CS446/646 Spring 2018

Test 1

There are 5 problems, each problem equally worth.

Problem 1. Two hosts A and B are connected by a link of length m meters. The link's rate is R bps. The signal's propagation speed is s meters/s. A has to send a message of size L bits to B

a) What is the propagation delay?
b) What is the transmission delay?
c) What is the end-to-end delay, ignoring the queuing and processing delays

Problem 2. Two hosts, A and B, are 4 hops apart in a packet-switched network (i.e., three routers on the path from A to B). All 4 links from A to B are at rate 10Mbps. Host A needs to send a file of size 10Gb to B. Ignore queuing, propagation, and processing delays. Assume packets do not contain headers.

a) What is the end-to-end delay if A sends the entire file as a packet
b) What is the end-to-end delay if A breaks the file up into packets of size 10Kb each?

Problem 3. A client sends a 128-byte request to a server located 100km away over a 1Gbps optical fiber. Light travels at the speed of light in optical fiber. What is the efficiency of the line during this remote procedure call.

Problem 4. Both UDP and TCP use port numbers to identify the destination entity when delivering a message. Give 2 reasons for why these protocols invented a new abstract ID (port numbers), instead of using process IDs, which already existed when these two protocols were designed.

Problem 5. UDP and TCP use 1s complement for checksum. Suppose you have the following three 8-bit bytes: 01010011, 01010100, 01110100. What is the 1s complement of the sum of these bytes? Show all work. Why is it that UDP takes the 1s complement of the sum; that is, why not just use the sum? With the 1s complement scheme, how does the receiver detect errors? Is it possible that a 1-bit error will go undetected? How about a 2-bit error?

Problem 6. Consider a channel that can lose packets but has a maximal delay that is known. Modify protocol rdt2.1 (we discussed this in the class, see diagram of this protocol in the lecture slides) to include sender timeout and retransmit. Informally argue why your proposed protocol can communicate correctly over this channel.

Problem 7. Consider a reliable data transfer protocol that uses only negative acknowledgements. Suppose that the sender sends data only infrequently. Would a NAK-only protocol be preferable to a protocol that uses ACK only? Why? Now, suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this second case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?