



NHẬP MÔN MẠNG MÁY TÍNH

Chương 4: Tầng Network

Chương 4: Tầng Network



A note on the use of these PowerPoint slides:

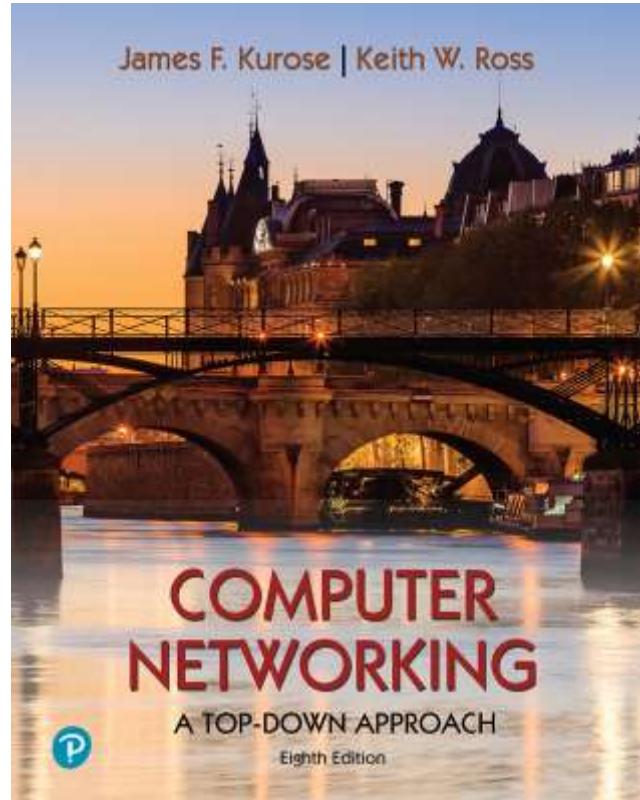
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use,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

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2020
J.F Kurose and K.W. Ross, All Rights Reserved



**Computer Networking:
A Top-Down Approach**
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Chương 4: Tầng Network (tầng Mạng)



Mục tiêu:

- Hiểu các nguyên lý nền tảng của các dịch vụ tầng Network
 - Các mô hình dịch vụ của tầng Network
 - Forwarding và routing
 - Cách router hoạt động
 - Routing
 - Broadcast, multicast
- Hiện thực trong Internet
 - Giao thức IP
 - NAT
 - SDN



Chương 4: Nội dung

4.1 Tổng quan

- Mức dữ liệu
- Mức điều khiển

4.2 Virtual circuit network và datagram network

- Các mạch ảo
- Mạng datagram

4.3 Cấu trúc bên trong router

- Cổng đầu vào, chuyển đổi, cổng đầu ra
- Quản lý bộ đệm, lập kế hoạch

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing

- DHCP
- NAT
- ICMP
- IPV6

4.5 Các thuật toán routing

- Link state
- Distance vector

4.6 Routing trong Internet

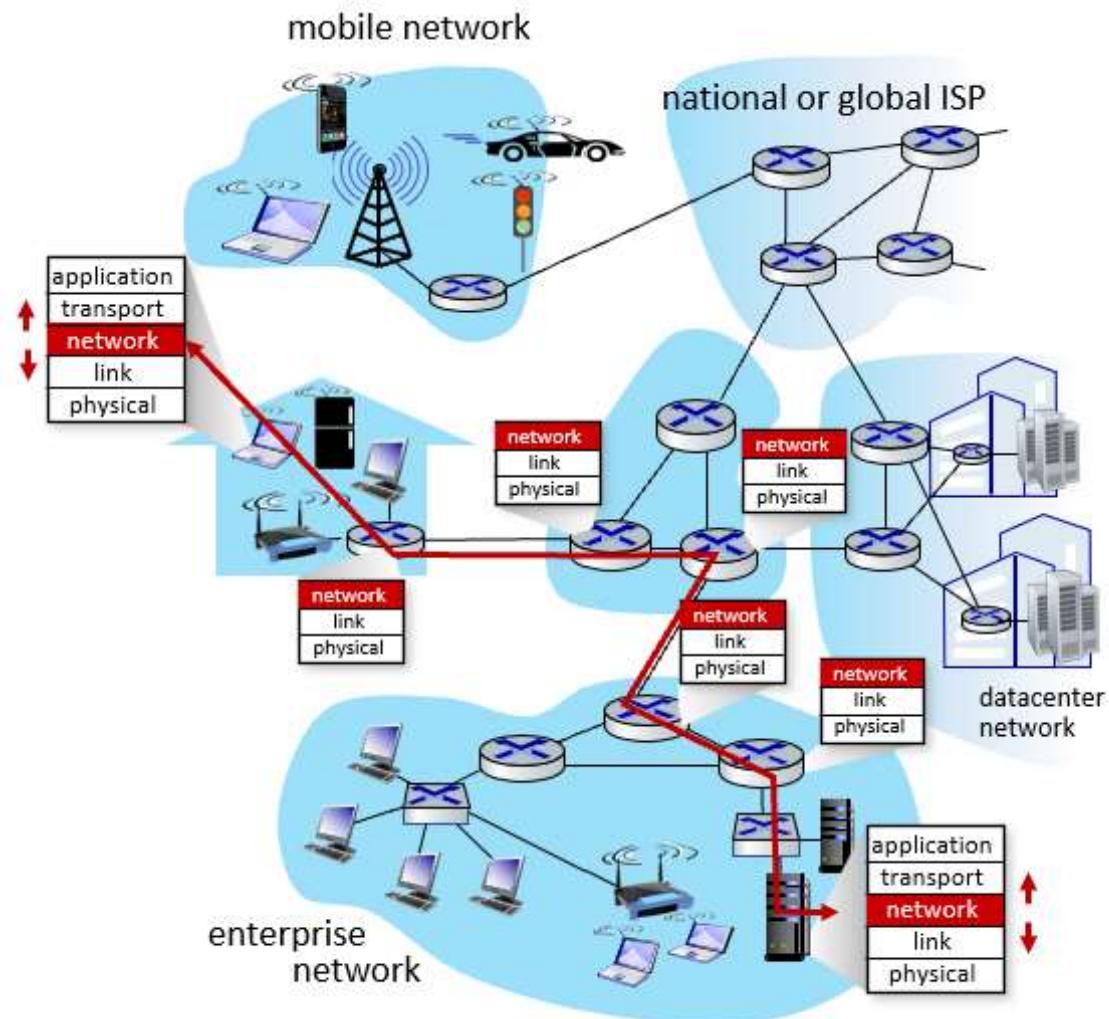
- Hierarchical routing
- RIP, OSPF, BGP
- SDN
- Quản lý mạng

4.7 Broadcast và Multicast routing



Các dịch vụ và giao thức tầng Network

- Truyền các segment từ máy gửi đến máy nhận
 - **Máy gửi:** đóng gói các segment thành các **datagram**, chuyển đến tầng liên kết
 - **Máy nhận:** phân phối các segment đến giao thức tầng vận chuyển
- Giao thức tầng network trong *thiết bị Internet*: máy tính, router
- **Router:**
 - Kiểm tra các trường trong phần header của tất cả các datagram đi qua nó
 - Chuyển các datagram từ cổng đầu vào sang cổng đầu ra để truyền datagram dọc theo đường dẫn end-end





Hai chức năng chính của tầng network

Chức năng tầng network:

- *Forwarding (chuyển tiếp)*: di chuyển các gói dữ liệu từ cổng đầu vào của router đến cổng đầu ra phù hợp của router
- *Routing (định tuyến)*: xác định tuyến đường di chuyển cho các gói dữ liệu từ nguồn đến đích
 - Các thuật toán định tuyến

Tương tự: tham gia một chuyến đi

- *Chuyển tiếp*: quá trình chuyển qua một nhánh đường xác định
- *Định tuyến*: quá trình lập kế hoạch chuyến đi từ nguồn đến đích



chuyển tiếp



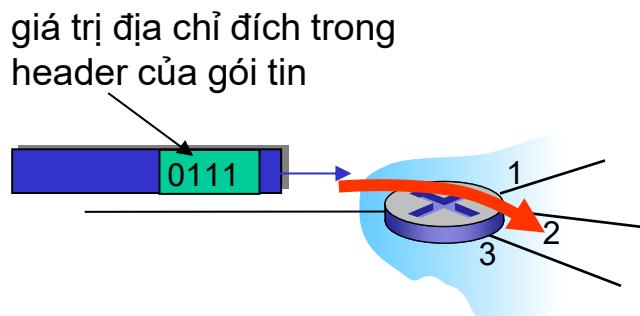
định tuyến (lộ trình)

Tầng network: mức dữ liệu (data plane) và mức điều khiển (control plane)



Mức dữ liệu:

- Cục bộ, trên mỗi router
- Xác định cách datagram đến cổng đầu vào của router được chuyển tiếp đến cổng đầu ra của router



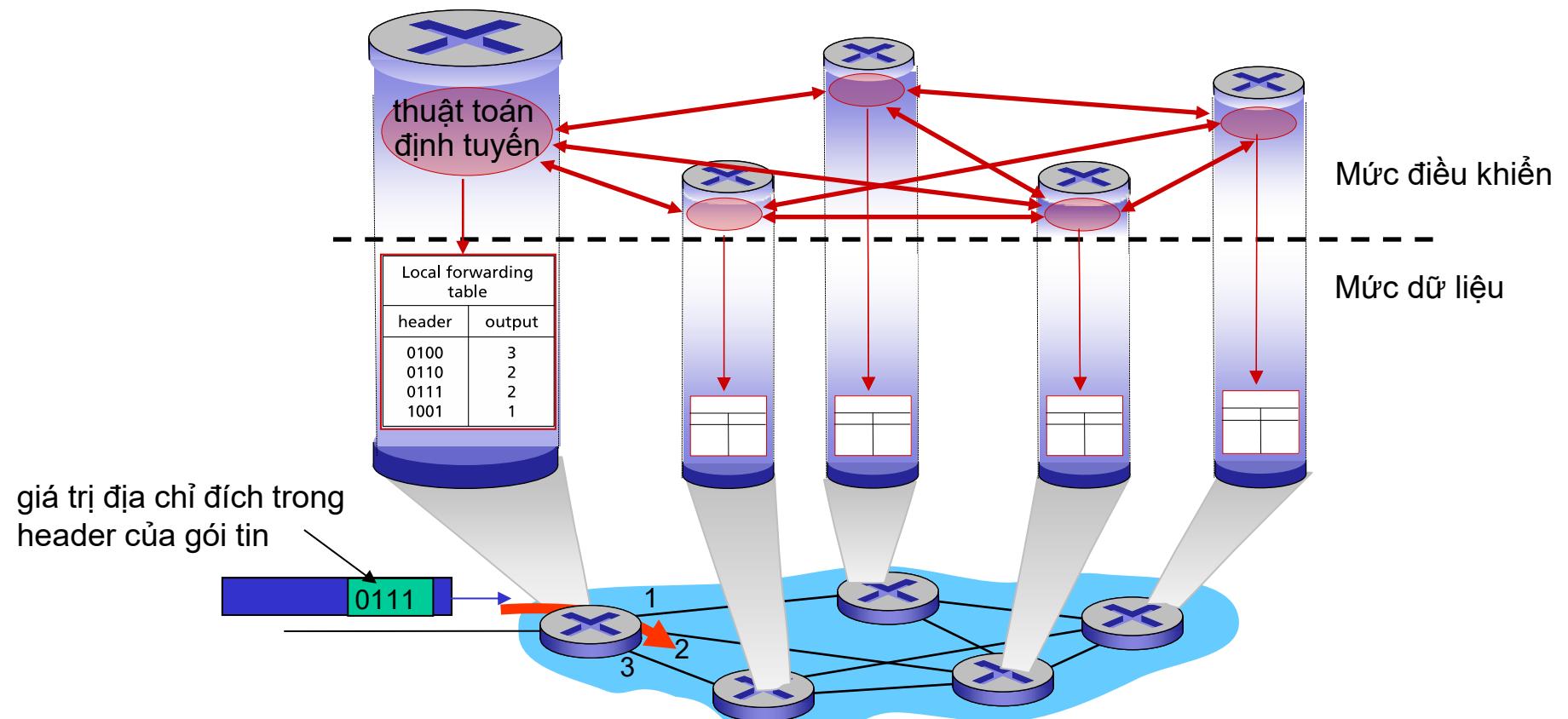
Mức điều khiển:

- Toàn mạng, logic
- Xác định cách datagram được định tuyến giữa các router dọc theo đường dẫn đầu cuối từ máy nguồn đến máy đích
- Hai cách tiếp cận mức điều khiển:
 - *Thuật toán định tuyến truyền thống*: được thực hiện trong router
 - *Mạng được xác định bằng phần mềm (SDN)*: được triển khai trong các máy chủ (từ xa)

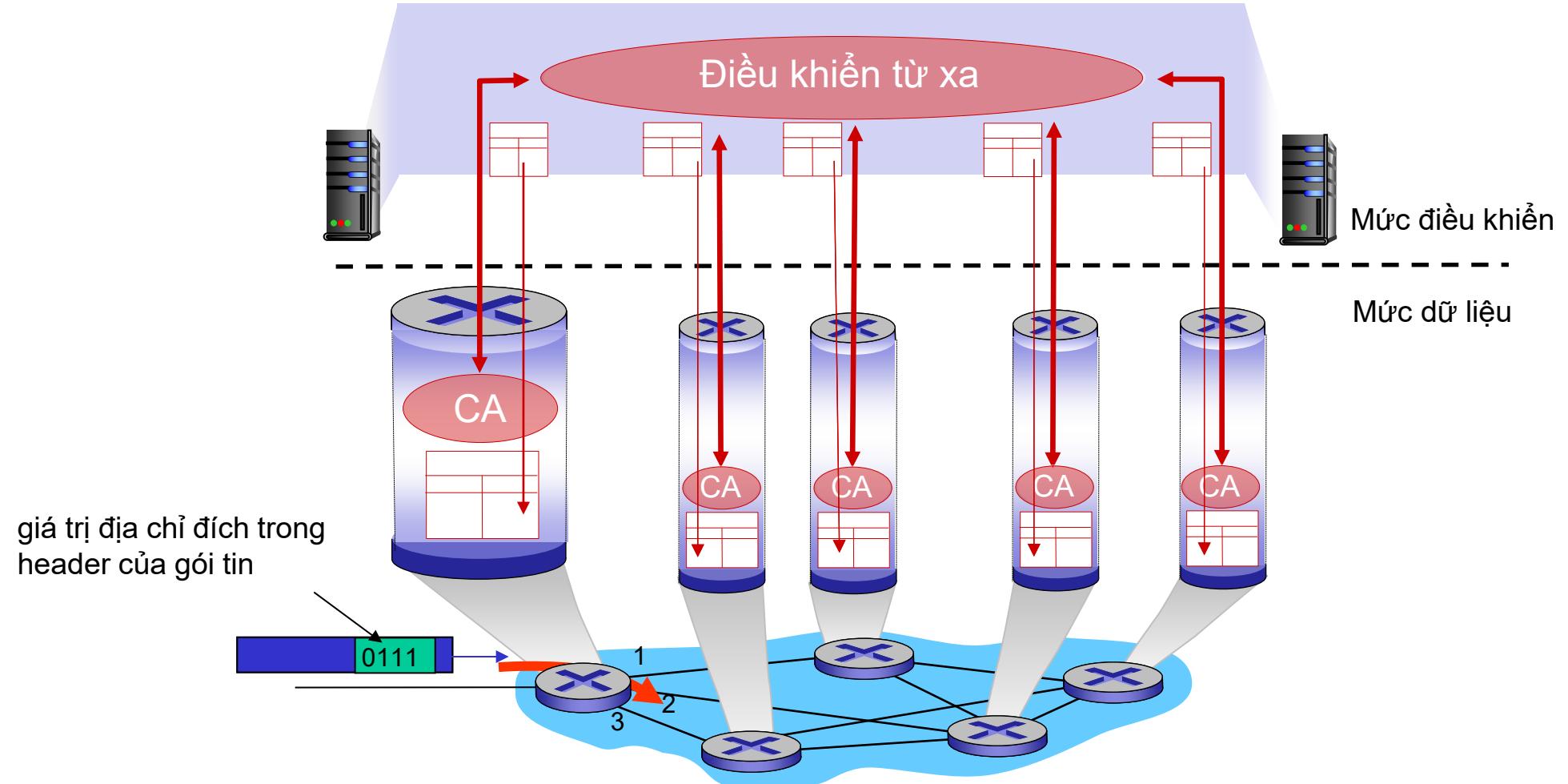
Mức điều khiển ở mỗi router



Các thành phần thuật toán định tuyến riêng lẻ *trong mỗi và mọi router* tương tác trong mức điều khiển



Mức điều khiển được xác định bằng phần mềm (SDN – Software-Defined-Networking)





Mô hình dịch vụ mạng

Các dịch vụ cho datagram *riêng lẻ*:

- Đảm bảo truyền
- Độ trễ dưới 40 mili giây

Các dịch vụ cho một *luồng* datagram:

- Truyền datagram theo thứ tự
- Đảm bảo băng thông tối thiểu để truyền
- Hạn chế các thay đổi trong khoảng trống giữa các gói tin

Hỏi: Mô hình dịch vụ nào
cho “kênh” truyền datagram
từ máy gửi đến máy nhận?



Mô hình dịch vụ tầng mạng

Kiến trúc Mạng	Mô hình dịch vụ	Đảm bảo chất lượng dịch vụ (QoS)?			
		Băng thông	Mất mát	Thứ tự	thời gian
Internet	nỗ lực tốt nhất	không có	KHÔNG	KHÔNG	KHÔNG

Mô hình dịch vụ Internet “nỗ lực tốt nhất”

Không đảm bảo về :

- i. Truyền thành công datagram tới đích
- ii. Thời gian hoặc thứ tự truyền
- iii. Băng thông sẵn có cho luồng đầu cuối



Mô hình dịch vụ tầng mạng

Kiến trúc Mạng	Mô hình Dịch vụ	Đảm bảo chất lượng dịch vụ (QoS)?			
		Băng thông	Mất mát	Thứ tự	Thời gian
Internet	nỗ lực tốt nhất	không có	không	không	không
ATM	tốc độ không đổi	tỷ lệ không đổi	có	có	có
ATM	tốc độ bit khả dụng	đảm bảo tối thiểu	không	có	không
Internet	đảm bảo dịch vụ (RFC 1633)	có	có	có	có
Internet	phân tán (RFC 2475)	khả thi	khả thi	khả thi	có



Chương 4: Nội dung

4.1 Tổng quan

- Mức dữ liệu
- Mức điều khiển

4.2 Virtual circuit network và datagram network

- Các mạch ảo
- Mạng datagram

4.3 Cấu trúc bên trong router

- Cổng đầu vào, chuyển đổi, cổng đầu ra
- Quản lý bộ đệm, lập kế hoạch

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing

- DHCP
- NAT
- ICMP
- IPV6

4.5 Các thuật toán routing

- Link state
- Distance vector

4.6 Routing trong Internet

- Hierarchical routing
- RIP, OSPF, BGP
- SDN
- Quản lý mạng

4.7 Broadcast và Multicast routing

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

- Mạng *datagram* cung cấp dịch vụ *connectionless* tại tầng network
- Mạng *mạch ảo (virtual-circuit network)* cung cấp dịch vụ *connection* tại tầng network
- Tương tự như các dịch vụ kết nối định hướng và không định hướng của tầng transport, nhưng:
 - *Dịch vụ:* từ host này đến host kia (host-to-host)
 - *Không lựa chọn:* network chỉ cung cấp 1 dịch vụ
 - *Thực hiện:* trong mạng lõi



Các mạch ảo (Virtual circuits)

- Thiết lập cuộc gọi, chia nhỏ mỗi cuộc gọi trước khi dữ liệu có thể truyền
- Mỗi packet mang định 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 mà gói đi qua.
- Đường link, các tài nguyên router (băng thông, bộ nhớ đệm) có thể được cấp phát cho kết nối ảo (các tài nguyên dành riêng= dịch vụ có thể dự đoán trước)

“Đường đi từ nguồn tới đích tương tự như mạng điện thoại
(telephone circuit)”

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



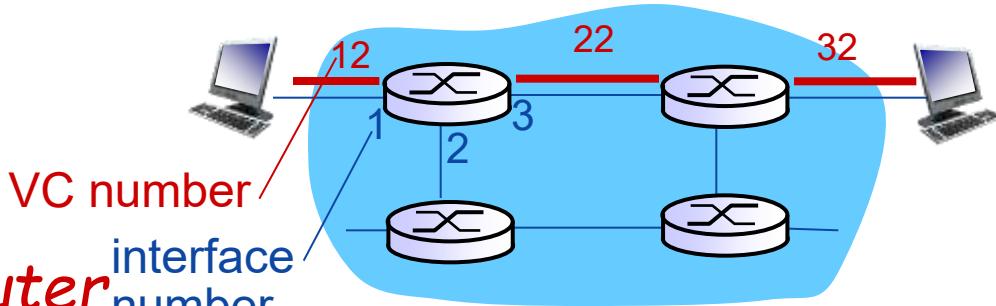
Sự thực hiện kết nối ảo (VC)

Một kết nối ảo bao gồm:

1. *Đường đi (path)* từ nguồn tới đích
2. *Số hiệu kết nối ảo (VC numbers)*, một số cho một kết nối dọc theo đường đi
3. *Các mục trong các bảng forwarding* ở trong các router dọc theo đường đi
 - Packet thuộc về kết nối ảo mang số hiệu (chứ không phải là điểm đến)
 - Số hiệu của kết nối ảo có thể được thay đổi trên mỗi kết nối
 - Số hiệu 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:

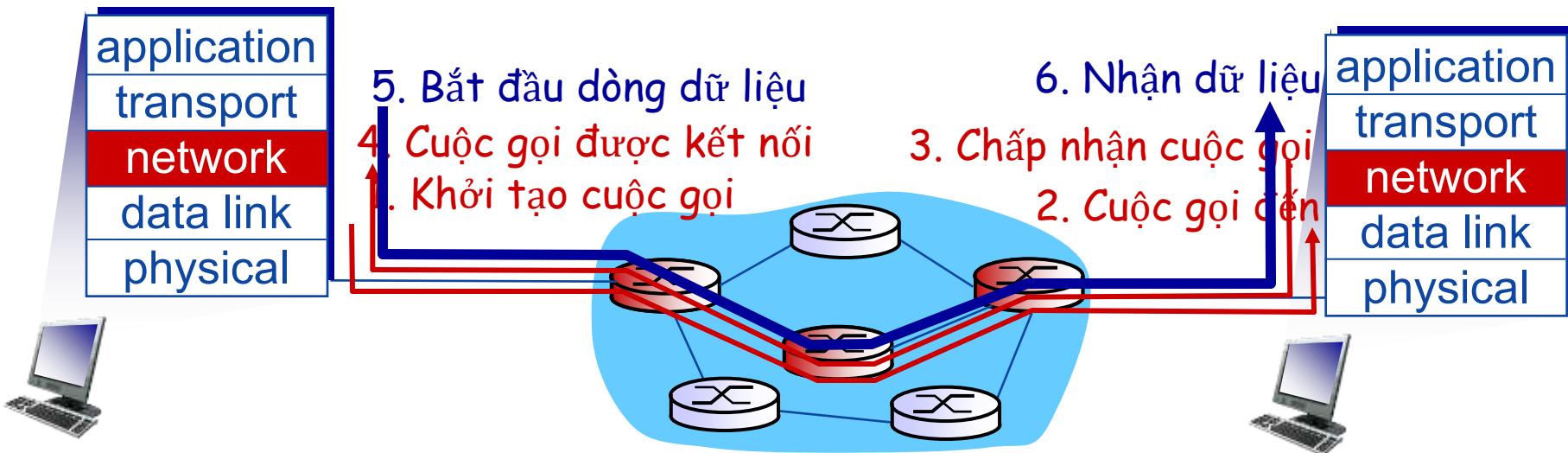
Cổng vào	số hiệu của kết nối ảo vào	Cổng ra	số hiệu của kết nối ảo ra
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...

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



Các mạch ảo: các giao thức gửi tín hiệu

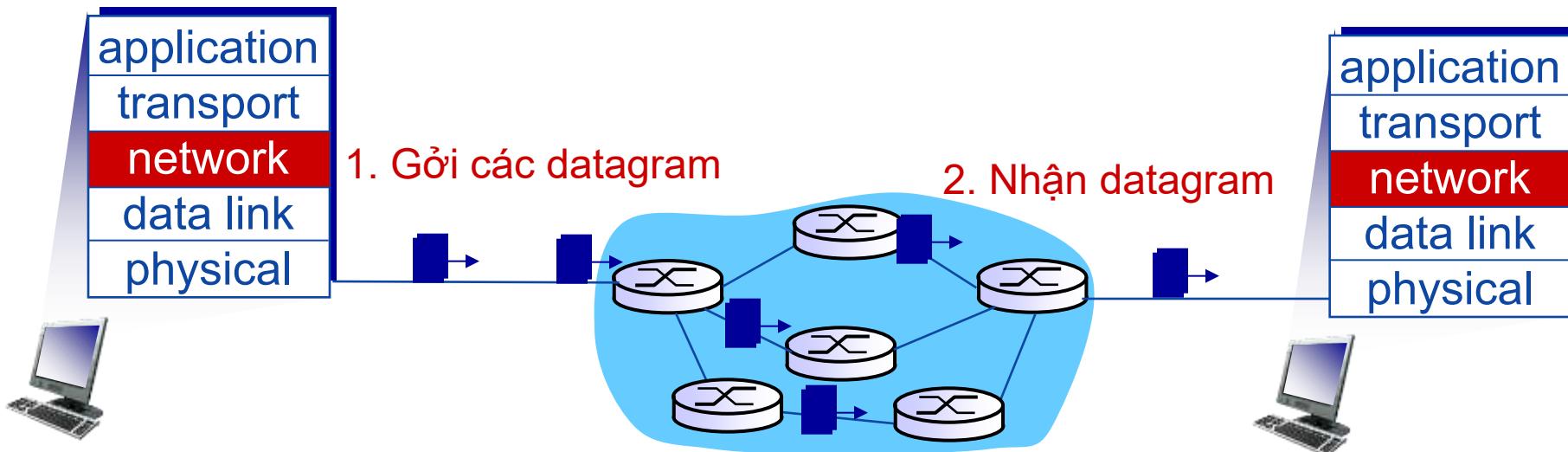
- Được dùng để thiết lập, duy trì kết nối ảo
- Được dùng trong ATM, frame-relay, X.25
- Không được sử dụng trong Internet ngày nay





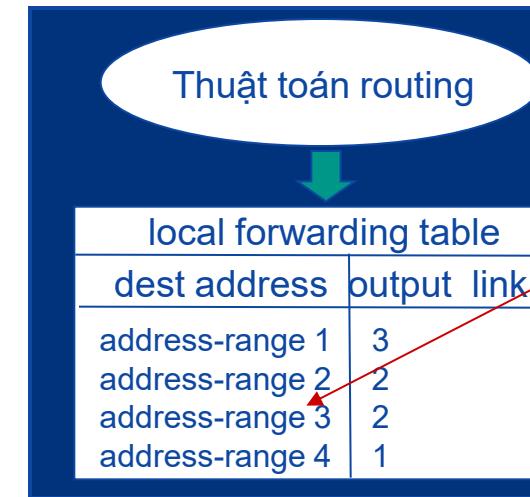
Mạng Datagram

- Không thiết lập cuộc gọi tại tầng network
- Các router: không có trạng thái về các kết nối giữa 2 điểm cuối
 - Không có khái niệm về mức network của “kết nối”
- Các packet được chuyển dùng địa chỉ của host đích

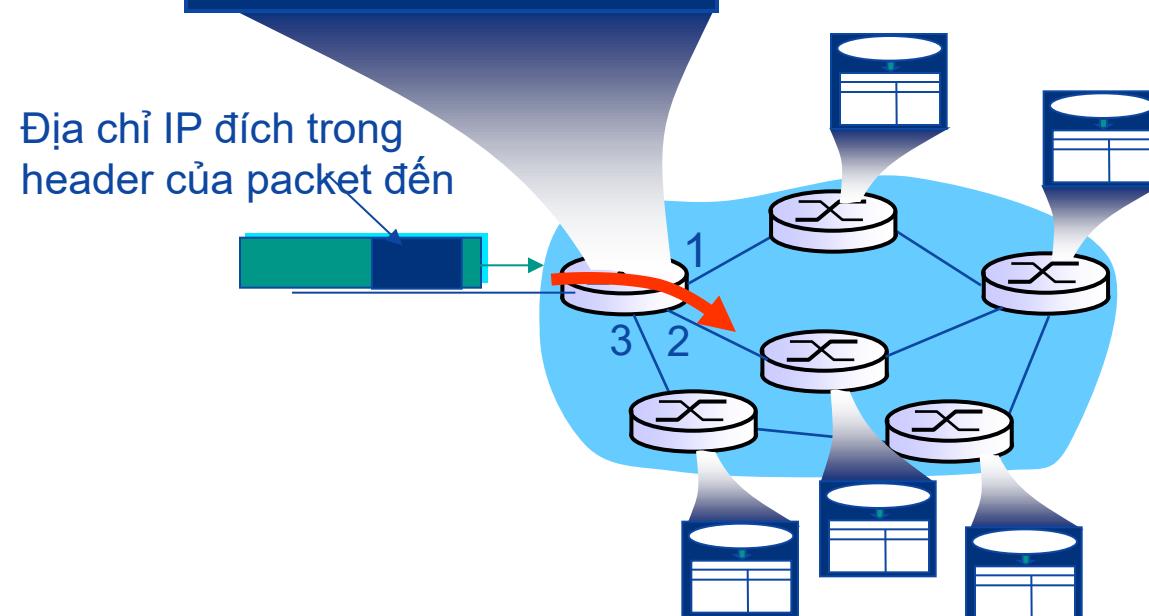




Bảng Datagram forwarding



4 tỉ địa chỉ IP, vì liệt kê ra 1 dãy các địa chỉ hơn là liệt kê ra từng địa chỉ (các mục tổng hợp)





Mạng datagram hoặc mạch ảo: tại sao?

Internet (datagram)

- Dữ liệu trao đổi giữa các máy tính
 - Dịch vụ “mềm dẻo, không định thì chặt chẽ.
- Nhiều kiểu link
 - 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
 - *Mạng bên trong đơn giản, phức tạp tại “mạng bên ngoài”*

ATM (mạch ảo)

- Được phát triển từ hệ thống điện thoại
- Đàm thoại của con người:
 - Định thì chặt chẽ, yêu cầu về độ tin cậy
 - Cần cho các dịch vụ bảo đảm
- Các hệ thống đầu cuối “ít thông minh”
 - Điện thoại
 - *Bên trong mạng phức tạp*



Chương 4: Nội dung

4.1 Tổng quan

- Mức dữ liệu
- Mức điều khiển

4.2 Virtual circuit network và datagram network

- Các mạch ảo
- Mạng datagram

4.3 Cấu trúc bên trong router

- Cổng đầu vào, chuyển đổi, cổng đầu ra
- Quản lý bộ đệm, lập kế hoạch

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing

- DHCP
- NAT
- ICMP
- IPV6

4.5 Các thuật toán routing

- Link state
- Distance vector

4.6 Routing trong Internet

- Hierarchical routing
- RIP, OSPF, BGP
- SDN
- Quản lý mạng

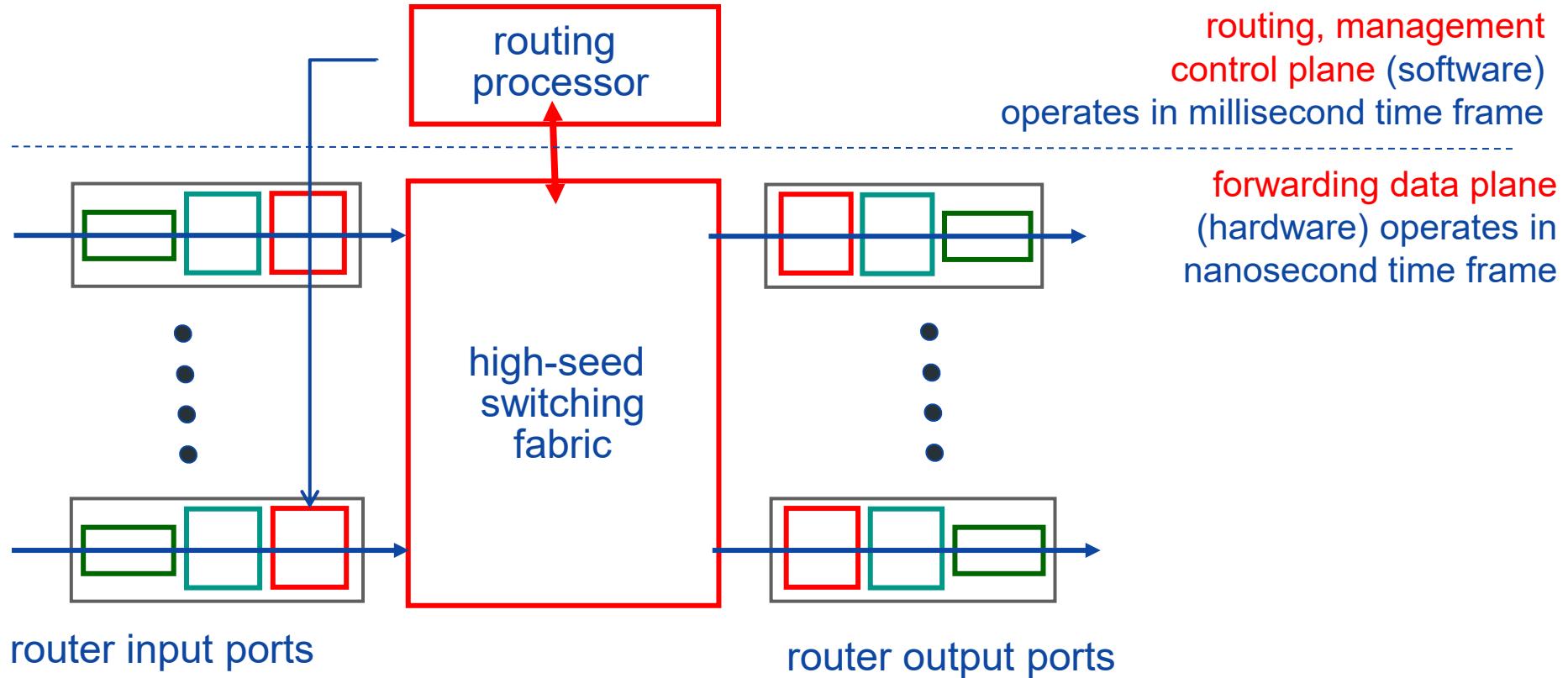
4.7 Broadcast và Multicast routing



Tổng quan kiến trúc Router

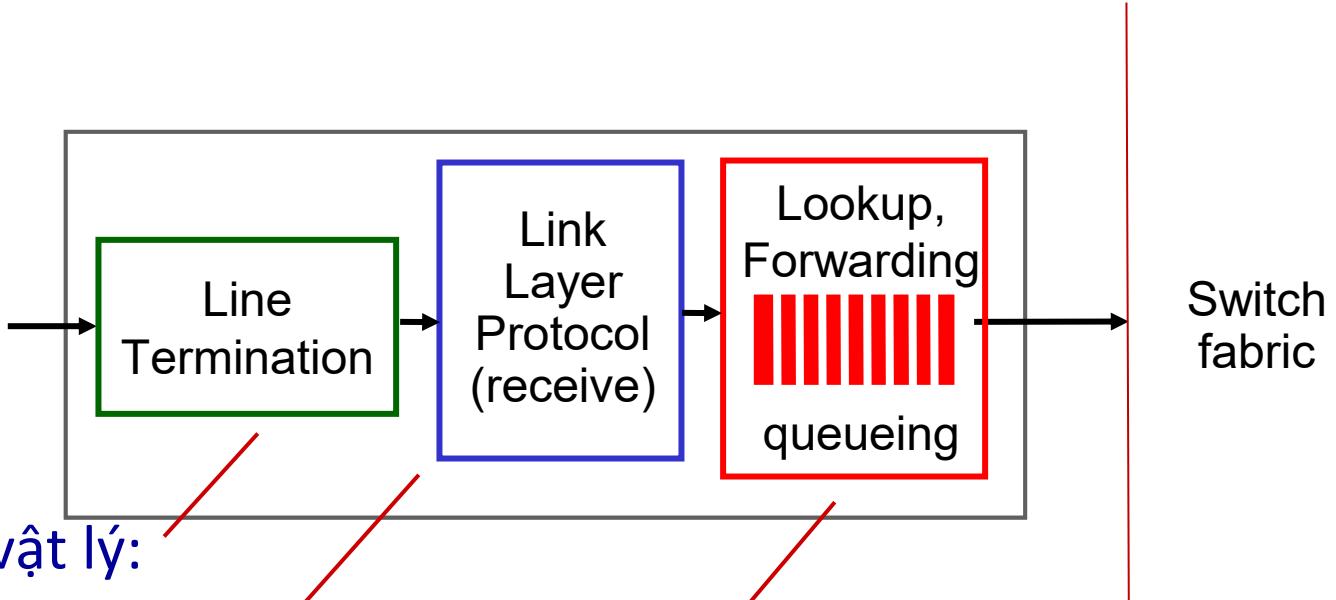
Hai chức năng chính của router:

- Chạy các giao thức/thuật toán routing (RIP, OSPF, BGP)
- Chuyển tiếp các datagram từ đường link vào tới đường link ra





Chức năng cổng đầu vào



Tầng vật lý:
tiếp nhận mức bit

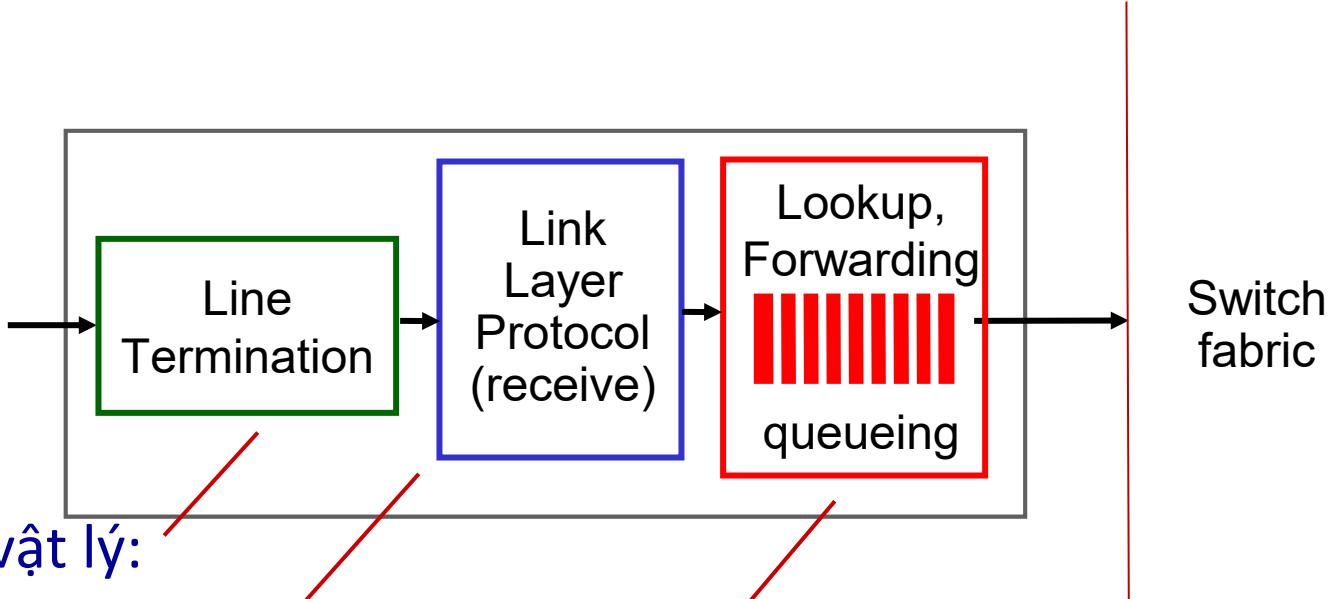
Tầng liên kết:
ví dụ: Ethernet
(Chương 5)

Chuyển mạch phi tập trung:

- Sử dụng các giá trị trong header, tra cứu cổng đầu ra bằng cách sử dụng bảng chuyển tiếp trong bộ nhớ cổng đầu vào ("match plus action")
- Mục tiêu: hoàn thành xử lý cổng đầu vào ở 'tốc độ đường truyền'
- Hàng đợi cổng đầu vào: nếu datagram đến nhanh hơn tốc độ chuyển tiếp vào



Chức năng cổng đầu vào



Tầng vật lý:
tiếp nhận mức bit

Tầng liên kết:
ví dụ: Ethernet
(Chương 5)

Chuyển mạch phi tập trung :

- Sử dụng các giá trị trong header, tra cứu cổng đầu ra bằng cách sử dụng bảng chuyển tiếp trong bộ nhớ cổng đầu vào (“*match plus action*”)
- **Chuyển tiếp dựa trên đích:** chỉ chuyển tiếp dựa trên địa chỉ IP đích (truyền thống)
- **Chuyển tiếp tổng quát:** chuyển tiếp dựa trên bất kỳ tập hợp giá trị nào trong header



Chuyển tiếp dựa trên địa chỉ đích đến

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00010000 00000100 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Hỏi: điều gì sẽ xảy ra nếu các phạm vi không được phân chia hợp lý?



Kết hợp tiền tố dài nhất

Tiền tố phù hợp dài nhất

Để tìm entry trong bảng chuyển tiếp phù hợp với địa chỉ đích nhất định, hãy sử dụng tiền tố địa chỉ *dài nhất* khớp với địa chỉ đích

Phạm vi địa chỉ đích	Cổng kết nối
11001000 00010111 00010 *** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011 *** *****	2
Ngược lại	3

ví dụ :

11001000 00010111 00010110 10100001

Cổng nào?

11001000 00010111 00011000 10101010

Cổng nào?



Kết hợp tiền tố dài nhất

Tiền tố phù hợp dài nhất

Để tìm entry trong bảng chuyển tiếp phù hợp với địa chỉ đích nhất định, hãy sử dụng tiền tố địa chỉ *dài nhất* khớp với địa chỉ đích

Phạm vi địa chỉ đích	Cổng kết nối
11001000 00010111 00010	0
11001000 00011111 00011000	1
11001000 00011111 00011000	2
nếu không thì	3

ví dụ :

11001000 00010111 00010110 10100001

Cổng nào?

11001000 00010111 00011000 10101010

Cổng nào?



Kết hợp tiền tố dài nhất

Tiền tố phù hợp dài nhất

Để tìm entry trong bảng chuyển tiếp phù hợp với địa chỉ đích nhất định, hãy sử dụng tiền tố địa chỉ *dài nhất* khớp với địa chỉ đích

Phạm vi địa chỉ đích	Cổng kết nối
11001000 00010111 00010 *** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011 *** *****	2
nếu không thì	3

↑
match!

11001000 00010111 00010110 10100001
11001000 00010111 00011000 10101010

Cổng nào?

Cổng nào?

ví dụ :



Kết hợp tiền tố dài nhất

Tiền tố phù hợp dài nhất

Khi tìm mục nhập bảng chuyển tiếp cho địa chỉ đích nhất định, hãy sử dụng tiền tố địa chỉ *dài nhất* khớp với địa chỉ đích.

Phạm vi địa chỉ đích	Cổng kết nối
11001000 00010111 00010 *** *****	0
11001000 00010111 00011000 *****	1
11001000 00011111 00011 *** *****	2
nếu không tr match!	3

ví dụ :

11001000 00011111 00010110 10100001
11001000 00010111 00011000 10101010

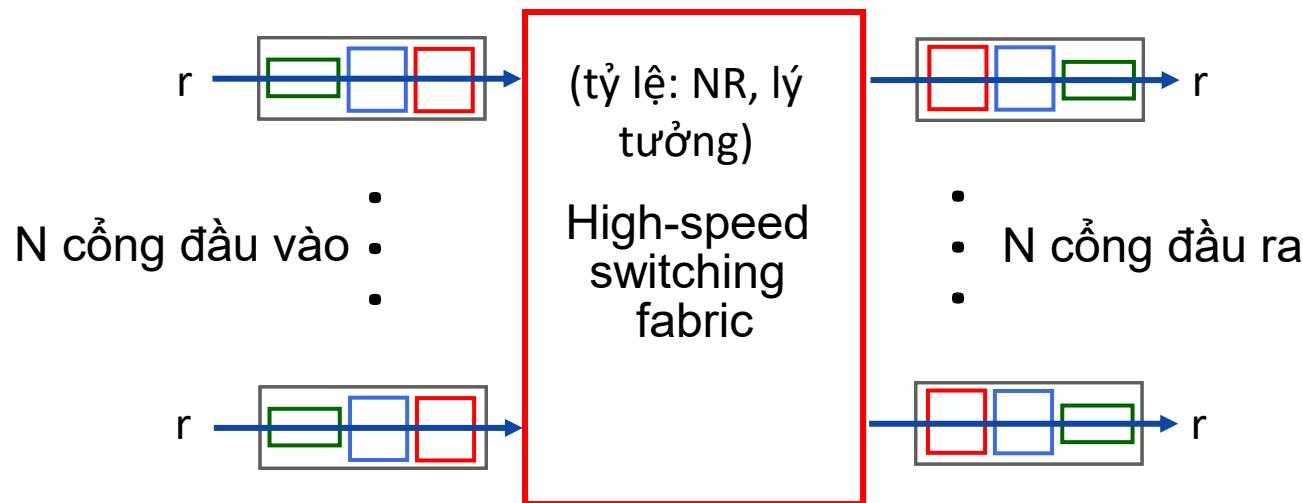
Cổng nào?

Cổng nào?



Switching fabrics

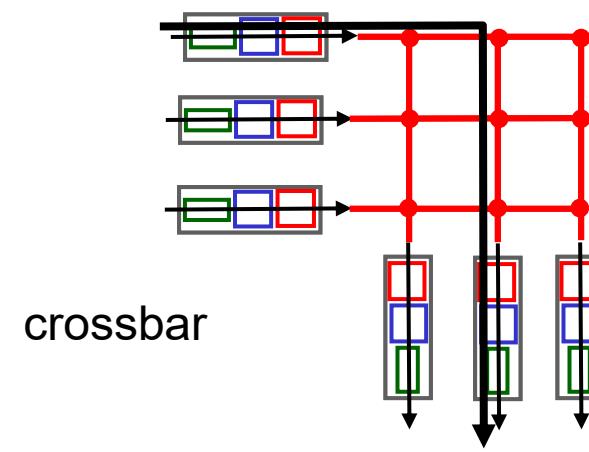
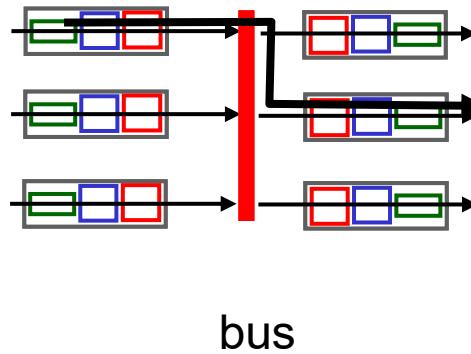
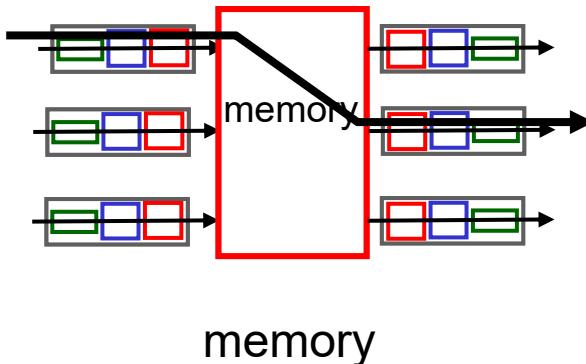
- Chuyển packet từ cổng đầu vào đến cổng đầu ra thích hợp
- Tốc độ chuyển đổi: tốc độ mà các packet có thể được chuyển từ đầu vào sang đầu ra
 - Thường được đo bằng bội số của tốc độ dòng đầu vào/đầu ra
 - N đầu vào: tốc độ chuyển đổi N lần tốc độ dòng mong muốn





Switching fabrics

- Chuyển packet từ cổng đầu vào đến cổng đầu ra thích hợp
- Tốc độ chuyển đổi: tốc độ mà các packet có thể được chuyển từ đầu vào sang đầu ra
 - Thường được đo bằng bội số của tốc độ dòng đầu vào/đầu ra
 - N đầu vào: tốc độ chuyển đổi N lần tốc độ dòng mong muốn
- Ba loại switching fabric chính:

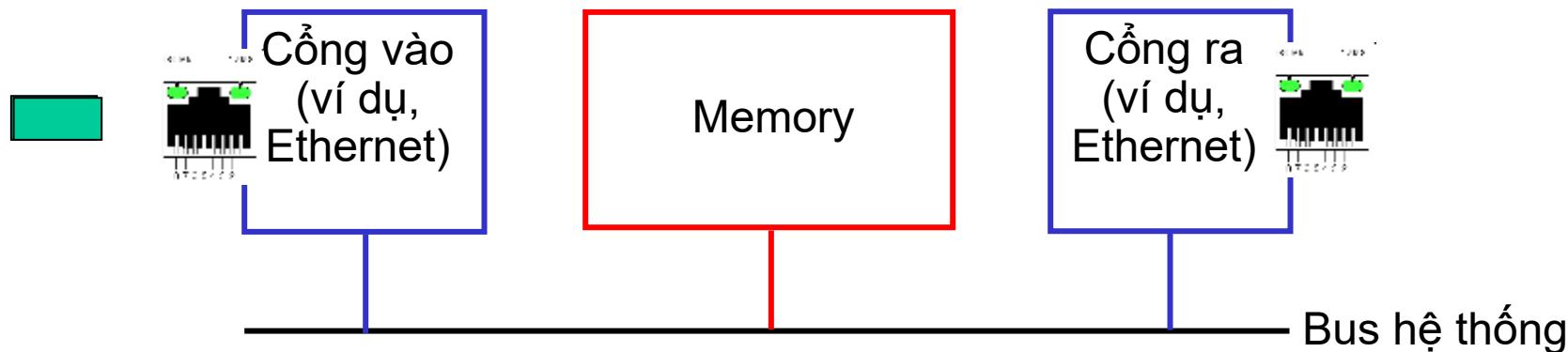




Switching thông qua bộ nhớ (memory)

Các router thế hệ đầu tiên:

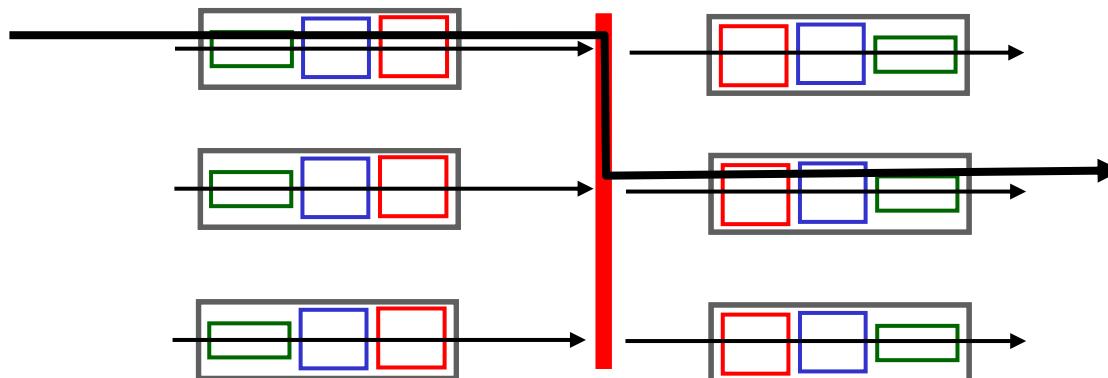
- Các máy tính cổ điển với switching dưới sự kiểm soát trực tiếp của CPU
- Packet được sao chép vào bộ nhớ của hệ thống
- Tốc độ bị giới hạn bởi băng thông của bộ nhớ (2 bus qua mỗi datagram)





Switching thông qua bus

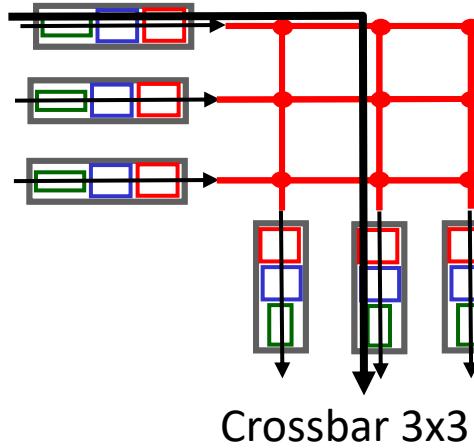
- Datagram từ bộ nhớ cổng đầu vào đến bộ nhớ cổng đầu ra thông qua một bus chia sẻ
- *Tranh chấp bus*: tốc độ chuyển mạch bị giới hạn bởi băng thông của bus
- Bus 32 Gbps trên Cisco 5600: đủ tốc độ cho các router truy cập



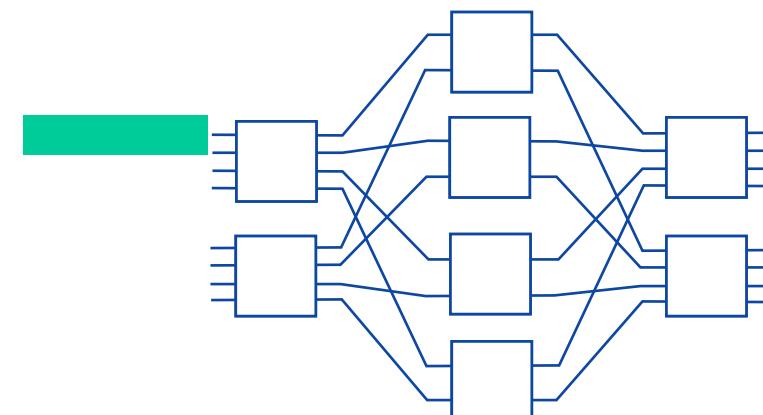


Switching thông qua interconnection network

- Mạng Crossbar, Clos, các mạng kết nối khác ban đầu được phát triển để kết nối các bộ vi xử lý trong bộ đa xử lý
- **Switch đa tầng:** switch $n \times n$ được tạo thành từ nhiều switch nhỏ hơn
- Khai thác song song:
 - Phân mảnh datagram vào các ô có độ dài cố định khi đi vào
 - Chuyển các ô thông qua fabric, tập hợp lại datagram khi đi ra



Crossbar 3x3



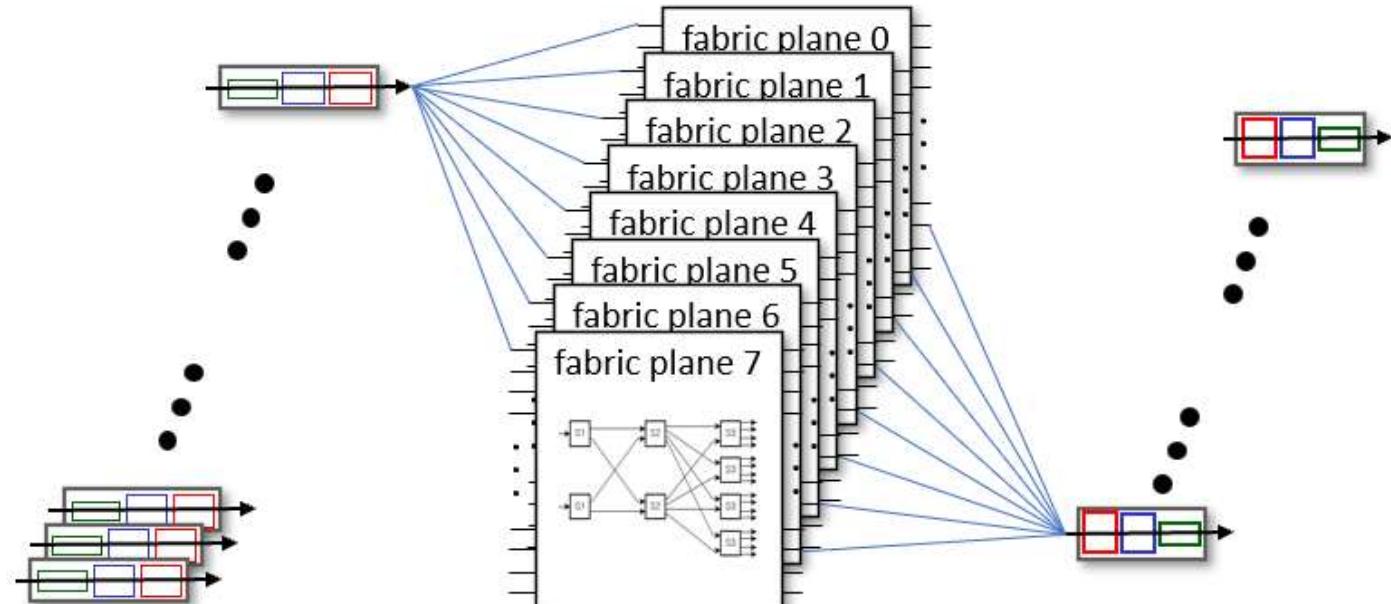
Switch đa tầng 8x8

được xây dựng từ các switch có kích thước nhỏ hơn

Switching thông qua interconnectiong network



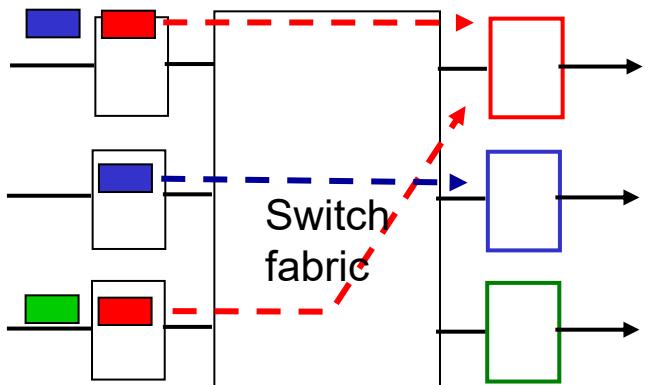
- Chia tỷ lệ, sử dụng song song nhiều “mặt phẳng” switching:
- Tăng tốc, mở rộng quy mô thông qua song song
- Router CRS của Cisco:
 - Đơn vị cơ bản: 8 switching planes
 - Mỗi mặt phẳng: mạng kết nối 3 tầng
 - Khả năng switching lên đến 100 Tbps



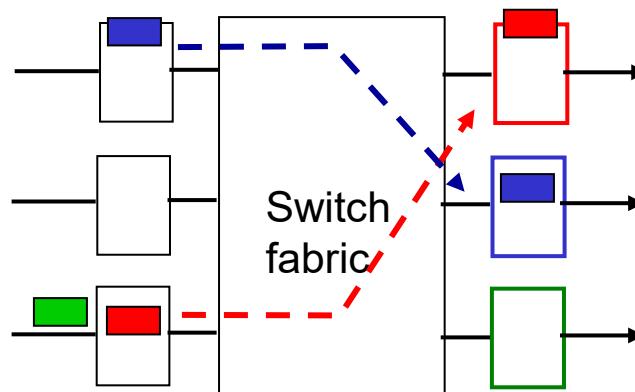


Hàng đợi tại cổng đầu vào

- Nếu việc switching chậm hơn so với các cổng đầu vào kết hợp -> hàng đợi có thể xảy ra ở đầu vào
- Chậm trễ và mất dữ liệu do tràn bộ đệm đầu vào!
- **Chặn Head-of-the-Line (HOL):** datagram xếp hàng ở phía trước hàng đợi ngăn những datagram khác trong hàng đợi di chuyển về phía trước



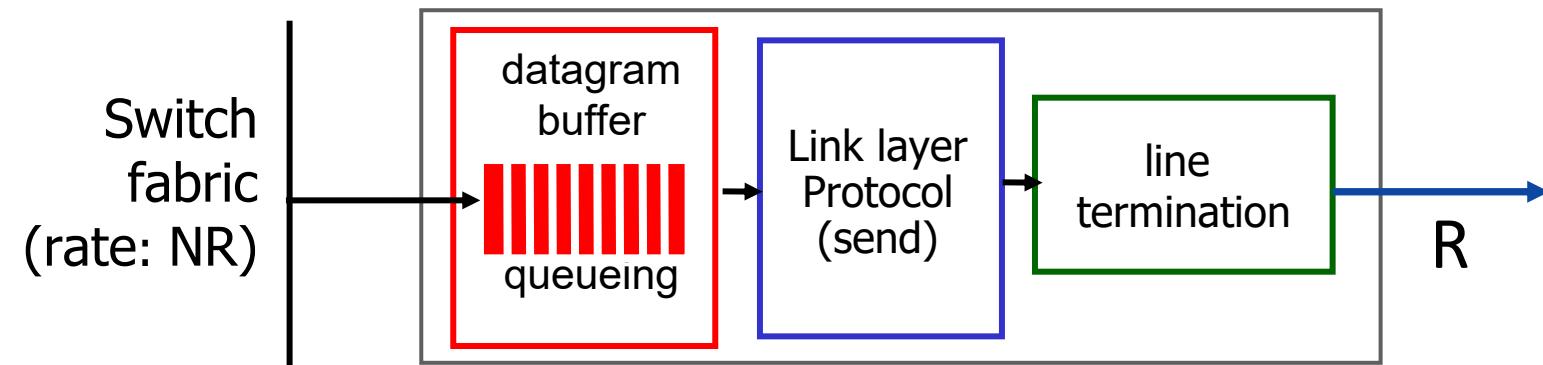
tranh chấp tại cổng đầu ra: chỉ có thể truyền một packet màu đỏ, packet màu đỏ thấp hơn sẽ bị **chặn**



One packet time later: packet màu xanh lá cây bị chặn HOL



Hàng đợi cổng đầu ra

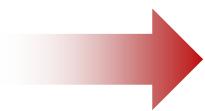


Đây là một slide thực sự quan trọng

- **Bộ đếm** khi các datagram đến từ fabric nhanh hơn tốc độ truyền của liên kết. **Chính sách loại bỏ:** datagram nào sẽ bị loại bỏ nếu không có bộ đếm trống?
- **Nguyên tắc lập lịch** chọn trong số các datagram được xếp hàng đợi để truyền



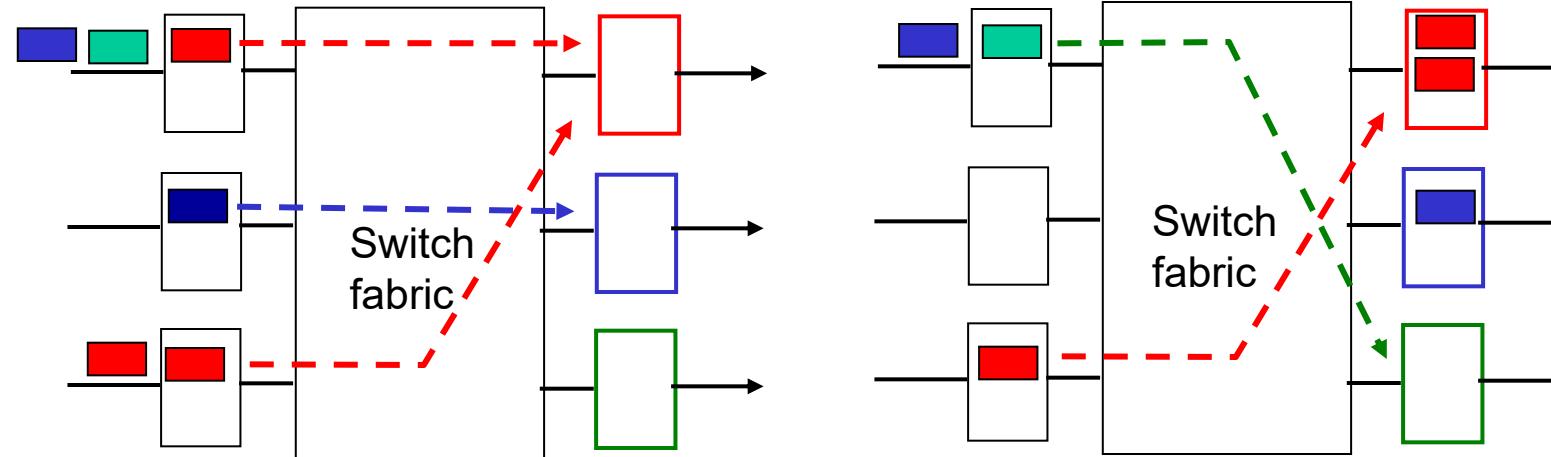
Datagram có thể bị mất do tắc nghẽn, thiếu bộ đếm



Lập kế hoạch ưu tiên – có hiệu suất tốt nhất, tính trung lập của mạng



Hàng đợi cổng đầu ra



tại t , các packet nhiều
hơn từ đầu vào đến
đầu ra

one packet time later

- Đếm khi tốc độ đến vượt quá tốc độ dòng đầu ra
- **Xếp hàng (trễ) và mất gói do tràn bộ đếm tại cổng đầu ra!**



Bao nhiêu bộ đệm?

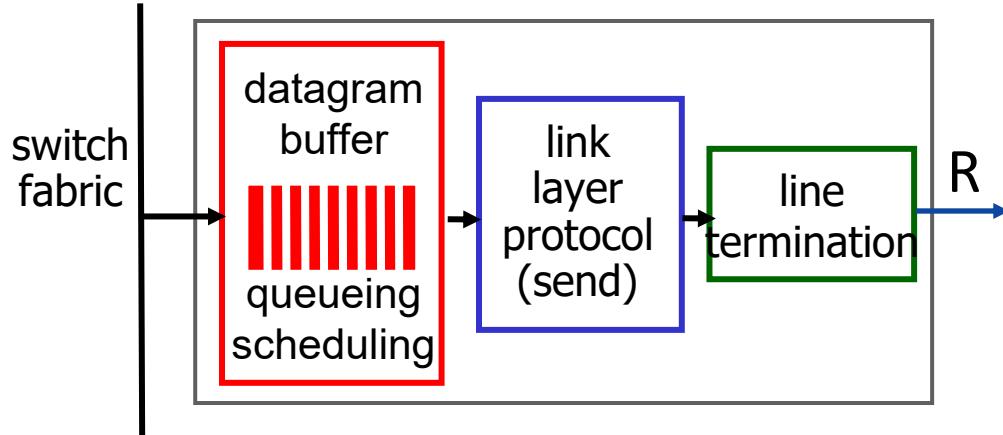
- Quy tắc ngón tay cái RFC 3439: bộ đệm trung bình bằng “typical” RTT (giả sử 250 mili giây) lần dung lượng đường link C
 - Ví dụ: liên kết C = 10 Gbps: bộ đệm 2,5 Gbit
- Đề xuất gần đây hơn: với N luồng, bộ đệm bằng

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

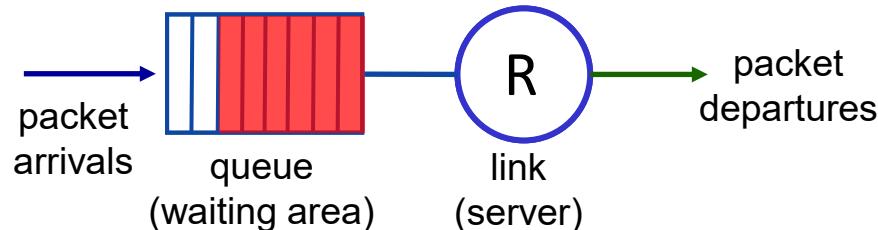
- Nhưng quá nhiều bộ đệm có thể làm tăng độ trễ (đặc biệt là trong các router gia đình)
 - RTT dài: hiệu suất kém cho các ứng dụng thời gian thực, phản hồi TCP chậm chạp
 - kiểm soát tắc nghẽn dựa trên độ trễ: “giữ liên kết cổ chai vừa đủ đầy (bận)”



Quản lý bộ đệm



Abstraction: queue



Quản lý bộ đệm:

- **Drop:** packet nào được thêm vào, drop khi bộ đệm đầy
 - Drop ở cuối: drop packet đến
 - Ưu tiên: drop/remove trên cơ sở độ ưu tiên
- **Đánh dấu:** đánh dấu packet để báo hiệu tắc nghẽn (ECN, RED)



Lập lịch truyền packet: FCFS

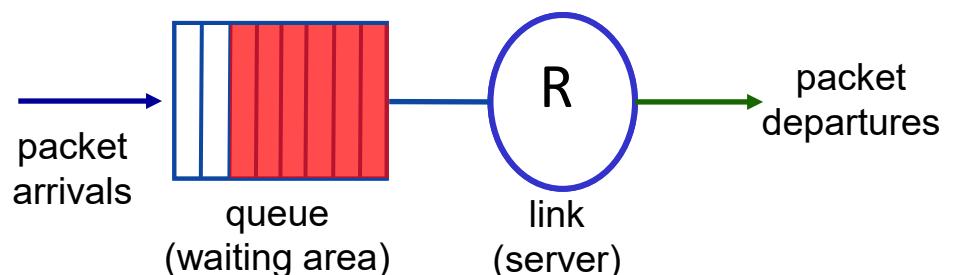
Lập lịch truyền packet: quyết định packet nào sẽ gửi tiếp theo trên liên kết

- Đến trước được phục vụ trước (FCFS)
- Độ ưu tiên (Priority)
- Thi đấu vòng tròn (Round Robin)
- Xếp hàng công bằng có trọng số

FCFS: các packet được truyền theo thứ tự đến cổng ra

- Còn được gọi là: Nhập trước xuất trước (FIFO)
- Ví dụ thế giới thực?

Abstraction: queue

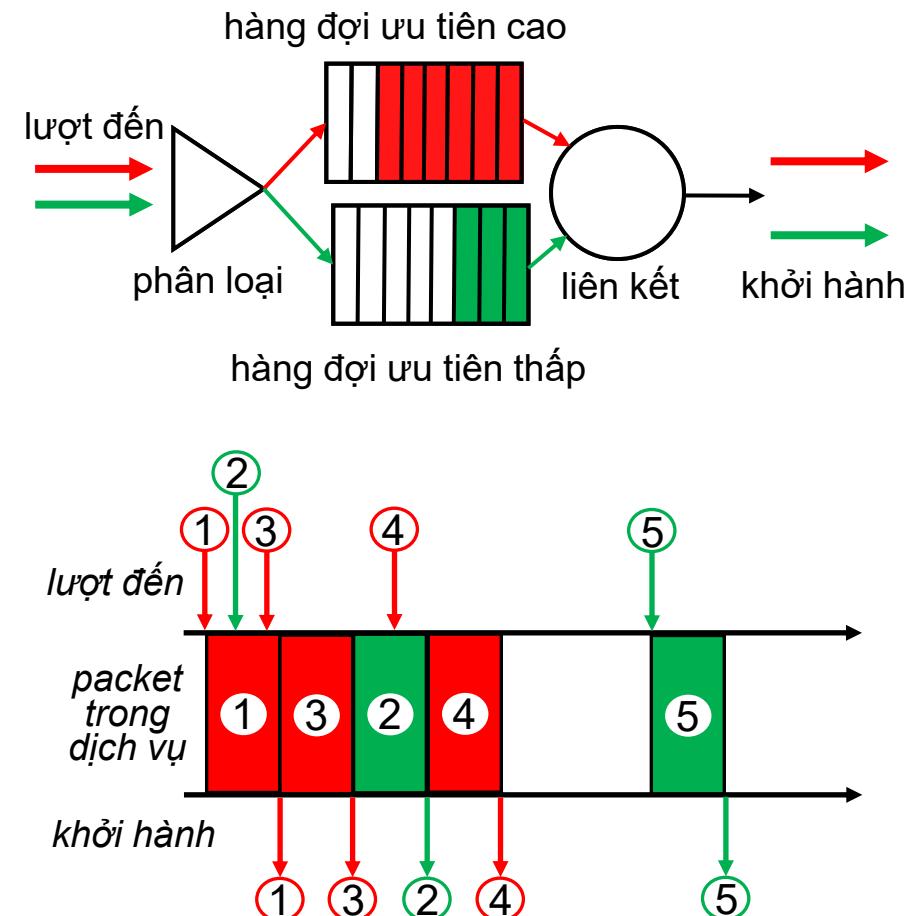




Chính sách lập kế hoạch: priority

Lập kế hoạch ưu tiên:

- Traffic đến được phân loại, xếp hàng theo lớp
 - Bất kỳ trường header nào cũng có thể được sử dụng để phân loại
- Gửi packet từ hàng đợi ưu tiên cao nhất có packet đệm
 - FCFS trong lớp ưu tiên

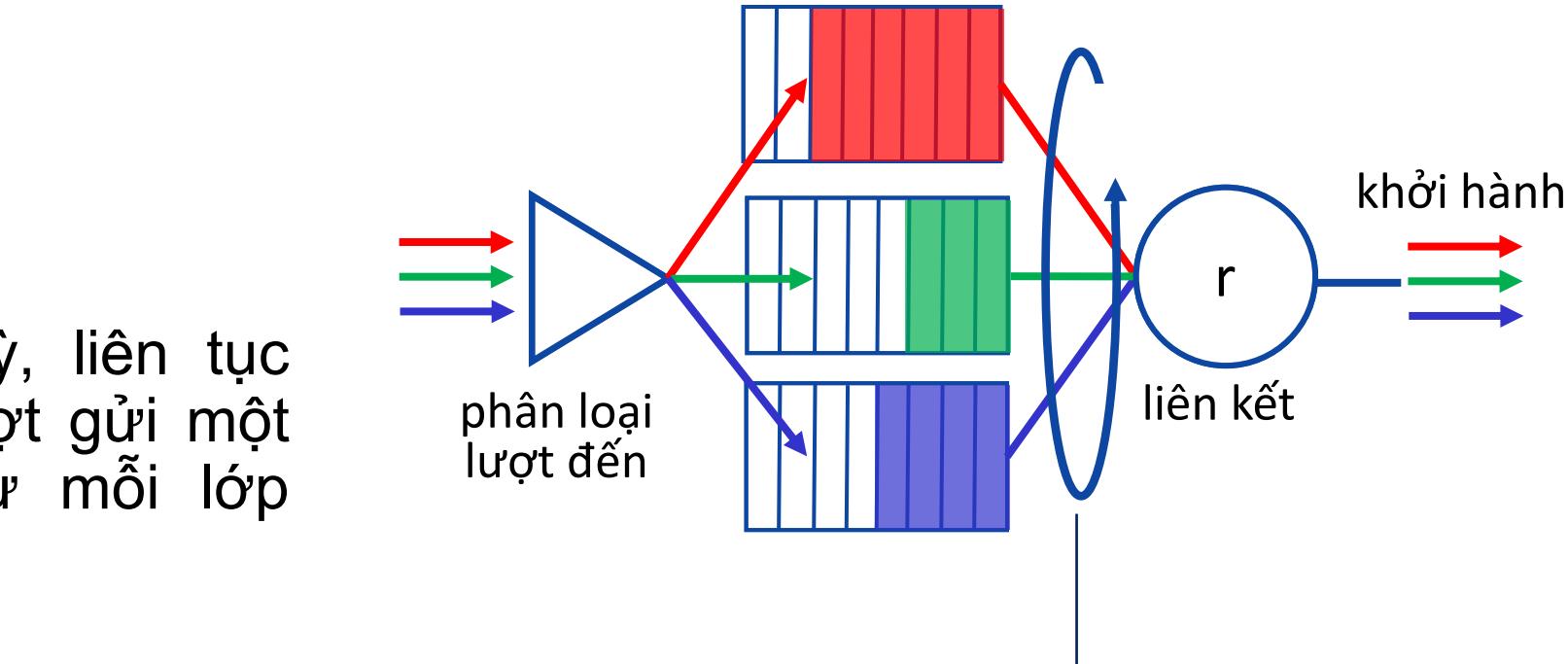




Chính sách lập kế hoạch: round robin

Lịch Round Robin (RR):

- Traffic đến được phân loại, xếp hàng theo lớp
 - Bất kỳ trường header nào cũng có thể được sử dụng để phân loại
- Máy chủ theo chu kỳ, liên tục quét hàng đợi, lần lượt gửi một packet hoàn chỉnh từ mỗi lớp (nếu có)





Chính sách lập kế hoạch: hàng đợi công bằng có trọng số

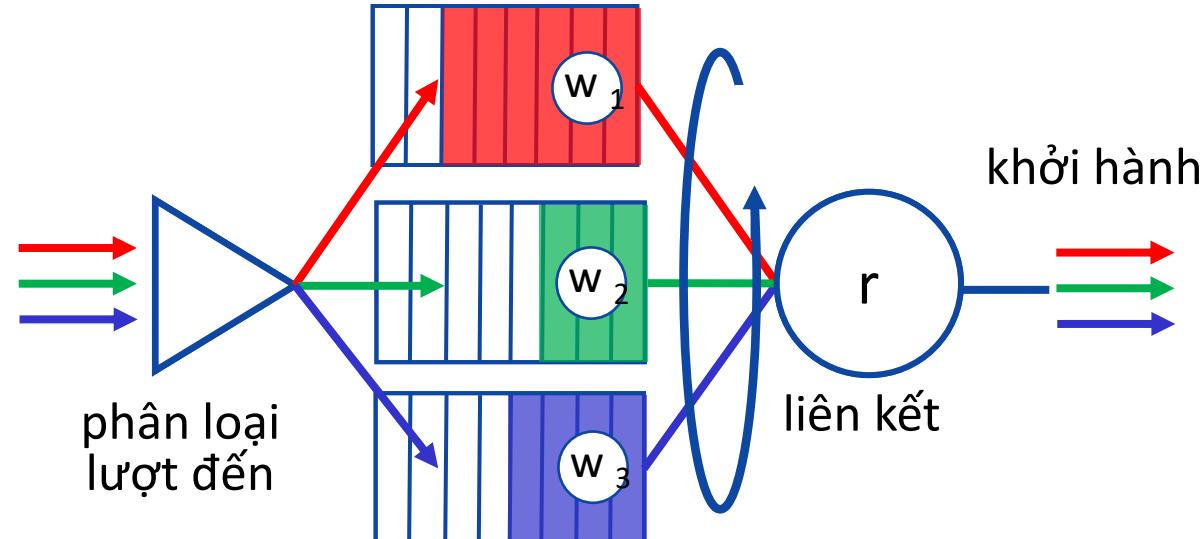
Xếp hàng công bằng có trọng số (WFQ):

- Round Robin tổng quát

- Mỗi lớp i , có trọng số w_i , và được trọng số lượng dịch vụ trong mỗi chu kỳ:

$$\frac{w_i}{\sum_j w_j}$$

- Đảm bảo băng thông tối thiểu (per-traffic-class)





Sidebar: Tính trung lập của mạng

Tính trung lập của mạng là gì?

- *Kỹ thuật:* cách một ISP nên chia sẻ/phân bổ tài nguyên của mình
 - Lập lịch packet, quản lý bộ đệm là các *cơ chế*
- Nguyên tắc *kinh tế, xã hội*
 - Bảo vệ tự do ngôn luận
 - Khuyến khích đổi mới, cạnh tranh
- Thi hành *hợp pháp* các quy tắc và chính sách

Các quốc gia khác nhau có “quan điểm” khác nhau về tính trung lập của mạng



Chương 4: Nội dung

4.1 Tổng quan

- Mức dữ liệu
- Mức điều khiển

4.2 Virtual circuit network và datagram network

- Các mạch ảo
- Mạng datagram

4.3 Cấu trúc bên trong router

- Cổng đầu vào, chuyển đổi, cổng đầu ra
- Quản lý bộ đệm, lập kế hoạch

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing

- DHCP
- NAT
- ICMP
- IPV6

4.5 Các thuật toán routing

- Link state
- Distance vector

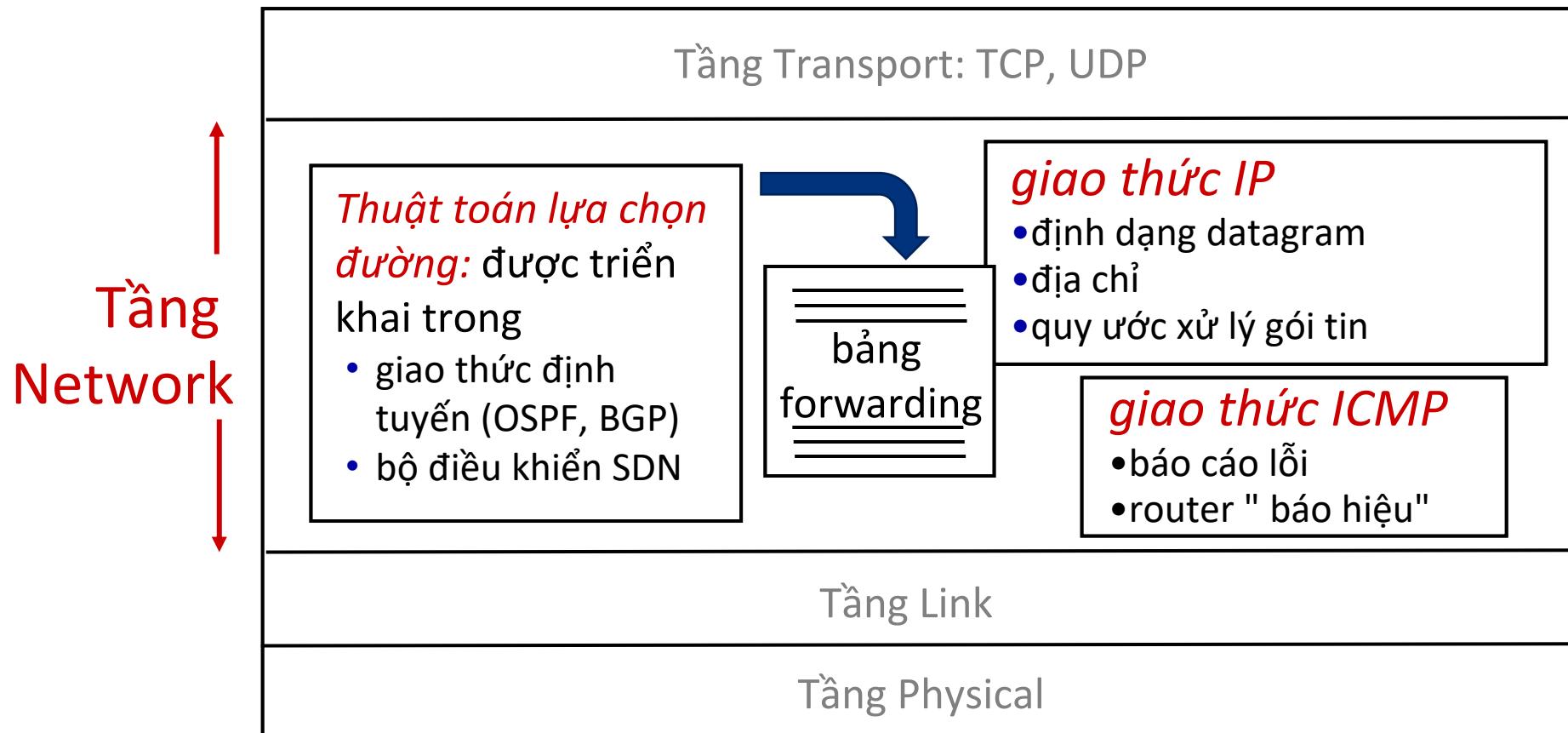
4.6 Routing trong Internet

- Hierarchical routing
- RIP, OSPF, BGP
- SDN
- Quản lý mạng

4.7 Broadcast và Multicast routing

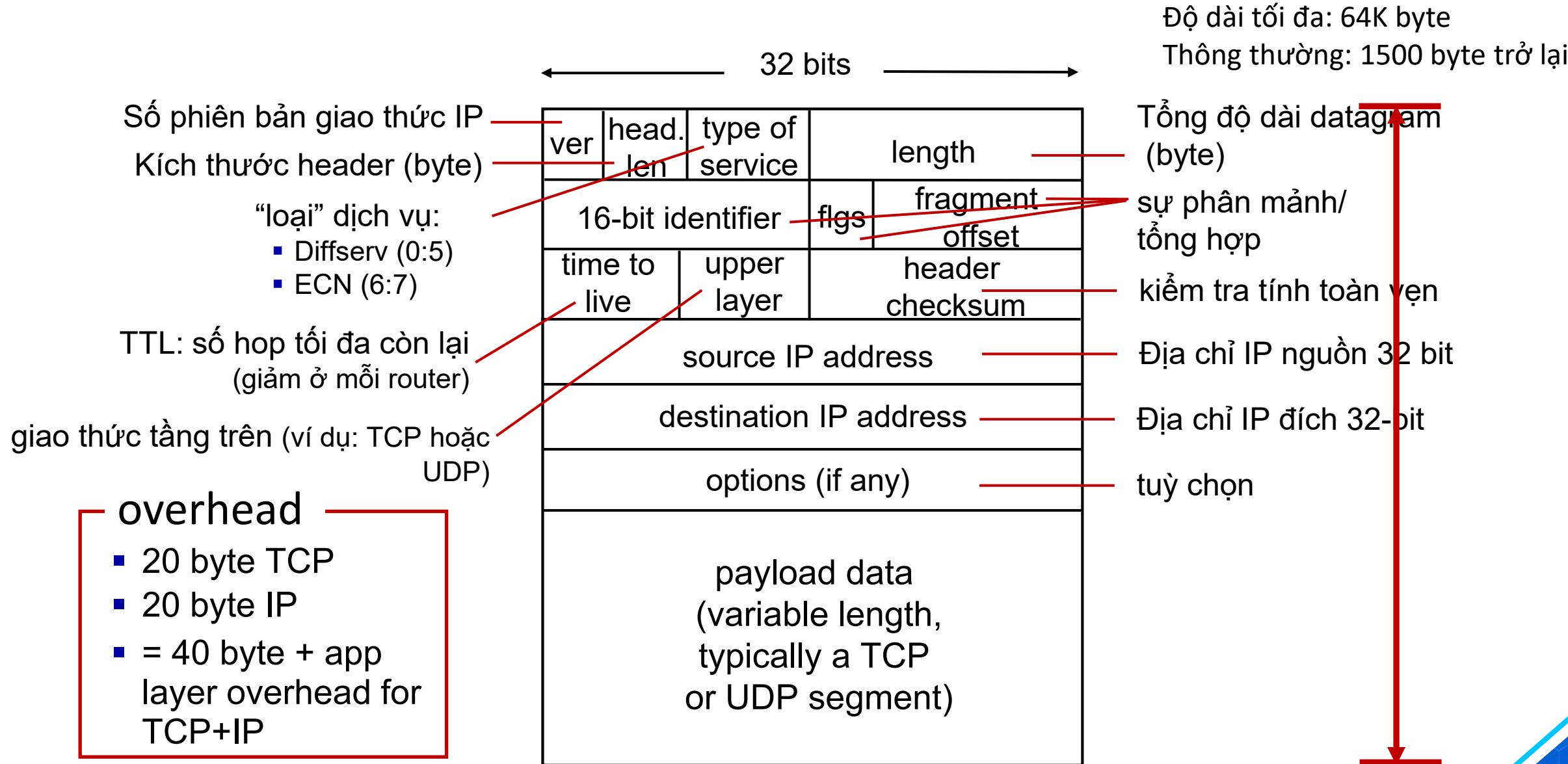


Chức năng tầng mạng của host, router:





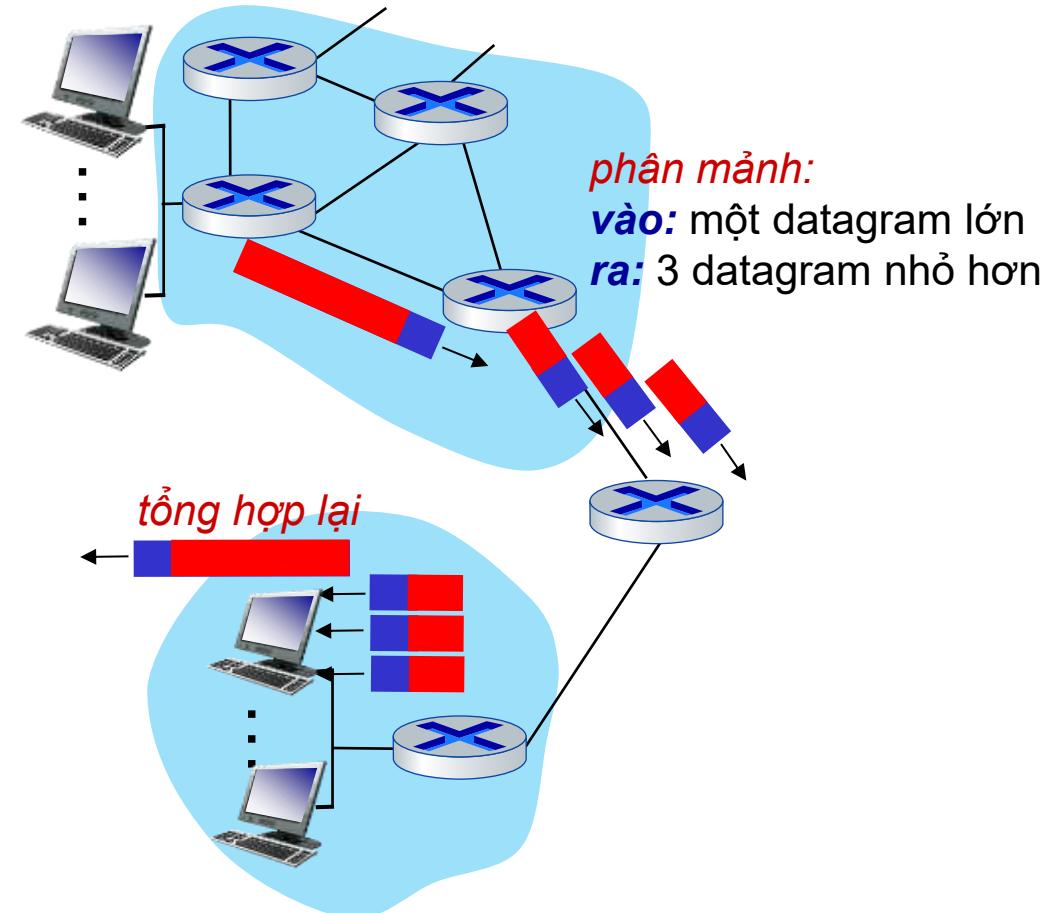
Định dạng của IP datagram





Phân mảnh và tổng hợp IP

- Các đường link mạng có MTU (max transfer size - kích thước truyền tối đa)
 - frame lớn nhất có thể ở mức kết nối
 - Loại liên kết khác nhau, MTU khác nhau
- Datagram lớn được chia (“phân mảnh”) trong mạng
 - Một datagram trở thành một số datagram
 - Chỉ “tổng hợp” tại *điểm đến*
 - Các bit của IP header được sử dụng để xác định, sắp xếp các fragment liên quan





Phân mảnh và tổng hợp IP

Ví dụ:

- Gói tin 4000 byte
- MTU = 1500 byte

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

*một datagram lớn trở thành
một số datagram nhỏ hơn*

1480 byte trong
trường dữ liệu

offset =
 $1480/8$

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

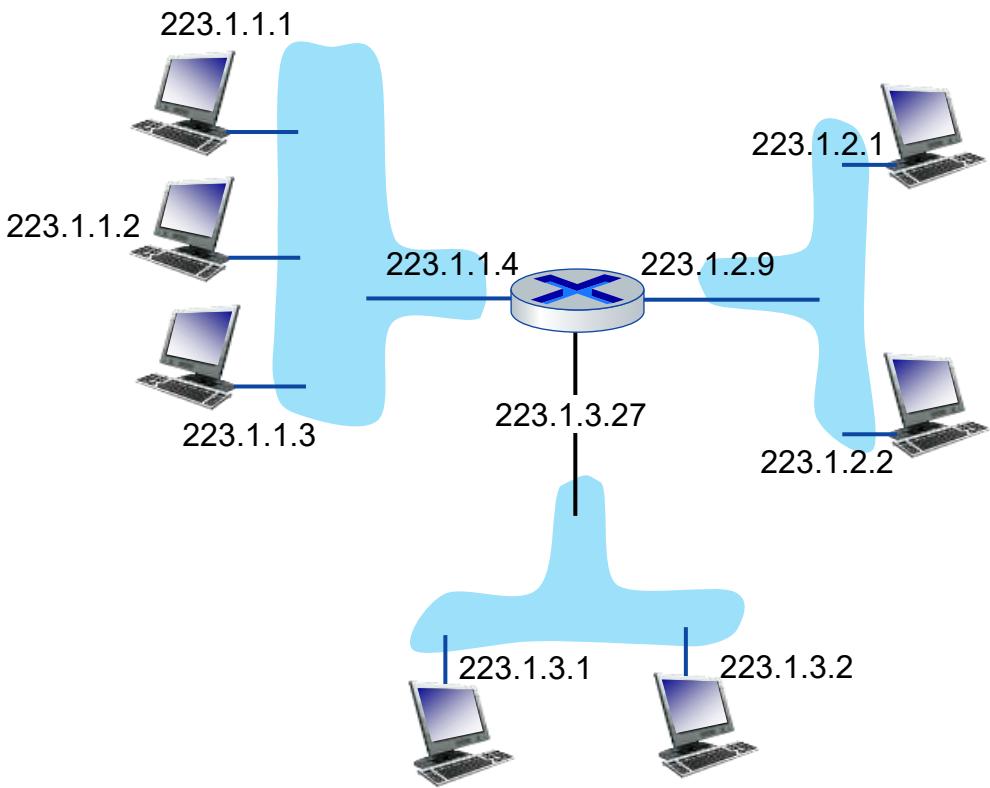
	length =1500	ID = x	fragflag =1	offset =185	
--	-----------------	-----------	----------------	----------------	--

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



Địa chỉ IP: giới thiệu

- Địa chỉ IP: Mã định danh 32 bit được liên kết với từng *interface* của host hoặc router
- Interface: kết nối giữa host/router và liên kết vật lý
 - Router thường có nhiều interface
 - Host thường có một hoặc hai interface (ví dụ: Ethernet có dây, 802.11 không dây)



ký hiệu địa chỉ IP dạng thập phân:

223.1.1.1 =

11011111	00000001	00000001	00000001
223	1	1	1



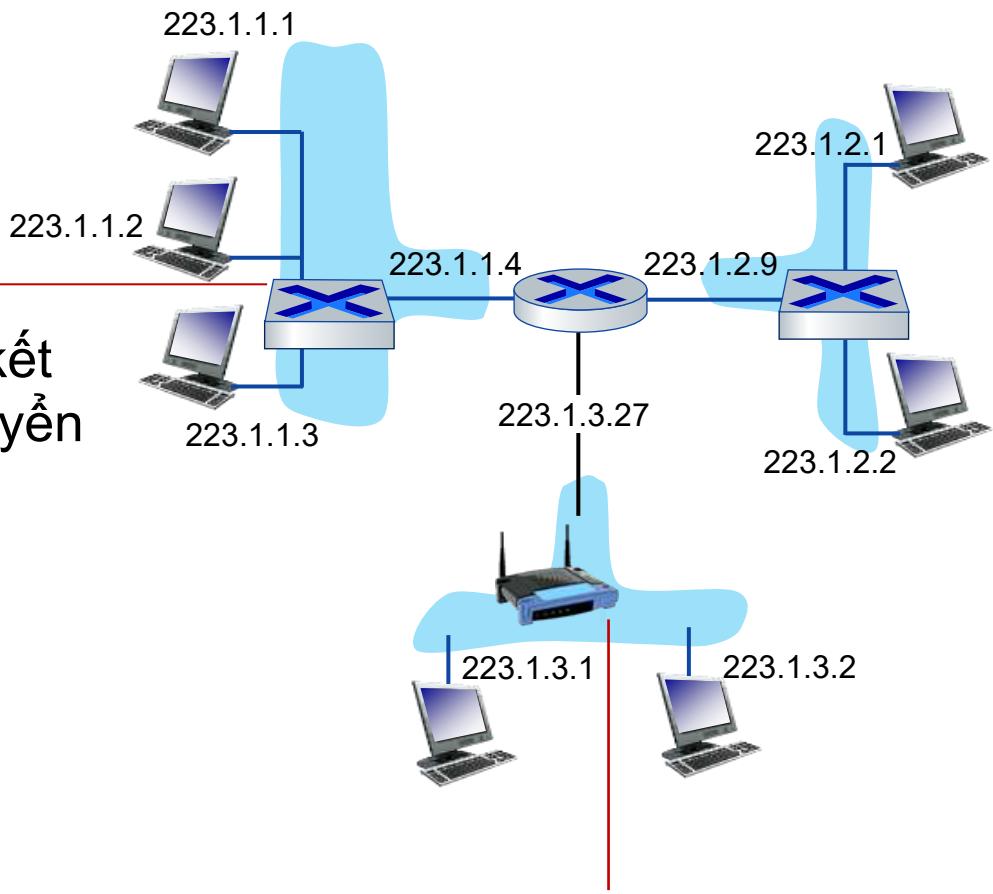
Địa chỉ IP: giới thiệu

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

Đáp: chúng ta sẽ tìm hiểu về điều đó trong phần sau

Hiện tại: không cần lo lắng về cách một interface được kết nối với interface khác (không có router can thiệp)

1: có dây
Interface được kết nối bằng bộ chuyển mạch Ethernet

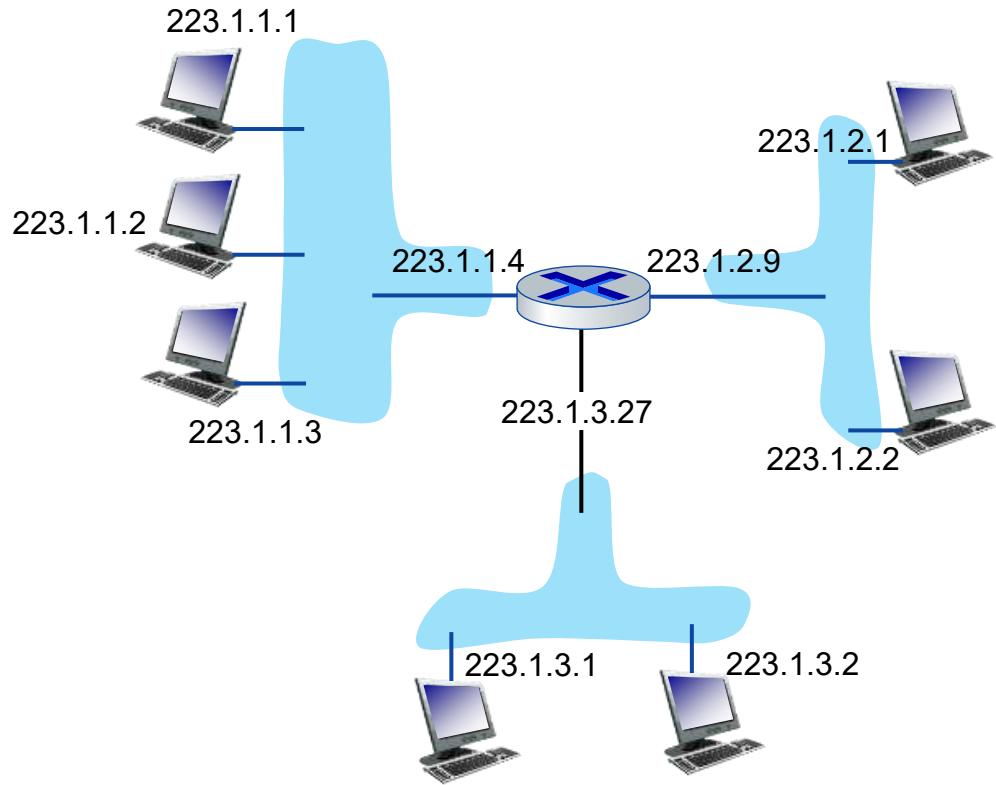


2: Interface không dây được kết nối bởi trạm phát WiFi



Mạng con (subnet)

- *Mạng con (subnet) là gì?*
 - Interface của thiết bị có thể tiếp cận vật lý với nhau **mà không cần thông qua router can thiệp**
- Địa chỉ IP có cấu trúc:
 - **Phần mạng con:** các thiết bị trong cùng một mạng con có chung các bit bậc cao
 - **Phần host:** các bit thứ tự thấp còn lại

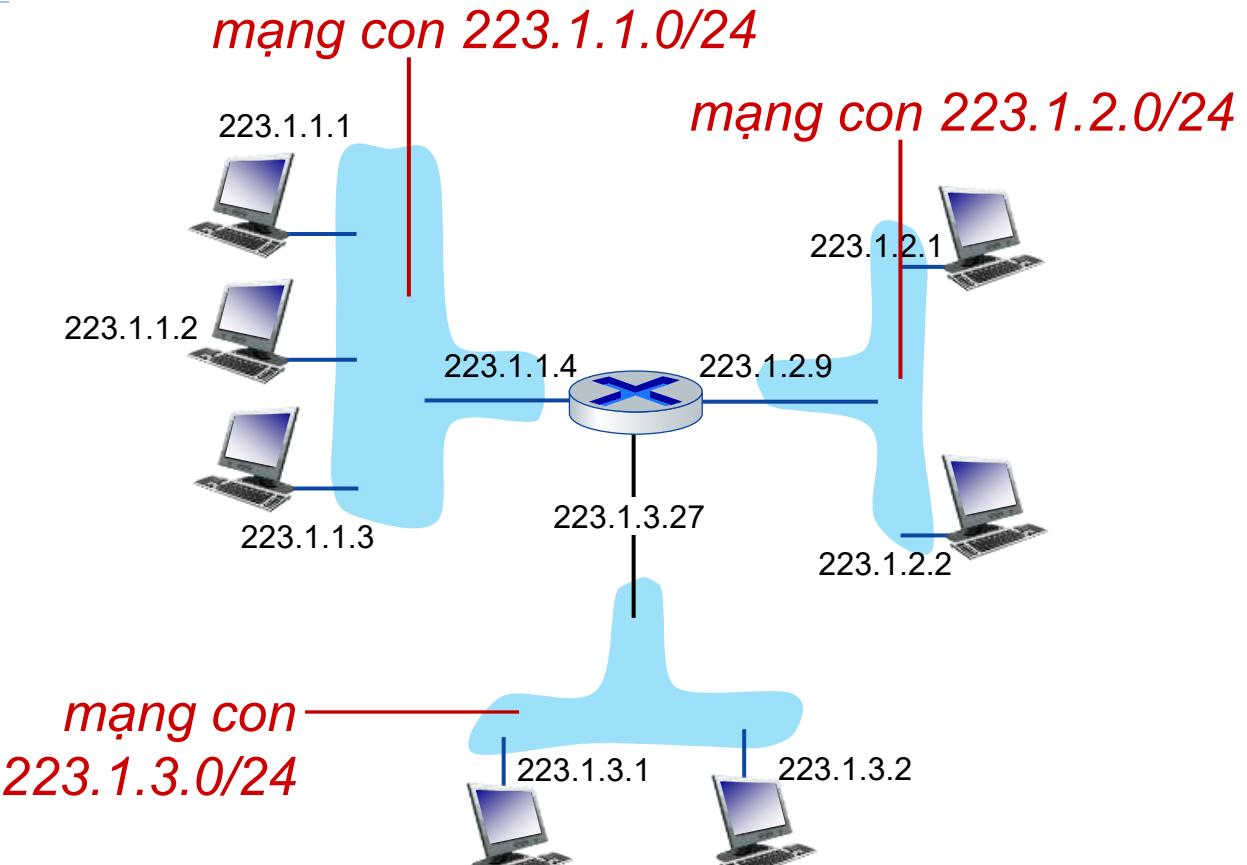


mạng gồm 3 mạng con



Mạng con

- *Công thức xác định mạng con:*
 - Tách từng interface khỏi host hoặc router của nó, tạo ra các “đảo” của các mạng bị cô lập
 - Mỗi mạng bị cô lập được gọi là *mạng con*

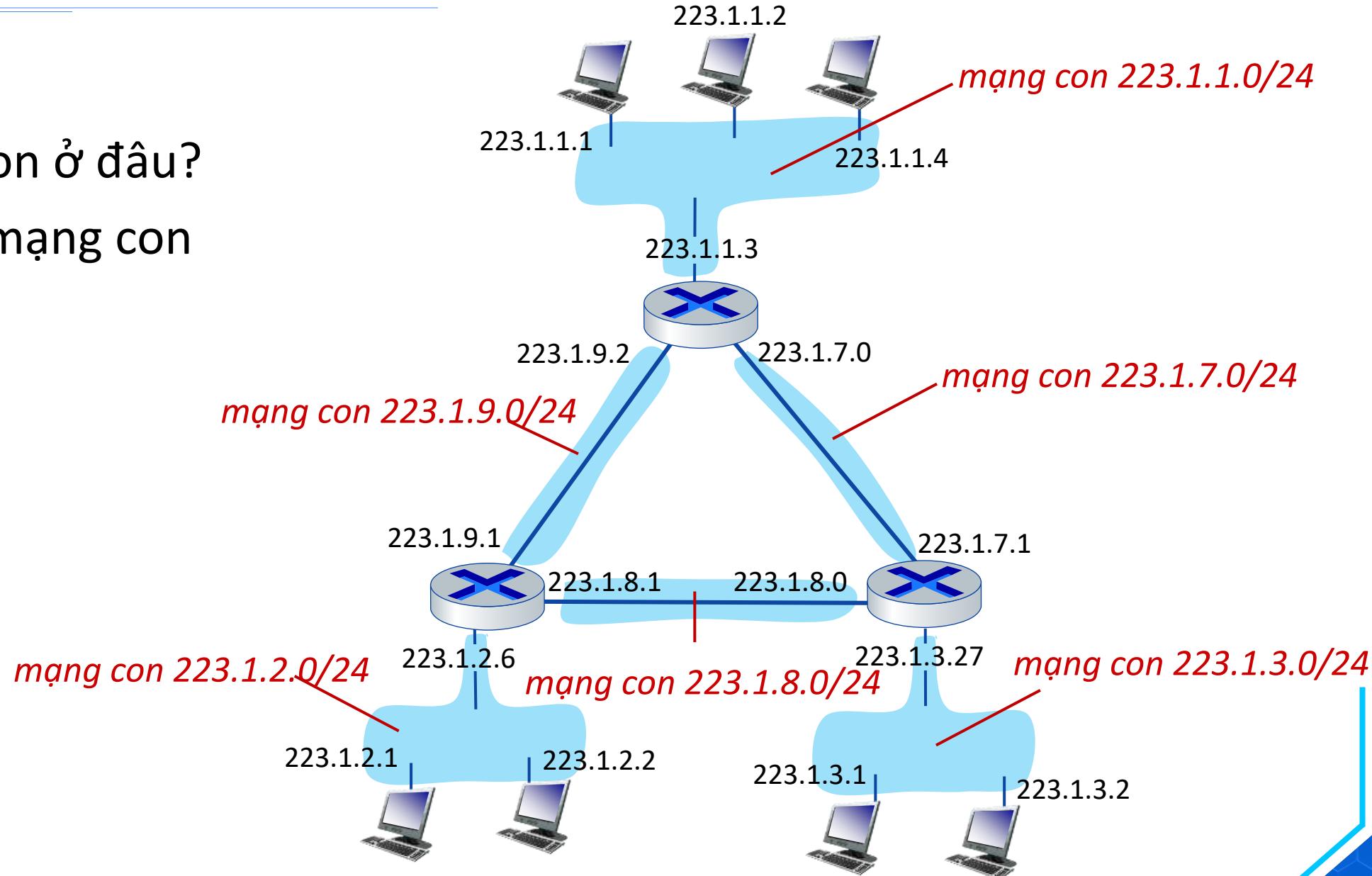


mặt nạ mạng con: /24
(24 bit bậc cao: phần mạng con của địa chỉ IP)



Mạng con

- Các mạng con ở đâu?
- /24 địa chỉ mạng con là gì?

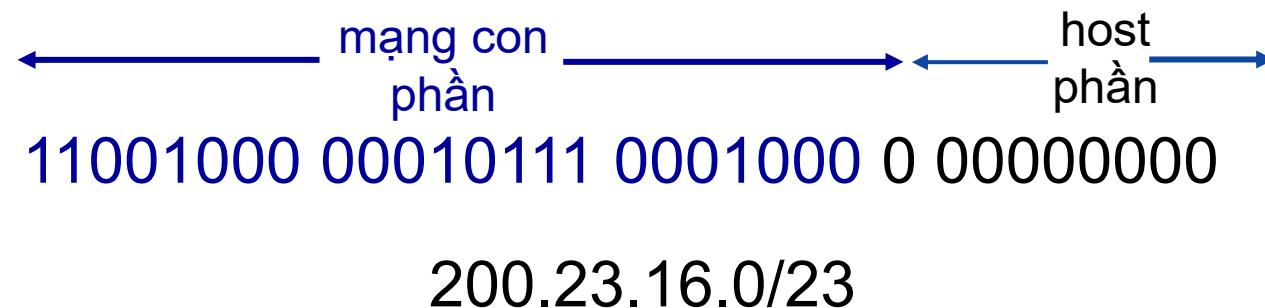




Địa chỉ IP: CIDR

CIDR: C lassless I nter D omain R outing (phát âm là “cider”)

- Phần mạng con của địa chỉ có độ dài tùy ý
- Định dạng địa chỉ: $abcd/x$, trong đó x là # bit trong phần mạng con của địa chỉ



Địa chỉ IP: làm thế nào để có được một địa chỉ?



Đó thực sự là **hai** câu hỏi:

1. Hỏi: Làm cách nào để *host* nhận được địa chỉ IP trong mạng của nó (phần địa chỉ host)?
2. Hỏi: Làm thế nào để một *mạng* có được địa chỉ IP cho chính nó (phần mạng của địa chỉ)

Làm thế nào để *host* có được địa chỉ IP?

- Được mã hóa cứng bởi sysadmin trong file cấu hình (ví dụ: /etc/rc.config trong UNIX)
- **DHCP: Dynamic Host Configuration Protocol:** tự động lấy địa chỉ từ máy chủ DHCP
 - “cắm là chạy”

DHCP: Giao thức cấu hình host động



Mục tiêu: host *tự động* lấy địa chỉ IP từ máy chủ mạng khi nó “tham gia” mạng

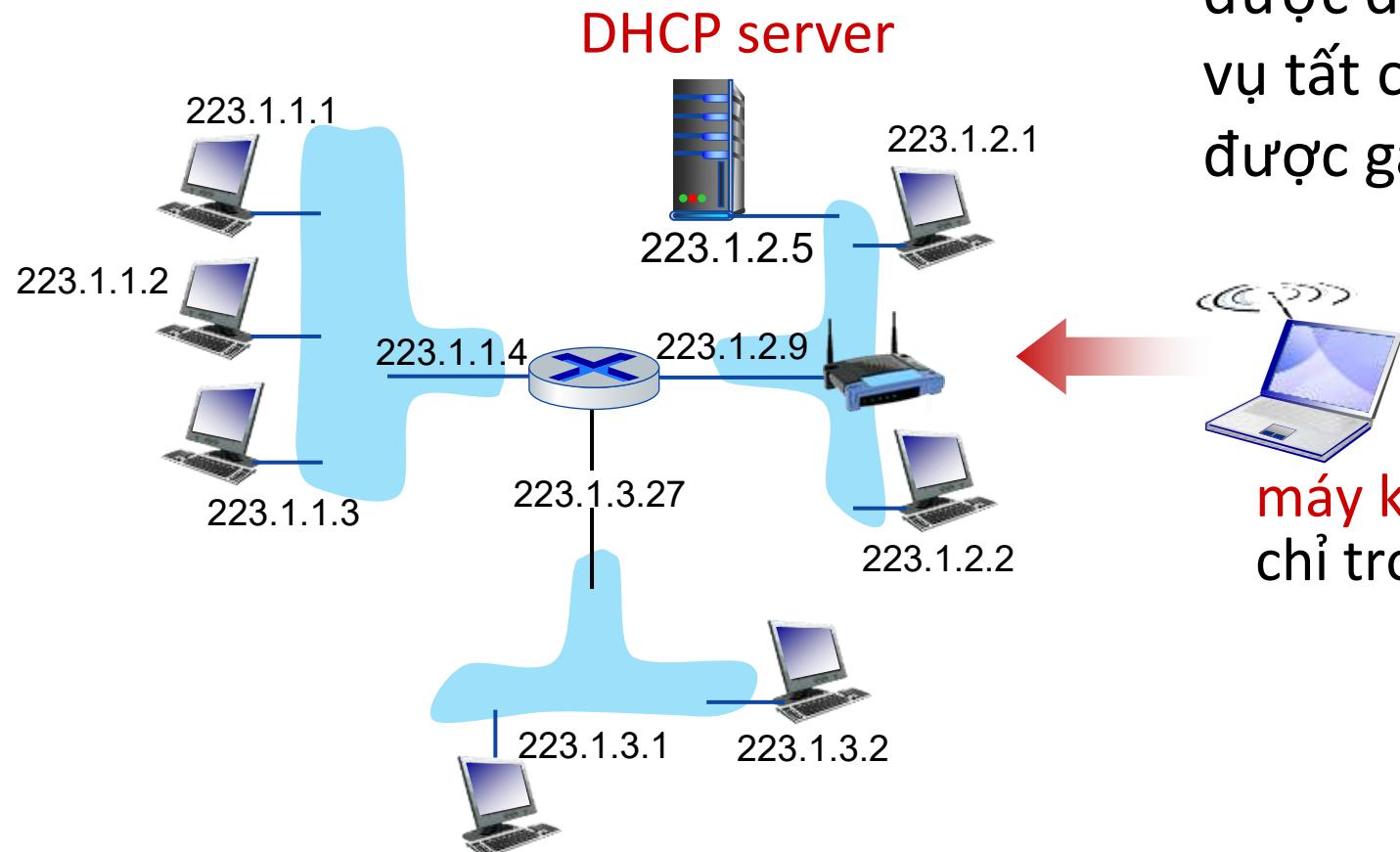
- Có thể giới hạn việc thuê địa chỉ đang sử dụng
- Cho phép sử dụng lại địa chỉ (chỉ giữ địa chỉ khi kết nối/ bật)
- Hỗ trợ cho người dùng di động tham gia/rời khỏi mạng

Tổng quan về DHCP:

- Host gửi quảng bá thông điệp **DHCP discover** [tùy chọn]
- Máy chủ DHCP phản hồi với thông điệp **DHCP offer** [tùy chọn]
- Host yêu cầu địa chỉ IP: thông điệp **DHCP request**
- Máy chủ DHCP gửi địa chỉ: thông điệp **DHCP ack**



Kịch bản DHCP client-server



Thông thường, máy chủ DHCP sẽ được đặt chung với router, phục vụ tất cả các mạng con mà router được gắn vào.

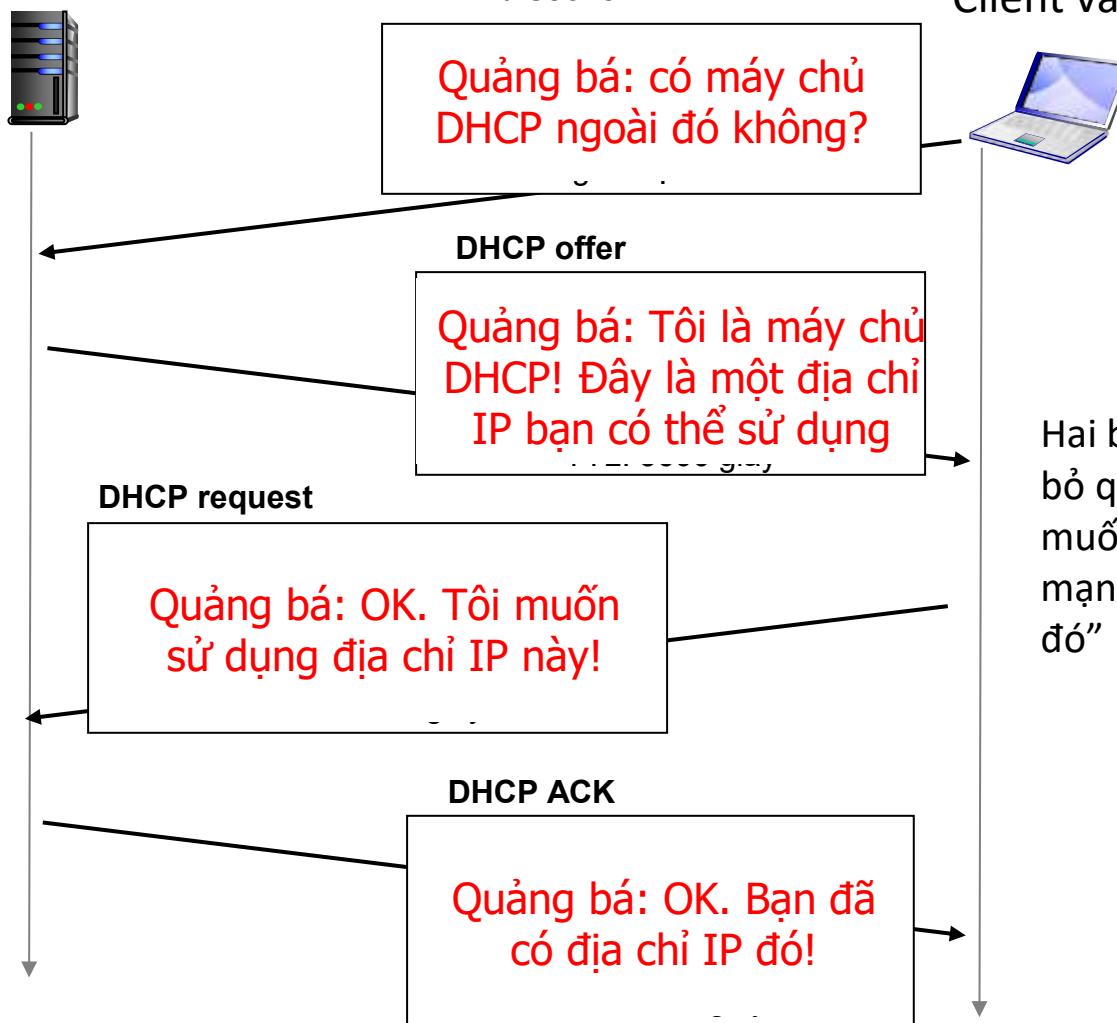


máy khách DHCP cần địa chỉ trong mạng này



Kịch bản DHCP client-server

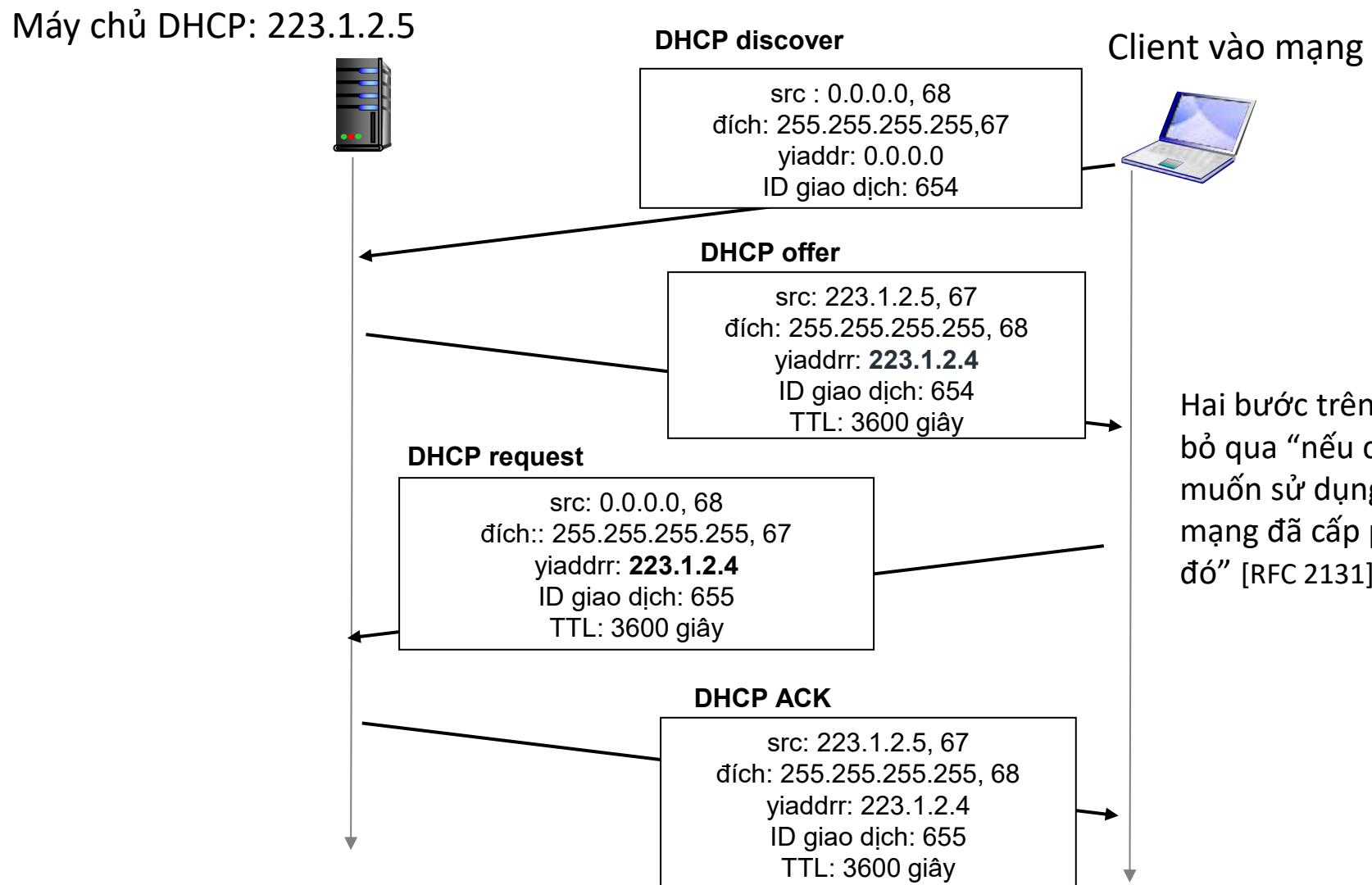
Máy chủ DHCP: 223.1.2.5 Client vào mạng



Hai bước trên có thể được bỏ qua “nếu client nhớ và muốn sử dụng lại địa chỉ mạng đã cấp phát trước đó” [RFC 2131]



Kịch bản DHCP client-server





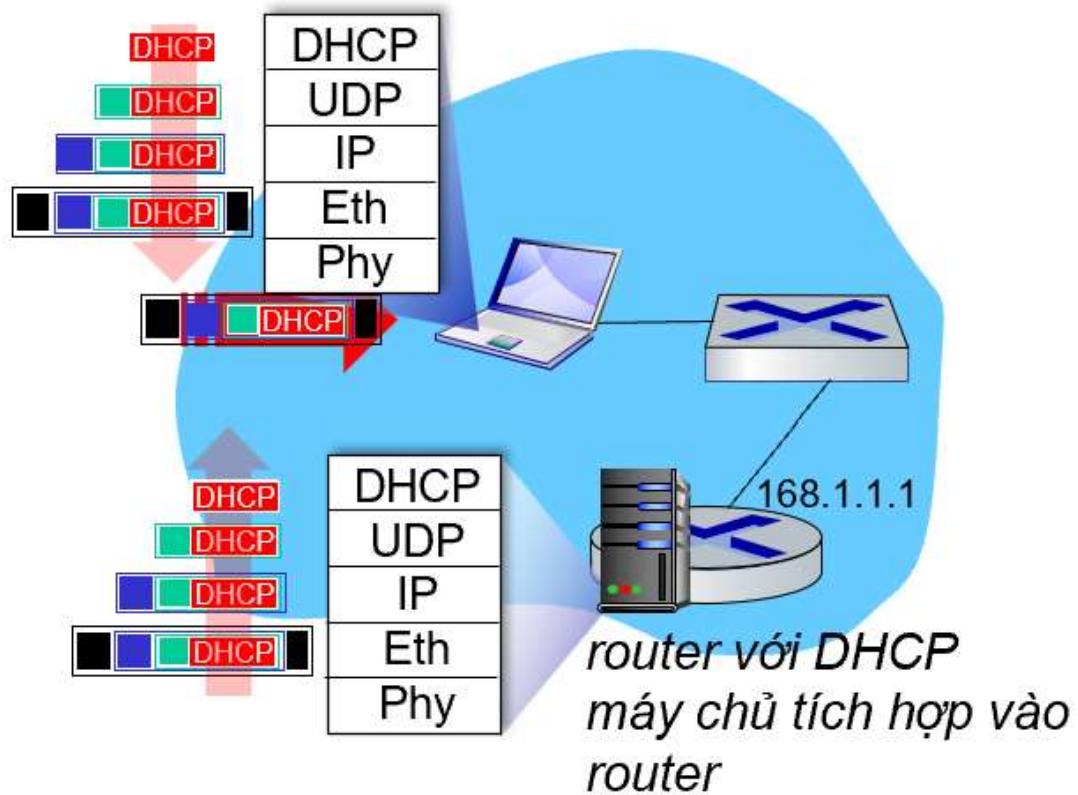
DHCP: không chỉ cung cấp địa chỉ IP

DHCP có thể cung cấp nhiều thông tin hơn ngoài địa chỉ IP được phân bổ trên mạng con:

- Địa chỉ first-hop của router (địa chỉ default gateway) cho client
- Tên và địa chỉ IP của máy chủ DNS
- Subnet mask (chỉ ra phần mạng và phần host của địa chỉ IP)

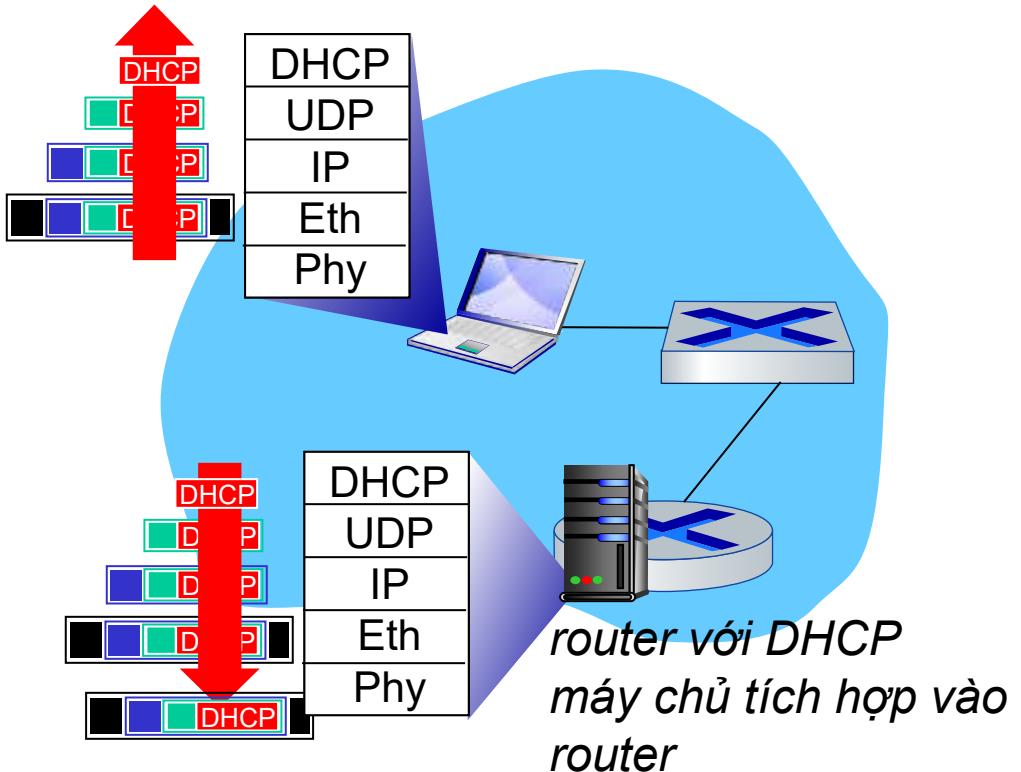


DHCP: ví dụ



- Laptop kết nối sẽ sử dụng DHCP để lấy địa chỉ IP, địa chỉ first-hop của router, địa chỉ của máy chủ DNS.
- Thông điệp DHCP REQUEST được đóng gói trong UDP, được đóng gói trong IP, được đóng gói trong Ethernet.
- Quảng bá frame Ethernet (đích: FFFFFFFFFFFF) trên mạng LAN, được nhận tại router chạy máy chủ DHCP.
- Ethernet được giải mã thành IP, giải mã UDP thành DHCP.

DHCP: ví dụ



- Máy chủ DHCP tạo DHCP ACK chứa địa chỉ IP của client, địa chỉ IP first-hop của router cho client, tên & địa chỉ IP của máy chủ DNS
- Phản hồi của máy chủ DHCP được đóng gói và được chuyển tiếp tới client, giải mã thành DHCP tại client
- Client bây giờ biết địa chỉ IP, tên và địa chỉ IP của máy chủ DNS, địa chỉ IP first-hop của router



Địa chỉ IP: làm thế nào để có được một địa chỉ?

Hỏi: Làm thế nào để mạng nhận được phần mạng con của địa chỉ IP?

Đáp: Được phân bổ một phần không gian địa chỉ của nhà cung cấp ISP

ISP's block

11001000 00010111 0001 0000 00000000 200.23.16.0/20

Sau đó, ISP có thể phân bổ không gian địa chỉ của mình thành 8 khối:

Tổ chức 0 11001000 00010111 0001000 0 00000000 200.23.16.0/23

Tổ chức 1 11001000 00010111 0001001 0 00000000 200.23.18.0/23

Tổ chức 2 11001000 00010111 0001010 0 00000000 200.23.20.0/23

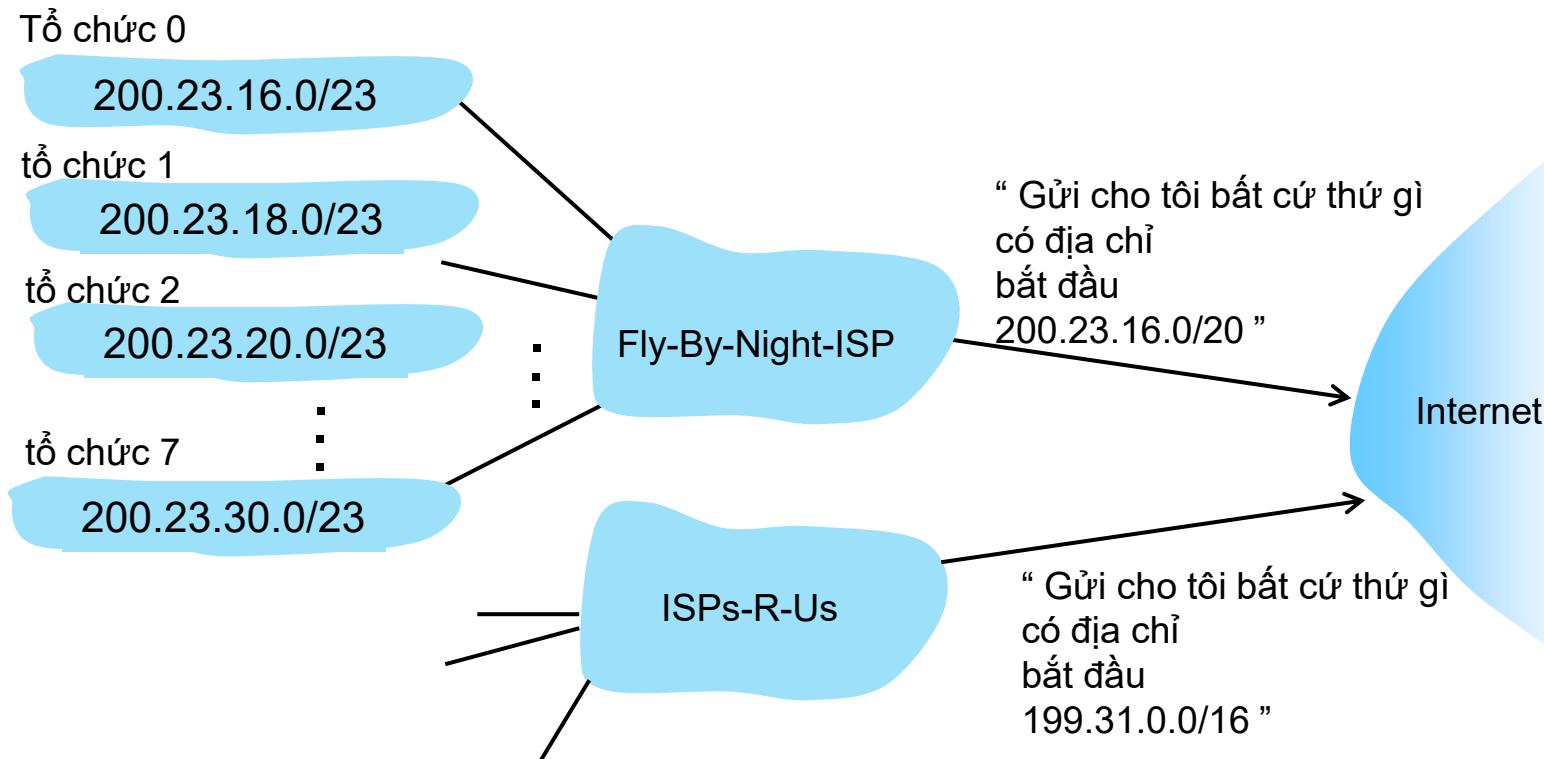
.....

Tổ chức 7 11001000 00010111 0001111 0 00000000 200.23.30.0/23

Địa chỉ phân cấp: tập hợp tuyến đường



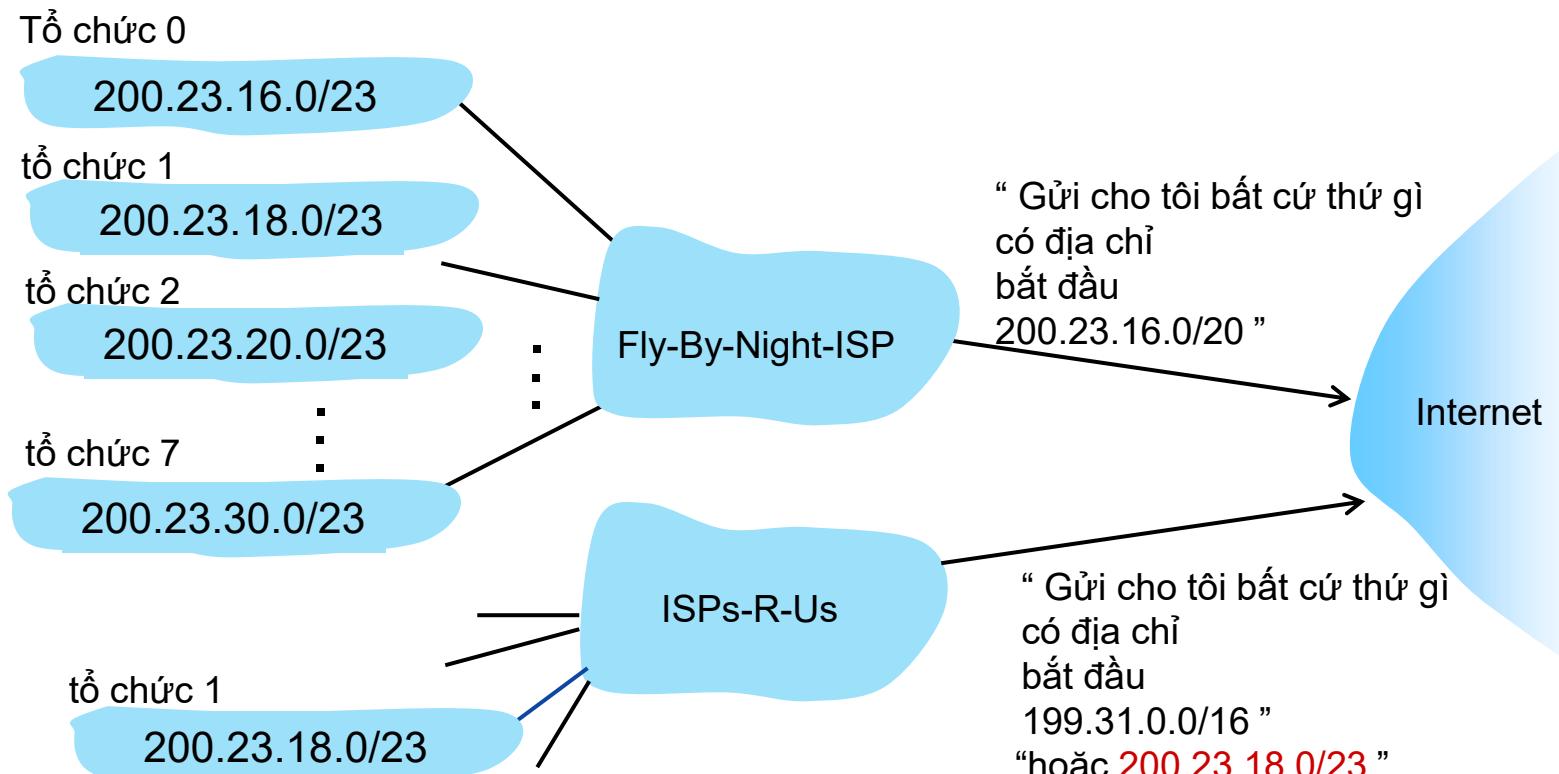
Địa chỉ phân cấp cho phép quảng cáo thông tin định tuyến hiệu quả:



Địa chỉ phân cấp : các tuyển cù thể hơn



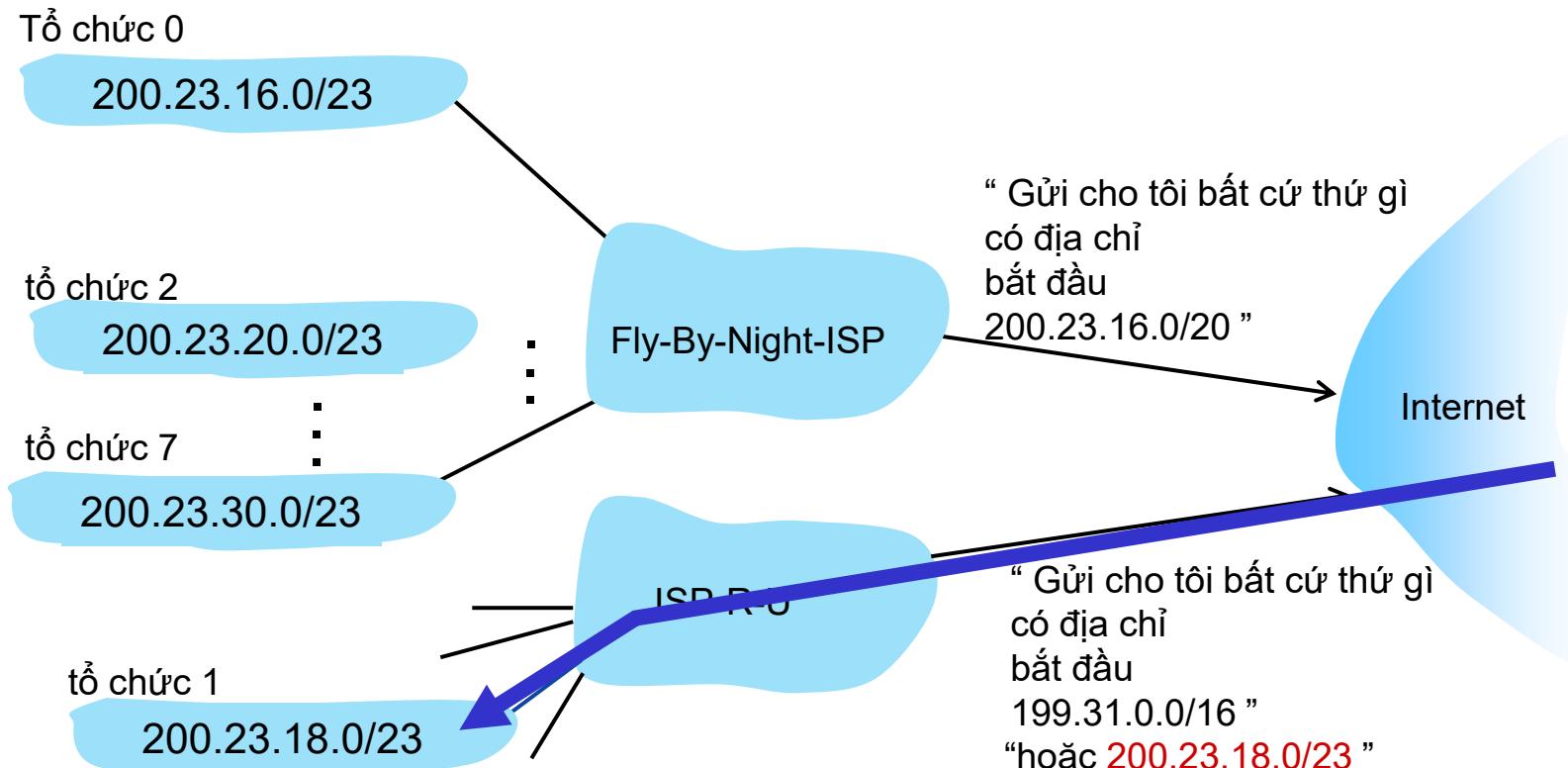
- Tổ chức 1 chuyển từ Fly-By-Night-ISP sang ISPs-R-U
- Các ISP-R-U hiện quảng cáo một lô trình cù thể hơn tới Tổ chức 1



Địa chỉ phân cấp : các tuyển cù thể hơn



- Tổ chức 1 chuyển từ Fly-By-Night-ISP sang ISPs-R-U
- Các ISP-R-U hiện quảng cáo một lô trình cù thể hơn tới Tổ chức 1





Địa chỉ IP: cuối cùng ...

Hỏi: làm cách nào để một ISP có được khối địa chỉ?

Đáp: ICANN : Internet Corporation for Assigned Names and Numbers

<http://www.icann.org/>

- phân bổ địa chỉ IP, thông qua **5 cơ quan đăng ký khu vực (RRs)** (những người sau đó có thể phân bổ cho các cơ quan đăng ký địa phương)
- quản lý vùng gốc DNS, bao gồm ủy quyền quản lý TLD riêng lẻ (.com, .edu, ...)

Hỏi: có đủ địa chỉ IP 32 bit không?

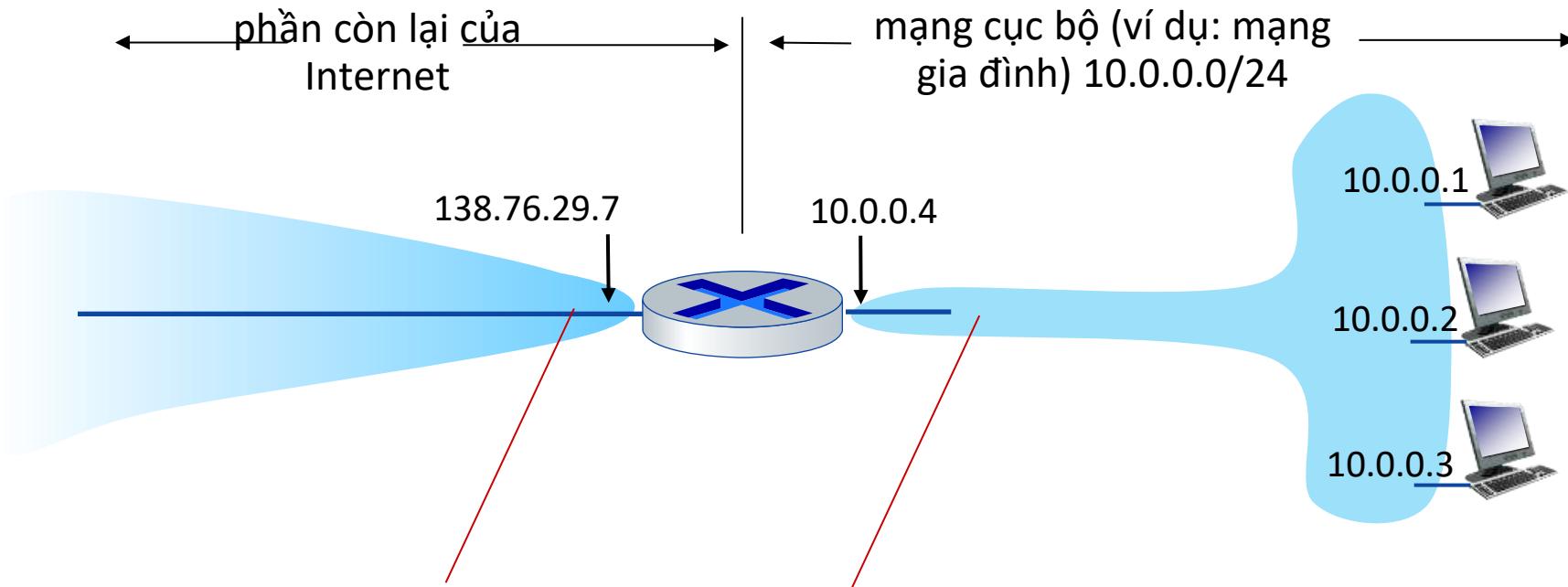
- ICANN đã phân bổ đoạn địa chỉ IPv4 cuối cùng cho các RRs vào năm 2011
- NAT (tiếp theo) giải quyết việc cạn kiệt không gian địa chỉ IPv4
- IPv6 có không gian địa chỉ 128-bit

*"Ai biết chúng ta cần bao nhiêu không gian địa chỉ?" Vint Cerf
(phản ánh quyết định tạo địa chỉ IPv4 dài 32 bit)*

NAT: Network Address Translation (chuyển dịch địa chỉ mạng)



NAT: tất cả các thiết bị trong mạng cục bộ chia sẻ **một** địa chỉ IPv4 khi có liên quan đến thế giới bên ngoài



tất cả các datagram *rời khỏi* mạng cục bộ có **cùng** địa chỉ IP NAT nguồn: 138.76.29.7, nhưng số cổng nguồn **khác nhau**

datagram có nguồn hoặc đích trong mạng này có địa chỉ 10.0.0.0/24 cho nguồn, đích (như thường lệ)

NAT: Network Address Translation (chuyển dịch địa chỉ mạng)



- Tất cả các thiết bị trong mạng cục bộ đều có địa chỉ 32 bit trong không gian địa chỉ IP “riêng tư” (tiền tố 10/8, 172.16/12, 192.168/16) chỉ có thể được sử dụng trong mạng cục bộ.
- Thuận lợi:
 - Chỉ cần **một** địa chỉ IP từ ISP của nhà cung cấp cho *tất cả* các thiết bị
 - Có thể thay đổi địa chỉ của host 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 của các thiết bị trong mạng nội bộ
 - Bảo mật: các thiết bị bên trong mạng cục bộ không thể định địa chỉ trực tiếp mà thế giới bên ngoài có thể nhìn thấy

NAT: Network Address Translation (chuyển dịch địa chỉ mạng)



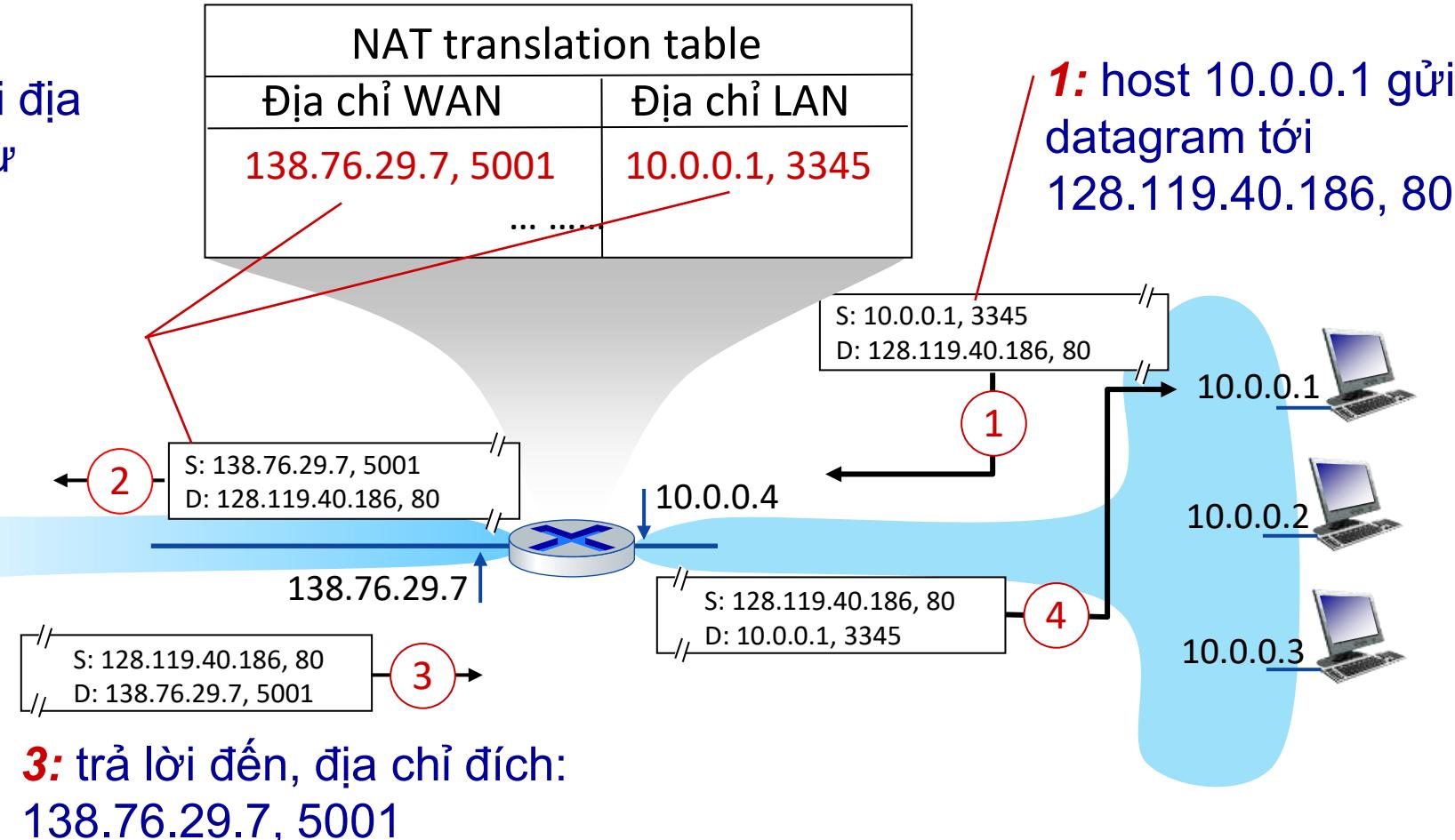
Triển khai: router NAT phải trong suốt:

- Các datagram gửi đi: thay thế (địa chỉ IP nguồn, cổng #) của mọi datagram gửi đi thành (địa chỉ NAT IP, cổng # mới)
 - Máy khách/máy chủ từ xa sẽ phản hồi bằng cách sử dụng (địa chỉ NAT IP, số cổng mới) làm địa chỉ đích
- Ghi nhớ (trong bảng dịch NAT) mọi cặp ánh xạ (địa chỉ IP nguồn, cổng #) sang (địa chỉ NAT IP, cổng # mới)
- Các datagram đến: thay thế (địa chỉ IP NAT, số cổng mới) trong các trường đích của mọi gói dữ liệu đến bằng (địa chỉ IP nguồn, cổng #) tương ứng được lưu trữ trong bảng NAT

NAT: Network Address Translation (chuyển dịch địa chỉ mạng)



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



NAT: Network Address Translation (chuyển dịch địa chỉ mạng)



- NAT đã gây tranh cãi:
 - Router “nên” chỉ xử lý tối đa lớp 3
 - Địa chỉ “thiếu” cần được giải quyết bằng IPv6
 - Vi phạm đối số end-to-end (thao tác cổng # bởi thiết bị lớp mạng)
 - Truyền tải NAT: nếu máy khách muốn kết nối với máy chủ sau NAT thì sao?
- Nhưng NAT vẫn ở đây:
 - Được sử dụng rộng rãi trong mạng gia đình và tổ chức, mạng di động 4G/5G

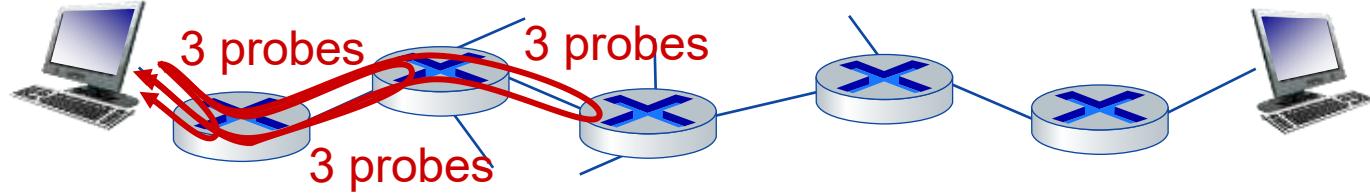


ICMP (Internet Control Message Protocol)

	<u>Loại</u>	<u>mã</u>	<u>Mô tả</u>
o Được sử dụng bởi các host và router để truyền thông tin tầng network	0	0	echo reply (ping)
■ Thông báo: host, mạng, cổng, giao thức không thể truy cập	3	0	dest. network unreachable
■ Phản hồi Request/Reply (được sử dụng bởi lệnh Ping)	3	1	dest host unreachable
■ Phản hồi Request/Reply (được sử dụng bởi lệnh Ping)	3	2	dest protocol unreachable
■ Phản hồi Request/Reply (được sử dụng bởi lệnh Ping)	3	3	dest port unreachable
■ Phản hồi Request/Reply (được sử dụng bởi lệnh Ping)	3	6	dest network unknown
■ Phản hồi Request/Reply (được sử dụng bởi lệnh Ping)	3	7	dest host unknown
o Tầng network “ở trên”:	4	0	source quench (congestion control - not used)
■ Các thông điệp ICMP được mang trong các IP datagram	8	0	echo request (ping)
■ Các thông điệp ICMP được mang trong các IP datagram	9	0	route advertisement
o <i>Thông báo ICMP</i> : loại, mã cộng với 8 byte đầu tiên của IP datagram gây ra lỗi	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
 - Segment thứ nhất có giá trị TTL =1,
 - Segment thứ hai có giá trị TTL=2,
 - V.v.
- Khi datagram thứ n đến router thứ n :
 - Router huỷ datagram và gửi thông điệp ICMP đến nguồn (loại 11, mã 0)
 - Thông điệp ICMP có thể bao gồm tên và địa chỉ IP của router
- Khi thông điệp ICMP đến, nguồn ghi lại giá trị RTTs
 - Tiêu chí dừng:
 - Segment UDP lần lượt đến tới host đích
 - Đích trả về thông điệp ICMP “port unreachable” (loại 3, mã 3)
 - Nguồn dừng lại



IPv6: động lực

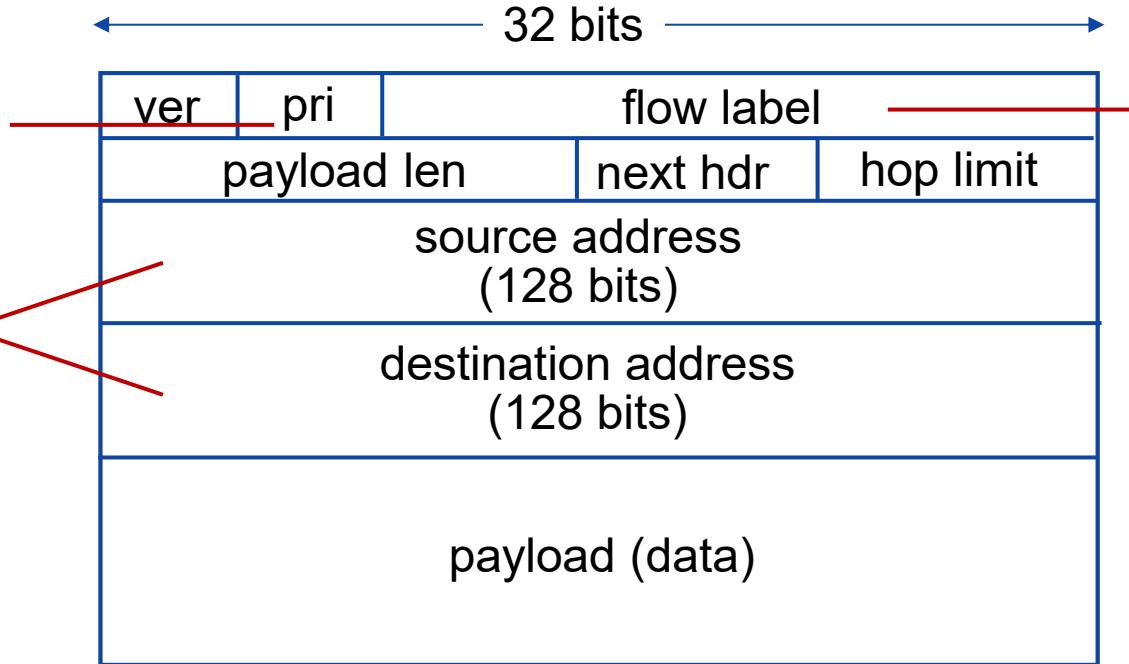
- **Động lực ban đầu:** Không gian địa chỉ IPv4 32 bit sẽ được phân bổ hoàn toàn
- **Động lực bổ sung:**
 - Tốc độ xử lý/chuyển tiếp: header có độ dài cố định 40 byte
 - Cho phép xử lý các “luồng” ở tầng mạng khác nhau



Định dạng datagram IPv6

priority: xác định mức độ ưu tiên giữa các datagram trong luồng

128-bit
địa chỉ IPv6



nhãn luồng: xác định các datagram trong cùng một “luồng”. (khái niệm về “luồng” không được định nghĩa rõ ràng)

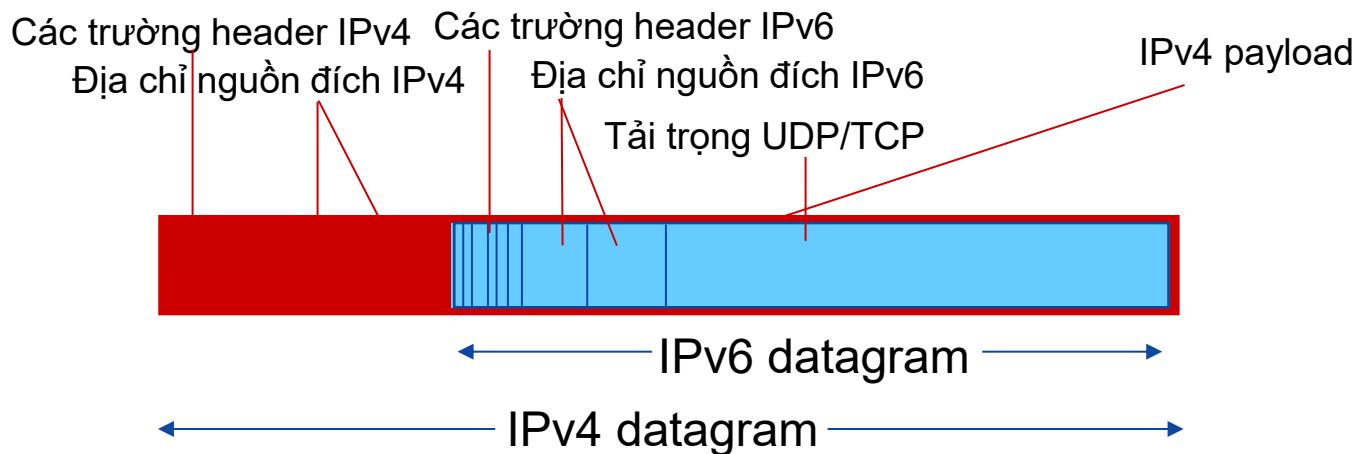
Những gì còn thiếu (so sánh với IPv4):

- Không có tính checksum (để tăng tốc độ xử lý tại các router)
- Không phân mảnh/tổng hợp
- Không có tùy chọn (có sẵn dưới dạng giao thức lớp trên, header kế tiếp tại router)



Chuyển đổi từ IPv4 sang IPv6

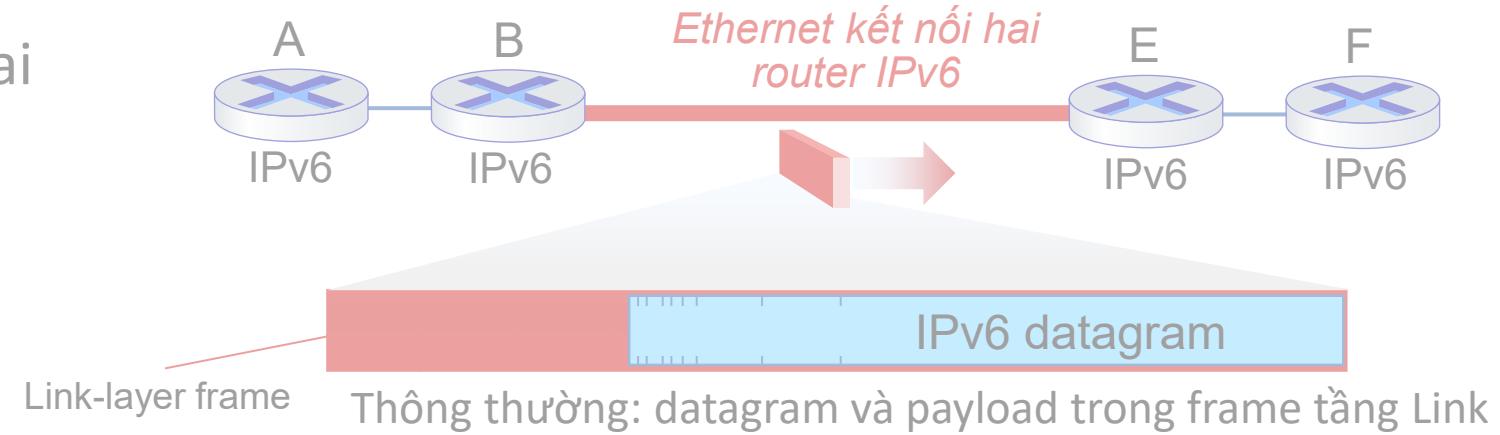
- Không phải tất cả các router đều có thể được nâng cấp đồng thời
 - Mạng sẽ hoạt động như thế nào với các router IPv4 và IPv6 hỗn hợp?
- **Đường hầm (tunneling):** gói dữ liệu IPv6 được mang dưới dạng *payload* trong gói dữ liệu IPv4 giữa các router IPv4 ("packet trong packet")
 - Đường hầm được sử dụng rộng rãi trong các ngữ cảnh khác (4G/5G)



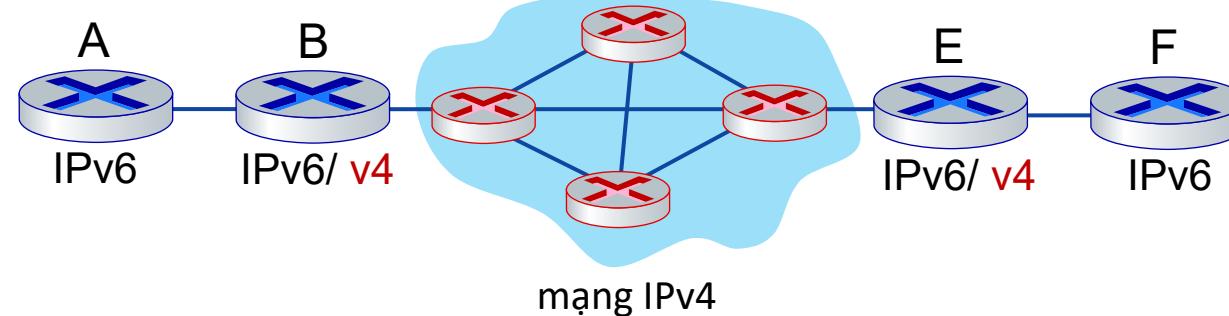


Đường hầm và đóng gói

Ethernet kết nối hai router IPv6:



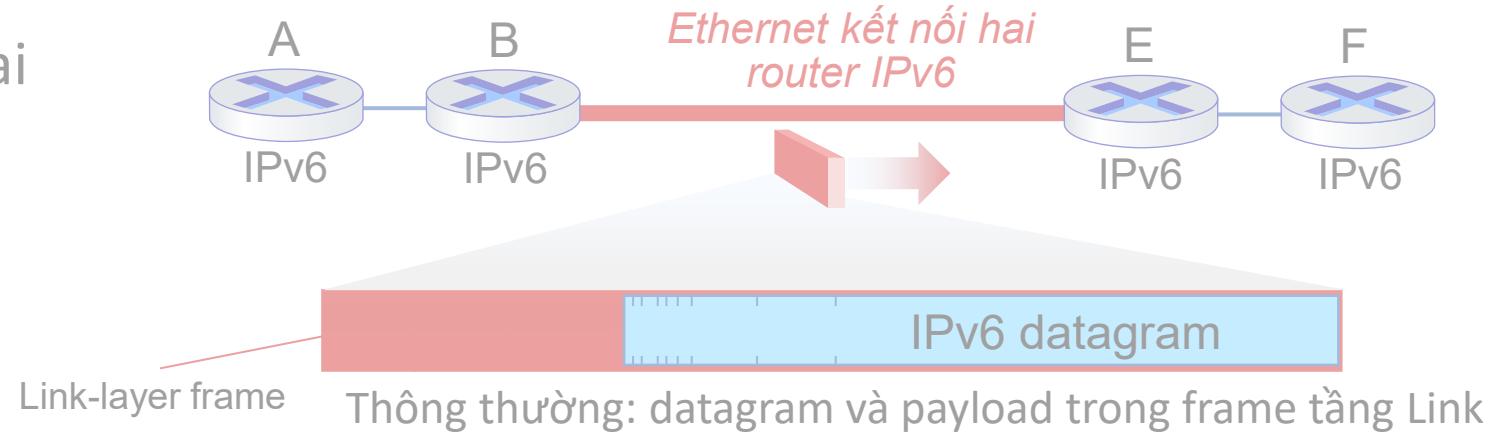
Mạng IPv4 kết nối hai router IPv6



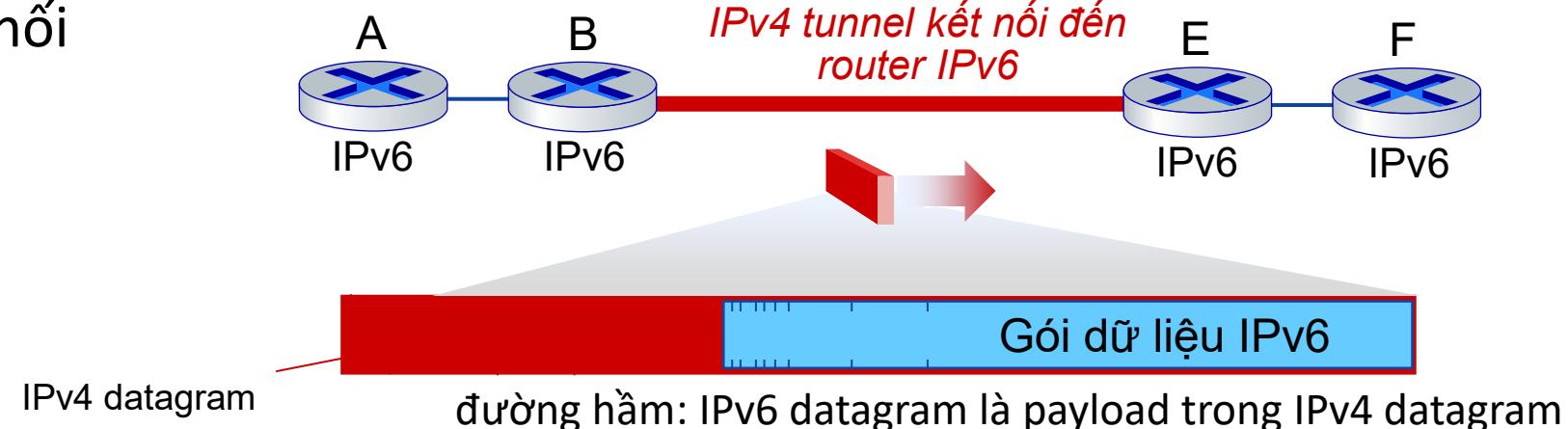


Đường hầm và đóng gói

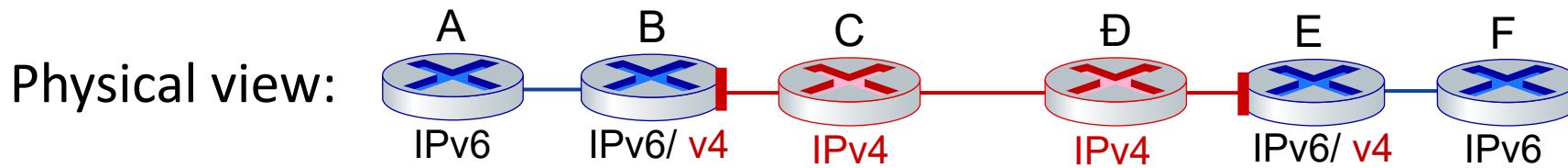
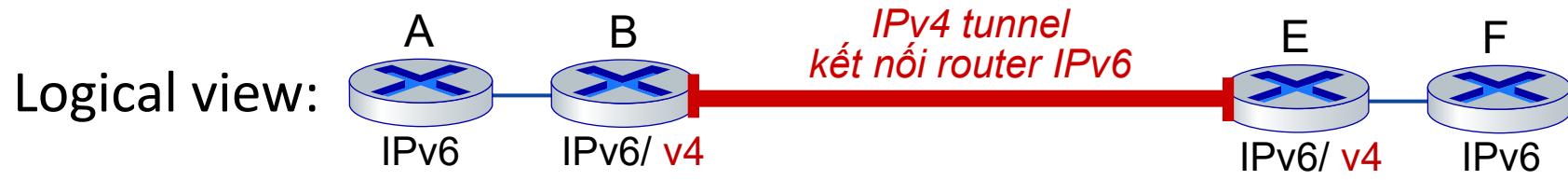
Ethernet kết nối hai router IPv6:



Mạng IPv4 kết nối hai router IPv6



Đường hầm



Lưu ý địa chỉ
nguồn và đích!

A đến B:
IPv6

từ B đến C:
IPv6 trong
IPv4

từ B đến C:
IPv6 trong
IPv4

từ B đến C:
IPv6 trong
IPv4

E đến F:
IPv6



Chương 4: Nội dung

4.1 Tổng quan

- Mức dữ liệu
- Mức điều khiển

4.2 Virtual circuit network và datagram network

- Các mạch ảo
- Mạng datagram

4.3 Cấu trúc bên trong router

- Cổng đầu vào, chuyển đổi, cổng đầu ra
- Quản lý bộ đệm, lập kế hoạch

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing

- DHCP
- NAT
- ICMP
- IPv6

4.5 Các thuật toán routing

- Link state
- Distance vector

4.6 Routing trong Internet

- Hierarchical routing
- RIP, OSPF, BGP
- SDN
- Quản lý mạng

4.7 Broadcast và Multicast routing

Giới thiệu



- Giao thức định tuyến
 - Link state
 - Distance vector
- Định tuyến nội bộ ISP: OSPF
- Định tuyến giữa các ISP: BGP
- Mức điều khiển SDN
- Quản lý mạng, cấu hình
 - SNMP
 - NETCONF/YANG





Chức năng tầng mạng

- **Chuyển tiếp:** di chuyển các gói từ đầu vào của router đến đầu ra router thích hợp
- **Định tuyến:** xác định tuyến đường được thực hiện bởi các gói từ nguồn đến đích

*Mức dữ liệu và
Mức điều khiển*

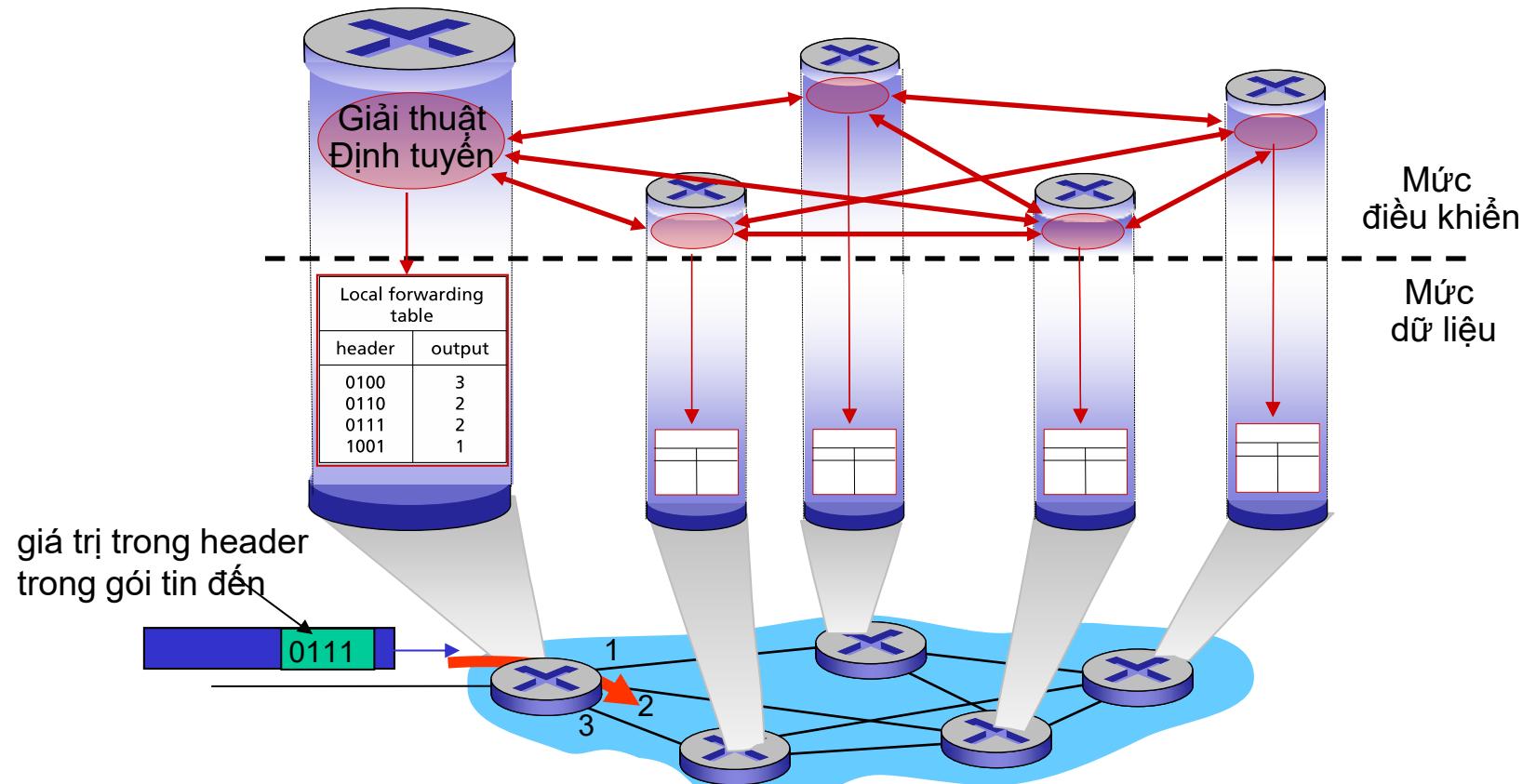
Hai cách tiếp cận để cấu trúc việc điều khiển mạng:

- Điều khiển trên mỗi router (truyền thống)
- Điều khiển tập trung hợp lý (mạng được xác định bằng phần mềm)



Mức điều khiển mạng tại mỗi router

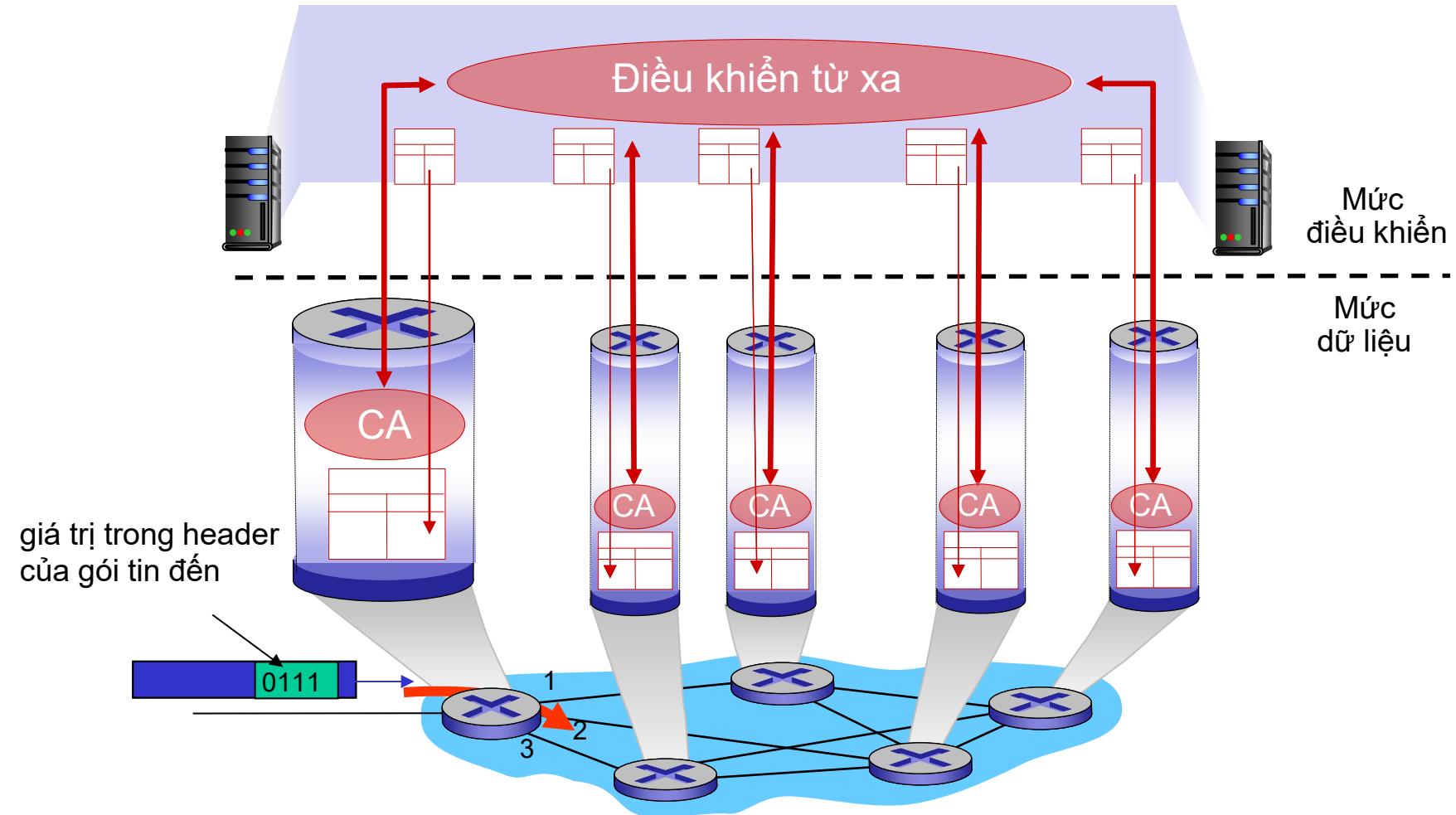
Các thành phần thuật toán định tuyến riêng lẻ *trong mỗi và mọi router* tương tác trong mức điều khiển



Mức điều khiển mạng được xác định bằng phần mềm (SDN)



Bộ điều khiển từ xa tính toán, cài đặt bảng chuyển tiếp trong router

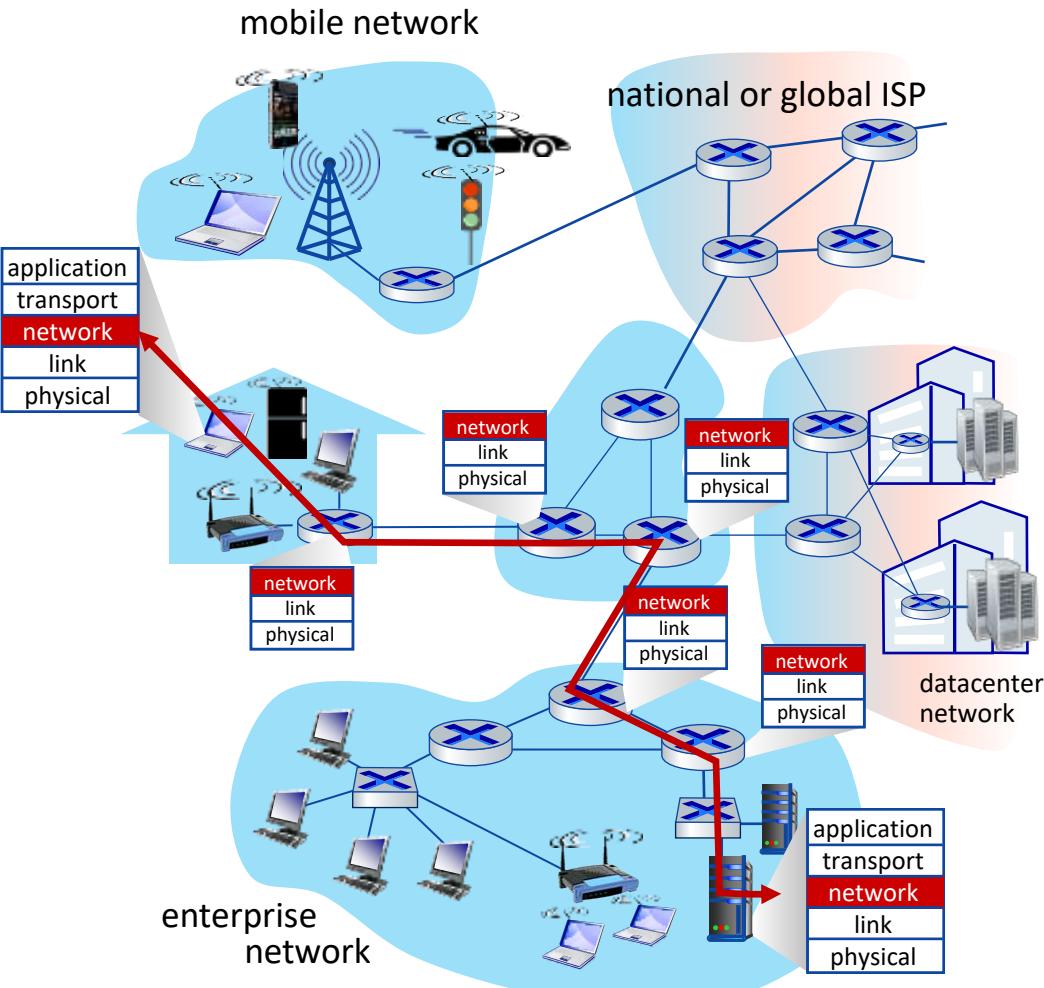


Giao thức định tuyến



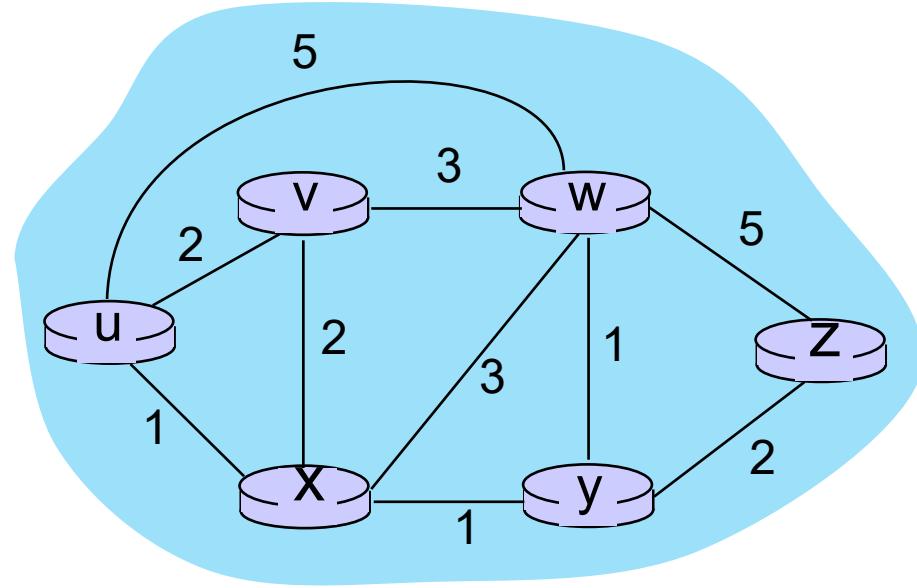
Định tuyến: xác định các đường đi “tốt” từ host gửi đến host nhận, thông qua mạng các router

- **Đường đi:** chuỗi các router đi từ host nguồn ban đầu đi đến host đích cuối cùng
- “tốt”: “chi phí” thấp nhất, “nhanh nhất”, “ít tắc nghẽn nhất”
- Định tuyến: thách thức kết nối mạng “top 10”!





Trừu tượng hóa đồ thị: chi phí liên kết



$c_{a,b}$: chi phí liên kết *trực tiếp* kết nối a và b
ví dụ: $c_{w,z} = 5, c_{u,z} = \infty$

chi phí do nhà điều hành mạng xác định: có thể luôn bằng 1 hoặc tỷ lệ nghịch với băng thông hoặc liên quan tới tắc nghẽn

đồ thị: $G = (N, E)$

N : router = { u, v, w, x, y, z }

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



Phân loại thuật toán định tuyến

nhanh
như thế
nào ?

tĩnh: tuyến đường thay đổi
chậm theo thời gian

toàn cầu: tất cả các router có cấu trúc
liên kết hoàn chỉnh, thông tin chi phí liên
kết

- thuật toán “Link state”

động : tuyến đường thay đổi
nhanh hơn

- cập nhật định kỳ hoặc để
đáp ứng với thay đổi chi phí
liên kết

phi tập trung: quá trình tính toán lặp đi lặp lại, trao
đổi thông tin với hàng xóm

- router ban đầu chỉ biết chi phí liên kết đến các
hang xóm được đính kèm
- thuật toán “Distance vector”

thông tin toàn cầu hay phi tập trung?



Thuật toán định tuyến Dijkstra's link-state

- Tập trung: tất cả các nút đều biết cấu trúc liên kết mạng và chi phí liên kết
 - Được thực hiện thông qua gói tin “link state broadcast”
 - Tất cả các nút có cùng thông tin
- Tính toán các đường đi có chi phí thấp nhất từ một nút (“nguồn”) đến tất cả các nút khác
 - Cung cấp *bảng chuyển tiếp* cho nút đó
- Lặp đi lặp lại: sau k lần lặp lại, biết đường đi chi phí thấp nhất đến k đích

ký hiệu

- $c_{x,y}$: chi phí liên kết trực tiếp từ nút x đến y ; $= \infty$ nếu không phải hàng xóm trực tiếp
- $D(v)$: *ước tính chi phí hiện tại* của đường đi có chi phí thấp nhất từ nguồn đến đích v
- $p(v)$: nút trước dọc theo đường đi từ nguồn đến v
- N' : tập hợp các nút có đường đi chi phí thấp nhất được biết



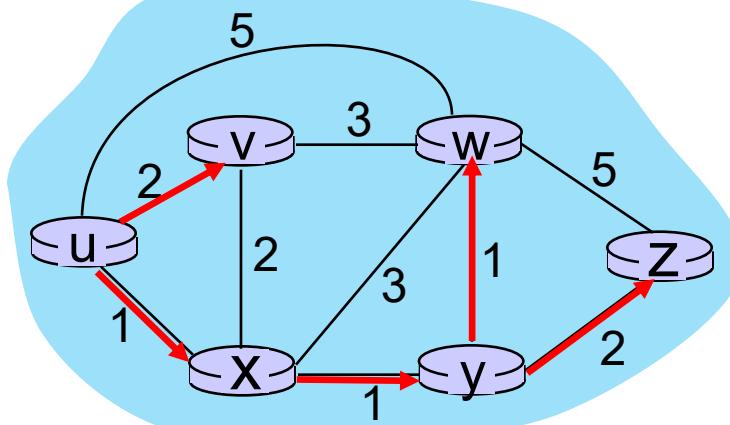
Thuật toán định tuyến Dijkstra's link-state

```
1 Initialization:
2    $N' = \{u\}$                                 /* compute least cost path from u to all other nodes */
3   for all nodes  $v$ 
4     if  $v$  adjacent to  $u$                       /*  $u$  initially knows direct-path-cost only to direct neighbors */
5       then  $D(v) = c_{u,v}$                       /* but may not be minimum cost!
6     else  $D(v) = \infty$ 
7
8 Loop
9   find  $w$  not in  $N'$  such that  $D(w)$  is a minimum
10  add  $w$  to  $N'$ 
11  update  $D(v)$  for all  $v$  adjacent to  $w$  and not in  $N'$ :
12     $D(v) = \min ( D(v), D(w) + c_{w,v} )$ 
13  /* new least-path-cost to  $v$  is either old least-cost-path to  $v$  or known
14  least-cost-path to  $w$  plus direct-cost from  $w$  to  $v$  */
15  until all nodes in  $N'$ 
```



Thuật toán Dijkstra: ví dụ

Step	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	u, x	2, u	4, x	2, x	∞	∞
2	u, x, y	2, u	3, y	∞	4, y	∞
3	u, x, y, v	∞	3, y	∞	4, y	∞
4	u, x, y, v, w	∞	∞	∞	4, y	∞
5	u, x, y, v, w, z	∞	∞	∞	∞	∞



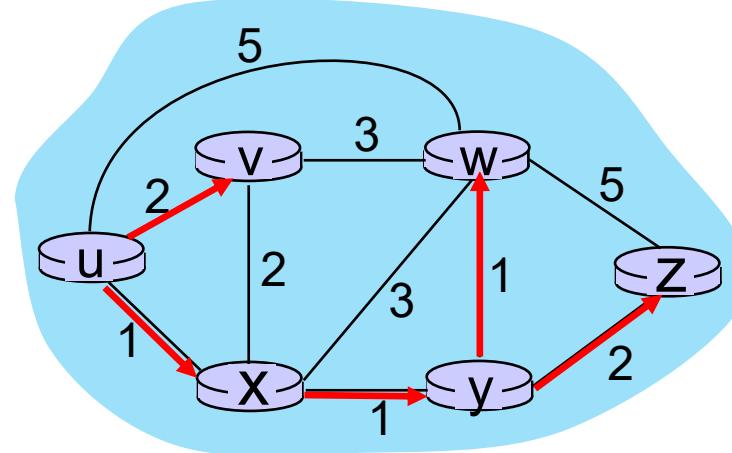
Initialization (step 0): For all a : if a adjacent to then $D(a) = c_{u,a}$

↓
 find a not in N' such that $D(a)$ is a minimum
 add a to N'
 update $D(b)$ for all b adjacent to a and not in N' :

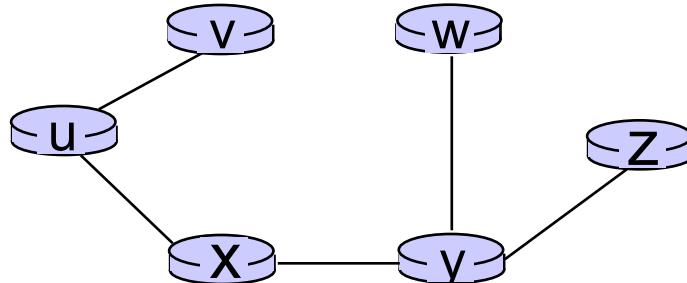
$$D(b) = \min (D(b), D(a) + c_{a,b})$$



Thuật toán Dijkstra: ví dụ



Kết quả cây đường dẫn chi phí thấp nhất từ u:



Bảng chuyển tiếp kết quả trong u:

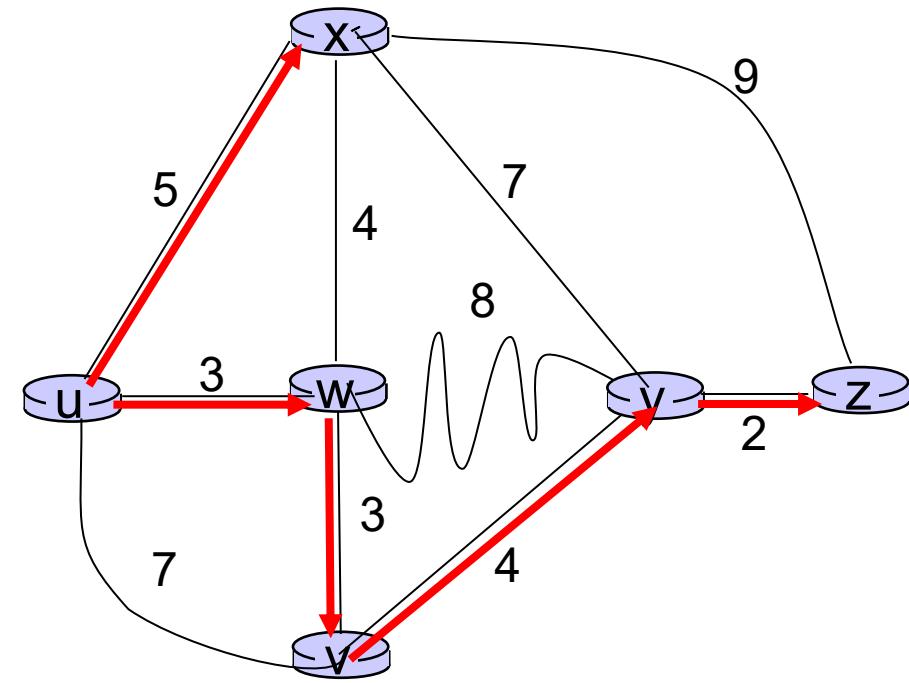
điểm đến	liên kết đi
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
x	(u,x)

tuyến đường từ u đến v trực tiếp
định tuyến từ u đến tất cả các đích khác qua x



Thuật toán Dijkstra: một ví dụ khác

Step	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	∞	∞
1	uw	6, w	5, u	11, w	∞	
2	uwx	6, w		11, w	14, x	
3	uwxv			10, v	14, x	
4	uwxvy				12, y	
5	uwxvzy					



Ghi chú:

- Xây dựng cây đường đi chi phí thấp nhất bằng cách truy tìm các nút trước

Thuật toán Dijkstra: thảo luận



Độ phức tạp của thuật toán: n nút

- Mỗi lần lặp n : cần kiểm tra tất cả các nút w không có trong N
- $n(n + 1)/2$: độ phức tạp $O(n^2)$
- Triển khai hiệu quả hơn có thể: $O(n \log n)$

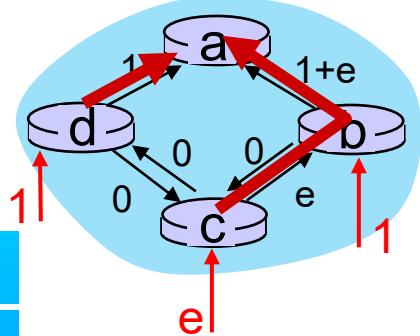
Độ phức tạp của thông điệp:

- Mỗi router phải *phát* thông tin trạng thái liên kết của nó tới n router khác
- Các thuật toán quảng bá hiệu quả (và thú vị!): $O(n)$ liên kết chéo để phổ biến một thông điệp quảng bá từ một nguồn
- Mỗi thông báo của router đi qua các liên kết $O(n)$: độ phức tạp của thông báo tổng thể: $O(n^2)$

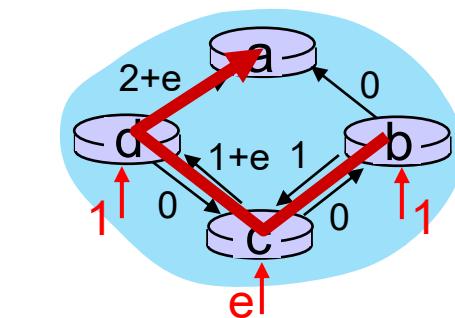


Thuật toán Dijkstra: khả năng dao động

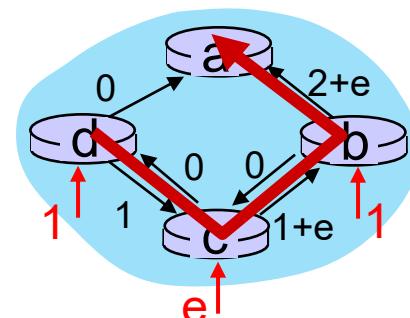
- Khi chi phí liên kết phụ thuộc vào lưu lượng mạng, việc định tuyến có thể thay đổi
- Kịch bản mẫu:
 - Định tuyến đến đích a, lưu lượng đi vào tại d, c, e với tốc độ 1, e (<1), 1
 - Chi phí liên kết là định hướng và phụ thuộc vào khối lượng



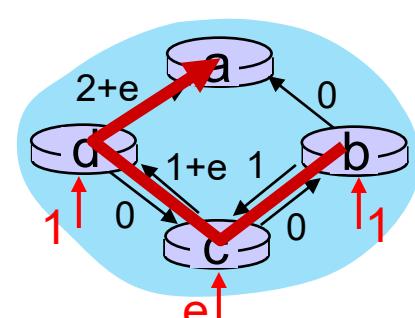
ban đầu



với những chi phí
này,
tìm định tuyến mới....
dẫn đến chi phí mới



với những chi phí này,
tìm định tuyến mới....
dẫn đến chi phí mới



với những chi phí này,
tìm định tuyến mới....
dẫn đến chi phí mới



Thuật toán vectơ khoảng cách (Distance vector)

Dựa trên công thức *Bellman-Ford* (BF) (lập trình động):

Công thức Bellman-Ford

Đặt $D_x(y)$: chi phí của đường đi có chi phí thấp nhất từ x đến y .

Sau đó:

$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

*min bao gồm tất cả
các lân cận v của x*

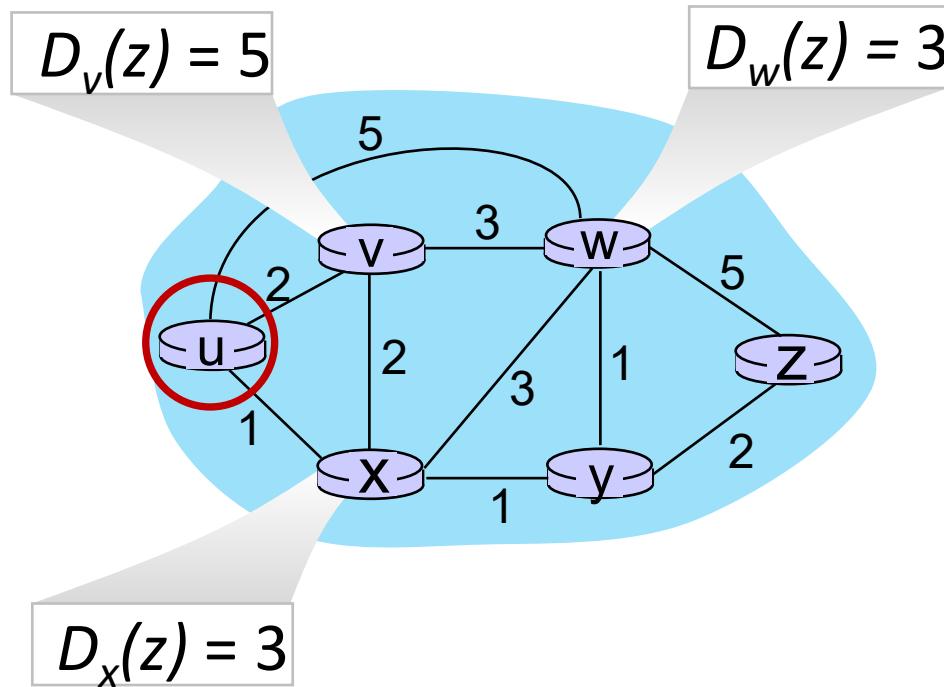
chi phí từ nút v đến y

chi phí trực tiếp của liên kết từ x đến v



Ví dụ Bellman-Ford

Giả sử rằng nút lân cận của u là s, x, v, w , biết rằng đối với đích z :



Công thức Bellman-Ford nói:

$$\begin{aligned} D_u(z) &= \min \{ c_{u,v} + D_v(z), \\ &\quad c_{u,x} + D_x(z), \\ &\quad c_{u,w} + D_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

nút đạt được giá trị cực tiểu (x) là bước nhảy tiếp theo trên đường ước tính có chi phí thấp nhất tới đích (z)



Thuật toán vectơ khoảng cách

Ý tưởng chính:

- Theo thời gian, mỗi nút sẽ gửi ước tính vectơ khoảng cách của nó cho các nút lân cận
- Khi x nhận được ước tính DV mới từ bất kỳ hàng xóm nào, nó sẽ cập nhật DV của chính nó bằng phương trình BF:

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

- Trong các điều kiện bình thường, ước tính $D_x(y)$ hội tụ đến chi phí thực tế thấp nhất $d_x(y)$



Thuật toán vectơ khoảng cách

Mỗi nút:

Chờ đợi (thay đổi chi phí liên kết cục bộ hoặc tin nhắn từ hàng xóm)

tính toán lại các ước tính DV sử dụng DV nhận được từ hàng xóm

nếu DV từ bất kỳ điểm đến nào đã thay đổi, hãy thông báo cho hàng xóm

Lặp đi lặp lại, không đồng bộ: mỗi lần lặp cục bộ gây ra bởi:

- Thay đổi chi phí liên kết cục bộ
- Tin nhắn cập nhật DV từ hàng xóm

Phân phối, tự dùng: mỗi nút chỉ thông báo cho các nút lân cận khi DV của nó thay đổi

- Hàng xóm sau đó thông báo cho hàng xóm của nó – chỉ khi cần thiết
- Không nhận được thông báo, không có hành động nào được thực hiện!



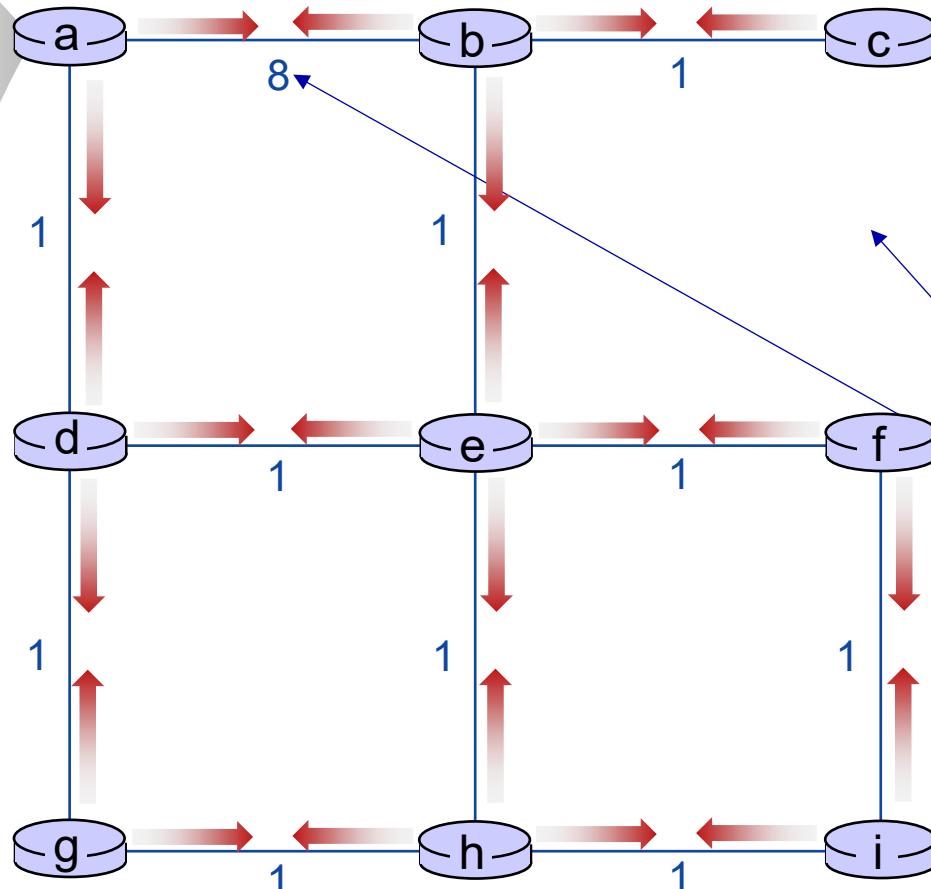
Vectơ khoảng cách: ví dụ



$t=0$

- Tất cả các nút có ước tính khoảng cách đến hàng xóm gần nhất
- Tất cả các nút gửi vectơ khoảng cách cục bộ của chúng tới các nút lân cận

DV in a:
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$



- Một vài bất đồng xứng
- Liên kết bị thiếu
 - Chi phí lớn hơn



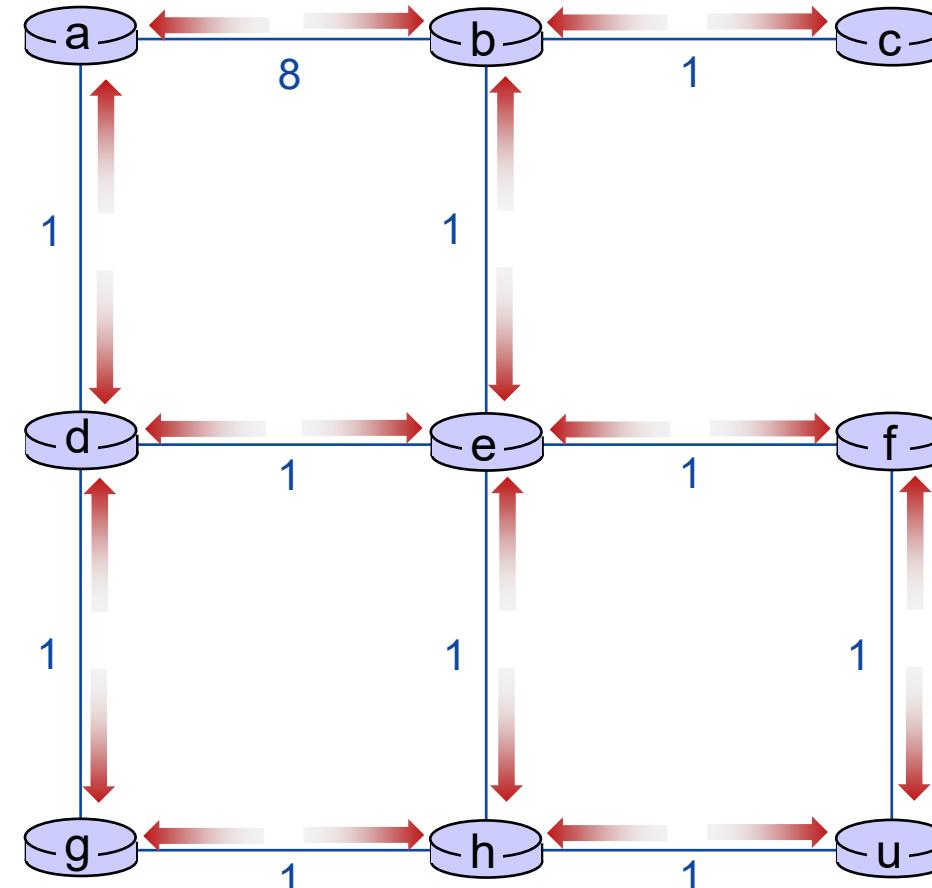
Ví dụ về vectơ khoảng cách: phép lặp



$t=1$

Tất cả các nút:

- Nhận vectơ khoảng cách từ hàng xóm
- Tính toán vectơ khoảng cách cục bộ mới của nó
- Gửi vectơ khoảng cách cục bộ mới của nó cho hàng xóm





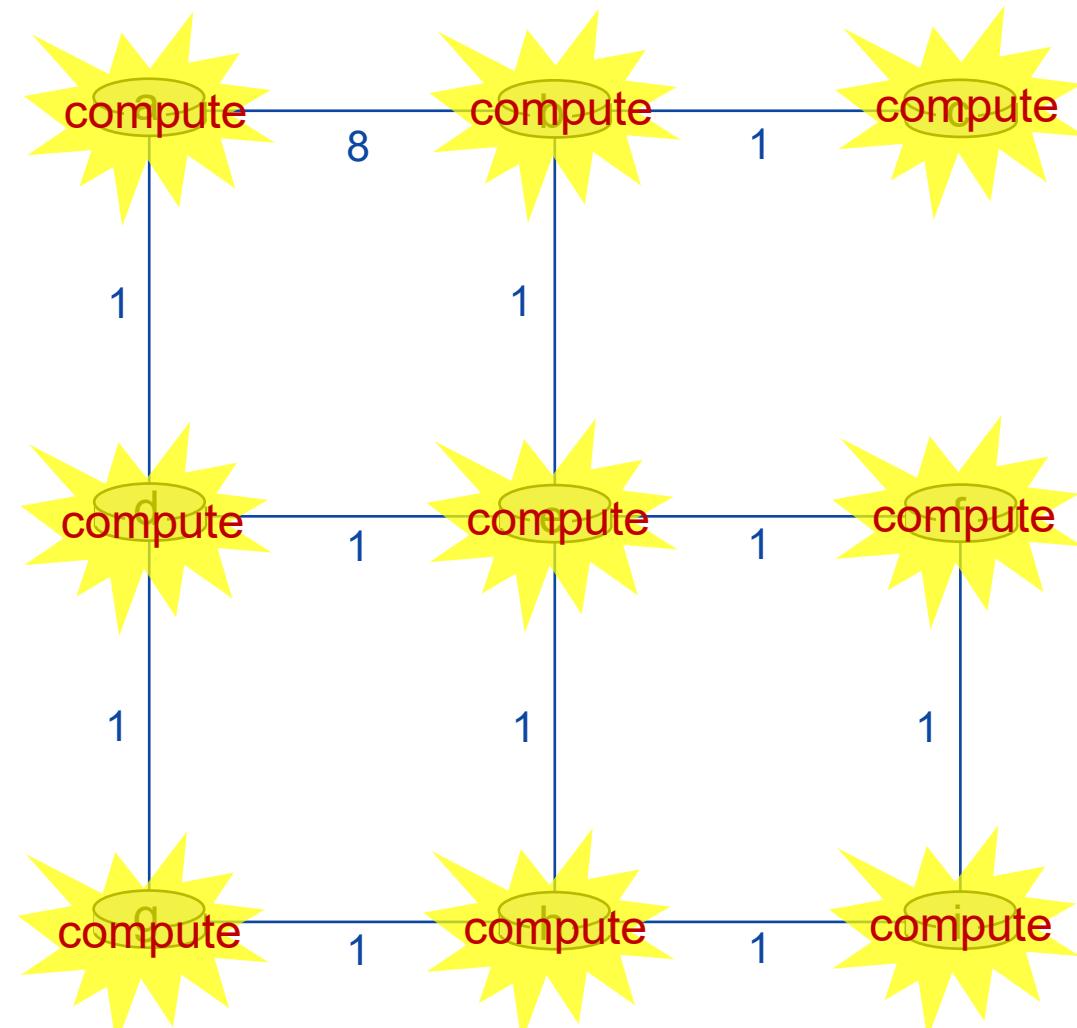
Ví dụ về vectơ khoảng cách: phép lặp



$t=1$

Tất cả các nút:

- Nhận vectơ khoảng cách từ hàng xóm
- Tính toán vectơ khoảng cách cục bộ mới của nó
- Gửi vectơ khoảng cách cục bộ mới của nó cho hàng xóm





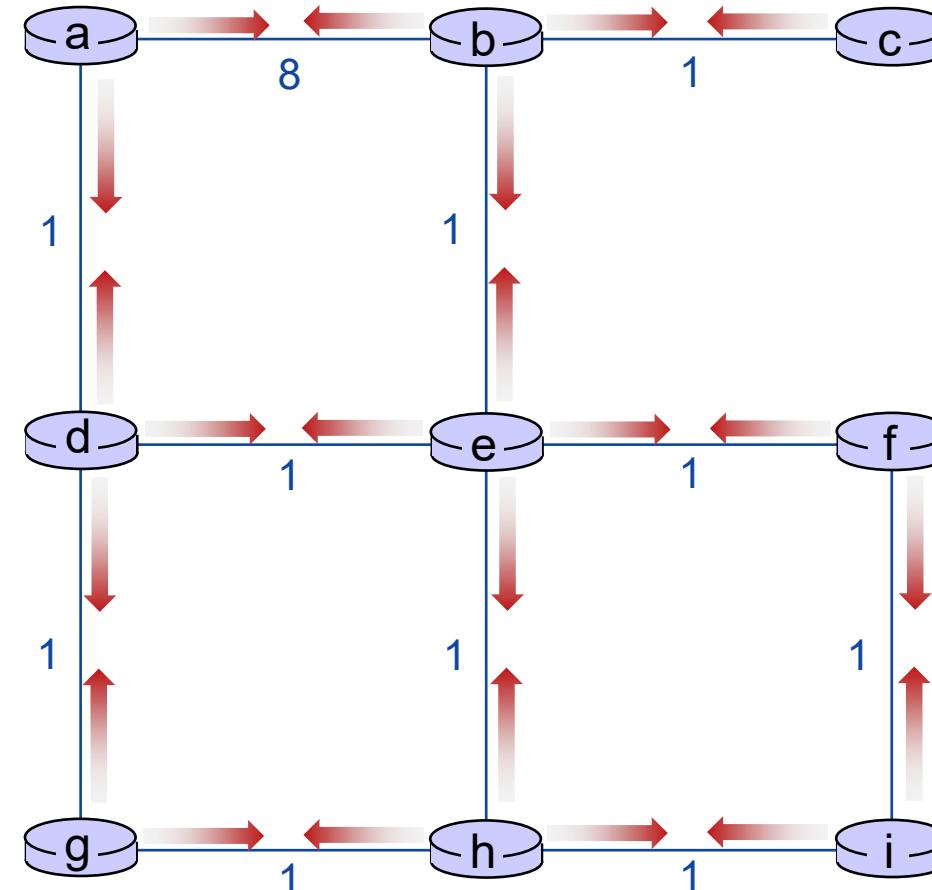
Ví dụ về vectơ khoảng cách: phép lặp



$t=1$

Tất cả các nút:

- Nhận vectơ khoảng cách từ hàng xóm
- Tính toán vectơ khoảng cách cục bộ mới của nó
- Gửi vectơ khoảng cách cục bộ mới của nó cho hàng xóm





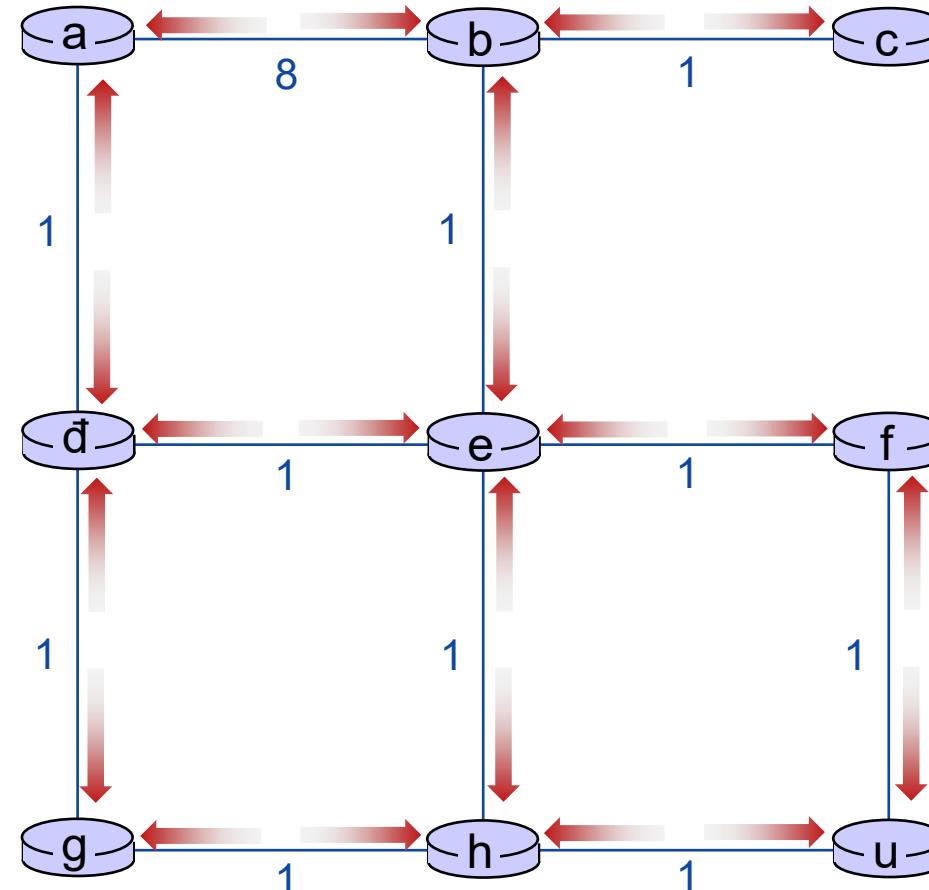
Ví dụ về vectơ khoảng cách: phép lặp



$t=2$

Tất cả các nút:

- Nhận vectơ khoảng cách từ hàng xóm
- Tính toán vectơ khoảng cách cục bộ mới của nó
- Gửi vectơ khoảng cách cục bộ mới của nó cho hàng xóm





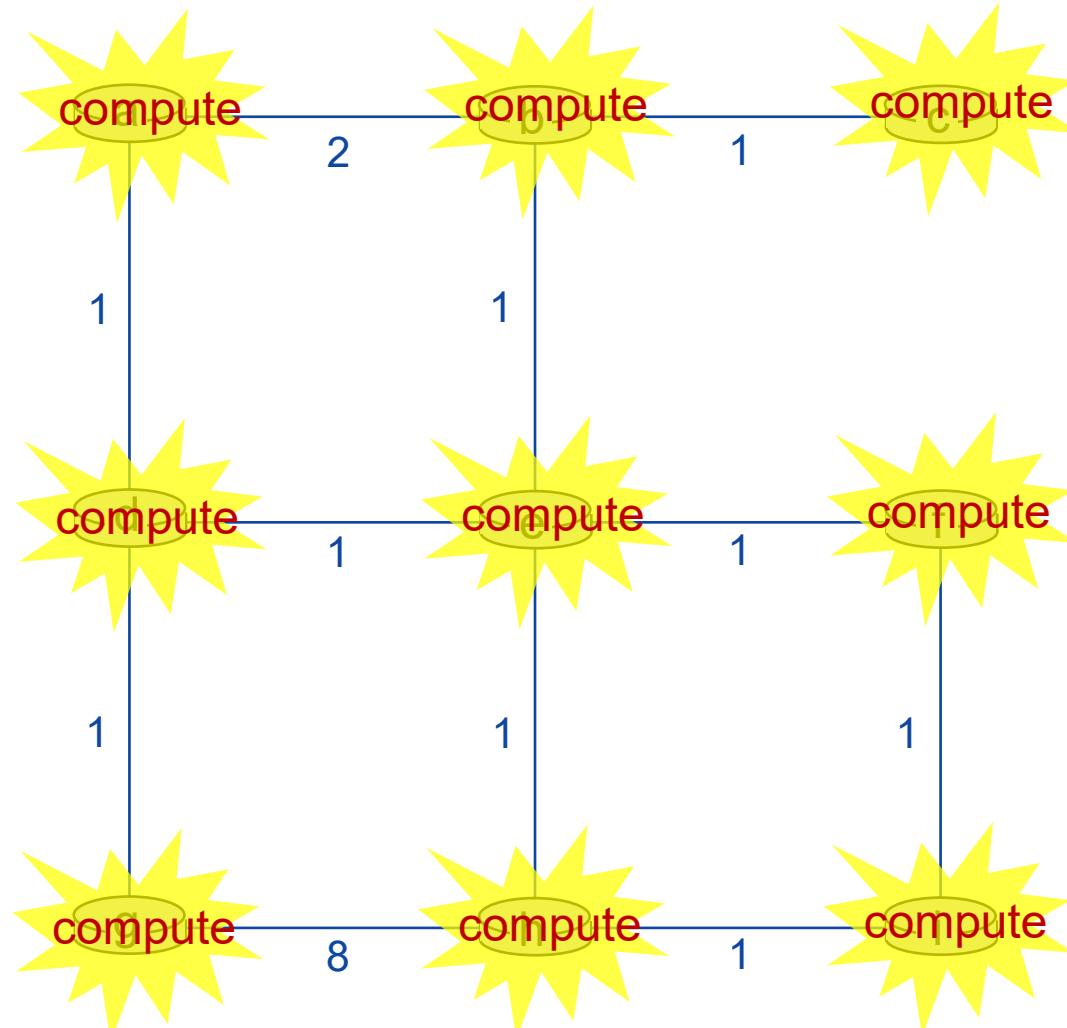
Ví dụ về vectơ khoảng cách: phép lặp



$t=2$

Tất cả các nút:

- Nhận vectơ khoảng cách từ hàng xóm
- Tính toán vectơ khoảng cách cục bộ mới của nó
- Gửi vectơ khoảng cách cục bộ mới của nó cho hàng xóm





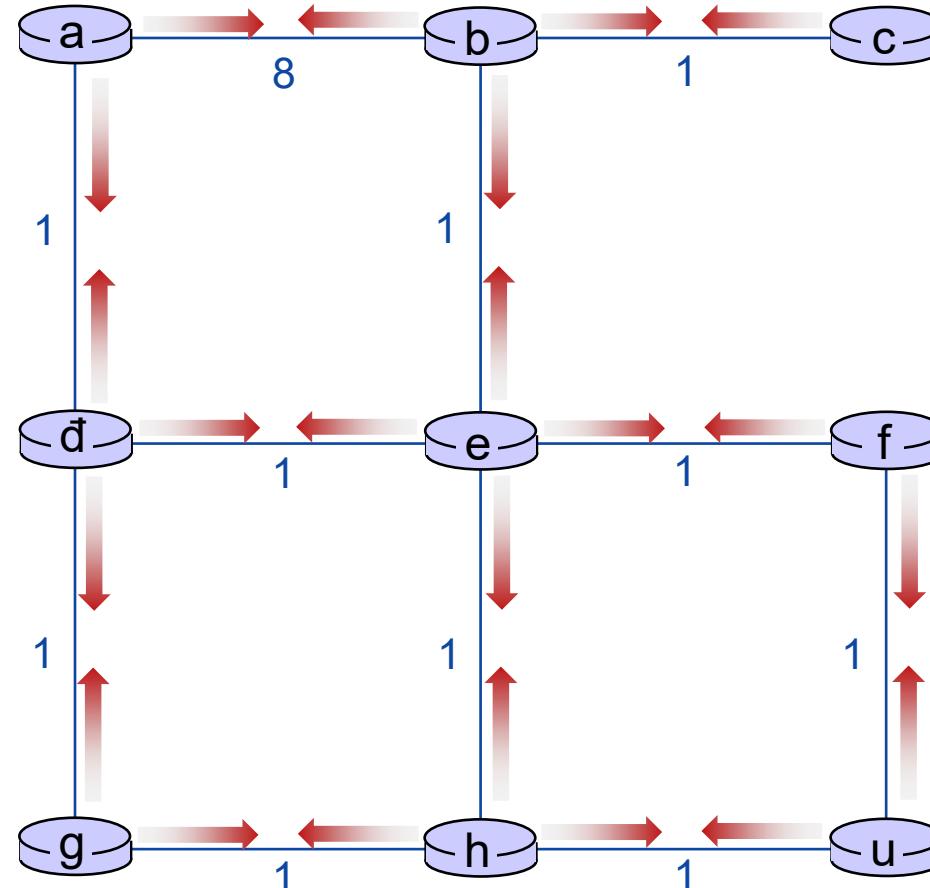
Ví dụ về vectơ khoảng cách: phép lặp



$t=2$

Tất cả các nút:

- Nhận vectơ khoảng cách từ hàng xóm
- Tính toán vectơ khoảng cách cục bộ mới của nó
- Gửi vectơ khoảng cách cục bộ mới của nó cho hàng xóm





Ví dụ về vectơ khoảng cách: phép lặp

.... và như thế

Tiếp theo chúng ta hãy xem các phép tính lặp tại các nút



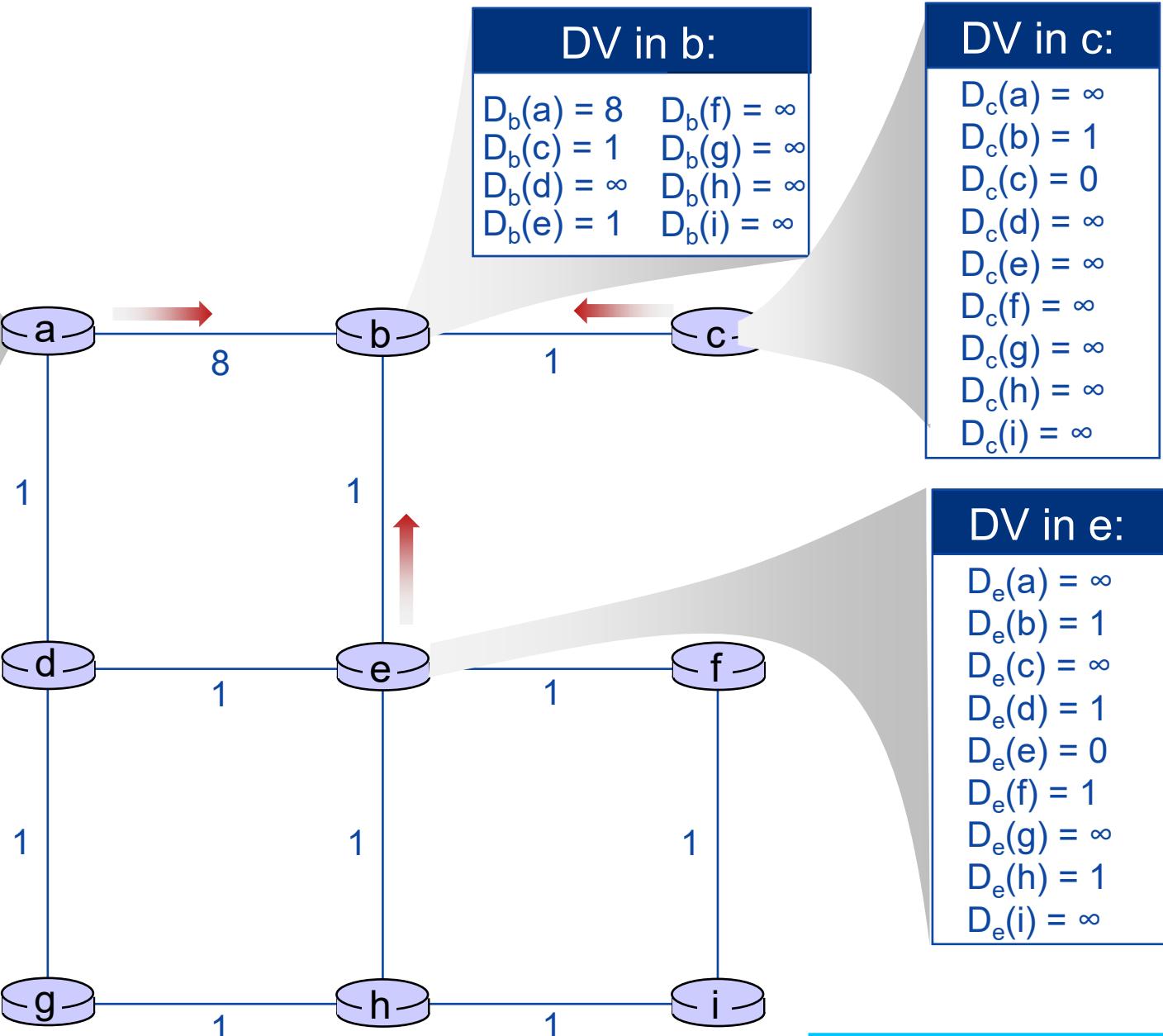
Ví dụ về vectơ khoảng cách: tính toán



t=1

- b nhận DVs từ a, c, e

DV in a:
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$





Ví dụ về vectơ khoảng cách: tính toán



t=1

- b nhận DVs từ a, c, e, tính toán:

$$D_b(a) = \min\{c_{b,a} + D_a(a), c_{b,c} + D_c(a), c_{b,e} + D_e(a)\} = \min\{8, \infty, \infty\} = 8$$

$$D_b(c) = \min\{c_{b,a} + D_a(c), c_{b,c} + D_c(c), c_{b,e} + D_e(c)\} = \min\{\infty, 1, \infty\} = 1$$

$$D_b(d) = \min\{c_{b,a} + D_a(d), c_{b,c} + D_c(d), c_{b,e} + D_e(d)\} = \min\{9, 2, \infty\} = 2$$

$$D_b(e) = \min\{c_{b,a} + D_a(e), c_{b,c} + D_c(e), c_{b,e} + D_e(e)\} = \min\{\infty, \infty, 1\} = 1$$

$$D_b(f) = \min\{c_{b,a} + D_a(f), c_{b,c} + D_c(f), c_{b,e} + D_e(f)\} = \min\{\infty, \infty, 2\} = 2$$

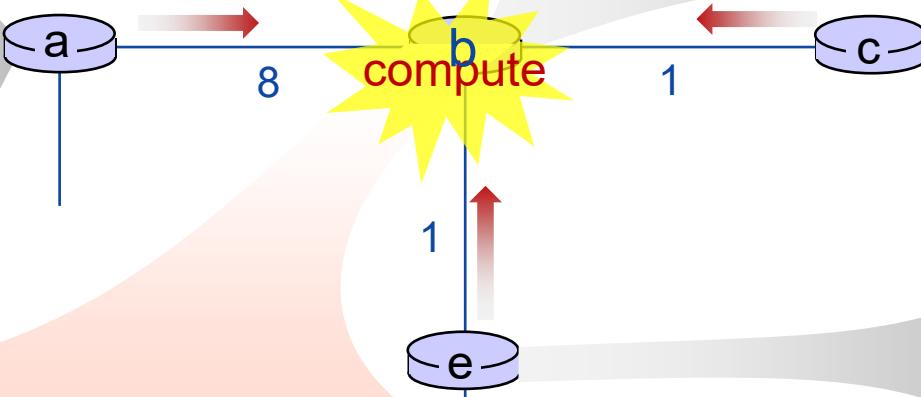
$$D_b(g) = \min\{c_{b,a} + D_a(g), c_{b,c} + D_c(g), c_{b,e} + D_e(g)\} = \min\{\infty, \infty, \infty\} = \infty$$

$$D_b(h) = \min\{c_{b,a} + D_a(h), c_{b,c} + D_c(h), c_{b,e} + D_e(h)\} = \min\{\infty, \infty, 2\} = 2$$

$$D_b(i) = \min\{c_{b,a} + D_a(i), c_{b,c} + D_c(i), c_{b,e} + D_e(i)\} = \min\{\infty, \infty, \infty\} = \infty$$

DV in a:

$$\begin{aligned} D_a(a) &= 0 \\ D_a(b) &= 8 \\ D_a(c) &= \infty \\ D_a(d) &= 1 \\ D_a(e) &= \infty \\ D_a(f) &= \infty \\ D_a(g) &= \infty \\ D_a(h) &= \infty \\ D_a(i) &= \infty \end{aligned}$$



DV in b:

$$\begin{aligned} D_b(a) &= 8 & D_b(f) &= \infty \\ D_b(c) &= 1 & D_b(g) &= \infty \\ D_b(d) &= \infty & D_b(h) &= \infty \\ D_b(e) &= 1 & D_b(i) &= \infty \end{aligned}$$

DV in c:

$$\begin{aligned} D_c(a) &= \infty \\ D_c(b) &= 1 \\ D_c(c) &= 0 \\ D_c(d) &= \infty \\ D_c(e) &= \infty \\ D_c(f) &= \infty \\ D_c(g) &= \infty \\ D_c(h) &= \infty \\ D_c(i) &= \infty \end{aligned}$$

DV in e:

$$\begin{aligned} D_e(a) &= \infty \\ D_e(b) &= 1 \\ D_e(c) &= \infty \\ D_e(d) &= 1 \\ D_e(e) &= 0 \\ D_e(f) &= 1 \\ D_e(g) &= \infty \\ D_e(h) &= 1 \\ D_e(i) &= \infty \end{aligned}$$

DV in b:

$$\begin{aligned} D_b(a) &= 8 & D_b(f) &= 2 \\ D_b(c) &= 1 & D_b(g) &= \infty \\ D_b(d) &= 2 & D_b(h) &= 2 \\ D_b(e) &= 1 & D_b(i) &= \infty \end{aligned}$$



Ví dụ về vectơ khoảng cách: tính toán

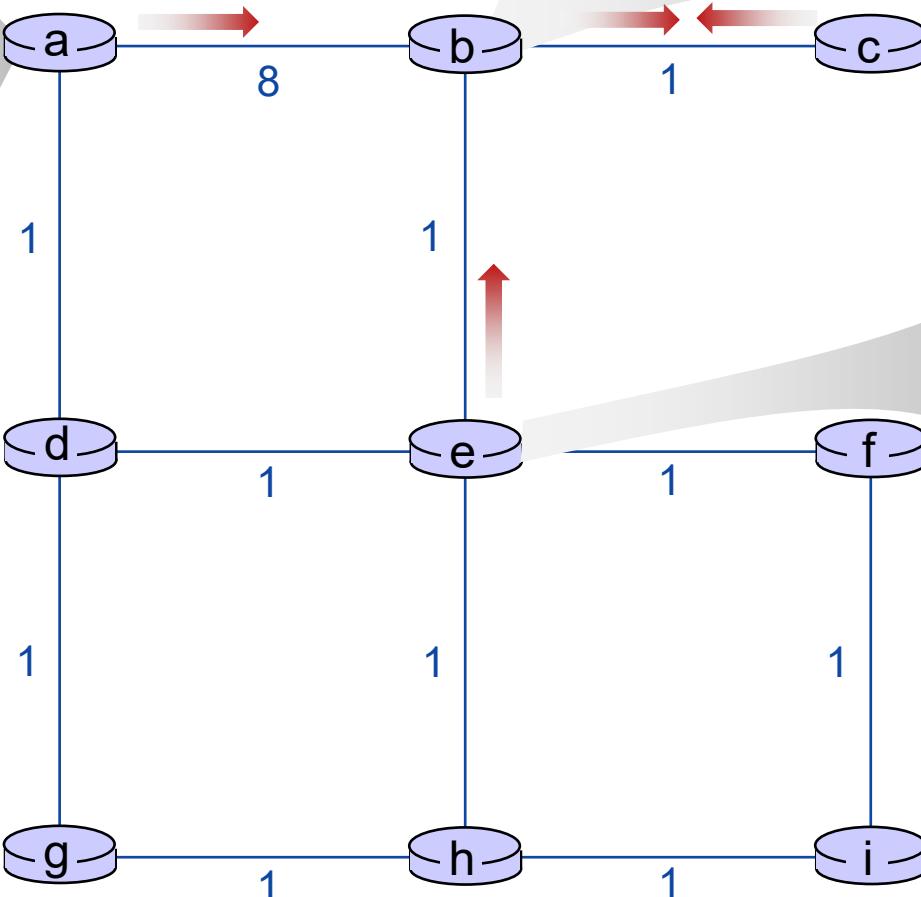


t=1

- c nhận DVs từ b

DV in a:

$$\begin{aligned} D_a(a) &= 0 \\ D_a(b) &= 8 \\ D_a(c) &= \infty \\ D_a(d) &= 1 \\ D_a(e) &= \infty \\ D_a(f) &= \infty \\ D_a(g) &= \infty \\ D_a(h) &= \infty \\ D_a(i) &= \infty \end{aligned}$$



DV in b:

$$\begin{aligned} D_b(a) &= 8 & D_b(f) &= \infty \\ D_b(c) &= 1 & D_b(g) &= \infty \\ D_b(d) &= \infty & D_b(h) &= \infty \\ D_b(e) &= 1 & D_b(i) &= \infty \end{aligned}$$

DV in c:

$$\begin{aligned} D_c(a) &= \infty \\ D_c(b) &= 1 \\ D_c(c) &= 0 \\ D_c(d) &= \infty \\ D_c(e) &= \infty \\ D_c(f) &= \infty \\ D_c(g) &= \infty \\ D_c(h) &= \infty \\ D_c(i) &= \infty \end{aligned}$$

DV in e:

$$\begin{aligned} D_e(a) &= \infty \\ D_e(b) &= 1 \\ D_e(c) &= \infty \\ D_e(d) &= 1 \\ D_e(e) &= 0 \\ D_e(f) &= 1 \\ D_e(g) &= \infty \\ D_e(h) &= 1 \\ D_e(i) &= \infty \end{aligned}$$

Ví dụ về vectơ khoảng cách: tính toán



$t=1$

- c nhận DVs từ b, tính toán:

$$D_c(a) = \min\{c_{c,b} + D_b(a)\} = 1 + 8 = 9$$

$$D_c(b) = \min\{c_{c,b} + D_b(b)\} = 1 + 0 = 1$$

$$D_c(d) = \min\{c_{c,b} + D_b(d)\} = 1 + \infty = \infty$$

$$D_c(e) = \min\{c_{c,b} + D_b(e)\} = 1 + 1 = 2$$

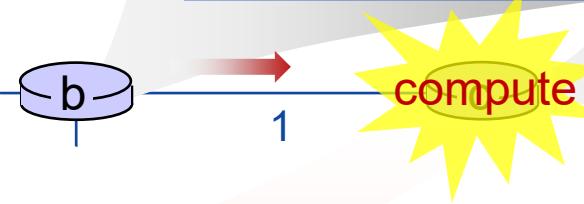
$$D_c(f) = \min\{c_{c,b} + D_b(f)\} = 1 + \infty = \infty$$

$$D_c(g) = \min\{c_{c,b} + D_b(g)\} = 1 + \infty = \infty$$

$$D_c(h) = \min\{c_{c,b} + D_b(h)\} = 1 + \infty = \infty$$

$$D_c(i) = \min\{c_{c,b} + D_b(i)\} = 1 + \infty = \infty$$

DV in b:	
$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$



DV in c:	
$D_c(a) = 9$	
$D_c(b) = 1$	
$D_c(c) = 0$	
$D_c(d) = 2$	
$D_c(e) = \infty$	
$D_c(f) = \infty$	
$D_c(g) = \infty$	
$D_c(h) = \infty$	
$D_c(i) = \infty$	

* Check out the online interactive exercises for more examples:
http://gaia.cs.umass.edu/kurose_ross/interactive/





Ví dụ về vectơ khoảng cách: tính toán



t=1

- e nhận DVs từ b, d, f, h

DV in d:

$$\begin{aligned} D_d(a) &= 1 \\ D_d(b) &= \infty \\ D_d(c) &= \infty \\ D_d(d) &= 0 \\ D_d(e) &= 1 \\ D_d(f) &= \infty \\ D_d(g) &= 1 \\ D_d(h) &= \infty \\ D_d(i) &= \infty \end{aligned}$$

DV in h:

$$\begin{aligned} D_h(a) &= \infty \\ D_h(b) &= \infty \\ D_h(c) &= \infty \\ D_h(d) &= \infty \\ D_h(e) &= 1 \\ D_h(f) &= \infty \\ D_h(g) &= 1 \\ D_h(h) &= 0 \\ D_h(i) &= 1 \end{aligned}$$

DV in b:

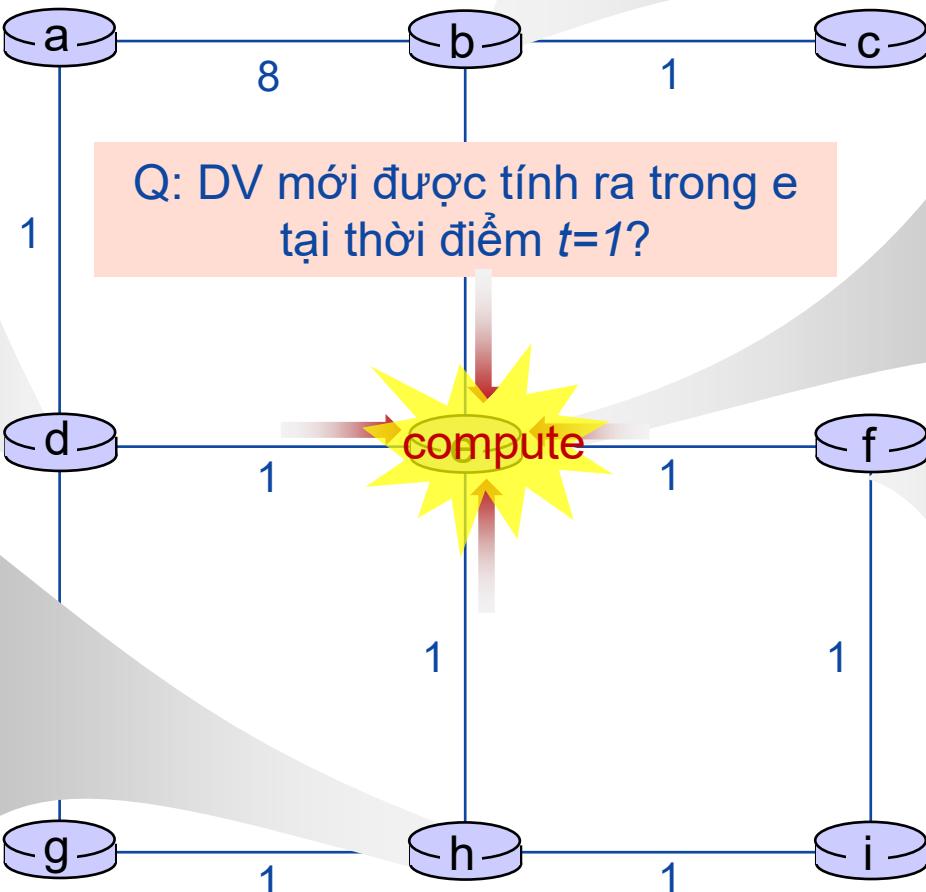
$$\begin{aligned} D_b(a) &= 8 & D_b(f) &= \infty \\ D_b(c) &= 1 & D_b(g) &= \infty \\ D_b(d) &= \infty & D_b(h) &= \infty \\ D_b(e) &= 1 & D_b(i) &= \infty \end{aligned}$$

DV in e:

$$\begin{aligned} D_e(a) &= \infty \\ D_e(b) &= 1 \\ D_e(c) &= \infty \\ D_e(d) &= 1 \\ D_e(e) &= 0 \\ D_e(f) &= 1 \\ D_e(g) &= \infty \\ D_e(h) &= 1 \\ D_e(i) &= \infty \end{aligned}$$

DV in f:

$$\begin{aligned} D_f(a) &= \infty \\ D_f(b) &= \infty \\ D_f(c) &= \infty \\ D_f(d) &= \infty \\ D_f(e) &= 1 \\ D_f(f) &= 0 \\ D_f(g) &= \infty \\ D_f(h) &= \infty \\ D_f(i) &= 1 \end{aligned}$$





Vectơ khoảng cách: khuếch tán thông tin trạng thái

Giao tiếp lặp, các bước tính toán khuếch tán thông tin qua mạng:

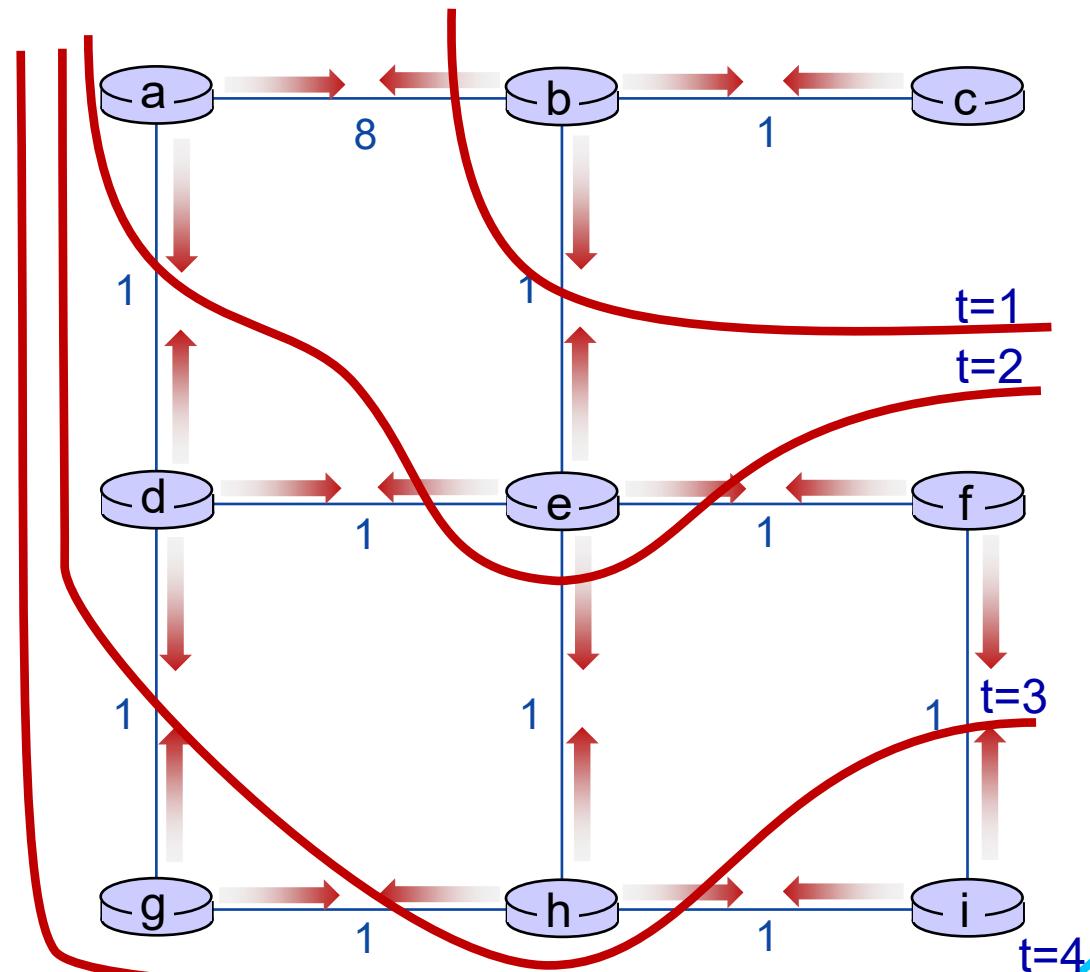
t=0 trạng thái của c tại t=0 chỉ tại c

t=1 trạng thái của c tại t=0 đã lan truyền tới b và có thể ảnh hưởng đến việc tính toán vectơ khoảng cách cách xa tới 1 bước nhảy, tức là tại b

t=2 trạng thái của c tại t=0 bây giờ có thể ảnh hưởng đến việc tính toán vectơ khoảng cách cách xa tới 2 bước nhảy, nghĩa là tại b và bây giờ tại a, e cũng vậy

t=3 trạng thái của c tại t=0 có thể ảnh hưởng đến việc tính toán vectơ khoảng cách cách xa tới 3 bước nhảy, tức là tại b,a,e và bây giờ tại d,f,h

t=4 trạng thái của c tại t=0 có thể ảnh hưởng đến việc tính toán vectơ khoảng cách cách xa tới 4 bước nhảy, nghĩa là tại b, a, e, d, f, h và bây giờ là cả g,i

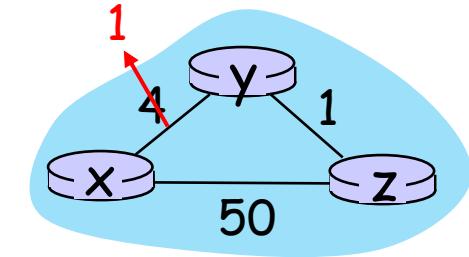




Vectơ khoảng cách: thay đổi chi phí liên kết

Thay đổi chi phí liên kết:

- Nút phát hiện thay đổi chi phí liên kết cục bộ
- Cập nhật thông tin định tuyến, tính toán lại DV cục bộ
- Nếu DV thay đổi, thông báo cho hàng xóm



t_0 : y phát hiện thay đổi chi phí liên kết, cập nhật DV của nó, thông báo cho các hàng xóm của nó.

t_1 : z nhận cập nhật từ y, cập nhật bảng của nó, tính toán chi phí nhỏ nhất mới cho x, gửi DV của nó cho các hàng xóm.

t_2 : y nhận cập nhật của z, cập nhật bảng DV của nó. Chi phí nhỏ nhất của y không thay đổi, vì vậy y không gửi tin nhắn đến z.

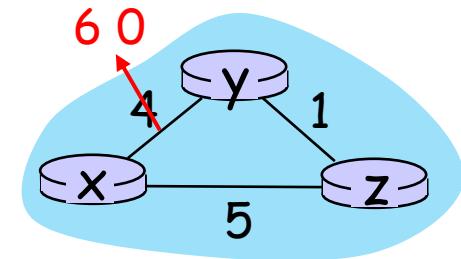
“tin tốt
đi nhanh”



Vectơ khoảng cách: thay đổi chi phí liên kết

Thay đổi chi phí liên kết:

- Nút phát hiện thay đổi chi phí liên kết cục bộ
- “**Tin xấu lan truyền chậm**” – bài toán đếm đến vô cùng:
 - y thấy liên kết trực tiếp tới x có chi phí mới là 60, nhưng z đã nói rằng nó có một đường dẫn với chi phí là 5. Vì vậy, y tính toán “chi phí mới của tôi tới x sẽ là 6, thông qua z); thông báo cho z về chi phí mới từ 6 đến x .
 - z biết rằng đường dẫn đến x qua y có chi phí mới là 6, vì vậy z tính toán “chi phí mới của tôi tới x sẽ là 7 qua y), thông báo cho y về chi phí mới từ 7 đến x .
 - y biết rằng đường dẫn đến x qua z có chi phí mới là 7, vì vậy y tính toán “chi phí mới của tôi tới x sẽ là 8 qua y), thông báo cho z về chi phí mới từ 8 đến x .
 - z biết rằng đường dẫn đến x qua y có chi phí mới là 8, vì vậy z tính toán “chi phí mới của tôi tới x sẽ là 9 qua y), thông báo cho y về chi phí mới từ 9 đến x .
- Xem tài liệu cho các giải pháp. Các *thuật toán phân tán* rất phức tạp!



So sánh thuật toán LS và DV



Độ phức tạp của thông điệp

LS: n router, gửi tin nhắn $O(n^2)$

DV: trao đổi hàng xóm với nhau; thời gian hội tụ thay đổi

Tốc độ hội tụ

LS: Thuật toán $O(n^2)$, tin nhắn $O(n^2)$

- Có thể có dao động

DV: thời gian hội tụ thay đổi

- Có thể có các vòng lặp định tuyến
- Bài toán đếm đến vô cùng

Điều gì xảy ra nếu router gặp trực tiếp hoặc bị xâm phạm?

LS:

- Router có thể quảng cáo chi phí *liên kết không chính xác*
- Mỗi router chỉ tính toán bảng *riêng* của nó

DV:

- Router DV có thể quảng cáo chi phí *đường dẫn không chính xác* ("Tôi có một đường dẫn chi phí *thực sự thấp* đến mọi nơi"): black-holing
- Mỗi bảng của router được sử dụng bởi những router khác: lối lan truyền qua mạng



Chương 4: Nội dung

4.1 Tổng quan

- Mức dữ liệu
- Mức điều khiển

4.2 Virtual circuit network và datagram network

- Các mạch ảo
- Mạng datagram

4.3 Cấu trúc bên trong router

- Cổng đầu vào, chuyển đổi, cổng đầu ra
- Quản lý bộ đệm, lập kế hoạch

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing

- DHCP
- NAT
- ICMP
- IPV6

4.5 Các thuật toán routing

- Link state
- Distance vector

4.6 Routing trong Internet

- Hierarchical routing
- RIP, OSPF, BGP
- SDN
- Quản lý mạng

4.7 Broadcast và Multicast routing



Giúp cho việc định tuyến có thể mở rộng

Nghiên cứu định tuyến cho đến nay - lý tưởng hóa

- Tất cả các router giống hệt nhau
- Mạng “phẳng”
- ... không đúng trong thực tế

Quy mô: hàng tỷ điểm đến:

- Không thể lưu trữ tất cả các điểm đến trong bảng định tuyến!
- Trao đổi bảng định tuyến sẽ tràn ngập các liên kết!

Quản trị vùng tự quản:

- Internet: mạng của các mạng
- Mỗi quản trị viên mạng có thể muốn kiểm soát định tuyến trong mạng riêng của mình

Cách tiếp cận Internet để định tuyến có thể mở rộng



Tổng hợp các router thành các vùng được gọi là “hệ thống tự quản” (AS) (còn gọi là “miền”)

intra-AS (hay còn gọi là “nội miền”):
định tuyến giữa các router cùng AS (“mạng”)

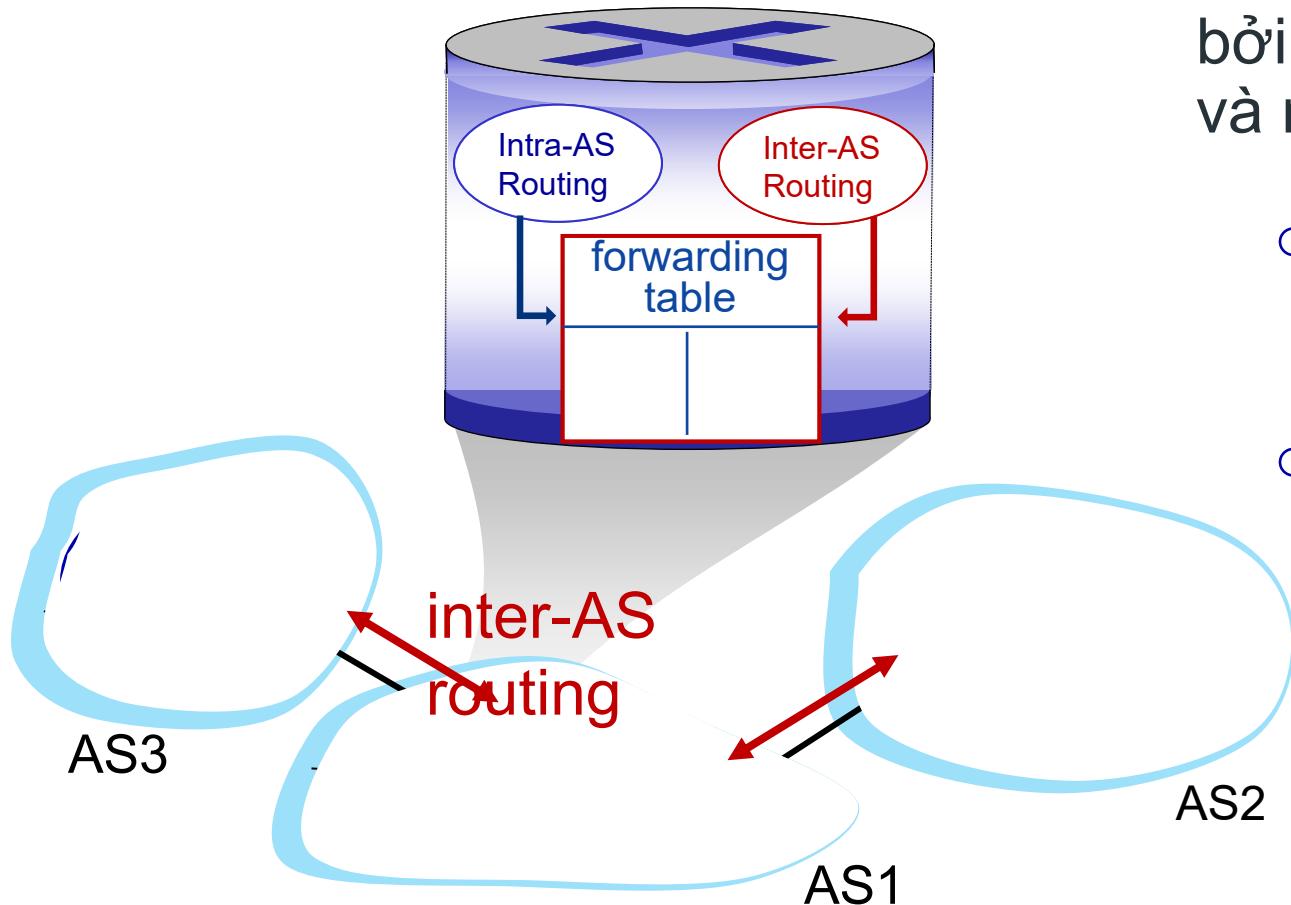
- Tất cả các router trong AS phải chạy cùng một giao thức nội miền
- Các router trong AS khác nhau có thể chạy các giao thức định tuyến nội miền khác nhau
- Router cổng: ở “cạnh” của AS của chính nó, có (các) liên kết đến (các) router trong các AS khác

inter-AS (còn gọi là “liên miền”):
Định tuyến giữa các AS

- Cổng thực hiện định tuyến liên miền (cũng như định tuyến nội miền)



Các AS được kết nối với nhau



Bảng chuyển tiếp được định cấu hình bởi các thuật toán định tuyến trong và ngoài AS

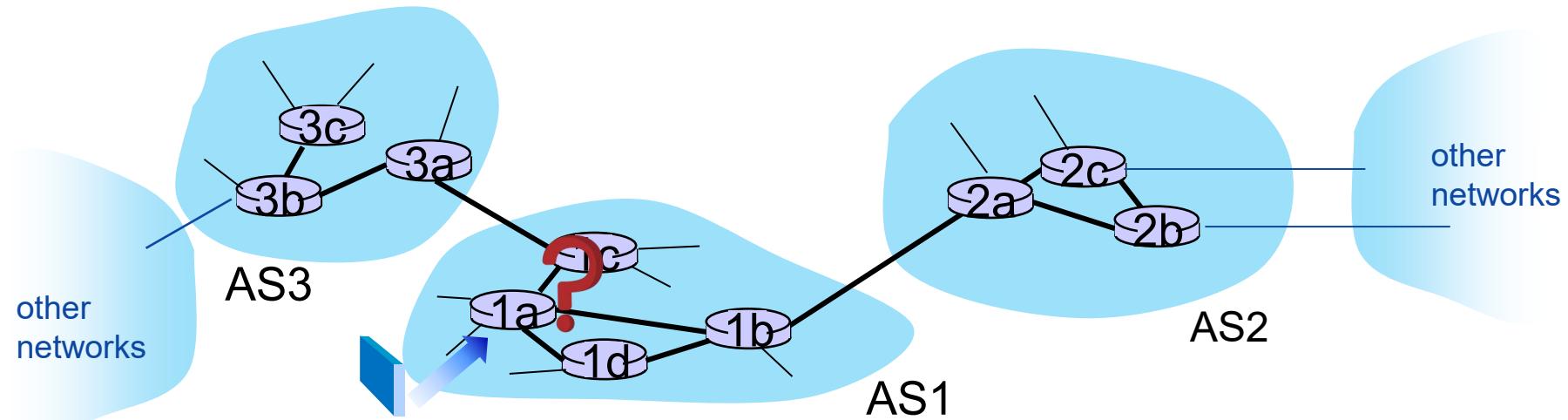
- Định tuyến trong AS xác định mục nhập cho các đích trong AS
- Inter-AS & intra-AS xác định mục nhập cho các điểm đến bên ngoài

Định tuyến giữa các AS: vai trò trong chuyển tiếp nội miền



- Giả sử router trong AS 1 nhận được gói dữ liệu được định sẵn bên ngoài AS 1 :

Router nên chuyển tiếp gói đến router cổng trong AS1, nhưng cái nào?



Định tuyến liên miền AS 1 phải:

1. Tìm hiểu điểm đến nào có thể truy cập thông qua AS2, điểm đến nào thông qua AS3
2. Truyền bá thông tin khả năng tiếp cận này đến tất cả các router trong AS 1



Định tuyến trong một AS

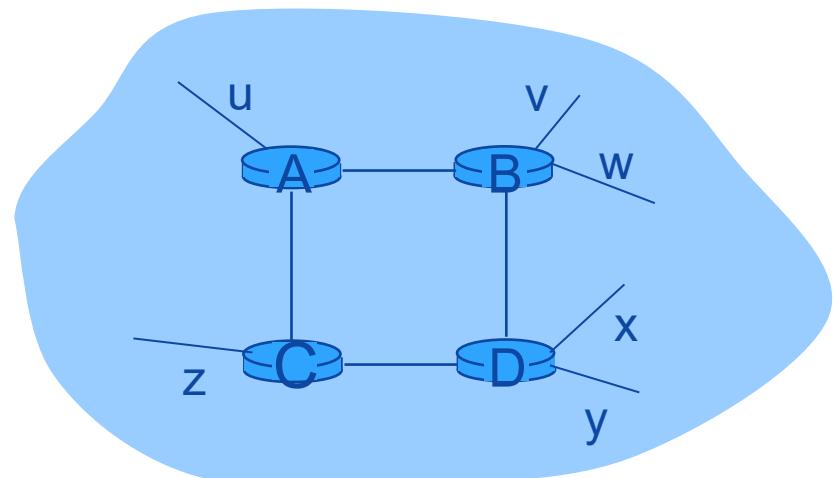
Các giao thức định tuyến nội bộ AS phổ biến nhất:

- RIP: Routing Information Protocol [RFC 1723]
 - DV cổ điển: DV được trao đổi cứ sau 30 giây
 - Không còn được sử dụng rộng rãi
- EIGRP: Enhanced Interior Gateway Routing Protocol
 - Dựa trên DV
 - Trước đây là độc quyền của Cisco trong nhiều thập kỷ (được mở vào năm 2013 [RFC 7868])
- OSPF: Open Shortest Path First [RFC 2328]
 - Định tuyến trạng thái liên kết
 - Giao thức IS-IS (tiêu chuẩn ISO, không phải tiêu chuẩn RFC) về cơ bản giống như OSPF



Định tuyến RIP (Routing Information Protocol)

- Công bố vào năm 1982 trong BSD-UNIX
- Thuật toán distance vector
 - Metric 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: danh sách lên đến 25 **subnet** đích

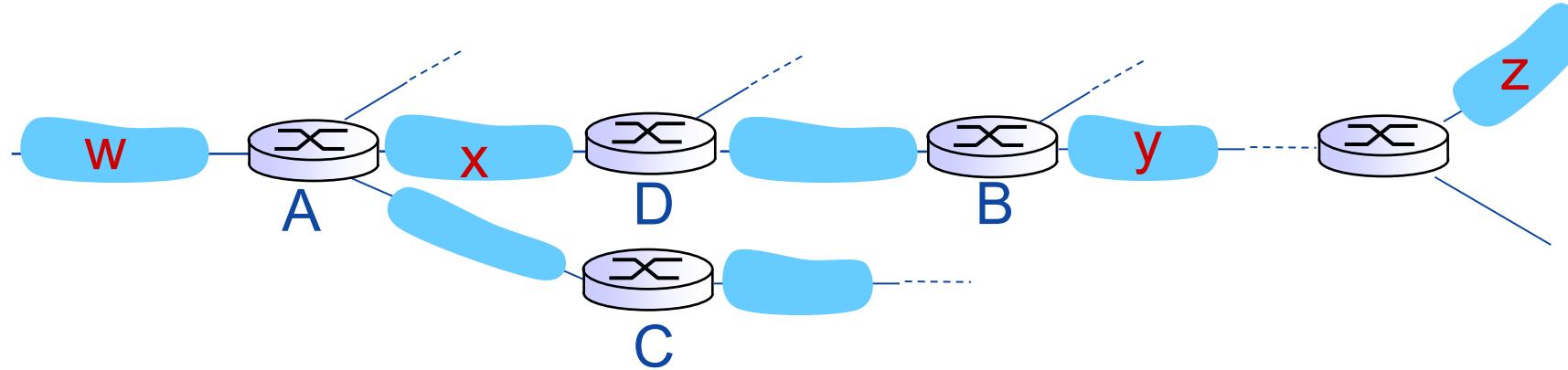


Từ router A đến các **subnet** đích:

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



Định tuyến RIP: ví dụ

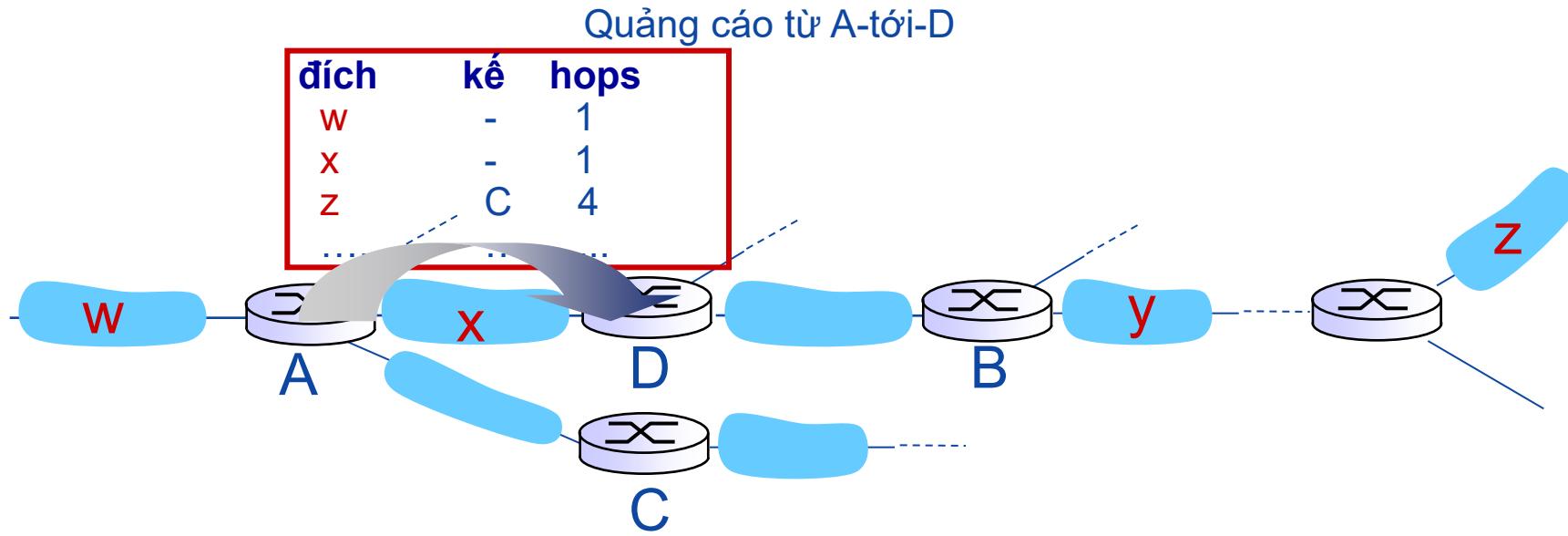


Bảng định tuyến trong router D

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



Định tuyến RIP: ví dụ



Bảng định tuyến trong router D

Subnet đích	router kế tiếp	số lượng hop tới đích
W	A	2
y	B	2
z	B	7
x	A	5
...	--	1
...

Định tuyến 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 bị vô hiệu
- Các quảng cáo mới được gửi tới các neighbor
- 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)

Định tuyến OSPF



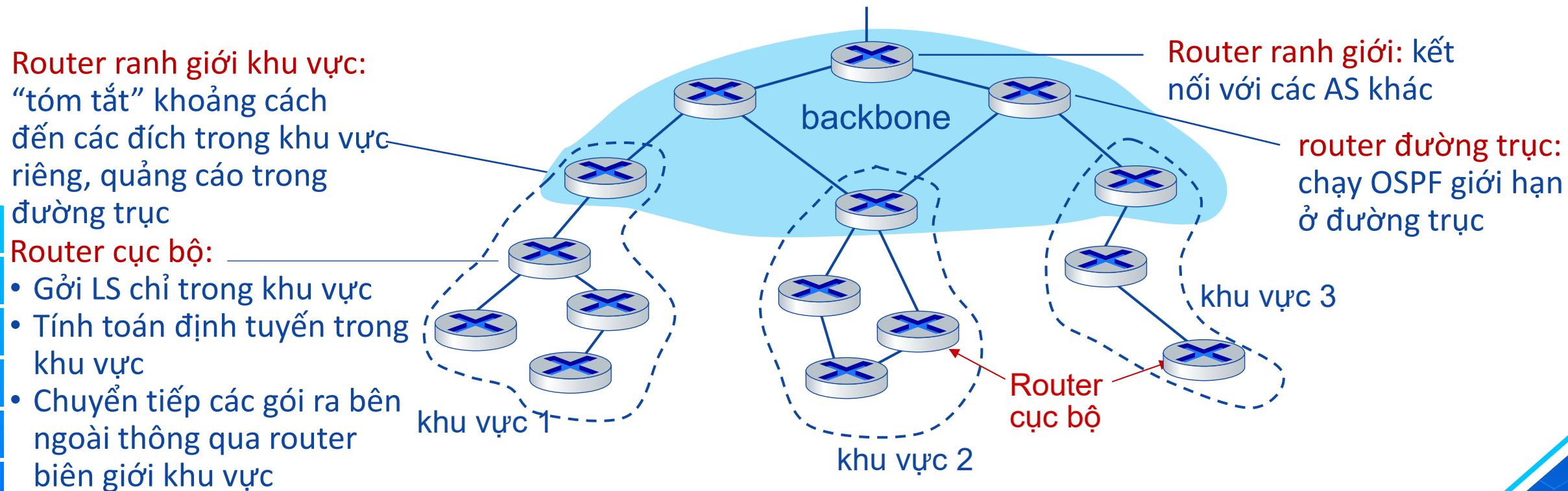
- “Mở”: công khai
- Trạng thái liên kết cổ điển
 - Mỗi router gửi quảng bá gói tin quảng cáo trạng thái liên kết OSPF (trực tiếp qua IP thay vì sử dụng TCP/UDP) tới tất cả các router khác trong toàn bộ AS
 - Nhiều chỉ số chi phí liên kết có thể: băng thông, độ trễ
 - Mỗi router có cấu trúc liên kết đầy đủ, sử dụng thuật toán Dijkstra để tính toán bảng chuyển tiếp
- *Bảo mật*: tất cả các thông báo OSPF được xác thực (để ngăn chặn sự xâm nhập độc hại)



OSPF phân cấp

- Phân cấp hai cấp: local area, backbone.

- Quảng cáo trạng thái liên kết chỉ được flood trong khu vực hoặc đường trục
- Mỗi nút có cấu trúc liên kết khu vực chi tiết; chỉ biết phương hướng để đi đến được các điểm đến khác



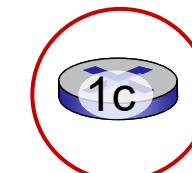
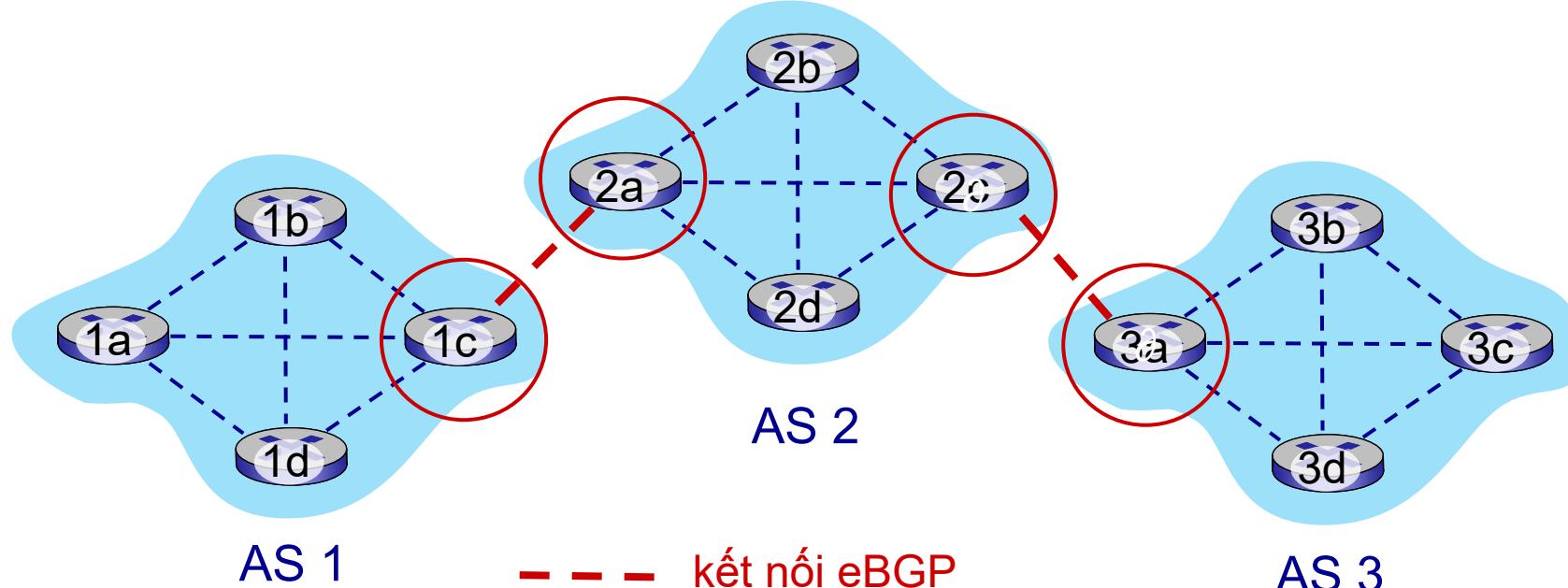


Định tuyến giữa các AS: BGP

- BGP (Border Gateway Protocol): giao thức định tuyến liên miền thực tế
 - “chất keo kết nối Internet với nhau”
- Cho phép mạng con quảng cáo sự tồn tại của nó và các điểm đến mà nó có thể tiếp cận với phần còn lại của Internet: “ *Tôi ở đây, đây là người tôi có thể tiếp cận và bằng cách*”
- BGP cung cấp cho mỗi AS một phương tiện để:
 - eBGP: lấy thông tin về khả năng tiếp cận mạng con từ các AS lân cận
 - iBGP: truyền thông tin về khả năng tiếp cận tới tất cả các router bên trong AS.
 - Xác định các tuyến “ tốt” đến các mạng khác dựa trên thông tin và *chính sách về khả năng tiếp cận*



Kết nối eBGP, iBGP



Gateway router chạy cả giao thức eBGP và iBGP



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

Chính sách:

- Inter-AS: quản trị viên muốn kiểm soát cách lưu lượng truy cập được định tuyến, ai định tuyến qua mạng của nó
- Intra-AS: quản trị viên duy nhất, vì vậy chính sách ít có vấn đề hơn

Tỉ lệ:

- Định tuyến phân cấp giúp tiết kiệm kích thước bảng, giảm lưu lượng cập nhật

Hiệu suất:

- Intra-AS: có thể tập trung vào hiệu suất
- Inter-AS: chính sách chi phối hiệu suất

Mạng được xác định bằng phần mềm (SDN)

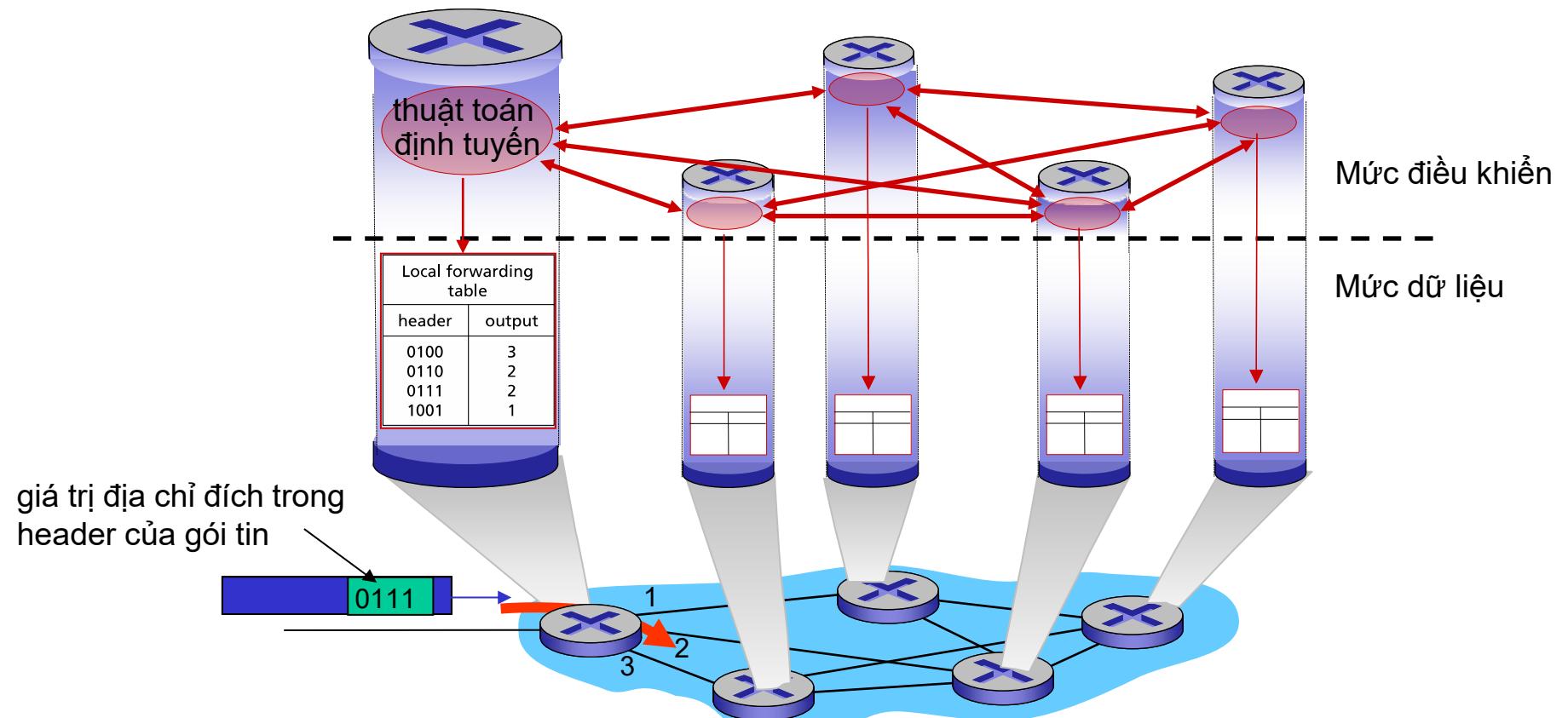


- Lớp mạng Internet: được triển khai theo lịch sử thông qua phương pháp kiểm soát phân tán trên mỗi router:
 - *Nguyên khối* chứa phần cứng chuyển mạch, chạy triển khai độc quyền các giao thức chuẩn Internet (IP, RIP, IS-IS, OSPF, BGP) trong hệ điều hành router độc quyền (ví dụ: Cisco IOS)
 - Các “hộp trung gian” khác nhau cho các chức năng lớp mạng khác nhau: tường lửa, bộ cân bằng tải, hộp NAT, ..
- ~2005: quan tâm mới đến việc xem xét lại mức điều khiển mạng



Mức điều khiển ở mỗi router

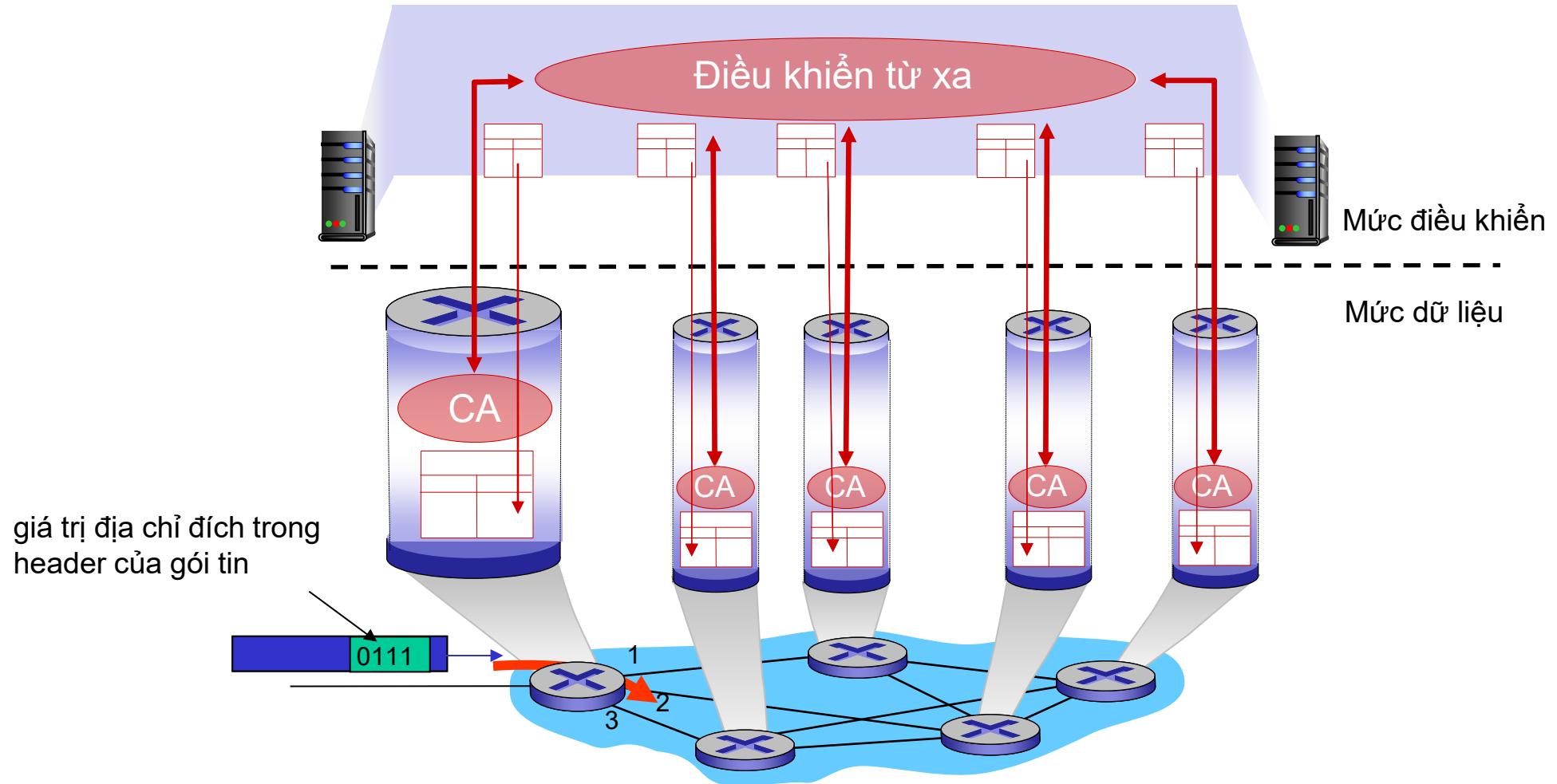
Các thành phần thuật toán định tuyến riêng lẻ *trong mỗi và mọi router* tương tác trong mức điều khiển





Mức điều khiển được xác định bằng phần mềm (SDN – Software-Defined-Networking)

Bộ điều khiển từ xa tính toán, cài đặt bảng chuyển tiếp trong router



Mạng được xác định bằng phần mềm (SDN)



Tại sao mức điều khiển tập trung là hợp lý?

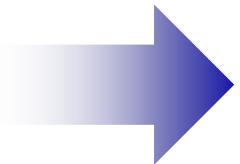
- Quản lý mạng dễ dàng hơn: tránh cấu hình sai router, luồng lưu lượng linh hoạt hơn
- Chuyển tiếp dựa trên bảng (gọi lại OpenFlow API) cho phép các router “lập trình”
 - “lập trình” tập trung dễ dàng hơn: tính toán bảng tập trung và phân phối
 - “lập trình” phân tán khó hơn: tính toán các bảng do thuật toán (giao thức) phân tán được triển khai trong từng router
- Triển khai mở (không độc quyền) của mức điều khiển
 - Thúc đẩy đổi mới: hãy để 1000 bông hoa nở



Tương tự SDN: cuộc cách mạng từ máy tính lớn đến PC

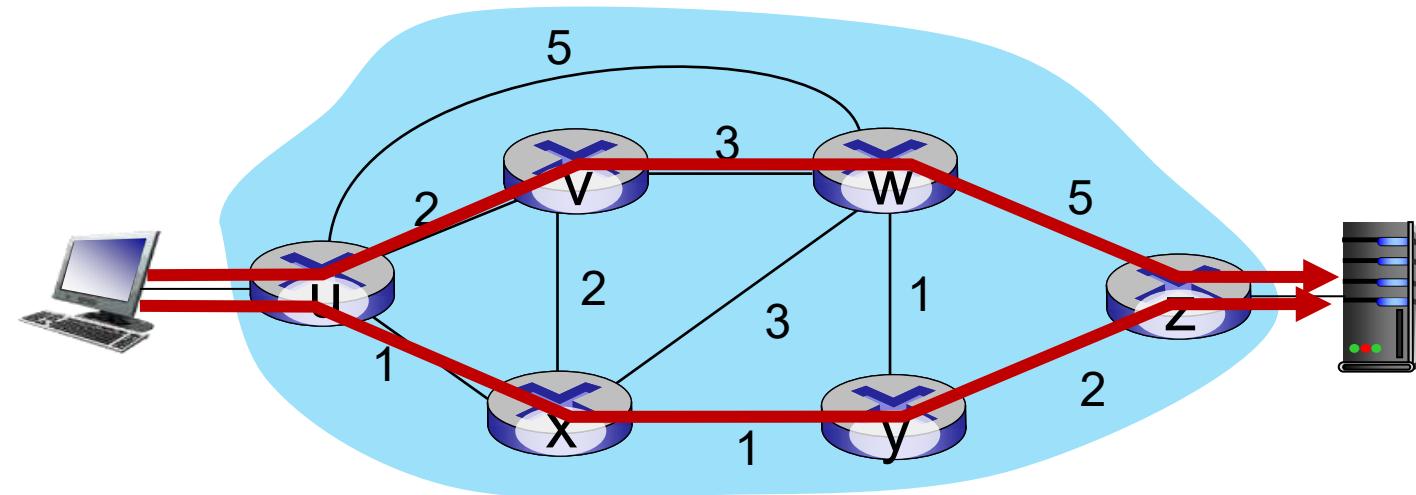


Tích hợp theo chiều dọc
Đóng cửa, độc quyền
Chậm đổi mới
Công nghiệp nhỏ



Tích hợp theo chiều ngang
Giao tiếp mở
Đổi mới nhanh chóng
Ngành công nghiệp khổng lồ

Kỹ thuật lưu lượng: khó khăn với định tuyến truyền thông



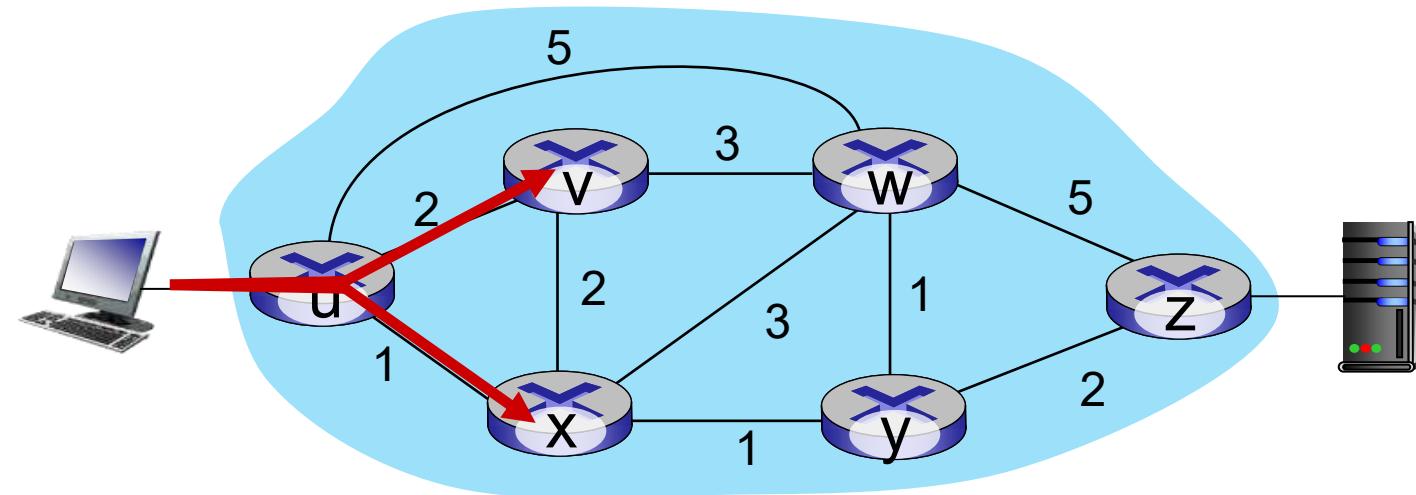
Hỏi: nếu nhà điều hành mạng muốn lưu lượng truy cập từ u đến z chạy dọc theo $uvwz$, thay vì $uxyz$ thì sao?

Trả lời: cần xác định lại trọng số liên kết để thuật toán định tuyến lưu lượng tính toán các tuyến tương ứng (hoặc cần một thuật toán định tuyến mới)!

Trọng số liên kết chỉ là “nút” điều khiển: không kiểm soát nhiều!



Kỹ thuật lưu lượng: khó khăn với định tuyến truyền thông

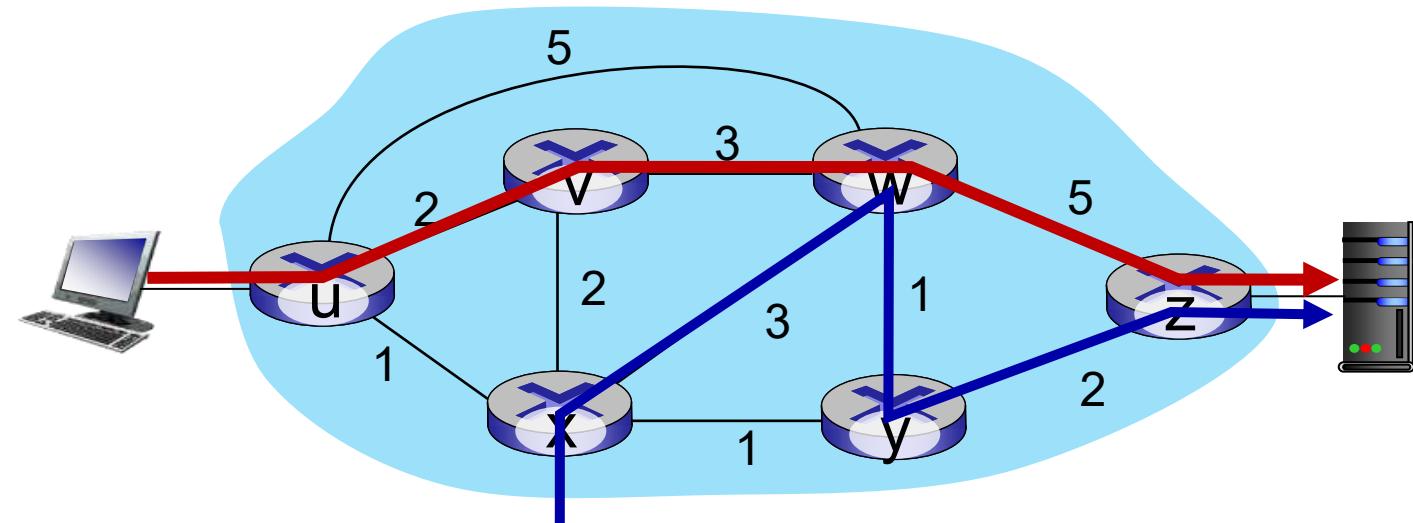


Hỏi: nếu nhà điều hành mạng muốn phân chia lưu lượng u-to-z dọc theo uvwz và uxyz (cân bằng tải) thì sao?

Trả lời: không làm được (hoặc cần thuật toán định tuyến mới)



Kỹ thuật lưu lượng: khó khăn với định tuyến truyền thống



Hỏi: Nếu w muốn định tuyến lưu lượng màu xanh và đỏ khác nhau từ w đến z thì sao?

Trả lời: Không thể thực hiện được (với chuyển tiếp dựa trên đích và định tuyến LS, DV)

Chuyển tiếp tổng quát và SDN có thể được sử dụng để đạt được bất kỳ định tuyến mong muốn nào



Mạng được xác định bằng phần mềm (SDN)

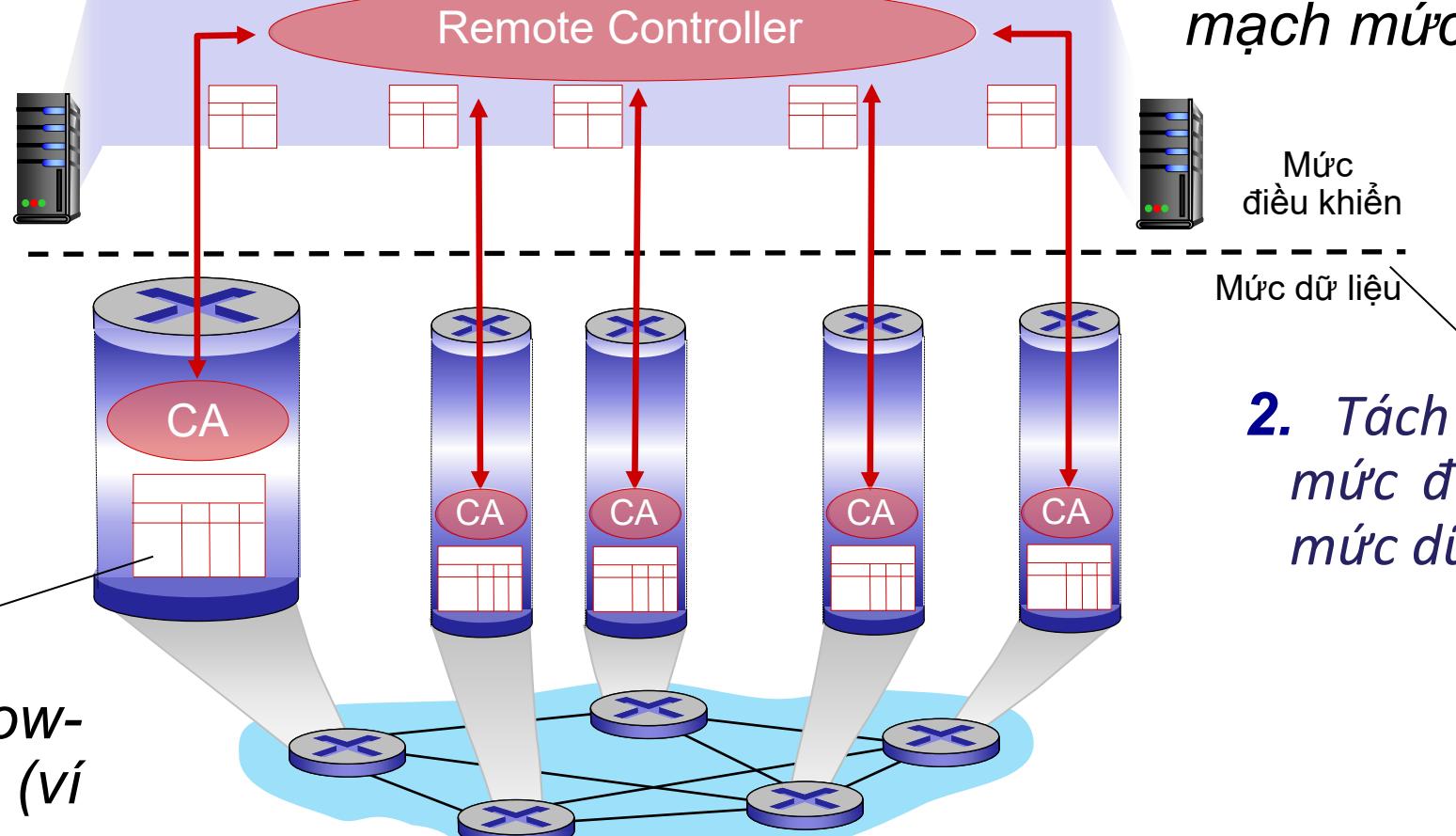
4. Các ứng dụng điều khiển có thể lập trình được

routing
access control

...

load balance

3. Các chức năng mức điều khiển bên ngoài đến các chuyển mạch mức dữ liệu



1. Chuyển tiếp “flow-based” tổng quát (ví dụ OpenFlow)

2. Tách riêng giữa mức điều khiển và mức dữ liệu

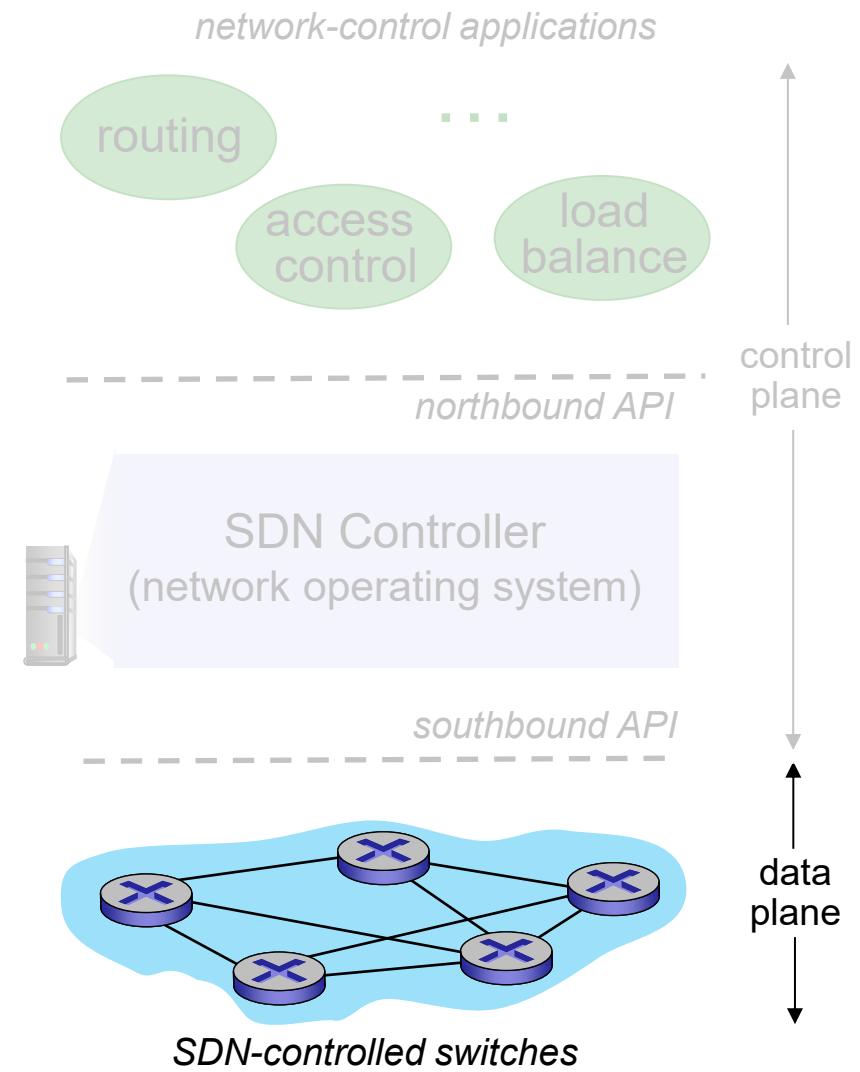
Mức điều khiển
Mức dữ liệu



Mạng được xác định bằng phần mềm (SDN)

Chuyển mạch ở mức dữ liệu:

- Chuyển mạch nhanh, đơn giản triển khai chuyển tiếp mức dữ liệu tổng quát trong phần cứng
- Bảng luồng (chuyển tiếp) được tính toán, cài đặt dưới sự giám sát của bộ điều khiển
- API cho điều khiển chuyển đổi dựa trên bảng (ví dụ: OpenFlow)
 - Xác định những gì có thể kiểm soát được, cái gì không
- Giao thức để giao tiếp với bộ điều khiển (ví dụ: OpenFlow)

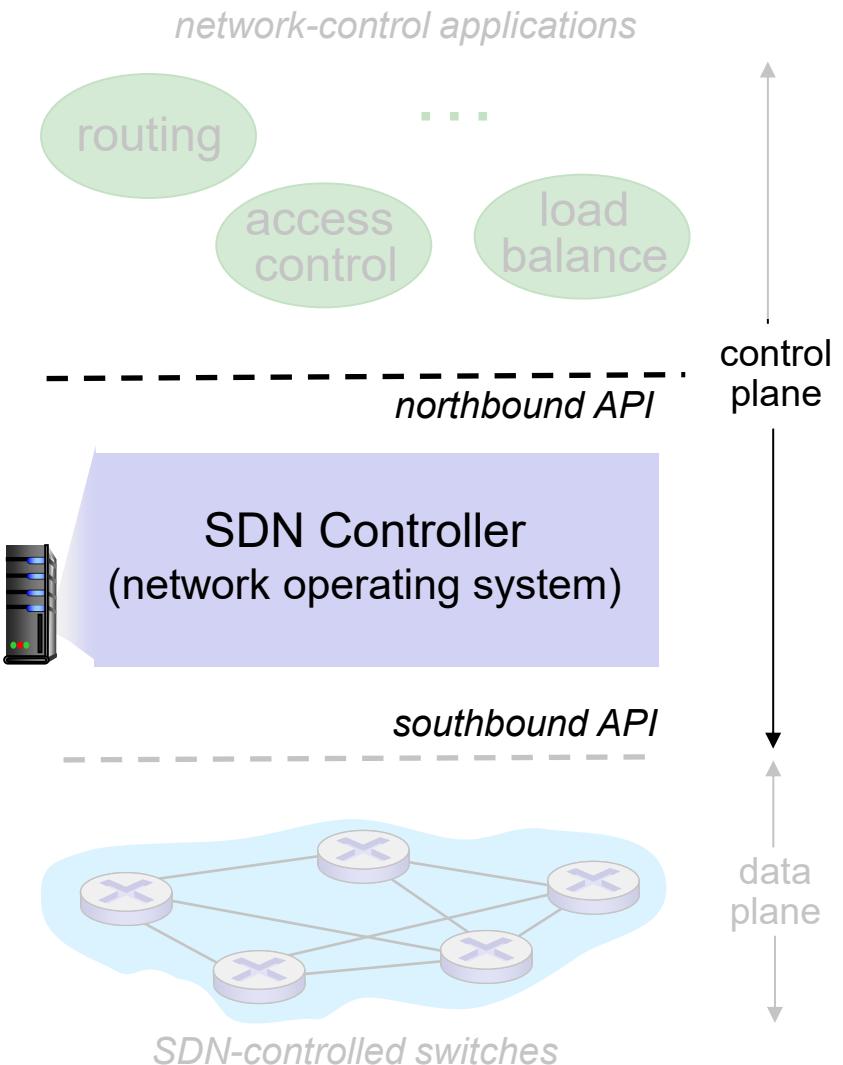




Mạng được xác định bằng phần mềm (SDN)

Bộ điều khiển SDN (hệ điều hành mạng):

- Duy trì thông tin trạng thái mạng
- Tương tác với các ứng dụng điều khiển mạng “ở trên” thông qua API hướng northbound
- Tương tác với các thiết bị chuyển mạch mạng “bên dưới” thông qua API southbound
- Được triển khai dưới dạng hệ thống phân tán cho hiệu suất, khả năng mở rộng, khả năng chịu lỗi, độ bền cao

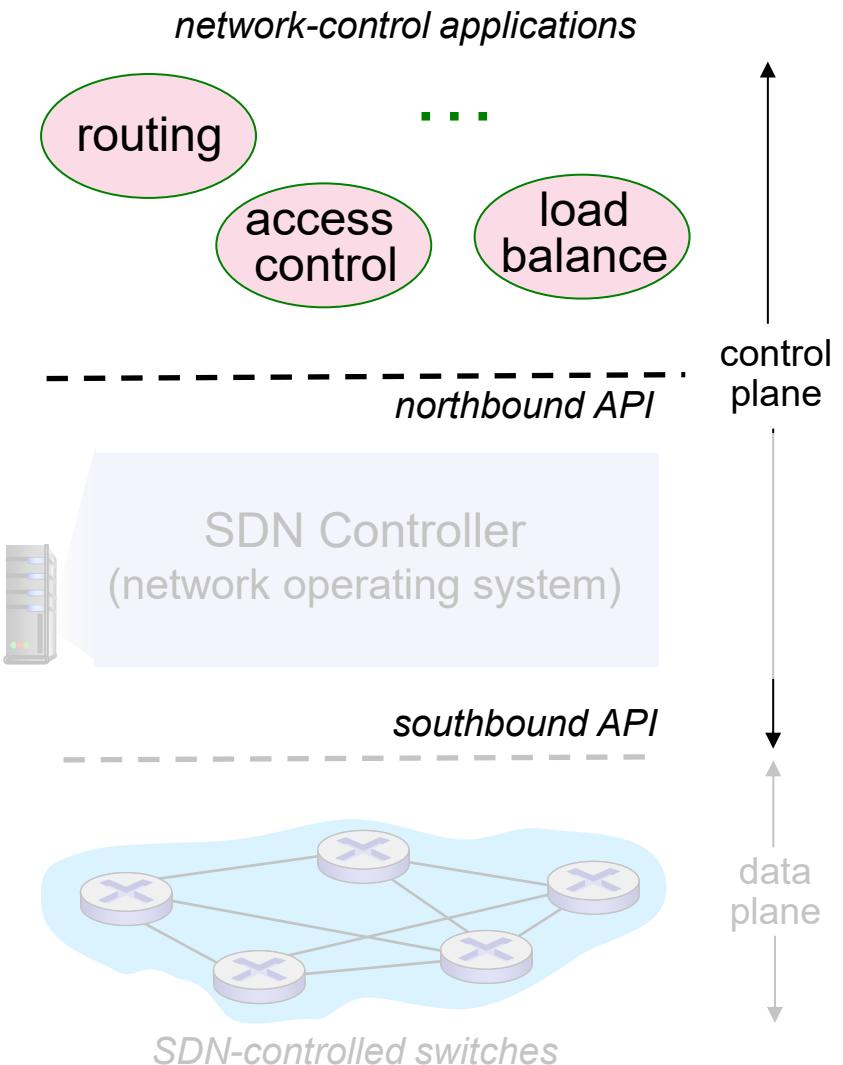




Mạng được xác định bằng phần mềm (SDN)

Ứng dụng điều khiển mạng:

- “bộ não” điều khiển: thực hiện các chức năng điều khiển bằng các dịch vụ cấp thấp hơn, API do bộ điều khiển SDN cung cấp
- Không được nhóm*: có thể được cung cấp bởi bên thứ 3: khác với nhà cung cấp định tuyến hoặc bộ điều khiển SDN



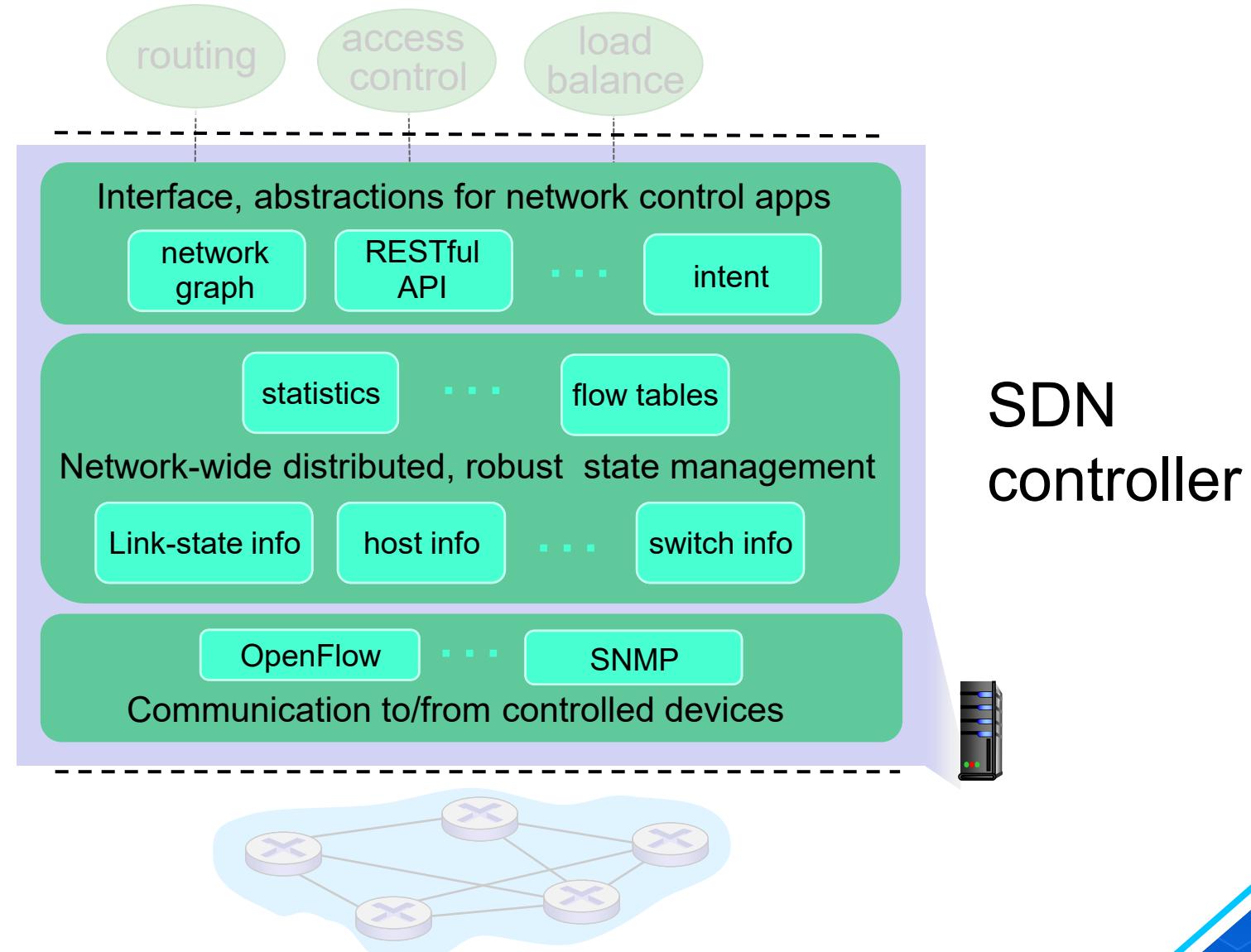


Các thành phần của bộ điều khiển SDN

Lớp giao diện cho các ứng dụng điều khiển mạng: API trừu tượng

Quản lý trạng thái toàn mạng : trạng thái của các liên kết mạng, chuyển mạch, dịch vụ: *cơ sở dữ liệu phân tán*

Truyền thông: giao tiếp giữa bộ điều khiển SDN và các bộ chuyển mạch được điều khiển

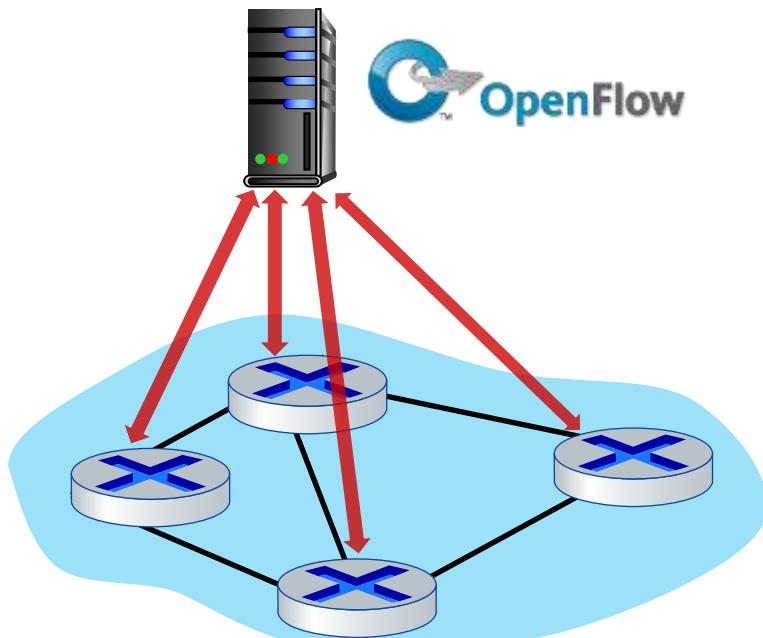


SDN
controller



- Hoạt động giữa bộ điều khiển, bộ chuyển đổi
- TCP dùng để trao đổi thông điệp
 - Mã hóa tùy chọn
- Ba lớp thông báo OpenFlow:
 - Bộ điều khiển sang bộ chuyển đổi
 - Không đồng bộ (chuyển đổi sang bộ điều khiển)
 - Đối xứng
- Khác biệt với OpenFlow API
 - API được sử dụng để chỉ định các hành động chuyển tiếp tổng quát

Bộ điều khiển OpenFlow



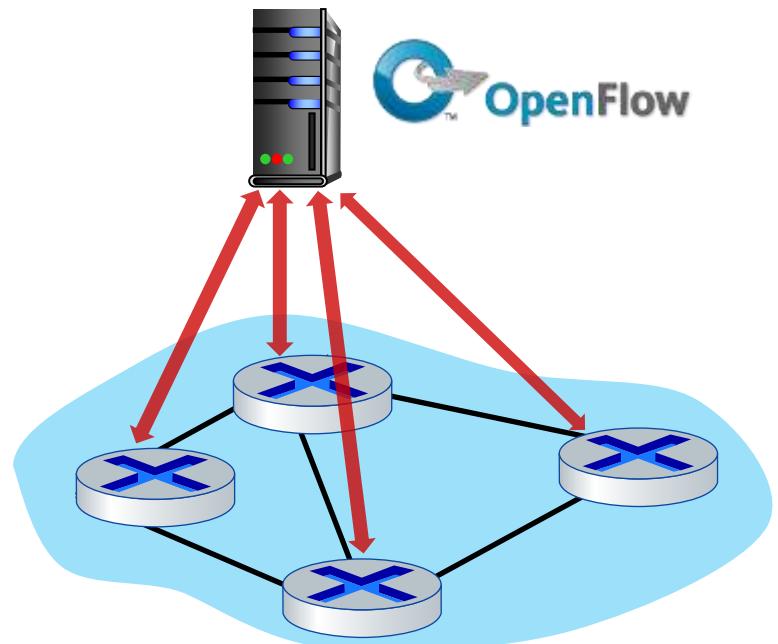
OpenFlow: thông điệp từ bộ điều khiển đến bộ chuyển đổi



Tin nhắn từ bộ điều khiển đến bộ chuyển đổi

- **Các tính năng:** các tính năng chuyển đổi truy vấn bộ điều khiển, trả lời chuyển đổi
- **Cấu hình:** các truy vấn điều khiển/thiết đặt tham số cấu hình chuyển đổi
- **Trạng thái sửa đổi:** thêm, xóa, sửa đổi các mục nhập luồng trong các bảng OpenFlow
- **Packet-out:** bộ điều khiển có thể gửi gói tin ra khỏi cổng chuyển đổi cụ thể

Bộ điều khiển OpenFlow



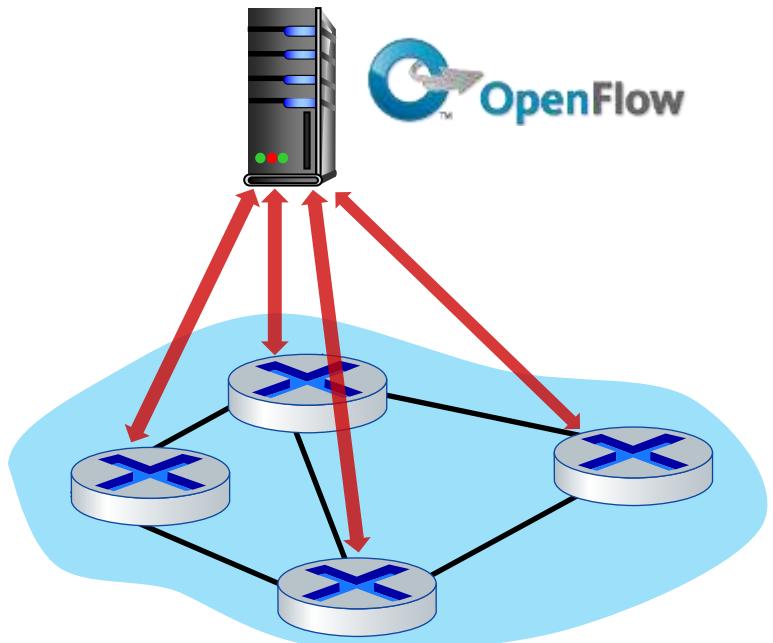
OpenFlow: thông điệp từ bộ chuyển đổi sang bộ điều khiển



Thông điệp từ bộ chuyển đổi sang bộ điều khiển

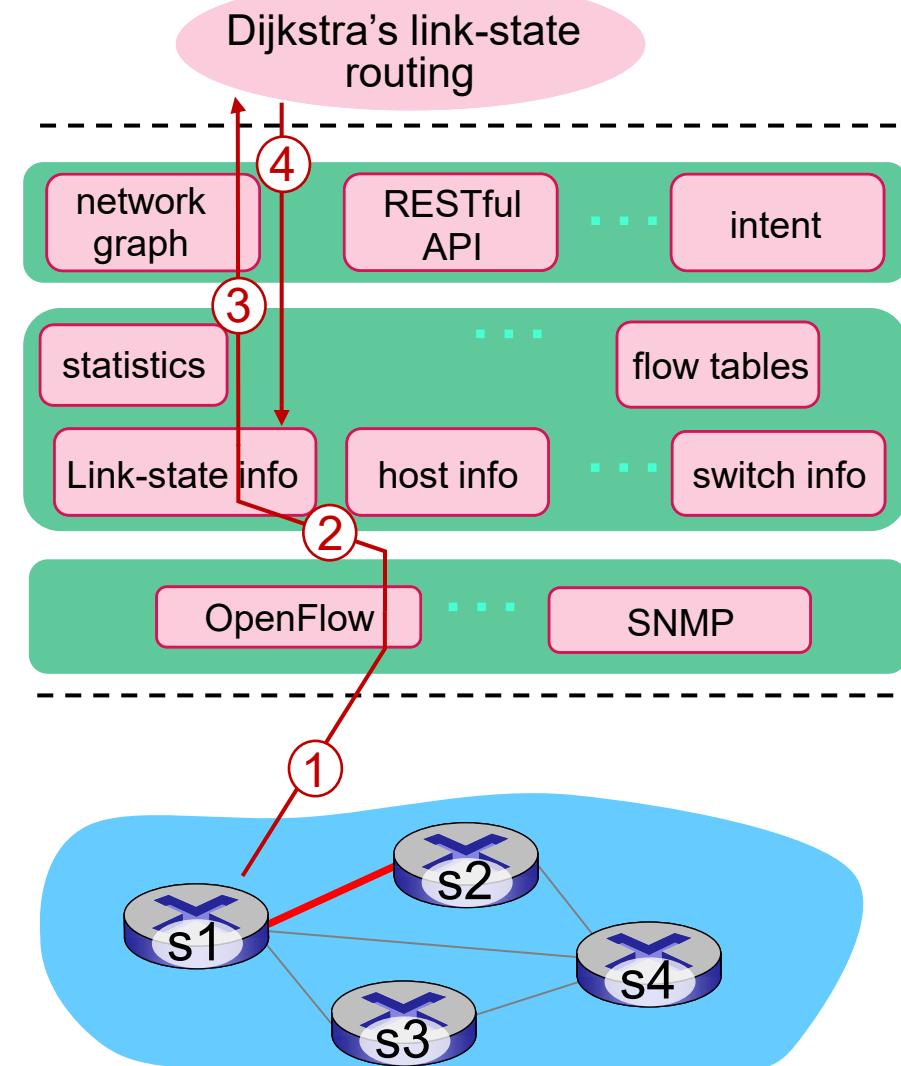
- **Packet-in:** chuyển gói (và điều khiển của nó) tới bộ điều khiển
- **Loại bỏ luồng:** mục nhập tương ứng bị xóa tại bộ chuyển đổi
- **Trạng thái cổng:** thông báo cho bộ điều khiển về sự thay đổi trên cổng

Bộ điều khiển OpenFlow



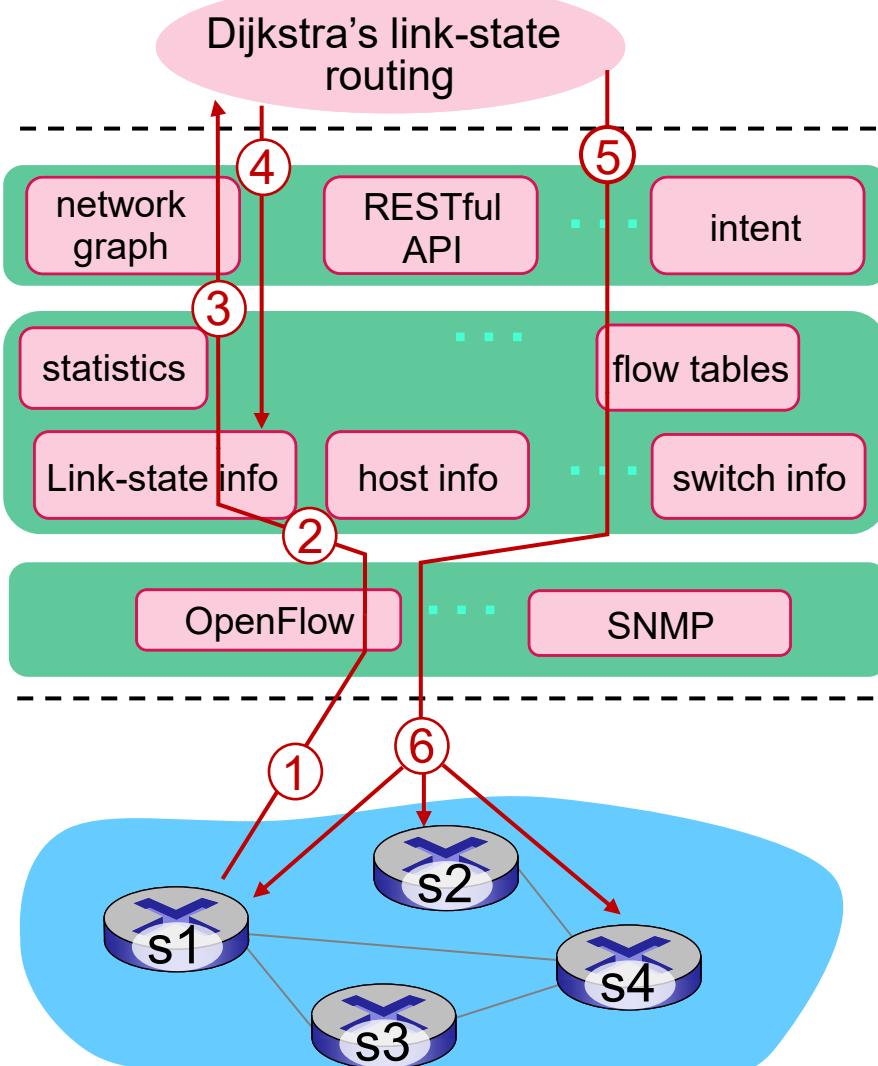
May mắn thay, các nhà khai thác mạng không "lập trình" chuyển mạch bằng cách tạo/gửi trực tiếp các tin nhắn OpenFlow. Thay vào đó là sử dụng trừu tượng hóa ở cấp cao hơn tại bộ điều khiển

SDN: ví dụ tương tác mức điều khiển/mức dữ liệu



- ① S1, thông báo trạng thái cổng OpenFlow để thông báo cho bộ điều khiển
- ② Bộ điều khiển SDN nhận thông báo OpenFlow, cập nhật thông tin trạng thái liên kết
- ③ Ứng dụng thuật toán định tuyến của Dijkstra trước đây đã đăng ký để gọi khi trạng thái liên kết thay đổi.
- ④ Thuật toán định tuyến của Dijkstra truy cập thông tin biểu đồ mạng, thông tin trạng thái liên kết trong bộ điều khiển, tính toán các tuyến mới

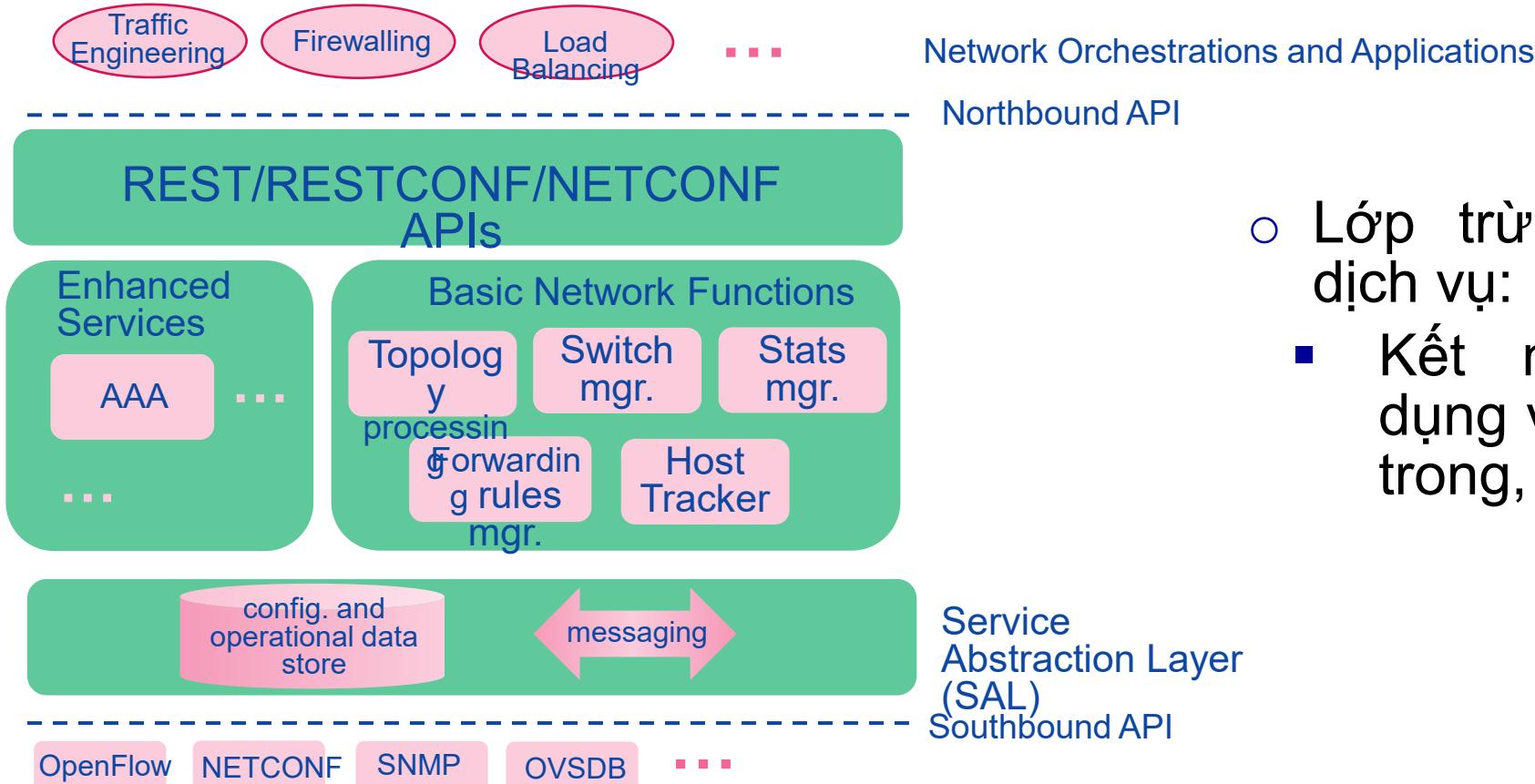
SDN: ví dụ tương tác mức điều khiển/mức dữ liệu



- ⑤ Ứng dụng định tuyến link state tương tác với thành phần tính toán của bảng lưu lượng trong bộ điều khiển SDN, tính toán các bảng lưu lượng mới cần thiết
- ⑥ Bộ điều khiển sử dụng OpenFlow để cài đặt các bảng mới trong các bộ chuyển đổi cần cập nhật



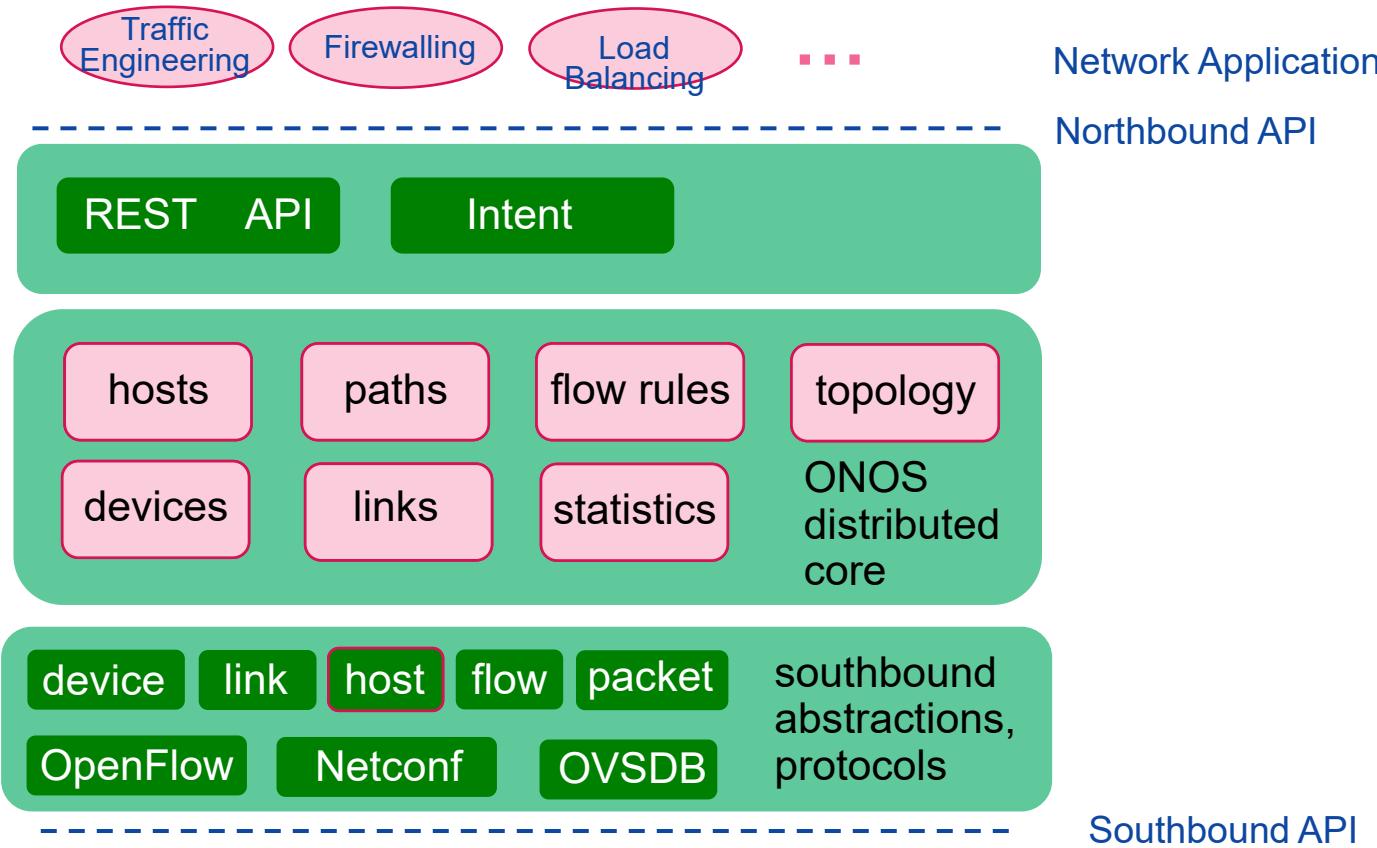
Bộ điều khiển OpenDaylight (ODL)



- Lớp trung tâm hóa dịch vụ:
 - Kết nối các ứng dụng và dịch vụ bên trong, bên ngoài



Bộ điều khiển ONOS



- Ứng dụng điều khiển tách biệt với bộ điều khiển
- Khung ý tưởng: đặc điểm kỹ thuật cấp cao của dịch vụ: cái gì là hơn và như thế nào
- Nhấn mạnh đáng kể vào lõi phân tán: độ tin cậy của dịch vụ, nhân rộng hiệu suất sao chép



- Củng cố mức điều khiển: độc lập, đáng tin cậy, có thể mở rộng hiệu suất, bảo mật cho hệ thống phân tán
 - Tận dụng lý thuyết về hệ thống phân tán tin cậy cho mức điều khiển
 - Độ tin cậy, bảo mật
- Mạng, giao thức đáp ứng yêu cầu nhiệm vụ cụ thể
 - Ví dụ: thời gian thực, cực kỳ đáng tin cậy, cực kỳ an toàn
- Mở rộng quy mô Internet: ngoài một AS
- SDN quan trọng trong mạng di động 5G

SDN và tương lai của các giao thức mạng truyền thống



- SDN tính toán các bảng chuyển tiếp router-máy tính:
 - Chỉ là một ví dụ về tính toán logic-tập trung so với tính toán giao thức
- Người ta có thể tưởng tượng kiểm soát tắc nghẽn được tính toán bằng SDN:
 - Bộ điều khiển đặt tốc độ người gửi dựa trên mức tắc nghẽn do router báo cáo (đến bộ điều khiển)



Việc triển khai chức năng mạng (SDN so với các giao thức) sẽ phát triển như thế nào?





Quản lý mạng là gì

- Hệ thống tự quản (hay còn gọi là “mạng”): 1000 thành phần phần cứng/phần mềm tương tác
- Các hệ thống phức tạp khác yêu cầu giám sát, cấu hình, kiểm soát:
 - Máy bay phản lực, nhà máy điện hạt nhân, những thứ khác?



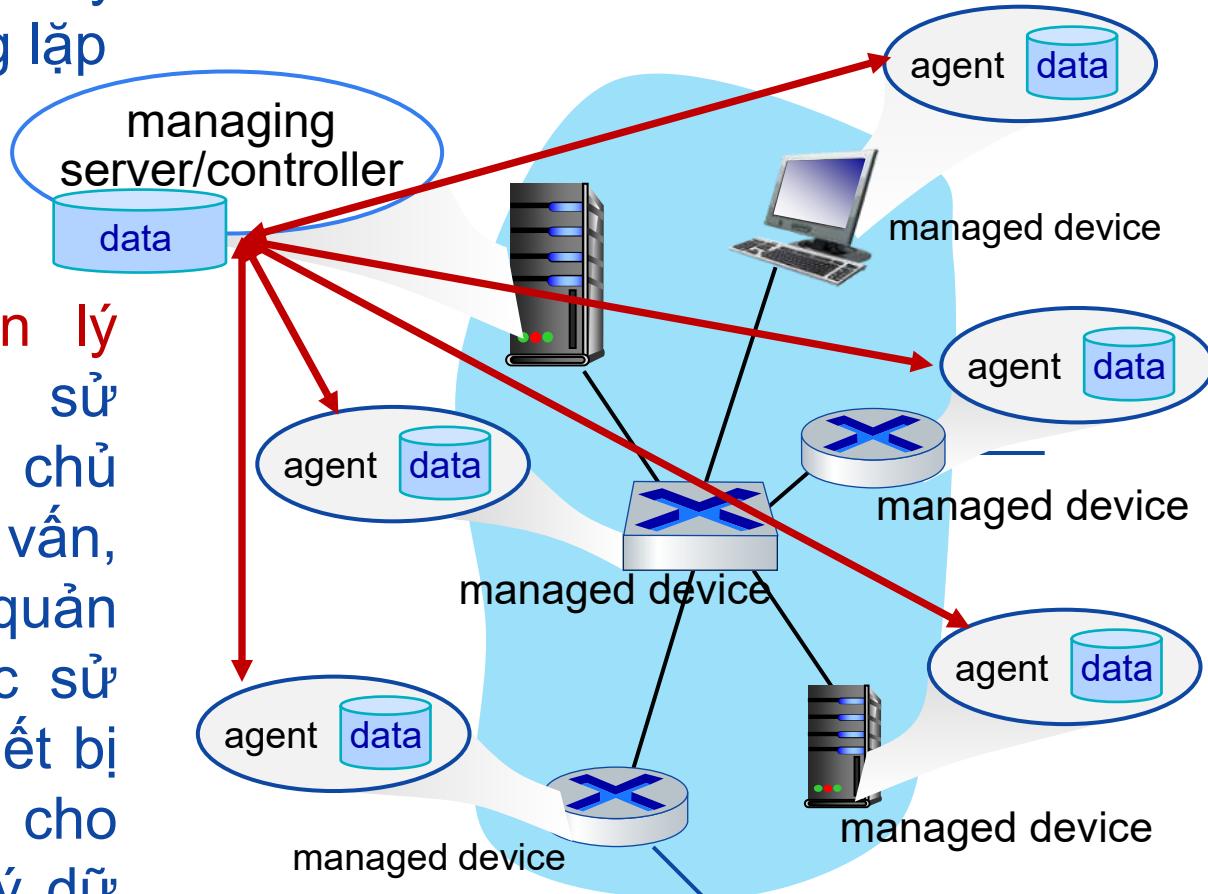
" Quản lý mạng bao gồm việc triển khai, tích hợp và phối hợp của phần cứng, phần mềm và con người, các yếu tố để giám sát, kiểm tra, thăm dò ý kiến, cấu hình, phân tích, đánh giá và kiểm soát mạng và tài nguyên phần tử để đáp ứng thời gian thực, hiệu suất hoạt động và chất lượng dịch vụ yêu cầu với chi phí hợp lý."



Các thành phần quản lý mạng

Quản lý máy chủ:
ứng dụng, điển hình
với người quản lý
mạng trong vòng lặp

Giao thức quản lý
mạng: được sử
dụng bởi máy chủ
quản lý để truy vấn,
định cấu hình, quản
ly thiết bị; được sử
dụng bởi các thiết bị
để thông báo cho
máy chủ quản lý dữ
liệu, sự kiện



Thiết bị được quản lý:
thiết bị có các thành
phần phần mềm, phần
cứng có thể quản lý,
có thể định cấu hình

Dữ liệu: dữ liệu cấu
hình “trạng thái” của
thiết bị, dữ liệu vận
hành, thống kê thiết
bị

Phương pháp tiếp cận cách quản lý của nhà điều hành mạng



CLI (Giao diện dòng lệnh)

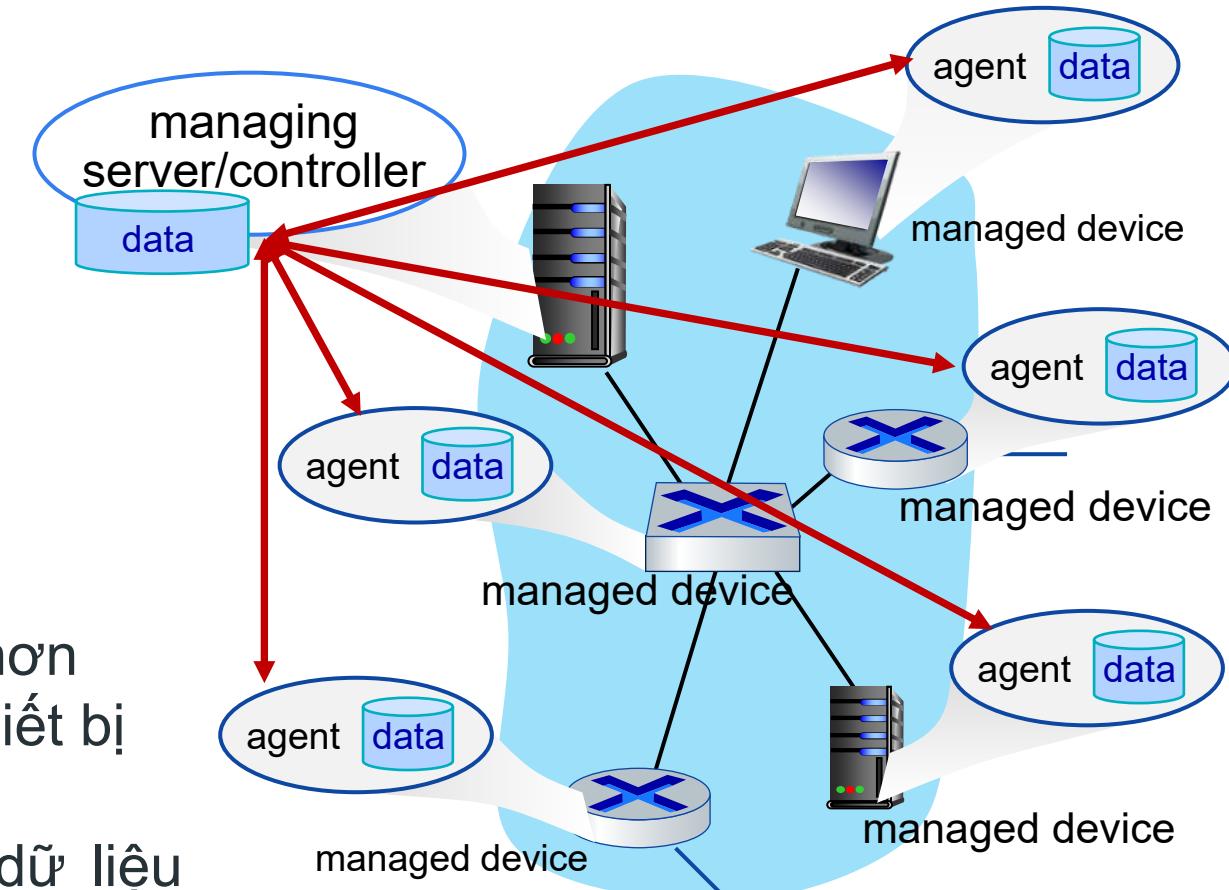
- Các sự cố của nhà điều hành (loại, tập lệnh) trực tiếp đến từng thiết bị (ví dụ: vis ssh)

SNMP/MIB

- Người vận hành truy vấn/đặt dữ liệu thiết bị (MIB) bằng Giao thức quản lý mạng đơn giản (SNMP)

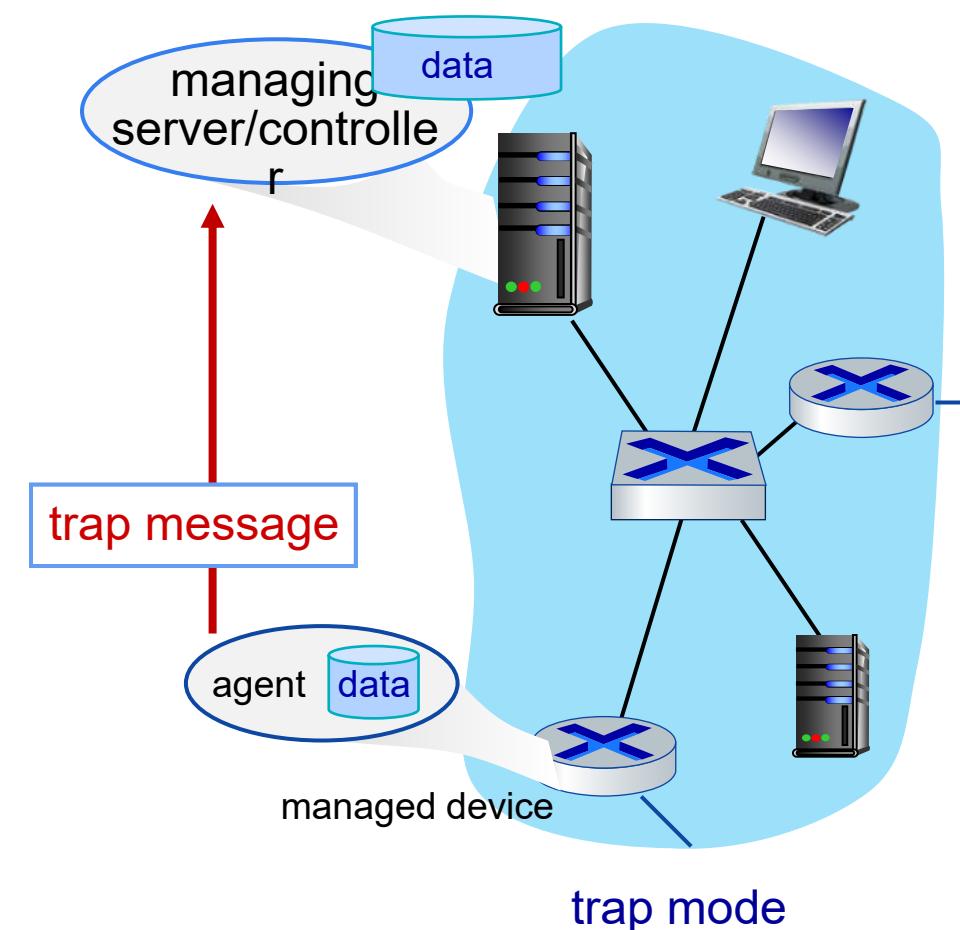
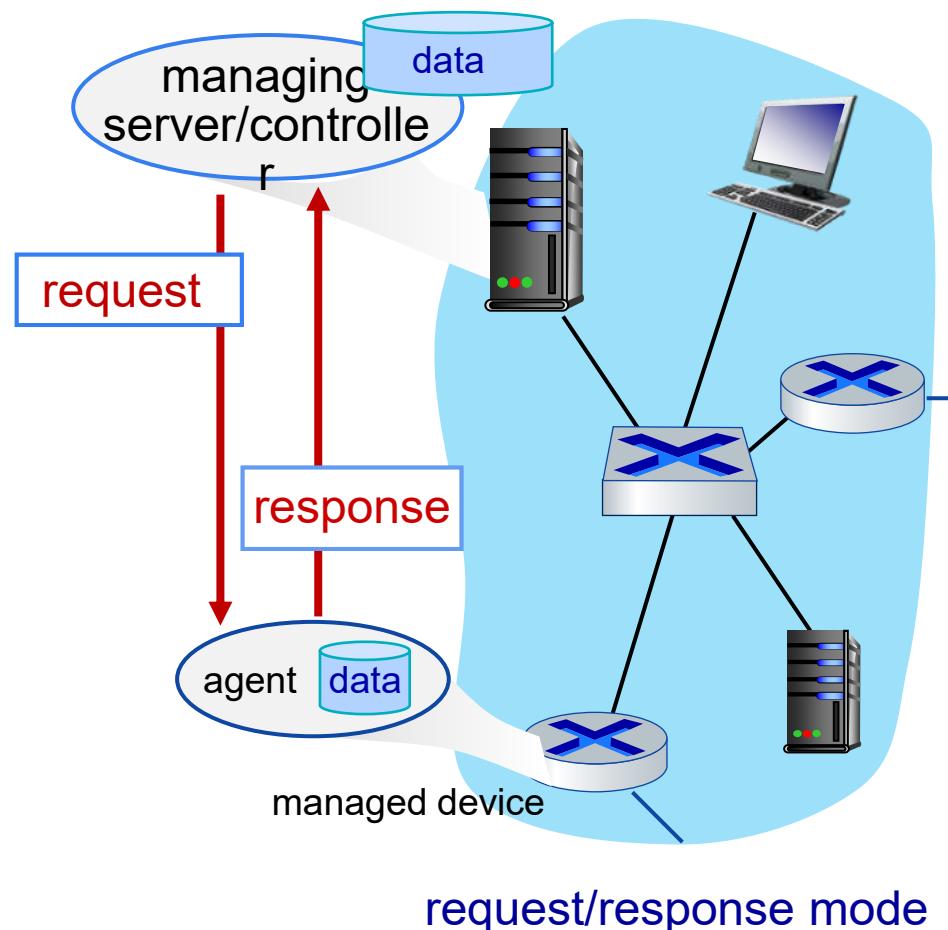
NETCONF/YANG

- Trùu tượng hơn, rộng hơn, toàn diện hơn
- Nhấn mạnh vào quản lý cấu hình đa thiết bị
- YANG: ngôn ngữ mô hình hóa dữ liệu
- NETCONF: giao tiếp các hành động/dữ liệu tương thích với YANG đến/từ/giữa các thiết bị từ xa





Hai cách để truyền tải thông tin MIB, lệnh:



Giao thức SNMP: các loại thông báo

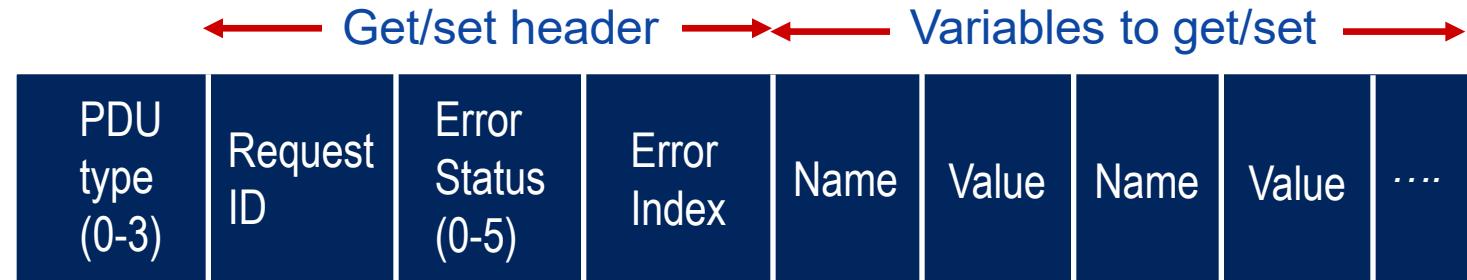


Loại thông điệp	Chức năng
GetRequest GetNextRequest GetBulkRequest	manager-to-agent: “ lấy dữ liệu cho tôi ” (ví dụ dữ liệu, dữ liệu tiếp theo trong danh sách, khối dữ liệu).
SetRequest	manager-to-agent: đặt giá trị MIB
Reponse	Agent-to-manager: giá trị, phản hồi Yêu cầu
Trap	Agent-to-manager: thông báo cho người quản lý sự kiện đặc biệt

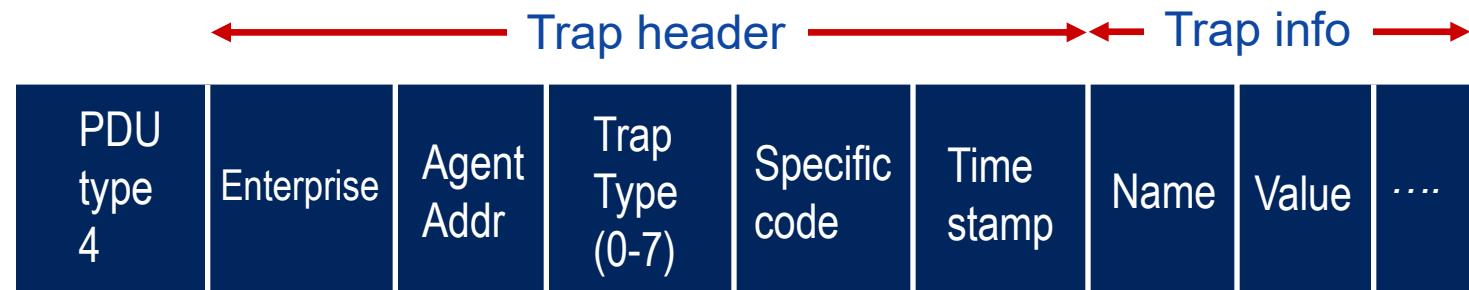


Giao thức SNMP: định dạng thông điệp

Thông điệp type 3



Thông điệp type 4

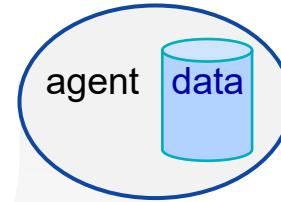


← → SNMP PDU



SNMP: Cơ sở quản lý thông tin (MIB)

- Dữ liệu hoạt động (và một số cấu hình) của thiết bị được quản lý
- Được tập hợp vào mô-đun MIB của thiết bị
 - 400 mô-đun MIB được xác định trong RFC's; nhiều MIB dành riêng cho nhà cung cấp hơn
 - **Cấu trúc thông tin quản lý (SMI):** ngôn ngữ định nghĩa dữ liệu

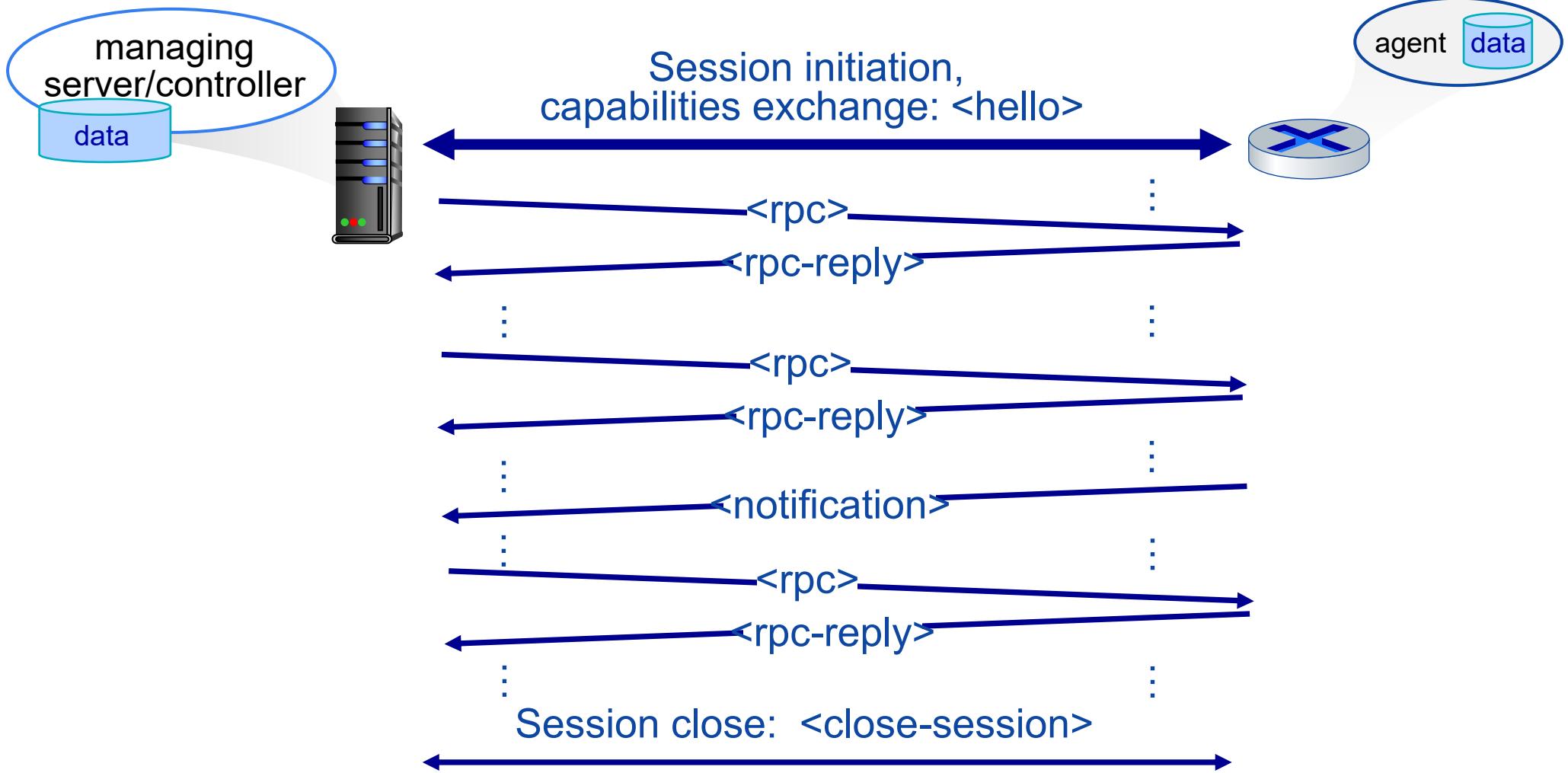




Tổng quan về NETCONF

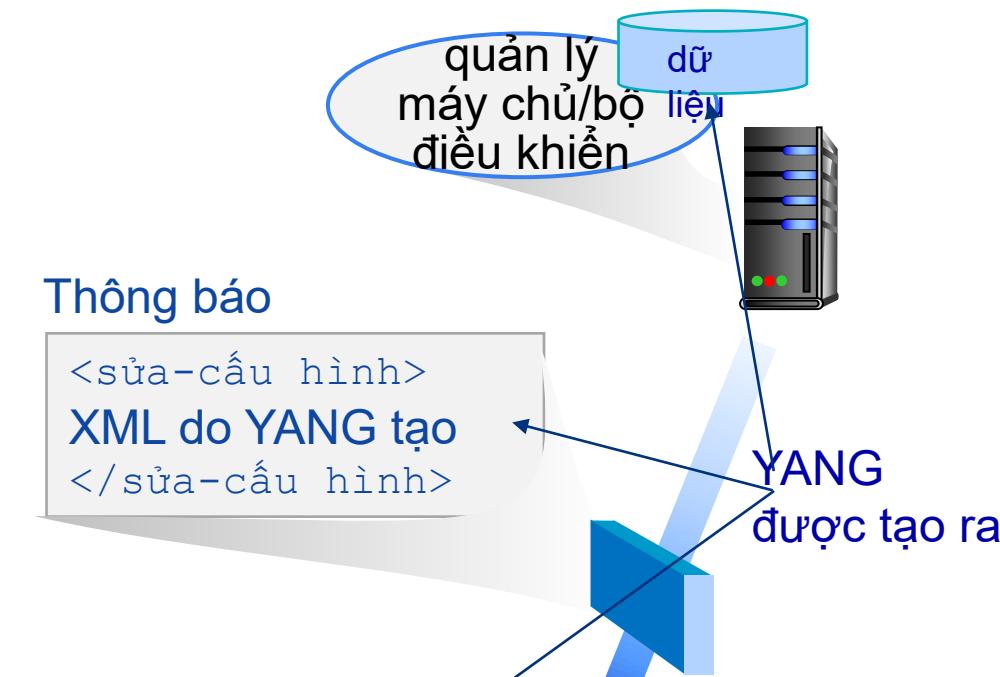
- Mục tiêu: chủ động quản lý/ cấu hình thiết bị trên toàn mạng
- Hoạt động giữa máy chủ quản lý và thiết bị mạng được quản lý
 - Hành động: truy xuất, đặt, sửa đổi, kích hoạt cấu hình
 - Hoạt động được cam kết trên nhiều thiết bị
 - Truy vấn dữ liệu hoạt động và thống kê
 - Đăng ký nhận thông báo từ các thiết bị
- Mô hình gọi thủ tục từ xa (RPC)
 - Thông báo giao thức NETCONF được mã hóa bằng XML
 - Giao thức vận chuyển an toàn, đáng tin cậy (ví dụ: TLS)

NETCONF khởi tạo, trao đổi, đóng phiên





- Ngôn ngữ mô hình hóa dữ liệu dùng để chỉ định cấu trúc, cú pháp, ngữ nghĩa của dữ liệu quản lý mạng NETCONF
 - Các kiểu dữ liệu tích hợp, như SMI
- Tài liệu XML mô tả thiết bị, các khả năng có thể được tạo từ mô tả YANG
- Có thể thể hiện các ràng buộc giữa các dữ liệu phải được thỏa mãn bởi cấu hình NETCONF hợp lệ
 - Đảm bảo các cấu hình NETCONF đáp ứng các ràng buộc về tính chính xác, nhất quán





Chương 4: Nội dung

4.1 Tổng quan

- Mức dữ liệu
- Mức điều khiển

4.2 Virtual circuit network và datagram network

- Các mạch ảo
- Mạng datagram

4.3 Cấu trúc bên trong router

- Cổng đầu vào, chuyển đổi, cổng đầu ra
- Quản lý bộ đệm, lập kế hoạch

4.4 IP: Internet Protocol

- Định dạng datagram
- IPv4 addressing

- DHCP
- NAT
- ICMP
- IPV6

4.5 Các thuật toán routing

- Link state
- Distance vector

4.6 Routing trong Internet

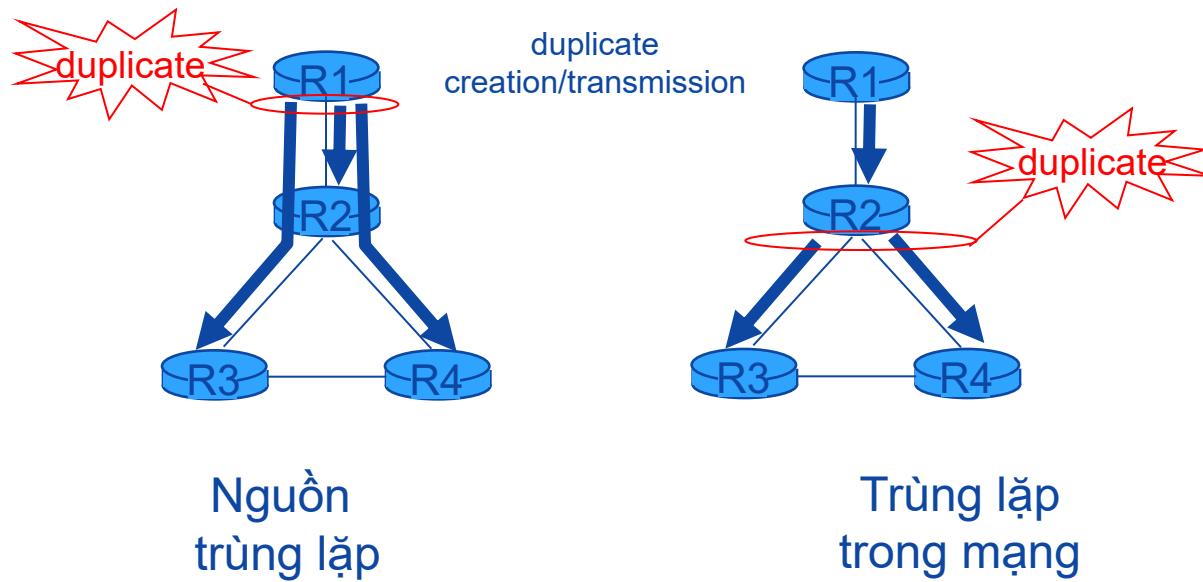
- Hierarchical routing
- RIP, OSPF, BGP
- SDN
- Quản lý mạng

4.7 Broadcast và Multicast routing



Broadcast routing

- Chuyển các packet 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 lắp trong mạng

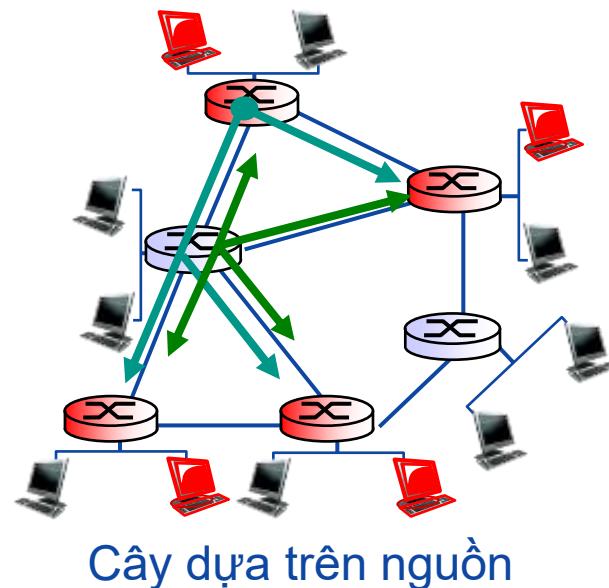
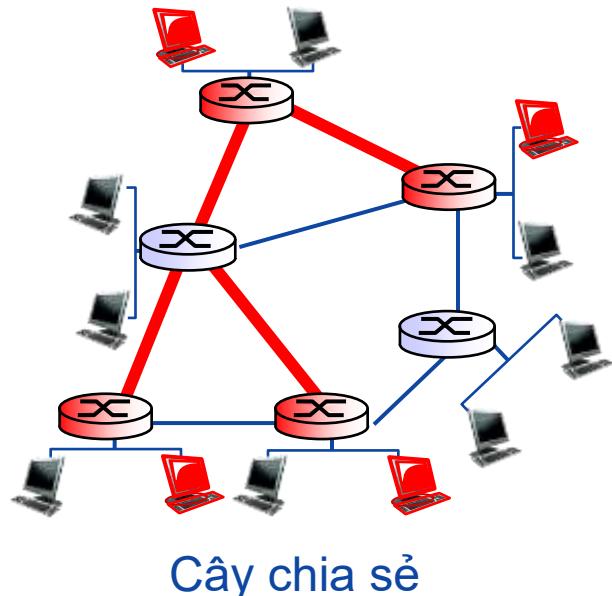
- *Flooding*: khi node nhận được packet broadcast, nó gởi bản sao đến tất cả các neighbor
 - Vấn đề: lặp lại & bão broadcast
- *Flooding có điều khiển*: node chỉ broadcast packet nếu nó không gửi broadcast giống như vậy trước đó
 - Node theo dõi các packet đã broadcast
 - Hoặc reverse path forwarding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

Multicast routing: phát biểu vắn 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



Ký hiệu	
	Group member
	not group member
	Router with a Group member
	Router without group member

Hướng 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
 - Cây đường đi ngược
- *Cây chia sẻ nhóm*: nhóm dùng 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 các giao thức cụ thể áp dụng cho các hướng tiếp cận này



Lớp mạng: Tóm tắt

Chúng ta đã học được rất nhiều!

- Tiếp cận mức điều khiển mạng
 - Điều khiển trên mỗi router (truyền thống)
 - Điều khiển tập trung hợp lý (mạng được xác định bằng phần mềm)
- Thuật toán định tuyến truyền thống
 - Triển khai trên Internet: OSPF , BGP
- Bộ điều khiển SDN
 - Triển khai trong thực tế: ODL, ONOS
- Quản lý mạng

điểm dừng tiếp theo: lớp liên kết!

Danh sách từ Anh – Việt trong chương 4



- Broadcast: quảng bá
- Control plane: mức điều khiển
- Datagram: gói tin
- Data plane: mức dữ liệu
- Fabric: cấu tạo, tổ chức
- Forwarding: chuyển tiếp
- Flooding: làm tràn ngập
- Interface: giao tiếp, cổng
- Multicast: đa hướng
- Neighbor: hàng xóm lân cận
- Network layer: tầng mạng
- Packet: gói tin
- Router: bộ định tuyến
- Routing: định tuyến
- Switching: chuyển mạch
- Virtual circuit network: mạng mạch ảo