CS3205 Computer Networks Lab
Assigment 4: Go Back N and Selective Repeat Protocol

G Shyam Sundar CS18B015

May 11, 2021

1 Objective

To implement Selective Repeat and Go Back N reliable transmission protocols and measure the round trip delays to compare them at different conditions.

2 Introduction

Go Back N Protocol :

- Go-Back-N ARQ is a specific instance of the automatic repeat request (ARQ) protocol, in which the sending process continues to send a number of frames specified by a window size even without receiving an acknowledgement (ACK) packet from the receiver.
- It uses the principle of protocol pipelining in which the multiple frames can be sent before receiving the acknowledgment of the first frame. If we have five frames and the concept is Go-Back-3, which means that the three frames can be sent, i.e., frame no 1, frame no 2, frame no 3 can be sent before expecting the acknowledgment of frame no 1.
- In Go-Back-N ARQ, the frames are numbered sequentially as Go-Back-N ARQ sends the multiple frames at a time that requires the numbering approach to distinguish the frame from another frame, and these numbers are known as the sequential numbers.
- The number of frames that can be sent at a time totally depends on the size of the sender's window. So, we can say that 'N' is the number of frames that can be sent at a time before receiving the acknowledgment from the receiver.
- If the acknowledgment of a frame is not received within an agreed-upon time period, then all the frames available in the current window will be retransmitted. Suppose we have sent the frame no 5, but we didn't receive the acknowledgment of frame no 5, and the current window is holding three frames, then these three frames will be retransmitted.
- The sequence number of the outbound frames depends upon the size of the sender's window.

Selective Repeat Protocol :

- Selective repeat protocol, also called Selective Repeat ARQ (Automatic Repeat reQuest), is a data link layer protocol that uses sliding window method for reliable delivery of data frames. Here, only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.
- It provides for sending multiple frames depending upon the availability of frames in the sending window, even if it does not receive acknowledgement for any frame in the interim.

- If the receiver receives a corrupt frame, it does not directly discard it. It sends a negative acknowledgment to the sender. The sender sends that frame again as soon as on the receiving negative acknowledgment. There is no waiting for any time-out to send that frame.
- It is mainly used because we need the receiver to be able to accept packets out-of-order using buffer space, for a superior protocol to combine advantages of both Stop-Wait and GBN.Selective Repeat attempts to retransmit only those packets that are actually lost (due to errors).
- The maximum number of frames that can be sent depends upon the size of the sending window.
- Induvidual acknowledgements are used in Selective Repeat Protocol

When Buffer Space is of more concern than bandwidth then Go Back N Protocol is used,if Bandwidth is of more concern than buffer space then Selective Repeat Protocol is used.In Go-Back-N we have Less complexity,less CPU cycles while for Selective Repeat Protocol More processing power,cpu cycles at receiver.If error rate is low, use Go-back-N else if error rate is high use Selective Repeat Protocol .

3 Experimental Details

3.1 Simulation setup

Selective Repeat Protocol:

This project requires two separate programs, running at the same time, on two different hosts: a sender program that generates and transmits packets; and a receiver, that accepts the packets, and transmits the acknowledgments to the sender. Note that the receiver does not send any data packet; it only sends acknowledgments. Communication between the sender and receiver is through UDP sockets.

Go Back N Protocol:

Setup is same as of 0 Seletive Repeat prorocol But Cumulative ACKs will be used.

3.2 Entities involved and functions in each entity

Selective Repeat Protocol:

- **Sender:**
  The main loop of sender has these main steps:
  a. Generate a packet of length, where the packet length follows a uniform distribution: Uniform(40, MAX_PACKET_LENGTH) bytes, where MAX_PACKET_LENGTH is commandline param eter). The first byte(s) of the packet contains the sequence number (depends on the number of bits in the sequence number field).
  b. Packets are generated at periodic time intervals specified by the PACKET_GEN_RATE parameter (packets/unit time). The transmit buffer has a capacity specified by the BUFFER_SIZE parameter (number of packets, not bytes). A newly generated packet will be dropped if the Buffer is full. A sequence number is assigned ONLY if the packet is added to the buffer.
  c. Transmit the packet based on the Window conditions. Start the timeout timer for this packet's sequence number. The timeout is set to 300 ms for the first 10 packets and then $2 \times RTT_{ave}$ (in milliseconds) for all other packets.
  d. Process the next packet (when available) and transmit it if the sender window is not exhausted, i.e. the total number of unacknowledged packets is at most WINDOW_SIZE. Given an nbit sequence number, the maximum window size will be 2^{n-1} for Selective Repeat.

e. If an ACK packet arrives, process it, update local state variables and cancel timers corresponding to acknowl edged packets. Remove the packet from the Transmit Buffer. Note that selective ACKs are used. For each sequence number acknowledged, calculate the RoundtripTime (RTT) for the packet and update the average RTT (RTT_{ave}) for the packets acknowledged so far.

f. If a timer expires, retransmit only the unacknowledged packet.

g. The sender terminates after MAX_PACKETS (a commandline parameter) have been successfully ACKNOWLEDGED (OR) if the maximum retransmission attempts for any sequence number exceeds 10.

- **Receiver:**
The receiver is always waiting to read a packet from the UDP socket it is listening to. Whenever a packet is delivered to the receiver:

a. The receiver randomly decides that packet is corrupted and decides to drop the packet; note that you can use rand, rand48, etc. The probability of packet drop is specified as a commandline parameter, denoted by PACKET_ERROR_RATE. This step is used to simulate random network errors.

b. If the packet is NOT corrupted (per step 1 above), the receiver reads the packet and extracts the sequence number. If the receiver buffer is FULL, then the received packets are discarded even if they were correctly received. Otherwise, it follows the Selective Repeat protocol for generating ACKs, and buffering outoforder packets.

c. The ACK packets are NOT dropped and are always assumed to be delivered to the sender.

d. The receiver terminates after acknowledging MAX_PACKETS (a commandline parameter).

Go Back N Protocol:

- **Sender:**
The main loop of sender has these main steps:

a. Generate a packet of length PACKET_LENGTH(command-line parameters) bytes. The first bytes of the packet contain the sequence number.Packet generation rate is given by the commandline parameter PACKET_GEN_RATE, in packets per second.

b. You might need to use a thread that generates packets periodically (based on the above rate) and stores them in a buffer used by the sender's protocol. The maximum size of this sender transmission buffer is given by the commandline parameter, MAX_BUFFER_SIZE (number of packets, not bytes). A newly generated packet will be dropped if the Buffer is full. A sequence number is assigned ONLY if the packet is added to the buffer.

c. Transmit the packet based on the Window conditions. Start the timeout timer for this packet's sequence number. The timeout is set to 100 ms for the first 10 packets and then $2 \times RTT_{ave}$ (in milliseconds) for all other packets.

d. Process the next packet (when available) and transmit it if the sender window is not exhausted, i.e. the total number of unacknowledged packets is at most WINDOW_SIZE.

e. If an ACK packet arrives, process it, update local state variables and cancel timers corresponding to acknowledged packets. Note that cumulative ACKs are used. For each packet received, calculate the Round-tri-pTime (RTT) for the packet and update the average RTT (RTT_{ave}) for the packets acknowledged so far.

f. If a timer expires, retransmit only the unacknowledged packet.

g. The sender terminates after MAX_PACKETS (a commandline parameter) have been successfully ACKNOWLEDGED (OR) if the maximum retransmission attempts for any sequence number exceeds 5.

- **Receiver:**
  The receiver is always waiting to read a packet from the UDP socket it is listening to. Whenever a packet is delivered to the receiver:

  a. The receiver randomly decides that packet is corrupted and decides to drop the packet; note that you can use rand, rand48, etc. The probability of packet drop is specified as a command-line parameter, denoted RANDOM_DROP_PROB. This step is used to simulate random network errors.

  b. If the packet is NOT corrupted (per step 1 above), the receiver reads the packet and extracts the sequence number. If sequence number matches the NEXT EXPECTED sequence number, it transmits an ACK to the sender, and updates local state variables. Note that Cumulative ACKs are used by the receiver.

  c. The ACK packets are NOT dropped and are always assumed to be delivered to the sender.

  d. The receiver terminates after acknowledging MAX_PACKETS (a commandline parameter).

3.3 Additional details

- Negative ACK's are excluded from Selective Repeat Protocol.
- OutPut for both Protocols In Non-Debug mode for Sender:
  - ReTransmission Ratio: Ratio of Total Number of Transmissions (including Retransmissions) to Number of Packets Acknowledged.
  - Average RTT Value for ALL Acknowledged Packets
- OutPut for both Protocols In Debug mode for Sender:
  - the Sender will print the following information for EACH packet when its ACK is received:
  - Seq : Time Generated: xx:yy RTT: zz Number of Attempts: aa
    where time is in milliseconds:microseconds format.
- OutPut for both Protocols In Debug mode for Receiver:
  - Receiver will print the following information for EACH packet when code stops executing. Note that the receiver will print this information ONLY in Sequence Number order.
  - Seq : Time Received: xx:yy
    where time is in milliseconds:microseconds format.

4 Results and Observations

Selective Repeat Protocol

Error_Rate	Average_RTT	Retransmission Ratio
0.2	3.90862	1.2325
0.1	2.72504	1.095
0.01	0.05496	1.0475

Go Back N Protocol

Error_Rate	Average_RTT	Retransmission Ratio
0.2	443.07	1.84
0.1	0.07898	1.1325
0.01	0.05745	1.0375

These are the outputs tested for Error_rates given in the table with Max_packets = 400 .

- In Both the protocols Average RTT increases with error rate But when we compare the increase in RTT for Error_rates 0.1 and 0.2. RTT for Go Back N protocol increses from nearly 0 to 443 due to high number of retransimissions.

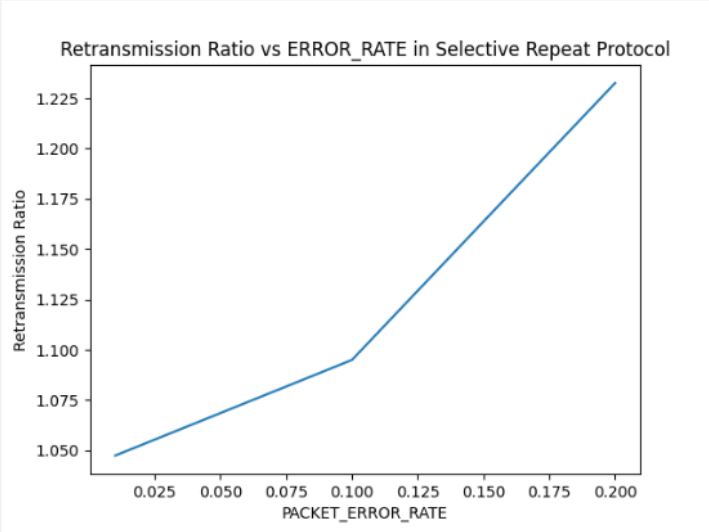
- So if the Packet_Error_Rate is so high then using Selective Repeat will give us good RTT.

- And For low Packet_Error_Rates we can observe that the Average RTT for Go Back N protocol is slightly good than the Selective Repeat Protocol. Which means when Error rates are low where retransmissions are less we may use Go Back N protocol so that we get Low RTT(fast transmissions).

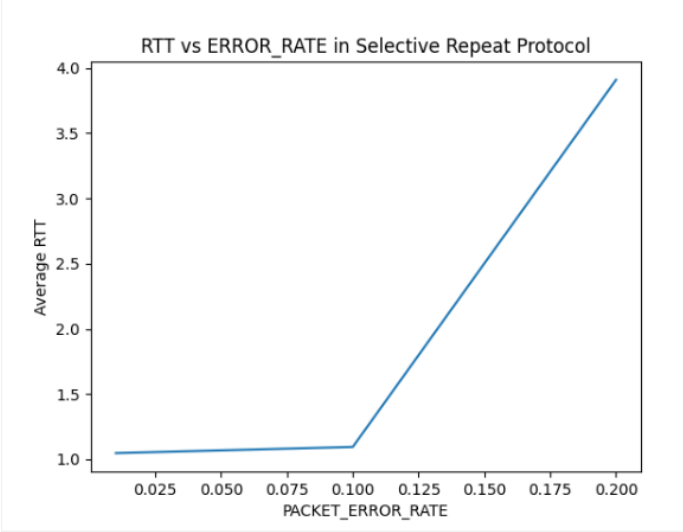
- Same as Average RTT, Retransmission ratio also increases with error rate but here the increase in the ratio is almost same in both protocols.

- And for any Error_rate Retransmission ratio is always less in Selective Repeat Protocol as we only retransmit the required packet.

- So If we want less traffic(less transmissions) in the network,we should use Selective repeat Protocol.

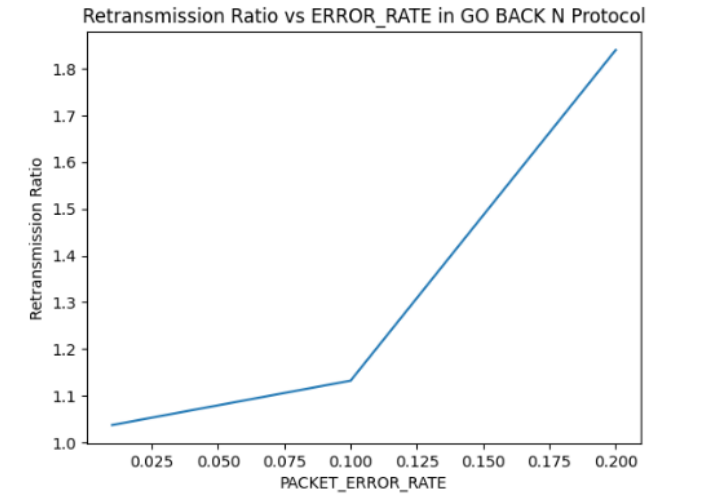


(a)

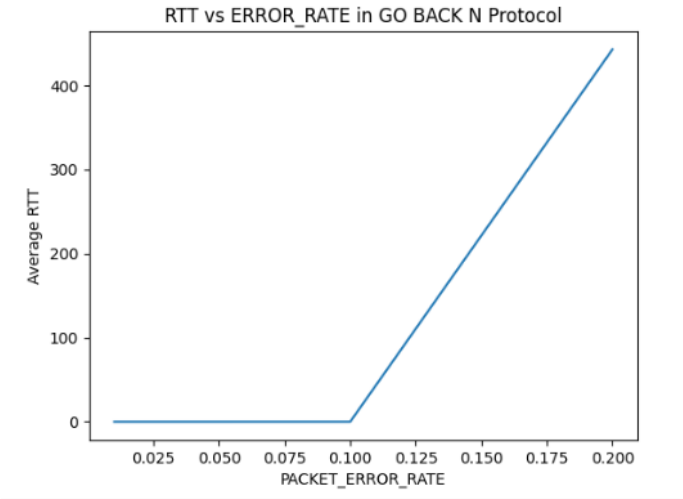


(b)

Figure 1: Selective Repeat Protocol



(a)



(b)

Figure 2: GO Back N Protocol

5 Learnings

- Learned the working Go Back N and Selective Repeat Protocols in detail.
- Learned Which protocol to use given the circumstances.
- Learned some Technical concepts like Using Locks and protecting the shared varibles.

6 Conclusion

- If the Packet_Error_Rate is so high then using Selective Repeat will give us good RTT.
- Else Using Go Back N gives us good RTT(fast transmission.)
- We may get a good RTT with Go Back N in case of low Error Rates but the retransmissions are more in case of Go Back N.
- So If we want less traffic(less transmissions) in the network,we should use Selective repeat Protocol.

7 References

- https://www.javatpoint.com/sliding-window-protocol
- https://www.tutorialspoint.com/difference-between-go-back-n-and-selective-repeat-protocol
- https://www.geeksforgeeks.org/sliding-window-protocol-set-3-selective-repeat/
- https://www.geeksforgeeks.org/udp-server-client-implementation-c/