SLIDING WINDOW PROTOCOL

EXPNO: 7

AIM. Write a program to implemen flow control at data link layer using sliding window protocol.

SENDER PROGRAM:

Input window size & text message from user.

Create frames with frame.no and send

Wait for ACK

Read Receiver-Buffer & check ACK field

If ACK is expected, send new frames.

else resend old frames

RECEIVER PROGRAM:

Read Sender-Buffer

Check frame no.

If frame is expected write ACK no to receiver-Butter else,

write rACK no. to Receiver-Butter

```
import time
Import random

class sliding WindowProtocol.
    self.window-size = window-size
    self.sender-window= window-size
    self. sequence- number=0
    self. acknowledged= set()

    def send-data(self. data_list):
        fordata in data_list:
            self. sender-window.append
        ({"data:data, 'sequence': self.sequence
        number})

        while self.sender-window:
         for packet in list(self.sender_windo
                if packet['sequence'] not in
         self.acknowledsed:
                print(f"sending data:{
packet['data']}(seq:{packet['Sequence']})")
```

```
self. sequence-number =
(self.sequence - number +1) %
(self. window - size * 2)

while self.sender_window:
    if packet['sequence'] not in
self acknowleged:
        printf(f"sending)
        time.sleep(1)

def transmit(self, packet):
    if random.random()>0.2:
    self.receiver_window.append
(packet)
  else:
      printf(f"Packet lost
during transmission");

list (self. receiver_window);
    if random.random()>0.1
        printf("Acknowledgment)
```

```
else:
    printf("Acknowledgment lost
for data: {packet['data']}
(seq: {packet['sequence']})")
    self.receiver_window.clear()
```

RESULT: 8/11

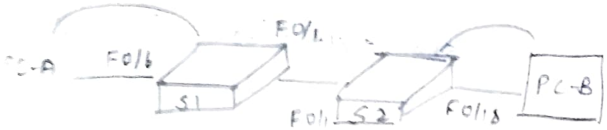
Thus the above program has been executed successfully.

CISCO PACKET TRACER

EXP NO: 8

AIM:

a) Simulate virtual LAN configuration using Cisco Packet trabet simulatin



PACKET TRANSFER

INSTRUCTIONS:

STEP1: BUILD THE NETWORK AS SHOWN IN TOPOLOGY:

Drag switches S1 & S2 to the Rack.

Drag PCs A and B to the table & turn them up.

Connect Straight - through cables as shown in topulog

Connect consules

STEP2: Contigure Basic settings for each switch

Console into each switch, enable privileged EXCE mode & contigure

Save the running contiguration to the startup contiguration file

STEP3: Contigure PC hosts

STEP4: Test connectivity

PART2: Create VLAN'S & Assign switch ports

STEP1:

a. S1 (config) # vlan10
 S1 (config-vlan) # name operatic.

b. Create the same VLANS on S2

c. Issue the show vlan brief

STEP2:

a) Assign VLANS to interface on S1

* Assign PC-A on VLAN
 From VLAN-1, remove the manageme..
 Assign PC-B to the operatin VLAN on S2

PART-3: Maintain VLAN Port Assignments & the VLAN Database

STEP1: Assign a VLAN to multiple interfaces

STEP2: Remove [illegible] assignment from

STEP 3: Remove a VLAN ID from VLA- database

PART-4 configure an 802.1Q Trunk between the switches

STEP1: Use DTP to initiate trunking on F0/1

STEP 2: Manually configure trunk interfa..

STUDENT OBSERVATIO

b) Show the ip configuration for each device

PC-A (VLAN 10): 192.168.10.10/24

PC-B (VLAN 20): 192.168.20.20/24

Switch (VLAN 1): 192.168.1.1/24

Switch (VLAN 1): 192.168.1.2/24

c) Write the commands for VLAN configuration in switch

```
Switch(config)# vlan 10
Switch(config-vlan)# name VLA...
Switch(config)# interface FastEthernet0/1
Switch(config)# interface vlan 10
```

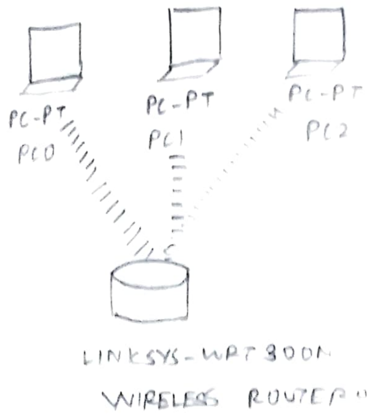
RESULT:

Thus the above program has been executed successfully

EXP.NO: 8

AIM:

b) Configuration of Wireless LAN using CISCO Packet transfer



Perform the following configuration.

Configure static IP on PC & Wireless Router.

Set SSID to Mother Network

Secure your network by configuring WAP.

STEP1: Click on wireless router.

Select the wireless security & change security mode to WEP

Now it is time to connect PC's from wireless router. To do so click PC select Desktop

Repeat the same process on PC1 & PC2

STUDENT OBSERVATION

a) What is SSID of a wireless route.

The service set identifier is the unique name of a wireless network. That allows devices to recognize & connect to specific wireless network.

b) What is security key in wireless route

A security key is a password or passphrase used to authenticate devices to a wireless network.

8/11

Result: The configuration of wireless LAN using cisco packet transfer is successfully.

EXP. NO 9

AIM: Implementation of subnetting in CISCO PACKET TRANSFER simulat.

CREATING A NETWORK TOPOLOGY IN PACKET TRACER

Select the new button & choose 'Network' & 'Generic' this will create a blank network [illegible]

ADDING DEVICES

[illegible] [illegible]

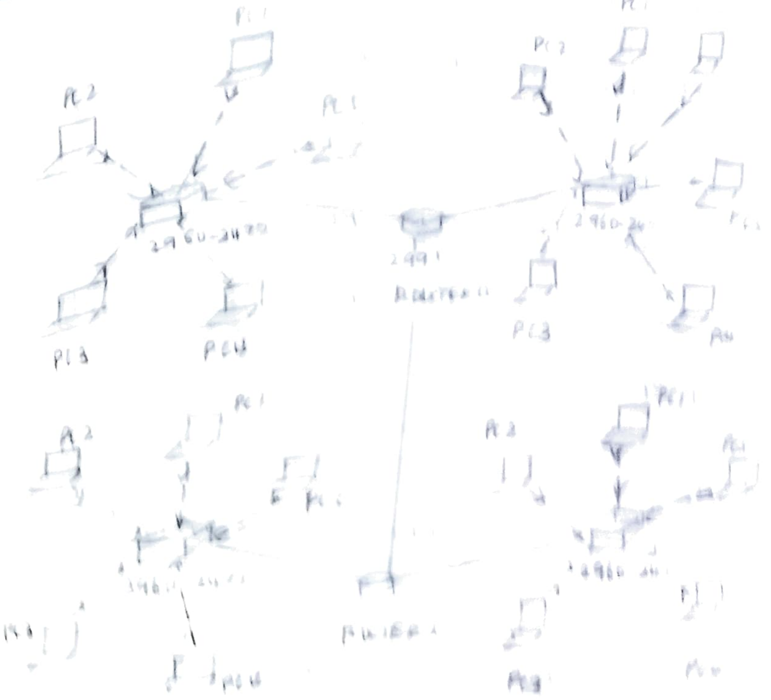
Drag & drop devices onto network [illegible]

Connect devices by [illegible] [illegible]

SUBNETTING

Use a /24 subnet mask.

This will provide 8 subnets with 30 host addresses each



CONFIGURING THE DEVICES

enable

configure terminal

interface FastEthernet0/0

ip address {ip address} [illegible]

no shutdown

exit

interface FastEthernet0/1

ip address {ip address} {subnet mask}

no shutdown

exit

TESTING THE NETWORK

Open command prompt try to ping the other PC.

If successfull, network functions properly.

STUDENT OBSERVATION

a) Write down your understanding of subnetting

It is the process of dividing a larger IP network into smaller, more manageable sub-networks

b) What is the advantase of implementi subnetting within a networ

Improved Network management.

Enhanced Security

Efficient IP Address use

RESULT:

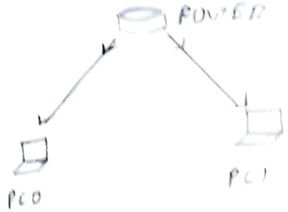
Thus the above program has been executed successfully

Exno: 10

aim

a) Internetworking with routers in CISCO PACKET TRACER simulator

a) Design & configure a simple internetwork using a router



PROCEDURE

Step1: Configuring Router1

Router > enable

Router # config t

Enter configuration commands

STEP2: (Configuring PCs):

Assign IP address to every PC

Assign the default gateway

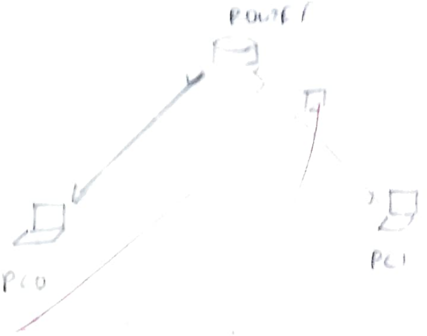
STEP3: (connecting PCs with router)

Connect FastEthernet0 port of PC0 with FastEthernet0/0

ROUTER CONFIGURATION TABLE

DEVICE NAME	IP ADDRESS	SUBNET MASK	IP ADDRESS	SUBNET MASK
ROUTER1	192.168.10.1	255.255.255.0	192.168.20.1	255.255.255.0

ACKNOWLEDGMENT FROM PC1 TO PC0



8/11

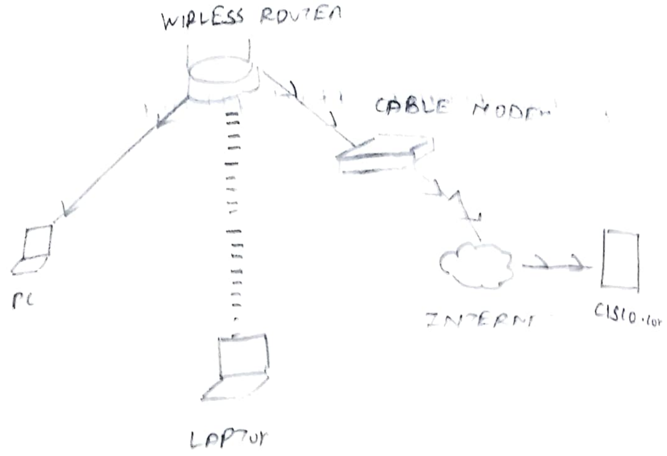
RESULT:

The above program has been executed successfully

EXP.NO: 10

AIM:

b). Design & configure an internetwork using wireless router, DHCP server & internet cloud



PART2: Configure the Network devices

PART-3: Test connectivity between

Part1: Build a Simple Network in the Logical topology Workspay

Step 2: Build the topology

a) Add Network devices to the worspau

b) Change display names of network devices

c) Add the physical cabling between devices on

PART2: Configure the Network devices

Step1: Configure the wireless router

a) Create the wireless network on the wireless router

b) Click on the save settings tab

Step2: configure the laptop

a) Configure the laptop to access the wireless network

Step3: Configure the PC

a) Configure the PC for wired networ

Step4: Configure the internet cloud

a) Install network modules if necessary

b) Identify the From & to Port

c) Identify the type of provide

Step 5: Configure the Cisco.com serve.

a) Configure the Cisco.com serve as a DHCP serve.

b) Configure the Cisco.com server as DNS server to provide domain name to IPv4 address resolution

c) Configure the Cisco. com server global settings

d) Configure the Cisco.com server FastEthernet0 Interface settings

PART3: Verify connectivity

Step1: Refresh IPv4 settings on PC

a) verify that the PC is receiving IPv4 configuration

b) Test connectivity to the Ciso.com

[illegible] [illegible]

1) Write down the key features of configuring Wireless router & DHCP server

Setting the router's IP address

Configuring wireless network

Setting up wireless security

Configuring Quality of service

DHCP server configuration

Setting the DHCP server

Configuring DHCP scope

2) What is the significance of DHCP server in Internetworking

Automatic IP address configuration

Scalability

Flexibility

8/11

RESULT:

Thus the above program has been executed successfully.

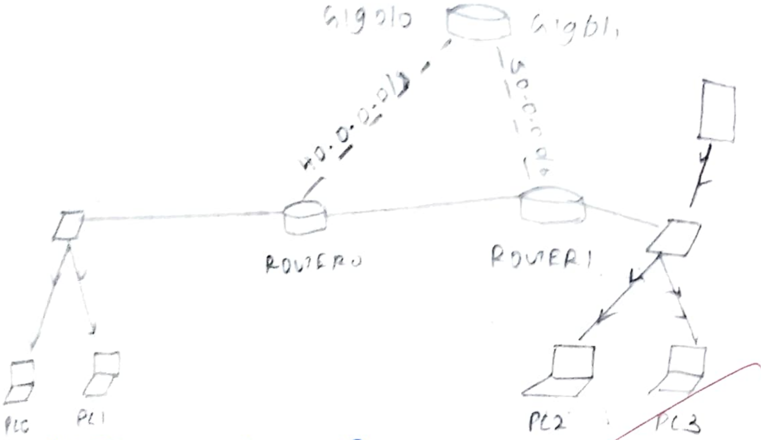
EXP.NO.11

STATIC ROUTING

Aim:

Simulate static routing configuratio using cisco tracer.

The following table lists the connected networks of each router



ROUTER 0 configuratio

```
ip route: 30.0.0.0 255.0.0.0 20.0.0.2 10
ip route: 30.0.0.0 255.0.0.0 40.0.0.2. 20
ip route 30.0.0.100 255.255.255.255
                        40.0.0.2 10
ip route 50.0.0.0 255.0.0.0 20.0.0.2 20
exit
```

ROUTER 1 CONFIGURATIO

```
enable
configure terminal
ip route 10.0.0.0 255.0.0.0 20.0.0.1 10
ip route 10.0.0.0 255.0.0.0 50.0.0.1 10
ip route 40.0.0.0 255.0.0.0 20.0.0.1 10
exit
```

ROUTER 2 CONFIGURATIO

```
enable
configure terminal
ip route 10.0.0.0 255.0.0.0 20.0.0.1 10
ip route 10.0.0.0 255.0.0.0 50.0.0.1 20
ip route 40.0.0.0 255.0.0.0 20.0.0.1 10
ip route 40.0.0.0 255.0.0.0 50.0.0.1 20
exit
```

VERIFYING STATIC ROUTING:

tracert command sends ping results to destination host and tracks the path they take to reach the destination

DELETING A STATIC ROUTE

show ip route static command is used to print all static routes

Note down the route you want to delete

RESULT

Thus the static routting configuration has been identified successfully

EXP.NO 11 RIP USING CISCO PACKET TRACER

AIM Simulate RIP using cisco packet tracer.

INITIAL IP CONFIGURATION

DEVICE	INTERFACE	IP CONF	CONNECTED WITH
ROUTER0	FA0/1	10.0.0.1/8	PC's
ROUTER0	S0/0/1	192.168.1.254/30	ROUTER1's
ROUTER1	S0/0/0	192.168.1.249/30	ROUTER0's S0/0/0

ASSIGN IP ADDRESS TO PCs

Double click PCs & click desktop menu item & click IP configuration assign IP address referring the above table

Assign the IP address to interface of routers

Show controllers interface gives whether the interface is DCE or DTE

CONFIGURE RIP ROUTING PROTOCOL.

ROUTER 0

```
router rip
network 10.0.0.0
network 192.168.1.252
network 192.168.1.248
```

ROUTER 1

```
router rip
network 192.168.1.244
network 192.168. 248
```

ROUTER 2

```
router rip
network 10.0.0.0
network 192.168.1.252
network 192.168.1.244
```

13/11

RESULT:

Thus the simulation of RIP using Cisco packet tracer has been implemented successfully.

EXP No 12

AIM:

a) Implement echo client using TCP/UDP sockets.

1) TCP server algorithm

```
Create a TCP socket
Bead the socket to a local address & port
Accept a client conect
Loop:
    Receive data from client
    If data is received, send it back to client
    else break the loop.
Close the connection
```

TCP client algorithm

```
Create a TCP socket
Connect to the server
Send a message to server
Display & close the socket
```

TCP SERVER

```
import socket
def tcp_server()
    server_socket = socket.socket(
        socket:AF_INET, socket sock_stream)

    server_socket.bind(("localhost",1234))
    server_socket.listen(1)

    print("TCP serveles is waiting for
        a connection").

    connection.client_address = server_socket.
                                    accept()

    print(f"connected to f-client_address})

    try:
        while True:
            data = connection.rec(1024)
            if data:
                print(f"Received:{data.
                            decode()}')
            else:
                break
    finally:
        connection.close()
```

```
if_name:=main
        tcp_server()
```

tcp_client.py

```
    import socket
    def tcp_client()

        client_socket = socket.socket
                (socket.AF_INET,socket,
                        SOCK STREAM)

        client_socket.connect(("localhost",
                        12345))

        try:
            message = input("Enter a
                        message to send'):
            client_socket.sendall(message.
                            encode())
            data = client.socket.recv(1024)
            print(f"Received from server:
                        {data.decode })
        finally:
            client.socket.close()
```

RESULT: The program for using echo client server using TCP has been executed

EX.NO. 13

AIM

To implement the chat client serve

using

CHAT SERVER:

Start the server by creating a socket

When a new client connects add client

keep running

CHAT CLIENT:

Connect to the server by creating a socket.

Start a process to listen to messages from the server.

keep running till the user decides request.

CHAT CLIENT.PY

```
import socket
Import threading
```

```
det
    while true
        try
            message= client_socket.recv(1024)
            if message:
                printt("server: f message")
        except Exception as e:
                printt(f'an error occured!{e}).
                break.

det start_client():
        nod='127.0.0.1'
        port=12345
        client_socket.connect(host, port))
        printt('connected to server.")
while true:
    message=input('You:")

if_name: ="_main_"
    start_client()
```

```
Python chat_server.py
Chat server started on 127.0.0.1:12345
New connection from ('127.0.0.1', 5726)
Received from client : Balaji
Type from message to client: Hello.
```

```
python chat.client.py
Connected to chat server
You: Balaji.G
You: Server: hello
```

RESULT:

(13) Thus the implementation of chart client server using TCP/UDP socket has been successfully executed & verified.

EXP.NO: 13

AIM: Implement your own ping program

ALGORITHM:

ping-client.py

1) Socket creation
2) Then set a timeout.
3) Send a ping message
4) It listens for a response & calculate the time diff.

PROGRAM:

```
import socket
def start.server (host='127.0.0.1', port
                  ='12340')
    with socket.socket(AF_INET)
        print(f"UDP server running on
        {host}{port}")
    while True:
        data, adds.recover
        s.sender
```

```
[illegible]

import socket

import time

def ping server(host='127.0.0.1', port=12345)

    try

        s.settimeout(1)

        start = time.time()

        s.sendto(b'ping', (host, port))

        data, addr = s.recvfrom(1024)

        end = time time()

        print(f "Received f data.decode

    except socket timeout.

            print('Request timeout')

if __name__ == main:

            ping server()
```

RESULT:

Thus the above program has been executed successfully.

EXP NO: 14

AIM

Write a code using python socket to implement packet sniffing

ALGORITHM

Install python & scapy

Create a program open text &

create a file

Set up the packet fram.

Capture network packets

Run the packet sniffer

PROGRAM

```
from scapy all import

def packet-callback(packet)

                    if IP in packet

                        ip_layer = packet[IP]

                    protocol = ip_layer.proto

                    dst.ip = ip_layer.dst


protocol_name = " "
if protocol == 1
        protocal_name = "ICMP"
```

```
elif protocal ==6.
    protocal-name = "TCP"

elif protocal ==17.
    protocal_name = "UDP"

else
    protocal_name='unknown Protocal'

print(f'protocal: {protocal-name}')
print(f'source IP: {src_ip}')
print(f'Destination IP: {dest_ip}')
print('-'*50)

def main():
    sniff(prn=packet_callback, filter='ip',
                          store=0)
if__name__=='__main__':
    main()
```

RESULT:

Thus the above program has been executed successfully

EXP No:15

AIM

To analyze the different types of weblogs using webalizes tool

ALGORITHM:

Step1: Run webalizer window version

Step2: Input web log file (download from web)

Step3: press run webalizer

13/11

RESULT:

Thus the above program to analyze different types of weblogs has been proved successfully

Completed

3/11