Week - 2

Date 10/11/22
Page

⇒ Aim :

Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.

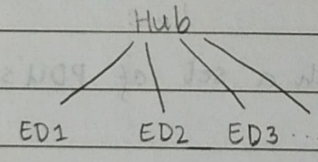
⇒ Topology Procedure

- connect end devices (4) [generic PC's] to a connecting device.
- configure the IP address for all the end devices.
- send a simple PDU from chosen source & destination
- In simulation mode, select capture / Forward to see the working

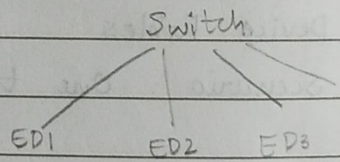
⇒ Topology :

Star Topology

i) Using Hub



ii) Using Switch



⇒ Observation :

⇒ - When the PC's are connected to a hub, the hub broadcasts the received PDU to every other PC irrespective of chosen destination
- Hub receives PDU & simply transfers it to next layers & then broadcasts it.
- If the destination address matches, PC accepts PDU.

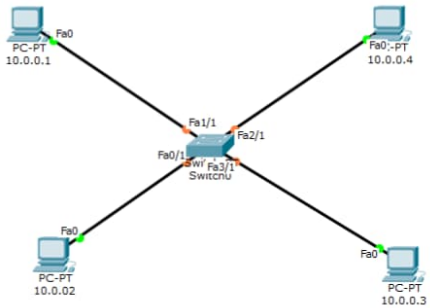
⇒ - When switch is used, initially broadcasts the PDU & forms switch table.
- In subsequent transfers, switch unicasts the PDU to particular PC with destination address using

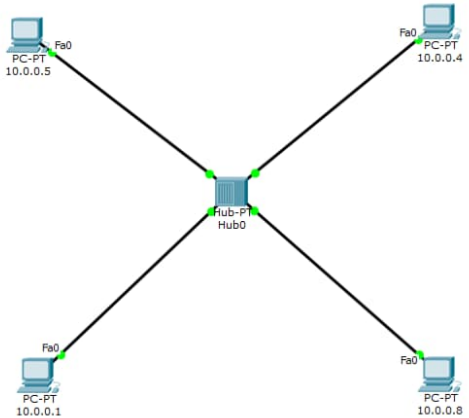
Date __/__/__
Page _____

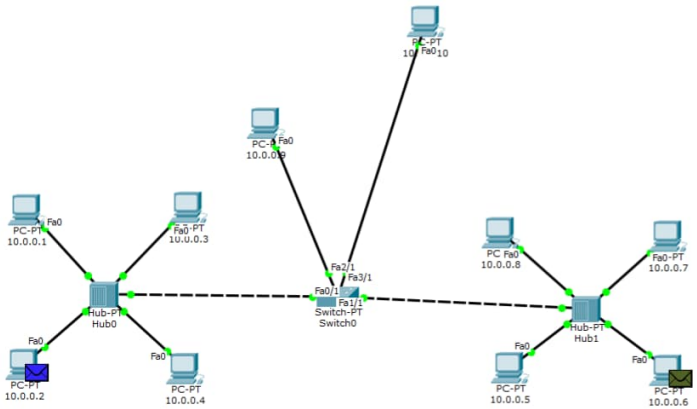
switch table.

- When switch receives a PDU, it undergoes decapsulation & ~~get~~ checks the destination adress & encapsulates it again & then transfers it to specific PC.

Neelima
17/11/2022


PC-PT
10.0.0.1
Fa0
Fa0
PC-PT
10.0.0.4
Fa1/1
Fa2/1
Fa0/1
Fa3/1
Switch0
Fa0
PC-PT
10.0.02
Fa0
PC-PT
10.0.0.3


Fa0
PC-PT
10.0.0.5
Fa0
PC-PT
10.0.0.4
Hub-PT
Hub0
Fa0
PC-PT
10.0.0.1
Fa0
PC-PT
10.0.0.8



Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	10.0.0.2	ICMP	
	0.000	--	10.0.0.6	ICMP	

Reset Simulation | Constant Delay | Captured to: 0.000 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

Date 17/11/22
Page

Week - 3

⇒ Aim:

Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.

⇒ Procedure:

- Connect two end devices (generic PC's) to a router.
- Configure the IP addresses for end devices which belong to different networks
- Configure the interface of Router using CLI with the below commands.

enable → to start configuring Router Interface

config terminal → to set necessary permissions

interface [interface name] → {Fa0/0, Fa1/0} Select interface

ip address 10.0.0.2 255.0.0.0 → set the IP address for interface & also subnet mask

no shutdown → to change state to active

- Similarly, configure the IP address for two router interfaces using command line
- Ping a device from one of the device.

Messages:

Request time out : As there is no gateway set

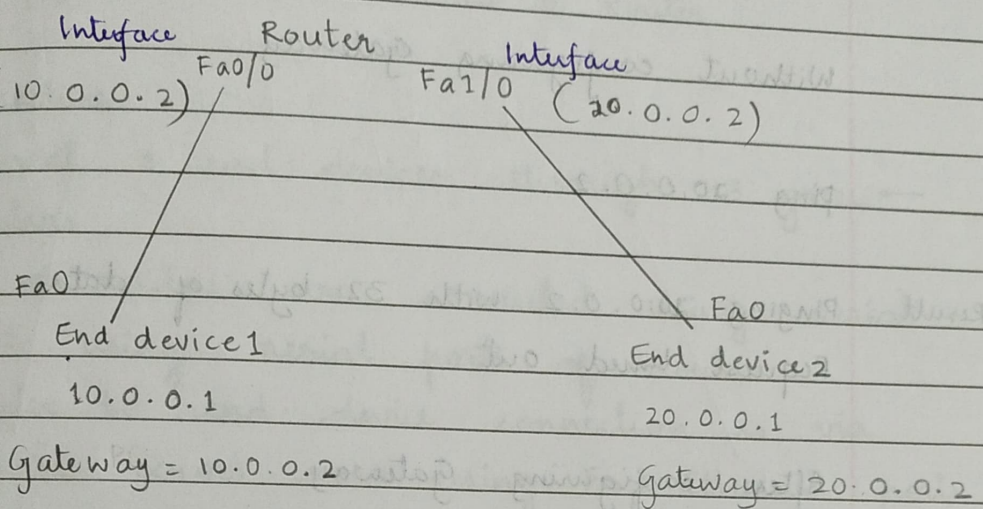
- Now, set the gateway for the end devices Gateway is the interface address of the router.
- Now, ping again from one of end device.

Messages:

Reply messages will be received

- When a device which is not in the network is pinged, a destination unreachable message is received.

⇒ Topology :



⇒ Outcome

- The following messages are explored when pinged from a end device :

* Request timed out
* Destination unreachable
* Reply

Request timed Out :

- When an end device is pinged without specifying the gateway for end devices, the message would be 'Request Timed Out'

Reply :

- After the end devices are configured with gateway, the ping will be successful and reply is received consisting of bytes, time & TTL

- Destination Unreachable :
When an end device which is not in the network is pinged, a 'destination unreachable' message is received.

Date __/__/__
Page

Example

Without configuring Gateway

→ ping 20.0.0.2

Result : Pinging 20.0.0.2 with 32 bytes of data
Request timed out

After configuring Gateway

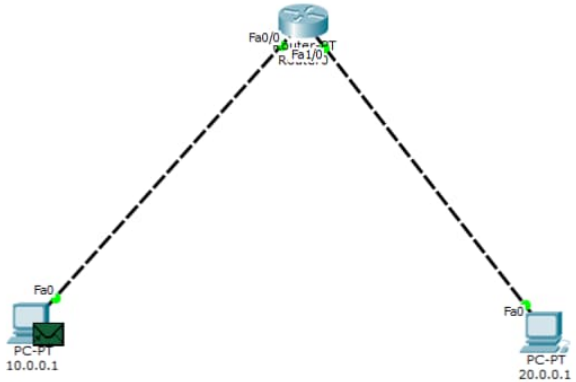
→ ping 20.0.0.1

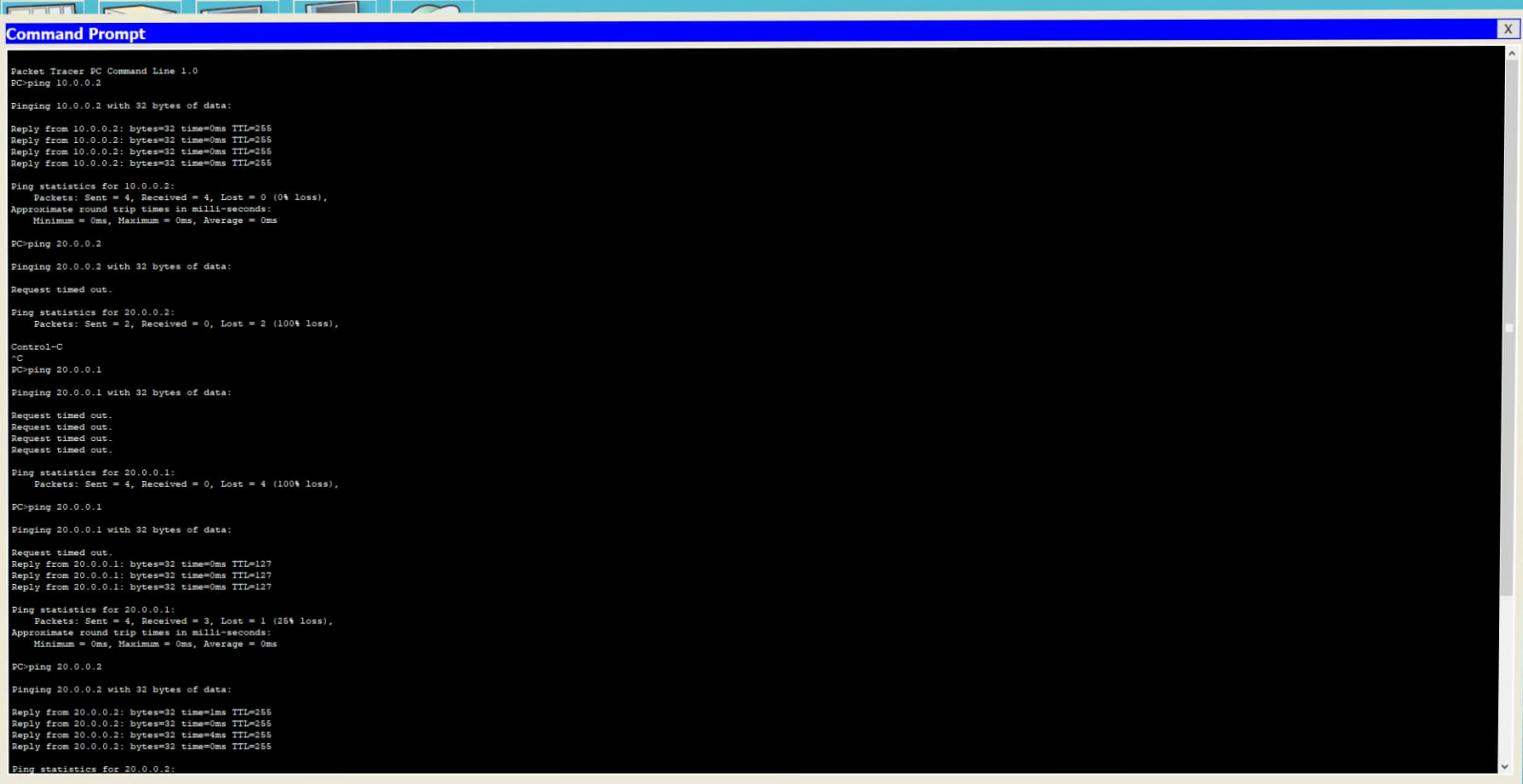
Result: Pinging 20.0.0.1 with 32 bytes of data

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

Ping statistics for 20.0.0.1
Packets: Sent = 4, Received = 3, Lost = 1
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximus = 0ms, Average = 0ms

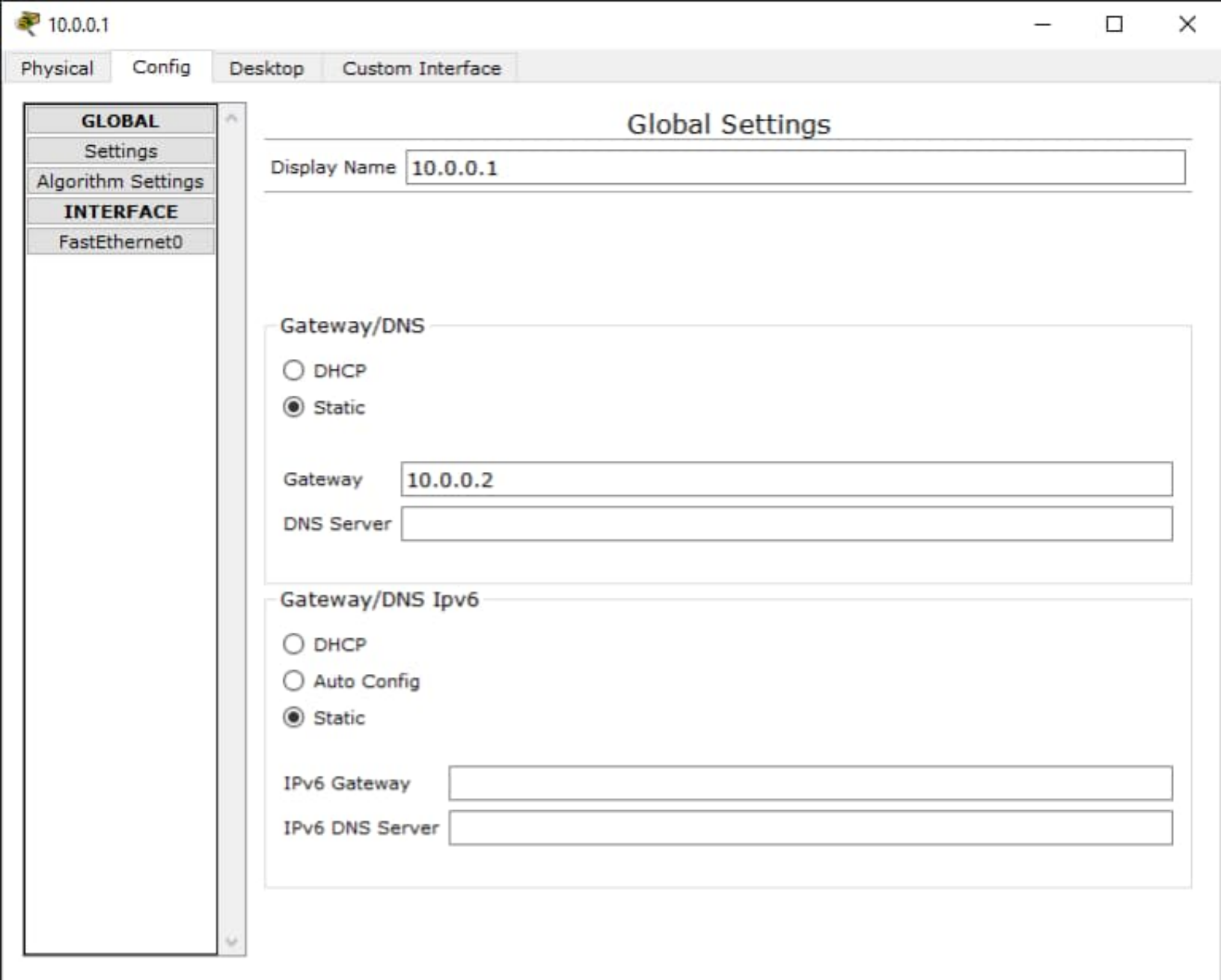
Neelima
13/11/2022


Fa0/0
Fa1/0
Fa0
PC-PT
10.0.0.1
Fa0
PC-PT
20.0.0.1


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=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
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 = 0ms, Average = 0ms
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Ping statistics for 20.0.0.2:
Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),
Control-C
^C
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
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=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 = 0ms, Average = 0ms
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=1ms TTL=255
Reply from 20.0.0.2: bytes=32 time=0ms TTL=255
Reply from 20.0.0.2: bytes=32 time=4ms TTL=255
Reply from 20.0.0.2: bytes=32 time=0ms TTL=255
Ping statistics for 20.0.0.2:

Command Prompt

```
Request timed out.

Ping statistics for 20.0.0.2:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),

Control-C
^C
PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

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

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

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=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 = 0ms, Average = 0ms

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=1ms TTL=255
Reply from 20.0.0.2: bytes=32 time=0ms TTL=255
Reply from 20.0.0.2: bytes=32 time=4ms TTL=255
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 = 4ms, Average = 1ms

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
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=5ms 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 = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 5ms, Average = 1ms

PC>
```


10.0.0.1
Physical
Config
Desktop
Custom Interface
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
Global Settings
Display Name
10.0.0.1
Gateway/DNS
DHCP
Static
Gateway
10.0.0.2
DNS Server
Gateway/DNS Ipv6
DHCP
Auto Config
Static
IPv6 Gateway
IPv6 DNS Server

Router0

Physical | Config | CLI

IOS Command Line Interface

```
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)


         --- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: n


Press RETURN to get started!



Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
exit
Router(config)#interface Fa1/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
exit
Router(config)#
```

Copy | Paste

Week - 4

Date 24/11/22
Page

⇒ Aim

Configuring ~~default~~ static Route to Router.

⇒ Procedure

- Connect 2 end devices through 3 generic routers
- Three routers are connected using serial DCE using serial ports of routers
- Router to end device connection is via copper cross over using Fast Ethernet ports
- Configure the IP address of the end devices.
- IP address of the routers are configured in the CLI using following commands

enable
config terminal
ip address 10.0.0.2 255.0.0.0
(255.0.0.0 : subnet mask)

no shutdown

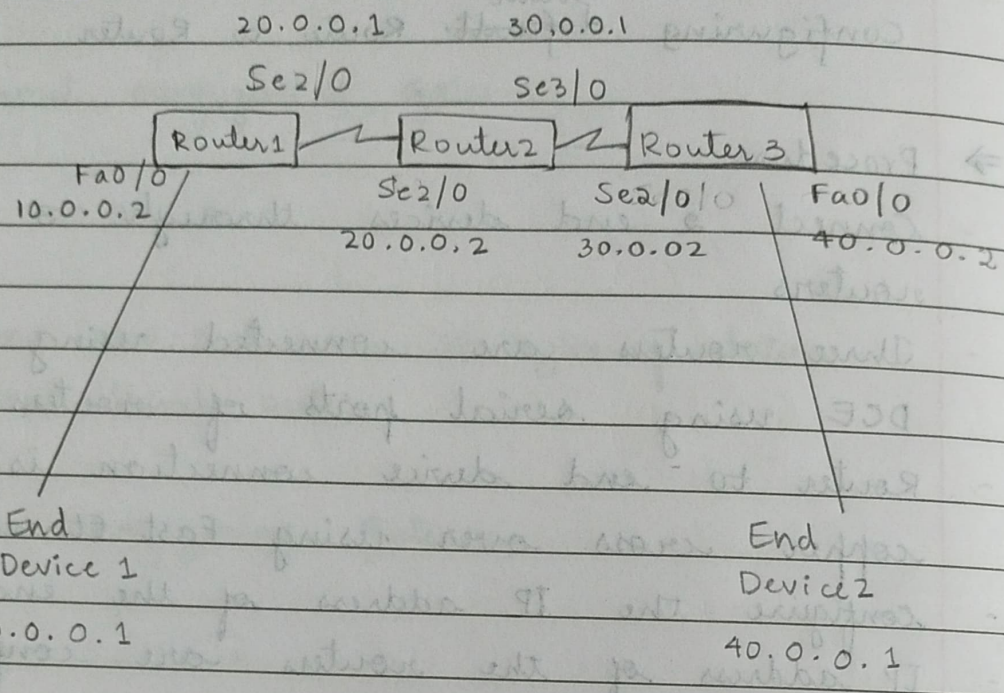
- Gateway for the end device is configured in the config tab of that device
- The routes for the routers is configured using CLI

R2(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2
(30.0.0.2 : Next Hop)

* show ip route : This shows the connected connections directly.

This command should be executed in privileage mode.

⇒ Topology :



Observation :

→ Before the routes for Routers was set up :

- Ping to an ip address within the same network will give the successful message.
- Ping to an ip address which is not in the same network but the gateway exists, the message / response is ~~Request~~ success message
- Ping to an ip address which is not in the same network but there isn't anyway to reach it, then the response is 'Request timed out'.
- Ping to an ip address which is in not in the same network, the response observed is 'Destination host unreachable'

Outcome :

⇒ Before setting up the routes for routers -

From End Device 1 (10.0.0.1)

ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data

Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 255

Ping statistics for 10.0.0.2

Packets : Sent = 4, Received = 4, Lost = 0

Approximate round trip times in ms

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

ping 40.0.0.1

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)

ping 30.0.0.1

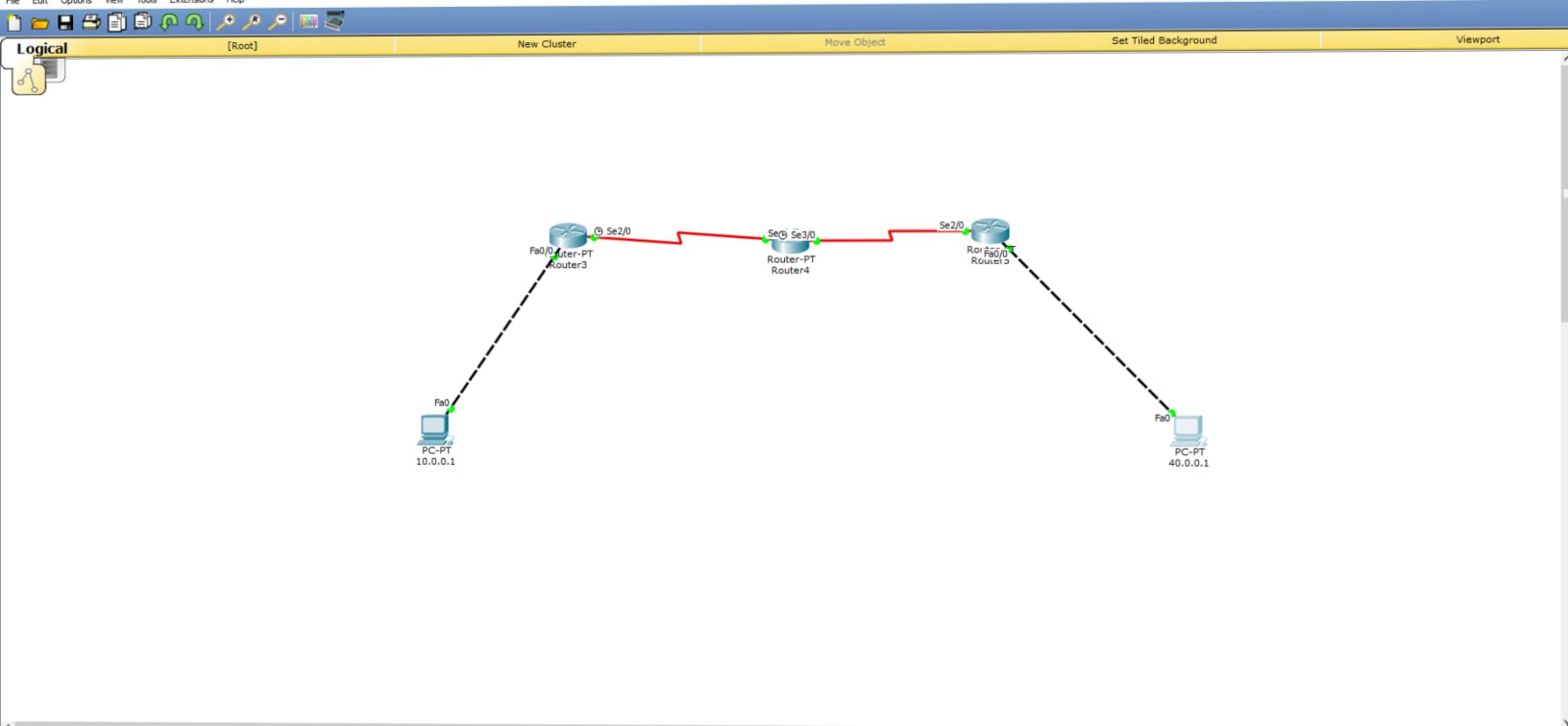
Pinging 30.0..0.1 with 32 bytes of data :

Request time out

⇒ After setting up routes for Routers :

Each & every ping will give the successful response.

Akshaia
24/11/2022


File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/0
Se2/0
Router-PT
Router3
Se2/0
Se3/0
Router-PT
Router4
Se2/0
Fa0/0
Router-PT
Router5
Fa0
PC-PT
10.0.0.1
Fa0
PC-PT
40.0.0.1

Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.000	--	10.0.0.1	ICMP	
	0.001	10.0.0.1	Router3	ICMP	
	0.002	Router3	Router4	ICMP	
	0.003	Router4	Router5	ICMP	
	0.004	Router5	40.0.0.1	ICMP	
	0.005	40.0.0.1	Router5	ICMP	
	0.006	Router5	Router4	ICMP	
	0.007	Router4	Router3	ICMP	
	0.008	Router3	10.0.0.1	ICMP	

Reset Simulation | Constant Delay | Captured to: 150.260 s

Play Controls

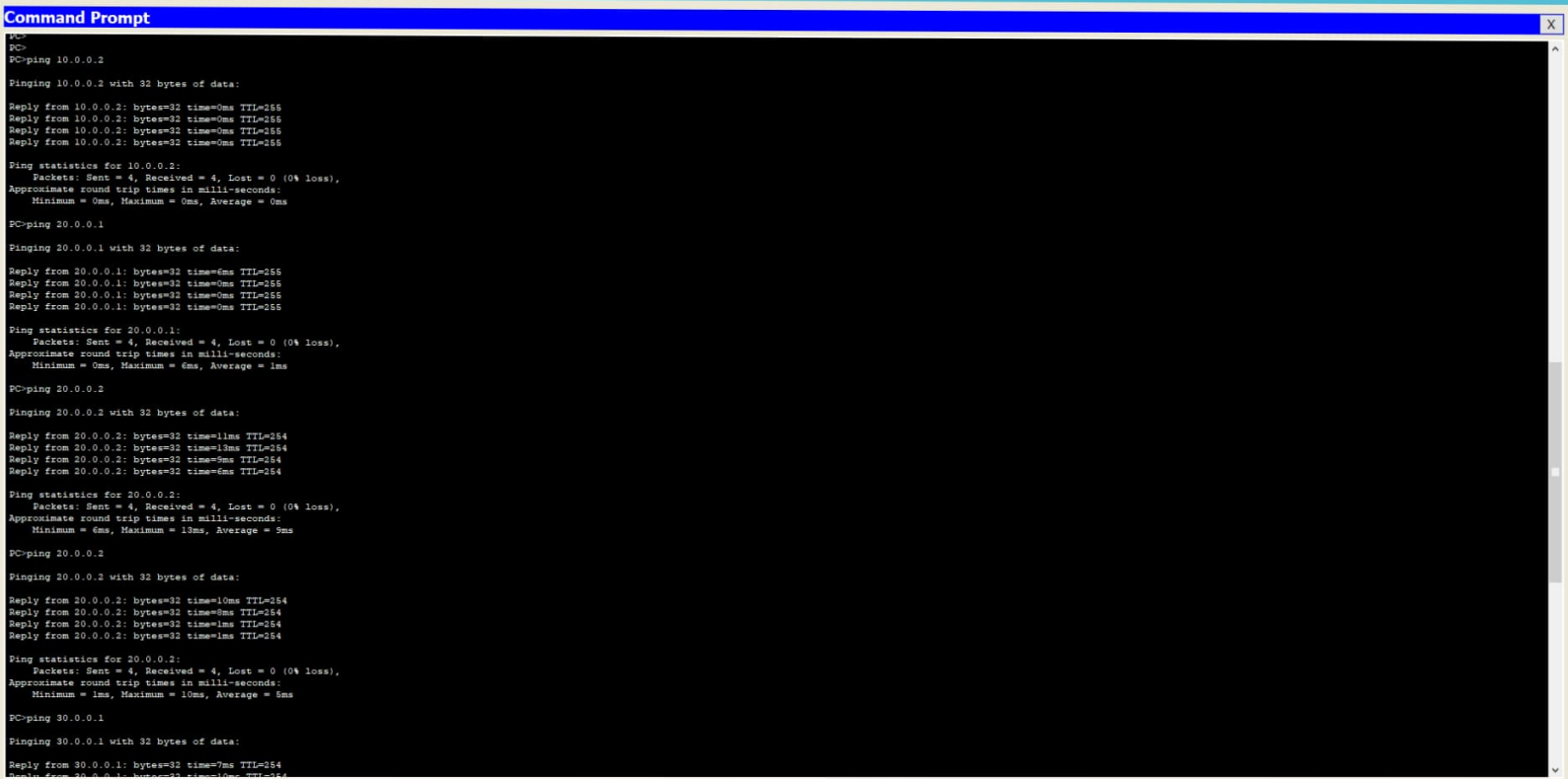
Back | Auto Capture / Play | Capture / Forward

Event List Filters - Visible Events

ARP, ICMP

Edit Filters | Show All/None


Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.2
Pinging 40.0.0.2 with 32 bytes of data:
Reply from 40.0.0.2: bytes=32 time=0ms TTL=255
Reply from 40.0.0.2: bytes=32 time=0ms TTL=255
Reply from 40.0.0.2: bytes=32 time=1ms TTL=255
Reply from 40.0.0.2: bytes=32 time=0ms TTL=255
Ping statistics for 40.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=0ms TTL=255
Reply from 30.0.0.2: bytes=32 time=0ms TTL=255
Reply from 30.0.0.2: bytes=32 time=0ms TTL=255
Reply from 30.0.0.2: bytes=32 time=0ms TTL=255
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 40.0.0.2: Destination host unreachable.
Reply from 40.0.0.2: Destination host unreachable.
Reply from 40.0.0.2: Destination host unreachable.
Reply from 40.0.0.2: Destination host unreachable.
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
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>


Command Prompt
X
PC>
PC>
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=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
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 = 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=6ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
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 = 6ms, Average = 1ms
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=11ms TTL=254
Reply from 20.0.0.2: bytes=32 time=13ms TTL=254
Reply from 20.0.0.2: bytes=32 time=9ms TTL=254
Reply from 20.0.0.2: bytes=32 time=6ms TTL=254
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 13ms, Average = 9ms
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=10ms TTL=254
Reply from 20.0.0.2: bytes=32 time=8ms TTL=254
Reply from 20.0.0.2: bytes=32 time=1ms TTL=254
Reply from 20.0.0.2: bytes=32 time=1ms TTL=254
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 10ms, Average = 5ms
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=7ms TTL=254


Command Prompt
X
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 10ms, Average = 5ms
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=7ms TTL=254
Reply from 30.0.0.1: bytes=32 time=10ms TTL=254
Reply from 30.0.0.1: bytes=32 time=8ms TTL=254
Reply from 30.0.0.1: bytes=32 time=10ms TTL=254
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 10ms, Average = 8ms
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=18ms TTL=253
Reply from 30.0.0.2: bytes=32 time=16ms TTL=253
Reply from 30.0.0.2: bytes=32 time=14ms TTL=253
Reply from 30.0.0.2: bytes=32 time=13ms TTL=253
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 13ms, Maximum = 18ms, Average = 15ms
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=18ms TTL=125
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 = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 18ms, Average = 9ms
PC>ping 40.0.0.2
Pinging 40.0.0.2 with 32 bytes of data:
Reply from 40.0.0.2: bytes=32 time=7ms TTL=253
Reply from 40.0.0.2: bytes=32 time=9ms TTL=253
Reply from 40.0.0.2: bytes=32 time=13ms TTL=253
Reply from 40.0.0.2: bytes=32 time=11ms TTL=253
Ping statistics for 40.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 13ms, Average = 10ms
PC>

Date 01/12/22
Page

Week - 5

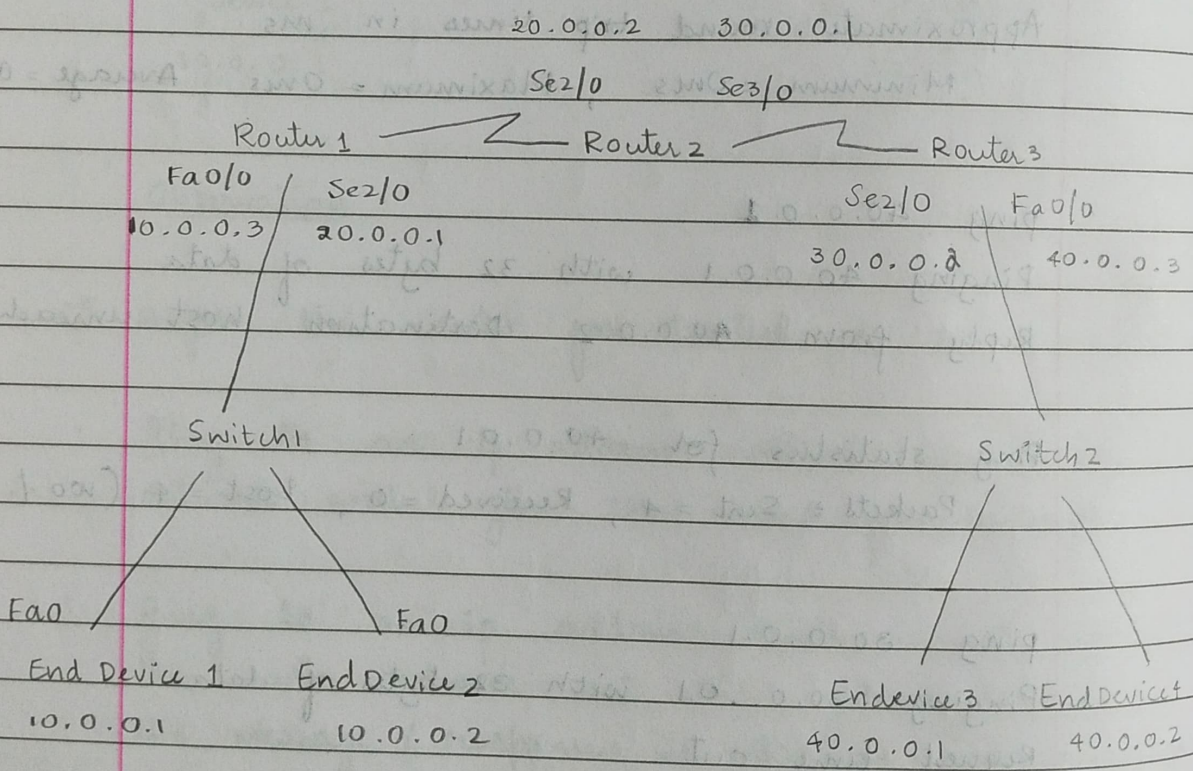
⇒ Aim

Configuring default route to routers

⇒ Procedures :

- Connect the end devices to a switch and then to routers.
- Configure the IP address for the end devices

Topology :



Observation :

- Now, the gateway for end devices is not set.
- The ping response from ED1 to 40.0.0.1 is 'Destination host unreachable'
- The ping response from ED1 to 20.0.0.1, ~~reqf~~

* After the default gateway for the end devices is set.

Date __/__/__
Page _____

- The ping response from ED1 to 10.0.0.3, 20.0.0.1 is successfull
- But for 20.0.0.2, 30.0.0.1, the ping response is request timed out because default route for the router is not set.

⇒ After the default routes for the ~~rand~~ routers is set.

from ED 2

ping 10.0.0.3

Pinging 10.0.0.03 with 32 bytes of data
Reply from 10.0.0.3 : bytes = 32 time = 0ms TTL = 255

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 = 0ms, Average = 0ms

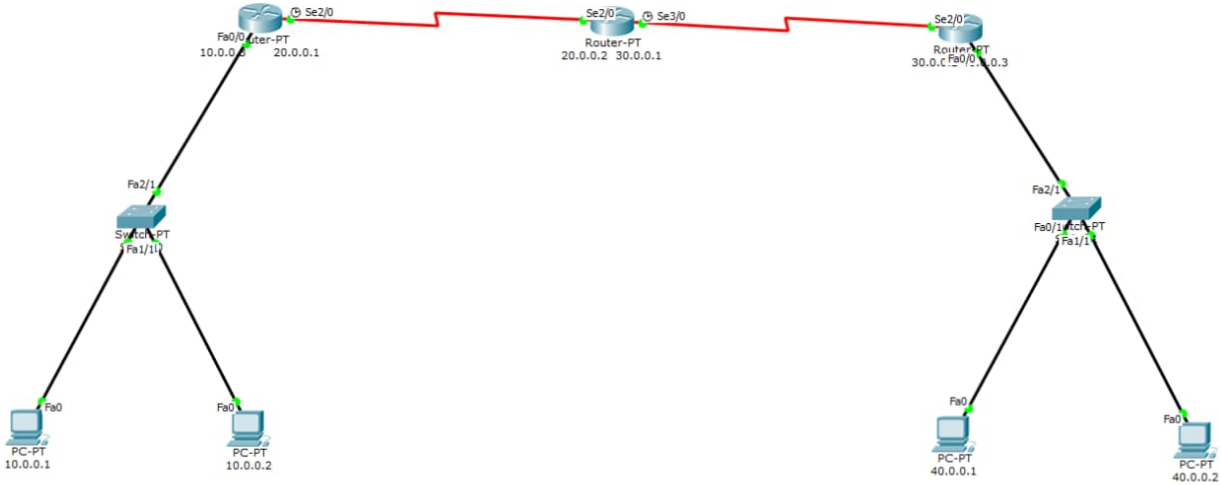
ping 20.0.0.1

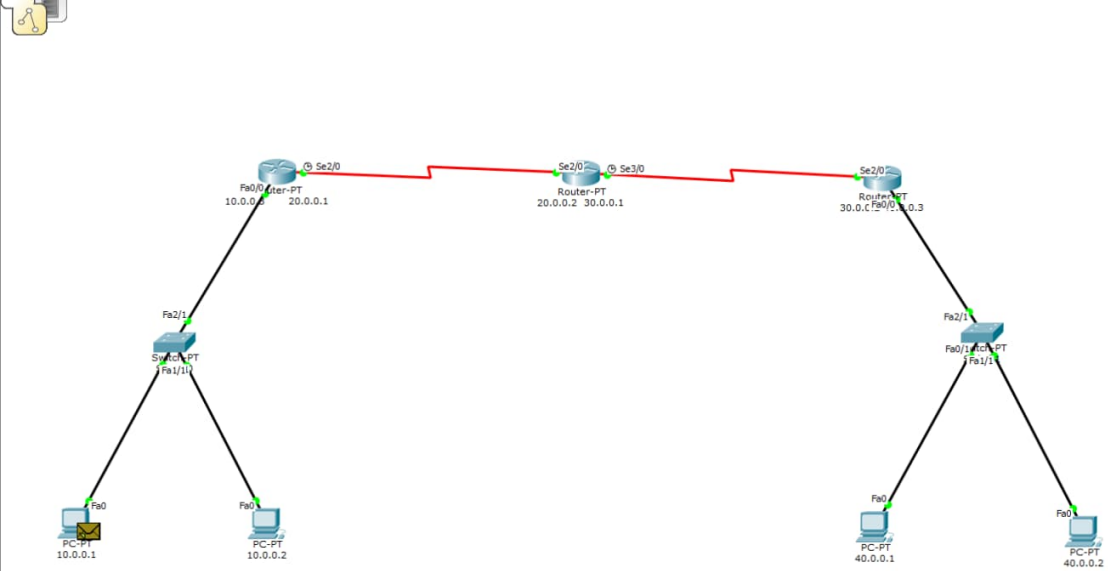
Pinging 20.0.0.1 with 32 bytes of data :
Reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 255
Ping statistics for 20.0.0.1
Packets : Sent = 4, Received = 4, Lost = 0 (0% lost)
Approximate round trip times in milli-seconds :
Minimum = 0ms, Maximum = 1ms, Average = 0ms

→ Before setting route :

ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data
Reply from 40.0.0.1 : Destination host unreachable


Se2/0
Fa0/0
20.0.0.1
Se2/0
Se3/0
Router-PT
20.0.0.2
30.0.0.1
Se2/0
Router-PT
Fa0/0
Fa2/1
Switch-PT
Fa1/1
Fa2/1
Fa0/1
Switch-PT
Fa1/1
Fa0
PC-PT
10.0.0.1
Fa0
PC-PT
10.0.0.2
Fa0
PC-PT
40.0.0.1
Fa0
PC-PT
40.0.0.2



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.004	20.0.0.2 ...	30.0.0.2...	ICMP	
	0.005	30.0.0.2...	Switch1	ICMP	
	0.006	Switch1	40.0.0.1	ICMP	
	0.007	40.0.0.1	Switch1	ICMP	
	0.008	Switch1	30.0.0.2...	ICMP	
	0.009	30.0.0.2...	20.0.0.2 ...	ICMP	
	0.010	20.0.0.2 ...	10.0.0.3 ...	ICMP	
	0.011	10.0.0.3 ...	Switch0	ICMP	
	0.012	Switch0	10.0.0.1	ICMP	

Reset Simulation | Constant Delay | Captured to: 0.012 s

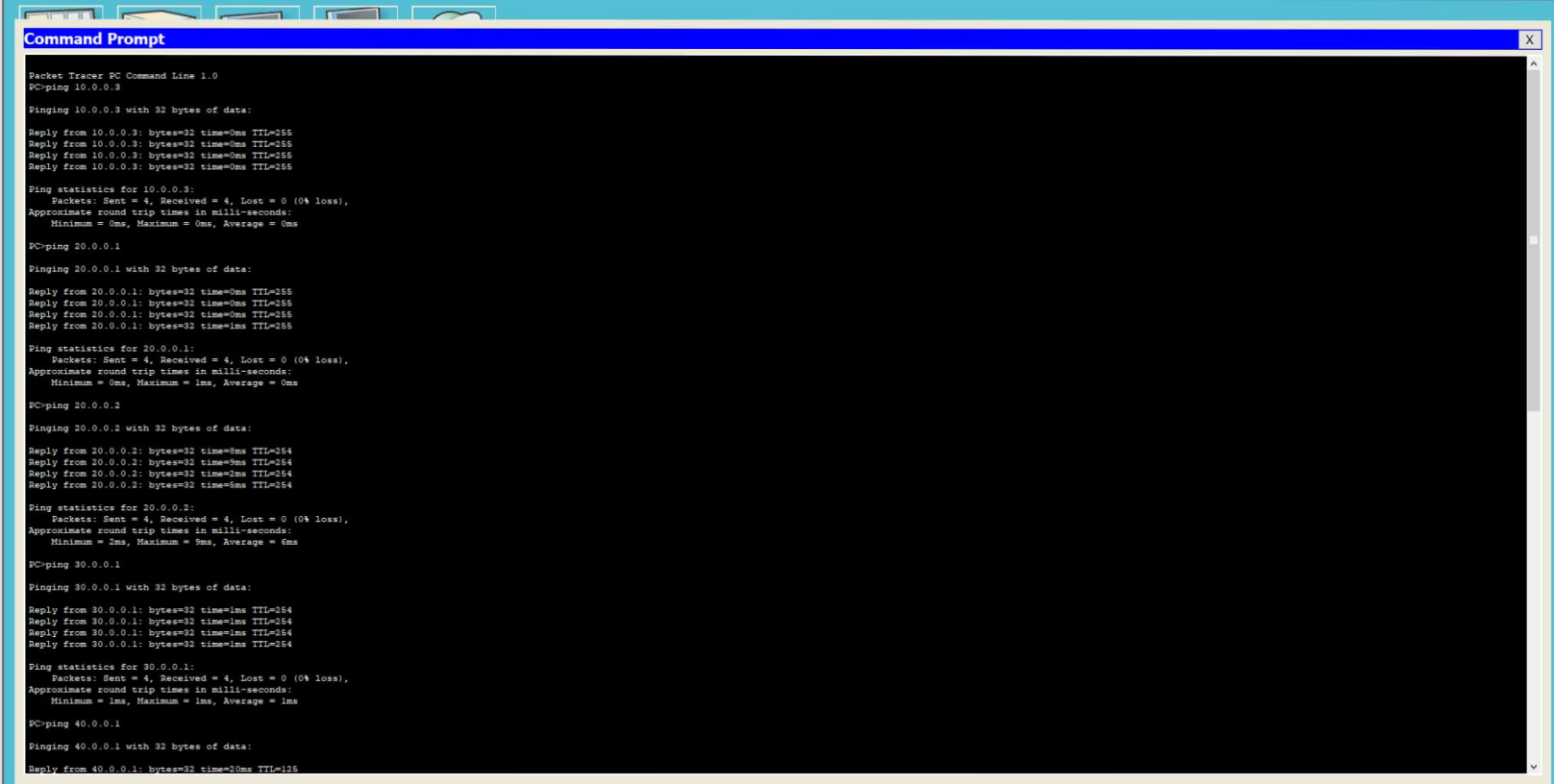
Play Controls

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


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=0ms TTL=255
Reply from 10.0.0.3: bytes=32 time=0ms TTL=255
Reply from 10.0.0.3: bytes=32 time=0ms TTL=255
Reply from 10.0.0.3: bytes=32 time=0ms TTL=255
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 = 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=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=1ms TTL=255
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>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 20.0.0.2: bytes=32 time=8ms TTL=254
Reply from 20.0.0.2: bytes=32 time=9ms TTL=254
Reply from 20.0.0.2: bytes=32 time=2ms TTL=254
Reply from 20.0.0.2: bytes=32 time=5ms TTL=254
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 9ms, Average = 6ms
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 1ms, Average = 1ms
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=20ms TTL=125

Week - 5

Date 8/12/22
Page

Aim: Configuring RIP routing protocol in router

Procedure:

- Connect 2 end devices to 2 switches each & configure the ip address.
- Then connect the 2 switches to 2 routers and connect the routers via another router.
- configure ip address of router switch interface
- configure the ip address of router to router interface using point-to-point protocol

```
enable
config terminal
interface Sc2/0
ip address 20.0.0.2 255.0.0.0
encapsulation PPP
clock rate 64000
no shutdown
```

If the above is done for router 1, then no need of setting clock rate at router 2 interface

- Then we have to configure default route to each ~~other~~ router

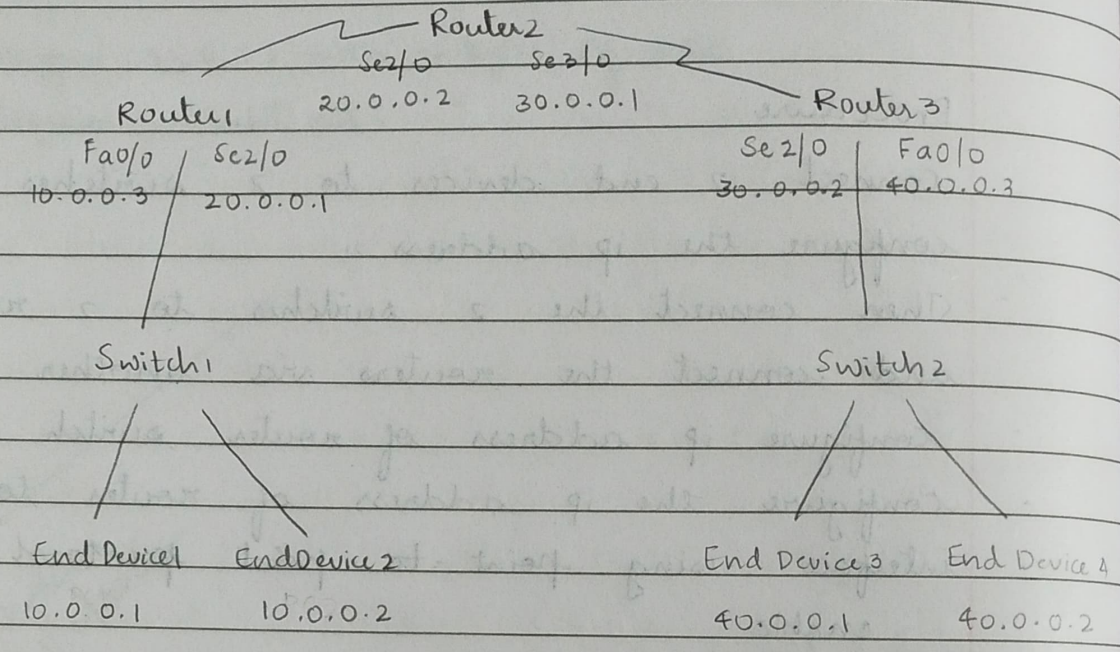
```
ip route 0.0.0.0 0.0.0.0 20.0.0.2
```

- configure Rip protocal for each router using the following command. In config mode,

```
router rip
network 10.0.0.0
network 20.0.0.0
```

Date __/__/__
Page

Topology :



Observation :

- We use PPP protocol while configuring ip address of interface of 2 routers
- PPP is point to - point protocol of the data link layer that is used to transmit multiprotocol data between two directly connected devices.
- Encapsulation PPP → This command encapsulates the datagram before it is transmitted & specifies the physical layer to transmit to.
- RIP (Route Information Protocol) is a distance vector protcol that uses hop count as its primary metric. This protocol defines how routers should share information when moving traffic

RESULTS:

From End Device 2 (10.0.0.2)

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 = 128
Reply from 10.0.0.1 : bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.1 : bytes = 32 time = 0ms TTL = 128

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 = 0ms, Average = 0ms

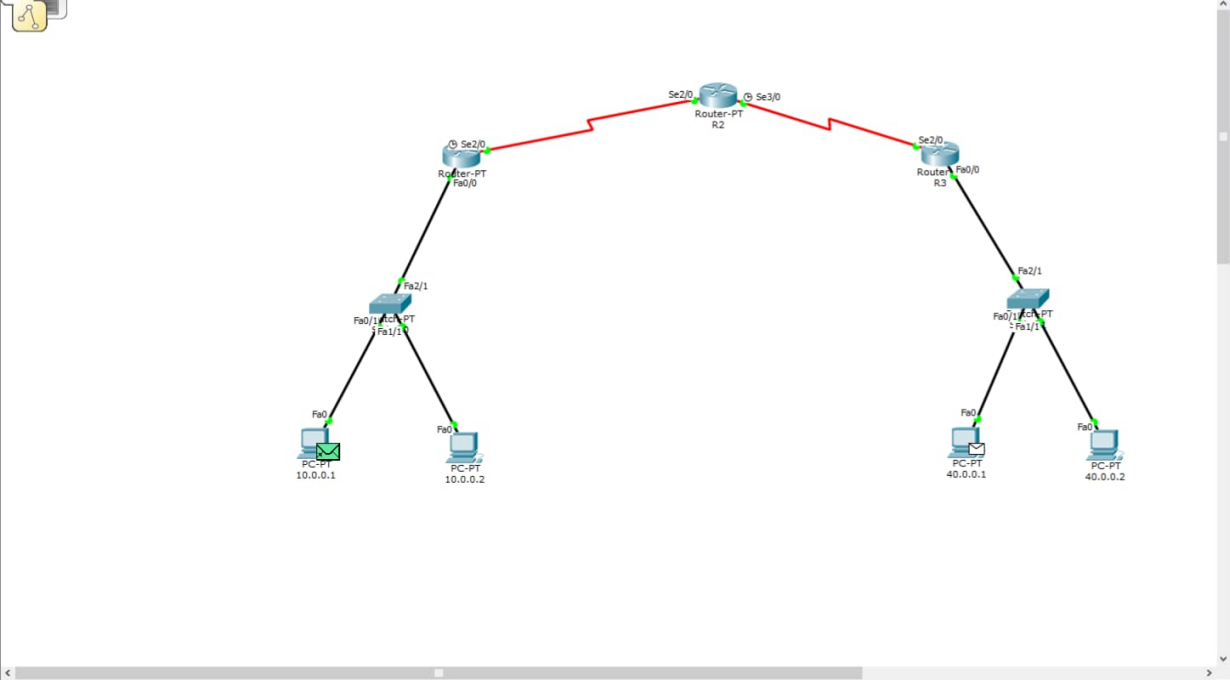
ping 40.0.0.2

⇒ Pinging 40.0.0.2 with 32 bytes of data:
Request timed out
Reply from 40.0.0.2 : bytes = 32 time = 11ms TTL = 125
"
"

Ping statistics for 40.0.0.2:
Packets: sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds
Minimum = 9ms, Maximum = 17ms, Average = 12ms

We get successful responses from all intfaces

Neelima
8/12/2022



Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.004	R2	R3	ICMP	
	0.005	R3	Switch1	ICMP	
	0.006	Switch1	40.0.0.1	ICMP	
	0.007	40.0.0.1	Switch1	ICMP	
	0.008	Switch1	R3	ICMP	
	0.009	R3	R2	ICMP	
	0.010	R2	R1	ICMP	
	0.011	R1	Switch0	ICMP	
	0.012	Switch0	10.0.0.1	ICMP	

Reset Simulation | Constant Delay | Captured to: 300.043 s

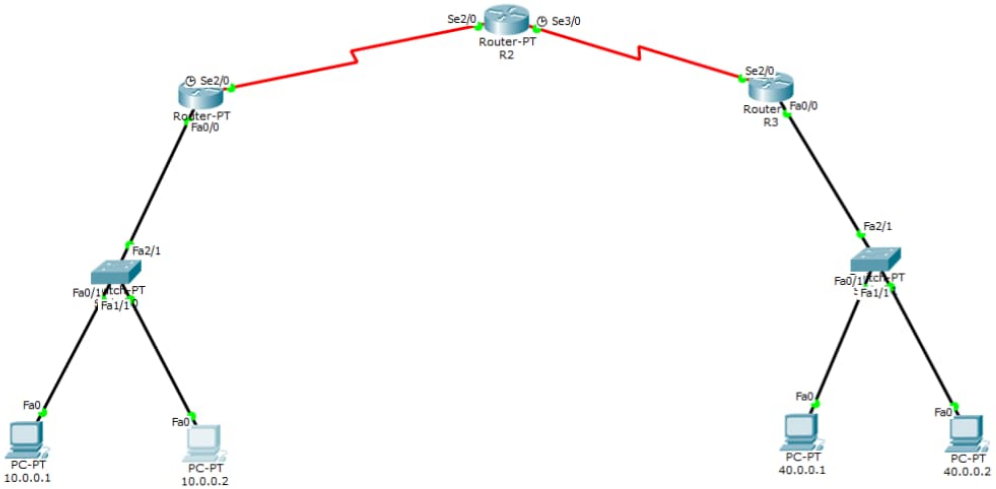
Play Controls

Back | Auto Capture / Play | Capture / Forward

Event List Filters - Visible Events

ARP, ICMP

Edit Filters | Show All/None


Se2/0
Se3/0
Router-PT
R2
Se2/0
Router-PT
Fa0/0
Se2/0
Router
R3
Fa0/0
Fa2/1
Fa0/1
Fa1/1
Fa2/1
Fa0/1
Fa1/1
Fa0
Fa0
Fa0
Fa0
PC-PT
10.0.0.1
PC-PT
10.0.0.2
PC-PT
40.0.0.1
PC-PT
40.0.0.2

Week - 6

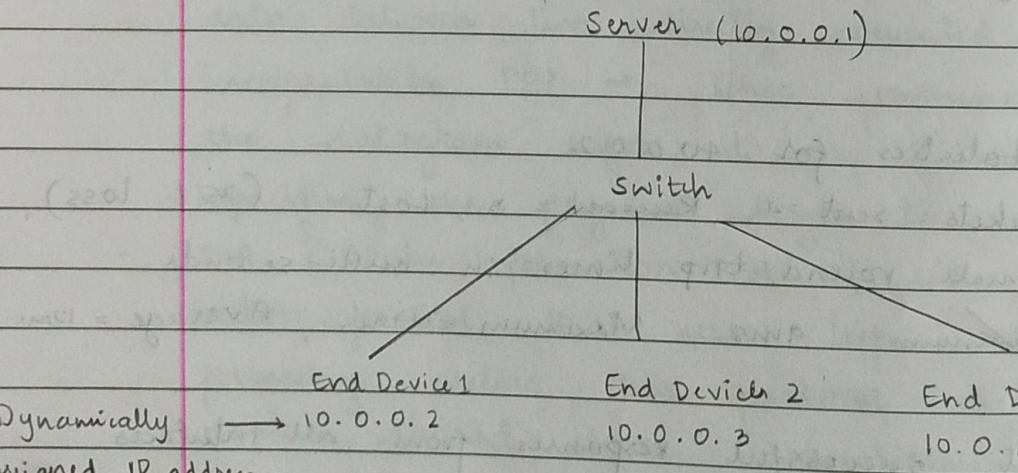
Date 15/12/22
Page

Aim: Configuring DHCP

Procedure:

- Connect end devices to a switch and then to a server using automatic connections.
- Configure the IP address for the server. Under services, select DHCP, switch on the service, enter the start IP address and then save.
- And then, dynamically assign the IP address for all end devices
  Click on End Device → Interface → choose DHCP under IP configuration
  This dynamically assigns the IP address according to start IP address.
  Similarly, assign IP address to all end devices

Topology:



Observation:

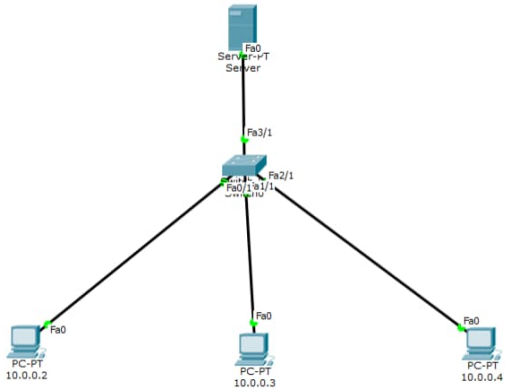
- After configuring IP address of the server, the End devices will be assigned IP addresses

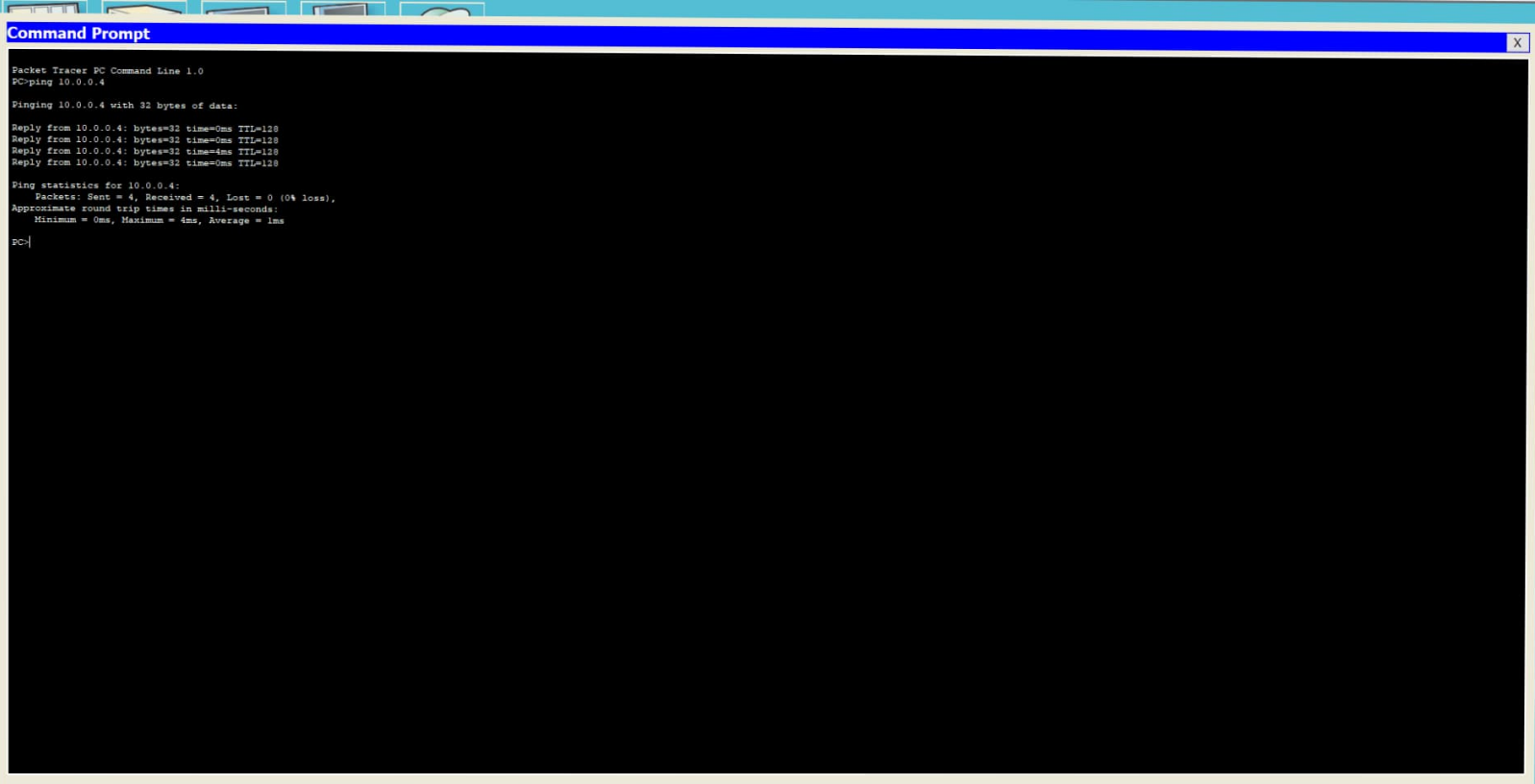
dynamically by choosing DHCP.

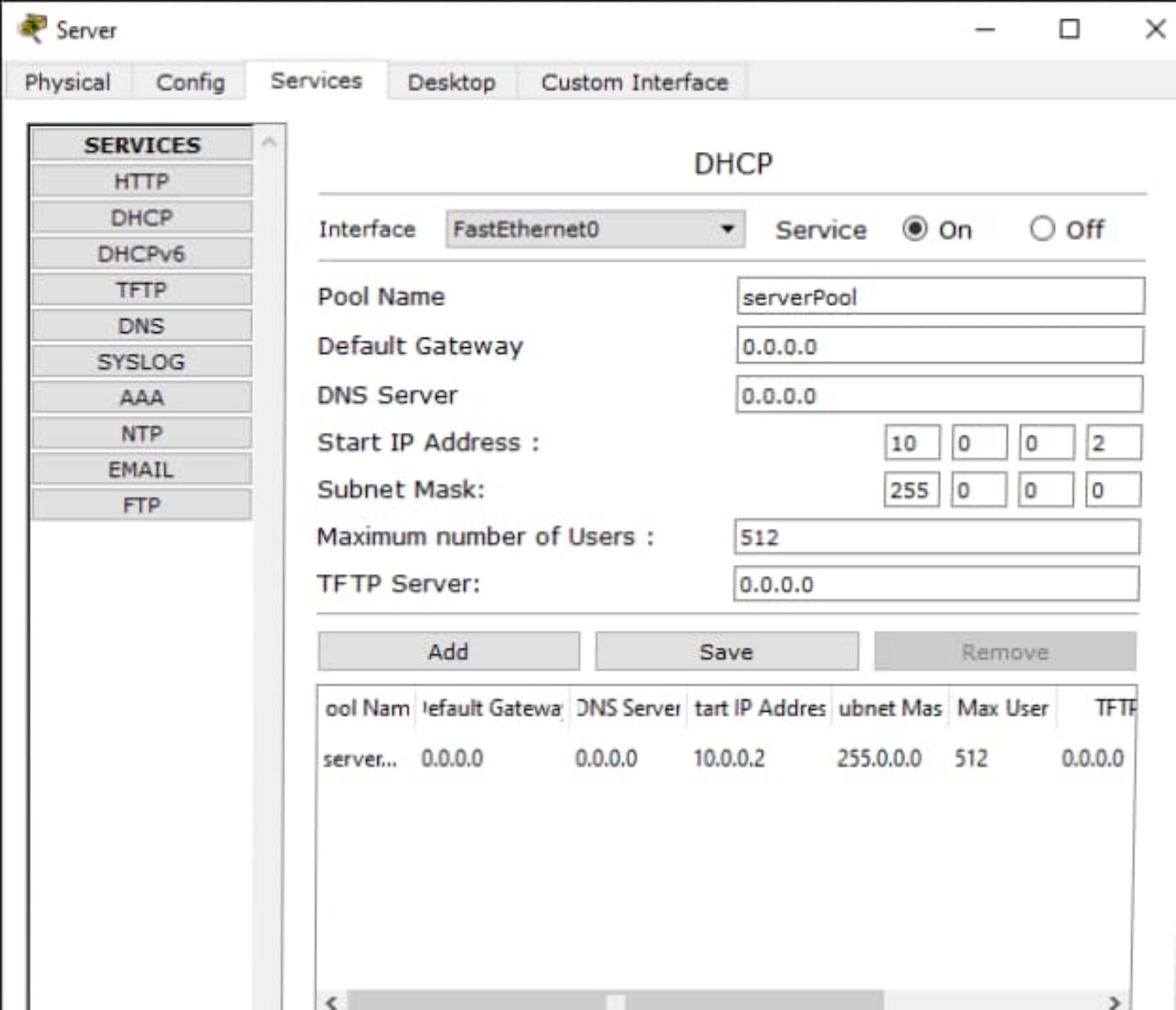
Result :

DHCP : Dynamic host configuration protocol.
It is a protocol used to dynamically assign IP addresses to end devices.

Neelema
15/12/2022


Fa0
Server-PT
Server
Fa3/1
Fa2/1
Fa0
PC-PT
10.0.0.2
Fa0
PC-PT
10.0.0.3
Fa0
PC-PT
10.0.0.4


Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=4ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 1ms
PC>


Server
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DHCP
Interface
FastEthernet0
Service
On
Off
Pool Name
serverPool
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
Start IP Address :
10
0
0
2
Subnet Mask:
255
0
0
0
Maximum number of Users :
512
TFTP Server:
0.0.0.0
Add
Save
Remove
ool Nam
efault Gatewa
DNS Server
tart IP Addres
ubnet Mas
Max User
TFTF
server...
0.0.0.0
0.0.0.0
10.0.0.2
255.0.0.0
512
0.0.0.0

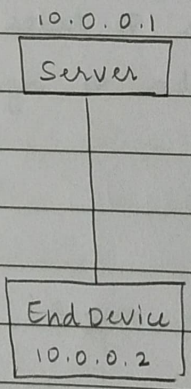
Date 15/12/22
Page

Aim : Configure Web Server and DNS Server

Procedure :

- Connect a generic end device to a server
- Configure the IP address of the server
- Configure IP address of the end devices statically
- Go to services of the server & choose DNS, switch on the service, provide the IP address and name and then save it

Topology :



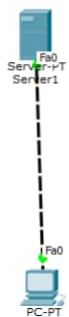
Neelima
15/12/2022

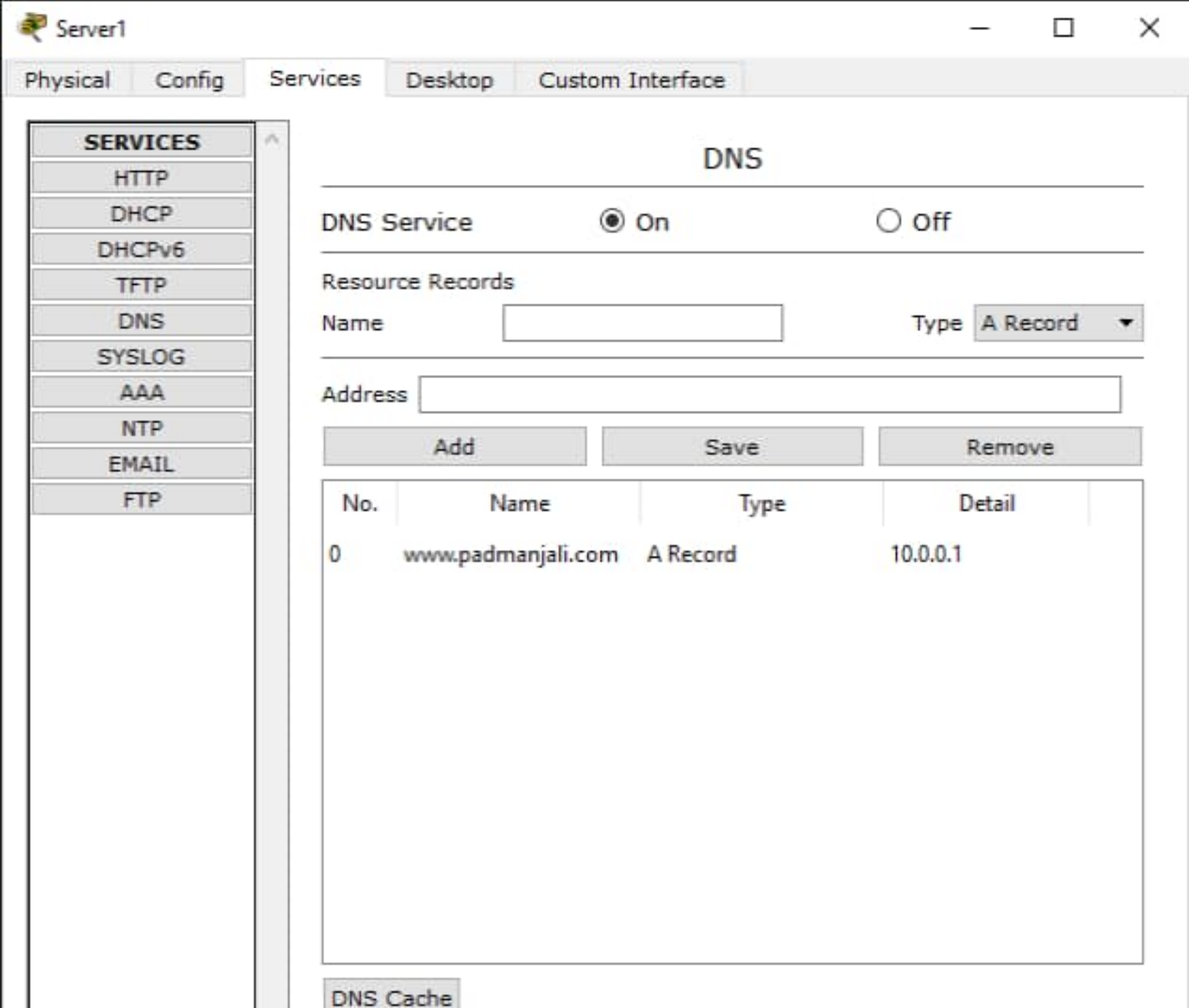
Observation :

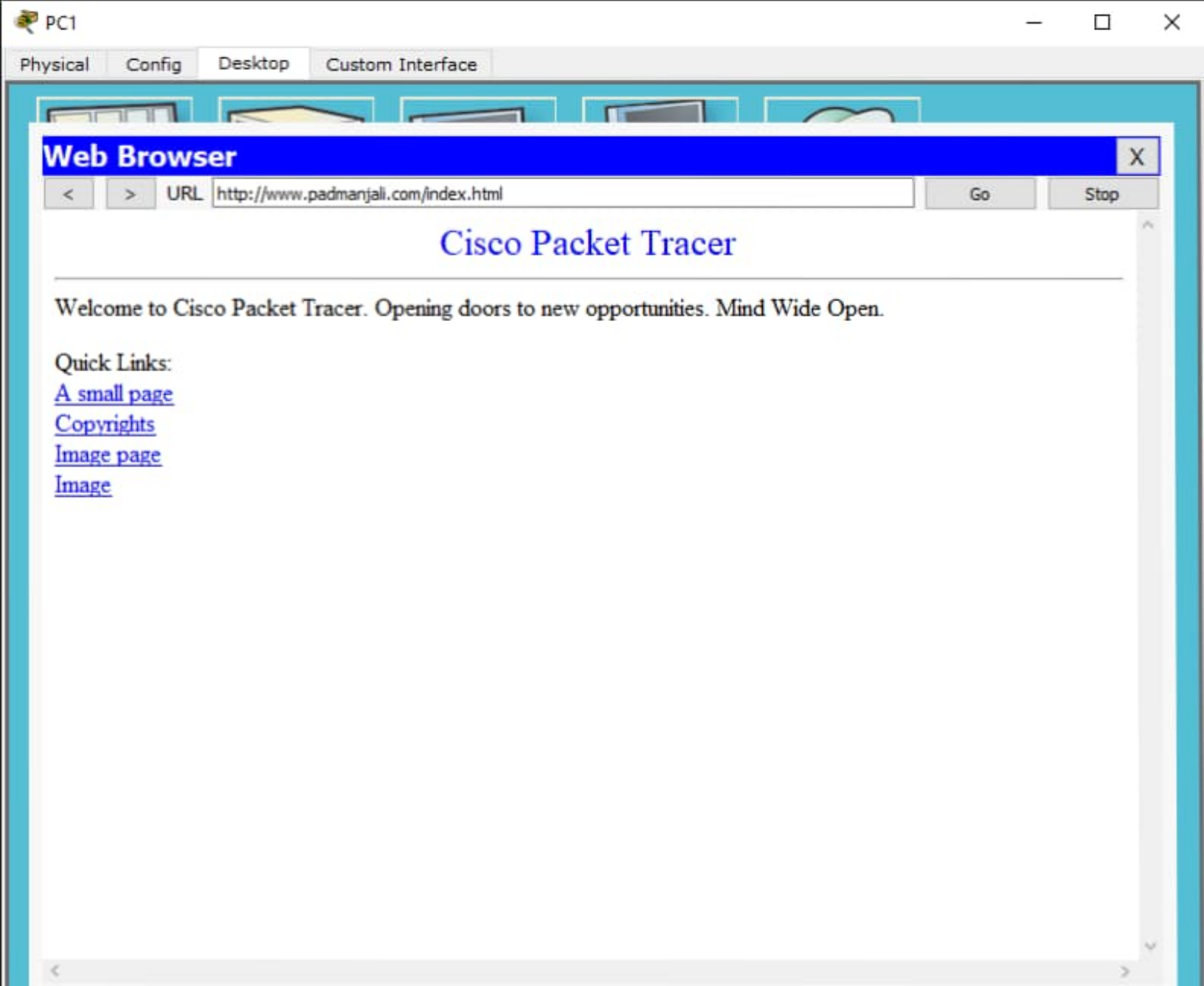
In the web ~~server~~ → browser of the end device, type the url of the server name which was configured & you get a web page sample.

Result :

DNS : Domain Name service is used to map the name of the service to its IP address
Ex: www.myweb.com can be assigned to IP address 10.0.0.1


Fa0
Server-PT
Server1
Fa0
PC-PT
PC1


Server1
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
A Record
Address
Add
Save
Remove
No.
Name
Type
Detail
0
www.padmanjali.com
A Record
10.0.0.1
DNS Cache


PC1
Physical
Config
Desktop
Custom Interface
Web Browser
X
<
>
URL
http://www.padmanjali.com/index.html
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

Week-7

Date __/__/__
Page

Write a program to Error detection using CRC-CCITT (16 bits)

```
#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 data 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("Error detected\n");
    else
        printf("No error detected");
}
```

```
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; j++)
            check_value[j] = check_value[j+1];
        check_value[j] = data[i++];
    } while (i <= data_length + N - 1);
}

int main()
{
    printf("Enter data to be transmitted");
    scanf("%s", data);
    printf("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';
    crc();
    for (i = data_length; i < data_length + N - 1; i++)
        data[i] = check_value[i - data_length];
    receiver();
    return 0;
}
```

Output :

Enter the data to be transmitted : 1011010101

Enter generating polynomial : 1011

Data padded with n-1 zeros : 1011010101000

CRC or check value is : 101

Final data to be sent : 1011010101101

Enter the received data : 1011010101101

Data received : 1011010101101

No error Detected.

```
Enter the data :
11010110
Enter the generator :
1010
CRC is
010
Checksum code :
11010110010
Data transmitted is valid
```

Week 8

Date 5/09/23
Page

Write a program for congestion control using Leaky Bucket algorithm

```
import random

packetcount = 0
buffer = 0
maxValue = random.choice([10, 20, 40, 80, 160])
packetSize = random.choice([10, 20, 40, 80, 160])

while buffer <= maxValue :
    buffer += packetSize
    packetCount += 1
    if buffer > maxValue :
        break

print ("max Value :", maxValue)
print ("packet size :", packetSize)
print ("packetCount :", packetCount - 1)
```

Java :

```
import java.util.*;
class LeakyBucket {
    public static void main (String args[])
    {
        int no_queries, buffer, output_pkt, size;
        int input_pkt_size, bucket_size, size_left;

        buffer = 0;
        Scanner read = new Scanner (System.in);
        System.out.println ("Enter queries");
        no_queries = read.nextInt();
        System.out.println ("Enter bucket size");
```

```
bucket_size = read.nextInt();
System.out.println("Enter output packet size");
output_pkt_size = read.nextInt();
for (int i=0 ; i < no_queries ; i++){
    input_pkt_size = read.nextInt();
    size_left = bucket_size - buffer;
    if (input_pkt_size <= (size_left)){
        buffer += input_pkt_size;
    }
    else
    {
        System.out.println("Packet loss = "
                +input_pkt_size);
    }
    System.out.println("Buffer Size " + buffer
            + " out of bucket size="
            + bucket_size);
    buffer -= out_pkt_size;
  }
 }
}
```

Output:

Enter Queries : 3

Enter Bucket size : 16

" Input packet size : 5

" output " " : 2

5/1/2023

```
Enter the number of packets
5
Enter the size of each packet
3
4
1
2
6
Clock   Packet Size       Accepted          Sent    Remaning
1               3              3          3       0
2               4              4          3       1
3               1              1          2       0
4               2              2          2       0
5               6              Dropped    0       0
```

Date 12/01/22
Page

Write a program for distance vector algorithm to find suitable path for transmission.

```
#include <stdio.h>
#include <stdlib.h>
int Bellman_Ford (int G[20][20], int V, int E,
                  int edge[20][12])
{
    int i, u, v, k, distance[20], parent[20], s, flag=1;
    for (i=0; i<V, i++)
        distance [i] = 1000, parent[i] = -1;
    printf("Enter source");
    scanf ("%d", &s);
    distance [s-1] = 0;

    for( i=0 ; i<V ; i++)
    {
        for(k=0 ; K<E ; K++)
        {
            u = edge[k][0], v = edge[k][1];
            if( distance[u] + G[u][v] < distance[v])
                distance[v] = distance[u] + G[u][v],
                    parent[v] = u;
        }
    }
    return flag;
}
int main()
{
    int V, edge[20][2], G[20][20], i, j, K=0;
    printf(" BELLMAN FORD\n");
    printf(" Enter no. of vertices: ");
    scanf("%d", &V);
```

```
printf ("Enter graph in matrix form \n");
for (i=0 ; i<v ; i++)
  for (j=0 ; j<v ; j++)
    {
      scanf ("%d", &G[i][j])
      if (G[i][j] != 0)
        edge[k][0] = i , edge[k++][1] = j;
    }
if (Bellman_Ford (G, V, K, edge))
    printf ("\n No negative weight cycle \n");
else
    printf ("\n Negative weight cycle exists \n");
```

Output :

```
BELLMAN FORD
Matrix :
0   22   99   99
2   0   99   3   99
2   99   0   6   4
99   3   6   0   5
 99   99   4   5   0
Enter source : 1

No negative weight cycle
```

12/1/2021

```
BELLMAN FORD
Enter no. of vertices: 5
Enter graph in matrix form:
0 6 5 0 0
0 0 0 -1 0
0 -2 0 4 3
0 0 0 0 3
0 0 0 0 0
Enter source: 0
Vertex 1 -> cost = 1000 parent = 0
Vertex 2 -> cost = 998 parent = 3
Vertex 3 -> cost = 1000 parent = 0
Vertex 4 -> cost = 997 parent = 2
Vertex 5 -> cost = 1000 parent = 0

No negative weight cycle
```

Date 12/1/23
Page

Implement Dijkstra's algorithm to compute the shortest path for a given topology

```
#include<stdio.h>
#include<conio.h>
#define INFINITY 9999
#define MAX 10

void dijkstra(int G[MAX][MAX], int n, int sn)
{
    for(i=0; i<n; i++)
        for(j=0; j<n; j++)
            if(G[i][j]==0)
                cost[i][j] = INFINITY;
            else cost[i][j] = G[i][j];

    for(i=0; i<n; i++)
    {
        distance[i] = cost[startnode][i];
        pred[i] = startnode;
        visited[i]= 0;
    }
    distance[startnode] =0;
    visited[startnode] =1;
    count =1;
    while(count < n-1)
    {
        mindistance = INFINITY;
        for(i=0; i<n; i++)
        {
            if distance[i] < mindistance && !visited[i]
            {
                mindistance = distance[i];
```

Date __/__/__
Page

```
            nextnode = i;
        }
    }
    visited[nextnode] = 1;
    for(i=0; i<n; i++)
        if(!visited[i]){
            if(mindistance[nextnode][i] <
                                distance[i])
            {
                distance[i] = mindistance + cost[nextnode]
                                                [i];
                pred[i] = nextnode;
            }
            count++;
        }
    for(i=0; i<n; i++)
        if(i != startnode)
        {
            printf("\nDistance of node %d = %d",
                    i, distance[i]);
            printf("\nPath = %d", i);
            j = i;
            do
            {
                j = pred[j];
                printf(" <- %d", j);
            } while(j != startnode);
        }
    }
```

Date __/__/__
Page _____

Output :

Enter no. of vertices : 4

Enter adjacency matrix :

0	5	999	999
2	0	4	999
999	999	0	6
4	7	5	0

Enter starting node : 0

Distance of node 1 = 5

Path = 1 ← 0

Distance of node 2 = 9

Path = 2 ← 1 ← 0

Distance of node 3 = 15

Path = 3 ← 2 ← 1 ← 0

```
Enter no. of vertices:5

Enter the adjacency matrix:
0 0 5 0 0
0 0 0 -1 0
0 -2
0 4 3
0 0 0 0 3
0 0 0 0 0

Enter the starting node:0

Distance of node1=9999
Path=1<-0
Distance of node2=5
Path=2<-0
Distance of node3=9999
Path=3<-0
Distance of node4=9999
Path=4<-0
```

LAB - 9

Date __/__/__
Page _____

Using TCP/IP sockets, write a client-server program to make client sending a file name & the server to send back the contents.

Server Side

```
import socket

server_socket = socket.socket(socket.AF_INET,
                              socket.SOCK_STREAM)
server_address = ('localhost', 12345)
server_socket.bind(server_address)

server_socket.listen(1)
print('Server is listening {}:{}'.format(*server_address))

While True:
    print('Waiting for connection')
    client_socket, client_add = server_socket.accept()
    print('Accepted Connection {}:{}',format(*client_add))

    try:
        file_name = client_socket.recv(1024).decode()
        with open(file_name, 'rb') as file:
            filecontents = file.read()
            client_socket.sendall(filecontents)
    except FileNotFoundError:
        ermsg = 'File not found\n'.encode()
        client_socket.sendall(ermsg)
    finally:
        client_socket.close()
```

Client side

```
import socket as s

client_socket = s.socket(s.AF_INET, s.SOCK_STREAM)
server_add = ('localhost', 12345)
client_socket.connect(server_add)

file_name = input("Enter file name").encode()
client_socket.sendall(file_name)

// Receive
fileContents = b''
while True:
    data = client_socket.recv(1024)
    if not data:
        break
    fileContents += data
print(fileContents.decode())
```

Output:

Server

```
Server is listening 127.0.0.1 : 1234
Waiting for connection
Accepted connection 37345
Waiting for connection
```

Client

```
Enter the file name:
sample.txt
Hi, Good Evening
```

```
ata\Local\Programs\Python\Python311\python.exe' 'c:\Users\
Asus\.vscode\extensions\ms-python.python-2022.20.1\pythonF
iles\lib\python\debugpy\adapter/../..\debugpy\launcher' '5
8884' '--' 'c:\Users\Asus\Desktop\CN-cycle2\TCPClient.py'
Enter the file name to be read : sample.txt
Hi , this is me!!
```

```
cher' '58879' '--' 'c:\Users\Asus\Desktop\CN-cycle2\TcpS
erver.py'
Server is listening localhost : 12345
Waiting for connection
Accepted Connection 127.0.0.1 : 58887
Waiting for connection
```

LAB - 10

Date __/__/__
Page _____

Using UDP sockets, write a client-server program to make client sending the file name & the server to send back the contents of the file.

Server Side

```
import socket as s

server_socket = s.socket (s.AF_INET , s.SOCK_DGRAM)
server_add = ('localhost' , 12345)
server_socket.bind( server_add)
server_socket.listen(1)
print (' server is listening {}:{}' .format (*server_add))

while True :
    data , client_add = server_socket.recvfrom (4096)
    fn = data.decode()
    with
    try :
        with open (fn , 'rb') as f :
            fc = f.read()
            server_socket.sendto (fc , client_add)
    except FileNotFoundError :
        server_socket.sendto (b'FileNotFound' , client_add)
```

Client Side

```
import socket as s

client_socket = s.socket (s.AF_INET , s.SOCK_DGRAM)
server_add = ('localhost', 12345)

filename = input ('Enter filename').encode()
client_socket.sendto (filename, server_add)

// Receive
    fc , _ = client_socket.recvfrom (4096)
    print ("File contents")
    print (fc.decode())
```

Output :

Server	Client
Server is listening	Enter file name: sample.txt File contents: Hi, Good Evening

```
Server is listening localhost : 12345
```

```
\python\debugpy\adapter/../..\debugpy\launcher' '59114'
'--' 'c:\Users\Asus\Desktop\CN-cycle2\UdpClient.py'
Enter the file name to be read : sample.txt
Hi , this is me!!
PS C:\Users\Asus\Desktop\CN-cycle2>
```