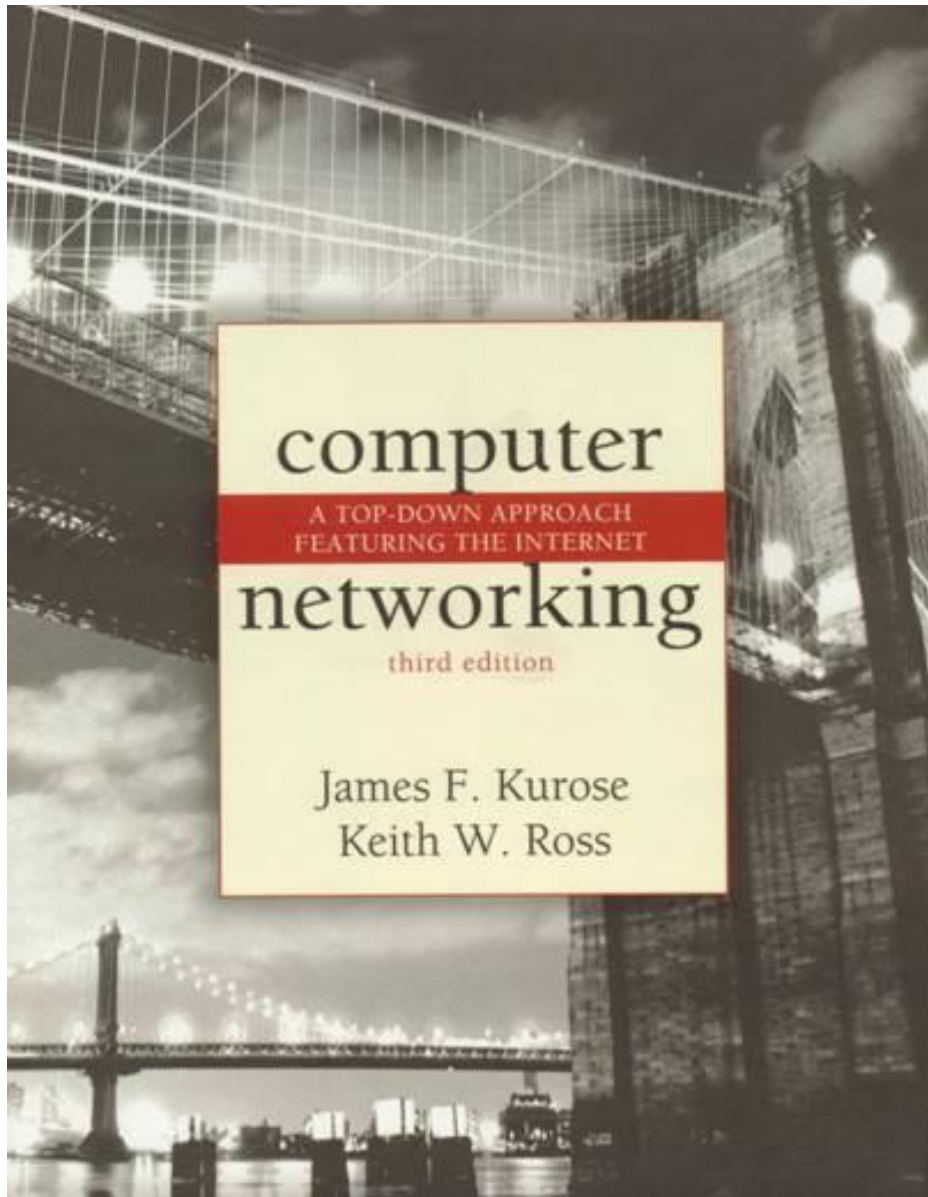


Chương 4

Lớp Network



*Computer Networking:
A Top Down Approach
Featuring the Internet,
3rd edition.*

Jim Kurose, Keith Ross
Addison-Wesley, July
2004.

Slide này được biên dịch sang tiếng Việt theo
sự cho phép của các tác giả

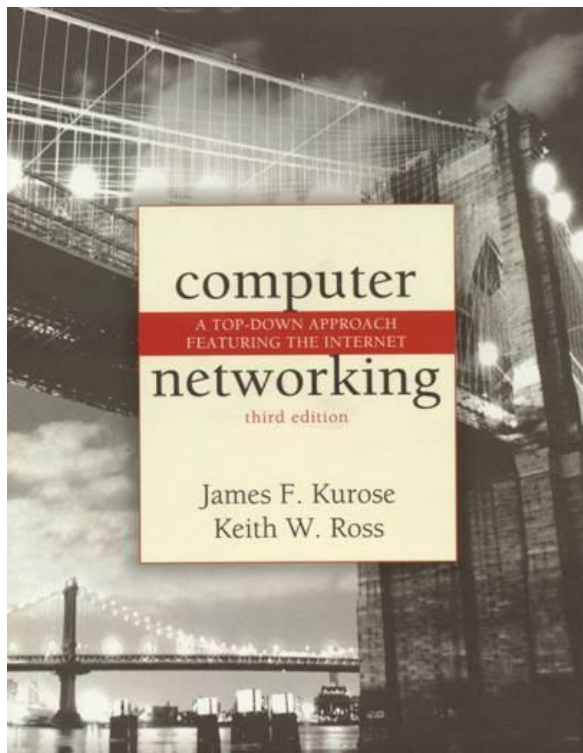
Chương 4: Lớp Network

Mục tiêu:

- ❑ hiểu các nguyên lý nền tảng của các dịch vụ lớp network:
 - các mô hình dịch vụ lớp network
 - forwarding và routing
 - một router làm việc như thế nào
 - routing (chọn đường)
 - xử lý với scale
 - các đề tài nâng cao: IPv6, mobility
- ❑ hiện thực trong Internet

Chương 4: Nội dung trình bày

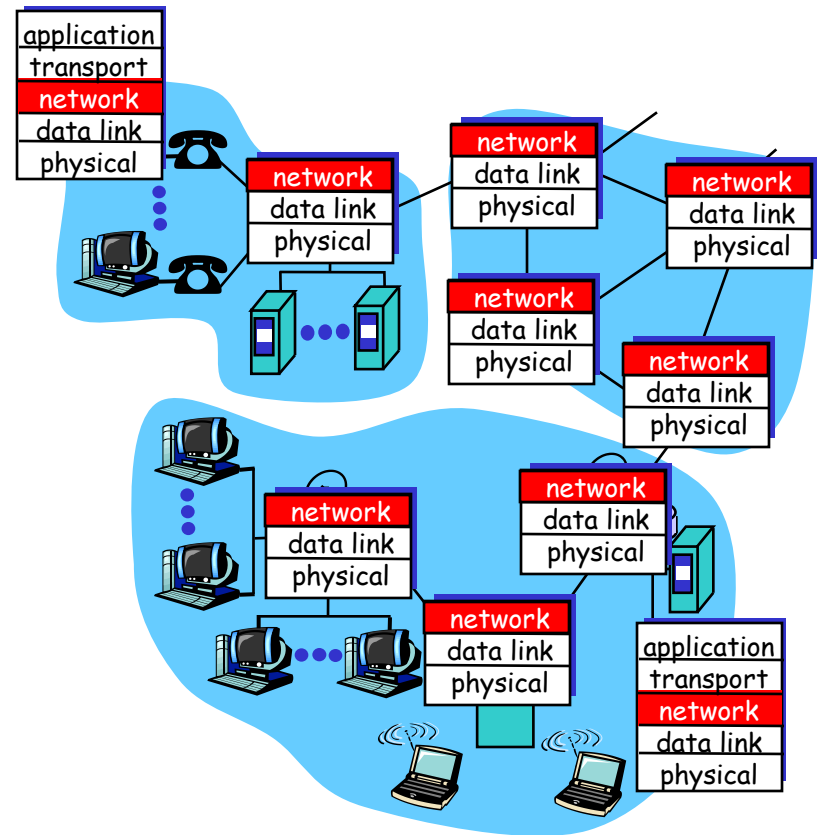
- ❑ 4.1 Giới thiệu
- ❑ 4.2 Virtual circuit và datagram networks
- ❑ 4.3 Bên trong một router?
- ❑ 4.4 IP: Internet Protocol
 - dạng thức Datagram
 - địa chỉ IPv4
 - ICMP
 - IPv6
- ❑ 4.5 các giải thuật Routing
 - Link state
 - Distance Vector
 - Hierarchical routing
- ❑ 4.6 Routing trong Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Broadcast và multicast routing



4. 1 Giới thiệu

lớp Network

- ❑ chuyển các đoạn từ host gửi đến host nhận
- ❑ bên gửi sẽ đóng gói các đoạn vào trong các datagram
- ❑ bên nhận sẽ chuyển các đoạn cho lớp transport
- ❑ các giao thức lớp network trong mọi host, router
- ❑ Router sẽ xem xét các trường header trong tất cả các IP datagram đã được chuyển cho nó



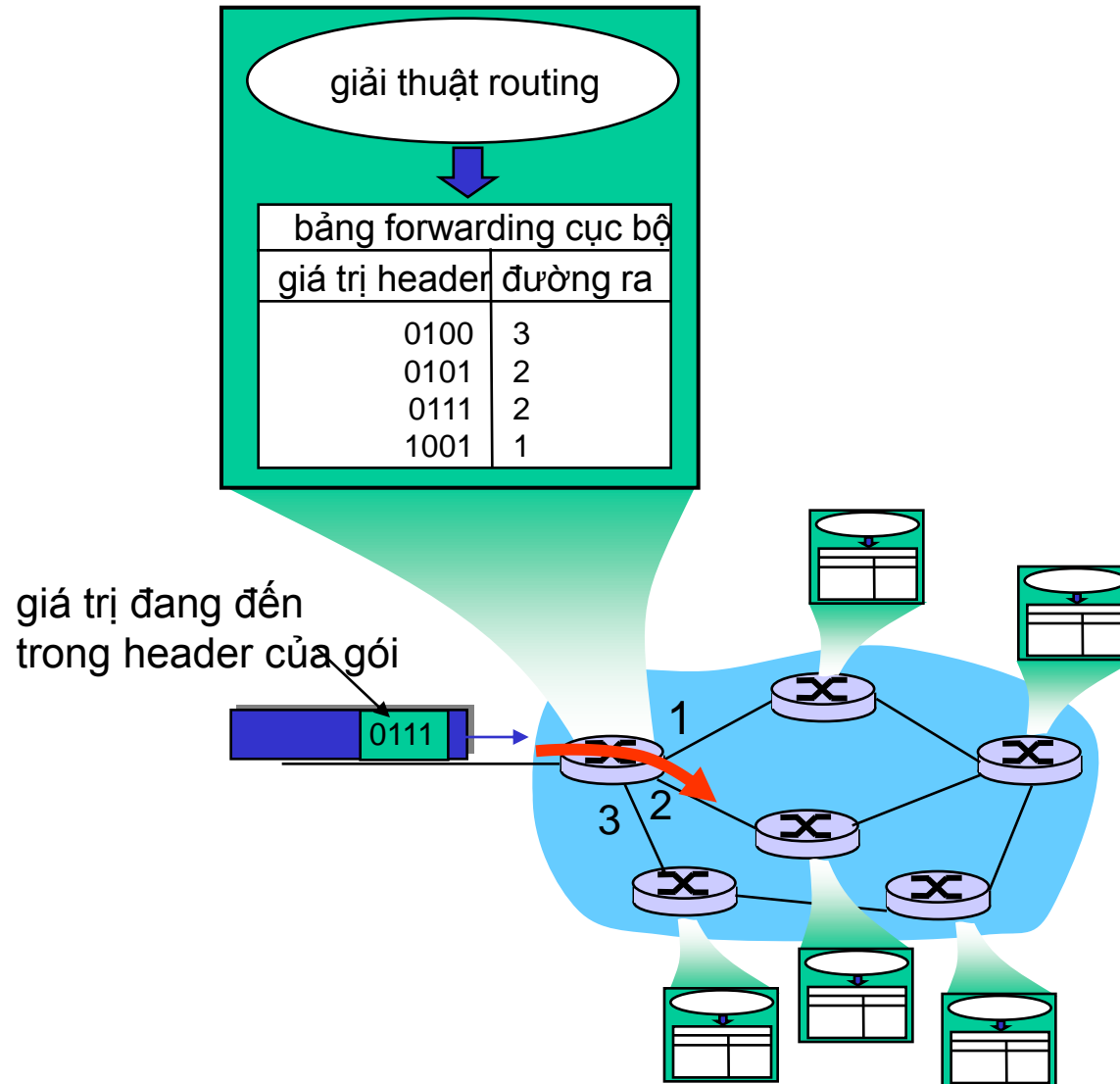
2 chức năng chính

- ❑ *forwarding*: di chuyển các gói từ đầu vào đến đầu ra thích hợp của router
- ❑ *routing*: xác định đường đi cho các gói từ nguồn đến đích
 - *các giải thuật routing*

tương tự:

- ❑ *routing*: tiến trình lập kế hoạch chuyển đi từ nguồn đến đích
- ❑ *forwarding*: tiến trình vận chuyển qua 1 giao điểm

Tác động qua lại giữa routing & forwarding



Thiết lập kết nối

- ❑ chức năng quan trọng thứ 3 của một số kiến trúc mạng:
 - ATM, frame relay, X.25
- ❑ trước khi các datagram chuyển đi, 2 host và các router trung gian thiết lập kết nối ảo
 - các router cũng liên quan
- ❑ dịch vụ kết nối lớp network với lớp transport:
 - **network**: giữa 2 host (có thể cũng chứa các router trung gian trong trường hợp kết nối ảo)
 - **transport**: giữa 2 tiến trình

mô hình dịch vụ Network

Hỏi: *Mô hình dịch vụ* là gì (cho kênh truyền các datagram từ bên gửi đến bên nhận)?

Ví dụ các dịch vụ cho các datagram riêng biệt:

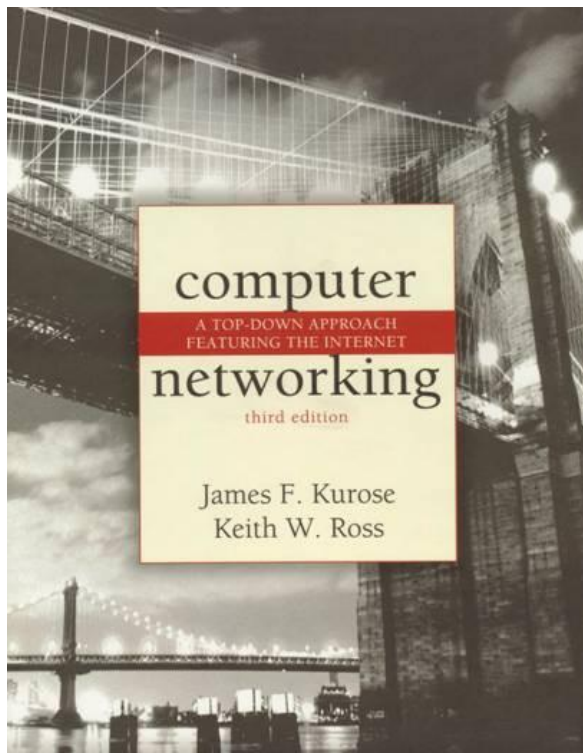
- ☐ giao nhận bảo đảm
- ☐ giao nhận bảo đảm với độ trễ < 40 ms

Ví dụ các dịch vụ cho 1 luồng các datagram:

- ☐ giao nhận datagram theo thứ tự
- ☐ bảo đảm băng thông tối thiểu cho luồng
- ☐ hạn chế các thay đổi trong khoảng trống giữa các gói

mô hình dịch vụ Network

kiến trúc Network	Mô hình dịch vụ	Bảo đảm?				phản hồi tắc nghẽn
		Băng thông	Mất mát	Thứ tự	Định thì	
Internet	best effort	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



4.2 Các mạng virtual circuit và datagram

Kết nối lớp network và dịch vụ không kết nối

- ❑ datagram network cung cấp dịch vụ không kết nối lớp network
- ❑ kết nối ảo cung cấp dịch vụ kết nối lớp network
- ❑ tương tự với các dịch vụ lớp transport, nhưng:
 - dịch vụ: host-to-host
 - không lựa chọn: network chỉ cung cấp 1 dịch vụ
 - hiện thực: bên trong phần lõi của network

các mạch ảo

“cách xử lý đường từ nguồn đến đích phải tương tự với mạch điện thoại”

- hiệu quả

- ❑ thiết lập cuộc gọi, chia nhỏ mỗi cuộc gọi *trước khi* dữ liệu có thể truyền
- ❑ mỗi gói mang nhân dạng kết nối ảo (không phải là địa chỉ đích)
- ❑ *mọi* router trên đường từ nguồn đến đích giữ nguyên “trạng thái” qua mỗi kết nối
- ❑ kết nối, các tài nguyên router (bảng thông, bộ đệm) có thể được cấp phát cho kết nối ảo (các tài nguyên dành riêng = dịch vụ có thể dự đoán trước)

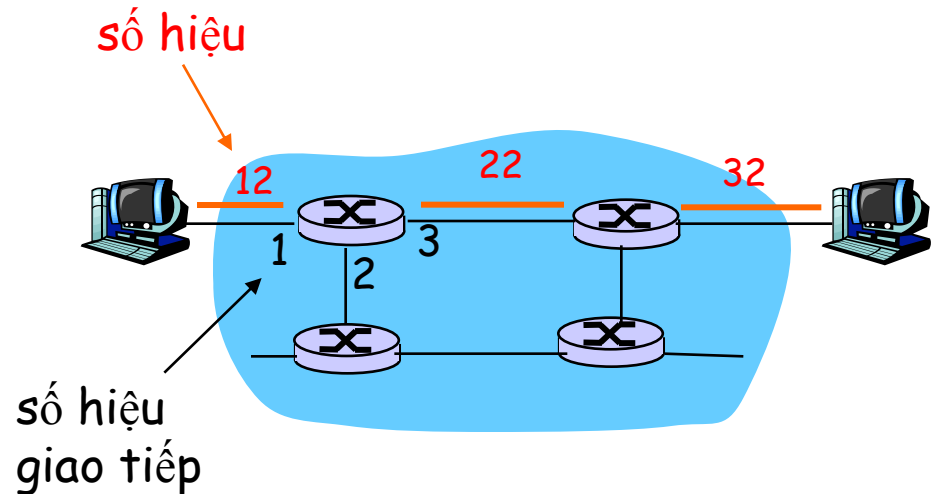
hiện thực kết nối ảo

một kết nối ảo bao gồm:

1. đường từ nguồn đến đích
 2. các số hiệu kết nối ảo, mỗi số dành cho mỗi kết nối dọc theo đường
 3. các điểm đăng ký vào các bảng forwarding trong router dọc theo đường
- gói thuộc về kết nối ảo mang số hiệu (không là địa chỉ đích)
 - số hiệu kết nối ảo có thể thay đổi trên mỗi kết nối
 - số hiệu mới được cấp từ bảng forwarding

Bảng Forwarding

bảng Forwarding trong
router góc tây-bắc:

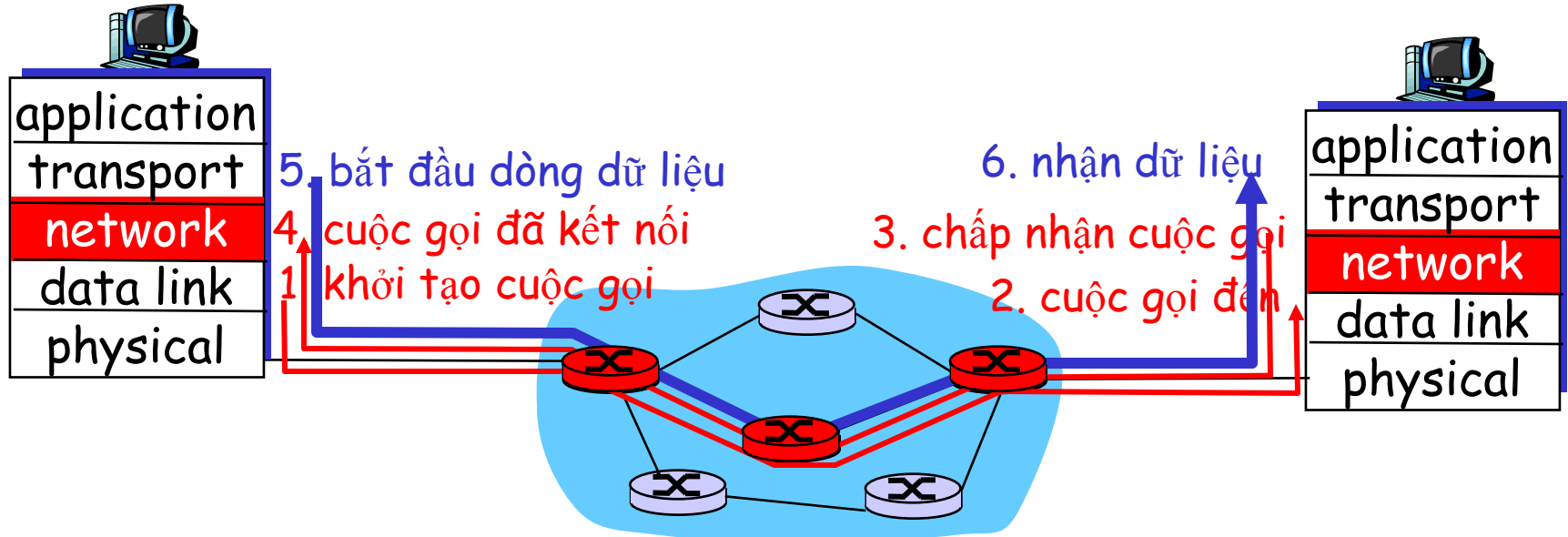


giao tiếp vào	số hiệu kết nối vào	giao tiếp ra	số hiệu kết nối ra
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Các Router giữ nguyên thông tin trạng thái kết nối!

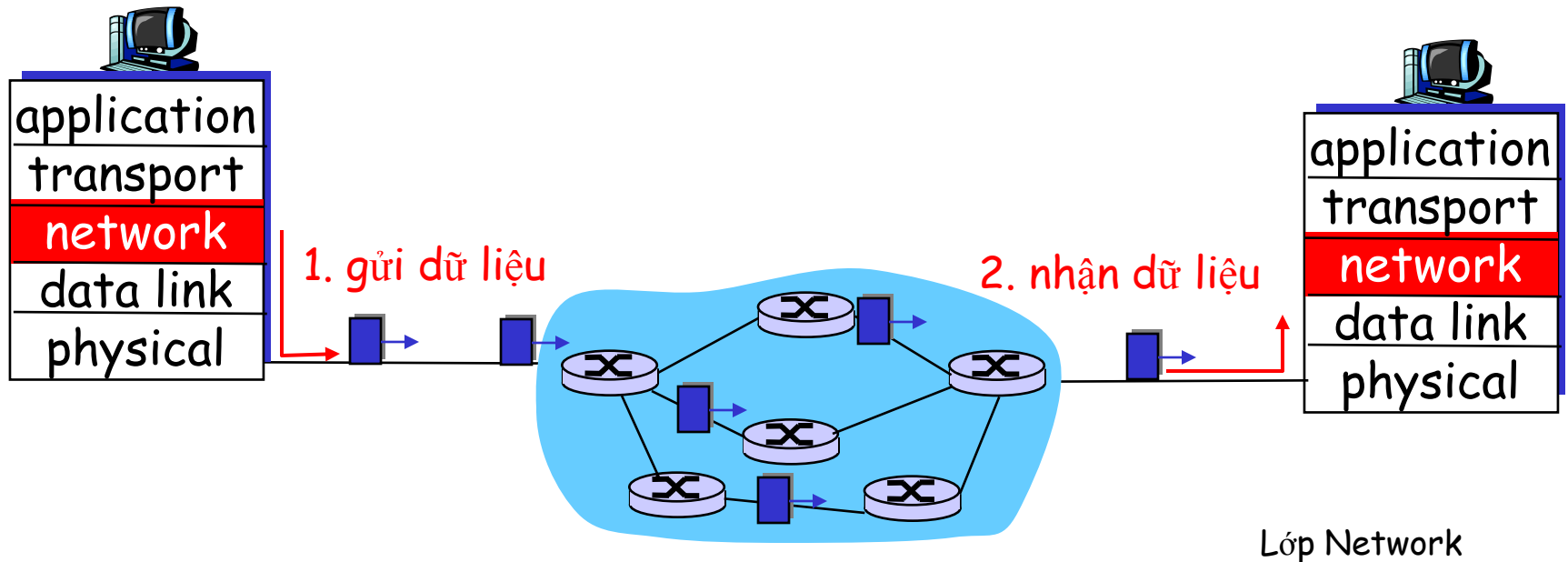
các mạch ảo: các giao thức gửi tín hiệu

- ❑ dùng để thiết lập, duy trì kết nối ảo
- ❑ dùng trong ATM, frame-relay, X.25
- ❑ không dùng trong Internet ngày nay



các mạng Datagram

- không thiết lập cuộc gọi tại lớp network
- các router: không có trạng thái về các kết nối end-to-end
 - không có khái niệm mức network của "kết nối"
- vận chuyển các gói dữ liệu địa chỉ host đích
 - các gói giữa cùng cặp nguồn-đích có thể có các đường đi khác nhau



bảng Forwarding

4 tỷ điểm
đăng nhập có thể

<u>Vùng địa chỉ đích</u>	<u>Giao tiếp kết nối</u>
11001000 00010111 00010000 00000000 đến	0
11001000 00010111 00010111 11111111	
11001000 00010111 00011000 00000000 đến	1
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 đến	2
11001000 00010111 00011111 11111111	
khác	3

So trùng prefix dài nhất

<u>So trùng prefix</u>	<u>Link Interface</u>
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
ngược lại	3

Các ví dụ:

DA: 11001000 00010111 00010110 10100001

Chọn interface nào?

DA: 11001000 00010111 00011000 10101010

Chọn interface nào?

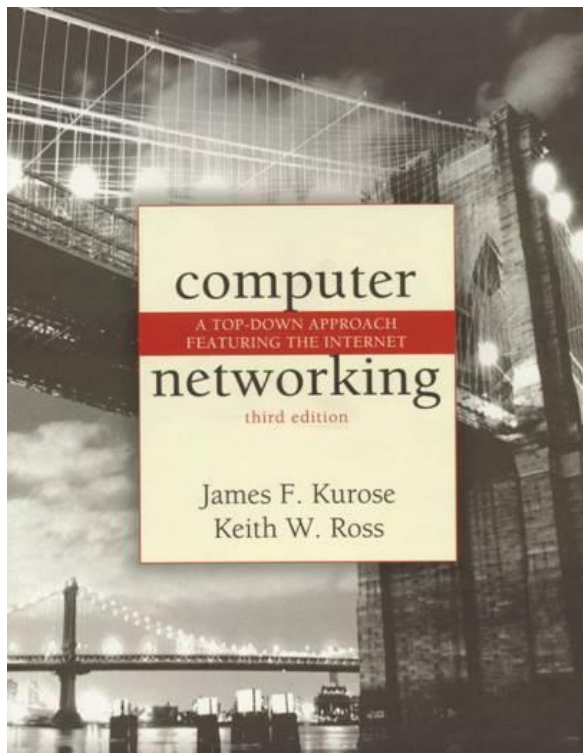
Datagram hoặc network: tại sao?

Internet (datagram)

- ❑ dữ liệu trao đổi giữa các máy tính
 - dịch vụ "mềm dẻo", không định thì chặt chẽ
- ❑ các hệ thống đầu cuối "thông minh" (các máy tính)
 - có thể thích ứng, điều khiển và sửa lỗi
 - "bên trong" mạng đơn giản, "bên ngoài" phức tạp
- ❑ nhiều kiểu kết nối
 - các đặc tính khác nhau
 - đồng nhất dịch vụ khó khăn

ATM (kết nối ảo)

- ❑ phát triển từ hệ thống điện thoại
- ❑ đàm thoại của con người:
 - định thì chặt chẽ, yêu cầu độ tin cậy
 - cần thiết cho các dịch vụ bảo đảm
- ❑ các hệ thống đầu cuối "ít thông minh"
 - điện thoại
 - "bên trong" mạng phức tạp

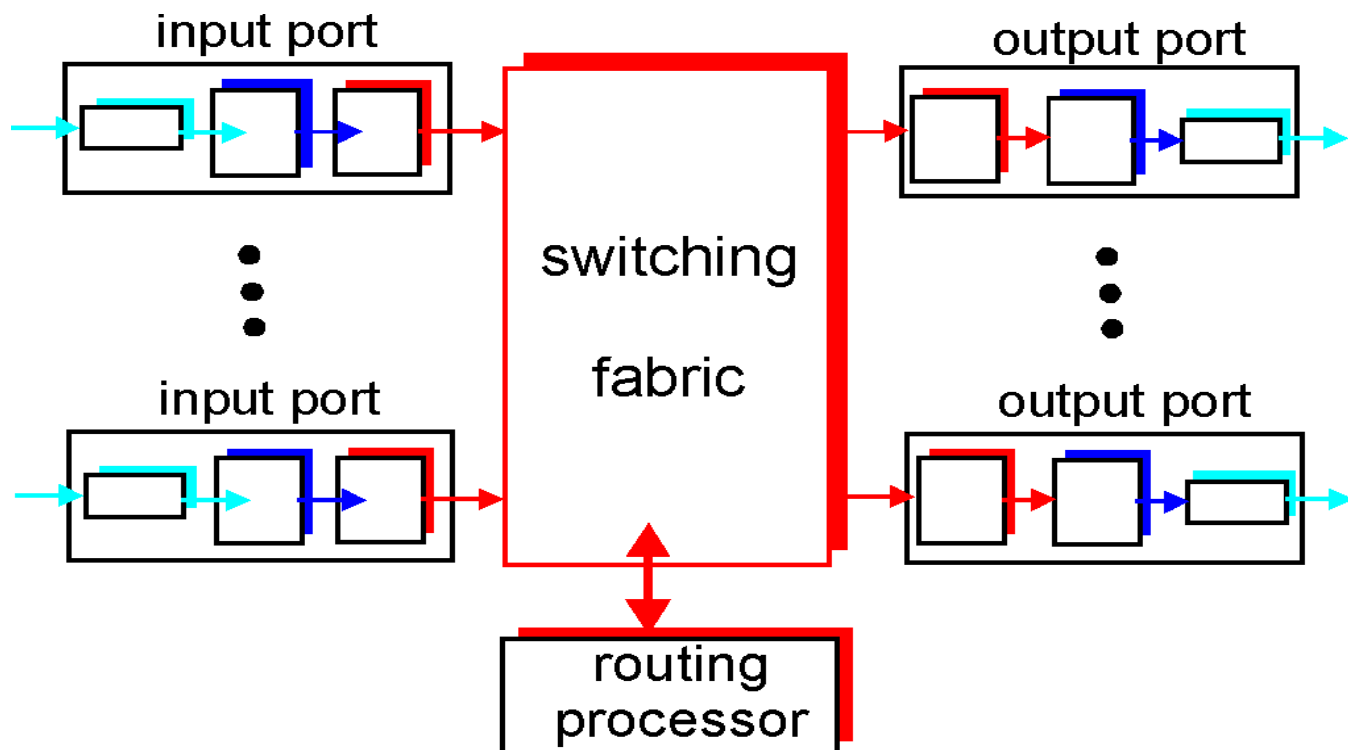


4.3 Router

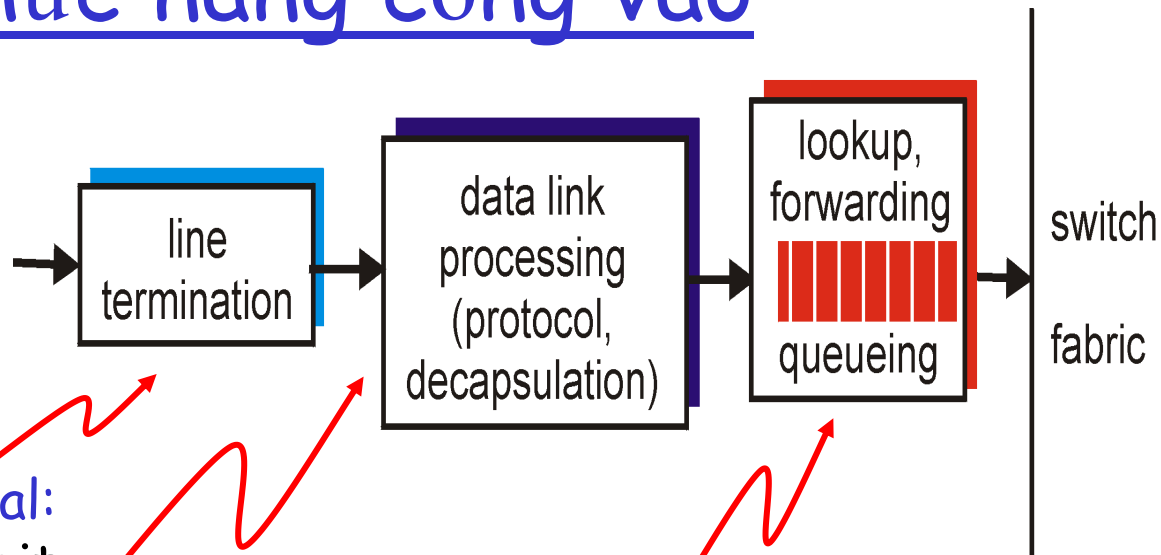
Tổng quan kiến trúc Router

2 chức năng chính:

- chạy các giao thức/giải thuật routing (RIP, OSPF, BGP)
- *đẩy* các datagram từ kết nối vào đến kết nối ra



Các chức năng cổng vào



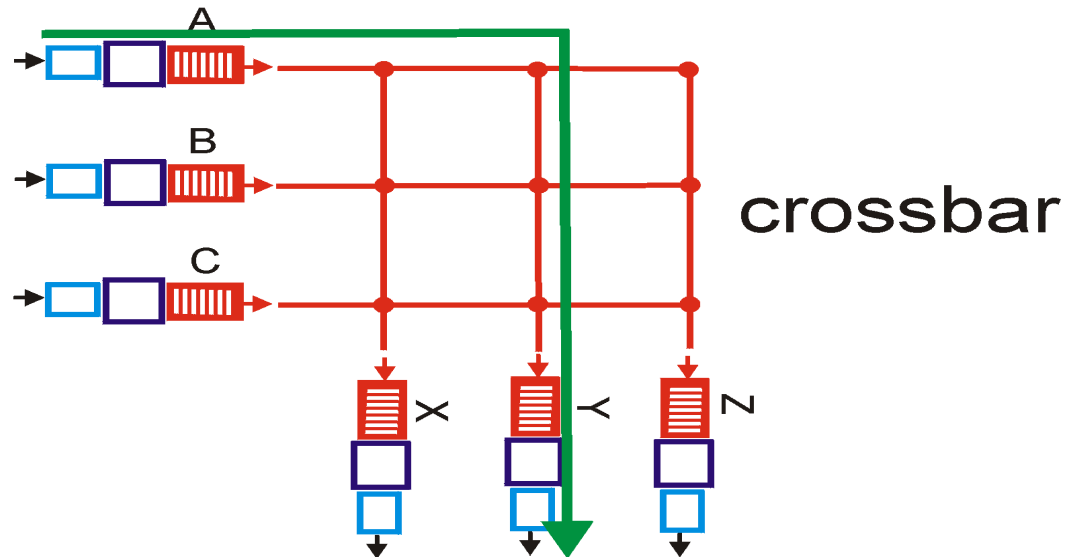
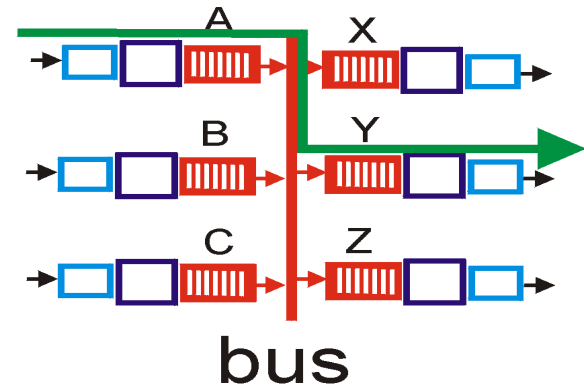
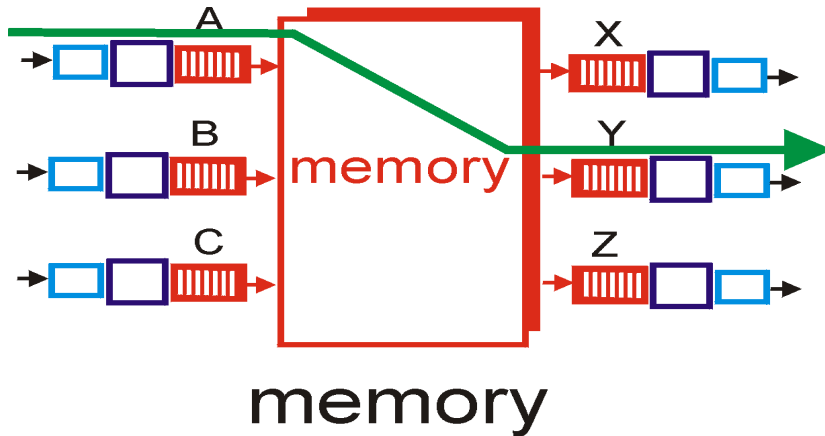
lớp Physical:
tiếp nhận mức bit

lớp Data link:
ví dụ: Ethernet
xem chương 5

switch không tập trung:

- ❑ với đích của datagram biết trước, tìm cổng ra dùng bảng forwarding trong bộ nhớ cổng vào
- ❑ mục tiêu: hoàn tất xử lý cổng vào dựa trên "tốc độ dòng"
- ❑ sắp hàng: nếu datagrams đến nhanh hơn tốc độ forwarding bên trong switch fabric

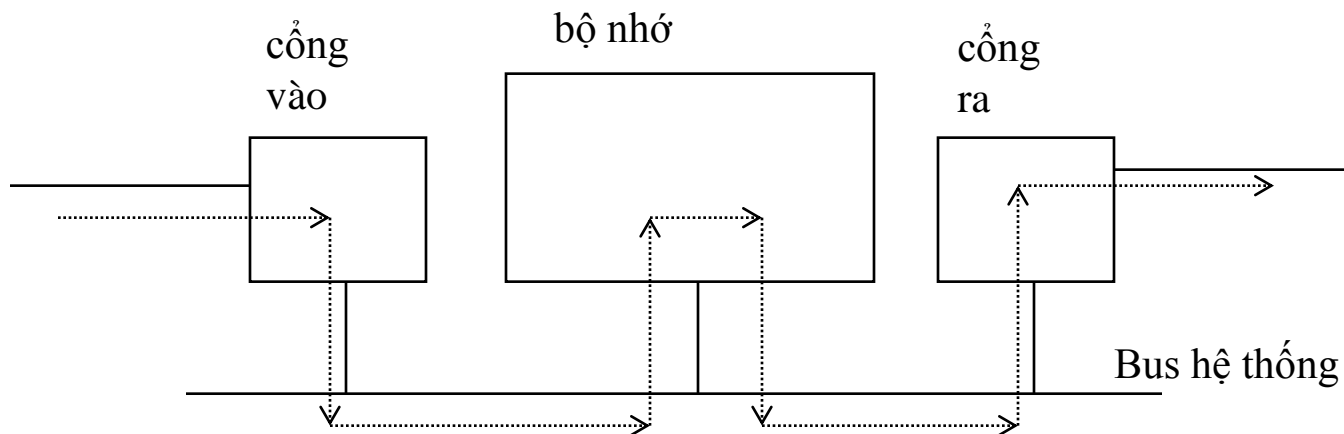
3 kiểu switching fabrics



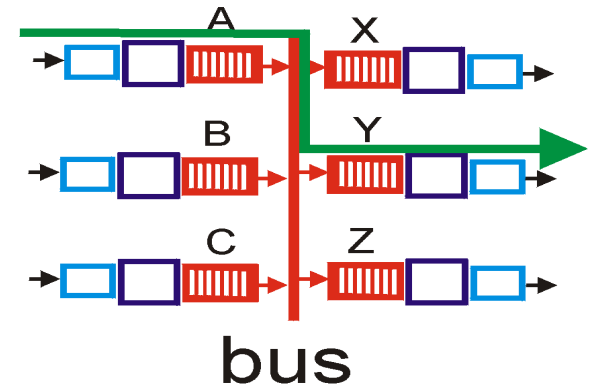
Switching thông qua bộ nhớ

Các router thế hệ thứ nhất:

- ❑ các máy tính cổ điển với switch dưới sự điều khiển trực tiếp của CPU
- ❑ gói được sao chép vào trong bộ nhớ hệ thống
- ❑ tốc độ giới hạn bởi băng thông bộ nhớ



Switch thông qua 1 Bus

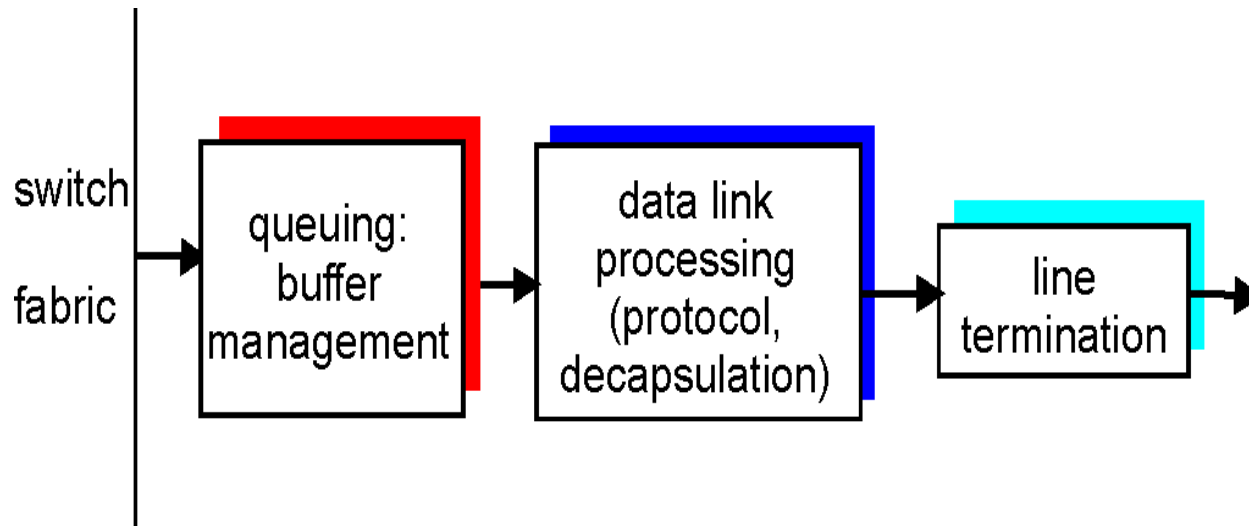


- ❑ datagram từ bộ nhớ cổng vào đến bộ nhớ cổng ra thông qua một bus chia sẻ
- ❑ **tranh chấp bus:** tốc độ switch giới hạn bởi băng thông của bus
- ❑ 1 Gbps bus, Cisco 1900: tốc độ đủ cho truy xuất các router

Switch thông qua 1 mạng kết nối nội bộ

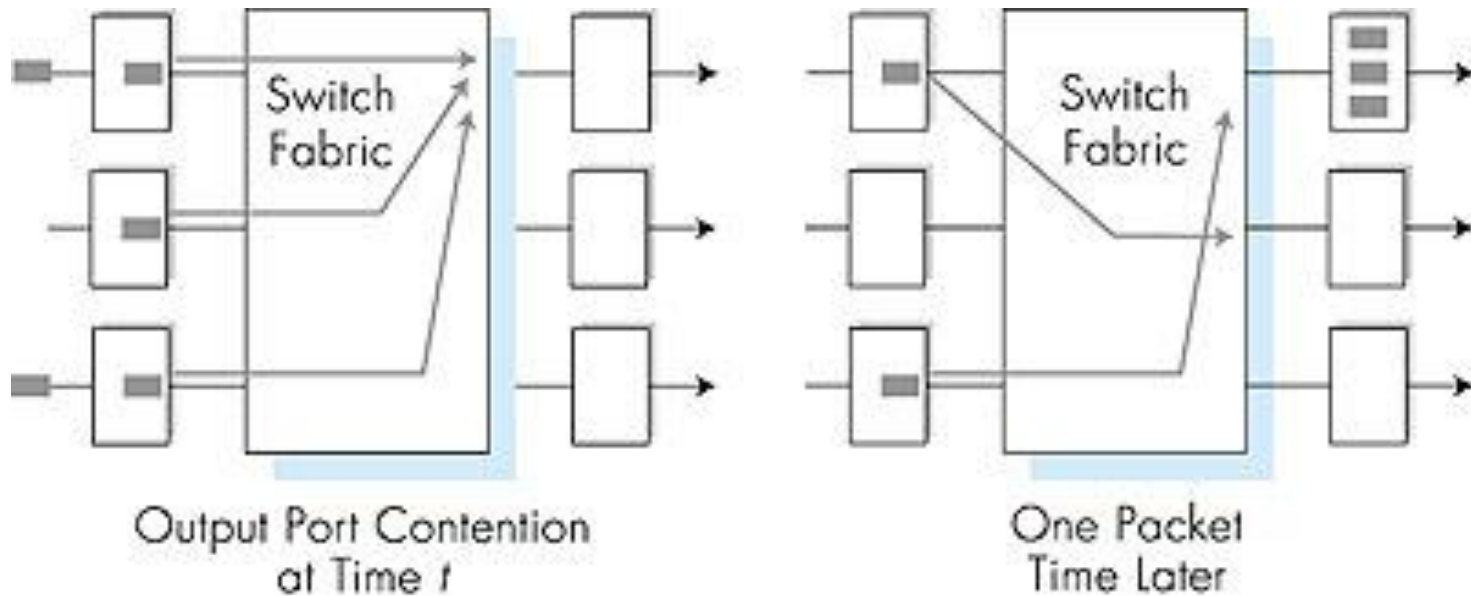
- ❑ vượt qua các giới hạn của băng thông bus
- ❑ các mạng kết nối nội bộ khác lúc đầu được dùng để kết nối các bộ xử lý trong thiết bị có nhiều bộ xử lý
- ❑ thiết kế nâng cao: phân mảnh datagram vào các ô độ dài cố định, chuyển các ô thông qua fabric.
- ❑ Cisco 12000: chuyển với tốc độ hàng Gbps thông qua kết nối nội bộ

Các công ra



- *Đệm* được yêu cầu khi các datagram đến từ fabric nhanh hơn tốc độ truyền
- *Scheduling discipline* chọn giữa những datagram đã sắp hàng để truyền

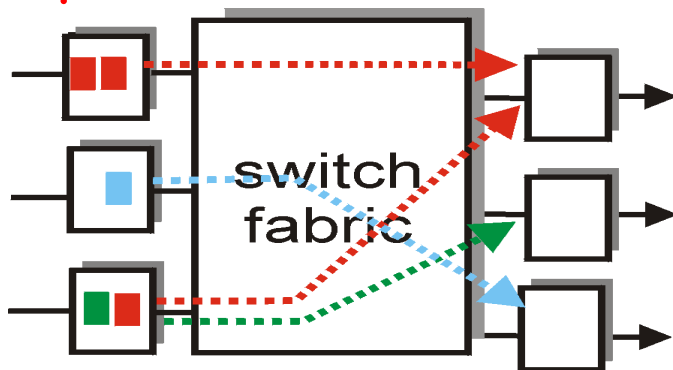
Sắp hàng tại cổng ra



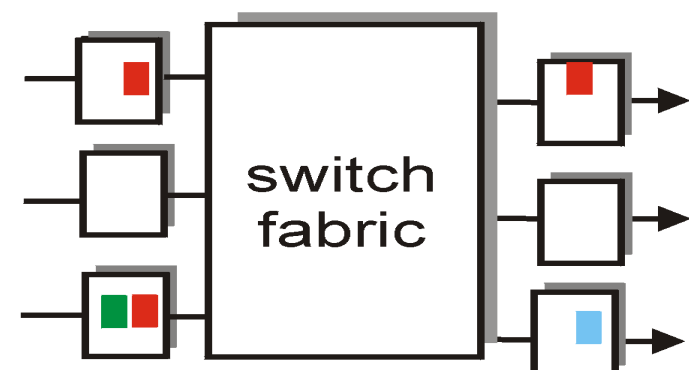
- ❑ đệm khi tốc độ đến thông qua switch vượt quá tốc độ dòng ra
- ❑ *sắp hàng (trễ) và mất mát bởi vì bộ đệm tại cổng ra bị tràn!*

Sắp hàng tại cổng vào

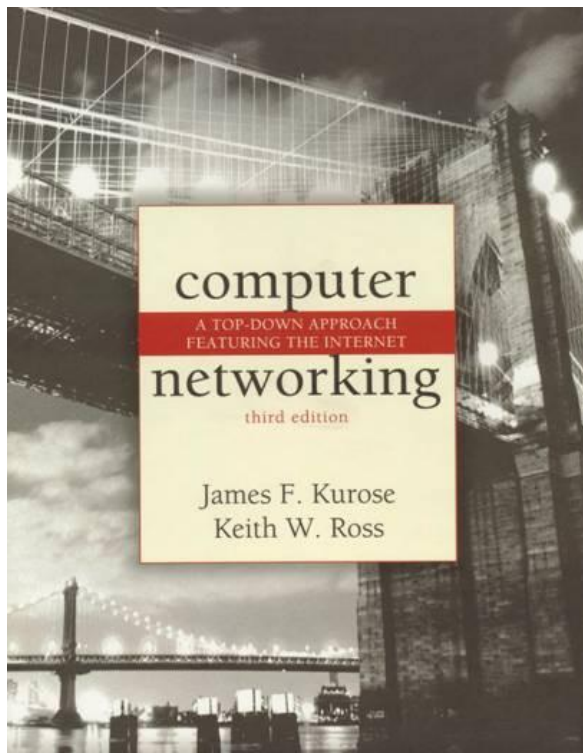
- ❑ Fabric chậm hơn sự phối hợp tại các cổng vào -> sắp hàng xảy ra tại các hàng vào
- ❑ **Tắc nghẽn Head-of-the-Line (HOL):** datagram đã sắp hàng phía trước của hàng ngăn cản các datagram khác di chuyển lên trước
- ❑ *sắp hàng (trễ) và mất mát bởi vì bộ đệm tại cổng vào bị tràn!*



output port contention
at time t - only one red
packet can be transferred



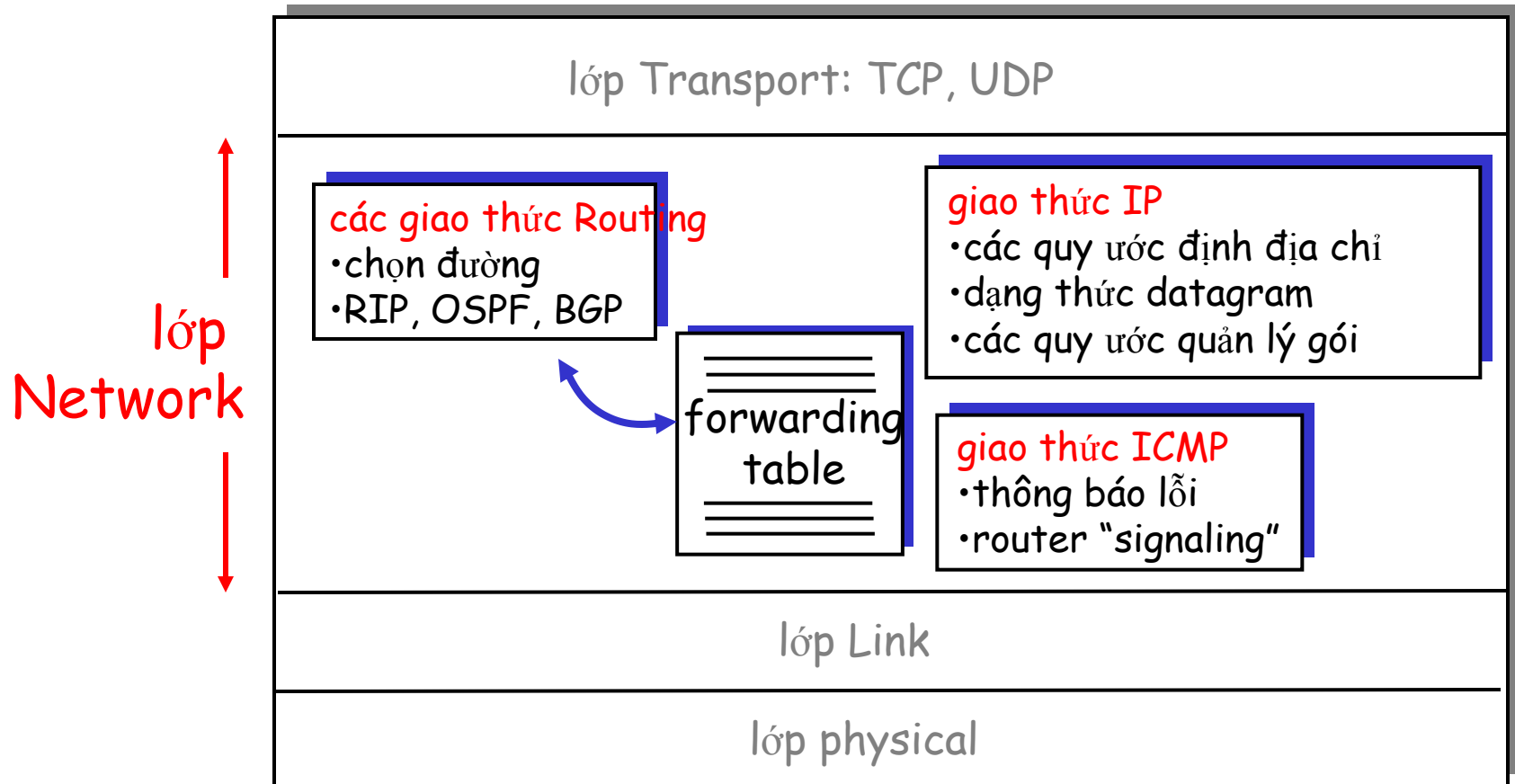
green packet
experiences HOL blocking



4.4 IP - Internet Protocol

Lớp Internet Network

Các chức năng:



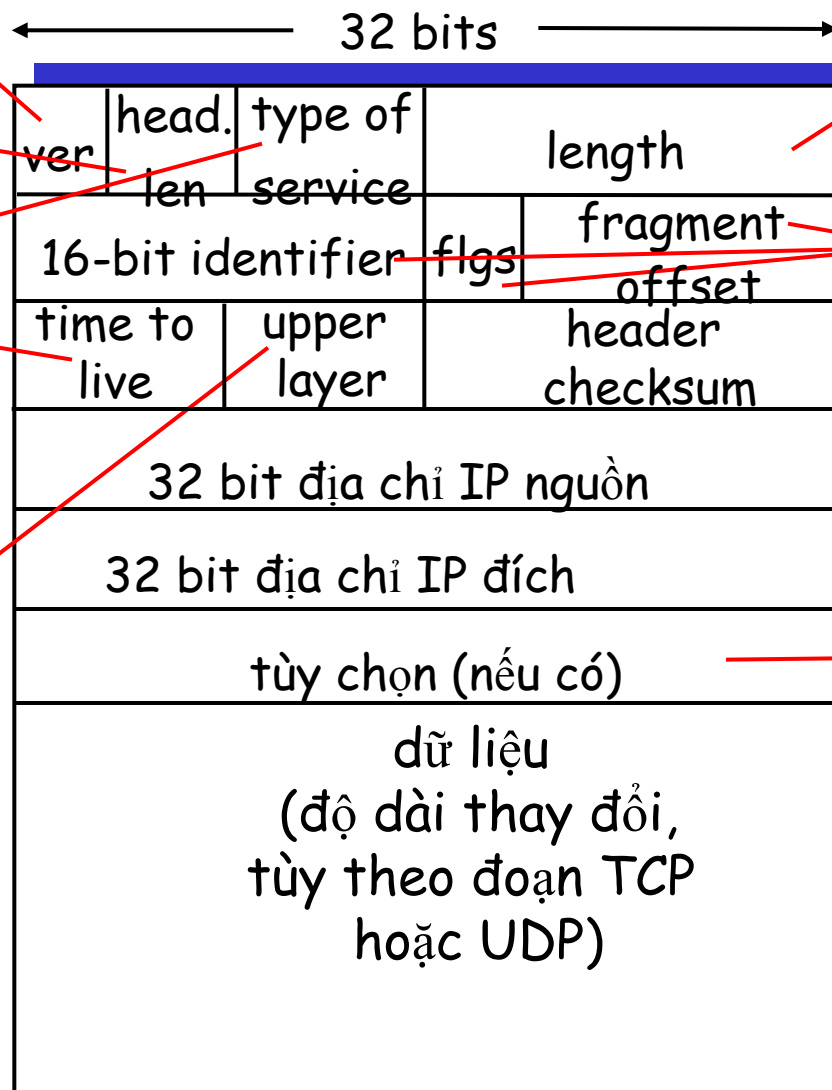
dạng thức IP datagram

số hiệu phiên bản
giao thức IP
độ dài header
(bytes)

"kiểu" của dữ liệu

số hop còn lại
tối đa
(giảm xuống tại
mỗi router)

giao thức lớp trên



tổng độ dài
datagram (bytes)

dành cho việc
phân mảnh/
tổng hợp

ví dụ: trường
timestamp
ghi nhận đường đi,
danh sách các
router
để đi đến

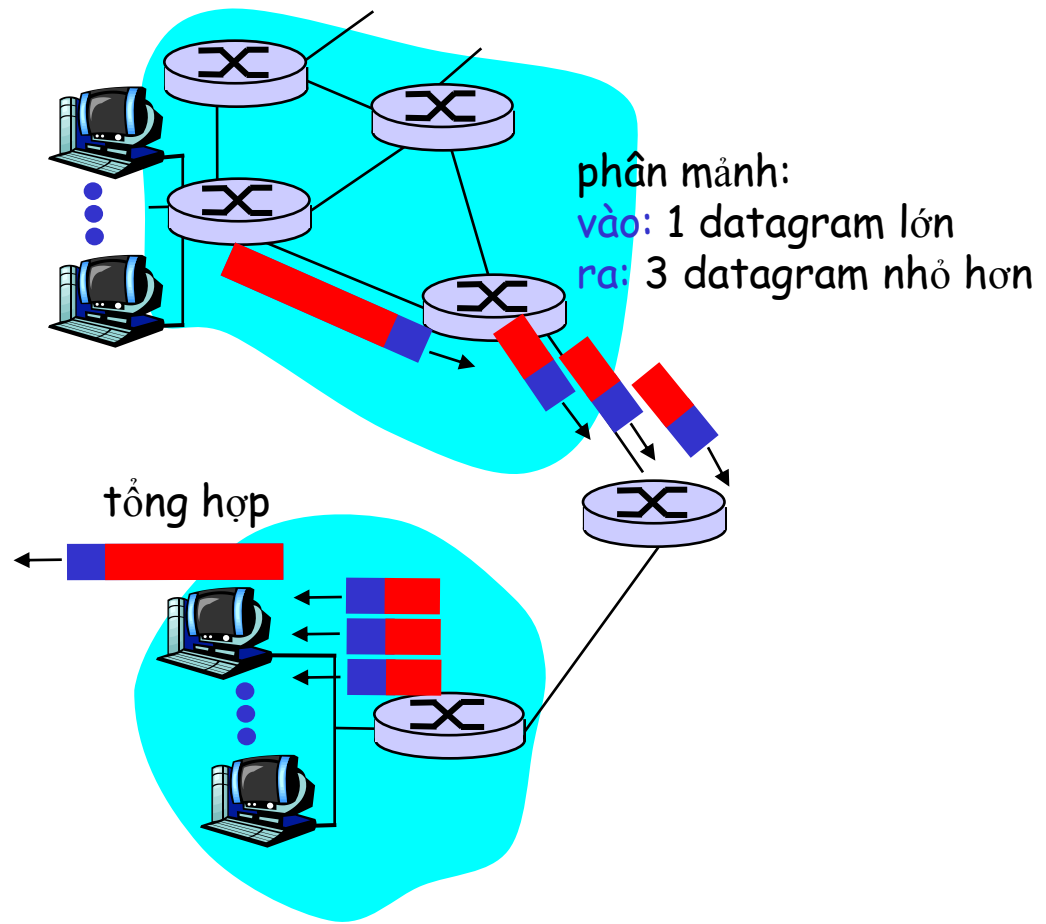
bao nhiêu overhead
với TCP?

- ❑ 20 bytes của TCP
- ❑ 20 bytes của IP
- ❑ = 40 bytes +
overhead lớp app

Lớp Network

Phân mảnh & tổng hợp IP

- ❑ các kết nối mạng có MTU (max.transfer size) - frame mức kết nối lớn nhất có thể.
 - các kiểu liên kết khác nhau, các MTU khác nhau
- ❑ các datagram lớn được chia (phân mảnh) bên trong mạng
 - 1 datagram thành một vài datagram
 - "tổng hợp" tại đích cuối cùng
 - các bit của IP header xác định, thứ tự liên quan các mảnh



Phân mảnh & tổng hợp IP

Ví dụ

- ❑ 4000 byte datagram
- ❑ MTU = 1500 bytes

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

1 datagram lớn thành một vài datagram nhỏ hơn

1480 bytes trong trường dữ liệu

offset =
 $1480/8$

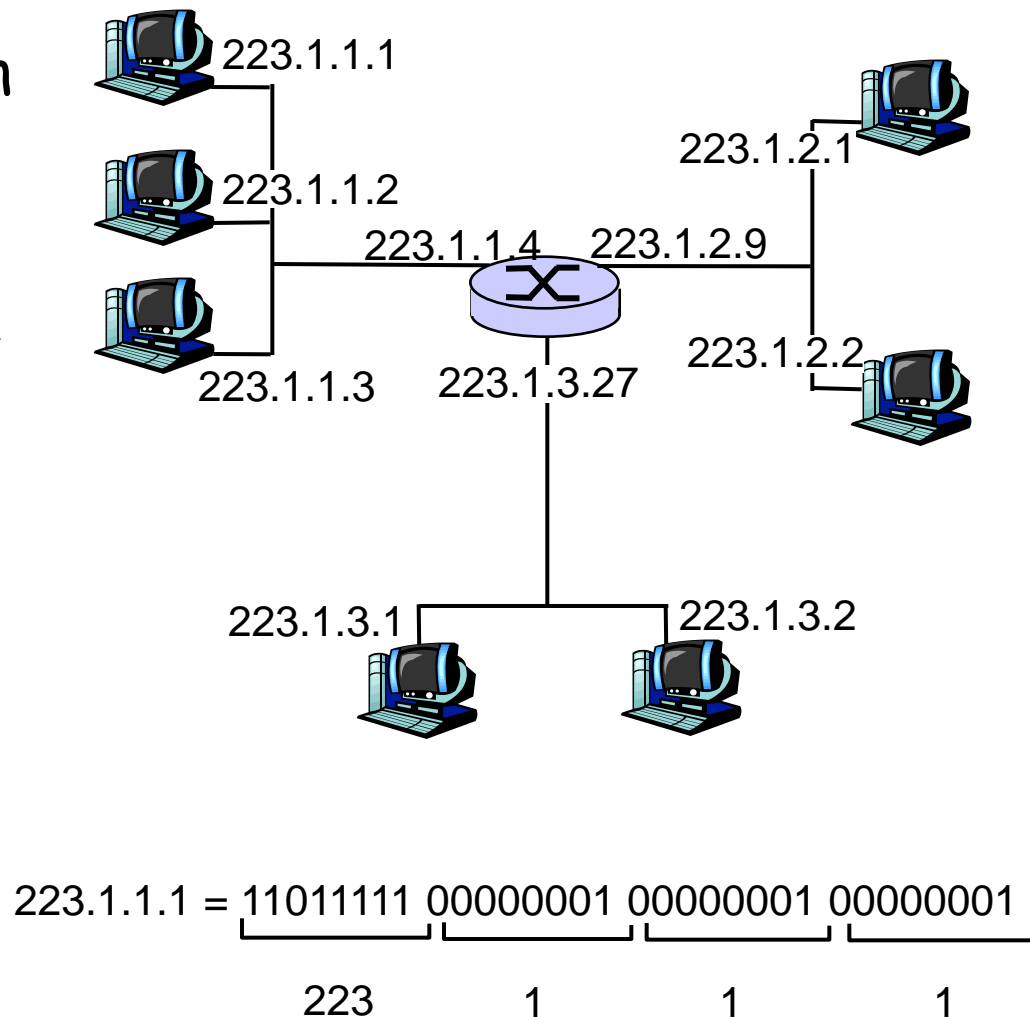
	length	ID	fragflag	offset	
	=1500	=x	=1	=0	

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

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

Định địa chỉ IP: giới thiệu

- ❑ địa chỉ IP: 32-bit nhận dạng cho host, router *interface*
- ❑ *interface*: kết nối giữa host/router và kết nối vật lý
 - router thường có nhiều interface
 - host thường có 1 interface
 - mỗi địa chỉ IP liên kết với mỗi interface



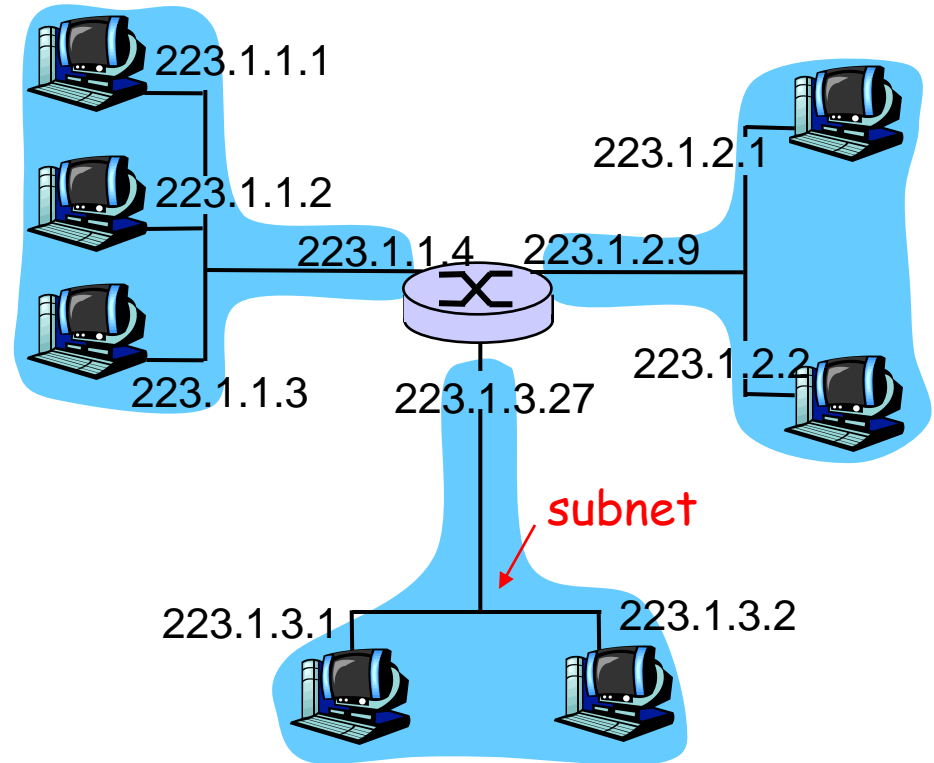
Các Subnet (mạng con)

□ địa chỉ IP:

- phần subnet (các bit có trọng số cao)
- phần host (các bit có trọng số thấp)

□ subnet là gì?

- các interface thiết bị có phần subnet của địa chỉ IP giống nhau
- có thể tìm thấy nhau không cần sự can thiệp của router

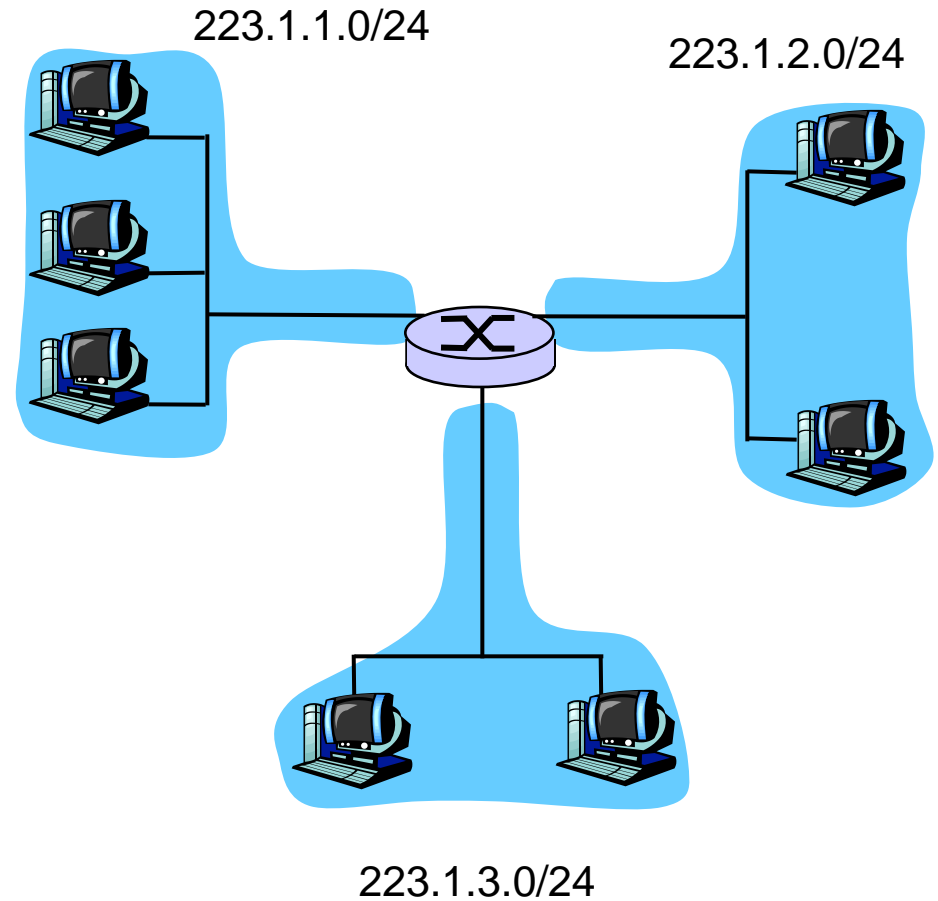


mạng gồm 3 subnets

Subnets

phương pháp

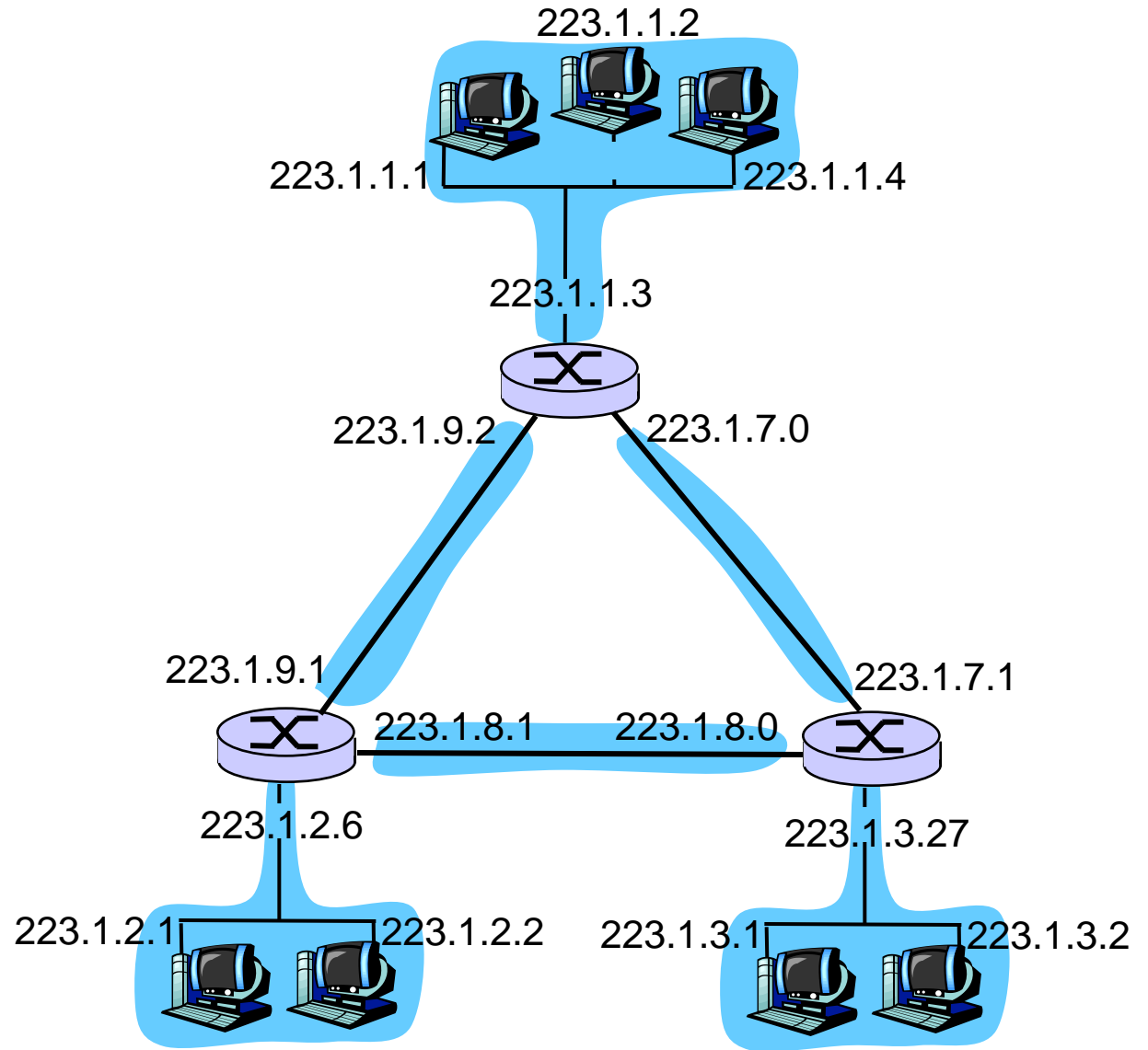
- Để xác định subnet, tách mỗi interface từ host hoặc router của nó, tạo vùng các mạng độc lập. Mỗi vùng mạng độc lập được gọi là một **subnet**.



Subnet mask: /24

Subnets

Bao nhiêu?

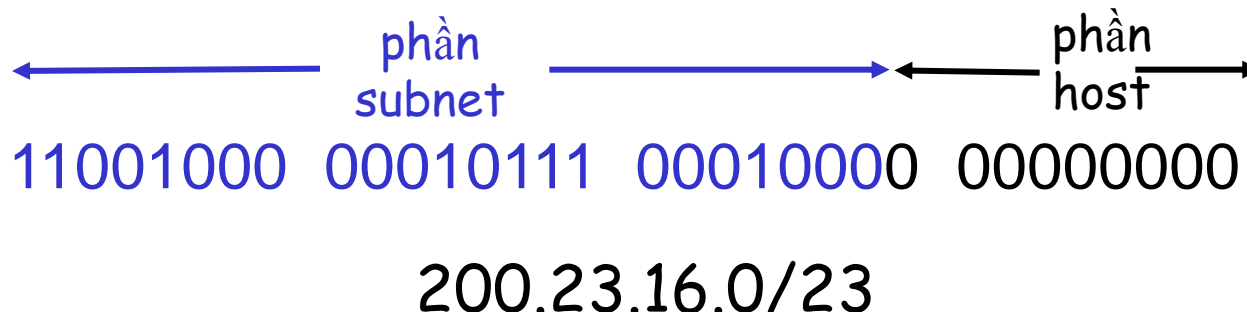


Lớp Network

Định địa chỉ IP: CIDR

CIDR: Classless InterDomain Routing

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



các địa chỉ IP: làm sao lấy một?

Hỏi: Làm sao *host* lấy được địa chỉ IP?

- ❑ mã hóa cứng do người quản trị hệ thống trong 1 file
 - Wintel: 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”
- (xem chương kế tiếp để biết rõ hơn)

các địa chỉ IP: làm sao lấy một?

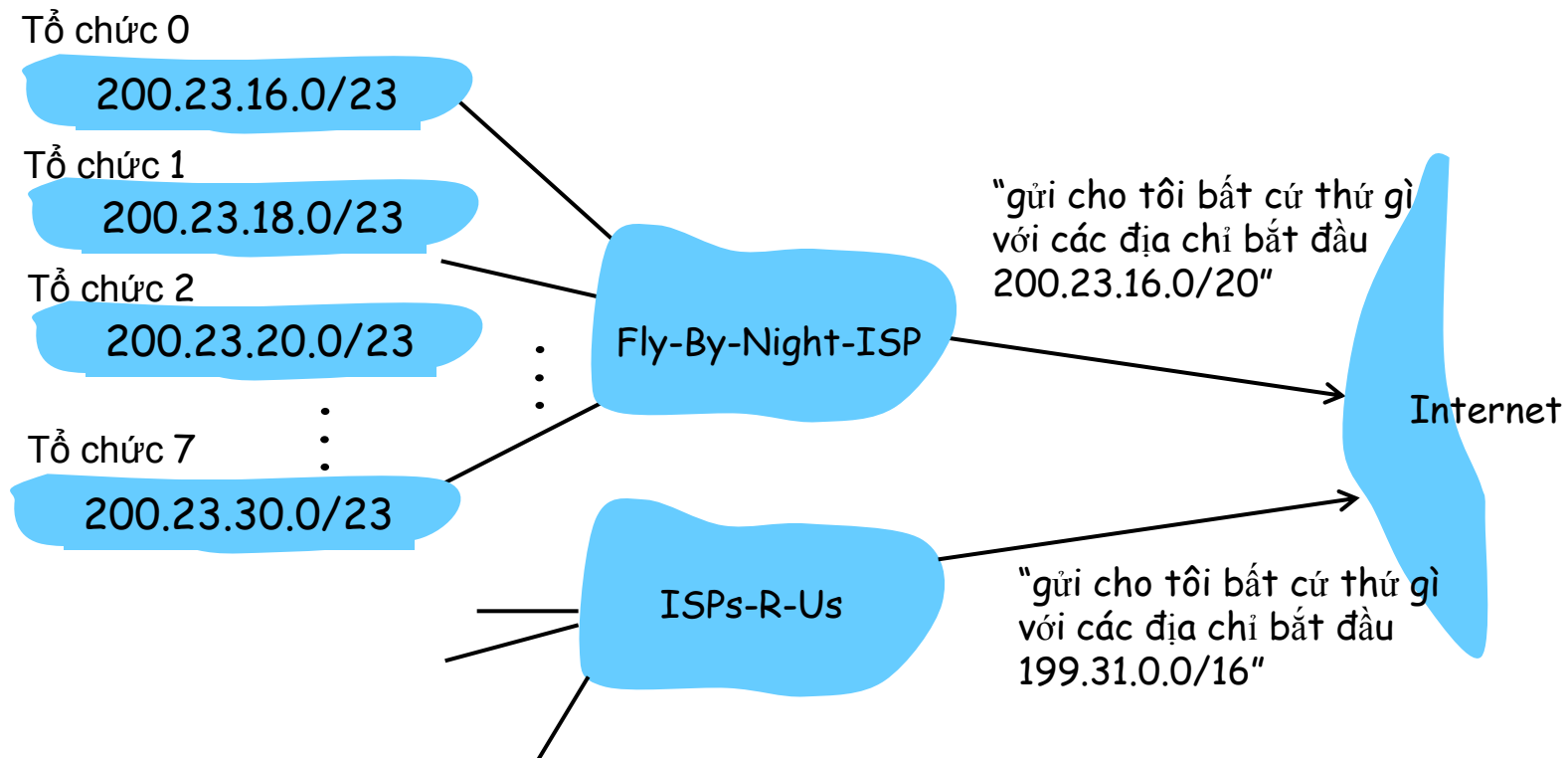
Hỏi: Làm sao *mạng* lấy được phần subnet của địa chỉ IP?

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

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

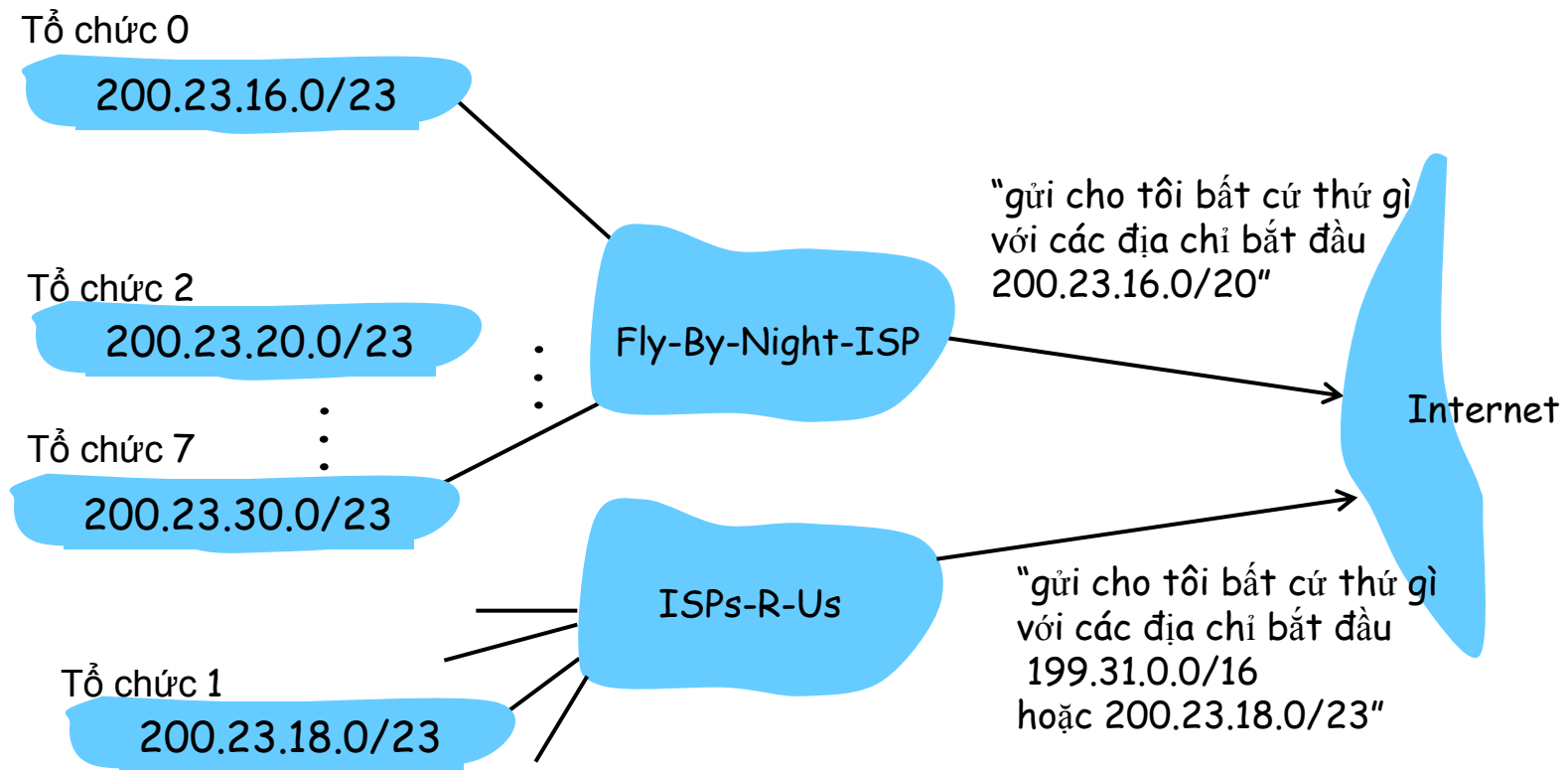
Định địa chỉ phân cấp: route tích hợp

cho phép thông báo hiệu quả thông tin routing:



Định địa chỉ phân cấp: nhiều cách route xác định

ISPs-R-Us có nhiều cách route đến Tổ chức 1



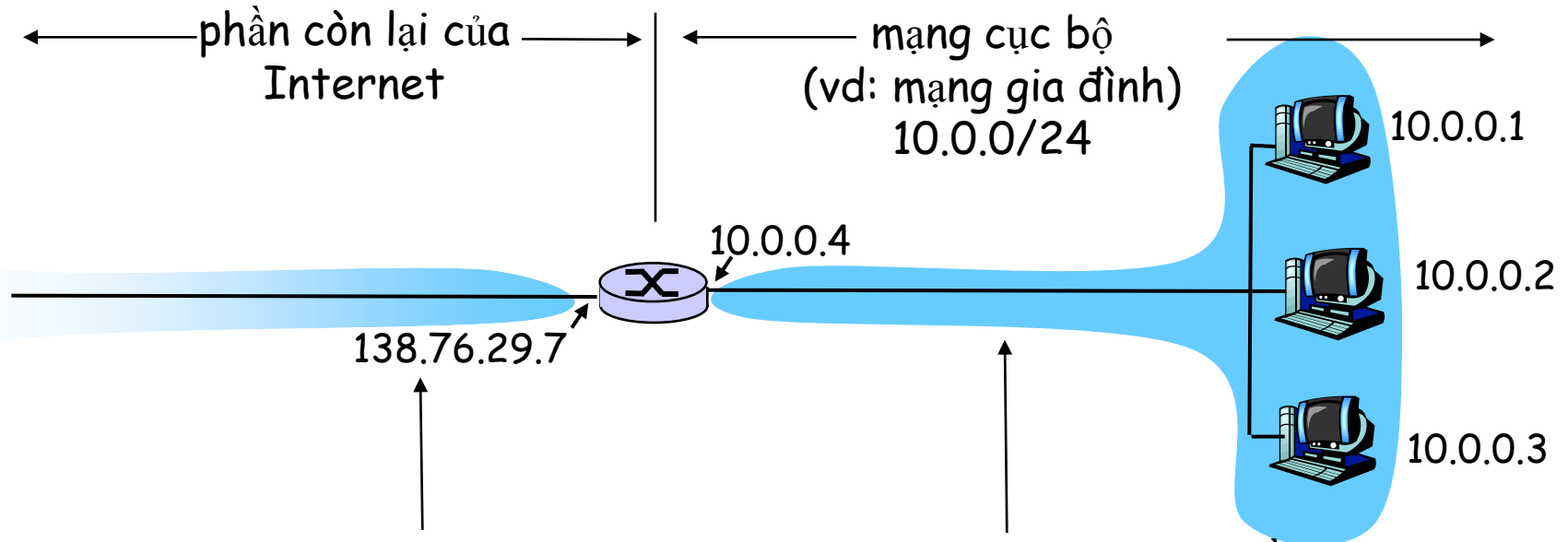
Định địa chỉ IP:...

Hỏi: Làm sao một ISP lấy được khối địa chỉ?

Đáp: **ICANN**: Internet **C**orporation for **A**ssigned
Names and **N**umbers

- cấp phát các 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 10.0.0/24

NAT: Network Address Translation

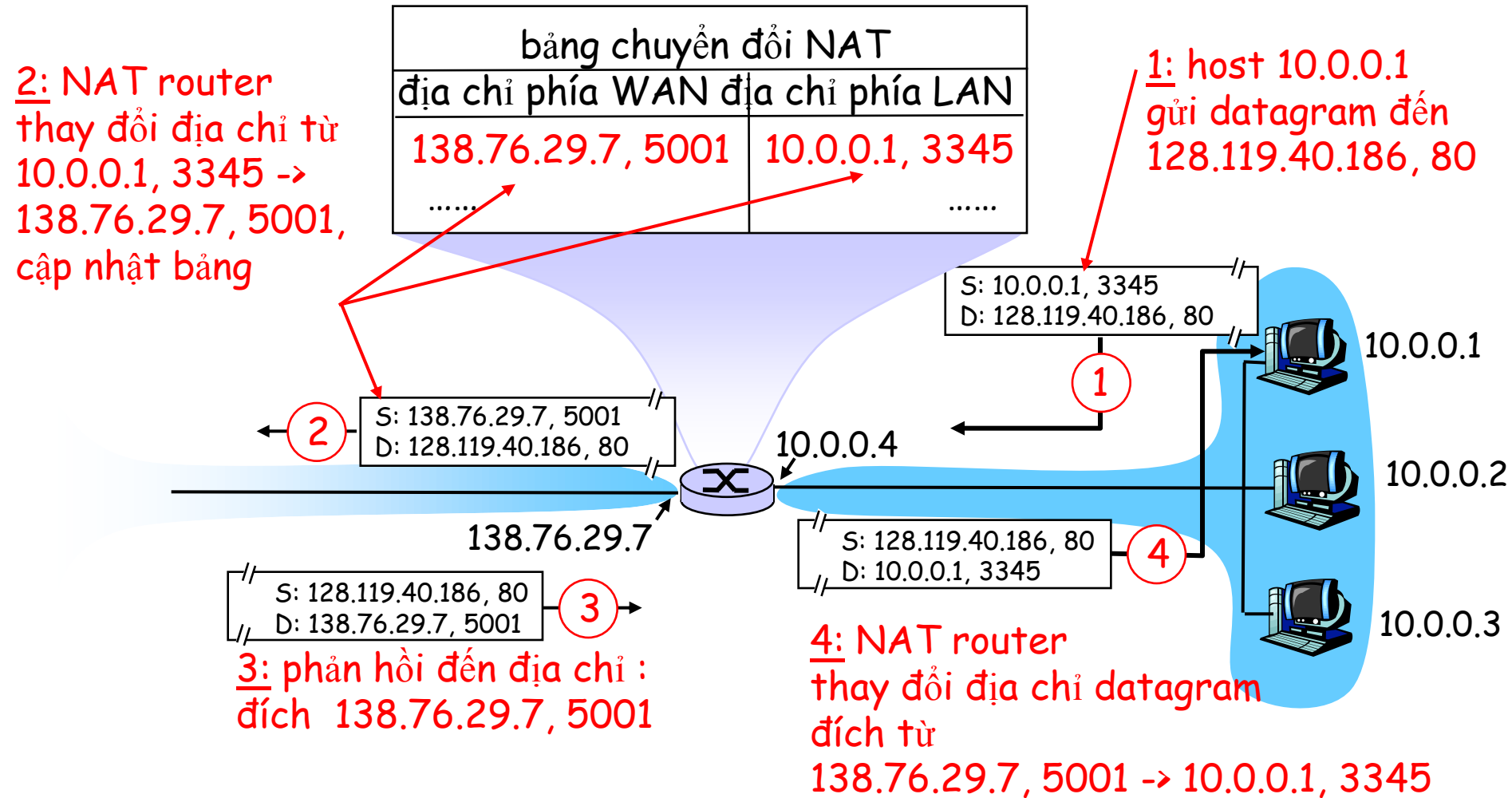
- ❑ Mạng cục bộ chỉ dùng 1 địa chỉ IP đối với bên ngoài:
 - không cần thiết dùng 1 vùng địa chỉ từ ISP: chỉ cần 1 cho tất cả các thiết bị
 - có thể thay đổi địa chỉ các thiết bị trong mạng cục bộ mà không cần thông báo với bên ngoài
 - có thể thay đổi ISP mà không cần thay đổi địa chỉ các thiết bị trong mạng cục bộ
 - các thiết bị trong mạng cục bộ không nhìn thấy, không định địa chỉ rõ ràng từ bên ngoài (tăng cường bảo mật)

NAT: Network Address Translation

Hiện thực: NAT router phải:

- *các datagram đi ra: thay thế* (địa chỉ IP và số hiệu cổng nguồn) mọi datagram đi ra bên ngoài bằng (địa chỉ NAT IP và số hiệu cổng nguồn mới)
... các clients/servers ở xa sẽ dùng (địa chỉ NAT IP và số hiệu cổng nguồn mới) đó như địa chỉ đích
- *ghi nhớ (trong bảng chuyển đổi NAT)* mọi cặp chuyển đổi (địa chỉ IP và số hiệu cổng nguồn) sang (địa chỉ NAT IP và số hiệu cổng nguồn mới)
- *các datagram đi đến: thay thế* (địa chỉ NAT IP và số hiệu cổng nguồn 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 cổng 16-bit:
 - 60,000 kết nối đồng thời chỉ với một địa chỉ phía LAN
- ❑ NAT còn có thể gây ra tranh luận:
 - các router chỉ xử lý đến lớp 3
 - vi phạm thỏa thuận end-to-end
 - những người thiết kế ứng dụng phải tính đến khả năng NAT, vd: ứng dụng P2P
 - sự thiếu thôn địa chỉ IP sẽ được giải quyết khi dùng IPv6

ICMP: Internet Control Message Protocol

- ❑ được các host & router dùng để truyền thông thông tin lớp network

- Thông báo lỗi: host, network, port, giao thức không có thực
- phản hồi request/reply (dùng bởi lệnh ping)

- ❑ lớp network "trên" IP:

- các thông điệp ICMP chứa trong các IP datagram

- ❑ **thông điệp ICMP:** kiểu, mã thêm với 8 byte đầu tiên của IP datagram gây ra lỗi

kiểu	mã	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 & ICMP

- ❑ nguồn gửi một chuỗi các đoạn UDP đến đích
 - đầu tiên có TTL = 1
 - thứ hai có TTL=2, tương tự.
 - không giống số port
 - ❑ khi datagram thứ n đến router n:
 - Router hủy datagram
 - và gửi đến nguồn một ICMP message (kiểu 11, mã 0)
 - thông điệp chứa tên của địa chỉ router & IP
 - ❑ Khi thông điệp ICMP đến, nguồn tính toán RTT
 - ❑ Traceroute thực hiện công việc này 3 lần
- tiêu chuẩn dừng
- ❑ đoạn UDP đến lần lượt tại host đích
 - ❑ đích trả về gói ICMP "host không có thực" (kiểu 3, mã 3)
 - ❑ Khi nguồn có ICMP này -> dừng.

IPv6

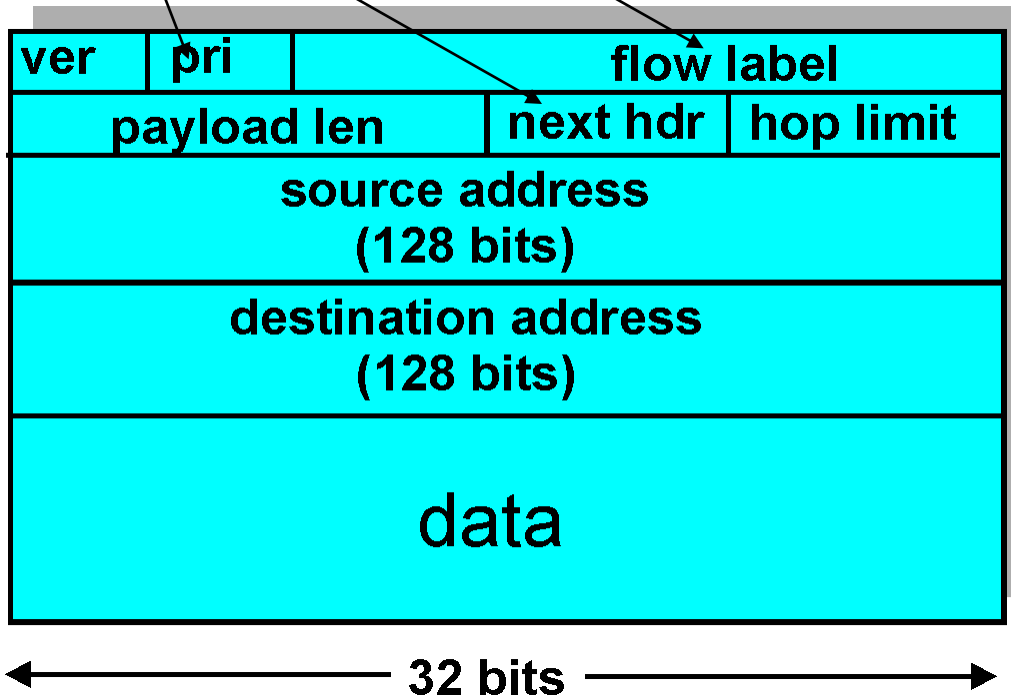
- ❑ động lực thúc đẩy ban đầu: *không gian địa chỉ 32-bit sớm được cấp phát cạn kiệt.*
- ❑ động lực bổ sung:
 - dạng thức header giúp tăng tốc xử lý/forwarding
 - header thay đổi tạo điều kiện thuận lợi cho QoS
- dạng thức IPv6 datagram:
 - 40 byte header, độ dài cố định
 - không cho phép phân mảnh

IPv6 Header (tt)

độ ưu tiên: xác định độ ưu tiên của các datagram trong luồng

nhãn luồng: xác định các datagram trong cùng "luồng"
(khái niệm "luồng" không được rõ ràng).

header kế tiếp: xác định giao thức lớp trên cho dữ liệu



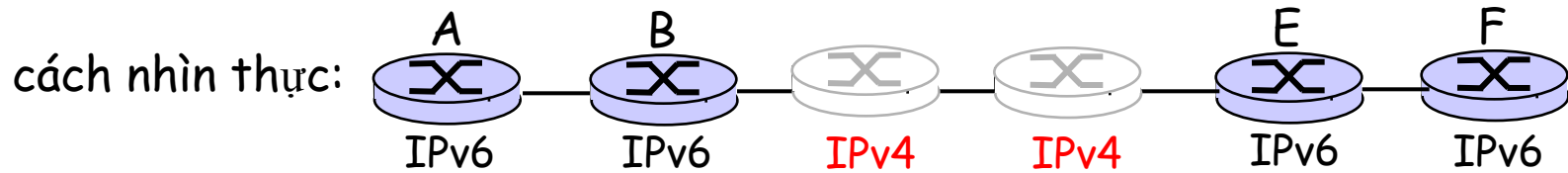
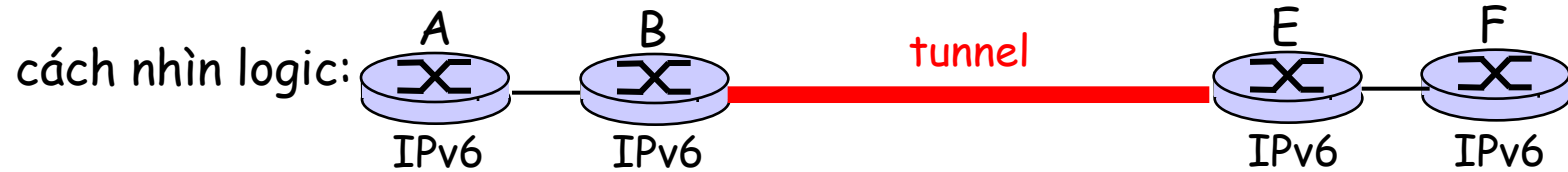
Những thay đổi khác nữa so với IPv4

- ❑ *Checksum*: bỏ hết, nhằm giảm thời gian xử lý tại hop
- ❑ *Options*: cho phép, nhưng nằm ngoài header, chỉ thị bởi trường "Next Header"
- ❑ *ICMPv6*: phiên bản mới của ICMP
 - các kiểu thông điệp bổ sung, vd "Packet Too Big"
 - các chức năng quản lý nhóm multicast

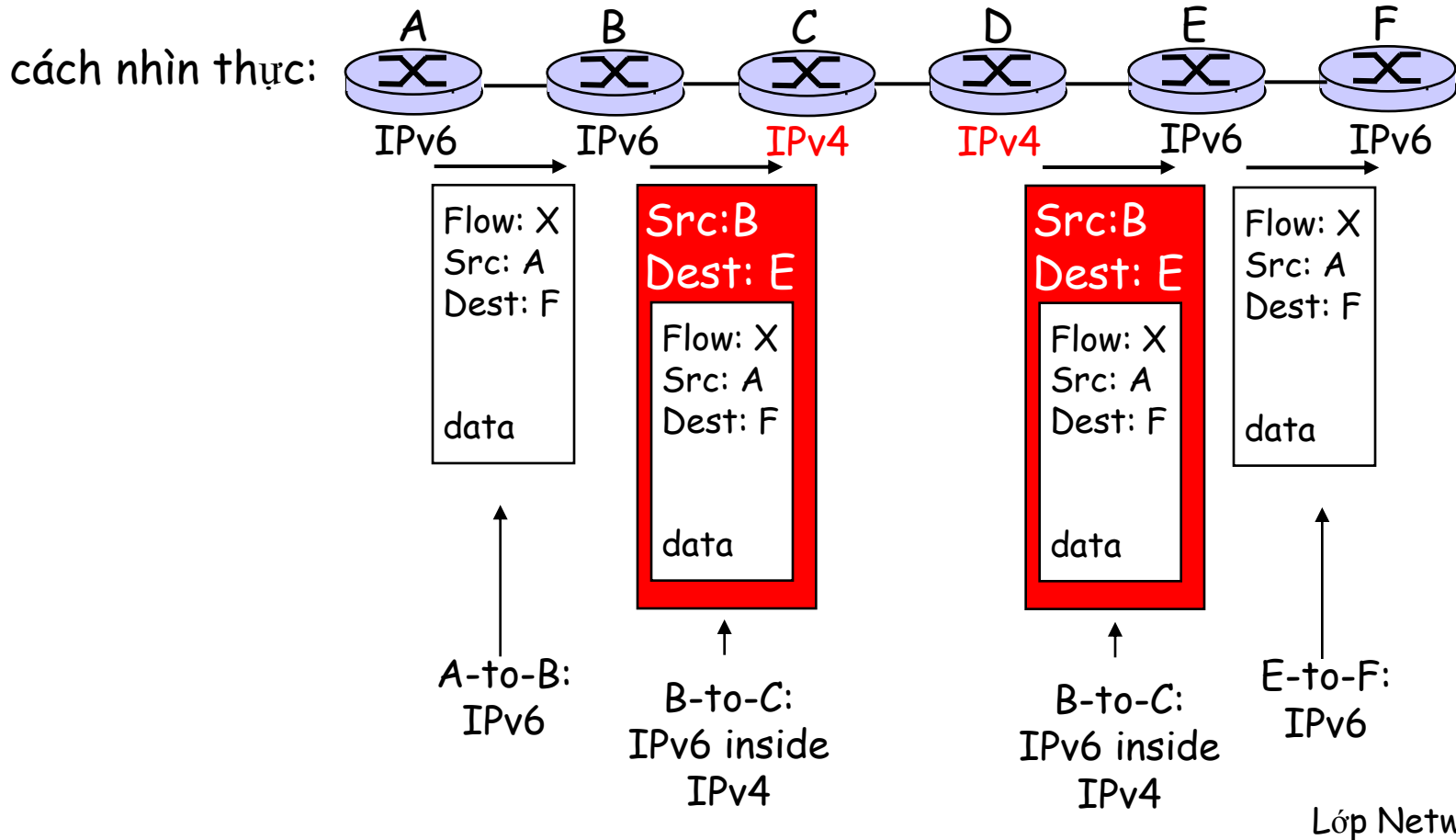
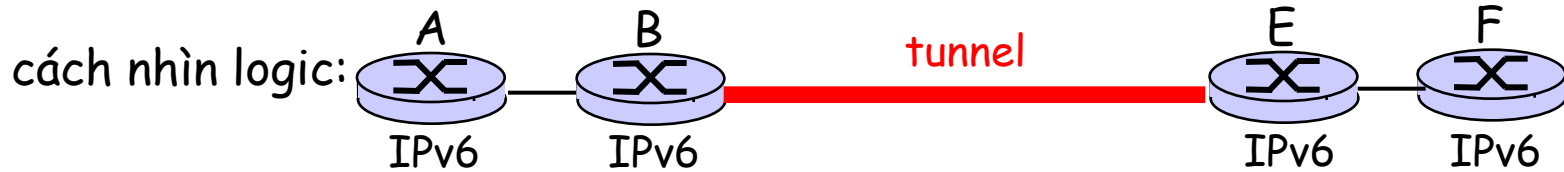
Chuyển từ IPv4 sang IPv6

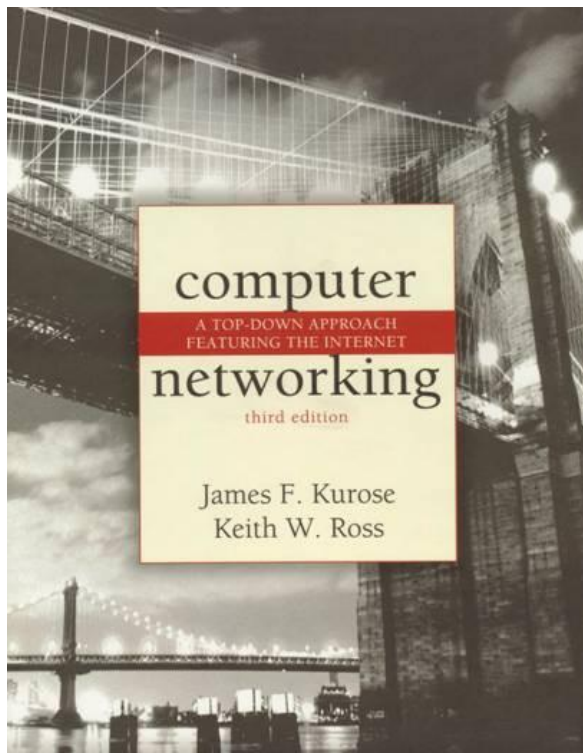
- ❑ không phải tất cả router đều có thể nâng cấp đồng thời
 - mạng có các router dùng cả IPv4 và IPv6 hoạt động thế nào?

Tunneling



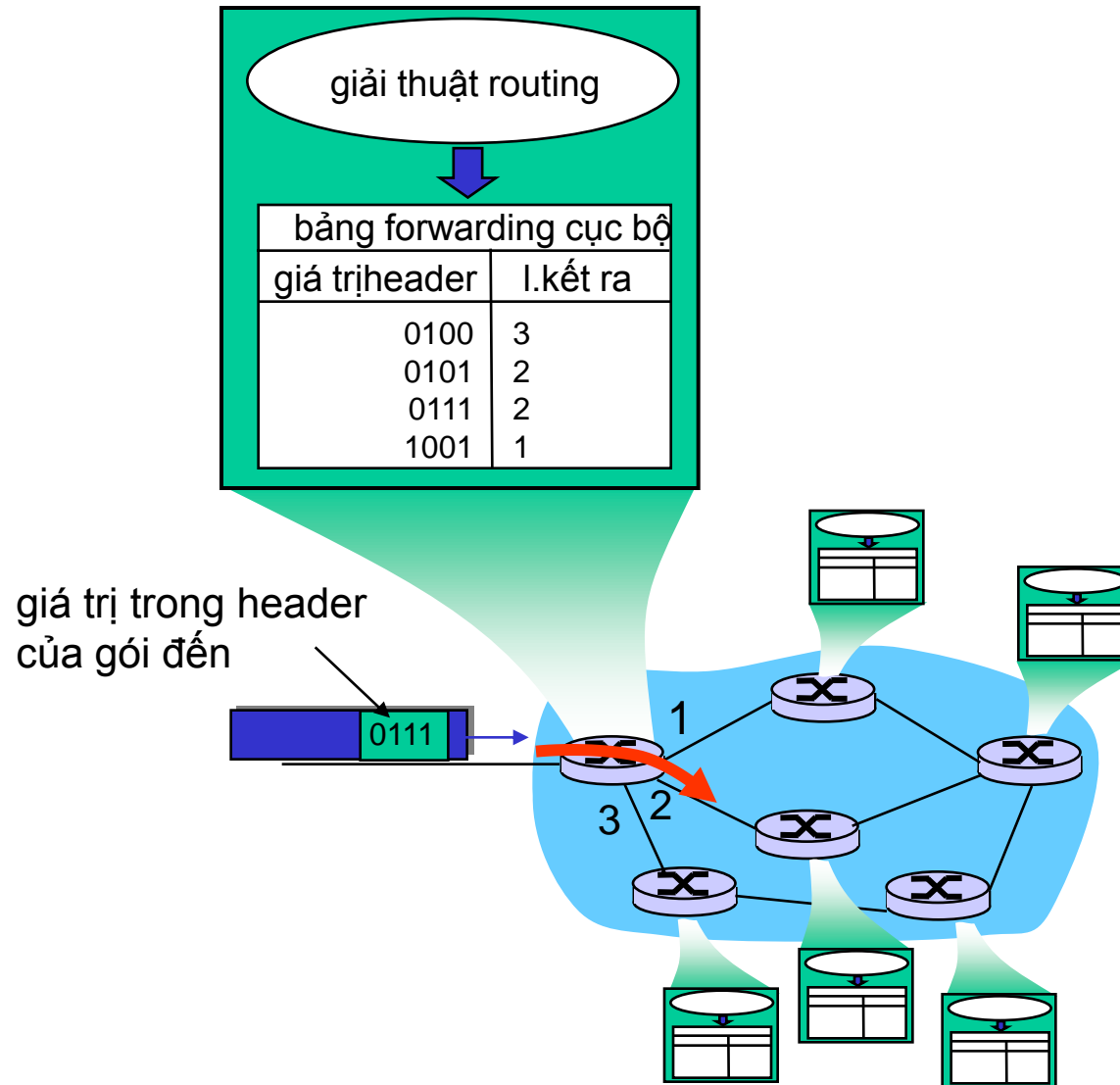
Tunneling



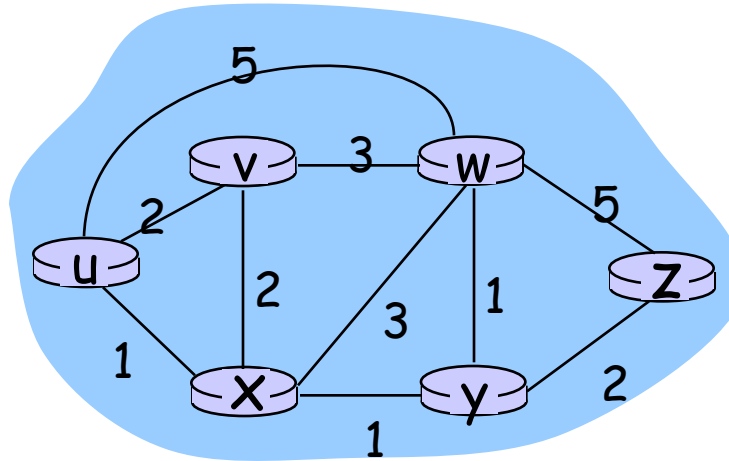


4.5 Các giải thuật Routing

Tác động lẫn nhau giữa routing, forwarding



Mô hình đồ thị



đồ thị: $G = (N, E)$

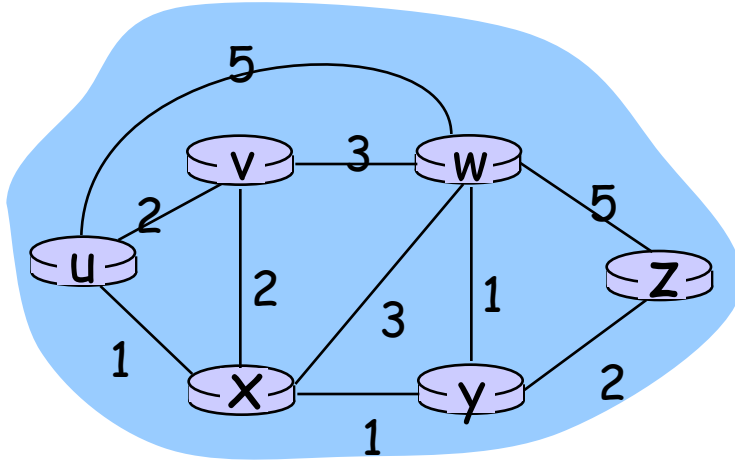
N = tập các routers = $\{ u, v, w, x, y, z \}$

E = tập 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 dùng được trong những ngữ cảnh khác

Ví dụ: P2P, trong đó N là tập các điểm và E là tập các kết nối TCP

Mô hình đồ thị: các chi phí



- $c(x,x')$ = chi phí kết nối (x,x')

- ví dụ: $c(w,z) = 5$

- chi phí có thể luôn luôn là 1, hoặc ngược lại liên quan đến băng thông, hay liên quan đến tắc nghẽn

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

Hỏi: chi phí thấp nhất trên đường từ u đến z ?

giải thuật Routing: giải thuật tìm đường có chi phí thấp nhất

phân lớp giải thuật Routing

thông tin toàn cục hoặc không tập trung

toàn cục:

- ❑ tất cả router có toàn bộ thông tin về chi phí kết nối, cấu trúc mạng
- ❑ các giải thuật "trạng thái kết nối" - Link State

không tập trung:

- ❑ biết các kết nối vật lý đến các điểm lân cận và chi phí của nó
- ❑ lặp lại quá trình tính toán, trao đổi thông tin với các điểm lân cận
- ❑ các giải thuật "vector khoảng cách" - Distance Vector

Tĩnh hay động?

Tĩnh:

- ❑ việc tìm đường đi thay đổi chậm chạp theo thời gian

Động:

- ❑ việc tìm đường đi thay đổi rất nhanh
 - cập nhật theo chu kỳ
 - phản ứng với những thay đổi chi phí kết nối

1 giải thuật Routing "trạng thái kết nối"

giải thuật Dijkstra

- biết chi phí kết nối, cấu trúc mạng của tất cả các nút
 - tất cả các nút có thông tin giống nhau
- tính toán đường đi chi phí thấp nhất từ 1 nút (nguồn) đến tất cả các nút khác
 - cho trước bảng **forwarding** của nút đó
- sau k lần duyệt, biết được đường đi chi phí thấp nhất của k đích

Ký hiệu:

- **$c(x,y)$** : chi phí kết nối từ nút x đến y; $= \infty$ nếu không kết nối trực tiếp đến điểm lân cận
- **$D(v)$** : giá trị chi phí hiện tại của đường từ nguồn đến đích v
- **$p(v)$** : nút trước nằm trên đường từ nguồn đến nút v
- **N'** : tập các nút mà đường đi chi phí 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 có trong N' nhưng $D(w)$ tối thiểu

10 thêm w vào N'

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

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

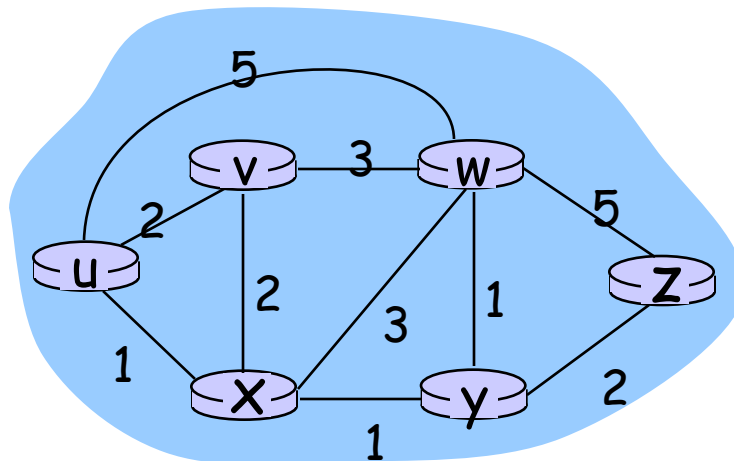
13 /* chi phí mới đến v là chính nó hoặc chi phí đường đi ngắn nhất

14 cộng với chi phí từ w đến v */

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

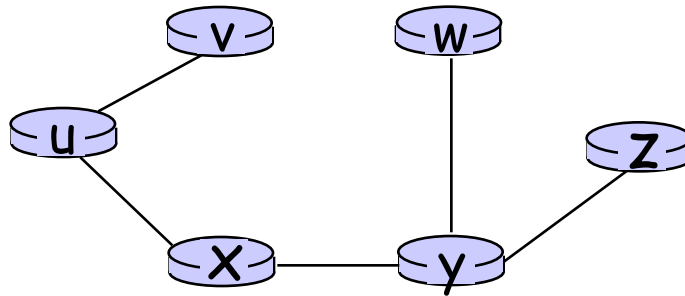
giải thuật Dijkstra: ví dụ

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



giải thuật Dijkstra: ví dụ (2)

Cây kết quả đường đi ngắn nhất từ u:



Bảng forwarding kết quả trong u:

đích	kết nối
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

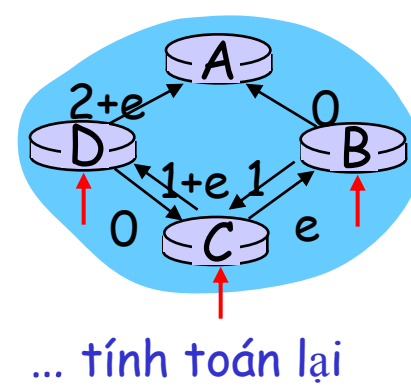
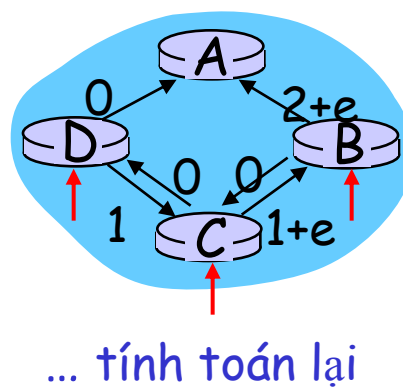
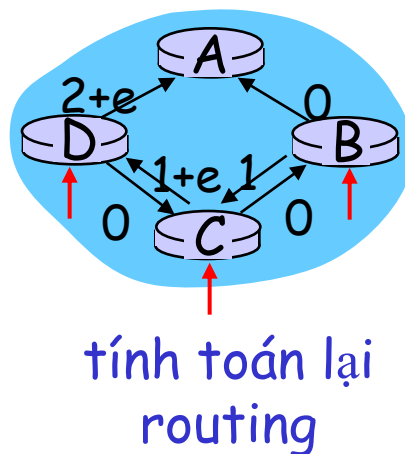
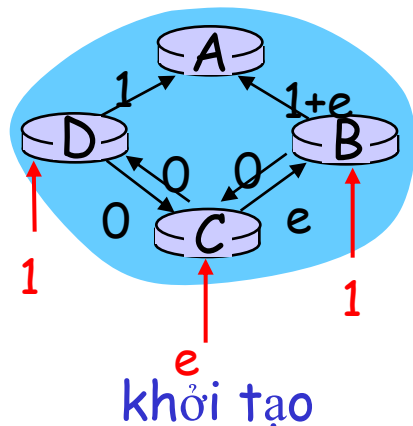
giải thuật Dijkstra: thảo luận

Độ phức tạp giải thuật: n nút

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

có thể dao động:

- vd: chi phí kết nối = số lượng lưu thông



giải thuật Vector khoảng cách

công thức Bellman-Ford

định nghĩa

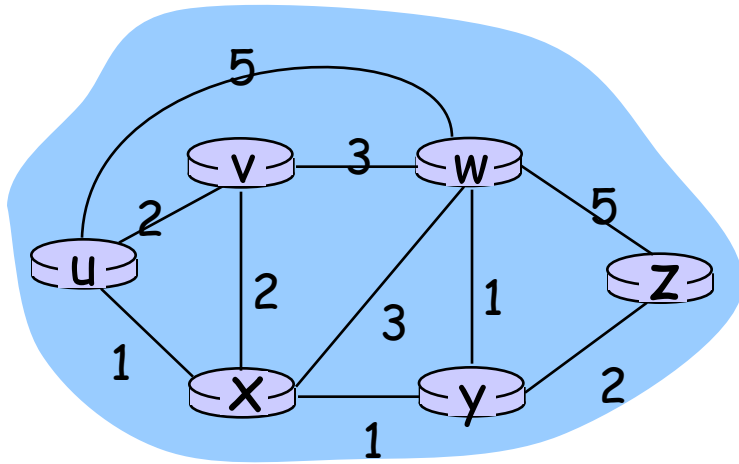
$d_x(y)$:= chi phí thấp nhất của đường đi từ x đến y

thì

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

trong đó min được tính trên tất cả lân cận v của x

Bellman-Ford: ví dụ



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

công thức B-F cho:

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

chú ý rằng tối tiểu đạt được là hop kế tiếp
trong đường đi ngắn nhất

giải thuật Vector khoảng cách

- $D_x(y)$ = ước lượng chi phí thấp nhất từ x đến y
- nút x biết chi phí đến mỗi lân cận v : $c(x,v)$
- Nút x duy trì vector khoảng cách $D_x = [D_x(y): y \in N]$
- Nút x cũng duy trì các vector khoảng cách đến các lân cận của nó
 - với mỗi lân cận v , x duy trì $D_v = [D_v(y): y \in N]$

giải thuật Vector khoảng cách (4)

Ý tưởng chính:

- mỗi nút định kỳ gửi ước lượng vector khoảng cách của nó đến các lân cận
- khi 1 nút x nhận ước lượng D_v mới từ lân cận, nó cập nhật DV của mình dùng công thức B-F:

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

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

giải thuật Vector khoảng cách (5)

lập, không đồng bộ: mỗi lập cục bộ được gây ra bởi:

- ❑ chi phí kết nối cục bộ thay đổi
- ❑ DV cập nhật thông báo từ lân cận

phân bố:

- ❑ mỗi nút thông báo đến các lân cận *chỉ khi* DV của nó thay đổi
 - các lân cận sau đó thông báo đến các lân cận của nó nếu cần thiết

mỗi nút:

chờ cho (thay đổi trong chi phí kết nối cục bộ hoặc thông báo từ lân cận)

tính toán lại các ước lượng

nếu DV đến bất kỳ đích nào có thay đổi, *thông báo* cho các 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 đến		
		x	y	z
từ	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

chi phí đến

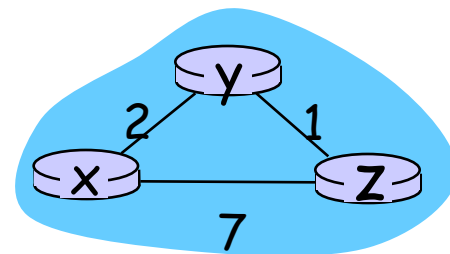
		x	y	z
từ	x	0	2	3
	y	2	0	1
	z	7	1	0

bảng nút y

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

bảng nút z

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



thời gian

Lớp Network

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

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

bảng nút x

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

bảng nút y

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

bảng nút z

	chi phí đến		
	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

chi phí đến

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

chi phí đến

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

chi phí đến

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

chi phí đến

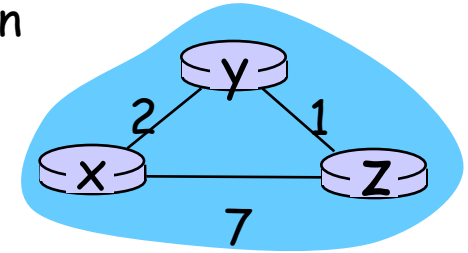
	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

chi phí đến

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

chi phí đến

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



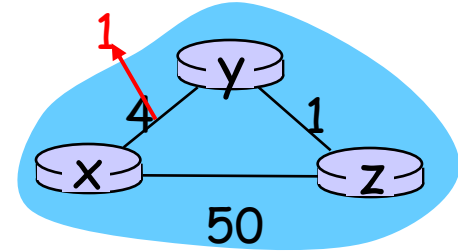
thời gian

Lớp Network

Vector khoảng cách: các thay đổi chi phí kết nối

các thay đổi chi phí kết nối:

- ❑ nút kiểm tra thay đổi chi phí kết nối
- ❑ cập nhật thông tin dẫn đường, tính toán lại vector khoảng cách
- ❑ nếu DV thay đổi, thông báo các lân cận



“duyet
tin tức
tốt
nhanh”

Tại thời điểm t_0 , y kiểm tra thay đổi chi phí kết nối, cập nhật DV và thông báo đến các lân cận của nó

Tại thời điểm t_1 , z nhận được cập nhật từ y và cập nhật bảng của nó. Nó tính toán chi phí thấp nhất mới đến x và gửi DV của nó đến các lân cận

Tại thời điểm t_2 , y nhận được cập nhật của z và cập nhật bảng khoảng cách của nó. Các chi phí thấp nhất của y không thay đổi và hơn nữa y không gửi bất kỳ thông báo nào đến z.

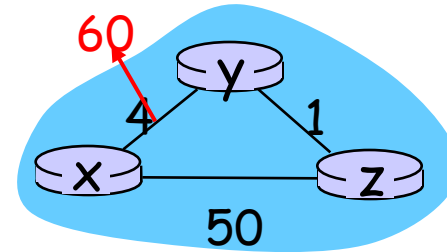
Vector khoảng cách: các thay đổi chi phí kết nối

các thay đổi chi phí kết nối:

- ❑ duyệt tin tức tốt nhanh
- ❑ duyệt tin tức xấu chậm - vấn đề "đếm đến vô tận"!
- ❑ 44 lần duyệt trước khi ổn định

Poisoned reverse:

- ❑ Nếu Z dẫn đường từ Y thẳng tới X:
 - Z nói với Y khoảng cách của nó đến X là ∞ (vì thế Y sẽ không dẫn đường đến X đi qua Z)
- ❑ sẽ giải quyết triệt để vấn đề đếm vô tận?



So sánh các giải thuật LS và DV

thông báo phức tạp

- ❑ LS: với n nút, E kết nối, $O(nE)$ các thông báo được gửi
- ❑ DV: chỉ trao đổi giữa các lân cận

Tốc độ hội tụ

- ❑ LS: giải thuật $O(n^2)$ yêu cầu $O(nE)$ thông báo
 - có thể có các dao động
- ❑ DV: thời gian hội tụ thay đổi
 - có thể do các quá trình lặp tìm đường
 - vấn đề đếm vô hạn

sự linh hoạt: điều gì 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
- bảng của nút có thể được nút khác dùng
 - lỗi lan truyền thông qua mạng

Hierarchical Routing

ngiên cứu trong môi trường lý tưởng hóa

- ❑ tất cả các router đồng nhất
- ❑ mạng “phẳng”

... không đúng trong thực tế

quy mô: với 200 triệu
đích

- ❑ không thể ghi nhớ tất cả đích trong bảng routing!
- ❑ bảng routing sẽ kiểm soát các kết nối!

quản trị

- ❑ internet = mạng của các mạng
- ❑ mỗi quản trị mạng muốn điều hành routing trong mạng của họ

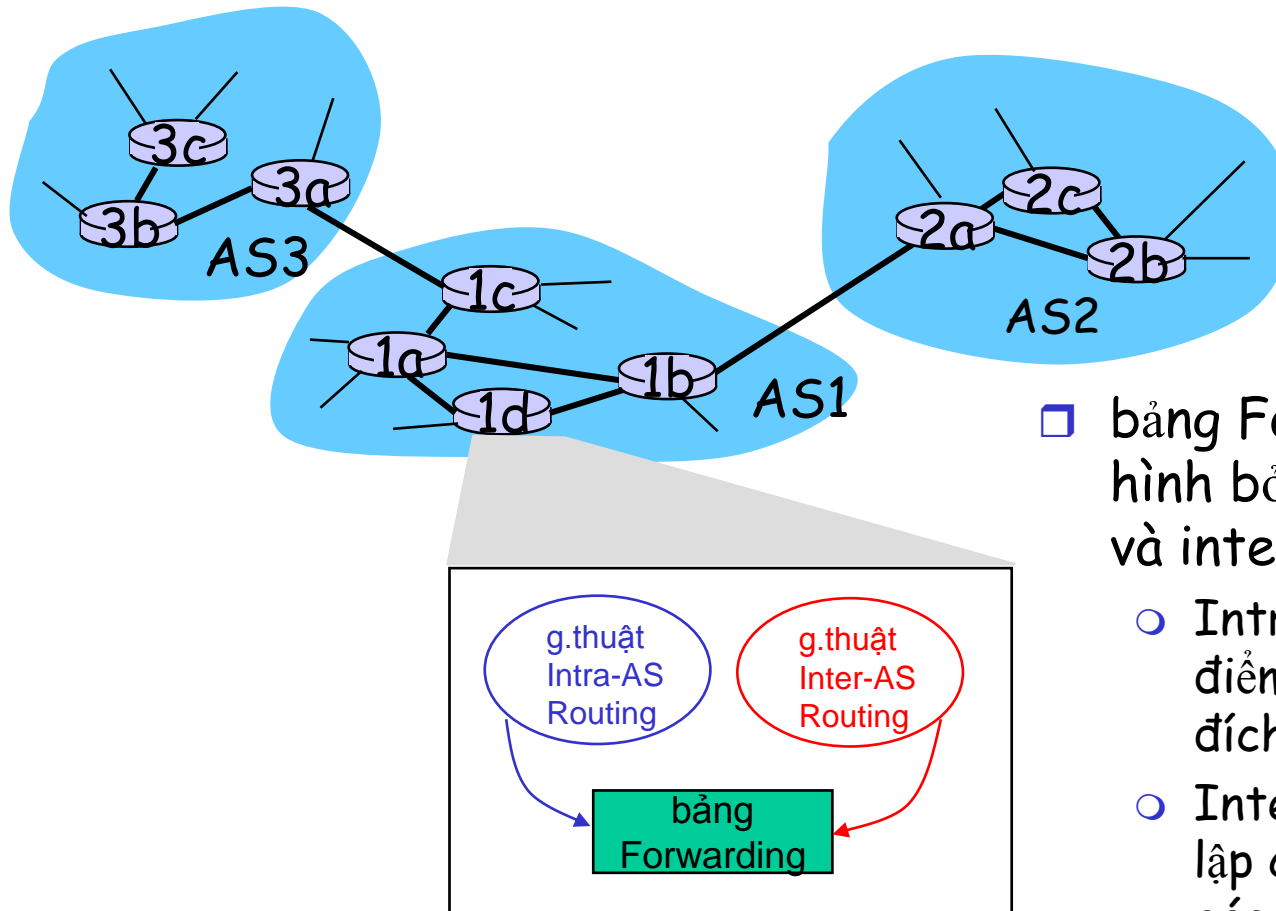
Hierarchical Routing

- ❑ các router gom thành các vùng, “các hệ thống tự trị-
autonomous systems” (AS)
- ❑ các router trong cùng AS chạy giao thức routing giống nhau
 - giao thức “**intra-AS**”
routing
 - các router trong AS khác nhau có thể chạy giao thức intra-AS routing

Gateway router

- ❑ trực tiếp kết nối đến router trong AS khác

Kết nối các AS



- bảng Forwarding được cấu hình bởi cả giải thuật intra- và inter-AS routing
 - Intra-AS thiết lập các điểm đăng nhập vào các đích nội mạng
 - Inter-AS & Intra-As thiết lập các điểm đăng nhập vào các đích ngoại mạng

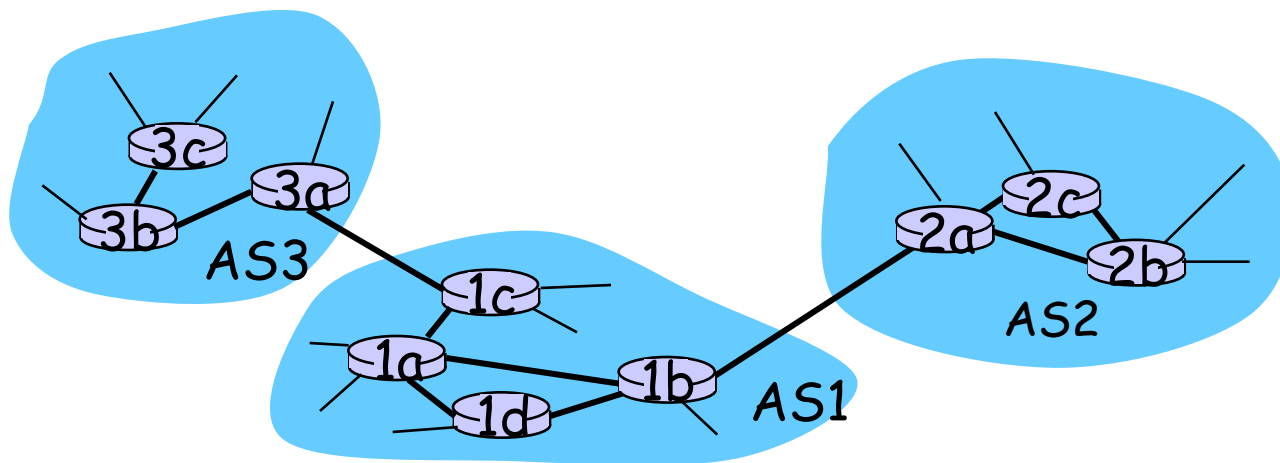
các tác vụ Inter-AS

- ❑ giả sử router AS1 nhận datagram với đích nằm ngoài nó
 - Router sẽ forward về một trong những gateway router kế tiếp, nhưng là cái nào?

AS1 cần:

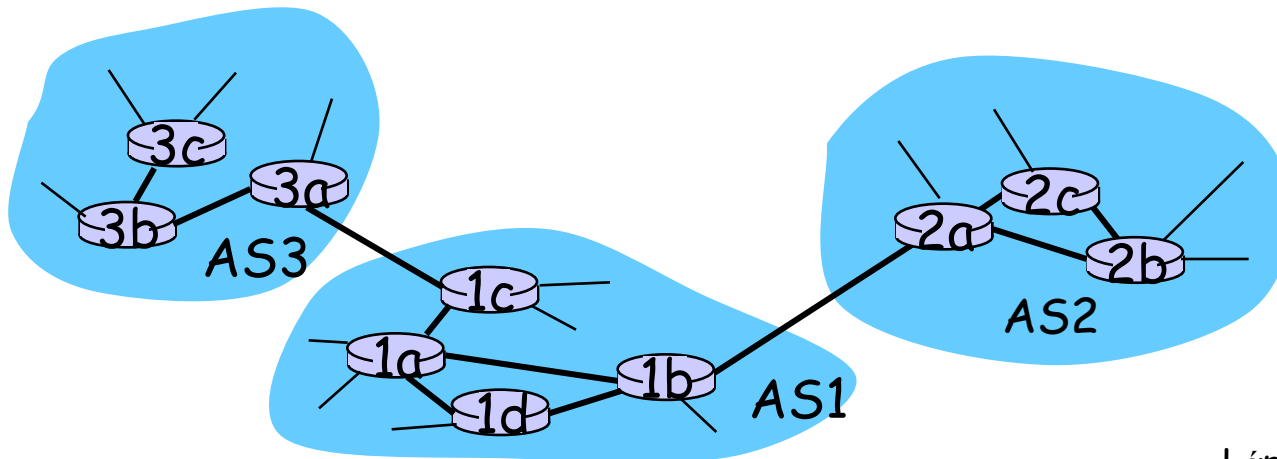
1. học các đích nào có thể chạm đến thông qua AS2 và AS3
2. lan truyền thông tin này đến tất cả các router trong AS1

công việc của inter-AS routing!



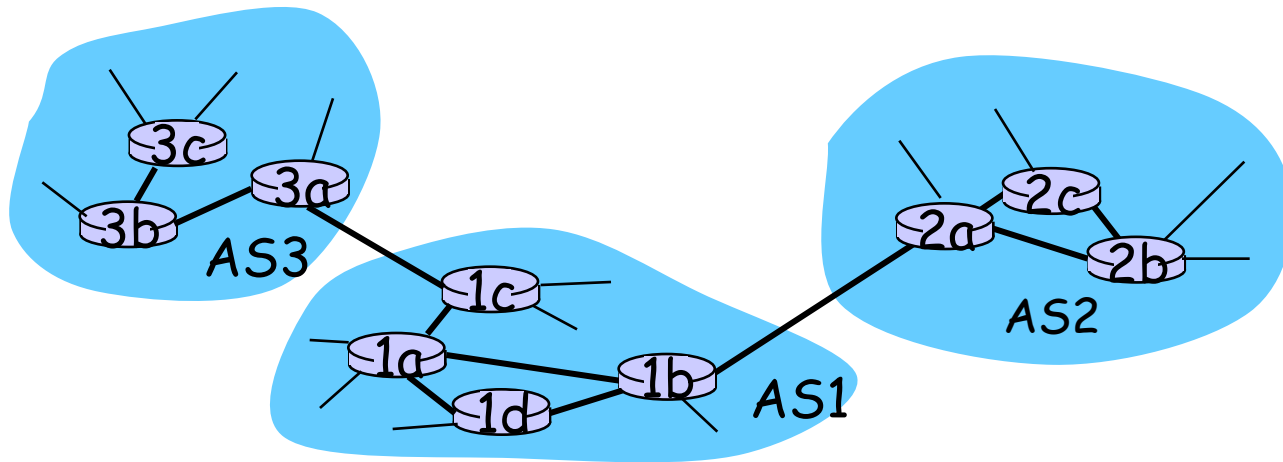
Ví dụ: thiết lập bảng forwarding trong router 1d

- ❑ Giả sử AS1 học (thông qua giao thức inter-AS) mà subnet **x** có thể chạm đến 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 intra-AS routing và **I** sẽ nằm trên đường đi chi phí thấp nhất đến 1c.
- ❑ Đưa giá trị **(x,I)** vào *bảng* forwarding



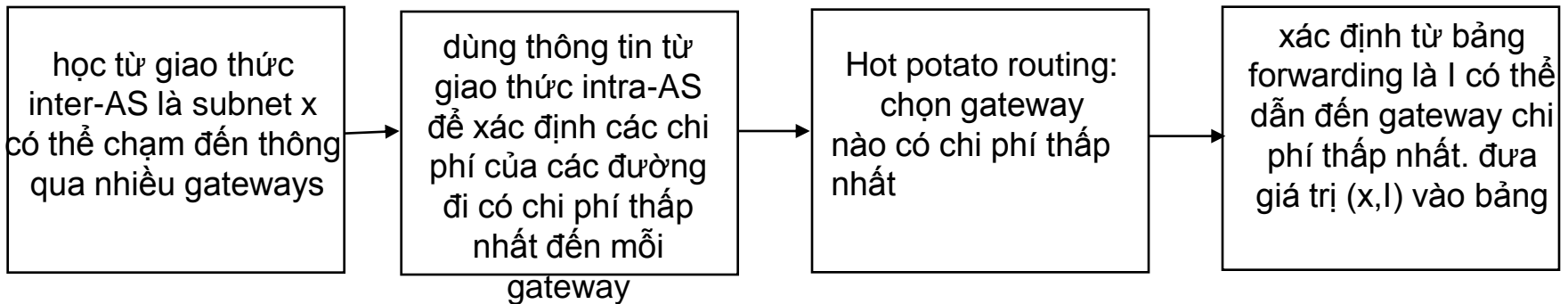
Ví dụ: Chọn giữa nhiều AS

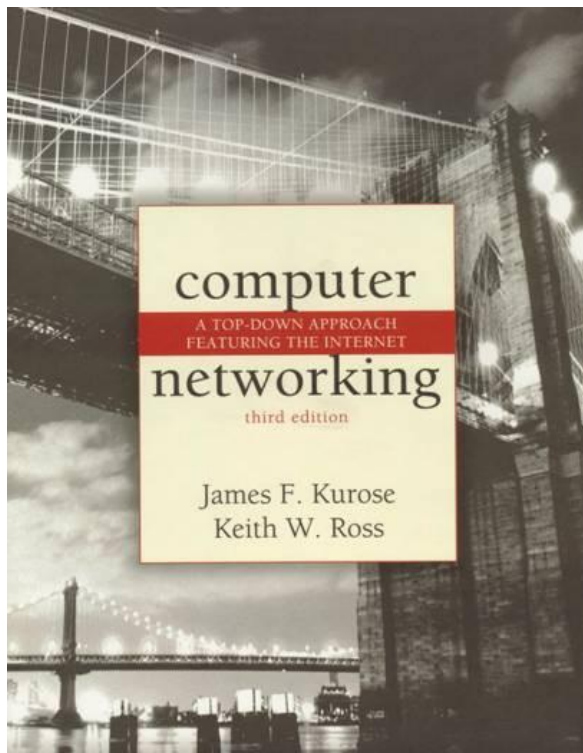
- Bây giờ giả sử AS1 học từ giao thức inter-AS là subnet **x** có thể chạm đến từ AS3 và từ AS2
- Để cấu hình bảng forwarding, router 1d phải xác định gateway nào được dùng để chuyển các gói đến đích **x**.
- đây cũng chính là công việc trên giao thức inter-AS routing!



Ví dụ: Chọn giữa nhiều AS

- **Hot potato routing:** gửi các gói đến 2 router gần nhất





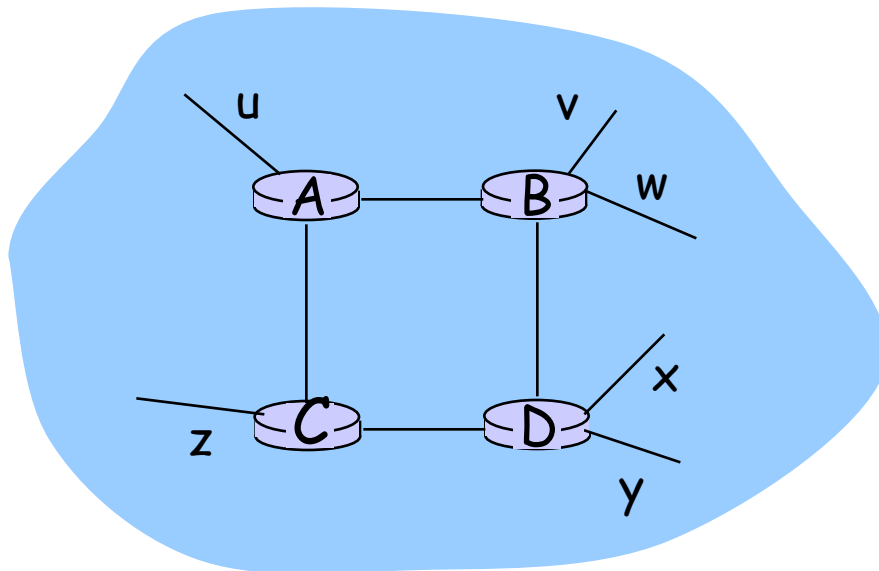
4.6 Routing trong Internet

Intra-AS Routing

- ❑ cũng gọi là **Interior Gateway Protocols (IGP)**
- ❑ các giao thức Intra-AS routing phổ biến:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IGRP: Interior Gateway Routing Protocol (Cisco độc quyền)

RIP (Routing Information Protocol)

- ❑ giải thuật vector khoảng cách
- ❑ công bố năm 1982 trong BSD-UNIX Distribution
- ❑ không gian khoảng cách: số lượng hop (tối đa 15 hop)



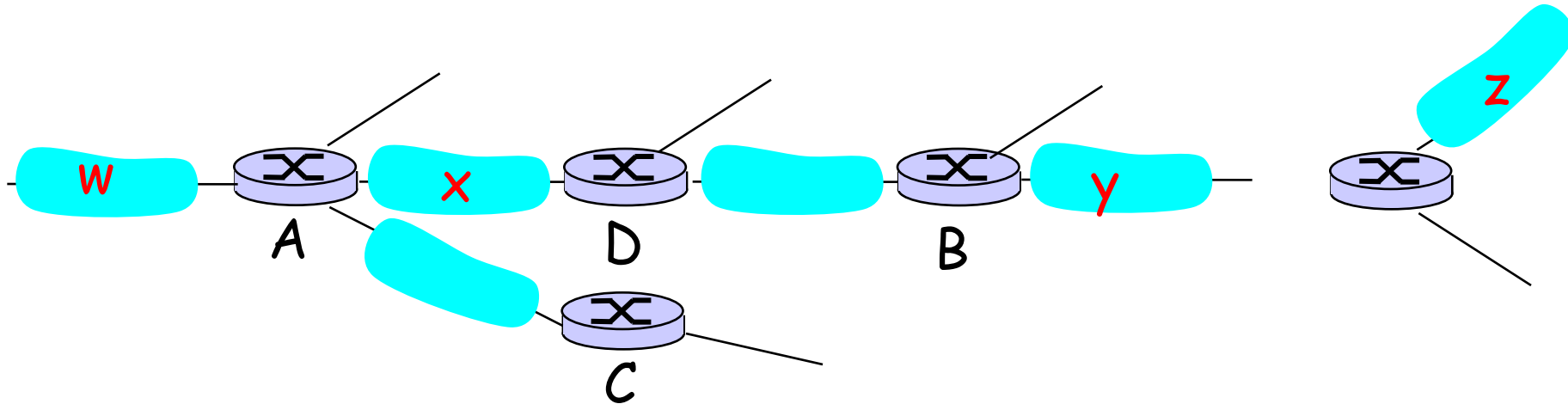
từ router A đến các subset:

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

các thông báo của RIP

- ❑ Các vector khoảng cách: trao đổi giữa các lân cận mỗi 30s thông qua Response Message (cũng được gọi là **thông báo**).
- ❑ Mỗi thông báo: danh sách lên đến 25 mạng đích trong mỗi AS

RIP: ví dụ



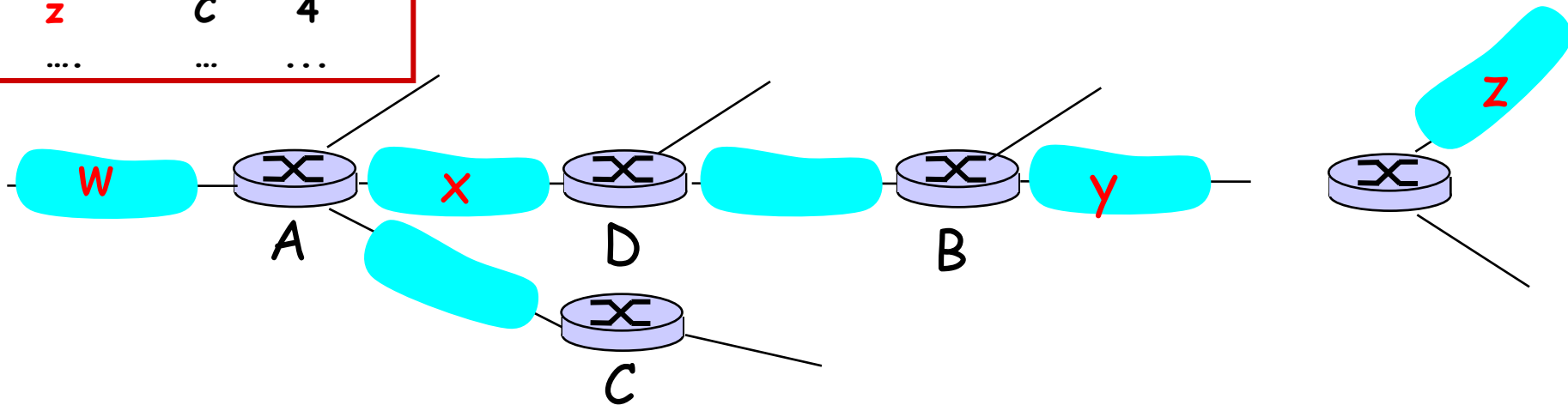
Network đích	Router kế tiếp	Số hop đến đích
w	A	2
y	B	2
z	B	7
x	--	1
....

bảng Routing trong D

RIP: ví dụ

Đích	Kế tiếp	Số hop
w	-	1
x	-	1
z	C	4
...

thông báo
từ A đến D



Network đích	Router kế tiếp	Số hop đến đích
w	A	2
y	B	2
z	B A	7 5
x	--	1
...

bảng Routing trong D

Lớp Network

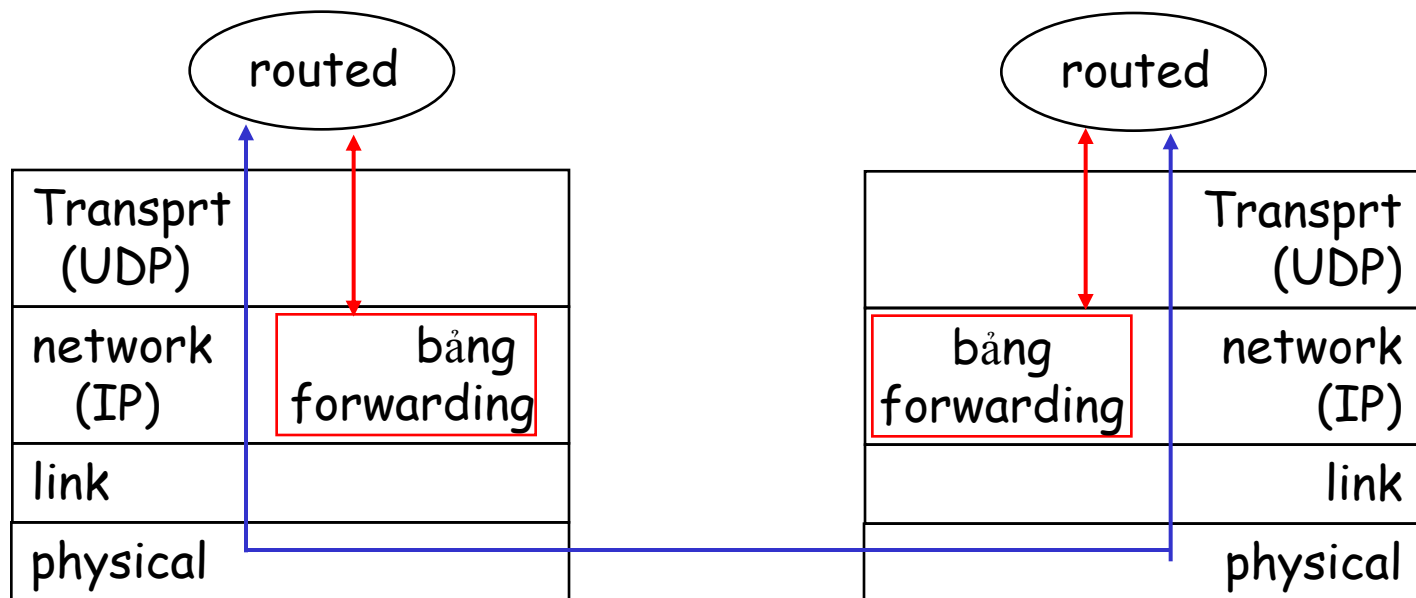
RIP: kết nối sai & phục hồi

Nếu không có thông báo nào sau 180s → lân cận/kết nối được xem như đã "chết"

- những đường đi qua lân cận không còn dùng được
- gửi thông báo mới cho các lân cận
- các lân cận tiếp tục gửi ra những thông báo mới đó (nếu các bảng thay đổi)
- thông tin kết nối lỗi nhanh chóng (?) lan truyền trên toàn mạng
- poison reverse dùng để ngăn chặn các vòng lặp ping-pong (khoảng cách vô hạn = 16 hop)

RIP: xử lý bảng

- ❑ các bảng RIP routing được quản lý bởi các tiến trình mức **application** gọi là route-d (daemon)
- ❑ các thông báo gửi trong các gói UDP, lặp lại theo chu kỳ



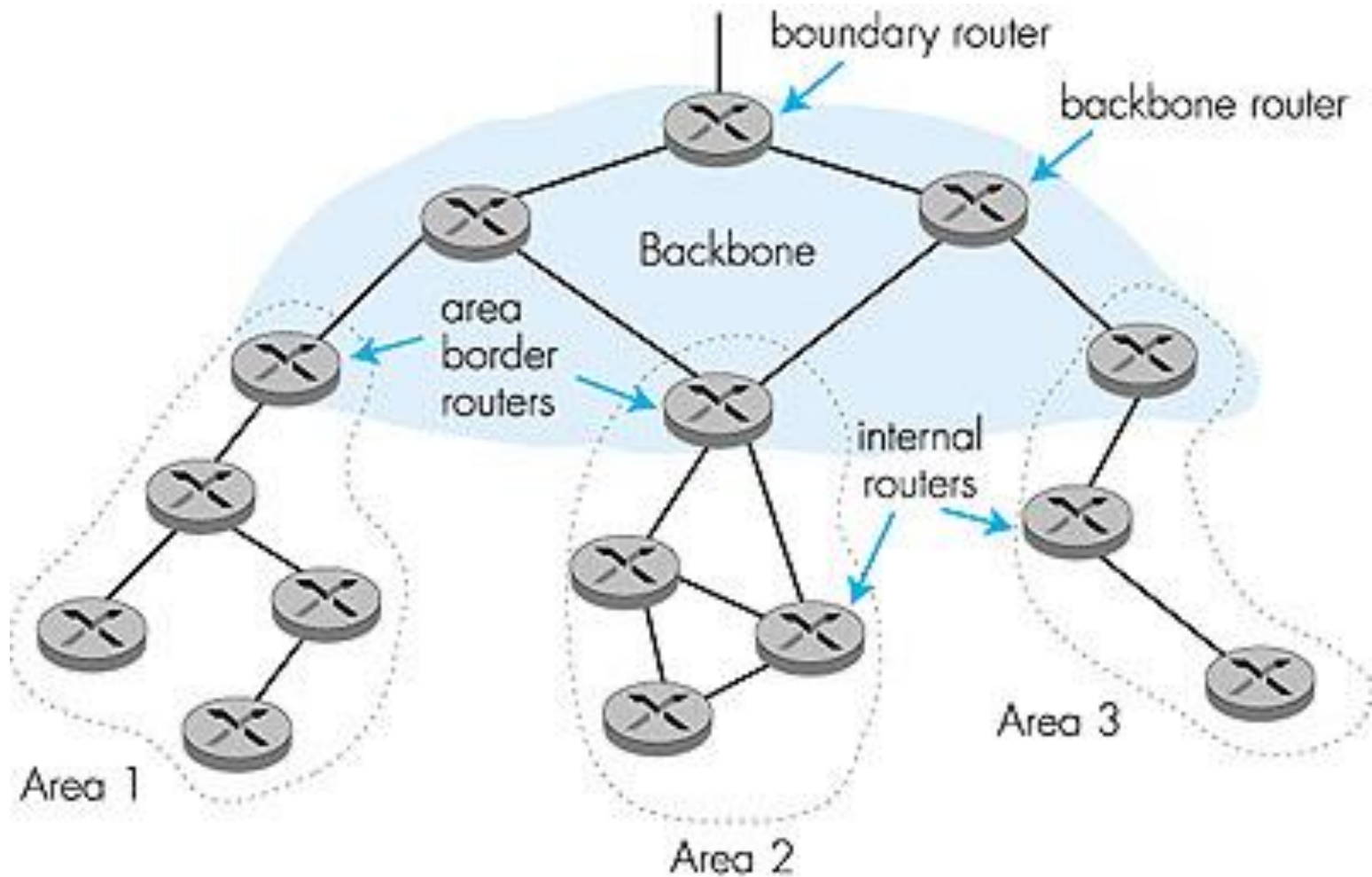
OSPF (Open Shortest Path First)

- ❑ “open”: 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 1 entry vào mỗi router lân cận
- ❑ các thông báo phân tán đến **toàn bộ** AS (thông qua cơ chế flooding)
 - thông điệp OSPF trực tiếp trên IP (chứ không phải là TCP hoặc UDP)

các đặc tính OSPF "cao cấp" (không có trong RIP)

- ❑ **bảo mật**: chứng thực tất cả các thông điệp OSPF (ngăn những kẻ có ý đồ xấu)
- ❑ cho phép **nhiều** đường đi có chi phí giống nhau (RIP chỉ cho 1)
- ❑ với mỗi kết nối, có nhiều không gian chi phí cho **TOS** khác nhau (vd: chi phí kết nối vệ tinh được thiết lập "thấp" để đạt hiệu quả tốt, "cao" cho thời gian thực)
- ❑ hỗ trợ **uni-** và **multicast** tích hợp:
 - Multicast OSPF (MOSPF) dùng cùng cơ sở dữ liệu cấu trúc như OSPF
- ❑ OSPF **phân cấp** trong những miền lớn.

OSPF phân cấp



OSPF phân cấp

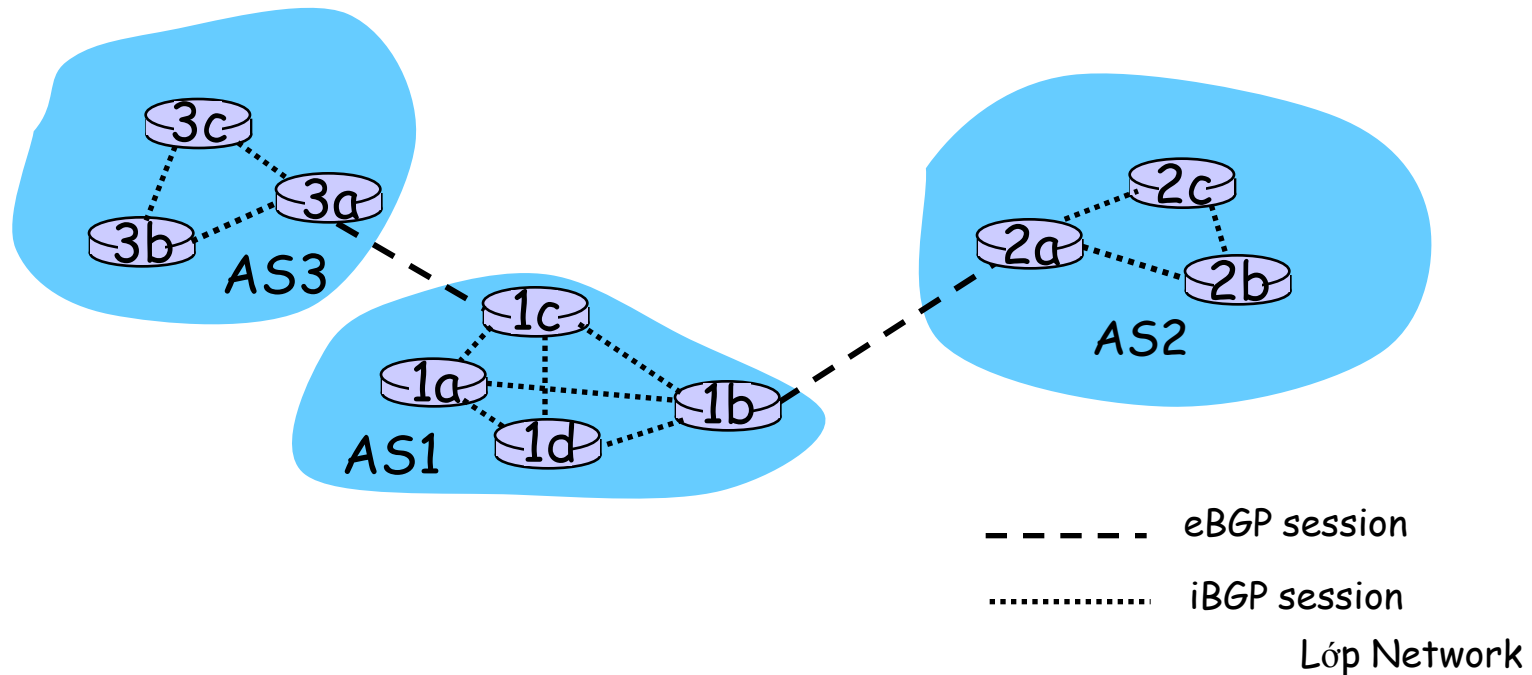
- ❑ **phân cấp mức 2:** vùng địa phương, backbone.
 - các thông báo Link-state chỉ bên trong vùng
 - mỗi nút có cấu trúc vùng chi tiết; chỉ biết hướng (đường đi ngắn nhất) đến các mạng trong các vùng khác
- ❑ **các router ngoài biên vùng:** “tổng hợp” khoảng cách đến các mạng trong vùng của nó, thông báo đến các router ngoài biên vùng
- ❑ **các Backbone routers:** chạy OSPF routing hạn chế đến backbone.
- ❑ **các router ngoài biên:** kết nối đến các AS khác.

Internet inter-AS routing: BGP

- ❑ **BGP (Border Gateway Protocol):** *chuẩn thực tế*
- ❑ BGP hỗ trợ cho mỗi AS:
 1. Lấy thông tin khả năng chạm subnet đích từ các AS lân cận.
 2. lan truyền thông tin đó đến tất cả các router bên trong AS.
 3. Xác định đường đi "tốt" đến các subnet dựa trên thông tin khả năng chạm subnet đích và chính sách.
- ❑ cho phép subnet thông báo sự tồn tại của nó trên Internet

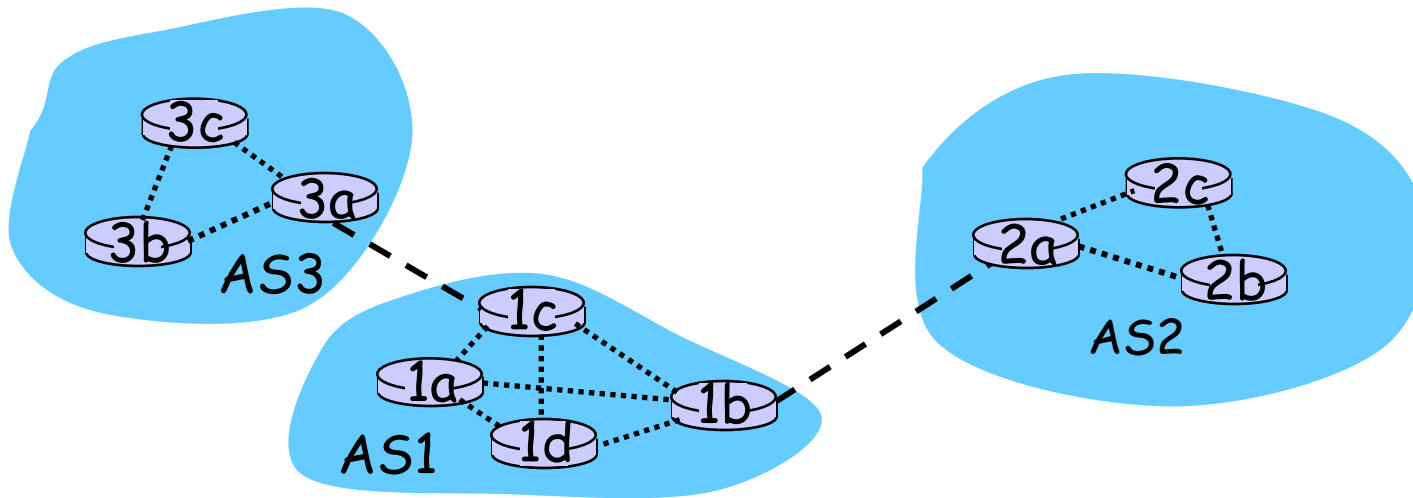
các cơ sở BGP

- ❑ Các cặp router (BGP peers) trao đổi thông tin routing trên các kết nối TCP bán bền vững: **BGP sessions**
 - các phiên BGP không cần các kết nối vật lý tương xứng
- ❑ Khi AS2 thông báo 1 prefix đến AS1, AS2:
 - AS2 có thể tích hợp các prefix trong thông báo của nó



Phân phối thông tin khả chạm đích

- ❑ Với phiên eBGP giữa 3a và 1c, AS3 gửi thông tin khả chạm cho AS1.
- ❑ 1c sau đó có thể dùng iBGP phân phối thông tin khả chạm đến tất cả router trong AS1.
- ❑ 1b sau đó có thể thông báo lại thông tin khả chạm đến AS2 trên phiên eBGP từ 1b-đến-2a
- ❑ Khi router học xong prefix mới, tạo entry cho prefix trong bảng forwarding của nó.



----- eBGP session

..... iBGP session

các thuộc tính đường đi & BGP

- ❑ khi thông báo 1 prefix, trong đó có chứa các thuộc tính BGP
 - prefix + các thuộc tính = "dẫn đường"
- ❑ 2 thuộc tính quan trọng:
 - **AS-PATH**: chứa các AS qua đó thông báo prefix truyền đi: AS 67 AS 17
 - **NEXT-HOP**: Chỉ định router bên trong AS là hop kế tiếp (có nhiều kết nối từ AS hiện tại đến AS hop kế tiếp)
- ❑ Khi gateway router nhận thông báo tìm đường, nó dùng **import policy** để chấp nhận/từ chối.

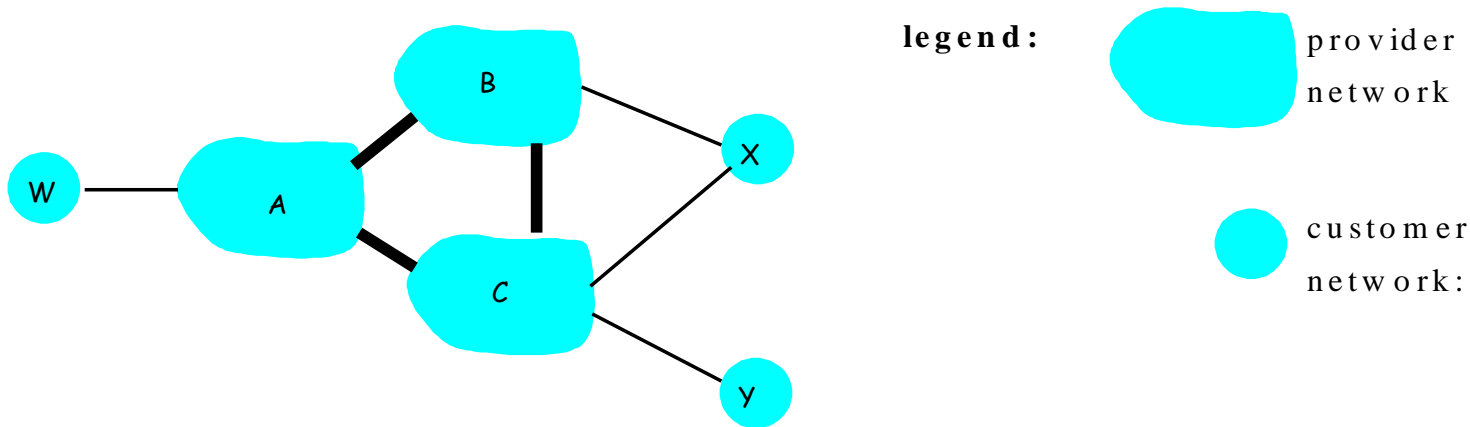
chọn BGP route

- ❑ Router có thể học nhiều hơn một nhưng chỉ được chọn một.
- ❑ Các quy tắc hạn chế:
 1. Thuộc tính giá trị ưu tiên cục bộ: quyết định chính sách
 2. AS-PATH ngắn nhất
 3. NEXT-HOP router gần nhất: hot potato routing
 4. Tiêu chuẩn bổ sung

các thông điệp BGP

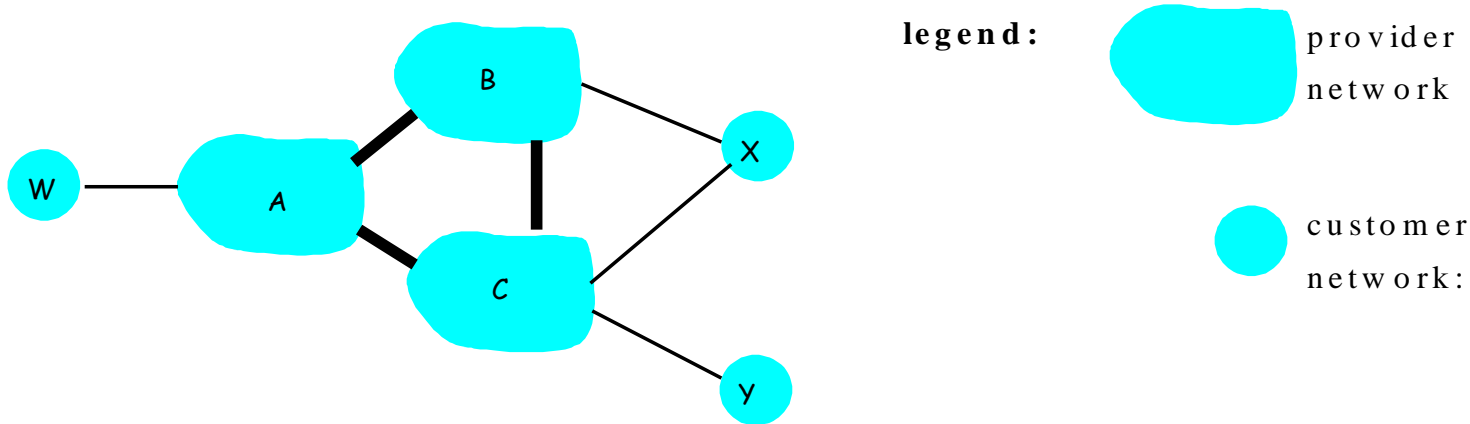
- ❑ trao đổi các thông điệp BGP dùng TCP
- ❑ các thông điệp BGP:
 - **OPEN**: mở kết nối TCP đến peer và chứng thực người gửi
 - **UPDATE**: thông báo đường đi mới
 - **KEEPALIVE** giữ kết nối sống (alive), cũng gọi là yêu cầu OPEN các ACK
 - **NOTIFICATION**: thông báo các lỗi trong thông điệp trước đó, dùng để đóng kết nối

chính sách BGP routing



- A,B,C là **những nhà cung cấp mạng**
- X,W,Y là khách hàng (của những nhà cung cấp mạng)
- X là **dual-homed**: gắn vào 2 mạng
 - X không muốn dẫn đường từ B qua X đến C
 - .. vì thế X sẽ không thông báo với B về đường đến C

chính sách BGP routing (2)



- ❑ A thông báo với B về đường AW
- ❑ B thông báo với X về đường BAW
- ❑ B sẽ thông báo với C về đường BAW?
 - Không có cách nào! B không có "lợi ích" về đường CBAW cũng như W không phải là những khách hàng của B, C
 - B muốn buộc C phải tìm đường đến W thông qua AB
 - B chỉ muốn tìm đường đến/từ khách hàng của nó!

Tại sao phải routing Intra- và Inter-AS khác nhau?

chính sách:

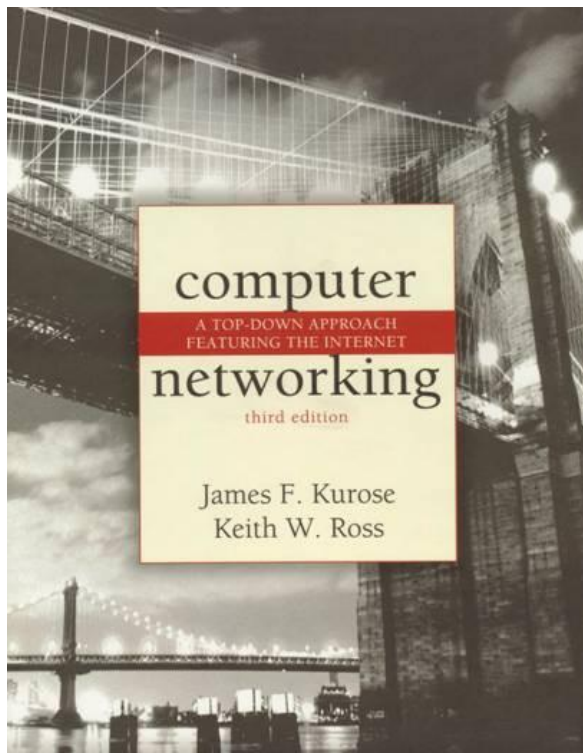
- ❑ Inter-AS: người quản trị muốn điều hành hoạt động lưu thông routing, ai routing thông qua mạng của họ
- ❑ Intra-AS: 1 người quản trị, vì thế không cần các quyết định chính sách

linh hoạt:

- ❑ routing phân cấp tiết giảm kích thước bảng, giảm lưu lượng cập nhật

hiệu suất:

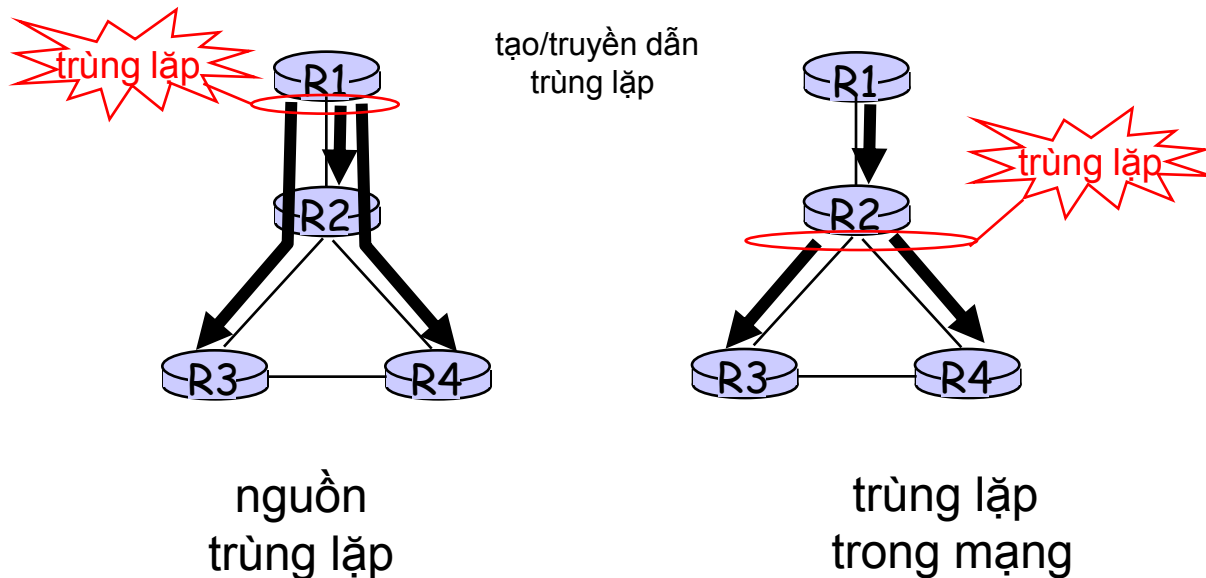
- ❑ Intra-AS: có thể tập trung vào hiệu suất
- ❑ Inter-AS: chính sách quan trọng hơn hiệu suất



4.7 Broadcast và multicast routing

Broadcast Routing

- ❑ Chuyển các gói từ nguồn đến tất cả các nút khác
- ❑ Nguồn trùng lặp thì không có hiệu quả



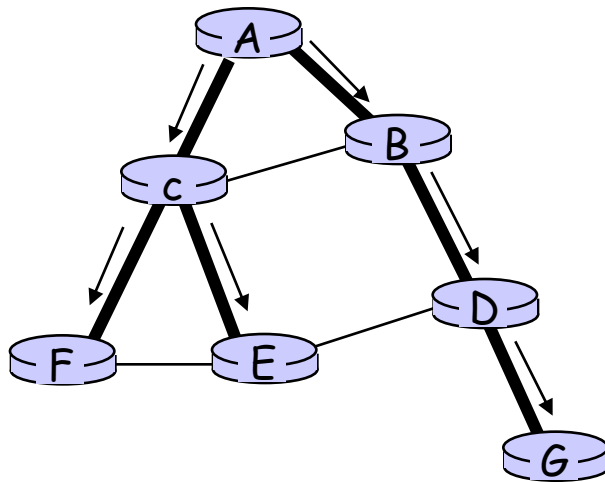
- ❑ Nguồn trùng lặp: làm sao xác định địa chỉ người nhận?

Trùng lặp trong mạng

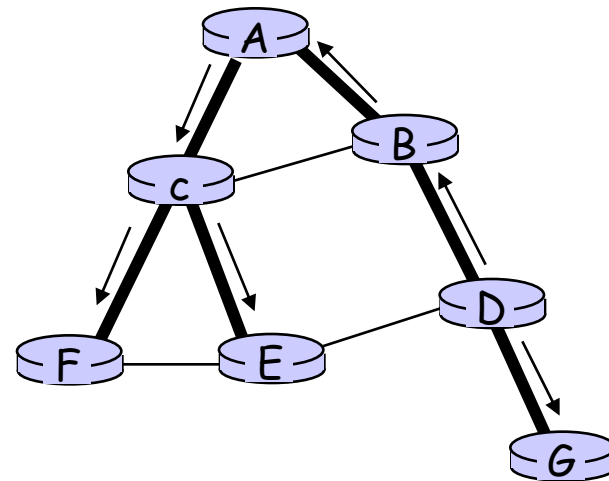
- ❑ Ngập lụt: khi nút nhận gói broadcast và gửi đến tất cả các lân cận
 - các vấn đề: bão broadcast & lặp lại
- ❑ Ngập lụt có điều khiển: nút chỉ gửi broadcast nếu nó không gửi gói nào giống như vậy trước đó
 - nút phải theo dõi các gói đã broadcast
 - hoặc reverse path forwarding (RPF): chỉ forward gói nếu nó đến trên đường ngắn nhất giữa nút và nguồn
- ❑ Cây mở rộng
 - tại bất kỳ nút nào cũng đều không nhận thừa các gói

Cây mở rộng

- ❑ Đầu tiên xây dựng Cây mở rộng
- ❑ Các nút forward các bản sao chỉ trên Cây mở rộng



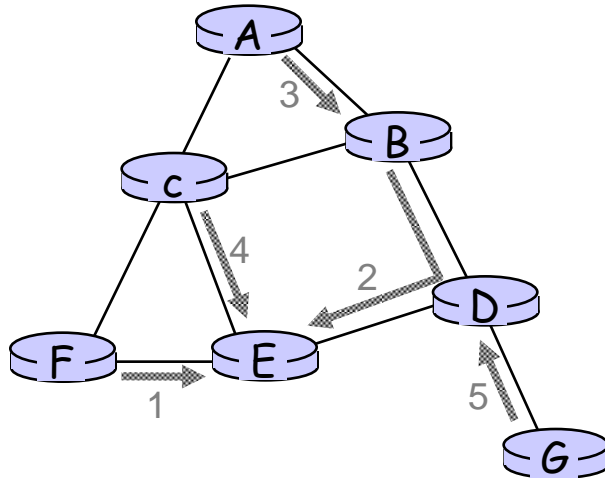
(a) Broadcast khởi đầu tại A



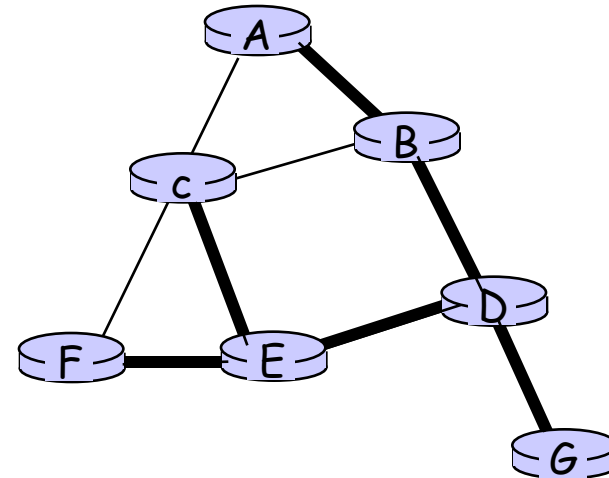
(b) Broadcast khởi đầu tại D

Cây mở rộng: tạo

- ❑ Nút trung tâm
- ❑ Mỗi nút gửi thông điệp gia nhập unicast đến nút trung tâm
 - Thông điệp forward cho đến khi gặp một nút đã nằm trên cây mở rộng



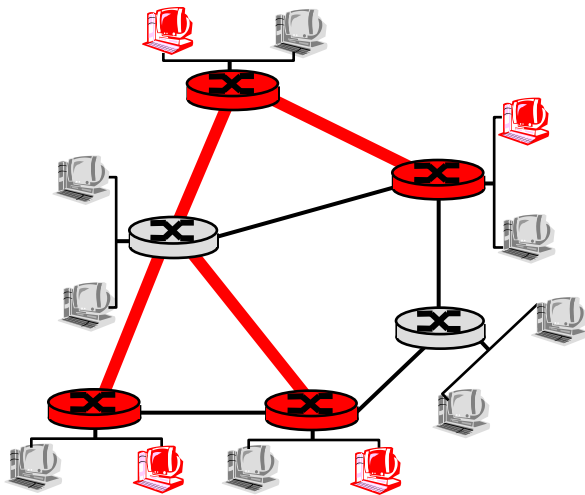
(a) các bước xây dựng một cây mở rộng



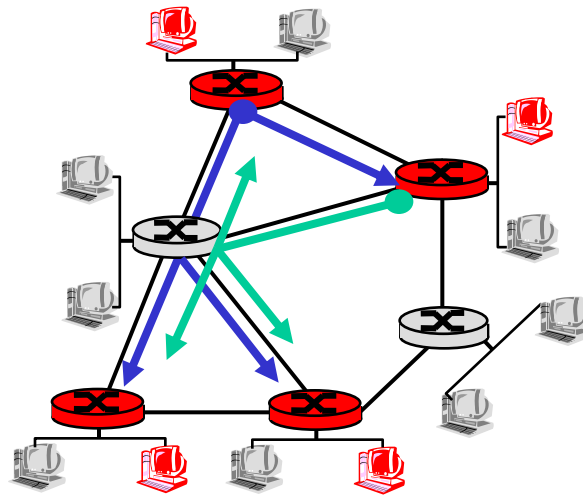
(b) cây mở rộng đã xây dựng xong

Multicast Routing: phát biểu vấn đề

- **Mục tiêu:** tìm một cây (trong các cây) kết nối các router có các thành viên nhóm Multicast
 - **cây:** không phải tất cả các đường đi giữa các router được dùng
 - **cây dựa trên nguồn:** cây khác nhau từ nơi gửi đến nơi nhận
 - **cây chia sẻ:** cây giống nhau dùng bởi tất cả các thành viên nhóm



cây chia sẻ



cây dựa trên nguồn

Các cách tiếp cận xây dựng các cây multicast

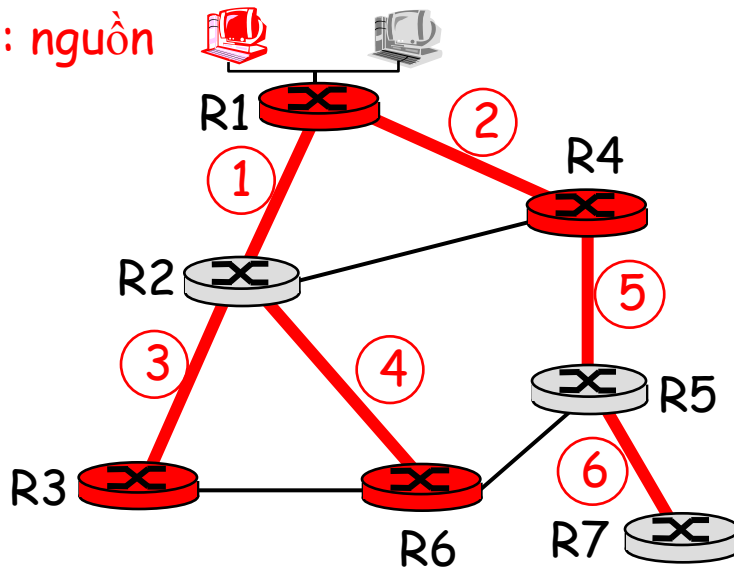
- ❑ **cây dựa trên nguồn:** một cây mỗi nguồn
 - các cây đường đi ngắn nhất
 - cây đường đi ngược
- ❑ **cây chia sẻ nhóm:** nhóm dùng 1 cây
 - mở rộng tối thiểu (Steiner)
 - các cây trung tâm

...nghiên cứu các cách tiếp cận cơ bản

Cây đường đi ngắn nhất

- cây đường đi ngược multicast: cây đường đi ngắn nhất dẫn đường từ nguồn đến tất cả các điểm nhận
 - giải thuật Dijkstra

S: nguồn



Ký hiệu



router với các thành viên của nhóm đã gắn vào



router không có các thành viên của nhóm gắn vào



kết nối dùng cho forward, i chỉ thứ tự kết nối thêm vào bởi giải thuật

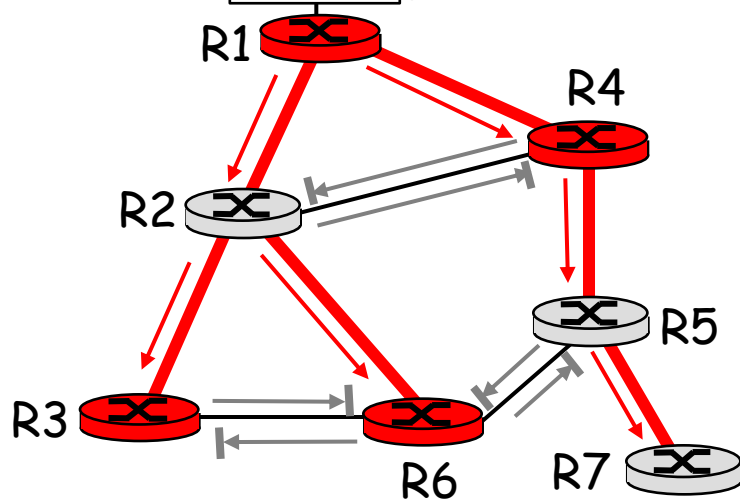
Cây đường đi ngược forward:

- ❑ phụ thuộc tri thức của router của đường đi ngắn nhất unicast từ nó đến nơi gửi
- ❑ mỗi router có cách xử lý forwarding đơn giản

if (multicast datagram nhận được trên kết nối đến trên đường đi ngắn nhất kể từ trung tâm)
then tràn ngập datagram lên tất cả các kết nối ra
else lơ đi datagram

Cây đường đi ngược forward: ví dụ

S: nguồn



Ký hiệu



router với các thành viên của nhóm đã gắn vào



router không có các thành viên của nhóm gắn vào



datagram sẽ được forward



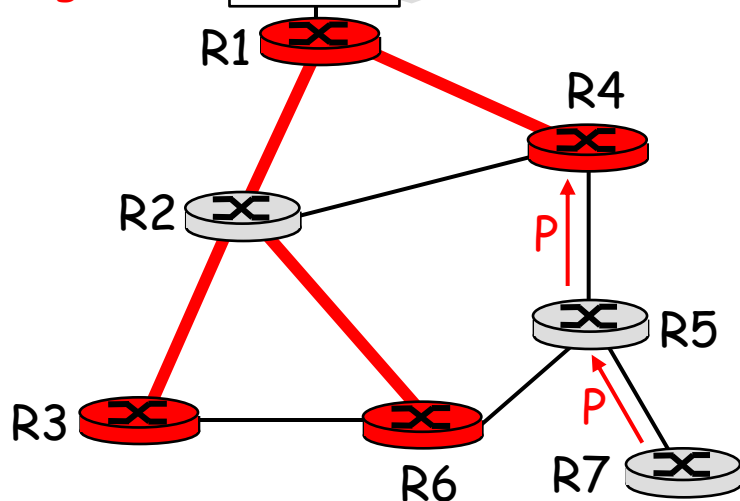
datagram sẽ không được forward

- kết quả là một cây SPT *đảo ngược*
 - có thể là một lựa chọn tồi với các kết nối không đồng bộ

Cây đường đi ngược forward: cắt giảm

- cây forward chứa các cây con với các thành viên nhóm không multicast
 - không cần forward các datagram xuống cây con
 - "cắt giảm" các thông điệp gửi lên bởi router

S: nguồn



Ký hiệu



router với các thành viên của nhóm đã gắn vào



router không có các thành viên của nhóm gắn vào



thông điệp cắt giảm



các kết nối với multicast forward

Cây chia sẻ: cây Steiner

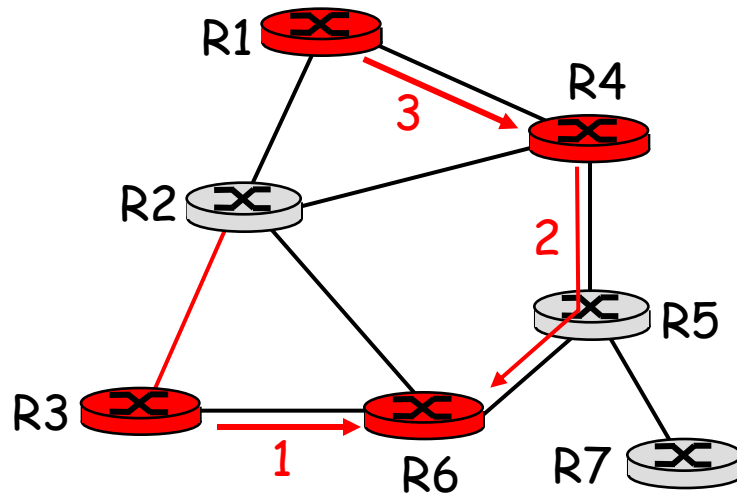
- ❑ **cây Steiner**: cây chi phí thấp nhất kết nối tất cả các router với các thành viên nhóm đã gắn vào
- ❑ vấn đề là NP-complete
- ❑ đã có các heuristic rất tốt
- ❑ không dùng trong thực tế:
 - độ phức tạp tính toán
 - cần thông tin về toàn bộ mạng
 - monolithic: chạy lại bất cứ khi nào 1 router cần gia nhập/rời khỏi

Các cây trung tâm




- ❑ một cây truyền nhận chia sẻ cho tất cả
- ❑ 1 router được gọi là "*trung tâm*" của cây
- ❑ để gia nhập:
 - bên ngoài gửi *thông điệp gia nhập* unicast cho router trung tâm
 - *thông điệp gia nhập* "được xử lý" bởi các router trung gian và chuyển đến router trung tâm
 - *thông điệp gia nhập* gặp nhánh của cây đã tồn tại hoặc đến được trung tâm
 - đường đi thu được khi *thông điệp gia nhập* đến trở thành nhánh mới của cây cho router này

Các cây trung tâm: ví dụ

giả sử R6 được chọn làm trung tâm:



Ký hiệu

-  router với các thành viên của nhóm đã gắn vào
-  router không có các thành viên của nhóm gắn vào
-  thứ tự đường đi trong ấy các thông điệp gia nhập đã sinh ra

Internet Multicasting Routing: DVMRP

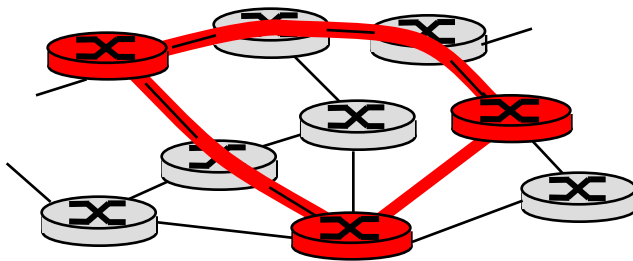
- ❑ **DVMRP**: giao thức multicast routing dùng vector khoảng cách, RFC1075
- ❑ ***flood & prune***: forward đường đi ngược, cây dựa trên nguồn
 - cây RPF dựa trên các bảng routing của DVMRP của riêng nó được xây dựng bởi truyền thông các router DVMRP
 - không có các giả thiết về unicast bên dưới
 - datagram ban đầu đến nhóm multicast làm tràn ngập mọi nơi thông qua RPF
 - các router không phải nhóm: gửi các thông điệp cắt giảm

DVMRP: (++)...

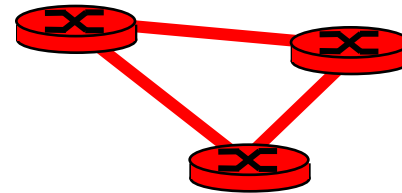
- ❑ trạng thái "mềm": router DVMRP chu kỳ (1 phút) "quên" các nhánh đã cắt giảm:
 - dữ liệu mcast một lần nữa đổ xuống các nhánh không cắt giảm
 - router dòng xuống: tái cắt giảm hoặc tiếp tục nhận dữ liệu
- ❑ các router có thể nhanh chóng tái cắt giảm
 - gia nhập IGMP tại các nút lá
- ❑ còn lại
 - đã hiện thực phổ biến trong các router thương mại
 - hoàn thành routing dùng DVMRP

Tunneling

Hỏi: Làm sao kết nối các "đảo" multicast router trong một "biển" các unicast router?



cấu trúc vật lý



cấu trúc logic

- ❑ multicast datagram được đóng gói trong datagram "thông thường" (không có multicast)
- ❑ datagram IP thông thường gửi thông qua "đường ống" và qua IP unicast đến router multicast nhận
- ❑ router multicast nhận mở gói để lấy multicast datagram

PIM: Protocol Independent Multicast

- ❑ không phụ thuộc vào bất kỳ giải thuật unicast routing bên dưới nào
- ❑ hai kịch bản phân phối multicast khác nhau

Trù mật:

- ❑ các thành viên nhóm đóng gói trữ mật
- ❑ bảng thông dư thừa

Thưa thớt:

- ❑ số lượng các mạng với các thành viên nhóm ít
- ❑ các thành viên nhóm "phân bố thưa thớt"
- ❑ bảng thông không dư thừa

Hậu quả sự phân chia thừa thớt-trù mật

Trù mật:

- ❑ nhóm các thành viên router là giả cho đến khi các router cắt giảm thực sự
- ❑ *kiến trúc hướng dữ liệu* trên cây multicast (vd: RPF)
- ❑ bảng thông và router không thuộc nhóm xử lý *phung phí*

Thừa thớt:

- ❑ không có thành viên cho đến khi có các router gia nhập thực sự
- ❑ *kiến trúc hướng người nhận* trên cây multicast (vd: cây trung tâm)
- ❑ bảng thông và router không thuộc nhóm xử lý *vừa phải*

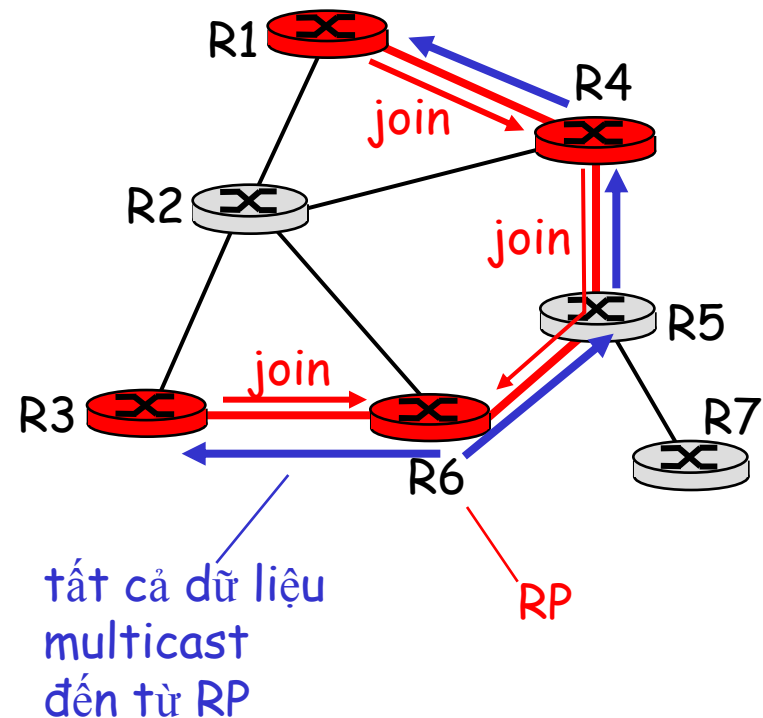
PIM- kiểu trù mật

flood-and-prune RPF, tương tự DVMRP,
nhưng:

- ❑ giao thức unicast bên dưới cung cấp thông tin RPF cho datagram đến
- ❑ ít phức tạp (ít hiệu quả)
- ❑ có cơ chế giao thức cho router để kiểm tra có phải router là nút lá

PIM - kiểu thưa thớt

- ❑ tiếp cận hướng trung tâm
- ❑ router gửi thông điệp gia nhập đến rendezvous point (RP)
 - các router trung gian cập nhật trạng thái và forward thông điệp *gia nhập*
- ❑ sau khi gia nhập bằng RP, router có thể chuyển sang cây xác định nguồn
 - hiệu suất tăng: ít tập trung, các đường đi ngắn



PIM - kiểu thừa thớt

bên gửi:

- ❑ dữ liệu unicast đến RP, RP phân phối xuống cây có nút gốc là RP
- ❑ RP có thể mở rộng cây multicast dòng lên đến nguồn
- ❑ RP có thể gửi thông điệp *dừng* nếu không có bên nhận nào được gắn vào
 - "không có ai đang lắng nghe!"

