INDEX

NAME : A·Kavin STD : III nd year SEC. : CSE-B ROLL NO. 220701122

S.No.	Date	Title	Page No.	Teacher's Sign/ Remarks
1.	16/7/24	Study of various Network commands used in Linux&windows		
2.	23/7/24	Study of Network cables		
3.	30/7/24	Experiments of CISCO PACKET TRACER(simulation tools		
4.	6/8/24	Setup and configure a LAN using a switch and Ethernet cable.		
5.	9/8/24	Experiments on packet capture tool; wireshark		
6.	16/8/24	Error correction at data link layer (Hamming code)		
7.	23/8/24	Flow control at data link layer (Sliding window protocol.		
8.	10/9/24	. Stimulate virtual LAN		
		cisco Packet Tracer		
9.	30/9/24	Implementation of subnetting in CISCO Packet tracer		
10.	4/10/24.	Internetworking using router DHCP server and internet cloud		
11.	8/10/24	. Stimulate static routing Protocol		
		configuration using CISO Packet & RIP		
12.	15/10/24	echo client TCP/UDP sockets chart client server TCP/UDP		
13	22/10/24.	write own Ping Problem		
14.	25/10/24	Raw sockets to implement		
		Packet sniffing		
15.	29/10/24	webilizer tool		

Completed

Palaniappa

Exp. NO : 07
Date : 23/08/2024

Practical-7

Aim

Write a program to implement flow control at data link layer using sliding window protocol. Simulate the flow of frames from one node to another.

create a sender program with following features.

1. Input window size from the user
2. Input a text message from the user
3. consider 1 character per frame
4. create a frame with following fields
5. Send the frames
6. Wait for the acknowledgement from the receiver
7. Reader a file called Receiver_buffer
8. check ack field for acknowledgement number
9. If the acknowledgement number is as expected, send new frames accordingly

create a receiver file with following features

1. Reader a file called Sender_Buffer
2. check the frame no.
3. If the frame no, as expected, write the appropriate ACK no. in the receiver-Buffer in file else write NACK no. in the Receiver-Buffer rule.

code:-

```
import math
import random
import time
def sender(message, window size):
    num-frames = math.ceil(len(message)/window_size)
    sent-frame =0
   while send-frame < num-frames :
      start = sent-frame * windows-size
      end = min(start + window_size len (message))
  Print (f"\n sender: sending frames from
     position (start+1) to {end}")
  for i in range (start, end):
     Print(f"Frame {i+1}: {message[i]} sent.")
 if random.choice([True, false]):
     Print (f" Receiver: Acknowledgement received
        for frames from{ start+ 1} to {end}")
        sent-frame +=1
   else:
      Print (f" Receiver: Acknowledgement NOT
   received for frames from { start+1} to {end}.
       Resending frames ...")
   Print (" All frames send successfully")
 def Receiver( message, window- size):
    pass
 if _name == "_main_":
    message = input ("Enter message to send")
    window_size = int (input ("Enter the window
                   size:"))
```

```
sender (message, window_size)
receiver (message, window_size)
```

Output:-

```
Enter the message to send : world
Enter the window size: 4
Sender: sending frames from position 1 to 4
Frame 1: 'w' sent.
Frame 2: 'o' sent
Frame 3: 'r' sent
Frame 4: 'l' send
Receiver: Acknowledgement not received for
frames 1 to 4. Resending frames
Sender: sending frames from position 1 to 4
Receiver: Acknowledgement not received for frame 5
                                              to 5
Sender: sending frameds from position 5 to 5
Frame: 5 'd' sent
Receivedr: Acknowledgement received from frame
 5 to 5.
```

23/8/24

Result

Thus the program is successfully executed and output is verified