

# Chapter 5

## Tầng Link

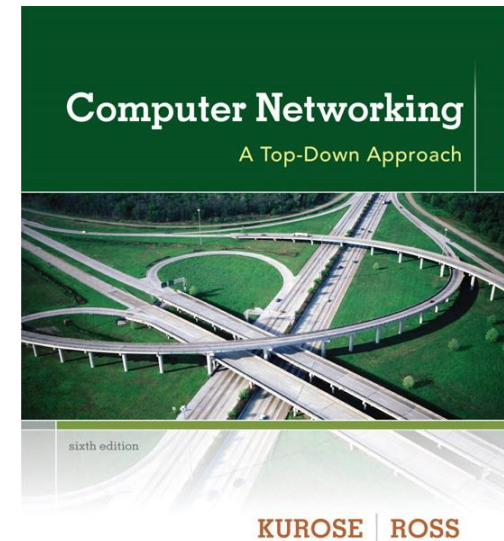
### A note on the use of these ppt slides:

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

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

Thanks and enjoy! JFK/KWR

© All material copyright 1996-2012  
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer  
Networking: A Top  
Down Approach  
6<sup>th</sup> edition  
Jim Kurose, Keith Ross  
Addison-Wesley  
March 2012*

# Chương 5: tầng Link

## *Mục tiêu:*

- ❖ Hiểu về các nguyên tắc của các dịch vụ tầng link:
  - Phát hiện lỗi và sửa lỗi
  - Chia sẻ kênh broadcast: đa truy cập
  - Định địa chỉ tầng link
  - local area networks: Ethernet, VLANs
- ❖ Khởi tạo và hiện thực một số công nghệ tầng link

# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

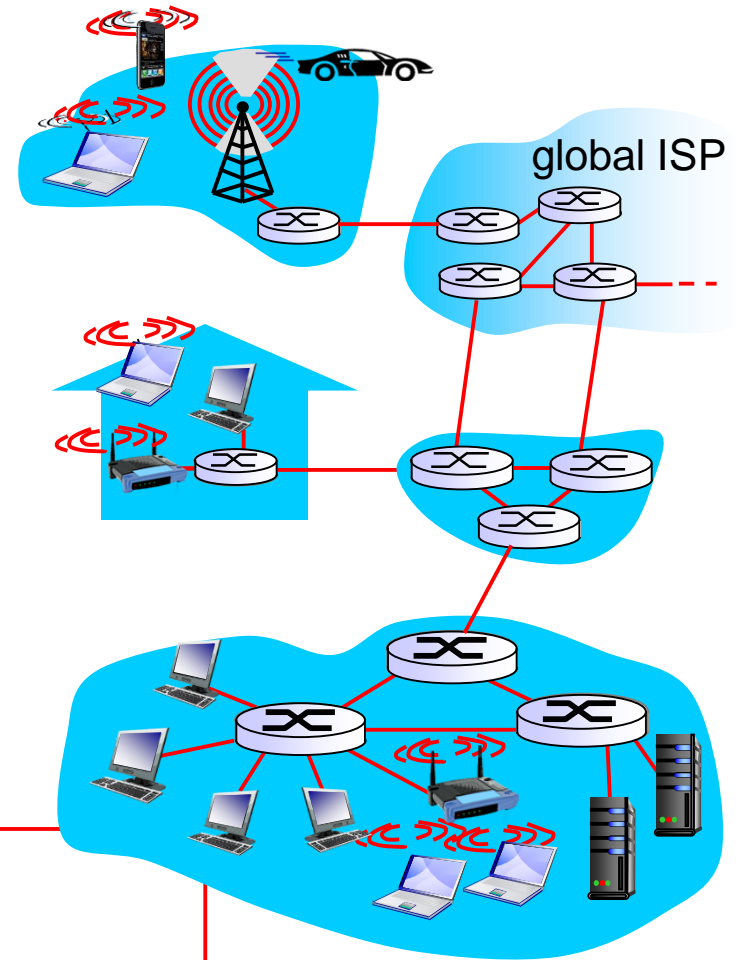
5.7 một ngày trong cuộc sống của một yêu cầu web

# Tầng Link: Giới thiệu

## *Thuật ngữ:*

- ❖ host và router: **node**
- ❖ Các kênh truyền thông kết nối các node lân cận (adjacent nodes) dọc theo đường truyền thông: **links**
  - Kết nối có dây (wired links)
  - Kết nối không dây (wireless links)
  - LANs
- ❖ Gói tin lớp 2: **frame**, đóng gói datagram

*Tầng data-link* có nhiệm vụ truyền datagram từ 1 node đến node lân cận vật lý (*physically adjacent* node) trên một đường liên kết



# Tầng Link: Ngữ cảnh

- ❖ datagram được truyền bởi các giao thức tầng link khác nhau trên các đường kết nối khác nhau:
  - Ví dụ: Ethernet trên đường kết nối thứ 1, frame relay trên các đường kết nối trung gian, 802.11 trên đường kết nối cuối cùng
- ❖ Mỗi giao thức tầng link cung cấp các dịch vụ khác nhau
  - Ví dụ: có thể hoặc không có thể cung cấp rdt trên đường kết nối

## *So sánh:*

- ❖ Hành trình từ Princeton đến Lausanne
  - limo: Princeton đến JFK
  - Máy bay: JFK đến Geneva
  - Xe lửa: Geneva đến Lausanne
- ❖ Khách du lịch = **datagram**
- ❖ segment tầng transport = liên kết truyền thông (**communication link**)
- ❖ Kiểu vận chuyển = **giao thức tầng link**
- ❖ Đại lý du lịch = **thuật toán định tuyến**

# Các dịch vụ tầng Link

## ❖ *Truy cập liên kết, framing:*

- Đóng gói datagram vào trong frame, thêm header và trailer
- Truy cập kênh truyền nếu môi trường được chia sẻ
- Các địa chỉ “MAC” được sử dụng trong các header để xác định nguồn và đích
  - Khác với địa chỉ IP!

## ❖ *Truyền tin cậy giữa các node lân cận(adjacent nodes)*

- Chúng ta đã tìm hiểu làm thế nào để thực hiện điều này ở chương 3!
- Ít khi được sử dụng trên đường kết nối lỗi thấp (cáp quang, một số loại cáp xoắn)
- Kết nối không dây: tỷ lệ lỗi cao
  - *Hỏi:* lý do độ tin cậy ở cả 2 cấp độ đường liên kết và end-end??

# Các dịch vụ tầng Link (tt)

## ❖ *Điều khiển luồng (flow control):*

- Điều khiển tốc độ truyền giữa các node gửi và nhận liên kết nhau

## ❖ *Phát hiện lỗi (error detection):*

- Lỗi gây ra bởi suy giảm tín hiệu.
- Bên nhận phát hiện lỗi:
  - Gửi tín hiệu để bên gửi truyền lại hoặc hủy bỏ frame bị lỗi

## ❖ *Sửa lỗi (error correction):*

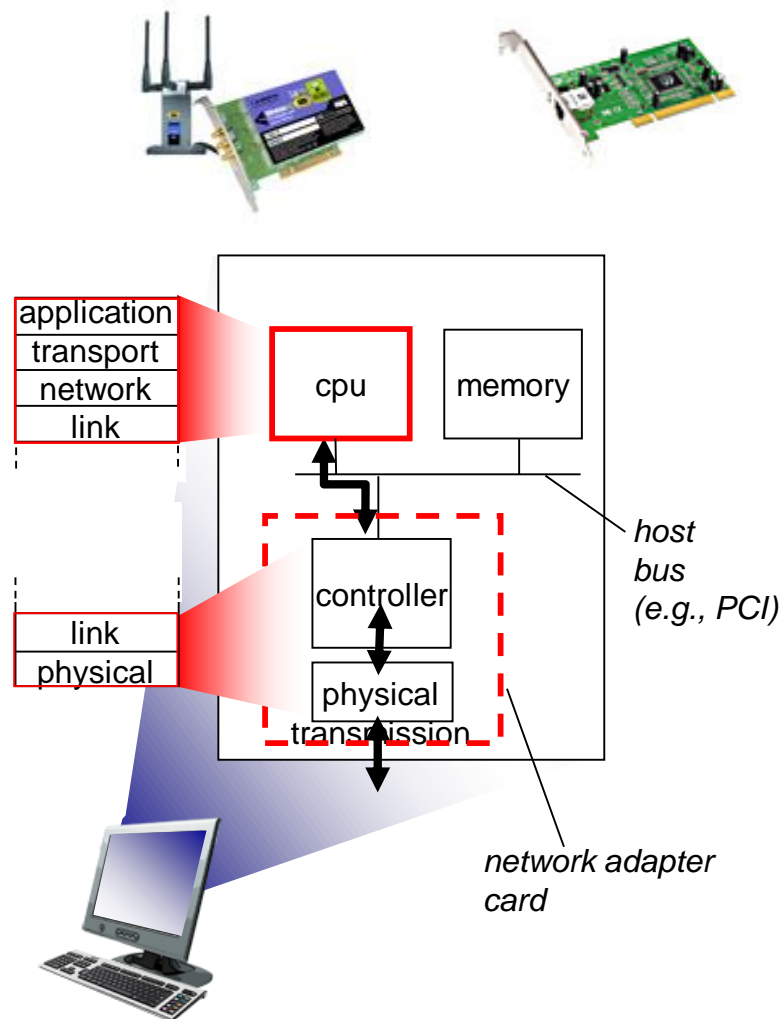
- Bên nhận xác định và **sửa** các bit lỗi mà không cần phải truyền lại

## ❖ *half-duplex và full-duplex*

- Với half duplex, các node tại các đầu cuối của kết nối có thể truyền, nhưng không đồng thời

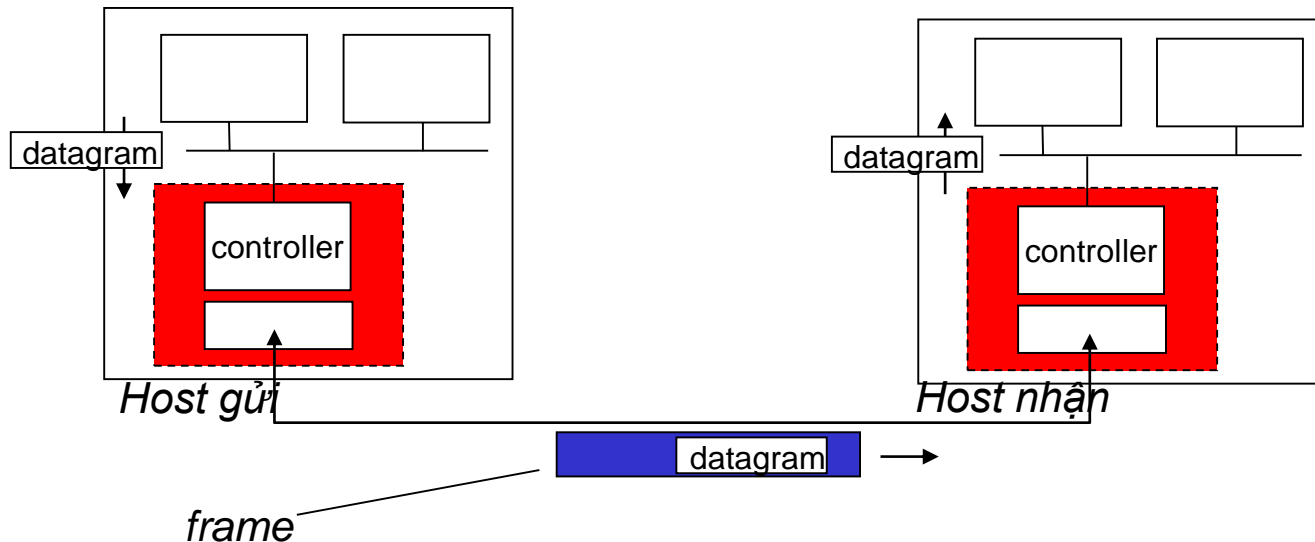
# Tầng link được thực hiện ở đâu?

- ❖ Trong mỗi và mọi host
- ❖ Tầng link được thực hiện trong “adaptor” (còn gọi là *network interface card* NIC) hoặc trên con chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - Thực hiện tầng physical và tầng link
- ❖ Gắn vào trong các bus hệ thống của host
- ❖ Sự kết hợp của phần cứng, phần mềm và firmware





# Các Adaptor trong truyền thông



## ❖ Bên gửi:

- Đóng gói datagram trong frame
- Thêm các bit kiểm tra lỗi, rdt và điều khiển luồng...

## ❖ Bên nhận

- Tìm lỗi, rdt và điều khiển luồng...
- Lấy ra các datagram, chuyển lên lớp trên tại nơi nhận

# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

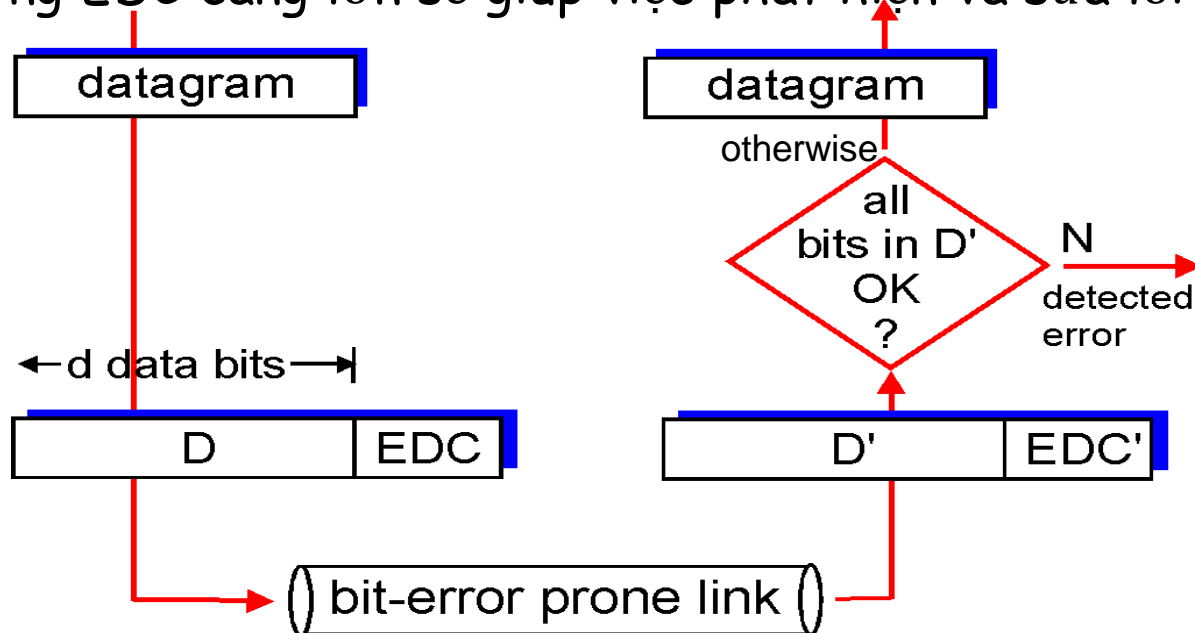
5.7 một ngày trong cuộc sống của một yêu cầu web

# Phát hiện lỗi

EDC= Error Detection and Correction bits (redundancy)

D = dữ liệu được bảo vệ bởi kiểm tra lỗi, có thể chứa các trường header

- Việc phát hiện lỗi không bảo đảm 100%!
  - giao thức có thể bỏ qua một số lỗi, nhưng hiếm khi
  - trường EDC càng lớn sẽ giúp việc phát hiện và sửa lỗi tốt hơn



# Kiểm tra chẵn lẻ (Parity checking)

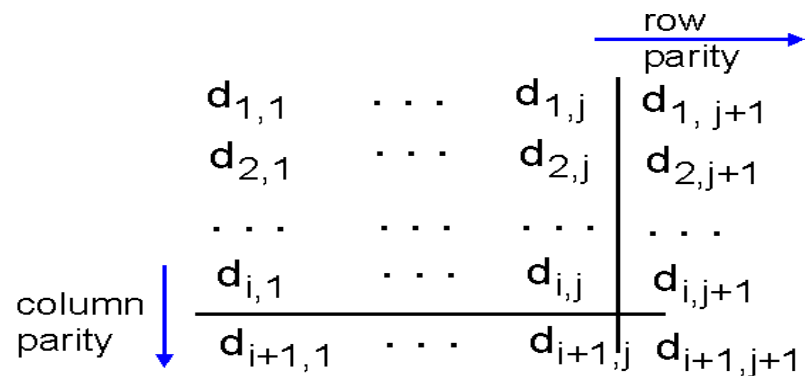
*bit parity đơn:*

❖ *Phát hiện các lỗi bit đơn*

*bit parity 2 chiều:*

❖ *phát hiện và sửa lỗi các bit đơn*

7 bits of data	(count of 1 bits)	8 bits including parity	
		even	odd
0000000	0	00000000	00000001
1010001	3	10100011	10100010
1101001	4	11010010	11010011
1111111	7	11111111	11111110



```

1 0 1 0 1 | 1
1 1 1 1 0 | 0
0 1 1 1 0 | 1
-----
0 0 1 0 1 | 0

```

*no errors*

```

1 0 1 0 1 | 1
1 1 1 1 0 | 0
0 1 1 1 0 | 1
-----
0 0 1 0 1 | 0

```

parity error

*correctable  
single bit error*

# Internet checksum

**Mục tiêu:** phát hiện "các lỗi" (ví dụ, các bit bị lộn) trong packet được truyền (chú ý: chỉ được dùng tại tầng transport)

## **Bên gửi:**

- ❖ Xử lý các nội dung của segment như một chuỗi các số nguyên 16-bit
- ❖ checksum: thêm(tổng bù 1) vào các nội dung của segment
- ❖ Bên gửi đặt các giá trị checksum vào trong trường checksum của UDP

## **Bên nhận:**

- ❖ Tính toán checksum của segment vừa nhận
- ❖ Kiểm tra xem có hay không giá trị của checksum vừa được tính có bằng với giá trị trong trường checksum:
  - không - phát hiện lỗi
  - có - không có lỗi được phát hiện. Nhưng có thể còn có lỗi khác không?

# Cyclic redundancy check

Now define

$T$  =  $n$ -bit frame to be transmitted

$D$  =  $k$ -bit block of data, or message, the first  $k$  bits of  $T$

$F$  =  $(n - k)$ -bit FCS, the last  $(n - k)$  bits of  $T$

$P$  = pattern of  $n - k + 1$  bits; this is the predetermined divisor

## 1. Given

Message  $D$  = 1010001101 (10 bits)

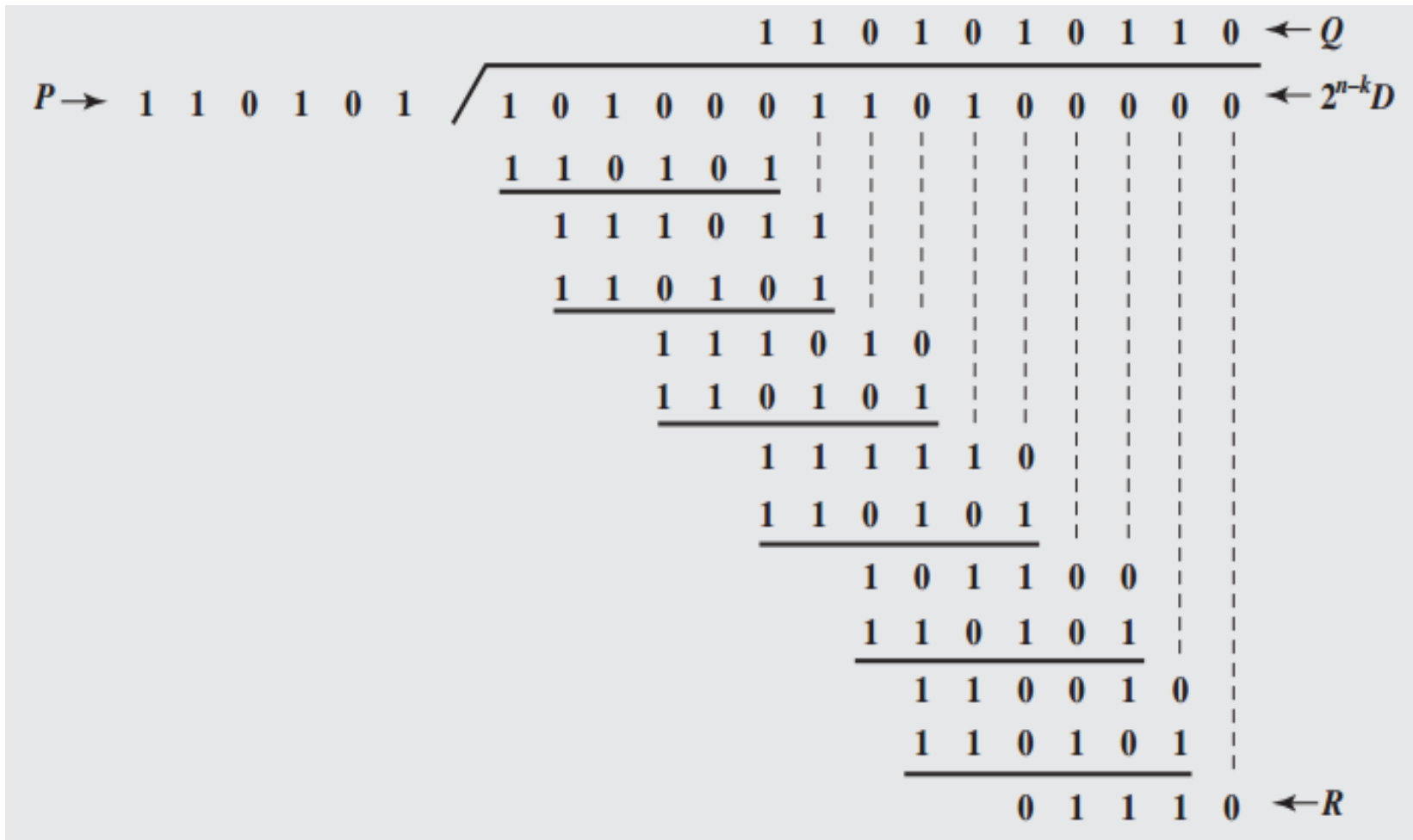
Pattern  $P$  = 110101 (6 bits)

FCS  $R$  = to be calculated (5 bits)

Thus,  $n = 15$ ,  $k = 10$ , and  $(n - k) = 5$ .

2. The message is multiplied by  $2^5$ , yielding 101000110100000.
3. This product is divided by  $P$ :

# Cyclic redundancy check



# Cyclic redundancy check

4. The remainder is added to  $2^5 D$  to give  $T = 101000110101110$ , which is transmitted.
5. If there are no errors, the receiver receives  $T$  intact. The received frame is divided by  $P$ :

$$\begin{array}{r}
 \begin{array}{cccccc}
 P \rightarrow & 1 & 1 & 0 & 1 & 0 & 1 & / & 
 \end{array}
 & 
 \begin{array}{cccccccccccccccc}
 & & & & & & & & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & \leftarrow Q \\
 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & & & \leftarrow T \\
 \hline
 1 & 1 & 0 & 1 & 0 & 1 & & & & & & & & & & & & \\
 \hline
 & 1 & 1 & 1 & 0 & 1 & 1 & & & & & & & & & & & \\
 & \hline
 & & 1 & 1 & 0 & 1 & 0 & 1 & & & & & & & & & & \\
 & & \hline
 & & & 1 & 1 & 1 & 0 & 1 & 0 & & & & & & & & & \\
 & & & \hline
 & & & & 1 & 1 & 0 & 1 & 0 & 1 & & & & & & & & \\
 & & & & \hline
 & & & & & 1 & 1 & 1 & 1 & 1 & 0 & & & & & & & \\
 & & & & & \hline
 & & & & & & 1 & 1 & 0 & 1 & 0 & 1 & & & & & & \\
 & & & & & & \hline
 & & & & & & & 1 & 0 & 1 & 1 & 1 & 1 & & & & & \\
 & & & & & & & \hline
 & & & & & & & & 1 & 1 & 0 & 1 & 0 & 1 & & & & \\
 & & & & & & & & \hline
 & & & & & & & & & 1 & 1 & 0 & 1 & 0 & 1 & & & \\
 & & & & & & & & & \hline
 & & & & & & & & & & 1 & 1 & 0 & 1 & 0 & 1 & & \\
 & & & & & & & & & & & \hline
 & & & & & & & & & & & & 0 & & & & \leftarrow R
 \end{array}
 \end{array}$$

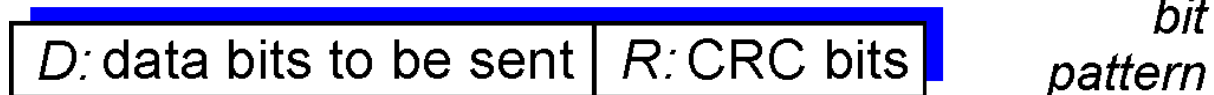
Because there is no remainder, it is assumed that there have been no errors.



# Cyclic redundancy check

- ❖ Phát hiện lỗi coding mạnh hơn
- ❖ Xem các bit dữ liệu, **D**, như một số nhị phân
- ❖ Chọn mẫu  $r+1$  bit (máy phát), **G**
- ❖ Mục tiêu: chọn  $r$  bit **CRC**, **R**, như thế
  - $\langle D, R \rangle$  chính xác chia hết cho  $G$  (theo cơ số 2)
  - Bên nhận biết  $G$ , chia  $\langle D, R \rangle$  cho  $G$ . Nếu phần dư khác không: lỗi được phát hiện!
  - Có thể phát hiện tất cả các lỗi nhỏ hơn  $r+1$  bits
- ❖ Được sử dụng rộng rãi trong thực tế (Ethernet, 802.11 WiFi, ATM)

←  $d$  bits → ←  $r$  bits →



$$D * 2^r \text{ XOR } R$$

*mathematical formula*

# CRC ví dụ

Muốn:

$$D \cdot 2^r \text{ XOR } R = nG$$

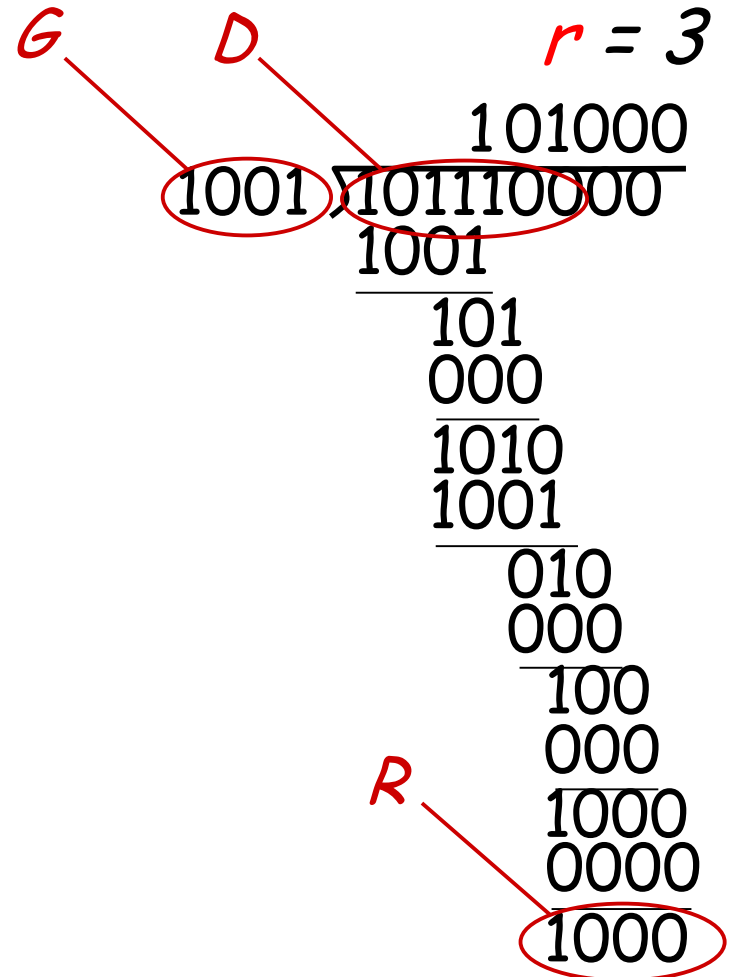
Tương đương:

$$D \cdot 2^r = nG \text{ XOR } R$$

Tương đương:

nếu chúng ta chia  $D \cdot 2^r$  cho  $G$ , có được phần dư  $R$  thỏa:

$$R = \text{remainder}\left[\frac{D \cdot 2^r}{G}\right]$$



# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

5.7 một ngày trong cuộc sống của một yêu cầu web

# Các giao thức và kết nối đa truy cập

2 kiểu “kết nối”:

## ❖ Điểm-điểm (point-to-point)

- PPP cho truy cập dial-up
- Kết nối point-to-point giữa Ethernet switch và host

## ❖ *broadcast (dây hoặc đường truyền được chia sẻ)*

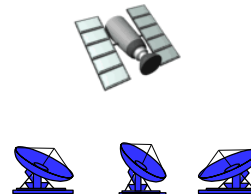
- Ethernet mô hình cũ
- upstream HFC
- 802.11 wireless LAN



shared wire (e.g.,  
cabled Ethernet)



shared RF  
(e.g., 802.11 WiFi)



shared RF  
(satellite)



Trong buổi tiệc cocktail  
(không khí và âm thanh  
được chia sẻ)

# Các giao thức đa truy cập

- ❖ Kênh broadcast được chia sẻ
- ❖ 2 hoặc nhiều việc truyền đồng thời bởi các node: giao thoa
  - *collision* (đụng độ) xảy ra nếu node nhận được 2 hoặc nhiều tín hiệu tại cùng thời điểm

## *Giao thức đa truy cập*

- ❖ Thuật toán phân phối (distributed algorithm) xác định cách các node chia sẻ kênh truyền, nghĩa là xác định khi nào node có thể truyền
- ❖ Truyền thông về kênh truyền chia sẻ phải sử dụng chính kênh đó!
  - Không có kênh khác để phối hợp

# Giao thức đa truy cập lý tưởng

*Cho trước:* kênh broadcast với tốc độ  $R$  bps

*Mong muốn:*

1. Khi 1 node muốn truyền, nó có thể gửi dữ liệu với tốc độ  $R$ .
2. Khi  $M$  node muốn truyền, mỗi node có thể gửi với tốc độ trung bình  $R/M$
3. Phân cấp hoàn toàn:
  - Không có node đặc biệt để các quá trình truyền phối hợp
  - Không đồng bộ các đồng hồ, slots
4. Đơn giản

# Các giao thức MAC: phân loại

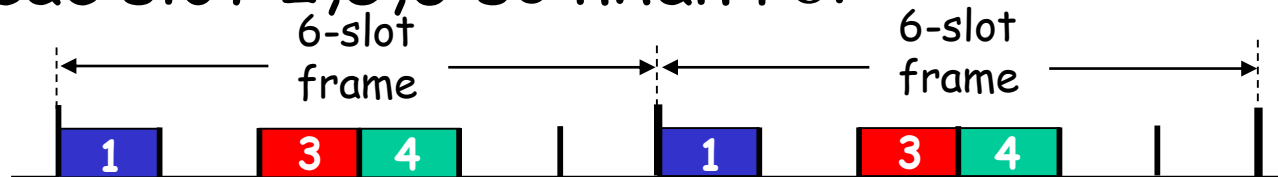
3 loại chính:

- ❖ *Phân hoạch kênh (channel partitioning)*
  - Chia kênh truyền thành “các mảnh” nhỏ hơn (các slot thời gian, tần số, mã)
  - Cấp phát mảnh này cho node để sử dụng độc quyền
- ❖ *Truy cập ngẫu nhiên (random access)*
  - Kênh truyền không được chia, cho phép đụng độ
  - “phục hồi” đụng độ
- ❖ *“xoay vòng”*
  - Các node thay phiên nhau, nhưng các node có quyền nhiều hơn có thể giữ phiên truyền lâu hơn

# Các giao thức MAC phân hoạch kênh: TDMA

## TDMA: time division multiple access

- ❖ Truy cập đến kênh truyền theo hình thức "xoay vòng"
- ❖ Mỗi trạm (station) có slot với độ dài cố định (độ dài = thời gian truyền packet) trong mỗi vòng (round)
- ❖ Các slot không sử dụng sẽ nhàn rỗi
- ❖ Ví dụ: LAN có 6 trạm, 1,3,4 có gói được gửi, các slot 2,5,6 sẽ nhàn rỗi

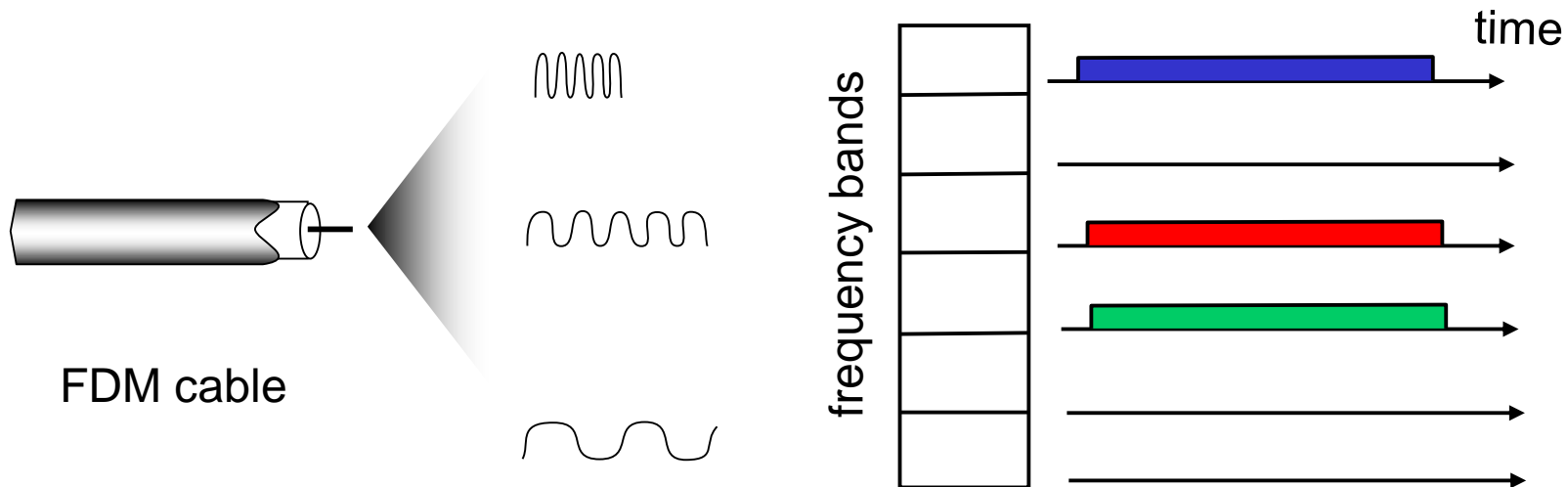




# Các giao thức MAC phân hoạch kênh: FDMA

## FDMA: frequency division multiple access

- ❖ Phổ kênh truyền được chia thành các dải tần số
- ❖ Mỗi trạm được gán một dải tần số cố định
- ❖ Thời gian truyền không được sử dụng trong dải tần số sẽ nhàn rỗi
- ❖ Ví dụ: LAN có 6 station, 1,3,4 có packet truyền, các dải tần số 2,5,6 nhàn rỗi



# Các giao thức truy cập ngẫu nhiên

- ❖ Khi node có packet cần gửi
  - Truyền dữ liệu với trọn tốc độ của kênh dữ liệu  $R$ .
  - Không có sự ưu tiên giữa các node
- ❖ 2 hoặc nhiều node truyền → “đụng độ”,
- ❖ **Giao thức MAC truy cập ngẫu nhiên** xác định:
  - Cách để phát hiện đụng độ
  - Cách để giải quyết đụng độ (ví dụ: truyền lại sau đó)
- ❖ Ví dụ các giao thức MAC truy cập ngẫu nhiên:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

# Slotted ALOHA

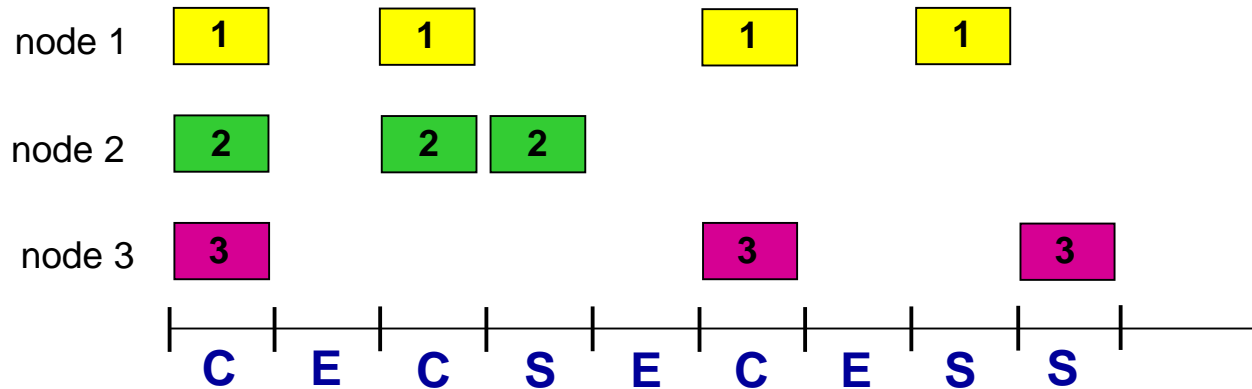
## *Giả thuyết:*

- ❖ Tất cả các frame có cùng kích thước
- ❖ Thời gian được chia thành các slot có kích thước bằng nhau ( thời gian để truyền 1 frame)
- ❖ Các node bắt đầu truyền chỉ ngay tại lúc bắt đầu slot
- ❖ Các node được đồng bộ hóa
- ❖ Nếu 2 hoặc nhiều node truyền trong slot, thì tất cả các node đều phát hiện đụng độ

## *Hoạt động:*

- ❖ Khi node có được frame mới, nó sẽ truyền trong slot kế tiếp
  - *Nếu không có đụng độ:* node có thể gửi frame mới trong slot kế tiếp
  - *Nếu có đụng độ:* node truyền lại frame trong mỗi slot tiếp theo với xác suất  $p$  cho đến khi thành công

# Slotted ALOHA



## *Ưu điểm:*

- ❖ Node đơn kích hoạt có thể truyền liên tục với tốc độ tối đa của kênh
- ❖ Phân cấp cao: chỉ có các slot trong các node cần được đồng bộ
- ❖ Đơn giản

## *Nhược điểm:*

- ❖ Đụng độ, lãng phí slot
- ❖ Các slot nhàn rỗi
- ❖ Các node có thể phát hiện đụng độ trong thời gian ít hơn để truyền packet
- ❖ Đồng bộ hóa

# Slotted ALOHA: hiệu suất

**Hiệu suất:** là phần slot truyền thành công trong số nhiều frame dự định truyền của nhiều node

- ❖ *Giả sử:* có  $N$  node với nhiều frame để truyền, mỗi cái truyền trong slot với xác suất là  $p$
- ❖ Xác suất để given node truyền thành công trong 1 slot =  $p(1-p)^{N-1}$
- ❖ Xác suất mà bất kỳ node nào truyền thành công =  $Np(1-p)^{N-1}$

- ❖ Hiệu suất cực đại: tìm  $p^*$  làm cực đại hóa  $Np(1-p)^{N-1}$
- ❖ Với nhiều node, tìm giới hạn của  $Np^*(1-p^*)^{N-1}$  khi  $N$  tiến tới vô cùng, cho:

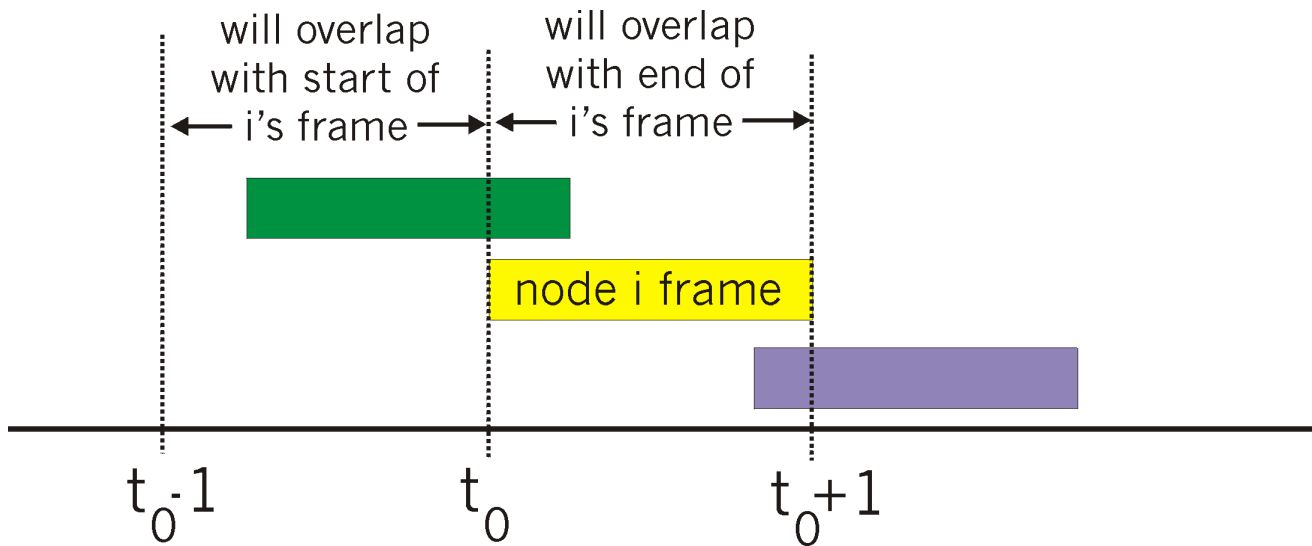
**hiệu suất cực đại =  $1/e = .37$**

**Tốt nhất:** kênh hữu dụng trong khoảng 37% thời gian!



# Pure (unslotted) ALOHA

- ❖ unslotted Aloha: đơn giản, không đồng bộ
- ❖ Khi frame đến đầu tiên
  - truyền lập tức
- ❖ Khả năng đụng độ tăng:
  - frame được truyền tại thời điểm  $t_0$  đụng độ với các frame khác được truyền trong thời điểm  $[t_0-1, t_0+1]$



# Pure ALOHA: hiệu suất

$P(\text{thành công với given node}) = P(\text{node truyền}) \cdot$

$P(\text{không có node khác truyền trong } [t_0-1, t_0])$

$P(\text{không có node khác truyền trong } [t_0-1, t_0])$

$$\begin{aligned} &= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1} \\ &= p \cdot (1-p)^{2(N-1)} \end{aligned}$$

... chọn  $p$  tối ưu và sau đó cho  $n \rightarrow \infty$

$$= 1/(2e) = .18$$

**Thậm chí không tốt bằng slotted Aloha!**

# CSMA (carrier sense multiple access)

*CSMA*: lắng nghe trước khi truyền:

Nếu kênh nhàn rỗi: truyền toàn bộ frame

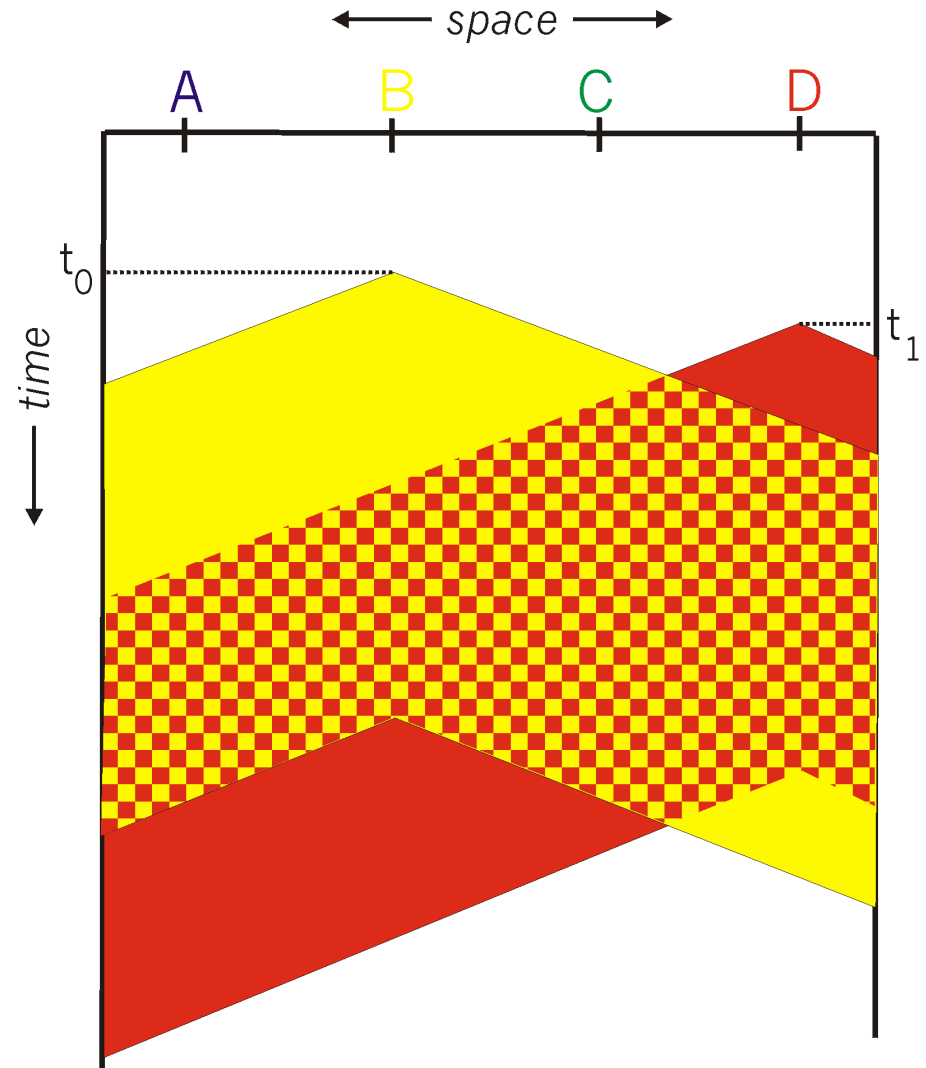
❖ Nếu kênh truyền bận, trì hoãn truyền

❖ So sánh với con người: đừng ngắt lời người khác!



# CSMA: đụng độ (collision)

- ❖ **Đụng độ có thể vẫn xảy ra:** trễ lan truyền nghĩa là 2 node không thể nghe thấy quá trình truyền lẫn nhau
- ❖ **Đụng độ:** toàn bộ thời gian truyền packet bị lãng phí
  - Khoảng cách và trễ lan truyền có vai trò trong việc xác định xác suất đụng độ

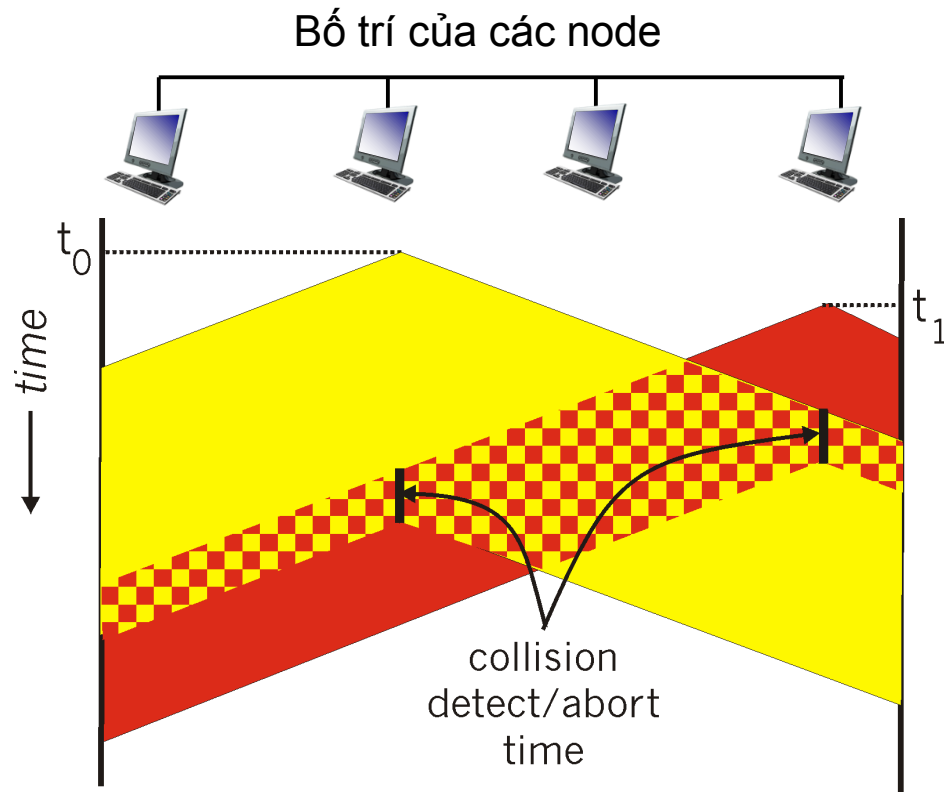


# CSMA/CD (collision detection)

*CSMA/CD*: carrier sensing, trì hoãn như trong CSMA

- Đụng độ được phát hiện trong thời gian ngắn
  - Việc truyền đụng độ được bỏ qua, giảm lãng phí kênh truyền.
- ❖ Phát hiện đụng độ:
- Dễ dàng trong các mạng LAN hữu tuyến: đo cường độ tín hiệu, so sánh với các tín hiệu đã được truyền và nhận
  - Khó thực hiện trong mạng LAN vô tuyến: cường độ tín hiệu được nhận bị áp đảo bởi cường độ truyền cục bộ
- ❖ Tương tự như hành vi của con người: đàm thoại lịch sự

# CSMA/CD (collision detection)



# Thuật toán Ethernet CSMA/CD

1. NIC nhận datagram từ tầng network, tạo frame
2. Nếu NIC cảm nhận được kênh rỗi, nó sẽ bắt đầu việc truyền frame. Nếu NIC cảm nhận kênh bận, đợi cho đến khi kênh rảnh, sau đó mới truyền.
3. Nếu NIC truyền toàn bộ frame mà không phát hiện việc truyền khác, NIC được truyền toàn bộ frame đó!
4. nếu NIC phát hiện có sự truyền khác trong khi đang truyền, thì nó sẽ hủy bỏ truyền và phát tín hiệu tắt nghẽn
5. Sau khi hủy bỏ truyền, NIC thực hiện *binary (exponential) backoff*:
  - Sau lần đụng độ thứ  $m$ , NIC chọn ngẫu nhiên số  $K$  trong khoảng  $\{0, 1, 2, \dots, 2^m - 1\}$ . NIC sẽ đợi  $K \cdot 512$  bit lần, sau đó trở lại bước 2
  - Đụng độ nhiều thì sẽ có khoảng thời gian backoff dài hơn

# CSMA/CD hiệu suất

- ❖  $T_{prop}$  = độ trễ lan truyền lớn nhất (max prop delay) giữa 2 node trong mạng LAN
- ❖  $t_{trans}$  = thời gian để truyền frame có kích thước lớn nhất

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

- ❖ Hiệu suất tiến tới 1
  - khi  $t_{prop}$  tiến tới 0
  - khi  $t_{trans}$  tiến tới vô cùng
- ❖ Hiệu suất tốt hơn ALOHA: đơn giản, rẻ và phân cấp!

# Các giao thức MAC “Xoay vòng”

Các giao thức phân hoạch kênh MAC (channel partitioning MAC protocols):

- Chia sẻ kênh hiệu quả và công bằng với tải lớn
- Không hiệu quả ở tải thấp: trễ khi truy cập kênh,  $1/N$  bandwidth được cấp phát thậm chí khi chỉ có 1 node hoạt động!

Các giao thức MAC truy cập ngẫu nhiên (random access MAC protocols)

- Hiệu quả tại tải thấp: node đơn có thể dùng hết khả năng của kênh
- Tải cao: ùn tắc độ cao

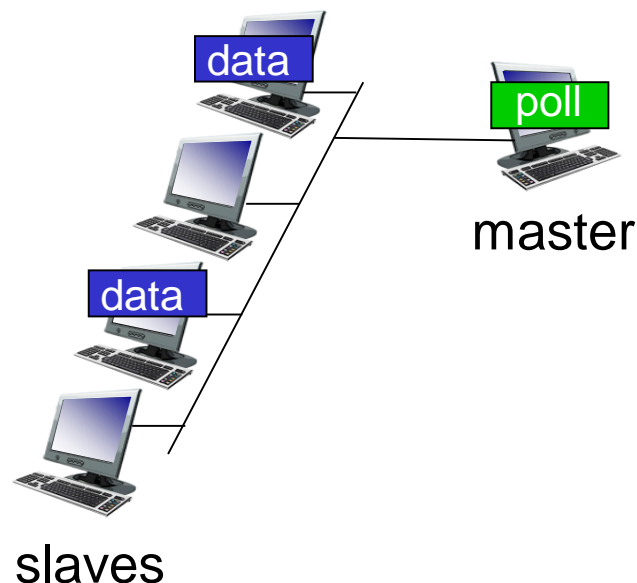
Các giao thức “Xoay vòng” (“taking turns” protocols)

Tìm kiếm giải pháp tốt nhất!

# Các giao thức MAC "Xoay vòng"

## *polling:*

- ❖ Node chủ (master node) "mời" các node con (slave node) truyền lần lượt
- ❖ Thường được sử dụng với các thiết bị con "đần độn"
- ❖ Quan tâm:
  - polling overhead
  - latency
  - Chỉ có 1 điểm chịu lỗi (master)

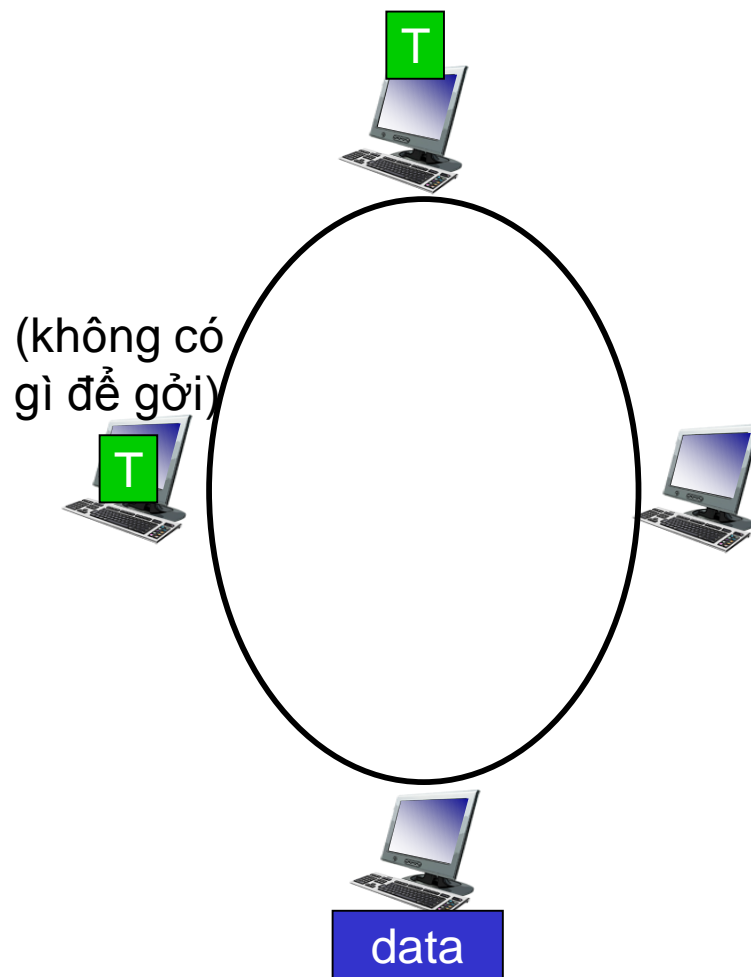


# Các giao thức MAC "Xoay vòng"

## *Chuyển token:*

- ❖ Điều hành *token* được chuyển từ 1 node đến node kế tiếp theo tuần tự.
- ❖ Thông điệp token
- ❖ Quan tâm:
  - token overhead
  - latency

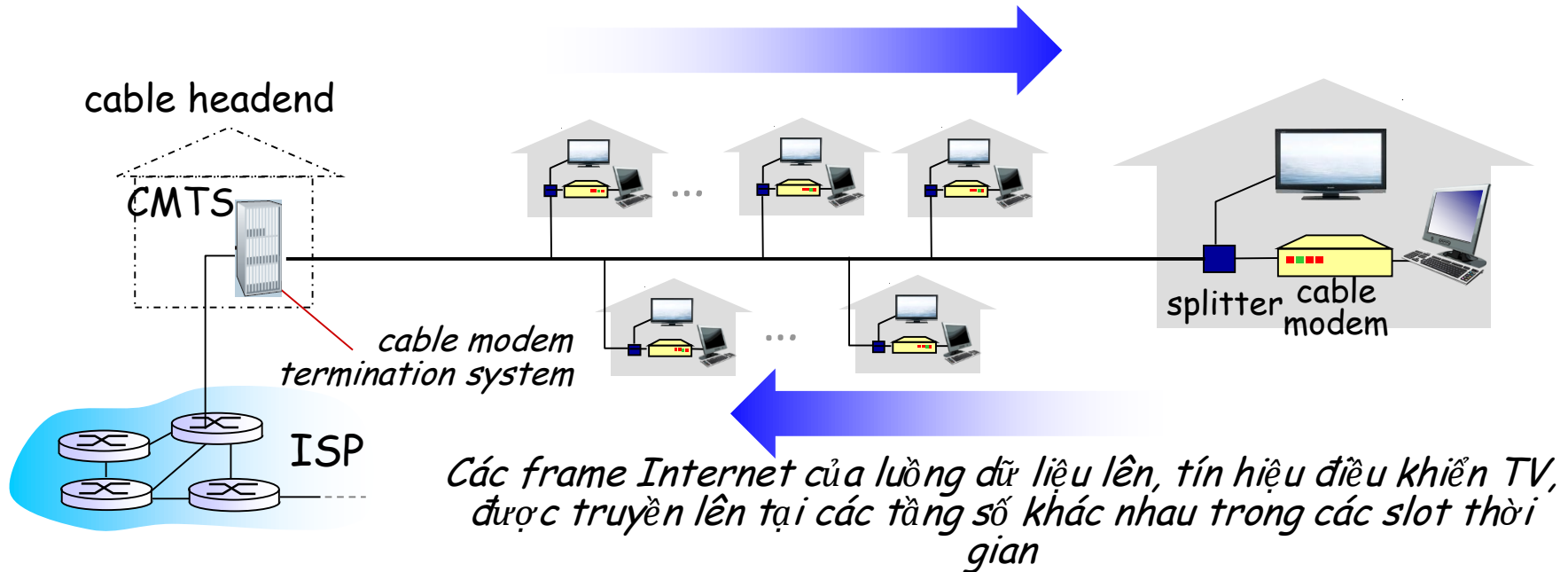
*Chỉ có 1 điểm chịu lỗi  
(master)*





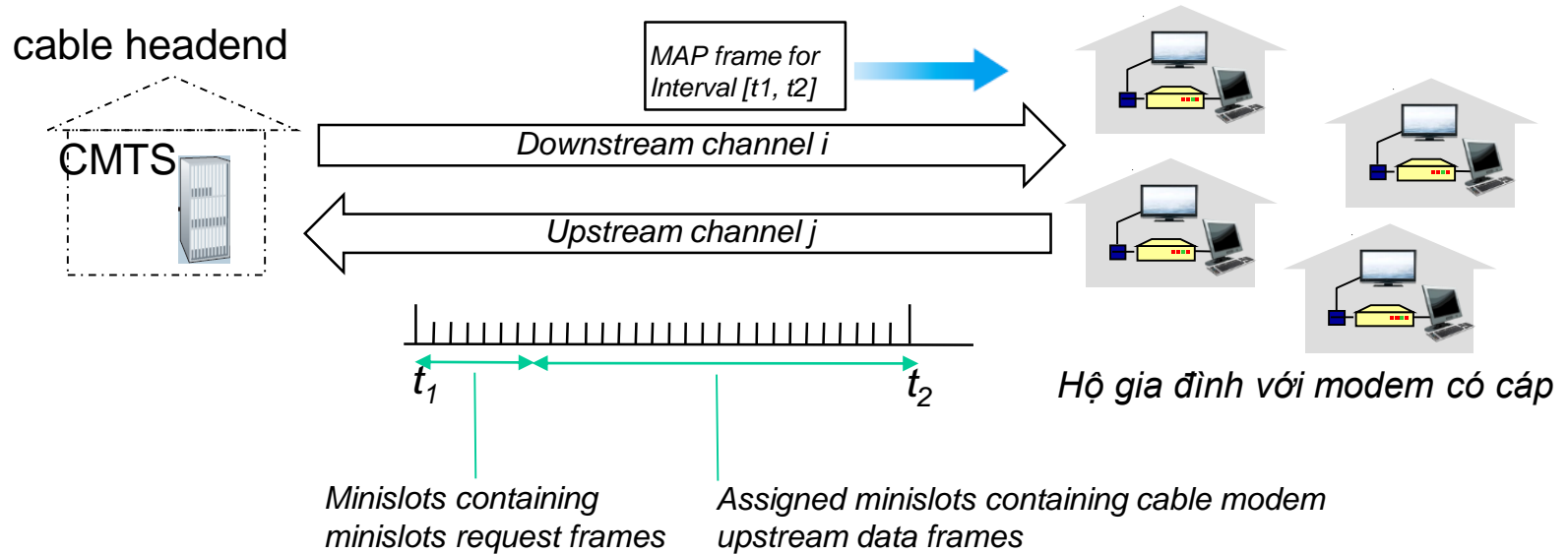
# Mạng truy cập cáp (Cable access network)

*Các frame Internet và kênh TV được truyền xuống với các tần số khác nhau*



- ❖ Nhiều kênh (broadcast) luồng dữ liệu xuống 40Mbps
  - CMTS đơn truyền vào trong các kênh
- ❖ Nhiều kênh luồng dữ liệu lên 30 Mbps
  - **Đa truy cập:** tất cả các user tranh giành các slot thời gian nhất định của kênh luồng dữ liệu lên

# Mạng truy cập cáp



## DOCSIS: data over cable service interface spec

- ❖ FDM trên các kênh tăng số luồng dữ liệu lên và xuống
- ❖ TDM luồng dữ liệu lên (upstream): một số slot được gán, một số slot có tranh chấp
  - MAP frame luồng dữ liệu xuống: gán các slot luồng dữ liệu lên
  - Yêu cầu cho các slot luồng dữ liệu lên (và dữ liệu) được truyền truy cập ngẫu nhiên (binary backoff) trong các slot được lựa chọn

# Tổng kết các giao thức MAC

- ❖ *Phân hoạch kênh*, theo thời gian, tần số hoặc mã
  - Phân chia theo thời gian, phân chia theo tần số
- ❖ *Truy cập ngẫu nhiên*(động),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - Cảm nhận sóng mang (carrier sensing): dễ dàng trong một số kỹ thuật (có dây), khó thực hiện trong các công nghệ khác (không dây)
  - CSMA/CD được dùng trong Ethernet
  - CSMA/CA được dùng trong 802.11
- ❖ *Xoay vòng*
  - polling từ site trung tâm, truyền token
  - bluetooth, FDDI, token ring

# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

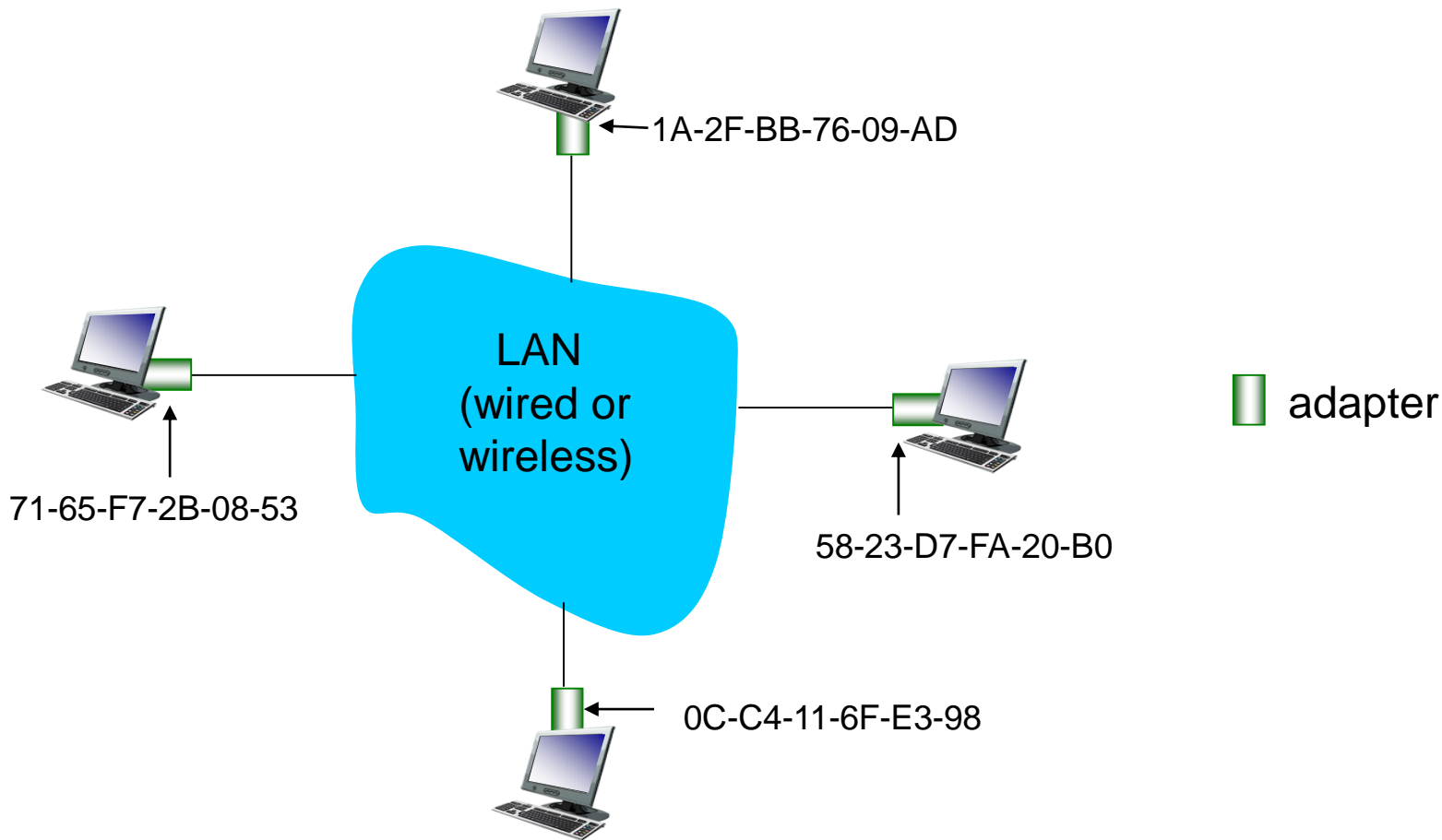
5.7 một ngày trong cuộc sống của một yêu cầu web

# Địa chỉ MAC và ARP

- ❖ Địa chỉ IP 32-bit:
  - *Địa chỉ tầng network* cho interface
  - Được sử dụng cho tầng 3 (tầng network) chuyển dữ liệu
- ❖ Địa chỉ MAC (hoặc LAN hoặc physical hoặc Ethernet):
  - Chức năng: được sử dụng "cục bộ" để chuyển frame từ 1 interface này đến 1 interface được kết nối vật lý với nhau (cùng mạng, trong ý nghĩa địa chỉ IP)
  - Địa chỉ MAC 48 bit (cho hầu hết các mạng LAN) được ghi vào trong NIC ROM, đôi khi cũng trong phần mềm
  - Ví dụ: 1A-2F-BB-76-09-AD
    - hexadecimal (base 16) notation
    - (mỗi "số" đại diện 4 bit)

# Địa chỉ MAC và ARP

Mỗi adapter trên mạng LAN có địa chỉ **LAN** duy nhất

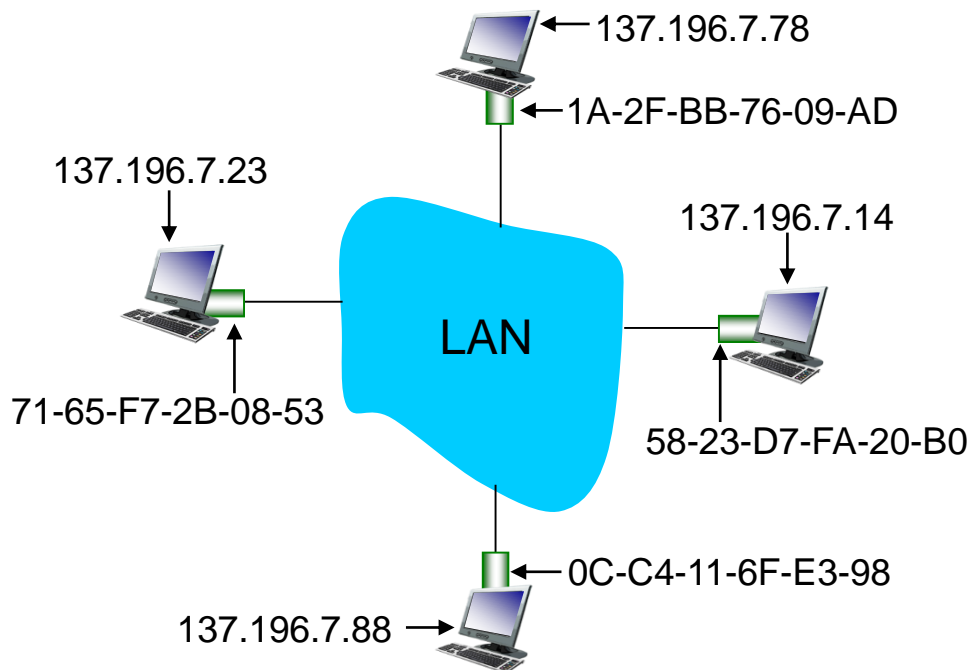


# Địa chỉ LAN(tt)

- ❖ Sự phân bổ địa chỉ MAC được quản lý bởi IEEE
- ❖ Nhà sản xuất mua phần không gian địa chỉ MAC (bảo đảm duy nhất)
- ❖ So sánh:
  - Địa chỉ MAC: như là số chứng minh nhân dân
  - Địa chỉ IP: như là địa chỉ bưu điện
- ❖ địa chỉ MAC phẳng → tính di động
  - Có thể di chuyển card LAN từ 1 mạng LAN này tới mạng LAN khác
- ❖ Địa chỉ IP phân cấp không di động
  - địa chỉ phụ thuộc vào subnet IP mà node đó gắn vào

# ARP: address resolution protocol

*Hỏi:* làm cách nào để xác định địa chỉ MAC của interface khi biết được địa chỉ IP của nó?



*Bảng ARP:* mỗi node IP (host, router) trên mạng LAN có bảng ARP

- Địa chỉ IP/MAC ánh xạ cho các node trong mạng LAN:  
< địa chỉ IP; địa chỉ MAC; TTL >
- TTL (Time To Live): thời gian sau đó địa chỉ ánh xạ sẽ bị lãng quên (thông thường là 20 phút)



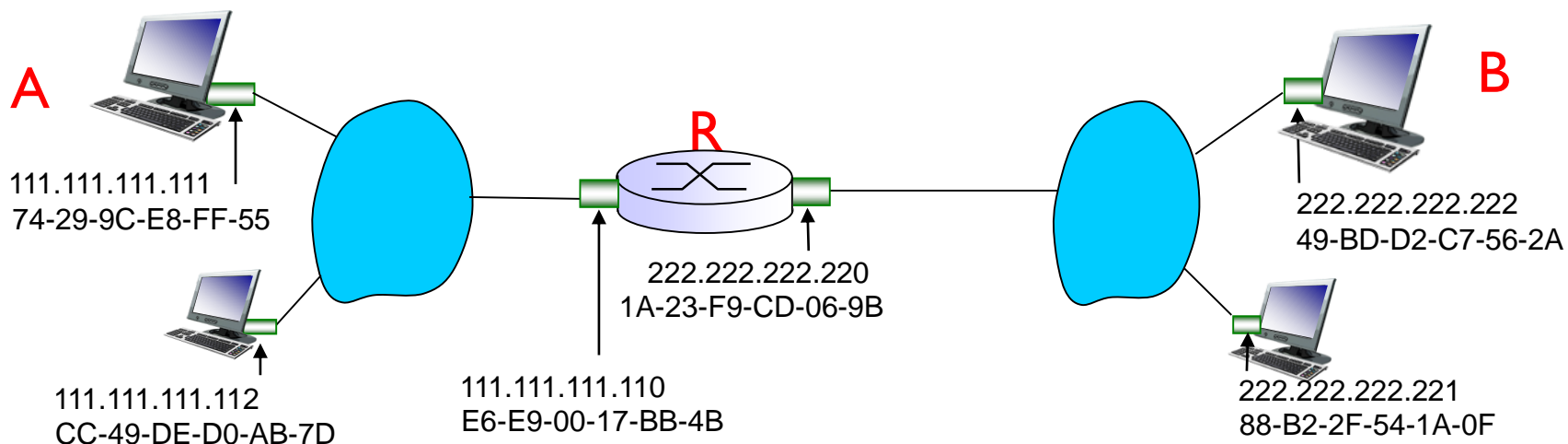
# Giao thức ARP: cùng mạng LAN

- ❖ A muốn gửi datagram tới B
  - Địa chỉ MAC của B không có trong bảng ARP của A.
- ❖ A sẽ **quảng bá (broadcasts)** ARP query packet có chứa địa chỉ IP của B
  - Địa chỉ MAC đích = FF-FF-FF-FF-FF-FF
  - Tất cả các node trên mạng LAN sẽ nhận ARP query này
- ❖ B nhận ARP packet, trả lời tới A với địa chỉ MAC của B
  - frame được gửi tới địa chỉ MAC của A (unicast)
- ❖ A sẽ lưu lại cặp địa chỉ IP-tới-MAC trong bảng ARP của nó cho tới khi thông tin này trở nên cũ (hết hạn sử dụng)
  - soft state: thông tin hết hạn (bỏ đi) trừ khi được làm mới
- ❖ ARP là “plug-and-play”:
  - Các nodes tạo bảng ARP của nó không cần sự can thiệp của người quản trị mạng

# Addressing: định tuyến tới mạng LAN khác

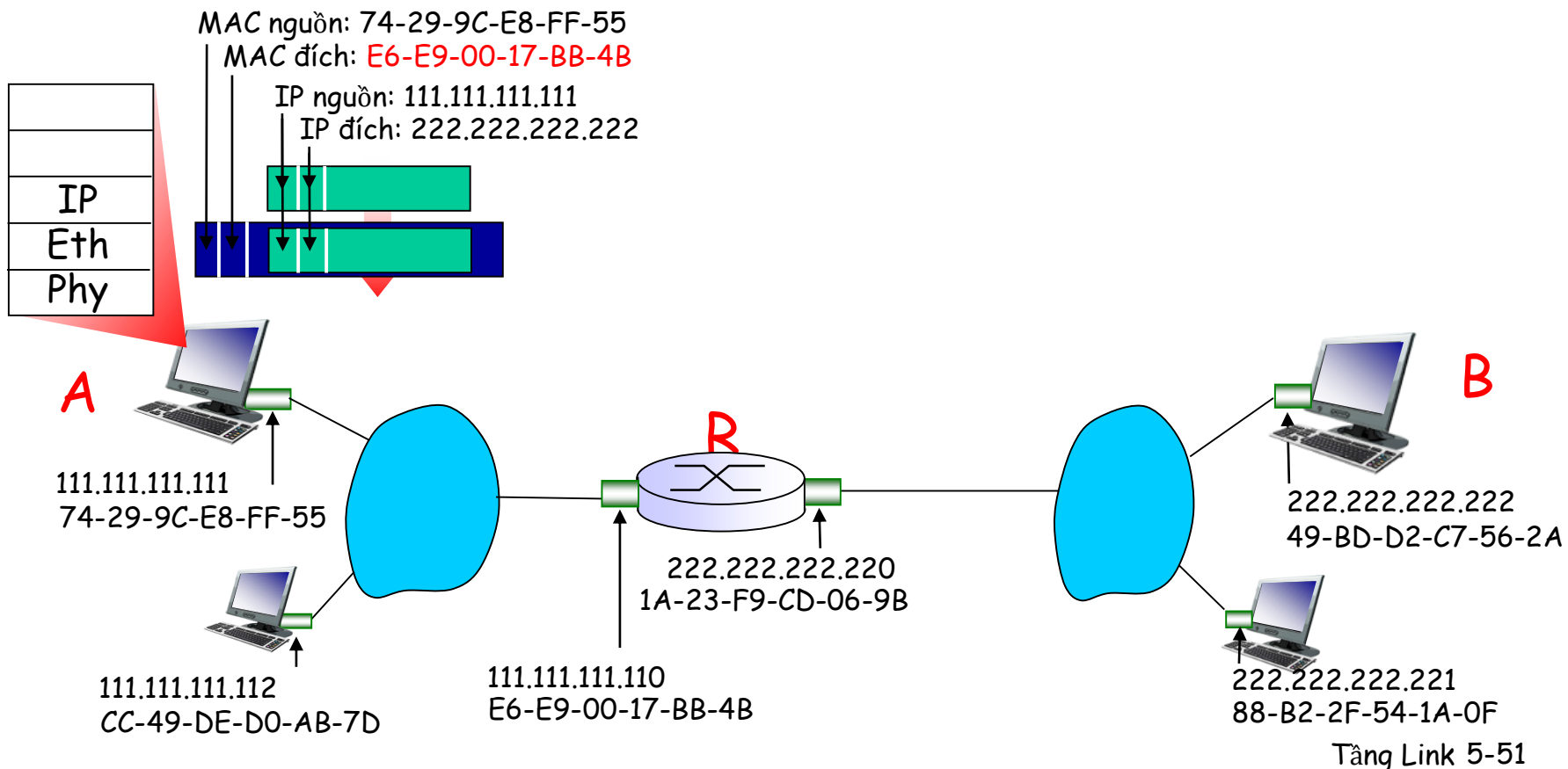
walkthrough: **gửi datagram từ A tới B thông qua R**

- tập trung vào addressing - tại tầng IP (datagram) và MAC (frame)
- giả sử A biết địa chỉ IP của B
- giả sử A biết địa chỉ IP của router first hop, R (cách nào?)
- giả sử A biết địa chỉ MAC của R (cách nào?)



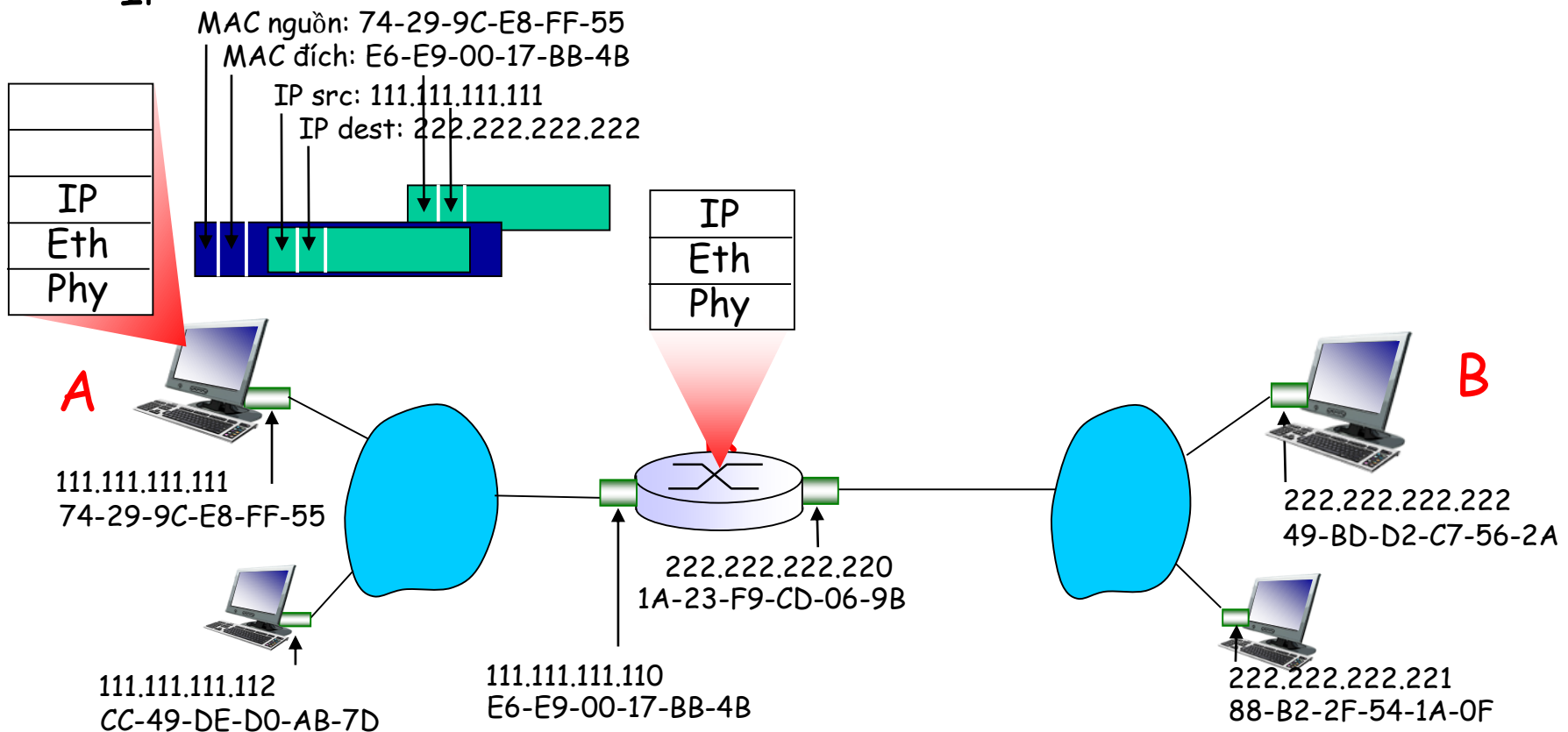
# Addressing: định tuyến tới mạng LAN khác

- ❖ A tạo IP datagram với IP nguồn A, đích B
- ❖ A tạo frame tầng link với địa chỉ MAC của R như là địa chỉ đích, frame này chứa IP datagram từ A tới B



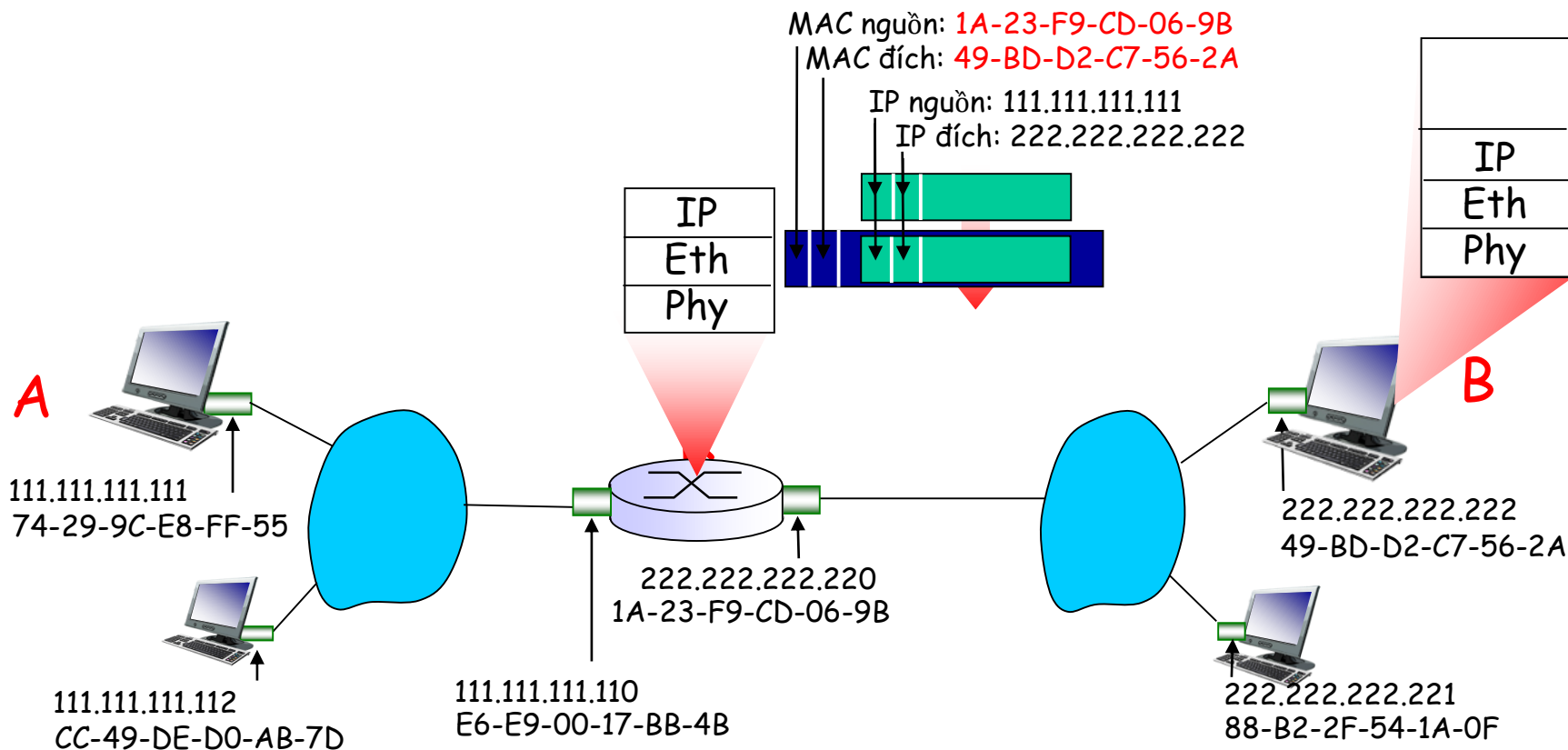
# Addressing: định tuyến tới mạng LAN khác

- ❖ frame được gửi từ A tới R
- ❖ frame được nhận tại R, datagram được gỡ bỏ, được chuyển tới IP



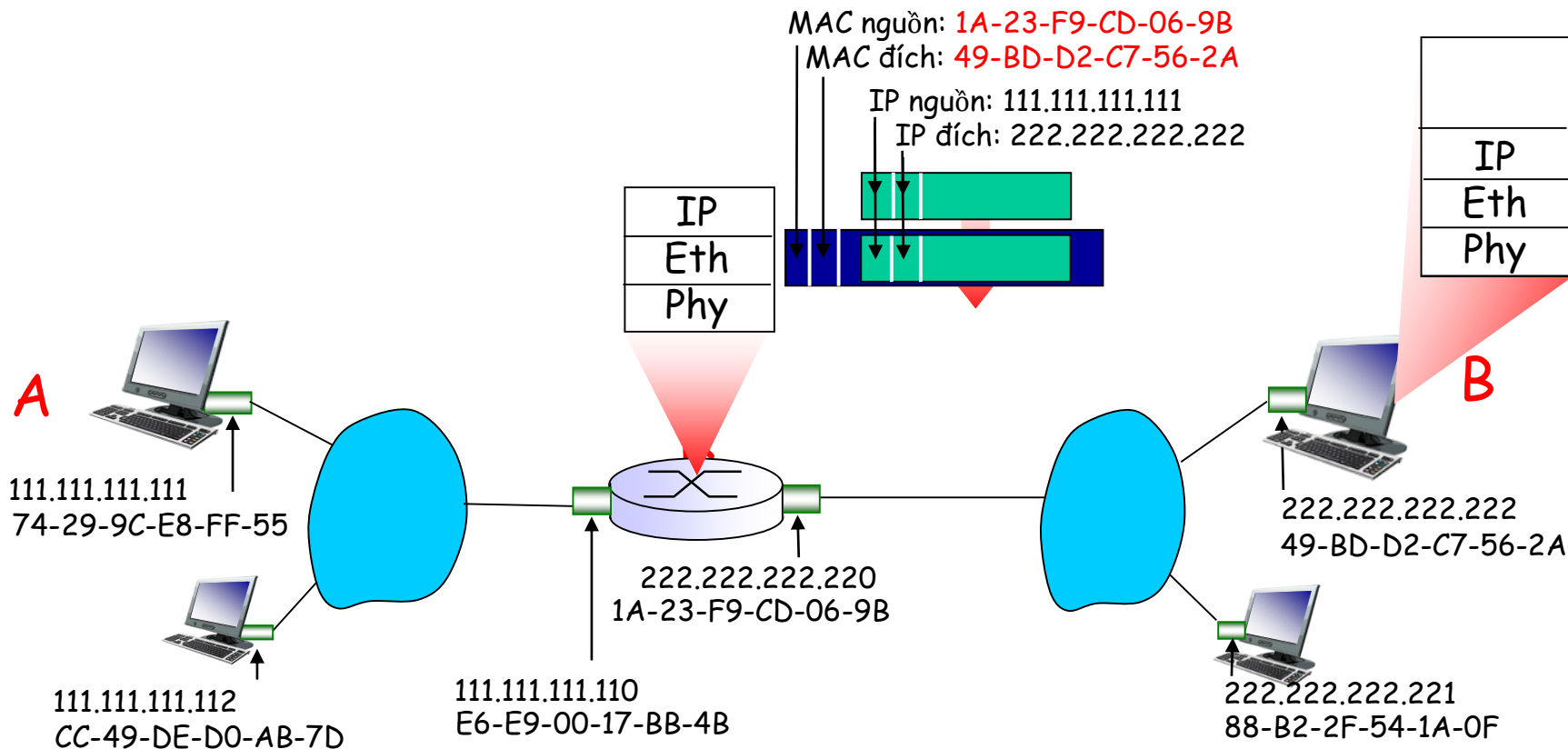
# Addressing: định tuyến tới mạng LAN khác

- ❖ R sẽ chuyển tiếp datagram với IP nguồn A, đích B
- ❖ R tạo frame tầng link với địa chỉ MAC của B như là địa chỉ đích, frame này chứa IP datagram từ A tới B



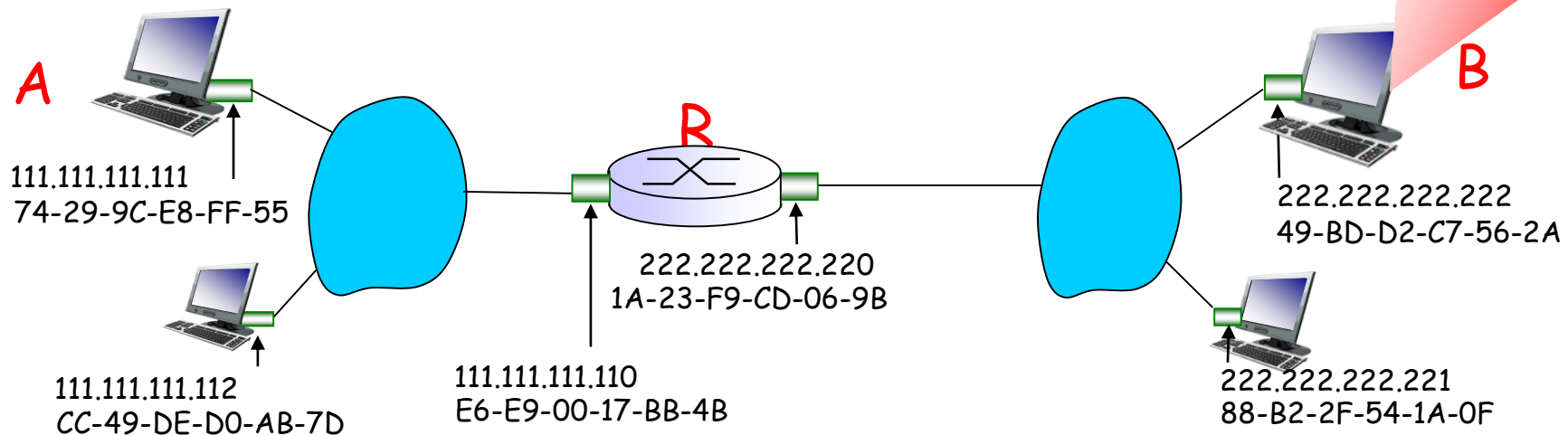
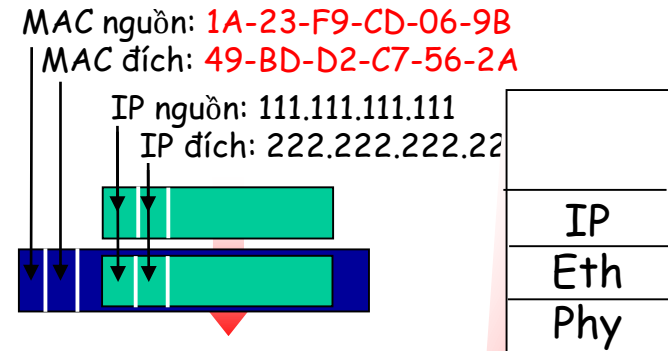
# Addressing: định tuyến tới mạng LAN khác

- ❖ R chuyển tiếp datagram với IP nguồn A, đích B
- ❖ R tạo frame tầng link với địa chỉ MAC của B như là địa chỉ đích, frame này chứa IP datagram từ A-tới-B



# Addressing: định tuyến tới mạng LAN khác

- ❖ R chuyển tiếp datagram với IP nguồn A, đích B
- ❖ R tạo frame tầng link với địa chỉ MAC của B như là địa chỉ đích, frame chứa IP datagram từ A-tới-B



# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

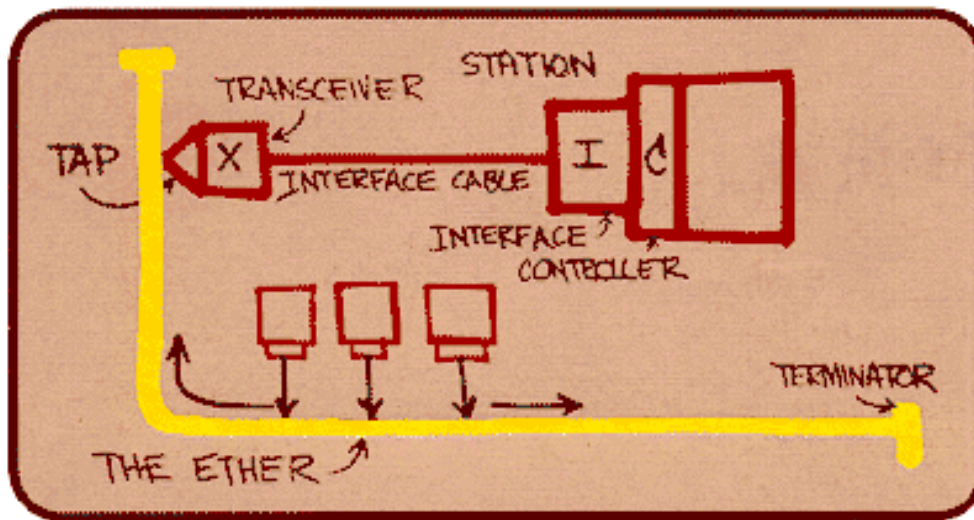
5.7 một ngày trong cuộc sống của một yêu cầu web



# Ethernet

Công nghệ mạng LAN hữu tuyến “chiếm ưu thế”:

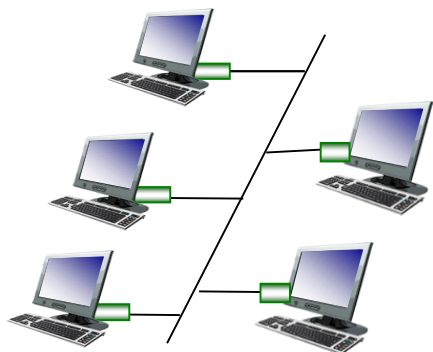
- ❖ \$20 cho NIC
- ❖ Công nghệ mạng LAN được sử dụng rộng rãi lần đầu tiên
- ❖ Đơn giản hơn, rẻ hơn mạng LAN token và ATM
- ❖ Giữ tốc độ trung bình từ: 10 Mbps - 10 Gbps



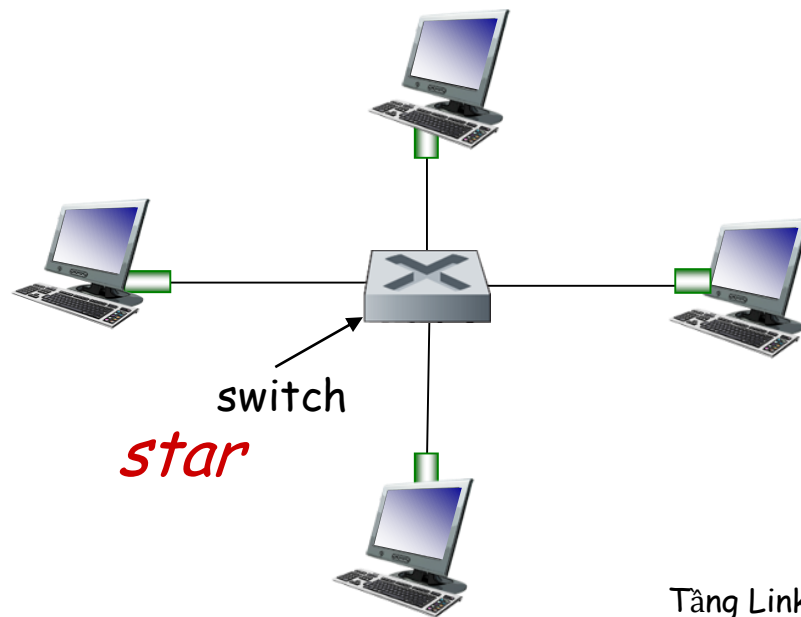
*Phác thảo Ethernet của Metcalfe*

# Ethernet: cấu trúc vật lý

- ❖ *bus*: phổ biến trong giữa thập niên 90
  - Tất cả các node trong cùng collision domain (có thể đụng độ lẫn nhau)
- ❖ *star*: chiếm ưu thế ngày nay
  - *switch* hoạt động ở trung tâm
  - Mỗi “spoke” chạy một (riêng biệt) giao thức Ethernet (các node không đụng độ lẫn nhau)



*bus*: cáp đồng trục



# Cấu trúc frame Ethernet

adapter gửi sẽ đóng gói IP datagram (hoặc packet giao thức khác của tầng mạng) trong **Ethernet frame**



## **preamble:**

- ❖ 7 byte với mẫu 10101010 được theo sau bởi 1 byte với mẫu 10101011
- ❖ được sử dụng để đồng bộ tốc độ đồng hồ của người gửi và nhận

# Cấu trúc frame Ethernet (tt)

- ❖ **addresses:** 6 byte địa chỉ MAC nguồn, đích
  - Nếu adapter nhận frame với địa chỉ đích đúng là của nó, hoặc với địa chỉ broadcast (như là ARP packet), thì nó sẽ chuyển dữ liệu trong frame tới giao thức tầng network
  - Ngược lại, adapter sẽ hủy frame
- ❖ **type:** chỉ ra giao thức tầng cao hơn (thường là IP nhưng cũng có thể là những cái khác như là Novell IPX, AppleTalk)
- ❖ **CRC:** cyclic redundancy check tại bên nhận
  - Lỗi được phát hiện: frame bị bỏ



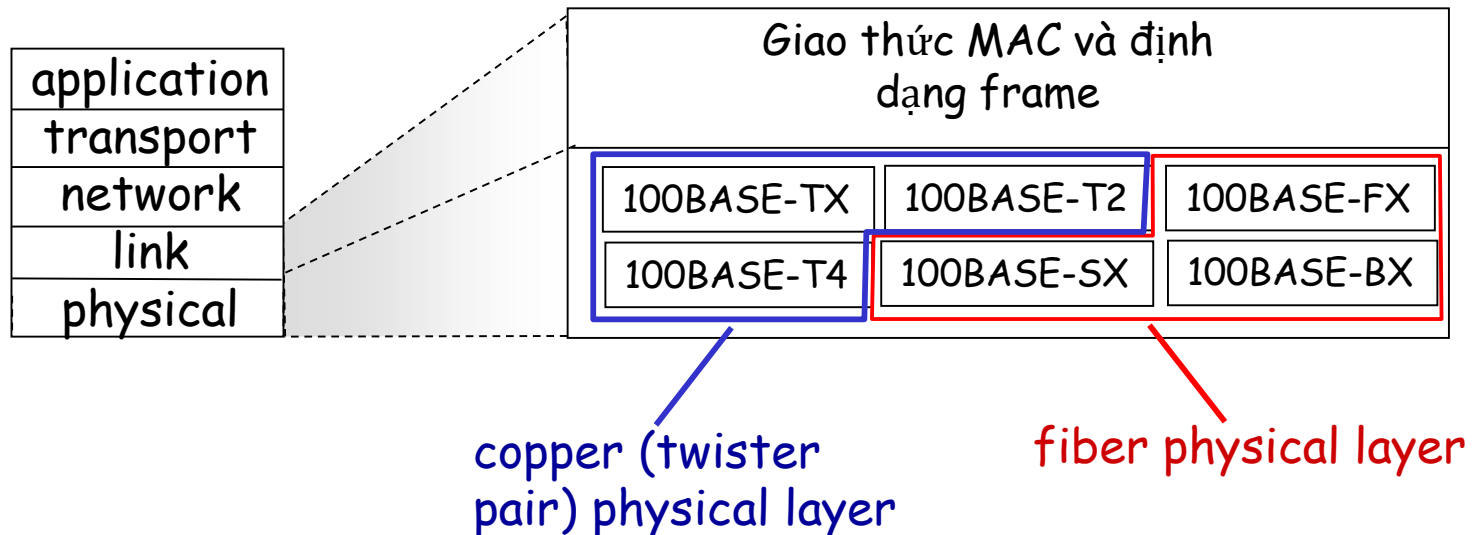
# Ethernet: không tin cậy, không kết nối

- ❖ *Connectionless (không kết nối)*: không bắt tay giữa các NIC gửi và nhận
- ❖ *Unreliable (không tin cậy)*: NIC nhận sẽ không gửi thông báo nhận thành công (acks) hoặc không thành công (nacks) đến các NIC gửi
  - Dữ liệu bị trong các frame bị bỏ sẽ được khôi phục lại chỉ khi nếu bên gửi dùng dịch vụ tin cậy của tầng cao hơn (như là TCP) còn không thì dữ liệu mà đã bị bỏ sẽ mất luôn
- ❖ Giao thức MAC của Ethernet: *CSMA/CD với binary backoff*

# Chuẩn Ethernet 802.3 : tầng link & physical

## ❖ *Nhiều* chuẩn Ethernet khác nhau

- Giao thức MAC thông dụng và định dạng frame
- Tốc độ khác nhau: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
- Môi trường truyền tầng vật lý khác nhau: fiber, copper và wireless



# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

5.7 một ngày trong cuộc sống của một yêu cầu web

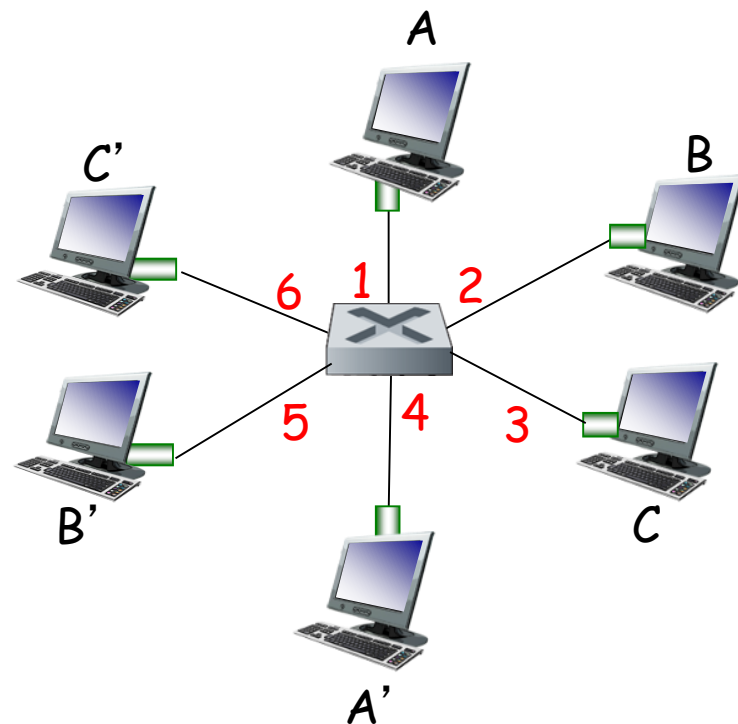
# Ethernet switch

- ❖ **Thiết bị tầng link: giữa vai trò tích cực**
  - Lưu (store) và chuyển tiếp (forward ) các frame Ethernet
  - Xem xét địa chỉ MAC của frame đến, **chọn lựa** chuyển tiếp frame tới 1 hay nhiều đường link đi ra khi frame được chuyển tiếp vào segment, dùng CSMA/CD để truy nhập segment
- ❖ **Transparent (trong suốt)**
  - Các host không nhận thức được sự hiện diện của các switch
- ❖ **plug-and-play, tự học**
  - Các switch không cần được cấu hình



# Switch: nhiều sự truyền đồng thời

- ❖ Các host kết nối trực tiếp tới switch
- ❖ Switch lưu tạm (buffer) các packet
- ❖ Giao thức Ethernet được sử dụng trên *mỗi* đường kết nối vào, nhưng không có đụng độ; full duplex
  - Mỗi đường kết nối là 1 miền collision (đụng độ) của riêng nó
- ❖ *switching*: A-tới-A' và B-tới-B' có thể truyền đồng thời mà không có đụng độ xảy ra



*switch với 6 interface  
(1,2,3,4,5,6)*

# Bảng switch forwarding

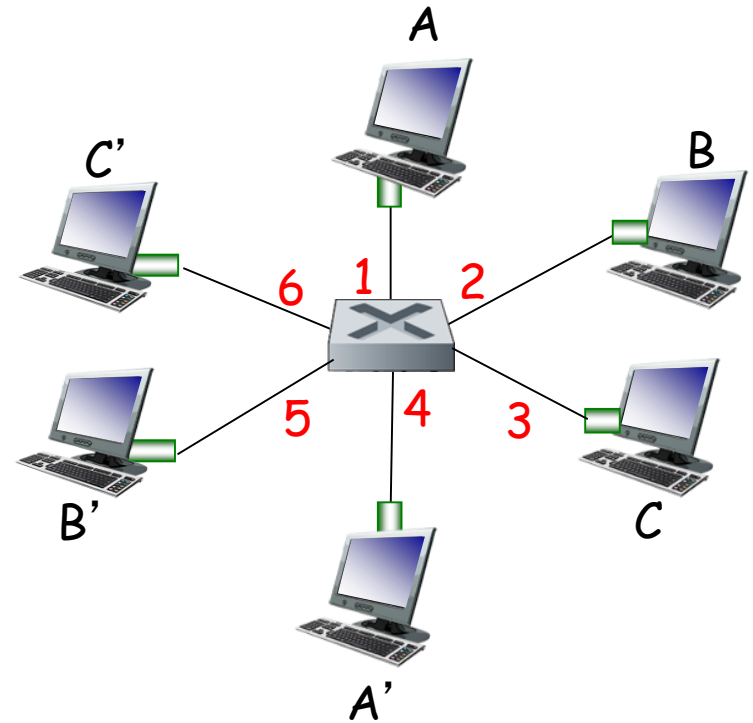
Hỏi: làm thế nào để switch biết tới A' thì sẽ thông qua interface 4 và tới B' thì thông qua interface 5?

❖ Đáp: mỗi switch có *một bảng switch*, mỗi entry:

- (địa chỉ MAC của host, interface để tới được host đó, time stamp)
- Giống như bảng định tuyến!

Q: những entry được tạo và được duy trì như thế nào trong bảng switch?

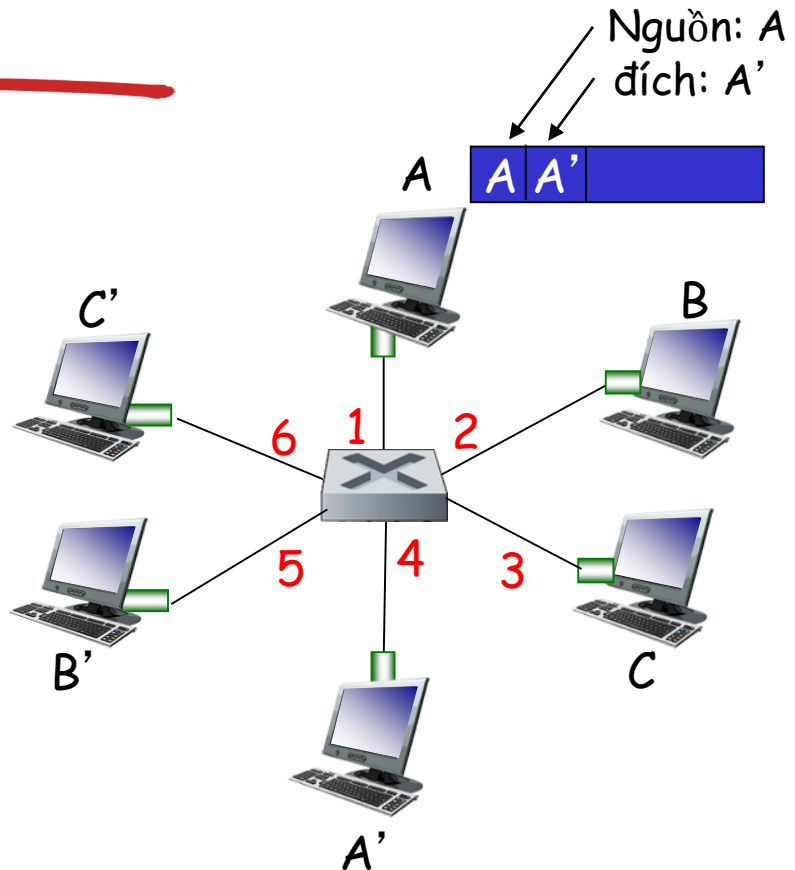
- Có giống như giao thức định tuyến hay không?



switch với 6 interface  
(1,2,3,4,5,6)

# Switch: tự học

- ❖ switch *học* các host có thể tới được thông qua các interface kết nối với các host đó
  - Khi frame được nhận, switch “học” vị trí của bên gửi: incoming LAN segment
  - Ghi lại cặp bên gửi/vị trí trong bảng switch



MAC addr	interface	TTL
A	1	60

*Bảng Switch  
(ban đầu trống)*

# Switch: lọc/chuyển tiếp frame

Khi frame được nhận tại switch:

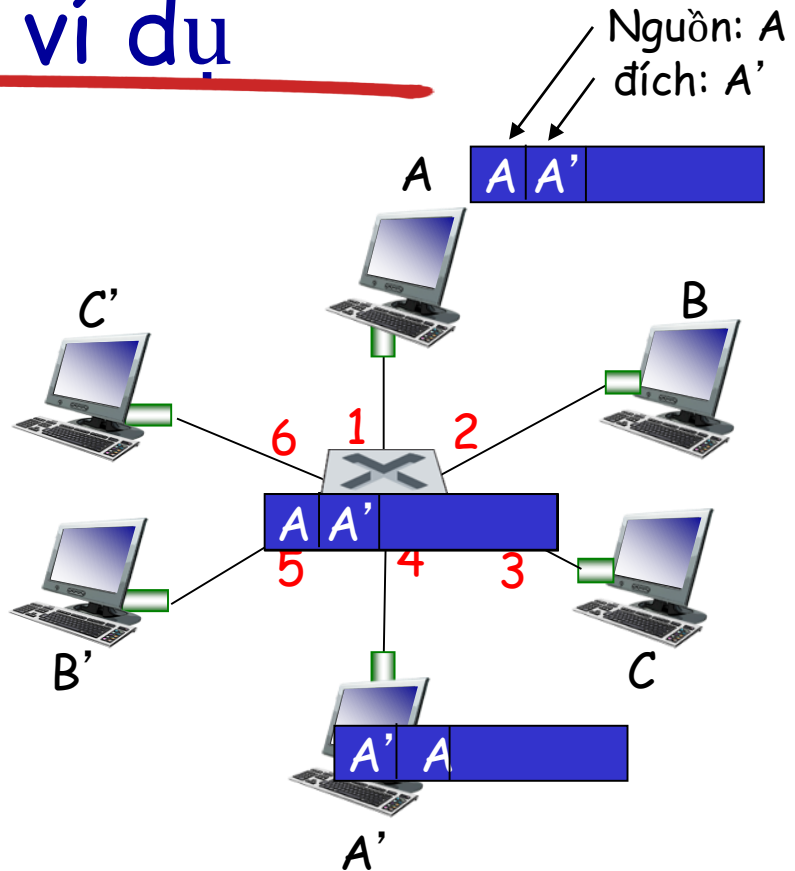
1. Ghi lại đường kết nối vào, địa chỉ MAC của host gửi
2. Ghi vào mục lục bảng switch với địa chỉ MAC đích
3. Nếu entry được thì thấy cho đích đến đó  
thì {  
    nếu đích đến nằm trên phân đoạn mạng từ cái mà  
    frame đã đến  
    thì bỏ frame  
    ngược lại chuyển tiếp frame trên interface  
    được chỉ định bởi entry  
}  
    ngược lại flood /\* chuyển tiếp trên tất cả  
    interface ngoại trừ interface mà dữ liệu đó đã đến  
    từ đó\*/

# Tự học, chuyển tiếp: ví dụ

❖ frame có đích đến là  $A'$ , vị trí của  $A'$  không biết: *flood*

❖ Đích  $A$  có vị trí đã được biết trước:

*gửi chọn lọc chỉ trên 1 đường kết nối duy nhất*

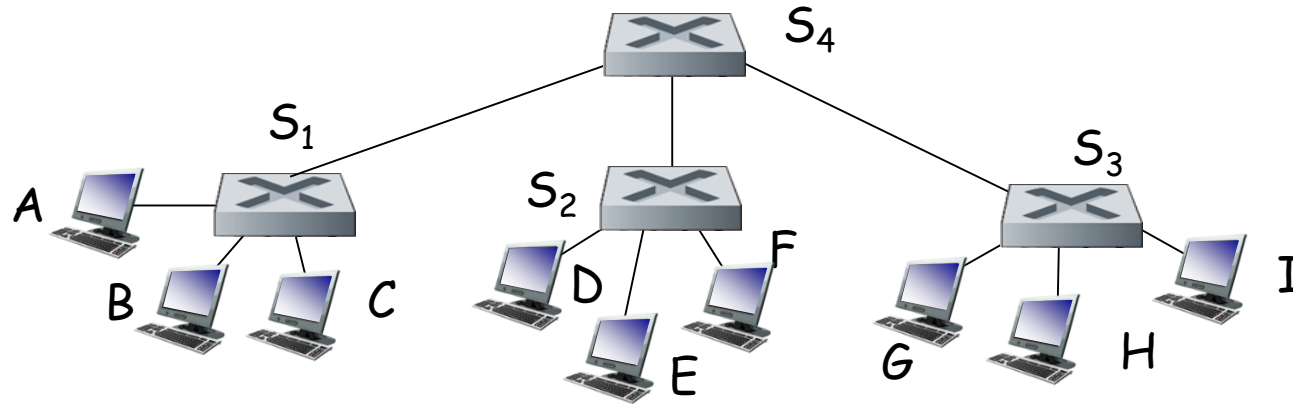


MAC addr	interface	TTL
$A$	1	60
$A'$	4	60

*Bảng switch  
(ban đầu trống)*

# Kết nối các switch với nhau (Interconnecting switches)

- ❖ Các switch có thể được kết nối với nhau

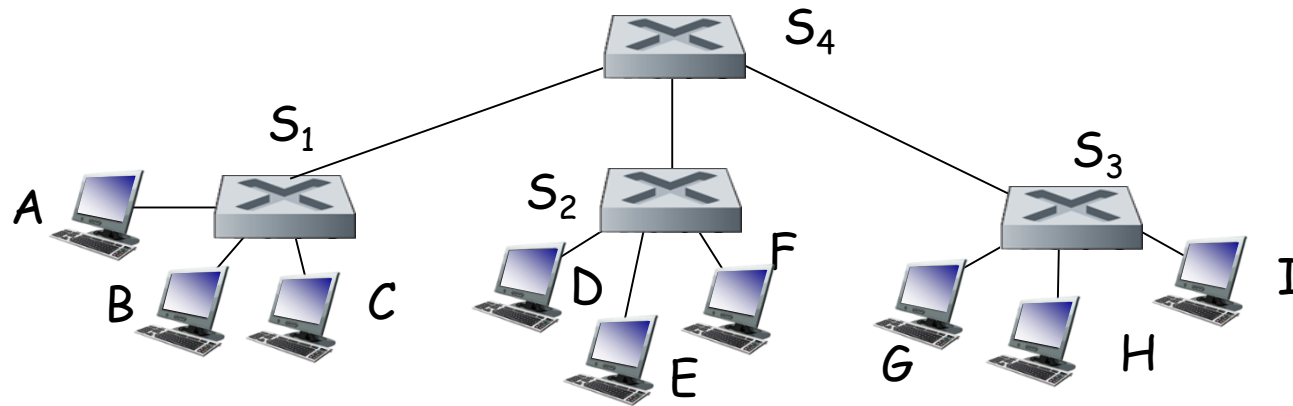


Hỏi: gửi từ A tới G - làm cách nào S<sub>1</sub> biết đề chuyển tiếp frame tới F thông qua S<sub>4</sub> và S<sub>3</sub>?

- ❖ Trả: tự học! (làm việc giống *y chang* như trong trường hợp chỉ có 1 switch!)

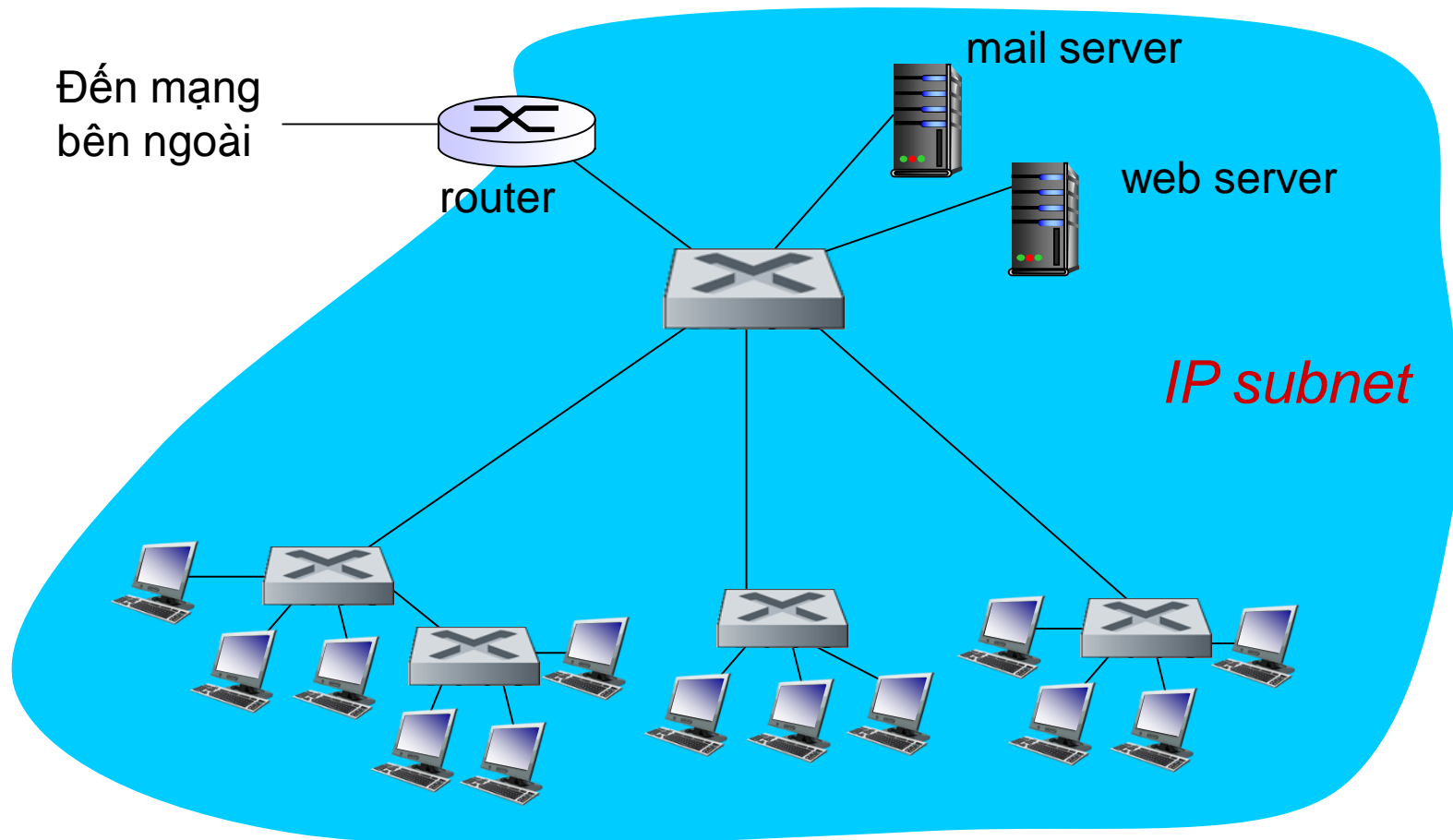
# Ví dụ nhiều switch tự học

Giả sử C gửi frame tới I, I trả lời cho C



- ❖ Hỏi: trình bày các bảng của các switch và cách packet được chuyển đi tại các switch  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$

# Mạng của tổ chức





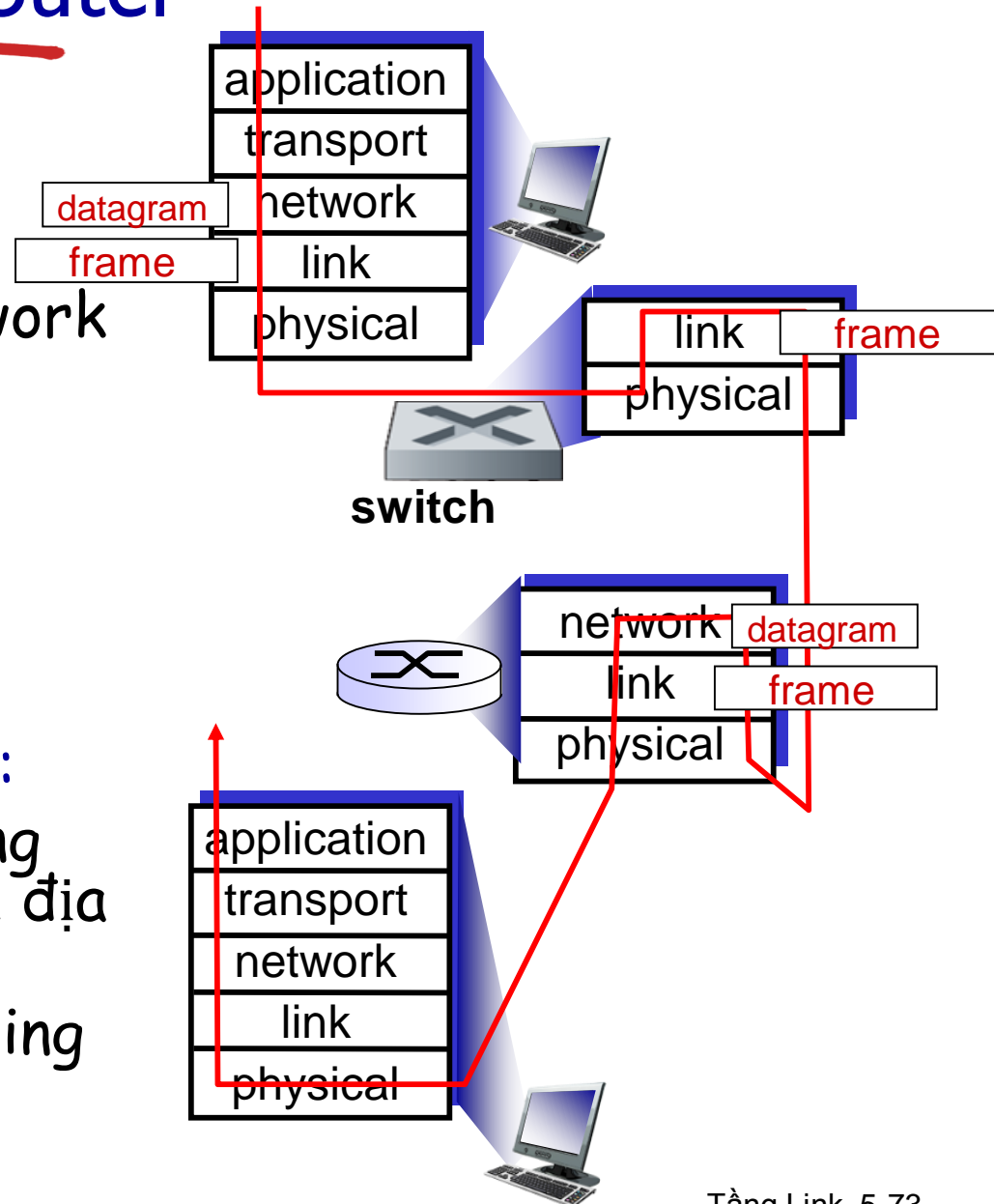
# So sánh Switch và router

Cả 2 đều lưu và chuyển tiếp (store-and-forward):

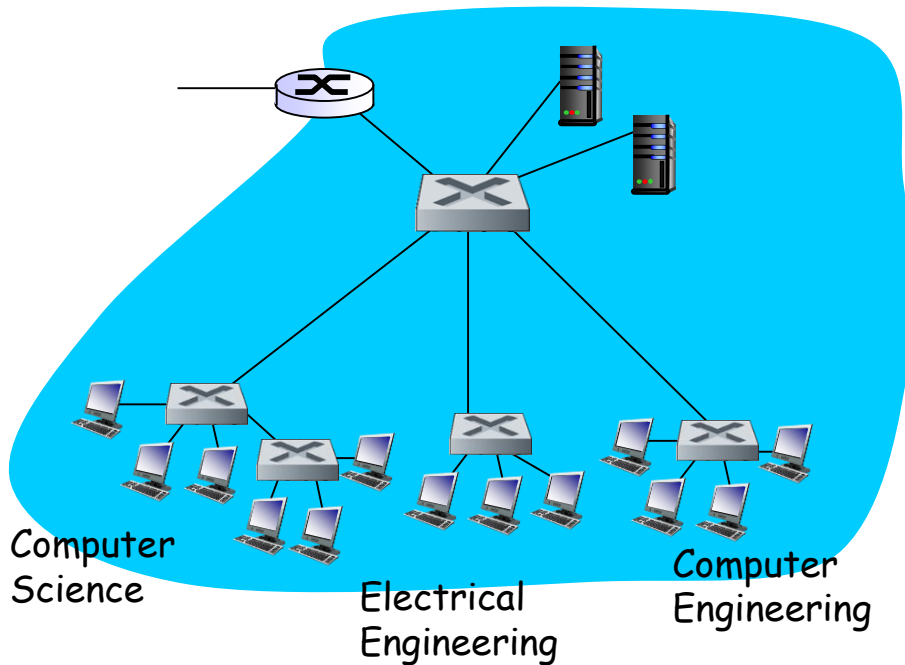
- **router**: thiết bị tầng network (khảo sát header của tầng network)
- **switch**: thiết bị tầng link (khảo sát header của tầng link)

Cả 2 đều có bảng forwarding:

- **router**: tính toán bảng dùng các thuật toán định tuyến, địa chỉ IP
- **switch**: học bảng forwarding dùng flooding, học, địa chỉ MAC



# VLANs: trình bày



## *Xem xét:*

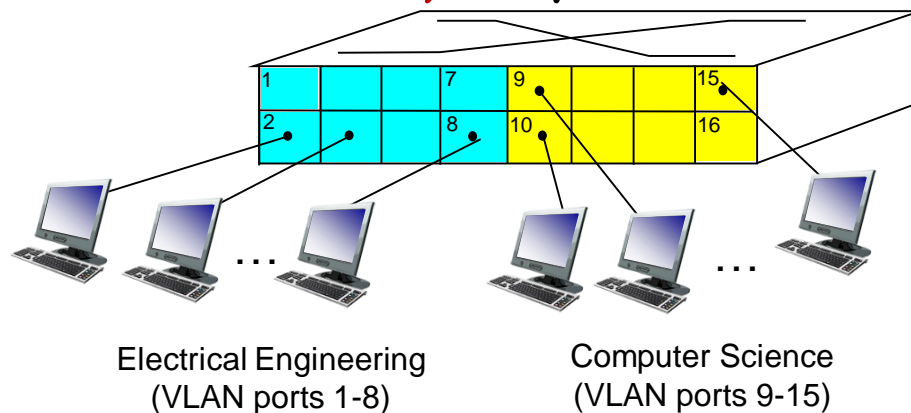
- ❖ Người dùng bên CS di chuyển văn phòng sang EE, nhưng vẫn muốn kết nối CS switch?
- ❖ Miền broadcast đơn:
  - Tất cả lưu lượng broadcast tầng 2 (ARP, DHCP, địa chỉ MAC không biết vị trí đích đến ở đâu) phải đi qua toàn mạng LAN
  - An ninh/riêng tư, các vấn đề về hiệu suất

# VLANs

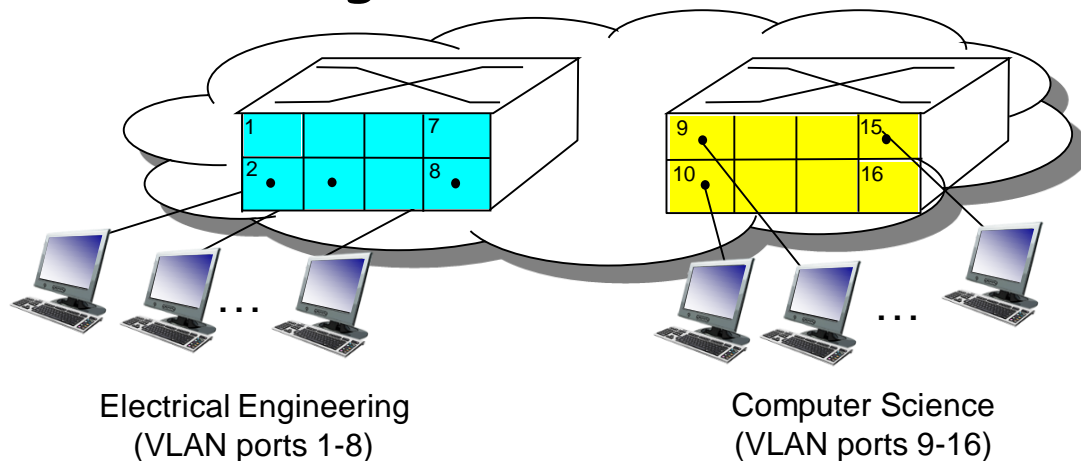
## **Virtual Local Area Network**

Các switch hỗ trợ khả năng VLAN có thể được cấu hình để định nghĩa nhiều mạng LAN ảo (multiple virtual LANS) trên một hạ tầng vật lý của mạng LAN.

**port-based VLAN:** các port của switch được nhóm lại (bởi phần mềm quản lý switch) để trở thành một switch *vật lý* duy nhất...

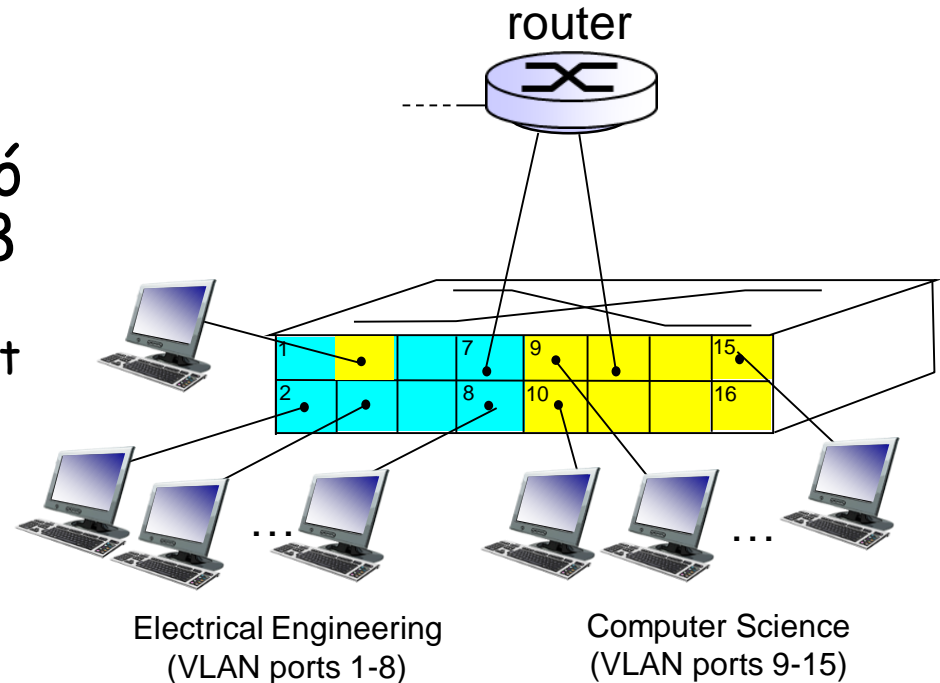


...hoạt động như là *nhiều* switch ảo

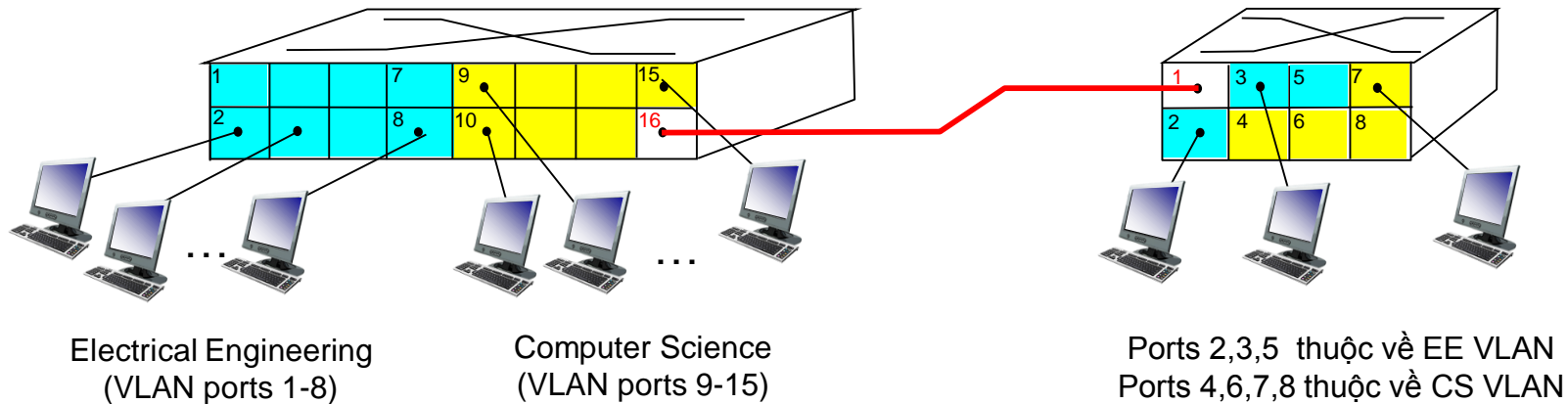


# Port-based VLAN

- ❖ *traffic isolation (cô lập traffic)*: các frame đến/từ các port 1-8 chỉ có thể tới được các port 1-8
  - Cũng có thể định nghĩa VLAN dựa trên địa chỉ MAC của thiết bị đầu cuối, hơn là dựa trên port của switch
- ❖ *dynamic membership*: các port có thể được gán động giữa các VLAN
- ❖ *Chuyển tiếp giữa các VLAN*: được thực hiện thông qua định tuyến (cũng giống như các switch riêng biệt)
  - Trên thực tế, các nhà cung cấp bán các thiết bị switch kết hợp với các router

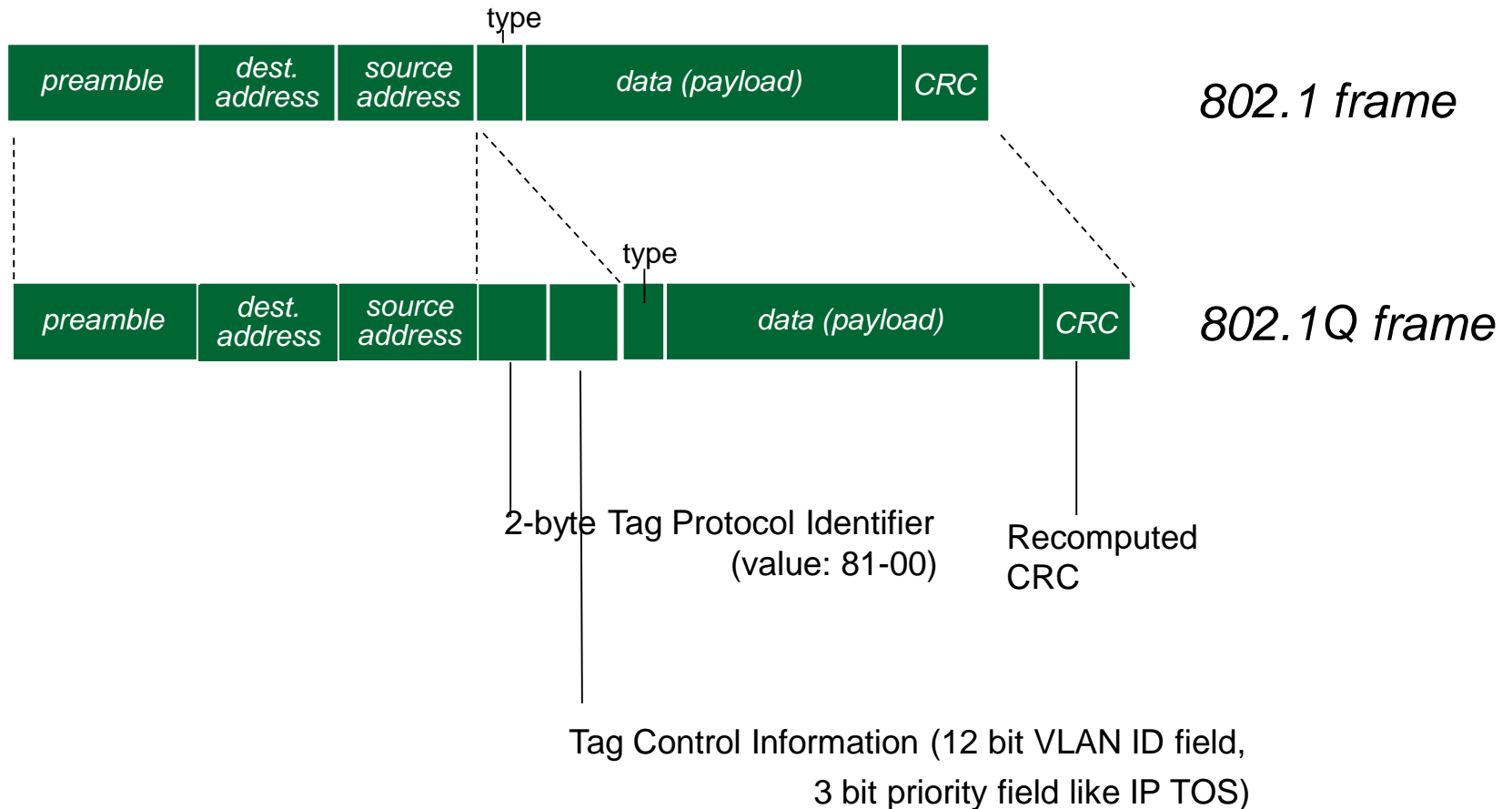


# VLANs kéo dài qua nhiều switch



- ❖ **trunk port:** mang các frame giữa các VLAN được định nghĩa trên nhiều switch vật lý
  - Các frame được chuyển tiếp bên trong VLAN giữa các switch không thể là các frame 802.1 (phải mang thông tin VLAN ID)
  - Giao thức 802.1q thêm/gỡ bỏ các trường header được thêm vào cho các frame được chuyển tiếp giữa các trunk port

# Định dạng frame VLAN 802.1Q



# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

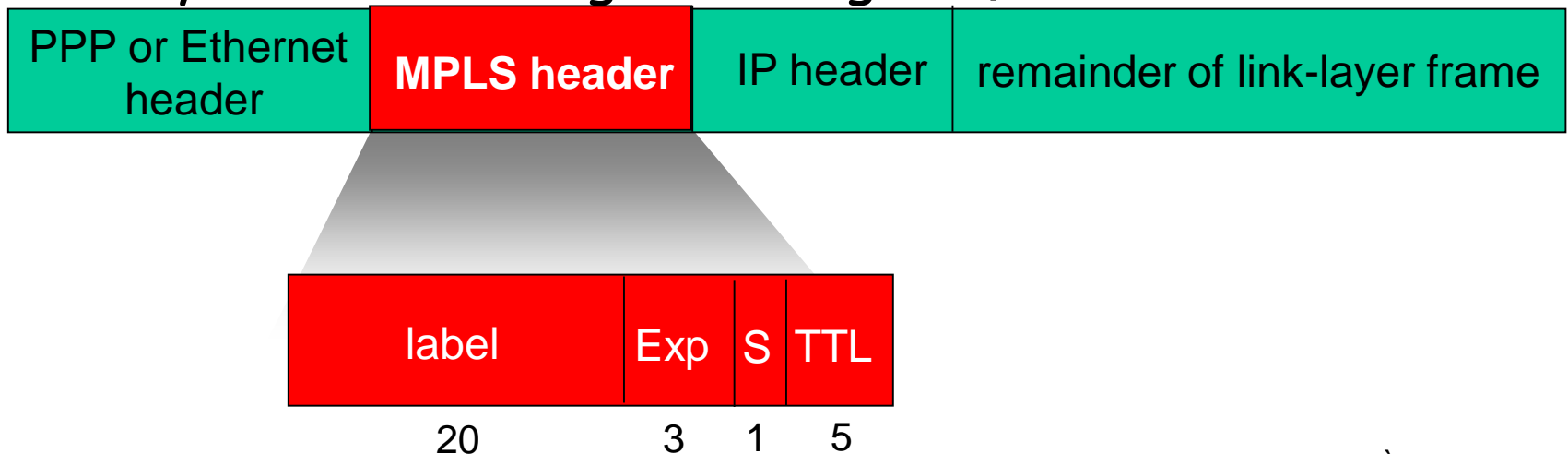
5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

5.7 một ngày trong cuộc sống của một yêu cầu web

# Multiprotocol label switching (MPLS)

- ❖ Mục tiêu ban đầu: chuyển tiếp IP tốc độ cao dùng nhãn có độ dài cố định (fixed length label) (thay thế cho địa chỉ IP)
  - Tra cứu nhanh dùng định dạng có chiều dài cố định (fixed length identifier) (chứ không dùng sự phù hợp với prefix ngắn nhất)
  - Lấy ý tưởng từ hướng tiếp cận của Virtual Circuit (VC)
  - Tuy nhiên IP datagram vẫn giữ địa chỉ IP!

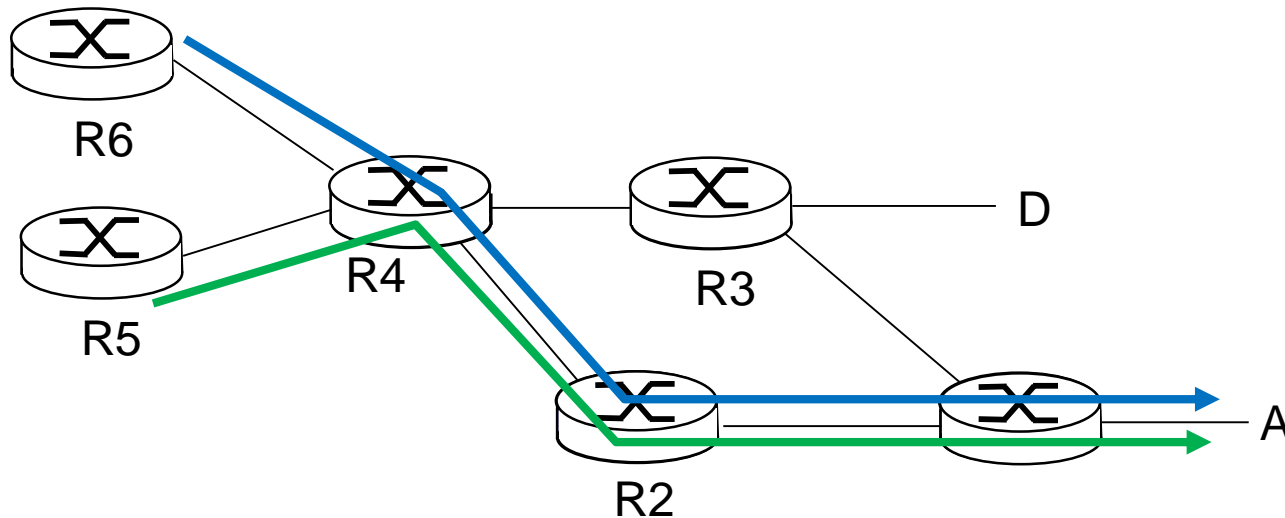




# Router có khả năng MPLS

- ❖ Còn gọi là Router chuyển mạch nhãn (label-switched router)
- ❖ Chuyển tiếp các packet tới interface đầu ra chỉ dựa trên giá trị nhãn (label value) (*không kiểm tra địa chỉ IP*)
  - Bảng chuyển tiếp MPLS (MPLS forwarding table) khác với bảng chuyển tiếp IP (IP forwarding tables)
- ❖ **Linh hoạt:** các quyết định chuyển tiếp MPLS có thể khác với IP của chúng
  - Dùng địa chỉ đích và nguồn để định tuyến các luồng dữ liệu tới cùng đích đến một cách khác nhau (same destination differently) (traffic engineering)
  - Định tuyến lại các luồng dữ liệu nhanh chóng nếu đường liên kết hỏng: các đường dẫn dự phòng được tính toán trước (hữu dụng cho VoIP)

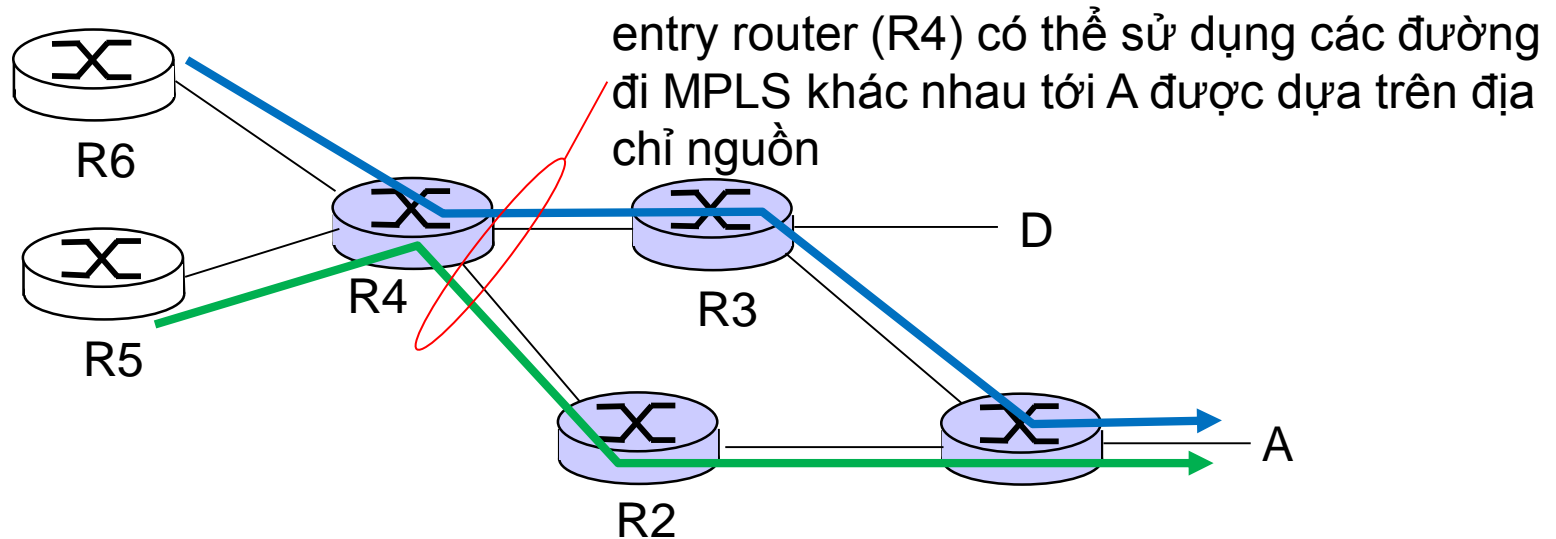
# So sánh đường đi MPLS và IP



- ❖ **Định tuyến IP:** đường tới đích đến được xác định bởi 1 địa chỉ đích



# So sánh đường đi MPLS và IP



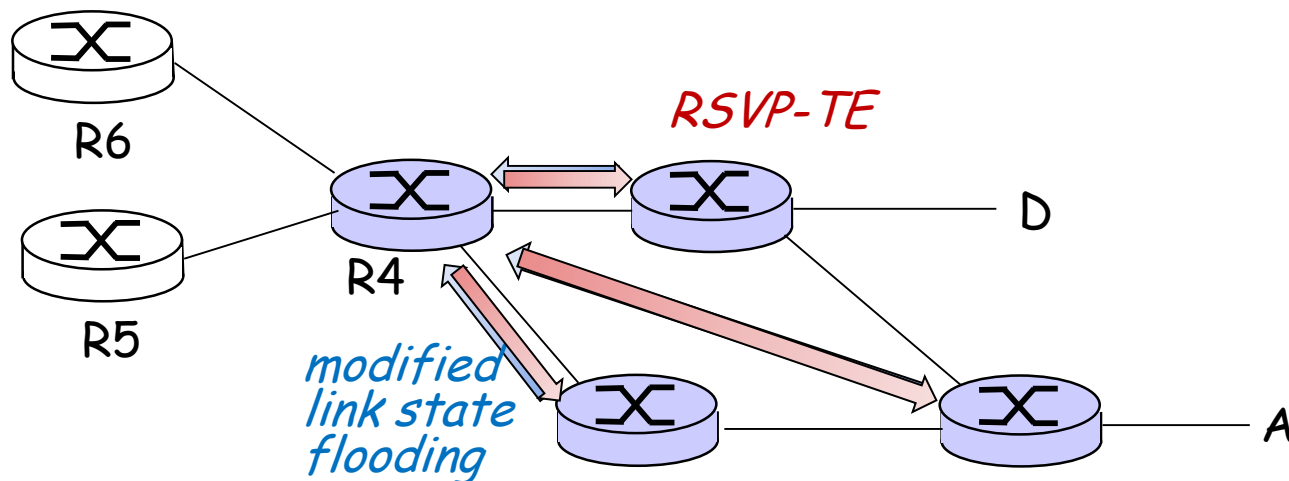
 IP-only router

 MPLS và IP router

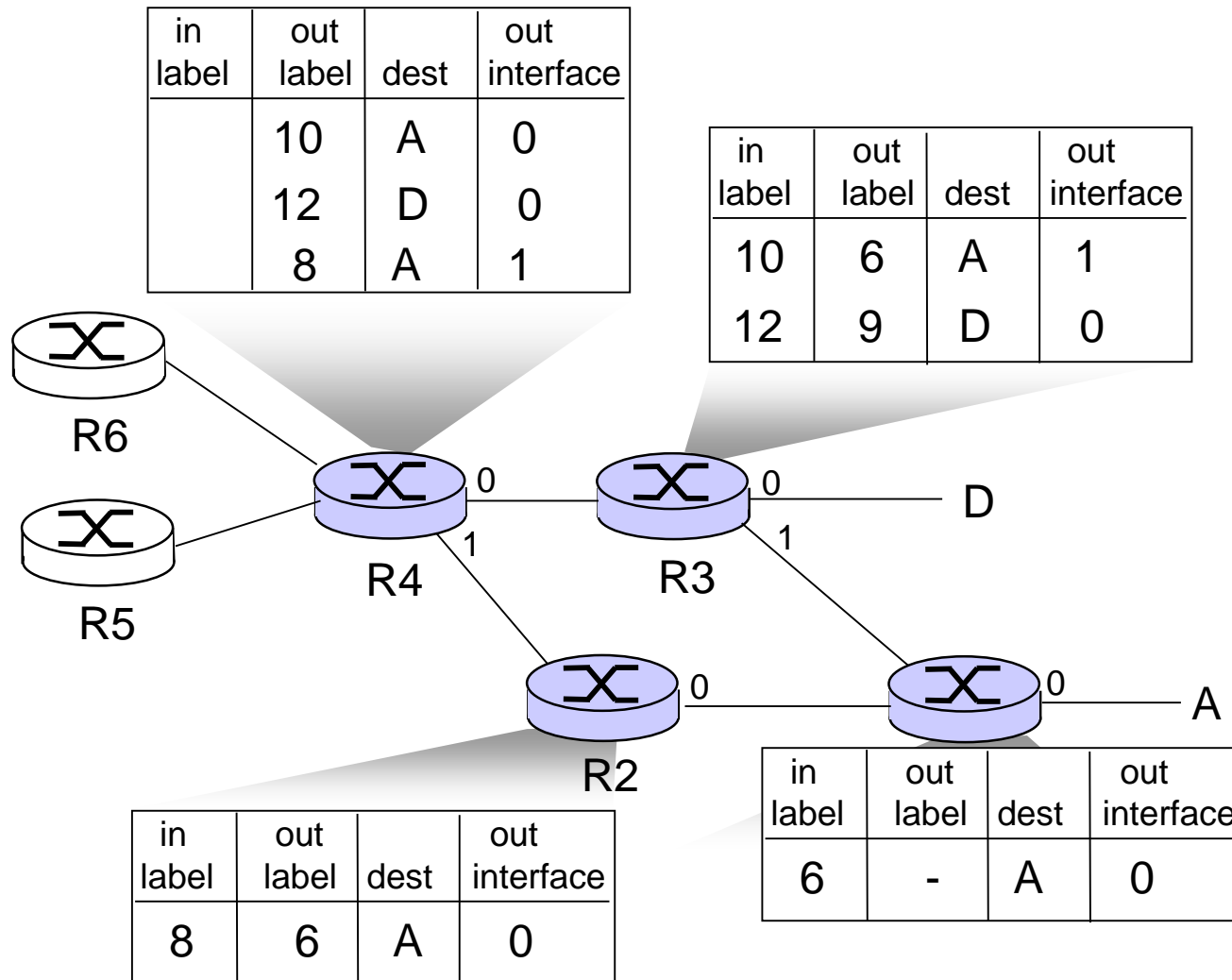
- ❖ **Định tuyến IP:** đường tới đích đến được xác định bởi chỉ cần địa chỉ đích
- ❖ **Định tuyến MPLS:** đường tới đích đến có thể được dựa trên địa chỉ nguồn và đích
  - **Định tuyến lại nhanh chóng (fast reroute):** tính toán lại các đường đi dự phòng trong trường hợp đường kết nối bị hỏng

# Tín hiệu MPLS

- ❖ Chỉnh sửa các giao thức flooding IS-IS link-state, OSPF để mang thông tin được sử dụng bởi định tuyến MPLS,
  - Ví dụ: link bandwidth, số lượng băng thông đường link "được dành riêng"
- ❖ *entry của router MPLS sử dụng giao thức tín hiệu RSVP-TE để thiết lập chuyển tiếp MPLS tại các router luồng dưới*



# Bảng chuyển tiếp MPLS



# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

5.7 một ngày trong cuộc sống của một yêu cầu web

# Mạng trung tâm dữ liệu

- ❖ 10 đến 100 ngàn host:
  - e-business (e.g. Amazon)
  - content-servers (như là YouTube, Akamai, Apple, Microsoft)
  - search engines, data mining (như là Google)
- ❖ Thách thức:
  - Nhiều ứng dụng, mỗi cái phục vụ số lượng lớn client
  - Quản lý/cân bằng tải, tránh tắc nghẽn dữ liệu, mạng và tiến trình

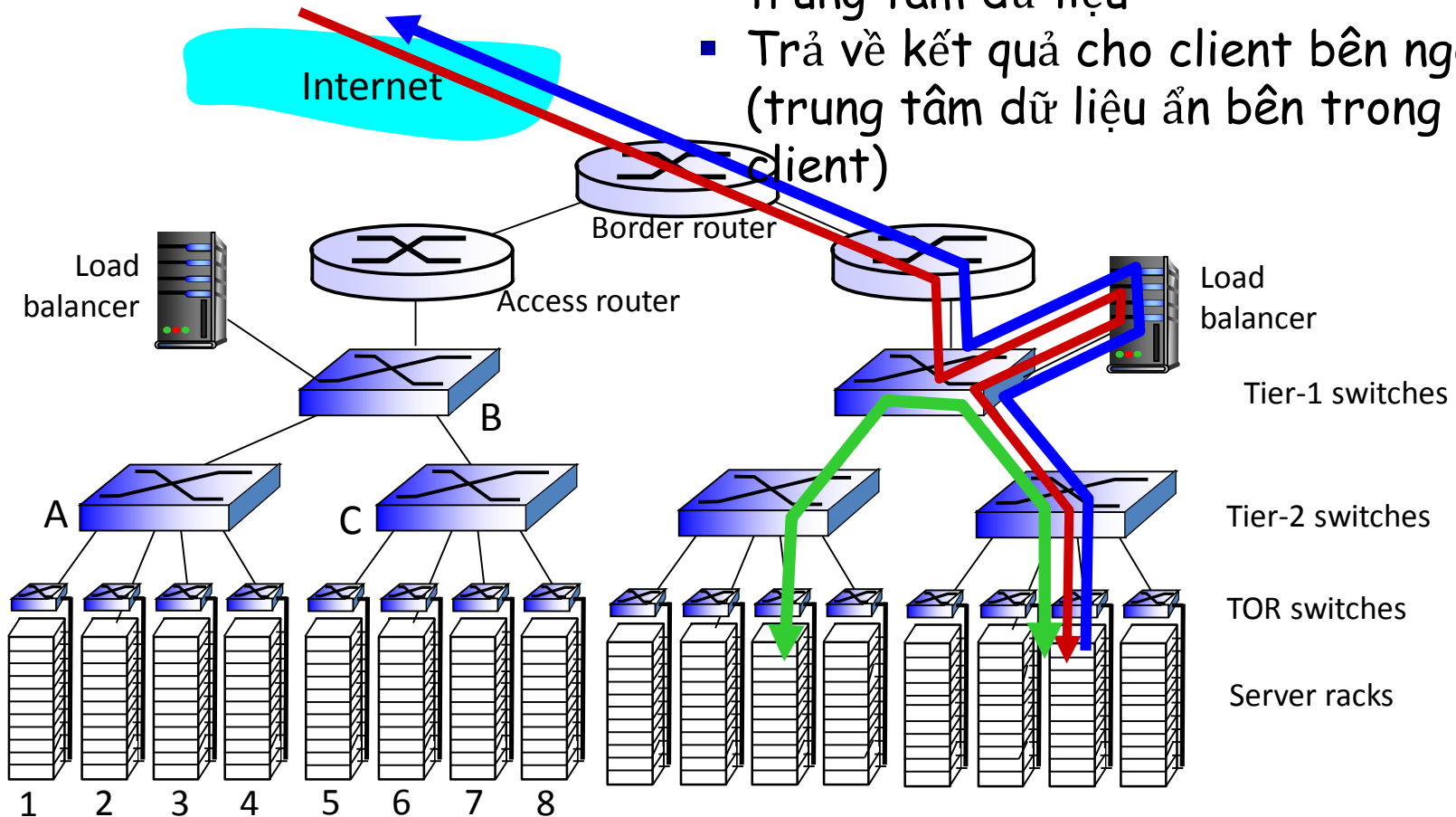


Inside a 40-ft Microsoft container,  
Chicago data center

# Mạng trung tâm dữ liệu

*Cân bằng tải: định tuyến tầng application*

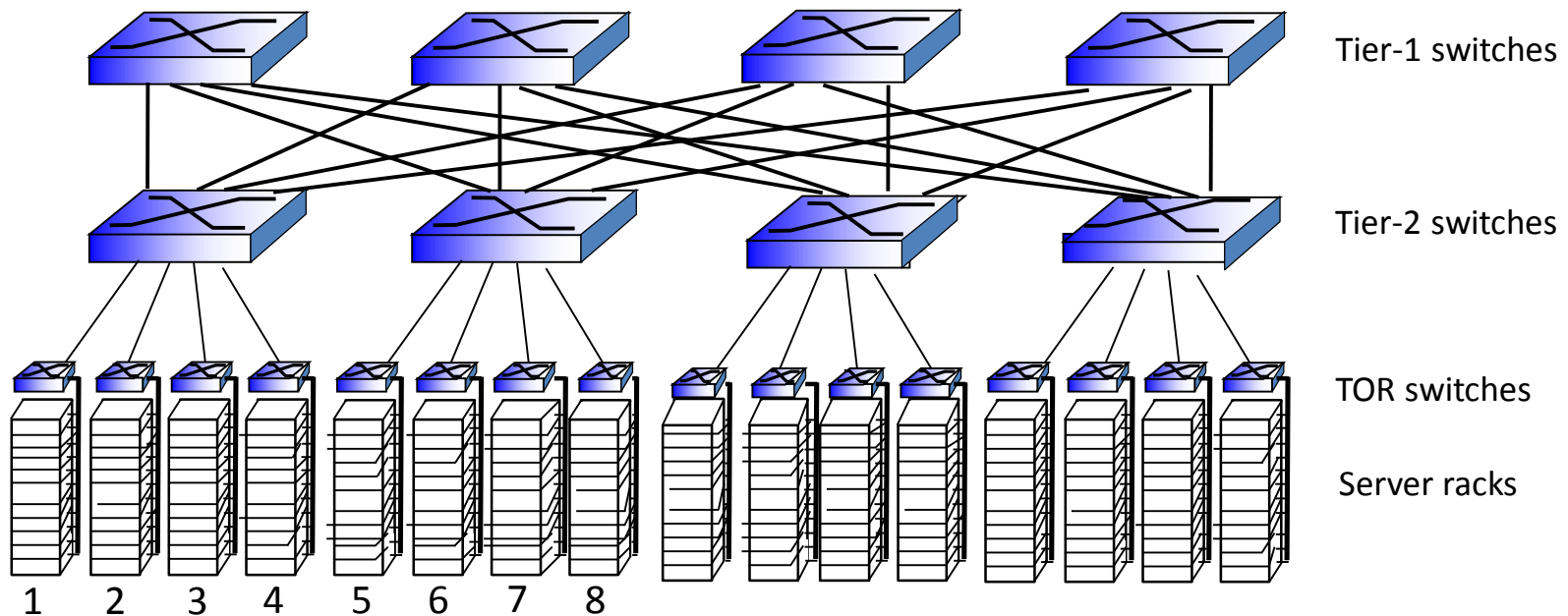
- Nhận các yêu cầu client bên ngoài
- Hướng dẫn khối lượng công việc trong trung tâm dữ liệu
- Trả về kết quả cho client bên ngoài (trung tâm dữ liệu ẩn bên trong đối với client)





# Mạng trung tâm dữ liệu

- ❖ Rất nhiều kết nối giữa các switch và rack:
  - Thông lượng được tăng lên giữa các rack (nhiều đường định tuyến có thể dùng được)
  - Độ tin cậy và khả năng dự phòng tăng lên



# Tầng Link và mạng LAN: Nội dung

5.1 Giới thiệu và các dịch vụ

5.2 phát hiện lỗi và sửa lỗi

5.3 các giao thức đa truy cập

5.4 mạng LAN

- Định địa chỉ, ARP
- Ethernet
- switches
- VLANs

5.5 link virtualization: MPLS

5.6 mạng trung tâm dữ liệu

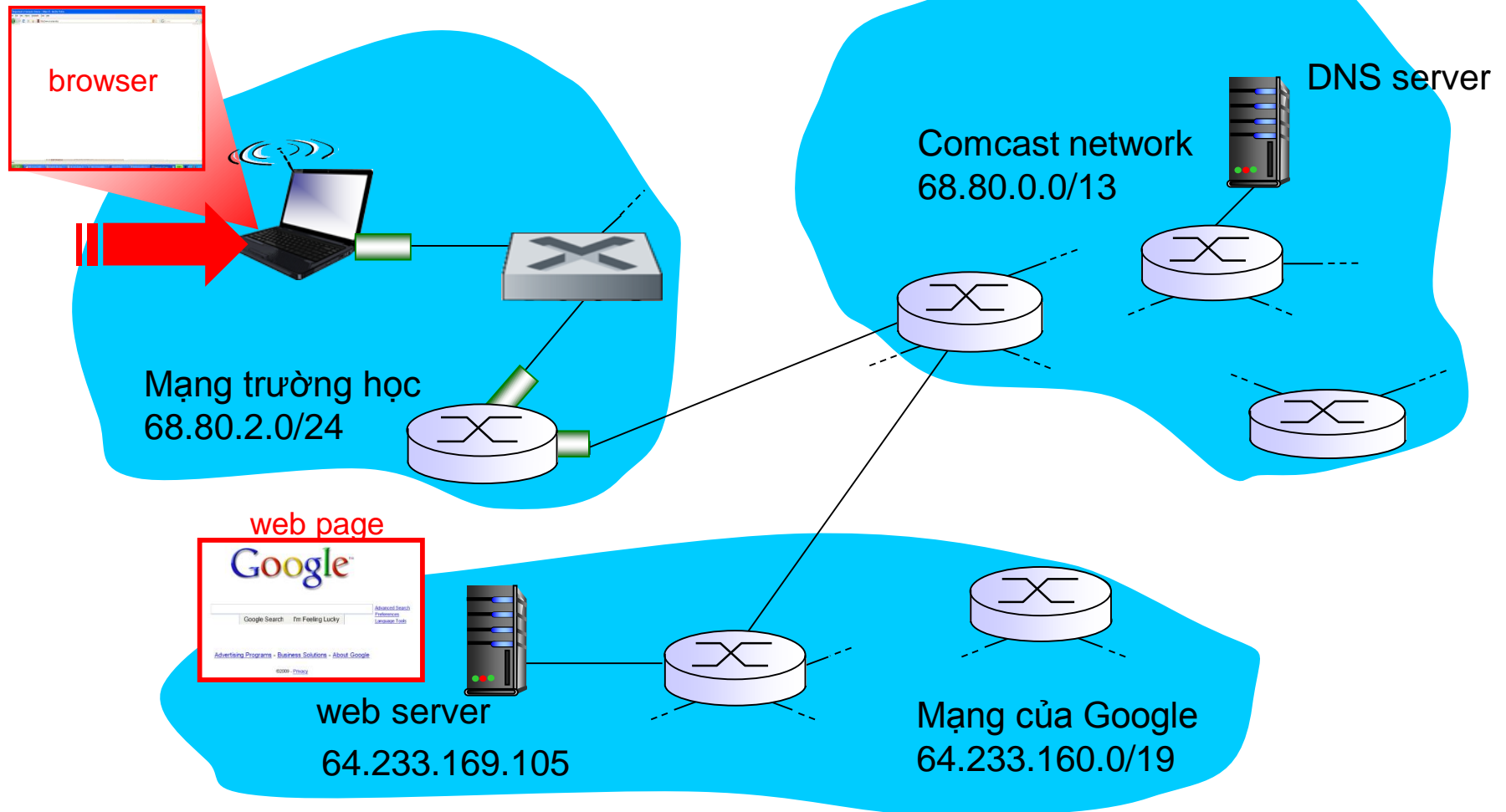
5.7 một ngày trong cuộc sống của một yêu cầu web

# *Synthesis*: một ngày trong cuộc sống của một truy vấn web

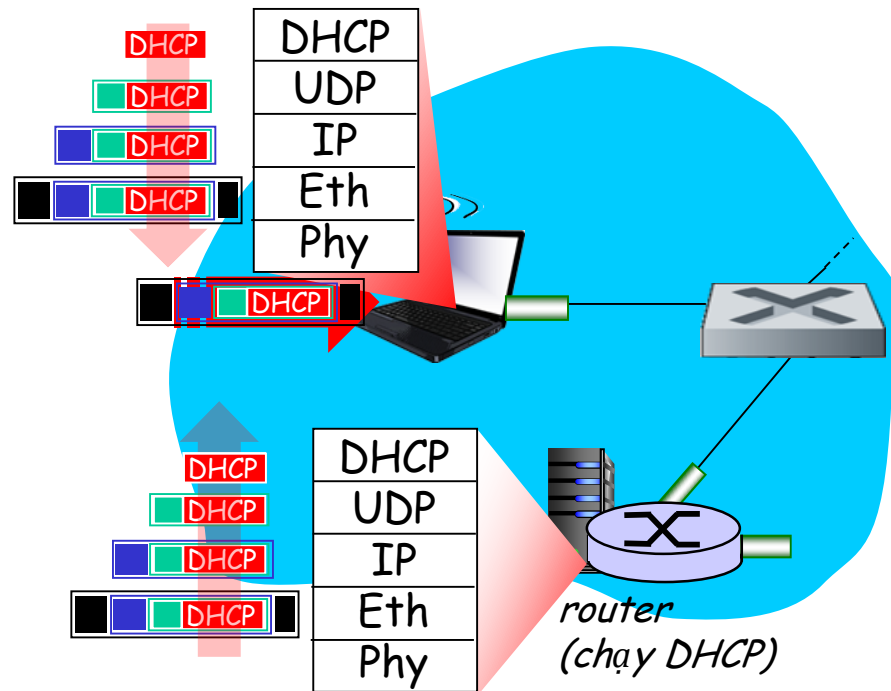
---

- ❖ Tìm hiểu *đầy đủ* chồng giao thức từ trên xuống dưới!
  - application, transport, network, link
- ❖ putting-it-all-together: synthesis!
  - *Mục tiêu*: xác định, xem xét, hiểu các giao thức (tại tất cả các tầng) được tham gia vào tình huống đơn giản: truy vấn trang www
  - *Ngữ cảnh*: sinh viên kết nối máy tính xách tay vào mạng của tòa nhà trường học, yêu cầu/nhận [www.google.com](http://www.google.com)

# Một ngày trong cuộc sống: ngữ cảnh

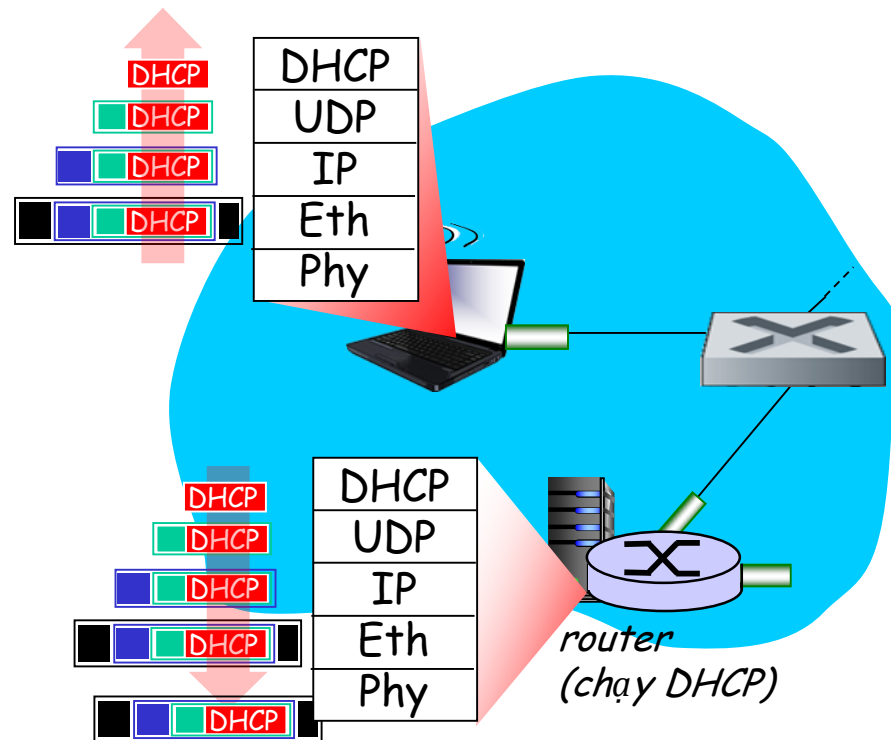


# Một ngày trong cuộc sống ... kết nối tới Internet



- ❖ Kết nối máy tính xách tay cần có địa chỉ IP của riêng nó, địa chỉ của router first-hop, địa chỉ của DNS server: dùng **DHCP**
- ❖ Yêu cầu DHCP được đóng gói trong **UDP**, được đóng gói trong **IP**, được đóng gói trong **802.3 Ethernet**
- ❖ Ethernet frame **broadcast** (dest: FFFFFFFFFFFFFFFF) trên LAN, được nhận tại router chạy **DHCP** server
- ❖ Ethernet **demuxed** to IP demuxed, UDP demuxed to DHCP

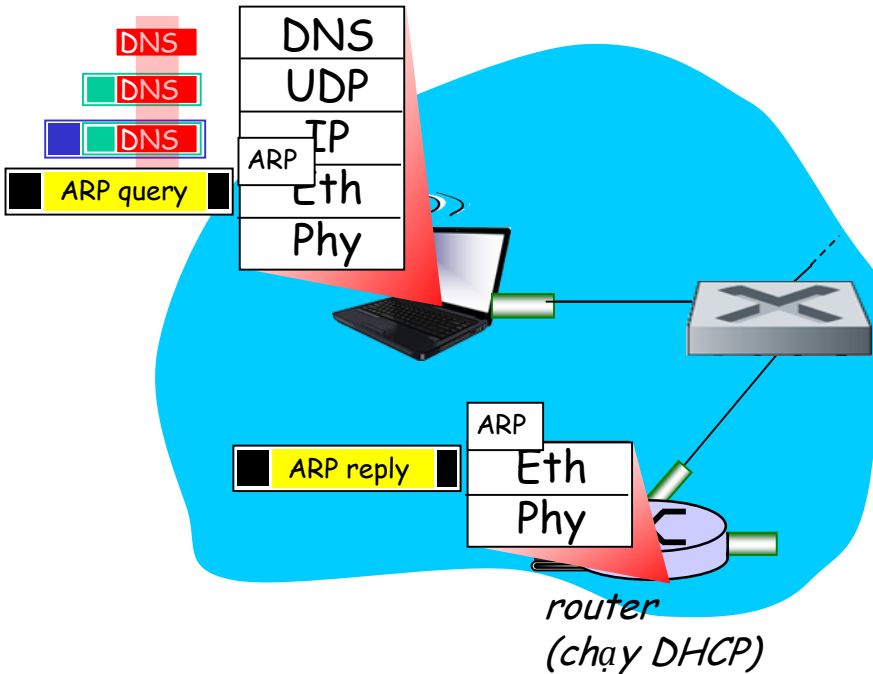
# Một ngày trong cuộc sống ... kết nối tới Internet



- ❖ DHCP server lập **DHCP ACK** chứa địa chỉ IP của client, địa chỉ IP của router first-hop cho client đó, tên và địa chỉ IP của DNS server
- ❖ Đóng gói tại DHCP server, frame được chuyển tiếp (**sự học của switch**) thông qua mạng LAN, tách ra tại client
- ❖ DHCP client nhận DHCP ACK reply

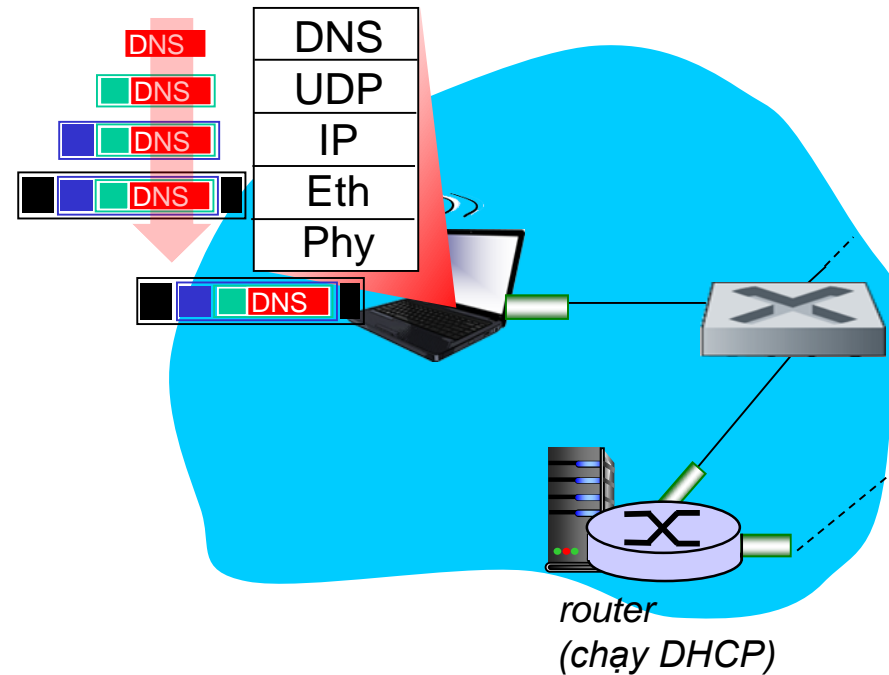
*Bây giờ, Client có địa chỉ address, biết tên và địa chỉ của DNS server, địa chỉ IP của router first-hop*

# Một ngày trong cuộc sống... ARP (trước DNS, trước HTTP)

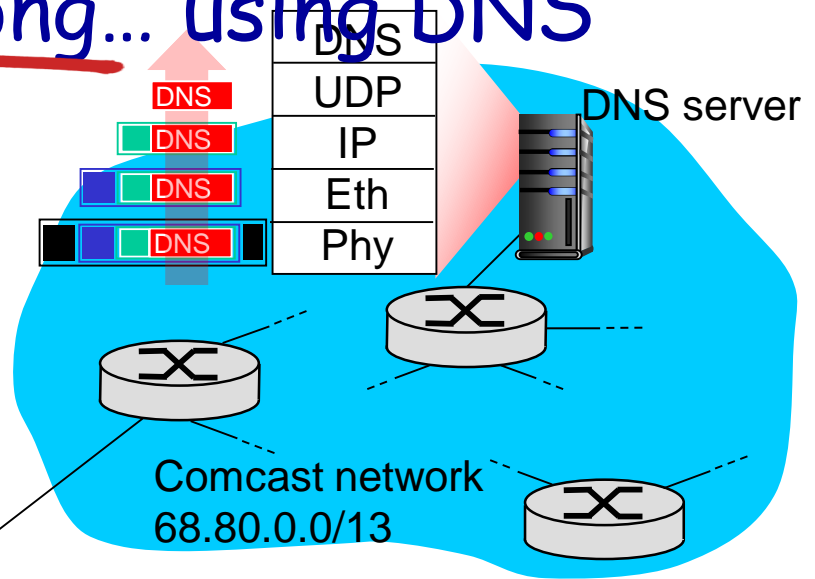


- ❖ Trước khi gửi *HTTP* request, cần địa chỉ IP của `www.google.com`: *DNS*
- ❖ DNS query được tạo, đóng gói trong UDP, được đóng gói trong IP, được đóng gói trong Eth. Gửi frame tới router, cần địa chỉ MAC của interface của router interface: *ARP*
- ❖ *ARP query* broadcast, được nhận bởi router, router này sẽ trả lời lại với *ARP reply* cung cấp địa chỉ MAC của interface của router này
- ❖ Hiện tại, client biết địa chỉ MAC của router first hop, vì vậy nó có thể gửi frame chứa DNS query

# Một ngày trong cuộc sống... using DNS



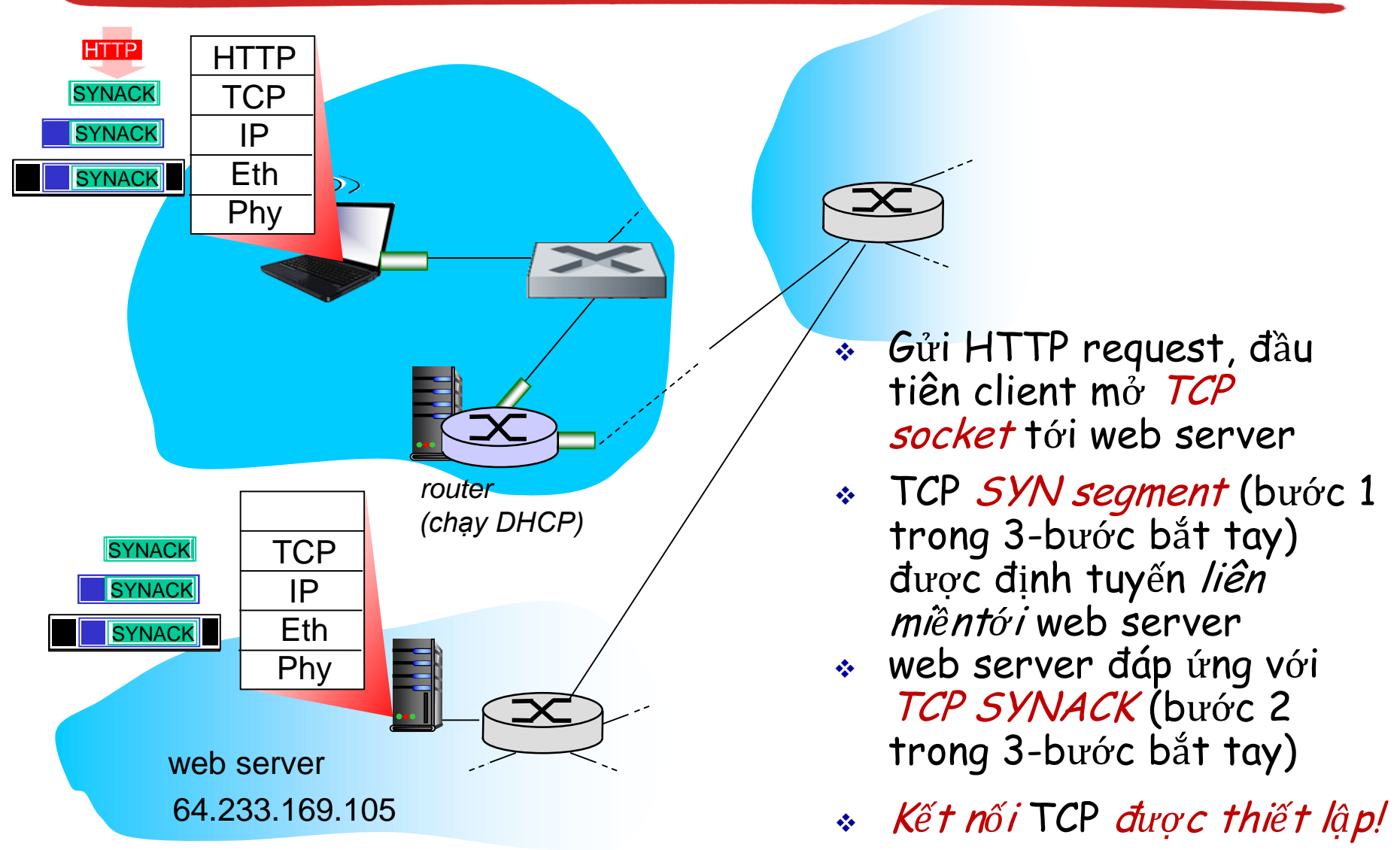
- ❖ IP datagram chứa DNS query được chuyển tiếp thông qua switch của mạng LAN từ client tới router hop thứ nhất



- ❖ IP datagram được chuyển tiếp từ mạng campus tới mạng Comcast, được định tuyến (các bảng được tạo bởi các giao thức định tuyến *RIP, OSPF, IS-IS* và/hoặc *BGP*) tới DNS server
- ❖ Được tách/ghép tới DNS server
- ❖ DNS server trả lời cho client với địa chỉ IP address của [www.google.com](http://www.google.com)

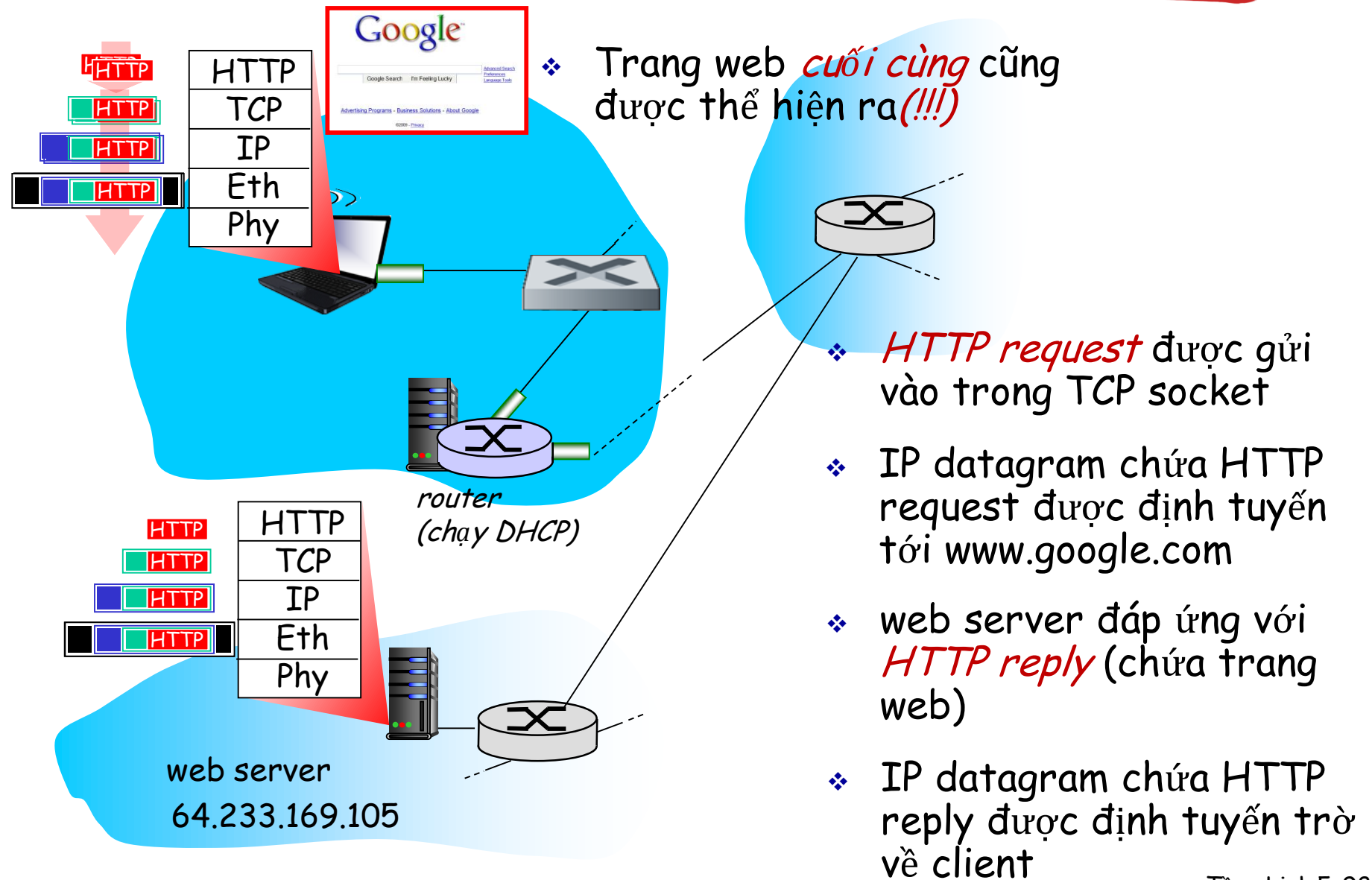


# Một ngày trong cuộc sống...kết nối TCP mang HTTP



# Một ngày trong cuộc sống...

## HTTP yêu cầu/trả lời (request/reply)



# Chương 5: Tổng kết

- ❖ các nguyên lý của các dịch vụ tầng data link:
  - Phát hiện và sửa chữa lỗi
  - Chia sẻ kênh broadcast: đa truy cập
  - Định địa chỉ tầng link
- ❖ Thực hiện các công nghệ khác nhau của tầng link
  - Ethernet
  - Mạng LAN và VLAN chuyển mạch
  - Mạng ảo hóa như là một lớp tầng link: MPLS
- ❖ Tổng hợp: một ngày trong cuộc sống của truy vấn web

# Chương 5: let's take a breath

- ❖ Tìm hiểu *đầy đủ* chồng giao thức từ trên xuống dưới (ngoại trừ PHY)
- ❖ Hiểu về các nguyên tắt mạng và hiện thực
- ❖ ..... Có thể dừng tại đây .... Nhưng có một số chủ đề thú vị!
  - wireless
  - multimedia
  - security
  - network management