

# Chương 4: Tầng mạng-mặt phẳng dữ liệu

---

## 4.1 Tổng quan tầng mạng

- Mặt phẳng dữ liệu
- Mặt phẳng điều khiển

## 4.2 Bên trong router có gì

## 4.3 IP: Internet Protocol

- Định dạng datagram
- Phân mảnh
- Đánh địa chỉ IPv4
- Dịch địa chỉ mạng
- IPv6

## 4.4 Chuyển tiếp tổng quát và SDN

- match
- action
- OpenFlow examples of match-plus-action in action

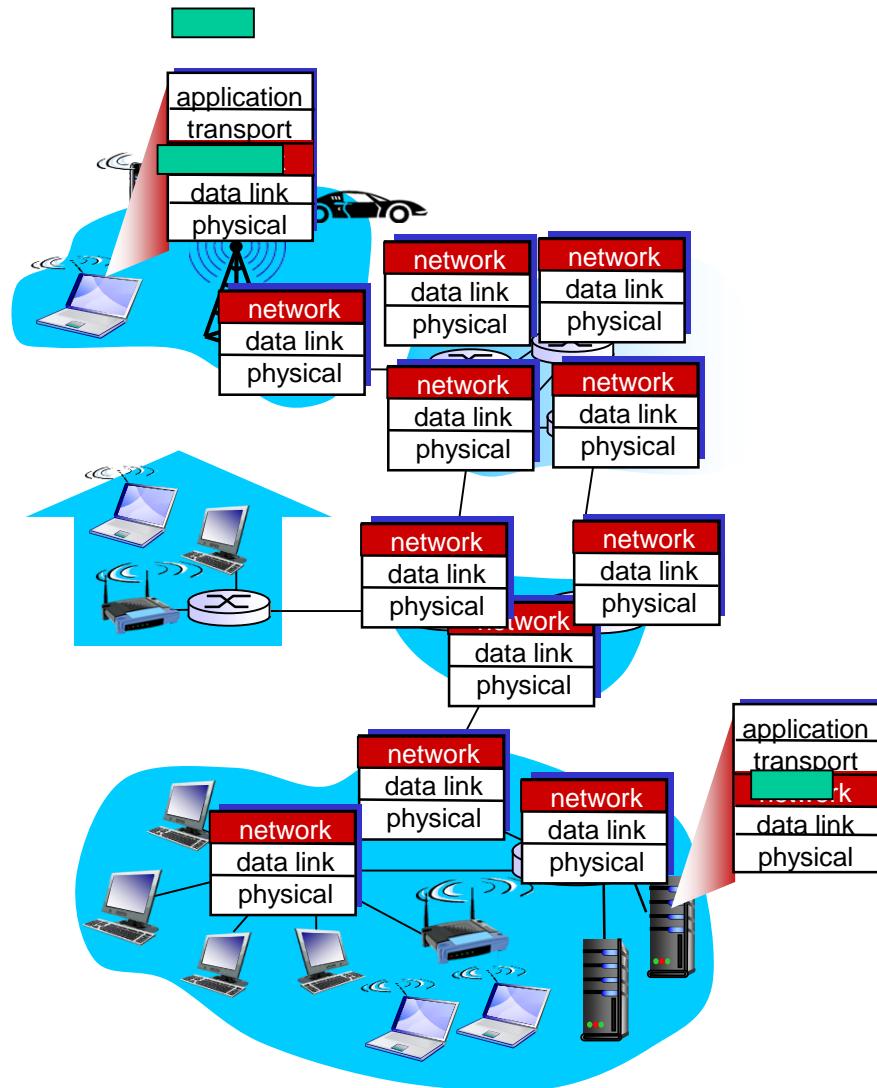
# Chương 4: Tầng mạng-mặt phẳng dữ liệu

## *Mục đích của chương:*

- Hiểu các nguyên tắc đằng sau các dịch vụ tầng mạng tập trung vào mặt phẳng dữ liệu:
  - Các mô hình dịch vụ tầng mạng
  - Chuyển tiếp so với định tuyến
  - Router làm việc như thế nào
  - Chuyển tiếp tổng quát
- Cài đặt, thực hiện trong Internet

# Network layer

- Vận chuyển segment từ nút gửi sang nút nhận
- Phía đầu gửi đóng gói segments thành datagrams
- Phía đầu nhận, chuyển các segments tới tầng giao vận
- Các giao thức tầng mạng trong mỗi host, router
- Router kiểm tra các trường tiêu đề trong tất cả các IP datagrams đi qua nó.



# Two key network-layer functions

## Các chức năng tầng mạng:

- **Chuyển tiếp:** chuyển các gói tin từ đầu vào của router tới đầu ra router tương ứng
- **Định tuyến:** xác định lộ trình của gói tin đi từ nguồn tới đích.
  - Các giải thuật định tuyến

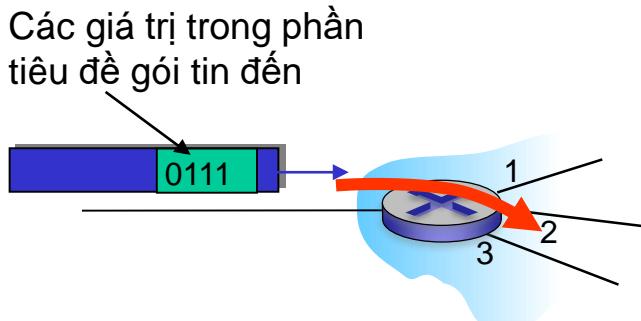
## Giống nhau: Thực hiện một chuyến đi

- **Chuyển tiếp:** quá trình vượt qua một nút giao duy nhất
- **routing:** quá trình lập kế hoạch cho chuyến đi từ nguồn đến đích

# Network layer: data plane, control plane

## Mặt phẳng dữ liệu

- Chức năng cục bộ, trên mỗi router
- Xác định gói dữ liệu đến đâu vào router được chuyển tiếp tới cổng đầu ra của router như thế nào.
- Chức năng chuyển tiếp

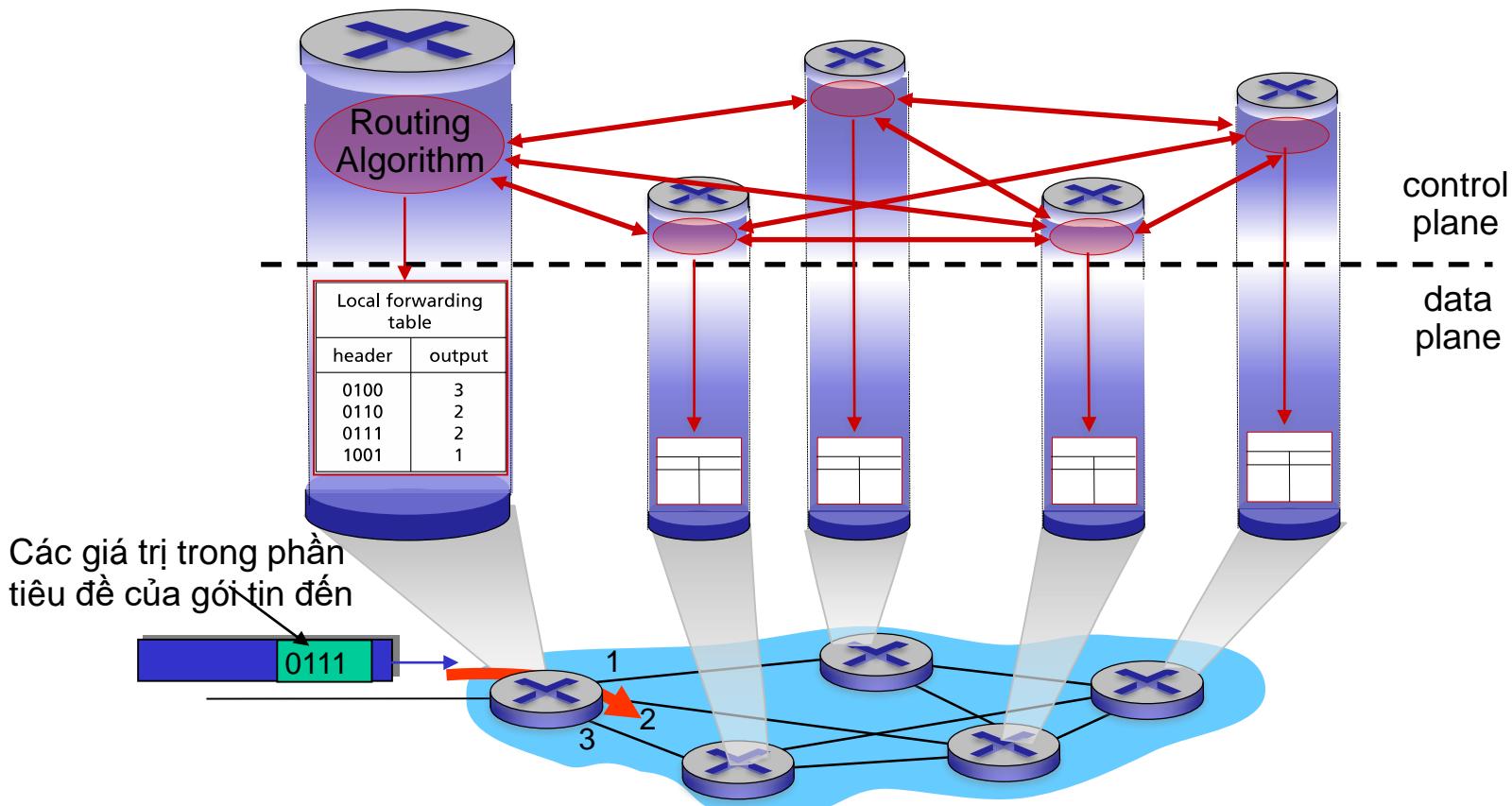


## Mặt phẳng điều khiển

- Logic toàn mạng
- Xác định gói dữ liệu được định tuyến giữa các router dọc theo đường dẫn đầu cuối từ host nguồn đến host đích.
- 2 hướng tiếp cận mặt phẳng điều khiển:
  - *Các giải thuật định tuyến truyền thống*: được thực hiện trong các routers
  - *Mạng định nghĩa mềm software-defined networking (SDN)*: thực hiện ở các server (từ xa)

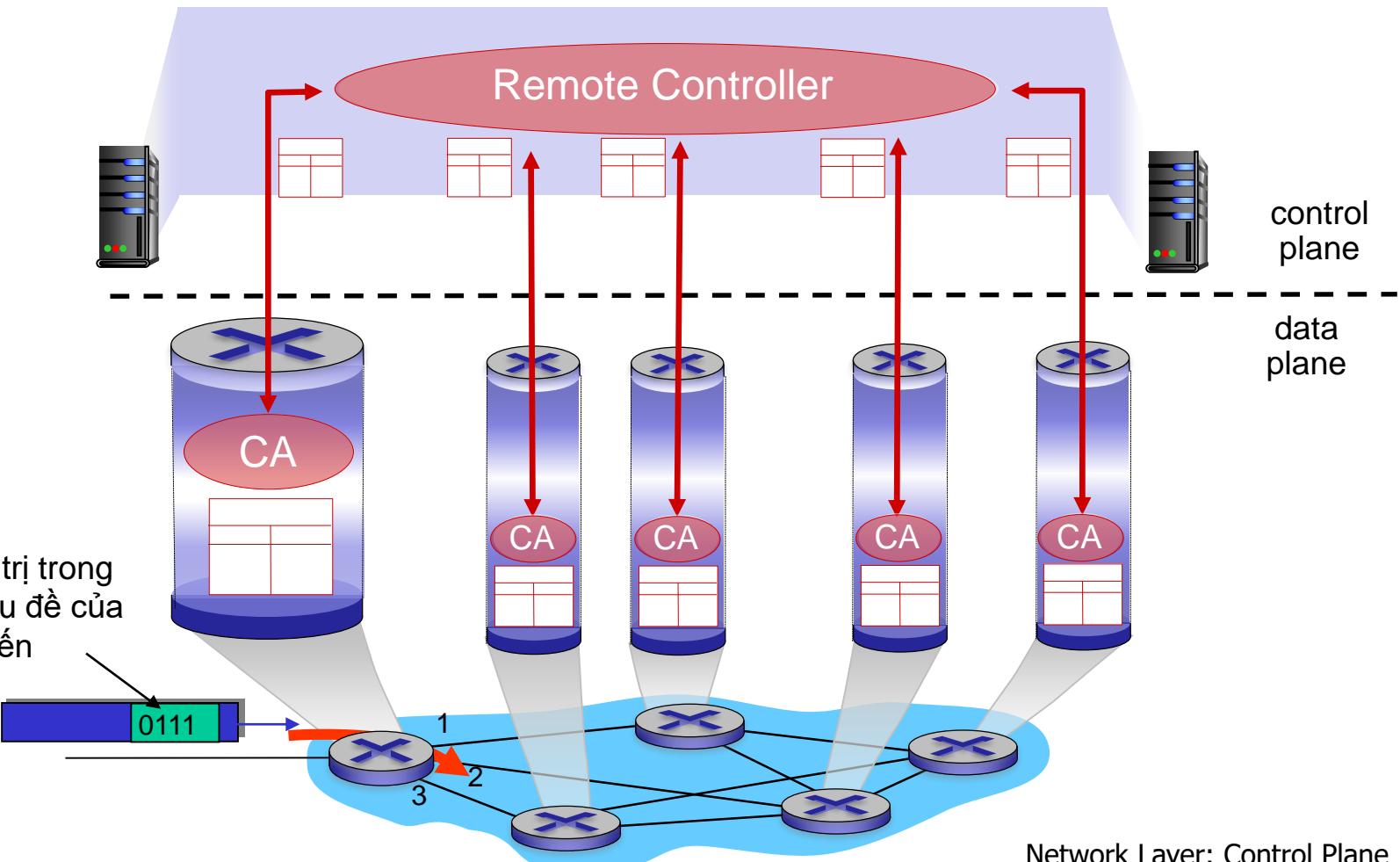
# Mặt phẳng điều khiển trên từng router

Các thành phần giải thuật định tuyến riêng trong mỗi router tương tác trong mặt phẳng điều khiển.



# Mặt phẳng điều khiển tập trung hợp lý

Một bộ điều khiển riêng biệt (thường là từ xa) tương tác với các tác nhân điều khiển cục bộ (CAs)



# Mô hình dịch vụ mạng

## *Mô hình dịch vụ mạng*

- Xác định các đặc trưng truyền đầu cuối của các gói tin từ host gửi sang host nhận.

### *Ví dụ các dịch vụ cho các datagrams riêng :*

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

### *Ví dụ các dịch vụ cho một luồng datagrams:*

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

# Các mô hình dịch vụ tầng mạng:

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

# Chapter 4: outline

## 4.1 Tổng quan tầng mạng

- Mặt phẳng dữ liệu
- Mặt phẳng điều khiển

## 4.2 Bên trong router có gì

## 4.3 IP: Internet Protocol

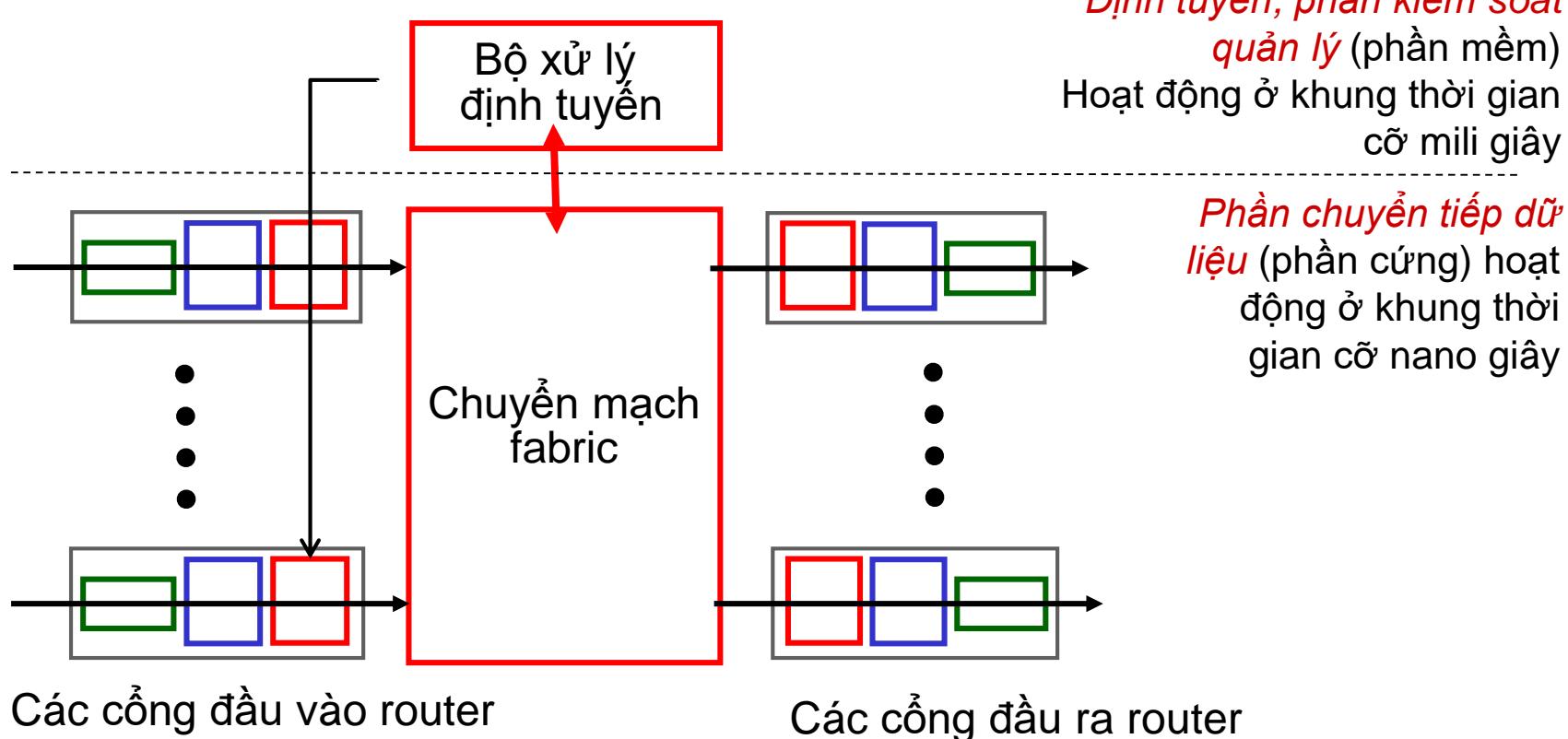
- Định dạng datagram
- Phân mảnh
- Đánh địa chỉ IPv4
- Dịch địa chỉ mạng
- IPv6

## 4.4 Chuyển tiếp tổng quát và SDN

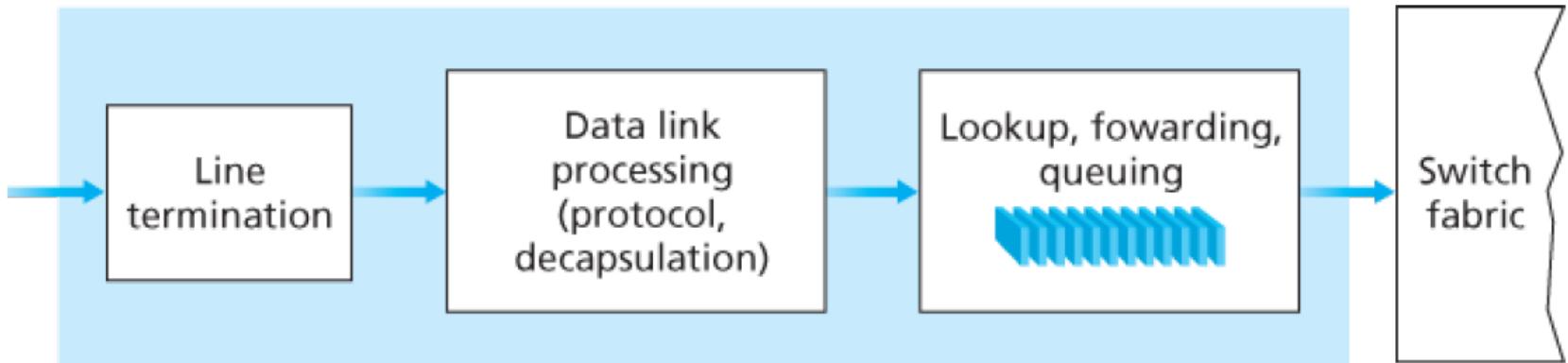
- match
- action
- OpenFlow examples of match-plus-action in action

# Tổng quan kiến trúc Router

- Xem xét ở mức cao về kiến trúc chung của router:



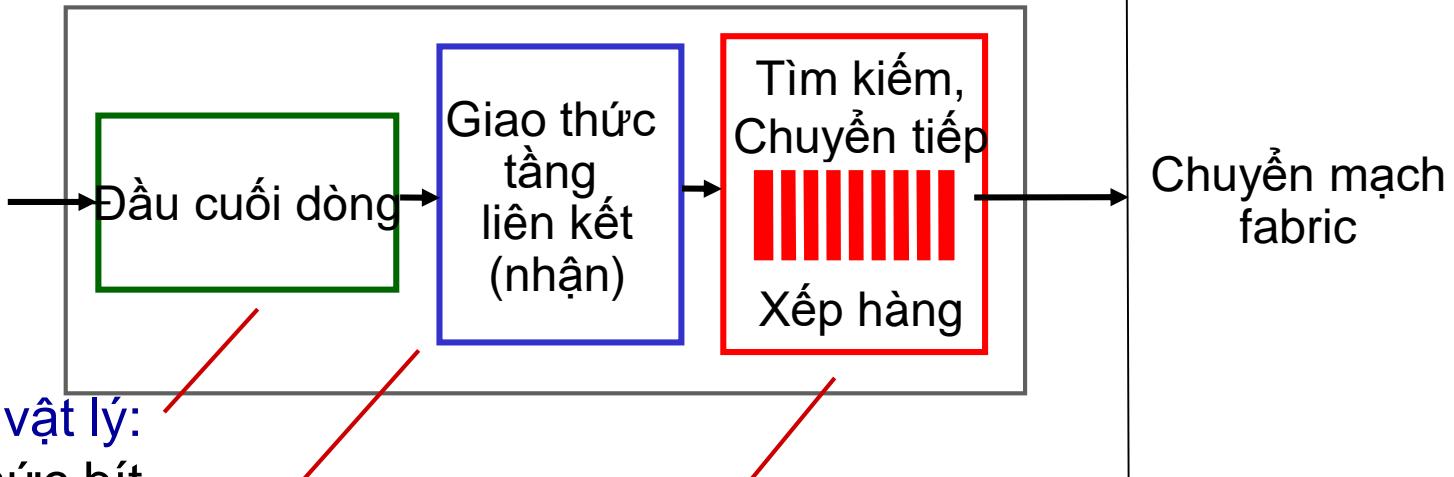
# Các chức năng cổng đầu vào



3 chức năng:

- Đầu cuối dòng (line termination): thực hiện ở tầng vật lý
- Xử lý tầng liên kết: thực hiện ở tầng liên kết
- Lookup: Tra cứu để tìm output cho gói tin đến router: tra cứu **forwarding table**. Với đích của datagram đã biết, tìm cổng ra bằng bảng chuyển tiếp trong bộ nhớ cổng vào
- Forwarding: chuyển tiếp gói tin ra switch fabric
- Queueing: nếu các datagram đến nhanh hơn tốc độ chuyển tiếp trong bộ chuyển mạch fabric

# Các chức năng cổng đầu vào



Tầng vật lý:  
Tiếp nhận mức bít

Tầng liên kết dữ liệu:  
Ví dụ, Ethernet  
Xem chương 5

Chuyển mạch không tập trung:

- Sử dụng các giá trị trường tiêu đề, tìm cổng đầu ra sử dụng bảng chuyển tiếp trong bộ nhớ cỗng đầu vào (“so khớp và hành động”)
- Chuyển tiếp dựa trên đích đến (destination-based forwarding):* chuyển tiếp dựa trên địa chỉ IP của đích đến (cách tiếp cận truyền thống)
- Chuyển tiếp tổng quát (generalized forwarding):* chuyển tiếp dựa trên tập các giá trị trường tiêu đề bất kỳ

# Chuyển tiếp dựa trên cơ sở đích

Bảng chuyển tiếp

Dải địa chỉ đích	Giao diện liên kết
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
Khác	3

Q: Nhưng điều gì xảy ra nếu như các giải địa chỉ không được phân chia hợp lý?

# Longest prefix matching

## *So khớp tiền tố dài nhất*

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

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

Ví dụ:

DA: 11001000 00010111 00010110 10100001

Giao diện nào?

DA: 11001000 00010111 00011000 10101010

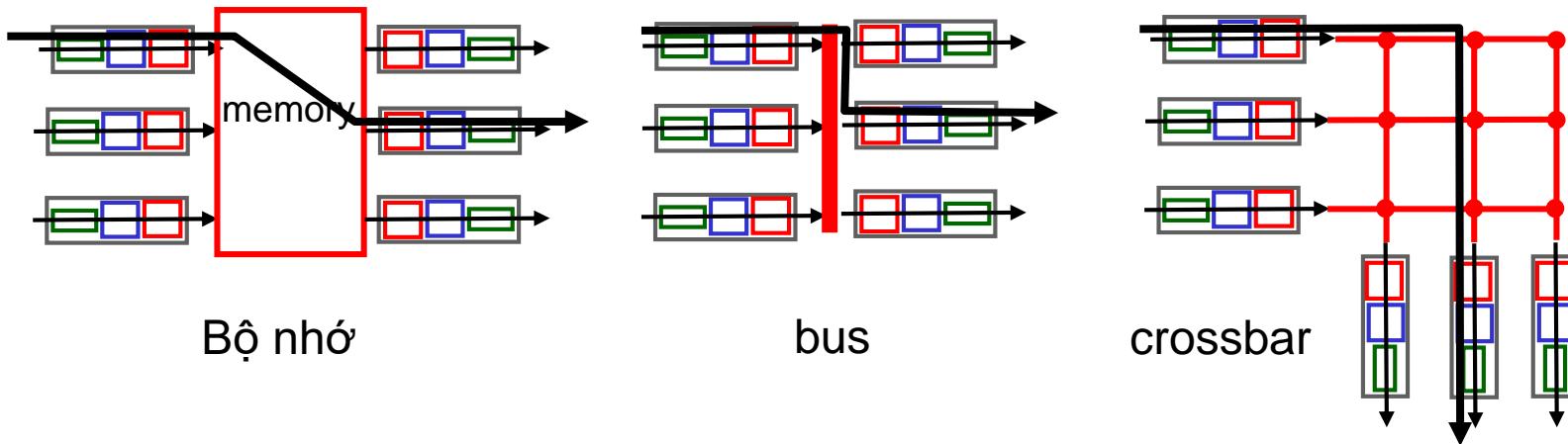
Giao diện nào?

# So khớp tiền tố dài nhất

- So khớp tiền tố dài nhất: thường được thực hiện với Ternary Content Addressable Memories (TCAMs)
  - *Nội dung có thể đánh địa chỉ*: thể hiện địa chỉ trong TCAM: truy xuất địa chỉ trong một chu kỳ đồng hồ, bất kể kích thước bảng chuyển tiếp
  - Cisco Catalyst: có thể lên đến ~1M mục bảng định tuyến trong TCAM

# Chuyển mạch (Switching fabrics)

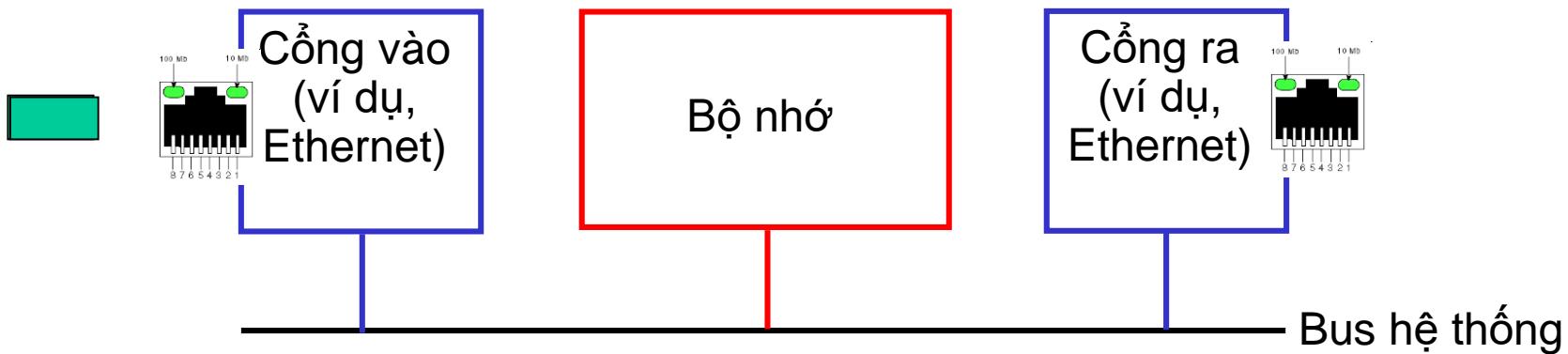
- Truyền gói tin từ bộ nhớ đệm đầu vào tới bộ nhớ đệm đầu ra tương ứng
- Tốc độ chuyển mạch: tốc độ các gói tin có thể được truyền từ các inputs tới các outputs
  - Thường được đo bằng bội số của tốc độ đường input/output
  - N inputs: tốc độ chuyển mạch gấp N lần tốc độ đường mong muốn
- 3 kiểu switching fabrics



# Chuyển mạch qua bộ nhớ

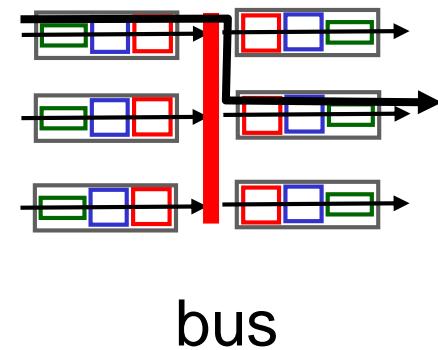
*Các router thế hệ đầu tiên:*

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



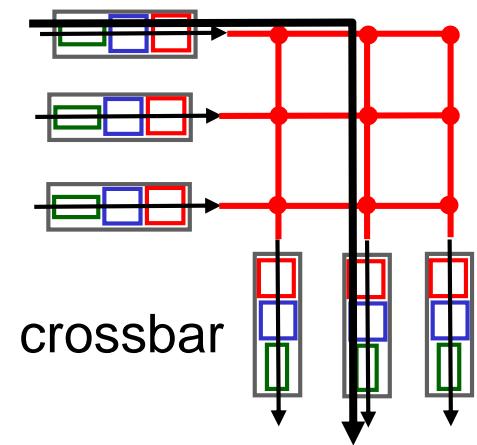
# Chuyển mạch qua một bus

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



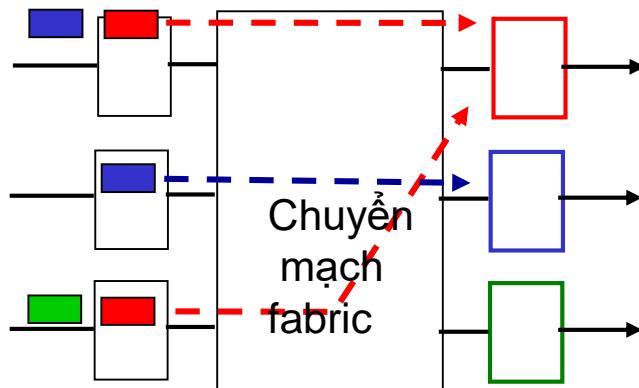
# Chuyển mạch qua mạng kết nối nội bộ

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

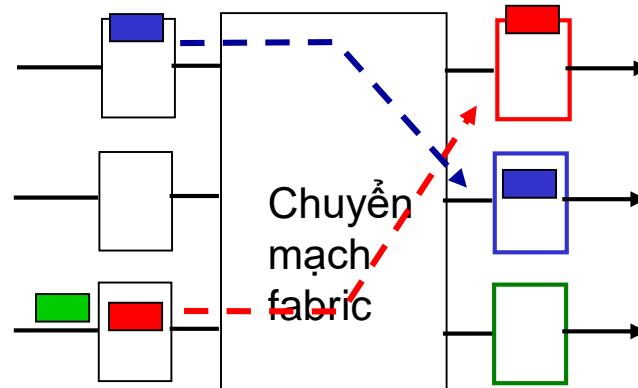


# Xếp hàng ở cổng vào

- Nếu các datagram từ các cổng vào đến nhanh hơn tốc độ chuyển tiếp trong bộ chuyển mạch fabric -> xếp hàng ở các hàng đợi vào
  - *Trễ xếp hàng và mất gói tin do bộ nhớ đệm vào bị tràn!*
- **Chặn Head-of-the-Line (HOL):** datagram xếp hàng trước hàng đợi ngăn những datagram khác trong hàng tiến lên

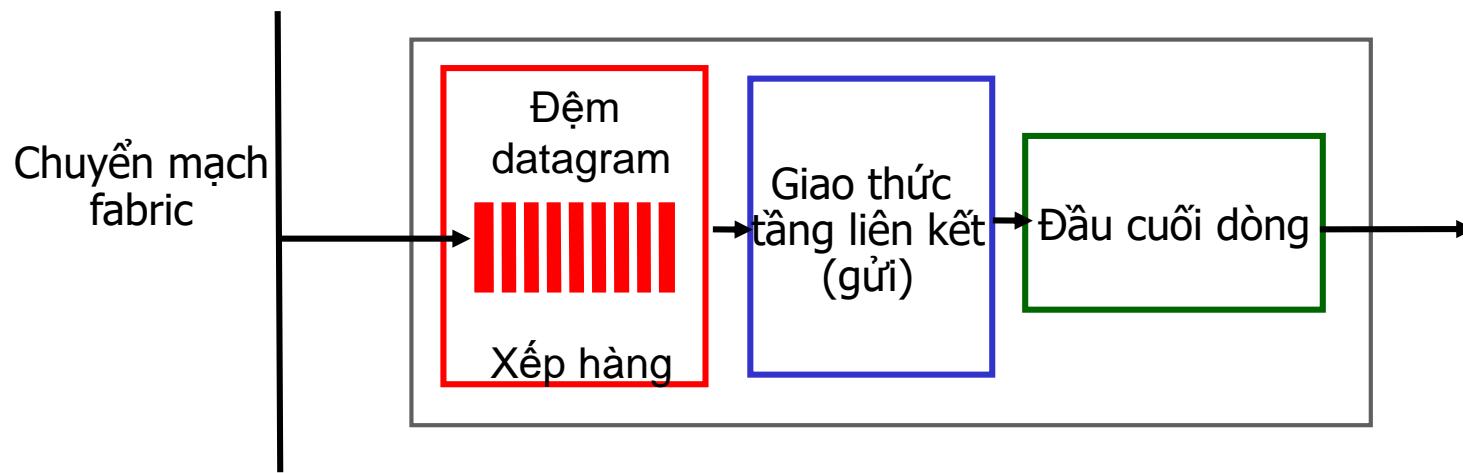


Tranh chấp cổng đầu ra:  
Chỉ có một datagram màu đỏ có  
thể truyền.  
Gói tin màu đỏ phía dưới bị chặn



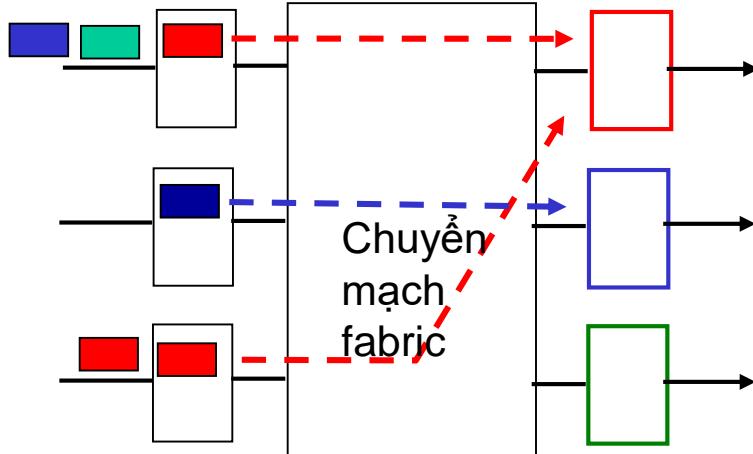
Mỗi lần truyền một  
gói: Gói màu xanh lá  
cây chịu chặn HOL

# Các cổng ra

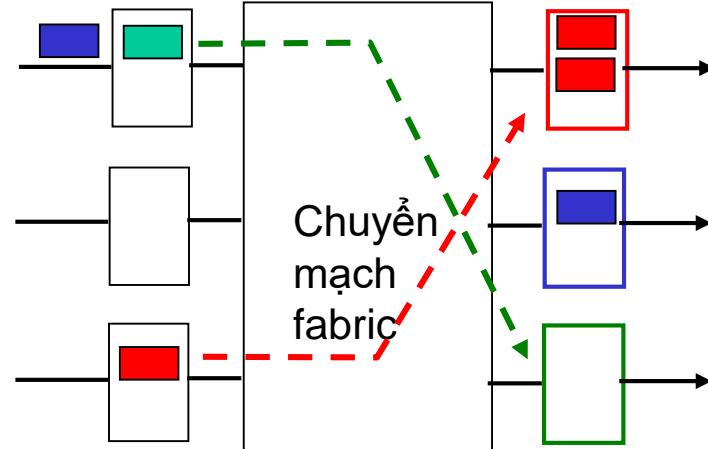


- ❖ *Việc đếm* được yêu cầu khi các datagram đến từ fabric nhanh hơn tốc độ truyền đi
- ❖ *Lịch truyền* sẽ lựa chọn các datagram trong hàng đợi để truyền

# Xếp hàng ở cổng ra



Tại thời điểm  $t$ , nhiều  
gói tin từ cổng vào đi  
đến một cổng ra



Một gói một lần

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

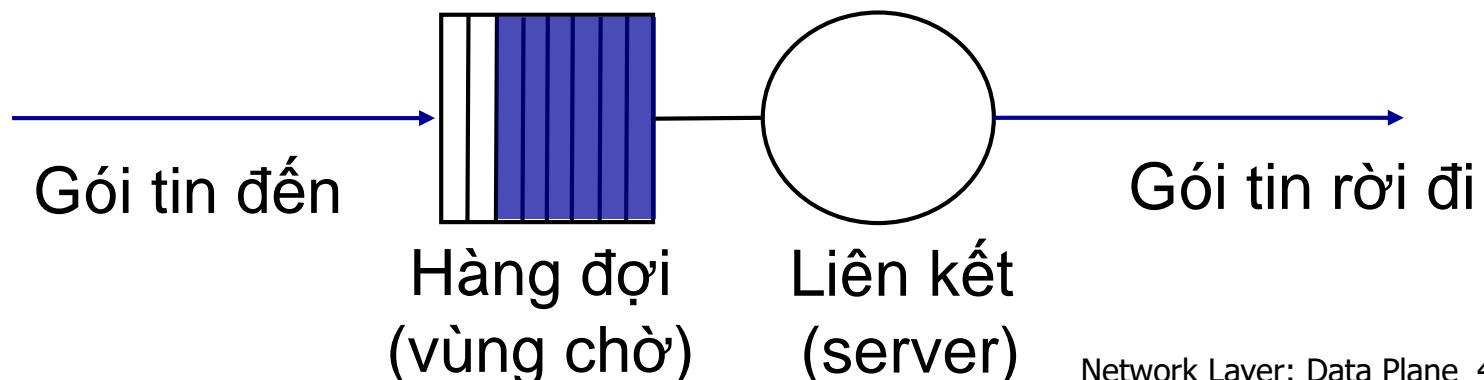
# Cần bao nhiêu cho bộ đệm?

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

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

# Các cơ chế xếp lịch

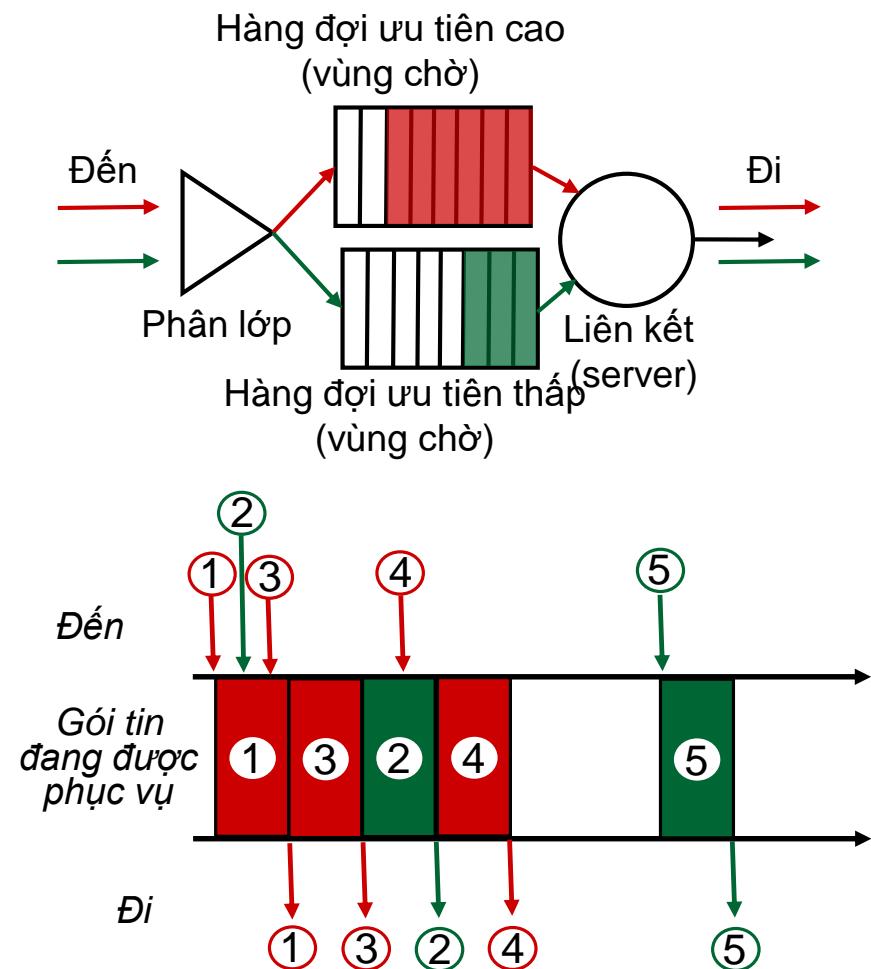
- **Xếp lịch:** lựa chọn gói tin tiếp theo gửi lên đường liên kết
- **Xếp lịch FIFO (first in first out):** gửi theo thứ tự đến hàng đợi
  - *Chính sách loại bỏ:* Nếu gói tin đến hàng đợi đầy: gói nào sẽ bị loại bỏ?
    - *Bỏ đuôi:* bỏ gói tin đang đến
    - *Ưu tiên:* bỏ dựa trên cơ sở ưu tiên
    - *Ngẫu nhiên:* bỏ ngẫu nhiên



# Các chính sách xếp lịch: ưu tiên

**Xếp lịch ưu tiên:** gửi gói tin xếp hàng ưu tiên cao nhất

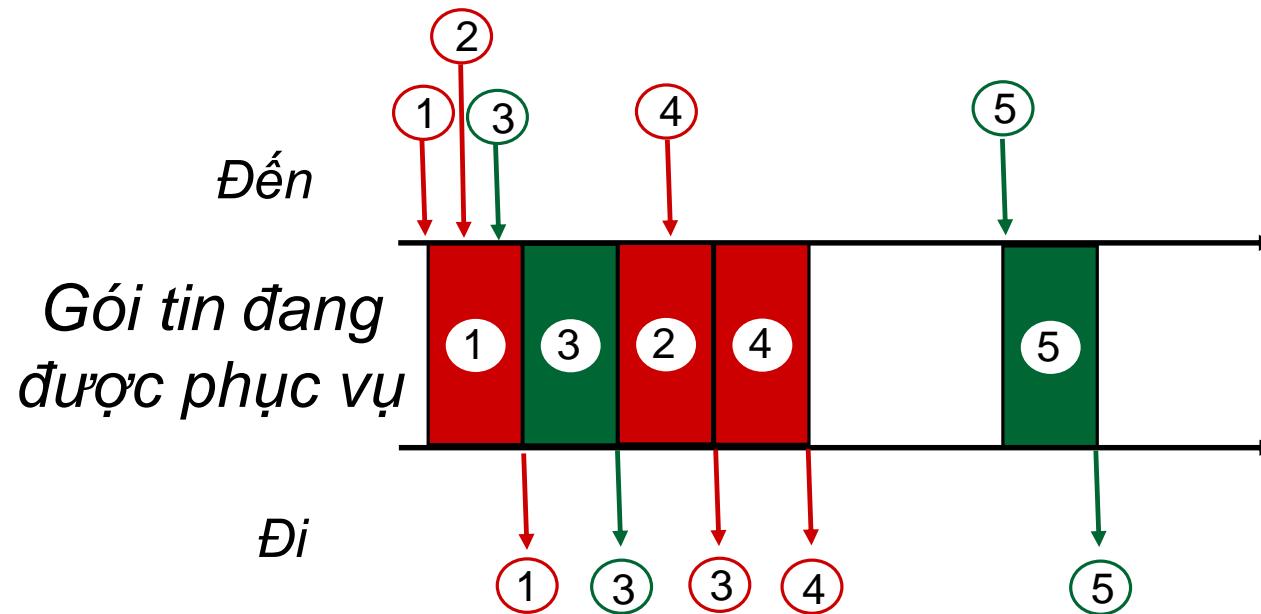
- Nhiều lớp, với các mức ưu tiên khác nhau
  - Lớp có thể phụ thuộc vào đánh dấu hoặc thông tin tiêu đề khác, ví dụ IP nguồn/đích, các số hiệu cổng...



# Các chính sách xếp lịch: vòng tròn

Xếp lịch vòng tròn - Round Robin (RR):

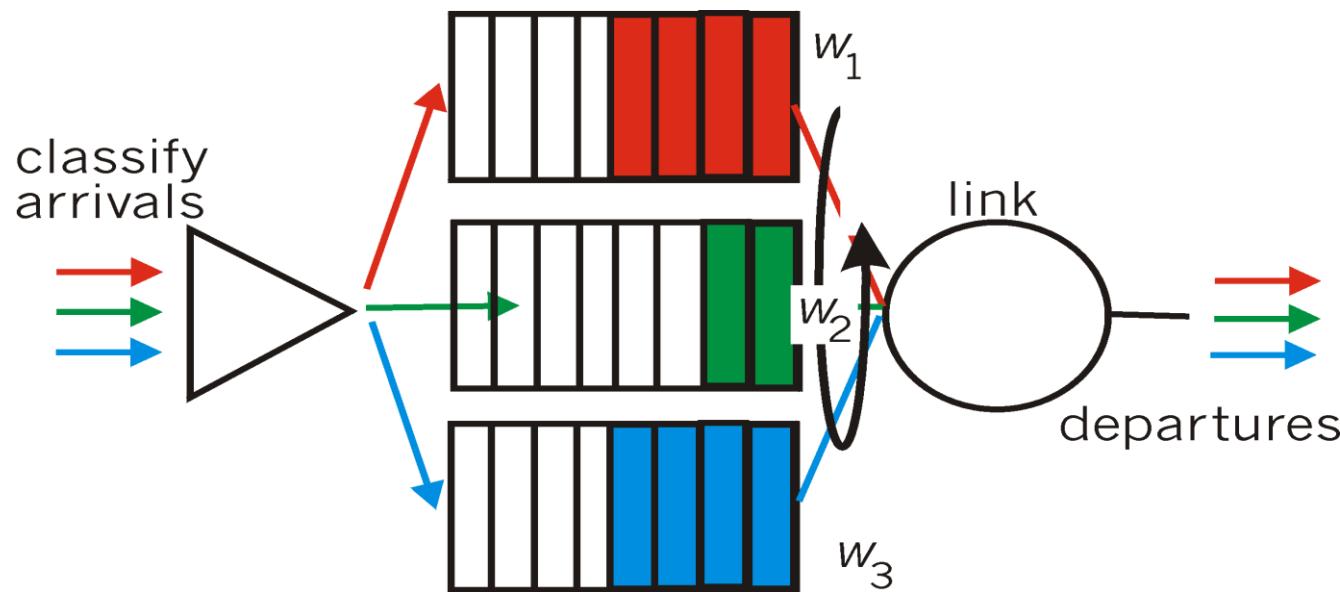
- Nhiều lớp
- Quét hàng đợi lớp theo chu kỳ, gửi một gói hoàn chỉnh từ mỗi lớp (nếu có)?



# Các chính sách xếp lịch:

## Weighted Fair Queuing (WFQ):

- Trường hợp tổng quát của Round Robin
- Mỗi lớp nhận một lượng dịch vụ có trọng số trong mỗi vòng



# Review

1. 2 chức năng cơ bản của tầng mạng?
2. Chức năng nào được thực hiện ở mặt phẳng dữ liệu? Chức năng nào được thực hiện mở mặt phẳng điều khiển?
3. Có mấy hướng tiếp cận để giải quyết vấn đề định tuyến?
4. Chuyển tiếp dựa trên cơ sở đích đến (destination-based forwarding)?
  - Forwarding table: chứa cái gì
  - Longest prefix matching: có nghĩa là gì?
  - Match and action
5. Chuyển tiếp tổng quát (generalized forwarding)
  - Flow table
  - Match and actions

# Chapter 4: outline

## 4.1 Overview of Network layer

- data plane
- control plane

## 4.2 What's inside a router

## 4.3 IP: Internet Protocol

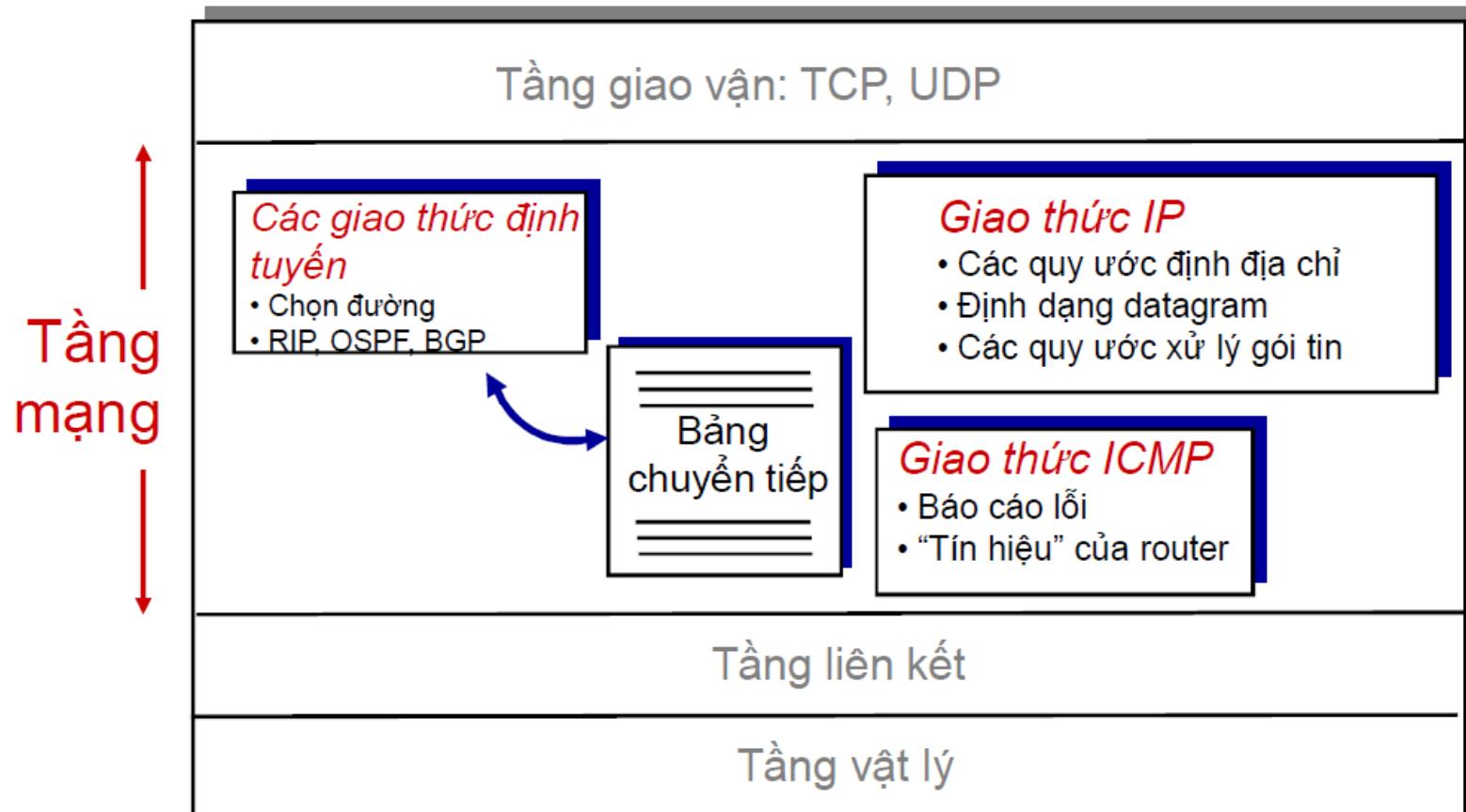
- datagram format
- fragmentation
- IPv4 addressing
- network address translation
- IPv6

## 4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

# Tầng mạng Internet

Các chức năng của tầng mạng tại host và router:



# Định dạng IP datagram

Phiên bản giao thức IP

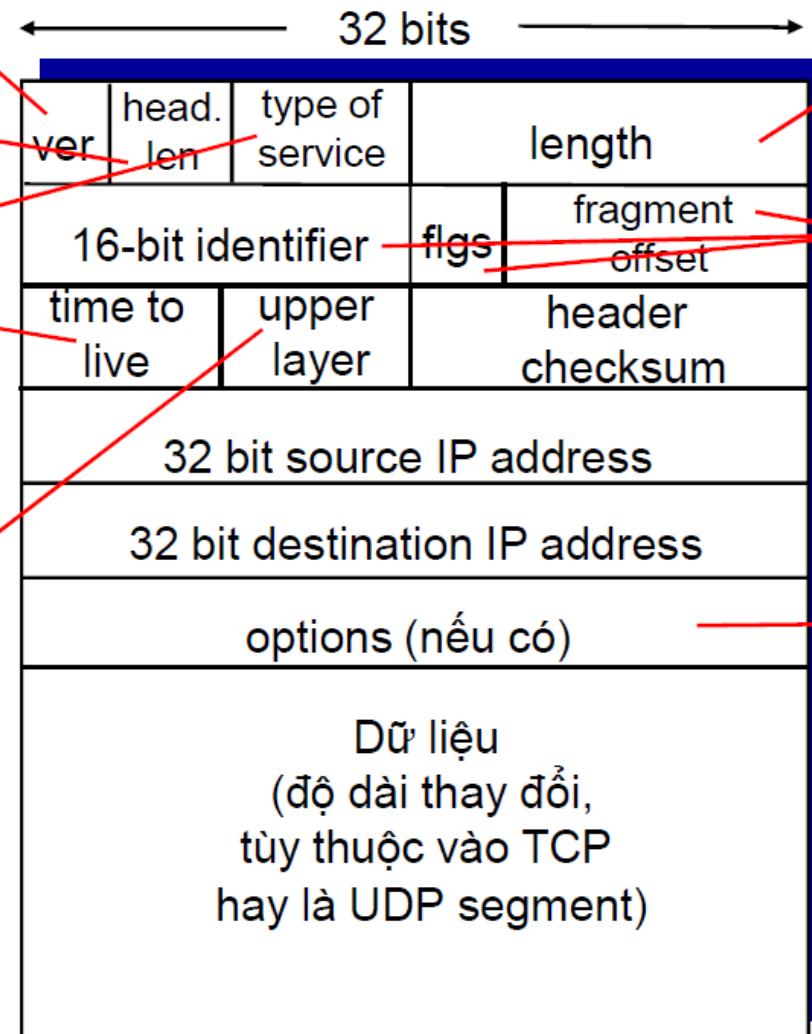
Chiều dài phần tiêu đề (byte)  
“Loại” dữ liệu

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

Giao thức tầng cao hơn thực hiện phân phối payload đi

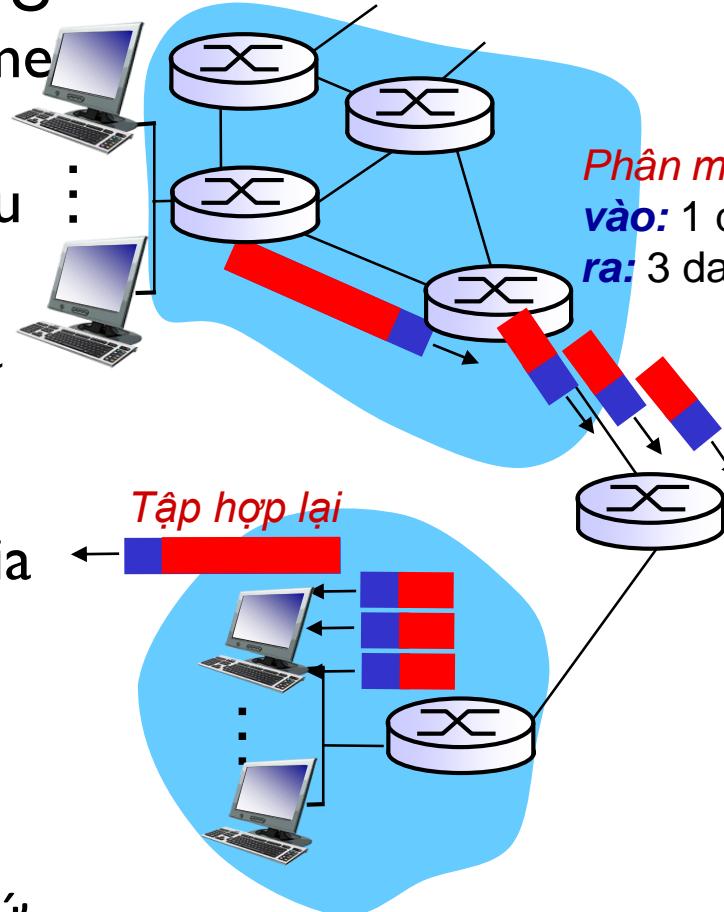
**Bao nhiêu byte cho phần tiêu đề gói tin?**

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



# Phân mảnh và tập hợp lại gói tin IP

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



# Phân mảnh và tập hợp lại gói tin IP

Ví dụ:

- ❖ Datagram 4000 byte
- ❖ MTU = 1500 bytes

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

*Một datagram lớn được chia thành  
một số datagrams nhỏ hơn*

1480 bytes trong  
trường dữ liệu

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

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

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

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

# Chapter 4: outline

## 4.1 Overview of Network layer

- data plane
- control plane

## 4.2 What's inside a router

## 4.3 IP: Internet Protocol

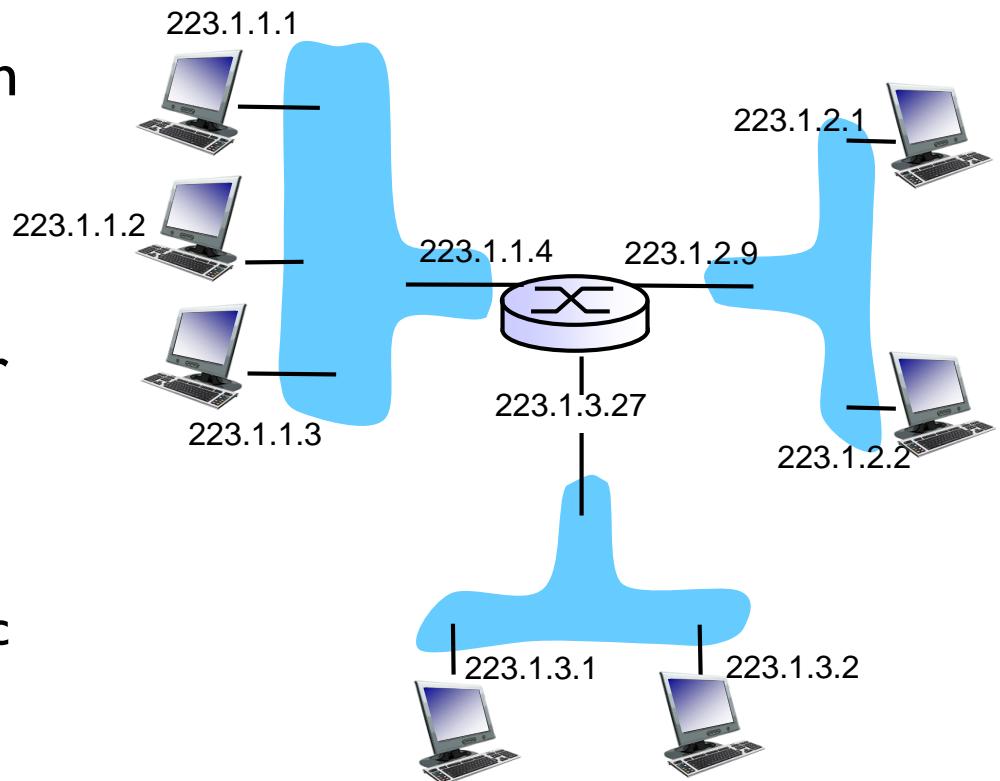
- datagram format
- fragmentation
- IPv4 addressing
- network address translation
- IPv6

## 4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

# Địa chỉ IP: giới thiệu

- **Địa chỉ IP:** số định danh 32-bit cho host, router interface
- **Giao diện (interface):** kết nối giữa host/router và liên kết vật lý
  - Router thường có nhiều giao diện
  - host thường có một hoặc 2 giao diện (ví dụ, Ethernet, 802.11 không dây)
- **Các địa chỉ IP được gán cho từng giao diện**



$223.1.1.1 = \underbrace{11011111}_{223} \underbrace{00000001}_{1} \underbrace{00000001}_{1} \underbrace{00000001}_{1}$

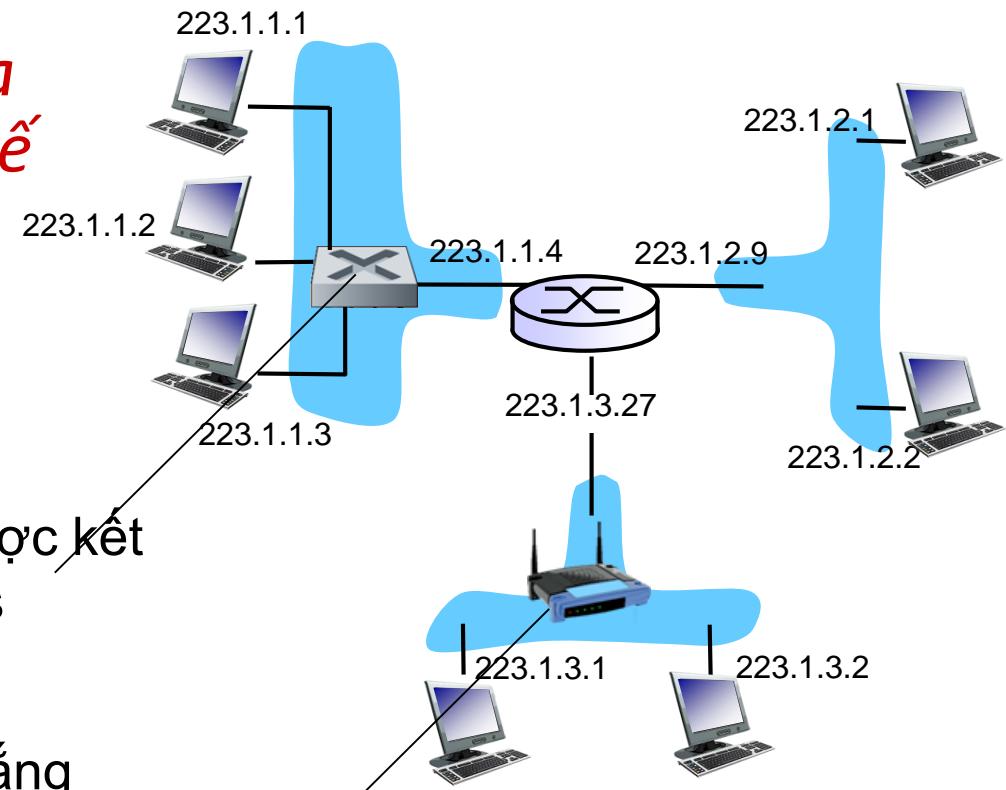
# Địa chỉ IP: giới thiệu

*Q: Các giao diện thực ra kết nối với nhau như thế nào?*

*A: Sẽ học ở chương 5,6*

*A: Các giao diện Ethernet được kết nối bởi các Ethernet switches*

*Hiện tại:* không cần phải lo lắng về cách một giao diện được kết nối với một giao diện khác (không có bộ định tuyến can thiệp)



*A: các giao diện WiFi được kết nối bởi trạm cơ sở WiFi*

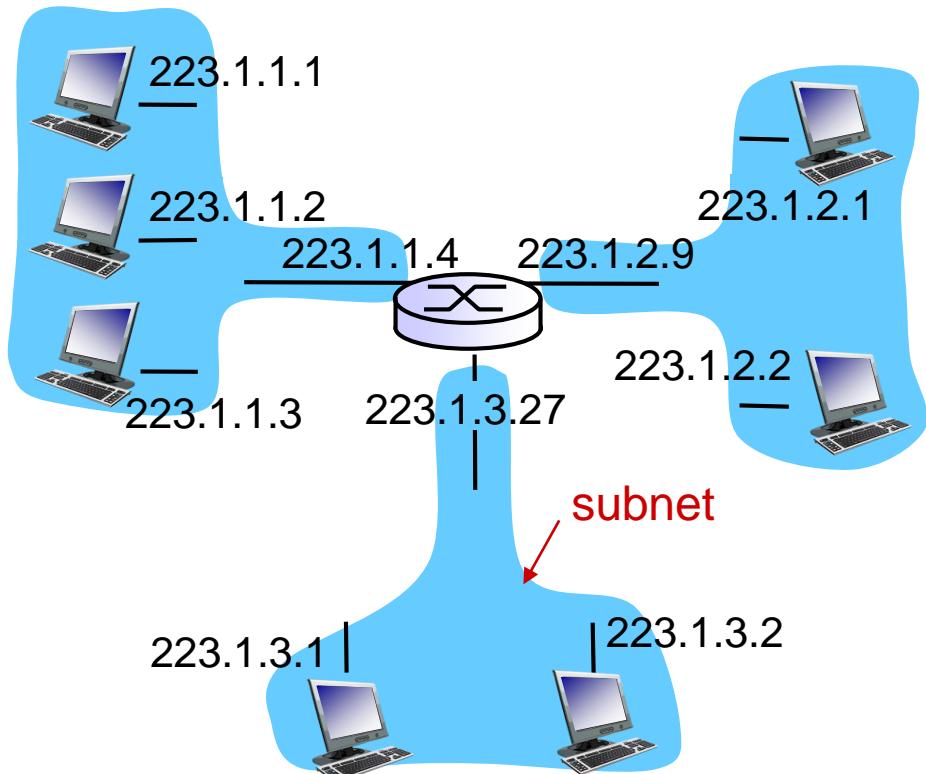
# Các mạng con

## ■ Địa chỉ IP:

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

## ■ Subnet là gì?

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

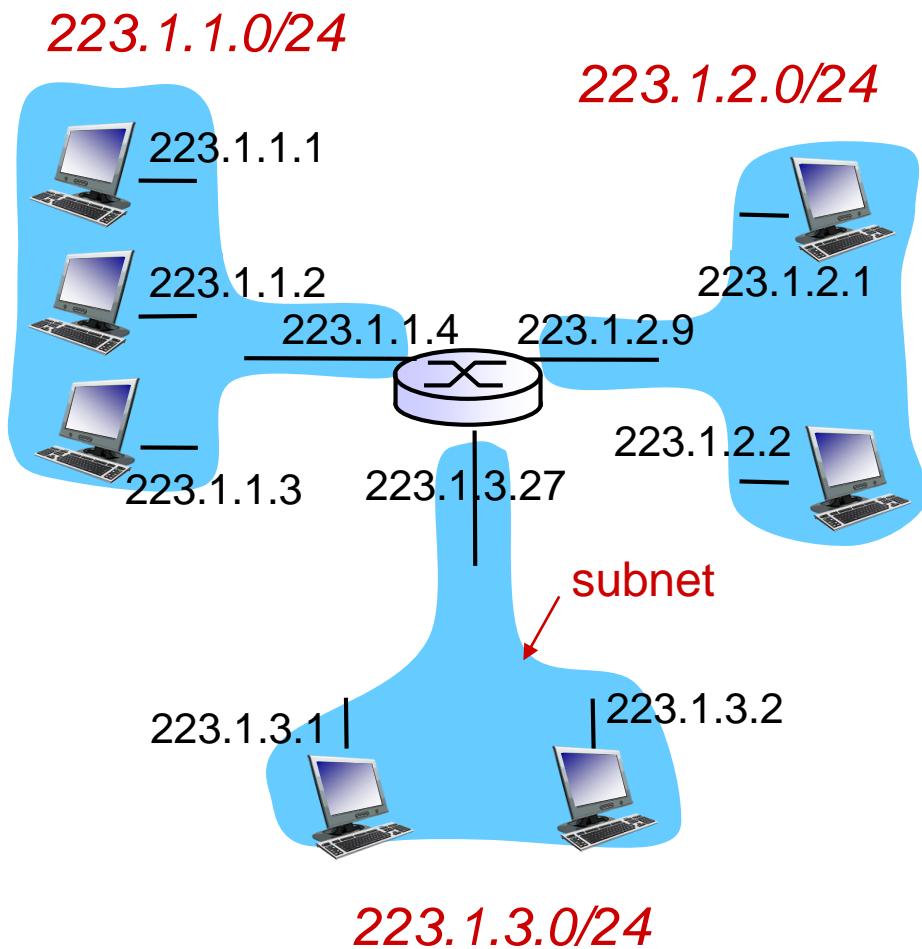


Mạng gồm 3 mạng con

# Các mạng con

## Phương pháp

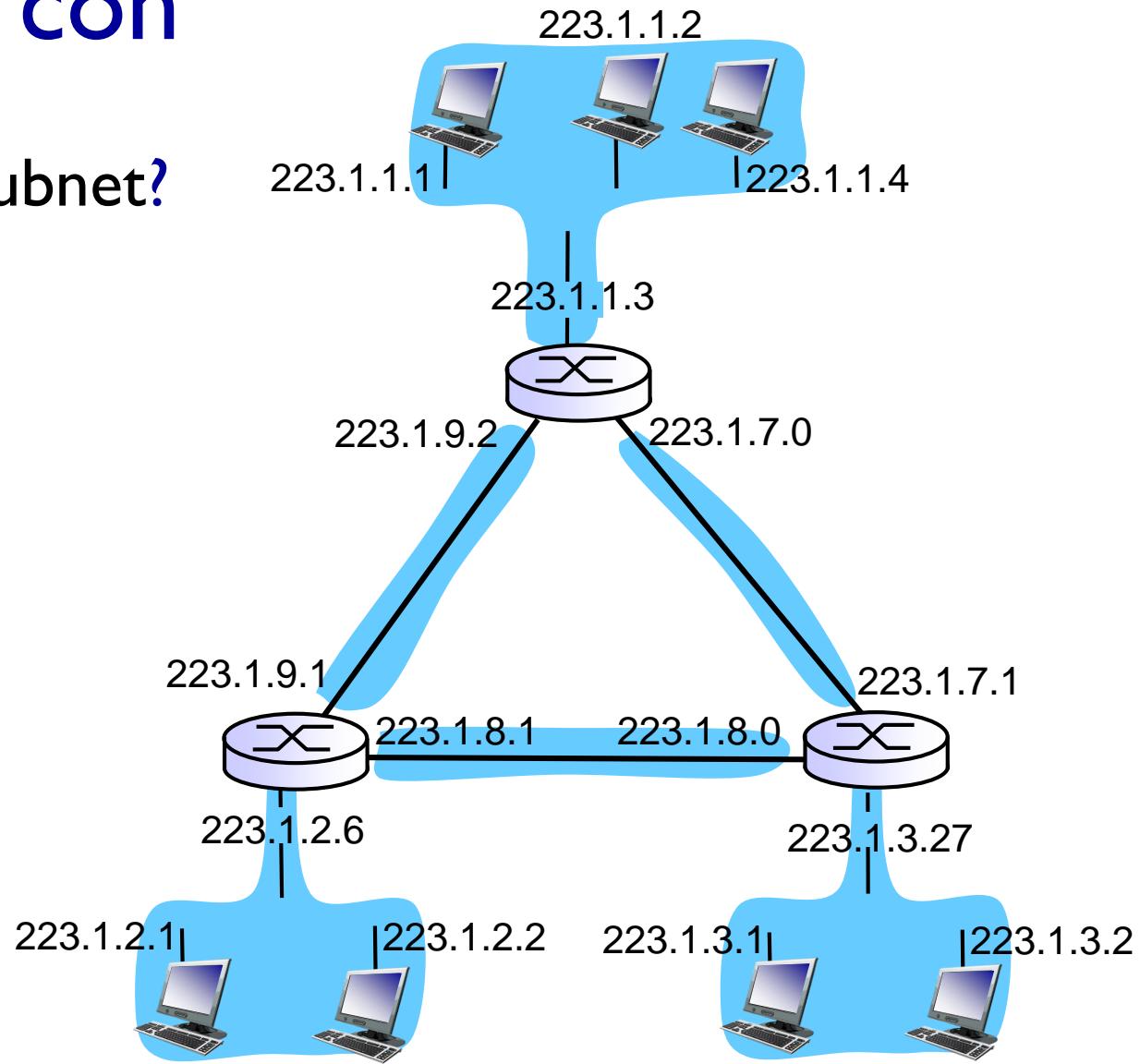
- Để xác định các subnet, tách mỗi giao diện từ host hoặc router, tạo thành các vùng mạng độc lập
- Mỗi mạng độc lập được gọi là một **subnet**



subnet mask: /24

# Các mạng con

Có bao nhiêu subnet?



# Địa chỉ IP: CIDR

## CIDR: Classless InterDomain Routing

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



# Làm thế nào để có được một địa chỉ IP?

---

Mã hóa cứng trong một tệp bởi người quản trị hệ thống

- Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- **DHCP: Dynamic Host Configuration Protocol:** tự động lấy địa chỉ từ server
    - “plug-and-play”

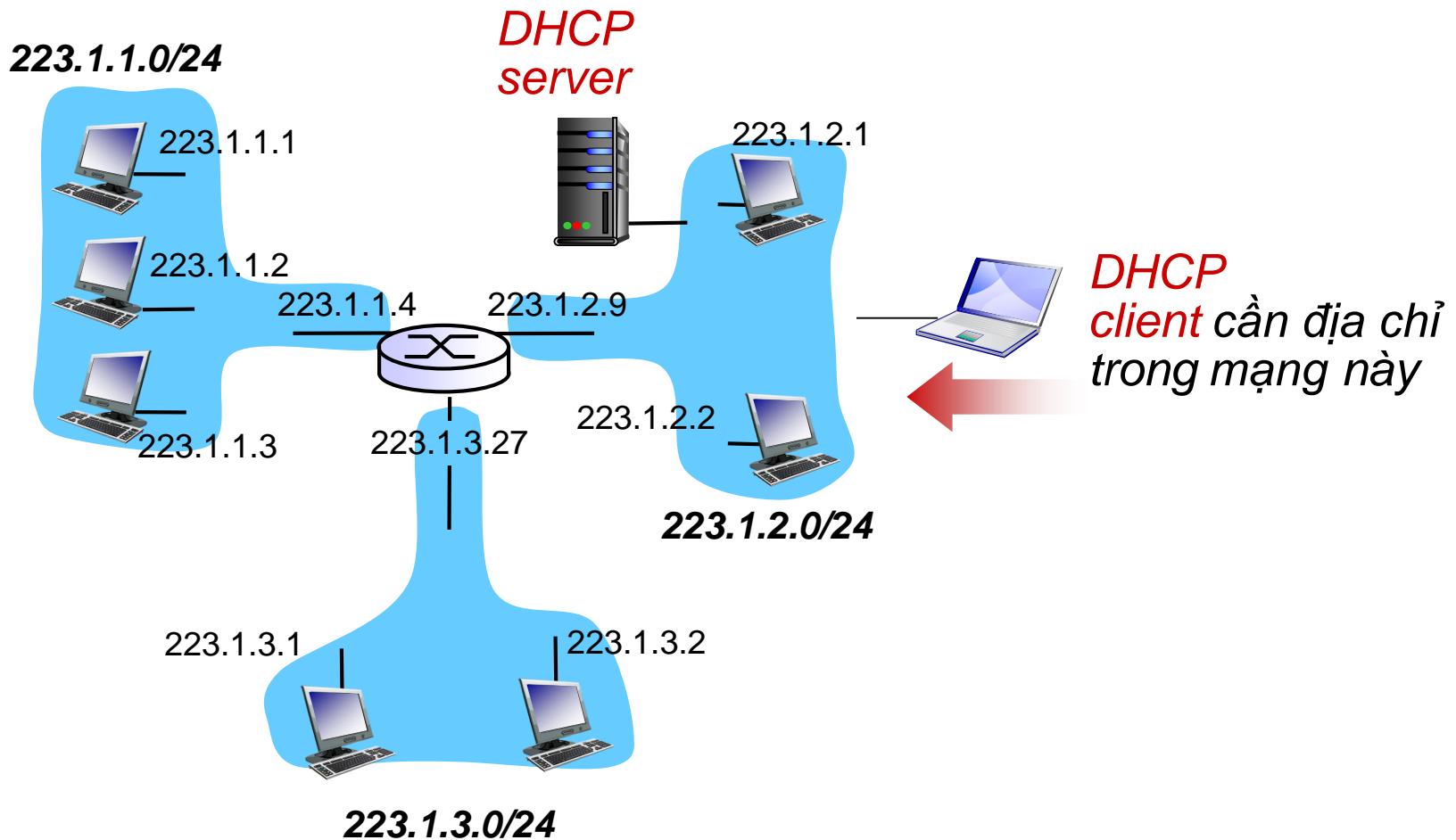
# DHCP: Dynamic Host Configuration Protocol

- **Mục đích:** cho phép host có được địa chỉ IP một cách tự động từ server mạng khi kết nối vào mạng
  - Có thể làm mới địa chỉ đang dùng
  - Cho phép dùng lại địa chỉ (chỉ giữ địa chỉ khi đang kết nối)
  - Hỗ trợ cho người dùng di động khi muốn kết nối vào mạng

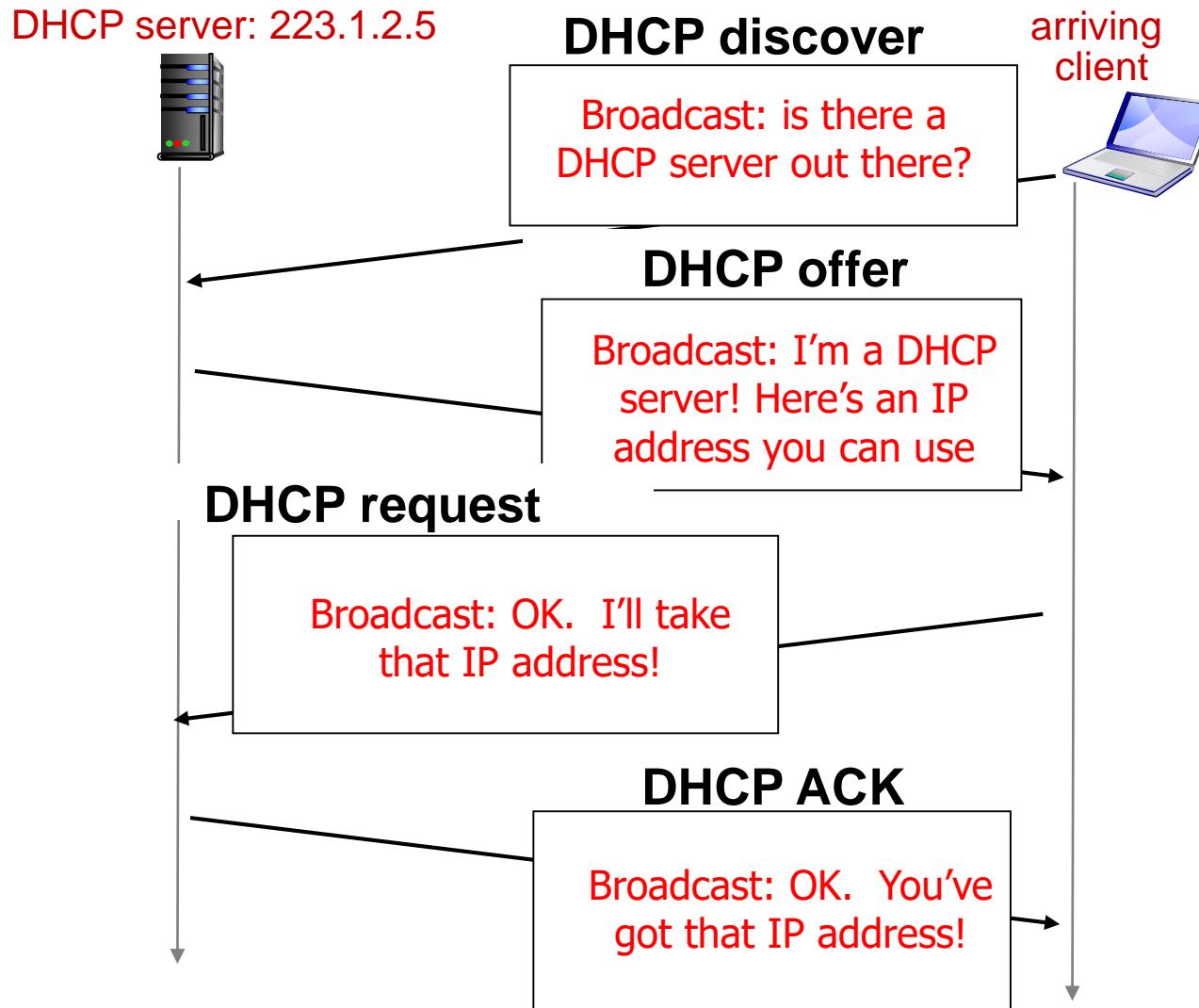
## *Khái quát DHCP:*

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

# Kịch bản DHCP client-server



# Kịch bản DHCP client-server

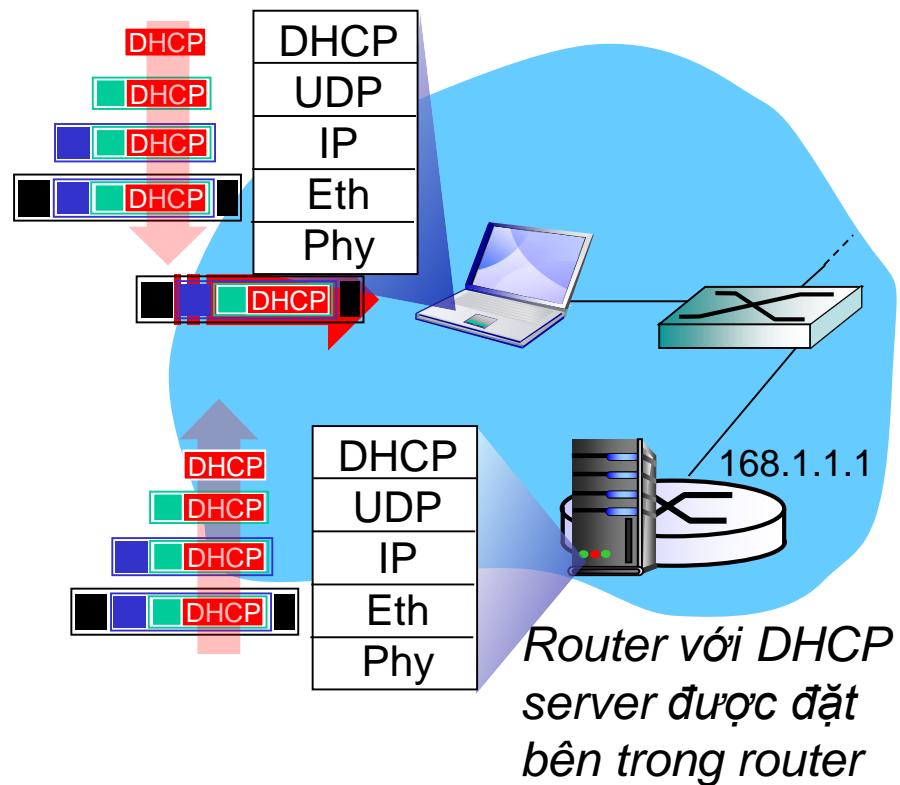


# DHCP: có nhiều địa chỉ IP hơn

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

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

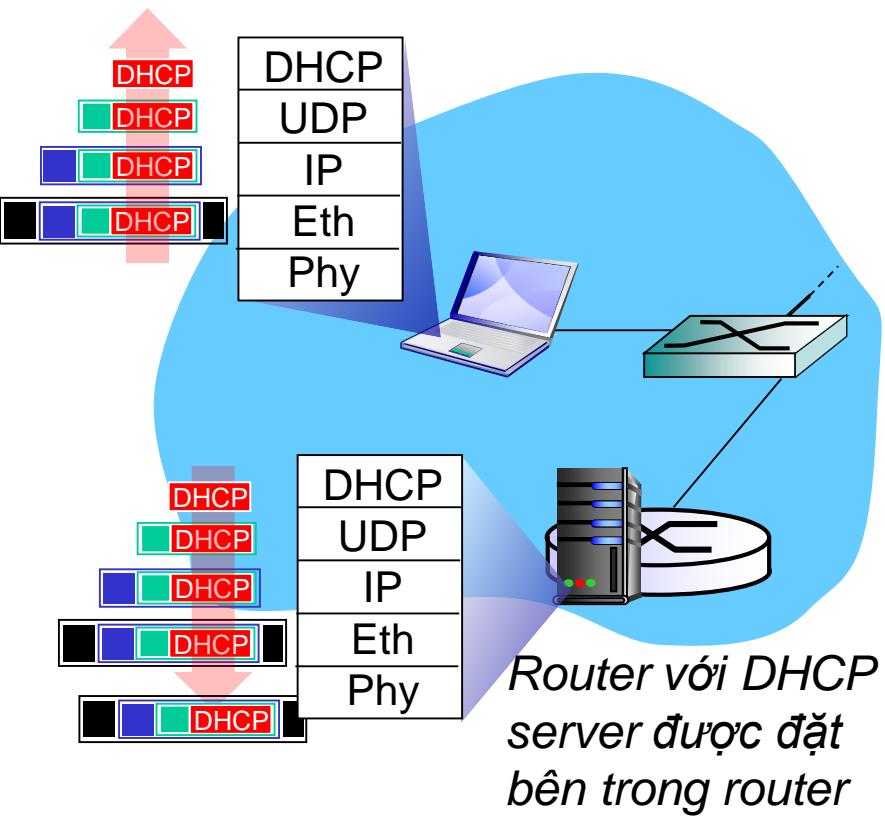
# DHCP: ví dụ



- Ethernet được cắt bỏ phần tiêu đề thành IP, IP được cắt bỏ phần tiêu đề thành UDP, UDP được cắt bỏ phần tiêu đề thành DHCP

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

# DHCP: ví dụ



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

# DHCP: đầu ra trong Wireshark (LAN ở nhà)

Đáp ứng

Message type: **Boot Request (1)**

Hardware type: Ethernet

Hardware address length: 6

Hops: 0

**Transaction ID: 0x6b3a11b7**

Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)

Client IP address: 0.0.0.0 (0.0.0.0)

Your (client) IP address: 0.0.0.0 (0.0.0.0)

Next server IP address: 0.0.0.0 (0.0.0.0)

Relay agent IP address: 0.0.0.0 (0.0.0.0)

**Client MAC address: Wistron\_23:68:8a (00:16:d3:23:68:8a)**

Server host name not given

Boot file name not given

Magic cookie: (OK)

Option: (t=53,l=1) **DHCP Message Type = DHCP Request**

Option: (61) Client identifier

Length: 7; Value: 010016D323688A;

Hardware type: Ethernet

Client MAC address: Wistron\_23:68:8a (00:16:d3:23:68:8a)

Option: (t=50,l=4) Requested IP Address = 192.168.1.101

Option: (t=12,l=5) Host Name = "nomad"

**Option: (55) Parameter Request List**

Length: 11; Value: 010F03062C2E2F1F21F92B

**1 = Subnet Mask; 15 = Domain Name**

**3 = Router; 6 = Domain Name Server**

44 = NetBIOS over TCP/IP Name Server

Yêu cầu

Message type: **Boot Reply (2)**

Hardware type: Ethernet

Hardware address length: 6

Hops: 0

**Transaction ID: 0x6b3a11b7**

Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)

**Client IP address: 192.168.1.101 (192.168.1.101)**

Your (client) IP address: 0.0.0.0 (0.0.0.0)

**Next server IP address: 192.168.1.1 (192.168.1.1)**

Relay agent IP address: 0.0.0.0 (0.0.0.0)

Client MAC address: Wistron\_23:68:8a (00:16:d3:23:68:8a)

Server host name not given

Boot file name not given

Magic cookie: (OK)

**Option: (t=53,l=1) DHCP Message Type = DHCP ACK**

**Option: (t=54,l=4) Server Identifier = 192.168.1.1**

**Option: (t=1,l=4) Subnet Mask = 255.255.255.0**

**Option: (t=3,l=4) Router = 192.168.1.1**

**Option: (6) Domain Name Server**

Length: 12; Value: 445747E2445749F244574092;

IP Address: 68.87.71.226;

IP Address: 68.87.73.242;

IP Address: 68.87.64.146

**Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."**

# Làm thế nào để có được một địa chỉ IP?

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

**A:** Lấy theo phần được phân bổ từ không gian địa chỉ của nhà cung cấp ISP

Khối của ISP      11001000 00010111 00010000 00000000    200.23.16.0/20

Tổ chức 0      11001000 00010111 00010000 00000000    200.23.16.0/23

Tổ chức 1      11001000 00010111 00010010 00000000    200.23.18.0/23

Tổ chức 2      11001000 00010111 00010100 00000000    200.23.20.0/23

...

.....

....

....

Tổ chức 7      11001000 00010111 00011110 00000000    200.23.30.0/23

# Định địa chỉ phân cấp: tích hợp định tuyến

Định địa chỉ 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

ISPs-R-Us

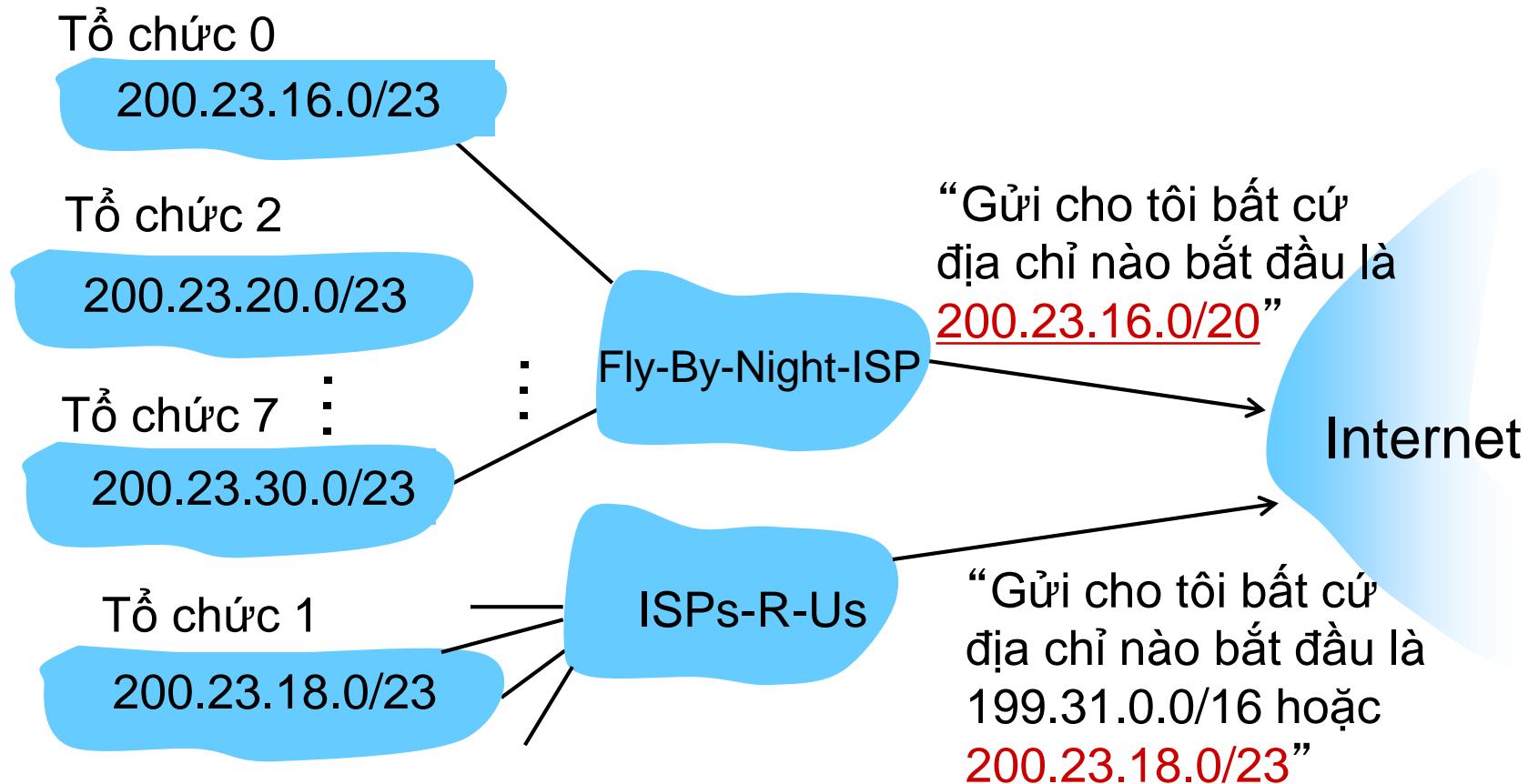
“Gửi cho tôi bất cứ  
địa chỉ nào bắt đầu là  
200.23.16.0/20”

Internet

“Gửi cho tôi bất cứ  
địa chỉ nào bắt đầu là  
199.31.0.0/16”

# Định địa chỉ phân cấp: định tuyến cụ thể hơn

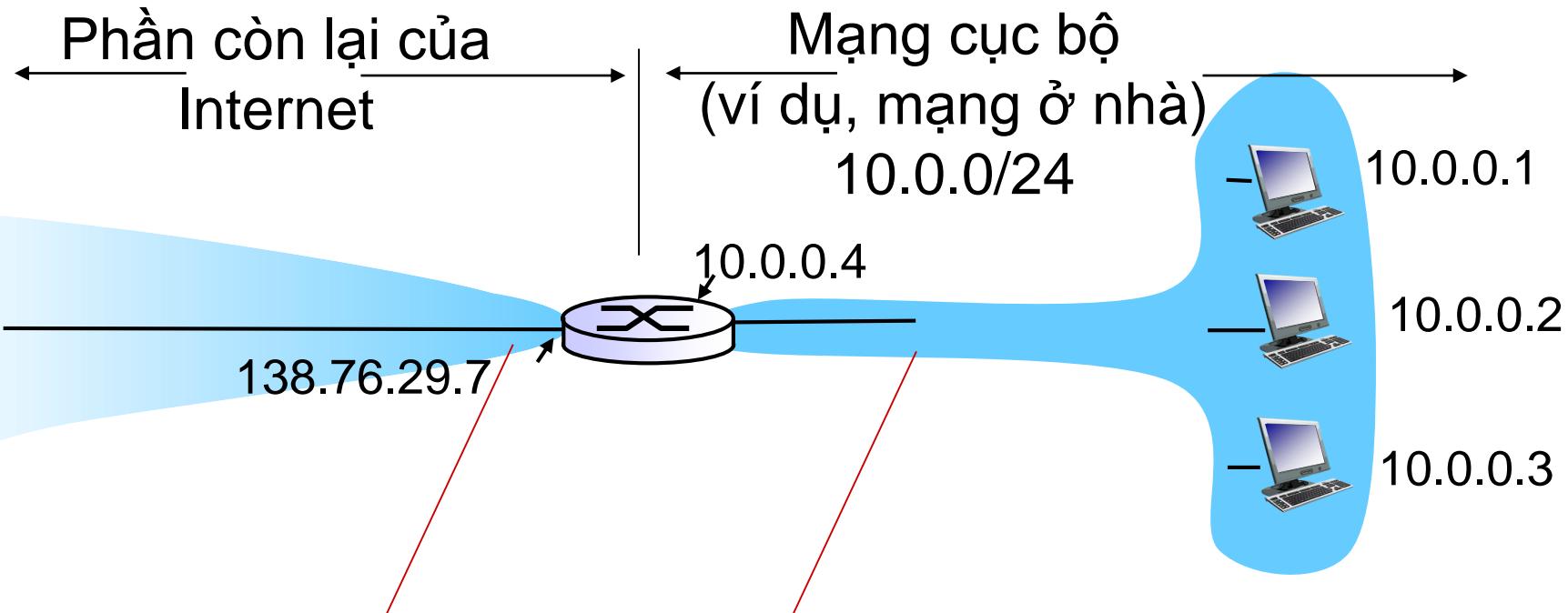
ISPs-R-Us có nhiều cách định tuyến cụ thể hơn đến Tổ chức I



## Định địa chỉ IP: ...

- **Q:** Làm thế nào một ISP có thể lấy được khối địa chỉ?
- **A:** ICANN: Internet Corporation for Assigned Names and Numbers <http://www.icann.org/>
  - Phân bổ địa chỉ
  - Quản lý DNS
  - Gán các tên miền, giải quyết tranh chấp

# NAT: network address translation



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

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

# NAT: network address translation

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

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

# NAT: network address translation

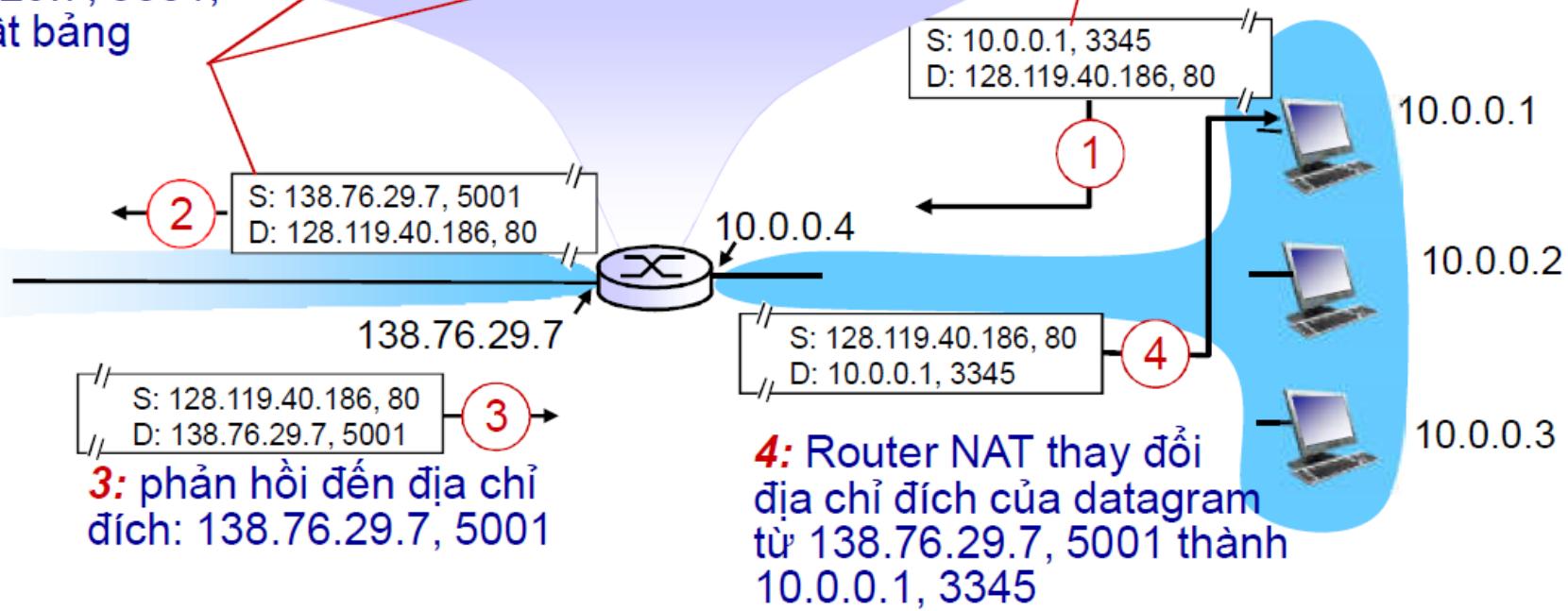
**Cài đặt:** Router NAT phải :

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

# NAT: network address translation

2: Router NAT thay đổi địa chỉ nguồn của datagram từ 10.0.0.1, 3345 thành 138.76.29.7, 5001, cập nhật bảng

Bảng chuyển đổi NAT	
Địa chỉ phía WAN	Địa chỉ phía LAN
138.76.29.7, 5001	10.0.0.1, 3345
.....	.....



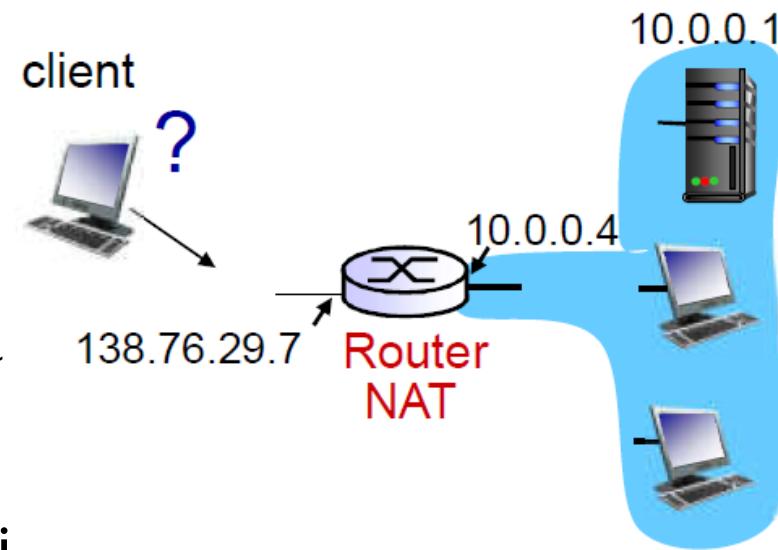
# NAT: network address translation

- Trường số hiệu cổng gồm 16-bit :
  - 60.000 kết nối đồng thời chỉ với một địa chỉ phía LAN!
- NAT hiện vẫn còn đang gây tranh cãi
  - Các router chỉ nên xử lý đến tầng 3
  - Vấn đề thiếu hụt địa chỉ IP nên được giải quyết bởi IPv6
  - Vi phạm thỏa thuận end-to-end
    - Các nhà thiết kế ứng dụng phải xem xét đến khả năng NAT, ví dụ ứng dụng P2P
- Vấn đề đi qua NAT: điều gì xảy ra nếu client muốn kết nối tới server sau NAT?

# NAT: network address translation

## ■ Vấn đề đi qua NAT

- Client muốn kết nối tới server có địa chỉ 10.0.0.1
  - Địa chỉ 10.0.0.1 của server được đặt trong mạng LAN (client không thể sử dụng địa chỉ này là địa chỉ đích)
  - Từ bên ngoài, client chỉ nhìn thấy địa chỉ NAT là 138.76.29.7
- **Giải pháp I:** Cấu hình tĩnh NAT để chuyển tiếp các yêu cầu kết nối đến tới cổng đã xác định của server
  - Ví dụ: (138.76.29.7, cổng 2500) sẽ luôn được chuyển tiếp tới (10.0.0.1, cổng 25000)



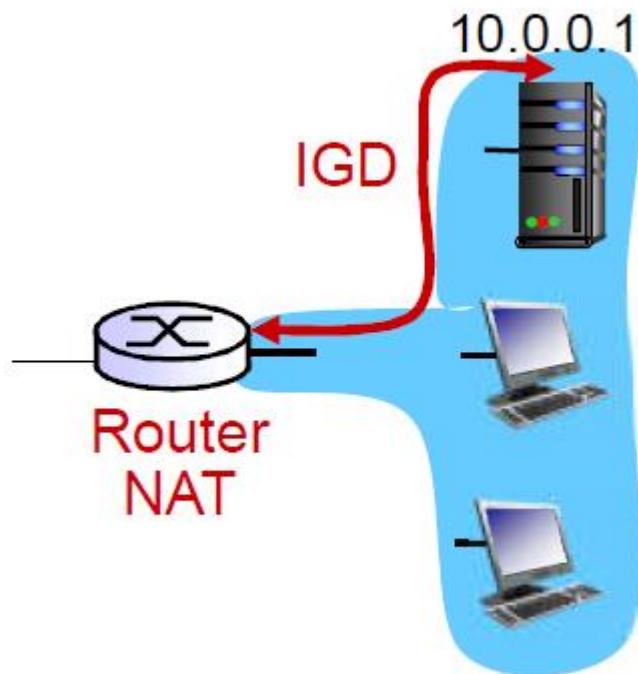
# NAT: network address translation

## ■ Vấn đề đi qua NAT

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

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

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



# NAT: network address translation

## ■ Vấn đề đi qua NAT

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



# Chapter 4: outline

## 4.1 Overview of Network layer

- data plane
- control plane

## 4.2 What's inside a router

## 4.3 IP: Internet Protocol

- datagram format
- fragmentation
- IPv4 addressing
- network address translation
- **IPv6**

## 4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

# IPv6: Lý do

- **Động lực thúc đẩy ban đầu:** không gian địa chỉ 32-bit sắp được cấp phát hết.
- **Động lực bổ sung:**
  - Định dạng tiêu đề (header) giúp tăng tốc độ xử lý/chuyển tiếp
  - Tiêu đề thay đổi giúp tạo điều kiện cho QoS

## **Định dạng IPv6 datagram:**

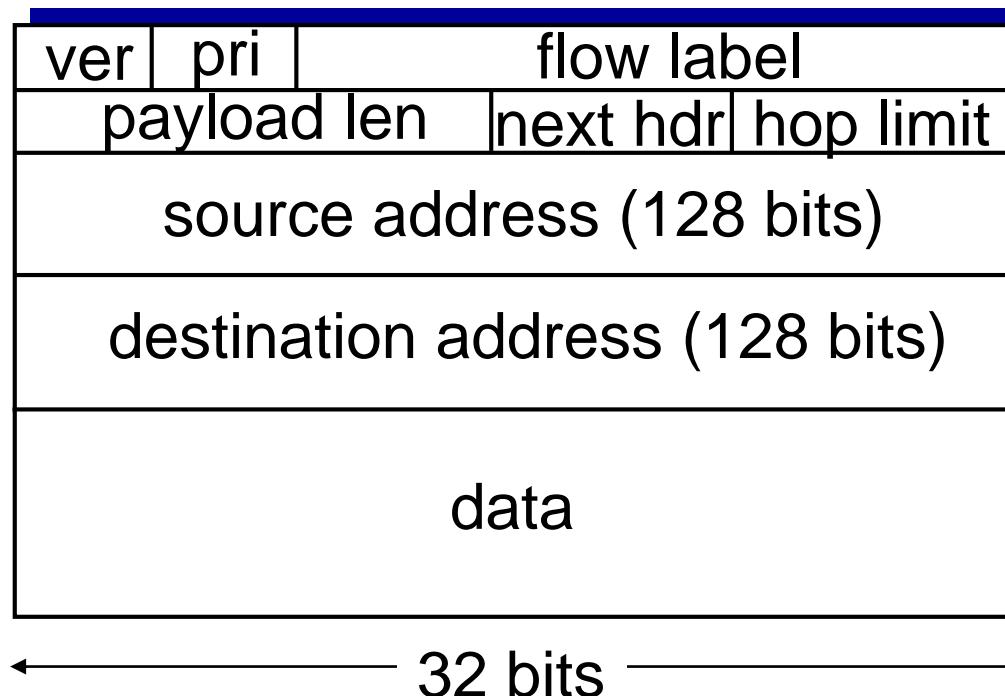
- Phần tiêu đề có chiều dài cố định 40 byte
- Không cho phép phân mảnh gói tin

# Định dạng IPv6 datagram

**Ưu tiên:** Xác định ưu tiên giữa các datagrams trong luồng

**Nhãn luồng:** Xác định các datagrams trong cùng một “luồng”

**Tiêu đề tiếp theo:** xác định giao thức tầng cao hơn cho dữ liệu

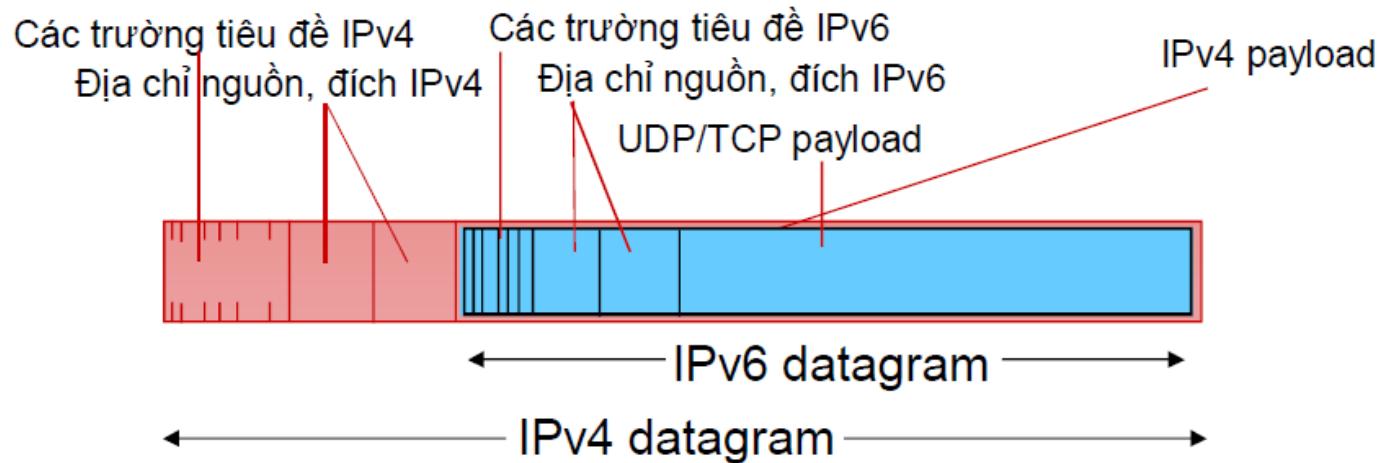


# Những thay đổi của IPv6 so với IPv4

- **Checksum:** bỏ hoàn toàn, nhằm giảm thời gian xử lý tại mỗi hop
- **Options:** được phép, nhưng nằm ngoài phần tiêu đề, được xác định trong trường “Next Header”
- **ICMPv6:** phiên bản mới của ICMP
  - Các loại thông điệp bổ sung, ví dụ: “Packet Too Big”
  - Các chức năng quản lý nhóm multicast

# Chuyển đổi từ IPv4 sang IPv6

- Không phải tất cả các router đều có thể được nâng cấp đồng thời
  - Không có ngày dành riêng cho việc chuyển đổi (flag days)
  - Mạng sẽ hoạt động như thế nào với việc sử dụng đồng thời các router IPv4 và IPv6?
- **Tunneling (đường hầm):** Payload của IPv6 datagram được mang trong IPv4 datagram giữa các router IPv4

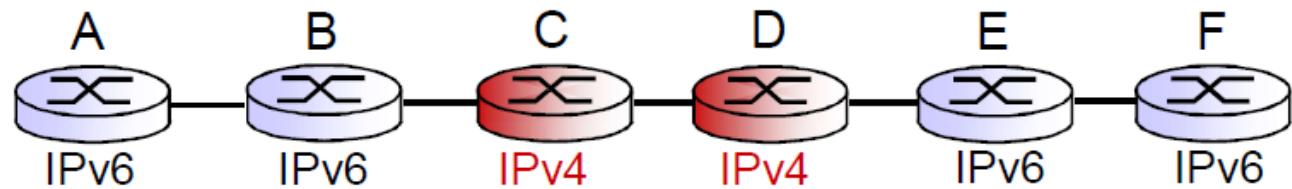


# Tunneling

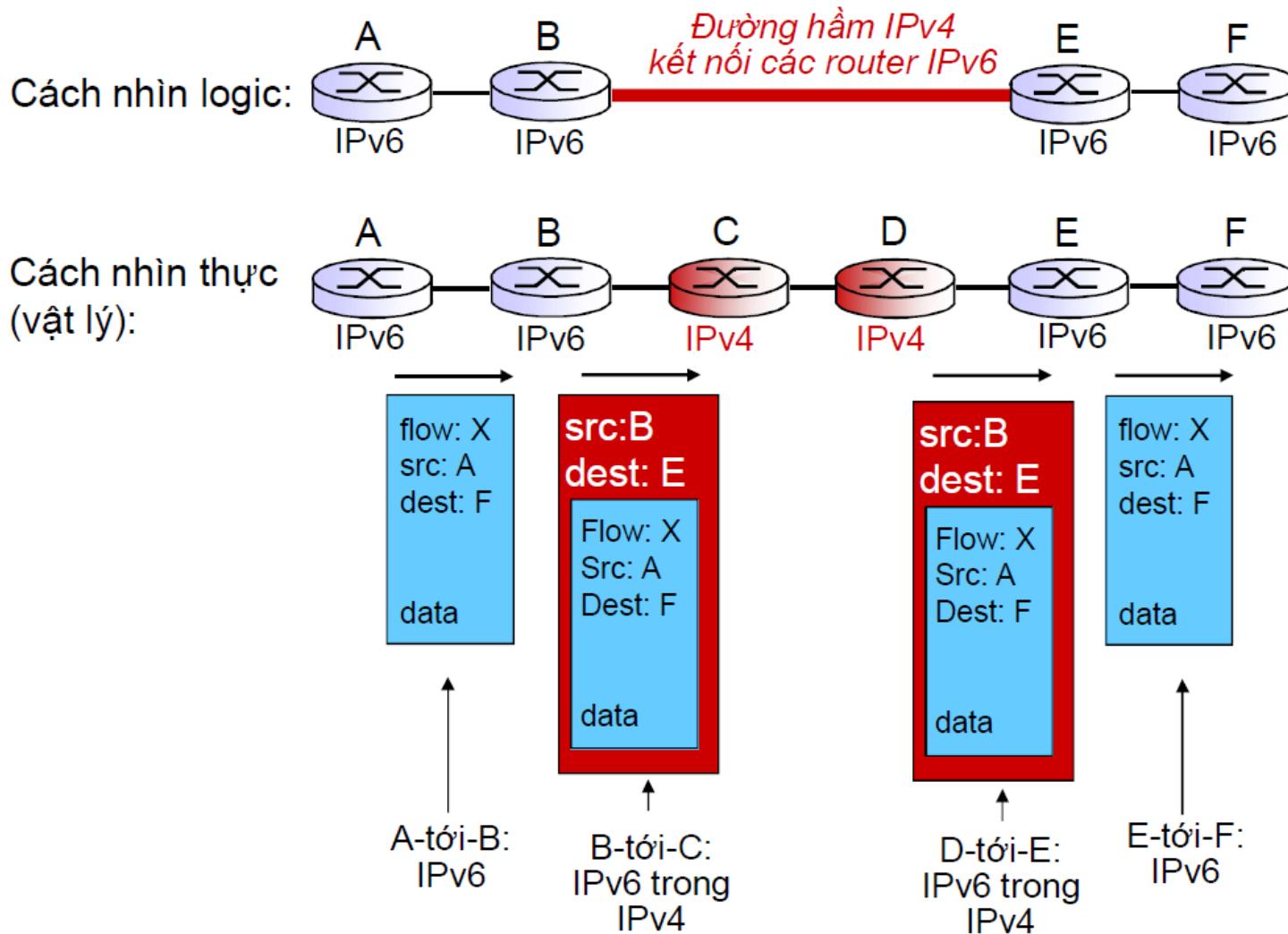
Cách nhìn logic:



Cách nhìn thực  
(vật lý):



# Tunneling



# IPv6: áp dụng

- Google: 8% các dịch vụ truy cập của các client thông qua IPv6
- NIST: 1/3 các tên miền của chính phủ Mỹ có khả năng IPv6
- *Thời gian dài để triển khai, sử dụng*
  - 20 năm và tiếp tục!
  - Nghĩ về những thay đổi mức ứng dụng trong 20 năm qua: WWW, Facebook, streaming media, Skype, ...
  - *Tại sao?*

# Chapter 4: outline

## 4.1 Overview of Network layer

- data plane
- control plane

## 4.2 What's inside a router

## 4.3 IP: Internet Protocol

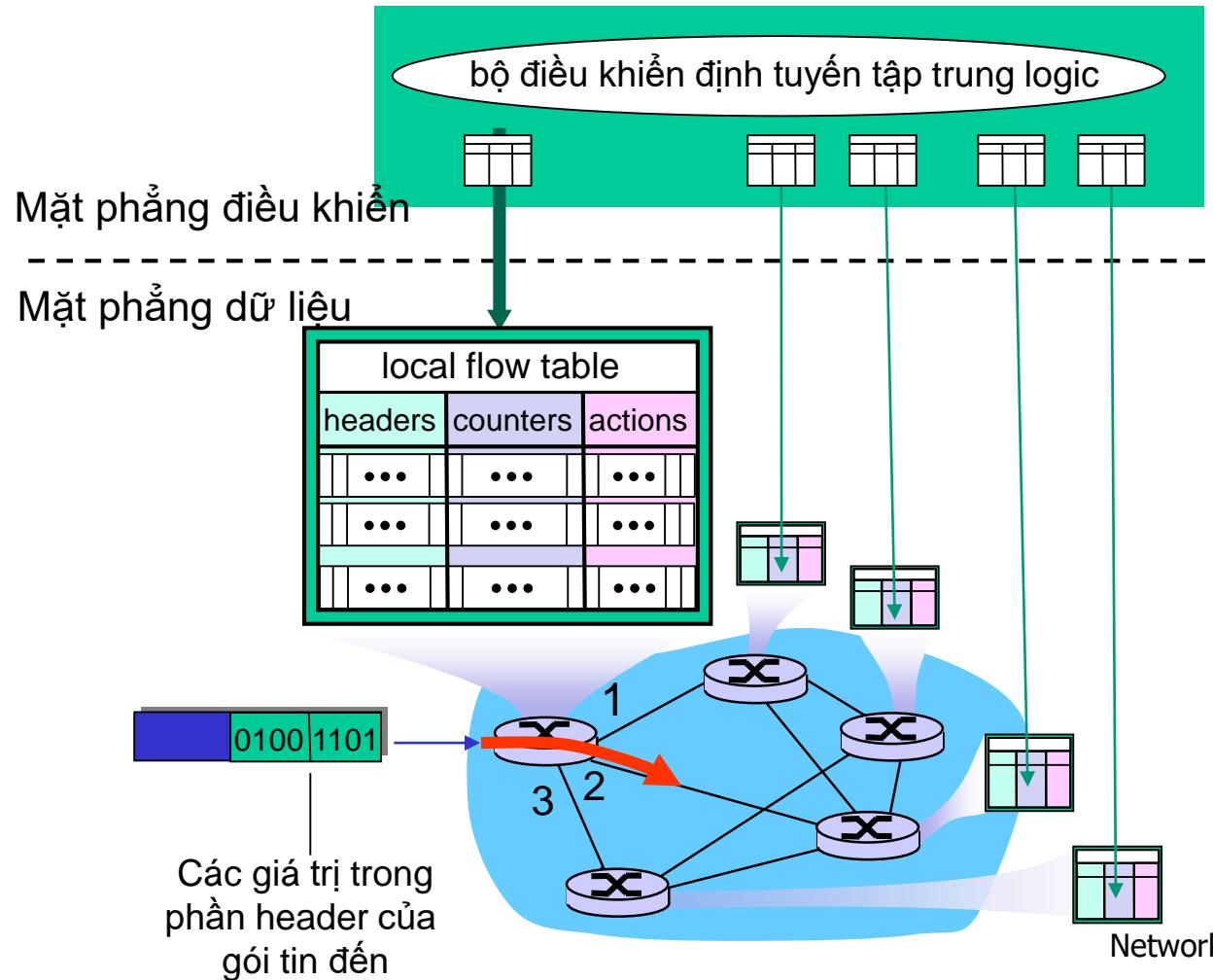
- datagram format
- fragmentation
- IPv4 addressing
- network address translation
- IPv6

## 4.4 Generalized Forward and SDN

- match
- action
- OpenFlow examples of match-plus-action in action

# Chuyển tiếp tổng quát và SDN

Mỗi router chứa một *flow table* được tính toán và phân phối bởi bộ điều khiển định tuyến tập trung logic



# Chuyển tiếp tổng quát và SDN

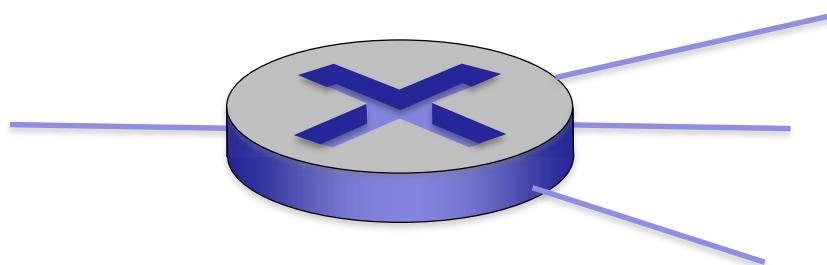
Mỗi router chứa một *flow table* được tính toán và phân phối bởi định tuyến tập trung hợp lý

- *Tập các giá trị trường header* mà gói tin đến sẽ được khớp với chúng.
- *Tập các bộ đếm*: được cập nhật khi các gói tin khớp với các mục trong flow table. Các bộ đếm này gồm số packet được khớp bởi table entry, và thời gian kể từ khi table entry được cập nhật lần cuối.
- *Tập các hành động*: các hành động được thực hiện khi một gói tin khớp với một flow table entry.
  - Các hành động này có thể là chuyển tiếp gói tin tới cổng ra tương ứng, loại bỏ gói tin, sao chép gói tin và gửi chúng tới nhiều cổng đầu ra, và/hoặc ghi lại các trường tiêu đề được chọn.

local flow table		
headers	counters	actions
...	...	...
...	...	...
...	...	...

# Mặt phẳng dữ liệu OpenFlow

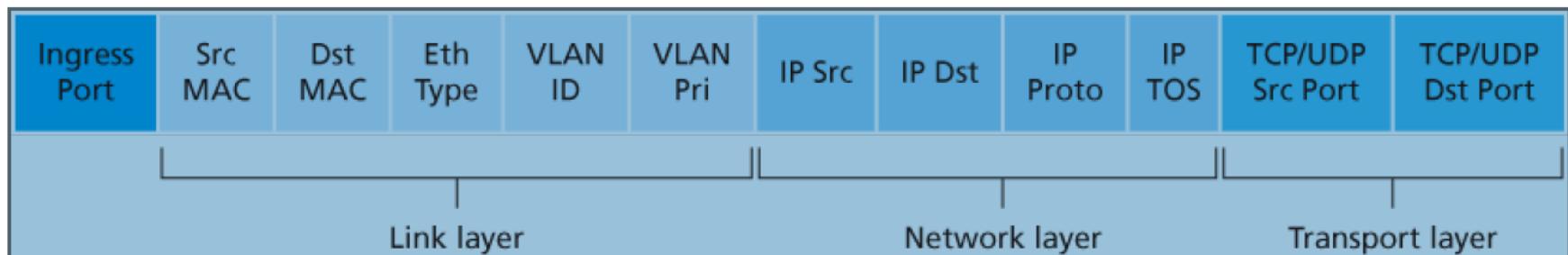
- *flow*: Được xác định bởi các trường tiêu đề
- generalized forwarding: simple packet-handling rules
  - *Pattern*: các giá trị *so khớp* trong các trường tiêu đề gói tin
  - *Actions*: *đối với các gói tin khớp*: loại bỏ, chuyển tiếp, thay đổi gói tin khớp hoặc gửi gói tin khớp tới bộ phận điều khiển.
  - *Priority*: Phân biệt các pattern chồng chéo
  - *Counters*: Số bytes và số packets



*Flow table trong một router (được tính toán và phân phối bởi bộ điều khiển) xác định các luật match+action của router*

# Mặt phẳng dữ liệu OpenFlow

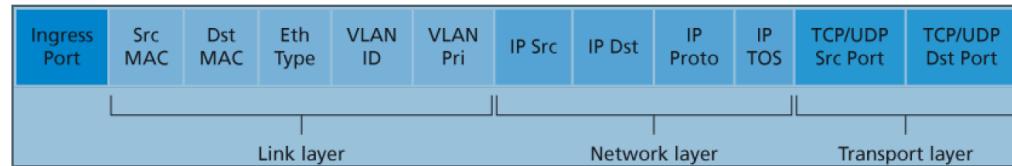
- *Pattern*: các giá trị *so khớp* trong các trường tiêu đề gói tin



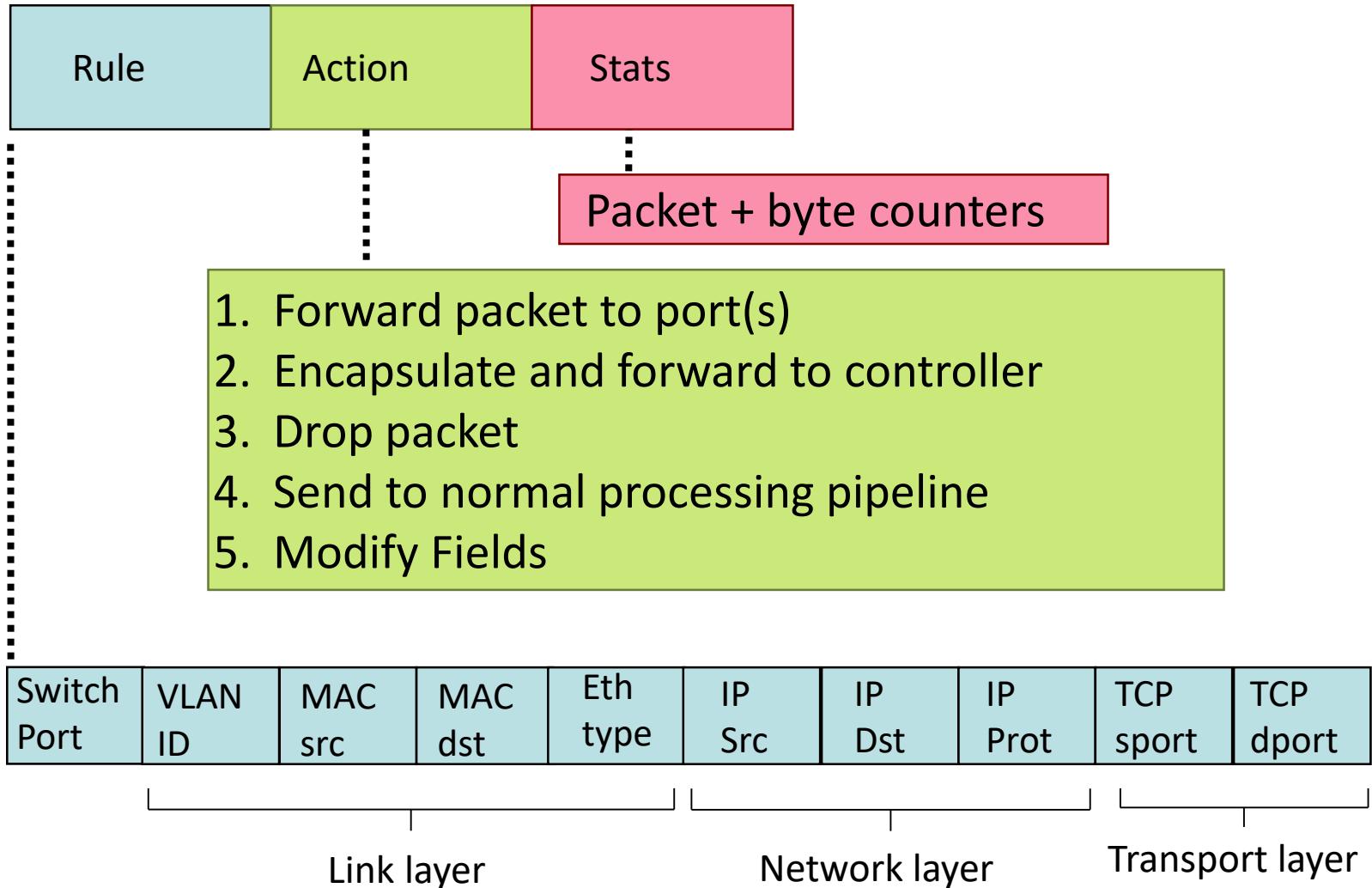
# Mặt phẳng dữ liệu OpenFlow

## Actions

- Chuyển tiếp
  - Một gói tin đến có thể được chuyển tiếp tới một cổng đầu ra vật lý cụ thể, truyền broadcast tới tất cả các cổng (trừ cổng nó đã đến) hay truyền multicast tới một số cổng đã chọn. Gói tin có thể được đóng gói và gửi tới bộ phận điều khiển từ xa. Bộ phận điều khiển từ xa có thể (hoặc không) thực hiện hành động lên gói tin, gồm cài đặt các flow table entries mới, và có thể chuyển trả gói tin về thiết bị để chuyển tiếp theo tập các flow table rules đã được cập nhật.
  - Loại bỏ
    - Một flow table entry không có hành động chứng tỏ gói tin so khớp nên bị loại bỏ.
  - Trường thay đổi (Modify-field)
    - Các giá trị trong 10 trường tiêu đề gói tin (trừ IP Protocol field) có thể được ghi lại (re-written) trước khi gói tin được chuyển tiếp tới cổng ra được chọn.



# OpenFlow: Flow Table Entries



# Examples

## Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

*IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6*

## Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

*do not forward (block) all datagrams destined to TCP port 22*

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

*do not forward (block) all datagrams sent by host 128.119.1.1*

# Examples

Destination-based layer 2 (switch) forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	port6

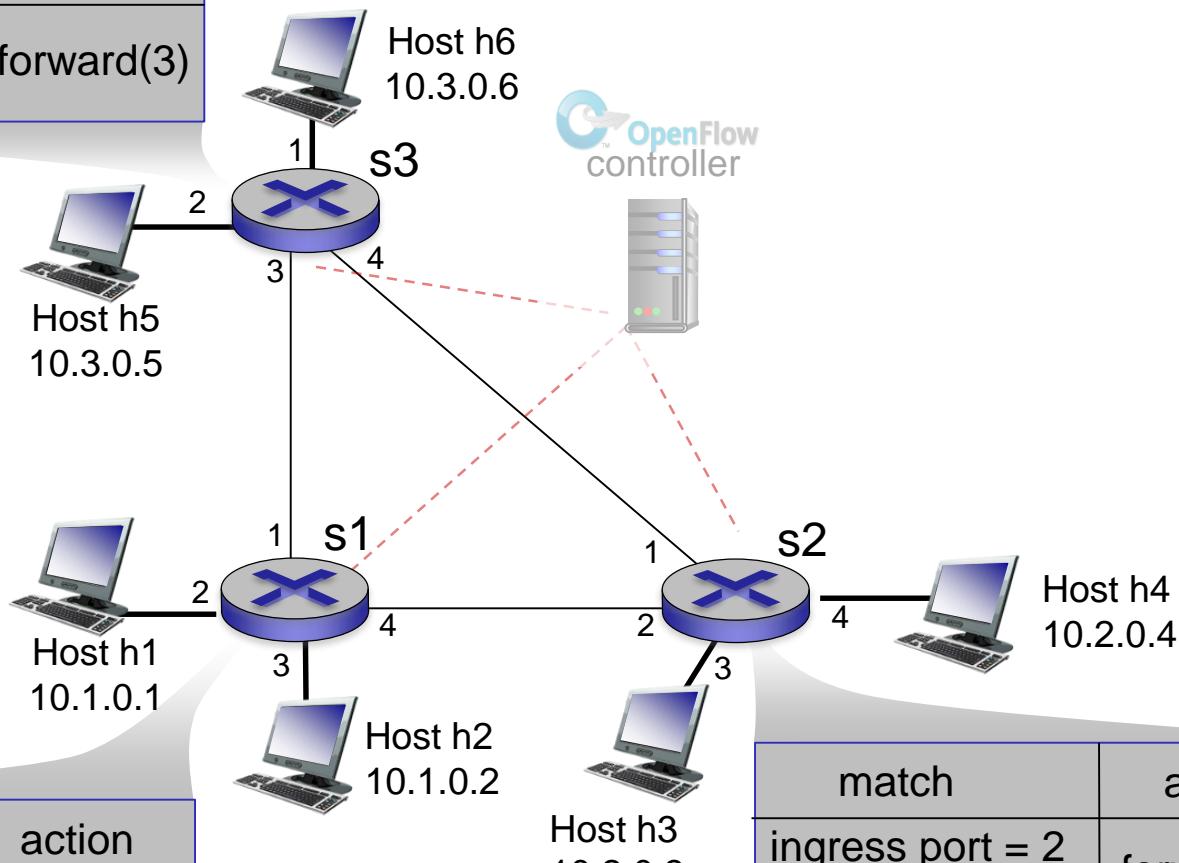
*layer 2 frames from MAC address 22:A7:23:11:E1:02  
should be forwarded to output port 6*

# OpenFlow abstraction

- *match+action*: unifies different kinds of devices
- Router
  - *match*: longest destination IP prefix
  - *action*: forward out a link
- Switch
  - *match*: destination MAC address
  - *action*: forward or flood
- Firewall
  - *match*: IP addresses and TCP/UDP port numbers
  - *action*: permit or deny
- NAT
  - *match*: IP address and port
  - *action*: rewrite address and port

# OpenFlow example

match	action
IP Src = 10.3.*.*	
IP Dst = 10.2.*.*	forward(3)



*Example:* datagrams from hosts h5 and h6 should be sent to h3 or h4, via s1 and from there to s2

match	action
ingress port = 1	
IP Src = 10.3.*.*	forward(4)
IP Dst = 10.2.*.*	

match	action
ingress port = 2	
IP Dst = 10.2.0.3	forward(3)
ingress port = 2	
IP Dst = 10.2.0.4	forward(4)

# Chapter 4: done!

4.1 Overview of Network layer: data plane and control plane

4.2 What's inside a router

4.3 IP: Internet Protocol

- datagram format
- fragmentation
- IPv4 addressing
- NAT
- IPv6

4.4 Generalized Forward and SDN

- match plus action
- OpenFlow example

**Question:** how do forwarding tables (destination-based forwarding) or flow tables (generalized forwarding) computed?

**Answer:** by the control plane (next chapter)

# Chương 5: Tầng mạng-mặt phẳng điều khiển

---

5.1 Giới thiệu

5.2 Các giải thuật định tuyến

5.3 Định tuyến Intra-AS  
trên Internet: OSPF

5.4 Định tuyến giữa các  
ISP: BGP

5.5 Mặt phẳng điều khiển  
SDN

5.6 ICMP

5.7 Quản lý mạng và  
SNMP

# 5.1 Giới thiệu

---

- 2 hướng tiếp cận để tính toán, duy trì và cài đặt forwarding table và flow table
  - Điều khiển tại mỗi router
    - Gồm cả chức năng chuyển tiếp và định tuyến trong mỗi router.
    - Mỗi router có thành phần định tuyến giao tiếp với các thành phần định tuyến trong các router khác để tính toán forwarding table.
    - Được sử dụng trong Internet qua nhiều thập kỷ.
    - OSPF và BGP

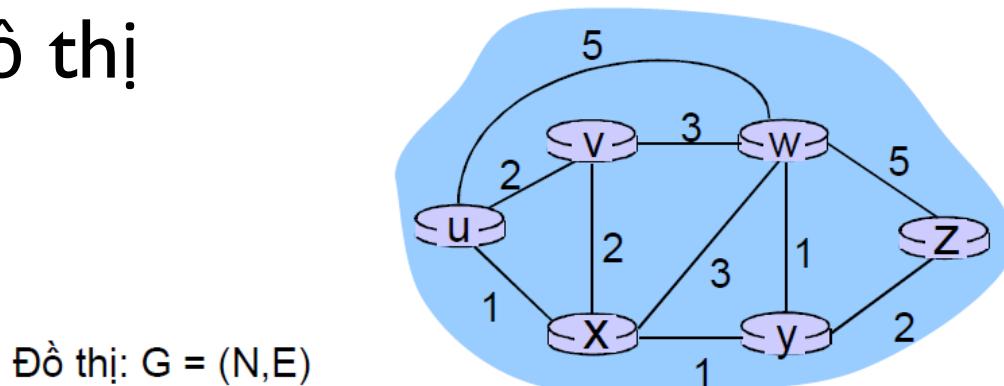
# 5. I Giới thiệu

---

- 2 hướng tiếp cận để tính toán, duy trì và cài đặt forwarding table và flow table
  - Điều khiển tập trung logic
    - Bộ điều khiển trung tâm tính toán và phân phối các forwarding tables sẽ được sử dụng bởi mỗi router.
    - Chuyển tiếp tổng quát cho phép router thực hiện chức năng chuyển tiếp IP truyền thống cũng như các chức năng khác (chia sẻ tải, tường lửa, và NAT) mà trước đó vốn được thực hiện ở các hộp trung gian riêng biệt

## 5.2 Các giải thuật định tuyến

- Tìm đường đi tối ưu từ đầu gửi tới đầu nhận cho gói tin trong mạng router.
  - Tối ưu  $\Leftrightarrow$  chi phí thấp nhất
- Mô hình đồ thị



Đồ thị:  $G = (N, E)$

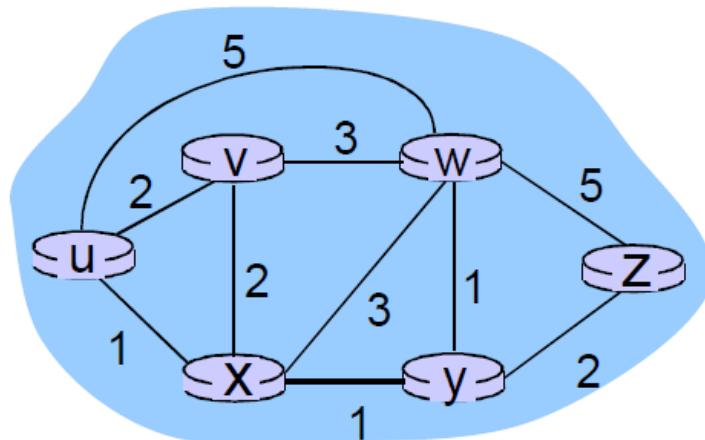
$N = \text{Tập các router} = \{ u, v, w, x, y, z \}$

$E = \text{Tập các liên kết} = \{ (u, v), (u, x), (v, x), (v, w), (w, x), (x, y), (w, y), (y, z) \}$

Chú ý: Mô hình đồ thị cũng được dùng trong các ngữ cảnh mạng khác, như P2P, trong đó  $N$  là tập các peer và  $E$  là tập các kết nối TCP.

## 5.2 Các giải thuật định tuyến

- Mô hình đồ thị: Các chi phí



$c(x,x')$  = chi phí của kết nối  $(x,x')$   
ví dụ:  $c(w,z) = 5$

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

Chi phí của đường đi  $(x_1, x_2, x_3, \dots, x_p)$  =  $c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

*Hỏi:* Chi phí thấp nhất đường đi từ u đến z là bao nhiêu?

*Giải thuật định tuyến:* giải thuật tìm đường đi có chi phí thấp nhất

## 5.2 Các giải thuật định tuyến

---

- Phân loại giải thuật định tuyến

Hỏi: Thông tin là tập trung hay không tập trung?

- Tập trung:

- Tất cả các router đều có thông tin đầy đủ về cấu trúc mạng và chi phí của các liên kết
    - Giải thuật “link state” (trạng thái kết nối)

- Không tập trung:

- Router biết về các hàng xóm có kết nối vật lý với nó và chi phí liên kết tới các hàng xóm này.
    - Lặp lại quá trình tính toán, trao đổi thông tin với các hàng xóm
    - Giải thuật “distance vector” (véc-tơ khoảng cách)

## 5.2 Các giải thuật định tuyến

---

- Phân loại giải thuật định tuyến

Hỏi: Động hay tĩnh?

- **Tĩnh:**

- Việc định tuyến thay đổi chậm theo thời gian

- **Động:**

- Việc định tuyến thay đổi nhanh hơn:

- Cập nhật định kỳ

- Phản ứng với những thay đổi chi phí liên kết

## 5.2 Các giải thuật định tuyến: Link state

---

### ■ Giải thuật Dijkstra

- Tất cả các nút đều biết được cấu trúc mạng và chi phí của các liên kết trên mạng
  - Được thực hiện bằng cách “quảng bá trạng thái liên kết”
  - Tất cả các nút có thông tin giống nhau
- Tính toán chi phí thấp nhất đường đi từ một nút (“nguồn”) đến tất cả các nút khác.
  - Cho **bảng chuyển tiếp** của nút đó
  - Lặp: sau k lần duyệt, sẽ biết được chi phí thấp nhất tới k đích

## 5.2 Các giải thuật định tuyến: Link state

---

### ■ Giải thuật Dijkstra: Ký hiệu

- $c(x,y)$ : chi phí liên kết từ nút  $x$  tới  $y$ ;  $= \infty$  nếu không có kết nối trực tiếp đến nút lân cận
- $u$ : nút nguồn,
- $v$ : hàng xóm của  $u$  (liền kề và có liên hệ với  $u$ , không bị ngắt quãng)
- $D(v)$ : giá trị hiện tại của chi phí đường đi từ nguồn đến đích
- $p(v)$ : nút trước nằm trên đường đi từ nguồn đến  $v$
- $N'$ : tập các nút mà chi phí đường đi thấp nhất đã được xác định

## 5.2 Các giải thuật định tuyến: Link state

---

### ■ Giải thuật Dijkstra:

1 *Khởi tạo:*

2  $N' = \{u\}$

3 for tất cả các nút v

4 if v kề với u

5      then  $D(v) = c(u,v)$

6 else  $D(v) = \infty$

7

8 *Lặp*

9 tìm w không thuộc  $N'$  mà  $D(w)$  là nhỏ nhất

10 thêm w vào  $N'$

11 Cập nhật  $D(v)$  cho tất cả v kề với w và không thuộc  $N'$ :

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

13 /\* chi phí mới đến v là chính giá trị cũ hoặc chi phí đường đi ngắn

14 nhất đã tính tới w cộng với chi phí từ w tới v \*/

15 **cho đến khi tất cả các nút nằm trong  $N'$**

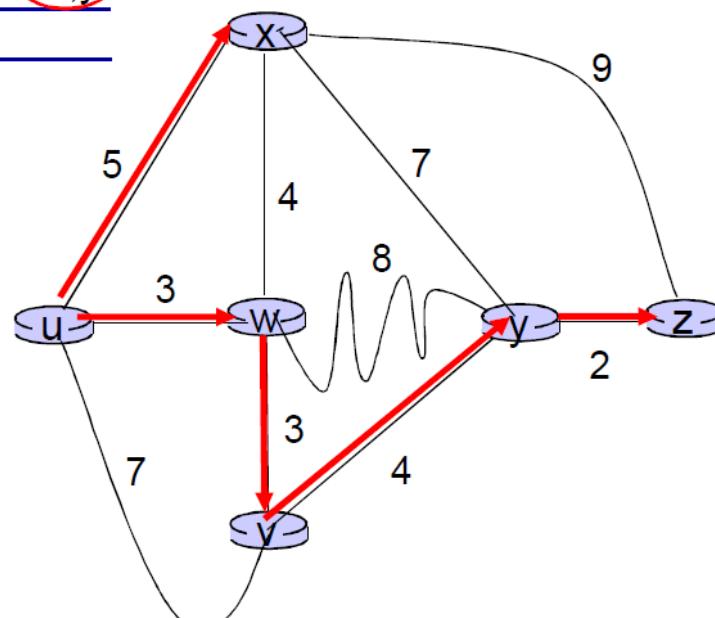
## 5.2 Các giải thuật định tuyến: Link state

### ■ Giải thuật Dijkstra: ví dụ

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

*Chú ý:*

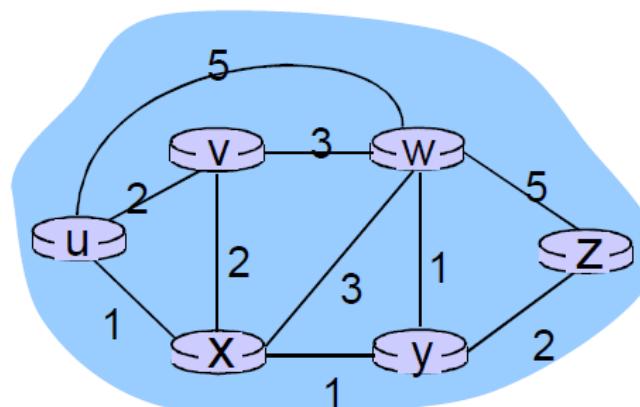
- ❖ Xây dựng cây đường đi ngắn nhất bằng cách đi tìm các nút trước.
- ❖ Các mối quan hệ có thể tồn tại (có thể được chia tùy ý)



## 5.2 Các giải thuật định tuyến: Link state

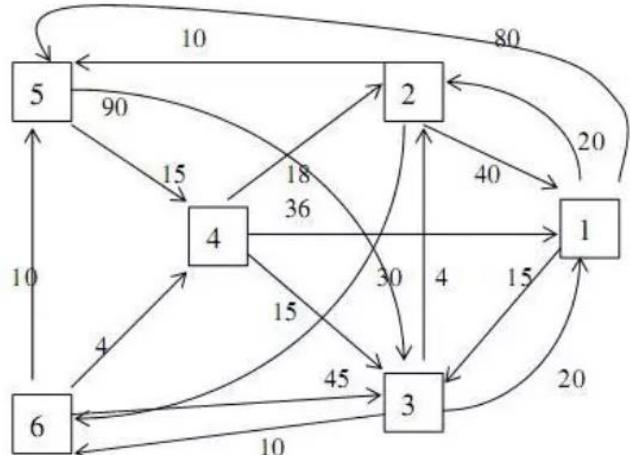
### ▪ Giải thuật Dijkstra: ví dụ

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	uxyw					4,y
5	uxyvwz					



## 5.2 Các giải thuật định tuyến: Link state

### ■ Giải thuật Dijkstra:



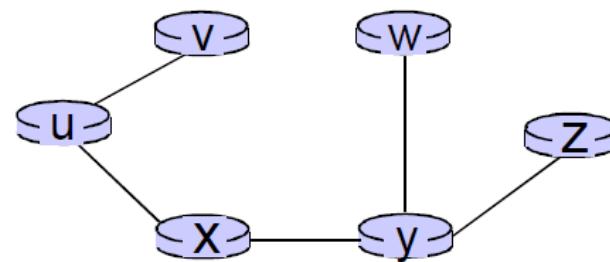
Bước lặp	Đường đi ngắn nhất là đường đi từ đỉnh 1	đến đỉnh	Chiều dài của đường đi ngắn nhất từ đỉnh s (=1) đến các đỉnh khác : tsnn[]					
			1	2	3	4	5	6
Bước1	1→3	3	-	20	15	$\infty$	80	$\infty$
Bước2	1→3→2	2	-	19	-			25
Bước3	1→3→6	6	-	-	-		29	25
Bước4	1→3→6→4	4	-	-	-	29		-
Bước5	1→3→2→5	5	-	-	-	-	29	-

## 5.2 Các giải thuật định tuyến: Link state

---

- Giải thuật Dijkstra: ví dụ

Kết quả cây đường đi ngắn nhất từ u:

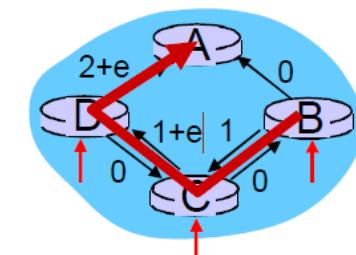
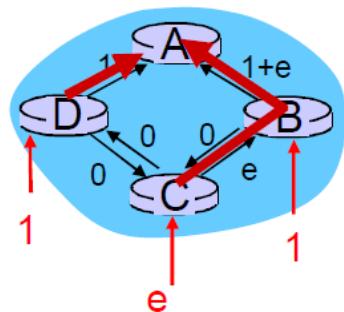


Kết quả bảng chuyển tiếp trong u:

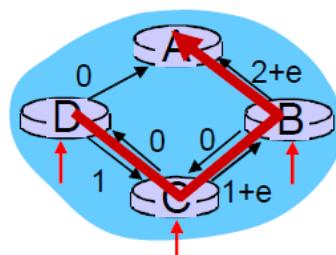
Đích	Liên kết
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

## 5.2 Các giải thuật định tuyến: Link state

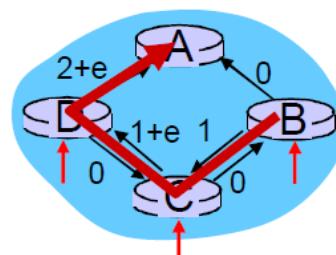
- Độ phức tạp của giải thuật Dijkstra: n nút
  - Mỗi lần duyệt: cần kiểm tra lại tất cả các nút w không thuộc N
  - $n(n+1)/2$  phép so sánh:  $O(n^2)$
  - Có thể cài đặt hiệu quả hơn:  $O(n \log n)$
- Có thể dao động:
  - Ví dụ: chi phí kết nối bằng tổng lưu lượng mạng:



Với chi phí đã biết,  
tính toán định tuyến  
mới.... kết quả có chi  
phí mới



Với chi phí đã biết,  
tính toán định tuyến  
mới.... kết quả có chi  
phí mới



Với chi phí đã biết,  
tính toán định tuyến  
mới.... kết quả có chi  
phí mới

## 5.2 Các giải thuật định tuyến: Distance-Vector

---

*Công thức Bellman-Ford*

Cho

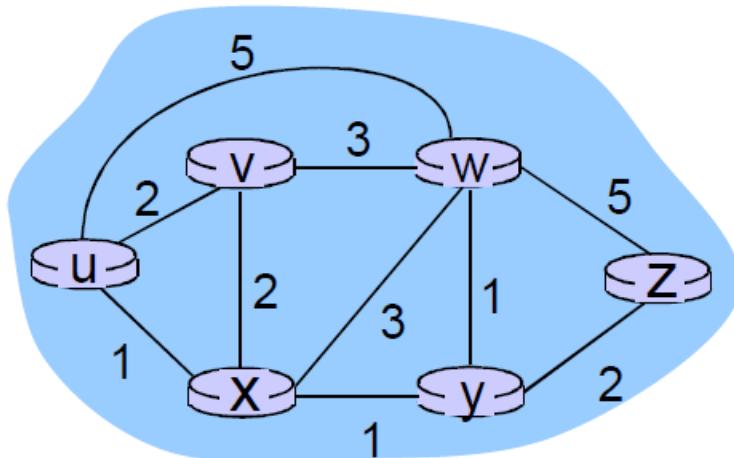
$d_x(y) :=$  chi phí thấp nhất đường đi từ  $x$  đến  $y$   
thì

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

Chi phí từ lân cận  $v$  tới đích  $y$   
Chi phí tới lân cận  $v$

$\min$  được tính trên tất cả các lân cận  $v$  của  $x$

## 5.2 Các giải thuật định tuyến: Distance-Vector



Có:  $d_v(z) = 5$ ,  $d_x(z) = 3$ ,  $d_w(z) = 3$

Công thức B-F cho:

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

Giá trị nhỏ nhất đạt được của nút là hop kế tiếp trong đường đi ngắn nhất, được dùng trong bảng chuyển tiếp.

## 5.2 Các giải thuật định tuyến: Distance-Vector

---

- ❖  $D_x(y)$  = ước lượng chi phí thấp nhất từ  $x$  đến  $y$ 
  - $x$  duy trì véc-tơ khoảng cách  $\mathbf{D}_x = [D_x(y): y \in N]$
- ❖ Nút  $x$ :
  - Biết chi phí đến mỗi nút lân cận  $v$ :  $c(x,v)$
  - Duy trì véc-tơ khoảng cách của các nút lân cận của. Với mỗi nút lân cận  $v$ ,  $x$  duy trì  $\mathbf{D}_v = [D_v(y): y \in N]$

## 5.2 Các giải thuật định tuyến: Distance-Vector

---

- **Ý tưởng chính:**

- Mỗi nút định kỳ gửi ước lượng véc-tơ khoảng cách của nó đến các nút lân cận
- Khi x nhận được ước lượng DV mới từ nút lân cận, nó sẽ cập nhật DV của nó bằng cách dùng công thức B-F:

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

- Dưới các điều kiện tự nhiên, ước lượng  $D_x(y)$  sẽ hội tụ về giá trị chi phí thấp nhất thực tế  $d_x(y)$

## 5.2 Các giải thuật định tuyến: Distance-Vector

- **Lặp, không đồng bộ:** mỗi lần lặp cục bộ có nguyên nhân từ:
  - Thay đổi chi phí liên kết cục bộ
  - Thông báo cập nhật DV từ nút lân cận
- **Phân bố:**
  - Mỗi nút chỉ thông báo đến các nút lân cận khi DV của nó thay đổi
    - Các lân cận sau đó sẽ thông báo đến các lân cận khác của nó nếu cần

*Mỗi nút:*

*Chờ* cho (thay đổi trong chi phí liên kết cục bộ hoặc thông báo từ nút lân cận)

*Tính toán lại* các ước lượng

Nếu DV tới bất kỳ đích nào có thay đổi, thì *thông báo* cho các nút lân cận

## 5.2 Các giải thuật định tuyến: Distance-Vector

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

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

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

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

Bảng  
nút x

	x	y	z
x	0	2	7
y	$\infty$	$\infty$	$\infty$
z	$\infty$	$\infty$	8

Chi phí tới

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

Bảng  
nút y

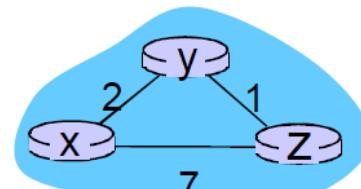
	x	y	z
x	$\infty$	$\infty$	$\infty$
y	2	0	1
z	$\infty$	$\infty$	$\infty$

Chi phí tới

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

Chi phí tới

Thời gian



## 5.2 Các giải thuật định tuyến: Distance-Vector

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

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

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

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

**Bảng nút x**

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

**Chi phí tới**

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

**Chi phí tới**

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

**Bảng nút y**

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

**Chi phí tới**

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

**Chi phí tới**

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

**Bảng nút z**

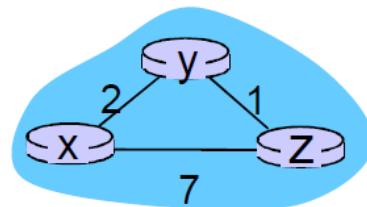
	x	y	z
x	$\infty$	$\infty$	$\infty$
y	$\infty$	$\infty$	$\infty$
z	7	1	0

**Chi phí tới**

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

**Chi phí tới**

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

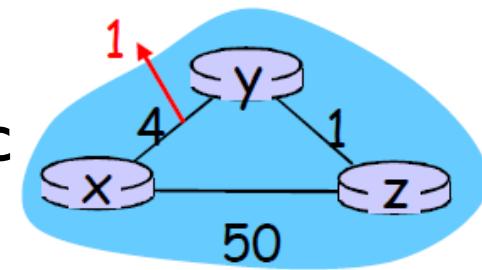


Thời gian

## 5.2 Các giải thuật định tuyến: Distance-Vector

### ■ Chi phí kết nối thay đổi:

- Nút kiểm tra thay đổi chi phí kết nối cục
- Cập nhật thông tin tìm đường, tính toán khoảng cách
- Nếu DV thay đổi, thông báo đến các lân cận
  - $t_0$ : y kiểm tra thay đổi chi phí kết nối, cập nhật lại DV của nó, và thông báo cho các lân cận.
  - $t_1$ : z nhận thông tin cập nhật từ y, cập nhật lại bảng của nó, tính toán chi phí thấp nhất mới tới x, và gửi DV của nó tới các lân cận.
  - $t_2$ : y nhận cập nhật của z, cập nhật lại bảng khoảng cách của nó. Chi phí thấp nhất của y không thay đổi, nên y không cần gửi thông báo tới z.



## 5.2 Các giải thuật định tuyến: Distance-Vector

---

- **Chi phí kết nối thay đổi:**

- Nút kiểm tra thay đổi chi phí cục bộ
- **Tin xấu truyền đi chậm** – vấn đề “đếm vô hạn”!
- 44 lần duyệt trước khi thuật toán ổn định: (xem thêm trong tài liệu)

- **poisoned reverse:**

- Nếu Z định tuyến Y đi thẳng tới X :
  - Z thông báo cho Y khoảng cách (của Z) tới X là vô hạn (để Y không đi tới X qua Z)
- Sẽ giải quyết triệt để vấn đề đếm vô hạn?

## 5.2 Các GTĐT: So sánh LS và DV

---

- Độ phức tạp thông báo
  - LS: với  $n$  nút,  $E$  liên kết,  $O(nE)$  thông báo được gửi đi
  - DV: chỉ trao đổi giữa các nút lân cận
    - Thời gian hội tụ thay đổi
- Tốc độ hội tụ
  - LS: Thuật toán  $O(n^2)$  yêu cầu  $O(nE)$  thông báo
    - Có thể có dao động
  - DV: Thời gian hội tụ thay đổi
    - Có thể lặp định tuyến
    - Vấn đề đếm vô hạn

## 5.2 Các GTĐT: So sánh LS và DV

---

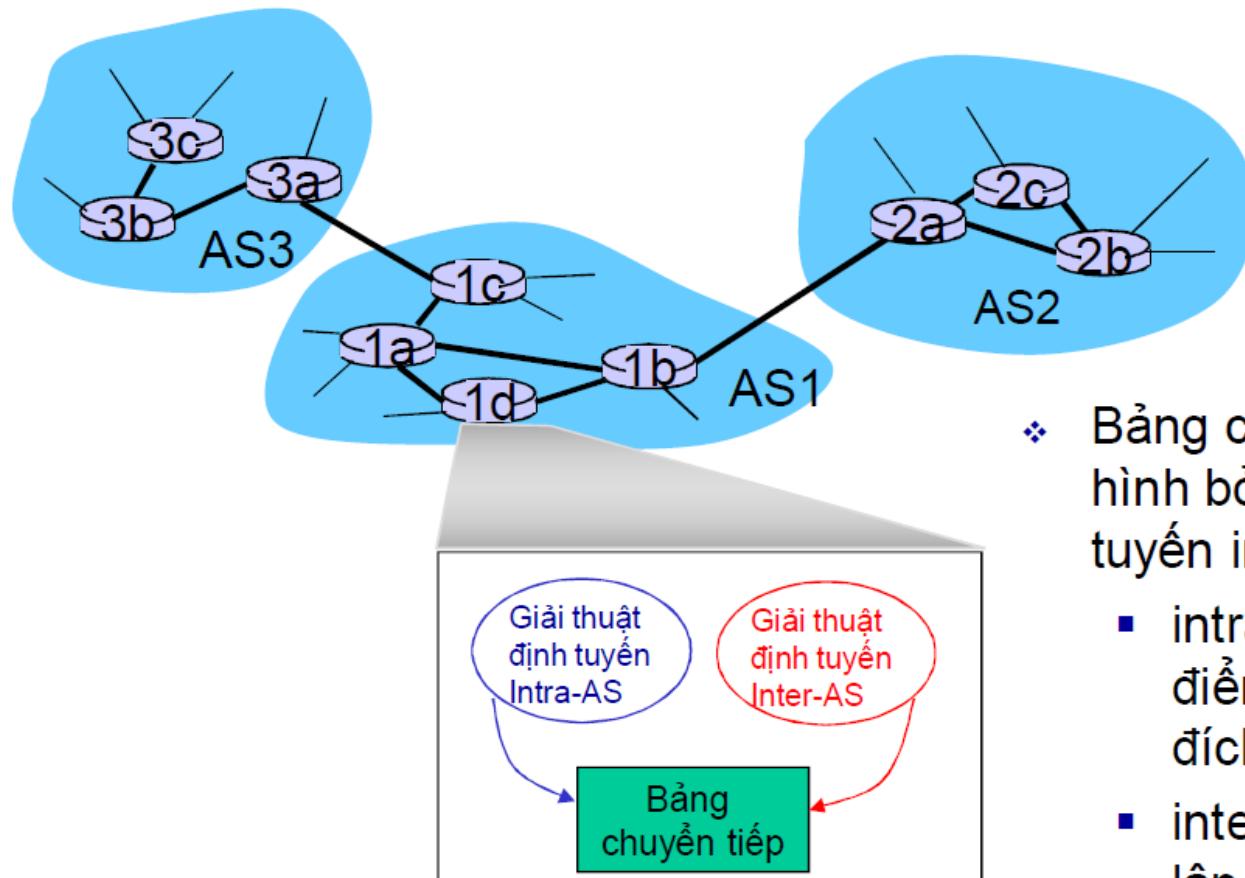
- **Sự linh hoạt:** điều gì sẽ xảy ra nếu router hoạt động sai chức năng?
  - LS:
    - Nút có thể thông báo chi phí **kết nối** không chính xác
    - Mỗi nút chỉ tính toán bảng riêng của nó
  - DV:
    - Nút có thể thông báo chi phí **đường đi** không chính xác
    - Mỗi bảng của nút được dùng bởi các nút khác
      - Lỗi lan truyền thông qua mạng

## 5.3 Định tuyến Intra-AS trên Internet

---

- Các router được tập hợp lại thành các vùng, “hệ thống tự trị” (**autonomous systems - AS**)
- Các router trong cùng AS sẽ chạy cùng giao thức định tuyến
  - Giao thức định tuyến “nội vùng-AS” (**intra-AS**)
  - Các router trong các AS khác nhau có thể chạy các giao thức định tuyến intra-AS khác nhau.
- **Gateway router:**
  - Tại “cạnh” của AS riêng của nó
  - Có liên kết tới router trong AS khác

## 5.3 Định tuyến Intra-AS trên Internet



- ❖ Bảng chuyển tiếp được cấu hình bởi cả giải thuật định tuyến intra- và inter-AS
  - intra-AS thiết lập các điểm đăng nhập cho các đích nội mạng
  - inter-AS & intra-AS thiết lập các điểm đăng nhập cho các đích ngoại mạng

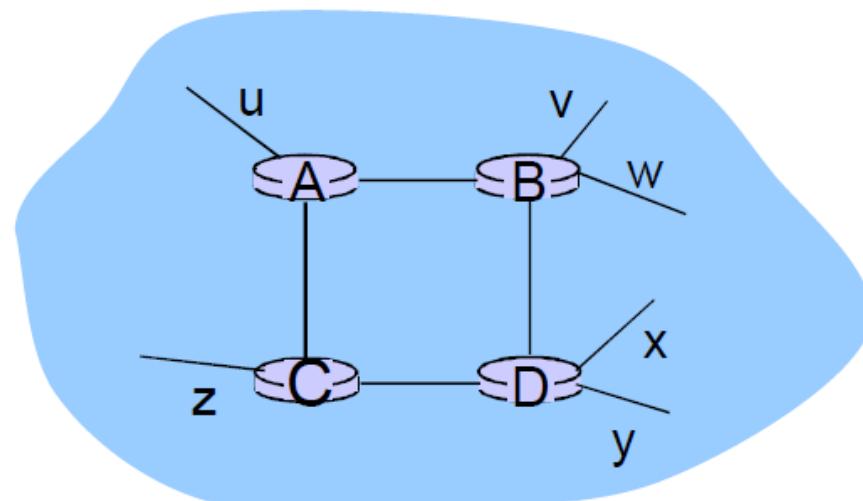
## 5.3 Định tuyến Intra-AS trên Internet

---

- Còn được gọi là các giao thức cổng nội mạng (**interior gateway protocols - IGP**)
- Các giao thức định tuyến intra-AS phổ biến nhất:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP/ EIGRP : Interior Gateway Routing Protocol (Cisco độc quyền)

## 5.3 Định tuyến Intra-AS trên Internet: RIP

- RIP: Routing Information Protocol
- Được công bố trong BSD-UNIX distribution năm 1982
- Giải thuật distance vector
  - Độ đo khoảng cách: số hop (lớn nhất = 15 hop), mỗi liên kết có chi phí là 1

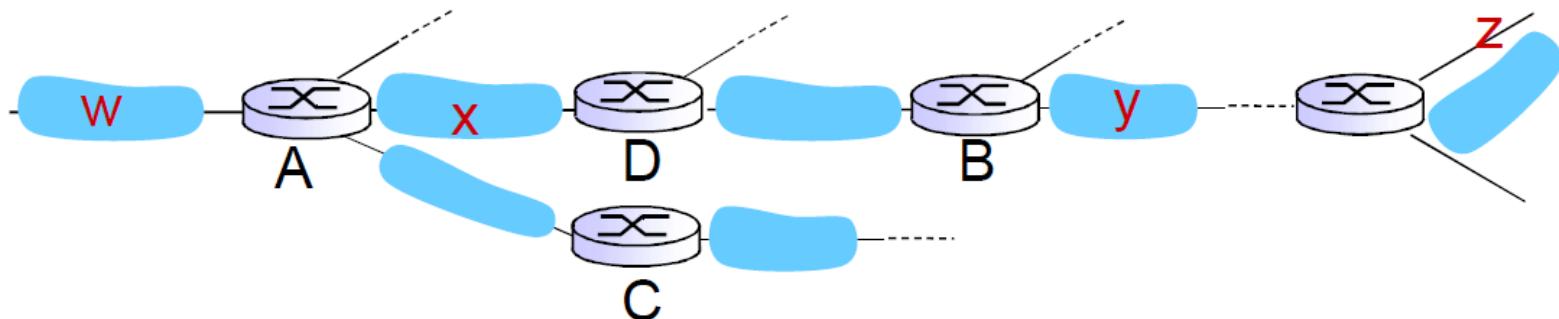


Từ router A đến các **subnet** đích:

<u>subnet</u>	<u>hop</u>
u	1
v	2
w	2
x	3
y	3
z	2

# 5.3 Định tuyến Intra-AS trên Internet: RIP

- RIP: ví dụ

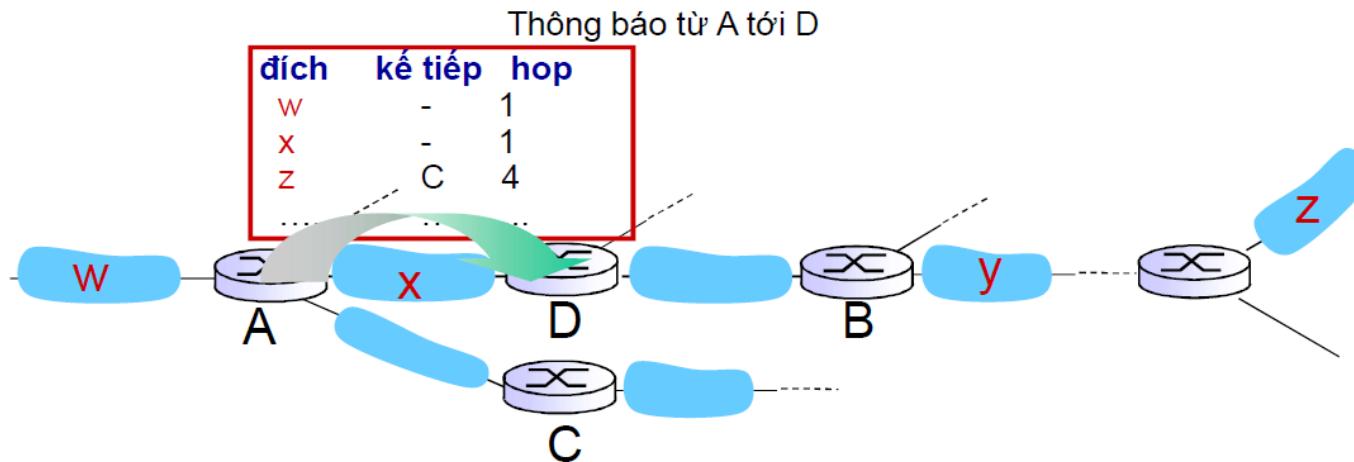


Bảng định tuyến trong router D

Subnet đích	Router kế tiếp	Số hop đến đích
W	A	2
y	B	2
Z	B	7
X	--	1
....	....	....

# 5.3 Định tuyến Intra-AS trên Internet: RIP

- RIP: ví dụ



Bảng định tuyến trong router D

Subnet đích	Router kế tiếp	Số hop đến đích
W	A	2
y	B	2
z	<del>B</del> A	<del>7</del> 5
x	--	1
....	....	....

## 5.3 Định tuyến Intra-AS trên Internet: RIP

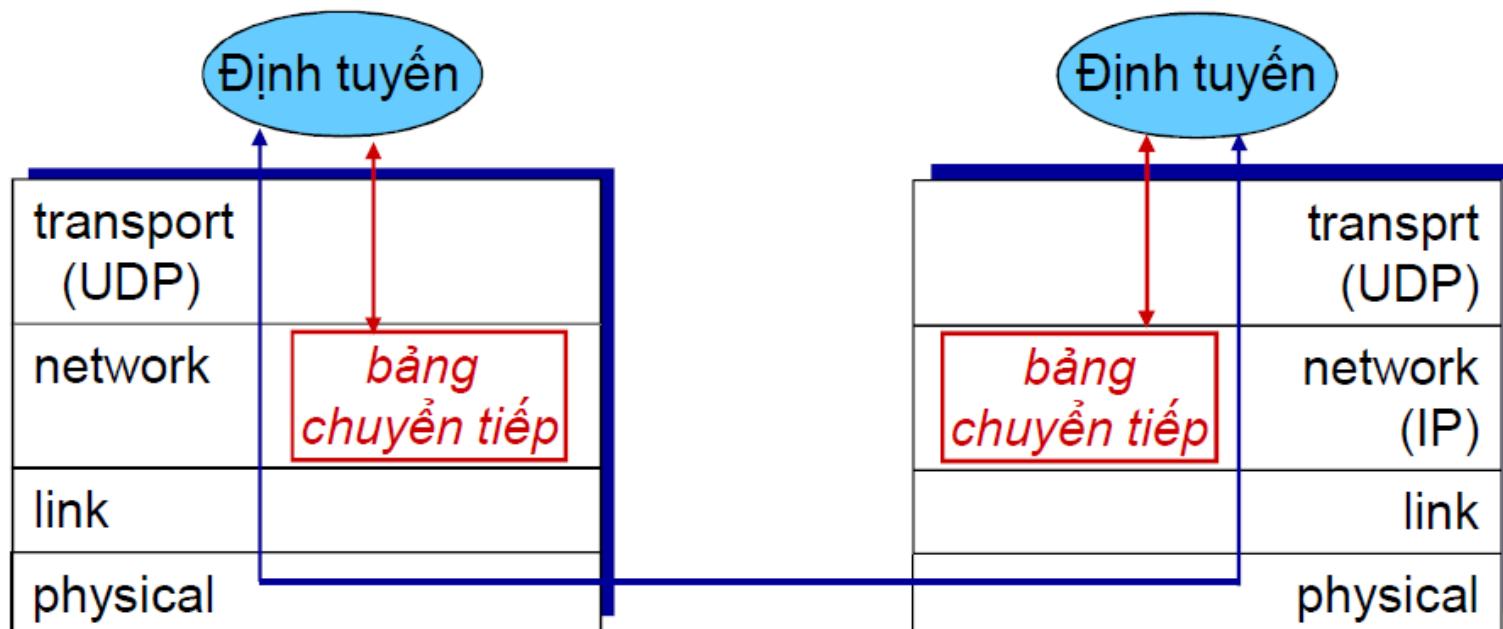
---

- RIP: Lỗi liên kết và khôi phục
  - Nếu không thấy có thông báo sau khoảng 180s thì lân cận/liên kết được coi là “đã chết”.
    - Các tuyến đường qua lân cận là không còn dùng được
    - Các thông báo mới được gửi tới các lân cận
    - Các lân cận tiếp tục gửi các thông báo mới (nếu các bảng bị thay đổi)
    - Thông báo lỗi liên kết lan truyền nhanh chóng (?) trên toàn bộ mạng
    - **Poison reverse** được dùng để ngăn chặn các vòng lặp ping-pong (khoảng cách vô hạn = 16 hop)

## 5.3 Định tuyến Intra-AS trên Internet: RIP

### ▪ Xử lý bảng RIP

- Các bảng định tuyến RIP được quản lý bởi tiến trình **tầng ứng dụng** được gọi là route-d (daemon)
- Các thông báo được gửi trong các gói tin UDP, lặp lại định kỳ



## 5.3 Định tuyến Intra-AS trên Internet: OSPF

---

- OSPF (Open Shortest Path First)
  - “Mở”: sẵn sàng công khai
  - Dùng giải thuật link state
    - Phân phối gói LS
    - Bản đồ cấu trúc mạng tại mỗi nút
    - Tính toán đường đi dùng giải thuật Dijkstra
  - Thông báo OSPF mang một điểm truy nhập vào mỗi lân cận
  - Các thông báo được phân phối đến **toàn bộ** AS (qua cơ chế flooding)
    - Các thông điệp OSPF được mang trực tiếp trên IP (chứ không phải là TCP hay UDP)
  - **Giao thức định tuyến IS-IS:** gần giống với OSPF

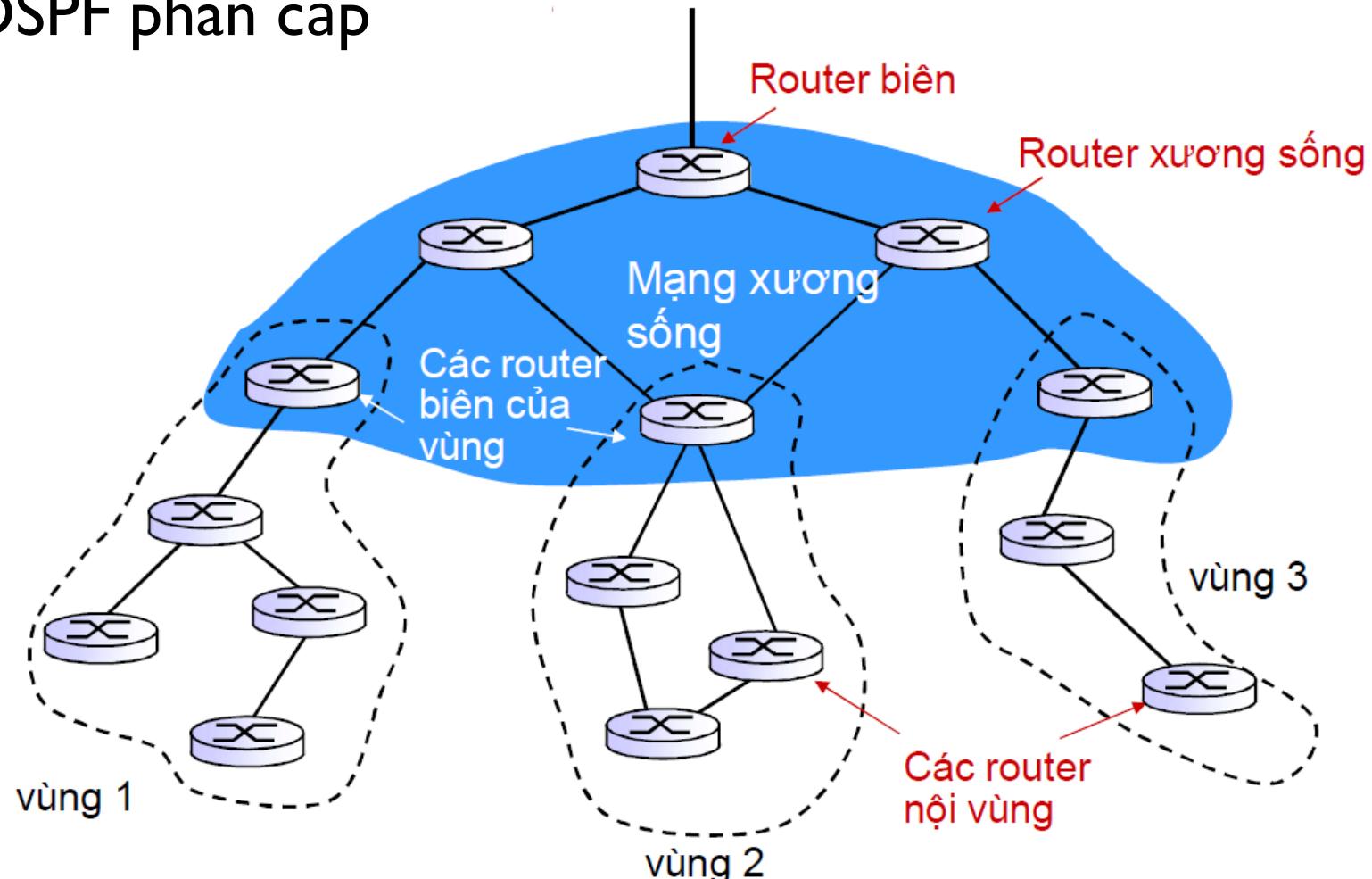
## 5.3 Định tuyến Intra-AS trên Internet: OSPF

---

- Các đặc tính “cải tiến” trong OSPF (không có trong RIP)
  - **Bảo mật:** Tất cả các thông điệp OSPF đều được chứng thực (để ngăn chặn những xâm nhập xấu)
  - Cho phép có **nhiều tuyến đường đi với cùng chi phí** (trong RIP chỉ có một)
  - Với mỗi liên kết, có nhiều độ đo chi phí cho các **TOS** khác nhau. (Ví dụ: chi phí liên kết vệ tinh được thiết lập “thấp” để đạt hiệu quả tốt; “cao” cho thời gian thực).
  - Hỗ trợ tích hợp uni- và **multicast**:
    - Multicast OSPF (MOSPF) dùng cơ sở dữ liệu cùng cấu trúc như OSPF
    - OSPF **phân cấp** trong các miền lớn.

## 5.3 Định tuyến Intra-AS trên Internet: OSPF

- OSPF phân cấp



## 5.3 Định tuyến Intra-AS trên Internet: OSPF

---

### ▪ OSPF phân cấp

- **Phân cấp 2 mức:** vùng cục bộ, vùng xương sống.
  - Chỉ dùng thông báo link-state bên trong vùng
  - Mỗi nút có cấu trúc vùng chi tiết; chỉ biết hướng (đường đi ngắn nhất) đến các mạng trong các vùng khác.
- **Các router biên của vùng:** “tổng hợp” khoảng cách đến các mạng trong vùng của nó, thông báo tới các router biên của vùng khác.
- **Các router xương sống:** chạy định tuyến OSPF hạn chế đến mạng xương sống
- **Các router biên:** kết nối tới các router biên của các AS khác.

## 5.3 Định tuyến Intra-AS trên Internet: IGRP/EIGRP

---

- IGRP/EIGRP: Interior Gateway Routing Protocol (Cisco độc quyền)
  - Distance Vector + Link States.
    - Là giao thức định tuyến theo vector khoảng cách nâng cao nhưng khi cập nhật và bảo trì thông tin lảng giềng và thông tin định tuyến thì nó làm việc giống như một giao thức định tuyến theo trạng thái đường liên kết.

## 5.3 Định tuyến Intra-AS trên Internet: IGRP/EIGRP

---

- IGRP/EIGRP: Ưu điểm

- Khả năng hội tụ nhanh: vì chúng sử dụng DUAL.
  - DUAL bảo đảm hoạt động không bị lặp vòng khi tính toán đường đi, cho phép mọi Router trong hệ thống mạng thực hiện đồng bộ cùng lúc khi có sự thay đổi xảy ra.

## 5.3 Định tuyến Intra-AS trên Internet: IGRP/EIGRP

---

### ■ IGRP/EIGRP: Ưu điểm

- Bảo tồn băng thông và sử dụng băng thông một cách hiệu quả
  - Chỉ gửi thông tin cập nhật một phần và giới hạn chứ không gửi toàn bộ bảng định tuyến. Điều này tương tự như hoạt động cập nhật của OSPF
    - Router EIGRP chỉ gửi thông tin cập nhật một phần cho Router nào cần thông tin đó chứ không gửi mọi Router khác trong vùng như OSPF.
  - Chính vì hoạt động cập nhật theo chu kỳ, các Router EIGRP giữ liên lạc với nhau bằng các gói hello rất nhỏ. Việc trao đổi các gói hello theo định kỳ không chiếm nhiều băng thông đường truyền.

## 5.3 Định tuyến Intra-AS trên Internet: IGRP/EIGRP

---

- IGRP/EIGRP: Ưu điểm

- Hỗ trợ VLSM (Variable Length Subnet Mask) và CIDR (Classless Inter Domain Routing).
- Không giống như IGRP, EIGRP có thể trao đổi thông tin ở các IP khác lớp mạng

## 5.3 Định tuyến Intra-AS trên Internet: IGRP/EIGRP

---

- IGRP/EIGRP: Ưu điểm

- Hỗ trợ IP, IPX, Apple talk

- Vì Talk nhờ có cấu trúc từng phần theo giao thức (PDMs – Protocol dependent modules).
    - EIGRP có thể phân phối thông tin của IPX, RIP để cải tiến hoạt động toàn diện. Trên thực tế, EIGRP có thể điều khiển giao thức này.
    - Router EIGRP nhận thông tin định tuyến và dịch vụ, chỉ cập nhật cho các Router khác khi thông tin trong bảng định tuyến thay đổi.

## 5.3 Định tuyến Intra-AS trên Internet: IGRP/EIGRP

---

- IGRP/EIGRP: Ưu điểm

- Chạy trực tiếp trên IP và protocol number là 88
- Load balancing trên tất cả các cost không bằng nhau
- Hỗ trợ tất cả các giao thức và cấu trúc dữ liệu ở layer 2
- Không dùng broadcast và dùng Multicast hoặc Unicast trong từng trường hợp cụ thể
- Hỗ trợ việc chứng thực
- Manual Summary trên bất kỳ interface nào

## 5.3 Định tuyến Intra-AS trên Internet: IGRP/EIGRP

---

- IGRP/EIGRP: Nhược điểm
  - Là giao thức độc quyền của Cisco nên nó chỉ chạy trên thiết bị của cisco.

# 5.4 Định tuyến giữa các ISP: BGP

---

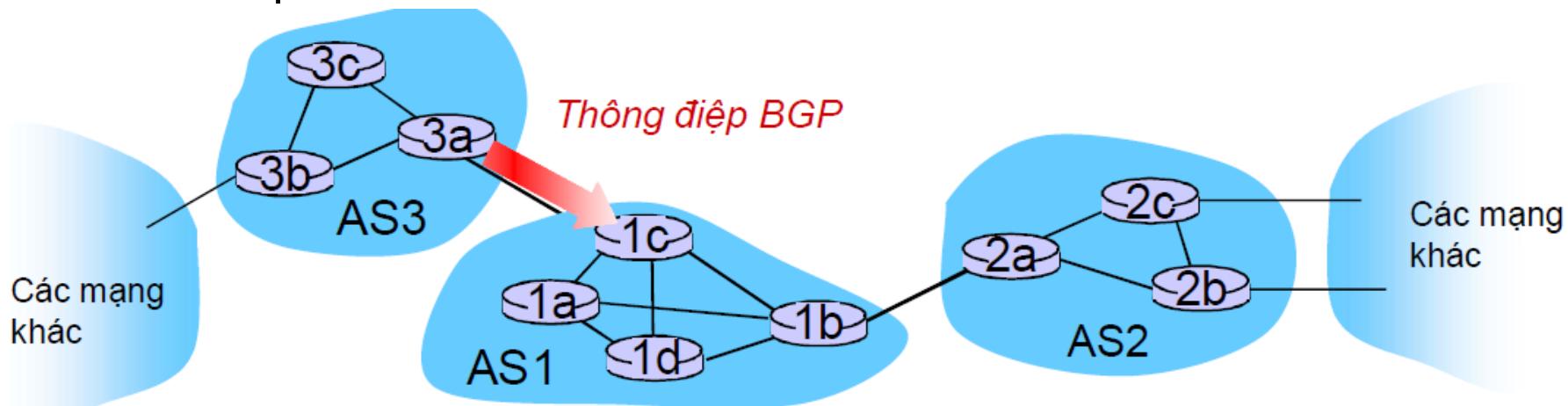
## ■ Định tuyến inter-AS trên Internet

- **BGP (Border Gateway Protocol)**: Giao thức định tuyến liên miền thực tế
  - “gắn kết mọi người lại với nhau trên Internet”
- BGP cung cấp cho mỗi AS:
  - **eBGP**: lấy thông tin đi đến subnet từ các AS lân cận.
  - **iBGP**: lan truyền thông tin đến tất cả các router bên trong AS.
  - Xác định đường đi “tốt” tới các mạng khác dựa trên thông tin đường đi và chính sách
- Cho phép subnet thông báo sự tồn tại của nó đến phần còn lại của Internet.

## 5.4 Định tuyến giữa các ISP: BGP

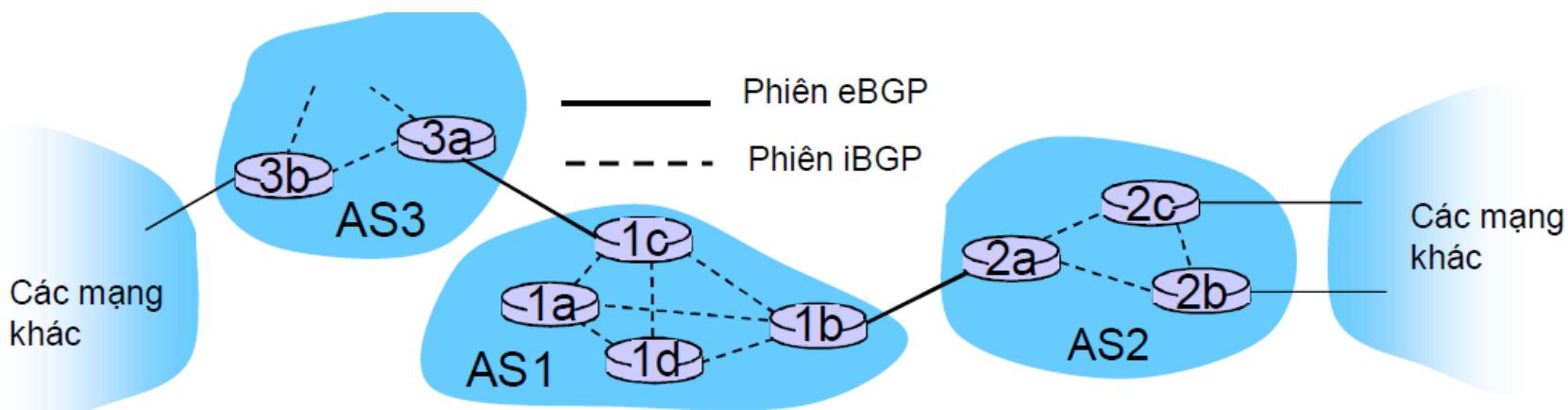
### ■ Cơ sở của BGP

- **Phiên BGP:** Hai router BGP (“các peer”) trao đổi các thông điệp BGP:
  - Thông báo **đường đi** tới các tiền tố (prefix) mạng đích khác nhau (giao thức “path vector” (véc-tơ đường))
  - Được trao đổi qua các kết nối TCP bán bền vững
- Khi AS3 thông báo một prefix đến AS1:
  - AS3 **hứa hẹn** nó sẽ chuyển tiếp các datagram hướng tới prefix đó



## 5.4 Định tuyến giữa các ISP: BGP

- Các cơ sở của BGP: phân phối thông tin đường đi
  - Dùng phiên eBGP giữa 3a và 1c, AS3 gửi thông tin đường đi (prefix) cho AS1.
    - Tiếp theo, 1c có thể dùng iBGP để phân phối thông tin prefix mới cho tất cả các router trong AS1
    - Sau đó, 1b có thể thông báo thông tin đường đi mới tới AS2 qua phiên eBGP từ 1b-đến-2a.
  - Khi router học được prefix mới, nó sẽ tạo ra điểm truy nhập cho prefix trong bảng chuyển tiếp của nó.



## 5.4 Định tuyến giữa các ISP: BGP

---

- Các thuộc tính đường và định tuyến BGP
  - Prefix được thông báo chứa các thuộc tính BGP
    - Prefix + các thuộc tính = “định tuyến”
  - Hai thuộc tính quan trọng:
    - **AS-PATH:** chứa các AS qua đó thông báo prefix nào được truyền. Ví dụ: AS 67, AS 17
    - **NEXT-HOP:** xác định router AS nội vùng nào là AS kế tiếp.  
(Có thể có nhiều liên kết từ AS hiện tại tới AS kế tiếp).
  - Gateway router nhận thông báo định tuyến bằng cách dùng **import policy (chính sách nhập)** để chấp nhận/từ chối
    - Ví dụ: không bao giờ định tuyến qua AS x
    - Định tuyến **dựa trên chính sách**.

## 5.4 Định tuyến giữa các ISP: BGP

---

### ■ Chọn định tuyến BGP

- Router có thể học được nhiều đường đi đến AS đích, và việc chọn tuyến đường được dựa trên:
  - 1. Thuộc tính giá trị ưu tiên cục bộ: quyết định chính sách
  - 2. AS-PATH ngắn nhất
  - 3. Router NEXT-HOP gần nhất: định tuyến hot potato
  - 4. Tiêu chuẩn bổ sung

## 5.4 Định tuyến giữa các ISP: BGP

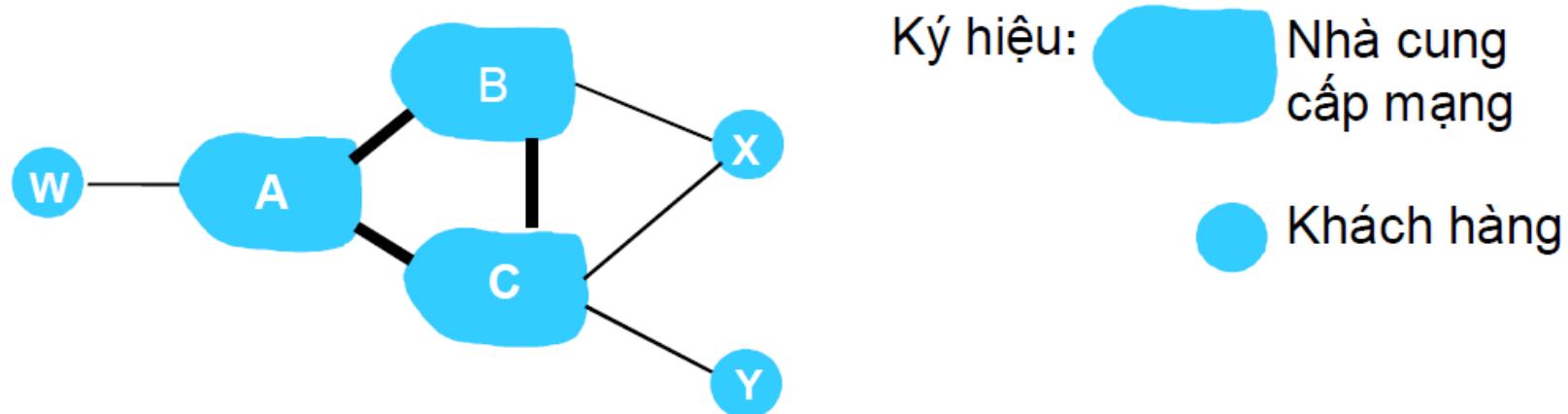
---

### ■ Các thông điệp BGP

- Các thông điệp BGP được trao đổi giữa các peer qua kết nối TCP
- Các thông điệp BGP:
  - **OPEN**: Mở kết nối TCP tới peer và xác thực bên gửi
  - **UPDATE**: thông báo đường đi mới (hoặc xóa bỏ đường cũ)
  - **KEEPALIVE**: giữ kết nối tồn tại khi UPDATES thiếu; cũng có thể yêu cầu ACKs OPEN
  - **NOTIFICATION**: báo cáo

## 5.4 Định tuyến giữa các ISP: BGP

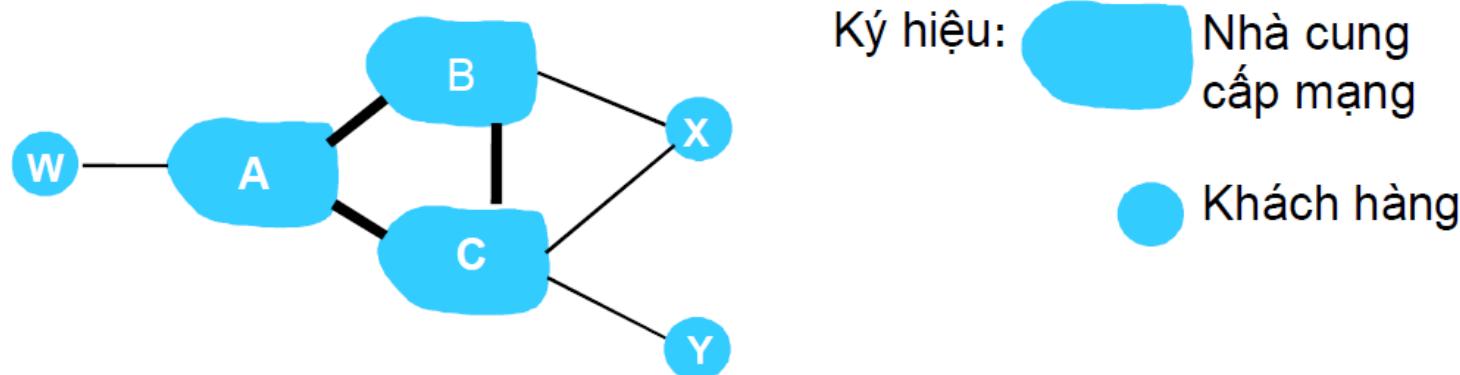
### ■ Chính sách định tuyến BGP



- ❖ A, B, C là *các nhà cung cấp mạng*
- ❖ X, W, Y là khách hàng (của nhà cung cấp mạng)
- ❖ X là *dual-homed*: được gắn vào hai mạng
  - X không muốn định tuyến từ B đến C qua X
  - ..do vậy, X sẽ không thông báo tới B về đường đi đến C

## 5.4 Định tuyến giữa các ISP: BGP

### ■ Chính sách định tuyến BGP



- ❖ A thông báo đường đi AW đến B
- ❖ B thông báo đường đi BAW đến X
- ❖ B sẽ thông báo đường đi BAW đến C?
  - Không! B không nhận “thu thập” cho định tuyến CBAW vì W và C đều không phải là khách hàng của B
  - B muốn buộc C phải định tuyến tới w qua A
  - B *chỉ* muốn định tuyến từ/tới khách hàng của nó!

## 5.4 Định tuyến giữa các ISP: BGP

---

- Tại sao định tuyến Intra-, Inter-AS khác nhau?
  - Chính sách:
    - Inter-AS: nhà quản trị muốn điều hành định tuyến lưu lượng và ai định tuyến qua mạng của họ.
    - Intra-AS: Quản trị riêng, vì vậy không cần các quyết định chính sách
  - Quy mô:
    - Định tuyến phân cấp tiết kiệm kích thước bảng, giảm lưu lượng cập nhật
  - Hiệu năng:
    - Intra-AS: có thể tập trung vào hiệu năng
    - Inter-AS: chính sách quan trọng hơn hiệu suất

## 5.5 Mặt phẳng điều khiển SDN

---

- Đặc điểm chính của kiến trúc SDN
  - Chuyển tiếp dựa trên cơ sở luồng
    - Các luật chuyển tiếp gói tin được chỉ ra trong một **flow table** của router.
    - Các mục trong flow table được tính toán, quản lý và cài đặt bởi mặt phẳng điều khiển SDN

## 5.5 Mặt phẳng điều khiển SDN

---

- Đặc điểm chính của kiến trúc SDN
  - Tách mặt phẳng dữ liệu và mặt phẳng điều khiển
    - Mặt phẳng dữ liệu gồm các router trong mạng thực hiện các luật “match and action” trong các flow table.
    - Mặt phẳng kiểm soát gồm các server và phần mềm xác định và quản lý các flow table cả các router.

## 5.5 Mặt phẳng điều khiển SDN

---

- Đặc điểm chính của kiến trúc SDN
  - Các chức năng điều khiển mạng
    - SDN controller và các network control applications
  - Mạng có thể lập trình
    - Thông qua network control applications

## 5.5 Mặt phẳng điều khiển SDN

---

### ■ SDN controller

- Chức năng được tổ chức thành 3 tầng
  - Tầng truyền thông:
    - Truyền thông giữa SDN controller và controlled network devices (SDN-enabled switch, host, or other device).
    - Cần có giao thức điều khiển việc truyền thông này: OpenFlow là giao thức phổ biến nhất trong các SDN controllers
  - Tầng quản lý trạng thái toàn mạng
  - Giao diện cho tầng ứng dụng điều khiển mạng

## 5.5 Mặt phẳng điều khiển SDN

---

### ■ Giao thức OpenFlow

- Truyền tin TCP, cổng 6653
- Các thông điệp quan trọng truyền từ bộ điều khiển tới thiết bị được điều khiển
  - **Cấu hình:** Thông điệp cho phép truy vấn và thiết lập các tham số cấu hình của thiết bị (router)
  - **Trạng thái-thay đổi:** thông điệp này được sử dụng bởi một bộ điều khiển để thêm, xóa hay thay đổi các mục trong flow table và thiết lập các thuộc tính của cổng chuyển mạch.

## 5.5 Mặt phẳng điều khiển SDN

---

### ■ Giao thức OpenFlow

- Các thông điệp quan trọng truyền từ bộ điều khiển tới thiết bị được điều khiển
  - **Trạng thái-đọc:** thông điệp này được sử dụng bởi bộ điều khiển để thu thập thống kê và các giá trị bộ đếm từ flow table và port của router.
  - **Gửi-gói tin:** thông điệp này được sử dụng bởi bộ điều khiển để gửi một gói tin cụ thể ra khỏi một cổng cụ thể ở router được điều khiển. Thông điệp này bên thân nó cũng chứa gói tin được gửi đi trong payload của nó.

## 5.5 Mặt phẳng điều khiển SDN

---

### ■ Giao thức OpenFlow

- Các thông điệp từ router được điều khiển bởi SDN và bộ điều khiển gồm:
  - **Luồng-bị xóa:** thông điệp này thông báo cho bộ điều khiển rằng một mục trong flow table đã bị xóa, ví dụ do hết thời gian chờ hay do nhận được thông điệp trạng thái-thay đổi
  - **Trạng thái-cổng:** thông điệp này được sử dụng bởi router để thông báo cho bộ điều khiển về một thay đổi trong trạng thái cổng.

## 5.5 Mặt phẳng điều khiển SDN

---

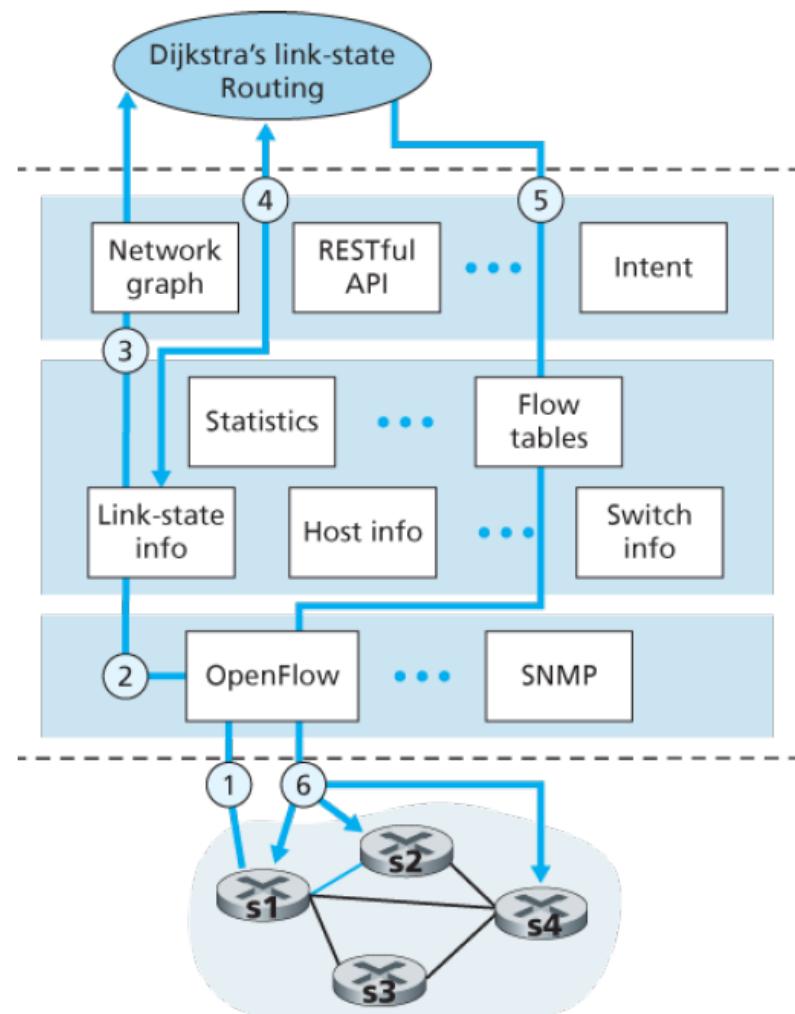
### ■ Giao thức OpenFlow

- Các thông điệp từ router được điều khiển bởi SDN và bộ điều khiển gồm:
  - **Gói tin đến:** thông điệp được dùng để gửi các gói tin tới bộ điều khiển
    - Gói tin đến một cổng của router và không giống với bất kỳ mục nào trong flow table sẽ được gửi tới bộ điều khiển để xử lý sau này.
    - Các gói giống cũng có thể được gửi tới bộ điều khiển như là một hành động được thực hiện khi khớp.

## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

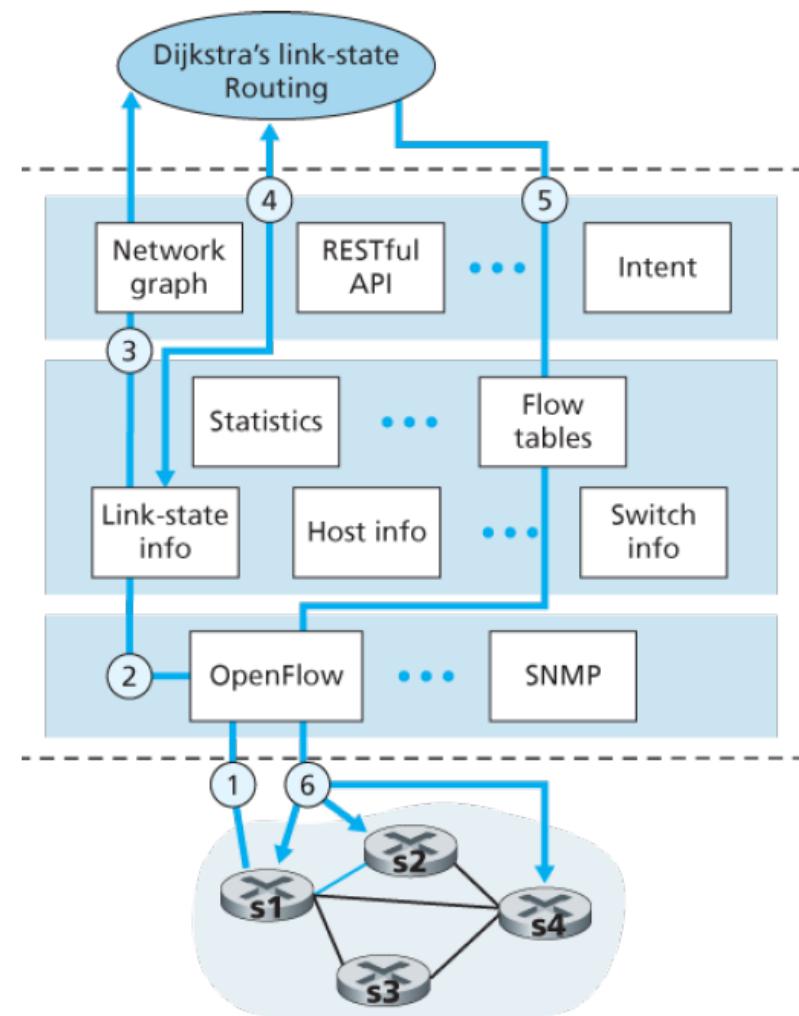
- Sử dụng giải thuật Dijkstra để tìm đường đi ngắn nhất
- Giải thuật Dijkstra được thực thi như một ứng dụng riêng biệt, ngoài router
- Router gửi các cập nhật liên kết tới bộ điều khiển SDN mà không gửi cho nhau.



## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

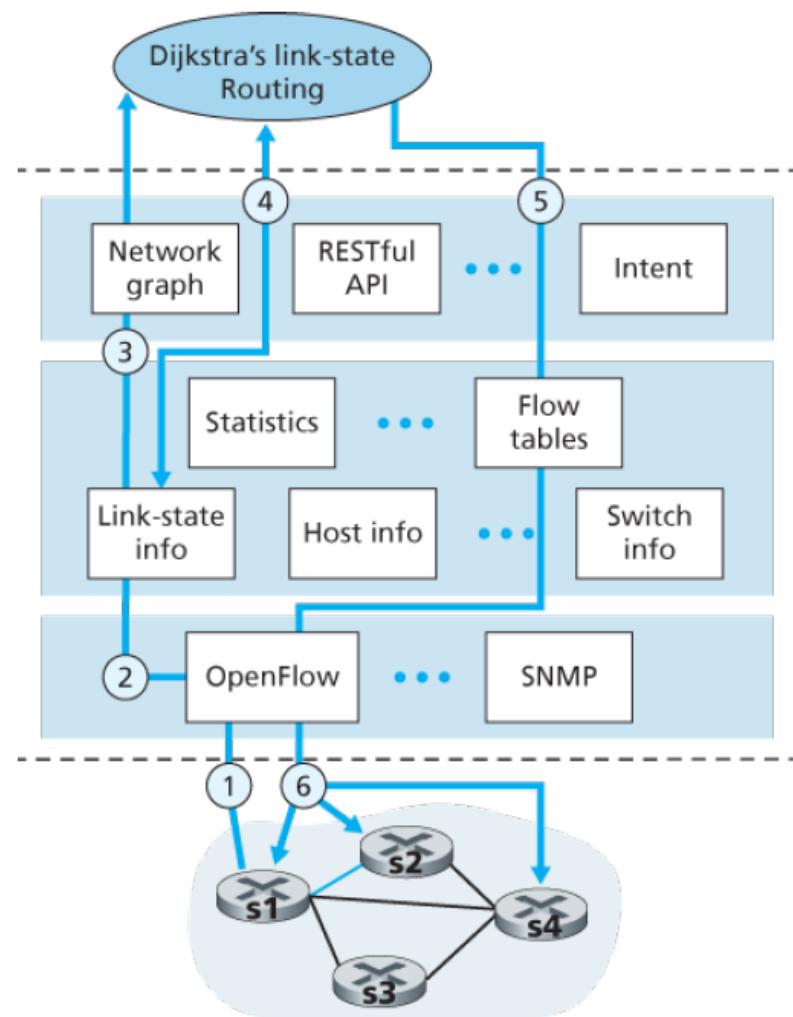
- Giả sử link giữa s1 và s2 bị hỏng, và thực hiện định tuyến đường ngắn nhất => hậu quả các nguyên tắc chuyển tiếp luồng đi và đến tại s1, s3, s4 bị ảnh hưởng, nhưng hoạt động của s2 không thay đổi.



## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

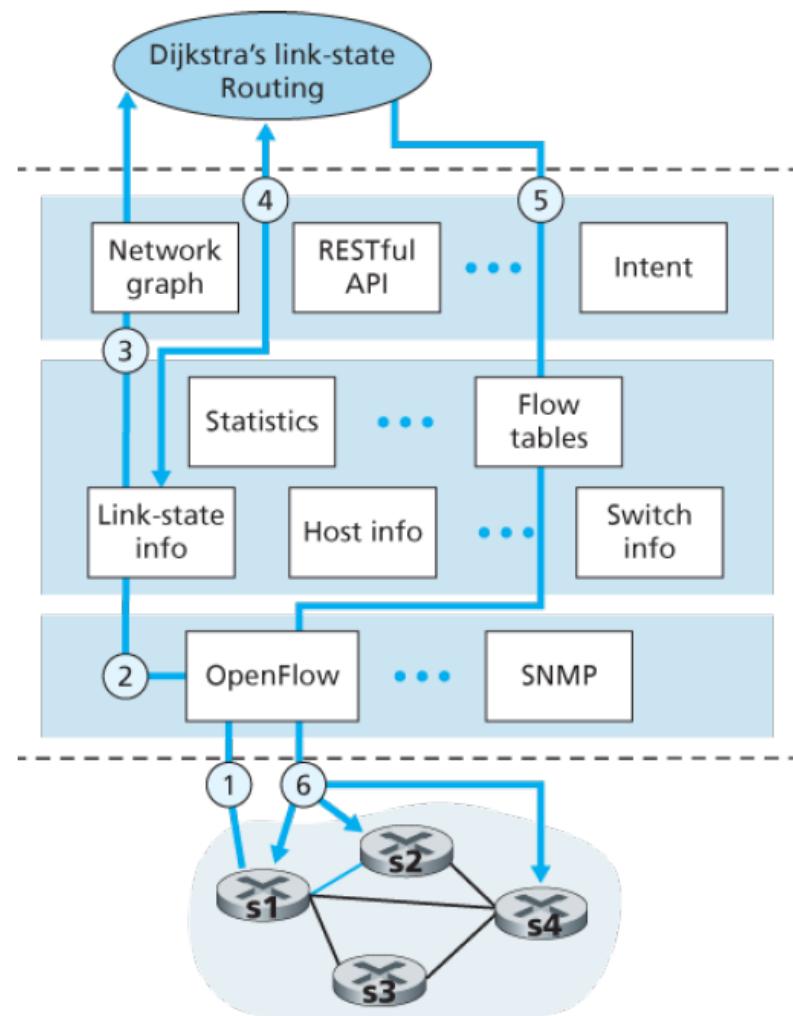
- Giả sử dùng OpenFlow là giao thức tầng truyền thông và mặt phẳng điều khiển không thực hiện chức năng nào khác ngoài chức năng định tuyến trạng thái liên kết



## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

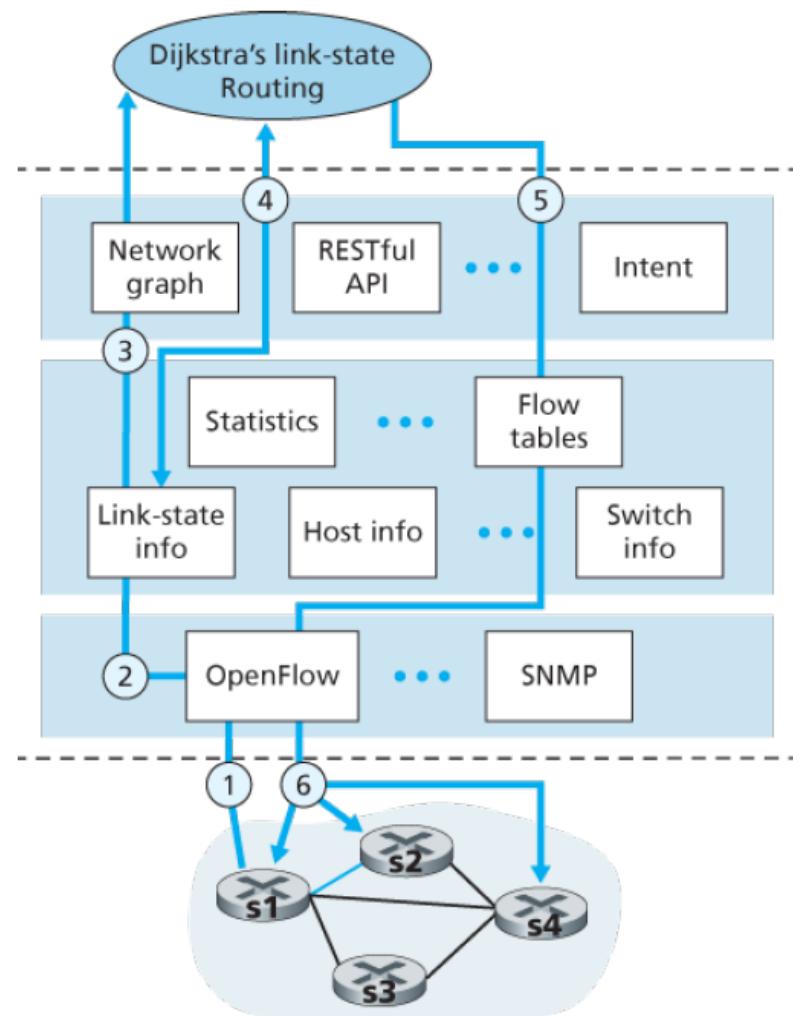
I. S1 gặp lỗi liên kết giữa chính nó và s2, thông báo cho bộ điều khiển SDN về thay đổi link-state này sử dụng thông điệp **trạng thái-cổng** OpenFlow



## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

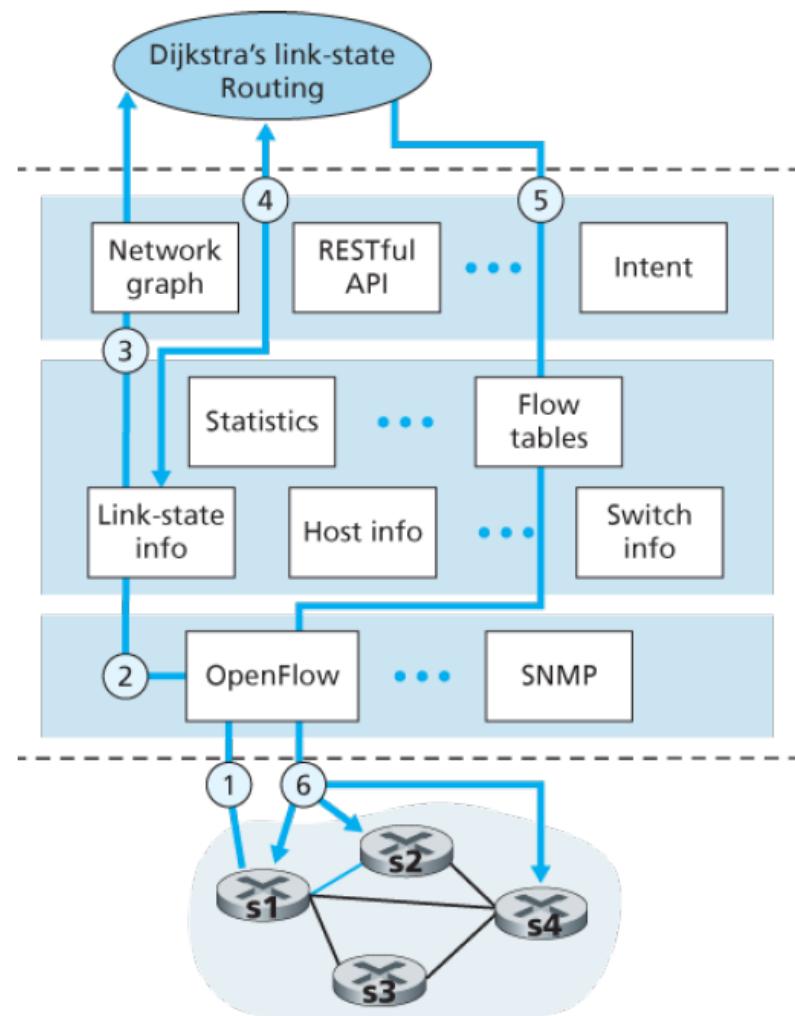
2. Bộ điều khiển SDN nhận được thông điệp OpenFlow báo thay đổi link-state, và thông báo cho trình quản lý link-state cập nhật cơ sở dữ liệu link-state.



## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

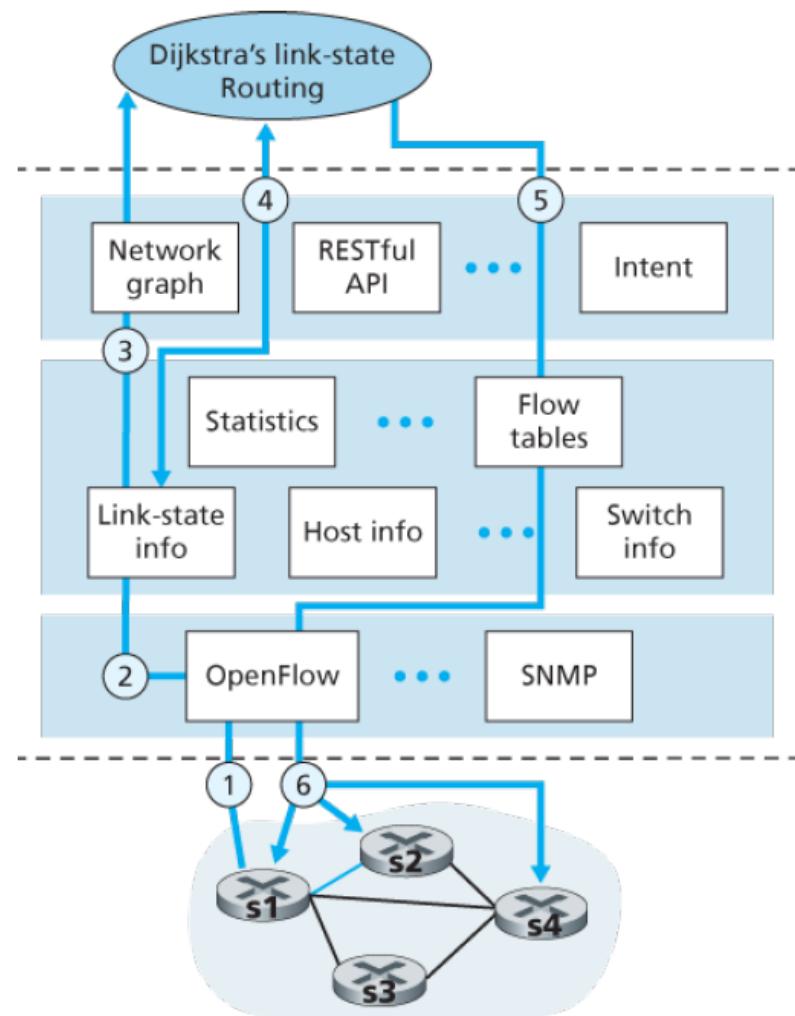
3. Network-control application (thực hiện định tuyến link-state Dijkstra) được thông báo khi trạng thái liên kết thay đổi) nhận được thông báo về thay đổi link-state



## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

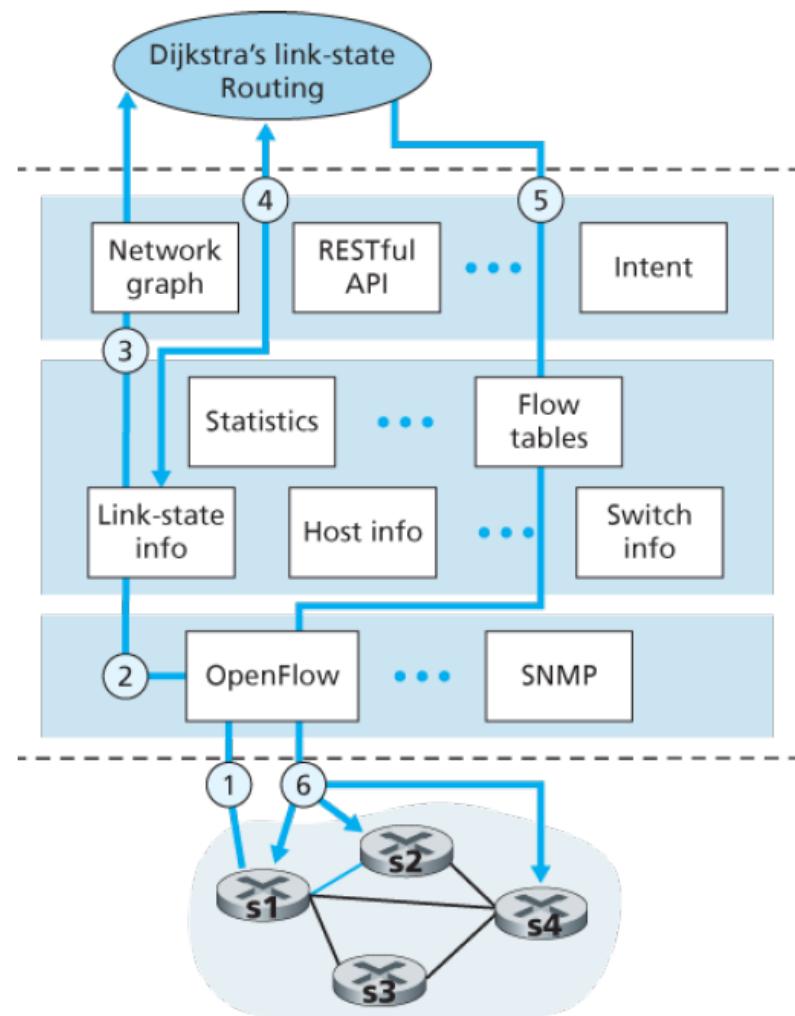
4. Ứng dụng định tuyến link-state tương tác với trình quản lý link-state để nhận trạng thái liên kết đã được cập nhật; nó có thể tham khảo các thành phần khác trong tầng quản lý trạng thái. Sau đó nó tính toán các đường đi mới với chi phí thấp nhất.



## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

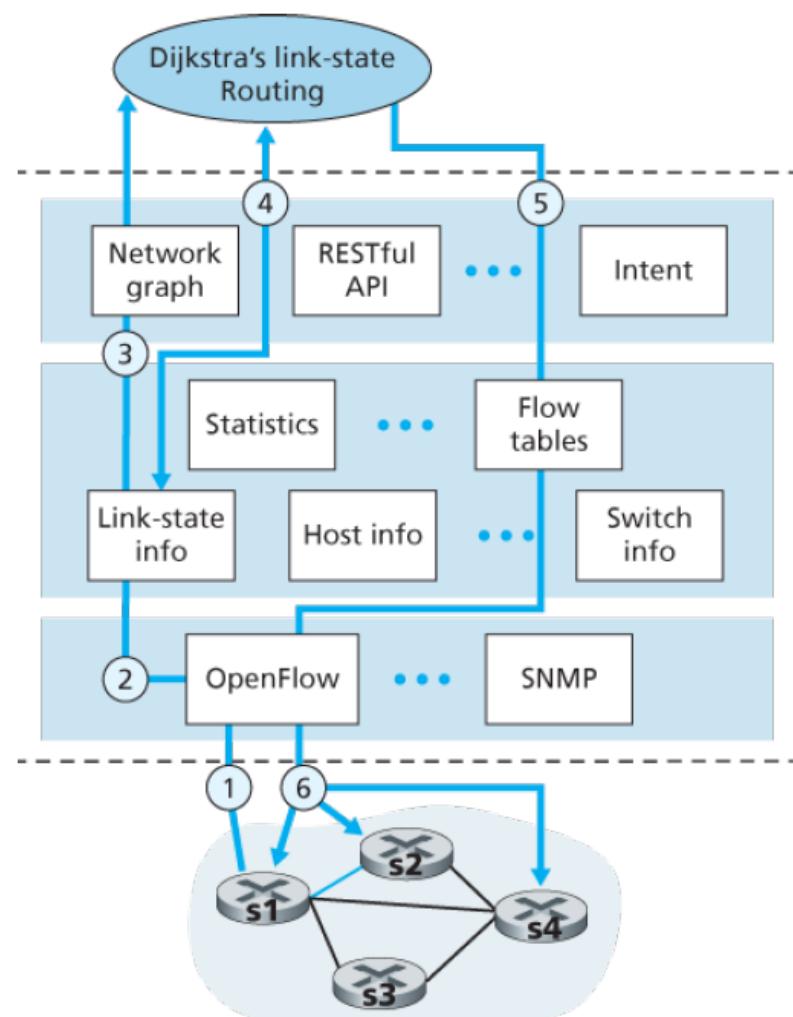
5. Ứng dụng định tuyến link-state sau đó tương tác với trình quản lý flow table xác định các flow table được cập nhật



## 5.5 Mặt phẳng điều khiển SDN

- Tương tác giữa mặt phẳng điều khiển và mặt phẳng dữ liệu: Ví dụ

6. Trình quản lý flow table sau đó sử dụng giao thức OpenFlow để cập nhật các mục trong flow table ở các router bị ảnh hưởng – s1 (sẽ định tuyến gói tin tới đích s2 qua s4), s2 (bắt đầu nhận các gói tin từ s1 qua router trung gian s4), và s4 (phải chuyển tiếp các gói tin từ s1 tới đích s2)



## 5.6 ICMP The Internet Control Message Protocol

- Được sử dụng bởi các host & các router để truyền thông tin tầng mạng
  - Báo cáo lỗi: không tìm được host, mạng, cổng, giao thức
  - Phản hồi yêu cầu/đáp ứng (được dùng bởi ping)
- “Ở phía trên” trong tầng mạng:
  - Các thông điệp ICMP được mang trong các IP datagram
- **Thông điệp ICMP:** type, code và 8 byte đầu tiên của IP datagram mô tả nguyên nhân lỗi

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

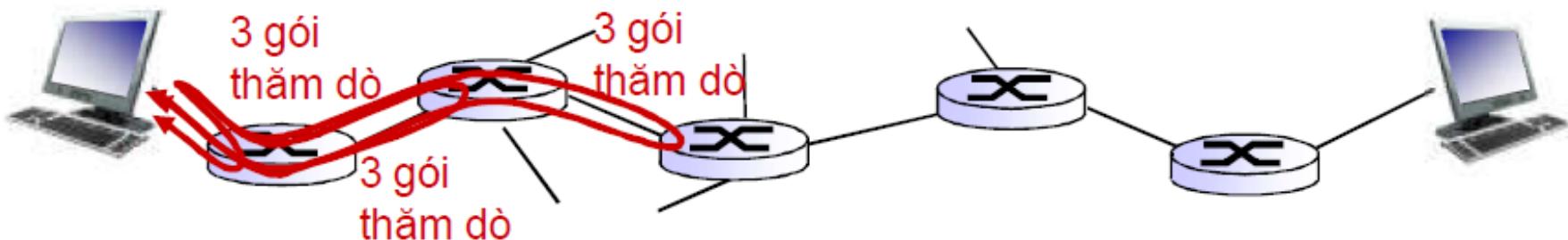
## 5.6 ICMP The Internet Control Message Protocol

### ■ Traceroute và ICMP

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

## 5.6 ICMP The Internet Control Message Protocol

- Traceroute và ICMP
  - Khi thông điệp ICMP đến, nguồn tính toán các RTT
- Điều kiện dừng:
  - UDP segment cuối cùng đến được host đích.
  - Đích trả lại thông điệp ICMP “port unreachable” (type 3, code 3) → cổng không có
  - Nguồn dừng lại



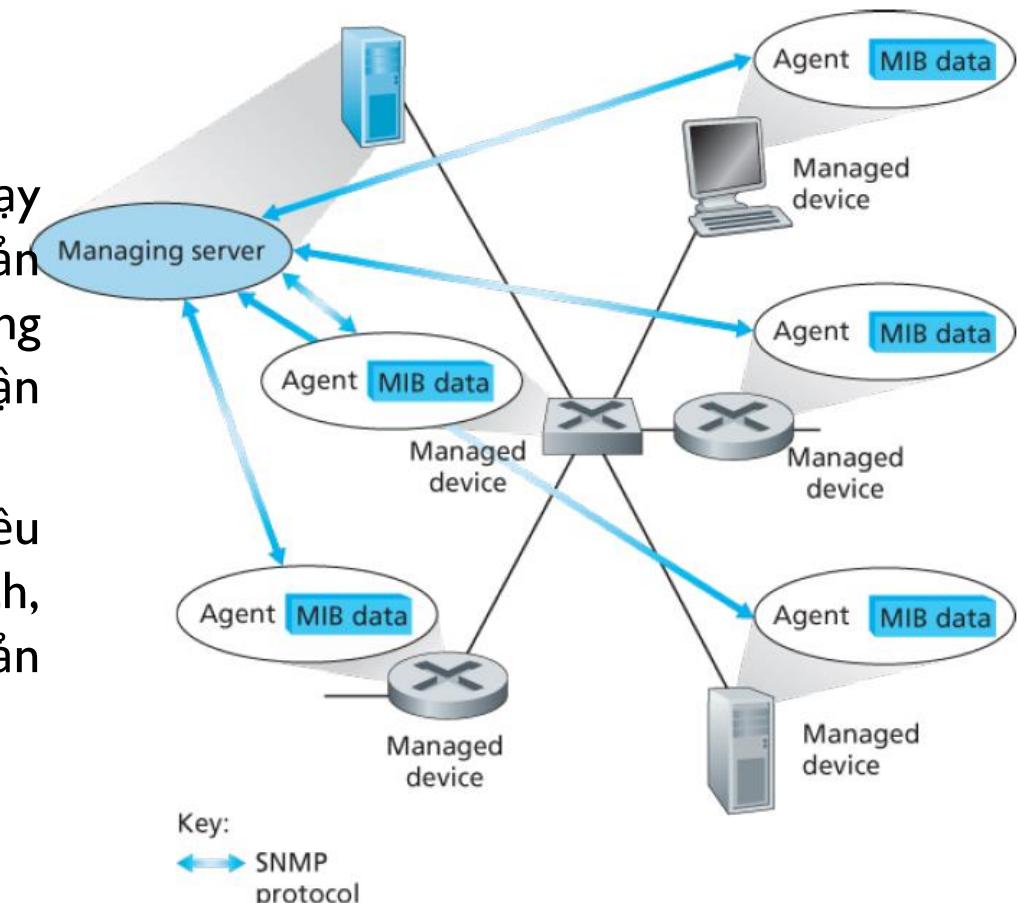
# 5.7: Quản lý mạng và SNMP

- Các thành phần chính của quản lý mạng

- Server quản lý

- Là một ứng dụng chạy trong một server quản lý mạng tập trung trong trung tâm vận hành mạng.

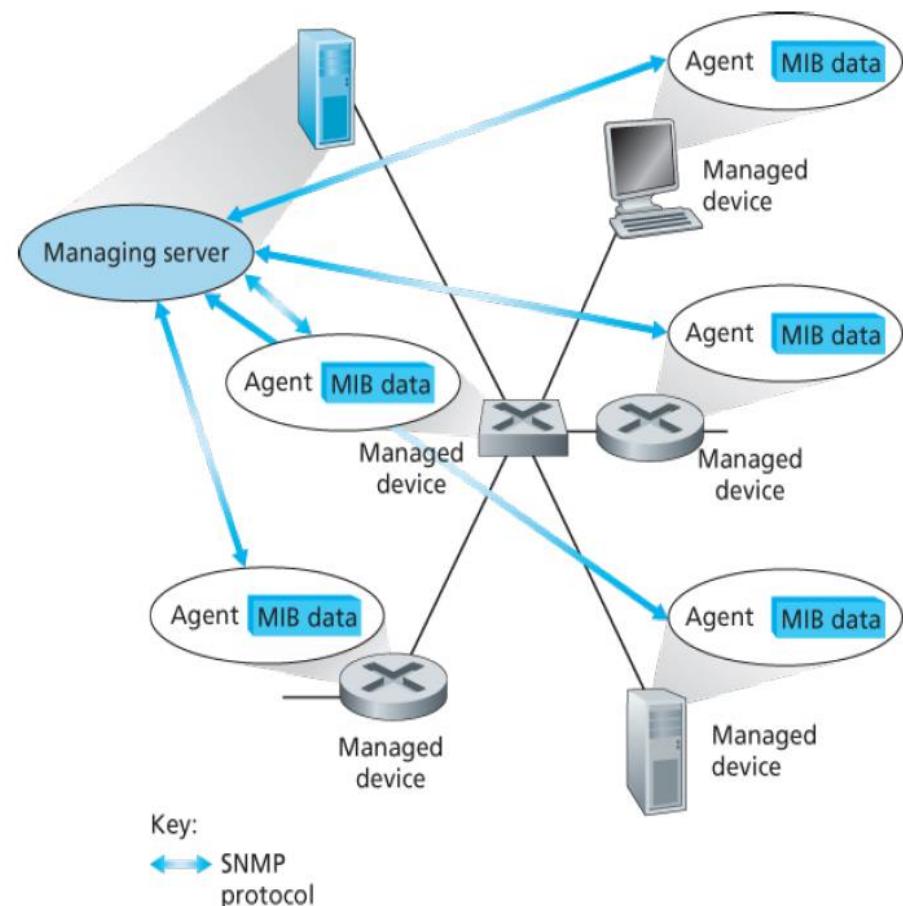
- Server quản lý điều khiển, xử lý, phân tích, hiển thị thông tin quản lý mạng



# 5.7: Quản lý mạng và SNMP

## ■ Các thành phần chính của quản lý mạng

- Thiết bị được quản lý
  - Là thiết bị mạng (gồm cả phần mềm của nó)
    - Host, router, switch, middlebox, modem, thermometer...
  - Trong thiết bị mạng được quản lý có thể gồm nhiều đối tượng được quản lý
    - NIC (trong host hay router)
    - Các tham số cấu hình (giao thức định tuyến OSPF...)



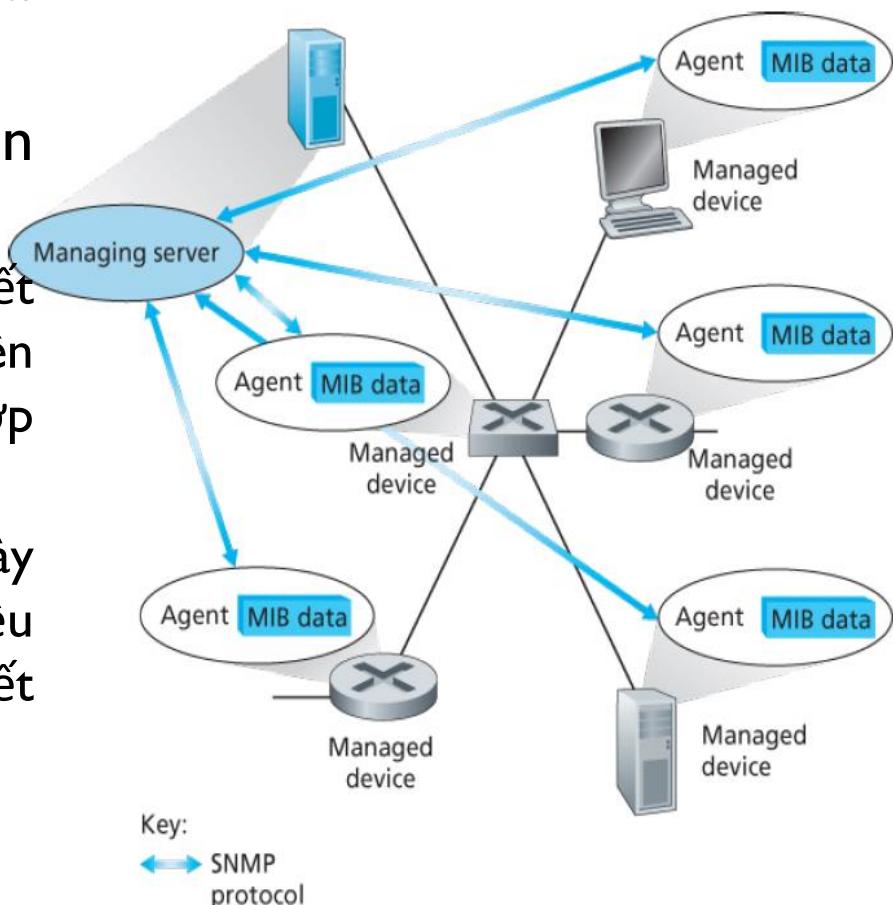
# 5.7: Quản lý mạng và SNMP

## ■ Các thành phần chính của quản lý mạng

- MIB (Management Information Base) data

- Mỗi đối tượng trong một thiết bị được quản lý có thông tin liên quan đến nó được tập hợp trong MIB

- Giá trị của những thông tin này là có sẵn cho (và trong nhiều trường hợp có thể được thiết lập bởi) máy chủ quản lý.



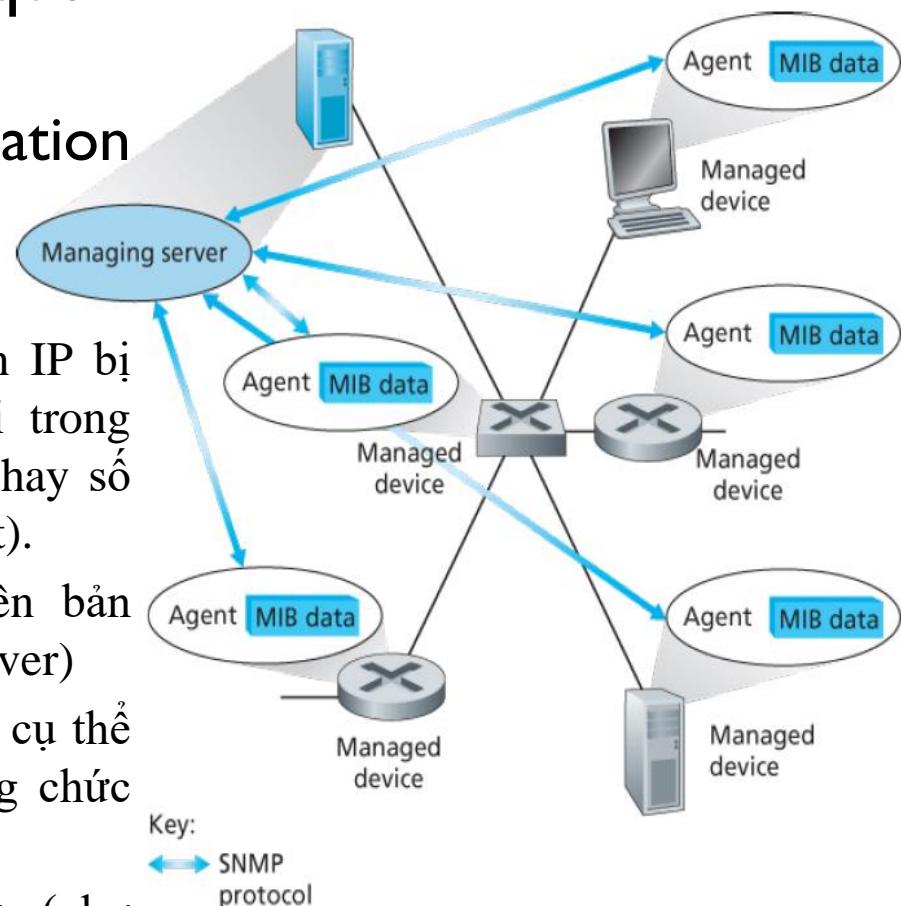
# 5.7: Quản lý mạng và SNMP

## ■ Các thành phần chính của quản lý mạng

- MIB (Management Information Base) data

- Một đối tượng MIB có thể là

- Một bộ đếm (như số gói tin IP bị loại bỏ ở router do các lỗi trong phần tiêu đề của gói tin IP, hay số gói tin UDP nhận được ở host).
    - Thông tin mô tả (như phiên bản phần mềm chạy trên DNS server)
    - Thông tin trạng thái (thiết bị cụ thể nào đó đang hoạt động đúng chức năng)
    - Thông tin cụ thể về giao thức (như đường định tuyến tới đích )



# 5.7: Quản lý mạng và SNMP

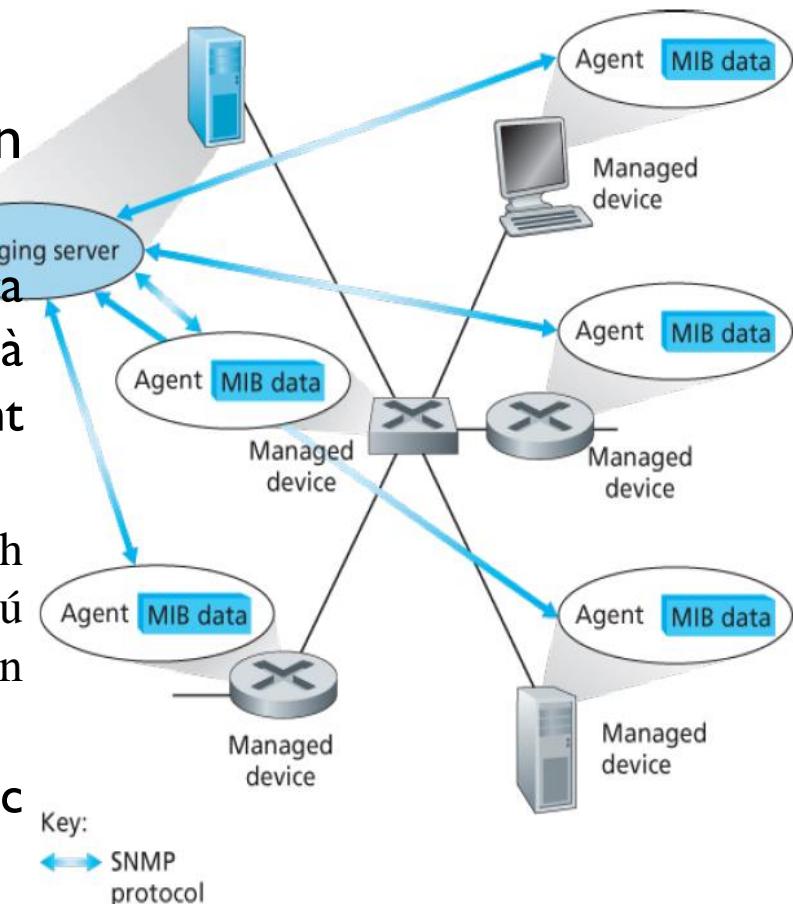
## ■ Các thành phần chính của quản lý mạng

- MIB (Management Information Base) data

- Các đối tượng MIB được chỉ ra trong một ngôn mô tả dữ liệu gọi là SMI (Structure of Management Information)

- Một ngôn ngữ định nghĩa chính thức được sử dụng để đảm bảo cú pháp và ngữ nghĩa của dữ liệu quản lý mạng được xác định rõ ràng.

- Các đối tượng MIB liên quan được tập hợp thành các module MIB.

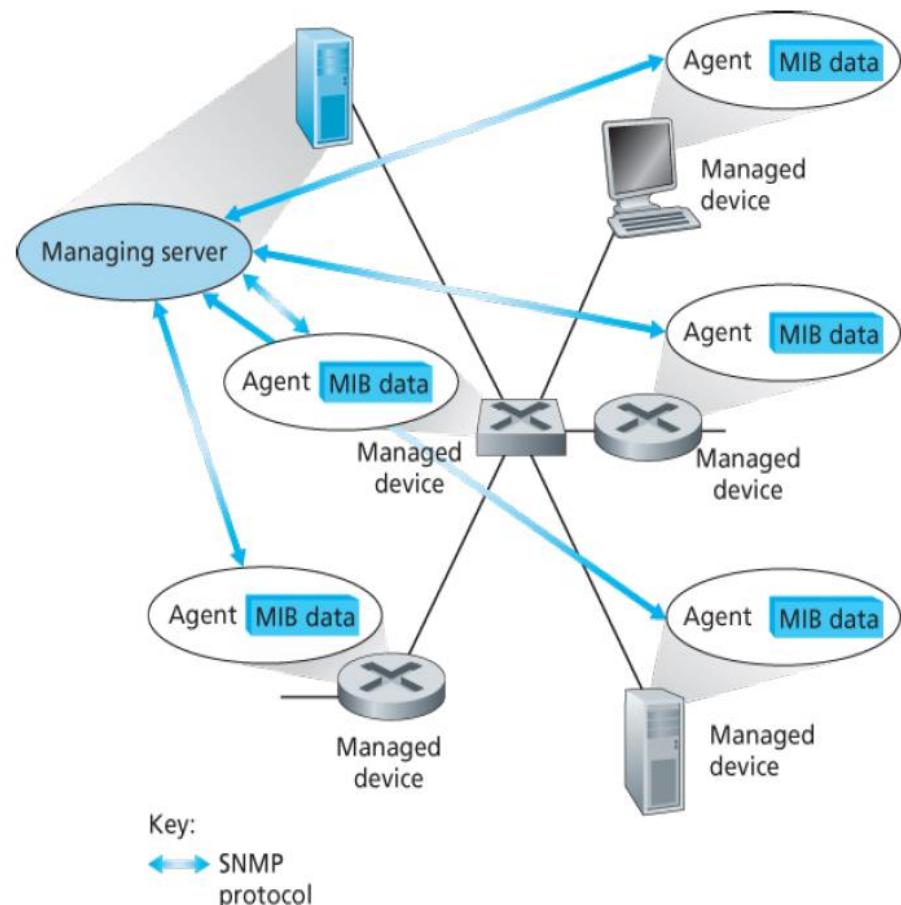


# 5.7: Quản lý mạng và SNMP

## ■ Các thành phần chính của quản lý mạng

- Tác nhân quản lý mạng (Network management agent)

- Một tiến trình (chạy trong thiết bị được quản lý) giao tiếp với server quản lý, thực hiện các hành động cục bộ ở thiết bị được quản lý theo lệnh và điều khiển của server quản lý



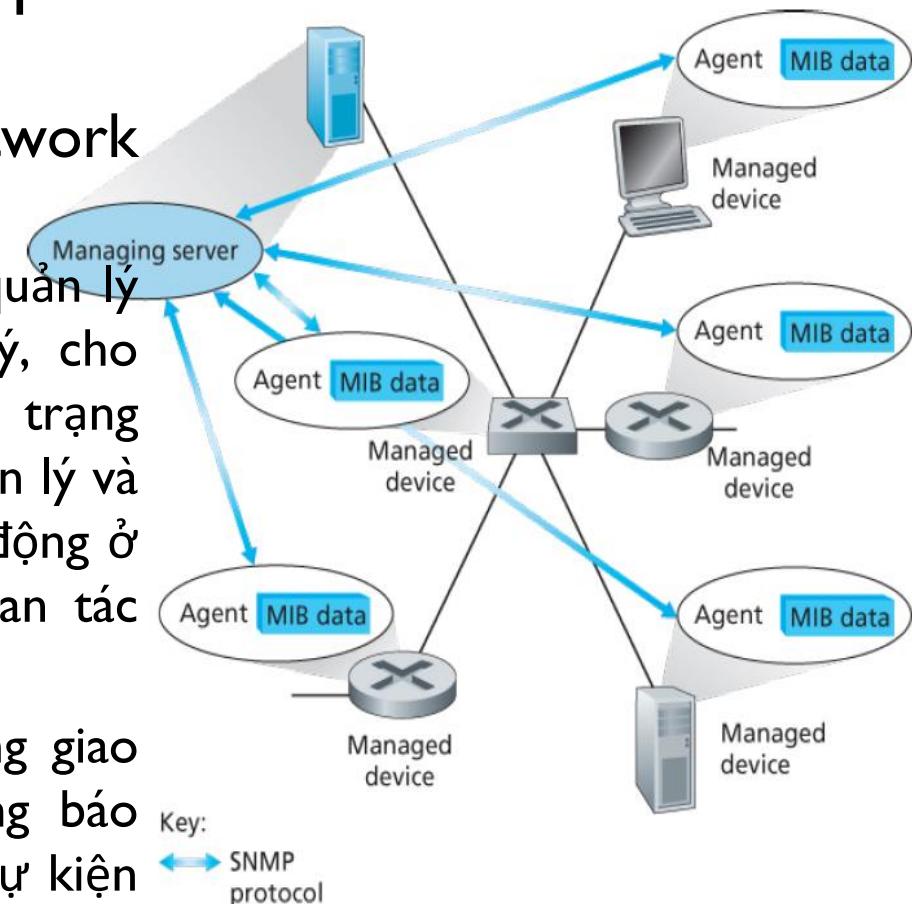
# 5.7: Quản lý mạng và SNMP

## ■ Các thành phần chính của quản lý mạng

- Giao thức quản lý mạng (network management protocol)

- Giao thức chạy giữa server quản lý và các thiết bị được quản lý, cho phép server quản lý truy vấn trạng thái của các thiết bị được quản lý và gián tiếp thực hiện các hành động ở những thiết bị này thông qua tác nhân của nó.

- Các tác nhân có thể sử dụng giao thức quản lý mạng để thông báo cho máy chủ quản lý về các sự kiện ngoại lệ (các lỗi thành phần hay vi phạm ngưỡng hiệu suất)



# 5.7: Quản lý mạng và SNMP

## ■ SNMP

- Giao thức tầng ứng dụng dùng để chuyển đạt các thông điệp thông tin và điều khiển quản lý mạng giữa một server quản lý và một tác nhân thực thi thay mặt cho server quản lý đó.
- Sử dụng phổ biến nhất của SNMP là ở chế độ request-response, thực hiện hành động nào đó, và gửi một reply tới cho request
  - Đển hình, một request truy vấn hay thay đổi các giá trị đối tượng MIB trong một thiết bị được quản lý

# 5.7: Quản lý mạng và SNMP

## ■ SNMP

- Sử dụng phổ biến thứ 2 của SNMP là cho tác nhân gửi một thông điệp không được yêu cầu, gọi là thông điệp bẫy (trap) tới một server quản lý
  - Các thông điệp trap được dùng để thông báo cho máy chủ quản lý về một tình huống ngoại lệ (giao diện liên kết đóng hay mở) dẫn đến những thay đổi đối với các giá trị đối tượng MIB.

# 5.7: Quản lý mạng và SNMP

## ■ SNMP

- SNMPv2: 7 kiểu thông điệp PDU (protocol data units)

Kiểu thông điệp	Bên gửi-bên nhận	Mô tả
GetRequest	Manager to agent	Nhận giá trị của một hoặc nhiều mẫu đối tượng MIB
GetNextRequest	Manager to agent	Nhận giá trị của mẫu đối tượng MIB tiếp theo trong danh sách hay trong bảng
GetBulkRequest	Manager to agent	Nhận các giá trị trong khối dữ liệu lớn, ví dụ, các giá trị một bảng lớn

# 5.7: Quản lý mạng và SNMP

## ■ SNMP

- SNMPv2: 7 kiểu thông điệp PDU (protocol data units)

Kiểu thông điệp	Bên gửi-bên nhận	Mô tả
InformRequest	Manager to Manager	Thông báo thực thể quản lý từ xa các giá trị MIB truy cập từ xa vào nó
SetRequest	Manager to agent	Thiết lập giá trị của một hoặc nhiều mẫu đối tượng MIB

# 5.7: Quản lý mạng và SNMP

## ■ SNMP

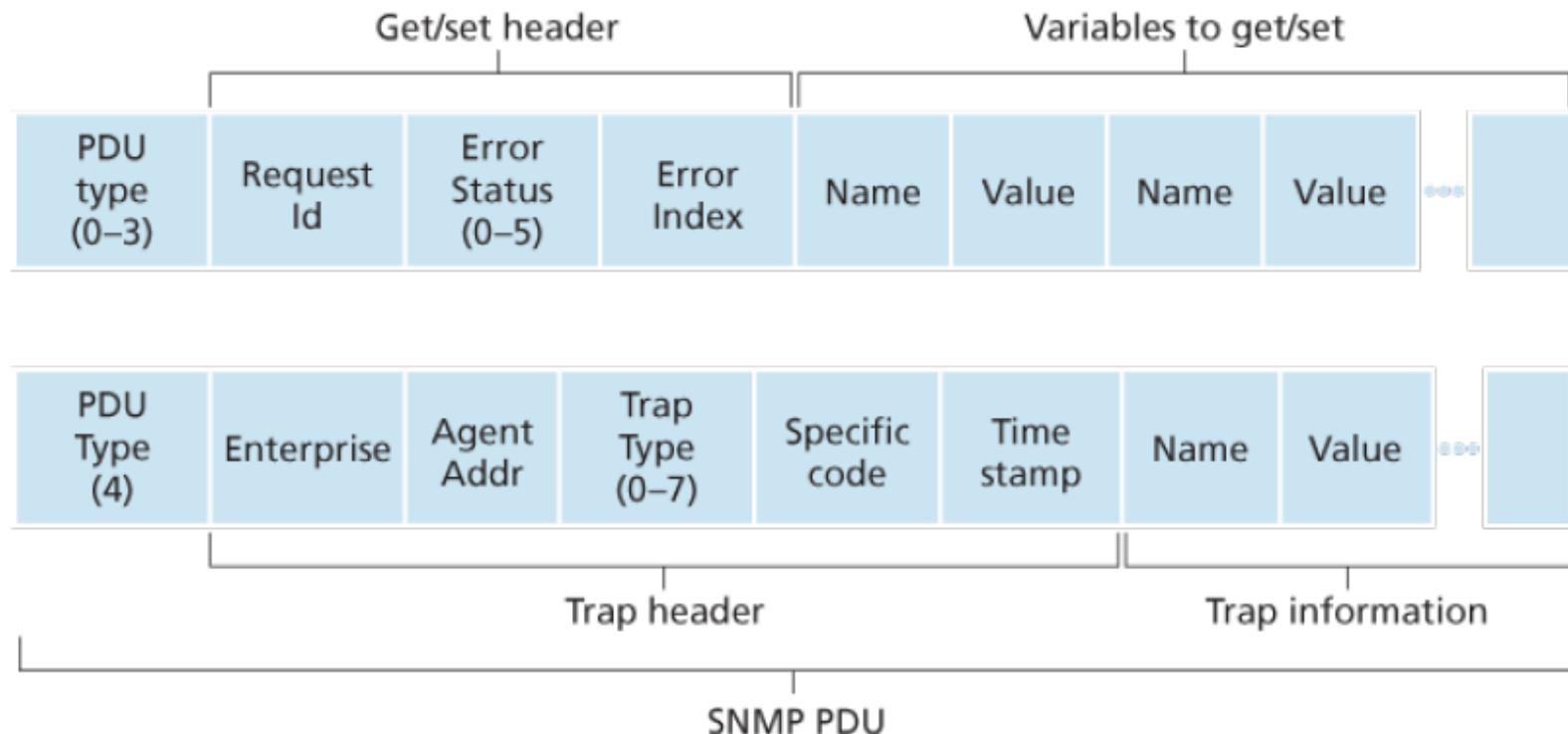
- SNMPv2: 7 kiểu thông điệp PDU (protocol data units)

Kiểu thông điệp	Bên gửi-bên nhận	Mô tả
Response	Được tạo ra để trả lời	GetRequest, GetNextRequest, GetBulkRequest, SetRequest PDU, hay InformRequest
SNMPv2-Trap	Agent to Manager	Thông báo cho manager về sự kiện ngoại lệ

# 5.7: Quản lý mạng và SNMP

## ■ SNMP

- Định dạng của PDU



# 5.7: Quản lý mạng và SNMP

- **SNMP**

- SNMP có 3 phiên bản

- SNMPv1
  - SNMPv2
  - SNMPv3

- Giống như SNMPv2 nhưng bổ sung thêm khả năng quản trị và an toàn