

BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING GIỮA KỲ HỌC KỲ I NĂM HỌC 2023 – 2024



Sharing is learning



 **BAN HỌC TẬP**

Khoa Công nghệ Phần mềm

Trường Đại học Công nghệ Thông tin

Đại học Quốc gia thành phố Hồ Chí Minh

 **CONTACT**

bht.cnpm.uit@gmail.com

fb.com/bhtcnpm

fb.com/groups/bht.cnpm.uit

TRAINING

Nhập Môn Mạng Máy Tính

- ⌚ **Thời gian:** 10:00 thứ 5 ngày 26/10/2023
- 📍 **Địa điểm:** Phòng B708 – Tòa nhà B
- 👤 **Trainers:** Huỳnh Lê Đan Linh – KTPM2022.2
Trương Đoàn Vũ – MMTT2022.3



Sharing is learning

Chương I : Giới Thiệu

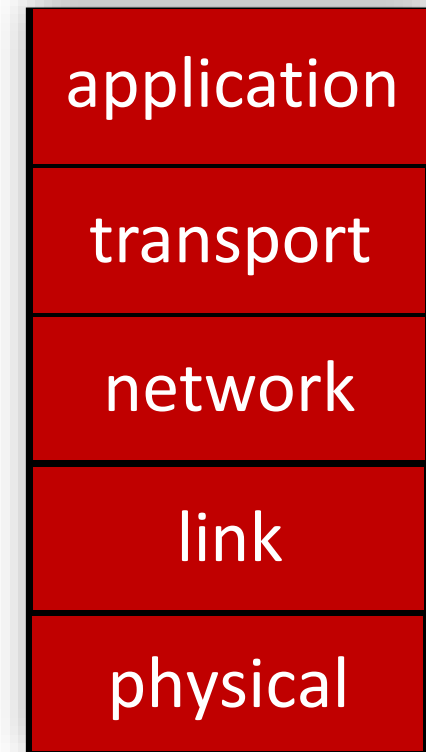
1. Câu hỏi Lý Thuyết Chương I
2. Dạng toán về Độ trễ



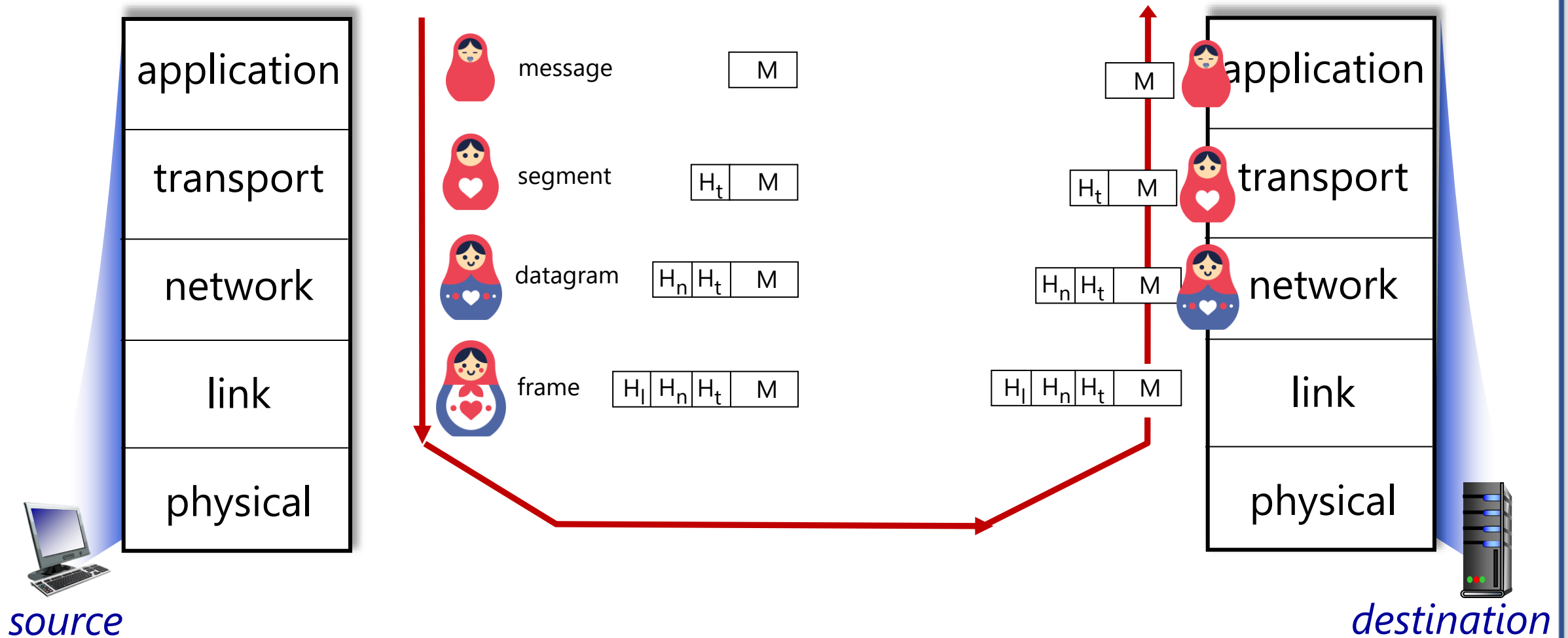
Chồng giao thức Internet phân tầng (5 Tầng)

- *application*: (data)
- *transport*: TCP, UDP (segment)
- *network*: datagram-(packet)
- *link*: (frame)
- *physical*: (bits)

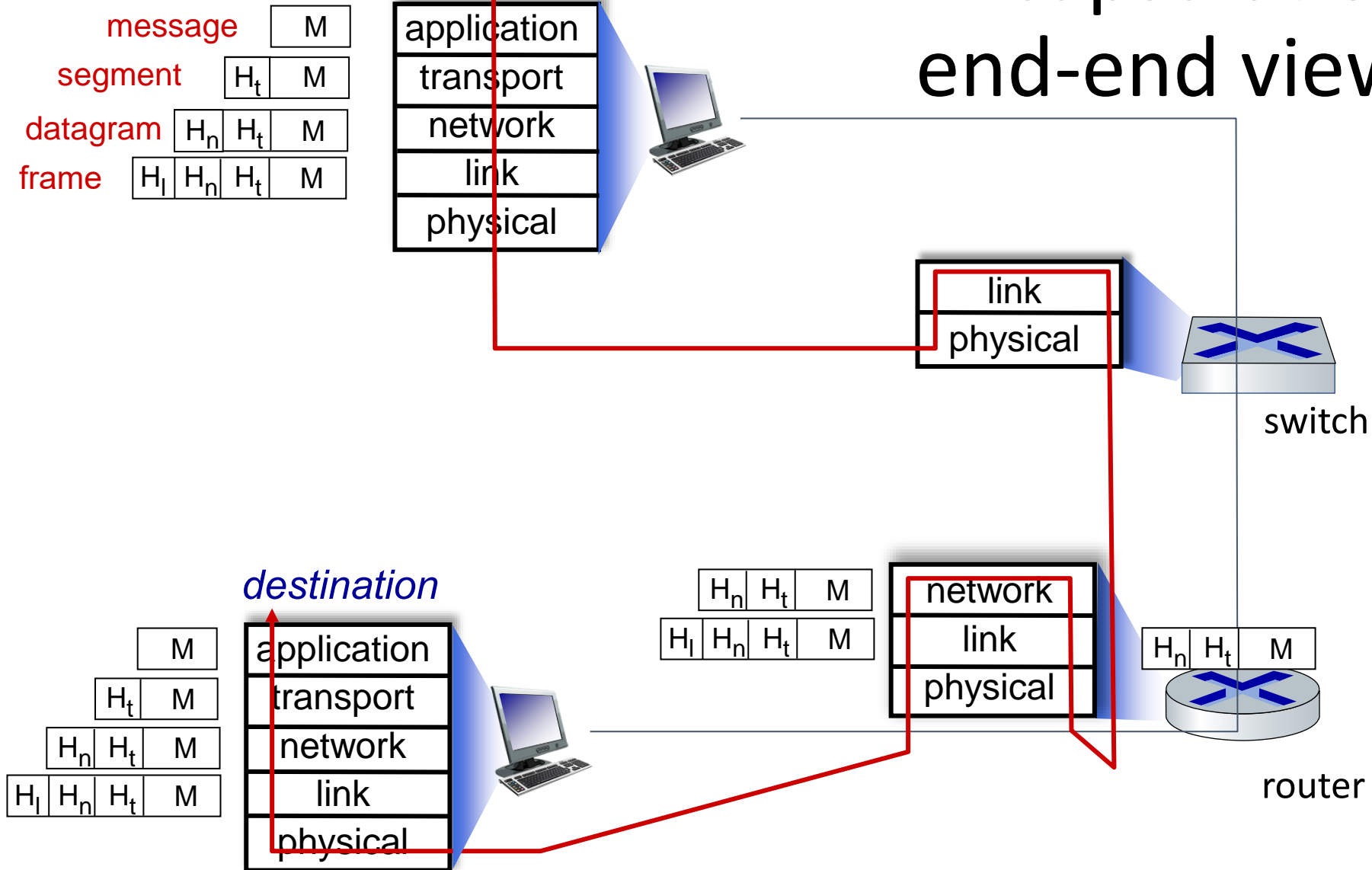
Cần nắm thứ tự và các đơn vị dữ liệu



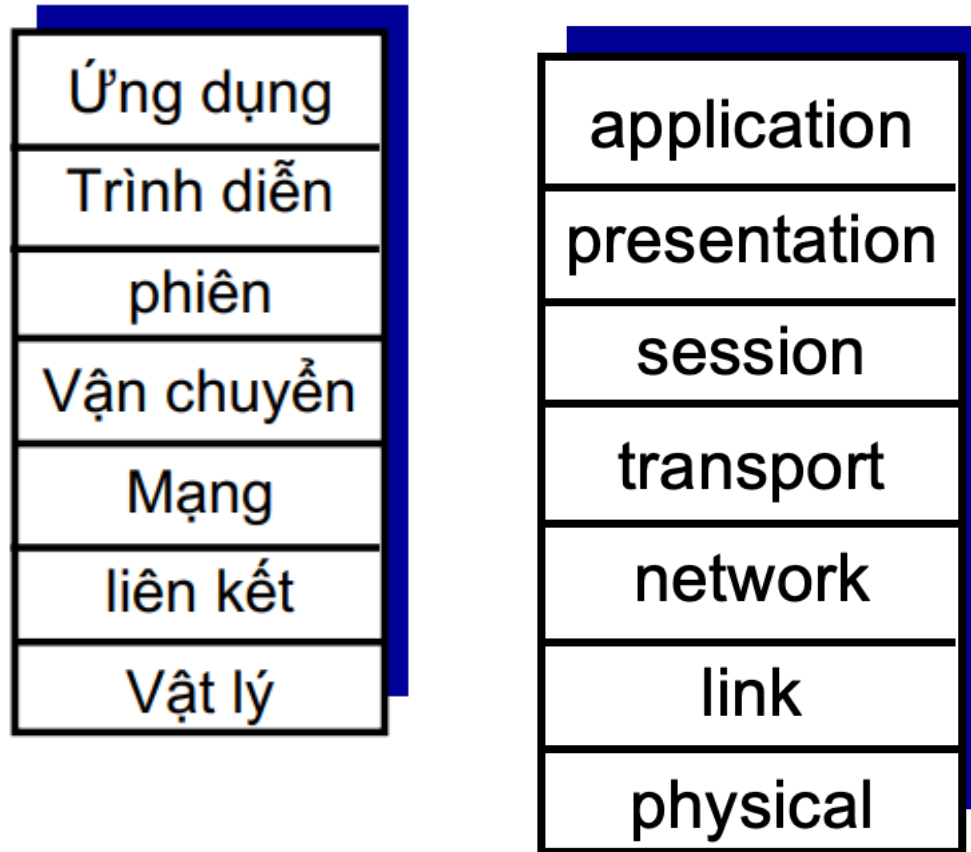
Định vụ ,Phân Tầng và Quá trình đóng gói



Encapsulation: an end-end view



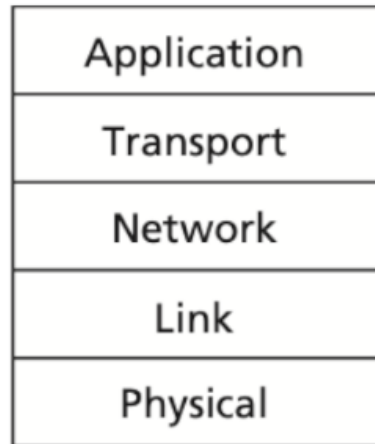
Mô Hình OSI



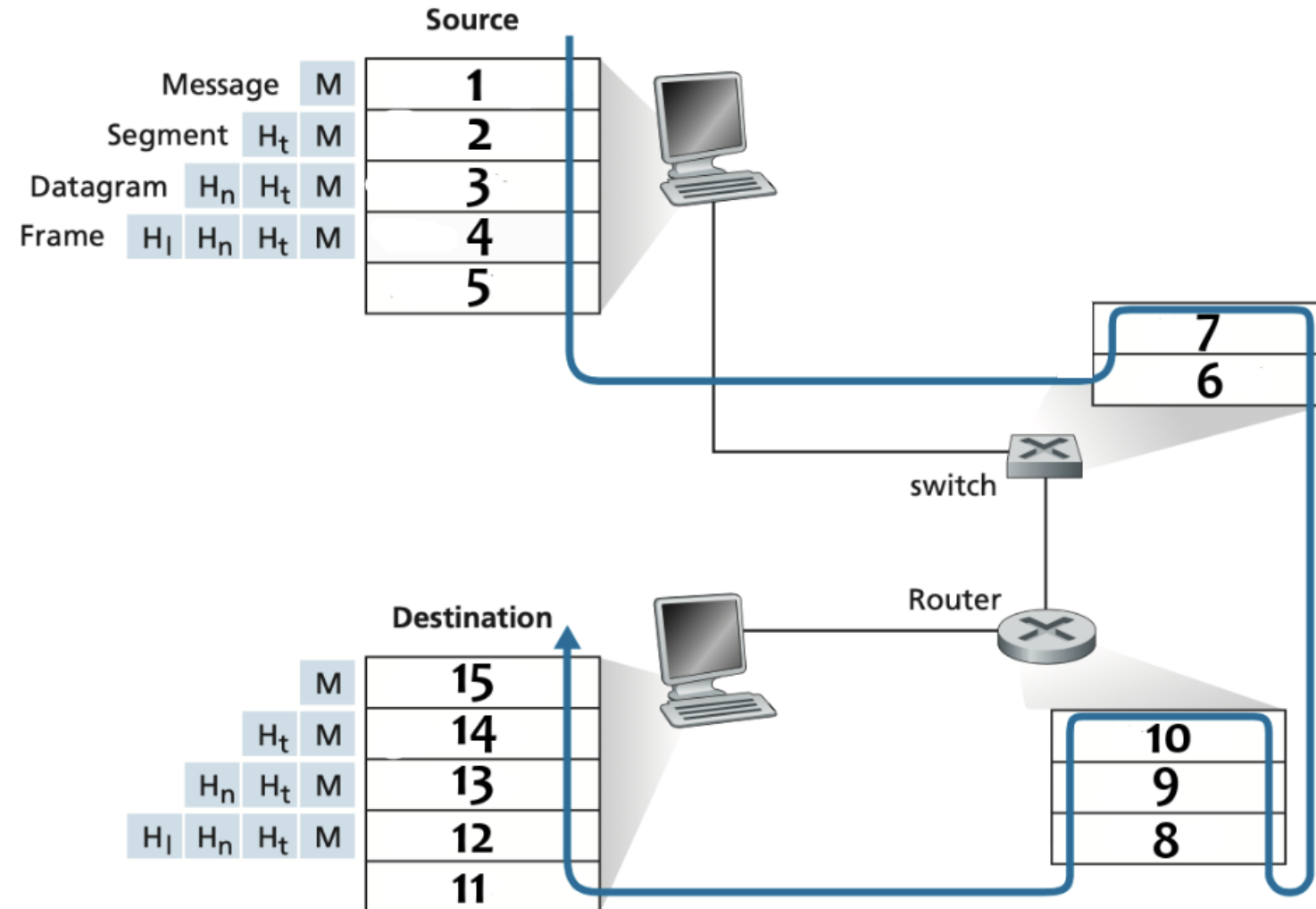
Cần nắm thứ tự các tầng từ trên xuống dưới và ngược lại

Câu Hỏi Lý Thuyết Chương I

GK 2020- Cho mô hình chồng giao thức 5 tầng



Five-layer
Internet
protocol stack



GK 2020

Câu 1: Cho Biết Tầng nào ở Vị Trí 3.

A.Application

B.Tầng Transport

C.Tầng Network

D.Tầng Phisical

Chọn C

GK 2020

Câu 2: Cho biết Tầng và Đơn vị Dữ Liệu tương ứng ở vị trí 12

- A. Tầng Applicaton và Message
- B. Tầng Network và Segment
- C. Tầng Link và Frame
- D. Tầng Physical và Frame

Chọn C .

GK 2019

Câu 1: Tầng nào dưới đây không thuộc mô hình TCP/IP

A. Tầng Applicaton

B. Tầng Session

C. Tầng Link

D. Tầng Network

Chọn B .

GK 2019

Câu 2: Đơn vị dữ liệu theo các tầng trong mô hình TCP/IP

- A. Data ,Packet ,Segment , Bit, Frame
- B. Data ,Packet ,Segment , Frame , Bit
- C. Data ,Segment,Packet,Frame,Bit
- D. Data ,Segment,Packet Bit,Frame

Chọn C.

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 1 :Thứ tự đúng từ trên xuống dưới trong mô hình OSI là:

- A. Application – Transport – Session – Presentation – Network – Physical – Data Link
- B. Application – Presentation – Session – Transport – Network – Data Link - Physical
- C. Application – Session – Presentation – Transport – Network – Data Link – Physical
- D. Application – Presentation – Session – Transport – Internet – Data Link - Physical

Chọn B

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 2: TCP và UDP hoạt động ở tầng nào các chồng giao thức?

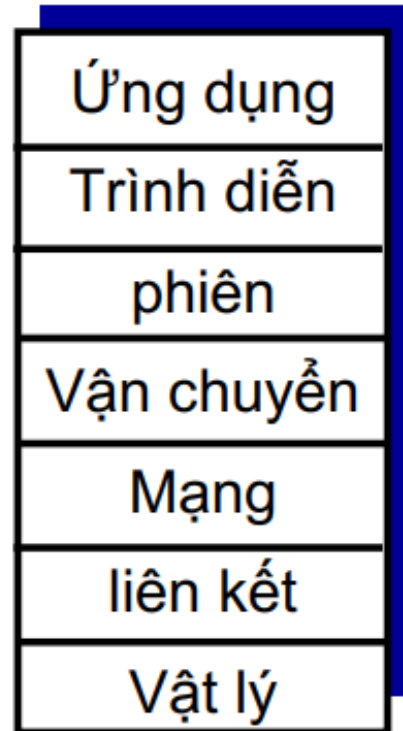
- A. Tầng vận chuyển
- B. Tầng mạng
- C. Tầng liên kết
- D. Tầng Ứng Dụng

Chọn A

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 3: Mô hình OSI có bao nhiêu lớp

- A. 5
- B. 6
- C. 7
- D. 8



Chọn C

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 4: Thứ tự đúng từ trên xuống dưới trong mô hình chồng giao thức Internet phân tầng

- A. Application – Transport – Network – Physical – Link
- B. Physical – Link – Network – Transport - Application
- C. Application – Network – Transport – Link - Physical
- D. Application – Transport – Network – Link – Physical

Chọn D

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 5: Quá trình một thông điệp (message) được đưa qua các tầng (layer) và được thêm các thông tin điều khiển vào đầu (header) trong hoạt động của thiết bị mạng được gọi là gì ?

- A.Compress
- B.Encapsulation
- C.De-Encapsulation
- D.Header-Adding

Chọn B

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 6: Khi gói dữ liệu di chuyển từ lớp cao xuống lớp thấp hơn thì các header được

- A. Loại bỏ dần
- B. Thêm vào dần
- C. Sắp xếp lại
- D. Đổi vị trí

Chọn B

Dạng bài câu hỏi về các mô hình dịch vụ

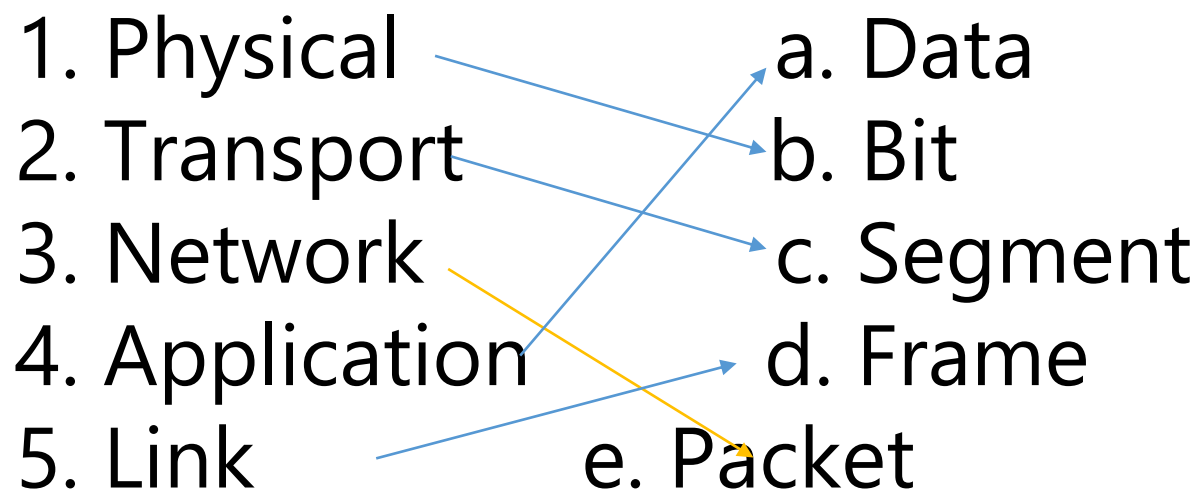
Câu 7: Khi gói dữ liệu di chuyển từ lớp thấp lên lớp cao hơn thì các header được

- A. Loại bỏ dần
- B. Thêm vào dần
- C. Sắp xếp lại
- D. Đổi vị trí

Chọn B

Câu 8

Ghép giữa đơn vị dữ liệu tương ứng với các tầng trong các Tầng giao thức internet



A. 1b, 2c, 3d, 4a, 5e

B. 1b, 2e, 3d, 4a, 5c

C. 1b, 2d, 3c, 4a, 5e

D. 1b, 2c, 3e, 4a, 5d

Chọn D

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 9: Các đơn vị dữ liệu giao thức trong mô hình OSI được gọi là:

- A.UDP
- B.PDU
- C.OSI
- D.Packet

Chọn B

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 10: Giao thức nào thuộc tầng Application?

- A. TCP
- B. HTTP
- C. NFS
- D. IP

Chọn B

Dạng bài câu hỏi về các mô hình dịch vụ

Câu 11: Chọn lớp OSI nào sau đây không tồn tại?

- A. Internet
- B. Network
- C. Session
- D. Data Link

Chọn A

Đặt vấn đề: Dạng toán tính Độ Trễ

Câu 1 : Một gói tin có độ dài $L=30000$ bits được truyền trên đường liên kết giữa 2 router có tốc độ truyền $R=10\text{Mbps}$, khoảng cách giữa 2 Router $d=300\text{km}$ và tốc độ lan truyền là $s=3.10^8\text{m/s}$. Để giảm độ trễ đầu cuối đi một nửa ($d_{new} = \frac{d}{2}$) có thể điều chỉnh tốc độ truyền R_{new} như thế nào ?(Bỏ qua độ trễ xử lý và độ trễ xếp hàng)

A. $R_{new} = 3R$

B. $R_{new} = 2R$

C. $R_{new} = R$

D. $R_{new} = R + R/2$



Tóm Tắt Đề : Đề thi

Câu 1 :

Độ dài gói tin = 30000 bits

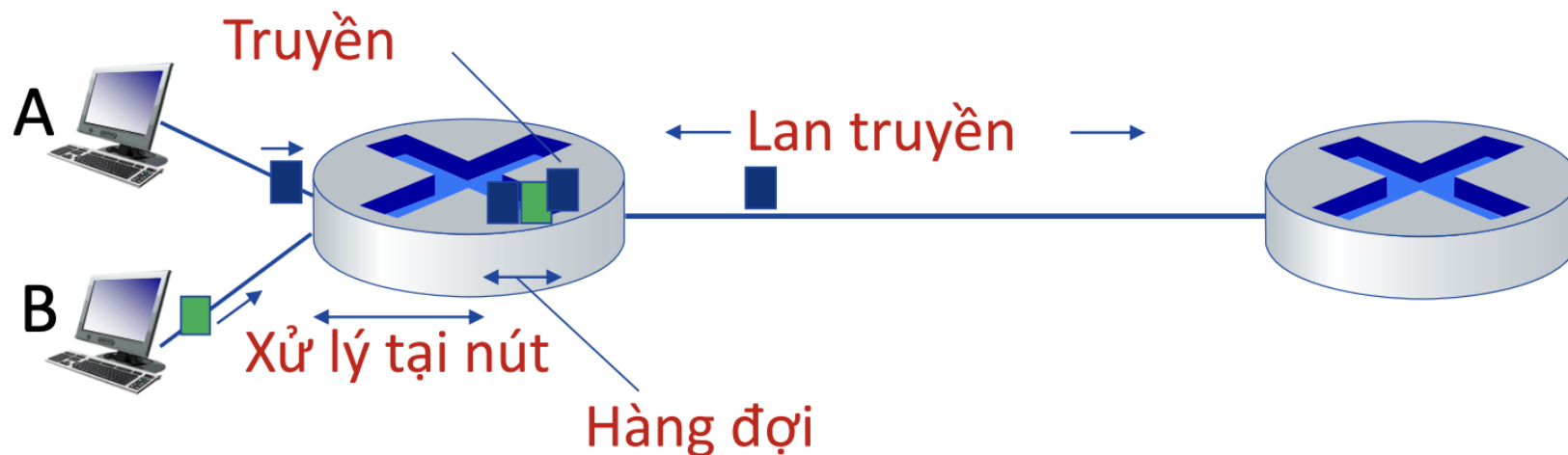
Tốc độ truyền: $10\text{Mbps} = 10 \cdot 10^6 \text{ bps}$.

Tốc độ lan truyền: $3 \cdot 10^8 \text{ m/s}$

Độ dài đường truyền : $300\text{km} = 300000 \text{ m}$

Để giảm độ trễ đầu cuối đi một nửa ($d_{new} = \frac{d}{2}$) , hỏi tốc độ truyền mới $R_{new} = ?$





Tổng Độ Trễ = Độ Trễ Truyền + Độ Trễ Lan Truyền

$$\text{Độ Trễ Truyền : } d_{\text{trans}} = \frac{\text{Độ dài gói tin}}{\text{Tốc độ truyền}}$$

$$\text{Độ Trễ Lan truyền : } d_{\text{prob}} = \frac{\text{Độ dài đường truyền}}{\text{Tốc độ lan truyền}}$$

$$d = d_{\text{trans}} + d_{\text{prob}} = \frac{\text{Độ dài gói tin}}{\text{Tốc độ truyền}} + \frac{\text{Độ dài đường truyền}}{\text{Tốc độ truyền}}$$

Tóm Tắt đề

Độ dài gói tin = 30000 bits

Tốc độ truyền : 10Mbps = $10 \cdot 10^6$ bps.

Tốc độ lan truyền: $3 \cdot 10^8$ m/s

Giải :

$$d = d_{\text{trans}} + d_{\text{prob}} = \frac{\text{Độ dài gói tin}}{\text{Tốc độ truyền}} + \frac{\text{Độ dài đường truyền}}{\text{Tốc độ truyền}}$$

$$= \frac{30000}{10 \cdot 10^6} + \frac{300000}{3 \cdot 10^8} = 0,004s$$

$$d_{\text{new}} = \frac{d}{2} = \frac{0,04}{2} = 0,002s$$



Tóm Tắt đề

Độ dài gói tin = 30000 bits

Tốc độ truyền: $10\text{Mbps} = 10 \cdot 10^6 \text{ bps}$.

Tốc độ lan truyền: $3 \cdot 10^8 \text{ m/s}$

Giải :

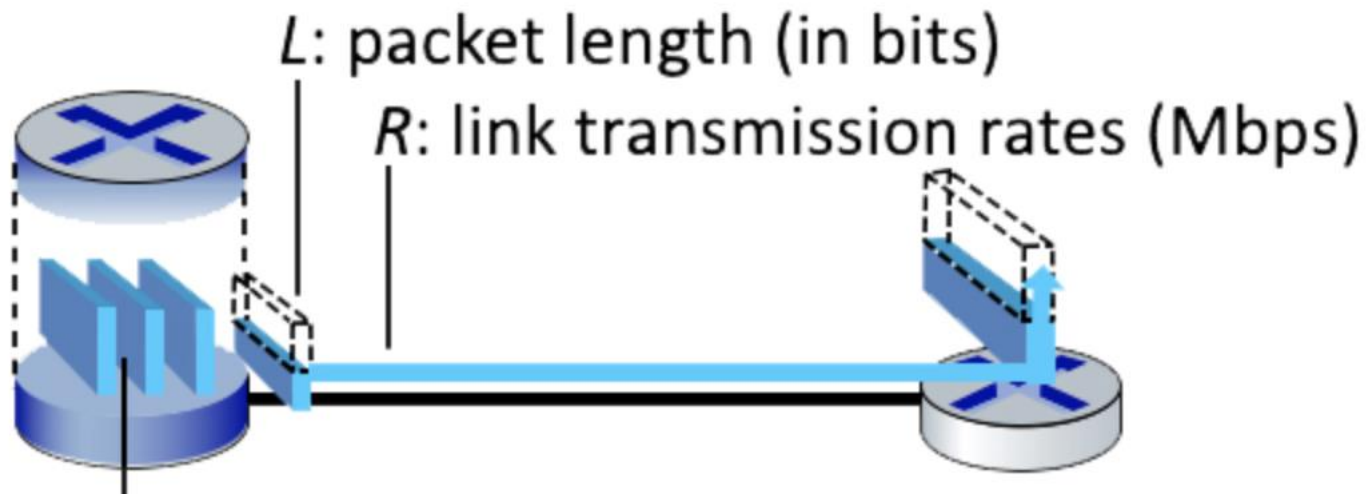
$$d_{new} = d_{\text{trans-new}} + d_{\text{prob}} = > 0,002 = \frac{30000}{R_{new}} + \frac{300000}{3 \cdot 10^8}$$

$$R_{new} = 30 \cdot 10^6 \text{ (bps)} = 3R$$

Chọn A. $R_{new} = 3R$



Câu 1:



Tốc độ truyền $R = 100 \text{ Mbps} = 100 \cdot 10^6 \text{ bps}$

Độ dài gói tin $L = 12000 \text{ bits}$

a) Tính Độ trễ truyền

b) Hỏi tối đa có bao nhiêu gói tin được truyền trong 1s?

Câu 1:

Tốc độ truyền : $R=100Mbps= 100.10^6 \text{ bps}$

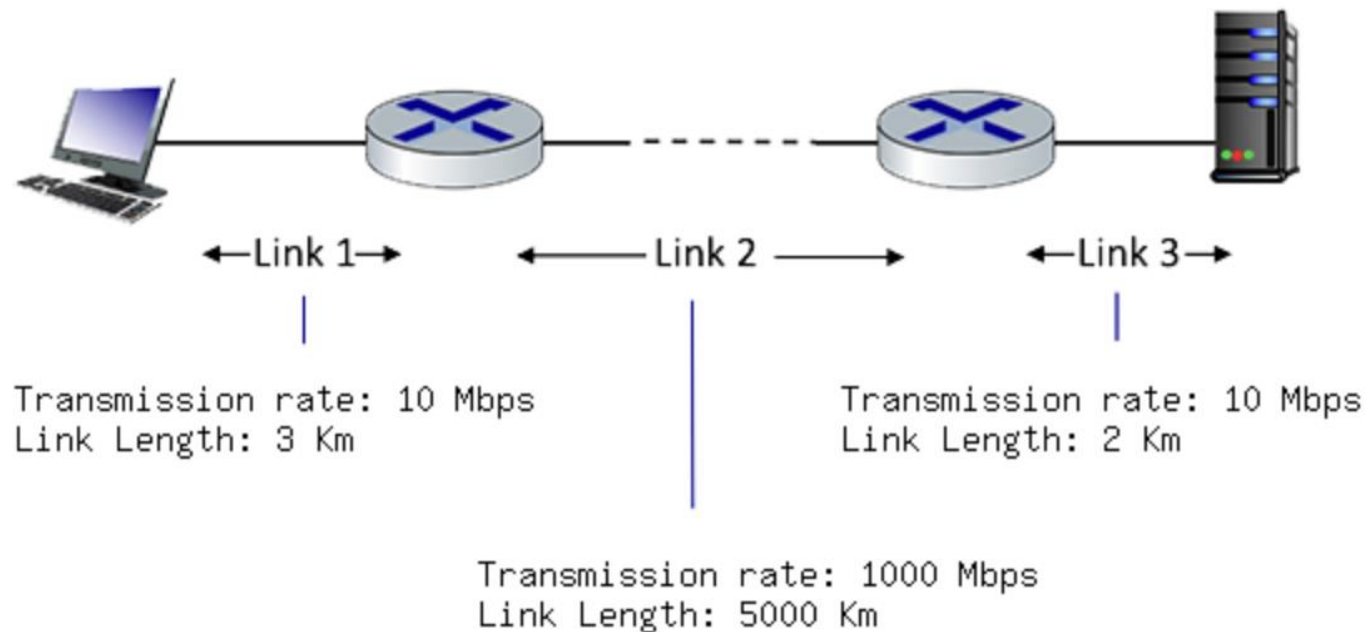
Độ Dài một gói tin : $L=12000\text{bits}$

a) Độ Trễ truyền $d_{\text{trans}} = \frac{\text{Độ dài gói tin}}{\text{Tốc độ truyền}} = \frac{12000}{100.10^6} = 1,2. 10^{-4} \text{ s}$

b) Trong 1 s truyền được 100.10^6 bits
mà 1 gói tin dài 12000bits

Số gói tin tối đa truyền trong 1s là : $\frac{100.10^6}{12000} = 8333$ gói tin

Câu 2:



Dài Gói Tin: 8000 bits

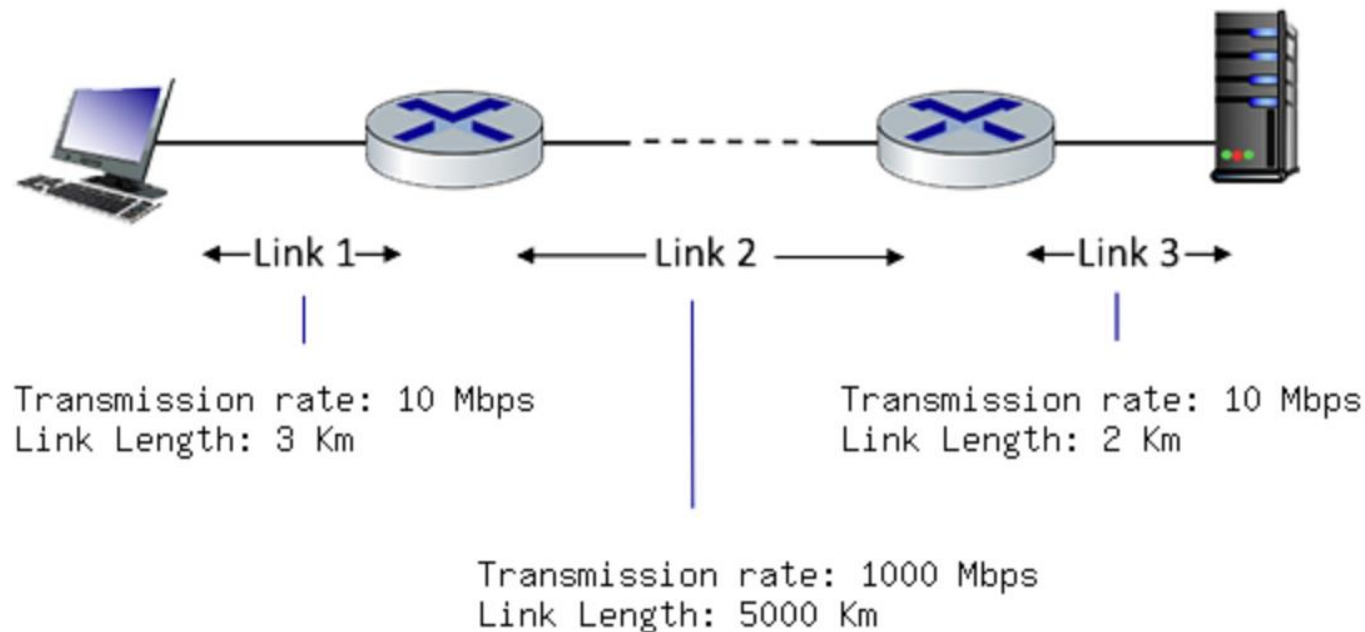
Tốc độ lan truyền : 3×10^8 m/sec

a) Tính Tổng độ trễ Link 1

b) Tính Tổng độ trễ Link 2

c) Tính tổng độ trễ cả 3 link

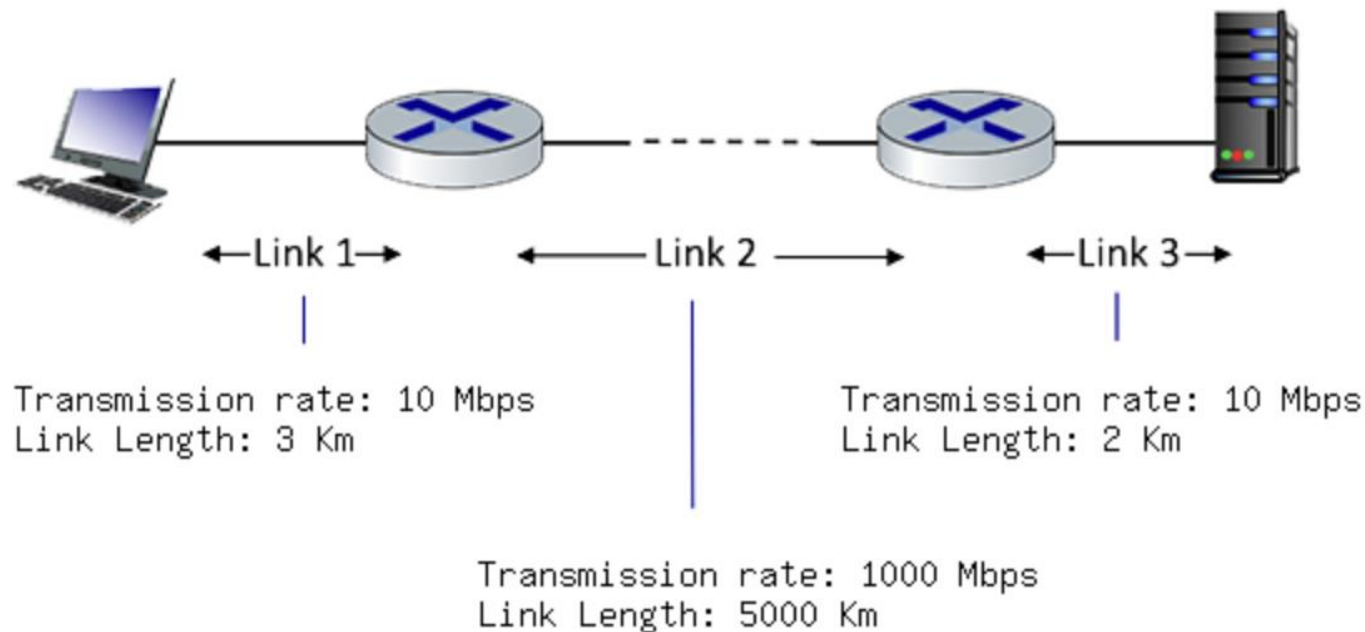
Câu 2:



a) Tính Tổng độ trễ Link 1

$$d_1 = d_{\text{trans}} + d_{\text{prob}} = \frac{\text{Độ dài gói tin}}{\text{Tốc độ truyền}} + \frac{\text{Độ dài đường truyền}}{\text{Tốc độ lan truyền}}$$
$$= \frac{8000}{10 \cdot 10^6} + \frac{3000}{3 \cdot 10^8} = 8,1 \cdot 10^{-4} s$$

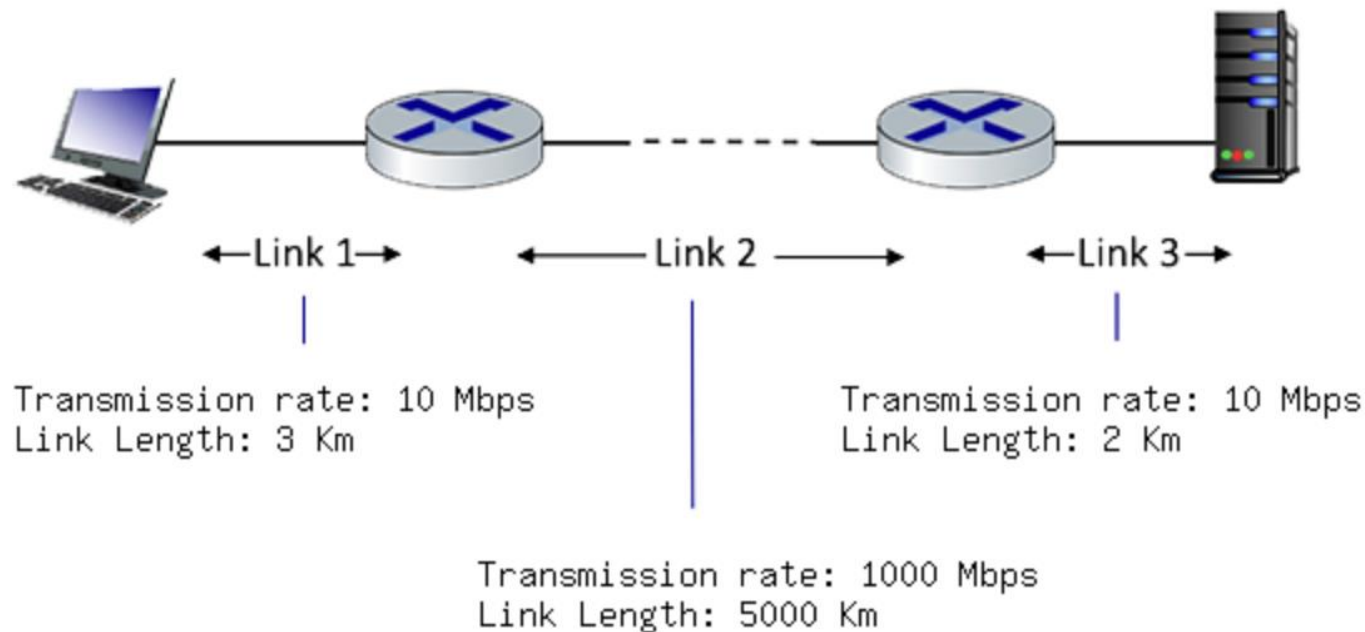
Câu 2:



b) Tính Tổng độ trễ Link 2

$$d_2 = d_{\text{trans}} + d_{\text{prob}} = \frac{\text{Độ dài gói tin}}{\text{Tốc độ truyền}} + \frac{\text{Độ dài đường truyền}}{\text{Tốc độ truyền}}$$
$$= \frac{8000}{1000 \cdot 10^6} + \frac{5000000}{3 \cdot 10^8} = 0,01667s$$

Câu 2:



c) Tính Tổng độ trễ cả 3 link

$$d_3 = d_{\text{trans}} + d_{\text{prob}} = \frac{8000}{10 \cdot 10^6} + \frac{2000}{3 \cdot 10^8} = 0,00081s$$

$$d = d_1 + d_2 + d_3 = 8,1 \cdot 10^{-4} + 0,01667 + 0,00081 = 0,01829s$$

Web tự luyện : https://www-net.cs.umass.edu/kurose_ross/interactive/end-end-delay.php

Câu 3: Cho

Độ dài gói tin = 100bits

Tốc độ truyền: $28\text{Kbps} = 28 \cdot 10^3\text{bps}$.

Tốc độ lan truyền: $2,5 \cdot 10^8\text{m/s}$

Tính độ dài đường truyền, biết độ trễ truyền bằng độ trễ lan truyền.

Giải :Gọi độ dài đường truyền là a (m)

Ta có $d_{\text{trans}} = d_{\text{prob}}$

$$\Leftrightarrow \frac{100}{28 \cdot 10^3} = \frac{a}{2,5 \cdot 10^8} \Rightarrow a = 892857\text{m}$$

Thi Online Ôn Tập Chương I

<https://azota.vn/de-thi/g43ytk>

2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng

- **Ứng dụng mạng:** Chạy trên các hệ thống đầu cuối khác nhau, các hệ thống đầu cuối này giao tiếp qua mạng
- Các thiết bị trong mạng lõi **không chạy** các ứng dụng của người dùng

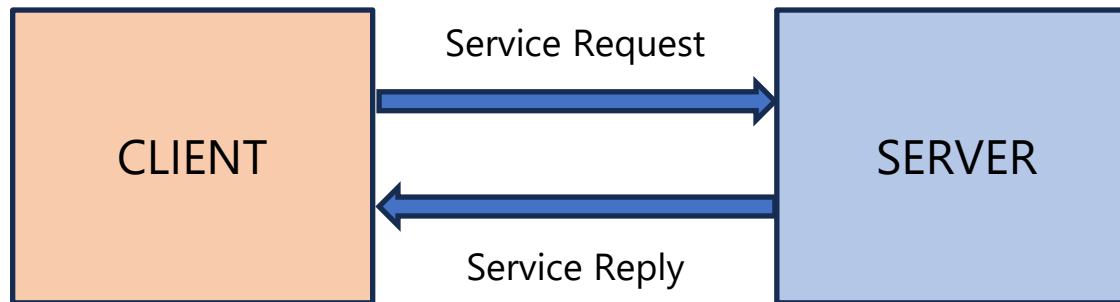


2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các kiến trúc

Mô hình Client – Server (HTTP, IMAP, FTP)

- **Server:** Luôn hoạt động, địa chỉ IP cố định, thường tổ chức thành các trung tâm dữ liệu để mở rộng quy mô
- **Client:** Liên lạc, giao tiếp với server, có thể kết nối không liên tục, có thể thay đổi địa chỉ IP, không giao tiếp trực tiếp với các máy khách khác



2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các kiến trúc

Kiến trúc Peer-to-Peer (P2P)

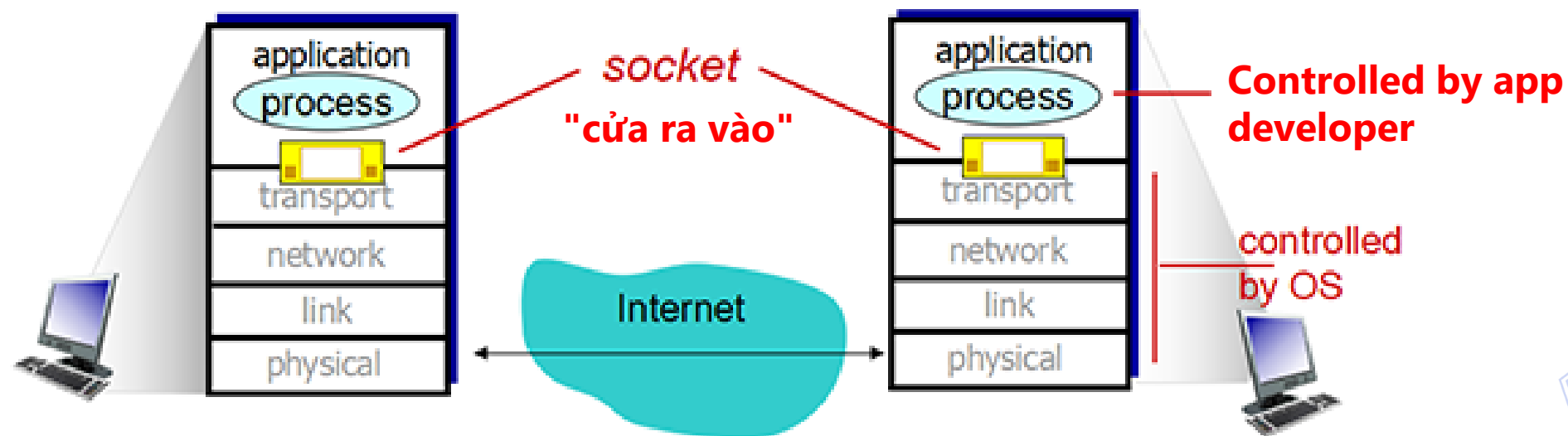
- **Không có máy chủ** luôn hoạt động, **các hệ thống đầu cuối giao tiếp** trực tiếp với nhau
- **Khả năng tự mở rộng** – các peer cung cấp dịch vụ mới, có nhu cầu mới
- Các peer **kết nối không liên tục** và **đổi địa chỉ IP** => **Quản lý phức tạp**



2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Socket

- **Kết nối** dạng logic giữa tiến trình ở máy này với tiến trình ở máy khác
- **"Cửa ra vào"**: Thông điệp đẩy ra qua socket và nhận về từ socket



2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Định danh tiến trình

- Định danh bằng **địa chỉ IP** (32-bit) và **số hiệu cổng (port - mô tả trong RFC)**
- **Port của một số giao thức**

- FTP data: 20
- FTP control: 21
- Telnet: 23
- HTTPS: 443
- POP3: 110
- SMTP: 25

TCP

- RIP: 520
- TFTP: 69
- SNMP 161
- DNS: 53
- HTTP: 80
- SSH: 22

UDP

Cả hai



Sharing is learning

2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các yêu cầu của giao thức vận chuyển

Khác nhau trên từng ứng dụng

- Toàn vẹn dữ liệu (Data integrity)
- Định thì (Timing)
- Thông lượng (Throughput)
- An ninh (Security)



2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Giao thức tầng ứng dụng

Định nghĩa những vấn đề:

- Các **loại thông điệp** được trao đổi
- **Cú pháp** thông điệp
- **Ngữ nghĩa** của thông điệp
- Các **quy tắc**: Khi nào và cách thức các tiến trình gửi và phản hồi các thông điệp
- Các **giao thức mở**: Định nghĩa trong các RFC, quyền truy cập mở
- Các **giao thức độc quyền**: Skype, Zoom...

* **RFC (Request for Comments)**: Bảng mô tả các giao thức - **quy chuẩn chung**

2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các yêu cầu của giao thức vận chuyển

Ứng dụng	Mất dữ liệu	Thông lượng	Độ nhạy thời gian
Truyền/tải tập tin	Không	Mềm dẻo	Không
Thư điện tử	Không	Mềm dẻo	Không
Tài nguyên web	Không	Mềm dẻo	Không
Audio/video thời gian thực	Chịu lỗi	audio: 5Kbps-1Mbps video: 10Kbps-5Mbps	Có, 10's msec
streaming audio/video	Chịu lỗi	Như trên	Có, few secs
Game tương tác	Chịu lỗi	Kbps+	Có, 10's msec
Nhắn tin	Không	Mềm dẻo	Có và không

2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các dịch vụ giao thức vận chuyển

TCP:

- Truyền tải **tin cậy** (reliable transport)
- **Điều khiển luồng** (flow control): bên gửi không gửi quá khả năng bên nhận
- **Điều khiển tắc nghẽn** (congestion control): điều tiết bên gửi khi mạng quá tải
- **Hướng kết nối** (connectionoriented): thiết lập kết nối giữa bên gửi và bên nhận
- **Không cung cấp**: định thì, thông lượng tối thiểu, an ninh

2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các dịch vụ giao thức vận chuyển

UDP:

- Truyền tải **không tin cậy**
- Đảm bảo **tốc độ**
- **Không cung cấp:** tính tin cậy, điều khiển luồng, điều khiển tắc nghẽn, định thì, thông lượng đảm bảo, an ninh hay thiết lập kết nối



2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các dịch vụ giao thức vận chuyển

Ứng dụng	Giao thức Tầng Ứng dụng	Giao thức Vận chuyển
Truyền/tải tập tin	FTP [RFC 959]	TCP
Thư điện tử	SMTP [RFC 5321]	TCP
Tài nguyên Web	HTTP [RFC 7230, 9110]	TCP
Điện thoại Internet	SIP [RFC 3261], RTP [RFC 3550], hoặc các giao thức độc quyền	TCP or UDP
Streaming audio/video	HTTP [RFC 7230], DASH	TCP
Game tương tác	WOW, FPS (proprietary)	UDP or TCP

2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các dịch vụ giao thức vận chuyển

Bảo mật TCP

- TCP và UDP Socket: **Không mã hóa**
- Transport Layer Security (**TLS**): Cung cấp các **kết nối TCP có mã hóa**

=> Toàn vẹn dữ liệu

=> Chứng thực đầu cuối

- TLS được triển khai ở **Tầng Ứng dụng**

=> Các ứng dụng sử dụng các thư viện TLS => làm việc với TCP



2. APPLICATION LAYER

2.1. Nguyên lý của các ứng dụng mạng - Các dịch vụ giao thức vận chuyển

Bảo mật TCP

- TCP và UDP Socket: **Không mã hóa**
- Transport Layer Security (**TLS**): Cung cấp các **kết nối TCP có mã hóa**

=> Toàn vẹn dữ liệu

=> Chứng thực đầu cuối

- TLS được triển khai ở **Tầng Ứng dụng**

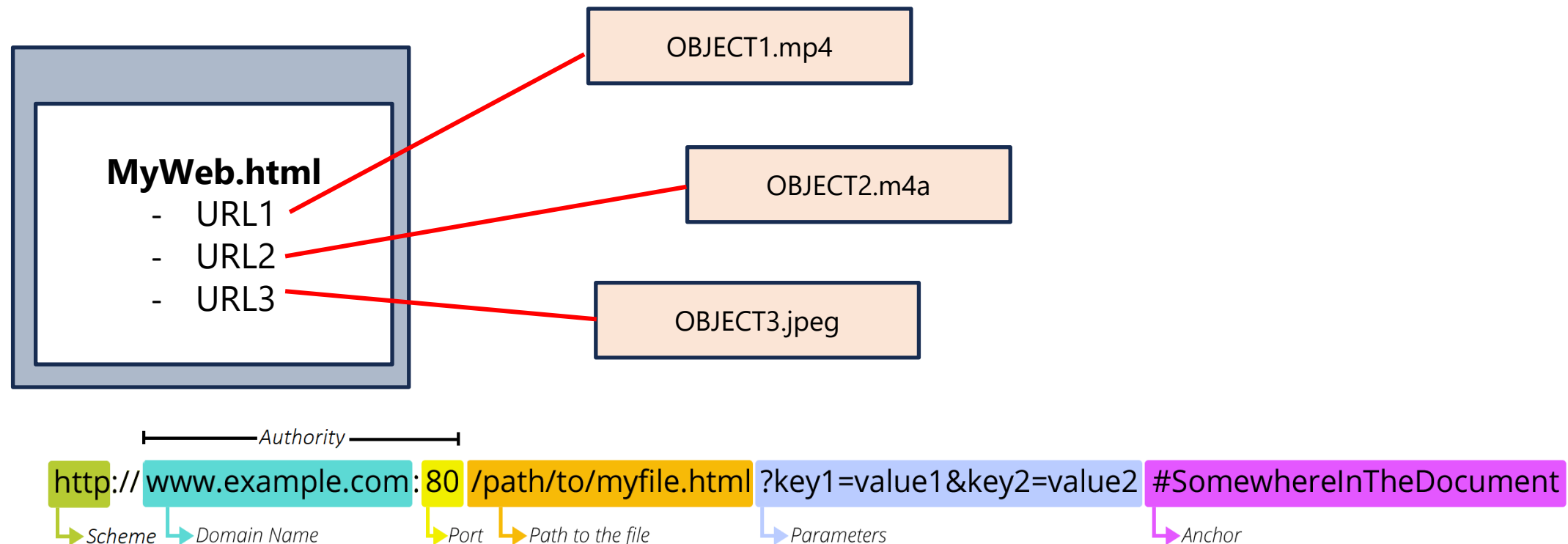
=> Các ứng dụng sử dụng các thư viện TLS => làm việc với TCP



2. APPLICATION LAYER

2.2. Web và HTTP

- **Web: Ứng dụng** tập hợp các object thích hợp với World Wide Web, được thực thi ở trình duyệt web.



2. APPLICATION LAYER

2.2. Web và HTTP - Tổng quan về HTTP

Giao thức tầng ứng dụng cho Web

Mô hình Client – Server

- **Client:** trình duyệt - yêu cầu, tiếp nhận (dùng giao thức HTTP), hiển thị các đối tượng Web
- **Server:** máy chủ web - gửi (dùng giao thức HTTP) các đối tượng => đáp ứng yêu cầu

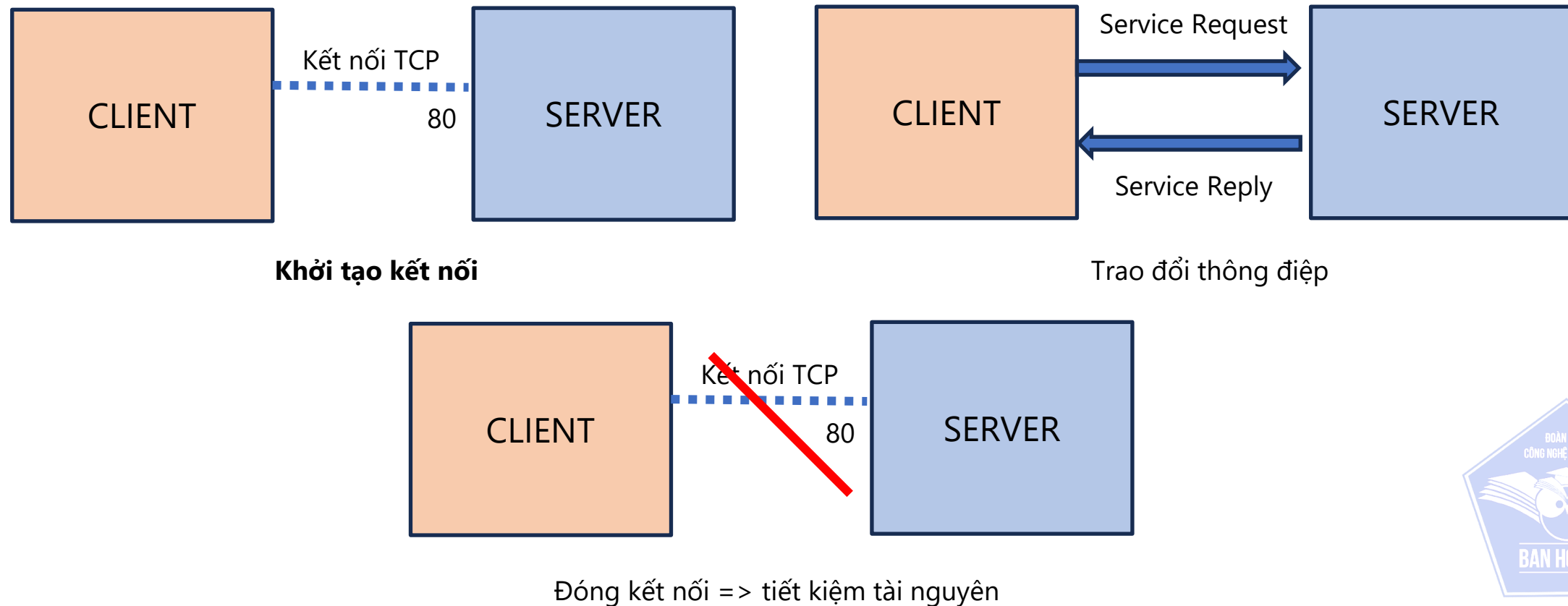


Sharing is learning

2. APPLICATION LAYER

2.2. Web và HTTP - Tổng quan về HTTP

Dùng TCP



2. APPLICATION LAYER

2.2. Web và HTTP - Tổng quan về HTTP

- HTTP **không lưu trạng thái**: Máy chủ không duy trì thông tin về các yêu cầu trước đó của máy khách
- **RTT (Round Trip Time)**: khoảng thời gian (tính bằng ms) để một gói tin nhỏ đi từ Client đến Server và quay ngược trở lại.



2. APPLICATION LAYER

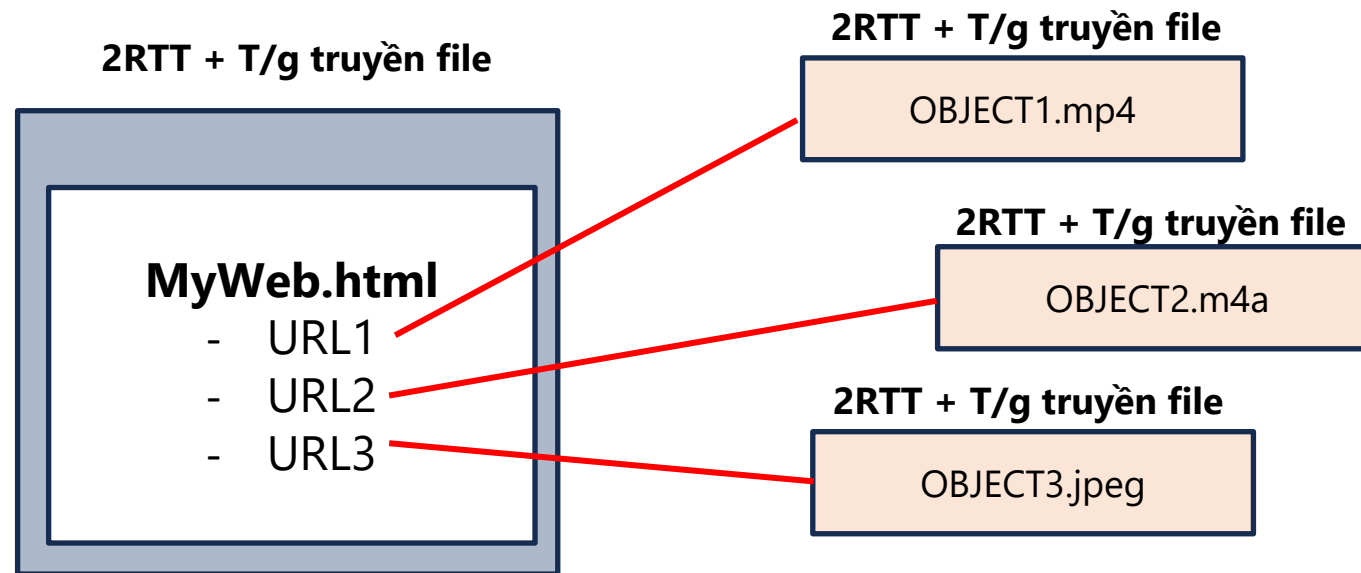
2.2. Web và HTTP - Các kết nối HTTP

HTTP không bền vững (HTTP/1.0.)	HTTP bền vững (HTTP/1.1.)
Chỉ tối đa một đối tượng được gửi qua kết nối	Nhiều đối tượng có thể được gửi thông qua duy nhất một kết nối TCP
2RTT + Thời gian truyền file /Mỗi đối tượng	With pipelining: 2RTT + 1RTT (cho tất cả đối tượng tham chiếu) + Thời gian truyền file Without pipelining: 2RTT + nRTT (cho n đối tượng tham chiếu) + Thời gian truyền file
GET/POST/HEAD	GET/POST/HEAD/PUT/DELETE

2. APPLICATION LAYER

2.2. Web và HTTP - Các kết nối HTTP

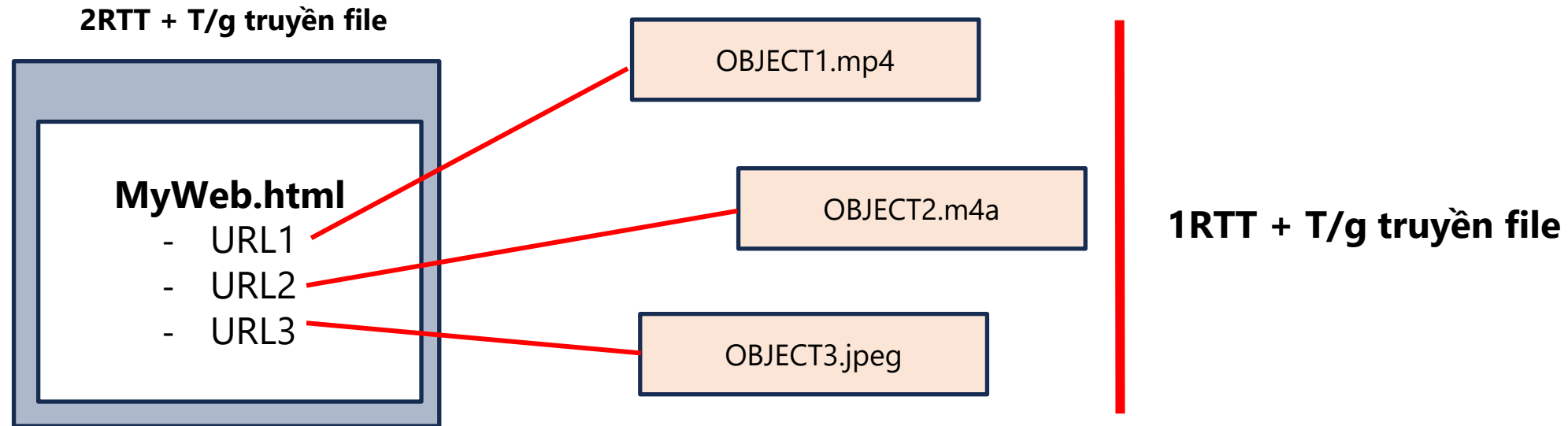
KHÔNG BỀN VỮNG



2. APPLICATION LAYER

2.2. Web và HTTP - Các kết nối HTTP

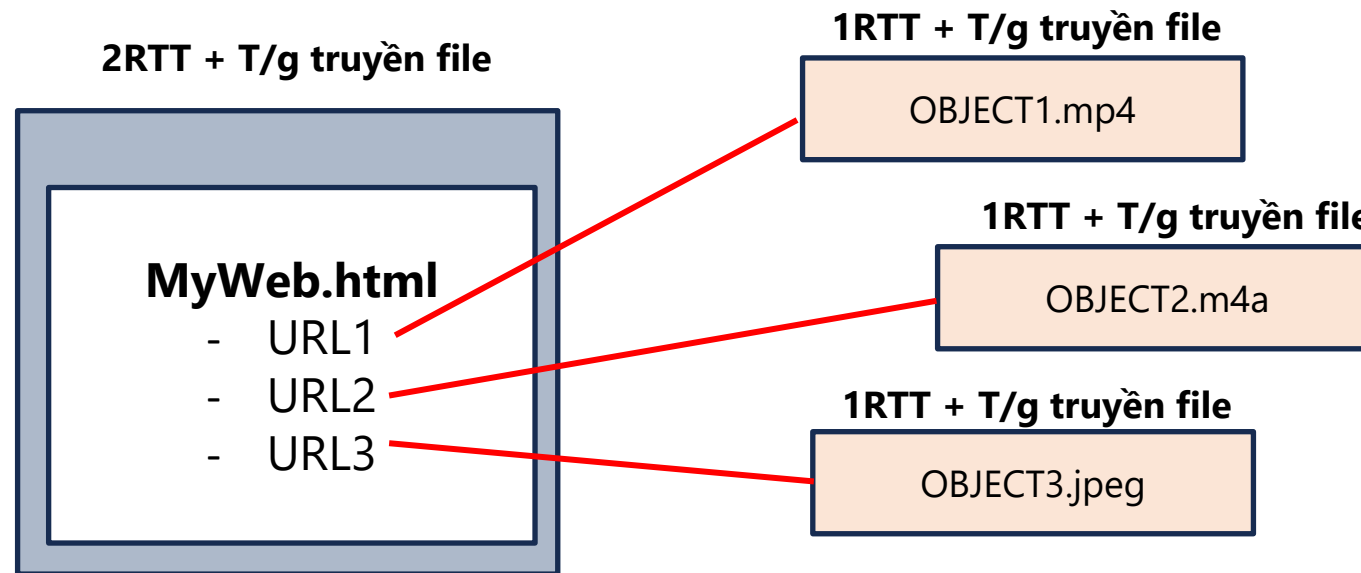
BỀN VỮNG - With pipelining



2. APPLICATION LAYER

2.2. Web và HTTP - Các kết nối HTTP

BỀN VỮNG - Without pipelining



2. APPLICATION LAYER

2.2. Web và HTTP - Thông điệp yêu cầu HTTP

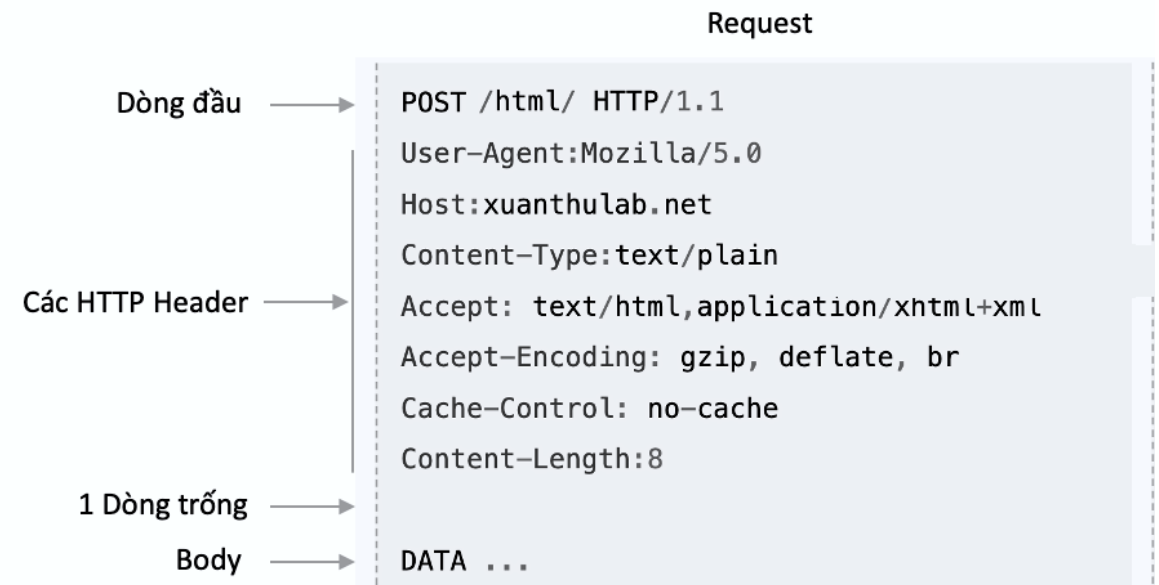
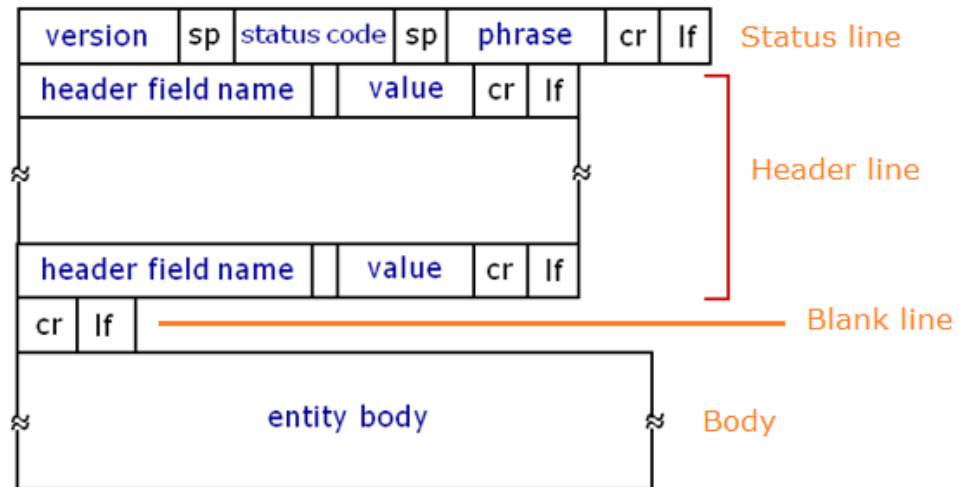
- **GET:** yêu cầu một đối tượng được xác định trong trường URL
- **POST:** Client gửi input lên Server (Entity body của thông điệp yêu cầu chứa thông tin mà người dùng đã điền)
- **HEAD:** Tương tự như GET nhưng chỉ được trả về các dòng header
- **PUT:** Tải đối tượng mới lên máy chủ, có thể ghi đè đối tượng đã có



2. APPLICATION LAYER

2.2. Web và HTTP - Thông điệp HTTP

- Viết bằng ASCII



2. APPLICATION LAYER

2.2. Web và HTTP - Thông điệp phản hồi HTTP

- Mã trạng thái xuất hiện ở dòng 1 trong thông báo phản hồi từ máy chủ đến máy khách.
- **200 OK:** Yêu cầu thành công, đối tượng được yêu cầu kèm sau trong thông điệp này
- **301 Moved Permanently:** Đối tượng được yêu cầu đã di chuyển, vị trí mới được xác định sau trong thông điệp này (trong trường Location)
- **400 Bad Request:** Máy chủ không hiểu thông điệp yêu cầu
- **404 Not Found:** Tài liệu được yêu cầu không tìm thấy trên máy chủ này
- **505 HTTP Version Not Supported**



2. APPLICATION LAYER

2.2. Web và HTTP - Cookie

Duy trì một số trạng thái giữa các phiên làm việc => quản lý phiên chạy web, cá nhân hóa trải nghiệm người dùng

Bốn thành phần:

- **Dòng header cookie** của **thông điệp phản hồi** HTTP
- **Dòng header cookie** trong **thông điệp yêu cầu** HTTP tiếp theo
- **Tập cookie** được lưu giữ trên máy của người dùng, được quản lý bởi trình duyệt của người dùng
- **Cơ sở dữ liệu** tại Web site



2. APPLICATION LAYER

2.2. Web và HTTP - Cookie

Công dụng:

- Authorization (Cấp phép)
- Giỏ hàng
- Khuyến nghị
- Trạng thái phiên người dùng (Web e-mail)
- Theo dõi hành vi của người dùng trên một trang web nhất định (cookie của bên thứ nhất)
- Theo dõi hành vi của người dùng trên nhiều trang web (cookie của bên thứ ba) mà người dùng không truy cập trang web theo dõi

2. APPLICATION LAYER

2.2. Web và HTTP - Cookie

Theo dõi bên thứ ba (Third party tracking) qua cookie:

- Bị tắt theo mặc định trong trình duyệt Firefox, Safari
- Sẽ bị vô hiệu hóa trong trình duyệt Chrome vào năm 2023

GDPR (Quy định chung về bảo vệ dữ liệu của EU) và cookie

- Khi cookie có thể xác định một cá nhân, cookie được coi là dữ liệu cá nhân, tuân theo các quy định về dữ liệu cá nhân GDPR



Sharing is learning

2. APPLICATION LAYER

2.2. Web và HTTP – Web caches (Proxy Server)

Mục tiêu:

- Thỏa mãn yêu cầu của client **không cần liên quan đến server**
- **Giảm thời gian đáp ứng** cho yêu cầu của client
- **Giảm lưu lượng** trên đường liên kết truy cập ra Internet



Sharing is learning

2. APPLICATION LAYER

2.2. Web và HTTP – HTTP/2

- Mục tiêu chính: giảm độ trễ trong các yêu cầu HTTP đa đối tượng

* **HOL (Head-of-Line Blocking)**: Một đối tượng lớn được truyền tốn nhiều thời gian
=> các đối tượng khác phải chờ gây nghẽn

HTTP/1.1.	HTTP/2
Thứ tự FCFS	Thứ tự dựa trên mức độ ưu tiên đối tượng do client chỉ định
HOL blocking, trì hoãn do phục hồi mất mát	Chia đối tượng thành các frame, lên lịch các frame => giảm HOL blocking

2. APPLICATION LAYER

2.3. Email, SMTP, IMAP - Email

Gồm 3 thành phần

- **User agents - “mail reader”**
 - Soạn thảo, chỉnh sửa, đọc thư điện tử
 - Thư đi, thư đến được lưu trữ trên máy chủ
- **Máy chủ thư điện tử (mail server)**
 - Hộp thư (mailbox) chứa thông điệp đến user
 - Hàng thông điệp (message queue) của các thông điệp mail ra ngoài (chuẩn bị gửi)

2. APPLICATION LAYER

2.3. Email, SMTP, IMAP - Email

Gồm 3 thành phần

- **Giao thức SMTP**
 - Giao thức trao đổi thư e-mail
 - Được định nghĩa trong RFC 5321

2. APPLICATION LAYER

2.3. Email, SMTP, IMAP – SMTP RFC (5321)

Mô hình **máy khách - máy chủ**, sử dụng **TCP** để truyền thông điệp thư điện tử một cách tin cậy từ máy khách đến **cổng 25** máy chủ

- Truyền trực tiếp: máy chủ gửi thư đến máy chủ nhận



2. APPLICATION LAYER

2.3. Email, SMTP, IMAP – SMTP RFC (5321)

Ba giai đoạn truyền:

- "Bắt tay"
- Truyền thông điệp
- Đóng kết nối

Tương tác lệnh/phản hồi

- Lệnh: Văn bản ASCII
- Phản hồi: Mã trạng thái và cụm từ



Sharing is learning

2. APPLICATION LAYER

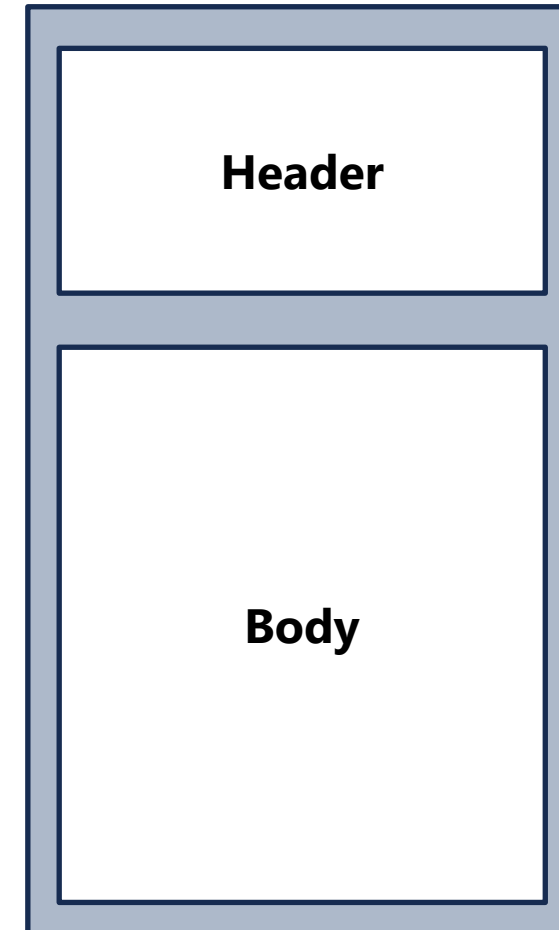
2.3. Email, SMTP, IMAP – SMTP RFC (5321): So sánh với HTTP

HTTP	SMTP
pull (kéo)	push (đẩy)
Tương tác lệnh/phản hồi, các mã trạng thái dạng ASCII	
Mỗi đối tượng được đóng gói trong thông điệp phản hồi của nó	Nhiều đối tượng được gửi trong thông điệp chứa nhiều phần
Kết nối bền vững và không bền vững	Kết nối bền vững
	Yêu cầu thông điệp (header & body) phải ở dạng ASCII 7-bit SMTP máy chủ dùng ký tự CRLF.CRLF để xác định kết thúc thông điệp

2. APPLICATION LAYER

2.3. Email, SMTP, IMAP – Định dạng thư

- RFC 2822 định nghĩa cú pháp cho thông điệp email
- Header, nằm trong phần nội dung DATA của thư
- Body: "thông điệp", chỉ các ký tự ASCII



2. APPLICATION LAYER

2.3. Email, SMTP, IMAP – Giao thức truy xuất email

- SMTP: truyền/lưu trữ vào server của người nhận.
- POP: Post Office Protocol [RFC 1939]: ủy quyền, download.
- IMAP: Internet Mail Access Protocol [RFC 1730]: truy xuất, xóa, thao tác thư mục thư được lưu trên server
- HTTP: cung cấp interface cho SMTP, IMAP (hoặc POP)

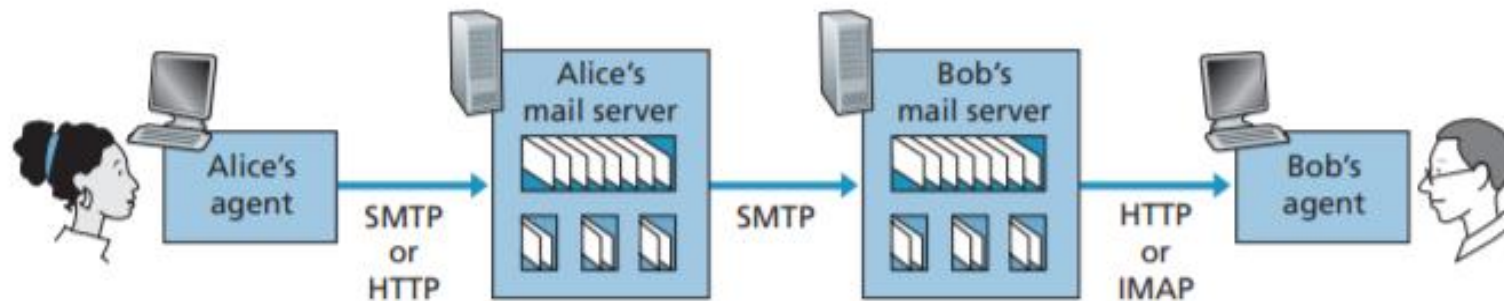


Figure 2.16 ♦ E-mail protocols and their communicating entities

2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS - Tổng quan

Định danh Internet hosts, routers:

- Địa chỉ IP address (32 bit) – được dùng để định địa chỉ gói tin
- “tên”, ví dụ uit.edu.vn – được dùng bởi con người

=> **Dịch từ tên sang địa chỉ IP và ngược lại**

=> **Dịch vụ DNS**



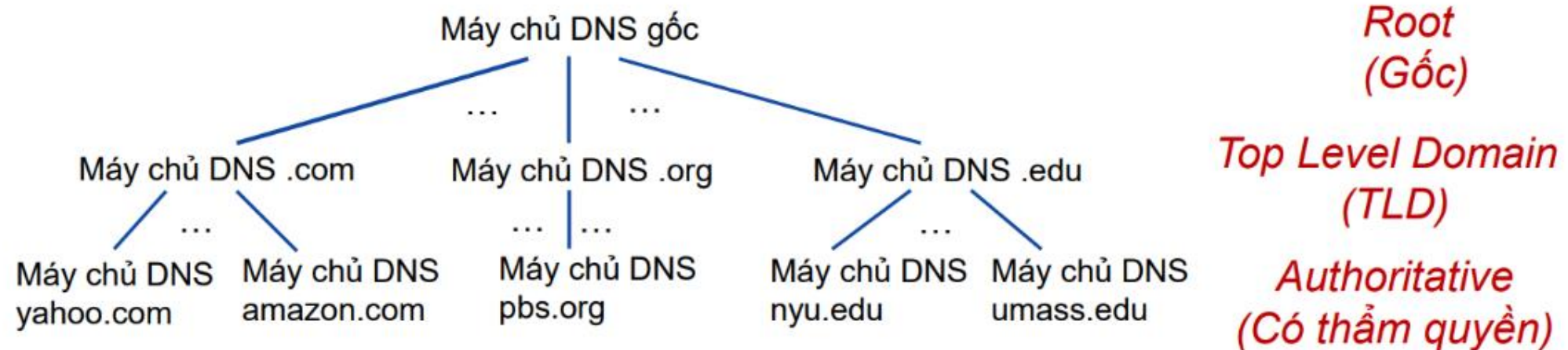
Sharing is learning

2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS - Tổng quan

Hệ thống tên miền - Domain Name System (DNS)

- Cơ sở dữ liệu phân tán được triển khai trong hệ thống phân cấp của nhiều máy chủ



2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS - Tổng quan

Các dịch vụ DNS

- Dịch tên máy ra địa chỉ IP
- Bí danh máy: Lưu các tên gốc, bí danh tương ứng
- Bí danh máy chủ thư điện tử
- Cân bằng tải: Các bản sao cho web server: nhiều địa chỉ IP tương ứng cho 1 tên miền



2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Các máy chủ

Máy chủ DNS gốc

- Chức năng Internet tối quan trọng
- DNSSEC - cung cấp bảo mật (xác thực, tính toàn vẹn của tin nhắn)
- ICANN (Internet Corporation for Assigned Names and Numbers) quản lý miền DNS gốc



2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Các máy chủ

Top-Level Domain (TLD)

- **.com, .org, .net, .edu, .aero, .jobs, .museums** và tất cả các **miền quốc gia cấp cao nhất**, ví dụ: **.cn, .uk, .fr, .ca, .jp**
- Network Solutions: đăng ký có thẩm quyền cho **.com, .net TLD**
- Tổ chức Educause: **.edu TLD**



2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Các máy chủ

Máy chủ có thẩm quyền

- (Các) máy chủ DNS riêng của tổ chức
- Cung cấp tên được cấp phép và ánh xạ địa chỉ IP cho các hệ thống đầu cuối được đặt tên của tổ chức
- Có thể được quản lý bởi tổ chức hoặc nhà cung cấp dịch vụ



2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Các máy chủ

Máy chủ DNS cục bộ

- Khi host (hệ thống đầu cuối) thực hiện truy vấn DNS, truy vấn sẽ được gửi đến máy chủ DNS cục bộ của nó
- **Máy chủ DNS cục bộ không hoàn toàn thuộc về hệ thống phân cấp**

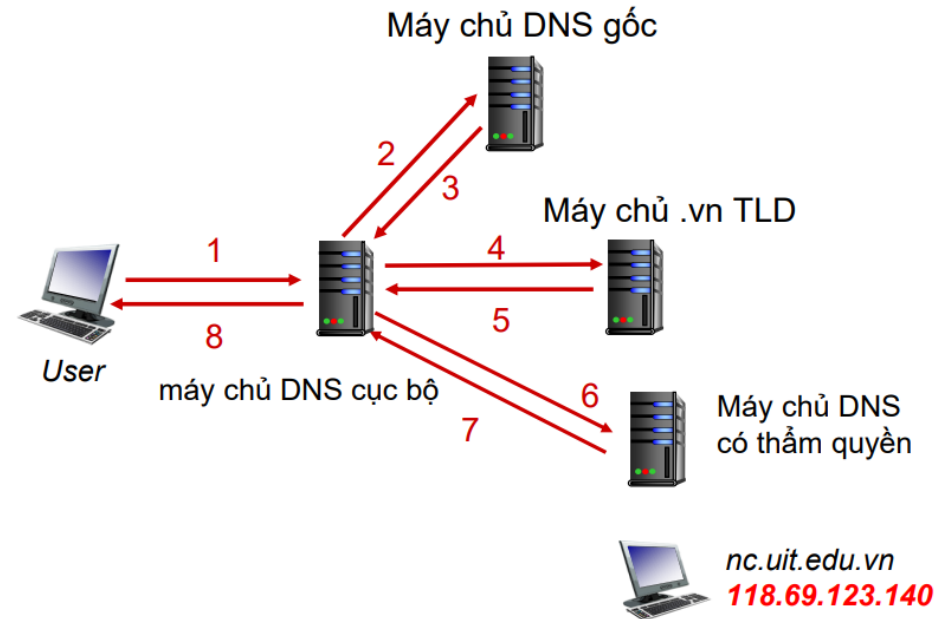


2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Phân giải tên miền DNS

Truy vấn tuần tự:

- Máy chủ được hỏi sẽ trả lời với tên của máy chủ quản lý vùng liên quan
- Nói cách khác:
 - C: "Biết A ở đâu không?"
 - S1: "Không biết, nhưng hình như S2 biết đó"
 - C: "Biết A ở đâu không?"
 - S2: "Hình như S3 biết đó"



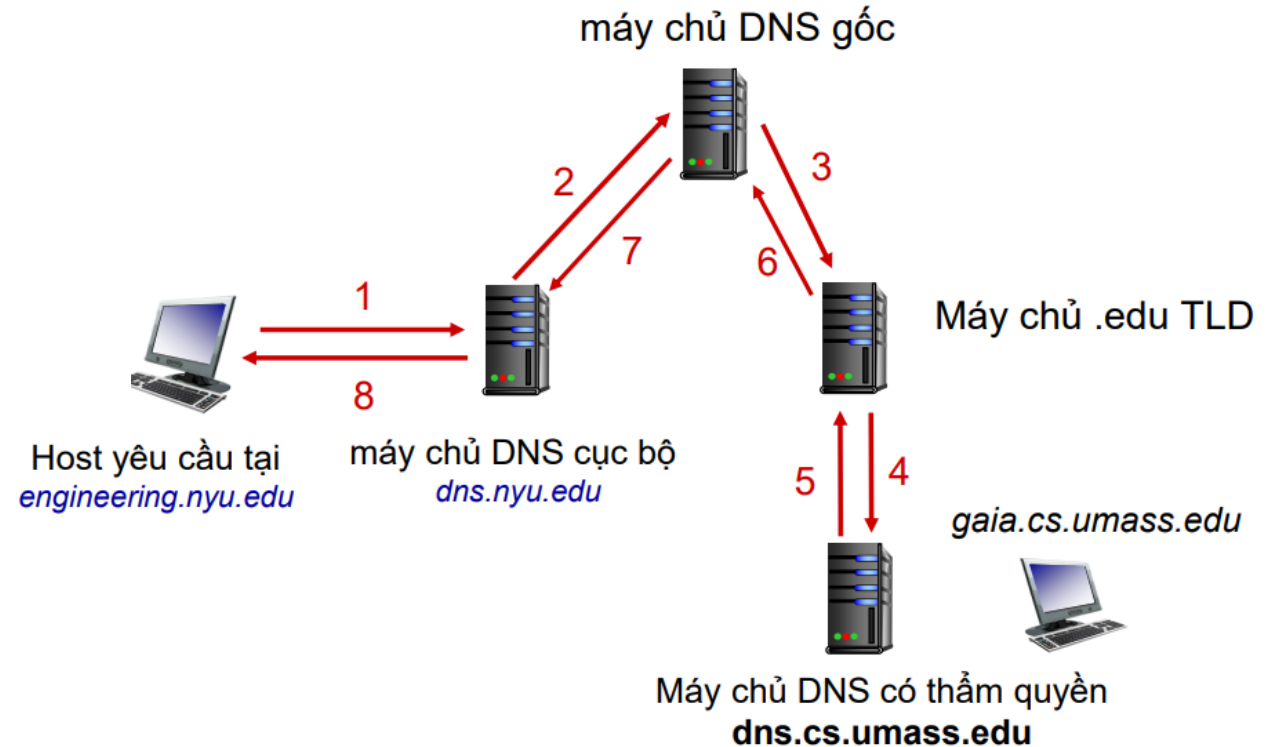
2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền

DNS – Phân giải tên miền DNS

Truy vấn đệ quy:

- Máy chủ được hỏi sẽ tiếp tục hỏi máy chủ quản lý vùng liên quan và câu trả lời được gửi về theo tuần tự
- Tải nặng ở cấp trên của hệ thống phân cấp



2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Caching DNS

Khi máy chủ tên miền học về 1 ánh xạ, nó sẽ lưu ánh xạ vào bộ nhớ cache và ngay lập tức trả về ánh xạ được lưu trong bộ nhớ cache để đáp ứng truy vấn

- Cải thiện thời gian phản hồi
- Các mục cache hết hạn (biến mất) sau một thời gian (TTL)
- Thông tin trong các máy chủ TLD thường được lưu trong bộ nhớ cache trong các máy chủ DNS cục bộ



2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Thông điệp DNS

Các dạng DNS Records

Loại	Name	Value	Lưu ở
type=A	Tên máy (host name)	Địa chỉ IP	authoritative
type=CNAME	Tên thực VD: www.ibm.com có tên thực là servereast.backup2.ibm.com	Tên gốc	TLD
type=NS	Tên miền	Tên của DNS quản lý	TLD
type=MX		Tên máy chủ thư SMTP liên kết với name	TLD

2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền

DNS – Thông điệp DNS

Định dạng thông điệp

Header:

- **identification:** số 16 bit xác định 1 truy vấn, hoặc trả lời cho truy vấn có cùng số này
- **Các cờ (flags)**

2 bytes	2 bytes
identification	flags
# questions	# answer RRs
# authority RRs	# additional RRs
questions (variable # of questions)	
answers (variable # of RRs)	
authority (variable # of RRs)	
additional info (variable # of RRs)	

2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền

DNS – Thông điệp DNS

Định dạng thông điệp

- Trường name, type của truy vấn
- Record trong trả lời của truy vấn
- Record cho server có thẩm quyền
- Một số thông tin thêm

2 bytes	2 bytes
identification	flags
# questions	# answer RRs
# authority RRs	# additional RRs
questions (variable # of questions)	
answers (variable # of RRs)	
authority (variable # of RRs)	
additional info (variable # of RRs)	

2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Nhược điểm của DNS

- Nếu điểm tập trung bị hỏng, toàn bộ hệ thống sẽ tê liệt.
- Số lượng yêu cầu phục vụ tại điểm tập trung rất lớn.
- Chi phí bảo trì hệ thống rất lớn.
- Khó khắc phục khi xảy ra sự cố, dễ bị tấn công



2. APPLICATION LAYER

2.4. Hệ thống phân giải tên miền DNS – Tấn công DNS

Tấn công DDoS

- Bắn phá các máy chủ gốc bằng lưu lượng truy cập: không thành công cho đến nay
- Bắn phá các máy chủ TLD: tiềm ẩn nhiều nguy hiểm hơn

Các cuộc tấn công giả mạo

- Chặn truy vấn DNS, trả về câu trả lời không có thật
- DNS cache poisoning (Tấn công đầu độc DNS cache)
- RFC 4033: Dịch vụ xác thực DNSSEC



2. APPLICATION LAYER

2.5. Lập trình Socket với UDP và TCP – UDP

UDP: datagram không tin cậy

- **Không "bắt tay"** trước khi gửi dữ liệu (không kết nối giữa client – server)
- Bên gửi chỉ rõ địa chỉ IP đích và số port cho mỗi packet
- Bên nhận lấy địa chỉ IP và số port của người gửi từ packet được nhận

=> Dữ liệu được truyền có thể bị mất hoặc được nhận không thứ tự



Sharing is learning

2. APPLICATION LAYER

2.5. Lập trình Socket với UDP và TCP – Ví dụ UDP Client

Python UDPClient

Bao gồm thư viện socket của Python	→	from socket import *
		serverName = 'hostname'
		serverPort = 12000
Tạo UDP socket	→	clientSocket = socket(AF_INET, SOCK_DGRAM)
Nhận thông tin nhập bằng bàn phím người dùng	→	message = input('Input lowercase sentence:')
Đính kèm tên máy chủ, cổng vào tin nhắn; gửi vào Socket	→	clientSocket.sendto(message.encode(), (serverName, serverPort))
Đọc dữ liệu trả lời (byte) từ Socket	→	modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
In ra chuỗi nhận được và đóng Socket	→	print(modifiedMessage.decode()) clientSocket.close()

2. APPLICATION LAYER

2.5. Lập trình Socket với UDP và TCP – Ví dụ UDP Server

Python UDPServer

```
from socket import *
serverPort = 12000
Tạo UDP socket → serverSocket = socket(AF_INET, SOCK_DGRAM)
Gắn socket với số cổng 12000 → serverSocket.bind(('', serverPort))
print('The server is ready to receive')
Lặp vô hạn → while True:
    Đọc từ UDP Socket vào tin nhắn, lấy địa chỉ → message, clientAddress = serverSocket.recvfrom(2048)
    của máy khách (IP máy khách và cổng)      modifiedMessage = message.decode().upper()
    Gửi chuỗi chữ hoa trở lại cho máy khách → serverSocket.sendto(modifiedMessage.encode(),
                                                                    clientAddress)
```

2. APPLICATION LAYER

2.5. Lập trình Socket với UDP và TCP – UDP

TCP: tin cậy, byte được định hướng dòng (stream-oriented)

- Server tạo socket để mời client đến liên lạc.
- Tạo socket TCP, xác định địa chỉ IP, số port của tiến trình server
- Khi client tạo socket: client TCP **thiết lập kết nối** đến server TCP.
- Khi đã tiếp xúc với client, server TCP tạo socket mới cho tiến trình server để truyền thông với client đó.

=> Truyền các byte tin cậy và theo thứ tự giữa client và server



Sharing is learning

2. APPLICATION LAYER

2.5. Lập trình Socket với UDP và TCP – Ví dụ TCP Client

Python TCPClient

Tạo Socket TCP đến máy chủ, cổng từ xa 12000

Không cần đính kèm tên máy chủ, cổng

```
from socket import *
serverName = 'servername'
serverPort = 12000
→ clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input('Input lowercase sentence:')
clientSocket.send(sentence.encode())
→ modifiedSentence = clientSocket.recv(1024)
print ('From Server:', modifiedSentence.decode())
clientSocket.close()
```

2. APPLICATION LAYER

2.5. Lập trình Socket với UDP và TCP – Ví dụ TCP Server

Python TCPServer

	→	from socket import *
		serverPort = 12000
Tạo Socket TCP chào đón	→	serverSocket = socket(AF_INET, SOCK_STREAM)
		serverSocket.bind(('', serverPort))
Máy chủ bắt đầu lắng nghe các yêu cầu TCP đến	→	serverSocket.listen(1)
		print('The server is ready to receive')
Lập vô hạn	→	while True:
Máy chủ đợi accept() cho yêu cầu đến, socket mới được tạo trở về	→	connectionSocket, addr = serverSocket.accept()
Đọc các byte từ socket (nhưng không đọc địa chỉ như UDP)	→	sentence = connectionSocket.recv(1024).decode()
		capitalizedSentence = sentence.upper()
		connectionSocket.send(capitalizedSentence.encode())
Đóng kết nối đến máy khách này (nhưng không đóng socket chào đón)	→	connectionSocket.close()

2. APPLICATION LAYER

2.6. Bài tập

Sử dụng HTTP 1.1. Giả sử **bốn máy chủ DNS** được truy cập trước khi máy chủ nhận được địa chỉ IP từ DNS. Các máy chủ DNS có RTT lần lượt là **42, 5, 21 và 13ms**. Trang Web được liên kết với **văn bản HTML tham chiếu tới 7 đối tượng**. RTT giữa máy chủ cục bộ và máy chủ Web chứa đối tượng là $RTT_{\text{HTTP}} = 34 \text{ ms}$. **Bỏ qua thời gian truyền tập tin, cần bao nhiêu thời gian?**

2. APPLICATION LAYER

2.6. Bài tập

Sử dụng HTTP 1.1. Giả sử **bốn máy chủ DNS** được truy cập trước khi máy chủ nhận được địa chỉ IP từ DNS. Các máy chủ DNS có RTT lần lượt là **42, 5, 21 và 13ms**. Trang Web được liên kết với **văn bản HTML tham chiếu tới 7 đối tượng**. RTT giữa máy chủ cục bộ và máy chủ Web chứa đối tượng là $RTT_{\text{HTTP}} = 34 \text{ ms}$. **Bỏ qua thời gian truyền tập tin, cần bao nhiêu thời gian?**

Văn bản HTML: 2RTT

7 đối tượng: 1RTT/ đối tượng

Trước đó phải đi qua 4 máy chủ DNS có RTT là 42, 5, 21, 13ms

2. APPLICATION LAYER

2.6. Bài tập

Sử dụng HTTP 1.1. Giả sử **bốn máy chủ DNS** được truy cập trước khi máy chủ nhận được địa chỉ IP từ DNS. Các máy chủ DNS có RTT lần lượt là **42, 5, 21 và 13ms**. Trang Web được liên kết với **văn bản HTML tham chiếu tới 7 đối tượng**. RTT giữa máy chủ cục bộ và máy chủ Web chứa đối tượng là $RTT_{\text{HTTP}} = 34 \text{ ms}$. **Bỏ qua thời gian truyền tập tin, cần bao nhiêu thời gian?**

$$\text{Thời gian} = 42 + 5 + 21 + 13 + 2RTT + 7RTT = 387\text{ms}$$

2. APPLICATION LAYER

2.6. Bài tập

HTTP không bền vững (non-persistent HTTP) có nghĩa là:

- A. Chỉ tối đa một đối tượng được gửi qua kết nối TCP. Kết nối sau đó sẽ bị đóng.
- B. Nhiều đối tượng có thể được gửi qua một kết nối TCP giữa client và server.
- C. Chỉ tối đa một webpage được gửi qua kết nối TCP. Kết nối sau đó sẽ bị đóng.
- D. Nhiều webpage có thể được gửi qua một kết nối TCP giữa client và server.

2. APPLICATION LAYER

2.6. Bài tập

HTTP không bền vững (non-persistent HTTP) có nghĩa là:

A. Chỉ tối đa một đối tượng được gửi qua kết nối TCP. Kết nối sau đó sẽ bị đóng.

B. Nhiều đối tượng có thể được gửi qua một kết nối TCP giữa client và server.

C. Chỉ tối đa một webpage được gửi qua kết nối TCP. Kết nối sau đó sẽ bị đóng.

D. Nhiều webpage có thể được gửi qua một kết nối TCP giữa client và server.

2. APPLICATION LAYER

2.6. Bài tập

Một địa chỉ IP có thể ứng với bao nhiêu miền?

- A. 1
- B. 2
- C. 3
- D. Nhiều

2. APPLICATION LAYER

2.6. Bài tập

Một địa chỉ IP có thể ứng với bao nhiêu miền?

A. 1

B. 2

C. 3

D. Nhiều

2. APPLICATION LAYER

2.6. Bài tập

Phân tích một phần gói tin HTTP request từ trình duyệt gửi lên Web Server như sau:

GET /docs/index.html HTTP/1.1\r\n

Host: www-net.cs.umass.edu\r\n

- A. Trình duyệt dùng kết nối không thường trực (non-persistent) và URL đầy đủ của trang web được yêu cầu là: `www-net.cs.umass.edu/docs/index.html`
- B. Trình duyệt dùng kết nối thường trực (persistent) và URL đầy đủ của trang web được yêu cầu là: `www-net.cs.umass.edu/docs/index.html`
- C. Trình duyệt dùng kết nối thường trực (persistent) và URL đầy đủ của trang web được yêu cầu là: `www-net.cs.umass.edu/index.html`
- D. Trình duyệt dùng kết nối thường trực (persistent) và URL đầy đủ của trang web được yêu cầu là: `www-net.cs.umass.edu`

2. APPLICATION LAYER

2.6. Bài tập

Phân tích một phần gói tin HTTP request từ trình duyệt gửi lên Web Server như sau:

GET /docs/index.html HTTP/1.1\r\n

Host: www-net.cs.umass.edu\r\n

A. Trình duyệt dùng kết nối không thường trực (non-persistent) và URL đầy đủ của trang web được yêu cầu là: `www-net.cs.umass.edu/docs/index.html`

B. Trình duyệt dùng kết nối thường trực (persistent) và URL đầy đủ của trang web được yêu cầu là: `www-net.cs.umass.edu/docs/index.html`

C. Trình duyệt dùng kết nối thường trực (persistent) và URL đầy đủ của trang web được yêu cầu là: `www-net.cs.umass.edu/index.html`

D. Trình duyệt dùng kết nối thường trực (persistent) và URL đầy đủ của trang web được yêu cầu là: `www-net.cs.umass.edu`

3. TRANSPORT LAYER

3.1. Các dịch vụ tầng vận chuyển - Tổng quan

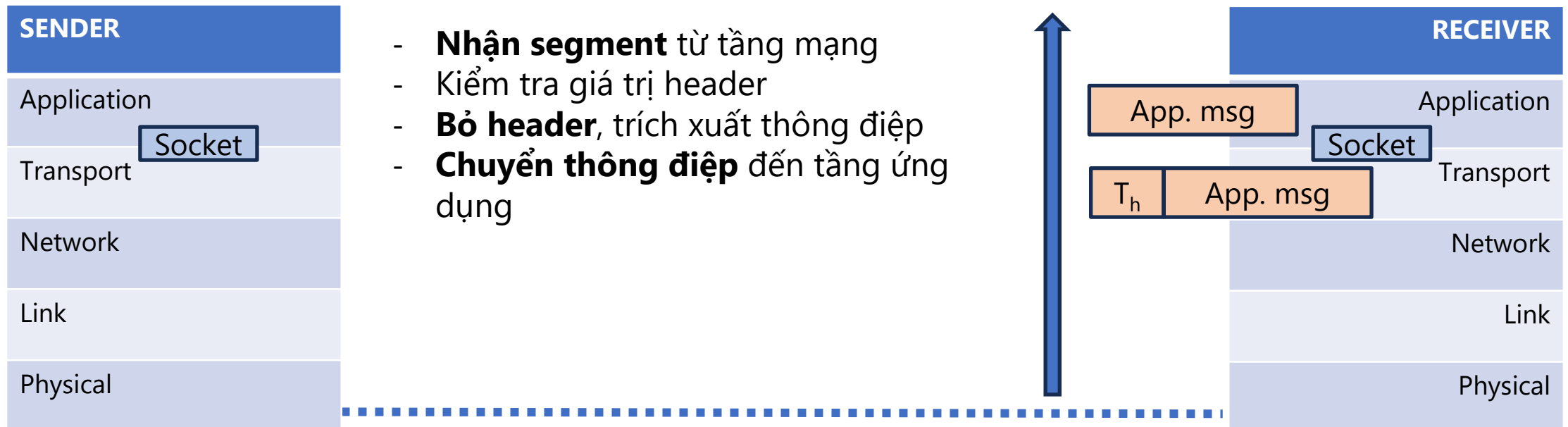
- Cung cấp “**truyền thông luận lý (logical communication)**” giữa các tiến trình trên các “host” khác nhau
- Hoạt động trên các thiết bị đầu cuối



3. TRANSPORT LAYER

3.1. Các dịch vụ tầng vận chuyển - Tổng quan

- Cung cấp “**truyền thông luận lý (logical communication)**” giữa các tiến trình trên các “host” khác nhau
- Hoạt động trên các thiết bị đầu cuối



3. TRANSPORT LAYER

3.1. Các dịch vụ tầng vận chuyển - TCP (Transmission Control Protocol)

- Tin cậy, vận chuyển đúng thứ tự
- Điều khiển tắc nghẽn
- Điều khiển luồng
- Thiết lập kết nối

3. TRANSPORT LAYER

3.1. Các dịch vụ tầng vận chuyển - UDP (User Datagram Protocol)

- Không tin cậy, truyền nhận không đúng thứ tự
- Phần mở rộng của giao thức IP "best-effort"

3. TRANSPORT LAYER

3.1. Các dịch vụ tầng vận chuyển

Quan hệ giữa Tầng Vận chuyển và tầng Mạng

- Tầng Mạng: truyền thông logic giữa các host
- Tầng Vận chuyển: truyền thông logic giữa các tiến trình. Dựa trên dịch vụ tầng mạng

3. TRANSPORT LAYER

3.2. Multiplexing và Demultiplexing

Ví dụ: Có 4 tiến trình ứng dụng A, B, C, D đang chạy trên Client và sẵn sàng nhận messages từ Server.

Vấn đề: *Làm sao để biết message nào nên được chuyển đến đâu?*

=> **Định danh của Socket => Ghép các kênh lại để gửi qua Client (Multiplexing)**

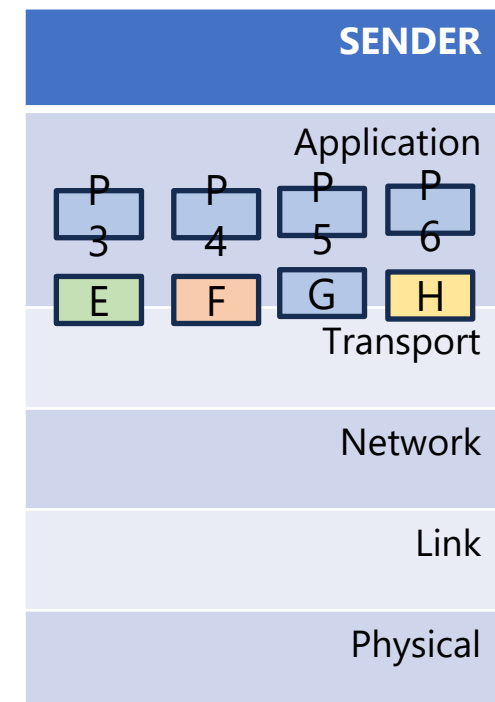
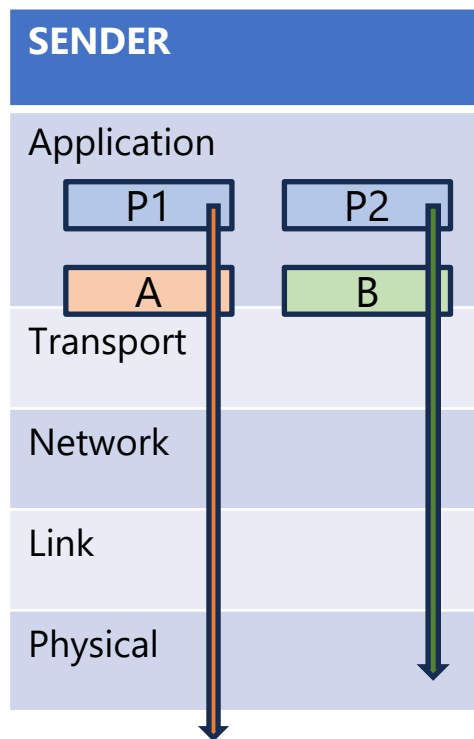
=> **Dựa vào định danh => Tách các kênh ra và đưa đến tiến trình hợp lý (Demultiplexing)**

*** Lưu ý: Multiplexing và Demultiplexing hoạt động ở TẤT CẢ CÁC TẦNG**

3. TRANSPORT LAYER

3.2. Multiplexing và Demultiplexing - Multiplexing

- Tại **bên gửi**
- Nhận message từ **socket**, **thêm header** của tầng vận chuyển.



3. TRANSPORT LAYER

3.2. Multiplexing và Demultiplexing - Demultiplexing

- Tại **bên nhận**
- Sử dụng **thông tin trong header** để chuyển segment nhận được đến **đúng socket**.



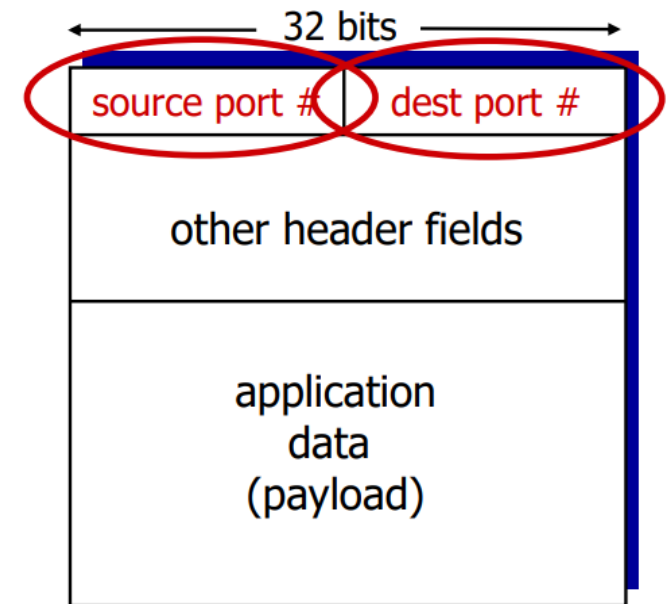
3. TRANSPORT LAYER

3.2. Multiplexing và Demultiplexing - Demultiplexing

- Host nhận **gói dữ liệu** (IP nguồn - IP đích)
- Mỗi gói chứa một **segment** (số port nguồn - số port đích)

=> Dùng **IP và số port** để đưa message đến socket đúng

ĐỊNH DANH



TCP/UDP segment format

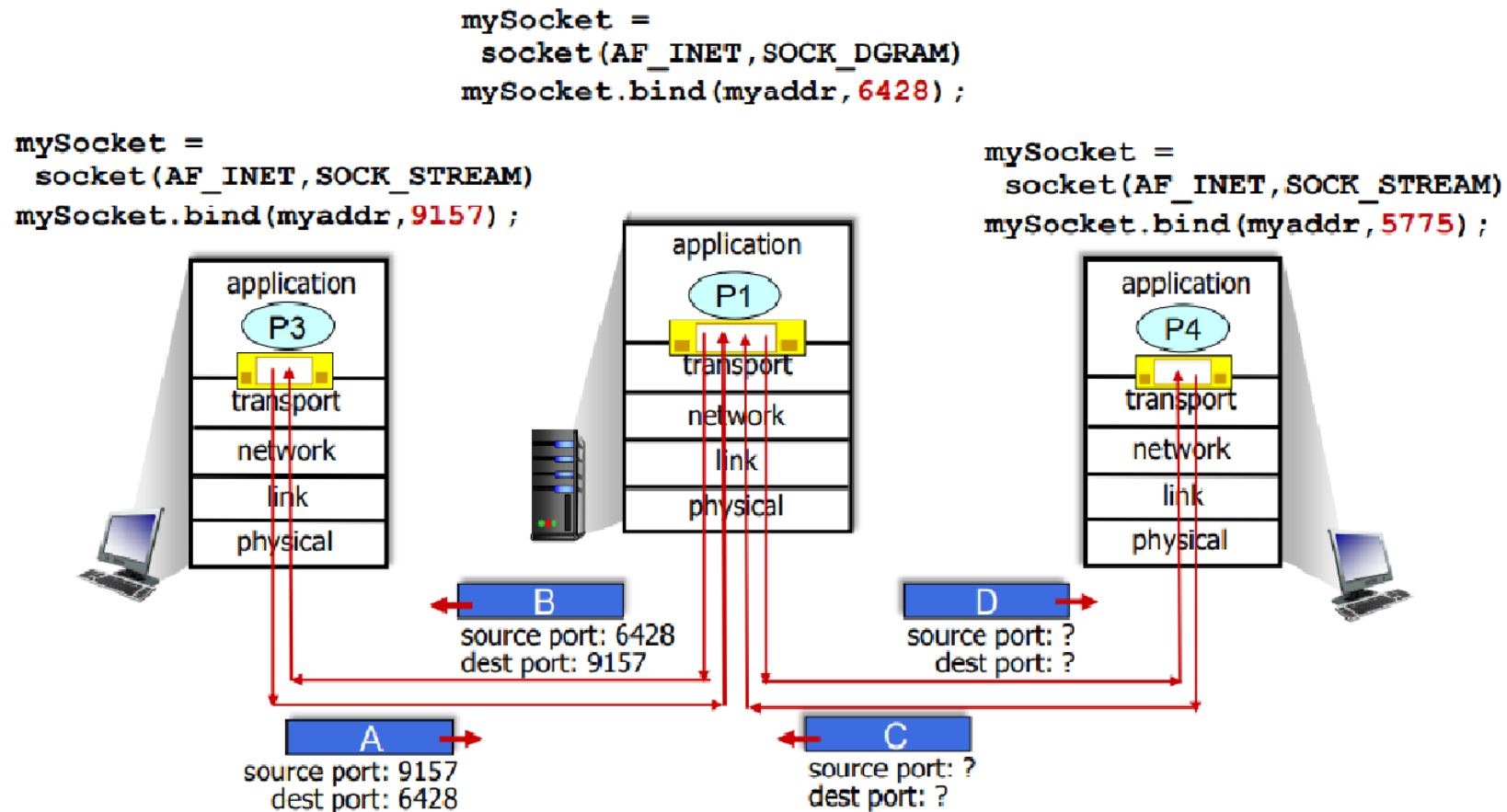
3. TRANSPORT LAYER

3.2. Multiplexing và Demultiplexing – Demultiplexing không kết nối

- **Host nhận segment UDP: Kiểm tra số port đích trong segment. Đưa segment UDP đến socket tương ứng**
- Các gói dữ liệu IP với **cùng số port đích**, nhưng khác địa chỉ IP nguồn và/hoặc khác số port nguồn sẽ được chuyển đến **cùng socket tại máy đích**

3. TRANSPORT LAYER

3.2. Multiplexing và Demultiplexing – Demultiplexing không kết nối



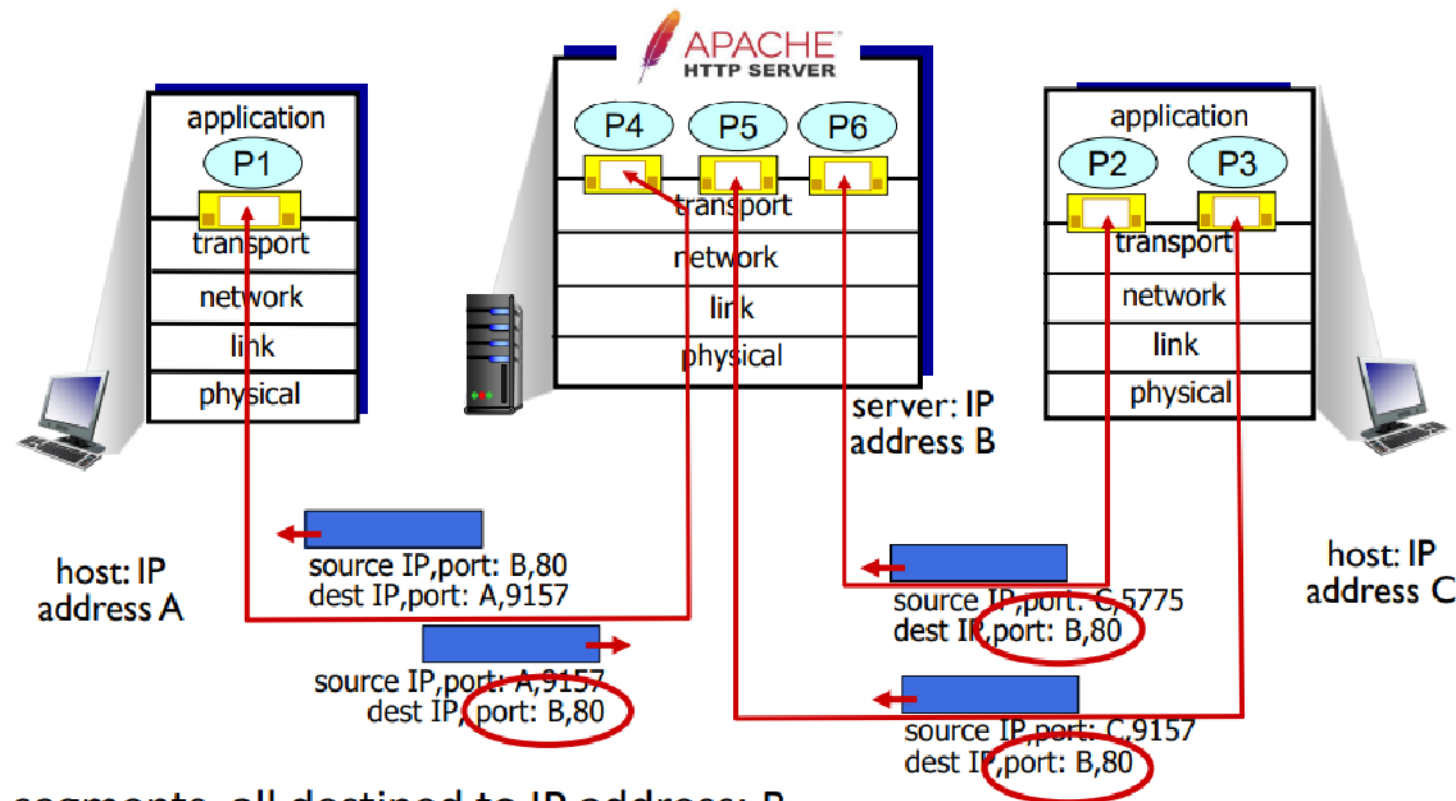
3. TRANSPORT LAYER

3.2. Multiplexing và Demultiplexing – Demultiplexing hướng kết nối

- TCP socket được xác định bởi **4-tuple (4 thông tin chính)**: IP nguồn, số port nguồn, IP đích, số port đích
 - Demux: bên nhận **sử dụng cả 4 thông tin chính** để chuyển segment đến đúng socket
 - Server có thể hỗ trợ nhiều TCP socket cùng lúc tương ứng với **mỗi bộ thông tin: mỗi client đang kết nối tới nó**
- * Lưu ý: Kết nối HTTP không bền vững sẽ có socket khác nhau cho mỗi yêu cầu**

3. TRANSPORT LAYER

3.2. Multiplexing và Demultiplexing – Demultiplexing hướng kết nối



Three segments, all destined to IP address: B,
dest port: 80 are demultiplexed to *different* sockets

3. TRANSPORT LAYER

3.3. Vận chuyển phi kết nối UDP (RFC 768) - Tổng quan

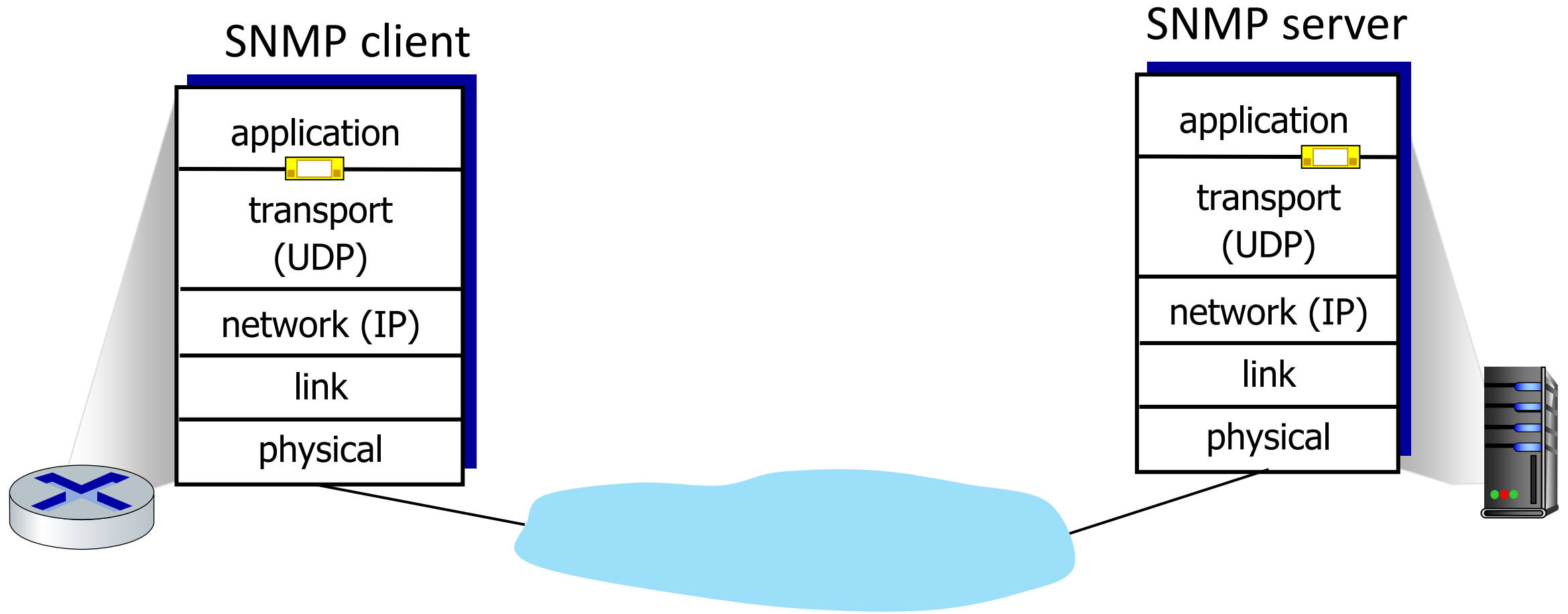
Giao thức đơn giản:

- Segment có thể bị mất, sai thứ tự
- “Best-effort”: gửi và hy vọng có kết quả tốt nhất
- Connectionless: không "bắt tay" giữa bên nhận và bên gửi, xử lý độc lập mỗi segment
- **Ưu điểm:**
 - Không có quá trình thiết lập kết nối
 - Có checksum => kiểm tra lỗi

Ứng dụng: DNS, SNMP... Các ứng dụng đa phương tiện trực tuyến **chịu mất mát data nhưng cần tốc độ**

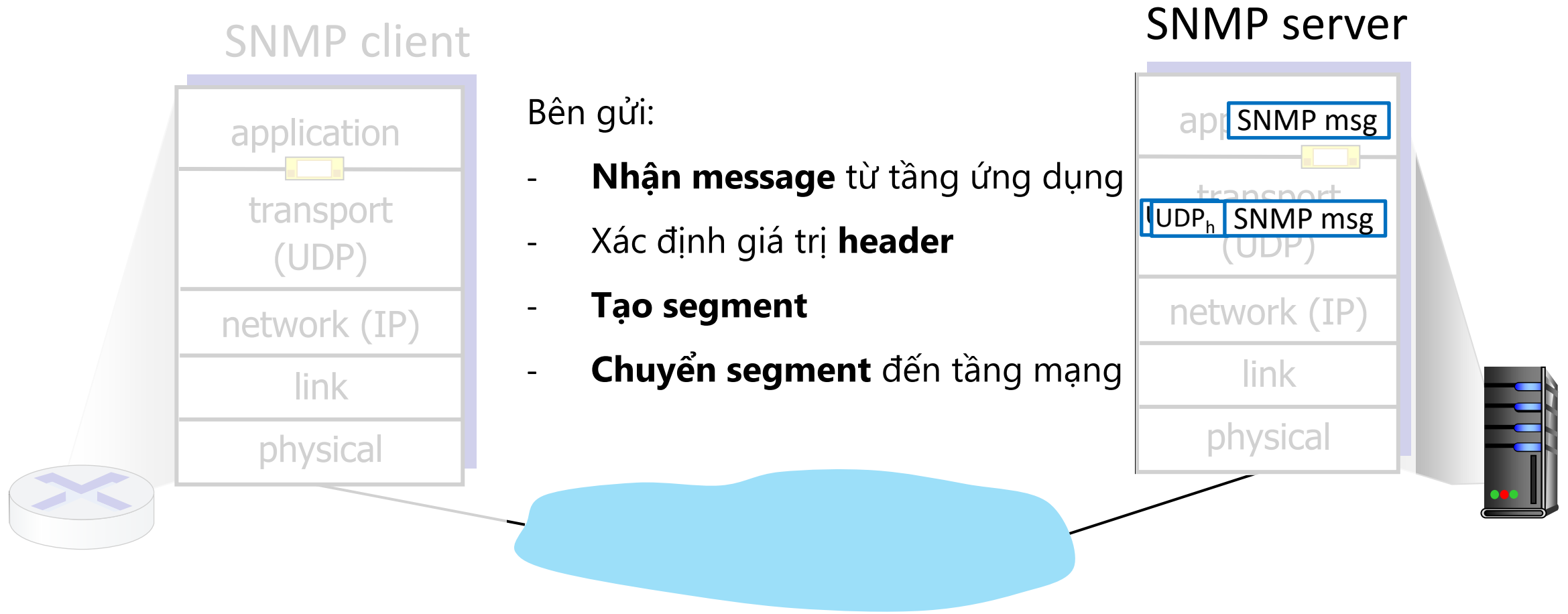
3. TRANSPORT LAYER

3.3. Vận chuyển phi kết nối UDP (RFC 768) - Hoạt động



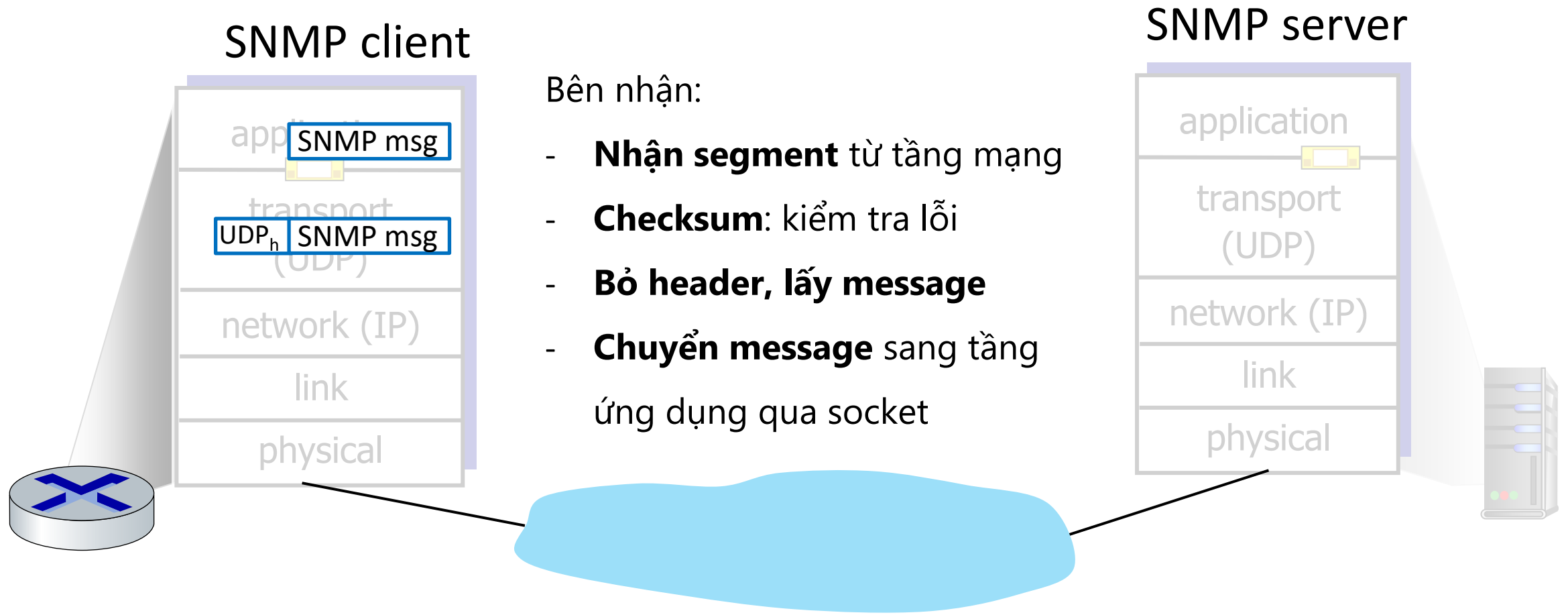
3. TRANSPORT LAYER

3.3. Vận chuyển phi kết nối UDP (RFC 768) - Hoạt động



3. TRANSPORT LAYER

3.3. Vận chuyển phi kết nối UDP (RFC 768) - Hoạt động



3. TRANSPORT LAYER

3.3. Vận chuyển phi kết nối UDP (RFC 768) - Header và Checksum

Header: 8 bytes (Mỗi phần là một chuỗi 16bits)

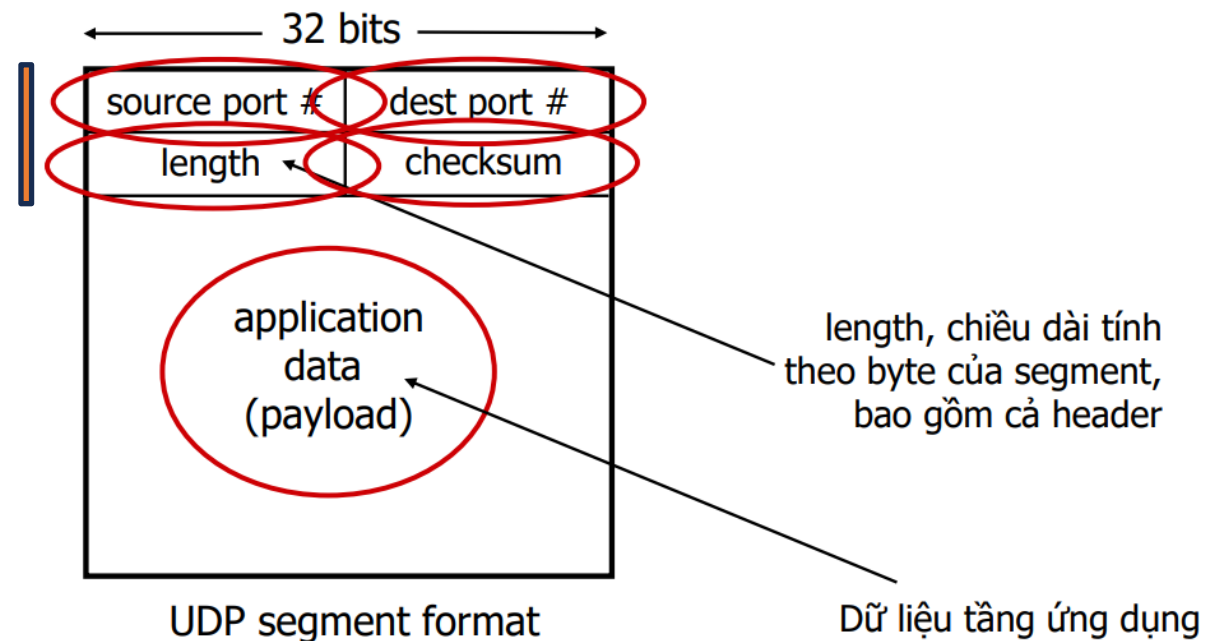
Checksum:

- Một giá trị từ bên gửi
- **Để kiểm tra lỗi truyền**
- Bên nhận sẽ tính checksum của segment nhận được => so sánh

Vấn đề của checksum:

- Nếu bị lỗi nhiều bit, có thể không phát hiện lỗi

HEADER



3. TRANSPORT LAYER

3.3. Vận chuyển phi kết nối UDP (RFC 768) - Checksum

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
wraparound	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
sum	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

LỖI TRUYỀN



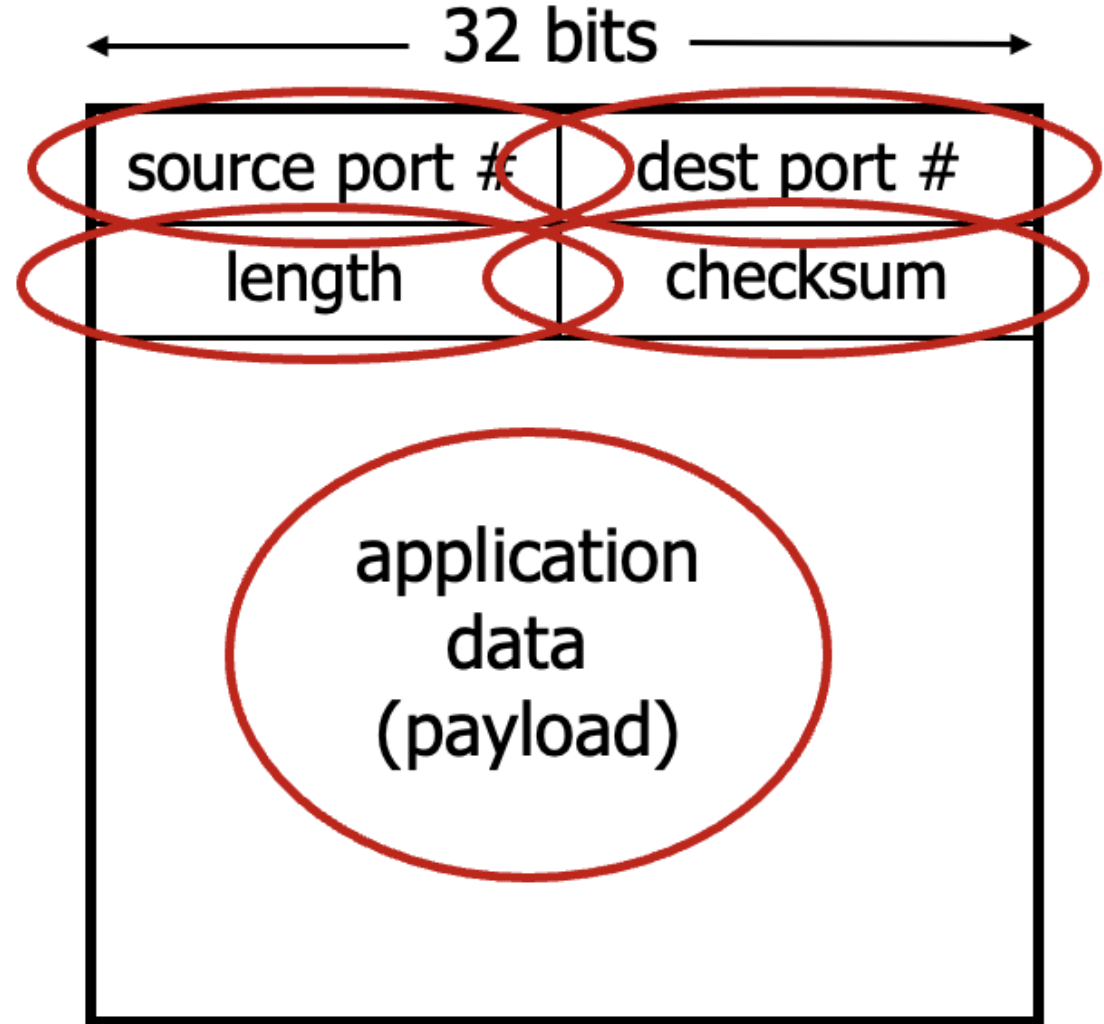
**Checksum: tổng bù 1
của hai chuỗi bit**

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
wraparound	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1
sum	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

Khi có 2 bit bị thay đổi, checksum không thay đổi

UDP segment header

- UDP Header bao gồm:
- Source Port (2 bytes),
- Destination Port (2 bytes),
- Length (2 bytes),
- Checksum (2 bytes).
- Tổng cộng là 8 bytes.



UDP segment format

Internet checksum: Ví Dụ

ví dụ : cộng 2 chuỗi 16-Bits

	1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
	1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
	<hr/>
wraparound	1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
	<hr/>
sum	1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0
checksum	0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

Note: nếu kết quả là 17 bits, lấy bit cuối cùng bên trái cộng vào kết quả.

Bài Tập Tự Luyện Checksum

[:http://gaia.cs.umass.edu/kurose_ross/interactive/internet_checksum.php](http://gaia.cs.umass.edu/kurose_ross/interactive/internet_checksum.php)

Dạng bài câu hỏi về UDP

Câu 1 :Giao thức nào dưới đây không đảm bảo dữ liệu gửi đi có tới máy nhận hoàn chỉnh hay không?

- A.TCP
- B.UDP
- C.ARP
- D.RARP

Chọn B.UDP



Dạng bài câu hỏi lý thuyết về UDP

Câu 2 :Các ứng dụng dùng giao thức UDP:

- A. Web, truyền file, Email
- B. Web, DNS, điện thoại Internet
- C. Hội thảo từ xa, điện thoại Internet, streaming media
- D. Telnet, DNS, Email

Chọn B



Sharing is learning

Dạng bài câu hỏi lý thuyết về UDP

Câu 3 :Độ dài của UDP Header là:

- A. 20 bytes
- C. 4 bytes
- B. 8 bytes
- D. 6 bytes

UDP Header bao gồm: Source Port (2 bytes), Destination Port (2 bytes), Length (2 bytes), Checksum (2 bytes). Tổng cộng là 8 bytes.

Chọn B



Dạng bài câu hỏi lý thuyết về UDP

Câu 4 :Tính UDP checksum của dữ liệu được mô tả bằng 2 dãy số nhị phân sau: 1001 0110 1000 0011 và 0101 0110 1010 1001.

- A. 1001 0110 1101 0010
- B. 1 0110 1101 0010 1100
- C. 0 1001 0110 1101 0010
- D. 0001 0010 1101 0011

Chọn D



Giao Thức Truyền Tin Cây RDT

- **RDT 1.0**: Không thành phần
- **RDT 2.0**:ACKs, NAKs,Checksum
- **RDT 2.1**:ACKs,NAKs,Checksum,Sequence number
- **RDT 2.2** :ACKs,Checksum,Sequence number
- **RDT 3.0** :ACKs,Checksum,Sequence number,Timer

Lưu ý : RDT 3.0 là mô hình đầu tiên trong loạt định dạng RDT có khả năng xử lý gói tin bị mất.

Dạng bài câu hỏi về RDT

Câu 1 :Trong giao thức truyền dữ liệu tin cậy (RDT), giao thức nào được xử lí trong trường hợp mất gói tin ACK?

A.RDT 3.0

B.RDT 2.2

C.RDT 2.1

D.Không thể xử lí được việc mất gói tin ACK

Chọn A



Cơ Chế Go Back N (RDT 3.0)

https://media.pearsoncmg.com/aw/ecs_kurose_comp_network_7/cw/content/interactiveanimations/go-back-n-protocol/index.html

Go Back N

Bên gửi: gửi tối đa N packets liên tiếp mà không cần chờ ACK

- Tính timer cho packet gửi sớm nhất nhưng chưa được ACK
- timeout(n): **gửi lại tất cả** các packet trong từ vị trí (n)

Go Back N

1.Nhận packet đúng thứ tự: gửi ACK tương ứng với packet đúng thứ tự

2.Nhận packet không đúng thứ tự:

- Gửi lại ACK tương ứng với packet đúng thứ tự có số **TT cao nhất**

Go-Back-N in action

sender window (N=4)

0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8

0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8

0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8

sender

send pkt0
send pkt1
send pkt2
send pkt3
(wait)

rcv ack0, send pkt4
rcv ack1, send pkt5

ACK trùng lặp



pkt 2 timeout

send pkt2
send pkt3
send pkt4
send pkt5

receiver

receive pkt0, send ack0
receive pkt1, send ack1

receive pkt3, discard,
(re)send ack1

receive pkt4, discard,
(re)send ack1

receive pkt5, discard,
(re)send ack1

rcv pkt2, deliver, send ack2
rcv pkt3, deliver, send ack3
rcv pkt4, deliver, send ack4
rcv pkt5, deliver, send ack5

Dạng bài câu hỏi về RDT

Câu 1: Trong nguyên lý truyền dữ liệu tin cậy, giao thức nào sau đây mà bên gửi cho phép gửi nhiều gói đồng thời mà không cần chờ ACK ?

- A. rdt 1.0
- B. rdt 2.0
- C. rdt 3.0
- D. Pipelined

Chọn C



Dạng bài câu hỏi về Go Back N

Câu 2: Trong hoạt động go back N, phía gửi phát đồng thời 5 gói 0,1,2,3,4. Phía nhận thu chính xác 5 gói và trả về 5 ACK nhưng phía gửi chỉ nhận được ACK(0),ACK(1),ACK(4). Tiếp theo phía gửi sẽ phát ?

- A. Phát gói 5,6,7 và chờ hết thời gian để phát lại gói 2,3
- B. Phát gói 5,6 và chờ hết thời gian để phát lại gói 2,3,4
- C. Phát gói 5,6,7,8,9
- D. Chờ hết thời gian để phát lại gói 2,3

Chọn B



Câu 3 :Trong RDT 3.0, chuyện gì sẽ xảy ra khi bên gửi không nhận được ACK của bên nhận?

- A. Bên gửi gửi ACK trùng lặp cho bên nhận để báo hiệu về lỗi phát sinh
- B. Bên gửi tự phát hiện lỗi và gửi lại gói tin sau khi thời gian chờ hết hạn
- C. Bên gửi gửi NAK cho bên nhận để báo hiệu về lỗi phát sinh
- D. Bên gửi sẽ dừng quá trình truyền dữ liệu cho bên nhận

Chọn B



Ôn Tập

<https://azota.vn/de-thi/9nih2r>

Tài Liệu Tham Khảo: [Link Tham Khảo](#)



QR : Điểm Danh



Sharing is learning

BAN HỌC TẬP CÔNG NGHỆ PHẦN MỀM

TRAINING GIỮA KỲ HỌC KỲ I NĂM HỌC 2022 – 2023



Sharing is learning

HẾT

**CẢM ƠN CÁC BẠN ĐÃ THEO DÕI
CHÚC CÁC BẠN CÓ KẾT QUẢ THI THẬT TỐT!**

 **BAN HỌC TẬP**

Khoa Công nghệ Phần mềm

Trường Đại học Công nghệ Thông tin

Đại học Quốc gia thành phố Hồ Chí Minh

 **CONTACT**

bht.cnpm.uit@gmail.com

fb.com/bhtcnpm

fb.com/groups/bht.cnpm.uit