VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

D GOWRI CHARAN (1BM21CS059)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
JUN-2023 to SEP-2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "LAB COURSE **TITLE**" carried out by **NAME (USN),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Course Title - (Course code)** work prescribed for the said degree.

Name of the Lab-Incharge	**Dr. Jyothi S Nayak**
Designation Department of CSE BMSCE, Bengaluru	Professor and Head Department of CSE BMSCE, Bengaluru

`

Index

Sl. No.	Date	Experiment Title	Page No.
1	15-06-23	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.	3-10
2	22-6-23	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.	11-26
3	13-7-23	Configure default route, static route to the Router.	27-33
4	13-7-23	Configure DHCP within a LAN and outside LAN	34-43
5	20-7-23	Configure Web Server, DNS within a LAN.	44-50
6	20-7-23	Configure RIP routing Protocol in Routers.	51-54
7	27-7-23	Configure OSPF routing protocol.	55-58
8	3-8-23	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).	59-63
9	10-8-23	To construct a VLAN and make a pc communicate among VLAN.	64-68
10	10-8-23	Demonstrate the TTL/ Life of a Packet.	69-73
11	10-8-23	To construct a WLAN and make the nodes communicate wirelessly.	69-73
12	10-8-23	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	74-76
		CYCLE 2	
13	17-08-23	Write a program for error detecting code using CRC CCITT (16-bits).	77-81
14	17-08-23	Write a program for congestion control using Leaky bucket algorithm.	82-86
15	24-08-23	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	87-92
16	24-08-23	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	93-98
17	24-08-23	Tool Exploration -Wireshark	99-106

Experiment No 1

Cycle-1

Question

Aim of the program Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

Observation

Aim:- Create a topology hence simulate sending a simple PDU from source to destination using hub & switch. as connecting domains

2. Diagram:- Topology

3. procedure:- 1. At first open real time interface.

2. Take 3 computers and hub.

3. connect all the connections from the pc to hub.

4. connect with hub by the copper straight wire

5. click on pc1 and configure ip-address.

6. do the same for all the remaining pc's

7. so, now you can see the green indications at the wire of both ends

8. Now go to simulation mode and send the packets from the hub to pc's

9. you can see that the data will be transfared to every pc and takes back.

Output:-

Pinging 10.0.0.2 with 32 bytes of data:

Request time out!

Reply from 10.0.0.2: bytes=32 time=4ms TTL=128

Reply from 10.0.0.2: bytes=32 time=4ms TTL=128

Ping statistics for 10.0.0.2:

Packets Sent=4, Recieved=3, Lost=1 (25% loss),

Approximate round trip times in milli-seconds:

Minimum=4ms, Maximum 4ms, Average=4ms

Observation:- when we send some packet from hub to pc. at first it send and recieve from all the pc and send info of the packets to all the pc's connected to the hub.

Switch

Topology:-

procedure :-

1. First open real time interface
2. Take 5 computers and switch.
3. connect with the switch for the all the connections from pc to hub.
4. connect with switch by the copper straight wire
5. click on pc1 and configure ip-address 10.0.0.4 and so on.
6. do the same for all remaining pc's
7. So, Now you can see the green indications at the wire of both ends.
8. Now go to simulations mode and send the packets from the ~~hub~~ switch to pc's
9. you can see the data will be transfered to every pc and takes back

output :-

PC>ping 10.0.0.5

pinging 10.0.0.5 with 32 bytes of data:

Reply from 10.0.0.5 bytes=32 time=0ms TTL=128
Reply from 10.0.0.5 bytes=32 time=0ms TTL=128
Reply from 10.0.0.5 bytes=32 time=0ms TTL=128
Reply from 10.0.0.5 bytes=32 time=0ms TTL=128

ping statistics ~~from~~ for 10.0.0.5:

Packets: sent=4, received=4, lost=0 (0% loss)
Approximate round trip times in milli-seconds:
minimum=0ms

observation:- in simulation mode

PC3 sends packets to switch and it sends to both PC4 To PC5 in first round.

PC4 reject PC5 accepts and send acknologment packets to both PC3 PC4 as cacs it PC5 accepts it also when PC3 sends packets it sends only to PC5.

Topology:-

procedure:-

1. Connect the hub and the switch using copper cross-over.
2. Connect the pc such to hub and switch
3. Select a source pc from the one that is connected to the hub select a distination pc that is connected to the switch.

4. select PDU packets to the source and such on the distination from PC 0 To PC4.

output :- Reply from 10.0.0.4 bytes = 32
time = 0ms TTL = 128

Reply from 10.0.0.4 : bytes = 32
time = 0 ms TTL = 128.

Reply from 10.0.0.4 : bytes = 32
time = 0ms TTL = 32.

ping statistics for 10.0.0.4.
minimum = 0ms, Maximum = 4ms Average = 1ms.

observation :- In simulation mode In sim pc0 sends packets To hub. Hub sends it to pc1, pc2 and switch.

Hub is broadcast it to all pcs only pc0 accept it and others discord.
In second second pc0 such packet to hub.
Switch Now switch broad cost if any to pc3 this switch is a smart device.

15/6/2023

Topology

Output

PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=1ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Ping statistics for 192.160.1.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

Experiment No 2

Cycle-1

Question

Aim of the program Configure IP address to routers (one and three) in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

Observation

Step1:- type No & press Enter.

Step2:- type enable & press enter.

Step3:- type config T & press Enter.

Step4:- type interface fast Ethernet 1 & press Enter.

Step5:- type Ip address 10.0.0.10 255.0.0.0 & press Enter.

Step6:- type No shut & press enter.

Step7:- type Exit.

Step8:- type interface fast Ethernal 40 & press Enter.

Step9:- type Ip address 20.0.0.10 25.0.0.0 & press Enter.

Step10:- type No shut & press Enter.

Step11:- type Exit.

Step12:- type Exit.

Step13:- type show Ip route [for sting the connection states] close the table & click on pc to go to command prompt. type ping 20.0.0.1 to send packets across. Atleast send packets in simulation mode to get a sucess ful transmission.

PING output:

Packet traces pc command line 1.0.

PC > ping 20.0.0.1

pinging 20.0.0.1 with 32 bytes of data.

Request timed out.

Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 127

Reply from 20.0.0.1 bytes = 32 times = 0ms TTL = 127

Reply from 20.0.0.1 bytes = 32 times = 10ms TTL = 127

Aim :- Configure IP address to a single router Explore the following message ; ping message destination. unreachable request timed out . reply.

Topology

Procedure :-

① Select one generic router and 2 generic pc's. connect the pc's to router using copper cross-over Cable

② set the IP address of both pc's by clicking on pc and config tab Along with Ip address.

③ To set the IP address of router, click on it & go to CLI tab and type the following commands

Ping statistics for 20.0.0.1

Packet : sent 4, recieved = 3, lost = 1 (25% loss)

Approximate round trip times in milli-second.

minimum = 0ms, Maximum = 10ms, Average = 3ms

Aim:- configure IP address to three routers in packets tracer explore the following messages: ping response destination unreachable, request timed out. reply.

Topology

Procedure:-

① Connect 2 pc's & 3 routers using copper cross over cable for pc to router and serial DCE cable to connect the router to router.

② set the IP address of both pc's and gateway numbers;

③ Now for setting IP address & gateway number to routers select one route and perform the following commands.

step 1:- type No & press Enter.

2:- type Enable & press Enter.

3:- type config T & press enter.

Step4:- type interface first ethernet 1/0 & press Enter

Step5:- type ip address 10.0.0.10 255.0.0.0 & press enter

Step6:- type No shut & press Enter.

Step7:- type Exit.

Step8:- type interface se 2/0 & press Enter

Step9:- type ip address 20.0.0.10 255.0.0.0 & press Enter

Step10:- type No shut & press Enter.

Step11:- type Exit.

Step12:- type Exit

* Repeat these command for the other two routers with respective IP/gateway address.

* Now to interduce the other two ip address to the first router we type the following commands.

Step1:- type config t & press Enter.

Step2:- type Ip route 30.0.0 255.0.0.0 20.0.0.20.

Step3:- type Ip route 40.0.0 255.0.0 20.0.0.20

Step4:- Exit

Step5:- Exit.

Step6:- type show ip route.

* Repeat these steps for other two routers with appropriate addresses

* Go to command prompt by clicking on pc & config tab. Type ping message to send packets to the destination address

PING OUTPUT

output - 1

packet tracer PC command line 1.0

PC > ping 40.0.0.1

ping 40.0.0.1 with 32 bytes of data reply from 10.0.0.10 :- Destination host unrelatable reply from 10.0.0.10 : destination host unreachable.

Reply from 10.0.0.10 Destination host unreachable.

Request timed out

Ping statistics for 40.0.0.1

packets :- Sent = 4 Recieved = 0 lost = 4 (100% loss)

output 2

packet tracer PC command line 1.0

PC> ping 10.0.0.1

Ping 10.0.0.1 with 32 bytes of data.

Reply from 10.0.0.1 byte = 32 times = 2ms TTL = 125

Reply from 10.0.0.1 : byte = 32 times = 8ms TTL = 125

Reply from 10.0.0.1 byte = 32 times = 2ms TTL = 125

Reply from 10.0.0.1 byte = 32 time 2ms TTL = 125

Ping statistics for 10.0.0.1 :

packets sent = 4 recived = 4 lost = 0 (0% loss)

Approximate round trip time in multi seconds

minimum = 2ms, maximum = 8ms, Average = 3ms

PING OUTPUT

output - 1

packet tracer PC command line 1.0

PC > ping 40.0.0.1

ping 40.0.0.1 with 32 bytes of data reply from 10.0.0.10 :- Destination host unreachable reply from 10.0.0.10 : destination host unreachable.

Reply from 10.0.0.10 Destination host unreachable.

Request timed out

Ping statistics for 40.0.0.1

packets :- Sent = 4 Recieved = 0 lost = 4 (100% loss)

output 2

Packet tracer PC command line 1.0

PC> ping 10.0.0.1

Ping 10.0.0.1 with 32 bytes of data.

Reply from 10.0.0.1 byte = 32 times = 2ms TTL = 125

Reply from 10.0.0.1 : byte = 32 times = 8ms TTL = 125

Reply from 10.0.0.1 byte = 32 times = 2ms TTL = 125

Reply from 10.0.0.1 byte = 32 time 2ms TTL = 125

Ping satistics for 10.0.0.1 :

Packets sent = 4 Recived = 4 lost = 0 (0% loss)

Approximate round trip time in multi seconds

minimum = 2ms, Maximum = 8ms, Average = 3ms

Observation:-

In program 2.1 when we ping the destination address we get alocated with 32 bytes. In this first 8 bytes are used to lead about the route & their addresses. Rest bytes are used for sending packets to destination address. Then again if we ping all bytes are used for message sending and there will be no timed-out message.

* In program 2.2 when the route doesn't know about the remaining address, and we ping a message we get host unreadable message, once the ping reaches route / knowledge about other adress. message will be set sucessfully.

Topology

PROGRAM 2.1

PROGRAM 2.2

OUTPUT:

PROGRAM 2.1

PROGRAM 2.2

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Request timed out.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100%
loss),
PC>

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 8ms, Average = 3ms
PC>

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Simulation Panel
Event List
Vis. Time(sec) Last De At Dev Type Info
28.315 -- Rout... CDP
28.316 Router5 PC0 CDP
28.316 Router5 Rout... CDP
45.862 -- Rout... CDP
45.862 -- Rout... CDP
Reset Simulation
Constant Delay
Captured to: 45.862 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
Router1
Router5
Time: 01:54:00.015
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Statu Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC1 IC... 0.000 N 0 (ed... (delete)

WEEK 3

AIM : Configure default route, static route to the Router.

Observation:

1/10/23 Program - 3.1

Aim:- Configure default route : static route to the router.

Topology:-

Procedure:-

1. connect 3 Routers and 2 pc's using copper cross-over cable for PC to router and a serial DCE cable to connect router to router.
2. If the IP Adress of both pc's and respective gate ways turns.
3. For all 3 Routers set the respective 2 IP address in CLI mode by using these commands.

Step1:- Enable

Step2:- Configure T

Step3:- Interface fast ethernet 0/0

Step4:- Ip Address 10.0.0.10 255.0.0.0.

Step5:- No shut.

Step6:- Exit.

Step7:- Interface Se 2/0

Step8:- IP Address 20.0.0.10 255.0.0.0.

Step9:- No shut.

Step10:- Exit.

> Repeat these command for other two lowest with the respective Ip Addresses.

> for router 1. let the Ip Address route of others Ip Address actually by using following steps.

Step1: configT

Step2:- Ip route 10.0.0.0 255.0.0.0 20.0.0.10

Step3:- Ip route 40.0.0.0 255.0.0.0 30.0.0.20

Step4:- Exit

Step 5:- Exit

Step 6:- show Ip route

> for routes 0 & routes 2 we set default routes which means it can route access any ip address with any subset mask.

> But the default Ip route by following these commands.

Step1:- configT

Step2:- Ip route 0.0.0.0 0.0.0.0 2.0.0.0.20

step 3: Ip route 40.0.0.0 255.0.0.0 30.0.0.20.

step4: exit.

step5: exit.

step6: show Ip route.

for router 0 & router 2 we set default ~~to~~ Ip router by following these commands

step1: config T

step2: Ip router 0.0.0.0 0.0.0.0 20.0.0.20.

step3: Ip router 0.0.0.0 0.0.0.0 30.0.0.10.

step2 is given for Router 0 & step 3 command for router 1.

> Goto PC's command prompt and type ping message to send packets across.

PING OUTPUT

Packet Tracer PC command line 1.0.

PC> ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data.

Request timed out.

Reply from 40.0.0.1 bytes 32 times =2ms TTL=125

Reply from 40.0.0.1 bytes 32 times = 16ms TTL=125

Reply from 40.0.0.1 bytes 32 times =2ms TTL=125

Ping statistics for 40.0.0.1 :

Packets : send = 4 Recived = 3 Lost = 1 (25 % loss)

Approximate round trip time in milli seconds

Minimum = 2ms, Maximum = 16ms Average = 6ms.

OBSERVATION :-

> A default rate is the which takes effect when no other router is available for an IP address destination.

> If a packet is recived, the device first checks the IP destination address, if the ip distination address is not local the devio checks its routing table.

> If the remote destination subnet is not listed then the packet is forwarded to the next hop towards the destination using the default route.

> The process supseeds until the packet is delivered.

Topology:

OUTPUT:

Output:

PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 16ms, Average = 6ms

PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time=21ms TTL=125
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 21ms, Average = 9ms

PC>
```

WEEK 4

Aim :-Configure DHCP within a LAN and outside LAN.

Observation:-

Week 4

Program - 4.1

AIM :-

Configure DHCP within a LAN and outside LAN

Topology :-

Procedure :-

> connect 3 pc's and 1 server to a switch using upper straight through cable.

> click on server and go to services tab select DHCP and turn on the DHCP service.

> set the IP address of the start IP address as 10.0.0.2 and click on save button.

> Before this, set the IP address of server in. config Tab under fastnernet as 10.0.0.1

> Next click on PC0 and go to desktop tab, run click on IP configuration select DHCP here. It will request for an IP address and successfully gets the DHCP request also site the IP address.

> Repeat this steps for other 2 pc's

> To send a packet across pc's, go to pc's command prompt and type ping distination ip address.

PING output:

> packet from PC command line 1.0

PC> ping 10.0.0.3.

pinging 10.0.03 with 32 bytes of data.

Reply from 10.0.0.3: bytes = 32 times = 0ms TTL = 128

Reply from 10.0.0.3: bytes = 32 times = 0ms TTL = 128

Reply from 10.0.0.3: bytes = 32 times = 1ms TTL = 128

Reply from 10.0.0.3: bytes = 32 times = 0ms TTL = 128

Ping statistics from 10.0.0.3.

Packets: Send = 4, Recived = 4, lost = 0 (0% loss)

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 1ms, Average = 0ms

Observation:-

> DHCP is used to dynamically assign an IP address to any device or node.

> It is a client-server protocol in which servers manage a pool of unique ip addresses & also about client configuration parameters.

> DHCP-enabled clients sends a request to DHCP server, when they want to connect to a network.

> The DHCP server responds to the client request by providing ip configuration information from address pools, previously specified by a network administrator.

Program 4.2

AIM :- Configure DHCP within a LAN and outside LAN:

Topology :-

Procedure :-

> Add a Router, a switch and 2 pc's to 4.1 program Network & connect the router to both switches.

> Set the server Ip address of server and with the help of Server set the first 3 pc's Ip address through DHCP

> Now set the Router Ip address with the following commands statically.

Step1 : No

step2 : Enable.

step3 : Config T

Step4 : Interface fastEthernet 4/0.

Step5 : IP address 10.0.0.20 255.0.0.0

step6 : No Shut.

step7 : Exit.

step8 : interface fastEthernet 0/0.

step9 : IP address 20.0.0.20 255.0.0.0

Step 10 : No shut.

step11 : Exit.

step12 : exit.

step13 : show IP route.

> Go to server and set the gateway as 10.0.0.20

> Again go to router CLI and follow this commands

step14 : Config T

Step15 : interface fastEthernet 0/0

step16 : IP helper - address 10.0.0.1

Step17 : No shut.

step18 : Exit.

> Now go to server services and add one more poolname as server pool1, start IP address as 20.0.0.2 and default gateway as 2000.0.20 Then click and save.

> Now set the other two PC's IP address by going to this Desktop → IP configuration and selecting DHCP which will automatically generate its IP address

> Now the PC to others by typing ping destination IP address in their respetive command prompts

PING OUTPUT:-

packet tracer PC command time 1.0

PC > ping 20.0.0.2

pinging 20.0.0.2 with 32 bytes of data.

Request time out.

Reply from 20.0.0.2 : bytes = 32 time = 0ms TTL = 127

Reply from 20.0.0.2 : bytes = 32 time = 0ms TTL = 127

Reply from 20.0.0.2 : bytes = 32 time = 0ms TTL = 127.

ping statistics for 20.0.0.2.

packet sent = 4, Recieved = 3, Lost = 1 (25% loss)

Approximate round trip time in milli-seconds:

minimum = 0ms, maximum = 0ms, Average = 0ms.

OBSERVATION:

DHCP is used to assign IP address dynamically to different devices.

To assign continously IP address we create a sever pool where we assign the starting IP address and a default gateway number. for PC's under different switches we create a different servespool again and start.

This takes care of delivering again and start.

This takes care of destination IP address and also sends back the ACK to the intial device.

Topology:

PROGRAM 4.1:

PROGRAM 4.2:

OUTPUT:

PROGRAM 4.1:

PROGRAM 4.2:

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 20.0.0.3
Pinging 20.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab4.2.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
PC-PT
PC3
PC-PT
PC4
Simulation Panel
Event List
Vis. Time(sec) Last De At Dev Type Info
2.992 Switch0 Serve... STP
2.992 Switch0 Rout... STP
2.992 Switch0 PC0 STP
2.992 Switch0 PC1 STP
2.992 Switch0 PC2 STP
Reset Simulation
Constant Delay
Capturing...
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:32:11.943
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Scenario 0
New
Delete
Fire Last Stat Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC3 IC... 0.000 N 0 (ed... (delete)

WEEK 5

AIM:Configure Web Server, DNS within a LAN. OBSERVATION:

Observation:-

lab-5

AIM :- configure web server, DNS contain a LAN

Topology :-

Procedure :-

> connect a switch PC and a server to form a LAN
> set pc's IP address by clicking on it and go to config tab. There in fast Ethernet option set IP address as 10.0.0.1 and subnet mask.
> set server's IP address as 10.0.0.2 and subnet mask respectively.
> Go to PC's Desktop and click on web browser, in the URL tab type 10.0.0.2. You will get a default display
> To make a cv here, we need to make changes in server services.
> Go to server -> services -> HTTP -> index.html. Here. create the cv and click on save.
> Again go to PC -> Desktop -> web browser and type 10.0.0.2

you will the CV or content that is changed

> Next, go to server → services → DNS and switch on the service Now add a Domain name and type the Ip address as 10.0.0.2 press add & save it.

> Again Go back to PC → Desktop → webbrowser and type the given domain name. then we can see the CV which had been create earlier.

Output:-

web browser,

[<] [>] URL [http://gaurricharan] [Go] [Stop]

CV

Gowri charan.
USN : 1BM21CS059.
Language: C/C#/Java.

Image

OBSERVATION:-

> If you wanted to go to a certain website you would open web browser and type domain name of that website or else you can also type the Ip adress instead if you known that website Ip address.

> since we can't remember Ip address of all website DSN Server will search through it's cache to find a matching Ip address for that Domain names & when it finds it will resolve that domain name to Ip address of website, once that is done then completes is able to communicate with a webserver & retrieve the webpage

Topology:-

OUTPUT:

WEEK 6

AIM: Configure RIP routing Protocol in Routers.

OBSERVATION:

week-6

AIM:- configure RIP routing protocol in Routers.

Topology:-

Procedure:-

> create a Network using 3 routers and 2 pc's connect routers using serial DCE cable and PC to router using copper - cross over cable.

> set the IP address and gate way no for both pc's as,

10.0.0.0.1 - IP 10.0.0.10 gateway - PC0.

40.0.0.1 - IP 40.0.0.10 - gateway - PC1 respectively.

Go to Router -> CLI mode and execute the following.

Commands:-

Step1 - No

Step2 - Enable

Step3 - Config T

Step4 - interface fastethernet 0/0

Step 5 - ip address 10.0.0.10 255.0.0.0.

Step 6 - No shut.

Step 7 - Exit.

Step 8 - Interface s0/0.

Step 9 - Ip address 20.0.0.10 255.0.0.0.

Step 10 - Encapsulation ppp.

Step 11 - clock rate 64000.

Step 12 - No shut.

> Here for router with fast Ethernet execute only till step 1 and type No shut.

> only for Router to Router connection execute all steps. connection with has a clock symbol at start.

> Again go to Router 0 -> CLI mode and type these steps.

Step 1: configT.

Step 2: router rip.

Step 3: Network 10.0.0.0.

Step 4: Network 20.0.0.0.

Step 5: Exit.

Step 6: Repeat these steps for all routers.

> At last now go to Each router and type show Ip route these the Ip address associated with that router will be labelled as c and other ip addresses are labelled as R.

> lastly go to PC0 and ping a message to PC1 using ping destination Ip address command.

PING OUTPUT:-

Packet tracer PC command line 10.

PC > ping 40.0.0.1 with 32 bytes of data.

Request timed out.

Reply from 40.0.0.1 : bytes 32 times : 8ms TTL = 125

Reply from 40.0.0.1 bytes 32 times = 5ms TTL = 125.

Reply from 40.0.0.1 : bytes 32 times = 10ms TTL = 125

Ping statistics for 40.0.0.1 :

Packet sent = 4, Recieved = 3 lost = 1 (25% loss)

Approximate round trip times = 3 lost in milli seconds

Minimum = 5ms Maximum = 10ms Average = 7ms.

OBSERVATION:-

> Routing information protocol (RIP) is a dynamic routing protocol that uses hop count as a routing metric to find the best path between source and destination. It is a distance - vector routing protocol.

> Hop count is the no of routers coming in between source and destination. The path which least hop count is selected.

> updates of the Network are exchanged periodically

TOPOLOGY:

OUTPUT:

Logical
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
20.0.0.20
30.0.0.10
10.0.0.10
20.0.0.10
30.0.0.20
40.0.0.10
10.0.0.1
40.0.0.1
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Simulation

WEEK 7

OBSERVATION:

Week-7

AIM:- configure OSPF routing protocol:

procedure:-

> create the topology using 3 routers & 2 pc's
> configure the pc's with Ipaddress and gateway.
> configure each of routers acc to Ip address given.
> During configuration, encapsulation PPP and clock rate should be set as done in RIP protocol
> Then execute the following commands.

Router → CLI → configmode.

Step1 - router ospf 1

Step2. router-id 1.1.1.1

Step3. Network 10.0.0.0 0.255.255.255 area 3.

Step4. Network 20.0.0.0 0.255.255.255 area 1.

Step5. Exit.

> Repeat these commands for other routers.

> Then type show ip route.

> Next to set loopbacks.

Step1: cin.config - if mode). interface loopback 0.

Step2: ip address 192.168.1.252 255.255.0.0.

Step3:- no shut down.

> Repeat these steps for other 2 routers.

> create a virtual link between R1, R2, by this we create a virtual link to connect to area 0.

In config mode of R1

Step1: router ospf 1

Step2: area 1 virtual - link 2.2.2.2.

Step3: # enter / exit.

In router 2 config mod.

Step 1 : # router ospf 1.

Step2 : area 1 virtual 1.1.1.1

Step3 : exit.

Step4 : #

> check the routing table show ip route

> lastly ping message from pc to pc.

PING OUTPUT

Packet tracer PC command line 1.0

PC> Ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data.

Request timed out.

Reply from 40.0.0.10 : bytes=32 time=11ms TTL=125

Reply from 40.0.0.10 : bytes = 32 times = 11ms TTL = 125
Reply from 40.0.0.10 : bytes = 32 times = 11ms TTL = 125
Reply from 40.0.0.10 : bytes = 32 times = 8ms TTL = 125

Ping statistics for 40.0.0.10.
packet sent = 4 Recived = 3, lost = 1 (25% loss)
Approximate round trip times in milli seconds.
Minimum = 8ms, Maximum = 11ms, Average = 10ms.

Observation:-

> OSPF is a link-state routing protocol that is used to find the best Path between source destination routers using it's own SPF Algorithm.
> This Network is divided into 4 areas. where area is the backbone.
> After we make the virtual-link between the area which is not connected to the observation, we pinging message succefully.

TOPOLOGY:

OUTPUT:

WEEK 8

AIM-To construct a simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).

OBSERVATION:

LAB - 8 3/Aug/2023

Aim: To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Topology:-

Procedure:-

1. create a topology of 4 pcs and a server.
2. Ip address assigned to all
3. Connect them through a switch
4. use the inspect tool to click on a pc to see the ARP Table.
5. Command in CLI for the same is arp - a
6. Initially ARP Table is empty.
7. In CLI of switch, the command - show mac address

table can be clearly noted.

8. Observe the switch as well the nodes update the ARP table as and new communication starts.

Ping output:-

pc> arp -a

Internet Address	Physical Address	Type
10.0.0.1	0001.c904.5457	dynamic

Switch> show mac address-table

Mac Address Table

Vlan	Mac Address	Type	Ports
1	0.001.6400.8ec6	DYNAMIC	Fa1/1
1	0.001.C904.5457	DYNAMIC	Fa3/1
1	0009.7C19.7260	DYNAMIC	Fa1/1
1	00d0.d350.5b40	DYNAMIC	Fa0/1

Observation:-

When we ping PC and server the address of the server is known to PC and vice versa when we ping between other two PC's similarly the address of the each other are known by ARP protocol. As switch is the common between all, so when you see the mac address table you can see all the PC and server mac address.

TOPOLOGY:

OUTPUT:

Logical
IOS Command Line Interface

WEEK 9

AIM:To construct a VLAN and make a pc communicate among VLAN.

OBSERVATION:

> Go to router -> CLI and type the following command

Step1:- config T
Step2:- interface fa 0/0
Step3:- ipaddress 192.168.1.1 255.255.255.0
Step4:- No shut.
Step5:- exit.
Step6:- config T
Step7:- interface fa 0/0.1
Step8:- encapsulate dot1q 2.
Step9:- ip address 192.168.20.4 255.255.255.
Step10:- no shut
Step11:- exit

ping message from pc to another VLANPC.

PING OUTPUT

Packet Tracer PC command line 1.0

PC> ping 192.168.20.3.
Pinging 192.168.20.3 with 32 bytes of data:
Requested timed out.
Reply from 192.168.20.3 : bytes = 32 time = 0ms, TTL=127
Reply from 192.168.20.3 bytes = 32 time = 5ms TTL=127

Ping statistics for 192.168.20.3.

Minimum = 0ms, Maximum = 5ms, Average = 1ms

Observations -

> we can have one device on one VLAN & another in another VLAN connected to the same switch. They will only hear other VLAN s, as if they over connected to two switches.

> These VLANS dosent use Ip address instead deal with subnets / classe type address.

TOPOLOGY:

Tools Extensions Help
[Root]
New Cluster
Move Object
Set Tiled
192.168.1.1
192.168.20.1
PC-PT
PC0
192.168.1.2
PC-PT
PC1
192.168.1.3
PC-PT
PC2
192.168.20.2
PC-PT
PC3
192.168.20.3

OUTPUT:

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3
Pinging 192.168.20.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.20.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 5ms, Average = 1ms
PC>

Logical
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
Simulation Panel
Event List
Play Controls
Simulation

WEEK 10

AIM:Demonstrate the TTL/ Life of a Packet. OBSERVATION:

Observation:

LAB - 10

Aim:- Demonstrate the TTL / life of a packet.

Topology:-

Procedure:-

- create a topology as shown above with two pc's and 3 routers.
- set the IP address and gateway for both pc's.
- configure the routers either static/default routing way
- In simulation mode send a simple PDU from one pc to another.
- use capture button to capture every transfer.
- click on the PDU during every transfer to see the In bound & outbound PDU details.

Observation:-

- The no. of hops the packet travels before being discarded or TTL
- The routers reduce TTL value by one while forwarding the packets

OUTPUT :-

IP

0	4	8	16	19	31
4	HL	DSCP	TL:28		
ID 0x6			0x	0x0	
TTL:255		PRO:0x1	CHKSUM		
SRC IP: 10.0.0.1					
Dst IP: 40.0.0.1					
OPT 0x0				0x0	

LAB - 1 (DATA C VARIABLE LENGTH)

TOPOLOGY:

OUTPUT:

AREA 1
20.0.0.2
10.0.0.1
20.0.0.1
40.0.0.1
AREA 3
AREA 2
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10

AREA 1
20.0.0.2
10.0.0.1
20.0.0.1
40.0.0.1
AREA 3
AREA 2
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10

AREA 1
20.0.0.2
10.0.0.1
20.0.0.1
40.0.0.1
AREA 3
AREA 2
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10

WEEK 11

AIM:To construct a WLAN and make the nodes communicate wirelessly

OBSERVATION:

LAB-11 (PATH VARIABLE LENGTH)

AIM :- To construct a WAN and make the nodes communicate wirelessly.

Topology :-

Procedure :-

- construct the above topology.
- configure E0 & router0 as normally done.
- Configure Access point 1 - port 1 → SSID Name - WLAN
- select WEP & give any 10 digit hexat key standards
- switch off the device, drag the existing PT-Host -

NM-1AM to the component listed in LHS. Drag wireless interface and on the device.

> In the configure tab a new wireless interface would have been added, and gateway to the device.

> ping from every device to every other device.

pingoutputs: packet Traces pc command time 1.0.

pc> ping 10.0.0.3,
pinging 10.0.0.3 with 32 bytes of data:
Request timed out,
Reply from 10.0.0.3: bytes=32 time=0ms TTL=127.
ping statistics for 10.0.0.3,
packets: sent=4, Recieved=3, lost=1 (25% loss)
Approximate round trip time in milli seconds:
minimum=0ms, maximum=1ms, Average=0ms.

OBSERVATIONS:> A WLAN is a group of colocated devices that form a network based on radio transmissions.

> Data sent in packets contain layers with labels & instructions, MAC address to endpoints for routing.

> with one access point we can connect to multiple device wirelessly & transmit data.

TOPOLOGY:

Logical
10.0.0.1
Router0
Fa0/1
Fa0/1
Switch0
Access Point0
PC-PT
PC0
10.0.0.2
PC-PT
PC2
10.0.0.3
Laptop-PT
10.0.0.4
Realtime

PC1
Physical
Config
Desktop
Custom Interface
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

Laptop0
Physical
Config
Desktop
Custom Interface
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

OUTPUT:

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=21ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=10ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 21ms, Average = 11ms
PC>

WEEK 12

AIM:To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

OBSERVATION:

LAB-12

Aim :- To understand the operational of TELNET by accessing the router in server room from a pc in IT office.

Topology :-

procedure :-

> create a topology as shown in the figure
> configure the router by existing the following commands.

step1 :- enable.
step2 :- config T
step3 :- hostname r1
step4 :- enable secret p1
step5 :- interface fastethernet 0/0
step6 :- IP address 10.0.0.1 255.0.0.0.0
step7 :- No shut
step8 :- line vty 0 5
step9 :- login
step10 :- Password p0
step11 :- exit, exit
step12 : wr

Ping message to router
Password for user Access verification is P0.
pasword for enable is P1
Assining router c1 from PC.
show Ip route.

PING OUTPUTS:-

Packet Tracer PC command line 1.0
PC> Ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data.
Reply from 10.0.0.1: bytes 32 times = 0ms TTL = 255
ping statistics for 10.0.0.1
packets sent = Recieved = lost = 0 (0% loss)
PC> telnet 10.0.0.1
Typing 10.0.0.1... open.
User Access verification.
Password P0
r1> enable.
password: P1:
r1 # show Ip route.
C 10.0.0.0/8 is directly connected, fast Ethernet -

Observation:-

> TELENET stands for Teletype Network. It is a type of protocol that exabids one computers to connect to the local computer.

> It is used as a standard TCP/IP protocol for virtual terminal arrival provided by ISO.

TOPOLOGY:

OUTPUT:

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=1ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
% Password: timeout expired!
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
Password:
Password:
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
r1>enable
Password:
r1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
r1#

CYCLE 2

WEEK 13

Write a program for error detecting code using CRC- CCITT (16-bits).

CODE:

```
#include<stdio.h> int arr[17];

void xor(int x[], int y[]) {

int k=0;
 for(int i=1;i<16;i++) {
```

```
if(x[i]==y[i]) arr[k++]=0;

else arr[i]=1;

} }

void main() {

int dd[17],div[33],ze[17],i,k;

printf("Enter the dataword \n"); for(i=0;i<17;i++)

scanf("%d",&div[i]);

for(i=i;i<33;i++) div[i]=0;

for(i=0;i<17;i++) ze[i]=0;
printf("Enter dividend \n");


for(i=0;i<17;i++) scanf("%d",&dd[i]);

i=0; k=0;

for(i=i;i<17;i++) arr[k++]=div[i];

while(i<33) {

if(arr[0]==0) xor(arr,ze);

else xor(arr,dd);

arr[16]=div[i++];
```

```
}
 k=0; for(i=17;i<33;i++)

div[i]=arr[k++]; printf("Codeword: ");

for(i=0;i<33;i++) printf("%d",div[i]);

for(i=0;i<17;i++) arr[i]=0;

printf("\nAt receiver end \n");

k=0; for(i=i;i<17;i++)

arr[k++]=div[i]; while(i<33)

{
if(arr[0]==0) xor(arr,ze);

else xor(arr,dd);

arr[16]=div[i++];

}
 k=0; for(i=17;i<33;i++)

div[i]=arr[k++];

printf("Codeword: "); for(i=0;i<33;i++)

printf("%d",div[i]);
}
```

OUTPUT:

```
C:\Users\Admin\Desktop\1BM21CS047\ADA\CRC16\bin\Debug\CRC16.exe
Enter the dataword
1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1
Enter dividend
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1
Codeword: 10110011110010111000000000011011
At receiver end
Codeword: 10110011110010111000000000000000
Process returned 1 (0x1)   execution time : 49.507 s
Press any key to continue.
```

WEEK 13

Write a program for error detecting code using CRC- CCITT (16-bits).

CODE:

```
#include<stdio.h> int arr[17];

void xor(int x[], int y[]) {

int k=0;
 for(int i=1;i<16;i++) {

if(x[i]==y[i]) arr[k++]=0;

else arr[i]=1;

} }

void main() {

int dd[17],div[33],ze[17],i,k;

printf("Enter the dataword \n"); for(i=0;i<17;i++)
```

```
scanf("%d",&div[i]);

for(i=i;i<33;i++) div[i]=0;

for(i=0;i<17;i++) ze[i]=0;

printf("Enter dividend \n");





for(i=0;i<17;i++) scanf("%d",&dd[i]);

i=0; k=0;

for(i=i;i<17;i++) arr[k++]=div[i];

while(i<33) {

if(arr[0]==0) xor(arr,ze);
```

```
else xor(arr,dd);

arr[16]=div[i++];

}
 k=0; for(i=17;i<33;i++)

div[i]=arr[k++]; printf("Codeword: ");

for(i=0;i<33;i++) printf("%d",div[i]);

for(i=0;i<17;i++) arr[i]=0;

printf("\nAt receiver end \n");

k=0; for(i=i;i<17;i++)

arr[k++]=div[i]; while(i<33)

{

if(arr[0]==0) xor(arr,ze);
```

```
else xor(arr,dd);

arr[16]=div[i++];

}
 k=0; for(i=17;i<33;i++)

div[i]=arr[k++];

printf("Codeword: "); for(i=0;i<33;i++)

printf("%d",div[i]); }
```

OUTPUT:

```
C:\Users\Admin\Desktop\1BM21CS047\ADA\CRC16\bin\Debug\CRC16.exe
Enter the dataword
1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1
Enter dividend
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1
Codeword: 10110011110010111000000000011011
At receiver end
Codeword: 10110011110010111000000000000000
Process returned 1 (0x1)   execution time : 49.507 s
Press any key to continue.
```

OBSERVATION :

17/August

Lab 13

13- Aim:- write a program for error detecting code using CRC-CCITT (16-bits)

Code:-

```
#include <stdio.h>
int arr[17];
void xor (int x[], int y[])
{ int k=0;
  for (int i=1; i<16; i++)
  {
    if (x[i] == y[i])
        arr[k++]=0;
    else
        arr[i]=1;
  }
}
void main()
{
  int dd[17], div[33], ze[17], i, k;
printf("Enter the data word \n");
for (i=0; i<17; i++)
    scanf("%d", &div[i]);

for (i=1; i<33; i++)
    div[i]=0;

for (i=0; i<17; i++)
    ze[i]=0;

printf("Enter dividend \n");
for (i=0; i<17; i++)
    scanf("%d", &dd[i]);
```

```
i=0;
k=0;
for (i=i; i<17; i++)
    arr[k++] = div[i];
while (i<33)
{
  if (arr[0] == 0)
     xor (arr, 7e)
  else
     xor (arr, dd);
arr[k] = div[i++];
}
k=0;
for (i=17; i<33; i++)
div[i]=arr[k++];
printf("codeword:");
  for (i=0; i<33; i++)
  printf("%d", div[i]);
for (i=0; i<17; i++)
    arr[i] = 0;
printf("\n At received end\n");
k=0;
  for (i=i; i<17; i++)
    arr[k++]= div[i];
  while (i<33)
  {
    if (arr[0] == 0)
  xor (arr, 7e)
  else
  xor (arr, dd);
```

```
arr[16] = div[i++];
}
k=0;
for (i=17; i<33; i++)
div[i] = arr[k++];
  printf("codeword");
    for (i=0; i<33; i++)
      printf(i=0; i<33; i++)
        printf("%d", div[i]);
}
```

output:-

enter the data word.

10110011110010111

enter the divisor.

10001000001000 11

codeword : 10110011110010111000000000001101

At receiver end.

codeword : 10110011110010111000000000000000

WEEK 14

Write a program for congestion control using Leaky bucket algorithm.

CODE:

```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function int main()
{


int buckets, outlets, k = 1, num, remaining; printf("Enter Bucket size and
outstream size\n"); scanf("%d %d", &buckets, &outlets);
 remaining = buckets;


while (k) {


num = rand() % 1000; // Generate a random number between 0 and 999 if
(num < remaining)
 {


remaining = remaining - num;


printf("Packet of %d bytes accepted\n", num); // Added missing variable }


else {
```

```
printf("Packet of %d bytes is discarded\n", num); }

if (buckets - remaining > outlets) {

remaining += outlets; // Fixed the calculation }

else
 remaining = buckets;

printf("Remaining bytes: %d \n", remaining);
 printf("If you want to stop input, press 0, otherwise, press 1\n");
scanf("%d", &k);

}
 while (remaining < buckets) // Fixed the condition {

if (buckets - remaining > outlets) {

remaining += outlets; // Fixed the calculation }

else
 remaining = buckets;

printf("Remaining bytes: %d \n", remaining); }
```

return 0; // Added a return statement to indicate successful completion }

OUTPUT:

```
PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc bucket.c -o bucket } ; if ($?) { .\bucket }
Enter Bucket size and outstream size
2000
100
Packet of 41 bytes accepted
Remaining bytes: 2000
If you want to stop input, press 0, otherwise, press 1
1
Packet of 467 bytes accepted
Remaining bytes: 1633
If you want to stop input, press 0, otherwise, press 1
1
Packet of 334 bytes accepted
Remaining bytes: 1399
If you want to stop input, press 0, otherwise, press 1
1
Packet of 500 bytes accepted
Remaining bytes: 999
If you want to stop input, press 0, otherwise, press 1
1
Packet of 169 bytes accepted
Remaining bytes: 930
If you want to stop input, press 0, otherwise, press 1
1
Packet of 724 bytes accepted
Remaining bytes: 306
If you want to stop input, press 0, otherwise, press 1
1
Packet of 478 bytes is discarded
Remaining bytes: 406
If you want to stop input, press 0, otherwise, press 1
1
Packet of 358 bytes accepted
Remaining bytes: 148
If you want to stop input, press 0, otherwise, press 1
1
Packet of 962 bytes is discarded
Remaining bytes: 248
If you want to stop input, press 0, otherwise, press 1
0
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
```

```
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
PS D:\VS Code\OS>
```

OBSERVATION :

Week-14

Aim:- To write a program for congestion control using leaky bucket algorithm.

Code:-

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  int bucket, outlets, k=1, num, remaining;
printf("Enter Bucket size and outstream size\n");
scanf("%d %d", &buckets, &outlets);
remaining = buckets;
while(k)
{
  num = rand()%1000;
  if(num<remaining)
  {
  remaining = remaining - num;
printf("packet of %d bytes accepted\n", num);
}
else
  {
  printf("packet of %d bytes is discarded\n",
}
if(buckets - remaining > outlets)
{
  remaining += outlets;
} else
    remaining = buckets;
printf("Remaining bytes : %d \n", remaining
Printf("If you want to stop input, press 0, otherwise,
\n");
```

```
scanf("%d", &k);
}
while (remaining < buckets)
{
  if (buckets - remaining > outlets) .
   { remaining -= outlets ;
   }
   else ,
    remaining = buckets ;
   printf("Remaining bytes %d \n", remaining);
 }
 return 0;
}.
```

output:-

Enter Bucket size and out stream size :

1000 . 200 .

packets of 41 bytes are accepted .

Remaining bytes : 10000

If you want to stop input press 0 , otherwise press 1 .

packet of 467 bytes are accepted .

Remaining bytes 733 .

If you want to stop input press 0, otherwise press 1

1

packets of 334 bytes are accepted .

Remaining bytes : 599 .

WEEK 15

Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

CODE:

ClientTCP.py

```
 from socket import *
 serverName = “127.0.0.1”
 serverPort = 12000
 clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort)) sentence = input(“\nEnter file
name: “) clientSocket.send(sentence.encode())
 filecontents = clientSocket.recv(1024).decode() print (“\nFrom Server:\n”)
 print(filecontents)
 clientSocket.close()ServerTCP.py
 from socket import *
 serverName=”127.0.0.1”
 serverPort = 12000
 serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort)) serverSocket.listen(1)
 while 1:
 print (“The server is ready to receive”) connectionSocket, addr =
serverSocket.accept() sentence = connectionSocket.recv(1024).decode()
file=open(sentence,”r”)
 l=file.read(1024) connectionSocket.send(l.encode())
        print (“\nSent contents of “ + sentence) file.close()
 connectionSocket.close()
```

OUTPUT:

Observation:

Lab - 15

Date 24/08/2021

Aim:- Using TCP/IP sockets, write a client-server program to make client sending the file named and the server to send back the contents of the requested file if present.

code:-

clientTCP.py

```
from socket import*
server name = '127.0.0.1'
server port = 12000
client socket = socket (AF_INET, SOCK_STREAM)
client socket.connect ((server Name, server port))
sentence = input ("Enter file name:")
client socket.send (sentence.encode())
file contents = client socket.recv(1024).decode()
print ("In From server:\n")
print (file contents)
client socket.close()
```

serverTCP.py

```
from socket import*
servername = "127.0.0.1"
serverport = 12000
server socket = socket (AF_INET, SOCK_STREAM)
server socket.bind ((server Name, server port))
server socket.listen (1)
while 1:
print ("The Server is ready to receive")
connectionSocket, addr = server socket.accept()
sentence = connectionSocket.recv (1024).decode()
```

```
file = open(sentence,"r")
l = file.read(1024)
connection socket.send(l.encode())
print("in sent contents of", sentence)
file.close()
connection socket.close()
```

Output :-

After running.

Enter file name : server.py

from server :

```
from socket import *
serverName = "127.0.0.1"
server port = 12000
server socket = socket(AF_INET, SOCK_STREAM)
server socket bind((serverName, server port))
server socket.listen(1)
while 1:
    print("The server is ready to receive")
    connection socket, addr = serversocket.recv(1024).dec
    file = open(sentence,"r")
    l = file.read(1024)
    connection socket.send(l.encode())
    print("in sent contents of", sentence)
    file.close()
    connection socket.close()
```

WEEK 16

Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

CODE: ClientUDP.py

```
from socket import *
 serverName = “127.0.0.1”
 serverPort = 12000
 clientSocket = socket(AF_INET, SOCK_DGRAM)
 sentence = input(“\nEnter file name: “)
clientSocket.sendto(bytes(sentence,”utf-8”),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)


print (“\nReply from Server:\n”) print (filecontents.decode(“utf-8”)) # for i in
filecontents:
 # print(str(i), end = “ ”) clientSocket.close() clientSocket.close()
```

ServerUDP.py

```
from socket import *
 serverPort = 12000
 serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind((“127.0.0.1”, serverPort))
 print (“The server is ready to receive”)
 while 1:
 sentence, clientAddress = serverSocket.recvfrom(2048) sentence =
sentence.decode(“utf-8”) file=open(sentence,”r”)
```

con=file.read(2048) serverSocket.sendto(bytes(con,”utf-8”),clientAddress)
print (“\nSent contents of “, end = “ “)
print (sentence)
 # for i in sentence:
 # print (str(i), end = “ “)
file.close()

OUTPUT:

Observation:

Aim:

LAB-16

Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

Code:

```
from socket import *
serverName = "127.0.0.1"
serverport = 12000
clientsocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name: ")

clientsocket.sendto(bytes(sentence, "utf-8"), (serverName,
serverport))

filecontents, serverAddress = clientSocket.recvfrom(2048)
print("In Reply from server:\n")
print(filecontents.decode("utf-8"))

# for i in filecontents:
    # print(str(i), end="")

clientsocket.close()
clientsocket.close()


from socket import *

serverport = 12000
serversocket = socket(AF_INET, SOCK_DGRAM)
serversocket.bind(("127.0.0.1", serverPort))
Printf("The server is ready to recieve")
while 1:
```

```
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file = open(sentence,"r")
Con = file.read(2048)

serverSocket.sendto(bytes(Con,"utf-8"), clientAddress)
print("in send contents of ", end =")
print(sentence)
# for i in sentence:
    # print(str(i), end ="")
file.close()
```

output:

```
Enter file name: serverUDP.py
Reply from server:

The server is ready to receive
sent contents of serverUDP.py
The server is ready to receive
```

WEEK 17

Aim:Tool Exploration -Wireshark OBSERVATION:

Lab - 17

Aim:- Tool Exploration - wireshark.

wireshark is an open-source packet analyzer, which is used for education, analysis, software development communication protocol development and network troubleshooting. It is used to track packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer. It is also used by network security engineers to examine security problems.

wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as free packet sniffer computer application. It puts the network card into an unselective mode, i.e, to accept all packets which it receives use of wireshark.

1. It is used by network security ways: examine security problems.
2. It allows the users to watch all the traffic being passed over the network.
3. It is used by network engineers to troubleshoot network sheet issues.
4. It also helps to troubleshoot latency issues & malicious activities on your network.

5. It can also analyze dropped packets.

6. It helps us to known how all the device like lap top, mobile phone, desktop, switch, routers etc. communicate in a local network or the rest of the world.

Functionality of wire shark:

wireshark is similar to tcpdump in networking Tcpdump is a common packet analyzer which allows the user to display other packets and over a network attached to computer. It has a graphic end and some sorting & filtering functions wireshark users can see all traffic passing through network. wireshark can also monitor the unicast traffic which is not sent to network's MAC address. interface But the switch does not pass all traffic to the port. Hence, the promiscuous mode is not sufficient to see all the traffic, port mirroring is a method to monitor network traffic, when it is enabled, the switch send the copies of all network packets present at one port to another port Features of wireshark.

run on Linux, windows, OS X, free BSD, Net BSD, etc.