

## Computer Networks

### Assignment 2: Reliable UDP

In this assignment I have created a reliable service over UDP without using a connection oriented approach. The server client pair have the following capabilities:-

- Congestion Control.
- Adaptive round trip time and timeout calculation.
- Selective acknowledgment for lost packets.
- Buffering for out of order packets.
- Retransmission in case of packet loss.

#### **Assumptions:-**

- The data only flows from server to client.
- It is a file transfer protocol.
- The client has unlimited input buffer i.e. there is no issue of flow control.
- The initial sequence number is 100 for both client and server.

**The Header:** The header I used has 3 components- acknowledgment number, acknowledgment bit and data/payload. The acknowledgment number tells the receiver that all the packets up to ack num -1 have been received and I need ack num numbered packet now. The acknowledgment bit tells whether the packet is for acknowledgment of a request (if value is 1) or has some initial setup info (if value is 0). The payload is the data sent by the sender.

**Model:** The following are the steps taken for the file transfer:-

- Client sends a request for a particular file name.
- Server receives the request and loads the file.
- Server calculates the number of 1300 byte packets that need to be sent to send the whole file successfully.
- Number of packets is sent to the client and state descriptor arrays are initialized for each packet.
- The client receives the info and sets up its own variables to track the status of each packet.
- The client sends the acknowledgment for the first data packet.
- Server sends the packet in accordance to the congestion window and starts the timer for each packet.
- On receiving an acknowledgment of the packet the server closes its timer, updates the congestion window with respect to the ssthresh and the timeout interval value.
- In case an acknowledgment is not received in time, the congestion window is reset to 1, ssthresh becomes half of previous congestion window and the packet is retransmitted with timer being reset.
- In case of the client, if it receives a packet that it has already been received before then it drops the packet and sends an ack for the next required packet.
- In case it receives an out of order packet, it stores the data in a buffer (sorted queue) and sends ack for next required packet.
- In case a correct packet is received, the data is displayed on screen, buffer is checked for subsequent packets received if any and displayed themselves and a cumulative ack is sent for the next required packet.

#### **Points full filled:**

1. The header has been implemented correctly with only acknowledgment flag being used.
2. Sliding window algorithm has been implemented that allows the server to use the network much more efficiently and the client to send cumulative acknowledgments.
3. RTT is re calculated on successful receipt of a packet.
4. Congestion control is implemented and the server prints when it switches between phases. Also at the end of execution the number of packets sent in each phase is printed.
5. Selective acknowledgments are implemented.