Computer Networks
CS3205
Assignment 4

Go-Back-N and Selective Repeat Protocols

Soumith Basina - CS18B005

Date Submitted: May 11th, 2021

Instructor ..... Prof. Siva Ram Murthy

Contents

1 Objective

The objective of this assignment is to implement Go-Back-N and Selective Repeat, two reliable data transmission protocols and observe the differences.

2 Introduction

Go-Back-N and Selective Repeat are two protocols that aim to achieve reliable data transfer without compromising data transfer rate. They use frame pipelining to send multiple frames without waiting for acknowledgements (ACKs) to achieve better link utilisation and throughput. Both of the protocols use sliding windows at sender and receiver to achieve this, albeit in slightly different ways.

Go-Back-N has Sender Window Size (SWS) as N and Receiver Window Size (RWS) as 1. It uses cumulative acknowledgements. Receiver only can receive frames in-order, any frame that is out-of-order is discarded. In the case of a dropped frame, sender must resend the entire window.

Selective Repeat has SWS and RWS as N. It uses independent acknowledgements. Receiver now has a buffer and can receive frames out-of-order. This avoids re-sending the entire window every time a frame is dropped, which prevents unnecessary re-sends.

3 Experimental details

Both the algorithms have 3 adjustable parameters

- `PACKET_GEN_RATE` - The rate at which new packets are generated.
- `PACKET_LEN` - Length of the packets
- `PACKET_DROP_PROB` - Probability that the packet is dropped.

3.1 Experimental/Simulation setup

These parameters are passed to the respective executables and run in separate terminal windows. The flag `-d` can be used to turn on debug mode, which will print information about all the packets that are successfully sent/received. At the end of the session, the sender will print retransmission rate and average RTT or it will terminate prematurely if the no. of attempts to send a packet exceeds 10.

3.2 Entities involved and functions in each entity

The code for the implementation of Selective Repeat is in `SenderSR.cpp` and `ReceiverSR.cpp` and the code for the implementation of Go-Back-N is in `SenderGBN.cpp` and `ReceiverGBN.cpp`. Inputs are given to the executables as indicated in the problem statement.

Both of the senders have 3 threads running in parallel. The main thread which sends the packets to the receivers, a thread for the `listenACK()` which is in an infinite loop to

receive any incoming ACKs and marks the packet in the buffer and a thread which periodically repeats execution of `generatePackets()`, which generates a new packet to be sent and stores it in the buffer.

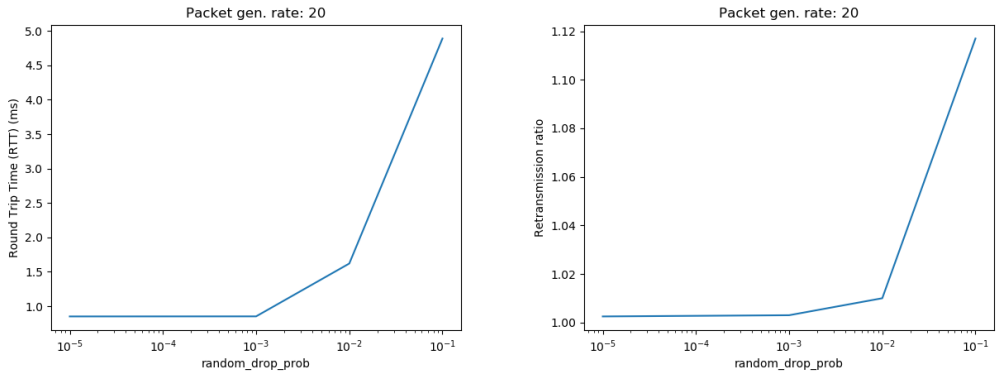
Receiver in Selective Repeat has 2 threads, one main thread which manipulates the sliding window as the packets are received from the buffer and sends ACKs accordingly, a thread for the `listenforPackets()` which is in an infinite loop to receive any incoming packets and drop the packet or store it in the buffer according to the error rate.

Receiver in Go-Back-N has only a single thread where it receives and moves the sliding window and sends ACKs.

Mutexes are used for the buffer to prevent any race conditions as multiple threads try to access the buffer at a time.

4 Results and Observations

4.1 Selective Repeat



(a) Drop prob. vs. RTT

(b) Drop prob. vs. Retrans. ratio

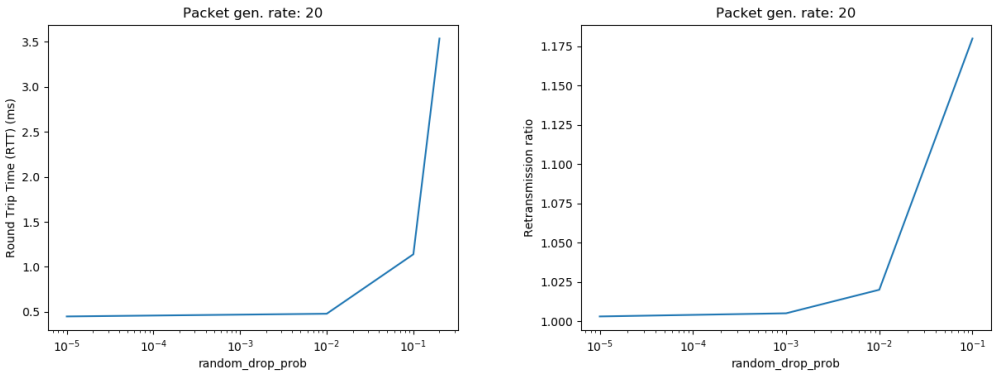
Figure 1: Plots for Selective Repeat

The x-axes of the plots are in logarithmic scale.

As we can see, the plot for RTT is increasing as drop probability increases due to the additional retransmissions required to compensate for the drops. As drop probability increases, the number of retransmissions required also increases resulting in the increase of retransmission ratio as shown on the right.

Increase in packet generation rate does not affect RTT and retransmission ratios.

4.2 Go-Back-N



(a) Drop prob. vs. RTT

(b) Drop prob. vs. Retrans. ratio

Figure 2: Plots for Go-Back-N

The x-axes of the plots are in logarithmic scale.

As we can see, the plot for RTT increases more sharply as drop probability increases when compared to Selective Repeat. This is because for a drop, the whole window needs to be retransmitted in the case of Go-Back-N, resulting in unnecessary retransmits for the packet. Similarly due to the increased number of retransmissions, retransmission ratios are higher when compared to Selective Repeat.

Similar to SR, increase in packet generation rate does not affect RTT and retransmission ratios.

4.3 Comparision

- Increase in Round Trip Time is sharper in the case of Go-Back-N due to the increase in retransmission of the entire window, making it a poor choice if the network is unreliable.
- Retransmission ratios are higher in the case of Go-Back-N as entire window needs to be retransmitted for every drop, wasting the network bandwidth.

As we can see, SR is a better choice for transmitting in unreliable conditions as it takes less network bandwidth and does not need the unnecessary retransmits. But the implementation of SR is more complex as a result of having a receiver buffer.

5 Learnings

This assignment has been helpful in

- Learning some insights on how reliable data transfer can be achieved using unreliable communication channels using Selective Repeat, Go-Back-N.
- Learning the differences between Selective Repeat and Go-Back-N.
- Learning some basic threading and usage of mutexes in C++.

6 Conclusion

Selective Repeat and Go-Back-N are two ways of achieving reliable data transfer using unreliable communication channels. With this assignment, by implementing both protocols, we have learned that Selective Repeat can be a better choice in the case of an unreliable network having high error rates and it can help save network bandwidth.