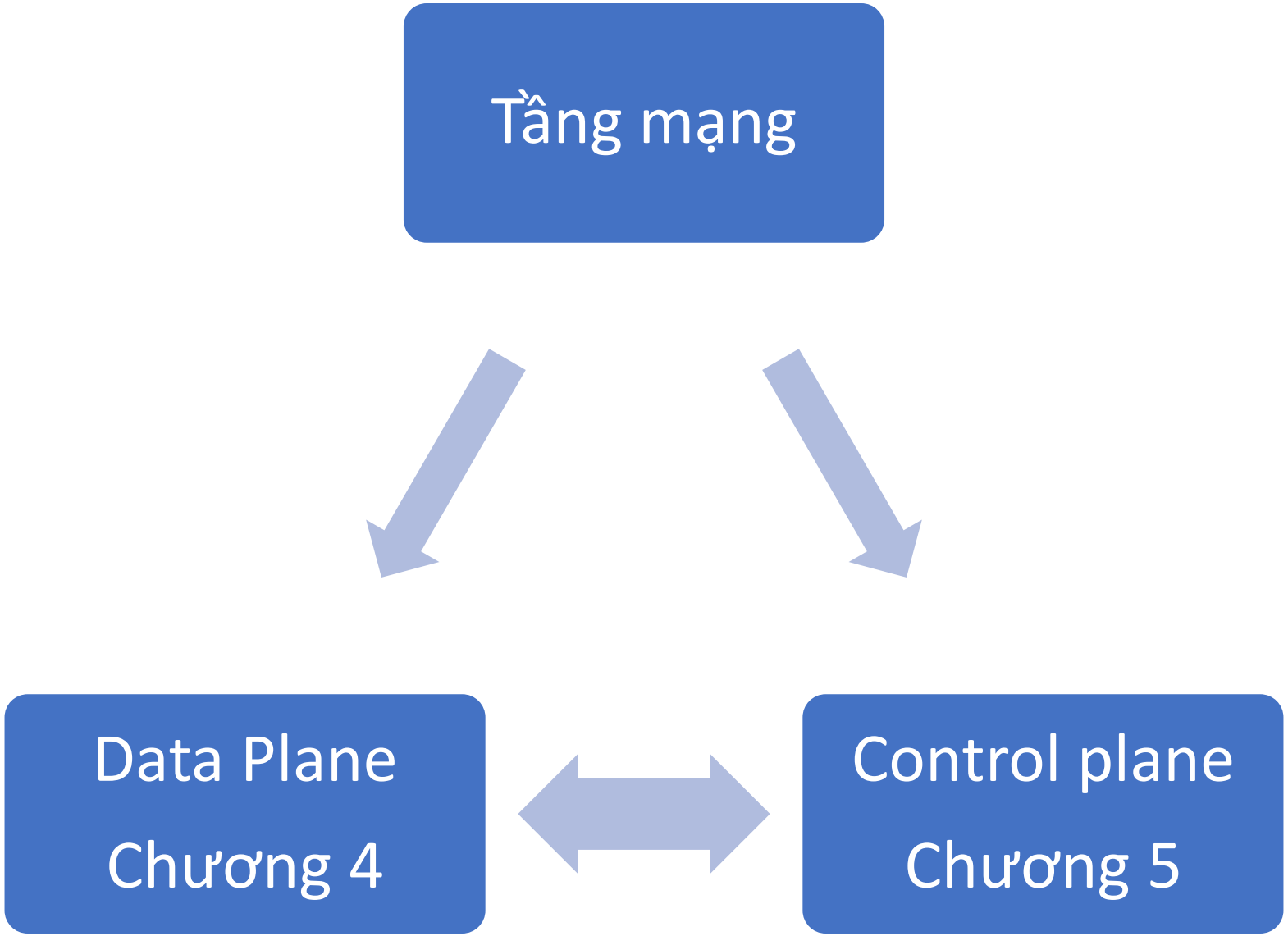
Chương 4&5
Tầng mạng

ThS. Cấn Thị Phượng


Tầng mạng
Data Plane
Chương 4
Control plane
Chương 5

Tài liệu tham khảo

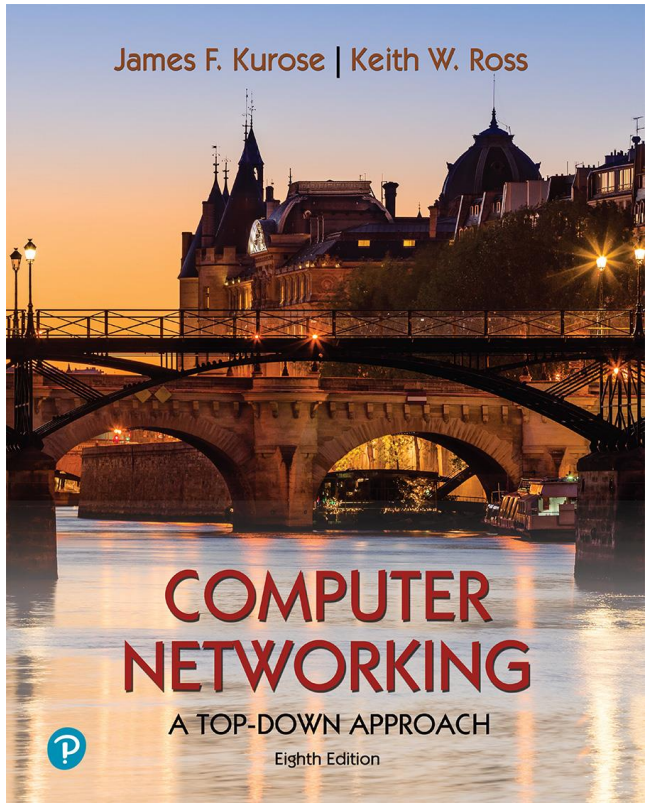
A note on the use of these PowerPoint slides:

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



Computer Networking: A Top-Down Approach

8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Network layer: Mục tiêu

- Hiểu các nguyên lý các dịch vụ tầng mạng tại data plane:
  - Dịch vụ tầng mạng
  - forwarding versus routing
  - Router làm việc
  - Địa chỉ
  - generalized forwarding
  - Kiến trúc Internet
- Hiện tại tầng mạng trên Internet
  - IP protocol
  - NAT, middleboxes

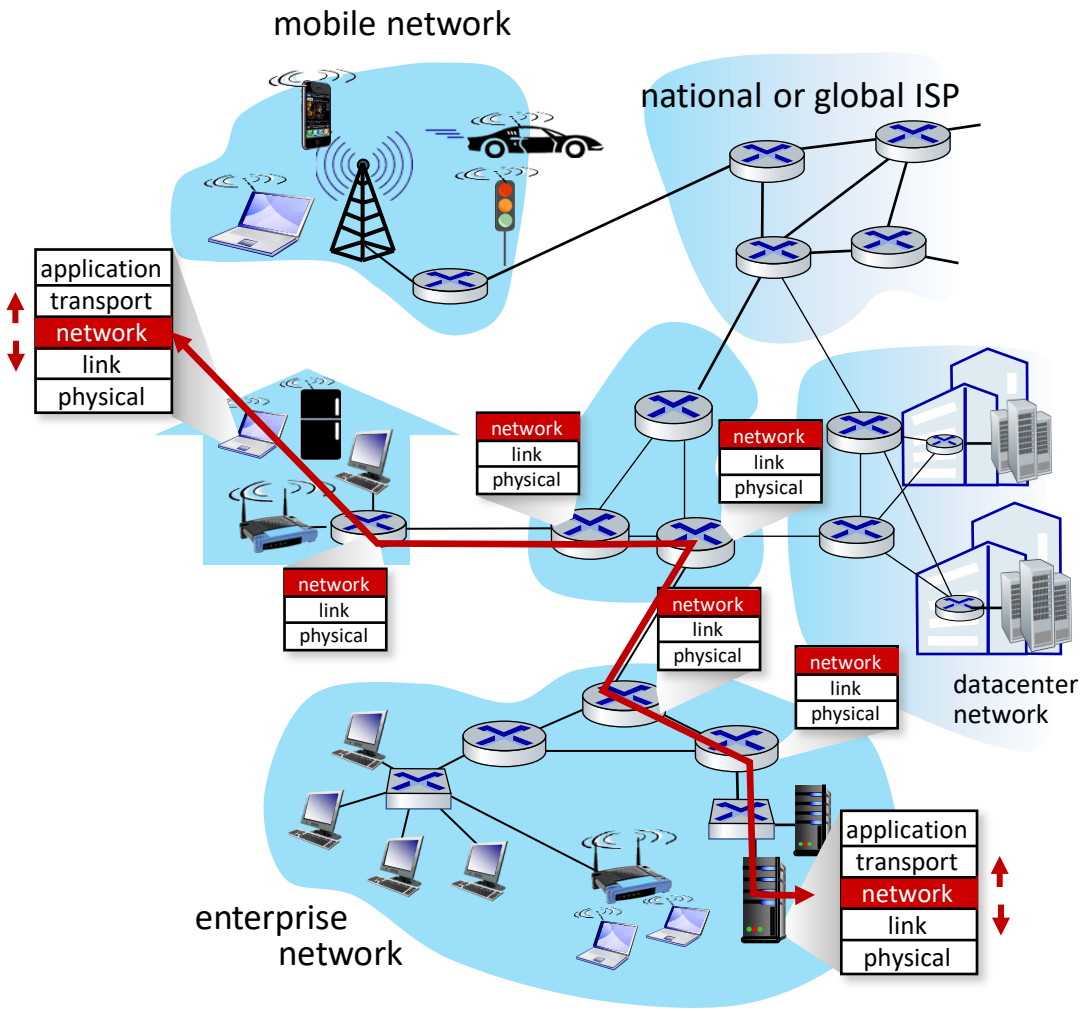
Network layer: “data plane”

- Network layer: tổng quan
  - data plane
  - control plane
- Cái gì trong router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - Định dạng datagram
  - Địa chỉ
  - network address translation (NAT)
  - IPv6

- Generalized Forwarding, SDN
  - Match+action
  - OpenFlow: match+action in action
- Middleboxes

Network-layer services and protocols

- Vận chuyển segment từ host gửi tới host nhận
  - sender: đóng gói segments vào datagrams, gửi xuống link layer
  - receiver: phân phát segment tới transport layer protocol
- network layer protocols trong mọi *Internet device*: hosts, routers
- routers:
  - Kiểm tra các trường trong header trong tất cả các IP datagram.
  - Di chuyển datagrams từ input ports đến output ports để truyền datagrams dọc theo đường end-end



Hai chức năng chính của network-layer

network-layer functions:

- *forwarding:* đưa gói từ link đầu vào của router tới link đầu ra tương ứng tới router kế tiếp.
- *routing:* quyết định tuyến đường gói sẽ đi từ source đến destination
  - *routing algorithms*

Tương tự: tham gia một chuyến đi

- *forwarding:* Xử lý để đi qua một nút giao thông
- *routing:* kế hoạch chuyến đi từ nguồn tới đích

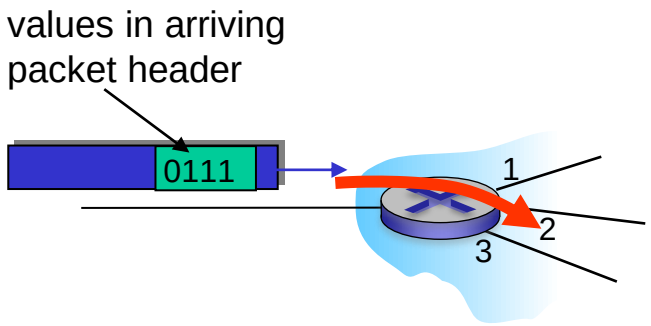
forwarding

routing

Network layer: data plane, control plane

Data plane:

- *local*, chức năng tại router
- Quyết định phương thức làm thế nào để datagram nhận trên link đầu vào sẽ được đẩy ra đến router kế tiếp

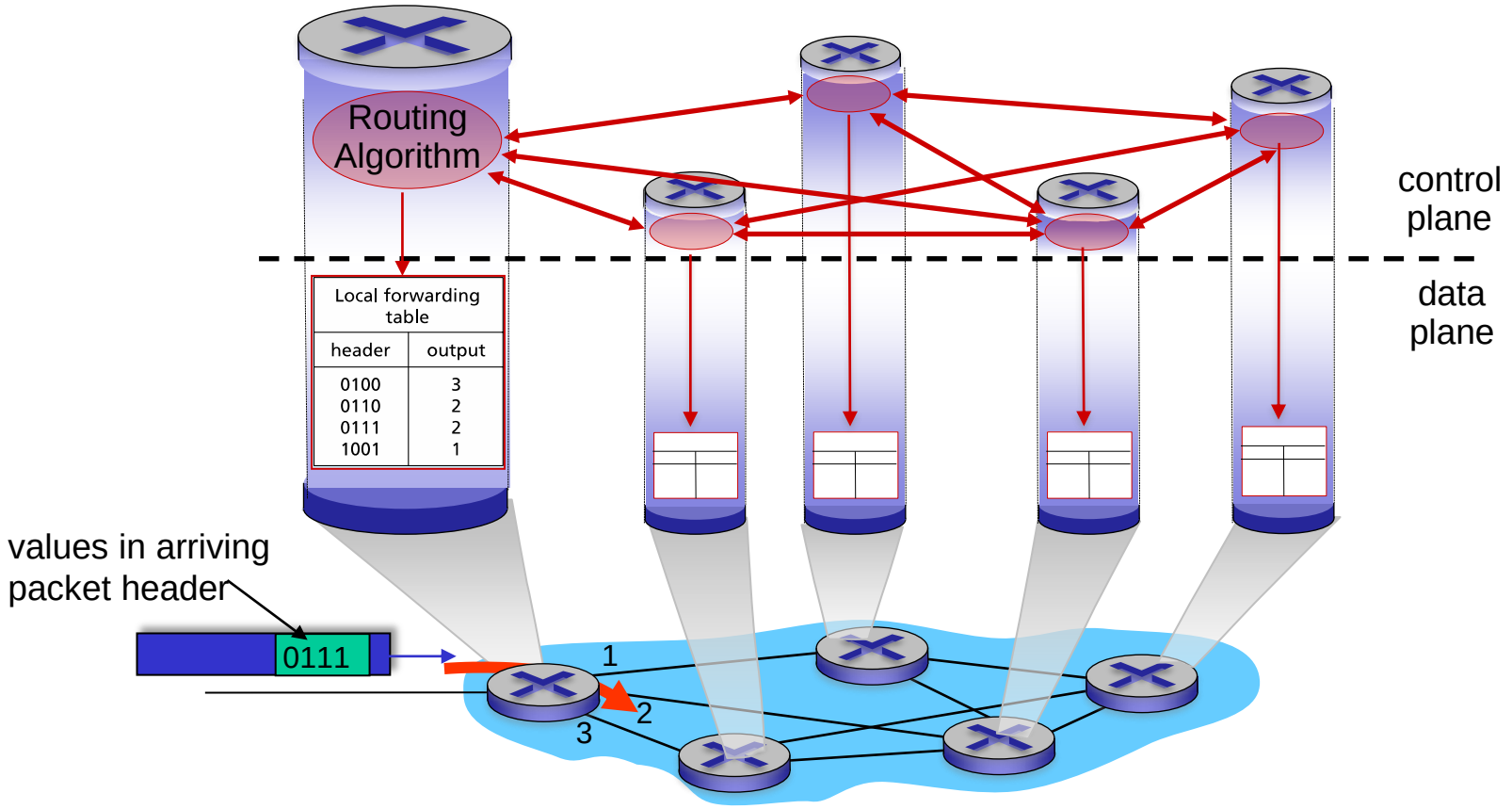


Control plane

- *network-wide* logic
- Quyết định phương thức nào để datagram được định tuyến dọc theo tuyến đường end-end từ nguồn tới đích
- Hai cách tiếp cận control-plane:
  - *traditional routing algorithms:* thi hành tại router
  - *software-defined networking (SDN)*: thực thi tại server từ xa (bộ điều phối)

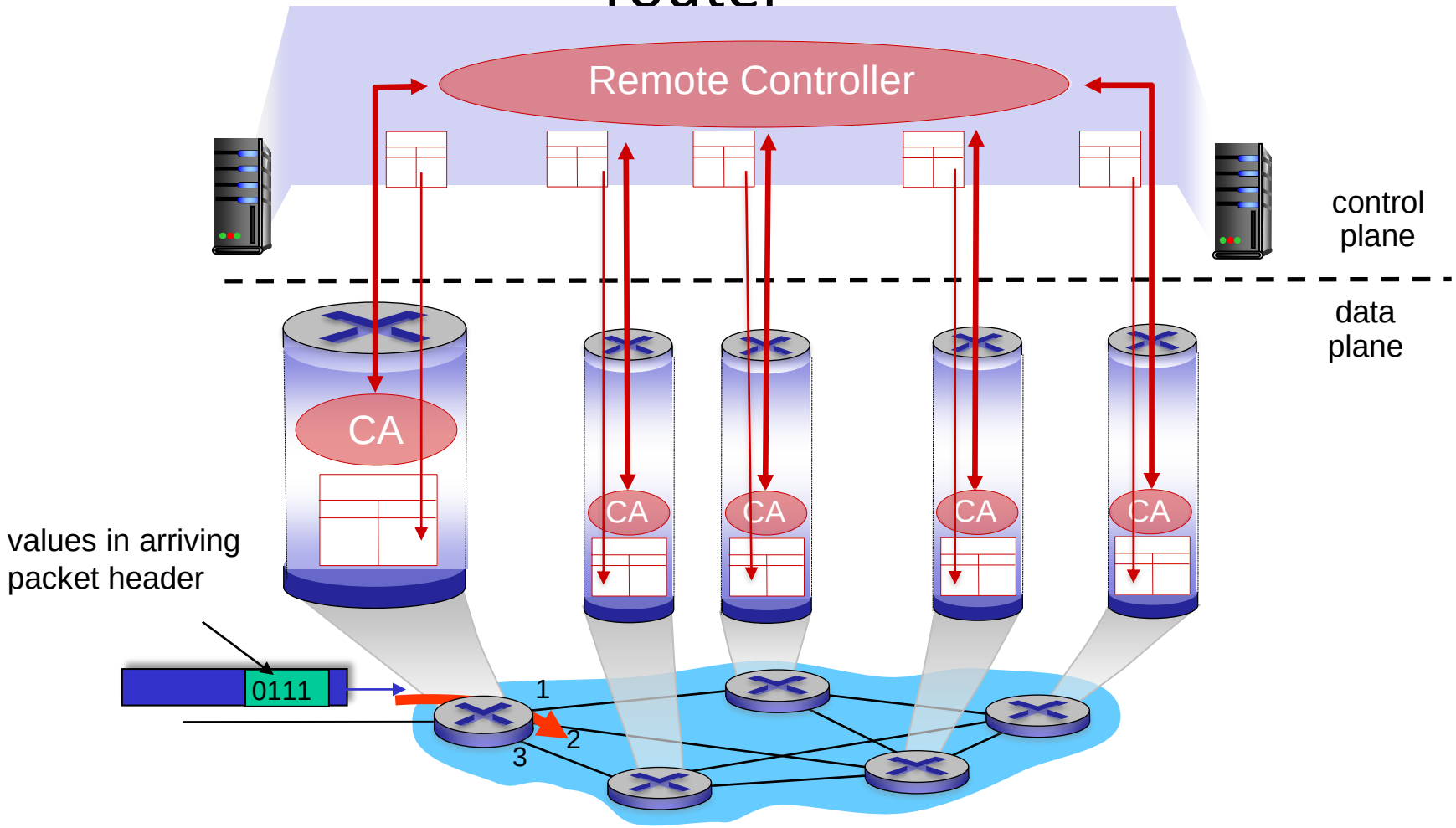
Per-router control plane

Thành phần của giải thuật định tuyến trong mỗi router và mọi router tương tác với nhau trong control plane



Software-Defined Networking (SDN) control plane

controller từ xa tính toán, cài đặt forwarding tables trong các router



Network service model

Q: mô hình dịch vụ cho "channel" truyền datagrams từ người gửi tới người nhận?

Ví dụ dịch vụ cho từng datagrams:

- Phân phát đảm bảo
- Đảm bảo độ trễ nhỏ hơn 40 msec

Ví dụ dịch vụ cho luồng datagrams:

- Phân phát datagram theo thứ tự
- Đảm bảo băng thông tối thiểu cho luồng
- Hạn chế thay đổi khoảng cách giữa các gói

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) có đảm bảo không?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no

Internet "best effort": Mô hình dịch vụ "nỗ lực tối đa"

No (không đảm bảo):

i. Phân phát datagram tới đích
ii. Thời gian hoặc thứ tự
iii. Băng thông sẵn có cho luồng end-end

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet	Diffserv (RFC 2475)	possible	possibly	possibly	no

best-effort service cho phép:

- Cơ chế đơn giản: Cho phép Internet triển khai mở rổng dễ dàng
- Cung cấp đủ băng thông: cho phép hiệu năng của các ứng dụng thời gian thưc (interactive voice, video) có thể "đủ tốt" cho mọi thời gian.
- Nhân bản dịch vụ ứng dụng phân tán (datacenters, content distribution networks) kết nối gần hơn với mạng của khách hàng, cung cấp dịch vụ nhiều nơi.
- Kiểm soát tắc nghẽn mềm dẻo

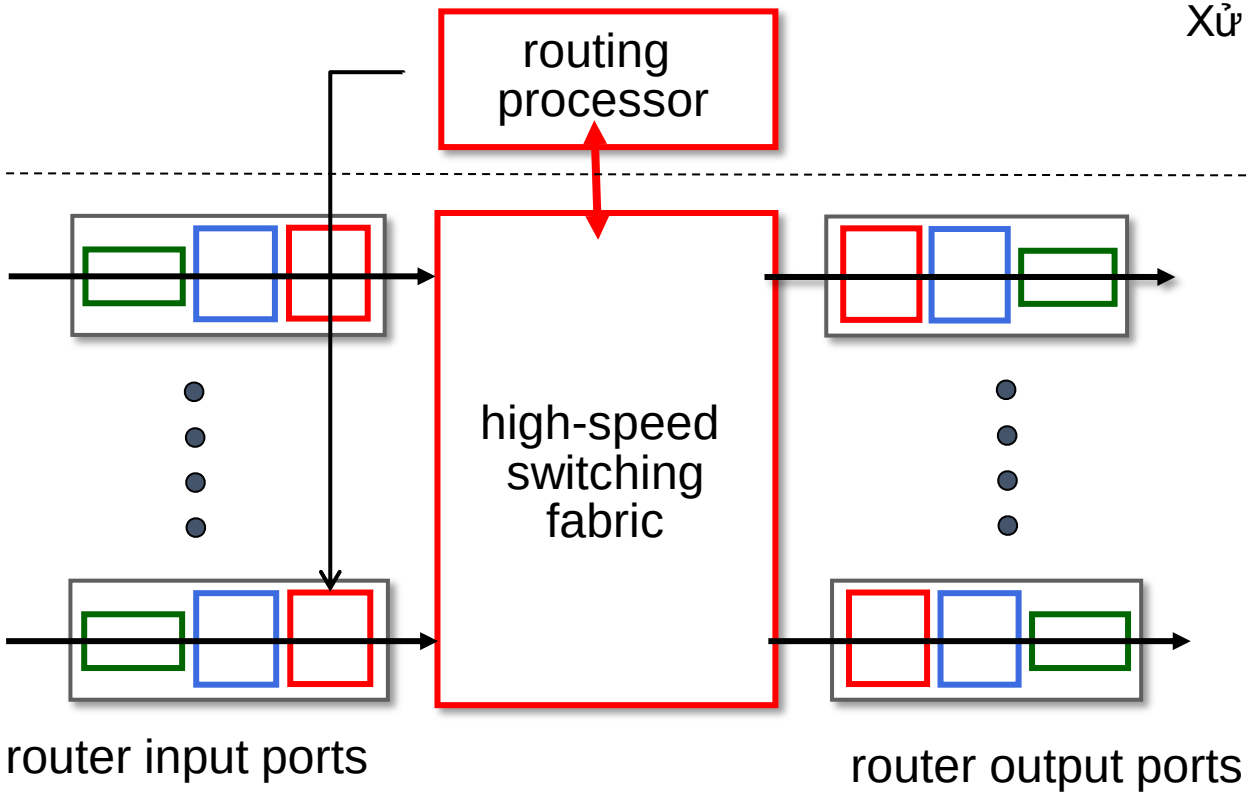
It's hard to argue with success of best-effort service model

Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- Bên trong router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized Forwarding, SDN
  - Match+action
  - OpenFlow: match+action in action
- Middleboxes

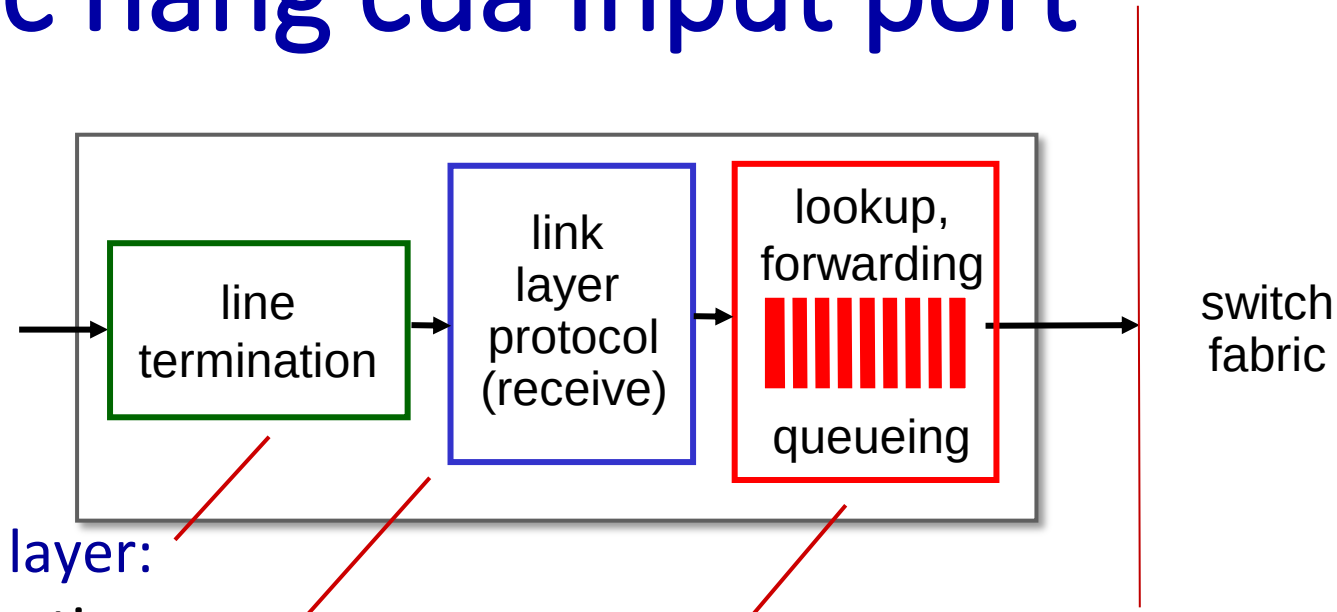
Tổng quan kiến trúc Router



routing, management control plane (software)
Xử lý theo khung thời gian tính bằng

forwarding data plane (hardware)
Xử lý theo khung thời gian tính bằng nanosecond

Chức năng của Input port



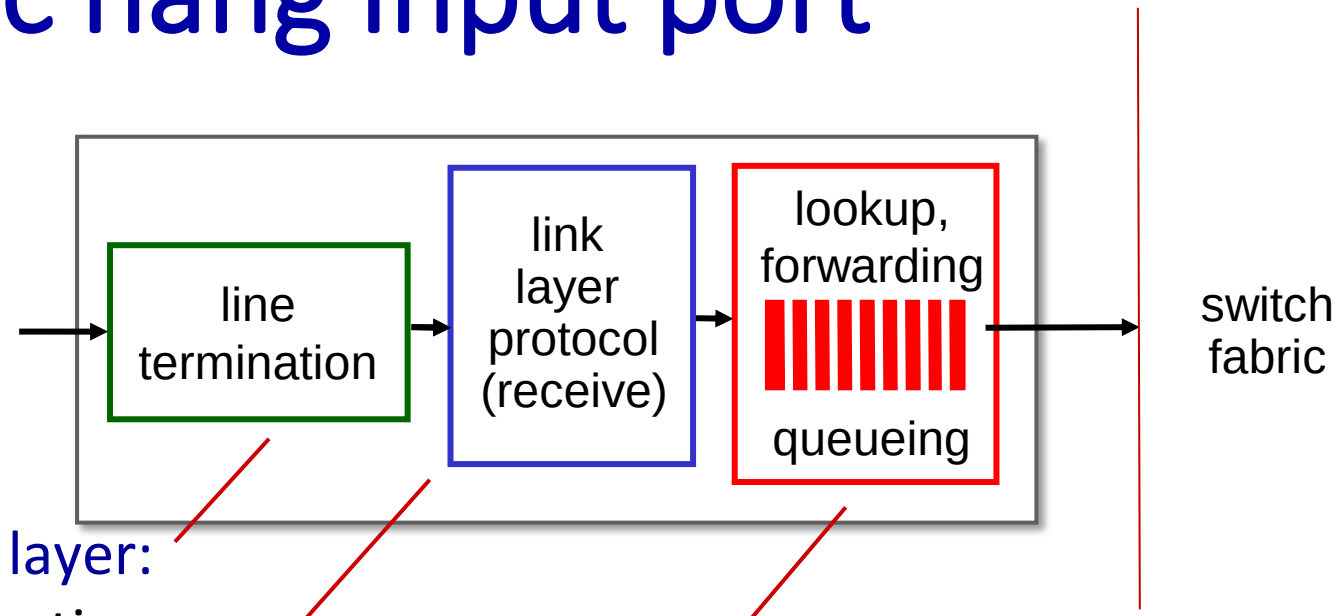
physical layer:
bit-level reception

link layer:
e.g., Ethernet
(chapter 6)

Chuyển mạch phi tập trung:

- Dùng giá trị trong các trường của header, tìm output port dùng forwarding table trong bộ nhớ của cổng input port *"match plus action"*
- Mục tiêu: hoàn thành xử lý ở tại 'line speed'
- input port queuing: nếu datagrams đến nhanh hơn tốc độ forwarding vào switch fabric

Chức năng Input port



physical layer:
bit-level reception

link layer:
e.g., Ethernet
(chapter 6)

Chuyển mạch phi tạp trung:

- Dùng giá trị trong các trường của header, tìm output port dùng forwarding table trong bộ nhớ của cổng input port *"match plus action"*
- destination-based forwarding: chuyển tới đích dựa vào IP đích (truyền thống)
- generalized forwarding: chuyển dựa vào bất kì tập giá trị của các trường trong header (thế hệ mới)

Destination-based forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010000 00000100	0
11001000 00010111 00010000 00000101 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: điều gì xẩy ra nếu các dải phân chia không “đẹp”?

Khớp số bit prefix dài nhất (Longest prefix matching)

longest prefix match

Khi tra cứu trong forwarding table cho địa chỉ đích, sử dụng so khớp số bít dài nhất của prefix.

Destination Address Range	Link interface
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

match!

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

match!

examples:

11001000 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
otherwise	3

match!

examples:

11001000 00010111 00010110 10100001 which interface?

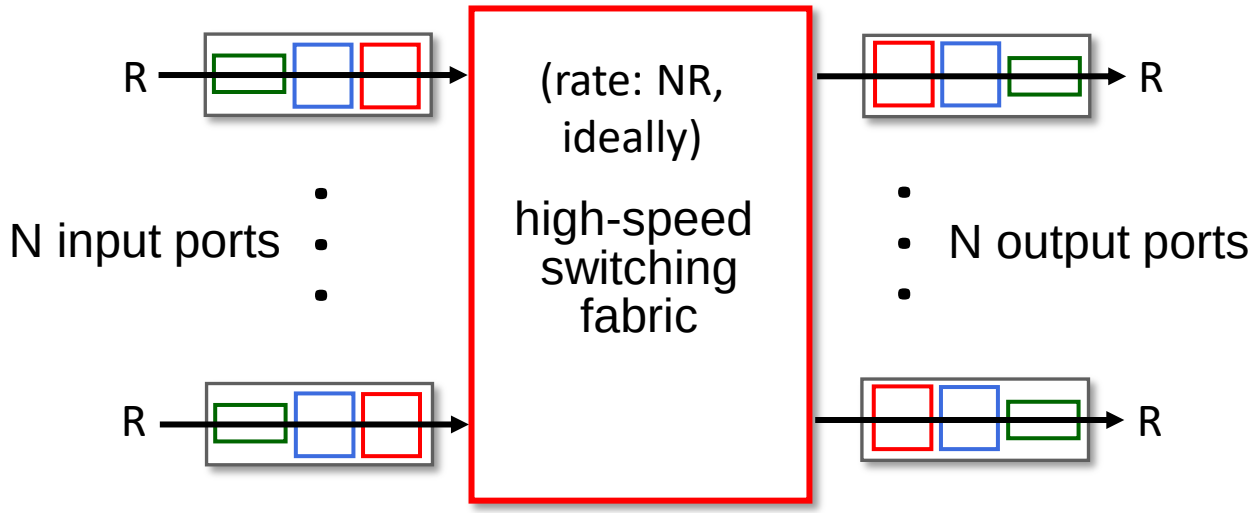
11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

- Chúng ta sẽ nghiên cứu chi tiết trong phần địa chỉ
- longest prefix matching: thường được thực thi sử dụng bộ nhớ TCAM (ternary content addressable memories) (TCAMs)
  - *content addressable: đặt địa chỉ trong TCAM: truy xuất địa chỉ trong 1 chu kì đồng hồ bất kể kích thước của bảng*
  - Cisco Catalyst: ~1tỉ entry tuyến đường (routing table) trong TCAM

Switching fabrics

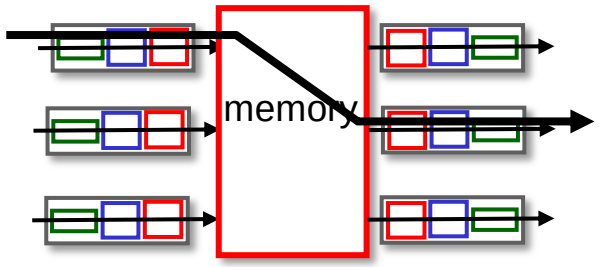
- Truyền gói từ input link đến output link tương ứng
- switching rate: tốc độ khi gói được truyền từ đầu vào tới đầu ra
- Thường được đo bằng bội số tốc độ line đầu vào/ra
  - N đầu vào: switching rate N lần tốc độ line rate



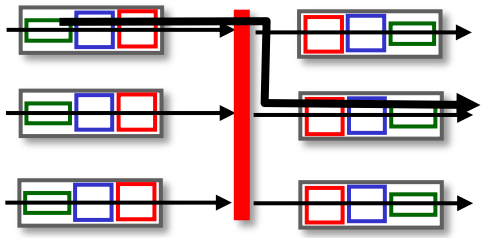
Switching fabrics

- Truyền gói từ input link đến output link tương ứng
- switching rate: tốc độ khi gói được truyền từ đầu vào tới đầu ra
  - Thường được đo bằng bội số tốc độ line đầu vào/ra
  - N đầu vào: switching rate N lần tốc độ line rate

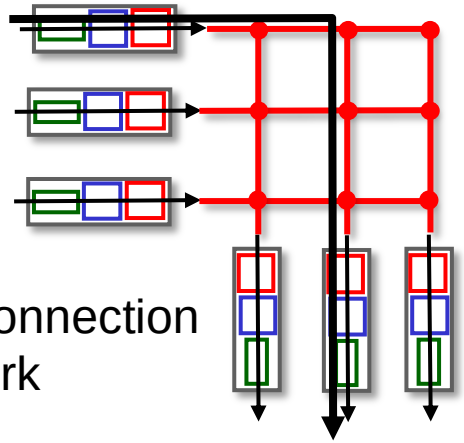
- Ba kiểu switching fabrics:



memory



bus

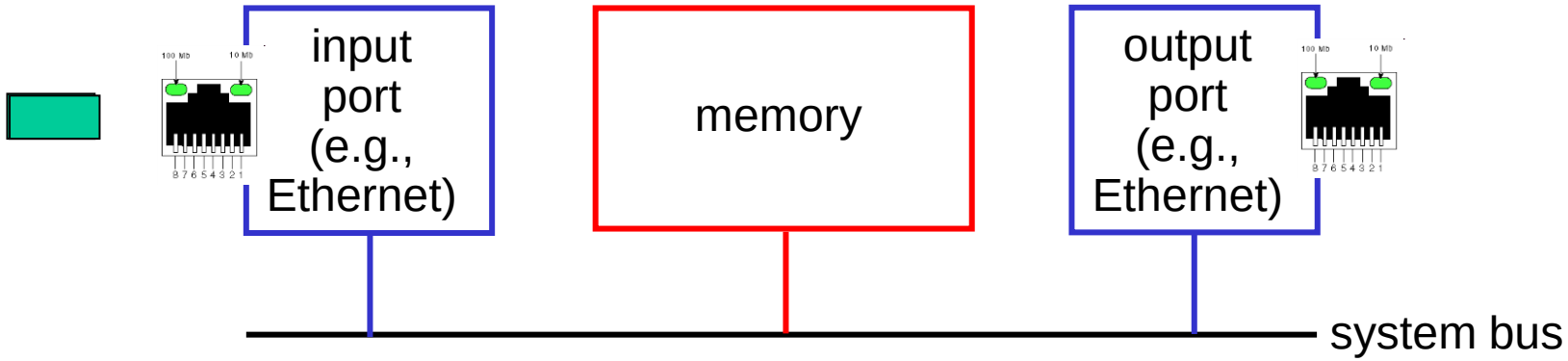


interconnection network

Switching theo kiểu memory

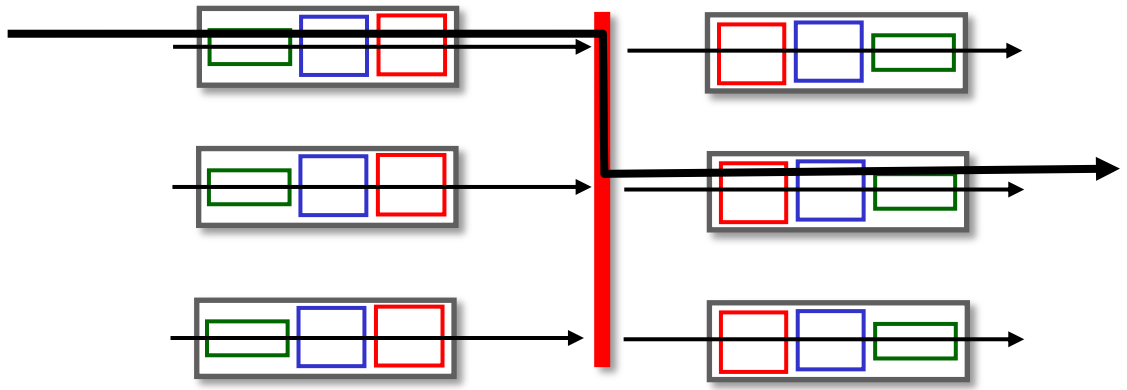
Router thế hệ đầu tiên:

- Máy tính truyền thống với chuyển mạch được điều khiển trực tiếp của CPU
- Gói được copy tới bộ nhớ hệ thống
- Tốc độ bị giới hạn bởi memory bandwidth (2 bus crossings per datagram)



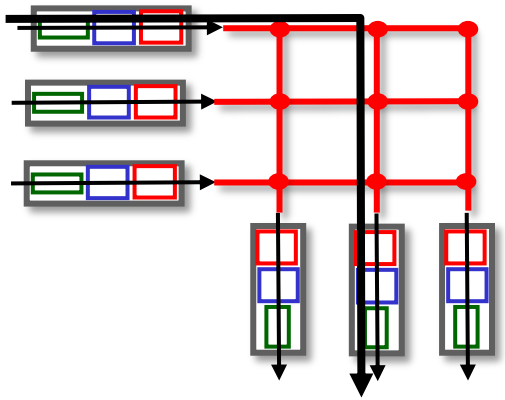
Switching theo kiểu bus

- datagram từ bộ nhớ của input port tới bộ nhớ của output port memory theo đường bus chia sẻ
- *bus contention:* tốc độ chuyển mạch bị giới hạn bởi bus bandwidth
- 32 Gbps bus, Cisco 5600: tốc độ thỏa mãn cho router truy cập



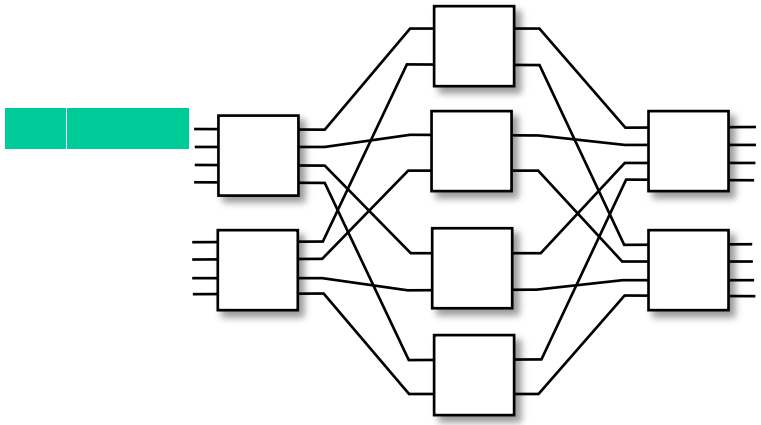
Switching theo kiểu interconnection network

- Crossbar, Clos networks, other interconnection nets để kết nối các bộ vi xử lý trong nhiều bộ vi xử lý



3x3 crossbar

- multistage switch: *nxn* switch từ nhiều trạng thái của smaller switches
- Xử lý song song:
  - Chia nhỏ datagram thành các cell có kích thước cố định
  - Chuyển mạch các cell vào fabric, đóng gói lại datagram tại đầu ra

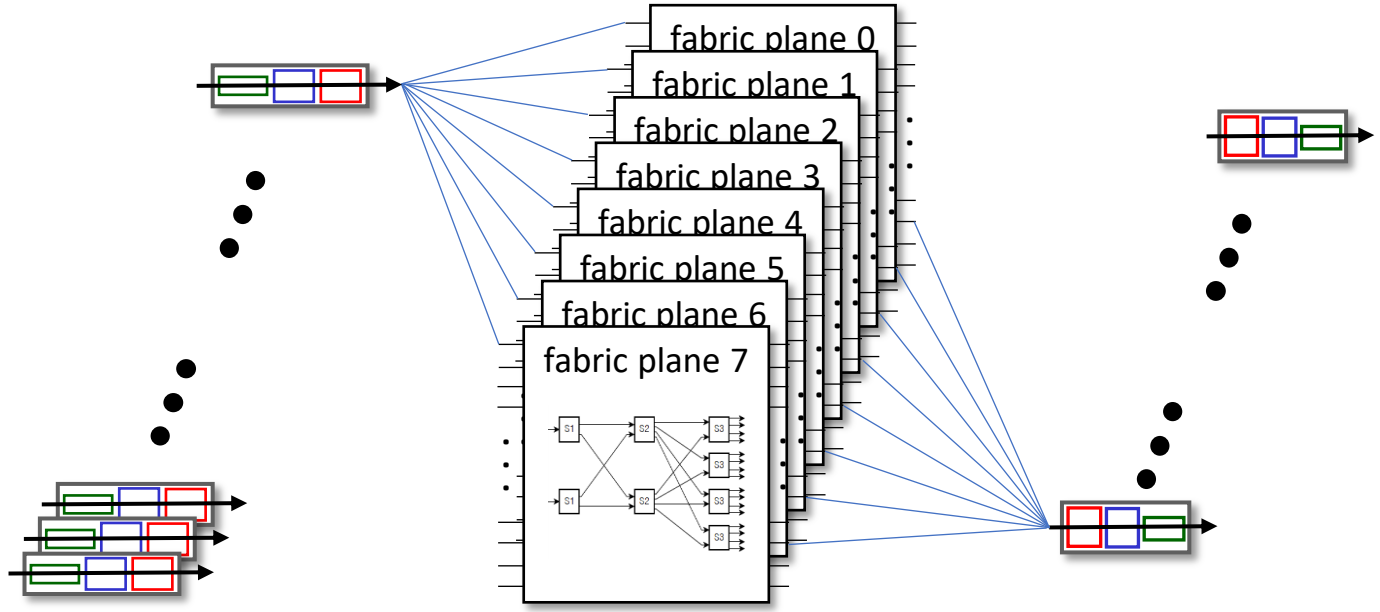


8x8 multistage switch
built from smaller-sized switches

Switching theo kiểu interconnection network

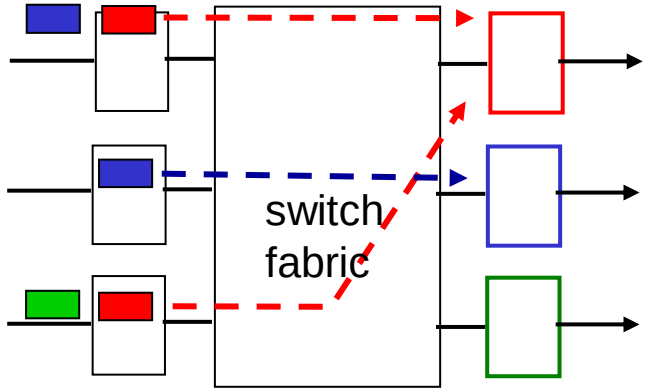
- scaling, sử dụng nhiều switching “planes” song song:
  - speedup, scaleup theo kiểu song song

- Cisco CRS router:
  - basic unit: 8 switching planes
  - Mỗi plane: 3-stage interconnection network
  - Lên tới 100’s Tbps switching capacity

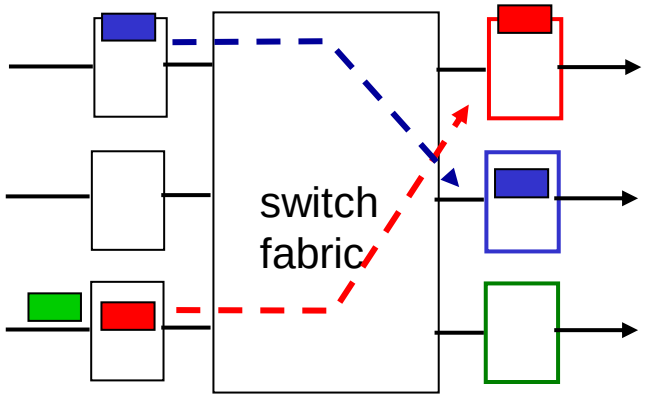


Input port queuing

- Nếu switch fabric chậm hơn các input ports -> queueing có thể xẩy ra tại đầu vào = input queues
  - queueing delay and loss phụ thuộc vào bộ đệm đầu vào!
- Head-of-the-Line (HOL) blocking: datagram trong hàng đợi phía trước chặn các datagram khác trong hàng đợi di chuyển phía trước

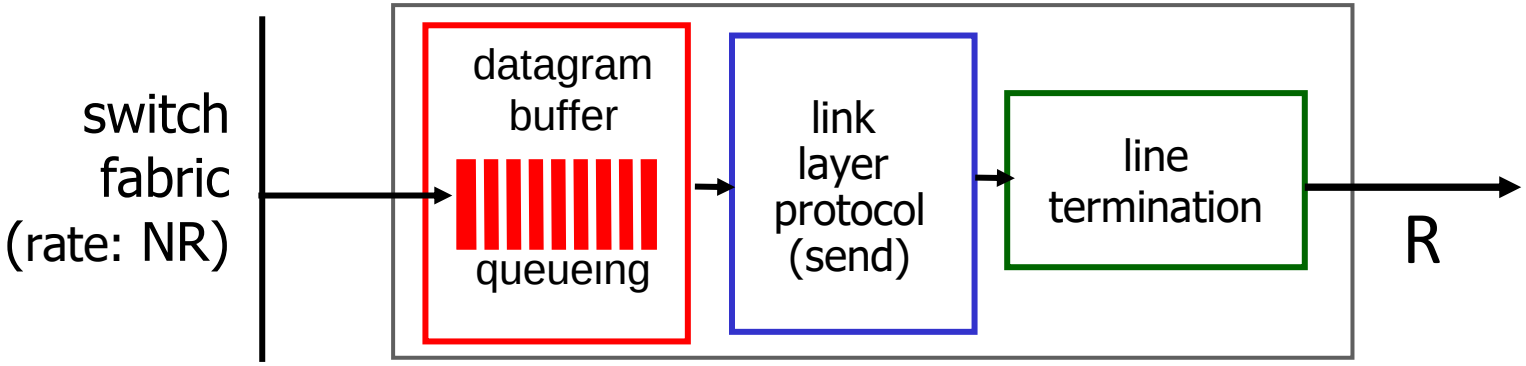


Tranh chấp ở đầu ra: chỉ có một datagram màu đỏ được truyền, datagram màu đỏ còn lại bị khóa



Một thời gian sau: gói xanh lá cây sẽ trải qua HOL

Output port queuing



This is a really important slide

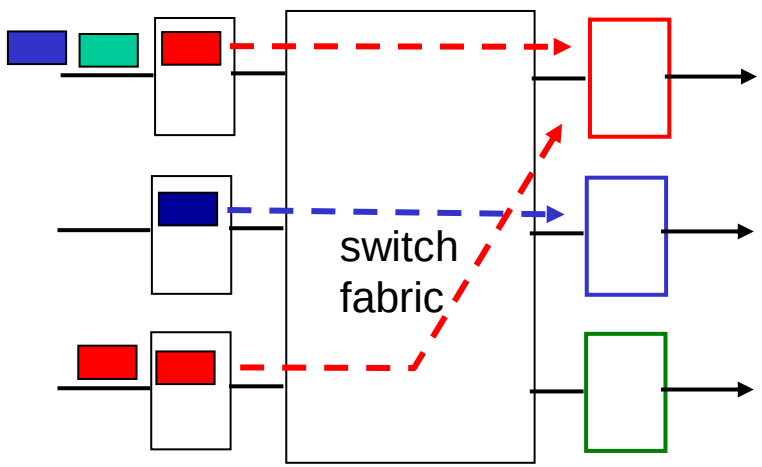
- *Buffering* được dùng khi datagrams đến từ fabric nhanh hơn tốc độ truyền của link. *Chính sách drop:* datagrams to drop if no free buffers?

→ Datagrams có thể mất nếu tắc nghẽn, thiếu dung lượng bộ đệm

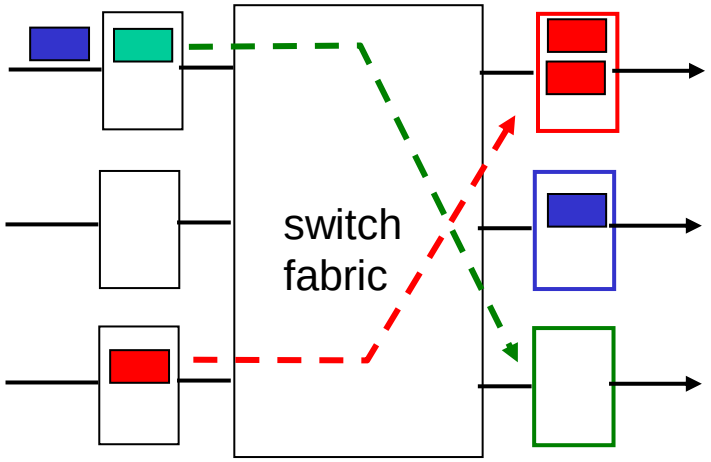
- *Chính sách lập lịch Scheduling* chọn datagram trong hàng đợi để truyền

→ Lập lịch ưu tiên– ai có thể phải ưu tiên nhất

Output port queuing



at *t*, packets more from input to output



one packet time later

- buffering khi tốc độ đến theo switch vượt quá tốc độ link đầu ra
- *queueing (delay) and loss phụ thuộc vào bộ đệm cổng ra!*

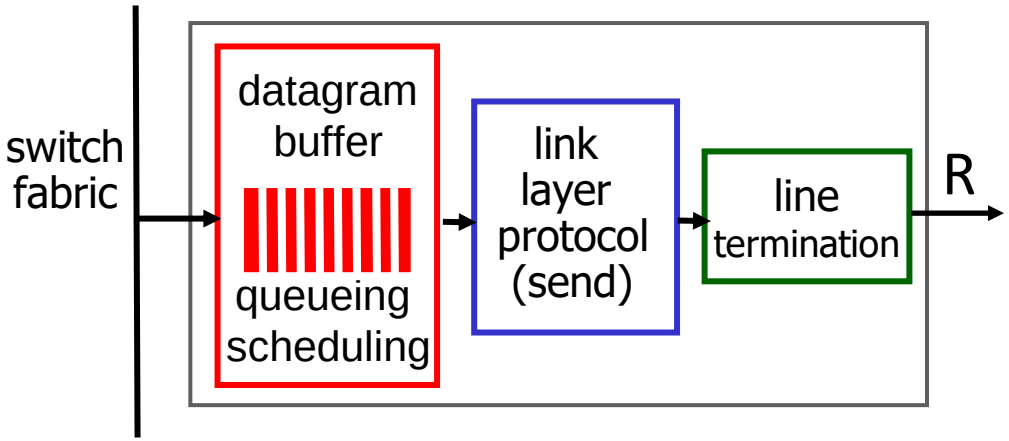
Bộ đệm bao nhiêu?

- RFC 3439 đưa ra luật: bộ đệm trung bình = RTT đặc biệt (250 msec) nhân với khả năng của link C
  - e.g., C = 10 Gbps link suy ra bộ đệm =0.25*10= 2.5 Gbit buffer
- Đề xuất: với N luồng, bộ đệm

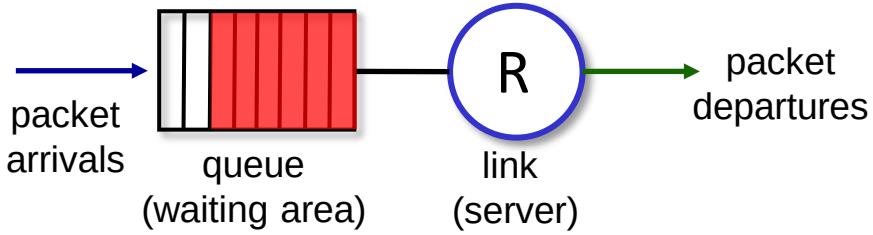
$$\frac{RTT \cdot C}{\sqrt{N}}$$

- Nhưng quá nhiều bộ đệm có thể tăng độ trễ
  - long RTTs: hiệu năng nghèo nàn cho các ứng dụng thời gian thực, phản hồi TCP chậm chạp
  - Nhớ lại kiểm soát tắc nghẽn dựa vào độ trễ: "giữ link đủ bận rộn nhưng không quá đầy"

Quản lý bộ đệm



Abstraction: queue



buffer management:

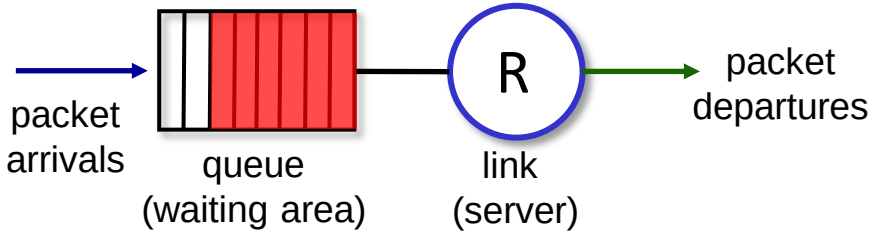
- drop: gói nào thêm, vứt khi bộ đệm đầy
  - Vứt đuôi: vứt các gói đang đến
  - Ưu tiên: vứt/gỡ bỏ dựa vào mức độ ưu tiên.
- Đánh dấu: gói được đánh dấu thông báo có tắc nghẽn (ECN, RED)

Packet Scheduling: FCFS

packet scheduling: quyết định gói nào được gửi tiếp theo trên link

- first come, first served
- priority
- round robin
- Trọng số công bằng

Abstraction: queue



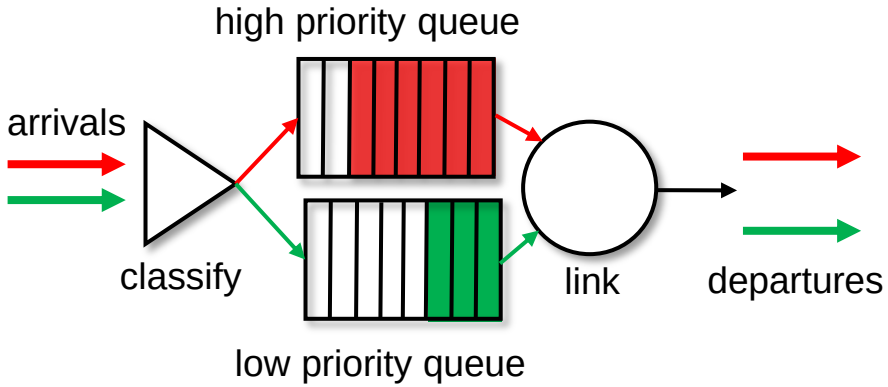
FCFS: gói truyền theo thứ tự đến output port

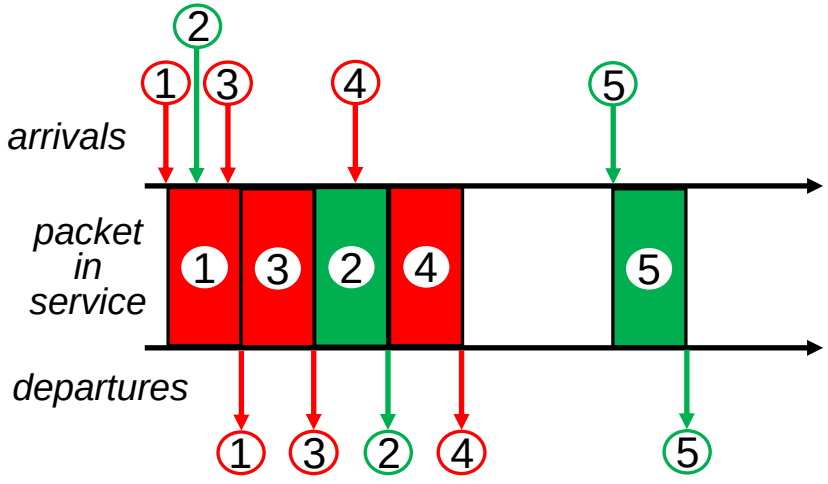
- Giống như: First-in-first-out (FIFO)
- Lấy ví dụ trong cuộc sống?

Scheduling policies: ưu tiên

Priority scheduling:

- Lưu lượng đến được phân lớp, đợi theo lớp
  - Bất cứ trường nào trong header dùng để phân lớp
- Gửi gói trong hàng đợi có độ ưu tiên cao hơn.
  - FCFS trong phạm vi các gói có độ ưu tiên

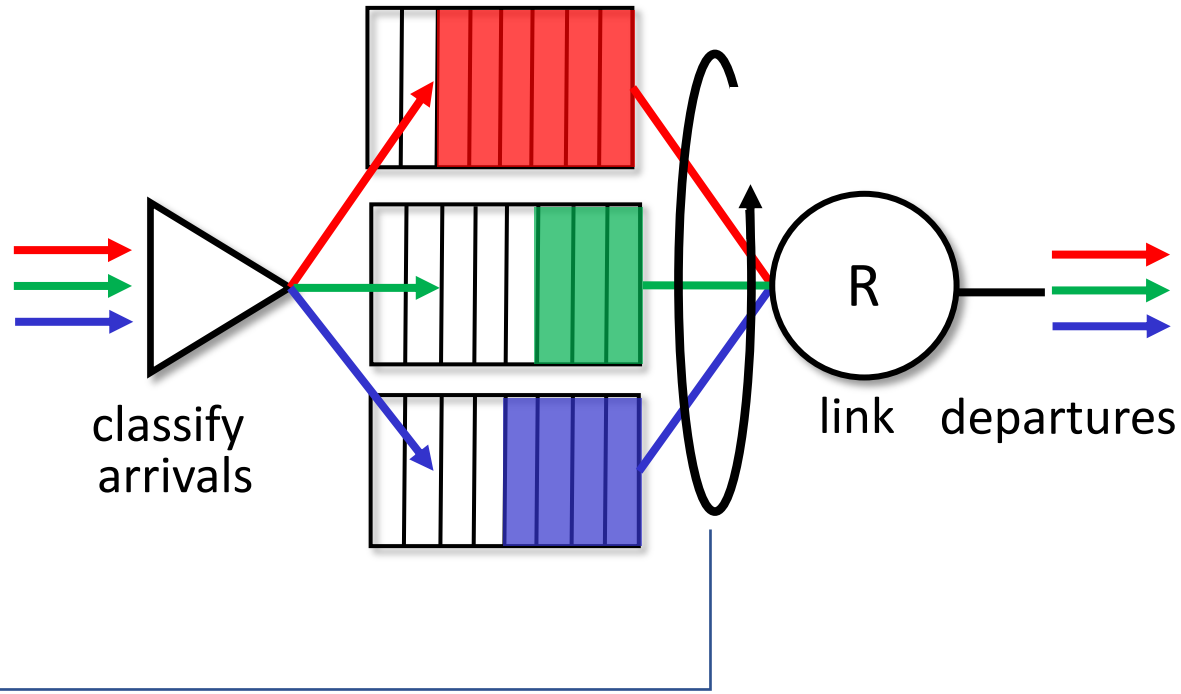




Scheduling policies: round robin

Round Robin (RR) scheduling:

- Lưu lượng đến được phân lớp, đợi theo lớp
  - Bất cứ trường nào trong header dùng để phân lớp
- Phục vụ theo chu kì, lặp đi lặp lại quét lớp hàng đợi gửi một gói của lớp đi sau đó lặp lại



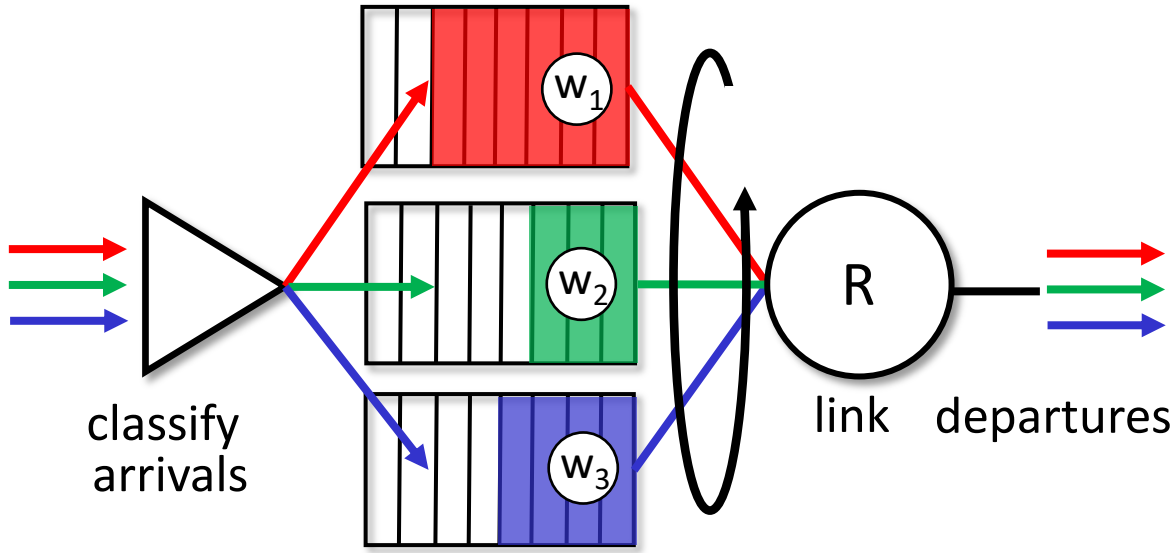
Scheduling policies: trọng số (*Weighted Fair Queuing)*

Weighted Fair Queuing (WFQ):

- Tạo ra Round Robin
- Mỗi lớp, *i*, có trọng số w_i, và tổng trọng số mỗi chu kì:

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

- Tối thiểu băng thông được đảm bảo(per-traffic-class)



Sidebar: Mạng trung lập (Network Neutrality)

Cái gì là Network Neutrality

- *technical:* ISP nên chia sẻ/ định vị tài nguyên như thế nào?
  - packet scheduling, buffer management are the *mechanisms*
- *Nguyên lý kinh tế, xã hội*
  - Bảo vệ quyền tự do
  - Khuyến khích cải tiến và cạnh tranh
- Thực hiện luật và chính sách

Các quốc gia khác nha có network neutrality khác nhau

Sidebar: Network Neutrality

2015 US FCC *Order on Protecting and Promoting an Open Internet:* three "clear, bright line" rules:

- no blocking … "shall not block lawful content, applications, services, or non-harmful devices, subject to reasonable network management."
- no throttling … "shall not impair or degrade lawful Internet traffic on the basis of Internet content, application, or service, or use of a non-harmful device, subject to reasonable network management."
- no paid prioritization. … "shall not engage in paid prioritization"

Các quốc gia đang đưa ra các chính sách về mạng đặc biệt mạng xã hội?

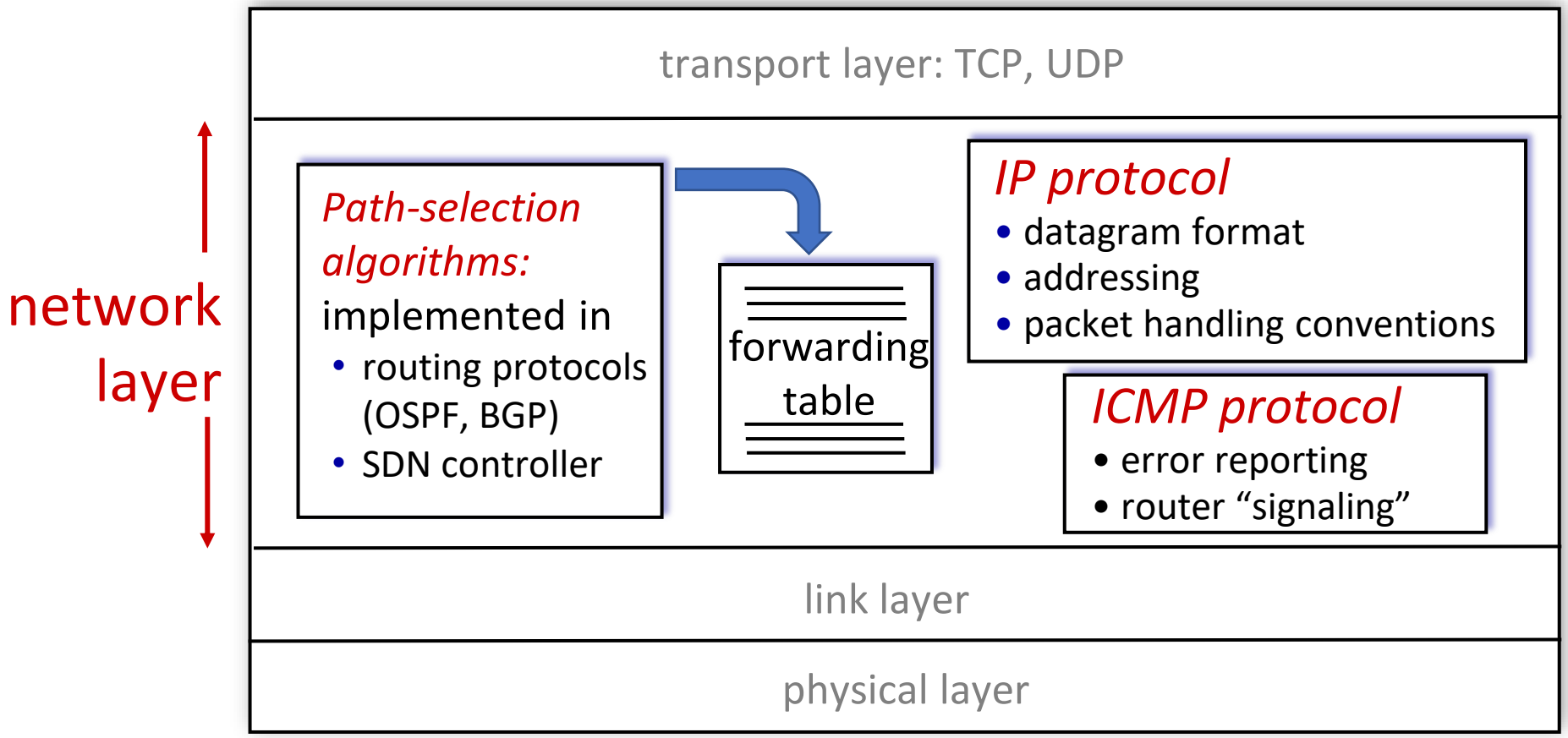
Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

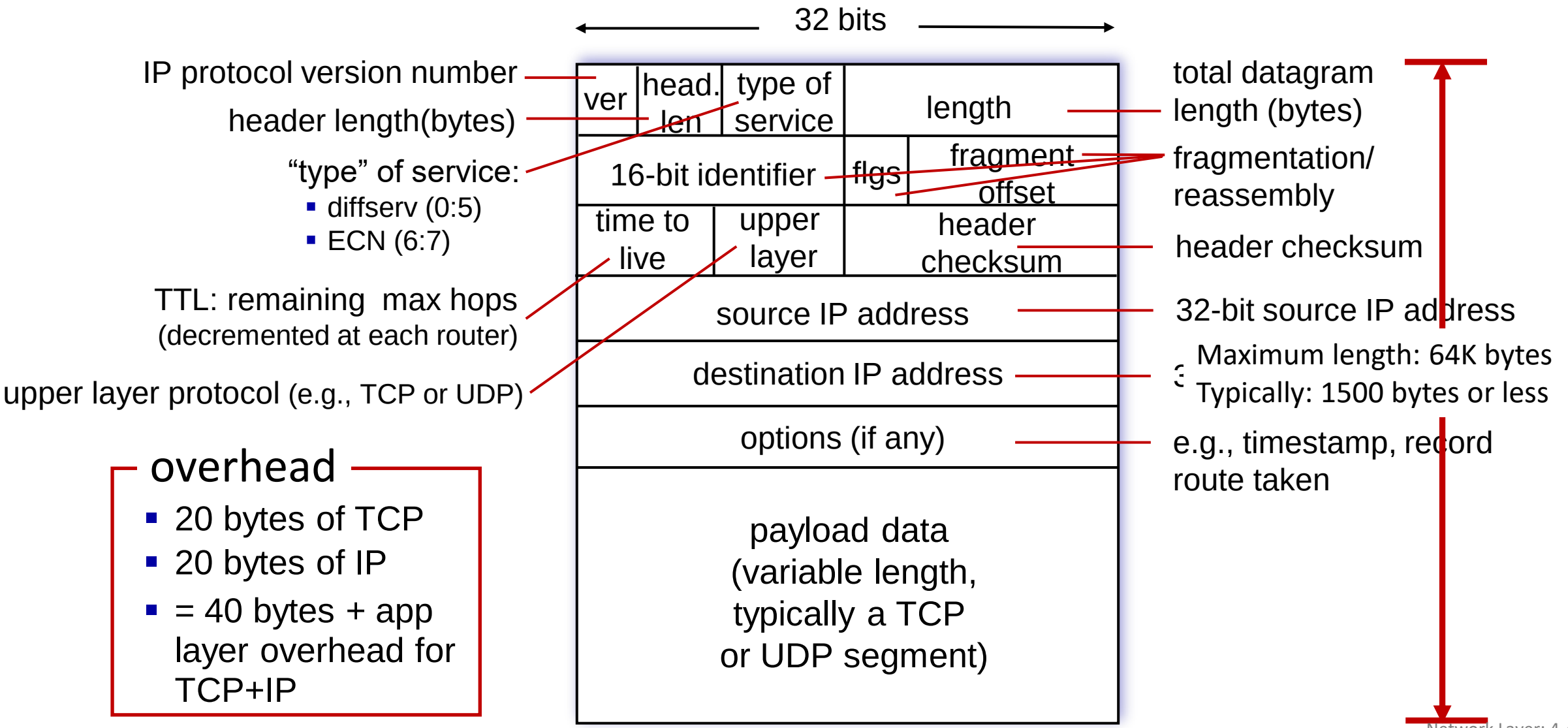
- Generalized Forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes

Network Layer: Internet

Chức năng của host, router network layer:



Định dạng của IP Datagram

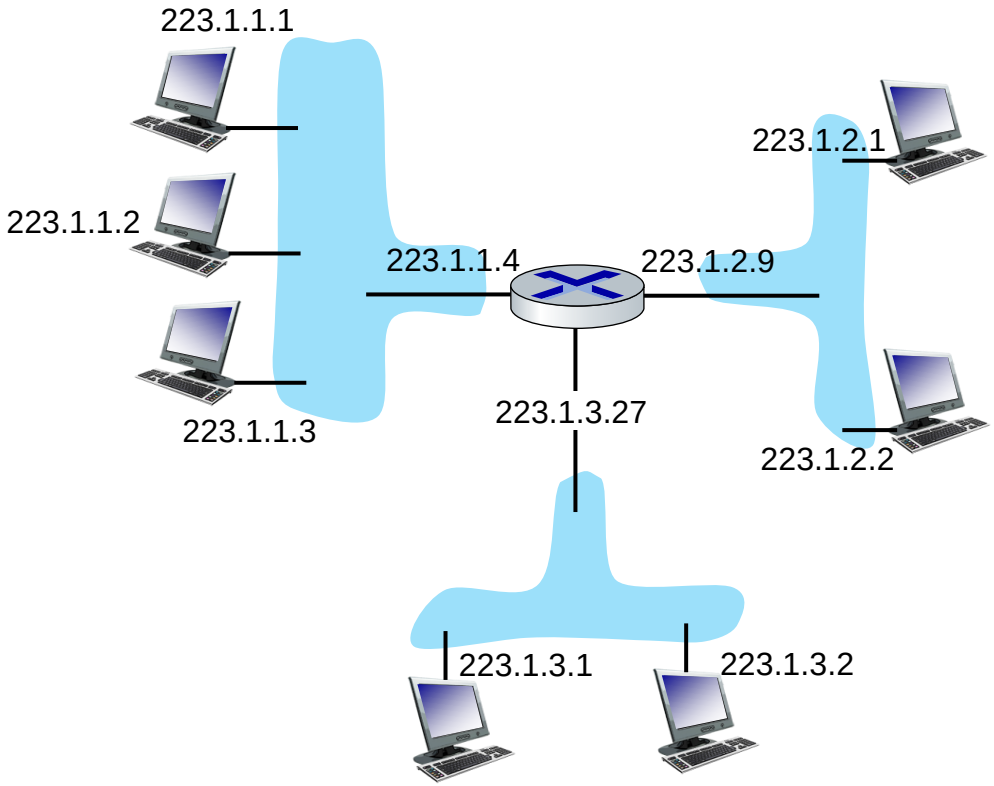


overhead

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead for TCP+IP

IP addressing (địa chỉ IP): giới thiệu

- IP address: 32-bit định danh host hoặc router *interface*
- interface: kết nối giữa host/router và physical link
  - Router có nhiều interface
  - host thường chỉ có 1 hoặc 2 interfaces (có dây, không dây) (e.g., wired Ethernet, wireless 802.11)



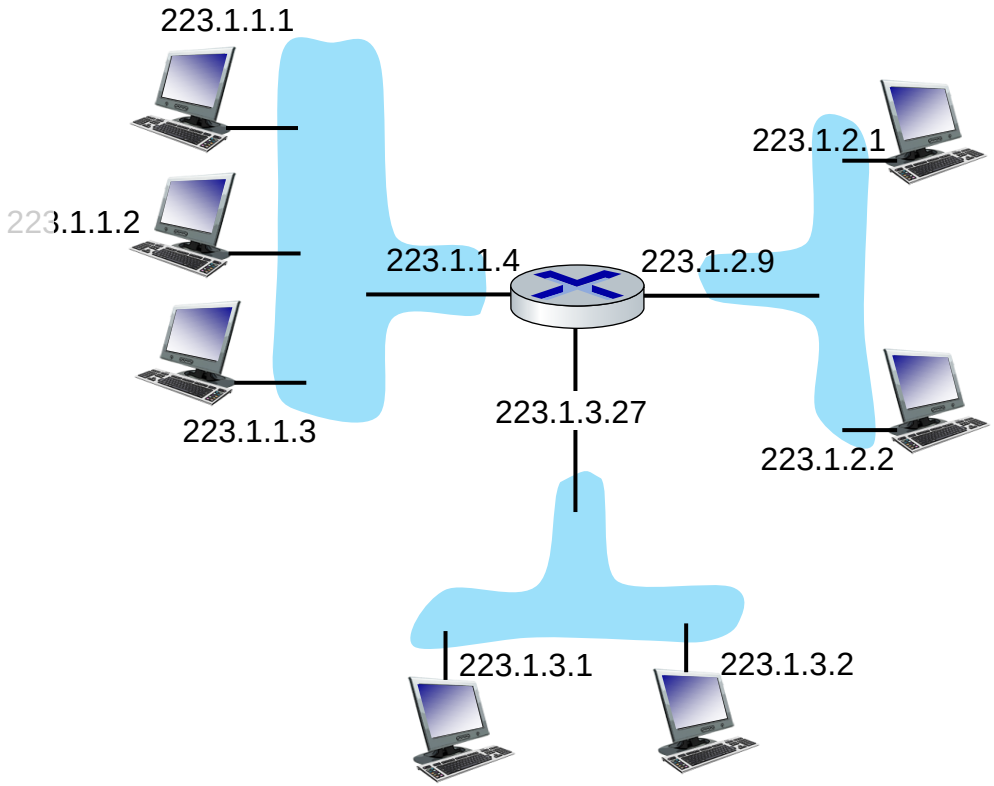
dotted-decimal IP address notation:

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

IP addressing: introduction

- IP address: 32-bit định danh host hoặc router *interface*
- interface: kết nối giữa host/router và physical link
  - Router có nhiều interface
  - host thường chỉ có 1 hoặc 2 interfaces (có dây, không dây) (e.g., wired Ethernet, wireless 802.11)



dotted-decimal IP address notation:

223.1.1.1 = 11011111 00000001 00000001 00000001

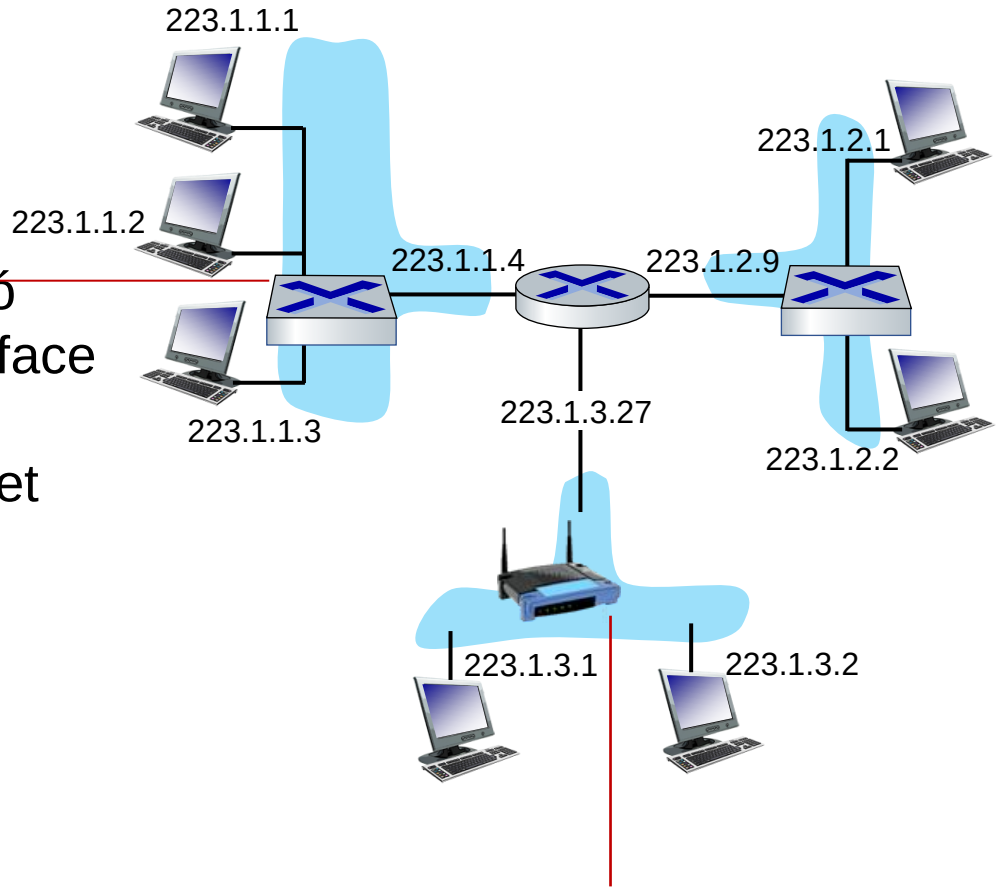
223 1 1 1

IP addressing: giới thiệu

Q: interface thực sự kết nối như thế nào?

A: Chương 6, 7

A: một kết nối có dây kết nối interface Ethernet với interface Ethernet trên switch



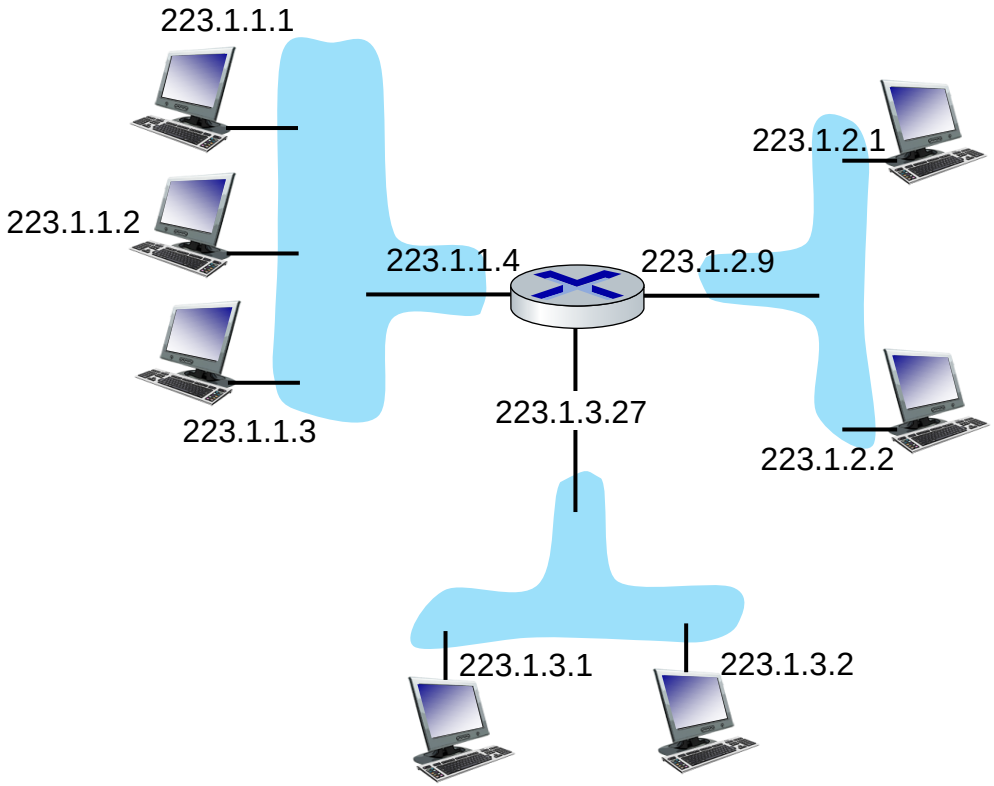
A: Interface không dây kết nối với trạm thu phát song Wifi

For now: don't need to worry about how one interface is connected to another (with no intervening router)

Subnets (mạng con)

- *Mạng con là cái gì*
  - Các interface có thể kết nối vật lý với nhau mà không cần có sự can thiệp của router?

- Địa chỉ IP có cấu trúc:
  - Phần mạng con: các thiết bị cùng subnet có số bit trọng số cao (bit phía bên trái) giống nhau
  - Phần trạm (phần host): phần bit còn lại

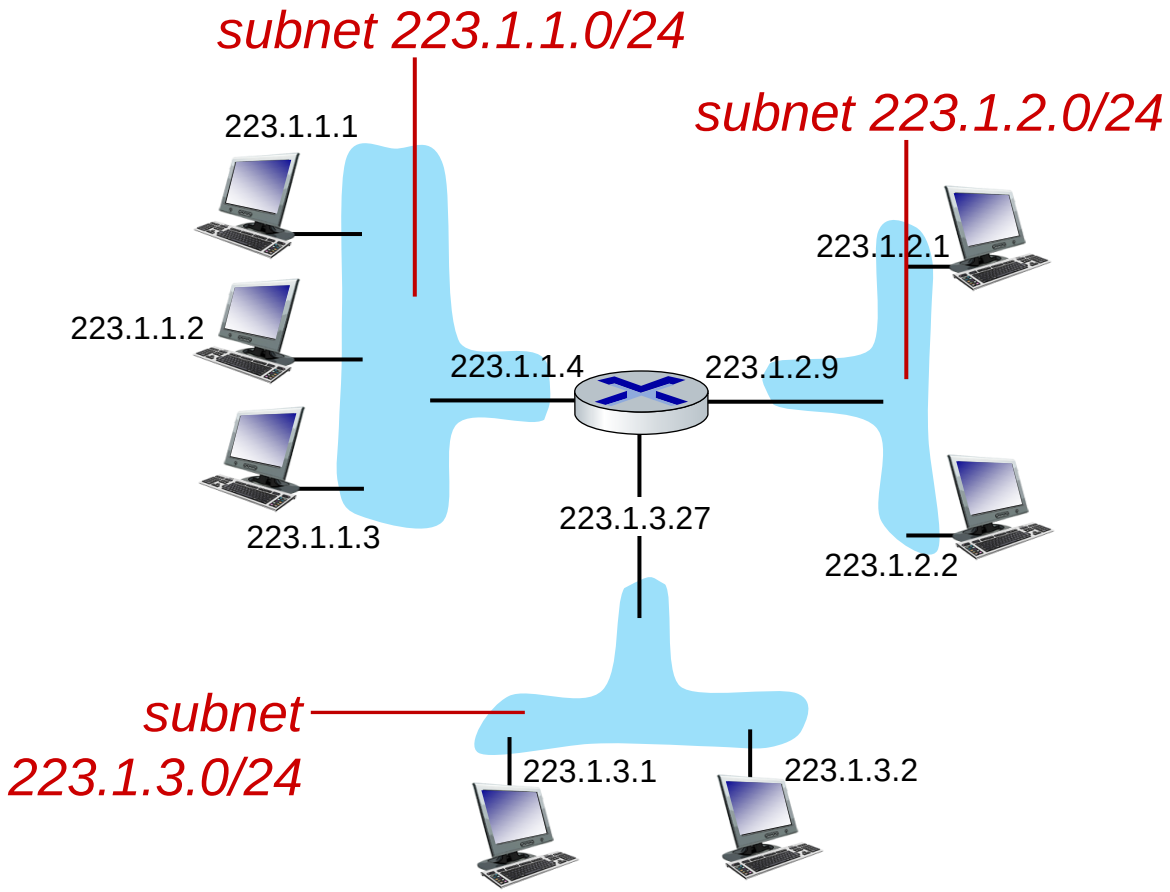


network consisting of 3 subnets

Subnets

Công thức xác định subnets:

- Mỗi interface của router là một hòn đảo độc lập – mạng độc lập
- Mỗi mạng độc lập được gọi là *subnet*

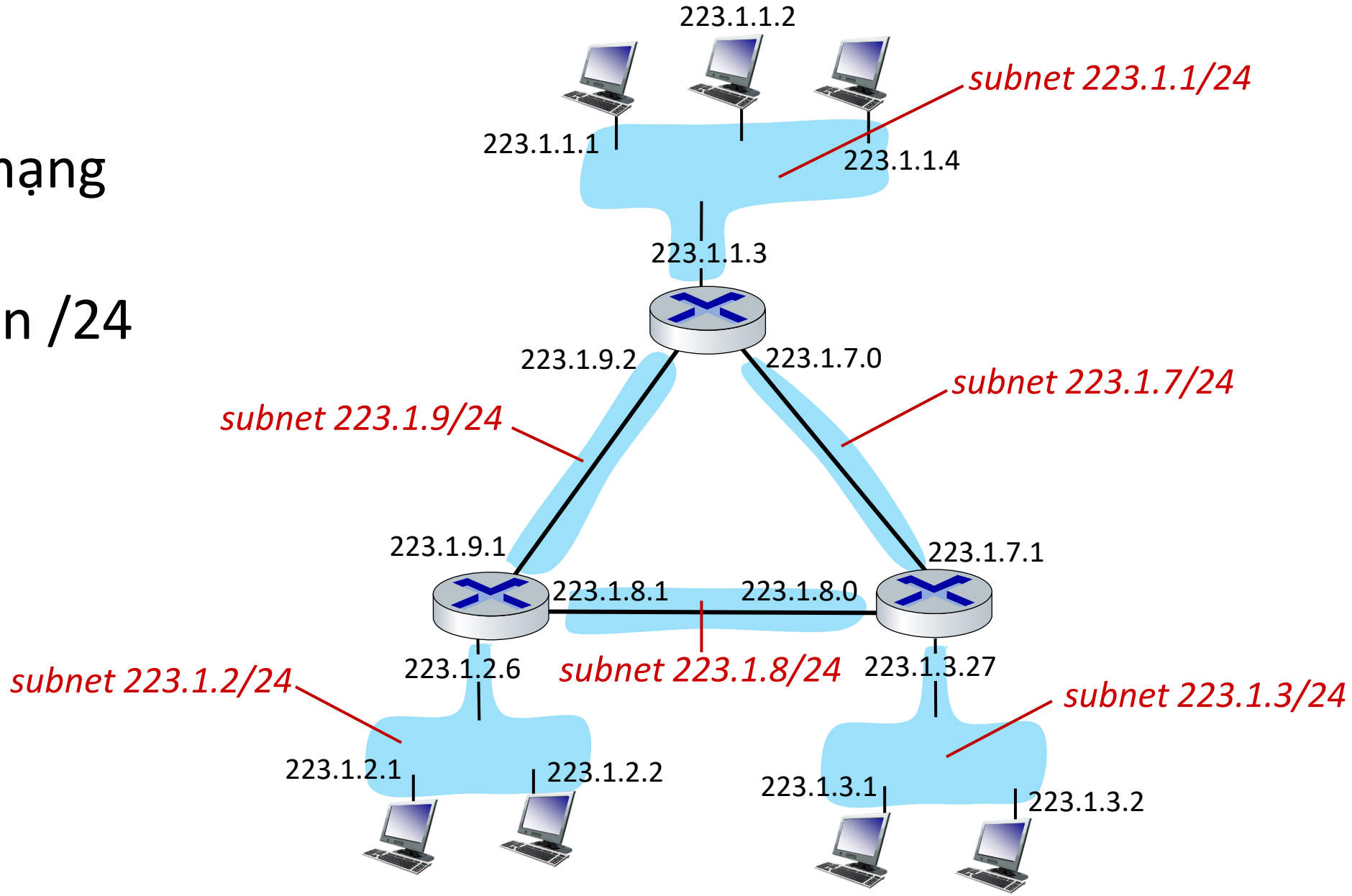


subnet mask: /24

(high-order 24 bits: subnet part of IP address)

Subnets

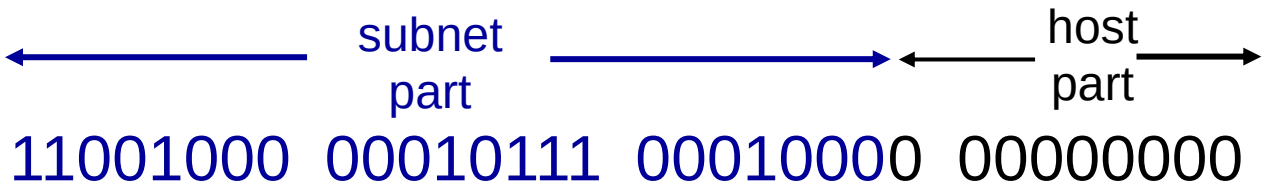
- Có Các mạng con?
- Mạng con /24 là gì



IP addressing: CIDR

CIDR: Classless InterDomain Routing (phát âm là “cider”)

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



200.23.16.0/23

IP addresses: lấy địa chỉ bằng cách nào

Có hai câu hỏi thực sự:

1. Q: 1 host lấy địa chỉ trong mạng của nó như thế nào? (phần host)
2. Q: một mạng lấy địa chỉ mạng cho nó bằng cách nào(network part)

Host lấy địa chỉ IP như thế nào?

- Được mã hóa cứng trong file config (e.g., /etc/rc.config in UNIX)
- DHCP: Dynamic Host Configuration Protocol: tự động lấy địa chỉ từ server
  - "plug-and-play"

DHCP: Dynamic Host Configuration Protocol

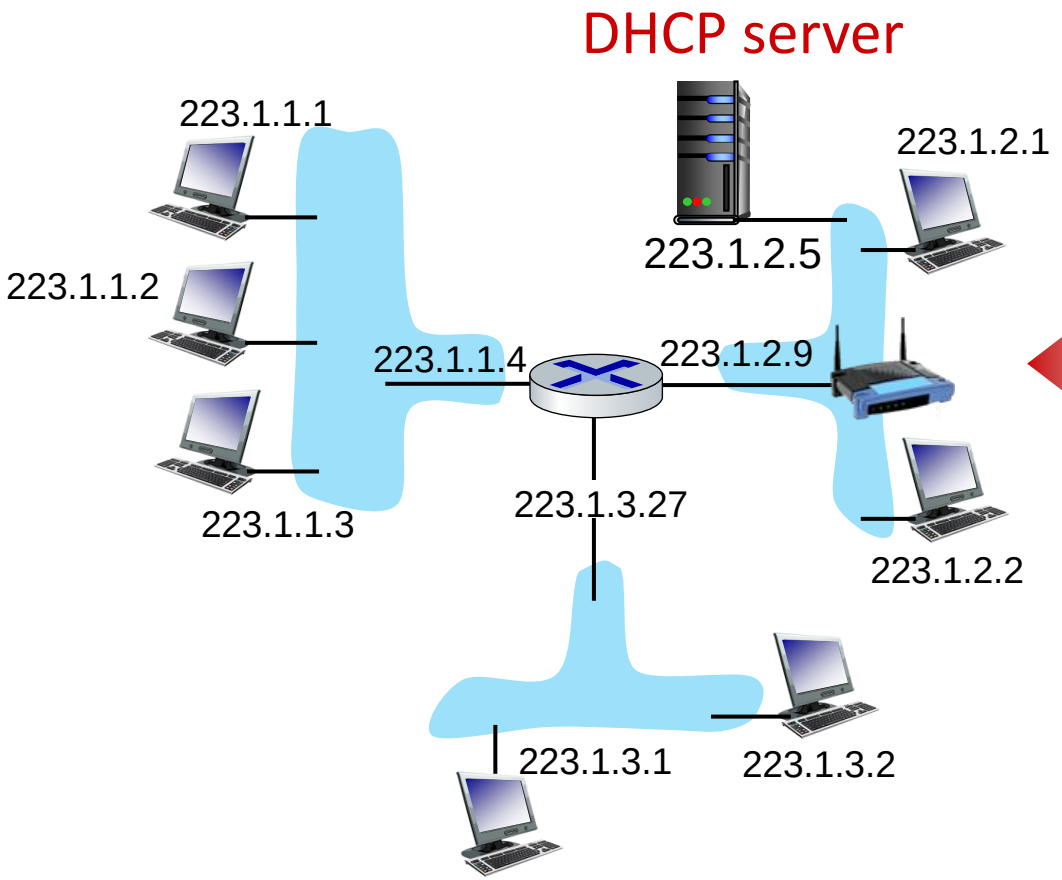
Mục tiêu: host tự động lấy địa chỉ IP từ máy chủ mạng khi nó tham gia vào mạng

- Có thể thuê mới địa chỉ trong quá trình dùng
- Cho phép sử dụng lại địa chỉ (chỉ giữ địa chỉ khi kết nối)
- Hỗ trợ người dùng di động trên các mạng khác nhau

DHCP tổng quan: quá trình DORA

- host broadcasts DHCP discover msg [optional]
- DHCP server responds with DHCP offer msg [optional]
- host requests IP address: DHCP request msg
- DHCP server sends address: DHCP ack msg

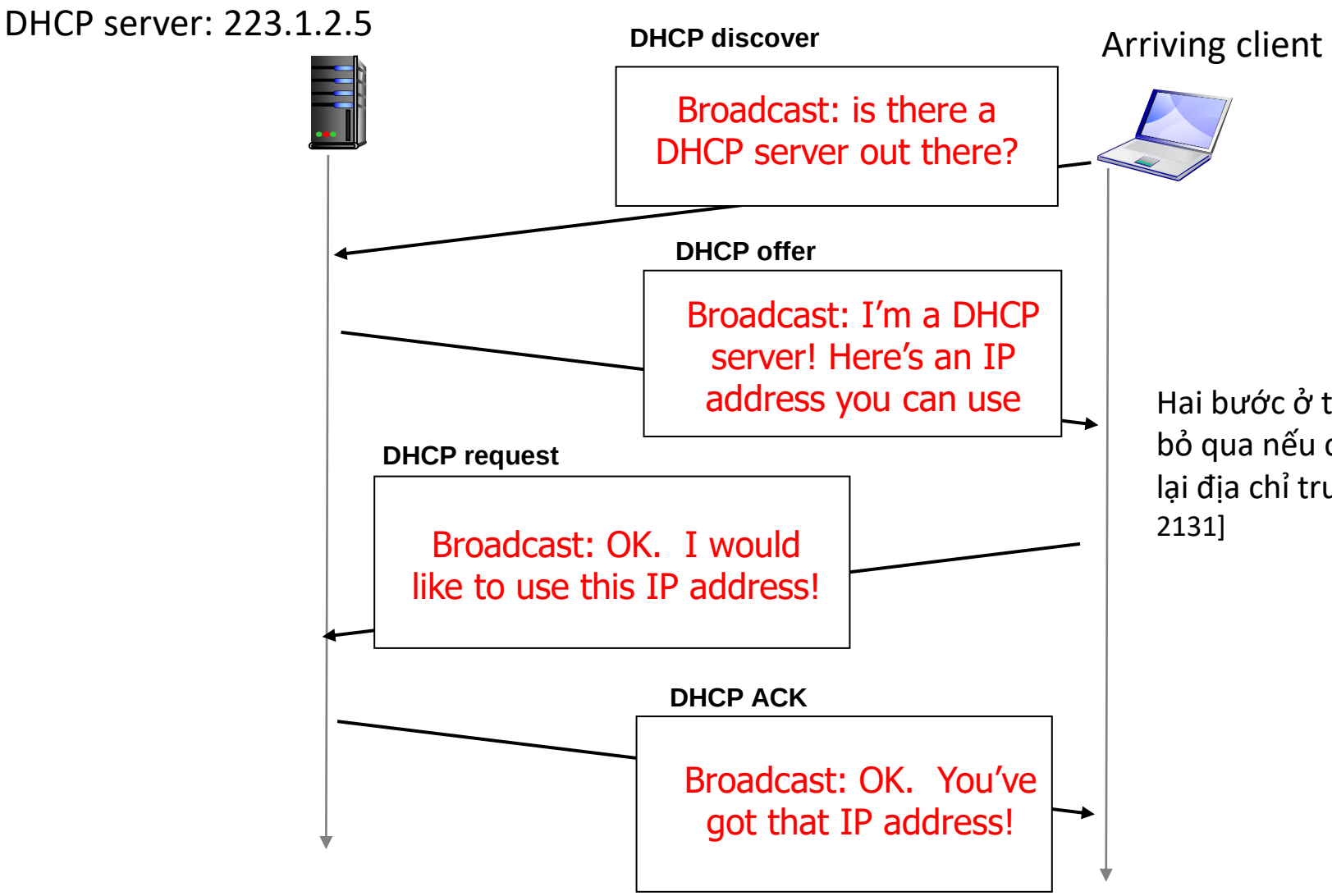
DHCP mô hình client-server



Thông thường, DHCP server sẽ kết hợp với router để cung cấp địa chỉ cho toàn bộ các mạng kết nối tới router (thậm chí router đóng vai trò là dhcp server)

arriving DHCP client needs address in this network

DHCP client-server scenario



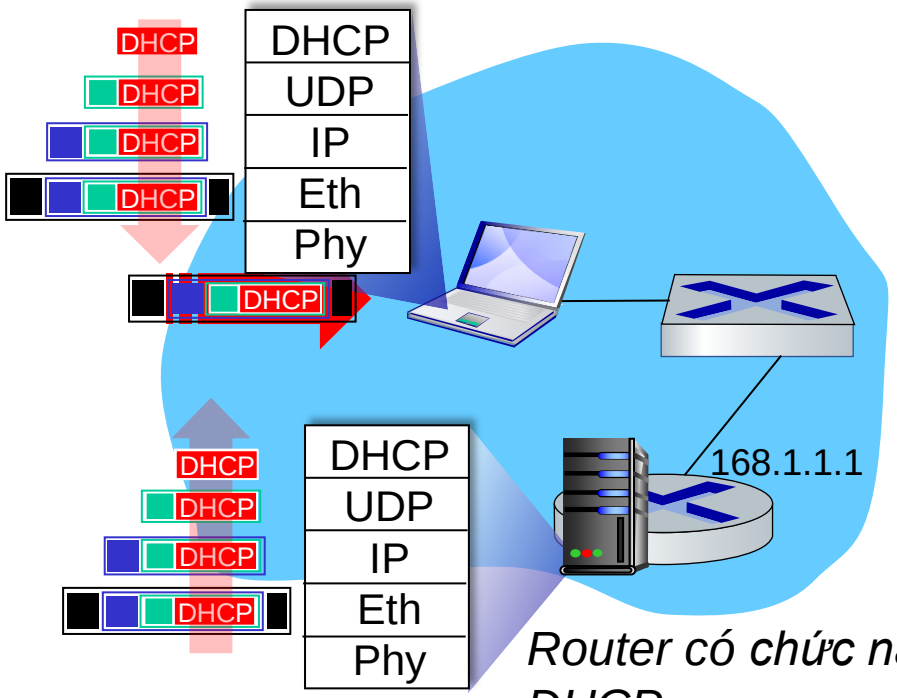
Hai bước ở trên có thể bị bỏ qua nếu client sử dụng lại địa chỉ trước đó [RFC 2131]

DHCP: cung cấp nhiều thông tin hơn bên cạnh địa chỉ

DHCP có thể cấp phát nhiều thông tin hơn bên cạnh địa chỉ:

- Địa chỉ gateway
- Tên và địa chỉ DNS server
- Mặt nạ mạng (xác định phần mạng trong địa chỉ)

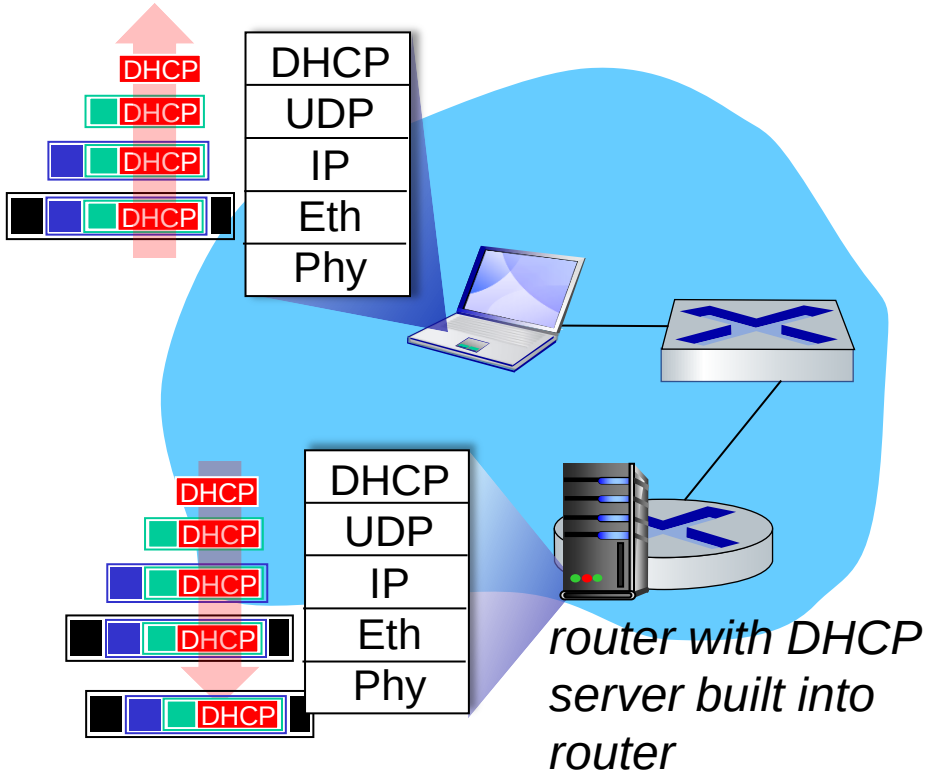
DHCP: example



Router có chức năng của DHCP

- Labtop kết nối sẽ dùng DHCP để lấy địa chỉ IP, gateway, mặt nạ mạng, địa chỉ của DNS server.
- Thông điệp DHCP REQUEST được đóng gói trong UDP, đóng gói trong IP, đóng gói trong Ethernet
- Frame quảng bá Ethernet (dest: FFFFFFFFFFFF) được nhận tại router
- Ethernet bóc tách thông tin đến DHCP

DHCP: example



- DCP server tạo thông điệp DHCP ACK chứa địa chỉ IP, gateway, mặt nạ, địa chỉ DNS server sẽ cấp cho client
- Gửi tới client, bóc tách thông tin
- Client bây giờ sẽ có IP, gateway, mặt nạ, địa chỉ DNS server

IP addresses: lấy địa chỉ bằng cách nào

Q: lấy địa chỉ mạng như thế nào?

A: lấy từ không gian địa chỉ của ISP

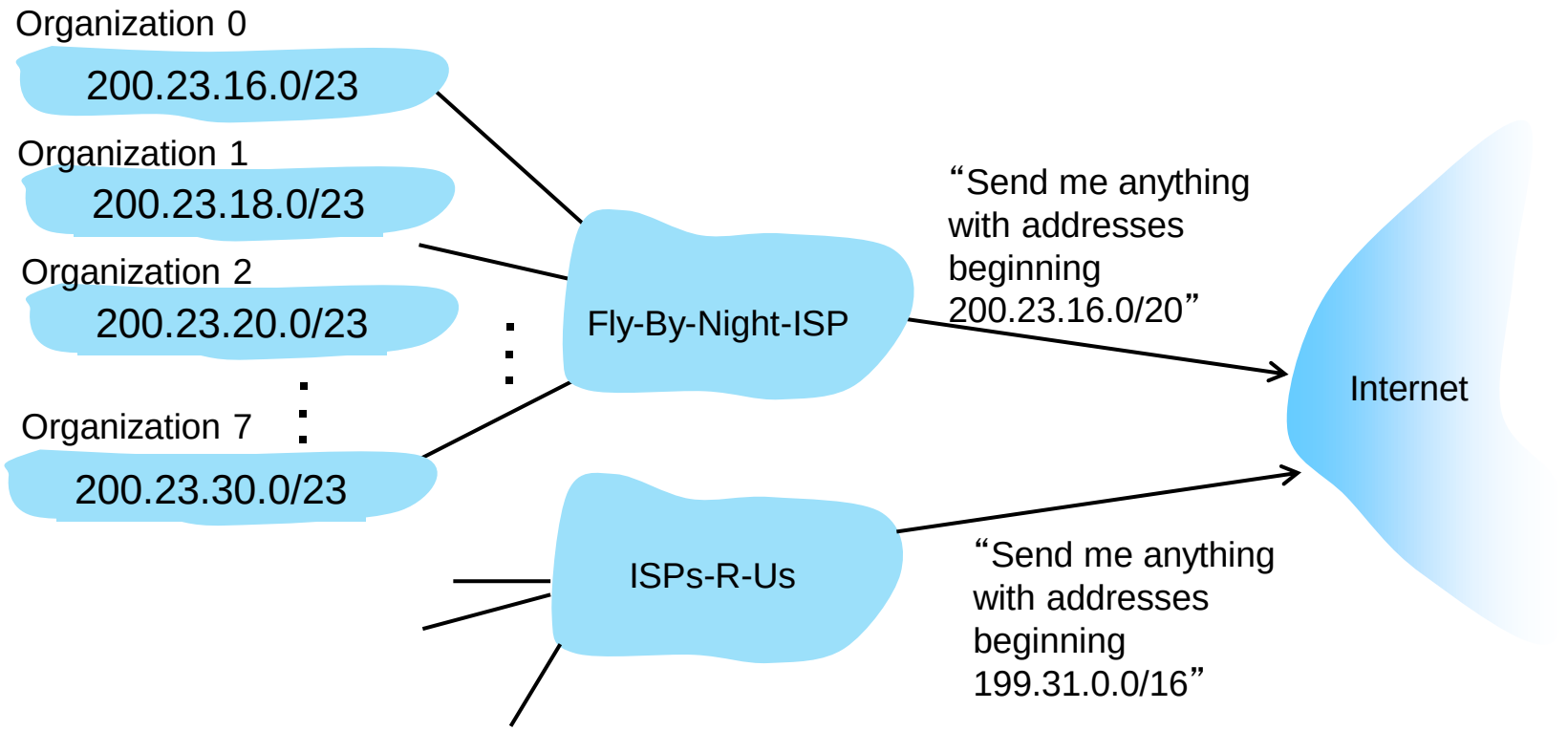
ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20

ISP có thể chia nhỏ khối thành 8 khối:

Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
…	……	….	….
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

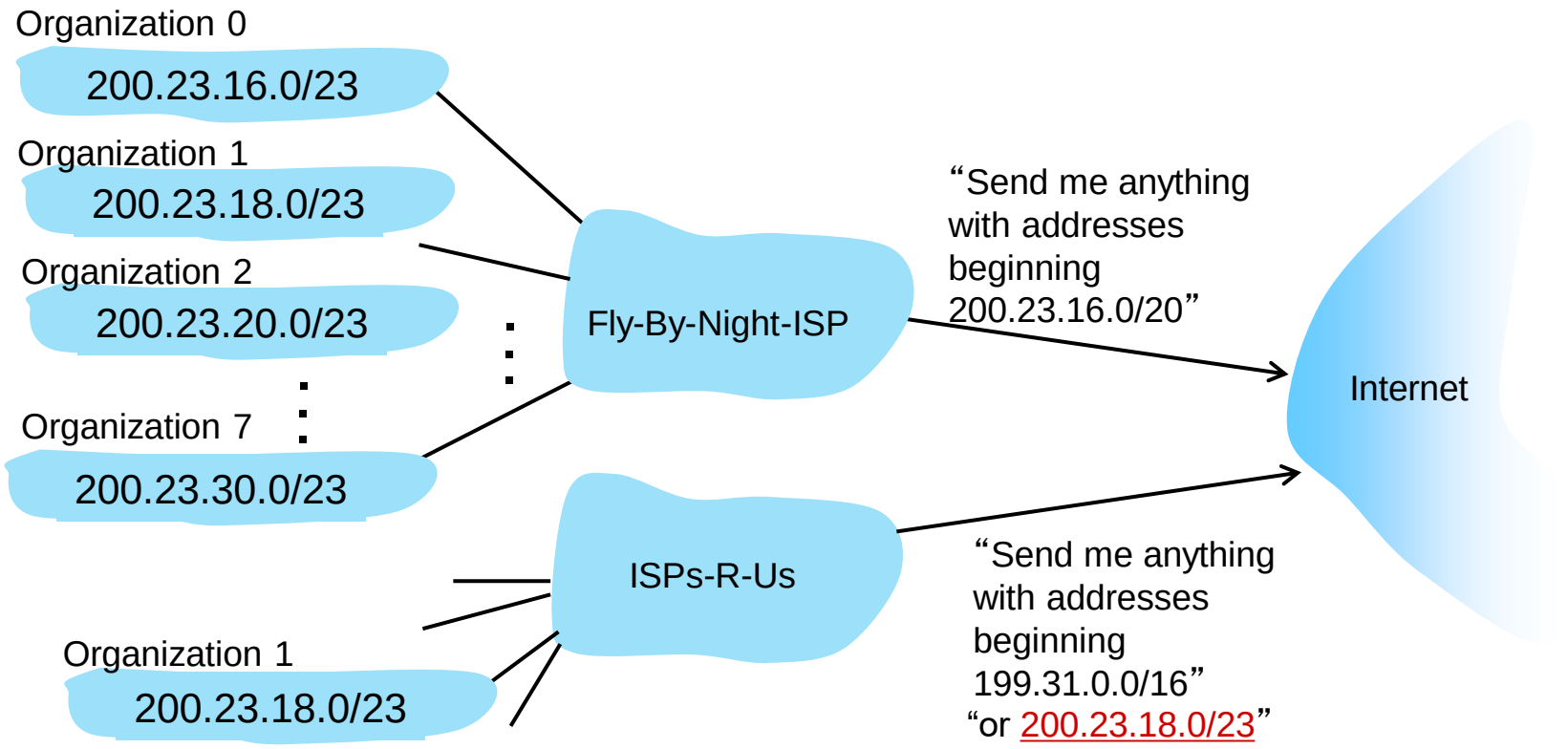
Phân cấp địa chỉ: tổng hợp tuyến đường

Phân cấp địa chỉ cho phép quảng bá thông tin định tuyến hiệu quả.



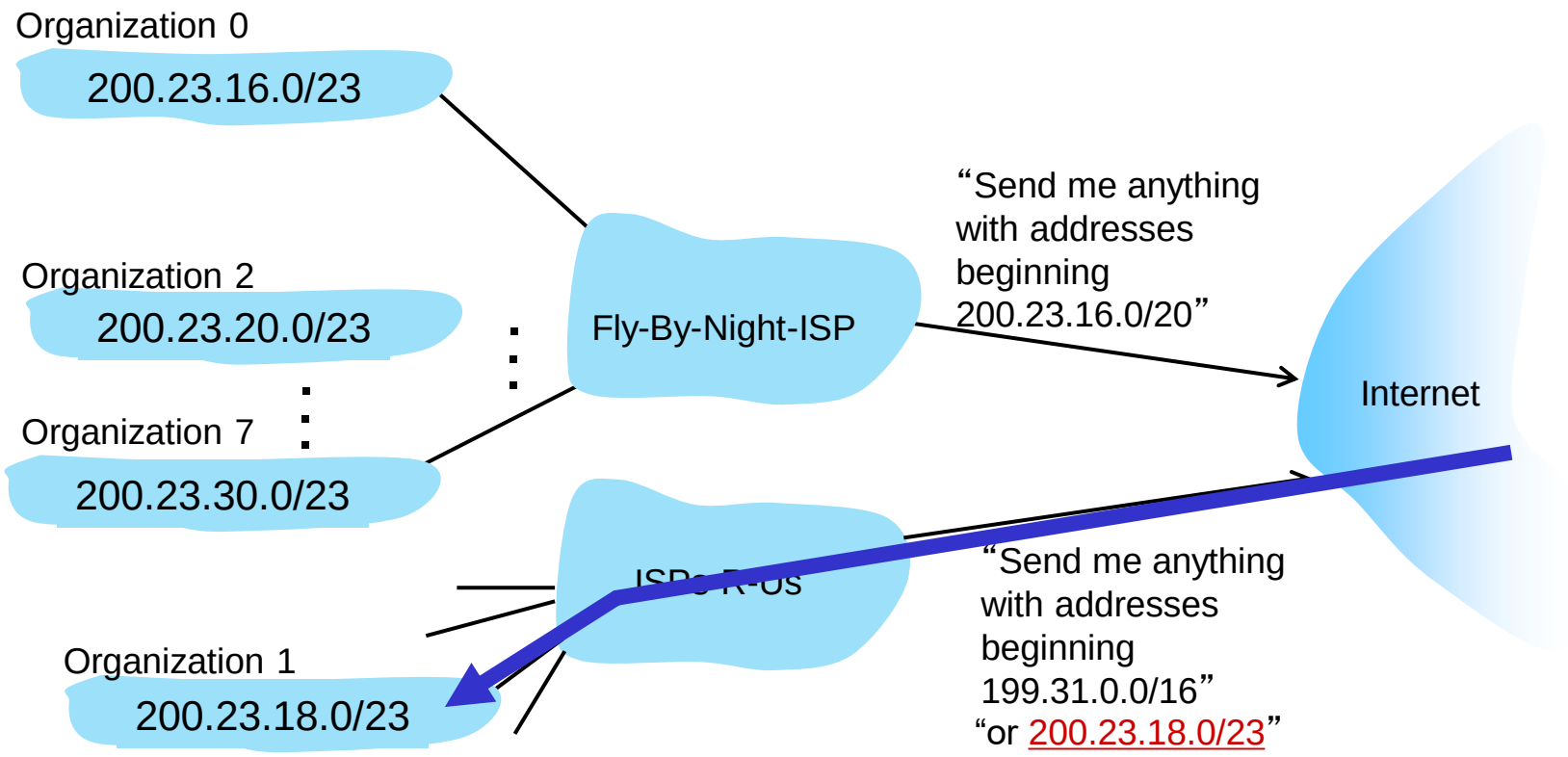
Hierarchical addressing: tuyến đường cụ thể

- Organization 1 di chuyển từ Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us quảng bá tuyến đường cụ thể Organization 1



Hierarchical addressing: more specific routes

- Organization 1 di chuyển từ Fly-By-Night-ISP to ISPs-R-Us
- ISPs-R-Us quảng bá tuyến đường cụ thể Organization 1



IP addressing: last words ...

Q: ISP lấy khối địa chỉ từ đâu?

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

- Cấp phát địa chỉ thông qua 5 regional registries (RRs) (5 tổ chức này sau đó sẽ cấp phát tới các ISP local)
- Quản lý DNS root zone, bao gồm cả quản lý TLD (.com, .edu , …)

Q: 32-bit IP addresses có đủ không?

- ICANN cấp phát chunk cuối cùng của IPv4 addresses đến RRs in 2011
- NAT (next) giúp giải quyết vấn đề cạn kiệt IPv4
- IPv6 có 128-bit

"Ai mà biết được chúng ta cần bao nhiêu địa chỉ?" Vint Cerf (reflecting on decision to make IPv4 address 32 bits long)

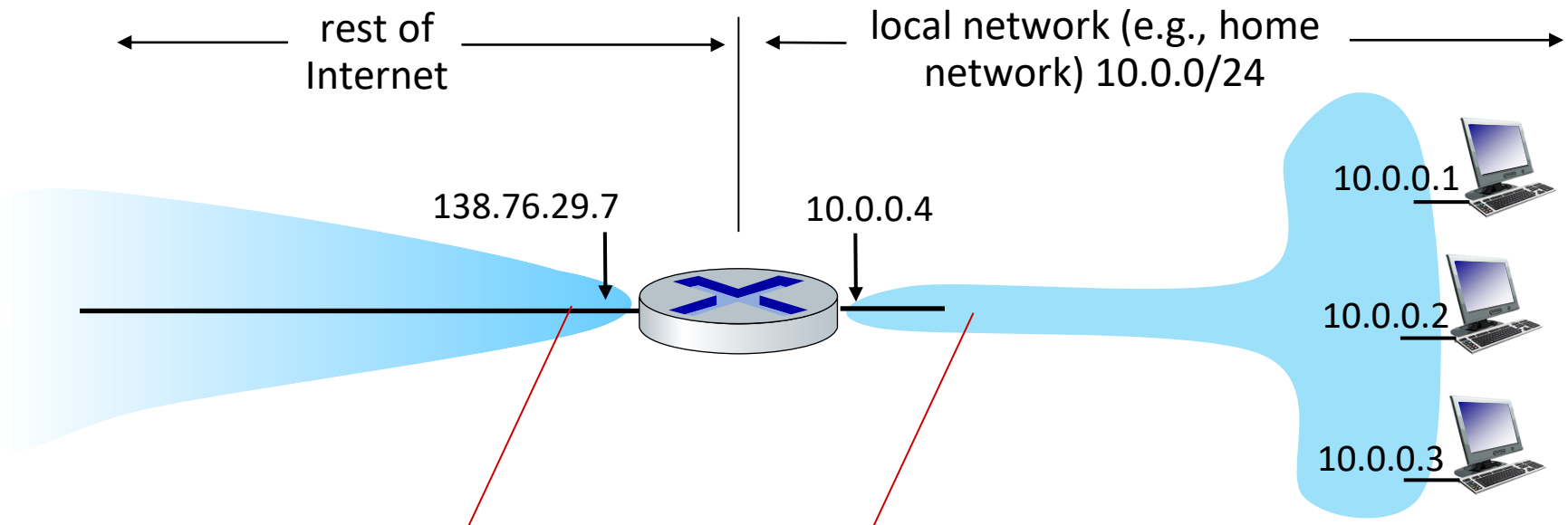
Network layer: "data plane" roadmap

- Network layer: overview
  - data plane
  - control plane
- What's inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized Forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes

NAT: network address translation

NAT: các thiết bị trong mạng cục bộ chia sẻ một địa chỉ IPv4 khi đi ra Internet



Tất cả các datagramsdời mạng cục bộ với cùng địa chỉ IP nguồn đã được NAT. : 138.76.29.7, nhưng khác địa chỉ cổng nguồn

datagrams với nguồn hoặc đích trong mạng này có địa chỉ trong mạng 10.0.0/24

NAT: network address translation

- Tất cả các thiết bị trong mạng cục bộ có địa chỉ 32-bit trong không gian địa chỉ riêng tư (10/8, 172.16/12, 192.168/16 prefixes) chỉ dùng cục bộ
- Ưu điểm:
  - Chỉ cần 1 địa chỉ từ ISP cho tất cả các thiết bị
  - Có thể thay đổi địa chỉ của host trong mạng cục bộ không cần thông tin với thế giới bên ngoài
  - Có thể thay đổi ISP mà không cần thay đổi địa chỉ các thiết bị trong mạng
  - Bảo mật: thiết bị bên trong mạng cục bộ không được xác định trực tiếp, không hiển thị với thế giới bên ngoài.

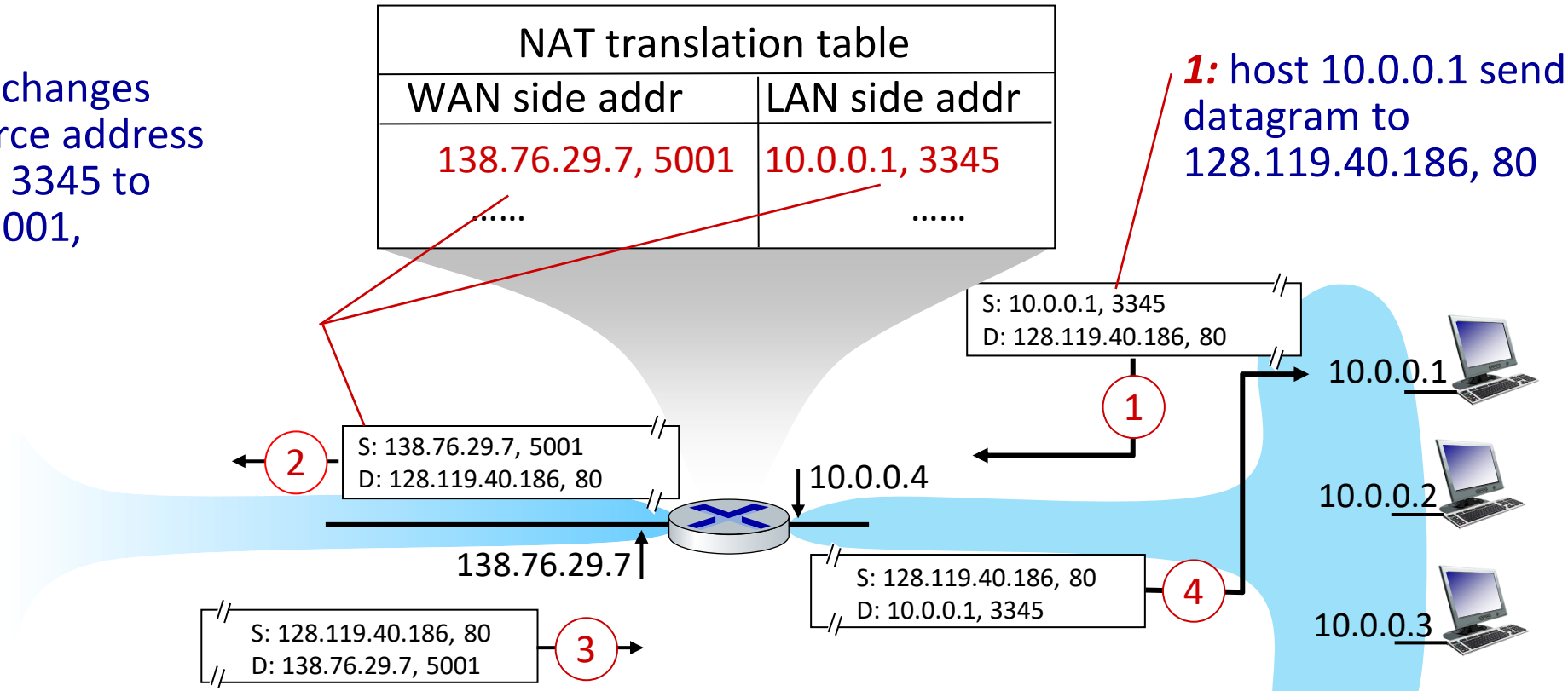
NAT: network address translation

triển khai: NAT phải trong suốt:

- Datagrams đi ra ngoài: thay thế (IP nguồn, cổng #) của mọi datagram sang (địa chỉ IP được NAT, cổng mới #)
  - clients/servers từ xa sẽ phản hồi lại dùng địa chỉ IP đã NAT và địa chỉ cổng mới như địa chỉ đích
- Nhớ (trong bảng NAT translation table) mọi cặp (source IP address, port #) được NAT (NAT IP address, new port #)
- Datagrams đi vào mạng: thay thế (NAT IP address, new port #) trong địa chỉ đích của mọi datagram đến với (source IP address, port #) lưu trong bảng NAT table

NAT: network address translation

2: NAT router changes datagram source address from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table



3: reply arrives, destination address: 138.76.29.7, 5001

NAT: network address translation

- NAT has been controversial:
  - routers “should” only process up to layer 3
  - address “shortage” should be solved by IPv6
  - violates end-to-end argument (port # manipulation by network-layer device)
  - NAT traversal: what if client wants to connect to server behind NAT?
- but NAT is here to stay:
  - extensively used in home and institutional nets, 4G/5G cellular nets

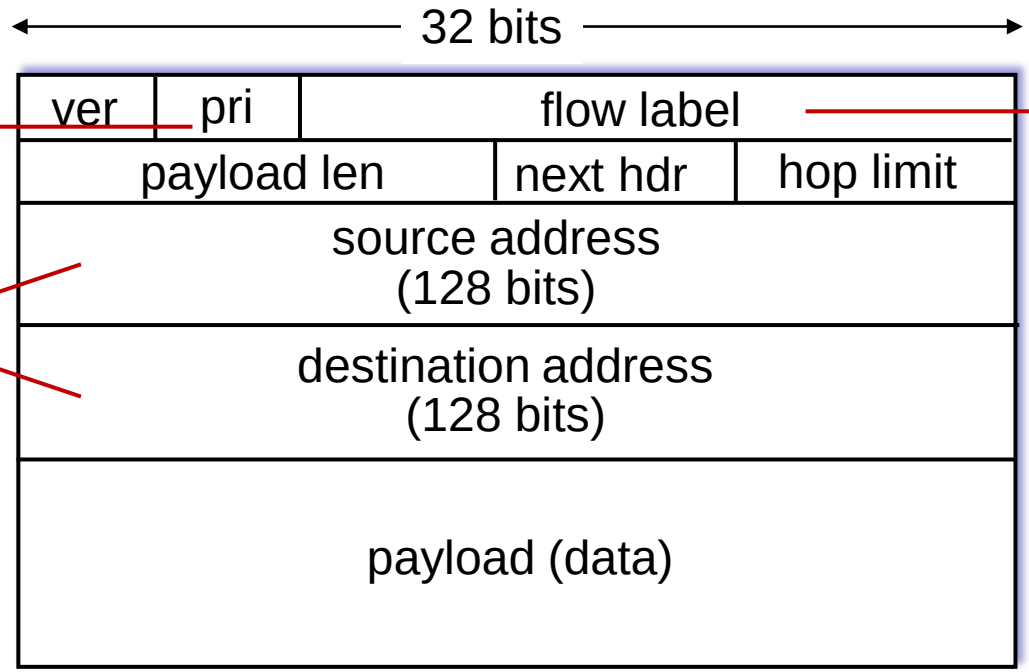
IPv6: dịch chuyển sang IPv6

- Động lực: không gian 32-bit IPv4 cạn kiệt
- additional motivation:
  - Tốc độ xử lý/forwarding: 40-byte fixed length header
  - Cho phép xử lý các luồng ở tầng mạng khác nhau

IPv6 datagram format

priority: identify priority among datagrams in flow

128-bit IPv6 addresses



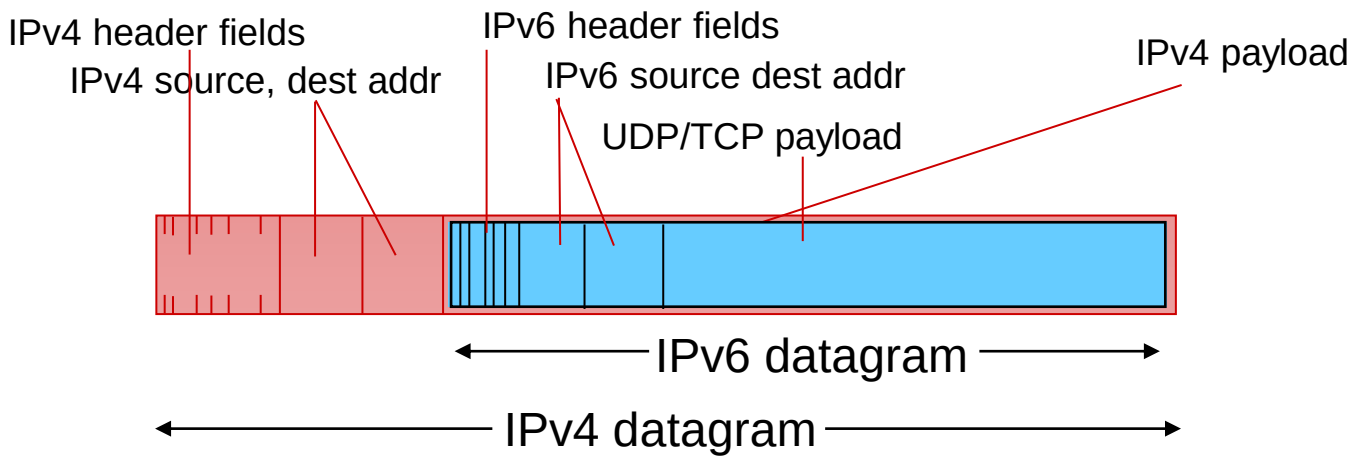
flow label: identify datagrams in same "flow." (concept of "flow" not well defined).

So với IPv4:

- no checksum (tăng tốc độ tại router)
- no fragmentation/reassembly
- no options (Tầng trên sẽ có trường này nếu cần thiết)

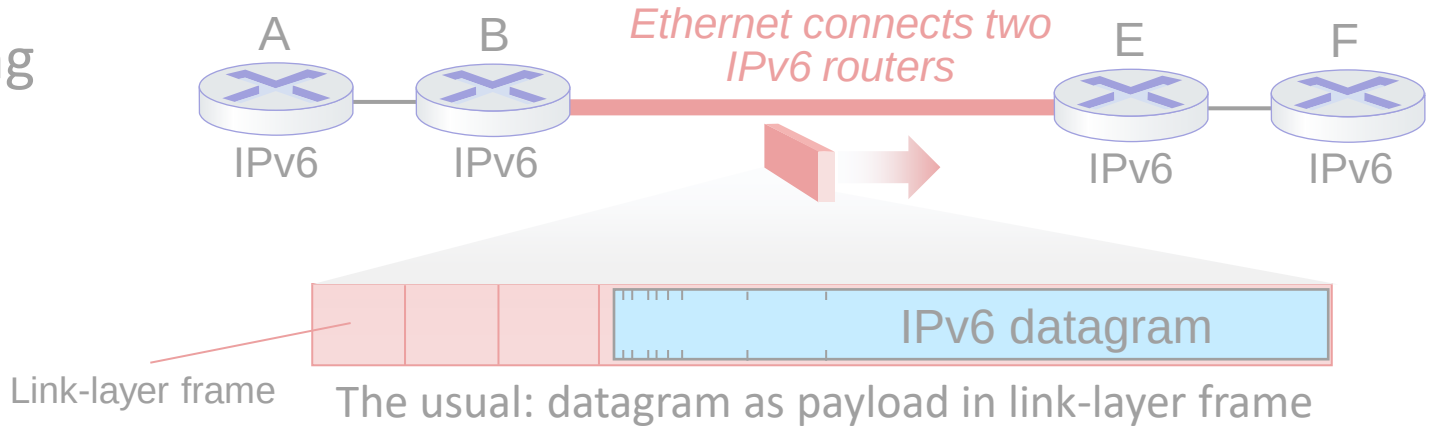
Từ IPv4 đến IPv6

- Không phải tất cả các router được nâng cấp đồng thời
  - no “flag days”
  - Mạng sẽ xử lý việc tồn tại cả IPv4 và IPv6 như thế nào
- Tunneling (đường hầm): IPv6 datagram được coi như *payload* trong IPv4 datagram tại các router IPv4 (“packet within a packet”)
  - Đường hầm được định nghĩa trong các kiểu mạng khác (4G/5G)

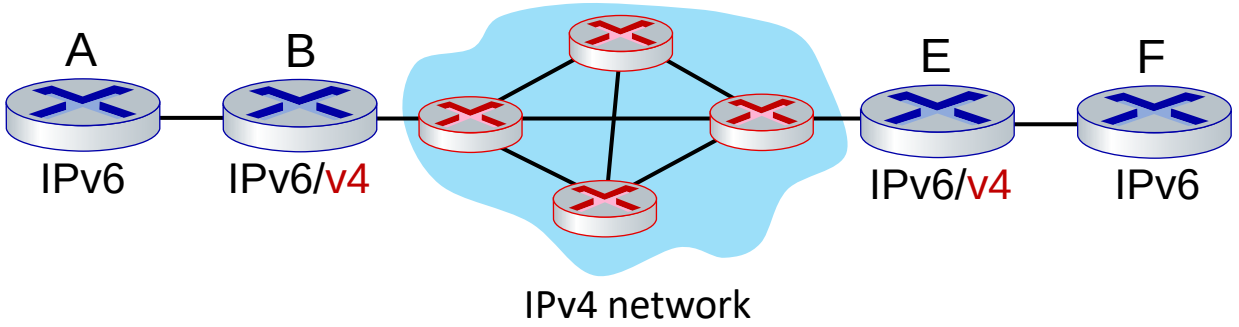


Đường hầm và đóng gói (Tunneling and encapsulation

Ethernet connecting two IPv6 routers:

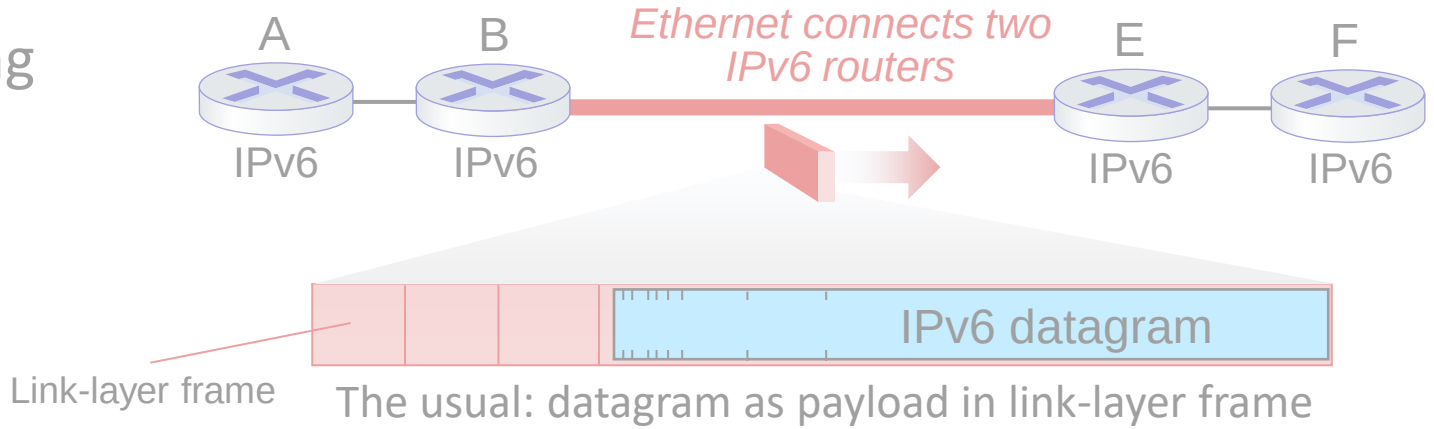


IPv4 network connecting two IPv6 routers



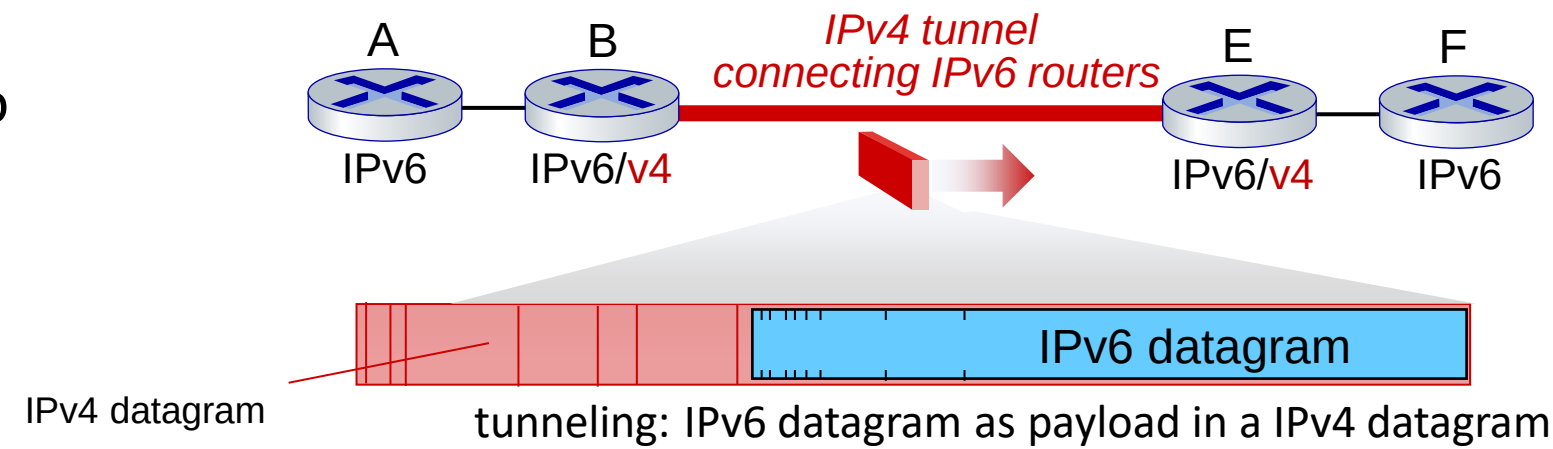
Tunneling and encapsulation

Ethernet connecting two IPv6 routers:



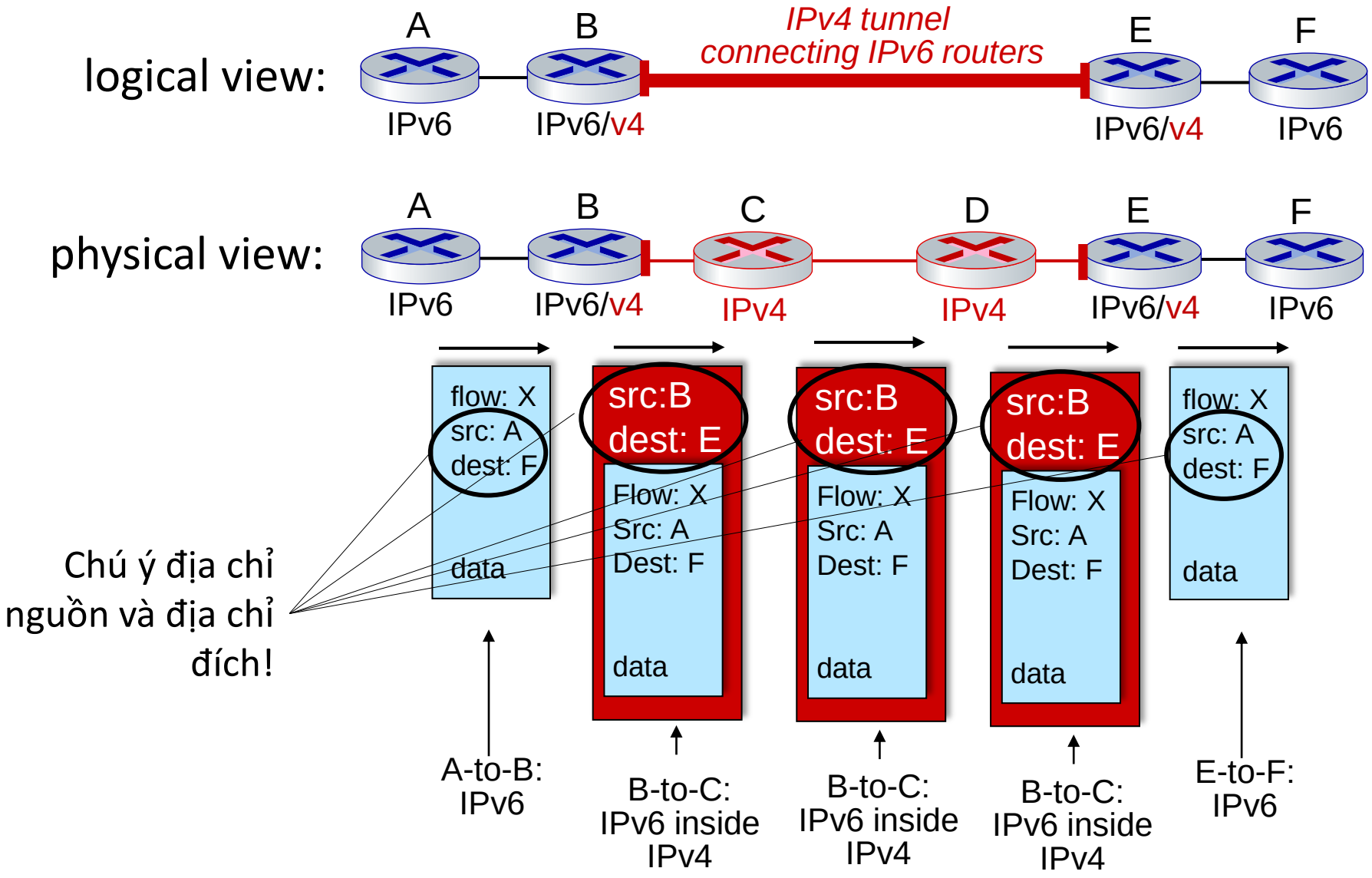
The usual: datagram as payload in link-layer frame

IPv4 tunnel connecting two IPv6 routers



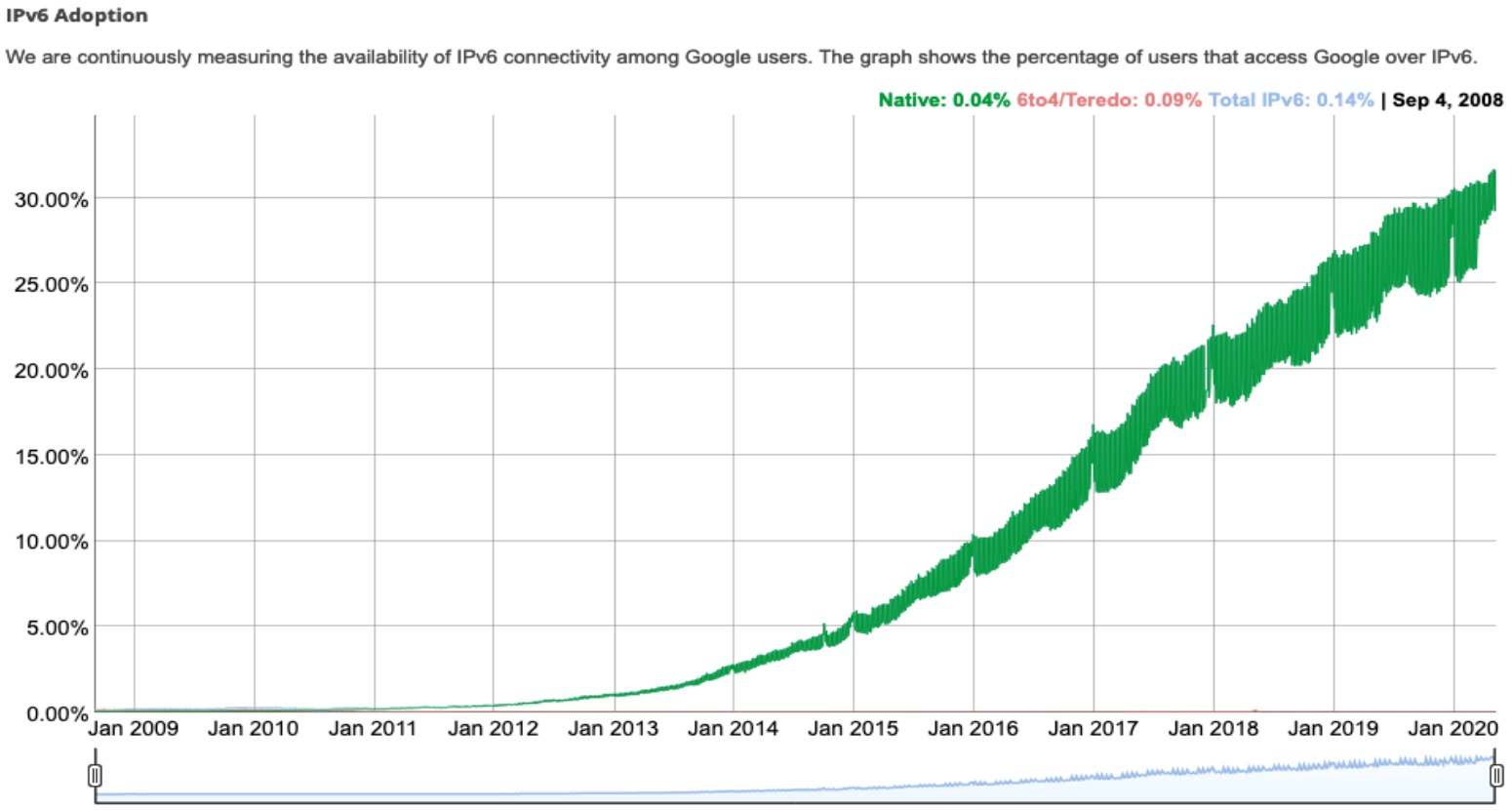
tunneling: IPv6 datagram as payload in a IPv4 datagram

Tunneling


logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6/v4
IPv6/v4
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6/v4
IPv4
IPv4
IPv6/v4
IPv6
Chú ý địa chỉ nguồn và địa chỉ đích!
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

IPv6: triển khai

- Google[1]: ~ 30% dịch vụ khách hàng là IPv6
- NIST: 1/3 domain của chính phủ Mỹ là có khả năng IPv6



1

https://www.google.com/intl/en/ipv6/statistics.html

IPv6: triển khai

- Google[1]: ~ 30% dịch vụ khách hàng là IPv6
- NIST: 1/3 domain của chính phủ Mỹ là có khả năng IPv6
- Thời gian rất dài để triển khai
  - 25 năm và hơn nữa!
  - Trong 25 có thể các dịch vụ tầng ứng dụng sẽ thay đổi rất nhiều: WWW, social media, streaming media, gaming, telepresence, …
  - Thực sự việc thay đổi giao thức tầng mạng là một câu chuyện quá khó …
  - *Why?*

[1] https://www.google.com/intl/en/ipv6/statistics.html

Network layer: "data plane" roadmap

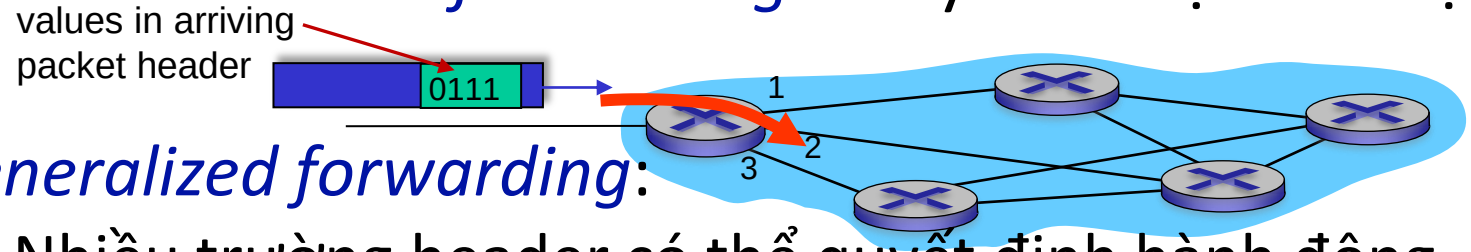
- Network layer: overview
  - data plane
  - control plane
- What's inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized Forwarding, SDN
  - Match+action
  - OpenFlow: match+action in action
- Middleboxes

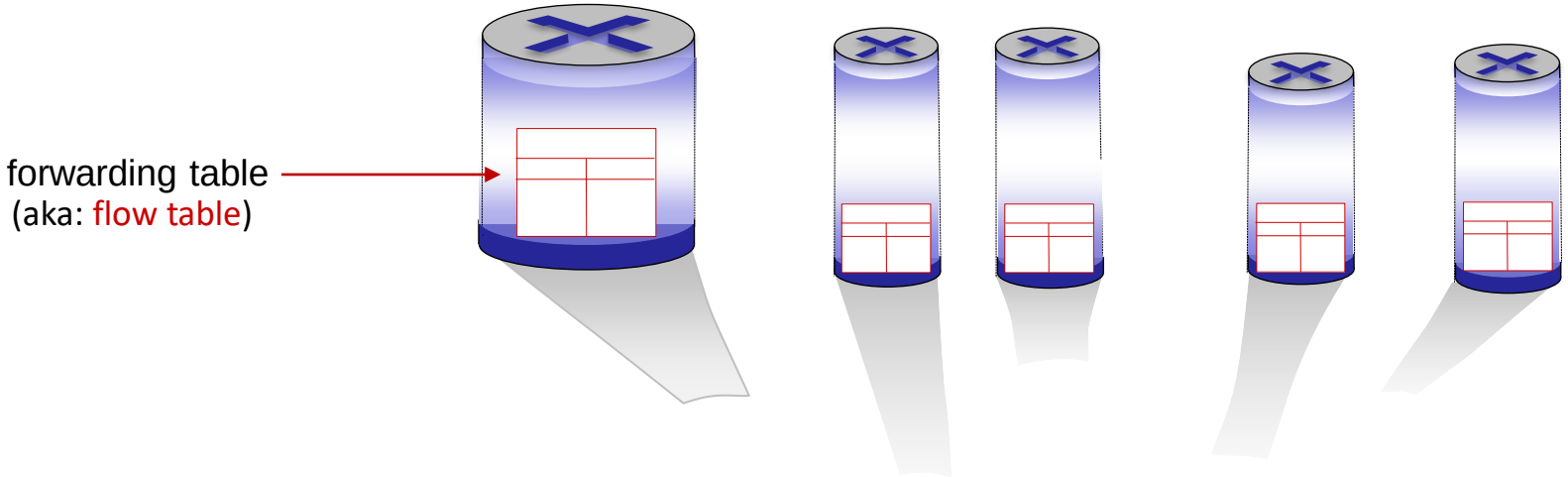
Generalized forwarding (chuyển tiếp tập chung): match plus action

Review: mỗi router có một bảng forwarding table (aka: flow table)

- "match plus action" : khớp bit trong gói tin đến, có hành động tương ứng
  - *destination-based forwarding:* chuyển đi dựa vào địa chỉ IP đích

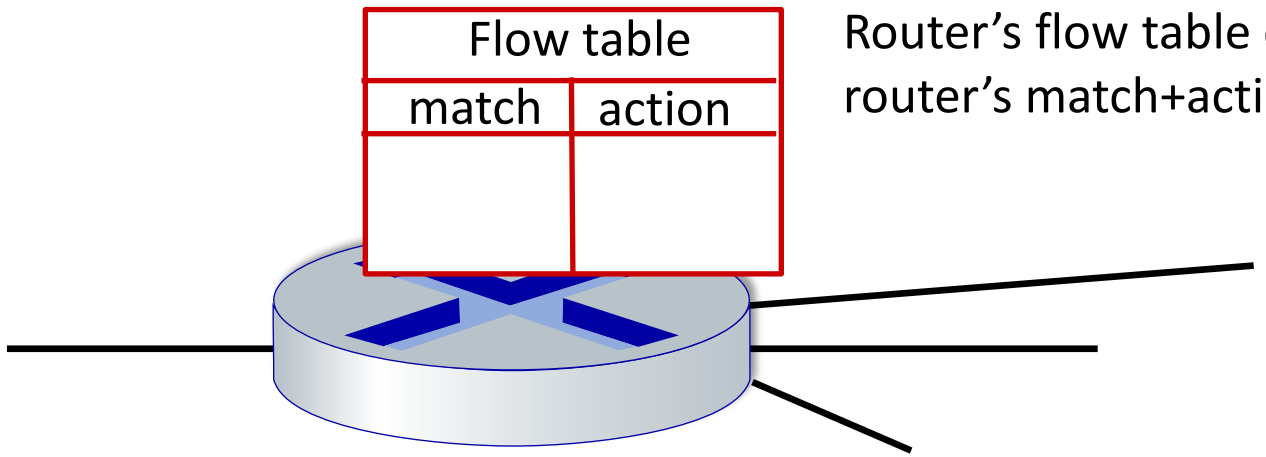


  - *generalized forwarding*:
    - Nhiều trường header có thể quyết định hành động
    - Các hành động có thể: drop/copy/modify/log packet



Flow table

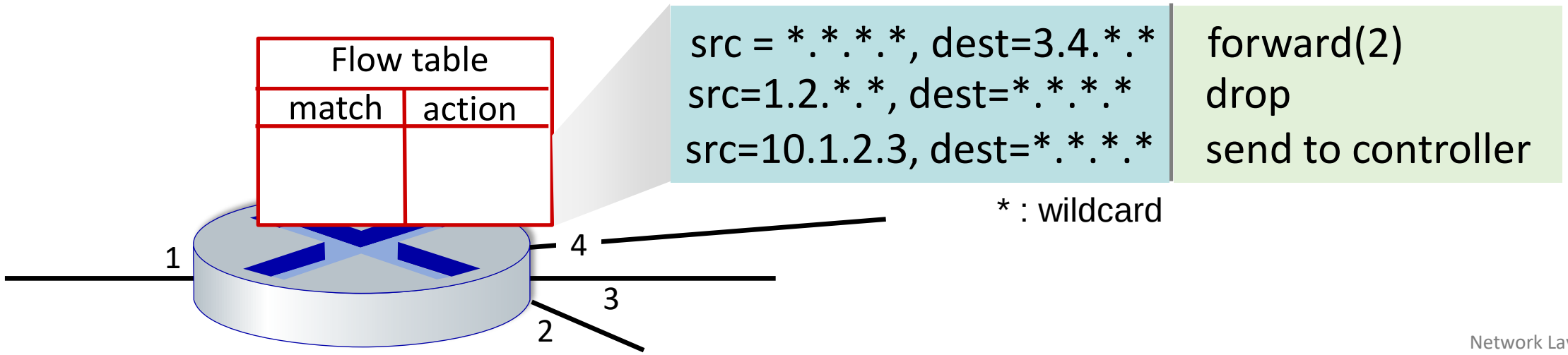
- flow: được định nghĩa với giá trị của trường header tại nhiều tầng(in link-, network-, transport-layer fields)
- generalized forwarding: luật đơn giản rules
  - match: so khớp giá trị trong các trường của header
  - actions: sau khi đã khớp: vứt bỏ, chuyển đi, chỉnh sửa các gói đã khớp hoặc gửi gói đã khớp tới controller
  - priority: phân biệt các mẫu bị chồng chéo
  - counters: số byte (#bytes) và số gói (#packets)



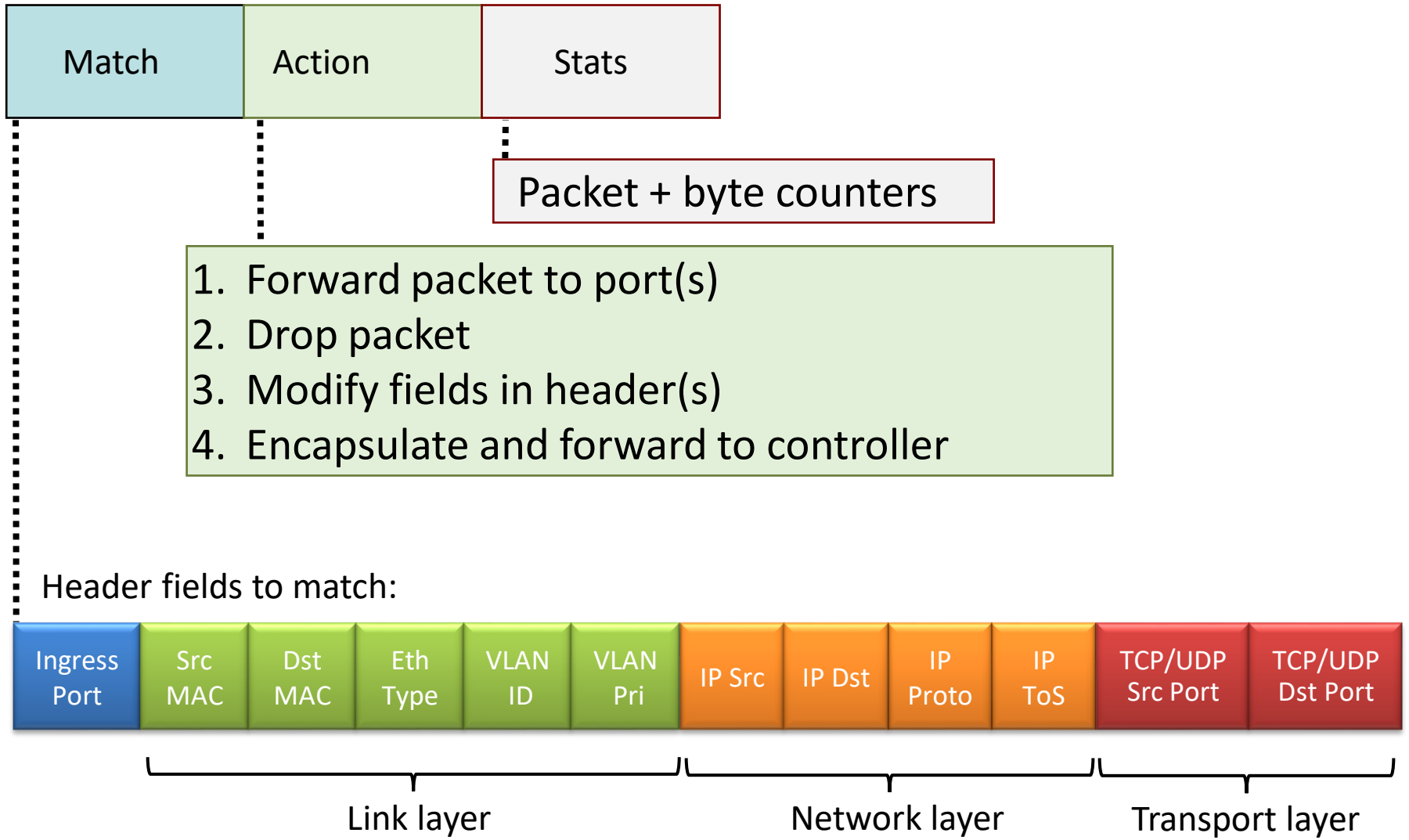
Router's flow table define router's match+action rules

Flow table abstraction

- flow: được định nghĩa với giá trị của trường header (in link-, network-, transport-layer fields)
- generalized forwarding: luật đơn giản rules
  - match: so khớp giá trị trong các trường của header
  - actions: sau khi đã khớp: vứt bỏ, chuyển đi, chỉnh sửa các gói đã khớp hoặc gửi gói đã khớp tới controller
  - priority: phân biệt các mẫu bị chồng chéo
  - counters: số byte (#bytes) và số gói (#packets)



OpenFlow: các flow table



OpenFlow: ví dụ

Gửi dựa trên địa chỉ đích:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	51.6.0.8	*	*	*	*	port6

IP datagrams dành cho dích 51.6.0.8 nên hướng ra cổng router là 6

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	22	drop

Chặn (không forward) tất cả datagrams trên cổng TCP port 22 (ssh port #)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	drop

Chặn (do not forward) all datagrams gửi bởi host 128.119.1.1

OpenFlow: ví dụ

Chuyển tiếp dựa vào địa chỉ tầng 2:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	22:A7:23:11:E1:02	*	*	*	*	*	*	*	*	*	port3

layer 2 frames với MAC dích là 22:A7:23:11:E1:02 hướng ra ngoài cổng 3

OpenFlow abstraction

- match+action: thống nhất các loại thiết bị

Router

- *match:* so khớp số bit dài nhất trong IP prefix
- *action:* forward out a link

Switch

- *match:* so khớp MAC đích
- *action:* forward or flood

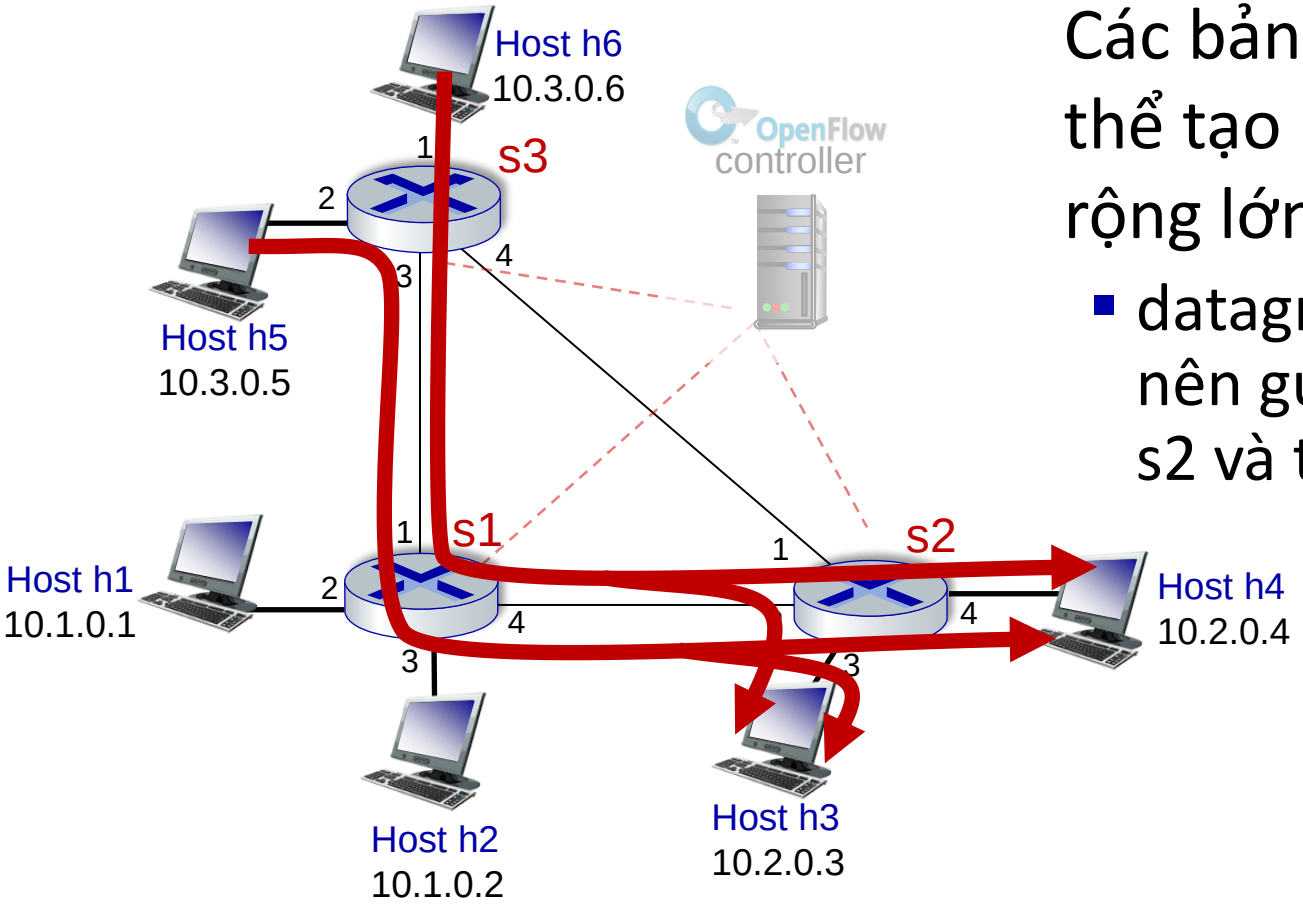
Firewall

- *match*: so khớp địa chỉ IP và địa chỉ cổng
- *action:* permit or deny

NAT

- *match:* địac chỉ IP và địa chỉ cổng
- *action:* thay đổi địa chỉ IP và địa chỉ cổng

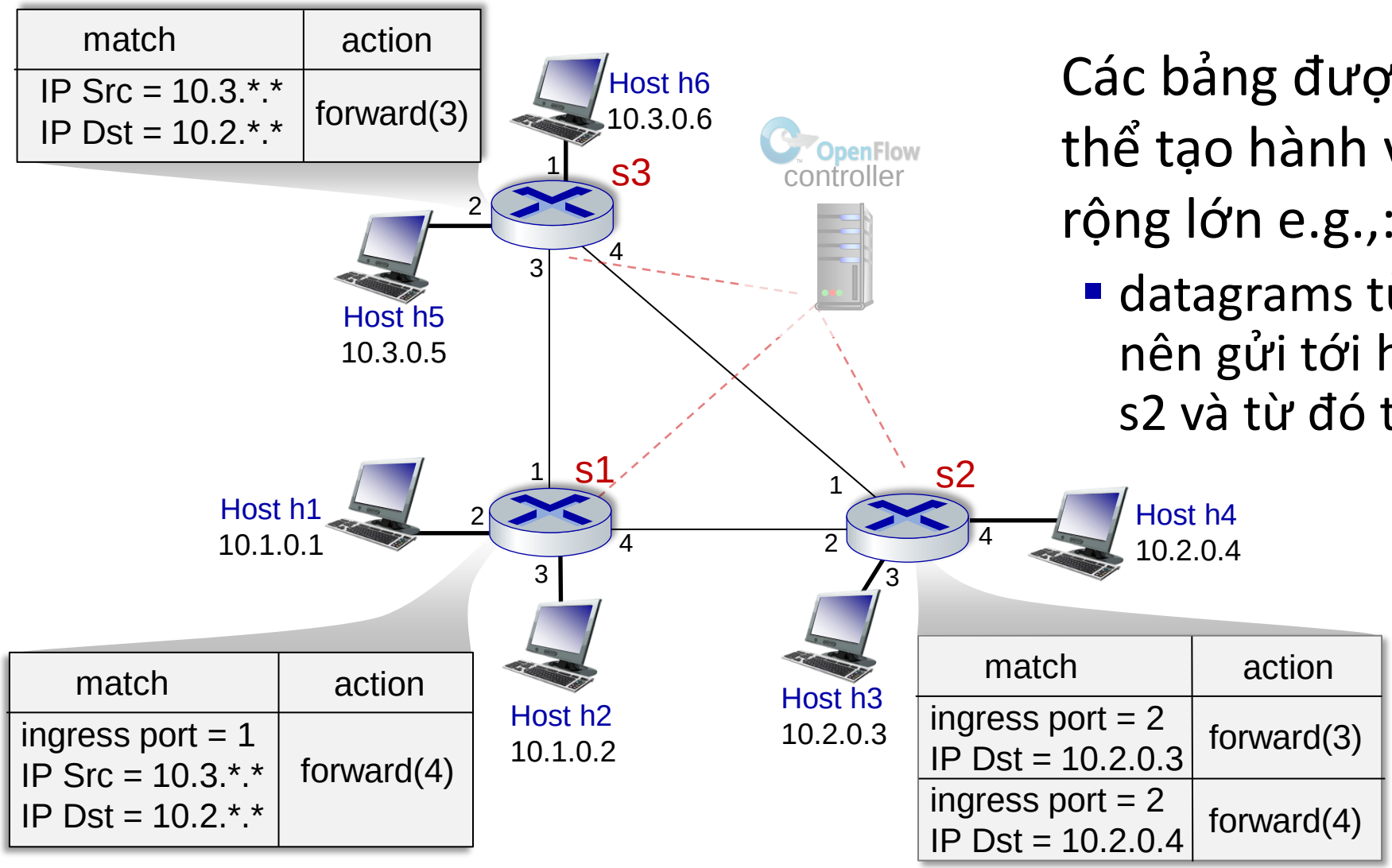
OpenFlow example



Các bảng được tổ chức tốt có thể tạo hành vi cho một mạng rộng lớn e.g.,:

- datagrams từ host h5 và h6 nên gửi tới h3 hoặc h4, theo s2 và từ đó tới s2

OpenFlow example



Các bảng được tổ chức tốt có thể tạo hành vi cho một mạng rộng lớn e.g.,:

- datagrams từ host h5 và h6 nên gửi tới h3 hoặc h4, theo s2 và từ đó tới s2

Generalized forwarding: tóm tắt

- "match plus action" abstraction: khớp các bit trong gói tin đến tại bất cứ tầng nào và hành động
  - So khớp nhiều trường của các tầng khác nhau (link-, network-, transport-layer)
  - local actions: drop, forward, modify, hoặc gửi gói tin so khớp tới controller
  - Chương trình hành vi cho toàn mạng
- Mạng có thể lập trình
  - Có thể lập trình, xử lý trên gói
  - *historical roots:* active networking
  - *today:* lập trình được phổ biến có thể xem trong https://p4.org/

Network layer: “data plane” roadmap

- Network layer: overview
- What’s inside a router
- IP: the Internet Protocol
- Generalized Forwarding
- Middleboxes
  - middlebox functions
  - evolution, architectural principles of the Internet

Middleboxes

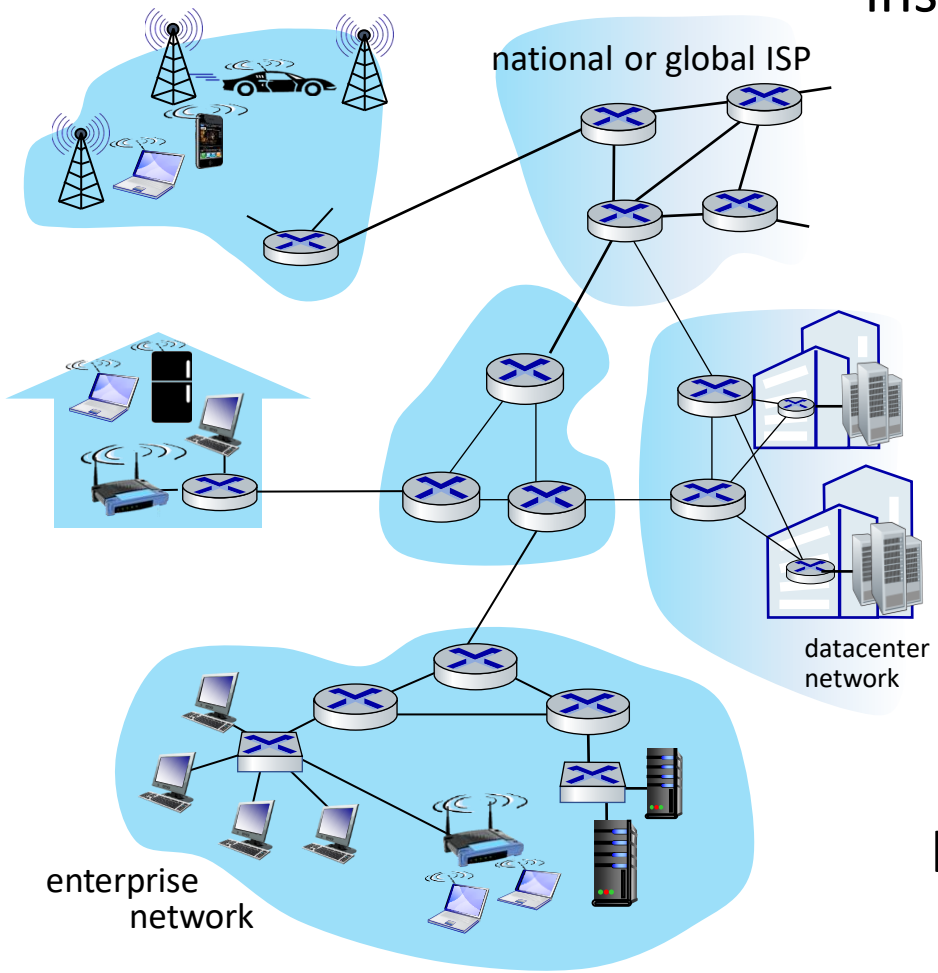
Middlebox (RFC 3234)

"bất kì box trung gian có chức năng khác ngoài các chức năng được tiêu chuẩn của router IP trên đường đi của dữ liệu từ nguồn tới đích"

Middleboxes ở mọi nơi!

NAT: home, cellular, institutional

Application-specific: service providers, institutional, CDN



Firewalls, IDS: corporate, institutional, service providers, ISPs

Load balancers: corporate, service provider, data center, mobile nets

Caches: service provider, mobile, CDNs

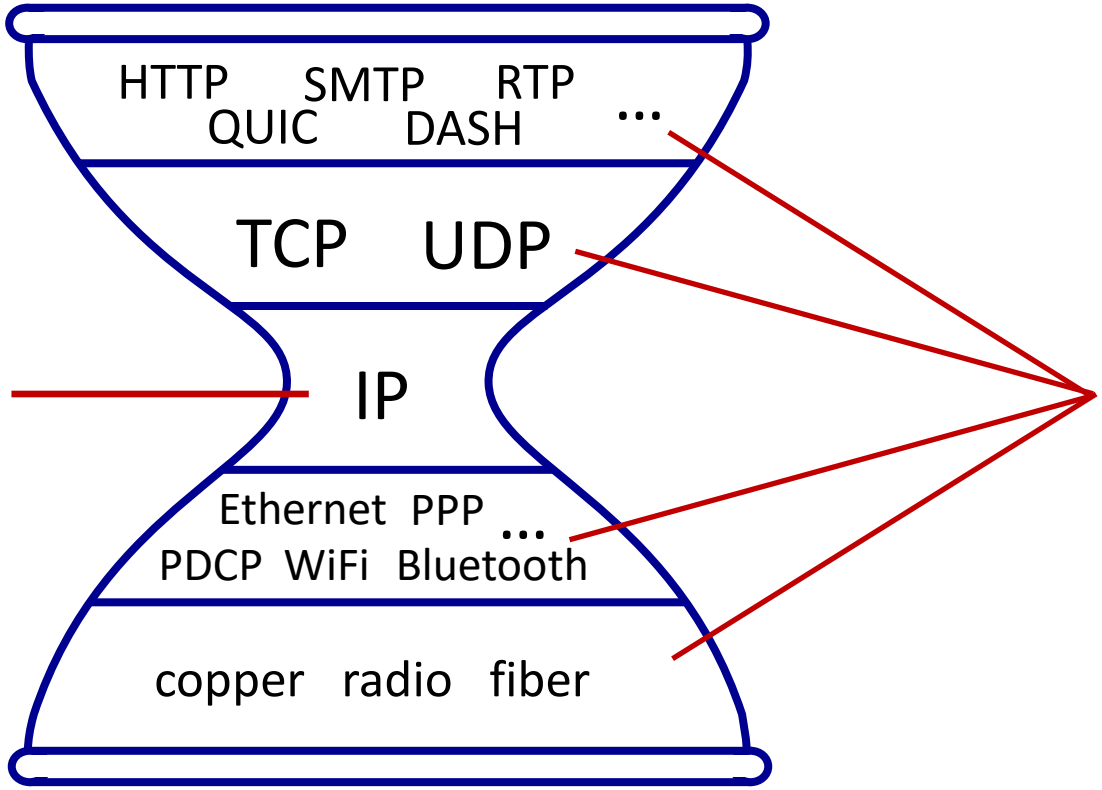
Middleboxes

- Ban đầu: là các giải pháp phần cứng độc quyền
- Chuyển sang phần cứng "whitebox" thực thi các open API
  - Thoát khỏi các giải pháp phần cứng độc quyền
  - Hành động cục bộ có thể lập trình được bằng cách match+action
  - Hướng tới sự cải tiến/khác biệt trong phần mềm
- SDN: (logically) kiểm soát tập trung và quản lý cấu hình trong private/public cloud
- network functions virtualization (NFV): Các dịch vụ và chức năng mạng có thể lập trình được

The IP hourglass (đồng hồ cát)

Internet's "thin waist":

- *Chỉ một giao thức tầng mạng*: IP
- *Phải thực thi hàng tỉ*(billions) kết nối giữa các thiết bị trên Internet

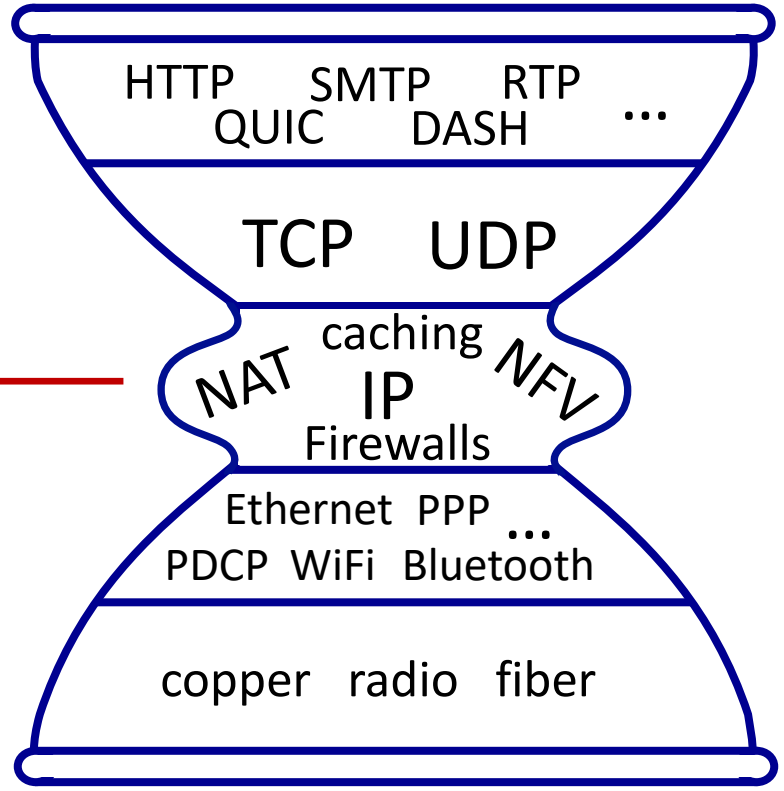


Trong khi các tầng khác có nhiều giao thức tại tầng dịch vụ, tầng vận chuyển, tầng link, tầng vật lý

The IP hourglass, at middle age

Internet's middle age "love handles"?

- Có nhiều hơn middlebox xử lý trong tầng mạng



Architectural Principles of the Internet

RFC 1958

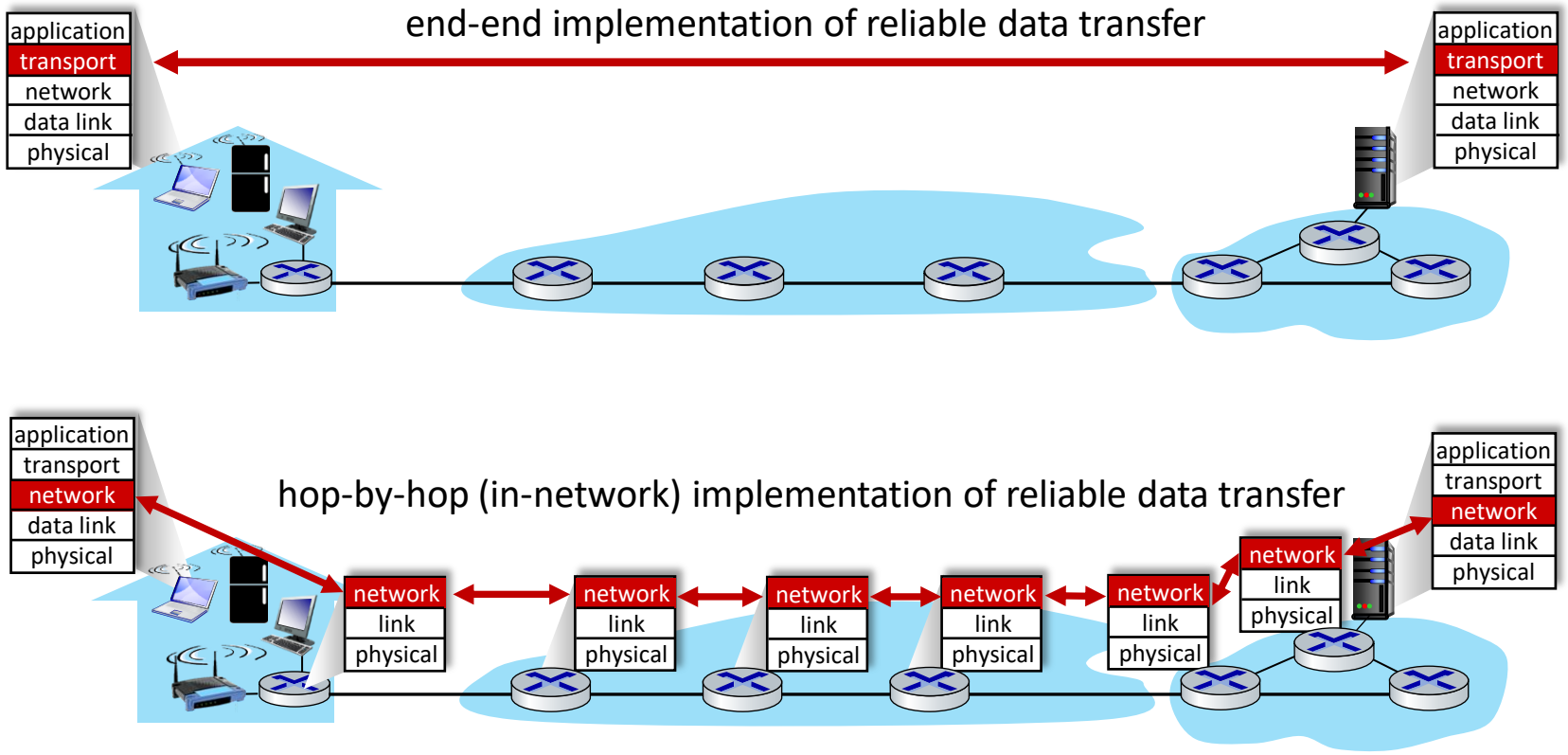
"Many members of the Internet community would argue that there is no architecture, but only a tradition, which was not written down for the first 25 years (or at least not by the IAB). However, in very general terms, the community believes that the goal is connectivity, the tool is the Internet Protocol, and the intelligence is end to end rather than hidden in the network."

Ba quan điểm nền tảng:

- Kết nối đơn giản
- IP protocol: điểm mấu chốt ở giữa
- Thông minh, phức tạp để tại biên của mạng

Tranh luận end-end

- Một số chức năng của mạng ví dụ truyền tin cậy, xử lý tắc nghẽn có thể thực thi tại trong mạng hoặc biên mạng

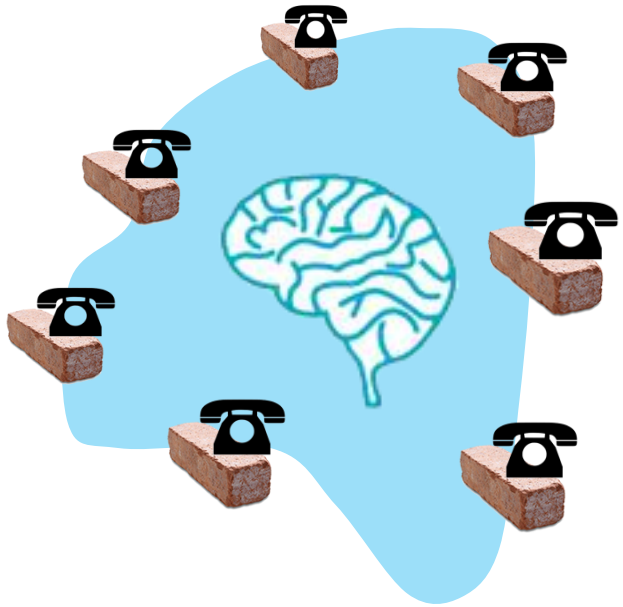


Tranh luận end-end

- Một số chức năng của mạng ví dụ truyền tin cậy, xử lý tắc nghẽn có thể thực thi tại trong mạng hoặc biên mạng

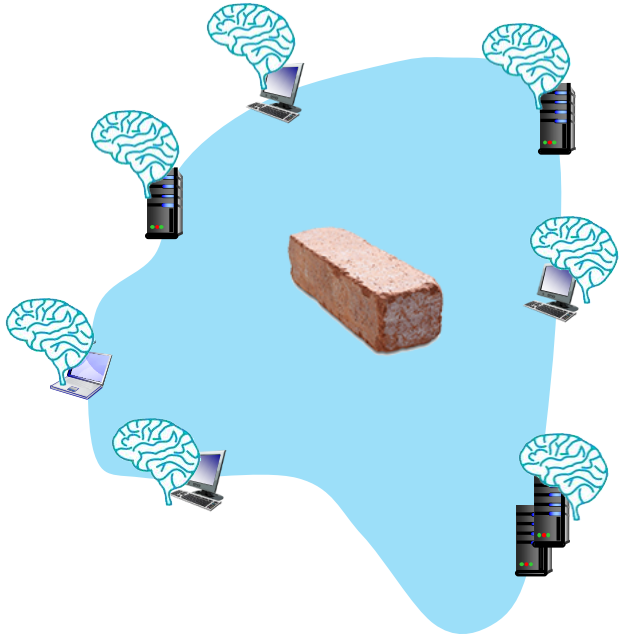
> "The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)
>
> We call this line of reasoning against low-level function implementation the "end-to-end argument."
>
> Saltzer, Reed, Clark 1981

Thông minh ở đâu?



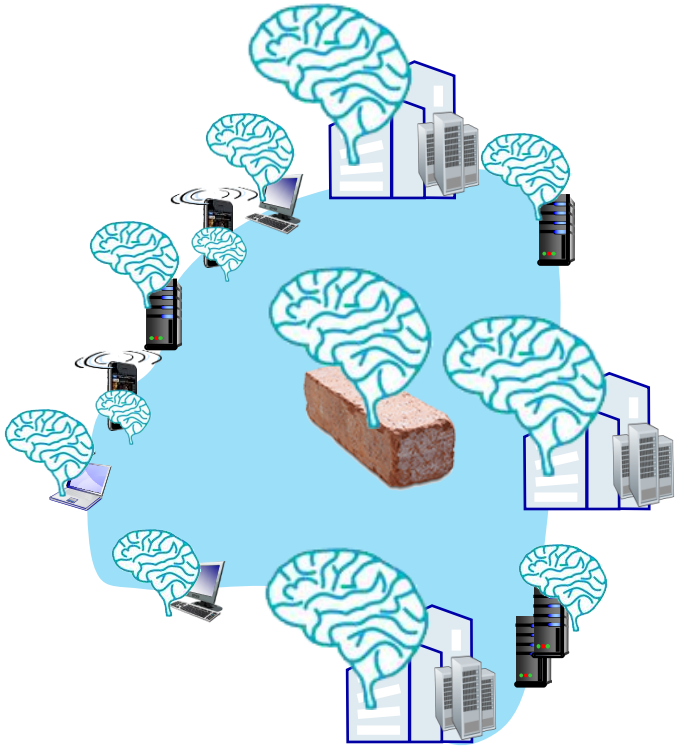
20th century phone net:

- intelligence/computing at network switches



Internet (pre-2005)

- intelligence, computing at edge



Internet (post-2005)

- programmable network devices
- intelligence, computing, massive application-level infrastructure at edge

Chapter 4: Xong!

- Network layer: overview
- What's inside a router
- IP: the Internet Protocol
- Generalized Forwarding, SDN
- Middleboxes

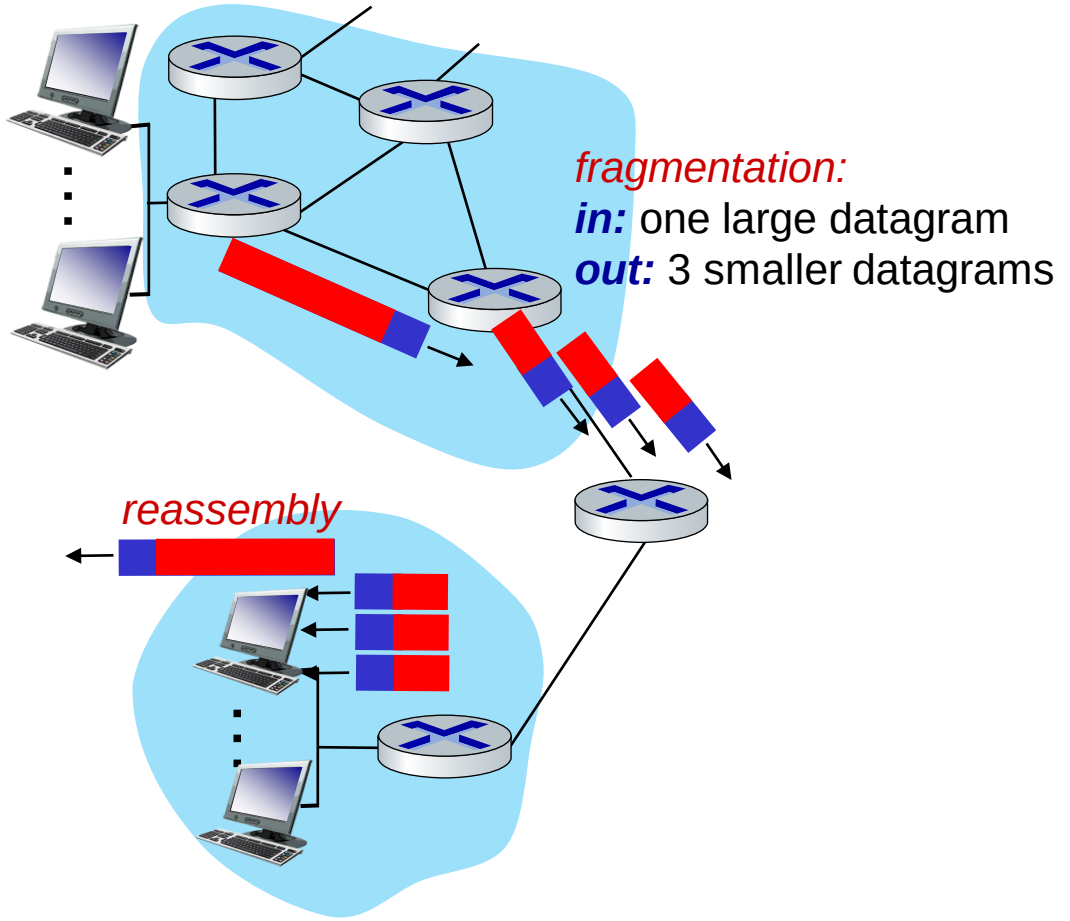
Question: how are forwarding tables (destination-based forwarding) or flow tables (generalized forwarding) computed?

Answer: by the control plane (next chapter)

Additional Chapter 4 slides

Phân mảnh IP /ráp lại (fragmentation/reassembly)

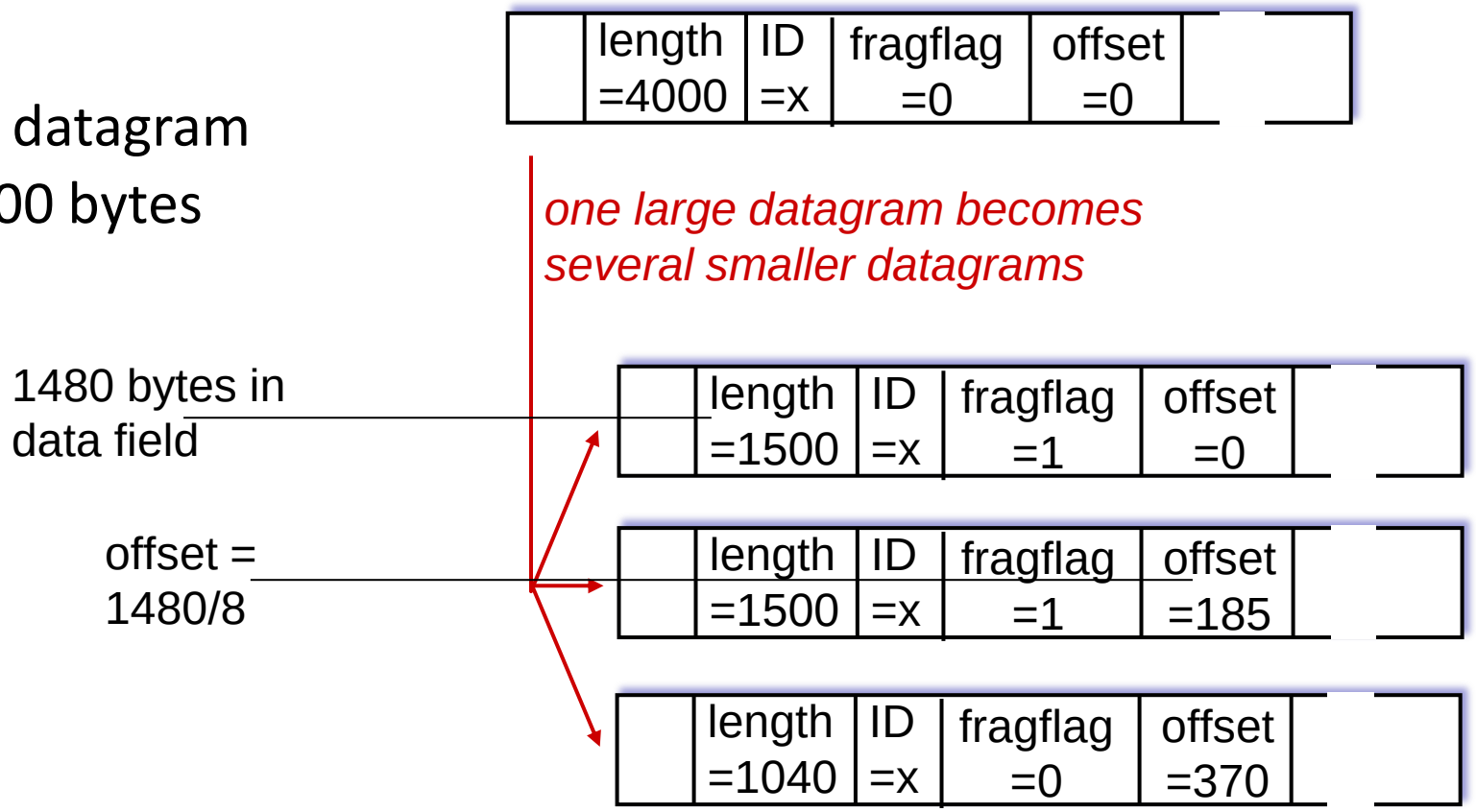
- network links có MTU (max transfer size) – lớn hơn khả năng link-level frame
  - link khác nhau, MTU khác nhau
- IP datagram lớn bị chia nhỏ ("fragmented") trong phạm vi mạng
  - 1 datagram thành nhiều several datagrams
  - "reassembled" chỉ tại đích
  - Các bit trong IP header dùng định danh các mảnh theo thứ tự



IP fragmentation/reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes



DHCP: Wireshark output (home LAN)

request

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) **DHCP Message Type = DHCP Request**
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
    ……

reply

Message type: **Boot Reply (2)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
    IP Address: 68.87.71.226;
    IP Address: 68.87.73.242;
    IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."