

# Chương 4 Tầng Mạng (Network layer)

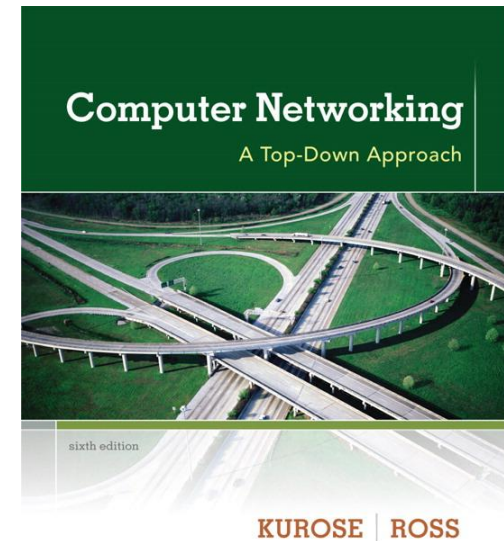
## A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- ❖ If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- ❖ If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

© 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 các nguyên lý nền tảng của các định vụ tầng Mạng:
  - Các mô hình dịch vụ tầng Mạng
  - Forwarding so với routing
  - Cách mà router hoạt động
  - Routing (chọn đường)
  - Broadcast, multicast
- ❖ Hiện thực trong Internet

# Chương 4: Nội dung

## 4.1 Giới thiệu

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

4.3 Cấu trúc bên trong  
router

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

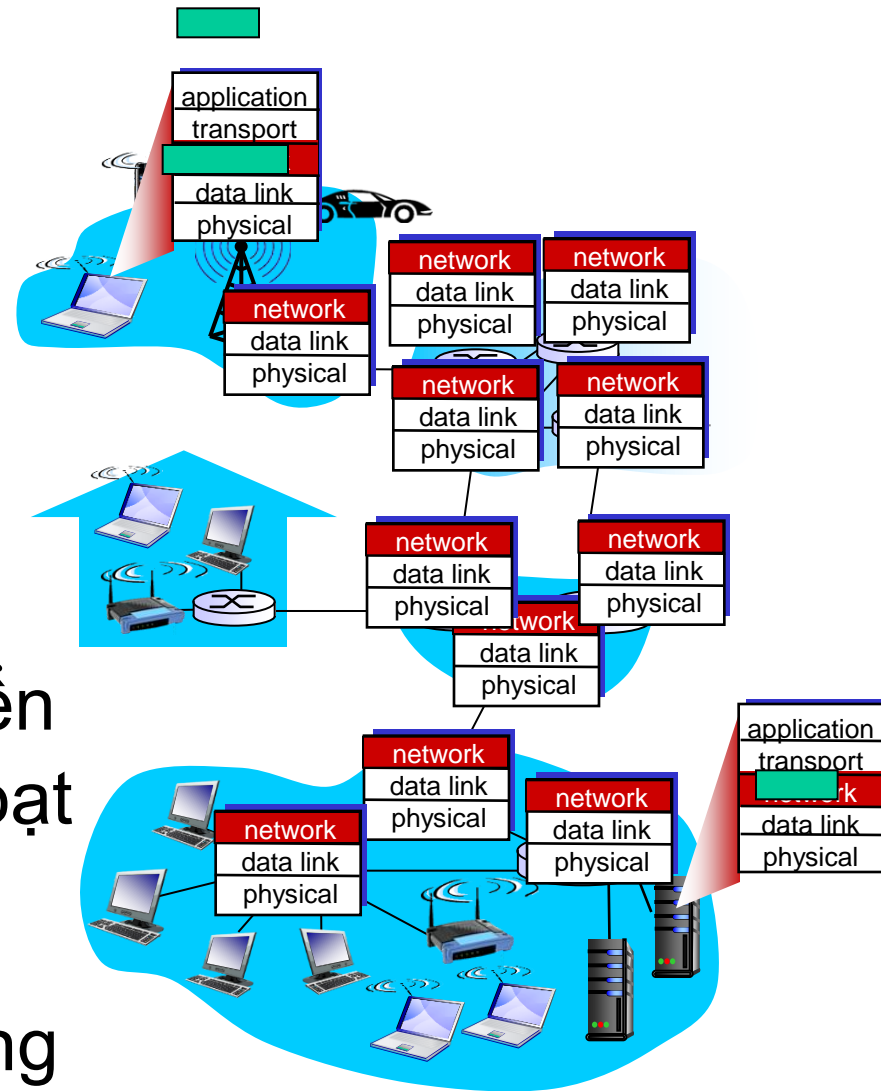
4.6 Routing trong Internet

- RIP
- OSPF
- BGP

4.7 Broadcast và multicast  
routing

# Tầng Mạng

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



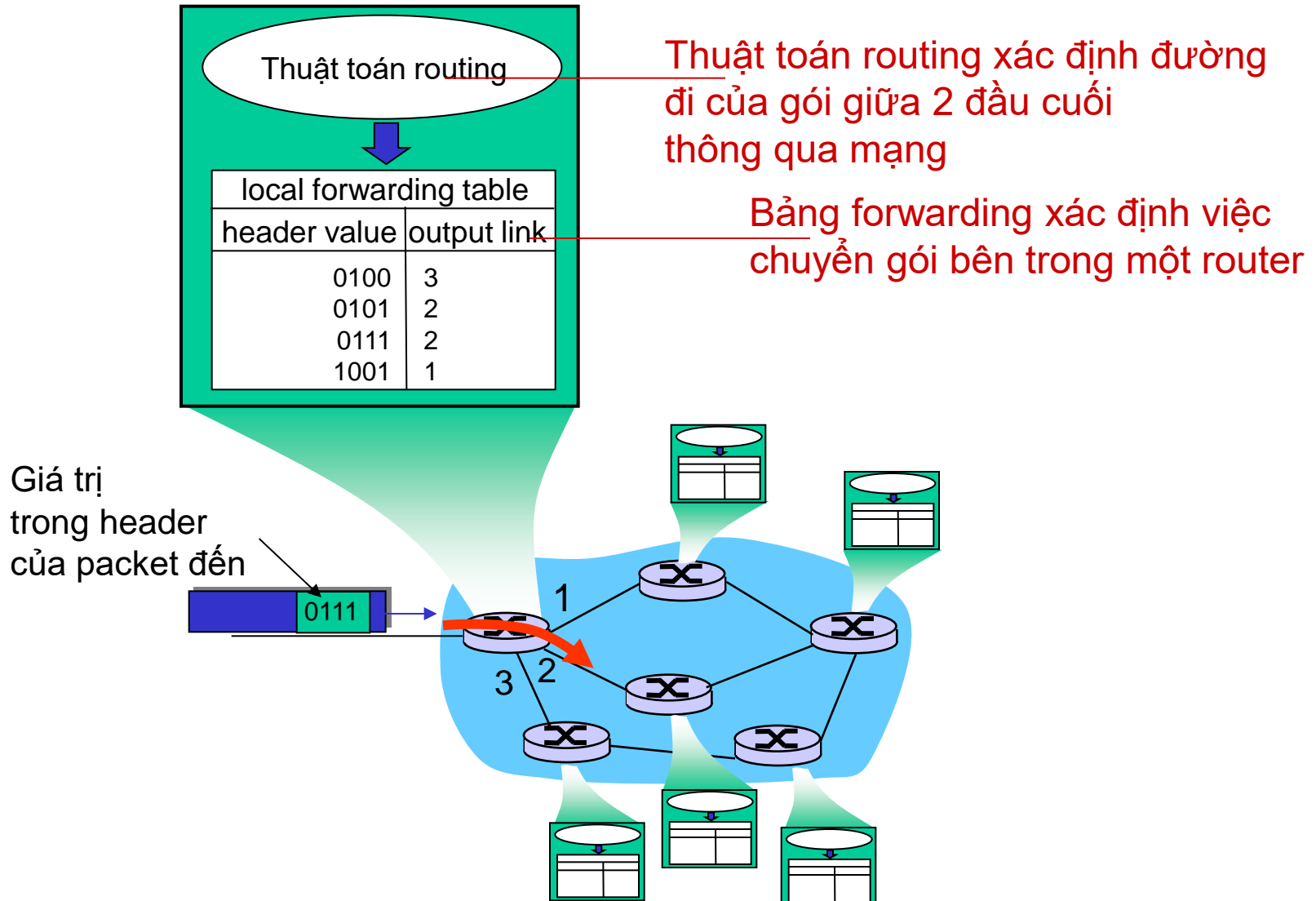
# Hai chức năng chính của tầng Mạng

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

## *Tương tự:*

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

# Tác động qua lại giữa routing và forwarding



# Thiết lập kết nối

- ❖ Chức năng quan trọng thứ 3 trong một số kiến trúc mạng:
  - ATM, frame relay, X.25
- ❖ Trước khi chuyển các datagram đi, 2 thiết bị đầu cuối và các router trung gian (intervening routers) thiết lập kết nối ảo (virtual connection)
  - Các router cũng tham gia
- ❖ So sánh giữa dịch vụ kết nối lớp Mạng và lớp Vận chuyển :
  - *Tầng Mạng*: giữa 2 hosts (cũng có thể bao gồm các router trung gian trong trường hợp kết nối ảo)
  - *Tầng Vận chuyển*: giữa 2 tiến trình

# Mô hình dịch vụ Mạng

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

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

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

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

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



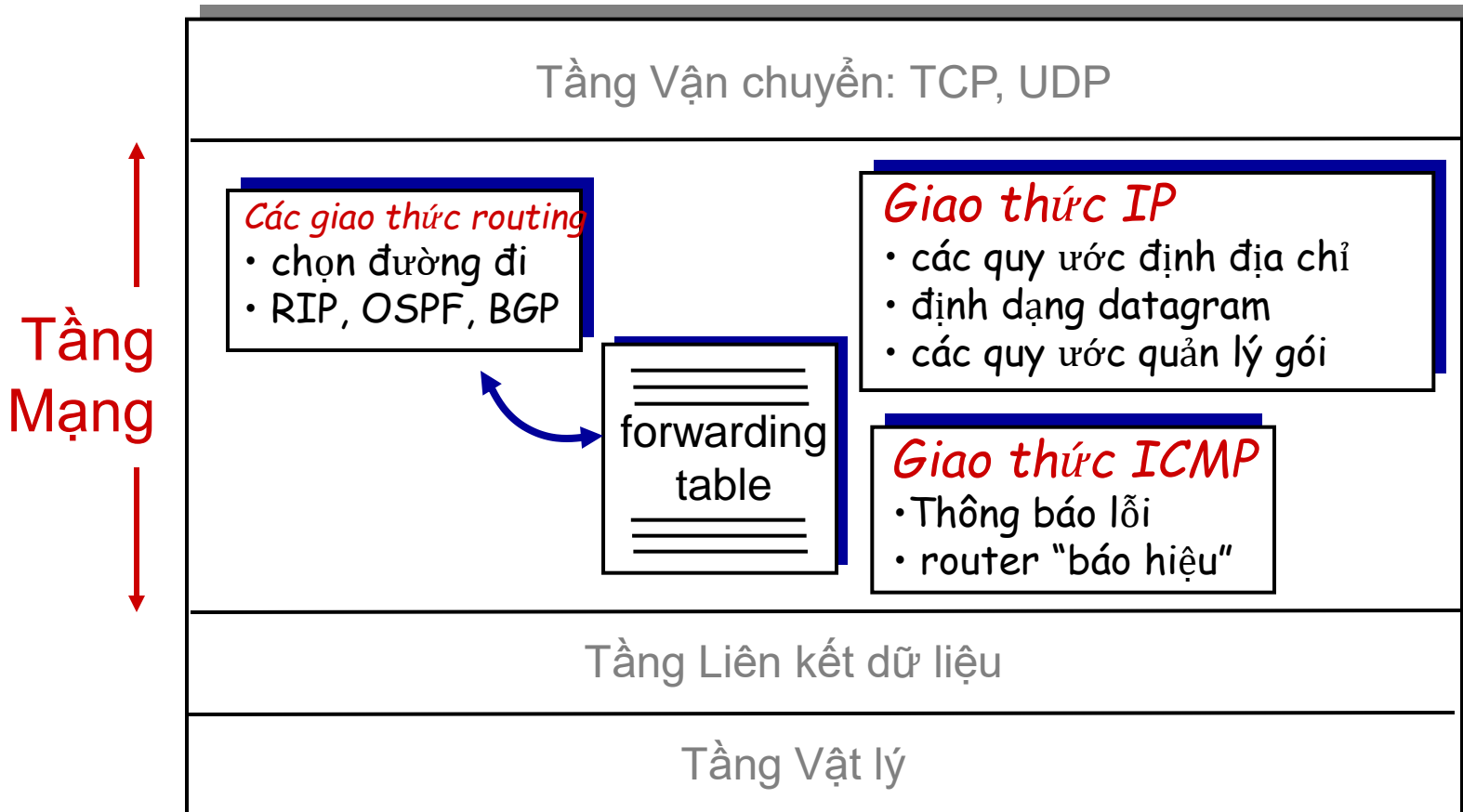
# Các mô hình dịch vụ tầng Mạng:

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

best-effort service

# Tầng Mạng Internet

Các chức năng tầng Mạng của host và router:



# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing

# Dịch vụ connection-oriented (hướng kết nối) và connection-less (phi kết nối)

- ❖ Mạng *datagram* cung cấp dịch vụ *phi kết nối* tại tầng Mạng
- ❖ Mạng *mạch ảo (virtual-circuit network)* cung cấp dịch vụ *hướng kết nối* tại tầng Mạng
- ❖ Tương tự như các dịch vụ kết nối định hướng và không định hướng của tầng Vận chuyển, nhưng:
  - *Dịch vụ*: kết nối giữa 2 máy đầu cuối
  - *Không lựa chọn*: hệ thống mạng chỉ cung cấp 1 trong 2 dịch vụ
  - *Triển khai*: bên trong phần lõi của mạng

# Các mach ảo (Virtual circuits)

“các hoạt động trên đường đi từ nguồn tới đích tương tự như mạng điện thoại”

- Hiệu quả hoạt động tốt
- Các hoạt động của mạng dọc theo đường đi từ nguồn tới đích

- ❖ Thiết lập cuộc gọi mỗi cuộc gọi trước khi dữ liệu có thể truyền. Ngắt kết nối sau khi kết thúc cuộc gọi.
- ❖ Mỗi packet mang số nhận dạng của kết nối ảo (VC identifier) (không phải là địa chỉ của host đích)
- ❖ *Mỗi* router trên đường đi từ nguồn tới đích duy trì trạng thái cho mỗi kết nối đi qua nó.
- ❖ Đường kết nối, các tài nguyên router (bảng thông, bộ nhớ đệm) có thể được cấp phát cho từng kết nối ảo (các tài nguyên dành riêng => dịch vụ có thể dự đoán trước)

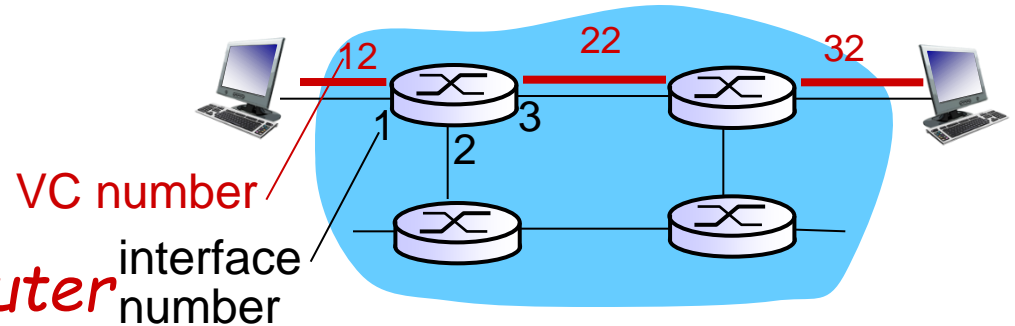
# Triển khai kết nối ảo (VC)

*Một kết nối ảo bao gồm:*

1. *Đường đi (path)* từ nguồn tới đích
  2. *Số hiệu nhận dạng kết nối ảo (VC numbers)*, mỗi số cho mỗi kết nối dọc theo đường đi
  3. *Các mục trong các bảng forwarding* ở trong các router dọc theo đường đi
- ❖ Gói thuộc về 1 kết nối ảo mang số nhận dạng của kết nối ảo đó (không dùng địa chỉ đích)
  - ❖ Số nhận dạng kết nối ảo có thể thay đổi trên mỗi đoạn kết nối
    - Số nhận dạng mới của kết nối ảo được cấp phát từ bảng forwarding

# Bảng forwarding của kết nối ảo

*Bảng forwarding trong router ở góc trái trên cùng:*

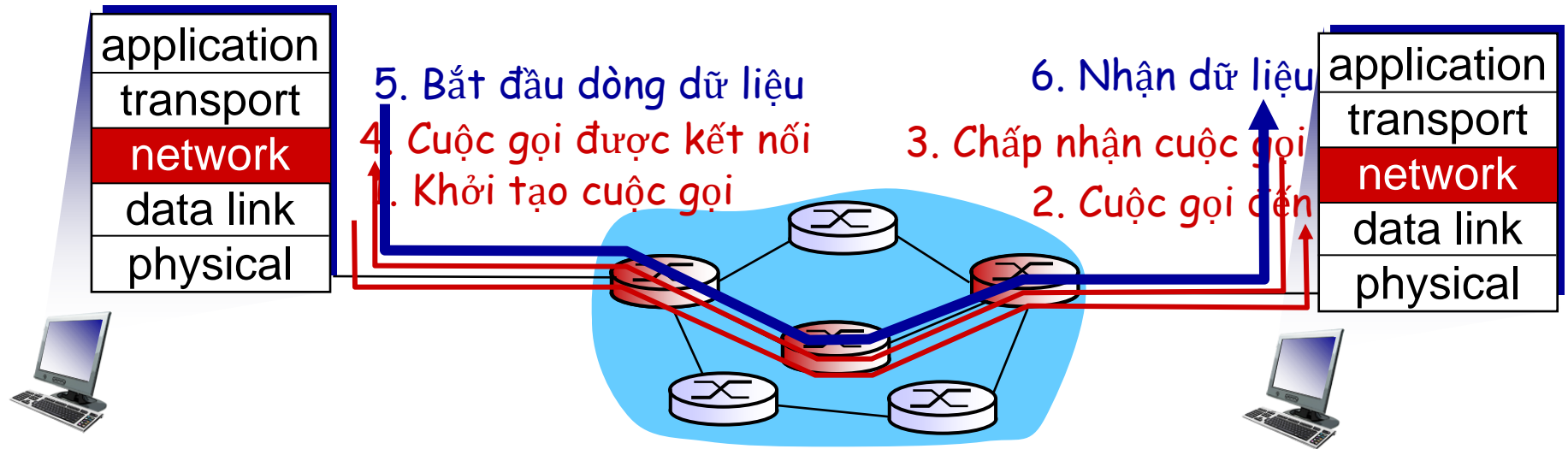


Cổng vào	số hiệu của kết nối ảo vào	Cổng ra	số hiệu của kết nối ảo ra
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

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

# Các mạch ảo: các giao thức gửi tín hiệu

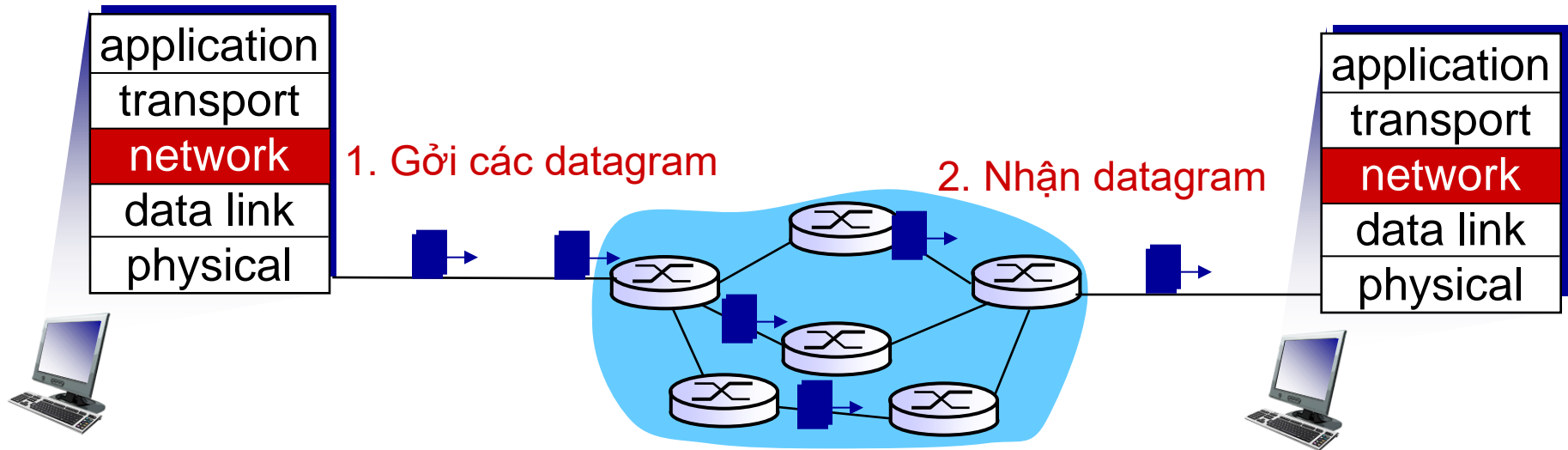
- ❖ Được dùng để thiết lập, duy trì kết nối ảo
- ❖ Được dùng trong ATM, frame-relay, X.25
- ❖ Không được sử dụng trong Internet ngày nay



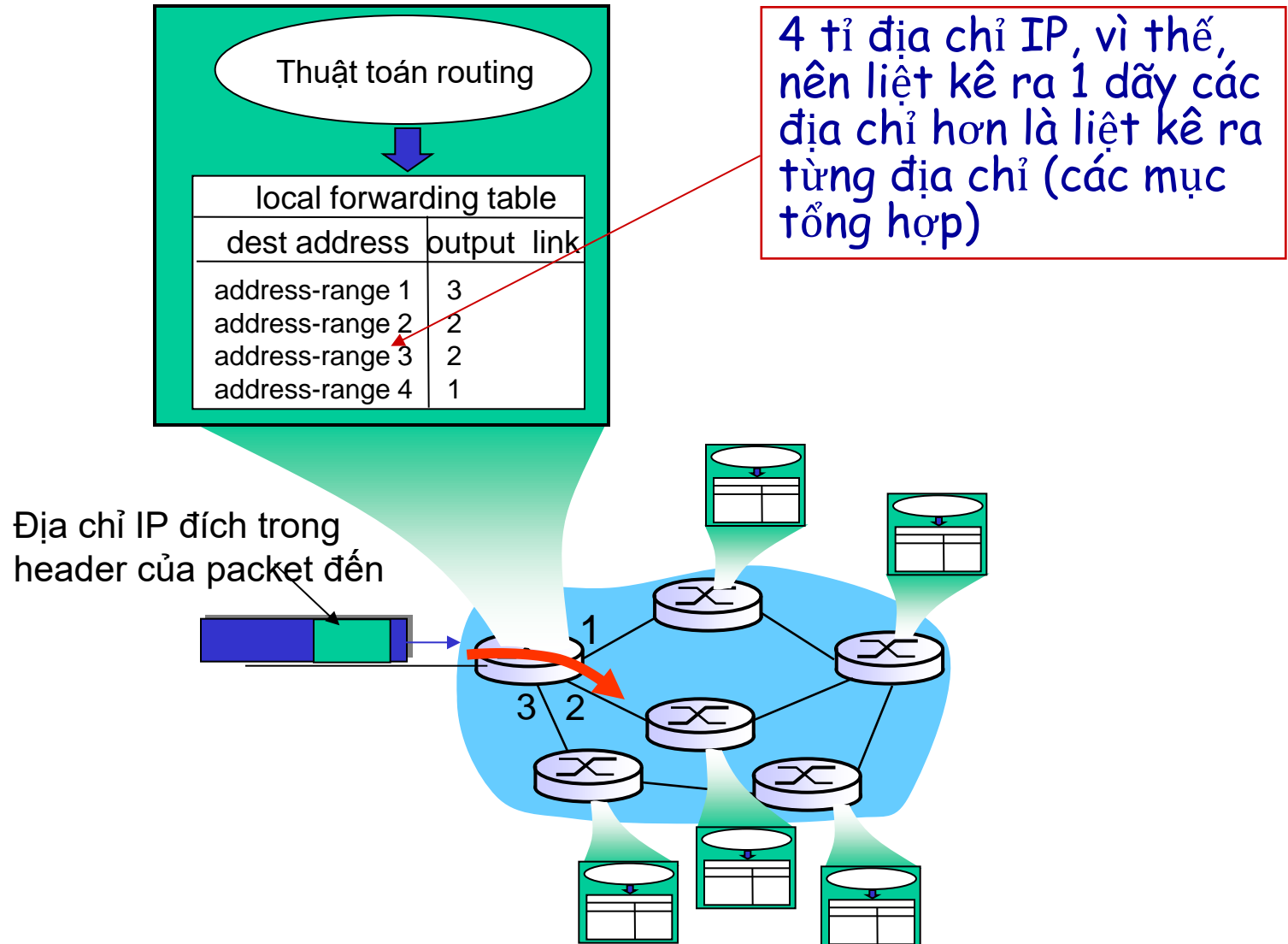


# Mạng chuyển gói (Datagram network)

- ❖ Không thiết lập cuộc gọi tại tầng Mạng
- ❖ Các router: không giữ trạng thái về các kết nối giữa 2 điểm cuối
  - Không có khái niệm “kết nối” ở tầng Mạng
- ❖ Vận chuyển các gói dùng địa chỉ máy đích



# Bảng Datagram forwarding



# Bảng Datagram forwarding

Dãy địa chỉ đích	Link Interface
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

**Q:** Nhưng điều gì sẽ xảy ra nếu các dãy địa chỉ này không được chia hợp lý?

# So trùng phần đầu dài nhất (Longest prefix matching)

## *So trùng phần đầu dài nhất*

Khi tìm kiếm 1 cổng ra trong bảng forwarding cho địa chỉ đích, so phần đầu dài nhất trùng giữa địa chỉ trong bảng và địa chỉ đích.

Dãy địa chỉ đích	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

Ví dụ:

DA: 11001000 00010111 00010110 10100001

Interface nào?

DA: 11001000 00010111 00011000 10101010

Interface nào?

# Mạng chuyển gói hay mạch ảo: tại sao?

---

## *Internet (datagram)*

- ❖ Dữ liệu trao đổi giữa các máy tính
  - Dịch vụ “mềm dẻo, không yêu cầu chặt chẽ về độ trễ.
- ❖ Nhiều kiểu kết nối
  - Các đặc tính khác nhau
  - Khó đồng nhất dịch vụ
- ❖ Các hệ thống đầu cuối “thông minh” (các máy tính)
  - Có thể thích ứng, điều khiển và sửa lỗi
  - **“phần lỗi” mạng đơn giản, “phần cạnh” phức tạp**

## *ATM (mạch ảo)*

- ❖ Được phát triển từ hệ thống điện thoại
- ❖ Đàm thoại của con người:
  - Yêu cầu chặt chẽ về độ trễ, độ tin cậy
  - Cần cho các dịch vụ cần bảo đảm
- ❖ Các hệ thống đầu cuối “ít thông minh”
  - Điện thoại
  - **“phần lỗi” mạng phức tạp**

# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

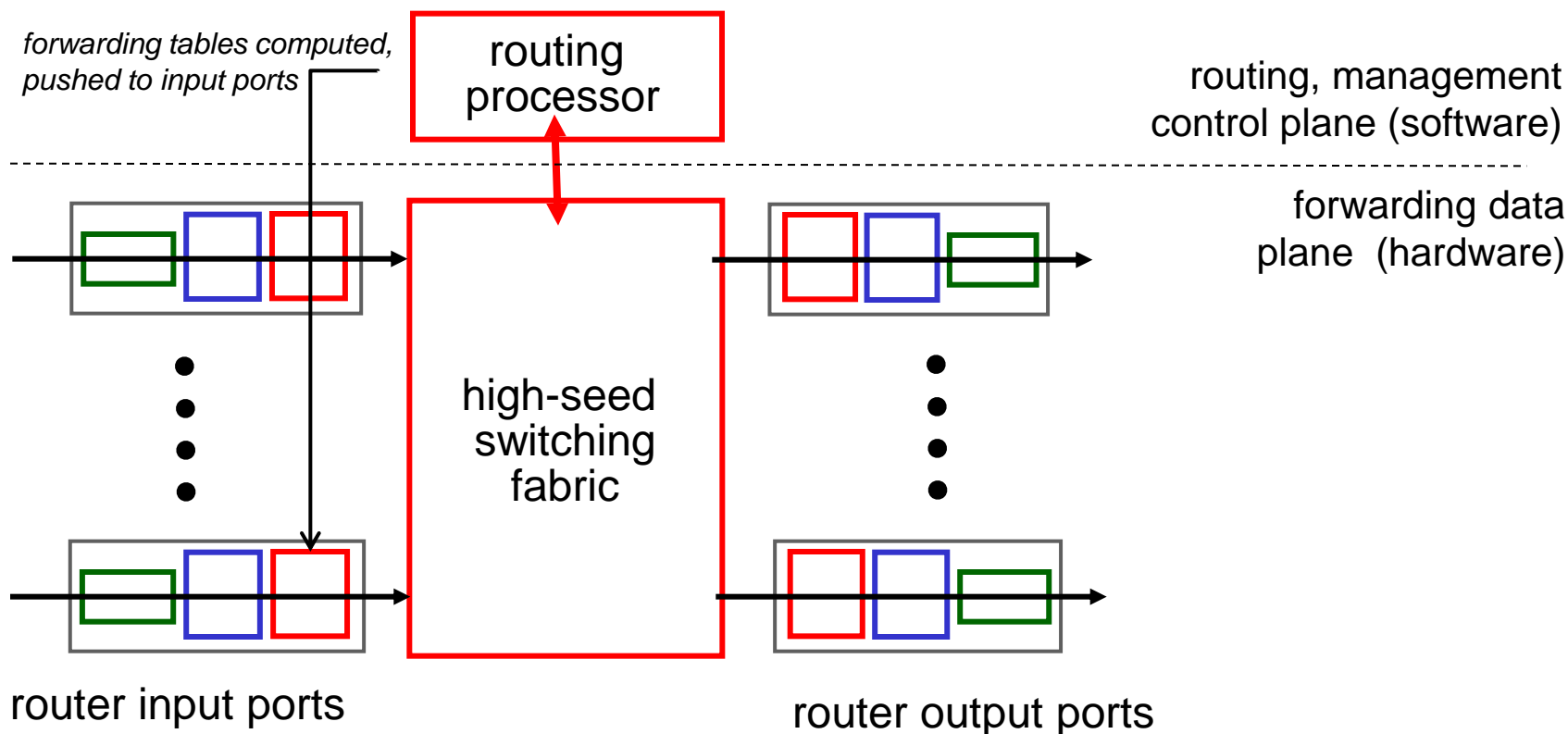
- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing

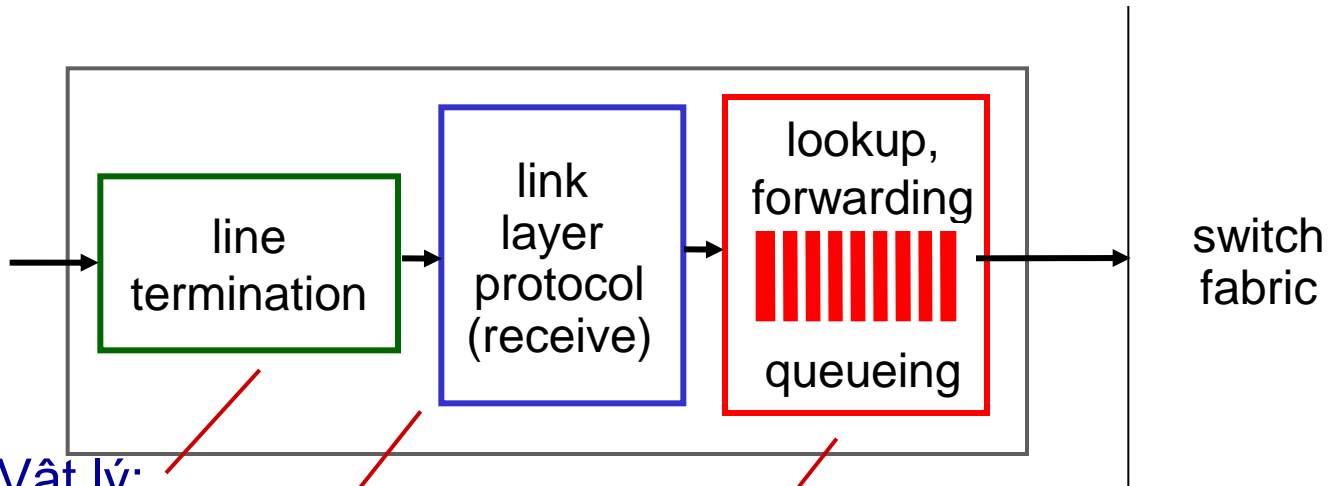
# Tổng quan kiến trúc Router

2 chức năng chính của router:

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



# Các chức năng của cổng vào (Input)



Tầng Vật lý:

Tiếp nhận mức bit

Tầng Liên kết dữ liệu:

Như là Ethernet

Xem chương 5

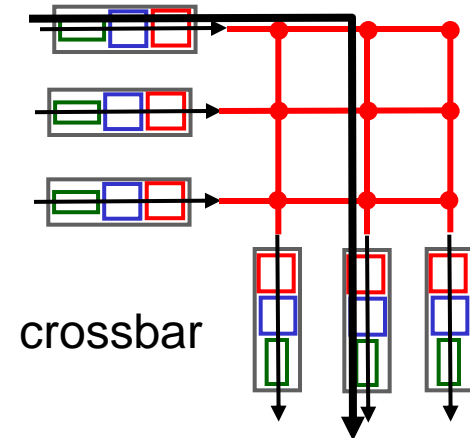
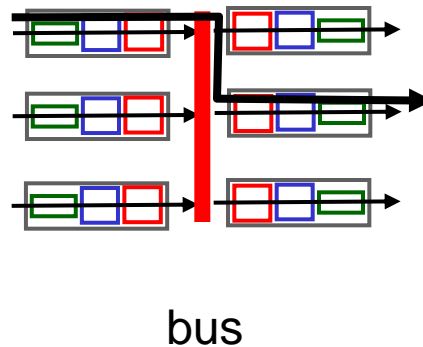
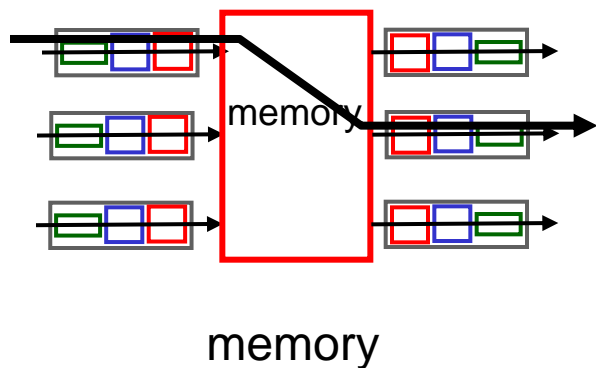
**Decentralized switching (chuyển mạch không tập trung):**

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



# Mạch chuyển mạch (Switching fabrics)

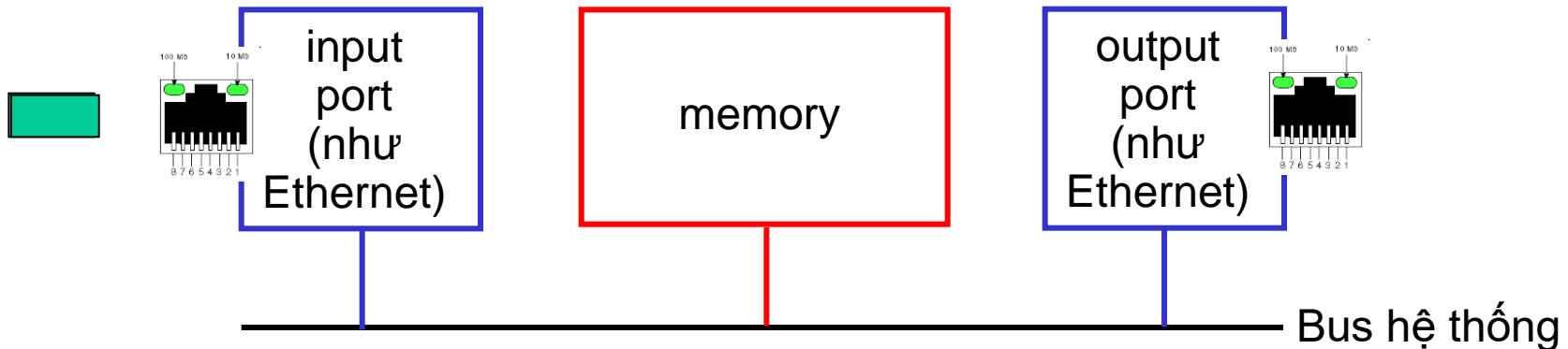
- ❖ Truyền packet từ bộ nhớ đệm đầu vào đến bộ nhớ đệm đầu ra thích hợp
- ❖ Tốc độ switching: tốc độ mà các packet có thể được truyền từ đầu vào (inputs) đến đầu ra (outputs)
  - Thường được tính là tốc độ  $N \times$  (số luồng của đầu vào/đầu ra)
- ❖ 3 kiểu switching fabrics



# Chuyển gói thông qua bộ nhớ (memory)

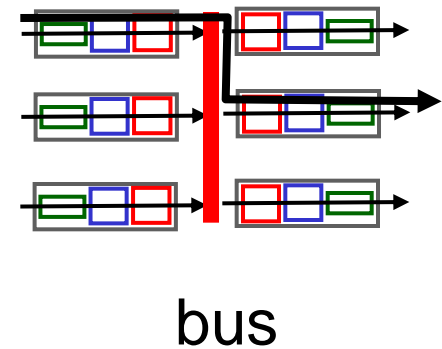
## *Các router thế hệ đầu tiên:*

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



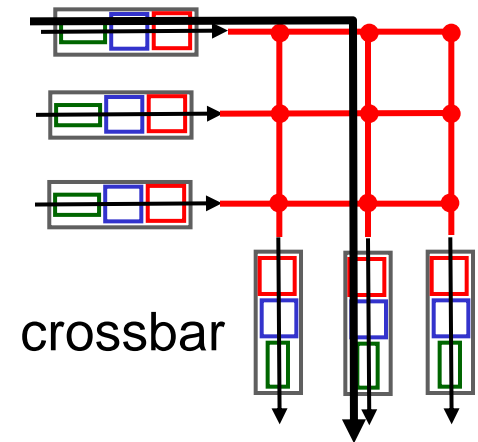
# Chuyển gói thông qua bus

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

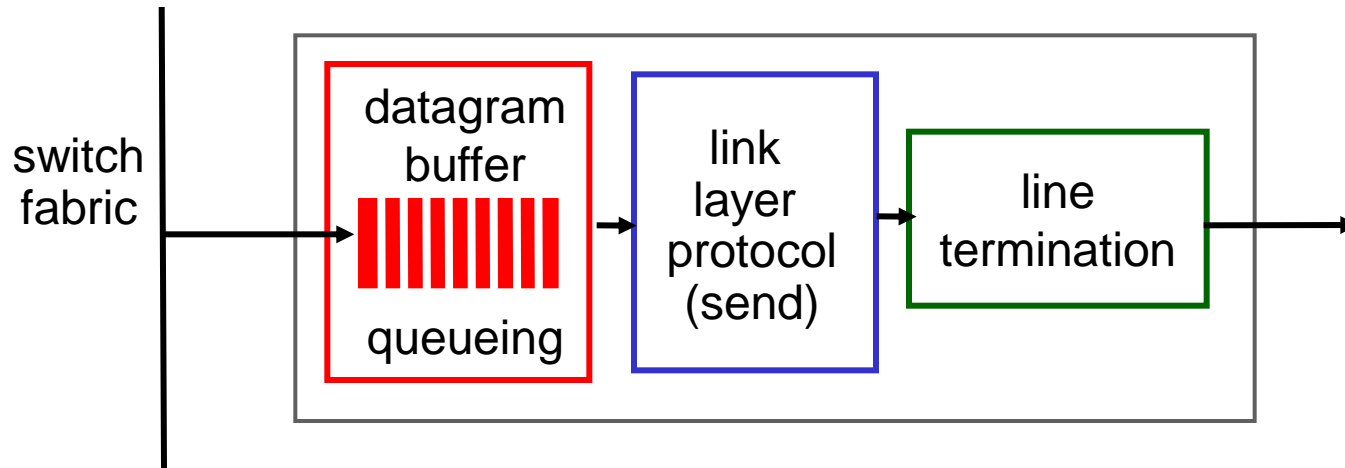


# Chuyển mạch thông qua interconnection network

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

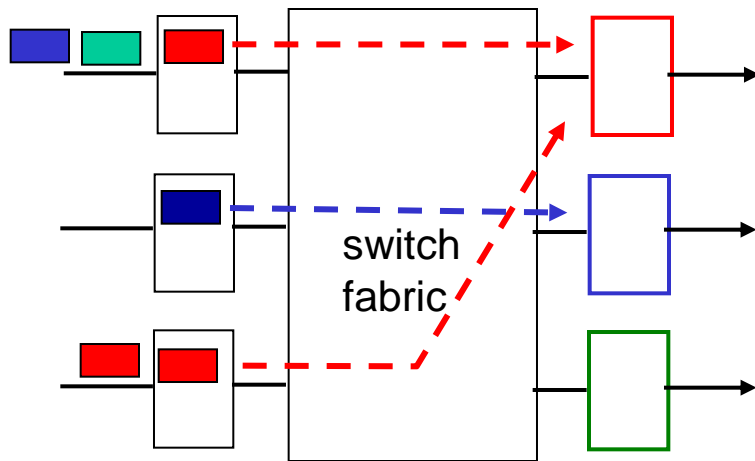


# Các cổng ra (Output)

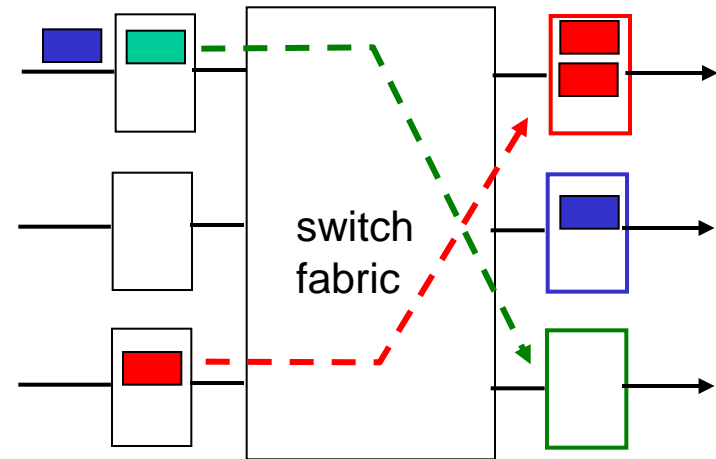


- ❖ *Xếp hàng trong bộ đệm* được yêu cầu khi tốc độ chuyển các gói tin đến từ mạch xử lý của switch nhanh hơn tốc độ truyền
- ❖ *Lập lịch (Scheduling discipline)* cách thức chọn trong số các gói tin đã xếp hàng để truyền

# Sắp hàng tại cổng ra



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



one packet time later

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

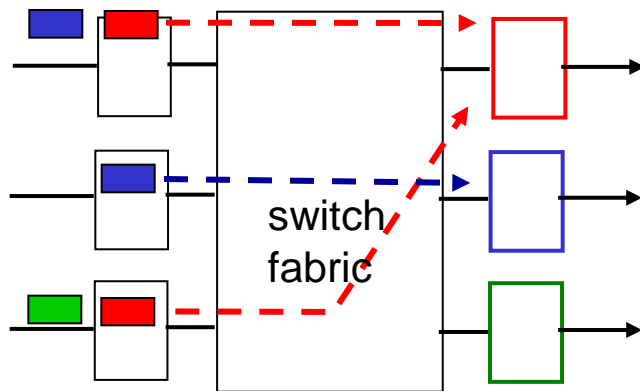
# Kích thước buffer?

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

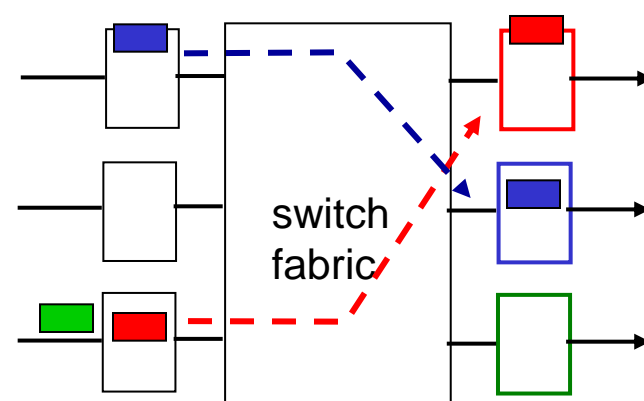
$$\frac{RTT \cdot C}{\sqrt{N}}$$

# Sắp hàng tại cổng vào

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



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



packet màu xanh lá  
bị chặn HOL  
blocking



# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

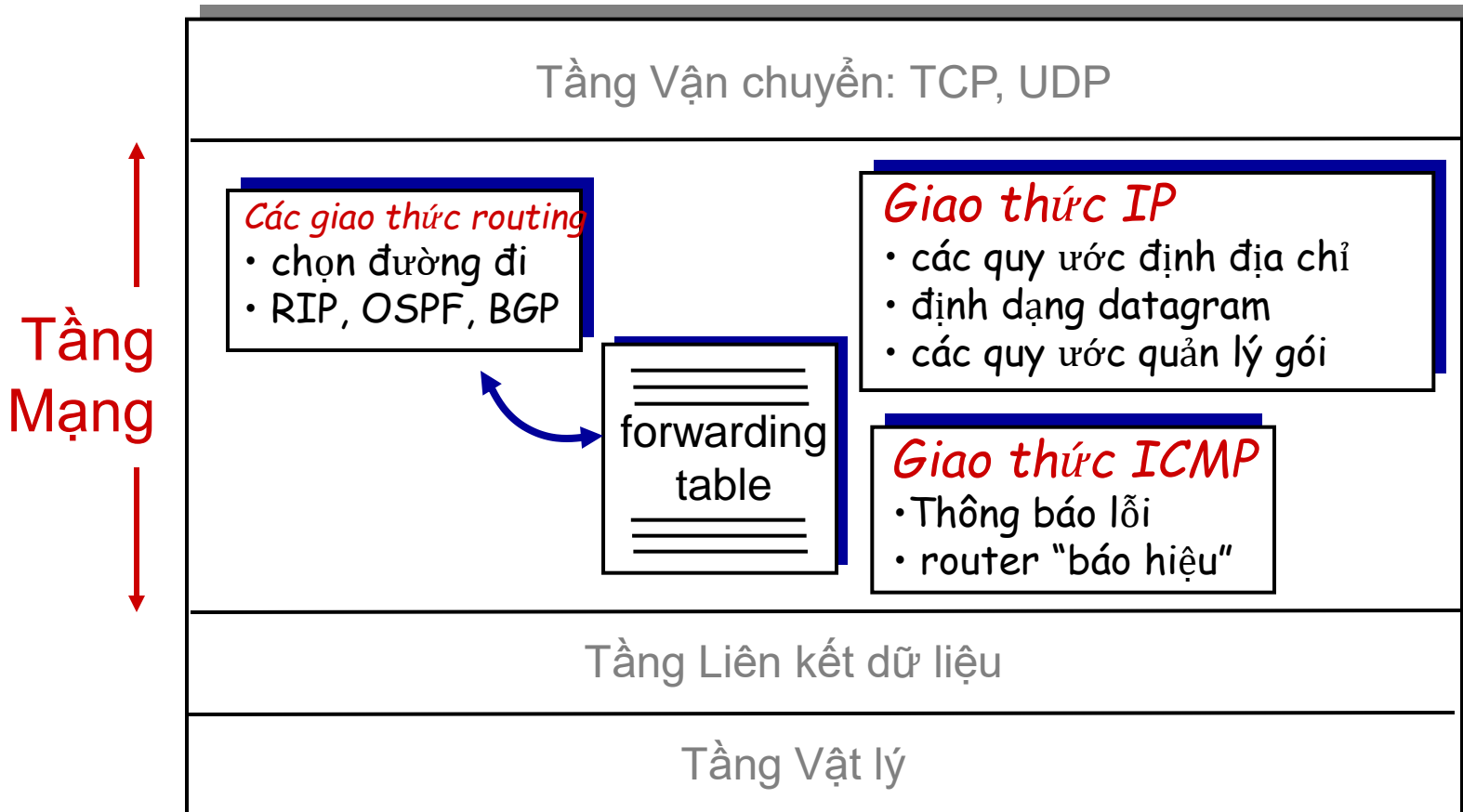
## 4.6 Routing trong Internet

- RIP
- OSPF
- BGP

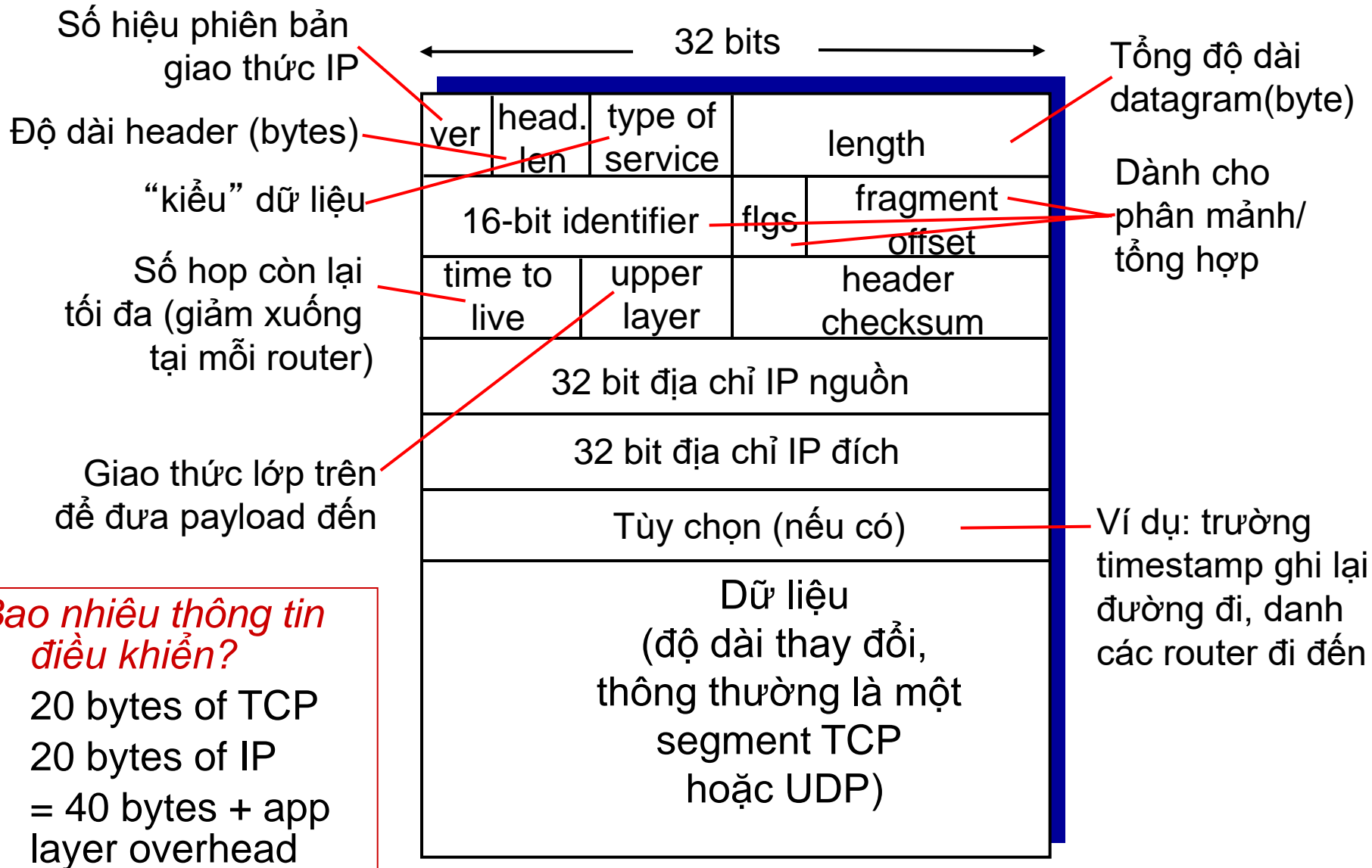
## 4.7 Broadcast và multicast routing

# Tầng Mạng Internet

Các chức năng tầng Mạng của host và router:

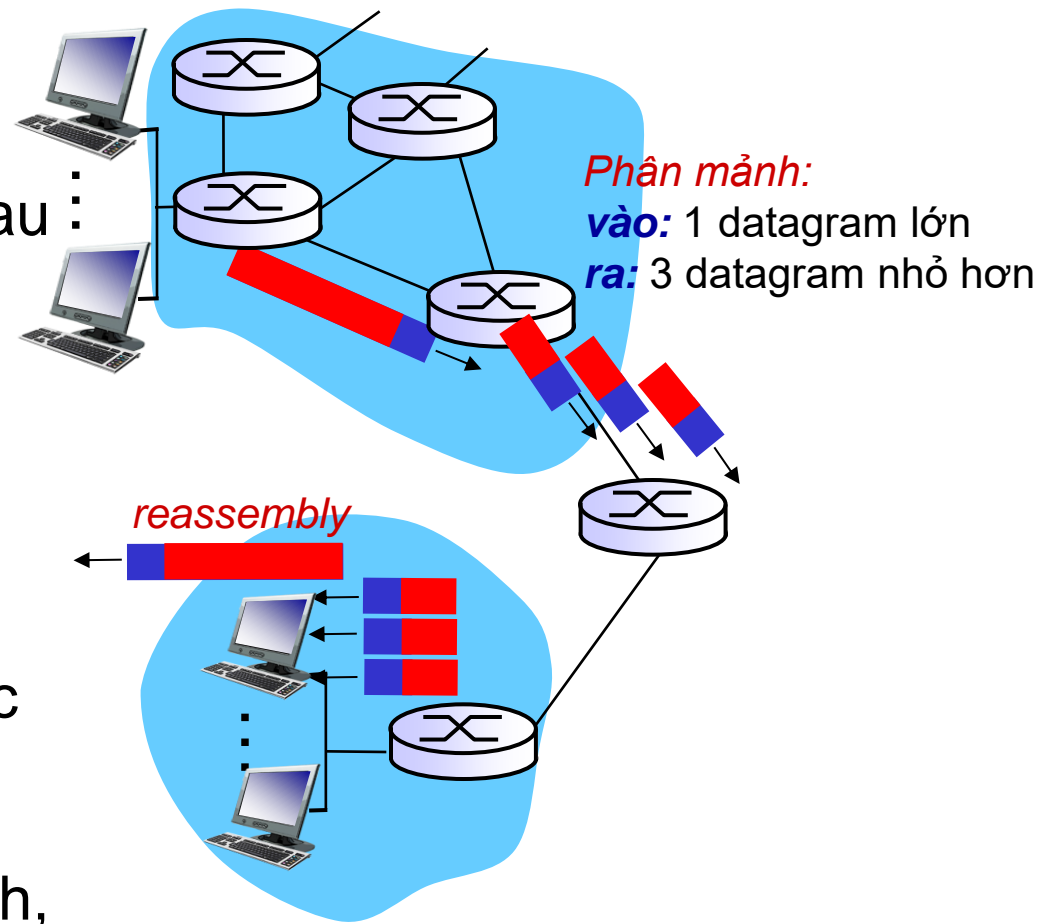


# Định dạng IP datagram



# Phân mảnh và tổng hợp IP

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



# Phân mảnh và tổng hợp IP

*Ví dụ:*

- ❖ 4000 byte datagram
- ❖ MTU = 1500 bytes

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

*1 datagram lớn thành vài datagram nhỏ hơn*

1480 bytes  
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 Virtual circuit network (Mạng mạch ảo) và datagram network (Mạng chuyển gói)

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

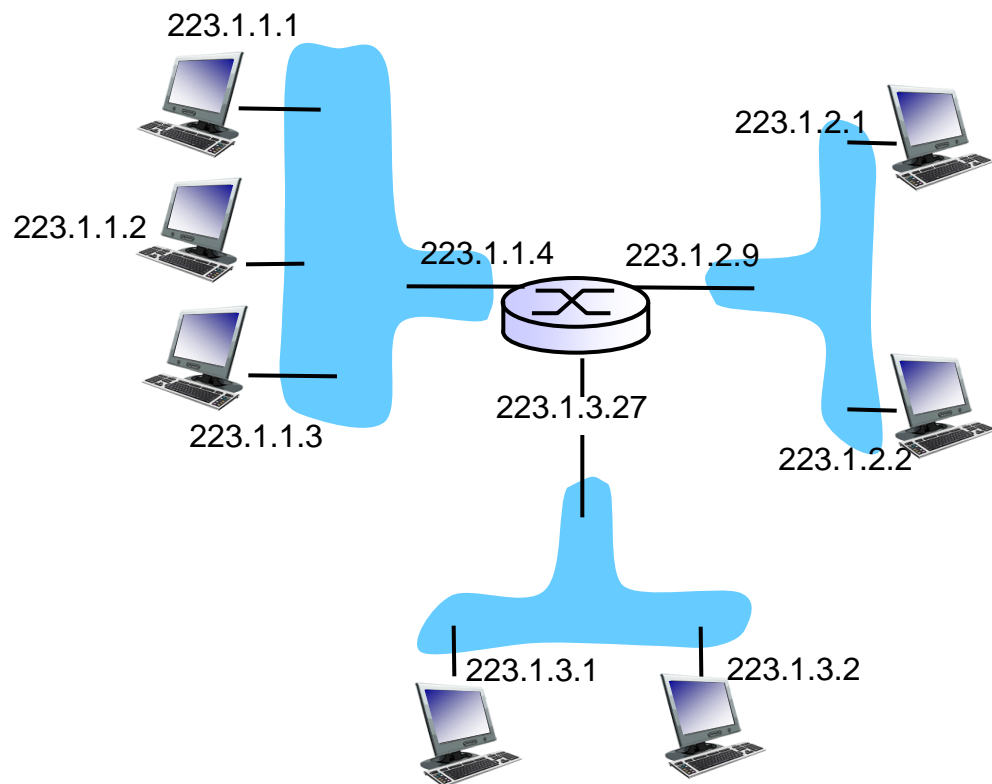
## 4.6 Routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing

# Định địa chỉ IP: giới thiệu

- ❖ **Địa chỉ IP:** 32-bit nhận dạng cho host, router interface
- ❖ **Interface:** kết nối giữa host/router và đường kết nối vật lý
  - Router thường có nhiều interface
  - Host thường có 1 hoặc 2 interface (ví dụ wired Ethernet, wireless 802.11)
- ❖ **Mỗi địa chỉ IP được liên kết với mỗi interface duy nhất**
- ❖ Các host/router interface nằm trong cùng một đoạn mạng cục bộ được gán các địa chỉ IP tương tự nhau (Có phần đầu giống nhau)



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

# Địa chỉ IP: giới thiệu

- ❖ **Địa chỉ IP**: là địa chỉ có cấu trúc với một con số có kích thước 32 bit, chia thành 4 phần, mỗi phần có 8 bit gọi là octet hoặc byte  
223.1.1.2

- ❖ Ví dụ:

- 172.16.30.56
- 10101100 00010000 00011110 00111000
- AC 10 1E 38

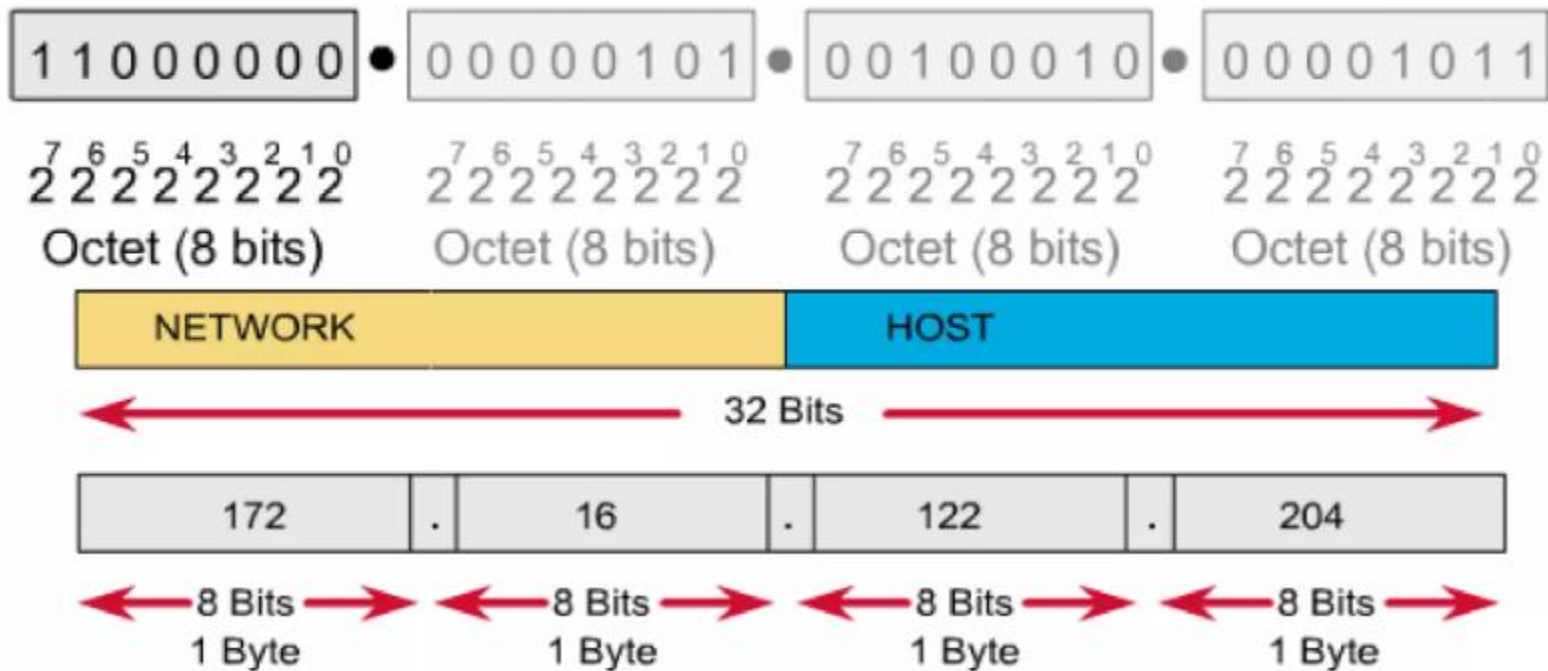


# Địa chỉ IP: giới thiệu

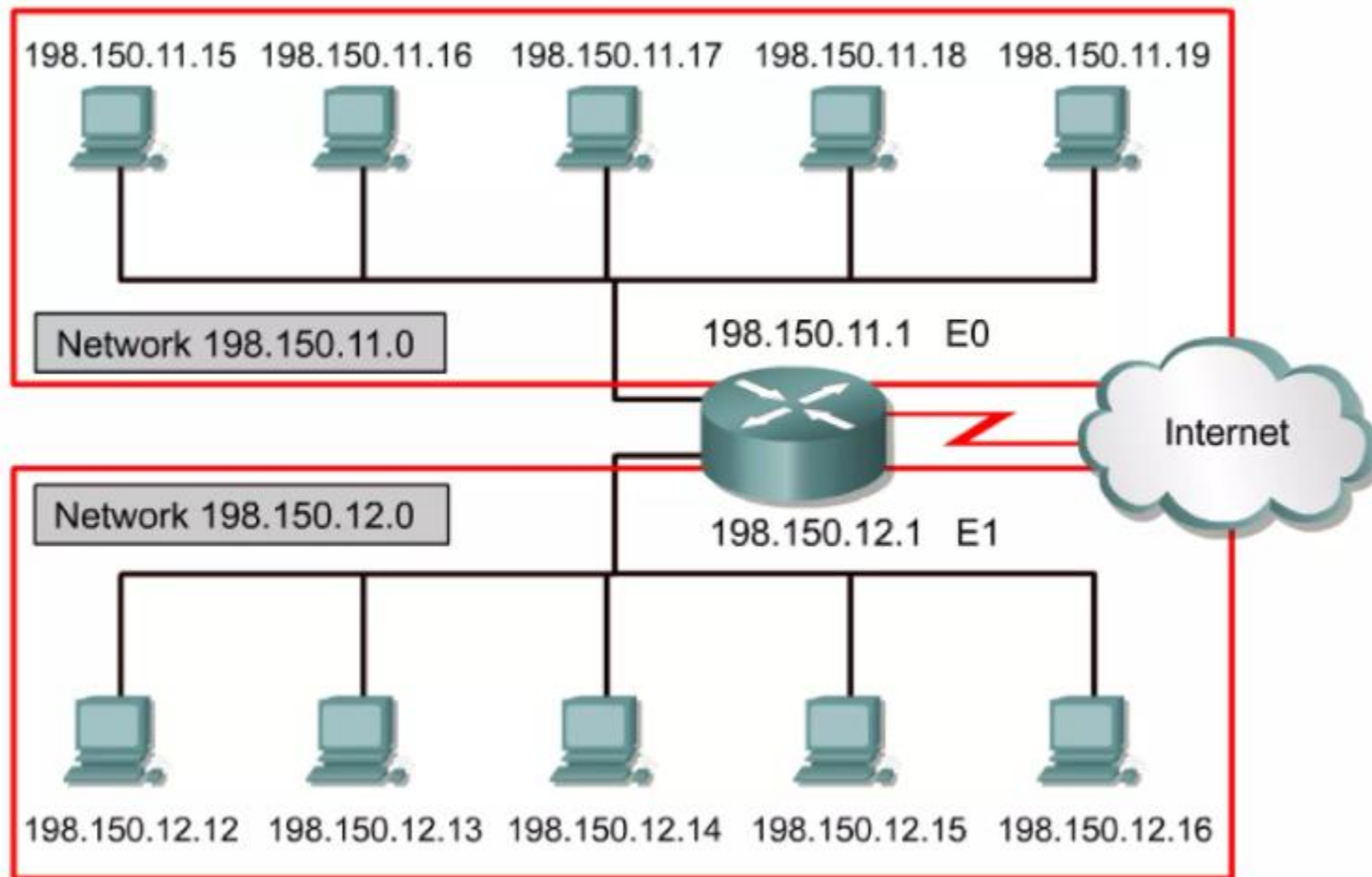
## ❖ 1 số khái niệm liên quan:

- **Địa chỉ host:** là địa chỉ IP có thể dùng để đặt cho các interface của các hos. Hai host nằm cùng một mạng sẽ có địa chỉ mạng (network\_id) giống nhau và địa chỉ host (host\_id) khác nhau.
- **Địa chỉ mạng:** là địa chỉ IP dùng để dta859 cho các mạng. Phần host\_id của địa chỉ chỉ chứa các bit 0.
  - Ví dụ: 172.29.0.0, 192.168.1.0, 10.0.0.0
- **Địa chỉ broadcast (địa chỉ quảng bá):** là địa chỉ IP cuối cùng của mạng. Phần host\_id của địa chỉ này chủ chứa các bit 1.
  - Ví dụ: 172.29.255.255

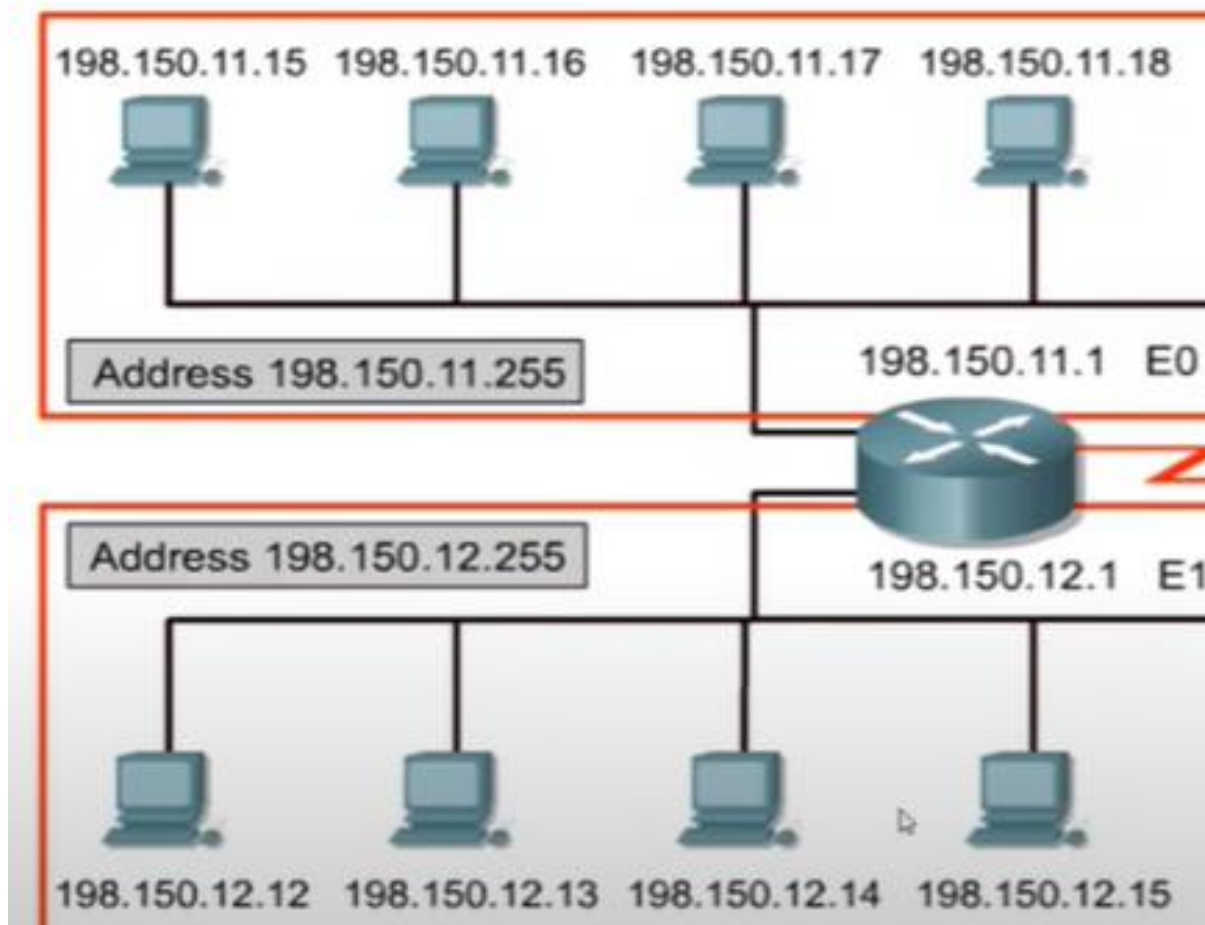
# Địa chỉ IP: giới thiệu



# Địa chỉ IP: giới thiệu

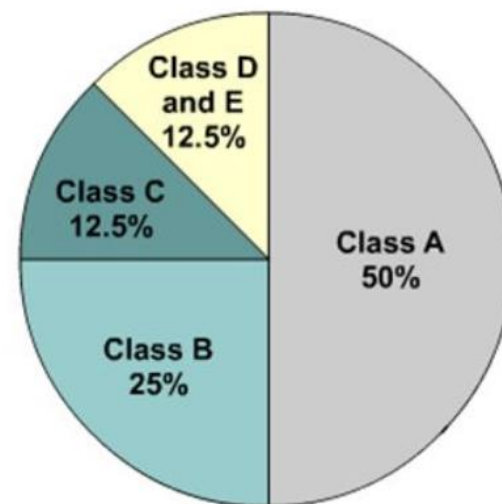


# Địa chỉ IP: giới thiệu



# Các lớp địa chỉ IP

- ❖ Không gian địa chỉ IP được chia thành 5 lớp (classes) A, B, C, D, E.
- ❖ Các lớp A, B và C được triển khai để đặt cho các host trên mạng Internet. Lớp D dùng cho các nhóm multicast, còn lớp E phục vụ cho mục đích nghiên cứu.



	8 bits	8 bits	8 bits	8 bits
Class A:	Network	Host	Host	Host
Class B:	Network	Network	Host	Host
Class C:	Network	Network	Network	Host
Class D:	Multicast			
Class E:	Research			

# Các lớp địa chỉ IP

CLASS	LEADING BITS	NET ID BITS	HOST ID BITS	NO. OF NETWORKS	ADDRESSES PER NETWORK	START ADDRESS	END ADDRESS
CLASS A	0	8	24	$2^7$ ( 128 )	$2^{24}$ (16,777,216)	0.0.0.0	127.255.255.255
CLASS B	10	16	16	$2^{14}$ ( 16,384 )	$2^{16}$ ( 65,536 )	128.0.0.0	191.255.255.255
CLASS C	110	24	8	$2^{21}$ ( 2,097,152 )	$2^8$ ( 256 )	192.0.0.0	223.255.255.255
CLASS D	1110	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	224.0.0.0	239.255.255.255
CLASS E	1111	NOT DEFINED	NOT DEFINED	NOT DEFINED	NOT DEFINED	240.0.0.0	255.255.255.255

# Địa chỉ dành riêng

- ❖ Được sử dụng trong các mạng cục bộ (LAN)
- ❖ Những địa chỉ ngoài 3 dãy địa chỉ dành riêng này là những địa chỉ được sử dụng thông dụng trên mạng Internet

Private Addresses		
Class	RFC 1918 Internal Address Range	CIDR Prefix
A	10.0.0.0–10.255.255.255	10.0.0.0/8
B	172.16.0.0–172.31.255.255	172.16.0.0/12
C	192.168.0.0–192.168.255.255	192.168.0.0/16

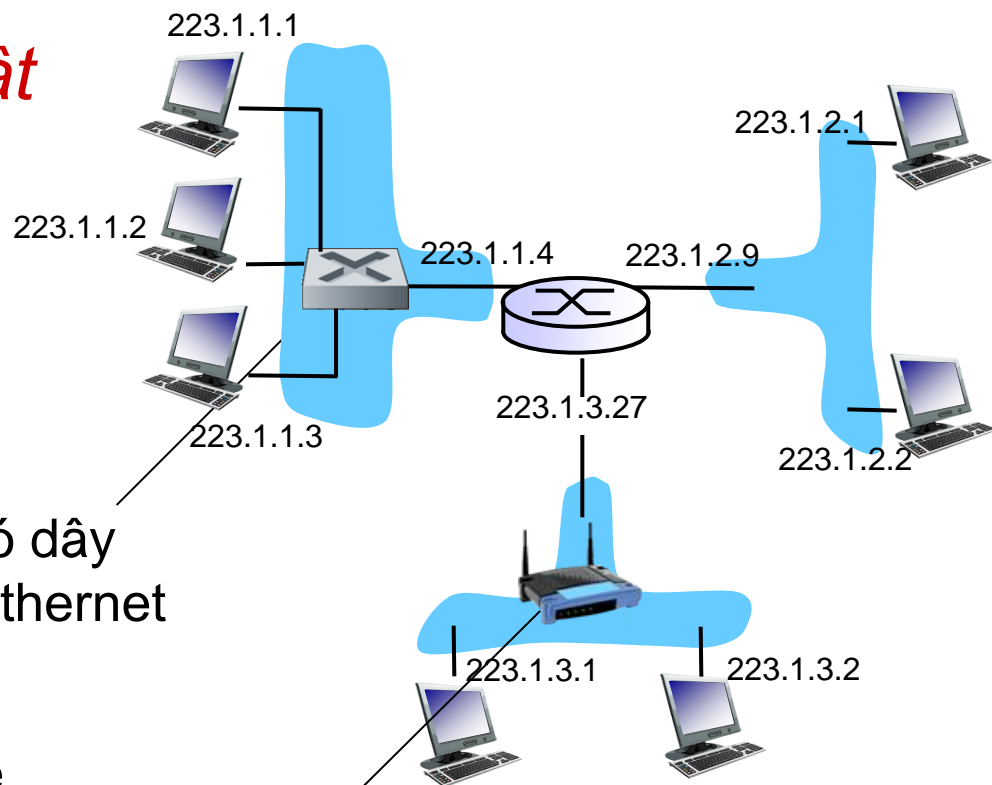
# Định địa chỉ IP: giới thiệu

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

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

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

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



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



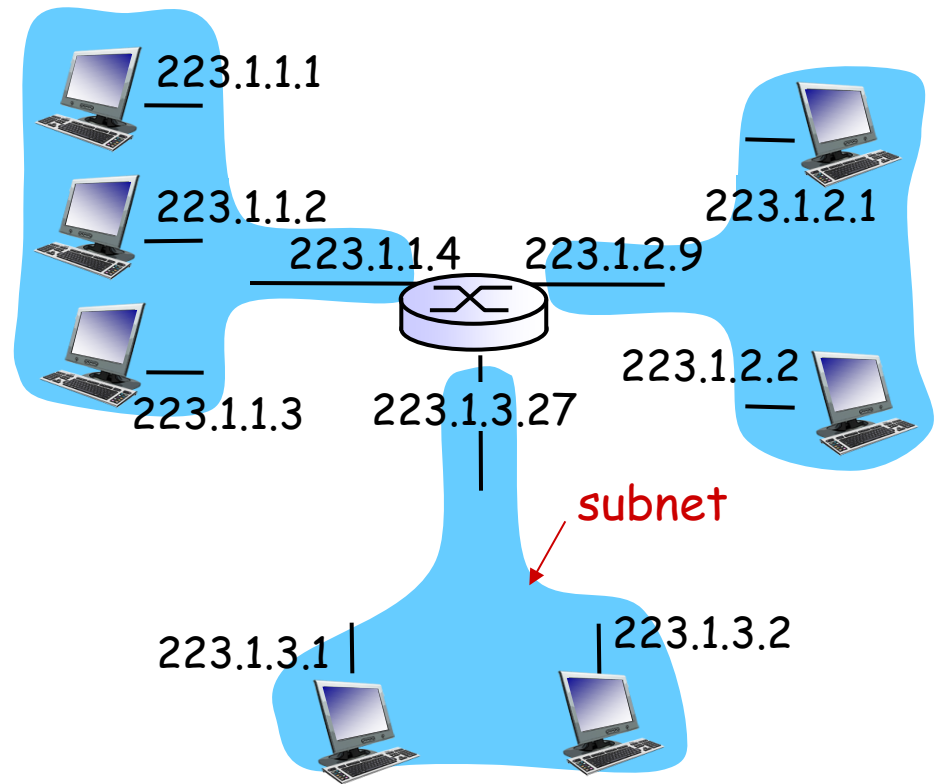
# Các Subnet (mạng con)

## ❖ Đại chỉ IP:

- Phần subnet – các bit bên trái
- Phần host – các bit bên phải

## ❖ *subnet là gì?*

- Các interface của thiết bị có phần subnet của địa chỉ IP giống nhau
- Có thể giao tiếp vật lý với nhau mà không cần *router trung gian* cần thiết

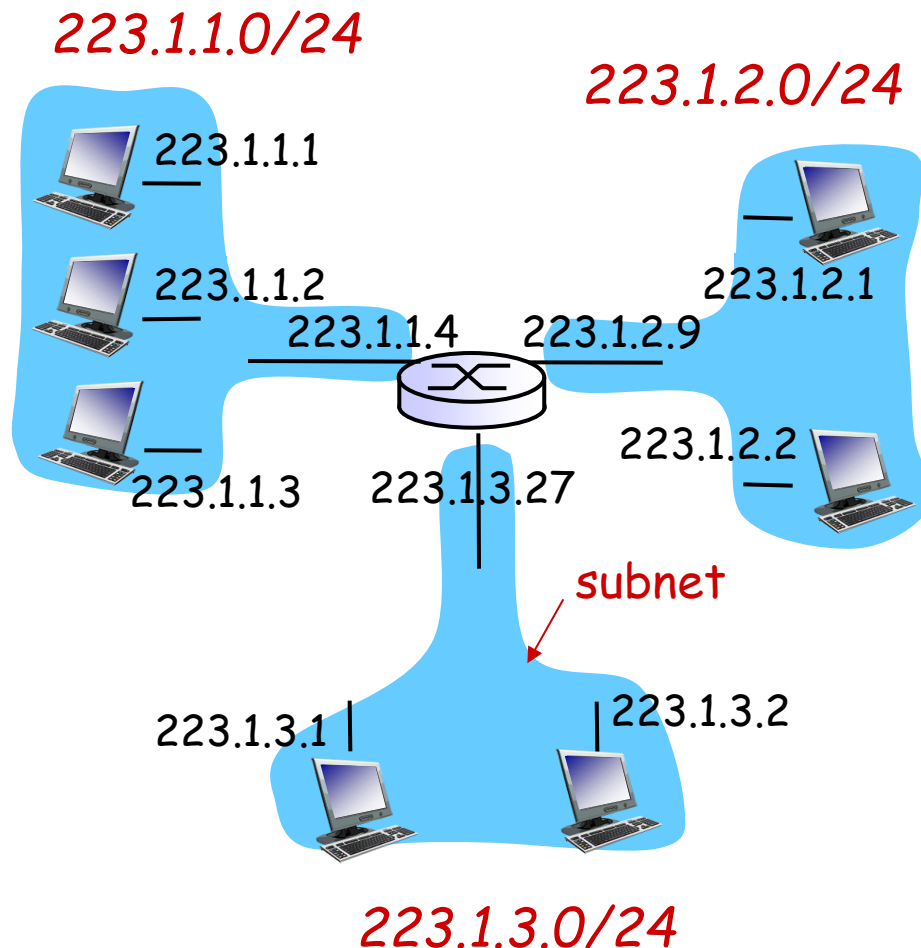


Mạng gồm 3 subnet

# Các mạng con (subnet)

## *Phương pháp*

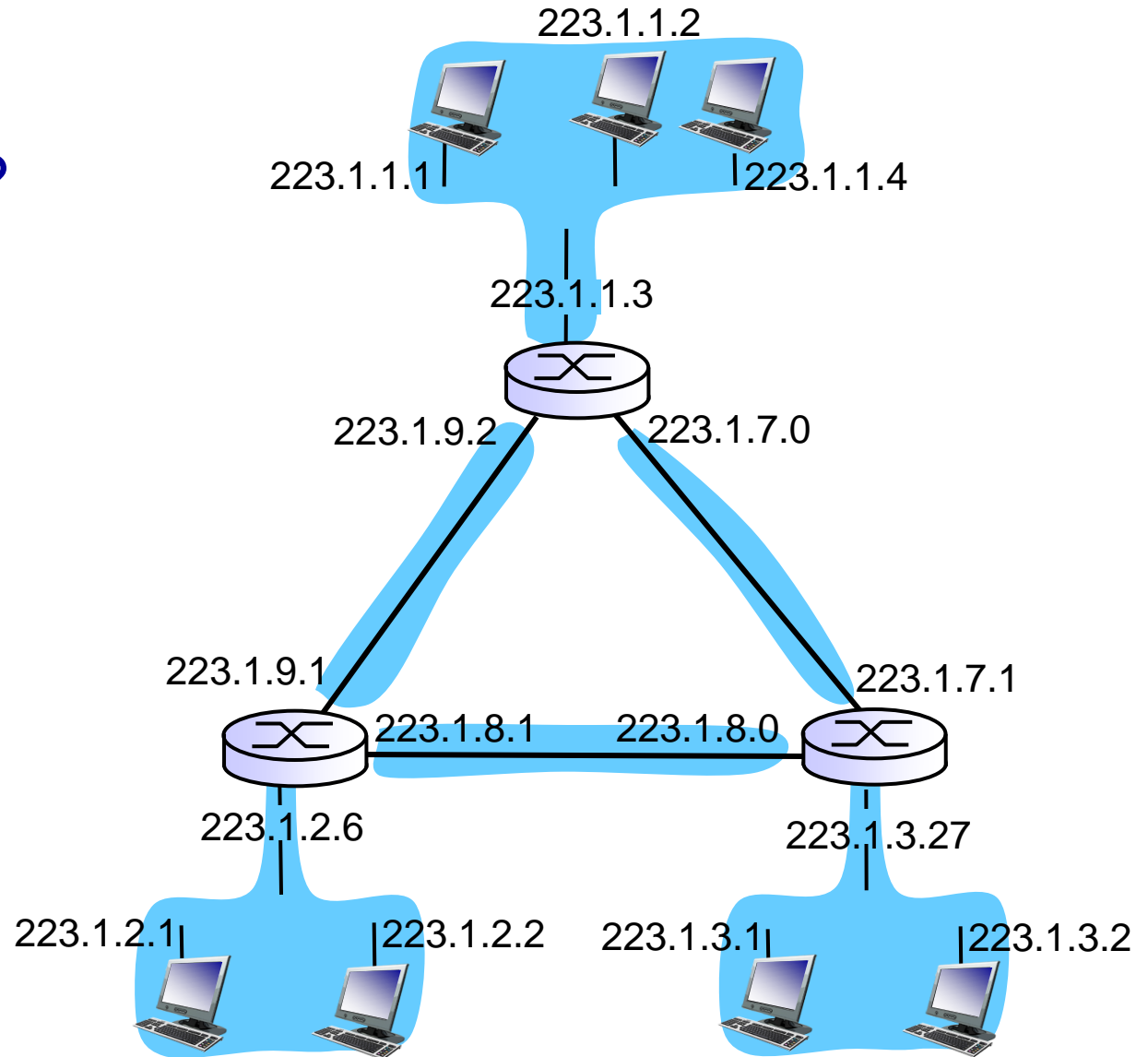
- ❖ Để xác định các subnet, tách mỗi interface từ host hoặc router của nó, tạo vùng các mạng độc lập
- ❖ Mỗi mạng độc lập được gọi là một *subnet*



subnet mask: /24

# Subnets

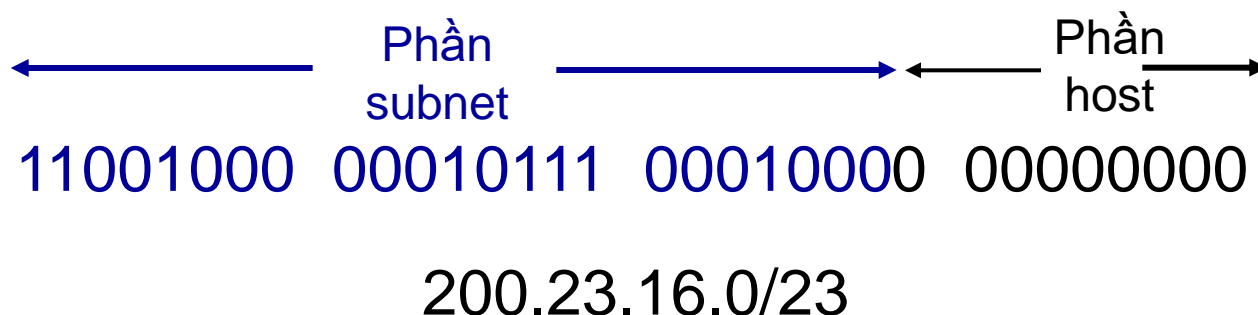
Có bao nhiêu?



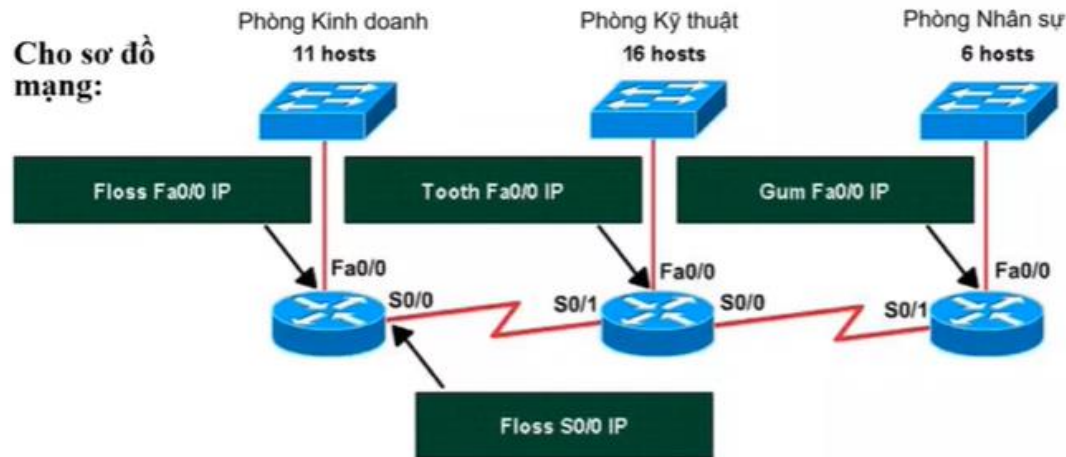
# Định địa chỉ IP: CIDR

## CIDR: Classless InterDomain Routing

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



# Vấn đề chia mạng con



- ❖ Chia mạng ban đầu thành nhiều mạng nhỏ hơn với các ưu điểm sau:
  - Tiết kiệm địa chỉ IP
  - Hạn chế các gói tin quảng bá, tiết kiệm tài nguyên mạng (CPU, băng thông,...)
  - Tăng cường bảo mật
  - Cho phép áp dụng các cấu hình khác nhau trên mỗi mạng con
  - Dễ dàng khắc phục sự cố mạng trên các mạng LAN nhỏ

# Vấn đề chia mạng con

## ❖ **Mặt nạ mạng con (Subnet Mask)**

- Là một chuỗi các bit được sử dụng trong hệ thống mạng để phân chia địa chỉ IP thành hai phần: phần mạng (network) và phần máy chủ (host).
- Giúp xác định phần nào của địa chỉ IP thuộc về mạng và phần nào thuộc về các thiết bị cụ thể (máy tính, máy in, v.v.) trong mạng đó.
- Cấu trúc của subnet mask thường bao gồm một dãy các bit 1 liên tục (xác định phần mạng) theo sau là các bit 0 (xác định phần máy chủ).
- Các subnet mask thường gặp trong hệ thống IPv4 là:
  - 255.255.255.0
  - 255.255.0.0
  - 255.0.0.0

# Vấn đề chia mạng con

## ❖ Mặt nạ mạng con (Subnet Mask)

192.168.1.8

Decimal	192	168	1	8
Binary	11000000	10101000	00000001	00001000
Network mask	11111111	11111111	11111111	00000000

Network Prefix  
24 bits

Host Portion  
8 bits

Destination Address	192.168.1.8	11000000 10101000 00000001	00001000
Network Mask	255.255.255.0	11111111 11111111 11111111	00000000
Network number	192.168.1.0	11000000 10101000 00000001	00000000

192.168.1.8/24

# Vấn đề chia mạng con

- ❖ Mượn một số bit trong phần host\_id ban đầu để đặt cho các mạng con
- ❖ Cấu trúc của địa chỉ IP sẽ gồm 3 phần: network\_id, subnet\_id và host\_id



- ❖ Số bit dùng trong subnet\_id tùy thuộc vào chiến lược chia mạng con, tuy nhiên phải luôn thỏa mãn điều kiện:

$$\text{subnet\_id} \leq \text{host\_id} - 2$$



# Vấn đề chia mạng con

- ❖ Số lượng bit tối đa có thể mượn:
  - **Lớp A:** 22 ( $= 24 - 2$ ) bit  $\rightarrow$  chia được  $2^{22} - 2 = 4194302$  mạng con
  - **Lớp B:** 14 ( $= 16 - 2$ ) bit  $\rightarrow$  chia được  $2^{14} - 2 = 16382$  mạng con
  - **Lớp C:** 06 ( $= 8 - 2$ ) bit  $\rightarrow$  chia được  $2^6 - 2 = 62$  mạng con
- ❖ Số bit trong phần subnet\_id xác định số lượng mạng con. Với số bit là  $x$  thì  $2^x$  là số mạng con có được
- ❖ Nếu muốn chia 6 mạng con thì cần 3 bit ( $2^3 = 8$ ), chia 10 mạng con thì cần 4 bit ( $2^4 \geq 12$ ),...

# Vấn đề chia mạng con

- ❖ Việc chia mạng con được thực hiện theo 3 bước:
  - Bước 1: Xác định lớp (class) của địa chỉ và subnet mask mặc nhiên
  - Bước 2: Xác định số bit cần mượn và subnet mask mới, tính số lượng mạng con, số host thực sự có được
  - Bước 3: Xác định các vùng địa chỉ host và chọn mạng con muốn dùng

# Vấn đề chia mạng con

- ❖ Ví dụ 1: Cho một mạng có địa chỉ mạng là 172.16.0.0. Hãy chia mạng này thành 8 mạng con và có ít nhất 1000 host trên mỗi mạng con đó.

Giải:

- Bước 1: Xác định class và subnet mask mặc định
  - Xác định lớp của IP trên: Lớp B
  - Xác định subnet mask mặc nhiên: 255.255.0.0

# Vấn đề chia mạng con

❖ Bước 2: Số bit cần mượn, subnet mask mới, số lượng mạng con, số lượng host trên mỗi mạng con

▪ Cần mượn 3 bit ( $N = 3$ ), vì:

• Số mạng con có thể:  $2^3 = 8$

• Số host của mỗi mạng con có thể:

$$2^{16-3} - 2 = 2^{13} - 2 > 1000$$

• Xác định subnet mask mới:

→ 11111111 11111111 111 00000 00000000

→ 255. 255. 224. 0

# Vấn đề chia mạng con

## ❖ Bước 3: Xác định vùng địa chỉ host

STT	SubnetID	Vùng HostID	Broadcast
1	172.16.0.0	172.16.0.1 – 172.16.31.254	172.16.31.255
2	172.16.32.0	172.16.32.1 – 172.16.63.254	172.16.63.255
...	...	...	...
7	172.16.192.0	172.16.192.1 – 172.16.223.254	172.16.223.255
8	172.16.224.0	172.16.224.1 – 172.16.255.254	172.16.255.255

11111111 11111111 11000000 00000000

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

---

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

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

# DHCP: Dynamic Host Configuration Protocol

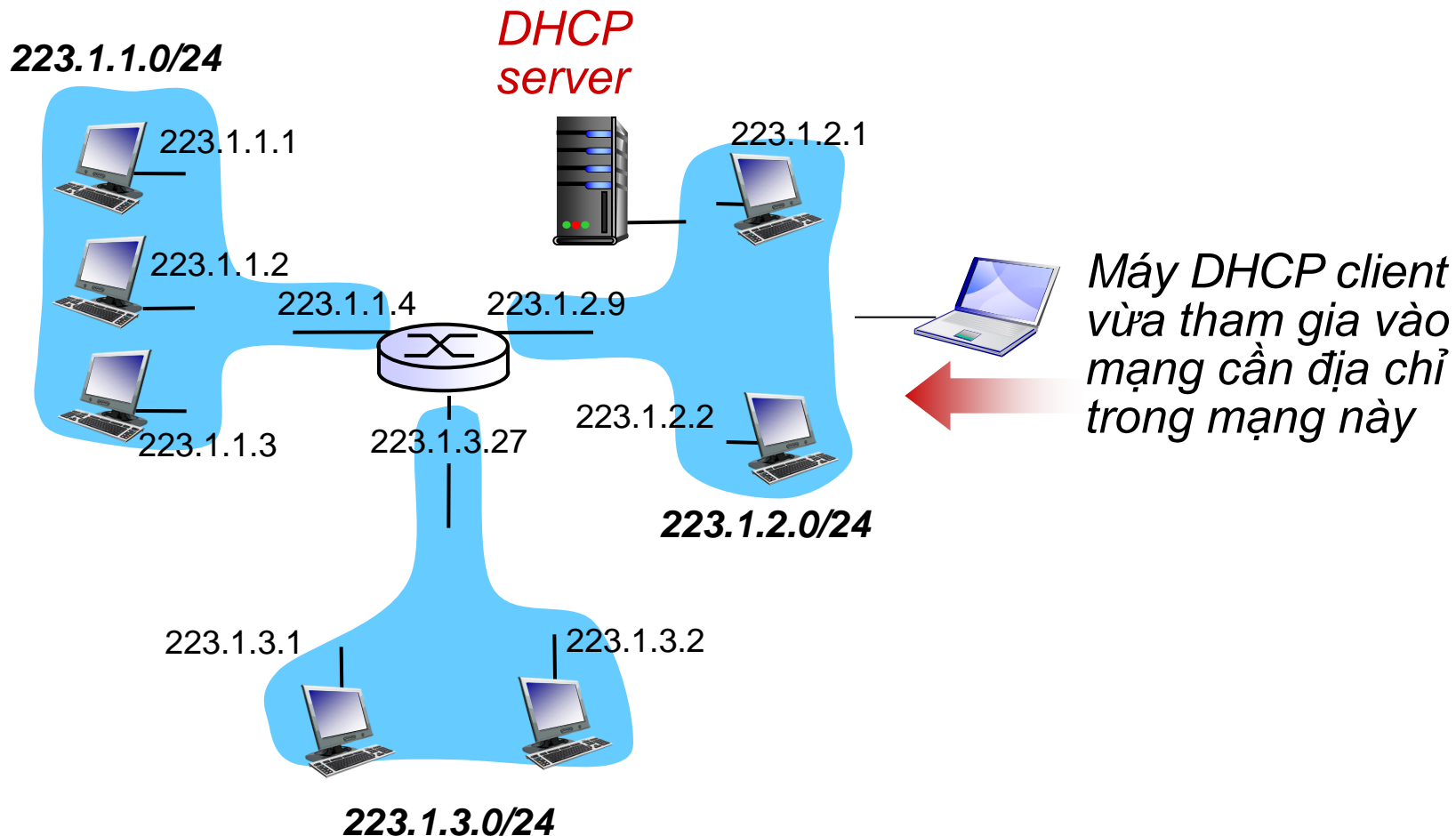
**Mục tiêu:** cho phép host (máy) tự động lấy địa chỉ IP của nó từ server trong mạng khi host đó tham gia vào mạng

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

## **Tổng quan DHCP :**

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

# Ngữ cảnh DHCP client-server





# Ngũ cảnh DHCP client-server

DHCP server: 223.1.2.5

DHCP discover

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

arriving  
client



DHCP offer

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

DHCP request

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

DHCP ACK

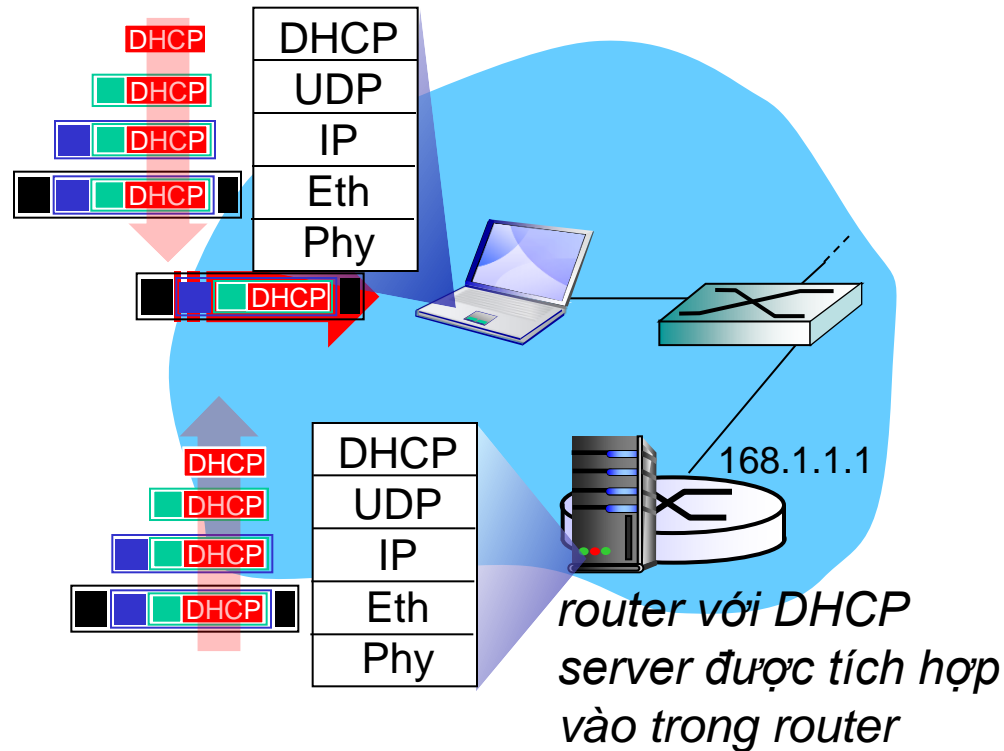
src: 223.1.2.5, 67  
dest: 255.255.255.255, 68  
yiaddr: 223.1.2.4  
transaction ID: 655  
lifetime: 3600 secs

# DHCP: cung cấp nhiều thông tin

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

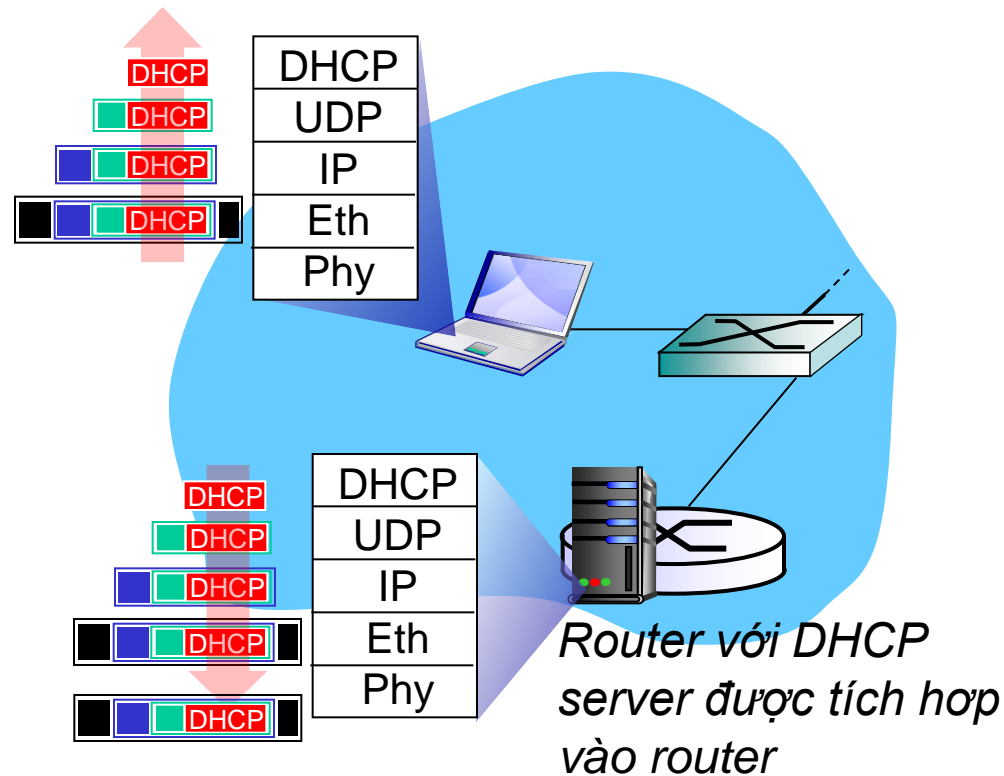
- Địa chỉ của router ở cửa ngõ kết nối ra ngoài mạng của client (default gateway)
- Tên và địa chỉ IP của DNS sever
- Network mask (cho biết phần của mạng và phần host của địa chỉ IP)

# DHCP: ví dụ



- ❖ Laptop tham gia vào mạng cần địa chỉ IP của nó, địa chỉ của default gateway, địa chỉ của DNS server: dùng DHCP
- ❖ DHCP request được đóng gói trong gói tin UDP, rồi được đóng gói trong gói IP, rồi được đóng gói trong gói 802.1 Ethernet
- ❖ Gói tin Ethernet frame được gửi broadcast (đích: FFFFFFFFFFFFFFFF) trên LAN, được nhận tại router đang chạy DHCP server
- ❖ Ethernet gửi gói lên IP, rồi lên UDP, rồi lên DHCP

# DHCP: ví dụ



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

# DHCP: Wireshark output (home LAN)

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

Hardware type: Ethernet

Hardware address length: 6

Hops: 0

**Transaction ID: 0x6b3a11b7**

Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)

Client IP address: 0.0.0.0 (0.0.0.0)

Your (client) IP address: 0.0.0.0 (0.0.0.0)

Next server IP address: 0.0.0.0 (0.0.0.0)

Relay agent IP address: 0.0.0.0 (0.0.0.0)

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

Server host name not given

Boot file name not given

Magic cookie: (OK)

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

Option: (61) Client identifier

Length: 7; Value: 010016D323688A;

Hardware type: Ethernet

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

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

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

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

Length: 11; Value: 010F03062C2E2F1F21F92B

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

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

44 = NetBIOS over TCP/IP Name Server

.....

Yêu cầu

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

Hardware type: Ethernet

Hardware address length: 6

Hops: 0

**Transaction ID: 0x6b3a11b7**

Seconds elapsed: 0

Bootp flags: 0x0000 (Unicast)

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

Your (client) IP address: 0.0.0.0 (0.0.0.0)

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

Relay agent IP address: 0.0.0.0 (0.0.0.0)

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

Server host name not given

Boot file name not given

Magic cookie: (OK)

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

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

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

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

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

Length: 12; Value: 445747E2445749F244574092;

IP Address: 68.87.71.226;

IP Address: 68.87.73.242;

IP Address: 68.87.64.146

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

Trả lời

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

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

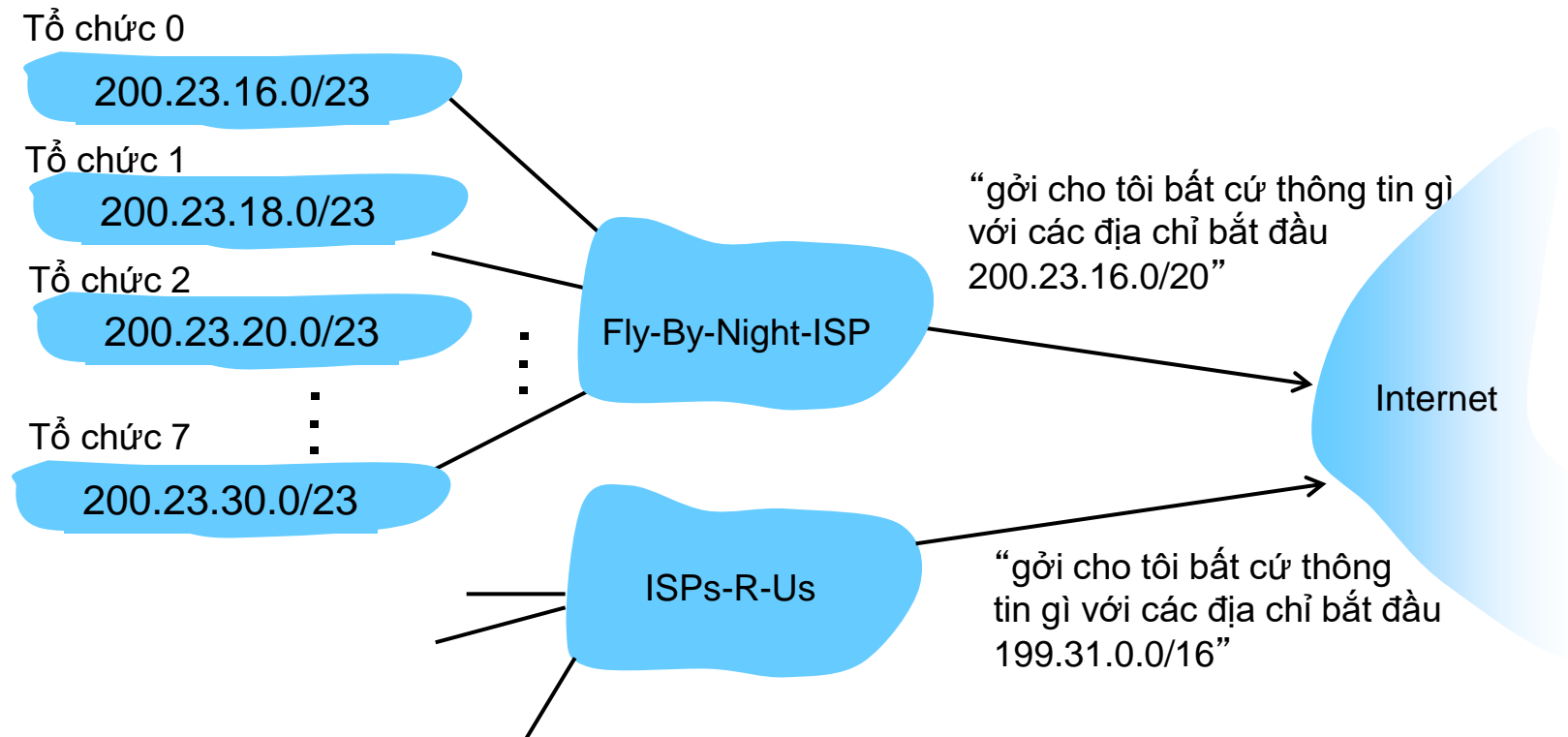
**Đáp:** Lấy phần đã được cấp phát của không gian địa chỉ IP do ISP cung cấp

Dãy địa chỉ của ISP      11001000 00010111 00010000 00000000  
200.23.16.0/20

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

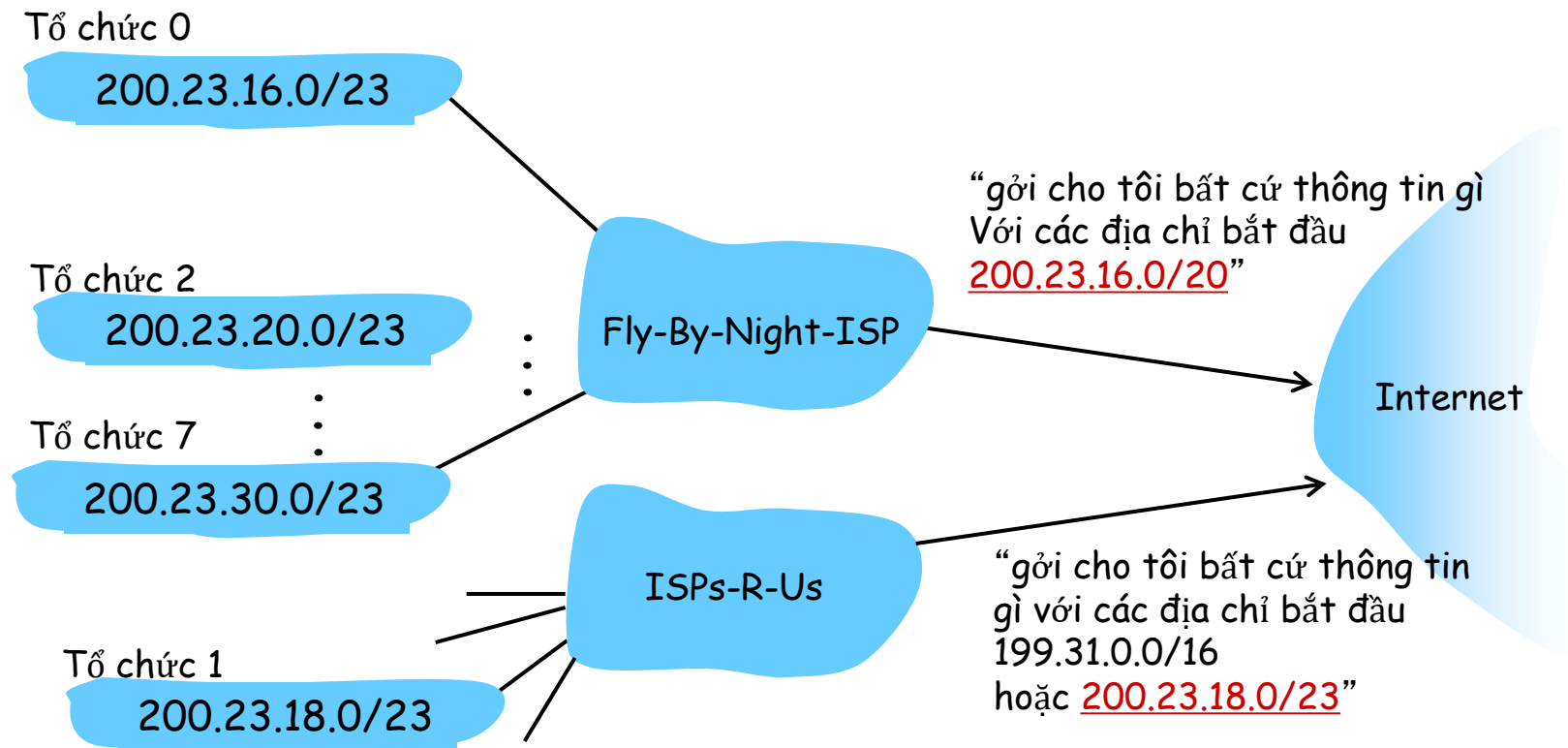
# Định địa chỉ phân cấp: định tuyến tích hợp

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



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

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





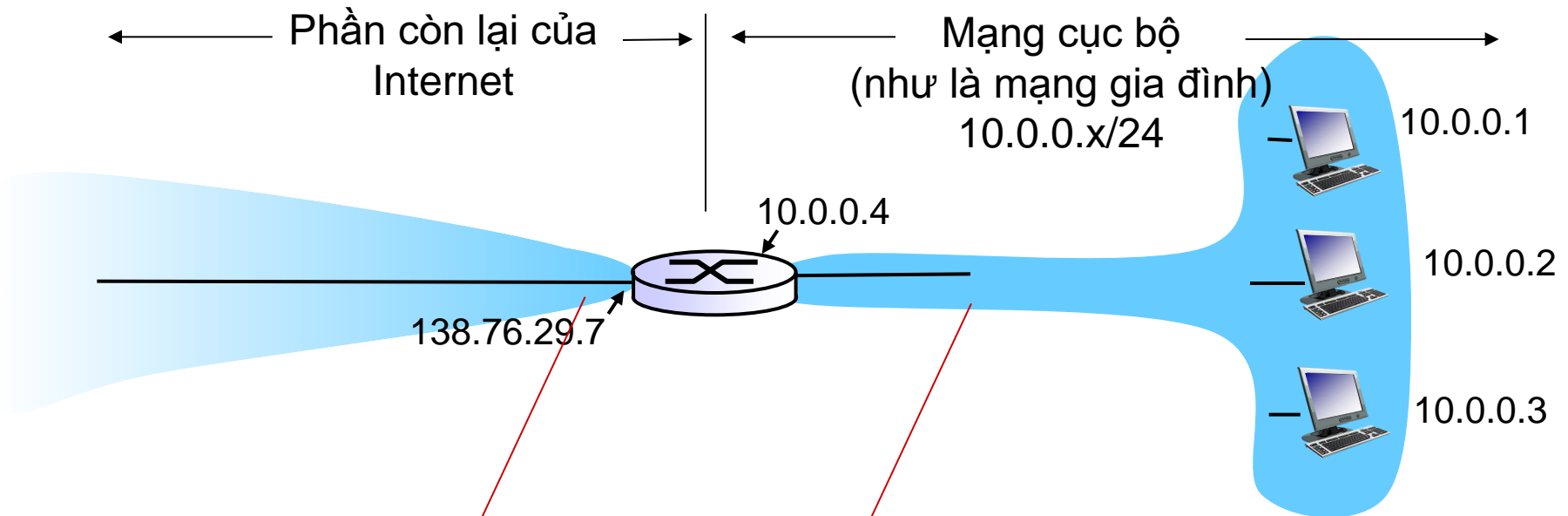
# Định địa chỉ IP: lời cuối...

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

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

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

# NAT: network address translation



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

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

# NAT: network address translation

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

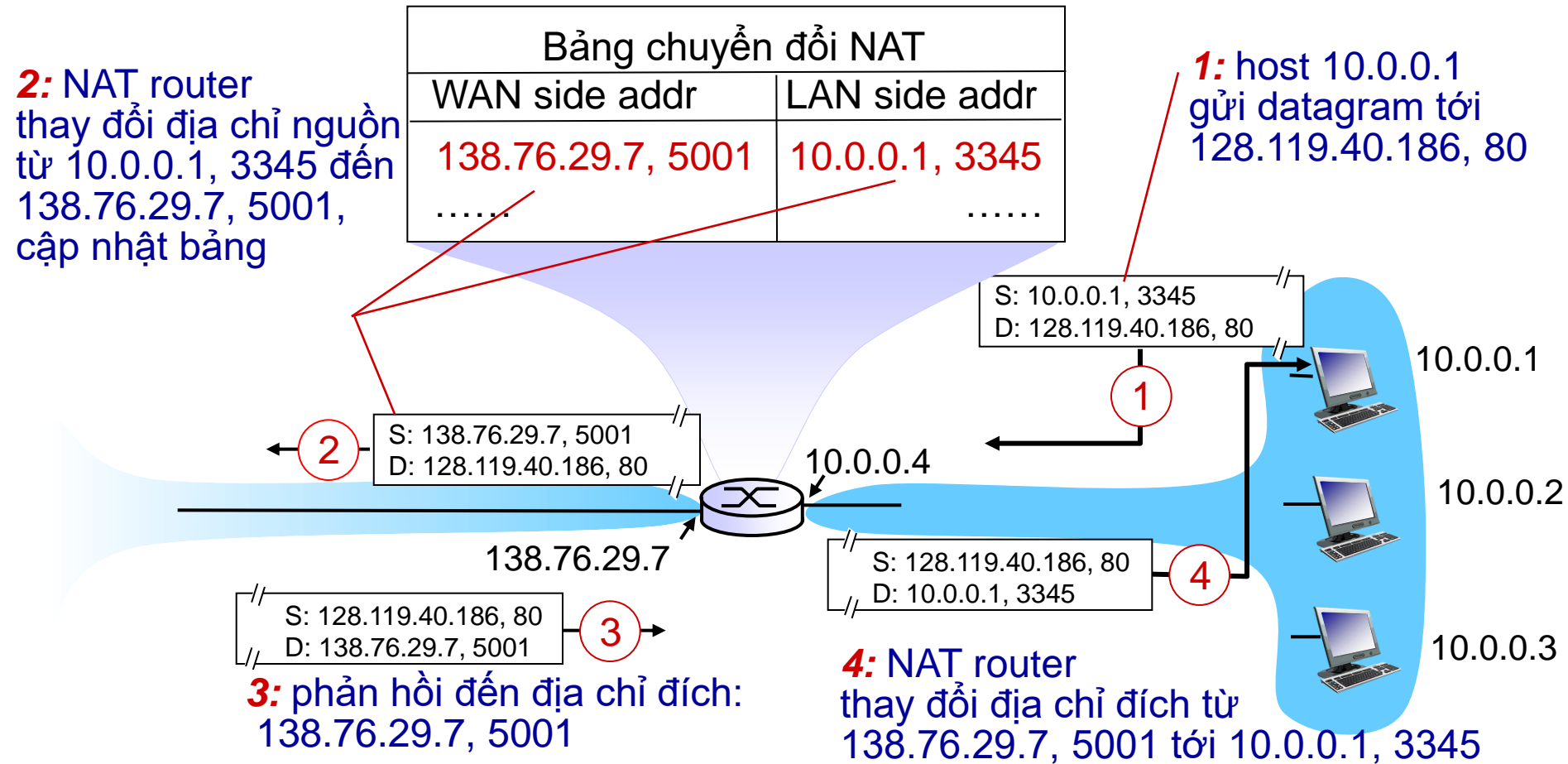
- Không cần thiết dùng 1 vùng địa chỉ từ ISP: chỉ cần 1 địa chỉ IP cho tất cả các thiết bị
- Có thể thay đổi các địa chỉ IP của các thiết bị trong mạng cục bộ mà không cần thông báo cho thế giới bên ngoài
- Có thể thay đổi ISP mà không cần thay đổi địa chỉ IP của các thiết bị trong mạng nội bộ
- Bên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

# NAT: network address translation

*Triển khai:* NAT router phải:

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

# NAT: network address translation

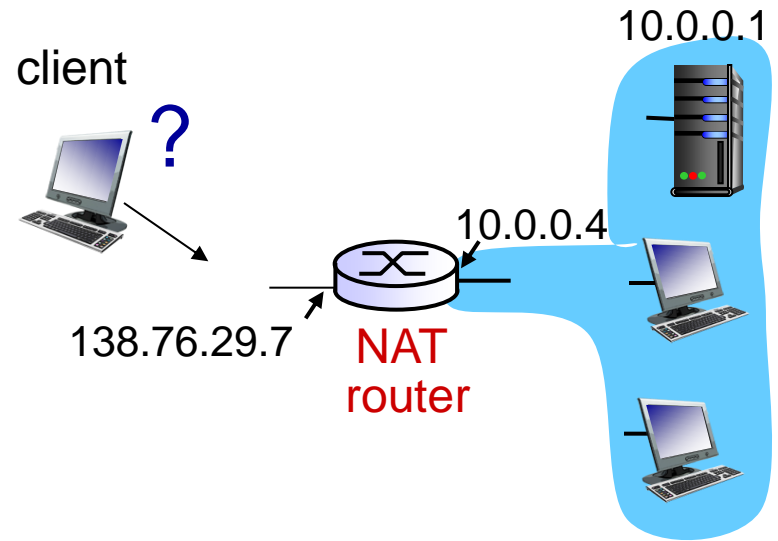


# NAT: network address translation

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

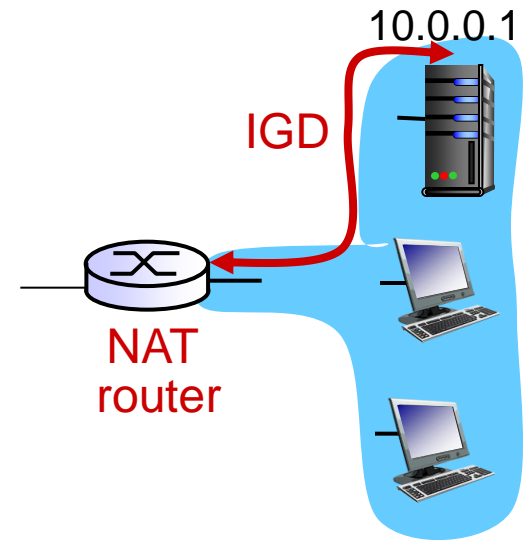
# Vấn đề chuyển gói NAT

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



# Vấn đề chuyển gói NAT

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





# Vấn đề chuyển gói NAT

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



# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing

# ICMP: Internet Control Message Protocol

- ❖ Được sử dụng bởi các host và router để truyền thông tin tầng Mạng

- Thông báo: host, network, port, giao thức không có thực
- Các dạng gói tin echo request/reply (được dùng bởi ping)

- ❖ Tầng Mạng “trên” IP:

- Các thông điệp ICMP được gửi trong các IP datagram

- ❖ **Thông điệp ICMP:** loại, mã cộng thêm 8 byte đầu tiên của IP datagram gây ra lỗi

<u>Loại</u>	<u>mã</u>	<u>Mô tả</u>
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

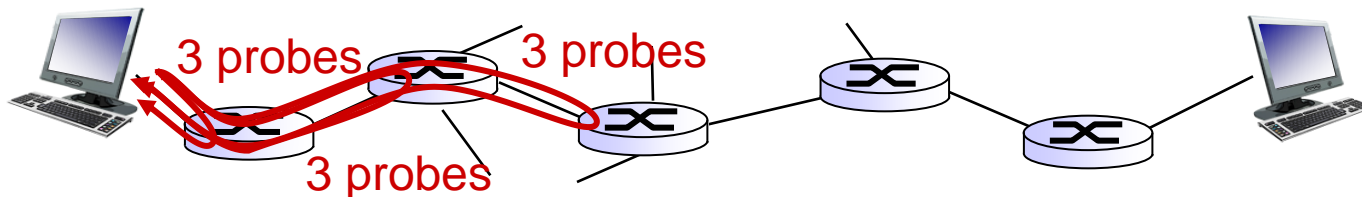
# Traceroute và ICMP

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

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

## *Tiêu chuẩn dừng:*

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



# IPv6: Động lực

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

## *Định dạng datagram IPv6 :*

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

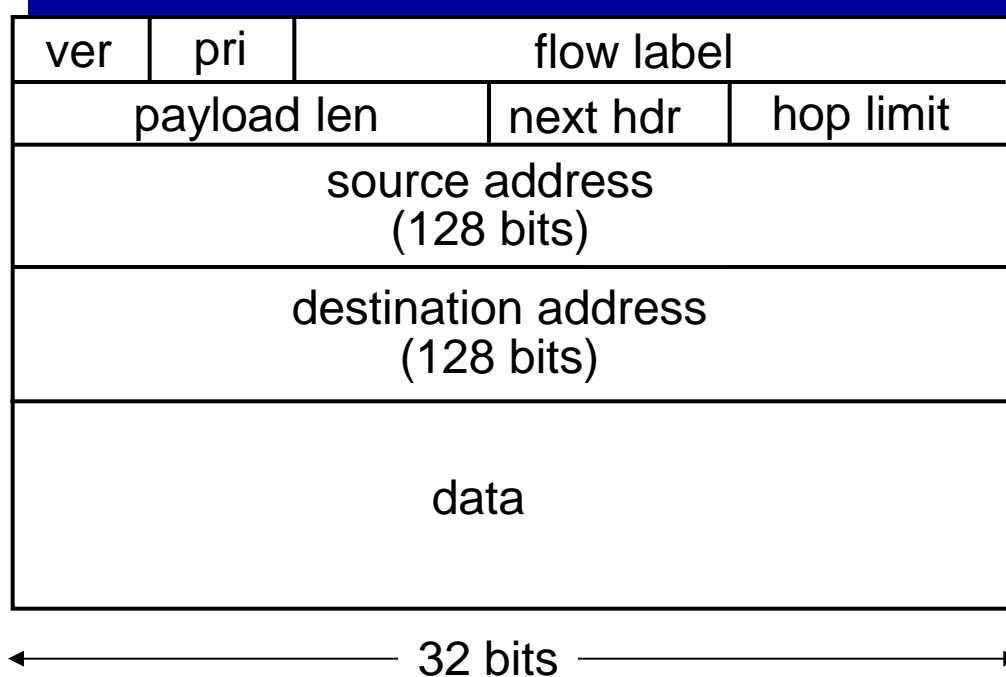
# Định dạng datagram IPv6

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

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

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

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



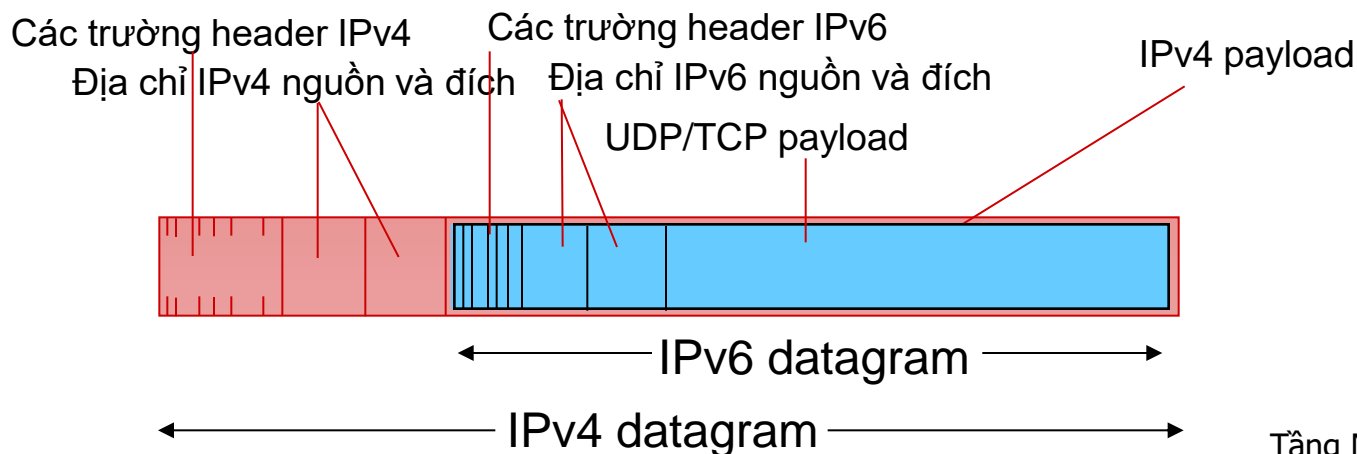
# Những thay đổi khác so với IPv4

---

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

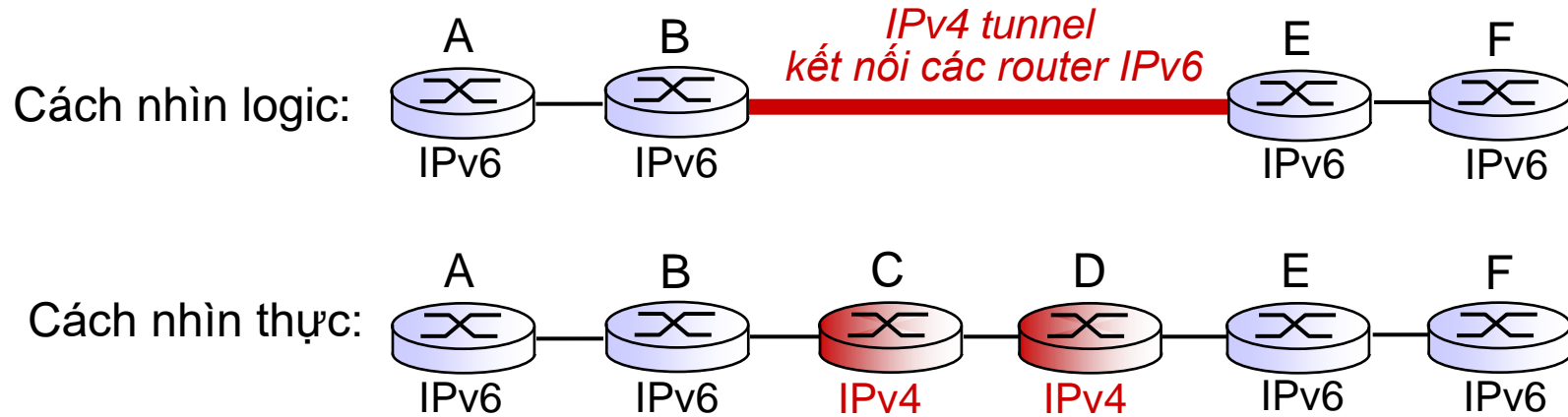
# Chuyển từ IPv4 sang IPv6

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

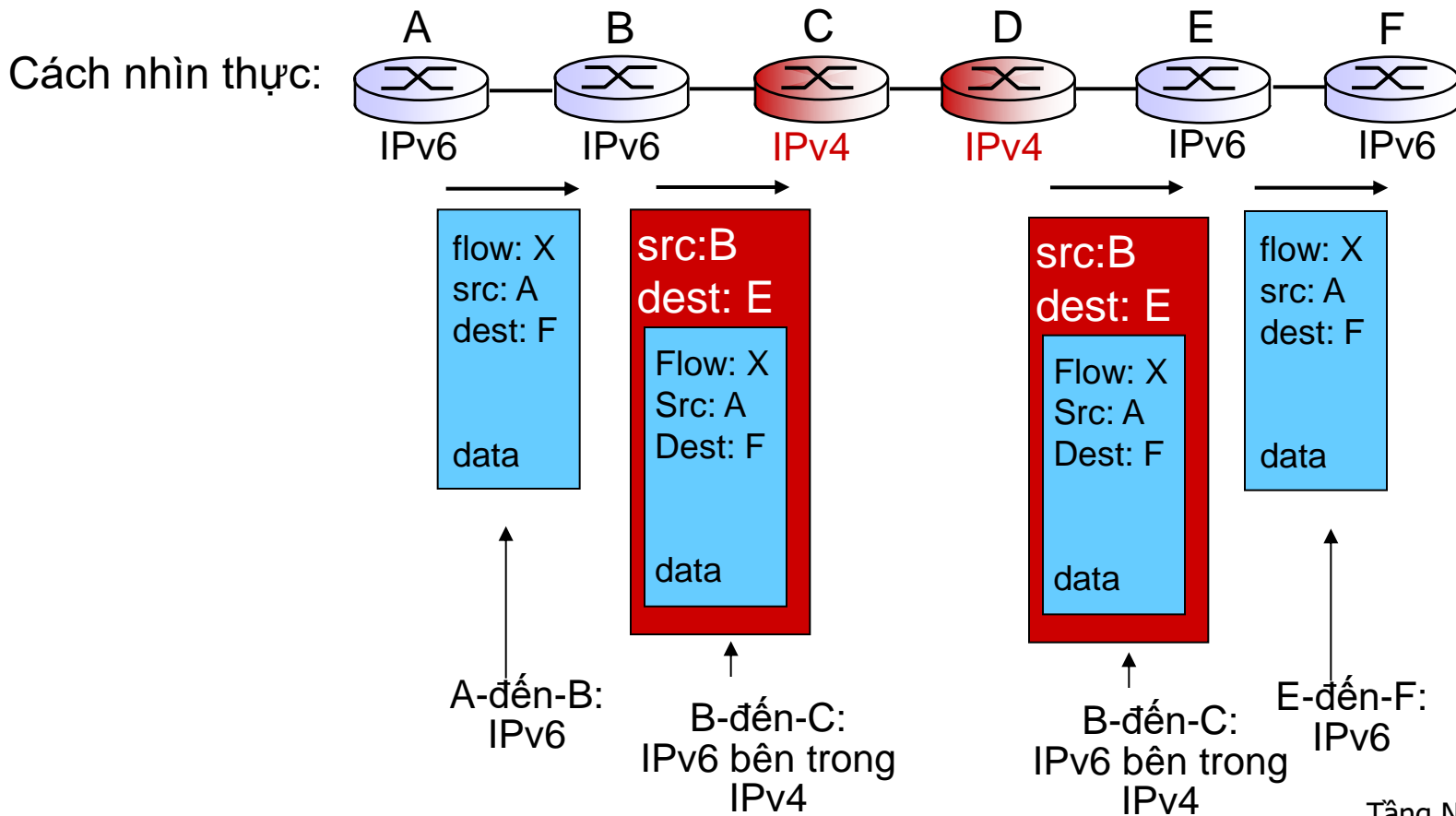




# Tunneling



# Tunneling



# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

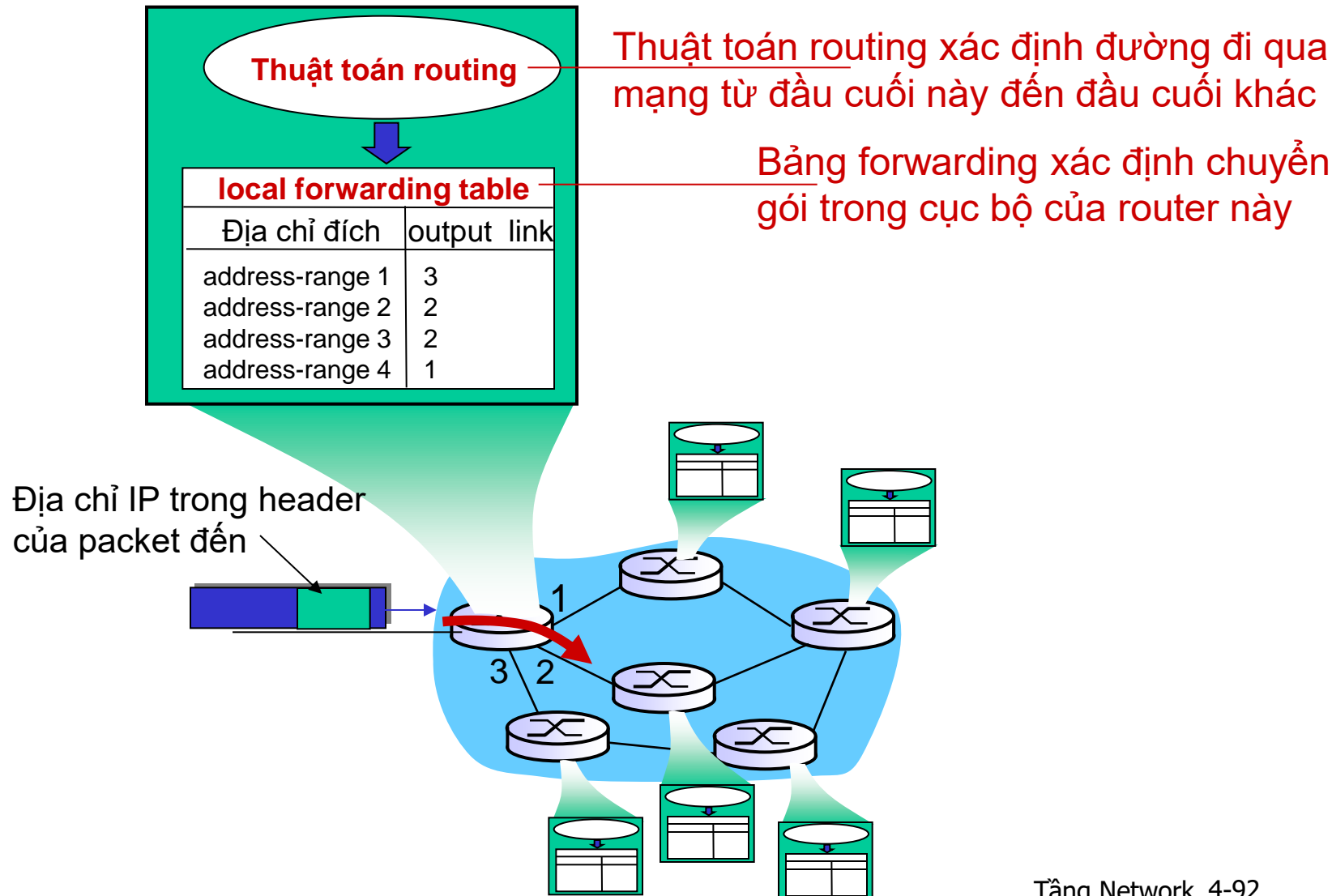
- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

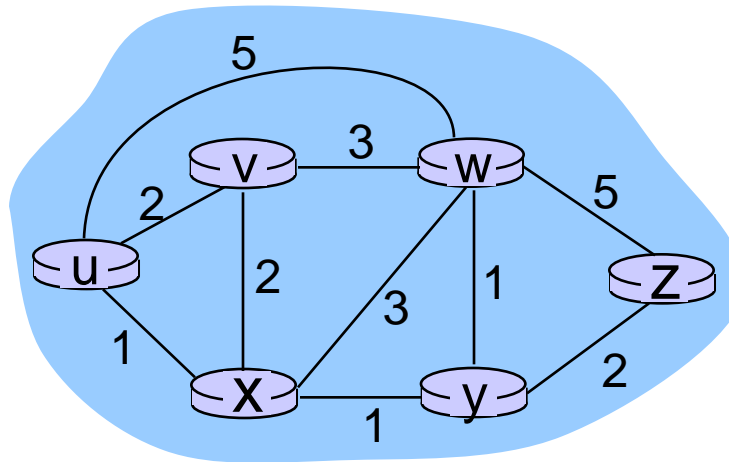
- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing

# Tác động lẫn nhau giữa routing và forwarding



# Mô hình đồ thị



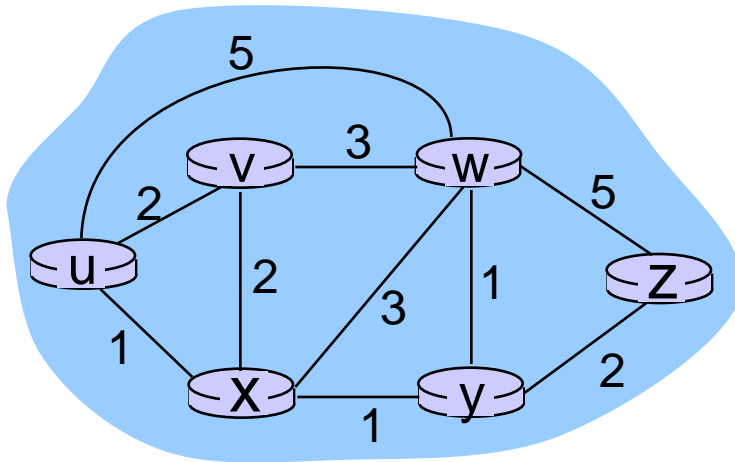
Đồ thị:  $G = (N, E)$

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

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

*Ghi chú:* mô hình đồ thị cũng hữu ích trong các ngữ cảnh khác.  
Ví dụ, P2P, trong đó  $N$  là tập các peer và  $E$  là tập các kết nối TCP

# Mô hình đồ thị: Chi phí



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

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

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

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

**Hỏi:** Chi phí đường đi thấp nhất từ u tới z?

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

# Phân loại thuật toán Routing

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

*Toàn cục:*

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

*Phân cấp:*

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

*Q: tĩnh hay động?*

*Tĩnh:*

- ❖ Các đường đi được cập nhật chậm theo thời gian

*Động:*

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

# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing



# Thuật toán routing Link-State

## *Thuật toán Dijkstra*

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

## *Ký hiệu:*

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

# Thuật toán Dijkstra

1 **Khởi tạo:**

2  $N' = \{u\}$

3 for tất cả các  $v$

4 if  $v$  liền kề với  $u$

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

6 else  $D(v) = \infty$

7

8 **Lặp**

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

10 thêm  $w$  vào tập  $N'$

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

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

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

14 cộng với chi phí từ  $w$  đến  $v$ \*/

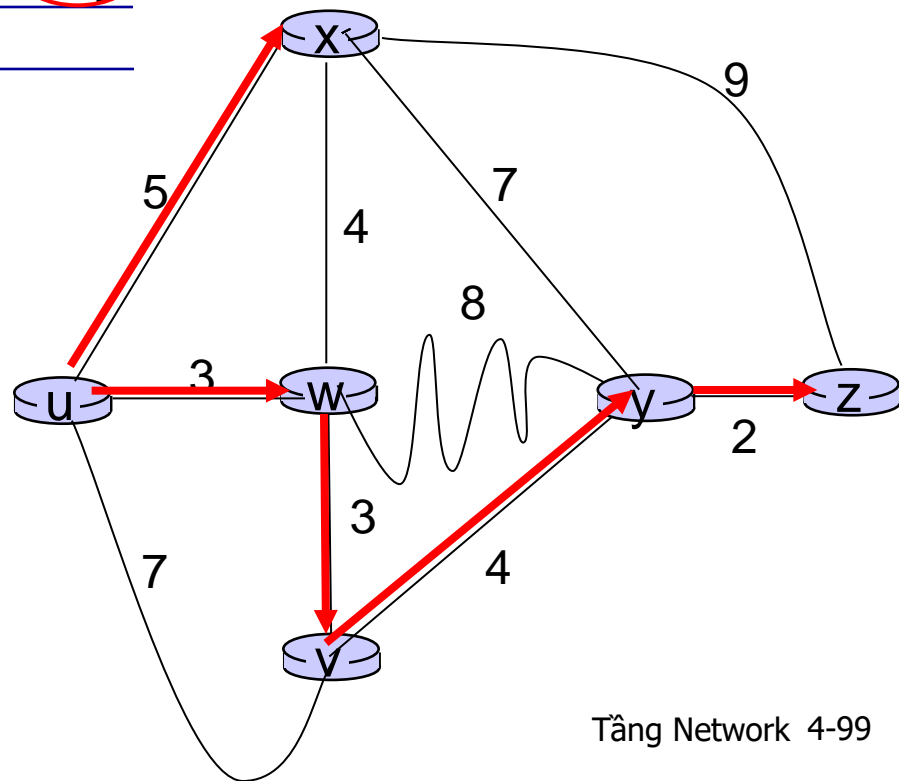
15 **Cho đến khi tất cả các node trong  $N'$**

# Thuật toán Dijkstra : ví dụ

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

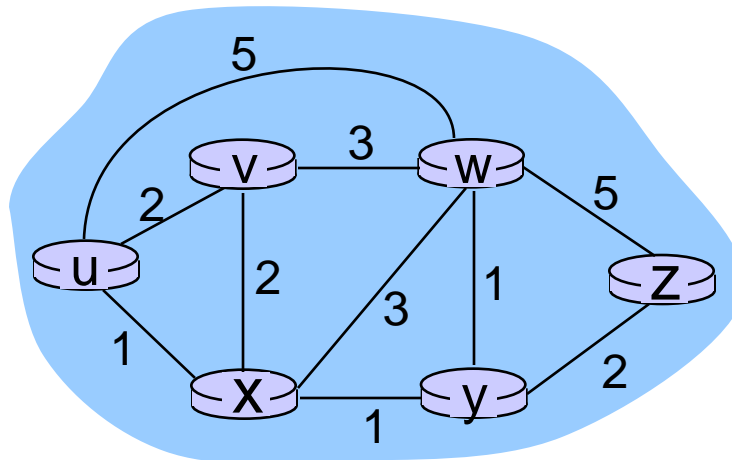
## *Ghi chú:*

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



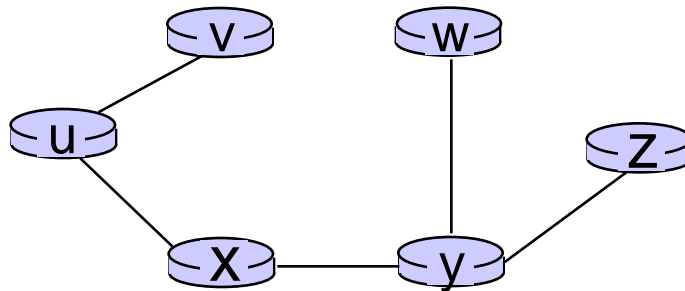
# Thuật toán Dijkstra: ví dụ

Bước	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	$\infty$	$\infty$
1	ux	2,u	4,x		2,x	$\infty$
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



# Thuật toán Dijkstra: ví dụ (2)

Kết quả cây đường đi ngắn nhất từ u:



Kết quả bảng forwarding trong u:

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

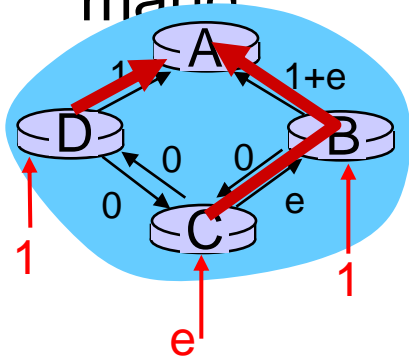
# Thuật toán Dijkstra, thảo luận

*Độ phức tạp của thuật toán:* n nodes

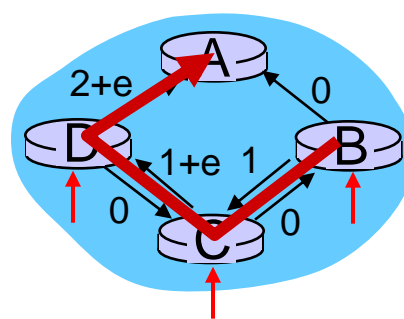
- ❖ Mỗi lần duyệt: cần kiểm tra tất cả các node, w, không có trong N
- ❖  $n(n+1)/2$  phép so sánh:  $O(n^2)$
- ❖ Có nhiều cách thực hiện hiệu quả hơn:  $O(n \log n)$

*Có thể dao động:*

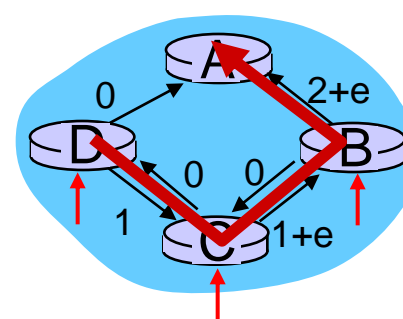
- ❖ Ví dụ: chi phí kết nối bằng với số lượng lưu thông được mang



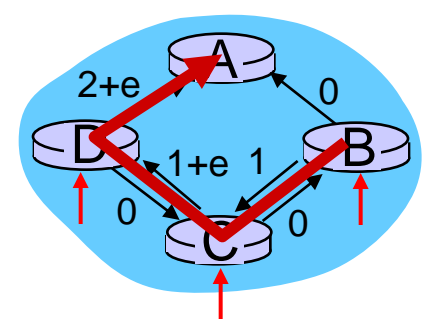
Khởi tạo



Cho các chi phí,  
Tìm định tuyến mới....  
Kết quả trong chi phí mới



Cho các chi phí,  
Tìm định tuyến mới....  
Kết quả trong chi phí mới



Cho các chi phí,  
Tìm định tuyến mới....  
Kết quả trong chi phí mới

# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing

# Thuật toán Distance vector

*Công thức Bellman-Ford (dynamic programming)*

cho

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

thì

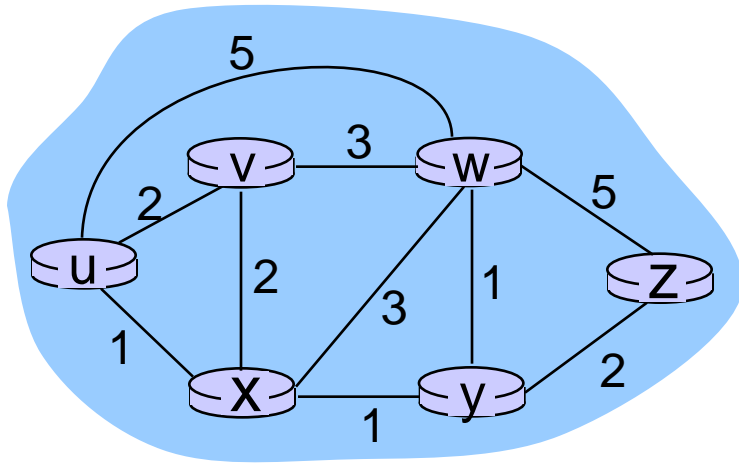
$$d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$$

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

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



# Bellman-Ford ví dụ



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

Biểu thức B-F cho kết quả:

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

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

# Thuật toán Distance vector

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

# Thuật toán Distance vector

## *Ý tưởng chính:*

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

F:

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

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

# Thuật toán Distance vector

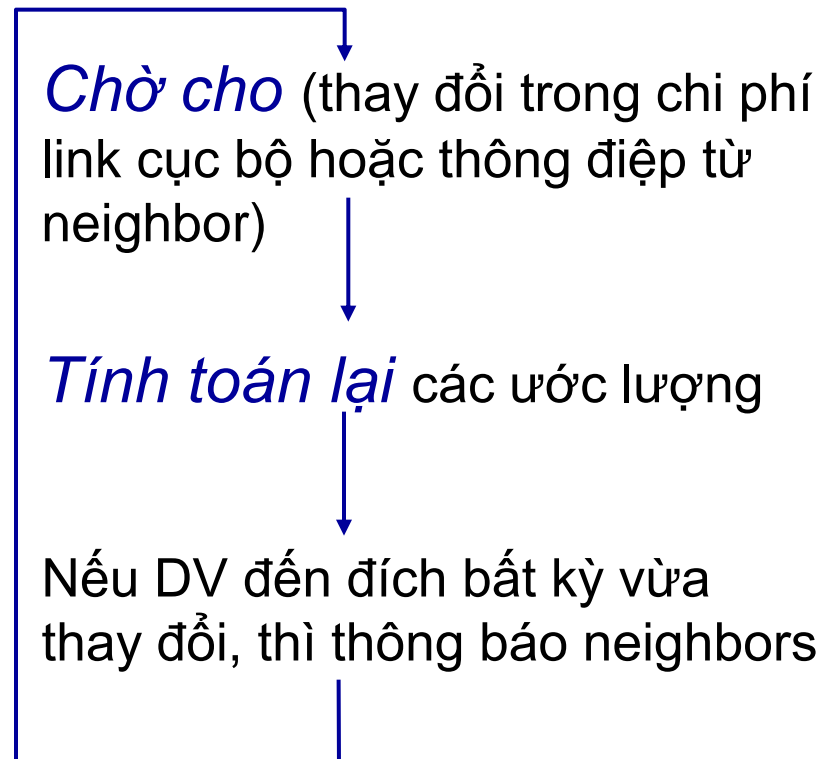
**Lặp, không đồng bộ:** mỗi  
lặp cục bộ tại 1 node được  
gây ra bởi:

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

**Phân bố:**

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

**Mỗi node:**



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

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

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

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

**Bảng  
node x**

		Chi phí đến		
		x	y	z
Từ	x	0	2	7
	y	$\infty$	$\infty$	$\infty$
	z	$\infty$	$\infty$	$\infty$

Chi phí đến

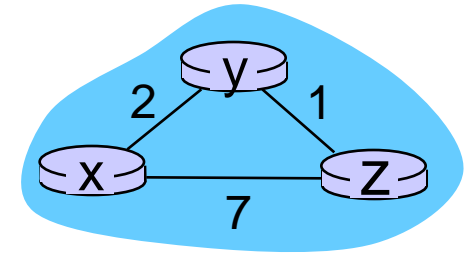
		x	y	z
Từ	x	0	2	3
	y	2	0	1
	z	7	1	0

**Bảng  
node y**

		Chi phí đến		
		x	y	z
Từ	x	$\infty$	$\infty$	$\infty$
	y	2	0	1
	z	$\infty$	$\infty$	$\infty$

**Bảng  
node z**

		Chi phí đến		
		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  
node x**

	chi phí đến		
	x	y	z
x	0	2	7
từ y	$\infty$	$\infty$	$\infty$
z	$\infty$	$\infty$	$\infty$

**Bảng  
node y**

	chi phí đến		
	x	y	z
x	$\infty$	$\infty$	$\infty$
từ y	2	0	1
z	$\infty$	$\infty$	$\infty$

**Bảng  
node z**

	chi phí đến		
	x	y	z
x	$\infty$	$\infty$	$\infty$
từ y	$\infty$	$\infty$	$\infty$
z	7	1	0

	chi phí đến		
	x	y	z
x	0	2	3
từ y	2	0	1
z	7	1	0

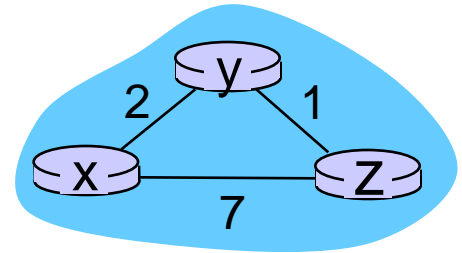
	chi phí đến		
	x	y	z
x	0	2	7
từ y	2	0	1
z	7	1	0

	chi phí đến		
	x	y	z
x	0	2	7
từ y	2	0	1
z	3	1	0

	chi phí đến		
	x	y	z
x	0	2	3
từ y	2	0	1
z	3	1	0

	chi phí đến		
	x	y	z
x	0	2	3
từ y	2	0	1
z	3	1	0

	chi phí đến		
	x	y	z
x	0	2	3
từ y	2	0	1
z	3	1	0

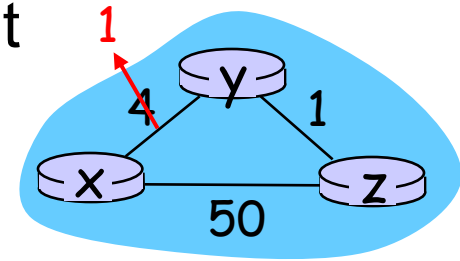


Thời gian

# Distance vector: chi phí kết nối thay đổi

## Chi phí kết nối thay đổi:

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



“tin  
tốt  
đi  
nhanh”

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

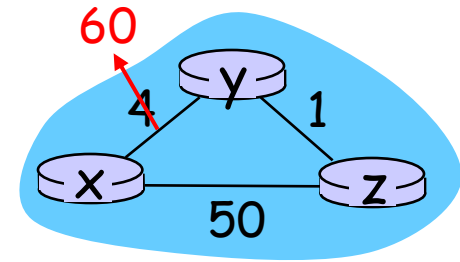
$t_1$ : z nhận được cập nhật từ y, và cập nhật bảng của nó, tính chi phí mới thấp nhất đến x, gửi DV của nó đến các neighbor của nó.

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

# Distance vector: Chi phí kết nối thay đổi

## *Chi phí kết nối thay đổi:*

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



## *Đầu độc ngược (Poisoned reverse):*

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



# So sánh giữa thuật toán LS và DV

## *Độ phức tạp của thông điệp*

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

## *Tốc độ hội tụ*

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

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

### **LS:**

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

### **DV:**

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

# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing

# Định tuyến có cấu trúc (Hierarchical routing)

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

- ❖ Tất cả các router là đồng nhất
- ❖ Mạng “phẳng”

*... không đúng trong thực tế*

**Quy mô:** với 600 triệu đích đến:

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

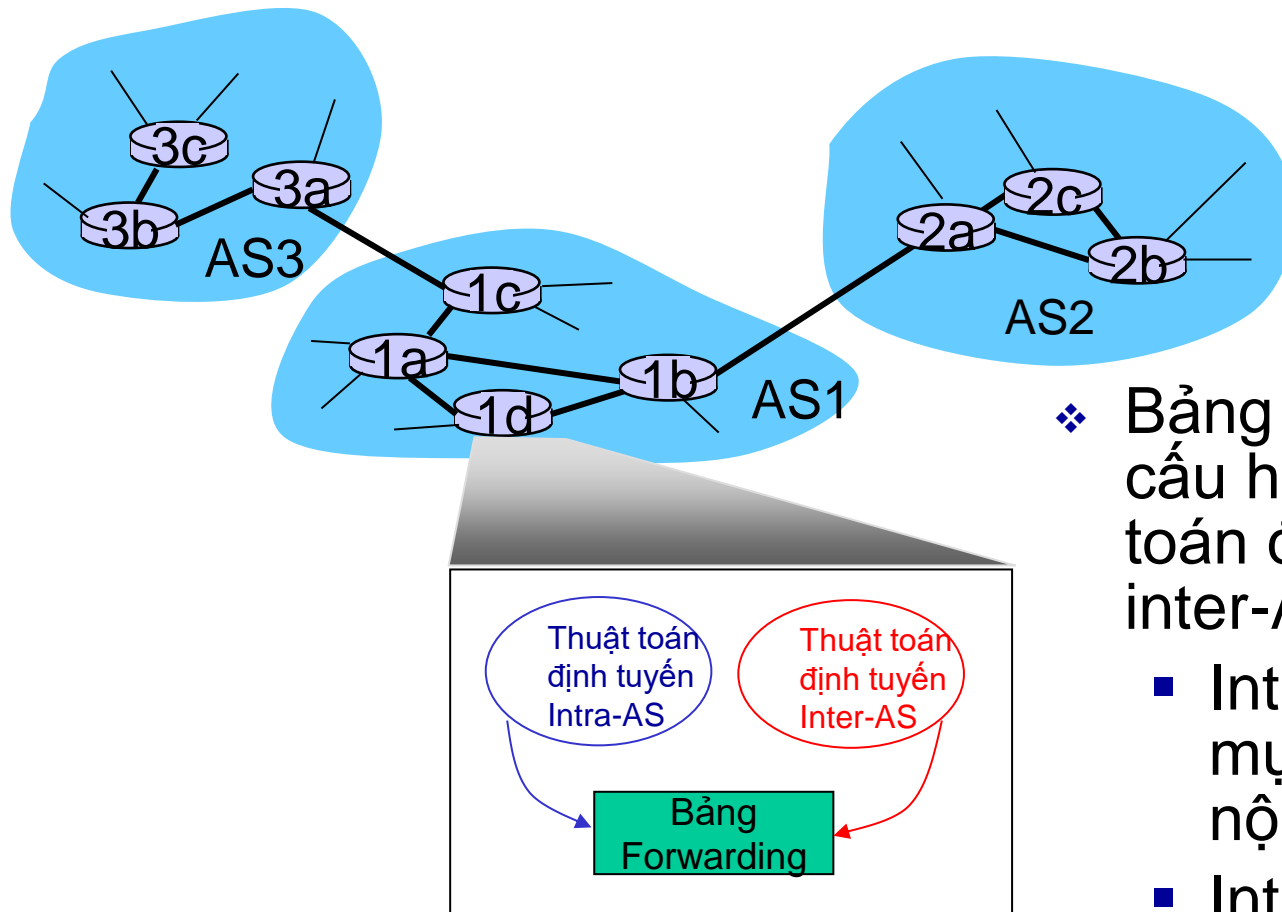
**Quản trị**

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

# Định tuyến có cấu trúc

- ❖ Các router được gom vào các vùng tự trị “autonomous systems” (AS)
  - ❖ Các router trong cùng AS chạy cùng giao thức định tuyến với nhau
    - Giao thức định tuyến “intra-AS”
    - Các router trong các AS khác nhau có thể chạy các giao thức định tuyến intra-AS khác nhau
- Gateway router:*
- ❖ Tại “biên” (“edge”) của AS của nó
  - ❖ Có liên kết đến router trong AS khác

# Kết nối các AS



- ❖ Bảng forwarding được cấu hình bởi cả thuật toán định tuyến intra- và inter-AS
  - Intra-AS thiết lập các mục cho các đích đến nội mạng
  - Inter-AS & intra-AS thiết lập các mục cho các đích đến ngoại mạng

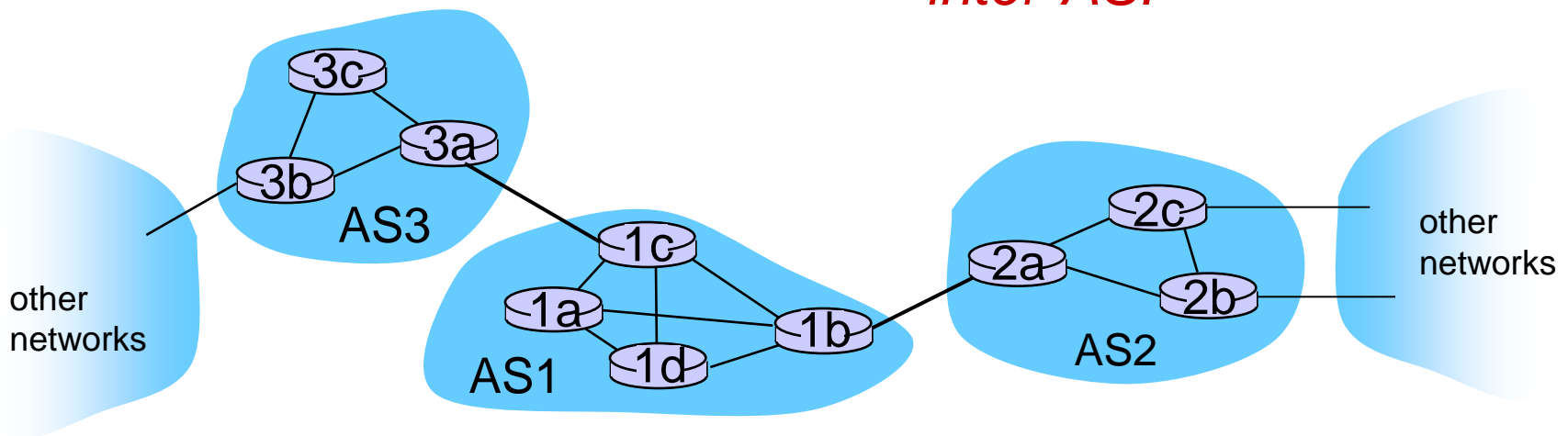
# Tác vụ Inter-AS

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

*AS1 phải:*

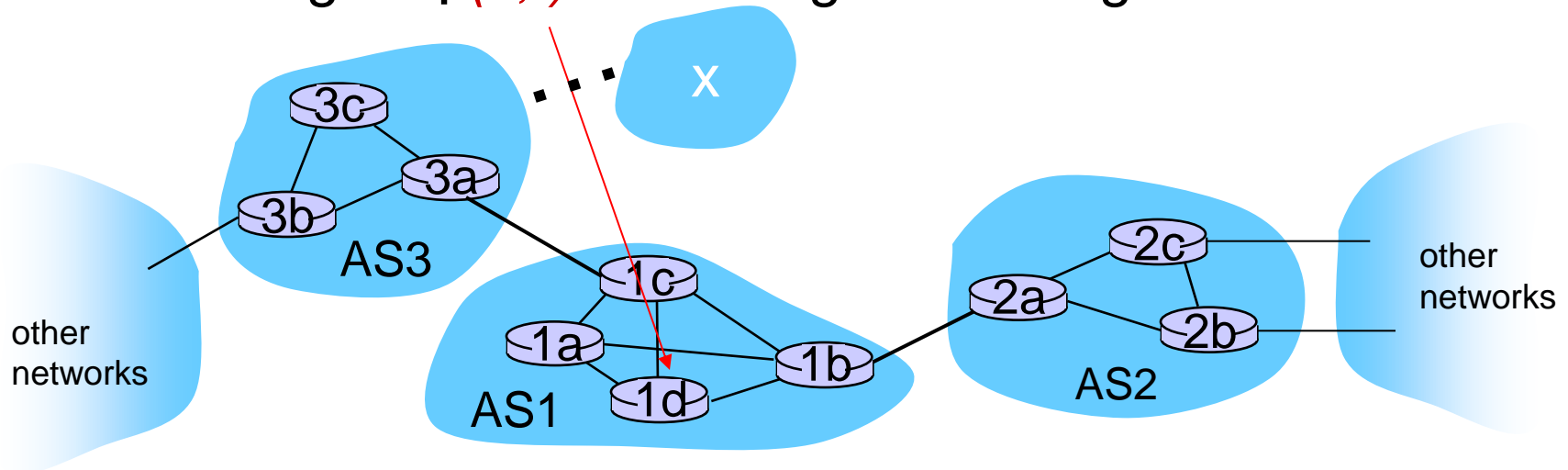
1. Học các đích đến nào có thể tới được thông qua AS2 và AS3
2. Lan truyền thông tin này đến tất cả các router trong AS1

*Công việc của định tuyến inter-AS!*



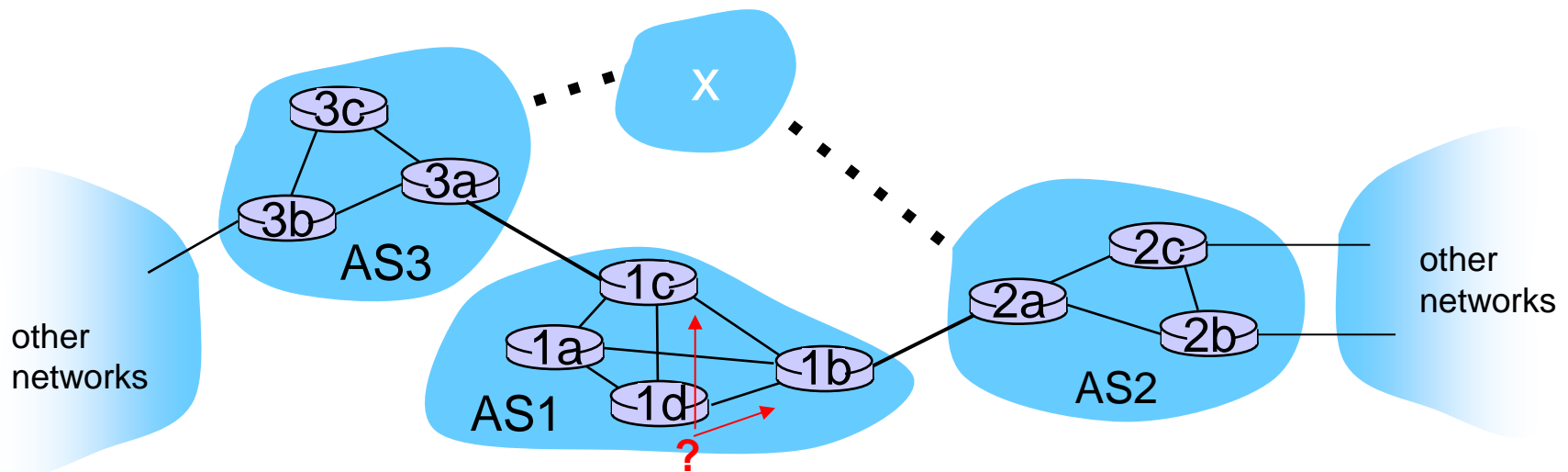
# Ví dụ: thiết lập bảng forwarding trong router 1d

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



# Ví dụ: chọn giữa nhiều AS

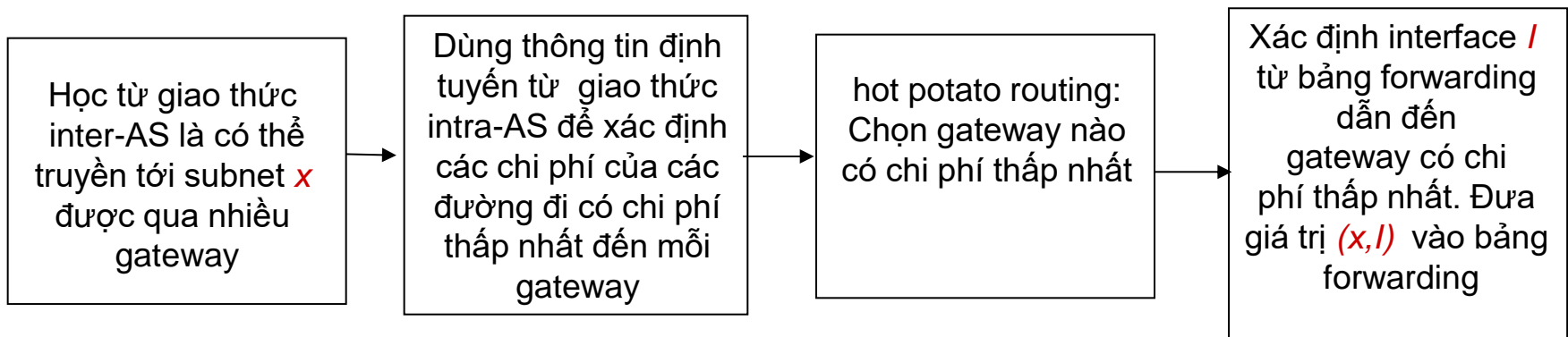
- ❖ Bây giờ, giả sử AS1 học từ giao thức inter-AS là có thể truyền tới subnet **x** qua AS3 và qua AS2.
- ❖ Để cấu hình bảng forwarding, router 1d phải xác định gateway nào mà nó nên dùng để chuyển các packet đến đích **x**
  - Đây cũng là công việc của giao thức định tuyến inter-AS!





# Ví dụ: chọn giữa nhiều AS

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



# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

- RIP
- OSPF
- BGP

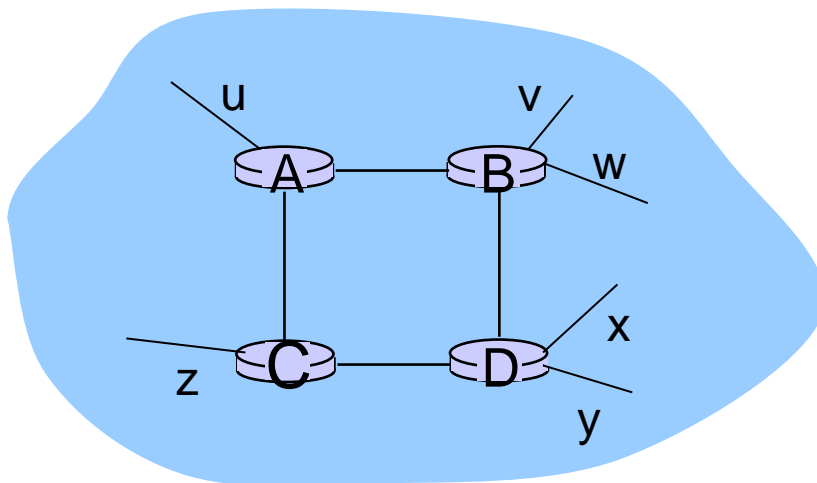
## 4.7 Broadcast và multicast routing

# Định tuyến Intra-AS

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

# RIP ( Routing Information Protocol)

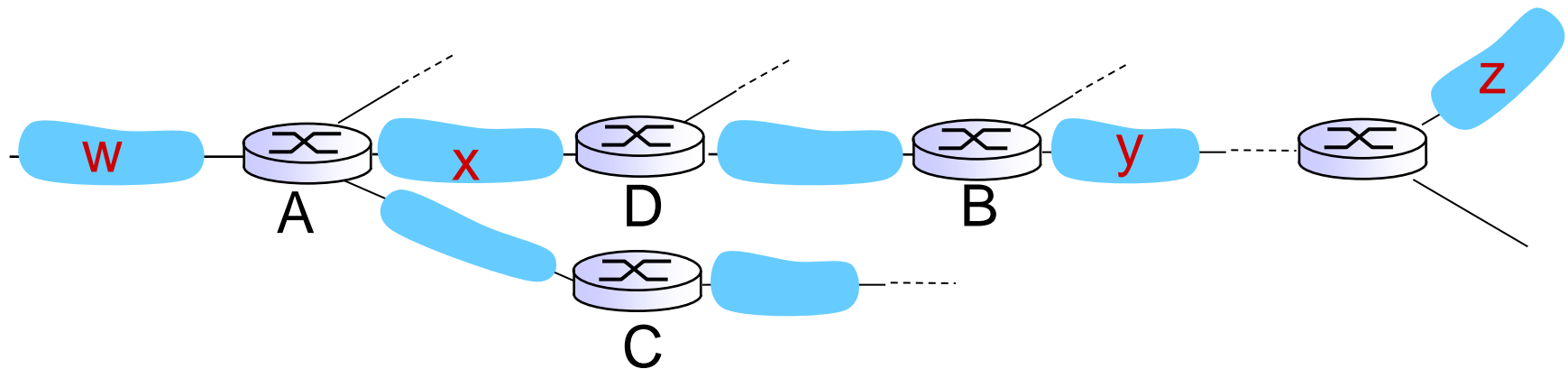
- ❖ Công bố vào năm 1982 trong BSD-UNIX
- ❖ Thuật toán distance vector
  - Đơn vị đo khoảng cách: số lượng hop (max = 15 hops), mỗi link có giá trị là 1
  - Các DV được trao đổi giữa các neighbors mỗi 30 giây trong thông điệp phản hồi (còn gọi là **advertisement**)
  - Mỗi advertisement: liệt kê lên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



Từ router A đến các **subnet** đích:

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

# RIP: ví dụ



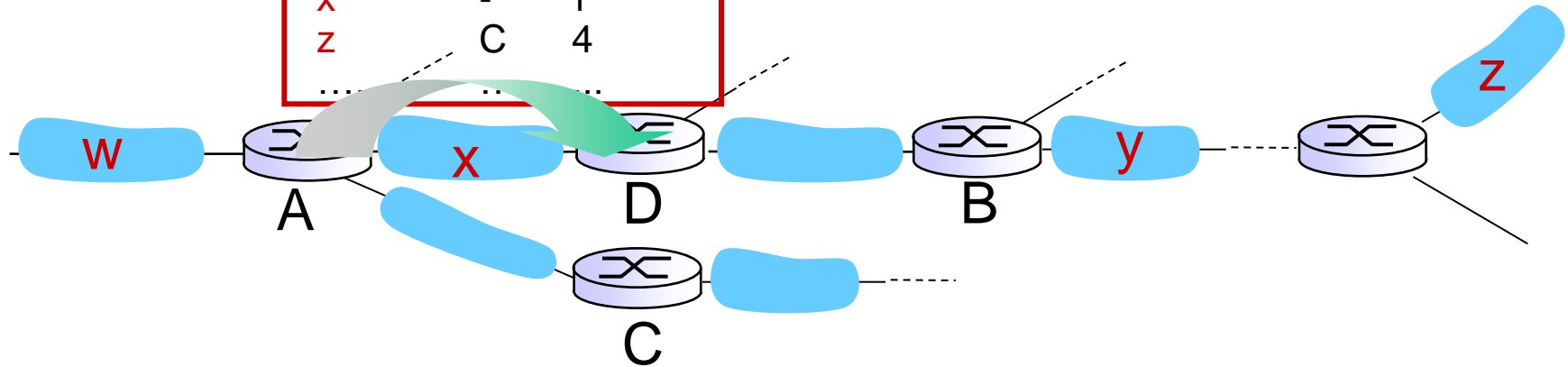
Bảng định tuyến trong router D

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

# RIP: ví dụ

Quảng cáo từ A-tới-D

đích	kế	hops
w	-	1
x	-	1
z	C	4
...	...	...



Bảng định tuyến trong router D

Subnet đích	router kế tiếp	số lượng hop đến đích
w	A	2
y	B	2
z	<del>B</del> → A	<del>7</del> → 5
x	--	1
....	....	....

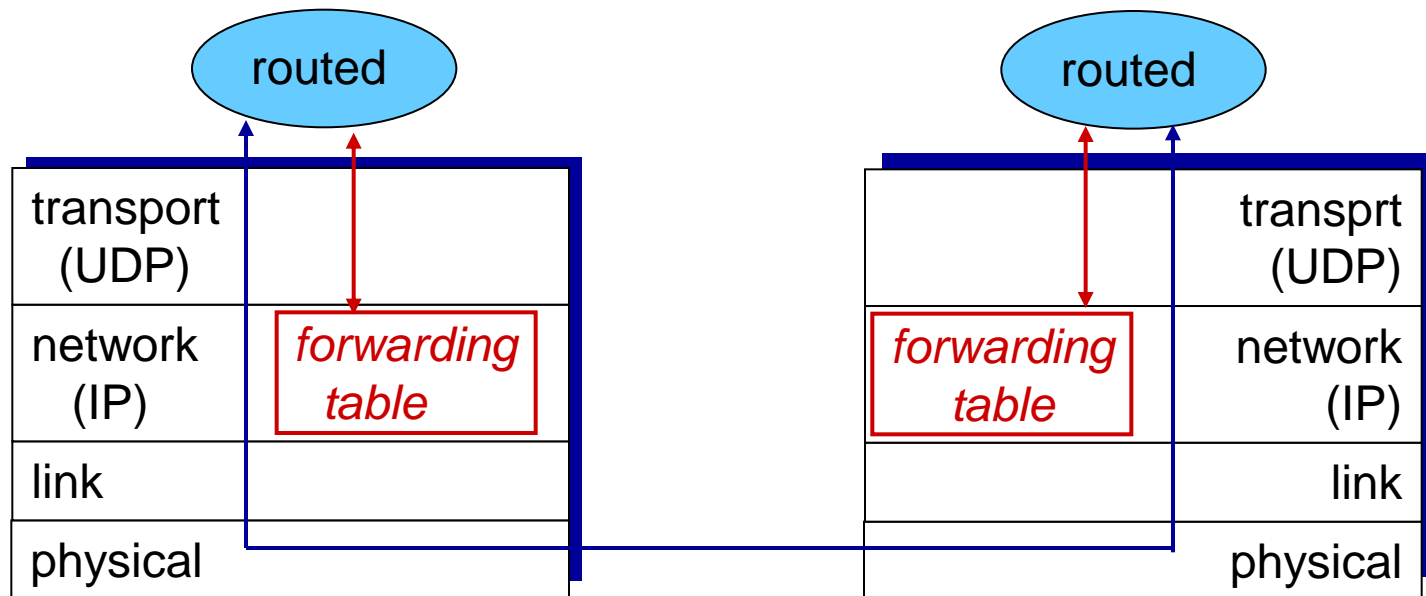
# RIP: lỗi đường kết nối và phục hồi

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

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

# RIP: xử lý bảng

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





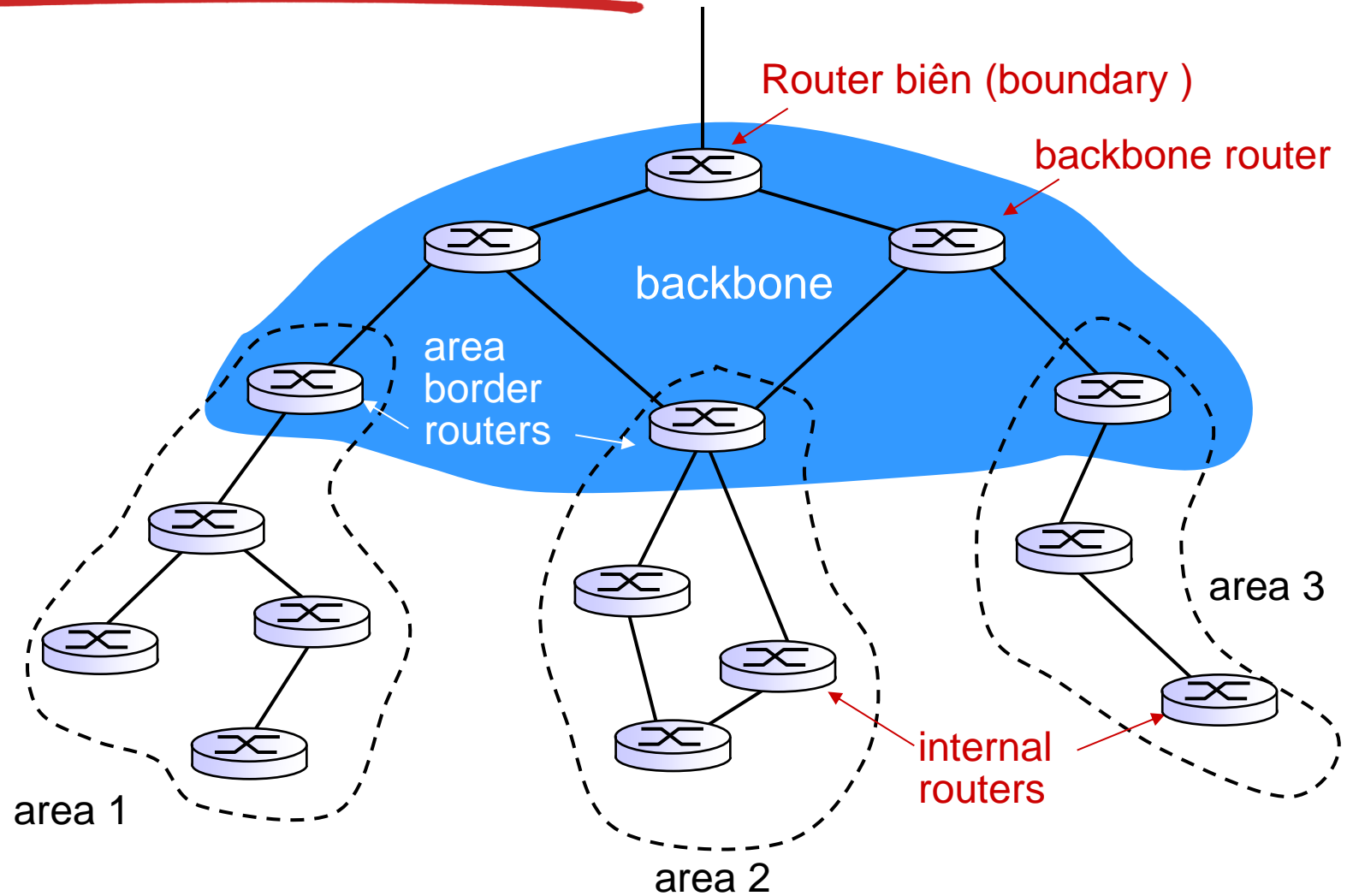
# OSPF (Open Shortest Path First)

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

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

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

# OSPF phân cấp



# OSPF phân cấp

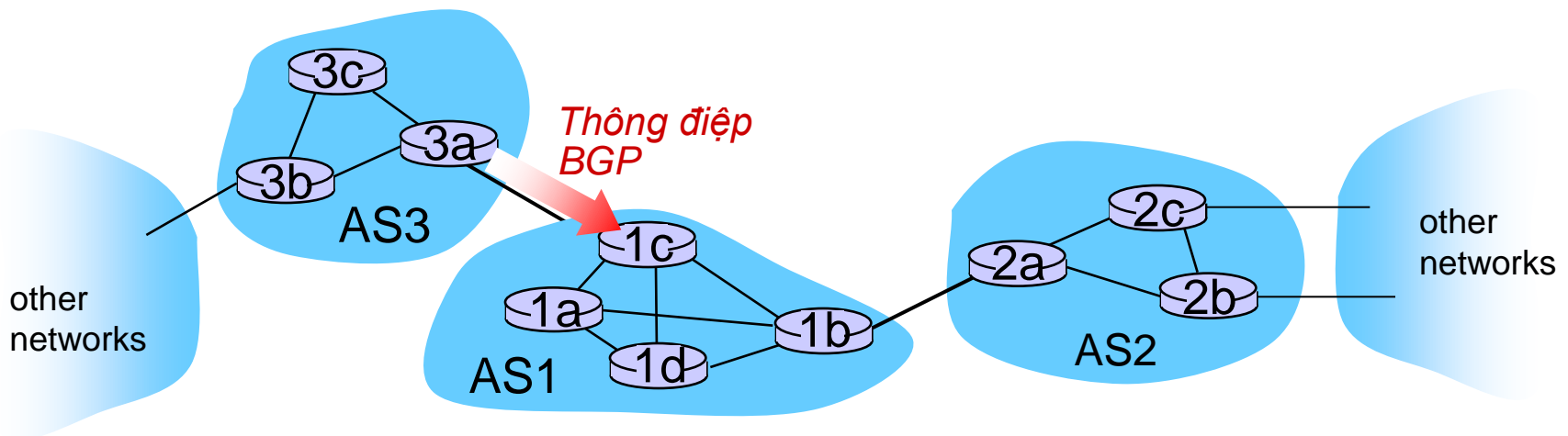
- ❖ *Phân 2 cấp*: vùng cục bộ (local area), vùng backbone.
  - Các quảng cáo link-state chỉ trong 1 vùng
  - Mỗi node có chi tiết cấu trúc của vùng; chỉ biết hướng (đường ngắn nhất) đến các mạng trong các vùng khác.
- ❖ *Các router tại biên vùng (Area Border routers)* : “tóm tắt” các khoảng cách đến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

# Định tuyến Internet inter-AS: BGP

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

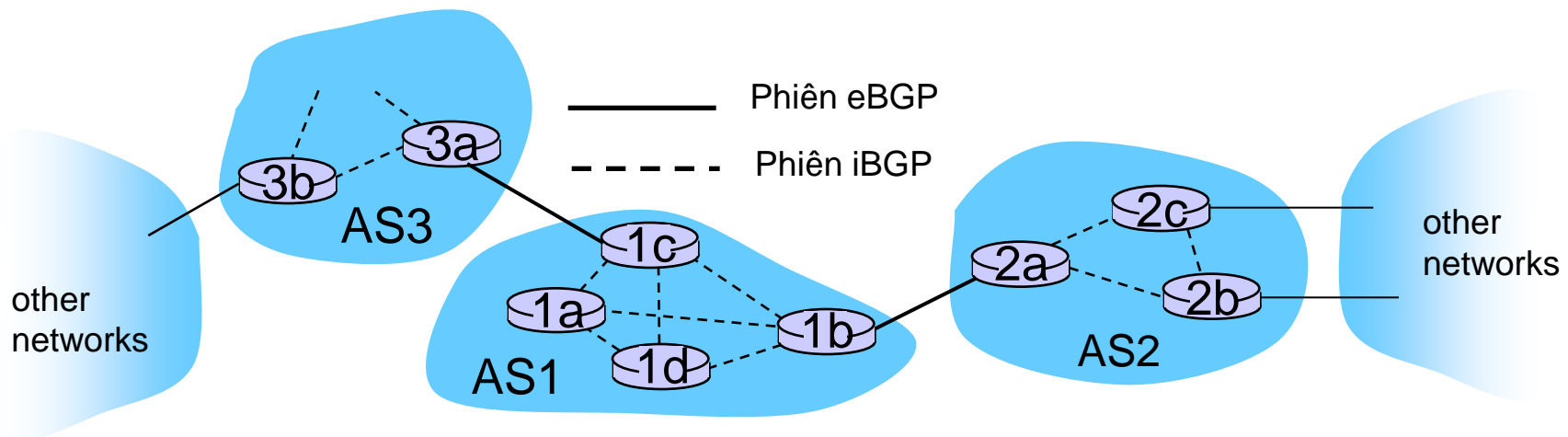
# BGP cơ bản

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



# BGP cơ bản: phân phối thông tin đường đi

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



# Các thuộc tính đường đi và các đường BGP

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



# Sự lựa chọn BGP route

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

# Các thông điệp BGP

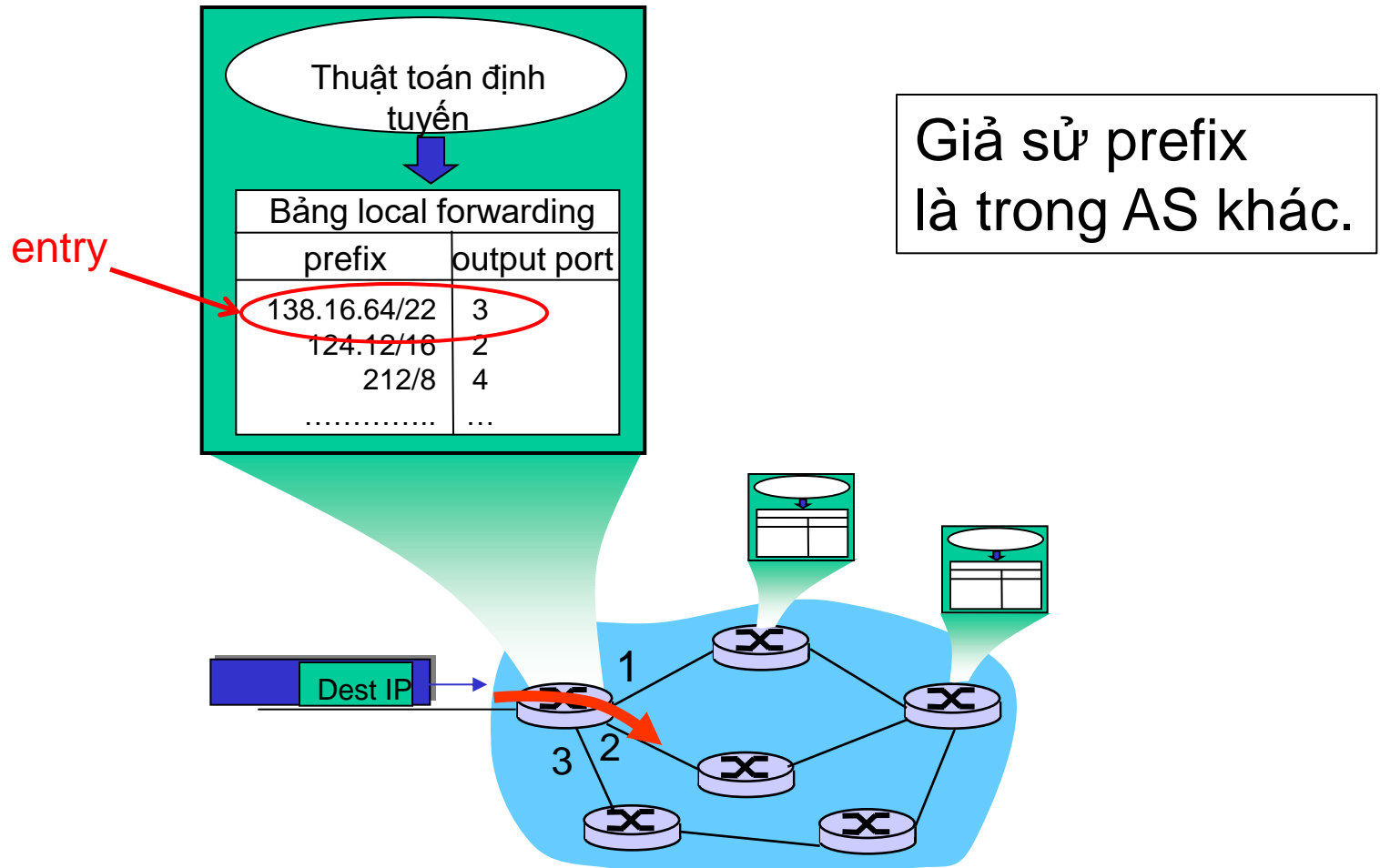
- ❖ Các thông điệp BGP được trao đổi giữa các peer trên kết nối TCP
- ❖ Các thông điệp BGP:
  - **OPEN**: mở kết nối TCP đến peer và chứng thực bên gửi
  - **UPDATE**: quảng cáo đường đi mới (hoặc báo hủy đường đi cũ)
  - **KEEPALIVE**: giữ kết nối hoạt động khi không có các UPDATES; còn đóng vai trò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

Dùng nhiều giao thức định tuyến trong cùng hệ thống:

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

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

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

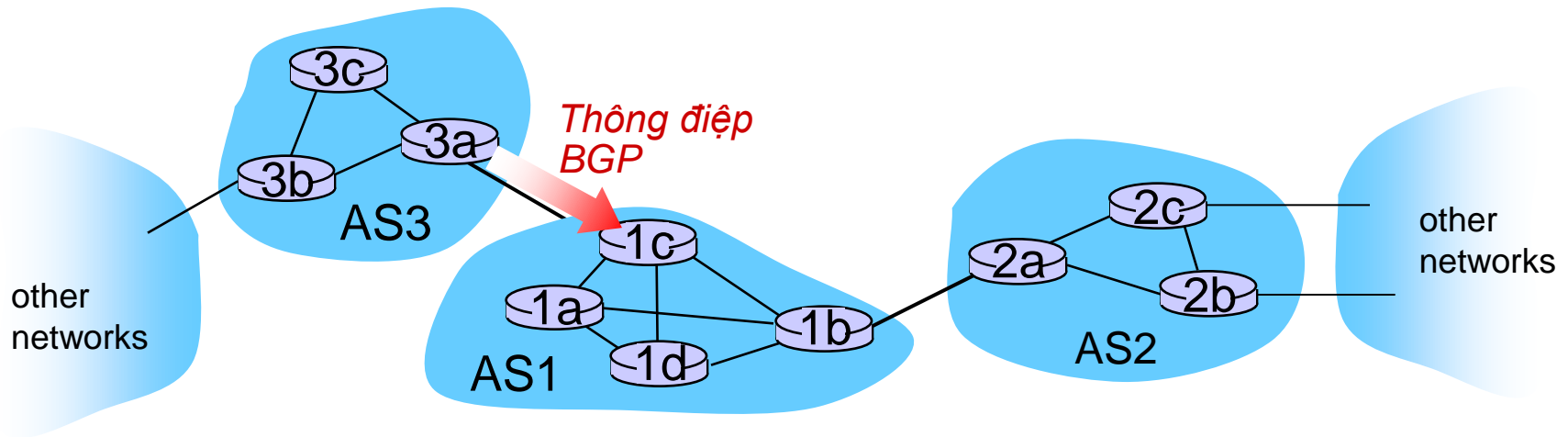


# Làm thế nào để cập nhật một mục vào bảng forwarding?

---

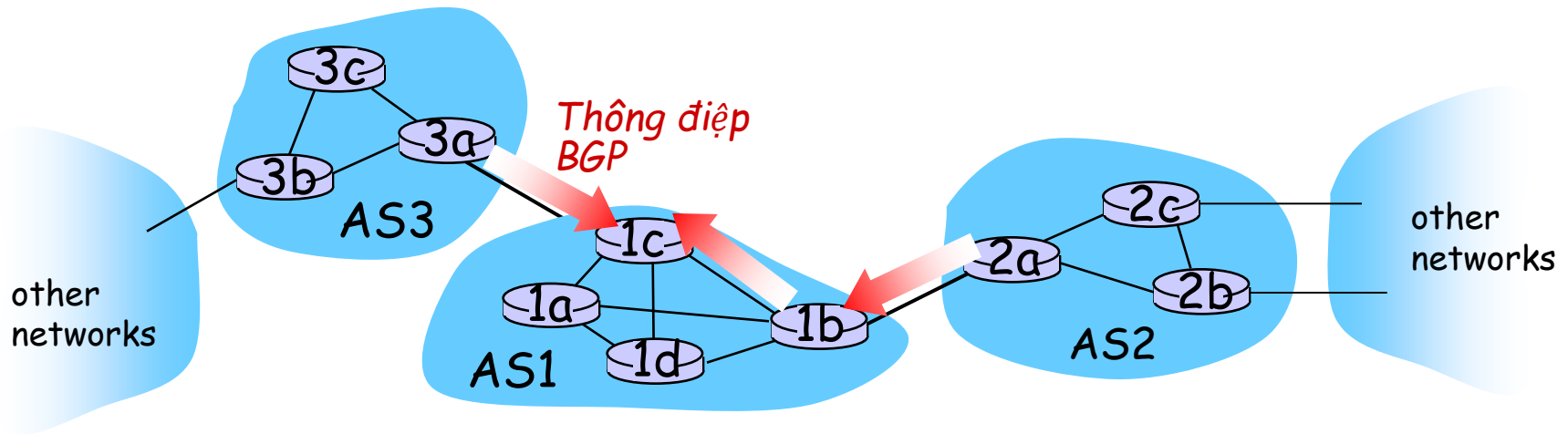
1. Router được prefix
2. Router xác định port ra (output port) cho prefix
3. Router đưa thông tin về cổng/prefix đó vào trong bảng forwarding

# Router nhận được prefix



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

# Router có thể nhận được nhiều route



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

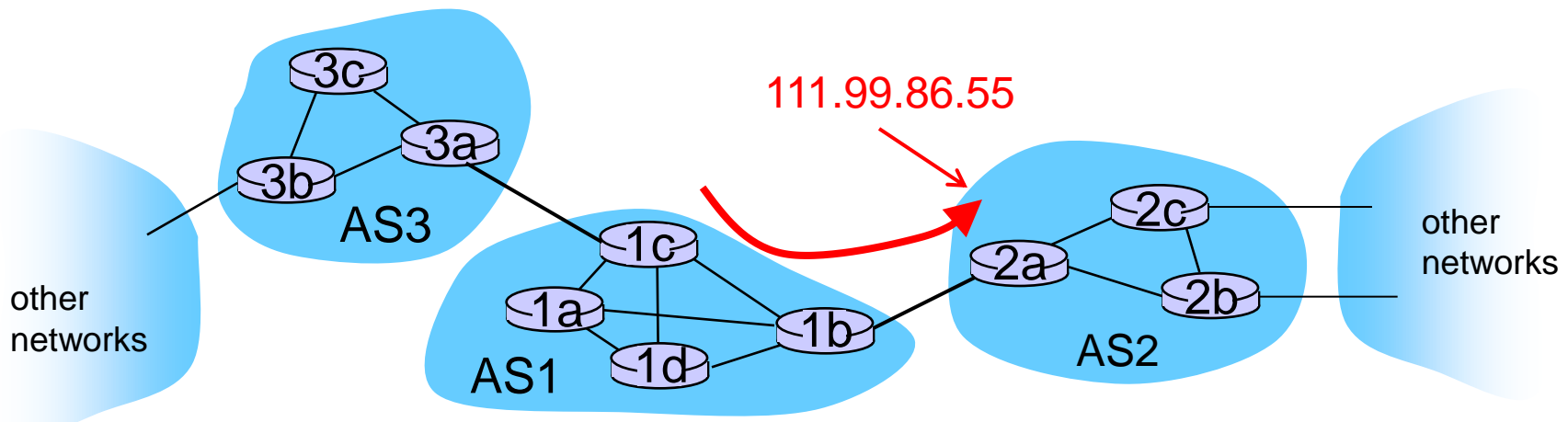
# Chọn route BGP tốt nhất tới prefix

- ❖ Router chọn route dựa trên AS-PATH ngắn nhất
- ❖ Ví dụ:
  - ❖ AS2 AS17 to 138.16.64/22 Chọn
  - ❖ AS3 AS131 AS201 to 138.16.64/22
- ❖ Điều gì sẽ xảy ra nếu có mối ràng buộc?  
Chúng ta sẽ bàn sau!



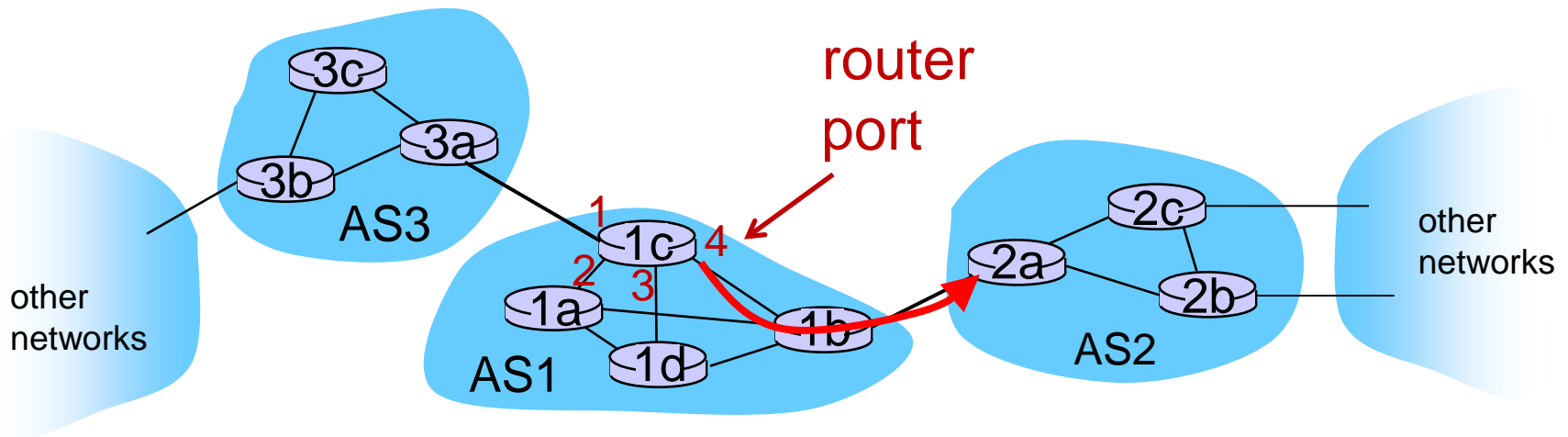
# Tìm route nội bộ tốt nhất đến route BGP

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



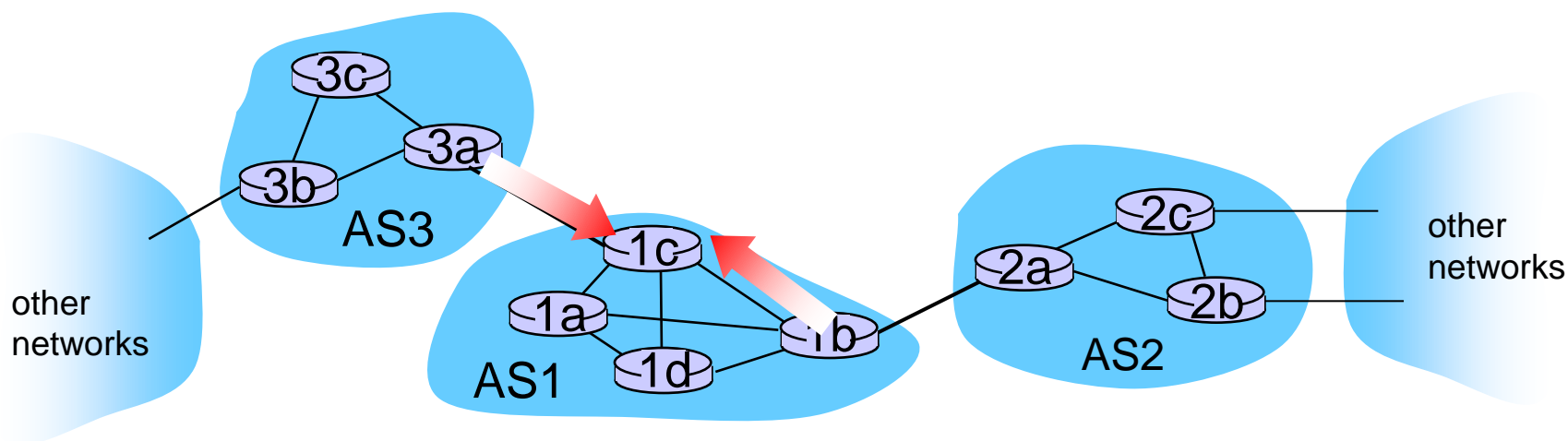
# Router xác định port cho route

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



# Định tuyến Hot Potato

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



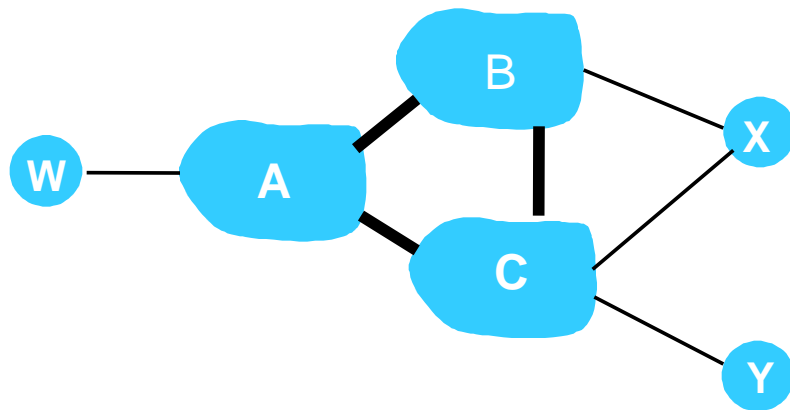
# Làm thế nào để nhập 1 thông tin định tuyến vào bảng forwarding?


---


## Tóm tắt

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

# Chính sách BGP routing

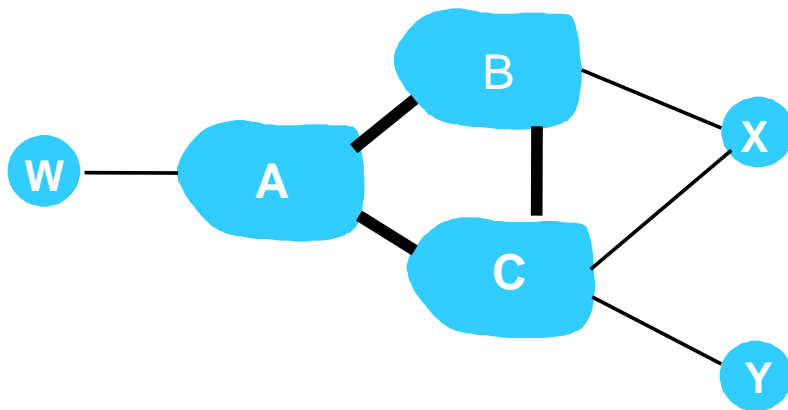


Ký hiệu:  Mạng của nhà cung cấp

 Mạng của khách hàng

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

# Chính sách BGP routing (2)



Ký hiệu:  Mạng của nhà cung cấp  
 Mạng của khách hàng

- ❖ A quảng cáo đường đi AW cho B
- ❖ B quảng cáo đường đi BAW cho X
- ❖ B có nên quảng cáo đường đi BAW cho C hay không?
  - Không nên! B không nhận được “lợi ích” cho việc định tuyến CBAW vì cả W và C không phải là khách hàng của B
  - B muốn bắt buộc C đi tới w thông A
  - B *chỉ* muốn dẫn đường đi/đến các khách hàng của nó!

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

---

## *Chính sách:*

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

## *Linh hoạt:*

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

## *Hiệu suất:*

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

# Chương 4: Nội dung

## 4.1 Giới thiệu

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

## 4.3 Cấu trúc bên trong router

## 4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing
- ICMP
- IPv6

## 4.5 Các thuật toán routing

- Link state
- Distance vector
- Hierarchical routing

## 4.6 Routing trong Internet

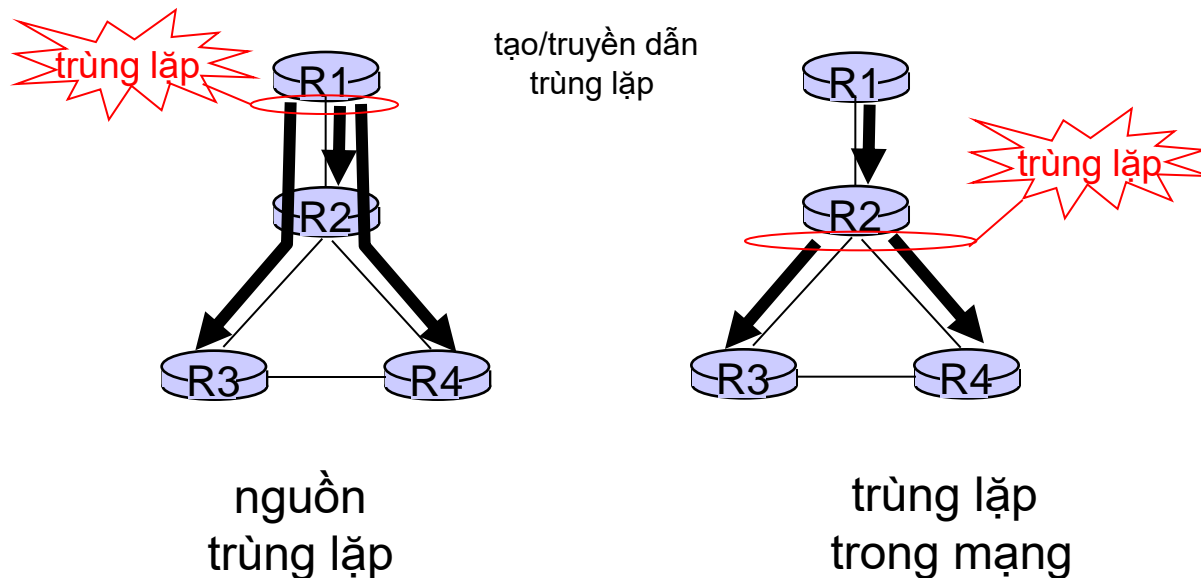
- RIP
- OSPF
- BGP

## 4.7 Broadcast và multicast routing



# Broadcast routing

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



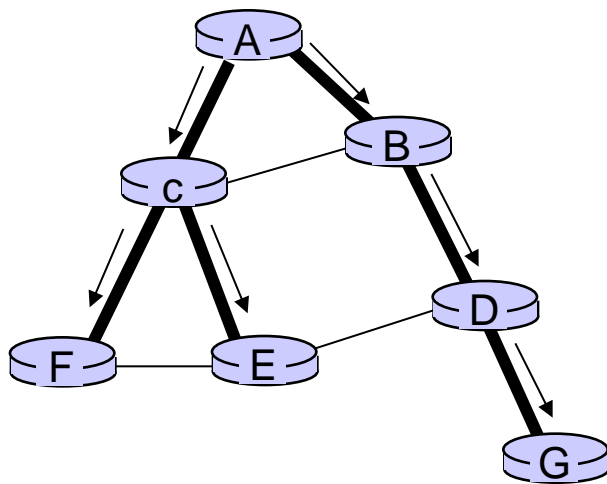
- ❖ Nguồn trùng lặp: làm sao xác định được địa chỉ người nhận?

# Trùng lặp trong mạng

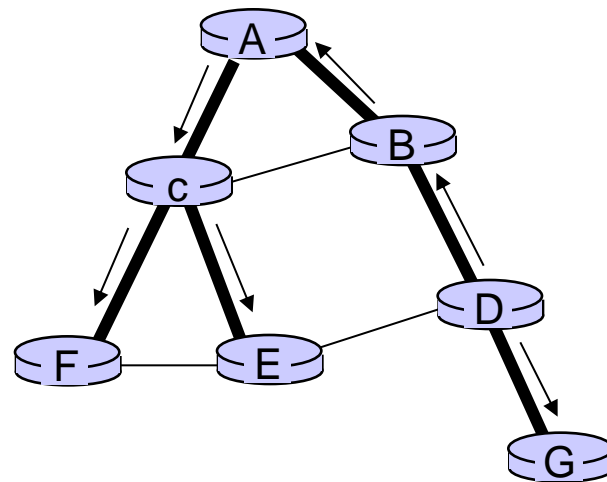
- ❖ *flooding*: khi node nhận được packet broadcast, nó gửi bản sao đến tất cả các node lân cận
  - Vấn đề: lặp lại & bão broadcast
- ❖ *flooding có điều khiển*: node chỉ broadcast packet nếu nó không gửi broadcast giống như vậy trước đó
  - node theo dõi các packet ID đã broadcast
  - Hoặc reverse path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

# Spanning tree

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



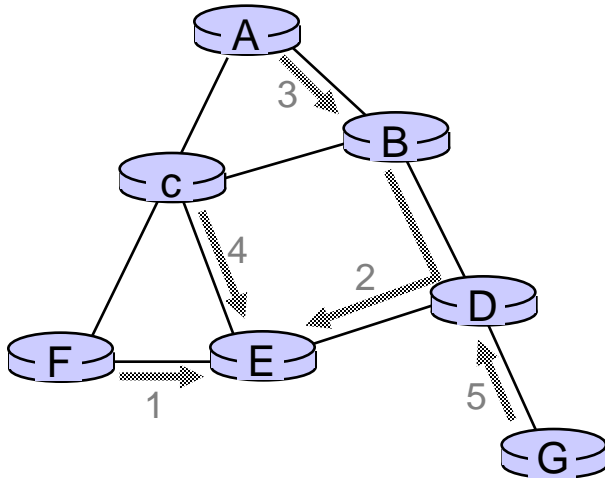
(a) broadcast được khởi tạo tại A



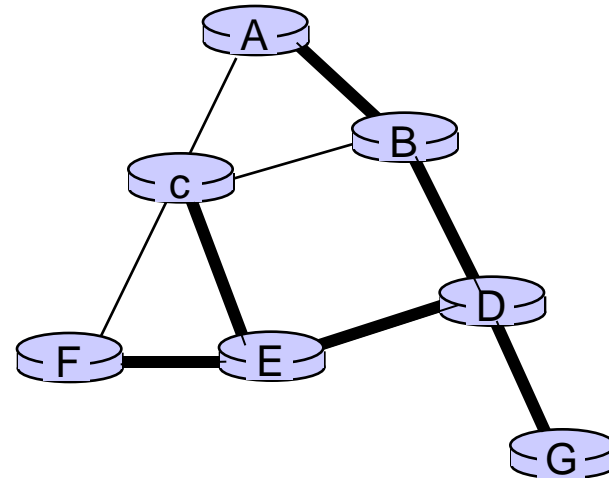
(b) broadcast được khởi tạo tại D

# Spanning tree: tạo cây

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



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

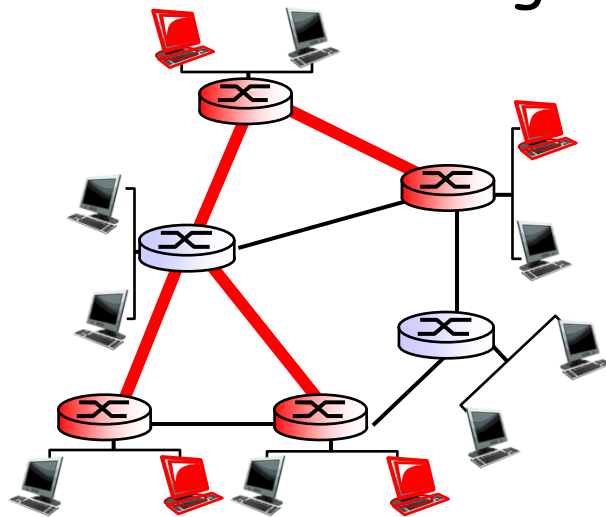


(b) spanning tree đã được xây dựng xong

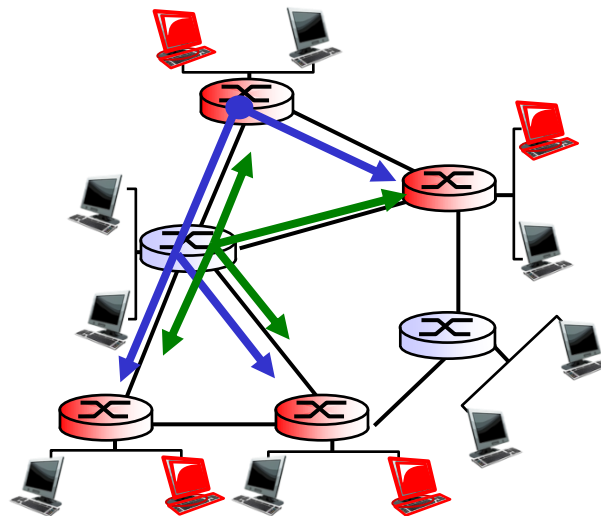
# Multicast routing: mô tả vấn đề

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

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



Cây chia sẻ



Cây dựa trên nguồn

Ký hiệu



Thành viên nhóm



Không thuộc nhóm



Router có thành viên



Router không có thành viên

# Các cách tiếp cận để xây dựng các cây multicast

---

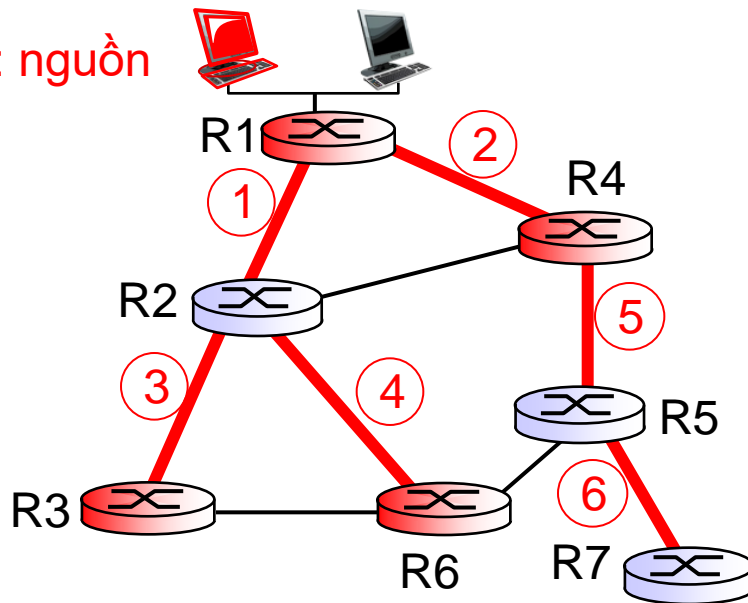
Các hướng tiếp cận:

- ❖ *Cây dựa trên nguồn (source-based tree)*: một cây cho mỗi nguồn
    - Các cây đường đi ngắn nhất
    - Chuyển tiếp theo đường đi ngược (Reverse path forwarding)
  - ❖ *Cây chia sẻ nhóm*: cả nhóm dùng chung 1 cây
    - Mở rộng tối thiểu (Steiner)
    - Các cây dựa trên trung tâm (center-based trees)
- ...tìm hiểu các cách tiếp cận cơ bản, sau đó chọn các giao thức cụ thể áp dụng cho các hướng tiếp cận này

# Cây đường đi ngắn nhất

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

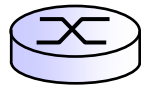
s: nguồn



Ký hiệu



router với các thành viên của nhóm đã được gắn vào



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



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

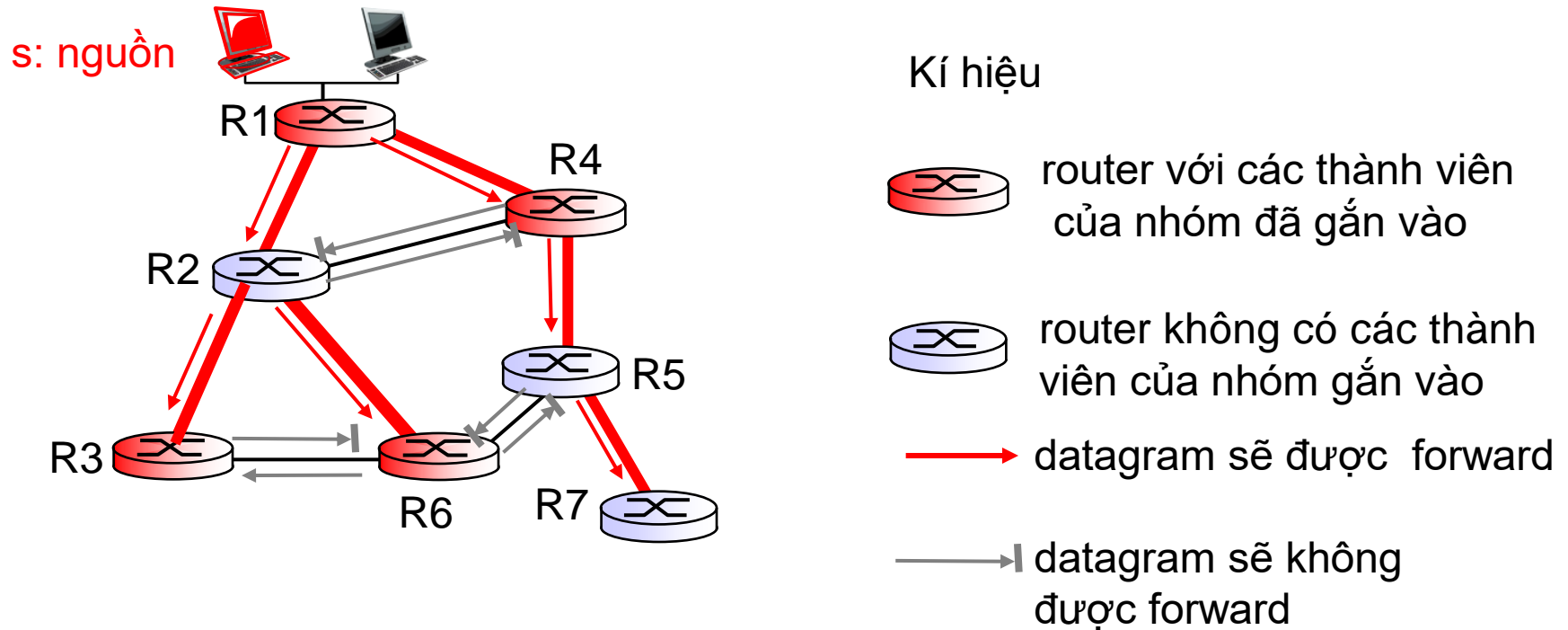
# Reverse path forwarding

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

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



# Reverse path forwarding: ví dụ

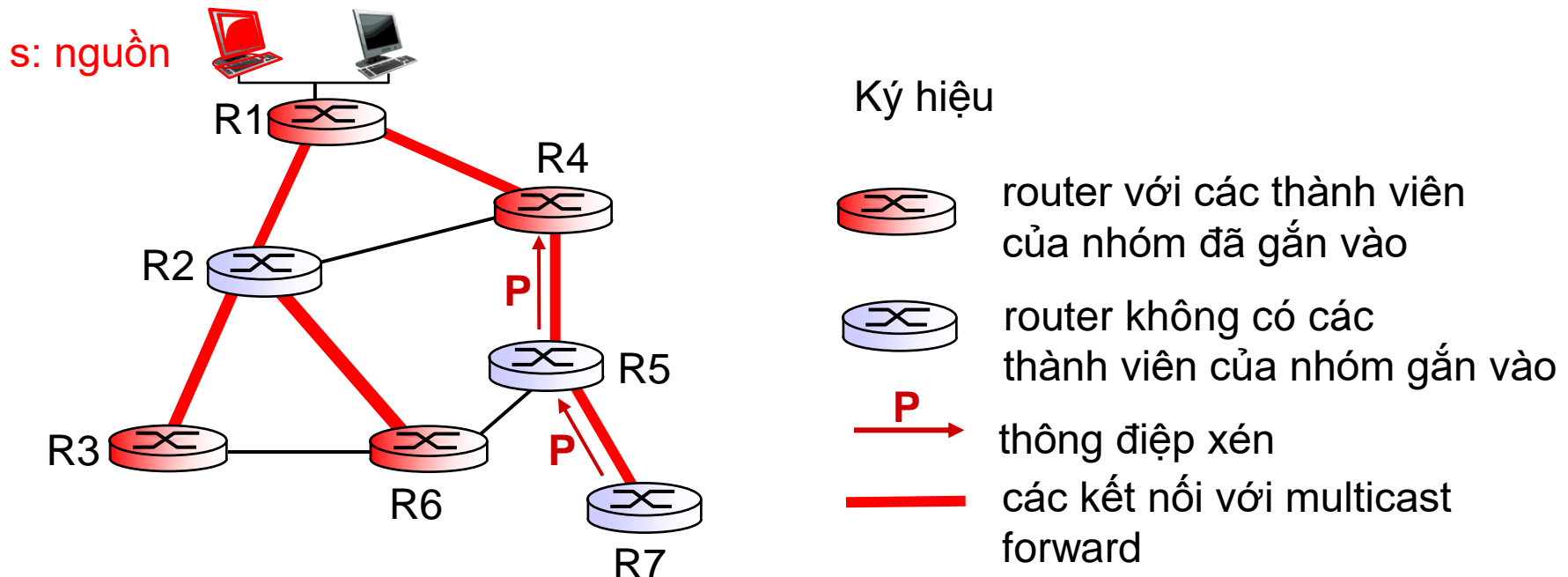


❖ Kết quả là một cây SPT đảo ngược tùy theo từng nguồn

- Có thể là một lựa chọn không tốt với các kết nối không đồng bộ

# Reverse path forwarding: xén (pruning)

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



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

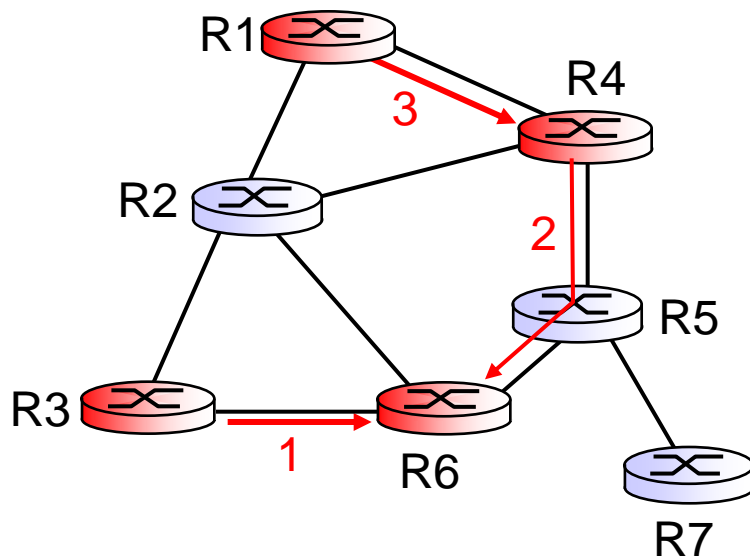
- ❖ *Cây steiner* : cây có chi phí thấp nhất kết nối đến tất cả các router có nối các thành viên nhóm
- ❖ Vấn đề là NP-complete (vấn đề phức tạp trong toán học đồ thị)
- ❖ Hiện đã có các thuật giải rất tốt
- ❖ Không sử dụng trong thực tế:
  - Độ phức tạp trong tính toán cao
  - Cần thông tin về về toàn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

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




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

# Cây dựa vào trung tâm: ví dụ

Giả sử R6 được chọn là trung tâm:



## LEGEND

-  router có các thành viên của nhóm nối vào
-  router không có các thành viên của nhóm nối vào
-  thứ tự đường đi mà các thông điệp gia nhập đã tạo ra

# Internet Multicasting Routing: DVMRP

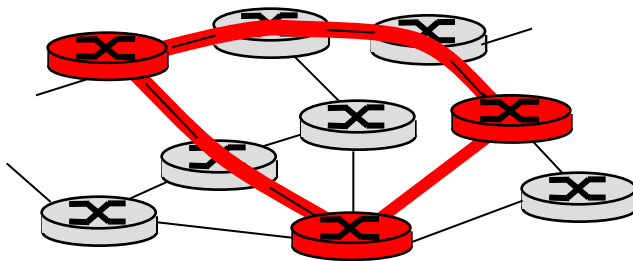
- ❖ **DVMRP**: distance vector multicast routing protocol, RFC1075
- ❖ *flood và prune*: chuyển gói theo đường đi ngược (reverse path forwarding), cây dựa vào nguồn (source-based tree)
  - Cây RPF được xây dựng dựa trên việc trao đổi các bảng định tuyến của DVMRP giữa các router hỗ trợ DVMRP
  - Không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

# DVMRP: tiếp tục...

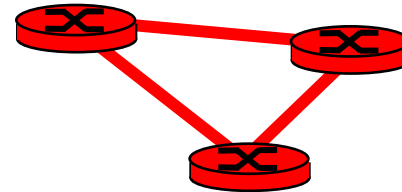
- ❖ *Trạng thái mềm*: router DVMRP theo chu kỳ (1 phút) sẽ “quên” các nhánh cây bị cắt xén:
  - Dữ liệu mcast một lần nữa đổ xuống các nhánh không được cắt xén
  - Router phía dưới: cắt xén lần nữa hoặc tiếp tục nhận dữ liệu
- ❖ Các router có thể nhanh chóng ghép lại vào cây
  - Gia nhập IGMP tại các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

# Tunneling

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



Cấu trúc vật lý



Cấu trúc logic

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



# PIM: Protocol Independent Multicast

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

## *Dày đặc:*

- ❖ Các thành viên nhóm đóng gói dày đặc, trong khoảng cách "gần".
- ❖ Bảng thông dư thừa

## *Thưa thớt:*

- ❖ Số lượng các mạng với các thành viên nhóm ít
- ❖ Các thành viên nhóm "được phân bố thưa thớt"
- ❖ Bảng thông không dư thừa

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

## *Dày đặc*

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

## *Thừa thớt:*

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

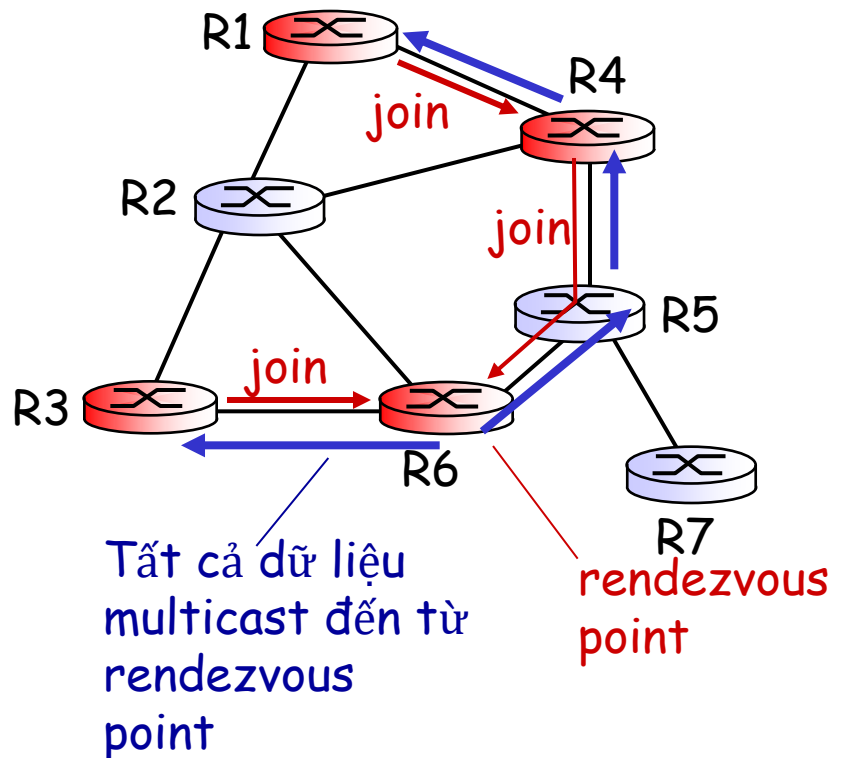
# PIM - trạng thái dày đặc

**flood-and-prune RPF**: tương tự như DVMRP nhưng...

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

# PIM – trạng thái thưa thớt

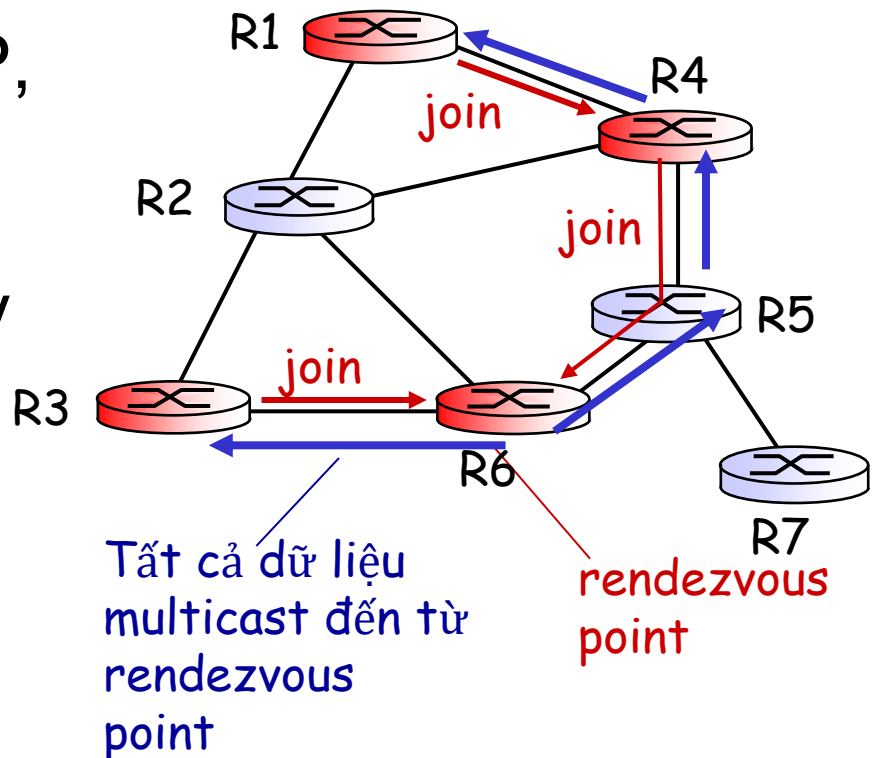
- ❖ Tiếp cận hướng trung tâm (center-based)
- ❖ Router gửi thông điệp *gia nhập* (*join* msg) đến điểm tập hợp (rendezvous point-RP)
  - Các router trung gian cập nhật trạng thái và chuyển tiếp thông điệp *gia nhập*
- ❖ Sau khi gia nhập thông qua RP, router có thể chuyển sang cây với nguồn cụ thể (source-specific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



# PIM – trạng thái thừa thớt

## *(Các) Bên gửi:*

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



# Chương 4: Hoàn thành!

## 4.1 Giới thiệu

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

## 4.3 Bên trong router

## 4.4 IP: Internet Protocol

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

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

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

## ❖ Triển khai thực tế trên Internet

## 4.5 Các thuật toán định tuyến

- Link state, distance vector, định tuyến phân cấp

## 4.6 Định tuyến trong Internet

- RIP, OSPF, BGP

## 4.7 Broadcast and multicast routing