COMPUTER NETWORKS LAB REPORT

Name:	Abhishek De
Roll no:	001710501027
Class:	BCSE-III
Section:	A1
Assignment:	2

ASSIGNMENT 2

Problem statement:

Sender, Receiver and Channel all are independent processes. There may be multiple Transmitter and Receiver processes, but only one Channel process. The channel process introduces random delay and/or bit error while transferring frames. Define your own frame format or you may use IEEE 802.3 Ethernet frame format.

Hints: Some points you may consider in your design.

Following functions may be required in Sender.

Send: This function, invoked every time slot at the sender, decides if the sender should (1) do nothing, (2) retransmit the previous data frame due to a timeout, or (3) send a new data frame. Also, you have to consider current network time measure in time slots.

Recv_Ack: This function is invoked whenever an ACK packet is received. Need to consider network time when the ACK was received, ack_num and timestamp are the sender's sequence number and timestamp that were echoed in the ACK. This function must call the timeout function.

Timeout: This function should be called by ACK method to compute the most recent data packet's round-trip time and then recompute the value of timeout.

Following functions may be required in Receiver.

Recv: This function at the receiver is invoked upon receiving a data rame from the sender.

Send_Ack: This function is required to build the ACK and transmit.

Sliding window:

The sliding window protocols (Go-Back-N and Selective Repeat) extend the stop-and-wait protocol by
allowing the sender to have multiple frames outstanding (i.e., unacknowledged) at any given time. The
maximum number of unacknowledged frames at the sender cannot exceed its "window size". Upon receiving a frame, the receiver sends an ACK for the frame's sequence number. The receiver then buffers the received frames and delivers them in sequence number order to the application.

Performance metrics: Receiver Throughput (packets per time slot), RTT, bandwidth-delay product, utilization percentage.

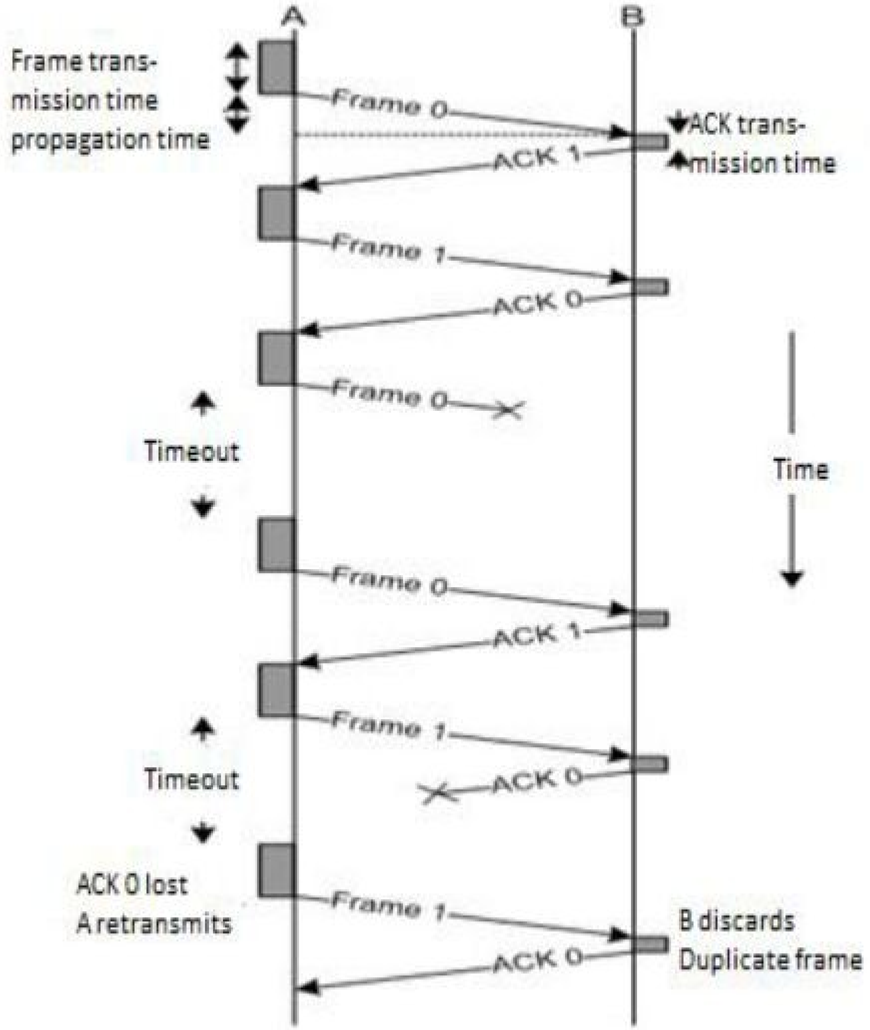
Design

Purpose of the program

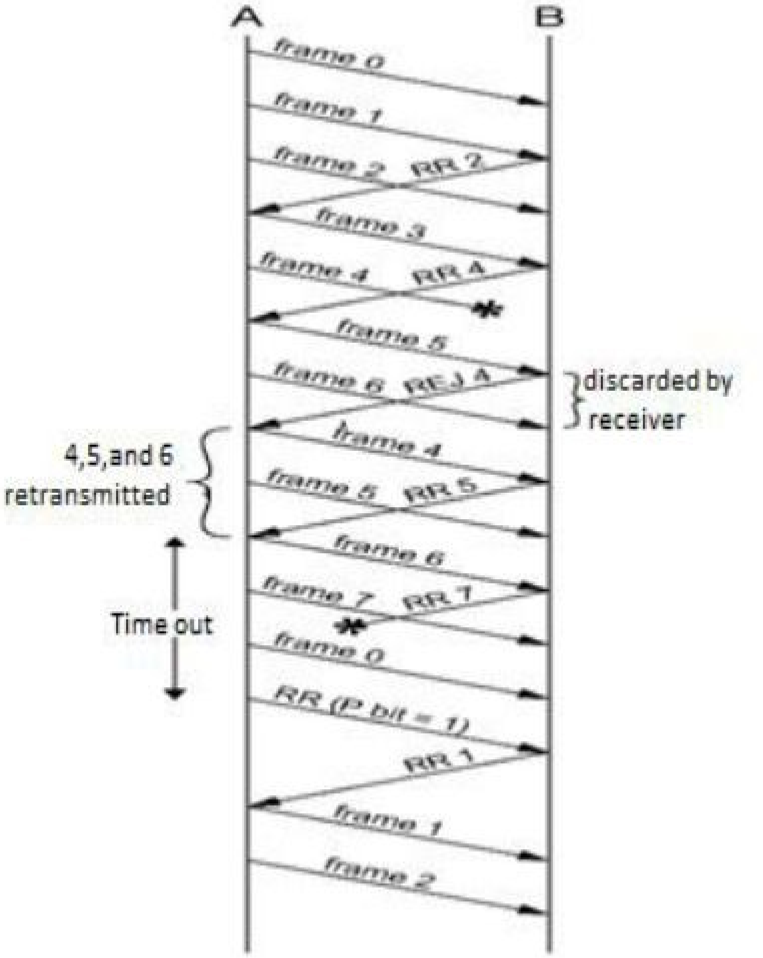
The program implements three flow methods namely stop and wait arq, go back n and selective repeat. The purpose of the program is to check that packets are reaching the destination correctly using each flow control method and calculate various parameters for each technique such as throughput and bandwidth delay product. To distinguish between two data frames a unique number is used for each data frames which is
known as sequence number. When the receiver receives a correct frame, it sends an acknowledgement to the sender indicating that it has received the frame. When the receiver sends an acknowledgement, it sends a number along with it to specify the next frame to be sent. The sender maintains a clock and has a specific timeout period. If an acknowledgement of a
data frames previously transmitted does not arrive before timeout, the sender retransmits the frame, thinking that either the frame or the acknowledgement has been lost in transit.

Stop and Wait ARQ:

Sender: Transmits a single frame at a time.
The algorithm is that sender waits to receive ACK within time out.
Receiver transmits acknowledgement (ACK) as it receives a frame.
We have to do the same thing when ACK is received or time out is hit.If a frame or ACK is lost during transmission then the frame is
re-transmitted. This re-transmission process is known as ARQ
(automatic repeat request).

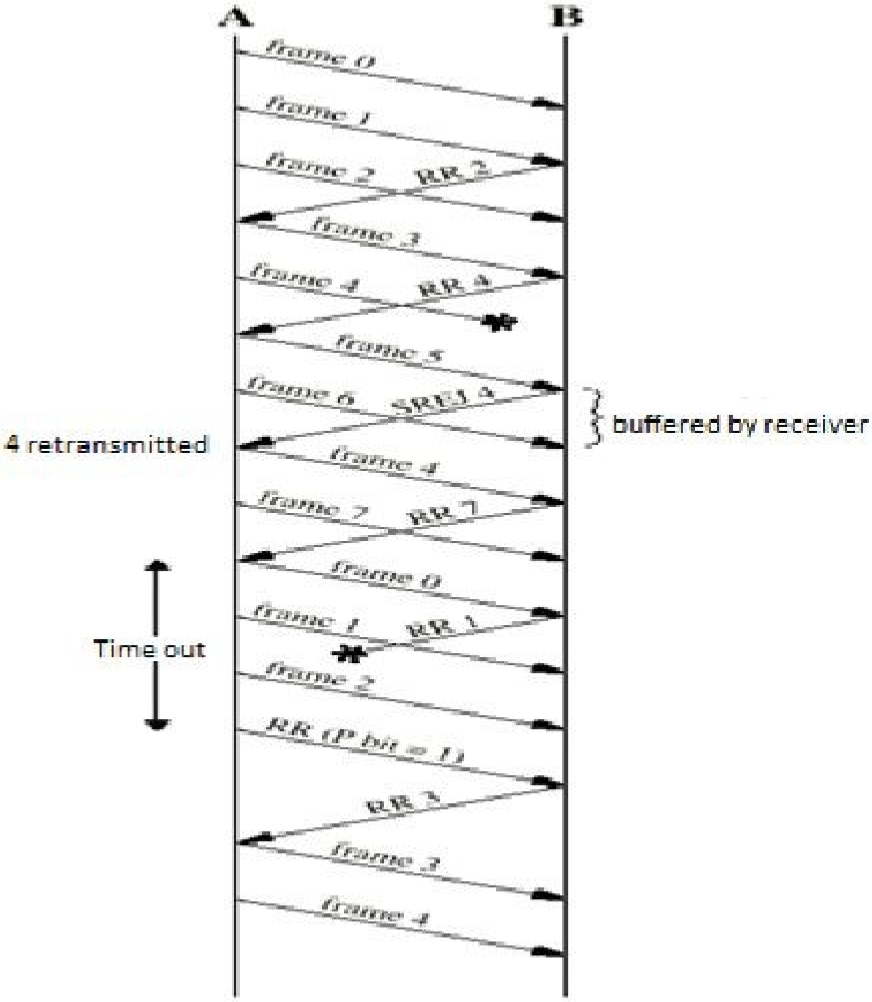


Go Back N Sliding Window Protocol:
The sender sends N packets which are equal to the window size. Once the entire window is sent,the sender then waits for a cumulative ACK to send more packets. On the receiver end, it receives only in-order packets and discards out-of-order packets. As in case of packet loss, the the entire window would be re-transmitted.



Sliding Window Based Selective Repeat Protocol:

The go-back-n protocol works well if errors are less, but if the line is poor it wastes a lot of bandwidth on retransmitted frames. An alternative strategy, the selective repeat protocol, is to allow the receiver to accept and buffer the frames following a damaged or lost one. Selective Repeat attempts to retransmit only those packets that are actually lost (due to errors). This protocol(SRP) is mostly identical to GBN protocol, except that buffers are used and the receiver, and the sender, each maintain a window of size. SRP works better when the link is very unreliable. Because in this case, retransmission tends to happen more frequently, selectively retransmitting frames is more efficient than retransmitting all of them. SRP also requires a full duplex link. backward acknowledgements are also in progress. Sender’s Windows (Ws) = Receiver’s Windows (Wr). Window size should be less than or equal to half the sequence number in SR protocol. This is to avoid packets being recognized incorrectly. If the windows size is greater than half the sequence number space, then if an ACK is lost, the sender may send new packets that the receiver believes are retransmissions. Sender can transmit new packets as long as their number is with W of all unACKed packets.Sender retransmit un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.Receiver ACKs all correct packets. Receiver stores correct packets until they can be delivered in order to the higher layer.In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.



Output:

```
Demonstrating Stop-and-wait protocol:

Enter the number of frames to read from file:5

Reading of frames from file completed

Sending 0's frame
Receiver gets = 1011001
0's ACK RECEIVED

Sending 1's frame
```

```
Receiver gets = 0101101
1's ERROR detected

Sending 1's frame
Receiver gets = 0100101
1's ERROR detected

Sending 1's frame
Receiver gets = 0100101
1's ERROR detected

Sending 1's frame
Receiver gets = 0100101
1's ACK RECEIVED

Sending 2's frame
Receiver gets = 1010100
2's ACK RECEIVED

Sending 3's frame
Receiver gets = 1010100
3's ACK SENDING FAILED

Sending 3's frame
Receiver gets = 1010100
3's ERROR detected

Sending 3's frame
Receiver gets = 1010100
3's ACK RECEIVED

Sending 4's frame
Receiver gets = 1001001
4's ERROR detected

Sending 4's frame
Receiver gets = 1001001
4's ERROR detected
```

```
Sending 4's frame
Receiver gets = 1001001
4's ACK SENDING FAILED

Sending 4's frame
Receiver gets = 1001001
4's ACK RECEIVED



Sent at = 1583985387 ACK Received at = 1583985390 RTT= 3
Sent at = 1583985402 ACK Received at = 1583985403 RTT= 1
Sent at = 1583985404 ACK Received at = 1583985406 RTT= 2
Sent at = 1583985415 ACK Received at = 1583985418 RTT= 3
Sent at = 1583985428 ACK Received at = 1583985430 RTT= 2
Value in the 0th frame = 1011001
Value in the 1th frame = 0100101
Value in the 2th frame = 1010100
Value in the 3th frame = 1010100
Value in the 4th frame = 1001001
Receiver Throughput = 0.125
Utilization percentage = 0.384615
Bw delay prod = 1.25


Demonstrating Go-Back-N protocal

Enter the window size  :8
Sending 0's frame
Sending 1's frame
Sending 2's frame
Sending 3's frame
Sending 4's frame
Receiver gets = 1011001
0's ACK RECEIVED

Receiver gets = 0100101
1's ACK RECEIVED
```

```
Receiver gets = 1010100
2's ACK RECEIVED

Receiver gets = 1000100
3's ERROR detected

Sending 3's frame
Sending 4's frame
Receiver gets = 1010100
3's ACK RECEIVED

Receiver gets = 1001001
4's ACK SENDING FAILED

Sending 4's frame
Receiver gets = 1001001
4's ACK RECEIVED

Value in the 0th frame = 1011001
Value in the 1th frame = 0100101
Value in the 2th frame = 1010100
Value in the 3th frame = 1010100
Value in the 4th frame = 1001001


Sent at = 1583985445 ACK Received at = 1583985446 RTT= 1
Sent at = 1583985445 ACK Received at = 1583985446 RTT= 1
Sent at = 1583985445 ACK Received at = 1583985446 RTT= 1
Sent at = 1583985447 ACK Received at = 1583985449 RTT= 2
Sent at = 1583985450 ACK Received at = 1583985453 RTT= 3
Bw delay prod = 4.44444
Receiver Throughput = 0.714286
Utilization percentage = 0.625


Demonstrating Selective Repeat ARQ:

Enter the window size  :8
Sending 0's frame
Sending 1's frame
Sending 2's frame
Sending 3's frame
Sending 4's frame
Receiver gets = 1011001
0's ACK RECEIVED

Receiver gets = 0100101
1's ACK SENDING FAILED

Sending 1's frame
Receiver gets = 1010100
2's ACK RECEIVED

Receiver gets = 1010100
3's ACK SENDING FAILED

Sending 3's frame
Receiver gets = 1001001
4's ACK SENDING FAILED

Sending 4's frame
Receiver gets = 0100101
1's ACK RECEIVED

Receiver gets = 1010100
3's ACK SENDING FAILED

Sending 3's frame
Receiver gets = 1001001
4's ACK RECEIVED

Receiver gets = 1010100
3's ACK RECEIVED

Value in the 0th frame = 1011001
Value in the 1th frame = 0100101
Value in the 2th frame = 1010100
Value in the 3th frame = 1010100
Value in the 4th frame = 1001001


Sent at = 1583985458 ACK Received at = 1583985459 RTT= 1
Sent at = 1583985460 ACK Received at = 1583985468 RTT= 8
Sent at = 1583985458 ACK Received at = 1583985463 RTT= 5
Sent at = 1583985469 ACK Received at = 1583985472 RTT= 3
Sent at = 1583985466 ACK Received at = 1583985472 RTT= 6
Bw delay prod = 7.66667
Receiver Throughput = 0.384615
Utilization percentage = 0.555556
```

Analysis:

We understand that the major difference in the three protocols is in the sender and receiver window lengths which progressively give benefits from one protocol to the next. For stop and wait, both are 1, in case of Go Back N, sender is N and receiver is 1, while in case of Selective Repeat both window lengths are N. This gives greater flexibility as one could not send frame x until Ack for frame x-1 was received in Stop and Wait but it is possible in Go Back N. Also, this has cumulative delivery which means that if frame x is delivered, all frames before x would have been delivered. In case of selective repeat, there is anadditional facility that the delivery can be out of order as well which helps in such a case when if a frame is lost, the frames after that which have already been successfully transmitted will not need to be transmitted again. These points are further illustrated by our results and the sequence in the given outputs.

Comments:

The lab assignment was a little difficult. My most significant learning from the assignment was realizing in practice how the flow control protocols work and analyzing them through the different performance metrics.