"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".

University at Buffalo, State University of New York

Modern Networking Concepts
CSE-589

Programming Assignment-2

REPORT

Prof. Dimitrios Koutsonikolas

-Submitted by
Srinivasan Rajappa
50134691
srajappa@buffalo.edu

Acknowledgements

I would like to sincerely thank my Professor Dimitrios Koutsonikolas to have give me the opportunity to perform this experiment- Simulation of reliable data protocols viz. Alternating-Bit, Go-Back-N, and Selective repeat. This experiment helped me to understand the concepts and nuances of these protocols that work in Transport layer of computer networks. There are various instances where I was forced to think about the basics of sending packets from one side to another. It helped me to understand the perspective of how these protocols are designed and how protocols in future would be designed based on the principles which were used in the above mentioned protocols. I would also like to mention that Professor Dimitrios was committed in solving our problems and should great enthusiasm in his teaching. He constantly replied to all the students over the wordpress blog without much delay. This was also a factor that motivated to do more and made me persistent. I would also like to thank the Teaching Assistants viz. Swetank Saha and Ramanujan Sheshadri who despite incessant doubts replied promptly. In addition to this I would like to mention the role played my peers viz. Rakesh Balasubramanian, Vinayak Karuppasamy, and Mithun who helped when I faced difficulty in understanding some of the concepts related to the implementation. The notes and the Finite state machine diagrams in the book Computer Networking: A top-down approach by James F. Kurose, Keith W. Ross was indeed helpful. Although, I couldn't complete all the objectives in the project but I tried giving my individual effort. In this process I enjoyed making several revisions of my work and also learnt many new things. This project assignment was a fulfilling and rewarding in it's own way.

-Srinivasan Rajappa

Introduction

Stop and Wait

Also called as Alternating bit protocol involves sending a packet in a network and wait for the acknowledgement to arrive. It is known as Alternating bit because alternate bit values i.e. 0 and 1 are used to validate the state of the protocol. The data from application is always buffered if it does not comply with the state which is prevalent. In events of packet corruption or packet loss there are measures to ensure that the same is not passed to the upper layer. However, this protocol offers low throughput/Goodput when there is packet corruption and packet loss in the network.

Go-Back-N

Here packets are sent over a window of some numerical size i.e. number of packets allowed to be sent over network in a burst. The acknowledgement of each packet involves the progressive movement of the window. Such a setup better utilizes network resources as many packets are sent at once. In case there is a loss of packet or packet corruption, packets over the entire window is sent to the receiver.

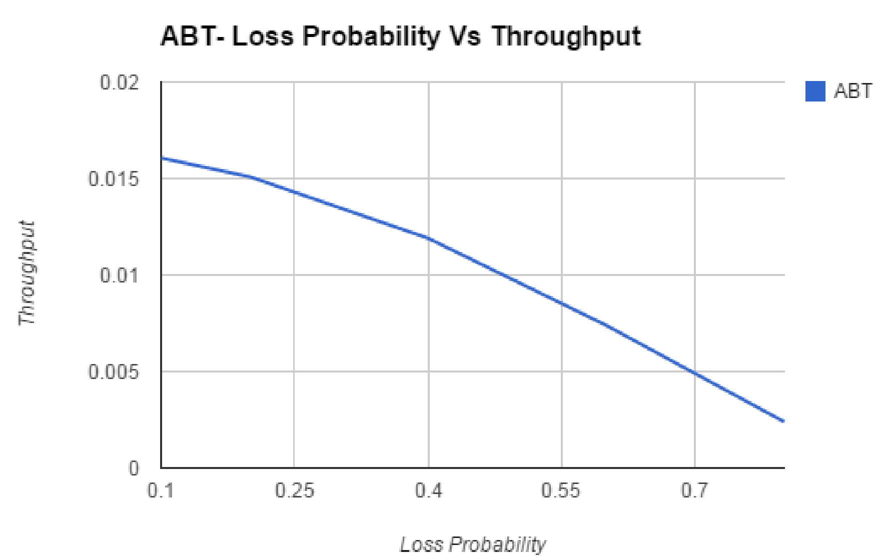
Selective Repeat

This protocol is similar to the Go-Back-N protocol but the difference between the two is instead of sending an entire window in case of packet loss or corruption, the sender here will only send the packet which was involved in packet loss or corruption. This significantly increases the throughput/Goodput and provides better results.

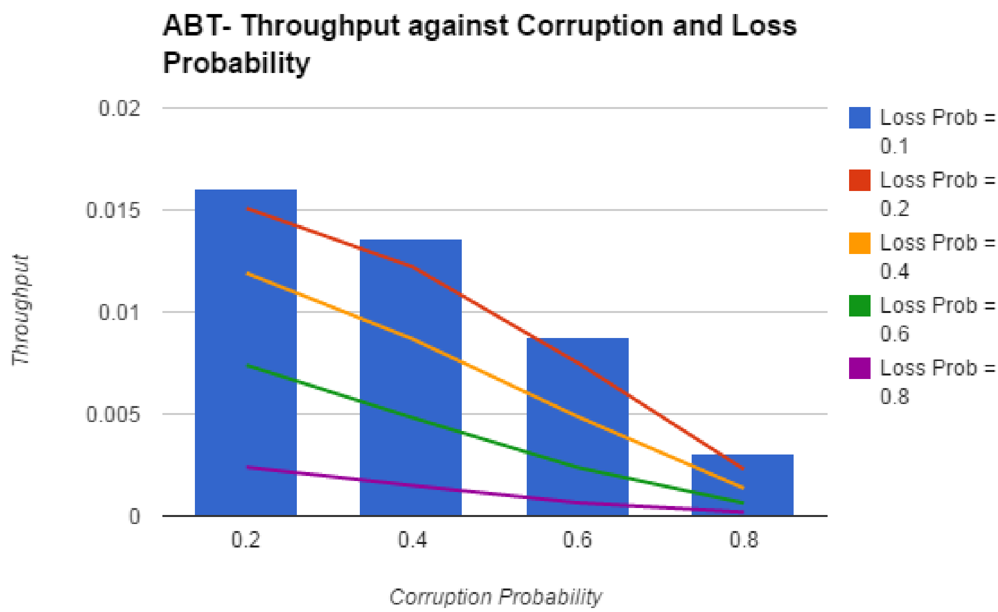
Performance

Stop & Wait

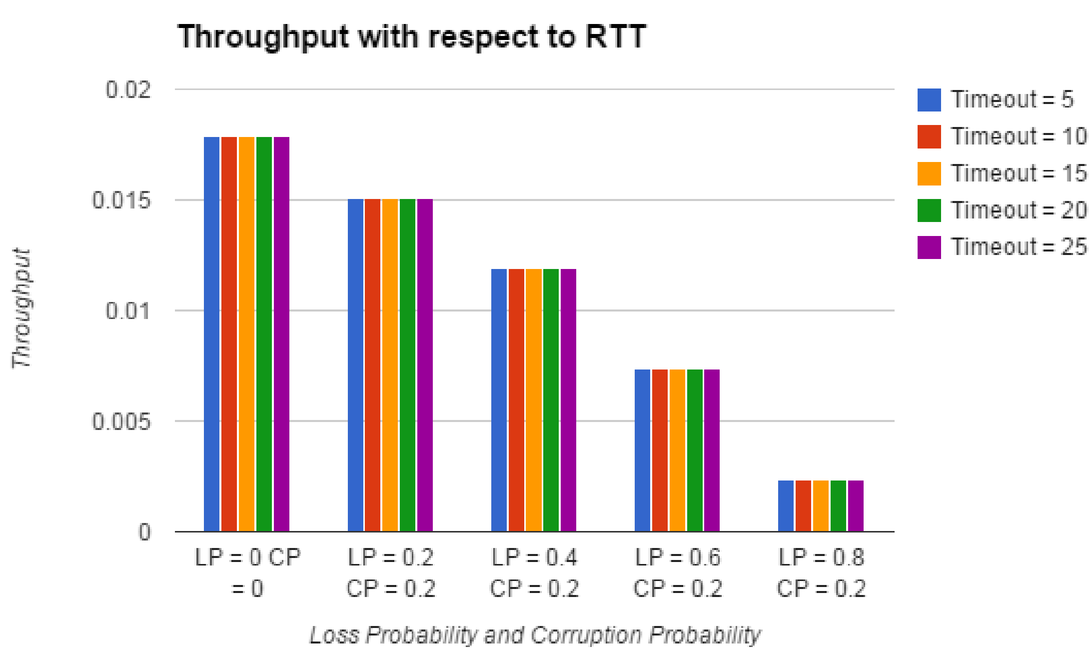
As mentioned earlier the throughput decreases as the loss probability and (or) the network corruption increases. The following graph shows the throughput against a varying loss rate and a fixed corruption rate probability of 0.2. I have derived the results after sending 1000 data packets from application.



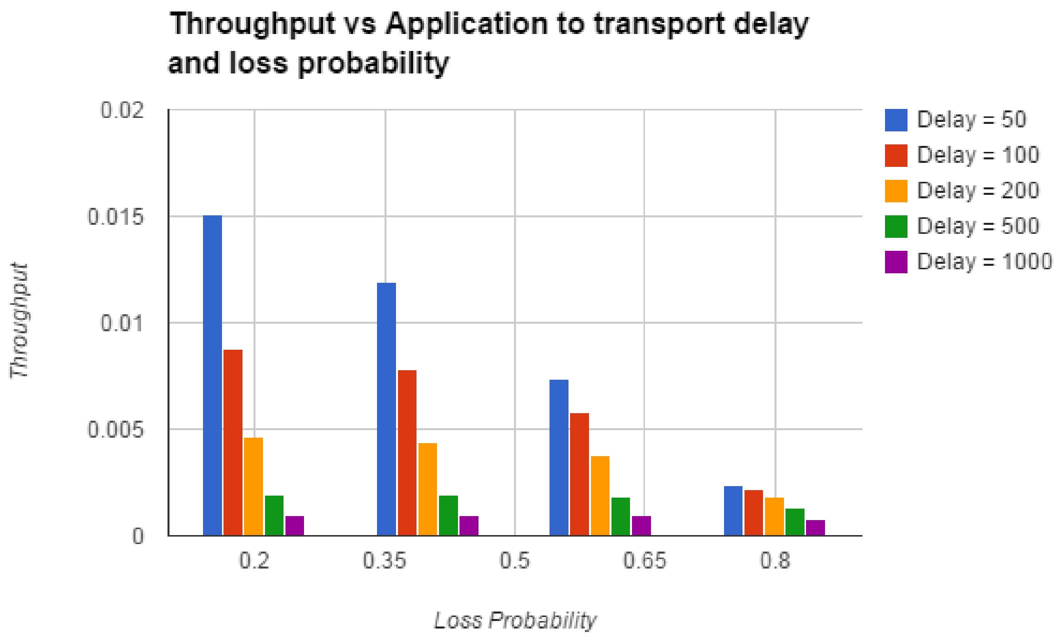
Now I also have analysed the relationship between how the throughputs vary with respect to corruption rate and loss rate unison. One can observe the sudden plunge in throughput when the packet is sent into a medium which is both corrupts packets and congested.



I also examined if the Round trip time (RTT) had an impact on the proper functioning. I sent across 1000 messages and varied the packet loss rate keeping the corruption rate constant. By looking at the chart below one can come to the conclusion that RTT does not determine and it should not be a parameter worth considering.

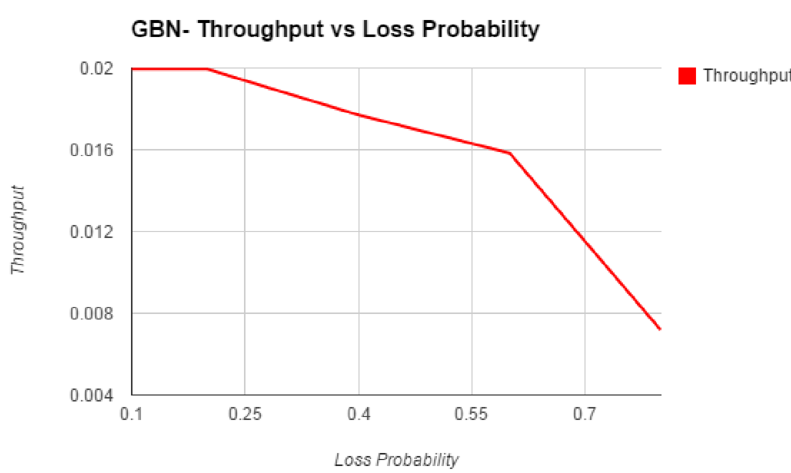


I examined how time taken for application to send to transport layer impact the throughput. It is obvious that the throughput will decrease if the delay is larger as throughput is inversely proportional to the time taken to send packets to the other side. But the important observation in the figure below is the relation between the loss and the delay, it can be seen that for higher delays the throughput is roughly same for any value of loss probability. Thus providing a conclusion that higher delay will compensate and always provide a consistent performance even in a congested network.

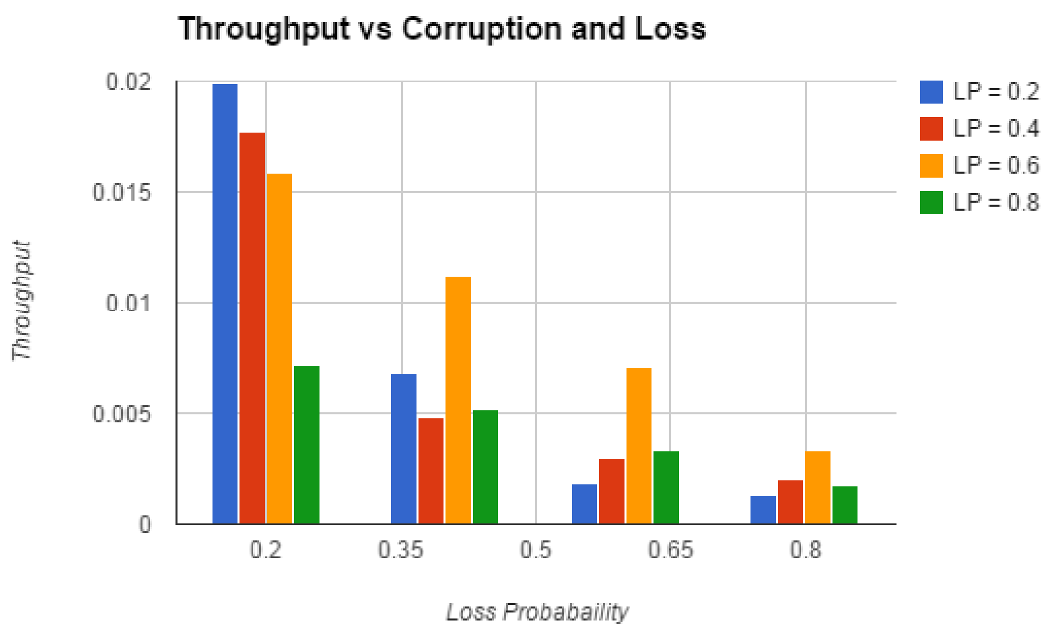


Go-Back-N

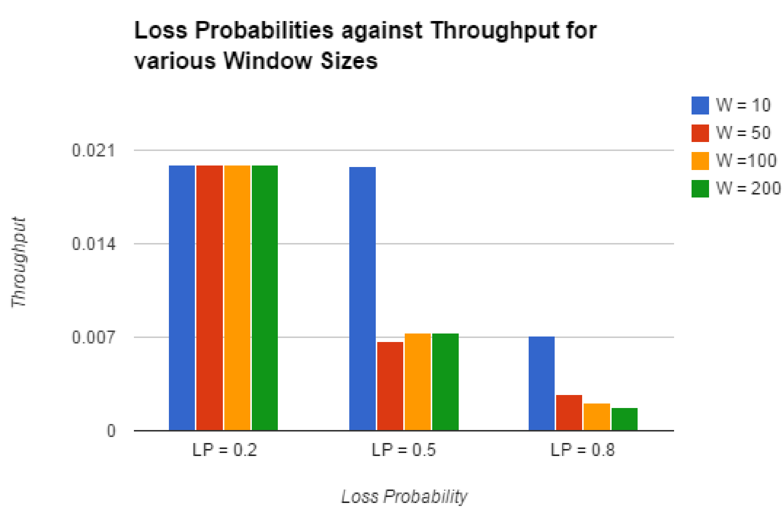
Go-back-N promises better performance with respect to Stop and wait. The use of pipelining makes this model efficient. On performing the tests I found the throughput decreasing against increasing loss probability. The following chart shows the exact same phenomenon. The sender sends 1000 packets where the window size is 10.



On varying corruption probabilities with various values of loss probabilities I get the following figure. There is an interesting observation here as the throughput varies across a range of corruption and loss probabilities. The anomaly is when the loss probability is 0.6, 0.2. The 0.6 value gives great throughput when the corruption probability increases. I think given the particular window size (here 10) and the timeout/RTT value (15) the results represent a conducive environment where the exceptional results are noticed. Logically, if the corruption rate and loss rate increases the number of retransmissions increase and effectively bringing down the throughput. But, if such ground rule conditions are violated in form of results here then we can give room to tweaking and calculating the maximum permissible limit of such rates up to which we can expect good throughput.



On observing the change with respect to the Window size I get the following response. I avery the loss probability values for the window size 10, 50, 100, and 200.



On observation one can clearly observe the high throughput provided by window of smaller size, here it is 10. For LP = 0.5 one finds an anomaly where Window size of 100 and 200 provide better throughput than 50. This can be attributed to the inherent variation in the simulation that seem to provide such results as it is logical that if there is a high probability of loss then the number of retransmissions will increase and a larger window will have to flush even more packets thus degrading the throughput.

Design

***Stop and Wait*:**

As seen above the timeout value or the RTT has had not much impact on the throughput values, thus from design perspective this is a straightforward algorithm to implement this ARQ in the simulator.

Go-Back-N:

Here I used to send all the packets in the window in case of timeout or when packet corruption has been detected. This brought the throughput significantly down. I then only sent the packet that were acknowledged this provided better results. Thus some finer details from the notes in the course book was helpful.

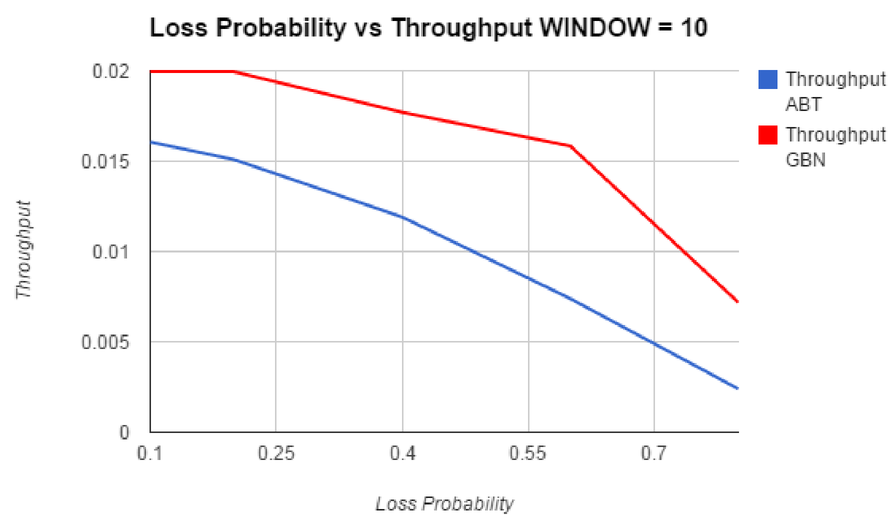
Selective Repeat:

Although the implementation of this protocol has not been completed, but still I have a running version of the same which works on the principles where only single hardware timer is being used.

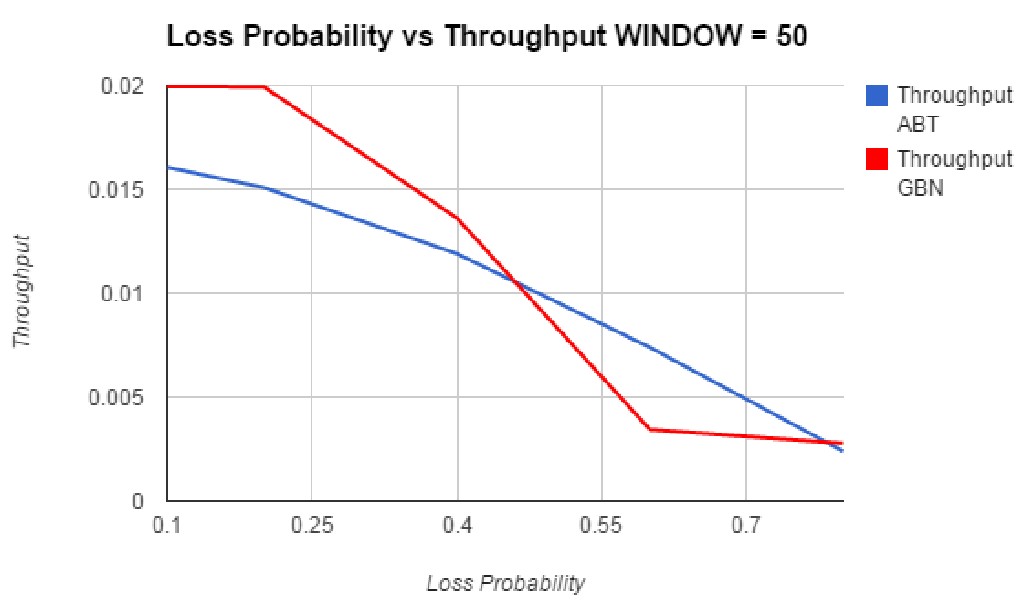
1. I created a provision for storing time information (in a structure) about the each packet which needs to be sent to the network.
2. The time value is typically TIMEOUT + current time (which is found by the global variable time).
3. Now, in any event viz. Acknowledgment reception, the current time and the ones in the structures are being compared (subtracted). The one have a comparatively less value will qualify for the packet under scrutiny and accordingly the action is being taken either to retransmit or accept the acknowledgment and move the window forward.

Experiment - Number 1

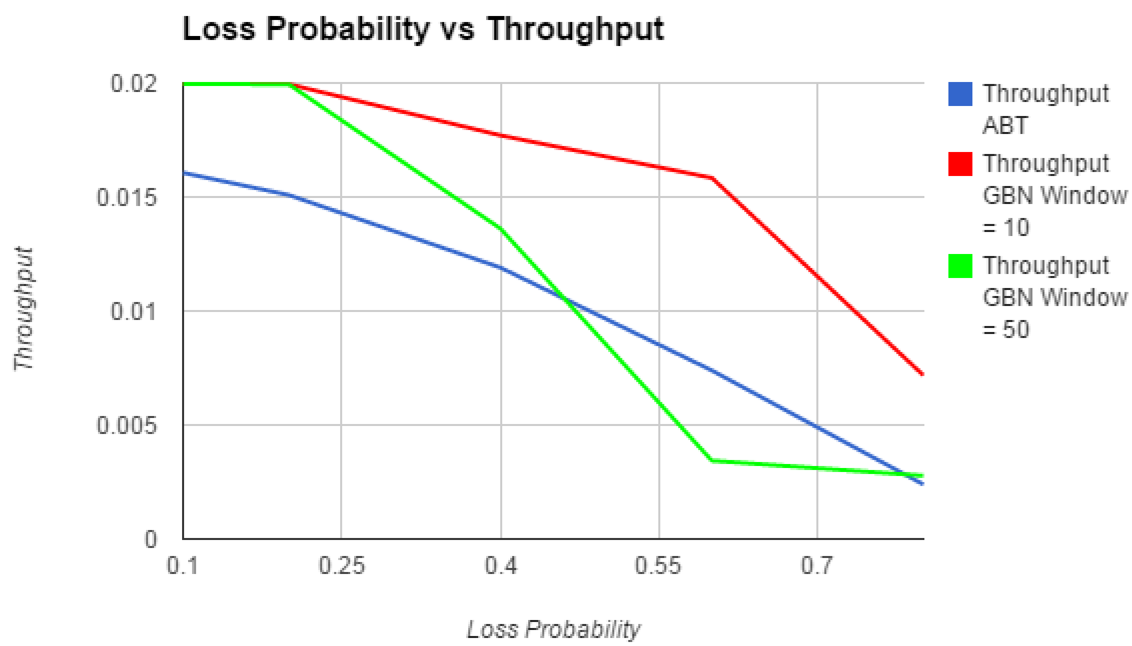
Comparing the 3 protocols' throughputs for the loss probabilities of 0.1, 0.2, 0.4,0.6, and 0.8. I have also used the same probabilities for different window sizes for Go-Back-N protocol.



One can clearly notice how the throughput of GBN is significantly higher than the ABT protocol. The above GBN protocol had a window size of 10. Now once I use window size of 50 I have the following result. The results show for higher window size in GBN the throughput decreases for higher rates of Loss probability.

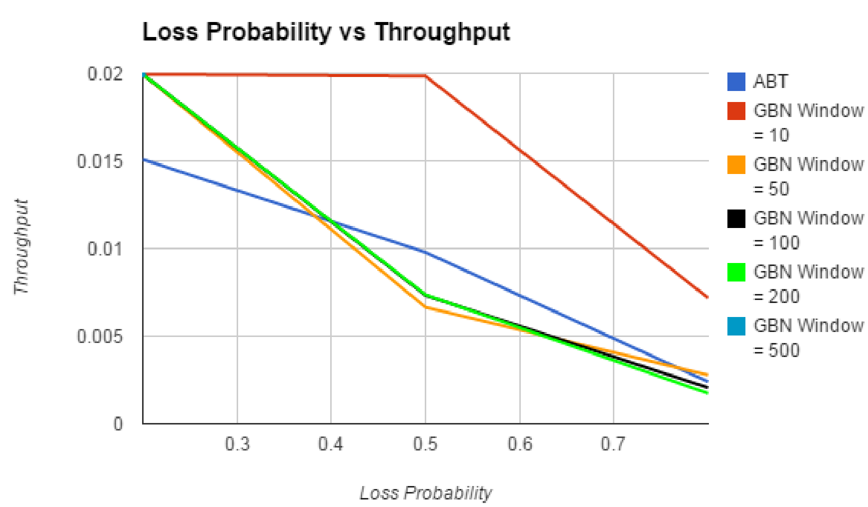


For more analysis I have included the above two graphs in one below to realize the fact that WINDOW size does make an impact on throughput. The fact is the throughput might improve on smaller window size but the time taken to transmit all these packets will be significantly low. While on the other hand large window sizes although involve less throughput but the chances that the time taken to send all the packets will be minimized as multiple packets are sent in the same channel.



Experiment 2

Here I exhibit the results for throughput for various Window sizes against a set of Loss Probabilities. The Window Size varies as 10, 50, 100, 200, 500 while the loss probabilities varies as 0.2, 0.5, 0.8. This effectively shows that a small sized window can deliver better throughputs when the loss rate in the network is high.



Conclusion

1. The results which we see above represent the underlying nuances of the behavior of these protocols based on the corruption and loss rates in the underlying network.
2. But repetitively performing such tests on a live network one can create a scheme to counter such situation or crisis.
3. Throughput is essential for today's Internet and we can provide such provisions if we we can come up with algorithms that can send packets in a single burst as well as take care of all the vagaries present in the network layer below.
4. A larger window only promises better results when the loss rate and corruption rates are minimum. In case of loss the results are not satisfactory.
5. The RTT has minimum impact on the throughput of the protocol.