

CS118 Project2 Report

Cheng Ma 105033453, Shuo Bai 505032786

March 16, 2018

1 Introduction

In this project, we try to achieve the reliable data transmission based on UDP, and the window-based protocol is Select Repeat(SR).

In the implementation, we use the struct, packet, to denote the packet form of the transmission. And this contains five fields, *type*, *syn*, *fin*, *number* and *data*. *type* indicates the type of the packet, namely ACK or SEQ_NO. *syn*, *fin* are used in establishing and disconnecting the connection. *number* represents the sequence number or acknowledge number. Finally, *data* means the packet payload.

According to the description, our maximum packet size is 1024 bytes. The maximum sequence number is 30720 bytes, the window size is 5120 byte, and the retransmission time out value is 500 ms.

1.1 server.c

In the server.c, when receiving the required file from the client, the server will first check if this file exists. If not, the server will just return the "File not found". If exists, the server will retrieve the packet and then divide the file into small pieces.

In order to achieve the retransmission, for every sent but not acknowledged packet, we store the packets and record the time when it was transmitted. If a time-out event happens, this packet will be retransmitted.

Another thing is to decide whether the window should move forward or not. If the acked packet is in the first slot of the window, then the window should move; While if it is not, the server will just record the acked number and decide not to move the window.

1.2 client.c

In the client side, just the same as the server side, it should also maintain a window to indicate whether a specified packet is received or not. If the sequence number of the received packet is at the first place of the window, the window should move forward and write all the packets with continuous sequence number into the file; If it's not, the client should just store the packet. In both cases, the client should send back the ack packet with the corresponding ack number.

In the final phase, when the server sends all the packets and decides to shut down the connection, the server will send a packet with the FIN flag set. When receiving this particular packet, the client will first send the acked packet to allow the disconnection. But the client can not disconnect immediately, because this ack packet is not guaranteed to be received by the server. Thus, the client should wait for a certain amount of time until it can safely assume that this ack packet is indeed received successfully. In this experiment, we set this time out value to be the double duration of the retransmission timeout.

2 Difficulties

In the beginning, we didn't care too much about the corner cases, such as when the sequence number reaches the maximum number. This brings us much trouble. Another problem is how to deal with the final phase. When the server sends the FIN packet, after a certain timeout, it should again resend the packet. Also, when the client has sent the ack packet, it should wait in case there is another FIN packet coming because the ack packet is lost.