# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belgaum -590014, Karnataka.**

**LAB REPORT**
**on**

# COMPUTER NETWORKS

***Submitted by***

**ARAVIND ANAND (1BM21CS030)**

***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 **COMPUTER NETWORKS**" carried out by **ARAVIND ANAND (1BM21CS030),** who is a 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.

Name of the Lab-Incharge: Dr. Shyamala G
Designation: Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

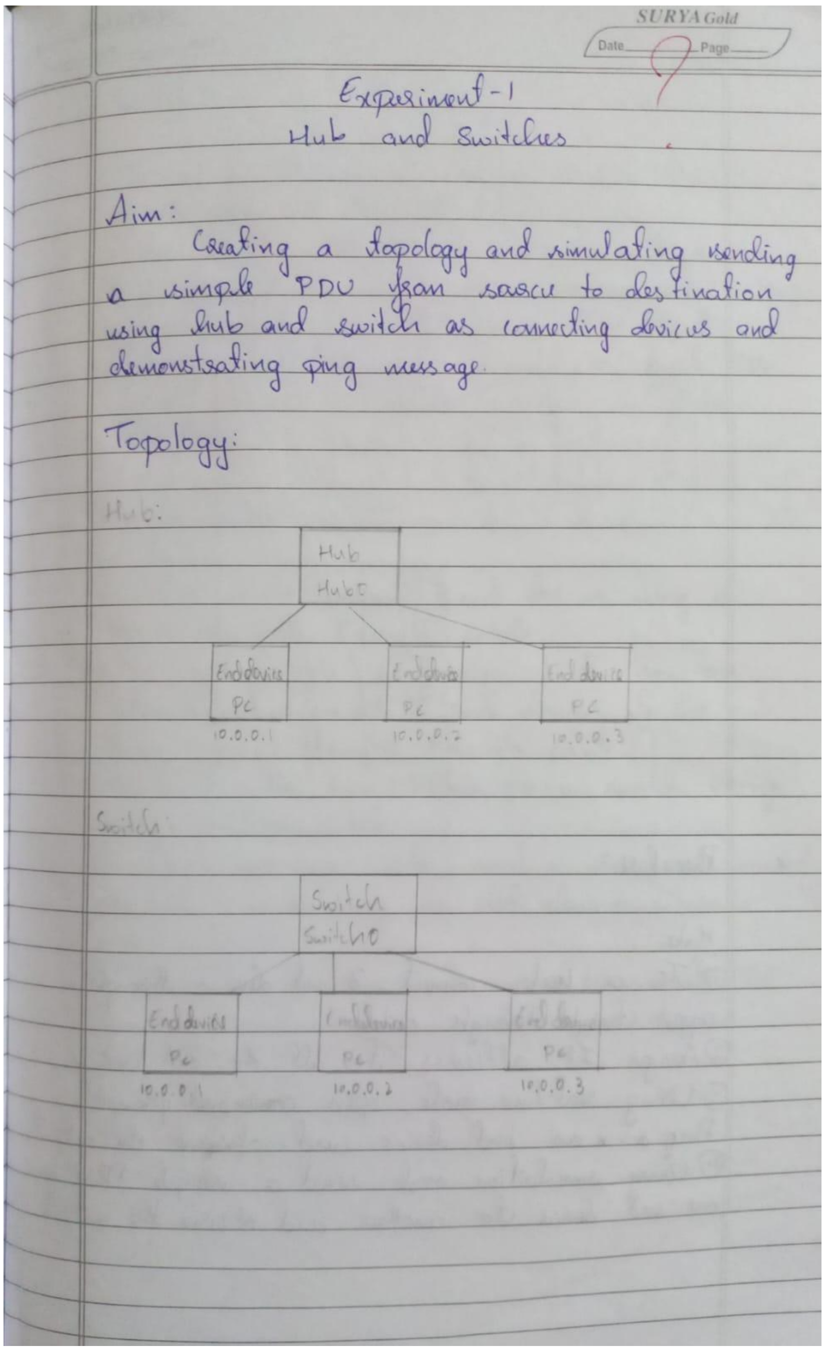
`

# Index

# EXPERIMENT-1

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:



SURYA Gold

Date Page

Experiment-1

Hub and Switches

Aim:

Creating a topology and simulating sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrating ping message.

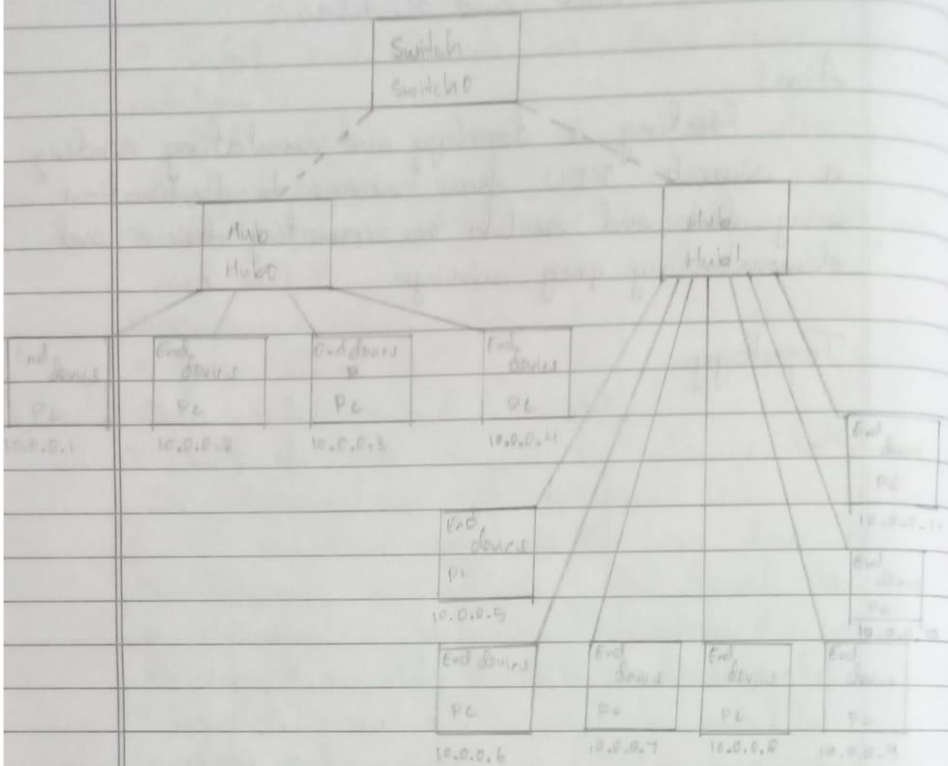
Topology:

Hub:



Switch:

Switch-hub




Procedure:

Hub:

① To a hub, connect 3 end devices through a copper straigh-through cable.
② Assign IP addresses to all the end devices.
③ Using real-time mode, open 'command prompt'. Ping a n an end device and observe the output.
④ Using simulation mode, send a simple PDU from one end device to another and observe the output.

Switch:

① To a switch, connect 3 end devices through a copper-straight through cable.
② Assign IP addresses to all the end devices.
③ Using real time mode, open 'command prompt'. Ping an end ~~user~~ device and observe the output.
④ Using simulation mode, ~~o~~ send a simple PDU from one end device to another and observe output.

Switch-hub:

① To a hub, connect 4 ~~3~~ end devices using a copper-straight through cable.
② To another hub, customize it to have an extra port. Connect 7 end devices to the hub using copper-straight through cable.
③ Connect the two hubs to a switch using copper crossover cable.
④ Using real-time mode, ~~send a~~ ~~si~~ open 'command prompt' and ping an end device. Observe ~~o~~ the output.
⑤ Using simulation mode, send a simple PDO from one end device to another and observe output.

Result:

Hub:

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

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

Switch:

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=5ms TTL=128

Ping statistics for 10.0.0.2:
Packets: Sent=4, Recieved=4, Lost=0 (0% Loss)

Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=5ms, Average=1ms.

SURYA Gold
Date______ Page______

Switch-hub:

Pinging 10.0.0.6 with 32 bytes of data:

Reply from 10.0.0.6: bytes=32 time=1ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128

Ping statistics from 10.0.0.6:
Packets: Sent=4, Received=4, Lost=0 (0% Loss)
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=1ms, Average=0ms

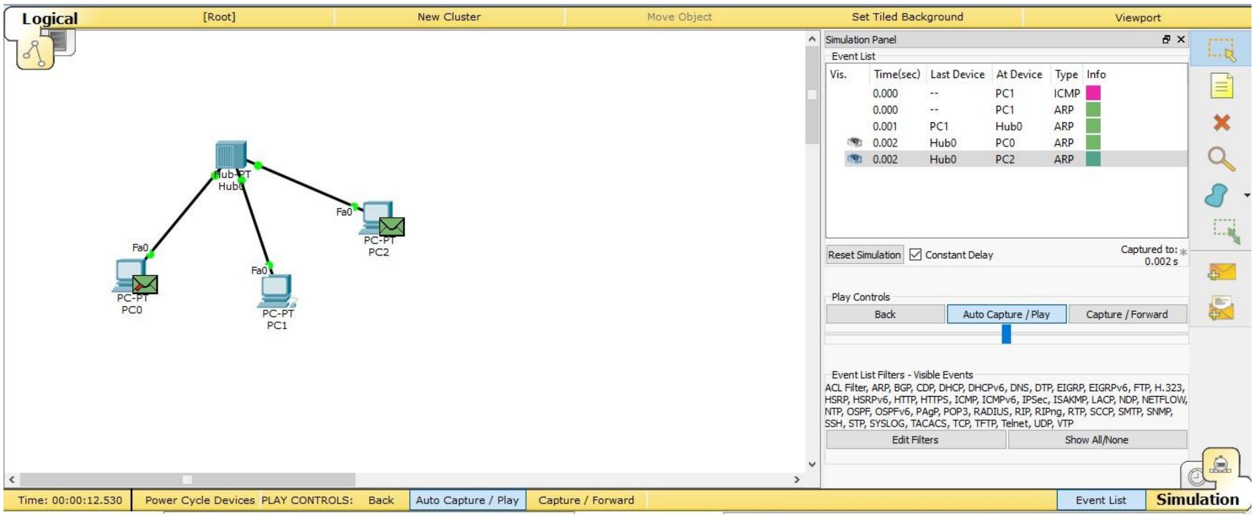
Observation:

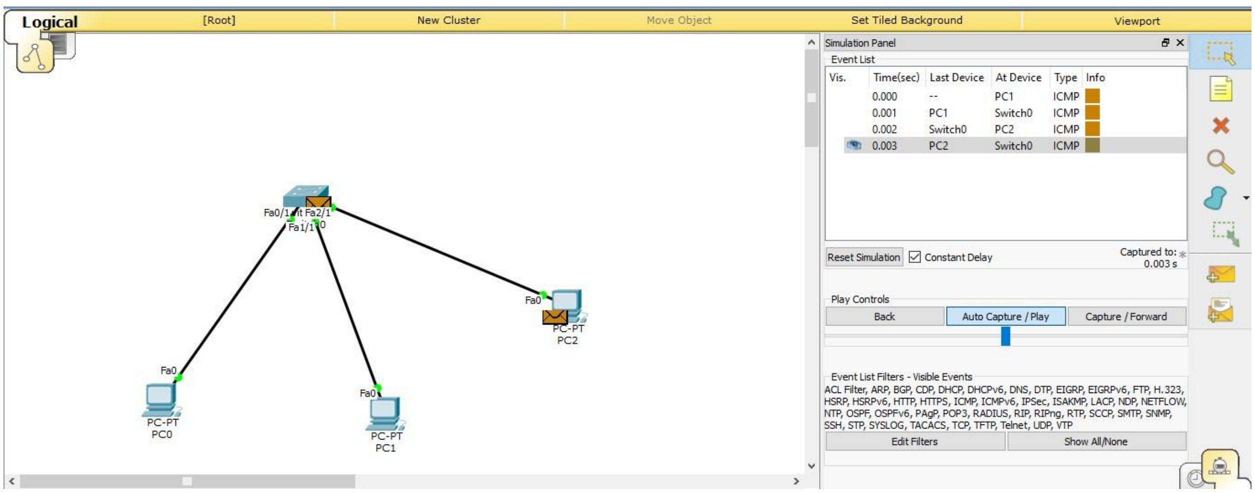
Hub:
Networking device which is used to transmit the signal to each port to respond from which the signal was recieved. It transmits signal to every port except the port from which the signal is recieved.

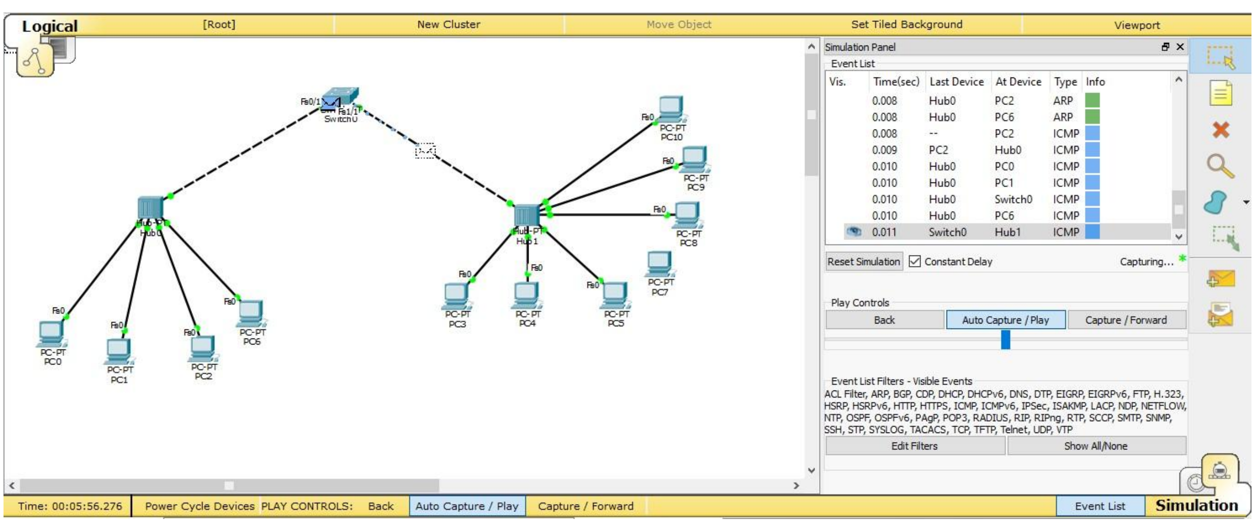
Switch:
Switch is an intelligent device which sends message to selected destination device only. First it examine the destination address and send the message to corresponding devices.

22/1/23

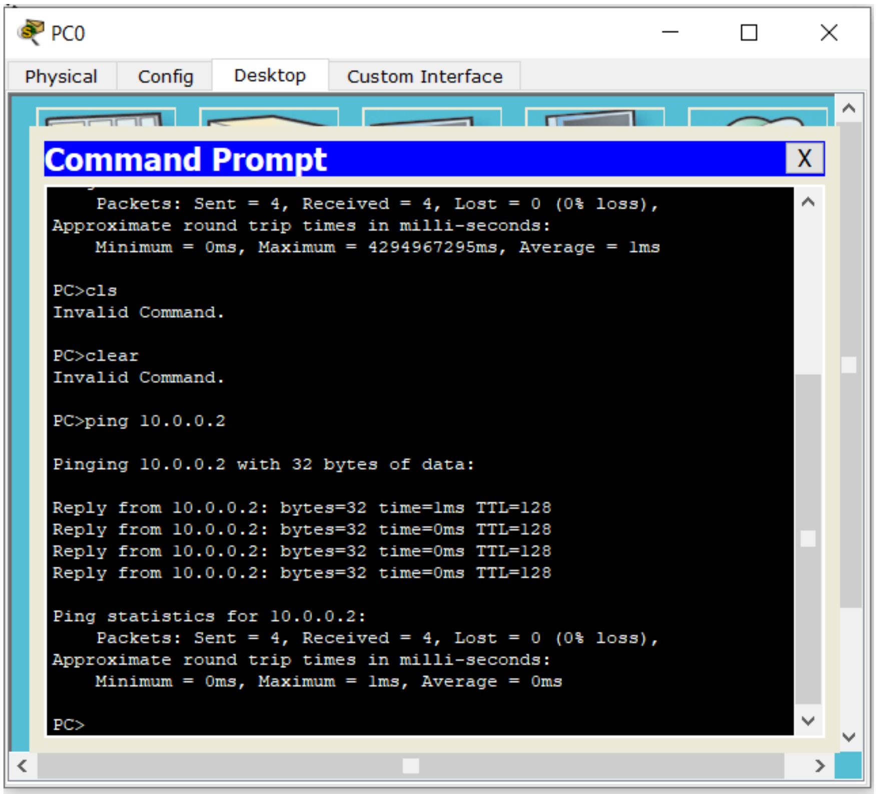
Topology:

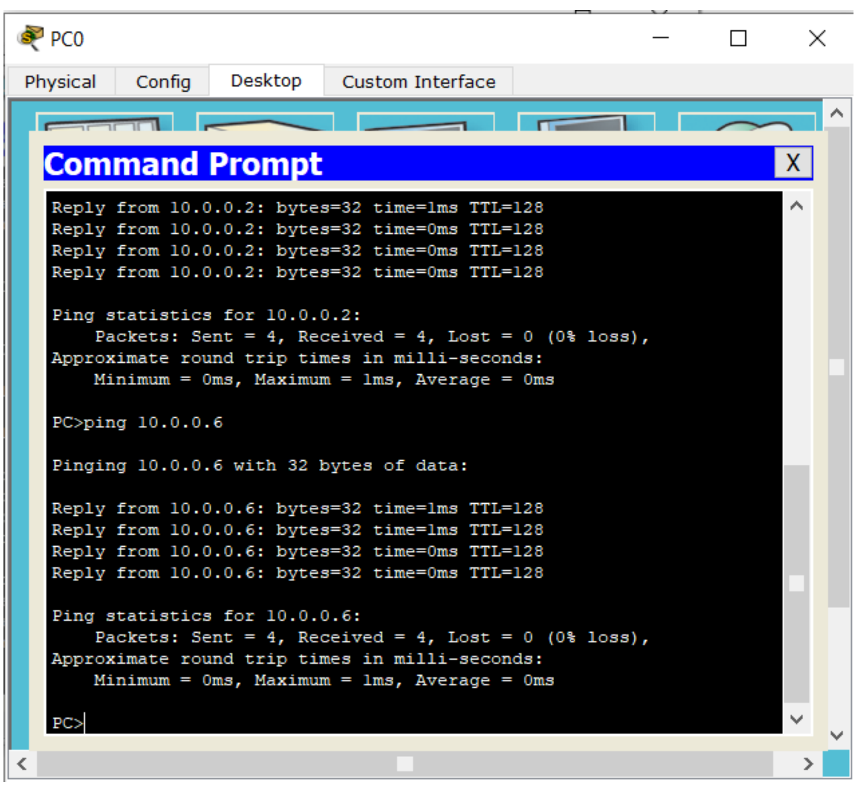

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Hub-PT
Hub0
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.000 -- PC1 ICMP
0.000 -- PC1 ARP
0.001 PC1 Hub0 ARP
0.002 Hub0 PC0 ARP
0.002 Hub0 PC2 ARP
Reset Simulation
Constant Delay
Captured to:
0.002 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: 00:00:12.530
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/1
Fa2/1
Fa1/1
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.000 -- PC1 ICMP
0.001 PC1 Switch0 ICMP
0.002 Switch0 PC2 ICMP
0.003 PC2 Switch0 ICMP
Reset Simulation
Constant Delay
Captured to:
0.003 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


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Switch0
Hub-PT
Hub0
Hub-PT
Hub1
PC0
PC1
PC2
PC6
PC3
PC4
PC5
PC7
PC8
PC9
PC10
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.008 Hub0 PC2 ARP
0.008 Hub0 PC6 ARP
0.008 -- PC2 ICMP
0.009 PC2 Hub0 ICMP
0.010 Hub0 PC0 ICMP
0.010 Hub0 PC1 ICMP
0.010 Hub0 Switch0 ICMP
0.010 Hub0 PC6 ICMP
0.011 Switch0 Hub1 ICMP
Reset Simulation
Constant Delay
Capturing...
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: 00:05:56.276
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation

# Result:

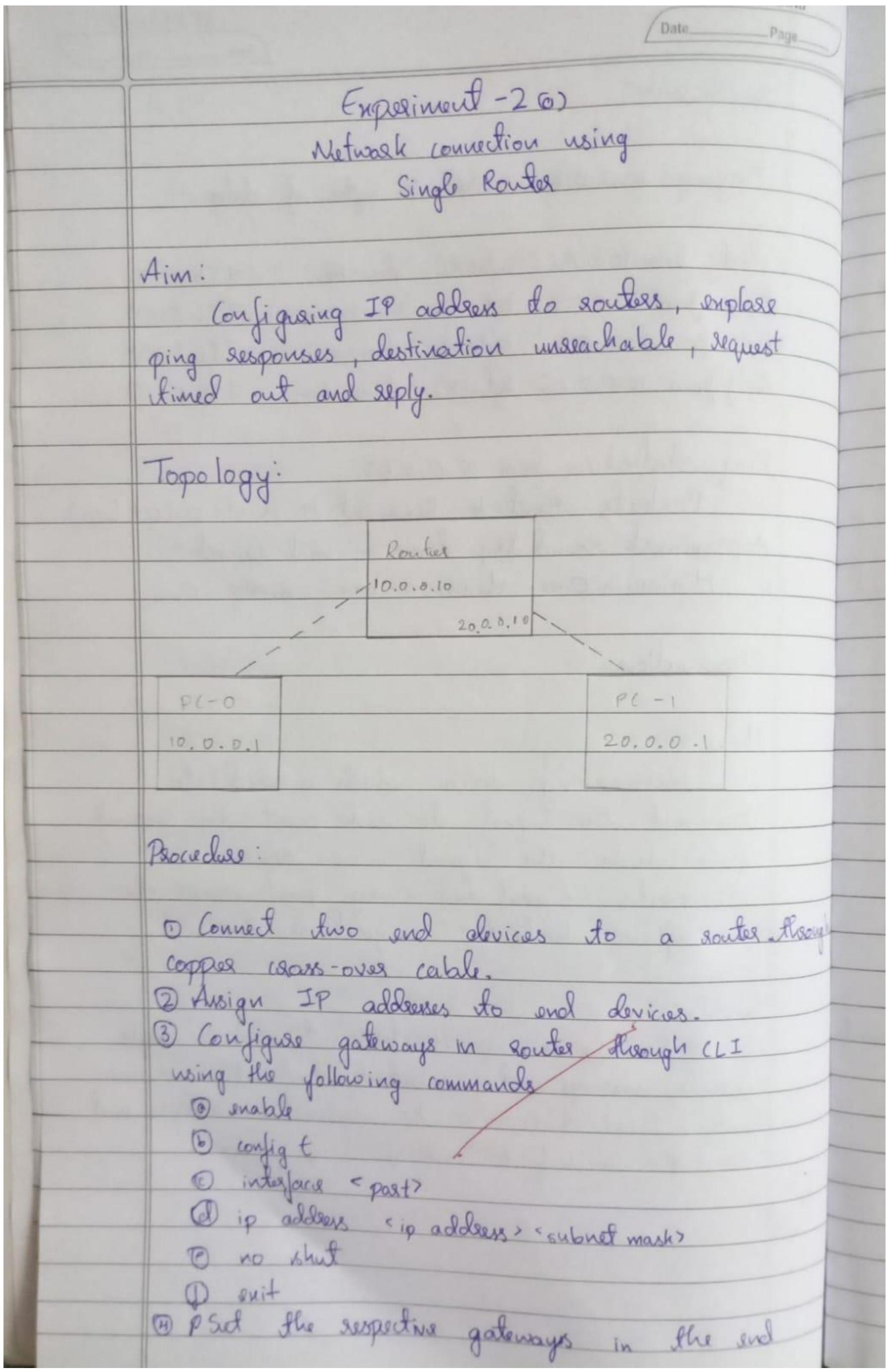

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4294967295ms, Average = 1ms
PC>cls
Invalid Command.
PC>clear
Invalid Command.
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
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 = 1ms, Average = 0ms
PC>


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
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 = 1ms, Average = 0ms
PC>ping 10.0.0.6
Pinging 10.0.0.6 with 32 bytes of data:
Reply from 10.0.0.6: bytes=32 time=1ms TTL=128
Reply from 10.0.0.6: bytes=32 time=1ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.6:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

# EXPERIMENT-2

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

Observation:



Date_____ Page_____

Experiment -2 (a)
Network connection using
Single Router

Aim:
Configuring IP address to routers, explore ping responses, destination unreachable, request timed out and reply.

Topology:

Router
10.0.0.10
20.0.0.10

PC-0
10.0.0.1

PC-1
20.0.0.1

Procedure:

① Connect two end devices to a router through copper cross-over cable.
② Assign IP addresses to end devices.
③ Configure gateways in router through CLI using the following commands
ⓐ enable
ⓑ config t
ⓒ interface <port>
ⓓ ip address <ip address> <subnet mask>
ⓔ no shut
ⓕ exit
④ Set the respective gateways in the end

devices.
⑤ Ping from one end user to another

Result:

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out
Reply from 20.0.0.1: bytes=32 time=1ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127

Ping statistics from 20.0.0.1:
Packets: Sent=4, Recieved=3, Lost=1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=1ms, Average=0ms

Observation:

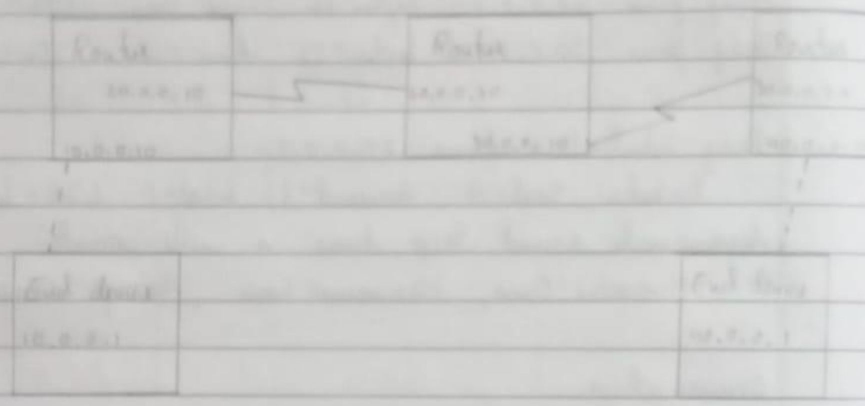
Router is a device used to connect multiple networks. Router is capable of transferring packets ~~from~~ from one network to another.
End device sends data packet to router. The destination IP address is noted by the router. The packet is redirected towards the concerned network by the router.

# Experiment - 2 (b)
## Network with multiple routers

**Aim:**

Configuring IP address of multiple routers, exploring ping responses, destination unreachable, request timed out and reply.

**Topology:**




**Procedure:**

① Add two end devices and three routers to work space.
② Connect routers to through serial DTE cable and end devices to routers through copper cross-over cable
③ Assign IP addresses to all devices and gateways.
④ Configure gateways through CLI using following commands.
   ⓐ enable
   ⓑ config t
   ⓒ interface <port>
   ⓓ ip address <ip address> <subnet mask>

(e) no shut

(f) exit

(5) Using command ip route <destination ip> <forwarding ip>, set path for each router.

(6) Ping from one end device to another.

Result:

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out
Reply from 40.0.0.1: bytes=32 time=12ms TTL=127
Reply from 40.0.0.1: bytes=32 time=12ms TTL=127
Reply from 40.0.0.1: bytes=32 time=14ms TTL=127.

Ping statistics from 40.0.0.1:
Packets: Sent=4, Received=3, Lost=1 (25% loss)
Approximate round trip times in milli-seconds
Minimum=12ms Maximum=14ms Average=12ms

Observation:

Destination host Unreachable:
For each router, we need to define a route for packets to be moved to different networks. Unless route is defined, packet will not reach destination. Following result is obtained when gateway isn't set

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

SURYA Gold

Date ______ Page ______

The above message signifies gateway 10.0.0.10 does not know where to redirect the packet to.

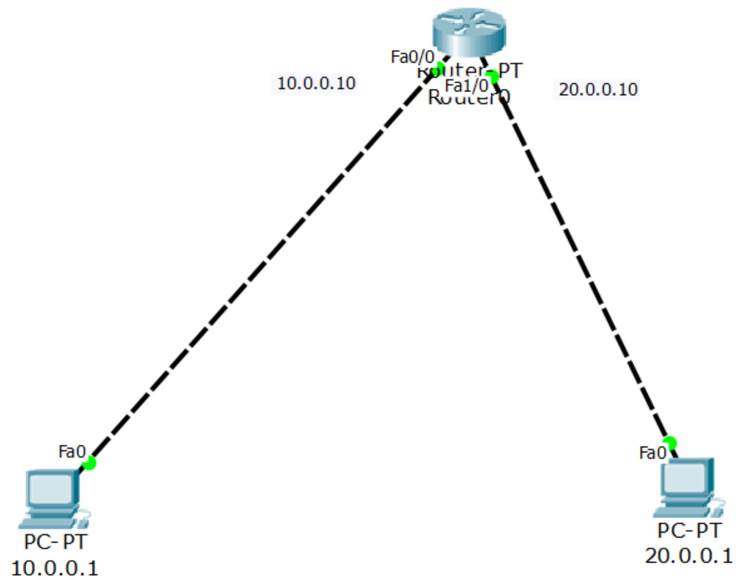
Request timed out:

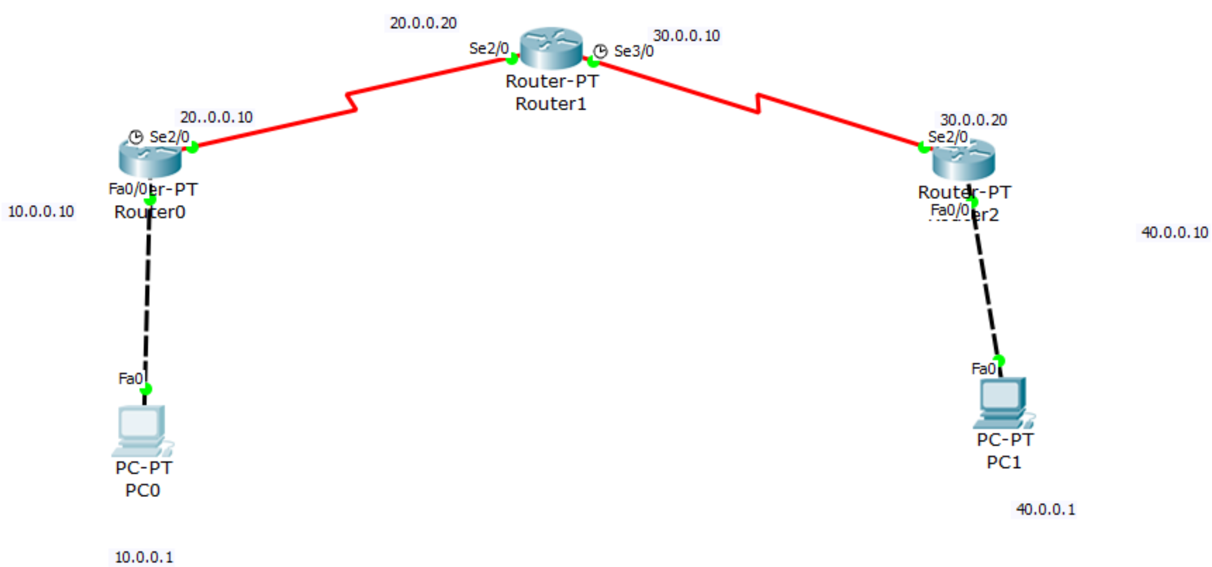
On successful transmission from source to destination, an acknowledgement is sent from destination host to source host in the form of ICMP packets. If the acknowledgement ICMP message does not reach source, a 'Request timed out' message is shown. It may be due to packet loss, physical issue in transmission or incorrect gateway assignment.

Successful Reply when Static Routing Command and

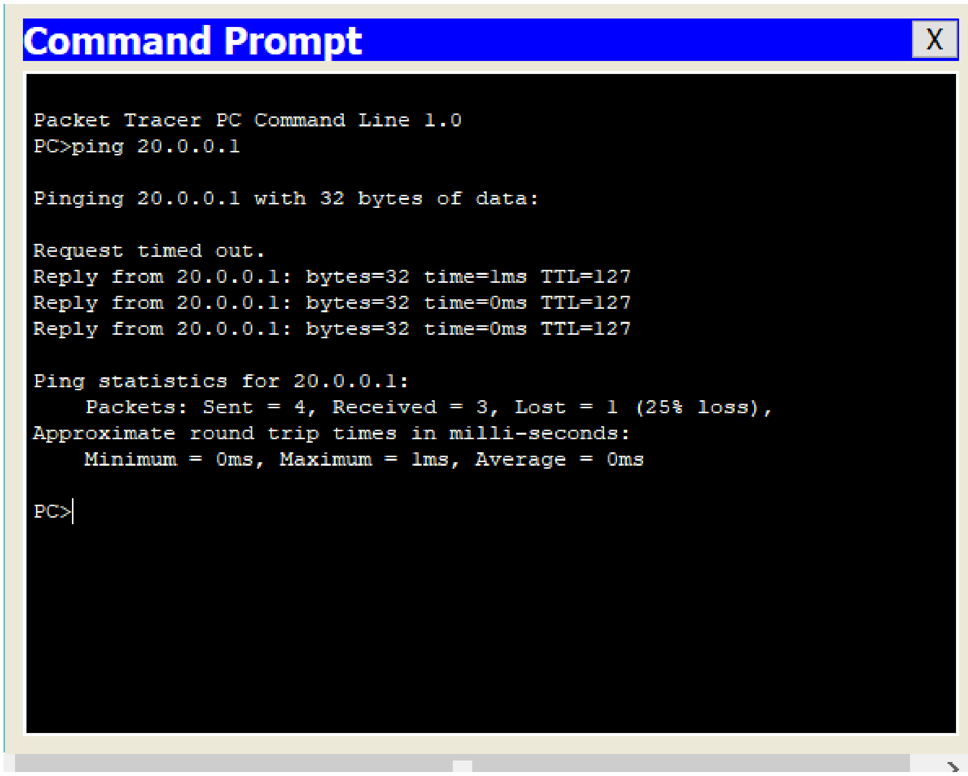
26/6/23

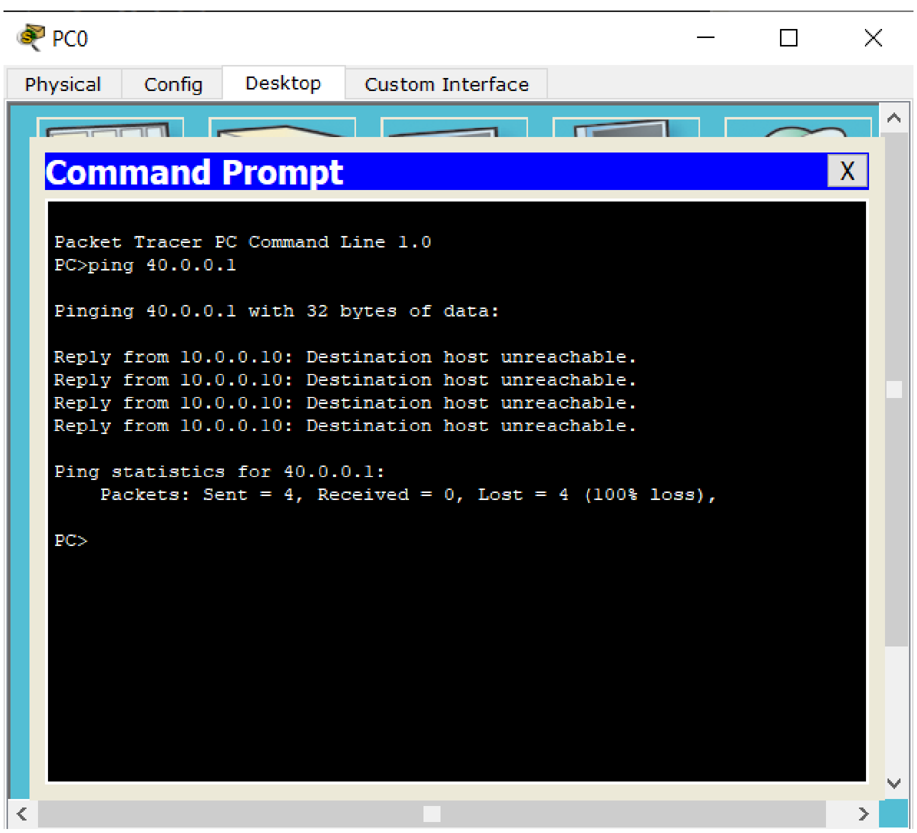
Topology:

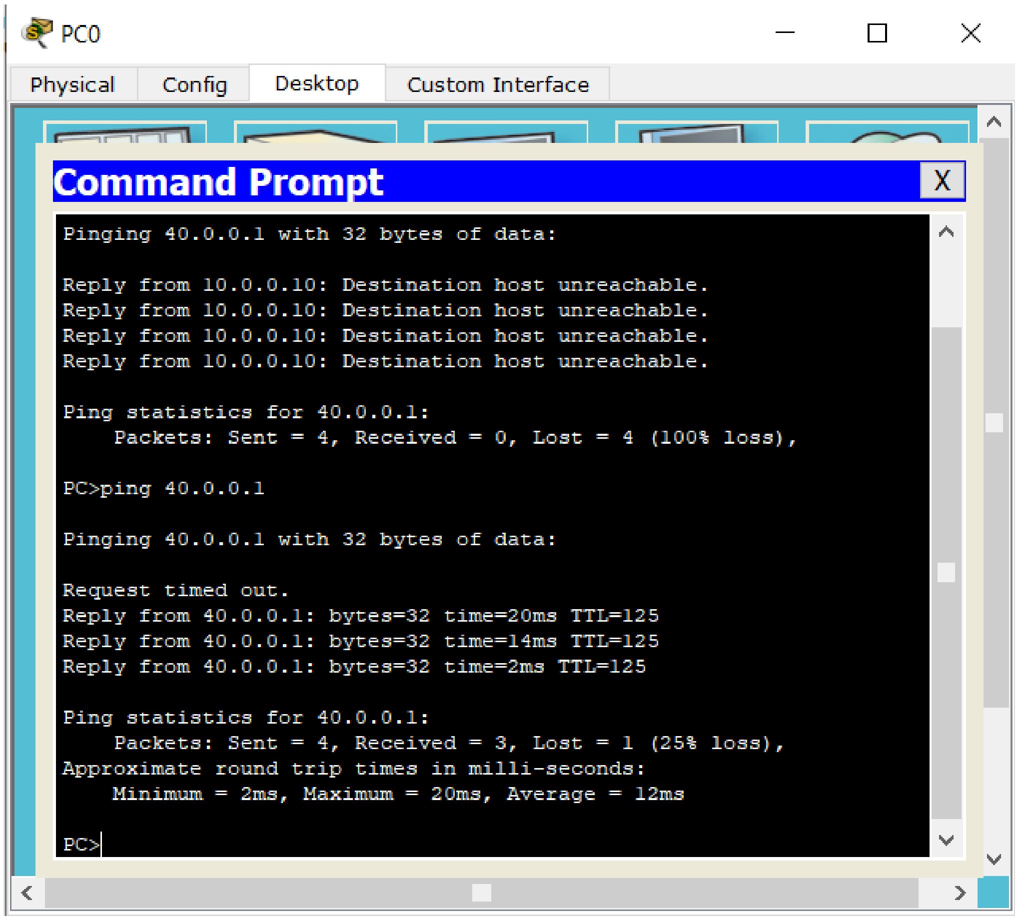

Fa0/0
Router-PT
Fa1/0
Router0
10.0.0.10
20.0.0.10
Fa0
PC-PT
10.0.0.1
Fa0
PC-PT
20.0.0.1


20.0.0.20
Se2/0
Se3/0
30.0.0.10
Router-PT
Router1
20..0.0.10
Se2/0
Fa0/0
Router-PT
Router0
10.0.0.10
30.0.0.20
Se2/0
Router-PT
Fa0/0
Router2
40.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1

Result:


Command Prompt
X
Packet Tracer 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=1ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>


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.
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>


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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>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=20ms TTL=125
Reply from 40.0.0.1: bytes=32 time=14ms 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 = 20ms, Average = 12ms
PC>

# EXPERIMENT-3

Configure default route, static route to the Router.

13/7/23

SURYA Gold

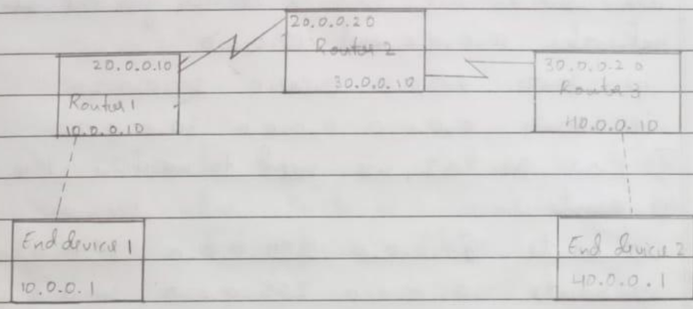
Date______ Page______

## Experiment-3

**Aim:**

Configure default route and static route to the router.

**Topology:**




**Procedure:**

1. Add two end devices and three routers to the workspace.
2. Connect routers through Serial DTE cable and end devices and routers through copper crossover cable.
3. Configure IP addresses of end devices. ~~and~~
   Ip address of End device 1 : 10.0.0.1
   Ip address of End device 2 : 40.0.0.1
4. Configure routers. Use the following commands to set IP address to routers.

```
Router > enable
Router # config t
Router (config) # interface <port>
Router (config-if) # ip address <ip address> <subnet mask>
```

Router(config-if)# no shut
Router (config-if)# exit

5) Set gateways for end devices
End device 1 : 10.0.0.10
End device 2 : 40.0.0.10

6) Default routing is possible for networks 10.0.0.0 and 40.0.0.0. Following commands are used to set default routes for the ab networks 10.0.0.0 and 40.0.0.0

ip route 0.0.0.0 0.0.0.0 20.0.0.20
ip route 0.0.0.0 0.0.0.0 30.0.0.20

7) For router2, we need to mention the ip route.

ip route 40.0.0.0 255.0.0.0 30.0.0.10
ip route 10.0.0.0 255.0.0.0 20.0.0.10

8) Ping from one end device to another.

Result:

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=15ms TTL=125
Reply from 40.0.0.1: bytes=32 time=11ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.1:
Packets: Sent=4, Recieved=3, Lost=1 (25% loss)

Approximate round trip times in milli-seconds:
Minimum=2ms, Maximum=15ms, Average=9ms

IP route of Router 1

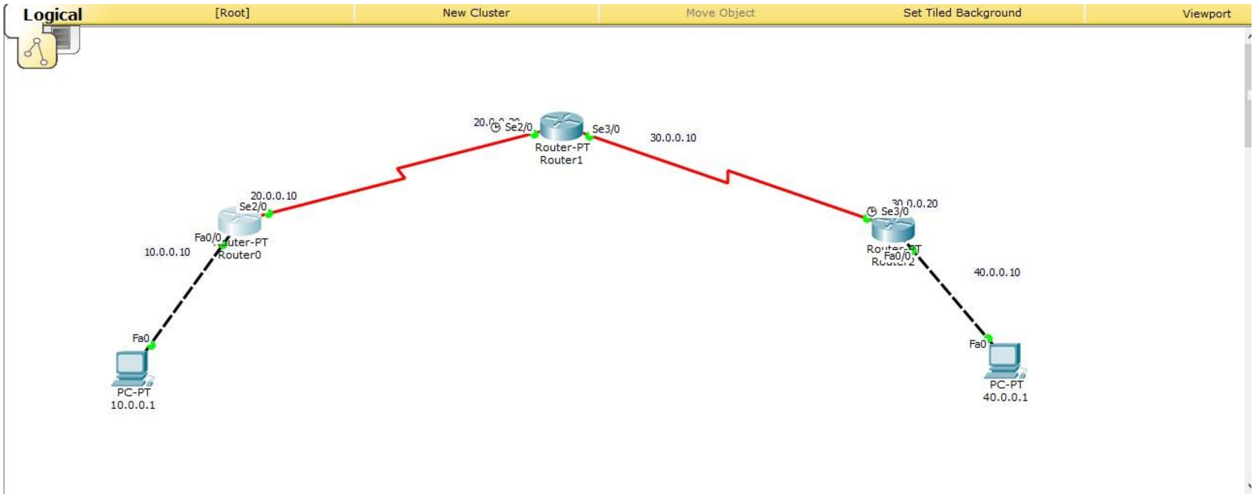
Gateway of last resort is 20.0.0.10 to network 0.0.0.0

C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 20.0.0.0/8 is directly connected, Serial2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.10
[1/0] via 20.0.0.20

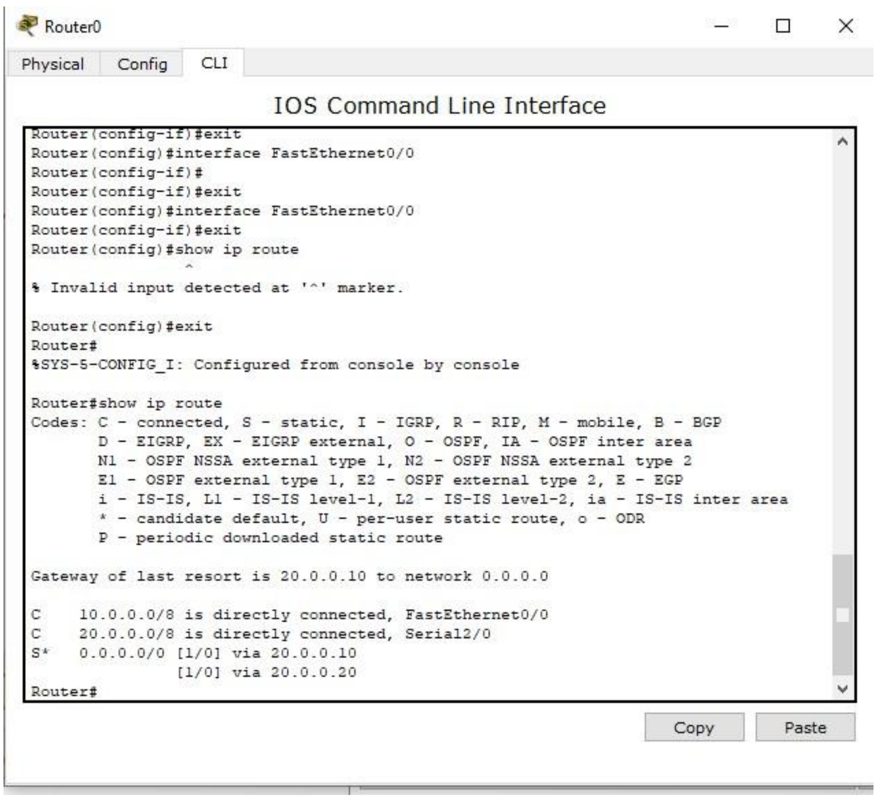
Observation:

Default route refers to the route taken when no other route is available for the given IP destination address.
When a packet is recieved, it is decapsulated and destination IP address of packet is checked. If there is no registered path to the destination IP address, the packet is transmitted onto the default route.
Static routing refers to specifying a route that for a given destination IP address.
The destination IP address of a packet is checked. For a given destination IP address, the next hop is stored in the routing table.

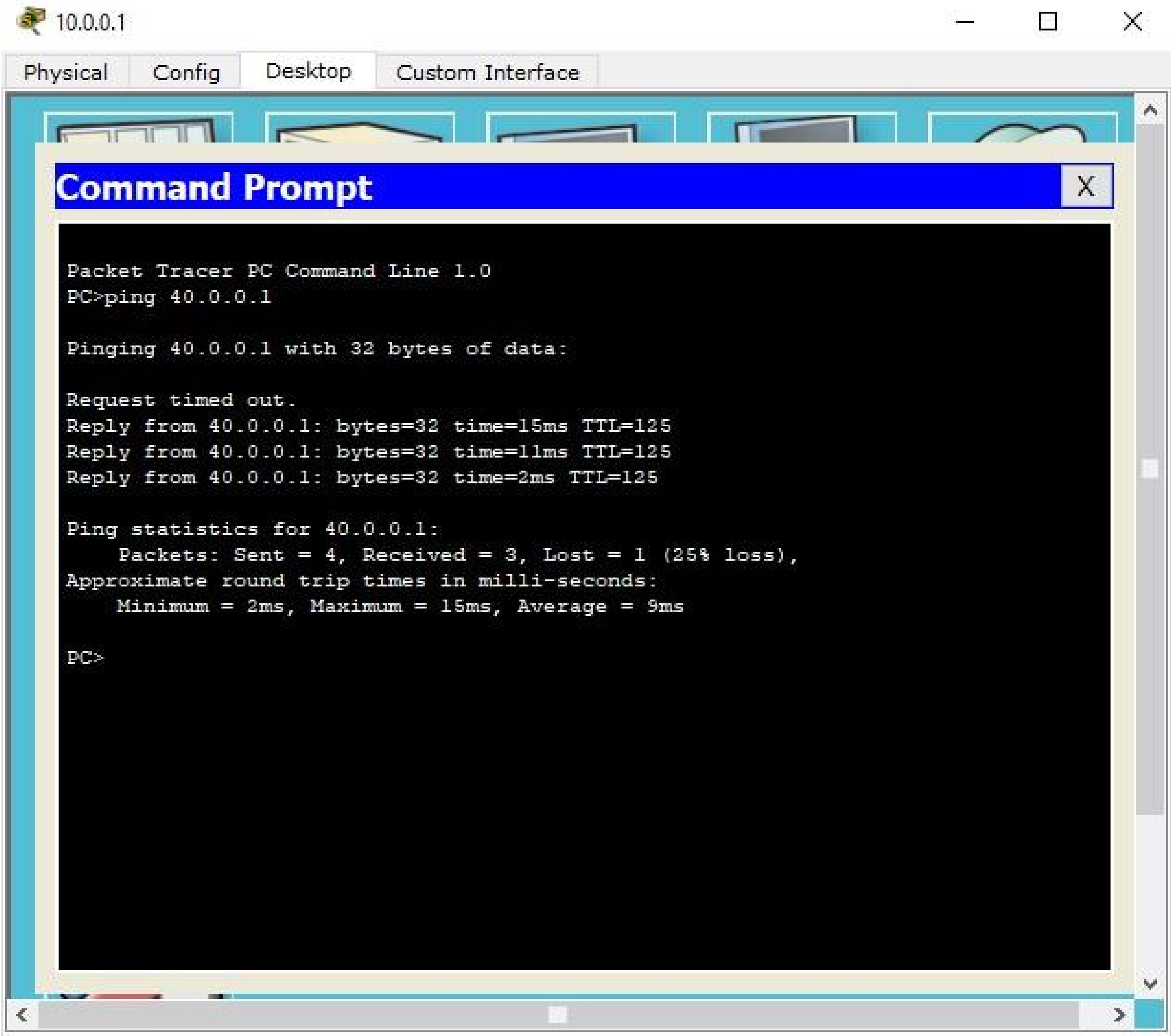
Topology:




Result:




```
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#exit
Router(config)#show ip route
                ^
% Invalid input detected at '^' marker.

Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

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 20.0.0.10 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 20.0.0.10
               [1/0] via 20.0.0.20
Router#
```


10.0.0.1
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=15ms TTL=125
Reply from 40.0.0.1: bytes=32 time=11ms 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 = 15ms, Average = 9ms
PC>

# EXPERIMENT-4

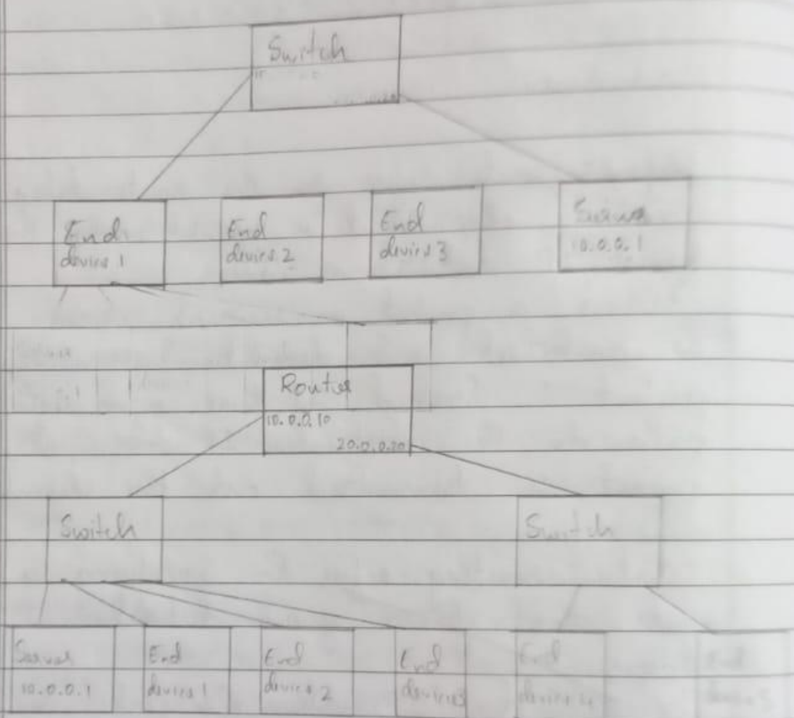
Configure default route, static route to the Route

Experiment-4

**Aim:**

Configure DHCP within a LAN and outside LAN

**Topology:**




**Procedure:**

① Connect 3 end devices and 1 server to a switch using copper straight through cable

② Go to service tab in server and turn on DHCP service.

③ Set IP address of server to 10.0.0.1. Set start IP address of to 10.0.0.2.

(4) Click on end device, go to desktop tab, go to IP configuration. Select DHCP. Repeat process for all end devices within the LAN.

(5) Ping end devices and observe output.

(6) Add one router, a switch and two end devices.

(7) Change server pool and set start IP address to 20.0.0.1.

(8) Configure ~~a~~ router IP address. Use the following commands

(i) enable Router> enable

(ii) config t Router# config t

Router(config)# interface <port>

Router(config-if)# ip address <ip address> <subnet mask>

Router(config-if)# no shut

Router(config-if)# exit

(9) Go to server and set gateway as 10.0.0.10.

(10) Move to router CLI, interface connecting secondary LAN. Use command

~~ip~~ Router(config-if)# ip helper-address <server ip address>

Router(config-if)# ip helper-address 10.0.0.1

(11) Repeat step (4) for all end devices in secondary LAN.

(12) Ping end devices and observe output.

Result.

ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3: bytes=32 time=0ms TTL=127
Reply from 10.0.0.3: bytes=32 time=0ms TTL=127
Reply from 10.0.0.3: bytes=32 time=1ms TTL=127
Reply from 10.0.0.3: bytes=32 time=0ms TTL=127

Ping statistics from 10.0.0.3:
packets sent=4, Recieved=4, Lost=0 (0% Loss)
Approximate round trip times in milli seconds
Minimum=0ms, Maximum=1ms Average=0ms

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

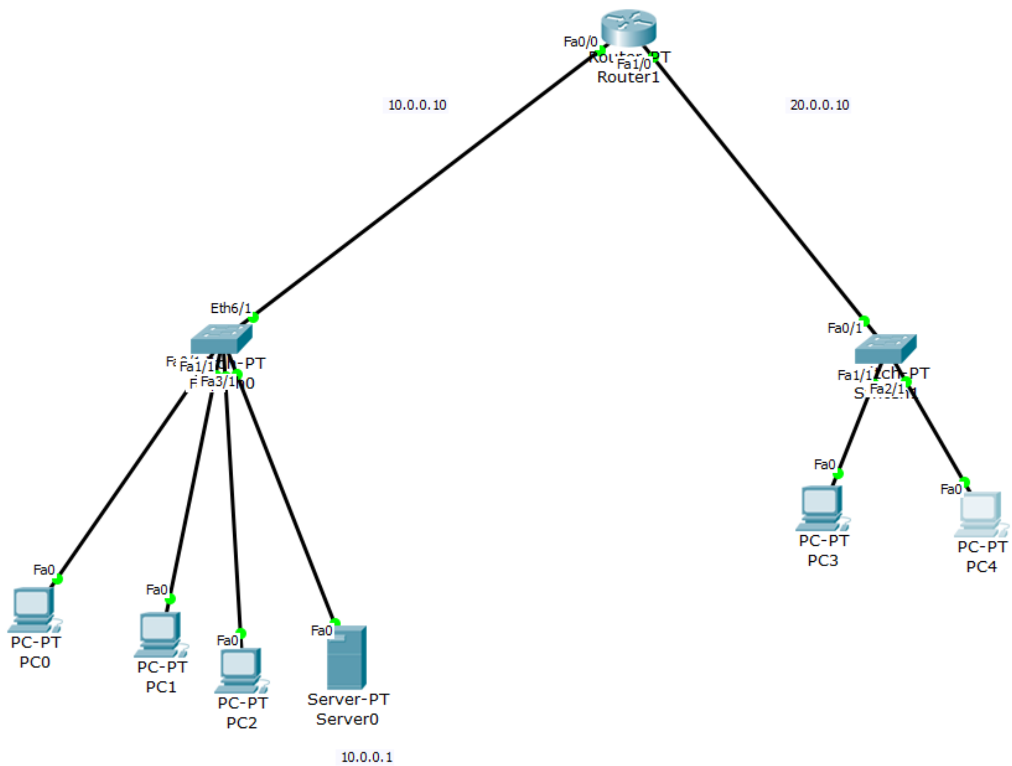
Ping statistics for 20.0.0.2
Packets sent=4, Recieved=3, Lost=1 (25% Loss)
Approximate round trip times in milli seconds
Minimum=0ms, Maximum=0ms, Average=0ms.

Observation:

10

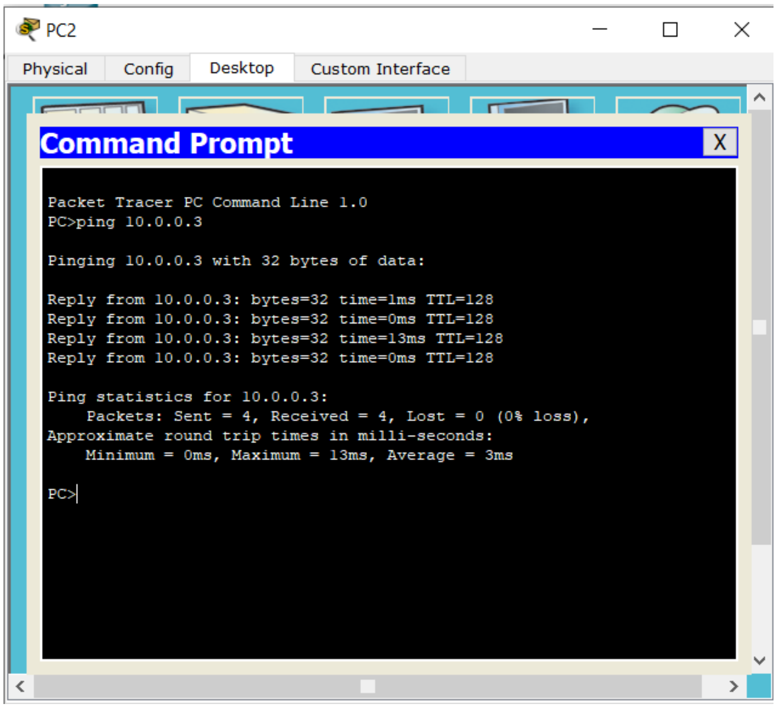
DHCP (Dynamic Host Configuration Protocol) is used to dynamically assign IP addresses to other devices.
The server manages a pool of IP addresses (known as server pool). The server responds to a client request. The provided IP configuration is based on information from address pools.

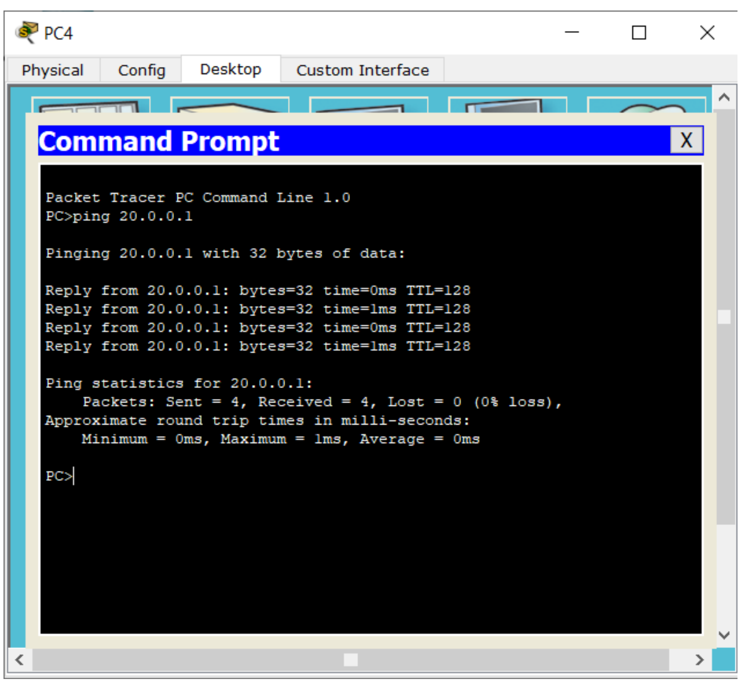
Topology:



Fa0/0
Fa1/0
Router1
10.0.0.10
20.0.0.10
Eth6/1
Fa1/1
Fa3/1
Fa0/1
Fa2/1
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
10.0.0.1
PC-PT
PC3
PC-PT
PC4

Result:


PC2
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=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=13ms 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 = 13ms, Average = 3ms
PC>


PC4
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=0ms TTL=128
Reply from 20.0.0.1: bytes=32 time=1ms TTL=128
Reply from 20.0.0.1: bytes=32 time=0ms TTL=128
Reply from 20.0.0.1: bytes=32 time=1ms TTL=128
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

# EXPERIMENT-5

Configure RIP routing Protocol in Routers.

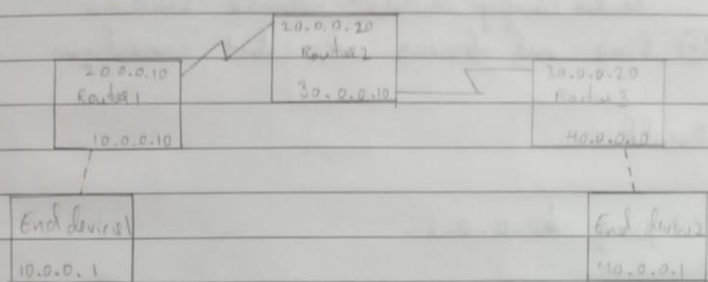
Date ______ Page ______

## Experiment - 5

**Aim:**

Configure RIP routing protocol in routers

**Topology:**




**Procedure:**

① Connect 3 routers and 2 end devices.
② Assign IP addresses to both end devices.
③ Assign IP addresses to all routers. Use the following commands

Router> enable
Router # config t
Router (config) # interface <port>
Router (config-if) # ip address <ip address> <subnet mask>
Router (config-if) # no shut
Router (config-if) # exit

④ Set gateways to end devices.
End device 1 : 10.0.0.10
End device 2 : 40.0.0.10

⑤ Assign routes to all routers. Move to configure mode of router and use the following commands.

Router (config)# router rip
Router (config-router)# network <network address>
Router (config-router) # network <network address>

For router 1,

Router (config)# router rip
Router (config-router)# network 10.0.0.0
Router (config-router)# network 20.0.0.0

⑥ Ping end devices to test connection.

Result:

> 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=5ms TTL=125
Reply from 10.0.0.1: bytes=32 time=21ms 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=1 (25% loss)

Approximate round trip times in milli-seconds:
Minimum=2ms, Maximum=21ms, Average=10ms.

Observation:

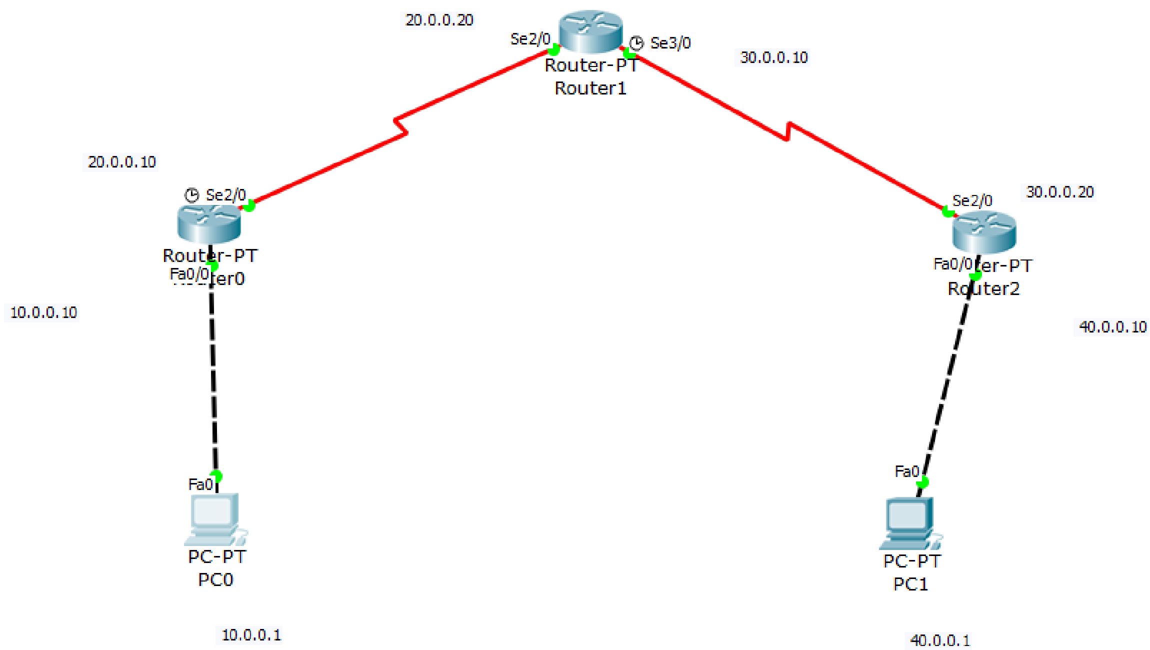
Routing Information Protocol (RIP) is a dynamic routing protocol that uses hop count as routing metric to find best path between source and destination networks. Hop count is the number of routers occuring in

SURYA Gold

Date______ Page______

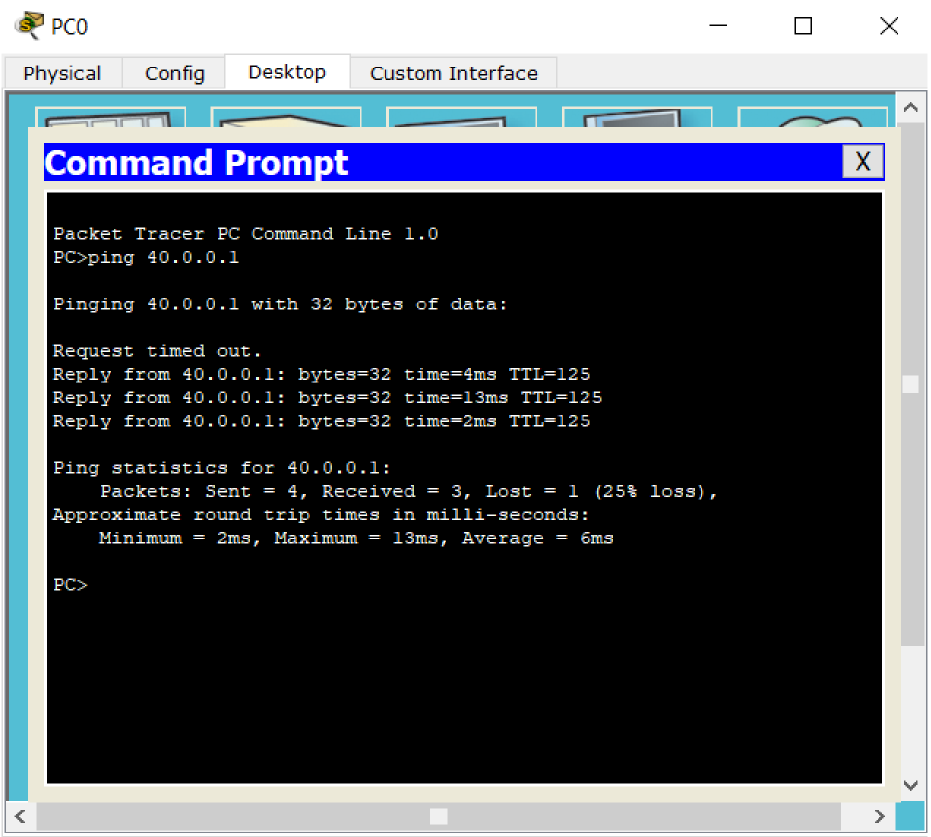
between source and destination networks.

Topology:



20.0.0.20
Se2/0
Se3/0
Router-PT
Router1
30.0.0.10
20.0.0.10
Se2/0
Se2/0
30.0.0.20
Router-PT
Fa0/0
Router0
Fa0/0
Router2
10.0.0.10
40.0.0.10
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
10.0.0.1
40.0.0.1

Result:


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=4ms TTL=125
Reply from 40.0.0.1: bytes=32 time=13ms 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 = 13ms, Average = 6ms
PC>

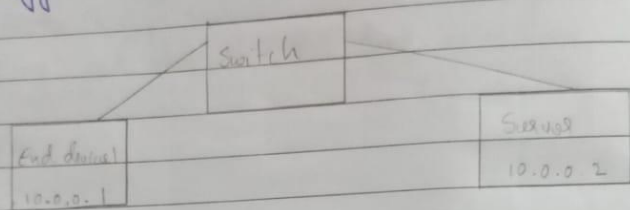
# EXPERIMENT-6

Configure Web Server, DNS within a LAN.

## Experiment-6

Aim:
Configure web server, DNS within a LAN.

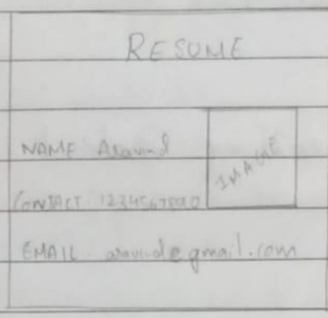
Topology:




Procedure:

① Connect an end device and a server to a switch.
② Assign IP addresses to both end device and server.
③ Configure DNS service on server.
Move to the services tab and open DNS server from the menu.
④ Turn ON the DNS service.
⑤ Add and save domain names with their IP addresses.
⑥ Open web browser on end device, enter domain name and observe the output.

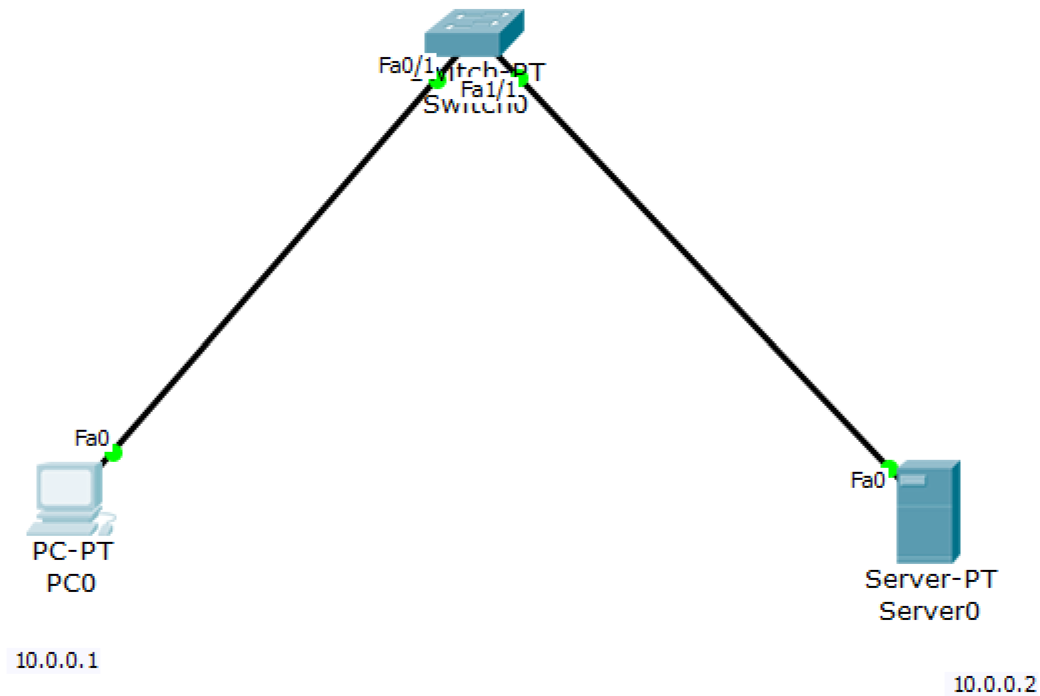
Result:




Observation:

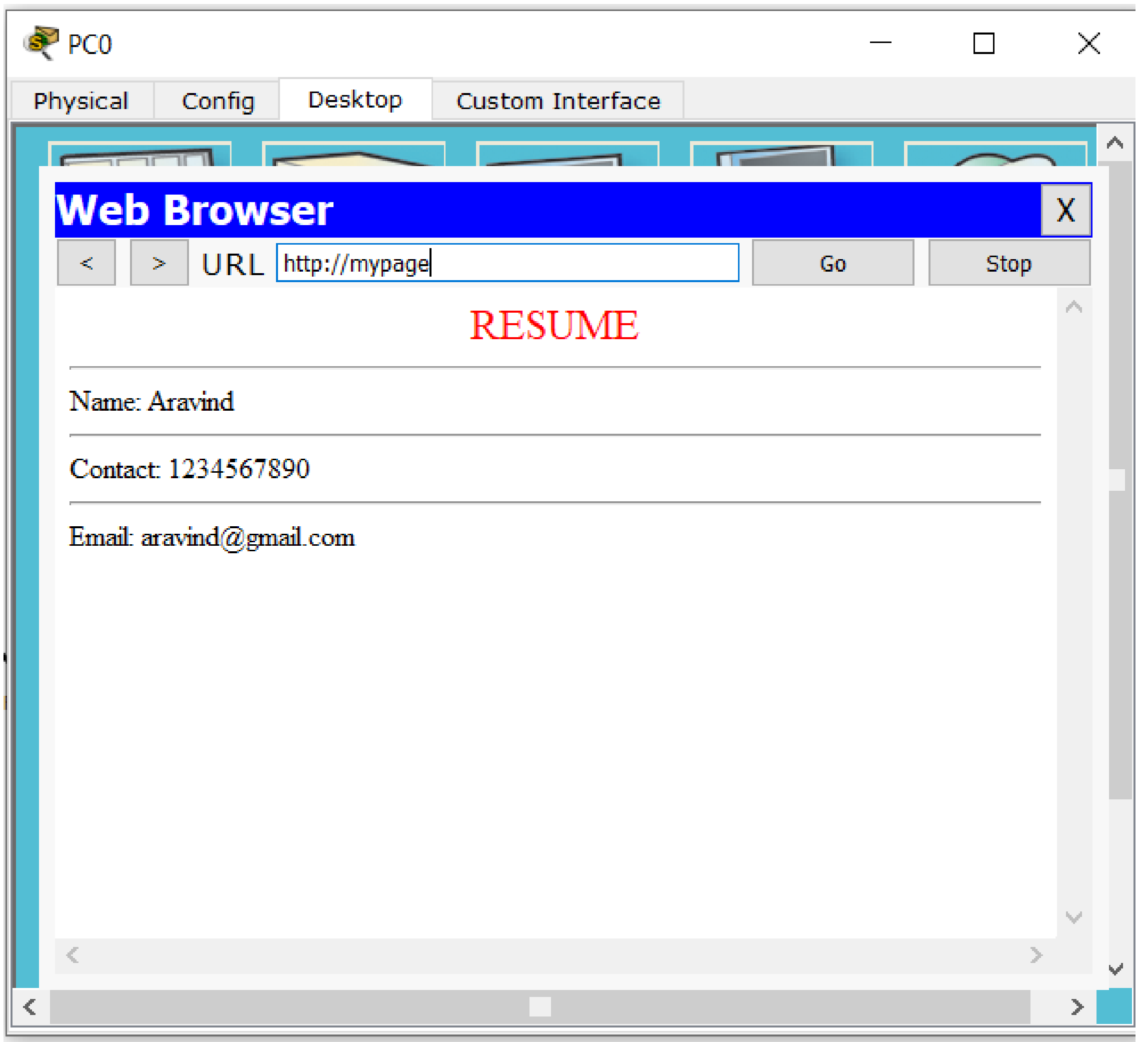
On entering domain name on web browser, we obtain stored web page. Domain Name Service (DNS) translates domain ~~ma~~ names to their respective IP addresses, enabling system communication even when user doesn't know the IP address.

Topology:



Fa0/1
Switch-PT
Switch0
Fa1/1
Fa0
PC-PT
PC0
10.0.0.1
Fa0
Server-PT
Server0
10.0.0.2

Result:


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
<
>
URL
http://mypage
Go
Stop
RESUME
Name: Aravind
Contact: 1234567890
Email: aravind@gmail.com

# EXPERIMENT-7

Configure OSPF routing protocol.

SURYA Gold

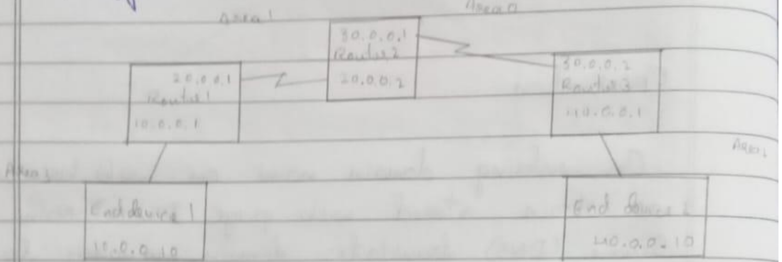
Date______ Page______

## Experiment - 7

**Aim:**

Configure OSPF routing Protocol

**Topology:**




**Procedure:**

1. Connect 3 routers and 2 end devices.
2. Configure end devices with IP addresses and gateways according to topology seen above.
3. Configure all routers according to the IP addresses as shown in the topology.
4. Encapsulation ppp and clock rate need to be set as done in RIP protocol experiment.
5. Enable IP routing by configuring OSPF protocol in all routers.

```
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.0.0.0 0.0255.255.255
                                              area 3.
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 area1.
R2(config-router)# network 30.0.0.0 0.255.255.255 area0.
R2(config-router)# exit

In Router R3,

R3(config)# router ospf 1
R3(config-router)# router-id 3.3.3.3
R3(config-router)# network 30.0.0.0 0.255.255.253 area0
R3(config-router)# network 40.0.0.0 0.255.255.255area 2
R3(config-router)# exit.

6) Create loopback on interfaces. Only consider router-router interfaces.

R1(config-if)# interface loopback 0
R1(config-if)# ip address 172.16.1.252 255.255.0.0
R1(config-if)# no shutdown

R2(config-if)# interface loopback0
R2(config-if)# ip address 172.16.1.253 255.255.0.0
R2(config-if)# no shutdown

R3(config-if)# interface loopback 0
R3(config-if)# ip address 172.16.1.254 255.255.0.0
R3(config-if)# no shutdown

⑦ Create virtual link between R1, R2 to connect to area 0.

In Router 1,

R1 (config)# router ospf 1

R1 (config-router)# area 1 virtual-link 2.2.2.2

In Router 2,

R2 (config) # router ospf 1

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

⑧ To test the connection, ping end devices.

Result:

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 = 10ms TTL = 125

Reply from 40.0.0.10: bytes = 32 time = 2ms TTL = 125

Reply from 40.0.0.10: bytes = 32 time = 9ms TTL = 125

Ping statistics from 40.0.0.10:

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

Approximate round trip times in milliseconds:

Minimum = 2ms, Maximum = 10ms, Average = 7ms

Router 1:

show ip route.

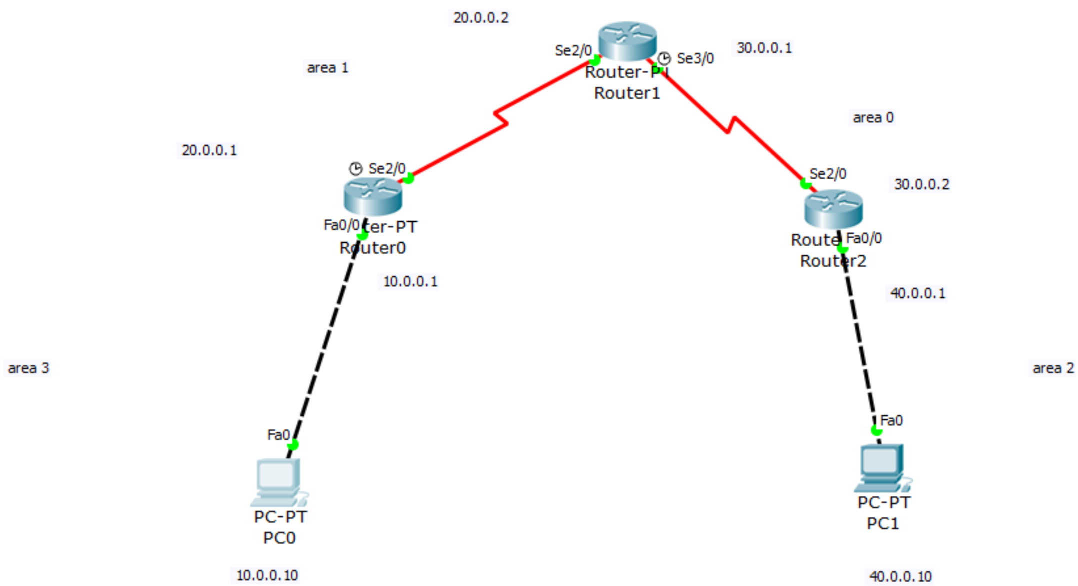
O IA 10.0.0.0/8 [110/65] via 20.0.0.1 00:00:11, Serial2/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks.
c 20.0.0.0/8 is directly connected, Serial2/0
c 20.0.0.1/32 is directly connected, Serial2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
c 30.0.0.0/8 is directly connected, Serial3/0
c 30.0.0.2/32 is directly connected, Serial 3/0
O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 00:04:44, Serial3/0.
c 172.16.0.0/16 is directly connected, Loopback0.

Observation:

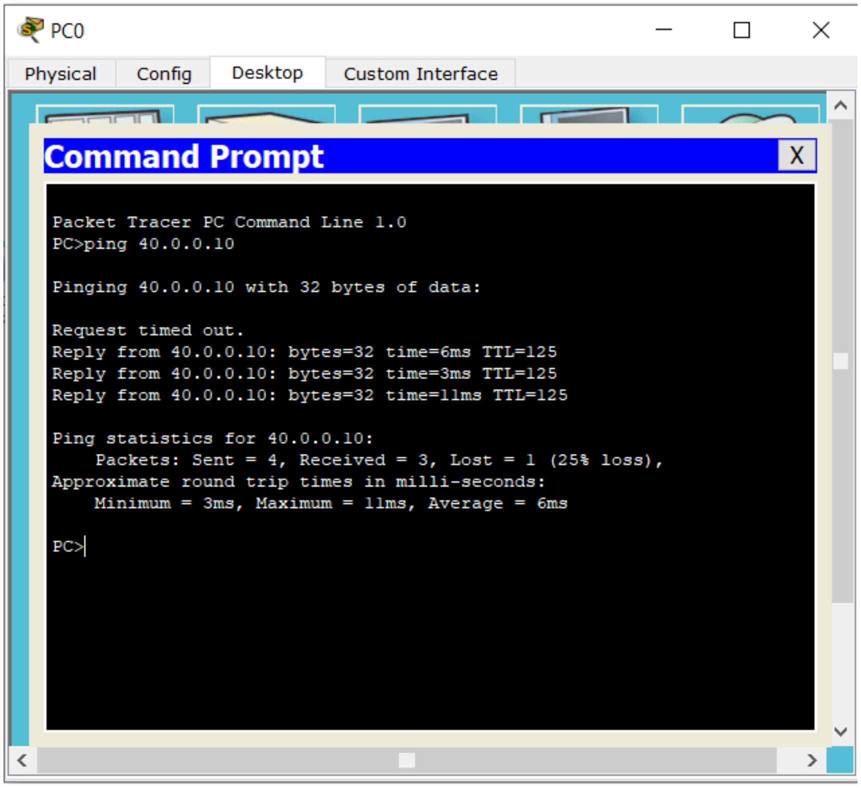
OSPF (Open Shortest Path First) protocol is a dynamic routing protocol that uses Dijikstra Algorithm to calculate shortest path to reach a destination network. OSPF routers share information about their directly connected networks. OSPF router interfaces are assigned an area and routers exchange OSPF information only within their assigned area.

3/3/23

Topology:


20.0.0.2
Se2/0
Router-PT
Router1
Se3/0
30.0.0.1
area 1
area 0
20.0.0.1
Se2/0
Se2/0
30.0.0.2
Fa0/0
Router0
10.0.0.1
Fa0/0
Router2
40.0.0.1
area 3
area 2
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
10.0.0.10
40.0.0.10

Result:

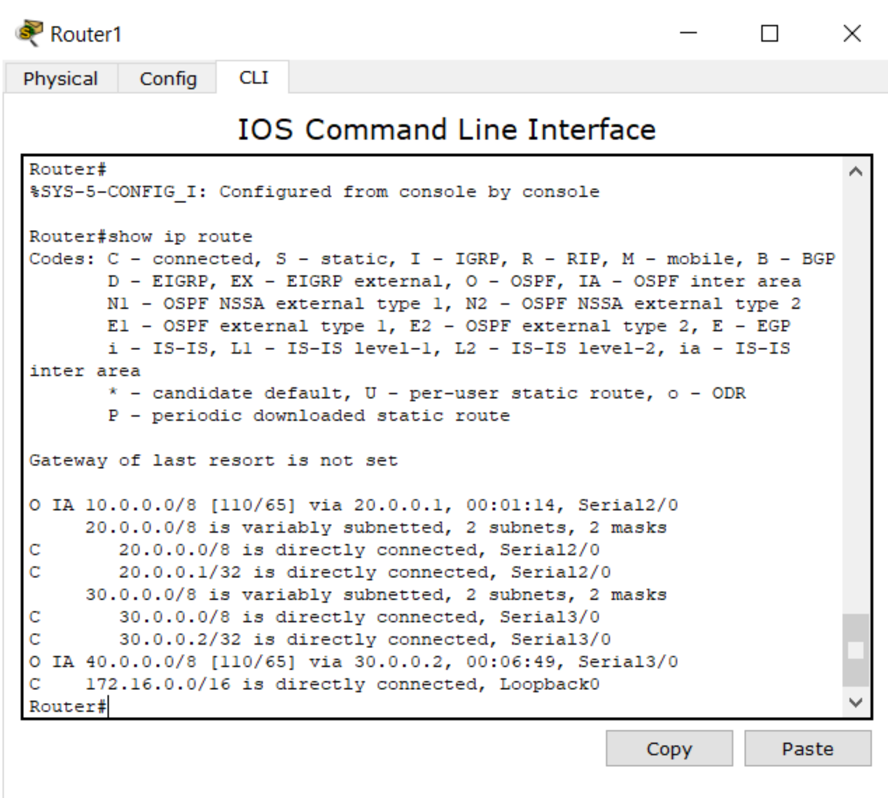

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=3ms TTL=125
Reply from 40.0.0.10: bytes=32 time=11ms 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 = 3ms, Maximum = 11ms, Average = 6ms

PC>


Router1
Physical
Config
CLI
IOS Command Line Interface
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

O IA 10.0.0.0/8 [110/65] via 20.0.0.1, 00:01:14, Serial2/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.1/32 is directly connected, Serial2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, Serial3/0
C 30.0.0.2/32 is directly connected, Serial3/0
O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 00:06:49, Serial3/0
C 172.16.0.0/16 is directly connected, Loopback0
Router#
Copy
Paste

# EXPERIMENT-8

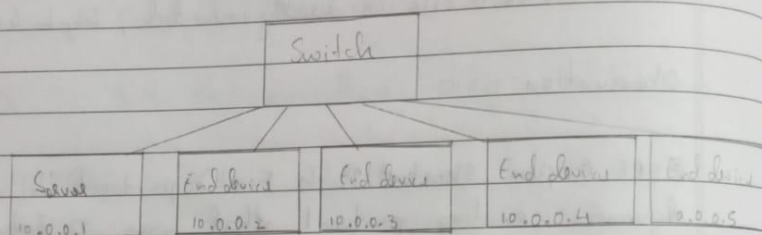
To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Date ____ Page ____

Experiment-8

Aim: Construction of a simple LAN to understand concept and operation of Address Resolution Protocol (ARP).

Topology:




Procedure:

1. Connect 4 PCs and a server to a switch.
2. Assign IP addresses to all devices.
3. Use the 'inspect tool' and click on PC to view its ARP table.
4. In command prompt of PC, use command 'arp -a' to view ARP table. Initially ARP Table will be empty.
5. In the CLI of switch, use command 'show mac address-table' to observe ARP Table of the switch.
6. In the simulation mode, capture packets from one device to another and observe updation of ARP table.

Result:

In command prompt of 10.0.0.2

> arp -a

| Internet address | Physical Address | Type |
|---|---|---|
| 10.0.0.1 | 0090.2b50.070a | dynamic |
| 10.0.0.3 | 00d0.ba36.9687 | dynamic |
| 10.0.0.4 | 00d0.97ed.5ae9 | dynamic |
| 10.0.0.5 | 0002.17d5.c0a0 | dynamic. |

In CLI of switch

> show mac-address-table

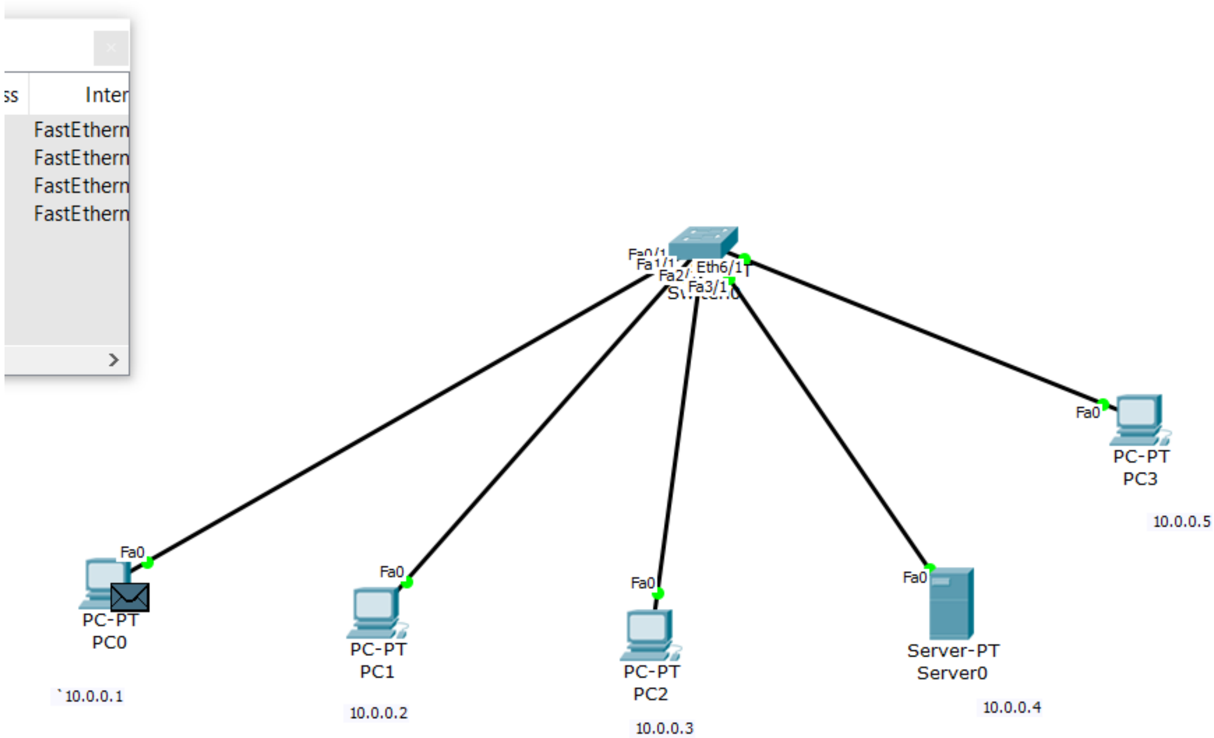
Mac Address Table

| Vlan | Mac address | Type | Ports |
|---|---|---|---|
| 1 | 0002.17d5.c0a0 | dynamic | Eth 6/1 |
| 1 | 0003.e417.b2d5 | dynamic | Fa 1/1 |
| 1 | 0090.2b50.070a | dynamic | Fa0/1 |
| 1 | 00d0.97ed.5ae9 | dynamic | Fa 3/1 |
| 1 | 00d0.ba36.9687 | dynamic | Fa2/1 |

Observation:

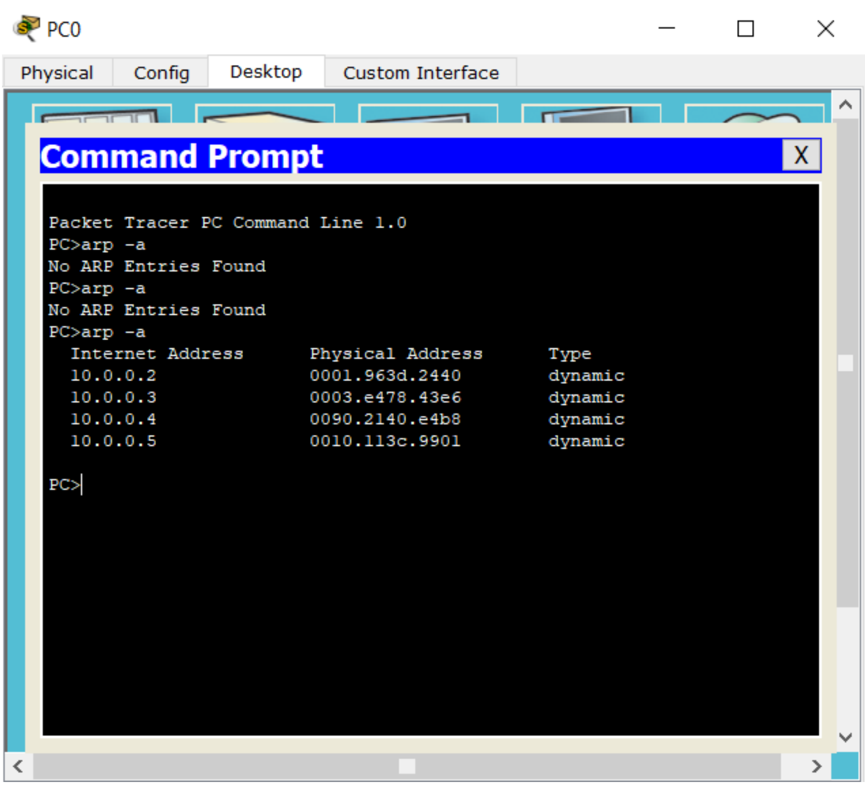
ARP (Address Resolution Protocol) is a layer 3 protocol that is used to find the MAC address from the IP address. An ARP broadcast is sent requesting MAC address of ~~required~~ mentioned IP address. The called device unicasts with its MAC address. The MAC address is updated in the ARP table with

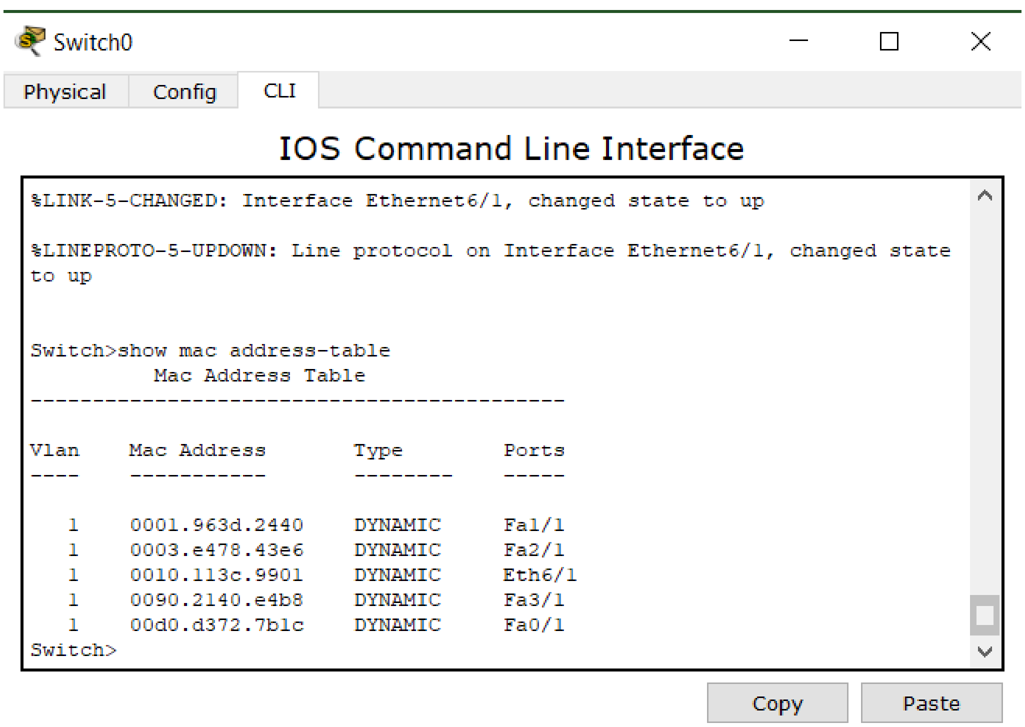
Topology:



Inter
FastEthern
FastEthern
FastEthern
FastEthern
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
10.0.0.2
Fa0
PC-PT
PC2
10.0.0.3
Fa0
Server-PT
Server0
10.0.0.4
Fa0
PC-PT
PC3
10.0.0.5
Eth6/1
Fa3/1

Result:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>arp -a
No ARP Entries Found
PC>arp -a
Internet Address Physical Address Type
10.0.0.2 0001.963d.2440 dynamic
10.0.0.3 0003.e478.43e6 dynamic
10.0.0.4 0090.2140.e4b8 dynamic
10.0.0.5 0010.113c.9901 dynamic
PC>


Switch0
Physical
Config
CLI
IOS Command Line Interface
%LINK-5-CHANGED: Interface Ethernet6/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet6/1, changed state to up
Switch>show mac address-table
Mac Address Table
-------------------------------------------
Vlan Mac Address Type Ports
---- ----------- -------- -----
1 0001.963d.2440 DYNAMIC Fa1/1
1 0003.e478.43e6 DYNAMIC Fa2/1
1 0010.113c.9901 DYNAMIC Eth6/1
1 0090.2140.e4b8 DYNAMIC Fa3/1
1 00d0.d372.7b1c DYNAMIC Fa0/1
Switch>
Copy
Paste

# EXPERIMENT-9

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

3/8/23

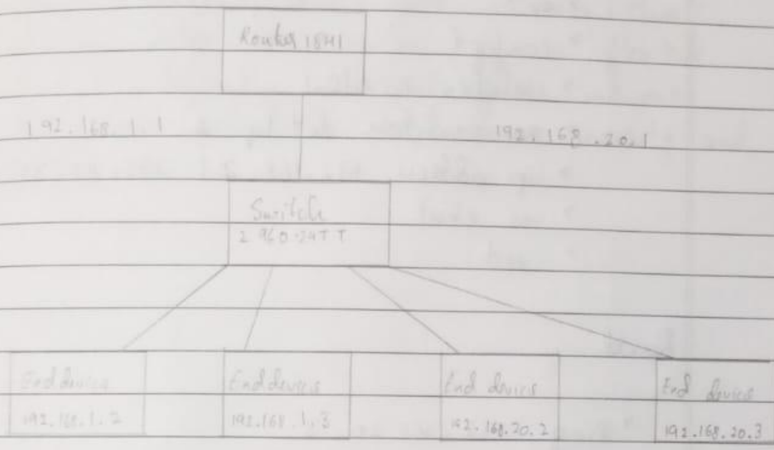
SURYA Gold

Date ______ Page ______

## Experiment -9

Aim: To construct a VLAN and make a PC communicate among VLAN.

Topology:




Procedure:

① Create topology with one 1841 router connected to a 2960-24TT switch with 4 connected end devices.

②. Set IP addresses to the router and 4 end devices. Use Class C addresses. Also set gateway.

③ In switch, go to config tab and select VLAN database. Provide a VLAN number and name.

④ Go to the switch-router interface and set it as trunk.

⑤ select switches under 2nd interface which has interfaces connecting to virtual network end devices. Click on each one and set VLAN number.

⑥ Go to router's config tab and VLAN's database and enter VLAN number and name.

① Go to router CLI and use following commands

> config t
> interface Fa 0/0
> Ip address 192.168.1.1 255.255.255.0
> no shut
> exit
> config t
> interface Fa 0/0.1
> encapsulation dot 1q 2
> ip address 192.168.20.1 255.255.255.0
> no shut
> exit

Result:

> 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 milliseconds:
Minimum=0ms, Maximum=5ms, Average=1ms

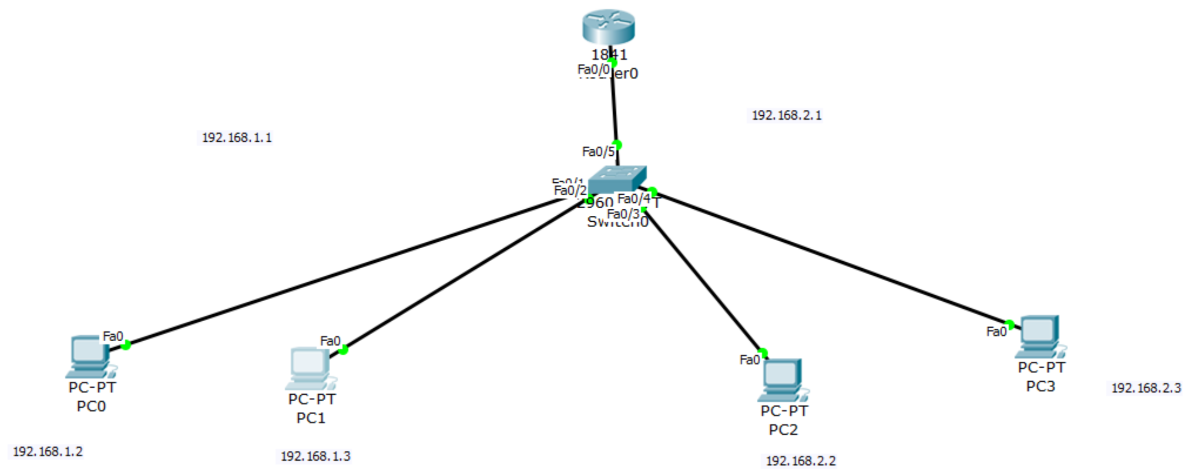
SURYA Gold

Date ______ Page ______

Observation

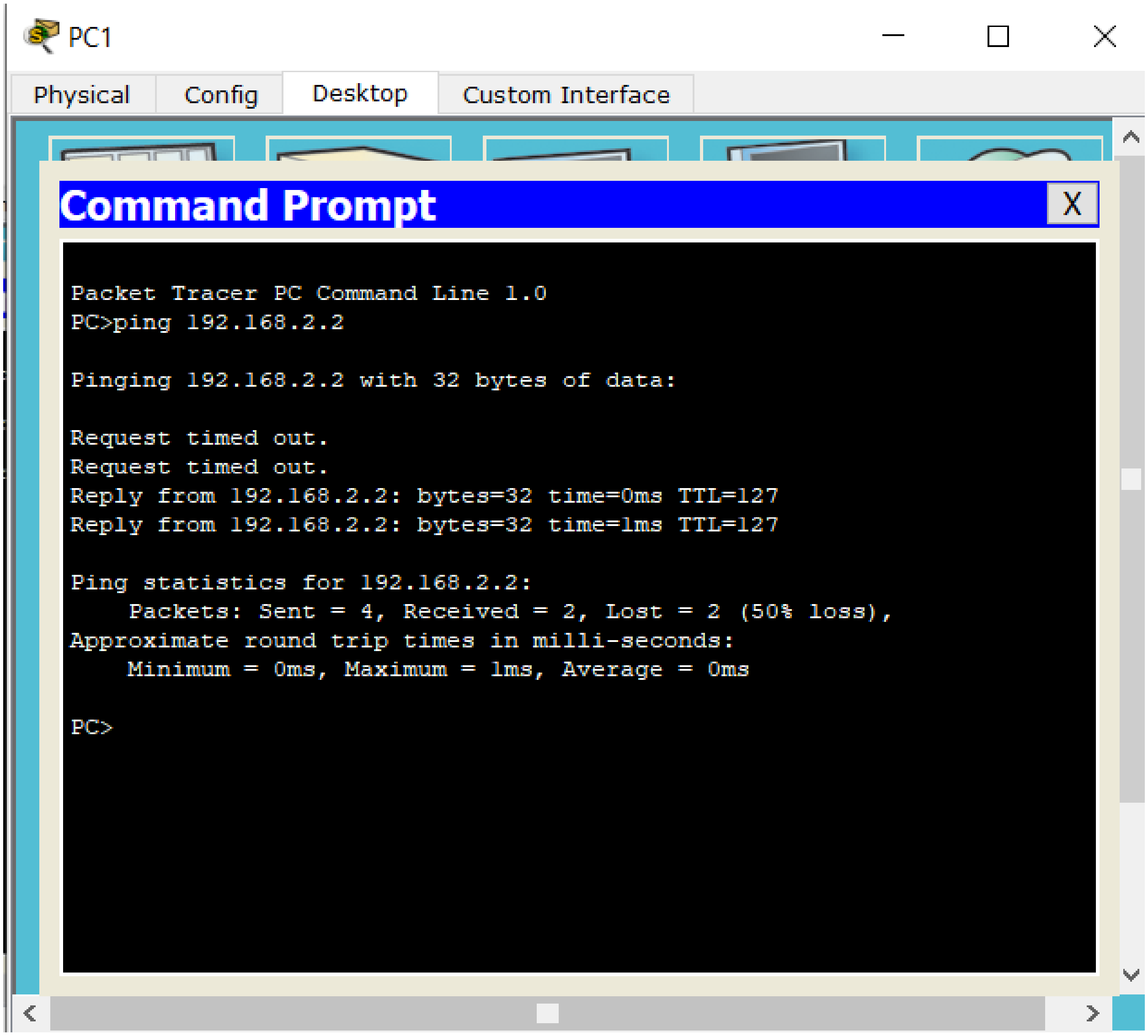
We have two devices, each on a different VLAN but on the same switch. They will only hear other broadcast traffic from within their VLANs. Here, VLANs don't use IP addresses, instead deal with subnets / class C addresses. Inter-VLAN routing gives a flexible tool to logically subdivide their networks that has potential to enhance security and performance.

Topology:



1841
Fa0/0
Router0
192.168.1.1
192.168.2.1
Fa0/5
Fa0/1
Fa0/2
2960-24TT
Switch0
Fa0/4
Fa0/3
Fa0
Fa0
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
192.168.1.2
192.168.1.3
192.168.2.2
192.168.2.3

Result:


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2
Pinging 192.168.2.2 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=1ms TTL=127
Ping statistics for 192.168.2.2:
Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

# EXPERIMENT- 10

Demonstrate the TTL/ Life of a Packet

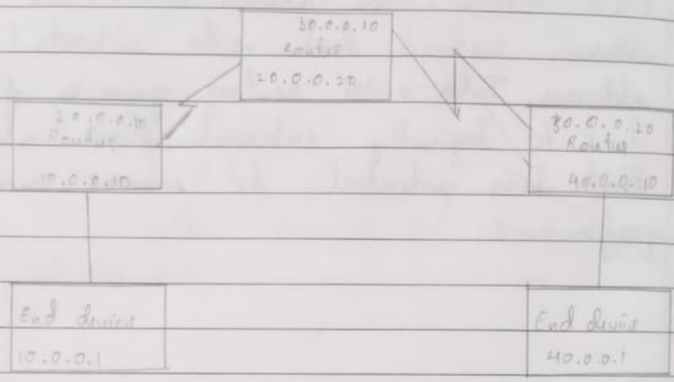
SURYA Gold

Date ______ Page ______

## Experiment - 10

Aim: Demonstrate TTL / life of a packet

Topology:




Procedure:

① Create a topology as shown with 2 end devices and 3 routers.
② Set ip addresses to end devices and also set gateways.
③ Configure routers either statically ~~or dynamically~~ / default routing.
④ In simulation mode, send a simple PDU from one end device to another.
⑤ Use capture button to view each movement of the PDU.
⑥ Click on the ~~to~~ PDU at each step and observe inbound and outbound details.

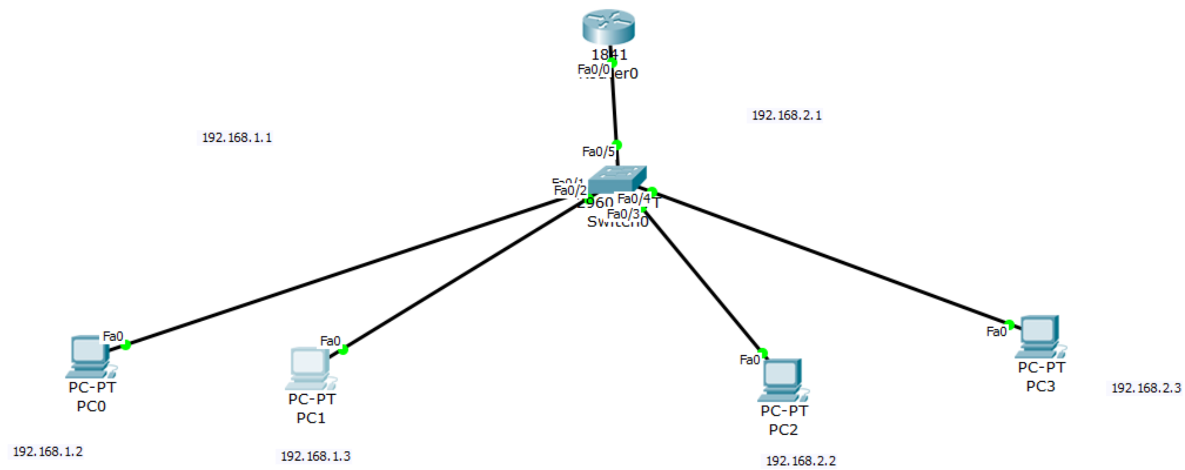
Results:

| | | | | |
|---|---|---|---|---|
| V4 | IHL | DSCP | | TTL LEN [illegible] |
| ID: 0x6 | | | 0x | 0x0 |
| TTL 255 | | PRO 0x1 | CHKSUM | |
| SRC IP: 10.0.0.1 | | | | |
| DST IP: 40.0.0.1 | | | | |
| OPT 0x0 | | | | 0x0 |
| DATA | | | | |

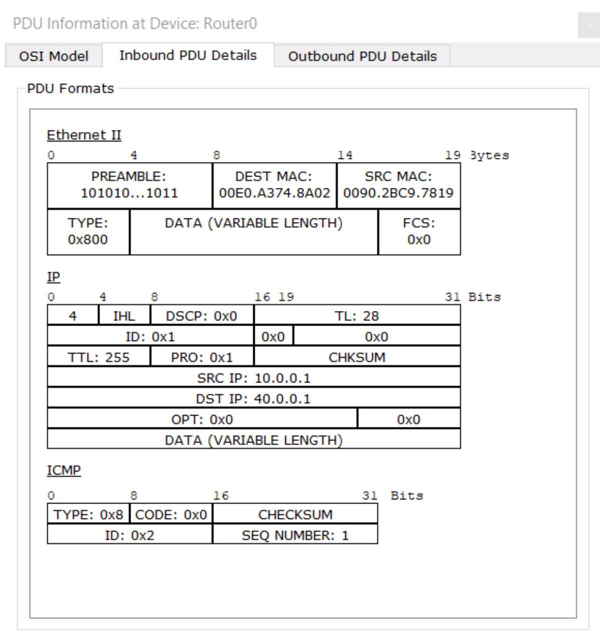
Observation:

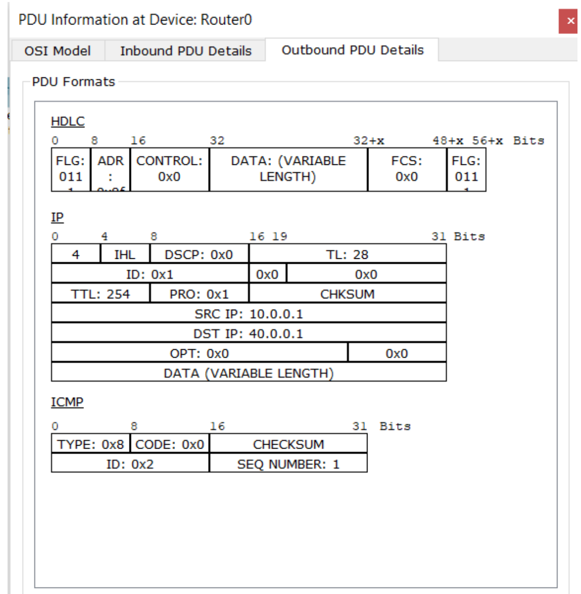
The number of hops the packet travels before being dropped is known as Time-to-Live (TTL. Datagram TTL field is set by the sender and reduced by each router along the path to its destination. The router reduces TTL value by one while forwarding the packets. When TTL value is 0, the router dropps packet and sends an ICMP message to sender.

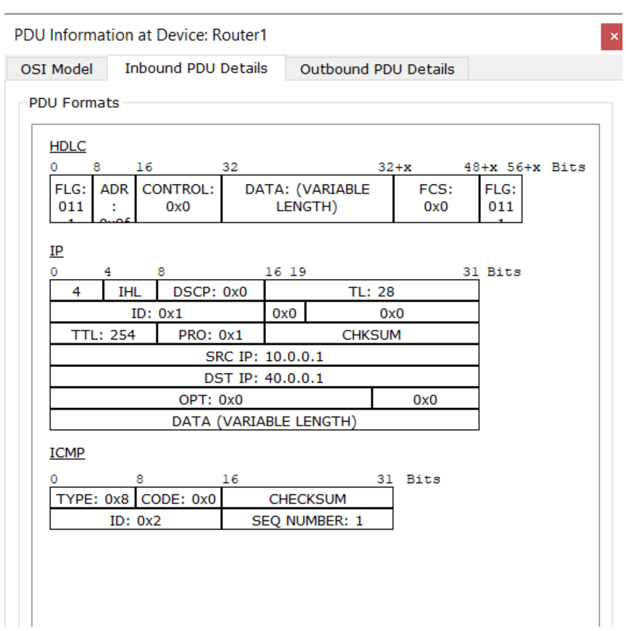
Topology:

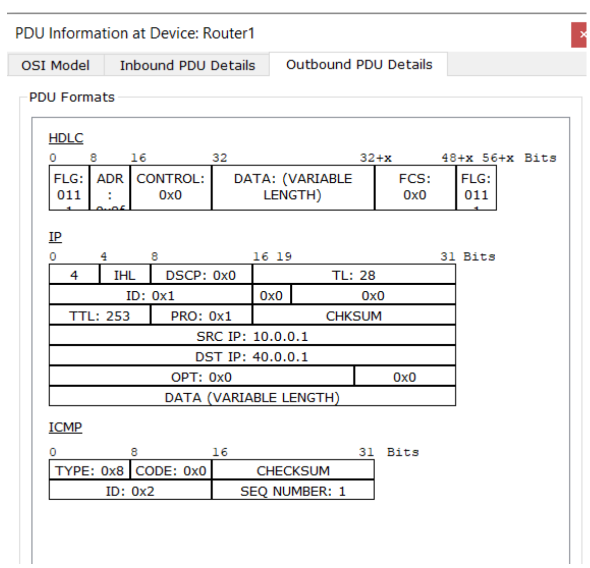

1841
Fa0/0
192.168.2.1
192.168.1.1
Fa0/5
Fa0/2
Fa0/4
Fa0/3
Fa0
PC-PT
PC0
192.168.1.2
Fa0
PC-PT
PC1
192.168.1.3
Fa0
PC-PT
PC2
192.168.2.2
Fa0
PC-PT
PC3
192.168.2.3

# Result:


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Bytes
PREAMBLE: 101010...1011
DEST MAC: 00E0.A374.8A02
SRC MAC: 0090.2BC9.7819
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x1
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x2
SEQ NUMBER: 1


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x Bits
FLG: 011
ADR :
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 011
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x1
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x2
SEQ NUMBER: 1


PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x Bits
FLG: 011
ADR :
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 011
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x1
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x2
SEQ NUMBER: 1


PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x Bits
FLG: 011
ADR :
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 011
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x1
0x0
0x0
TTL: 253
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x2
SEQ NUMBER: 1

# EXPERIMENT- 11

To construct a WLAN and make the nodes communicate wirelessly

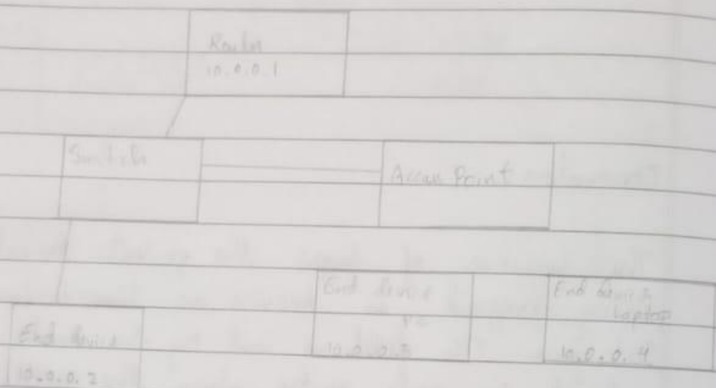
SURYA Gold

Date______ Page______

Experiment - 11

Aim: To construct a WLAN and make nodes communicate wirelessly.

Topology:




Procedure:

① Construct above shown topology.
② Configure end device connected to switch and router.
③ Configure Accesspoint port1 and set SSID name.
④ Select WEP and provide a 10 digit Hexdecimal password.
⑤ Configure PC and laptop with wireless standards.
⑥ Switch off the device, drag PT-Host-NM1AM component back to tab and drag WPC300N wireless interface to empty port and switch on the device.
⑦ A new wireless interface would be added to config tab. Configure SSID, WEP, WEP key, IP address and gateway to devices.

(10) Ping from any device to another.

Results

> 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
Reply from 10.0.0.3: bytes=32 time=0ms TTL=127
Reply from 10.0.0.3: bytes=32 time=2ms TTL=127

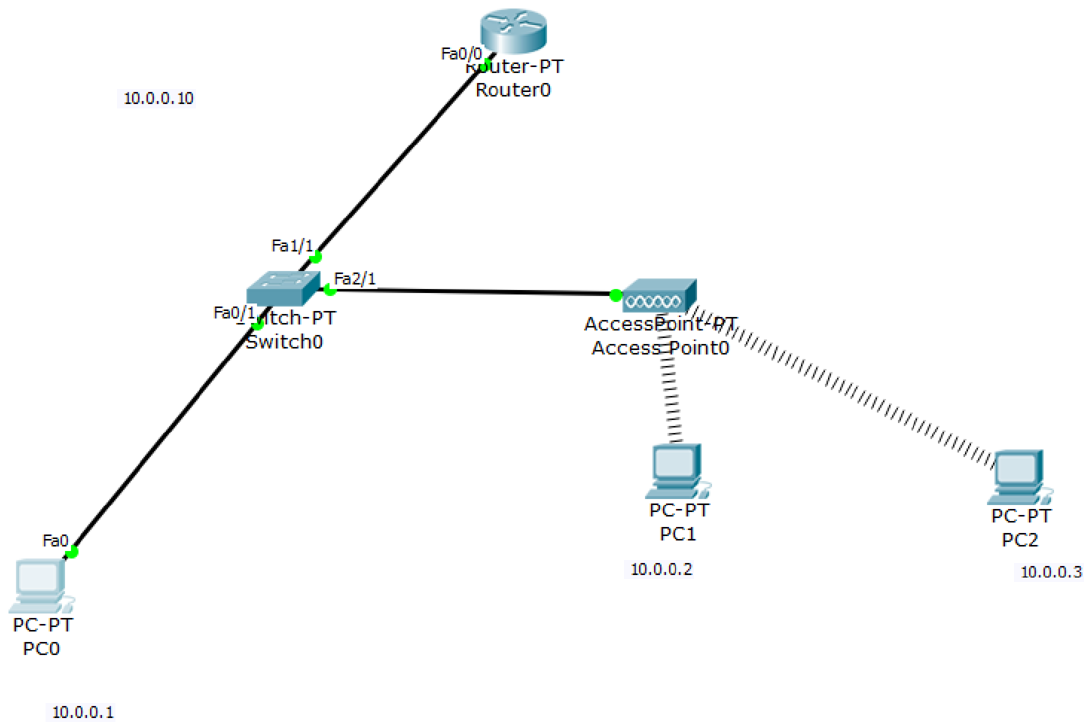
Ping statistics for 10.0.0.3:
Packets: Sent=4 Received=3 Lost=1 (25% loss)

Approximate round trip times in milliseconds
Minimum = 0ms, Maximum = 2ms, Average = 0ms

Observation:

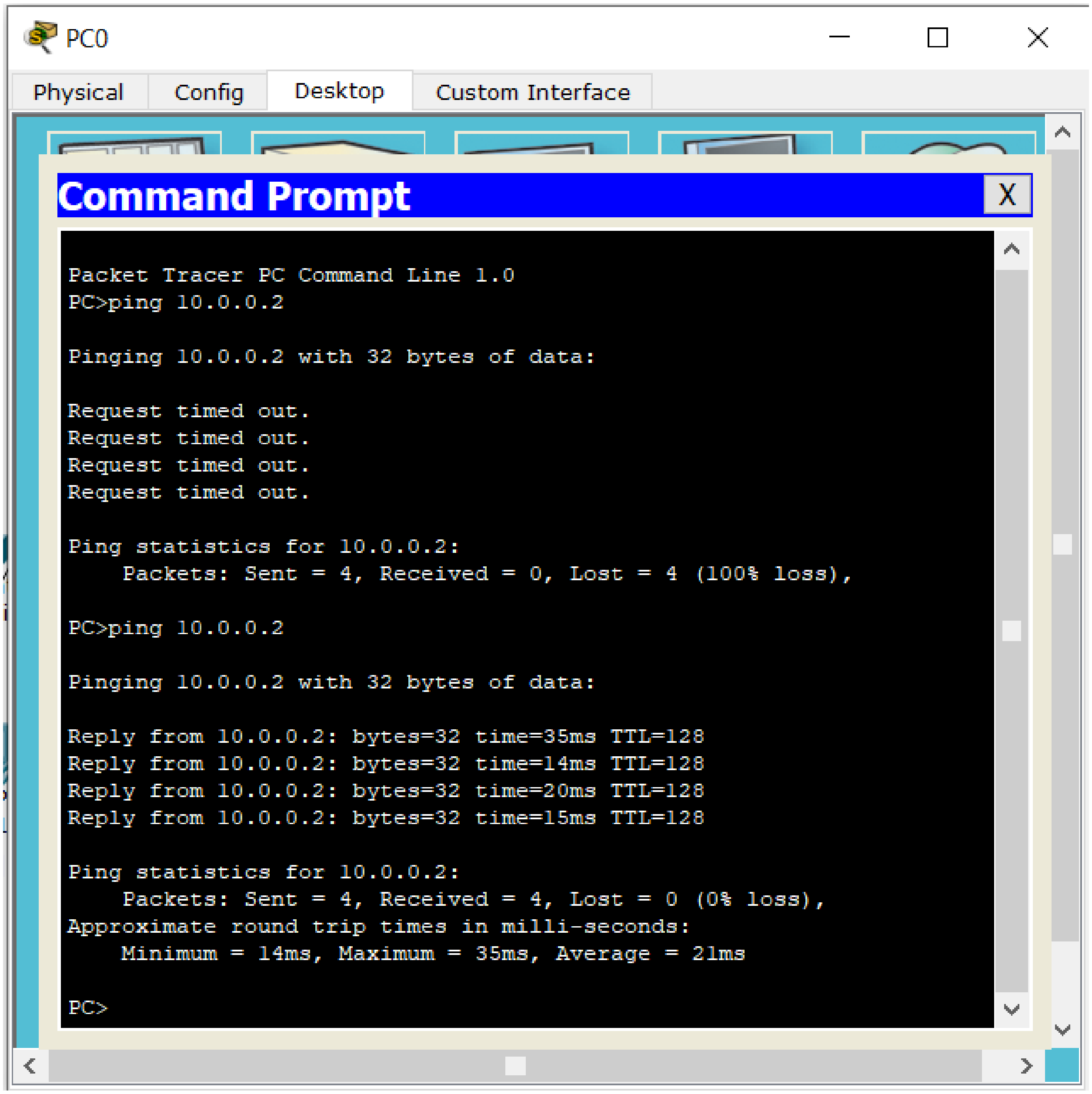
A WAN is a group of co-located devices that form a network based on radio transmissions. Data sent in packets contain layers with labels and instructions, MAC addresses to end points for routing. Accesspoint is the base station that serves as a hub to which other stations connect. We can connect multiple devices to a single accesspoint wirelessly.

Topology:



Fa0/0
Router-PT
Router0
10.0.0.10
Fa1/1
Fa2/1
Fa0/1
Switch-PT
Switch0
AccessPoint-PT
Access Point0
Fa0
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3
PC-PT
PC0
10.0.0.1

Result:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=35ms TTL=128
Reply from 10.0.0.2: bytes=32 time=14ms TTL=128
Reply from 10.0.0.2: bytes=32 time=20ms TTL=128
Reply from 10.0.0.2: bytes=32 time=15ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 14ms, Maximum = 35ms, Average = 21ms
PC>

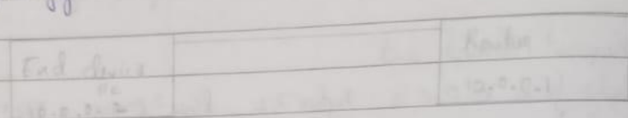
# EXPERIMENT- 12

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

Experiment - 12

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

Topology:




Procedure:

1. Create topology as shown above.
2. Configure IP address and gateway for end device.
3. Configure router using following commands.
   - > enable
   - > config t
   - > hostname r1
   - > enable secret p1
   - > interface Fa 0/0
   - > ip address 10.0.0.1 255.0.0.0
   - > no shut
   - > line vty 0 5
   - > login
   - > password p0
   - > exit
   - > exit
   - > wr
4. Ping router.

Password for user Access verification is po
password for enable is p1

Result:

> 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 (0% loss)

Approximate roundtrip times in milliseconds:
Minimum=0ms, Maximum=0ms, Average=0ms

IP route:

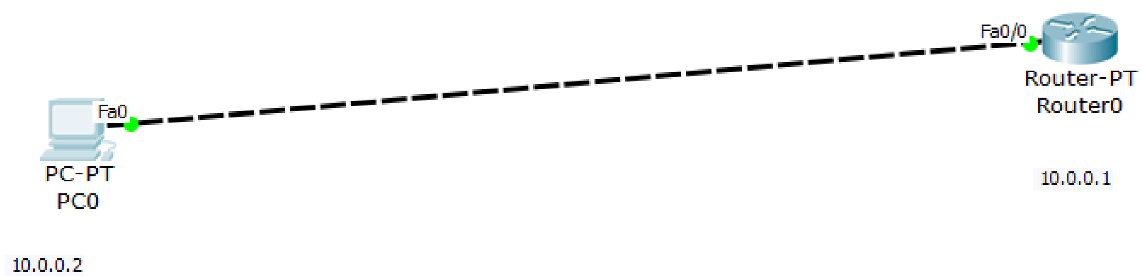
PC> telnet 10.0.0.1
Trying 10.0.0.1 ... open
User Access Verification
Password: po
Router> enable
Password p1
Router# show IP route

C 10.0.0.0/8 is directly connected, FastEthernet 0/0.

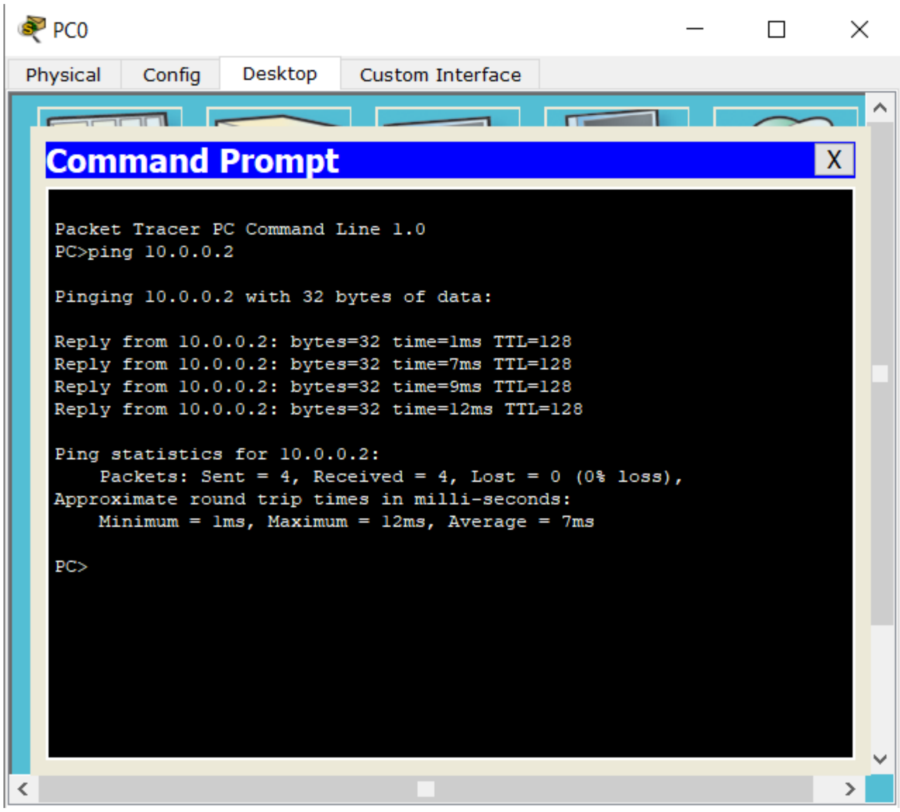
Observation:

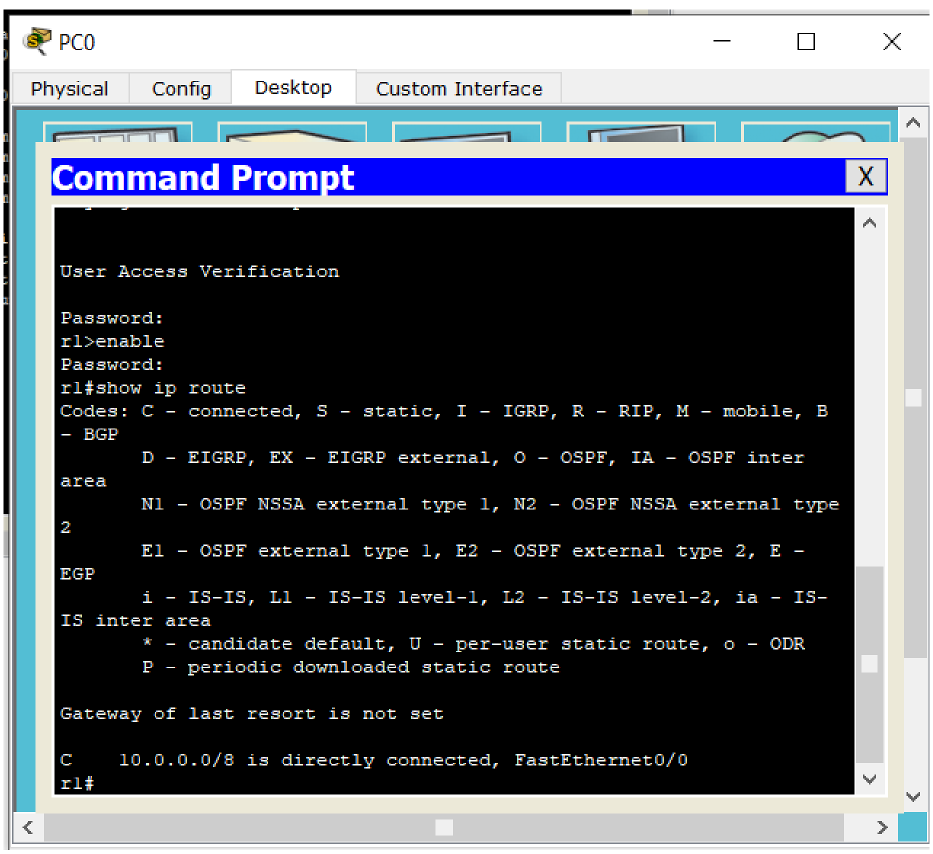
TELNET stands for Teletype network. It is a type of protocol that enables one computer to connect to the local computer. It is used as a standard TCP/IP protocol for virtual terminal service provided by ISO.
During TELNET operation, whatever is being performed on the remote computer will be displayed by the local computer. Telnet operates on a client/server principle.

## Topology:




## Result:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=7ms TTL=128
Reply from 10.0.0.2: bytes=32 time=9ms TTL=128
Reply from 10.0.0.2: bytes=32 time=12ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 12ms, Average = 7ms
PC>


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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#

# EXPERIMENT- 13

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

17/8/23

Experiment-13

Aim: To write a program for error detection using CRC-CCITT (16-bits).

```
# include< stdio.h>
# include <string.h>

# define N strlen(divisor)

char  data[30];
char  rem [30];
char  divisor[20];
int   dlength, i, j;

void xor (){
  for (j=1; j<N; j++)
     rem [j] = (( rem[j]==divisor[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++];
    } while (i <= dlength+N-1);
}

void reciver(){

    printf("Enter data to be recived: ");
    scanf("%d", &data);
    printf("Data recieved: %s", data);
    crc();
    for (i=0; i<N-1 && rem[i] != '1' ; i++);
        ~~if (i~~
    if (i<N-1)
        printf(" Error in transmission");
    else
        printf (" No error in transmission");
}

int main(){

    printf("Enter data to be transmitted ");
    scanf("%s", &data);
    printf("Enter divisor");
    scanf("%s", &divisor);
    dlength = strlen(data);

    for (i=dlength; i<dlength+N-1 ; i++)
            data[i]= '0';

    crc();
    for( i=dlength; i<dlength+N-1; i++)
            data[i]= rem [i-dlength];
```

SURYA Gold

Date_____ Page_____

```
    printf ("Data being sent : %s", data);
    receiver ();

    return 0;
}
```

Output:

Enter data to be transmitted
1011
Enter divisor
10001000000100001
Data being sent: 1011 1 0 100 11011 0001

Enter data recieved : 10111010011011 0000
Data recieved : 10 11 10 100 11011 0000
Error in transmission

Code:

```
def crc(s: str)->str:

    bits=[int(t) for t in s]

    polynomial=0b10001000000100001

    crc=0b1111111111111111



    for bit in bits:

        crc^=(bit<<15)

        if (crc>>15)&1:

            crc=(crc<<1)^polynomial

        else:

            crc<<=1



    crc&=0b1111111111111111

    return bin(crc)[2:].zfill(16)



data=input("Enter data ")

rem=crc(data)

print("Remainder will be:",rem)

print("Data being transmitted is: ",(data+rem))
```

```python
rec=input("Enter recieved data ")

r=crc(rec)

print(r)

if int(r)==0:

    print("No Error detected")

else:

    print("Error detected")
```

Result:

```
PS C:\Users\aravi\OneDrive\Desktop\notes\CN> & C:/Python311/python.exe c:/U
Enter data 1011
Remainder will be: 1011111101110100
Data being transmitted is:  10111011111101110100
Enter recieved data 10111011111001110100
0011001100110001
Error detected
```

```
PS C:\Users\aravi\OneDrive\Desktop\notes\CN> & C:/Pythor
Enter data 1011
Remainder will be: 1011111101110100
Data being transmitted is:  10111011111101110100
Enter recieved data 10111011111101110100
0000000000000000
No Error detected
```

# EXPERIMENT- 14

Write a program for congestion control using Leaky bucket algorithm.

Experiment-14

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

```
# include <stdio.h>
# include <conio.h>

void main(){
   int bucket_size, dr;

  printf("Enter bucketsize and data rate");
  scanf("%d %d", &bucket_size, &dr);

  int emp= bucket_size;

  while(1) {
     int ch, ps;

 printf("Enter packet size\n");
  scanf("%d", &ps);

    if (ps<= bucket_size){

         if (ps<= emp)
              printf("packet size %d sent", ps);
         else{
              printf("packet size %d dropped", ps);
         ps=0; }
       emp= emp - ps + dr
  } else {
     printf("packet dropped");
```

```
    printf("continue transmission? ");
    scanf("%d",&ch);

    if (ch==0)
        break;
  }
}
```

Output:

Enter bucketsize and data rate
4000 250
Enter packet size
5000
packet dropped
Continue transmission?
1
Enter packet size
1000
packet size 1000 sent
Continue transmission? 1
Enter packet size
3000
packet size 3000 sent
Continue transmission?
1
Enter packet size
750
packet size 750 dropped
Continue transmission?
0

Code:

```
bucket=int(input("Enter bucket capacity "))

rate=int(input("Enter rate of data transmission "))

remaining=bucket

while True:

    packet=int(input("Enter packet size(-1 for no packets, -2 for end of transmission) "))



    if packet==-2:

        break




    if packet>bucket:

        print("Packet can't be sent")

    else:

        if remaining+rate>=packet:

            remaining-=packet

            print("Packet sent")

        else:

            print("Packet can't be sent")
```

```
    if remaining==bucket:

        continue



    if bucket-remaining<rate:

        remaining=bucket

        continue

    remaining+=rate
```

Result:

```
PS C:\Users\aravi\OneDrive\Desktop\notes\CN> & C:/Python311/python.exe c:/Users/a
.py
Enter bucket capacity 4000
Enter rate of data transmission 250
Enter packet size(-1 for no packets, -2 for end of transmission)5000
Packet can't be sent
Enter packet size(-1 for no packets, -2 for end of transmission)1000
Packet sent
Enter packet size(-1 for no packets, -2 for end of transmission)3000
Packet sent
Enter packet size(-1 for no packets, -2 for end of transmission)750
Packet sent
Enter packet size(-1 for no packets, -2 for end of transmission)500
Packet can't be sent
Enter packet size(-1 for no packets, -2 for end of transmission)-1
Packet sent
Enter packet size(-1 for no packets, -2 for end of transmission)500
Packet sent
Enter packet size(-1 for no packets, -2 for end of transmission)-2
```

# EXPERIMENT- 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.

Date ____ Page ____

## Experiment- 15

Aim: Using TCP/IP sockets, write a client-server program to make client requesting file name and server sending contents of the requested file.

Client TCP.py

```
from socket import *

name = "127.0.0.1"
port = 12000
socket = socket(AF_INET, SOCK_STREAM)
socket.connect((name, port))
message = input("Enter file to be received")
socket.send(message.encode())
received = socket.recv(1024).decode()
print("File contents" + received)
socket.close().
```

serverTCP.py

```
from socket import *

name = "127.0.0.1"
port = 12000
socket = socket(AF_INET, SOCK_STREAM)
socket.bind((name, port))
socket.listen(1)
while1:
    print("Ready to receive")
    socket, addr = socket.accept()
```

```
sentence = socket.recv(1024).decode
file = open (sentence, 'r')
l = file.read(1024)
socket.send (l.encode())
print("Sent contents")
file.close()
socket.close()
```

Output:

Server:
Ready to receive
Sent contents

Client:

Enter file to be received
server.py
~~Reply to~~ File contents
// contents of server.py

## Code:

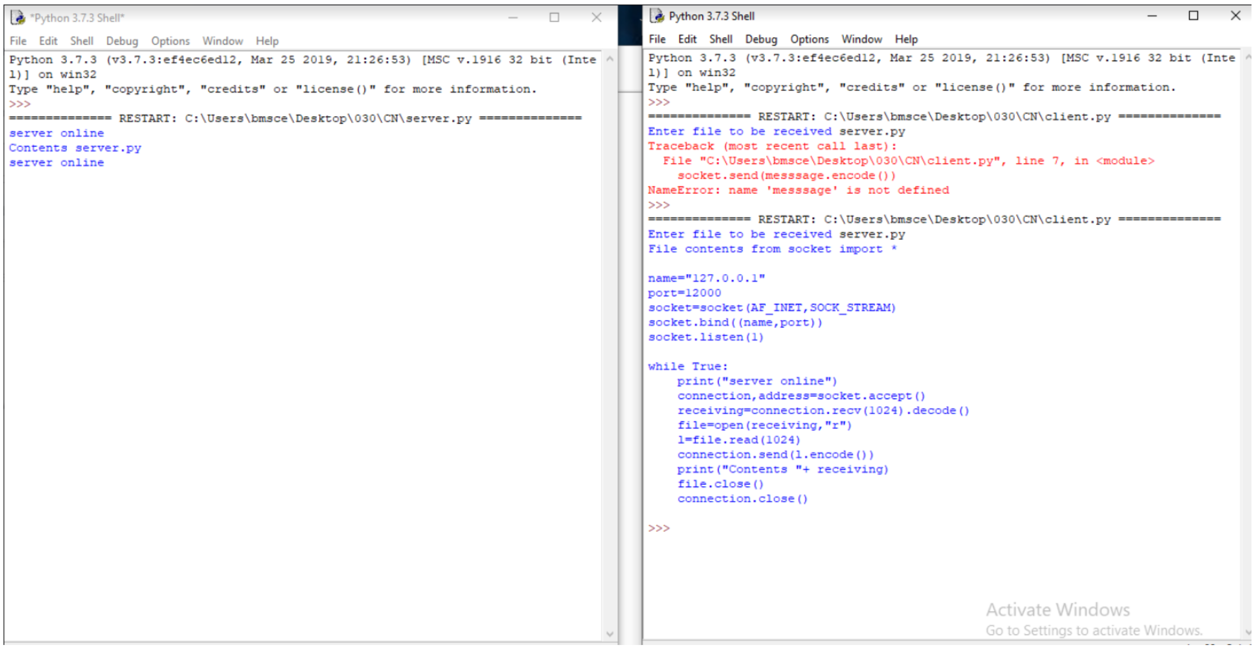
Client:

```
from socket import *

socket=socket(AF_INET,SOCK_STREAM)
socket.connect(("192.168.238.1",3000))
ask=input("Enter file name ")
socket.send(ask.encode())
print(socket.recv(1024).decode())
socket.close()
```

Server:

```
from socket import *

socket=socket(AF_INET,SOCK_STREAM)
socket.bind(("192.168.238.1",3000))
socket.listen()
while True:
    client,address=socket.accept()
    message=client.recv(1024).decode()
    file=open(message,'r')
    l=file.read(1024)
    client.send(l.encode())
    file.close()
    client.close()
```

# Result:


*Python 3.7.3 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\bmsce\Desktop\030\CN\server.py ==============
server online
Contents server.py
server online
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\bmsce\Desktop\030\CN\client.py ==============
Enter file to be received server.py
Traceback (most recent call last):
File "C:\Users\bmsce\Desktop\030\CN\client.py", line 7, in <module>
socket.send(messsage.encode())
NameError: name 'messsage' is not defined
>>>
============== RESTART: C:\Users\bmsce\Desktop\030\CN\client.py ==============
Enter file to be received server.py
File contents from socket import *
name="127.0.0.1"
port=12000
socket=socket(AF_INET,SOCK_STREAM)
socket.bind((name,port))
socket.listen(1)
while True:
print("server online")
connection,address=socket.accept()
receiving=connection.recv(1024).decode()
file=open(receiving,"r")
l=file.read(1024)
connection.send(l.encode())
print("Contents "+ receiving)
file.close()
connection.close()
>>>
Activate Windows
Go to Settings to activate Windows.

# EXPERIMENT- 16

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.

Experiment - 16

Aim: Using UDP packets, write client server program to make client server file name and server. Send back contents of requested file.

Client UDP.Py

```
from socket import *

name="127.0.0.1"
port= 12000
socket = socket (AF_INET, SOCK-DGRAM)
message = input(" Enter filename")
socket.sendto (bytes (message,"utf-8"), (name, port))
filecontents , address = socket.recvfrom(2048)
print ("From Server")
 print( filecontents.decode("utf-8"))
 socket. close()
```

Server UDP .py

```
from socket import *

name= "127.0.0.1"
port= 12000
socket= socket (AF_INET, SOCK_DGRAM)
socket. bind ((name, port))
print (" Server ready"),
while 1:
        message address= socket.recvfrom (2048)
        message= message. decode ("utf-8")
```

Date______ Page______

```
file = open(message, 'r')
conn = file.read(2048)
socket.sendto(bytes(con, "utf-8"), address)
print(message)
file.close().
```

Output:

Server:

Server ready

Server UDP.py

Client:

Enter file name

Server UDP.py

From server

// contents of server UDP.py

## Code:

Client:

```
from socket import *

socket=socket(AF_INET,SOCK_STREAM)

ask=input("Enter file name ")
socket.sendto(bytes(ask,"utf-8"),("192.168.238.1",3000))
file,address=socket.recvfrom(2048)
print(file.decode("utf-8"))
socket.close()
```

Server:

```
from socket import *

socket=socket(AF_INET,SOCK_STREAM)
socket.bind(("192.168.238.1",3000))

while True:
    message,address=socket.recvfrom(2048)
    message=message.decode("utf-8")
    file=open(message,'r')
    l=file.read(2048)
    socket.sendto(bytes(l,"utf-8"),address)
    file.close()
    socket.close()
```

## Result:

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

C:\Users\aravi\OneDrive\Desktop\notes\CN>python client.py
Enter file name server.py
from socket import *

socket=socket(AF_INET,SOCK_DGRAM)
socket.bind(("192.168.238.1",3000))

while True:
    message,address=socket.recvfrom(2048)
    message=message.decode("utf-8")
    file=open(message,'r')
    l=file.read(2048)
    socket.sendto(bytes(l,"utf-8"),address)
    file.close()
    socket.close()


C:\Users\aravi\OneDrive\Desktop\notes\CN>
```

# Wireshark

## Wireshark

~~Wireshark is an open-source~~

Aim: Exploring functionalities of wireshark.

Overview: Wireshark is an open-source application that captures and displays data traveling back and forth on a network. Wireshark is a packet sniffer and network analyser.

Capturing packets:

→ First select a network from which we require to sniff packets.

→ Wireshark begins capturing packets from selected network. All captured packets are shown in the top section of the panel.

→ On selecting a particular packet, we observe the structure of the packet in the middle section of the panel. Various structures can be seen with respect to the protocol.

→ The following packet details are shown

- Time stamp
- Source IP
- Destination IP
- Protocol name
- Length of packet.

Filtering in Wireshark:

Wireshark provides a filter function to better analyse network data. Wireshark also allows creating custom filters.

An example of a filter is to select only packets from HTTP protocol is

Date_____ Page_____

tcp.port ==80 || udp.port ==80

Packet details:

The middle section of the panel, detail panel, presents the protocols and protocol fields of the selected packet in a collapsible format. We can apply additional filters and right on click on protocol ~~to~~ ~~vie~~ for a detailed view.

At the bottom panel, raw data of the selected packet is seen in hexadecimal format. It is called as hex dump. It contains 16 ~~a~~ hexadecimal bytes and 16 ASCII bytes alongside the data offset.

Selecting a specific portion of this raw data automatically highlights its corresponding section in packet details pane. Bytes that cannot be represented is shown as a period ('.')?.