Networks Assignment 4: Selective Repeat and Go Back N Protocols

Rishika Varma K
CS18B045

May 2021

Index

Aim....2
Introduction....2
Experiment Details....3
Results....4
Observations....5
Additional Thoughts....6
Learning Outcomes....6
Conclusion....6
References....6

1 Aim

To implement a simulation of both Selective Repeat and Go Back N Protocols using UDP and taking note of the round trip times with variation in factors to observe how these protocols differ.

2 Introduction

In computer networking, a reliable protocol is a communication protocol that notifies the sender whether or not the delivery of data to intended recipients was successful.Reliable protocols typically incur more overhead than unreliable protocols, and as a result, function more slowly and with less scalability.A reliable delivery protocol can be built on an unreliable protocol.Selective repeat and Go back N protocols are instances of sliding window protocols used for re-transmissions in reliable communication. In our experiment, they are built over UDP. Some differences between SR and GBN are as follows:

Basis for Comparison	Go-Back-N	Selective Repeat
Basic	Retransmits all the frames that sent after the frame which suspects to be damaged or lost.	Retransmits only those frames that are suspected to lost or damaged.
Bandwidth Utilization	If error rate is high, it wastes a lot of bandwidth.	Comparatively less bandwidth is wasted in retransmitting.
Complexity	Less complicated.	More complex as it require to apply extra logic and sorting and storage, at sender and receiver.
Window size	N-1	<= (N+1)/2
Sorting	Sorting is neither required at sender side nor at receiver side.	Receiver must be able to sort as it has to maintain the sequence of the frames.
Storing	Receiver do not store the frames received after the damaged frame until the damaged frame is retransmitted.	Receiver stores the frames received after the damaged frame in the buffer until the damaged frame is replaced.

Basis for Comparison	Go-Back-N	Selective Repeat
Searching	No searching of frame is required neither on sender side nor on receiver	The sender must be able to search and select only the requested frame.
Use	It more often used.	It is less in practice because of its complexity.

3 Experiment Details

3.1 Simulation Setup

The codes are written in $C++$, and so is compiled using $G++$. The algorithm involves using pthreads and mutexes. There are 4 code files, 2 each for selective repeat and go back n protocols corresponding to the sender and the receivers. To simulate dropping of packets in both protocols and packet lengths in sr we use random bernoulli and uniform distribution generators respectively.

3.2 Entities involved and functions in each entity

In both protocols the sender files there are 3 main function threads involved apart from main. Main calls 2 of these threads and continues to generate packets periodically until maximum packets have been acknowledged. At the end it prints information that has been gathered during experiment.

The first function thread is psend and this function is involved in sending the packets that have been generated by main. It calls another function thread presend.

The next function thread is precv and it is responsible for processing the packets whose acks have been sebt by the receiver.

The presend function thread mentioned earlier periodically (for the first 10 packets time interval is 300 ms for SR and 100 ms for GBN and for the rest of the packets it is 10*RTT) checks if the ack for that packet has been recieved and if it was not then it sends the packet again. If number of resends exceeds 10 and 5 for sr and gbn respectively, the script quits and gives results from existing information.

Apart from this the scripts contain auxillary functions for calculating averages for the output.

In both receiver files the only function present is the main and this receives the packets from the sender, keeps or drops it according to the error

rate and accordingly processes and sends the acknowledgement to the receiver.

4 Results

For this experiment the values of packet generation rate, packet length and random drop probability are varied to see observe the changes. The window size, buffer size and maximum acknowledged packets values in all the cases that I took are 4, 400 and 100 respectively.

Table 1: packet generation rate is 20

For SR:

Max Packet length/Drop probability	0.001	0.00001	0.0000001
256	Avg RTT=76.0875 Retransmission ratio=1.015	Avg RTT=65.82 Retransmission ratio=1.045	Avg RTT=76.1025 Retransmission ratio=1.1125
1500	Avg RTT=21.16 Retransmission ratio=1.0225	Avg RTT=67.4925 Retransmission ratio=1.07	Avg RTT=65.2475 Retransmission ratio=1.075

For GBN:

Max Packet length/Drop probability	0.001	0.00001	0.0000001
256	Avg RTT=60.9175 Retransmission ratio=1.045	Avg RTT=82.26 Retransmission ratio=1.2	Avg RTT=93.7325 Retransmission ratio=1.225
1500	Avg RTT=36.225 Retransmission ratio=1.0325	Avg RTT=41.2825 Retransmission ratio=1.115	Avg RTT=58.71 Retransmission ratio=1.1425

Table 2: packet generation rate is 300

For SR:

Max Packet length/Drop probability	0.001	0.00001	0.0000001
256	Avg RTT=47.1075 Retransmission ratio=1.0375	Avg RTT=46.285 Retransmission ratio=1.1175	Avg RTT=37.99 Retransmission ratio=1.075
1500	Avg RTT=66.8175 Retransmission ratio=1.145	Avg RTT=74.815 Retransmission ratio=1.13	Avg RTT=72.0375 Retransmission ratio=1.1175

For GBN:

Max Packet length/Drop probability	0.001	0.00001	0.0000001
256	Avg RTT=51.0275 Retransmission ratio=1.0975	Avg RTT=60.23 Retransmission ratio=1.06	Avg RTT=25.21 Retransmission ratio=1.015
1500	Avg RTT=85.4275 Retransmission ratio=1.1325	Avg RTT=90.2125 Retransmission ratio=1.1825	Avg RTT=60.0325 Retransmission ratio=1.0875

5 Observations

From the results we can derive some observations as follows:

In general, there does not seem to be a fixed pattern in the variation of retransmission ratio with packet length. This can be understood since retransmission mainly depends on packets dropped.

However the packet length does seem to affect RTT such that as packet length increases RTT also increases. This could be due to needing longer time to send, receive and process longer packets.

There also seems to be a relationship between Retransmission ratio and Drop probability such that with increase in probability retransmission ratio also increases although this is not very pronounced in some cases where I believe the system itself is slow and so by the time a packet is acknowledged another is sent.

On comparison of values of SR and GBN, in general GBN has considerably higher retransmission ratio and this can be explained due to the fact that in GBN all the packets in the window are resent each and every time even though unnecessary. It is observed that this difference is particularly more pronounced when drop probability is lower.

With change in drop probability, RTT remains comparable for the most part with very slight variations in most cases.

The RTT also increases in GBN as compared to SR in most cases. This could possibly be due to more amount of sending and receiving of packets

that are unnecessary which may delay the receiving of packet that is actually required.

6 Additional Thoughts

Finding trends in this experiment was much harder than anticipated due to system factors also affecting the way in which the protocol panned out and this also differed in each run due to the random adding and dropping of packets. Some situations led to a bigger disturbance in packet reliability compared to others. I think that a better way to analyse is by running the program for the same values for a large number of times and taking the average of all the results. This will give a more accurate scenario to compare results.

7 Learning Outcomes

From this experiment, I was able to learn about both go back n and selective repeat protocols and how they are implemented. I was also able to see how both of them differ in various circumstances and in which situations what is used. I also saw the ways in which threads and Mutexes are used and its implementation. I learnt about UDP as well this time in single client setting.

8 Conclusion

Through this experiment we see the various aspects of reliability protocols and how they are used in networks. Although it is not a full fledged model it captures the essence of the underlying algorithm by using UDP and similar mechanism. Through this simulation we could see how the packet reliability is maintained in different situations.

9 References

https://en.wikipedia.org/wiki/Communication_protocol
https://techdifferences.com/difference-between-go-back-n-and-selective