Brett Chalabian

UID 004159487

Chul Hee Woo

UID \_\_\_\_\_\_\_\_\_\_\_\_\_\_

CS118 Project 2: Simple Window-based Reliable Data Transfer

Things still to do:

* If we get any packets that don’t have the proper header, then we should be cut off the transfer.
* We should put the file request from the receiver in a packet and just mark it as an ack packet
* Think about adding one of the extra features

**Implementation**

Data Transfer

Packet Model

Our packets consist of a header and body. In the header there is a character which denotes the type of packet, an unsigned long which represents the total size of the file, and an unsigned int which represents the sequence number of the packet. We have 4 characters that denote the types of packets and are *#defined* as ‘ACKPACKET’, ‘RETRANSMITPACKET’, ‘SENDPACKET’, and ‘FILENOTFOUNDPACKET’. Our packet size is fixed as per the email that we received from the TA denoting that minimum window size is 1kB.

Window Linked-List Model

We created a sort of Linked-List Window object that identified the packet, the state of the packet, and the timer of the packet. Our Window object keeps track of the window size by allowing a certain number of packets to exist in the window. The functions enumerated in sll.h describe methods to manage the window and the window elements that make up the linked list within. Many of the functions deal with basic linked list operations. The following

*setWindowElementStatus and ackWindowElement*

These functions deal with marking window elements as being acknowledged so the elements can later be removed if they are at the front of the window. Window elements cannot be removed immediately due to the sliding window of the Selective-Repeat Protocol. Used in the Data Transfer part of our code to mark window elements given an incoming ack packet’s sequence number.

*addWindowElement*

Adds a packet to a new window element and adds it to the tail of the window if possible. While the function returns true, packets can continue to be added to the window so the window always stays full unless all the packets have been added to the window

*getElementFromWindow*

Returns the first element in the window that needs to be sent. Elements marked for retransmission or elements marked as not sent are returned, and their statuses and timers are updated by the main function after packets in window elements are sent to the receiver.

*cleanWindow*

Removes items from the beginning of the window if they have been acknowledged and “slides the window”. This function also manages timers: it re-marks packets as needing to be retransmitted if their timers are up.

**Difficulties Faced**

Window Implementation

Timer Implementation

**Work Distribution**

Brett Chalabian

Chul Hee Woo