

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT
on
COMPUTER NETWORKS

Submitted by

NITHIN B S (1BM20CS100)

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
October-2022 to Feb-2023

**B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “**LAB COURSE COMPUTER NETWORKS**” carried out by **NITHIN BS (1BM20CS100)**, 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 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (20CS5PCCON)** work prescribed for the said degree.

Lohith JJ

Designation
Department of CSE
BMSCE, Bengaluru

,

Dr. Jyothi S Nayak

Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

Sl. No.	Date	Experiment Title	Page No.
1	10-11-2022	Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.	4-8
2	17-11-2022	Configuring IP address to Routers in Packet Tracer. Explore following messages: Ping Responses, Destination unreachable, Request timed out, Reply	9-15
3	24-11-2022	Configuring static route to the Router	16-22
4	01-12-2022	Configuring default route to the Router	23-30
5	08-12-2022	Configuring RIP Routing Protocol in Routers	31-37
6	15-12-2022	Configuring DHCP within a LAN in a packet Tracer	38-41
7	15-12-2022	Demonstration of WEB server and DNS using Packet Tracer	42-46
8	29-12-2022	Write a program for error detecting code using CRC-CCITT (16-bits).	47-50
9	05-01-2023	Write a program for congestion control using Leaky bucket algorithm.	51-53
10	12-01-2023	Write a program for distance vector algorithm to find suitable path for transmission.	54-56
11	12-01-2023	Implement Dijkstra's algorithm to compute the shortest path for a given topology.	57-59
12	28-01-2023	Using TCP/IP sockets, write a client-server program to make sending the file name and the server to send back the contents of the file if present.	60-64
13	29-01-2023	Using UP sockets, write a client-server program to make client the file name and the server to send back the contents of the requested file if present.	65-67

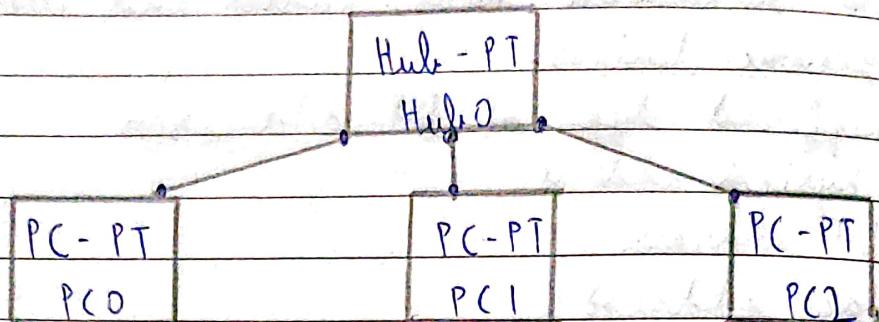
10/11/22

LAB - 1

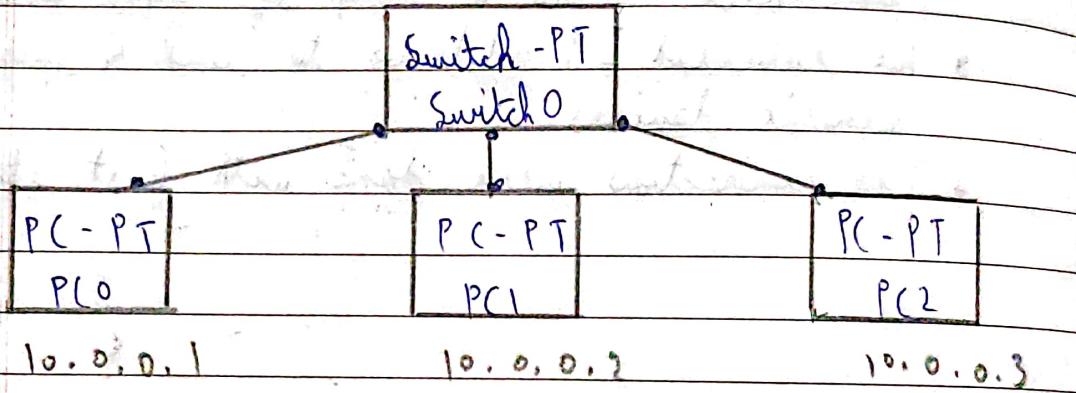
AIM: Creating a topology & simulate simple PDU from source to dest. using simple hub & switch

TOPOLOGY:

Using hub:

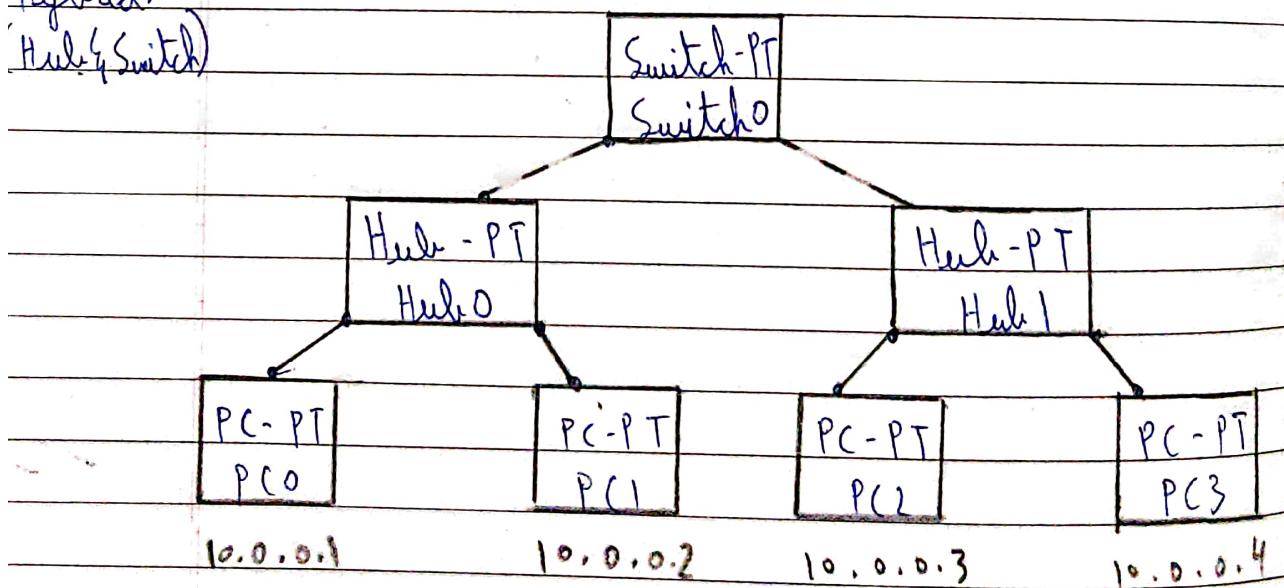


Using switch:



Hybrid:

(Hub & Switch)



Procedure:

Using Hub: Add generic hub & the PC's to workspace.

- Configure IP address of each PC in configuration tab.

- Connect all PC's to hub using copper straight wire.
- Hub & PC is connected to each other's fast ethernet connection.

Real time: Select source PC & in desktop tab, select CMD and type ping <IP-address> to generate response.

Simulation: Select simple PDU & select source & dest. computer.

time: clicking on auto capture, to see how ports are transferred to & from device.

Using Switch: Same similar steps to hub.

Real time: Same as hub.

Simulation: Same as hub connection, but not all transfer are successful.

Hybrid: We connect 2 hubs from a switch & generic devices to hubs, similar as above steps config. IP address

Real time: Similar steps as above.

Simulation:

Observation

→ Hub, * When source sends a packet in network the hub ~~sends~~ source the packet & ends broadcast over the network, it sends data to all the end devices in network & node where it matches with specified address accept the packet & acknowledge it. Remaining nodes ignore message.

- * Conn' b/w hub & end device is established through Copper straight through wire as they belong to different layers.

Switch \Rightarrow Similar as hub, but here it takes some time called learning time.

Result: 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 = 0 ms

Reply from 10.0.0.3: bytes = 32 time = 0 ms

Reply from 10.0.0.3: bytes = 32 time = 0 ms

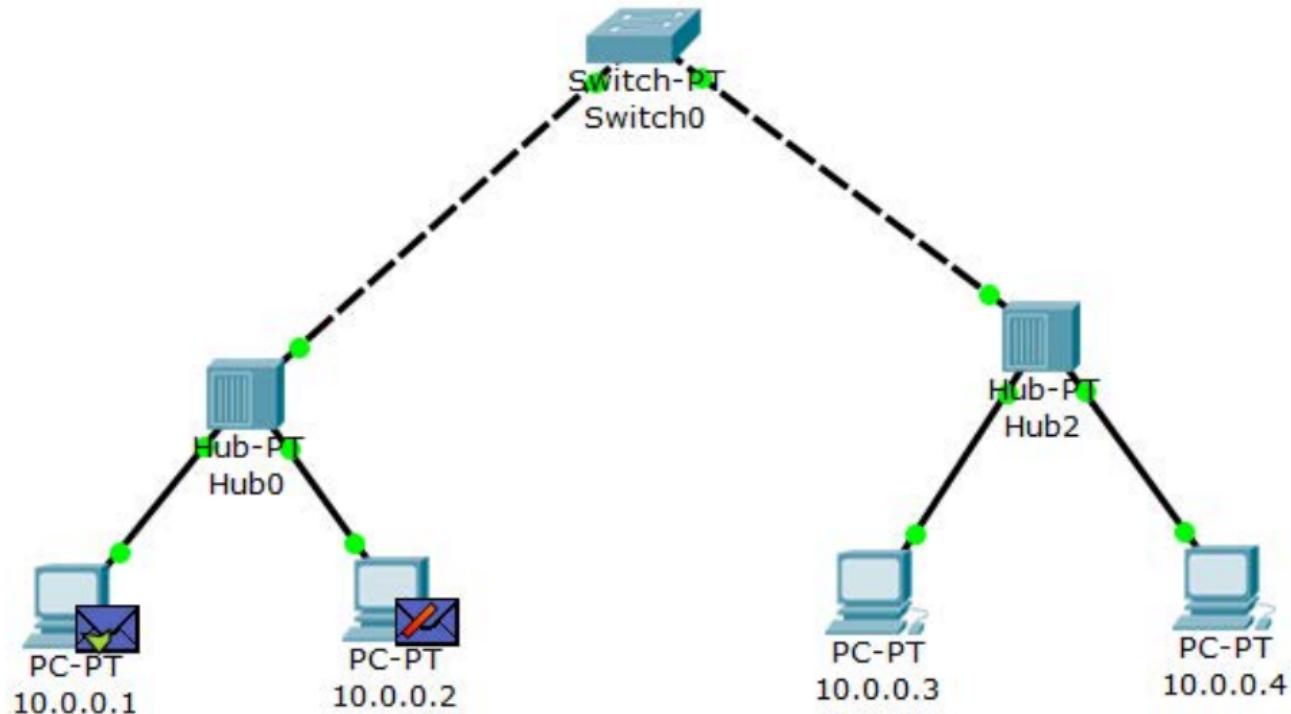
Reply from 10.0.0.3: bytes = 32 time = 0 ms

ping statistics for 10.0.0.3

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

Neelam
17/11/2022

We'll notice the packets sent ~~not~~ successfully without any interruption.



Command Prompt

X

```
Request timed out.  
Reply from 10.0.0.3: bytes=32 time=12ms TTL=128  
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128  
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
```

```
Ping statistics for 10.0.0.3:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 6ms, Maximum = 12ms, Average = 8ms
```

```
PC>ping 10.0.0.3
```

```
Pinging 10.0.0.3 with 32 bytes of data:
```

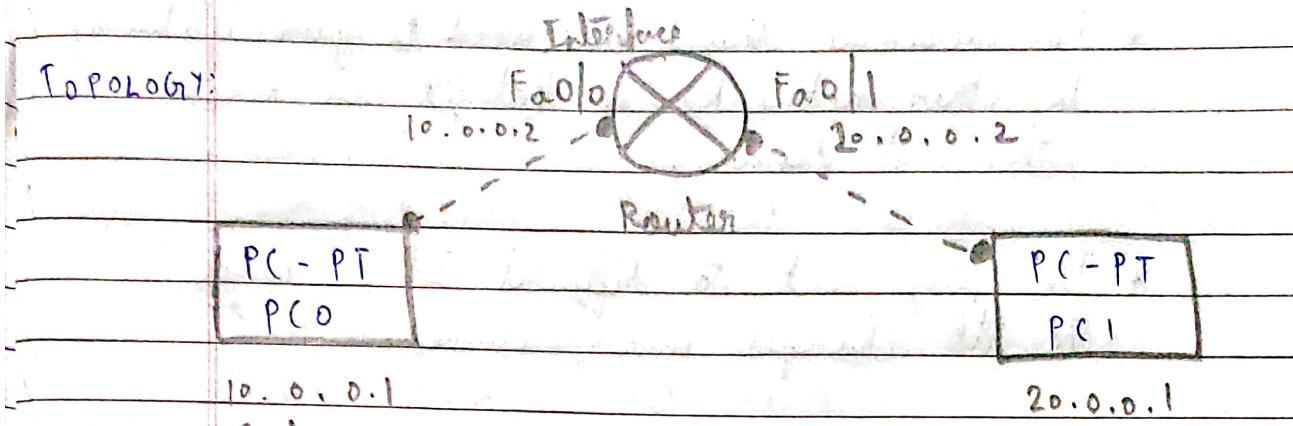
```
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
```

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

```
PC>
```

L AB - 2

AIM: Configuring IP address to routers in Packet tracer.
Explore: Ping responses, Destination unreachable, Reply, Request timed out.



PROCEDURE* A topology was created using 2 end devices with different IP address configuration.

- * Router was configured with respective interface and IP.
- * In end devices, we changed the gateway to different IP address.
- * In (MD), we will ping to other IP's connected by router.

OBSERVATION: Router (CLI Commands):

enable
config terminal
interface Fa0/0

ip address 10.0.0.2 255.0.0.0

no shutdown

exit

no shutdown

exit

no shutdown

OBSERVATION: First we'll notice request timed out, because of TTL (time to leave) was 0, and whenever passed after that will be discarded.

- * To overcome this, we need to give gateway to other networks so that it can access other end devices.
- * The ping sent to different IP's and the TTL changes were observed.
- * Only then the ping response will be successful.

RESULT: P(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

ping statistics for 10.0.0.2

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

* Successful reply on pinging from one end device to different end device, only after specifying gateway, through a router network.

- * Similarly, from P(1) we'll ping to 20.0.0.1 & 20.0.0.2, to get a successful reply.
- , And also from P(2) we'll ping to 10.0.0.1, 20.0.0.2, 10.0.0.2, to get a successful reply.
- * Without gateway, reply timed out occurs.

PC > ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes = 32

Request timed out.

Request timed out.

Request timed out.

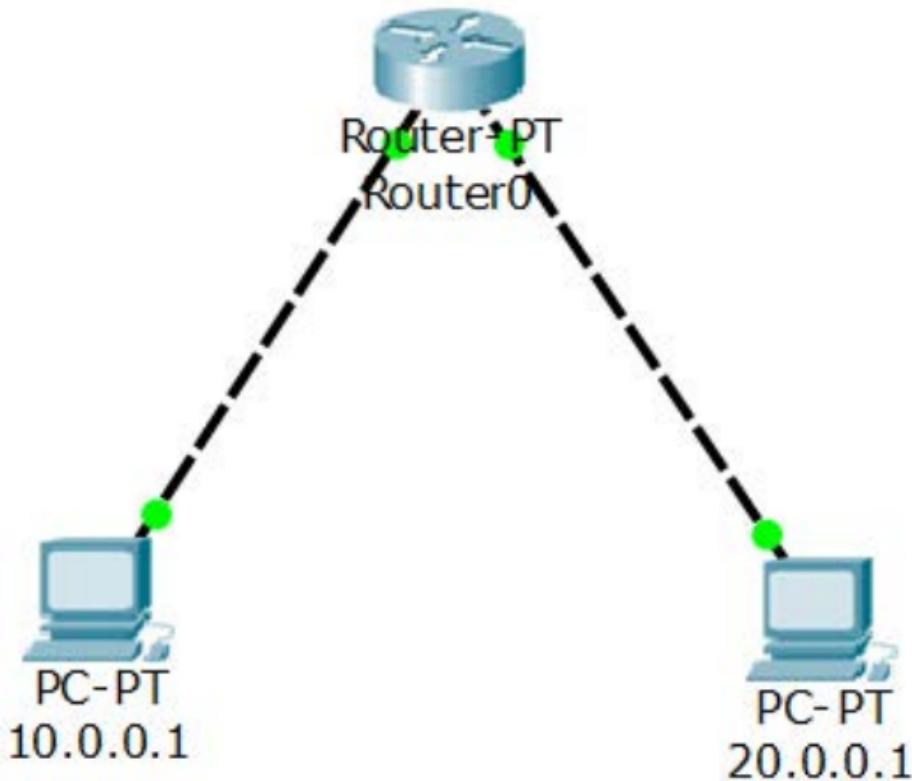
Request timed out.

ping statistics for 10.0.0.2

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

This is for request timed out scenario,
when gateway is not specified.

Neelima
17/11/2021



PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.

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:

Reply from 20.0.0.1: bytes=32 time=6ms TTL=127

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

Reply from 20.0.0.1: bytes=32 time=1ms 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 = 6ms, Average = 1ms

PC>

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.

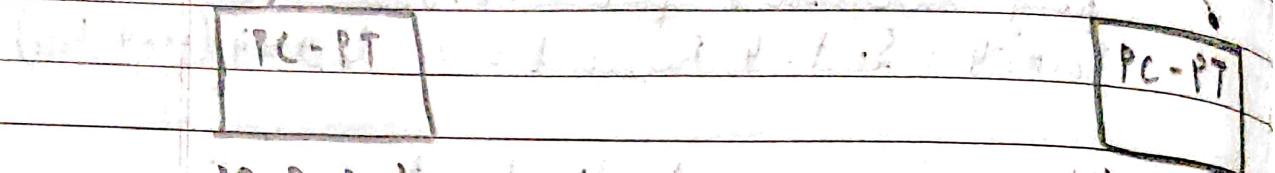
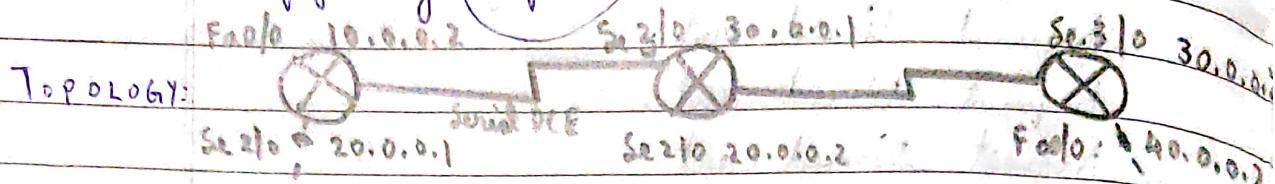
Ping statistics for 20.0.0.1:

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

27/11/22

LAB - 3

AIM: Configuring static ~~Default~~ route to the Router.



10.0.0.1 is default gateway and is 20.0.0.1

Gateway = 10.0.0.2 for PC and Gateway = 40.0.0.2

PROCEDURE: A topology was created with 3 generic routers and 2 generic end devices.

- * For the router connection we used Serial DCE and for end devices to router , automatic connection.
- * For Serial DCE we used Serial 2/0 & Serial 3/0
- * For automatic connection , we used Fastethernet.
- * Then we configure with different IP address.

OBSERVATION

- * Creating a multiple router topology by connecting routers to serial ports & PC-router through fastethernet.
- * Assigning gateway to routers as we did in the previous lab .
- * Pinging PC1 from PC0 at this state gives DESTINATION Host UNREACHABLE.
- * Viewing the routers ip routes using SHOW IP ROUTE

- * Adding static routes to the routers using
ip ~~address~~ route < dest. network > < subnet mask >
< next hop > in privileged (enable) configure
(configure terminal) mode.
- * Pinging PC1 from PC0 & PC0 from PC1
now works as expected.
- * As we did show ip route we'll see that
10.0.0.0 & 20.0.0.0 is directly connected
whereas the 30.0.0.0 & 40.0.0.0 were
not connected successfully.
- * So to avoid unreachable we use the alone
format to send successfully.

RESULT: In CLI, for 1st router,

show ip route

C 10.0.0.0/8 is directly connected, FastEthernet0/0

C 20.0.0.0/8 is directly connected, Serial 2/0

* After using...

ip ~~address~~ route 30.0.0.0 255.0.0.0 20.0.0.2

exit

ip ~~address~~ route 40.0.0.0 255.0.0.0 20.0.0.2

exit

o/p * C 10.0.0.0/8 is directly connected, FastEthernet0/0

C 20.0.0.0/8 is directly connected, Serial 2/0

S 30.0.0.0/8 [1/0] via 20.0.0.2

S 40.0.0.0/8 [1/0] via 20.0.0.2

Before PC > ping 40.0.0.1

Gateway Ping 40.0.0.1 with 32 bytes of data:

Request timed out.

Request timed out.

Request timed out.

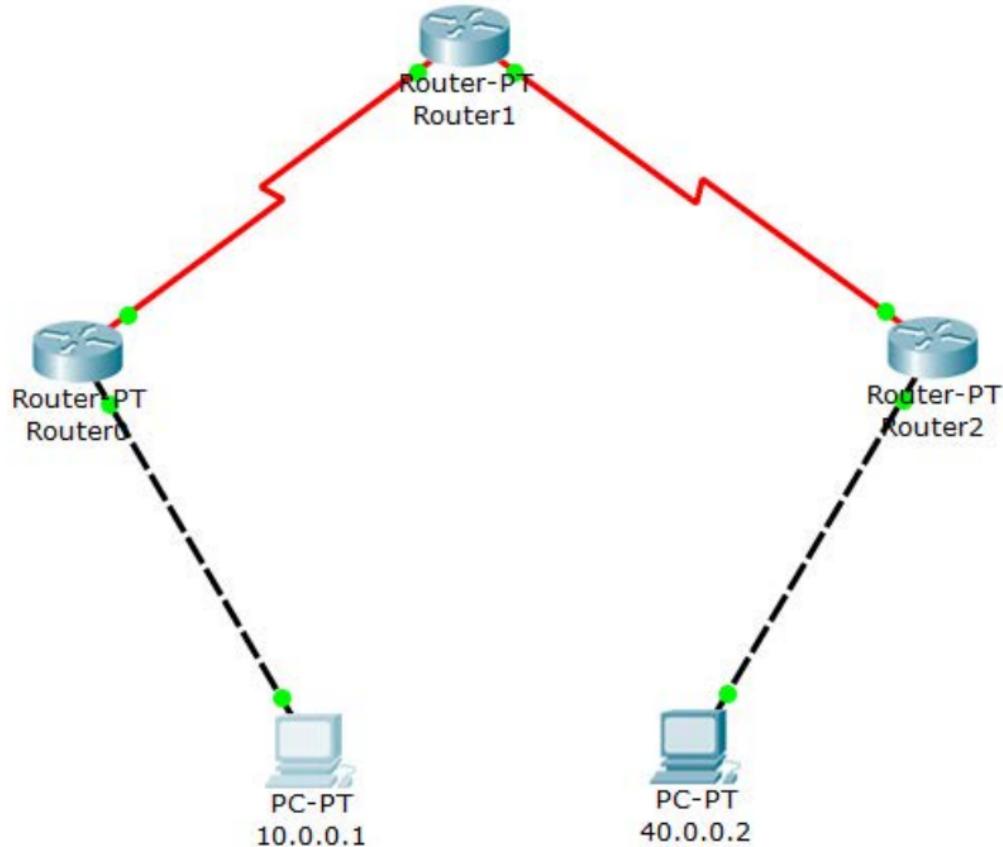
Packet sent = 4, Received = 0 (100.0% Loss)

After gateway Ping 40.0.0.1
Reply from 190.0.0.2 : Destination host unreachable
Reply from 190.0.0.2 : Destination host unreachable
Packets: Sent = 4, Received = 0 (100% loss)

→ After ip route config to all 3 routers.

Ping 40.0.0.1
Reply from 40.0.0.1 : bytes = 32, time = 20 ms TTL = 125
Reply from 40.0.0.1 : bytes = 32, time = 30 ms TTL = 125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 3 (25% loss)

Nexting
10/1/2022



PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.

Ping statistics for 40.0.0.2:

 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>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Reply from 40.0.0.2: bytes=32 time=4ms TTL=125

Ping statistics for 40.0.0.2:

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

Approximate round trip times in milli-seconds:

Minimum = 2ms, Maximum = 4ms, Average = 2ms

PC>

1/12/22

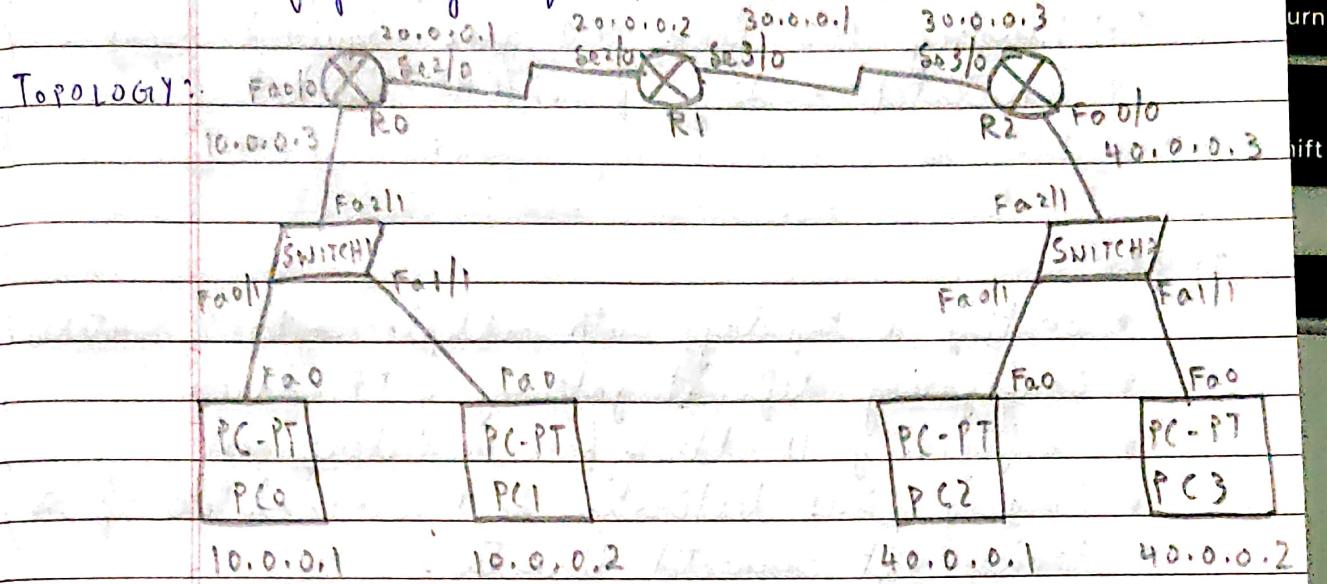
King

Page 10
of 10

Next

LAB - 4

AIM: Configuring default route to the router



PROCEDURE: A topology was created using 3 router-PT, 2 Switch PT and 2 PC's connected to each switch using Copper straight through connections & serial DCE connection.

- * Default gateways & unique IP addresses were configured for each PC.
- * IP addresses were configured for each interface using CLI
- * Pinging PC2 from PC0 gave destination host unreachable.

OBSERVATION

- * Pinging PC2 from PC0 gave destination host unreachable.
- * If routes for each router was viewed using :
show ip route.
- * Static ip route was configured for R1 using CLI commands: ip route dest-network subnet mask next-hop-address.

- * Default ip route was configured for R0 & R2 using CLI commands:
- ```
ip 0.0.0.0 0.0.0.0 next-hop-address
```
- \* Pinging PC2 from PC0 gave required reply.
  - \* Simulated sending of an ICMP packet from PC0 to PC2.

#### OUTCOMES:

- \* Creating a topology with multiple routers & switches.
- \* Configuring default gateway & IP address.
- \* Configuring IP address for the interfaces.
- \* Pinging gives destination host unreachable if the devices not connected directly.
- \* Configuring default IP routes to a router ensures that the packet passes through the default route when no other route is available.
- \* On configuring default IP routes, gives required reply.
- \* The simulation of sending a simple PDU from source to destination shows the route taken by the ICMP packet.

RESULT: Before setting default router path

> ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data  
Request timed out.

Request timed out.

"

"

> ping 30.0.0.1 & all other interfaces after that to end devices of other network.

Pinging 30.0.0.1 with 32 bytes of data

Reply from 10.0.0.3: Destination host unreachable

"

"

"

Ping statistics

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

> Ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data.

Reply from 10.0.0.2: bytes = 32 time = 17 ms TTL = 125

"

time = 3 ms "

"

time = 2 ms "

"

time = 2 ms "

Ping statistics for 10.0.0.2

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

Approx. round trip time in (ms)

Minimum = 2 ms, Maximum = 17 ms

Average = 6 ms

→ After setting default routes path for all routers -

> Ping 30.0.0.1 & 30.0.0.2

Pinging 30.0.0.1 with 32 bytes of data.

Request timed out

Reply from 30.0.0.1: bytes = 32 time = 2 ms TTL = 253

Request timed out

Reply from 30.0.0.1: bytes = 32 time = 12 ms TTL = 253

→ ~~Ping statistics~~

> Ping 40.0.0.1 & other interfaces

Pinging 40.0.0.1

Reply from 40.0.0.1: bytes = 32 time = 16 ms TTL = 253

"

time = 12 ms

"

time = 2 ms

"

time = 9 ms

Ping statistics for 40.0.0.1

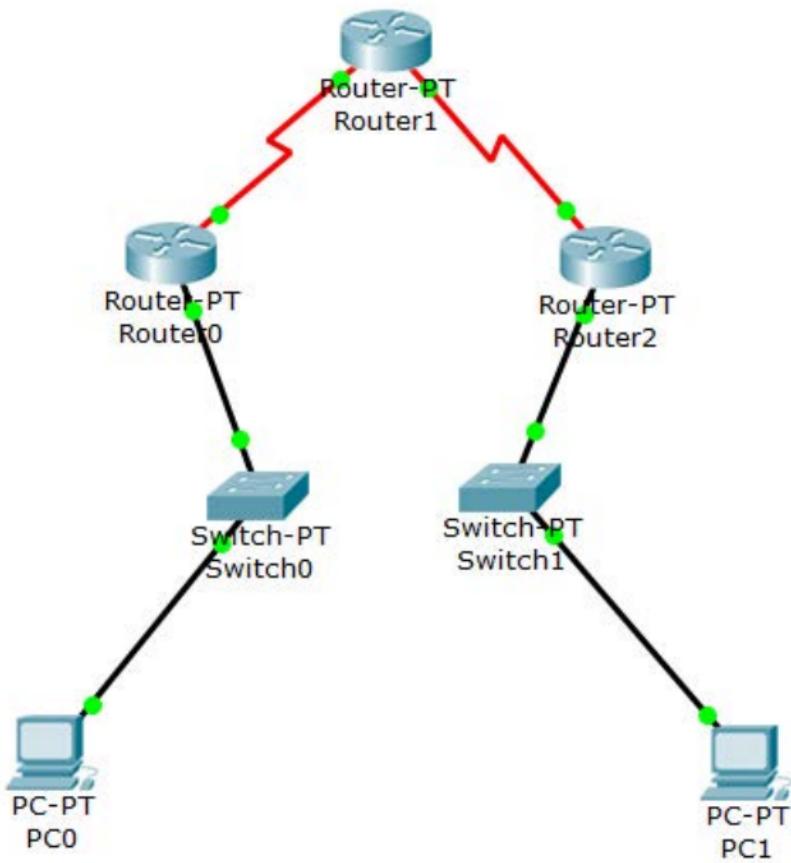
Packets Sent = 4, Received = 4, Lost = 0% (0' lost)

Aprox round trip, (ms)

Min = 2 ms, Max = 16 ms

Average = 9 ms

Deviation  
112/20%



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=14ms TTL=125

Reply from 40.0.0.2: bytes=32 time=11ms TTL=125

Reply from 40.0.0.2: bytes=32 time=11ms TTL=125

Reply from 40.0.0.2: bytes=32 time=11ms TTL=125

Ping statistics for 40.0.0.2:

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

Approximate round trip times in milli-seconds:

Minimum = 11ms, Maximum = 14ms, Average = 11ms

PC>

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>

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 40.0.0.1: Destination host unreachable.

Request timed out.

Reply from 40.0.0.1: Destination host unreachable.

Reply from 40.0.0.1: Destination host unreachable.

Ping statistics for 10.0.0.1:

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

PC>

8/12/22

King

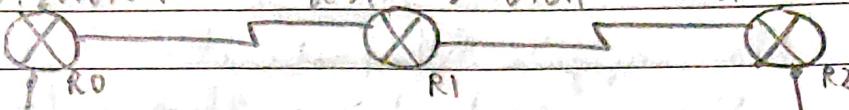
DATE \_\_\_\_\_  
PAGE NO. \_\_\_\_\_

## LAB - 5

Aim: Configuring RIP Routing Protocol in Routers

Fa0/0: 10.0.0.3    Se2/0: 20.0.0.2    Fa0/0: 40.0.0.3

Topology: Se2/0: 20.0.0.1    Se3/0: 30.0.0.1    Se3/0: 30.0.0.3



SWITCH

SWITCH

PC-PT

10.0.0.1

PC-PT

10.0.0.2

PC-PT

40.0.0.1

PC-PT

40.0.0.2

Gateway: 10.0.0.3

10.0.0.3

40.0.0.3

40.0.0.3

PROCEDURE: A topology was created with 3 routers - PT,  
2 switch - PT & 2 PC-PT's connected to each  
switches using Copper straight through connection &  
serial DCE connections.

\* First we configured end devices giving IP  
address 10.1, 10.2 & 40.1, 40.2.

\* Then we configured routers using ip address  
<address> <subnet mask>.

\* encapsulation ppp and clock rate 64000  
used to specify ppp protocol and clock rate  
in routers 0 and 1 for the serial ports only.

\* Configured RIP routing using router rip  
then, network <address>, where address  
refers to the networks directly connected to  
the router.

\* We use default configurations before encapsulation  
~~ip route 0.0.0.0 0.0.0.0 20.0.0.2~~

OBSERVATION PPP → point to point protocol for serial clock connection.

encapsulation PPP for serial interfaces only  
clock rate only for clock interfaces.

To configure RIP protocol  
click on router 1 → enable → config terminal

```
> router rip network 10.0.0.0
> network 20.0.0.0
> network 30.0.0.0 exit
```

Router 2

```
> router rip
> network 20.0.0.0
> network 30.0.0.0
> exit
```

RIP is configured  
in all the routers.

Router 3

```
> router rip net 40.0.0
> network 30.0.0.0
> exit
```

To check if router is collecting router info  
> show ip route.

To config S/L  
Fast Ethernet 0

For every F.DCE connection to configure RIP with defined clock rate.

```
> encapsulation PPP
> clock rate 64000
```

RESULT: ping 40.0.0.1 TTL

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

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

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

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

Packet sent = 4, Received = 4 (0% loss)

Approx. around ip in milliseconds.

Minimum = 2 ms Maximum = 40 ms

Average = 21 ms

Since RIP has been established, IP route does not have to be set for each router.

Before RIP has been set:

ping 10.0.0.1 → 40.0.0.1 : Dest. host unreachable.

Before RIP

ping 10.0.0.1 → 20.0.0.2 : Request timed out.

~~RIP~~ Only on analysing correct gateway and protocol is the reply received properly.

∴ RIP is established in the network correctly.

~~RIP : Routing Information Protocol~~

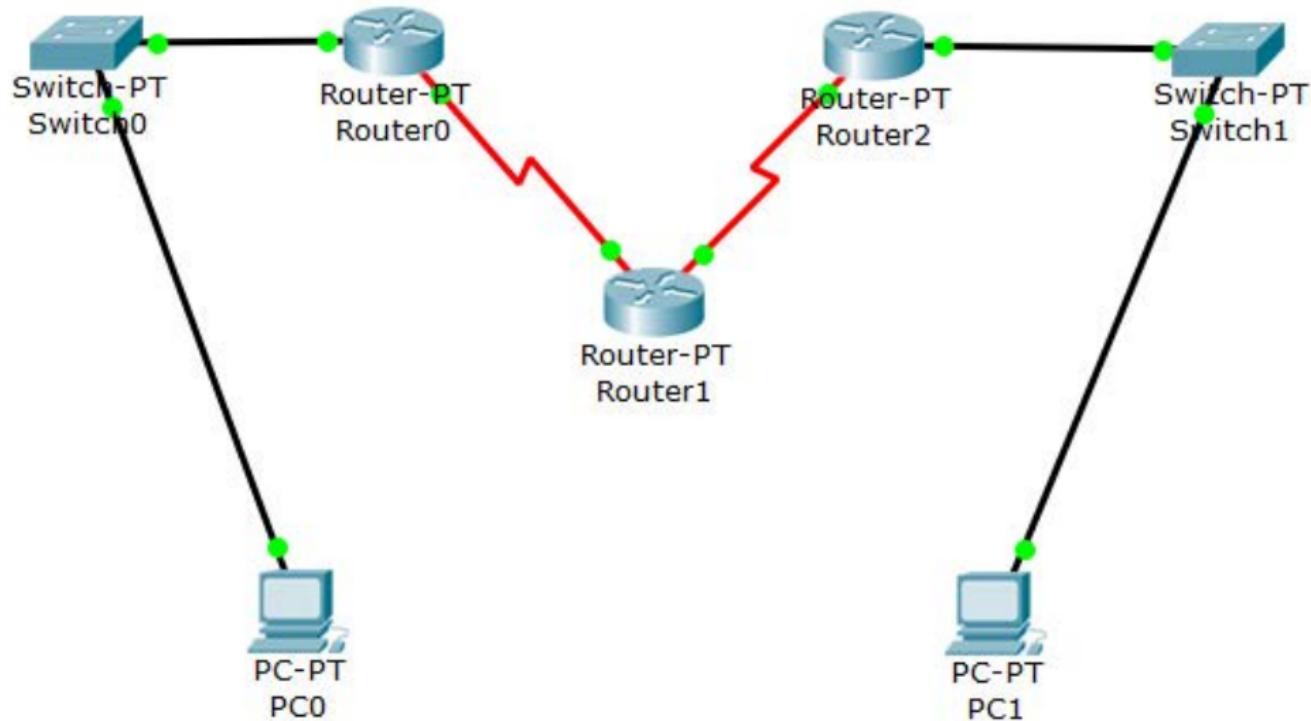
We use RIP to configure routers.

It is a distance vector routing protocol.

It knows only neighbours and it doesn't know

entire topology routing. It is a dynamic routing

protocol that uses hop count as a metric to find the best path b/w source & destination network.



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=2ms TTL=125

Reply from 40.0.0.2: bytes=32 time=12ms TTL=125

Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Reply from 40.0.0.2: bytes=32 time=12ms TTL=125

Ping statistics for 40.0.0.2:

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

Approximate round trip times in milli-seconds:

Minimum = 2ms, Maximum = 12ms, Average = 7ms

PC>ping 20.0.0.2

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),

PC>ping 40.0.0.2

PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: Destination host unreachable.

Reply from 10.0.0.2: Destination host unreachable.

Reply from 10.0.0.2: Destination host unreachable.

Request timed out.

Ping statistics for 40.0.0.2:

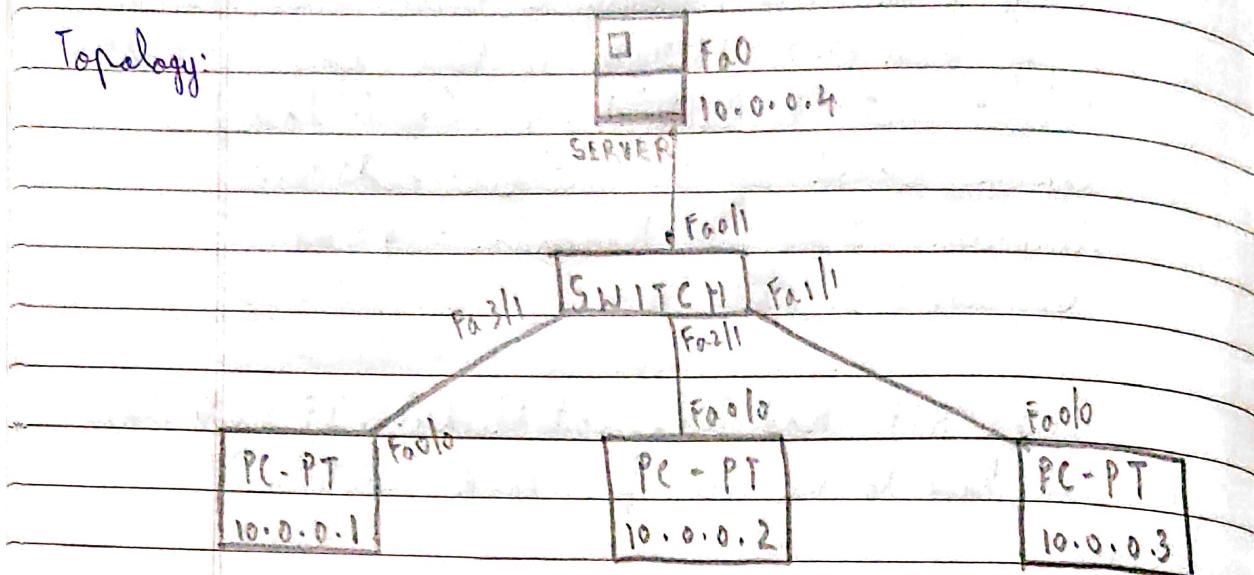
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

15/12/22

## LAB - 6

### Aim: Configure DHCP Server

Topology:



PROCEDURE: Construct the following topology i.e server is connected to a switch, and further is connected to PC's (end devices)

- \* Configuring IP address of the server as 10.0.0.4
- \* Add same IP address if the end devices in, Start IP address of DHCP, After switching on the DHCP server.
- \* After saving addresses, change the IP address of the end devices from static to DHCP.
- \* Ping the IP addresses across all the end device.

### OBSERVATION

- \* The messages pinged from an end device to the server, or from an end device to another end device is successfully sent.
- \* A dynamic IP address is set to end devices when the DHCP is initiated in the server.
- \* RARP is used to assign the IP address of a

device if MAC address is known.

Q1 P: i) PC > ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes = 32 time = 1ms TTL = 128

Reply from 10.0.0.3: bytes = 32 time = 0ms TTL = 128

Ping statistics for 10.0.0.3

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

Approximate round trip time in milliseconds.

Minimum = 0ms, Maximum = 4ms, Average = 1ms

2) SERVER > 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

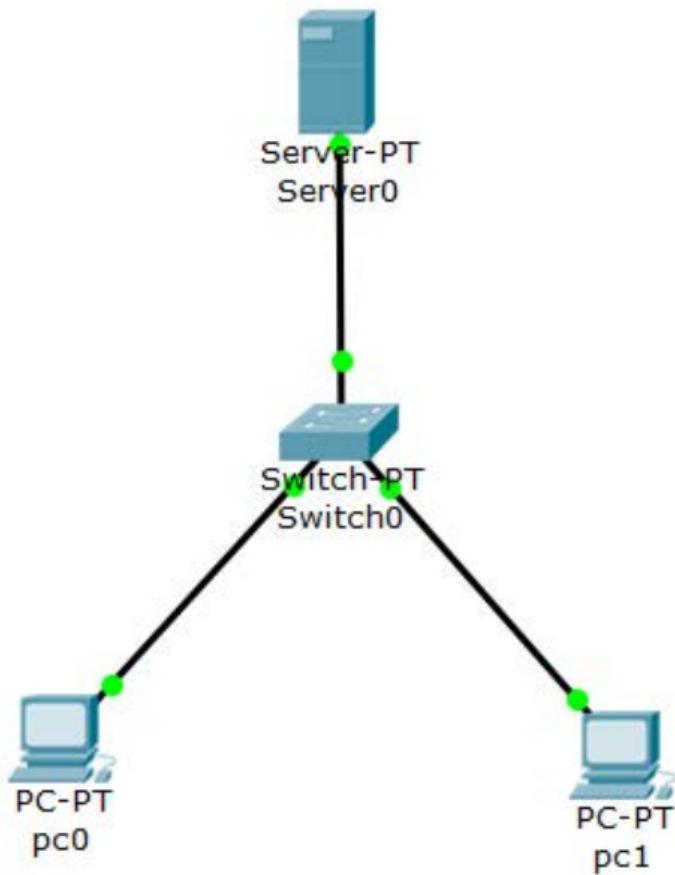
Reply from 10.0.0.1: bytes = 32 time = 0ms TTL = 128

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

Ping statistics:

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

15/12/2022





Physical

Config

Desktop

Custom Interface

## Command Prompt

X

Packet Tracer PC Command Line 1.0

PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=1ms TTL=128

Reply from 10.0.0.3: bytes=32 time=0ms TTL=128

Reply from 10.0.0.3: bytes=32 time=3ms 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 = 3ms, Average = 1ms

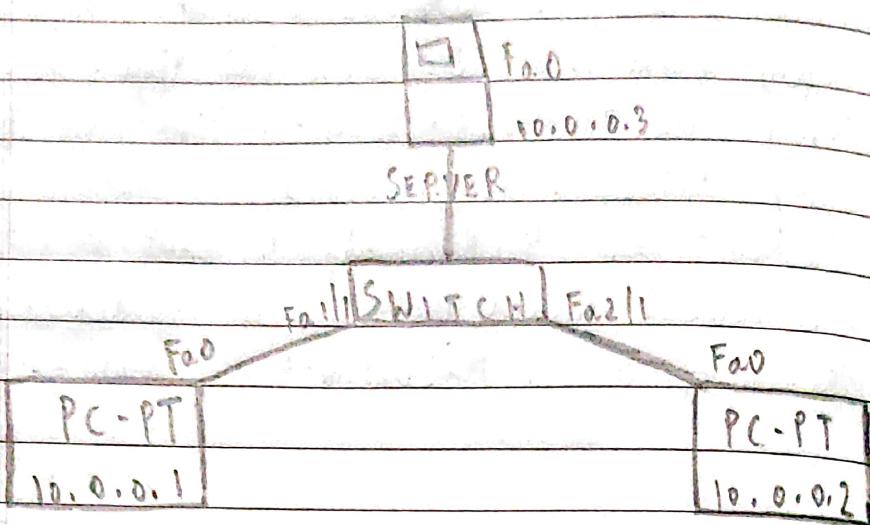
PC>

15/12/22

## LAB - 7

### AIM: Configuring webserver & DNS server

#### TOPOLOGY:



#### PROCEDURE:

- \* Construct the following topology, where server is connected to end devices through switches.
- \* Configure IP address of the server & end devices.
- \* Set the HTTP and DNS server to (on) state.
- \* Set the server domain name & address of it as same as server and save the following.
- \* From one of the end devices, check if the webserver is reachable from the end devices by entering URL.

#### For DNS:

- \* Click on DNS service on 'services'.
- \* Set the name & IP addresses & click add button.
- \* Somesror is checked.  
Name set → cse lmsce

IP address  $\rightarrow$  10.0.0.3

in URL enter

URL  $\Rightarrow$  http://cse.lumsce

It displays the index.html page.

- $\rightarrow$  A HTML page is created displaying one's short CV.
- $\rightarrow$  The page is connected to the original (alone) page.
- $\rightarrow$  The same modified page is seen in the web browser.

#### OBSERVATION

- \* We can view the webpage, when we type 'cse.lumsce' in browser, because 10.0.0.3 address is linked to the name 'cse.lumsce' according to DNS.
- \* Since it's difficult for users to remember IP addresses, linking is required i.e., If it is linked with a domain name.
- \* If the system / server hasn't been configured default gateway the Host Unresolved is shown.
- \* If configured properly the page of cisco packet tracer is opened.

~~Absent~~

6/1/2023

RESULT: Web Browser

< > URL `http://csebmse`

BMSCE Packet Tracer

Welcome to BMSCE packet tracer

Quick links

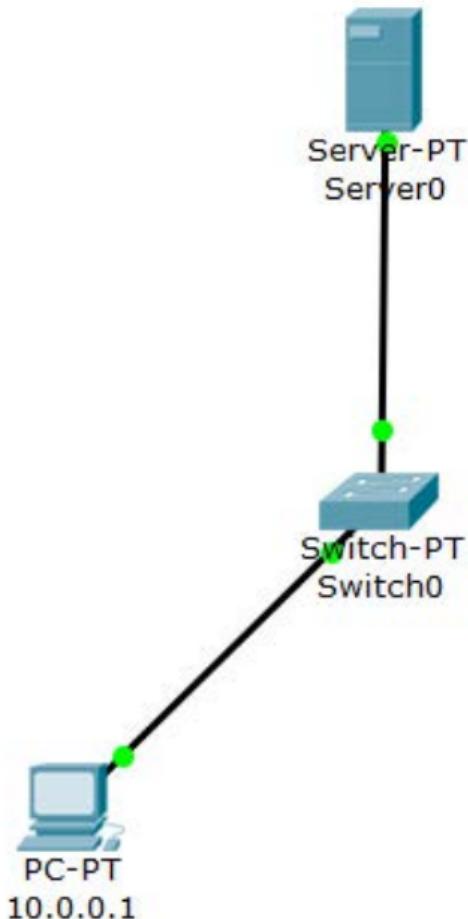
my CV

`http://csebmse/myCV.html`

|         |            |
|---------|------------|
| NAME    | NITHIN     |
| AGE     | 20         |
| DOB     | 27/05/2002 |
| COLLEGE | BMSCE      |
| BRANCH  | CSE        |
| YEAR    | 3RD        |

69665

010100101100110010101





Physical

Config

Desktop

Custom Interface

## Web Browser

X



URL

http://csebmsce

Go

Stop

# BMSCE

Welcome to BMSCE. Opening doors to new opportunities. Mind Wide Open.

Quick Links:

[A small page](#)

[Copyrights](#)

[Image page](#)

[Image](#)

29/12/22

LAB - 8

DATE \_\_\_\_\_  
PAGE NO. \_\_\_\_\_

Write a program for error detection using  
(CRC - 16 bit)

Data : 1011101

$G(x) : 10001000000100001$

Modified Data : Data + 16 zero's

Codeword = Data + Checksum.

```
include < stdio.h >
include < stdlib.h >
include < string.h >
char g[] = "10001000000100001", p[50], c[50];
int m1, m2;
→ void xor() { int i;
 for(i=0; i<_n1; i++) {
 if (c[i] == g[i]) {
 c[i] = '0';
 } else {
 c[i] = '1';
 }
 }
}
```

~~```
→ void crc() { int i, count;
    for (count = 0; count < n1; count++) {
        c[
```~~

```
→ void ore(int u) { int i, j;
    for(i=0; i< u; i++)
        temp[j] = m[j];
    for (i=0; i< 16; i++)
        r[j] = m[i];
    for (j=0; j< u-16; j++) {
        if (r[0] == '1') {
```

```

g[i] = '1';
caltran(); }

else { g[i] = '0';
shiftl();
r[16] = m[17+i];
r[17] = '10';

for (j = 0; j <= 17; j++)
temp[j] = r[j]; }

g[u-16] = '10'; }

```

```

void caltran() { int i, j;
for (i = 1; i <= 16; i++)
r[i-1] = ((int) temp[i] - 48) ^ ((int) g[i] - 48) + 48;

void shiftl() { int i;
for (i = 1; i <= 16; i++)
r[i-1] = r[i]; }

```

```

void caltrans(int n) {
int i, K = 0;
for (i = n-16; i < n; i++)
u[i] = ((int) n[i] - 48) ^ ((int) r[K++]-48) + 48;
m[i] = '10'; }

```

O/P : Enter data = 1011101
 $g(x) = 10001000000100001$
Modified data = 101110100000000000000000

Transmitted data : 101110110001010101000

Enter transmitted data : 101110110001010101000

CRC check

Reminder : 0000000000000000

Received data is correct.

Enter data to be transmitted: 1011010101

Enter the Generating polynomial: 1010

Data padded with n-1 zeros : 1011010101000

CRC or Check value is : 000

Final data to be sent : 1011010101000

Enter the received data: 1011010101000

Data received: 1011010101000

No error detected

Enter data to be transmitted: 1011010101

Enter the Generating polynomial: 1010

Data padded with n-1 zeros : 1011010101000

CRC or Check value is : 000

Final data to be sent : 1011010101000Enter the received data: 1011010101001

Data received: 1011010101001

Error detected

9/1/23

L&B - 9

King
DATE _____
PAGE NO. _____

Write a program for Leaky bucket algorithm.

```

import java.util.Scanner;
class main {
    public static void main (String args []) {
        int size = 0;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter capacity of the buffer");
        int max = sc.nextInt();
        System.out.println ("Enter output rate");
        int out = sc.nextInt();
        size += in;
        if (size > max) {
            System.out.println ("Input rate is greater than buffer size");
            size -= in;
        } else {
            System.out.println ("Buffer size: " + size);
            size -= out;
            size = Math.max (size, 0);
            System.out.println ("Buffer size after outflow: " + size);
        }
        System.out.println ("Continue (1 = Yes, 2 = No)");
        int t = sc.nextInt();
        if (t == 2)
            break;
    }
}

```

OIP : Enter capacity = 500

Output rate = 40

Input rate = 100

Buffer size : 100

Set after outflow = 60

Continue? (1 = Yes, 2 = No) : 2

ND
5/1/2023

~~LastBucket~~_BucketCapacity

Enter the bucket capacity :

500

Enter output rate

200

Enter the input rate :

300

Bucket Capacity is 100

Do you want to continue, 2 to exit ,1 to continue

1

Enter the input rate :

300

Bucket Capacity is 200

Do you want to continue, 2 to exit ,1 to continue

1

Enter the input rate :

400

Bucket Overflow

Do you want to continue, 2 to exit ,1 to continue

1

12/1/23

King

LAB - 10

Bellman Ford

```
# include <iostream>
#define MAX 10
using namespace std;
int src, dest, wt;
}
void bell (int nv, edge e[], int src_graph, int ne) {
    int u, v, weight, i, j = 0;
    int dis[MAX];
    for (i = 0; i < nv; i++) {
        dis[i] = 999;
    }
    dis[src_graph] = 0;
    for (i = 0; i < nv - 1; i++) {
        for (j = 0; j < ne; j++) {
            u = e[j].src;
            v = e[j].dest;
            weight = e[j].wt;
            if (dis[u] == 999 && dis[u] + weight < dis[v]) {
                dis[v] = dis[u] + weight;
            }
        }
    }
    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 << "Neg cycle present";
        }
    }
}
```

```
cout << "Vertex" << "Dist from source";
for (i=1; i<=nv; i++) {
    cout << "\n" << i << " " << dis[i];
}
```

Q1:- Enter no. of vertices : 5

Enter adj matrix :

| | | | | |
|---|---|---|---|---|
| 0 | 6 | 5 | 0 | 0 |
| 6 | 0 | 0 | 7 | 0 |
| 5 | 0 | 0 | 4 | 3 |
| 0 | 7 | 4 | 0 | 2 |
| 0 | 0 | 3 | 2 | 0 |

Enter starting node : 1

Dist of 0 = 6

Path = 0 <-> 1

Count = 2

Dist of 2 = 11

Path = 2 <-> 0 <-> 1

Count = 3

Dist of 3 = 7

Path = 3 <-> 1

Count = 2

Dist of 4 = 9

Path = 4 <-> 3 <-> 1

Count = 3

NP
P1/P2/P3

Enter no. of vertices: 4

Enter graph in matrix form:

0 5 17 3

2 0 3 5

8 5 0 2

1 3 2 0

Enter source: 1

Vertex 1 -> cost = 0 parent = 0

Vertex 2 -> cost = 5 parent = 1

Vertex 3 -> cost = 5 parent = 4

Vertex 4 -> cost = 3 parent = 1

Dijkstra's

```
from collections import defaultdict
```

```
class Graph():
    def __init__(self):
```

```
        self.edges = defaultdict(list)
```

```
        self.weights = {}
```

```
    def addEdge(self, fromnode, tonode, wt):
```

```
        self.edges[fromnode].append(tonode)
```

```
        self.edges[tonode].append(fromnode)
```

```
        self.weights[(fromnode, tonode)] = wt
```

```
        self.weights[(tonode, fromnode)] = wt
```

```
    def dijkstra(self, graph, initial, end):
```

```
        shortest_path = {initial: (None, 0)}
```

```
        curr_node = initial
```

```
        vis = set()
```

```
        while curr_node != end:
```

```
            vis.add(curr_node)
```

```
            dest = graph.edges[curr_node]
```

```
            wt_to_curr = shortest_path[curr_node][1]
```

```
            for next_node in dest:
```

```
                wt = graph.weights[(curr_node, next_node)] + wt_to_curr
```

```
                if next_node not in shortestpath:
```

```
                    shortestpath[next_node] = (curr_node, wt)
```

```
                else:
```

```
                    curr_short_wt = shortestpath[next_node][1]
```

```
                    if curr_short_wt > wt:
```

```
                        shortestpath[next_node] = (curr_node, wt)
```

`nxt_dest = {node : shortestpath[node] for node in
shortestpath if node not in visited}`

`if not nxt_dest:`

`return "Route not possible"`

`curnode = min(nxt_dest, key = lambda K: nxt_dest[K][1])`

`path = []`

`while curnode is not None:`

`path.append(curnode)`

`nxt_node = shortestpath[curnode][0]`

`curnode = nxt_node`

~~def~~ # Test case

`g = Graph()`

`g.addEdge('a', 'b', 4)`

`g.addEdge('a', 'c', 2)`

`g.addEdge('b', 'c', 1)`

`g.addEdge('b', 'd', 5)`

`g.addEdge('c', 'd', 8)`

`g.addEdge('c', 'e', 10)`

`g.addEdge('d', 'e', 2)`

`g.addEdge('d', 'z', 6)`

`g.addEdge('e', 'z', 9)`

`dijkstra(g, 'a', 'z')`

o/p: Shortest weight = 14

`['a', 'c', 'b', 'd', 'z']`

NP
12/11/2023

IDLE Shell 3.11.1

```
Python 3.11.1 (v3.11.1:a7a450f84a, Dec  6 2022, 15:24:06) [Clang 13.0.0 (clang-1  
300.0.29.30)] on darwin  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: /Users/nithinbs/Downloads/1BM18CS063_CN-master/Cycle 2 Lab/Lab - 3/Dj  
isktras.py  
Shortest Weigth: 14  
['a', 'c', 'b', 'd', 'z']  
>>> |
```

LAB - 12

TCP | IP

A.I.R: Using TCP | IP sockets , write client server program to make client sending the filename & server to send back contents of requested file if present.

(Client.py :-

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientsock = socket (AF_INET, SOCK_STREAM)
clientsock.connect ((serverName, serverPort))
msg = input (" \n Enter filename: ")
clientsock.send (msg.encode ())
filecont = clientsock.recv(1024).decode ()
print (filecont)
clientsock.close ()
```

Server.py :-

```
from socket import *
sname = '127.0.0.1'
sport = 12000
sock = socket (AF_INET, SOCK_STREAM)
sock.bind ((sname, sport))
sock.listen (1)
while 1:
    print (" Ready to receive")
    (sock, addr) = sock.accept()
    msg = sock.recv(1024).decode ()
    file = open (msg, "r")
    l = file.read (1024)
    file.close()
    sock.send (l.encode ())
    print (' \n Sent content of ' + msg)
    file.close()
    sock.close()
```

O/P for TCP/IP:

ServerTCP

The server is ready to receive

Sent contents of ServerTCP.py

Ready to receive

ClientTCP

Enter file name: ServerTCP.py

From Server:

```
from socket import *
```

```
serverName = "127.0.0.1"
```

```
sport = 12000
```

```
socket = socket (AF_INET, SOCK_STREAM)
```

```
socket.bind ((serverName, sport))
```

```
socket.listen(1) . . .
```

```
>>>
```

```
File Edit Format Run Options Window Help  
from socket import *  
serverName = '127.0.0.1'  
serverPort = 12000  
  
2  
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()
```

```
File Edit Shell Debug Options Window Help  
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]  
on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
=>>>  
===== RESTART: C:/Users/mdsur/Desktop/client.py =====  
  
Enter file name: server.py  
  
From Server:  
  
from socket import *  
serverName="127.0.0.1"  
serverPort = 12000  
serverSocket = socket(AF_INET,SOCK_STREAM)  
serverSocket.bind((serverName,serverPort))  
serverSocket.listen(1)  
while 1:  
    print ("The server is ready to receive")  
    connectionSocket, addr = serverSocket.accept()  
    sentence = connectionSocket.recv(1024).decode()  
    file=open(sentence,"r")  
    l=file.read(1024)  
    connectionSocket.send(l.encode())  
    print ('\nSent contents of ' + sentence)  
    file.close()  
    connectionSocket.close()
```

>>>

File Edit Format Run Options Window Help

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

File Edit Shell Debug Options Window Help

```
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec  6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/mdsur/Desktop/server.py =====
The server is ready to receive
```

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

File Edit Format Run Options Window Help

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

File Edit Shell Debug Options Window Help

```
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec  6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/mdsur/Desktop/server.py =====
The server is ready to receive
```

Ln: 1 Col: 1

Ln: 5 Col: 0



28°C

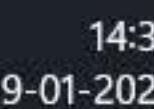
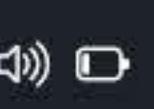
Sunny



Search



ENG
IN



14:37
29-01-2023

UPP

A14: Using UDP sockets, write client server program to make client sending the file name & the server to send back the contents of requested file if present.

Client.py

```
from socket import *
sname = '127.0.0.1'; sport = 12000
csock = socket(AF_INET, SOCK_DGRAM);
msg = input("\n Enter filename");
csock.sendto (bytes(msg, "utf-8"), (sname, sport))
content, addr = csock.recvfrom(2048)
print(content.decode("utf-8"))
csock.close()
```

Server.py

```
from socket import *
sname = '127.0.0.1'; sport = 12000
ssock = socket(AF_INET, SOCK_DGRAM);
ssock.bind(("127.0.0.1", sport))
print("Ready to listen");
while 1:
    msg, add = ssock.recvfrom(2048)
    msg = msg.decode("utf-8")
    file = open(msg, "r")
    l = file.read(2048)
    ssock.sendto (bytes(l, "utf-8"), add)
    print(msg)
    file.close()
```

OP for UDP:

Server-UDP

The server is ready to receive

Send contents of Server-UDP.py

Ready to receive.

Client-UDP

Enter file name: Server-UDP.py

Reply from Server:

from socket import *

sport = 12000

socket = socket (AF_INET, SOCK_DGRAM)

.. file.close()

>>>

```
C:\Windows\System32\cmd.e X + C:\Windows\System32\cmd.e X + - □ X
C:\Users\mdsur\Desktop\UDP>python -u serverUDP.py
The server is ready to receive
Sent contents of  serverUDP.py
|
```

```
C:\Users\mdsur\Desktop\UDP>python -u 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")
    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()

C:\Users\mdsur\Desktop\UDP>
```

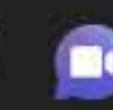


28°C

Sunny

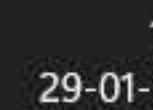
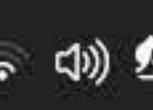


Search



ENG

IN



14:55

29-01-2023 1