Darren Leung

Shantanu Mantri

Jason Wu

RELDAT Protocol Specication

**Connection Establishment**

RELDAT is a connection oriented transport protocol. As such, the server and the client must first establish a connection before exchanging data. This connection establishment will be modeled after TCP’s 3 way handshake. To initiate a connection, the client will send an empty packet with a SYN flag, indicating to the server that it wishes to establishes a connection. The server acknowledges this request along with sending its own SYN packet by sending an empty packet with both SYN and ACK flags. Finally, the client ACKs the server’s SYN packet.

**File Pipelining (Packet Drop and Duplicates)**

RELDAT supports reliable pipelining using the go back N protocol. A window size argument is specified on both client and server side. Once the connection has been established, the client will start sending data in packets ordered by sequence numbers starting with 0 for simplicity. RELDAT also supports the transfer of data between both the client and the server in both directions by allocating a socket for receiving and a socket for sending. RELDAT accounts for lost packets by using a retransmission timer. If the timer expires, the sender will simply resend the packets in the window. The receiver will ignore any duplicate packets by keeping track of the most recently received sequence number. Once the sender has sent all file packets and has ensured that the receiver has received all packets, the sender will send an empty packet with a FIN flag, indicating the end of the file transfer. Once the receiver receives the FIN, it will order the packets in the receive buffer.

**Packet Corruption**

RELDAT accounts for packet corruption by storing a CRC32 checksum of the packet data in the header.

**Packet Ordering**

For pipelined transmission, RELDAT will store all received packets in a buffer and reorder all packets after file transmission is complete.

RELDAT Header

* Source Port (32 bit)
* Destination Port (32 bit)
* Sequence Number (32 bit)
* CRC32 Checksum (32 bit)
* Control flags (ACK, NACK, SYN, FIN)

State Machine:

OPEN

FIN + NACK

SYN

SYN

SYN + ACK

CONNECTED

LISTEN

CLOSED

ESTABLISHING

CONNECTION