

Assignment 5

CSL 374/672

Due date: April 22, 2013, midnight (Monday)

Note: *Solve the problem on your own.* Approach the instructor for clarifications. Upload your socket programming code to Moodle.

1. Write two socket programs, `sender.c` and `receiver.c`, that together communicate using “Datagram Sockets”. The two programs should run on different machines. The `sender.c` program should take as a command line input the IP address of the machine on which `receiver.c` is running and the port number on which `receiver.c` is listening.

We will build some features of TCP such as sliding window, congestion control, and packet retransmissions on top of these two programs which use UDP. Assume that $MSS=1000$ bytes, that is, packet sizes must be less than or equal to this size. Each packet has a sequence number representing the byte number of the first byte of the packet relative to the first byte in the flow, and a field indicating the packet size. Thus the first packet will have sequence number 0. Assume that window W is initialized to 1 MSS. Every time a packet is sent out, a timer of 1 second corresponding to that packet is started.

The receiver, on getting a data packet, replies with a small packet of 20 bytes containing a cumulative ACK number x . This means that all bytes from 0 to $x - 1$ were received successfully. Note that implementing this is a bit tricky because data packets being received are not all of the same size, and some packets may get lost etc.

On receiving an ACK, the sender increases W by $(MSS)^2/W$ bytes and cancels the timers for all packets ACKed so far. The sender ensures that W bytes beyond the largest received ACK number are transmitted into the network, except for the case where the ACK number exceeds 1,00,000, in which case the sender program terminates. As far as possible, it sends out packets of size MSS, but is also allowed to send smaller packets. For example, if on receipt of an ACK the sender is allowed to send out 1900 bytes, then it will send out a 1000 byte packet and a 900 byte packet.

If a timer expires for any packet, set $W = MSS$ and transmit one packet of size MSS with sequence number equal to the largest ACK number observed so far.

At the sender, print out the values of W , time expired since the start of the experiment, and sequence number, every time a packet is transmitted. At the sender, print out the same quantities whenever an ACK arrives, except that instead of a sequence number it prints the ACK number.

Now let us simulate packet loss. Write `sender.c` to take as command line input with the `-l` flag a number which is either 0 or 1. If the value input is 0 then `sender.c` behaves as above. If the value is 1, then each packet is dropped at the sender itself with probability 0.05. This means that if the packet is to be dropped, then the sender does not actually transmit the packet to the receiver.

Comment your socket programming code well. Upload to moodle in separate files (i) `sender.c` and (ii) `receiver.c`