
Computer Networks Class Test 2
Itish Agarwal
18CS30021

Q1.

Here, we can separate out the transport layer segments between these two connections by seeing the **port numbers** through which the connection is being established between the sender and the receiver. Parameters that help us differentiate between the two connections are:
-> initial sequence number
-> acknowledgement number
-> window size of the segments.

The **transport header** will contain those parameters.

Q2.

If both the ends agree on a fixed initial sequence number, (say 1), then there may arise a possibility of segments (or packets) from different connections getting mixed up.

For example, suppose we establish a TCP connection and send a segment containing bytes 1 through 60. However, there occurred a problem with the internet network that caused the segment to be delayed by some time period, and eventually, the TCP connection itself to be terminated.

What do we do now?

We have to again start up a new connection. Here, we again used a starting sequence number of 1. Now as soon as this connection is started, the old segment with bytes 1 through 60 showed up. This leads the other device to think those bytes were part of the new connection, which is not the case.

Q3

My roll number is **18CS30021**

1 -> 1

8 -> 8
C -> 3
S -> 19
3 -> 3
0 -> 0
0 -> 0
2 -> 2
1 -> 1
Hence d = (1 + 8 + 3 + 19 + 3 + 0 + 0 + 2 + 1) (mod 10)
= (37) (mod 10)
= 7
Now,

(a) A will release the connection after time interval = 2*d = **14**
B will release the connection after time interval: 3*d = **21**

(b) A will release the connection after time interval = 3d + 2d = 5*d = **35**
B will release the connection after time interval = 3d + 3d = 6*d = **42**

(c) A will release the connection after time interval = N * 3d = 5 * 3 * 7 = **105**
B will release the connection after time interval = d + 15*d = 16*d = 16 * 7 = **112**

Q4.
My roll number -> 18CS30021
tc = 7
Now, coordinates are = (0,0) , (15,0), (tc+10,tc+10), (tc+25,tc+10)
Hence,
Coordinates are (0, 0), (15, 0), (17, 17), (32, 17)

Q5.

Three-way handshaking ensures correctness during connection establishment. However, it alone cannot ensure loss-free connection release because in three way handshaking protocol the sender (say person A) and the receiver (say person B) both agree to establish a connection.

Now, if we implement the same protocol in releasing the connection then there will always remain a situation in which neither side is prepared to disconnect as it is not convinced that the other side is prepared to disconnect too. And if they decide to do so then it will lead to data loss and hence, the disconnection may never happen loss-free.

Q6.

(a) FALSE

It does not fully depend on the datagram routing delay of the network layer. Instead, it depends on the read() speed on the receiver's side. Also it depends on the availability of receiver buffer space.

(b) TRUE

Only one bit sequence is sufficient for stop-and-wait ARQ because only one bit is needed to separately identify each frame and the attached acknowledgement. This is because stop-and-wait protocol does not send the next segment until the previous one is acknowledged by the receiver.

(c) FALSE

Although sliding window protocols are more efficient than stop-and-wait, however there may be cases where the segment size is much more than the BDP (Bandwidth delay product). In such a case stop-and-wait will be more desirable than sliding window protocols.

(d) FALSE

This will not solve the problem as this method requires the router clocks to be synchronized, which itself is a non-trivial task as physical (ie, manual) synchronization of all clocks in the network is not possible.

(e) FALSE

MIAD is not an effective alternative to the widely used AIMD technique for congestion control as it not as it does not converge to the optimal point. Instead, it converges to one of the ends allotting full bandwidth to either of the users.