Sender process(rt\_srv):

1.Get data from local app, calculate the deliveryTime and store it into the window and Send to the Receiver.(deliveryTime = sendTS+ baseDelta + drift + lantency Window)

2.Check whether we should shift the window. (packet delivery time < present time)

3.Receive the packet from receiver and update the Base\_Delta, Drift.

Send the ACKACK packet And check whether we have NACK and resend packet if it is not expired.

Receiver process(rt\_rcv): The receiver has three main responsibilities.

The first responsibility is Receiving the packet from Sender.

There could be two types of packet.

1. Data Packet: Check whether we already had the packet. If not, check whether the delivery time is not expired. If not write it into our buffer.
2. ACKACK Packet: the receiver gets packets and adjust the Base\_delta and Drift.

Base\_delta =( recvTime2(present time) – recvTime1) /2

drift = recvTime2 (present time) – (ACKACK\_TS + baseDelta)

The second responsibility is sending the ACK and NACK to sender.

The third responsibility is Delivering the packet on Delivery Time.

Data Structure of Sender

2d array Buffer Window

Int Base\_Delta

Time drift

Data Structure of Receiver

2d array Latency Window

Int Base\_Delta

Time drift

Data Structure of Message

/\* 0-> Sender sends data to Receiver

1->Sender sends ACKACK

2->Sender Receive ACK from Receiver\*/

Type

Seq /\* Sequence number of Data Packet\*/

Delivery Time /\* The delivery time of data\*/

DATA

ACK

NACK

ACKACK /\* ACK of ACK\*/

Send\_TS /\*Send time of packet

It could change when we resend the Packet\*/

Receive1\_TS /\*First time receiver receives the packet \*/

ACKACK\_TS /\*The time when sender gets the ACK \*/

Receive2\_TS /\* No need, it is the time when receiver gets ACKACK \*/

Base\_Delta /\*Measure of ½ RTT\*/

Drift /\*The difference between time in Sender and time in Receiver \*/