VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COURSE TITLE

Submitted by

AADISHWAR RAMESH (1BM21CS002)

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 "Computer Networks Lab" carried out by **AADISHWAR RAMESH (1BM21CS002),** 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 **Computer networks lab - (22CS4PCCON)** work prescribed for the said degree.

Dr. Nandhini Vineeth
Assistant professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

Sl. No.	Date	Experiment Title	Page No.
1.	15/6/23	Hub and Switch Experiment	1-11
2.	22/6/23	Static routing experiment	12-15
3.	14/6/23	Dynamic routing experiment	16-23
4.	14/6/23	DHCP LAN within and outside a network	24-28
5.	20/7/23	Domain Name System(NDS)	29-35
6.	20/7/23	Routing Information Protocol(RIP)	36-42
7.	27/8/23	OSPF Protocol	43-51
8.	3/8/23	ARP Protocol	52-55
9.	3/8/23	VLAN Experiment	56-61
10.	10/8/23	TTL/ Life of a packet	62-65
11.	10/8/23	TELNET experiment	66-71
12.	10/8/23	WLAN Experiment	72-76
13.	17/8/23	CRC implementation	77-82
14.	17/8/23	Leaky bucket problem	83-86
15.	24/8/23	TCP packet transfer,UDP packet transfer	87-96
16.	31/8/23	Wireshark Experiment	97-98

EXPERIMENT 1

AIM:

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:

PROCEDURE:

HUB:

① Select a generic hub and 4 generic computers (PC)

② Connect all the generic PC's to the hub using a copper Straight-Through wire and see if all the lights are green in colour which shows the link is up and working. The connections must be made in FastEthernet port.

③ Assign IP address for all the 4 generic PC's.
PC1 → FastEthernet → IP address → 192.168.0.100
and subnet mask appears on its own.
PC2 → IP address → 192.168.0.101
PC3 → IP address → 192.168.0.102.
PC4 → IP address → 192.168.0.103.

④ Now first using Simulation mode send a simple PDU from one PC to another. Then, using Real-time mode ~~pin~~ go to desktop → then to command prompt and ping another PC.

RESULT:

HUB

Packet Tracer PC Command Line 1.0
PC > ping 192.168.0.103

Pinging 192.168.0.103 with 32 bytes of data:

Reply from 192.168.0.103: bytes=32 time=0ms TTL=128
Reply from 192.168.0.103: bytes=32 time=0ms TTL=128
Reply from 192.168.0.103: bytes=32 time=0ms TTL=128

Bafna Gold
Date: Page:

Reply from 192.168.0.103 : bytes = 32 time = 0ms TTL = 128

Ping statistics for 192.168.0.103
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss);
Approximate round trip times in milli-seconds :
Minimum = 0ms, Maximum = 1ms, Average = 0ms

OBSERVATION :

1. The PC with IP address 192.168.0.100 sends a packet to the HUB.
2. The HUB broadcasts the packet to all the other PC's in the topology.
3. The PC with the IP address 192.168.0.103 accepts the packet whereas the other PC's reject the packet.
4. The acknowledgement packet goes from the PC 192.168.0.103 to the sender. The other devices ignore the packet.
5. This cycle con't repeats again everytime the sender sends a packet.

PROCEDURE:

SWITCH:

1. Select a generic switch and 4 generic computers. (PC).
2. Connect all the generic PC's to the switch using a copper Straight-through wire and see if all the lights are green in colour which shows the link is up and working. The connections must be made in Fast Ethernet port.
3. Assign IP address for all the 4 generic PC's
   PC 1 → Fast Ethernet → IP address → 192.168.0.100 and subnet mask appears on its own.
   PC 2 → IP address → 192.168.0.101
   PC 3 → IP address → 192.168.0.102
   PC 4 → IP address → 192.168.0.103
4. Now first using Simulation mode send a simple PDU from one PC to another. Then, using real-time mode go to desktop → then to command prompt and ping another PC.

RESULT:

SWITCH:

Packet Tracer PC Command Line 1.0
PC > ping 192.168.0.103

Pinging 192.168.0.103 with 32 bytes of data:

Reply from 192.168.0.103: bytes = 32 time = 0ms TTL = 128

Bafna Gold
Date: Page:

Reply from 192.168.0.103: bytes=32 time=0ms TTL=128
Reply from 192.168.0.103: bytes=32 time=0ms TTL=128
Reply from 192.168.0.103: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.0.103:
Packets: Sent=4, Received=4, Lost=0 (0% Loss),
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=0ms, Average=0ms.

OBSERVATION:

1. The packet transmission starts from the source device and reaches the switch.
2. Switch sends the packet to all devices connected to it.
3. The destinated device receives the packet and sends an acknowledgement back to switch stating it has received the packet.
4. Switch remembers the device sending the acknowledgement and only communicates with that device for further transmission.
5. Other devices do not receive the packet from next transmission.

15-06-23

AIM:

Create a topology involving multiple hubs and a switch connecting them to simulate a simple PDU

TOPOLOGY:

Bafna Gold
Date: Page:

PROCEDURE:

1. Select the end devices and connect them to two different separate hubs.
2. Connect the two hubs to a switch.
3. End devices are connected to the hub using copper straight through wire.
4. Hubs are connected to switch using copper cross-over wire.
5. Select the message. Select the source device connected to one hub and destination device connected to another hub.
6. Observe the packet transmission and acknowledgement procedure.

OBSERVATION:

1. The packet transmission starts from ~~same~~ source device which sends it to the hub connected to it.
2. This hub sends the packet to all the devices connected to it and to the switch.
3. Switch sends the packet to the other hub connected to it which in turns sends the packet to all end devices connected to it.
4. The destinated device sends back the acknowledgement that it received the message.
5. All other devices ignore the received packet.
6. The acknowledgement from second hub is sent back to the source device through switch in the same manner.

RESULT:

PC > ping 192.168.0.114

Reply from 192.168.0.114: bytes=32 time=1ms TTL=128
Reply from 192.168.0.114: bytes=32 time=1ms TTL=128
Reply from 192.168.0.114: bytes=32 time=1ms TTL=128
Reply from 192.168.0.114: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.0.114:
Packets: Sent=4, Received=4, Lost=0 (0% Loss),
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=10ms, Average=2ms.

22/6/23 10/10

Result:

Hub:

Cisco Packet Tracer Student - C:\Users\BMSCECSEIL74\Cisco Packet Tracer 6.2sv\saves\aadi_Hub.pkt
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.002 Hub0 PC0 ICMP
0.002 Hub0 PC1 ICMP
0.002 Hub0 PC2 ICMP
0.002 Hub0 PC3 ICMP
0.003 PC2 Hub0 ICMP
0.004 Hub0 PC4 ICMP
0.004 Hub0 PC0 ICMP
0.004 Hub0 PC1 ICMP
0.004 Hub0 PC3 ICMP
Reset Simulation
Constant Delay
Captured to: 313.618 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:07:42.411
Power Cycle Devices PLAY CONTROLS:
Event List
Simulation
Switches
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
Successful PC4 PC2 ICMP 0.000 N 0
Copper Straight-Through
PC-PT PC4
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
Fa0

Cisco Packet Tracer Student - C:\Users\BMSCECSEIL74\Cisco Packet Tracer 6.2sv\saves\aadi_Hub.pkt
Logical
[Root]
New Cluster
PC4
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.0.105
Pinging 192.168.0.105 with 32 bytes of data:
Reply from 192.168.0.105: bytes=32 time=0ms TTL=128
Reply from 192.168.0.105: bytes=32 time=0ms TTL=128
Reply from 192.168.0.105: bytes=32 time=1ms TTL=128
Reply from 192.168.0.105: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.105:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>
Time: 00:25:27
Power Cycle Devices Fast Forward Time
Realtime
Switches
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
Successful PC4 PC2 ICMP 0.000 N 0
Copper Straight-Through
PC-PT PC4
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
Fa0

Switch:

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa3/1
Fa0/1
Fa2/1
Fa1/1
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.000 -- PC0 ICMP
0.000 -- PC0 ARP
0.001 PC0 Switch0 ARP
0.002 Switch0 PC1 ARP
0.002 Switch0 PC2 ARP
0.002 Switch0 PC3 ARP
0.003 PC3 Switch0 ARP
0.004 Switch0 PC0 ARP
0.004 -- PC0 ICMP
Reset Simulation
Constant Delay
Captured to: 12.481 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
Time: 00:03:41.425
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Switches
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
Successful PC0 PC3 ICMP 0.000 N 0
Activate Windows
Go to PC settings to activate Windows.

Cisco Packet Tracer Student - C:\Users\BMSCECSEIL74\Cisco Packet Tracer 6.2sv\saves\Aadi_Switch.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
PC>ping 192.168.0.104
Pinging 192.168.0.104 with 32 bytes of data:
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.104:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 192.168.0.104
Pinging 192.168.0.104 with 32 bytes of data:
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.104:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>
Fa3/1
Fa0/1
Fa2/1
Fa1/1
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
Time: 00:03:21
Power Cycle Devices
Fast Forward Time
Realtime
Switches
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
-- PC0 PC3 ICMP 0.000 N 0
Activate Windows
Go to PC settings to activate Windows.

Hybrid:

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.002 Hub0 Switch0 ICMP
0.003 Switch0 Hub1 ICMP
0.004 Hub1 PC10 ICMP
0.004 Hub1 PC4 ICMP
0.004 Hub1 PC5 ICMP
0.004 Hub1 PC6 ICMP
0.004 Hub1 PC7 ICMP
0.004 Hub1 PC8 ICMP
0.004 Hub1 PC9 ICMP
Reset Simulation
Constant Delay
Captured to: 0.004 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
Switch0
Hub0
Hub1
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
PC-PT PC6
PC-PT PC7
PC-PT PC8
PC-PT PC9
PC-PT PC10
Time: 00:07:35.521
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Simulation
Connections
Copper Cross-Over
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
In Progress PC0 PC8 ICMP 0.000 N 0
Activate Windows
Go to PC settings to activate Windows.

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.0.114
Pinging 192.168.0.114 with 32 bytes of data:
Reply from 192.168.0.114: bytes=32 time=1ms TTL=128
Reply from 192.168.0.114: bytes=32 time=10ms TTL=128
Reply from 192.168.0.114: bytes=32 time=0ms TTL=128
Reply from 192.168.0.114: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.114:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 10ms, Average = 2ms
PC>
Time: 00:08:38
Power Cycle Devices Fast Forward Time
Realtime
Connections
Copper Cross-Over
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
Successful PC0 PC8 ICMP 0.000 N 0
Activate Windows
Go to PC settings to activate Windows.

EXPERIMENT 2

AIM:

Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

OBSERVATION:

PROCEDURE:

⇒ Take 2 PCs and place them as shown, assuming 2 different IP addresses (10.0.0.1 & 40.0.0.1) as they belong to 2 different networks.

- Place 2 routers belonging to these 2 networks (10.0.0.10 & 40.0.0.10) being their gateways. and place the 3rd router in between to connect the 2 networks.

- Set gateways for the 2 PCs and then going to the CLI interface for each routers, specify the IP route (set up) for file transfer using the commands:

```
Router > enable
Router # config terminal
Router (config) : interface <port>
Router (config-if) : ip address <ip> <subnet mask>
Router (config-if) : no shut;
```

Do this for all these routers.

- Then we go to the terminal of either PCs, and try to ping to the other one; The message fails to deliver due to not setting up the Network Static Routes & Next Hop.

- We again go to CLI of each router and setup the "next hop" using the command:

```
> ip route <network-id> <mask> <next-hop>
```

eg: > ip route 40.0.0.0 255.0.0.0 20.0.0.20
(for router 1)

- This is done so that the router recognizes which pathway to take when packet is received for a particular destination.

RESULT:

(i) > 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.
Reply from 10.0.0.10: Destination Host Unreachable

Ping Statistics:
Packets: sent = 4; Received = 0; Loss = 4 (100% Loss)

(ii) > 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 = 128~~
Reply from 40.0.0.1 : Bytes = 32 time = 2ms TTL = 128
Reply from 40.0.0.1 : Bytes = 32 time = 4ms TTL = 128
Reply from 40.0.0.1 : Bytes = 32 time = 3ms TTL = 128

Ping Statistics
Packets: sent = 4, Received = 1, Loss = 1 (25% Loss)
Approx time in ms:

minimum = 2ms Maximum = 4ms average = 3ms

Result:

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router-PT
Router-PT
PC-PT
PC-PT
PC0
PC1
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:
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.
Reply from 10.0.0.10: Destination host unreachable.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>
Realtime
Copper Cross-Over

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router-PT
Router-PT
PC-PT
PC-PT
PC0
PC1
Command Prompt
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
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=16ms TTL=123
Reply from 10.0.0.1: bytes=32 time=16ms TTL=123
Reply from 10.0.0.1: bytes=32 time=16ms TTL=123
Reply from 10.0.0.1: bytes=32 time=16ms TTL=123
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 16ms, Maximum = 16ms, Average = 16ms
PC>
Realtime
Copper Cross-Over

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router-PT
Router-PT
PC-PT
PC-PT
PC0
PC1
Command Prompt
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 30.0.0.10
Pinging 30.0.0.10 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 30.0.0.10:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
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=13ms TTL=125
Reply from 40.0.0.1: bytes=32 time=13ms TTL=123
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125
Reply from 40.0.0.1: bytes=32 time=12ms TTL=123
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 9ms, Maximum = 13ms, Average = 12ms
PC>
Realtime
Copper Cross-Over

EXPERIMENT 3

AIM:

Configure default route, static route to the Router.

OBSERVATION:

PROCEDURE:

→ Take 2 PC's and place them as shown, assuming 2 different IP addresses (10.0.0.1 & 40.0.0.1) as they belong to 2 different networks.

→ Take 3 generic routers and connect the PC's to different routers with copper cross over and connect both the routers to main router with serial cables.

→ Set the IP addresses of PC and gateways.

→ Set the gateway address in all the routers taking interface as fastethernet for the PC's and serial for routers.

→ Connect the PC's to the interfaces.

→ Config - steps -

```
> enable (router 1)
# config t
# interface fastethernet 0/0
# IP address 10.0.0.10 255.0.0.0
# no shut
# exit
# interface serial 2/0
# IP address 20.0.0.1 255.0.0.0
# no shut
# exit
```

Similarily for router - 0

```
> enable
# config t
# interface serial 2/0
# IP address 20.0.0.2 255.0.0.0
# no shut
# exit
# interface serial 3/0
# IP address 30.0.0.1 255.0.0.0
# no shut
# exit
```

For router - 2.

```
> enable
# config t
# interface fastethernet 0/0
# IP address 40.0.0.10 255.0.0.0
# no shut
# exit
# interface serial 2/0
# IP address 30.0.0.2 255.0.0.0
# no shut
# exit
```

We need to set IP routes for all routes via routers.

For router - 1 & router - 2, we do default routing and for router - 0, static routing is done.

For router-1

```
# config t
# ip route 0.0.0.0 0.0.0.0 20.0.0.2
# no shut
# exit
show ip route
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 20.0.0.0/8 is directly connected, Serial 2/0
S 0.0.0.0/0 [1/0] via 20.0.0.2
```

Similarly for router-2

```
# config t
# ip route 0.0.0.0 0.0.0.0 30.0.0.1
# exit
show ip route
```

For router-0 (static routing)

```
# config t
# ip route 10.0.0.0 255.0.0.0 20.0.0.0
# ip route 40.0.0.0 255.0.0.0 30.0.0.0
# exit
show ip route
S 10.0.0.0/8 [1/0] via 20.0.0.0
C 20.0.0.0/8 is directly connected, serial 2/0
C 30.0.0.0/8 is directly connected, serial 3/0
S 40.0.0.0/8 [1/0] via 30.0.0.0
```

→ Now, we ping 10.0.0.1 from the command prompt of 40.0.0.1.

RESULT:

> ping 40.0.0.1

pinging 40.0.0.1 with 32 bytes of data.

Reply from 40.0.0.1: Bytes = 32 time = 2ms TTL = 128
Reply from 40.0.0.1: Bytes = 32 time = 2ms TTL = 128
Reply from 40.0.0.1: Bytes = 32 time = 2ms TTL = 128
Reply from 40.0.0.1: Bytes = 32 time = 2ms TTL = 128.

Ping statistics.
Packets: Sent = 4, Received = 4, Loss = 0 (0% Loss)
Approx time in ms:

minimum = 2ms Maximum = 4ms Average = 3ms.

OBSERVATIONS:

If router has only one pathway to go it can be default routing to send packets if any destination to its adjacent neighbour. This was the case with router 0 and router 1 where in other router, we do usual static routing.

10/10

20/7/23

Result:

Router 0:

Router 1:

```
Press RETURN to get started!


%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up


Router>enable
Router#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

S    10.0.0.0/8 [1/0] via 20.0.0.1
C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
S    40.0.0.0/8 [1/0] via 30.0.0.2
Router#
```

Router 2:

Router2
Physical
Config
CLI
IOS Command Line Interface
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
Press RETURN to get started!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#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 30.0.0.1 to network 0.0.0.0
C 30.0.0.0/8 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1
Router#
Copy
Paste

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 40.0.0.1: bytes=32 time=7ms TTL=128
Reply from 40.0.0.1: bytes=32 time=6ms TTL=128
Reply from 40.0.0.1: bytes=32 time=5ms TTL=128
Reply from 40.0.0.1: bytes=32 time=6ms TTL=128
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 7ms, Average = 6ms
PC>

EXPERIMENT 4

AIM:

Configure DHCP within a LAN and outside LAN.

OBSERVATION:

PROCEDURE:

→ Select two or more PC and a server connecting to switch and another network with only end devices and switch.

→ Connect both switches to router.

→ Set IP address of server as 10.0.0.1

→ Now, go to services < select DHCP < save the current IP address 20.0.0.2.

→ Now, check the IP addresses of other devices in the network in the IP configuration in desktop.

→ Now in the CLI of router enable follow steps:

```
> enable
# config t
# interface fastethernet 4/0.
# IP address 10.0.0.20 255.0.0.0
# no shut
# exit
# interface fastethernet 0/0
# IP address 20.0.0.20 255.0.0.0
# no shut
# exit
```

→ Go to server < config < gateway 10.0.0.20

→ Now in router we need to set IP address of server.

config t
fastethernet 0/0
IP helper-address 10.0.0.1
no shut
exit

→ Now go to server < services < DHCP < add new IP address 20.0.0.2.

→ To check the connection, go to the IP configuration of PC outside the network and click on DHCP and IP gateway will be visible.

RESULT:

From server from PC2 to PC0 whose ip address is 10.0.0.2.

PC > ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 10.0.0.2: byte = 32 time = 6ms TTL = 125
Reply from 10.0.0.2: byte = 32 time = 2ms TTL = 125
Reply from 10.0.0.2: byte = 32 time = 12ms TTL = 125

ping statistics for 10.0.0.2:
Packets: Sent 4 Received = 3 Lost = 1
Approximate round trip time in milliseconds:
Minimum = 2ms, Maximum = 12ms, Average = 6ms.

OBSERVATION:

→ Each end device get IP address assigned from the server.

→ Message is pinged from a device in one network to another.

→ It can be seen that the end devices get the IP address from configured start IP address.

DHCP config pool:	Gateway	Start IP.
Server pool	20.0.0.10	20.0.0.1
Server pool	10.0.0.10	10.0.0.2

10/10

20/7/23

Result:

Logical
10.0.0.20
Fa4/0
Router-PT
Router0
Fa0/0
20.0.0.20
Fa4/1
Fa0/1
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
10.0.0.1
PC-PT
PC3
PC-PT
PC4
Realtime

PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
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>

EXPERIMENT 5

AIM:

Configure RIP routing Protocol in Routers

OBSERVATION:

AIM:

Configuring RIP routing protocol in system of 3 routers.

TOPOLOGY:

PROCEDURE:

1) Select two PCs and 3 routers, connect the PCs to two routers using copper cross over wires and connect the routers to another router with serial DCE with timer.

2) Set the IP address of both the PCs as 10.0.0.1 and 40.0.0.1 respectively and their gateways as 10.0.0.10 and 40.0.0.10 respectively.

3. Router 1 configuration

```
> enable
> config t
# interface Fastethernet 0/0
# ip address 10.0.0.10 255.0.0.0
# no shut
# exit.
# interface serial 2/0
# ip address 20.0.0.10 255.0.0.0
# encapsulation ppp
# clock rate 64000
# no shut
# exit.
```

Router 2 configuration.

```
> enable
> config t
# interface fastethernet 0/0
# ip address 40.0.0.10 255.0.0.0
# no shut
# exit
# interface serial 3/0
# ip address 30.0.0.20 255.0.0.0
# encapsulation ppp
# no shut
# exit.
```

Router 0 configuration

```
> enable
> config t
# interface serial 2/0
# ip address 20.0.0.20 255.0.0.0
```

#encapsulation ppp
#no shut
#exit
#interface serial 3/0
#ip address 30.0.0.10 255.0.0.0
#encapsulation ppp
#clockrate 64000
#no shut
#exit

4. Now, network router configuration for Routing Information Protocol (RIP) is done as follows

Router 1
#router-rip
#network 10.0.0.0
#network 20.0.0.0
#exit

Router 2
#router-rip
#network 30.0.0.0
#network 40.0.0.0
#exit

Router 0
#router-rip
#network 20.0.0.0
#network 30.0.0.0
#exit

5. After RIP configuration of all routers, we check the routing table of all by giving show ip route command.

For router 0

show ip rout
R 10.0.0.0/8 via 20.0.0.10, 00:00:13, serial
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks.
C 20.0.0.0/8 is directly connected, serial 2/0
C 30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks.
C 30.0.0.0/8 is directly connected, serial 3/0
C 30.0.0.0/32 is directly connected, serial 3/0
R 40.0.0.0/8 via 30.0.0.20, 00:00:12, serial 3/0

6. Now, ping 10.0.0.1 from the command prompt of 40.0.0.1 and vis versa.

RESULT:
from 40.0.0.1
>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Request timed out
Reply from 10.0.0.1 : bytes = 32, time = 12ms TTL = 125
Reply from 10.0.0.1 : bytes = 32 time = 12ms 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 = 3, Lost = 01.

Approximate round trip time in milli-seconds.
Minimum = 2ms, Maximum = 8ms, Average = 2ms.

from 10.0.0.1
>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time = 2ms 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 = 4, Lost = 0
Approximate round trip time in milli-seconds:
Minimum = 2ms, Maximum = 8ms, Average = 2ms.

10/10

8/23

Result:

Cisco Packet Tracer Student - C:\Users\aadis\Downloads\Aadi_Router_RIP.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Se2/0
20.0.0.10
Fa0/0
Router-PT
Router0
10.0.0.10
20.0.0.20
Se3/0
30.0.0.10
Router-PT
Router1
Se2/0
Fa0/0
Router-PT
Router2
30.0.0.20
40.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1

Cisco Packet Tracer Student - C:\Users\aadis\Downloads\Aadi_Router_RIP.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router0
Physical
Config
CLI
IOS Command Line Interface
Press RETURN to get started!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#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
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.20/32 is directly connected, Serial2/0
R 30.0.0.0/8 [120/1] via 20.0.0.20, 00:00:18, Serial2/0
R 40.0.0.0/8 [120/2] via 20.0.0.20, 00:00:18, Serial2/0
Router#
Copy
Paste

Cisco Packet Tracer Student - C:\Users\aadis\Downloads\Aadi_Router_RIP.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Se2/0
20.0.0.10
Fa0/0
Router-PT
Router0
10.0.0.10
20.0.0.20
Se3/0
30.0.0.10
Router-PT
Router1
Se2/0
Fa0/0
Router-PT
Router2
30.0.0.20
40.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1
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:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms 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 = 10ms, Average = 5ms
PC>

EXPERIMENT 6

AIM:

Configure OSPF routing protocol.

OBSERVATION:

27-07-23

AIM:

Configure OSPF routing protocol.

TOPOLOGY:

Area 1

Area 0

R1

R2

R3

20.0.0.1

20.0.0.2

30.0.0.2

30.0.0.1

Area 3

Area 2

PC

PC

10.0.0.10

DG → 10.0.0.1

40.0.0.10

DG → 40.0.0.1

PROCEDURE:

1. Configure your PC's with the IP address and Gateway as given in the topology.
2. Config your each of your routers with IP addresses given in the topology.
3. Encapsulation ppp and clock rate need to be set as set in RIP protocol.

4. Now, enable IP routing by configuring ospf routing protocol in all routers.

In Router R1

R1 (config) # router ospf 1
R1 (config-router) # router-id 1.1.1.1
R1 (config-router) # network 10.0.0.0 0.255.255.255 area 1
R1 (config-router) # network 20.0.0.0 0.255.255.255 area 1
R1 (config-router) # exit

In Router R2

R2 (config) # router ospf 1
R2 (config-router) # router-id 2.2.2.2
R2 (config-router) # network 20.0.0.0 0.255.255.255 area 1
R2 (config-router) # network 30.0.0.0 0.255.255.255 area 0
R2 (config-router) # network exit

In Router R3

R3 (config) # router ospf 1
R3 (config-router) # router ospf 1
R3 (config-router) # router-id 3.3.3.3
R3 (config-router) # network 30.0.0.0 0.255.255.255 area 0
R3 (config-router) # network 40.0.0.0 0.255.255.255 area 2
R3 (config-router) # ~~network~~ exit.

R1 (config) # interface serial 2/0

4. R1 (config-if) # interface loopback 0
R1 (config-if) # ip add 172.16.1.252 255.255.0.0
R1 (config-if) # no shutdown.
R2 (config) # interface ~~3/0~~ serial 3/0
R2 (config-if) # interface loopback 0
R2 (config-if) # ip add 172.16.1.253 255.255.0.0
R2 (config-if) # no shutdown.
R3 (config) # interface serial 2/0
R3 (config-if) # interface loopback 0
R3 (config-if) # ip add 172.16.1.254 255.255.0.0
R3 (config-if) # no shutdown.

5. Creating a virtual link.

In Router R1,
R1 (config) # router ospf 1
R1 (config-router) # area 1 virtual-link 2.2.2.2
R1 (config-router) #

In Router R2,

R2 (config) # router ospf 1
R2 (config-router) # area 1 virtual-link 1.1.1.1
R2 (config-router) # exit

OUTPUT:

Router 0.

C 10.0.0.0/8 is directly connected, FastEthernet 0/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial 2/0
C 20.0.0.0/8 is directly connected, Serial 2/0
O 30.0.0.0/8 [110/129] via 20.0.0.2, 00:03:30, Serial 2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:03:30, Serial 2/0
C 172.16.0.0/16 is directly connected, Loopback 0

PC 1 - 10.0.0.10

ping 40.0.0.10

pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes = 32 time = 8ms TTL = 125.
Reply from 40.0.0.10: bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.10: bytes = 32 time = 7ms TTL = 125
Reply from 40.0.0.10: bytes = 32 time = 13ms TTL = 125

Ping statistics for 40.0.0.10:
Packets: sent = 4, Received = 4, Lost = 0 (0% Loss)
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 13ms, Average = 7ms

PC2: 40.0.0.10

ping 10.0.0.10

pinging 10.0.0.10 with 32 bytes of data:

Reply from 10.0.0.10: bytes = 32 time = 9ms TTL = 125
Reply from 10.0.0.10: bytes = 32 time = 6ms TTL = 125
Reply from 10.0.0.10: bytes = 32 time = 16ms TTL = 125.
Reply from 10.0.0.10: bytes = 32 time = 2ms TTL = 125.

Ping statistics for 10.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss).
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 16ms, Average = 8ms

10/10

2/8/23

Result:

Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\OSPF.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
30.0.0.1
20.0.0.2 Se2/0
Se3/0
Area 0
Area 1
Router-PT
Router1
30.0.0.2
Se2/0
Se2/0
20.0.0.1
Fa0/0
Router-PT
Router0
10.0.0.1
Router-PT
Fa0/0
Router2
40.0.0.1
Area 2
Area 3
Fa0
Fa0
PC-PT
PC0
10.0.0.10
10.0.0.1
PC-PT
PC1
40.0.0.10
40.0.0.1

Router0
Physical
Config
CLI
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
00:00:16: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FULL, Loading Done
00:00:30: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on OSPF_VL0 from LOADING to FULL, Loading Done
Router>enable
Router#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
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.2/32 is directly connected, Serial2/0
O 30.0.0.0/8 [110/128] via 20.0.0.2, 00:00:41, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:41, Serial2/0
C 172.16.0.0/16 is directly connected, Loopback0
Router#
Copy
Paste

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 40.0.0.1: bytes=32 time=2ms TTL=253
Reply from 40.0.0.1: bytes=32 time=17ms TTL=253
Reply from 40.0.0.1: bytes=32 time=10ms TTL=253
Reply from 40.0.0.1: bytes=32 time=2ms TTL=253
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 = 17ms, Average = 7ms
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 time=20ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 8ms, Maximum = 20ms, Average = 13ms
PC>

EXPERIMENT 7

AIM:

Demonstrate the TTL/ Life of a Packet.

OBSERVATION:

10-8-22

AIM:

To demonstrate the TTL (time to live)/ Life of a Packet.

TOPOLOGY:

PROCEDURE:

1. Select 2 PC's and three routers and create the above topology and connect them using copper crossover cables. Connect all the routers using automatic cables.
Set the IP address and gateway for both PC's.

2. Configure routers
for Router 1
Router > enable
Router # config t
Router (config) # interface fastethernet 0/0

Bafna Gold
Date: Page:

Router (config-if) # ip address 10.0.0.10 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.10 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit

Router 2 :

Router > enable
Router # config t
Router (config) # interface serial 2/0
Router (config) # ip address 20.0.0.20 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # interface serial 3/0
~~30.0.0.0 255.0.0.0~~
Router (config-if) # ip address 30.0.0.10 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit

Router 3 :
Router > enable
Router # config t
Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.20 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 40.0.0.10 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit.
Router (config) # exit.

Router 1:

Router # config t.
Router (~~#~~ config) # ip route 30.0.0.0 255.0.0.0 20.0.0.20
Router (config) # ip route 40.0.0.0 255.0.0.0 20.0.0.20
Router (config) # exit

Similarly for Router 2:

Router # config t
Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.10
Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.20
Router (config) # exit.

Router 3:
Router # config t
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.10
Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.10

3. Select simulation mode, select simple PDU send a simple PDU from one PC to another.

4. Use capture button to capture every transfer.

5. Click on the PDU during every transfer to see the inbound and outbound PDU details. Observe that there is a difference of 1 in the TTL when it crosses every router.

OBSERVATION:
Time to live is the amount of time or hops that a packet is set to exist inside a network before being discarded by a router.

RESULT:
Select the PDU during every transfer to see ~~inbound and~~ inbound and outbound PDU details.

PDU information at Device : PC 0
Outbound PDU details
TTL : 255.
PDU information at Device : Router 0
Inbound PDU details
TTL : 255.
Outbound PDU details
TTL : 254.
PDU information at Device : Router 1
Inbound PDU details :
TTL : 254
Outbound PDU details :
TTL : 253
PDU information at Device : Router 2
Inbound PDU details :
TTL : 253
Outbound PDU details :
TTL : 252
PDU information at Device : PC 1
Inbound PDU details :
TTL : 252.

17/8/23
10/10

Result:

PC0:

Router 0:

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
20.0.0.10
30.0.0.20
10.0.0.10
Fa0/0
Router3
Router2
40.0.0.10
PC-PT
PC0
10.0.0.1
PC1
40.0.0.1
PDU Information at Device: Router3
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
IP
ICMP
Simulation Panel
Event List
Play Controls
Time: 00:16:35.374
Simulation

Router 1:

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
20.0.0.10
30.0.0.20
10.0.0.10
Fa0/0
Router3
Router2
40.0.0.10
PC-PT
PC0
10.0.0.1
PC1
40.0.0.1
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
IP
ICMP
Simulation Panel
Event List
Play Controls
Time: 00:16:45.339
Simulation

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router3
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
IP
ICMP
Simulation Panel
Event List
Play Controls
Simulation

Router 2:

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router3
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
IP
ICMP
Simulation Panel
Event List
Play Controls
Simulation

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
20.0.0.10
30.0.0.20
10.0.0.10
Se2/0
Fa0/0
Router-PT
Router3
Se2/0
Fa0/0
Router-PT
Router2
40.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1
PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0004.9A06.4985
SRC MAC: 0010.117A.166B
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x8
0x0
0x0
TTL: 128
PRO: 0x1
CHKSUM
SRC IP: 40.0.0.1
DST IP: 10.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x13
SEQ NUMBER: 18
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
0.000 -- PC0 ICMP
0.001 PC0 Router3 ICMP
0.002 Router3 Router1 ICMP
0.003 Router1 Router2 ICMP
0.004 Router2 PC1 ICMP
0.005 PC1 Router2 ICMP
0.006 Router2 Router1 ICMP
0.007 Router1 Router2 ICMP
Reset Simulation
Constant Delay
Captured to: 0.007 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:16:55.335
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
In Progress
PC0
PC1
ICMP
0.000
N
0
(edit)
(delete)

PC1:

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
20.0.0.10
30.0.0.20
10.0.0.10
Se2/0
Fa0/0
Router-PT
Router3
Se2/0
Fa0/0
Router-PT
Router2
40.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0004.9A06.4985
SRC MAC: 0010.117A.166B
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x9
0x0
0x0
TTL: 128
PRO: 0x1
CHKSUM
SRC IP: 40.0.0.1
DST IP: 10.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x14
SEQ NUMBER: 19
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
0.000 -- PC0 ICMP
0.001 PC0 Router3 ICMP
0.002 Router3 Router1 ICMP
0.003 Router1 Router2 ICMP
0.004 Router2 PC1 ICMP
0.005 PC1 Router2 ICMP
0.006 Router2 Router1 ICMP
Reset Simulation
Constant Delay
Captured to: 0.006 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:17:04.917
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
In Progress
PC0
PC1
ICMP
0.000
N
0
(edit)
(delete)

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
20.0.0.10
30.0.0.20
10.0.0.10
Se2/0
Fa0/0
Router-PT
Router3
Se2/0
Fa0/0
Router-PT
Router2
40.0.0.10
Fa0/1
PC-PT
PC0
10.0.0.1
Fa0/1
PC-PT
PC1
40.0.0.1
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
4
IHL
DSCP: 0x0
TL: 28
ID: 0x9
0x0
0x0
TTL: 127
PRO: 0x1
CHKSUM
SRC IP: 40.0.0.1
DST IP: 10.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x14
SEQ NUMBER: 19
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.001 PC0 Router3 ICMP
0.002 Router3 Router1 ICMP
0.003 Router1 Router2 ICMP
0.004 Router2 PC1 ICMP
0.005 PC1 Router2 ICMP
0.006 Router2 Router1 ICMP
0.007 Router1 Router2 ICMP
0.008 Router2 Router1 ICMP
0.009 Router1 Router2 ICMP
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:17:04.920
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
In Progress PC0 PC1 ICMP 0.000 N 0 (edit) (delete)

EXPERIMENT 8

AIM:

Configure Web Server, DNS within a LAN.

OBSERVATION:

Bafna Gold
Date: Page:

PROCEDURE:

1. Select a switch and connect it with a PC and a server to form a LAN.
2. Set PC's IP address by clicking on it and go to config tab, there in Fastethernet option set IP address as 10.0.0.1 and subnet mask.
3. Set server's IP address as 10.0.0.2 and subnet mask respectively.
4. 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.
5. Go to server → services → HTTP → index.html, there create a CV and click on save.
6. Go to ~~server → services → HTTP~~ PC → Desktop → webbrowser and type 10.0.0.2 you will the CV as content that is changed.
7. Next go to server → services → DNS and switch on services. Now add a Domain server name and type the IP address as 10.0.0.2. Press add and save it.
8. Again go back to PC → Desktop → webbrowser and type the given domain name, there we can see the CV which had been created earlier.

RESULT:

Web Browser				x
< >	URL	http://bmscse	Go	Stop.

CV

Aadishwar Ramesh

USN: 1BM21CS002

Languages: C/C#/Java

Image:

OBSERVATION:

The Domain name system can be used to address a webpage and in cisco packet tracer, it can be controlled and updated in the server controller. The contents and the name of webpage can also be changed as and when required and when this domain name is browsed in the PC. The contents of the webpage are displayed.

10/10

15/22

Result:

Cisco Packet Tracer Student - C:\Users\aadis\Downloads\Aadi_DNS.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
PC0
Physical
Config
Desktop
Custom Interface
Web Browser
URL http://aadi.com
Go
Stop
Aadishwar Ramesh
aadishwarr@gmail.com
CAREER OBJECT:
Want to be the part of a software developer organization and utilize my knowledge,Skills and expertise to contribute towards organization and professional brilliance and to learn more with the passage of time from all types of situations and circumstances.
EDUCATION AND QUALIIFICATIONS:
Qualification | Maximun Marks | Obtained Marks | Division | Institution | Bord/University
BE in progress | ----- | ----- | None | BMSCE | University
12th | 500 | 495 | A | DPS South | School
10th | 500 | 490 | A | PSBB | School
TECHNICAL EXPERIENCE:
Fa0
PC-PT
PC0
Fa0
Server-PT
Server0

EXPERIMENT 9

AIM:

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

OBSERVATION:

Bafna Gold

AIM:

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

TOPOLOGY:

PROCEDURE:

1. Create a topology consisting of 4 PC's, 1 server and a switch.
2. ~~Name~~ Give IP address to all of the PC's and server – Server → 10.0.0.1, PC0 → 10.0.0.2, PC1 → 10.0.0.3, PC2 → 10.0.0.4, PC3 → 10.0.0.5
3. Connect all the PC's and the server to a switch using a copper-straight through wire.
4. Select the inspect tool from the tool bar

and open the ARP tables of all the devices.

5. Then, ping the devices from the command prompt of other devices and click on capture in the simulation mode to know the packet routing.

6. With every ping, the arp tables of devices get filled with MAC address of the corresponding devices.

7. Even the switch learns about the MAC addresses of all devices during pinging processes.

8. Once you have pinged all the devices, you can check the arp table of each device in command prompt of PCs.

>arp -a

Internet address	Physical address	Type
10.0.0.1	0060.47a4.a043	Dynamic
10.0.0.2	00.60.47e5.1424	Dynamic
10.0.0.3	0005.5ea1.0b46	Dynamic

9. In the switch → CLI, you can check the MAC addresses of the devices as follows

Switch > show mac address-table.

VLAN	MAC address	Type	Ports
1	0005.5ea1.0b46	Dynamic	Fa 3/1
1	000c.8546.bace	Dynamic	Fa 3/1
1	0060.47a4.a032	Dynamic	Fa 0/1
1	0060.47e5.1424	Dynamic	Fa 1/1

OUTPUT:

1. Ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2 : bytes = 32 time = 0 ms TTL = 125
Reply from 10.0.0.2 : bytes = 32 time = 2ms TTL = 125
Reply from 10.0.0.2 : bytes = 32 time = 3ms TTL = 125
Reply from 10.0.0.2 : bytes = 32 time = 1ms TTL = 125.

Ping statistics for 10.0.0.2
Packets : Sent = 4, Received = 4, Lost = 0 (0% Loss)
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 1.

OBSERVATION:

Every time when a ping is done from a end device to another, it gets to know about the MAC address of the other ~~Pe~~ end device and stores it in the ARP table. The ARP tables are show table:

ARP table for Server 0

IP address	Hardware Address	Interface
10.0.0.2	0060.47e5.1424	FastEthernet0
10.0.0.3	0005.5ea1.0b46	FastEthernet0
10.0.0.4	00DC.8546.6aec	FastEthernet0

ARP table for PC0

IP address	Hardware Address	Interface
10.0.0.1	0060.47a4.a043	FastEthernet0
10.0.0.3	0005.5ea1.0b46	FastEthernet0
10.0.0.4	000c.8546.6aec	FastEthernet0

ARP table for PC1

IP address	Hardware Address	Interface
10.0.0.1	0060.47a4.a043	FastEthernet0
10.0.0.2	0060.47e5.1424	FastEthernet0

19/10

ARP table for PC2

IP address	Hardware Address	Interface
10.0.0.1	0060.47a4.a043	FastEthernet0
10.0.0.2	0060.47e5.1424	FastEthernet0

18/23

Result:

Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\ARP.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled B
Fa0/1
Fa3/1
Fa0
Server-PT
Server0
10.0.0.1
Fa0
PC-PT
PC0
10.0.0.2
Fa0
PC-PT
PC1
10.0.0.3
Fa0
PC-PT
PC2
10.0.0.4

Switch0
Physical
Config
CLI
IOS Command Line Interface
up
%LINK-5-CHANGED: Interface FastEthernet1/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet2/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet3/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, changed state to up
Switch>show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0003.e4e0.80e5 DYNAMIC Fa2/1
1 0007.ec98.34e8 DYNAMIC Fa3/1
1 000a.41db.4158 DYNAMIC Fa0/1
1 000b.be99.d829 DYNAMIC Fa1/1
Switch>
Copy
Paste

Server0
Physical
Config
Services
Desktop
Custom Interface
Command Prompt
X
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms 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 = 0ms, Maximum = 2ms, Average = 0ms
SERVER>10.0.0.4
Invalid Command.
SERVER>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
SERVER>arp -a
Internet Address Physical Address Type
10.0.0.2 000b.be99.d829 dynamic
10.0.0.3 0003.e4e0.80e5 dynamic
10.0.0.4 0007.ec98.34e8 dynamic
SERVER>

EXPERIMENT 10

AIM:

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

OBSERVATION:

10-8-23

AIM:

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

TOPOLOGY:

PROCEDURE:

1. Select a PC and a router drop them in the workspace and connect them using a copper wire cross over connection and set the IP address of the PC as 10.0.0.2. Select router CLI

Router>enable
Router #configt
Router (config) # hostname r1
r1 (config) # enable secret p1
r1 (config) # interface fastethernet 0/0
r1 (config-if) # ip address 10.0.0.1 255.0.0.0
r1 (config-if) # no shut
r1 (config-if) # line vty 0 5
r1 (config-line) # login
% login disabled on line 132, until 'password' is set
% login disabled on line 133, until 'password' is set.
% login disabled on line 134, until 'password' is set.

%login disabled on line 134, until 'password' is set
%login disabled on line 135, until 'password' is set.
r1 (config-line)# password pw
r1 (config-line)# exit
r1 (config) # exit
r1 # wr
Building configuration...
[OK]
r1 #

RESULT:
Select PC0 → Desktop → command prompt.
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 = 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
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
Approximate round trip times in milli-seconds
minimum = 0ms, maximum = 0ms, Average = 0ms
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 → statistic, GRPR-R
M → Mobile, B → BGP, D - EIGRP, EOL-

external, O - OSPF, IA - OSPF, inter area
N_1 - OSPF NSSA external type 1, N_2 = OSPF.
NSSA external type 2 E_1 - OSPF external type 1,
E_2 - OSPF external type 2, E - EGP
i - ISIS, L1 - IS-IS level-1, L_2 - IS-IS
level-2, ia - IS-IS inter area.
* - candidate default, U - per user static
route, o - ODR.
P - periodic download static route.

Gateway at last resort is not set
C 10.0.0.0/8 is directly connected,
Fast Ethernet 0/0
r1#

OBSERVATION:
Telnet stands for teletype Network, but it can also be used to establish a connection using telnet protocol. It is used for accessing remote computer over TCP/IP networks like the Internet.

10/10

17/8/23

Result:

Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\TELNET.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Fa0
Fa0/0
PC-PT
PC0
Router-PT
Router1

PC0
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=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=7ms 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 = 7ms, Average = 1ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
r1>enable
Password:
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#

EXPERIMENT 11

AIM:

To construct a VLAN and make the PC's communicate among a VLAN.

OBSERVATION:

3-5-23

AIM:

To construct a VLAN and make the PC's communicate among a VLAN.

TOPOLOGY:

PROCEDURE:

1) Set up the topology as shown above, use a 1841 router.

2) Add an extra port to the switch as its needed.

3) Use copper straight through wire. Set the

IP address and gateway.

4. In Switch → config → VLAN database, give any VLAN number, here 20, and VLAN name here → NEW VLAN

5. Select add. Select the interface here fastethernet 0/1 (nearest to the switch from router) & make it trunk (from Access)

6. Look into fastethernet 3/1 & Ethernet 6/1 and chant VLAN 1 to 20 : NEW VLAN.

7. In Router → Select VLAN DATABASE, enter the number and name of the VLAN created. In CLI of router.

~~fastethernet 0/~~
Router (vlan) #exit
APPLY completed.
Exiting ....
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 192.168.1.1 255.255.255.0
Router (config-if) # no shut
Router (config) # interface fastethernet 0/0.1
Router (config-subif) # encapsulation dot1q 20
Router (config-subif) # ip address 192.168.20.1 255.255.255.0
Router (config-subif) # no shut
Router (config-subif) # exit.

10/10
17/8/23

Bafna Gold
Date: Page:

RESULT: (in PC0)

PC > ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

Reply from 192.168.1.3: bytes = 32 time = 1ms TTL = 128
Reply from 192.168.1.3: bytes = 32 time = 0ms TTL = 128
Reply from 192.168.1.3: bytes = 32 time = 0ms TTL = 128
Reply from 192.168.1.3: bytes = 32 time = 0ms TTL = 128

Ping statistics for 192.168.1.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss)
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms.

PC > ping 192.168.20.3.

Pinging 192.168.20.3 with 32 bytes of data:

~~Reply from~~ Request timed out.
Reply from 192.168.20.3: bytes = 32 time = 1ms TTL = 127
Reply from 192.168.20.3: bytes = 32 time = 0ms 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
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

OBSERVATIONS:

1. VLAN → Virtual Local Area Network is any broadcast domain that is partitioned and isolated in a computed network at the data link layer.

2. It is a virtualized connection that connects multiple devices and network nodes from different LAN's into one logical network.

Result:

Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\VLAN.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/0
Router0
192.168.1.1
192.168.20.1
Fa3/1
Eth6/1
Fa0/1
Fa2/1
Fa0
PC-PT
PC0
192.168.1.2
Fa0
PC-PT
PC1
192.168.1.3
Fa0
PC-PT
PC2
192.168.20.2
Fa0
PC-PT
PC4
192.168.20.3

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.2
Pinging 192.168.20.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.20.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

PC4
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=1ms TTL=127
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

EXPERIMENT 12

AIM:

To construct a WLAN and make the nodes communicate wirelessly.

OBSERVATION:

10-1-23

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

TOPOLOGY:

PROCEDURE:

1) Select a switch, router, Access-point-PT, two PC's and a laptop and construct the above topology using copper straight through cables.

2. Configure the connected PC's and router as normally done.

3. Configure the connected PC and access point → port 1 → SSID as WLAN.

4. Select WEP and enter any 10 digit hex key → i.e 1234567890.

5. To configure PC0 and laptop with wireless standards. Switch off the device. Drag existing PT → Host - NM - LAN to the component listed in the LHS. Drag WMP300N wireless interface to the empty port. Switch on the device.

6. PC0 → config → new wireless interface would have been added. Configure SSID, WEP, WEP key, IP address and Gateway to the device.
SSID → WLAN, authentication → select → WEP enter the same key → 1234567890
IP address → 10.0.0.3 and
subnet mask will be 255.0.0.0.
to configure router.

Router > enable
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # no shut

RESULT:

In PC 0 (10.0.0.2)

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 = 22ms TTL = 128
Reply from 10.0.0.3: bytes = 32 time = 10ms TTL = 128
Reply from 10.0.0.3: bytes = 32 time = 4ms TTL = 128
Reply from 10.0.0.3: bytes = 32 time = 5ms TTL = 128

Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 22ms, Average = 10ms

OBSERVATION:

WLAN → Wireless Local Area Network is a group of wirelessly located computers or other devices that form a network based on radio transmissions rather than wire connections.

After the WLAN is setup, the lined connection appears in the topology from the access point.

10/10

17/8/23

Result:

Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\WLAN.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/0
Router0
Fa1/1
Fa2/1
Fa0/1
Switch0
Access Point0
Fa0
PC-PT
PC0
PC-PT
PC1
Laptop-PT
Laptop0

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
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=32ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=16ms TTL=128
Reply from 10.0.0.3: bytes=32 time=11ms 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 = 9ms, Maximum = 32ms, Average = 17ms
PC>

CYCLE 2

AIM:

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

PROGRAM:

```
CYCLE -2
12-8-23
Write a program for error detecting code
using CRC. CCITT (16-bits).

#include <stdio.h>
#include <string.h>
#define N strlen (divisor)
char data [30];
char rem [30];
char division [10];
int dlength, i, j;
void xor ()
    { for (j=1 ; j<N ; j++)
      rem [j] = ((rem[j] == division [j] ? '0':'1')
    }
void crc()
    { for (i=0 ; i<N ; i++)
        rem [i] = data[i];
  do {
        if (rem[0] = = '1') xor ();
      for (j = 0 ; j<N-1 ; j++)
        rem [j] = rem [j+1];
        rem [j] = data [i+1];
    }.
    while (i <= dlength + 16);
  }
void receiver ()
  { printf ("Enter the data being received : ");
    scanf ("%s", data);
    printf (" Data received : %s ", data);
    crc ();
    for (i=0, (i<N-1) && (rem[i] != '1', i++)
    { if (i<N-1) printf ("\n Error detection
                                  in data \n");
```

```
else printf("\n No error detected in data\n");
  }
}
int main() { int c=0;
  printf("\n Enter data to be transmitted:")
  scanf("%s", data);
  printf("\n Enter the divisor:");
  scanf("%s", divisor);
  dlength = strlen(data);
  for(i=dlength; i<dlength+16, i++)
  data[i] = '0';
  printf("\n data padded with 16 zeros:%s",
                                  data);
  crc();
printf("\n The remainder or CRC is:%s", rem);
  for(i=dlength+16; i<dlength+16, i++)
    data[i] = rem[i-dlength];
printf("\n Final data being sent:%d", data)
  receiver();
  return 0;
}
```

OUTPUT:

CRC-16?

1) Enter the data to be transmitted: 1001101
Enter the divisor: 1011
Data padded with 16 zeros: 1001101 0000000000000
The remainder or CRC is: 111
Final data being sent: 1001101 0000000000111
Enter the data being received: 1001101 0000000000111
Data received: 1001101 0000000000111
No error detected in data.

2. Enter the data to be transmitted: 1001101
Enter the divisor: 1011
Data padded with 3 zeros: 1001101000000000000
The remainder or CRC is: 101
Final data being sent: 1001101000000000000111
Enter the data being received: 1001101000000000000111
Data received: 1001101000000000001111
Error detected in data.

10/10

N 28/8/23

PROGRAM:

```
#include<stdio.h>
#include<string.h>
#define N strlen(divisor)
char data[28];
char rem[28];
char divisor[10];
int dlength,i,j;
void XOR(){
    for(j = 1;j < N; j++)
    rem[j] = (( rem[j] == divisor[j])?'0':'1');

}

void receiver(){

    printf("Enter the received data: ");
    scanf("%s", data);
    printf("\n\n");
    printf("Data received: %s", data);

    crc();

    for(i=0;(i<N-1) && (rem[i]!='1');i++);
        if(i<N-1)
            printf("\nError detected\n\n");
        else
            printf("\nNo error detected\n\n");
}

void crc(){

    for(i=0;i<N;i++)
        rem[i]=data[i];
    do{

        if(rem[0]=='1')
            XOR();

        for(j=0;j<N-1;j++)
            rem[j]=rem[j+1];
```

```
        rem[j]=data[i++];
    }
    while(i<=dlength+16);

}

int main()
{  int c=0;

    printf("\nEnter data to be transmitted: ");
    scanf("%s",data);
    printf("\n Enter the Divisor: ");
    scanf("%s",divisor);
    dlength=strlen(data);
    for(i=dlength;i<dlength+16;i++)
        data[i]='0';
    printf("\n");
    printf("\n Data padded with n-1 zeros : %s",data);
    printf("\n");
    crc();
    printf("\nCRC or Check value is : %s",rem);
    printf("\n rem strlen is : %d ", strlen(rem));
    for(i=dlength+13;i<dlength+16;i++)
      {    printf("\n %s",data);
           data[i]= rem[c++];
           }
    printf("\n");

    printf("\n Final data to be sent : %s",data);
    printf("\n\n");

    receiver();
        return 0;
}
```

Output:

```
Enter data to be transmitted: 1001101

 Enter the Divisor: 1011


 Data padded with n-1 zeros : 10011010000000000000000

CRC or Check value is : 111
 rem strlen is : 3
 10011010000000000000000
 10011010000000000000100
 10011010000000000000110

 Final data to be sent : 10011010000000000000111

Enter the received data: 10011010000000000000111


Data received: 10011010000000000000111
No error detected
```

```
Enter data to be transmitted: 1001101

 Enter the Divisor: 1011


 Data padded with n-1 zeros : 10011010000000000000000

CRC or Check value is : 111
 rem strlen is : 3
 10011010000000000000000
 10011010000000000000100
 10011010000000000000110

 Final data to be sent : 10011010000000000000111

Enter the received data: 10011010000000000001111


Data received: 10011010000000000001111
Error detected



...Program finished with exit code 0
Press ENTER to exit console.
```

CYCLE 2

AIM:

Write a program for congestion control using Leaky bucket algorithm.

PROGRAM:

12-8-23

Write a program for congesion control using Leaky bucket algorithm.

```
#include <stdio.h>
int main()
{ int in, out, bsize, n, available = 0;
  printf ("Enter the bucket size:");
  scanf ("%d", &bsize);
  printf ("Enter the outgoing rate:");
  scanf ("%d", &out);
  printf ("Enter the no. of inputs:");
  scanf ("%d", &n);
  while (n!=0)
   {
     printf("Enter the incoming packetsize:")
     scanf ("%d", &in);
     printf ("Incoming packetsize %d\n", in);
     if (in <= (bsize - available))
   { available += in;
     printf ("Bucket buffer size %d out of
              %d \n", available, bsize);
   }
  else
  { printf ("Dropped %d no. of packets \n",
                 in - (bsize - available));
  printf ("Bucket buffer size %d out of %d\n",
                 available, bsize);
     available = bsize;
   }.
   available = available - out;
  printf ("After outgoing %d packets left
        out of %d in buffer\n", available, bsize)
     n--;
}}
```

OUTPUT:

Enter the bucket size : 1000
Enter the outgoing rate : 200
Enter the no. of inputs : 6
Enter the incoming packet size : 200
Incoming packet size 200
Bucket buffer size 200 out of 1000
After outgoing 0 packets left out of 1000 in buffer
Enter the incoming packet size : 400
Incoming packet size 400
Bucket buffer size 400 out of 1000
After outgoing 200 packets left out of 1000 in buffer.
Enter the incoming packet size : 450
Incoming packet size 450 out of 1000
Bucket buffer size 650 ~~packets left out of 1000 in buffer~~
After going 450 packets left out of 100 in buffer.
Enter the incoming packet size : 500
Incoming packet size 500
Bucket buffer size 950 out of 1000
After outgoing 750 packets left out of 1000 in buffer
Enter the incoming packet size : 100
Bucket buffer size 850 out of 1000
After outgoing 650 packets left out of 1000 in buffer
Enter incoming packet size : 0
Bucket buffer size 650 out of 1000.
After outgoing 450 packets left out of 1000 in buffer.

10/10

28/8/23

PROGRAM:

```
#include<stdio.h>

int main(){
   int in, out, bsize, n, available = 0;
   printf("Enter the bucket size: ");
   scanf("%d", &bsize);
   printf("Enter the outgoing rate: ");
   scanf("%d", &out);
   printf("Enter the no of inputs: ");
   scanf("%d", &n);

   while (n != 0) {
      printf("Enter the incoming packet size : ");
      scanf("%d", &in);
      printf("Incoming packet size %d\n", in);
      if (in <= (bsize - available)){
         available += in;
         printf("Bucket buffer size %d out of %d\n", available, bsize);
      } else {
         printf("Dropped %d no of packets\n", in - (bsize - available));
         printf("Bucket buffer size %d out of %d\n", available, bsize);
         available = bsize;
      }
      available = available - out;
      printf("After outgoing %d packets left out of %d in buffer\n", available, bsize);
      n--;
   }
}
```

Output:

```
Enter the bucket size: 1000
Enter the outgoing rate: 200
Enter the no of inputs: 6
Enter the incoming packet size : 200
Incoming packet size 200
Bucket buffer size 200 out of 1000
After outgoing 0 packets left out of 1000 in buffer
Enter the incoming packet size : 200
Incoming packet size 200
Bucket buffer size 200 out of 1000
After outgoing 0 packets left out of 1000 in buffer
Enter the incoming packet size : 400
Incoming packet size 400
Bucket buffer size 400 out of 1000
After outgoing 200 packets left out of 1000 in buffer
Enter the incoming packet size : 450
Incoming packet size 450
Bucket buffer size 650 out of 1000
After outgoing 450 packets left out of 1000 in buffer
Enter the incoming packet size : 500
Incoming packet size 500
Bucket buffer size 950 out of 1000
After outgoing 750 packets left out of 1000 in buffer
Enter the incoming packet size : 100
Incoming packet size 100
Bucket buffer size 850 out of 1000
```

CYCLE 2

AIM:
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.

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.

PROGRAM:

Bafna Gold
Date: Page:

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

CODE:

1. ClientTCP.py.

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\n Enter file name:")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('\n From Server:\n')
print(filecontents)
clientSocket.close()
```

2. 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('\n Sent contents of' + sentence)
file.close()
connectionSocket.close()
```

OUTPUT:

* When you run ServerTCP.py.

The server is ready to receive

* When you run ClientTCP.py

Enter file name: ServerTCP.py
From Server:
(The files from ServerTCP.py will be ~~dis~~ copied and displayed here).

* In ServerTCP.py

The server is ready to receive
Sent contents of ServerTCP.py
The server is ready to receive.

10/10
28/8/23

Bafna Gold
Date: Page:

Aim:
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.

Code:

1. ClientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket (AF_INET, SOCK_DGRAM)

sentence =input ("\n Enter file name: ")
clientSocket . sendto (bytes(sentence, "utf-8"), (serverName, serverPort))

fileContents, serverAddress = clientSocket . recvfrom(2048)
print ('\n Reply from server: \n')
print ( fileContents. decode ("utf-8"))
#for i in fileContents:
  #print (str(i), end = '')
  clientSocket . close()
  clientSocket . close()
```

2. 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, clientAddresse = serverSocket.recvfrom(2048)
    sentence = sentence.decode ("utf-8")
    file = open (sentence, "r")
    con = file.read (2048)

    serverSocket.sendto (bytes(con, "utf-8"), clientAddress)

print ("\n Sent contents of ", end = ")
print (sentence)
#for i in sentence:
    # print (str(i), end = "")
file.close ()
```

OUTPUT:

* When you run ServerUDP.py

The server is ready to receive

* When you run ClientUDP.py

Enter file name: ServerUDP.py
Reply from server:
(The files from ServerUDP.py will be copied and displayed here)

10/10

28/8/23

In ServerUDP.py
The server is ready to receive

Sent contents of ServerUDP.py
The server is ready to receive

PROGRAM:

1.

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()
```

2.

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:

serverTCP.py - C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverTCP.p...
File Edit Format Run Options Window Help
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()
clientTCP.py - C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientTCP.py...
File Edit Format Run Options Window Help
from socket import*
serverName="127.0.0.1"
serverPort=12000
clientSocket=socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence=input("\nEnter the file name:")
clientSocket.send(sentence.encode())
filecontents=clientSocket.recv(1024).decode()
print("\nFrom server:\n")
print(filecontents)
clientSocket.close()

clientUDP.py - C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientUDP.p...
File Edit Format Run Options Window Help
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.recvform(2048)
print("\nReply from server:\n")
print(filecontents.decode("utf-8"))
clientSocket.close()
serverUDP.py - C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverUDP....
File Edit Format Run Options Window Help
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")
while1:
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)
file.close()
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C Mostly cloudy
ENG
10:30
24-08-2023

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 6
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serv
y
The server is ready to receive
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientTCP.p
y
Enter the file name:
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C Mostly cloudy
ENG
10:13
24-08-2023

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
line 6, in <module>
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName,serverPort))
NameError: name 'serverName' is not defined. Did you mean: 'servername'?
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientUD
y
Enter file nameclientTCP.py
Traceback (most recent call last):
File "C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientUDP.p
line 8, in <module>
filecontents,serverAddress=clientSocket.recvform(2048)
AttributeError: 'socket' object has no attribute 'recvform'. Did you mean: 're
from'?
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientUD
y
Enter file nameclientTCP.py
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientUD
y
Enter file nameclientTCP.py
Reply from server:
from socket import*
serverName="127.0.0.1"
serverPort=12000
clientSocket=socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence=input("\nEnter the file name:")
clientSocket.send(sentence.encode())
filecontents=clientSocket.recv(1024).decode()
print("\nFrom server:\n")
print(filecontents)
clientSocket.close()
>>>
Ln: 42
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverUDP.p
y
The server is ready to receive
Traceback (most recent call last):
File "C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverUDP.py",
line 12, in <module>
serverSocket.sendto(bytes(con,"utf-8").clientAddress)
AttributeError: 'bytes' object has no attribute 'clientAddress'
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverUDP.p
y
The server is ready to receive
Sent contents of clientTCP.py
Ln: 15 Col: 0
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C Mostly cloudy
ENG
10:33
24-08-2023

```
*IDLE Shell 3.10.0*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/server
y
The server is ready to receive
Traceback (most recent call last):
  File "C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverTCP
 line 10, in <module>
    sentence=connectionSocket.recv(1024).decoder()
AttributeError: 'bytes' object has no attribute 'decoder'. Did you mean: 'd
'?
>>>
========================================= RESTART: C:/Users/bmscecse/AppDat
al/Programs/Python/Python310/serverTCP.py =================================
===
The server is ready to receive

Sent contents ofserverTCP.py
The server is ready to receive
```

```
IDLE Shell 3.10.0
File  Edit  Shell  Debug  Options  Window  Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientTCP.p
y

Enter the file name:serverTCP.py

From server:

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()

>>>
```

Activate Windows
Go to Settings to activate Windows.

CYCLE 2

AIM:
Tool Exploration -Wireshark

OBSERVATION:

31-8-23

WIRESHARK

Wireshark is a powerful and widely used network protocol analyzer that provides a deep insight into the traffic flowing across computer networks. With its user-friendly interface and extensive feature set, Wireshark allows users to capture, analyse and interpret network packets in real-time. It supports various protocols across different network layer, making it a valuable tool for troubleshooting network issues, detecting security vulnerabilities, and optimizing network performance.

cmd:

>ipconfig.
Windows IP configuration.

Ethernet adapter Ethernet:

Connection-specific DNS suffix:
Link-Local IPv6 Address fe80::4586::26B3
:baba:4282%3
IPv4 Address ..... : 10.124.2.85
Subnet mask ..... : 255.255.0.0
Default Gateway ..... : 10.124.0.11

Some of the features of wireshark are:

- Data can be captured from the wire from a live network connection or read from a file of already captured packets.
- Live data can be read from a number of types of networks including Ethernet, IEEE 802.11, PPP and loopback.
- Data display can be refined using a display filter.
- The IP address of the device can be used in the filter to capture only the packets sent out and to that IP address.
- VOIP calls in the captured traffic can be detected. If encoded in a compatible encoding, the media flow can even be played.
- Raw USB traffic can be captured.
- Various settings, timers and filters can be set that ensure only triggered traffic appear.
- The information of the packets include ID number, time, source IP address, destination IP address, protocol name, length and other important information.

31/8/23