Answers for Chapter 3 Assignments

P13.

Consider a reliable data transfer protocol that uses only negative acknowledgments. Suppose the sender sends data infrequently. Would a NAK-only protocol be preferable to a protocol that uses ACKs? Why? Now suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this second case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

Solution:

In a NAK only protocol, the receiver receives packet x-1 and packet x+1 consecutively, and then it knows packet x was lost. If data are sent infrequently, the loss of packet will be found after the long interval. So a NAK-only protocol is not preferred.

If there are a lot of data to send, and the interval is very short, it responds very fast. And meanwhile, NAK packets are only sent when error occurs, and no ACK packets are needed so that the feedback procedure would be simplified. So a NAK-only protocol is preferred.

P23.

We have said that an application may choose UDP for a transport protocol because UDP offers finer application control (than TCP) of what data is sent in a segment and when.

a. Why does an application have more control of what data is sent in a segment?

b. Why does an application have more control on when the segment is sent?

Solution:

a) In TCP, data are sent with fixed-length buffer, which means in a buffer there might be part of one message. But in UDP, data are sent with a whole message, which means it sends a complete message regardless of its length. So we know what's in a UDP data segment.

b) In TCP, due to flow control and congestion control, there may be significant delay. It's from the time when an application writes data to its send buffer to the time when the data is given to the network layer. UDP does not have those delays. Thus, in UDP we know better when the segment is sent.

P34.

Compare GBN, SR, and TCP (no delayed ACK). Assume that the timeout values for all three protocols are sufficiently long such that 5 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A) respectively. Suppose Host A sends 5 data segments to Host B, and the 2^{nd} segment (sent from A) is lost. In the end, all 5 data segments have been correctly received by Host B.

a. How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.

b. If the timeout values for all three protocols are much longer than 5 RTT, then which protocol success fully delivers all five data segments in shortest time interval?

Solution:

a).

GBN:

A sends 9 segments: 1 2 3 4 5 2 3 4 5

B sends 8 ACKs: 1 1 1 1 2 3 4 5

SR:

A sends 6 segments: 1 2 3 4 5 2

B sends 5 ACKs: 1 3 4 5 2

TCP:

A sends 6 segments: 1 2 3 4 5 2

B sends 5 ACKs: 2 2 2 2 6

b). TCP.

TCP uses fast retransmit without waiting until time out.

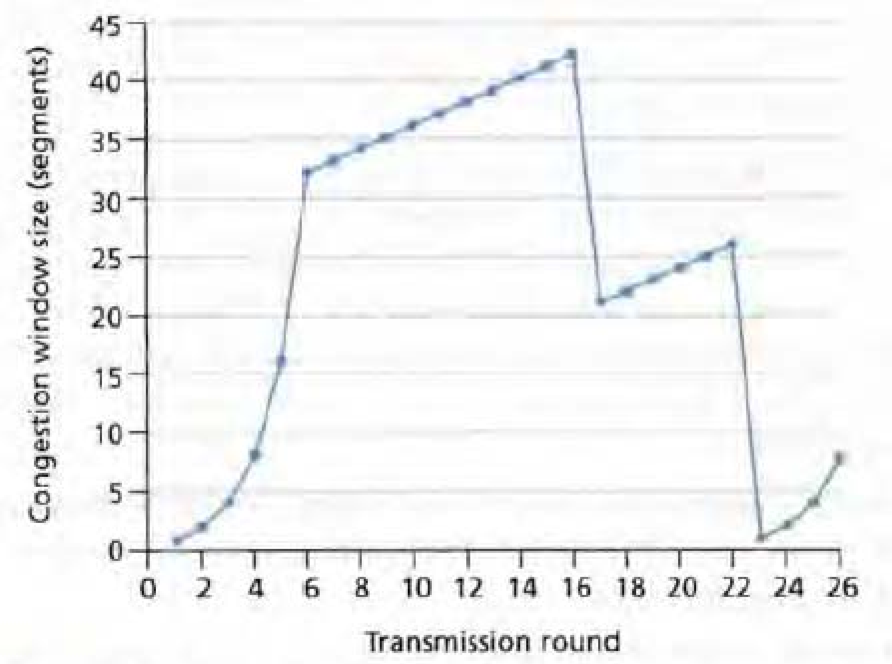

Figure 3.58 • TCP window size as a function of time

P37.

Consider Figure 3.58. 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.

Solution: [1, 6] and [23, 26]

b. Identify the intervals of time when TCP congestion avoidance is operating.

Solution: [6, 16] and [17, 22]

c. After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

Solution: a triple duplicate ACK

d. After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

Solution: a timeout

e. What is the initial value of threshold at the first transmission round?

Solution: 32

f. What is the value of threshold at the 18th transmission round?

Solution: 42/2=21

g. What is the value of threshold at the 24th transmission round?

Solution: 26/2=13

h. During what transmission round is the 70th segment sent?

Solution: at 6th round, it has sent 1+2+4+8+16+32=63 segments

At 7th round, it sends 33 segments from 64, including 70th segment.

i. Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of threshold?

Solution: 8/2=4

j. Suppose TCP Tahoe is used (instead of TCP Reno), and assume that triple duplicate ACKs are received at the 16th round. What are the threshold and the congestion window size at the 19th round?

Solution: threshold=42/2=21

Congestion window size=1*2*2=4

k. Again suppose TCP Tahoe is used, and there is a timeout event at 22nd round. How many packets have been sent out from 17th round till 22nd round, inclusive?

Solution: 1+2+4+8+16+21=52

P40.

Host A is sending an enormous file to Host B over a TCP connection. Over this connection there is never any packet loss and the timers never expire. Denote the transmission rate of the link connecting Host A to the Internet by *R* bps. Suppose that the process in Host A is capable of sending data into its TCP socket at a rate S bps, where S = 10 *R*. Further suppose that the TCP receive buffer is large enough to hold the entire file, and the send buffer can hold only one percent of the file. What would prevent the process in Host A from continuously passing data to its TCP socket at rate S bps? TCP flow control? TCP congestion control? Or something else? Elaborate.

Solution:

1. Receiving buffer is large enough, so it’s not for TCP flow control.
2. No packet loss and timeout, so it’s not for TCP congestion control.
3. Sending buffer is limited sized, so as soon as sending buffer is full, the transmission rate drops.