COEN 146 Lab Assignment 4
Winter 2012

Stop and Go Packet Transfer DUE: 02/25/2010 11:59PM

In this lab, you are going to implement a Stop and Go protocol to send and receive 5 packets between the client and the server processes. Also, you are going to implement a mechanism to identify each packet and ensure the correctness of the message sent in the packet.

When packets are transferred over the network, the order in which the packets are received at the destination may not match the order in which the packets are sent. So we need to implement a mechanism to track the packets. One method is to number the packets in sequence and include the number in the packet as a variable. In general, the correctness of the message is checked by computing the checksum and including it in the packet sent. The server computes the checksum of the received message and compares it with the value in the packet to verify the correctness. In this lab, you are required to provide a function that checks if the packet received was correct (for simplicity, you may substitute a shortcut for the checksum by verifying a numeric flag inserted at the end of the packet if you prefer).

The client sends a packet to the server and does not proceed to send another packet until it receives a confirmation packet from the server. For every packet received, the server will modify the packet and send it back to the client. “Modifying” the message will be as follows: after receiving the packet, the server prints the message, length of the message and checksum value, increments the sequence number, converts message to upper case and sends it back to the client. The sequence number is incremented to inform the client of the next packet the server is expecting. The client receives the packet and prints the contents of the packet. The sequence number of the received packet is used as the sequence number of the next packet sent.

You are going to implement this assignment by extending your previous code. You are going to accept a short text string as the message (you can pre-define these strings), and are to add two more variables to the packet data structure to define the sequence number and checksum field. Both the variables may be defined as an unsigned short. The packets are to be numbered starting from 0. You may assume a constant value for checksum as 17, or calculate a checksum based on the contents of the packet, the choice is yours. This value should be verified by a corresponding function at the recipient.

Submit well commented client and server programs with snapshots of the output.
Grade: 80% code submission, 20% demo/Q&A

Note: Submissions are to be made at the CAMINO drop box. Late submissions will not be accepted.