



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



# NHẬP MÔN MẠNG MÁY TÍNH

**TRAINER**

**Nguyễn Tấn Đạt - 21520703**

**Ngô Võ Quang Minh - 21521129**

**Đỗ Nhật Tân - 21522575**

**Nguyễn Chí Thi - 21522614**



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



**NHẬP MÔN  
MẠNG MT**

**31/10  
1/11/2022  
9h30**



**cơ sở  
DỮ LIỆU**

**02/11/2022  
9h30**



**TỔ CHỨC  
CTMT 2**

**03/11/2022  
9h30**





## Tài Liệu Tham Khảo

*Tài liệu tóm tắt kiến thức của BHT*





# CHƯƠNG I: GIỚI THIỆU



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

### 1. WLAN là viết tắt của khái niệm nào?

- A. Mạng truy cập gia đình
- B. Mạng cục bộ diện rộng
- C. Mạng cục bộ không dây
- D. Mạng không dây diện rộng



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

### 2. Trước khi truyền tin, máy trạm phải được xác định bởi

- A. Một địa chỉ IP hoặc một tên miền
- B. Máy chủ tên miền
- C. Một dãy địa chỉ IP
- D. Vị trí vật lý của máy trạm



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

### 3. Định tuyến là chức năng chính của thiết bị:

- A. Router
- B. Card mạng
- C. Switch
- D. Modem



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

**4. Thứ tự đúng của các tầng từ trên xuống dưới trong mô hình TCP/IP là:**

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



# Chương I: Giới thiệu

**Mô Hình TCP/IP  
(Chòng Giao Thức Internet)**

Application	Data
Transport	Segment
Network	Packet (Datagram)
Data Link	Frames
Physical	Bits

**Mô hình OSI**

Application	Data
Presentation	Data
Session	Data
Transport	Segment
Network	Packet (Datagram)
Link	Frames
Physical	Bits



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

**5. Đơn vị dữ liệu của các lớp theo thứ tự trong mô hình TCP/IP là:**

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



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

### 6. Internet không bao gồm? (nuts and bolts view)

- A. Mạng cung cấp dịch vụ
- B. Mạng lõi
- C. Mạng biên
- D. Mạng truy nhập



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

### 7. Protocol là gì?

- A. Khuôn dạng và thứ tự truyền, nhận thông điệp
- B. Thứ tự truyền, nhận thông điệp
- C. Quy tắc truyền thông
- D. Các hành động tương ứng khi nhận thông điệp



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

### 8. Băng thông là gì?

- A. Tốc độ đến của gói tin
- B. Tốc độ mà tại đó các bit được truyền giữa bên gửi và bên nhận
- C. Số lượng dữ liệu được truyền trong một đơn vị thời gian
- D. Đường truyền mà tại đó giới hạn toàn bộ băng thông của tuyến





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

### 9. Mạng truy cập nào sau đây là hữu tuyến:

- A. Ethernet
- B. WLAN
- C. 3G, 4G
- D. Internet



## Chương I: Giới thiệu

### 10. Đặc điểm nào dưới đây gắn với kỹ thuật chuyển mạch gói? Chọn tất cả các câu trả lời đúng.

- A. Sử dụng phổ biến trên Internet
- B. Dự trữ tài nguyên cần thiết cho cuộc gọi từ nguồn đến đích.
- C. Chia sẻ tài nguyên
- D. Tiết kiệm thời gian hơn trong trường hợp gửi trọng vẹn dữ liệu cỡ lớn
- E. Ghép kênh phân chia theo tần số (FDM) và Ghép kênh phân chia theo thời gian (TDM) là 2 cách tiếp cận để thực hiện kỹ thuật này.
- F. Kỹ thuật này là cơ sở cho việc chuyển đổi cuộc gọi điện thoại trong suốt thế kỷ 20 và đầu thế kỷ này.
- G. Tài nguyên được sử dụng theo yêu cầu, không đặt trước.
- H. Dữ liệu có thể được xếp hàng đợi trước khi truyền do dữ liệu của người dùng khác cũng đang xếp hàng để truyền.
- I. Kỹ thuật này có thể xảy ra tình trạng mất tắc nghẽn và độ trễ đầu cuối thay đổi.





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

**11. Thiết bị nào dưới đây được xem như là phần biên của mạng?**

- A. Modem
- B. Switch
- C. Router
- D. Host



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



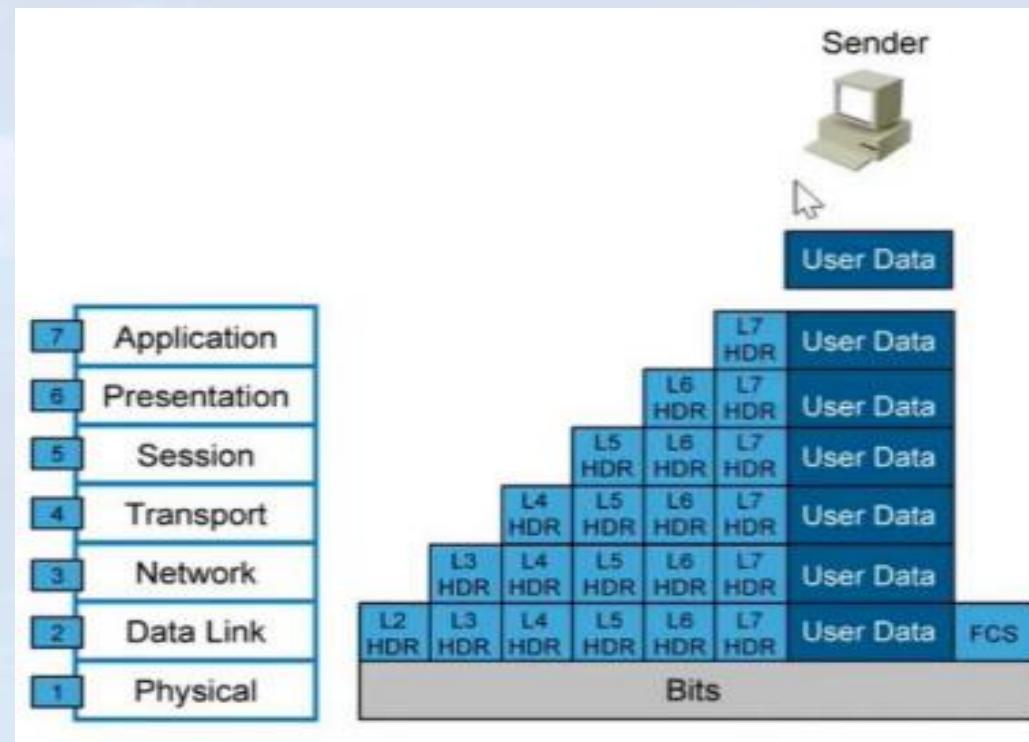
## Chương I: Giới thiệu

**12. Khi dữ liệu di chuyển từ tầng cao xuống tầng thấp hơn (top-down) thì các phần đầu (header) sẽ được?**

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



# Chương I: Giới thiệu



Mỗi gói tin dữ liệu khi được đưa xuống các tầng thì được gắn các header của tầng đó



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

**13. Trong bốn nguồn gây ra chậm trễ gói tin trong quá trình truyền dữ liệu, thì khái niệm “trễ do truyền” (dtrans) là khái niệm nào sau đây?**

- A. Chậm trễ do xử lý tại nút (kiểm tra lỗi bit, xác định cổng ra...)
- B. Chậm trễ trong quá trình lan truyền
- C. Chậm trễ trong quá trình truyền gói tin từ hàng đợi ra đường truyền
- D. Chậm trễ do nằm trong hàng đợi tại nút

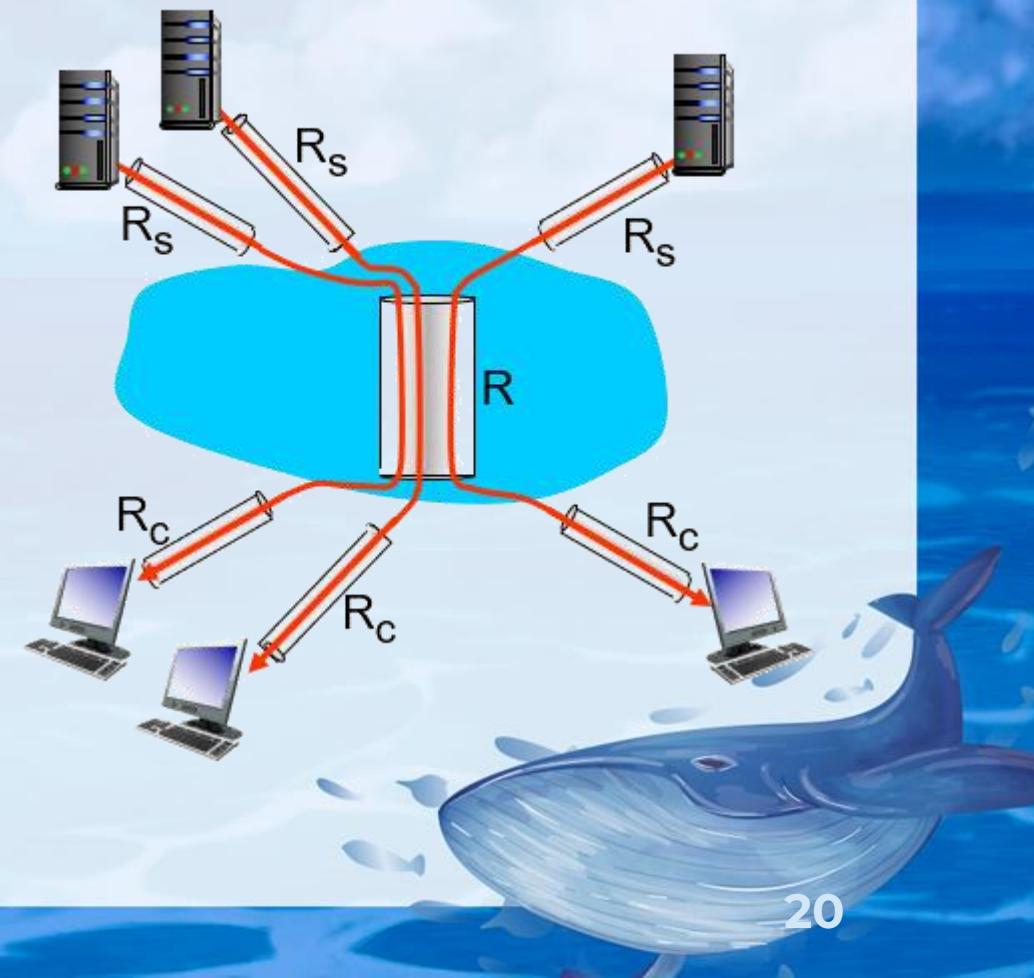




## Chương I: Giới thiệu

**14. Giả sử Host A muốn gửi 1 file có kích thước lớn đến Host B. Từ Host A đến Host B phải đi qua 3 đường liên kết, với tốc độ truyền tương ứng là  $R_s = 600 \text{ Kbps}$ ,  $R = 2 \text{ Mbps}$ ,  $R_c = 1 \text{ Mbps}$ . Thông lượng đường truyền (throughput) từ Host A đến Host B là:**

- A. 2 Mbps
- B. 1 Mbps
- C. 600 Kbps
- D. 1200 Kbps





## Chương I: Giới thiệu

**15. Giả sử Host A muốn gửi 1 file có kích thước 3 triệu bytes đến Host B. Từ Host A đến Host B có 3 đoạn đường truyền nối tiếp nhau, với tốc độ truyền tương ứng là  $R_1 = 1 \text{ Mbps}$ ,  $R_2 = 5 \text{ Mbps}$ ,  $R_3 = 2 \text{ Mbps}$ . Thời gian truyền file đến host B là:**

- A. 3s
- B. 4.8s
- C. 24s
- D. 12s



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

**16. Cần truyền gói tin kích thước 1KB từ Host A đến Host B, trên đường truyền dài 2500 km, tốc độ lan truyền  $2.5 \times 10^8$  m/s, và tốc độ truyền 2 Mbps. Giả sử rằng thời gian xử lý và thời gian chờ tại hàng đợi không đáng kể. Thời gian cần để truyền gói tin từ host A đến Host B là:**

- A. 10ms
- B. 10.5ms

- C. 14ms
- D. 14s



## Chương I: Giới thiệu

17. Hai máy tính A và B kết nối với nhau qua một đường truyền có tốc độ R bps, và khoảng cách là m (mét). Tốc độ lan truyền của tín hiệu trên đường truyền là s (m/s). Máy A gửi 1 gói tin có kích thước L bits đến máy B. Cho  $s = 2.5 \times 10^8$  (m/s),  $L = 100$  bits,  $R = 28$  Kbps. Hãy xác định khoảng cách m để thời gian truyền gói tin có kích thước L (transmission time) bằng với thời gian lan truyền tín hiệu (propagation delay) từ máy A đến máy B.

- A. 714 km
- B. 893 km

- C. 2500 km
- D. 714 m



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

**18. Các gói tin có độ dài  $L = 1000$  bytes được truyền trên một kết nối có tốc độ truyền là  $R = 1000$  Kbps. Hỏi tối đa có bao nhiêu gói tin được truyền trong 1s? (Chỉ điền số vào khung đáp án)**

→ Đáp án: 125



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương I: Giới thiệu

**19. Tính thời gian, theo s, để truyền 1 file có kích thước 5 (MB), từ Host A đến Host B trên mạng Circuit Switching. Biết rằng:**

- Đường truyền có tổng băng thông là 8 Mbps.**
  - Đường truyền sử dụng TDM với 14 (slot/s)**
  - Thời gian thiết lập mạch nối giữa 2 điểm A và B là 0.5s**
- Đáp án: 70.5



## CHƯƠNG II: TẦNG APPLICATION



UIT  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Tài Liệu Tham Khảo Chương II

*Tài liệu tóm tắt kiến thức của BHT*





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



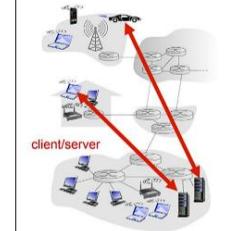
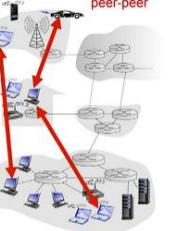
## Chương II: Tầng Application

**1. Xét theo chức năng của các máy trong mạng có thể phân mạng máy tính thành những mô hình chủ yếu nào?**

- Mô hình ngang hàng (P2P) và mô hình khách chủ (Client – Server)
- Mô hình hàng ngang (P2P)
- Mô hình khách – chủ
- Tất cả đều sai



# Tài Liệu Tham Khảo

Các kiến trúc ứng dụng		
	Client – Server	P2P (Peer to peer)
<b>Mình Hợp</b>		
<b>Server</b>	Luôn hoạt động Địa chỉ IP cố định. Trung tâm phục vụ và lưu trữ dữ liệu.	Không có Server luôn hoạt động
<b>Client</b>	Giao tiếp với Server. Không giao tiếp trực tiếp với các Client khác.  Có thể kết nối không liên tục, dùng địa chỉ IP động.  Clients gửi request và nhận response từ Server. Giao tiếp với Clients khác qua Server	Các hệ thống đầu cuối giao tiếp trực tiếp với nhau.  Các Peer được kết nối không liên tục và có thể thay đổi địa chỉ IP. ⇒ Quản lý phức tạp  Các Peer vừa có thể tiêu thụ và cung cấp tài nguyên cho các peer khác ⇒ Khả năng mở rộng tốt

*Tài liệu tóm tắt kiến thức của BHT  
(Chương 2, tuần 5, trang 2)*



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

**2. Trong các mô hình sau, mô hình nào là mô hình mạng được dùng phổ biến hiện nay:**

- Peer - to – Peer
- Remote Access
- Terminal – Mainframe
- Client – Server



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### 3. Chọn câu trả lời đúng nhất

- Trong mô hình client/server chỉ có duy nhất một máy chủ
- Trong mô hình P2P, một máy tính đóng vai trò là máy chủ khi cung cấp tài nguyên cho các máy khác và đóng vai trò là máy khách khi sử dụng tài nguyên do máy khác cung cấp
- Mô hình P2P tốt hơn mô hình client/server vì mỗi máy trong mô hình đó vừa là máy chủ, vừa là máy khách
- Mỗi máy trong P2P đều được bảo mật, nên toàn bộ mạng được bảo mật tốt hơn.



## Chương II: Tầng Application

### 4. Tính chất nào sau đây không được cung cấp bởi TCP service:

- Truyền tin cậy (Reliable transmission)
- Điều khiển dòng (Flow control)
- Bảo đảm hiệu suất tối thiểu (Minimum throughput guarantee)
- Điều khiển tắc nghẽn (Congestion control)



# Tài Liệu Tham Khảo



- Là giao thức điều khiển truyền nhận. Giao thức này đảm bảo sự chuyển giao thông tin từ nơi nguồn tới nơi nhận một cách an toàn, bảo đảm toàn vẹn và đúng thứ tự.
- TCP có thể điều tiết được người gửi khi mạng quá tải và thiết lập yêu cầu giữa tiến trình client và server.

**Dịch vụ UDP:**

- Là một giao thức cho phép gửi các gói tin mà không cần dùng một kết nối hoàn chỉnh giữa các thiết bị kết nối mạng
- Không hỗ trợ độ tin cậy, điều khiển luồng, điều khiển tắt nghẽn, bảo đảm thông lượng, bảo mật, và thiết lập kết nối.

**So sánh TCP – UDP:**

**Giêng:** TCP và UDP đều là các giao thức được sử dụng để gửi các bit dữ liệu – được gọi là các gói tin – qua Internet. Cả hai giao thức đều được xây dựng trên giao thức IP.

Khác:

	<b>TCP</b>	<b>UDP</b>
Thiết Kế	Định hướng kết nối (connection oriented).	Kêm kết nối hơn (connectionless).
Độ tin cậy	Độ tin cậy cao.	Độ tin cậy thấp.
Truyền dữ liệu	Gửi gói tin dạng luồng, xử lý theo thứ tự gói tin.	Dữ liệu không được truyền theo thứ tự.
Hiệu năng	Tốc độ truyền thấp hơn.	Tốc độ truyền cao hơn.
Xử lý vấn đề về gói tin	Gói tin có thể được gửi lại nếu cần hoặc bị mất.	Gói tin không được truyền lại

**Web và HTTP**

HTTP (Hypertext Transfer Protocol) là giao thức web ở tầng Application.

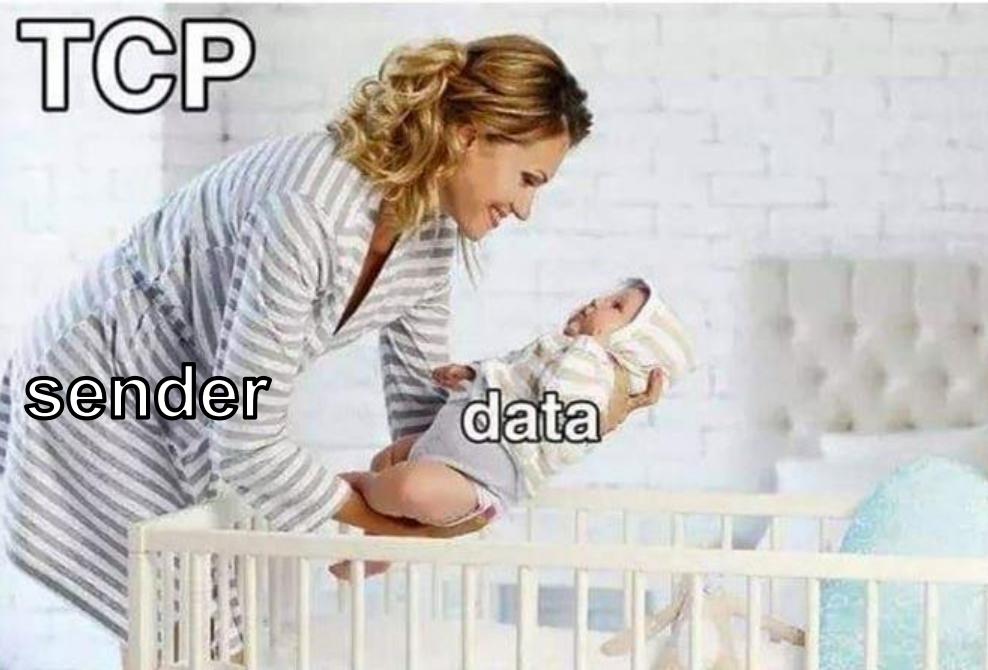
Mô hình client/server:

- Client: trình duyệt yêu cầu và nhận (sử dụng giao thức HTTP) ⇒ Hiển thị các Object của Website.

*Các dịch vụ giao thức Transport Internet  
(Chương 2, tuần 5, trang 5)*



## Phân biệt TCP và UDP





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

**5. Giao thức nào dưới đây không đảm bảo dữ liệu gửi đi có tới máy nhận hoàn chỉnh không?**

- UDP
- ARP
- TCP
- RARP



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### 6. TCP và UDP là giao thức của tầng nào?

- Tầng Ứng dụng
- Tầng Vận chuyển
- Tầng Mạng
- Tầng Trệt



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

**7. Đơn vị dữ liệu tương ứng với tầng Application là:**

- Segments
- Data
- Bits
- Frames



## Chương II: Tầng Application

Mô Hình TCP/IP (Chòng Giao Thức Internet)	
Application	Data
Transport	Segment
Network	Packet (Datagram)
Data Link	Frames
Physical	Bits

Mô hình OSI	
Application	Data
Presentation	Data
Session	Data
Transport	Segment
Network	Packet (Datagram)
Link	Frames
Physical	Bits



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

**8. Các đơn vị dữ liệu giao thức trong mô hình OSI được gọi là:**

- PDU (Protocol Data Unit)
  - Packet
  - CSU
  - Frame



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### 9. HTTP sử dụng Port nào

- 53
- 80
- 21
- 443



# Tài Liệu Tham Khảo



- Là giao thức điều khiển truyền nhận. Giao thức này đảm bảo sự chuyển giao thông tin từ nơi nguồn tới nơi nhận một cách an toàn, bảo đảm toàn vẹn và đúng thứ tự.
- TCP có thể điều tiết được người gửi khi mạng quá tải và thiết lập yêu cầu giữa tiến trình client và server.

**Dịch vụ UDP:**

- Là một giao thức cho phép gửi các gói tin mà không cần dùng một kết nối hoàn chỉnh giữa các thiết bị kết nối mạng
- Không hỗ trợ độ tin cậy, điều khiển luồng, điều khiển tắt nghẽn, bảo đảm thông lượng, bảo mật, và thiết lập kết nối.

**So sánh TCP – UDP:**

**Giống:** TCP và UDP đều là các giao thức được sử dụng để gửi các bit dữ liệu – được gọi là các gói tin – qua Internet. Cả hai giao thức đều được xây dựng trên giao thức IP.

**Khác:**

	<b>TCP</b>	<b>UDP</b>
Thiết Kế	Định hướng kết nối (connection oriented).	Kém kết nối hơn (connectionless).
Độ tin cậy	Độ tin cậy cao.	Độ tin cậy thấp.
Truyền dữ liệu	Gửi gói tin dạng luồng, xử lý theo thứ tự gói tin.	Dữ liệu không được truyền theo thứ tự.
Hiệu năng	Tốc độ truyền thấp hơn.	Tốc độ truyền cao hơn.
Xử lý vấn đề về gói tin	Gói tin có thể được gửi lại nếu cần hoặc bị mất.	Gói tin không được truyền lại

**Web và HTTP**

HTTP (Hypertext Transfer Protocol) là giao thức web ở tầng Application.

Mô hình client/server:

- **Client:** trình duyệt yêu cầu và nhận (sử dụng giao thức HTTP) ⇒ Hiển thị các Object của Website.

*Các tiến trình liên lạc  
(Chương 2, tuần 5, trang 3)*



## Tài Liệu Tham Khảo

Protocol	Port Number	Protocol	Port Number
FTP	20	DNS	53
HTTP	80	SSH	22
HTTPS	443	TFTP	69
SMTP	25	Telnet	23
LPD	515	NFS	2049



## Chương II: Tầng Application

### 10. Nhiệm vụ của giao thức HTTP là gì?

- Hiển thị các trang Web từ xa trên màn hình và giúp người dùng tương tác với chúng
- Cung cấp dữ liệu từ server sử dụng giao thức truyền file (File Transport Protocol)
- Cung cấp một cơ chế để lấy dữ liệu từ server chuyển đến client
- Cung cấp giao diện người dùng như các nút bấm, thanh trượt



# Tài Liệu Tham Khảo



- Là giao thức điều khiển truyền nhận. Giao thức này đảm bảo sự chuyển giao thông tin từ nơi nguồn tới nơi nhận một cách an toàn, bảo đảm toàn vẹn và đúng thứ tự.
- TCP có thể điều tiết được người gửi khi mạng quá tải và thiết lập yêu cầu giữa tiến trình client và server.

**Dịch vụ UDP:**

- Là một giao thức cho phép gửi các gói tin mà không cần dùng một kết nối hoàn chỉnh giữa các thiết bị kết nối mạng
- Không hỗ trợ độ tin cậy, điều khiển luồng, điều khiển tắt nghẽn, bảo đảm thông lượng, bảo mật, và thiết lập kết nối.

**So sánh TCP – UDP:**

**Giống:** TCP và UDP đều là các giao thức được sử dụng để gửi các bit dữ liệu – được gọi là các gói tin – qua Internet. Cả hai giao thức đều được xây dựng trên giao thức IP.

**Khác:**

	<b>TCP</b>	<b>UDP</b>
Thiết Kế	Định hướng kết nối (connection oriented).	Kém kết nối hơn (connectionless).
Độ tin cậy	Độ tin cậy cao.	Độ tin cậy thấp.
Truyền dữ liệu	Gửi gói tin dạng luồng, xử lý theo thứ tự gói tin.	Dữ liệu không được truyền theo thứ tự.
Hiệu năng	Tốc độ truyền thấp hơn.	Tốc độ truyền cao hơn.
Xử lý vấn đề về gói tin	Gói tin có thể được gửi lại nếu cần hoặc bị mất.	Gói tin không được truyền lại

**Web và HTTP**

HTTP (Hypertext Transfer Protocol) là giao thức web ở tầng Application.

Mô hình client/server:

- **Client:** trình duyệt yêu cầu và nhận (sử dụng giao thức HTTP) ⇒ Hiển thị các Object của Website.

## Web và HTTP

(Chương 2, tuần 5, trang 5)





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### 11. Hãy chọn phát biểu sai

- HTTP 1.0 sử dụng HTTP không bền vững, HTTP 1.1 sử dụng HTTP bền vững
- HTTP là giao thức phi trạng thái (stateless)
- HTTP 1.0 sử dụng HTTP bền vững, HTTP 1.1 sử dụng HTTP không bền vững
- HTTP 1.0 sử dụng HTTP không bền vững, HTTP 1.1 sử dụng HTTP bền vững. HTTP là giao thức phi trạng thái (stateless)



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

**12. HTTP không bền vững sử dụng các phương thức nào sau đây?**

- GET, POST, HEAD
  - PUT, GET, HEAD, POST
  - GET, DELETE, PUT
  - POST, HEAD, DELETE, PUT



## Chương II: Tầng Application

### 13. Cho nội dung của một gói tin HTTP GET được gửi từ một trình duyệt như sau:

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gaia.cs.umass.edu<cr><lf>User-Agent:  
Mozilla/5.0 (Windows;U; Windows NT 6.1; en-US; rv:1.7.2) Gecko/20040804  
Netscape/7.2 (ax)<cr><lf> Accept:text/xml, application/xml, application/xhtml+xml,  
text/html;q=0.9, text/plain;q=0.8, image/png,*/*;q=0.5<cr><lf>Accept-Language: en-  
us,en; q=0.5<cr><lf> Accept-Encoding: zip,deflate <cr><lf>Accept-Charset: ISO-8859-  
1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive:300<cr><lf>Connection:keep-  
alive<cr><lf><cr><lf>
```

### URL của file mà trình duyệt yêu cầu là gì?

- <http://gaia.cs.umass.edu/cs453/index.html>  
Gecko/20040804Netscape/7.2
- <http://gaia.cs.umass.edu/Gecko/20040804Netscape/7.2/cs453/index.html>
- <http://gaia.cs.umass.edu/cs453/index.html>

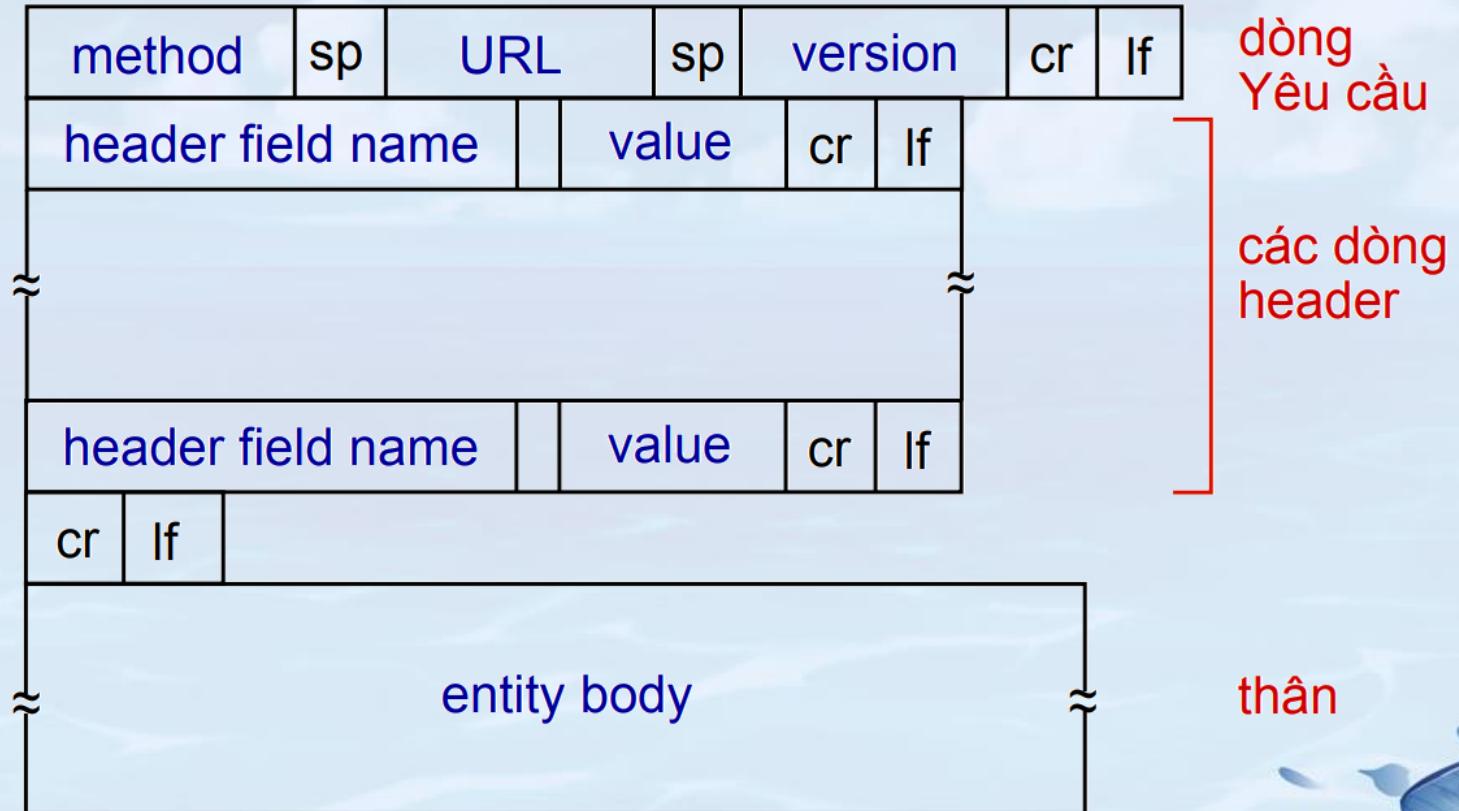


UIT  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin

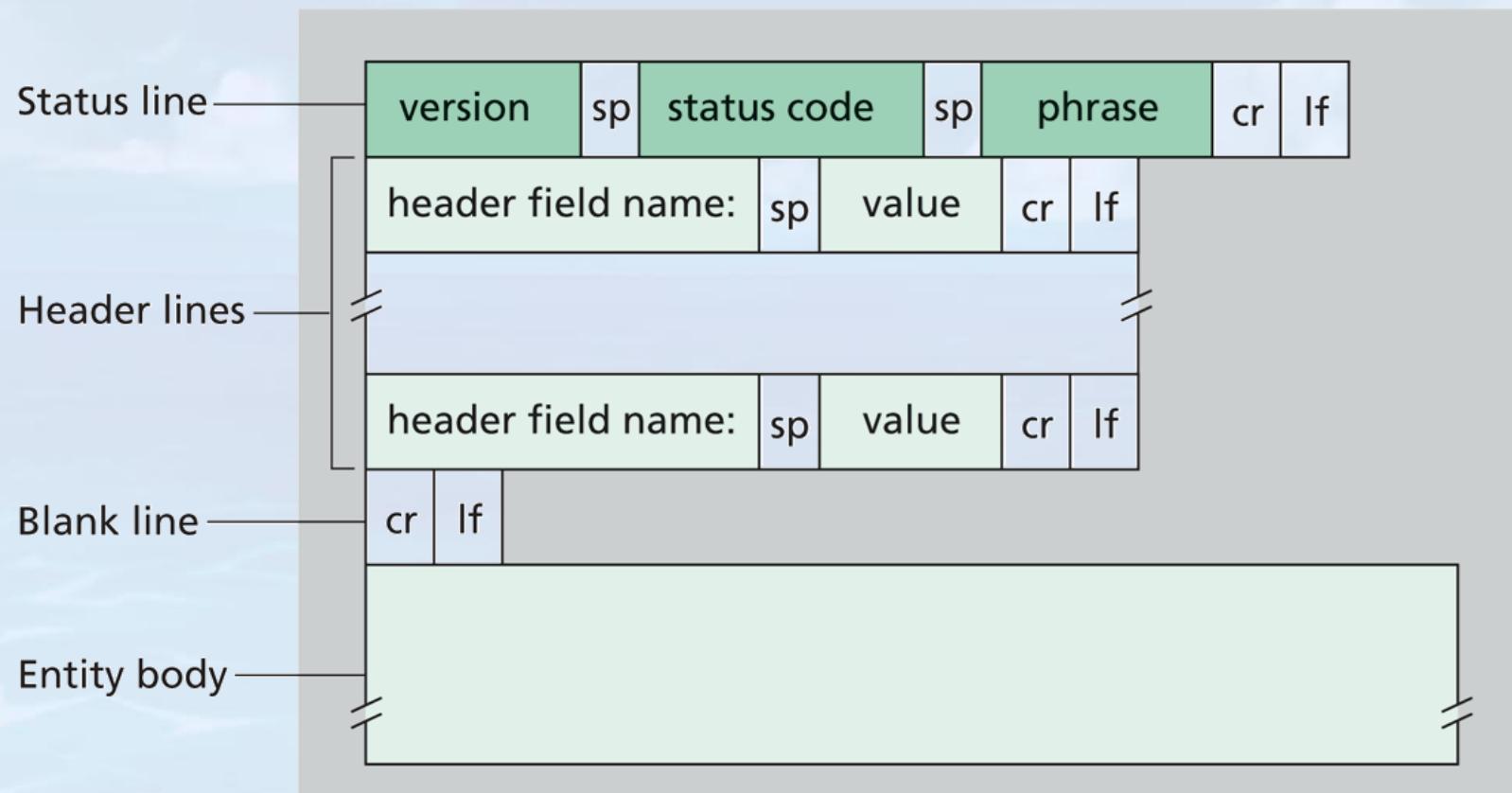


## Thông điệp yêu cầu (Request Message)





# Thông điệp phản hồi (Response Message)





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Thông điệp phản hồi (Response Message)

dòng trạng thái  
(giao thức)

mã trạng thái

cụm từ trạng thái

các dòng  
header

Dữ liệu, ví dụ,

file HTML

được yêu cầu

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT
\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html;
    charset=ISO-8859-1\r\n
\r\n
data data data data data ...
```



## Chương II: Tầng Application

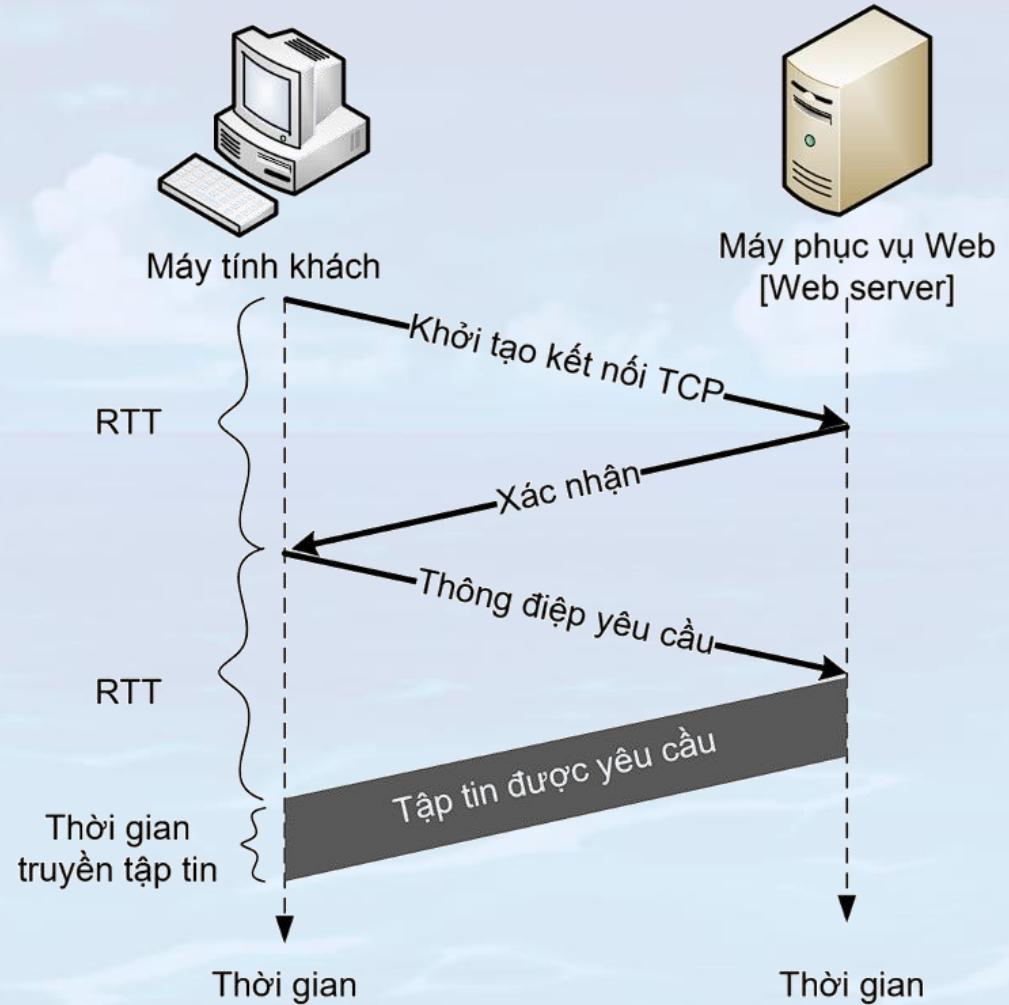
**14. Giả sử trong trình duyệt web, bạn nhấn vào một link chứa một trang web. Bạn cần lấy một địa chỉ IP của trang web đó chưa có trong cache. Bạn phải đi qua n DNS sau đó mới có được thông tin đó. Mỗi lần truy vấn một DNS phải mất RTT thời gian lần lượt là RTT<sub>1</sub>, RTT<sub>2</sub>... RTT<sub>n</sub>. Trong trang web đó có một đối tượng text. Bạn mất RTT<sub>0</sub> để đi từ host tới server chứa đối tượng đó. Tính thời gian từ khi nhấn vào link đến khi nhận được đối tượng**

- RTT<sub>0</sub> + RTT<sub>1</sub> + RTT<sub>2</sub> + ... + RTT<sub>n</sub>
- 2\*(RTT<sub>0</sub> + RTT<sub>1</sub> + ... + RTT<sub>n</sub>)
- 2\*RTT<sub>0</sub> + RTT<sub>1</sub> + ... + RTT<sub>n</sub>
- 2\*RTT<sub>0</sub> + RTT<sub>1</sub> + ... + 2.RTT<sub>n</sub>



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin





**UIT**  
Trường Đại học  
Công nghệ Thông tin

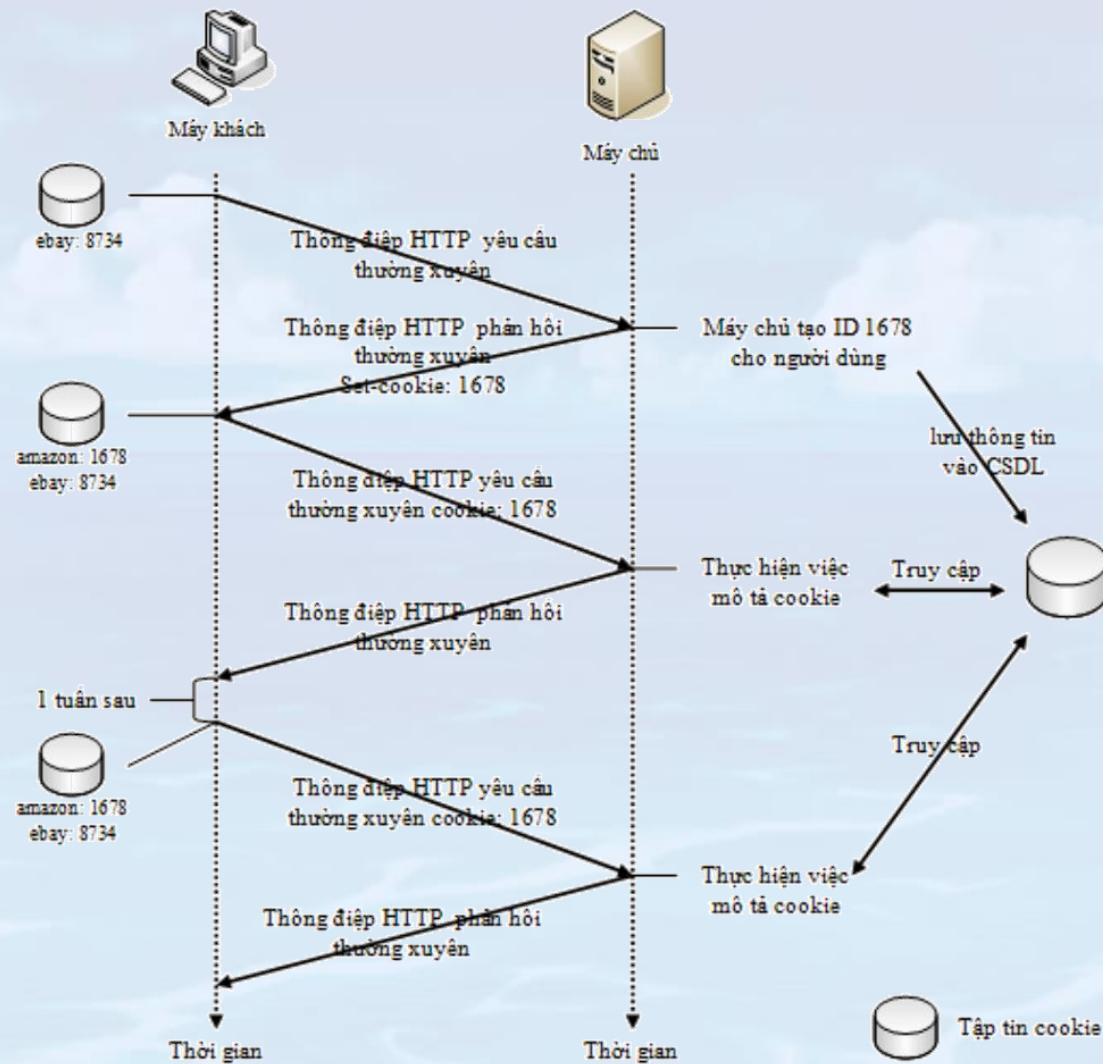
Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### 15. Cookies có thể được dùng cho:

- Lưu trạng thái phiên làm việc của User
- Lưu bản sao của các trang web User truy cập
- Kết nối HTTP bền vững
- Lựa chọn khác





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

**16. Để giảm lưu lượng trên đường liên kết truy cập ra Internet của một tổ chức, người ta thường dùng:**

- Cookies
- Load Balancing
- Web Caching
- Proxy



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

17. Client gửi một thông điệp HTTP request đến server với trường IF-MODIFIED-SINCE trong header. Giả sử đối tượng đó trong server không có sự thay đổi nào kể từ lần cuối cùng client lấy đối tượng từ server, thì client sẽ nhận lại được thông điệp HTTP response có status code (mã trạng thái) là gì?

- 404
- 200
- 304
- 301



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### Các mã trạng thái đáp ứng HTTP (HTTP response status codes)

Xuất hiện trong dòng đầu tiên trong thông điệp đáp ứng từ Server.

Một vài mã trạng thái thường gặp:

- [200 OK](#) – Yêu cầu thành công, đối tượng được yêu cầu 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.
- [304 Not Modified](#) – Đối tượng được yêu cầu chưa được điều chỉnh (và đã lưu trong cache), client có thể sử dụng đối tượng được phản hồi trong bộ nhớ cache.
- [400 Bad Request](#) – Thông điệp yêu cầu không được hiểu bởi Server.
- [404 Not Found](#) – Tài liệu được yêu cầu không tìm thấy trên server này.

# 502 Bad Gateway

---

nginx/1.20.1



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### 18. Dịch vụ mạng SMTP dùng để

- Gửi thư điện tử
  - Nhận thư điện tử
  - Phân giải tên và địa chỉ
  - Lấy địa chỉ IP từ địa chỉ MAC



# Tài Liệu Tham Khảo



- Là giao thức điều khiển truyền nhận. Giao thức này đảm bảo sự chuyển giao thông tin từ nơi nguồn tới nơi nhận một cách an toàn, bảo đảm toàn vẹn và đúng thứ tự.
- TCP có thể điều tiết được người gửi khi mạng quá tải và thiết lập yêu cầu giữa tiến trình client và server.

**Dịch vụ UDP:**

- Là một giao thức cho phép gửi các gói tin mà không cần dùng một kết nối hoàn chỉnh giữa các thiết bị kết nối mạng
- Không hỗ trợ độ tin cậy, điều khiển luồng, điều khiển tắt nghẽn, bảo đảm thông lượng, bảo mật, và thiết lập kết nối.

**So sánh TCP – UDP:**

**Giống:** TCP và UDP đều là các giao thức được sử dụng để gửi các bit dữ liệu – được gọi là các gói tin – qua Internet. Cả hai giao thức đều được xây dựng trên giao thức IP.

**Khác:**

	<b>TCP</b>	<b>UDP</b>
Thiết Kế	Định hướng kết nối (connection oriented).	Kém kết nối hơn (connectionless).
Độ tin cậy	Độ tin cậy cao.	Độ tin cậy thấp.
Truyền dữ liệu	Gửi gói tin dạng luồng, xử lý theo thứ tự gói tin.	Dữ liệu không được truyền theo thứ tự.
Hiệu năng	Tốc độ truyền thấp hơn.	Tốc độ truyền cao hơn.
Xử lý vấn đề về gói tin	Gói tin có thể được gửi lại nếu cần hoặc bị mất.	Gói tin không được truyền lại

**Web và HTTP**

HTTP (Hypertext Transfer Protocol) là giao thức web ở tầng Application.

Mô hình client/server:

- **Client:** trình duyệt yêu cầu và nhận (sử dụng giao thức HTTP) ⇒ Hiển thị các Object của Website.

*Email*

*(Chương 2, tuần 5, trang 8)*



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

**19. Trong quá trình truyền thông điệp thư điện tử với giao thức SMPT, thông điệp phải ở dạng nào?**

- ASCII 8 bit
- ASCII 7 bit
- Nhị phân
- Tất cả đều sai



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

**19. Cho một phiên làm việc của SMTP, hãy sắp xếp trình tự giao tiếp phía client cho đúng.**

- HELLO, DATA, MAIL FROM, QUIT, RCPT TO
- HELLO, MAIL FROM, RCPT TO, DATA, QUIT
- HELLO, RCPT TO, DATA, QUIT, MAIL FROM
- HELLO, MAIL FROM, DATA, RCPT TO, QUIT



## Chương II: Tầng Application

**20. Trong các dịch vụ sau, đâu KHÔNG PHẢI là dịch vụ do DNS cung cấp?**

- Bí danh Mail Server
- Định nghĩa định dạng, thứ tự các thông điệp được gửi và nhận
- Dịch tên Host ra địa chỉ IP
- Phân phối tải



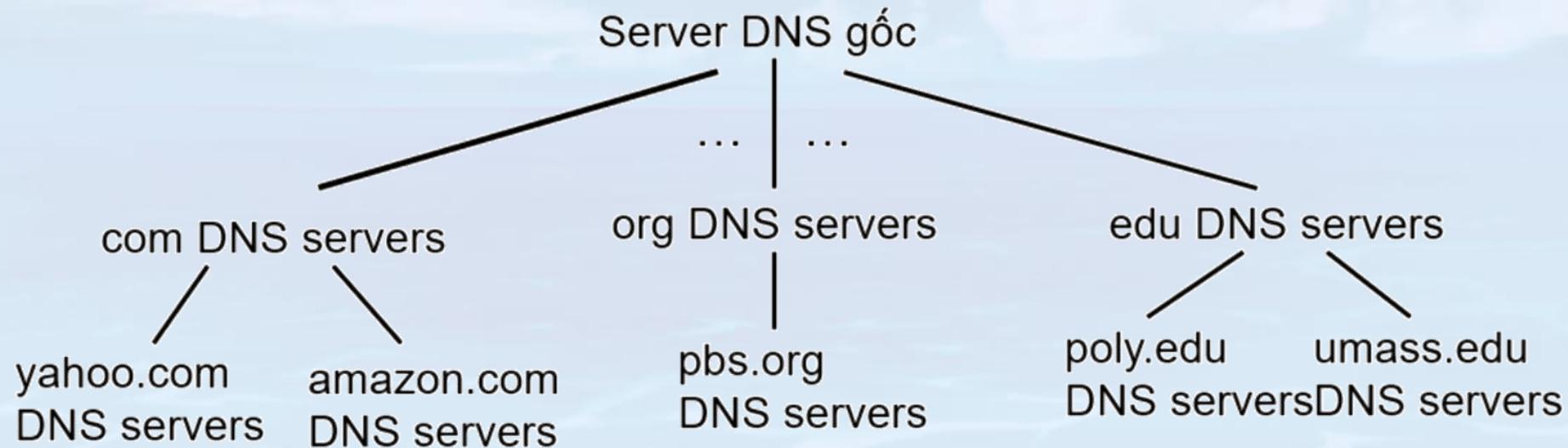
**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

DNS là một cơ sở dữ liệu phân tán



### Các máy chủ tên miền gốc (Root DNS server)

- Trên thế giới hiện có 13 root DNS server
- Quản lý tất cả các tên miền top level

### Các top-level domain server (TLD servers):

- Chịu trách nhiệm cho tên miền com, org, net, edu, aero, jobs, museums, và tất cả các tên miền cấp cao nhất của quốc gia, như là: uk, fr, ca, jp
- TLD server quản lý máy chủ chứa các thông tin của vùng mình quản lý.

### Các DNS máy chủ có thẩm quyền (Authoritative DNS server):

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

### DNS server cục bộ (Local DNS server)

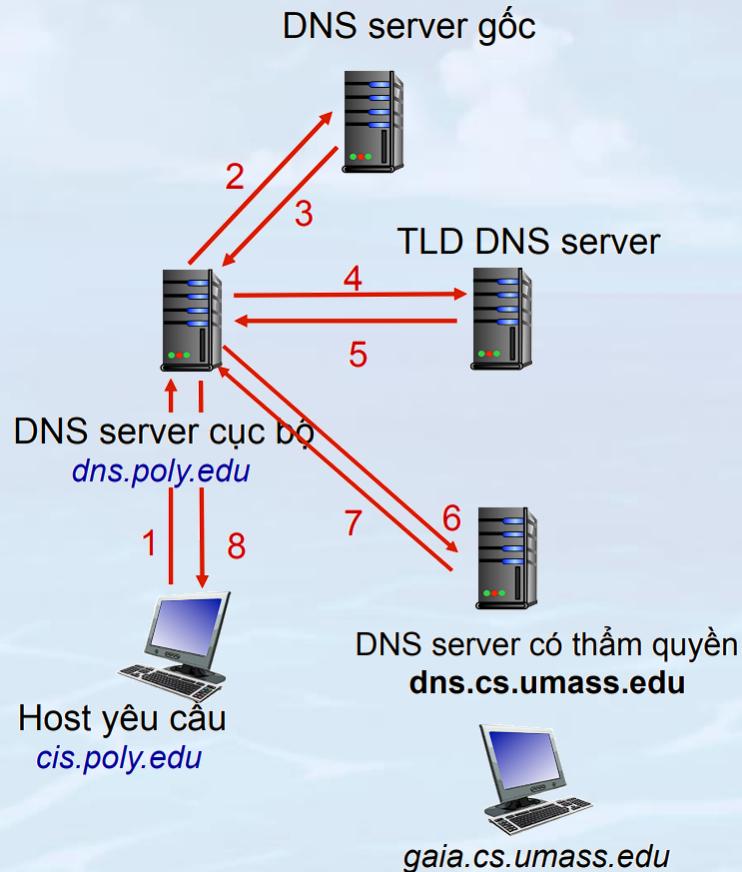


**UIT**  
Trường Đại học  
Công nghệ Thông tin

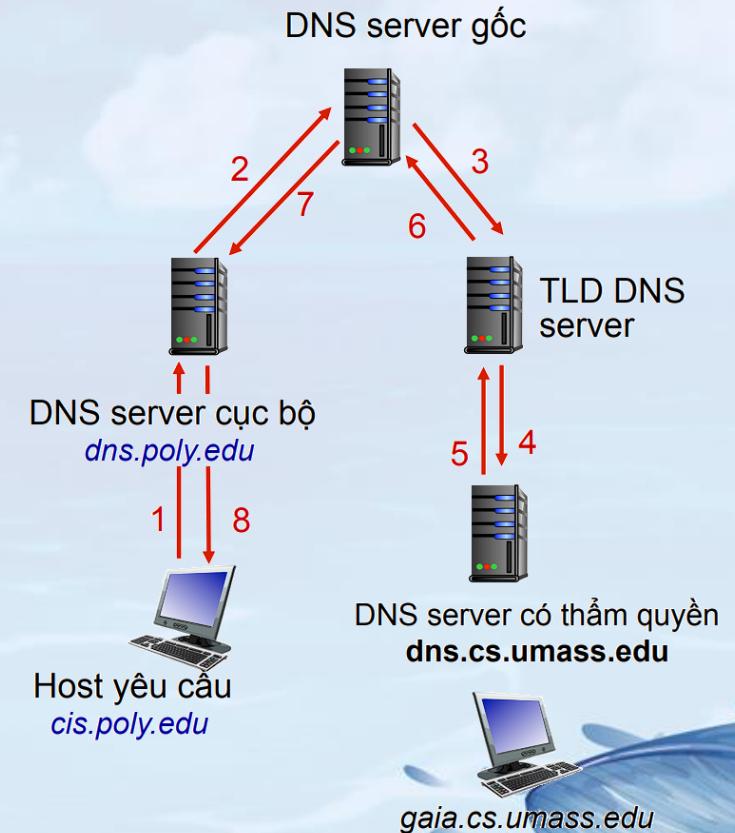
Khoa Khoa học  
và Kỹ thuật Thông tin



## Truy vấn lặp



## Truy vấn đệ quy





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### 21. Có bao nhiêu cách phân giải tên miền DNS?

- 1
- 2
- 3
- 4



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương II: Tầng Application

### 21. Resource record (RR) trong DNS với type là MX dùng để làm gì?

- Định chuyển gói tin
- Dùng cho LAN backup
- Dùng cho dịch vụ FTP
- Dùng cho dịch vụ chuyển mail



## Các DNS record

**DNS:** cơ sở dữ liệu phân tán lưu trữ các record tài nguyên (RR)

Định dạng RR: (name, value, type, ttl)

### type=A

- **Name** là tên host
- **value** là địa chỉ IP

### type=NS

- **Name** là tên miền (e.g., foo.com)
- **Value** là tên host của name server có thẩm quyền cho tên miền này

### type=CNAME

- **Name** là bí danh của một số tên “chuẩn” (tên thực)  
`www.ibm.com` is really  
`servereast.backup2.ibm.com`
- **Value** là tên chuẩn (tên thật)

### type=MX

- **value** là tên của mail server được liên kết với **name**

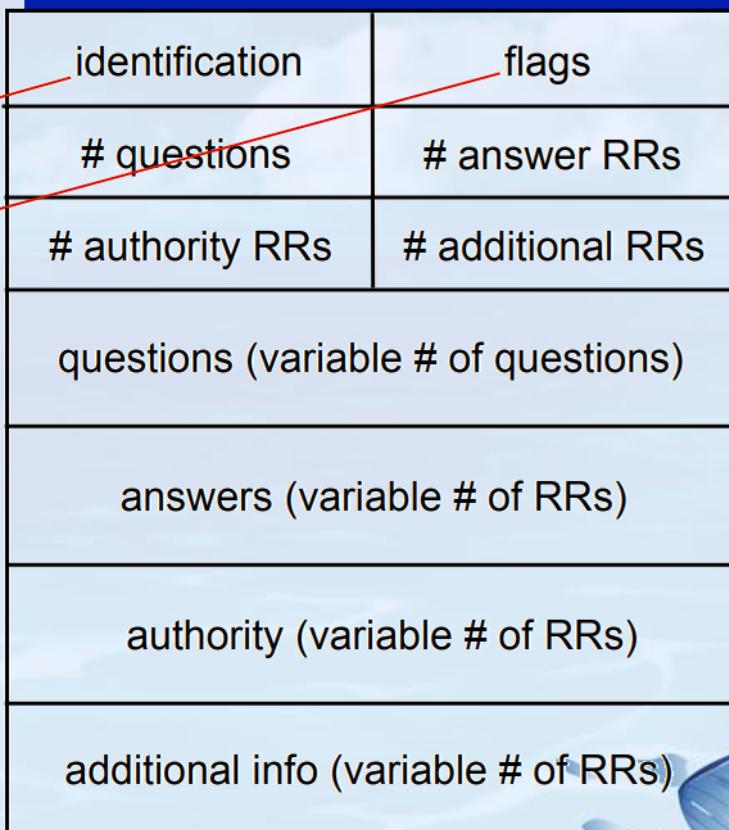


- ❖ Các thông điệp truy vấn (query) và trả lời (reply), đều có cùng định dạng thông điệp

↔ 2 bytes ↔ 2 bytes

### msg header

- ❖ identification: 16 bit # cho truy vấn, trả lời cho truy vấn dùng cùng #
- ❖ flags:
  - Truy vấn hoặc trả lời
  - Đệ quy mong muốn
  - Đệ quy sẵn sàng
  - Trả lời có thẩm quyền





Các trường name,  
type cho một truy vấn

Các RRs để trả  
lời truy vấn

Các record cho các  
server có thẩm quyền

thông tin “hữu ích”  
bổ sung có thể sẽ dùng

↔ 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)	



## Chương II: Tầng Application

**Cho đoạn code socket như bên,  
Đây là lập trình socket ứng dụng gì**

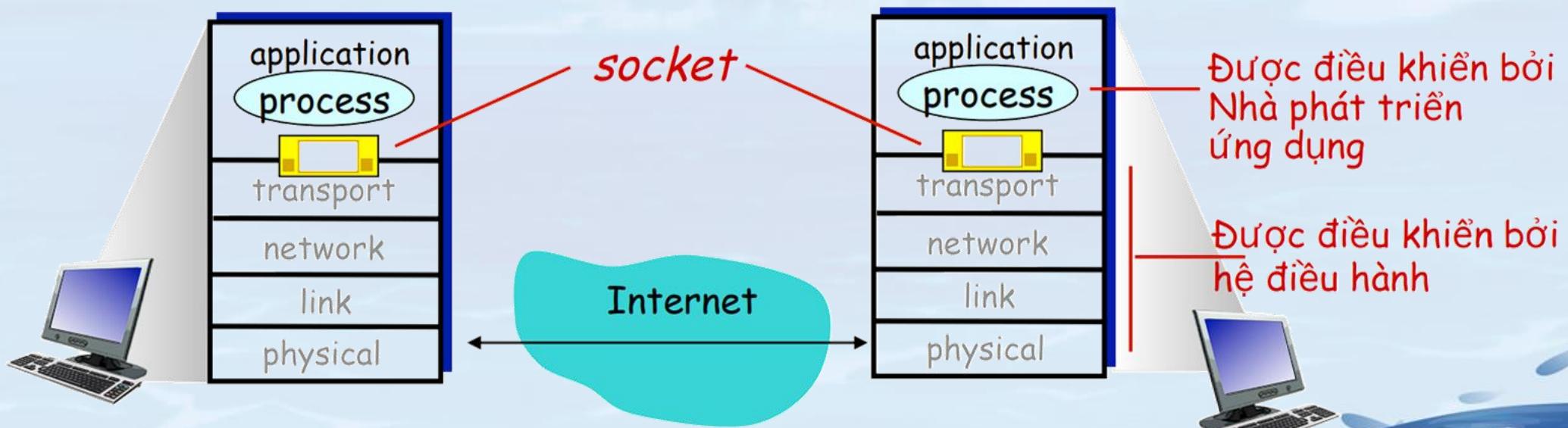
- Ứng dụng TCP Socket phía client
- Ứng dụng TCP Socket phía server
- Ứng dụng UDP Socket phía client
- Ứng dụng UDP Socket phía server

```
from socket import *
serverName = 'hostname'
serverPort = 12000
clientSocket = socket(socket.AF_INET,
                      socket.SOCK_DGRAM)
message = raw_input('Input lowercase sentence:')
clientSocket.sendto(message, (serverName,
                               serverPort))
modifiedMessage, serverAddress =
    clientSocket.recvfrom(2048)
print modifiedMessage
clientSocket.close()
```



## Lập trình Socket với UDP và TCP

xây dựng các ứng dụng client/server





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Lập trình Socket với UDP và TCP

### Phía Client

```
from socket import *
serverName = 'servername'
serverPort = 12000
clientSocket = socket(socket.AF_INET, socket.SOCK_DGRAM)
message = raw_input('Input lowercase sentence:')
clientSocket.sendto(message, (serverName, serverPort))
modifiedMsg, serverAddress = clientSocket.recvfrom(2048)
print modifiedMsg
clientSocket.close()
```



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Lập trình Socket với UDP và TCP

### (UDP) Phía Server

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', serverPort))
print "The server is ready to receive"
while 1:
    message, clientAddress = serverSocket.recvfrom(2048)
    modifiedMessage = message.upper()
    serverSocket.sendto(modifiedMessage, clientAddress)
```



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Lập trình Socket với UDP và TCP

### (TCP) Phía Server

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('', serverPort))
serverSocket.listen(1)
print 'The server is ready to receive'
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence)
    connectionSocket.close()
```



# CHƯƠNG III: TẦNG TRANSPORT



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



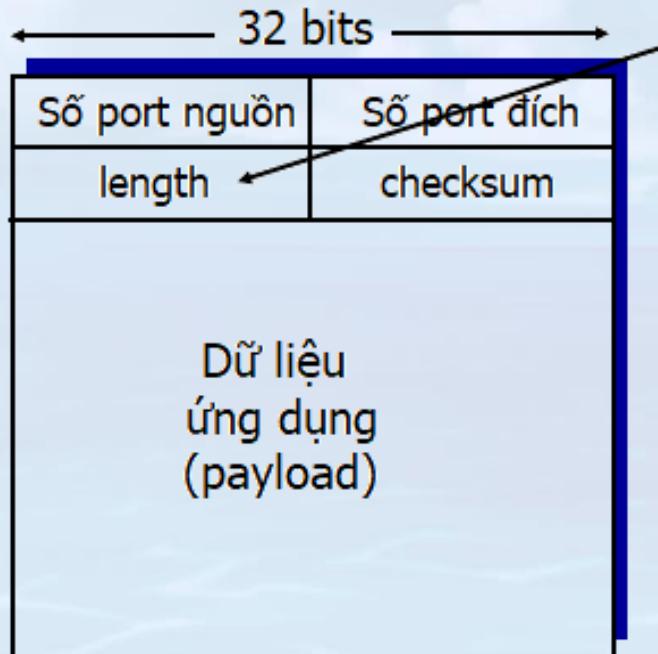
## Chương III: Tầng Transport

### 1. Trong các định dạng header của segment UDP có 4 field:

- Source port, destination port, length, checksum
- Source port, destination port, head length, checksum
- Source port, destination port, sequence number, acknowledgement number
- Source port, destination port, sequence number, receive window



## Chương III: Tầng Transport



Định dạng segment UDP

- + Tổng kích thước header: 8 bytes
- + Length: Tính cả dung lượng header lẫn payload, checksum cũng vậy
- + Cách tính checksum: Tính tổng bù 1 các đoạn 16 bits trong segment
- + Port: 0 – 1023 là cổng phổ biến (well-known port)



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 2. Trong cấu trúc header của TCP có 6 cờ là:

- SYN, DAT, PSH, RST, FIN, URG
- CON, ACK, PSH, RST, FIN, URG
- SYN, ACK, PSH, RST, FIN, URG
- SYN, ACK, PSH, DAT, CON, URG



## Cấu trúc TCP segment

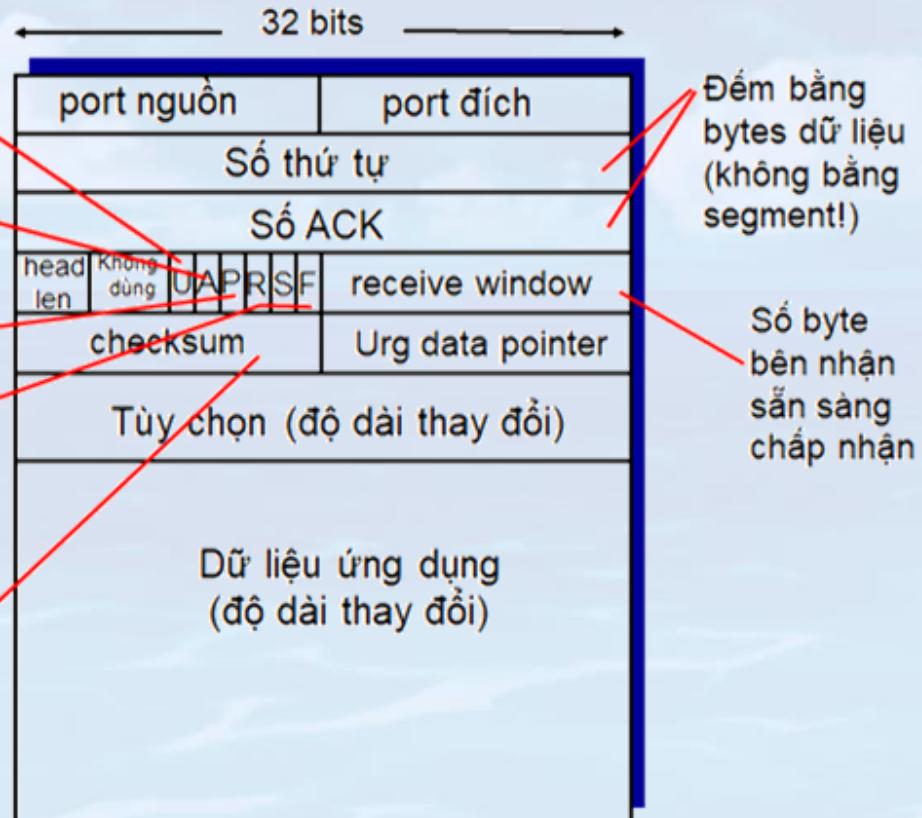
URG: dữ liệu khẩn cấp  
(thường không dùng)

ACK: ACK #  
hợp lệ

PSH: push data now  
(thường không dùng)

RST, SYN, FIN:  
thiết lập kết nối  
(setup, teardown  
commands)

Internet  
checksum  
(giống như UDP)



Kích thước Header:  
20 bytes  
Checksum: Tính  
bằng toàn bộ  
segment TCP



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

Ví dụ, giả sử rằng một tiến trình P1 trong máy A muốn gửi tập tin có kích thước 500000 bytes đến tiến trình P2 trong máy B thông qua một kết nối TCP. Và giả sử rằng giá trị MSS của kết nối TCP là 1000 bytes và bytes đầu tiên trong chuỗi dữ liệu được đánh số là 0. TCP tạo 500 gói tin cho chuỗi dữ liệu của tập tin. Trường số thứ tự của gói tin TCP đầu tiên được gán bằng 0, gói tin thứ hai là 1000, thứ ba là 2000 và tương tự như vậy.

MSS: Maximum Segment Size

MTU: Maximum Transmission Unit

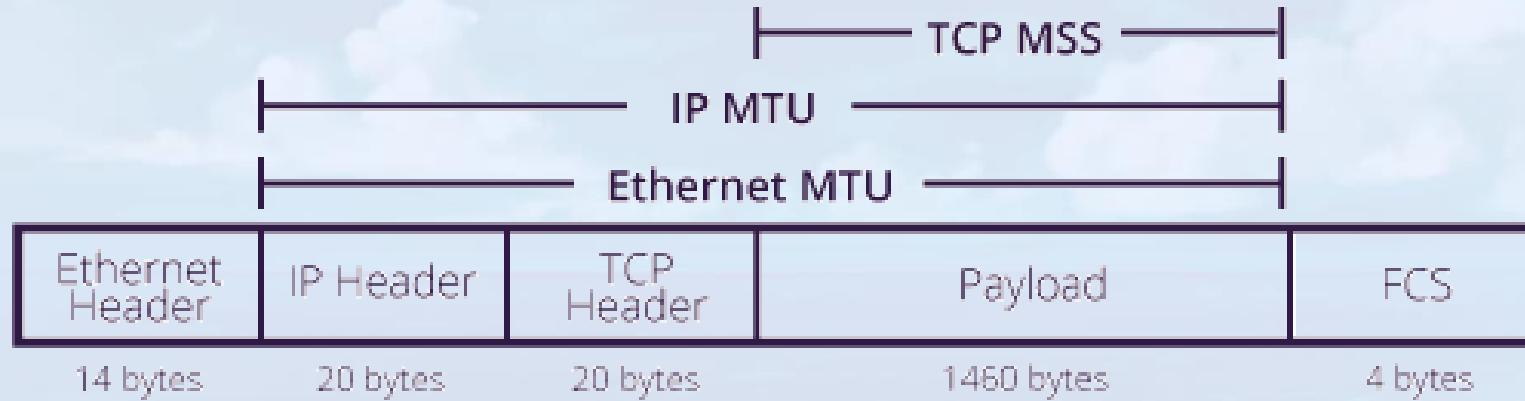


**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport



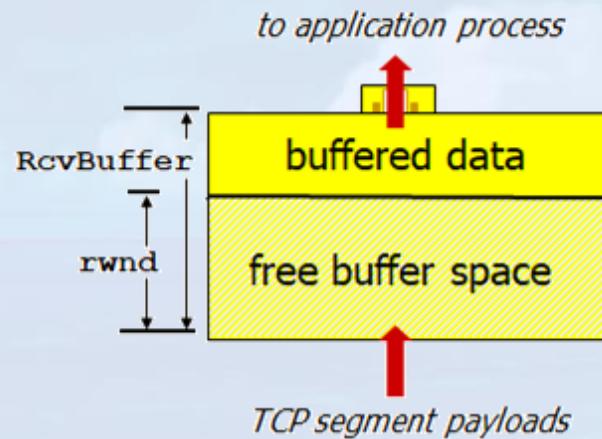
MSS: Maximum Segment Size

MTU: Maximum Transmission Unit



## TCP điều khiển luồng

- ❖ Bên nhận “quảng cáo” không gian bộ nhớ đệm còn trống bằng cách thêm giá trị **rwnd** trong TCP header của các segment từ bên nhận đến bên gửi
  - Kích thước của **RcvBuffer** được thiết lập thông qua các tùy chọn của socket (thông thường mặc định là 4096 byte)
  - Nhiều hệ điều hành tự động điều chỉnh **RcvBuffer**
- ❖ Bên gửi giới hạn số lượng dữ liệu không cần ACK tới giá trị **rwnd** của bên nhận
- ❖ Bảo đảm bộ đệm bên nhận sẽ không bị tràn



Bộ đệm phía bên nhận



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**3. Gói tin khởi tạo kết nối TCP từ client gửi đến server là gói**

SYN

ACK

FIN

RST



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**4. Trong giao thức truyền dữ liệu tin cậy (rdt), giao thức nào sau đây xử lý được trường hợp mất gói tin ACK?**

- rdt2.1
- rdt2.2
- rdt3.0
- Không xử lí được việc mất gói tin ACK





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

- + rtd 1.0: Truyền dữ liệu tin cậy. (Giả sử không có lỗi hay mất mát gì)
- + rtd 2.0: Kênh truyền không làm mất gói.

Dùng checksum, ACK (Acknowledgement) và NAK (Negative ACK) để check lỗi gói tin gửi về sender.

- + rtd 2.1: Kênh truyền có ACK, NAK bị lỗi

Bên gửi: Đánh số thứ tự 0, 1 cho các gói từ sender, gửi lại gói nếu NAK hoặc ACK/NAK bị lỗi (nhận biết bằng checksum trong gói), nhận được ACK thì mới gởi gói đánh số tiếp theo

Bên nhận: Có cơ chế loại packet bị trùng





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin

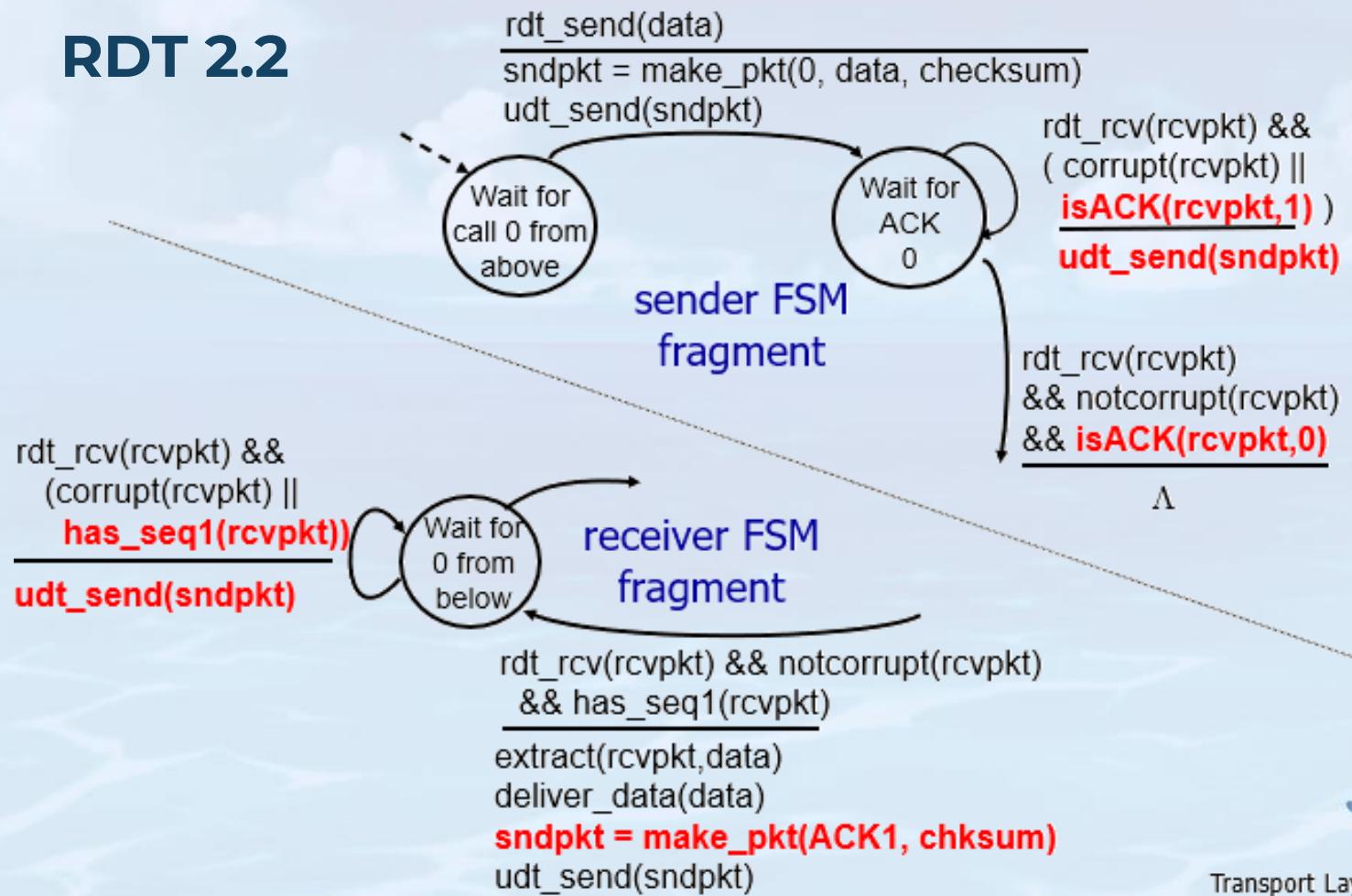


## Chương III: Tầng Transport

- + rtd 2.2: Không dùng NAK, thay bằng gửi ACK gói gần nhất nhận thành công
  - Bên gửi: Gửi lại gói nếu nhận ACK trùng lặp
  - Bên nhận: Gửi ACK gói gần nhất thành công, đánh số thứ tự 0,1 cho ACK
- + rtd 3.0: Xuất hiện mất gói
  - Bên gửi chờ ACK trong khoảng thời gian “Hợp lí”. Dùng Timeout (Bộ định thì). Gửi lại pckt khi hết thời gian.
  - Các trường hợp gửi lại gói: Mất gói, mất ACK, thời gian chờ ngắn/delayed ACK
- => Hiệu suất thấp:  $U_{\text{sender}} = (L/R)/(RTT+L/R)$  do phải nhận ACK rồi mới gửi gói tiếp được



## RDT 2.2





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 5. Ưu điểm của giao thức UDP so với TCP là:

- UDP bảo đảm các gói tin đơn lẻ được gửi đến đích đúng thứ tự
- UDP hỗ trợ cơ chế đồng bộ để đảm bảo không làm tràn ngập thông tin ở máy nhận
- UDP sử dụng ít tài nguyên máy tính hơn vì không phải duy trì trạng thái các kết nối
- UDP có cơ chế sửa lỗi tốt hơn TCP



## So sánh TCP và UDP

TCP:

- + Có cơ chế truyền tin cậy (không mất gói), đảm bảo thứ tự
- + Có cơ chế phân luồng (bên gửi không làm quá tải bên nhận)
- + Full duplex: Truyền và nhận cùng lúc
- + Có cơ chế bắt tay (Connection – oriented)
- + Giải quyết được tắc nghẽn
- + Chậm hơn UDP (thường do cơ chế sửa lỗi)
- + Dùng cho công việc cần bảo đảm dữ liệu: mail, web, file

UDP:

- + Gói tin gọn
- + Nhanh hơn TCP
- + Truyền không tin cậy
- + Dùng cho công việc cần tốc độ, băng thông tối thiểu: game online, video call, stream



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 6. TCP Header thường có kích thước là

- 4 bytes
- 8 bytes
- 16 bytes
- 20 bytes



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 7. Gói tin yêu cầu kết nối sẽ có giá trị các cờ như thế nào?

- ACK = 1, SYN = 1
- ACK = 0, SYN = 1
- RST = 1, SYN = 1
- FYN = 1, SYN = 0



## TCP bắt tay 3-way

*Trạng thái client*

LISTEN

Chọn số thứ tự ban đầu, x  
Gửi TCP SYN msg

SYNSENT

ESTAB

SYNACK(x) vừa được nhận  
cho hay server vẫn còn sống;  
send ACK for SYNACK;  
this segment may contain  
client-to-server data



SYNbit=1, Seq=x

SYNbit=1, Seq=y  
ACKbit=1; ACKnum=x+1

ACKbit=1, ACKnum=y+1

*Trạng thái server*

LISTEN

SYN RCVD

Chọn số thứ tự ban đầu, y  
gửi TCP SYNACK  
msg, acking SYN

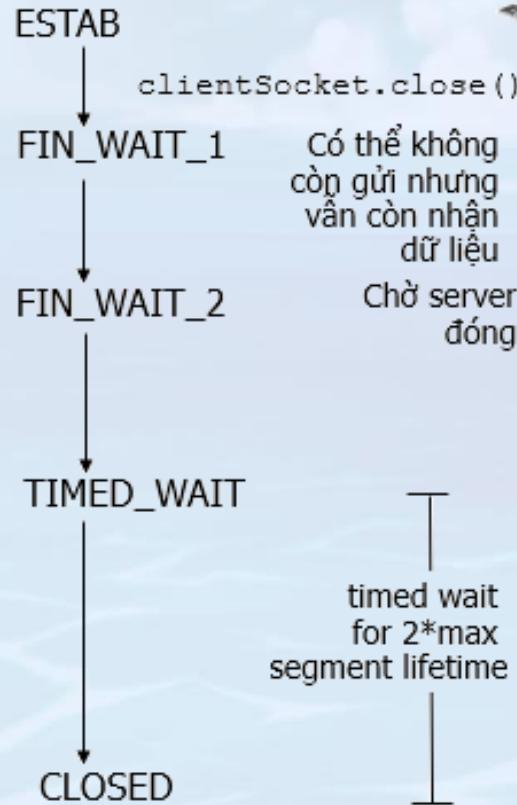
ESTAB

ACK(y) vừa được nhận  
cho hay client vẫn sống

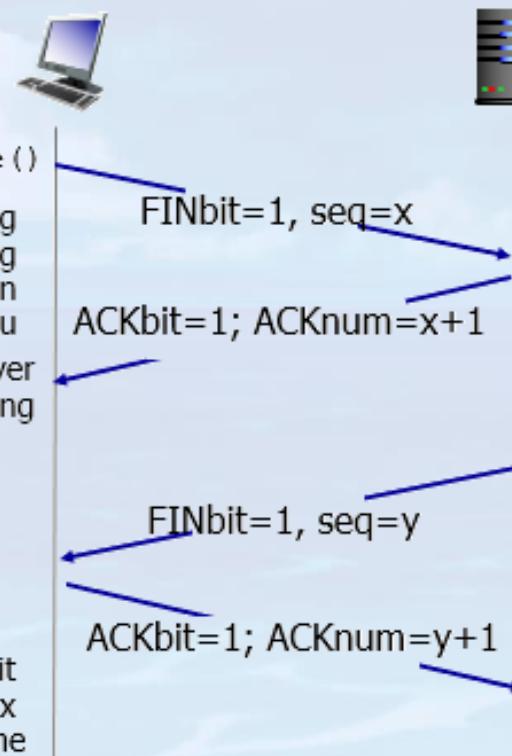
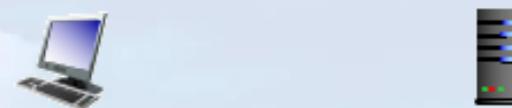


## TCP đóng kết nối

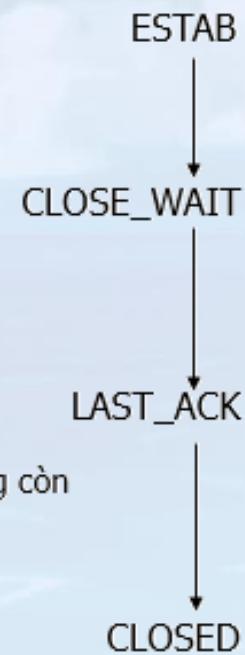
*Trạng thái client*



Có thể không  
còn gửi nhưng  
vẫn còn nhận  
dữ liệu  
Chờ server  
đóng



*Trạng thái server*



vẫn có thể  
gửi dữ liệu

Có thể không  
còn  
gửi dữ liệu



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 8. Trường Window trong header TCP liên quan đến

- Kích thước bộ nhớ khả dụng
  - Hệ điều hành
  - Kích thước màn hình
  - Tất cả đều sai



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**9. Giả sử thực thể TCP A cần gửi 1300 bytes cho thực thể giao vận B. Gói thứ nhất chứa 1000 bytes dữ liệu, trường Sequence Number của gói này là 100. Trường Sequence Number của gói thứ hai sẽ là?**

- 300
- 400
- 1100
- 1400



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

- 9. Giả sử thực thể TCP A cần gửi 1300 bytes cho thực thể giao vận B. Gói thứ nhất chứa 1000 bytes dữ liệu, trường Sequence Number của gói này là 100. Trường Sequence Number của gói thứ hai sẽ là?**

Seq 1 = 100, payload 1 = 1000 bytes => ACK phản hồi = 1100 bytes => Seq2 = 1100



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**10. Khi thực thi TCP gửi một gói SYN-ACK segment với trường Acknowledgement Number = 100, điều này có nghĩa là?**

- Gói dữ liệu nó gửi đi bắt đầu từ byte thứ 100 trong dòng dữ liệu
- Byte đầu tiên trong dòng dữ liệu sẽ gửi đi có số thứ tự là 100
- Nó sẽ gửi từ byte thứ 100
- Nó hy vọng nhận được dữ liệu bắt đầu bằng byte có số thứ tự 100



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 11. Để phát hiện lỗi trong gói tin người ta sử dụng kỹ thuật:

- Số thứ tự (Sequence number)
- Số thứ tự ghi nhận (Acknowledgement number)
- Checksum
- Bộ định thời (Timer)





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 12. Mục đích của sequence number trong Header TCP là gì?

- Số thứ tự byte đầu tiên của trường dữ liệu trong dòng byte gửi đi
- Định danh các ứng dụng của tầng Application
- Xác định số byte kế tiếp
- Hiển thị số byte tối đa cho phép truyền trong 1 session



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**13. Gói tin TCP từ server gửi trả về cho client khi client yêu cầu kết nối sẽ có giá trị các cờ là?**

- ACK = 0, SYN = 1
- ACK = 1, SYN = 0
- ACK = 1, SYN = 1
- ACK = 0, SYN = 0



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**14. Khái niệm Multiplexing và Demultiplexing được sử dụng ở tầng nào trong mô hình TCP/IP?**

- Application
- Network
- Transport
- Data Link



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin

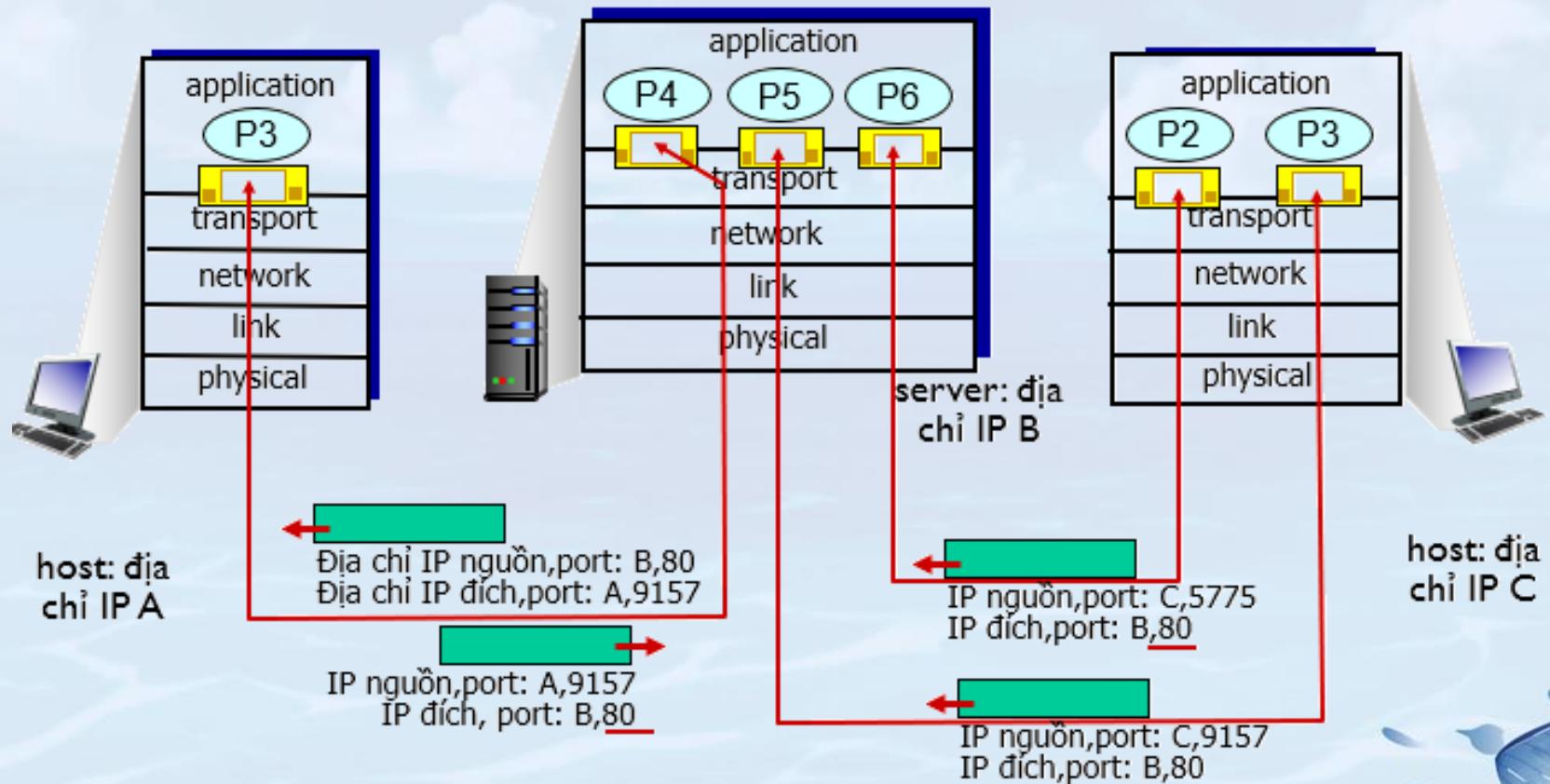


## Multiplexing và Demultiplexing

- + Demultiplexing: phân các segment về đúng socket dựa trên những trường dữ liệu nằm trên segment
- + Multiplexing: tổng hợp data chunks từ socket lại thành segment, đánh header phục vụ demultiplexing



## Multiplexing và Demultiplexing





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 15. TCP socket được xác định bởi bộ 4:

- Địa chỉ IP nguồn, địa chỉ IP đích, số port nguồn, số port đích
- Địa chỉ MAC nguồn, địa chỉ MAC đích, số port nguồn, số port đích
- Địa chỉ IP nguồn, địa chỉ IP đích, địa chỉ MAC nguồn, địa chỉ MAC đích
- Đáp án khác



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 16. Trong RDT 3.0, chuyện gì sẽ xảy ra khi bên gửi không nhận được ACK của bên nhận?

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



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**17. 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?**

- rdt 1.0
- rdt 2.0
- rdt 3.0
- Pipelined



## Go-back-N

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

sender

send pkt0  
send pkt1  
send pkt2  
send pkt3  
(wait)

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

rcv ack0, send pkt4  
rcv ack1, 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



*pkt 2 timeout*

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

send pkt2  
send pkt3  
send pkt4  
send pkt5

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



## Selective Repeat

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

sender

send pkt0  
send pkt1  
send pkt2  
send pkt3  
(wait)

rcv ack0, send pkt4  
rcv ack1, send pkt5

record ack3 arrived



*pkt 2 timeout*

send pkt2

record ack4 arrived

record ack4 arrived

receiver

receive pkt0, send ack0  
receive pkt1, send ack1

receive pkt3, buffer,  
send ack3

receive pkt4, buffer,  
send ack4

receive pkt5, buffer,  
send ack5

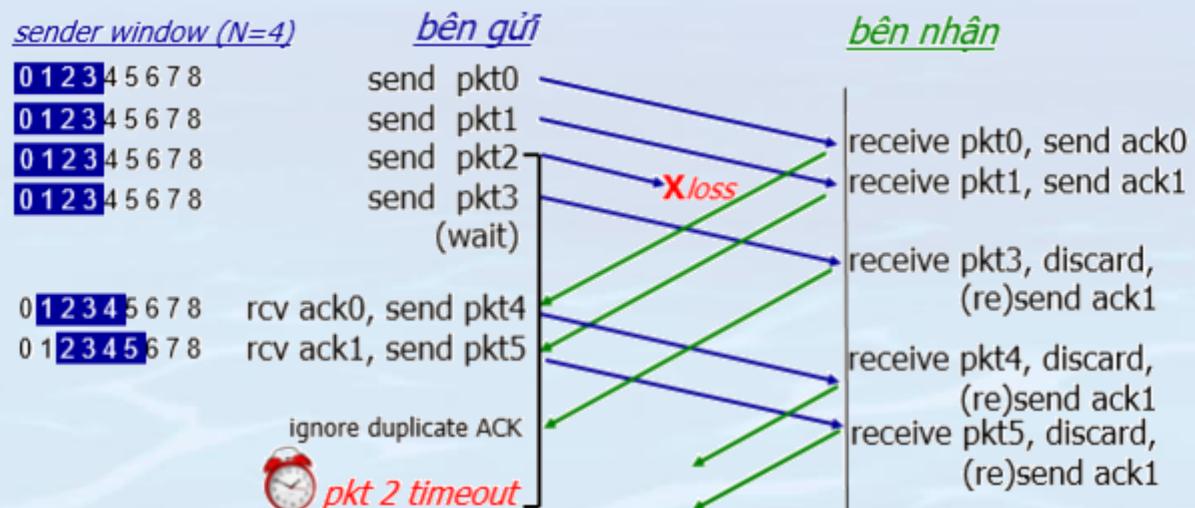
rcv pkt2; deliver pkt2,  
pkt3, pkt4, pkt5; send ack2

*Q: what happens when ack2 arrives?*



## Chương III: Tầng Transport

**18. Xem hình mô tả hoạt động của Go-back-N dưới đây, sau thời gian timeout, bên gửi sẽ hành động như thế nào?**

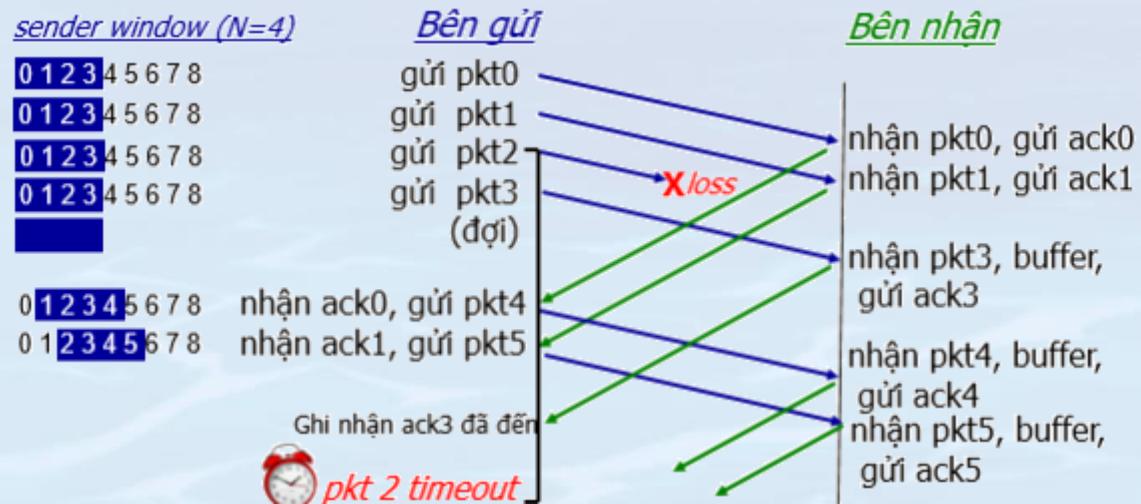


- Chỉ gởi lại pkt2
- Gởi lại pkt0, pkt1, pkt2, pkt3
- Gởi lại pkt1, pkt2, pkt3, pkt4
- Gởi lại pkt2, pkt3, pkt4, pkt5



## Chương III: Tầng Transport

**19. Xem hình mô tả hoạt động của Lặp lại có lựa chọn (Selective repeat) dưới đây, sau thời gian timeout, bên gửi sẽ hành động như thế nào?**



- Chỉ gởi lại pkt2
- Gởi lại pkt0, pkt1, pkt2, pkt3
- Gởi lại pkt1, pkt2, pkt3, pkt4
- Gởi lại pkt2, pkt3, pkt4, pkt5



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**20. Lượng dữ liệu lớn nhất mà một segment có thể chứa được giới hạn bởi thông số nào sau đây?**

- Băng thông tối đa
- MTU (Maximum Transmission Unit)
- MSS (Maximum Segment Size)
- Băng thông tối đa và MTU



**UIT**  
Trường Đại học  
Công nghệ Thông tin

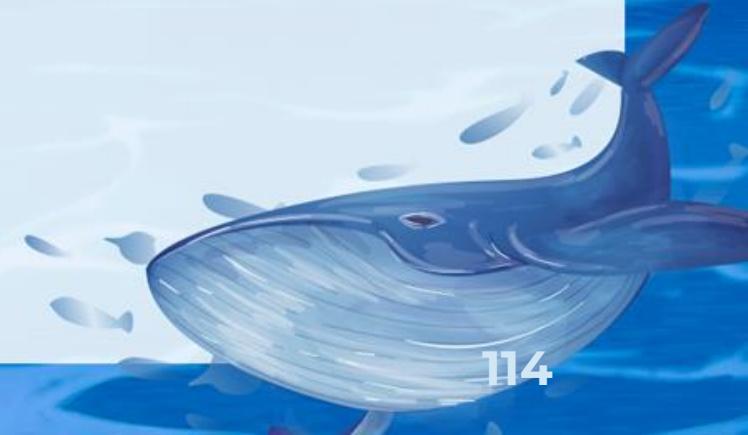
Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 21. Trong giao thức TCP, Initial Sequence Number (ISN) sẽ bằng

- 1
- 100
- 0
- Do hệ điều hành tạo ra bằng thuật toán





**UIT**  
Trường Đại học  
Công nghệ Thông tin

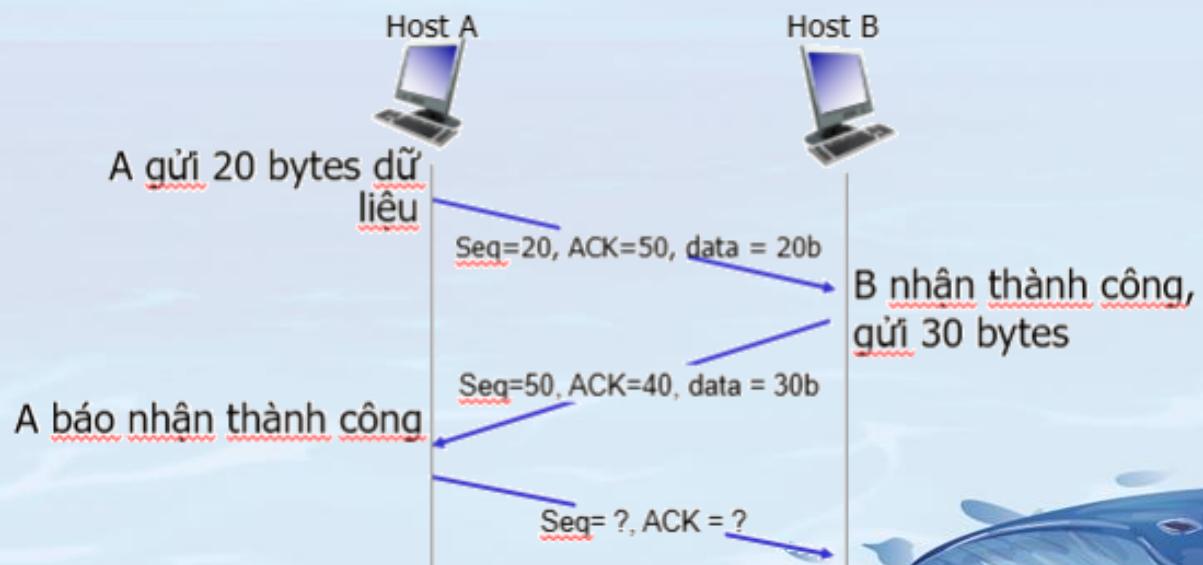
Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

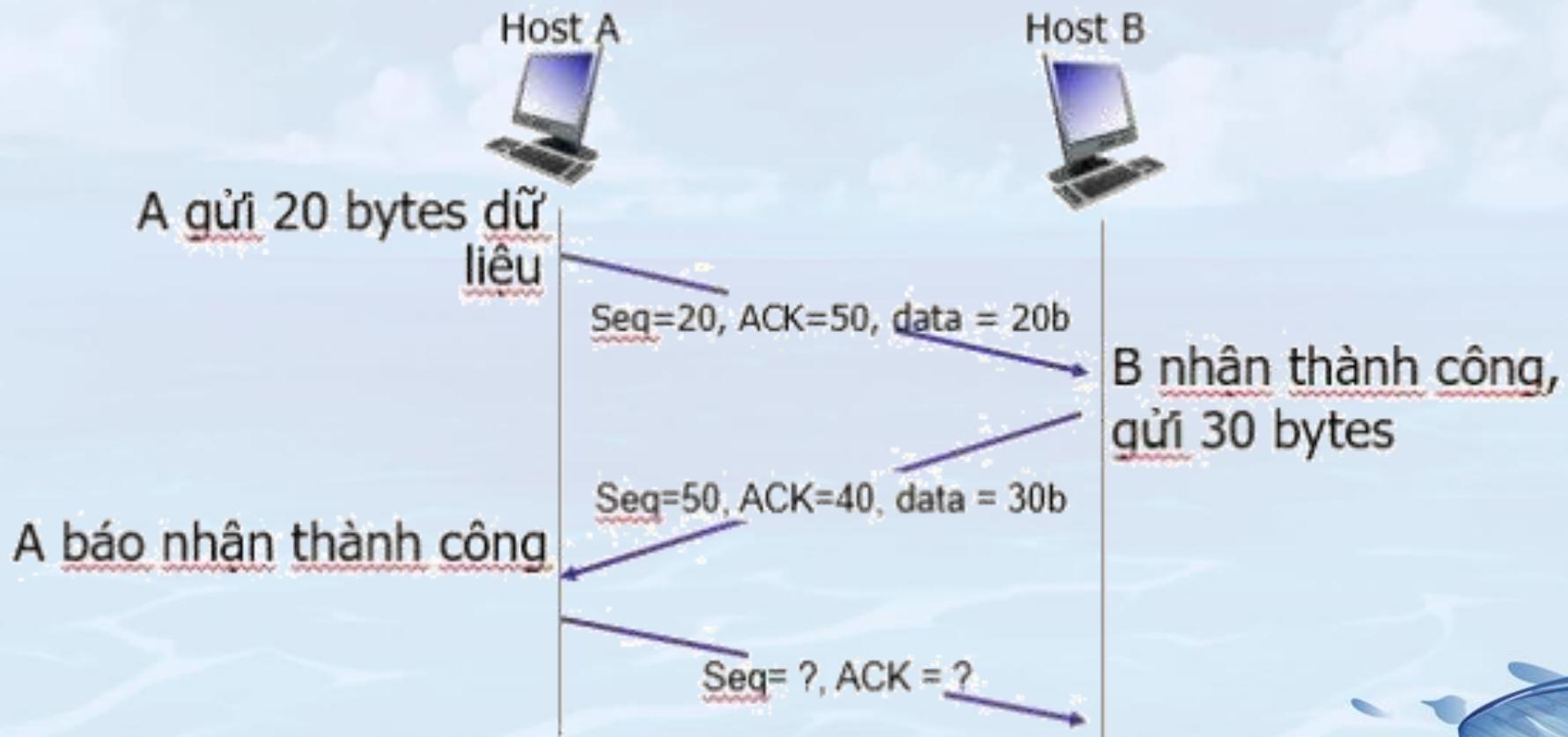
**22. Dựa trên hình dưới đây, giá trị của số thứ tự (SEQ) và số ACK trong gói tin cuối cùng là bao nhiêu?**

- Seq = 80, ACK = 50
- Seq = 40, ACK = 50
- Seq = 50, ACK = 80
- Seq = 40, ACK = 80





## Chương III: Tầng Transport





### Chương III: Tầng Transport

23. Giả sử một kết nối TCP có 4 segment ACK quay về Bên Gửi và nhờ đó người ta đo được thời gian đi-về của segment thứ nhất (SampleRTT1) là 90 msec, thứ hai (SampleRTT2) là 110 msec, thứ ba (SampleRTT3) là 114 msec, và và thứ tư (SampleRTT4) là 88 msec. Giả sử hệ số  $\alpha=0.2$ . Người ta ước lượng được giá trị EstimatedRTT ngay sau khi ACK thứ hai quay về là bao nhiêu?

- 92.88 msec
- 100.5 msec
- 94 msec
- Không ước lượng được



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

Các công thức:

$$\text{EstimatedRtt} = (1 - \alpha) * \text{EstimatedRtt} + \alpha * \text{SampleRtt}$$

$$\text{DevRtt} = (1 - \beta) * \text{DevRtt} + \beta * |\text{SampleRtt} - \text{EstimatedRtt}|$$

$$\text{TimeoutInterval} = \text{EstimatedRtt} + 4 * \text{DevRtt}$$



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

23. Giả sử một kết nối TCP có 4 segment ACK quay về Bên Gửi và nhờ đó người ta đo được thời gian đi-về của segment thứ nhất (SampleRTT1) là 90 msec, thứ hai (SampleRTT2) là 110 msec, thứ ba (SampleRTT3) là 114 msec, và và thứ tư (SampleRTT4) là 88 msec. Giả sử hệ số  $\alpha=0.2$ . Người ta ước lượng được giá trị EstimatedRTT ngay sau khi ACK thứ hai quay về là bao nhiêu?

Có  $\text{EstimatedRtt1} = \text{SampleRtt1} = 90 \text{ msec}$

$$\Rightarrow \text{EstimatedRtt2} = (1 - \alpha) * \text{EstimatedRtt1} + \alpha * \text{SampleRtt2} = (1 - 0.2)90 + 0.2 * 110 = 94 \text{ msec}$$



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

- \* Nếu hỏi giá trị time out sau lần đo thứ 2 (110 msec) thì sao? Biết độ lệch RTT của lần đo đầu tiên DevRTT = 10 ms,  $\beta = 0.25$

Lần 2:

$$\begin{aligned} \text{DevRtt2} &= (1 - \beta) * \text{DevRtt} + \beta * |\text{SampleRtt2} - \text{EstimatedRtt2}| = \\ &(1 - 0.25) * 10 + 0.25 |110 - 94| = 11.5 \end{aligned}$$

$$\begin{aligned} \text{TimeoutInterval} &= \text{EstimatedRtt2} + 4 * \text{DevRtt} = 94 + 4 * 11.5 = \\ &140 \text{ msec} \end{aligned}$$



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**24. Trong giao thức TCP, SYN segment của quá trình bắt tay 3 bước sẽ có Sequence Number (Seq) và giá trị SYN flag là bao nhiêu?**

- Seq = 0, SYN = 0
- Seq = ISN, SYN = 1 (ISN: initial sequence number)
- Seq = 1, SYN = 1
- Seq = ISN, SYN = 0



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

**25. Tính checksum của 2 chuỗi 16 bit sau:  
1010 1100 0101 0001 và 0100 1001 1100 1100**

- 0000 1001 1110 0010
- 0100 1001 1110 0010
- 0000 1001 1111 0010
- 0001 1001 1110 0010



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

1010 1100 0101 0001  
0100 1001 1100 1100  
1111 0110 0001 1101  
0000 1001 1110 0010



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### Điều khiển tắc nghẽn TCP

- Bên gửi tăng tốc độ truyền (kích thước window), thăm dò băng thông có thể sử dụng, cho đến khi mất mát gói xảy ra
- Congestion window (CWND) quyết định số lượng byte có thể được gửi đi



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### Giải thích thuật toán TCP Reno qua các giai đoạn

#### 1. Slow Start

- Khi kết nối bắt đầu, cwnd = 1 MSS. Tăng kích thước cwnd theo cấp số nhân cho đến khi sự kiện mất gói đầu tiên xảy ra. Tốc độ ban đầu chậm, nhưng nó sẽ tăng lên theo cấp số nhân
- Điều kiện:  $cwnd < ssthresh$  (slow start threshold) hoặc timeout (trường hợp timeout bắt đầu slow start lại)
- Các giá trị:
  - $cwnd_{next} = cwnd_{prev} * 2$
  - $ssthresh = \frac{1}{2} cwnd$  lần tắc nghẽn liền trước



**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 2. Congestion avoidance (CA)

- Điều kiện: cwnd  $\geq$  ssthresh
- Các giá trị:
  - cwndnext = cwndprev + 1 MSS (đơn vị dữ liệu gửi trong thời gian)
  - ssthresh giữ nguyên

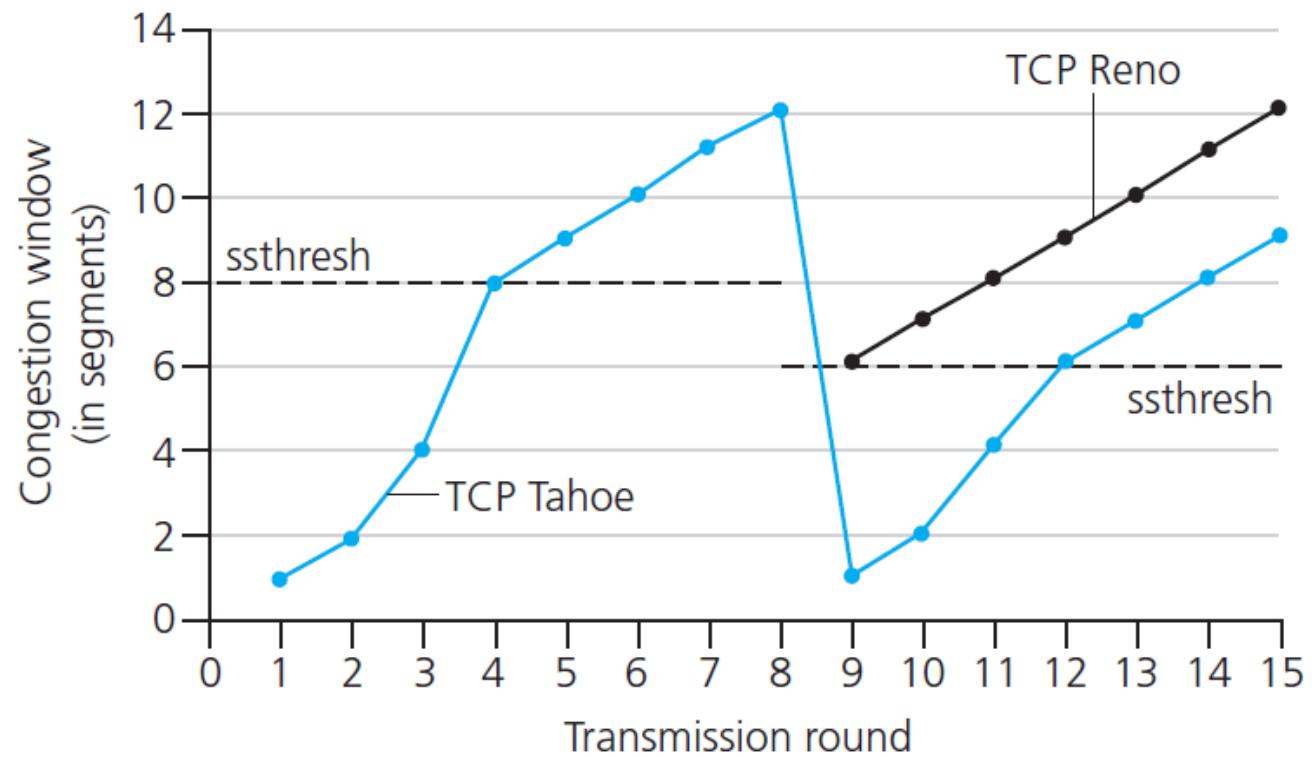


**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chuyển từ Slow Start sang CA





**UIT**  
Trường Đại học  
Công nghệ Thông tin

Khoa Khoa học  
và Kỹ thuật Thông tin



## Chương III: Tầng Transport

### 3. Fast Recovery

- Cách nhận biết: 3 ACK trùng

=> Fast – Retransmit rồi vào Fast - Recovery

- Tham số:

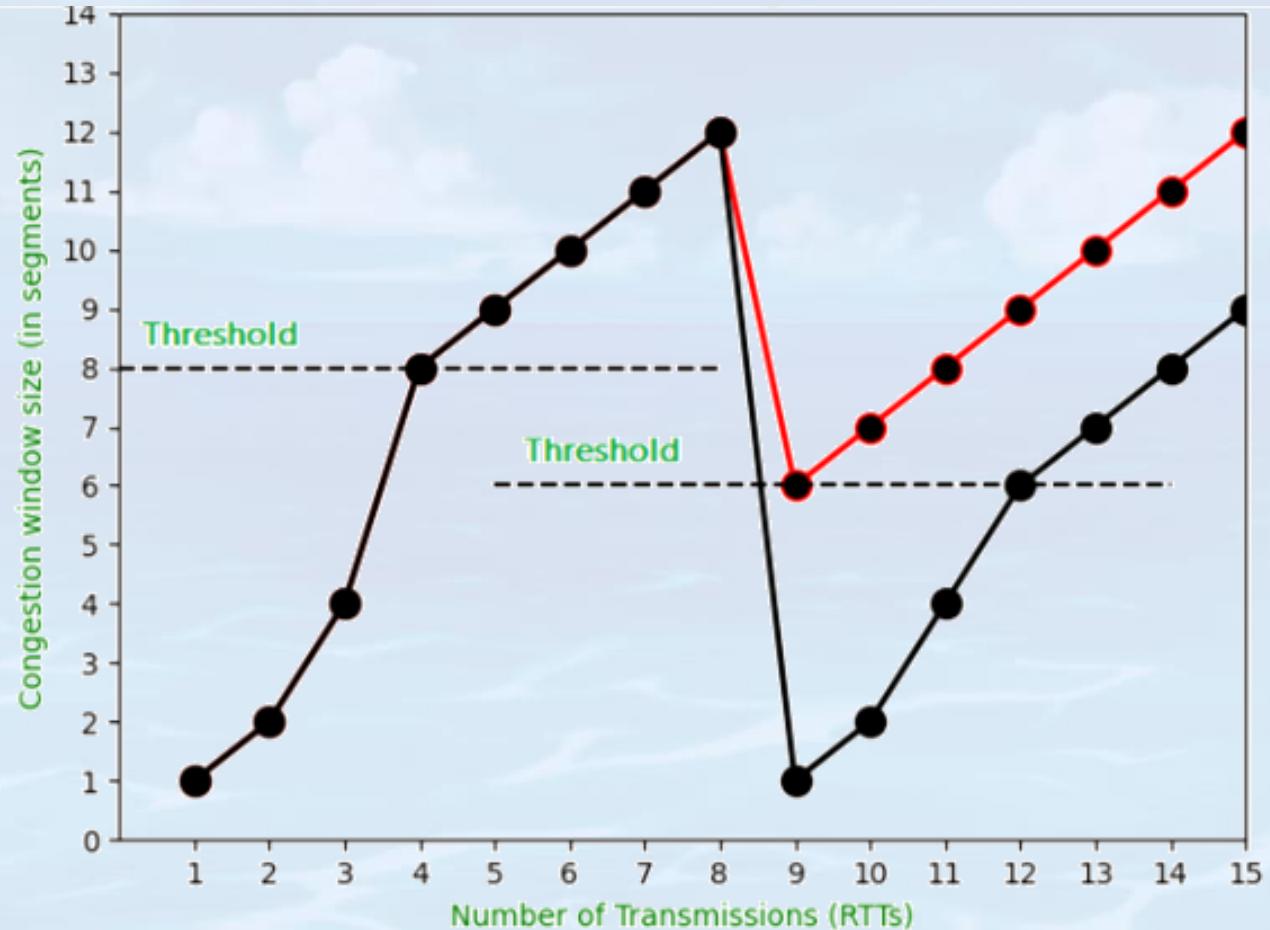
- $cwnd = \frac{1}{2} cwnd_{prev}$  (do có 3 gói tin phản hồi ACK trùng) và  $cwnd_{next} = cwnd_{prev} + MSS$

- $ssthresh = new cwnd$

=> Nếu gặp time out lần nữa thì Start slow lại



## Chương III: Tầng Transport





Cảm ơn các bạn đã tham gia buổi Training của BHT Khoa KHKTTT

# NHẬP MÔN MẠNG MÁY TÍNH



Điểm danh



Thi Thử