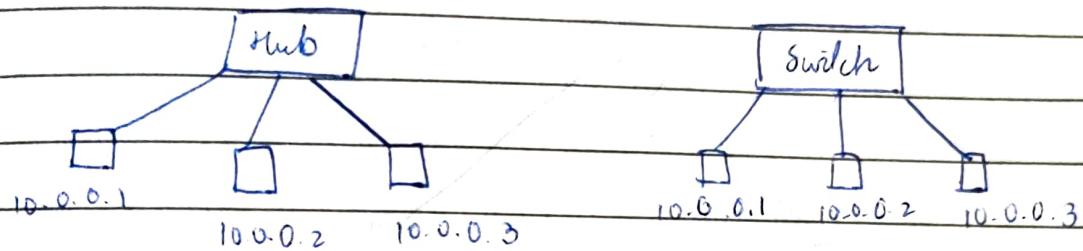


Experiment - 1

Aim: Create a topology and simulate sending a simple PDU from source to destination using hub and switch

Topology:



Procedure:

- we demonstrated star topology using hub and switch and demonstrated message passing
- configure IP addresses of PCs and form network
- In simulation mode, send PDU from source PC to destination PC

Observation:

- Hubs broadcast PDU to all the connected devices in a network.
- switches initially broadcast a PDU to all remaining connected devices. Once established unicasting takes place b/w destination and source
- message will not reach if host is not connected. (host unreachable)

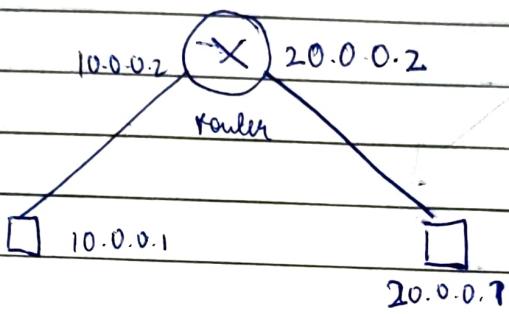
Result:

Transmitting of PDUs were successful b/w source & destination

Experiment - 2

Aim: Create an end to end connection using router and configure IP addresses, explore ping sequence, duration unreachable, seq. no. out.

Topology:



Procedure

- configure IP of end devices
- configure IP of router CLI

no

enable

config terminal

interface Fa0/0

ip address 10.0.0.2 255.0.0.0

no shutdown

exit

- update gateway addresses of PCs
- ping interface 20.0.0.1 from 10.0.0.1

Observation:

1) ping 10.0.0.2

Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 225

Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 225

Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 225

Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 225

ping statistics for 10.0.0.2

packets = send = 4, received = 4, lost = 0 (0% loss)

Avg round trip time in milliseconds

2) ping my 20.0.0.2

Reply from

ping my 20.0.0.2 with 32 bytes of data

Request timed out

Request timed out

Request timed out

Request timed out.

ping statistics for 20.0.0.2

packets sent = 4, received = 0, lost = 4 (100% loss)

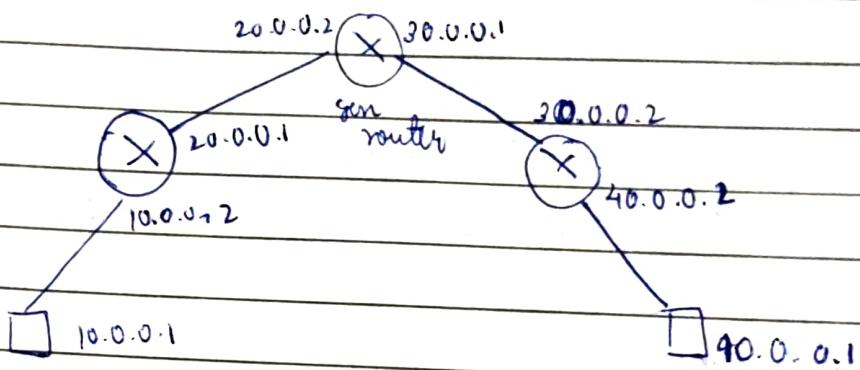
Result:

- Request timed out when gateway address is not set
- when IP is set and gateway is set, ping is ~~un~~ successful.

Experiment - 3

Aim: To configure static ip route address to the router

Topology :



Procedure :-

- configure IP addresses of PCs
- configure router interface
- set gateway addresses of the PCs
- for each other router, check networks if directly connected to another router

Show ip routes

→ connect to other networks in only one

config enable
ip route 30.0.0.0 255.0.0.0 20.0.0.0
network subnet mask next-hop

Observation :

Before connecting networks,

ping 40.0.0.1 from 10.0.0.1

pinging 40.0.0.1 with 32 bytes of data

Reply from 40.0.0.2 Destination host unreachable

Ping statistics for 40.0.0.1

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

After ping 20.0.0.2

ping Request timed out

Request timed out

Request timed out

Request timed out

Ping statistics for 20.0.0.2

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

After connecting networks

connect ping 20.0.0.1 with 32 bytes of data

Reply from 40.0.0.1

Reply from 40.0.0.1

Reply from 40.0.0.1

Reply from 40.0.0.1

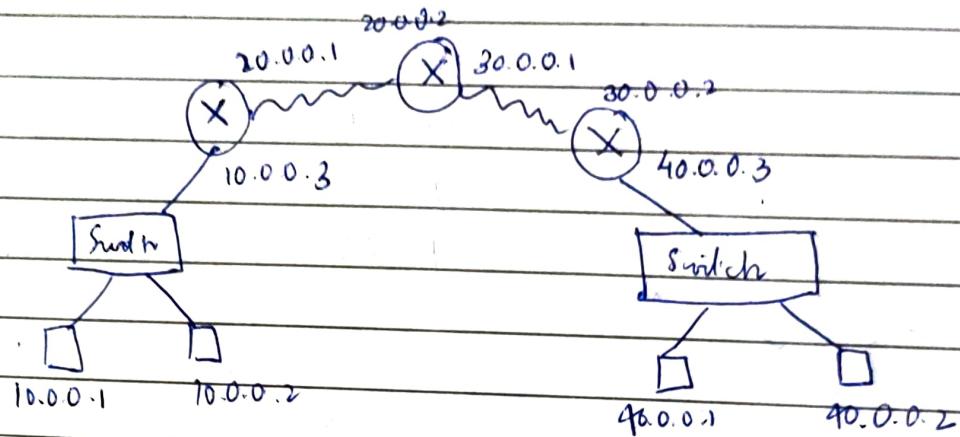
Result :

Successful ping only when networks connected manually

Experiment - 4

Pm:- To configure default routes to the routers

Configuration



Procedure :

Configure PC's and router interfaces

Set gateway addresses of PCs

Ping all interfaces

Configure default route

IP route	0.0.0.0	2.0.0.0	20.0.0.2
my/network addr	submt mask	next hop	

Observation :

before default route

pinging 20.0.0.2 with 32 bytes of data

Request timed out

Request timed out

Request timed out

Request timed out

Ping statistics for 20.0.0.2

packets : sent - 4, received - 0, lost - 4 (100% loss)

R

pinging 30.0.0.1 with 32 bytes of data

Reply from 10.0.0.3 destination host unreachable

after setting up the routes (default)

pinging 40.0.0.1 with 32 bytes of data

Reply from 40.0.0.1 bytes from 32 TTL=225

Reply from 40.0.0.1 bytes = 32 TTL = 225

Reply from 40.0.0.1 bytes = 32 TTL = 225

Reply from 40.0.0.1 bytes = 32 TTL = 225

Ping statistics for 40.0.0.1:

packets : sent - 4, received - 4, lost - 0 (0% loss)

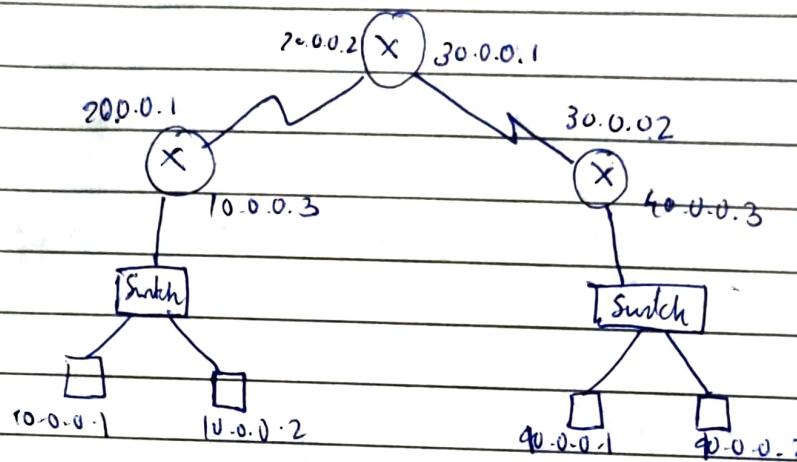
Result :

We set up all default routes for the routes.

Experiment - 5

Aim: to configure RIP protocols for a router

Topology:



Procedure:

- Configure IP addresses of PCs
- Configure routers interfaces and set serial
- ~~set gateway for interface encapsulation with ppp cmd~~
~~set clock rate~~

interface Sc 2/0

ip address 20.0.0.1 255.0.0.0

encapsulation ppp

clock rate 64000

no shutdown

Once the connections are up, set the routers up for all remaining routers and connect them.

to the known networks

```

router rip
network 10.0.0.0
network 20.0.0.0
exit.

```

Observation :

before RIP is set up, when destination is pinged
the device will be unreachable

pinging 40.0.0.2 with 32 bytes of data

Reply from 10.0.0.3 destination host unreachable

Ping statistics for 40.0.0.2.

packet: send=4, received=0, lost=4 (100% loss)

After RIP is set up

Pinging 40.0.0.2 with 32 bytes of data

Reply from 40.0.0.2 bytes = 32 TTL = 225

Reply from 40.0.0.2 bytes = 32 TTL = 225

Reply from 40.0.0.2 bytes = 32 TTL = 225

Reply from 40.0.0.2 bytes = 32 TTL = 225

Ping statistics for 40.0.0.2

packet: send=4, received=4, lost=0 (0% loss)

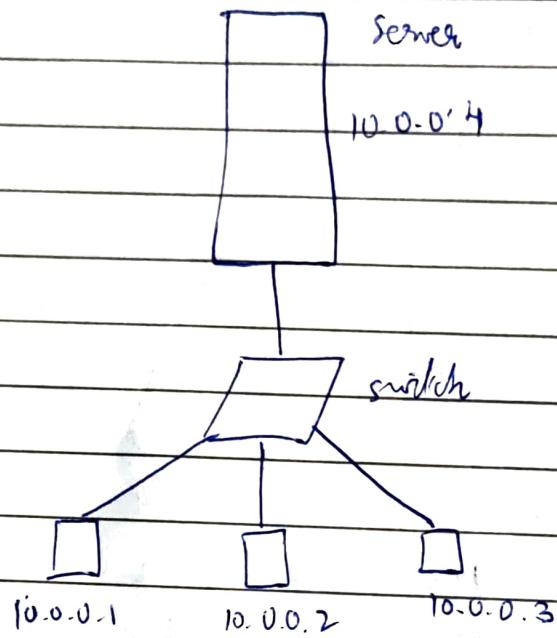
Result:

RIP protocol is set up for the router.

Experiment - 6

Aim : configure DHCP for server

Topology



Procedure:

- Construct topology & configure the IP addresses of all the devices
- Server : under services choose DHCP
Turn Service on
Set start IP address = 10.0.0.1
Subnet mask = 255.0.0.0
- Change PC's IP address from static to DHCP
 - PC's IP address is updated dynamically
 - RARP protocol is used.

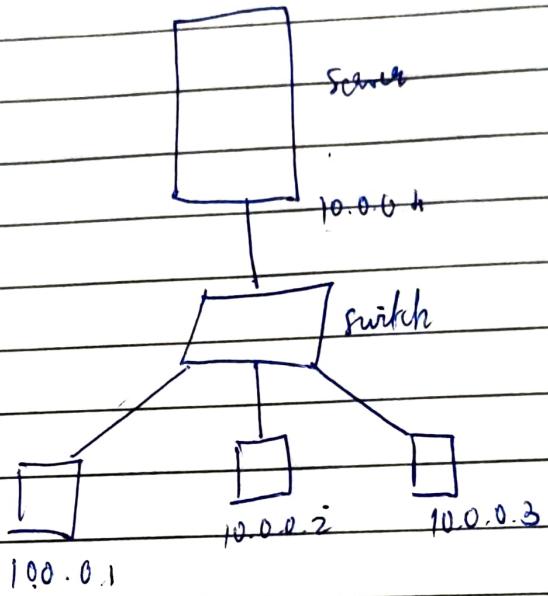
Observation :

IP's are dynamically allocated to all PC's
successful pings to all PCs.

Experiment - 7

Aim: How to configure web server and DNS server

Topology:



Procedure:

- construct topology & configure IP addresses
- server → Turn HTTP service on
- Turn DNS on
- give domain name, address cmd addl
- PC: on web browser, give address or domain name
to load webpage

Observation:

both domain name & IP can be used to access
the web page to read the page.

Experiment - 8

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

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
char data[30];
```

```
printf("Enter data");
```

```
scanf(" %s", &data);
```

```
for (int i = 0, i < 16, i++)
```

```
{
```

```
data[n+i] = '0';
```

```
data[n+i+1] = '0';
```

```
}
```

```
char dir[7] = "1000100000010001";
```

```
char x;
```

```
for (int i = 0, i < n, i++)
```

```
{
```

```
x = data[i];
```

```
if (x == '1')
```

```
{
```

```
for (int j = 0, j < 17, j++)
```

```
{
```

```
if data[i+j]
```

```
if data[i+j] != dir[j]
```

```
data[i+j] = '1';
```

```
else
```

```
data[i+j] = '0';
```

```
}
```

```
}
```

```
printf("%s", data);
```

```
}
```

Bellman Ford

#include <iostream>

typedef struct edge

{

int src;

int dest;

int weight;

} edge;

void BellmanFord (int m, edge e[], int src-graph, int n)

{

int u, v, weight, i, j = 0;

int dis [mnv];

for (i = 0; i < mn; i++)

{ dis [i] = 999;

}

dis [src-graph] = 0

for (i = 0; i < mn - 1; i++)

{

u = e [i].src;

v = e [i].dest;

if dis [u] + 999 < dis [v] + weight

{

dis [v] = dis [u] + weight;

}

j =

for ($j = 0$; $j < ne$; $j++$)

$u = e[j]. src;$

$v = e[j]. dest;$

weight = $e[j]. wt;$

}

if ($dis[u] + weight < dis[v]$)

{

cout << "cycle present \n";

}

cout << "In relax" << "distance from source";

for ($j = 1$; $j < nr; j++$)

{

cout << "In" << i << "at" << dis[i];

}

}

for

Output:

4 vertices

source relax : 1

edge 1

$sv = 1$

$dv = 2$

$wt = 4$

edge 2

$sv = 1$

$dv = 3$

$wt = 3$

edge 3

$sv = 2$

$dv = 4$

$wt = 7$

edge 4

$sv = 3$

$dv = 4$

$wt = 2$

vertex	dist
1	0
2	4
3	3
4	1

Leaky Bucket Algorithm

#include <fstream>

int main()

{

int buf;

int rate;

cout << " Enter buffer size and output rate \n";

cin >> buf;

cin >> rate;

int ip;

int cap. buffer

while (1)

{

cout << " Enter input rate \n";

cin >> ip;

if (ip > cap)

cout << " too much ";

else

{ cap = cap - ip

else

{

cout << " Out put of " << rate << "\n";

cout << " In Buffer : " << (buf - cap);

cap = cap + rate;

}

}

3

Output:

Enter buffer size and output rate

100 10

Enter input rate = 50

Output of 10

In Buffer = 40

Enter input rate = 100

Queues capacity

Physical Config

Services

Desktop

Custom Interface

SERVICES

HTTP

DHCP

DHCPv6

TFTP

DNS

SYSLOG

AAA

NTP

EMAIL

FTP

DHCPInterface Service On OffPool Name Default Gateway DNS Server Start IP Address : Subnet Mask: Maximum number of Users : TFTP Server:

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP
server...	10.0.0.4	10.0.0.4	10.0.0.1	255.0.0.0	512	0.0.0.0



Logical

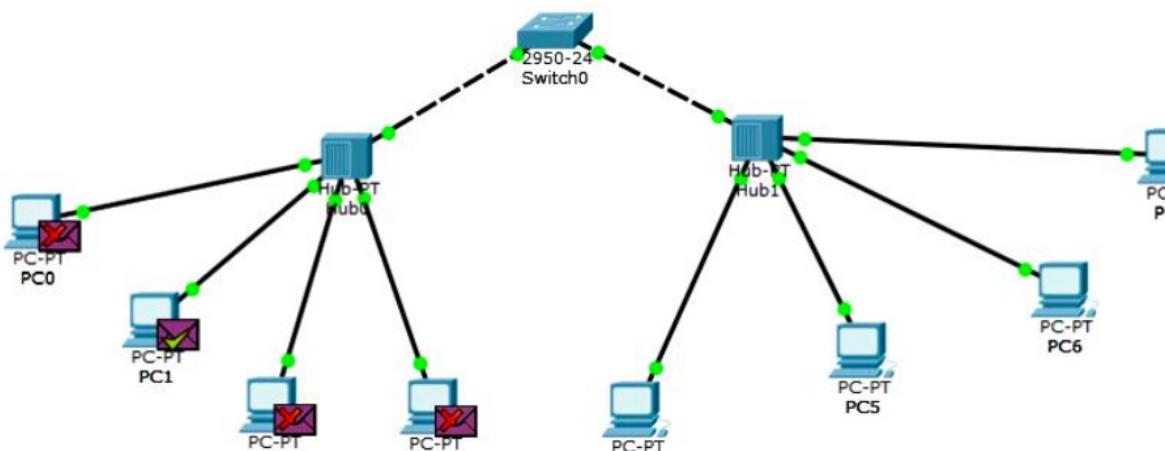
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.014	Hub1	PC4	ICMP	
	0.014	Hub1	PC6	ICMP	
	0.014	Hub1	PC7	ICMP	
	0.014	Hub1	Switch0	ICMP	
	0.015	Switch0	Hub0	ICMP	
	0.016	Hub0	PC0	ICMP	
	0.016	Hub0	PC1	ICMP	
	0.016	Hub0	PC2	ICMP	
	0.016	Hub0	PC3	ICMP	

Reset Simulation Constant DelayCaptured to: *
0.016 s

Play Controls

Event List Filters - Visible Events

ACL Filter, ARP, CDP, DHCPv6, DTP, EIGRPv6, FTP, H.323, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPFv6, PAgP, POP3, RADIUS, RIPng, RTP, SCOP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, Telnet, UDP, VTP



Time: 00:06:13.269

Power Cycle Devices

PLAY CONTROLS:

Back

Auto Capture / Play

Capture / Forward

Event List

Simulation



Connections



Automatically Choose Connection Type



Scenario 0

New

Delete

Toggle PDU List Window

Fire

Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Ed
Successful	PC1	PC5	ICMP	Blue	0.000	N	0	(



Logical

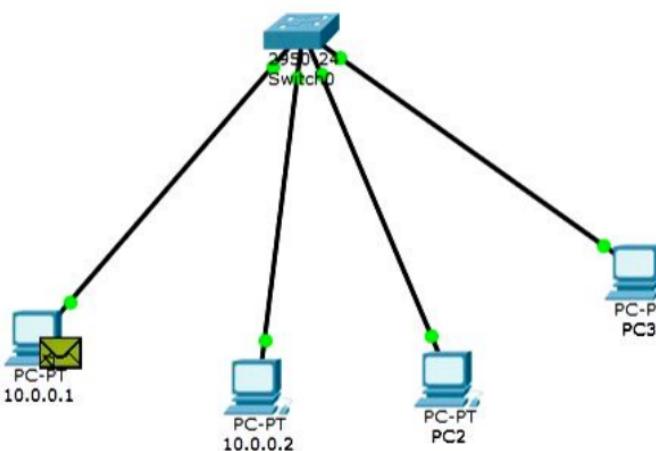
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.002	Switch0	PC2	ARP	
	0.002	Switch0	PC3	ARP	
	0.003	PC2	Switch0	ARP	
	0.004	Switch0	10.0.0.1	ARP	
	0.004	--	10.0.0.1	ICMP	
	0.005	10.0.0.1	Switch0	ICMP	
	0.006	Switch0	PC2	ICMP	
	0.007	PC2	Switch0	ICMP	
	0.008	Switch0	10.0.0.1	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, CDP, DHCPv6, DTP, EIGRPv6, FTP, H.323, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPFv6, PAgP, POP3, RADIUS, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, Telnet, UDP, VTP

Edit Filters

Show All/None



Time: 00:00:45.649

Power Cycle Devices

PLAY CONTROLS: Back

Auto Capture / Play

Capture / Forward

Event List

Simulation

Scenario 0
New Delete

Toggle PDU List Window

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Peri
●	Successful	10.0.0.1	PC2	ICMP	green	0.000	



Logical

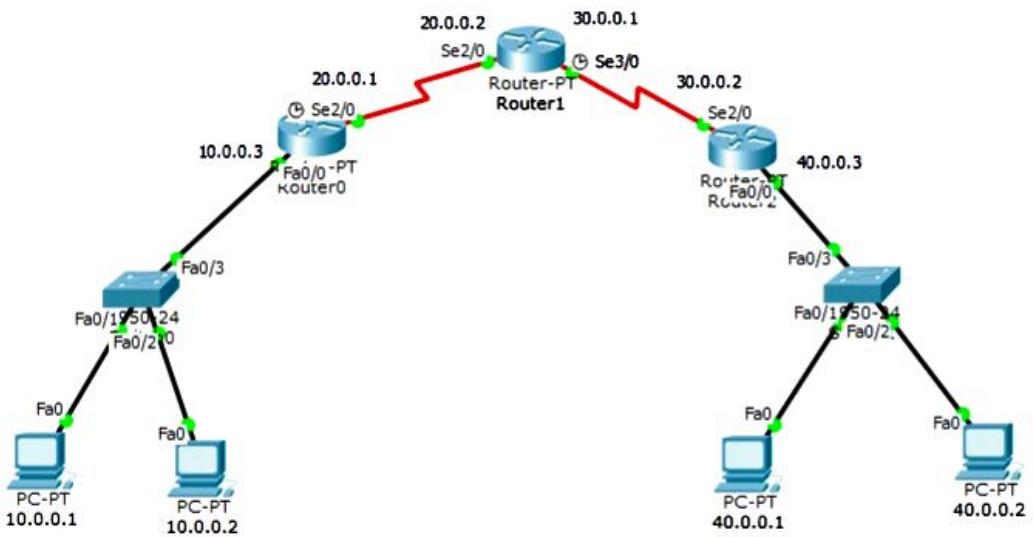
[Root]

New Cluster

Move Object

Set Tiled Background

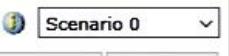
Viewport



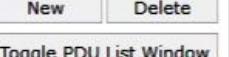
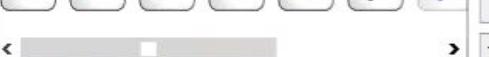
Time: 01:18:39

Power Cycle Devices Fast Forward Time

Realtime



Fire	Last Status	Source	Destination	Type	Color	T





Logical

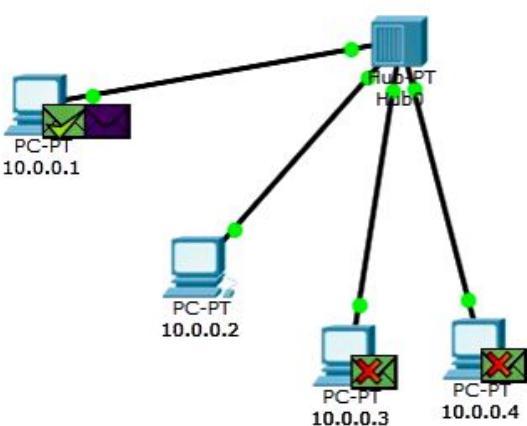
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
0.001		10.0.0.1	Hub0	ARP	
0.002		Hub0	10.0.0.2	ARP	
0.002		Hub0	10.0.0.3	ARP	
0.002		Hub0	10.0.0.4	ARP	
0.003		10.0.0.2	Hub0	ARP	
0.004		Hub0	10.0.0.1	ARP	
0.004		Hub0	10.0.0.3	ARP	
0.004		Hub0	10.0.0.4	ARP	
0.004	--		10.0.0.1	ICMP	

Reset Simulation

 Constant DelayCaptured to:
0.004 s

Play Controls

Back

Auto Capture / Play

Capture / Forward

Event List Filters - Visible Events

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

Edit Filters

Show All/None

Time: 00:00:12.537

Power Cycle Devices

PLAY CONTROLS: Back

Auto Capture / Play

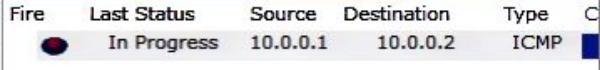
Capture / Forward

Simulation



New Delete

Toggle PDU List Window



Select a Device to Drag and Drop to the Workspa



Physical Config Desktop Custom Interface

Command Prompt

X

```
Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 3ms, Maximum = 16ms, Average = 9ms
```

```
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=8ms TTL=254  
Request timed out.  
Reply from 20.0.0.2: bytes=32 time=6ms TTL=254
```

```
Ping statistics for 20.0.0.2:
```

```
    Packets: Sent = 4, Received = 2 (50% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 6ms, Maximum = 8ms, Average = 7ms
```

```
PC>ping 30.0.0.1
```

```
Pinging 30.0.0.1 with 32 bytes of data:
```

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

```
Ping statistics for 30.0.0.1:
```

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

Enter the number the routers(<10): 5
Enter 1 if the corresponding router is adjacent to routerA else enter 99:
B C D E
Enter matrix:1 1 99 99

Enter 1 if the corresponding router is adjacent to routerB else enter 99:
A C D E
Enter matrix:1 99 99 99

Enter 1 if the corresponding router is adjacent to routerC else enter 99:
A B D E
Enter matrix:1 99 1 1

Enter 1 if the corresponding router is adjacent to routerD else enter 99:
A B C E
Enter matrix:99 99 1 99

Enter 1 if the corresponding router is adjacent to routerE else enter 99:
A B C D
Enter matrix:99 99 1 99

Router Table entries for router A:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 0 1 1 99 99

Router Table entries for router B:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 1 0 99 99 99

Router Table entries for router C:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 1 99 0 1 1

Router Table entries for router D:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 99 99 1 0 99

Router Table entries for router E:-
Destination Router: A B C D E
Outgoing Line: A B C D E
Hop Count: 99 99 1 99 0



10.0.0.1



Physical Config Desktop Custom Interface

Command Prompt



```
PC>ping 20.0.0.2
```

```
Pinging 20.0.0.2 with 32 bytes of data:
```

```
Reply from 20.0.0.2: bytes=32 time=0ms TTL=255
```

```
Ping statistics for 20.0.0.2:
```

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

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
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=127
```

```
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 = 0ms, Average = 0ms
```

```
PC>
```



Physical Config Desktop Custom Interface

Command Prompt

X

```
PC>ping 40.0.0.1
```

```
Pinging 40.0.0.1 with 32 bytes of data:
```

```
Request timed out.
```

```
Request timed out.
```

```
Reply from 40.0.0.1: bytes=32 time=13ms TTL=125
```

```
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
```

```
Ping statistics for 40.0.0.1:
```

```
    Packets: Sent = 4, Received = 2 (50% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 8ms, Maximum = 13ms, Average = 10ms
```

```
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=14ms TTL=125
```

```
Reply from 40.0.0.1: bytes=32 time=11ms TTL=125
```

```
Reply from 40.0.0.1: bytes=32 time=3ms TTL=125
```

```
Reply from 40.0.0.1: bytes=32 time=13ms TTL=125
```

```
Ping statistics for 40.0.0.1:
```

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

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 3ms, Maximum = 14ms, Average = 10ms
```

```
PC>
```



Logical

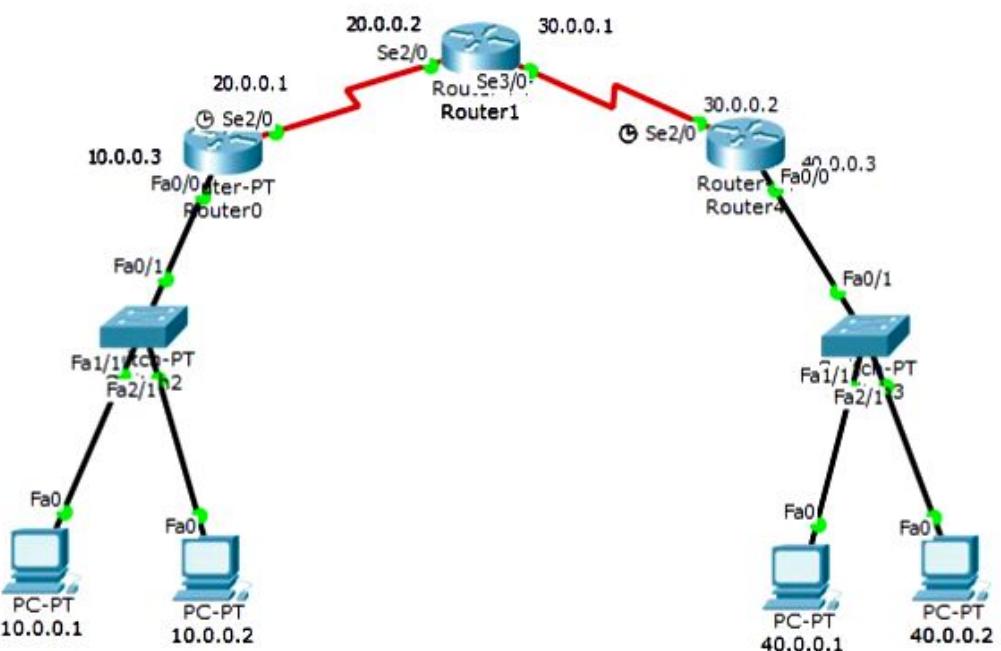
[Root]

New Cluster

Move Object

Set Tiled Background

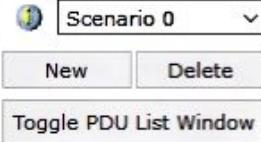
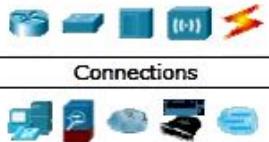
Viewport



Time: 01:10:41

Power Cycle Devices Fast Forward Time

Realtime



Fire	Last Status	Source	Destination	Type

Automatically Choose Connection Type



Logical

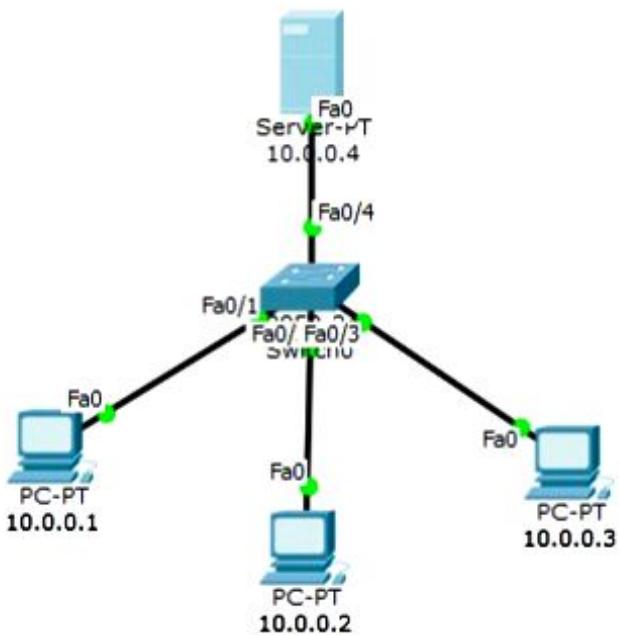
[Root]

New Cluster

Move Object

Set Tiled Background

Viewport



Time: 00:47:21

Power Cycle Devices Fast Forward Time

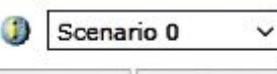
Realtime



Connections

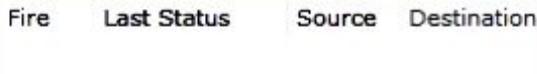


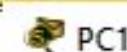
Automatically Choose Connection Type



New Delete

Toggle PDU List Window





PC1



Physical Config Desktop Custom Interface

Command Prompt



Packet Tracer PC Command Line 1.0

PC>ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data:

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Ping statistics for 30.0.0.1:

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:

Reply from 40.0.0.1: bytes=32 time=17ms TTL=128

Reply from 40.0.0.1: bytes=32 time=6ms TTL=128

Reply from 40.0.0.1: bytes=32 time=6ms TTL=128

Reply from 40.0.0.1: bytes=32 time=6ms TTL=128

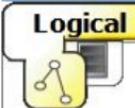
Ping statistics for 40.0.0.1:

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

Approximate round trip times in milli-seconds:

Minimum = 6ms, Maximum = 17ms, Average = 8ms

PC>



[Root]

New Cluster

Move Object

Set Tiled Background

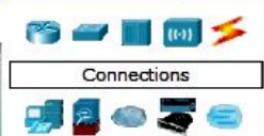
Viewport



< >

Time: 00:57:18	Power Cycle Devices	Fast Forward Time
----------------	---------------------	-------------------

Realtime



Fire	Last Status	Source	Destination	Type	Color	Time(sec)

Automatically Choose Connection Type



Logical

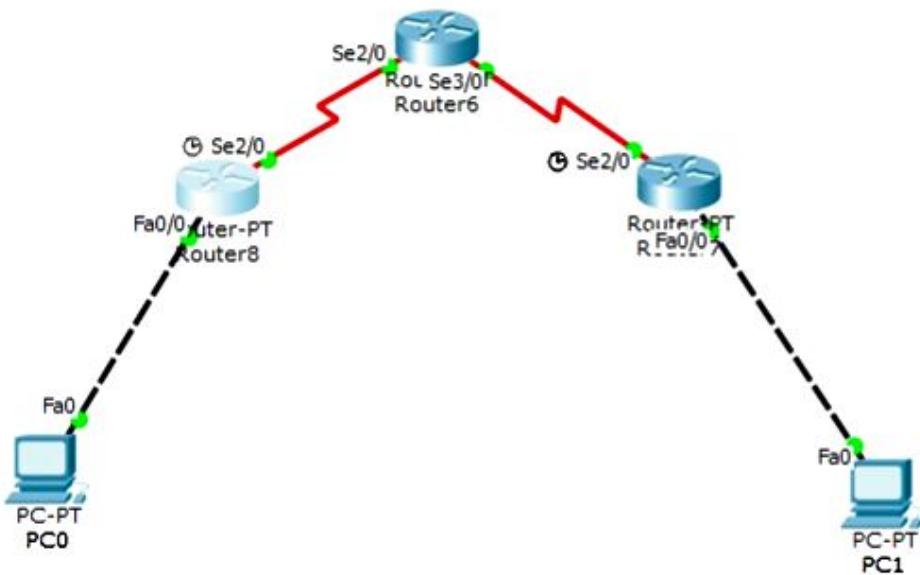
[Root]

New Cluster

Move Object

Set Tiled Background

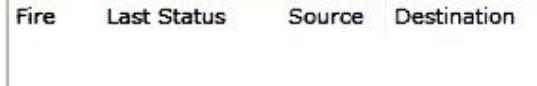
Viewport



Time: 00:53:31

Power Cycle Devices Fast Forward Time

Realtime



C:\Users\3\mikew\, OneDrive\Desktop, 2021-06-01 10:50:00

The server is ready to receive

Sent contents of serverTCP.py

The server is ready to receive



```
Enter file name: serverTCP.py
```

```
From Server:
```

```
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()
```

```
PS C:\Users\Lokesh R\Desktop\socket> & "C:/Program Files/Python311/python.exe" "c:/Users/Lokesh R/Desktop/socket/serverudp.py"
The server is ready to receive
```

Sent contents of serverudp.py



Physical Config CLI

- □ ×

IOS Command Line Interface

```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 20.0.0.1
Router(config)#ip route 0.0.0.0 0.0.0.0 30.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
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 20.0.0.1 to network 0.0.0.0

C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
S*   0.0.0.0/0 [1/0] via 20.0.0.1
                 [1/0] via 30.0.0.2
Router#
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Copy

Paste



Physical Config Desktop Custom Interface

Web Browser

X



URL

<http://www.google.com>

Go

Stop

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](#)



Physical Config Desktop Custom Interface

Command Prompt

X

```
PC>ping 40.0.0.1
|
Pinging 40.0.0.1 with 32 bytes of data:

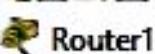
Reply from 10.0.0.3: Destination host unreachable.

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=6ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 10ms, Average = 6ms
PC>ping 40.0.0.2
```



Physical Config CLI

IOS Command Line Interface

```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
exit
Router(config)#interf
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to
% Incomplete command.
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#exit
Router(config)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

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

Copy

Paste

```
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")
    l=file.read(2048)

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



PC1

- □ X

Physical Config Desktop Custom Interface

Command Prompt

X

```
Packet Tracer PC Command Line 1.0
```

```
PC>ping 30.0.0.1
```

```
Pinging 30.0.0.1 with 32 bytes of data:
```

```
Request timed out.
```

```
Ping statistics for 30.0.0.1:
```

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

```
PC>
```



Physical Config CLI

IOS Command Line Interface

```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 20.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#
|
```

Router con0 is now available

Copy

Paste



Physical Config

Services

Desktop

Custom Interface

SERVICES

HTTP

DHCP

DHCPv6

TFTP

DNS

SYSLOG

AAA

NTP

EMAIL

FTP

DNS

DNS Service

On

Off

Resource Records

Name

Type

Address

Add

Save

Remove

No.	Name	Type	Detail
0	www.google.com	A Record	10.0.0.4

DNS Cache



Router6

- □ ×

[Physical](#) [Config](#) [CLI](#)

IOS Command Line Interface

```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
exit
```

Router con0 is now available

[Copy](#)[Paste](#)

Enter the graph

0 9 2 5
9 0 6 8
2 6 0 0
5 8 0 0

Vertex Distance from Source
0 0
1 8
2 2
3 5





Physical Config CLI

IOS Command Line Interface

```
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 20.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#exit
Router(config)#
|
```

Router con0 is now available

Copy

Paste