Name: Rajat Rajesh Shetty

CWID:10477784

Assignment 7

16. Suppose an application uses *rdt 3.0* as its transport layer protocol. As the stop-and-wait protocol has very low channel utilization (shown in the cross-country example), the designers of this application let the receiver keep sending back a number (more than two) of alternating ACK 0 and ACK 1 even if the corresponding data have not arrived at the receiver. Would this application design increase the channel utilization? Why? Are there any potential problems with this approach? Explain.

- Suppose an application uses rdt 3.0 as its transport layer protocol. Then It stop-and-wait protocol is used very low utilization in the application.
- The application allows the receiver to send the acknowledgements and sends the next data packet. It is used as a pipelined data in the channel.
- There is a chance of missing data before it reaches the receiver.
- So the sender (who is using rtd 3.0 protocol) will not retransmit the data. The missing or lost some data.
- Thus, designing of the application need to adopt some mechanism to over come this problem.

24. Answer true or false to the following questions and briefly justify your answer:

a. With the SR protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.

b. With GBN, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.

c. The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1.

d. The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1.

24) a) True. suppose the sender hal a window size of 3 & sends packets 1,2,3 at t0. At t1 (t1>t0) the reciever ACKs 1,2,3. At t2 ((t2>t1) the sender times out & resendr 1,2,3. At t3 the reciever recieves the duplicates & re acknowledges 1,2,3. At t4 the sender recieves the ACKs that the reciever sent at t1 & advances its window to 4,5,6. At t5 the sender recieves the ACK 1,2,3 the reciever sent at t2. There ACKs are outside its window.

b) True, By essentially the same scenario as in (a)

c) True

d) True. Note that with window size of 1, SR, GBN & the alternating bit protocol are funtionally equivalent. The window size of 1 precludes the possibility of out-of-order packets. A cumulative ACK is just an orginally ACK in the situatn, since it can only refer to the single packet within the window.

26. 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.

26)

a) The size of TCP sequence no. field is 4 bytes

$= 4 \times 8$ bits

$= 32$ bits

so the sequence no. of bits are 2^{32}.

The max file sent from host A to host B representable by 2^{32}.

$\approx 2^2 \times 2^{30}$ bytes

$= 2^2$ Gbytes

$= 4$ Gbytes

b) max segment size (MSS) = 536 bytes

segments data $= 2^{32}/536$

$= 8012999$

Total header field = 66 bytes

Total no of bytes through the 155Mbps link $= 8012999 \times 66$ bytes

$= 528857934$ bytes

Transmitted data $= (2^{32} + 528857934)$

$= 4.824 \times 10^9$ bytes

Transmit time $= \dfrac{4.824 \times 10^9 \times 8 \text{ bits}}{155 \times 10^6 \text{ bps}} \approx \boxed{249}$ seconds

Thus it would take 249 seconds to transmit the file over 155 Mbps link.

27. Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.

a. In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?

b. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?

c. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?

d. Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number.

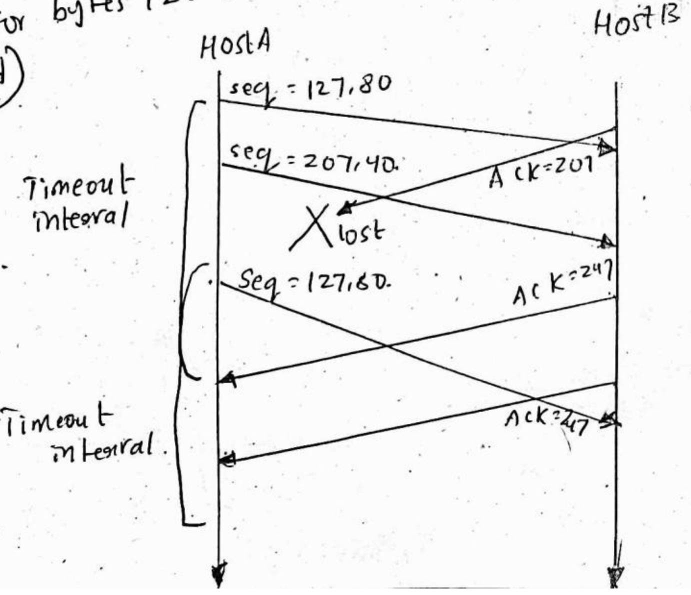
27)

a) In the second segment from host A to B, the sequence no is 207, source port no is 302 & destination port no is 80.

b) if the first segment arrives before the second, in the acknowledgemont no is 207, the source port no is 80 & the destinat port no is 302.

c) if the second segment arrives before the first segment, the in the acknowledgement of the first arriving segment, acknowledgement no is 127, indicating that it is still waiting for bytes 127 & onwards.

d)



28. Host A and B are directly connected with a 100 Mbps 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 120 Mbps but Host B can read out of its TCP receive buffer at a maximum rate of 50 Mbps. Describe the effect of TCP flow control.

(28) Since the line capacity is only 100 Mbps, so host A's sending rate can be at most 100 Mbps. Still host A sends data into the reciever's buffer faster than Host B can remove data from the buffer. The reciever buffer fill up at a rate of roughly 40 Mbps. When the buffer is full, Host B signals to Host A to stop sending data by setting Rcvwindow =0. Host A then stops sending until it recieves a TCP segment with Rcwindow > 0. Host A will thus repeatedly stop & start sending as a function of the Rcvwindow values it recieves from Host B. On average, the long-term rate at which Host A sends data to Host B as part of this connection is no more than 60 Mbps.

34. What is the relationship between the variable *SendBase* in **Section 3.5.4** and the variable *LastByteRcvd* in **Section 3.5.5** ?

Relationship between the variables, SendBase and LastByteRcvd:

SendBase-1 <=LastByteRcvd

Detailed Explanation:

- SendBase-1 is used to find the ***sequence number*** of the last byte(Refer Section 3.5.4 from the text book)
- LastByteRcvd is used to find the **number of the last byte** in the stream of data arrival from network to buffer. So, SendBase is the LastByteRcvd at the receiver end.

35. What is the relationship between the variable *LastByteRcvd* in **Section 3.5.5** and the variable y in Section **3.5.4**?

35) when at time t, the sender recieves an acknowledgment with value y, the sender knows for sure that the reciever has recieved everything up through y-1. The actual last byte recieved at the reciever at time t may be greater if y ≤ sendbase or if there are other acknowledgments in the pipe.

$y - 1 \leq$ LastByteRcvd.