Chapter 6
The Link Layer and LANs

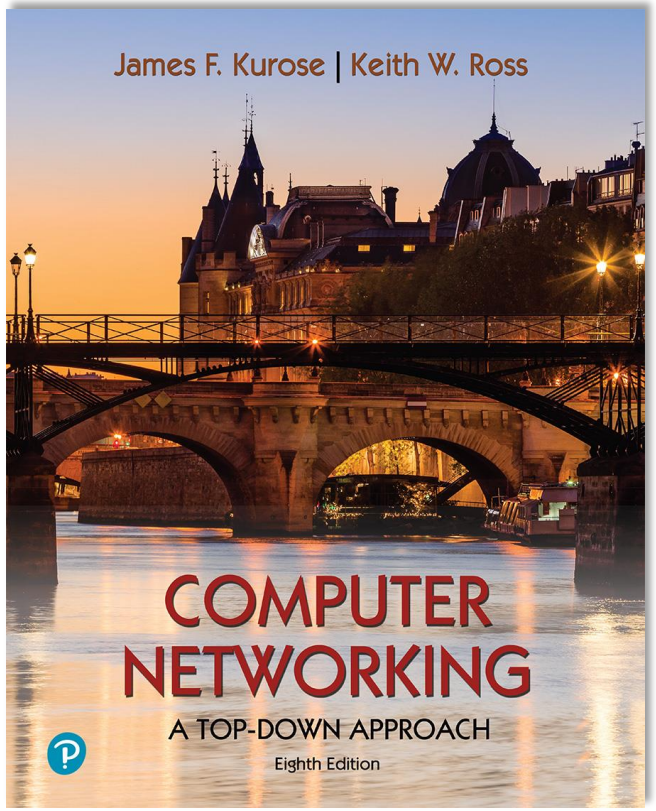
A note on the use of these PowerPoint slides:

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

Link layer and LANs: mục tiêu

- Nguyên lý:
  - Phát hiện lỗi, sửa lỗi
  - Chia sẻ kênh quảng bá: multiple access
  - Địa chỉ link layer
  - Mạng cục bộ: Ethernet, VLANs
- Mạng datacenter

- Công nghệ mạng tại tầng link layer

Link layer, LANs: roadmap

- Giới thiệu
- Phát hiện và sửa lỗi
- Giao thức đa truy cập
- LANs
  - Địa chỉ, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

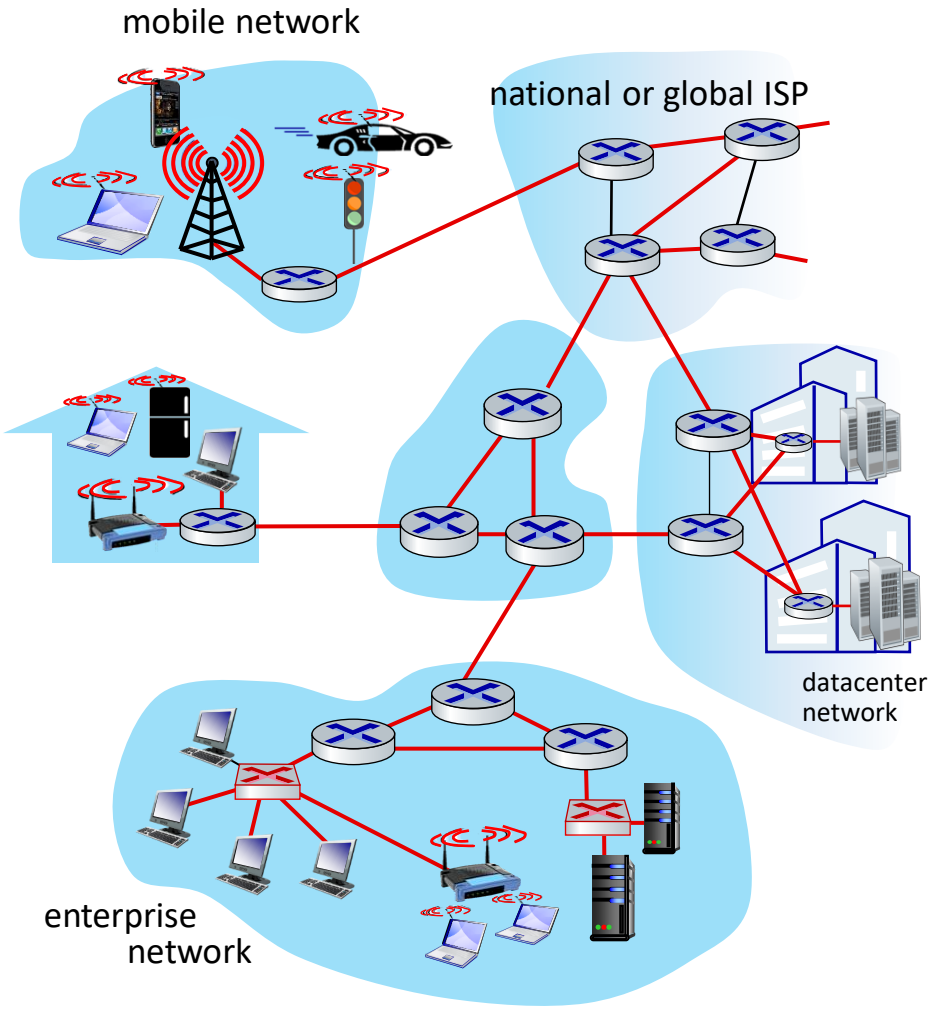
- a day in the life of a web request

Link layer: giới thiệu

Thuật ngữ:

- hosts và routers: nodes
- Kênh kết nối các node lại với nhau: links
  - wired
  - wireless
  - LANs
- layer-2 packet: *frame*, đóng gói datagram tầng network

link layer có nhiệm vụ truyền datagram từ 1 node tới node vật lý Thông qua một link



Link layer: ngữ cảnh

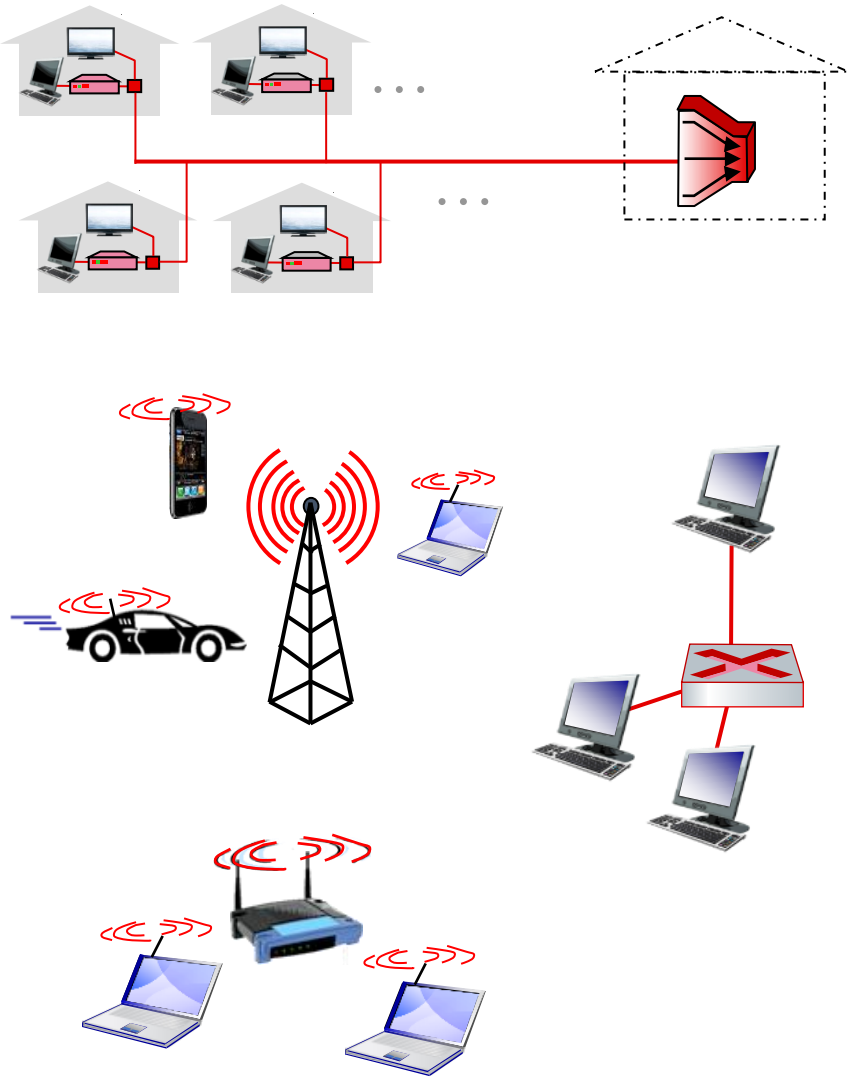
- datagram được truyền bởi giao thức link khác nhau thông qua các link khác nhau:
  - e.g., WiFi trên link đầu tiên, Ethernet trên link kế tiếp
- Mỗi link cung cấp dịch vụ khác nhau
  - e.g., có thể tin cậy hoặc không tin cậy

Ví dụ tương tự:

- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link-layer protocol
- travel agent = routing algorithm

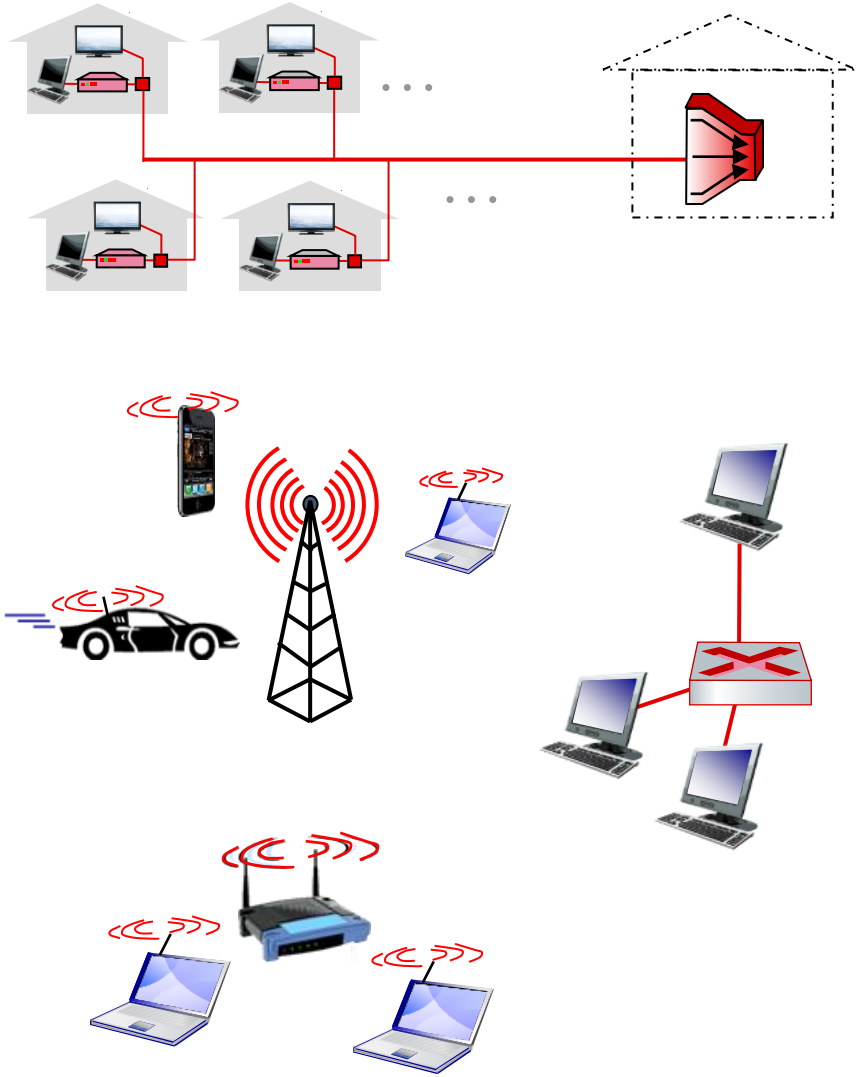
Link layer: dịch vụ

- Đóng gói:
  - Đóng gới datagram vào frame, thêm header, trailer
  - Chia sẻ môi trường truyền
  - Địa chỉ "MAC" addresses trong frame headers xác định nguồn và đích (khác với địa chỉ IP.
- Tin cậy giữa các node liền kề
  - Hiếm khi trên các link có tỉ lệ bit lỗi thấp
  - wireless links: tỉ lễ lỗi cao
    - *Q:* tại sao cấp độ link và end-end tin cậy?



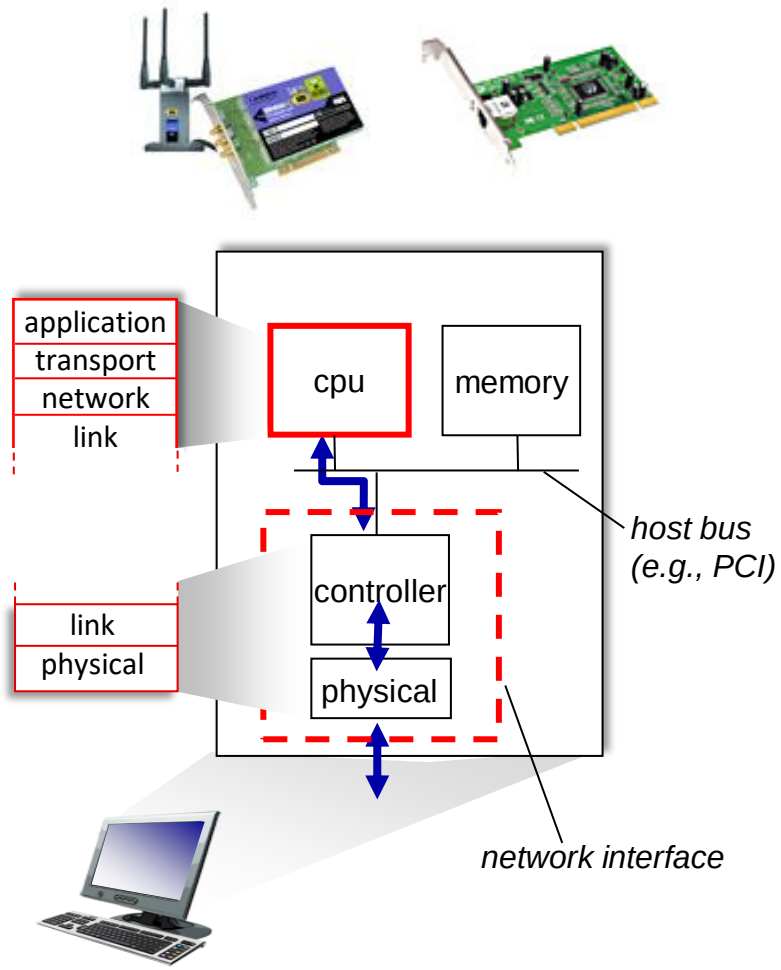
Link layer: dịch vụ

- Kiểm soát luồng:
  - Giữ nhịp gửi giữa hai node láng giềng
- Phát hiện lỗi:
  - Lỗi gây ra bởi nhiễu, …
  - Người nhận phát hiện lỗi, tín hiệu có thể được truyền lại hoặc frame bị vứt bỏ
- Sửa lỗi:
  - Người nhận xác định bit lỗi và sửa không cần truyền lại
- half-duplex và full-duplex:
  - Với half duplex, các nodes tại các thiết bị cuối có thể truyền nhưng không đồng thời.

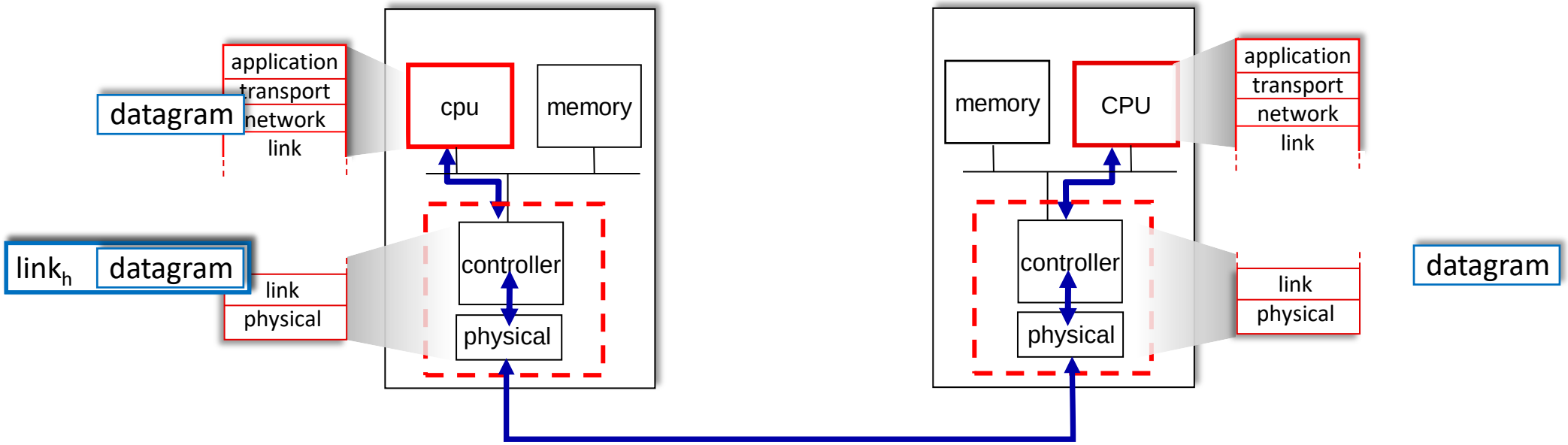


Link layer thực hiện ở đâu?

- Mỗi host (máy tính, ..)
- link layer thực thi tại *network interface card* (NIC) hoặc trên chip
  - Ethernet, WiFi card hoặc chip
  - Thực thi link, physical layer
- Đính kèm vào bus của host
- Kết hợp phần cứng phần mềm, hệ điều hành



Interfaces kết nối



Bên gửi:

- Đóng gói datagram thành frame
- Thêm bit kiểm tra lỗi, truyền tin cậy, kiểm soát luồng

Bên nhận:

- Kiểm tra lỗi, truyền tin cậy, kiểm soát luồng, etc.
- Trích rút datagram, truyền lên tầng trên phía bên nhận

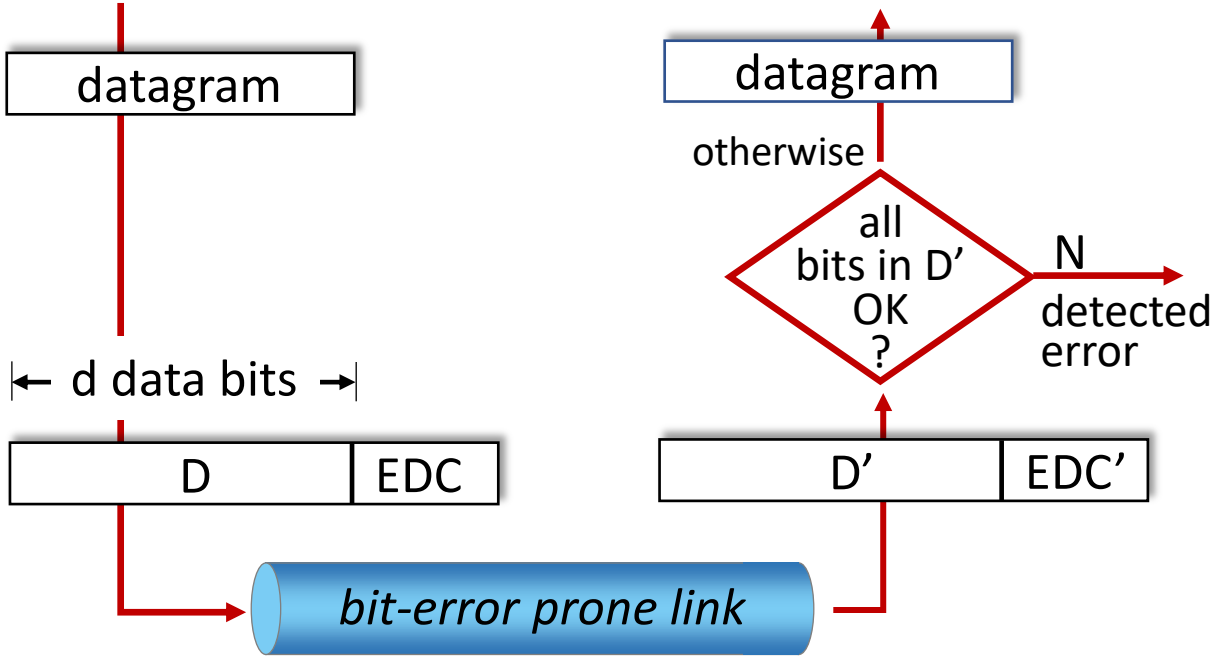
Link layer, LANs: roadmap

- introduction
- Phát hiện và sửa lỗi
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Phát hiện lỗi

EDC: bit phát hiện và sửa lỗi (e.g., dư thừa)
D: bảo vệ dữ liệu bằng kiểm tra lỗi có thể bao gồm trong các trường header



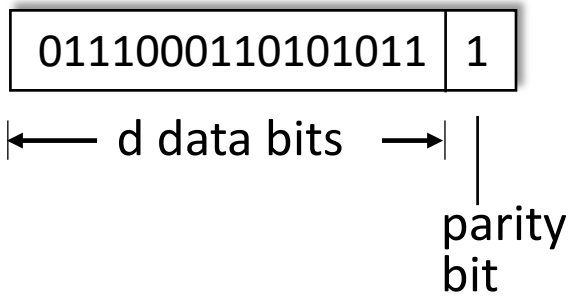
Phát hiện lỗi chưa chắc tin cậy 100%!

- Giao thức có thể nhầm
- Một số lỗi chưa không phải tất cả
- Trường EDC lớn hơn thì phát hiện và sửa lỗi tốt hơn

Kiểm tra chẵn lẻ (parity)

1 bit chẵn lẻ:

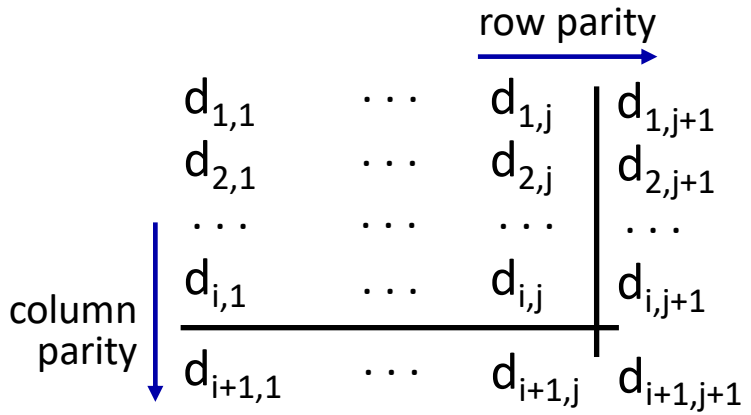
- Phát hiện 1 bit lỗi



Chẵn bit: tập bit chẵn lẻ
là một số chẵn các bit 1

Bit chẵn lẻ hai chiều:

- phát hiện và sửa 1 bit lỗi



no errors:

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
1	0	1	0	1	0

detected and correctable single-bit error:

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
1	0	1	0	1	0

parity error

parity error

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

Internet checksum (ôn lại)

Goal: phát hiện lỗi(*i.e.,* các bit bị lật) trong segment đã truyền

Người gửi:

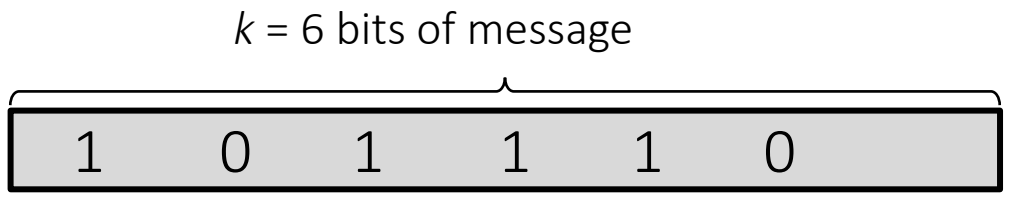
- Xem nội dung của UDP segment (bao gồm các trường của UDP và địa chỉ IP) như choỗi số nguyên 16 bit
- checksum: bù 1 của tổng
- checksum đặt vào trường UDP checksum

Người nhận:

- Tính toán checksum đã nhận
- Tính toán lại checksum có bằng với giá trị checksum trong trường checksum không:
  - Không bằng- phát hiện lỗi
  - Bằng – không phát hiện lõi, nhưng có lỗi.

Cyclic redundancy check (CRC)

- Biểu diễn thông điệp k bit bằng đa thức có bậc k-1
  - **Mỗi hệ số của đa thức tương ứng là 1 và 0:**



$$M(x) = 1x^5 + 0x^4 + 1x^3 + 1x^2 + 1x + 0$$

Toán học modulo-2

- Thêm và trừ là sử dụng xor bit

Ví dụ phép nhân

```
   1101
    110
   0000
  11010
 110100
 101110
```

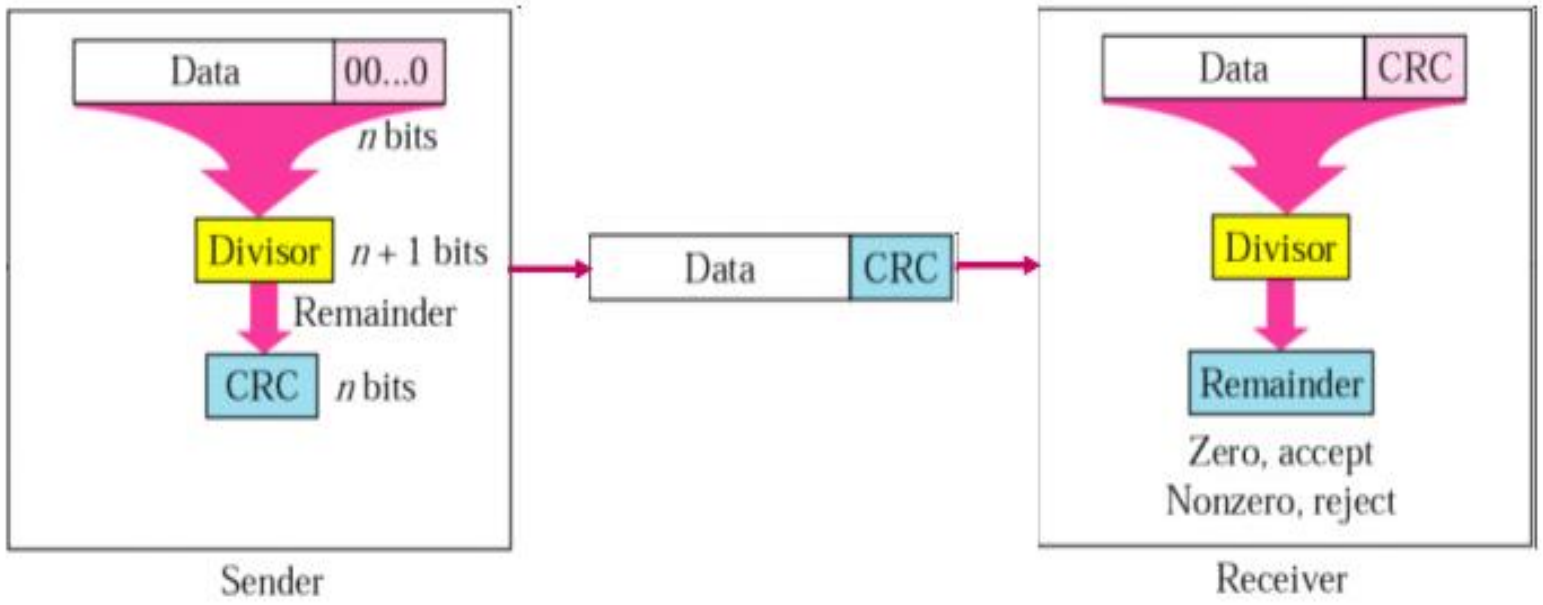
Ví dụ phép chia 101110 cho 110

```
      1101
110 | 101110
      110
       111
       110
        011
        000
         110
         110
```

Ví dụ

- Hãy thực hiện phép chia module 2:
- A) 11010110110000/10011
- B) 11100110/11001

CRC


Data
00...0
n bits
Divisor
n + 1 bits
Remainder
CRC
n bits
Sender
Data
CRC
Data
CRC
Divisor
Remainder
Zero, accept
Nonzero, reject
Receiver

CRC phía người gửi

- ***M*(x)** là thông điệp có chiều dài ***k***
  - ***e.g.:*** $M(x) = x^5 + x^3 + x^2 + x \;\; (k = 6)$

1 0 1 1 1 0

- Người gửi và nhận đồng ý một đa thức sinh ***G*(*x*)** với bậc g-1
  - ***e.g.:*** $G(x) = x^3 + 1 \;\; (g = 4)$

1 0 0 1

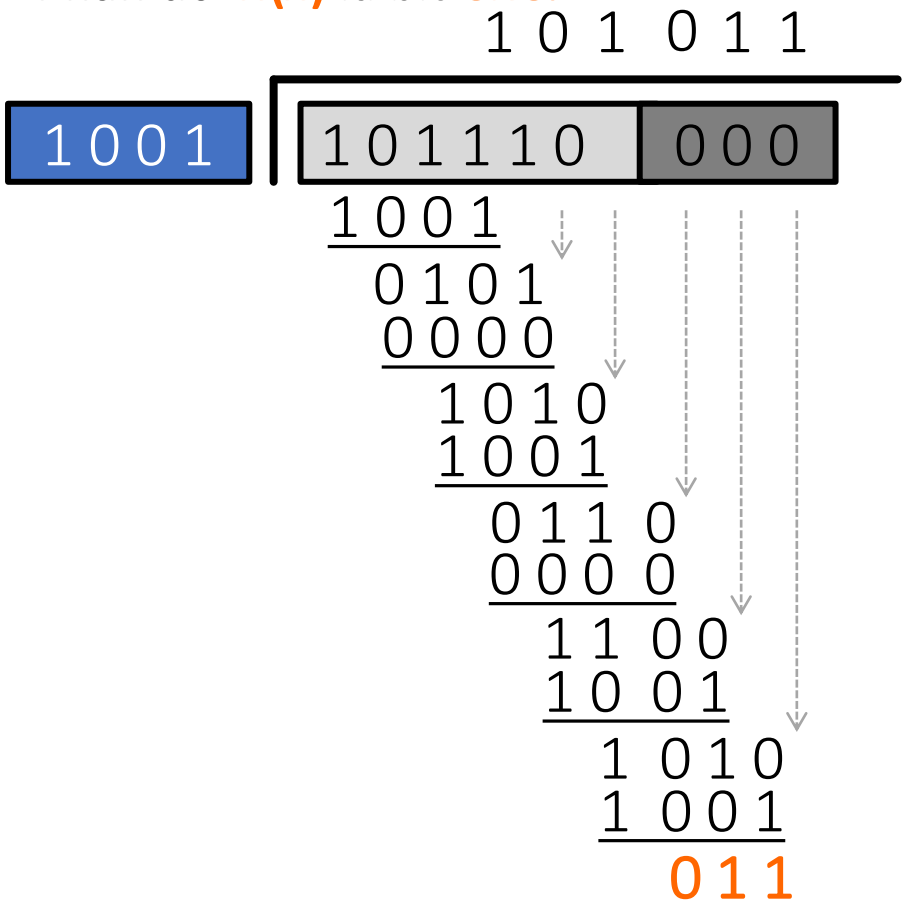
  - Bậc g xác định như thế nào?
  - Chính là số lượng bit thêm vào thông điệp

1. Tính toán thông điệp thêm vào thông điệp chính ***T*(*x*)** $= M(x) \cdot x^{g-1}$
   - *i.e.*, thêm vào bên phải với $g - 1$ bit 0
   - ***e.g.:*** $T(x) = M(x) \cdot x^3 = x^8 + x^6 + x^5 + x^4$

1 0 1 1 1 0 | 0 0 0

CRC tại người nhận

2. **Tách thông điệp phụ được thêm vào dùng đa thức sinh: chia *T*(*x*)** cho đa thức sinh ***G*(*x*)**
   - **Phần dư *R*(*x*)** là bit **CRC:**



$R(x) = x + 1$

CRC tại phía người nhận

3. **Người gửi truyền từ mã $C(x) = T(x) + R(x)$**
   - *Người gửi truyền thông điệp gốc với bit CRC thêm vào cuối*

   - Ví dụ, $C(x) = x^8 + x^6 + x^5 + x^4 + x + 1$

1 0 1 1 1 0	0 1 1

Bài tập

- Tạo từ mã CRC (7,4) cho bảng sau với ma trận sinh là g(x)=x3 + x + 1

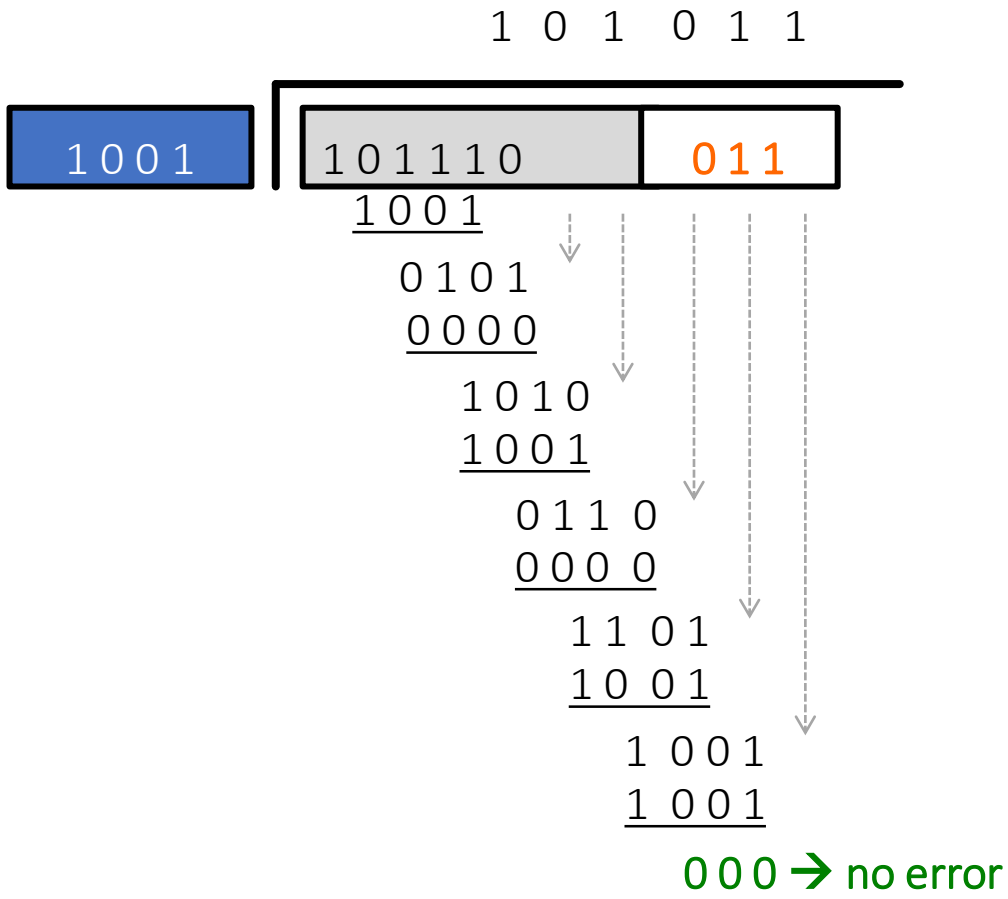
Dataword	*Codeword*	*Dataword*	*Codeword*
0000		1000	
0001		1001	
0010		1010	
0011		1011	
0100		1100	
0101		1101	
0110		1110	
0111		1111	

Thuộc tính CRC

- Nhớ: phần dư của phép chia [$T(x)/G(x)$] = $R(x)$
  - Nếu chia $C(x)$ / $G(x)$?
- $C(x) = T(x) + R(x)$ vì vậy phần dư
  - [$T(x)/G(x)$] = $R(x)$, **cộng**
  - [$R(x)/G(x)$] = $R(x)$
- Xor bit
  - [$C(x)/G(x)$] = $R(x) + R(x) = 0$

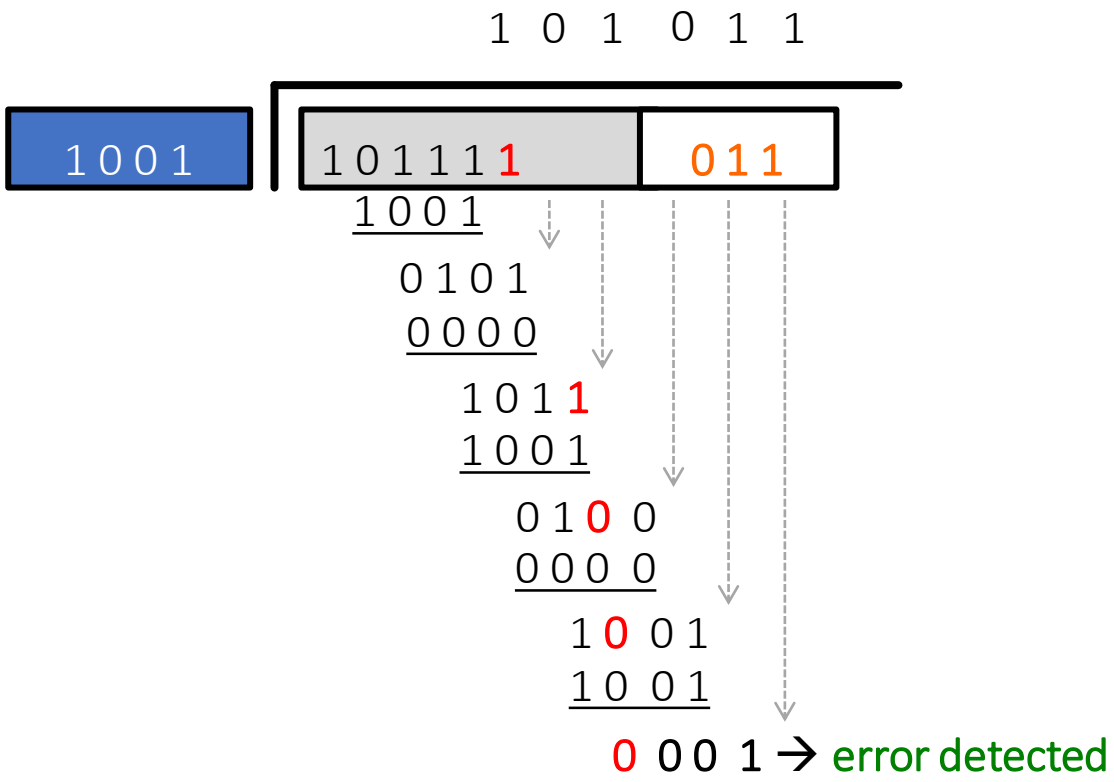
Phát hiện lỗi phía người nhận

- **Người nhận chia thông điệp nhận được *C'*(*x*) cho *G*(*x*)**
  - **Nếu không có lỗi phần dư sẽ là 0**



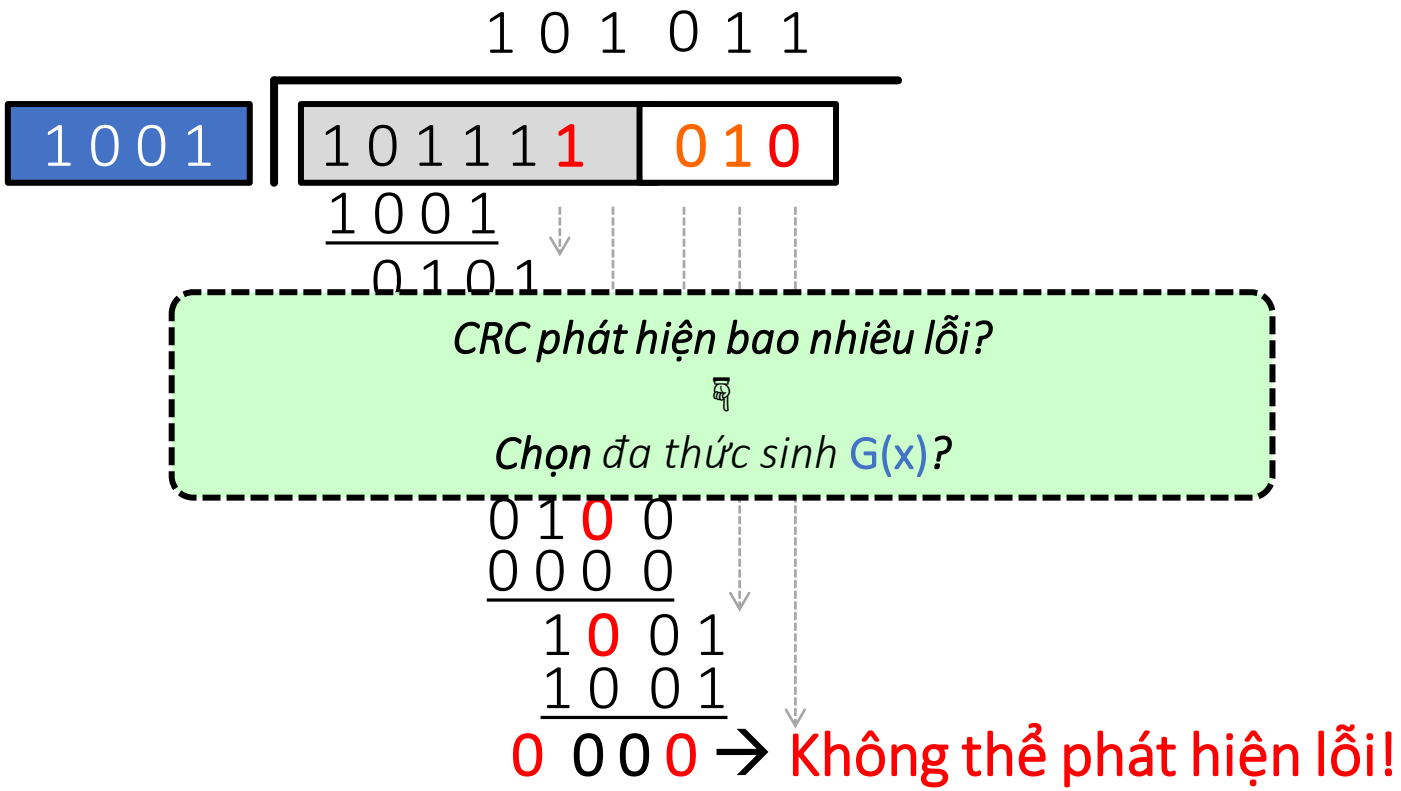
Phát hiện lỗi phía người nhận

- **Người nhận chia thông điệp nhận được *C'*(*x*) cho *G*(*x*)**
  - **Nếu không có lỗi phần dư sẽ có thể khác 0**



Phát hiện lỗi bên nhận

- **Người nhận chia thông điệp nhận được *C'*(*x*) cho *G*(*x*)**
  - **Nếu không có lỗi phần dư có thể khác 0**



Phát hiện lỗi với CRC

- *Đa thức lỗi* $E(x) = C(x) + C'(x)$ thể hiện sự khác nhau giữa C(x) và C'(x)
  - $E(x)$ cho biết **bit nào bị lật**
- Có thể viết $C'(x) = C(x) + E(x)$, vì vậy:
  - **Phần dư** $[C'(x) / G(x)]$ = **phần dư** $[E(x) / G(x)]$

- Khi nào lỗi không được phát hiện?
  - Khi phần dư$[E(x) / G(x)] = 0$

Bài tập

- Cho ma trận sinh là g(x)=x3 + x +1
- hãy kiểm tra xem trong hai trường hợp sau, từ mã nào là đúng, từ mã nào có lỗi? Giải thích tại sao?
  - 1001110
  - 1000110

Ứng dụng mã CRC

Code	Generator Polynomial g(X)	Appended Bits	Applications
CRC-4	$X^4 + X + 1$	4	ITU G.704
CRC-8	$X^8 + X^2 + X + 1$	8	ATM header
CRC-10	$X^{10} + X^9 + X^5 + X^4 + X + 1$	10	ATM AAL
CRC-16-CCITT	$X^{16} + X^{12} + X^5 + 1$	16	Bluetooth
CRC-32	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$	32	LANs

Bài tập

- Cho ma trận sinh là g(x)=x4+x+1 cho mã CRC (15,11)
- A) Tìm các mã CRC (15,11)
- B) khoảng cách tối thiểu là?
- C) có thể sửa được bao nhiêu lỗi trên một mã?
- D) có thể phát hiện được bao nhiêu lỗi
- E) Tạo ma trận sinh

Link layer, LANs: roadmap

- introduction
- error detection, correction
- Giao thức đa truy cập
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Giao thức, link đa truy cập

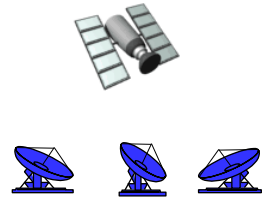
Hai kiểu link:

- point-to-point
  - point-to-point link giữa switch và host
  - PPP cho truy cập dial-up
- broadcast (có dây hoặc môi trường không dây)
  - Ethernet
  - upstream HFC trong mạng truy cập
  - 802.11 wireless LAN, 4G/4G. satellite

shared wire (e.g., cabled Ethernet)

shared radio: 4G/5G

shared radio: WiFi



shared radio: satellite

humans at a cocktail party (shared air, acoustical)

Giao thức đa truy cập

- Chia sẻ shared broadcast channel
- Hai hoặc nhiều node truyền đồng thời:
  - *collision* (đụng độ) nếu nêu hai hoặc nhiều tín hiệu đồng thời

multiple access protocol

- Giải thuật phân tán quyết định node truyền thế nào khi nào truyền
- Kết nối nào sử dụng kênh mà nó có

Ý tưởng giao thức đa truy cập

given: multiple access channel (MAC) tốc độ *R* bps

Ý tưởng:

1. Khi 1 node muôn truyền, nó có thể gửi với tốc độ R.
2. Khi M node muốn truyền, mỗi node có thể truyền ở tốc độ *R/M*
3. Phân cấp:
   - Không có sự hợp tác khi truyền
   - Không đồng bộ đồng hồ, slots
4. Đơn giản

Giao thức MAC: taxonomy

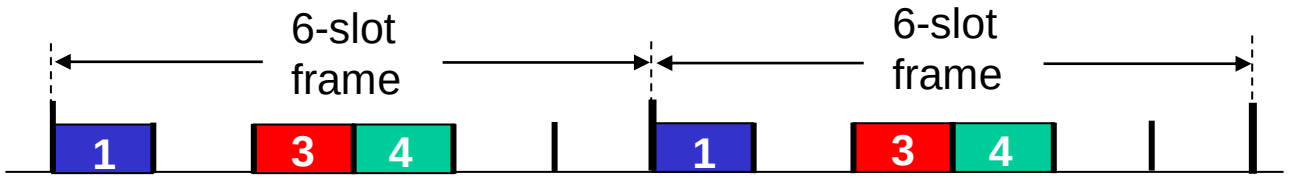
Ba lớp:

- Chia kênh
  - Chia kênh thành miếng nhỏ hơn (time slots, frequency, code)
  - Dành mỗi miếng cho mỗi node
- *random access*
  - channel không chia cho phép đụng độ
  - Khôi phục từ đụng độ
- Thay phiên nhau
  - Các node thay phiên nhau truyền, nhưng node truyền nhiều có thể lâu thay phiên hơn

Channel partitioning MAC protocols: TDMA (chia kênh)

TDMA: time division multiple access

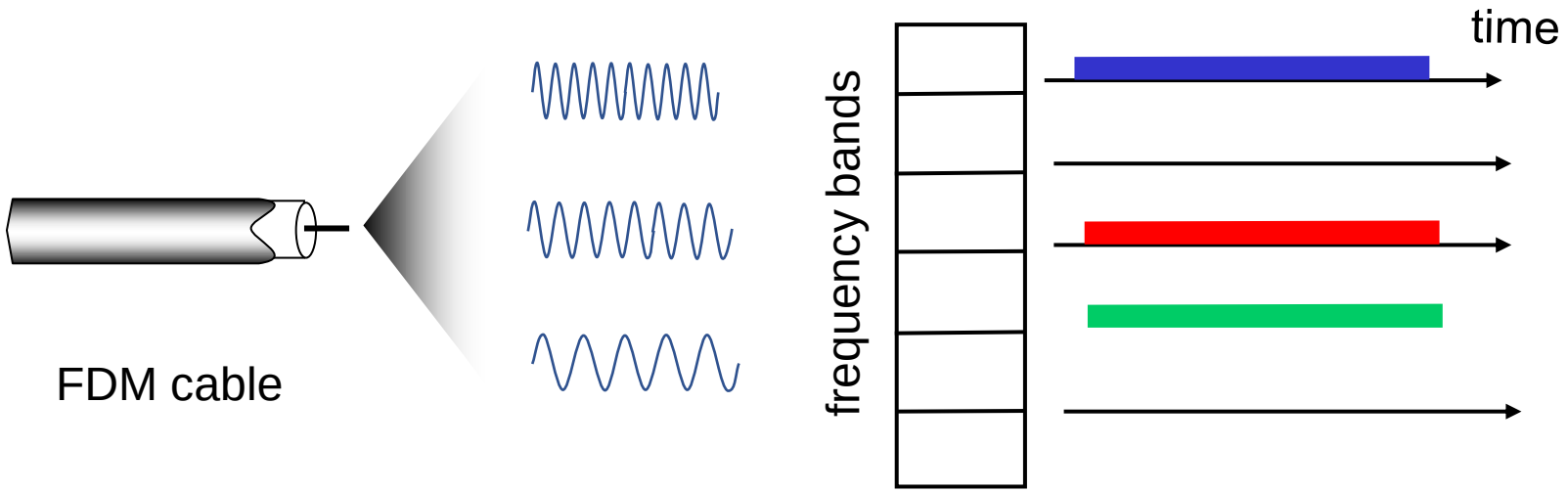
- Truy cập đến kênh theo vòng
- Mỗi trạm lấy slot dài cố định (chiều dài=thời gian truyền gói) trong mỗi vòng
- Slot không dùng thì sẽ vào trạng thái nghỉ - idle
- Ví dụ: 6-station LAN, 1,3,4 có gói truyền, slots 2,5,6 idle



Channel partitioning MAC protocols: FDMA

FDMA: frequency division multiple access

- channel chia thành dải tần
- Mỗi trạm gán dải tần cố định
- Thời gian không truyền tại băng tần nào thì băng tần đó ở trạng thái nghỉ -idle
- Ví dụ: 6-station LAN, 1,3,4 có gói truyền, băng tần 2,5,6 idle



Giao thức truy cập ngẫu nhiên

- Khi node có gói truyền
  - Truyền trên cả kênh với tốc độ R.
  - Không chia sẻ với node khác
- Hai hoặc nhiều node truyền: “collision”
- random access MAC protocol specifies:
  - Làm sao phát hiện đụng độ
  - Làm sao khôi phục sau đụng độ(e.g., via delayed retransmissions)
- Ví dụ giao thức truy cập ngẫu nhiên:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

Biết rằng:

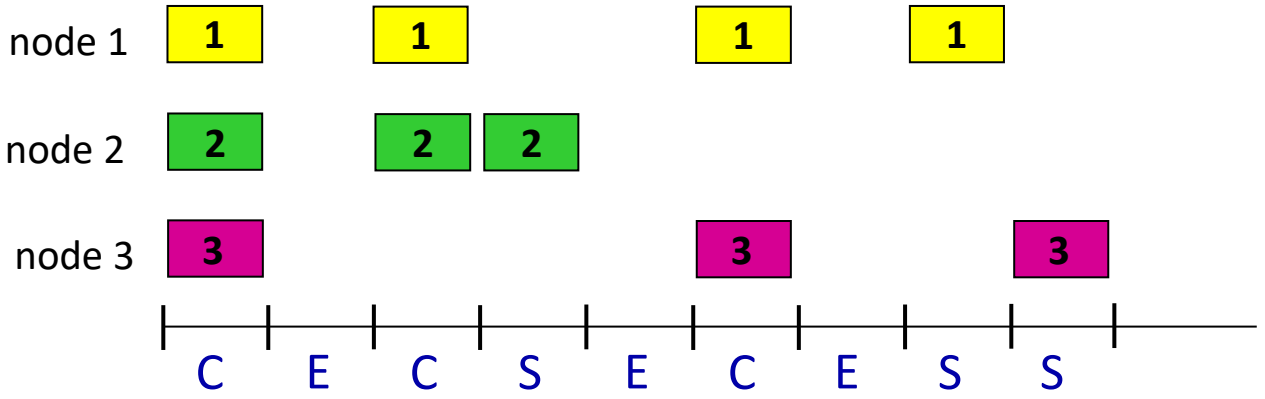
- Mọi frame cùng kích thước
- Thời gian chia thành các slot bằng nhau (thời gian để truyền 1 frame)
- Nodes bắt đầu bắt đầu truyền chỉ khi slot bắt đầu
- nodes là đồng bộ
- Nêu 2 hoặc nhiều node truyền trong slot, tất cả các node phát hiện đụng độ

Xử lý:

- Khi node chứa frame mới, truyền trong slot tiếp theo
  - *if no collision:* node có thể gửi trong slot kế tiếp
  - *if collision:* node truyền lại frame trong mỗi slot tiếp theo với xác suất p tới khi thành công.

randomization – *why*?

Slotted ALOHA



C: collision
S: success
E: empty

ưu:

- Một node hoạt động có thể truyền liên tục tại tốc độ kênh truyền
- Phân cấp cao: chỉ các slot trong các node cần mới đồng bộ
- Đơn giản

Cons:

- collisions, đợi slot
- idle slots
- nodes có thể phát hiện đụng độ trong thời gian nhỏ hơn thời gian truyền gói
- Đồng bộ đồng hồ

CSMA (carrier sense multiple access)

CSMA: nghe trước khi truyền:

- if kênh rảnh: truyền toàn bộ frame
- if kênh bận: trì hoãn truyền

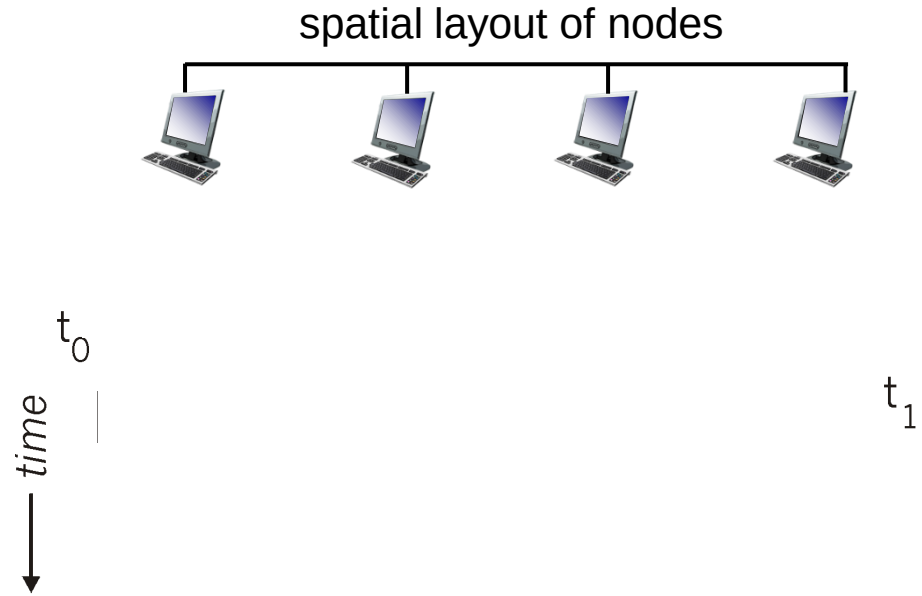
- Giống như người: không ngắt ai

CSMA/CD: CSMA có *collision detection (phát hiện đụng độ)*

- collisions *được phát hiện trong thời gian ngắn*
- Truyền khi xung đột sẽ bị hủy bỏ nên tránh lãng phí kênh
- Có dây phát hiện dễ, không dây phát hiện đụng độ khó

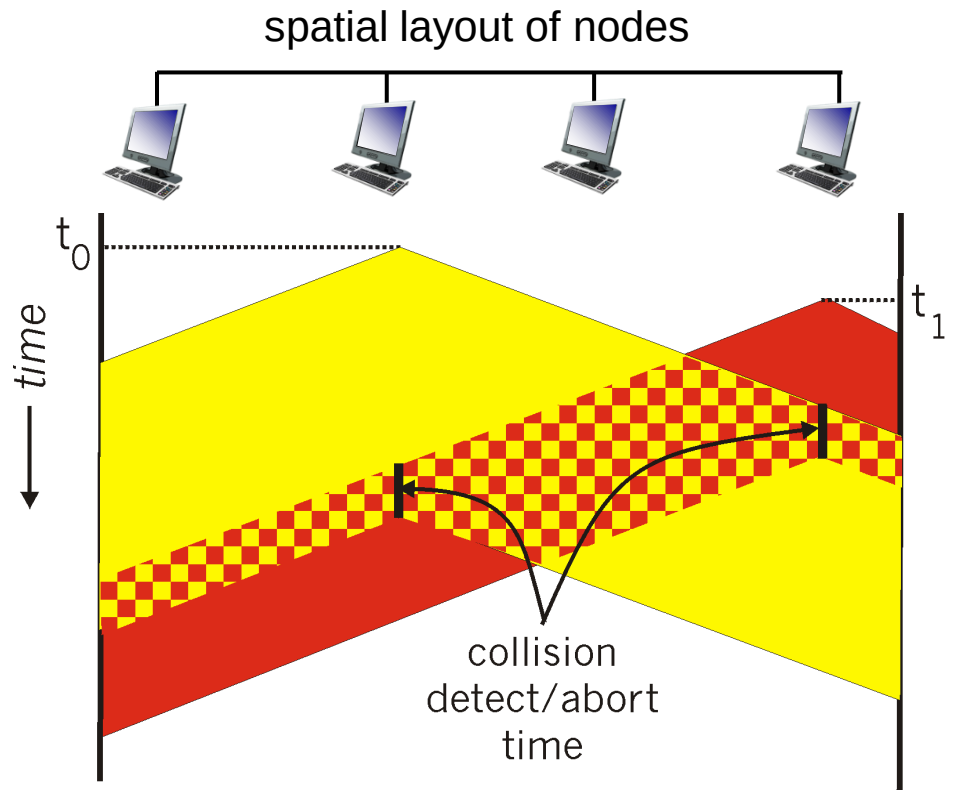
CSMA: đụng độ

- collisions *có thể vẫn xảy ra ở chế độ lắng nghe*:
  - Trễ lan truyền làm cho hai node có thể không nghe thấy nhau khi bắt đầu truyền
- collision: toàn bộ gói truyền đi bị lãng phí
  - Khoảng cách và độ trễ lan truyền đóng vai trò quan trọng trong quyết định có hay không có đụng độ



CSMA/CD:

- CSMA/CS giảm thời gian lãng phí khi đụng độ
  - Không được truyền khi phát hiện đụng độ



Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel:

   if idle: start frame transmission.

   if busy: wait until channel idle, then transmit
3. If NIC transmits entire frame without collision, NIC is done with frame !
4. If NIC detects another transmission while sending: abort, send jam signal
5. After aborting, NIC enters *binary (exponential) backoff:*
   - after *n*th collision, NIC chooses *K* at random from $\{0,1,2, \ldots, 2^n-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
   - more collisions: longer backoff interval

Hiệu quả CSMA/CD

- T_{prop} = max độ trễ lan truyền giữa 2 nodes trong LAN
- t_{trans} = time để truyền frame có kích thước tối đa

$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- efficiency tiến tới 1
  - khi t_{prop} tiến tới 0
  - Khi t_{trans} tiến tới vô hạn
- Hiệu năng tốt hơn ALOHA: đơn giản, rẻ, phi tập trung!

“Taking turns” MAC protocols (thay phiên)

channel partitioning MAC protocols:

- Chia sẻ kênh hiệu quả và công bằng khi tải cao
- Không hiệu quả khi tải thấp: trễ trong kênh chia sẻ, 1/N bandwidth được cấp phát thậm chí chỉ khi có 1 node hoạt động!

random access MAC protocols

- Hiệu quả khi tải thấp: một node có thể dùng toàn kênh
- Tải cao: đụng độ

“taking turns” protocols

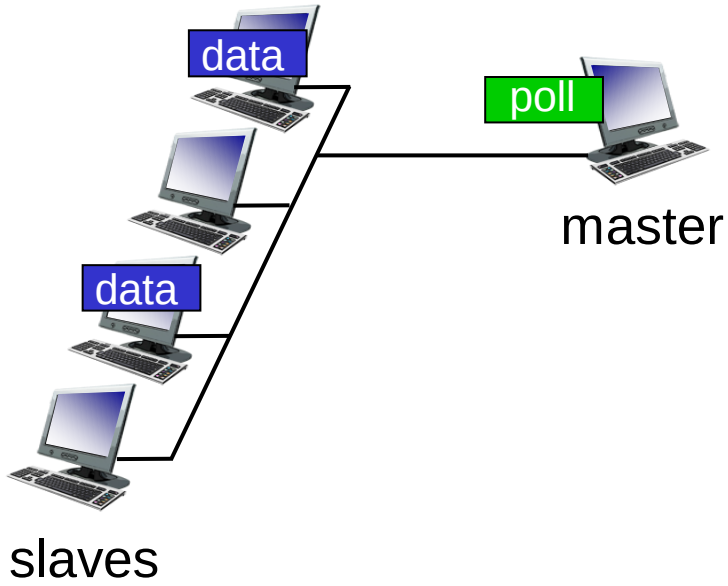
- Có vẻ tốt hơn cả

“Taking turns” MAC protocols

Giám sát:

- master node mời nodes khác truyền thay phiên

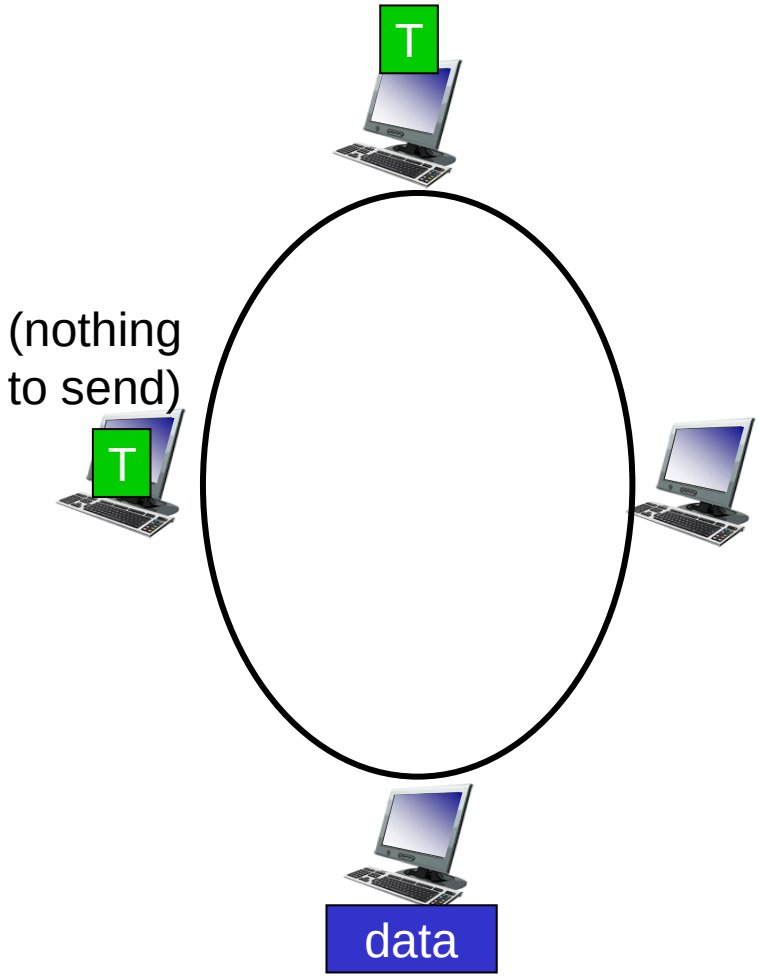
- Cân nhắc:
  - Giám sát quá tải
  - Trễ
  - single point of failure (master)



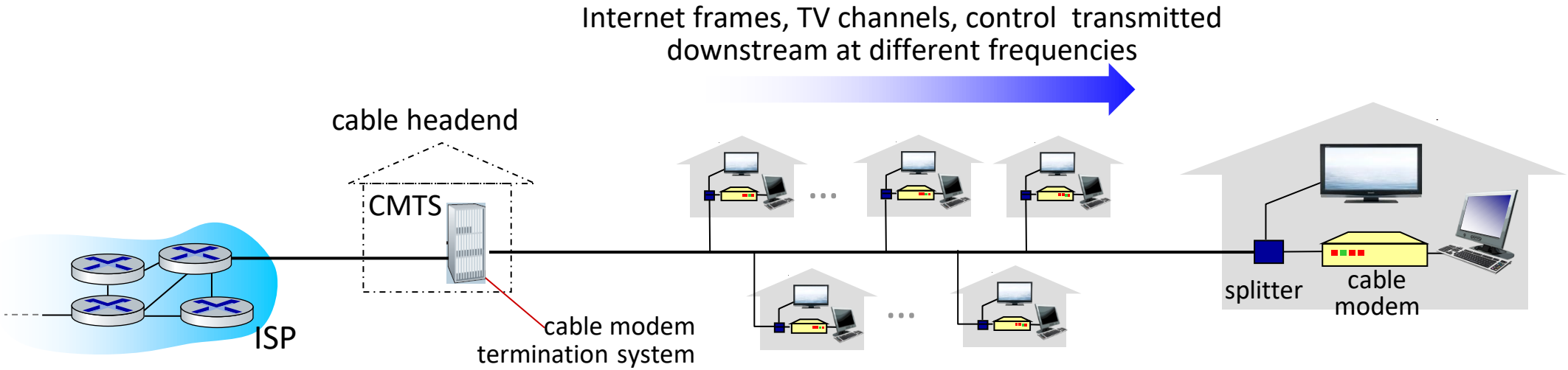
“Taking turns” MAC protocols

Truyền token:

- Kiểm soát token các node tuần tự.
- token message
- concerns:
  - token quá tải
  - Trễ
  - single point of failure (token)

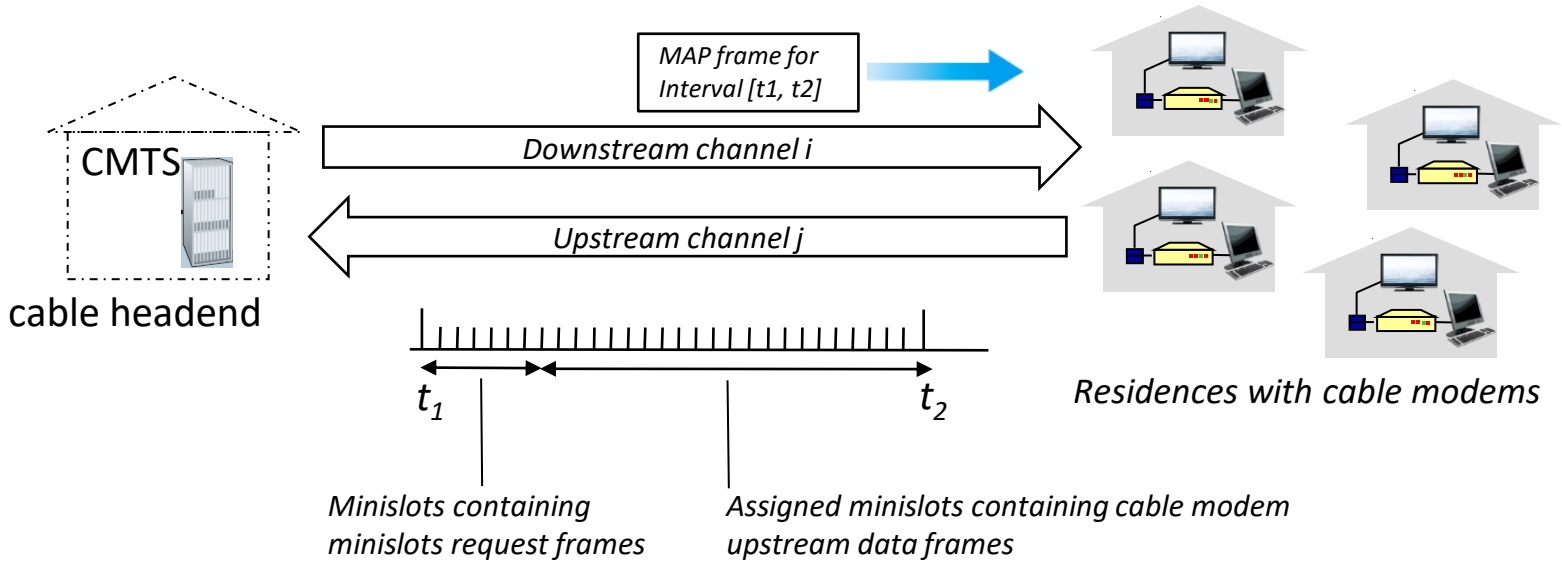


Mạng dùng cáp try cập: FDM, TDM *and* random access!



- Đa lưu lượng tải xuống(broadcast) FDM channels: up to 1.6 Gbps/channel
  - Một CMTS truyền trên kênh
- Kênh tải lên (up to 1 Gbps/channel)
  - multiple access: tất cả lưu lượng truyền của người dùng gán theo kênh thời gian, còn lại gán TDM

Mạng dùng cáp truy cập:



DOCSIS: (data over cable service interface specification) dùng:

- FDM upstream, downstream frequency channels
- TDM upstream: một vài slots được gán, thỉnh thoảng có nội dung contention
  - downstream MAP frame: để gán slot cho upstream
  - Yêu cầu slot cho upstream và truyền dữ liệu được thực hiện ngẫu nhiên theo kiểu giải thuật binary backoff

Giao thức MAC

- Kênh chẵn lẻ, theo thời gian, theo tần số
  - Time Division, Frequency Division
- Truy cập ngẫu nhiên (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - Cảm nhận truy cập: dễ trong một số công nghệ (có dây), khó trong một số công nghệ (không dây)
  - CSMA/CD được dùng trong Ethernet
  - CSMA/CA được dùng trong trong 802.11
- taking turns
  - Giám sát tại site trung tâm, truyền token
  - Bluetooth, FDDI, token ring

Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Địa chỉ MAC (nhớ phân biệt với giao thức MAC)

- Địa chỉ IP 32 bit:
  - *Cho các interface tại tầng mạng*
  - e.g.: 128.119.40.136

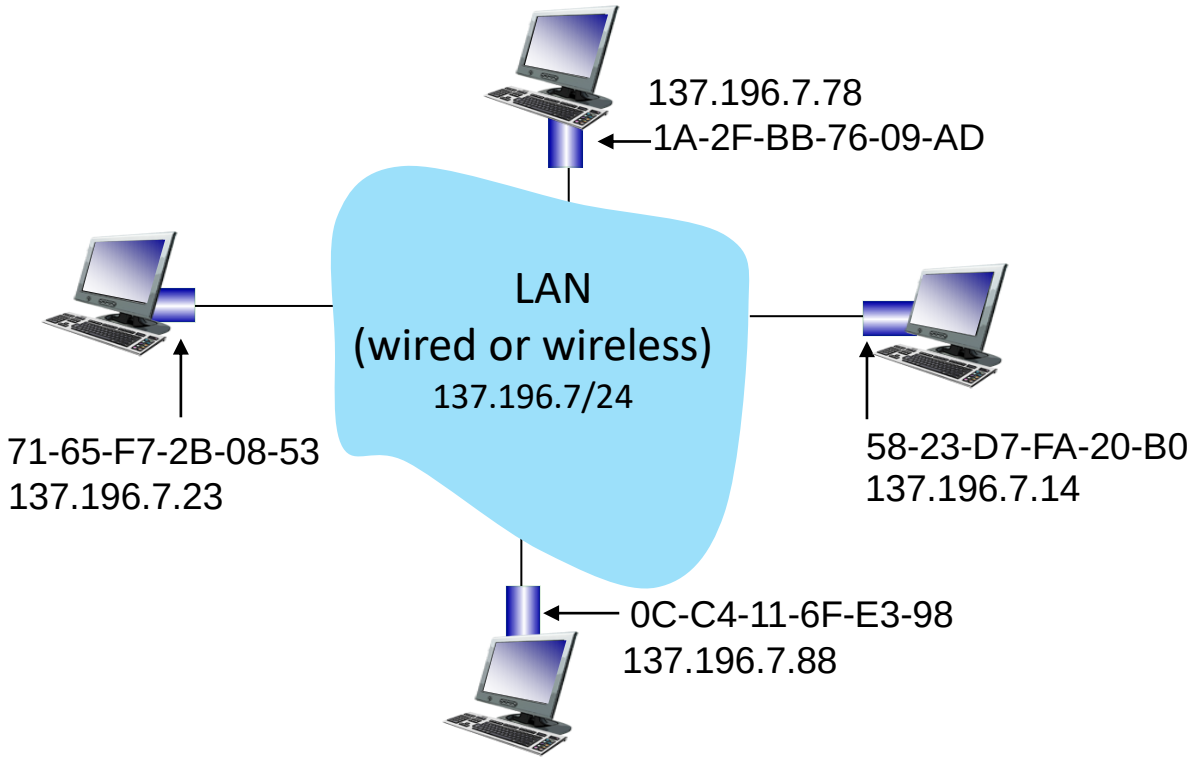
- MAC (or LAN or physical or Ethernet) address (địa chỉ vật lý):
  - Chức năng: được dùng để lấy frame từ 1 interface đến một interface kết nối trực tiếp (cùng subnet, trong dải địa chỉ ip)
  - 48-bit MAC address (for most LANs) gắn trong ROM của card mạng (NIC ROM), hoặc dùng phần mềm
  - e.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "numeral" represents 4 bits)

MAC addresses

Mỗi interface

- Có duy nhất 48-bit MAC address
- Có duy nhất 32-bit IP address (as we've seen)

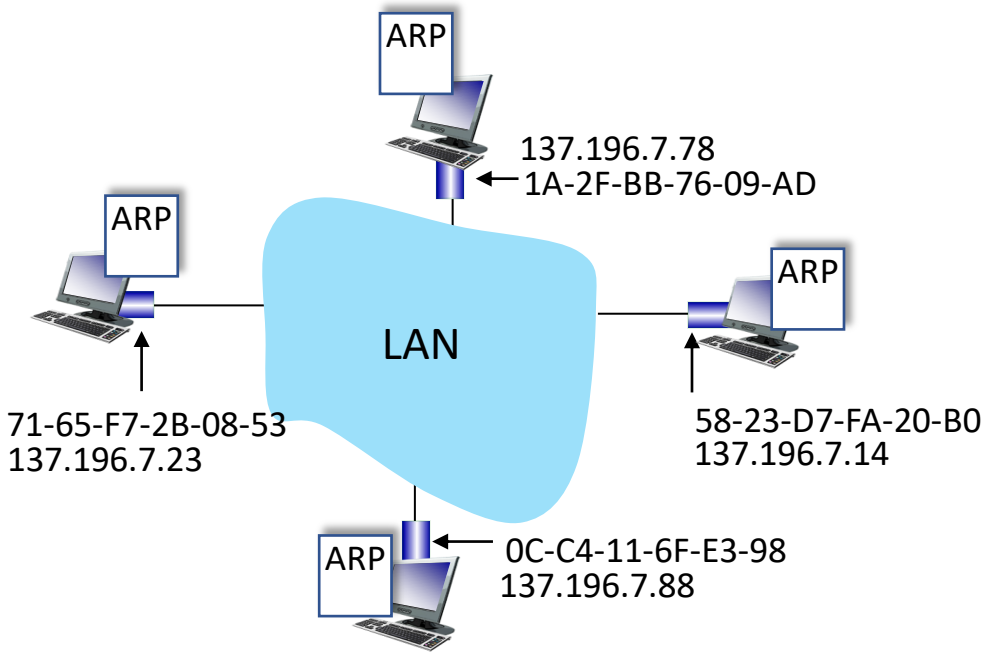


MAC addresses

- MAC được quản trị bởi IEEE
- Nhà sản xuất mua không gian địa chỉ MAC (đảm bảo duy nhất)
- Tương tự:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address:di động
  - Có thể di chuyển interface từ LAN này tới LAN khác
  - IP không di động: phụ thuộc vào mạng con mà interface được gán vào

ARP: address resolution protocol

Question: làm thế nào biết MAC, nếu biết IP?



ARP table: mỗi node (host, router) có một bảng MAC

- IP/MAC được khớp:

  < IP address; MAC address; TTL>
- TTL (Time To Live): thời gian khớp địa chỉ MAC với IP (thường 20 phút)

ARP

example: A muốn gửi datagram tới B

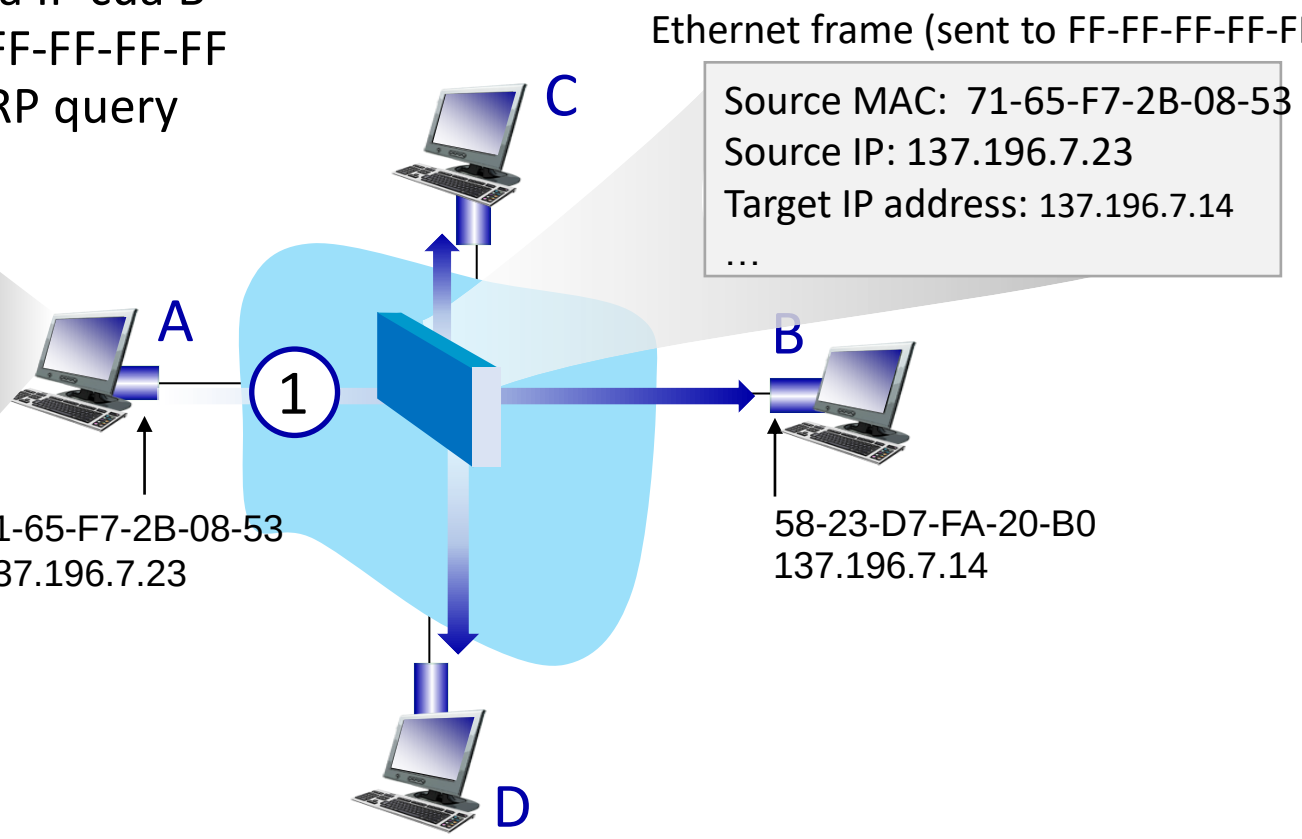
- MAC của B không có trong bảng ARP của A, A dùng ARP để tìm MAC B.

A gửi quảng bá truy vấn ARP , chứa IP của B

1
- MAC đích là quảng bá= FF-FF-FF-FF-FF-FF
- Tất cả node trong Lan nhận ARP query

ARP table in A

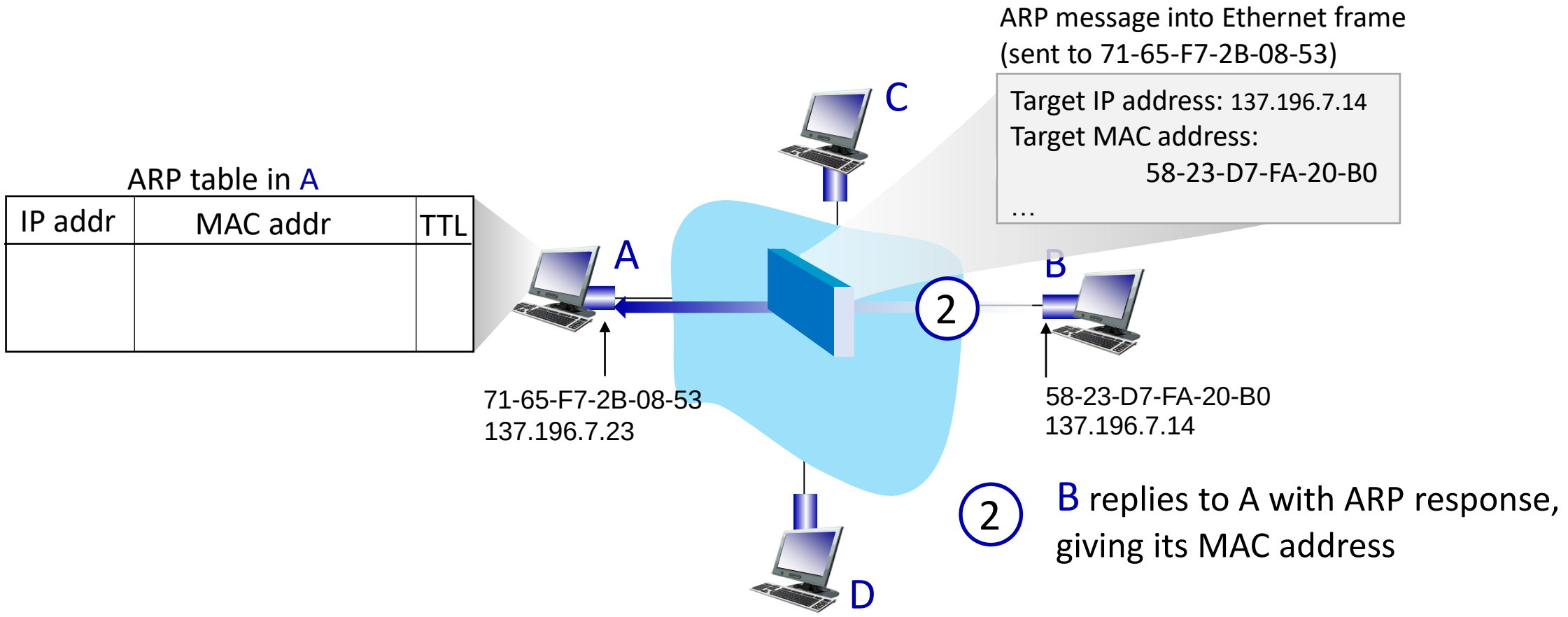
IP addr	MAC addr	TTL



ARP

example: A muốn gửi datagram tới B

- MAC của B không có trong bảng ARP của A, A dùng ARP để tìm MAC B.



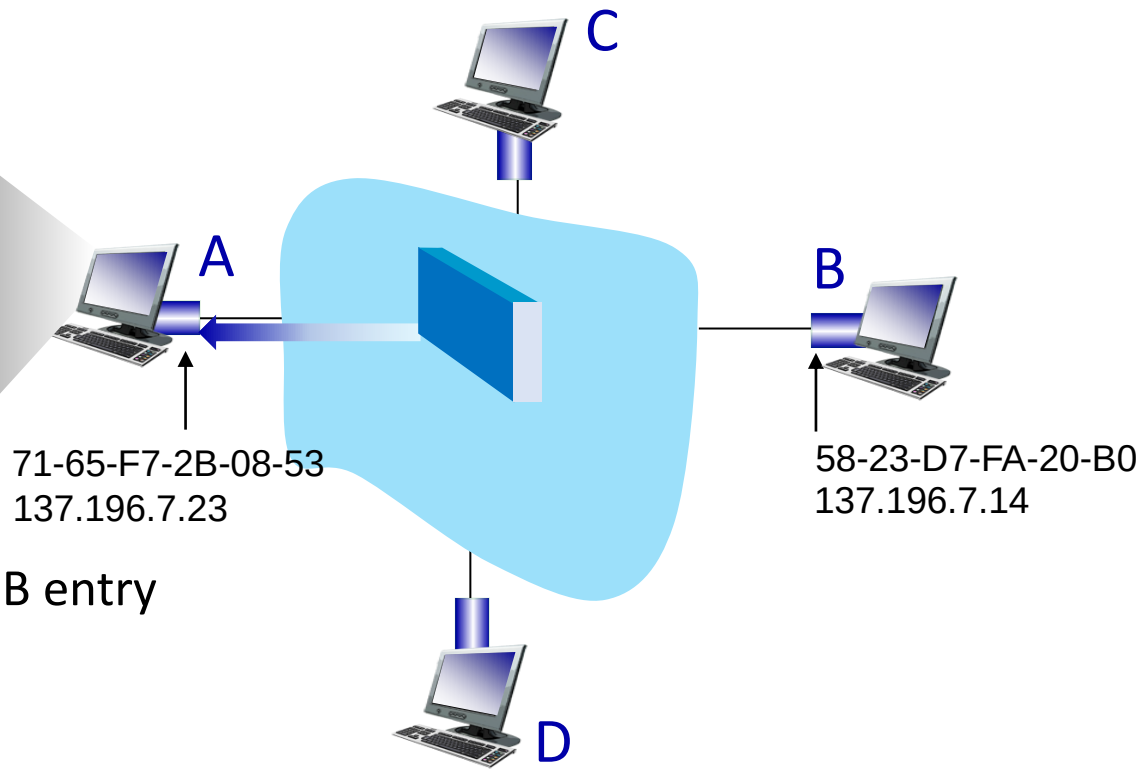
ARP

Ví dụ: A muốn gửi datagram tới B

- MAC của B không có trong bảng ARP của A, A dùng ARP để tìm MAC B.

ARP table in A

IP addr	MAC addr	TTL
137.196.7.14	58-23-D7-FA-20-B0	500

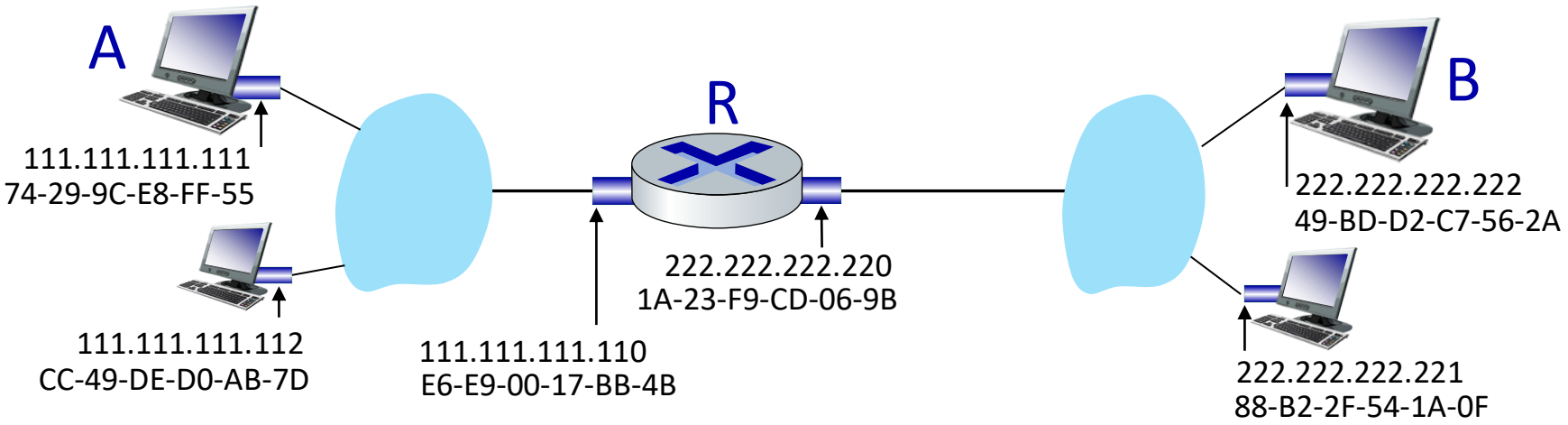


③ A receives B's reply, adds B entry into its local ARP table

Định tuyến tới một mạng khác: addressing

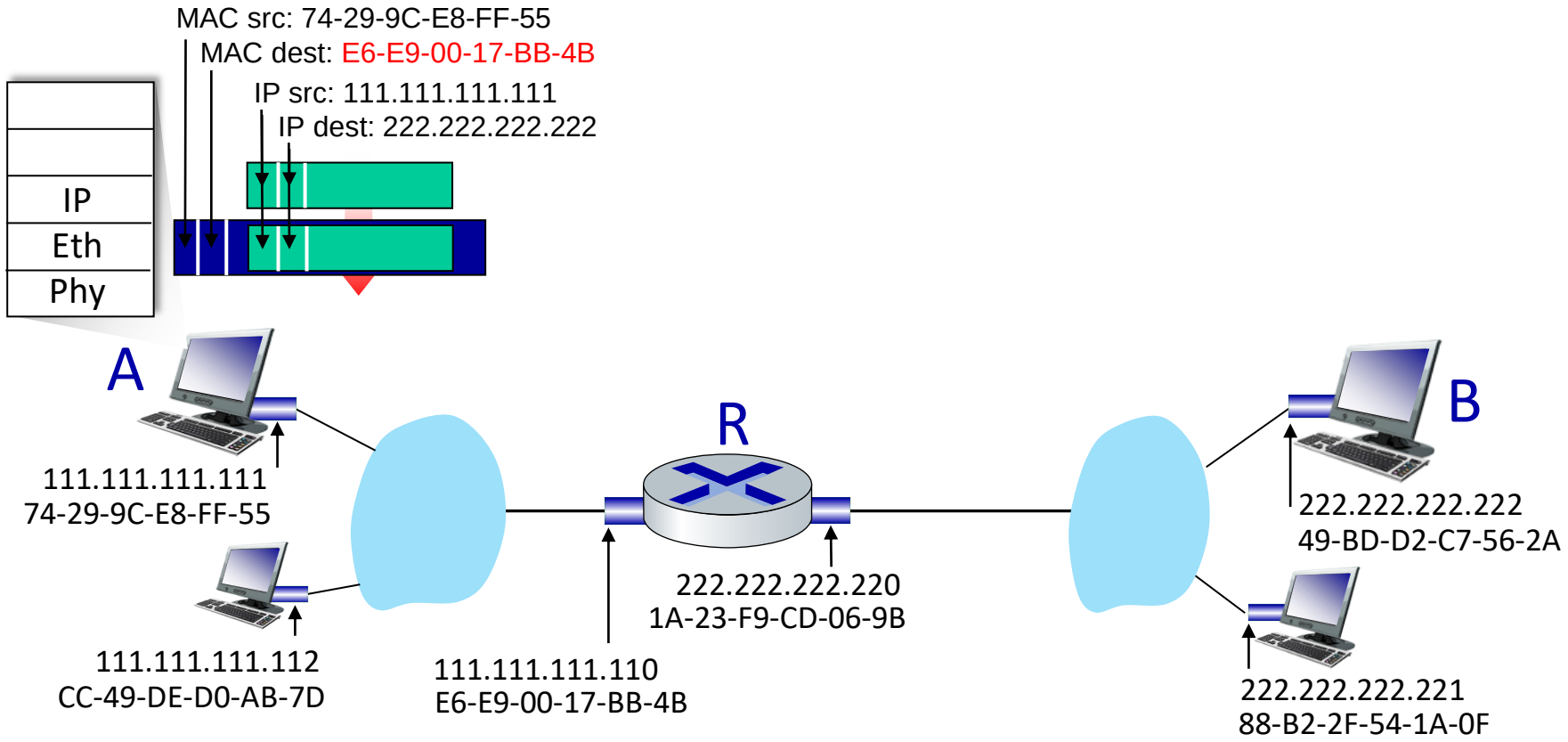
walkthrough: gửi datagram từ A tới B qua R

- Theo địa chỉ– at IP (datagram) and MAC layer (frame) levels
- Biết:
  - A biết IP của B
  - A biết gateway trên R (how?)
  - A biết MAC của R (how?)



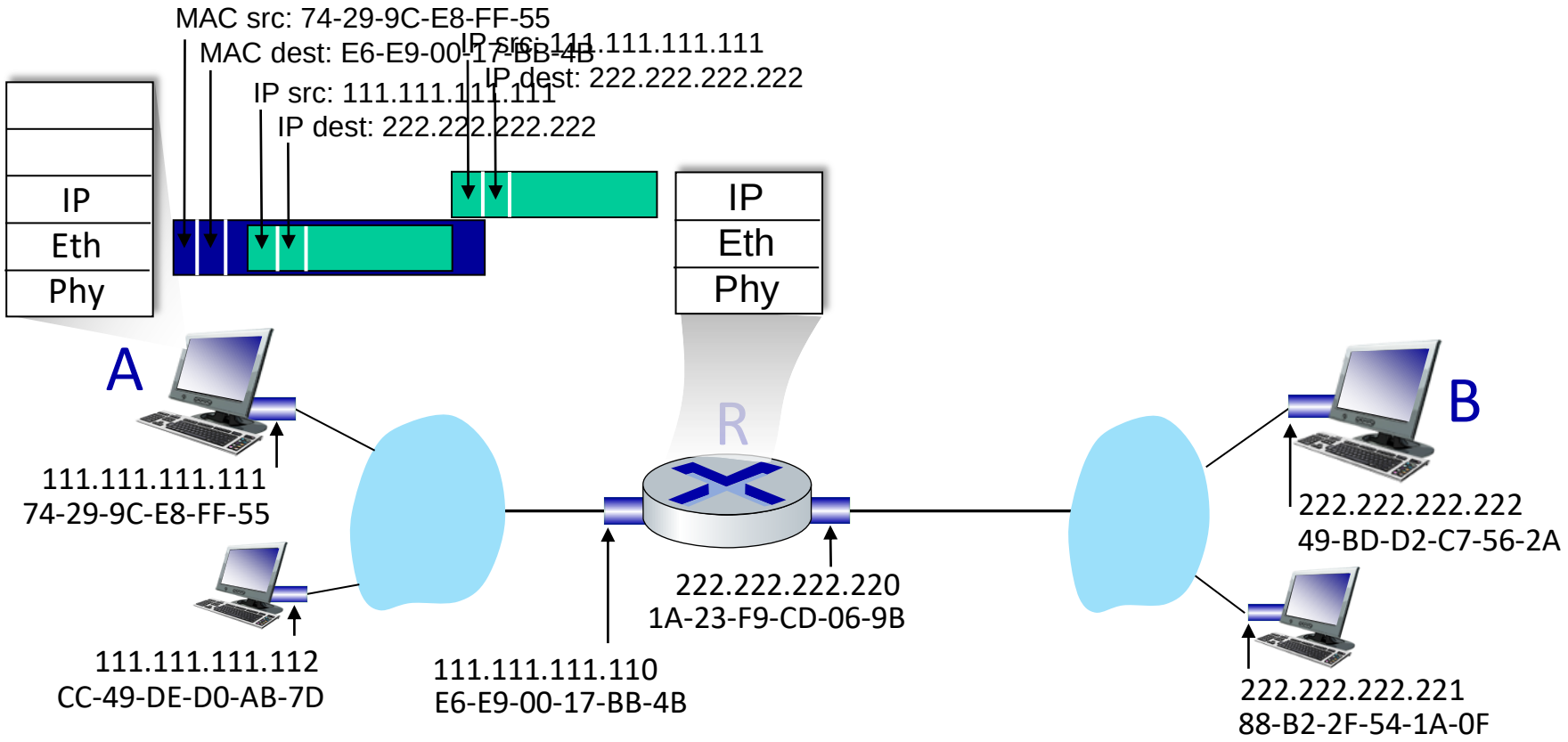
Định tuyến tới một mạng khác: addressing

- A tạo gói IP datagram với IP source A, destination B
- A tạo link-layer frame chứa A-to-B IP datagram
  - MAC của R là địa chỉ đích của frame



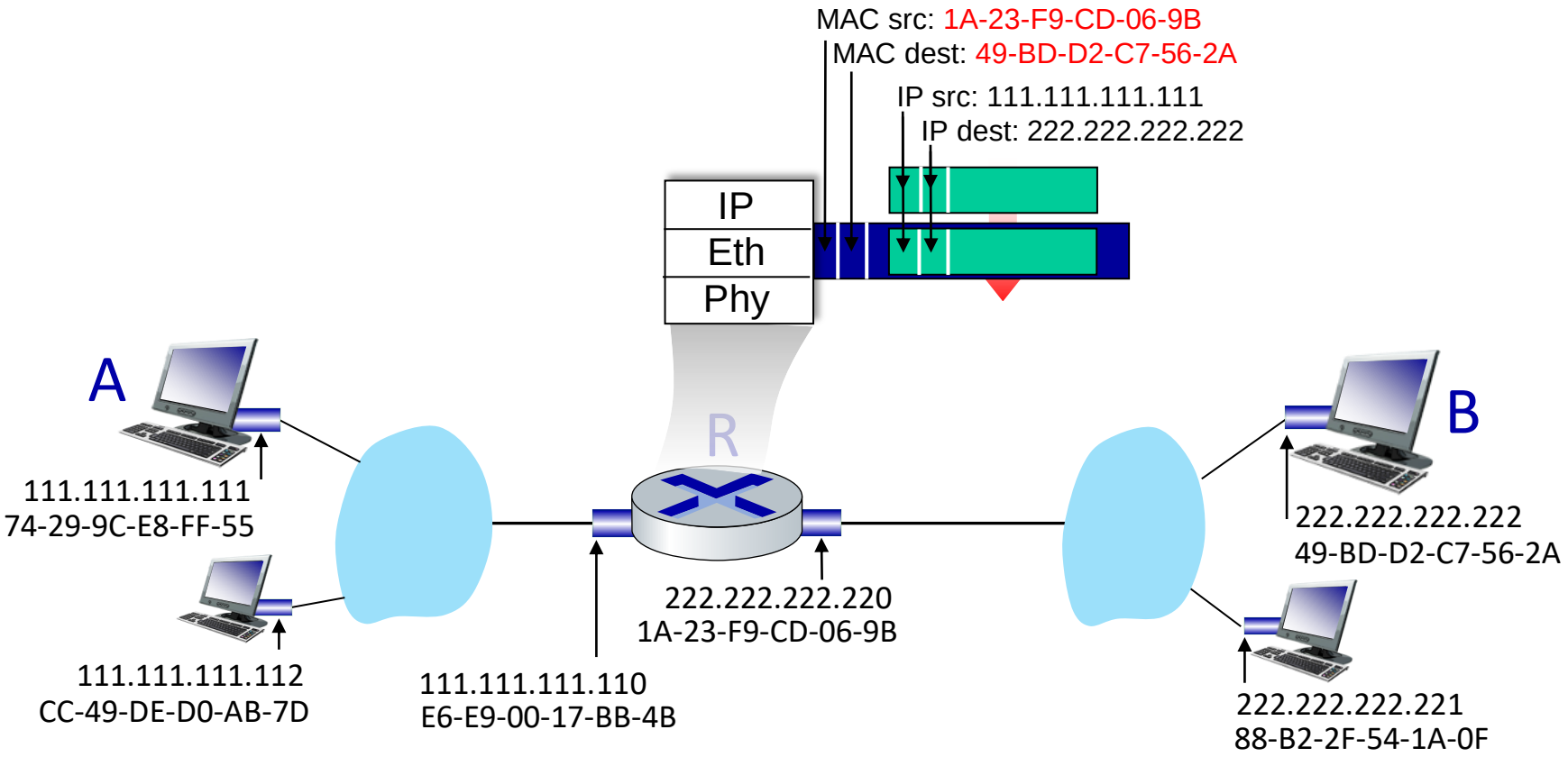
Định tuyến tới một mạng khác: addressing

- frame gửi từ A đến R
- frame nhận tại R, datagram được gỡ ra gửi tới IP



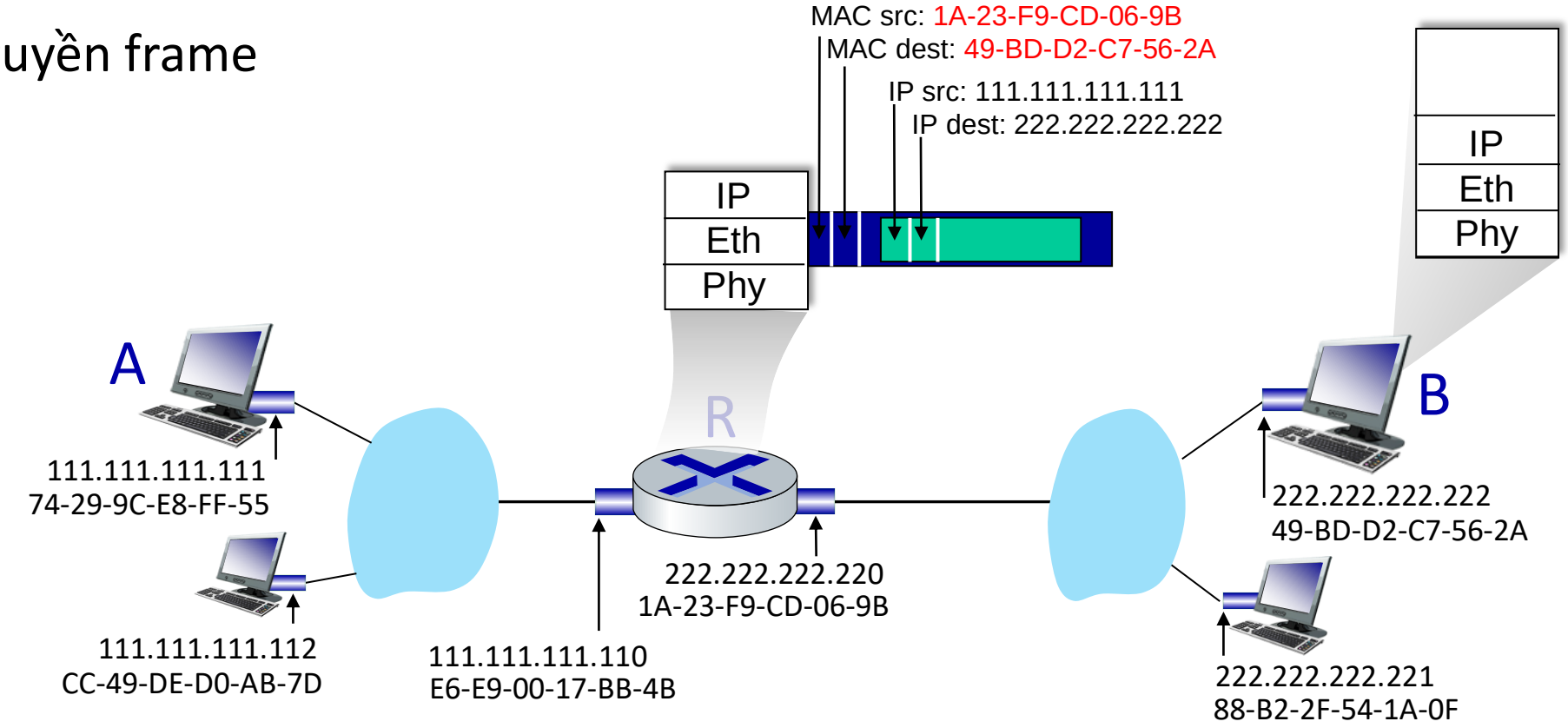
Định tuyến tới một mạng khác: addressing

- R chỉ interface đi ra, gửi datagram với IP source A, destination B đến link layer
- R tạo link-layer frame chứa A-đến-B IP datagram. Địa chỉ frame đích: địa chỉ MAC của B



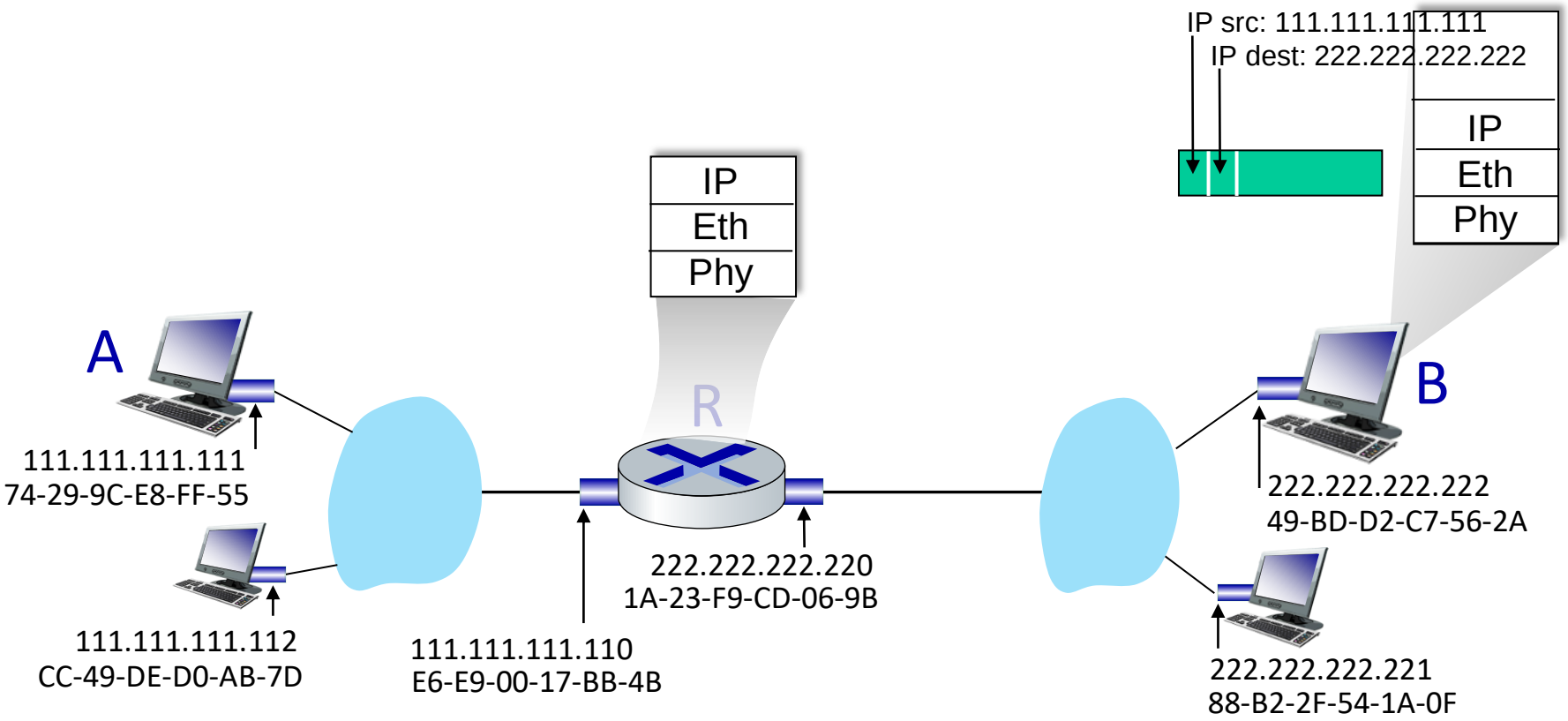
Định tuyến tới một mạng khác: addressing

- R chỉ interface đi ra, gửi datagram với IP source A, destination B đến link layer
- R tạo link-layer frame chứa A-đến-B IP datagram. Địa chỉ frame đích: địa chỉ MAC của B
- Truyền frame



Định tuyến tới một mạng khác: addressing

- B nhận frame, trích IP datagram địa chỉ đích B
- B gửi datagrame lên tầng trên đến IP



Link layer, LANs: roadmap

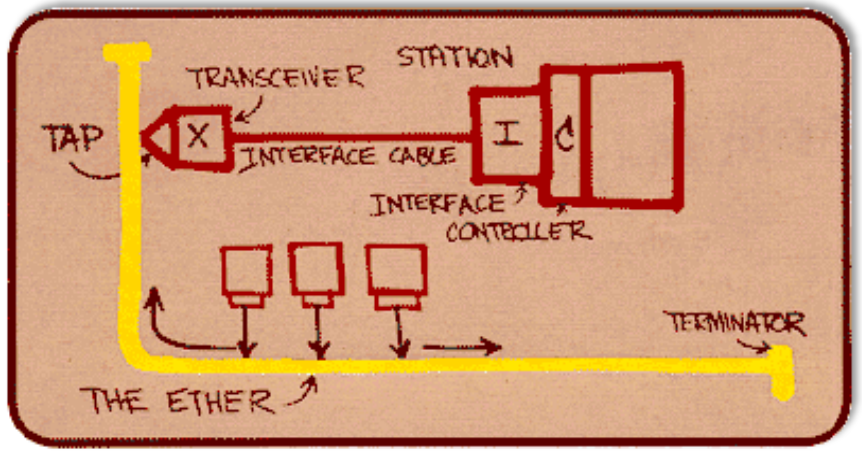
- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Ethernet

Công nghệ mạng có dây:

- Công nghệ Lan được dùng rộng rãi
- Đơn gian, rẻ
- Tốc độ: 10 Mbps – 400 Gbps
- single chip, multiple speeds (e.g., Broadcom BCM5761)



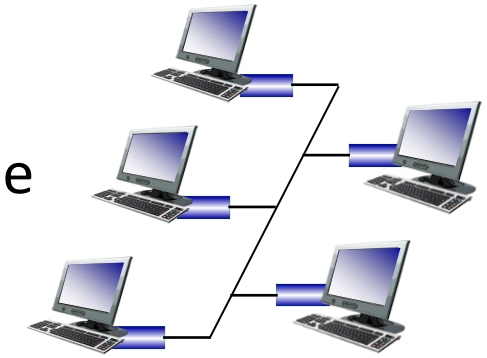
Metcalfe's Ethernet sketch

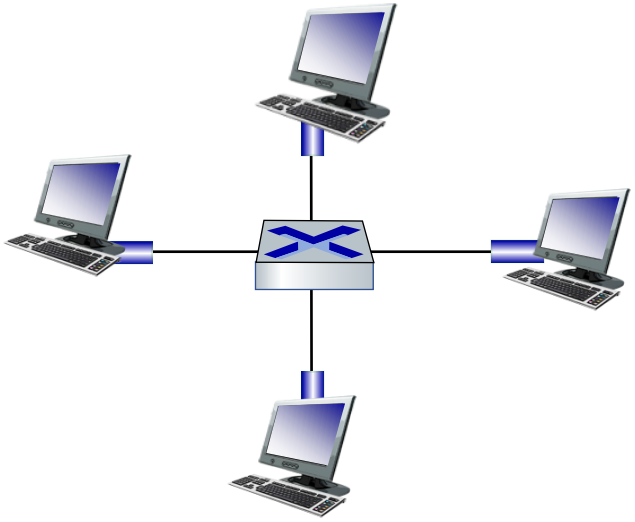
https://www.uspto.gov/learning-and-resources/journeys-innovation/audio-stories/defying-doubters

Ethernet: topo vật lý

- bus: thập kỉ 90
  - Các node cùng miền đụng độ
- switched: hiện tại ngày nay
  - Switch trung tâm
  - Mỗi node chạy riêng không đụng độ node khác

bus: coaxial cable





switched

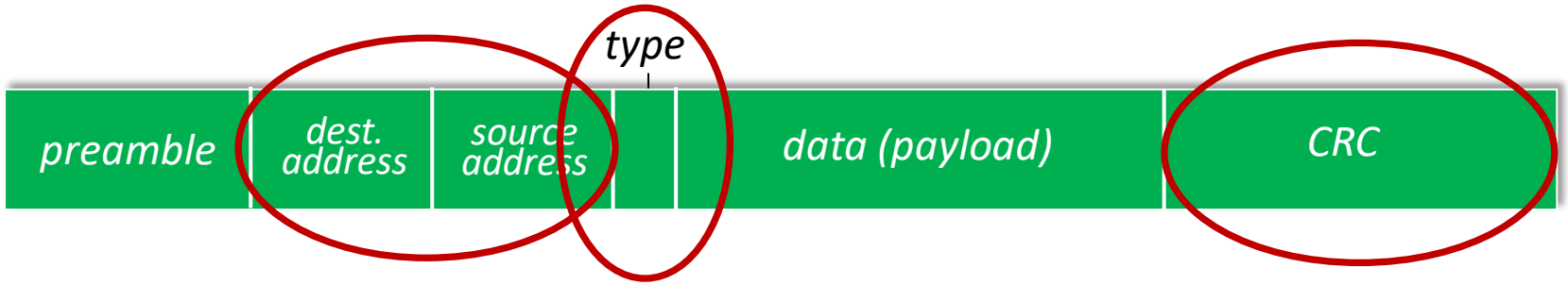
Cấu trúc frame Ethernet

Gửi frame đóng gói IP datagram (hoặc giao thức tầng mạng) trong frame Ethernet

preamble:

- được dùng để đồng bộ giữa người nhận và người gửi
- 7 bytes đầu tiên có giá trị 10101010 dùng để đánh thức bên nhận, byte cuối cùng là 10101011

Cấu trúc frame Ethernet(more)



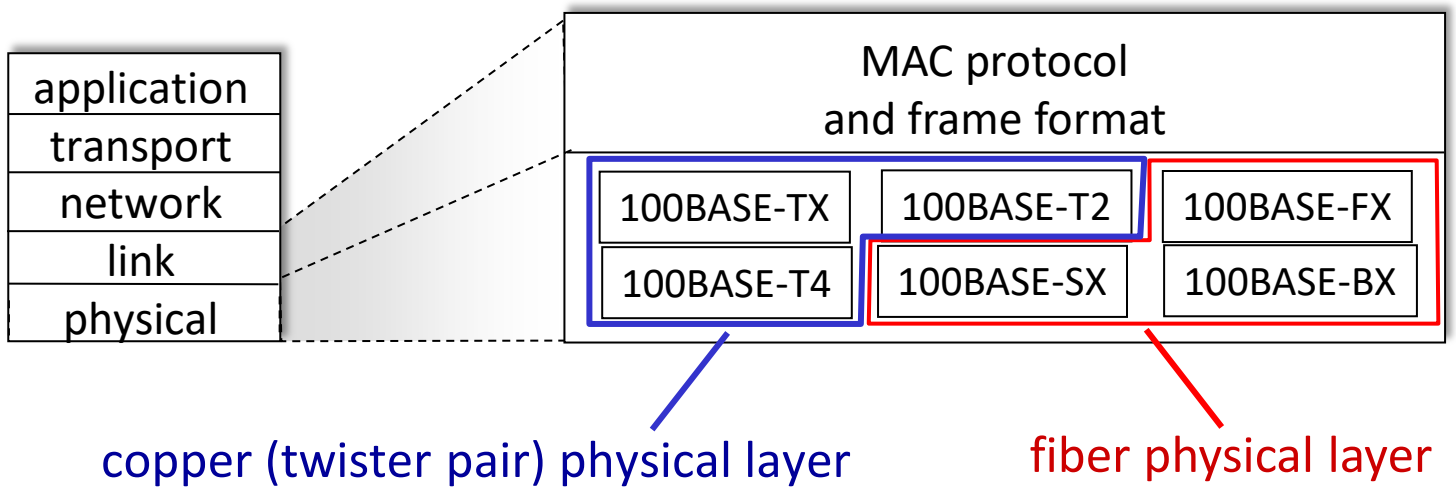
- addresses: 6 byte nguồn, đích
  - Nếu card mạng nhận frame khớp địa chỉ đích hoặc nhận được địa chỉ quảng bá ví dụ gói ARP, nó truyền dữ liệu trong frame lên tang mạng.
  - Ngược lại sẽ vứt bỏ frame
- type: chỉ định giao thức tầng cao hơn
  - Hầu như là IP nhưng cũng có thể l:, Novell IPX, AppleTalk
- CRC: bit kiểm tra lỗi
  - Phát hiện lỗi: frame lỗi sẽ bị vứt bỏ

Ethernet: không tin cậy và không hướng kết nối

- connectionless: không bắt tay giữa NIC gửi và NIC nhận
- unreliable: NIC nhận không gửi ACKs NAKs NIC nhận
  - Dữ liệu bị vứt chỉ khôi phục nếu người nhận khởi tạo sử dụng rdt tầng cao hơn ngược lại mất dữ liệu bị vứt bỏ.
- Giao thức MAC của Ethernet: CSMA/CD không chia slot kết hợp với giải thuật binary backoff

Tiêu chuẩn Ethernet 802.3 : link & physical layers

- *Nhiều giao thức MAC và định dạng frame*
  - Tốc độ khác nhau: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps
  - Môi trường truyền khác nhau: quang, đồng



Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

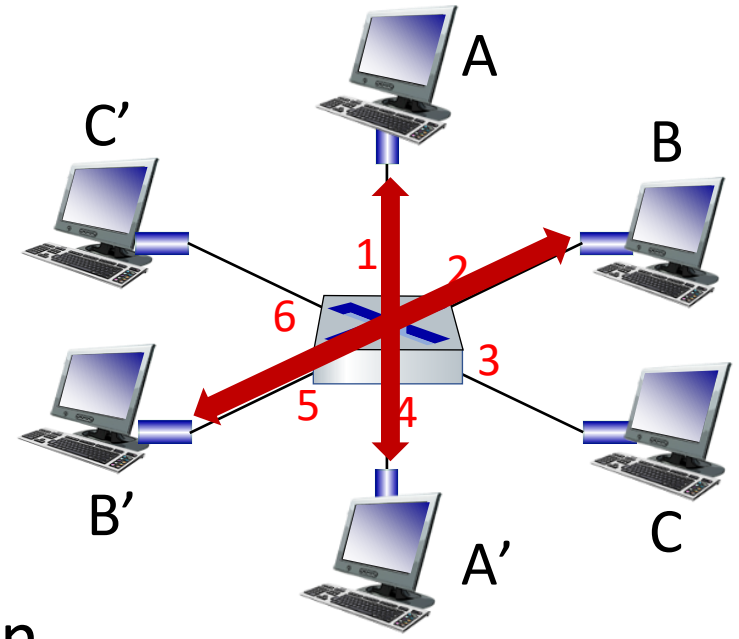
Ethernet switch

- Switch là thiết bị của tầng link:
  - Lưu và gửi Ethernet frames
  - Một frame đến với địa chỉ MAC, lựa chọn link để frame đi ra, sử dụng CSMA/CD để truy cập phân đoạn mạng.
- Trong suốt: host không biết về switch
- plug-and-play, tự học
  - switches không cần cấu hình

Switch: truyền đồng bộ đa luồng

- Host kết nối trực tiếp tới switch
- Switch đưa gói vào bộ đệm
- Ethernet được dùng cho mỗi link đầu vào nên:
  - Không có đụng độ: full duplex
  - Mỗi link là một miền đụng độ

- switching: A-to-A’ và B-to-B’ có thể truyền đồng thời không bị đụng độ

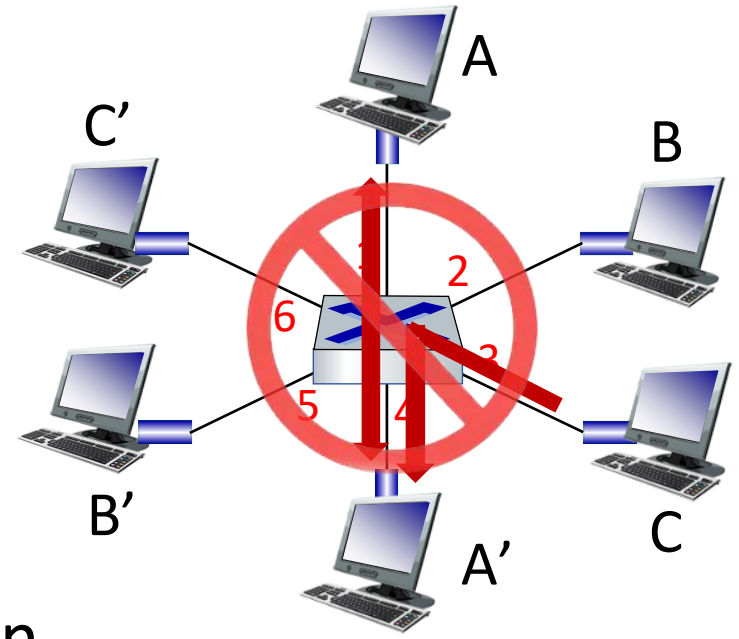


switch with six interfaces (1,2,3,4,5,6)

Switch: truyền đồng bộ đa luồng

- Host kết nối trực tiếp tới switch
- Switch đưa gói vào bộ đệm
- Ethernet được dùng cho mỗi link đầu vào nên:
  - Không có đụng độ: full duplex
  - Mỗi link là một miền đụng độ

- switching: A-to-A’ và B-to-B’ có thể truyền đồng thời không bị đụng độ
  - Nhưng A-to-A’ và C to A’ không thể đồng thời



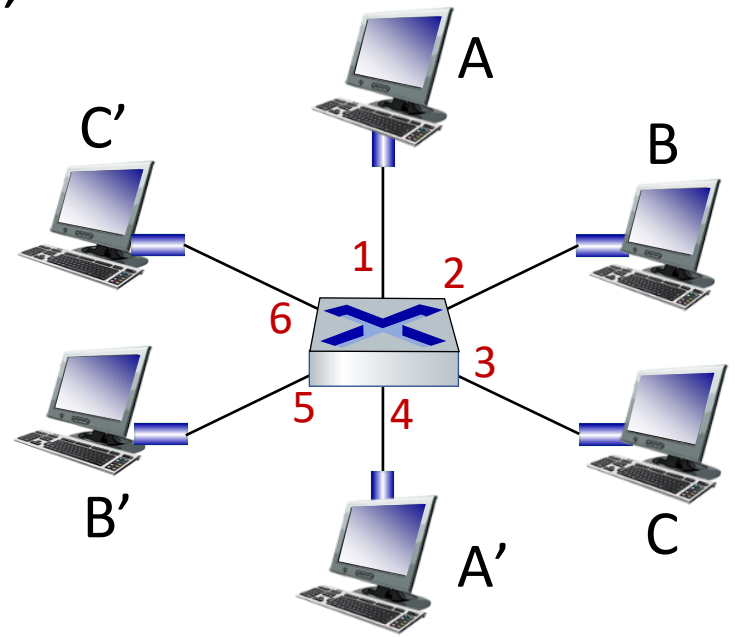
switch with six interfaces (1,2,3,4,5,6)

Switch forwarding table

Q: làm thế nào switch biết tới A’ qua cổng 4, B’ qua cổng 5?

A: mỗi switch có switch table, mỗi entry trong bảng là:

- (địa chỉ MAC, interface, nhãn thời gian)
- Giống như bảng định tuyến!

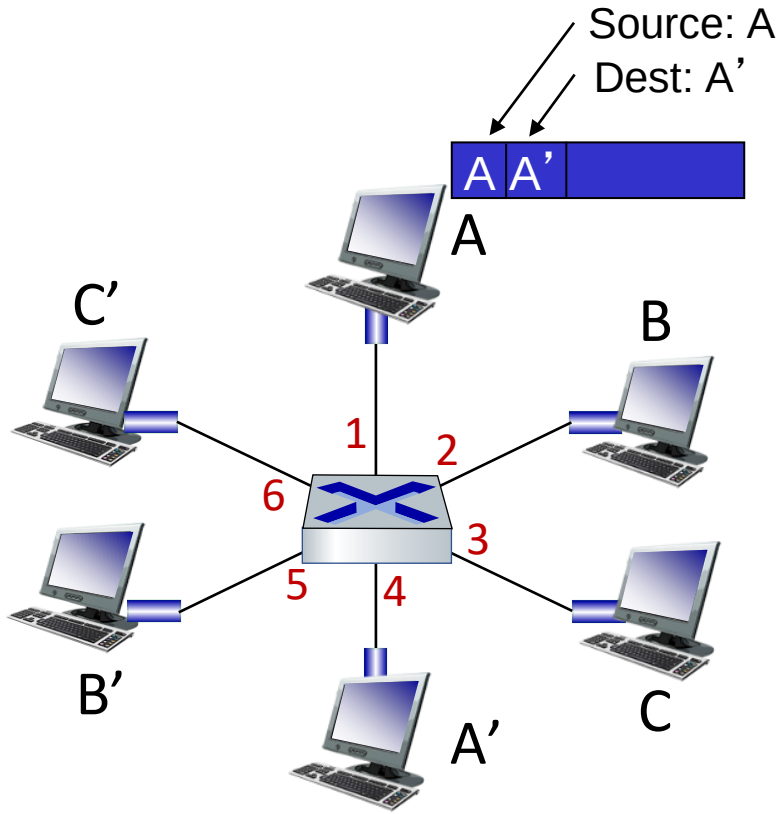


Q: các entry được tạo và duy trì như thế nào trong switch table?

- Có giống với giao thức định tuyến không?

Switch: tự học

- switch *học* hosts có thể đi tới thông qua interfaces nào?
  - Khi nhận frame, switch học địa chỉ của người gửi



MAC addr	interface	TTL
A	*1*	*60*

Switch table (initially empty)

Switch: frame filtering/forwarding

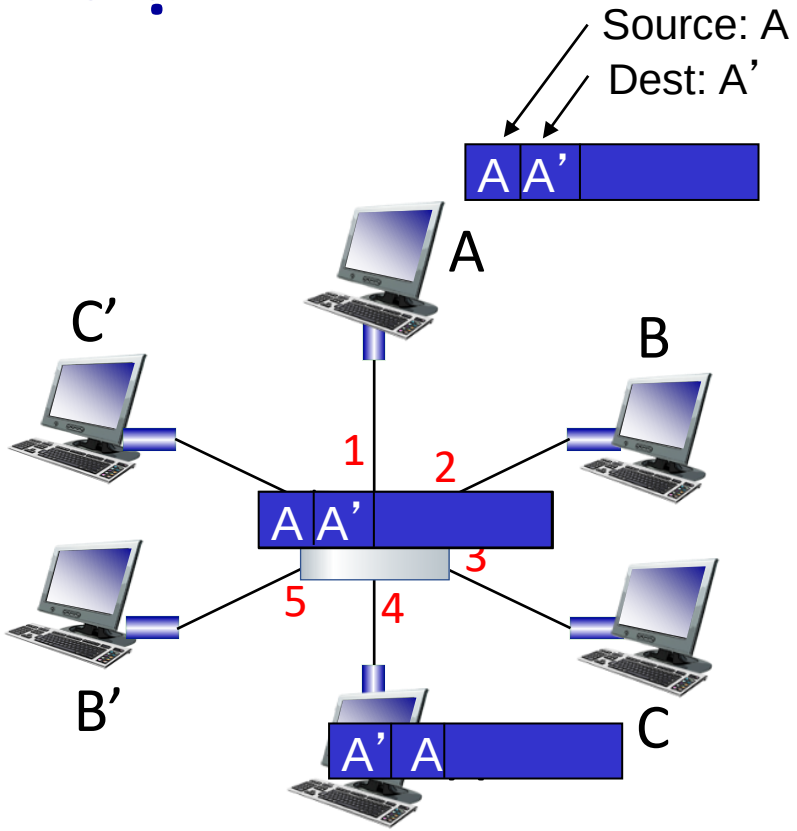
Khi frame nhận tại switch:

1. record incoming link, MAC address of sending host
2. index switch table using MAC destination address
3. if entry found for destination
   then {
   if destination on segment from which frame arrived
      then drop frame
       else forward frame on interface indicated by entry
    }
   else flood /* forward on all interfaces except arriving interface */

Self-learning, forwarding: ví dụ

- frame destination, A’, không biết: Làm lụt
- A đã biết:

  chọn link đầu ra và forward đi

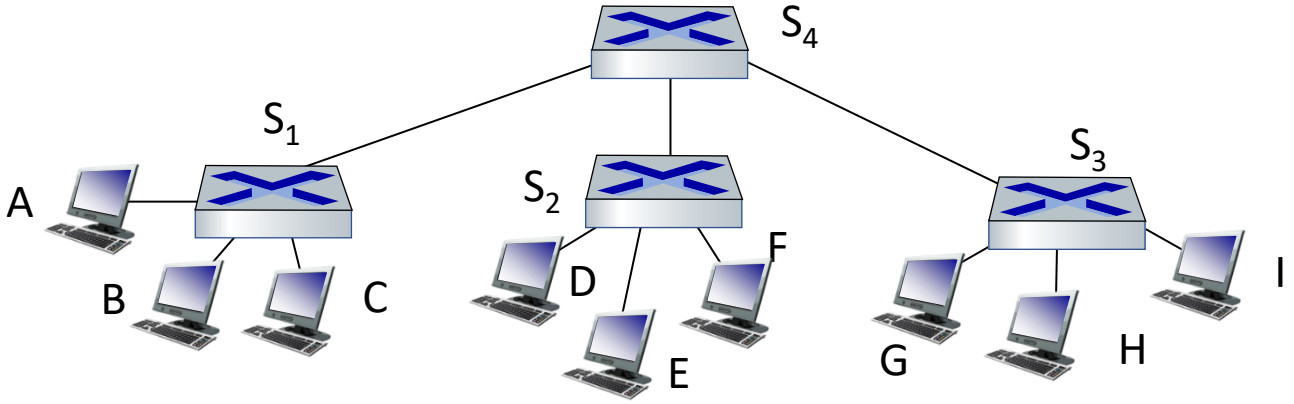


MAC addr	interface	TTL
A	*1*	*60*
A ’	*4*	*60*

switch table (initially empty)

Interconnecting switches

Các switch có thể nối với nhau:

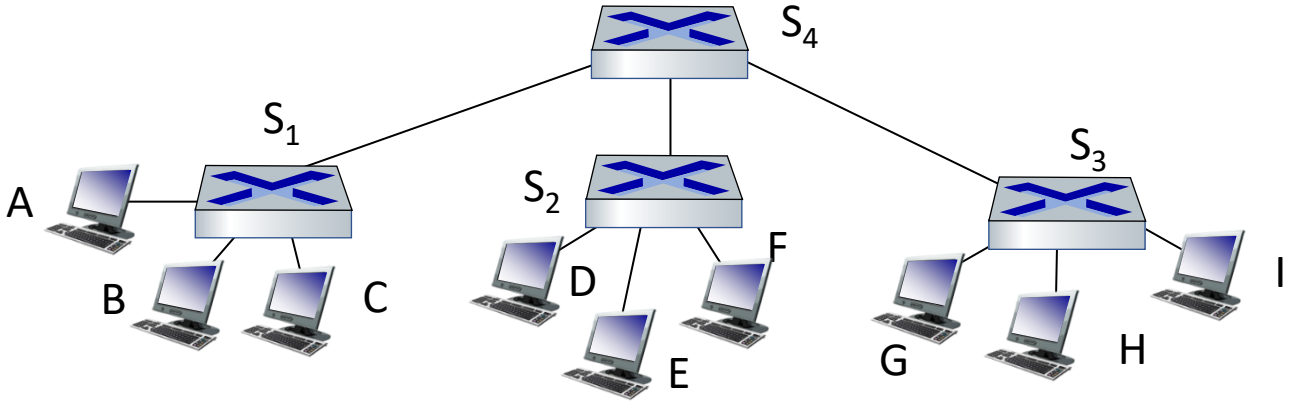


Q: A gửi tới G- S1 là cách nào biết frame tới G thông qua S4 và S3

- *A:* self learning! (làm việc chính xác như với 1 switch!)

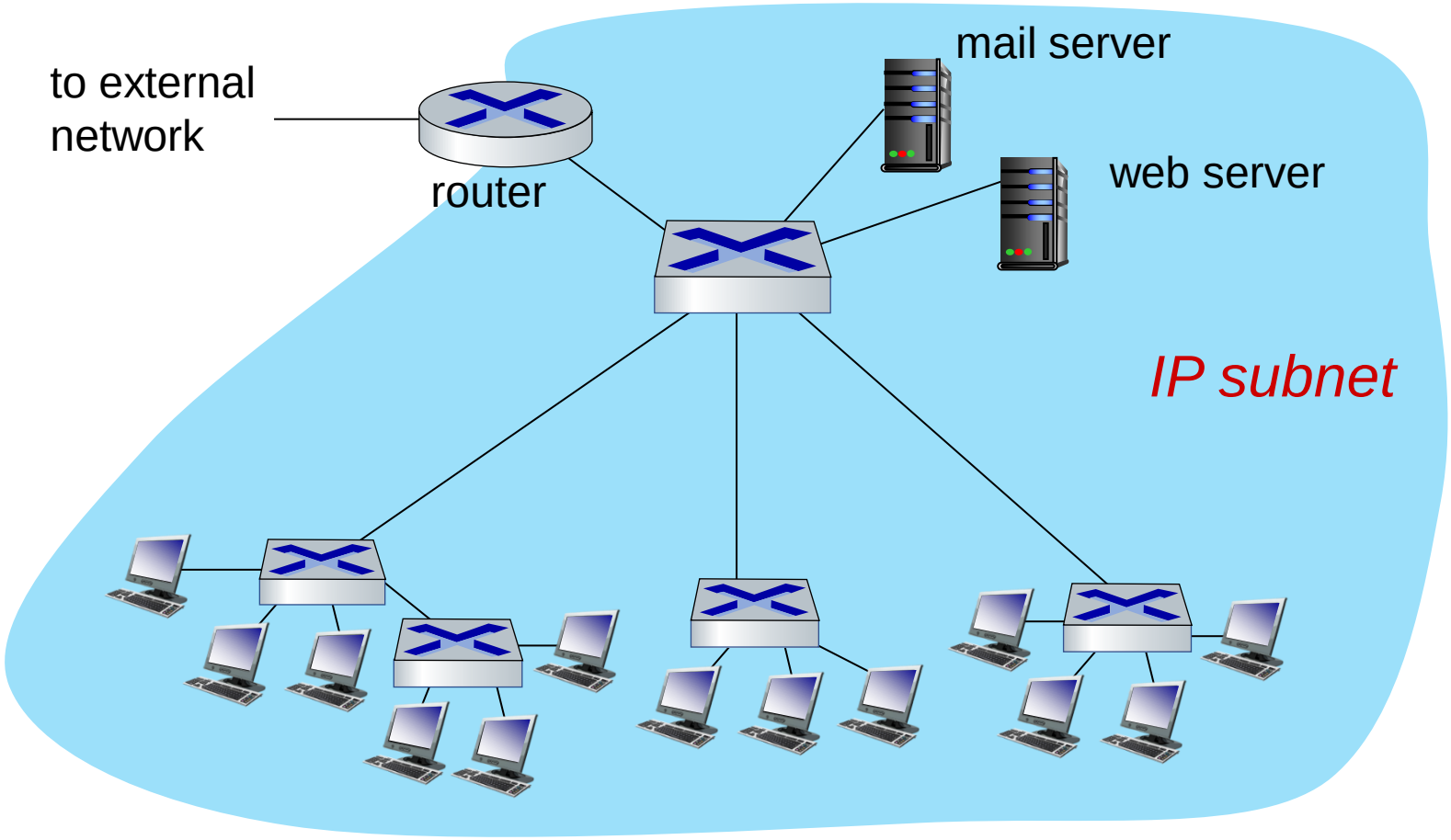
Self-learning multi-switch: ví dụ

C gửi cho I, I phản hồi C



Q: hiển thị bảng switch table của S_1, S_2, S_3, S_4

Mạng mô hình nhỏ



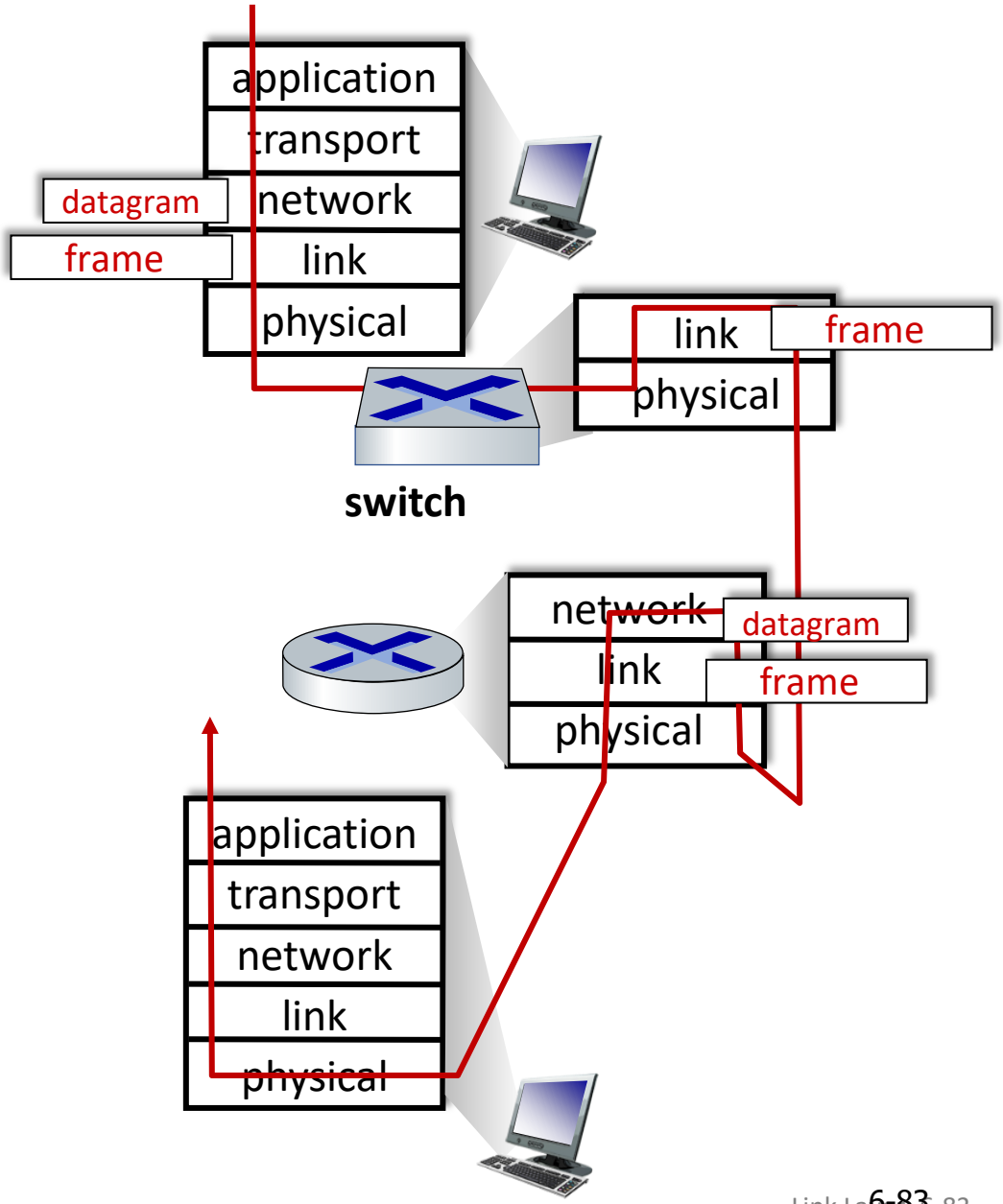
Switches vs. routers

Cả hai đều store-and-forward:

- *routers*: thiết bị tầng mạng (kiểm tra header tầng mạng)
- *switches:* thiết bị tầng link(kiểm tra header của tầng link-layer headers)

Đều có forwarding tables:

- *routers:* tính toán bảng sử dụng giao thức định tuyến, địa chỉ IP
- *switches:* xây dựng forwarding bằng cách làm lụt học địa chỉ MAC



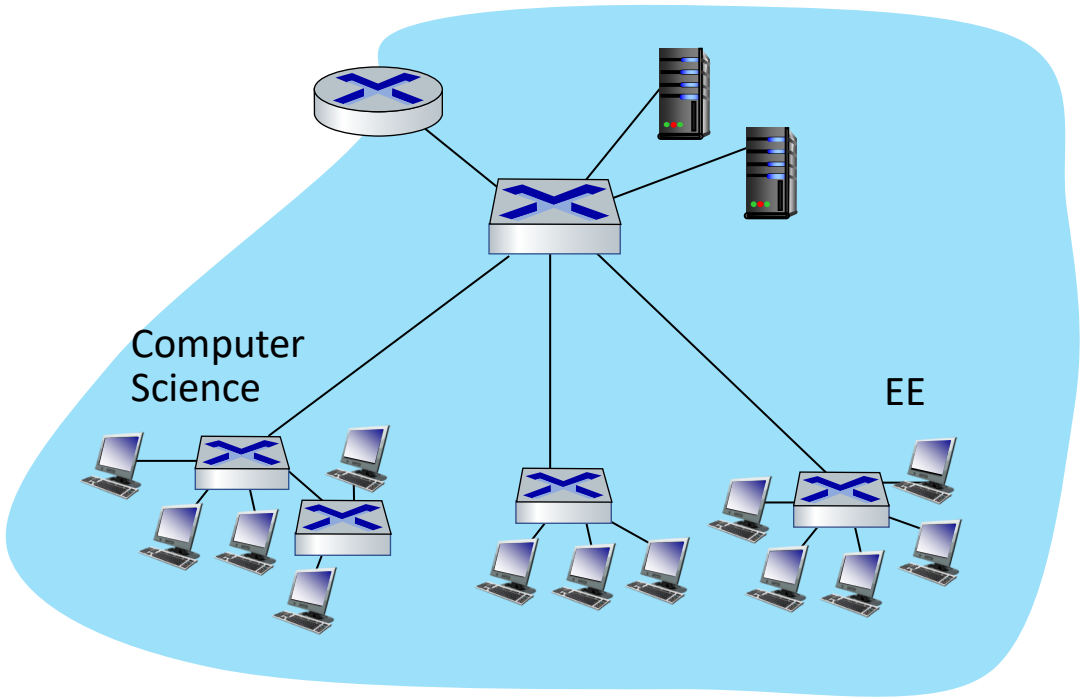
Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Virtual LANs (VLANs)

Q: Nếu LAN quá lớn ?

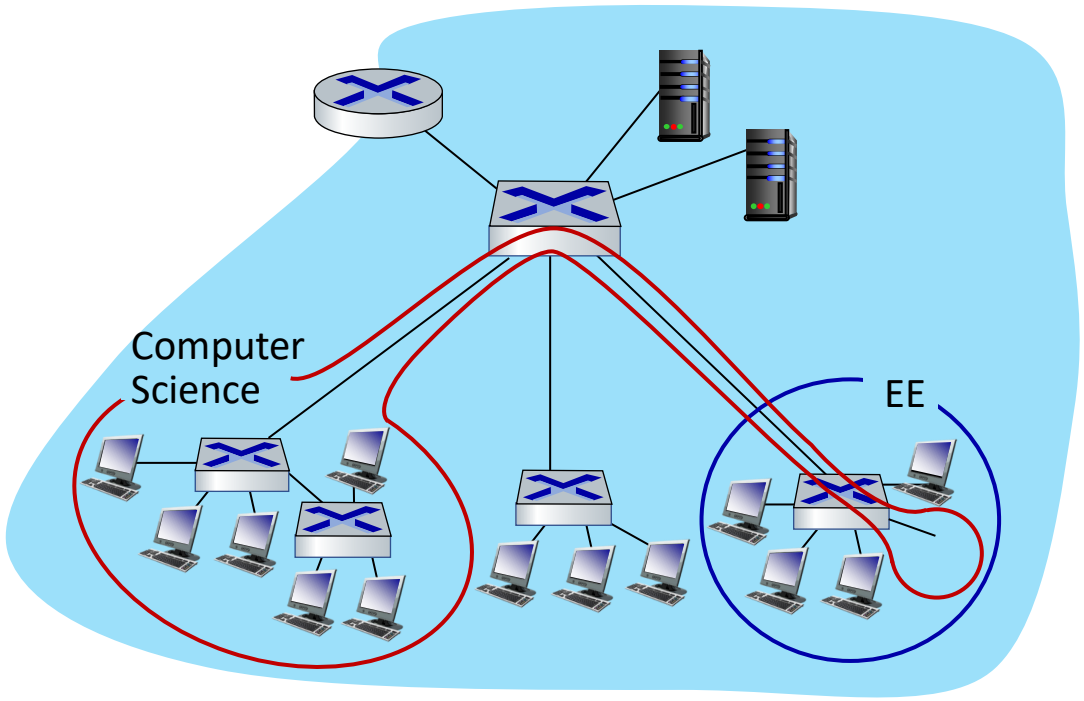


Một miền quảng bá:

- *scaling:* tất cả các lưu lượng quảng bá tầng 2 (ARP, DHCP, không biết MAC) phải truyền cho toàn bộ LAN
- Hiệu quả, bảo mật, chính sách sẽ bị ảnh hưởng

Virtual LANs (VLANs)

Q: Nếu LAN quá lớn ?



Một miền quảng bá:

- *scaling:* tất cả các lưu lượng quảng bá tầng 2 (ARP, DHCP, không biết MAC) phải truyền cho toàn bộ LAN
- Hiệu quả, bảo mật, chính sách sẽ bị ảnh hướng

Vấn đề quản trị:

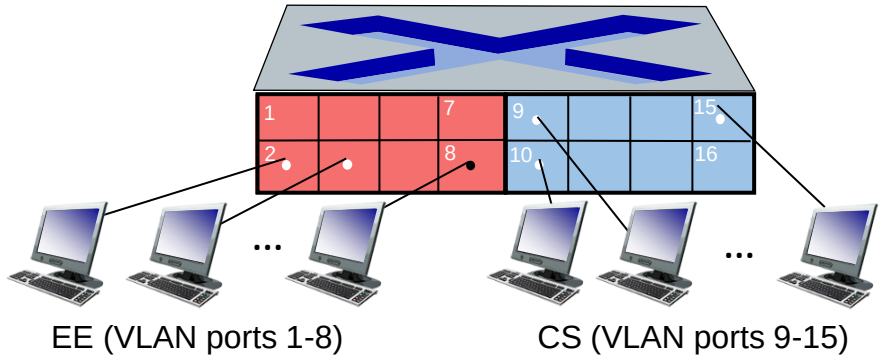
- Người dùng phía phòng khoa học máy tính CS di truyển tới phòng EE – *kết nối với switch của EE*, nhưng vẫn muốn giữ nguyên kết nối với mạng CS

Port-based VLANs

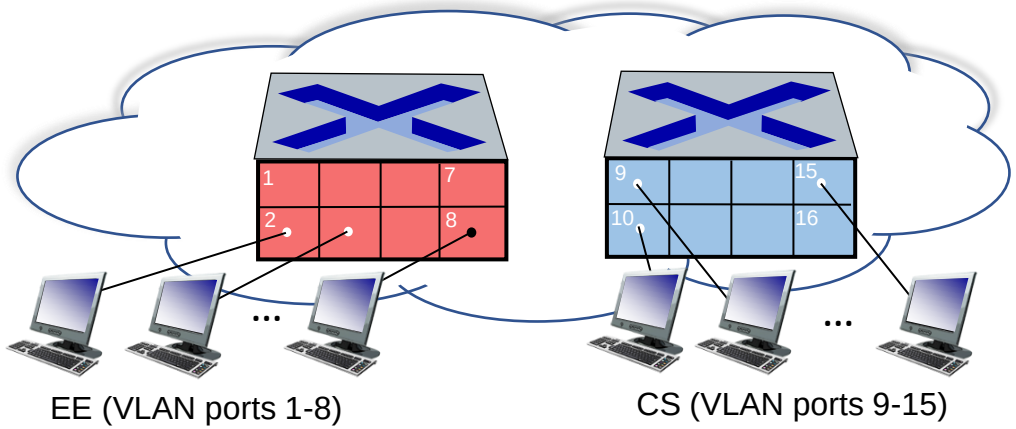
Virtual Local Area Network (VLAN)

switch(es) hỗ trợ VLAN có khả năng cho phép nhiều LAN ảo trong cùng một kiến trúc vật lý

port-based VLAN: các cổng của switch được nhóm lại(bằng phần mềm quản trị switch) vì vậy một switch đơn

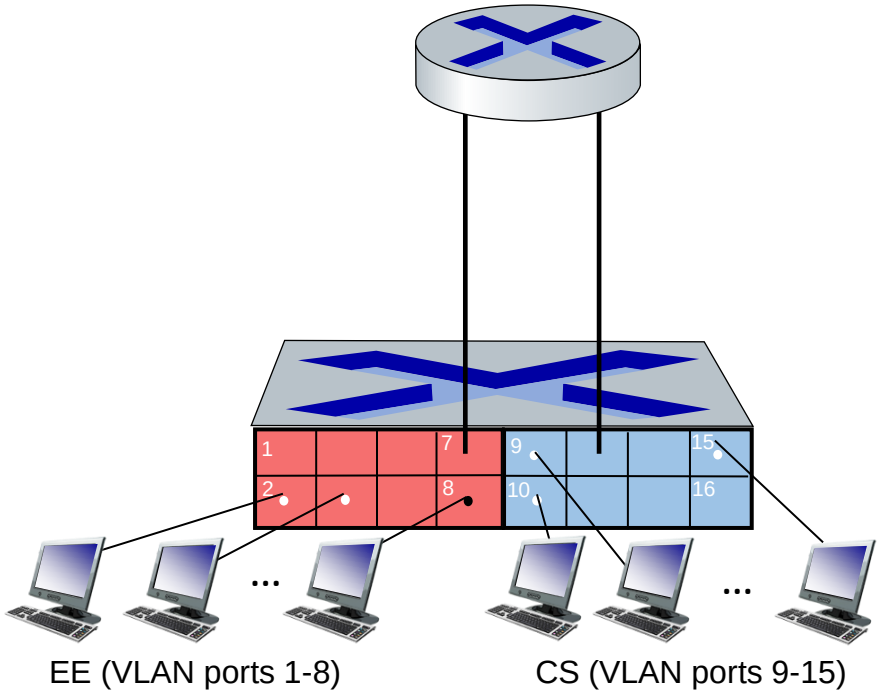


… có nhiều switch ảo

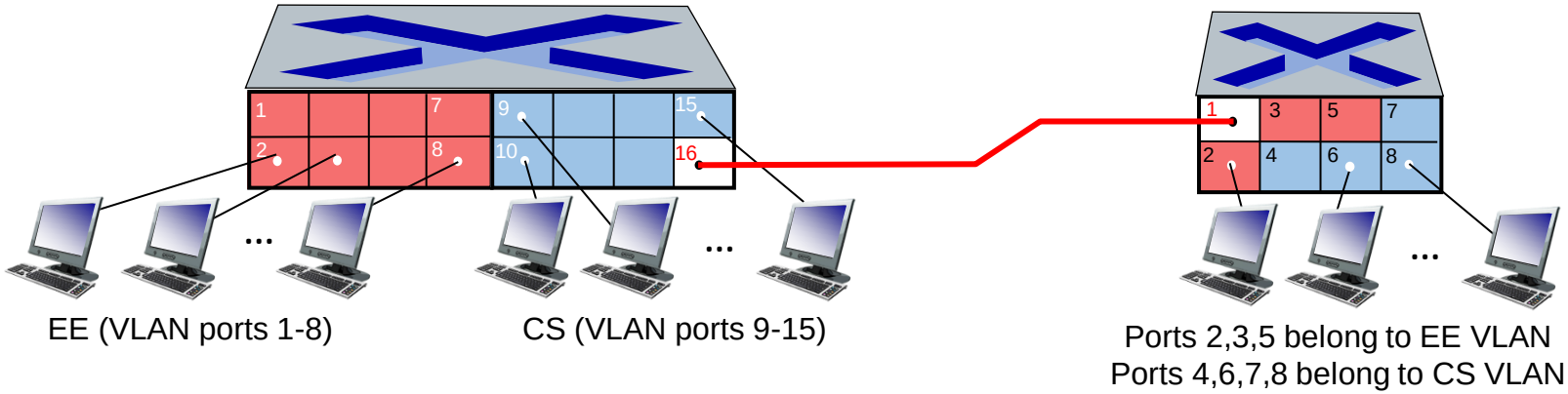


Port-based VLANs

- Cô lập lưu lượng: frame từ cổng 1-8 chỉ truyền trong phạm vi từ 1-8.
  - Có thể định nghĩa VLAN dựa vào địa chỉ MAC
- dynamic membership: port có thể gán động cho các VLANs
- Giữa các VLANS: được làm bằng định tuyến(giống như các switch độc lập)
  - Trong thực tế nhà cung cấp bán switch và cả router



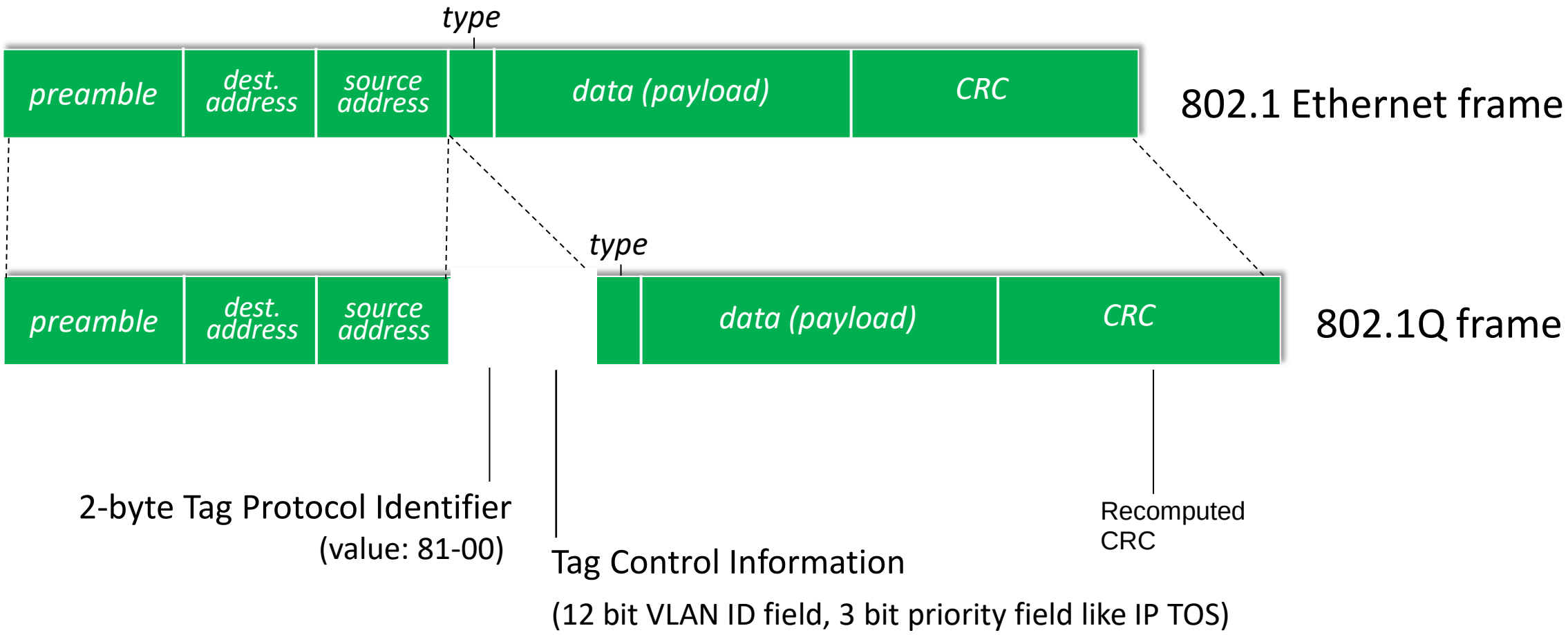
VLANS trên nhiều switch



Cổng trunk : kết nối các host cùng vlan giữa các switch vật lý với nhau

- Frame trong phạm vi VLAN nói chuyện với nhau, các frame phải được gán id
- Giao thức 802.1q protocol thêm một số thông tin vào header của frames để chuyển các frame trên các trunk port.

802.1Q VLAN frame format

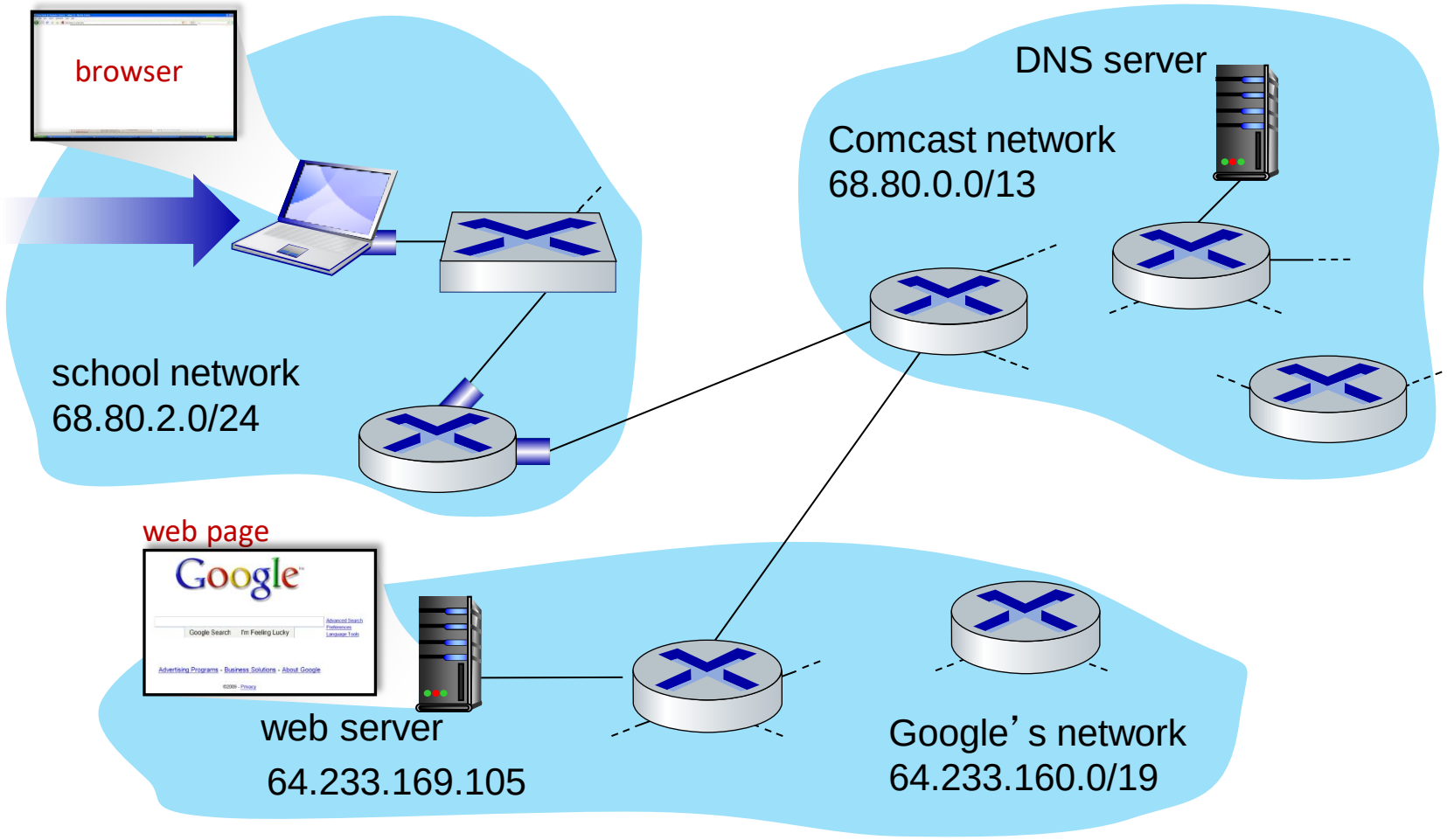


Link layer, LANs: roadmap

- introduction
- error detection, correction
- multiple access protocols
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs

- a day in the life of a web request

A day in the life: kịch bản

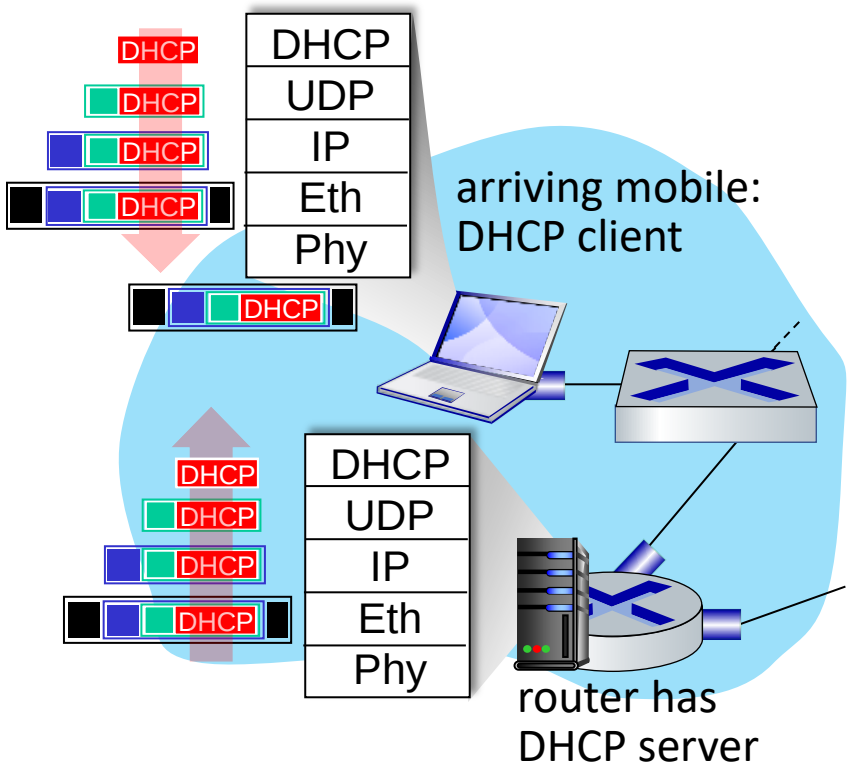


Kịch bản:

- Một client sử dụng laptop
- Yêu cầu truy cập web www.google.com

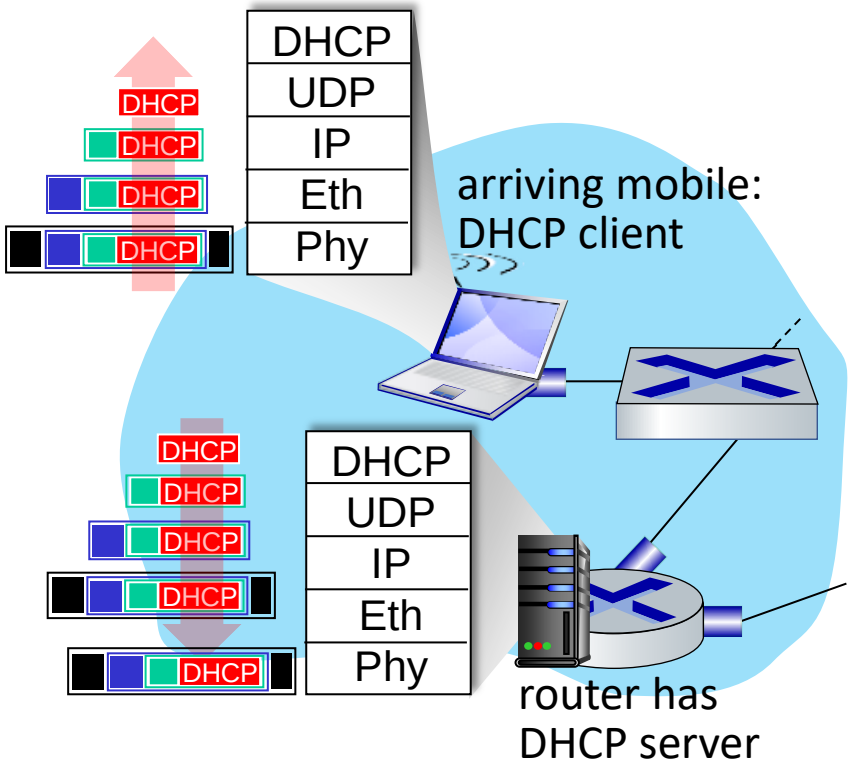
Sounds simple!

A day in the life: connecting to the Internet



- Laptop cần biết: địa chỉ ip của nó, gateway, địac chỉ DNS server: sử dụng DHCP
- DHCP request đóng gói trong encapsulated UDP, lại được đóng gói trong IP, lại được đóng gói trong 802.3 Ethernet
- Ethernet frame broadcast (đích: FFFFFFFFFFFF) trên LAN, được router chạy DHCP server nhận
- Bên client: Trích (tách) IP từ Ethernet, trích UDP từ IP, trích DHCP từ UDP

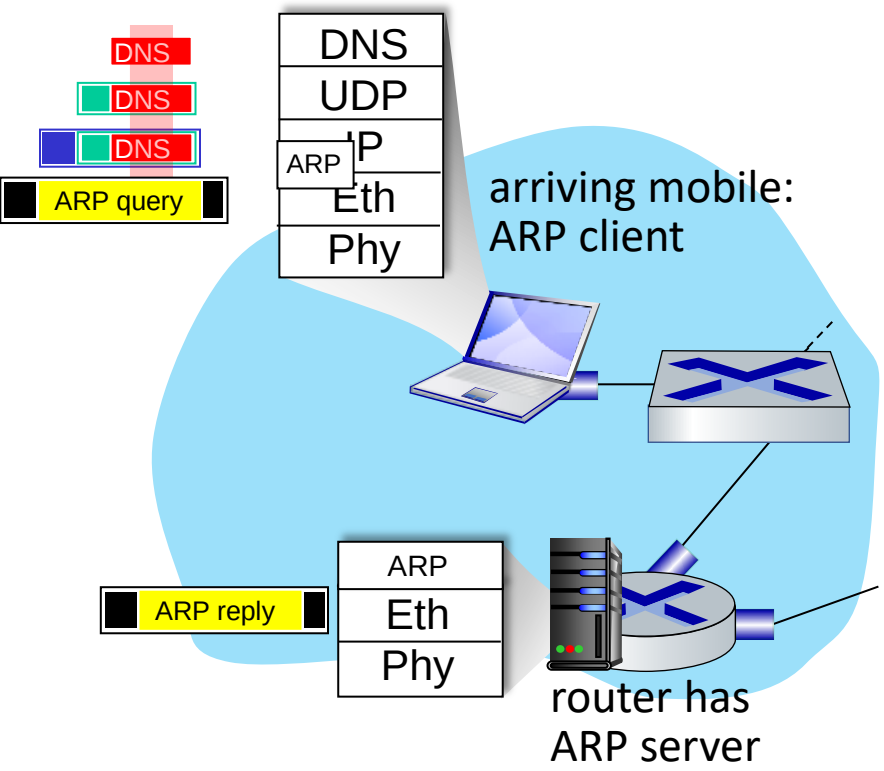
A day in the life: connecting to the Internet



- DHCP server tạo DHCP ACK chứa địa chỉ ip cho client, gateway, địa chỉ dns server
- Đóng gói bên server, tách gói bên client
- DHCP client nhận DHCP ACK reply

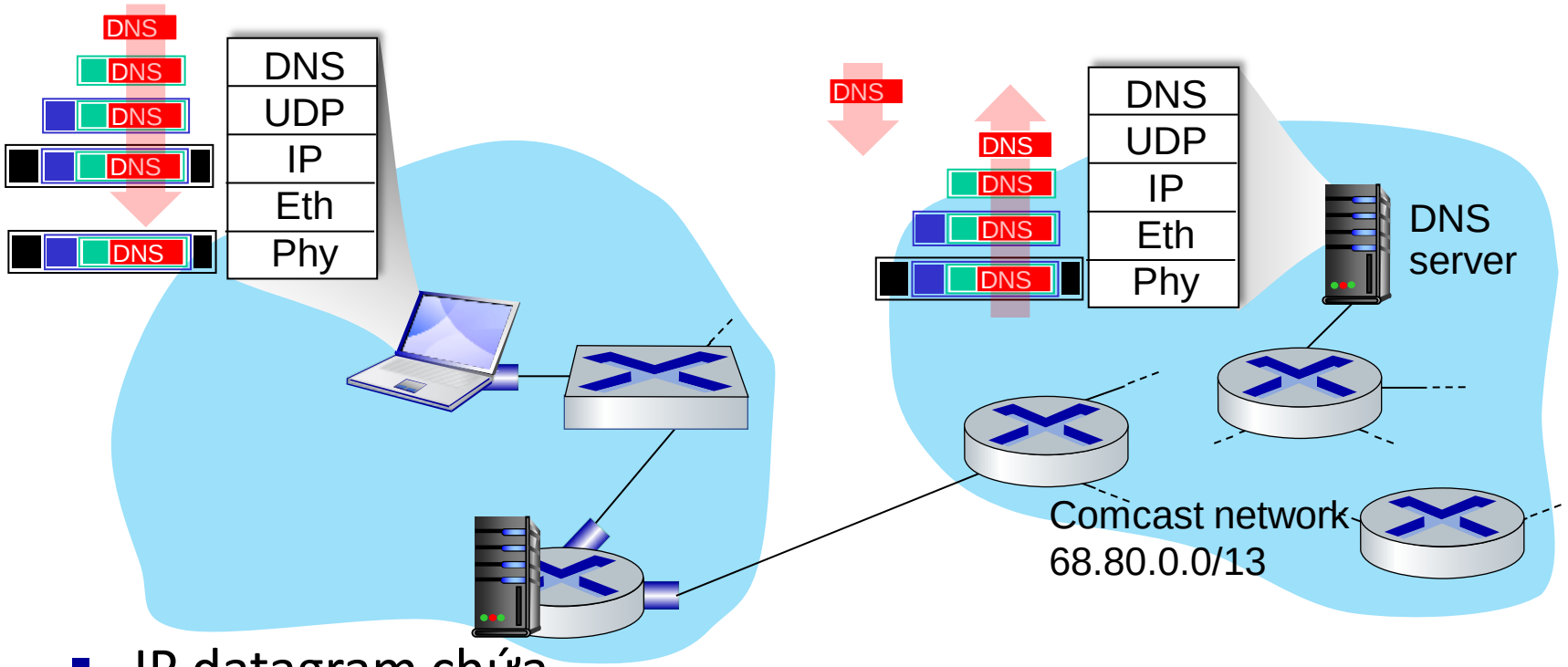
Client bây giờ có IP address, gateway, địa chỉ dns server

A day in the life… ARP (before DNS, before HTTP)



- Trước khi gửi HTTP request, cần biết địa chỉ ip của www.google.com: DNS
- DNS query được tạo, đóng gói trong UDP, sau đó lại đóng gói trong IP, sau đó trong Ethernet để gửi frame tới router, cần biết được MAC của interface trên rotuer: ARP
- ARP query quảng bá, được router nhận, trả lời với ARP reply thông tin địa chỉ MAC của interface router (gateway)
- Client biết địa chỉ MAC của gateway, vì vậy có thể gửi frame chứa DNS query

A day in the life… using DNS

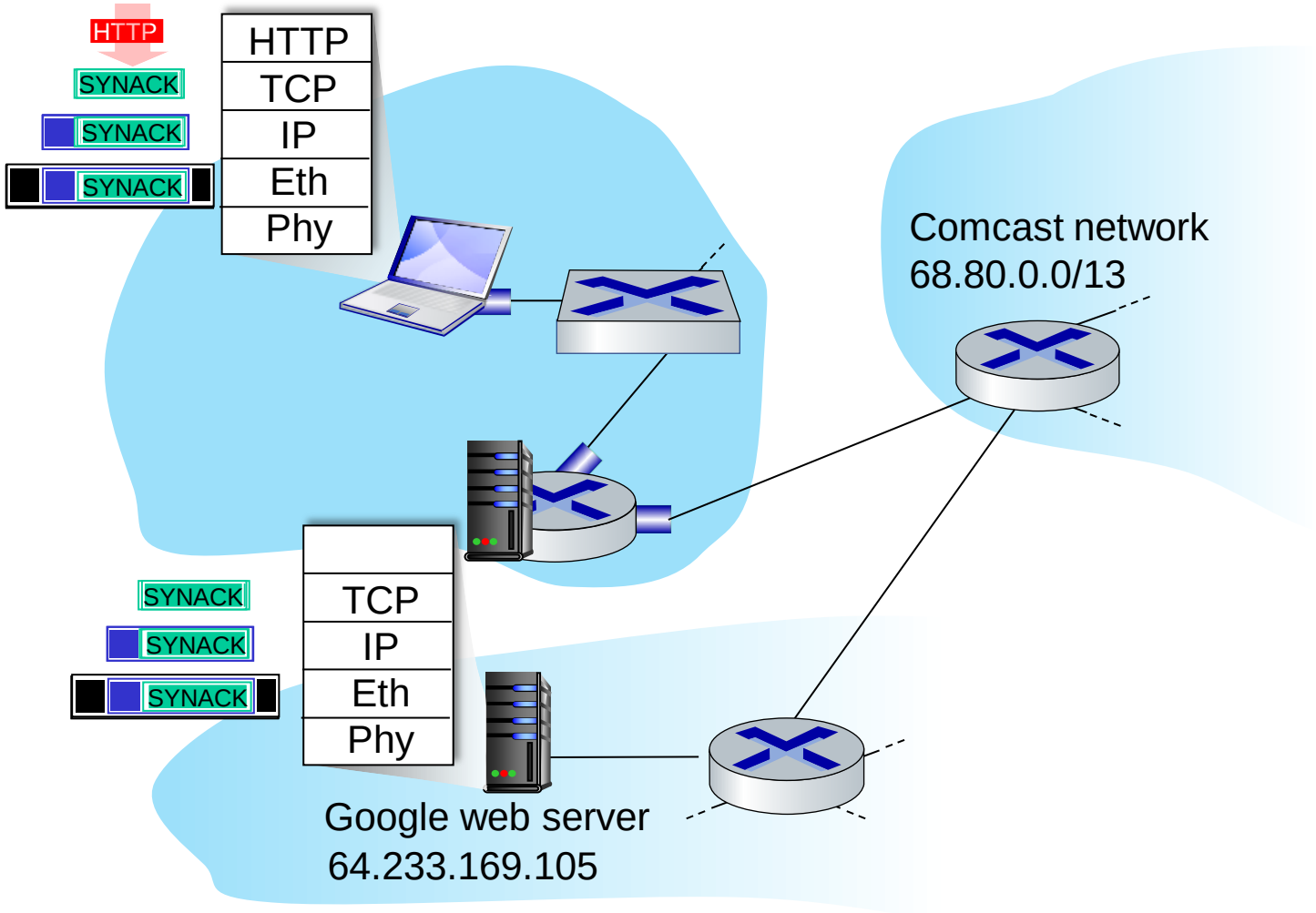


- Tách gói tới DNS
- DNS replies tới client với địa chỉ IP address của www.google.com

- IP datagram chứa DNS query được gửi từ host qua switch lên router thứ 1

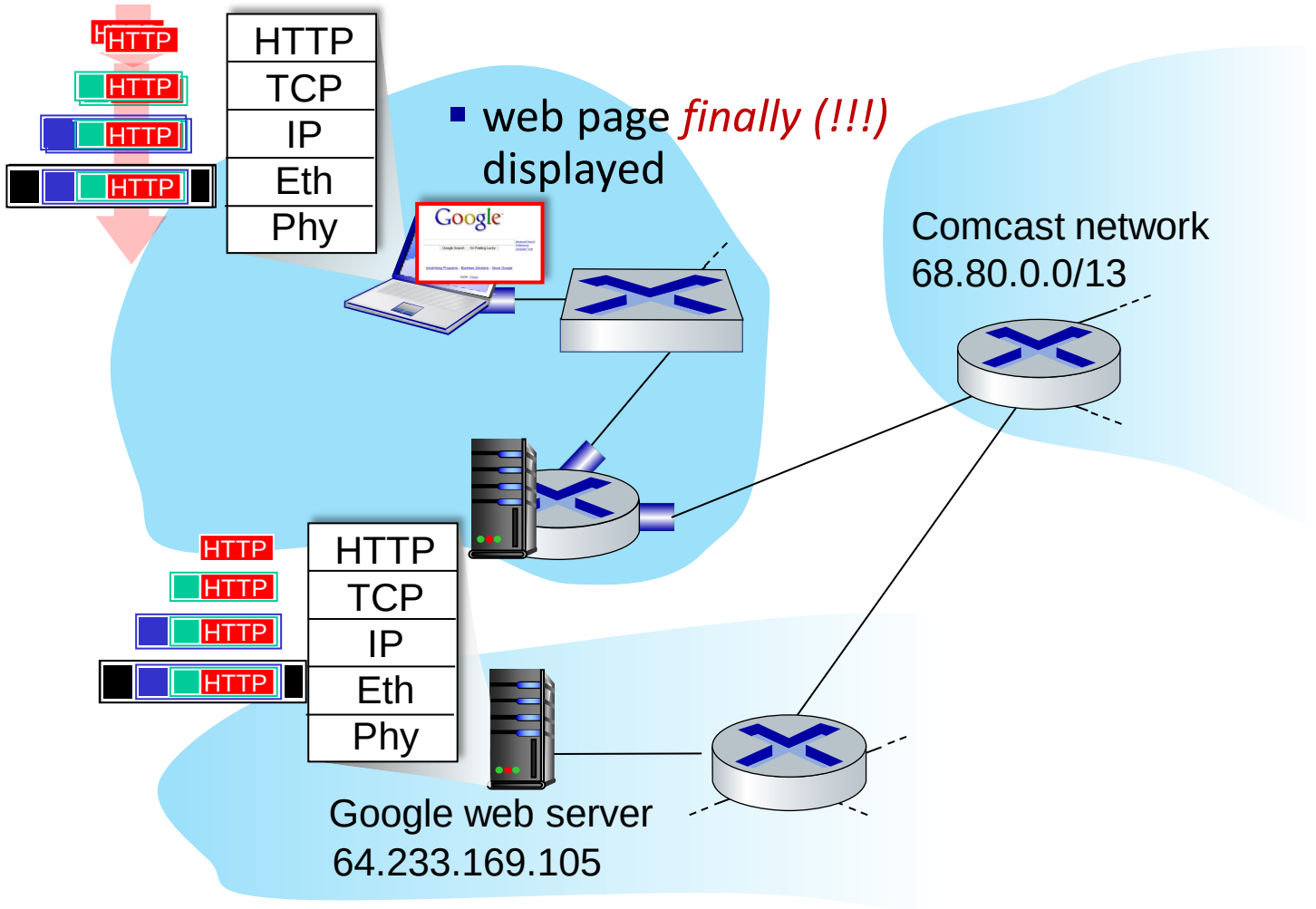
- IP datagram di chuyển trong mạng qua các tuyến đường được định tuyến bởi giao thức định tuyến: RIP, OSPF, IS-IS và/hoặc BGP) tới DNS server

A day in the life…kết nối TCP mang HTTP



- Để gửi HTTP request, client phải mở TCP socket tới web server
- TCP SYN segment (bước 1 của bắt tay 3 bước) liên miền giữa các router tới web server
- web server phản hồiTCP SYNACK (step 2 trongTCP 3-way handshake)
- TCP connection established!

A day in the life… HTTP request/reply



- HTTP request gửi tới TCP socket
- IP datagram chứa HTTP request định tuyến tới www.google.com
- web server phản hồi với HTTP reply (chứa trang web)
- IP datagram chứa HTTP reply được định tuyến trở lại client

Chapter 6: tổng kết

- Nguyên lý tầng link:
  - Phát hiện và sửa lỗi
  - Chia sẻ kênh quảng bá: đa truy cập
  - Địa chỉ tầng link
- Nhiều công nghệ khác nhau
  - Ethernet
  - switched LANS, VLANs
  - virtualized networks as a link layer: MPLS
- Kịch bản