Assignment 4

Meesa Shivaram Prasad (CS18B056)

1. Selective Repeat Protocol :

Aim : To implement the Selective Repeat reliable transmission protocol and measure the round trip delays.

Introduction :

Selective Repeat protocol is a sliding window protocol , it is more efficient than GBN , major problem in GBN is it sends the entire window when error occurs whereas selective repeat attempts to retransmit only those packets that are actually lost due to errors.Independent acknowledgements are used in selective repeat this makes sending packets more efficient but sorting mechanism at receiver side adds to more complex implementation.

Experimental details:

- Experimental/Simulation setup :
    - Here we are running two separate programs at same time on two different hosts : a sender program that generates and transmits packets and receiver that accepts packets and transmits the acknowledgements to the sender.
    - The receiver randomly decides whether to drop a packet or not with a random_drop_probab.
    - And finally we measure the round trip average time and retransmission ratio , For sending of packets we follow the selective repeat protocol.
    - Communication between the sender and receiver is through UDP sockets
    - The receiver is set up as a UDP server, and the sender is set up as a UDP client

- Entities involved and functions in each entity:
    - There are three important functions send_packet , packet_generator ,receive which are executed parallelly by different threads in sender.cpp.
    - Send_packet will send packets to the receiver based on the protocol ,packet_generator will generate packets based on packet_gen_rate and the receiving thread will continuously take acknowledgements from the receiver.
    - In the receiver there is a receive function it receives data packet from the sender added into the out of order buffer and also recv buffer , the recv buffer will pop_out the sequence num and send ack to sender.

Results and Observations:

I have conducted experiments on the hyperparameters:

Max_packets = 400, Window size = 50 , seq_num_field = 8 ,buffer_size = 100 , max_packet_length = 512 , the units of Avg RTT are microseconds.

Packet generation rate = 20 :

Avg RTT values:

Packet_len \ drop_probab	1e-3	1e-5	1e-7
256 bytes	115.49	72.34	175.537
1500 bytes	77.245	123.023	147.315

Retransmission ratio

Packet_len \ drop_probab	1e-3	1e-5	1e-7
256 bytes	1.0025	1	1
1500 bytes	1.0025	1	1

Packet generation rate = 300:

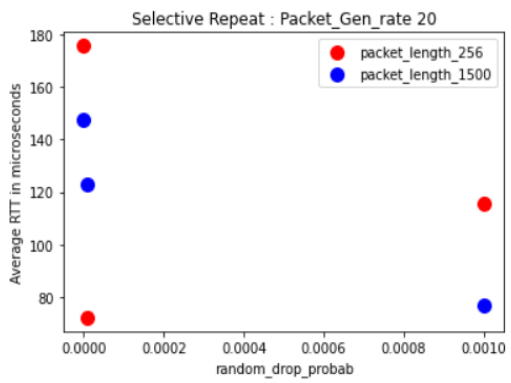
Avg RTT values:

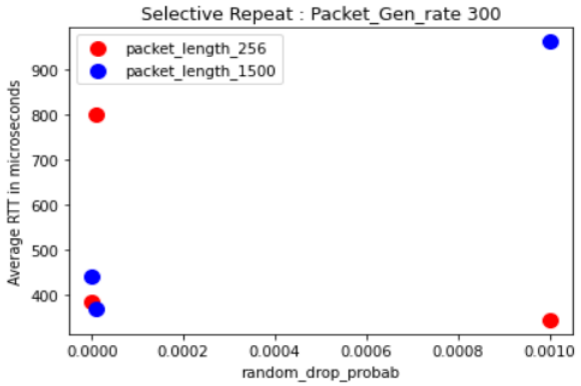
Packet_len \ drop_probab	1e-3	1e-5	1e-7
256 bytes	343.005	798.112	383.33
1500 bytes	960.77	369.565	439.055

Retransmission ratio

Packet_len \ drop_probab	1e-3	1e-5	1e-7
256 bytes	1.0025	1	1
1500 bytes	1.0025	1	1

Plots:





Observations:

Avg RTT value is higher if the packet length is higher as expected , all retransmission ratios are closer to 1 as the random probab is very low , the retransmission ratio is decreasing with the decrease in probability as expected , average RTT is less for small packet_generation rate. Average RTT is increasing with decrease in random_probab for smaller packet_generation rate , and for larger packet_generation rate it is varying randomly.

2. Go Back N Protocol :

Aim : To implement the Go-Back-N reliable transmission protocol and measure the round trip delays.

Introduction :

Go back N protocol is a sliding window protocol , here we use frame pipelining so as to send multiple frames without waiting for acks by which we achieve better link utilization,throughput so as to use bandwidth effectively.Here cumulative acknowledgements are used , receiver receives in-order frame only and cannot accept(discards) frames out of sequence. Sender must resend the entire window in the event of a lost frame.In GBN Sender window size is N and Receiver window size is always 1.

Experimental details:

- Experimental/Simulation setup :
  - Here we are running two separate programs at same time on two different hosts : a sender program that generates and transmits packets and receiver that accepts packets and transmits the acknowledgements to the sender.
  - The receiver randomly decides whether to drop a packet or not with a random_drop_probab.
  - And finally we measure the round trip average time and retransmission ratio , For sending of packets we follow the Go Back N protocol
  - Communication between the sender and receiver is through UDP sockets
  - The receiver is set up as a UDP server, and the sender is set up as a UDP client
- Entities involved and functions in each entity:
  - There are three important functions send_packet , packet_generator ,receive which are executed parallelly by different threads in sender.cpp.
  - Send_packet will send packets to the receiver based on the protocol ,packet_generator will generate packets based on packet_gen_rate and the receiving thread will continuously take acknowledgements from the receiver.

- Receiver is a simple program without multiple threads , it just waits for a data packet and drops with a probability of random_drop_probab , and if it is the expected seq_num it takes, else it ignores

Results and Observations:

I have conducted experiments on the hyperparameters:

Max_packets = 400, Window size = 50 , seq_num_field = 8 ,buffer_size = 100 , max_packet_length = 512, the units of Avg RTT are microseconds.

Packet generation rate = 20 :

Retransmission ratio

Packet_len \ drop_probab	1e-3	1e-5	1e-7
256 bytes	1.125	1	1
1500 bytes	1.125	1	1

Avg RTT values:

Packet_len \ drop_probab	1e-3	1e-5	1e-7
256 bytes	114.615	69.74	201.09
1500 bytes	417.408	348.255	90.732

Packet generation rate = 300:

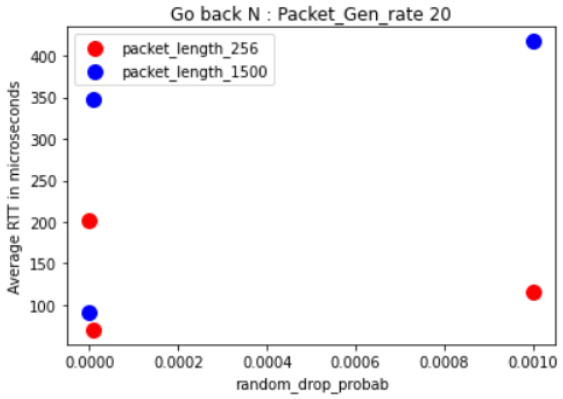
Retransmission ratio

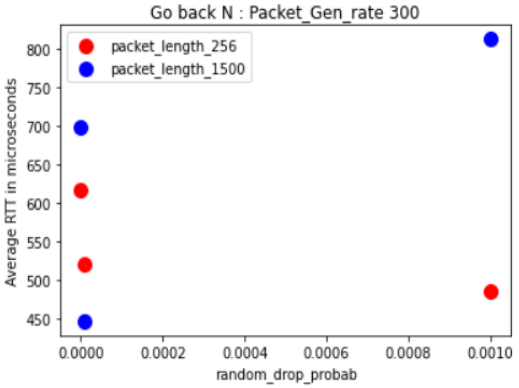
Packet_len \ drop_probab	1e-3	1e-5	1e-7
256 bytes	1.125	1	1
1500 bytes	1.125	1	1

Avg RTT values:

Packet_len \ drop_probab	1e-3	1e-5	1e-7
256 bytes	484.64	520.24	617.482
1500 bytes	813.238	446.03	698.48

Plots :





Observations:

Avg RTT value is higher if the packet length is higher as expected ,retransmission ratio for 1e-3 is higher than for the Selective Repeat it is expected to be high because we are sending all the packets in the frame if there is an error , and the avg RTT are nearing 1 as the random probab is very low , the retransmission ratio is decreasing with the decrease in probability as expected,average RTT is low for small packet_generation rate, Average RTT is behaving randomly wrt to random probab for both the packet_generation rate , in general it seems increasing but we can see few exceptions from the result.

Learnings :

1) Selective Repeat is efficient in terms of the number of retransmissions , we can observe from the results that retransmission ratio is quite low in case of Selective repeat than the Go back N.
2) Because of frame pipelining (so as to send multiple frames without waiting for ACK) we are achieving better link utilization.

3) Selective repeat is the best sliding window protocol with some sorting complexity at the receiver end else gobackN is best it is simple and we send multiple frames but defect is we retransmit everything in the frame when there is error for any of the packet.