

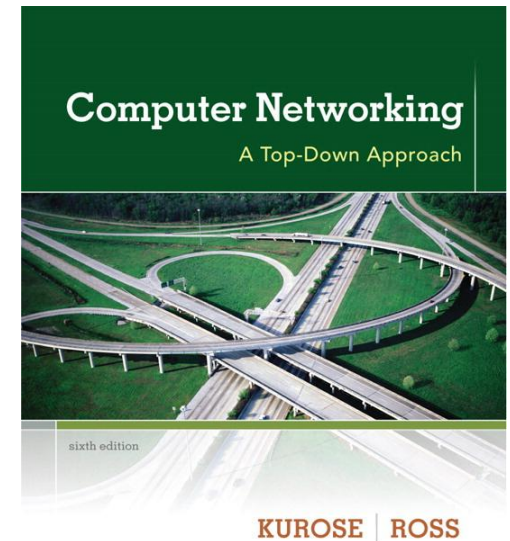
# Chương 4

## Tầng mạng

**Người dịch: Nguyễn Thanh Thủy**

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

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



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

# Chương 4: Tầng mạng

## *Mục tiêu:*

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

# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

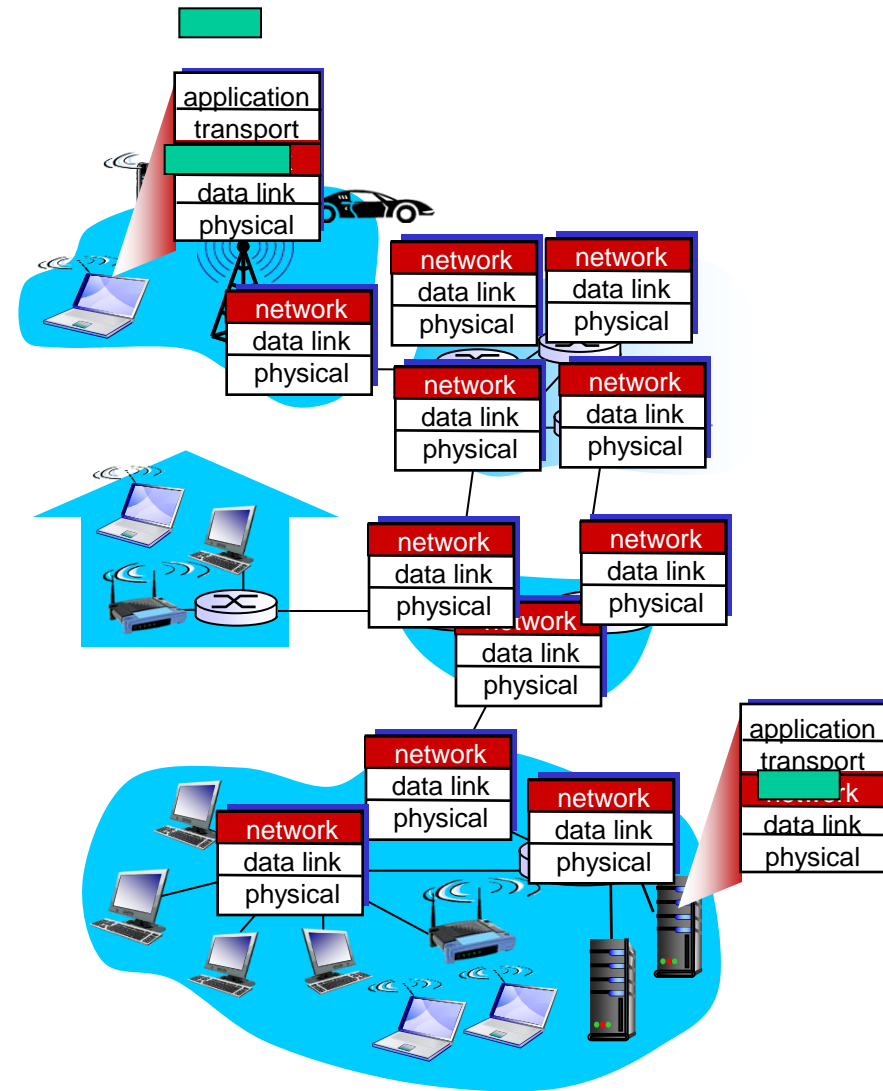
- Link state
- Distance vector
- Hierarchical routing

## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

# Tầng mạng

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



# Hai chức năng chính của tầng mạng

❖ *Chuyển tiếp (forwarding)*: chuyển các gói tin từ đầu vào tới đầu ra phù hợp của router

❖ *Định tuyến (routing)*: xác định tuyến đường đi cho các gói tin từ nguồn đến đích.

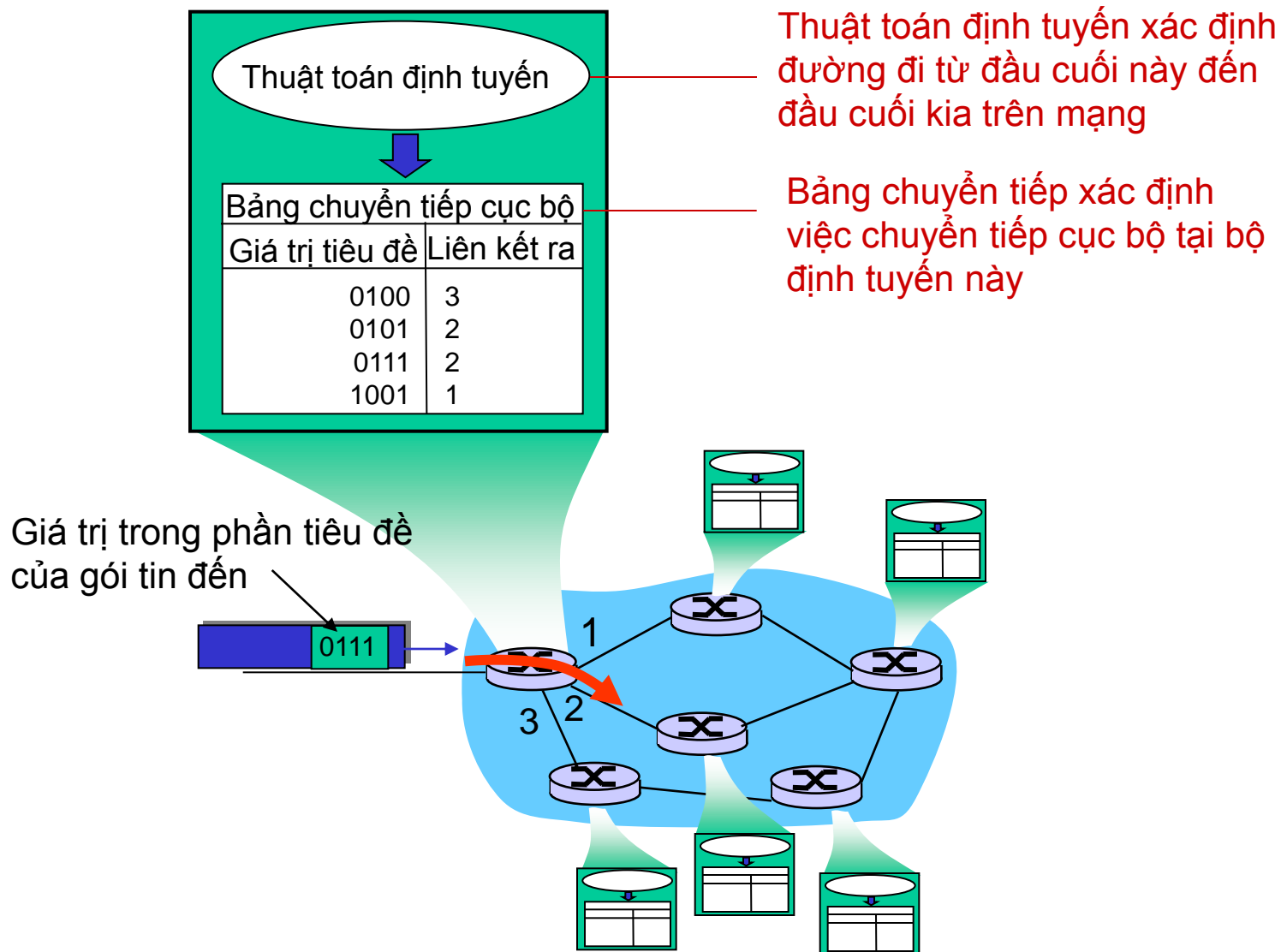
- *Các thuật toán định tuyến*

## *Tương tự:*

❖ *Định tuyến*: tiến trình lập kế hoạch chuyển đi từ nguồn đến đích

❖ *Chuyển tiếp*: tiến trình vận chuyển qua một giao điểm (nút)

# Tác động qua lại giữa định tuyến và chuyển tiếp



# Thiết lập kết nối

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

# Mô hình dịch vụ tầng mạng

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

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

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

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

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



# Các mô hình dịch vụ tầng mạng

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

# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

# Dịch vụ hướng kết nối và không kết nối

---

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

# Mạch ảo (Virtual circuit - VC)

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

- Hiệu suất tốt
- Mạng hoạt động theo đường từ nguồn đến đích

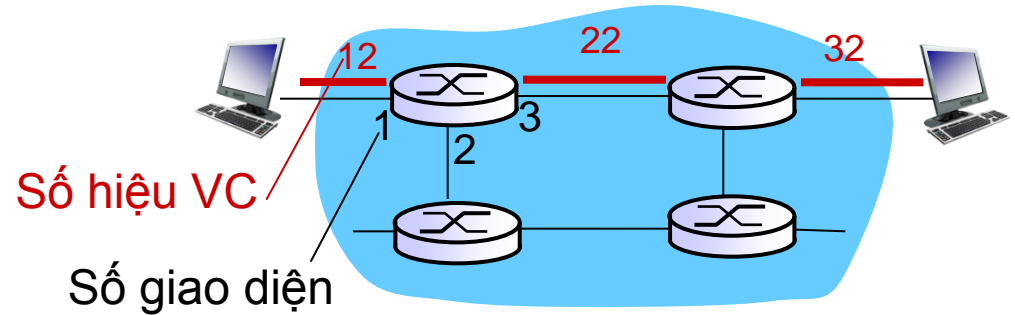
- ❖ Thiết lập cuộc gọi, chia nhỏ *mỗi* cuộc gọi trước khi dữ liệu được truyền đi
- ❖ Mỗi gói tin mang định danh mạch ảo (không phải là địa chỉ của host đích)
- ❖ *Mỗi* router trên đường đi từ nguồn đến đích duy trì “trạng thái” cho mỗi kết nối qua.
- ❖ Kết nối, các tài nguyên router (bảng thông, đệm)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

# Cài đặt mạng ảo

## *Một mạng ảo bao gồm:*

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

# Bảng chuyển tiếp mạch ảo



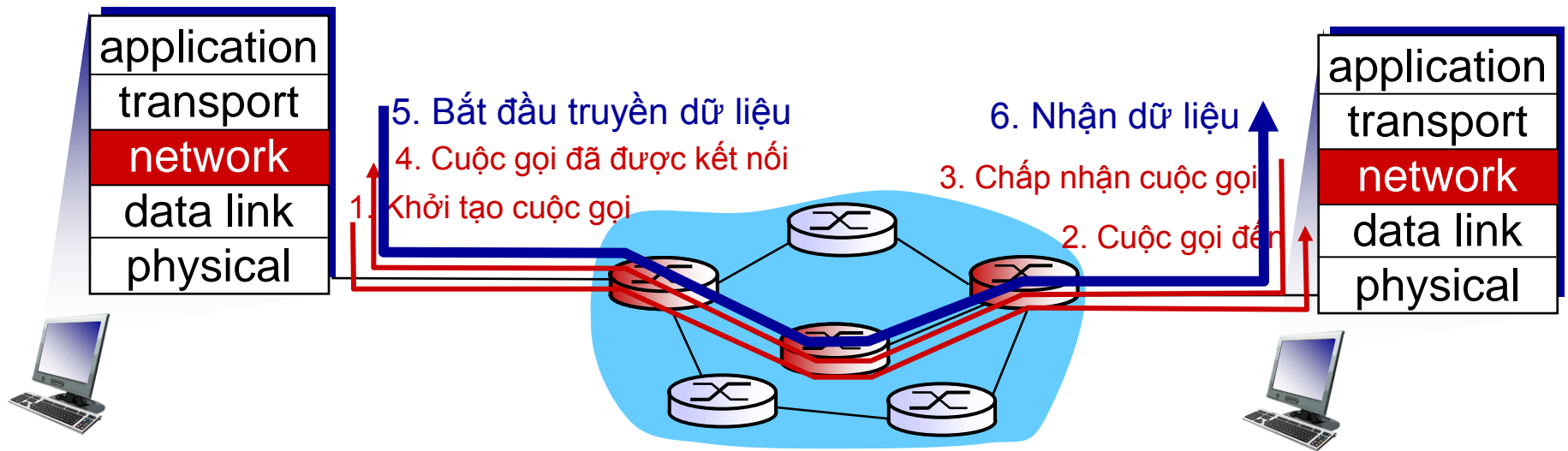
*Ví dụ bảng chuyển tiếp của router:*

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

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

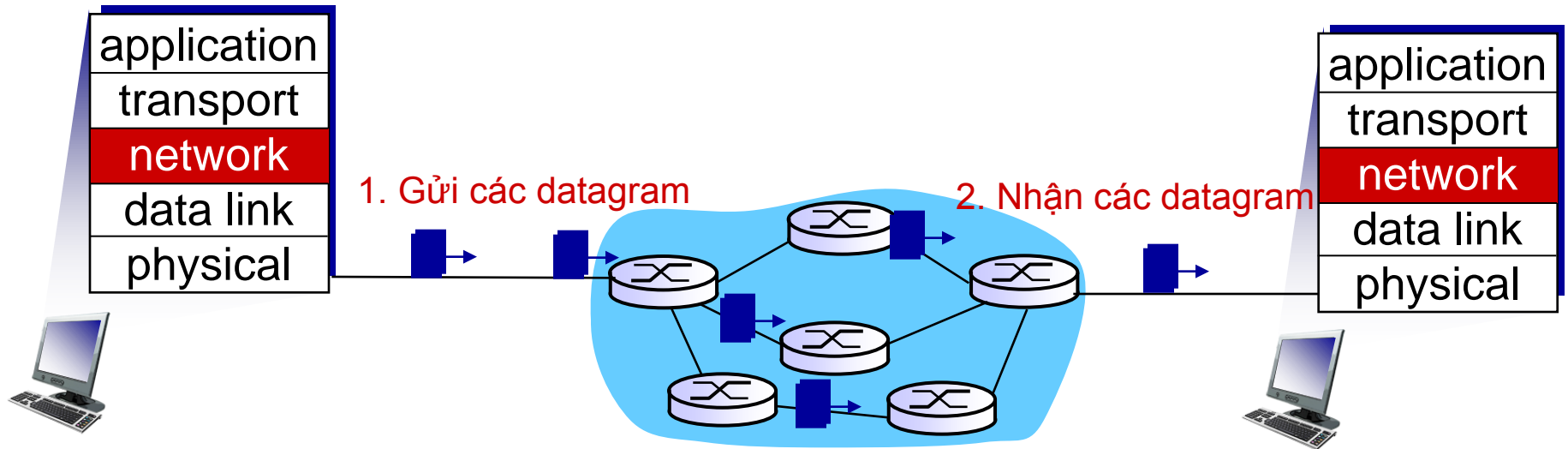
# Mạch ảo: Các giao thức báo hiệu

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



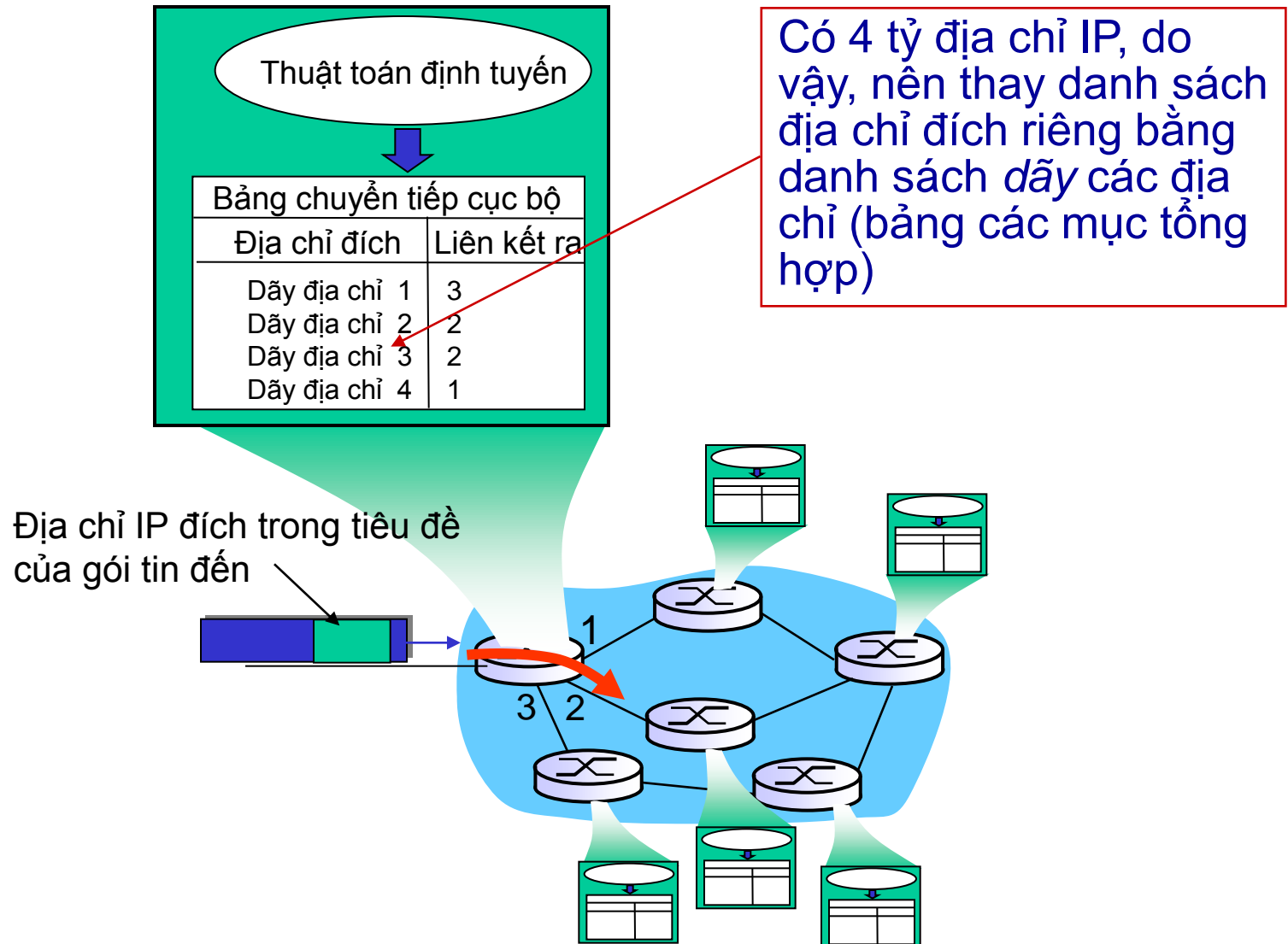
# Mạng chuyển gói

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





# Bảng chuyển tiếp datagram



# Bảng chuyển tiếp datagram

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

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

# So khớp tiền tố dài nhất

## *So khớp tiền tố dài nhất*

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

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

Ví dụ:

DA: 11001000 00010111 00010110 10100001

Giao diện nào?

DA: 11001000 00010111 00011000 10101010

Giao diện nào?

# Chuyển mạch gói hay chuyển mạch ảo: Tại sao?

## *Internet (datagram)*

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

## *ATM (VC)*

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

# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

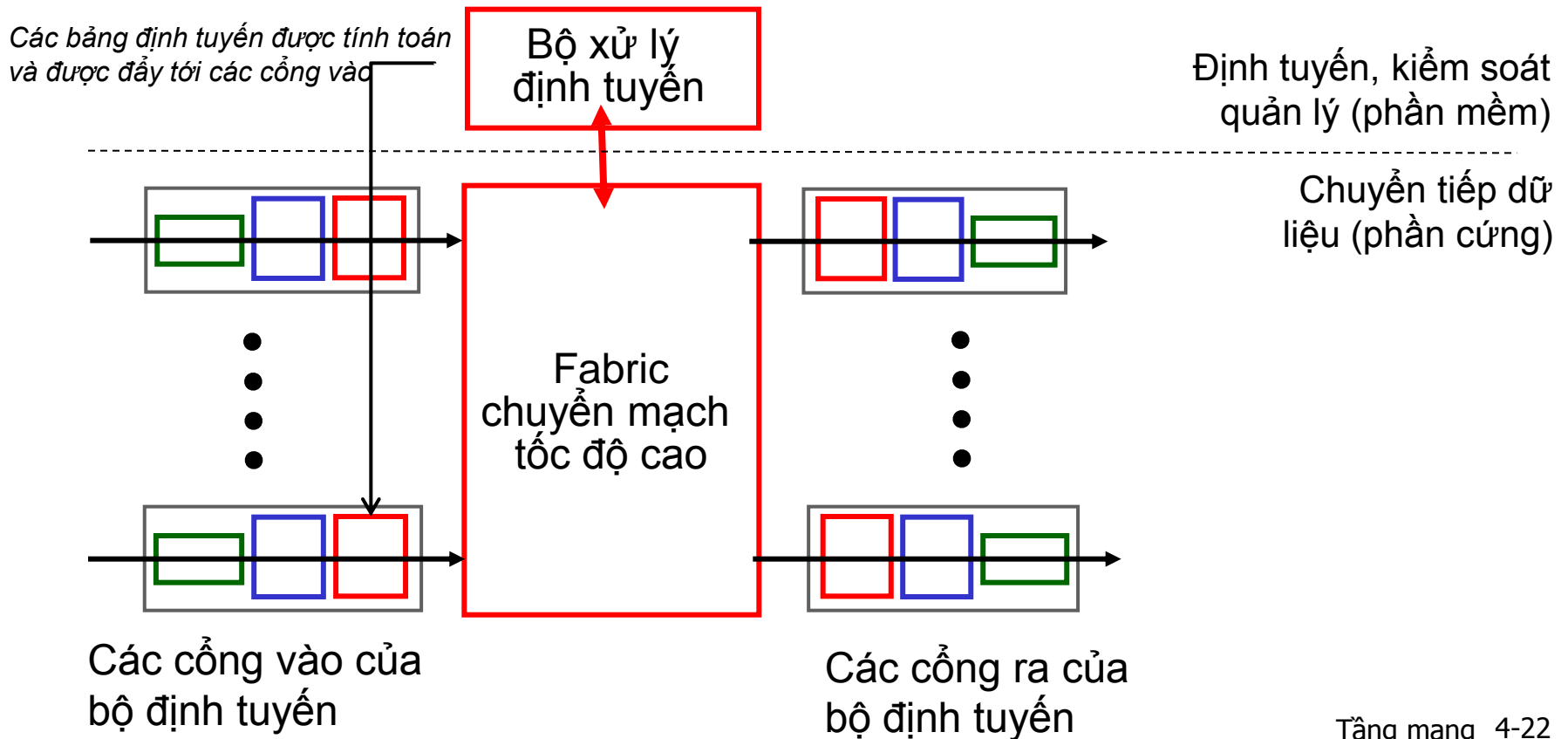
## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

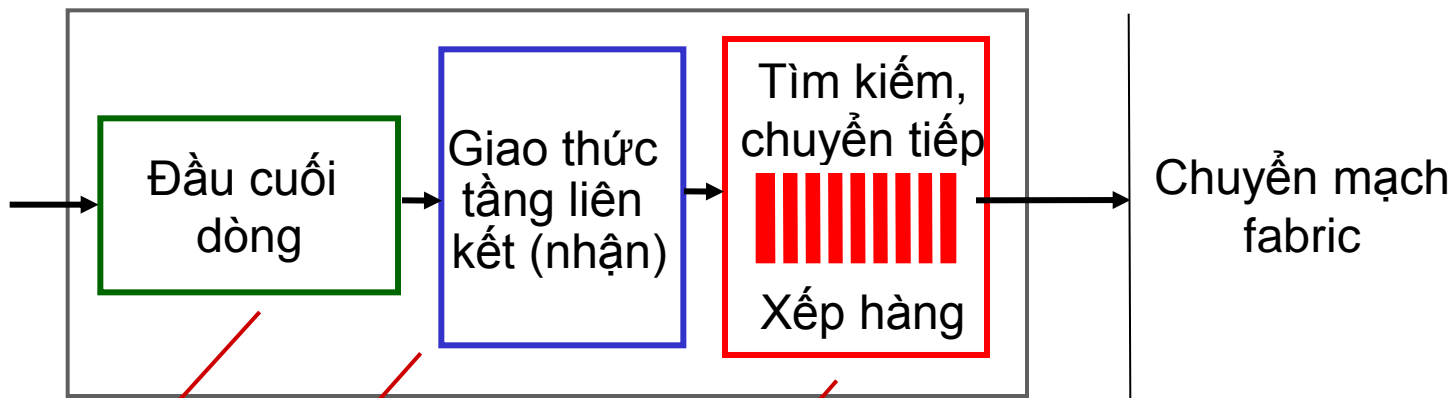
# Khái quát kiến trúc của bộ định tuyến

Hai chức năng chính của bộ định tuyến:

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



# Các chức năng của cổng vào



Tầng vật lý:  
Tiếp nhận mức bit

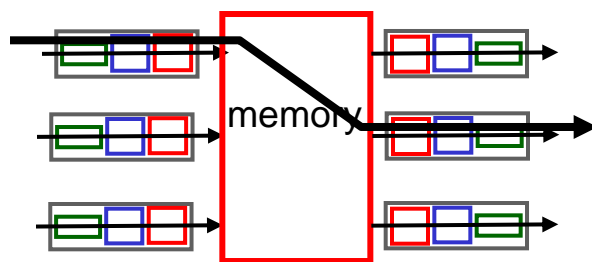
Tầng liên kết  
dữ liệu:  
Ví dụ: Ethernet  
(xem chương 5)

Chuyển mạch không tập trung:

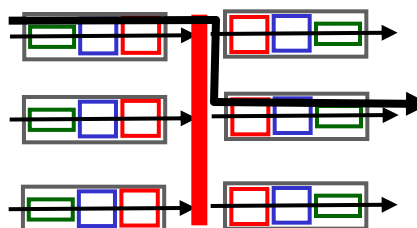
- ❖ Với đích của datagram đã biết, tìm cổng ra bằng bảng chuyển tiếp trong bộ nhớ cổng vào
- ❖ Mục đích: hoàn thành xử lý cổng vào theo “tốc độ dòng”
- ❖ Xếp hàng: nếu các datagram đến nhanh hơn tốc độ chuyển tiếp trong bộ chuyển mạch fabric

# Chuyển mạch fabric

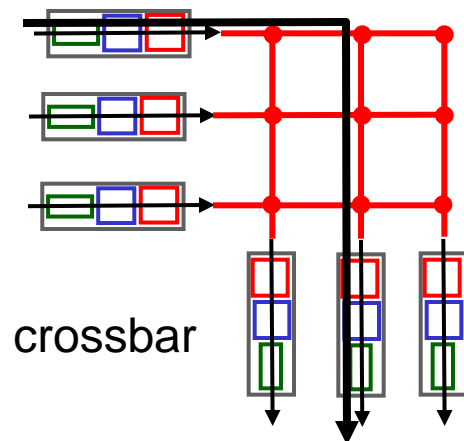
- ❖ Chuyển gói tin từ vùng đệm vào đến vùng đệm ra phù hợp
- ❖ Tốc độ chuyển mạch: là tốc độ mà các gói tin có thể được chuyển từ các đầu vào tới các đầu ra
  - Thường được đo như là bội số của tốc độ dòng vào/dòng ra
  - N đầu vào: tốc độ chuyển mạch bằng N lần tốc độ dòng mong muốn
- ❖ Có 3 loại chuyển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



memory  
(bộ nhớ)



bus



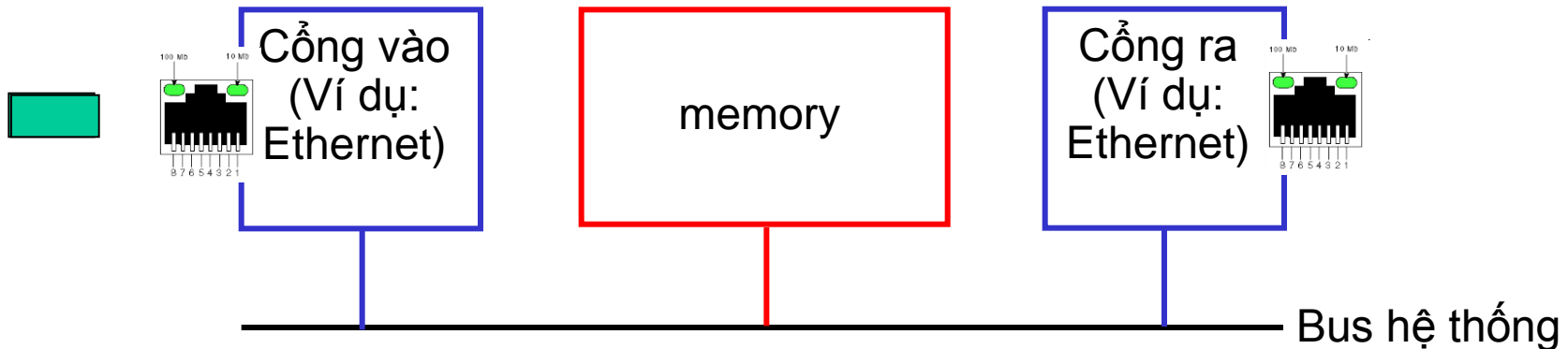
crossbar



# Chuyển mạch qua memory

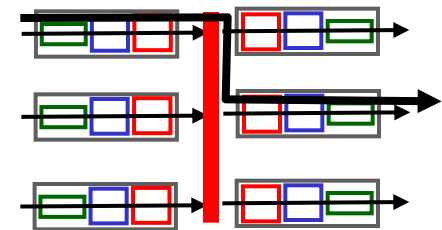
## *Các bộ định tuyến thế hệ đầu tiên:*

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



# Chuyển mạch qua bus

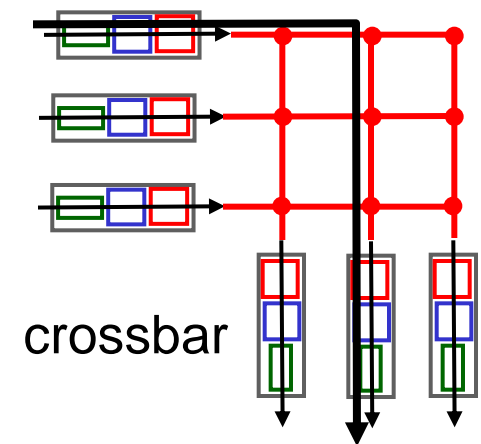
- ❖ Datagram từ bộ nhớ cổng vào tới bộ nhớ cổng ra thông qua một bus chung
- ❖ *Tranh chấp bus*: tốc độ chuyển mạch bị giới hạn bởi băng thông của bus
- ❖ 32 Gbps bus, Cisco 5600: tốc độ đủ cho các router truy nhập và các router của tổ chức.



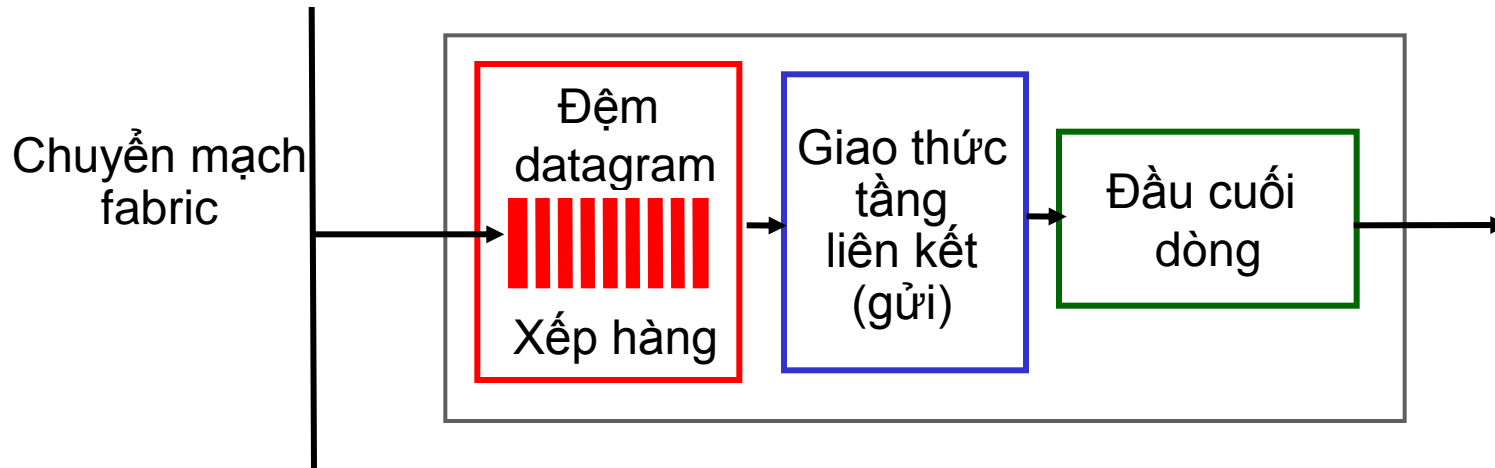
bus

# Chuyển mạch thông qua mạng kết nối nội bộ

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

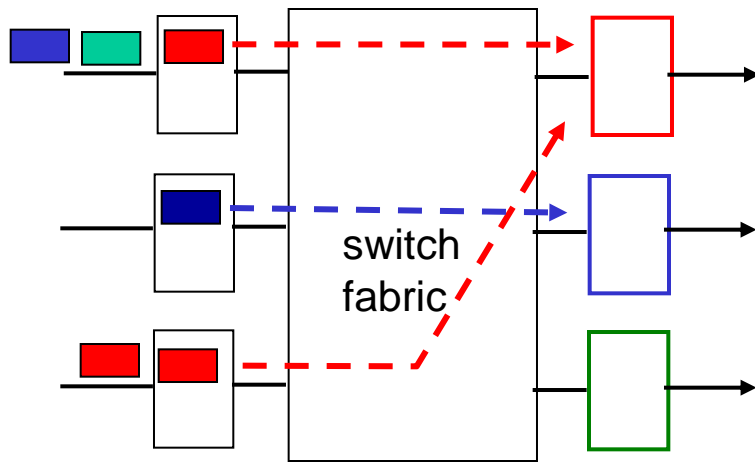


# Các cổng ra

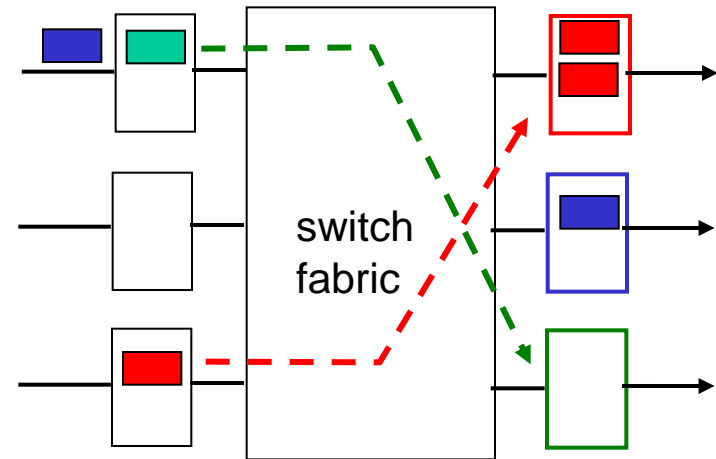


- ❖ *Việc đệm* được yêu cầu khi các datagram đến từ fabric nhanh hơn tốc độ truyền đi
- ❖ *Lịch truyền* sẽ lựa chọn các datagram trong hàng đợi để truyền

# Xếp hàng tại cổng ra



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



Mỗi lần một gói

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

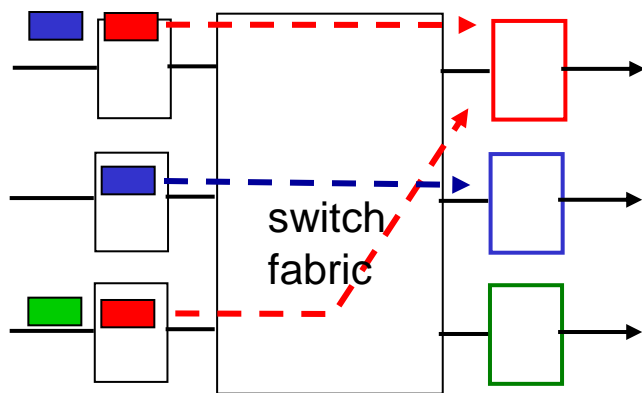
# Cần bao nhiêu cho bộ đệm?

- ❖ 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:

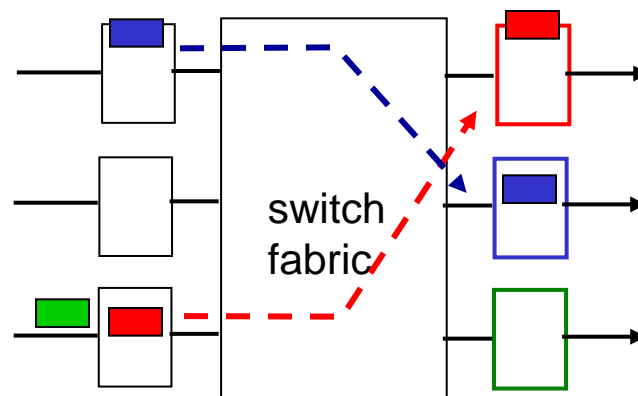
$$\frac{RTT \cdot C}{\sqrt{N}}$$

# Xếp hàng tại cổng vào

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



Tranh chấp cổng ra:  
Chỉ một datagram màu đỏ có thể được truyền.  
Gói tin màu đỏ ở phía dưới bị khóa



Mỗi lần một gói: Gói màu xanh lá cây bị chặn do khóa HOL

# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

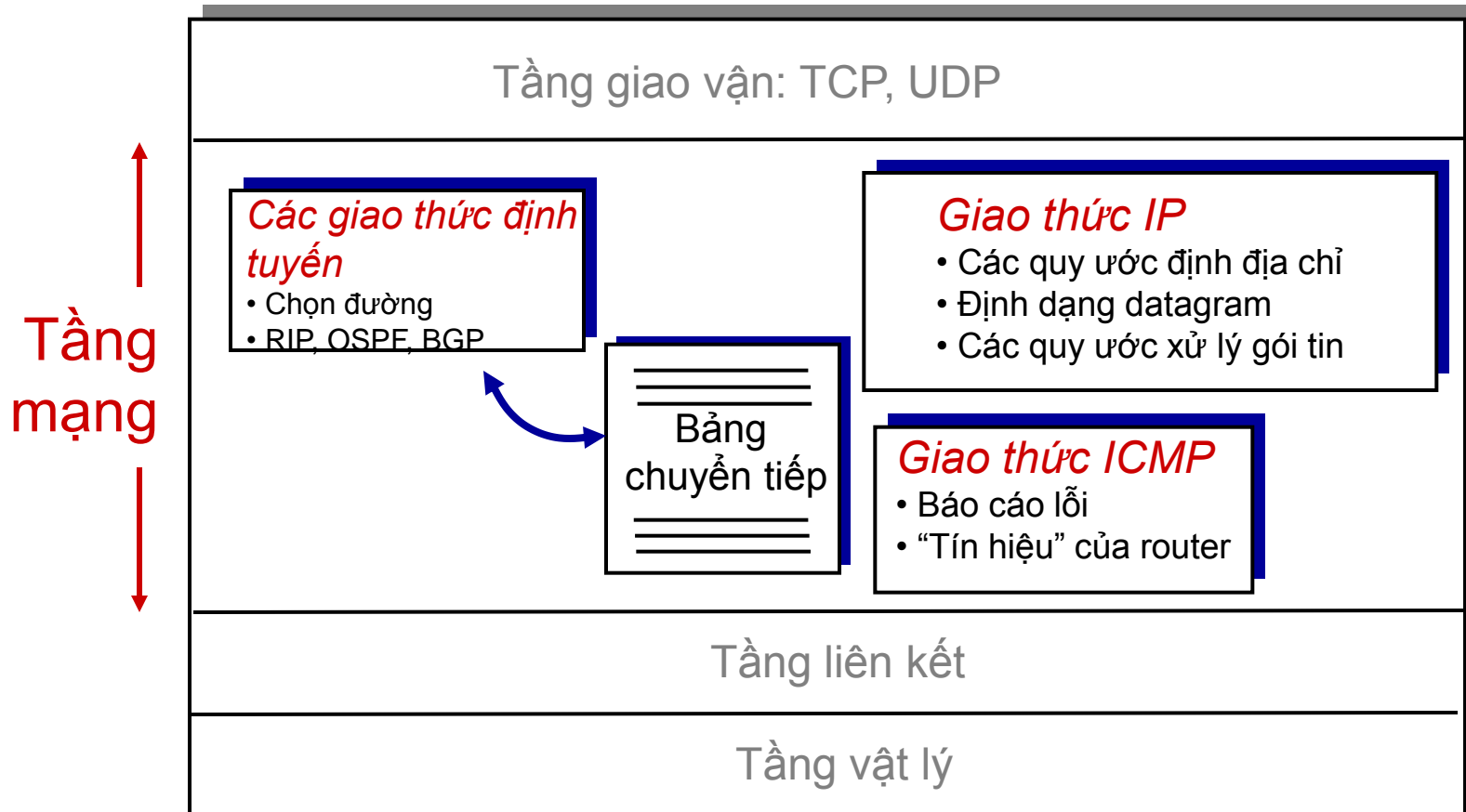
## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP



# Tầng mạng trong mạng Internet

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



# Định dạng IP datagram

Phiên bản giao thức IP

Chiều dài phần  
tiêu đề (byte)

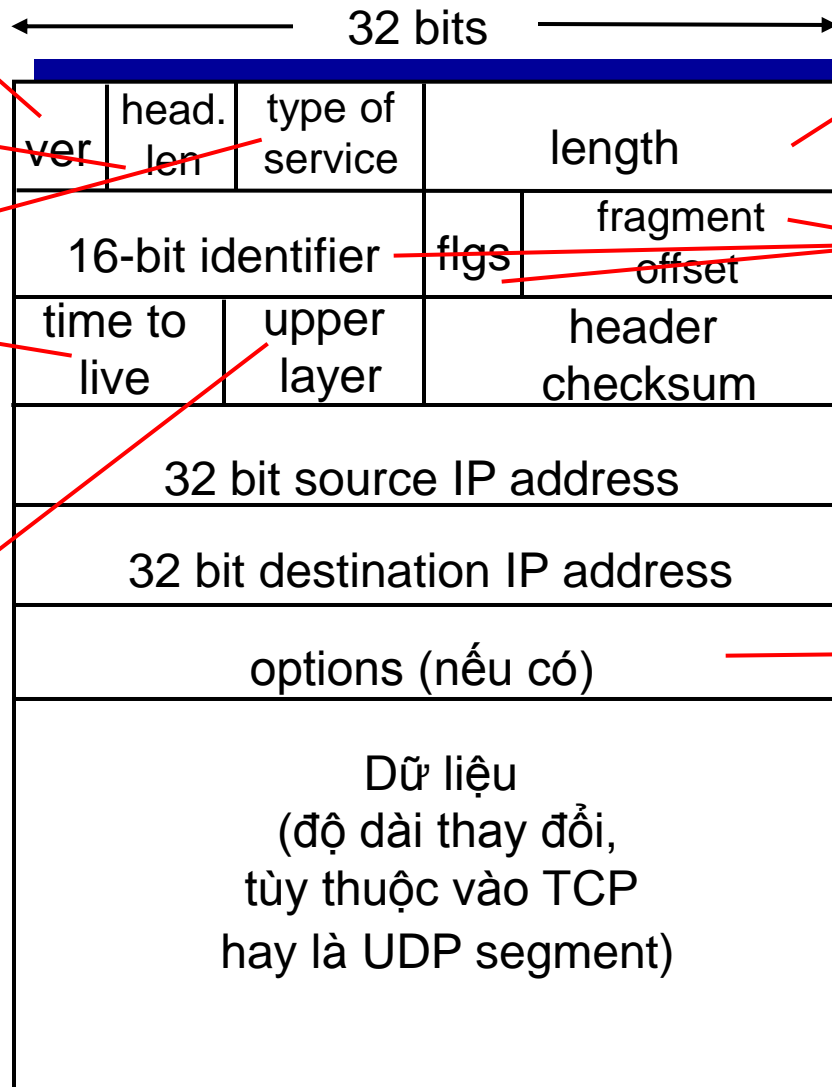
“Loại” dữ liệu

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

Giao thức tầng  
cao hơn thực hiện  
phân phối payload đi

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

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



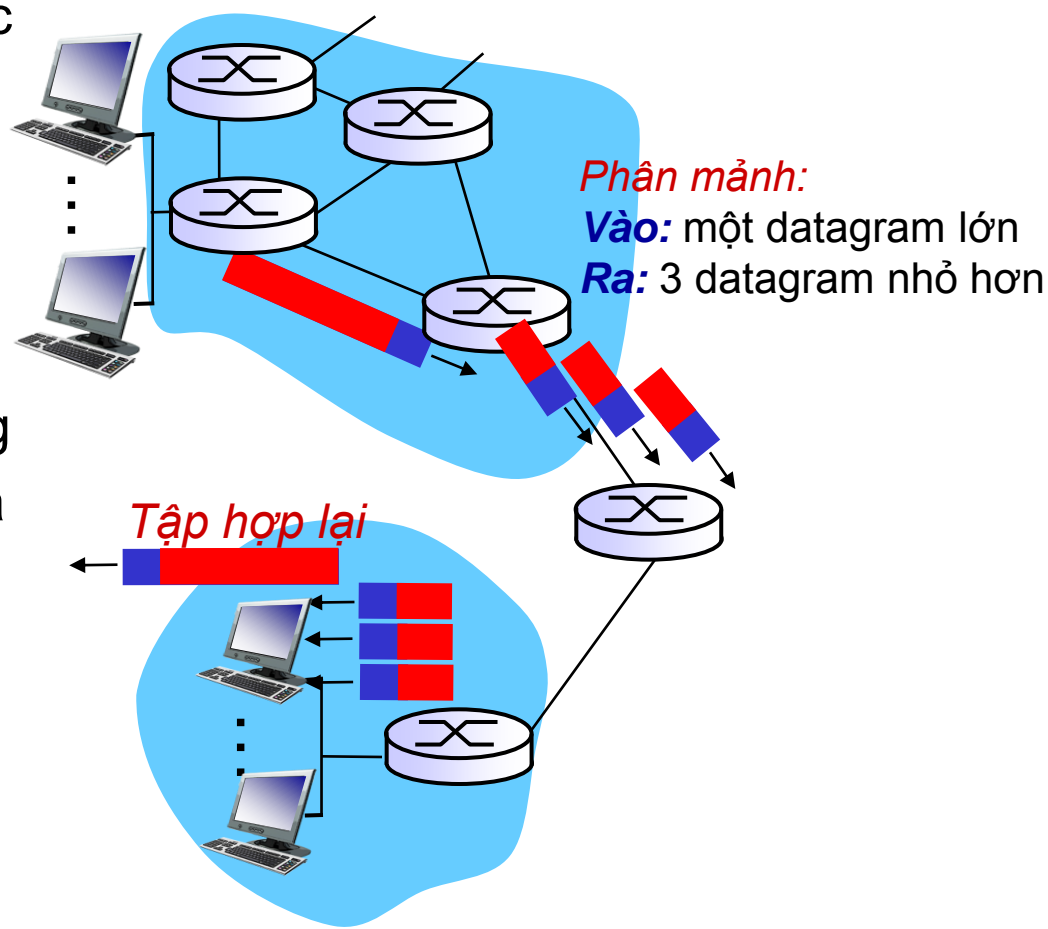
Tổng chiều dài  
datagram  
(byte)

Dùng cho việc  
phân mảnh/tập  
hợp lại

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

# Phân mảnh và tập hợp lại gói tin IP

- ❖ Các liên kết mạng có MTU (max.transfer unit) – kích thước khung (frame) lớn nhất có thể được truyền trên liên kết.
  - 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 byte

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

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

1480 byte trong  
trường dữ liệu

offset =  
 $1480/8$

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

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

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

# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

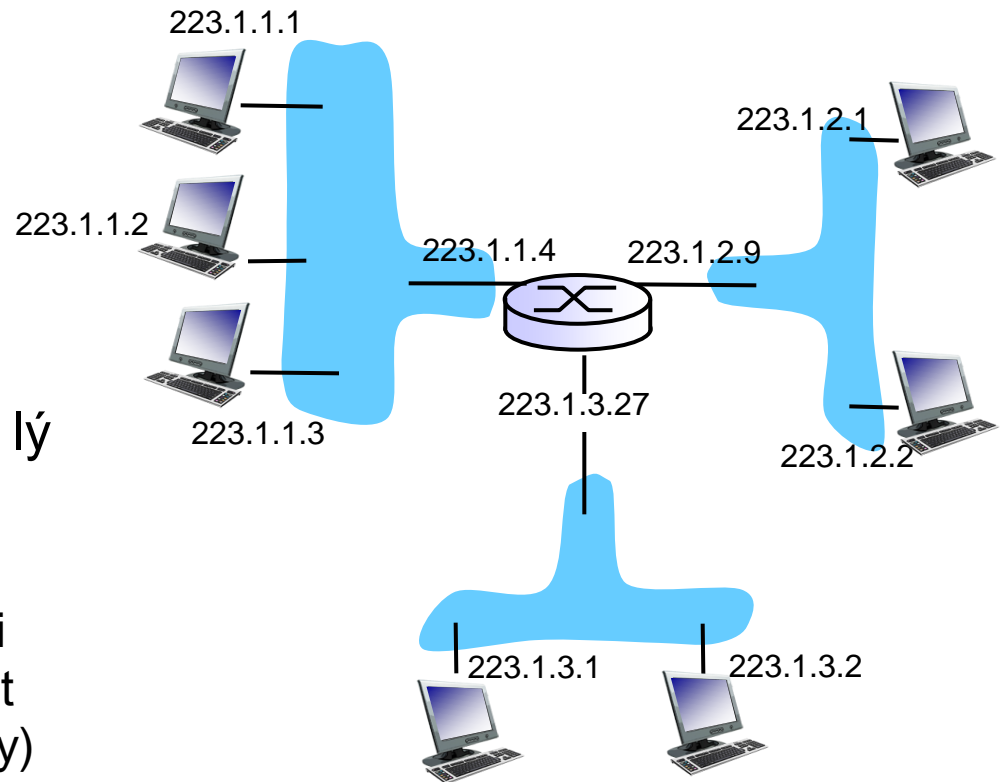
- Link state
- Distance vector
- Hierarchical routing

## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

# Định địa chỉ IP: giới thiệu

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



$$223.1.1.1 = \underbrace{11011111}_{223} \underbrace{00000001}_1 \underbrace{00000001}_1 \underbrace{00000001}_1$$

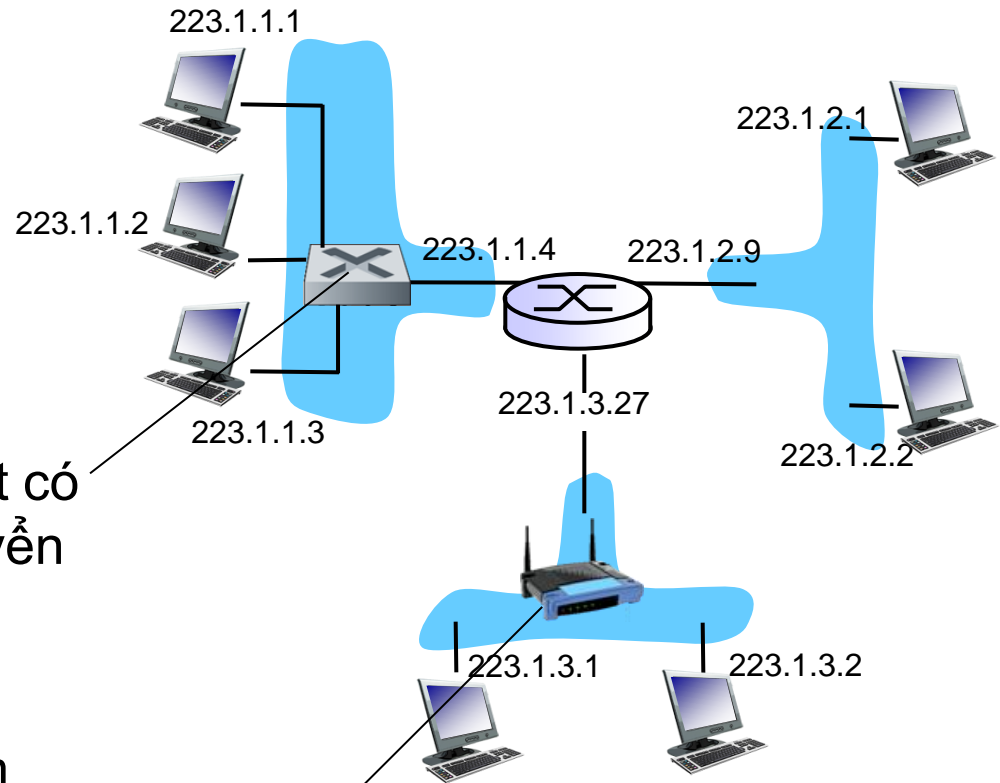
# Định địa chỉ IP: giới thiệu

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

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

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

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



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

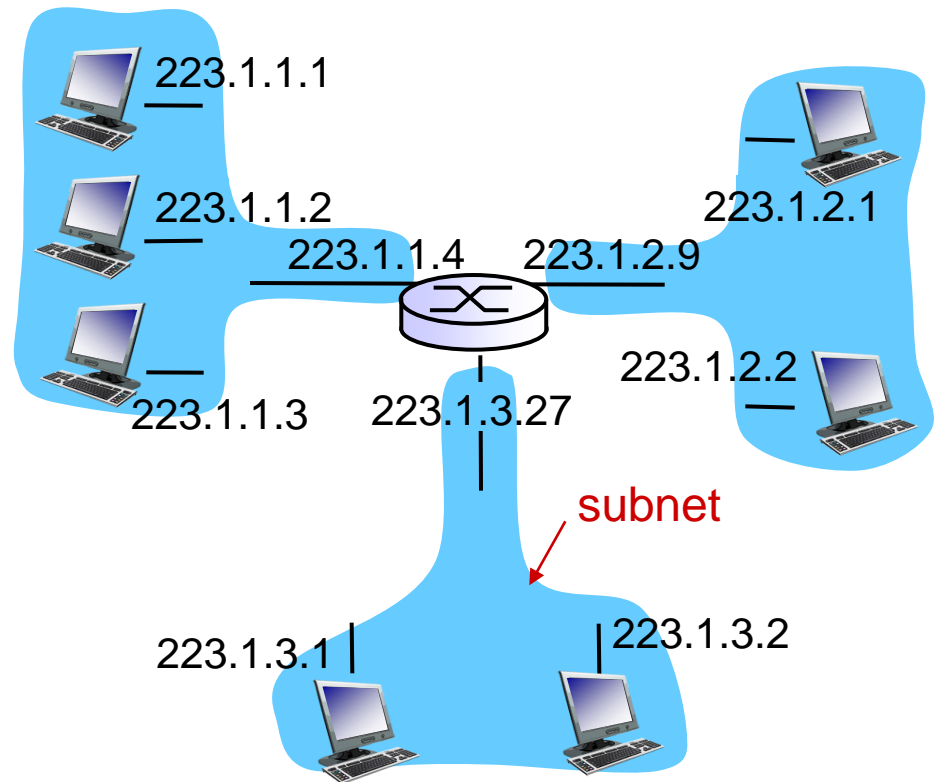
# Subnet (Mạng con)

## ❖ Địa chỉ IP:

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

## ❖ Subnet là gì?

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



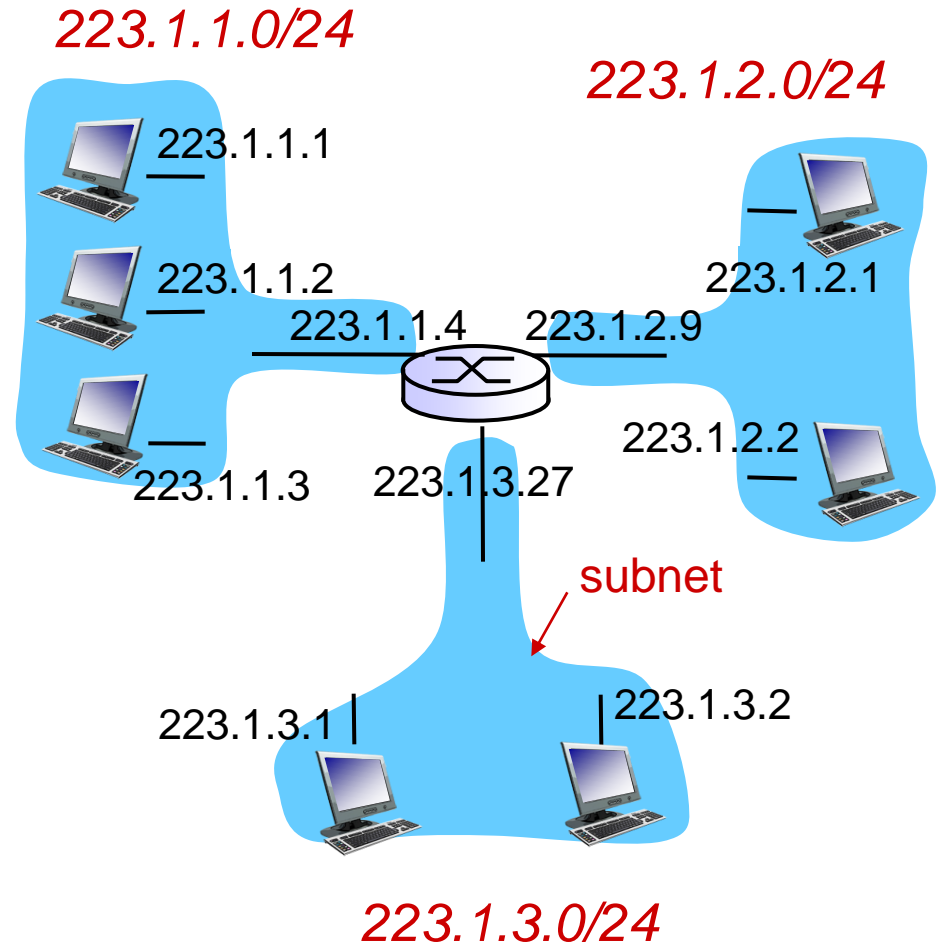
Mạng có 3 subnet



# Subnet

## *Phương pháp*

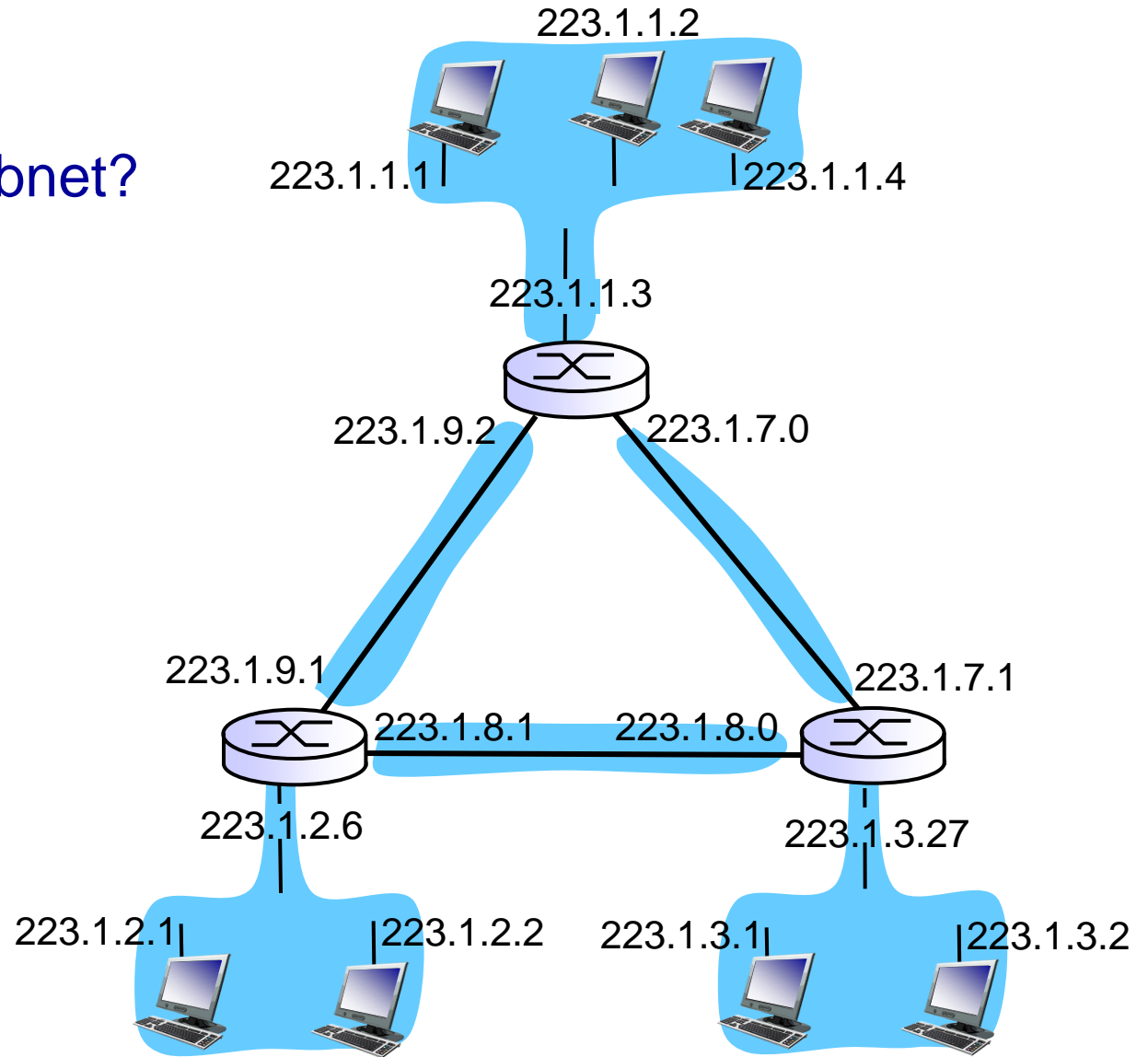
- ❖ Để xác định các subnet, tách mỗi giao diện từ host hoặc router, tạo thành các vùng mạng độc lập
- ❖ Mỗi mạng độc lập được gọi là một *subnet*



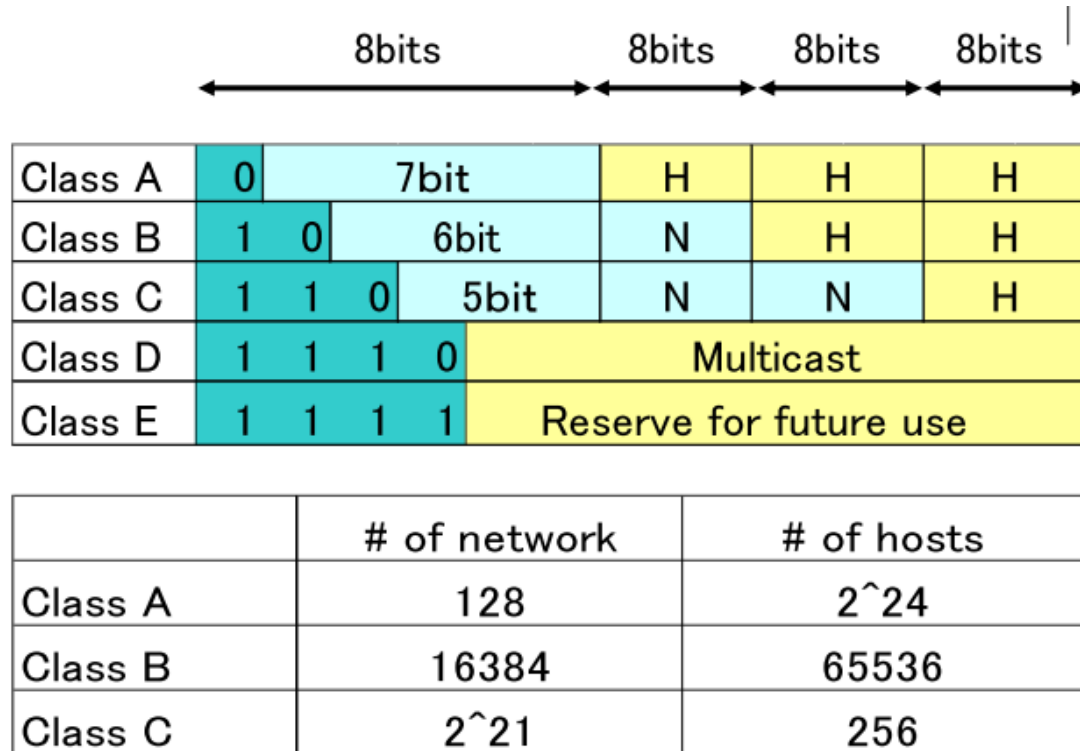
Mặt nạ mạng con (subnet mask): /24

# Subnet

Có bao nhiêu subnet?



# Định địa chỉ IP: Phân lớp địa chỉ IPv4



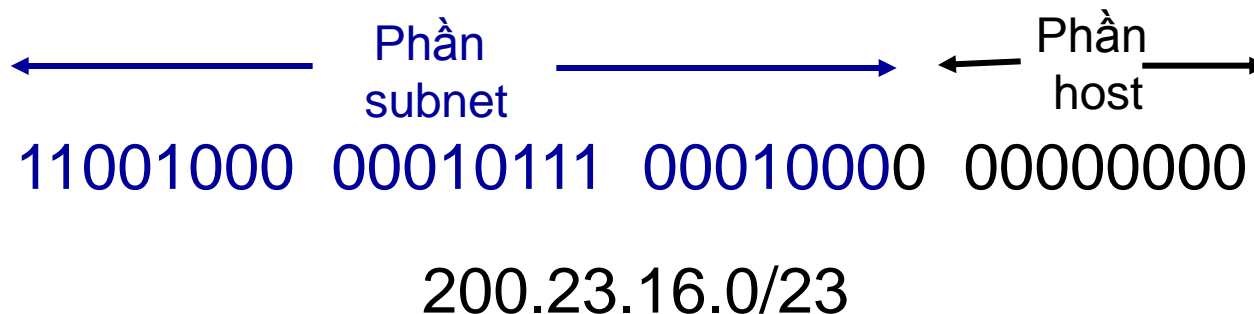
**Hạn chế:** lãng phí không gian địa chỉ

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

# Định địa chỉ IP: CIDR

## CIDR: Classless Inter Domain Routing

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



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

---

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

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

# DHCP: Dynamic Host Configuration Protocol

---

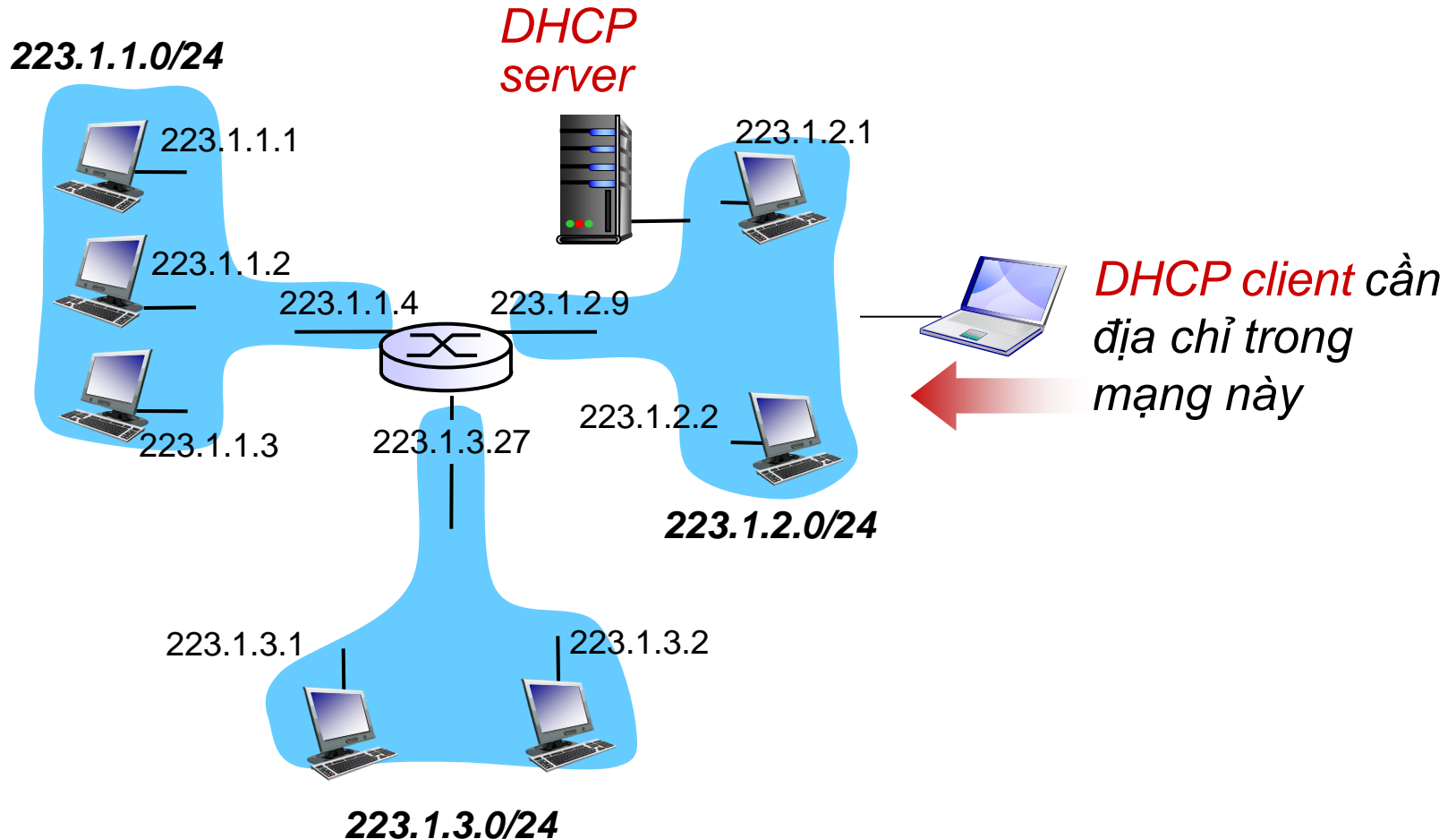
**Mục đích:** cho phép host có được địa chỉ IP *một cách tự động* từ server mạng khi kết nối vào mạng

- 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 server: 223.1.2.5

**DHCP discover**

client



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

**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

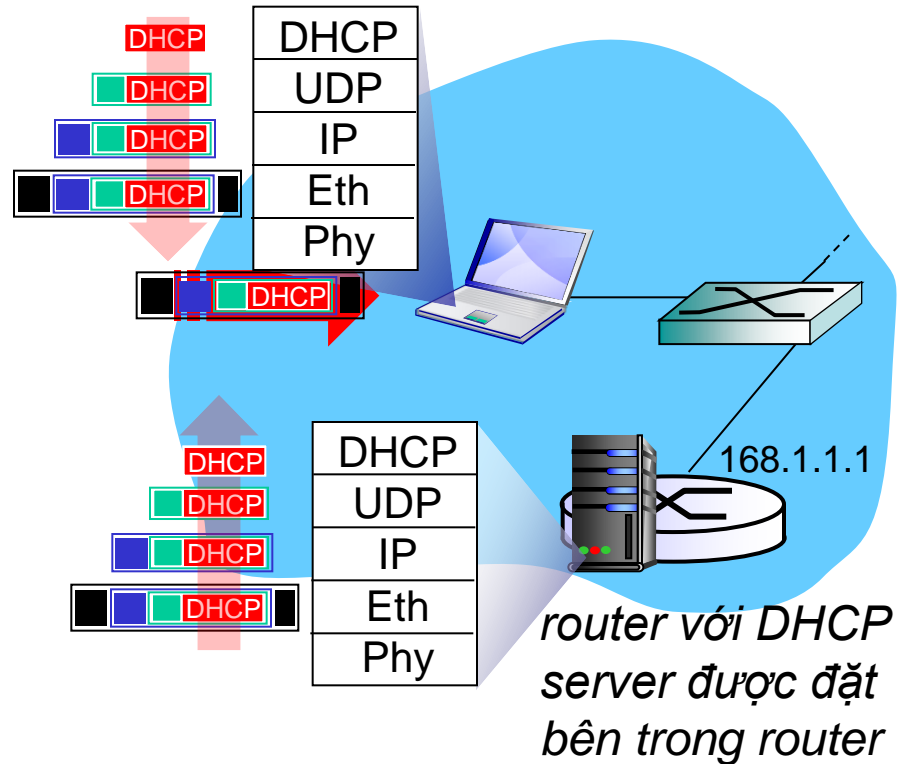


# DHCP: có nhiều địa chỉ IP hơn

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

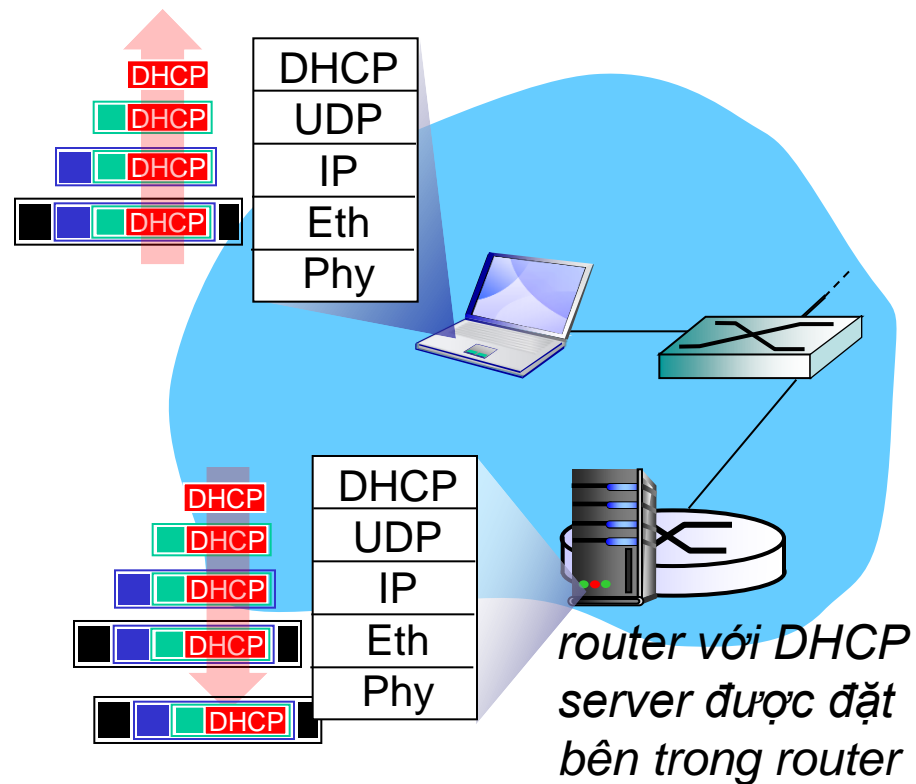
- Địa chỉ của router của hop đầu tiên cho client
- Tên và địa chỉ IP của DNS sever
- Mặt nạ mạng (chỉ ra phần host và phần mạng của một địa chỉ)

# DHCP: Ví dụ



- ❖ 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: FFFFFFFFFFFFFFFF) trên mạng LAN, được router đang chạy DHCP server nhận
- ❖ Ethernet được cắt bỏ phần tiêu đề thành IP, IP được cắt bỏ phần tiêu đề thành UDP, UDP được cắt bỏ phần tiêu đề thành DHCP.

# DHCP: Ví dụ



- ❖ DCP 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 DSN server, và địa chỉ IP của router của hop đầu tiên của nó.

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

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

Hardware type: Ethernet

Hardware address length: 6

Hops: 0

**Transaction ID: 0x6b3a11b7**

Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)

Client IP address: 0.0.0.0 (0.0.0.0)

Your (client) IP address: 0.0.0.0 (0.0.0.0)

Next server IP address: 0.0.0.0 (0.0.0.0)

Relay agent IP address: 0.0.0.0 (0.0.0.0)

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

Server host name not given

Boot file name not given

Magic cookie: (OK)

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

Option: (61) Client identifier

Length: 7; Value: 010016D323688A;

Hardware type: Ethernet

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

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

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

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

Length: 11; Value: 010F03062C2E2F1F21F92B

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

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

44 = NetBIOS over TCP/IP Name Server

.....

Yêu cầu

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

Hardware type: Ethernet

Hardware address length: 6

Hops: 0

**Transaction ID: 0x6b3a11b7**

Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)

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

Your (client) IP address: 0.0.0.0 (0.0.0.0)

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

Relay agent IP address: 0.0.0.0 (0.0.0.0)

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

Server host name not given

Boot file name not given

Magic cookie: (OK)

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

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

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

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

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

Length: 12; Value: 445747E2445749F244574092;

IP Address: 68.87.71.226;

IP Address: 68.87.73.242;

IP Address: 68.87.64.146

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

Đáp ứng

# Làm thế nào có được một địa chỉ IP?

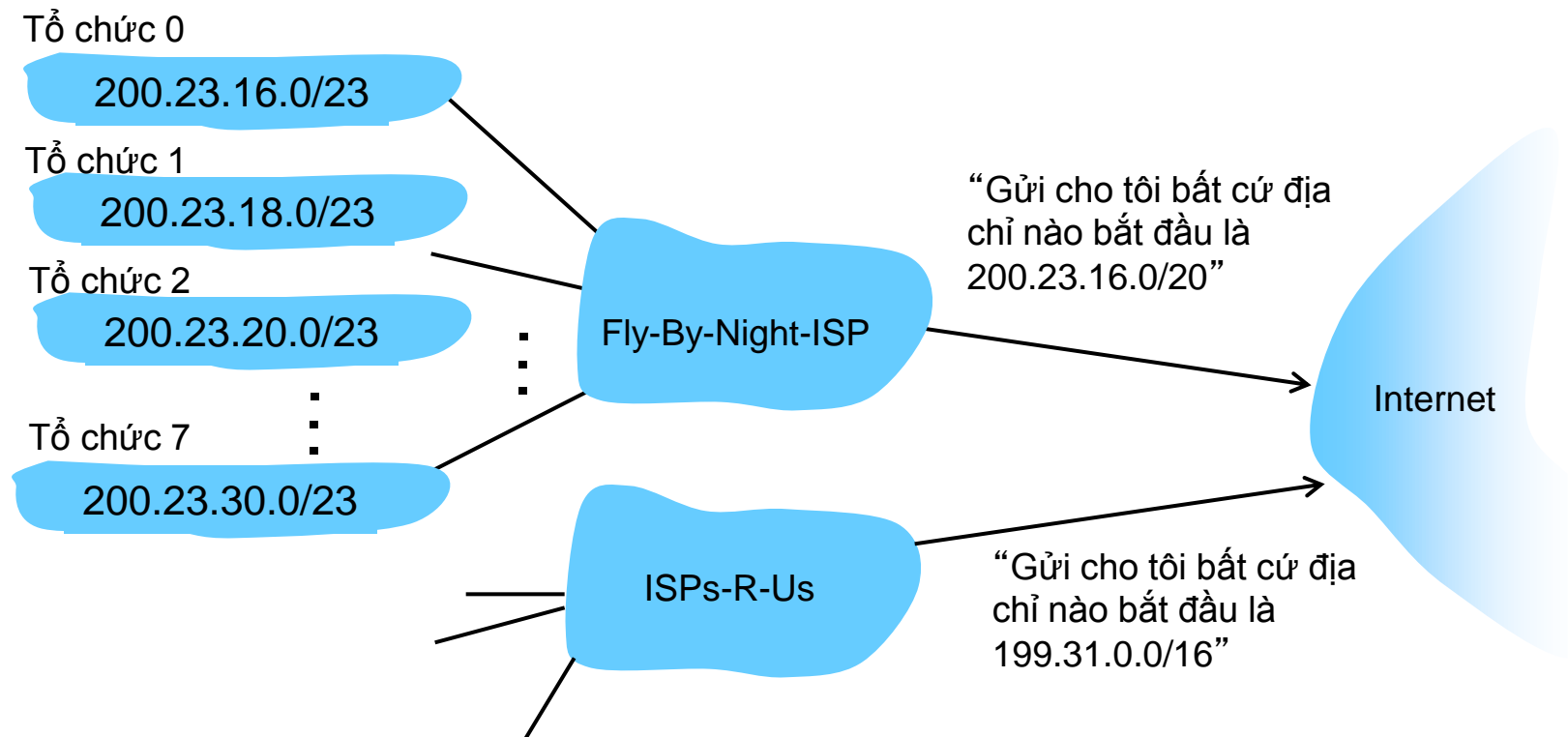
*Hỏi:* Làm thế nào để mạng có được phần subnet của địa chỉ IP?

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

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

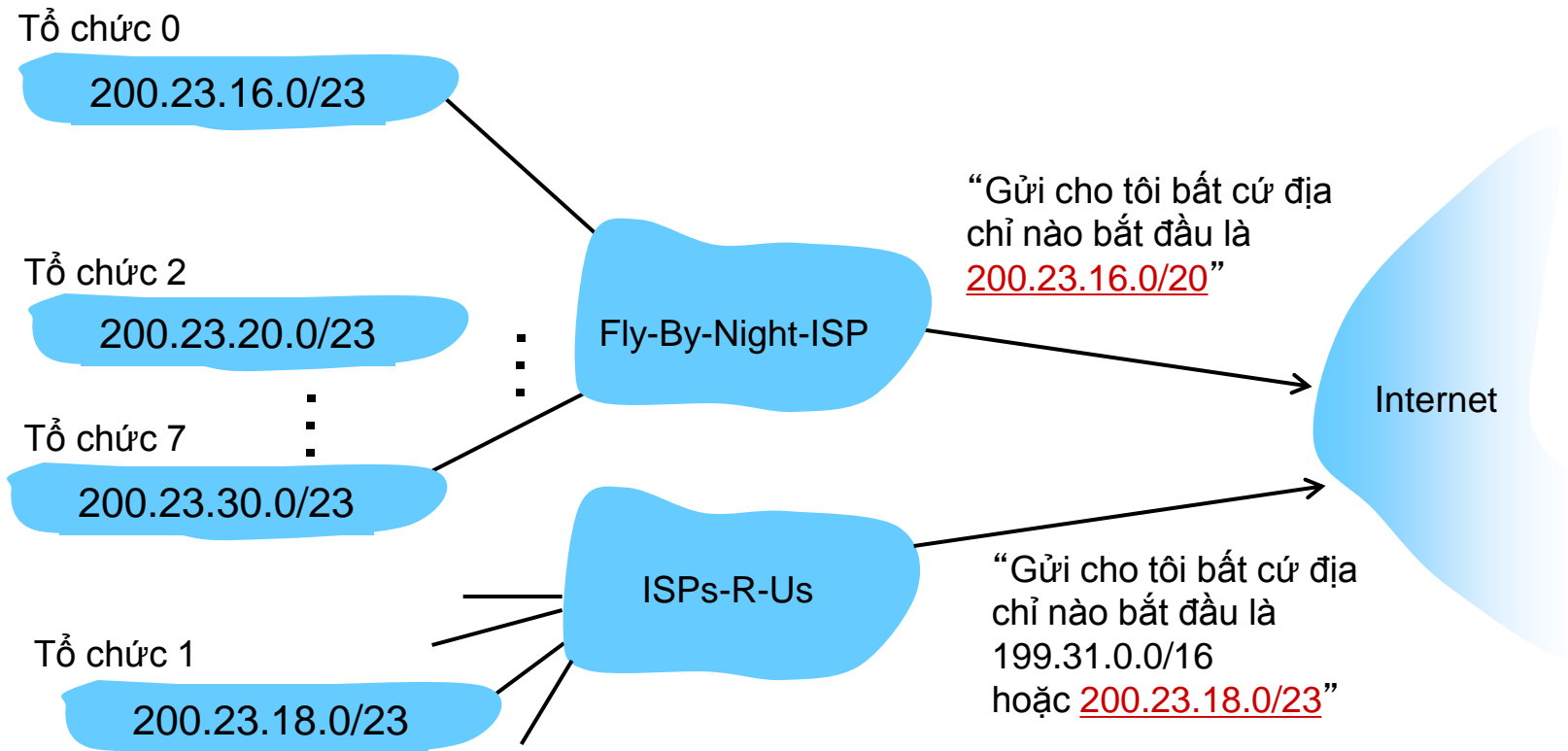
# Định địa chỉ phân cấp: tích hợp định tuyến

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



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

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



# Định địa chỉ IP...

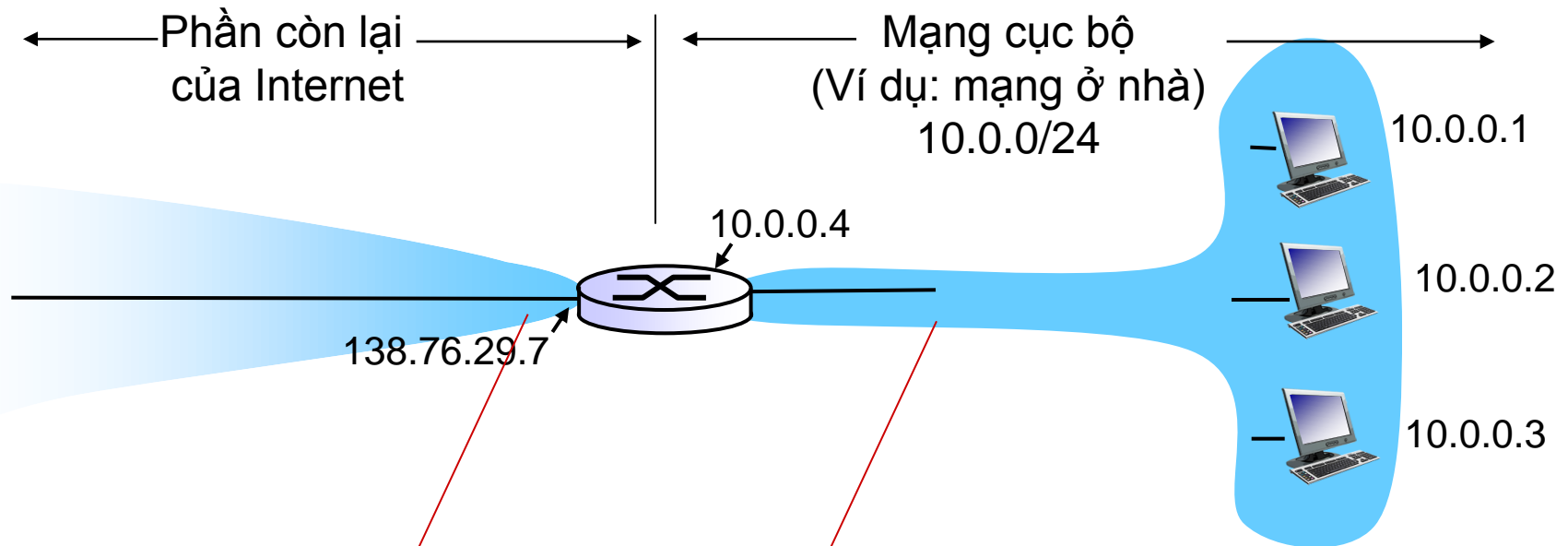
*Hỏi:* Làm thế nào một ISP có thể lấy được khối địa chỉ?

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

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



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



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

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

# NAT: network address translation

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

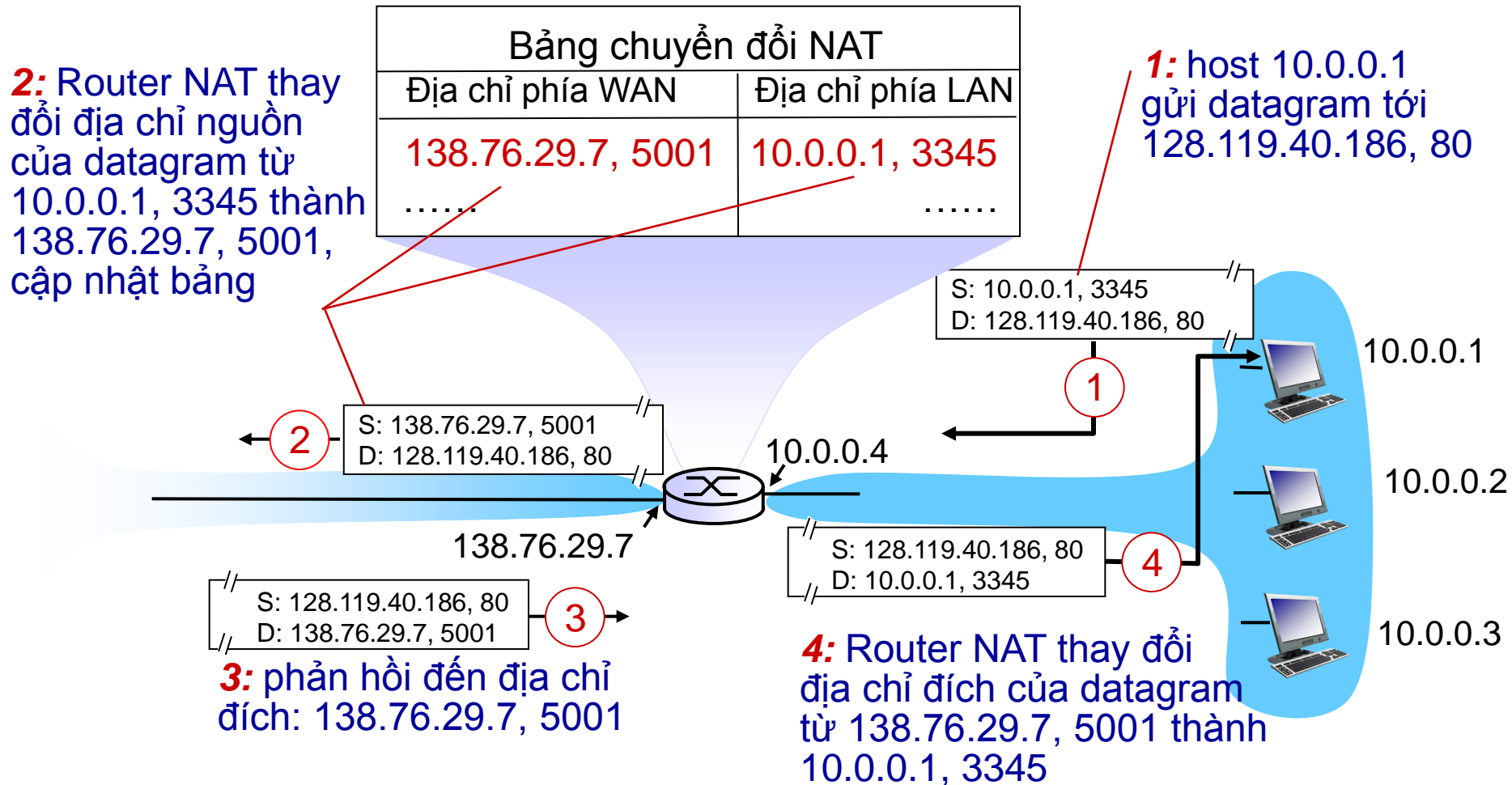
- Không cần thiết sử dụng cả dãy địa chỉ từ một ISP: chỉ cần một địa chỉ cho tất cả các dịch vụ
- Có thể thay đổi địa chỉ của dịch vụ trong mạng cục bộ mà không cần thông báo với hệ thống mạng bên ngoài.
- Có thể thay đổi ISP mà không cần thay đổi địa chỉ của các dịch vụ bên trong mạng cục bộ
- Hệ thống 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



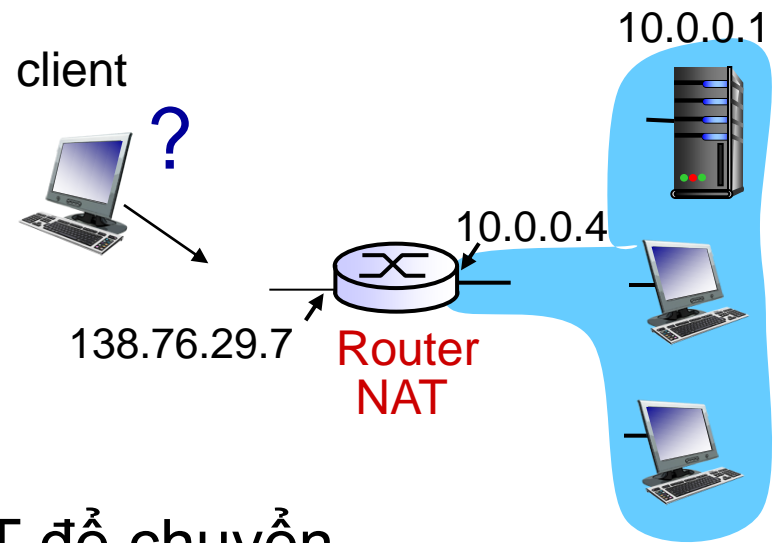
# NAT: network address translation

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

# Vấn đề đi qua NAT

- ❖ Client muốn kết nối tới server có địa chỉ 10.0.0.1

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



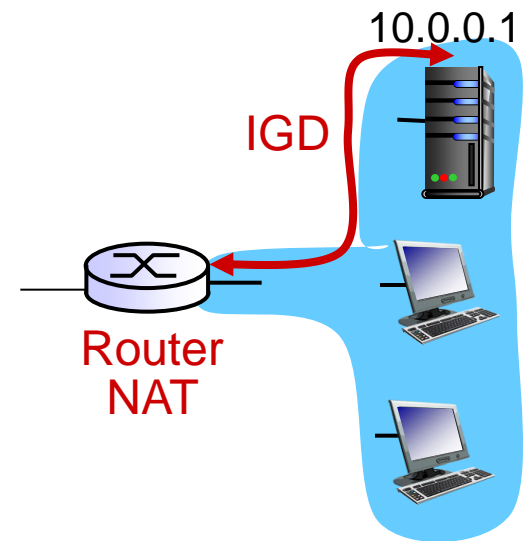
- ❖ **Giải pháp 1:** Cấu hình tĩnh NAT để chuyển tiếp các yêu cầu kết nối đến tới cổng đã xác định của server
  - Ví dụ: (138.76.29.7, cổng 2500) sẽ luôn được chuyển tiếp tới (10.0.0.1, cổng 25000)

# 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



# Vấn đề đi qua NAT

## ❖ *Giải pháp 3:* chuyển tiếp (được dùng trong Skype)

- Client NAT thiết lập kết nối để chuyển tiếp
- Client bên ngoài kết nối để chuyển tiếp
- Chuyển tiếp giữa các gói tin của các cầu để kết nối





# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

# ICMP: internet control message protocol

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

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

- ❖ “Ở phía trên” trong tầng mạng:

- Các thông điệp ICMP được mang trong các IP datagram

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

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

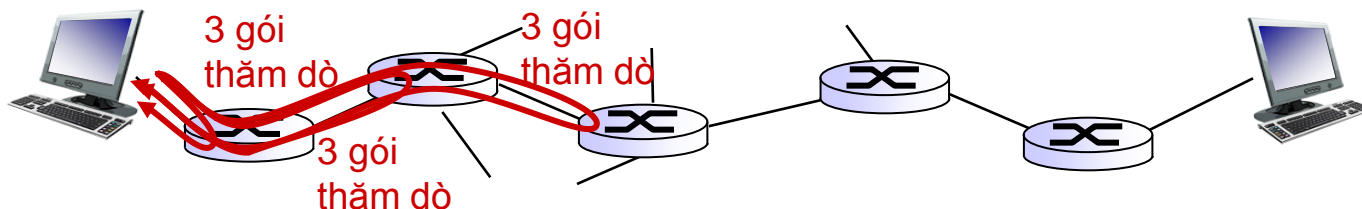
# Traceroute và ICMP

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

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

## *Điều kiện dừng:*

- ❖ UDP segment cuối cùng đến được host đích.
- ❖ Đích trả lại thông điệp ICMP “port unreachable” (type 3, code 3) → cổng không có
- ❖ Nguồn dừng lại



# IPv6: Lý do

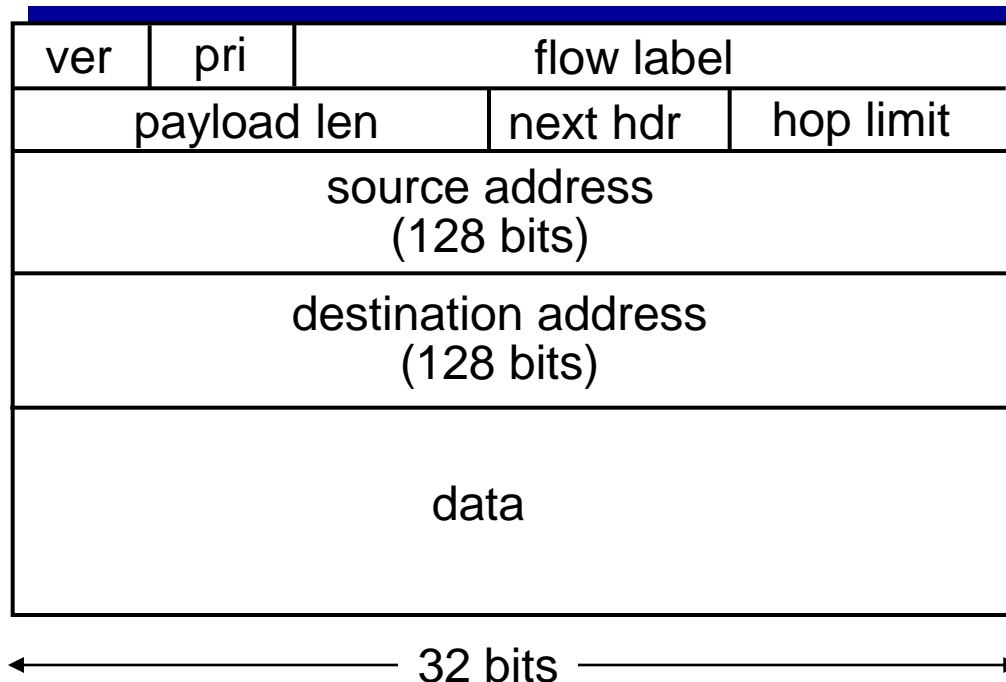
- ❖ *Động lực thúc đẩy ban đầu:* không gian địa chỉ 32-bit sắp được cấp phát hết.
- ❖ Động lực bổ sung:
  - Định dạng tiêu đề (header) giúp tăng tốc độ xử lý/chuyển tiếp
  - Tiêu đề thay đổi giúp tạo điều kiện cho QoS

## *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

- ❖ *Priority (ưu tiên)*: xác định ưu tiên giữa các datagram trong luồng
- ❖ *Flow Label (nhãn luồng)*: xác định các datagram trong cùng một “luồng”.
- ❖ *Next header*: xác định giao thức tầng cao hơn cho dữ liệu

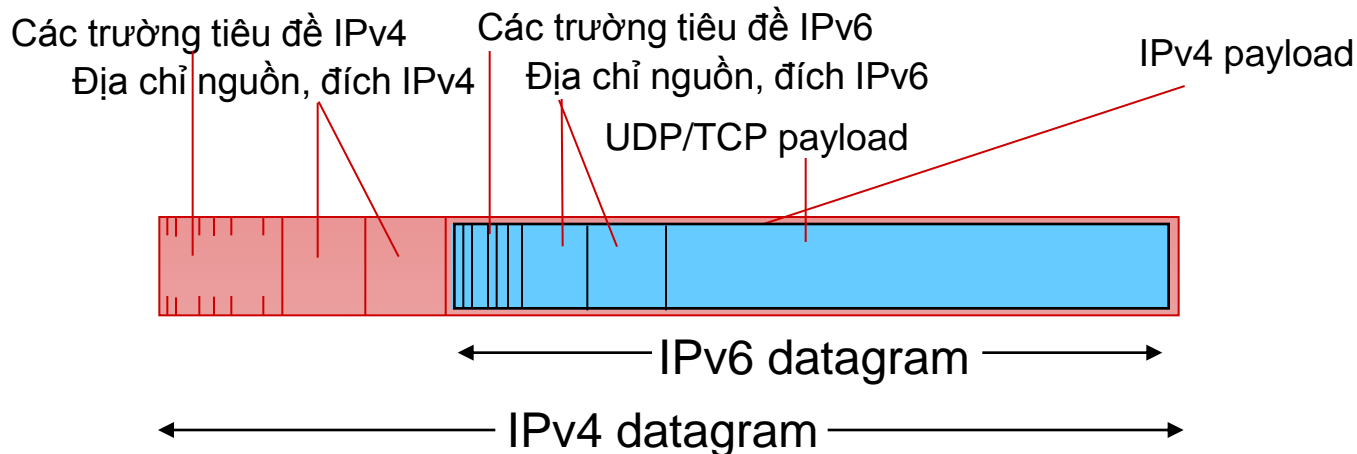


# Những thay đổi của IPv6 so với IPv4

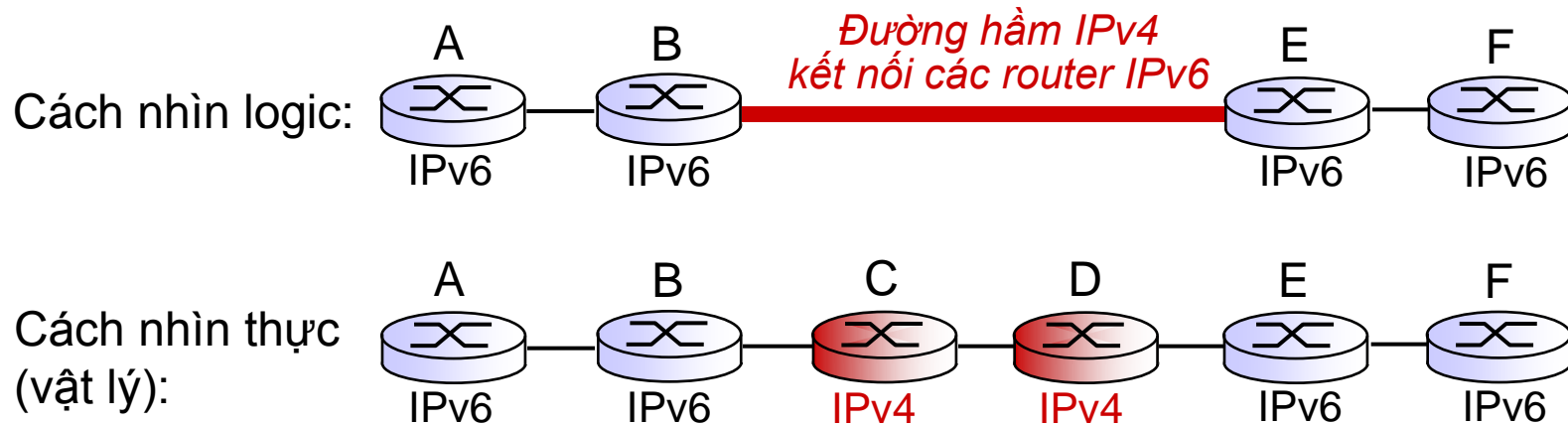
- ❖ *Checksum*: bỏ hoàn toàn, nhằm giảm thời gian xử lý tại mỗi hop
- ❖ *Options*: được phép, nhưng nằm ngoài phần tiêu đề, được xác định trong trường “Next Header”
- ❖ *ICMPv6*: phiên bản mới của ICMP
  - Các loại thông điệp bổ sung, ví dụ: “Packet Too Big”
  - Các chức năng quản lý nhóm multicast

# 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

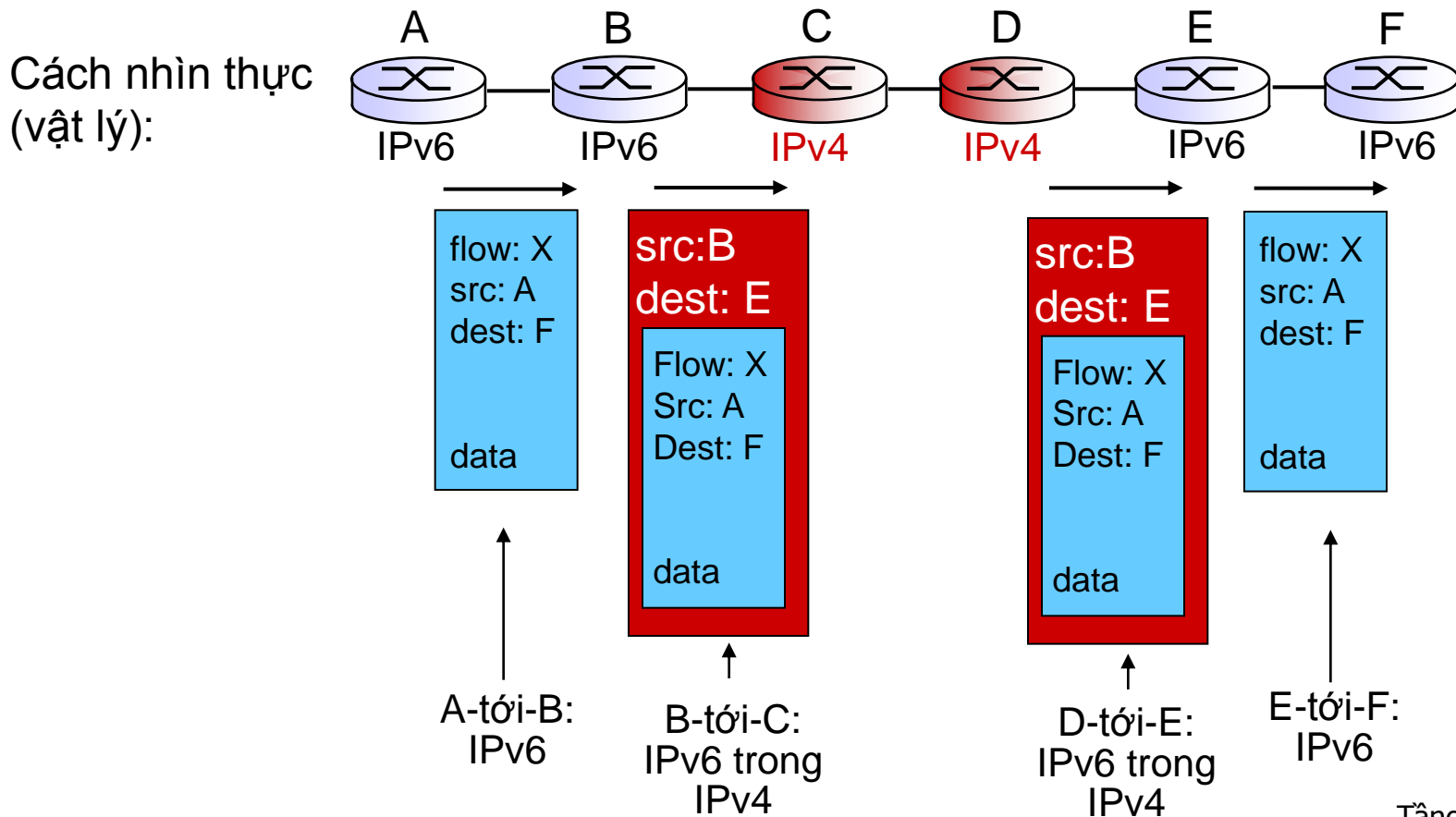


# Tunneling





# Tunneling



# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

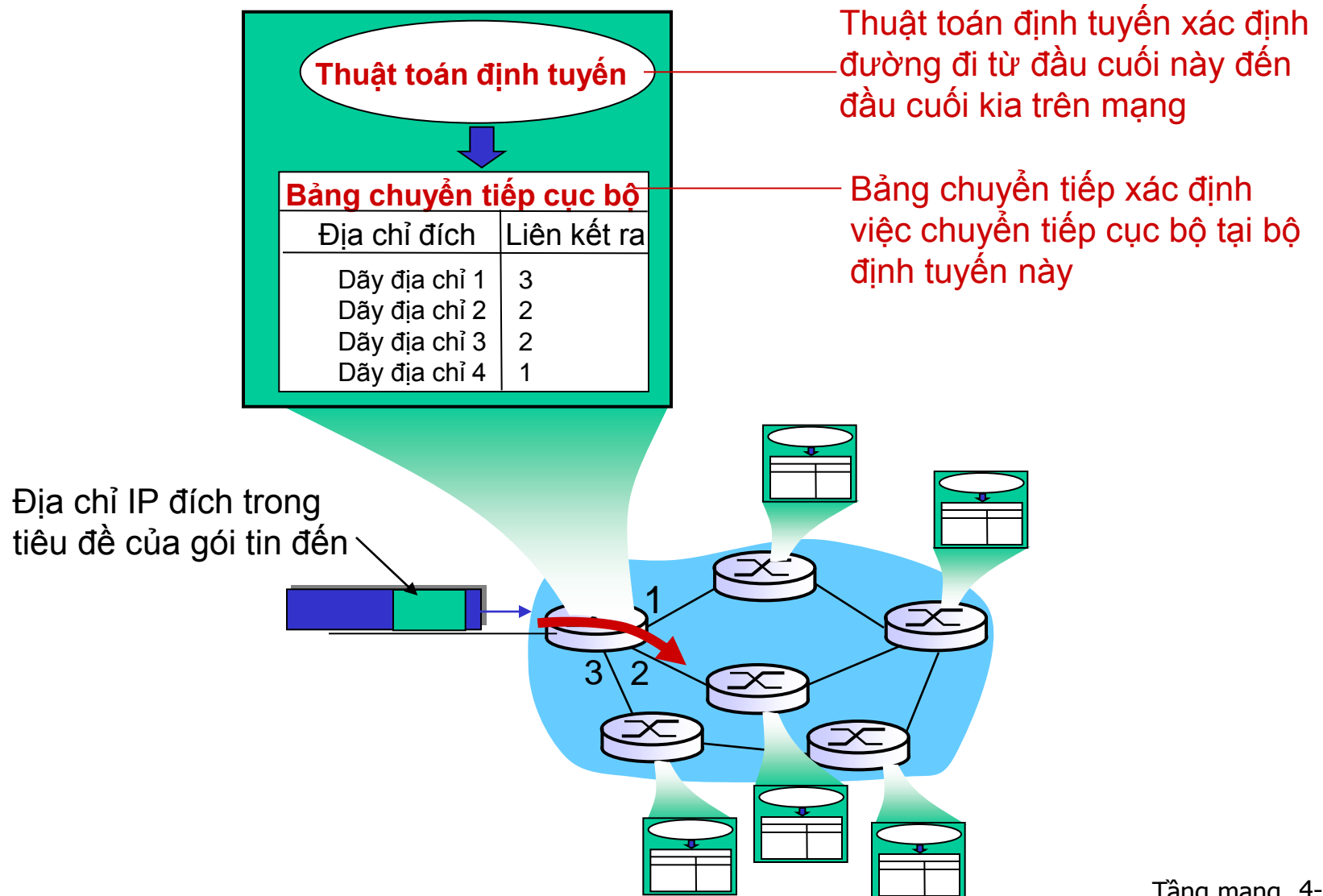
## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

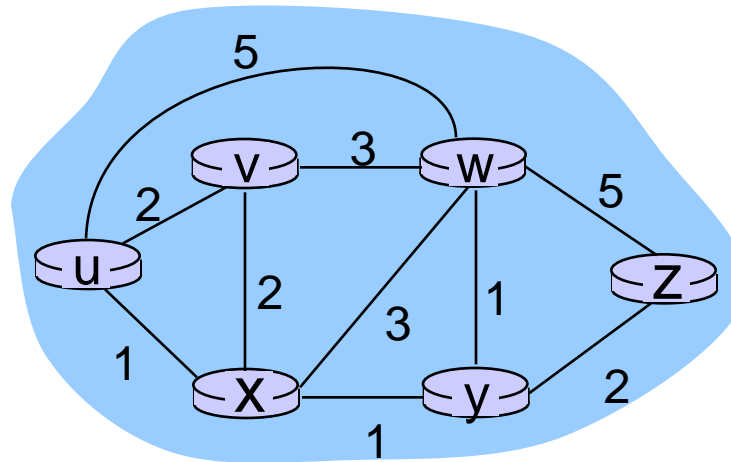
## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

# Tác động qua lại giữa định tuyến và chuyển tiếp



# Mô hình đồ thị



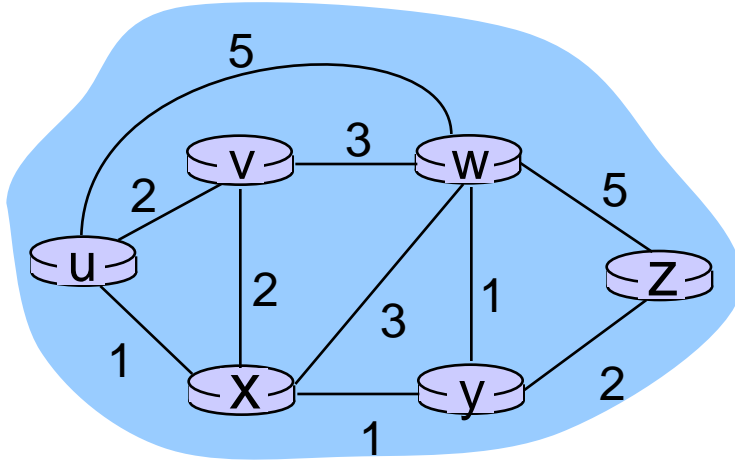
Đồ thị:  $G = (N, E)$

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

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

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

# 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

# Phân loại giải thuật định tuyến

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

*Tập trung:*

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

*Không tập trung:*

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

*Hỏi: Động hay tĩnh?*

*Tĩnh:*

- ❖ Việc định tuyến thay đổi chậm theo thời gian

*Động:*

- ❖ Việc định tuyến thay đổi nhanh hơn:
  - Cập nhật định kỳ
  - Phản ứng với những thay đổi chi phí liên kết

# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

# Một giải thuật định tuyến Link-State

## *Giải thuật Dijkstra*

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

## *Ký hiệu:*

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



# 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  $v$  \*/

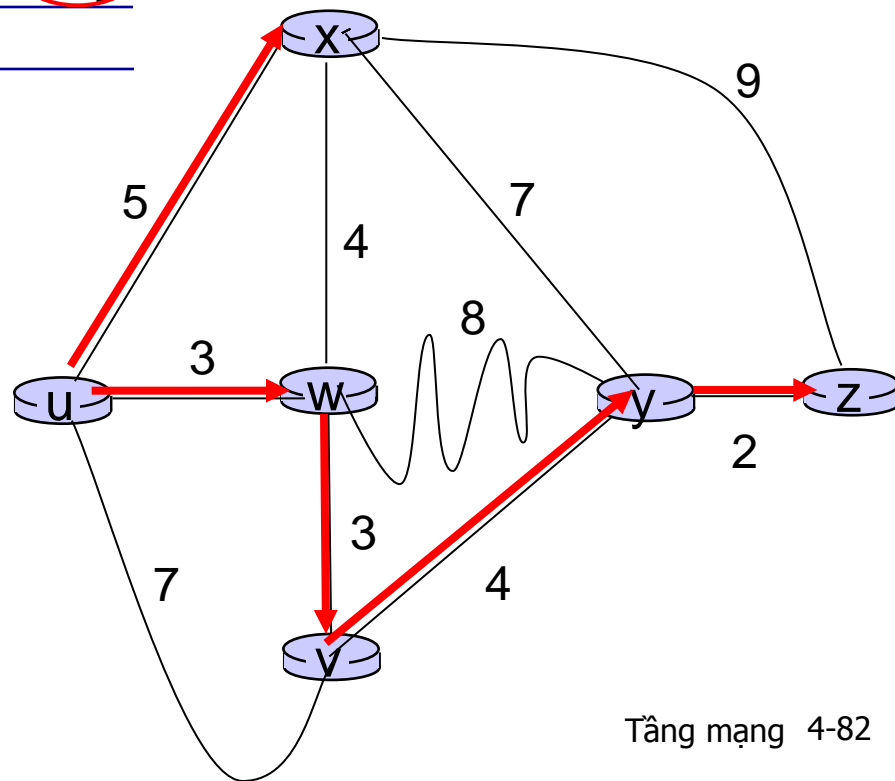
15 **cho đến khi tất cả các nút nằm trong  $N'$**

# 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	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					

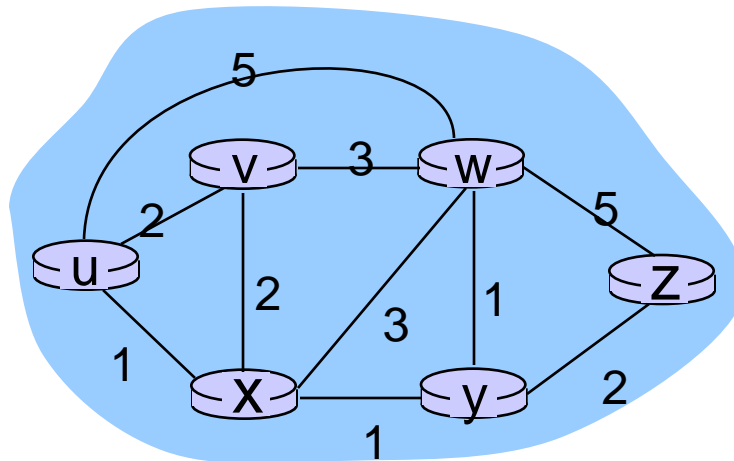
## Chú ý:

- ❖ 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)



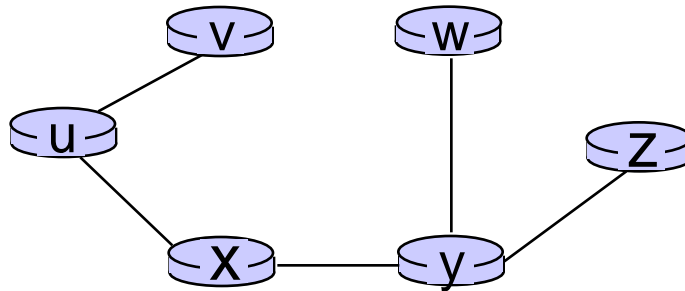
# Giải thuật Dijkstra: Một ví dụ khác

Bước	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	$\infty$	$\infty$
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					



# Giải thuật Dijkstra: Ví dụ (2)

Kết quả cây đường đi ngắn nhất từ u:



Kết quả bảng chuyển tiếp trong u:

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

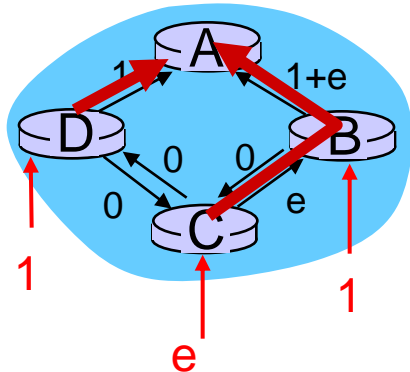
# Giải thuật Dijkstra: Thảo luận

**Độ phức tạp của giải thuật:**  $n \log n$

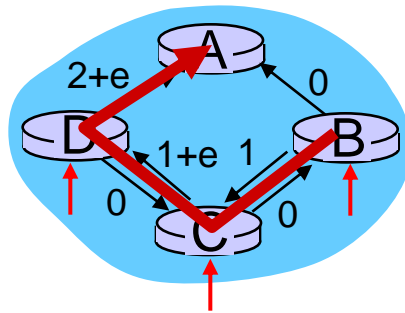
- ❖ Mỗi lần duyệt: cần kiểm tra lại tất cả các nút w không thuộc N
- ❖  $n(n+1)/2$  phép so sánh:  $O(n^2)$
- ❖ Có thể cài đặt hiệu quả hơn:  $O(n \log n)$

**Có thể dao động:**

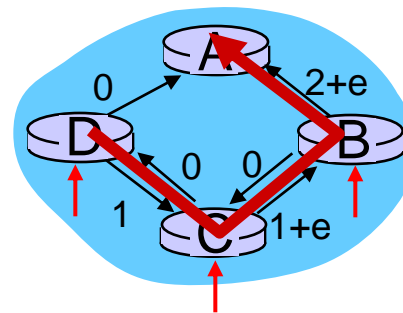
- ❖ Ví dụ: chi phí kết nối bằng tổng lưu lượng mang:



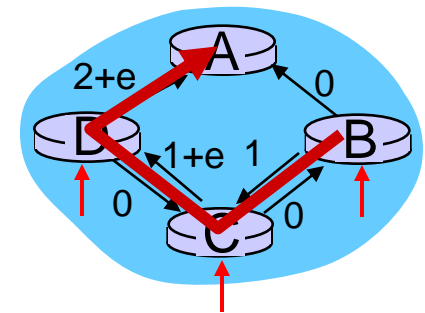
## Khởi tạo



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



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



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

# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

# Giải thuật distance vector

## *Công thức Bellman-Ford*

Cho

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

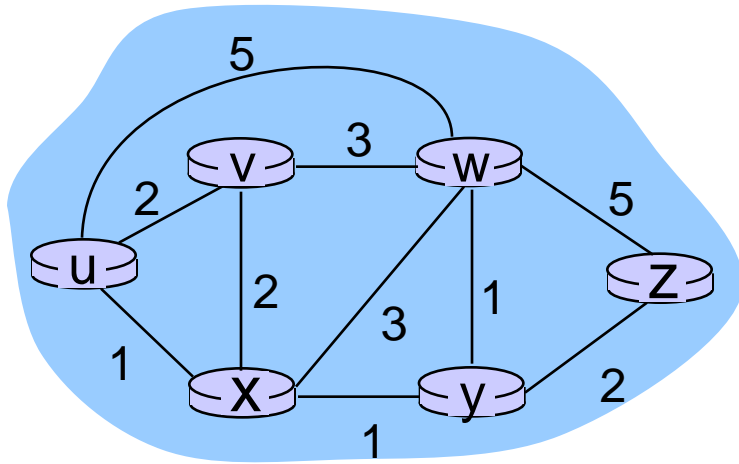
$$d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$$

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

# Ví dụ Bellman-Ford



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

Công thức B-F cho:

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

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



# Giải thuật 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  $\mathbf{D}_v = [D_v(y): y \in N]$

# Giải thuật distance vector

## *Ý tưởng chính:*

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

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

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

# Giải thuật distance vector

**Lặp, không đồng bộ:** mỗi lần lặp cục bộ có nguyên nhân từ:

- ❖ Thay đổi chi phí liên kết cục bộ
- ❖ Thông báo cập nhật DV từ nút lân cận

**Phân bố:**

- ❖ Mỗi nút *chỉ* thông báo đến các nút lân cận khi DV của nó thay đổi
  - Các lân cận sau đó sẽ thông báo đến các lân cận khác của nó nếu cần

**Mỗi nút:**

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

**Tính toán lại** các ước lượng

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

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

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

**Bảng nút x**

		Chi phí tới		
		x	y	z
từ	x	0	2	7
	y	$\infty$	$\infty$	$\infty$
	z	$\infty$	$\infty$	$\infty$

Chi phí tới

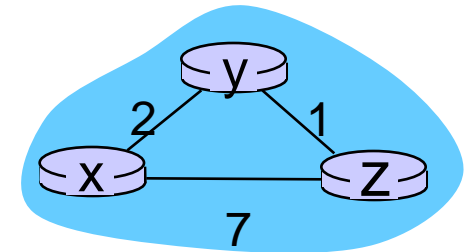
		x	y	z
từ	x	0	2	3
	y	2	0	1
	z	7	1	0

**Bảng nút y**

		Chi phí tới		
		x	y	z
từ	x	$\infty$	$\infty$	$\infty$
	y	2	0	1
	z	$\infty$	$\infty$	$\infty$

**Bảng nút z**

		Chi phí tới		
		x	y	z
từ	x	$\infty$	$\infty$	$\infty$
	y	$\infty$	$\infty$	$\infty$
	z	7	1	0



Thời gian

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

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

**Bảng nút x**

		Chi phí tới		
		x	y	z
từ	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

**Bảng nút y**

		Chi phí tới		
		x	y	z
từ	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

**Bảng nút z**

		Chi phí tới		
		x	y	z
từ	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

Chi phí tới

		x	y	z
từ	x	0	2	3
	y	2	0	1
	z	7	1	0

Chi phí tới

		x	y	z
từ	x	0	2	7
	y	2	0	1
	z	7	1	0

Chi phí tới

		x	y	z
từ	x	0	2	7
	y	2	0	1
	z	3	1	0

Chi phí tới

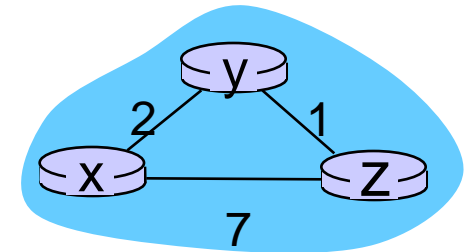
		x	y	z
từ	x	0	2	3
	y	2	0	1
	z	3	1	0

Chi phí tới

		x	y	z
từ	x	0	2	3
	y	2	0	1
	z	3	1	0

Chi phí tới

		x	y	z
từ	x	0	2	3
	y	2	0	1
	z	3	1	0

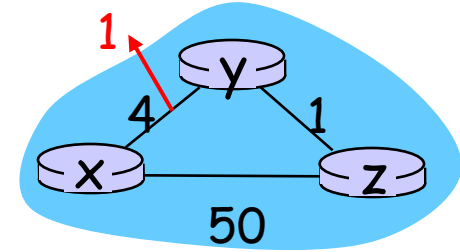


Thời gian

# Distance vector: Chi phí kết nối thay đổi

## Chi phí kết nối thay đổi:

- ❖ Nút kiểm tra thay đổi chi phí kết nối cục bộ
- ❖ Cập nhật thông tin tìm đường, tính toán lại véc-tơ khoảng cách
- ❖ Nếu DV thay đổi, thông báo đến các lân cận



“thông tin  
tốt truyền  
đi nhanh”

$t_0$ :  $y$  kiểm tra thay đổi chi phí kết nối, cập nhật lại DV của nó, và thông báo cho các lân cận.

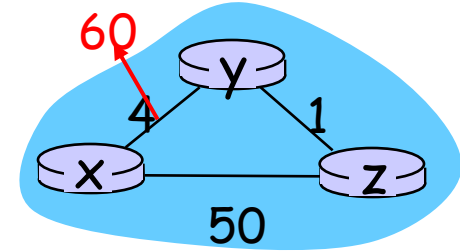
$t_1$ :  $z$  nhận thông tin cập nhật từ  $y$ , cập nhật lại bảng của nó, tính toán chi phí thấp nhất mới tới  $x$ , và gửi DV của nó tới các lân cận.

$t_2$ :  $y$  nhận cập nhật của  $z$ , cập nhật lại bảng khoảng cách của nó. Chi phí thấp nhất của  $y$  *không* thay đổi, nên  $y$  *không cần* gửi thông báo tới  $z$ .

# Distance vector: Chi phí kết nối thay đổi

## *Chi phí kết nối thay đổi:*

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



## *poisoned reverse:*

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

# So sánh các giải thuật LS và DV

## *Độ phức tạp thông báo*

- ❖ **LS:** với  $n$  nút,  $E$  liên kết,  $O(nE)$  thông báo được gửi đi
- ❖ **DV:** chỉ trao đổi giữa các nút lân cận
  - Thời gian hội tụ thay đổi

## *Tốc độ hội tụ*

- ❖ **LS:** Thuật toán  $O(n^2)$  yêu cầu  $O(nE)$  thông báo
  - Có thể có dao động
- ❖ **DV:** Thời gian hội tụ thay đổi
  - Có thể lập định tuyến
  - Vấn đề đếm vô hạn

**Sự linh hoạt:** điều gì sẽ xảy ra nếu router hoạt động sai chức năng?

### **LS:**

- Nút có thể thông báo chi phí *kết nối* không chính xác
- Mỗi nút chỉ tính toán bảng *riêng* của nó

### **DV:**

- Nút có thể thông báo chi phí *đường đi* không chính xác
- Mỗi bảng của nút được dùng bởi các nút khác
  - Lỗi lan truyền thông qua mạng



# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Định tuyến trong mạng Internet

- RIP
- OSPF
- BGP

# Hierarchical routing (Định tuyến phân cấp)

Những vấn đề định tuyến được học cho đến lúc này là với môi trường lý tưởng hóa

- ❖ Tất cả các bộ định tuyến là đồng nhất
  - ❖ Mạng “phẳng”
- ... *không* đúng trong thực tế!

*Quy mô:* với 600 triệu

đích:

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

*Tự quản*

- ❖ Internet = mạng của các mạng
- ❖ Mỗi nhà quản trị mạng có thể muốn điều hành định tuyến riêng trong mạng của họ

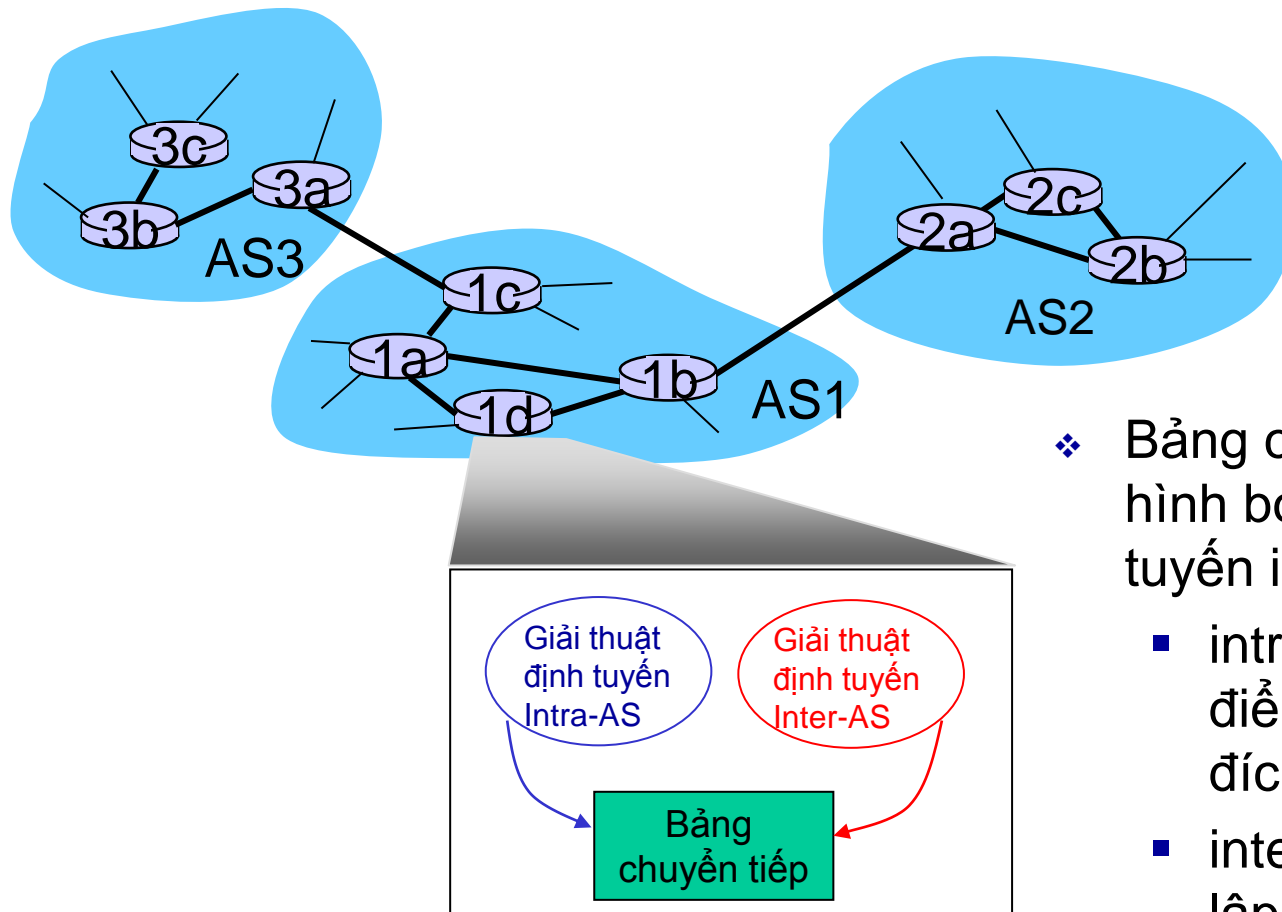
# Hierarchical routing

- ❖ Các router được tập hợp lại thành các vùng, “**hệ thống tự trị**” (**autonomous systems - AS**)
- ❖ Các router trong cùng AS sẽ chạy cùng giao thức định tuyến
  - Giao thức định tuyến “**nội vùng-AS**” (**intra-AS**)
  - Các router trong các AS khác nhau có thể chạy các giao thức định tuyến intra-AS khác nhau

## *Gateway router:*

- ❖ Tại “cạnh” của AS riêng của nó
- ❖ Có liên kết tới router trong AS khác

# Kết nối các AS



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

# Nhiệm vụ của Inter-AS

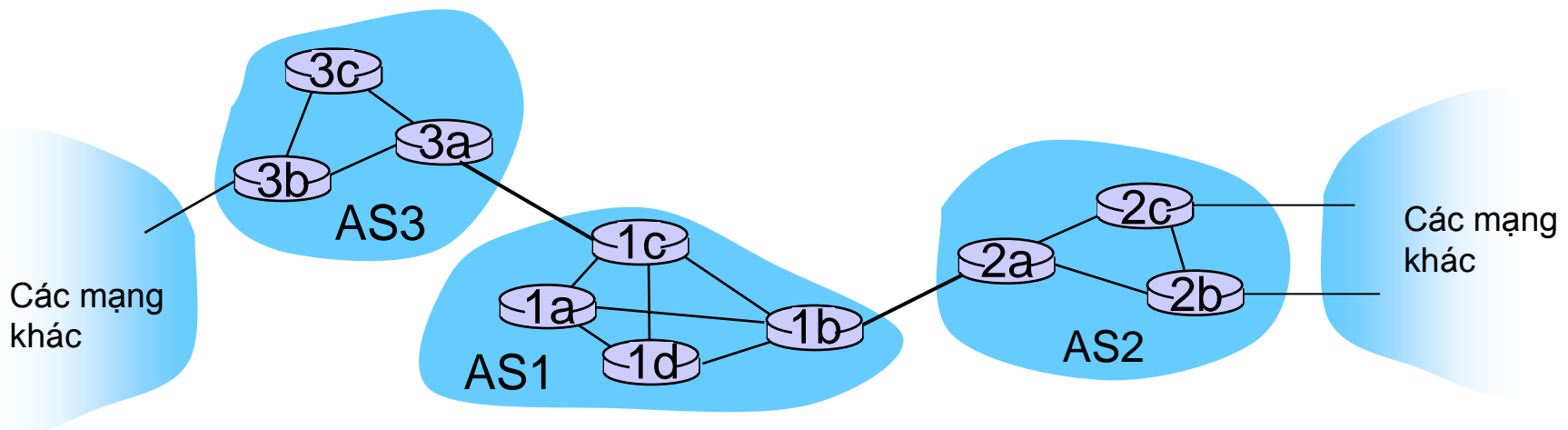
❖ Giả sử router trong AS1 nhận datagram có đích ở bên ngoài AS1:

- Router nên chuyển tiếp gói tin đến gateway router, nhưng mà là cái nào?

*AS1 phải:*

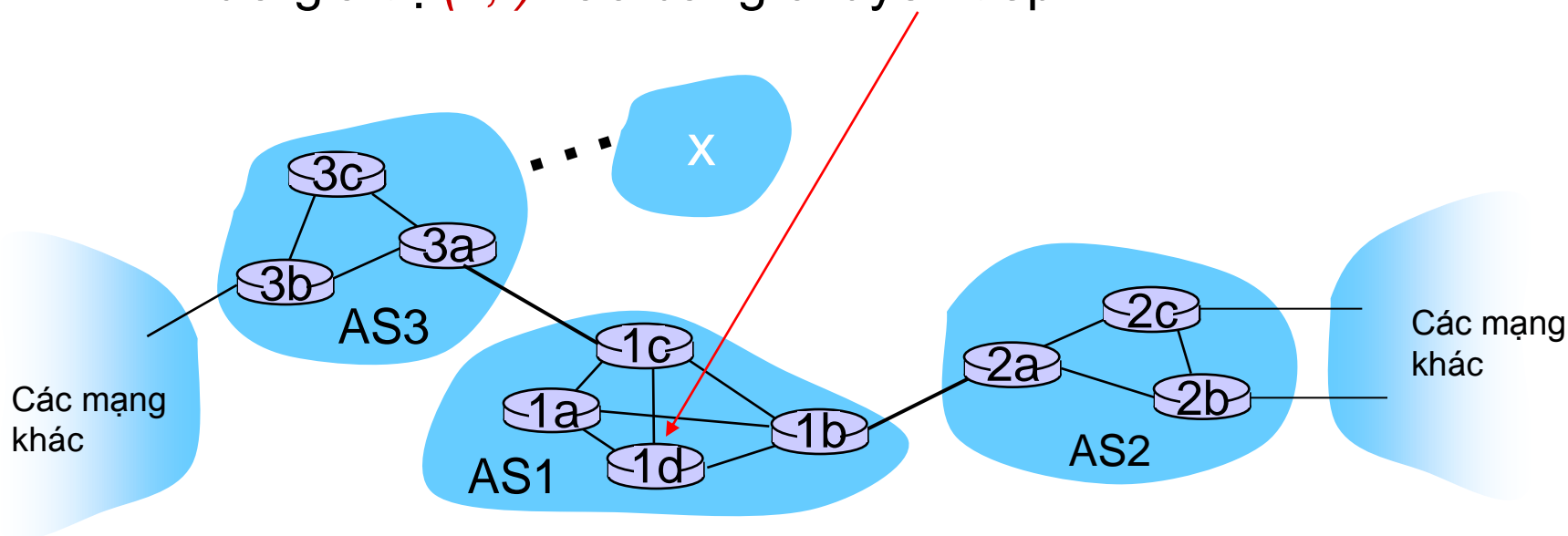
1. Học xem có thể đến được đích nào qua AS2, và AS3
2. Lan truyền thông tin này đến tất cả các router trong AS1

*Đây là nhiệm vụ của định tuyến inter-AS!*



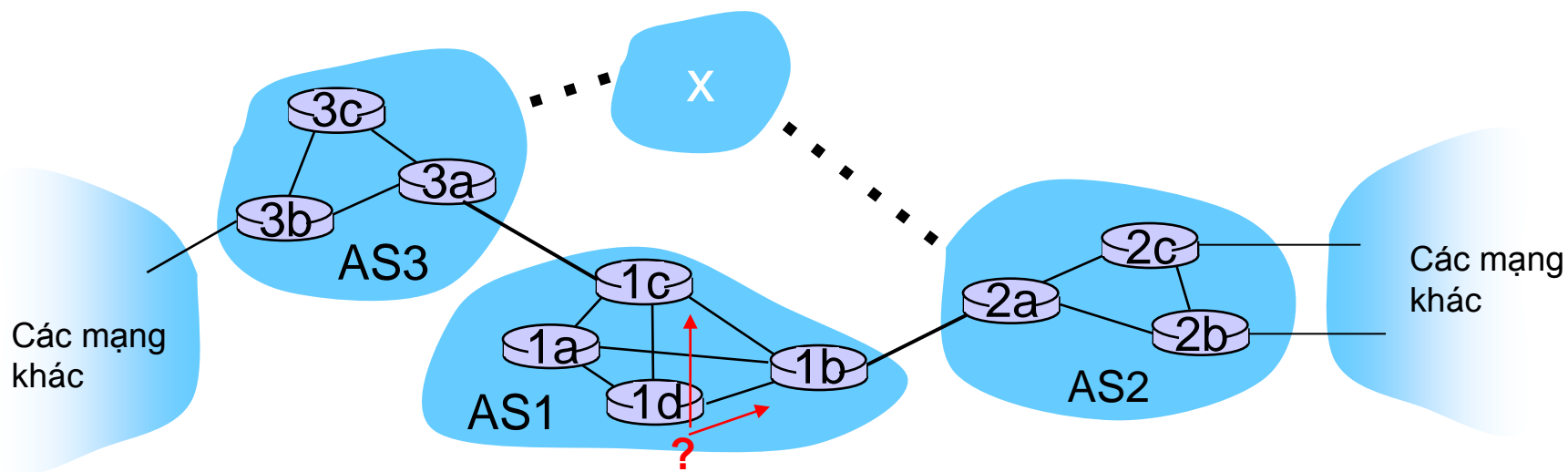
## Ví dụ: thiết lập bảng chuyển tiếp trong router 1d

- ❖ Giả sử AS1 học (qua giao thức inter-AS) được là subnet **x** có thể đến được qua AS3 (gateway 1c), nhưng không qua AS2
  - Giao thức inter-AS lan truyền thông tin đi được cho tất cả các router nội mạng
- ❖ Router 1d biết được từ thông tin định tuyến intra-AS là giao diện **/** của nó thuộc đường đi có chi phí thấp nhất tới 1c
  - Đưa giá trị **(x, /)** vào bảng chuyển tiếp



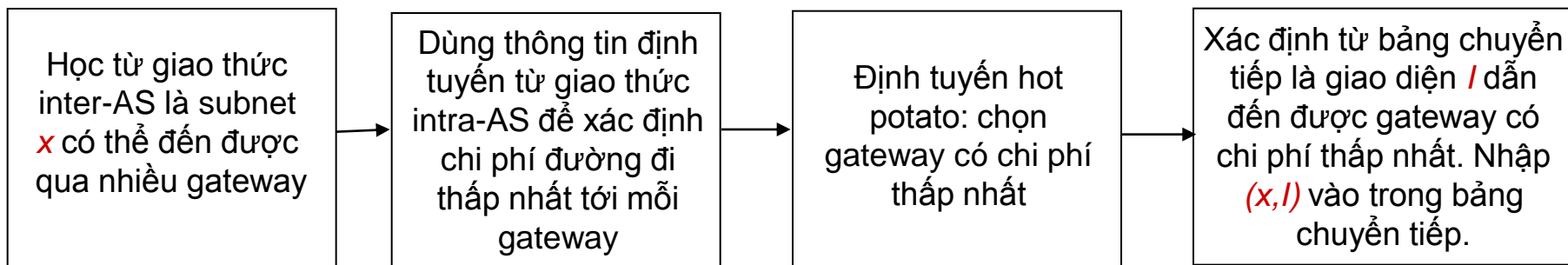
# Ví dụ: lựa chọn giữa nhiều AS

- ❖ Bây giờ, giả sử AS1 học từ giao thức inter-AS là subnet **x** có thể đến được từ AS3 và từ AS2.
- ❖ Để cấu hình bảng chuyển tiếp, router 1d cần phải xác định gateway nào mà nó nên chuyển tiếp các gói tin đến để tới được đích **x**
  - Đây là nhiệm vụ của giao thức định tuyến inter-AS!



# Ví dụ: lựa chọn giữa nhiều AS

- ❖ Bây giờ, giả sử AS1 học từ giao thức inter-AS là subnet **x** có thể đến được từ AS3 và từ AS2.
- ❖ Để cấu hình bảng chuyển tiếp, router 1d cần phải xác định gateway nào mà nó nên chuyển tiếp các gói tin đến để tới được đích **x**
  - Đây là nhiệm vụ của giao thức định tuyến inter-AS!
- ❖ **Định tuyến hot potato**: **gửi** gói tin đến router gần nhất trong hai router





# Chương 4: Nội dung

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của bộ định tuyến

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Định tuyến trong mạng Internet

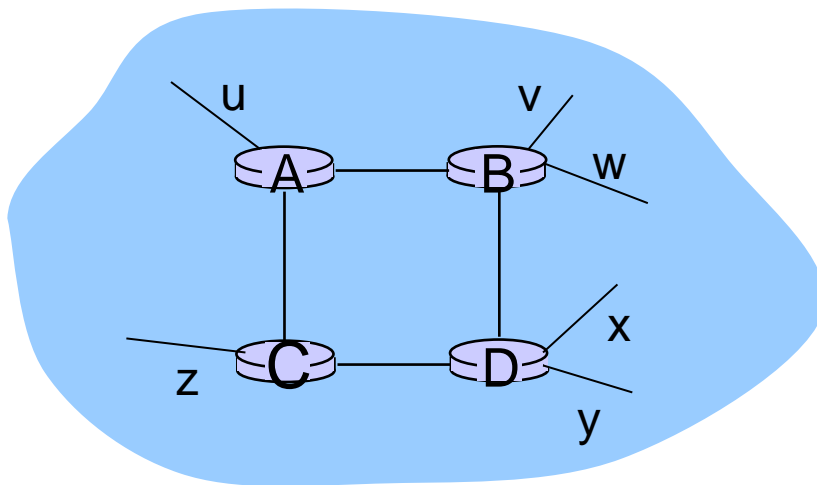
- RIP
- OSPF
- BGP

# Định tuyến Intra-AS

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

# RIP (Routing Information Protocol)

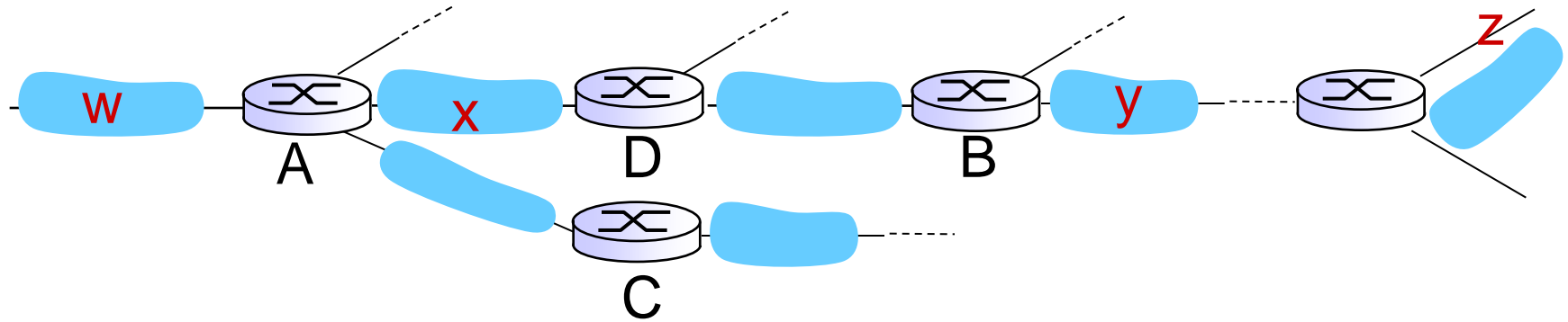
- ❖ Được công bố trong BSD-UNIX distribution năm 1982
- ❖ Giải thuật distance vector
  - Độ đo khoảng cách: số hop (lớn nhất = 15 hop), mỗi liên kết có chi phí là 1
  - Các DV được trao đổi giữa các điểm lân cận sau mỗi 30s bằng một thông điệp đáp ứng (còn được gọi là **thông báo (advertisement)**)
  - Mỗi thông báo: danh sách lên đến 25 **subnet** đích



Từ router A đến các **subnet** đích:

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

# RIP: Ví dụ



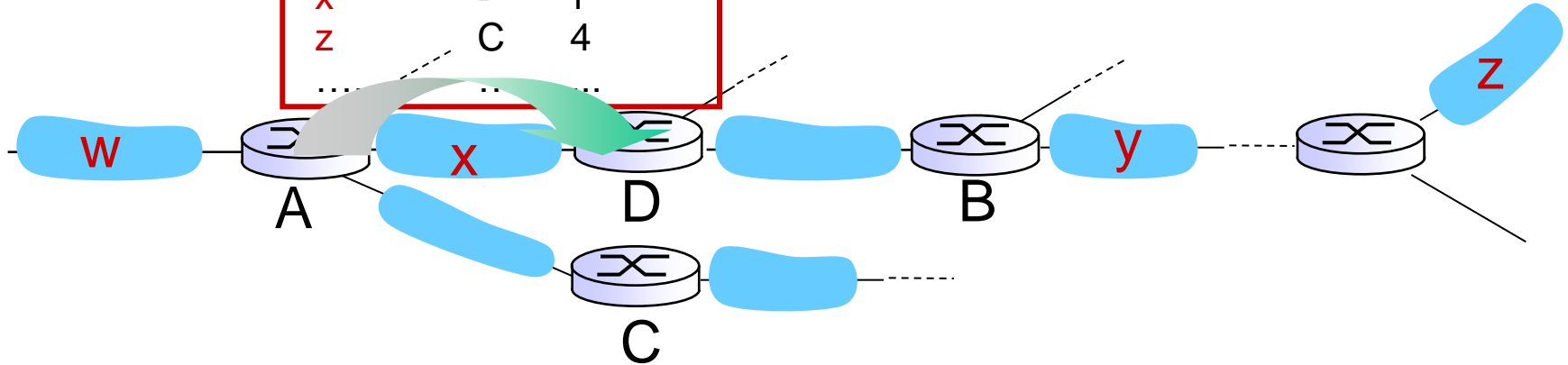
Bảng định tuyến trong router D

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

# RIP: Ví dụ

Thông báo từ A tới D

đích	kế tiếp	hop
w	-	1
x	-	1
z	C	4
...	...	...



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
....	....	....

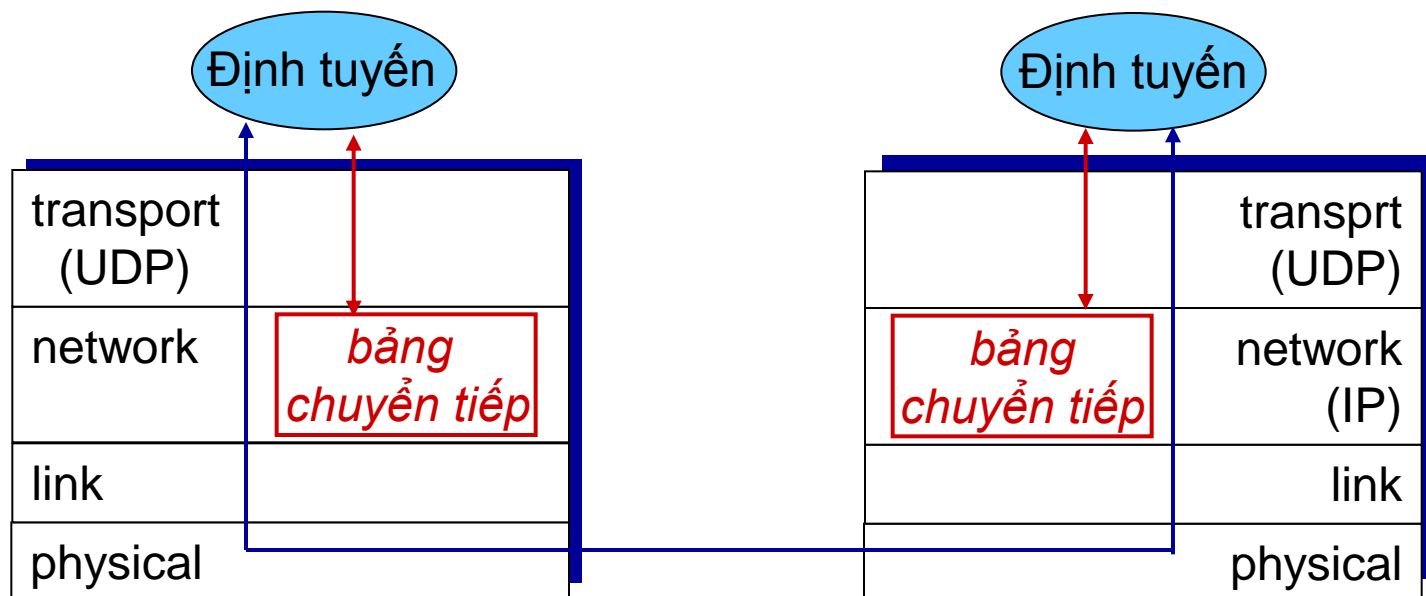
# RIP: Lỗi liên kết và khôi phục

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)

# 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



# 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

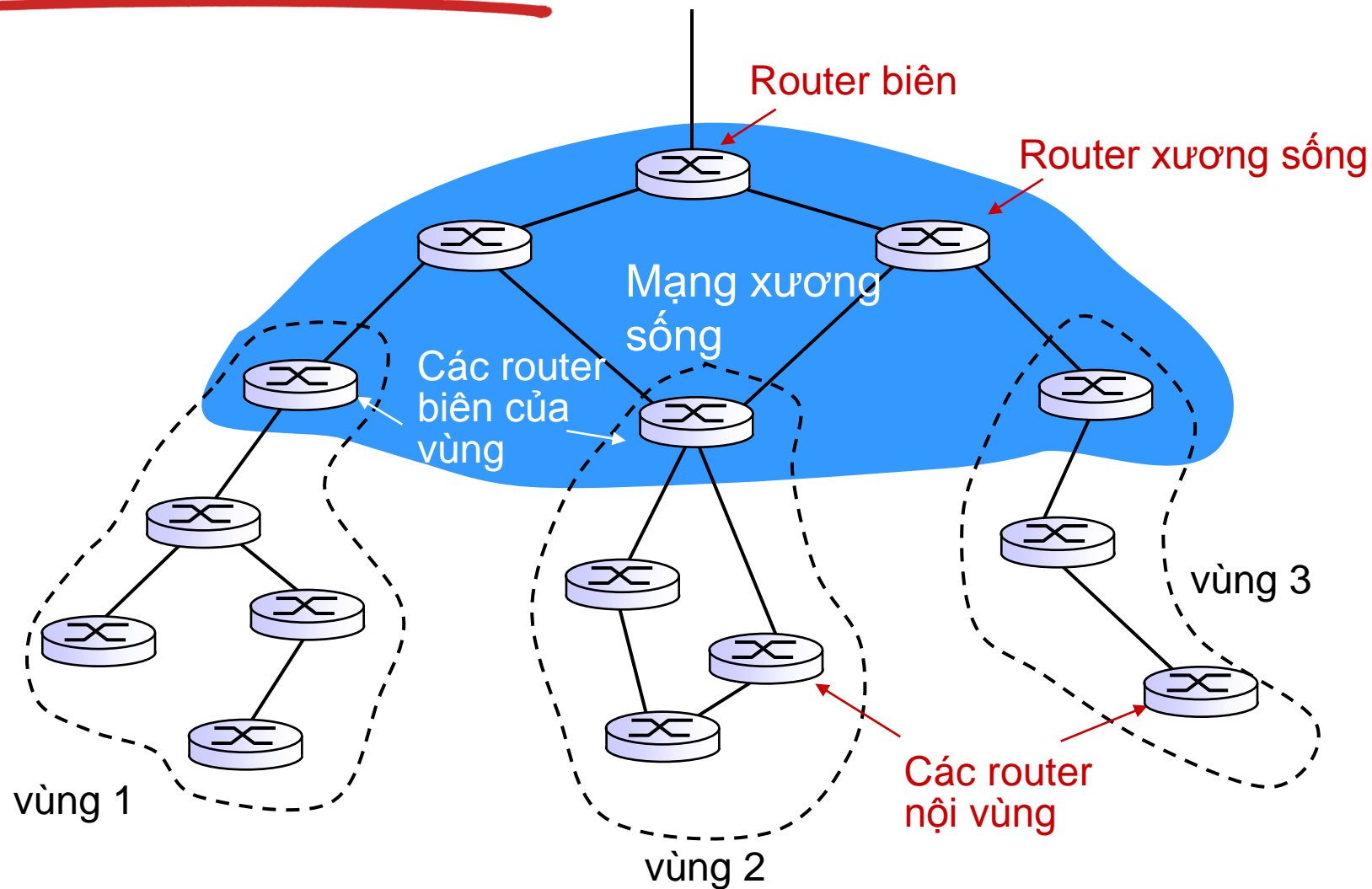


## Các đặc tính “cải tiến” trong OSPF (không có trong RIP)

---

- ❖ **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.

# OSPF phân cấp



# 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.

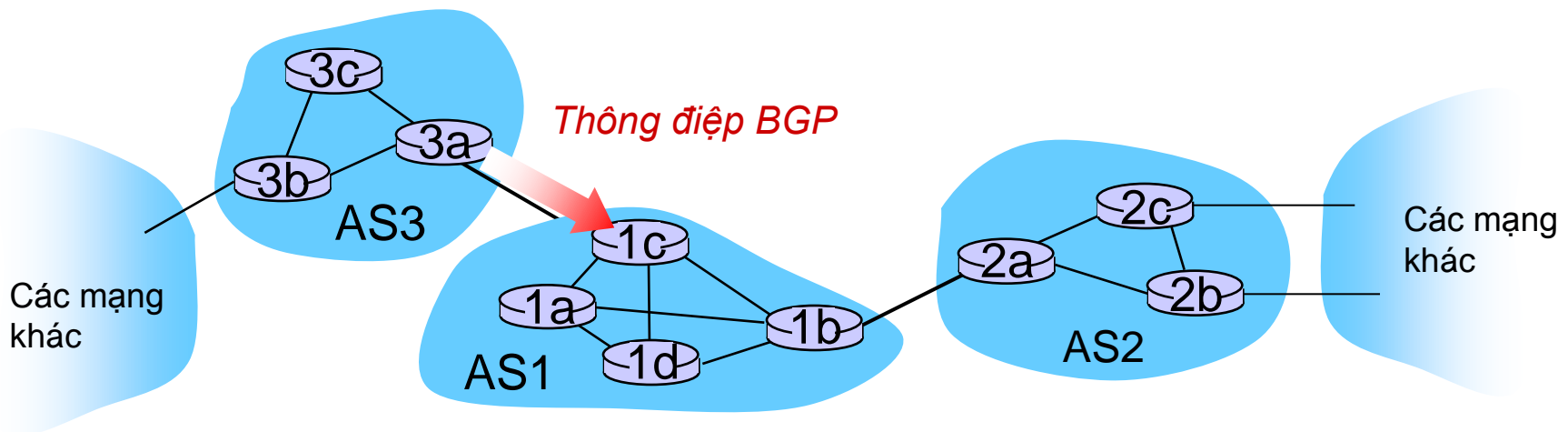
# Định tuyến inter-AS trên Internet: BGP

---

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

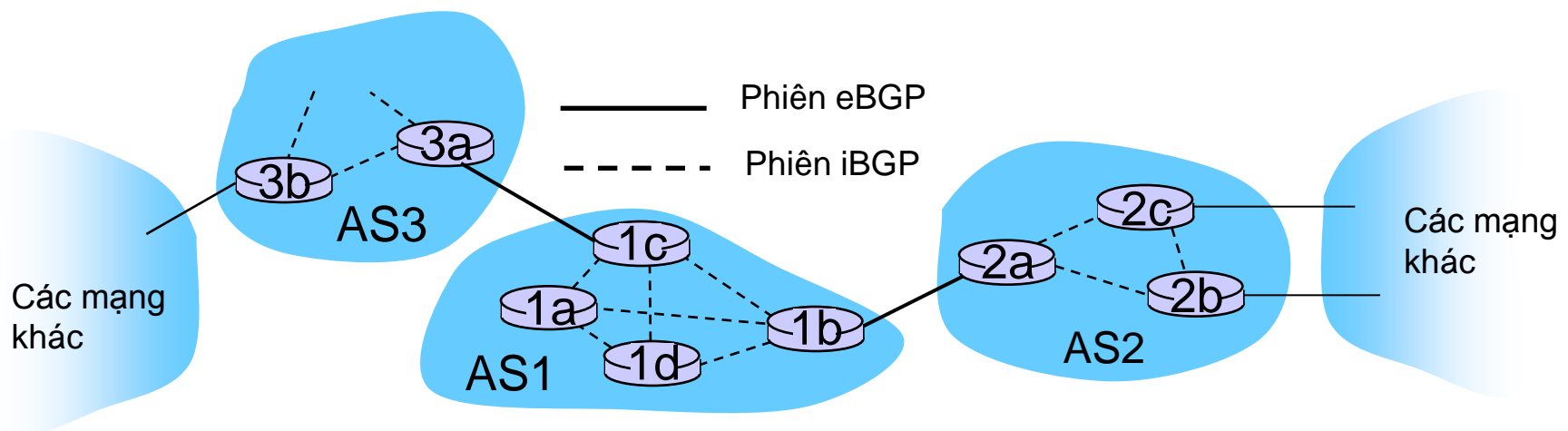
# Các cơ sở của BGP

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



# 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.



# 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*.

# Chọn định tuyến BGP

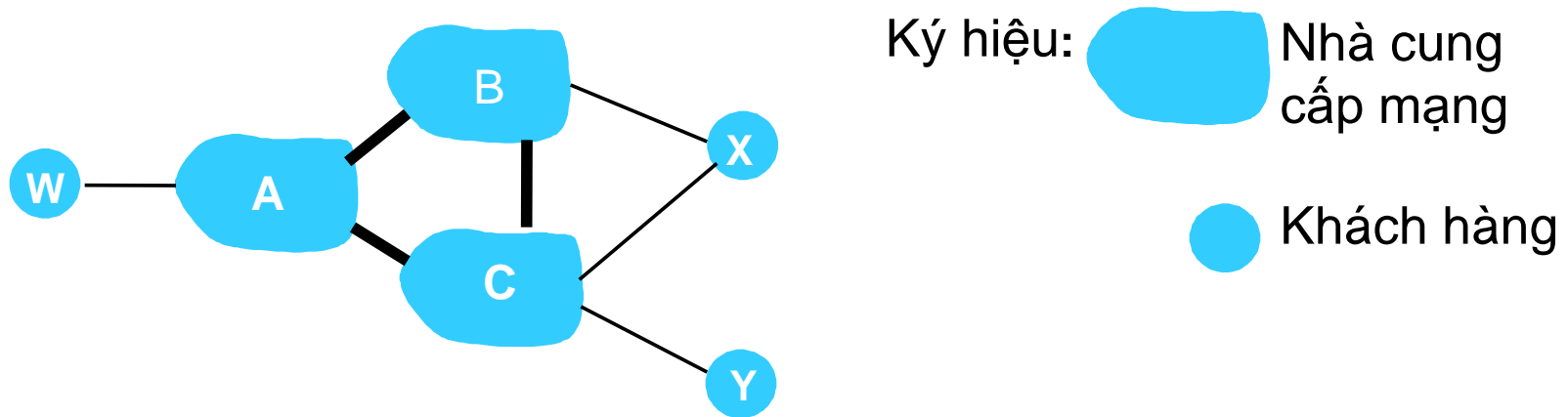
- ❖ Router có thể học được nhiều đường đi đến AS đích, và việc chọn tuyến đường được dựa trên:
  1. Thuộc tính giá trị ưu tiên cục bộ: quyết định chính sách
  2. AS-PATH ngắn nhất
  3. Router NEXT-HOP gần nhất: định tuyến hot potato
  4. Tiêu chuẩn bổ sung



# Các thông điệp BGP

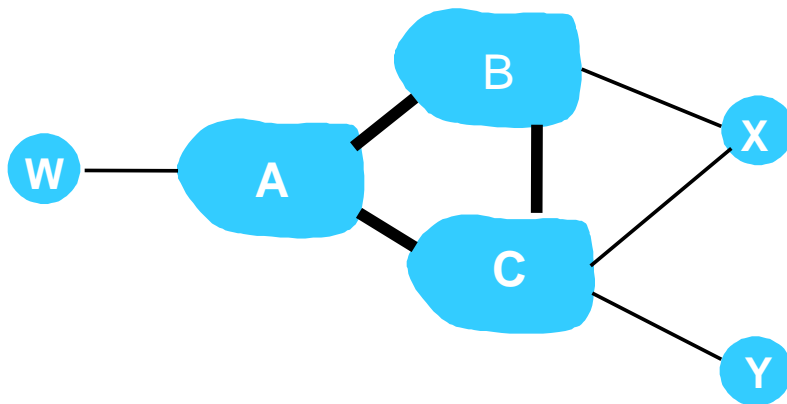
- ❖ Các thông điệp BGP được trao đổi giữa các peer qua kết nối TCP
- ❖ Các thông điệp BGP:
  - **OPEN:** Mở kết nối TCP tới peer và xác thực bên gửi
  - **UPDATE:** thông báo đường đi mới (hoặc xóa bỏ đường cũ)
  - **KEEPALIVE:** giữ kết nối tồn tại khi UPDATES thiếu; cũng có thể yêu cầu ACKs OPEN
 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

# 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

# Chính sách định tuyến BGP



Ký hiệu:  Nhà cung cấp mạng

 Khách hàng

- ❖ 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!

# 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

# Chương 4: *Hoàn thành!*

---

## 4.1 Giới thiệu

## 4.2 Các mạng mạch ảo và mạng chuyển gói

## 4.3 Kiến trúc của router

## 4.4 IP: Internet Protocol

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

## 4.5 Các giải thuật định tuyến

- link state, distance vector, hierarchical routing

## 4.6 Định tuyến trên Internet

- RIP, OSPF, BGP

- ❖ Hiểu được các nguyên lý bên trong các dịch vụ tầng mạng:
  - Các mô hình dịch vụ tầng mạng, tác động qua lại giữa định tuyến và chuyển tiếp, cách router hoạt động, định tuyến (chọn đường).
- ❖ Cài đặt, hiện thực trên mạng Internet