

# **Computer Networks Assignment 3**

## **Reliable Protocol Design**

### **Team Details:**

- Dhruv Shah - 2017A7PS0138H
- Vaibhav Yadav - 2017A7PS0107H
- Vedant Goyal - 2017A7PS0187H
- Ashish Gupta - 2017A7PS0056H
- Prakhar Gupta - 2017A7PS0121H

# Protocol Specification

The protocol has been designed over UDP with some reliability features. This Application Layer Protocol uses UDP sockets to transfer packets in a P2P (Peer to Peer) Setup.

## Assumptions about the following -

### 1)Application

- This application is designed to work in a two Peer Environment.
- Each message is treated as a separate packet.

### 2)Network

- There is a fixed congestion window.
- Selective Repeat is being followed.

The Congestion Window in the network has been fixed to size 100 packets. The sender follows the Selective Repeat in a way as only the packet for which Acknowledgement has been delayed or lost is retransmitted, as opposed to GBN wherein all packets in the congestion window after the lost/delayed packet are also retransmitted.

Also whenever a packet's Acknowledgement is received by the sender, the packet is removed from the congestion window. The packet is forwarded to the "delete\_list" where the packet will be subsequently destroyed. The deleted packet leaves an empty space in the congestion window. The indexes of the packets to the right of the space are shifted left by one position, which in turn generates space at the right end of the congestion window for any subsequent packets to enter the congestion window.

For example if window size is ten, and if the congestion window consists of packets one to ten, and if Acknowledgement is received for packet 2, then the packet 2 will be removed from the congestion window. The packet indices numbered from three to ten are shifted left. This generates a space at the right end of the congestion window which can be filled by the eleven'th packet. Thus at last, the congestion window will consist of packet one and packets numbered from three to eleven.

Moreover as can be deduced from the above assumptions the protocol follows selective Acknowledgement as one ACK acknowledges only that particular packet, as opposed to

Cumulative ACK wherein one ACK acknowledges that particular packet as well as all other packets before it.

## Strategies to tackle different situations -

### 1)Message Packet -

- If the sequence number of the packet received by the receiver is less than the expected sequence number (the sequence number of the packet for which the receiver is waiting), that means the packet was already received and currently received packet is duplicate, then drop the packet and resend its ACK.
- Else If the sequence number received by the receiver is equal to the expected sequence number (the sequence number of the packet for which the sender is waiting), then the receiver will add that packet to the buffer (**recv\_buff**). It will then increase the received space sequence number by 1 (i.e., **seq\_num\_recvd+=1**). It will then acknowledge the packet by sending an ACK for this packet to the sending process.
- Else if the sequence number received by the receiver is greater than the expected sequence number (the sequence number of the packet for which the receiver is waiting), then simply drop the packet and don't send an ACK for it. It will be resend by the sender in future.

### 2)Acknowledgement Packet -

If the packet type is ACK packet, then mark the corresponding packet for that ACK (i.e., mark **ack\_recvd** as true) in **send\_buff**.

That packet will then automatically be deleted from there later. (this will be done by the thread that handles the **send\_buff** i.e. **send\_thread\_util** function)

**3)Retransmissions:-** The protocol deals with the need of retransmissions in a generalised way. After sending the packet using UDP the protocol waits for one second for the corresponding Acknowledgement to reach the sender. If the sender does not receive the Acknowledgement by then, the **send\_thread\_util** method in the protocol again sends(retransmits) the packet over UDP.

**4)Message Packet Loss/ Delays:-** Whenever a Packet Delay or a Packet Loss occurs, the packet won't reach the receiver in time or won't reach at all. In both the cases the receiver won't send the Acknowledgement. As the receiver expects the Acknowledgement within one second, it will never receive it and thus will trigger a retransmission as explained above.

**5)Acknowledgement Packet Loss/Delays:-** After the packet is received by the Sender, if the Acknowledgement is delayed or lost, the Sender wont receive the packet in time(one second) and thus trigger a retransmission as described above.

**6)Packet Corruption:-** If a situation arises, wherein the packet gets corrupted, the UDP protocol discards the packet. Then, the situation on the receiver's side resembles the Packet Loss situation which is explained above.