MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Mini Project Report

of

Computer Networks LAB

TITLE: FLOW CONTROL IN DLL

SUBMITTED

BY

Atul Sahu – 200905090 (Roll no. 20)

Vikyath Shetty B – 200905156 (Roll no. 31)

ABSTRACT

In data communications, flow control is the process of managing the rate of data transmission between two nodes to prevent a fast sender from overwhelming a slow receiver. It provides a mechanism for the receiver to control the transmission speed, so that the receiving node is not overwhelmed with data from transmitting node.

Following are the flow control techniques of data-link layer in wired networks:

Noiseless channel : Simplest protocol, Stop-and-Wait protocol

Noisy channel : Stop-and-Wait Automatic Repeat Request, Go-back-N Automatic Repeat Request, Selective Repeat Automatic Repeat Request.

Automatic-repeat-request (ARQ) protocols are widely used in modern data communications to guarantee reliable transmission over imperfect physical links. The behavior of an ARQ protocol largely depends on a number of network parameters and traditionally simulation is used for their performance analysis. However, simulation provides less accurate results and usually requires enormous amount of CPU time in order to attain reasonable estimates. In spite of this, we try to present an analysis of performance of the various flow control techniques of data link layer.

INTRODUCTION

1.1 General Introduction to the Topic

Flow Control Techniques of Data-Link Layer-

1. Simplest Protocol

In simplest protocol, there is no flow control and error control mechanism. It is a unidirectional protocol in which data frames travel in only one direction (from sender to receiver). Also, the receiver can immediately handle any received frame with a processing time that is small enough to be negligible. The protocol consists of two distinct procedures : a sender and receiver. The sender runs in the data link layer of the source machine and the receiver runs in the data link layer of the destination machine. No sequence number or acknowledgements are used here.

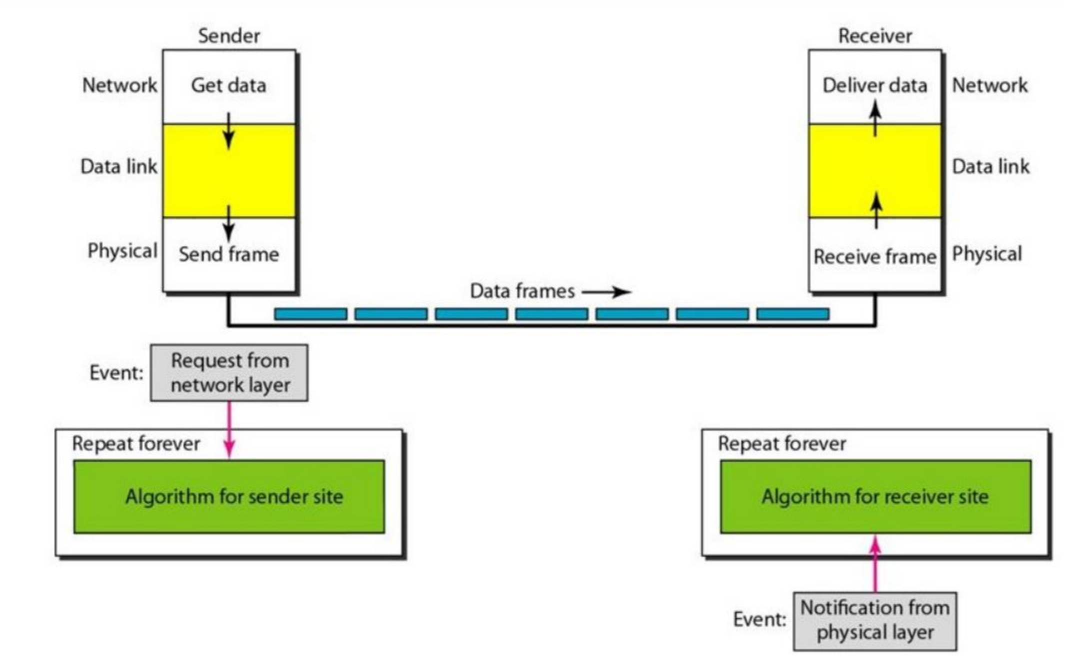


Fig A: Simplest Protocol

2. Simple Stop-and-Wait Protocol

In this method of flow control, the sender sends a single frame to receiver & waits for an acknowledgment. There is no chance of frame or acknowledgement as this protocol is used for noiseless channel.

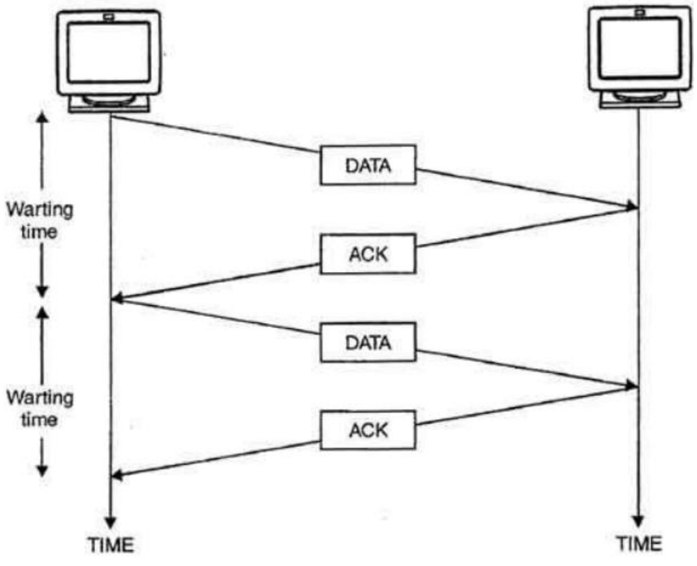


Fig B: Simple Stop-and-Wait Protocol

3. Stop-And-Wait ARQ Protocol

Stop-and-wait ARQ is a method to send information between two connected devices in a noisy channel. It ensures that information is not lost due to dropped packets and that packets are received in the correct order.

Characteristics:

It offers error and flow control

It is used in Data Link and Transport Layers

It uses link between sender and receiver as half duplex link

Throughput = 1 Data packet/frame per RTT

It is an special category of SWP where its window size is 1. Irrespective of number of packets sender is having stop and wait protocol requires only 2 sequence numbers 0 and 1

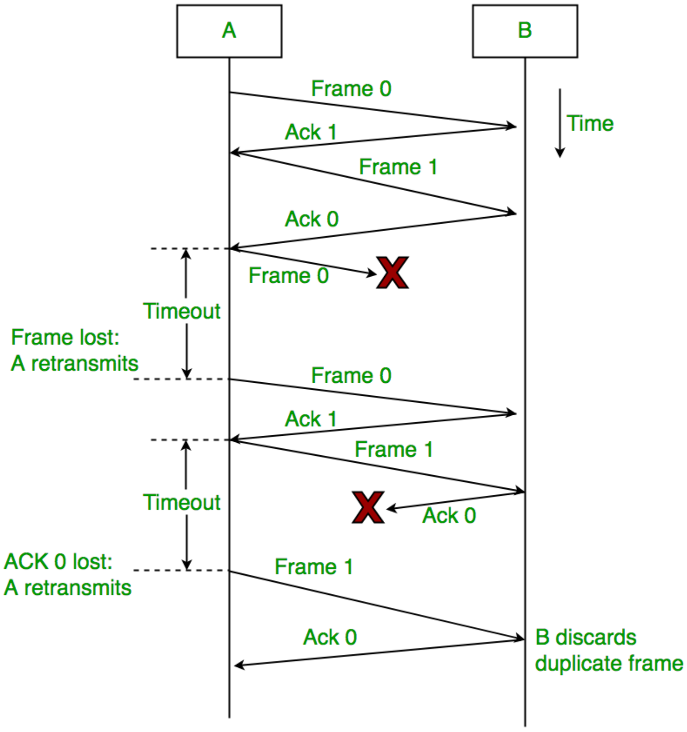


Fig C: Stop-And-Wait ARQ Protocol

4. Go-Back-N ARQ Protocol

Go-Back-N ARQ is a more efficient use of a connection than Stop-and-wait ARQ, since unlike waiting for an acknowledgement for each packet, the connection is still being utilized as packets are being sent.

Characteristics:

The key to Go-back-N is that several packets can be sent before receiving acknowledgments, but the receiver can only buffer one packet. Go back means sender has to go back N places from the last transmitted packet in the unacknowledged window and not from the point where the packet is lost. The sliding window size at sender's side is 2^(m-1) and receiver's side is 1.

GBN uses Cumulative Acknowledgement. At the receiver side, an acknowledgement timer is started whenever the receiver receives any packet which is fixed and when it expires, it sends a cumulative ACK for the number of packets recieved in that interval of timer i.e. if receiver has received N packets, then the acknowledgement number will be N+1.

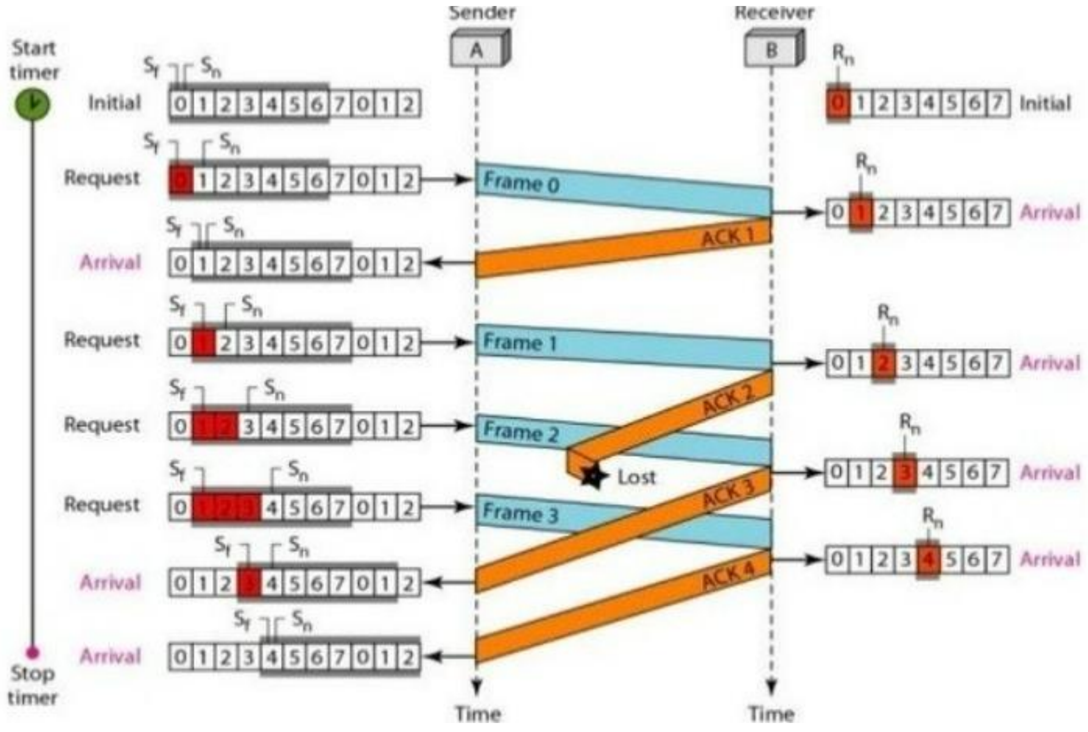


Fig D: Go-Back-N ARQ Protocol

5. Selective Repeat Request ARQ

Why 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) :Receiver must be able to accept packets out of order. Since receiver must release packets to higher layer in order, the receiver must be able to buffer some packets.

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 full duplex link. backward acknowledgements are also in progress.

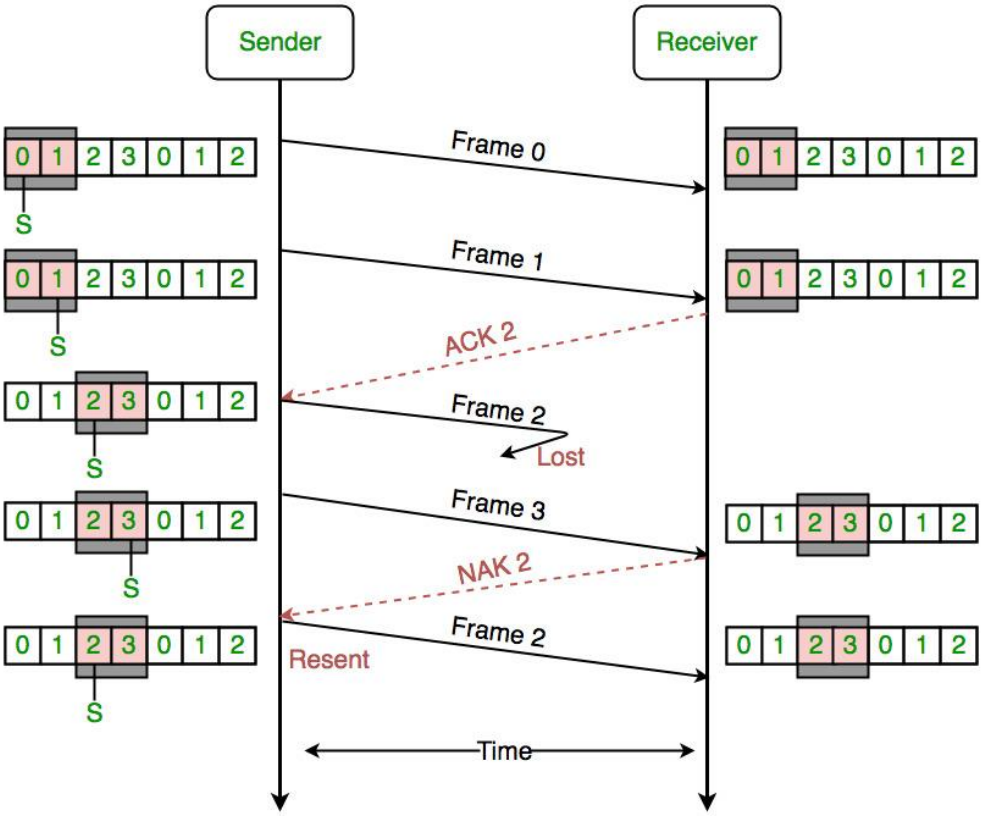


Fig E: Selective repeat ARQ

1.2 Hardware and Software Requirements

This project requires working pc with MATLAB software.

2: PROBLEM DEFINITION

Design of algorithms of different flow control techniques in data-link layer in wired networks.

3: OBJECTIVES

1. To simulate the above mentioned protocols in Matlab.
2. Performance analysis based on parameters such as efficiency, throughput and time delay will be presented. This will be done by using plotting graphs between efficiency and protocols and time delay and protocols.
3. Through this mini project I aim to gain in depth knowledge about the inner workings of the data link layer for flow control.

4: METHODOLOGY

1. Stop-and-Wait ARQ

This protocol is very inefficient if the channel is thick (large bandwidth) and long (round-trip delay is long). The product of the two is called the bandwidth delay-product. TThe bandwidth delay product is a measure of the number of bits we can send out of our system while waiting for news from the receiver. The protocol can be inefficient if channel is not used effectively.

The number of transmission in Stop-and-Wait ARQ is less than the other protocols. This is because the sender waits for an acknowledgement and retransmits only if the acknowledgement is not received or if it is lost.

Apart from this, pipelining is not implemented in this protocol. We need to wait for a frame to reach the destination and be acknowledged before the next frame can be sent. This further decreases the efficiency.

2. Go-Back-N ARQ

In this protocol, to improve the efficiency of transmission, multiple frames are in transition while waiting for acknowledgement. More than one frame are kept outstanding to keep the channel busy while the sender is waiting for acknowledgement.

However, this protocol involves a lot of unnecessary retransmissions. This is because all the frames in send window are transmitted when the timer expires even though some of the frames might have successfully reached the receiver.

Go-Back-N is also very inefficient for a noisy link. In a noisy link, a frame has a higher probability of damage, which means the resending of multiple frames. The resending uses up the bandwidth and slows down the transmission.

3. Selective Repeat ARQ

This protocol resends only the damaged frames and thus increases the efficiency in noisy links. However, the processing at the receiver is more complex. It is the most complex of the set of protocols for error control, it is however the most efficient scheme.

Resending only particular damaged frames drastically reduces the number of unnecessary transmissions, thus improving the efficiency.

Note: Graph of ratio of ideal to total number of transmissions

number of frames = 20 size of sequence number in bits = 4

5: IMPLEMENTATION DETAILS

1. SIMPLEST PROTOCOL

Sender: The data link layer at the sender site gets data from its network layer, makes a frame out of the data, and sends it.

Receiver: The data link layer at the receiver site receives a frame from its physical layer, extracts data from the frame, and delivers the data to its network layer.

2. SIMPLE STOP-AND-WAIT PROTOCOL

Sender: Send one data packet at a time.Send next packet only after receiving acknowledgement for previous.

Receiver: Send acknowledgement after receiving and consuming of data packet. After consuming packet acknowledgement need to be sent.

3. STOP-AND-WAIT ARQ PROTOCOL

Sender: Sends frame at a time. After sending each frame, the sender doesn't send any further frames until it receives an acknowledgement (ACK) signal. The timeout countdown is reset after each frame transmission.

Receiver: After receiving a valid frame, the receiver sends an ACK. If the ACK does not reach the sender before a certain time, known as the timeout, the sender sends the same frame again.

Design of the code:

In the matlab code, probabilities are assigned for Time-Out and ACK.

P(Time-Out) = 2/10

P(ACK) = 8/10

4. GO-BACK-N ARQ PROTOCOL

Sender: 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 is a special case of the general sliding window protocol with the transmit window size of N and receive window size of 1. It can transmit N frames to the peer before requiring an ACK.

Receiver: The receiver process keeps track of the sequence number of the next frame it expects to receive, and sends that number with every ACK it sends.

Design of the code:

In the matlab code, probabilities are assigned for Time-Out and ACK.

P(Time-Out) = 2/10

P(ACK) = 8/10

5. SELECTIVE REPEAT ARQ PROTOCOL

Sender: the sender sends a number of frames specified by a window size even without the need to wait for individual ACK from the receiver as in Go-Back-N ARQ. The sender individually retransmits frames that have timed out.

Receiver: The receiver may selectively reject a single frame, which may be retransmitted alone; this contrasts with other forms of ARQ, which must send every frame from that point again.

The receiver accepts out-of-order frames and buffers them.

Design of the code:

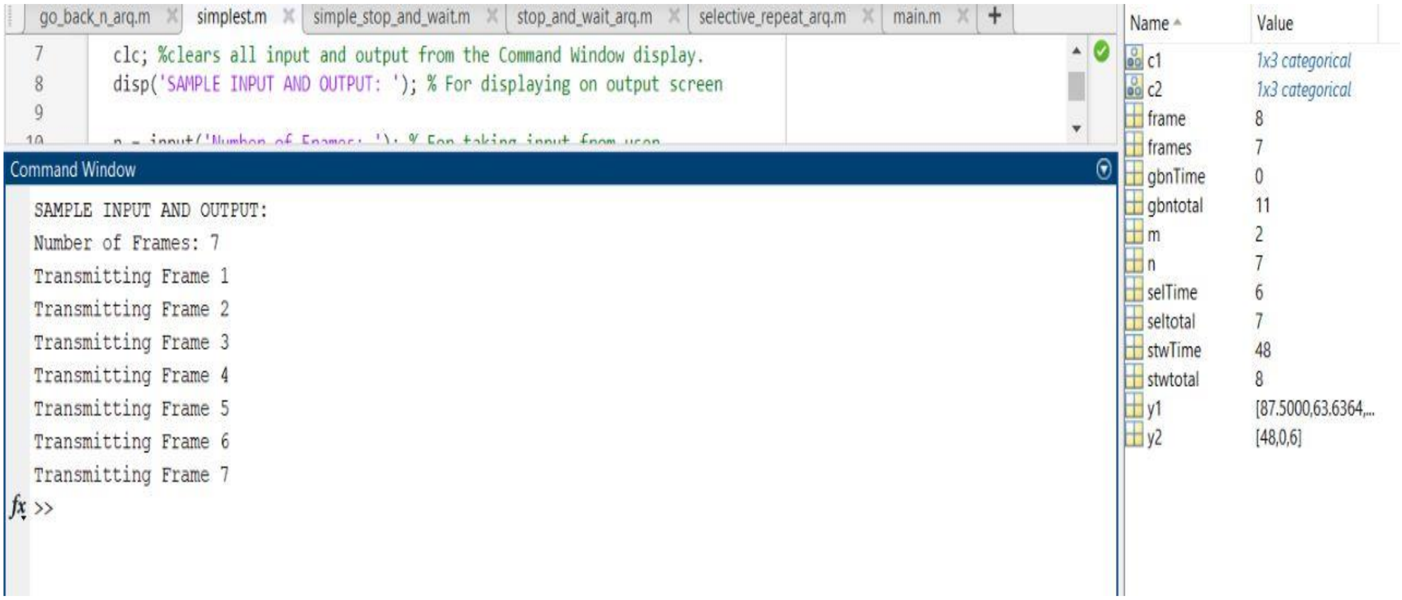
In the matlab code, probabilities are assigned for Positive ACK and Negative ACK.

P(NAK) = 2/10

P(PAK) = 8/10

OUTPUT

→ simplest.m (Fig 1)



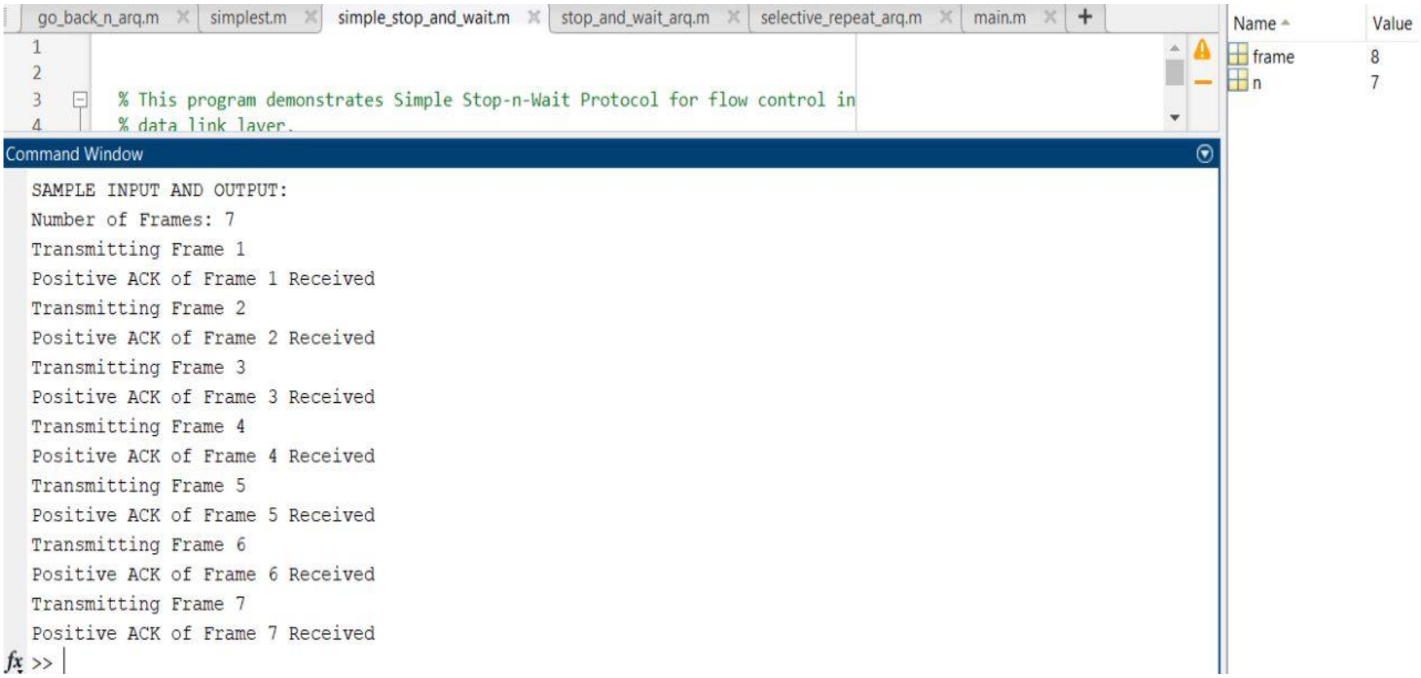
```
clc; %clears all input and output from the Command Window display.
disp('SAMPLE INPUT AND OUTPUT: '); % For displaying on output screen

n = input('Number of Frames: '); % For taking input from user
```

```
SAMPLE INPUT AND OUTPUT:
Number of Frames: 7
Transmitting Frame 1
Transmitting Frame 2
Transmitting Frame 3
Transmitting Frame 4
Transmitting Frame 5
Transmitting Frame 6
Transmitting Frame 7
>>
```

Name	Value
c1	1x3 categorical
c2	1x3 categorical
frame	8
frames	7
gbnTime	0
gbntotal	11
m	2
n	7
selTime	6
seltotal	7
stwTime	48
stwtotal	8
y1	[87.5000,63.6364,...
y2	[48,0,6]

Fig 1

→ simple_stop_and_wait.m (Fig 2)



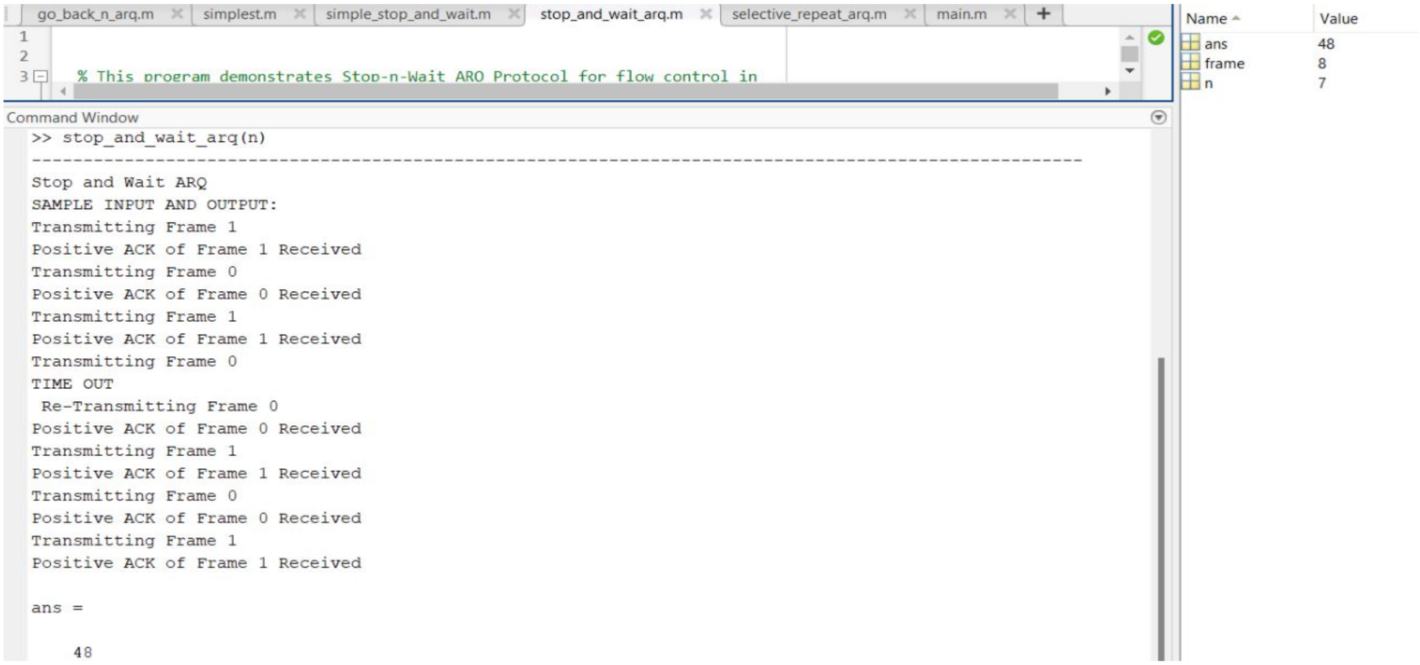
```
% This program demonstrates Simple Stop-n-Wait Protocol for flow control in
% data link layer.
```

Name	Value
frame	8
n	7

```
SAMPLE INPUT AND OUTPUT:
Number of Frames: 7
Transmitting Frame 1
Positive ACK of Frame 1 Received
Transmitting Frame 2
Positive ACK of Frame 2 Received
Transmitting Frame 3
Positive ACK of Frame 3 Received
Transmitting Frame 4
Positive ACK of Frame 4 Received
Transmitting Frame 5
Positive ACK of Frame 5 Received
Transmitting Frame 6
Positive ACK of Frame 6 Received
Transmitting Frame 7
Positive ACK of Frame 7 Received
fx >>
```

Fig 2

→ Stop_and_wait_arq.m (Fig 3)



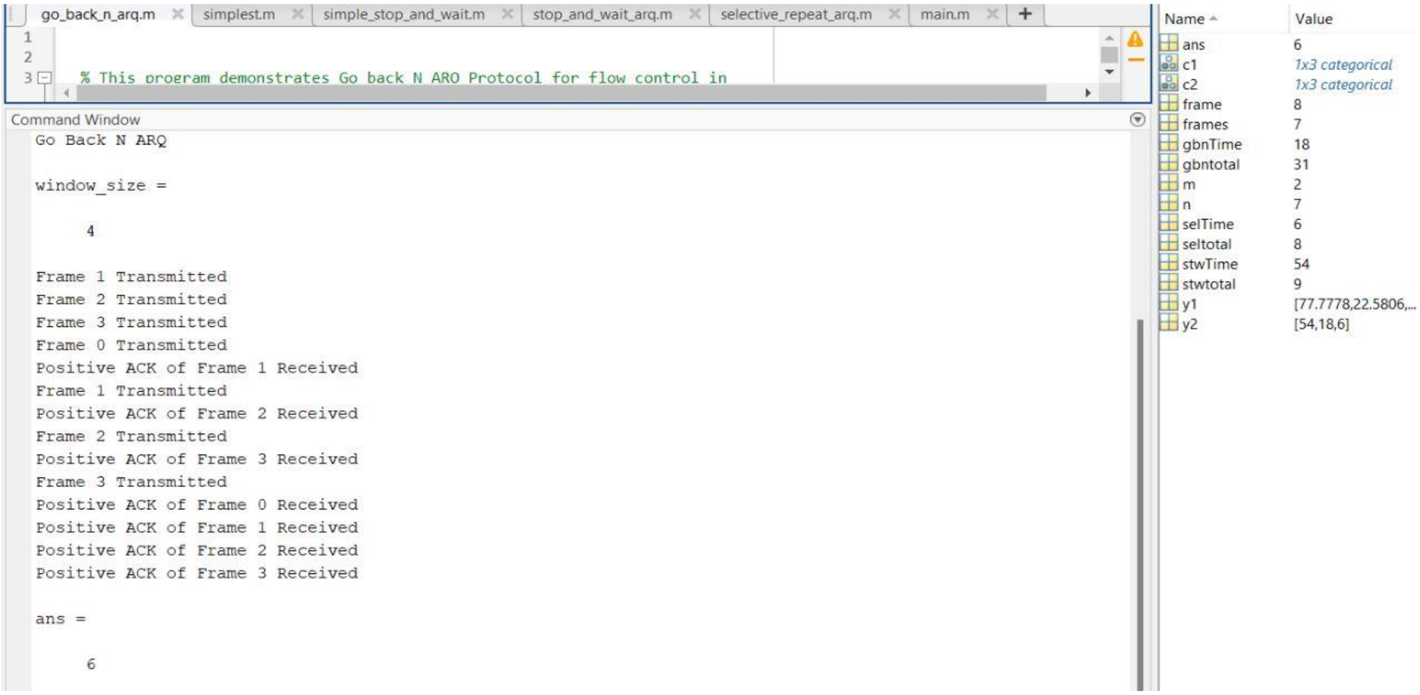
```
% This program demonstrates Stop-n-Wait ARQ Protocol for flow control in
```

Name	Value
ans	48
frame	8
n	7

```
>> stop_and_wait_arq(n)
-----------------------------------------------------------------------------------------------------
Stop and Wait ARQ
SAMPLE INPUT AND OUTPUT:
Transmitting Frame 1
Positive ACK of Frame 1 Received
Transmitting Frame 0
Positive ACK of Frame 0 Received
Transmitting Frame 1
Positive ACK of Frame 1 Received
Transmitting Frame 0
TIME OUT
 Re-Transmitting Frame 0
Positive ACK of Frame 0 Received
Transmitting Frame 1
Positive ACK of Frame 1 Received
Transmitting Frame 0
Positive ACK of Frame 0 Received
Transmitting Frame 1
Positive ACK of Frame 1 Received

ans =

    48
```

Fig 3

→ go_back_n_arq.m (Fig 4)



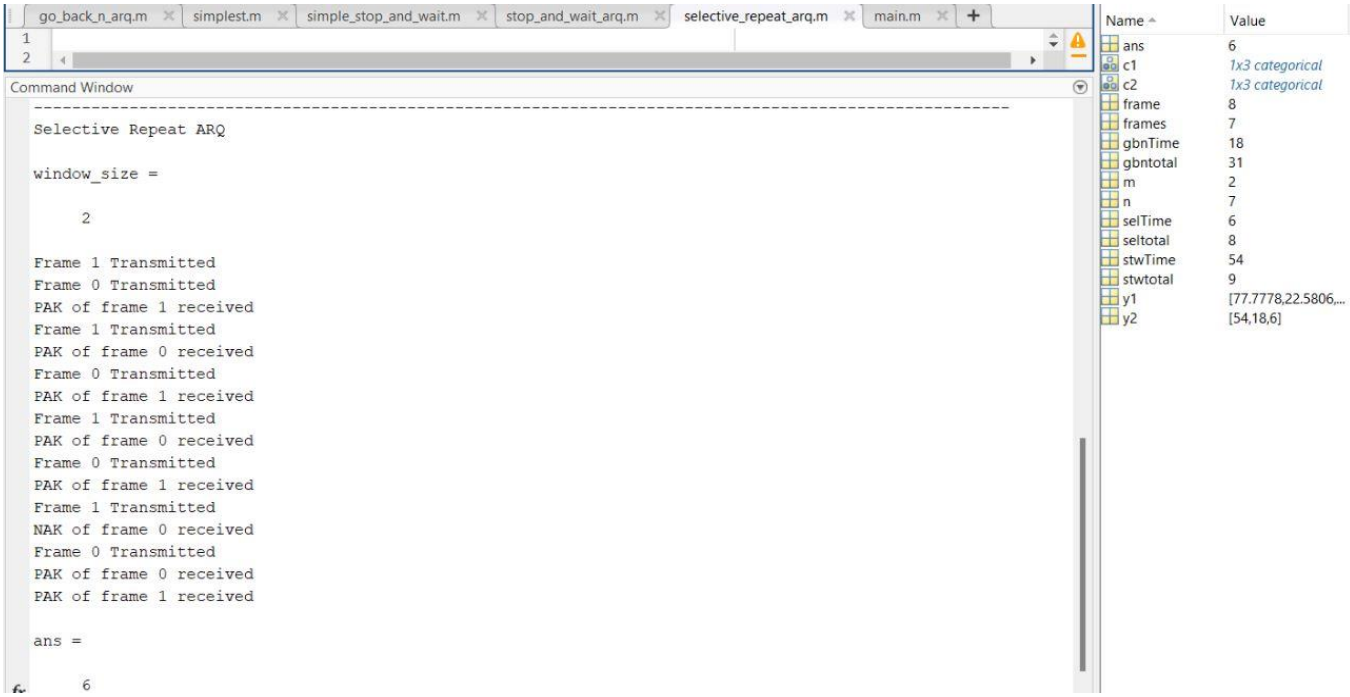
```
% This program demonstrates Go back N ARQ Protocol for flow control in
```

```
Go Back N ARQ

window_size =

     4

Frame 1 Transmitted
Frame 2 Transmitted
Frame 3 Transmitted
Frame 0 Transmitted
Positive ACK of Frame 1 Received
Frame 1 Transmitted
Positive ACK of Frame 2 Received
Frame 2 Transmitted
Positive ACK of Frame 3 Received
Frame 3 Transmitted
Positive ACK of Frame 0 Received
Positive ACK of Frame 1 Received
Positive ACK of Frame 2 Received
Positive ACK of Frame 3 Received

ans =

     6
```

Name	Value
ans	6
c1	1x3 categorical
c2	1x3 categorical
frame	8
frames	7
gbnTime	18
gbntotal	31
m	2
n	7
selTime	6
seltotal	8
stwTime	54
stwtotal	9
y1	[77.7778,22.5806,...
y2	[54,18,6]

Fig 4

→ Selective_repeat.m (Fig 5)



```
---------------------------------------------------------------------------------------------------
Selective Repeat ARQ

window_size =

     2

Frame 1 Transmitted
Frame 0 Transmitted
PAK of frame 1 received
Frame 1 Transmitted
PAK of frame 0 received
Frame 0 Transmitted
PAK of frame 1 received
Frame 1 Transmitted
PAK of frame 0 received
Frame 0 Transmitted
PAK of frame 1 received
Frame 1 Transmitted
NAK of frame 0 received
Frame 0 Transmitted
PAK of frame 0 received
PAK of frame 1 received

ans =

     6
```

Name	Value
ans	6
c1	1x3 categorical
c2	1x3 categorical
frame	8
frames	7
gbnTime	18
gbntotal	31
m	2
n	7
selTime	6
seltotal	8
stwTime	54
stwtotal	9
y1	[77.7778,22.5806,...
y2	[54,18,6]

Fig 5

➔ Main.m graph (Fig 6)

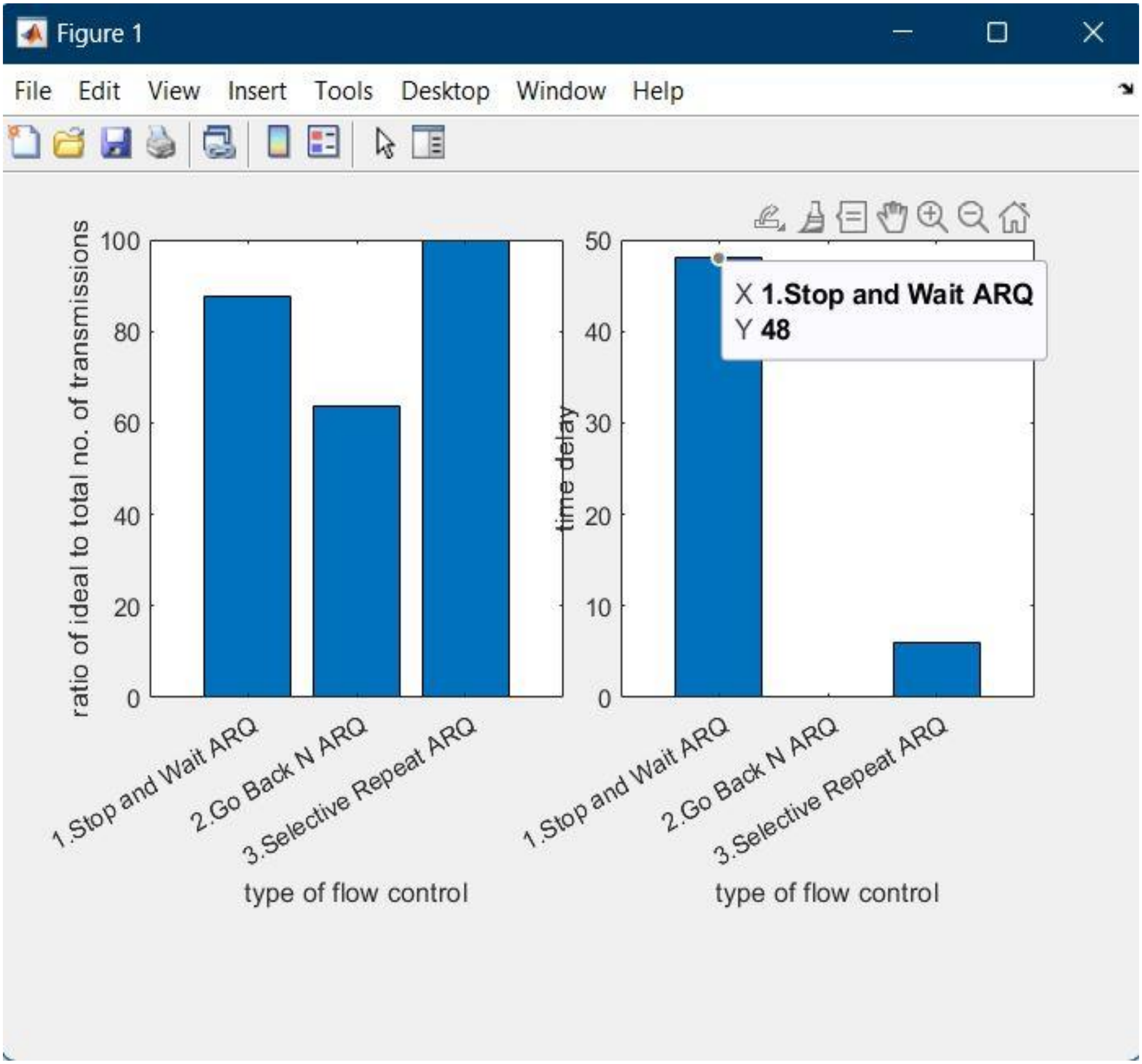


Fig 6

6: CONTRIBUTION DETAILS

Code related to Simplest, Simple Stop-and-wait - Vikyath

Code related to stop_and_wait_arq, go_back_n_arq - Atul

Code related to selective_repeat_arq, main program - Both

Design layout and Documentation – Both

7:REFERENCES

1. Computer Networking: A Top-down Approach (Book by Jim Kurose)

2. I. F. Akyildiz and W. Liu, "A general analysis technique for ARQ protocol performance in high speed networks," INFOCOM '91. Proceedings. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking in the 90s., IEEE, Bal Harbour, FL, 1991, pp. 498-507 vol.2.