Assignment 5: Transport Layer

(1) Suppose that the clock-driven scheme for generating initial sequence numbers is used with a 15-bit wide clock counter. The clock ticks once every 100 msec, and the maximum packet lifetime is 60 sec. How often is resynchronization needed:
(a) in the worst case?
(b) when the data consumes 240 sequence numbers/min?

(2) Imagine that a two-way handshake rather than a three-way handshake were used to set up connections. In other words, the third message was not required. Are deadlocks now possible? Give an example or show that none exist.

(3) Consider a simple application-level protocol built on top of UDP that allows a client to retrieve a file from a remote server residing at a well-known address. The client first sends a request with a file name, and the server responds with a sequence of data packets containing different parts of the requested file. To ensure reliability and sequenced delivery, client and server use a stop-and-wait protocol. Ignoring the obvious performance issue, do you see a problem with this protocol? Think carefully about the possibility of processes crashing.

(4) Consider two networks, N1 and N2, that have the same average delay between a source A and a destination D. In N1, the delay experienced by different packets is uniformly distributed with maximum delay being 10 seconds, while in N2, 99% of the packets experience less than one second delay with no limit on maximum delay. Discuss how RTP may be used in these two cases to transmit live audio/video stream.

(5) If the TCP round-trip time, RTT, is currently 30 msec and the following acknowledgements come in after 26, 32, and 24 msec, respectively, what is the new RTT estimate using the Jacobson algorithm? Use $\alpha = 0.9$.

(6) To get around the problem of sequence numbers wrapping around while old packets still exist, one could use 64-bit sequence numbers. However, theoretically, an optical fiber can run at 75 Tbps. What maximum packet lifetime is required to make sure that future 75-Tbps networks do not have wraparound problems even with 64-bit sequence numbers? Assume that each byte has its own sequence number, as TCP does.

(7) A host is receiving data from a remote peer by means of TCP segments with a payload of 1460 bytes. If TCP acknowledges every other segment, what is the minimum uplink bandwidth needed to achieve a data throughput of 1 MBytes per second, assuming there is no overhead below the network layer? (Note: Assume no options are used by TCP and IP.)

(8) One difficulty with the original TCP SRTT estimator is the choice of an initial value. In the absence of any special knowledge of network conditions, the typical approach is to pick an arbitrary value, such as 3 seconds, and hope that this will converge quickly to an accurate value. If this estimate is too small, TCP will perform unnecessary retransmissions. If it is too large, TCP will wait a long time before retransmitting if the first segment is lost. Also, the convergence may be slow, as this problem indicates.

   (a) Choose $\alpha = 0.85$ and SRTT(0) = 3 seconds, and assume all measured RTT values = 1 second and no packet loss. What is SRTT(19)?

   (b) Now let SRTT(0) = 1 second and assume measured RTT values = 3 seconds and no packet loss. What is SRTT(19)?

(9) Although slow start with congestion avoidance is an effective technique for coping with congestion, it can result in long recovery times in high-speed networks, as this problem demonstrates.

(a) Assume a round-trip time of 60 ms (about what might occur across a continent) and a link with an available bandwidth of 1 Gbps and a segment size of 576 octets. Determine the window size needed to keep the pipe full and the time it will take to reach that window size after a timeout using Jacobson's approach.

(b) Repeat (a) for a segment size of 16 Kbytes.

(10) A CPU executes instructions at the rate of 1000 MIPS. Data can be copied 64 bits at a time, with each word copied costing 10 instructions. If an coming packet has to be copied four times, can this system handle a 1-Gbps line? For simplicity, assume that all instructions, even those instructions that read or write memory, run at the full 1000-MIPS rate.