

| | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1. D | 2. B | 3. A | 4. C | 5. C | 6. A | 7. D | 8. B | 9. B | 10. C |
| 11. B | 12. A | 13. C | 14. C | 15. D | 16. C | 17. B | 18. A | 19. A | 20. B |
| 21. A | 22. B | 23. C | 24. D | 25. B | 26. A | 27. C | 28. C | 29. B | 30. D |

1. Giả sử Host A muốn gửi 1 file có kích thước 15 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 $R_1 = 5$ Mbps, $R_2 = 4$ Mbps, $R_3 = 6$ Mbps. Thời gian truyền file đến host B là:

- A. 37.5s
B. 3.75s
C. 3s
D. 30s

Giải thích:

Thông lượng của đường truyền = $\min\{R_1, R_2, R_3\} = \min\{5\text{Mbps}, 4\text{Mbps}, 6\text{Mbps}\} = 4\text{Mbps}$

Thời gian truyền file từ Host A đến Host B = L/R

2. Trong segment mà bên nhận gửi cho bên gửi, nếu số ACK là 200, có nghĩa là bên nhận đã nhận được byte thứ bao nhiêu:

- A. 200
B. 199
C. 201
D. Không có đáp án đúng

Giải thích: Mỗi byte gửi đi đều có một số thứ tự và khi nhận được thì máy tính nhận gửi lại tin báo nhận (ACK). ACK chỉ ra rằng ở lượt kế tiếp mong đợi nhận được byte thứ bao nhiêu ($ACK\ n = SEQ\ n+1$)

$\Rightarrow ACK = 200$ tức ở lượt kế tiếp mong đợi nhận được byte thứ 200 và trước đó đã nhận được byte thứ 199

3. Đường truyền nào sau đây có tốc độ truyền cao nhất và tốc độ lỗi thấp nhất trong thực tế?

- A. Cáp quang
B. Kênh vệ tinh
C. Cáp xoắn (TP)
D. Cáp đồng trục

4. Giả sử chương trình bắt gói tin Wireshark bắt được chuỗi byte (biểu diễn bằng ASCII) do một Web server gửi đến một trình duyệt Web (Web browser) như sau, để trả lời cho một request message trước đó mà nó nhận được. Trong đó, ký hiệu `<cr><lf>` là về đầu dòng tiếp theo:

HTTP/1.1 200 OK<cr><lf>Date: Fri, 11 Dec 2020 12:39:45 GMT..Server:
Apache/2.0.52 (Fedora) <cr><lf>Last-Modified: Thu, 10 Dec 2020 18:27:46
GMT<cr><lf>ETag: "526c3-f22-a88a4c80"Accept-Ranges: bytes<cr><lf>Content-

```
Length: 3784<cr><lf>Keep-Alive: timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-Type: text/html; charset=ISO-8859-1<cr><lf><cr><lf><!doctype html public "-//w3c//dtd html 4.0 transitional//en"><lf><html><lf><head><lf><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><lf><meta name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT 5.0; U Netscape]"><lf><title>CMPSCI 453 / 591 / NTU-ST550A Spring 2020 homepage</title><lf></head><lf><còn nữa nhưng không in ra ở đây>
```

Thời điểm tài liệu được cập nhật lần cuối cùng?

- A. Fri, 11 Dec 2020 12:39:45 GMT
- B. Thu, 10 Dec 2021 18:27:46 GMT
- C. Thu, 10 Dec 2020 18:27:46 GMT**
- D. Không có đáp án đúng

Giải thích: Ở dòng thứ 2, sau Last-Modified chỉ ra thời điểm cuối cùng mà tài liệu được cập nhật

5. Hai máy tính A và B kết nối với nhau qua một đường truyền có tốc độ R bps, và khoảng cách là m. Tốc độ lan truyền của tín hiệu trên đường truyền là s. Máy A gửi 1 gói tin đến máy B. Cho $s = 2.5 \times 10^5$ (km/s), $L = 200$ bits, $R = 60$ 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. 6667 m
- B. 833 m
- C. 833 km**
- D. 6667 km

Giải thích: $t_{trans} = t_{prop} \Leftrightarrow L/R = d/s$

6. Giả sử rằng độ dài gói là $L = 25000$ bits, và tốc độ truyền liên kết là $R = 200$ Mbps. Số gói tin tối đa trong một giây có thể được truyền bởi liên kết này là bao nhiêu?

- A. 8000**
- B. 125
- C. 1000
- D. Không có đáp án

Giải thích: $R/L = (200 \times 10^6)/25000 = 8000$ (gói)

7. Để tải 1 tài liệu văn bản với tốc độ 100 trang mỗi giây, ta giả sử rằng một trang tài liệu trung bình có 24 dòng với 80 ký tự (mỗi ký tự sử dụng mã 8 bit) trên mỗi dòng. Băng thông tối thiểu của kênh truyền là bao nhiêu?

- a. 12688000 bps
- b. 1586000 bps
- c. 15860 Kbps
- d. 1536000 bps**

Giải thích: Cứ 1 trang tài liệu có 24 dòng với 1 dòng chứa 80 ký tự

Tốc độ tải: $100 \times 24 \times 80 = 192000$ ký tự/s

Với mỗi ký tự sử dụng mã 8 bit \Rightarrow Tốc độ tải: $192000 \times 8 = 1536000$ bit/s

Băng thông tối thiểu của kênh truyền là 1536000 bps

8. Giả sử từ trình duyệt, bạn click vào 1 link dẫn đến 1 trang web. Giả sử địa chỉ IP của URL của link đó đã được lưu tại bộ nhớ cache máy tính của bạn, nên việc truy vấn DNS là không cần thiết. Ký hiệu RTT là thời gian đi – về 1 vòng giữa máy tính của bạn và Web server chứa trang web. Giả sử trang web bao gồm 1 trang cơ sở và 8 ảnh nhỏ. Giả sử thời gian đẩy dữ liệu lên đường truyền là không đáng kể so với RTT. Cần khoảng thời gian bao lâu, theo số RTT, tính từ khi bạn click vào link cho tới khi nhận được toàn bộ trang web trong trường hợp trình duyệt sử dụng kết nối HTTP thường trực (persistent HTTP), và pipelining.

A. 8 B. 3 C. 9 D. 4

9. Tầng Mạng cung cấp truyền thông logic giữa các thành phần nào sau đây?

A. Tiến trình B. Host
C. Host và tiến trình D. Router

Giải thích:

○ *Tầng Mạng: truyền thông logic giữa các host*

○ *Tầng Vận chuyển: truyền thông logic giữa các tiến trình dựa trên dịch vụ tầng Mạng*

10. Dịch vụ nào cho phép dùng tên miền thay vì dùng địa chỉ IP khi duyệt Web Internet

A. POST B. HTTP C. DNS D. FTP

11. Định danh (Identifier) của một tiến trình trên hệ thống đầu cuối bao gồm:

A. Địa chỉ MAC và số hiệu cổng
B. Địa chỉ IP và số hiệu cổng
C. Địa chỉ IP và giao thức
D. Địa chỉ IP và địa chỉ MAC

Giải thích: Định danh (identifier) bao gồm địa chỉ IP và số cổng (port number) được liên kết với tiến trình trên hệ thống đầu cuối.

12. Trong số các cặp giao thức và cổng dịch vụ sau, cặp nào là ĐÚNG?

A. DNS: TCP Port 53 B. HTTP: UDP Port 80
C. TFTP: TCP Port 69 D. SSH: TCP Port 23

Giải thích:

- B sai vì HTTP sử dụng TCP
- C sai vì TFTP sử dụng UDP
- D sai vì SSH sử dụng Port 22

13. Giả sử có một bản ghi của dịch vụ DNS là (cnn.com, r1.infor.cnn.com, CNAME).
Hãy chọn đáp án đúng

- A. r1.infor.cnn.com là tên miền phụ
- B. r1.infor.cnn.com là tên miền, không phải là tên một máy
- C. cnn.com là tên viết tắt (rút gọn) của r1.infor.cnn.com**
- D. cnn.com là tên của máy chủ tên miền r1.infor.cnn.com, có thẩm quyền quản lý tên miền này

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

Type = CNAME

- *name là bí danh của một tên “gốc” (tên thực)*

VD: www.ibm.com tên thực là máy chủ east.backup2.ibm.com

- *value là tên gốc*

14. Hãy chọn phát biểu đúng:

- A. Thông điệp HTTP response không bao giờ có phần thân (message body) rỗng
- B. Trường Date trong header của một thông điệp HTTP response chỉ ra thời điểm chỉnh sửa sau cùng của đối tượng
- C. Hai trang web khác nhau trên cùng một host (ví dụ, www.mit.edu/research.html và www.mit.edu/students.html) có thể được gửi trên cùng một kết nối bền vững**
- D. Đối với một kết nối không bền vững giữa trình duyệt và máy chủ, nó cho phép một gói tin TCP thực hiện 2 thông điệp HTTP request khác nhau

Giải thích:

- A sai vì thông điệp HTTP response có hoặc không có phần nội dung của thông điệp (message-body) đều được.
- B sai vì trường Date chứa thông tin ngày tháng thông tin (response) được phát sinh.
- D sai vì với kết nối HTTP không bền vững, không thể gửi nhiều đối tượng trên một kết nối.

15. Đặc điểm nào dưới đây KHÔNG được hỗ trợ trong dịch vụ TCP?

- A. Điều khiển luồng thông tin
- B. Điều khiển tắc nghẽn
- C. Hướng kết nối
- D. Bảo đảm thông lượng tối thiểu**

Giải thích:

- Những đặc điểm KHÔNG được hỗ trợ trong dịch vụ TCP: định thì, bảo đảm thông lượng tối thiểu, bảo mật

16. Một trang web có đính kèm 3 file hình và 11 file âm thanh. Nếu browser dùng non-persistent để download toàn bộ trang về để hiển thị thì mất bao nhiêu RTT?

- A. 14
- B. 28
- C. 30
- D. 15

Giải thích: Gồm 1 file HTML, 3 file hình và 11 file âm thanh \Rightarrow 15 đối tượng HTTP 1.0: Cần 2 RTT cho mỗi đối tượng \Rightarrow Cần tất cả $2 \times 15 = 30$ RTT để download toàn bộ.

17. Đoạn mã bên dưới được viết cho ứng dụng nào?

```
from socket import *
startPort = 12000
startSocket = socket(AF_INET, SOCK_STREAM)
startSocket.bind(('', startPort))
startSocket.listen(1)
while 1:
    connectionSocket, addr = startSocket.accept()
    sentence = connectionSocket.recv(1024)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence)
    connectionSocket.close()
```

- A. Ứng dụng TCP socket phía client
- B. Ứng dụng TCP socket phía server
- C. Ứng dụng UDP socket phía server
- D. Ứng dụng UDP socket phía client

Nếu là SOCK_DGRAM thì là
UDP Socket

Nếu là SOCK_STREAM thì là
TCP Socket

Không có vòng lặp vô tận while
1 của server để trực chờ nhận
request của client

```
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()
```

```
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)
```

18. Cho đoạn mã HTML sau:

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 22 June 2015 12:39:45GMT..Server:
Apache/2.0.52 (Fedora) <cr><lf>Last-Modified: Tue, 1 June 2014 18:27:46
GMT<cr><lf>ETag: "526c3-f22-88a4c80"<cr><lf>Accept-Ranges:
bytes<cr><lf>Content-Length: 8347<cr><lf>Keep-Alive:
timeout=max=100<cr><lf>Connection: KeepAlive<cr><lf>Content-Type: text/html;
charset=ISO-88591<cr><lf><cr><lf><!doctype html public "-//w3c//dtd html 4.0
transitional//en"><lf><html><lf><head><lf><meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf><meta name="GENERATOR"
content="Mozilla/4.79 [en] (Windows NT 5.0; U) Netscape]"><lf><title>Test
page</title><lf></head><lf> .....
```

7 bytes đầu tiên trang web trả về cho trình duyệt là?

- A. <!docty
- B. <meta
- C. HTTP/1
- D. text/ht

Giải thích: Dữ liệu được chia thành các dòng và định dạng kết thúc dòng là <CR><LF> (tương ứng với các ký tự 0x0A, 0x0D trong bảng mã ASCII). Riêng dòng cuối cùng của gói tin sẽ được báo hiệu bằng 2 lần định dạng kết thúc: <CR><LF><CR><LF>.

Chương 3:

19. Tính giá trị Internet checksum cho hai từ 16 bit này:

11100010 01000001
10111100 10100000

- A. 01100001 00011101
- B. 00001001 11100010
- C. 01001001 11100010
- D. 01100011 00011101

19. Đáp án A

Cộng 2 chuỗi 16 bit ta được: 1 10011110 11100001, xuất hiện thừa giá trị 1 bên trái, ta cộng vào chuỗi 10011110 11100001, được 10011110 11100010, checksum sẽ là tổng bù 1 của dãy trên, là: 01100001 00011101

20. Trong TCP header, số thứ tự (sequence number) thể hiện điều gì?

- A. Tổng số byte được gửi
- B. Số thứ tự của byte đầu tiên trong dữ liệu của segment
- C. Số thứ tự của segment được gửi

D. Tổng số byte bên nhận đang mong đợi sẽ được nhận tiếp tục

20. Đáp án B

21. Trong header của UDP, trường length là độ dài của thành phần nào sau đây?

A. Cả segment UDP

B. Chỉ phần đầu header của UDP

C. Chỉ phần dữ liệu (payload)

D. Trong header của UDP không có trường length

21. Đáp án A

Trường length tính cả độ dài của header và phần dữ liệu (payload) của segment

22. Trong TCP, khi timeout xảy ra, bên gửi sẽ thực hiện hành động nào sau đây?

A. Gửi gói tin tiếp theo

B. Gửi lại gói tin bị timeout

C. Thực hiện kết nối lại với bên nhận

D. Huỷ kết nối

22. Đáp án B

23. Cho các giá trị SampleRTT đo được sau mỗi lần gửi yêu cầu và nhận được phản hồi tương ứng là: 106 ms, 120 ms, 140 ms, 90 ms, and 115 ms. Biết giá trị EstimatedRTT ban đầu là 100ms, chỉ số $\alpha = 0.125$, $\beta = 0.25$, $DevRTT = 5ms$.

Tính xấp xỉ giá trị timeout sau lần đo SampleRTT thứ 2 (120 ms)

A. 128.6 msec

B. 127.36 msec

C. 135.16 msec

D. 214.23 msec

23. Đáp án C

Áp dụng công thức tính

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

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

*$TimeoutInterval = EstimatedRtt + 4 * DevRtt$*

Lần 1:

*$EstimatedRtt_1 = (1 - 0.125) * 100 + 0.125 * 106 = 100.75 msec$*

*$DevRtt = (1 - 0.25) * 5 + 0.25 * |106 - 100.75| = 5.0625 msec$*

Lần 2:

*$EstimatedRtt_2 = (1 - 0.125) * 100.75 + 0.125 * 120 = 103.16 msec$*

*$DevRtt = (1 - 0.25) * 5.0625 + 0.25 * |120 - 103.16| = 8 msec$*

*$TimeInterval = 103.16 + 4 * 8 = 135.16 msec$*

24. Tầng Transport sẽ đóng gói dữ liệu thành các:

A. Bit

B. Frame

C. Packet

D. Segment

24. Đáp án D

Application - Data

Transport - Segment

Network - Packet

Data Link - Frame

Physical - Bit

25. Cho mô hình trao đổi gói tin TCP như sau:

Số cần được điền vào chỗ trống lần lượt là:

A. 18, 91

B. 81, 19

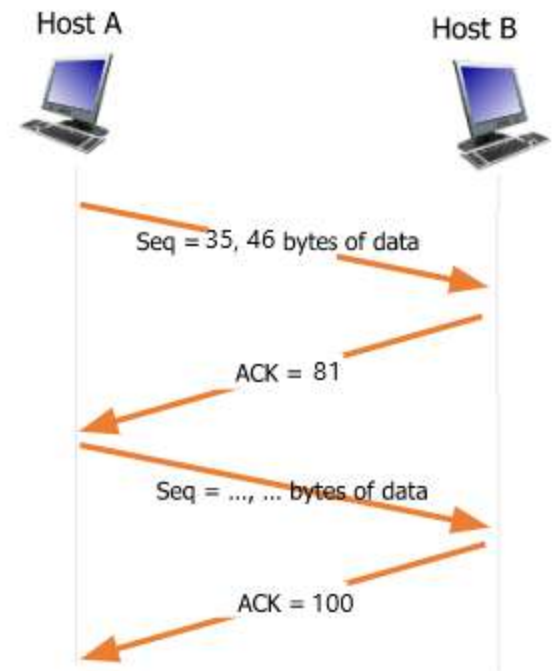
C. 100, 81

D. 81, 100

25. Đáp án B

Seq = 81

Bytes of data = ACK - Seq = 100 - 81 = 19



26. Trong các định dạng header của segment UDP có 4 :

A. Source port, destination port, length, checksum

B. Source port, destination port, head length, checksum

C. Source port, destination port, sequence number, acknowledgement number

D. Source port, destination port, sequence number, receive window

26. Đáp án A

27. Điều gì là đúng đối với các giao thức dạng connectionless (không kết nối)?

A. Hoạt động chậm hơn các giao thức dạng connection-oriented

B. Các gói dữ liệu có phần header phức tạp hơn so với giao thức dạng connection-oriented

C. Cung cấp một dịch vụ phân phát dữ liệu không đáng tin cậy

D. Nút gửi phải truyền lại những dữ liệu đã bị mất trên đường truyền

27. Đáp án C

Các tính chất còn lại là của giao thức dạng connection-oriented so với connectionless

28. Bên gửi gửi 1 TCP segment có Sequence Number = 76, và phần Payload (data) = 9 bytes. Bên nhận sẽ trả lời với Acknowledgement Number là bao nhiêu để báo nhận thành công TCP segment này?

A. 9

B. 76

C. 85

D. 57

28. Đáp án C

ACK = Seq liền trước + Payload liền trước = 85

Seq = ACK gói tin liền trước

29. Phiên bản RDT nào hỗ trợ những thành phần sau: Checksum, ACK/NAK, Sequence Number?

A. RDT 1.0

B. RDT 2.1

C. RDT 2.2

D. RDT 3.0

29. Đáp án B

Rdt 2.1:

- + *Bên gửi truyền lại packet nếu ACK/NAK bị hỏng*
- + *Bên gửi thêm số thứ tự (0,1) vào mỗi packet được gửi*
- + *Bên nhận hủy packet bị trùng lặp*

30. Một trình duyệt muốn gửi một yêu cầu HTTP GET đến một Web Server. Segment chứa HTTP GET có port đích là 80, vậy giá trị nhỏ nhất của port nguồn mà TCP có thể dùng trong trường hợp này là:

A. 1

B. 128

C. 256

D. 1024

30. Đáp án D

Cổng, port, là một con số 16 bit, có giá trị từ 0 đến 65535. Port có giá trị từ 0 đến 1023, được gọi là cổng phổ biến (well-known port number), bị giới hạn sử dụng; nghĩa là các cổng này được dành riêng cho các giao thức ứng dụng phổ biến như HTTP (cổng 80), FTP (cổng 21), và DNS (cổng 53). Còn các port có số cao hơn (từ 1024 trở lên) thì lại thường được sử dụng làm cổng tạm cho các dịch vụ ít phổ biến hơn.