Zeeshan Pirzada ID:404144615
Kevin Sheu ID: 204160159

Reliable Data Transfer in C Using UDP Sockets

In this project, we implemented reliable data transport using UDP sockets by implementing the Go-Back-N protocol. We have two sides, the receiver and the sender. First, the receiver sends a SYN to the sender, who replies with a SYNACK. The receiver finishes the handshake by sending the filename it wishes to be transferred. The sender then breaks down the data into packets, and sends them one by one, keeping track of a timer and a window to make sure that no packets are lost.

To do this, we implemented a packet container (struct packet) as in the following:

```
struct Packet {
        int size; int type; int seqNum, int ackNum, int checkSum
        char data[PACKET_LEN];
};
```

The data array is used to actually store the data with each of the other fields copied into the first 20 bytes. Since our PACKET_LEN = 1kb, that left us with 1004 bytes of data per packet. To keep track of our packet window, we also defined a struct packet_stream, which serves as basically a wrapper list class of packets that we had sent out. Keeping track of where base and nextSeqNum allowed for resending packets very easily.

Some difficulties we faced were debugging the actual file transfer process. Since there were two programs, it was hard to tell if the problem was on the receiver or sender side. Also, working with recvfrom() and sendto() can be very frustrating, as the

documentation is not entirely clear. For example, we were unsure how those functions would work in the presence of a null-terminated byte in their buffers. We solved these problems by using print statements that told us exactly when each packet would leave and arrive, as suggested by the spec.