I HAVE READ AND UNDERSTOOD THE COURSE ACADEMIC INTEGRITY POLICY.

Project 2 : Analysis Report

Name : Yuze Liu

UbPerson No. : 50207903

UBIT : yuzeliu

4/7/2017

1 TIME OUT SCHEME

1.1 ALTERNATING-BIT-PROTOCOL

The implementation of this algorithm is exactly follow the slides in the class.
For A side, whenever side A decides to send a message to layer 3, start the timer. When A side received a valid(expected) acknowledgment from B side of the sent message, stop the timer. When A resend the message, start a new timer.
Because in this scheme, we will have only one message in transmission all the time, other message will be buffered until the current message finish transmission, so one timer is sufficient.
The increment time units of the timer is 15 time units.

1.2 GO-BACK-N

The implementation of this algorithm is exactly follow the slides in the class.
In the GBN algorithm, for the A side,as long as the sequence number of the next message being sent falls in the window, A will push the message into layer3. The timer begins when the sequence number equals the bsae of the window(first packet falls in the window), when the valid acknowledgment of this message has been received, then the timer will be stopped. If the acknowledgment of the packet after the base packet has been received, then the timer will be restarted.

1.3 SELECTIVE -REPEAT

The implementation of the SR algorithm is the extended of the GBN algorithm.
In SR algorithm, each packet will have their own timer, but we only have one hardware timer. So for each packet that has been pushed into layer3 from A side, the absolute send time is saved associated with each packet. When a message is resend, the send time will also be updated.
In the algorithm, the earlist sent but unacked packet has the highest priority to be sent in the sender queue.So the timer is actually always be associated with this high priority packet. So when time out happened, the packet with highest priority will be resend, then the absolute send time wil be updated, the second highest priority packet will now be the highest priority packet, the time should be associated with this packet now. If we have multiple timer, then we don't need to bother, however we only have one timer, so the time count down of this new highest priority packet will be :
Send time of this packet + Increment − Current time.
In the code, this is reflected as :
starttimer(0, packet.send_tiem + increment - local_time of simulator)

2 Performance Evaluation

2.1 Experiment 1

2.1.1 Plot

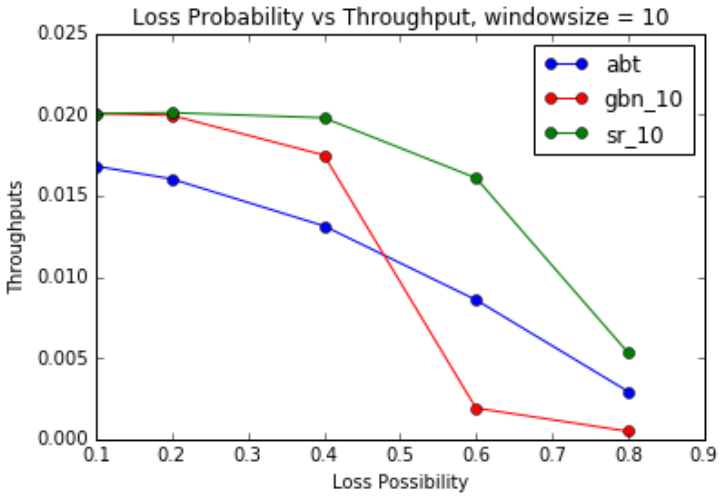

Figure 2.1: window size 10, Loss probability vs Throughput line plot

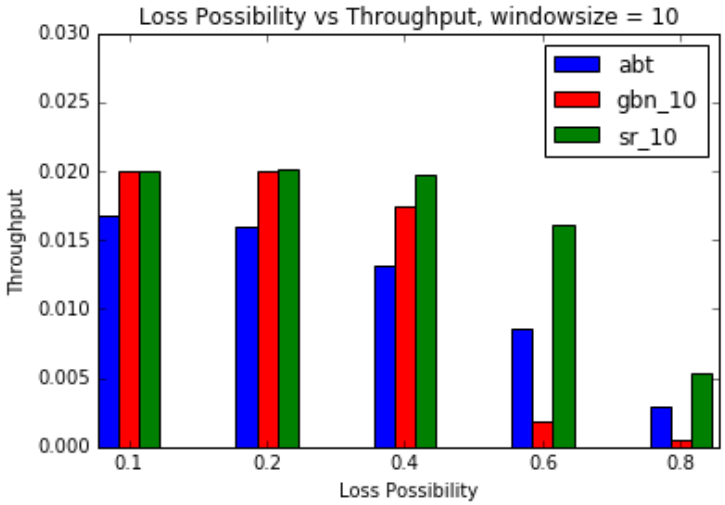

Figure 2.2: window size 10, Loss probability vs Throughput bar plot

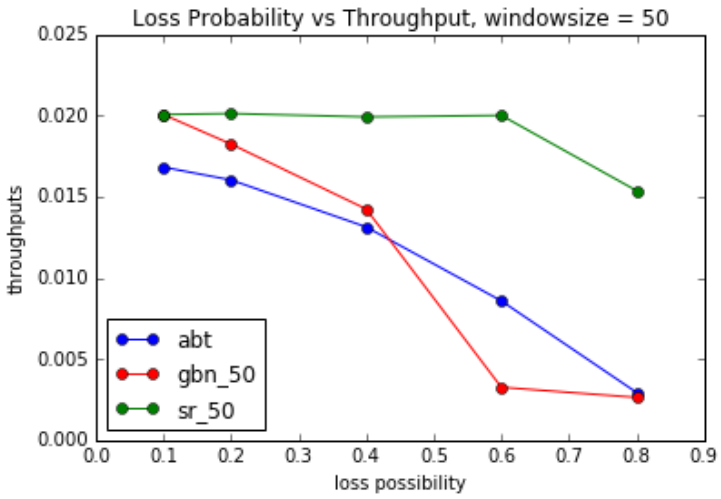

Figure 2.3: window size 50, Loss probability vs Throughput bar plot

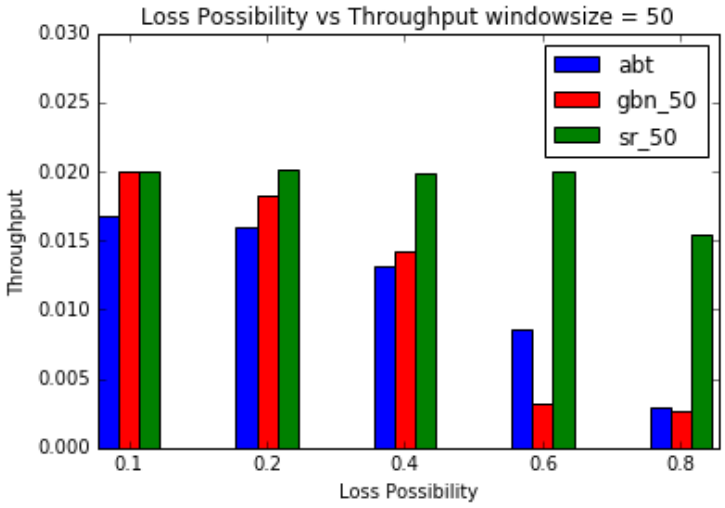

Figure 2.4: window size 50, Loss probability vs Throughput bar plot

2.1.2 Obervation and Analaysis

Observation :

- From all the four graphs above, we can see with loss possibility increasing, the throughput is decreasing in general.
- The SR algorithm performance is always better than ABT and GBN algorithm.
- When window size is 10, and the loss possibility is low, GBN and SR have similar performance, the throughput is almost the same. But when the window size is larger, the SR's performance is still better with high loss possibility and the throughput of GBN decreased.
- When loss possibility is high, the ABT's performance is actually better than GBN, especially when the widonsize of GBN is high.

Analysis :

- For all the experiment, the average time between messages from sender's layer 5 (-t) is 50 time units in the simulator. The maximum throughput is $1/50 = 0.02\ packet/timeunits$. So the maximum throughput is around 0.02 in the graph.

2.2 EXPERIMENT 2

2.2.1 PLOT

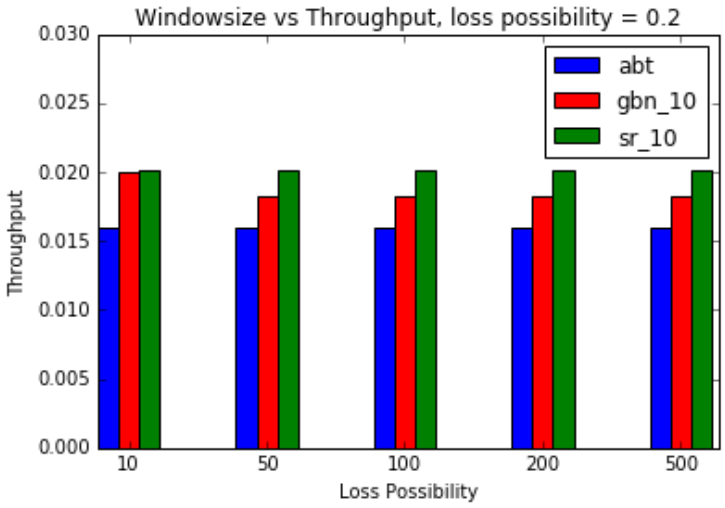

Figure 2.5: window size 10, Loss probability vs Throughput bar plot

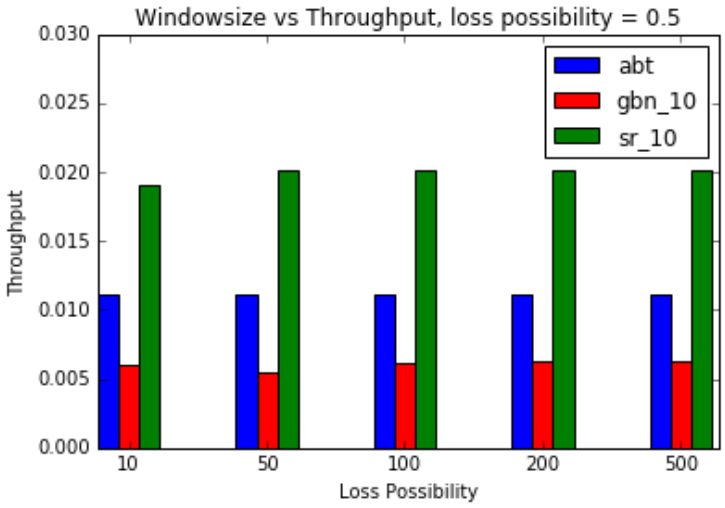

Figure 2.6: window size 50, Loss probability vs Throughput bar plot

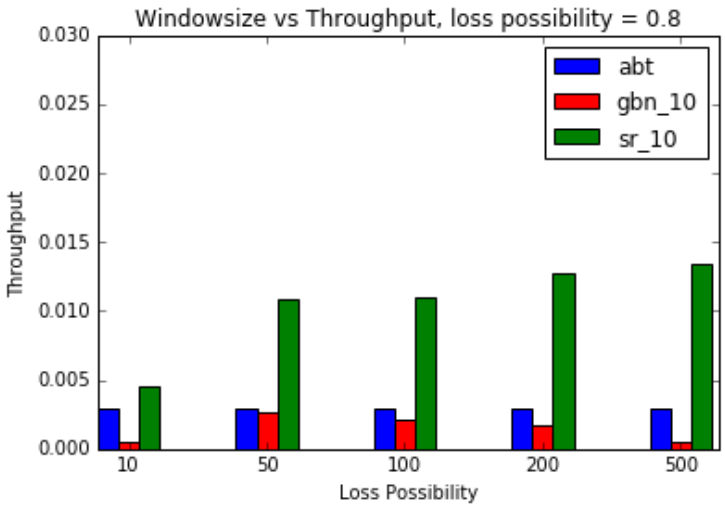

Figure 2.7: window size 50, Loss probability vs Throughput bar plot

2.2.2 Observation and Analysis

Observation :

- This experiment compares the throughput vs window size under three different loss possibility.
- The window szie has nothing to do with the ABT algorithm, so from Figure 2.5 to 2.7 we cna see that when th loss possibility is fixed, the throughput of the ABT is always the same.
- Figure 2.5 shoes that when the loss possibility is low(0.2), for GBN algorithm, when the window size increased, the throughput is actually decreased. But when the loss possibility is high, like the plot shos in figure 2.7, when window size is 50, 100 and 200, the throughput is higher than the window size is 10 or 500.
- When the loss possibility is low, the window size does not effect the throughput of the SR algorithm, the theoughput is almost 0.2 which is the maximum

throughput in this experiment.

- When the loss possibility is high, the SR algorithm can not achieve the max throughput, but when the winow size is larger, the throughput increased.

Analysis :

- For the SR algorithm, when the window size is larger, which means more packets are actually in transmission since each packet has its own timer and as long as the packet is unacked and falls inside the window, the packet can be sent from A side to B side. So larger window size means more chance for each message to be sent from A to B which means more chance for a packet to be successful delivered. So the throughput can be increased

3 SUMMARY

In general, when the loss possibility is low and the corruption is low, it doesn't matter what algorithm you will choose, the throughput is almost the same. But when the loss possibility and corruption possibility is high, the SR seems more stable. And the throughput can be increased when the window size is larger.
During the simulation, several things need to be mentioned. First thing is when the window size is larger, it takes hours and hours to run the SR and GBN algorithm and the running time is alo related to the timeout increment I set in the code. But I think this is something related to the simulator itself, that doesn't mean it will take more time in reality when using this algorithm.
Also, one drawback of SR is though increase window size can increase throughput, it can also caused problem when the window size is too larger. From the lecture, we know that the window size should be less or equal to the half of the sequence number. So SR is more complex in reality since the maximum sequence number my be differed and to choose a good window size is very important.

4 SUBMISSION FOR PROJECT

All the experiment data and the python script has also be attached in the tar file if you would like to check the validity of the graph.
Thanks!