Spring Assignment

Q 1: Design a simple UDP-based protocol for retrieving files from a server. No authentication is to be provided. Stop-and-wait transmission of the data may be used. Your protocol should address the following issues: (a) Duplication of the first packet should not duplicate the “connection.” (b) Loss of the final ACK should not necessarily leave the server in doubt as to whether the transfer succeeded. (c) A late-arriving packet from a past connection shouldn’t be interpretable as part of a current connection.

Q 2. How is the TCP sequence number assigned to a packet? What is the size of the sequence number header in the TCP packet? Is this size large enough to prevent wrap around of the sequence numbers or not? If it doesn’t prevent wrap around then what problems can be caused by the potential wrap around of the sequence numbers for the TCP connection?

Q 3. Let's say that we want to prove and establish the reliability of a network link by sending packets and finding the percentage of the packets which were received on the other end; routers for example carry out this practice. Explain what problem will be faced while doing this with a TCP connection and what can be its remedy?

Q 4: Suppose TCP operates over a 40-Gbps STS-768 link. Assuming TCP could utilize the full bandwidth continuously, how long would it take the sequence numbers to wrap around completely?

Q 5: Suppose x and y are two TCP sequence numbers. Write a function to determine if x comes chronologically before y. Your solution should work even when sequence numbers wrap around.

Q 6: If a packet arrives at host A with B’s source address, it could just as easily have been forged by any third host C. If, however, A accepts a TCP connection from B, then during the three-way handshake A sent ISN A to B’s address and received an acknowledgment of it. If C is not located so as to be able to eavesdrop on ISNA, then it might seem that C could not have forged B’s response.However, the algorithm for choosing ISN A does give other unrelated hosts a fair chance of guessing it. Specifically, A selects ISN A based on a clock value at the time of connection. Assume that this clock value is incremented every 4μs

(a) Given this simplified increment-once-per-second implementation, explain how an arbitrary host C could masquerade as B in at least the opening of a TCP connection.You may assume that B does not respond to SYN+ACK packets A is tricked into sending to it.
(b) Assuming real RTTs can be estimated to within 40 ms, about how many tries would you expect it to take to implement the strategy of part (a) with the unsimplified “increment every4μs” TCP implementation?

Q 7. Suppose a TCP connection, with window size 1, loses every other packet. Those that do arrive have RTT = 1 second. What happens? What happens to TimeOut? Do this for two cases: (a) After a packet is eventually received, we pick up where we left off, resuming with EstimatedRTT initialized to its pre-timeout value, and TimeOut double that. (b) After a packet is eventually received, we resume with TimeOut initialized to the last exponentially backed-off value used for the timeout interval. In the following four exercises, the calculations involved are straightforward with a spreadsheet.

Q 8. FInd out and briefly explain how TCP is supposed to respond if a FIN or an RST arrives with a sequence number other than *NextByteExpected*. Consider both, when the sequence number is within the receive window and when it is not.

Q 9. TCP is a very symmetric protocol, but the client/server model is not. Consider an asymmetric TCP-like protocol in which only the server side is assigned a port number visible to the application layers. Client-side sockets would simply be abstractions that can be connected to server ports. Propose header data and connection semantics to support this. What will you use to replace the client port number?

Q 10. Consider this:
"*A host MAY implement a "half-duplex" TCP close sequence, so that an application that has called CLOSE cannot continue to read data from the connection. If such a host issues a CLOSE call while received data is still pending in TCP, or if new data is received after CLOSE is called, its TCP SHOULD send an RST to show that data was lost."*

Sketch a scenario involving the above in which data sent by (not to!) the closing host is lost. You may assume that the remote host, upon receiving an RST, discards all received data still unread in buffers.

Q 11. When TCP sends a {SYN,SequenceNum = x} or {FIN, SequenceNum = x}, the consequent ACK has Acknowledgment = x + 1; that is, SYNs and FINs each take up one unit in sequence number space. Is this necessary? If so, give an example of an ambiguity that would arise if the corresponding Acknowledgment were x instead of x + 1; if not, explain why.

Q 12. The TCP header does not have a boot ID field. Why isn't there a problem with one end of a TCP connection crashing and rebooting, then sending a message with an ID it had previously used?

Submission: 25th March 2019