

Computer Networking Home Exam #03

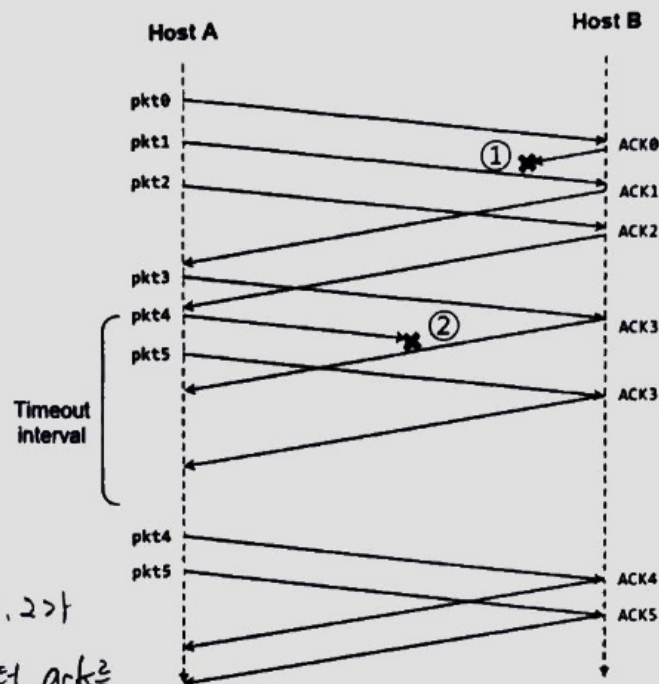
Date	Sep. 30 th , 2019	Instructor	Yoo, Younghwan
Student ID	201724557	Name	장 수현.

1. UDP and TCP use the 1's complement for their checksums. Suppose you have the following three 8-bit bytes: 01010011, 01100110, 01110100. What will be the checksum of these 8-bit bytes? (Note that although UDP and TCP use 16-bit words in computing the checksum, for this problem you are being asked to consider 8-bit sums.) (20 pts)

$$\begin{array}{rcl}
 \text{i)} & \begin{array}{r} 01010011 \\ + 01100110 \\ \hline 10111001 \end{array} & \rightarrow \text{ii)} \begin{array}{r} 10111001 \\ + 01110100 \\ \hline 10010101 \end{array} \rightarrow \text{iii)} \begin{array}{r} 10010101 \\ \Rightarrow 00010110 \\ \uparrow \\ = \text{Sum} \end{array}
 \end{array}$$

$$\therefore \text{checksum} = 111010001$$

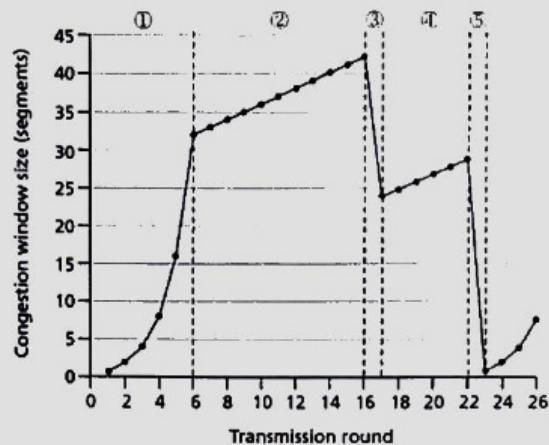
2. Suppose Host A and Host B use a Go-Back-N protocol with window size $N = 3$ and a long-enough range of sequence numbers. Explain how host A handled the situation of packet errors ① and ② in the right figure. (20 pts)



①) 처음에 window에 packet 0, 1, 2가 담겨서 전송. 이후 receiver로부터 ack를 받게 되면 window를 옮기는데, cumulative ack 방식이므로 ack0이 도착하지 않았지만 ack1을 받게되면 pkt0도 잘 받았다고 간주한다. 그래서 window는 두 칸 옮겨서 다른 패킷도 전송한다

②) 패킷 4를 보냈는데 중간에 에러가 나서 Host B까지 도달하지 못했다. 이후 패킷 5는 똑바로 전송이 되었지만 Host B는 다음 순서에 올 패킷 4가 없음을 알고 처음에 패킷을 제대로 받은 3에 대한 ack를 계속해서 보낸다. 그리고 패킷 4에 대한 타임아웃이 지나가면 다시 패킷 4를 전송하여 순서를 바로 잡는다. 이때 그전에 받은 Pkt 5는 버린다.

3. The right figure shows the change in the size of the congestion window every RTT round in a system using TCP Reno. Explain each situation ①~⑤ and its cause. (30 pts)



- ①) 시작은 slow start 이다 처음엔 window size를 1로 시작해서 congestion이 발생하지 않으면 계속해서 두 배씩 window size를 늘려나간다
- ②) ssthresh에 도달하게 되면 1MSS씩 증가시켜보면서 전송 가능한 상황인지 확인
- ③) congestion을 만났을때 fast retransmit 일 경우엔 초기화 하는 것이 아니라 사이즈를 절반으로만 감소시킨다.
- ④) 감소된 사이즈부터 다시 1MSS씩 증가시켜보면서 사이즈를 측정한다.
- ⑤) 또 다시 congestion을 만난 경우인데, 이번엔 fast retransmit이 아니라 time out으로 인한 것이기 때문에 window size를 초기화 하고 다시 slow start를 진행한다.

4. Consider that only a single TCP (Reno) connection uses one 10Mbps link. Suppose that this link is the only congested link between the sending and receiving hosts. Assume that the TCP sender has a huge file to send to the receiver, and the receiver's receive buffer is much larger than the congestion window. We also make the following assumptions: each TCP segment size is 1,500 bytes; the two-way propagation delay of this connection is 150 msec; and this TCP connection is always in the congestion avoidance phase, that is, ignore the slow start. (30 pts)

- 1) What is the maximum window size (in segments) that this TCP connection can achieve?

$$\text{윈도우의 메시지 크기} = \frac{\text{링크 전송률} \times \text{RTT}}{\text{세그먼트 크기}}$$

$$\therefore 125$$

- 2) What is the average window size (in segments) and average throughput (in bps) of this TCP connection?

$$\begin{aligned} \text{i) 윈도우 (최대 크기} &= 125 \text{ (4-1)의 M 포함)} \\ \text{최소 크기} &= 125/2 = 62.5 \\ \Rightarrow \text{평균 크기} &= 93.75 \end{aligned}$$

$$\begin{aligned} \text{ii) 평균 처리량} &= \text{평균 윈도우 크기} \times \text{세그먼트 크기} \times \text{RTT} \\ &\Rightarrow 7.5 \text{ Mbps} \end{aligned}$$