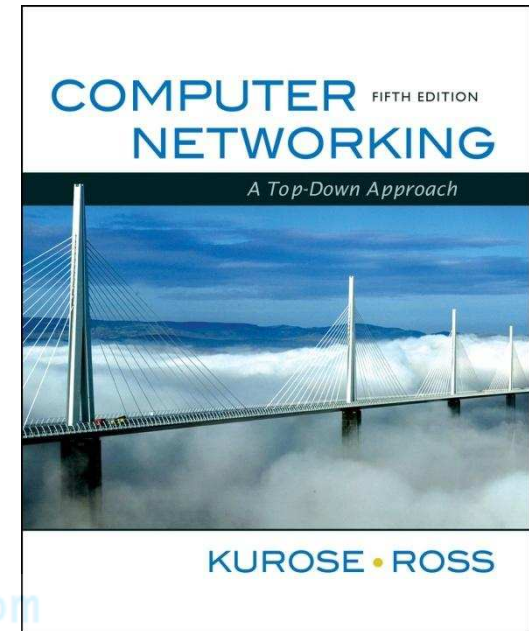


# Chapter 5

## Link Layer and LANs



### 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 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) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- ❑ If you post any slides in substantially unaltered form 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 1996-2009  
J.F Kurose and K.W. Ross, All Rights Reserved

*Computer Networking:  
A Top Down Approach  
5<sup>th</sup> edition.*

*Jim Kurose, Keith Ross  
Addison-Wesley, April  
2009.*

# Chapter 5: The Data Link Layer

## Our goals:

- understand principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - reliable data transfer, flow control: *done!*
- instantiation and implementation of various link layer technologies

cuu duong than cong . com

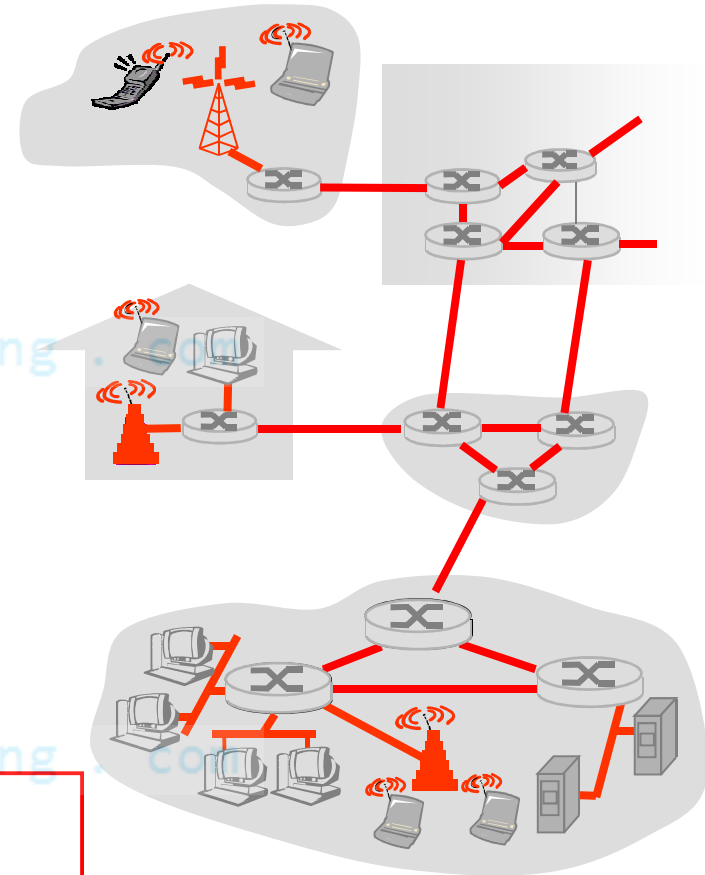
# Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-layer Addressing
- 5.5 Ethernet
- 5.6 Link-layer switches
- 5.7 PPP
- 5.8 Link virtualization: MPLS
- 5.9 A day in the life of a web request

# Link Layer: Introduction

## Vài thuật ngữ:

- hosts, routers gọi chung là **nút [nodes]**
- Kênh truyền thông kết nối các nút kề nhau theo đường truyền được gọi là **liên-kết [links]**
  - Liên-kết hữu tuyến [wired links]
  - Liên-kết vô tuyến [wireless links]
  - LANs
- Gói tin tầng 2 gọi là **khung [frame]**, chứa datagram



**Tầng data-link** chịu trách nhiệm chuyển giao datagram từ một nút đến một nút kế dọc theo liên-kết

# Link layer: context

- Các datagram có thể được chuyển giao theo các giao-thức-tầng-2 khác nhau khi đi qua các liên-kết khác nhau
  - e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link
- Mỗi giao-thức-tầng-2 cung cấp các dịch vụ khác nhau
  - Chẳng hạn, có thể có/không cung cấp dịch vụ truyền dữ liệu đáng tin cậy qua liên-kết

## transportation analogy

- 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 Services

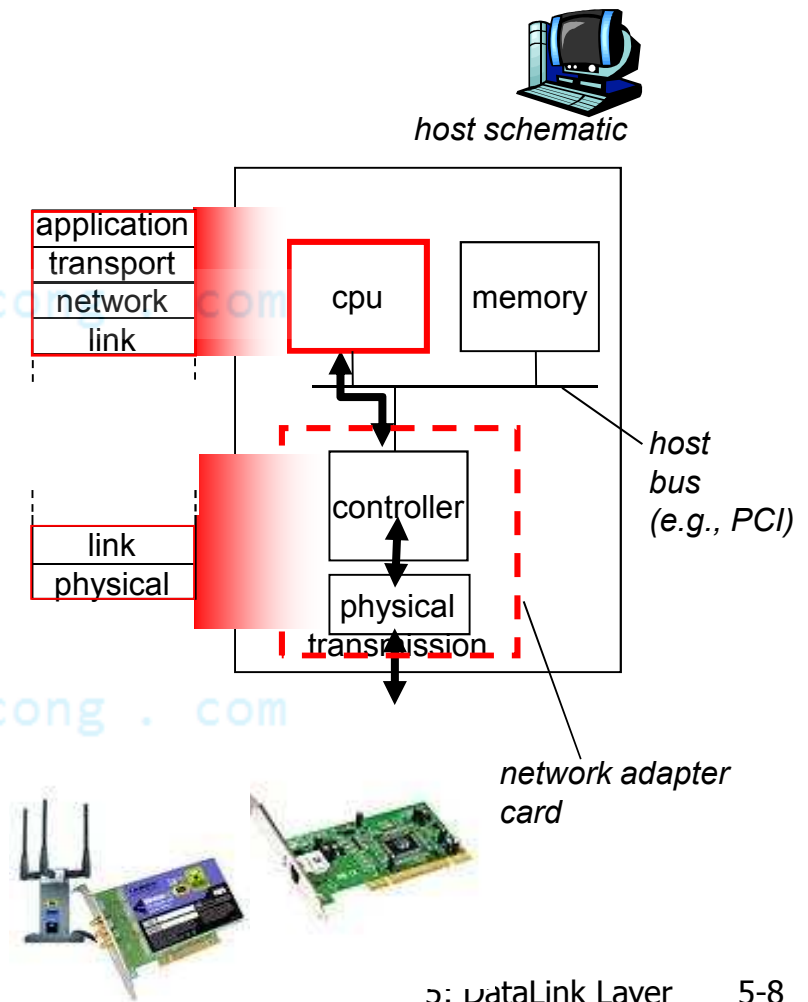
- *Đóng gói, truy cập kênh truyền:*
  - Đóng gói datagram vào các frame, bổ sung header, trailer
  - Truy cập kênh truyền nếu môi trường truyền dẫn là dùng chung
  - Địa chỉ MAC được sử dụng trong frame header để xác định nguồn gửi, đích nhận.
    - Địa chỉ MAC khác với địa chỉ IP
- *Chuyển giao đáng tin cậy giữa các nút kề nhau*
  - Đã đề cập trong chương 3 (giao thức rdt)
  - Hiếm khi được sử dụng trên các đường truyền ít lỗi bit (cáp quang, cáp xoắn đôi)
  - Đường vô tuyến: mức độ lỗi cao
    - Q: why both link-level and end-end reliability?

# Link Layer Services (more)

- *flow control:*
  - Điều tiết tốc độ giữa 2 nút gửi/nhận kề nhau.
- *Phát hiện lỗi [error detection]:*
  - Lỗi phát sinh do sự suy yếu tín hiệu, nhiễu tín hiệu.
  - Khi bên nhận phát hiện có lỗi
    - Báo cho bên gửi truyền lại hoặc vứt bỏ frame
- *Tự sửa lỗi [error correction]:*
  - Bên nhận nhận ra lỗi và tự sửa lỗi mà không cần yêu cầu truyền lại.
- *half-duplex and full-duplex*
  - Với half duplex, các nút ở đầu liên-kết có thể truyền cho nhau, nhưng không thể cùng lúc.

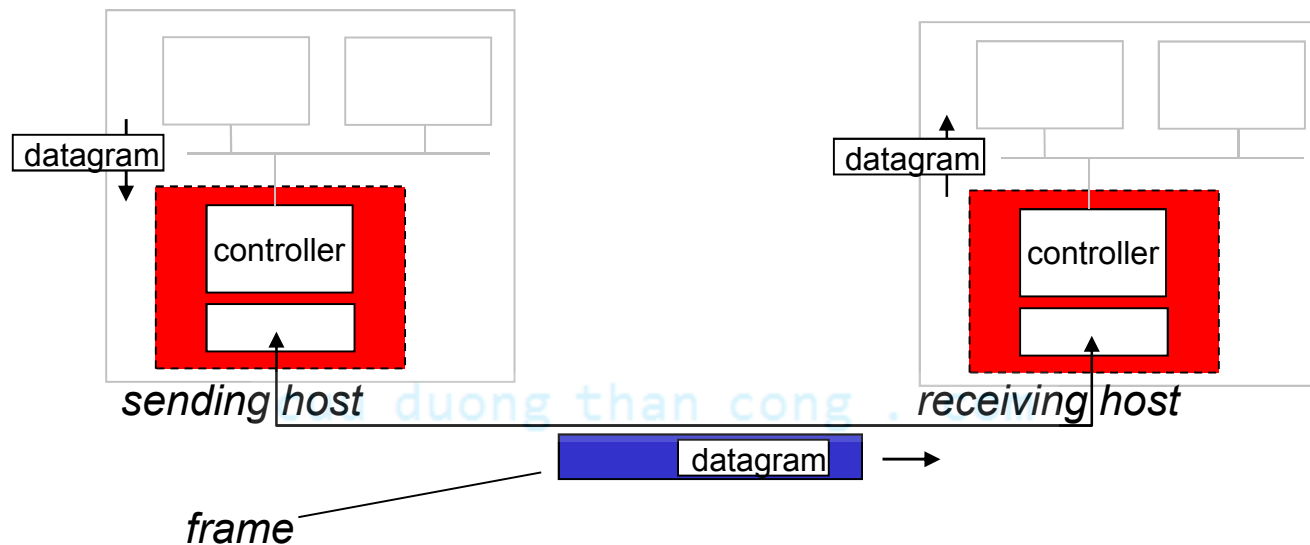
# Where is the link layer implemented?

- Tên mạng: tầng link được triển khai ở tất cả các nút.
- Trên máy tính: tầng link được triển khai ở các mạng (*network interface card, NIC*)
  - Ethernet card, PCMCIA card, 802.11 card
  - Các mạng được gắn vào hệ thống bus của máy tính (VD PCI) hoặc dạng built-in.
- Chức năng của tầng link được cài đặt ở phần cứng, phần mềm và firmware





# Adaptors Communicating



## ■ Bên gửi:

- Bao bọc datagram trong các frame
- Bổ sung các bit kiểm lỗi, rdt, flow control, ...

## ■ Bên nhận

- Xử lý lỗi, rdt, flow control
- Trích ra datagram và đẩy lên tầng trên

# Link Layer

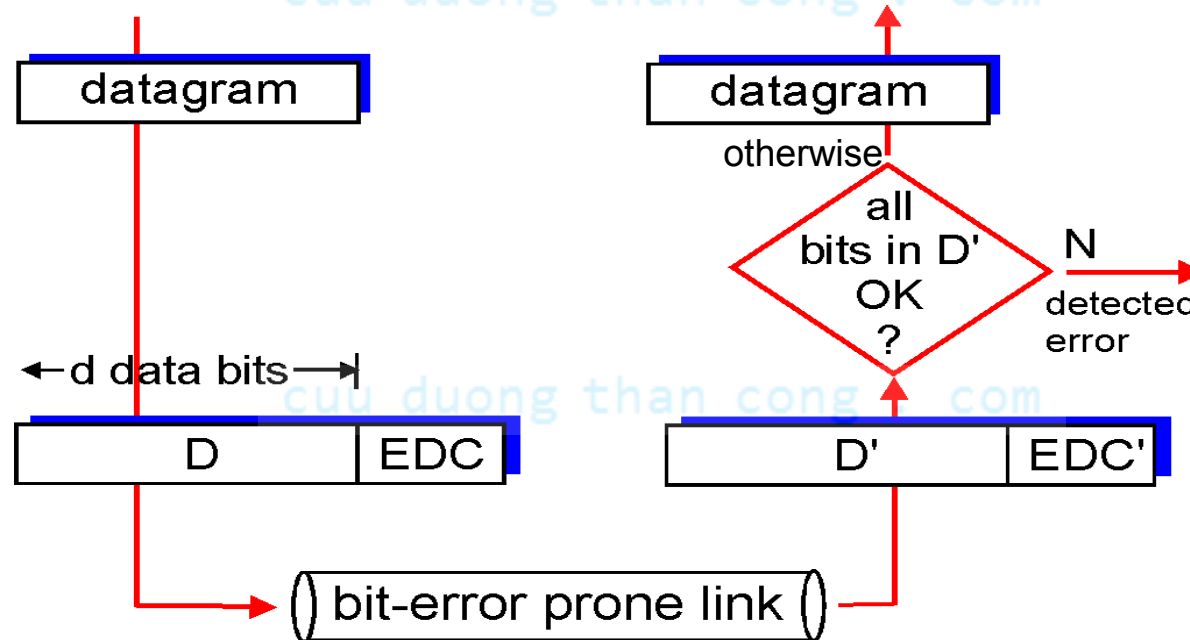
- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-layer Addressing
- 5.5 Ethernet
- 5.6 Link-layer switches
- 5.7 PPP
- 5.8 Link virtualization: MPLS
- 5.9 A day in the life of a web request

# Error Detection

EDC= Error Detection and Correction bits (redundancy)

D = Data protected by error checking, may include header fields

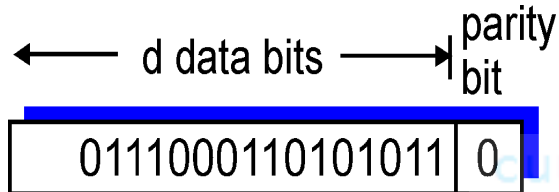
- Chức năng phát hiện lỗi không hoàn toàn đáng tin cậy 100%!
  - Dù rất hiếm, nhưng giao thức vẫn có thể sót lỗi.
  - Field EDC càng lớn thì càng dễ phát hiện và sửa lỗi hơn.



# Parity Checking

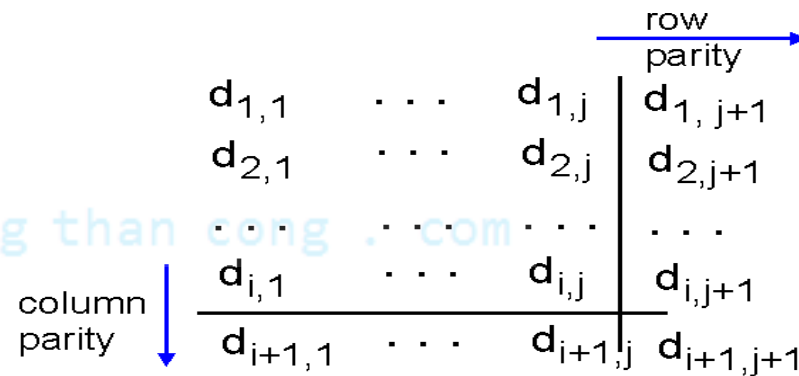
## Single Bit Parity:

Phát hiện có lỗi bit



## Two Dimensional Bit Parity:

Phát hiện và sửa lỗi bit



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  
parity error

correctable  
single bit error

# Internet checksum (review)

**Mục tiêu:** phát hiện “lỗi” (e.g., sai bit) trong gói tin được chuyển giao (chú ý: chỉ sử dụng ở tầng transport)

## **Bên gửi:**

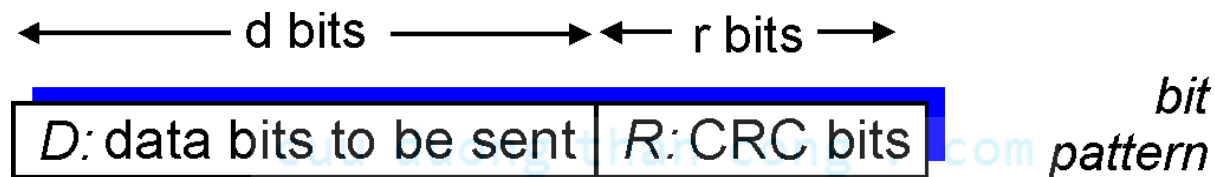
- Xem nội dung segment như các số nguyên dài 16 bit
- Checksum: bù 1 của tổng các số nguyên 16 bit
- Bên gửi phải đặt checksum vào UDP checksum field

## **Bên nhận:**

- Tính lại checksum của segment nhận được.
- So sánh checksum tính được bằng với checksum nhận được:
  - KHÁC – có lỗi
  - BẰNG – Không phát hiện ra lỗi. *Nhưng vẫn có thể có lỗi.*

# Checksumming: Cyclic Redundancy Check

- Xem các bit dữ liệu, gọi là **D**, như là những số nhị phân.
- Chọn mẫu dài  $(r+1)$  bit, gọi là bộ sinh (generator), gọi là **G**
- Mục tiêu: chọn  $r$  bit CRC, gọi là **R**, sao cho
  - $\langle D, R \rangle$  chia hết cho  $G$  (cơ số 2)
  - Bên nhận biết  $G$ . Khi nhận được  $\langle D, R \rangle$ , sẽ kiểm phần dư của phép chia  $\langle D, R \rangle$  cho  $G$ . Nếu phần dư khác 0: phát hiện có lỗi!
  - can detect all burst errors less than  $r+1$  bits
- Được sử dụng rộng rãi trong thực tế (Ethernet, 802.11 WiFi, ATM)



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

mathematical formula

# CRC Example

Muốn tìm R sao cho: tồn tại n nguyên để:

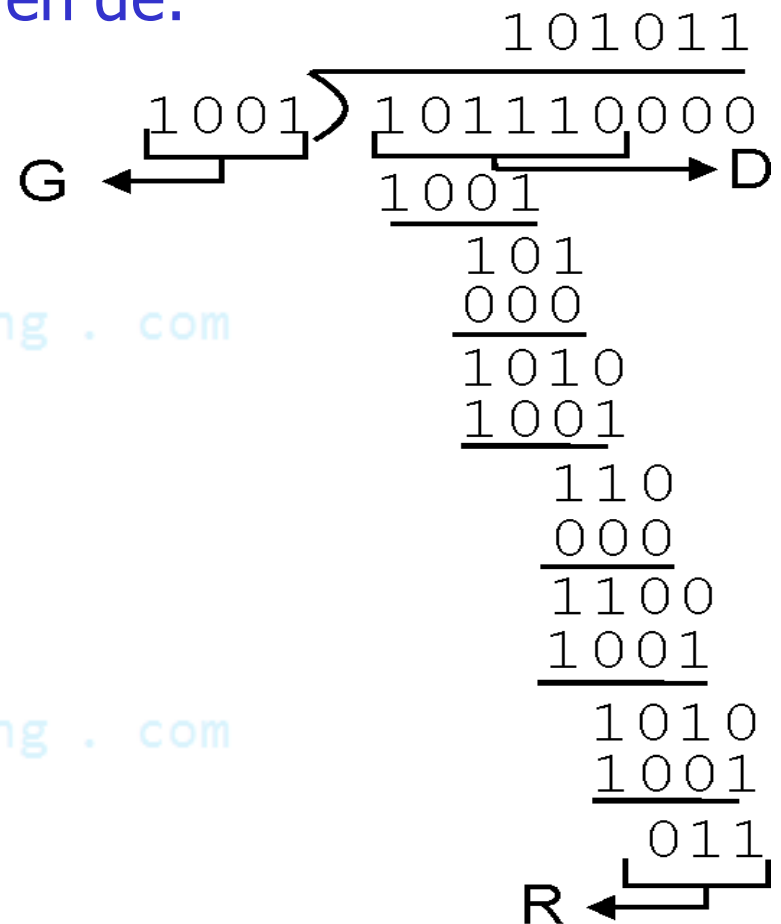
$$D \cdot 2^r \text{ XOR } R = nG$$

Thực hiện phép (XOR R) cả 2 vế:

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

*Tương đương:*

$$R = \text{Phần dư } [D \cdot 2^r / G]$$



# Link Layer

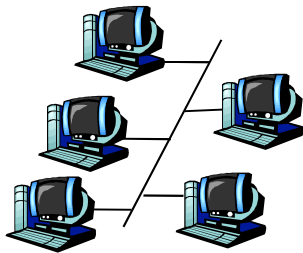
- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-layer Addressing
- 5.5 Ethernet
- 5.6 Link-layer switches
- 5.7 PPP
- 5.8 Link virtualization: MPLS
- 5.9 A day in the life of a web request



# Multiple Access Links and Protocols

Hai kiểu “liên-kết”:

- **Điểm-điểm [point-to-point]**: một phía của liên-kết là bên gửi, phía kia là bên nhận.
  - PPP dùng trong truy cập quay số [dialup access]
  - Liên-kết điểm-điểm giữa Ethernet switch và host
- **Quảng bá [broadcast]** nhiều nút gửi và nhận kết nối, tất cả được nối chung vào cùng môi trường truyền chung
  - Ethernet
  - 802.11 wireless LAN



shared wire (e.g.,  
cabled Ethernet)



shared RF  
(e.g., 802.11 WiFi)



shared RF  
(satellite)



humans at a  
cocktail party  
(shared air, acoustical)

# Multiple Access protocols

Xét môi trường truyền quảng bá/dùng chung.

- Hai hay nhiều cuộc truyền đồng thời từ các nút: tín hiệu từ các frame bị trộn lẫn, không thể tách rời=> ta nói có hiện tượng **va chạm** [collision]
  - Hiện tượng này gây lãng phí kênh truyền
- Vấn đề quan trọng ở tầng link: Làm thế nào để điều khiển việc **nhiều** nút cùng gửi và cùng nhận trên một môi trường truyền chung? (multiple access problem)
- Cần phải có quy tắc cho việc này, chính là **giao thức đa truy cập** [multiple access protocol]

# Multiple Access protocols

## Giao thức đa truy cập [multiple access protocol]

- Thuật toán phân bổ nhằm xác định cách thức các nút dùng chung kênh truyền, i.e. xác định thời điểm có thể truyền.
- Truyền thông về việc *dùng chung kênh truyền* cũng phải sử dụng chính kênh truyền.
  - Không có phần kênh truyền dành riêng cho việc điều phối.

cuu duong than cong . com

# Ideal Multiple Access Protocol

**Lý tưởng**, một giao thức đa truy cập kênh truyền quảng bá có khả năng truyền  $R$  bps nên có những đặc tính:

1. Khi chỉ có **một nút** gửi dữ liệu lên đường truyền, nó có thể truyền với thông lượng  $R$  bps (tức **tối đa**)
2. Khi có  **$M$  nút** muốn truyền, mỗi nút có thể truyền ở thông lượng trung bình  **$R/M$**  bps
3. Phi tập trung hoàn toàn
  - Không có nút đặc biệt lo chuyện điều phối việc truyền=> ko lo nút đó bị hỏng sẽ ảnh hưởng đến toàn mạng.
  - Không có sự đồng bộ hóa với tín hiệu đồng hồ, không chia slots
4. Đơn giản, dễ hiện thực hóa.

# MAC Protocols: a taxonomy

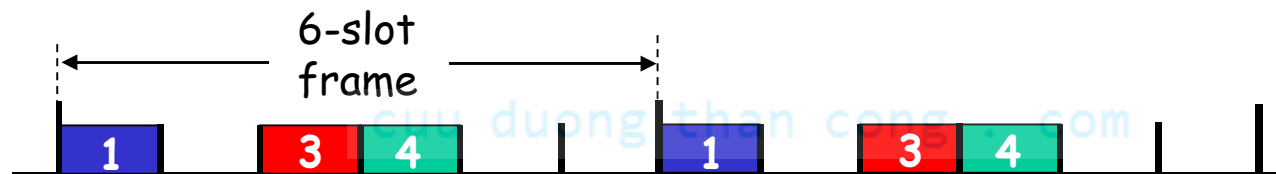
Có ba các giao thức truy cập đường truyền chính:

- **Phân chia kênh truyền [Channel Partitioning]**
  - Chia kênh truyền ra thành những “mảnh” nhỏ hơn (time slots, frequency, code)
  - Cấp phát các “mảnh kênh truyền” cho các nút sử dụng (riêng)
- **Truy cập ngẫu nhiên [Random Access]**
  - Kênh truyền không được chia nhỏ, cho phép sự va chạm.
  - Sử dụng cơ chế “khôi phục” sau va chạm.
- **“Lấy lượt” [“Taking turns”]**
  - Các nút có thể “lấy lượt”, nhưng các nút có nhu cầu truyền nhiều có thể có lượt truyền dài hơn.

# Channel Partitioning MAC protocols: TDMA

## TDMA: time division multiple access

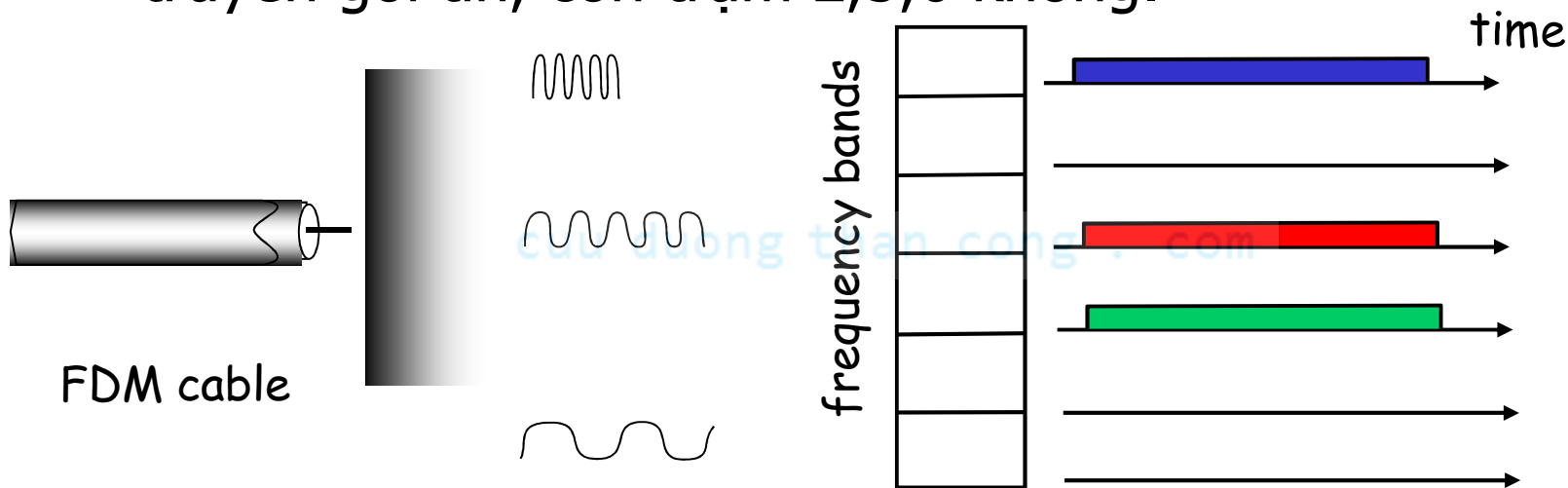
- Truy cập kênh truyền theo “tua”
- Mỗi trạm (station) nhận 1 khe có chiều dài cố định trong mỗi tua.
- Khe không sử dụng thì không có tín hiệu (i.e. lặng im)
- VD: với mạng LAN có 6 trạm như bên dưới: trạm 1,3,4 có truyền gói tin; trạm 2,5,6 thì không.



# Channel Partitioning MAC protocols: FDMA

## FDMA: frequency division multiple access

- Dải kênh truyền được chia thành nhiều băng tần
- Mỗi trạm được gán cho một tần số cố định
- Nếu trạm không có truyền gói tin thì băng tần cũng không có tín hiệu
- VD: Với mạng LAN có 6 trạm như dưới, trạm 1,3,4 có truyền gói tin, còn trạm 2,5,6 không.



# Random Access Protocols

- Khi nút có gói tin để gửi đi
  - Gửi dữ liệu lên toàn bộ băng thông kênh truyền ở mức truyền R
  - Không có sự điều phối *ưu tiên* giữa các nút.
- Khi có 2 hay nhiều nút cùng truyền → “va chạm”
- **Giao thức truy cập ngẫu nhiên MAC [random access MAC protocol]** xác định rõ:
  - Cách thức phát hiện hiện tượng va chạm
  - Cách thức khôi phục khi có va chạm (e.g. thông qua việc trì hoãn việc truyền lại)
- Ví dụ về các giao thức truy cập ngẫu nhiên MAC
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA



# Slotted ALOHA

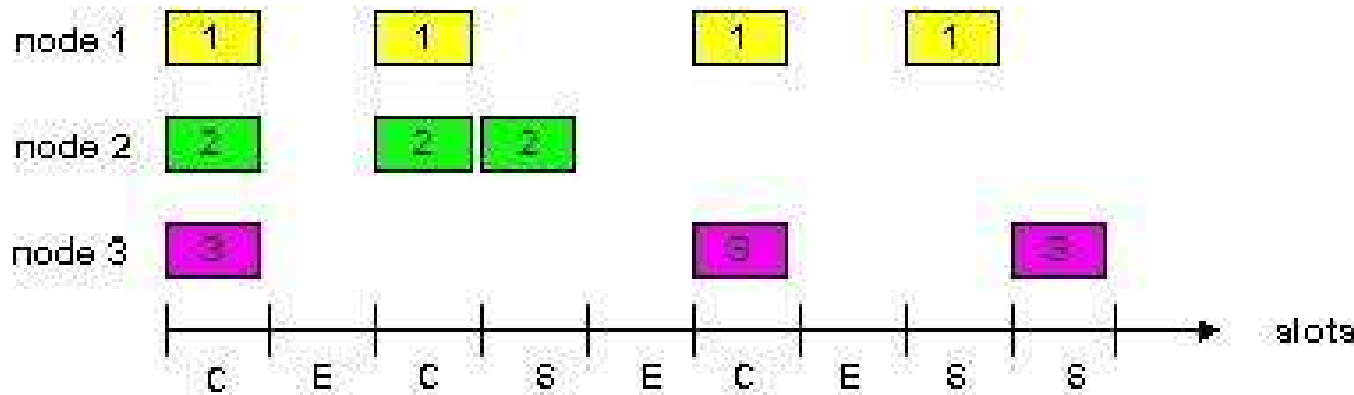
## Các giả định:

- Tất cả các frame đều có cùng kích thước.
- Thời gian truyền được chia thành các slot dài bằng nhau (là thời gian cần để truyền 1 frame)
- Nút bắt đầu truyền frame vào thời điểm bắt đầu slot
- Các nút được đồng bộ hóa, do đó mỗi nút biết thời điểm bắt đầu slot.
- Nếu có 2 hay nhiều frame va chạm nhau trên một slot, tất cả các nút sẽ phát hiện được hiện tượng này, trước khi slot kết thúc.

## Hoạt động:

- Khi nút gửi có frame dữ liệu để truyền, nó chờ đến khi bắt đầu slot kế tiếp và truyền frame vào slot này.
  - *Nếu không có va chạm:* nút có thể gửi frame mới (nếu có) vào slot kế tiếp.
  - *Nếu có va chạm:* nút gửi lại frame vào slot kế tiếp với xác suất  $p$  ( $0 < p < 1$ ) cho tới khi thành công

# Slotted ALOHA



## Thuận lợi

- Nút đang truyền có thể truyền liên tục ở thông lượng truyền cao nhất của kênh truyền
- Phi tập trung cao: Nút phát hiện va chạm và quyết định truyền lại một cách độc lập với nút khác.
- Giao thức hoạt động đơn giản

## Bất lợi

- Hiện tượng **va chạm**, gây lãng phí slot.
- Một phần **slot không được dùng đến**, do nút trì hoãn việc truyền với xác suất  $(1-p)$ .
- Nút phải phát hiện ra hiện tượng va chạm (nếu có) với **thời gian ngắn** hơn thời gian truyền.
- **Đồng bộ hóa** tín hiệu đồng hồ: Nút phải xác định được thời điểm bắt đầu 1 slot.

# Slotted Aloha efficiency

**Độ hiệu quả:** % slot thành công so với tổng số slot, khi có nhiều nút cùng truyền một thời gian đủ dài, mỗi nút có một số lượng lớn các frame cần truyền.

- *Giả sử:* có  $N$  nút với nhiều frame cần được gửi, mỗi nút truyền với xác suất truyền  $p$
- Xác suất **1 nút nhất định** sử dụng thành công 1 slot là  $p(1-p)^{N-1}$
- Xác suất **bất kỳ nút nào** cũng sử dụng thành công slot là  $Np(1-p)^{N-1}$

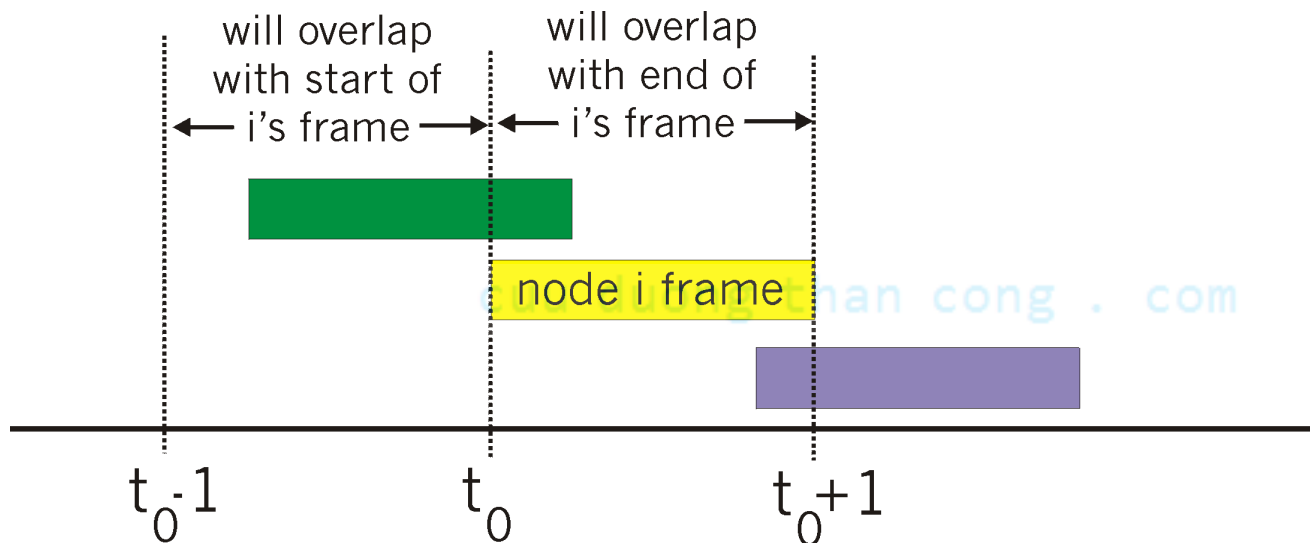
- Hiệu quả cực đại: **tìm  $p^*$  làm cực đại hóa** biểu thức  $Np(1-p)^{N-1}$
- Lấy giới hạn của  $Np^*(1-p^*)^{N-1}$  khi  $N \rightarrow \infty$ , được:

$$\text{Hiệu quả cực đại} = 1/e = .37$$

Khi có một số lớn các nút cùng truyền một số lượng lớn các frame, có tối đa 37% các slot làm việc có ích.

# Pure (unslotted) ALOHA

- unslotted Aloha: đơn giản hơn, không cần đồng bộ hóa
- Khi các frame đến lần đầu
  - Truyền đi ngay lập tức
- Khả năng va chạm gia tăng:
  - Frame được gửi ở thời điểm  $t_0$  và va chạm với các frame được gửi trong khoảng thời gian  $[t_0-1, t_0+1]$



# Pure Aloha efficiency

$P(\text{1 nút nhất định truyền th/công}) = P(\text{nút đó truyền}) \cdot$

$P(\text{ko có nút nào khác truyền trong } [t_0-1, t_0]) \cdot$

$P(\text{ko có nút nào khác truyền trong } [t_0, t_0 + 1] )$

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$

$$= p \cdot (1-p)^{2(N-1)}$$

Lấy giới hạn của biểu thức trên khi  $N \rightarrow \infty$ , được

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

**Còn tệ hơn slotted Aloha!**

# CSMA (Carrier Sense Multiple Access)

**CSMA**: (ý tưởng: lắng nghe trước khi truyền)

Nếu một nút “cảm được” kênh truyền đang lắng: nút đó truyền toàn bộ frame.

cuu duong than cong . com

- Tương tự cho con người: không ngắt lời người khác!

cuu duong than cong . com

# CSMA collisions

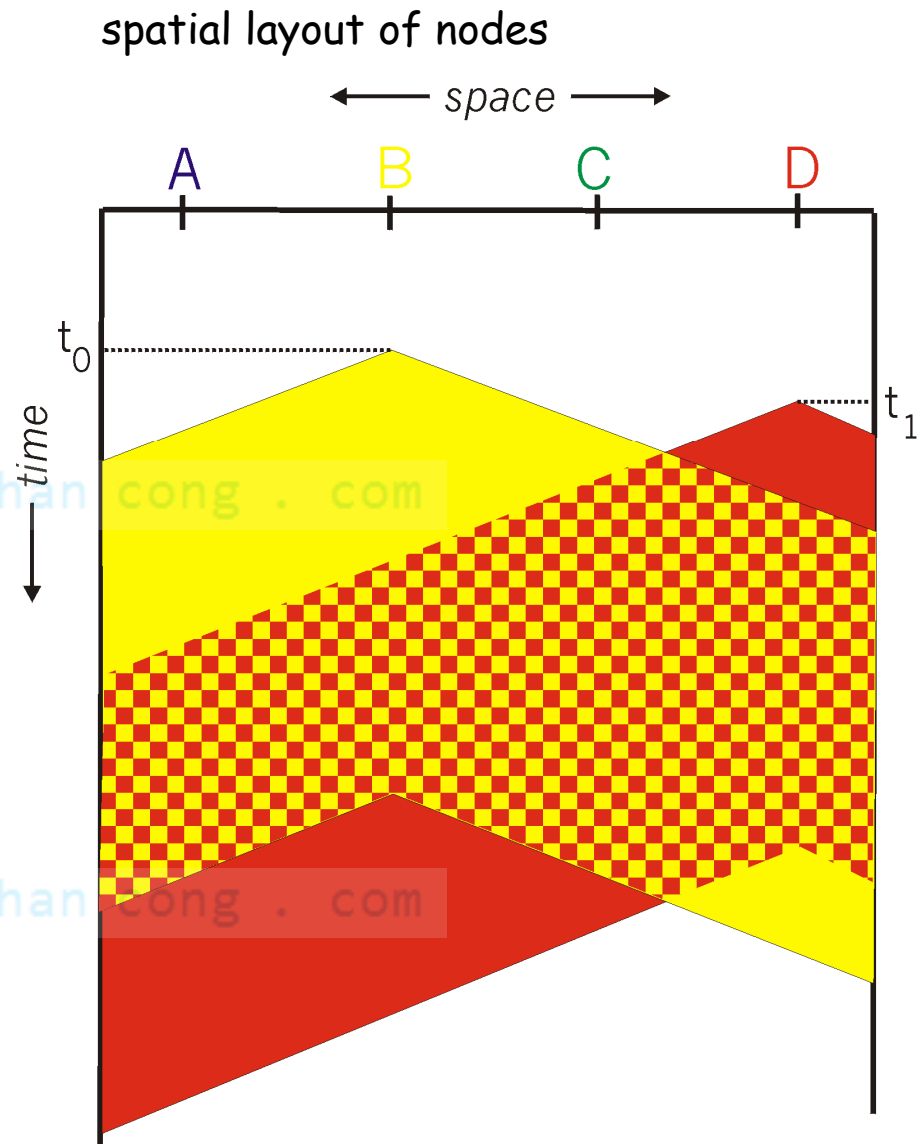
• Theo cách CSMA, sau khi truyền, va chạm vẫn *có thể* xảy ra. Đó là tình huống 1 nút ko “cảm được” đường truyền đang có frame đang trên đường đi đến.

## Khi có va chạm:

Toàn bộ công truyền gói tin bị lãng phí.

## Lưu ý:

Vai trò của khoảng cách và thời gian lan truyền có ảnh hưởng đến khả năng va chạm



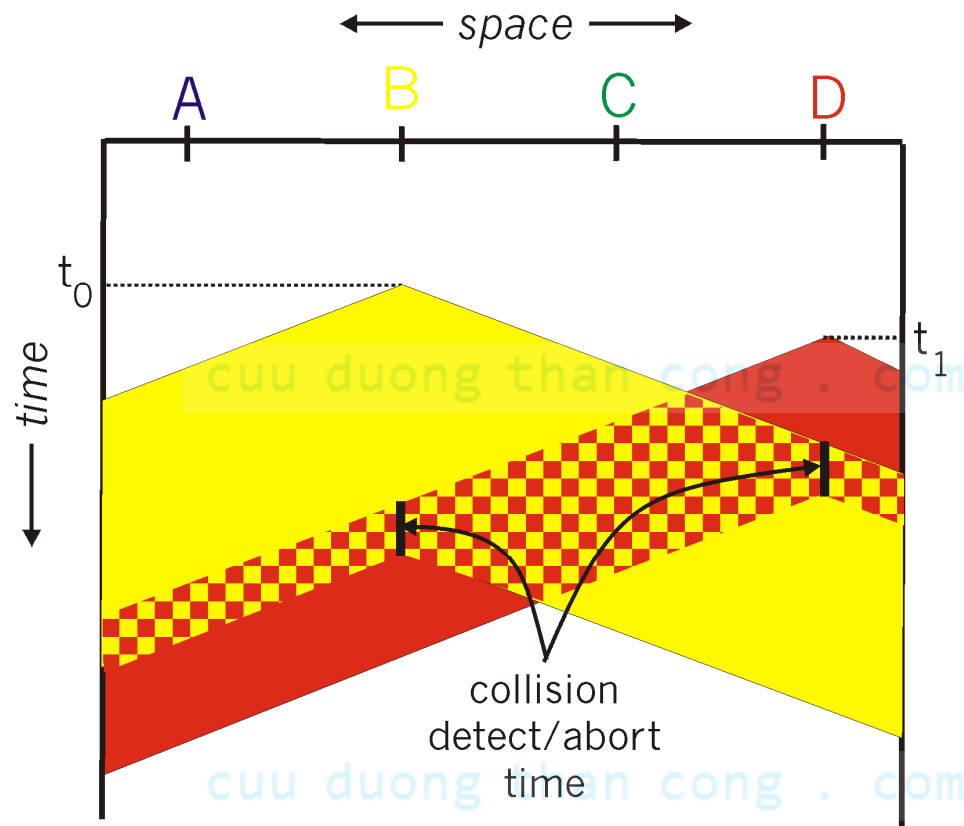
# CSMA/CD (Collision Detection)

**CSMA/CD:** carrier sensing, deferral as in CSMA

- Va chạm được phát hiện trong khoảng thời gian ngắn
- Các phiên truyền có sự va chạm đều bị bỏ ngang, giảm sự lãng phí kênh truyền.
- Phát hiện va chạm [collision detection]:
  - Dễ trong mạng LAN hữu tuyến: đo chiều dài tín hiệu, so sánh tín hiệu đã truyền và tín hiệu nhận được.
  - Khó trong mạng LAN vô tuyến: chiều dài tín hiệu nhận được bị chiều dài cuộc truyền cục bộ lấn át.
- Tương tự con người: the polite conversationalist



# CSMA/CD collision detection



# “Taking Turns” MAC protocols

Nhận xét:

Các giao thức MAC theo lối phân chia kênh truyền

- Phân chia kênh truyền một cách *hiệu quả* và *công bằng* khi có tải cao.
- Không hiệu quả khi tải thấp: chỉ cấp  $1/N$  băng thông mặc dù chỉ có 1 nút truyền!

Các giao thức MAC theo lối truy cập ngẫu nhiên

- Hiệu quả khi tải thấp: một nút có thể khai thác toàn bộ băng thông kênh truyền.
- Khi tải cao: trả giá cho hiện tượng va chạm.

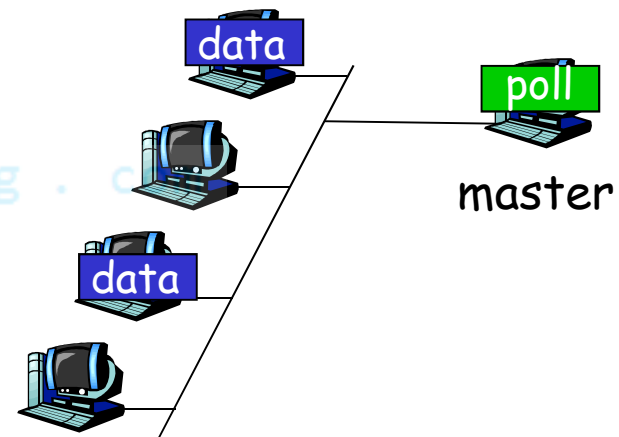
Các giao thức MAC theo lối lấy lượt (“taking turns”)

Tìm kiếm những điểm tốt nhất của 2 loại trên!

# “Taking Turns” MAC protocols

## Thăm dò nhu cầu [Polling]:

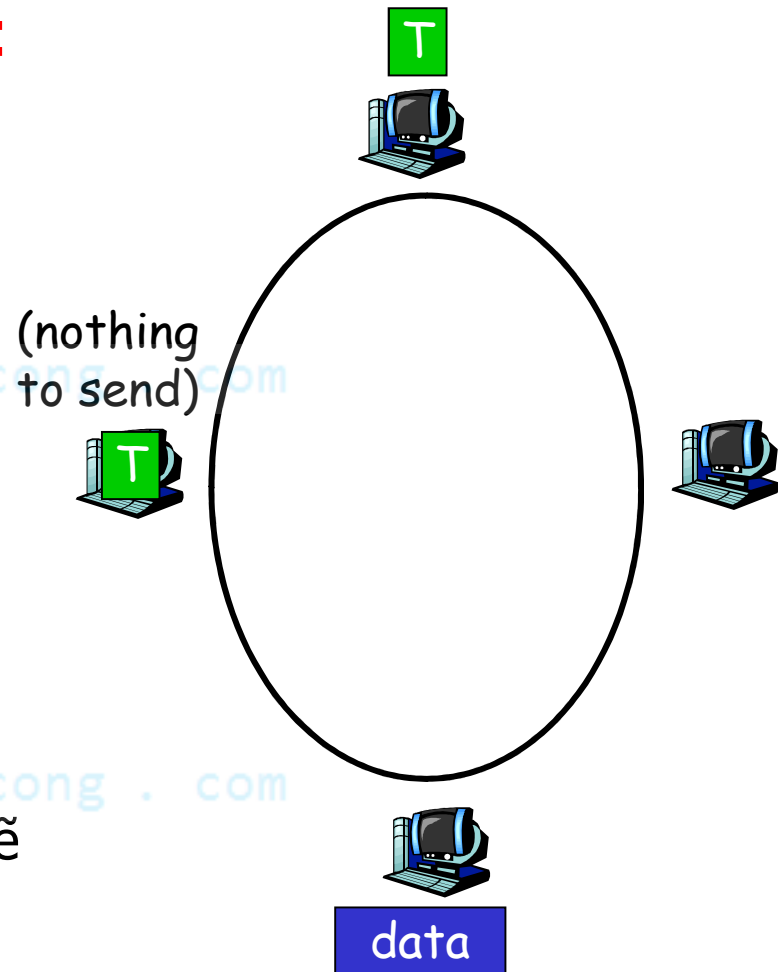
- Sử dụng 1 nút “master” làm nhiệm vụ “mời” các nút còn lại (“slave”) khi đến lượt.
- Thường được sử dụng cho các thiết bị slave “kém thông minh”
- Các vấn đề:
  - Việc master hỏi slave sẽ dẫn đến một phí tổn về thời gian=> hao băng thông.
  - Master trở thành điểm trọng yếu, nếu bị hỏng sẽ ảnh hưởng đến toàn hệ thống.



# “Taking Turns” MAC protocols

## Chuyển thẻ bài [Token passing]:

- ❑ Điều khiển thẻ bài [**token**] chuyển từ nút này đến nút khác theo thứ tự, để cấp quyền truy cập kênh truyền.
- ❑ token message
- ❑ Vấn đề:
  - Phí tổn thẻ bài
  - Độ trễ.
  - Token trở thành yếu tố quan trọng, sự mất/trùng lặp token sẽ ảnh hưởng đến toàn hệ thống.



# Summary of MAC protocols

- *Chia kênh truyền* theo thời gian, tần số hay code
  - Time Division, Frequency Division
- *Truy cập ngẫu nhiên* (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: dễ đối với các kỹ thuật hữu tuyến , nhưng lại khó đối với các kỹ thuật vô tuyến.
  - CSMA/CD được sử dụng trong Ethernet
  - CSMA/CA được sử dụng trong 802.11
- *Lấy lượt*
  - Thăm dò nhu cần truyền từ 1 điểm trung tâm
  - Chuyển thẻ bài
  - Bluetooth, FDDI, IBM Token Ring

# Link Layer

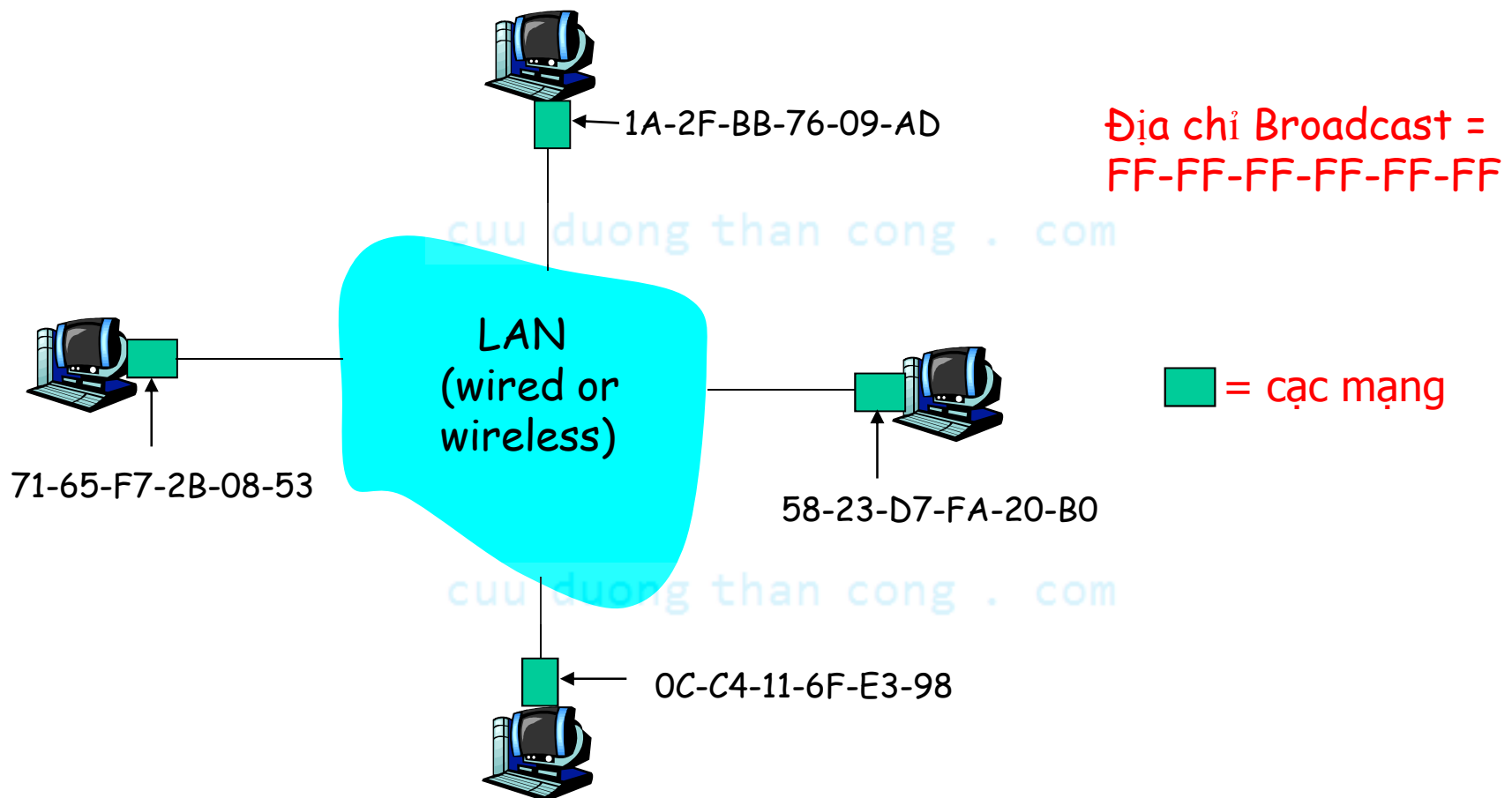
- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Link-layer switches
- 5.7 PPP
- 5.8 Link virtualization: MPLS
- 5.9 A day in the life of a web request

# MAC Addresses and ARP

- Địa chỉ IP 32-bit
  - Địa chỉ *tầng mạng*
  - Được sử dụng để đưa datagram đến IP subnet
- Địa chỉ MAC (hay LAN/vật lý/Ethernet)
  - Chức năng: chuyển frame từ 1 interface đến một interface khác có-nối-kết-vật-lý (trong cùng mạng)
  - Địa chỉ MAC có chiều dài 48 bit ( đối với hầu hết các LAN)
    - Được ghi chết (burned) vào bộ nhớ ROM của NIC

# LAN Addresses and ARP

Mỗi cạc mạng đều có 1 địa chỉ MAC duy nhất





# LAN Address (more)

- Việc cấp phát địa chỉ MAC được IEEE quản lý
- Các nhà sản xuất thiết bị mạng phải mua lại vùng không gian địa chỉ MAC (nhằm đảm bảo tính duy nhất)
- Tương tự:
  - (a) Địa chỉ MAC : tương tự số CMND
  - (b) Địa chỉ IP: tương tự địa chỉ gửi thư
- Địa chỉ MAC ko có cấu trúc phân cấp → khả chuyển
  - Có thể chuyển 1 LAN card từ 1 LAN đến 1 LAN khác.
- Địa chỉ IP phân cấp, không khả chuyển.
- IP hierarchical address NOT portable
  - Địa chỉ phụ thuộc vào subnet mà nút đó nối vào

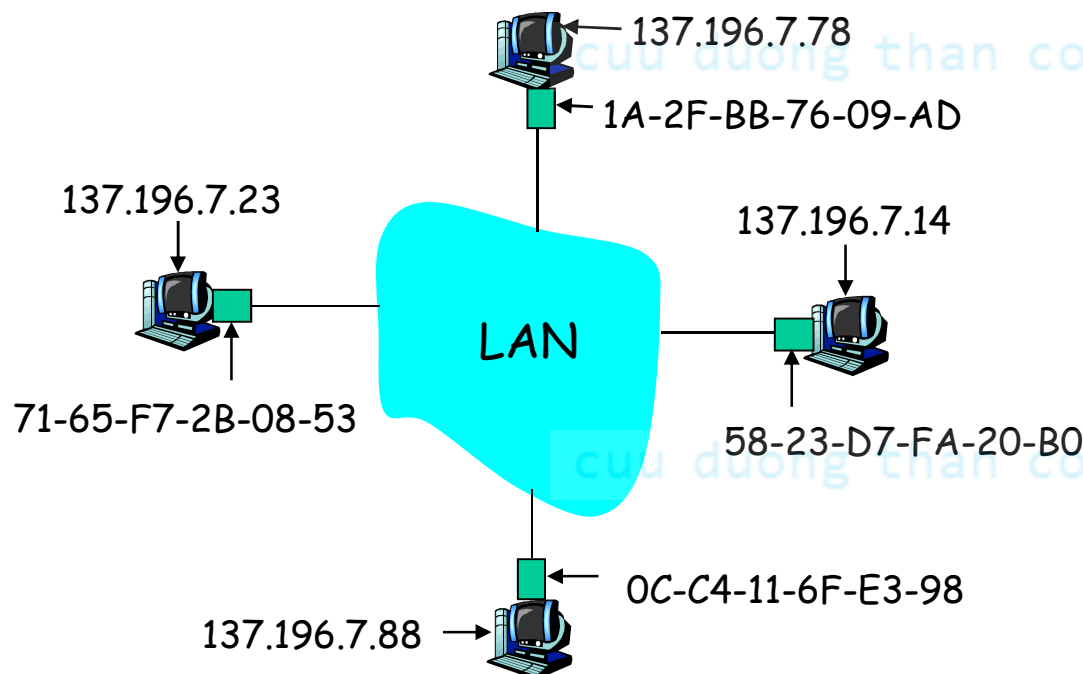
# ARP: Address Resolution Protocol

Câu hỏi: Làm thế nào để xác định được địa chỉ MAC của 1 interface khi biết địa chỉ IP của nó?

- Mỗi nút IP (host, router) trên mạng LAN đều có 1 bảng **ARP** của nó
- Bảng ARP: Lưu ảnh xạ IP/MAC của 1 số nút trên LAN

< IP address; MAC address; TTL >

- TTL (Time To Live): thời gian mà sau đó thì ảnh xạ không còn giá trị (thường khoảng 20 phút)



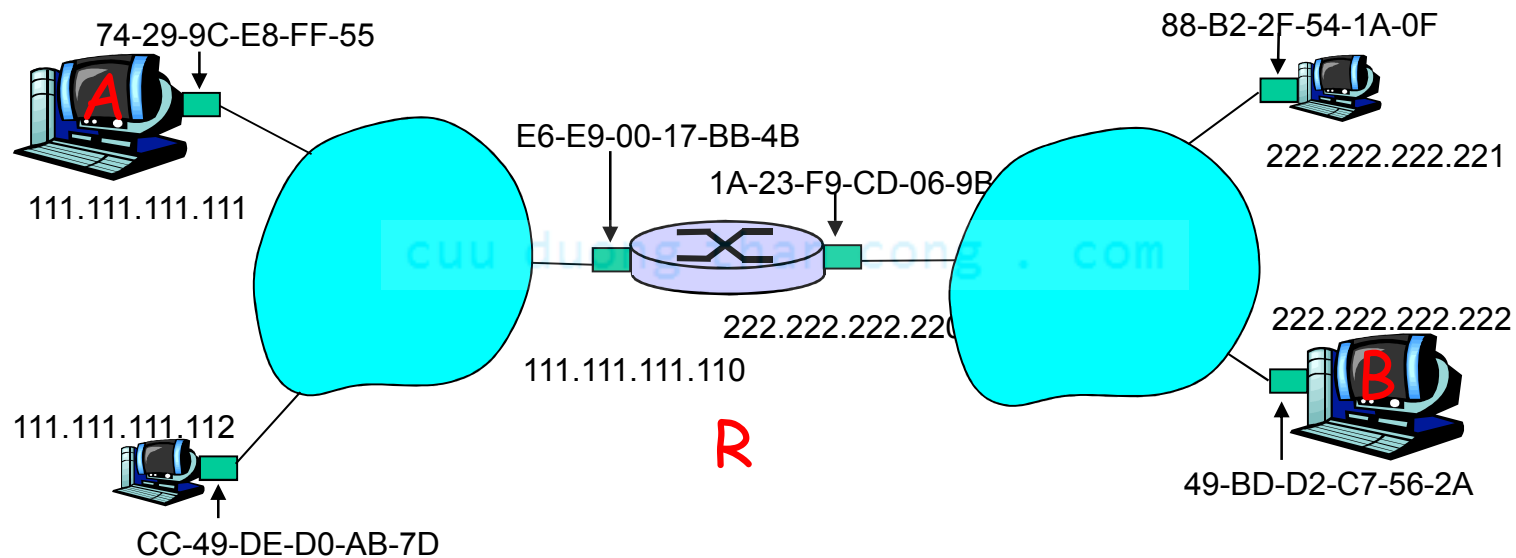
# ARP protocol: Same LAN (network)

- A muốn gửi 1 datagram cho B, và địa chỉ MAC của B không có trong bảng ARP của A.
- A gửi 1 ARP query vào địa chỉ **broadcast**, trong đó có chứa đựng địa chỉ IP của B.
  - broadcast **FF-FF-FF-FF-FF-FF**
  - Tất cả các máy trên LAN đều nhận được ARP query.
- B nhận được ARP packet, và nó trả lời cho A bằng địa chỉ MAC của chính nó (B).
  - Frame được gửi đến địa chỉ MAC của A (unicast)
- Bộ nhớ cache của A được dùng để lưu ánh xạ IP-to-MAC trong bảng ARP của nó, cho đến khi lỗi thời (time out)
  - soft state: thông tin bị lỗi thời bị vứt bỏ trừ khi nó được làm tươi [refreshed]
- ARP là giao thức “plug-and-play”:
  - Các nút duy trì bảng ARP của chính nó mà *không cần sự can thiệp của quản trị mạng*.

# Addressing: routing to another LAN

walkthrough: **gửi datagram từ A đến B via qua router R**

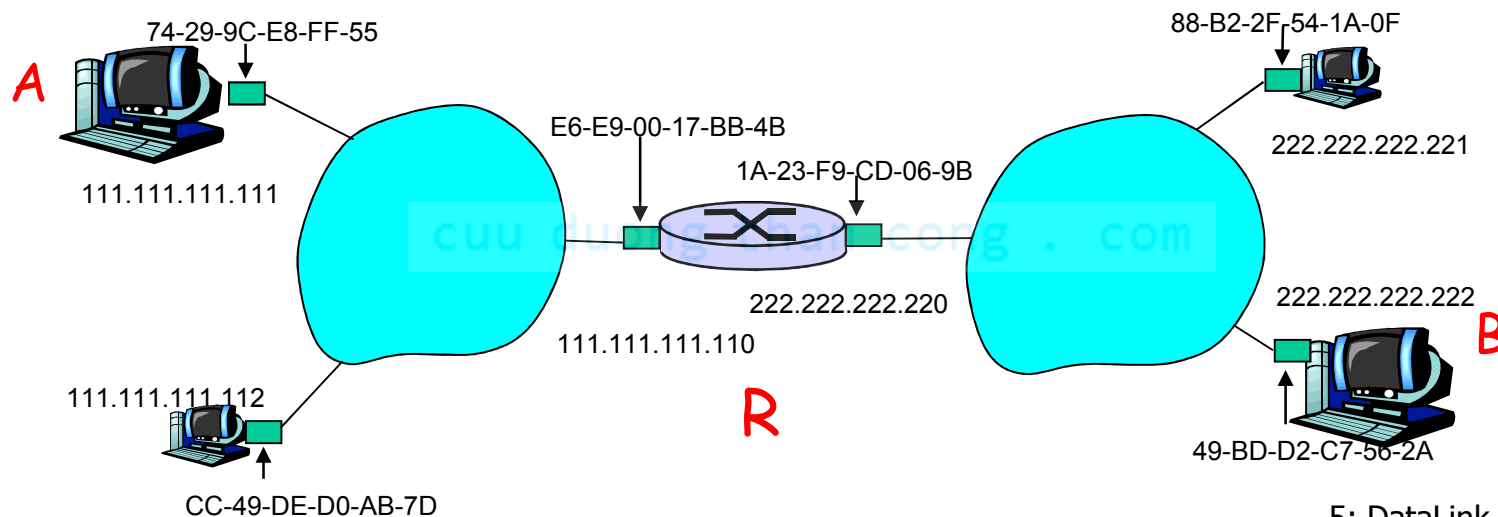
Giả sử A biết địa chỉ IP của B



- Có 2 bảng ARP tại R, mỗi bảng cho 1 mạng con IP

- A tạo ra IP Datagram với (source A, destination B )
- A sử dụng ARP, xác định được địa chỉ MAC của R là 111.111.111.110
- A tạo ra frame ở tầng link, với đích là địa chỉ MAC của R, và frame này chứa đựng IP Datagram mà A gửi cho B.
- Các mạng của A gửi frame đi.
- Các mạng của R nhận frame.
- R gỡ IP datagram ra từ Ethernet frame, đọc lấy địa chỉ đích (là địa chỉ IP của B)
- R sử dụng ARP để nhận được địa chỉ MAC của B.
- R tạo ra frame chứa IP datagram (mà A gửi cho B) và gửi frame đến B.

This is a **really** important example - make sure you understand!



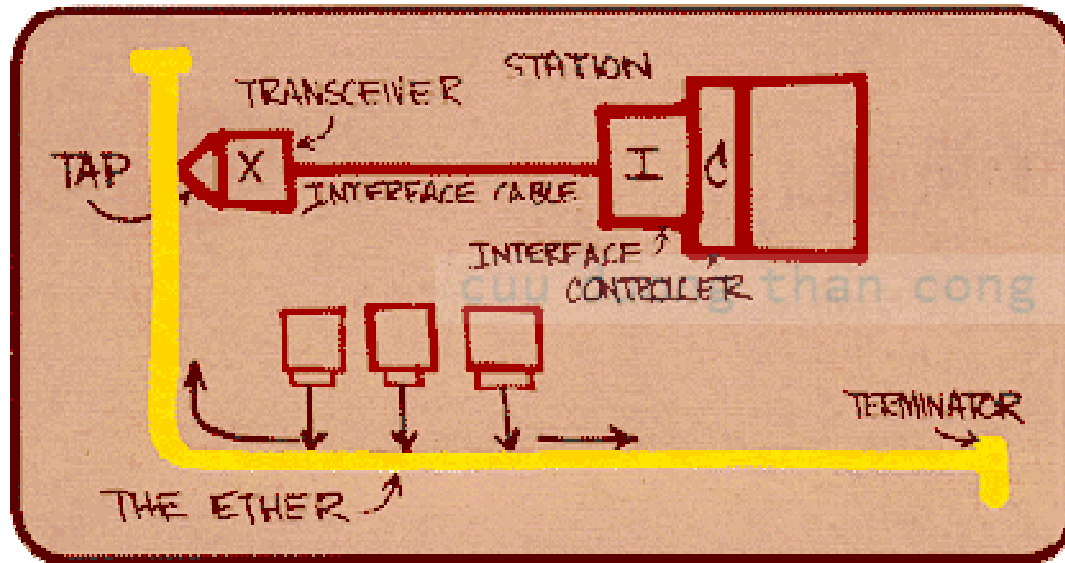
# Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Link-layer switches
- 5.7 PPP
- 5.8 Link virtualization: MPLS
- 5.9 A day in the life of a web request

# Ethernet

Là công nghệ LAN hữu tuyến “thống trị”

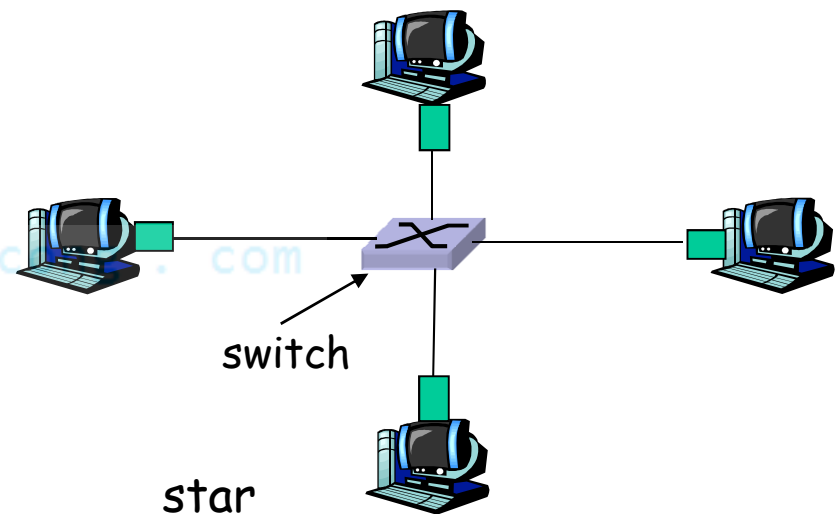
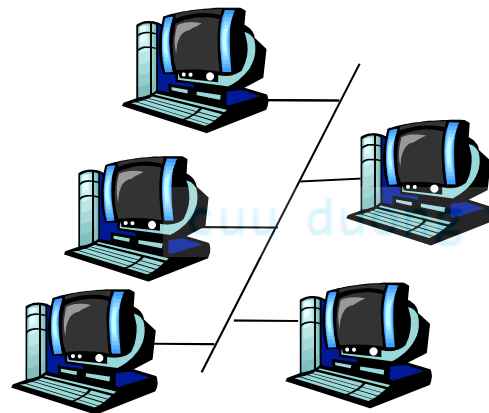
- Giá rẻ (20\$ cho NIC)
- Là công nghệ LAN đầu tiên được sử dụng rộng rãi
- Đơn giản hơn, chi phí thấp hơn LAN sử dụng token và ATM
- Khả năng truyền tải: 10 Mbps – 10 Gbps



Metcalfe's Ethernet sketch

# Star topology

- Topo dạng bus được sử dụng phổ biến cho đến giữa thập niên 1990
  - Tất cả các nút nằm cùng trên một collision domain (vùng có thể có sự va chạm giữa các gói tin)
- Ngày nay: topo dạng hình sao (star) được sử dụng phổ biến.
  - Sử dụng *switch* trung tâm để nối các trạm.
  - Các trạm truyền frame trên những đường tách biệt, không có sự va chạm với các trạm khác.

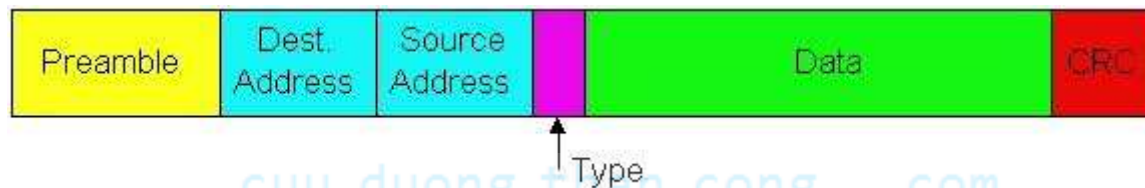


5: DataLink Layer 5-48



# Ethernet Frame Structure

Các mạng bên gửi đóng gói IP datagram ( hoặc các gói tin khác thuộc tầng mạng) vào trong Ethernet frame

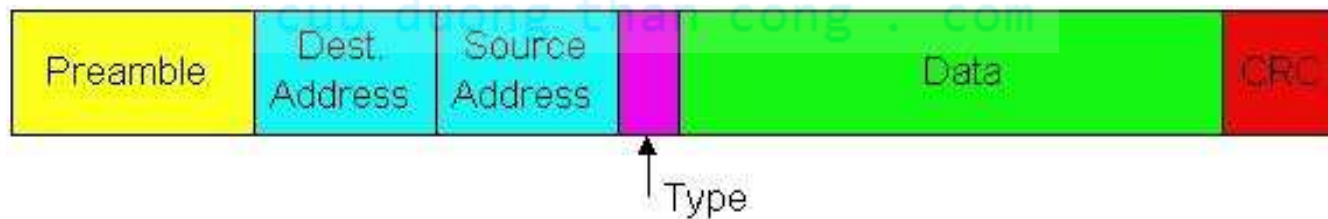


**Preamble:** 8 bytes

- 7 bytes đầu có giá trị 10101010 và byte kế 10101011
  - Được sử dụng để đồng bộ bên nhận và bên gửi
- cuu duong than cong . com

# Ethernet Frame Structure (more)

- **Addresses:** 6 bytes
  - Nếu các mạng nhận được frame có địa chỉ đích khớp với địa chỉ của nó hoặc nhận được frame có địa chỉ đích là broadcast (eg gói tin ARP), nó sẽ đẩy dữ liệu bên trong frame lên giao thức tầng mạng.
  - Ngược lại, các mạng vứt bỏ frame
- **Type:** chỉ rõ giao thức tầng trên ( thường là IP, nhưng cũng có thể khác, VD Novell IPX, AppleTalk)
- **CRC:** được bên nhận dùng để kiểm có lỗi truyền hay không. Nếu có thì frame sẽ bị vứt bỏ.



# Ethernet: Unreliable, connectionless

- **connectionless**: không có nghi thức bắt tay giữa NIC bên gửi và NIC bên nhận.
- **unreliable**: NIC bên nhận không gửi gói tin ACK hay NACK cho NIC bên gửi.
  - Dòng datagram đưa lên tầng mạng có thể bị khuyết datagram
  - Chỗ khuyết sẽ được “lấp đầy” nếu ứng dụng sử dụng TCP.
  - Nếu không, ứng dụng sẽ thấy được chỗ khuyết này.
- Giao thức truy cập đường truyền của Ethernet: unslotted **CSMA/CD**

# Ethernet CSMA/CD algorithm

1. NIC nhận datagram từ tầng mạng, nó tạo ra frame
  2. Nếu NIC cảm nhận được kênh truyền đang rảnh, nó bắt đầu truyền frame. Nếu NIC cảm nhận kênh truyền đang bận, nó chờ cho đến khi rảnh, sau đó mới truyền.
  3. Nếu NIC truyền toàn bộ frame mà không phát hiện ra một cuộc truyền nào khác, NIC hoàn thành việc truyền frame.
  4. Nếu NIC phát hiện ra có một cuộc truyền khác cũng đang truyền, nó ngưng truyền và gửi 1 tín hiệu cảnh báo [Jam signal]
  5. Sau khi ngưng truyền, NIC chuyển sang tình trạng **exponential backoff**: sau lần va chạm thứ  $n$ , NIC sẽ chọn một số  $K$  ngẫu nhiên từ vùng  $\{0, 1, 2, \dots, 2^m - 1\}$ . NIC chờ đợi  $K * 512$  bit times, quay trở lại bước 2
- Trong đó  $m = \min(n, 10)$

# Ethernet's CSMA/CD (more)

**Jam Signal:** nhằm chắc chắn rằng tất cả các bên truyền khác đều biết về sự kiện va chạm; dài 48 bits.

**Bit time:** thời gian để truyền 1 bit. Với Ethernet 10 Mbps, bit time = 0.1 microsec

Với  $K=1023$ , thời gian chờ đợi khoảng 50 msec

See/interact with Java applet on AWL Web site: highly recommended!

## **Exponential Backoff:**

- *Mục tiêu:* thời gian chờ truyền lại phải phù hợp với tải hiện hành
  - Tải càng lớn: thời gian chờ càng lâu
- Va chạm lần 1: chọn K từ  $\{0,1\}$ ; thời gian chờ là  $K \cdot 512$  bit time.
- Sau lần va chạm thứ 2: chọn K từ  $\{0,1,2,3\}$ ...
- Sau lần va chạm thứ 10, chọn K từ  $\{0,1,2,3,4,...,1023\}$

# CSMA/CD efficiency

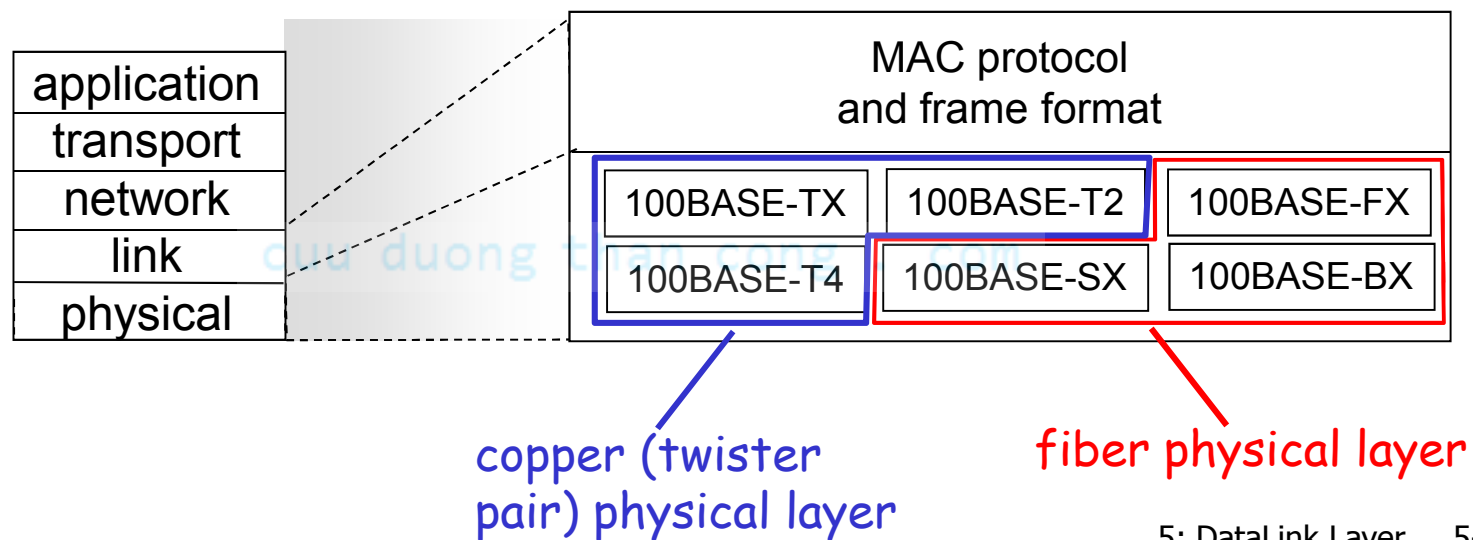
- $T_{\text{prop LAN}}$  = thời gian lan truyền lớn nhất giữa 2 nút trên LAN
- $t_{\text{trans}}$  = thời gian để truyền frame có kích thước lớn nhất.

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

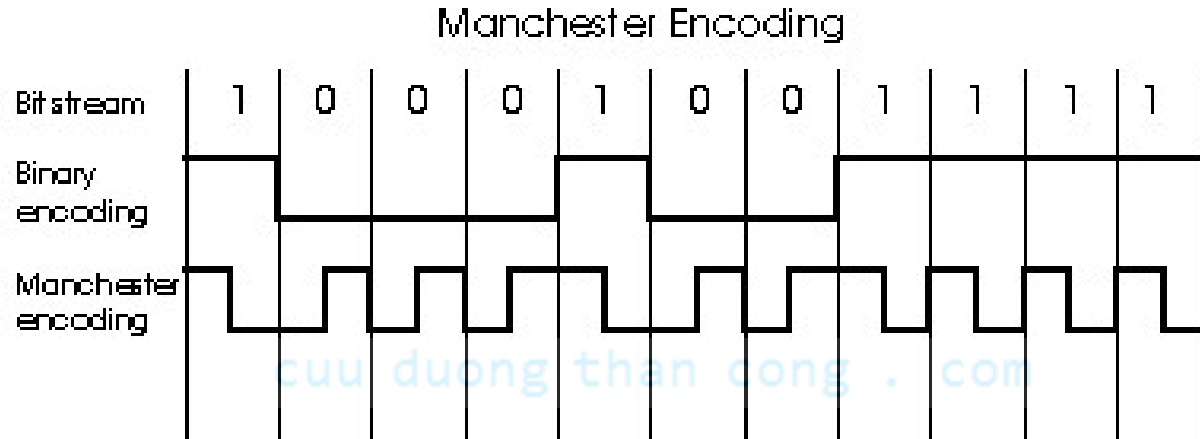
- Độ hiệu quả *efficiency* tiến về 1
  - Khi  $t_{\text{prop}}$  tiến về 0
  - Khi  $t_{\text{trans}}$  tiến về  $\infty$
- Hiệu quả hơn ALOHA, và lại đơn giản, chi phí thấp, phi tập trung!

## 802.3 Ethernet Standards: Link & Physical Layers

- *Có nhiều* chuẩn Ethernet khác nhau
  - Có chung giao thức MAC và khung dạng frame
  - Tốc độ khác nhau: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - Phương tiện truyền khác nhau: cáp quang, cáp xoắn đôi, cáp đồng trục.



# Manchester encoding



- Được sử dụng trong chuẩn 10BaseT
- Mỗi bit đều có chuyển trạng thái.
- Cho phép đồng hồ bên gửi và bên nhận đồng bộ lẫn nhau.
  - Không cần thiết phải sử dụng đồng hồ toàn cục cho tất cả các nút.



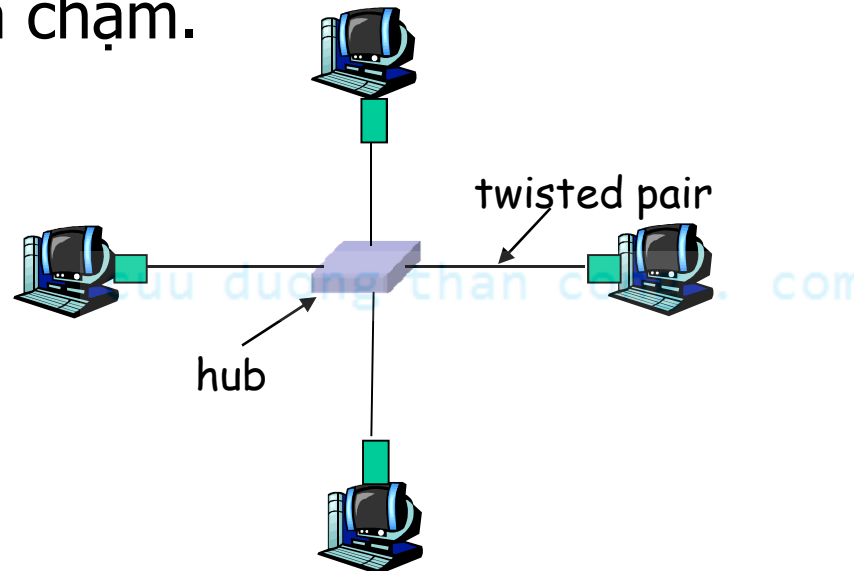
# Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-layer Addressing
- 5.5 Ethernet
- 5.6 Link-layer switches, LANs, VLANs
- 5.7 PPP
- 5.8 Link virtualization: MPLS
- 5.9 A day in the life of a web request

# Hubs

... là repeater hoạt động ở tầng vật lý:

- Các bit đi vào ở 1 liên-kết và ra **tất cả** các liên kết khác với cùng tốc độ.
- Tất cả các nút kết nối với hub có thể va chạm với một nút khác.
- Không có vùng đệm cho frame
- Không có CSMA/CD tại hub: các NICs phải tự phát hiện va chạm.

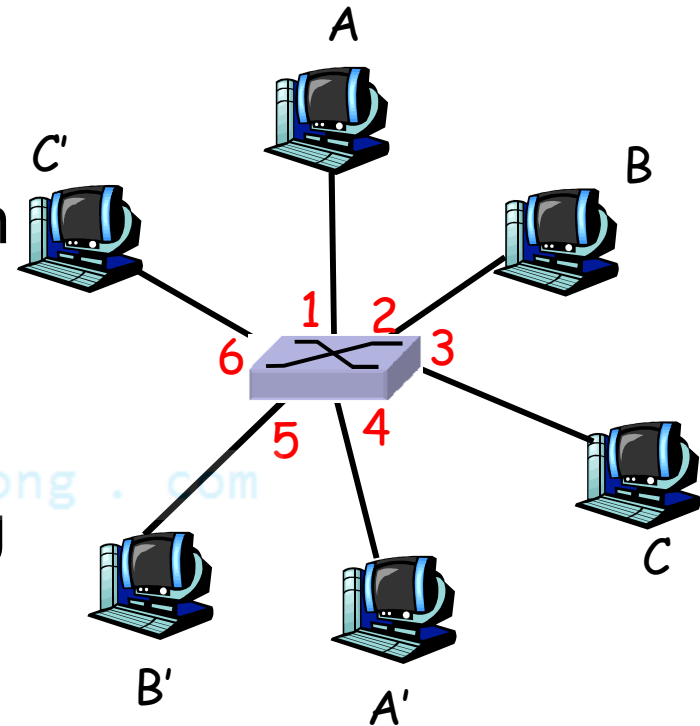


# Switch

- Là thiết bị ở tầng link: hubs thông minh, đóng vai trò *chủ động* trong:
  - Lưu và chuyển tiếp các Ethernet frames
  - Kiểm tra địa chỉ MAC của frame đến, chuyển tiếp một cách có lựa chọn frame đến một hoặc nhiều liên-kết ở ngõ ra, sử dụng CSMA/CD để truy cập đường truyền.
- *Tính trong suốt của switch*
  - Host gửi dữ liệu cho nhau, không cần biết đến sự hiện diện của switch.
- *plug-and-play, self-learning*
  - Gắn vào là chạy, switches không cần được cấu hình

## Switch: allows *multiple* simultaneous transmissions

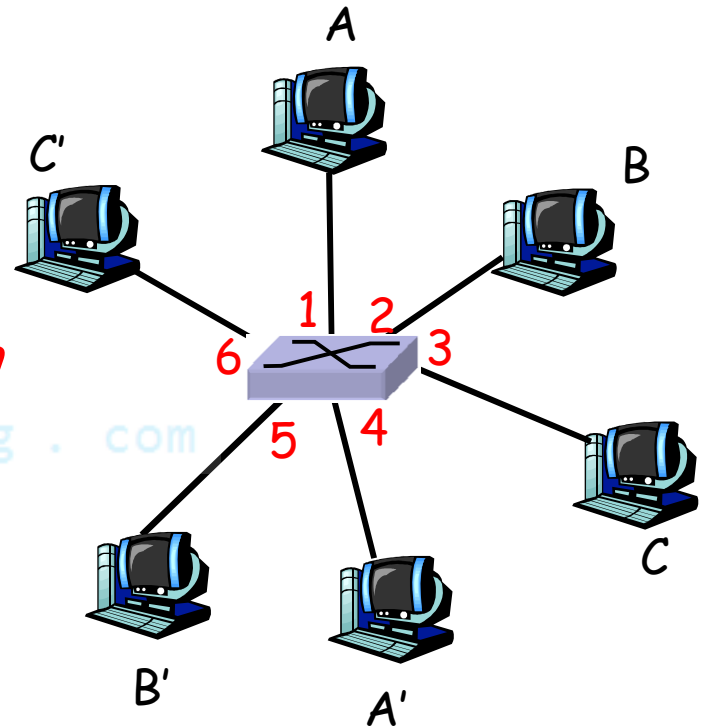
- Mỗi host có đường kết nối dành riêng, trực tiếp đến switch.
- Switch có bộ đệm các gói tin
- Ethernet được sử dụng trên từng liên-kết vào, nhưng không có va chạm; cho phép dữ liệu truyền theo 2 chiều đồng thời [full-duplex]
  - Mỗi liên-kết là collision domain của chính nó.
- **switching:** A-tới-A' và B-tới-B' một cách đồng thời, không có va chạm.
  - Điều này là không thể đ/v Hub



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

## Switch Table

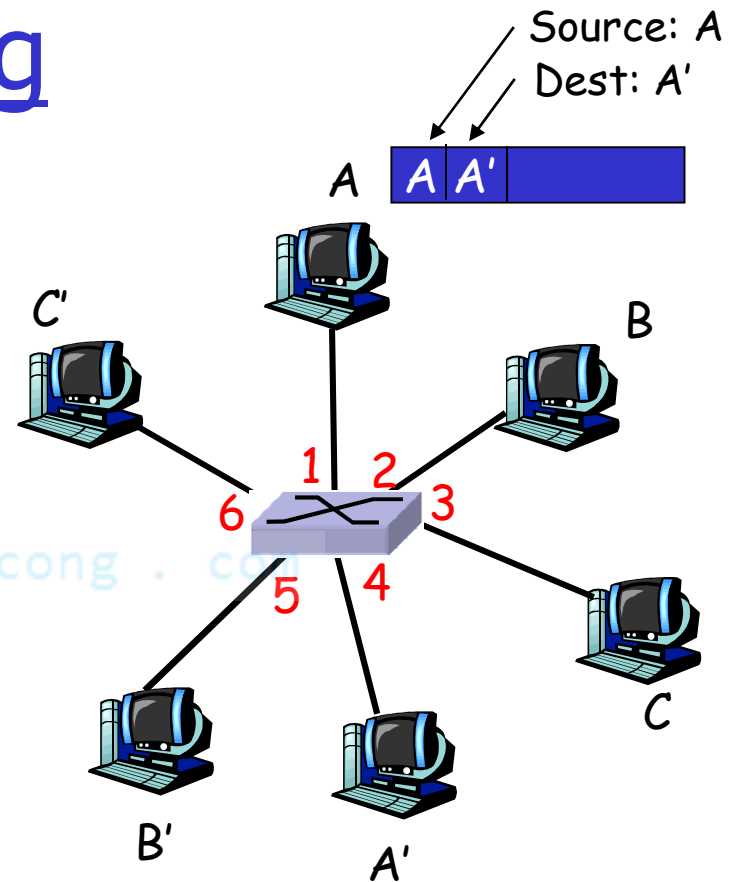
- Q: làm thế nào switch biết rằng A' là có thể "chạm đến được" thông qua interface 4, và B' là có thể "chạm đến được" nhờ interface 5?
- A: mỗi switch có một *bảng chuyển* [switch table], mỗi dòng của bảng chứa:
  - (địa chỉ MAC của host, interface để đến được host, nhãn thời gian)
- Trông giống như bảng định tuyến!
- Q: các dòng của bảng chuyển được tạo ra và duy trì như thế nào?
  - Có vài điều tương tự như giao thức định tuyến.



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

# Switch: self-learning

- switch *tự học* để biết những hosts nào là có thể đến được qua interface nào của nó.
  - Khi nhận được frame, switch “học” vị trí của host gửi: từ nhánh mạng đến switch.
  - Switch ghi nhận lại cặp thông tin về host gửi (MAC addr) và nhánh mạng chứa host gửi (interface)



MAC addr	interface	TTL
A	1	60

*Switch table  
(initially empty)*

# Switch: frame filtering/forwarding

## Khi nhận được frame:

1. Ghi nhận lại liên-kết (đại diện bằng interface) mà host gửi (đại diện bằng MAC addr) sử dụng.

2. Cập nhật *bảng chuyển*, chỉ mục theo "MAC dest address"

**3. If** có dòng trên bảng chuyển ứng với đích đến của frame  
**then {**

**if** đích đến nằm cùng segment với frame đến

**then** hủy bỏ frame /\* filter \*/

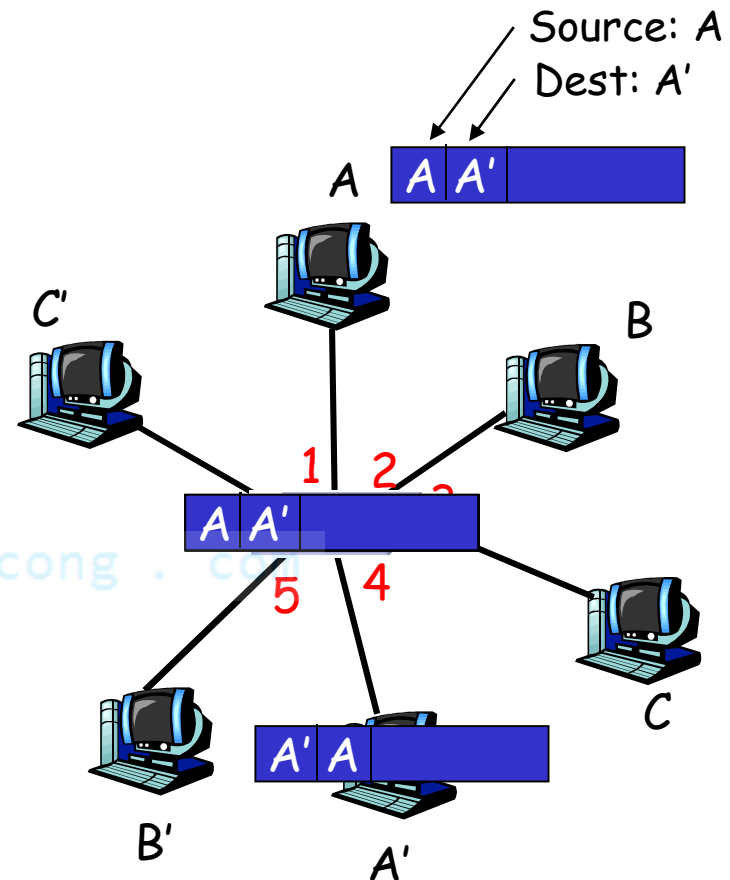
**else** chuyển frame đến interface được chỉ ra  
/\*forward\*/

**}**

**else** broadcast frame đến tất cả các interface, ngoại trừ interface mà frame đi vào /\***flooding**\*/

# Self-learning, forwarding: example

- Khi không biết frame destination: *flood*
- Khi xác định được: *gửi có chọn lọc*



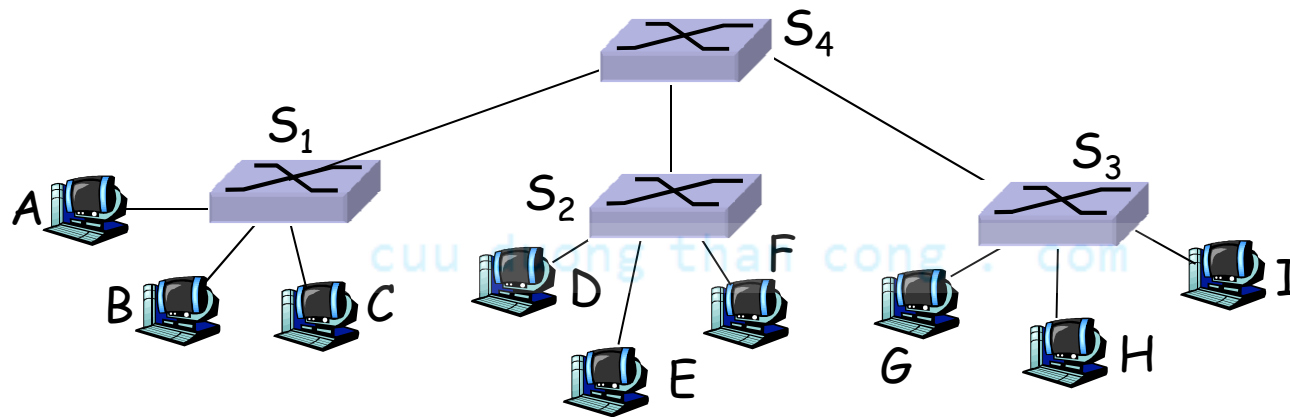
MAC addr	interface	TTL
A	1	60
A'	4	60

*Switch table  
(initially empty)*



# Interconnecting switches

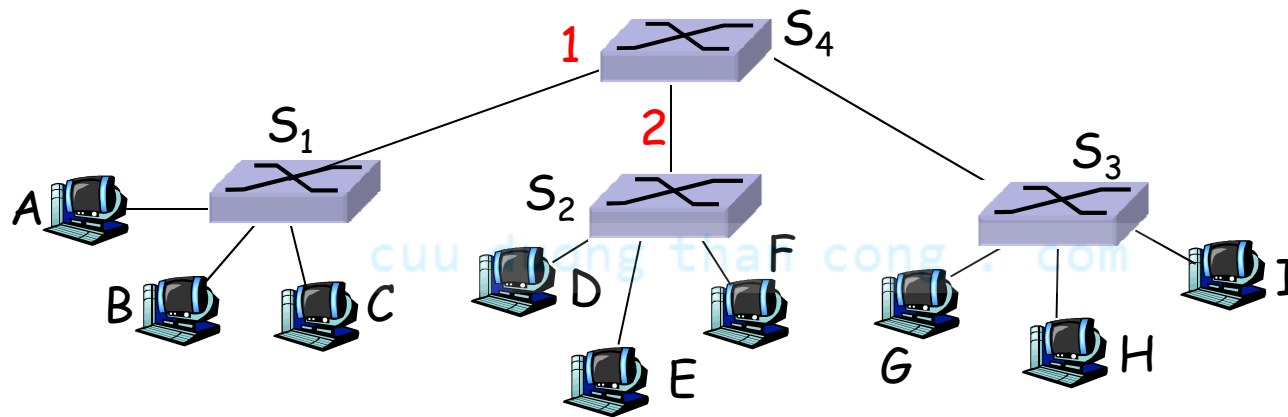
- Các switch có thể nối kết lại với nhau:



- ❑ Q: gửi từ A đến G – làm thế nào để S1 biết phải chuyển frame đến G thông qua S4 và S3?
- ❑ A: tự học! (làm việc giống như trường hợp có 1 switch)

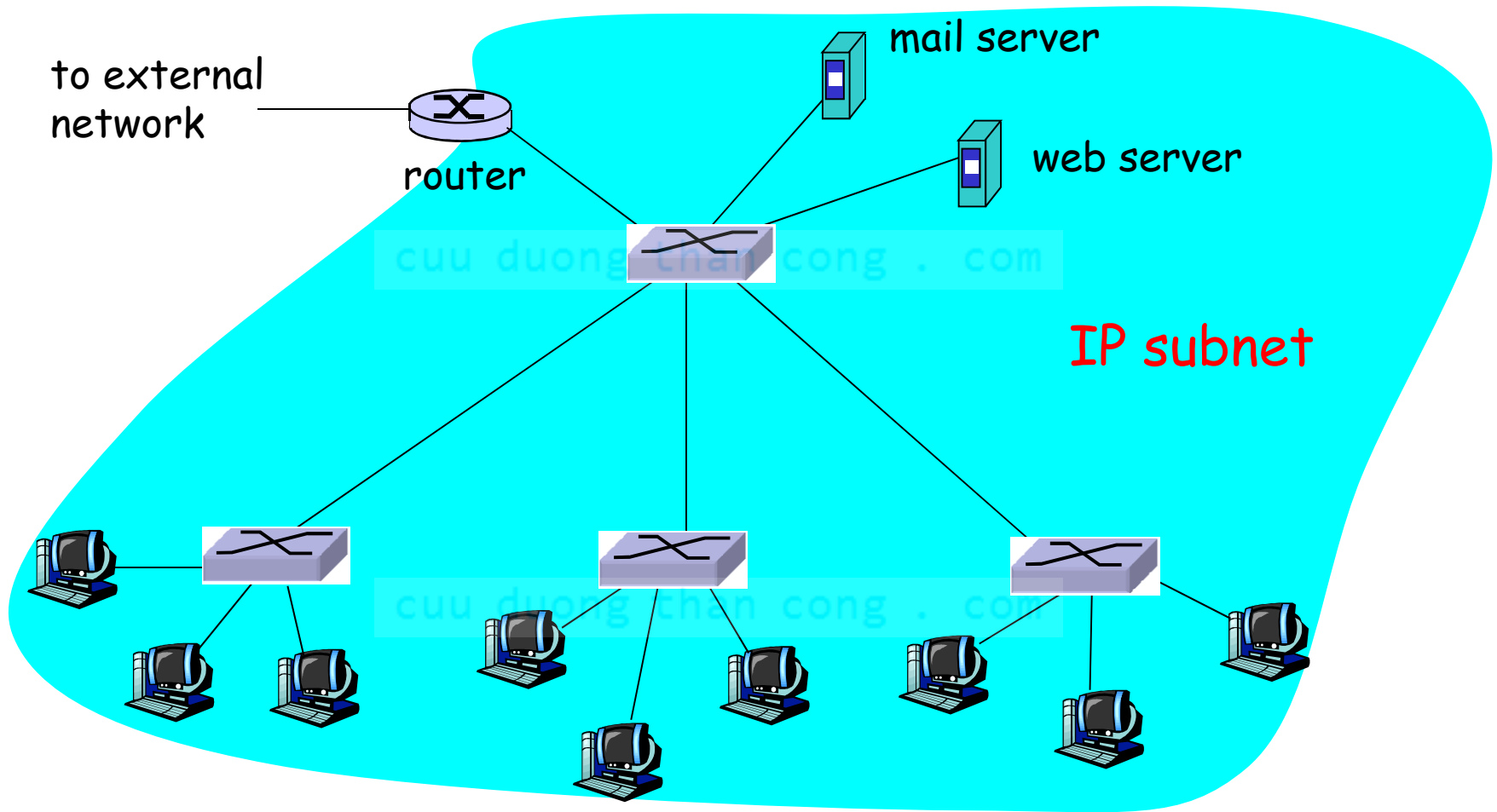
# Self-learning multi-switch example

Giả sử C gửi frame đến I, I phản hồi lại C.



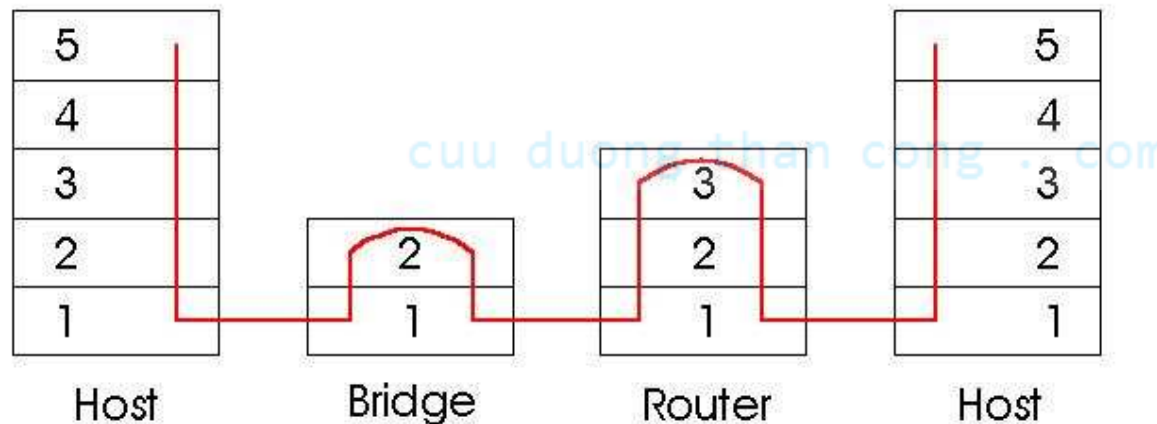
- Q: hãy chỉ ra bảng chuyển và cách chuyển frame tại các switch S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub> ?

# Institutional network



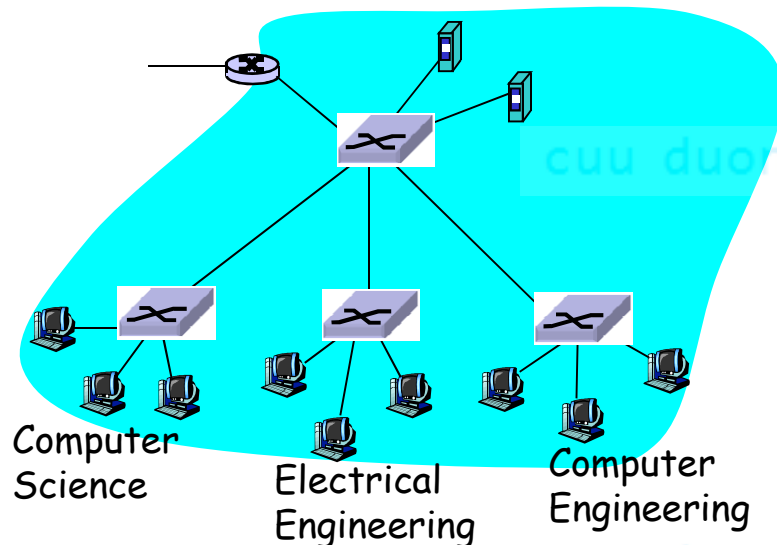
# Switches vs. Routers

- Cả hai đều là thiết bị lưu và chuyển [store-and-forward]
  - routers: thiết bị tầng Network (xử lý header ở tầng Network)
  - Switches: Thiết bị tầng Link
- Router duy trì bảng Định tuyến, cài đặt thuật toán định tuyến.
- Switch duy trì bảng Chuyển, cài đặt chức năng lọc [filtering], thuật toán tự học.



# VLANs: motivation

*Mạng như sơ đồ sau có gì BẤT ỔN?*



Điều gì sẽ xảy ra nếu:

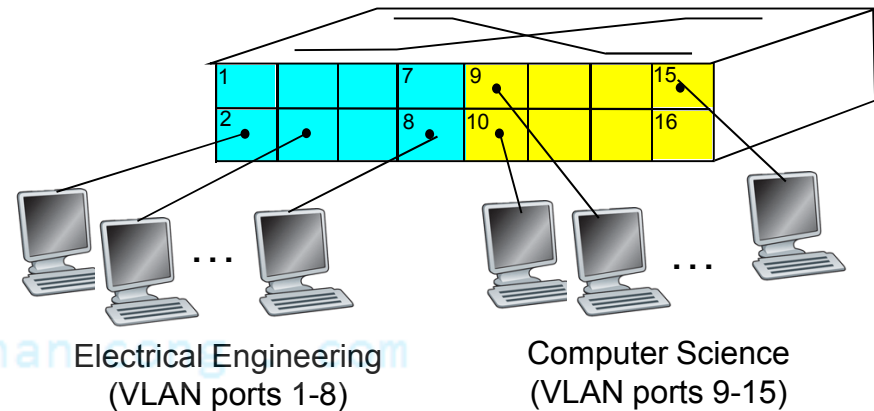
- Người dùng mạng CS đến làm việc tại mạng EE, nhưng vẫn muốn kết nối đến switch CS?
- Nếu chỉ 1 broadcast domain:
  - Tất cả các dữ liệu broadcast ở tầng 2 (ARP, DHCP) được gửi đến tất cả các LAN (này sinh vấn đề về an ninh/chính sách riêng tư)
- Switch ở mức thấp nhất chỉ có vài port được dùng=> kém hiệu quả.

# VLANs

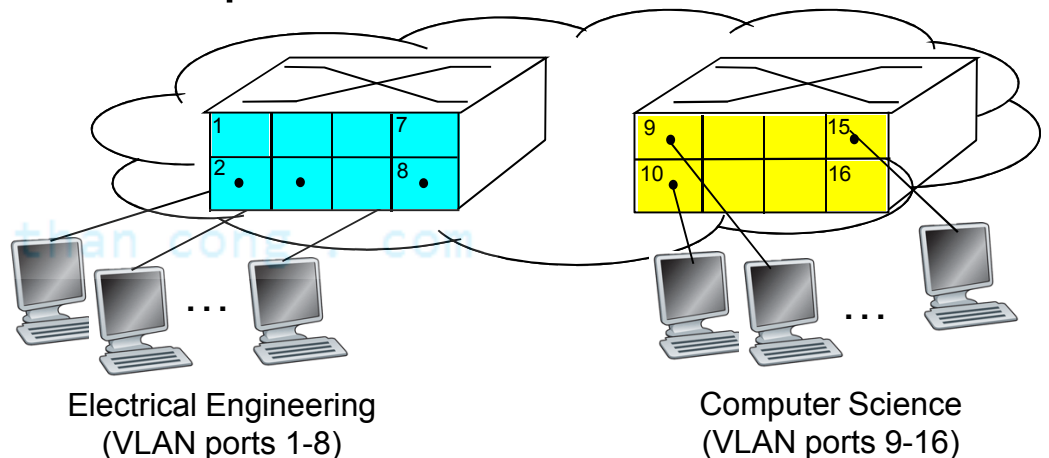
## Virtual Local Area Network

Switch hỗ trợ khả năng VLAN có thể được cấu hình để định nghĩa nhiều mạng LAN ảo trên 1 hạ tầng LAN.

**Port-based VLAN:** các port của switch được nhóm lại (nhờ phần mềm quản lý) để một switch vật lý cho phép ...

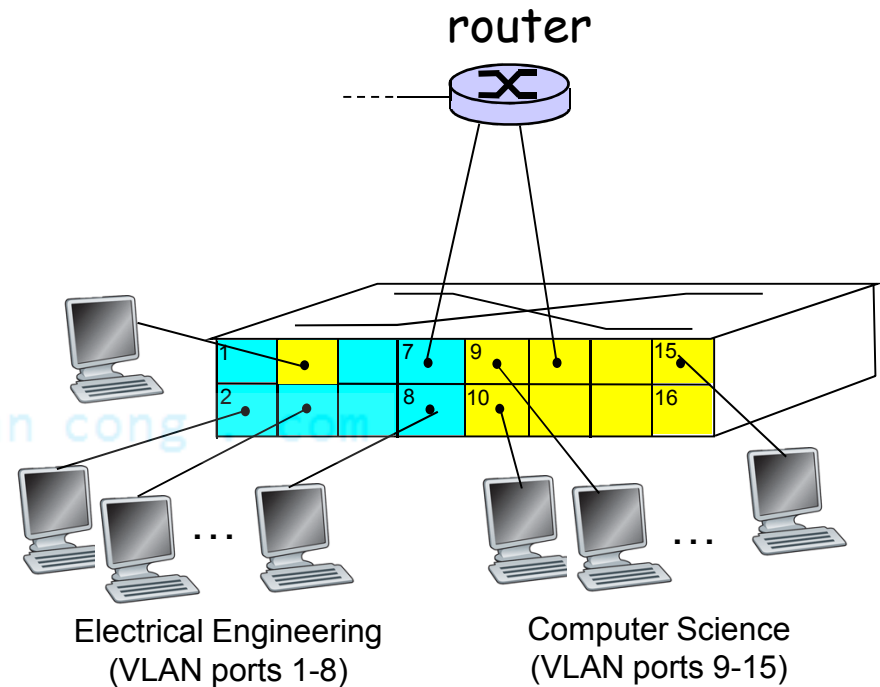


... làm việc như có **nhiều switch ảo**.

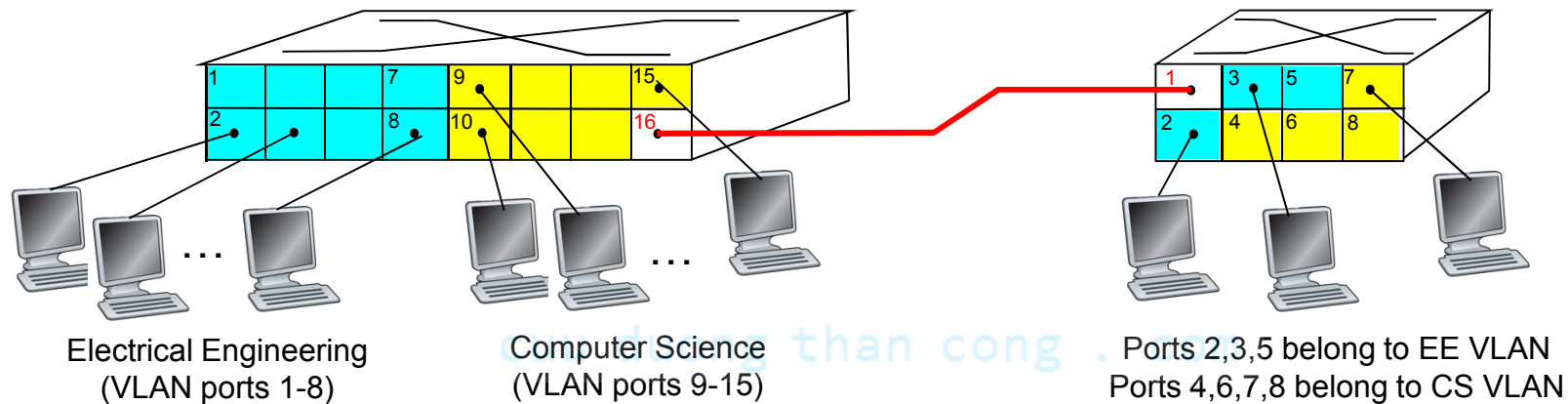


# Port-based VLAN

- **Tách rời luồng dữ liệu:**  
frames đến/từ ports 1-8 chỉ có thể đến ports 1-8
  - Cũng có thể định nghĩa VLAN dựa vào địa chỉ MAC của host thay vì port của switch.
- ❑ **dynamic membership:** các ports có thể được gán động cho các VLAN.
- ❑ **forwarding between VLANs:** được thực hiện thông qua định tuyến (như với các switch rời)
  - Thực tế, các nhà SX bán các thiết bị switch tích hợp với router.



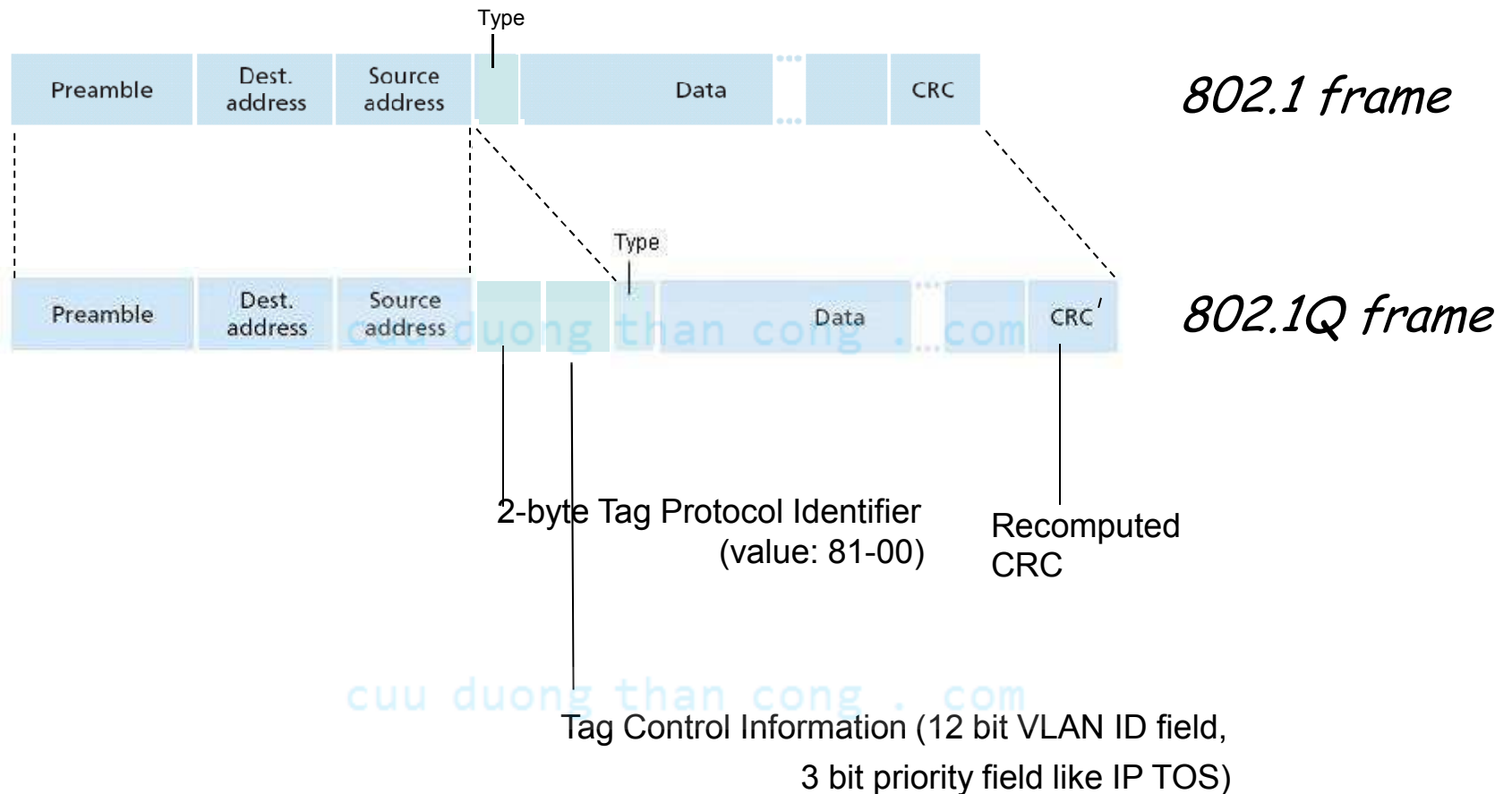
# VLANS spanning multiple switches



- **trunk port:** di chuyển frame giữa các VLAN được định nghĩa trên các switch vật lý.
  - Các frame được trong VLAN giữa các switch không thể chỉ có cấu trúc như 802.1 frame (vì phải có thông tin VLAN ID)
  - Giao thức 802.1q thêm/gỡ bỏ các field trong header của frame cần được chuyển giữa các trunk ports.



# 802.1Q VLAN frame format



# Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Link-layer switches
- 5.7 PPP
- 5.8 Link virtualization: MPLS
- 5.9 A day in the life of a web request

# Point to Point Data Link Control

- Một gửi, một nhận, trên 1 liên-kết: dễ hơn liên-kết quảng bá
  - Không cần điều khiển truy cập đường truyền
  - Không cần địa chỉ MAC
  - e.g., dialup link, ISDN line
- Các giao thức point-to-point phổ biến
  - PPP (point-to-point protocol)
  - HDLC: High level data link control (Data link used to be considered “high layer” in protocol stack!)

## PPP Design Requirements [RFC 1557]

- **packet framing:** đóng gói datagram ở tầng mạng vào frame ở tầng link.
  - Mang dữ liệu của bất kỳ 1 giao thức tầng mạng nào (không chỉ IP)
  - Khả năng chuyển dữ liệu lên tầng trên
- **bit transparency:** phải cho phép mang bất cứ dữ liệu nào trong trường chứa dữ liệu (e.g. trường info trong PPP data frame)
- **error detection** (nhưng không sửa lỗi)
- **connection liveness:** phải có khả năng phát hiện lỗi trên liên-kết và báo lỗi này cho tầng mạng.
- **network layer address negotiation:** một nút phải có cung cấp 1 cơ chế cho phép truyền thông với tầng mạng (VD IP) để học/cấu hình địa chỉ mạng của nút khác.

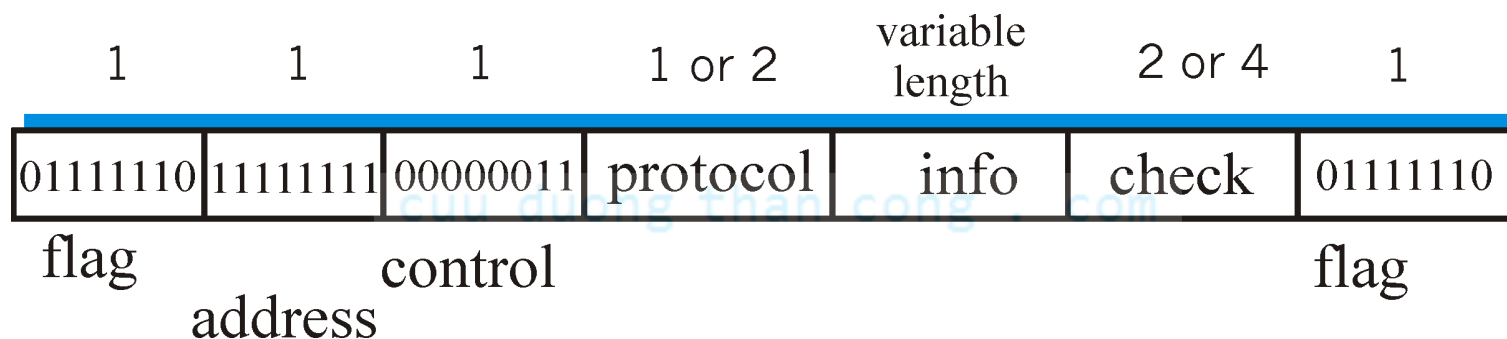
# PPP non-requirements

- PPP có phát hiện nhưng không đòi cơ chế sửa lỗi/khôi phục nguyên trạng
- PPP không cần điều tiết tốc độ truyền/nhận (no flow control). Việc điều tiết tốc độ được các giao thức tầng trên đảm nhiệm.
- Chấp nhận truyền nhận các frame sai thứ tự, điều này tương thích với mô hình dịch vụ của IP.
- PPP chỉ vận hành trên các liên-kết điểm-điểm (một gửi-một nhận). Trong khi đó, các giao thức khác ở tầng link (VD HDLC) có thể vận hành trên các liên-kết quảng-bá (một-gửi-nhiều- nhận)

**Error recovery, flow control, data re-ordering  
all relegated to higher layers!**

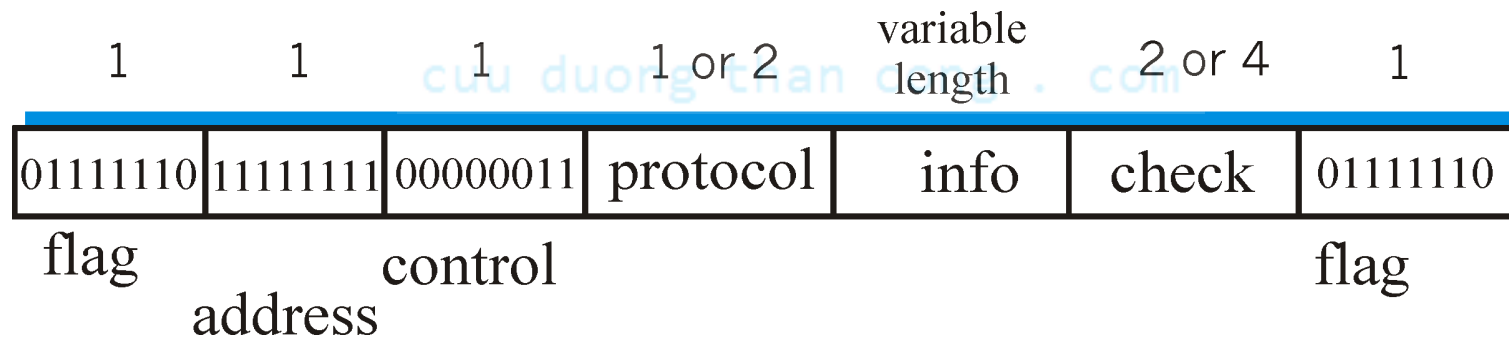
# PPP Data Frame

- **Flag:** dấu hiệu phân cách (bắt đầu và kết thúc frame)
- **Address:** hiện cố định 11111111, không sử dụng.
- **Control:** hiện cố định 00000011, không sử dụng, tương lai có thể định nghĩa các giá trị khác.
- **Protocol:** giao thức tầng trên chuyển dữ liệu vào frame. ( $21_{16}$ : IP;  $29_{16}$ : AppleTalk,  $27_{16}$ : DECnet, ...)



# PPP Data Frame

- **info:** dữ liệu từ tầng trên cần được vận chuyển
- **check:** CRC dùng để phát hiện lỗi.

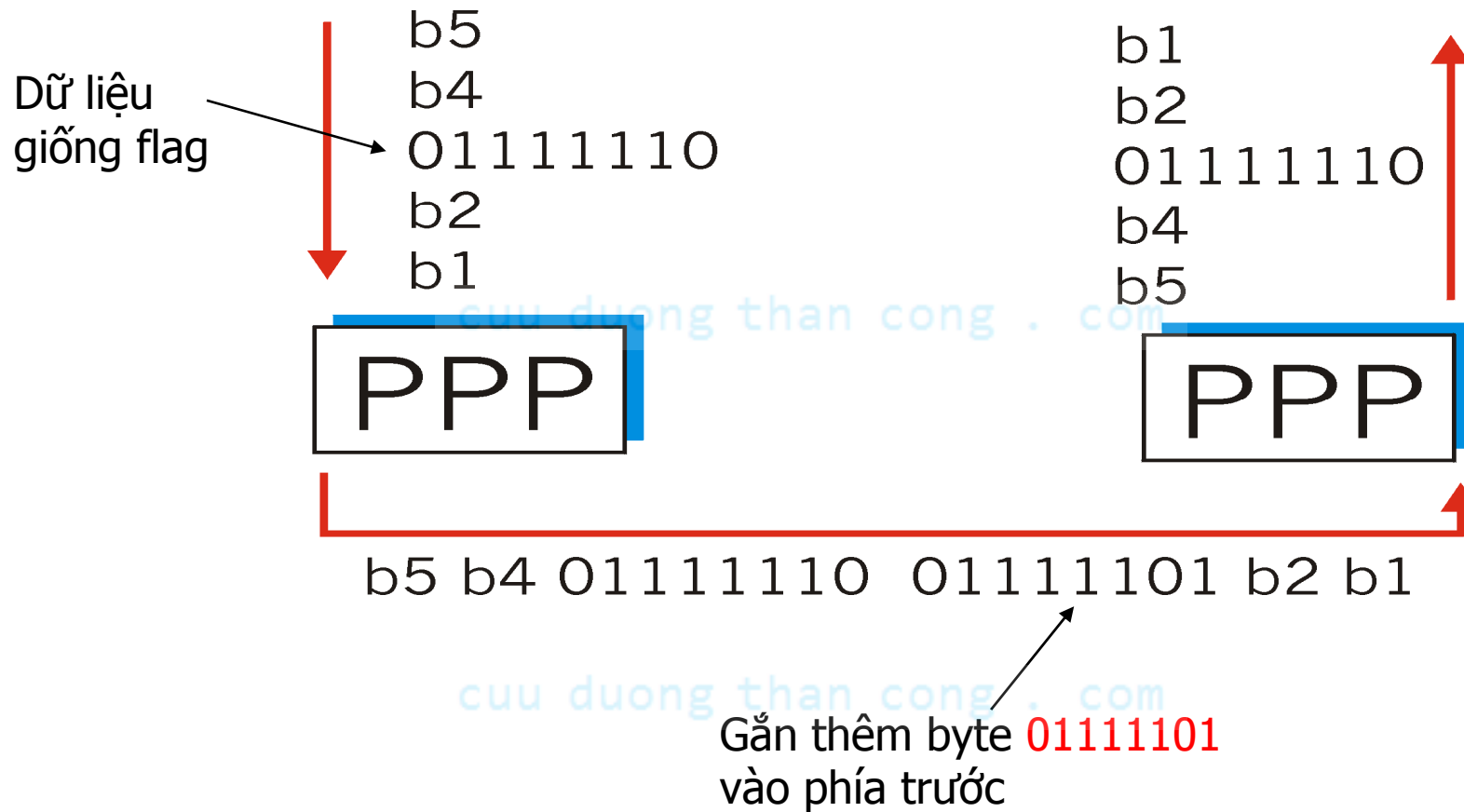


# Byte Stuffing

- Yêu cầu “data transparency” : vùng dữ liệu phải cho phép chứa dữ liệu giống như flag, tức **<01111110>**
  - **Q:** Khi nhận được **<01111110>**, đó là dữ liệu hay flag?
- **Bên gửi:** thêm vào 1 byte đặc biệt **<01111101>** trước mỗi byte dữ liệu dạng **<01111110>**, hoặc **<01111101>**, sau đó mới gửi đi.
- **Bên nhận:**
  - Nếu nhận được **<01111101>**, sau đó **<01111110>**, thì vứt bỏ **<01111101>**
  - Nếu nhận được 2 byte **<01111101>**, thì vứt bỏ 1 byte



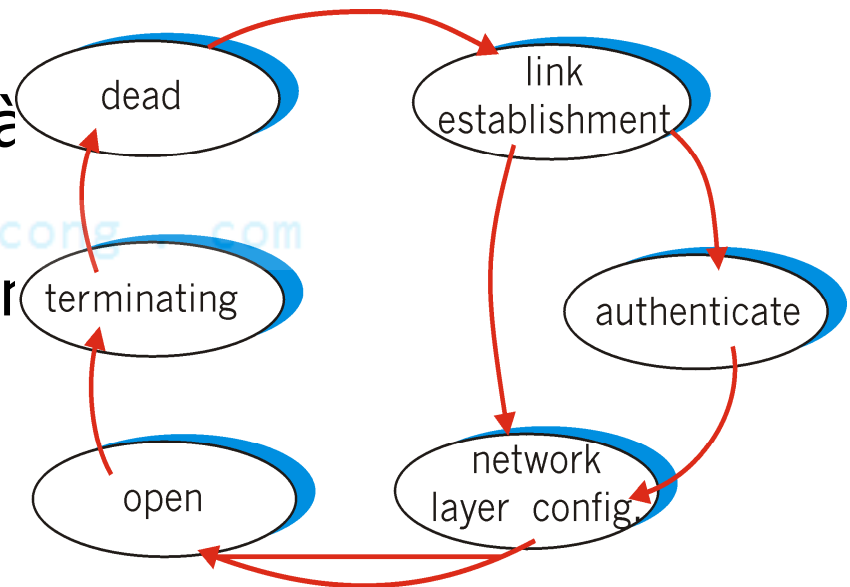
# Byte Stuffing



# PPP Data Control Protocol

Trước khi trao đổi dữ liệu tầng mạng, 2 bên gửi/nhận ở tầng link phải:

- **Cấu hình liên-kết PPP** (chiều dài tối đa frame, authentication)
- **Học/thiết lập cấu hình** thông tin tầng mạng
  - Với IP: sử dụng thông điệp giao thức của IPCP (IP Control Protocol) để cấu hình /học địa chỉ IP



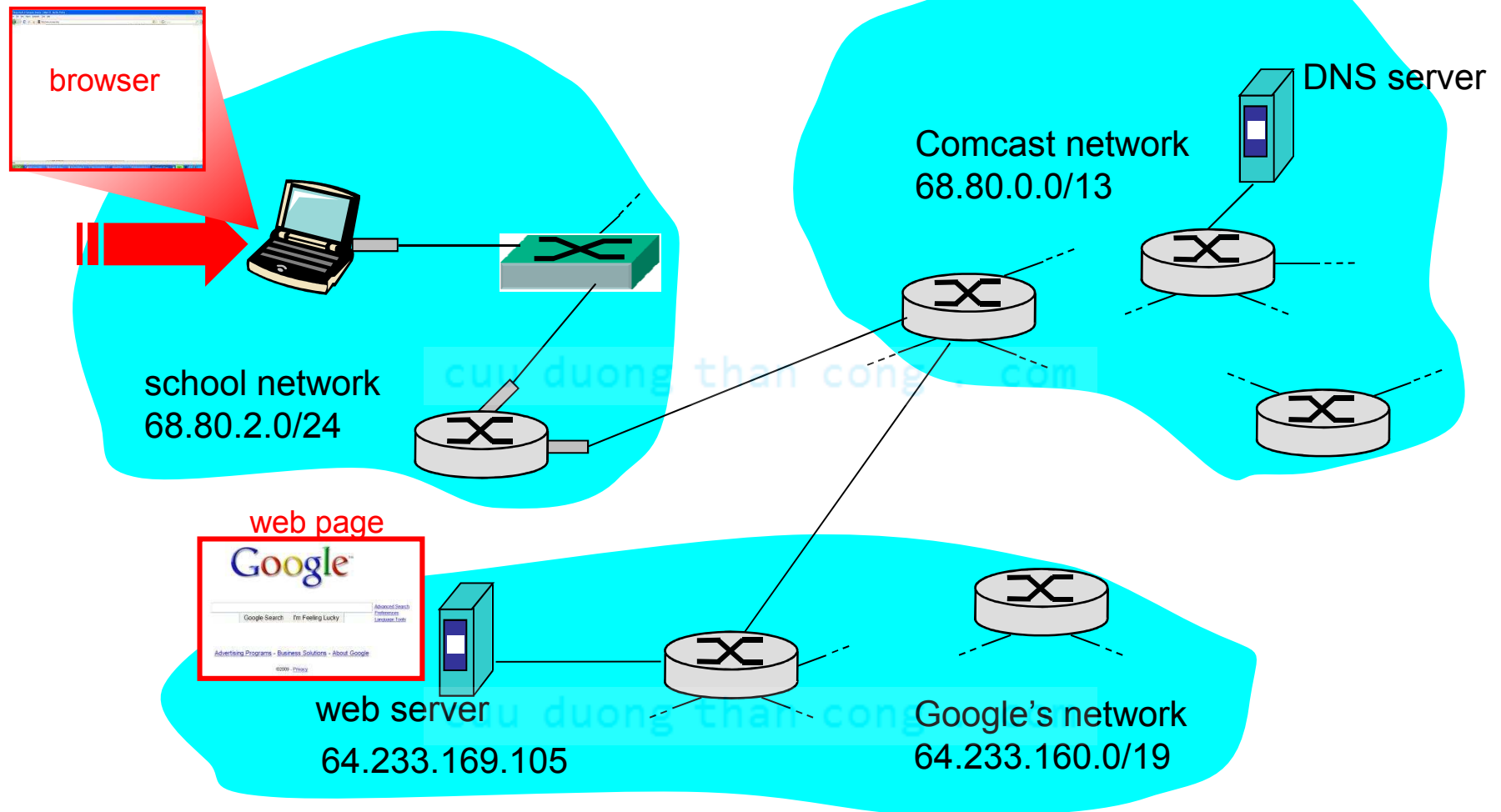
# Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Link-layer switches
- 5.7 PPP
- 5.8 Link virtualization: MPLS
- 5.9 A day in the life of a web request

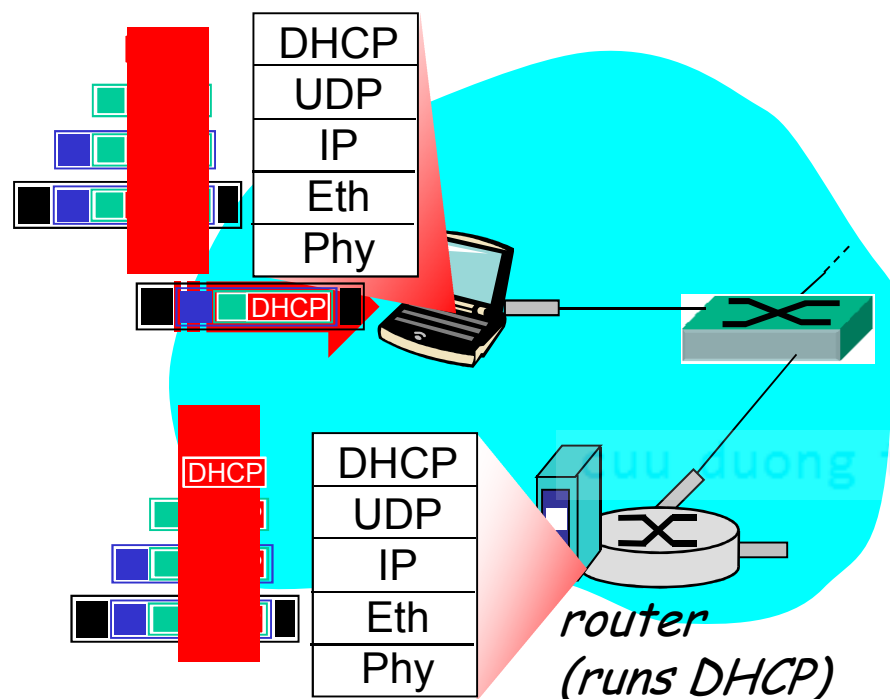
## Synthesis: a day in the life of a web request

- Đã khảo sát đủ các tầng trong chồng giao thức Internet
  - application, transport, network, link
- Kết hợp các tầng
  - *Mục tiêu:* nhận biết, ôn lại, hiểu các giao thức (ở tất cả các tầng) qua 1 kịch bản đơn giản: mở 1 trang web
  - *Kịch bản:* sinh viên găng máy laptop vào mạng trường, mở trang web [www.google.com](http://www.google.com)

# A day in the life: scenario

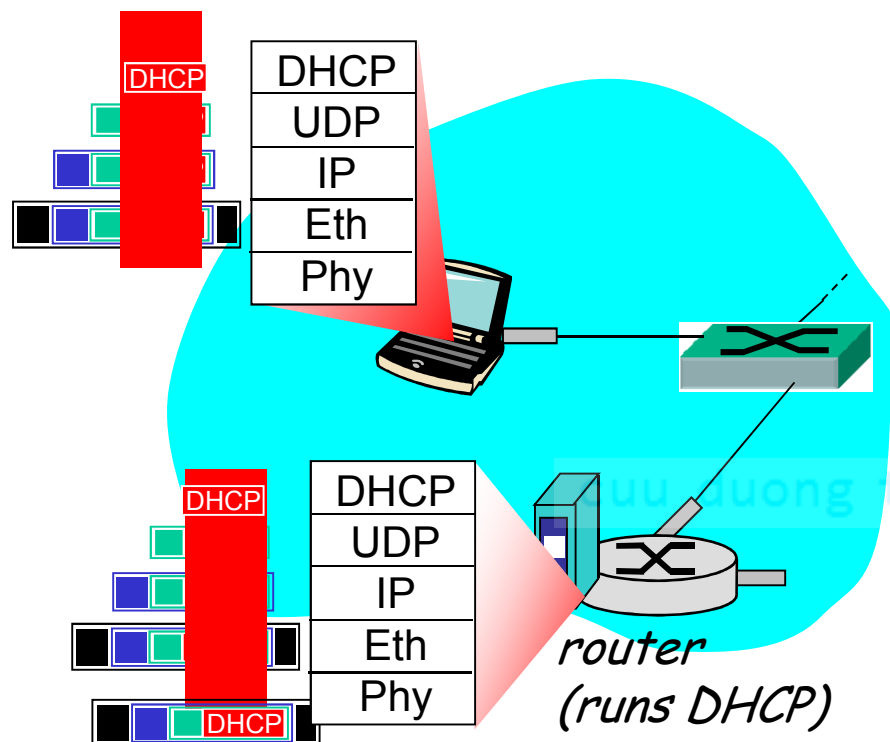


# A day in the life... connecting to the Internet



- ❑ Máy tính kết nối vào mạng, cần có đ/chỉ IP của chính nó, đ/chỉ IP của router đầu chặng, DNS server: sử dụng **DHCP**.
- ❑ DHCP request được đóng gói trong **UDP datagram**, được đóng gói tiếp trong 802.1 **Ethernet frame**.
- ❑ Ethernet frame được **gửi broadcast** (đích FFFFFFFFFFFFFFFF) trên LAN. Frame này được router có **DHCP** Server nhận.
- ❑ Tại đây, frame được tách lấy phần IP datagram, UDP segment, DHCP request

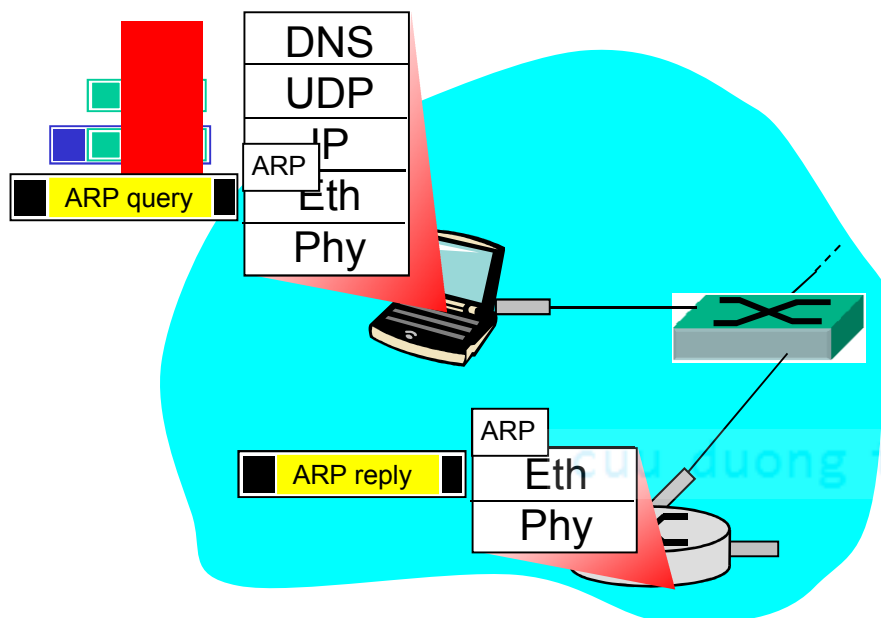
# A day in the life... connecting to the Internet



- DHCP Server hình thành **DHCP ACK**, trong đó chứa Đ/chỉ IP của client, của router đầu chặng, tên & Đ/chỉ IP của DNS Server.
- DHCP ACK được đóng gói cho đến frame, frame được chuyển trên LAN đến client. Quá trình tách và chuyển lên các tầng trên các PDU diễn ra ở client.
- DHCP client nhận được DHCP ACK reply.

*Client được cấp địa chỉ IP, biết tên và địa chỉ DNS server  
Địa chỉ IP của router đầu chặng*

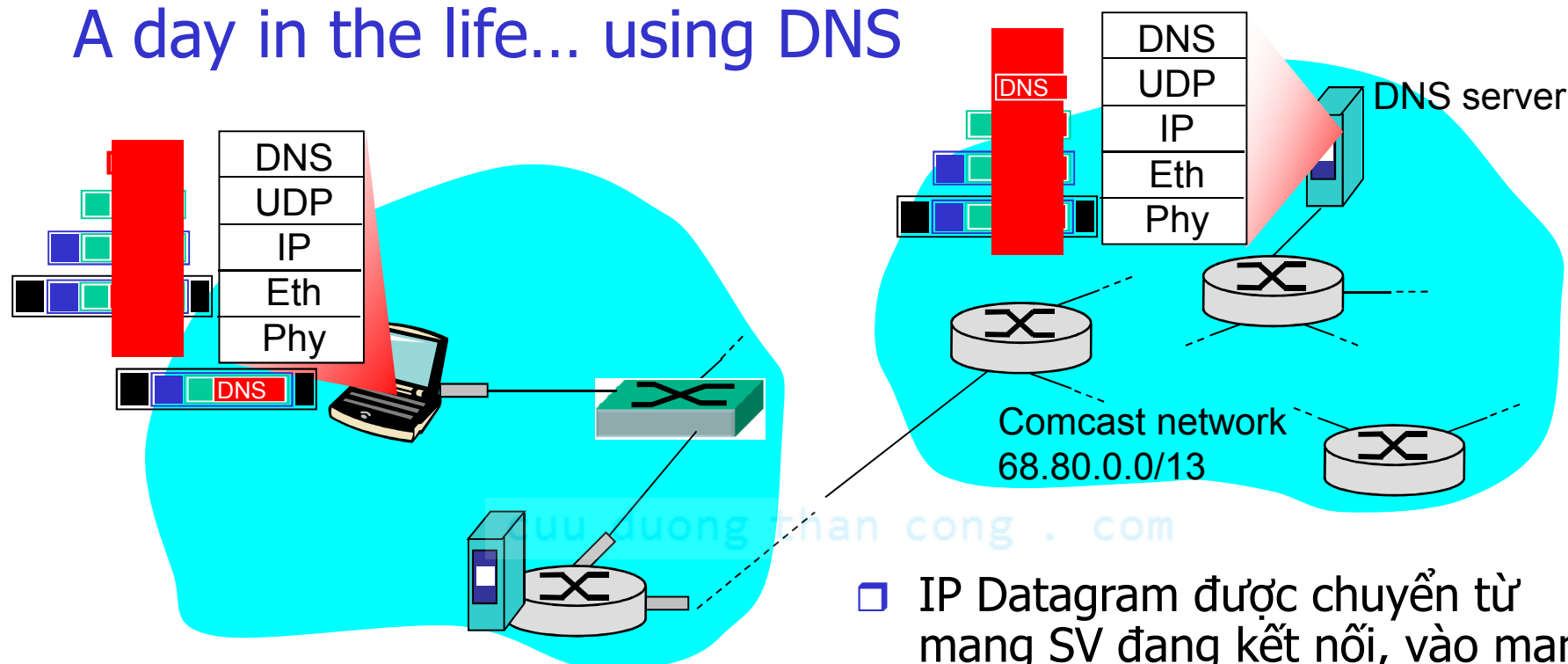
# A day in the life... ARP (before DNS, before HTTP)



- Trước khi gửi **HTTP** request, laptop cần biết địa chỉ IP của `www.google.com`: sử dụng **DNS**
- Một DNS query được tạo ra, đóng gói trong IP, đóng gói tiếp trong Ethernet frame. Để gửi frame đến được router, cần biết địa chỉ MAC của router (ở phía máy laptop). Sử dụng **ARP**.
- **ARP query** được gửi broadcast trên LAN và được router nhận. Router sẽ trả lời bằng **ARP reply**, trong đó chứa đựng địa chỉ MAC của router.
- Client biết địa chỉ MAC của router đầu chặng, do đó nó có thể gửi frame chứa DNS query.



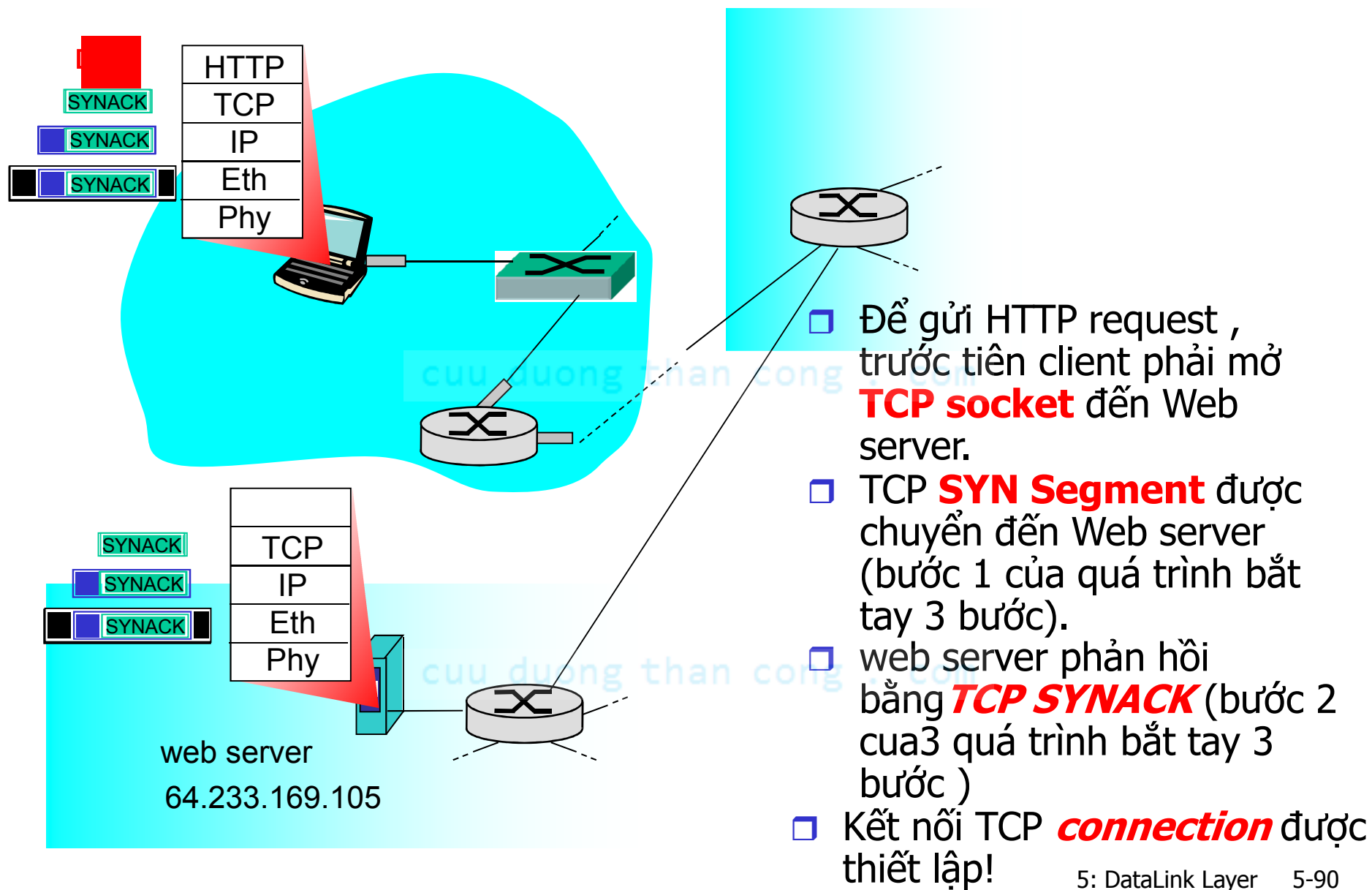
# A day in the life... using DNS



- IP datagram chứa đựng DNS query được chuyển từ client qua switch để đến router đầu chặng.

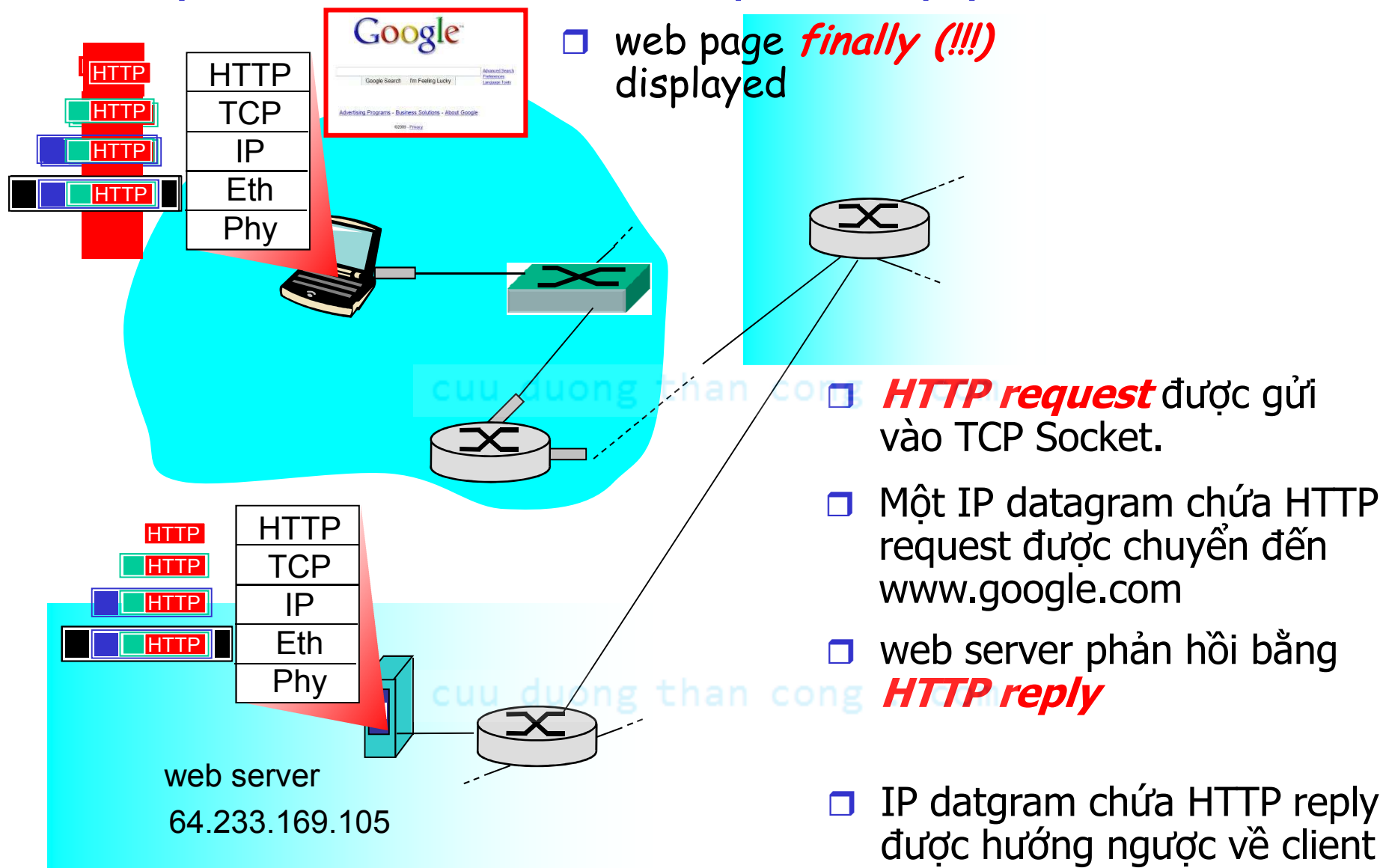
- IP Datagram được chuyển từ mạng SV đang kết nối, vào mạng Comcast, được định tuyến tới DNS Server thông qua các giao thức định tuyến **RIP, OSPF, IS-IS** và/hoặc **BGP**
- IP Datagram được trích/chuyển đến DNS server
- DNS server trả lời địa chỉ IP của **www.google.com** cho client

# A day in the life... TCP connection carrying HTTP



5: DataLink Layer 5-90

# A day in the life... HTTP request/reply



# Chapter 5: Summary

- Các nguyên tắc bên dưới của các dịch vụ tầng link
  - Phát hiện lỗi, sửa lỗi
  - Dùng chung kênh truyền quảng bá: đa truy cập
  - Đánh địa chỉ tầng liên-kết
- Sự thuyết minh và cài đặt nhiều công nghệ khác nhau ở tầng link
  - Ethernet
  - LANS dùng Switch, VLANs
  - PPP
  - Mạng được ảo hóa như tầng link: MPLS
- Tổng hợp: Một ví dụ xuyên suốt về truy cập Web

# Chapter 5: let's take a breath

- Hoàn tất một lượt khảo sát chồng giao thức Internet (ngoại trừ tầng Vật lý)
- Hiểu biết chắc chắn về các nguyên lý mạng, cũng như thực tế
- ..... Có thể dừng ở đây.... Tuy nhiên, vẫn còn **khá nhiều** chủ đề lý thú!
  - wireless
  - multimedia
  - security
  - network management

HẾT