3장 과제

수학과 202021224 주민찬

P4. Assume that a host receives a UDP segment with 01011101 11110010 (we separated the values of each byte with a space for clarity) as the checksum. The host adds the 16-bit words over all necessary fields excluding the checksum and obtains the value 00110010 00001101. Is the segment considered correctly received or not? What does the receiver do?

- 두 값을 더하면 10001111 11111111입니다. 이 값이 11111111 11111111가 아니어서 올바르게 수신되지 않았습니다. 따라서 수신자는 이 값을 폐기할 것입니다.

P15. Consider the cross-country example shown in Figure 3.17, with a 10 Gbps link. How big would the window size have to be for the channel utilization to be greater than 98 percent? Suppose that the size of a packet is 1,500 bytes, including header fields and data.

- $\frac{1500 \times 8 bits}{10^9 bits/\mathrm{sec}ond} = 0.000012\mathrm{sec}ond = 12\mu s$, RTT를 100ms로 가정하면

$$0.98 = \frac{0.012ms \times n}{100.012ms} \quad \Rightarrow \quad n = \frac{0.98 \times 100.012}{0.012} \approx 8168$$

P26. Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS of 536 bytes.

a. What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.
b. For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.

- a. $2^{32} = 4{,}294{,}967{,}296 bytes \qquad L = \left\lceil \frac{2^{32}}{536} \right\rceil \times 536 = 4{,}294{,}967{,}464 \approx 4GB$
- b. $\frac{2^{32}}{536} \approx 8{,}012{,}999 \qquad 8{,}012{,}999 \times 602 = 4{,}823{,}825{,}398 bytes$

$$155Mbps = 155 \times 10^6 bits/\mathrm{sec}ond = \frac{155 \times 10^6}{8} bytes/\mathrm{sec}ond = 19{,}375{,}000 bytes/\mathrm{sec}ond$$

$\frac{4{,}823{,}825{,}398}{19{,}375{,}000} \approx 249$ 약 249초

P28. Host A and B are directly connected with a 10 Gbps link. There is one TCP connection between the two hosts, and Host A is sending to Host B an enormous file over this connection. Host A can send its application data into its TCP socket at a rate as high as 1 Gbps, but Host B can read out of its TCP receive buffer at a maximum rate of 600 Mbps. Describe the effect of TCP flow control.

- Host B의 수신 처리 속도에 맞추어 Host A의 송신 속도를 조절합니다. Host A는 1 Gbps로 데이터를 보내려고 하지만, Host B는 최대 600 Mbps만 처리할 수 있기 때문에, 실제 전송 속도는 600 Mbps로 제한됩니다. Host B에 의해 Host A의 전송 속도가 조절되어 데이터 손실을 방지하고 안정적으로 전송이 이루어지게 됩니다.

P32. Consider the TCP procedure for estimating RTT. Suppose that α = 0.1. Let SampleRTT1 be the most recent sample RTT, let SampleRTT2 be the next most recent sample RTT, and so on.

a. For a given TCP connection, suppose four acknowledgments have been returned with corresponding sample RTTs: SampleRTT4, SampleRTT3, SampleRTT2, and SampleRTT1. Express EstimatedRTT in terms of the four sample RTTs.
b. Generalize your formula for n sample RTTs.
c. For the formula in part (b) let n approach infinity. Comment on why this averaging procedure is called an ㅊ

- a. $EstimatedRTT^{(4)} =$
$\alpha SampleRTT_1 + (1-\alpha)[\alpha SampleRTT_2 + (1-\alpha)[\alpha SampleRTT_3 + (1-\alpha)SampleRTT_4]]$
$= \alpha SampleRTT_1 + (1-\alpha)\alpha SampleRTT_2 + (1-\alpha)^2 \alpha SampleRTT_3$
$+ (1-\alpha)^3 SampleRTT_4$

- b. $EstimatedRTT^{(n)} = \alpha \sum_{i=1}^{n-1} (1-\alpha)^{i-1} SampleRTT_i \ + (1-\alpha)^{n-1} SampleRTT_n$

- c. $\lim_{n \to \infty} (1-\alpha)^n = 0$ 따라서 $EstimatedRTT^{(n)} = \alpha \sum_{i=1}^{\infty} (1-\alpha)^{i-1} SampleRTT_i$

'exponential moving average'라고 불리는 이유는 무한대로 가면 오래된 RTT 값들의 가중치가 0에 가까워져 최신 RTT 값이 큰 영향을 미친다.

P40. Consider Figure 3.61. Assuming TCP Reno is the protocol experiencing the behavior shown above, answer the following questions. In all cases, you should provide a short discussion justifying your answer.

a. Identify the intervals of time when TCP slow start is operating.
b. Identify the intervals of time when TCP congestion avoidance is operating.
c. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?
d. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

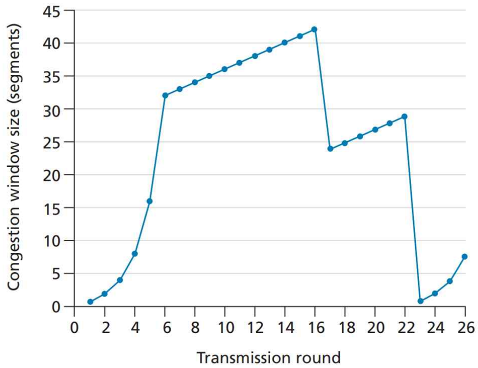

Figure 3.61 ♦ TCP window size as a function of time

- a. 구간 [1,6]와 구간 [23,26]
- b. 구간 [6,16]와 구간 [17,22]
- c. triple duplicate ACK
- d. timeout