Analysis_Assignment2
Tanvi Vijay Bhavsar

"I have read and understood the course academic integrity policy located under this link: http://www.cse.buffalo.edu/faculty/dimitrio/courses/cse4589_f14/index.html#integrity".

Multiple Timers in SR

When new packet arrives the time when it is sent is stored in structure which is given by time_local When first packet comes I start timer for that packet with timeout. Whether timer is started or not is told by time_started variable.
To maintain state of every packet following structure is used-

```
struct pack_details
{
        struct pkt my_packet; //stores packet received
        float starttime_packet;//stores time at which packet arrived
        float time_stop; //Stores time at which timer for packet is stopped
        int pk_timer_started;//tells whether timer for packet is started or not
        int ack;//tells whether ack is received or not for that packet

};
```

When a new packet arrives, first it is checked whether timer is started or not. timer_started is 0 is timer is not running. If timer is not running then packet with minimum startime is found . Then timeout for this packet is calculated by subtracting difference of current time and start time from timeout. This is done because packet is suppose to have time out after TIMEOUT time. Since some time is passed since it was snethat time is subtracted.
When acknowledgement is received it is checked whether timer is running for the packet for which acknowledgment is received. If yes timer_started and pd[i].pk_timer_started are made 0.
If timer_started is 0 then as mentioned above packet with minimum start time is found and timer is started for it.
In A_timerinterrupt() , details for packet for which timer expired is updated-

```
pd[i].pk_timer_started=0;
pd[i].starttime_packet=time_local;
```

Now this packet would be sent again so its start time is obtained from time_local.
As mentioned above again packet with minimum starttime is found, timeout is calculated and timer is started for it.

GBN Optimization

To increase throughput of GBN, I made following change in A_input()

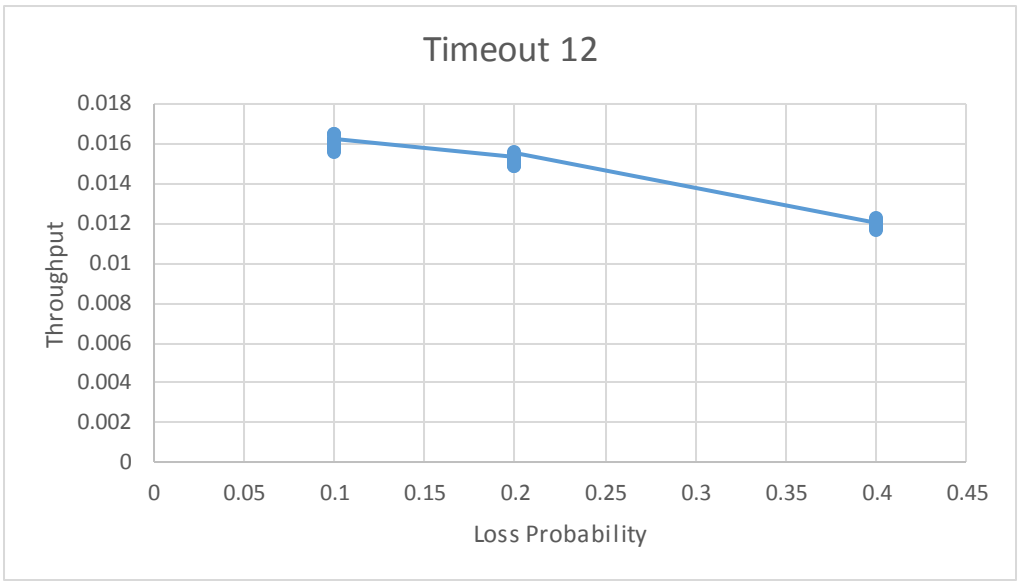
```
if(base != (packet.seqnum+1))
 {
base=packet.seqnum+1;
        if(base != (packet.seqnum+1))
                {
                        base=packet.seqnum+1;//move window ahead

                        if(base==nextseqnum)
                        {
                                if(timer_A_started==1)
                                {
                                        stoptimer(0);
                                        timer_A_started=0;
                                }
                        }
                        else
                        {
                                if(timer_A_started==1)
                                {

                                        stoptimer(0);
                                        timer_A_started=0;
                                }

                                starttimer(0,TIMEOUT);
                                timer_A_started=1;
                        }
                }
```

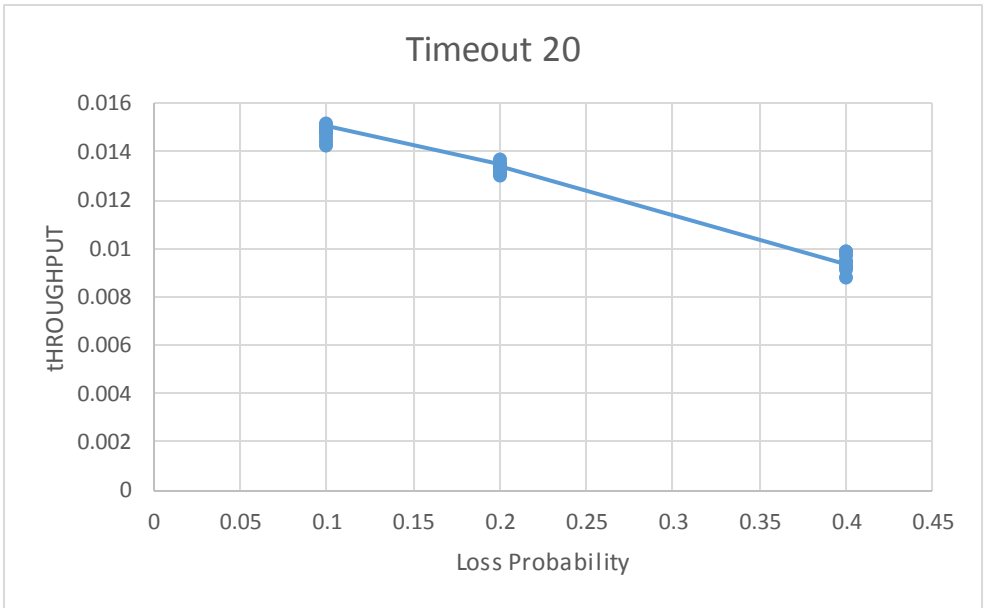
I added this condition- `if(base != (packet.seqnum+1))`

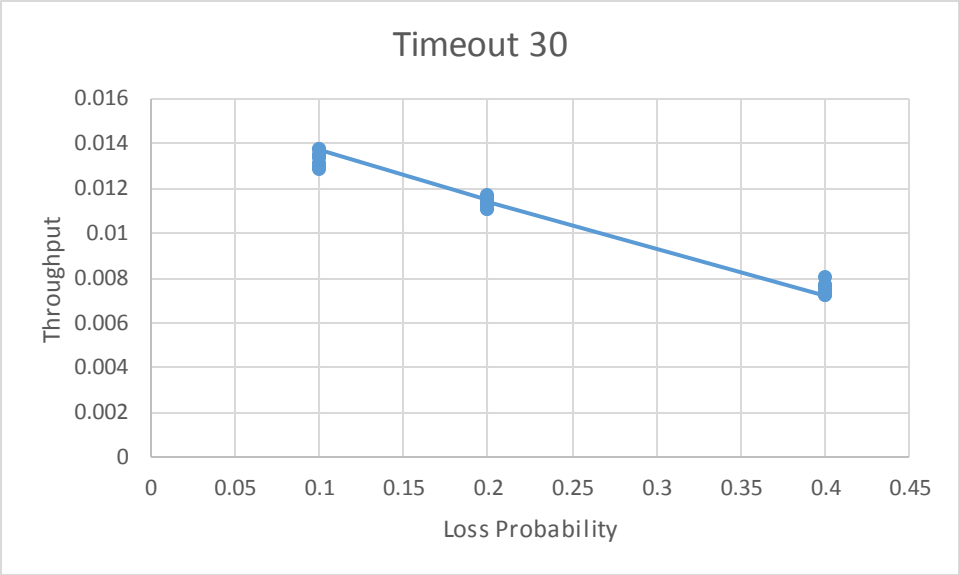
Initially if sender receives same acknowledgement more than once it used to stop and start time again and again if base is not equal to nextsequm. In this case packed retransmission is delayed. To avoid this above condition is added.

Timeout calculation-

1) ABT
   Ran script with messages 1000, corruption 0.2 and time between messages as 50.Following observations were made-


Timeout 12
Throughput
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
Loss Probability

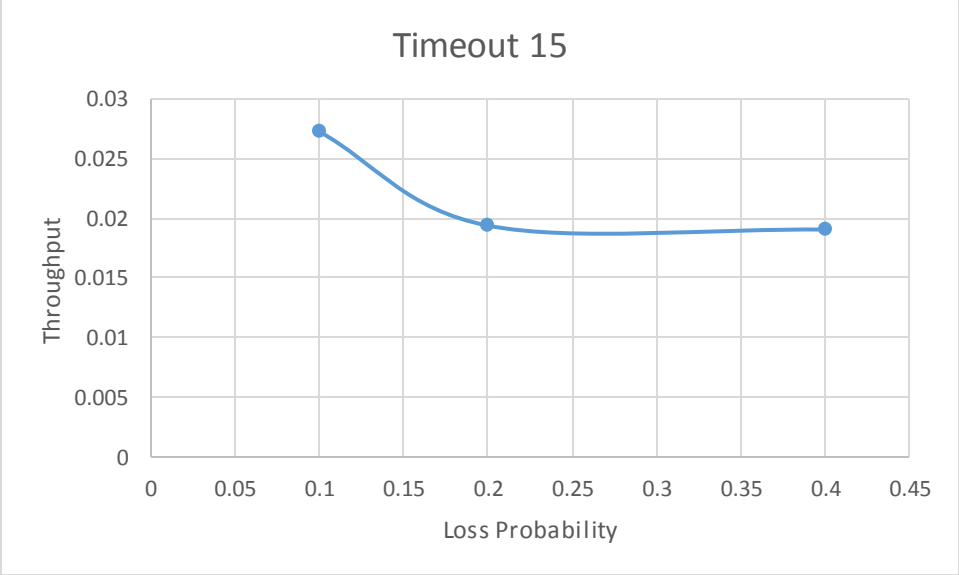

Timeout 20
tHROUGHPUT
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
Loss Probability

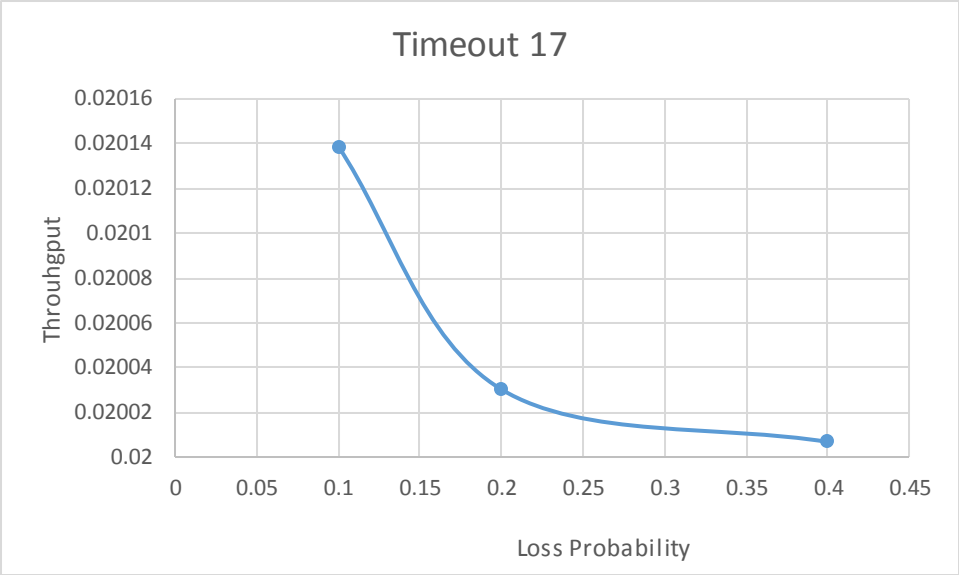


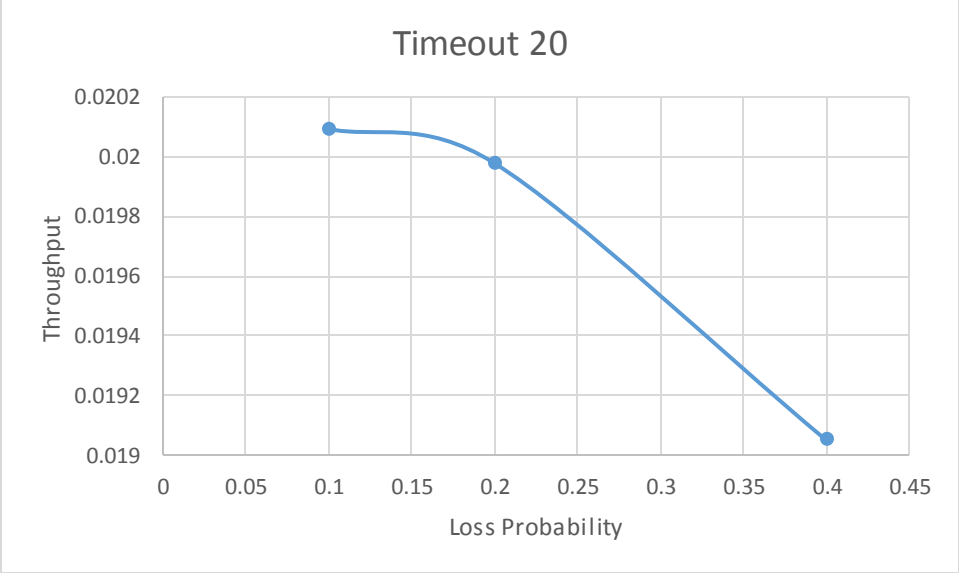
Since throughput was maximum for timeout 12 timeout 12 was selected.

2)GBN

Ran script with messages 1000, corruption 0.2, window 50 and time between messages as 50. Since we get 10 values mean of all was taken to plot graph.



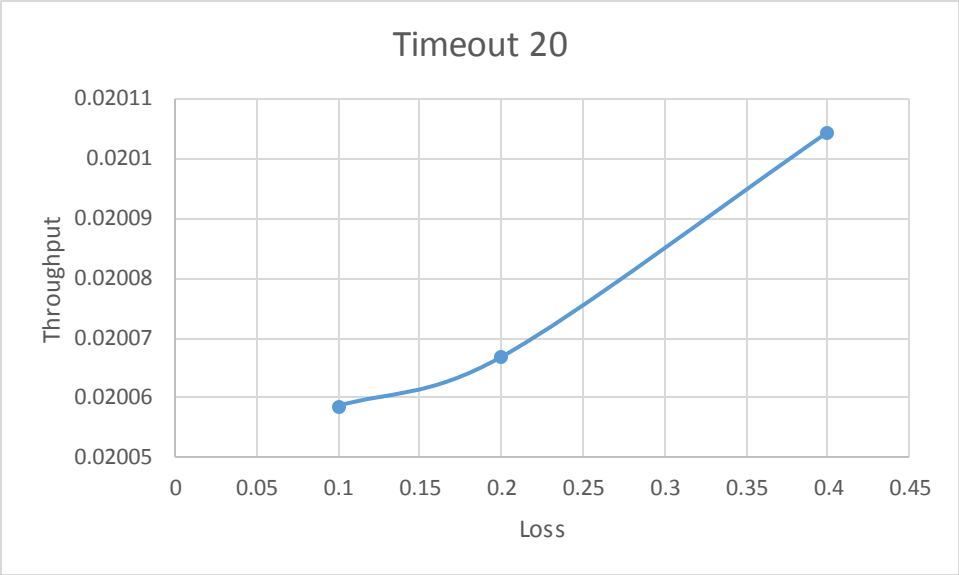




Since throughput was maximum for timeout 17, it was selected

2) Selective repeat

Ran script with messages 1000, corruption 0.2, window 50 and time between messages as 50. Since we get 10 values mean of all was taken to plot graph.



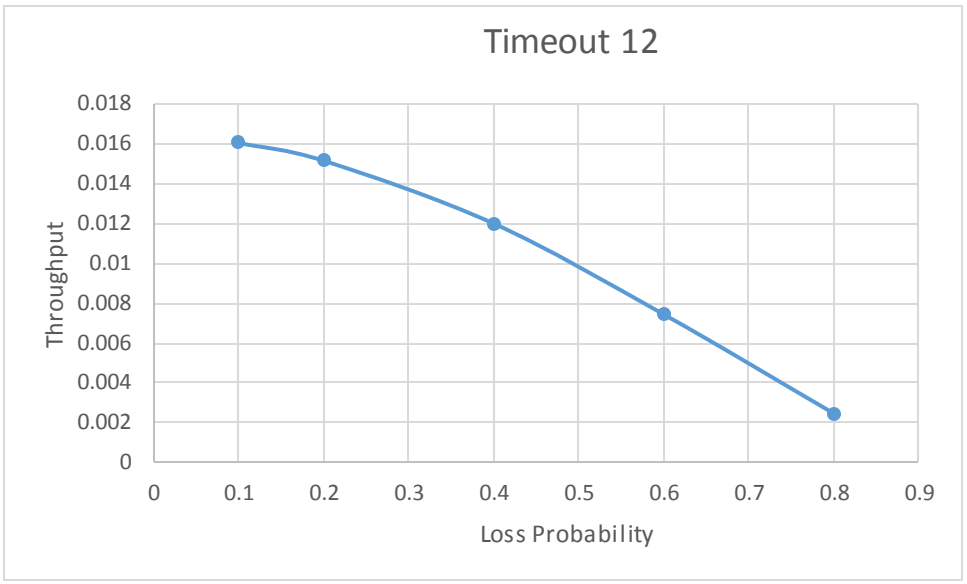
Since throughput was maximum for timeout 20, it was selected

ABT works well in less timeout. It has to wait for acknowledgement till it moves to next state. If a packet is lost and time out is large it will wait for lot of time and will not be able to send next packets in that time. Thus efficiency decreases for high timeout. Since RTT was given to be 10, timeout should not be less than 10. Hence 12 was taken.
If a small timeout is taken a packets are retransmitted frequently it increases congestion
If timeout is large , lot of time will be taken for completion since will have to wait for packet to be retransmitted. Thus value of timeout which is neither very big nor very small is selected.
In GBN receiver will reject packet if it does not have expected sequence number. So if desired packet is lost and timeout is high many packets will be lost and efficiency will decrease as that packet will be retransmitted late. So a timeout which is neither very high nor low is taken.

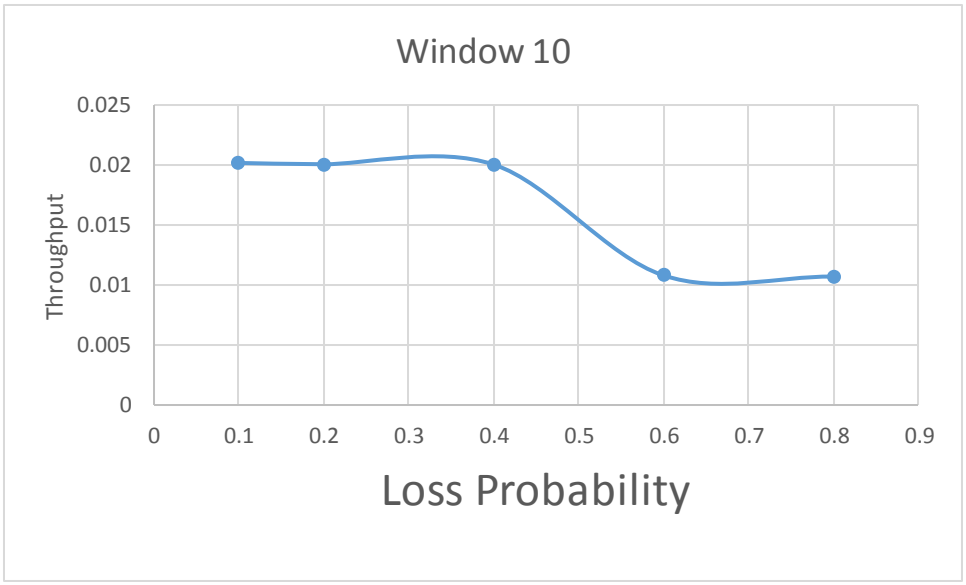
In all graphs below of Experiment 1 and 2 throughput is taken as average of the throughputs obtained when we run script

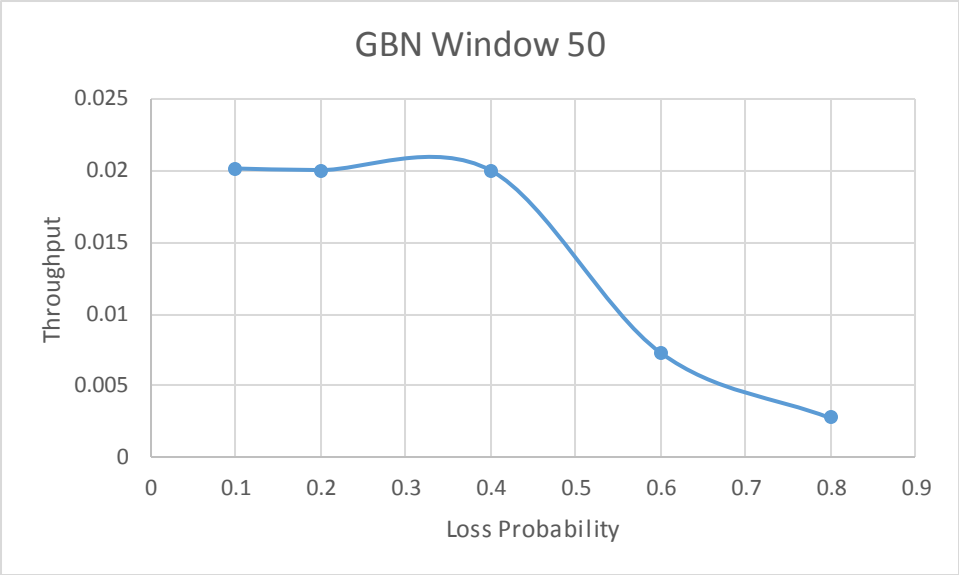
Experiment 1

ABT Timeout 12

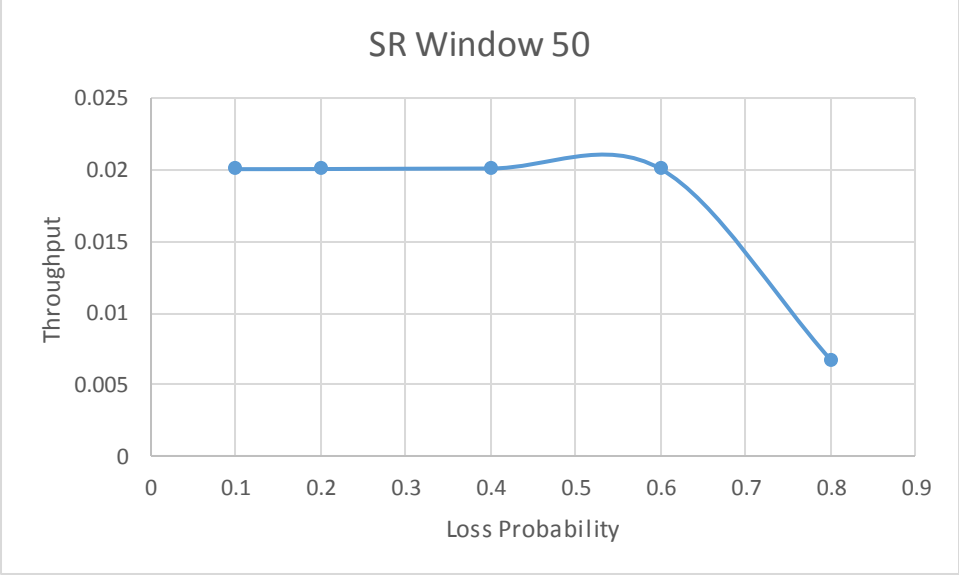


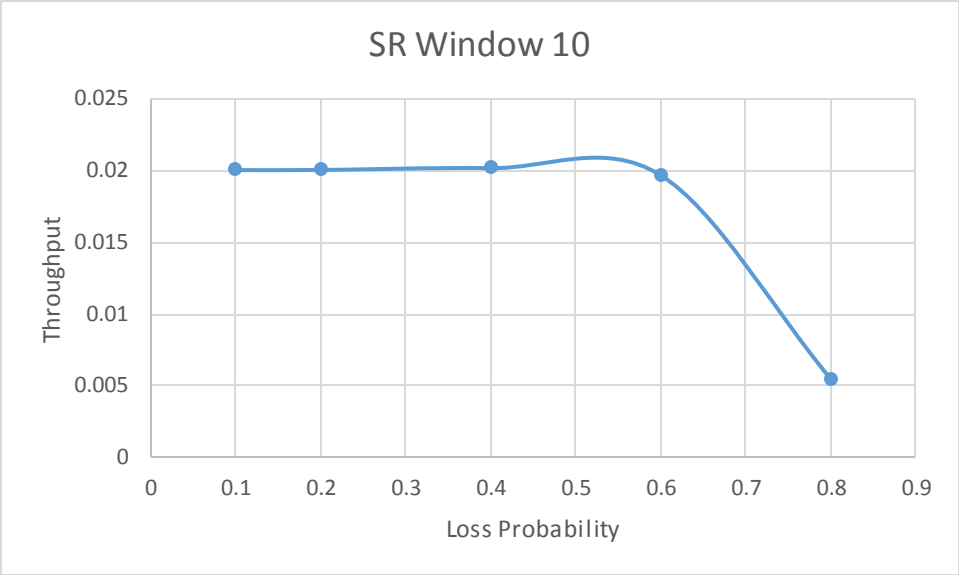
GBN Timeout 17





SR Timeout 20



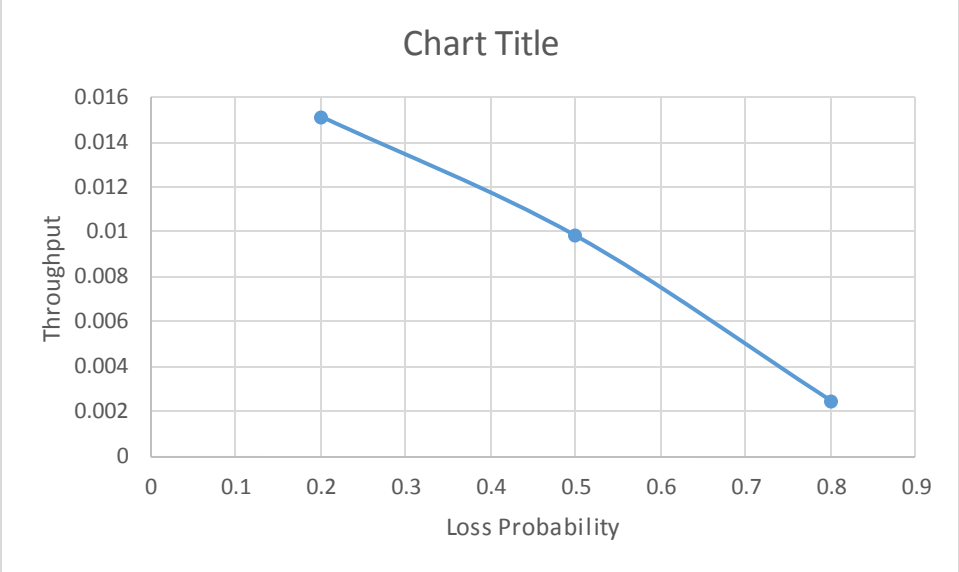


In ABT sender sends one packet and then waits for acknowledgement. Till it does not get desired acknowledgement it does not change state and thus does not send next packet. So lot of time is wasted in waiting. Utilization is less. Bandwidth utilization is less. Thus efficiency of ABT is least. In GBN and Selective repeat multiple packets are send at a time so higher efficiency.
In ABT both experiments throughput reduces with increase in loss as it will have to wait for acknowledgement for lot of time.
GBN rejects out of order packet and retransmits all packets in window if timer expires. This is not done in Selective Repeat. So SR should be better. However I have used single hardware timer to implement multiple timers. Processing for multiple timers might reduce performance. State for all packets needs to be saved for multiple timers. This would be an extra overhead. Maybe because of this there is not much difference between gbn and SR when loss probability is less.
Even if one packet which falls in window is lost or corrupted, all packets in window are retransmitted. If window size is big and timeout is not big, it can happen that when timer expires all packets assuming 500 are there in window are transmitted when expected sequence no is 400. By the time B receives even 300, the timer expires again and sends again. This may repeat again and again. So when window size is big and timeout small, lot of time is taken and throughput is less. We can see from graph that for high loss rate throughput decreases. In GBN throughput for window 50 is less than for 10 when loss probability is more i.e 0.6, 0.8

It can be see that SR performance is best. With increase in window size Selective Repeat performs better because it can send more number of packets at a time. In above graphs for window 10 and 50 for high loss SR with window 50 performs better.

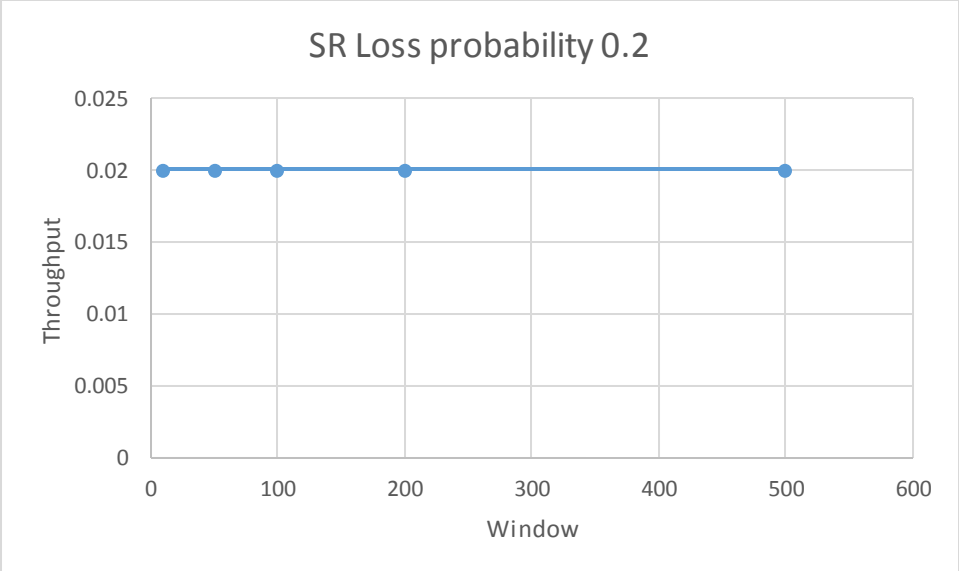
Experiment 2

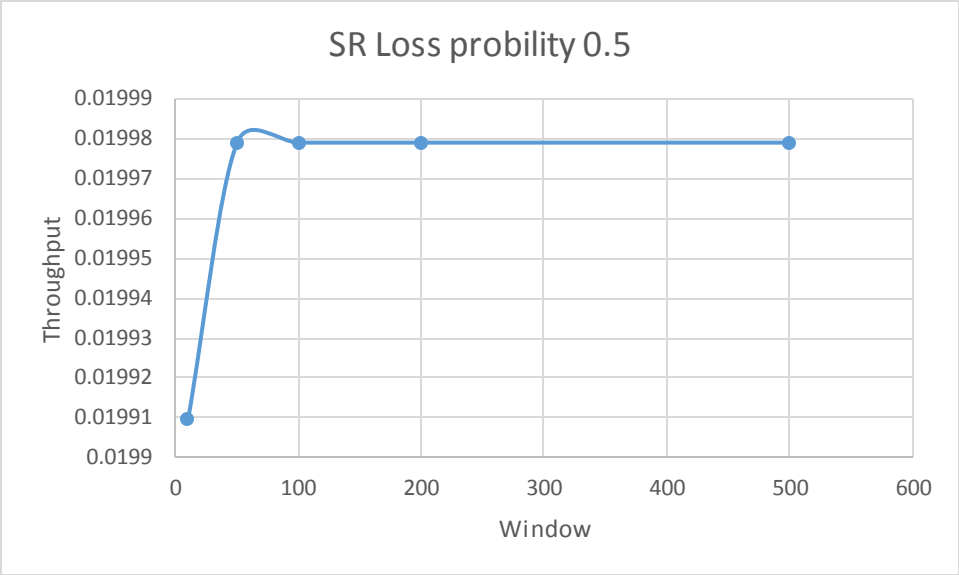
1)ABT-Timeout 12

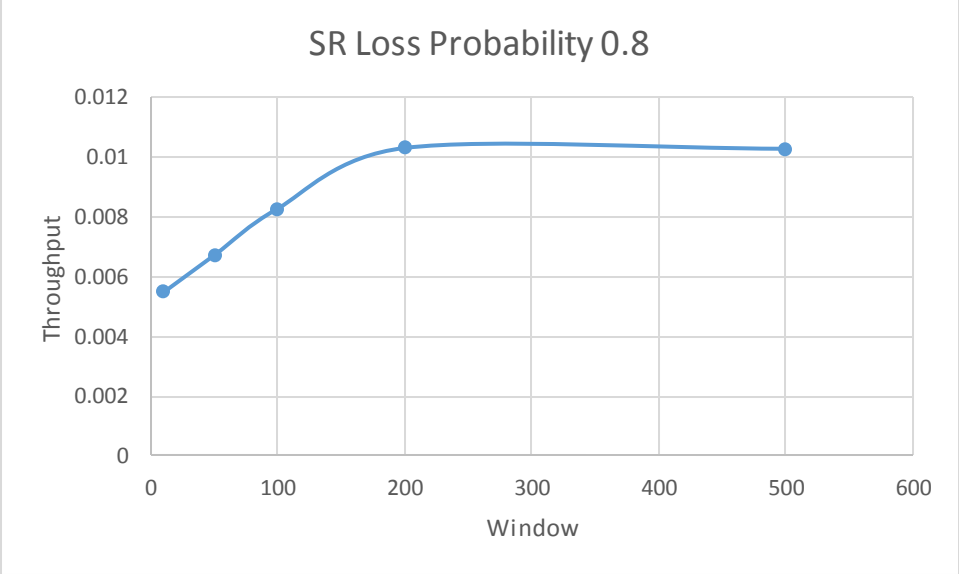


As discussed above throughput decreases with increase in loss.

2)SR Timeout 20

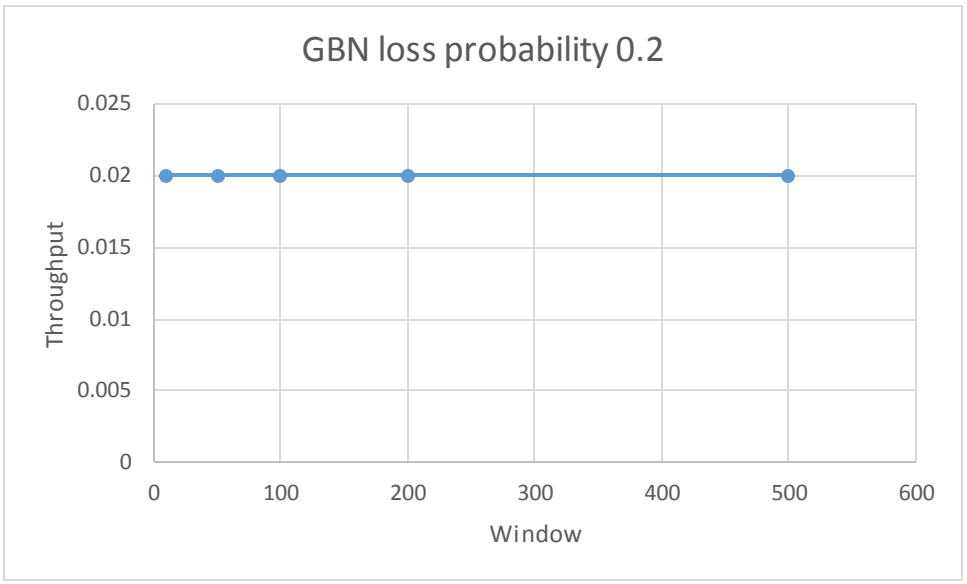


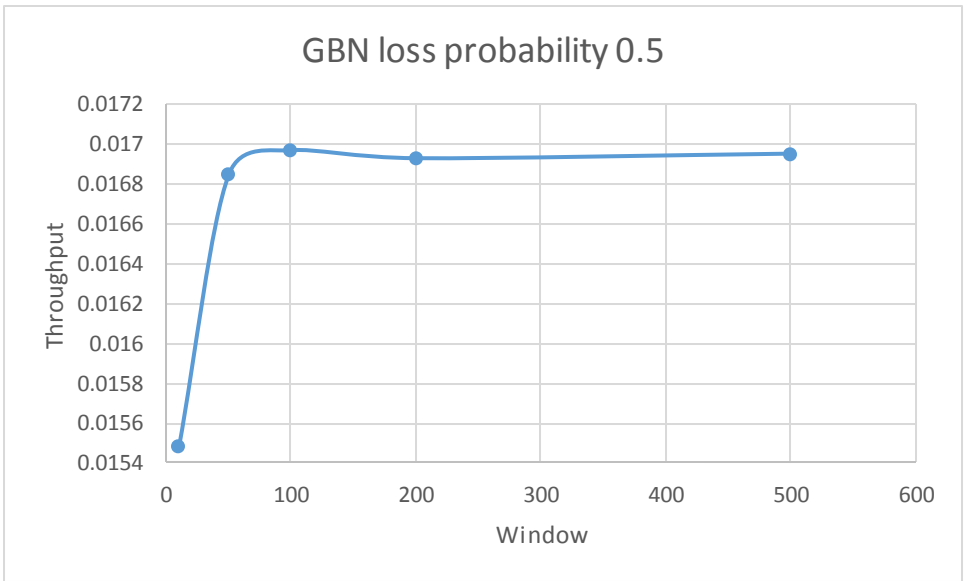


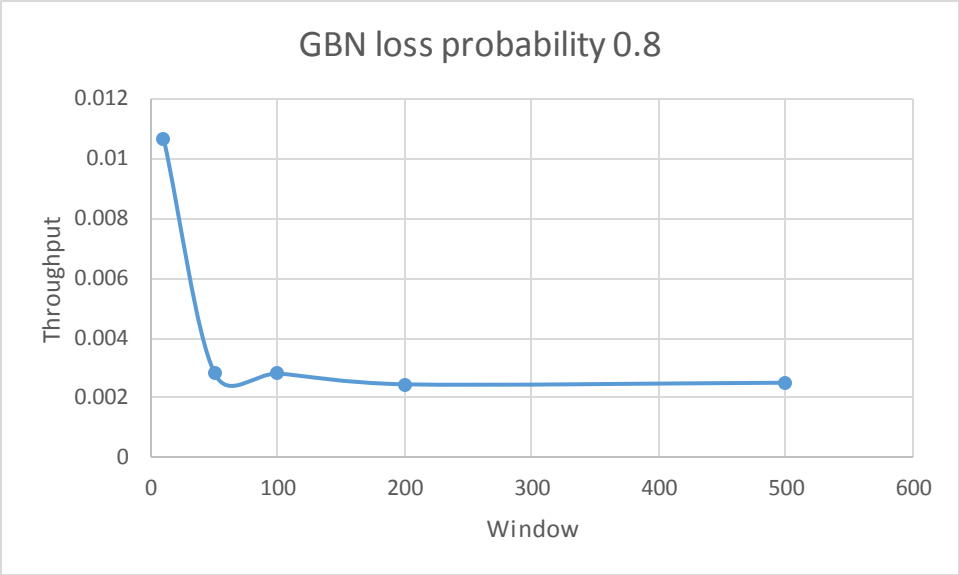


In selective repeat with increase in window size throughput increases as more number of packets are transmitted at a time.

3) GBN Timeout 17


GBN loss probability 0.2
0.025
0.02
0.015
0.01
0.005
0
Throughput
0
100
200
300
400
500
600
Window


GBN loss probability 0.5
0.0172
0.017
0.0168
0.0166
0.0164
0.0162
0.016
0.0158
0.0156
0.0154
Throughput
0
100
200
300
400
500
600
Window



In GBN throughput should decrease with increase in window size as when timer expires all packets in window are retransmitted. This increase is mostly evident at high loss rates because if loss is less retransmissions would be very less. So window would not effect performance much. Hence for 0.2 throughput is almost constant.

So we can conclude that in SR throughput increases with increase in window size and in gbn it decreases.