Raymond Fleming
Jeffrey Bromen
Systems and Networks II
Project 2 RDT 3.0
Protocol

Simulated Packet Structure

Simulated Packet (54 Bytes)					
	Source IP	**Source Port**	**Destination IP**	**Destination Port**	**Segment**
# of Bytes	**16**	**6**	**16**	**6**	**10**

Segment Structure

Segment (10 Bytes)		
	Header	**Payload**
# of Bytes	**1 (1 corruption bit, 7 segment number bits)**	**9**

Sender

The sender algorithm retrieves the arguments from the user, as well as the message from the user. The message is then segmented into the fewest segments possible for transmission and is sent piece by piece through the simulated packet and segment structure above. The destination address and port in the actual UDP packet will match the Network program, not the receiver program. On each send, an ACK is expected before sending the next packet. If one isn't received by the timeout then the current packet is resent. Once the entire message is transmitted the sender exits. For the Sender, there are two states which are tied to the sequence number. This is used to determine which sequence number to assign to a packet and to look out for in the ACK response.

Receiver

The receiver opens a socket and awaits a message from the Sender (which will actually arrive from the network). Once the packet is received, it is first checked for corruption. If it is not corrupt the packet is then checked to make sure it is the correct sequence number. If not, the prior ACK is resent to have the correct packet sent. If so, a new ACK is generated and sent to the sender. For the Receiver, there are two states which are tied to the sequence number. This is used to determine which sequence number to look out for in a received packet. No exit condition was implemented, so the program will run until manually stopped and sending a second message using a second Sender program will append the second message to the first message rather than separating the messages.

Network

The network program simulates the network between two hosts on an unreliable channel. Simulated packets are received both from the sender and receiver before being forwarded on to their destinations. On receipt of a packet, a random number generator is run to determine if the packet should be dropped first (since dropping it would nullify corruption/delay), then corrupt, then delayed. The likelihood of each outcome is determined through command line arguments. Since the network is simulated, the destination addresses are pulled from the simulated packet then sent on to the

destination. New hosts are checked for and logged so that sender and receiver messages may be recorded, along with all packet delay/loss/corruption. The Network program will print recorded network statistics every 2 packets during the run, and the program will run until manually stopped. This allows it to stay running through multiple runs of Sender and Receiver.