CS 640 Introduction to Computer Networks

Quiz 5

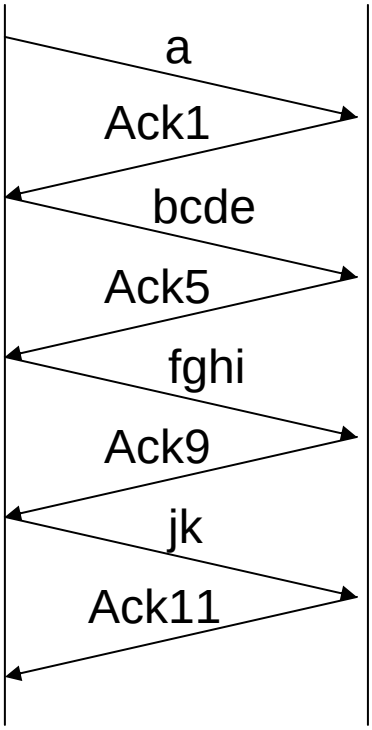
Write your name on the exam. Write something for every question. Students who do not write something for everything lose out over students who write down wild guesses. You will get some points if you attempt a solution but nothing for a blank sheet of paper. Write something down, even wild guesses. Problems take long to read but can be answered concisely.

Question	Maximum	Score
1	8	
2	12	
Total	20	

Question 1 – TCP, Nagle's algorithm

The Nagle algorithm, built into most TCP implementations, solves the "silly window syndrome". It requires the sender to hold a partial segment's worth of data until either a full segment accumulates or the most recent outstanding ACK arrives. Suppose the letters abcdefghijk are sent, one per second, over a TCP connection with an RTT of 4.1 seconds.

a) Draw a timeline indicating when each packet (and ACK) is sent and what it contains. Ignore congestion control, flow control, queuing delays, and assume no losses happen.



b) Suppose mouse position changes are being sent over the connection. Whenever the position of the mouse changes at the client it writes 4 bytes indicating the new position. Position changes occur a few times per second. The server sends replies updating the position of the mouse on the screen and these replies are never subject to Nagle's algorithm because they are large enough. How would the user perceive the mouse motion with and without the Nagle algorithm? How would the RTT affect this perception?

Because of the large roundtrip delay the movements of the mouse will be delayed (with or without Nagle). With Nagle's algorithm the mouse movements will also be jumpy as the mouse will "skip" all positions visited in the 4.1 seconds since the last packet and just jump to the final position. Lower RTTs would make the mouse movements smoother.

Question 2 – Miscellanea

a) Describe the fast retransmit feature of TCP congestion control and explain why it helps even when TCP's estimates of round-trip times are accurate.

Fast retransmit modifies TCP's behavior so that it retransmits a packet before the timeout occurs, after it receives 3 duplicate acknowledgements for the previous packet. It helps when the timers implemented by the operating system are coarse, so even though TCP has a good estimate of the round trip time the timer will fire much later.

b) An RPC protocol can use a tagged or untagged encoding of arguments and results. Which of these two encodings results in more overhead (more bytes being sent over the network)? Why must RPC protocols with untagged encodings use compiled stubs while protocols with tagged encodings can use either compiled or interpreted stubs?

Tagged has more overhead because the tags take space. Since tagged encodings can be self-explanatory, RPC mechanisms using them can have general purpose stubs that interpret the messages. For untagged encodings stubs must have specific knowledge of the type and order of values transmitted, so stubs are compiled based on a description of the parameters/results.

c) Give the advantages and disadvantages of setting large TTL values for DNS records.

A long TTL makes caching more effective: clients get better response times, servers have lower load. The disadvantage is that changes to the data take long to propagate because clients can cache (and trust) the old data for longer.