CMPT 371 Summer 2022

Assignment 3

Q1. (10 marks) Using the programming language of your choice, write two sets of client-server programs. In one set, the client and the server communicate over TCP, while in the other they communicate over UDP. In both sets, the client sends a "hello" message to the server, and the server replies with "back at you". The client then computes and prints the RTT. You must use sockets directly; you cannot use any higher-level frameworks.

a) Attach the source code of your programs and the resulting screenshots for each set.
b) How do the RTTs of UDP and TCP compare? Explain why it's so.
c) If both servers run simultaneously and use the same port number, won't the clients be mixed up when they contact their corresponding server on that port number? Why?

Q2. (10 marks) It is generally agreed that the maximum packet lifetime on the Internet is roughly 2 minutes; i.e., either the packet arrives at the destination before that, or the TTL will have killed the packet by then. We know that the *sequence number* in TCP counts transmitted octets over a connection and wraps around once it reaches the largest possible sequence number.

a) Calculate how long it will take for a TCP connection to wrap around over i) 56 Kpbs modem (old networks), ii) 10 Gbps network (today's networks). For simplicity, ignore lower layer protocols' (IP, Ethernet, …) overhead.
b) Explain why delayed/duplicate packets, such as those causing the *Incorrect Duplicate Detection* problem, suddenly become a serious problem for today's networks and not for the old networks.
c) RFC 7323 suggests using PAWS (Protect Against Wrapped Sequences), whereby a 32-bit timestamp, set in the TCP header's *options* field by the sender, is used to detect duplicates. Explain in 1 or 2 sentences (no fine details, just high level) how a timestamp would enable the receiver to detect duplicates.
d) For the timestamp to work, do the sender's and the receiver's clocks need to be synchronized? Why?

Q3. (10 marks) A source in Vancouver has established a TCP connection to a destination in Tokyo over a transatlantic T3 optical fiber line (~45 Mbps) with an RTT of 50 msec. The destination has indicated a window size equal to the maximum allowed by the *window size* field in TCP.

a) Ignoring TCP's slow start phenomenon and assuming no packet loss, what is the percentage of time that the source spends waiting for the destination's acknowledgement? Assume zero processing delay at the destination.
b) What is the effective bitrate of this connection?
c) What is the efficiency of this connection?
d) What would be the efficiency of this connection if it was over 56Kbps modem (old Internet), as opposed to T3? RTT is still 50 msec.
e) To alleviate the efficiency problem, RFC 7323 suggests extending TCP's *window size* field by an additional 14 bits, borrowed from TCP's *options* field. What would be the efficiency of the T3 connection under RFC 7323?