CSE 489/589
Programming Assignment 2
Reliable Transport Protocols

I, akumar39 have read and understood the course academic integrity policy.

Submitted By : Abhinav Kumar
Person # : 50247660

Table of Contents:

I. Brief description of the timeout scheme

Alternating Bit Protocol (ABT)

The message sent from network layer of A takes about 10 ticks to reach the receiver and 10 more ticks to send the acknowledgment back to the sender. Hence the total RTT would be 20 ticks and this is what we set the timeout value to.

A_timerinterrupt()

- If the timer exceeds the timeout , control is passed to this function.
- We send the packet again to layer 3 and start the timer ,keeping the timeout to 1 RTT again.

Go Back-N

In this protocol we set a global timer for a window of packets . This timeout is again set to 1 RTT i.e. 20 timer ticks . So when we send N packets i.e. the window size, it takes 10 ticks for all the packets to reach the receiver and 10 ticks for the cumulative acknowledgement to reach back the receiver.

A_timerinterrupt()

- When the timer for a packet expires, we send all the packets from the one with the expired timer till the last packet sent.
- We keep the timeout again as 1 RTT.

SELECTIVE REPEAT

In this protocol, we have set the timer to 50 ticks because we have designed an adaptive timeout mechanism in which each packet is set a timeout relative to the packet with the smallest timeout value. So, N packets are sent out from the network layer of A each with a timeout of 50 ticks, and the acknowledgment for each packet travels back to the sender in ½ RTT. As loss rate increases, SR is able to handle retransmissions efficiently so we allow enough time for acknowledgments to be received at the receiver.

A_timerinterrupt()

- When this function is called, we always send the first packet in the queue which is the packet whose timer has expired.
- We calculate the remaining time by comparing the timeout stored from the current time of arrival.
- We start the timer with the remaining time and send it to layer 3 of A.

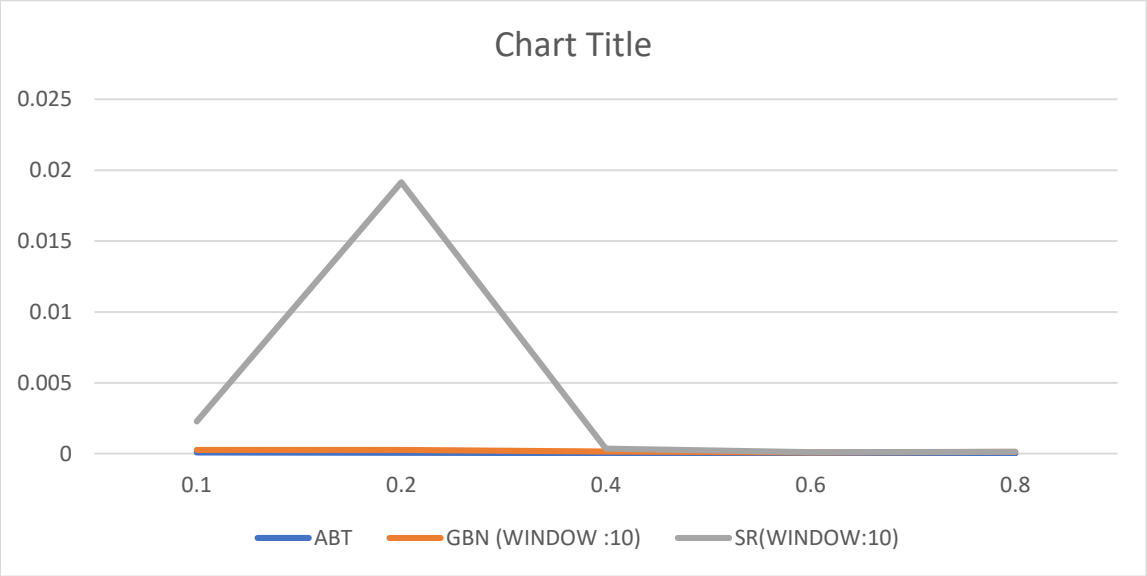
II. Brief description of the timeout scheme used in Selective Repeat

While implementing Selective Repeat, we are provided with only one hardware timer. We need multiple timers for each packet to be resent on a loss. The following design was undertaken to implement this scheme:

- Define a queue with members of type packet and a timeout variable.
- Whenever a packet leaves the sender, we enqueue the packet and calculate the timeout for the packet and store it.
- We infer if the timer is running by checking whether there is any packet in the queue else we start the timer and keep the timeout to 1 RTT.
- Whenever an ack is received we stop the timer for the packet and calculate the relative time for the other packets w.r.t to the packet with the oldest timeout i.e. the head element.
- We set the timeout to zero of the packet whose ack is received and dequeue the packet.
- When the timer of a packet expires , we call A_timerinterrupt is called.
- We again calculate the remaining time relative to the packet at the head of the queue i.e. the oldest unacknowledged packet by comparing the timeout stored to the current time.
- We start the timer again with the calculated remaining time and wait for an acknowledgement.

III Observations and Analysis

Experiment 1 : (Loss Probabilities : 0.1,0.2,0.4,0.6,0.8, Window size : 10)

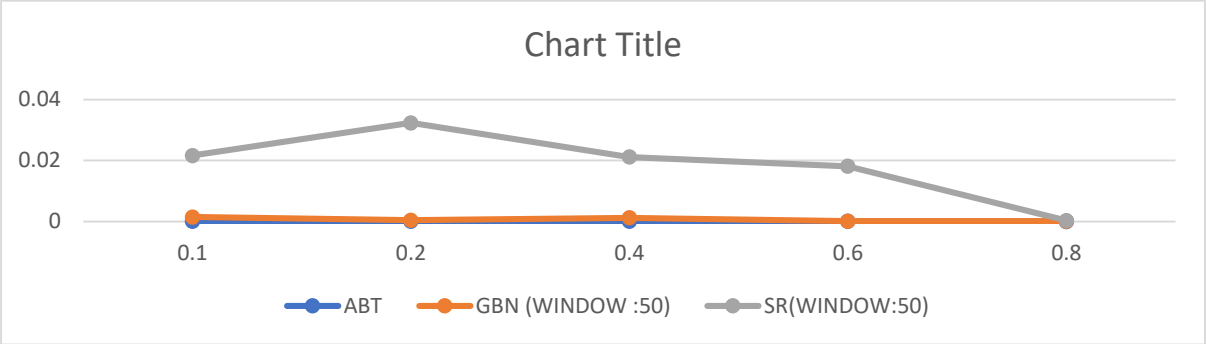


GBN and SR achieve higher throughput than SR but the throughput decrease as the loss rate increases.

Analysis :
ABT protocol will wait for one whole RTT i.e. 20 ticks for a packet before sending out another packet while GBN and SR will send out N packets and wait for one RTT before sending out another packet. Hence throughput is higher for GBN and SR. So if bandwith of the link A mbps, then ABT will achieve a throughput of 1/A while GBN and SR will achieve N/A Mbps of throughput.

When there is a packet loss, we need to resend the packet. So , ABT will wait an additional RTT before sending out another packet or timing out whereas GBN and SR ill have lesser throughput as N packets are being send N-m packets are being lost.

Experiment 1 : (Loss Probabilities : 0.1,0.2,0.4,0.6,0.8, Window size : 50)

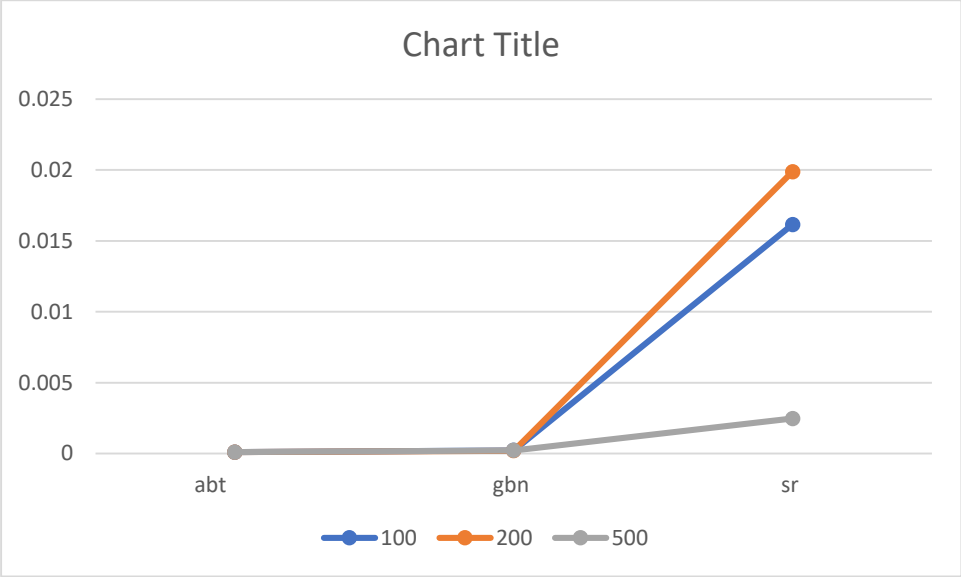


We infer that there is no major change when the window size increases, the throughput gives similar results , it decreases with higher loss.

Analysis :
SR will achieve a higher throughput as compared to other two when loss is higher as it only comprises of retransmitting the lost packet whereas in GBN we need to retransmit starting the packet with lost timer and this consumes bandwith.

Experiment 2:

- Loss Probability: 0.2

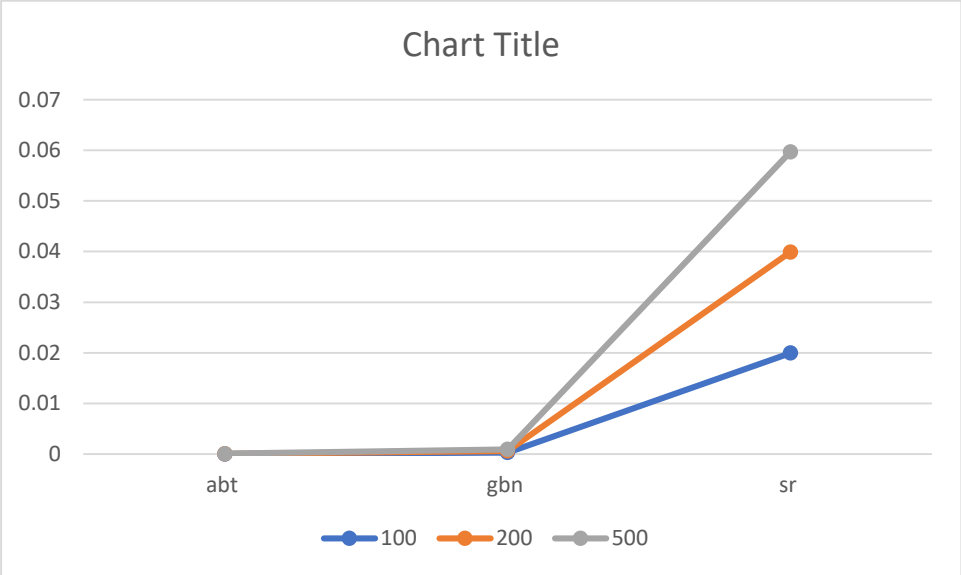


The throughput dips when the window size is 500 otherwise there is not much change for sizes 100,200 when there is less probability of loss.

Analysis :
As the window increases, more number of packets are being sent to the network and with almost no loss, packets are being transmitted at high speeds. Hence the output for window sizes 100 and 200. For window size

500, delays can cause the throughput to go down in GBN and ABT but for SR, the less number of retransmissions increases the throughput.

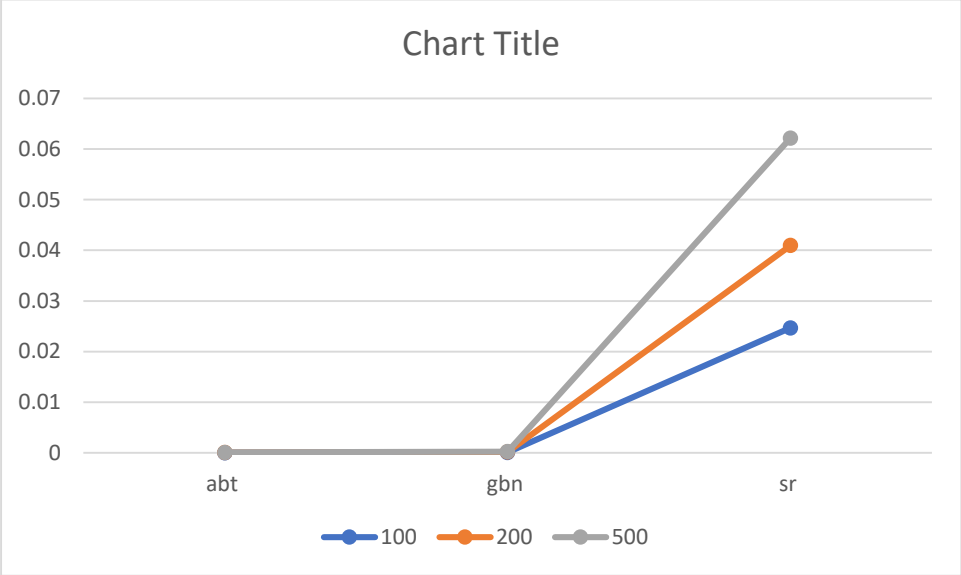
- Loss Probability 0.5



ABT and GBN perform worse when the loss rate is high. SR also incurs initial loss but handles loss well compared to the other two resulting in better throughput.

Analysis :
As network becomes more congested, loss rate increases, resulting in queuing of packets in the output buffer of the receiver and thus more retransmissions and lesser throughput . ABT has to deal with only one loss at a time but queuing time for other packets will increase. In comparison ,GBN will need to resend N packets even if there is one loss and more packets will be queued. So the throughput is affected. SR deals with loss better when we increase the window size as more packets are being acknowledgement even when there are unacknowledged packets in the window.

- Loss Probability 0.8



ABT and GBN perform worse when the loss rate is high. SR also incurs initial loss but handles loss well compared to the other two resulting in better throughput. But overall the throughput must decrease.

Analysis:
When the loss rate is very high, a number of packets will be buffered ,leading to more number of retransmissions. So, increasing the window size will decrease throughput as more packets are in the medium and the sender having no idea of the state of the packets retransmits them leading to more delays and congestion. The throughput will be extremely low when window size increases.