Project 2 : Implementing TCP for online gaming

Foundation of Computer Networks

Submitted by : Karan Bhagat

1) Brief description of implementation

Proposed TCP protocol is an overlay over the un-reliable UDP protocol. Main parts of the protocol are:

MyTCPSocket class : This class provides the functionality same as original TCP socket but with fast sending of data. Whole file can be just passed to send function by converting it into bytes and this socket will make sure it reaches destination.

MyTCPServerSocket class: This class is used by server to perform initial handshake with the client and to get socket connection with the connected client.

MyTCPPacket class: This class defines the TCP packet to send through the MyTCPSocket class object. This class is discussed in detail in section 2.

2) Header fields

Sequence number : This field is important for informing the sequence number of the packet being sent. This field help in reliably detect which packet are missing by checking which sequence number is expected by the client.

Ack number: This field contains the least unacknowledged sequence number or the sequence number which the server is expecting.

Receive window: This field is important to let the server know how much buffer is available on the client for storing incoming packets. This help in controlling flow of packets from the server by changing the sliding window size

Ack : This is a Boolean field to mark the packet as an acknowledgment packet for already received data packet.

Fin : This is a Boolean field to mark the packet as the finish packet which ends the connection.

Syn : This is a Boolean field to mark the packet as the handshake packet.

Checksum : checksum is used only for checking corruption but not for correcting it.

3) Description of Desgin

->Convert the file into a byte array.

->Used Go back N for implementing reliability with initial window size as 1 MSS. MSS is set as 10000 bytes.

->When MyTCPSocket object is created on the client , it performs initial handshake with MyTCPServerSocket on the server.

->byte array containing the data of file is send through using send function call in MyTCPSocket object.

->This byte array is then divided further in MyTCPPackets and sends the number of packets allowed in current window.

->Whenever acknowledgement is received for delivered packet it checks if the acknowledgement number is the one which is expected by the server. Congestion window is also increased whenever an acknowledgement is received. If timeout happens, congestion window is decreased to half. In case congestion window reaches the thresh hold value then window is not incremented beyond that.