

1 TCP Credit Allocation and Sliding Window

You are to design a sliding window protocol that is appropriate for TCP. Once designed, you are to implement the protocol using UDP sockets. Your design should be in agreement with a subset of the TCP header fields: sequence number, acknowledgment number and credit allocation. Recall all these are based on byte counts. Your design should place the protocol at the transport layer, meaning the protocol must interact with the application and the network layer. The handing of segment and receiving of segments to/from the network can be emulated with `sendto()` and `recvfrom()` functions. For the application layer, a simple solution is acceptable such as a text file used for the source of application data and a source file used for the delivery of segments to the application.

Timers will be needed for the protocol. `Select()` is one mechanism for doing this but others exist. You probably want to consider interrupt driven programming. While a timer is ticking down, your process should not hang in a busy wait. Instead, the process should continue accepting and sending messages as needed. I can provide information on interrupt driven programming if requested.

2 Testing

Problems on the network, such as delayed messages, corrupted messages and lost messages, do not occur in the Unix Lab. You will need to simulate these problems. We will discuss this in lecture.

Projects will need to be demonstrated to receive credit. Your program should convey the window on the sender and receiver and what is currently buffered. Your program should also indicate what is occurring on the network.

3 Environment

Your program can be written in C or Java. Your project must run on a Solaris 8 (SunOS 5.8) Unix Network (e.g., the department Unix lab).

4 Submitting

Include the following when you submit your project:

1. A README.txt file in the format that follows
2. source code
3. other necessary files

5 Grading

The grading of the project will consist of two parts. The first part is my compiling and testing your project. The second part is you demonstrating your project to me.

6 Due Date: March 30th at 11:59pm

Late assignments will not be accepted.