Class Test-2, Spring 2020-21
Computer Networks (CS31006)

Sample Solution

Students: 155
Date: 20-February-2021
Full marks: 30
Time: 60 minutes
Credit: 20%

INSTRUCTIONS: This is an OPEN-BOOK, OPEN-NOTES test. Please write your answers in a text file/.doc file, convert it to PDF, and submit this PDF file containing ONLY YOUR ANSWERS on Moodle. PLEASE DO NOT SUBMIT SCANNED HAND-WRITTEN ANSWERS, SUCH ANSWER-SCRIPTS WILL NOT BE GRADED. DO NOT FORGET TO WRITE YOUR NAME AND ROLL NUMBER AT THE TOP OF YOUR ANSWER SHEET. ANY DETECTED CASE OF PLAGIARISM WILL BE DEALT WITH STRICTLY, WITH ALL THE IMPLICATED STUDENTS RECEIVING ZERO IN THIS TEST. You may use calculators if required. This question paper contains two pages. ANSWER ALL QUESTIONS.

1. Consider two processes P1 and P2. Let there be two transport layer connections between P1 and P2. How do you separate out the transport layer segments between these two connections? Clearly write down the parameters that will help you to differentiate between the two connections. Which packet headers (application header, transport header, network header or data link header) will contain those parameters? [1 + 2 = 3]

Answer: At the transport layer, a connection is uniquely identified by six tuples – source IP, source port, source sequence number, destination IP, destination port and destination sequence number.

(There is also another factor, i.e protocol. One may use a single port to bind a UDP socket and TCP socket at the same time. Therefore, protocol can be also considered another parameter, marks will not be deducted if this is mentioned / not mentioned.)
(All the six tuples are necessary as the question does not mention whether the two processes are running on the same machine or on different machines. If some students mention just the port number, then give 0.5 marks)

In network layer header - source ip and destination ip, (and protocol) - 1 Mark
In transport layer header - source port , destination port and sequence number. - 1 Mark

2. We use connection establishment at the transport layer for synchronizing the initial sequence numbers between the two end hosts. However, this step may be avoided if both the ends agree on a fixed initial sequence number, say every connection starts from sequence number 1. Do you see any problem in this approach? Explain your answer. [2]

Answer: Random sequence number is required for every connection between the same hosts as the packets from a previous connection may get delayed and reach later (delayed duplicates), therefore may get overlapped with the next connection over the same port number. Therefore connections should have different initial sequence numbers. - 1 Mark

Further, a TCP sequence prediction attack is possible: an attempt to predict the sequence number used to identify the packets in a TCP connection, which can be used to counterfeit packets. - 1 Mark

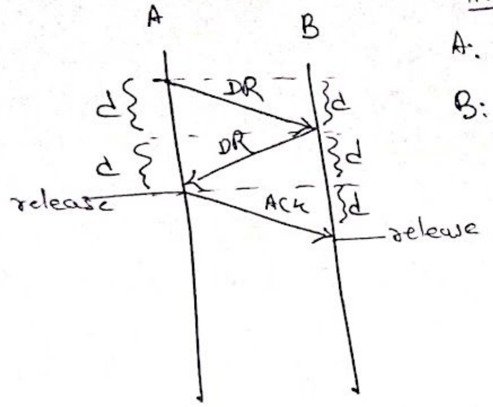
3. Consider two connected transport entities, A and B. A wants to terminate the connection, and hence, it initiates a connection release by sending a Disconnect Request (DR) segment to B. Assume that the two-parties implement the DR-DR-ACK mechanism of connection release taught in class. Further, assume that the one-way transmission delay between A and B to be a parameter d (in milliseconds), which remains constant for the entire session. The parameter d can be calculated from your roll number, by adding all the letters and digits, and then calculating the remainder modulo-10. The letters are assumed to have the following numerical values: A=1, B=2, · · · Z=26. For example, if your roll number is 15YZ12345, then d = (1 + 5 + 25 + 26 + 1 + 2 + 3 + 4 + 5) (mod 10) = 2 ms. Assume the delay to be symmetric in both the directions, and assume that segment processing and required actions happen instantaneously (incurring zero delay) at the two ends of the connection. Also, assume that the time-out interval for any segment sent by either party is 3d, and that unilateral connection release happens at A if there is no response after N = 5 timeouts following the initial sending of the DR segment, and that at B happens after a time interval 15d since the first response DR has been sent back to A. Given your roll number, calculate the time interval after the first DR segment is sent out by A, at which A and B will release the respective connections at their ends, in the following situation: (a) no segment sent by either party is lost; (b) the first DR segment sent by B in response to the DR segment received from A is only lost, and, (c) only the first DR segment sent from A to B is successfully delivered, every following segment from either parties is lost. [3+3+4=10]

Answer:

(a) A: 2d B: 3d
(b) A: 5d B: 6d
(c) A: 15d B: 16d

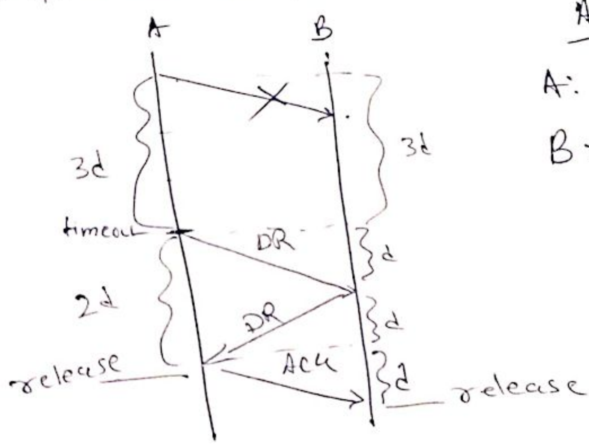
(a) no segment sent by each party is lost

Ans
A: 2d
B: 3d



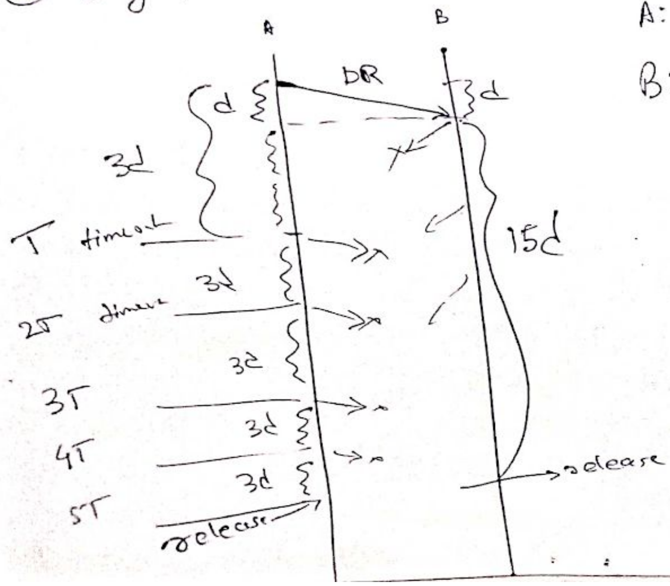
(b) first DR sent from A to B is lost

Ans
A: 5d
B: 6d



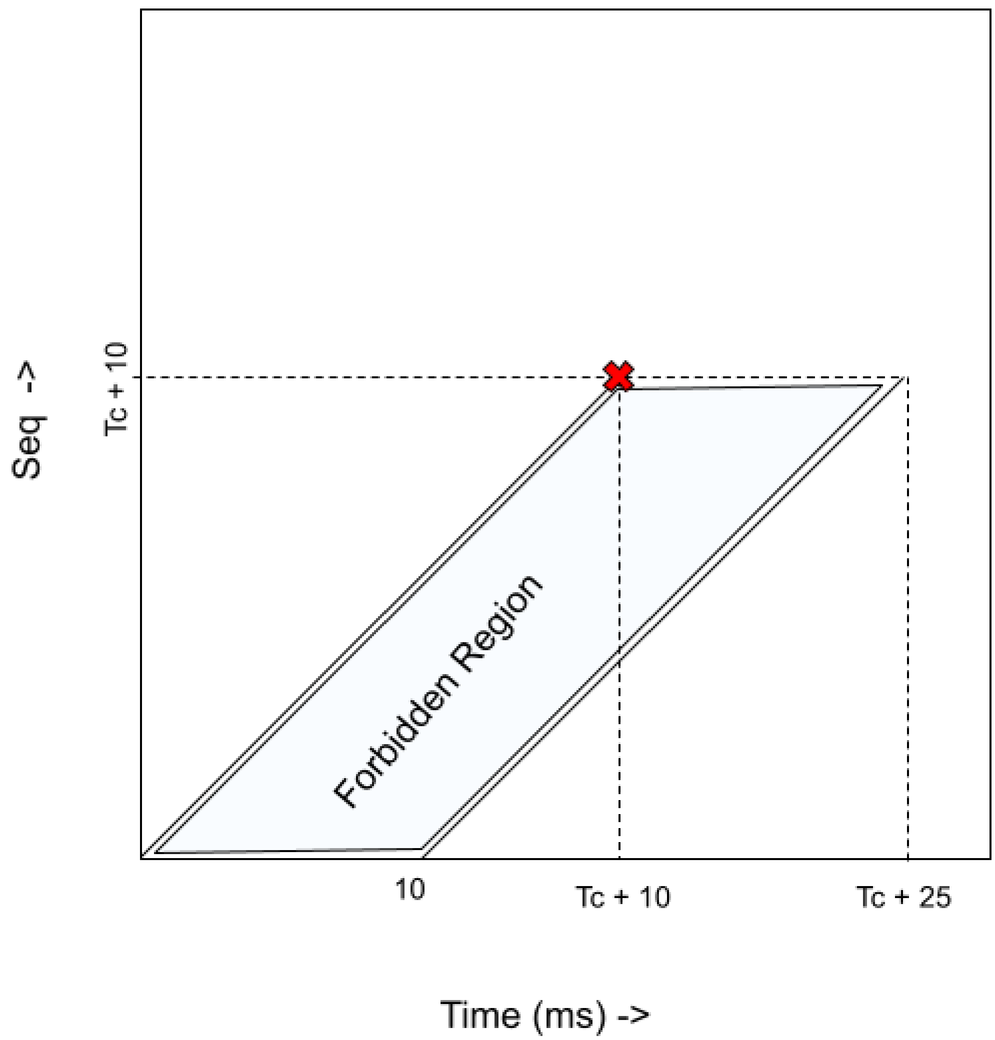
(c) Only first DR is sent. Rest lost.

Ans
A: 15d
B: 16d



4. Consider a connection in which segments are numbered starting from 0, and outgoing segments are generated by the sender at the rate of one segment per millisecond. The connection starts at time t = 0 ms, and the sender crashes at a time (Tc + 10) ms, where the numerical value of Tc can be calculated from your roll number in the same way that the parameter d was calculated in Question-3. Suppose, the expected lifetime of datagrams in the network between the sender and the receiver is 15 ms. Determine the coordinates of the vertices of the forbidden region on a plot of segment sequence no. (along vertical axis) vs. time (in ms, along horizontal axis). To simplify your analysis, ignore discretization of time. [3]

Answer:



5. Three-way handshaking ensures correctness during connection establishment; however it alone cannot ensure loss-free connection release – why? [2]

Answer: During connection establishment, the handshaking packets initializes the connection, and only after the connection is established, data transfer begins. Therefore, a loss in the connection establishment packets does not affect the data transmission directly (no data loss); only the connection cannot be established.

However, for connection release, the control packets (connection release messages) can be lost. Considering the "Two Army Problem", none of the two sides can ever ensure whether the packet has been reached successfully or not (note that the acknowledgement can also get lost -- so a transmitting end does not even guess whether the packet is lost or its acknowledgement is lost, in case it does not receive an acknowledgement. As the data packets can be delayed, therefore, a data packet may reach the destination after the destination has received a connection release message and closed the connection; in such cases, the data packet gets lost.

6. State whether each of the following statements is True or False, with a brief (1-2 sentence(s)) explanation in support of your answer:

(a) The rate at which segments are delivered to the application layer from the transport layer depends only on the datagram routing delay of the network layer. [2]
Answer: False
It also depends on Transport Buffer and out of order packets.

(b) One bit sequence number is sufficient for stop-and-wait ARQ. [2]
Answer: True
For stop and wait ARQ sender window size = receiver window size = 1

(c) The choice of sliding window protocols over stop-and-wait ARQ is always desirable. [2]
Answer: False
If a link cannot hold an entire segment completely, sliding window protocols do not improve performance.

(d) Timestamping every segment, while ensuring that not more than one segment is sent out per physical/logical clock-tick, is an effective solution to the delayed duplicate problem. [2]
Answer: False
Clock synchronization in the internet is very difficult especially to the granularity of sending rate of segments.

(e) Multiplicative Increase Additive Decrease (MIAD) is an effective alternative to the widely used AIMD technique for congestion control. [2]
Answer: False

In MIAD, the operating point does not converge to the theoretically optimal point. Instead, depending on the starting operating point, it diverges and settles at one of the two extreme operating points, allocating the entire bandwidth to the party which had more bandwidth initially, and zero bandwidth to the other.