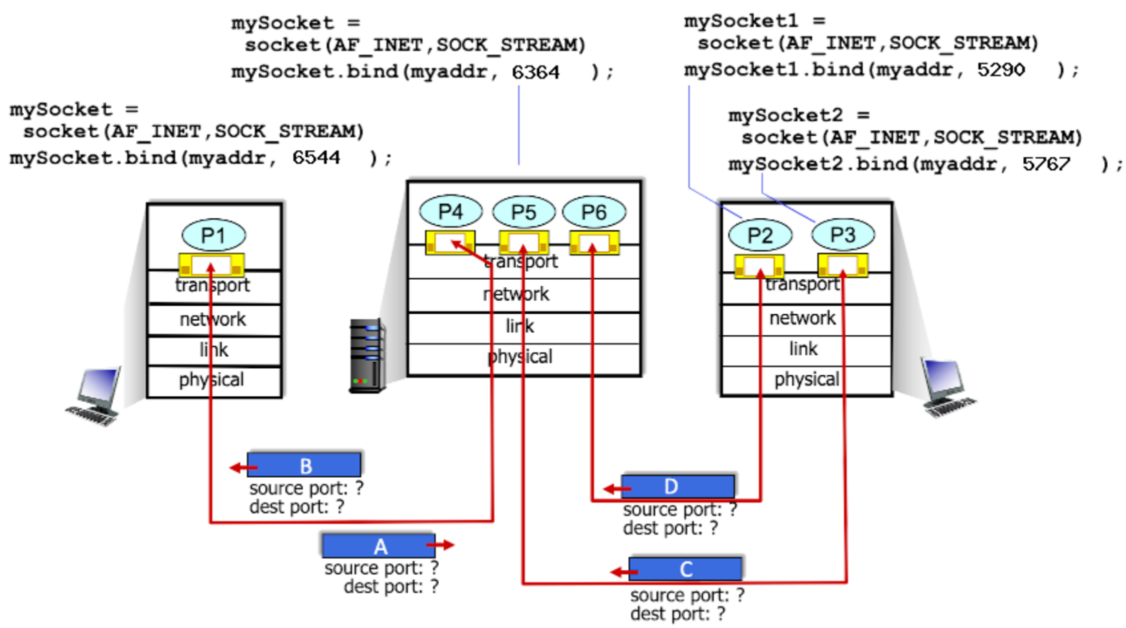


QUESTION LIST

1. What is the source port # for packet A?

2. What is the destination port # for packet A?

3. What is the source port # for packet C?

4. What is the destination port # for packet C?

5. What is the source port # for packet D?

6. What is the destination port # for packet D?

7. What is the source port # for packet B?

8. What is the destination port # for packet B?

SOLUTION

1. The source port for packet A is port 6544.

2. The destination port for packet A is port 6364.

3. The source port for packet C is port 5767.

4. The destination port for packet C is port 6364.

5. The source port for packet D is port 5290.

6. The destination port for packet D is port 6364.

7. The source port for packet B is port 6364.

8. The destination port for packet B is port 6544.

COMPUTING AN INTERNET CHECKSUM

Consider the two 16-bit words (shown in binary) below. Recall that to compute the Internet checksum of a set of 16-bit words, we compute the one's complement sum [1] of the two words. That is, we add the two numbers together, making sure that any carry into the 17th bit of this initial sum is added back into the 1's place of the resulting sum); we then take the one's complement of the result. Compute the Internet checksum value for these two 16-bit words:

00000101 00000001 *this binary number is 1281 decimal (base 10)*

01001110 00010010 *this binary number is 19986 decimal (base 10)*

QUESTION LIST

1. What is the sum of these two 16 bit numbers? Don't put any spaces in your answer

2. Using the sum from question 1, what is the checksum? Don't put any spaces in your answer

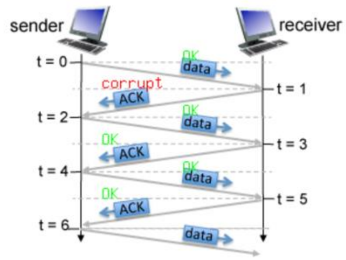
SOLUTION

1. The sum of 00000101 00000001 and 01001110 00010010 = 01010011 00010011

2. The internet checksum is the one's complement of the sum: 01010011 00010011 = 10101100 11101100

RDT 2.2

Suppose that the channel connecting the sender and receiver can corrupt but not lose or reorder packets. Now consider the figure below, which shows four data packets and three corresponding ACKs being exchanged between an rdt 2.2 sender and receiver. The actual corruption or successful transmission/reception of a packet is indicated by the corrupt and OK labels, respectively, shown above the packets in the figure below.



1. At time t=0, what is the sender state?

2. At time t=0, what is the receiver state?

3. At time t=0, what is the sequence/ack # of the packet?

4. At time t=1, what is the sender state?

5. At time t=1, what is the receiver state?

6. At time t=1, what is the sequence/ack # of the packet?

7. At time t=2, what is the sender state?

8. At time t=2, what is the receiver state?

9. At time t=2, what is the sequence/ack # of the packet?

10. At time t=3, what is the sender state?

11. At time t=3, what is the receiver state?

12. At time t=3, what is the sequence/ack # of the packet?

13. How many times is the payload of the received packet passed up to the higher layer?

SOLUTION

1. At time t=0, the sender state is: Wait for ACK 0

2. At time t=0, the receiver state is: Wait for 0 from below

3. At time t=0, the sequence # is: 0

4. At time t=1, the sender state is: Wait for ACK 0

5. At time t=1, the receiver state is: Wait for 1 from below

6. At time t=1, the ACK # is: 0

7. At time t=2, the sender state is: Wait for ACK 0

8. At time t=2, the receiver state is: Wait for 1 from below

9. At time t=2, the sequence # is: 0

10. At time t=3, the sender state is: Wait for ACK 0

11. At time t=3, the receiver state is: Wait for 1 from below

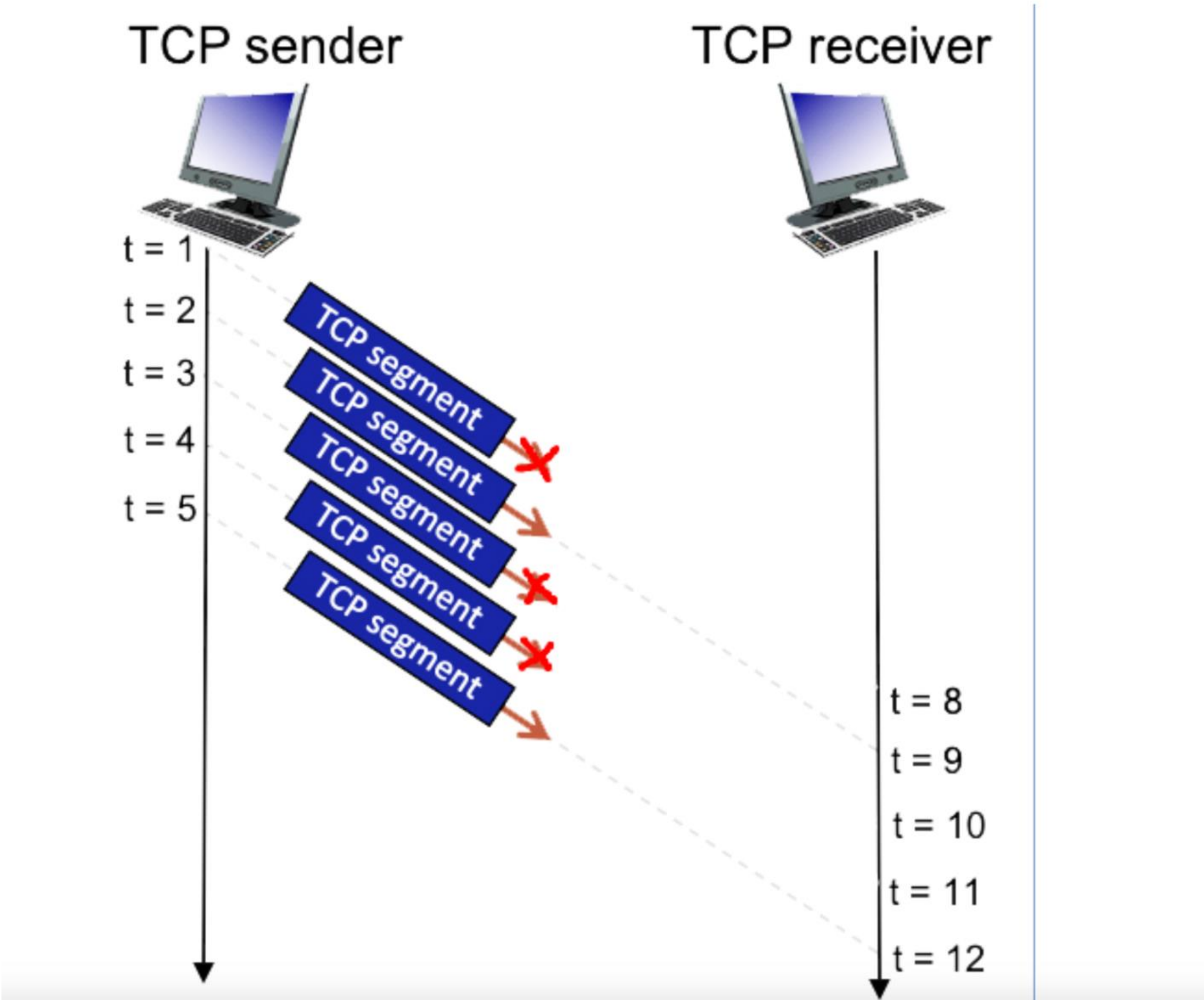
12. At time t=3, the ACK # is: 0

13. 2 packets were passed up to the higher layer by the receiver.

TCP Sequence number and ACK number

TCP SEQUENCE AND ACK NUMBERS WITH SEGMENT LOSS

Consider the figure below in which a TCP sender and receiver communicate over a connection in which the sender->receiver segments may be lost. The TCP sender sends an initial window of 5 segments. Suppose the initial value of the sender->receiver sequence number is 272 and the first 5 segments *each* contain 110 bytes. The delay between the sender and receiver is 7 time units, and so the first segment arrives at the receiver at t=8. As shown in the figure below, 3 of the 5 segment(s) are lost between the segment and receiver.



QUESTION LIST

1. Give the sequence numbers associated with each of the 5 segments sent by the sender. Format your answer as: a,b,c,...

2. Give the ACK numbers the receiver sends in response to each of the segments. If a segment never arrives use 'x' to denote it, and format your answer as: a,b,c,...

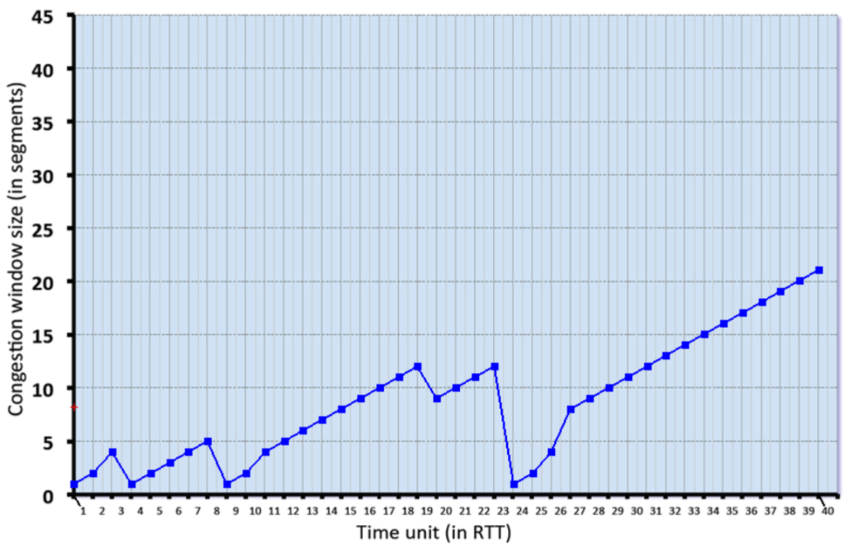
SOLUTION

1. The sender's sequence numbers are: 272,382,492,602,712

2. The receiver's ACKs are: x,272,x,x,272

TCP IN ACTION: SLOW START, CONGESTION AVOIDANCE, AND FAST RETRANSMIT

Consider the figure below, which plots the evolution of TCP's congestion window at the beginning of each time unit (where the unit of time is equal to the RTT); see Figure 3.53 in the text. In the abstract model for this problem, TCP sends a "flight" of packets of size *cwnd* at the beginning of each time unit. The result of sending that flight of packets is that either *(i)* all packets are ACKed at the end of the time unit, *(ii)* there is a timeout for the first packet, or *(iii)* there is a triple duplicate ACK for the first packet. In this problem, you are asked to reconstruct the sequence of events (ACKs, losses) that resulted in the evolution of TCP's *cwnd* shown below.



QUESTION LIST

1. Give the times at which TCP is in slow start. Format your answer like: 1,3,5,9 (If none submit blank)

2. Give the times at which TCP is in congestion avoidance. Format your answer like: 1,3,5,9 (If none submit blank)

3. Give the times at which TCP is in fast recovery. Format your answer like: 1,3,5,9 (If none submit blank)

4. Give the times at which packets are lost via timeout. Format your answer like: 1,3,5,9 (If none submit blank)

5. Give the times at which packets are lost via *triple ACK*. Format your answer like: 1,3,5,9 (If none submit blank)

6. Give the times at which the value of *ssthresh* changes (if it changes between t=3 and t=4, use t=4 in your answer)

SOLUTION

1. The times where TCP is in slow start are: 1,2,3,4,9,10,24,25,26

2. The times where TCP is in congestion avoidance are: 5,6,7,8,11,12,13,14,15,16,17,18,19,21,22,23,27,28,29,30,31,32,33,34,35,36,37,38,39,40

3. The times where TCP is in fast recovery are: 20

4. The times where TCP has a loss by timeout are: 3,8,23

5. The times where TCP has a loss by triple duplicate ACK are: 19

6. The times where the *ssthresh* changes are: 4,9,20

The complete solution is shown in the figure below:

- For intervals of time when TCP is in slow start, the plotted value of *cwnd* is shown as a green square
- For intervals of time when TCP is in congestion avoidance, the plotted value of *cwnd* is shown as a yellow square
- For intervals of time when TCP is in fast reccovery, the plotted value of *cwnd* is shown as an orange square
- The values for *ssthresh* are shown following a change as a red plus sign
- A flight of packets experiencing a loss has the loss type (which determines the next value of *cwnd*) labeled above