

## ĐÁP ÁN CHI TIẾT ĐỀ SỐ 5

**Câu 1:** ⑤

**Câu 2:** Giảm lưu lượng truyền thông – do chỉ lấy về những đối tượng mới hơn

**Câu 3:** ③      **Câu 4:** ①

**Câu 5:**

Sender gửi:  $[0,1,2] \times 1(\text{lần}) + [3,4] \times 3(\text{lần}) = 9 \text{ segment}$

Receiver xác nhận các gói theo thứ tự:  $[0, 1, 2, 4, 4, 3, 4] = 7 \text{ ACK}$

Vậy cả A và B đã gửi:  $9+7 = 16 \text{ segment}$

**Câu 6:**

Sender gửi:  $[0, 1, 2] \times 1(\text{lần}) + [0] \times 1(\text{lần}) + [3] \times 3(\text{lần}) = 7$

segment Receiver xác nhận các gói theo thứ tự:  $[0,1,2, 4, 3] = 5$

ACK

Vậy cả A và B đã gửi:  $7+5 = 12 \text{ segment}$

**Câu 7:** ③

**Câu 8:**

Tổng số segment cần gửi là 100  $\rightarrow$  B cần gửi 50 ACK

Số segment bị lỗi gửi lần đầu = 10  $\rightarrow$  B phải gửi thêm  $10 \times 2 = 20 \text{ ACK}$  cho các segment lỗi này. Vậy tổng số ACK B phải gửi đi là  $50 + 20 = 70 \text{ ACK}$

**Câu 9:** Congwin = 19

**Câu 10:** Chuỗi nhị phân của các ký tự trong đoạn text “CAT” là

$C(67) = 01000011$   $A(65) = 01000001$   $T(84) = 01010100$

Vậy UDP Checksum của đoạn text trên là

0100001101000001

0101010000000000

-----

1001011101000001  $\rightarrow$  Đảo bit có được UDP Checksum: 0110100010111110

**Câu 11:** Flowcontrol trong TCP sử dụng **Receiver Window size** trong TCP header segment

**Câu 12:**

Địa chỉ host A = 152.15.90.58/27  $\rightarrow$  máy A thuộc mạng 152.15.90.32/27

Mạng chứa A còn 5 bit phần host  $\rightarrow$  Nếu muốn tiếp tục chia subnet cho mạng này thì tối đa mượn được 3 bits  $\rightarrow$  chia được tối đa thành 8 mạng có địa chỉ:

152.15.90.32/30

152.15.90.48/30

152.15.90.36/30

152.15.90.52/30

152.15.90.40/30

152.15.90.56/30

152.15.90.44/30

152.15.90.60/30

**Câu 13:**

- Phân đoạn mạng 1 (MTU=1500) cần chuyển 45000 bytes data → phải chia thành 31 datagram (d1, d2, d3, ..., d30 chuyển được  $1480 \times 30 = 44400$  byte data; d31 chuyển nốt 600 bytes data cuối cùng).

- Ở phân đoạn mạng 2 (MTU=1200) mỗi datagram d1, d2, d3, ..., d30 bị chia thành 2 datagram nhỏ hơn là d11, d12, d21, d22, d31, d32, ..., d301, d302 trong đó:

- + d11, d21, d31, ..., d301 mỗi datagram chuyển 1180 byte data.
- + d12, d22, d32, ..., d302 mỗi datagram chuyển 300 byte data
- + d31 khi đi qua phân đoạn này không bị phân mảnh (vì chỉ chứa 600 byte < 1200)

**Vậy:** B nhận tổng cộng **61** datagram, datagram thứ 8 chứa **300** byte dữ liệu

#### Câu 14:

$A \rightarrow C \rightarrow D \rightarrow F$  (Giá trị:  $1 + 3 + 2 = 6$ ) hoặc  $A \rightarrow C \rightarrow F$  (giá trị =  $1 + 5$ )

*Lưu ý: Sinh viên phải trình bày bảng tính toán các bước thực hiện giải thuật Dijkstra*

#### Câu 15:

D <sub>A</sub>	B	C
B	(2)	9
C	6	(5)
D	(3)	10
E	(3)	8
F	(5)	12

#### Câu 16: ②

**Câu 17:** Mã của các ký tự trong chữ “CUTE” là 67-85-84-69.

Chuỗi nhị phân tương ứng : **01000011 01010101010100 01000101**

Ma trận kiểm tra chẵn lẻ 7x7

0	1	0	0	0	0	1
1	1	0	1	0	1	0
0	1	0	1	0	1	1
0	1	0	1	0	0	0
0	1	0	0	0	1	0
0	1	0	0	0	0	1
1	0	0	1	0	1	1

**Câu 18:** Mã ASCII của “c” là 99 → Mã nhị phân là: 01100011

→ mã Hamming của ký tự “c” là: **000111000011**

*Lưu ý: nếu SV áp dụng luật số lẻ thì mã hamming là **110011010011***

**Câu 19:** Mã hamming lỗi 1 bit nhận được **10101110110**

Xét vị trí 1: có 5 bit 1 tại các vị trí 1,3,5,7,9,11 → lỗi (bit parity =1)

- Xét vị trí 2: có 4 bit 1 tại các vị trí 2,3,6,7,10,11 → Không lỗi (bit parity =0)

- Xét vị trí 4: có 3 bit 1 tại các vị trí 4, 5,6,7 → lỗi (bit parity =1)

- Xét vị trí 8: có 2 bit 1 tại các vị trí 8, 9, 10, 11 → Không lỗi (bit parity =0)

Chuỗi nhị phân vị trí bit bị lỗi là 0101 → vậy bit số 5 đảo lại thành 0

Chuỗi nhị phân sửa lại là: **10100110110**

Mã nhị phân của ký tự bên gửi là: 1011110, không phải ký tự từ A-z

**Câu 20:**

Mã của các ký tự trong chữ “CUTE” là 67-85-84-69.

Chuỗi nhị phân tương ứng : 01000011 01010101 01010100 01000101

Vậy D: 01000011010101010101001000101 hoặc 0x 43555445

G = 10101, r=4 vậy

→Thực hiện thuật toán tính CRC sẽ thu được: R = 1111

**Phần II – trả lời tự luận**

Nhận định “*tốc độ tải trang web tăng lên kể cả với những đối tượng chưa được đưa vào cache tại proxy server*” là sai. Tốc độ chỉ tăng nếu đối tượng đó đã có sẵn trên proxy-server (Client load đối tượng thường với tốc độ của Ethernet LAN).

- + Ưu điểm của Proxy server: Tăng tốc độ tải đối tượng nếu các đối tượng này đã được cache tại server. Giảm thông lượng đường truyền tới original server
- + Nhược điểm: Nếu Proxy mà “xa” client hơn cả original server thì cho kết quả ngược lại. Đối với những đối tượng chưa từng được cache trên proxy → tốc độ load về client sẽ chậm hơn nếu load trực tiếp từ original server.