lợi cho QoS

## *Định dạng datagram IPv6 :*

- Header có độ dài cố định 40 byte
- Không cho phép phân mảnh

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Định dạng datagram IPv6

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

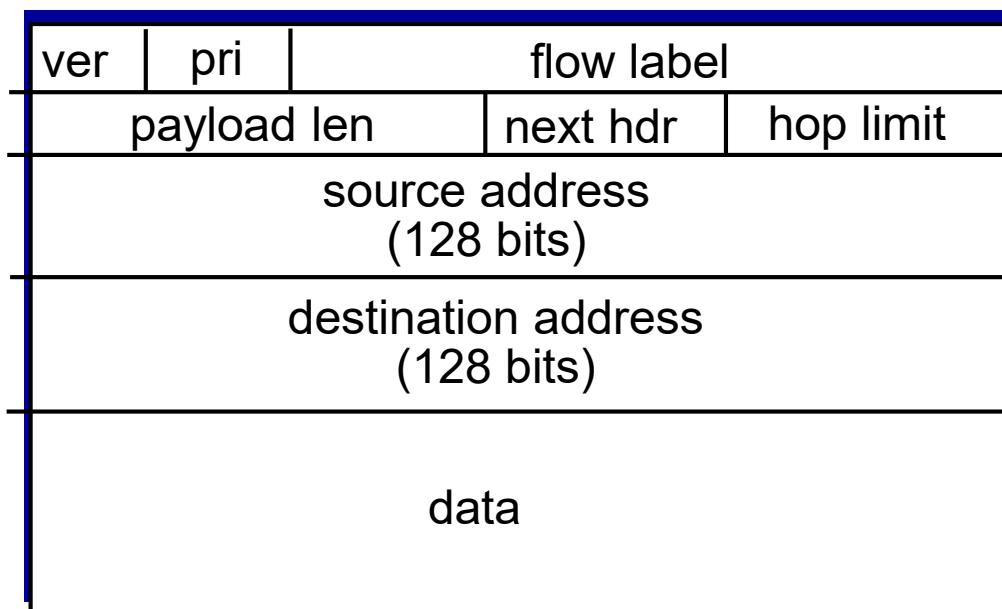
Routing trong internet

**Độ ưu tiên (priority):** xác định độ ưu tiên của các datagram trong luồng

**Nhãn luồng (flow Label):** xác định các datagram trong cùng “luồng”.

(khái niệm “luồng” không được định nghĩa rõ ràng).

**Header kế tiếp (next header):** xác định giao thức Tầng trên cho dữ liệu.



32 bits



# Những thay đổi khác so với IPv4

- **Checksum:** được bỏ toàn bộ để giảm thời gian xử lý tại mỗi hop
- **Options:** được cho phép, nhưng nằm ở ngoài header, được chỉ ra bởi trường “Next Header”
- **ICMPv6:** phiên bản mới của ICMP
  - Các kiểu thông điệp bổ sung. Ví dụ “Packet Too Big”
  - Các chức năng quản lý nhóm multicast

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

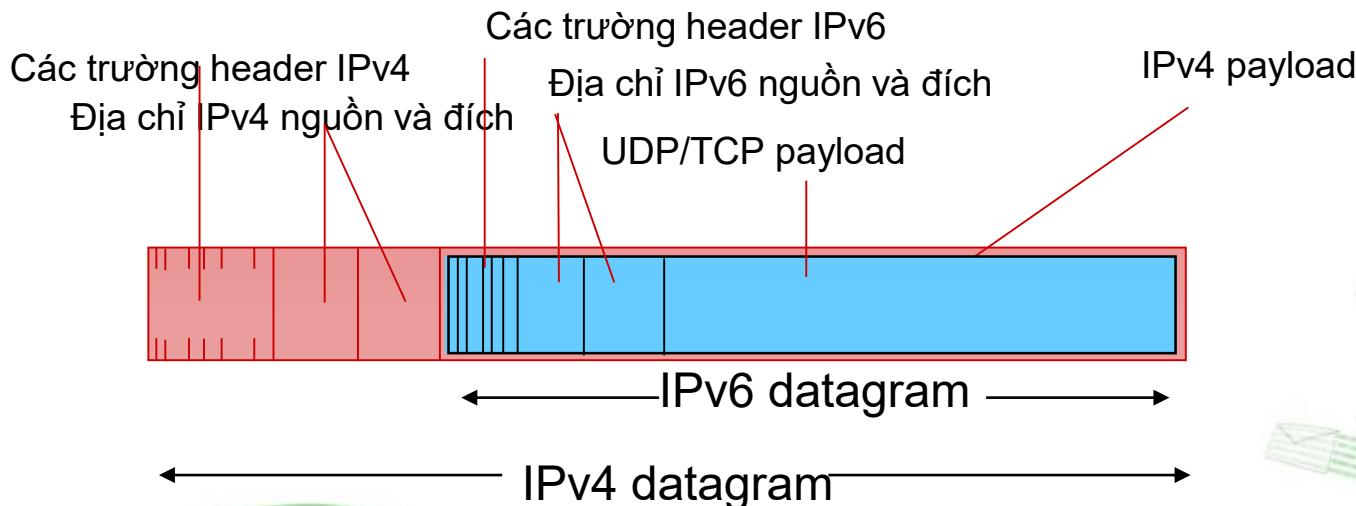
Thuật toán routing

Broadcast và multicast routing

Routing trong internet

# Chuyển từ IPv4 sang IPv6

- Không phải tất cả router đều có thể được nâng cấp đồng thời
  - Không có “ngày chuyển đổi”
  - Mạng sẽ hoạt động như thế nào với các router dùng cả IPv4 và IPv6?
- tunneling:* datagram IPv6 được đóng gói vào thân của datagram IPv4 và được chuyển đi giữa các router IPv4



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Tunneling

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

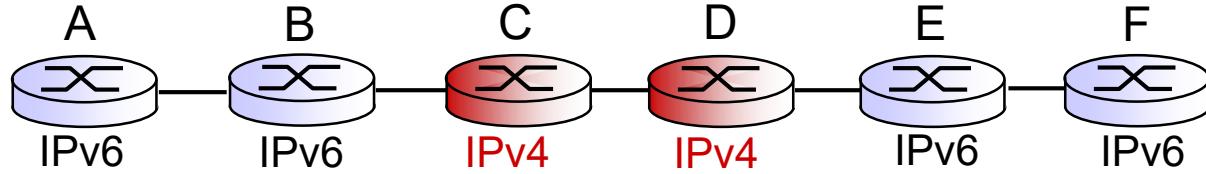
Broadcast và multicast routing

Routing trong internet

Cách nhìn logic:



Cách nhìn thực:



# Tunneling

Giới thiệu

Virtual circuit network and diagram network

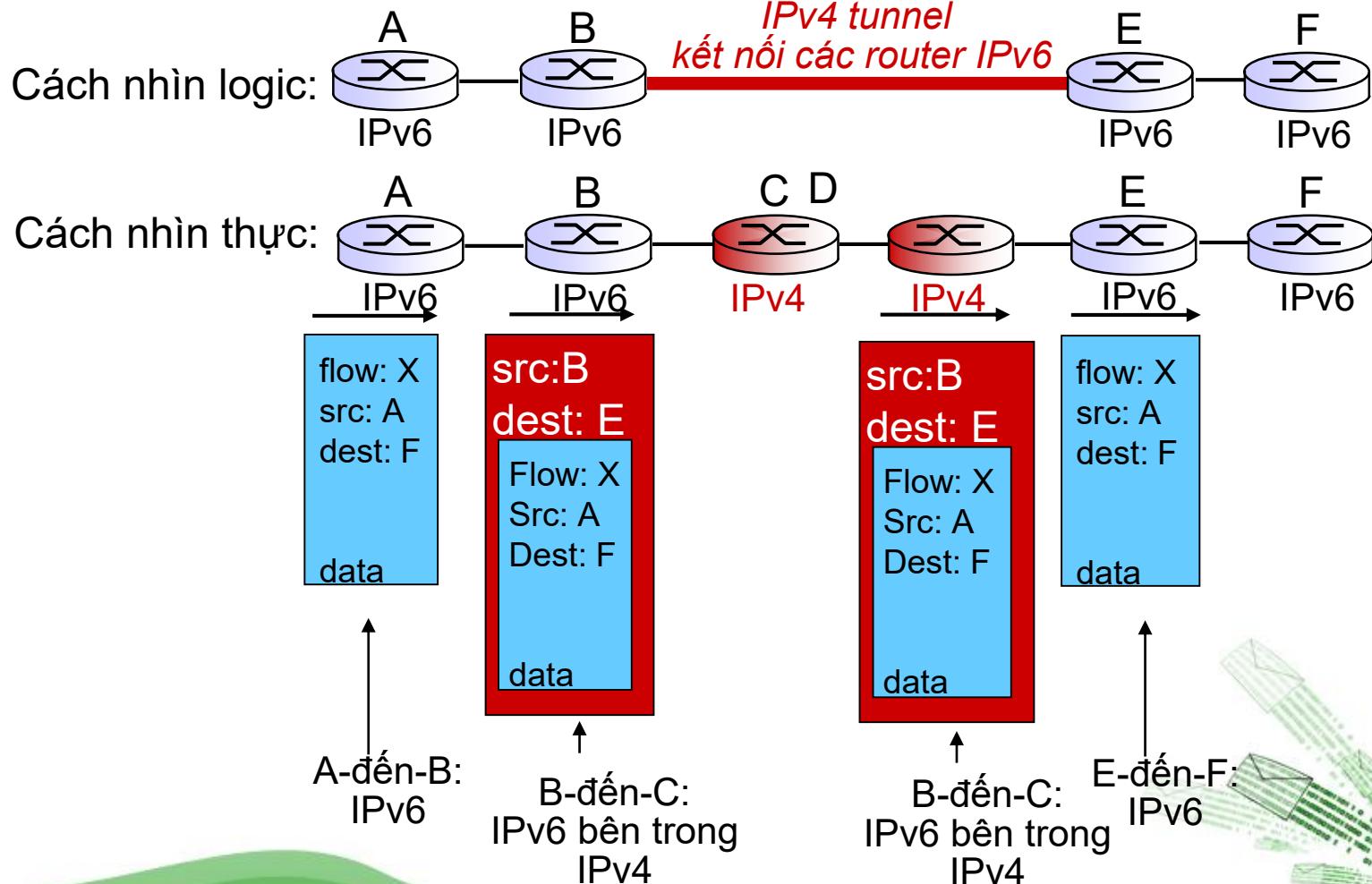
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Chương 4: Nội dung

4.1 Giới thiệu

4.2 virtual circuit network  
(Mạng mạch ảo) và  
datagram network  
(Mạng chuyển gói)

4.3 Cấu trúc bên trong  
router

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

4.5 các thuật toán routing

- link state
- distance vector

4.6 routing trong Internet

- RIP
- OSPF
- BGP



# Tác động lẫn nhau giữa routing và forwarding

Giới thiệu

Virtual circuit network and diagram network

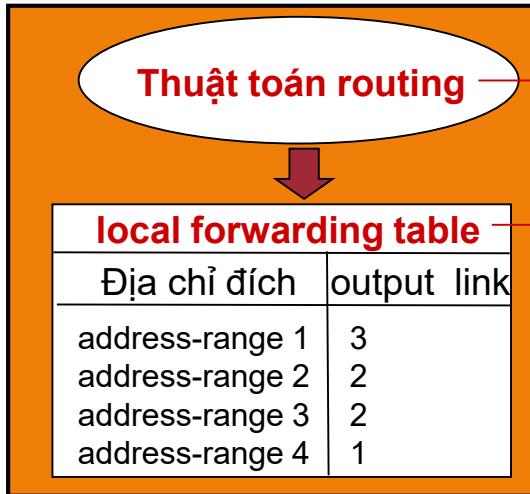
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

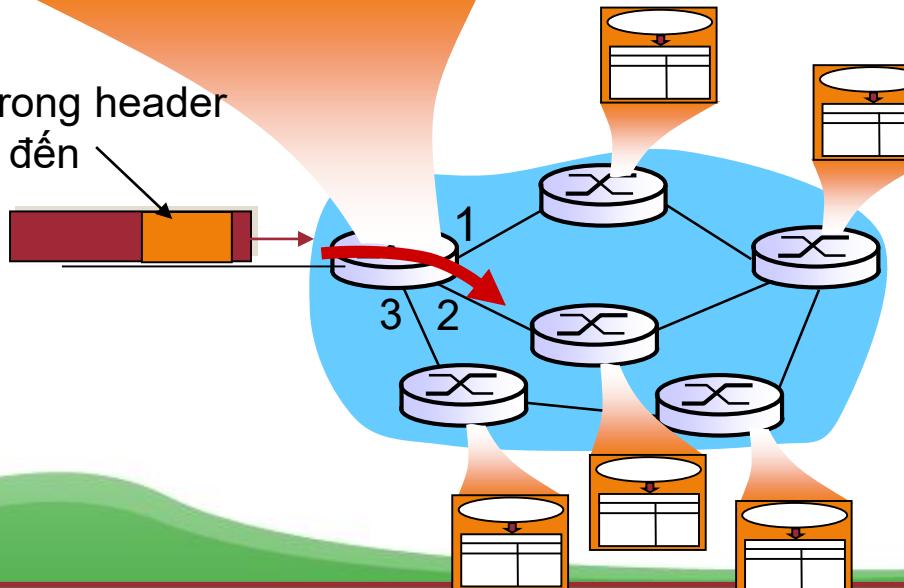
Routing trong internet



Thuật toán routing xác định đường đi qua mạng từ đầu cuối này đến đầu cuối khác

Bảng forwarding xác định chuyển gói trong cục bộ của router này

Địa chỉ IP trong header  
của packet đến



# Mô hình đồ thị

Giới thiệu

Virtual circuit network and diagram network

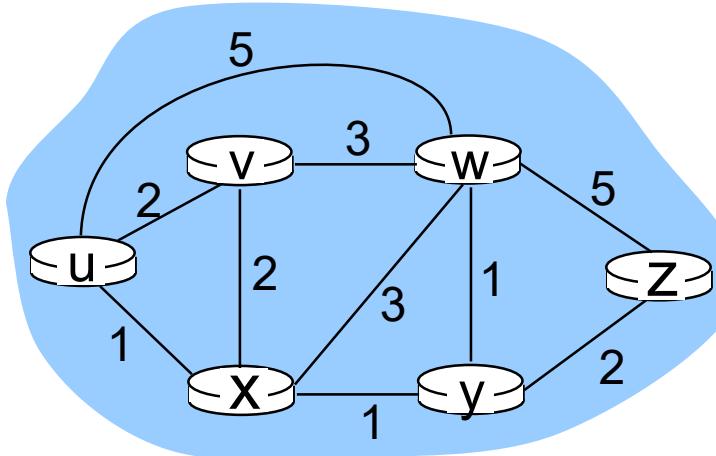
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



Đồ thị:  $G = (N, E)$

$N = \text{tập hợp các router} = \{ u, v, w, x, y, z \}$

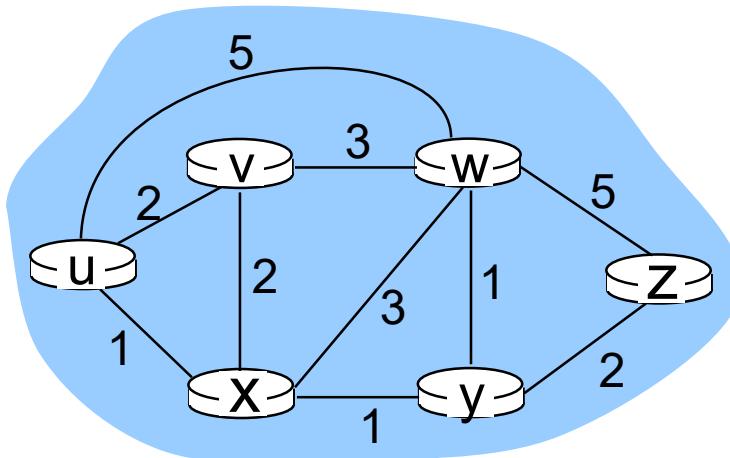
$E = \text{tập hợp các kết nối} = \{ (u, v), (u, x), (v, x), (v, w), (w, x), (x, y), (w, y), (w, z), (y, z) \}$

*Ghi chú:* mô hình đồ thị cũng hữu ích trong các ngũ cảnh khác.

Ví dụ, P2P, trong đó  $N$  là tập các peer và  $E$  là tập các kết nối TCP



# Mô hình đồ thị: Chi phí



Chi phí 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 đường đi thấp nhất từ u tới z?

*Thuật toán routing:* thuật toán tìm đường có chi phí thấp nhất

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

$c(x,x')$  = chi phí kết nối  $(x,x')$

VD:  $c(w,z) = 5$

Chi phí có thể luôn luôn là 1, hoặc liên quan đến băng thông, hoặc liên quan đến tắc nghẽn



# Phân loại thuật toán Routing

*Q: thông tin toàn cục hay phân cấp  
(global or decentralized)?*

*Toàn cục:*

- Tất cả các router có toàn bộ thông tin về chi phí kết nối, cấu trúc toàn mạng
- Thuật toán “link state”

*Phân cấp:*

- Router biết các router được kết nối vật lý trực tiếp với nó (neighbor), và chi phí kết nối đến neighbor đó
- Lặp lại qua trình tính toán, trao đổi thông tin với các neighbor
- Thuật toán “distance vector”

*Giới thiệu*

[Virtual circuit network and diagram network](#)

[Cấu trúc trong router](#)

[Ip:internet protocol](#)

[Thuật toán routing](#)

[Broadcast và multicast routing](#)

[Routing trong internet](#)

*Q: tĩnh hay động?*

*Tĩnh:*

- Các đường đi được cập nhật chậm theo thời gian

*Động:*

- Các đường đi thay đổi nhanh
  - Cập nhật theo chu kỳ
  - Cập nhật khi có những thay đổi về chi phí kết nối



# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 virtual circuit network

(Mạng mạch ảo) và  
datagram network  
(Mạng chuyển gói)

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

### Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## 4.5 các thuật toán routing

- link state
- distance vector
- hierarchical routing

## 4.6 routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 broadcast và multicast routing



# Thuật toán routing Link-State

## Thuật toán Dijkstra

- Tất cả các node (router/đỉnh trên đồ thị) đều biết chi phí kết nối, cấu trúc mạng
  - Được thực hiện thông qua gửi quảng bá các trạng thái kết nối (“link state broadcast”)
  - Tất cả các nodes có cùng thông tin với nhau
- Tính toán đường đi có chi phí thấp nhất từ 1 node (‘nguồn’) đến tất cả các node khác
  - Cho trước bảng *forwarding* của node đó
- Lặp lại: sau k lần lặp lại, biết được đường đi có chi phí thấp nhất tới k đích

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## Ký hiệu:

- $c(x,y)$ : chi phí kết nối từ node x đến y;  $c(x,y) = \infty$  nếu giữa x và y không có kết nối
- $D(v)$ : giá trị chi phí hiện tại của đường đi từ nguồn tới đích v
- $p(v)$ : node liền kề trước v nằm trên đường đi từ nguồn tới v
- $N'$ : tập các node mà chi phí đường đi thấp nhất đã được xác định



# Thuật toán Dijkstra

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

1 **Khởi tạo:**

2  $N' = \{u\}$

3 for tất cả các v

4 if v liền kề với u

5 then  $D(v) = c(u,v)$

6 else  $D(v) = \infty$

7

8 **Lặp**

9 tìm w chưa tồn tại trong  $N'$  có  $D(w)$  nhỏ nhất

10 thêm w vào tập  $N'$

11 cập nhật lại  $D(v)$  cho tất cả v kề với w và không có trong  $N'$ :

12  **$D(v) = \min(D(v), D(w) + c(w,v))$**

13/\* chi phí mới đến v là chính nó hoặc chi phí đường đi ngắn nhất

14 cộng với chi phí từ w đến v\*/

15 **Cho đến khi tất cả các node trong  $N'$**



# Thuật toán Dijkstra : ví dụ

## Giới thiệu

Virtual circuit network and diagram network

## Cấu trúc trong router

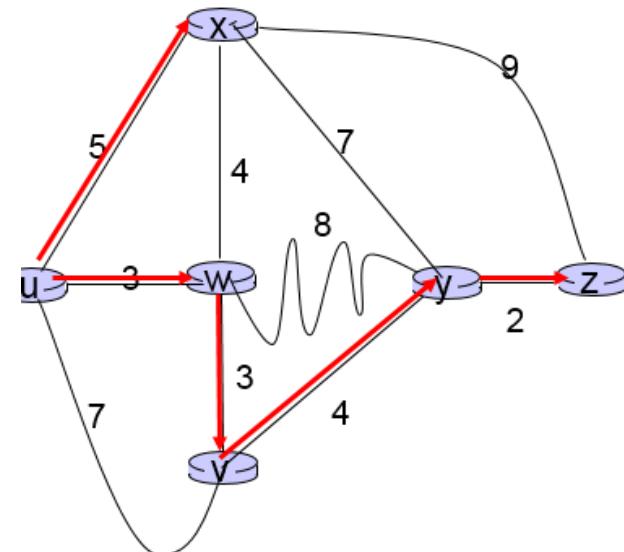
## Ip:internet protocol

## **Thuật toán routing**

---

### **Broadcast và `multicast routing**

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



## *Ghi chú:*

- ❖ Xây dựng cây đường đi ngắn nhất bằng cách lần theo các node liền kề trước đó
  - ❖ Các đường có chi phí bằng nhau có thể tồn tại (có thể bị chia tùy tiện)



# Thuật toán Dijkstra: ví dụ

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

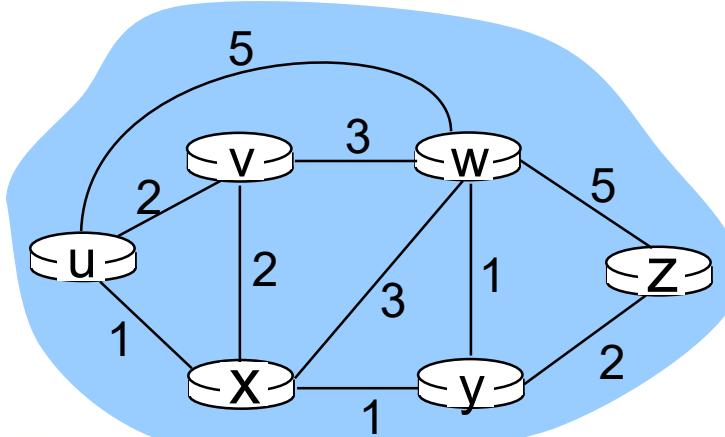
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

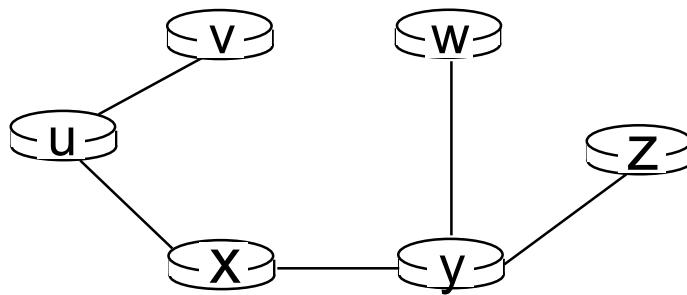
Routing trong internet

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	$\infty$	$\infty$
1	ux	2,u	4,x		2,x	$\infty$
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



# Thuật toán Dijkstra: ví dụ (2)

Kết quả cây đường đi ngắn nhất từ u:



Kết quả bảng forwarding trong u:

Đích đến	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



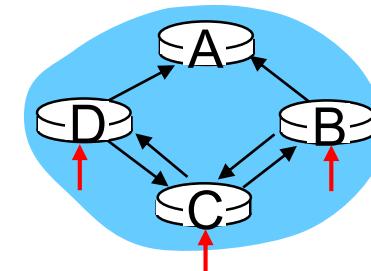
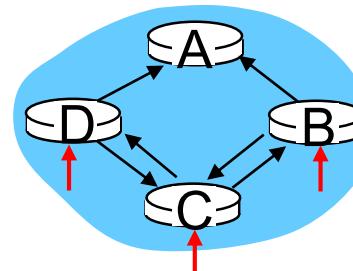
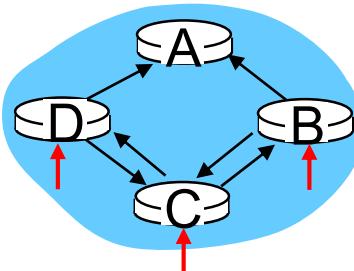
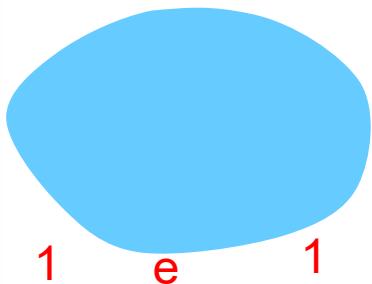
# Thuật toán Dijkstra, thảo luận

*Độ phức tạp của thuật toán:* n nodes

- ❖ Mỗi lần duyệt: cần kiểm tra tất cả các node, w, không có trong N
- ❖  $n(n+1)/2$  phép so sánh:  $O(n^2)$
- ❖ Có nhiều cách thực hiện hiệu quả hơn:  $O(n \log n)$

*Có thể dao động:*

- ❖ Ví dụ: chi phí kết nối bằng với số lượng lưu thông được mang



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 virtual circuit network

(Mạng mạch ảo) và  
datagram network  
(Mạng chuyển gói)

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## 4.5 các thuật toán routing

- link state
- distance vector
- hierarchical routing

## 4.6 routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 broadcast và multicast routing



# Thuật toán Distance vector

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

*Công thức Bellman-Ford (dynamic programming)*

cho

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

thì

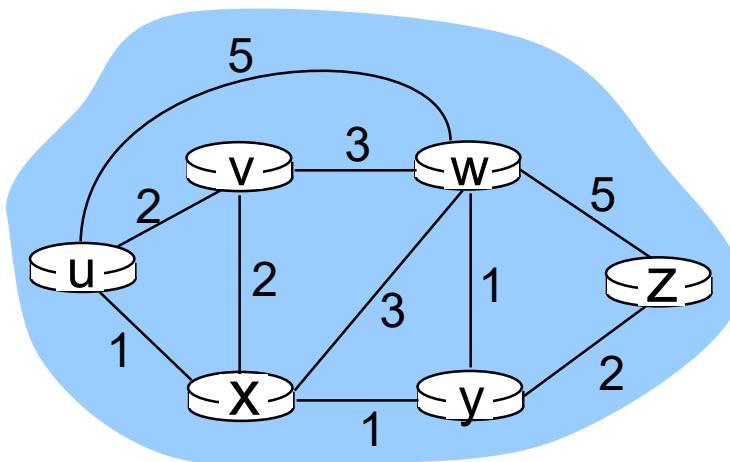
$$d_x(y) = \min \{ c(x, v) + d_v(y) \}$$

v | | |  
Chi phí từ neighbor v tới đích y  
Chi phí tới neighbor v

$\min$  được thực hiện trên tất cả các neighbor v của x



# Bellman-Ford ví dụ



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Rõ ràng,  $d_v(z) = 5$ ,  $d_x(z) = 3$ ,  $d_w(z) = 3$ 

Biểu thức B-F cho kết quả:

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

node có giá trị tối thiểu là trạm (hop) kế tiếp trong đường đi ngắn nhất, được sử dụng trong bảng forwarding



# Thuật toán Distance vector

- $D_x(y) =$  ước lượng chi phí thấp nhất từ x đến y
  - x lưu lại distance vector  $\mathbf{D}_x = [D_x(y): y \in N]$
- node x:
  - Biết chi phí đến mỗi neighbor v:  $c(x,v)$
  - Lưu lại distance vectors của các neighbor của nó. Cho mỗi neighbor v, x lưu lại $\mathbf{D}_v = [D_v(y): y \in N]$

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

# Thuật toán Distance vector

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## Ý tưởng chính:

- Mỗi node định kỳ gửi ước lượng distance vector của nó cho các neighbor
- Khi x nhận ước lượng DV mới từ neighbor, nó cập nhật DV cũ của nó dùng công thức B-F:

$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \text{ với mỗi node } y \in N$$

- ❖ Dưới những điều kiện tự nhiên, giá trị ước lượng  $D_x(y)$  sẽ hội tụ tới chi phí thực sự nhỏ nhất  $d_x(y)$



# Thuật toán Distance vector

*Lặp, không đồng bộ:* mỗi lặp cục bộ tại 1 node được gây ra bởi:

- Chi phí kết nối cục bộ thay đổi
- Nhận được DV thông báo cập nhật từ node lân cận (neighbor)

*Phân bố:*

- Mỗi node thông báo đến các node lân cận *chỉ khi* DV của nó thay đổi
  - Các node lân cận sau đó thông báo đến các node lân cận của nó nếu cần thiết

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## Mỗi node:

*Chờ cho* (thay đổi trong chi phí link cục bộ hoặc thông điệp từ neighbor)

*Tính toán lại* các ước lượng

Nếu DV đến đích bất kỳ vừa thay đổi, thì thông báo neighbors



# Thuật toán Distance vector

## Giới thiệu

Virtual circuit network and diagram network

## Cấu trúc trong router

## Ip:internet protocol

## Thuật toán routing

## Broadcast va`multicast routing

## Routing trong internet

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} \\ = \min\{2+0, 7+1\} = 2$$

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

## Bảng node x

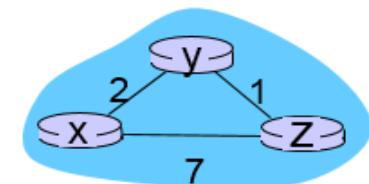
		Chi phí đến		
		x	y	z
Từ	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

## Bảng node y

	x	y	z
x	$\infty$	$\infty$	$\infty$
y	2	0	1
z	$\infty$	$\infty$	$\infty$

## Bảng node z

		Chi phí đến		
		x	y	z
Tùy	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0



#### Thời gian



# Thuật toán Distance vector

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

$$\begin{aligned} D_x(y) &= \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} \\ &= \min\{2+0, 7+1\} = 2 \end{aligned}$$

$$\begin{aligned} D_x(z) &= \min\{c(x,y) + \\ &\quad D_y(z), c(x,z) + D_z(z)\} \\ &= \min\{2+1, 7+0\} = 3 \end{aligned}$$

**Bảng node x**

	x	y	z
x	0	2	7
y	$\infty$	$\infty$	$\infty$
z	$\infty$	$\infty$	$\infty$

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

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

**Bảng node y**

	x	y	z
x	$\infty$	$\infty$	$\infty$
y	2	0	1
z	$\infty$	$\infty$	$\infty$

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

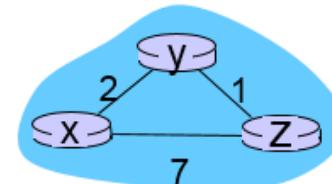
	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

**Bảng node z**

	x	y	z
x	$\infty$	$\infty$	$\infty$
y	$\infty$	$\infty$	$\infty$
z	7	1	0

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

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



Thời gian



# Distance vector: chi phí kết nối thay đổi

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

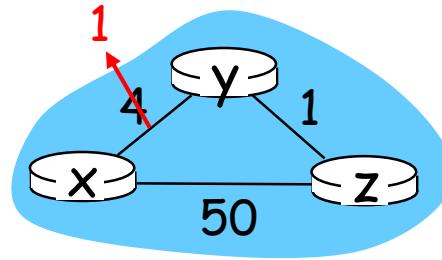
## Chi phí kết nối thay đổi:

- ❖ node phát hiện sự thay đổi chi phí kết nối cục bộ
- ❖ Cập nhật thông tin định tuyến, tính toán lại distance vector
- ❖ Nếu DV thay đổi, thì thông báo cho neighbor

$t_0$ : y phát hiện sự thay đổi chi phí kết nối, và cập nhật DV của nó, thông báo đến các neighbor của nó.

$t_1$ : z nhận được cập nhật từ y, và cập nhật bảng của nó, tính chi phí mới thấp nhất đến x, gởi DV của nó đến các neighbor của nó.

$t_2$ : y nhận được cập nhật của z, và cập nhật bảng distance của nó. Các chi phí thấp nhất của y không thay đổi, vì vậy y không gởi thông điệp đến z.



“tin  
tốt  
đi  
nhanh”



# Distance vector: Chi phí kết nối thay đổi

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

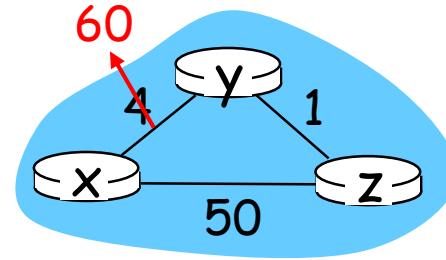
Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## Chi phí kết nối thay đổi:

- ❖ node phát hiện sự thay đổi chi phí kết nối cục bộ
- ❖ *Tin xấu đi chậm*- vấn đề “đếm đến vô tận”
- ❖ 44 lần duyệt trước khi thuật toán ổn định



## Đầu độc ngược (Poisoned reverse):

- ❖ Nếu Z đi qua Y để tới X :
  - Z nói với Y khoảng cách của nó đến X là không xác định (vì vậy Y sẽ không đi tới X thông qua Z)
- ❖ Điều này sẽ giải quyết được vấn đề đếm đến vô tận hay không?



# So sánh giữa thuật toán LS và DV

## Độ phức tạp của thông điệp

- **LS:** với  $n$  nodes,  $E$  kết nối, thì có  $O(nE)$  các thông điệp được gửi
- **DV:** chỉ trao đổi giữa các node lân cận với nhau
  - Thời gian hội tụ khác nhau

## Tốc độ hội tụ

- **LS:** thuật toán  $O(n^2)$  yêu cầu  $O(nE)$  thông điệp
  - Có thể có dao động
- **DV:** thời gian hội tụ khác nhau
  - Có thể định tuyến lặp
  - Vẫn đề đếm đến vô hạn (count-to-infinity)

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

**Sự linh hoạt:** điều gì xảy ra nếu router gặp sự cố?

**LS:**

- node có thể thông báo sai chi phí **kết nối**
- Mỗi node chỉ tính toán bảng riêng của nó

**DV:**

- DV node có thể quảng cáo sai về chi phí **đường đi**
- Bảng của mỗi node được sử dụng bởi các node khác
  - Lỗi bị lan truyền qua mạng



# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 virtual circuit network (Mạng mạch ảo) và datagram network (Mạng chuyển gói)

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## 4.5 các thuật toán routing

- link state
- distance vector
- **hierarchical routing**

## 4.6 routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 broadcast và multicast routing



# Định tuyến có cấu trúc (Hierarchical routing)

Cho tới đây, tìm hiểu về routing của chúng ta thực hiện trong môi trường lý tưởng hóa

- ❖ Tất cả các router là đồng nhất
- ❖ Mạng “phẳng”  
*... không đúng trong thực tế*

**Quy mô:** với 600 triệu đích đến:

- Không thể lưu trữ tất cả các đích đến trong các bảng định tuyến
- Việc trao đổi bảng định sẽ làm tràn ngập các liên kết!

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

**Quản trị**

- internet = mạng của các mạng
- Mỗi quản trị mạng có thể muốn điều hành việc định tuyến trong mạng riêng của họ



# Định tuyến có cấu trúc

- Các router được gom vào các vùng tự trị “autonomous systems” (AS)
- Các router trong cùng AS chạy cùng giao thức định tuyến với nhau
  - giao thức **định tuyến “intra-AS”**
  - Các router trong các AS khác nhau có thể chạy các giao thức định tuyến intra-AS khác nhau

## *Gateway router:*

- Tại “biên” (“edge”) của AS của nó
- Có liên kết đến router trong AS khác

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Kết nối các AS

❖ Bảng forwarding được cấu hình bởi cả thuật toán định tuyến intra- và inter-AS

- intra-AS thiết lập các mục cho các đích đến nội mạng
- inter-AS & intra-AS thiết lập các mục cho các đích đến ngoại mạng

Giới thiệu

Virtual circuit network and diagram network

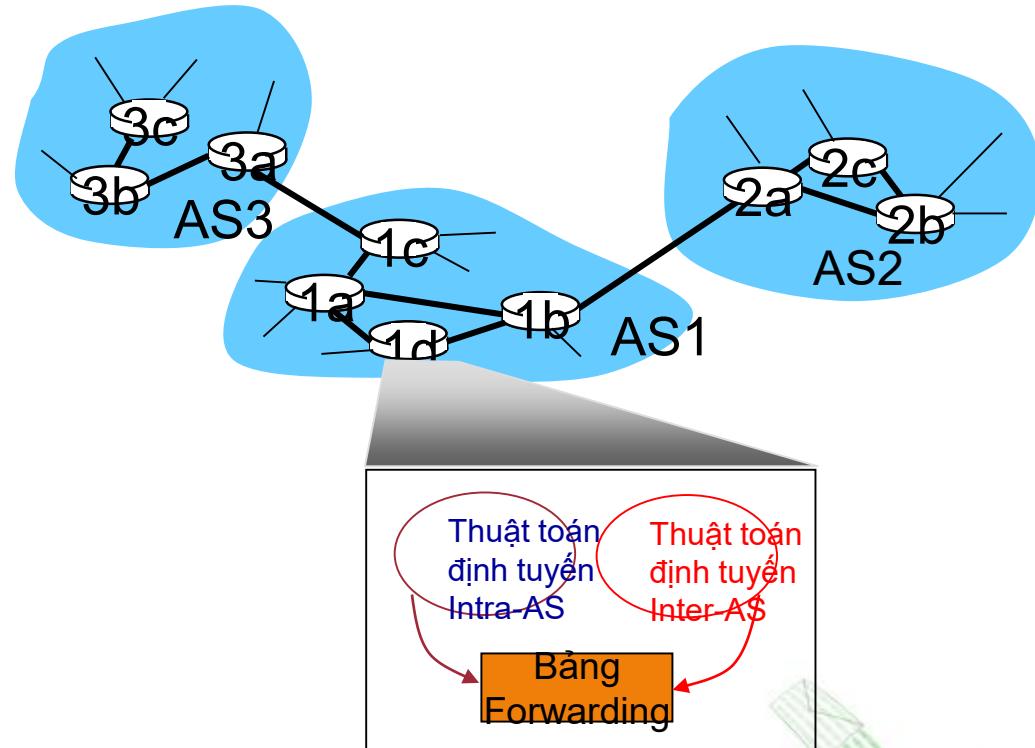
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

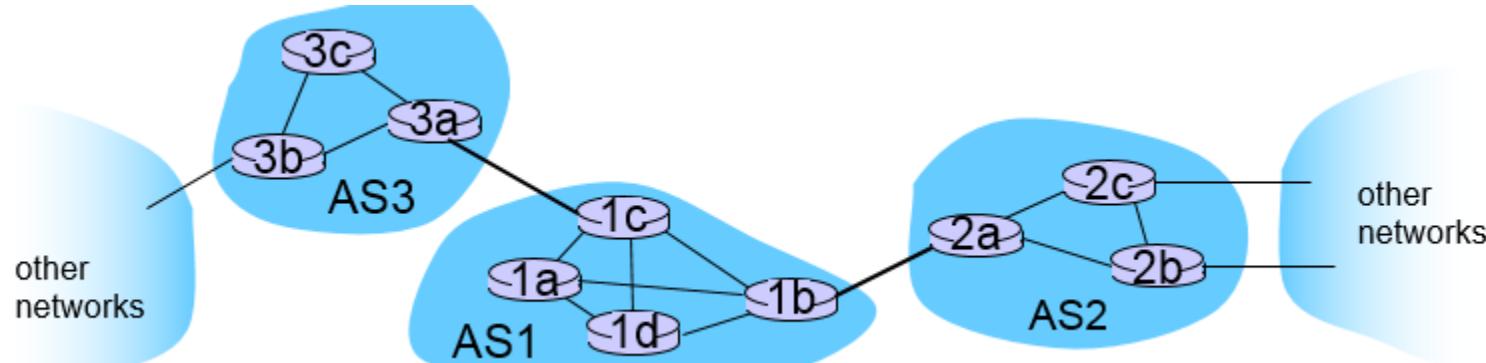
Broadcast và multicast routing

Routing trong internet



# Tác vụ Inter-AS

- ❖ Giả sử router trong AS1 nhận được datagram với đích đến nằm ngoài AS1:
    - router nên chuyển packet đến router gateway , nhưng là cái nào?



## Giới thiệu

Virtual circuit network and diagram network

## Cấu trúc trong router

## Ip:internet protocol

## Thuật toán routing

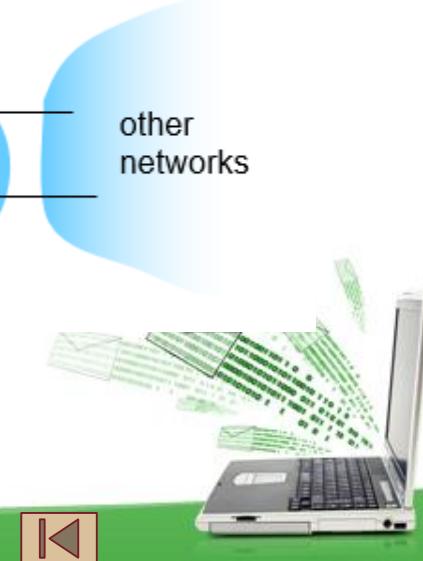
## Broadcast va`multicast routing

## Routing trong internet

# *AS1 phải*

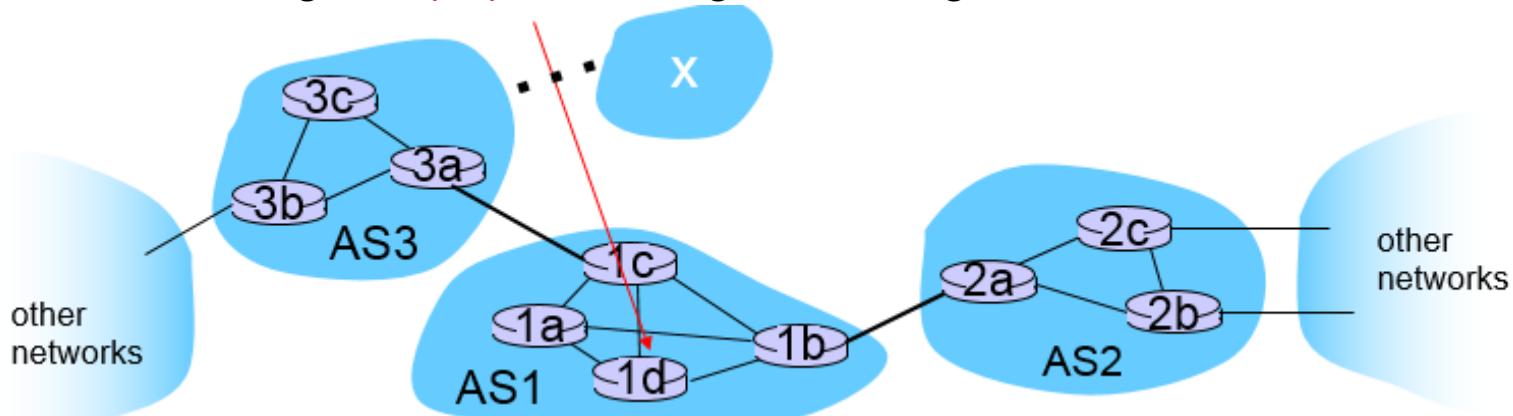
1. Học các đích đến nào có thể tới được thông qua AS2 và AS3
  2. Lan truyền thông tin này đến tất cả các router trong AS1

## *Công việc của định tuyến inter-AS!*



## Ví dụ: thiết lập bảng forwarding trong router 1d

- Giả sử AS1 học (through inter-AS) là có thể truyền tới subnet **X** thông qua AS3 (gateway 1c), nhưng không qua AS2
  - Giao thức inter-AS lan truyền thông tin này đến tất cả các router nội mạng
- router 1d xác định từ thông tin định tuyến intra-AS rằng interface **I** của nó nằm trên đường đi có chi phí thấp nhất tới 1c
  - Đưa giá trị (**x,I**) vào bảng forwarding



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Ví dụ: chọn giữa nhiều AS

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

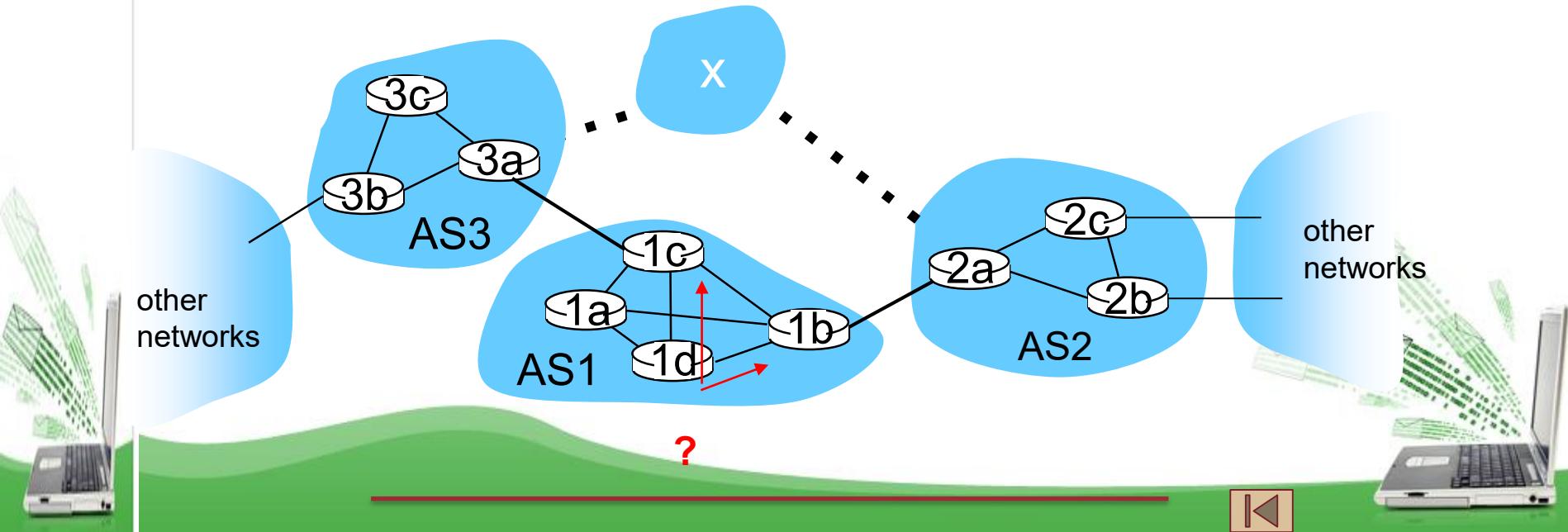
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

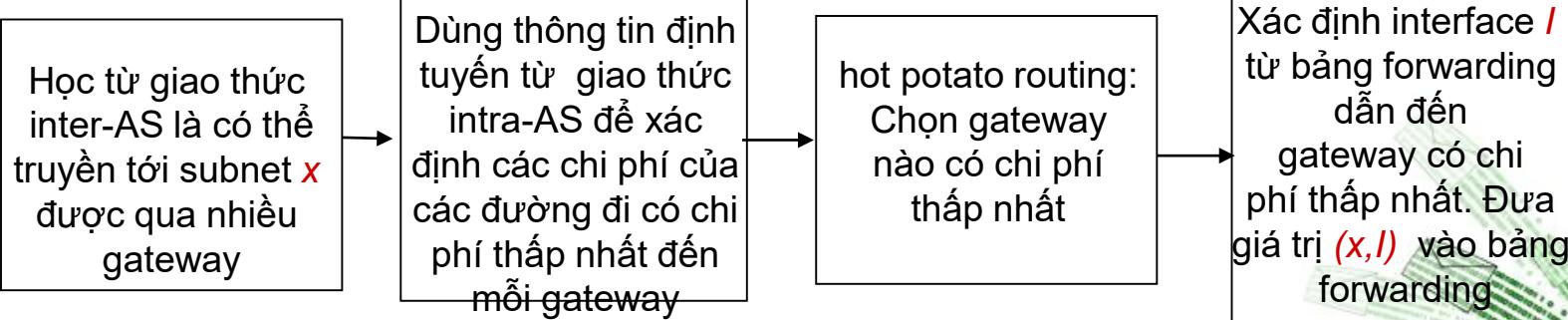
Routing trong internet

- Nay, giả sử AS1 học từ giao thức inter-AS là có thể truyền tới subnet **X** qua AS3 và qua AS2.
- Để cấu hình bảng forwarding, router 1d phải xác định gateway nào mà nó nên dùng để chuyển các packet đến đích **X**
  - Đây cũng là công việc của giao thức định tuyến inter-AS!



# Ví dụ: chọn giữa nhiều AS

- Nay giờ, giả sử AS1 học từ giao thức inter-AS là có thể truyền tới subnet **x** được qua AS3 và qua AS2.
- Để cấu hình bảng forwarding, router 1d phải xác định gateway nào nó nên dùng để chuyển packet tới đích x
  - Đây cũng là công việc của giao thức định tuyến inter-AS!
- ***hot potato routing***: gửi packet tới router gần nhất trong 2 router



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 virtual circuit network

(Mạng mạch ảo) và  
datagram network  
(Mạng chuyển gói)

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

### Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## 4.5 các thuật toán routing

- link state
- distance vector
- hierarchical routing

## 4.6 routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 broadcast và multicast routing



# Định tuyến Intra-AS

- ❖ Còn gọi là *interior gateway protocols (IGP)*
- ❖ Các giao thức định tuyến intra-AS phổ biến:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (độc quyền của Cisco)

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

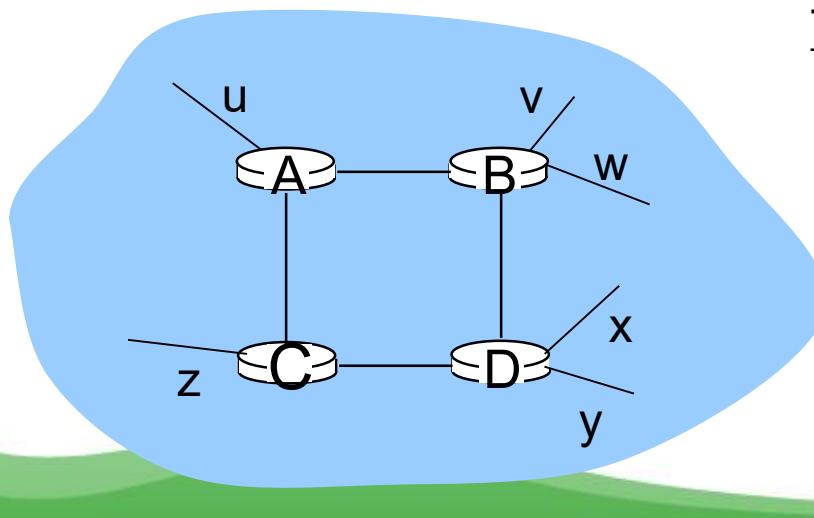
Broadcast và multicast routing

Routing trong internet



# RIP ( Routing Information Protocol)

- Công bố vào năm 1982 trong BSD-UNIX
- Thuật toán distance vector
  - Đơn vị đo khoảng cách: số lượng hop (max = 15 hops), mỗi link có giá trị là 1
  - Các DV được trao đổi giữa các neighbors mỗi 30 giây trong thông điệp phản hồi (còn gọi là **advertisement**)
  - Mỗi advertisement: liệt kê lên đến 25 **subnet** đích



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

**Routing trong internet**

Từ router A đến các **subnet** đích:

subnet	hops
u	1
v	2
w	2
x	3
y	3
z	2



# RIP: ví dụ

Giới thiệu

Virtual circuit network and diagram network

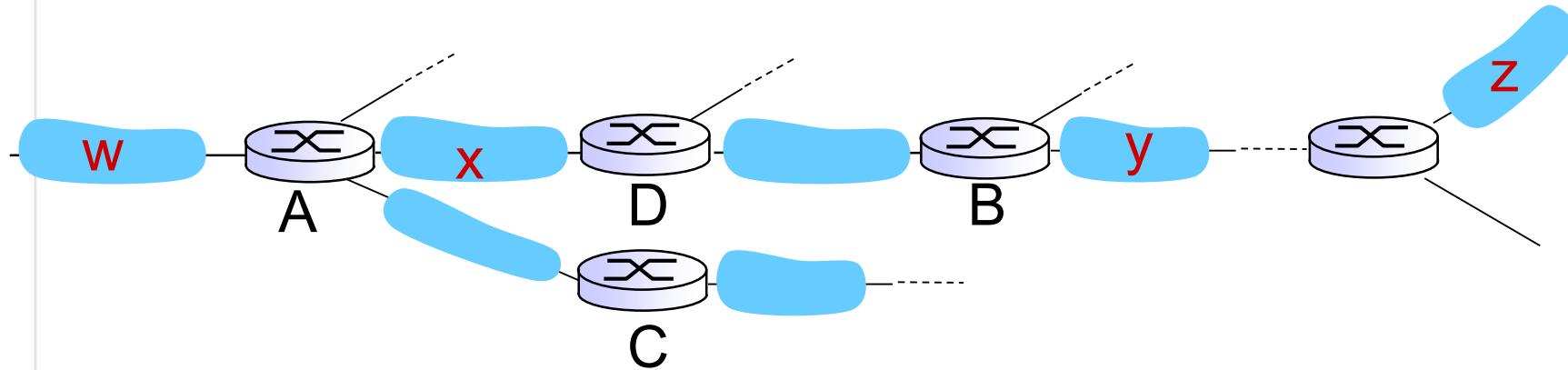
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



Bảng định tuyến trong router D

Subnet đích	router kế tiếp	số lượng hop đến đích
W	A	2
y	B	2
z	B	7
x	--	1
....	....	....



# RIP: ví dụ

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

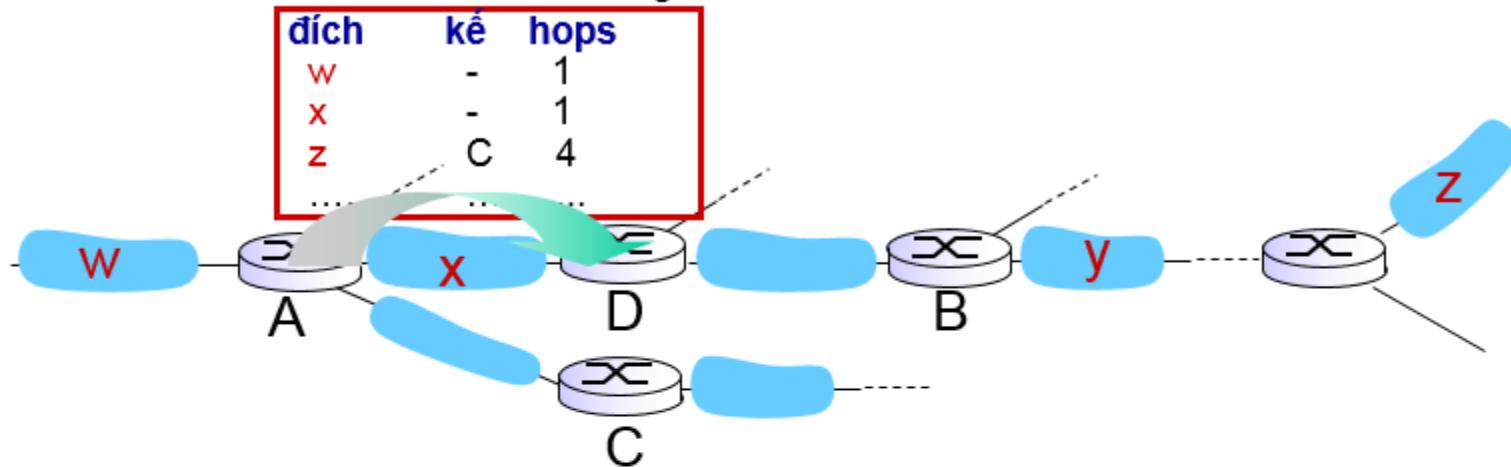
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Quảng cáo từ A-tới-D



Bảng định tuyến trong router D

Subnet đích	router kế tiếp	số lượng hop đến đích
W	A	2
y	B	2
z	B	7
x	--	1
...	...	...



# RIP: lỗi đường kết nối và phục hồi

Nếu không có quảng cáo nào sau 180 giây --> neighbor/kết nối được xem như đã chết

- Những đường đi qua neighbor này bị vô hiệu
- Các quảng cáo mới được gửi tới các neighbor còn lại
- Các neighbor đó tiếp tục gửi ra những quảng cáo mới đó (nếu các bảng bị thay đổi)
- Thông tin về lỗi đường kết nối nhanh chóng (?) lan truyền trên toàn mạng
- ***poison reverse*** được dùng để ngăn chặn vòng lặp ping-pong (khoảng cách vô hạn = 16 hops)

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

# RIP: xử lý bảng

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

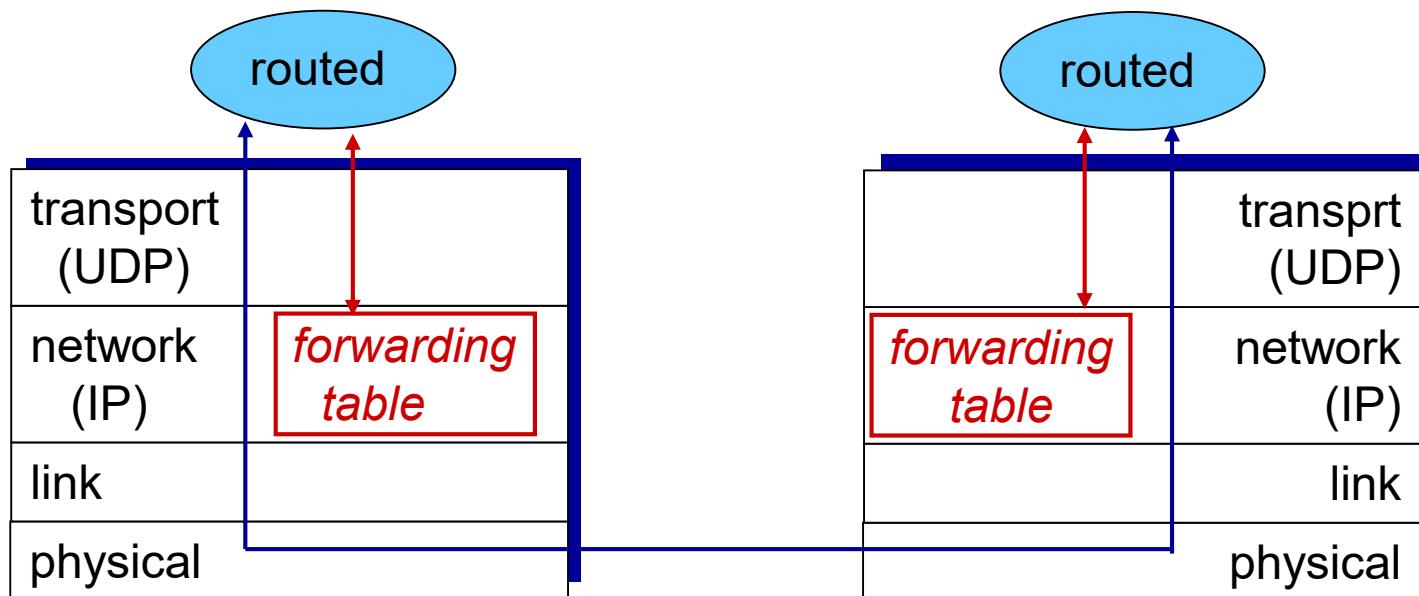
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- Các bảng định tuyến được quản lý bởi tiến trình ở tầng **Ứng dụng** được gọi là route-d (daemon)
- Các quảng cáo được gửi trong các packet UDP, được lặp lại theo chu kỳ



# OSPF (Open Shortest Path First)

- “open”: công khai cho mọi đối tượng sử dụng
- Dùng thuật toán Link State
  - Quảng bá gói tin LS
  - Bản đồ cấu trúc mạng tại mỗi node
  - Tính toán đường đi dùng thuật toán Dijkstra
- Thông điệp quảng bá OSPF chứa 1 mục thông tin cho mỗi router lân cận
- Các thông điệp này được phát tán đến **toàn bộ AS**
  - thông điệp OSPF được mang trực tiếp trên IP (chứ không phải là TCP hoặc UDP)
- Giao thức **định tuyến IS-IS** : gần giống với OSPF

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Các đặc tính “vượt trội” của OSPF (không có trong RIP)

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- **Bảo mật:** tất cả các thông điệp OSPF đều được chứng thực (để chống lại sự xâm nhập có hại)
- Cho phép có **nhiều đường đi** có chi phí như nhau (RIP chỉ cho 1)
- Với mỗi đường kết nối, có nhiều đơn vị tính chi phí (cost metrics) cho dịch vụ (**TOS**) khác nhau (ví dụ: chi phí đường kết nối vệ tinh được thiết lập “thấp” cho ToS nỗ lực tốt nhất; cao cho ToS thời gian thực)
- Hỗ trợ uni- và **multicast** tích hợp:
  - Multicast OSPF (MOSPF) dùng cùng cơ sở dữ liệu cấu trúc mạng như OSPF
- OSPF **phân cấp** được dùng trong các miền lớn (large domains).



# OSPF phân cấp

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

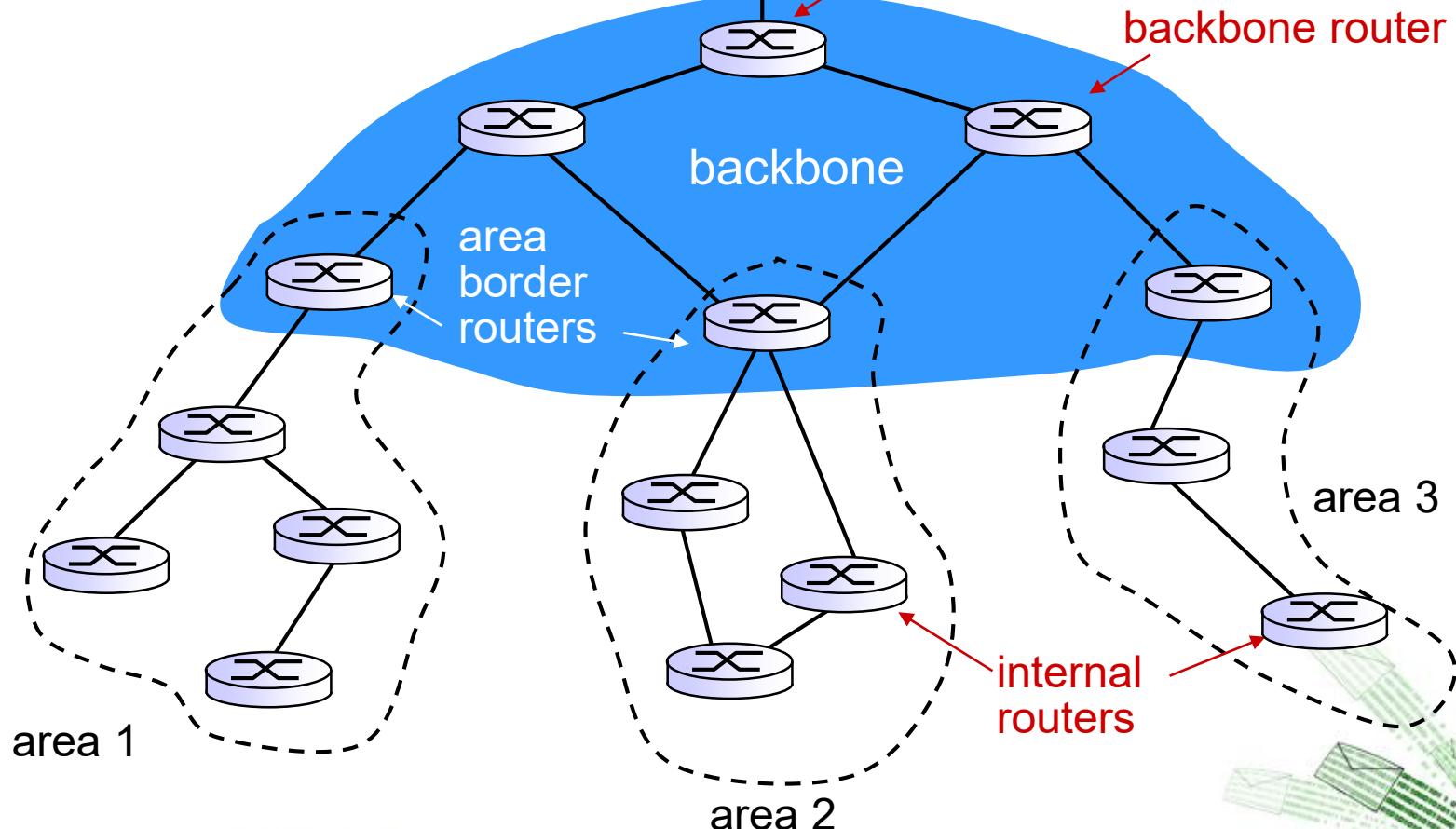
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Router biên (boundary )



# OSPF phân cấp

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- *Phân 2 cấp:* vùng cục bộ (local area), vùng backbone.
  - Các quảng cáo link-state chỉ trong 1 vùng
  - Mỗi node có chi tiết cấu trúc của vùng; chỉ biết hướng (đường ngắn nhất) đến các mạng trong các vùng khác.
- *Các router tại biên vùng (Area Border routers)* : “tóm tắt” các khoảng cách đến các mạng trong vùng của nó, quảng cáo đến các router biên của các vùng khác.
- *Các router trực (Backbone routers)*: chạy thuật toán định tuyến OSPF được giới hạn trong backbone.
- *Các router ngoại vi (Boundary routers)*: kết nối đến các AS khác.



# Định tuyến Internet inter-AS: BGP

- **BGP (Border Gateway Protocol):** giao thức định tuyến liên miền (inter-domain) trên thực tế
  - “gắn kết Internet lại với nhau”
- BGP cung cấp cho mỗi AS một phương tiện để:
  - **eBGP:** lấy thông tin về các subnet có thể truyền tới được từ các AS lân cận.
  - **iBGP:** lan truyền thông tin đó đến tất cả các router bên trong AS.
    - Xác định các đường đi “tốt” đến các mạng khác dựa trên thông tin về khả năng truyền tới và chính sách.
- Cho phép subnet quảng cáo sự tồn tại của nó đến phần còn lại của Internet: “*I am here*”

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast va`multicast routing

Routing trong internet



# BGP cơ bản

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

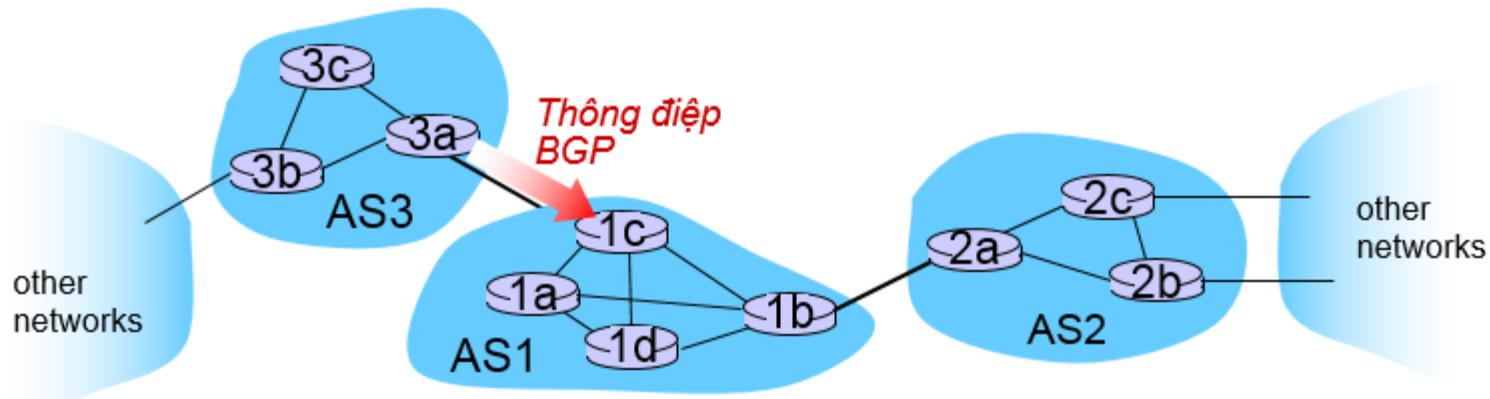
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- ❖ **BGP session:** 2 router BGP (“peer”) trao đổi các thông điệp BGP:
  - Quảng cáo **các đường đi** đến các mạng đích có phần đầu địa chỉ (prefix) khác nhau (Giao thức “path vector”)
  - Được trao đổi trên các kết nối TCP bán bền vững
- Khi AS3 quảng cáo 1 prefix tới AS1:
  - AS3 *hứa* là nó sẽ chuyển các datagram tới prefix đó
  - AS3 có thể gộp các prefixe trong quảng cáo của nó



# BGP cơ bản: phân phối thông tin đường đi

- Sử dụng phiên eBGP giữa 3a và 1c, AS3 gửi thông tin về prefix mạng có khả năng truyền tới được (prefix reachability info) tới AS1.
  - 1c sau đó có thể dùng iBGP phân phối thông tin prefix mới tới tất cả các router trong AS1
  - 1b sau đó có thể quảng cáo lại thông tin về khả năng truyền mới tới AS2 trên phiên làm việc eBGP từ 1b-tới-2a
- Khi router học prefix mới, thì nó sẽ tạo mục cho prefix đó trong bảng forwarding của nó.

Giới thiệu

Virtual circuit network and diagram network

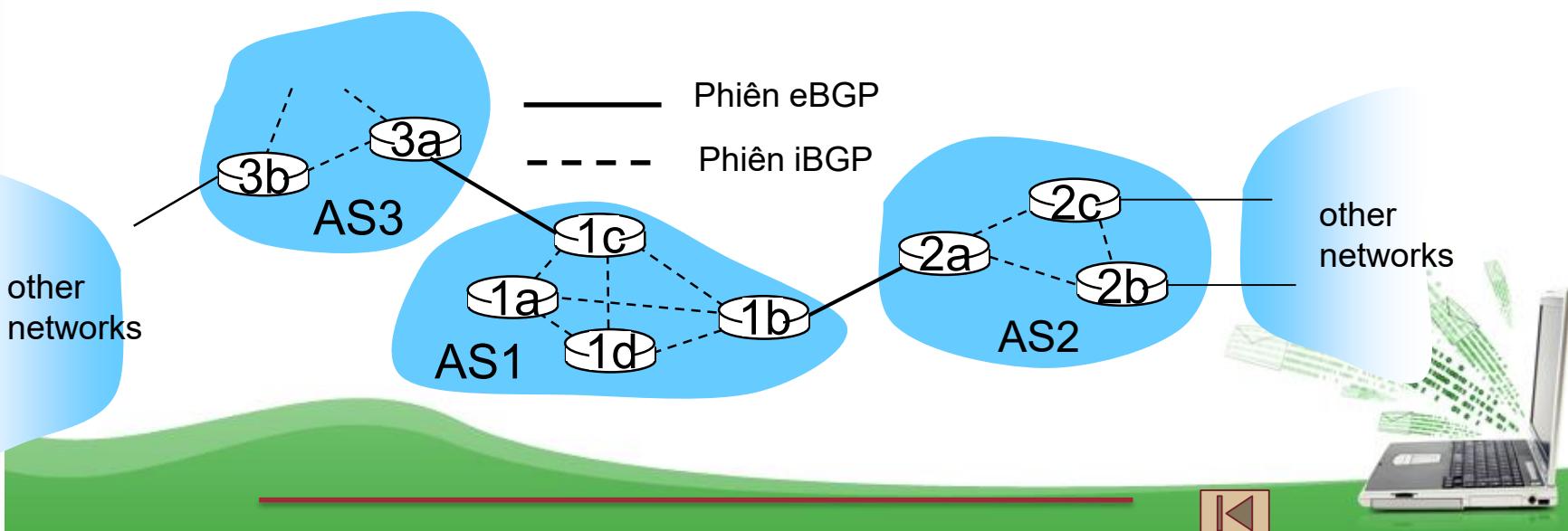
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Các thuộc tính đường đi và các đường BGP

- Prefix được quảng cáo bao gồm các thuộc tính BGP
  - prefix + attributes = “route”
- 2 thuộc tính quan trọng:
  - **AS-PATH**: chứa các AS đã chuyển các quảng cáo về prefix qua: ví dụ: AS 67, AS 17
  - **NEXT-HOP**: chỉ ra cụ thể router nào bên trong AS có kết nối tới AS kế tiếp. (có thể có nhiều đường kết nối từ AS hiện tại tới AS kế tiếp)
- router gateway có nhiệm vụ nhận quảng cáo về các đường đi sử dụng **chính sách nhập (import policy)** để quyết định chấp nhận/từ chối
  - Ví dụ: không bao giờ đi qua AS x
  - Định tuyến **dựa trên chính sách (policy-based routing)**

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Sự lựa chọn BGP route

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- Router có thể học nhiều hơn 1 đường đi tới AS đích, chọn đường đi dựa trên:
  - Thuộc tính giá trị ưu tiên cục bộ: quyết định dựa vào chính sách
  - AS-PATH ngắn nhất
  - NEXT-HOP router gần nhất: định tuyến hot potato
  - Tiêu chuẩn bổ sung



# Các thông điệp BGP

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

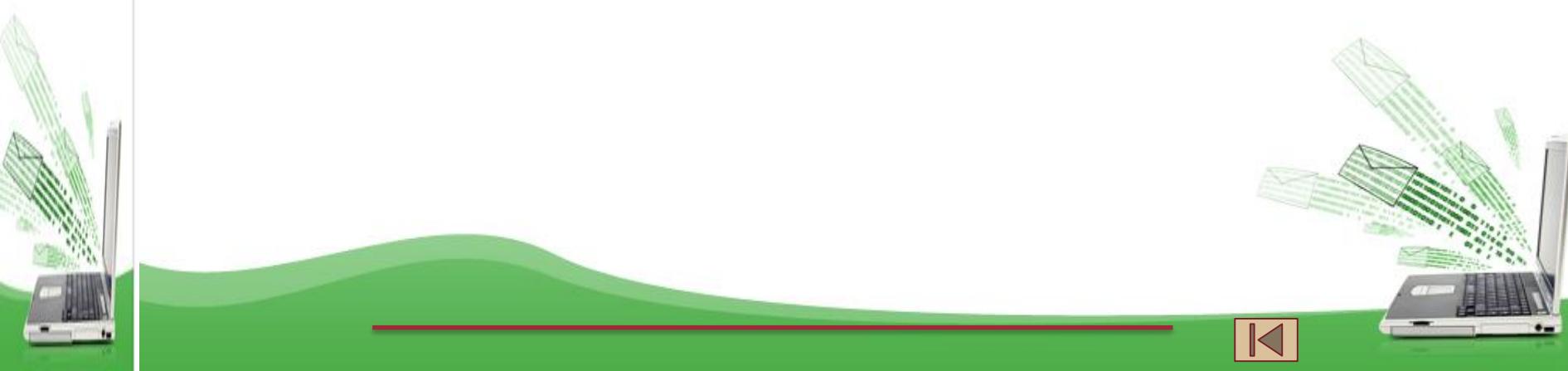
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- Các thông điệp BGP được trao đổi giữa các peer trên kết nối TCP
- Các thông điệp BGP:
  - **OPEN**: mở kết nối TCP đến peer và chứng thực bên gửi
  - **UPDATE**: quảng cáo đường đi mới (hoặc báo hủy đường đi cũ)
  - **KEEPALIVE**: giữ kết nối hoạt động khi không có các UPDATES; còn đóng vai trò là thông điệp ACK cho thông điệp OPEN
  - **NOTIFICATION**: thông báo các lỗi trong thông điệp trước đó; cũng được sử dụng để đóng kết nối



# Dùng nhiều giao thức định tuyến

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## Dùng nhiều giao thức định tuyến trong cùng hệ thống?

Làm thế nào để cập nhật thông tin trong bảng forwarding của 1 router?

- Câu trả lời rất phức tạp!
- Buộc các định tuyến phân cấp (mục 4.5.3) lại với nhau với BGP (4.6.3) và OSPF (4.6.2).
- Cung cấp tổng quan về BGP!



# Làm thế nào để cập nhật một mục vào bảng forwarding?

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

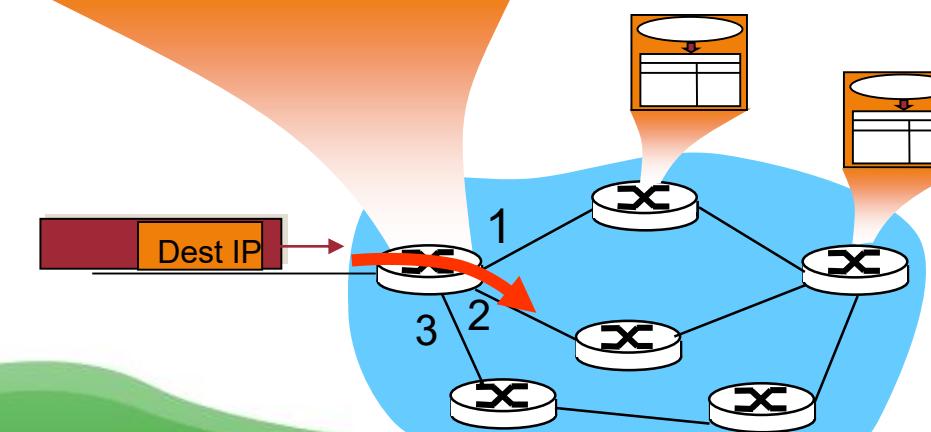
Routing trong internet

Thuật toán định tuyến

entry

prefix	output port
138.16.64/22	3
124.12/16	2
212/8	4
.....	...

Giả sử prefix  
là trong AS khác.



## Làm thế nào để cập nhật một mục vào bảng forwarding?

Giới thiệu

Virtual circuit network and diagram network

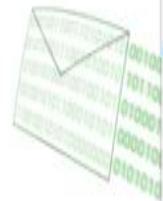
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



1. Router được prefix
2. Router xác định port ra (output port) cho prefix
3. Router đưa thông tin về cổng/prefix đó vào trong bảng forwarding



# Router nhận được prefix

Giới thiệu

Virtual circuit network and diagram network

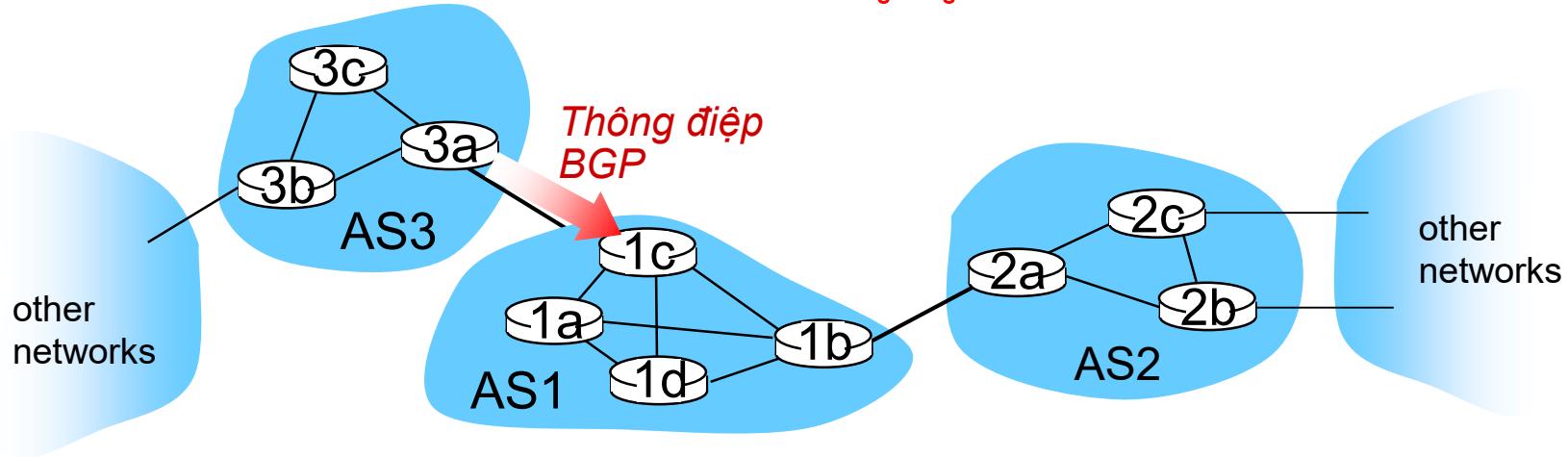
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



- ❖ Thông điệp BGP chứa “các đường đi” – “routes”
- ❖ “đường đi” gồm 1 prefix và các thuộc tính: AS-PATH, NEXT-HOP, ...
- ❖ Ví dụ: route:
  - ❖ Prefix: 138.16.64/22 ; AS-PATH: AS3 AS131 ; NEXT-HOP: 201.44.13.125



# Router có thể nhận được nhiều route

Giới thiệu

Virtual circuit network and diagram network

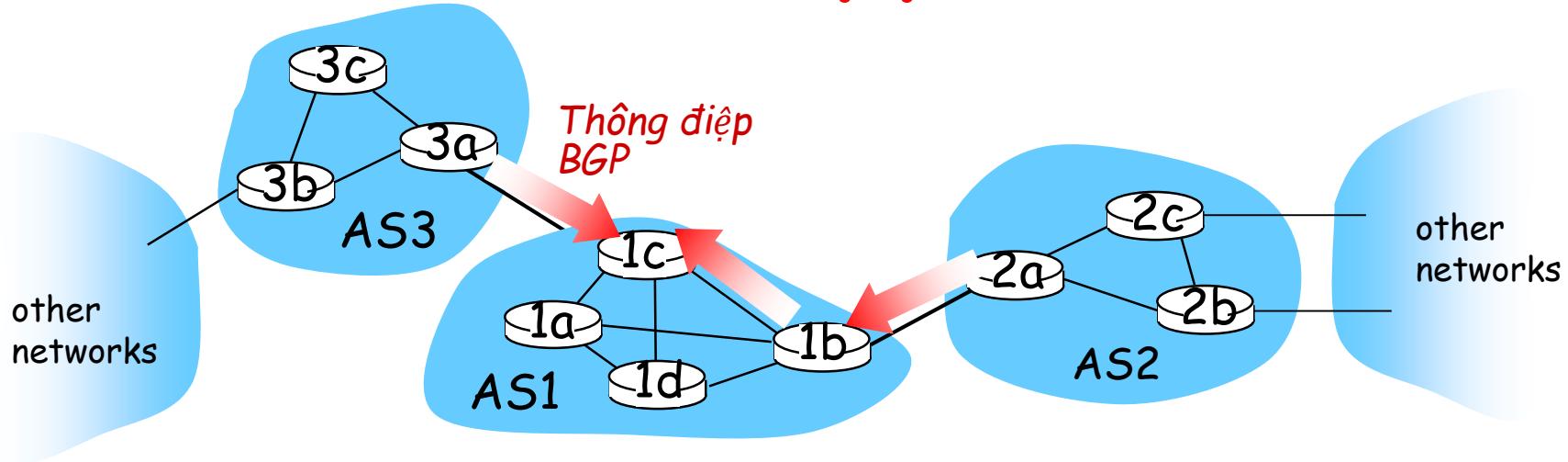
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



- ❖ Router có thể nhận được nhiều route cho cùng prefix
- ❖ Phải chọn 1 route



# Chọn route BGP tốt nhất tới prefix

- Router chọn route dựa trên AS-PATH ngắn nhất

- ❖ Ví dụ:

Chọn

- ❖ AS2 AS17 to 138.16.64/22
- ❖ AS3 AS131 AS201 to 138.16.64/22

- ❖ Điều gì sẽ xảy ra nếu có mối ràng buộc?  
Chúng ta sẽ bàn sau!

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Tìm route nội bộ tốt nhất đến route BGP

- Dùng thuộc tính NEXT-HOP của route được lựa chọn
  - Thuộc tính NEXT-HOP của Route là địa chỉ IP của interface của router đó, nơi bắt đầu AS PATH đó.
- Ví dụ:
  - ❖ AS-PATH: AS2 AS17 ; NEXT-HOP: 111.99.86.55
- Router sử dụng OSPF để tìm ra đường đi ngắn nhất từ 1c tới 111.99.86.55

Giới thiệu

Virtual circuit network and diagram network

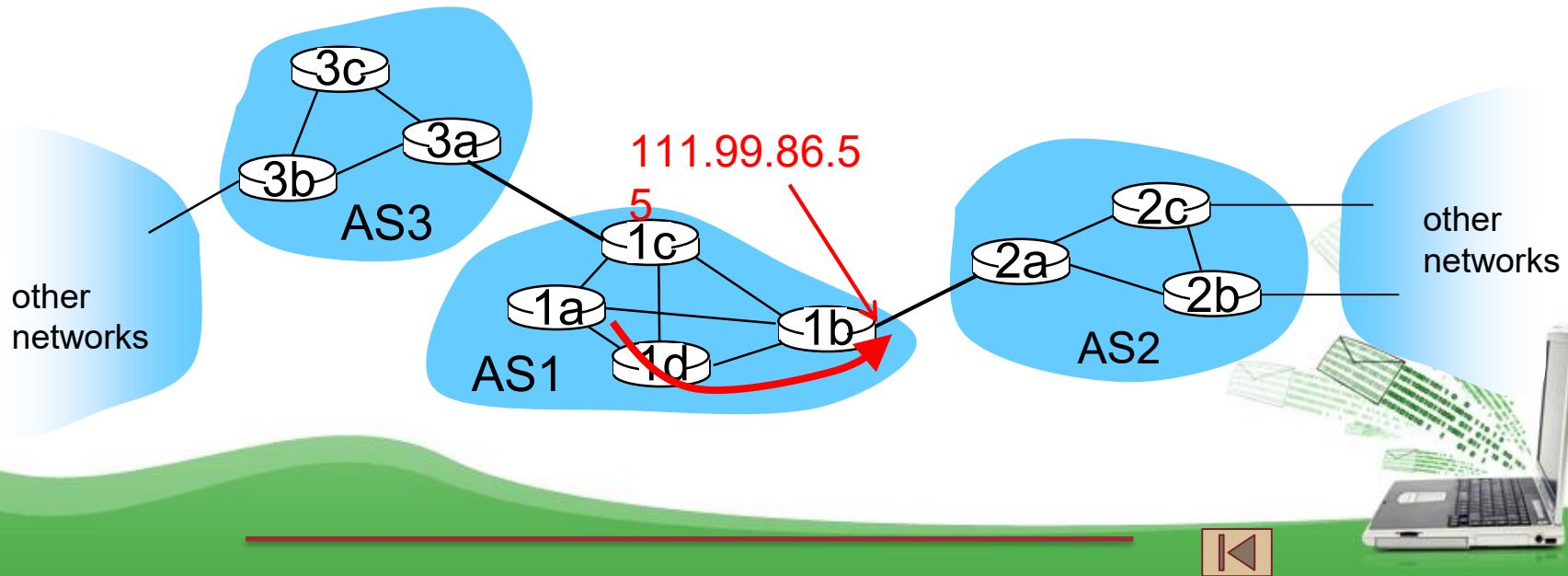
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Router xác định port cho route

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

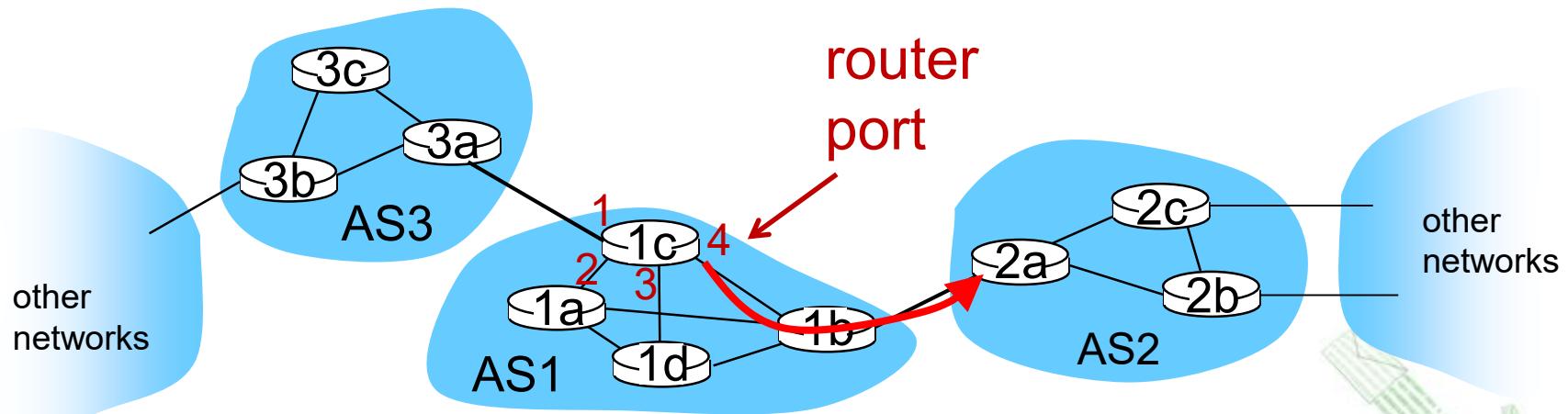
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- Xác định port theo đường đi OSPF ngắn nhất
- Thêm thông tin prefix-port vào bảng forwarding của nó:
  - (138.16.64/22 , port 4)



# Định tuyến Hot Potato

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

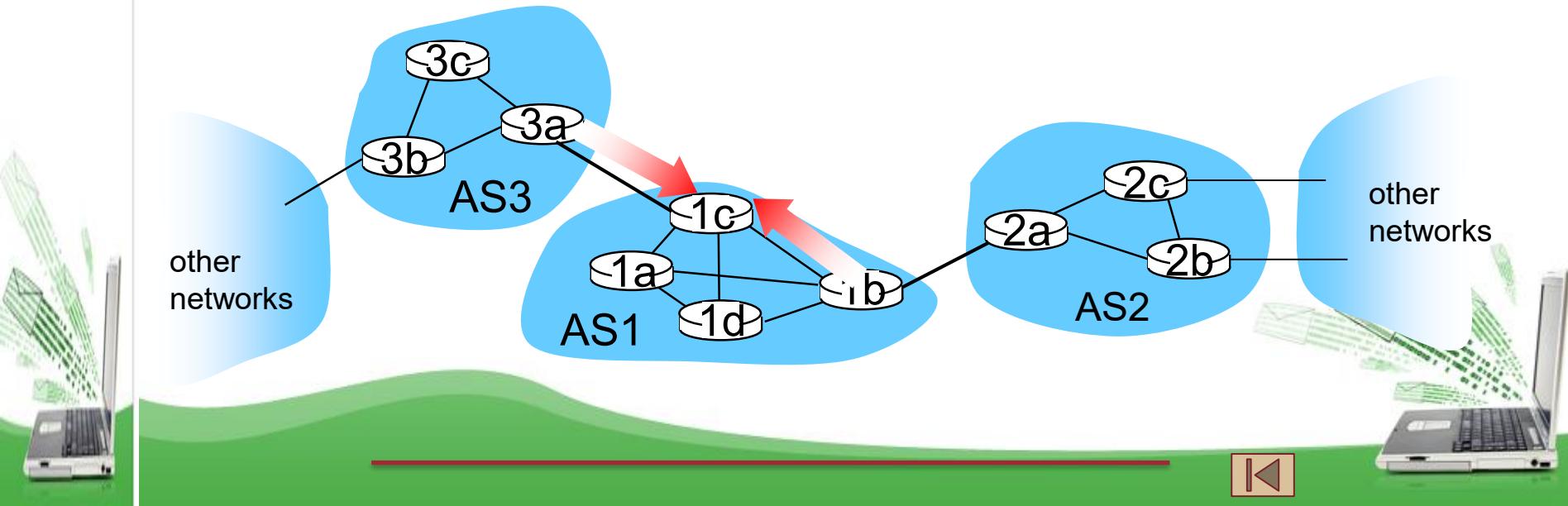
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

- Giả sử có 2 hoặc nhiều route liên tuyến tốt nhất (best inter-routes).
- Sau đó chọn route với NEXT-HOP gần nhất
  - Dùng OSPF để xác định cổng nào là gần nhất
  - Hỏi: từ 1c, chọn AS3 AS131 AS201 hoặc AS2 AS17?
  - Đáp: route AS3 AS131 AS201 vì nó gần hơn



# Làm thế nào để nhập 1 thông tin định tuyến vào bảng forwarding?

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## Tóm tắt

1. Router nhận biết về prefix
  - Thông qua các quảng cáo về đường đi của BGP từ các router khác
2. Xác định port ra của router để tới prefix đó
  - Dùng thông tin về đường đi của BGP vừa chọn để tìm ra đường đi liên vùng (inter-AS) tốt nhất
  - Dùng OSPF để tìm ra đường đi tốt nhất trong nội vùng (intra-AS) dẫn đến đường đi liên vùng tốt nhất (inter-AS route)
  - Router xác định port của router cho đường đi tốt nhất đó
3. Đưa thông tin về prefix-port này vào trong bảng forwarding



# Chính sách BGP routing

Giới thiệu

Virtual circuit network and diagram network

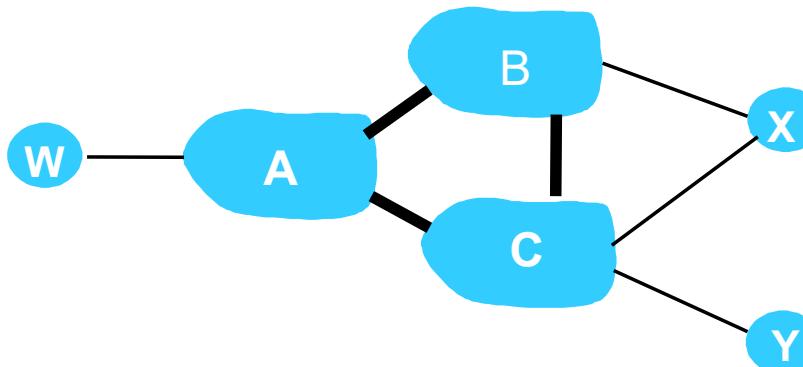
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



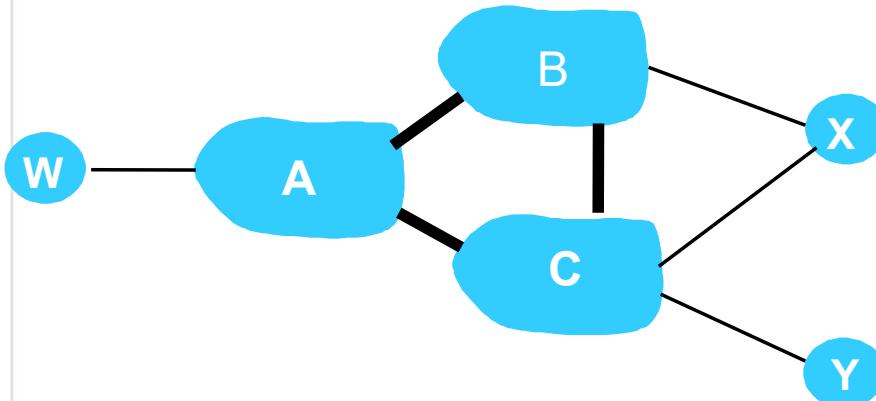
Ký hiệu:

Mạng của nhà cung cấp

Mạng của khách hàng

- ❖ A,B,C là các nhà cung cấp mạng
- ❖ X,W,Y là khách hàng (của nhà cung cấp mạng)
- ❖ X là mạng thuê bao kép (**dual-homed**): được kết nối vào 2 mạng
  - X không muốn có tuyến đường từ B thông qua X tới C
  - .. Vì vậy X sẽ không quảng cáo cho B tuyến đường

# Chính sách BGP routing (2)



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

Ký hiệu:

Mạng của nhà cung cấp

Mạng của khách hàng

- ❖ A quảng cáo đường đi AW cho B
- ❖ B quảng cáo đường đi BAW cho X
- ❖ B có nên quảng cáo đường đi BAW cho C hay không?

- Không nên! B không nhận được “lợi ích” cho việc định tuyến CBAW vì cả W và C không phải là khách hàng của B
- B muốn bắt buộc C đi qua A
- B **chỉ** muốn dẫn đường đi/đến các khách hàng của nó!



# Tại sao phải định tuyến Intra-AS, Inter-AS khác nhau?

## Chính sách:

- inter-AS: người quản trị muốn điều hành hoạt động định tuyến trong mạng, ai đi qua mạng của họ
- intra-AS: 1 người quản trị, vì thế không cần có chính sách quy định việc định tuyến

## Linh hoạt:

- Định tuyến phân cấp làm giảm kích thước bảng định tuyến, giảm lưu lượng cập nhật

## Hiệu suất:

- intra-AS: có thể tập trung vào hiệu suất
- inter-AS: chính sách quan trọng hơn hiệu suất

Giới thiệu

Virtual circuit network and diagram network

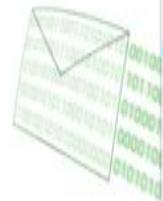
Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 virtual circuit network (Mạng mạch ảo) và datagram network (Mạng chuyển gói)

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet

## 4.5 các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

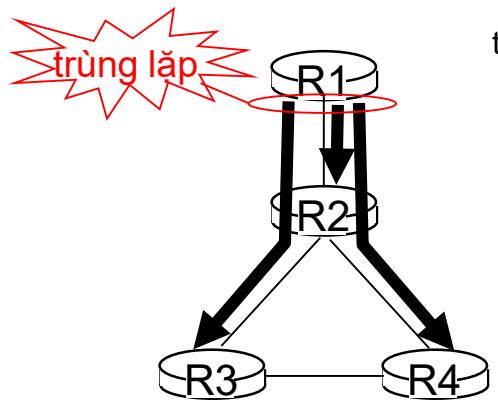
- RIP
- OSPF
- BGP

## 4.7 Broadcast và Multicast routing

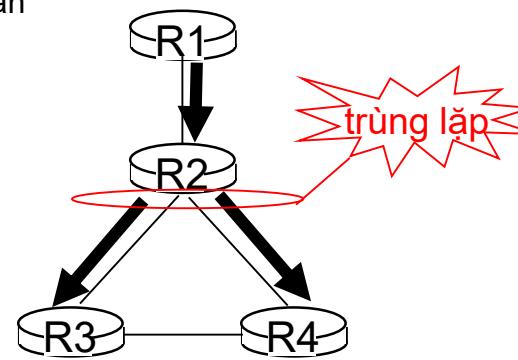


# Broadcast routing

- Chuyển các gói từ nguồn tới tất cả các node khác
- Nguồn trùng lắp thì không có hiệu quả:



tạo/truyền dẫn  
trùng lắp



nguồn  
trùng lắp

trùng lắp  
trong mạng

- ❖ Nguồn trùng lắp: làm sao xác định được địa chỉ người nhận?

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Trùng lặp trong mạng

- **Flooding:** khi node nhận được packet broadcast, nó gởi bản sao đến tất cả các node lân cận
  - Vấn đề: lặp lại & bão broadcast
- **Flooding có điều khiển:** node chỉ broadcast packet nếu nó không gửi broadcast giống như vậy trước đó
  - node theo dõi các packet ID đã broadcast
  - Hoặc reverse path forwarding (RPF): chỉ chuyển các packet nếu nó đã đến node trên đường đi ngắn nhất giữa node và nguồn
- **Spanning tree:**
  - Không có các packet trùng được bắt cứ node nào nhận

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Routing trong internet



# Spanning tree

Giới thiệu

Virtual circuit network and diagram network

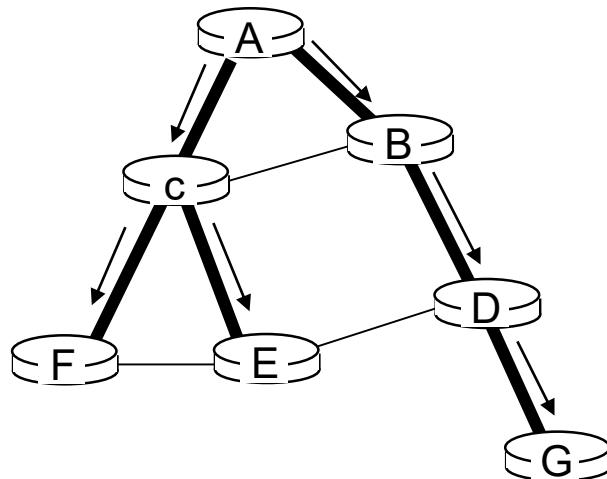
Cấu trúc trong router

Ip:internet protocol

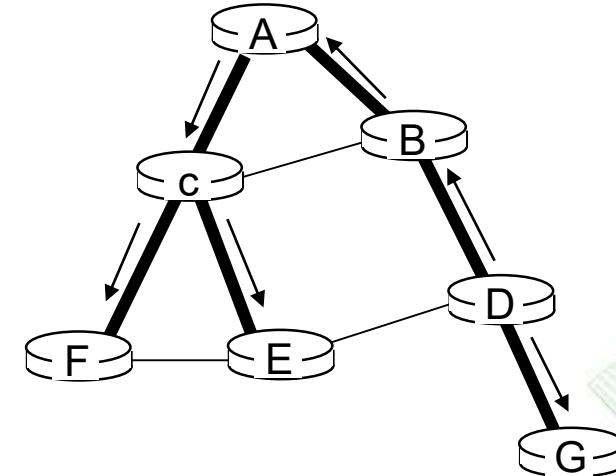
Thuật toán routing

Broadcast và multicast routing

- ❖ Đầu tiên xây dựng một spanning tree
- ❖ Sau đó các node chuyển tiếp/tạo các bản sao chỉ dọc theo spanning tree



(a) broadcast được khởi tạo tại A

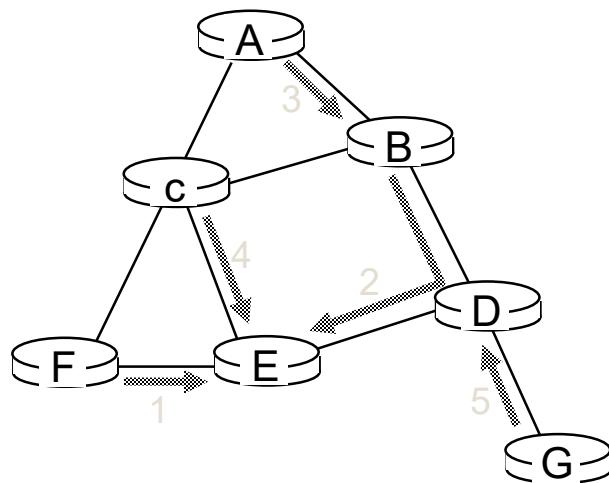


(b) broadcast được khởi tạo tại D

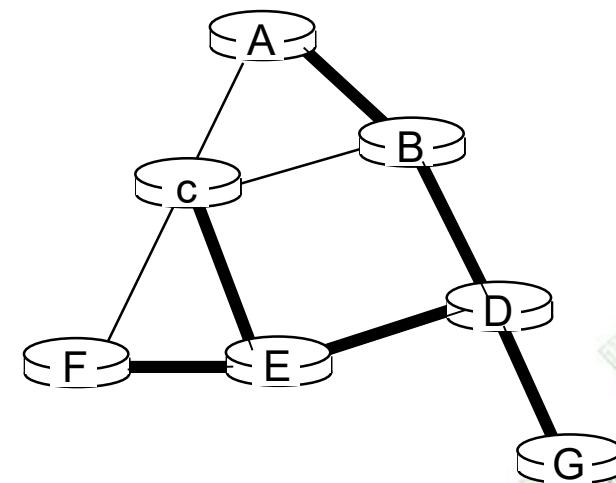


# Spanning tree: tạo cây

- ❖ Node trung tâm
- ❖ Mỗi node gửi thông điệp gia nhập (unicast join message) đến node trung tâm
  - Thông điệp này được chuyển tiếp cho đến khi nó đến được một node đã nằm trên spanning tree



(a) Các bước xây dựng spanning tree (center: E)



(b) spanning tree đã được xây dựng xong



# Multicast routing: mô tả vân đề

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

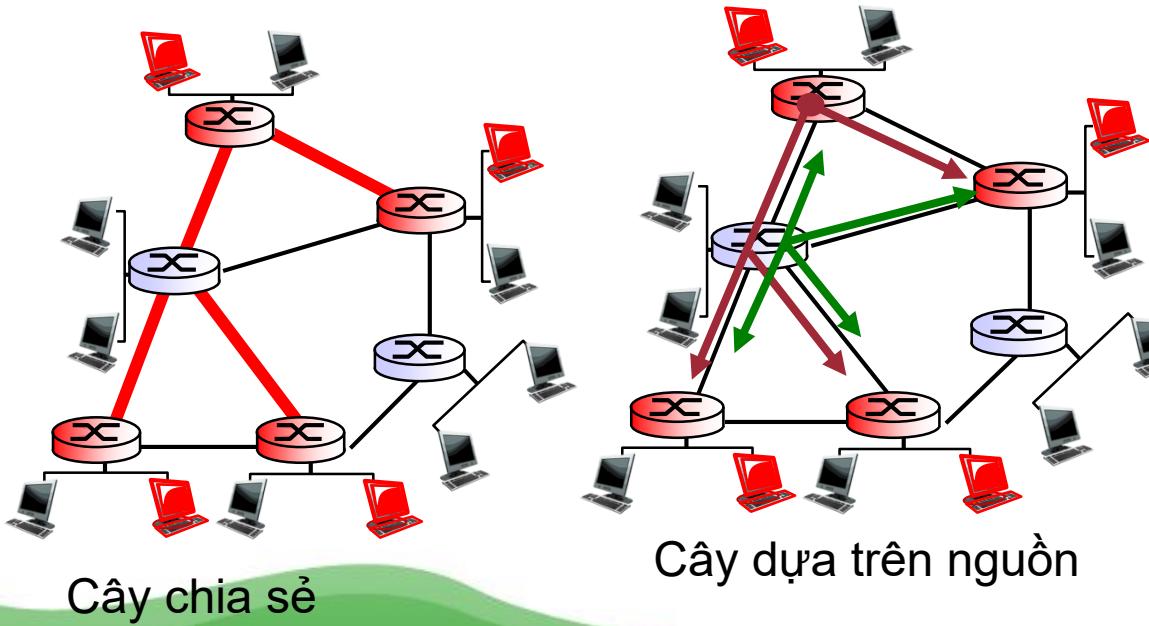
Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

**Mục tiêu:** tìm một cây (hoặc các cây) kết nối các router có các thành viên trong nhóm multicast

- **Cây (tree):** không phải tất cả các đường đi giữa các router đều được sử dụng
- **Cây chia sẻ (shared-tree):** cây giống nhau được sử dụng bởi các thành viên trong nhóm
- **Cây dựa trên nguồn(source-based):** cây khác nhau từ nơi gửi tới nơi nhận



## Ký hiệu

	Thành viên nhóm
	Không thuộc nhóm
	Router có thành viên
	Router không có thành viên



# Các cách tiếp cận để xây dựng các cây multicast

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Các hướng tiếp cận:

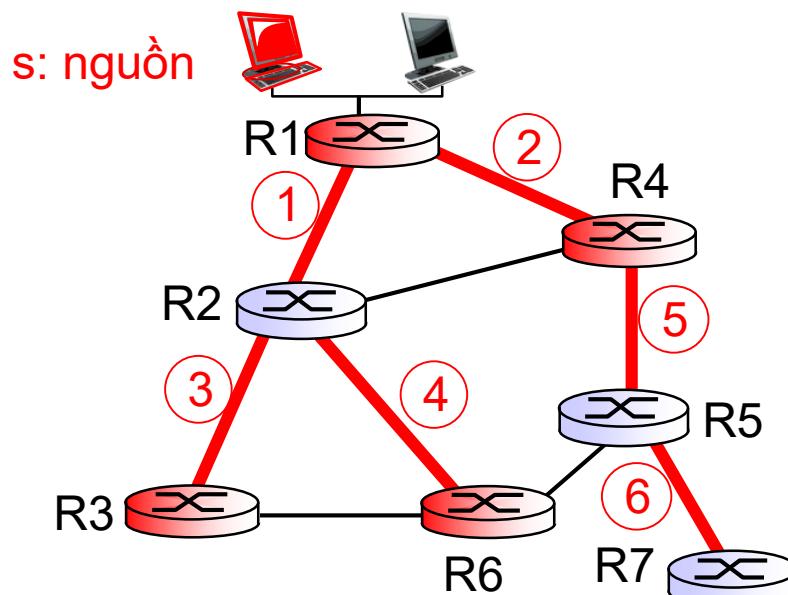
- **Cây dựa trên nguồn (source-based tree):** một cây cho mỗi nguồn
  - Các cây đường đi ngắn nhất
  - Chuyển tiếp theo đường đi ngược (Reverse path forwarding)
- **Cây chia sẻ nhóm:** cả nhóm dùng chung 1 cây
  - Mở rộng tối thiểu (Steiner)
  - Các cây dựa trên trung tâm (center-based trees)

...tìm hiểu các cách tiếp cận cơ bản, sau đó chọn các giao thức cụ thể áp dụng cho các hướng tiếp cận này



# Cây đường đi ngắn nhất

- Cây chuyển tiếp multicast (mcast forwarding tree): cây đường đi ngắn nhất dẫn đường từ nguồn tới tất cả các nơi nhận
  - Thuật toán Dijkstra



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Ký hiệu

router với các thành viên của nhóm đã được gắn vào

router không có các thành viên nào của nhóm được gắn vào

Đường kết nối được sử dụng cho forwarding,  
i chỉ thứ tự đường link được thêm vào bởi thuật toán



# Reverse path forwarding

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

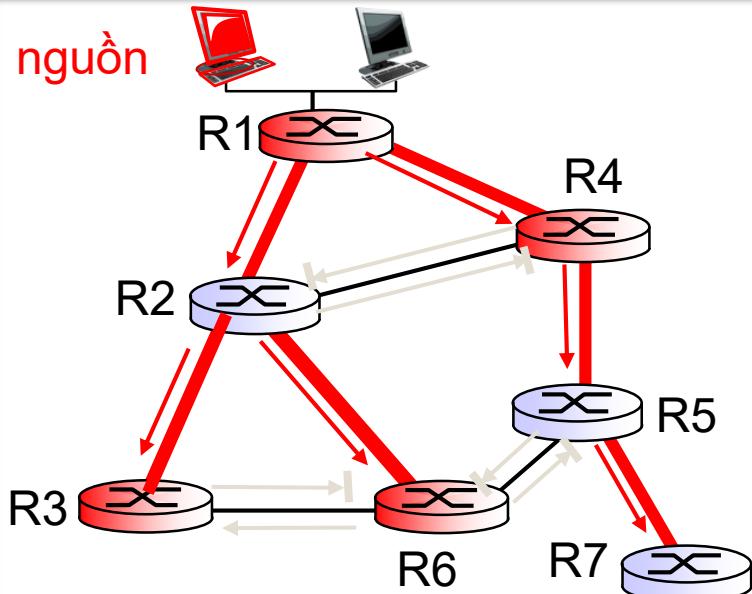
- ❖ Dựa vào thông tin của router về đường đi ngắn nhất khi gửi unicast từ nó đến nơi gửi
- ❖ Mỗi router có cách xử lý forwarding đơn giản:

*if* (datagram multicast được nhận trên đường kết nối đến (incoming link) nằm trên đường đi ngắn nhất nối về từ trung tâm)  
*then* gửi datagram tới tất cả các kết nối ra (outgoing link)  
*else* bỏ qua datagram



# Reverse path forwarding: ví dụ

s: nguồn



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

## Kí hiệu



router với các thành viên  
của nhóm đã gắn vào



router không có các thành  
viên của nhóm gắn vào

→ datagram sẽ được forward

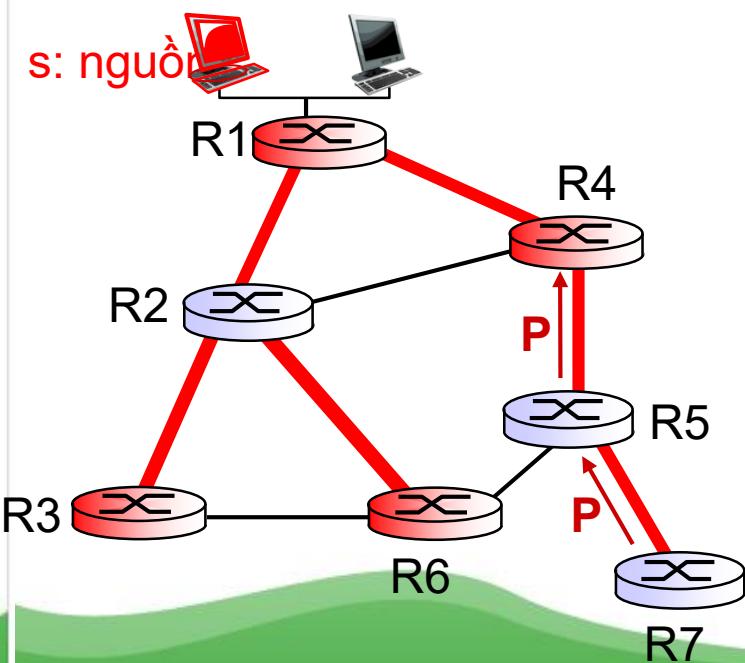
→ datagram sẽ không  
được forward

- ❖ Kết quả là một cây SPT đảo ngược tùy theo từng nguồn
  - Có thể là một lựa chọn không tốt với các kết nối không đồng bộ



# Reverse path forwarding: xén (pruning)

- Cây chuyển tiếp (forwarding tree) chứa các cây con không có các thành viên trong nhóm multicast
  - Không cần forward các datagram xuống những cây con này
  - Các thông điệp “xén” được router không có các thành viên nhóm gửi lên trên



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

## Ký hiệu



router với các thành viên của nhóm đã gắn vào



router không có các thành viên của nhóm gắn vào



thông điệp xén



các kết nối với multicast forward



# Cây chia sẻ (Shared-tree): cây steiner

- *Cây steiner* : cây có chi phí thấp nhất kết nối đến tất cả các router có nối các thành viên nhóm
  - Vấn đề là NP-complete (vấn đề phức tạp trong toán học đồ thị)
  - Hiện đã có các thuật giải rất tốt
  - Không sử dụng trong thực tế:
    - Độ phức tạp trong tính toán cao
    - Cần thông tin về vè toàn bộ mạng
    - monolithic: phải chạy lại mỗi khi router cần gia nhập/rời khỏi

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing



# Cây dựa vào trung tâm (Center-based trees)

- Một cây được dùng chung cho tất cả
- Một router được xác định là “*trung tâm*” của cây
- Để gia nhập:
  - router biên (edge) gửi thông điệp gia nhập (unicast *join-msg*) đến router trung tâm
  - *Thông điệp gia nhập (join-msg)* “được xử lý” bởi các router trung gian và chuyển đến router trung tâm
  - *Thông điệp gia nhập*, hoặc đến nhánh của cây trung tâm này, hoặc đến ngay trung tâm
  - Đường đi của *thông điệp gia nhập* trở thành nhánh cây mới nối router mới tham gia này

## Giới thiệu

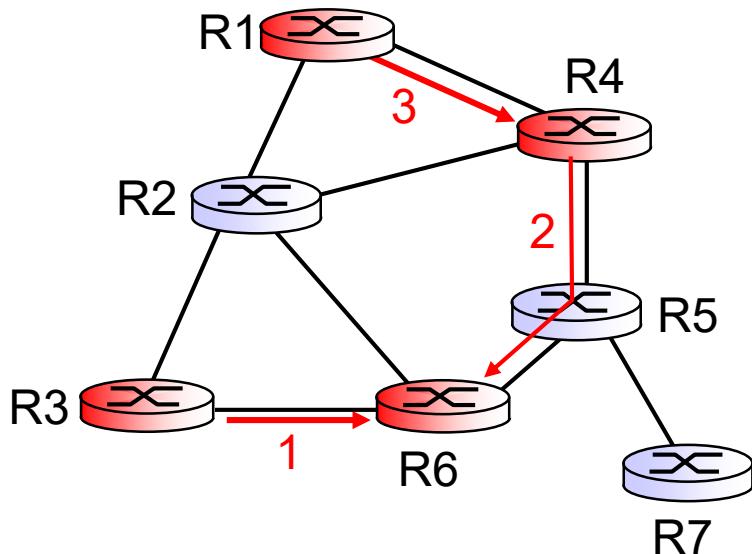
- Virtual circuit network and diagram network
- Cấu trúc trong router
- Ip:internet protocol
- Thuật toán routing

Broadcast và multicast routing



# Cây dựa vào trung tâm: ví dụ

Giả sử R6 được chọn là trung tâm:



Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

## LEGEND



router có các thành viên  
của nhóm nối vào



router không có các thành  
viên của nhóm nối vào



thứ tự đường đi mà các  
thông điệp gia nhập đã tạo  
ra



# Internet Multicasting Routing: DVMRP

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

- **DVMRP:** distance vector multicast routing protocol, RFC1075
- **flood và prune:** chuyển gói theo đường đi ngược (reverse path forwarding), cây dựa vào nguồn (source-based tree)
  - Cây RPF được xây dựng dựa trên việc trao đổi các bảng định tuyến của DVMRP giữa các router hỗ trợ DVMRP
  - Không có các giả định về các cách truyền unicast bên dưới
  - Datagram đầu tiên gửi đến nhóm multicast được gửi tới mọi nơi (**flood**) thông qua RPF
  - Các router không cần làm việc nhóm: gửi thông điệp cắt xén (**prune**) lên trên



# DVMRP: tiếp tục...

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

- **Trạng thái mềm:** router DVMRP theo chu kỳ (1 phút) sẽ “quên” các nhánh cây bị cắt xén:
  - Dữ liệu mcast một lần nữa đổ xuống các nhánh không được cắt xén
  - Router phía dưới: cắt xén lần nữa hoặc tiếp tục nhận dữ liệu
- Các router có thể nhanh chóng ghép lại vào cây
  - Gia nhập IGMP tại các lá
- Còn lại
  - Thường được thực hiện trong router thương mại



# Tunneling

Giới thiệu

Virtual circuit network and diagram network

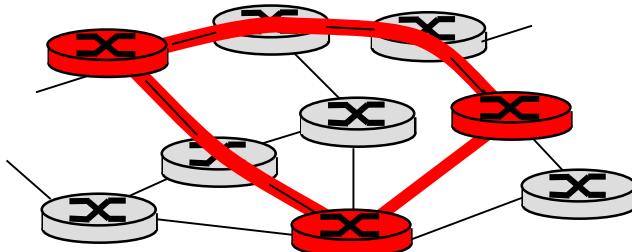
Cấu trúc trong router

Ip:internet protocol

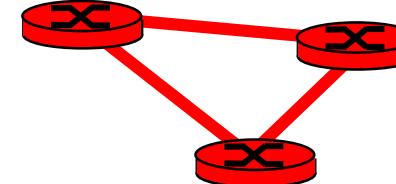
Thuật toán routing

Broadcast và multicast routing

Q: làm cách nào để kết nối các “đảo” router multicast trong một “biển” các router unicast?



Cấu trúc vật lý



Cấu trúc logic

- ❖ datagram multicast được đóng gói trong datagram “thông thường” (không có multicast)
- ❖ datagram IP thông thường được gửi thông qua “đường hầm” (“tunnel”) bằng gói IP unicast đến router multicast nhận (xem lại IPv6 bên trong đường hầm IPv4)
- ❖ router mcast nhận mở gói để lấy datagram multicast



# PIM: Protocol Independent Multicast

- Không phụ thuộc vào bất kỳ thuật toán định tuyến unicast bên dưới nào (underlying unicast routing algorithm) (làm việc với tất cả)
- 2 ngữ cảnh phân phối multicast khác nhau:

## Dài đặc:

- ❖ Các thành viên nhóm đóng gói dài đặc, trong khoảng cách “gần”.
- ❖ Băng thông dư thừa

## Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

## Thưa thớt:

- ❖ Số lượng các mạng với các thành viên nhóm ít
- ❖ Các thành viên nhóm “được phân bổ thưa thớt”
- ❖ Băng thông không dư thừa



# Kết quả của sự phân chia thưa thớt-dày đặc:

## Dày đặc

- Giả định các router đều là thành viên nhóm cho đến khi các router không liên quan tách ra khỏi nhóm
- Kiến trúc hướng dữ liệu (data-driven construction)* trên cây mcast (ví dụ RPF)
- Băng thông và router- không-thuộc-nhóm xử lý phung phí

## Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

## Thưa thớt:

- Không có thành viên cho đến khi các router thực sự gia nhập
- Kiến trúc hướng người nhận (receiver-driven construction)* của cây mcast (ví dụ cây dựa vào trung tâm)
- Băng thông và router- không-thuộc-nhóm xử lý vừa phải



# PIM - trạng thái dày đặc

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

Flood-and-prune RPF: tương tự như DVMRP  
nhưng...

- ❖ Giao thức bên dưới cung cấp thông tin RPF cho datagram đến
- ❖ Đẩy gói (flood) xuống dưới ít phức tạp (ít hiệu quả) hơn so với DVMRP giảm độ tin cậy của thuật toán định tuyến ở dưới
- ❖ Có cơ chế trong giao thức cho router tự phát hiện có phải là router ở node lá (leaf-node)



# PIM – trạng thái thưa thớt

- Tiếp cận hướng trung tâm (center-based)
- router gởi thông điệp *gia nhập* (join msg) đến điểm tập hợp (rendezvous point-RP)
  - Các router trung gian cập nhật trạng thái và chuyển tiếp thông điệp *gia nhập*
- Sau khi gia nhập thông qua RP, router có thể chuyển sang cây với nguồn cụ thể (source-specific tree)
  - Hiệu xuất tăng: ít tập trung, các đường đi ngắn hơn

Giới thiệu

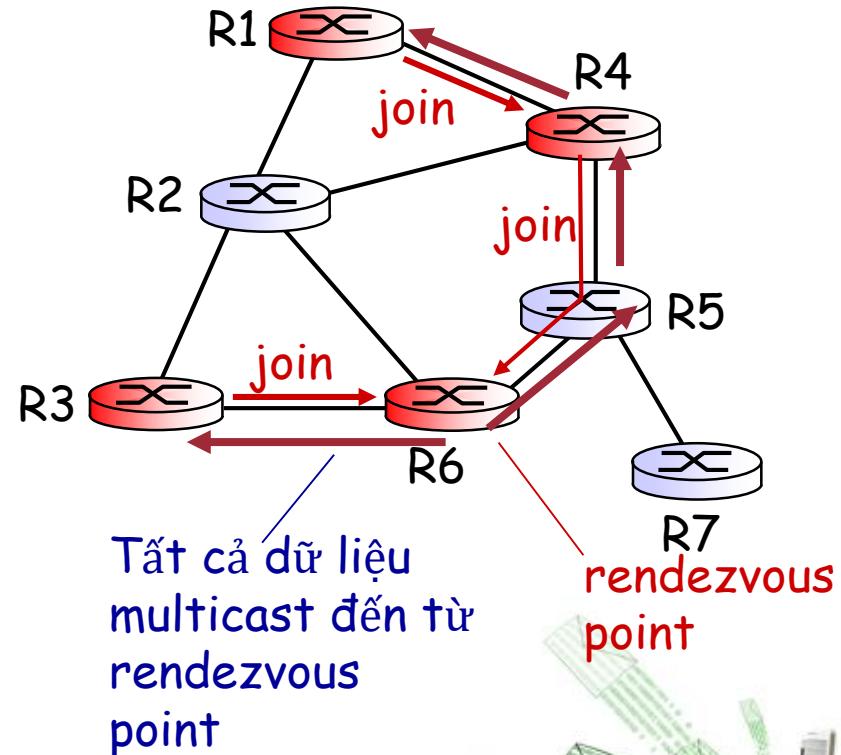
Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast va multicast routing



# PIM – trạng thái thưa thớt

## (Các) Bên gửi:

- Dữ liệu unicast đến RP, RP phân phối xuống cây có nút gốc là RP
- RP có thể mở rộng cây multicast ngược dòng lên đến nguồn
- RP có thể gửi thông điệp dừng (stop msg) nếu không nối tới bên nhận nào
  - “không có ai đang lắng nghe!”

Giới thiệu

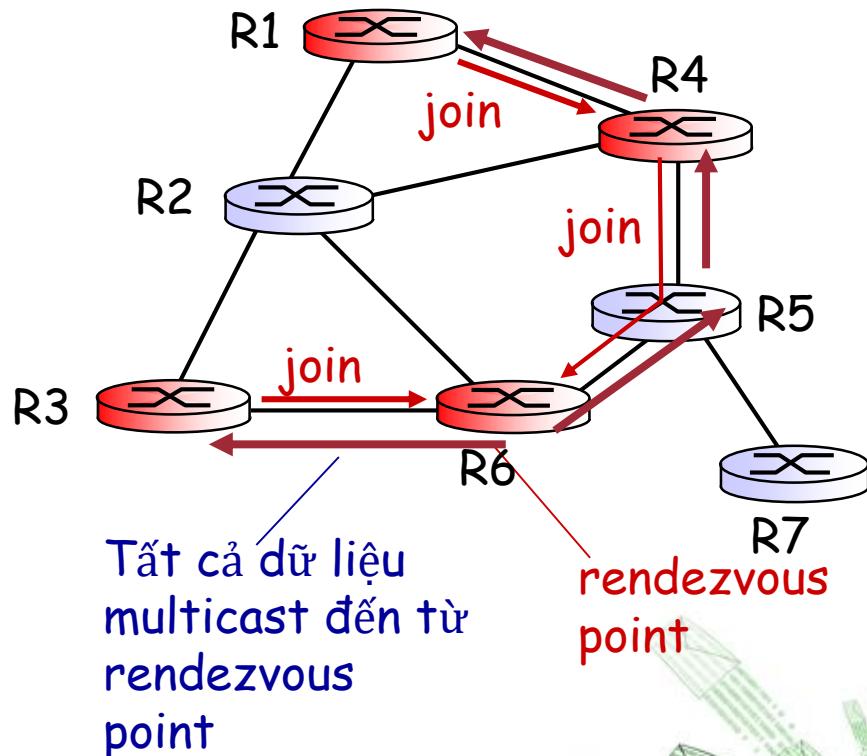
Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing



# Chương 4: Hoàn thành!

4.1 Giới thiệu

4.2 Mạng mạch ảo và mạng chuyên gói (virtual circuit and datagram networks)

4.3 bên trong router

4.4 IP: Internet Protocol

- datagram định dạng, định địa chỉ IPv4, ICMP, IPv6

❖ *Hiểu về các nguyên tắc đằng sau các dịch vụ tầng Mạng:*

- *Các mô hình dịch vụ tầng Mạng, so sánh cách mà router chuyển gói tin và định tuyến, broadcast, multicast*

❖ *Triển khai thực tế trên Internet*

Giới thiệu

Virtual circuit network and diagram network

Cấu trúc trong router

Ip:internet protocol

Thuật toán routing

Broadcast và multicast routing

4.5 các thuật toán định tuyến

- link state, distance vector, định tuyến phân cấp

4.6 định tuyến trong Internet

- RIP, OSPF, BGP

4.7 broadcast and multicast routing



# Mạng máy tính

ThS. Phạm Liệu  
Email: lieu.pham@stu.edu.vn

2018



# NỘI DUNG CHÍNH



Giới Thiệu



Các giao thức đa truy  
cập



Mạng LAN



# Chương 5: Tầng Liên kết dữ liệu

## Mục tiêu:

- Hiểu về các nguyên lý của các dịch vụ tầng Liên kết dữ liệu:
  - Phát hiện lỗi và sửa lỗi
  - Chia sẻ kênh broadcast: đa truy cập
  - Định địa chỉ tầng Liên kết dữ liệu
  - Local Area Networks: Ethernet, VLANs
- Khởi tạo và hiện thực một số công nghệ tầng Liên kết dữ liệu

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# Tầng Liên kết dữ liệu và mạng LAN

## 5.1 Giới thiệu và các dịch vụ

## 5.2 Phát hiện lỗi và sửa lỗi

## 5.3 Các giao thức đa truy cập

## 5.4 Mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANS

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# Tầng Liên kết dữ liệu: Giới thiệu

## Thuật ngữ:

- Host và router: **node**
- Các kênh truyền thông kết nối các node lân cận nhau (adjacent nodes) dọc theo đường truyền thông: **links**
  - Kết nối có dây (wired links)
  - Kết nối không dây (wireless links)
  - LANs
- Gói dữ liệu ở lớp 2: **frame**, đóng gói datagram

**Tầng data-link** có nhiệm vụ truyền datagram từ 1 node đến node lân cận vật lý (*physically adjacent node*) trên một đường kết nối

Giới thiệu và các dịch vụ

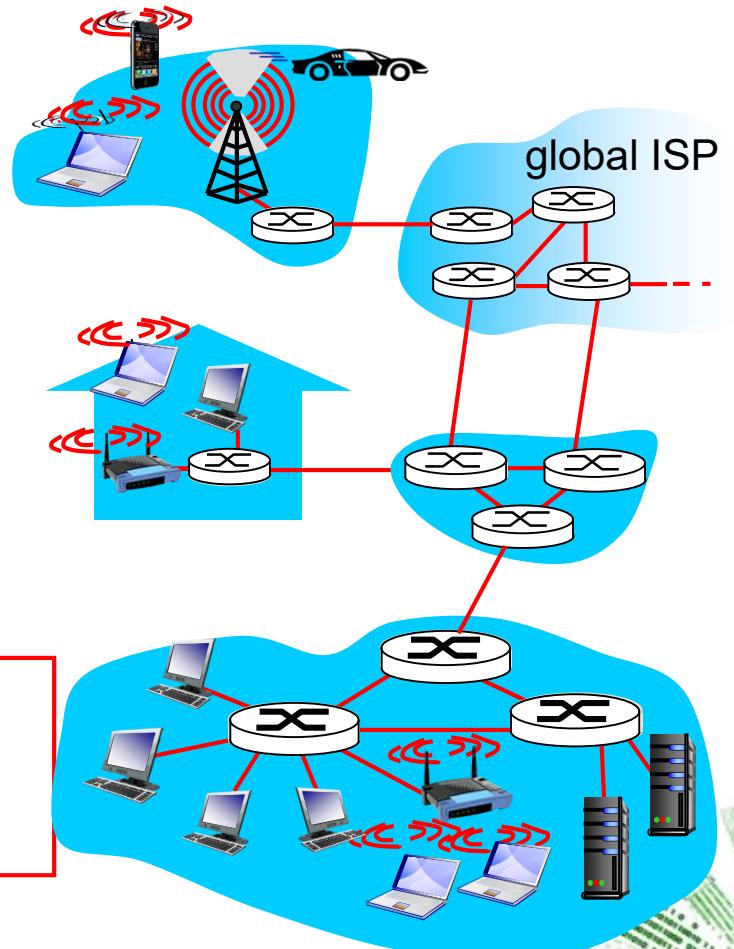
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# Tầng Liên kết dữ liệu: Ngữ cảnh

- Datagram được truyền bởi các giao thức khác nhau thuộc tầng Liên kết dữ liệu trên các đoạn kết nối khác nhau:
  - Ví dụ: Ethernet trên đoạn kết nối thứ 1, frame relay trên các đoạn kết nối trung gian, 802.11 trên đường kết nối cuối cùng
- Mỗi giao thức tầng Liên kết dữ liệu cung cấp các dịch vụ khác nhau
  - Ví dụ: có thể hoặc không thể cung cấp rdt trên đường kết nối

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

## So sánh:

- Hành trình từ Princeton đến Lausanne
  - limo: Princeton đến JFK
  - Máy bay: JFK đến Geneva
  - Xe lửa: Geneva đến Lausanne
- Khách du lịch = **datagram**
- Đoạn đường đi = **liên kết truyền thông (communication link)**
- Kiểu vận chuyển = **giao thức tầng Liên kết dữ liệu**
- Đại lý du lịch = **thuật toán định tuyến**



# Các dịch vụ tầng Liên kết dữ liệu

- *Truy cập đường truyền, đóng gói tin (framing):*
  - Đóng gói datagram vào trong frame, thêm header và trailer
  - Điều khiển truy cập kênh truyền nếu môi trường truyền được chia sẻ
  - Các địa chỉ “MAC” được sử dụng trong các header để xác định nguồn và đích
    - Khác với địa chỉ IP!
- *Truyền tin cậy giữa các node lân cận (adjacent nodes)*
  - Chúng ta đã tìm hiểu làm thế nào để thực hiện điều này ở chương 3!
  - Ít khi được sử dụng trên đường kết nối lỗi thấp (cáp quang, một số loại cáp xoắn)
  - Kết nối không dây: tỷ lệ lỗi cao

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# Các dịch vụ tầng Liên kết dữ liệu (tt)

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

- *Điều khiển luồng (flow control):*

- Điều khiển tốc độ truyền giữa các node gửi và nhận liền kề nhau

- *Phát hiện lỗi (error detection):*

- Lỗi gây ra bởi suy giảm tín hiệu, nhiễu.
- Bên nhận phát hiện sự xuất hiện lỗi:

- thông báo bên gửi truyền lại hoặc bỏ frame đó

- *Sửa lỗi (error correction):*

- Bên nhận xác định và **sửa** các bit lỗi mà không cần phải truyền lại

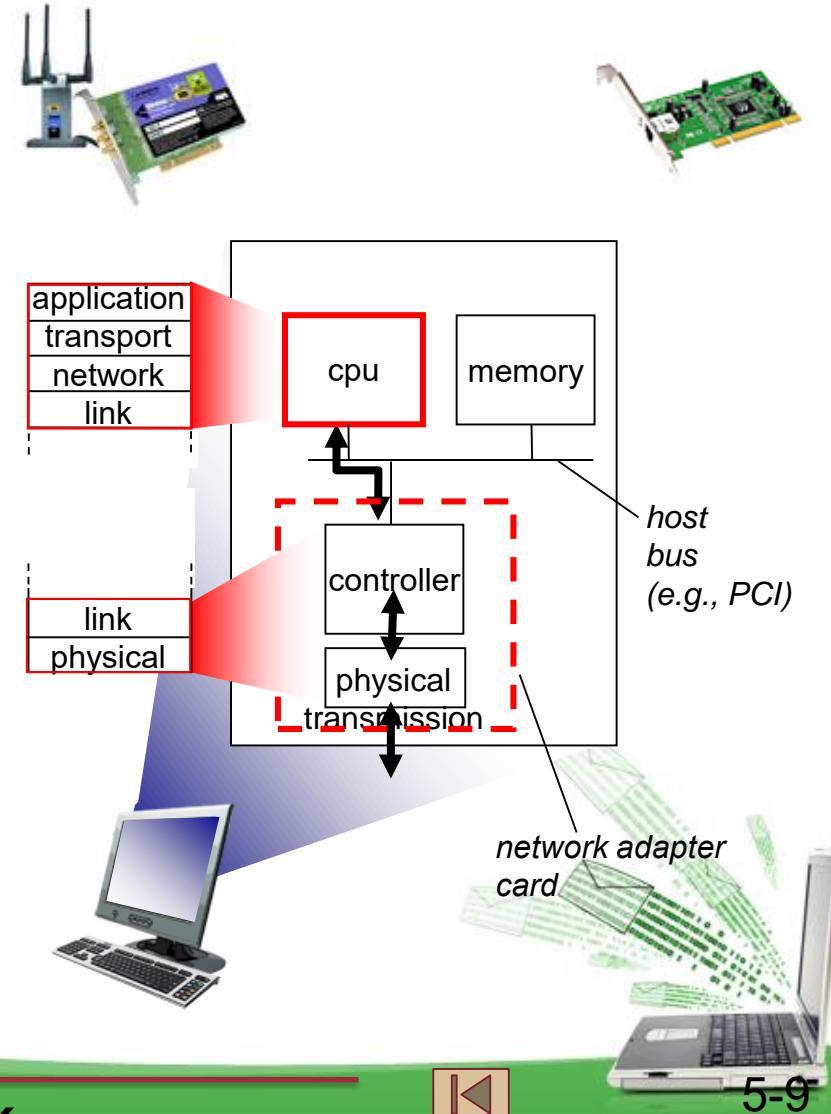
- *Half-duplex và full-duplex*

- Các node tại các đầu cuối của mỗi kết nối đều có thể truyền và nhận, nhưng với half duplex thì thao tác này không được thực hiện đồng thời



# Tầng Liên kết dữ liệu được triển khai ở đâu?

- Trong mỗi và mọi host
- Tầng Liên kết dữ liệu được triển khai trong “adaptor” (còn gọi là *network interface card* NIC)
  - Ethernet card, 802.11 card; Ethernet chipset
  - Triển khai cả tầng Vật lý và tầng Liên kết dữ liệu
- Nối vào trong các bus hệ thống của host
- Sự kết hợp của phần cứng, phần mềm và firmware



# Các Adaptor liên lạc

Giới thiệu và các dịch vụ

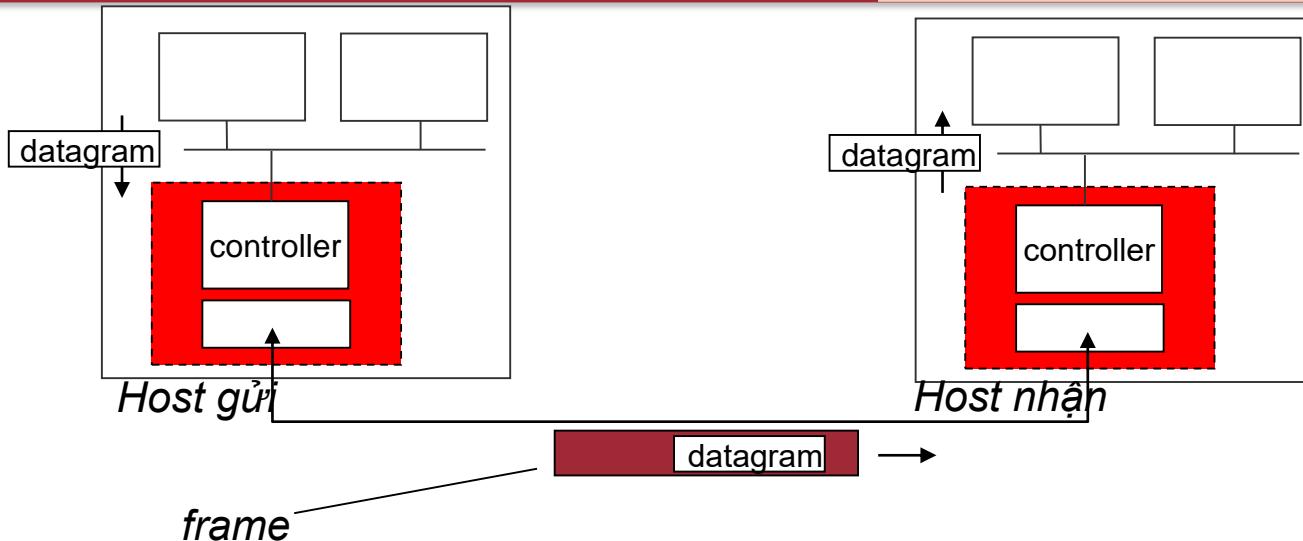
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



- Bên gửi:
  - Đóng gói datagram vào trong frame
  - Thêm các bit kiểm tra lỗi, rdt và điều khiển luồng...

- Bên nhận
  - Tìm lỗi, rdt và điều khiển luồng...
  - Tách các datagram ra, chuyển lên tầng trên tại nơi nhận

# Tầng Liên kết dữ liệu và mạng LAN

5.1 Giới thiệu và các dịch vụ

5.2 Phát hiện lỗi và sửa lỗi

5.3 Các giao thức đa truy cập

5.4 Mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANS

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN





# Phát hiện lỗi

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

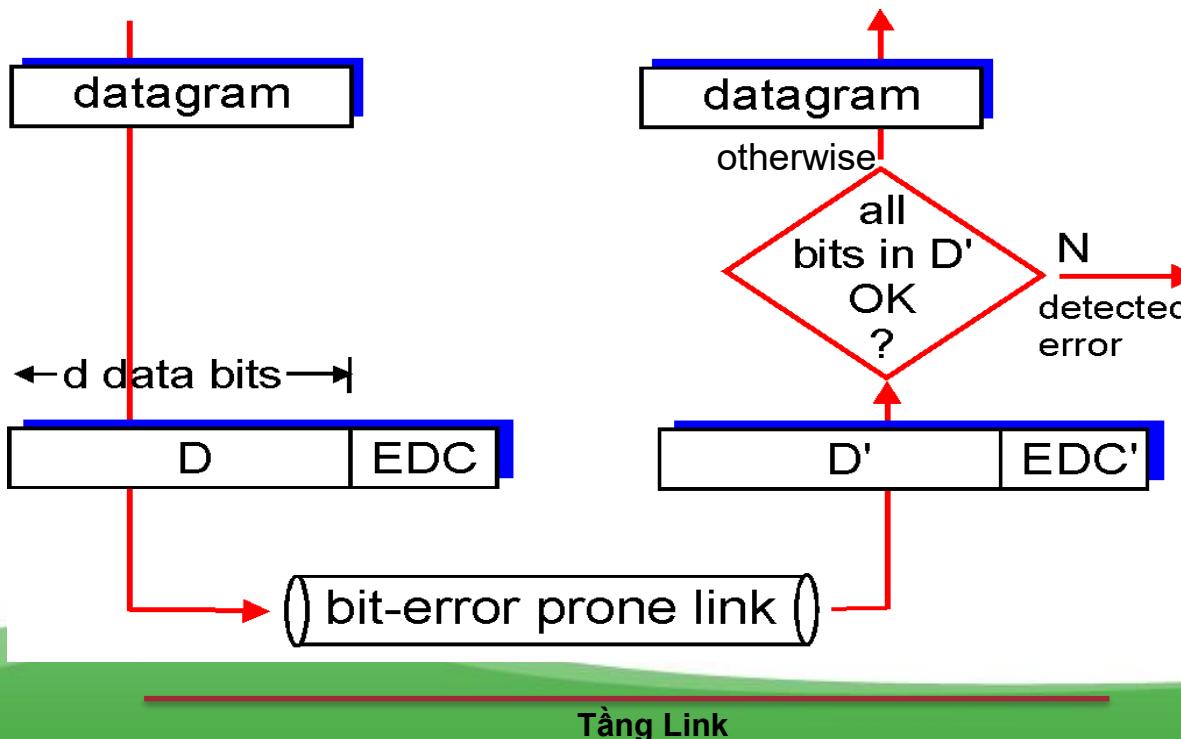
EDC= Error Detection and Correction bits (redundancy)

D = dữ liệu được bảo vệ bởi kiểm tra lỗi, có thể chứa các trường header

- Việc phát hiện lỗi không bảo đảm 100%!

- giao thức có thể bỏ qua một số lỗi, nhưng hiếm khi

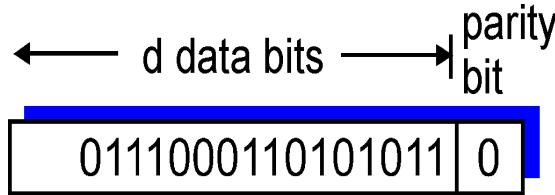
- trường EDC càng lớn thì việc phát hiện và sửa lỗi càng tốt hơn



# Kiểm tra chẵn lẻ (Parity checking)

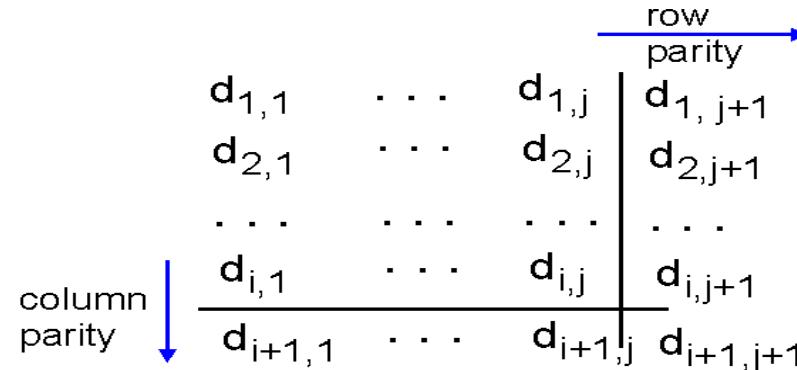
## Bit parity đơn:

- ❖ Phát hiện các lỗi bit đơn



## Bit parity 2 chiều:

- ❖ phát hiện và sửa lỗi các bit đơn



101011  
111100  
011101  
001010

*no errors*

101011  
101100  
011101  
001010

parity error  
parity error  
*correctable single bit error*

# Internet checksum

**Mục tiêu:** phát hiện “các lỗi” (ví dụ, các bit bị đảo) trong packet được truyền (chú ý: chỉ được dùng tại tầng Vận chuyển)

## Bên gửi:

- Xử lý các nội dung của segment như một chuỗi các số nguyên 16-bit
- checksum: là (tổng bù 1) của các nội dung của segment
- Bên gửi đặt giá trị checksum vào trong trường checksum của UDP

## Bên nhận:

- Tính toán checksum của segment vừa nhận
- Kiểm tra xem giá trị của checksum vừa được tính có bằng với giá trị trong trường checksum không:
  - không – phát hiện lỗi
  - có – không phát hiện lỗi. Nhưng có thể còn có lỗi khác không?



# Cyclic redundancy check

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

- Thuật toán phát hiện lỗi tốt hơn
- Xem đoạn bit dữ liệu, **D**, như một số nhị phân
- Chọn mẫu chiều dài  $r+1$  bit (bộ khởi tạo), **G**
- Mục tiêu: chọn  $r$  bit CRC, **R**, sao cho
  - $\langle D, R \rangle$  chia hết cho G (theo cơ số 2)
  - Bên nhận biết G, chia  $\langle D, R \rangle$  cho G. Nếu phần dư khác không: phát hiện lỗi!
  - Có thể phát hiện tất cả các lỗi nhỏ hơn  $r+1$  bits
- Được sử dụng rộng rãi trong thực tế (Ethernet, 802.11 WiFi, ATM)

$\longleftrightarrow$  d bits  $\longrightarrow$   $\longleftrightarrow$  r bits  $\longrightarrow$

**D: data bits to be sent | R: CRC bits**

*bit pattern*

$$D * 2^r \text{ XOR } R$$

*mathematical formula*

# CRC ví dụ

Muốn:

$$D \cdot 2^r \text{ XOR } R = nG$$

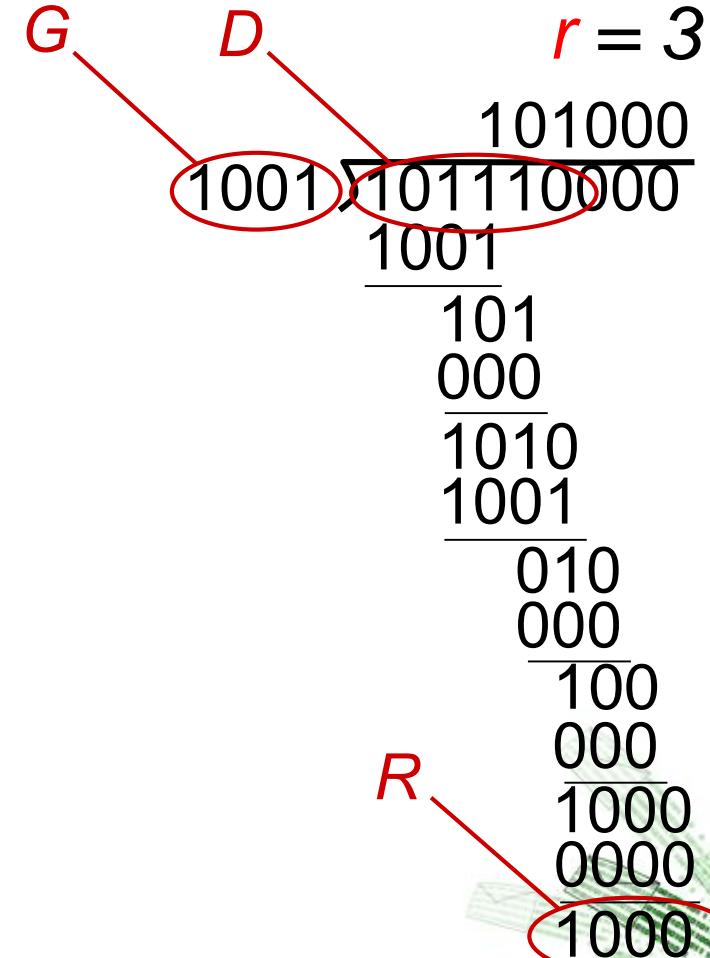
Tương đương:

$$D \cdot 2^r = nG \text{ XOR } R$$

Tương đương:

nếu chúng ta chia  $D \cdot 2^r$  cho  $G$ , có được phần dư  $R$  thỏa:

$$R = \text{remainder} \left[ \frac{D \cdot 2^r}{G} \right]$$



# Tầng Liên kết dữ liệu và mạng LAN

5.1 Giới thiệu và các dịch vụ

5.2 Phát hiện lỗi và sửa lỗi

5.3 Các giao thức đa truy cập

5.4 Mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANS

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# Các giao thức và kết nối đa truy cập

2 kiểu “kết nối”:

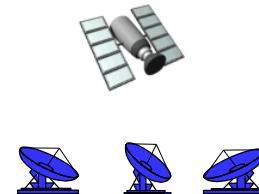
- Điểm-điểm (point-to-point)
  - PPP cho truy cập dial-up
  - Kết nối point-to-point giữa Ethernet switch và host
- *Broadcast (chia sẻ đường truyền dùng chung)*
  - Ethernet mô hình cũ
  - upstream HFC
  - 802.11 wireless LAN



shared wire (e.g.,  
cabled Ethernet)



shared RF  
(e.g., 802.11 WiFi)



shared RF  
(satellite)



Trong buổi tiệc coctail  
(không khí và âm thanh  
được chia sẻ)

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

# Các giao thức MAC: phân loại

3 loại chính:

- *Phân hoạch kênh (channel partitioning)*
  - Chia kênh truyền thành “các mảnh” nhỏ hơn (các slot thời gian - TDM, tần số - FDM, mã - CDM)
  - Cấp phát mảnh này cho node để sử dụng độc quyền
- *Truy cập ngẫu nhiên (random access)*
  - Kênh truyền không được chia, cho phép đụng độ
  - “giải quyết” đụng độ
- *“Xoay vòng”*
  - Các node thay phiên nhau, nhưng nút có nhiều dữ liệu hơn được giữ thời gian truyền lâu hơn

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# Các giao thức MAC phân hoạch kênh: TDMA

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

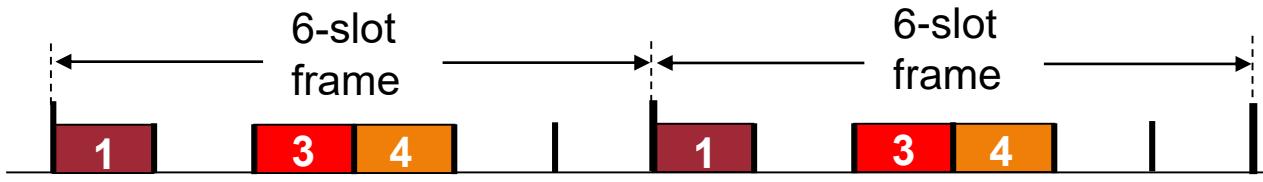
Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

## TDMA: time division multiple access

- Truy cập kênh truyền theo hình thức “xoay vòng”
- Mỗi trạm (station) có slot với độ dài cố định (độ dài = thời gian truyền packet) trong mỗi vòng (round)
- Các slot không sử dụng sẽ nhàn rỗi
- Ví dụ: LAN có 6 trạm, 1,3,4 có gói được gửi, các slot 2,5,6 sẽ nhàn rỗi



# Các giao thức MAC phân hoạch kênh: FDMA

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

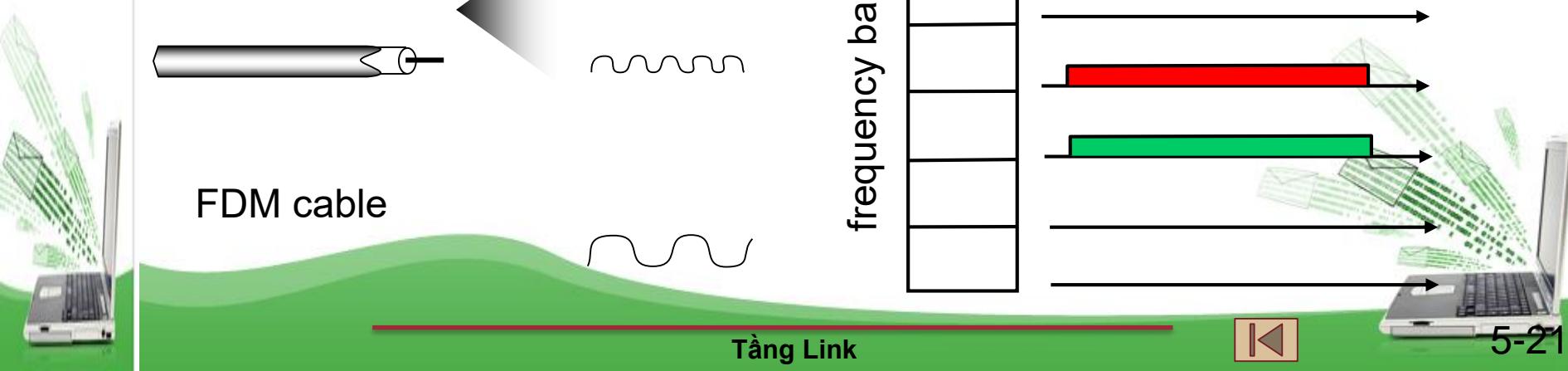
Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

FDMA: frequency division multiple access

- Phổ kênh truyền được chia thành các dải tần số
- Mỗi trạm được gán một dải tần số cố định
- Trong thời gian không truyền, các dải tần rảnh
- Ví dụ: LAN có 6 station, 1,3,4 có packet truyền, các dải tần số 2,5,6 nhàn rỗi



# Các giao thức truy cập ngẫu nhiên

- Khi 1 node có packet cần gửi
  - Truyền dữ liệu với trọng tốc độ của kênh dữ liệu R.
  - Không có sự ưu tiên giữa các node
- 2 hoặc nhiều node truyền → “đụng độ”,
- **Giao thức truy cập ngẫu nhiên MAC** xác định:
  - Cách để phát hiện đụng độ
  - Cách để giải quyết đụng độ (ví dụ: truyền lại sau đó)
- Ví dụ các giao thức MAC truy cập ngẫu nhiên:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# Slotted ALOHA

## Giả thuyết:

- Tất cả các frame có cùng kích thước
- Thời gian được chia thành các slot có kích thước bằng nhau (thời gian để truyền đủ 1 frame)
- Các node bắt đầu truyền chỉ ngay tại lúc bắt đầu slot
- Các node được đồng bộ hóa
- Nếu 2 hoặc nhiều node truyền trong cùng 1 slot, thì tất cả các node đều phát hiện đụng độ

## Hoạt động:

- Khi node có được frame mới, nó sẽ truyền trong slot kế tiếp
  - Nếu không có đụng độ: node có thể gửi frame mới trong slot kế tiếp
  - Nếu có đụng độ: node truyền lại frame trong mỗi slot tiếp theo với xác suất p cho đến khi thành công



# Slotted ALOHA

Giới thiệu và các dịch vụ

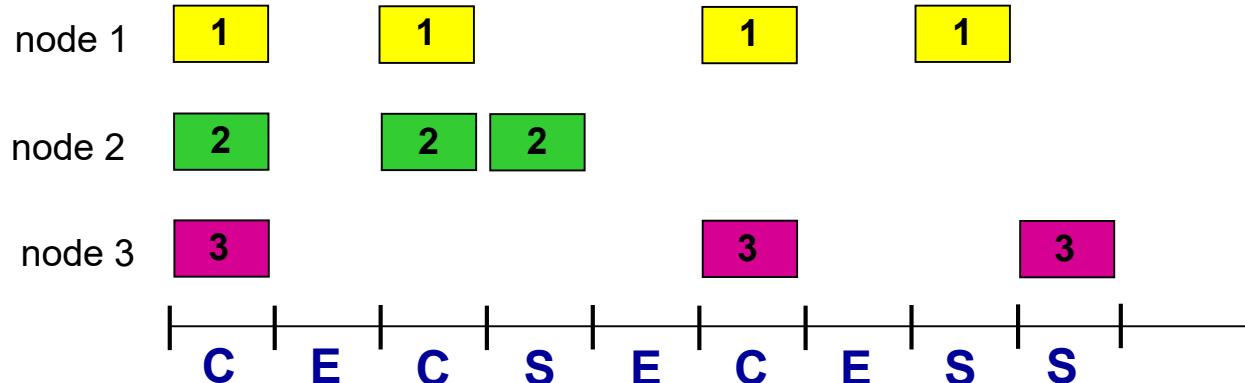
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



## Ưu điểm:

- Node hoạt động có thể truyền liên tục với tốc độ tối đa của kênh
- Phân tán cao: chỉ có các slot trong các node cần được đồng bộ
- Đơn giản

## Nhược điểm:

- Đụng độ, lãng phí slot
- Các slot trống
- Các node có thể phát hiện tranh chấp nhanh hơn thời gian để truyền gói
- Đồng bộ hóa



# Slotted ALOHA: hiệu suất

**Hiệu suất**: là phần slot truyền thành công trong số nhiều frame dự định truyền của nhiều node

- **Giả sử**: có  $N$  node với nhiều frame để truyền, mỗi frame truyền trong slot với xác suất là  $p$
- Xác suất để 1 node truyền thành công trong 1 slot =  $p(1-p)^{N-1}$
- Xác suất để bất kỳ node nào truyền thành công =  $Np(1-p)^{N-1}$

- Hiệu suất cực đại: tìm  $p^*$  làm cực đại hóa  $Np(1-p)^{N-1}$
- Với nhiều node, tìm giới hạn của  $Np^*(1-p^*)^{N-1}$  khi  $N \rightarrow \infty$ , cho:

$$\begin{aligned}\text{hiệu suất cực đại} &= \\ 1/e &= .37\end{aligned}$$

**Tốt nhất**: kênh hữu dụng trong khoảng 37% thời gian!



# Pure ALOHA: hiệu suất

$$\begin{aligned} P(\text{thành công cho node}) &= P(\text{node truyền}) \times \\ &\quad P(\text{không có node khác truyền trong } [t_0-1, t_0] \times \\ &\quad P(\text{không có node khác truyền trong } [t_0, t_0+1]) \end{aligned}$$

$$\begin{aligned} &= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1} \\ &= p \cdot (1-p)^{2(N-1)} \end{aligned}$$

... chọn  $p$  tối ưu và sau đó cho  $n \rightarrow \infty$

$$= 1/(2e) = .18$$

## Thậm chí tệ hơn slotted Aloha!



# CSMA (carrier sense multiple access)

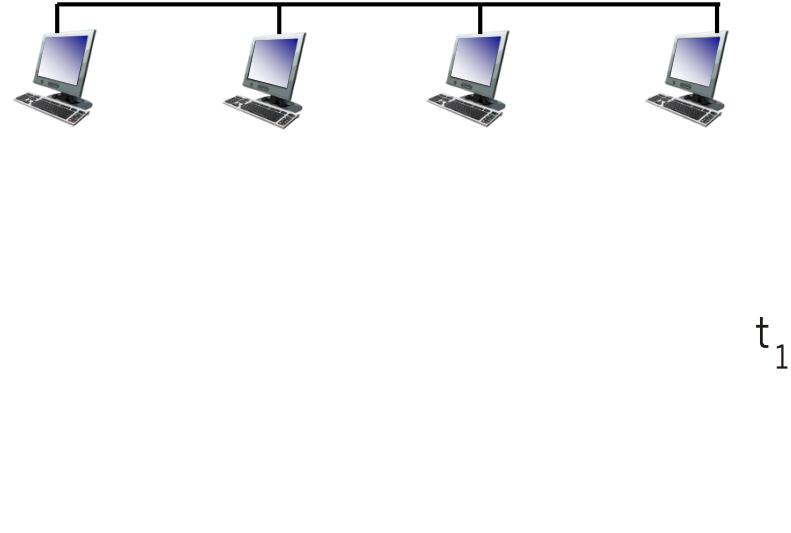
**CSMA:** lắng nghe trước khi truyền:

- Nếu kênh nhàn rỗi: truyền toàn bộ frame
- Nếu kênh truyền bận, trì hoãn truyền
- So sánh với con người: dừng ngắt lời người khác!



# CSMA: đụng độ (collision)

- **Đụng độ có thể vẫn xảy ra:** trẽ lan truyền nghĩa là 2 node không thể nghe thấy quá trình truyền lẫn nhau
- **Đụng độ:** toàn bộ thời gian truyền packet bị lãng phí
  - Khoảng cách và trẽ lan truyền có vai trò trong việc xác định xác suất đụng độ



Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# CSMA: dụng độ (collision)

*Đụng độ có thể vẫn xảy ra: trễ lan truyền nghĩa là 2 node không thể nghe thấy quá trình truyền lấn nhau*

*Đụng độ: toàn bộ thời gian truyền packet bị lãng phí*

*Khoảng cách và trễ lan truyền có vai trò trong việc xác định xác suất đụng độ*

Giới thiệu và các dịch vụ

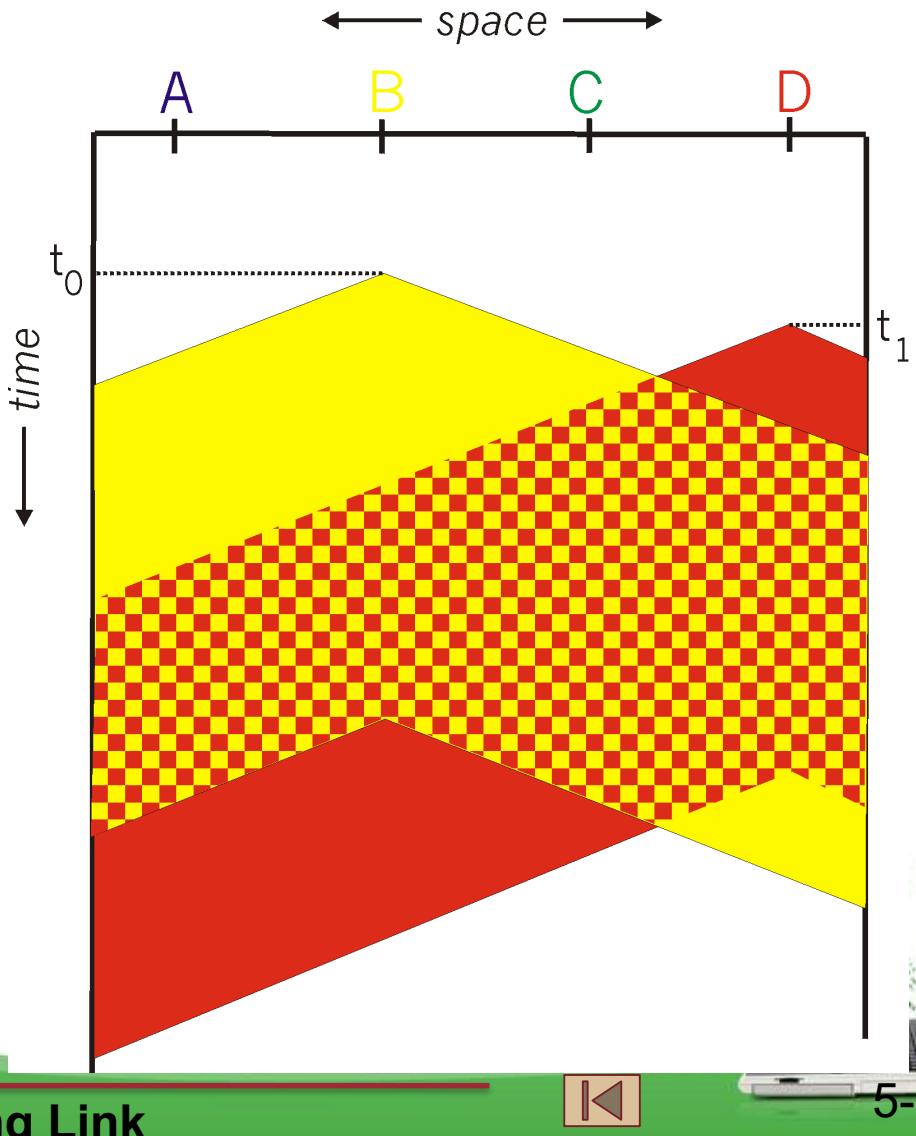
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# CSMA/CD (collision detection)

**CSMA/CD:** trì hoãn như trong CSMA

- Đụng độ được phát hiện trong thời gian ngắn
- Thông tin đang truyền bị hủy bỏ, giảm sự lãng phí kênh.
- Phát hiện đụng độ:
  - Dễ dàng trong các mạng LAN hữu tuyến: đo cường độ tín hiệu, so sánh với các tín hiệu đã được truyền và nhận
  - Khó thực hiện trong mạng LAN vô tuyến: cường độ truyền cục bộ lấn át cường độ tín hiệu nhận
- Tương tự như hành vi của con người: đàm thoại lịch sự

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



# CSMA/CD (collision detection)

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

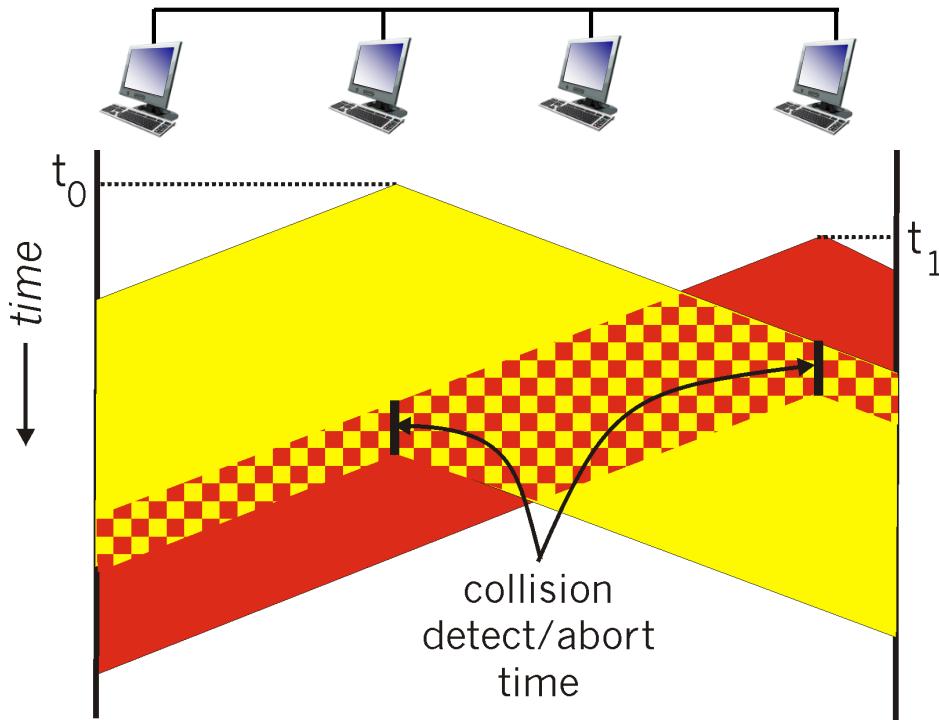
Các giao thức đa truy cập

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

Bố trí của các node



# Thuật toán Ethernet CSMA/CD

1. NIC nhận datagram từ tầng network, tạo frame
2. Nếu NIC dò được kênh rỗi, nó sẽ bắt đầu việc truyền frame. Nếu NIC dò được kênh bận, đợi cho đến khi kênh rảnh, sau đó mới truyền.
3. Nếu NIC truyền toàn bộ frame mà không phát hiện việc truyền khác, NIC được truyền toàn bộ frame đó!
4. Nếu NIC phát hiện có phiên truyền khác trong khi đang truyền, thì nó sẽ hủy bỏ việc truyền và phát tín hiệu tắc nghẽn
5. Sau khi hủy bỏ truyền, NIC thực hiện *binary (exponential) backoff:*
  - Sau lần đụng độ thứ  $m$ , NIC chọn ngẫu nhiên số  $K$  trong khoảng  $\{0, 1, 2, \dots, 2^m - 1\}$ . NIC sẽ đợi  $K \cdot 512$  thời gian truyền bit (bit time), sau đó trở lại bước 2
  - Càng nhiều đụng độ thì sẽ có khoảng thời gian chờ dài hơn

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN



# CSMA/CD hiệu suất

[Giới thiệu và các dịch vụ](#)[Phát hiện lỗi và sửa lỗi](#)[Các giao thức đa truy cập](#)[Link virtualization: MPLS](#)[Mạng trung tâm dữ liệu](#)[Hoạt động thường ngày - truy cập web](#)[Mạng LAN](#)

- $T_{prop}$  = độ trễ lan truyền lớn nhất (max prop delay) giữa 2 node trong mạng LAN
- $t_{trans}$  = thời gian để truyền frame có kích thước lớn nhất

$$\text{efficiency} = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- Hiệu suất tiến tới 1
  - khi  $t_{prop}$  tiến tới 0
  - khi  $t_{trans}$  tiến tới vô cùng
- Hiệu suất tốt hơn ALOHA: đơn giản, chi phí thấp và điều khiển phân tán!



# Các giao thức MAC “Xoay vòng”

Các giao thức phân hoạch kênh MAC (channel partitioning MAC protocols):

- Chia sẻ kênh hiệu quả và công bằng với tải lớn
- Không hiệu quả ở tải thấp: trễ khi truy cập kênh, 1 node chỉ được cấp phát  $1/N$  bandwidth ngay cả khi chỉ có 1 node hoạt động!

Các giao thức MAC truy cập nhẫu nhiên (random access MAC protocols)

- Hiệu quả tại tải thấp: node đơn có thể dùng hết khả năng của kênh
- Tải cao: đụng độ cao

Các giao thức “Xoay vòng” (“taking turns” protocols)

Tìm kiếm giải pháp dung hòa tốt nhất!

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Link virtualization: MPLS

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN

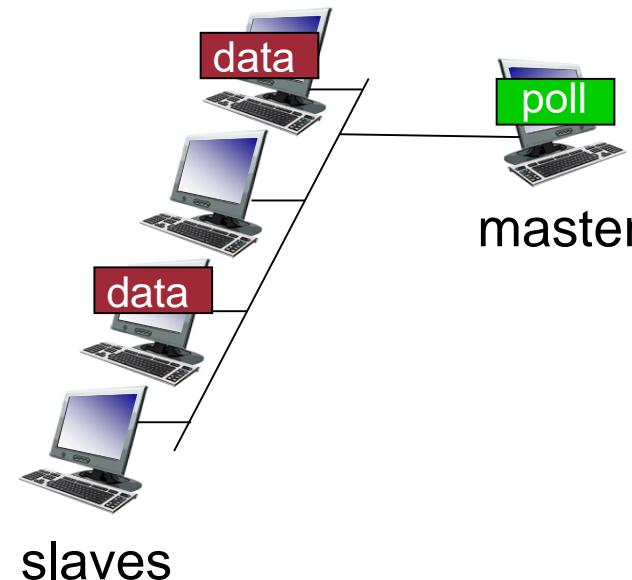


# Các giao thức MAC

## “Xoay vòng”

### Polling:

- Node chủ (master node) “mời” các node con (slave node) truyền lần lượt
- Thường được sử dụng với các thiết bị con “không thông minh”
- Quan tâm:
  - Chi phí cho việc điều phối (polling overhead)
  - Độ trễ (latency)
  - Có 1 điểm chịu lỗi (master)



Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Link virtualization: MPLS

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

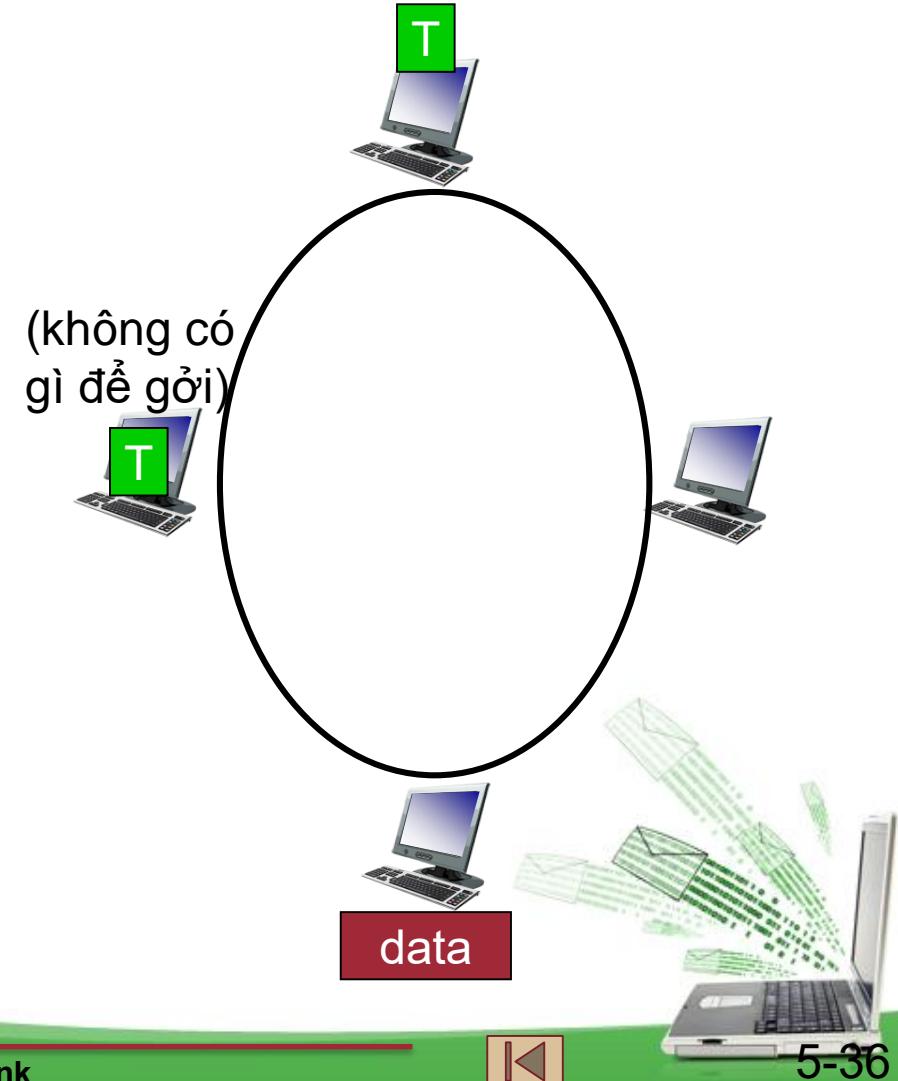
Mạng LAN



# Các giao thức MAC “Xoay vòng”

## *Chuyển token:*

- ❖ Điều hành việc chuyển *token* tuần tự từ 1 node đến node kế tiếp.
- ❖ Gói token
- ❖ Quan tâm:
  - Chi phí cho việc chuyển token (token overhead)
  - Độ trễ (latency)
  - Có 1 điểm chịu lỗi (token)



# Mạng truy cập cáp (Cable access network)

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

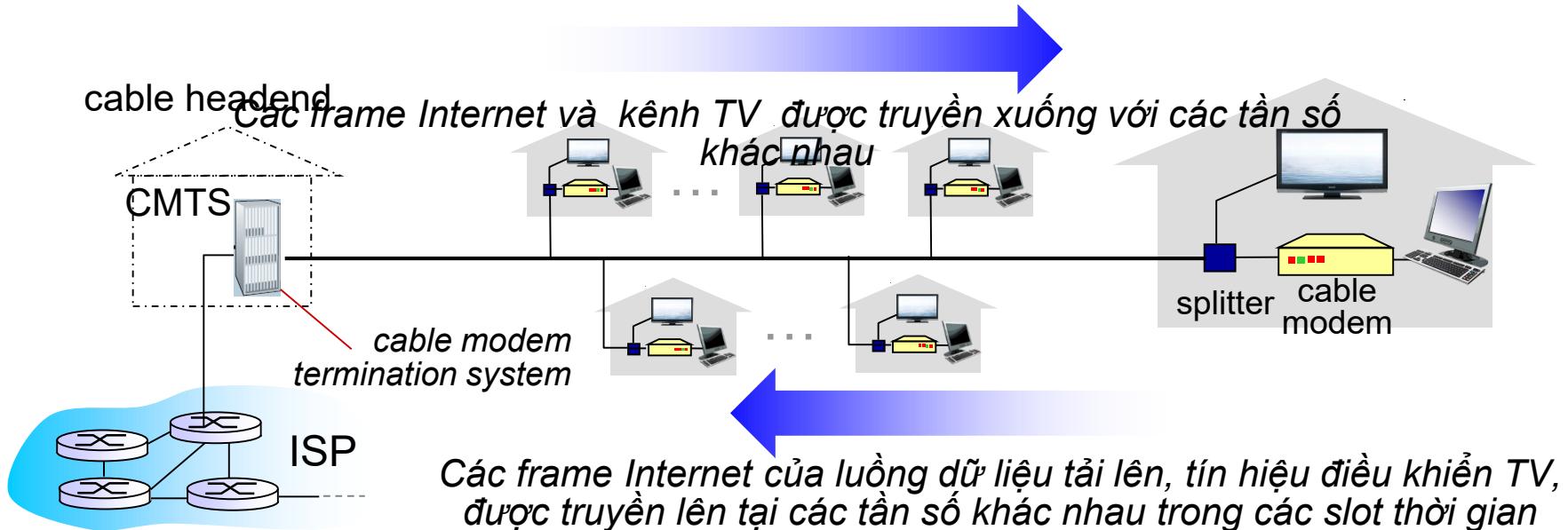
Các giao thức đa truy cập

Link virtualization: MPLS

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



- ❖ Nhiều kênh tải xuống 40Mbps
  - Một phiên truyền đơn từ trạm đầu cuối (CMTS) truyền vào trong các kênh
- ❖ Nhiều kênh tải lên 30 Mbps
  - **Đa truy cập:** tất cả các user tranh giành các slot thời gian trên kênh tải lên

# Mạng truy cập cáp

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

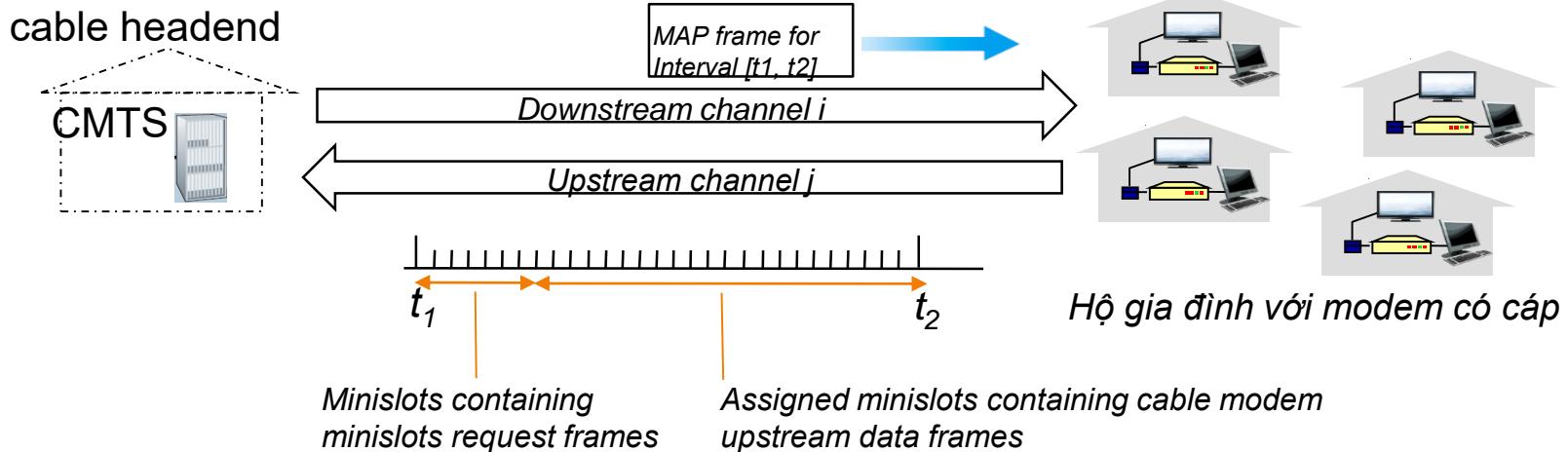
Các giao thức đa truy cập

Link virtualization: MPLS

Mạng trung tâm dữ liệu

Hoạt động thường ngày - truy cập web

Mạng LAN



**DOCSIS:** data over cable service interface spec

- ❖ FDM trên các kênh tần số luồng dữ liệu lên và xuống
- ❖ TDM luồng dữ liệu tải lên (upstream): một số slot được gán, một số slot có tranh chấp
  - MAP frame luồng dữ liệu tải xuống: gán cho các slot thuộc luồng dữ liệu tải lên
  - Yêu cầu cho các slot luồng dữ liệu tải lên (và dữ liệu) được truyền theo dạng truy cập ngẫu nhiên (binary backoff) trong các slot đã được chọn



# Tổng kết các giao thức MAC

[Giới thiệu và các dịch vụ](#)[Phát hiện lỗi và sửa lỗi](#)[Các giao thức đa truy cập](#)[Link virtualization: MPLS](#)[Mạng trung tâm dữ liệu](#)[Hoạt động thường ngày - truy cập web](#)[Mạng LAN](#)

- *Phân hoạch kênh*, theo thời gian, tần số hoặc mã
  - Phân chia theo thời gian (Time Division), phân chia theo tần số (Frequency Division)
- *Truy cập ngẫu nhiên* (động),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - Cảm nhận sóng mang (carrier sensing): dễ dàng trong một số kỹ thuật (có dây), khó thực hiện trong các công nghệ khác (không dây)
  - CSMA/CD được dùng trong Ethernet
  - CSMA/CA được dùng trong 802.11
- *Xoay vòng*
  - Điều phối từ đơn vị trung tâm, truyền token
  - bluetooth, FDDI, token ring



# Tầng Liên kết dữ liệu và mạng LAN

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

Link virtualization: MPLS

## 5.1 Giới thiệu và các dịch vụ

## 5.2 phát hiện lỗi và sửa lỗi

## 5.3 các giao thức đa truy cập

## 5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANS



# Địa chỉ MAC và ARP

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

Link virtualization: MPLS

- Địa chỉ IP 32-bit:
  - Địa chỉ tầng network cho interface
  - Được sử dụng trong chức năng chuyển dữ liệu tầng 3 (tầng network)
- Địa chỉ MAC (hoặc địa chỉ LAN hoặc physical hoặc Ethernet) :
  - Chức năng: **được sử dụng “cục bộ” để chuyển frame** từ 1 interface này đến 1 interface được kết nối vật lý trực tiếp với nhau (cùng mạng, trong ngữ cảnh vùng địa chỉ IP)
  - Địa chỉ MAC 48 bit (cho hầu hết các mạng LAN) được ghi vào trong NIC ROM, hoặc thiết lập trong phần mềm
  - Ví dụ: 1A-2F-BB-76-09-AD

Ghi dưới dạng số thập lục phân (hệ 16)  
(mỗi “số” đại diện 4 bit)



# Địa chỉ MAC và ARP

Giới thiệu và các dịch vụ

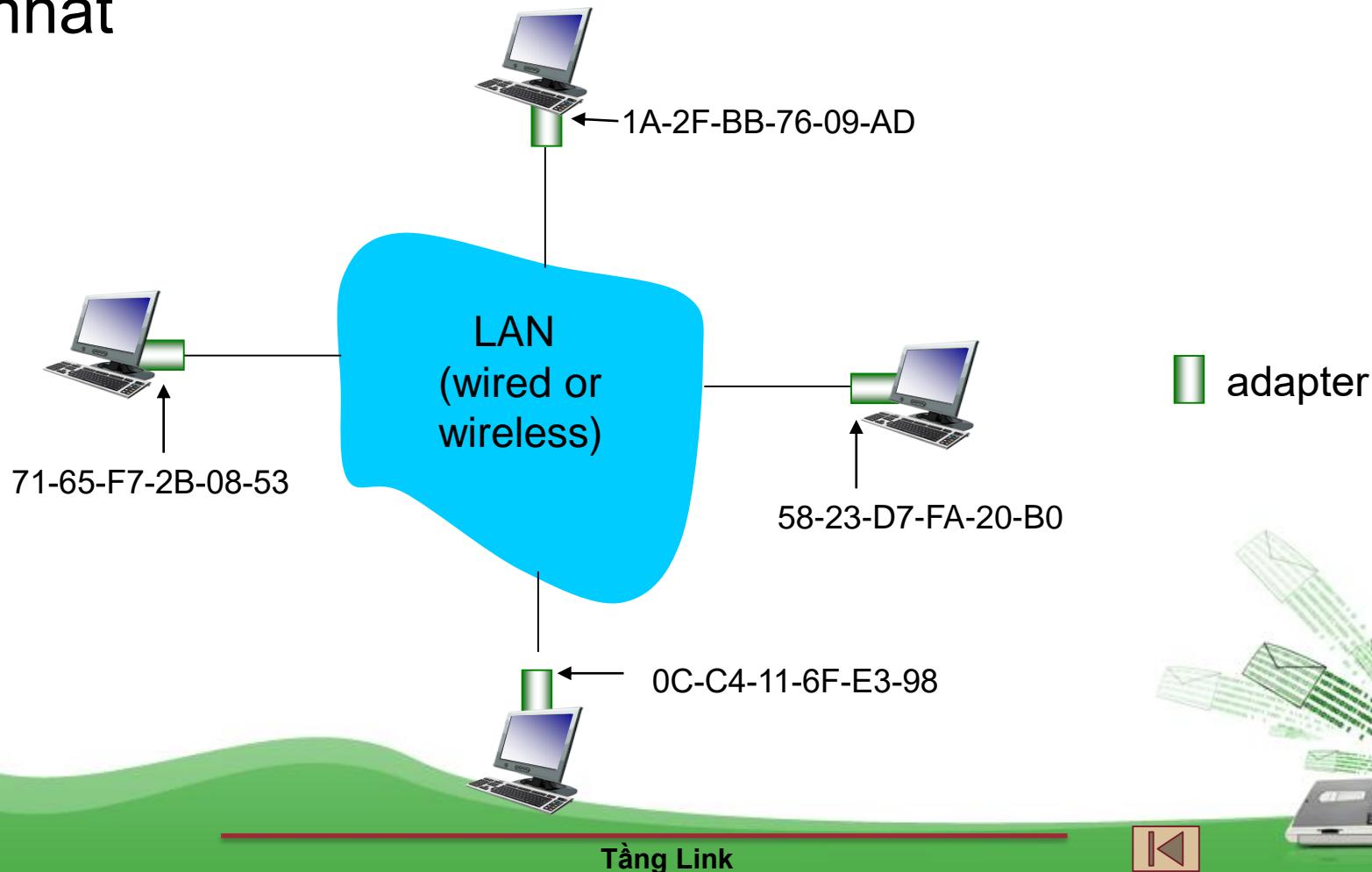
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

Link virtualization: MPLS

Mỗi adapter trên mạng LAN có địa chỉ *LAN* duy nhất



# Địa chỉ LAN(tt)

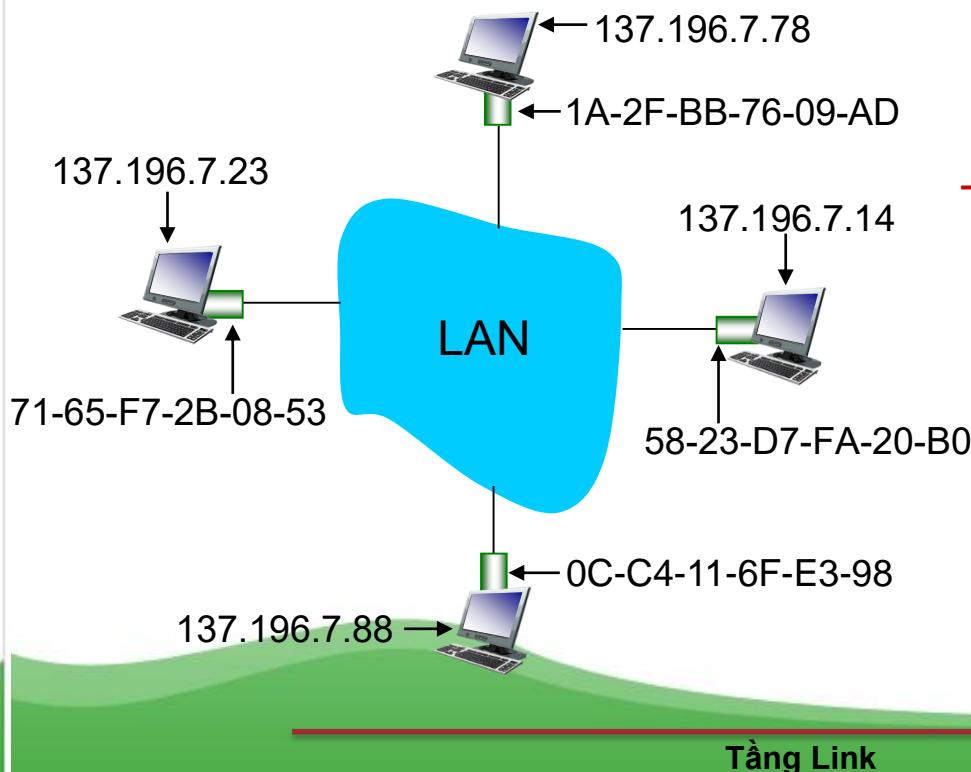
[Giới thiệu và các dịch vụ](#)[Phát hiện lỗi và sửa lỗi](#)[Các giao thức đa truy cập](#)[Mạng LAN](#)[Link virtualization: MPLS](#)

- Sự phân bổ địa chỉ MAC được quản lý bởi IEEE
- Nhà sản xuất mua phần không gian địa chỉ MAC (bảo đảm duy nhất)
- So sánh:
  - Địa chỉ MAC: như là số chứng minh nhân dân
  - Địa chỉ IP: như là địa chỉ bưu điện
- Địa chỉ MAC phẳng & có tính di chuyển
  - Có thể di chuyển card LAN từ 1 mạng LAN này tới mạng LAN khác
- Địa chỉ IP phân cấp, không di chuyển được
  - Địa chỉ phụ thuộc vào subnet IP mà node đó gắn vào



# ARP: address resolution protocol

**Hỏi:** làm cách nào để xác định địa chỉ MAC của interface khi biết được địa chỉ IP của nó?



Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

Link virtualization: MPLS

**Bảng ARP:** mỗi node IP (host, router) trên mạng LAN có bảng ARP

- Địa chỉ IP/MAC ánh xạ cho các node trong mạng LAN:  
*< địa chỉ IP; địa chỉ MAC; TTL>*
- TTL (Time To Live): thời gian sau đó địa chỉ ánh xạ sẽ bị lãng quên (thông thường là 20 phút)

# Giao thức ARP: cùng mạng LAN

- A muốn gửi datagram tới B
  - Địa chỉ MAC của B không có trong bảng ARP của A.
- A sẽ **quảng bá (broadcasts)** ARP query packet có chứa địa chỉ IP của B
  - Địa chỉ MAC đích = FF-FF-FF-FF-FF-FF
  - Tất cả các node trên mạng LAN sẽ nhận ARP query này
- B nhận ARP packet, trả lời tới A với địa chỉ MAC của B
  - frame được gửi tới địa chỉ MAC của A (unicast)
- A sẽ lưu lại cặp địa chỉ IP-MAC trong bảng ARP của nó cho tới khi thông tin này trở nên cũ (hết hạn sử dụng)
  - soft state: thông tin hết hạn (bỏ đi) trừ khi được làm mới
- ARP là “plug-and-play”:
  - Các nodes tạo bảng ARP của nó không cần sự can thiệp của người quản trị mạng

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN



# Addressing: định tuyến tới mạng LAN khác

Giới thiệu và các dịch vụ

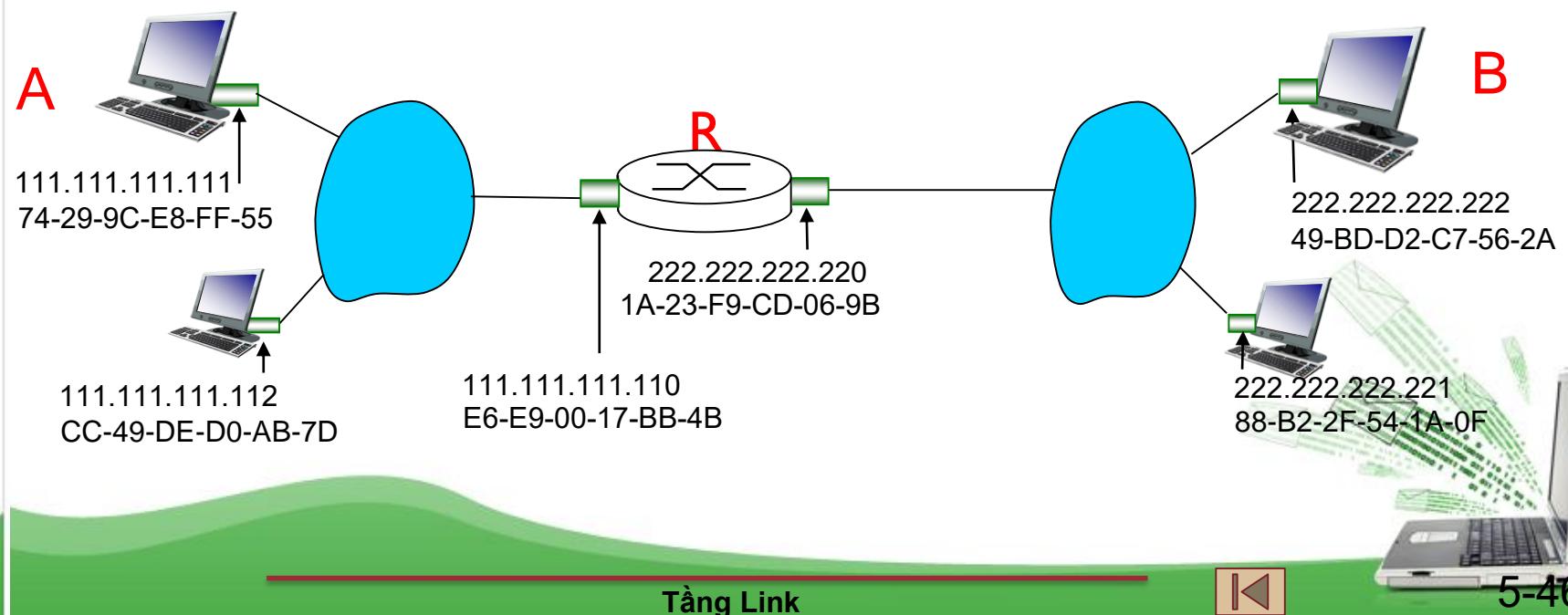
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

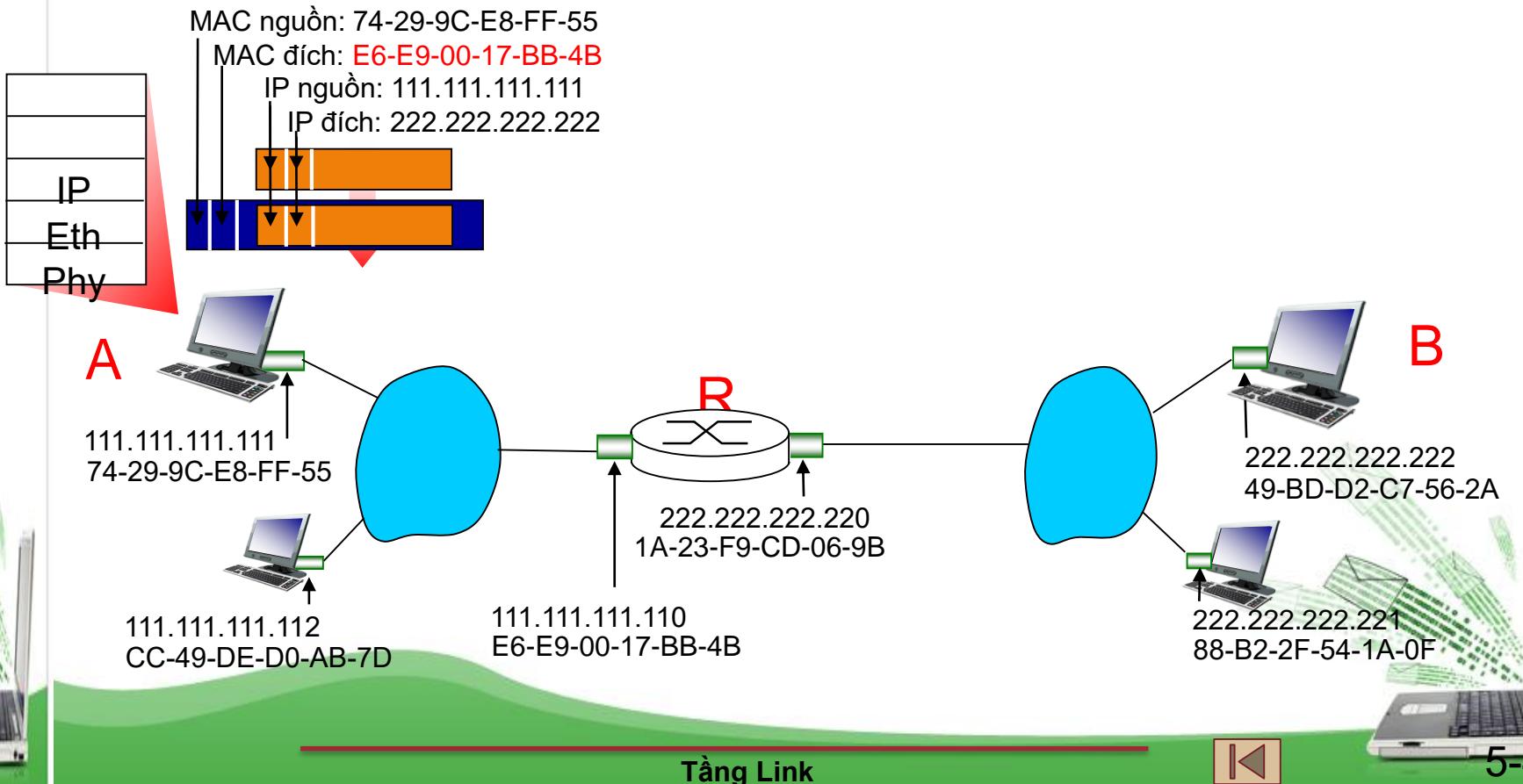
Từng bước: gửi datagram từ A tới B thông qua R

- tập trung vào gán địa chỉ – tại tầng IP (datagram) và MAC (frame)
- giả sử A biết địa chỉ IP của B
- giả sử A biết địa chỉ IP của router gateway, R (cách nào?)
- giả sử A biết địa chỉ MAC của R (cách nào?)



# Addressing: định tuyến tới mạng LAN khác

- ❖ A tạo IP datagram với IP nguồn A, đích B
- ❖ A tạo frame tầng Liên kết dữ liệu với địa chỉ MAC của R là địa chỉ đích, frame này chứa IP datagram từ A tới B



Giới thiệu và các dịch vụ

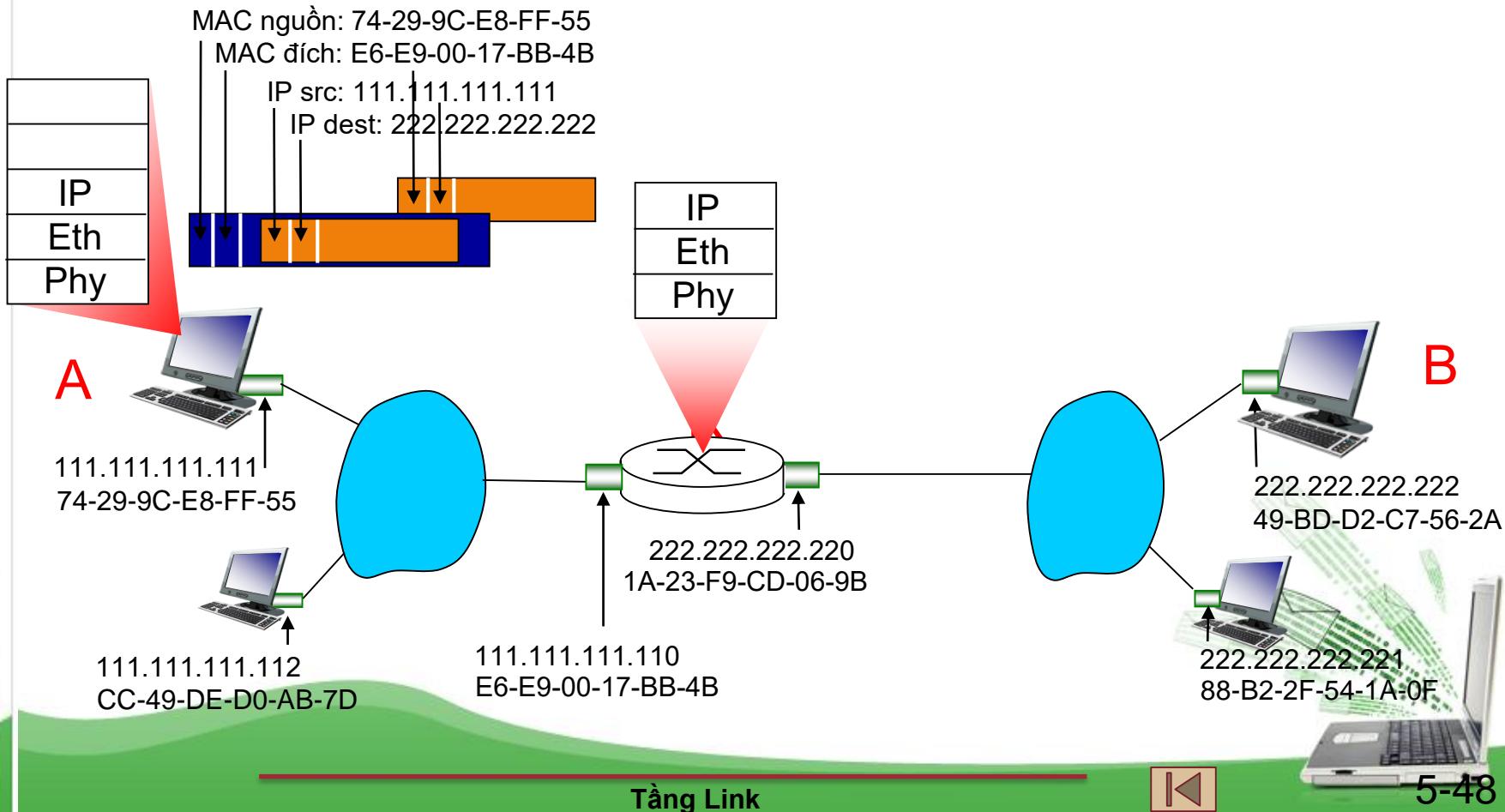
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

# Addressing: định tuyến tới mạng LAN khác

- Frame được gửi từ A tới R
- Frame được R nhận, datagram được gỡ bỏ, được chuyển tới IP



Giới thiệu và các dịch vụ

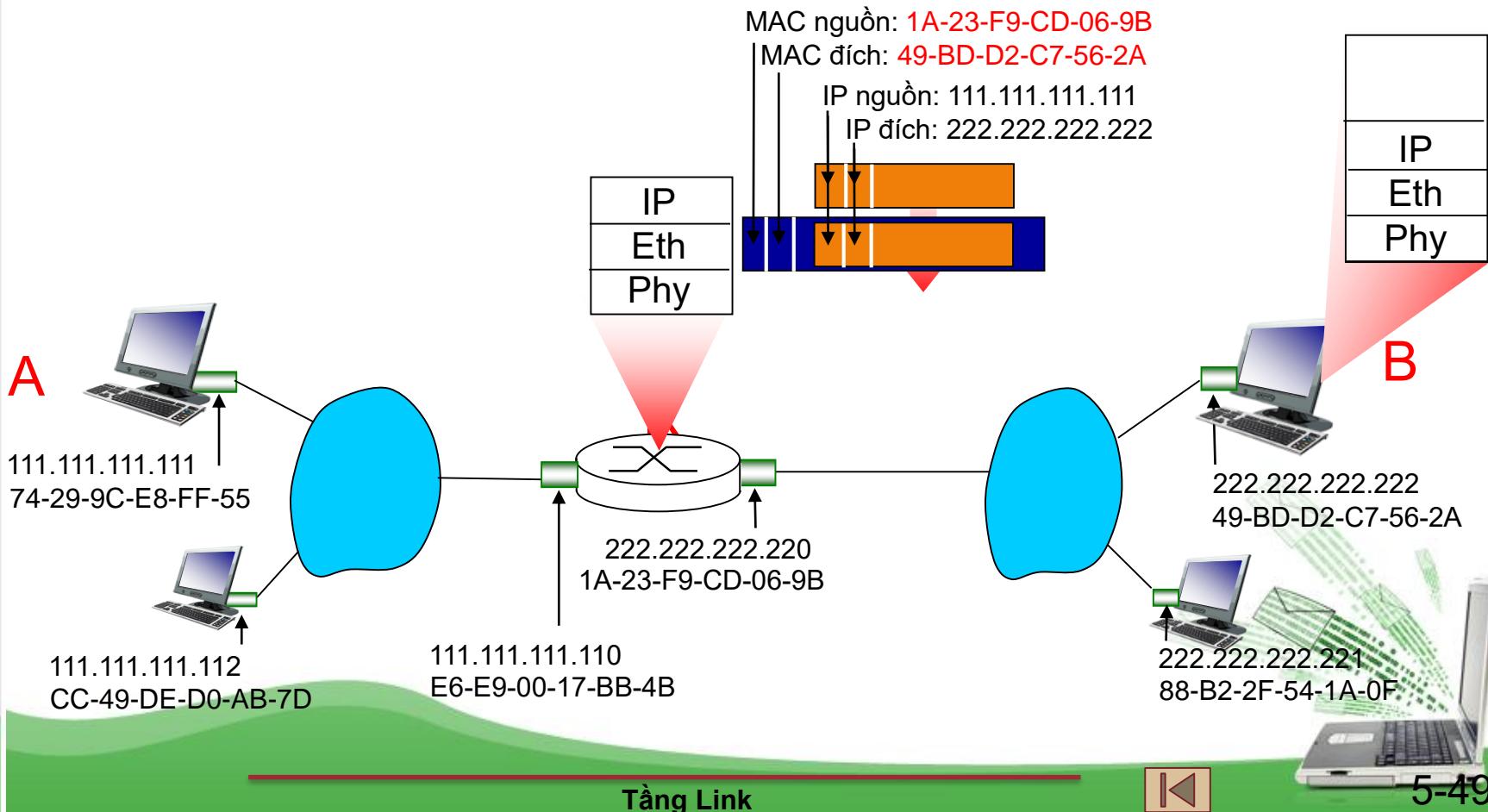
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

# Addressing: định tuyến tới mạng LAN khác

- ❖ R sẽ chuyển tiếp datagram với IP nguồn A, đích B
- ❖ R tạo frame tầng Liên kết dữ liệu với địa chỉ MAC của B là địa chỉ đích, frame này chứa IP datagram từ A tới B



Giới thiệu và các dịch vụ

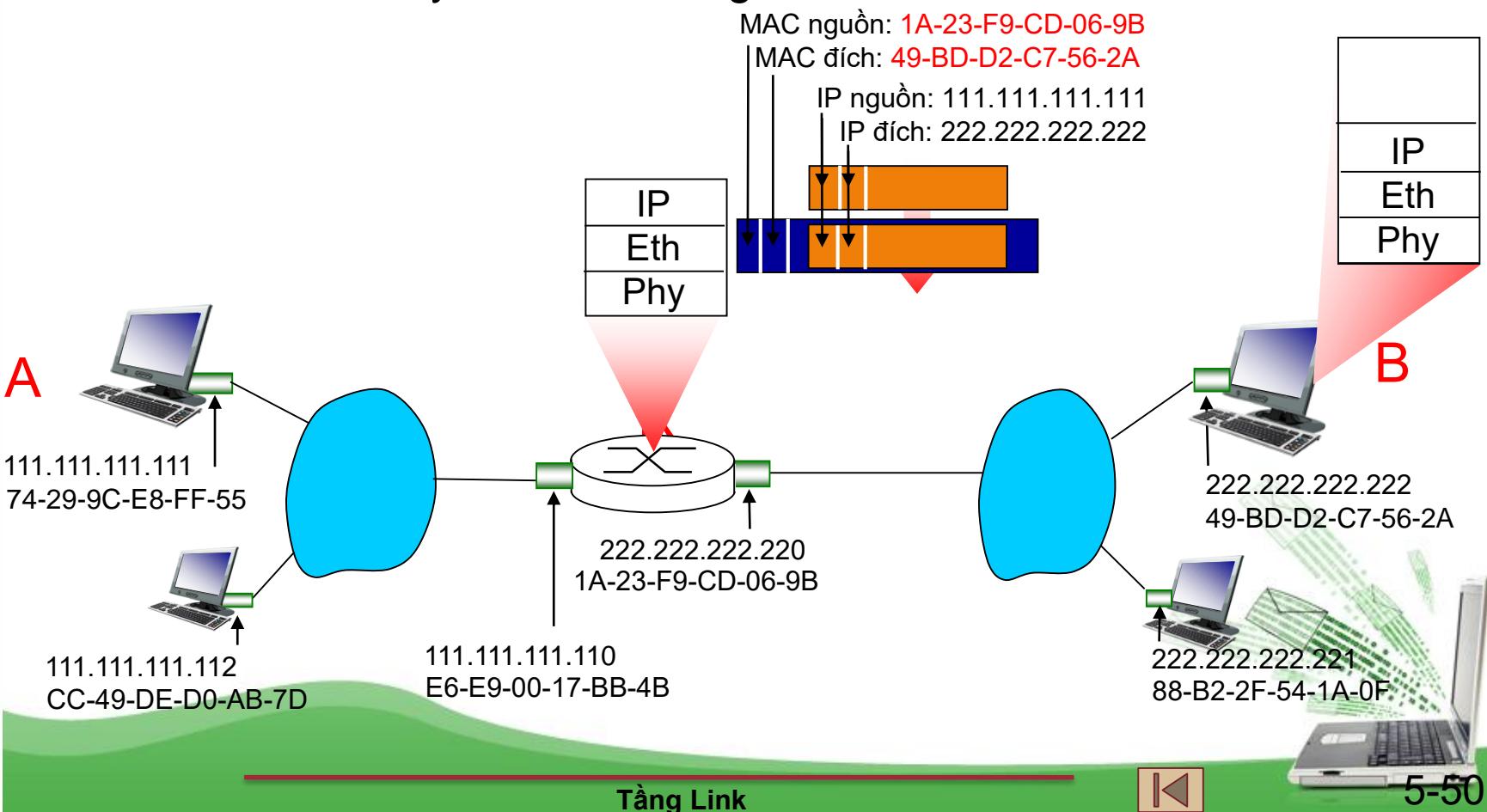
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

# Addressing: định tuyến tới mạng LAN khác

- ❖ R chuyển tiếp datagram với IP nguồn A, đích B
- ❖ R tạo frame tầng Liên kết dữ liệu với địa chỉ MAC của B là địa chỉ đích, frame này chứa IP datagram từ A-tới-B



Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

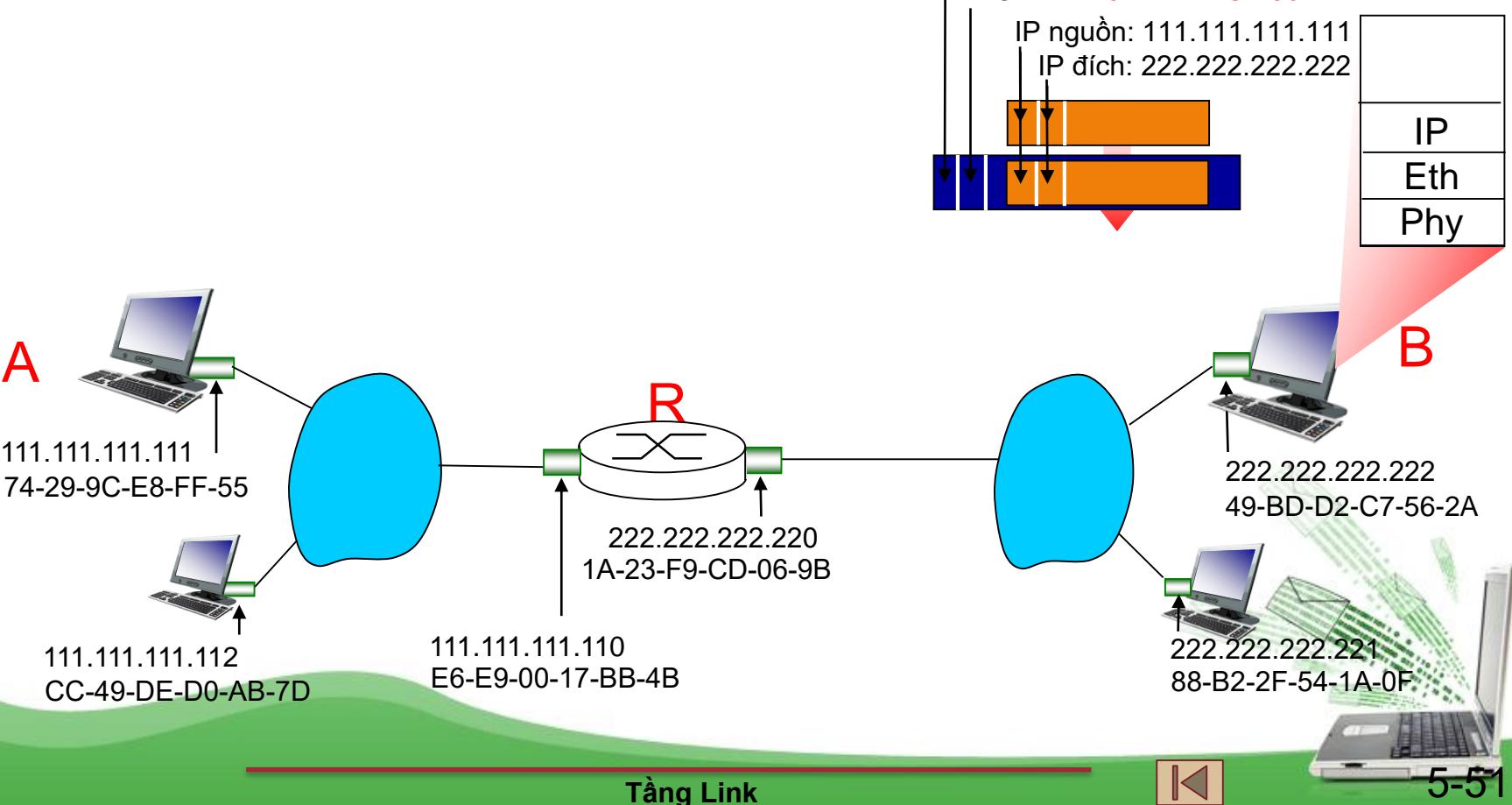
Các giao thức đa truy cập

Mạng LAN

# Addressing: định tuyến tới mạng LAN khác

- ❖ R chuyển tiếp datagram với IP nguồn A, đích B
- ❖ R tạo frame tầng Liên kết dữ liệu với địa chỉ MAC của B là địa chỉ đích, frame chứa IP datagram từ A-tới-B

Giới thiệu và các dịch vụ  
Phát hiện lỗi và sửa lỗi  
Các giao thức đa truy cập  
Mạng LAN



# Tầng Liên kết dữ liệu và mạng LAN

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANS

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN



# Ethernet

Giới thiệu và các dịch vụ

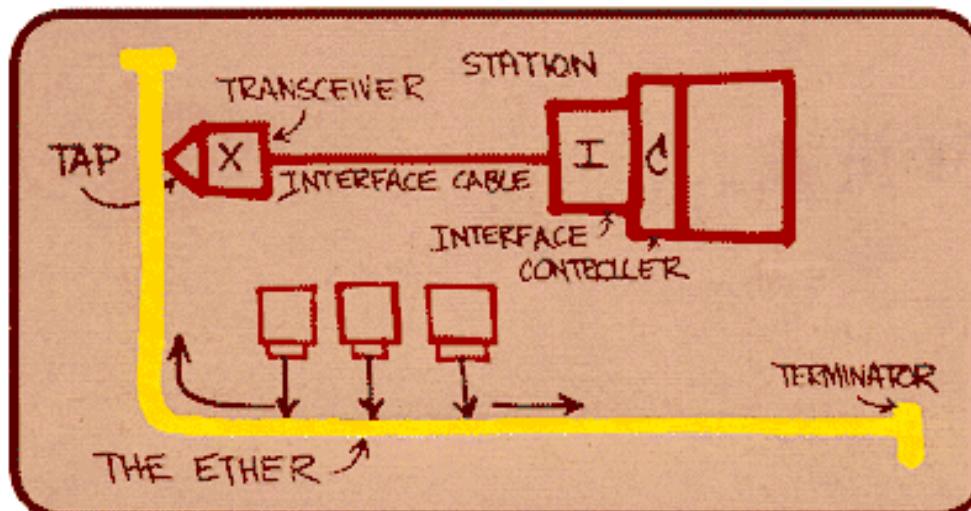
Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

Công nghệ mạng LAN hữu tuyến “chiếm ưu thế”:

- \$20 cho NIC
- Công nghệ mạng LAN đầu tiên được sử dụng rộng rãi
- Đơn giản hơn, rẻ hơn mạng LAN token và ATM
- Tăng tốc độ trung bình từ: 10 Mbps – 10 Gbps



Phác thảo Ethernet của Metcalfe



# Ethernet: cấu trúc vật lý

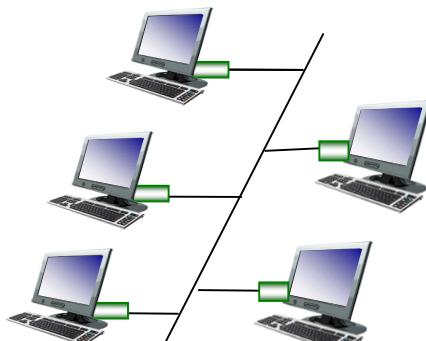
Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

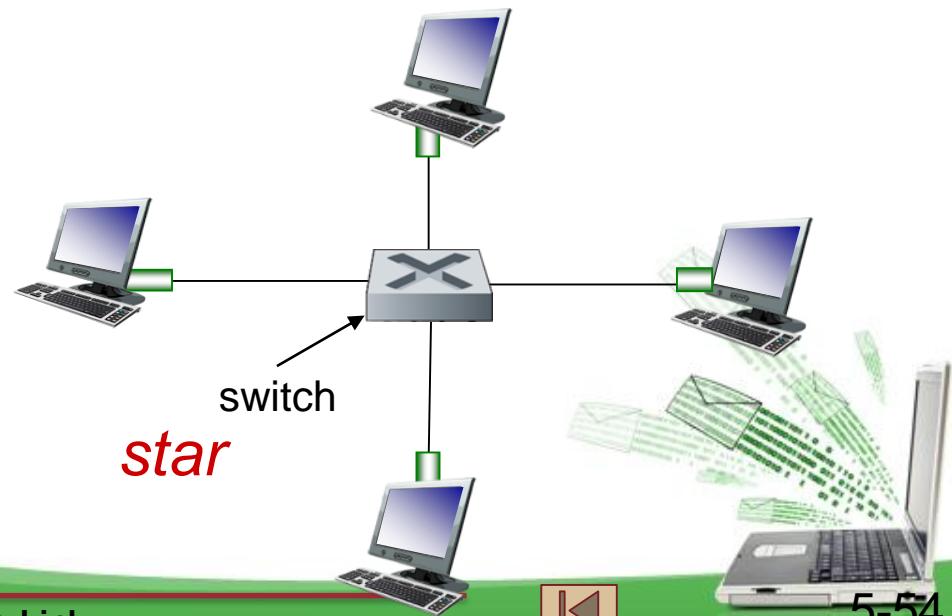
Các giao thức đa truy cập

Mạng LAN

- **Bus:** phổ biến trong giữa thập niên 90
  - Tất cả các node trong cùng vùng xung đột (collision domain) (có thể đụng độ lẫn nhau)
- **Star:** chiếm ưu thế ngày nay
  - **switch** hoạt động ở trung tâm
  - Mỗi chặng kết nối Ethernet hoạt động riêng biệt (các node không đụng độ lẫn nhau)



**bus:** cáp đồng trục



# Cấu trúc frame Ethernet

Adapter bên gửi sẽ đóng gói IP datagram (hoặc gói thuộc giao thức khác của tầng mạng) trong **Ethernet frame**



## *preamble:*

- 7 byte với mẫu 10101010 được nối tiếp bởi 1 byte với mẫu 10101011
- được sử dụng để đồng bộ tốc độ đồng hồ của bên gửi và nhận

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN



# Cấu trúc frame Ethernet (tt)

- **Addresses:** 6 byte địa chỉ MAC nguồn, đích
  - Nếu adapter nhận frame với địa chỉ đích đúng là của nó, hoặc với địa chỉ broadcast (như là ARP packet), thì nó sẽ chuyển dữ liệu trong frame tới giao thức tầng Mạng
  - Ngược lại, adapter sẽ hủy frame
- **Type:** chỉ ra giao thức tầng cao hơn (thường là IP nhưng cũng có thể là những loại khác như là Novell IPX, AppleTalk)
- **CRC:** cyclic redundancy check tại bên nhận
  - Lỗi được phát hiện: frame bị bỏ



# Ethernet: không tin cậy, không kết nối

- *Connectionless (không kết nối)*: không bắt tay giữa các NIC gửi và nhận
- *Unreliable (không tin cậy)*: NIC nhận sẽ không gửi thông báo nhận thành công (acks) hoặc không thành công (nacks) đến các NIC gửi
  - Dữ liệu trong các frame bị bỏ sẽ được khôi phục lại nếu bên gửi dùng dịch vụ tin cậy của tầng cao hơn (như là TCP) còn không thì dữ liệu đã bị bỏ sẽ mất luôn
- Giao thức MAC của Ethernet: unslotted *CSMA/CD* với *binary backoff*

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

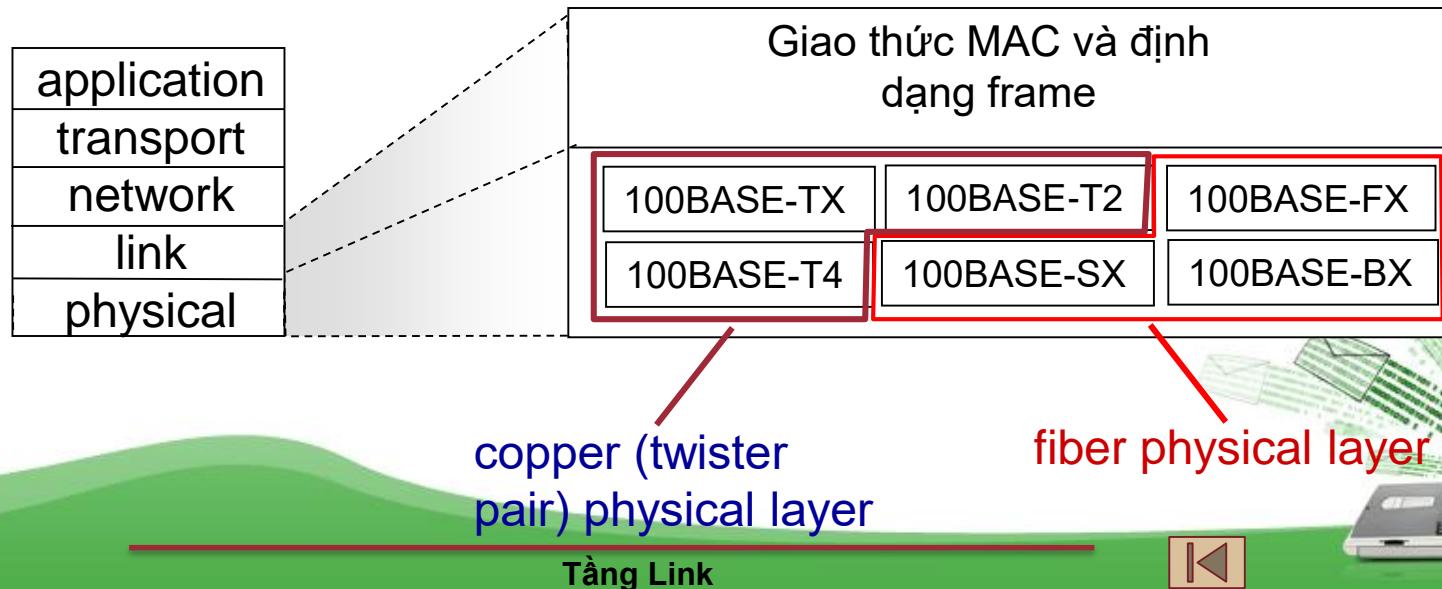
Mạng LAN



# Chuẩn Ethernet 802.3: Tầng Liên kết dữ liệu & Tầng Vật lý

[Giới thiệu và các dịch vụ](#)[Phát hiện lỗi và sửa lỗi](#)[Các giao thức đa truy cập](#)[Mạng LAN](#)

- **Nhiều** chuẩn Ethernet khác nhau
  - Cùng giao thức MAC và định dạng frame
  - Tốc độ khác nhau: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - Môi trường truyền tầng Vật lý khác nhau: fiber, cable



# Tầng Liên kết dữ liệu và mạng LAN

5.1 Giới thiệu và các  
dịch vụ

5.2 phát hiện lỗi và sửa  
lỗi

5.3 các giao thức đa  
truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN



# Ethernet switch

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

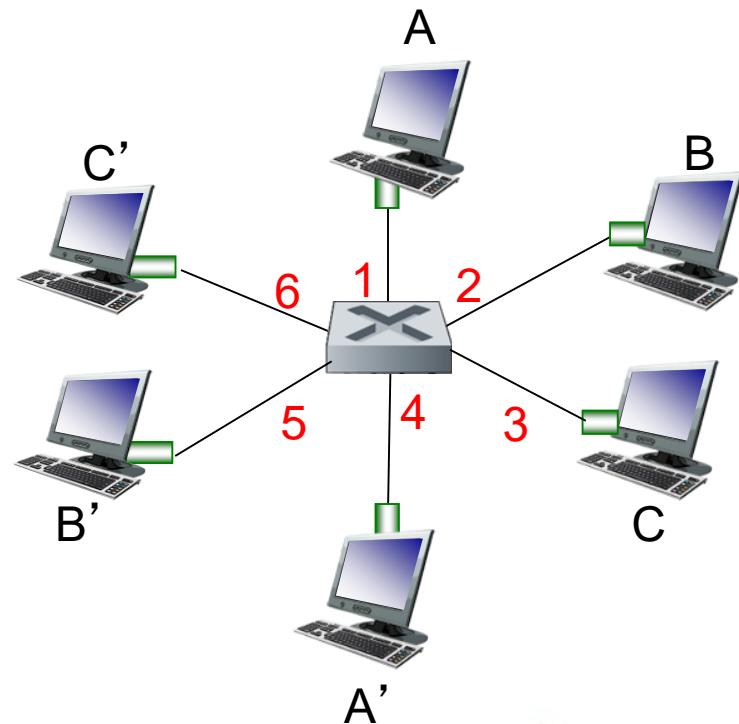
- Thiết bị tầng Liên kết dữ liệu: đóng vai trò tích cực
  - Lưu (store) và chuyển tiếp (forward ) các frame Ethernet
  - Xem xét địa chỉ MAC của frame đến, **chọn lựa** chuyển tiếp frame tới 1 hay nhiều đường link đi ra khi frame được chuyển tiếp vào từng chặng mạng, dùng CSMA/CD để truy nhập chặng mạng
- Transparent (trong suốt)
  - Các host không phát hiện được sự hiện diện của các switch
- plug-and-play, tự học
  - Các switch không cần được cấu hình



# Switch: nhiều phiên truyền đồng thời

- Các host kết nối trực tiếp tới switch
- Switch lưu tạm (buffer) các packet
- Giao thức Ethernet được sử dụng trên *mỗi* đường kết nối vào, nhưng không có đụng độ; full duplex
  - Mỗi đường kết nối là 1 miền đụng độ (collision domain) của riêng nó
- **switching:** A-tới-A' và B-tới-B' có thể truyền đồng thời mà không có đụng độ xảy ra

Giới thiệu và các dịch vụ  
Phát hiện lỗi và sửa lỗi  
Các giao thức đa truy cập  
Mạng LAN



switch với 6 interface  
(1,2,3,4,5,6)

# Bảng switch forwarding

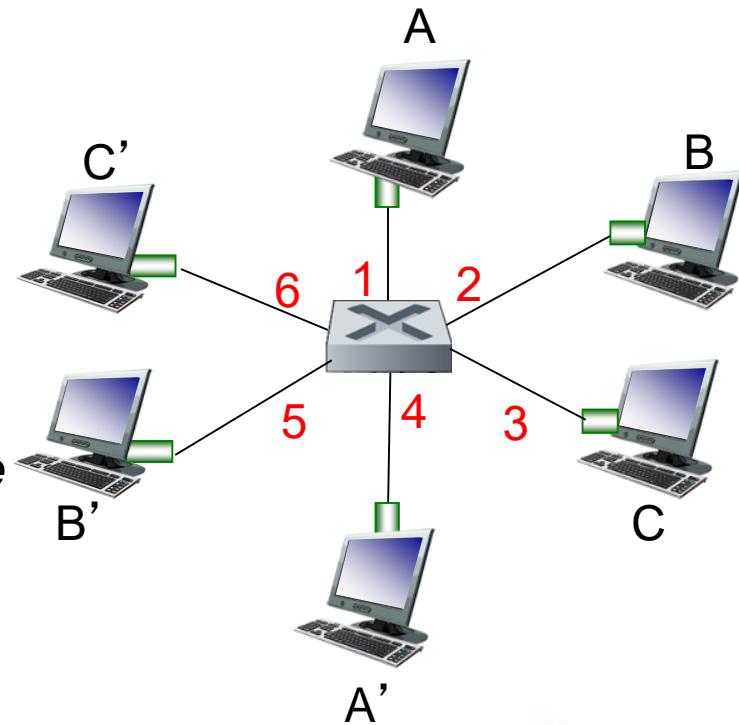
Hỏi: làm thế nào để switch biết tới A' thì sẽ thông qua interface 4 và tới B' thì thông interface 5?

- ❖ Đáp: mỗi switch có **một bảng switch**, mỗi dòng gồm:
  - (địa chỉ MAC của host, interface để tới được host đó, time stamp)
  - **Giống như bảng định tuyến!**

Q: những dòng được tạo và được duy trì như thế nào trong bảng switch?

- Có **giống như giao thức định tuyến hay không?**

Giới thiệu và các dịch vụ  
Phát hiện lỗi và sửa lỗi  
Các giao thức đa truy cập  
Mạng LAN



switch với 6 interface  
(1,2,3,4,5,6)



# Switch: tự học

- Switch **học** về vị trí các host có thể truyền tới được thông qua các interface kết nối với các host đó
  - Khi switch nhận được frame, switch “học” vị trí của bên gửi: cổng vào - incoming LAN segment
  - Ghi lại cặp (bên gửi-vị trí) vào trong bảng switch

MAC addr	interface	TTL
A	1	60

Tầng Link

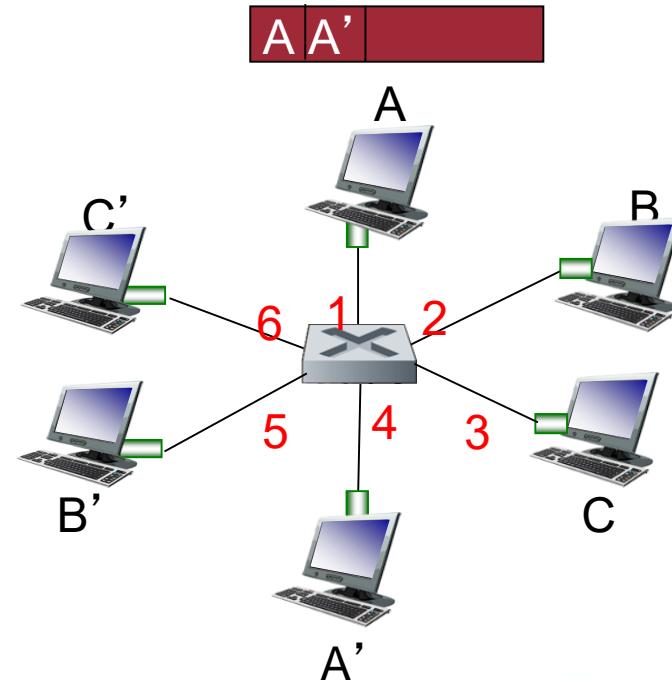
Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

Nguồn: A  
đích: A'



Bảng Switch  
(ban đầu trống)



# Switch: lọc/chuyển tiếp frame

Khi switch nhận được frame:

1. Ghi lại cổng kết nối vào, địa chỉ MAC của host gửi
2. Tạo chỉ mục bảng switch bằng địa chỉ MAC đích
3. Nếu tìm thấy thông tin đích đến  
thì {  
nếu đích đến nằm trên phân đoạn mạng gửi frame đến  
thì bỏ frame  
    ngược lại chuyển tiếp frame trên interface được chỉ định  
    bởi thông tin trong bảng switch  
}  
    ngược lại flood /\* chuyển tiếp trên tất cả interface ngoại trừ  
    interface nhận frame đó\*/}

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN



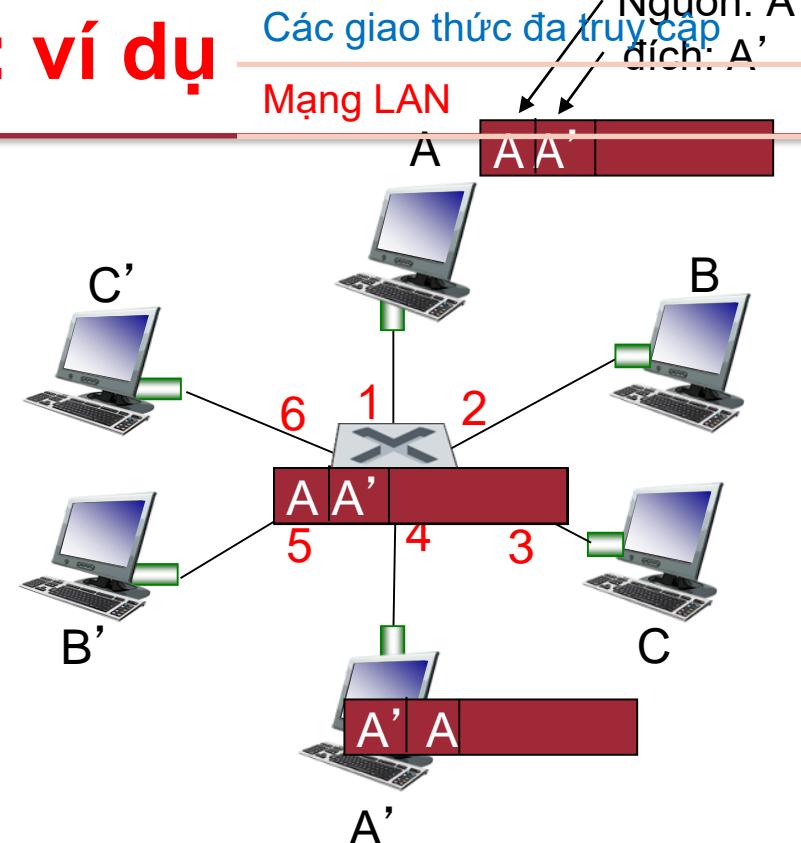
# Tự học, chuyển tiếp: ví dụ

- Frame có đích đến là A', vị trí của A' không biết:

*flood*

- ❖ Đích A có vị trí đã được biết trước:

*gửi chọn lọc chỉ trên 1 đường kết nối duy nhất*



MAC addr	interface	TTL
A	1	60
A'	4	60

Bảng switch  
(ban đầu trống)

# Kết nối các switch với nhau(Interconnecting switches)

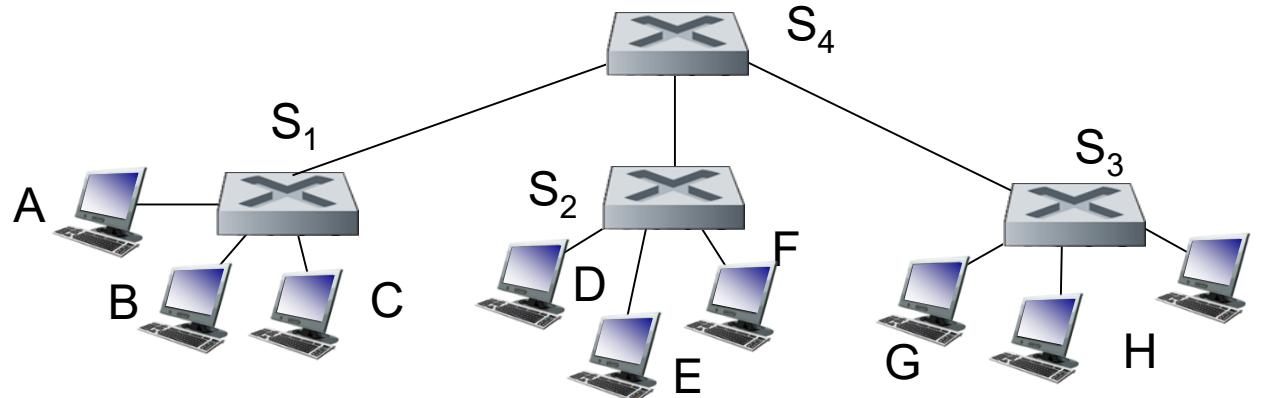
Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

- Các switch có thể được kết nối với nhau



**Hỏi:** gửi từ A tới G – làm cách nào  $S_1$  biết để chuyển tiếp frame tới F thông qua  $S_4$  và  $S_3$ ?

❖ **Trả:** tự học! (làm việc giống y chang như trong trường hợp chỉ có 1 switch!)



# Ví dụ nhiều switch tự học

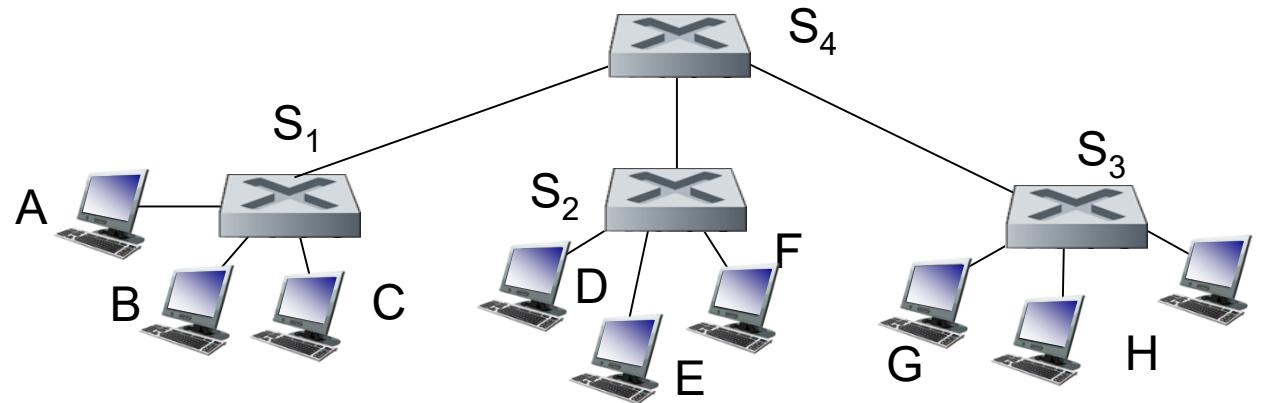
Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

Giả sử C gửi frame tới I, I trả lời cho C



- ❖ Hỏi: trình bày các bảng của các switch và cách các gói tin được chuyển đi tại các switch  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$

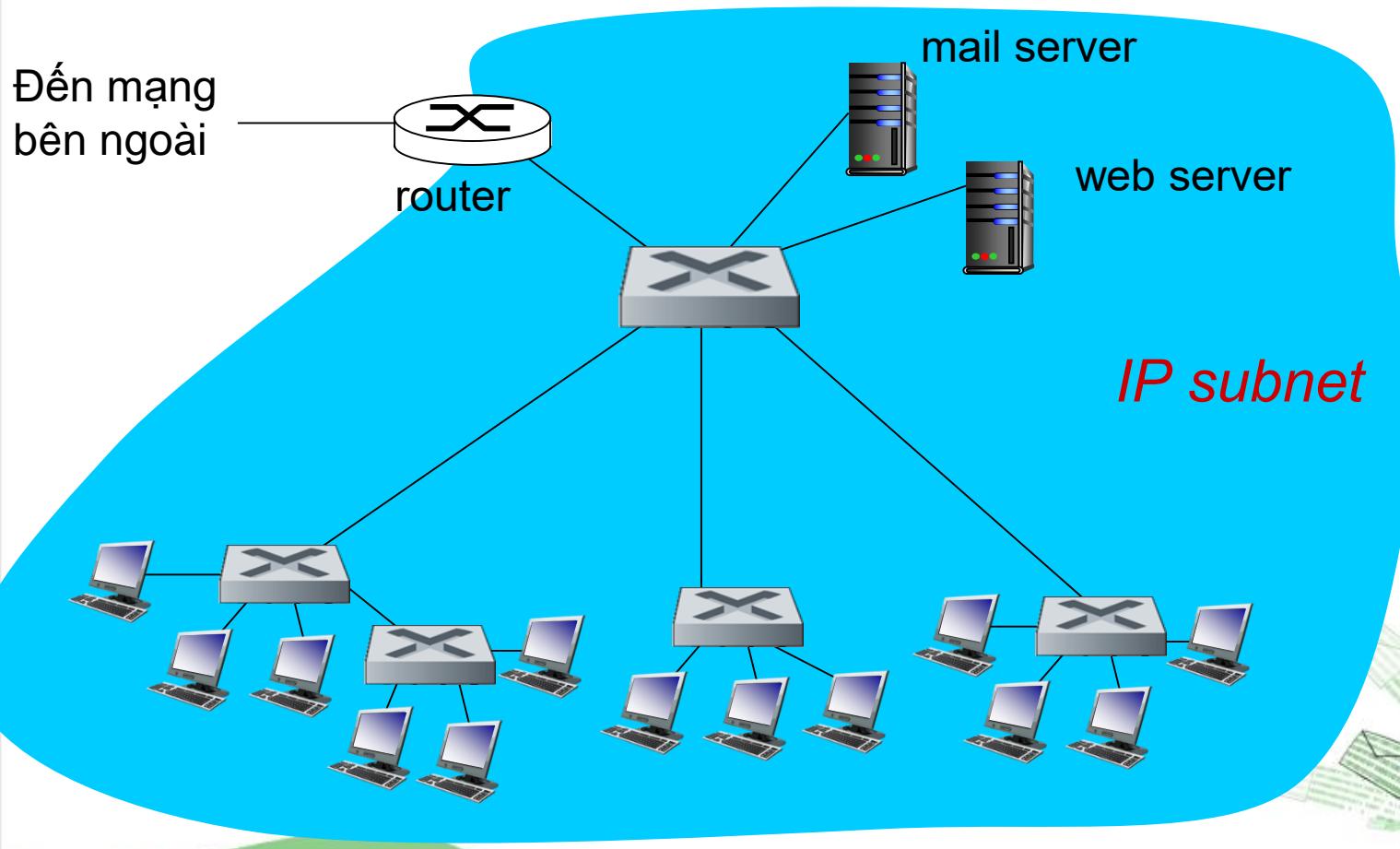
# Mạng của tổ chức

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN



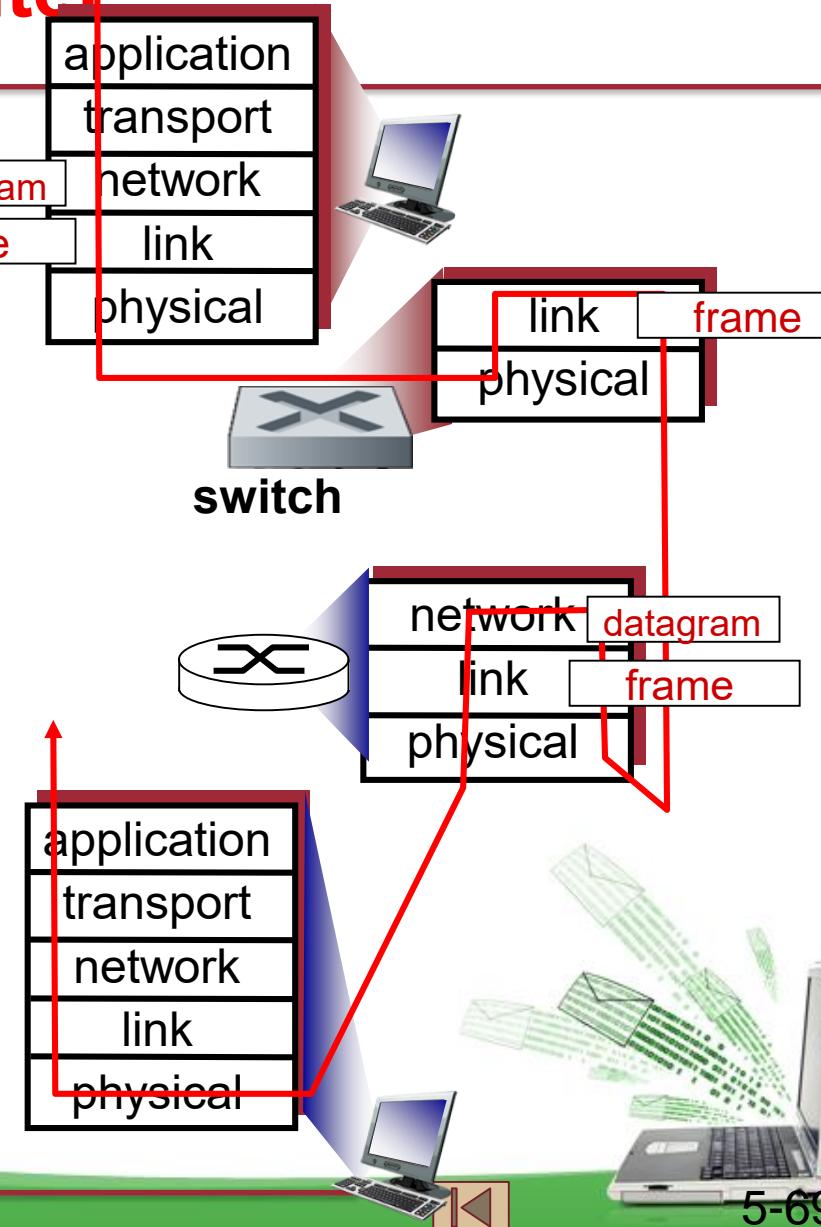
# So sánh Switch và Router

Cả 2 đều lưu và chuyển tiếp (store-and-forward):

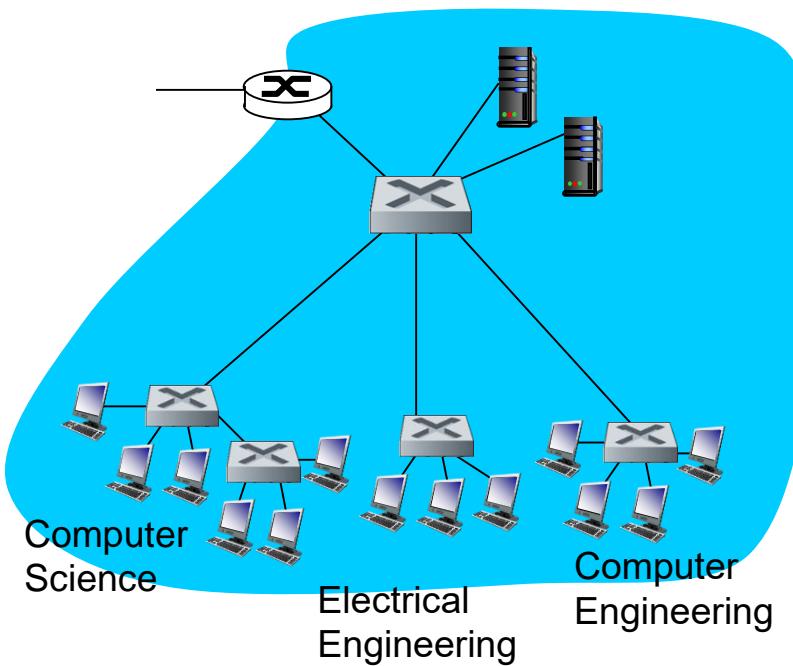
- **router:** thiết bị tầng Mạng (khảo sát header của tầng Mạng)
- **switch:** thiết bị tầng Liên kết dữ liệu (khảo sát header của tầng Liên kết dữ liệu)

Cả 2 đều có bảng forwarding:

- **router:** tính toán bảng dùng các thuật toán định tuyến, địa chỉ IP
- **switch:** học bảng forwarding dùng cơ chế flooding, tự học, địa chỉ MAC



# VLANs: trình bày



Tầng Link

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

## Xem xét:

- Người dùng bên CS di chuyển văn phòng sang EE, nhưng vẫn muốn kết nối CS switch?
- Miền broadcast đơn:
  - Tất cả lưu lượng broadcast tầng 2 (ARP, DHCP, địa chỉ MAC không biết vị trí đích đến ở đâu) phải đi qua toàn mạng LAN
  - An ninh/riêng tư, các vấn đề về hiệu suất

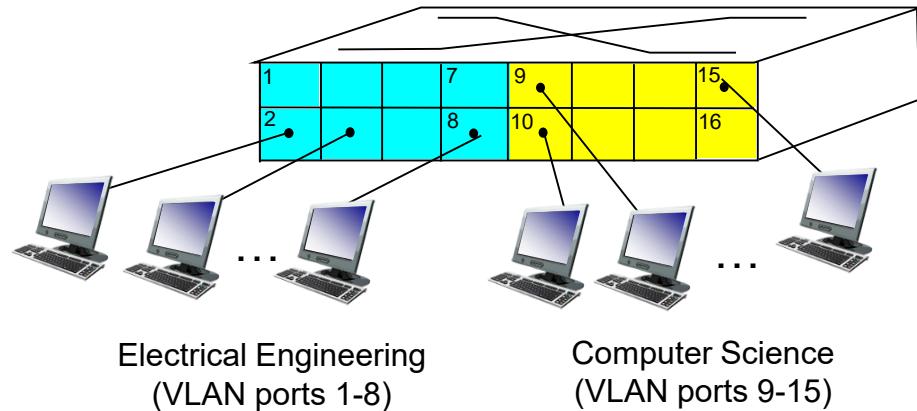


# VLANs

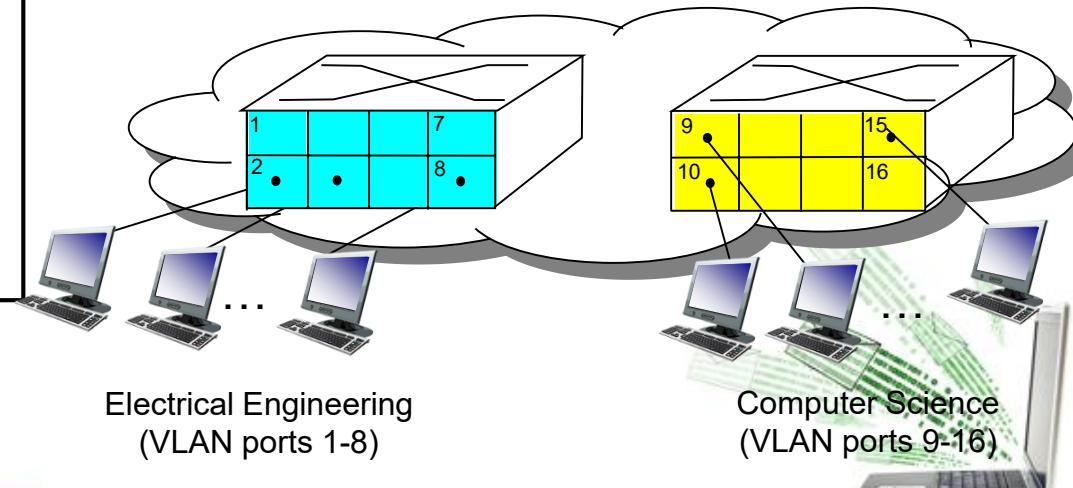
**port-based VLAN:** các port của switch được nhóm lại (bởi phần mềm quản lý switch) để trở thành một switch *vật lý* duy nhất...

## Virtual Local Area Network

Các switch hỗ trợ khả năng VLAN có thể được cấu hình để định nghĩa nhiều mạng LAN *ảo* (multiple *virtual* LANS) trên một hạ tầng vật lý của mạng LAN.



...hoạt động như là *nhiều* switch ảo



# Port-based VLAN

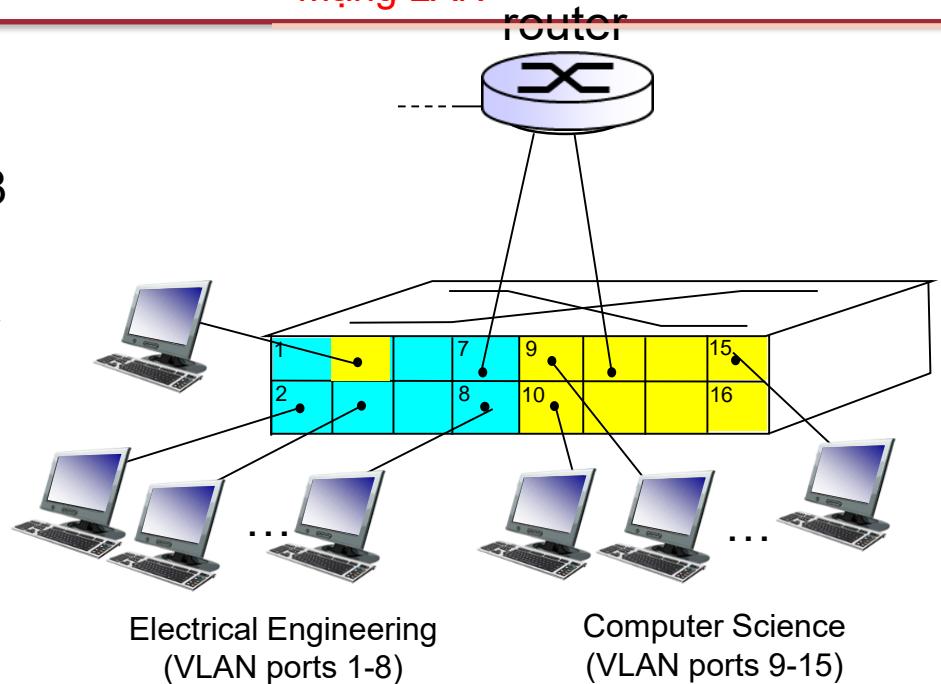
- **Traffic isolation (cô lập traffic):** các frame đến từ các port 1-8 chỉ có thể tới được các port 1-8
  - Cũng có thể định nghĩa VLAN dựa trên địa chỉ MAC của thiết bị đầu cuối, hơn là dựa trên port của switch
- ❖ **Thay đổi linh động (dynamic membership):** các port có thể được gán động giữa các VLAN
- ❖ **Chuyển tiếp giữa các VLAN:** được thực hiện thông qua định tuyến (cũng giống như nối các switch riêng biệt)
  - Trên thực tế, các nhà cung cấp bán các thiết bị switch đã kết hợp thêm một vài tính năng của router

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN



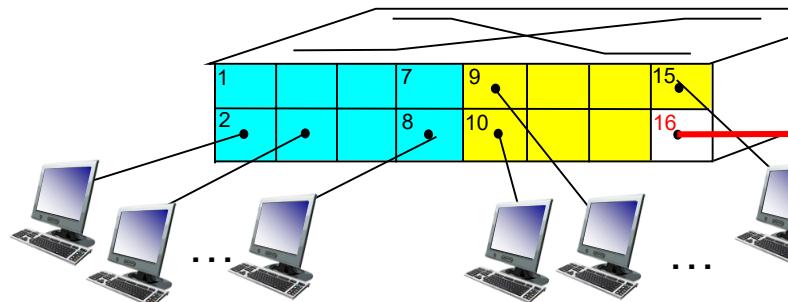
# VLANS nối nhiều switch

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

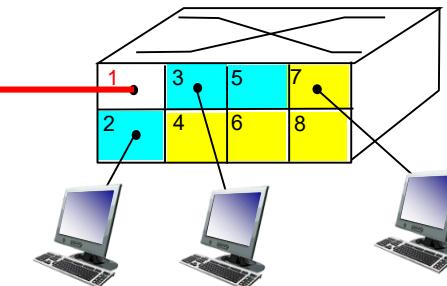
Các giao thức đa truy cập

Mạng LAN



Electrical Engineering  
(VLAN ports 1-8)

Computer Science  
(VLAN ports 9-15)



Ports 2,3,5 thuộc về EE VLAN  
Ports 4,6,7,8 thuộc về CS VLAN

- ***trunk port:*** mang các frame giữa các VLAN được định nghĩa trên nhiều switch vật lý
  - Các frame được chuyển tiếp bên trong VLAN giữa các switch không thể là các frame 802.1 (phải mang thông tin VLAN ID)
  - Giao thức 802.1q thêm/gỡ bỏ các trường header được thêm vào các frame được chuyển tiếp giữa các trunk port



# Định dạng frame VLAN 802.1Q

Giới thiệu và các dịch vụ

Phát hiện lỗi và sửa lỗi

Các giao thức đa truy cập

Mạng LAN

