VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS LAB

Submitted by

HARSHAVARDHAN HC (1BM22CS407)

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 **HARSHAVARDHAN HC (1BM22CS407),** 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 - (22CS4PCCON)** work prescribed for the said degree.

Name of the Lab-Incharge:
Swathi Sridharan
Designation: Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak

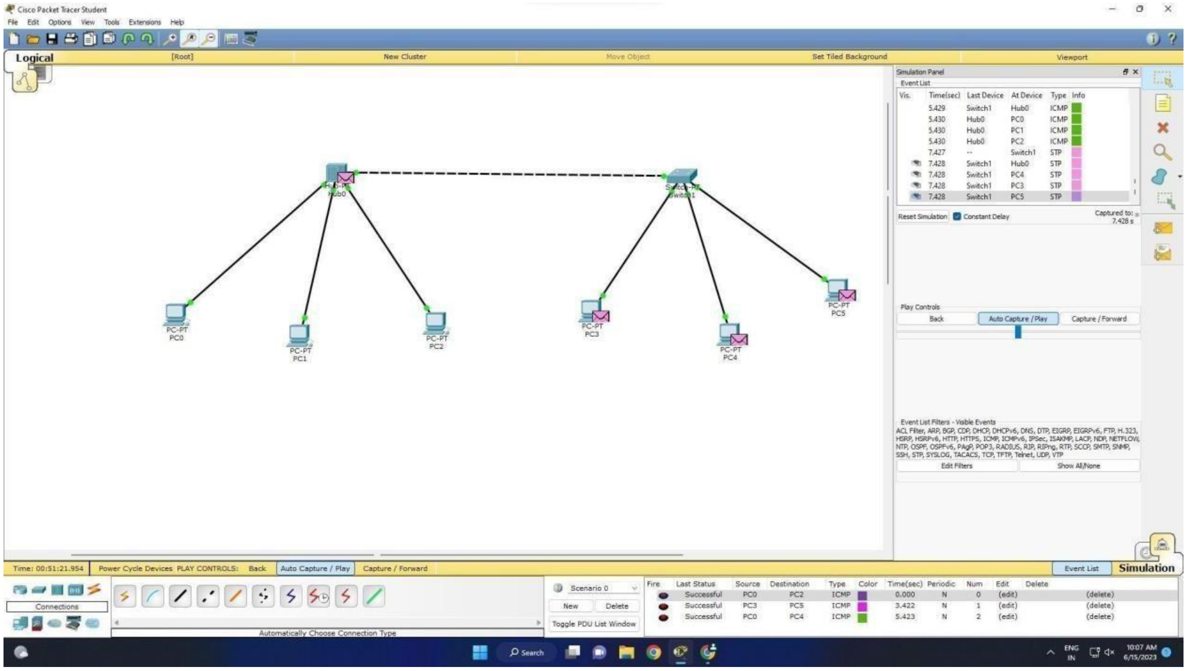
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

LAB-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.

TOPOLOGY:



Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	5.429	Switch1	Hub0	ICMP	🟩
	5.430	Hub0	PC0	ICMP	🟩
	5.430	Hub0	PC1	ICMP	🟩
	5.430	Hub0	PC2	ICMP	🟩
	7.427	--	Switch1	STP	🟪
👁	7.428	Switch1	Hub0	STP	🟪
👁	7.428	Switch1	PC4	STP	🟪
👁	7.428	Switch1	PC3	STP	🟪
👁	7.428	Switch1	PC5	STP	🟪

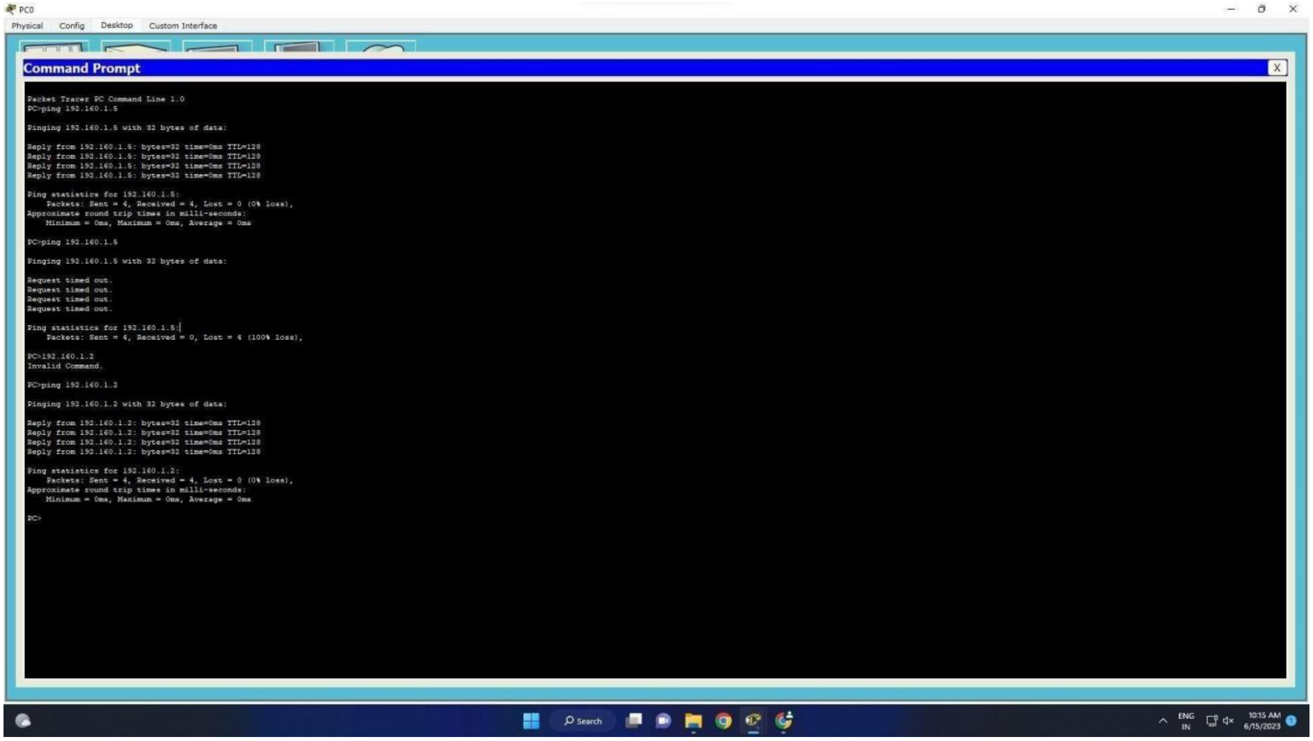
☑ Constant Delay

Captured to:
7.428 s

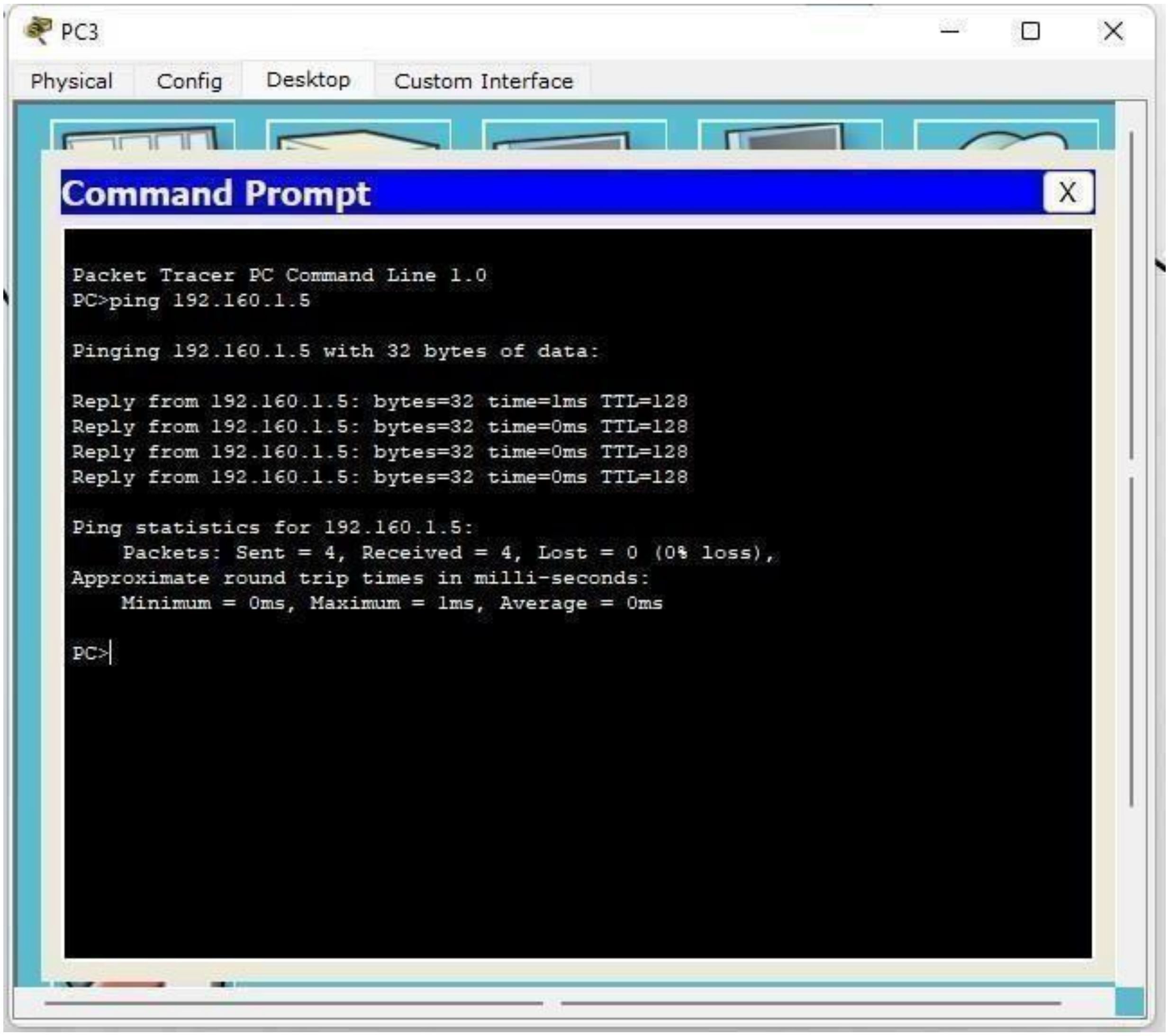
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC2	ICMP	🟪	0.000	N	0	(edit)	(delete)
●	Successful	PC3	PC5	ICMP	🟪	3.422	N	1	(edit)	(delete)
●	Successful	PC0	PC4	ICMP	🟩	5.423	N	2	(edit)	(delete)

OUTPUT:



```
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5

Pinging 192.160.1.5 with 32 bytes of data:

Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128

Ping statistics for 192.160.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 192.160.1.5

Pinging 192.160.1.5 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.160.1.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>192.160.1.2
Invalid Command.

PC>ping 192.160.1.2

Pinging 192.160.1.2 with 32 bytes of data:

Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128

Ping statistics for 192.160.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```



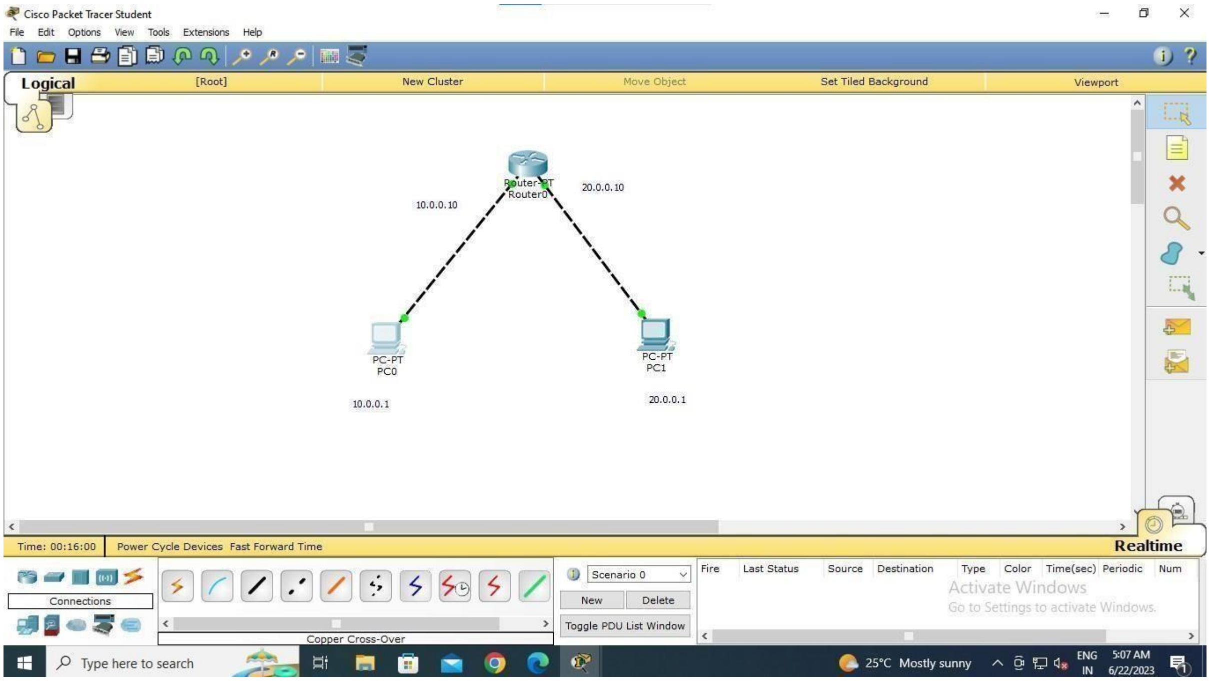
```
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5

Pinging 192.160.1.5 with 32 bytes of data:

Reply from 192.160.1.5: bytes=32 time=1ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128

Ping statistics for 192.160.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

LAB 2

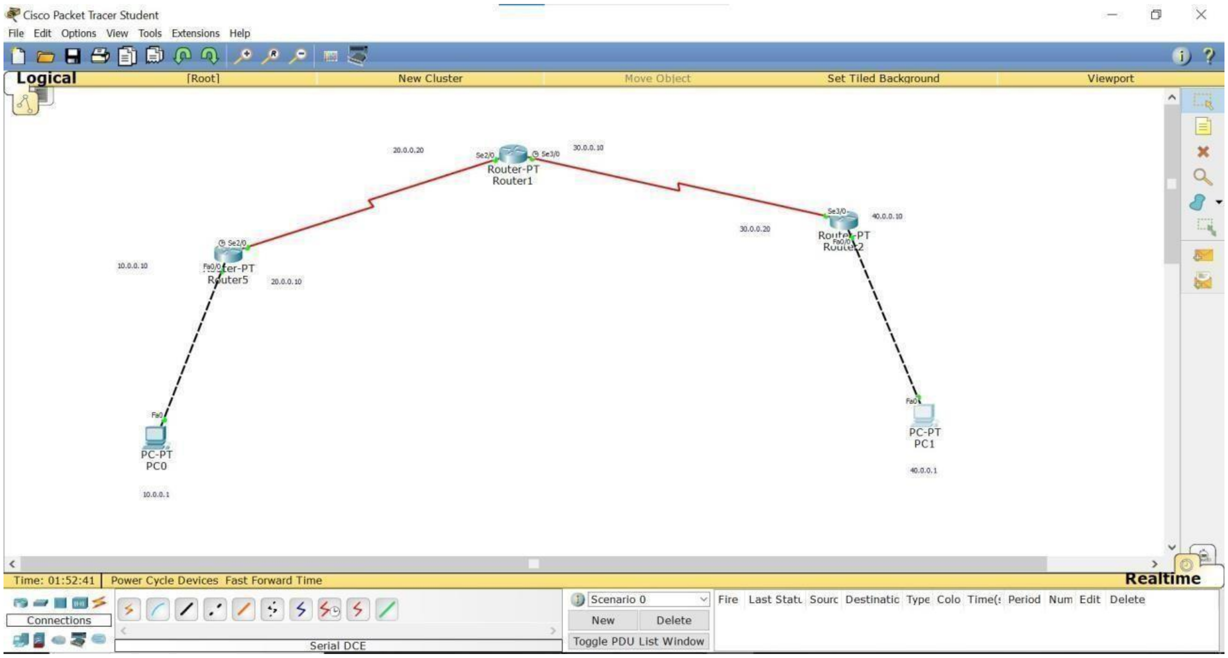
Configure IP address to routers (one and three) in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

TOPOLOGY:

PROGRAM 2.1

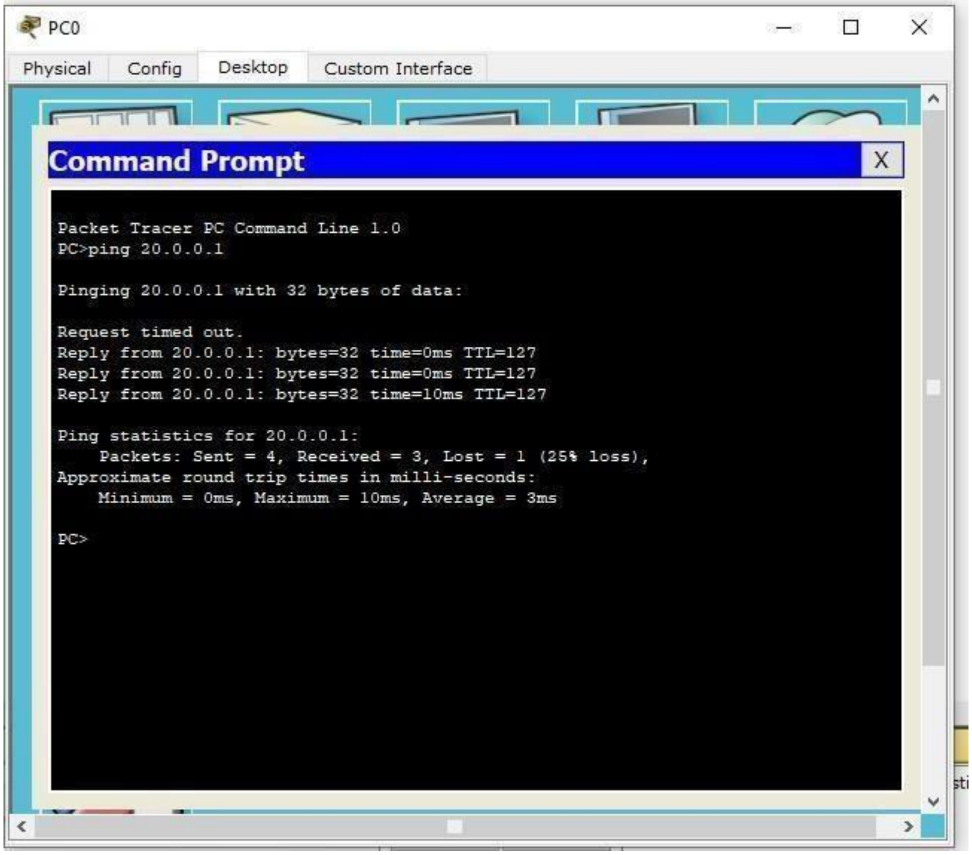


PROGRAM 2.2


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
Se2/0
Se3/0
30.0.0.10
Router-PT
Router1
Se3/0
40.0.0.10
30.0.0.20
Router-PT
Fa0/0
Router2
Se2/0
10.0.0.10
Fa0/0
Router-PT
Router5
20.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1
Time: 01:52:41
Power Cycle Devices Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Statu
Sourc
Destinatic
Type
Colo
Time(s
Period
Num
Edit
Delete

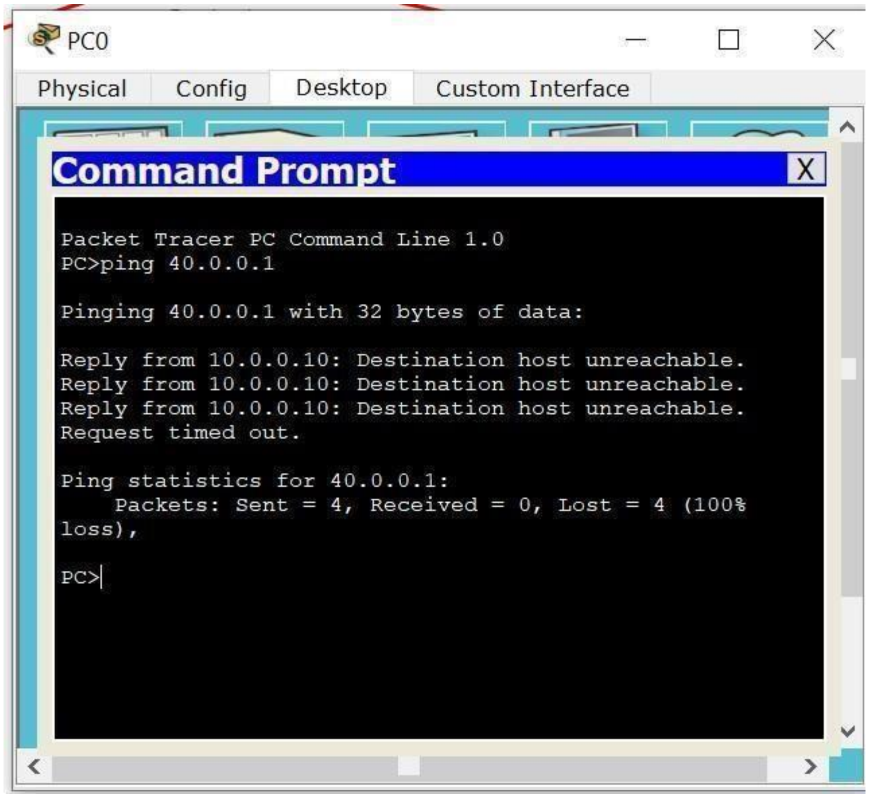
OUTPUT:

PROGRAM 2.1

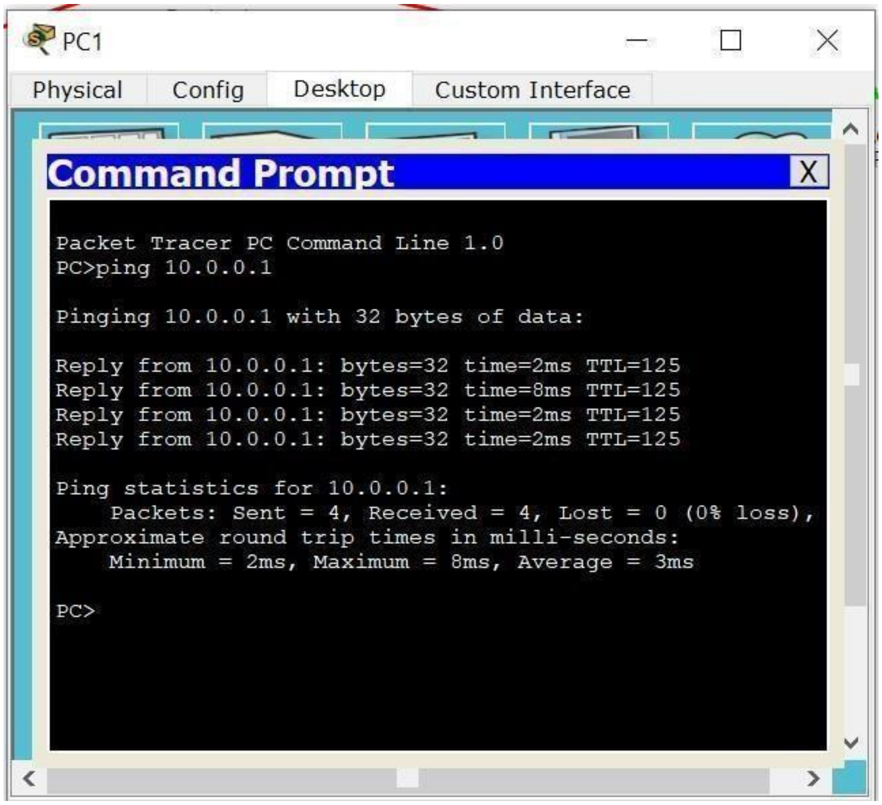


```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=10ms 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 = 10ms, Average = 3ms

PC>
```

PROGRAM 2.2



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

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Request timed out.

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100%
loss),

PC>
```

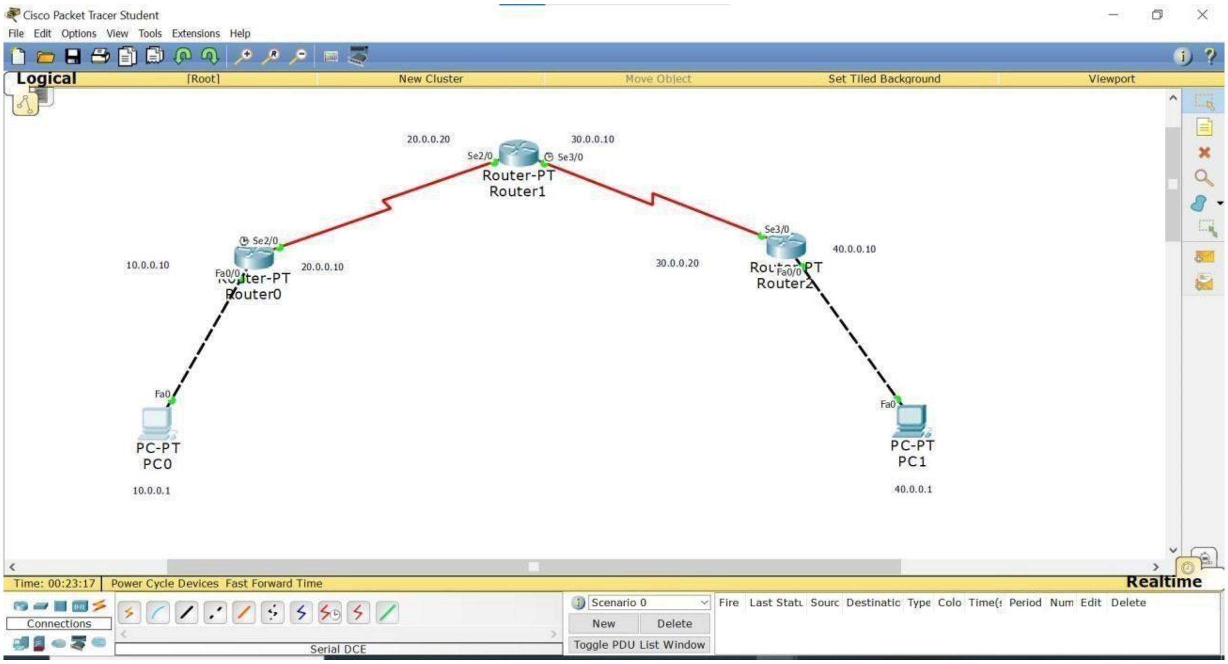


```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 3ms

PC>
```

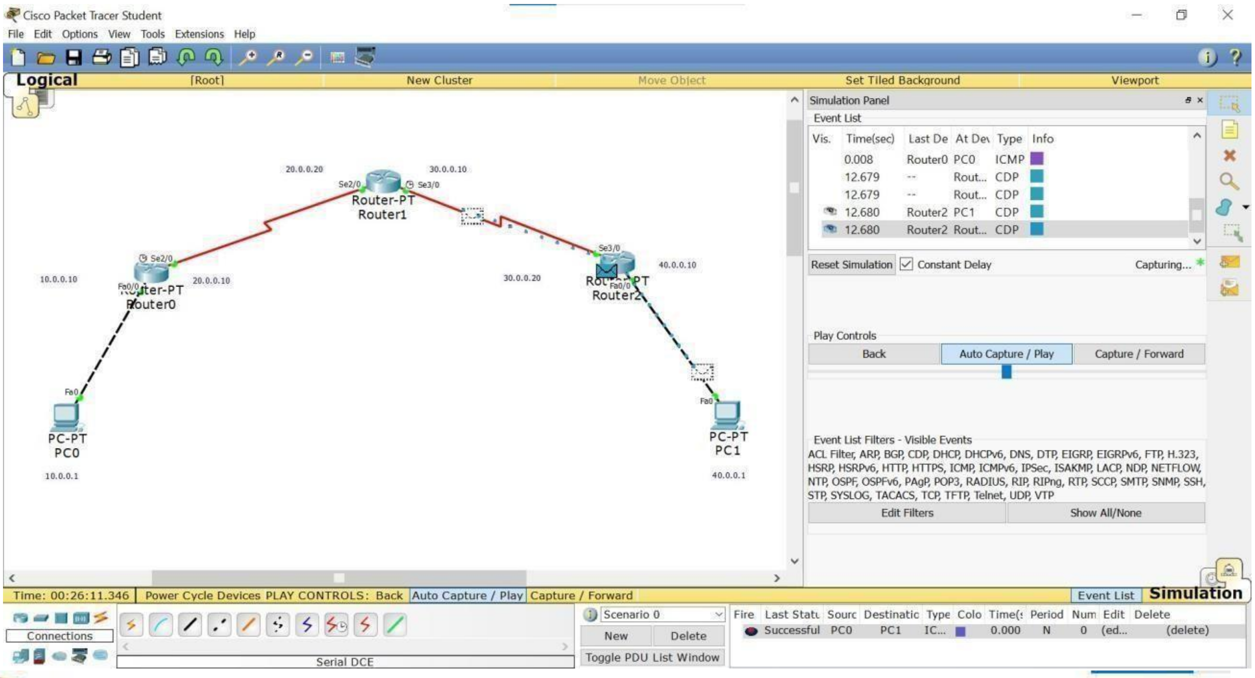
LAB 3

Configure default route to the Router.

TOPOLOGY:



OUTPUT:



Vis.	Time(sec)	Last De	At Dev	Type	Info
	0.008	Router0	PC0	ICMP	🟪
	12.679	--	Rout...	CDP	🟦
	12.679	--	Rout...	CDP	🟦
👁	12.680	Router2	PC1	CDP	🟦
👁	12.680	Router2	Rout...	CDP	🟦

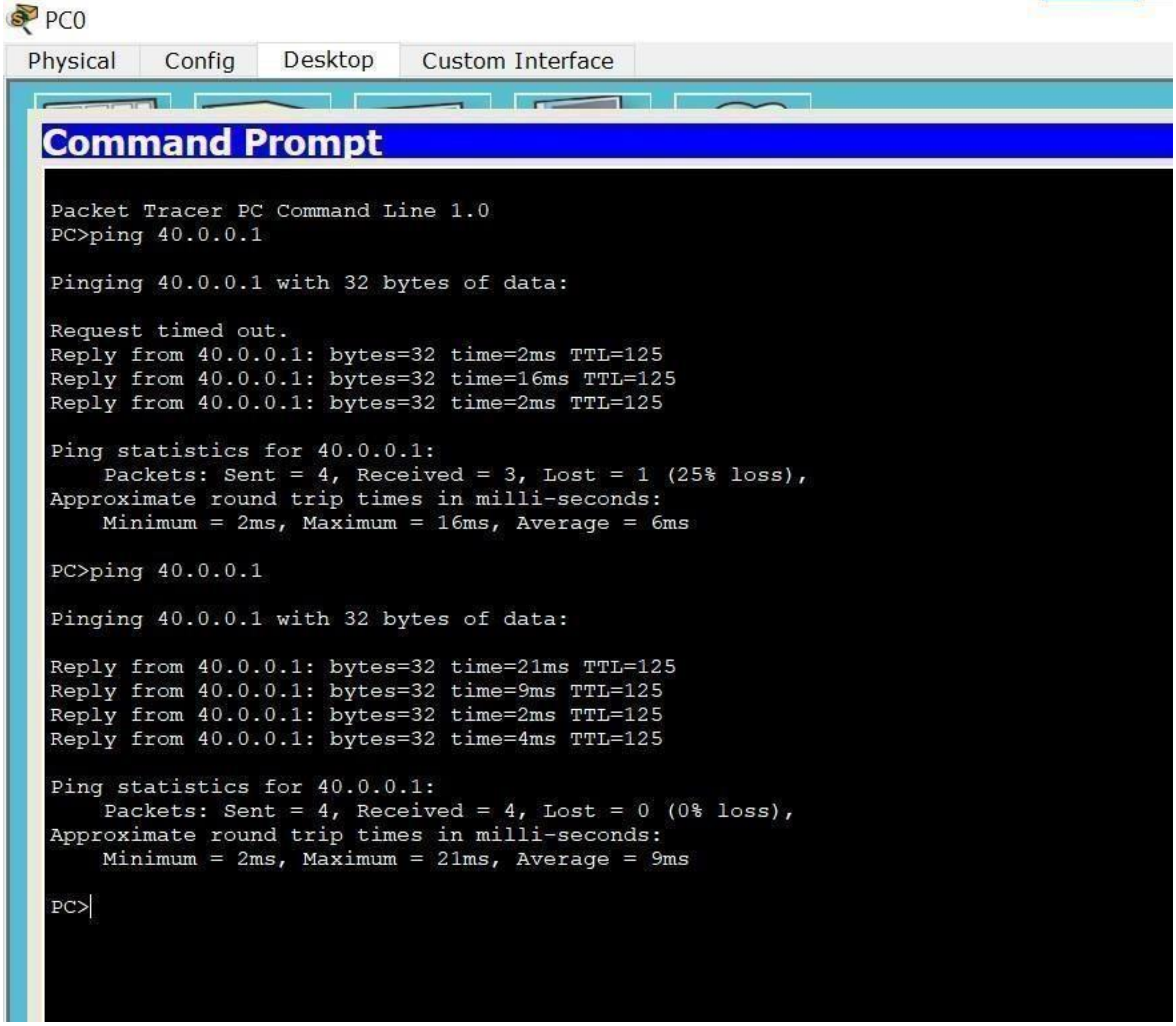
☑ Constant Delay

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

Time: 00:26:11.346 Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward

Scenario 0

Fire	Last Statu	Sourc	Destinatic	Type	Colo	Time(s	Period	Num	Edit	Delete
●	Successful	PC0	PC1	IC...	🟪	0.000	N	0	(ed...	(delete)



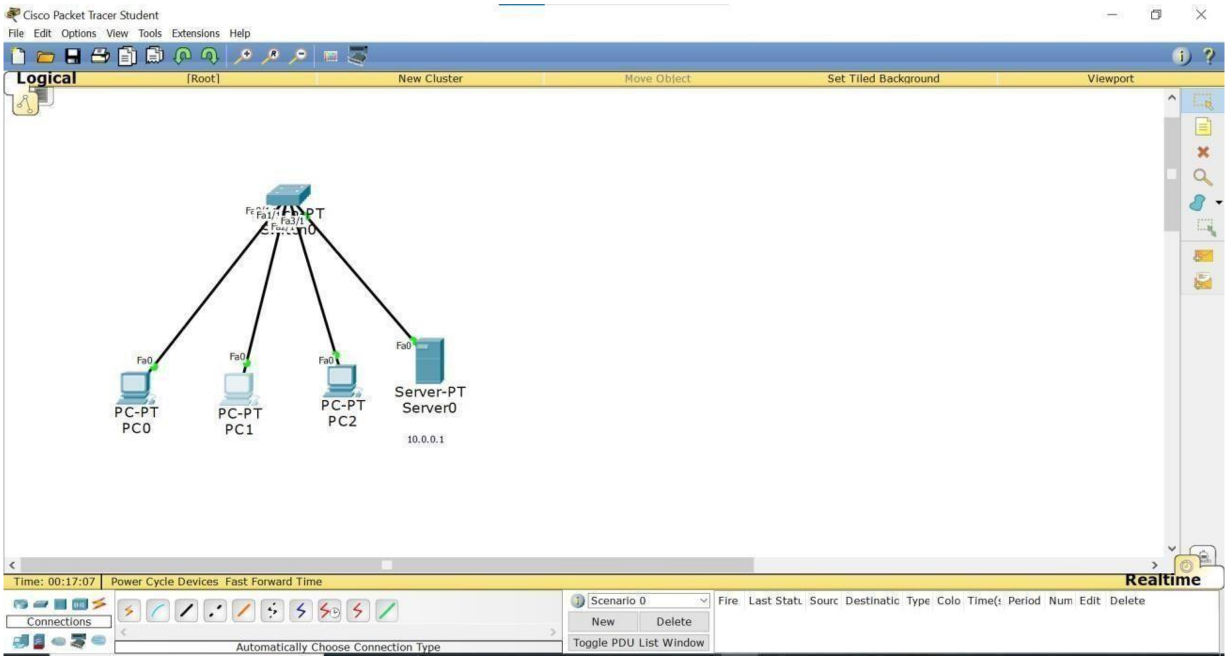
```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

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

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

PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

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

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

PC>
```

LAB 4

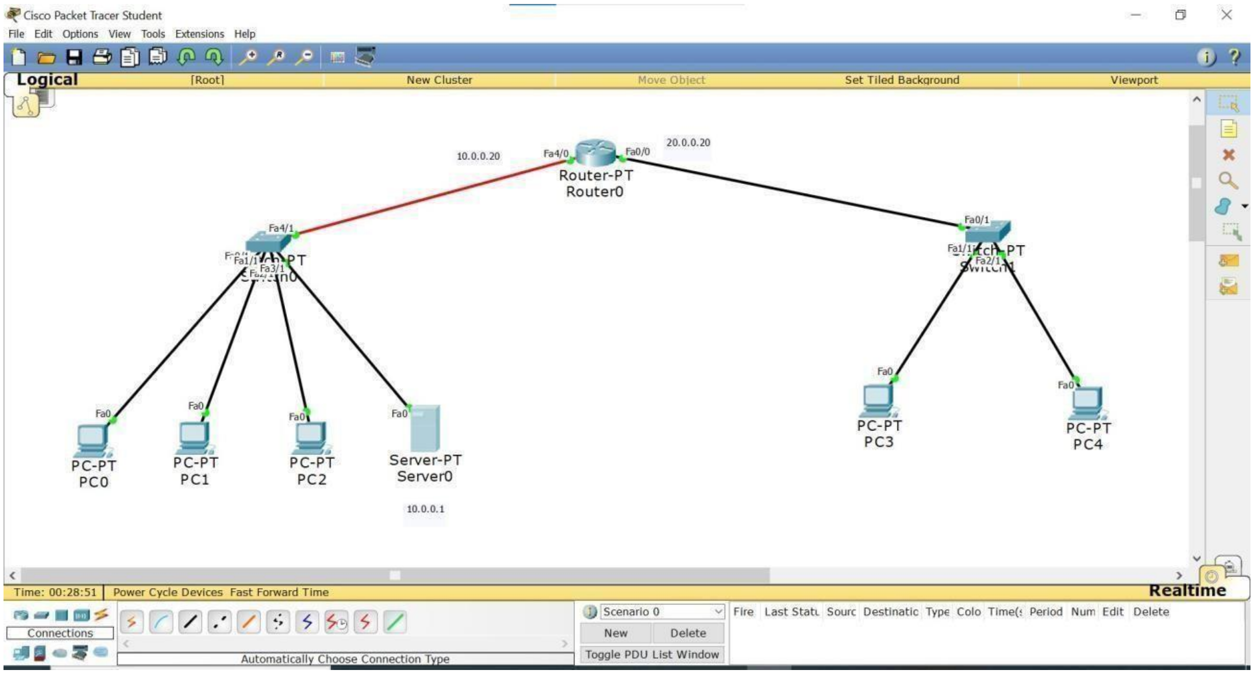
Configure DHCP within a LAN and outside LAN.

TOPOLOGY:

PROGRAM 4.1:

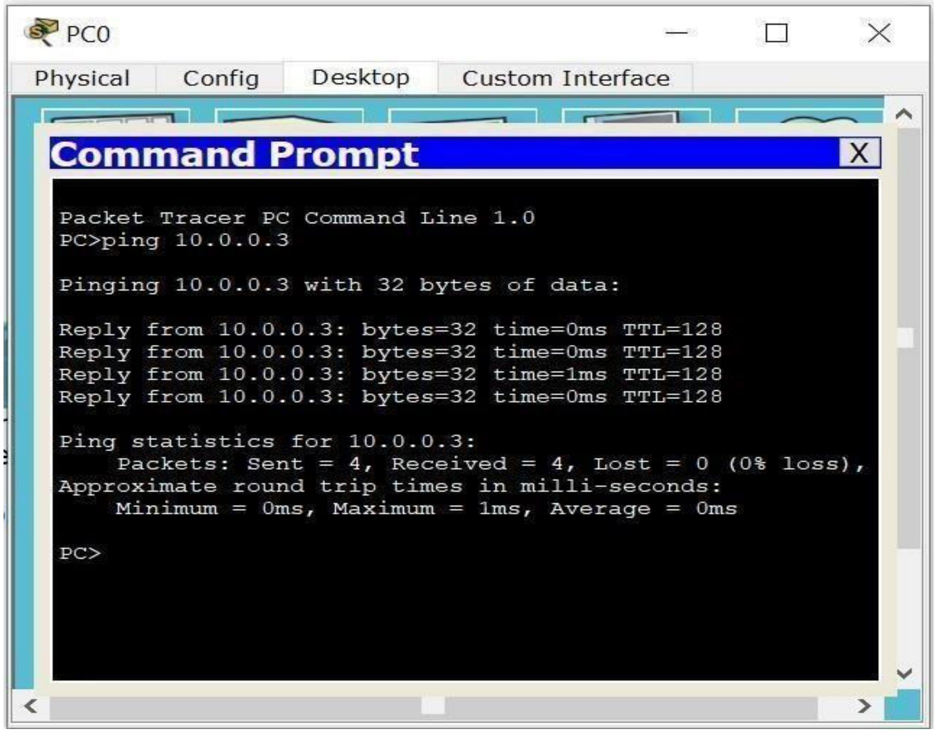


PROGRAM 4.2:

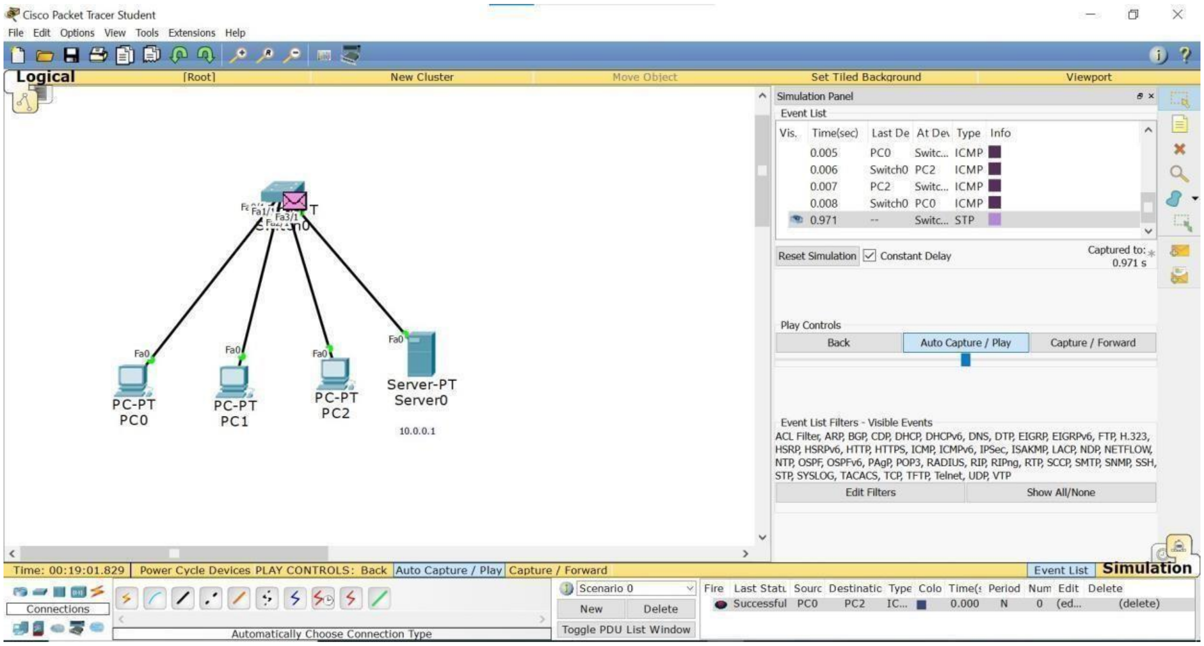


OUTPUT:

PROGRAM 4.1:



```
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=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms 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 = 1ms, Average = 0ms

PC>
```



Vis.	Time(sec)	Last De	At Dev	Type	Info
	0.005	PC0	Switc...	ICMP	■
	0.006	Switch0	PC2	ICMP	■
	0.007	PC2	Switc...	ICMP	■
	0.008	Switch0	PC0	ICMP	■
	0.971	--	Switc...	STP	■

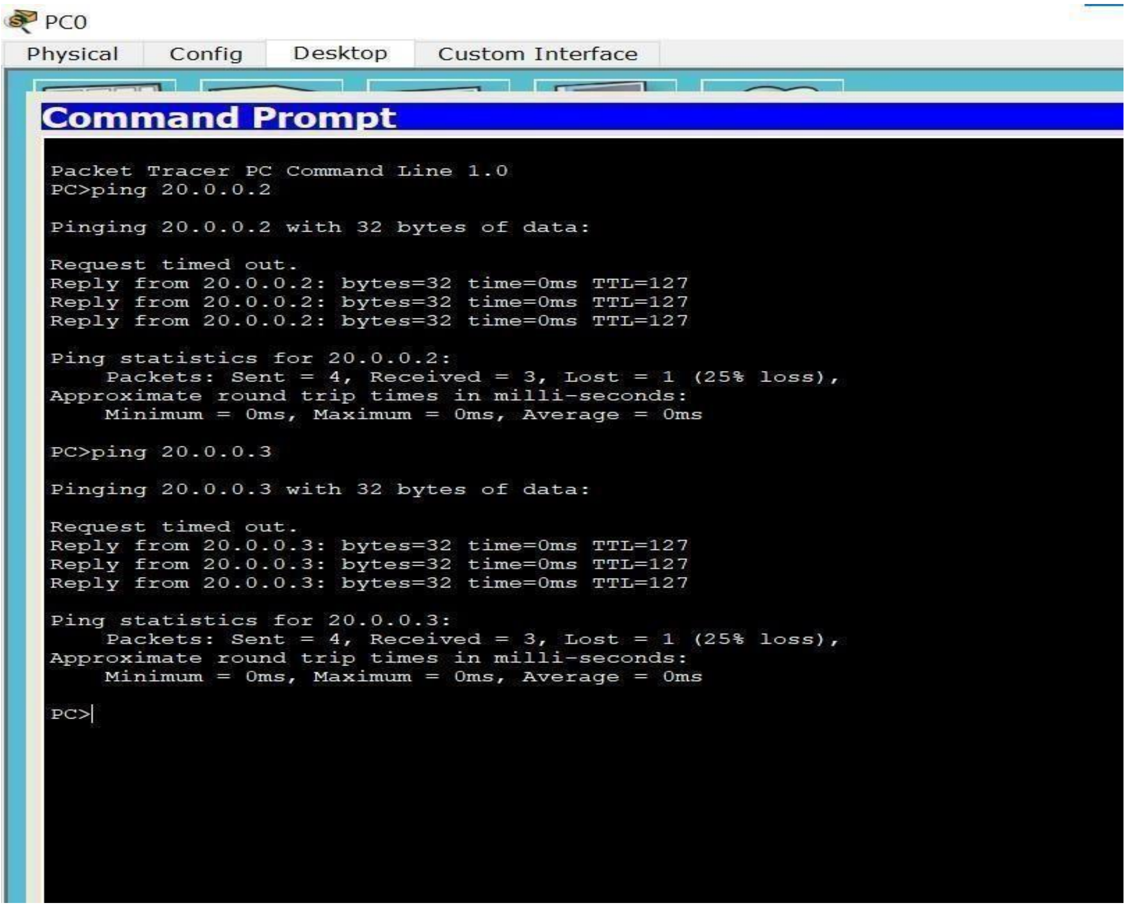
Captured to: 0.971 s

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

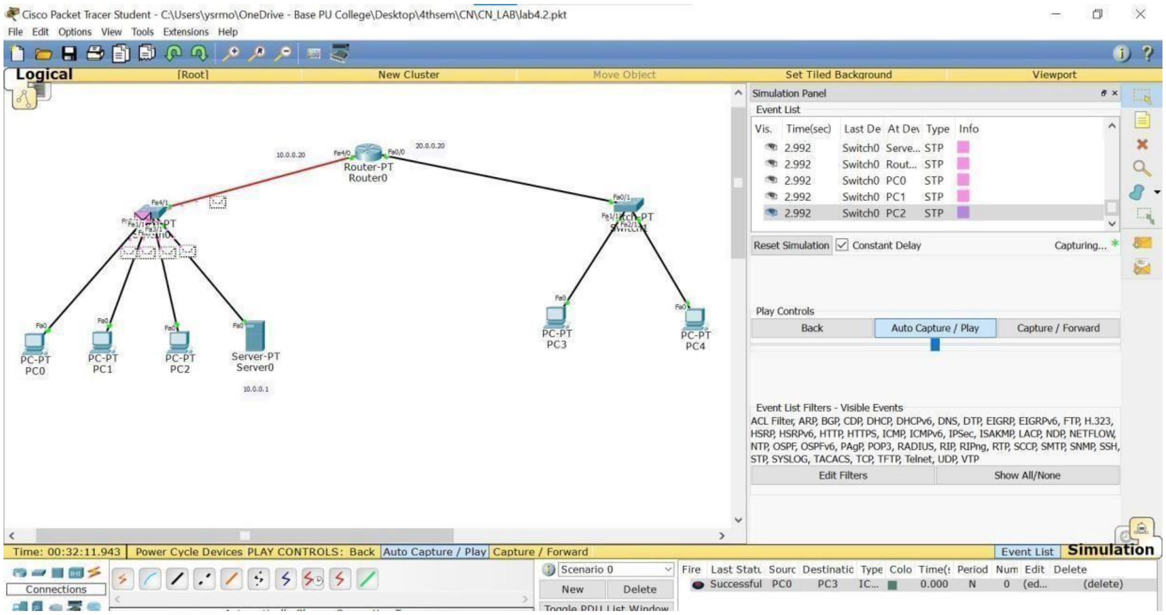
Time: 00:19:01.829

Fire	Last Statu	Sourc	Destinatic	Type	Colo	Time(s	Period	Num	Edit	Delete
●	Successful	PC0	PC2	IC...	■	0.000	N	0	(ed...	(delete)

PROGRAM 4.2:



```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

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

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab4.2.pkt

Simulation Panel

Event List

Vis.	Time(sec)	Last De	At Dev	Type	Info
	2.992	Switch0	Serve...	STP	
	2.992	Switch0	Rout...	STP	
	2.992	Switch0	PC0	STP	
	2.992	Switch0	PC1	STP	
	2.992	Switch0	PC2	STP	

Reset Simulation ☑ Constant Delay Capturing...

Play Controls

Back | Auto Capture / Play | Capture / Forward

Event List Filters - Visible Events

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

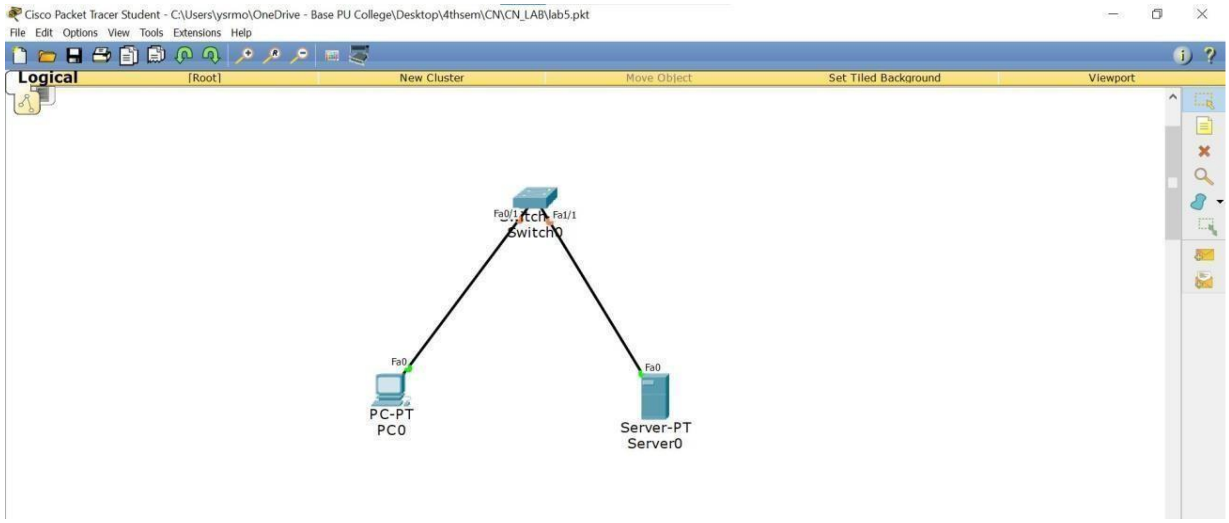
Time: 00:32:11.943 | Power Cycle Devices PLAY CONTROLS: Back | Auto Capture / Play | Capture / Forward

Fire	Last Statu	Sourc	Destinatic	Type	Colo	Time(s	Period	Num	Edit	Delete
●	Successful	PC0	PC3	IC...	🟩	0.000	N	0	(ed...	(delete)

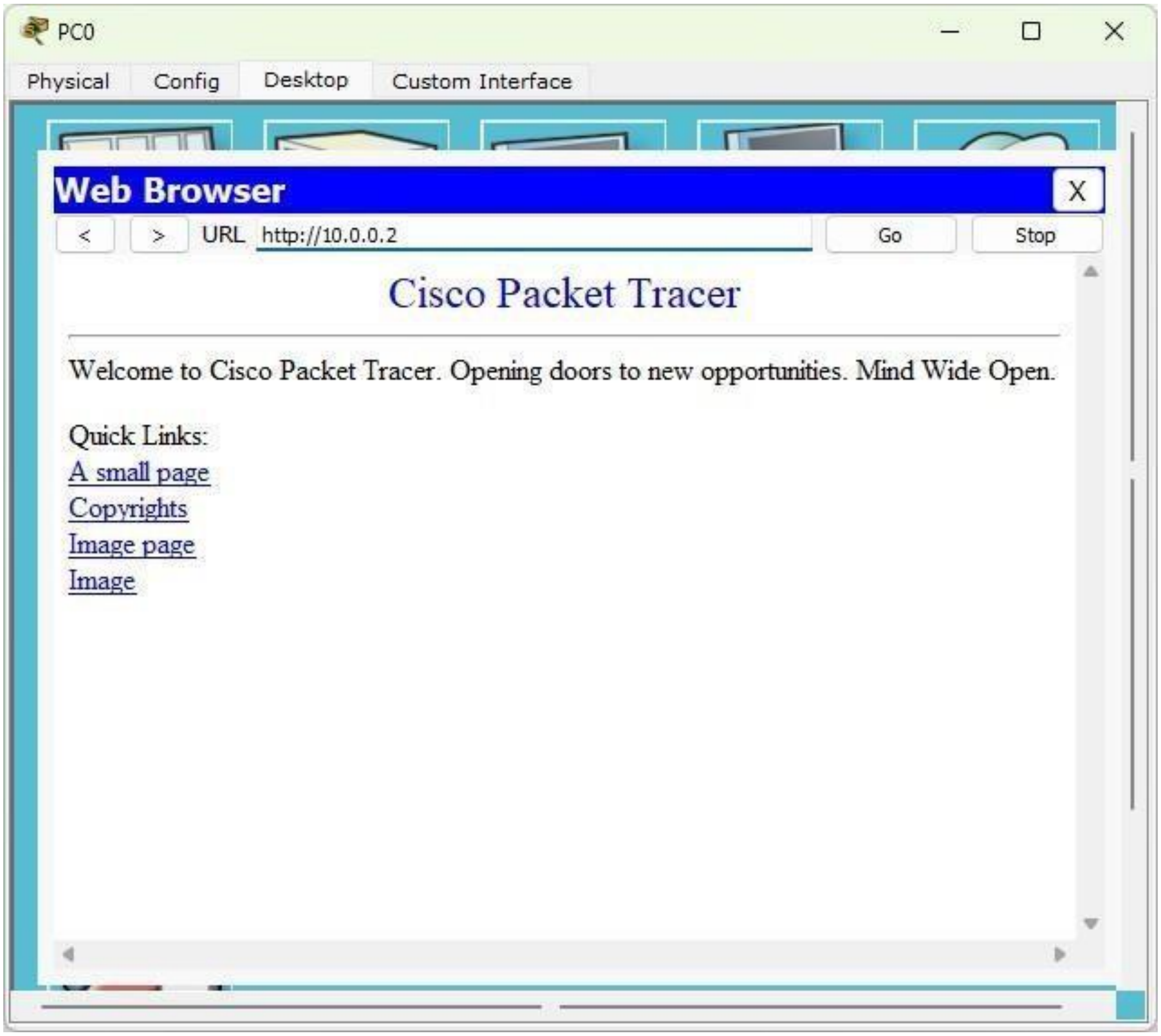
LAB 5

Configure Web Server, DNS within aLAN.

TOPOLOGY:


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab5.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/1
Fa1/1
Switch
Switch0
Fa0
PC-PT
PC0
Fa0
Server-PT
Server0

OUTPUT:



http://10.0.0.2

Cisco Packet Tracer

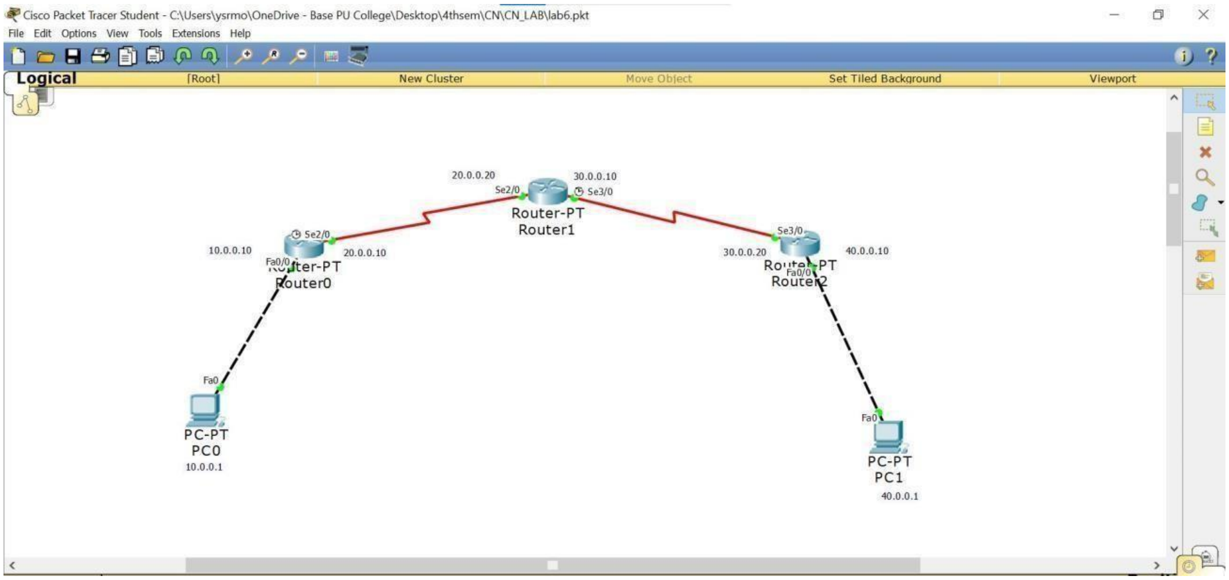
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

Quick Links:
A small page
Copyrights
Image page
Image

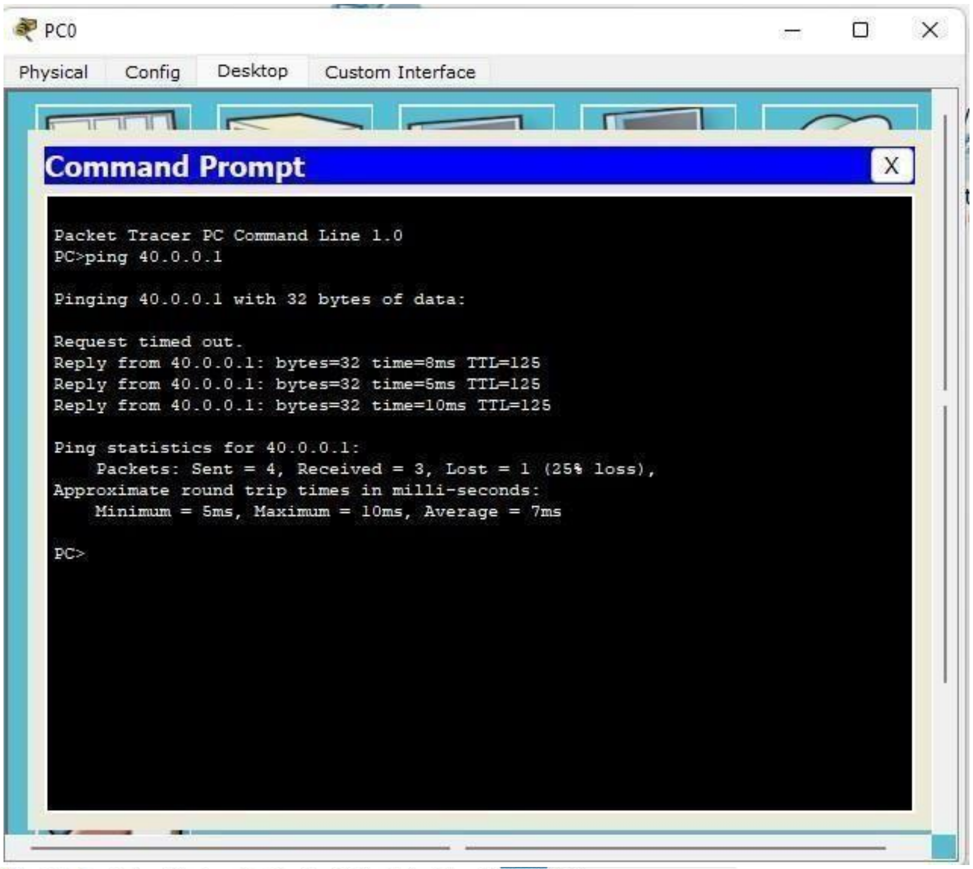
LAB 6

Configure RIP routing Protocol in Routers.

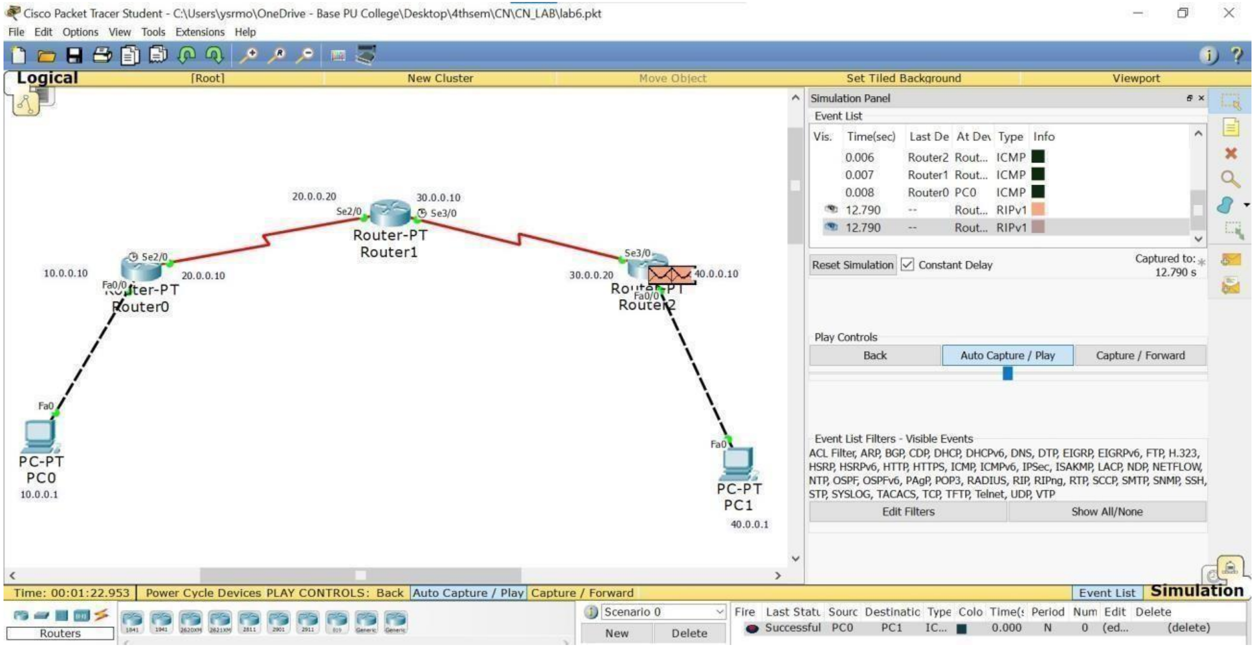
TOPOLOGY:



OUTPUT:



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

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=5ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms 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 = 5ms, Maximum = 10ms, Average = 7ms

PC>
```


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab6.pkt

Vis.	Time(sec)	Last De	At Dev	Type	Info
	0.006	Router2	Rout...	ICMP	■
	0.007	Router1	Rout...	ICMP	■
	0.008	Router0	PC0	ICMP	■
👁	12.790	--	Rout...	RIPv1	■
👁	12.790	--	Rout...	RIPv1	■

☑ Constant Delay

Captured to: 12.790 s

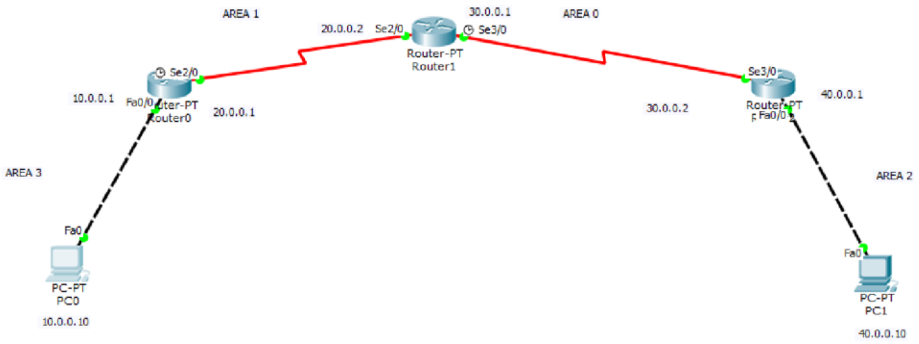
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Fire	Last Statu	Sourc	Destinatic	Type	Colo	Time(s	Period	Num	Edit	Delete
●	Successful	PC0	PC1	IC...	■	0.000	N	0	(ed...	(delete)

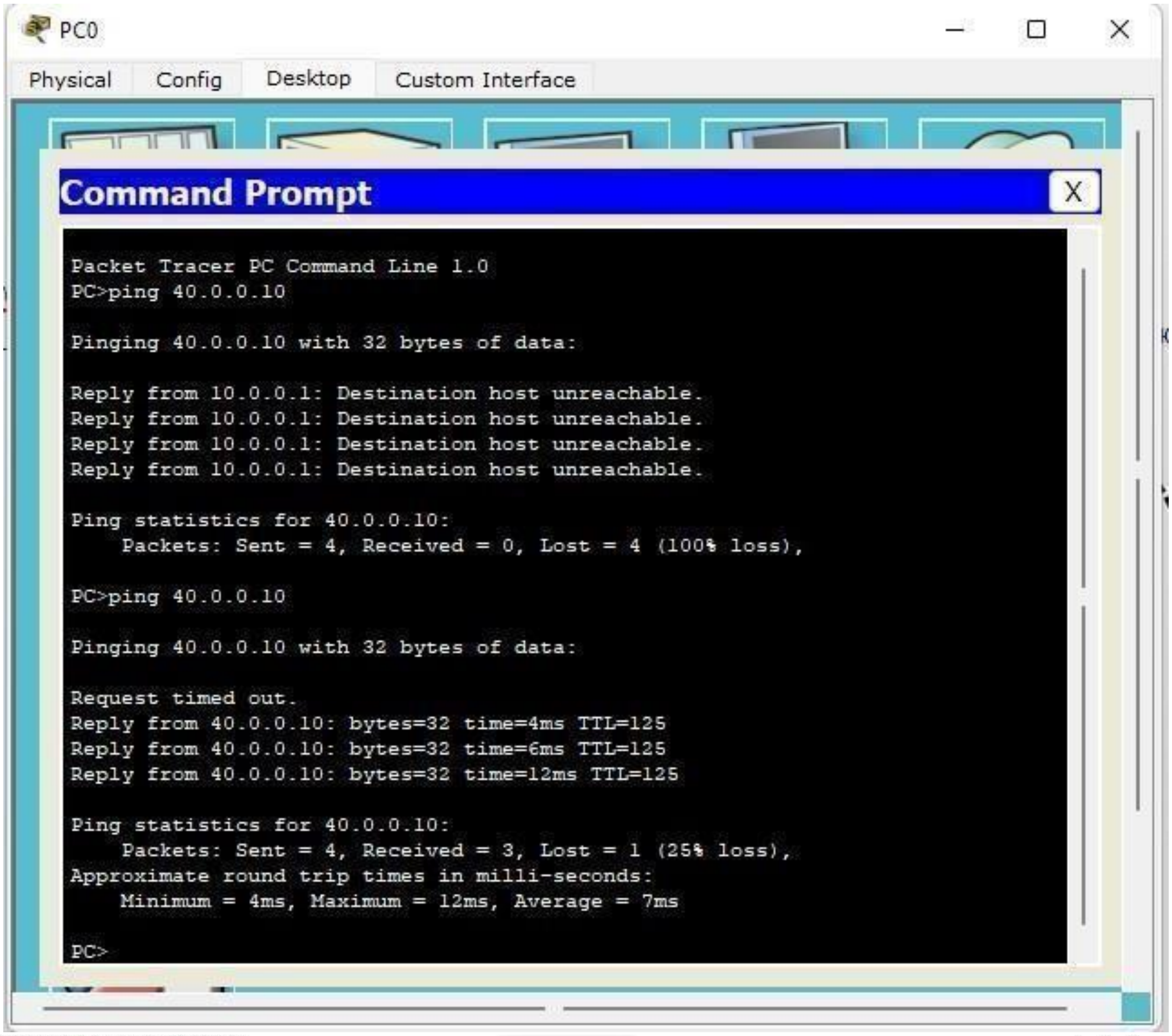
LAB 7

Configure OSPF routing protocol.

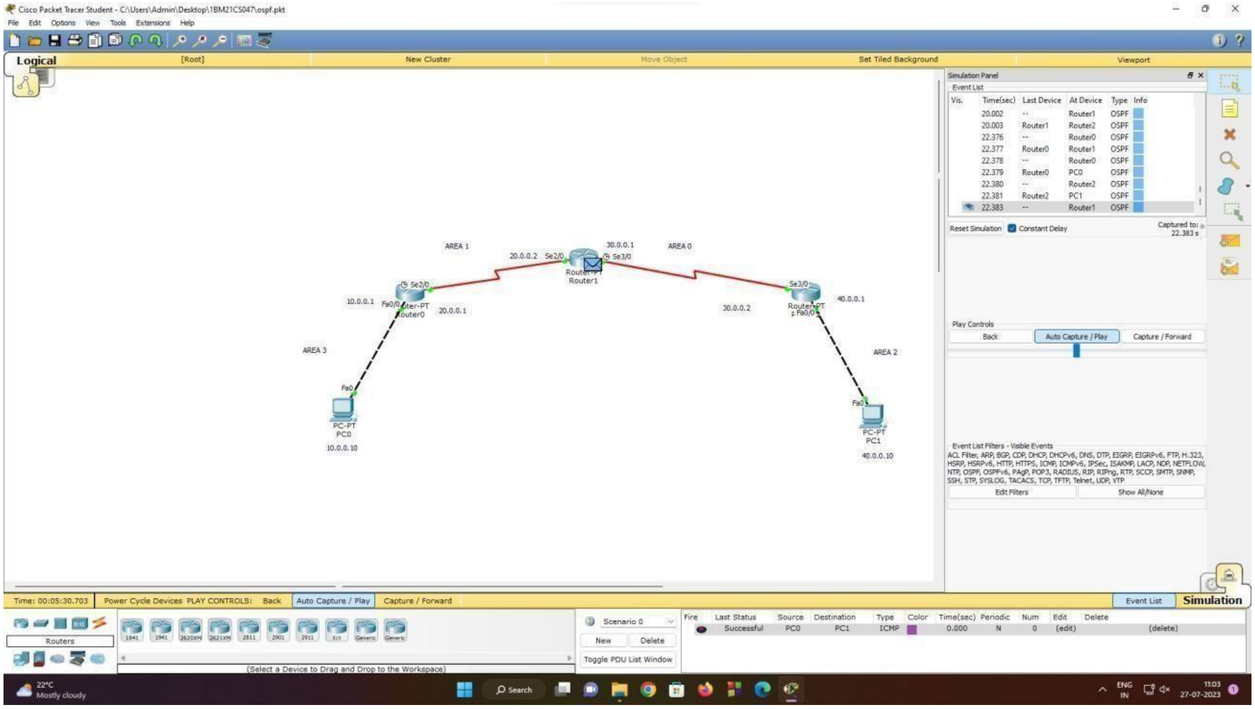
TOPOLOGY:



OUTPUT:



```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

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=4ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=12ms 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 = 4ms, Maximum = 12ms, Average = 7ms

PC>
```



Vis.	Time(sec)	Last Device	At Device	Type	Info
	20.002	--	Router1	OSPF	🟦
	20.003	Router1	Router2	OSPF	🟦
	22.376	--	Router0	OSPF	🟦
	22.377	Router0	Router1	OSPF	🟦
	22.378	--	Router0	OSPF	🟦
	22.379	Router0	PC0	OSPF	🟦
	22.380	--	Router2	OSPF	🟦
	22.381	Router2	PC1	OSPF	🟦
	22.383	--	Router1	OSPF	🟦

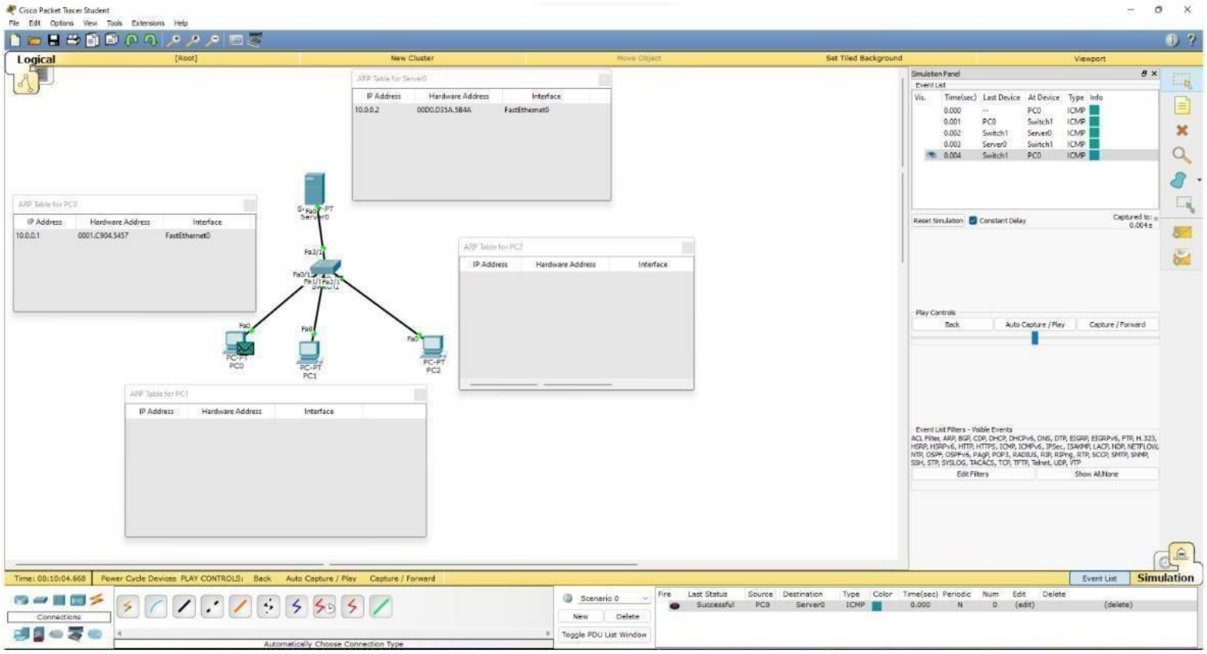
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	PC1	ICMP	🟪	0.000	N	0	(edit)	(delete)

LAB 8

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

TOPOLOGY:



ARP Table for Server0

IP Address	Hardware Address	Interface
10.0.0.2	00D0.D35A.5B4A	FastEthernet0

ARP Table for PC0

IP Address	Hardware Address	Interface
10.0.0.1	0001.C904.5457	FastEthernet0

ARP Table for PC2

IP Address	Hardware Address	Interface

ARP Table for PC1

IP Address	Hardware Address	Interface

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	🟩
	0.001	PC0	Switch1	ICMP	🟩
	0.002	Switch1	Server0	ICMP	🟩
	0.003	Server0	Switch1	ICMP	🟩
	0.004	Switch1	PC0	ICMP	🟩

☑ Constant Delay

Captured to: 0.004 s

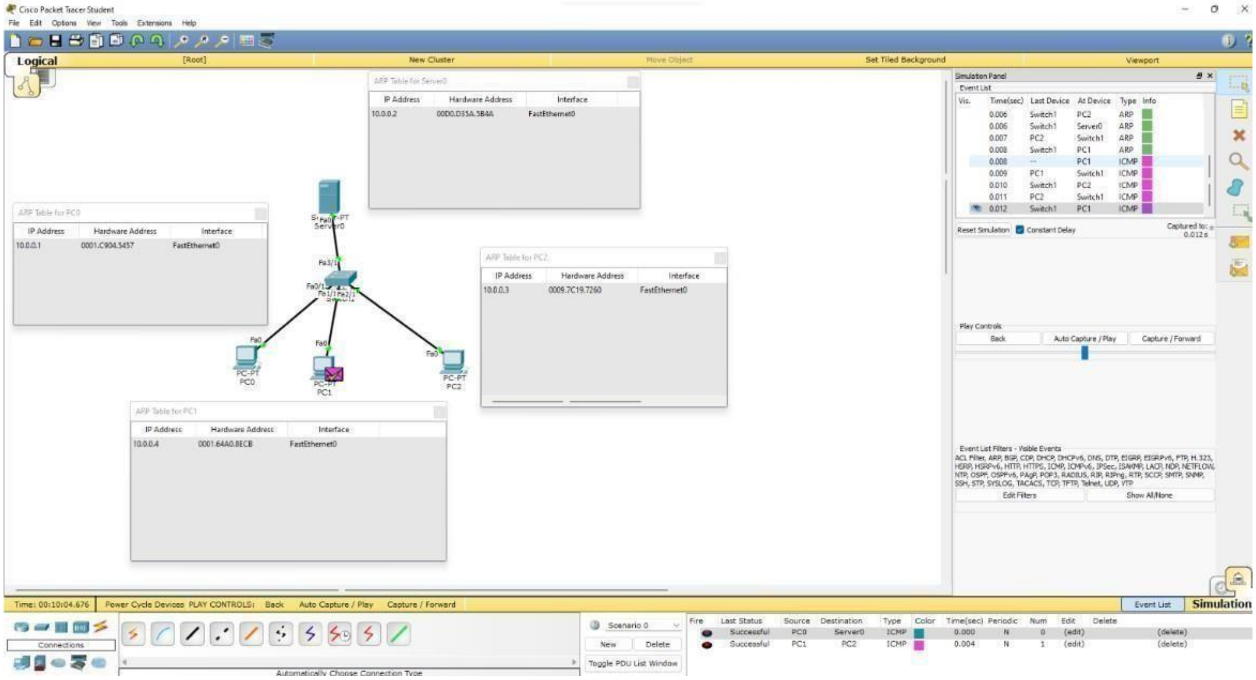
Event List Filters - Visible Events

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

Time: 00:10:04.668

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Server0	ICMP	🟦	0.000	N	0	(edit)	(delete)

OUTPUT:



ARP Table for Server0

IP Address	Hardware Address	Interface
10.0.0.2	00D0.D35A.5B4A	FastEthernet0

ARP Table for PC0

IP Address	Hardware Address	Interface
10.0.0.1	0001.C904.5457	FastEthernet0

ARP Table for PC2

IP Address	Hardware Address	Interface
10.0.0.3	0009.7C19.7260	FastEthernet0

ARP Table for PC1

IP Address	Hardware Address	Interface
10.0.0.4	0001.64A0.8ECB	FastEthernet0

Simulation Panel

Event List

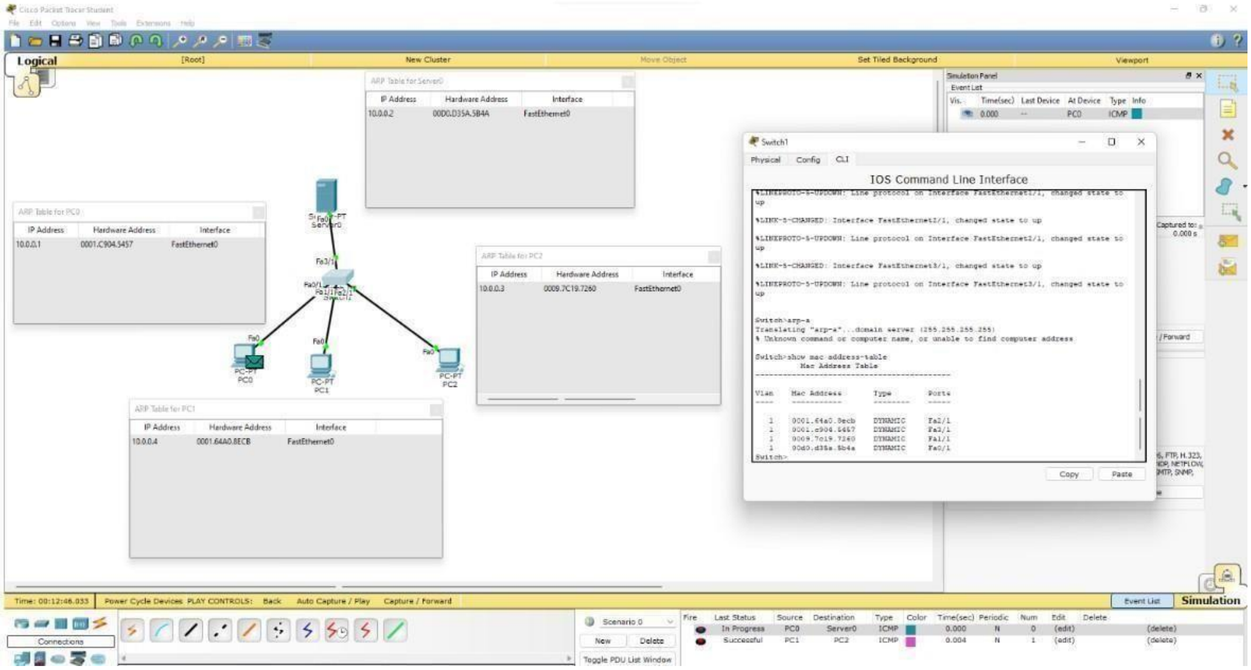
Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.006	Switch1	PC2	ARP	🟩
	0.006	Switch1	Server0	ARP	🟩
	0.007	PC2	Switch1	ARP	🟩
	0.008	Switch1	PC1	ARP	🟩
	0.008	--	PC1	ICMP	🟪
	0.009	PC1	Switch1	ICMP	🟪
	0.010	Switch1	PC2	ICMP	🟪
	0.011	PC2	Switch1	ICMP	🟪
	0.012	Switch1	PC1	ICMP	🟪

Captured to: 0.012 s

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

Time: 00:10:04.676

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Server0	ICMP	🟦	0.000	N	0	(edit)	(delete)
●	Successful	PC1	PC2	ICMP	🟪	0.004	N	1	(edit)	(delete)



ARP Table for Server0

IP Address	Hardware Address	Interface
10.0.0.2	00D0.D35A.5B4A	FastEthernet0

ARP Table for PC0

IP Address	Hardware Address	Interface
10.0.0.1	0001.C904.5457	FastEthernet0

ARP Table for PC2

IP Address	Hardware Address	Interface
10.0.0.3	0009.7C19.7260	FastEthernet0

ARP Table for PC1

IP Address	Hardware Address	Interface
10.0.0.4	0001.64A0.8ECB	FastEthernet0

Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	🟦

Captured to: 0.000 s

```
%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>arp-a
Translating "arp-a"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address

Switch>show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.64a0.8ecb    DYNAMIC     Fa2/1
   1    0001.c904.5457    DYNAMIC     Fa3/1
   1    0009.7c19.7260    DYNAMIC     Fa1/1
   1    00d0.d35a.5b4a    DYNAMIC     Fa0/1
Switch>
```

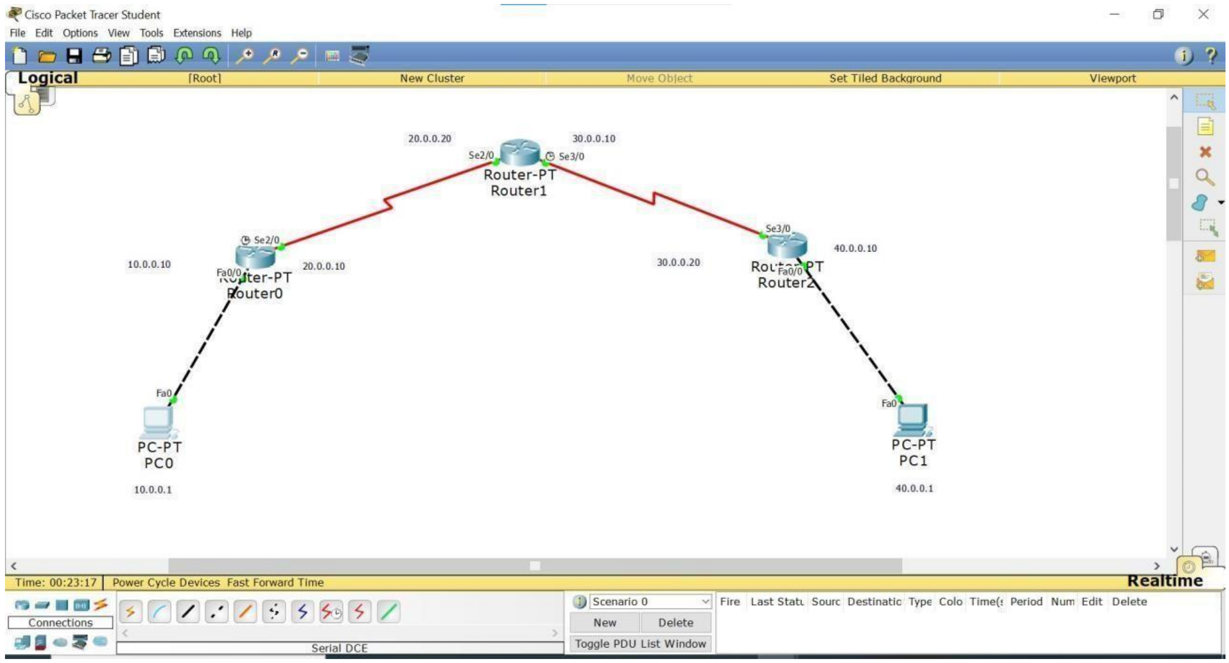
Time: 00:12:46.033

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	In Progress	PC0	Server0	ICMP	🟦	0.000	N	0	(edit)	(delete)
●	Successful	PC1	PC2	ICMP	🟪	0.004	N	1	(edit)	(delete)

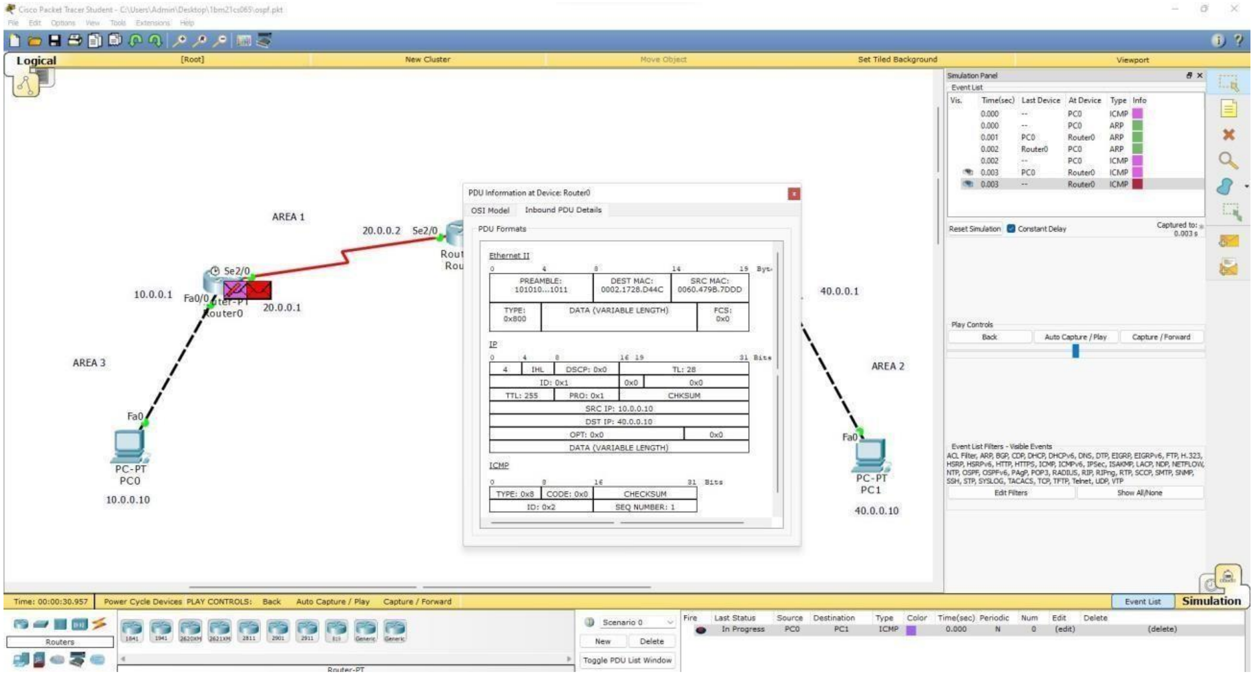
LAB 9

Demonstrate the TTL/ Life of a Packet.

TOPOLOGY:



OUTPUT:



PDU Information at Device: Router0

OSI Model | Inbound PDU Details

PDU Formats

```
Ethernet II
0            4            8           14          19  Bytes
| PREAMBLE:         | DEST MAC:       | SRC MAC:        |
| 101010...1011     | 0002.1728.D44C  | 0060.479B.7DDD  |
| TYPE:  | DATA (VARIABLE LENGTH)              | FCS:     |
| 0x800  |                                     | 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: 10.0.0.10                                    |
| DST IP: 40.0.0.10                                    |
| OPT: 0x0                             | 0x0           |
| DATA (VARIABLE LENGTH)                               |

ICMP
0           8           16                31 Bits
| TYPE: 0x8 | CODE: 0x0 | CHECKSUM          |
| ID: 0x2               | SEQ NUMBER: 1     |
```

Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	🟪
	0.000	--	PC0	ARP	🟩
	0.001	PC0	Router0	ARP	🟩
	0.002	Router0	PC0	ARP	🟩
	0.002	--	PC0	ICMP	🟪
👁	0.003	PC0	Router0	ICMP	🟪
👁	0.003	--	Router0	ICMP	🟥

Reset Simulation ☑ Constant Delay Captured to: 0.003 s

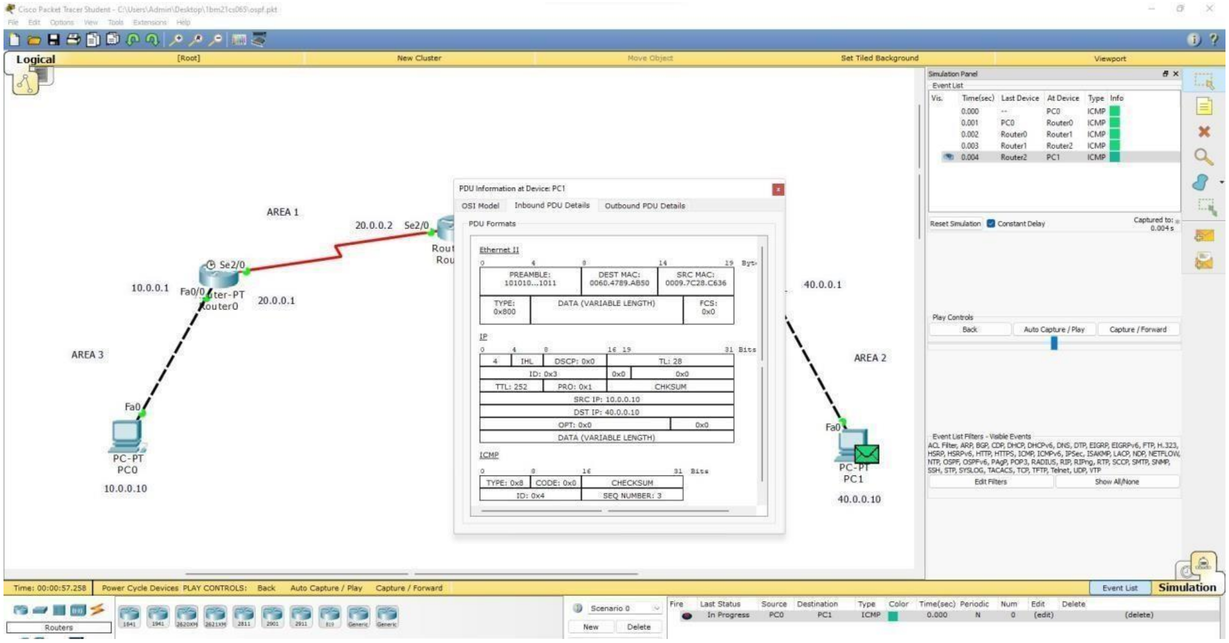
Play Controls: Back | Auto Capture / Play | Capture / Forward

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

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	In Progress	PC0	PC1	ICMP	🟪	0.000	N	0	(edit)	(delete)



PDU Information at Device: PC1

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

```
Ethernet II
0            4            8           14          19  Bytes
| PREAMBLE:         | DEST MAC:       | SRC MAC:        |
| 101010...1011     | 0060.4789.AB50  | 0009.7C28.C636  |
| TYPE:  | DATA (VARIABLE LENGTH)              | FCS:     |
| 0x800  |                                     | 0x0      |

IP
0      4      8          16 19                  31 Bits
| 4    | IHL  | DSCP: 0x0  | TL: 28                    |
| ID: 0x3                  | 0x0 | 0x0                 |
| TTL: 252    | PRO: 0x1   | CHKSUM                    |
| SRC IP: 10.0.0.10                                    |
| DST IP: 40.0.0.10                                    |
| OPT: 0x0                             | 0x0           |
| DATA (VARIABLE LENGTH)                               |

ICMP
0           8           16                31 Bits
| TYPE: 0x8 | CODE: 0x0 | CHECKSUM          |
| ID: 0x4               | SEQ NUMBER: 3     |
```

Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	🟩
	0.001	PC0	Router0	ICMP	🟩
	0.002	Router0	Router1	ICMP	🟩
	0.003	Router1	Router2	ICMP	🟩
👁	0.004	Router2	PC1	ICMP	🟩

Reset Simulation ☑ Constant Delay Captured to: 0.004 s

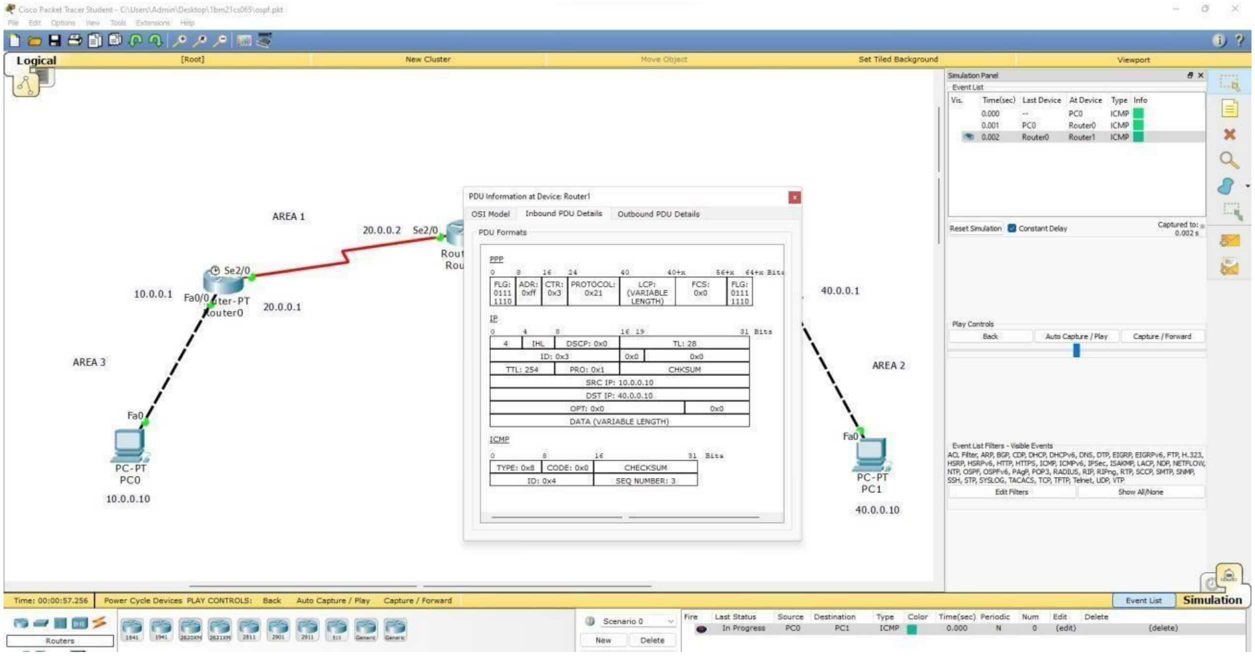
Play Controls: Back | Auto Capture / Play | Capture / Forward

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

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	In Progress	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(delete)



PDU Information at Device: Router1

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

```
PPP
0        8      16     24          40          40+x      56+x   64+x Bits
| FLG:  | ADR: | CTR: | PROTOCOL: | LCP:      | FCS: | FLG:  |
| 0111  | 0xff | 0x3  | 0x21      | (VARIABLE | 0x0  | 0111  |
| 1110  |      |      |           | LENGTH)   |      | 1110  |

IP
0      4      8             16 19                     31 Bits
| 4    | IHL  | DSCP: 0x0   | TL: 28                    |
| ID: 0x3                   | 0x0 | 0x0                 |
| TTL: 254    | PRO: 0x1    | CHKSUM                    |
| SRC IP: 10.0.0.10                                     |
| DST IP: 40.0.0.10                                     |
| OPT: 0x0                            | 0x0             |
| DATA (VARIABLE LENGTH)                                |

ICMP
0            8            16               31 Bits
| TYPE: 0x8  | CODE: 0x0  | CHECKSUM         |
| ID: 0x4                 | SEQ NUMBER: 3    |
```

Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	🟩
	0.001	PC0	Router0	ICMP	🟩
👁	0.002	Router0	Router1	ICMP	🟩

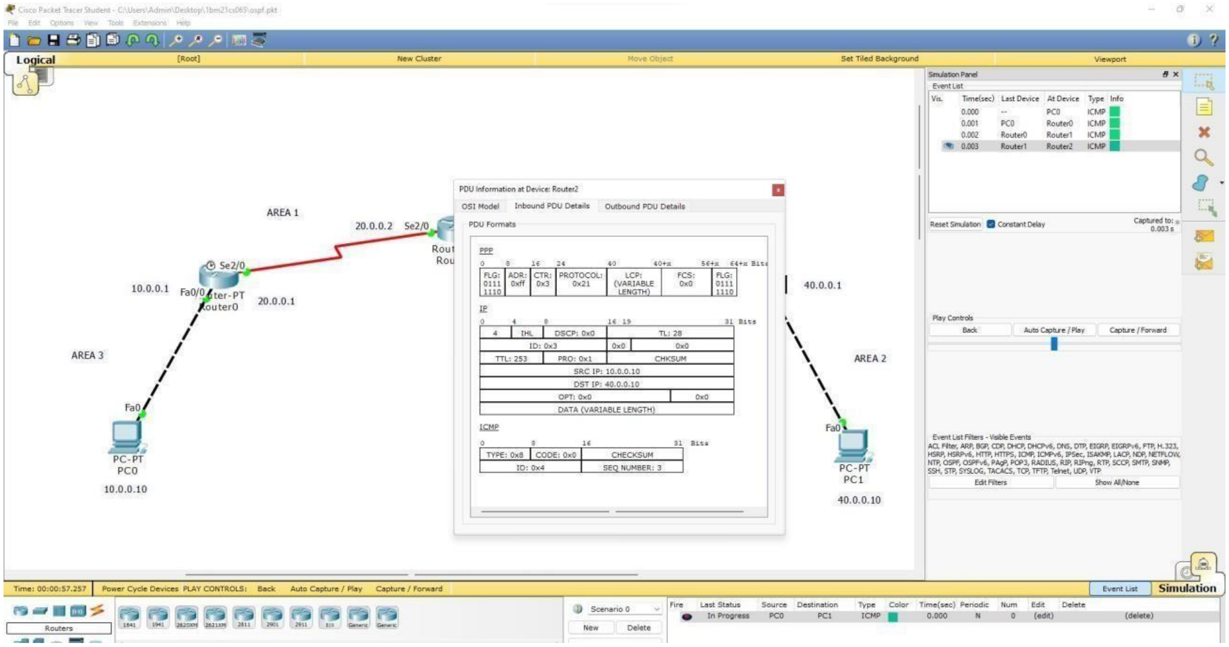
Reset Simulation ☑ Constant Delay Captured to: 0.002 s

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

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	In Progress	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(delete)



PDU Information at Device: Router2

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

```
PPP
0        8      16     24          40          40+x      56+x   64+x Bits
| FLG:  | ADR: | CTR: | PROTOCOL: | LCP:      | FCS: | FLG:  |
| 0111  | 0xff | 0x3  | 0x21      | (VARIABLE | 0x0  | 0111  |
| 1110  |      |      |           | LENGTH)   |      | 1110  |

IP
0      4      8             16 19                     31 Bits
| 4    | IHL  | DSCP: 0x0   | TL: 28                    |
| ID: 0x3                   | 0x0 | 0x0                 |
| TTL: 253    | PRO: 0x1    | CHKSUM                    |
| SRC IP: 10.0.0.10                                     |
| DST IP: 40.0.0.10                                     |
| OPT: 0x0                            | 0x0             |
| DATA (VARIABLE LENGTH)                                |

ICMP
0            8            16               31 Bits
| TYPE: 0x8  | CODE: 0x0  | CHECKSUM         |
| ID: 0x4                 | SEQ NUMBER: 3    |
```

Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	PC0	ICMP	🟩
	0.001	PC0	Router0	ICMP	🟩
	0.002	Router0	Router1	ICMP	🟩
👁	0.003	Router1	Router2	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

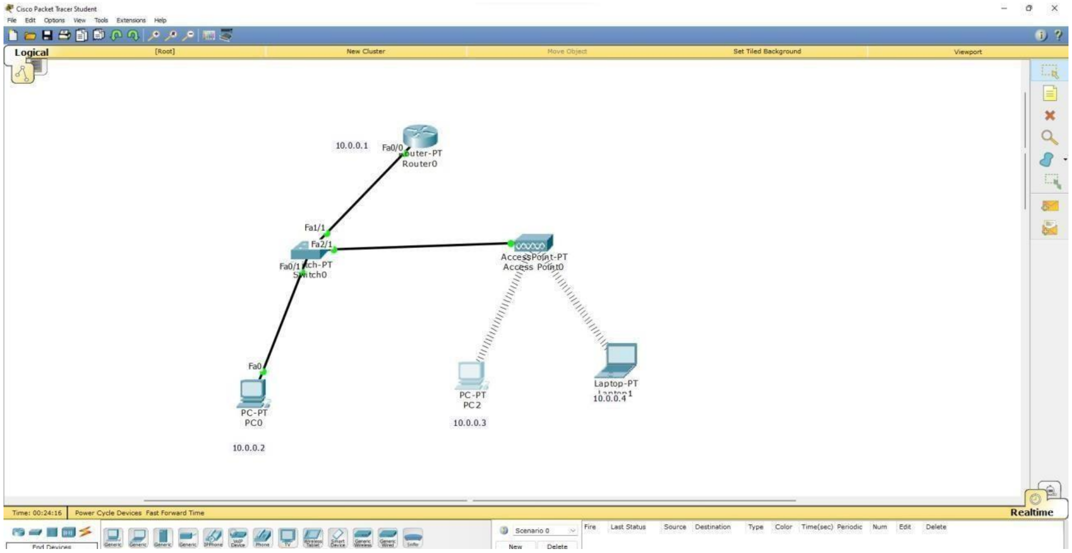
Edit Filters | Show All/None

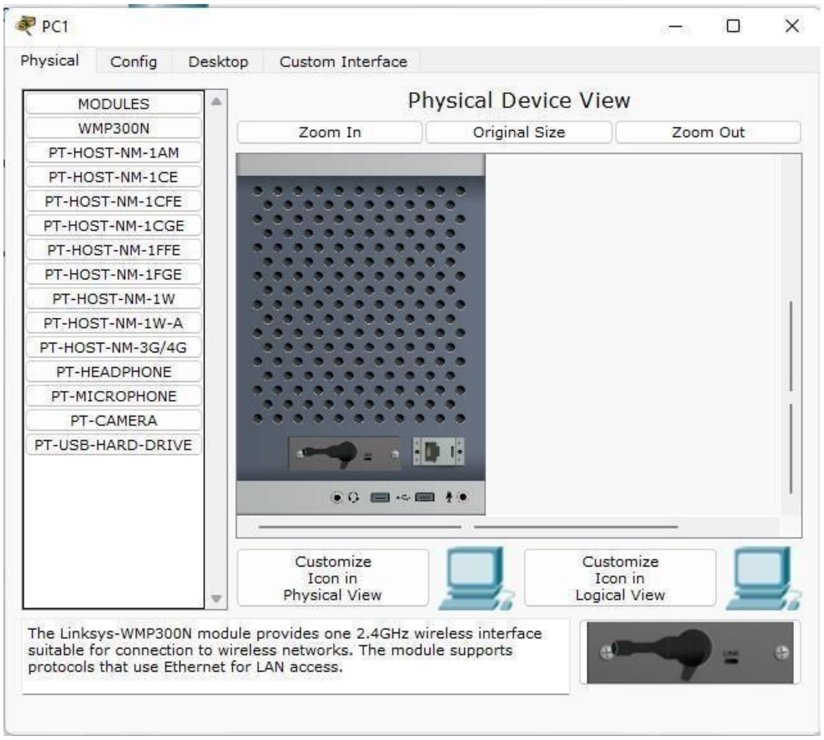
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	In Progress	PC0	PC1	ICMP	🟩	0.000	N	0	(edit)	(delete)

LAB 10

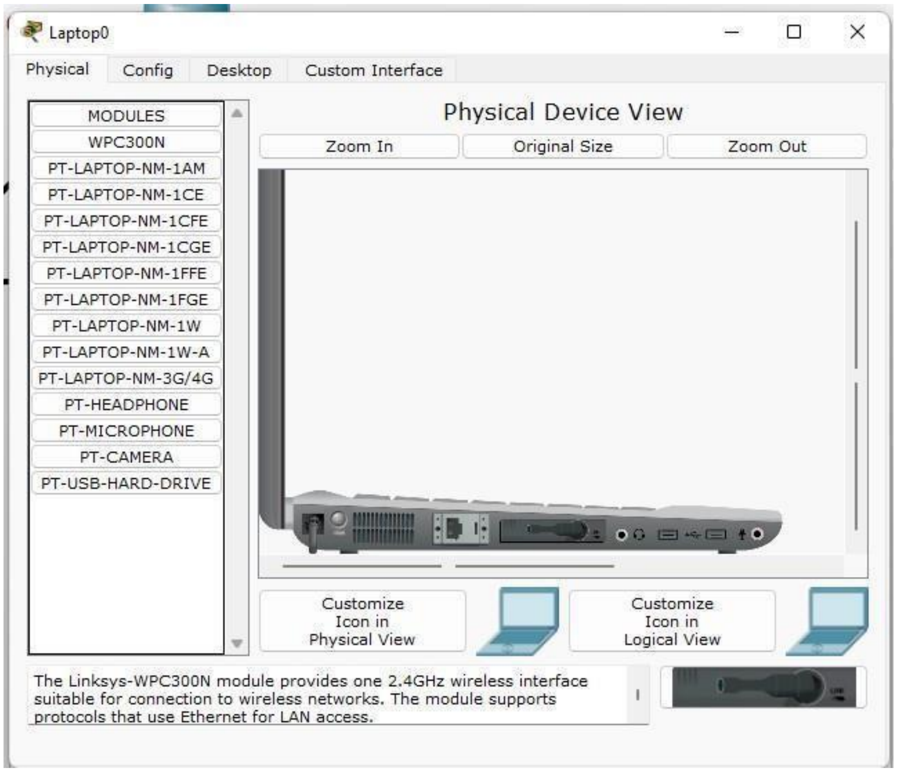
To construct a WLAN and make the nodes communicate wirelessly

TOPOLOGY:



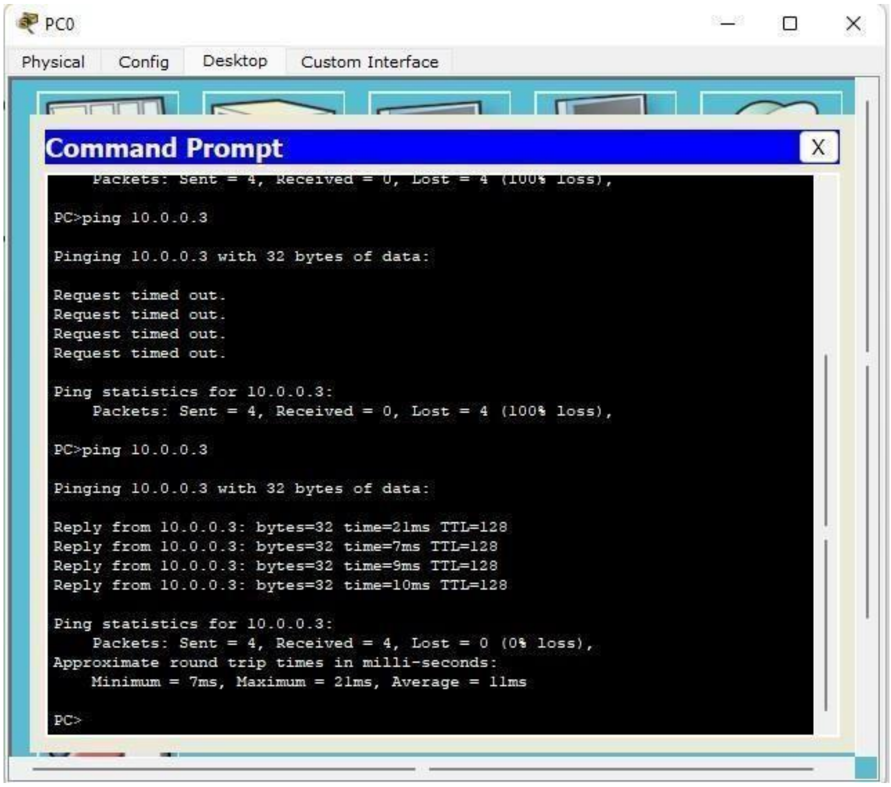


The Linksys-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.



The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

OUTPUT:

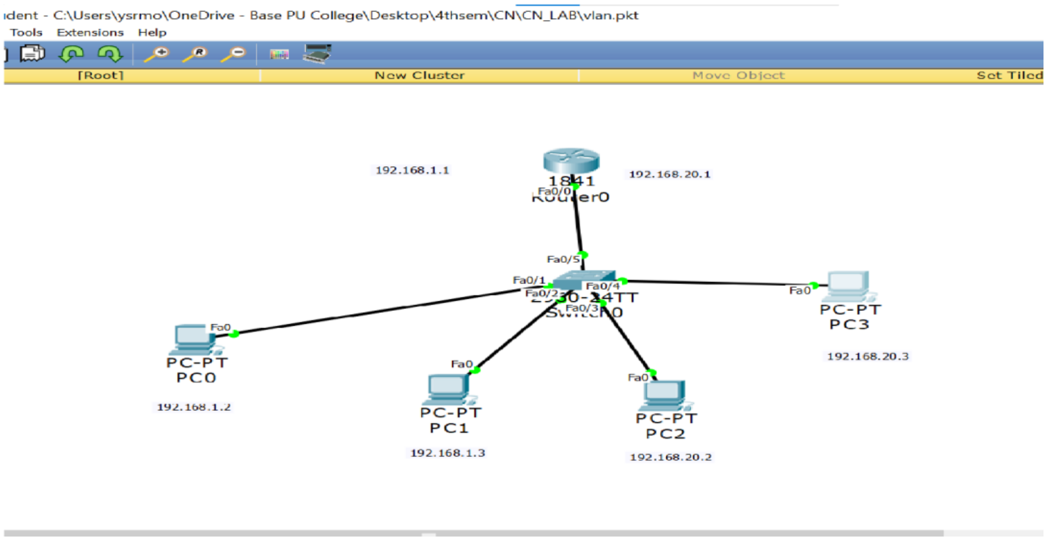


```
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=21ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=10ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 21ms, Average = 11ms

PC>
```

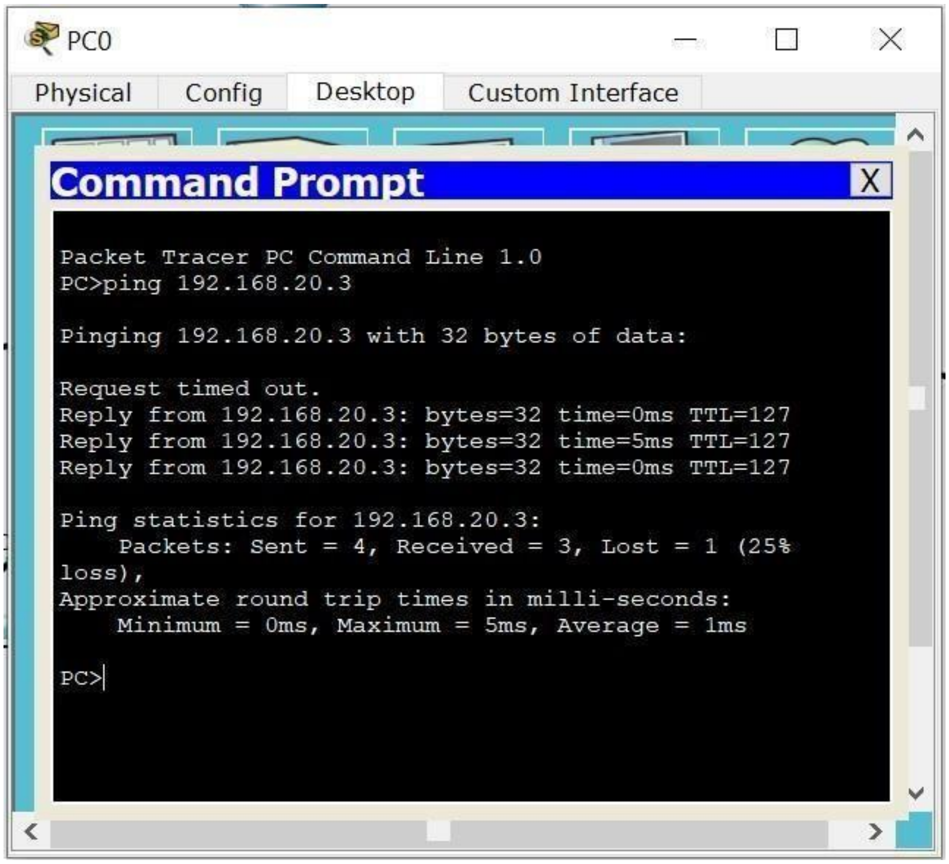
LAB 11

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

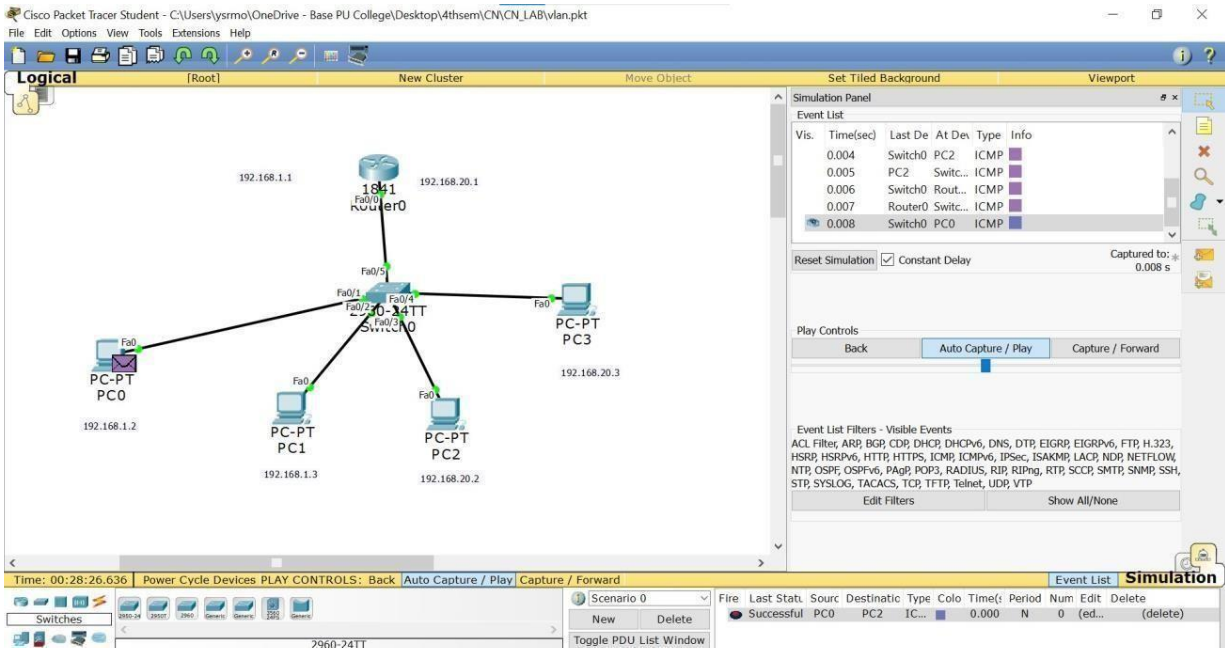
TOPOLOGY:



OUTPUT:



```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3

Pinging 192.168.20.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.20.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25%
loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 5ms, Average = 1ms

PC>
```



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt

Simulation Panel

Event List

Vis.	Time(sec)	Last De	At Dev	Type	Info
	0.004	Switch0	PC2	ICMP	🟪
	0.005	PC2	Switc...	ICMP	🟪
	0.006	Switch0	Rout...	ICMP	🟪
	0.007	Router0	Switc...	ICMP	🟪
	0.008	Switch0	PC0	ICMP	🟦

Reset Simulation ☑ Constant Delay

Captured to: 0.008 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

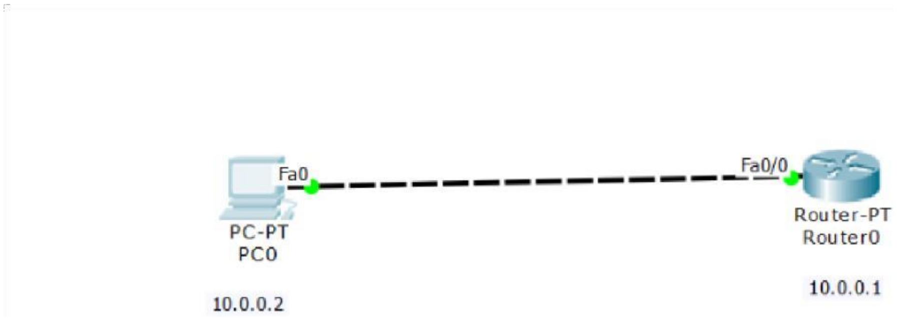
Time: 00:28:26.636 Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward

Fire	Last Statu	Sourc	Destinatic	Type	Colo	Time(s	Period	Num	Edit	Delete
●	Successful	PC0	PC2	IC...	🟦	0.000	N	0	(ed...	(delete)

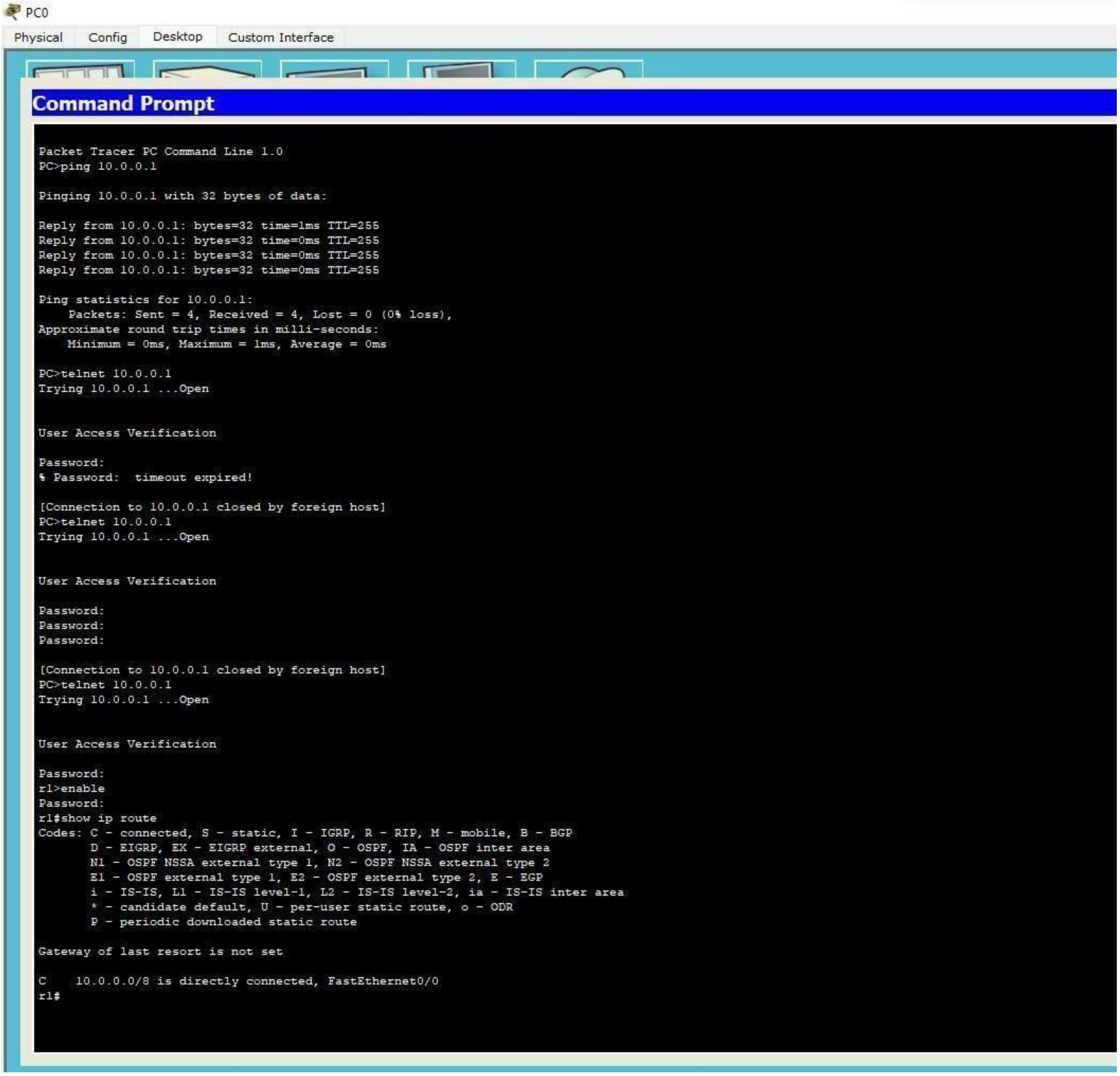
LAB 12

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

TOPOLOGY:



OUTPUT:



```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=1ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open


User Access Verification

Password:
% Password:  timeout expired!

[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open


User Access Verification

Password:
Password:
Password:

[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open


User Access Verification

Password:
r1>enable
Password:
r1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
r1#
```

LAB 13

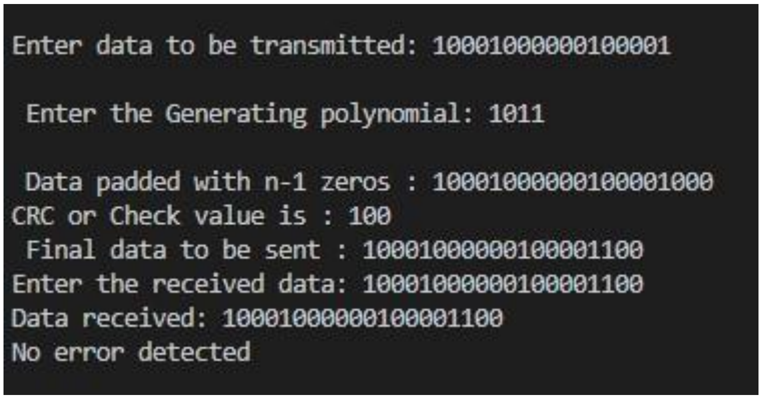
Write a program for error detecting code using CRC-CCITT (16-bits)

CODE:

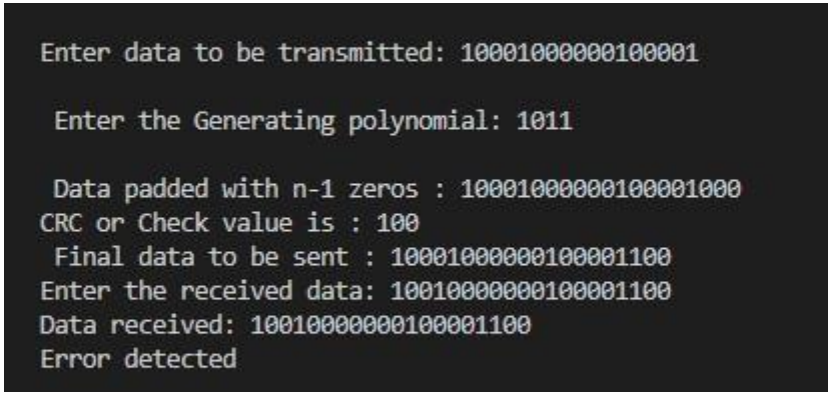
```
#include<stdio.h>#include<string.h>

#define N strlen(gen_poly) char data[28]; char check_value[28]; char
gen_poly[10]; int data_length,i,j; void XOR(){ for(j = 1;j < N; j++)
check_value[j] = (( check_value[j] == gen_poly[j])?'0':'1');} void receiver(){
printf("Enter the received data: "); scanf("%s", data); printf("Data received:
%s", data); crc(); for(i=0;(i<N-1) && (check_value[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++)
check_value[i]=data[i];
  do{ if(check_value[0]=='1')
      XOR(); for(j=0;j<N-1;j++)
    check_value[j]=check_value[j+1];
    check_value[j]=data[i++];
    }while(i<=data_length+N-1);
} int

main()
```

```
{ printf("\nEnter data to be transmitted: ");
   scanf("%s",data); printf("\n Enter the
   Generating polynomial: ");
   scanf("%s",gen_poly); data_length=strlen(data);
   for(i=data_length;i<data_length+N-1;i++)
      data[i]='0';
   printf("\n Data padded with n-1 zeros : %s",data); crc();
   printf("\nCRC or Check value is :
   %s",check_value);
   for(i=data_length;i<data_length+N-1;i++)
   data[i]=check_value[i-data_length]; printf("\n Final
   data to be sent : %s",data); receiver();
      return 0;
}
```

OUTPUT:



```
Enter data to be transmitted: 10001000000100001

 Enter the Generating polynomial: 1011

 Data padded with n-1 zeros : 10001000000100001000
CRC or Check value is : 100
 Final data to be sent : 10001000000100001100
Enter the received data: 10001000000100001100
Data received: 10001000000100001100
No error detected
```



```
Enter data to be transmitted: 10001000000100001

 Enter the Generating polynomial: 1011

 Data padded with n-1 zeros : 10001000000100001000
CRC or Check value is : 100
 Final data to be sent : 10001000000100001100
Enter the received data: 10010000000100001100
Data received: 10010000000100001100
Error detected
```

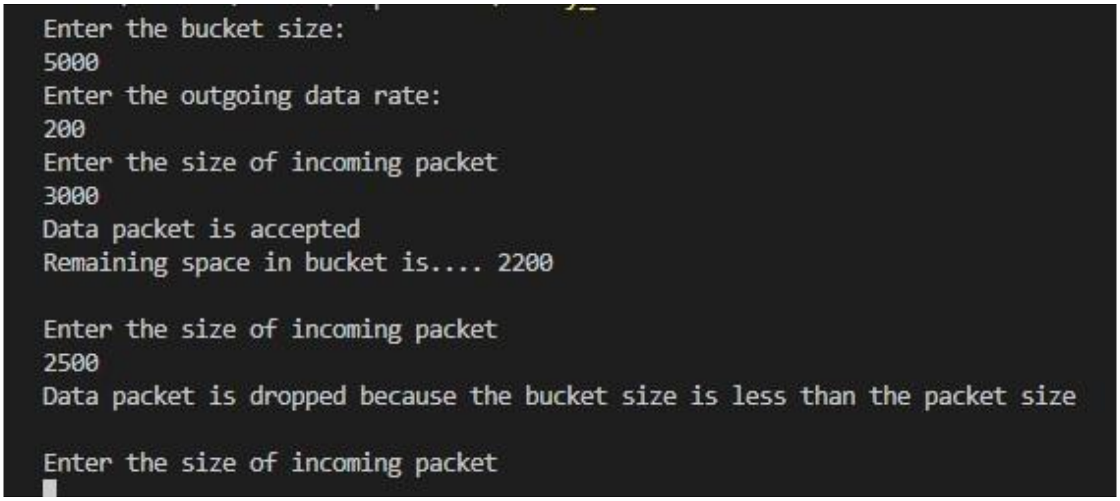
Lab 14

Write a program for congestion control using Leaky bucket algorithm.

CODE:

```
#include<stdio.h> void
main()
{ int b_size,d_rate,in_d_rate,rem_b_size;
   printf("Enter the bucket size:\n");
   scanf("%d",&b_size); rem_b_size=b_size;
   printf("Enter the outgoing data rate:\n");
   scanf("%d",&d_rate); while(1) {
   printf("Enter the size of incoming packet\n");
    scanf("%d",&in_d_rate); if(in_d_rate<=b_size)
    { if(in_d_rate<=rem_b_size) { rem_b_size=rem_b_size-
       in_d_rate; rem_b_size=rem_b_size+d_rate;
       printf("Data packet is accepted\n"); printf("Remaining
       space in bucket is....
       %d\n",rem_b_size); printf("\n");
       } else{ printf("Data packet is dropped because the bucket size is less than
       the packet
size\n"); printf("\n");
    }
    }
    }
}
```

OUTPUT:



```
Enter the bucket size:
5000
Enter the outgoing data rate:
200
Enter the size of incoming packet
3000
Data packet is accepted
Remaining space in bucket is.... 2200

Enter the size of incoming packet
2500
Data packet is dropped because the bucket size is less than the packet size

Enter the size of incoming packet
```

LAB 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.

SOLUTION:

ClientTCP.py

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

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

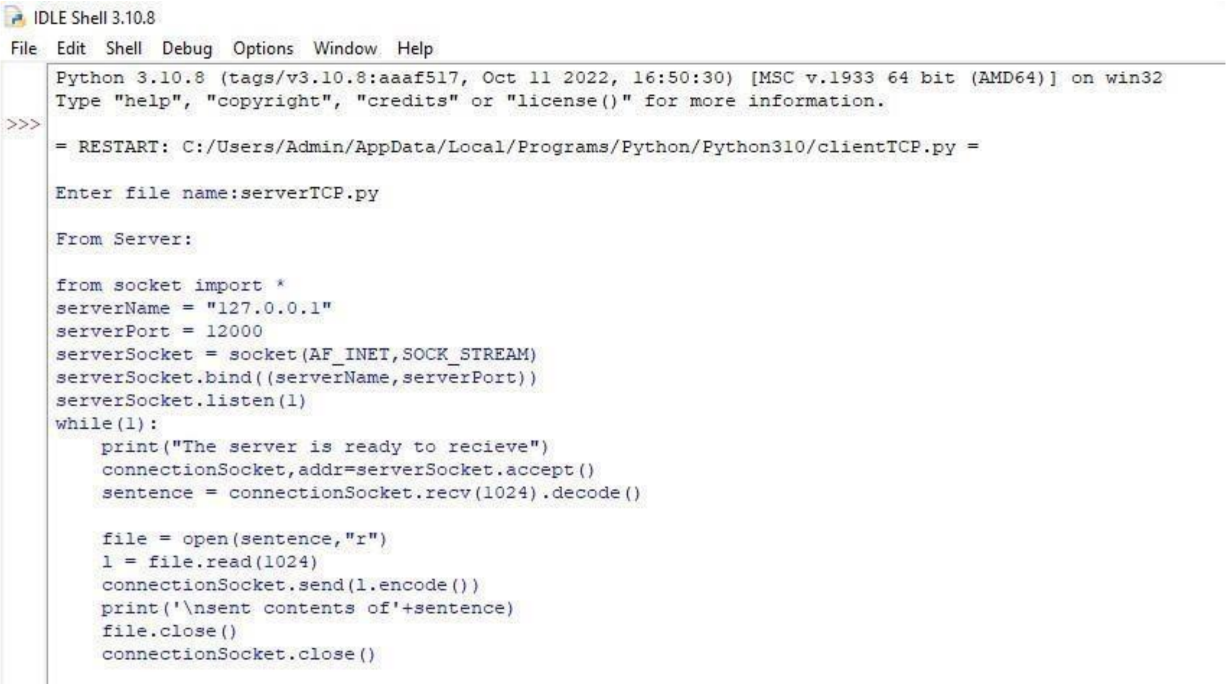
ServerTCP.py

```
from socket import * serverName="127.0.0.1"
serverPort = 12000 serverSocket =
socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1) while 1: print ("The server is
ready to receive") connectionSocket, addr =
serverSocket.accept() sentence =
connectionSocket.recv(1024).decode()
file=open(sentence,"r") l=file.read(1024)
```

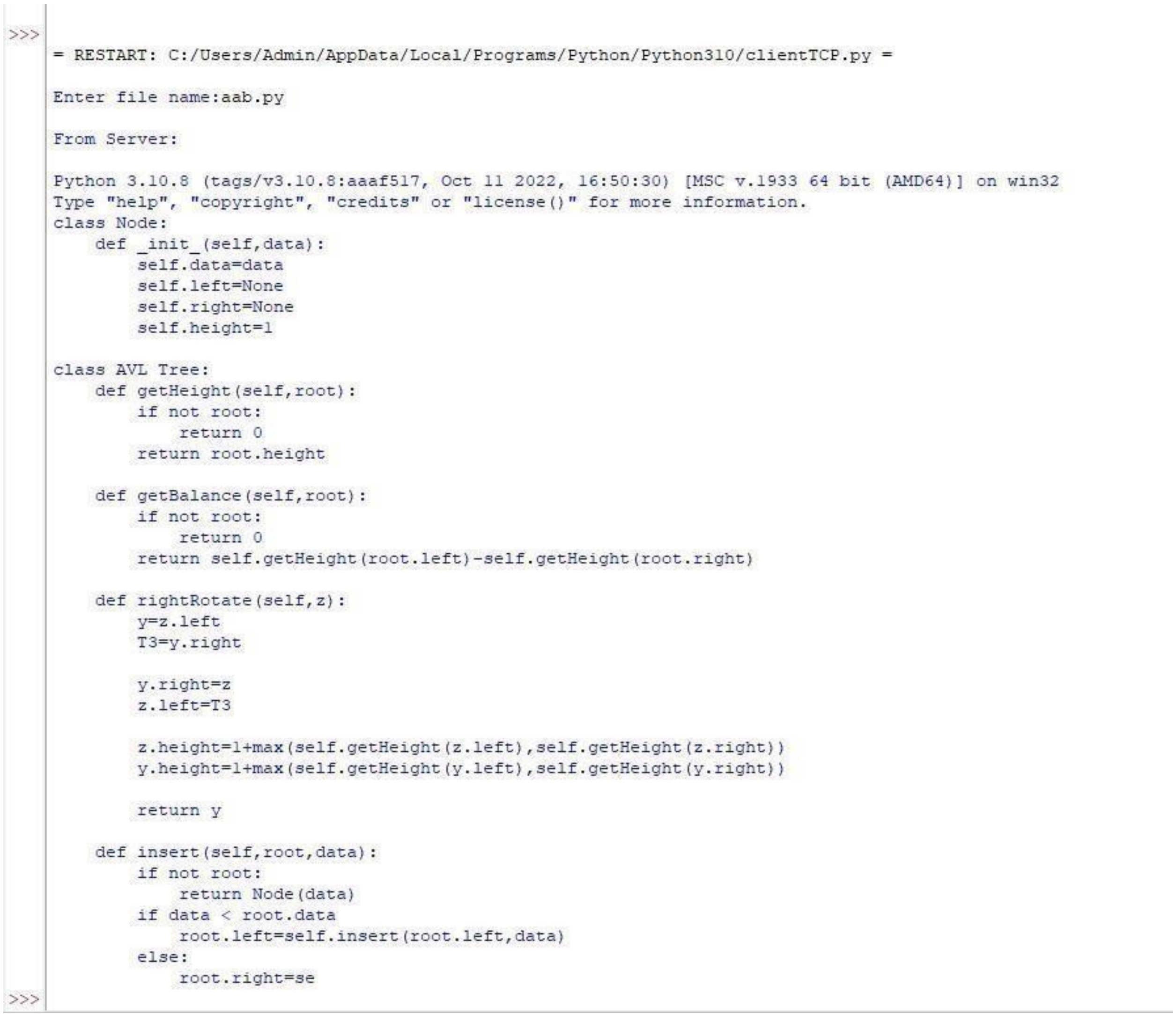
```
connectionSocket.send(l.encode())
print ('\nSent contents of ' + sentence)
file.close() connectionSocket.close()
```

OUTPUT:

Client:

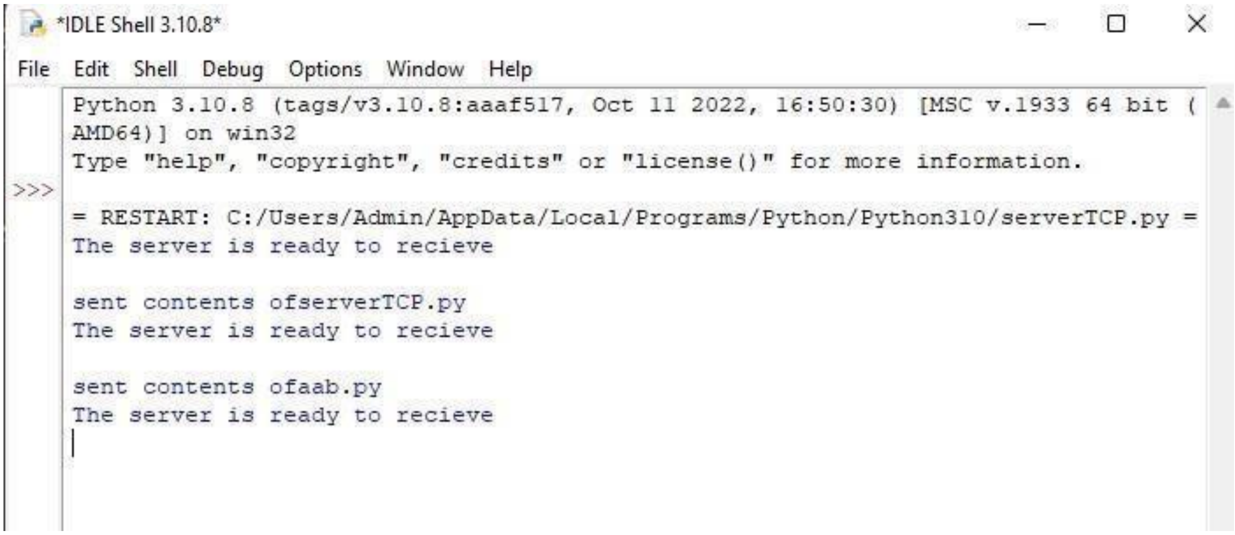


```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

Enter 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 recieve")
    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()
```



```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

Enter file name:aab.py

From Server:

Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
class Node:
    def _init_(self,data):
        self.data=data
        self.left=None
        self.right=None
        self.height=1

class AVL Tree:
    def getHeight(self,root):
        if not root:
            return 0
        return root.height

    def getBalance(self,root):
        if not root:
            return 0
        return self.getHeight(root.left)-self.getHeight(root.right)

    def rightRotate(self,z):
        y=z.left
        T3=y.right

        y.right=z
        z.left=T3

        z.height=1+max(self.getHeight(z.left),self.getHeight(z.right))
        y.height=1+max(self.getHeight(y.left),self.getHeight(y.right))

        return y

    def insert(self,root,data):
        if not root:
            return Node(data)
        if data < root.data
            root.left=self.insert(root.left,data)
        else:
            root.right=se
>>>
```

Server:



```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverTCP.py =
The server is ready to recieve

sent contents ofserverTCP.py
The server is ready to recieve

sent contents ofaab.py
The server is ready to recieve
```

Lab 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.

SOLUTION:

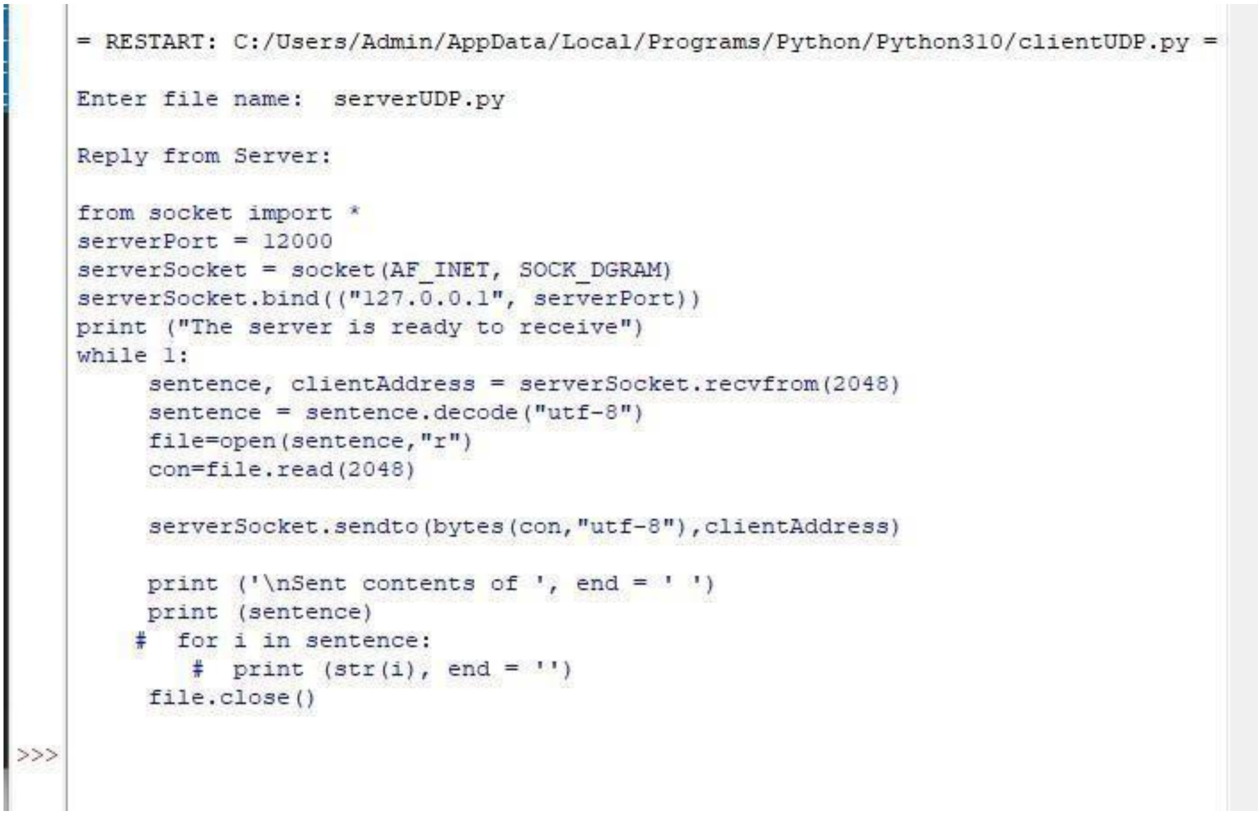
ClientUDP.py
```
from socket import * serverName = "127.0.0.1"

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()

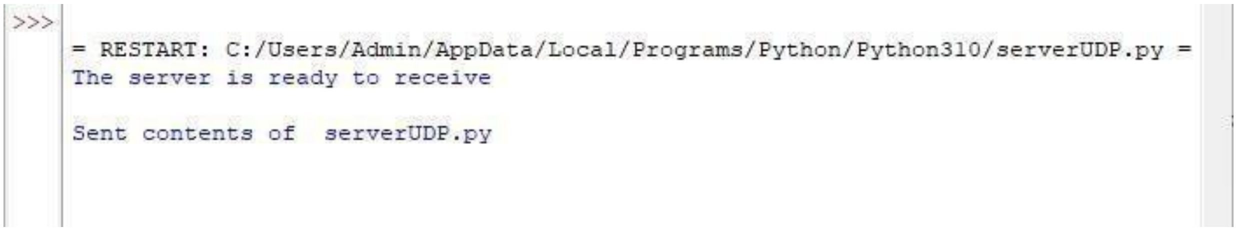
OUTPUT:

Client:



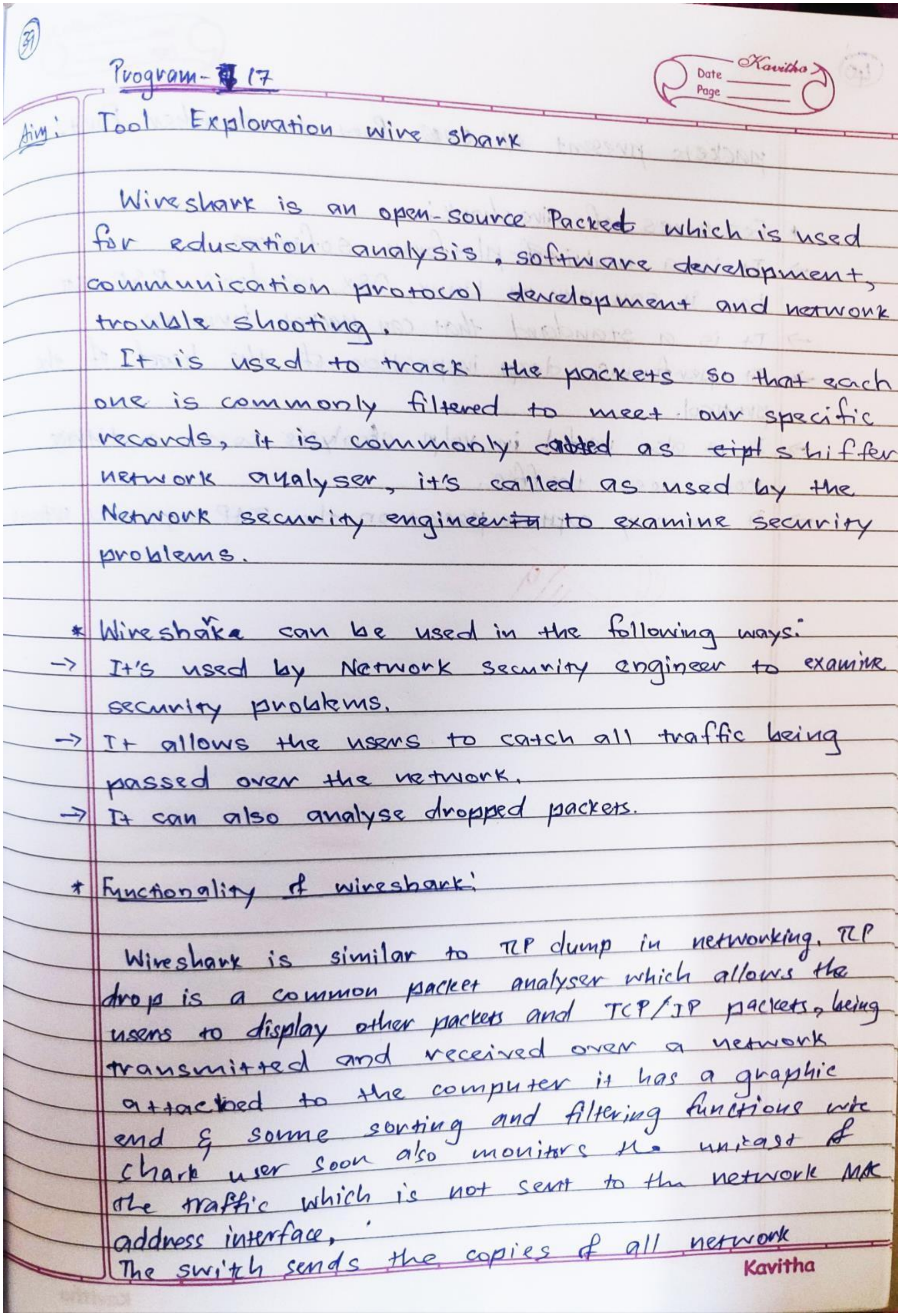
```
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientUDP.py =

Enter file name:  serverUDP.py

Reply from Server:

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()

>>>
```

Server:



```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverUDP.py =
The server is ready to receive

Sent contents of  serverUDP.py
```

Lab 17



37

Program-17

Aim: Tool Exploration wire shark

Wireshark is an open-source Packet which is used for education analysis, software development, communication protocol development and network trouble shooting.

It is used to track the packets so that each one is commonly filtered to meet our specific records, it is commonly called as ~~eipt~~ shiffer network analyser, it's called as used by the Network security engineer to examine security problems.

* Wireshark can be used in the following ways:

-> It's used by Network security engineer to examine security problems.

-> It allows the users to catch all traffic being passed over the network.

-> It can also analyse dropped packets.

* Functionality of wireshark:

Wireshark is similar to TCP dump in networking. TCP drop is a common packet analyser which allows the users to display other packets and TCP/IP packets, being transmitted and received over a network attached to the computer it has a graphic end & some sorting and filtering functions wire chark user soon also monitors the unicast of the traffic which is not sent to the network MAC address interface.

The switch sends the copies of all network

Kavitha

packets present at one Port to another Port.

* Features of wireshark:

→ It is a multi~~sof~~-platform software.
i.e, it san run on Linux, OSX, windows, BSO...etc

→ It is a standard that can packet browser.

→ It performs deep inspection of the head of the protocol.

→ It is also useful in volp analysis.com captures rows uses traffic.

→ It san only capture packet on the PCAP supported networks.

11/9