|  |  |
| --- | --- |
| EXP NO : 1 | **NETWORK COMMAND - STUDY** |
| DATE : |

# AIM:

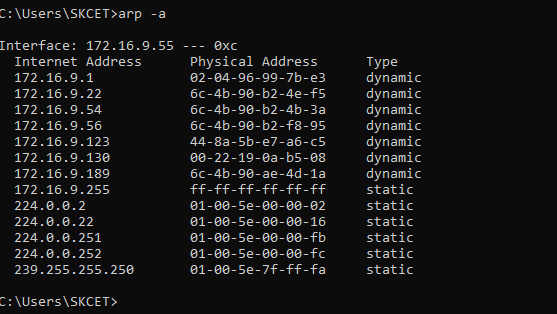
To study the network commands(windows based) by using command prompt and also find the IP address for familier websites.

# PROCEDURE :

1. Open a new terminal
2. Type the following commands in the terminal and get the output COMMANDS AND OUTPUT :
3. Command : **arp -a**

The arp -a command is used to display the current ARP (Address Resolution Protocol) table on a Windows computer. ARP is a protocol used to map an IP address to a physical (MAC) address on a local network.

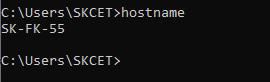
Output :



1. Command : **hostname**

It will display the name of the host.

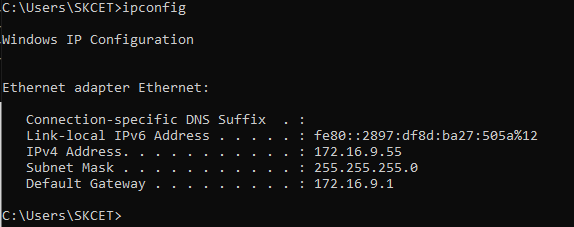
Output :



1. Command : **ipconfig**

The ipconfig command is a command-line utility used in Windows operating systems to display the configuration of the network interfaces on your computer. It provides information about the IP addresses, subnet masks, default gateways, and other network-related details for all active network interfaces.

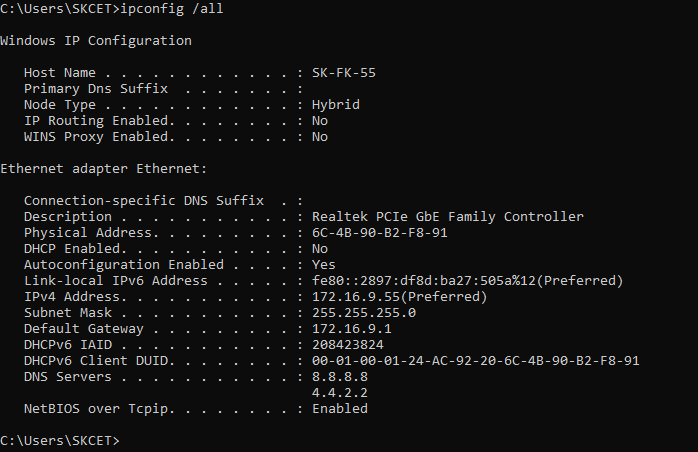
Output :



1. Command : **ipconfig/all**

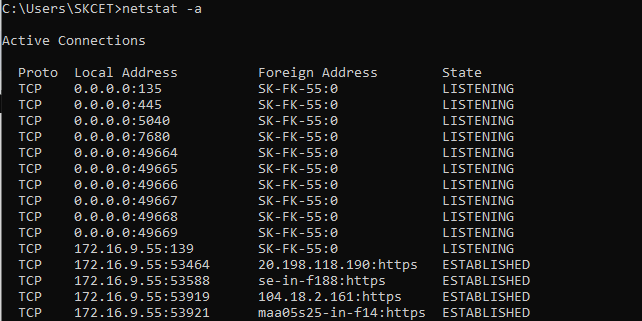
The ipconfig /all command is an extended version of the ipconfig command in Windows. It displays detailed information about all network interfaces, including physical and virtual ones, and provides additional information such as DNS server settings, DHCP lease information, and more.

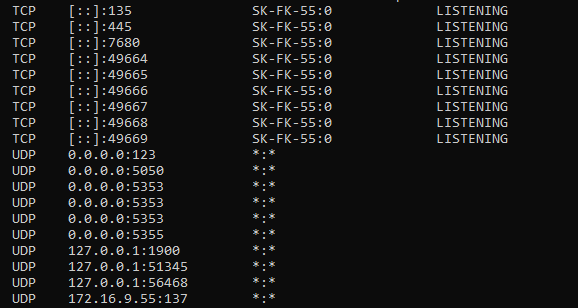
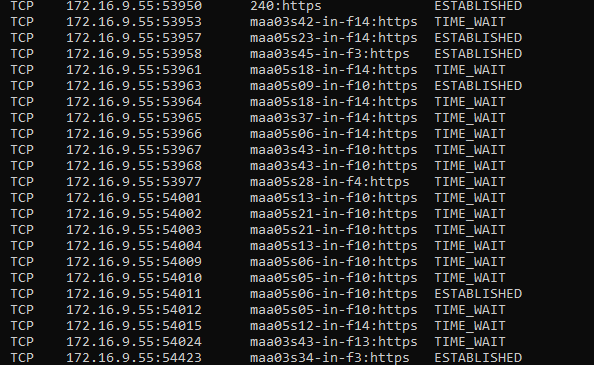
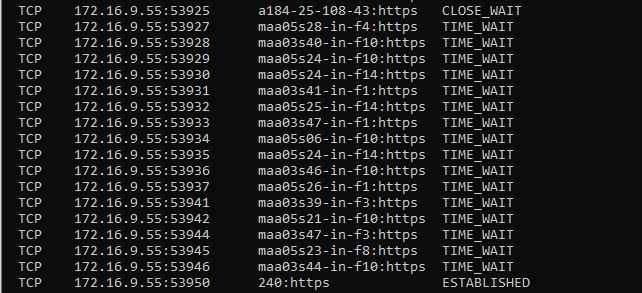
Output :

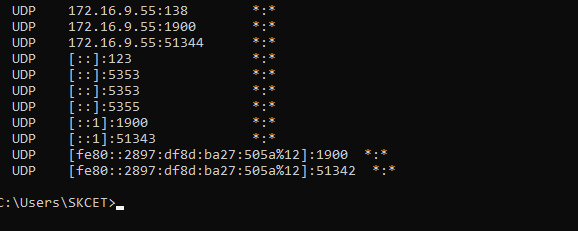


1. Command : **netstat -a**

The netstat -a command is used to display active network connections on a computer, along with the listening ports and their associated addresses. It provides a list of all open network connections, both incoming and outgoing, on the system. This command can be useful for troubleshooting network issues, monitoring network activity, and checking which ports are in use.

Output :



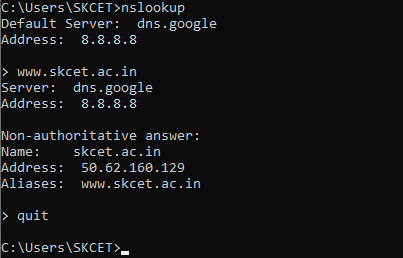


1. Command : **nslookup**

“nslookup” , which stands for "Name Server Lookup," is a command-line tool used for querying DNS (Domain Name System) servers to obtain domain name or IP address information. It is available on various operating systems, including Windows, macOS, and Linux, and is used to troubleshoot DNS-related issues, resolve domain names to IP addresses, and vice versa.

Output :

Includes [www.skcet.ac.in](http://www.skcet.ac.in/)

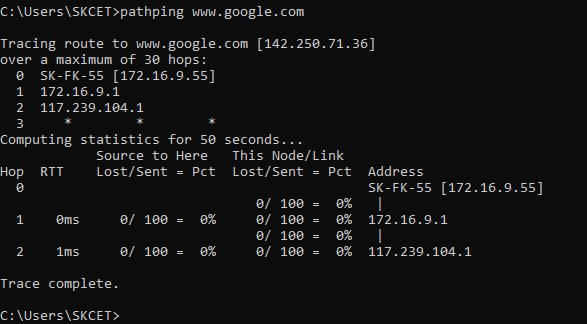


1. Command : **pathping** [**www.google.com**](http://www.google.com/)

**pathping** is a command-line network utility available in Windows operating systems. It combines the functionality of the ping command and the tracert (or traceroute in Unix-like systems) command to provide more detailed information about the network path between your

computer and a remote destination, such as a website or server. It is often used for network troubleshooting to identify and diagnose network issues along the route.

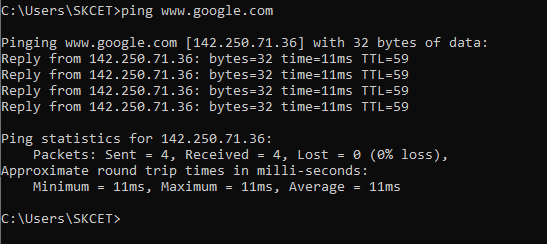
Output :



1. Command : **ping** [**www.google.com**](http://www.google.com/)

The ping command is a network utility available on most operating systems, including Windows, macOS, and Linux. It is used to test the connectivity between your computer and a remote host (usually specified by its IP address or domain name). The ping command sends ICMP (Internet Control Message Protocol) echo requests to the remote host and waits for ICMP echo replies. This is a simple and widely used tool for checking network connectivity and latency.

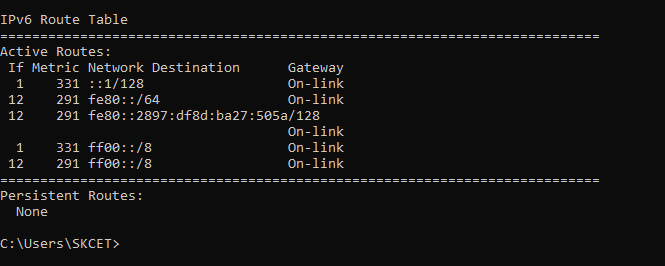
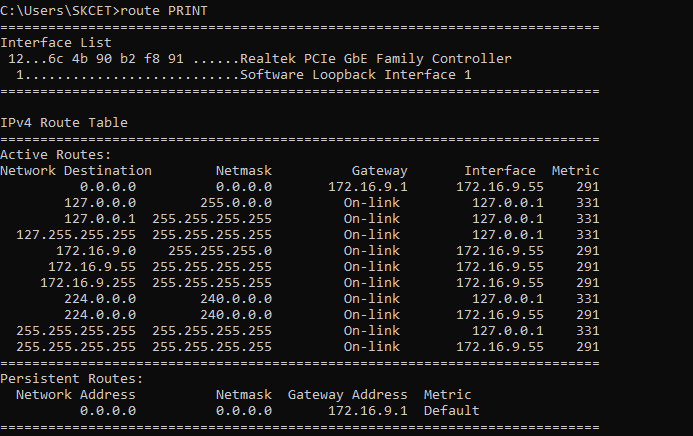
Output :



1. Command : **route PRINT**

The route command is a command-line utility used in various operating systems, including Windows, macOS, and Linux, to display and manipulate the routing table of a computer or network device. The routing table is a critical component of a network that helps determine how network traffic is directed to its destination. It contains information about which network interface to use and which gateway (router) to send data packets through to reach specific IP addresses or networks.

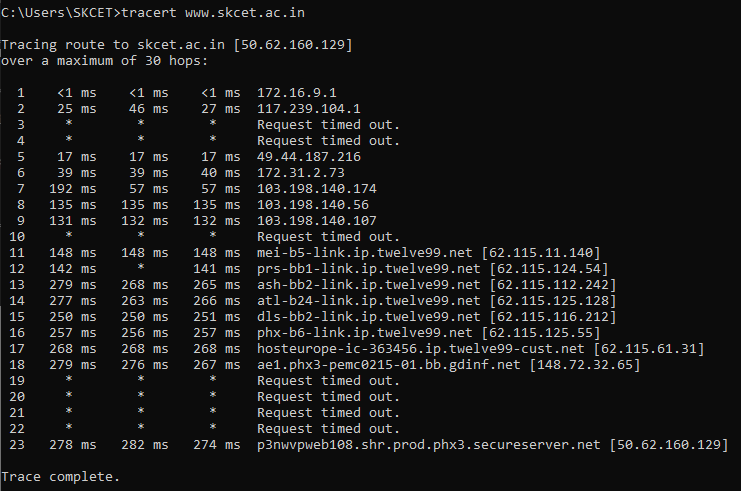
Output :



1. Command : **tracert** [**www.skcet.ac.in**](http://www.skcet.ac.in/)

The tracert (short for "traceroute" in Unix-like systems) command is a network utility used to trace the route that packets take from your computer to a destination host (specified by an IP address or domain name). It helps you visualize the network path and identify the individual network devices (routers) through which the packets pass to reach their destination. This command is available on Windows, macOS, and most Linux/Unix-like operating systems.

Output :



# RESULT

Thus the study of network commands through command prompt and few real time IP address were identified.

|  |  |
| --- | --- |
| EXP NO : 2 | **CREATION OF SIMPLE LAN** |
| DATE : |

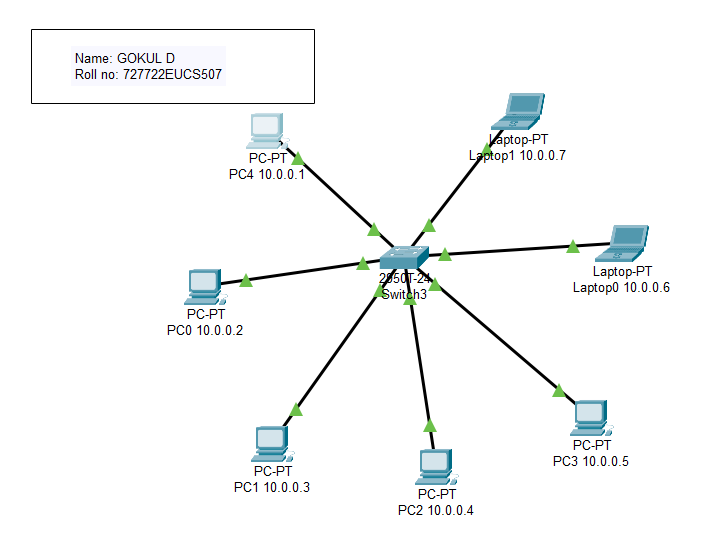
# AIM:

To create a simple LAN connection with 5 PCs and 2 Laptops and a Switch and test the connection.

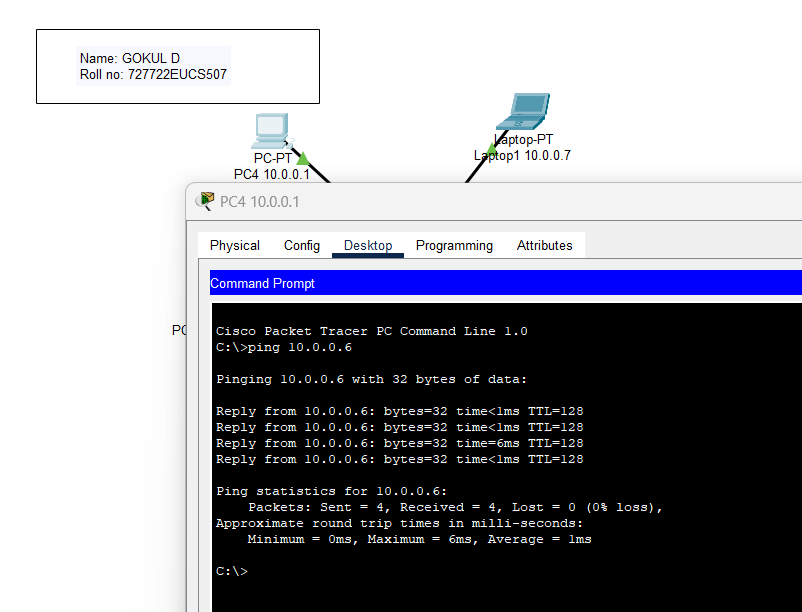
# 2A)WIRED LAN PROCEDURE

1. Procedure to create the network
   * Open a new workspace to create a network in a packet tracer.
   * From end devices choose switch , PC and laptop.
   * Connect these devices via cable.
2. Procedure to configure the network
   * Configure the static IPs for the end devices with 10.0.0.2 , 10.0.0.3 and so on and add subnet mask as 255.0.0.0.
3. Procedure to test the network
   * Use the ping command in the command prompt to check for successful transmission between the two networks.
   * If the IP address gets pinged in other system successfully then the network is working successfully.

# OUTPUT:

****

**ping 10.0.0.1 to 10.0.0.6 (PC to Laptop)**

****

# 2B)WIRELESS LAN PROCEDURE

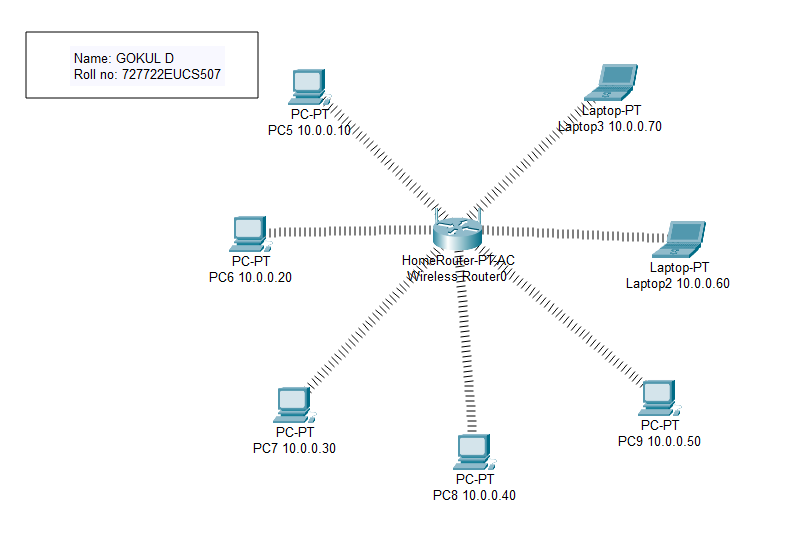
1. Procedure to create the network
   * Open a new workspace to create a network in a packet tracer.
   * From end devices choose switch , PC , laptop and a router (WRT300N wireless router). ii)Procedure to configure the network
   * Configure the static IPs for the end devices with 11.0.0.2 , 11.0.0.3 and so on and add

subnet mask as 255.0.0.0.

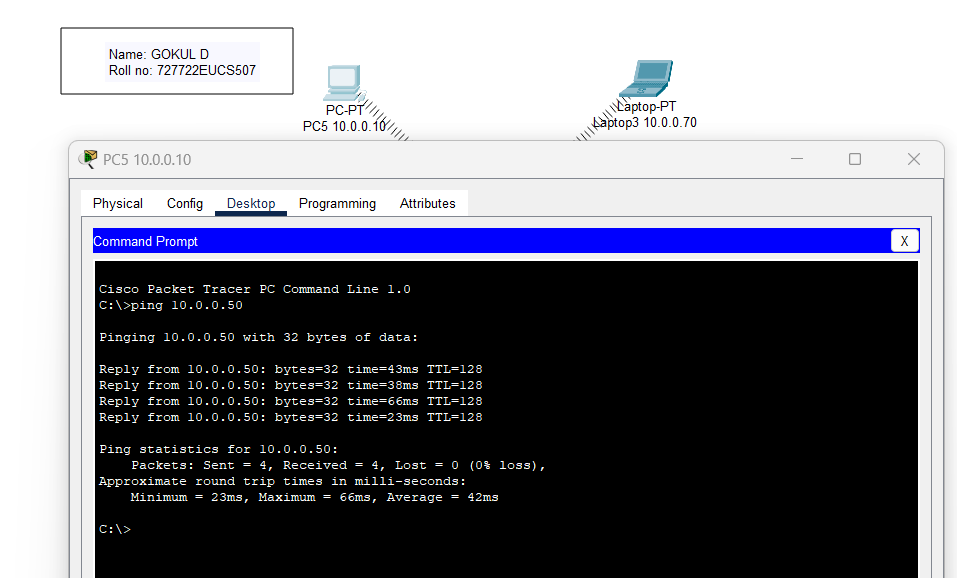
* + Now in each of the end devices , make sure to power off and remove the LAN port and add “WMP300N” and then power on the devices.
  + In the config section of the wireless router , in Wireless 2.4G section , provide the SSID for the wireless router and add the same SSID in the end devices of the network.

1. Procedure to test the network
   * Use the ping command in the command prompt to check for successful transmission between the two networks.
   * If the IP address gets pinged in other system successfully then the network is working successfully.

# OUTPUT:

****

**ping 10.0.0.10 to 10.0.0.50 (PC to PC)**

****

# RESULT

Thus, the connection of wired and wireless network has been established successfully.

|  |  |
| --- | --- |
| EXP NO : 3 | **WIRED AND WIRELESS LAN WITH ROUTER** |
| DATE : |

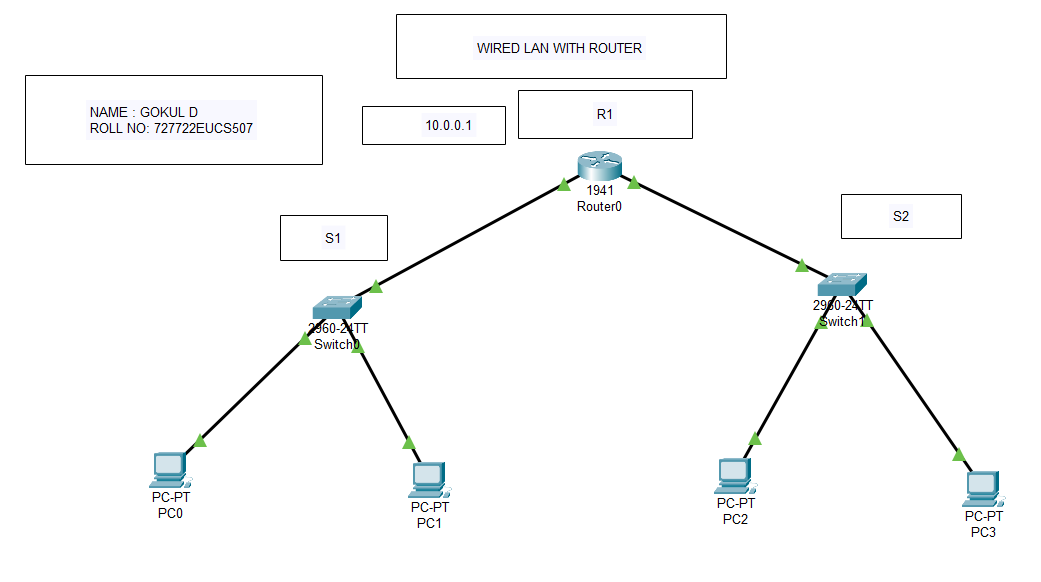
# AIM:

To create a simple wired LAN with router and add 2 PC and a switch (4) along with router and to test the connection.

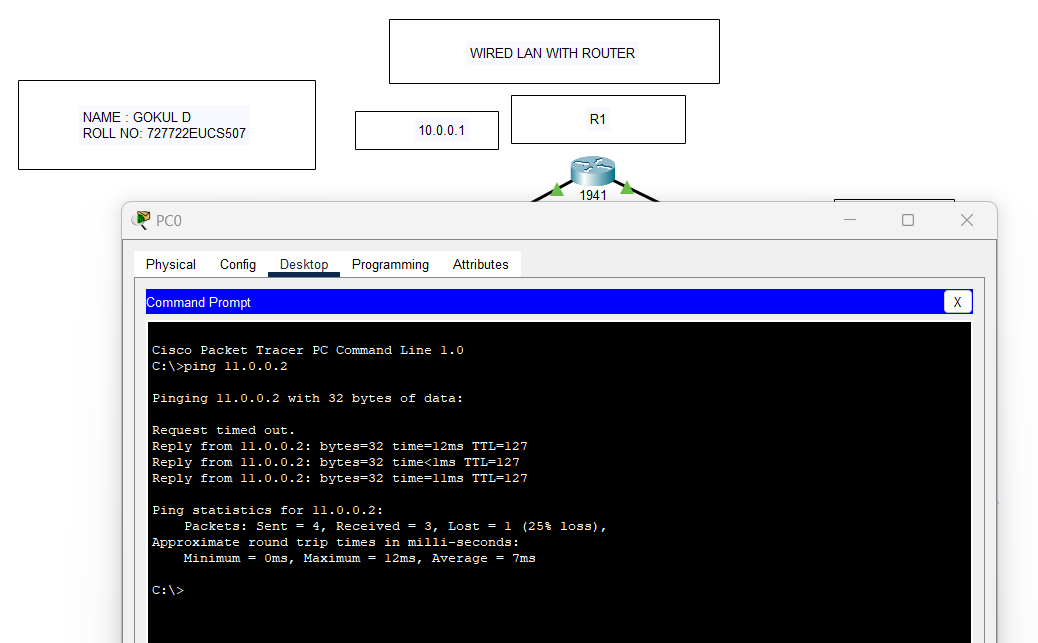
# 3A)WIRED LAN WITH ROUTER PROCEDURE:

1. Procedure to create the network
   * Open a new workspace to create a network in a packet tracer.
   * From end devices , choose switch , router PC.
   * Connect these devices via cable. ii)Procedure to configure the network
   * Configure the static IPs for the end devices of first network with 10.0.0.2 and so on.
   * Then change the IP address in the GigabitEthernet 0/0 of router as 10.0.0.1.
   * Repeat the same procedure for other networks.
   * (i.e) Change the IP address of PC of 2nd network as 11.0.0.2 and so on and in the GigabitEthernet 0/1of router as 11.0.0.1.
   * In each configuration of the first network, add subnet mask as 255.255.255.0 and default gateway as 10.0.0.1 in the PC’s.
   * In each configuration of the second network, add subnet mask as 255.255.255.0 and default gateway as 11.0.0.1 in the PC’s.
2. Procedure to test the network
   * Use the ping command in the command prompt to check for successful transmission between the two networks.
   * If the IP address gets pinged in other system successfully then the network is working successfully.

# OUTPUT:

****

**Ping 10.0.0.2 to 11.0.0.2 ( from LAN1 to LAN2 )**

****

# 3B)WIRELESS LAN WITH ROUTER

**PROCEDURE**

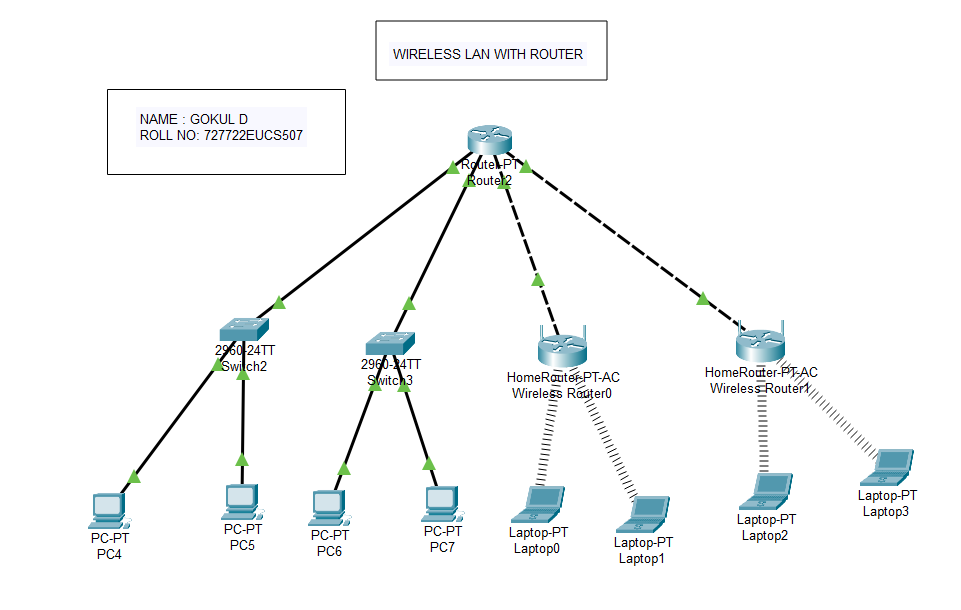
1. Procedure to create the network
   * Open a new workspace to create a network in a packet tracer.
   * From end devices choose switch , router ,wireless router and PC.
   * Connect these devices via cable. ii)Procedure to configure the network
   * Configure the static IPs for the end devices of first network with 192.168.0.2 and so on.
   * Then change the IP address in the FastEthernet 0/0 of router as192.168.0.1.
   * Repeat the same procedure for other networks with the IP 10.0.0.2 sequence and

11.0.0.2 sequence.

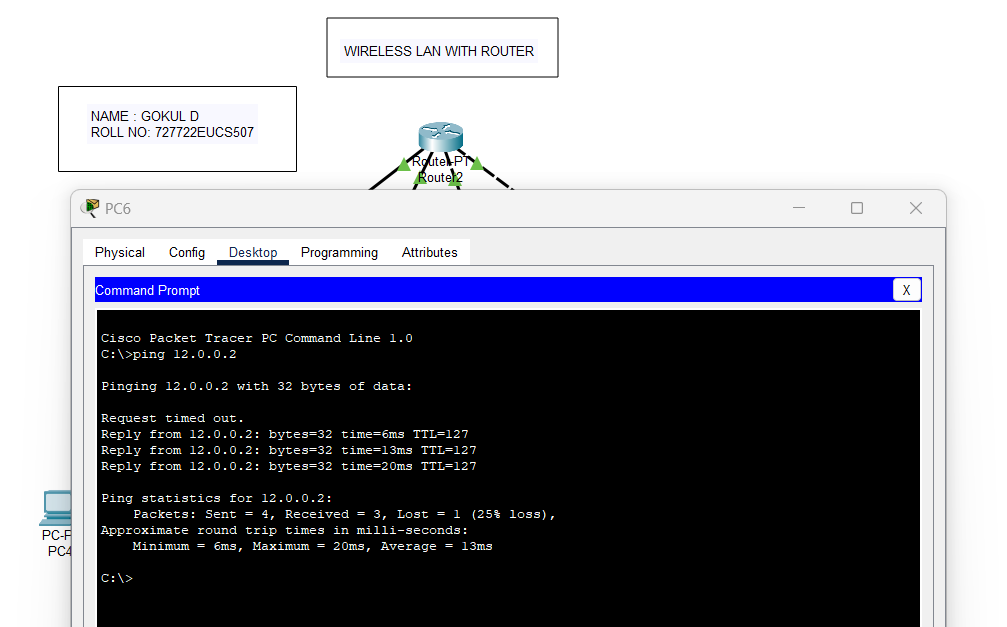
* + According to those sequence add the IP address in the GigabitEthernet of the router.
  + In each configuration of the devices, add subnet mask as 255.255.255.0 and default gateway as 10.0.0.1, 11.0.0.1 and 192.168.0.1 respectively in the end devices.
  + Now in each of the end devices , make sure to power off and remove the LAN port and add “WMP300N” and then power on the devices.
  + In the config section of the wireless router , in Wireless 2.4G section , provide the SSID for the wireless router and add the same SSID in the respective end devices of the network.

1. Procedure to test the network
   * Use the ping command in the command prompt to check for successful transmission between the two networks.
   * If the IP address gets pinged in other system successfully then the network is working successfully.

# OUTPUT:

****

**ping 11.0.0.2 to 12.0.0.2 (PC from LAN 2 to PC of LAN3)**

****

# RESULT

Thus, the connection using wired and wireless LAN using router is completed successfully.

|  |  |
| --- | --- |
| EXP NO : 4 | **ROUTER PROTOCOL**  **INTER ROUTER COMMUNICATION** |
| DATE : 13.09.2023 |

# AIM :

To connect two different networks using two routers by applying two routing protocols

on Cisco Packet Tracer.

# DYNAMIC ROUTING PROTOCOL :

**PROCEDURE:**

1. Open Cisco Packet Tracer in your system and login into your account.
2. To create the first network, in the bottom of the left corner, select “End Devices”. Drag 3 PCs and drop them on the screen (PC0, PC1, PC2).
3. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1 switch and drop it on the screen between the PCs.
4. Connect each PC with the switch via a cable from the connection section.
5. To configure the IP address of each PC for packet transmission, double-click on a PC, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 10.0.0.2.
6. Repeat step 5 for configuring each PC and type the IP address as 10.0.0.3 and so on.
7. In each configuration, add subnet mask as 255.0.0.0 and default gateway as 10.0.0.1
8. To create the second network, in the bottom of the left corner, select “End Devices”. Drag 2 PCs and drop them on the screen (PC3, PC4).
9. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1 switch and drop it on the screen between the PCs.
10. Connect each PC with the switch via a cable from the connection section.
11. To configure the IP address of each PC for packet transmission, double-click on a PC, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 192.168.1.2 and 192.168.1.3
12. In each configuration, add subnet mask as 255.255.255.0 and default gateway as 192.168.1.1
13. Select “4331 Router” from the Routers section. Drag two of them and drop them on the screen between the two networks.
14. Connect the two routers using cross wire
15. To configure one side of the router, click on the router, and add 10.0.0.1 as IP address in Gig0/0/0 (place the cursor and check the port) and turn it on. Add 11.0.0.1 in Gig0/0/1 for connecting to the second router, and turn it on.
16. To configure the other side of the router, click on the router, and add 192.168.1.1 as IP address in Gig0/0/1 (place the cursor and check the port) and turn it on. Add 11.0.0.2 in Gig0/0/1 for connecting to the second router, and turn it on.
17. Dynamic routing protocol- Add 10.0.0.0, 11.0.0.0 and 192.168.1.0 in the RIP section of both the routers.
18. Use the ping command in the command prompt to check for successful transmission between the two networks.

# STATIC ROUTING PROTOCOL :

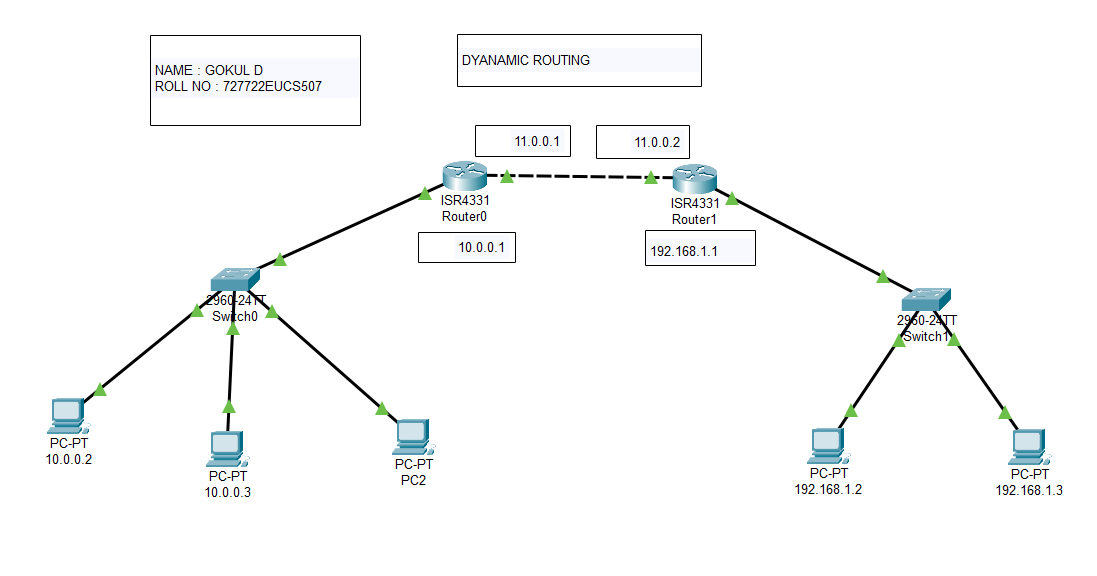
**PROCEDURE:**

1. Open Cisco Packet Tracer in your system and login into your account.
2. To create the first network, in the bottom of the left corner, select “End Devices”. Drag 3 PCs and drop them on the screen (PC0, PC1, PC2).
3. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1 switch and drop it on the screen between the PCs.
4. Connect each PC with the switch via a cable from the connection section.
5. To configure the IP address of each PC for packet transmission, double-click on a PC, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 10.0.1.2.
6. Repeat step 5 for configuring each PC and type the IP address as 10.0.1.3 and so on.
7. In each configuration, add subnet mask as 255.0.0.0 and default gateway as 10.0.1.1
8. To create the second network, in the bottom of the left corner, select “End Devices”. Drag 2 PCs and drop them on the screen (PC3, PC4).
9. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1 switch and drop it on the screen between the PCs.
10. Connect each PC with the switch via a cable from the connection section.
11. To configure the IP address of each PC for packet transmission, double-click on a PC, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 192.168.2.2 and 192.168.2.3
12. In each configuration, add subnet mask as 255.255.255.0 and default gateway as 192.168.2.1
13. Select “4331 Router” from the Routers section. Drag two of them and drop them on the screen between the two networks.
14. Connect the two routers using cross wire .
15. To configure one side of the router, click on the router, and add 10.0.1.1 as IP address in Gig0/0/0 (place the cursor and check the port) and turn it on. Add 11.0.1.1 in Gig0/0/1 for connecting to the second router, and turn it on.
16. To configure the other side of the router, click on the router, and add 192.168.2.1 as IP address in Gig0/0/1 (place the cursor and check the port) and turn it on. Add 11.0.1.2 in Gig0/0/1 for connecting to the second router, and turn it on.
17. Static routing protocol- Add 10.0.1.0 and 192.168.2.0 as IP and 11.0.1.2 as next hop on the first router and 192.168.2.0 and 10.0.1.0 as IP and 11.0.1.1 as next hop on the second router.
18. Use the ping command in the command prompt to check for successful transmission between the two networks.

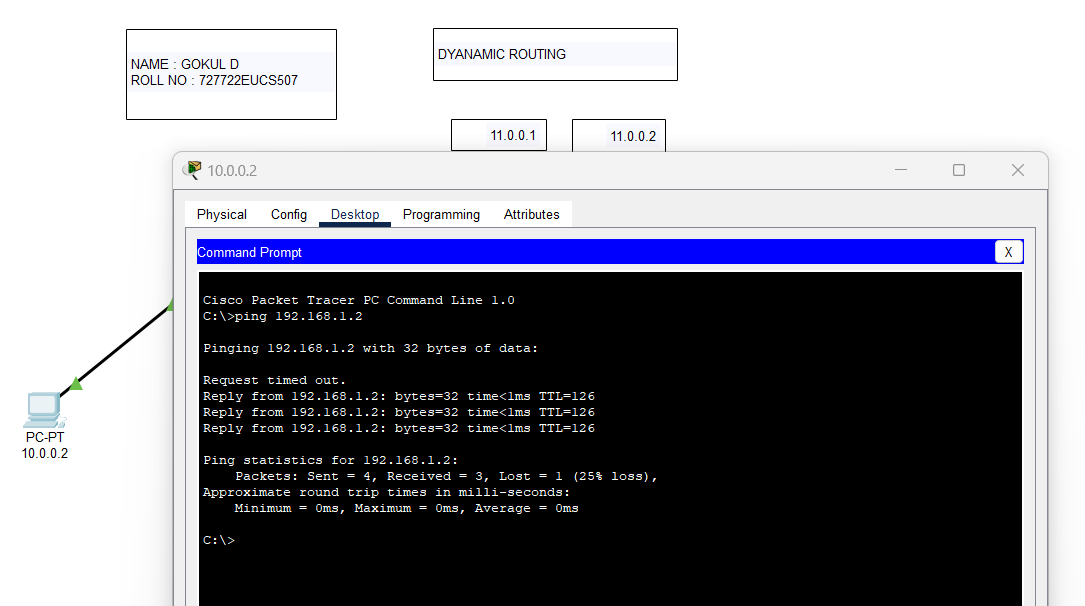
# OUTPUT :

**ROUTING PROTOCOL: Static & Dynamic Configurations NETWORK DIAGRAM:**

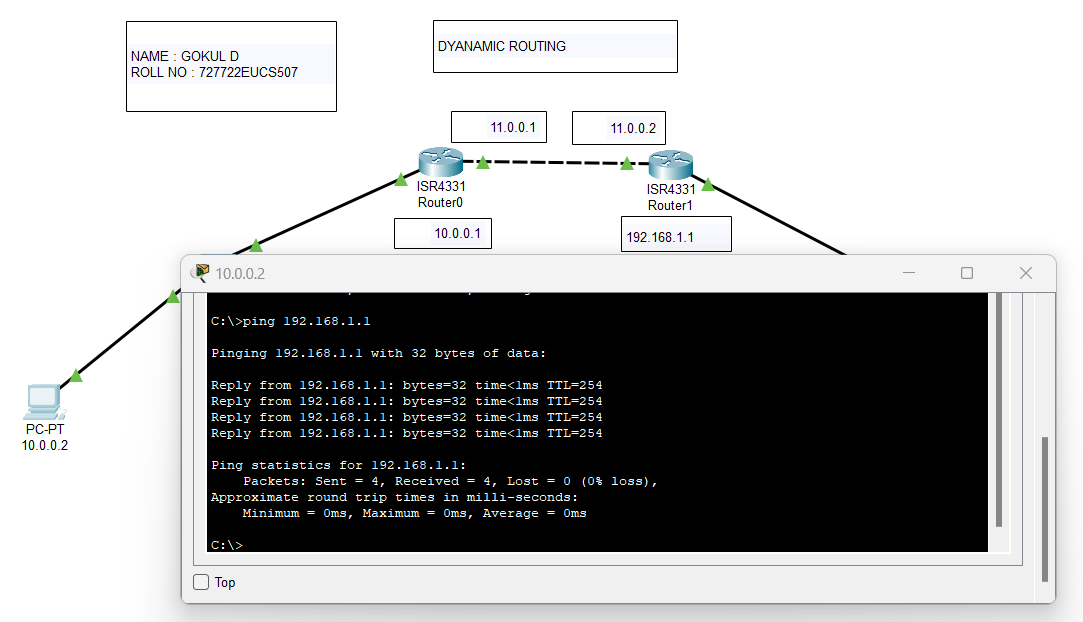
**DYANAMIC ROUTING:**

****

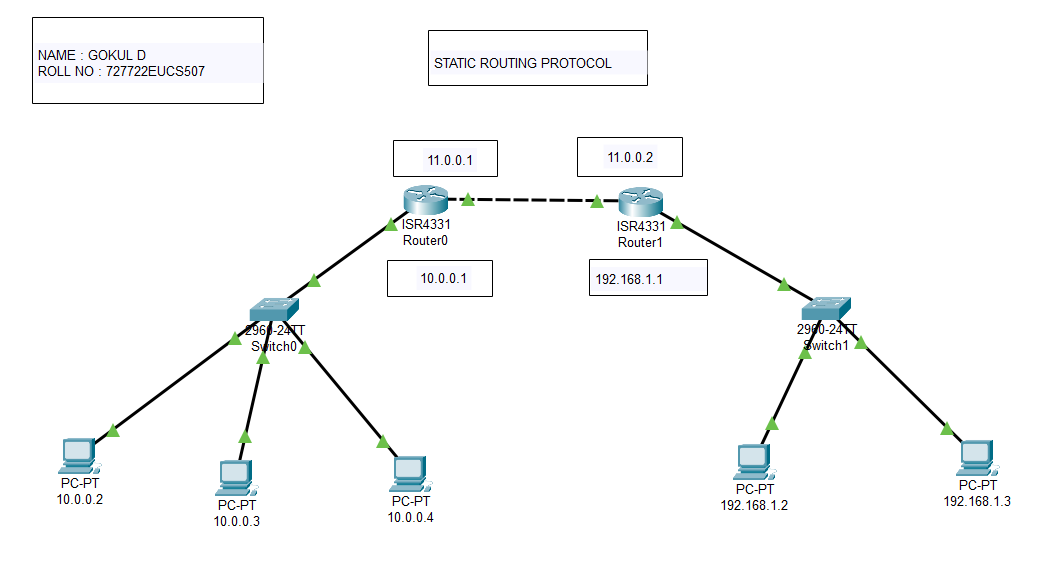
**ping 10.0.0.2 to 192.168.1.2 (PC from LAN1 to PC from LAN2)**

****

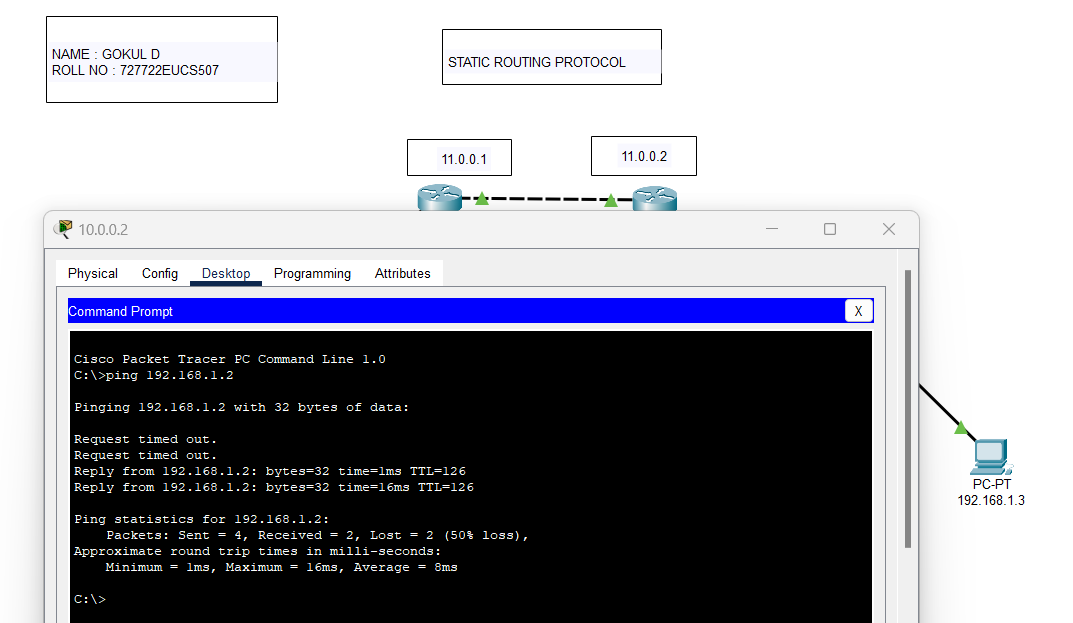
**ping 10.0.0.2 to 192.168.1.1 (PC from LAN1 to router of LAN2)**

****

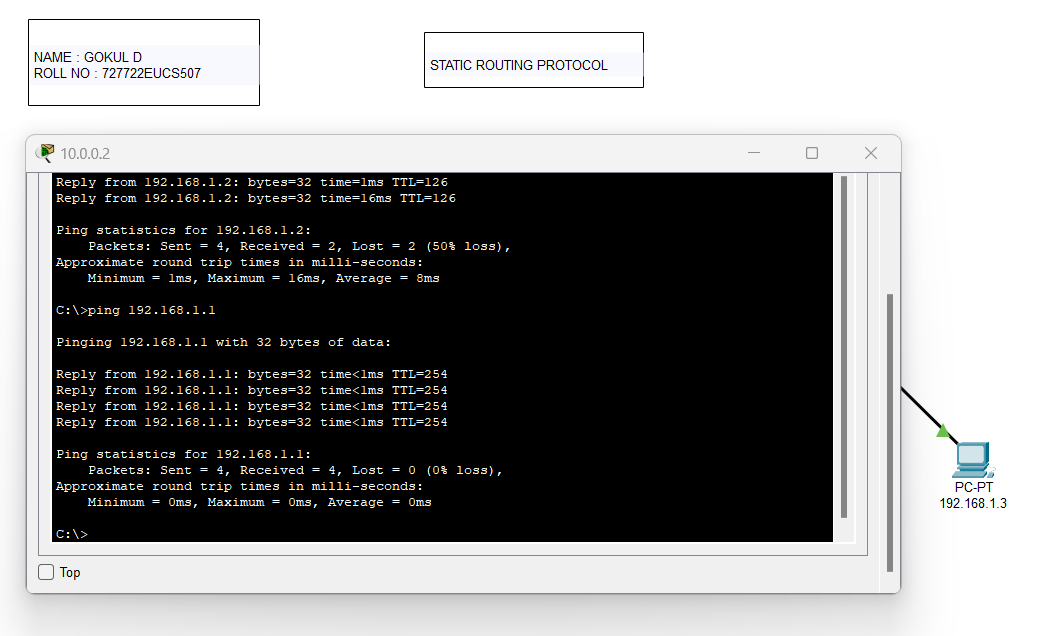
**STATIC ROUTING:**

****

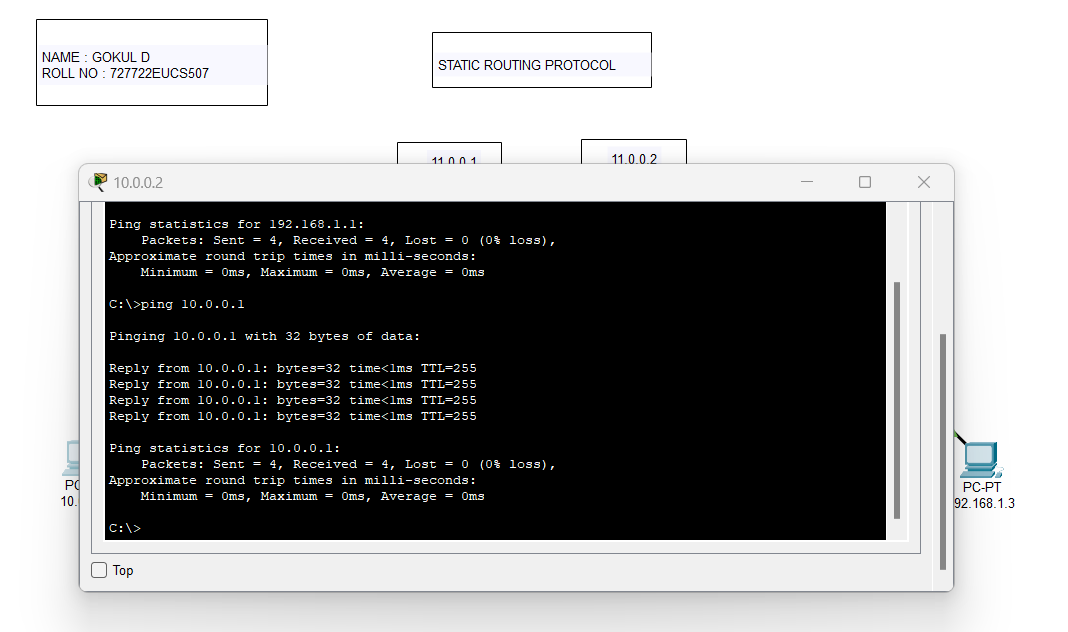
**ping 10.0.0.2 to 192.168.1.2 (PC from LAN1 to PC from LAN2)**

****

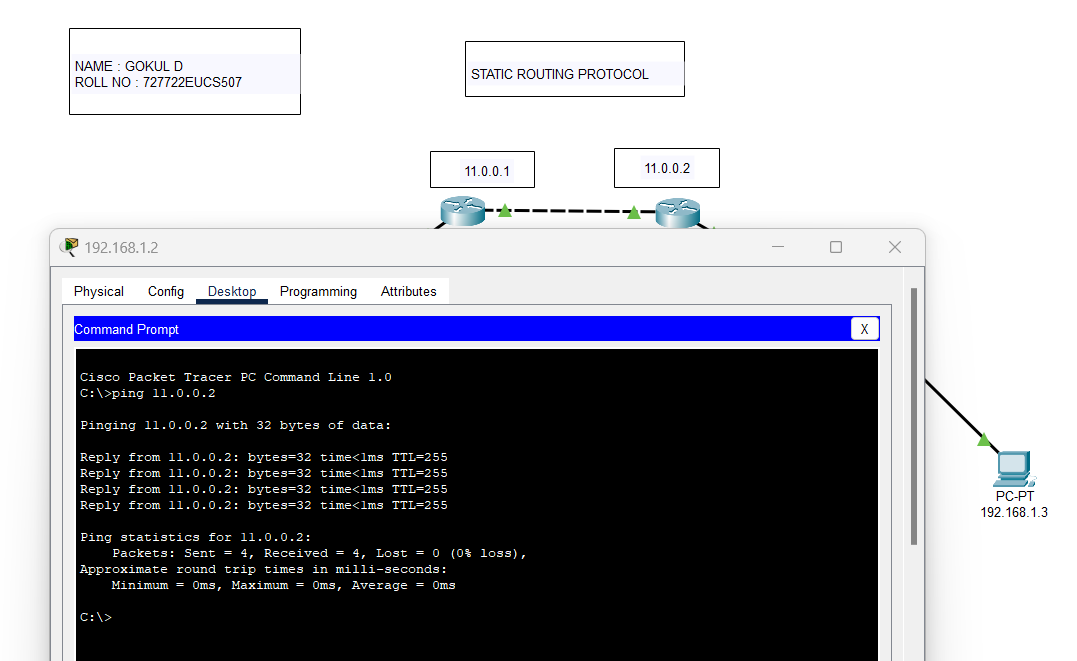
**ping 10.0.0.2 to 192.168.1.1 (PC from LAN1 to router of LAN2)**

****

**ping 10.0.0.2 to 10.0.0.1 (PC from LAN1 to router of LAN1)**

****

**ping 192.168.1.2 to 11.0.0.2 (PC from LAN2 to router of LAN2)**

****

# RESULT :

Hence, the two networks are interconnected with two routers by applying two different routing protocols using Packet Tracer.

|  |  |
| --- | --- |
| EXP NO : 5 | **SIMULATION OF DNS USING PACKET TRACER** |
| DATE : |

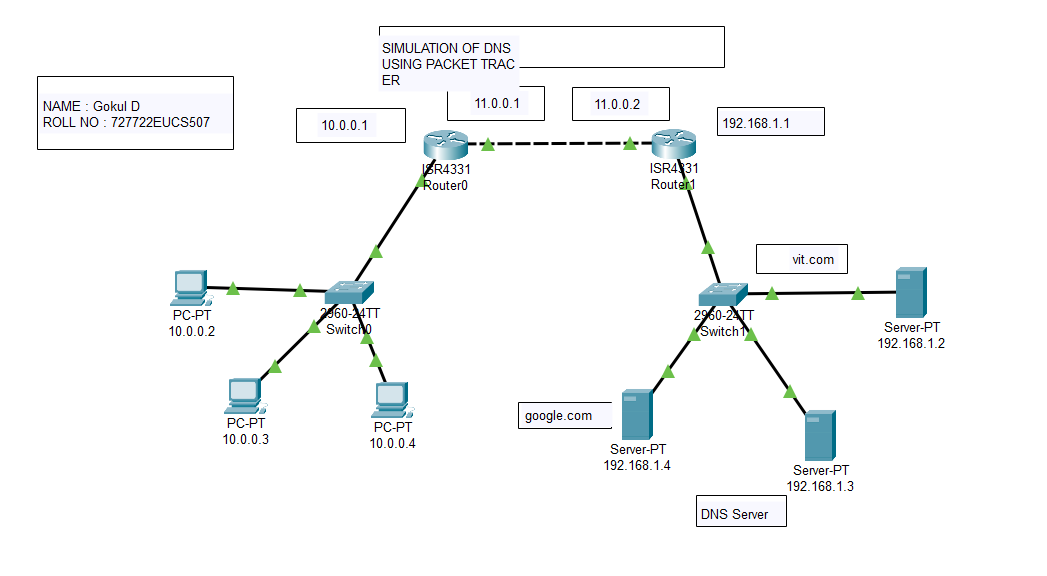
# AIM :

To simulate DNS on server using Cisco packet tracer

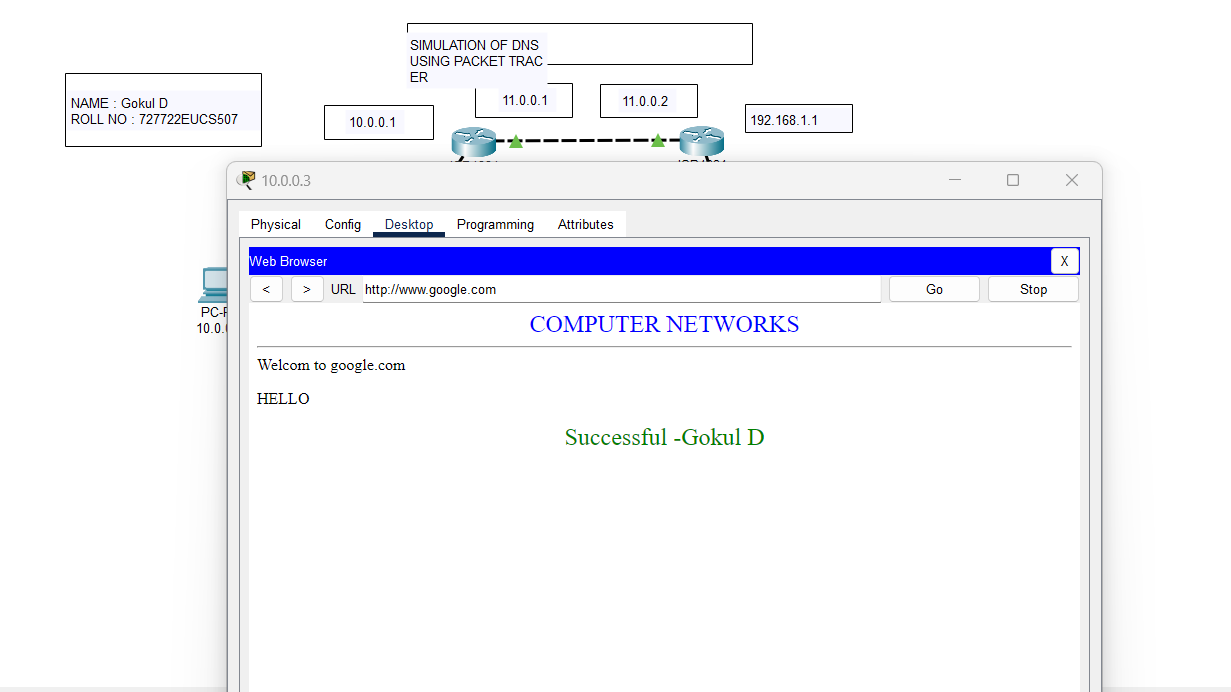
# PROCEDURE:

1. Open Cisco Packet Tracer in your system and login into your account.
2. To create the first network, in the bottom of the left corner, select “End Devices”. Drag 3 PCs and drop them on the screen (PC0, PC1, PC2).
3. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1 switch and drop it on the screen between the PCs.
4. Connect each PC with the switch via a cable from the connection section.
5. To configure the IP address of each PC for packet transmission, double-click on a PC, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 10.0.0.2.
6. Repeat step 5 for configuring each PC and type the IP address as 10.0.0.3 and so on.
7. In each configuration, add subnet mask as 255.0.0.0 and default gateway as 10.0.0.1
8. To create the second network, in the bottom of the left corner, select “End Devices”. Drag one Server and drop it on the screen.
9. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1switch and drop it on the screen and connect the server.
10. To configure the IP address of the server for packet transmission, double-click on it, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 192.168.1.2.
11. Add subnet mask as 255.255.255.0 and default gateway as 192.168.1.1
12. Select “4331 Router” from the Routers section. Drag two of them and drop them on the screen between the two networks.
13. Connect the two routers using cross wire .
14. To configure one side of the router, click on the router, and add 10.0.0.1 as IP address in Gig0/0/0 (place the cursor and check the port) and turn it on. Add 11.0.0.1 in Gig0/0/1 for connecting to the second router, and turn it on.
15. To configure the other side of the router, click on the router, and add 192.168.1.1 as IP address in Gig0/0/1 (place the cursor and check the port) and turn it on. Add 11.0.0.2 in Gig0/0/1 for connecting to the second router, and turn it on.
16. Dynamic routing protocol- Add 10.0.0.0, 11.0.0.0 and 192.168.1.0 in the RIP section of both the routers.
17. HTTP Configuration: Inside Services section in the server, turn on HTTP. Edit the index.html file and add content (for eg.: VIT Chennai)
18. Add a server for google.com with IP address 192.168.1.4 and add the google page source content.
19. DNS Configuration: Add one more server for DNS in the second network. Add its IP address as 192.168.1.3 and add it in the DNS Server section too. In the services section, click on DNS and turn it on. Add the web page name for eg: vit.com along with its IP address (192.168.1.2) and google.com along with its IP address (192.168.1.4).
20. Add DNS Server IP address 192.168.1.3 in all the End Devices.
21. From any PC, type vit.com and google.com to check if the webpage opens.

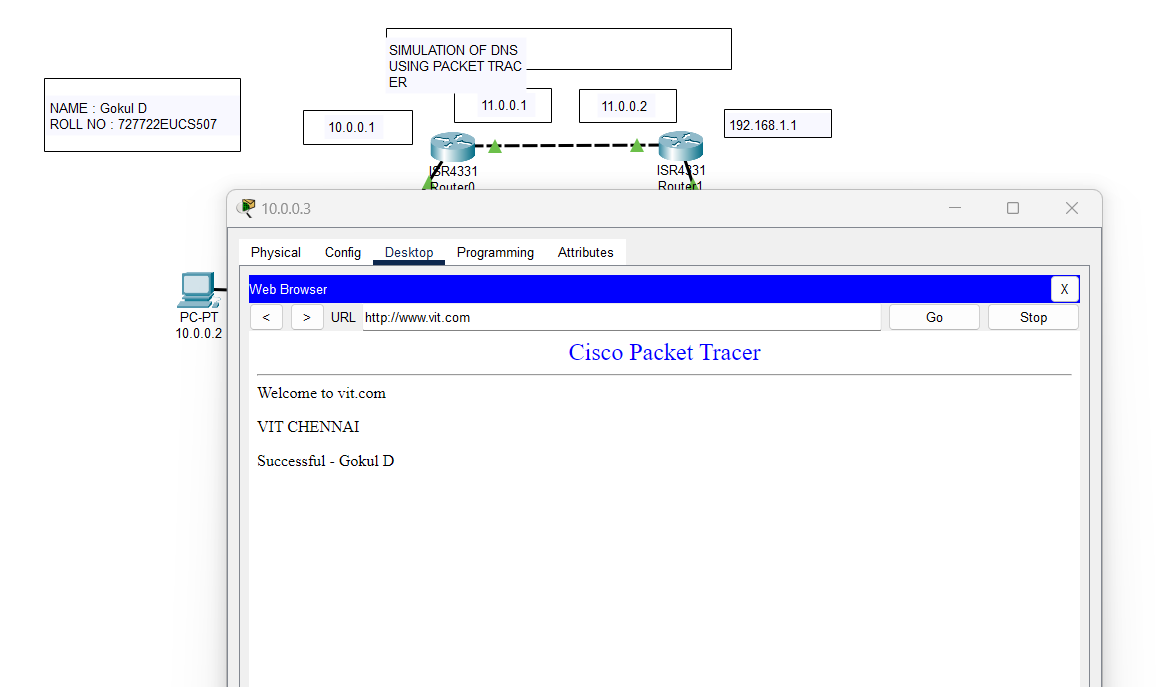
# OUTPUT :

****

**Provide** [**www.google.com**](http://www.google.com/) **URL in the web browser of the PC with IP 10.0.0.3**

****

# Provide [www.vit.com](http://www.vit.com/) URL in the web browser of the PC with IP 10.0.0.3



**RESULT:**

Hence, DNS server is simulated using Cisco Packet Tracer and 2 webpages are searched from a PC.

|  |  |
| --- | --- |
| EXP NO : 6 | **SIMULATION OF HTTP PROTOCOL USING PACKET TRACER** |
| DATE : |

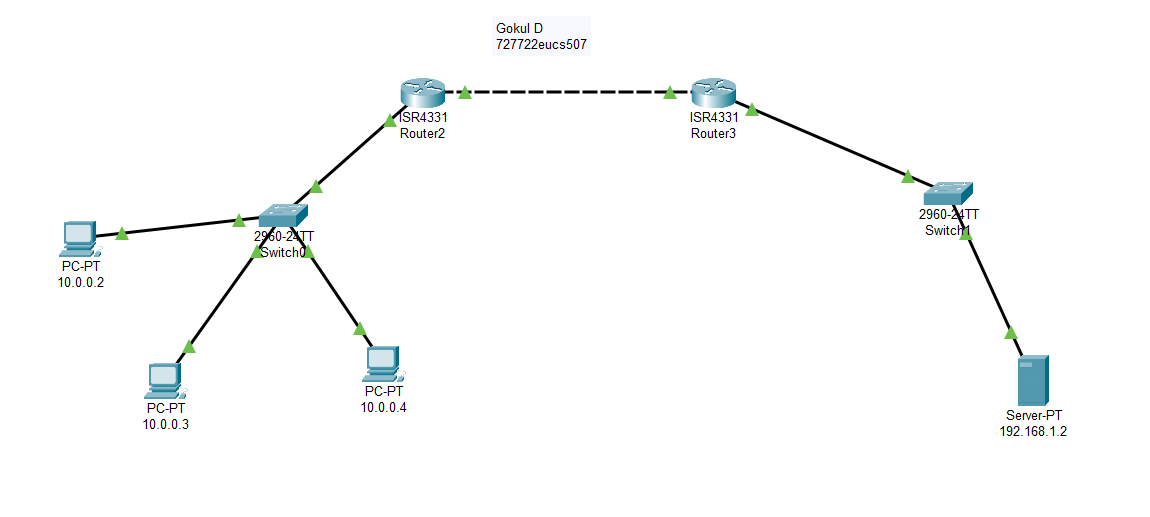
# AIM :

To simulate http protocol using Cisco packet tracer**.**

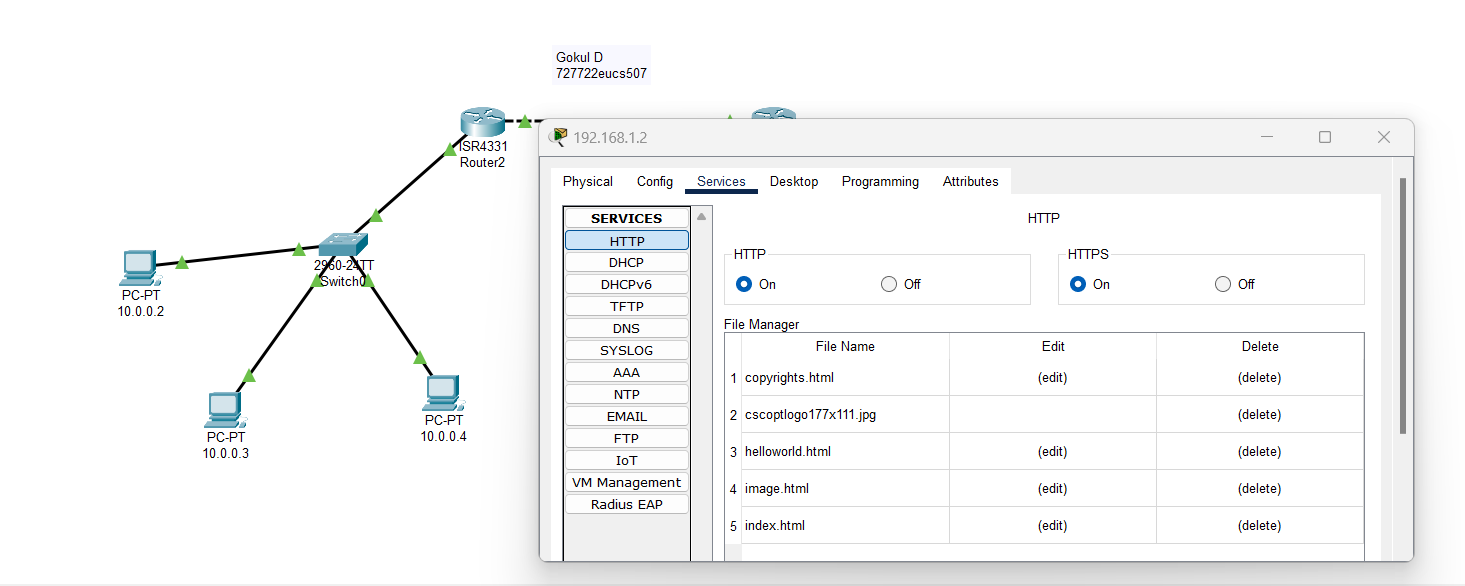
# PROCEDURE:

1. Open Cisco Packet Tracer in your system and login into your account.
2. To create the first network, in the bottom of the left corner, select “End Devices”. Drag 3 PCs and drop them on the screen (PC0, PC1, PC2).
3. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1 switch and drop it on the screen between the PCs.
4. Connect each PC with the switch via a cable from the connection section.
5. To configure the IP address of each PC for packet transmission, double-click on a PC, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 10.0.0.2.
6. Repeat step 5 for configuring each PC and type the IP address as 10.0.0.3 and so on.
7. In each configuration, add subnet mask as 255.0.0.0 and default gateway as 10.0.0.1
8. To create the second network, in the bottom of the left corner, select “End Devices”. Drag one Server and drop it on the screen.
9. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1 switch and drop it on the screen and connect the server.
10. To configure the IP address of the server for packet transmission, double-click on it, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 192.168.1.2.
11. Add subnet mask as 255.255.255.0 and default gateway as 192.168.1.1
12. Select “4331 Router” from the Routers section. Drag two of them and drop them on the screen between the two networks.
13. Connect the two routers using cross wire .
14. To configure one side of the router, click on the router, and add 10.0.0.1 as IP address in Gig0/0/0 (place the cursor and check the port) and turn it on. Add 11.0.0.1 in Gig0/0/1 for connecting to the second router, and turn it on.
15. To configure the other side of the router, click on the router, and add 192.168.1.1 as IP address in Gig0/0/1 (place the cursor and check the port) and turn it on. Add 11.0.0.2 in Gig0/0/1 for connecting to the second router, and turn it on.
16. Dynamic routing protocol- Add 10.0.0.0, 11.0.0.0 and 192.168.1.0 in the RIP section of both the routers.
17. HTTP Configuration: Inside Services section in the server, turn on HTTP. Edit the index.html file and add your content.
18. Check for the web page in the web browser section of any PC by searching for the server IP address 192.168.1.2

# NETWORK DIAGRAM :

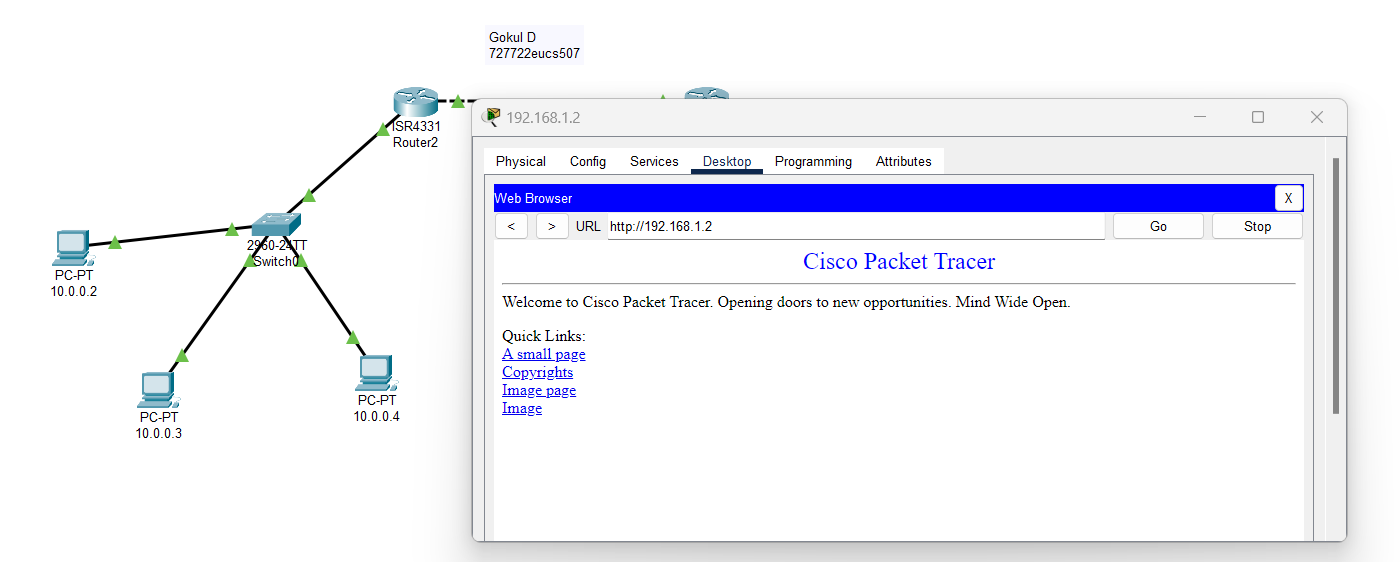
****

**HTTP Configuration index.html file:**

****

# OUTPUT :

**Searching IP address ( 192.168.1.2 ) from PC0 ( 10.0.02 ) :**

****

# RESULT:

Hence, HTTP protocol is simulated using Cisco Packet Tracer and one webpage is searched from a PC.

|  |  |
| --- | --- |
| EXP NO : 7 | **CONFIGURE SMTP USING PACKET TRACER** |
| DATE : |

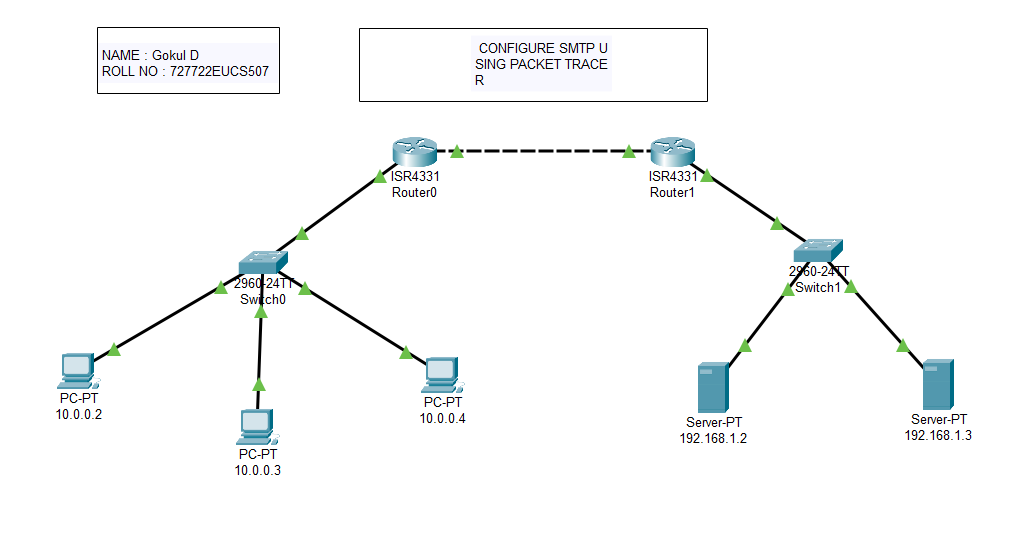
# AIM :

To configure SMTP protocol using Cisco packet tracer**.**

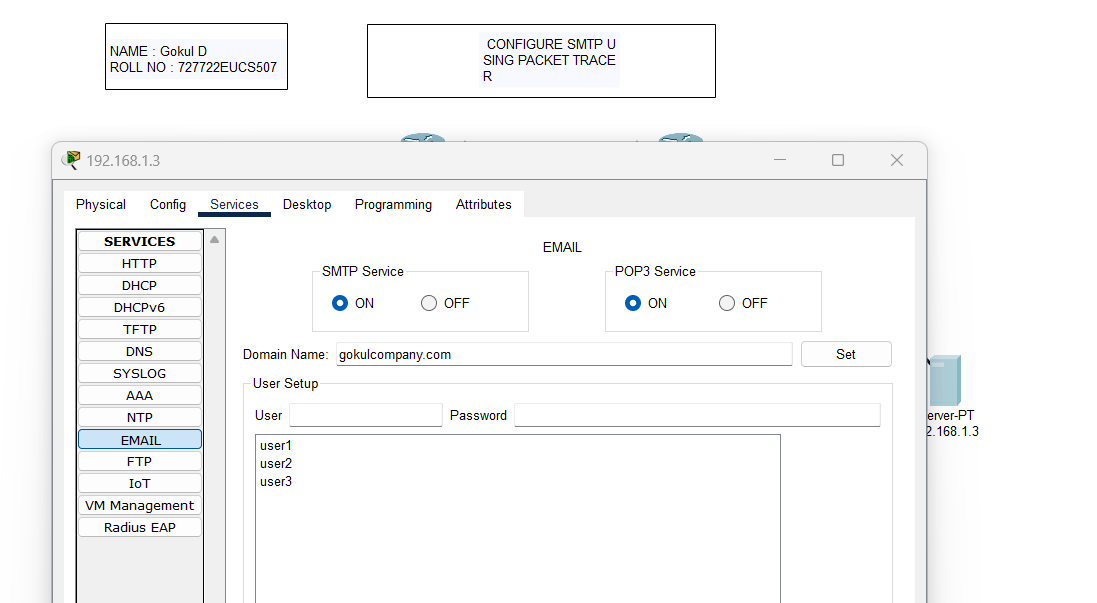
# PROCEDURE:

1. Open Cisco Packet Tracer in your system and login into your account.
2. To create the first network, in the bottom of the left corner, select “End Devices”. Drag 3 PCs and drop them on the screen (PC0, PC1, PC2).
3. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1switch and drop it on the screen between the PCs.
4. Connect each PC with the switch via a cable from the connection section.
5. To configure the IP address of each PC for packet transmission, double-click on a PC, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 10.0.0.2.
6. Repeat step 5 for configuring each PC and type the IP address as 10.0.0.3 and so on.
7. In each configuration, add subnet mask as 255.0.0.0 and default gateway as 10.0.0.1
8. To create the second network, in the bottom of the left corner, select “End Devices”. Drag 2 server and drop them on the screen .
9. Select “Switch 2960-24T logo” from the Network Devices section. Drag 1 switch and drop it on the screen between the servers.
10. Connect each server with the switch via a cable from the connection section.
11. To configure the IP address of each server for packet transmission, double-click on it, and choose desktop. Then click on IP configuration for static IP addressing (IPv4) and type in 192.168.1.2 and 192.168.1.3
12. In each configuration, add subnet mask as 255.255.255.0 and default gateway as 192.168.1.1
13. Select “4331 Router” from the Routers section. Drag two of them and drop them on the screen between the two networks.
14. Connect the two routers using cross wire .
15. To configure one side of the router, click on the router, and add 10.0.0.1 as IP address in Gig0/0/0 (place the cursor and check the port) and turn it on. Add 11.0.0.1 in Gig0/0/1 for connecting to the second router, and turn it on.
16. To configure the other side of the router, click on the router, and add 192.168.1.1 as IP address in Gig0/0/1 (place the cursor and check the port) and turn it on. Add 11.0.0.2 in Gig0/0/1 for connecting to the second router, and turn it on.
17. Dynamic routing protocol- Add 10.0.0.0, 11.0.0.0 and 192.168.1.0 in the RIP section of both the routers.
18. SMTP configuration- On one server (192.168.1.3), open services. In EMAIL, add a domain name along with the required usernames and passwords.
19. On each PC, configure the mail by adding the username, password, email address, incoming and outgoing mail servers (192.168.1.3).
20. Send a mail from one PC to another and reply to verify the configuration.

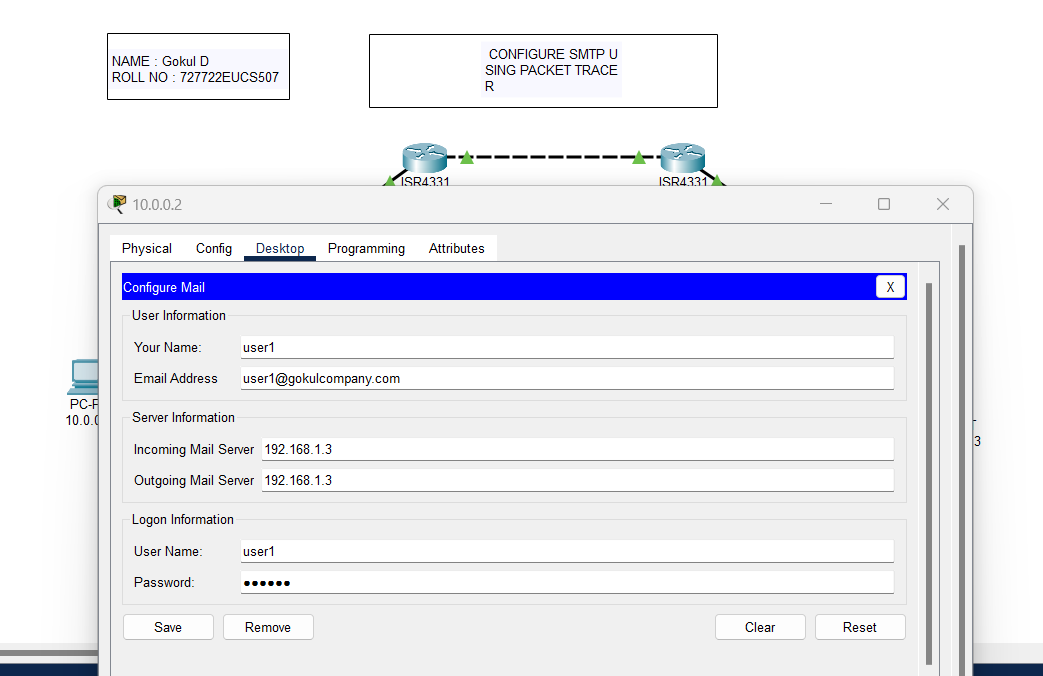
# NETWORK DIAGRAM :

****

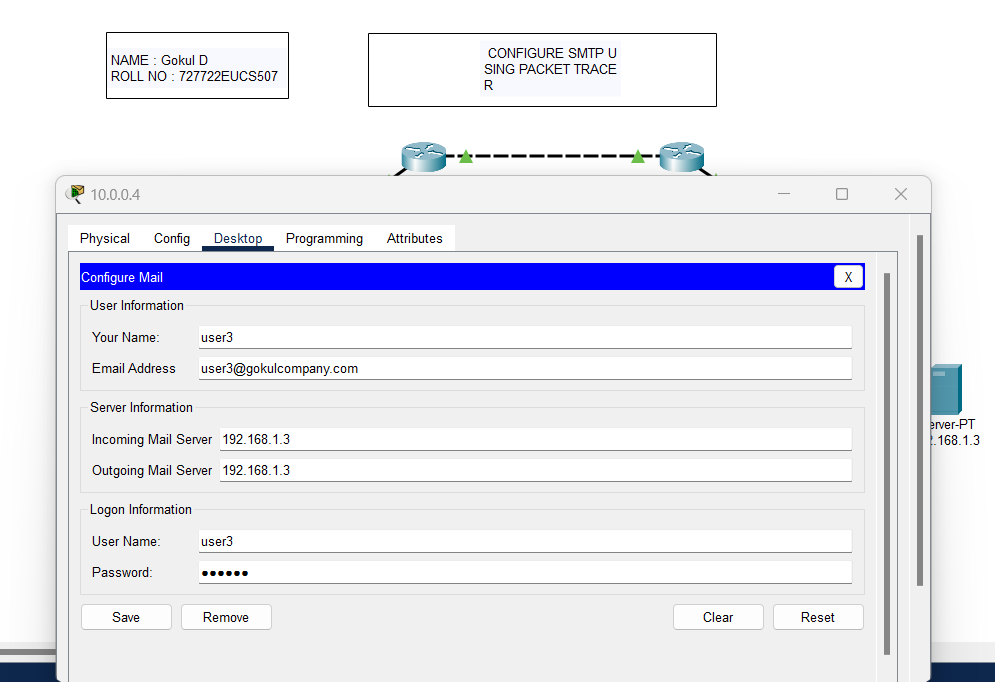
**SMTP configuration on server (192.168.1.3)**

****

# Cofigure mail in PCs : Configure mail in PC0 (10.0.0.2)

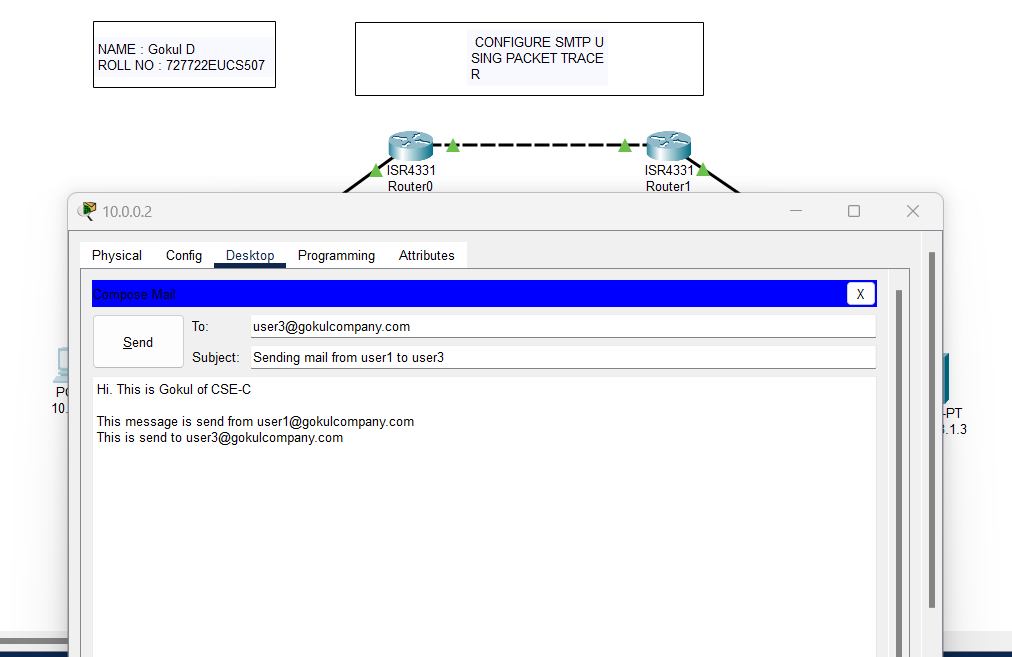
****

**Configure mail in PC2 (10.0.0.4)**

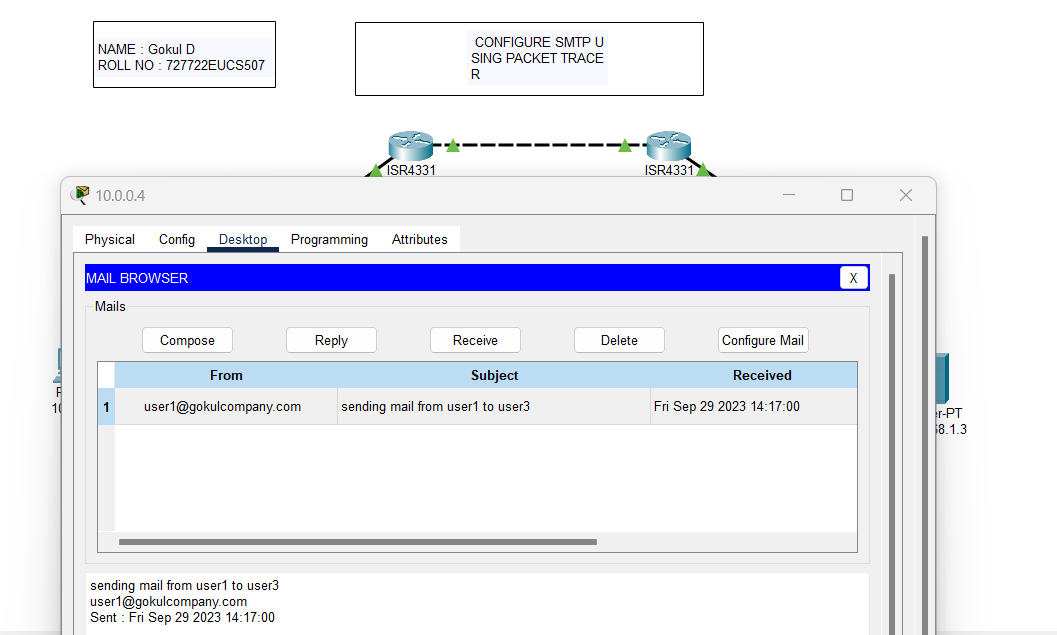
****

# OUTPUT :

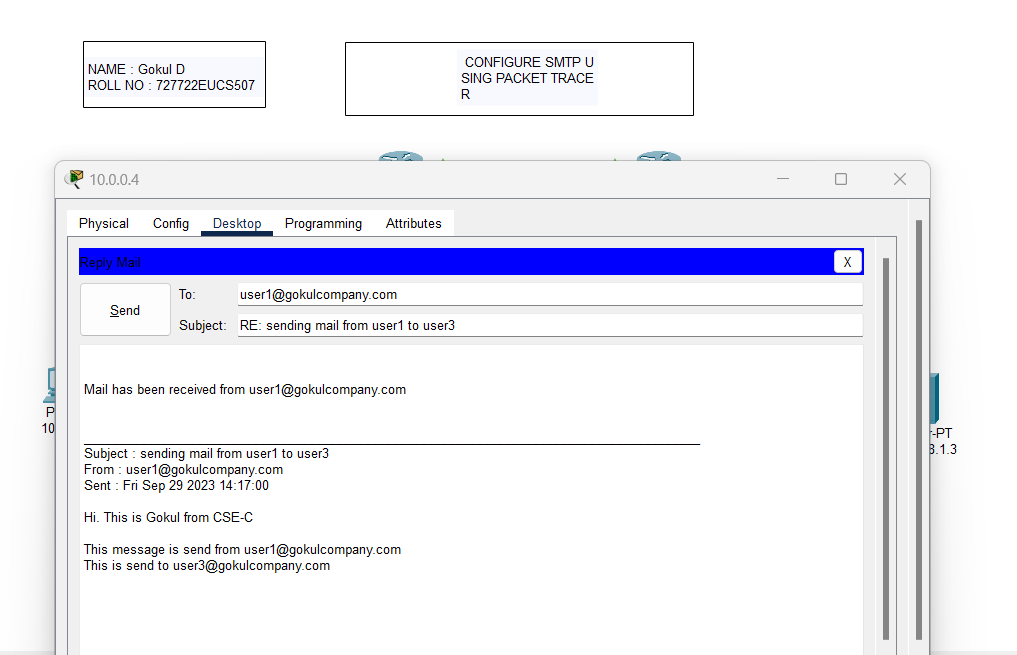
**Mail sent from PC0 (10.0.0.2) to PC2 (10.0.0.4) , and reply from PC2 received at PC0 : Mail sent from PC0 (10.0.0.2)**



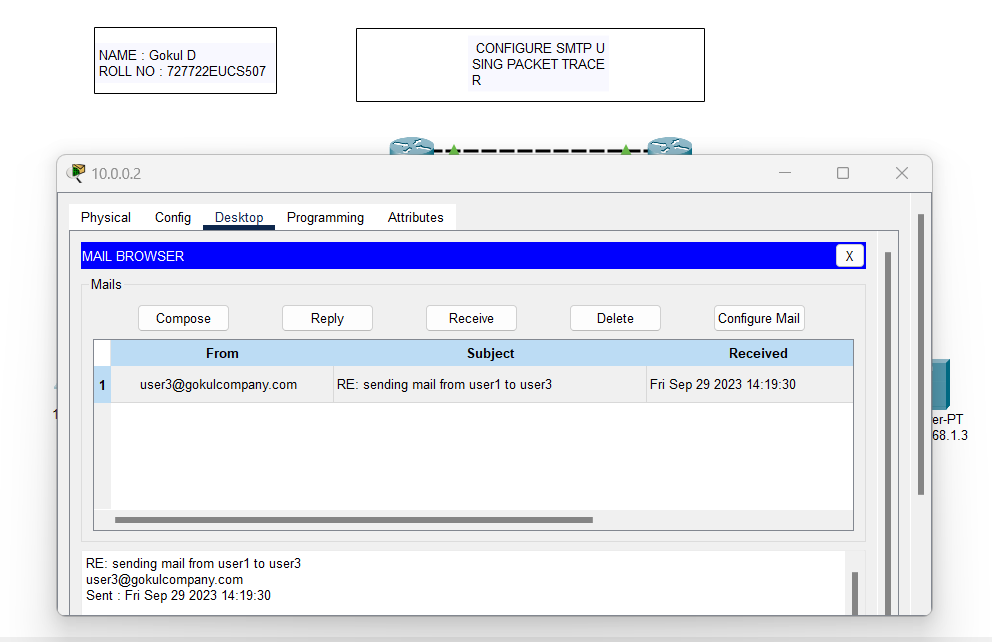
# Mail received at PC2 (10.0.0.4)

****

**Reply sent from PC2 (10.0.0.4)**

****

# Reply received at PC0 (10.0.0.2)

****

**RESULT:**

Hence, SMTP protocol is configured using Cisco Packet Tracer.

|  |  |
| --- | --- |
| EXP NO : 8 | **CONFIGURE SMART HOME NETWORK** |
| DATE : |

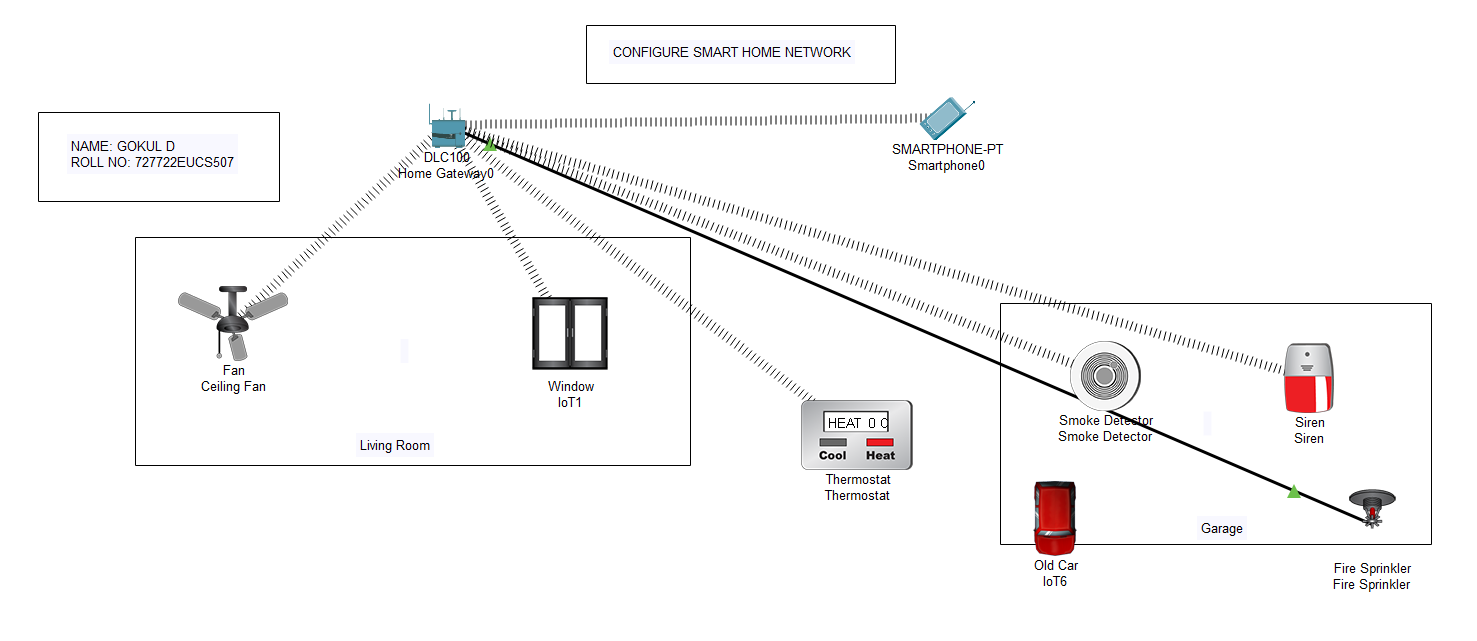
# AIM :

To Design a IoT based Smart Home Network using packet tracer Tool.

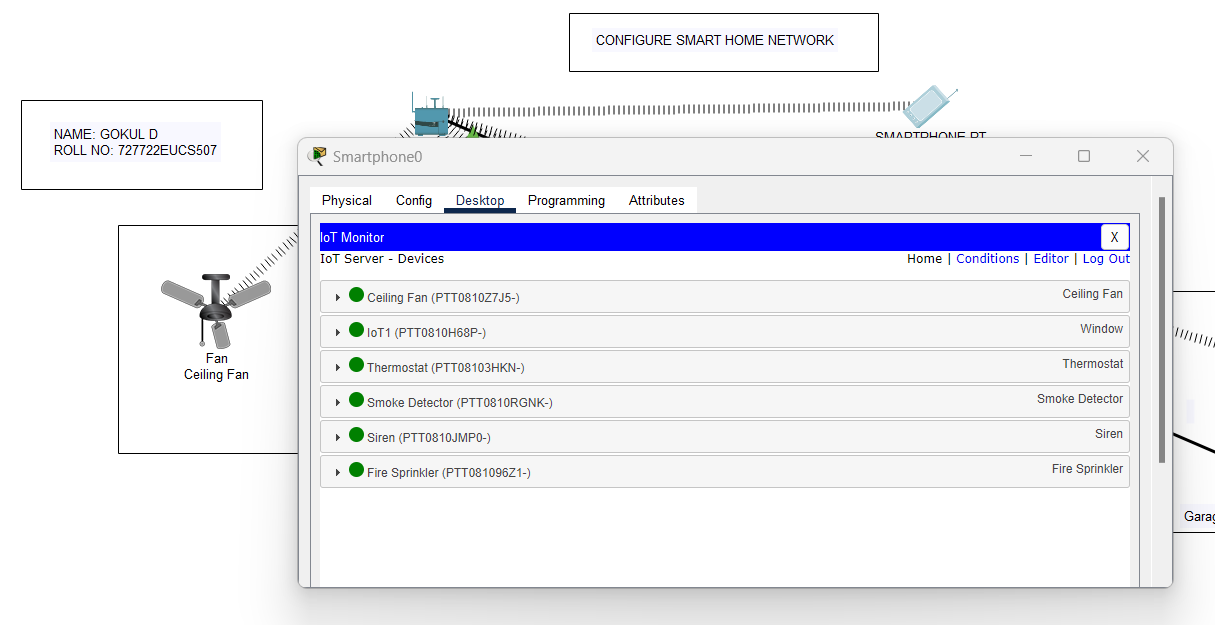
# PROCEDURE:

1. Open Cisco Packet Tracer in your system and login into your account.
2. Create a living room and garage
3. To create a Living room , Select a Ceiling fan and a Window.
4. Now choose the Home Gateway from the end devices.
5. To establish a wireless connection , click on the ceiling fan , and click on the advanced button on the bottom right corner and in network adaptor, choose PT-IOE-NM-1W and follow the same for the Window. Now the wireless connection has been made.
6. Now choose smartphone from the end device and connect it to the homegateway.
7. Copy the IP address of the homegateway and paste it in the IOT Monitor Application in the smart phone and login.This page shows the device connected to the homegateway.
8. To register the fan and the window in the IOT Monitor , Enable the Homegateway option in the IOE Server of both fan and the window.
9. Now these devices are seen on those IOT Monitor.
10. To make these devices automatically work based on the temperature of the environment , Thermostat device is used .
11. Select Thermostat from the end devices and provide the display name as Thermostat. Choose Homegateway in the IOE Server and DHCP in the IP configuration. Now the Thermostat is added in the Smart Phone.
12. In the Smart phone , IOT Monitor , Go to Conditions to add the respective conditions to monitor those devices.
13. Click Add , Name it as Turn the Ceiling fan Low and select any in match and Provide the condition as if Thermostat temperature >= 15 degree celcius , then set the ceiling fan status to low.
14. Again , Click Add , Name it as Turn the Ceiling fan Off and select any in match and Provide the condition as if Thermostat status is cooling , then set the ceiling fan status to Off.
15. Now check the Status of the ceiling fan according to the temperature changes shown in the Thermostat device.
16. In the garage , add the devices Siren , Smoke Detector and Fire sprinkler.
17. Now in all the devices , change the name of the devices and Choose Homegateway in the IOE Server and DHCP in the IP configuration. Now all these devices are added in the Smart Phone.
18. Now an old car is choosen to produce smoke effect.
19. Then the conditions are provided in the in IOT Monitor of the smartphone to monitor the devices.
20. Provide conditions like if the smoke detector detects the smoke level > 10 , then the siren alarm is on. Also the fire sprinkler is turned on.
21. Check those by creating smoke level according to the provided conditions.

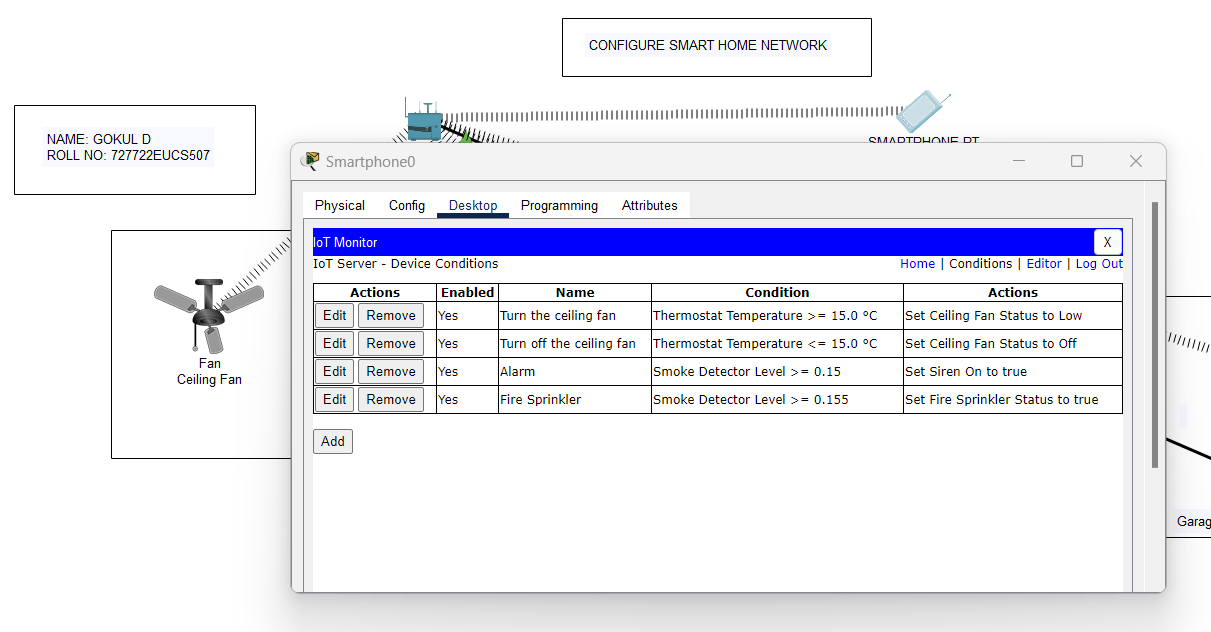
# NETWORK DIAGRAM :

****

**Devices connected**

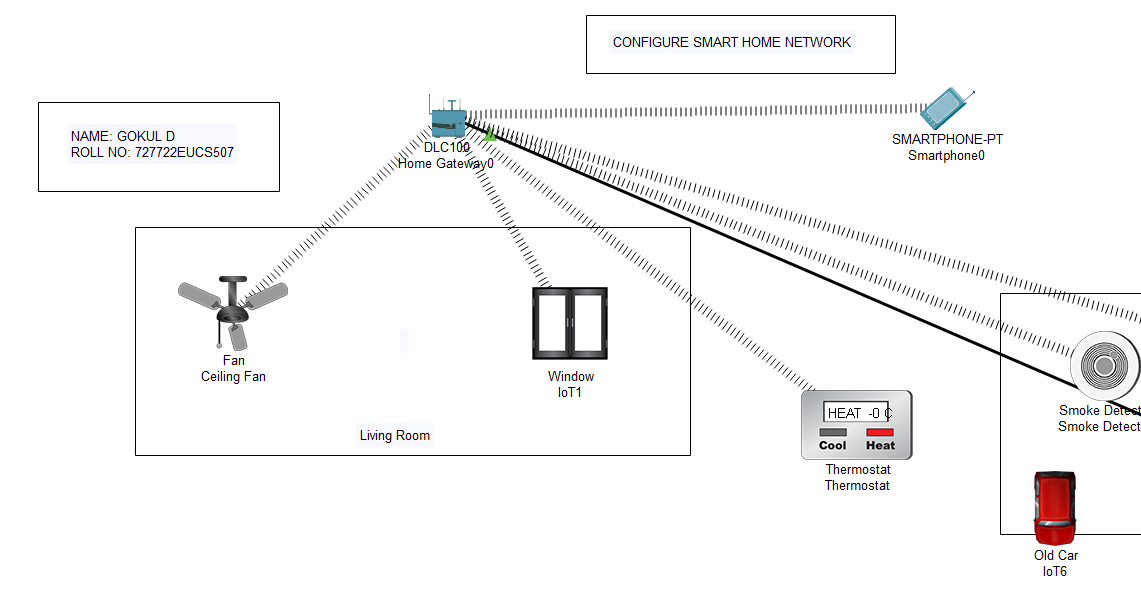
****

# Conditions provided

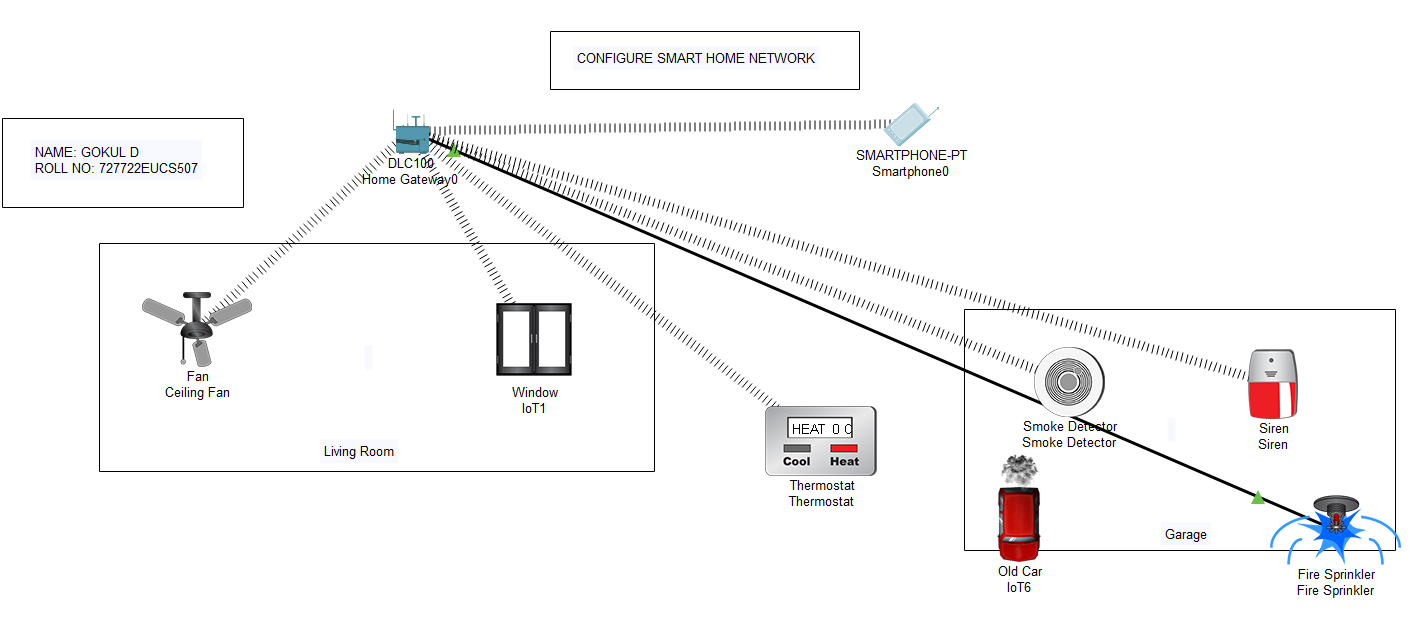
****

**OUTPUT :**

# Ceiling fan Switched to Low as the thermostat status is cooling

****

**Fire Sprinkler and Alarm turned on as the smoke is detected**

****

# RESULT:

Hence the IoT based Smart Home Network using packet tracer Tool has been created successfully.

|  |  |
| --- | --- |
| EXP NO : 9 | **IMPLEMENTATION OF SUBNETTING** |
| DATE : |

# AIM :

To identify the allocation of IP addresses to networks and subnets.

# DESCRIPTION:

**Subnetting:**

Subnetting is the strategy used to partition a single physical network into more than one smaller logical sub-networks (subnets). An IP address includes a network segment and a host segment. Subnets are designed by accepting bits from the IP address's host part and using these bits to assign a number of smaller sub-networks inside the original network. Subnetting allows an organization to add sub-networks without the need to acquire a new network number via the Internet service provider (ISP). Subnetting help

s to reduce the network traffic and conceals network complexity. Subnetting is essential when a single network number has to be allocated over numerous segments of a local area network (LAN).

Subnets were initially designed for solving the shortage of IP addresses over the Internet.

# ALGORITHM:

* 1. Enter the IP address as a string and split the string after every ‘.’
  2. Declare another string to store binary IP address.
  3. Convert the decimal IP to binary IP
  4. Enter the number of addresses in each subnet
  5. Calculate the mask
  6. Calculate the first address and last address
  7. Get first address by ANDing last n bits with 0
  8. Get last address by ORing last n bits with 1

# Description on classes, objects, constructors and methods used:

* split(String regex)-Splits this string around matches of the given regular expression.
* Integer.toBinaryString(int i)- Returns a string representation of the integer argument as anunsigned integer in base 2.
* log(double a)- Returns the natural logarithm (base e) of a double value.
* charAt(int index)- Returns the char value at thespecified index. An index ranges from 0 to length() - 1.
* substring(int beginIndex)- Returns a string that is a substring of this string.

# PROGRAM:

package CN;

import java.util.Scanner; public class Subnet {

static String blueColorCode = "\u001B[34m"; static String greenColorCode = "\u001B[32m"; static String resetColorCode= "\u001B[0m"; static String cyanColorCode= "\u001B[36m";

public static void main(String[] args) {

System.***out***.println(cyanColorCode+"NAME : GOKUL D\nROLL NO : 727722EUCS507\n"+resetColorCode);

Scanner sc = new Scanner(System.***in***); System.***out***.print("Ip address: "); String ip = sc.nextLine();

String split\_ip[] = ip.split("\\."); //SPlit the string after every . String split\_bip[] = new String[4]; //split binary ip

String bip = ""; for(int i=0;i<4;i++){ split\_bip[i] =

*appendZeros*(Integer.*toBinaryString*(Integer.*parseInt*(split\_ip[i]))); bip += split\_bip[i];

}

System.***out***.println("Binary Format "+bip);

System.***out***.print("Enter the number of addresses in each subnet: "); int n = sc.nextInt();

//Calculation of mask

int bits = (int)Math.*ceil*(Math.*log*(n)/Math.*log*(2)); int mask = 32-bits;

System.***out***.println("Subnet mask = "+mask);

//Calculation of first address and last address int fbip[] = new int[32];

for(int i=0; i<32;i++) fbip[i] = (int)bip.charAt(i)-48

for(int i=31;i>31-bits;i--)//Get first address by ANDing last n bits with 0 fbip[i] &= 0;

String fip[] = {"","","",""}; for(int i=0;i<32;i++)

fip[i/8] = new String(fip[i/8]+fbip[i]); System.***out***.print("Network address is = "); for(int i=0;i<4;i++){ System.***out***.print(Integer.*parseInt*(fip[i],2)); if(i!=3) System.***out***.print(".");

}

System.***out***.println();

int lbip[] = new int[32];

for(int i=0; i<32;i++) lbip[i] = (int)bip.charAt(i)-48;

for(int i=31;i>31-bits;i--)//Get last address by ORing last n bits with 1 lbip[i] |= 1;

String lip[] = {"","","",""}; for(int i=0;i<32;i++)

lip[i/8] = new String(lip[i/8]+lbip[i]); System.***out***.print("Broadcast address is = "); for(int i=0;i<4;i++){ System.***out***.print(Integer.*parseInt*(lip[i],2)); if(i!=3) System.***out***.print(".");

}

System.***out***.println();

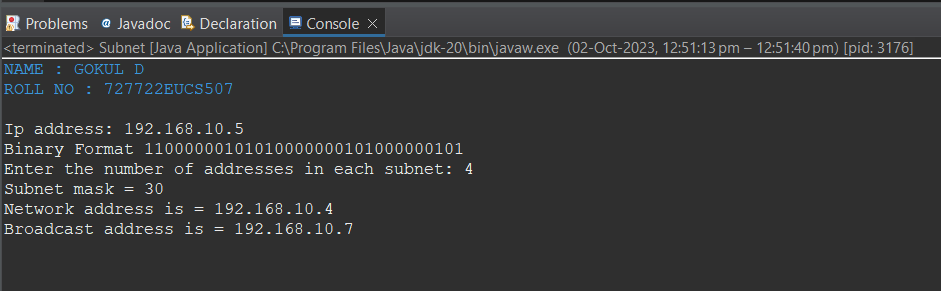
}

static String appendZeros(String s){ String temp = new String("00000000"); return temp.substring(s.length())+ s;

}

}

# OUTPUT :

****

**RESULT:**

Thus the allocation of IP addresses to networks and subnets was identified successfully.

|  |  |
| --- | --- |
| EXP NO : 10 | **APPLICATIONS USING TCP SOCKETS – CONCURRENT SERVER** |
| DATE : |

# AIM :

To execute multiple processes running simultaneously at the server using threads.

# DESCRIPTION:

The need to write concurrent applications introduced threads. In other words, threads are processes that share a single address space. Each thread has its own program counter and stack. Threads are often called lightweight processes. A sequence of executing instructions is called a thread that runs independently of other threads and yet can share data with other threads directly. A thread is contained inside a process. There can exist multiple threads within a process that share resources like memory, while different processes do not share these resources. Concurrency is a property of systems in which several computations are executing simultaneously, and potentially interacting with each other.

# ALGORITHM:

Client:

1. Get localhost ip
2. Establish the connection with server port 5057
3. Obtain input and out streams
4. Loop performs the exchange of information between client and client handler
5. If client sends exit,close this connection then break from the while loop
6. Print date or time as requested by client
7. Close resources Server:
8. Server is listening on port 5057
9. Run infinite loop to get client requests
10. Create socket object to receive incoming client requests
11. Obtain input and out streams
12. Create a new thread object
13. Invoke the start() method
14. Create a ClientHandler class
15. Ask user what he wants
16. Receive answer from the client
17. Create Date Object
18. Write on output stream based on answer from the client
19. Close resources

# Description on packages, classes, objects, constructors and methods: Client:

In Package java.net

-java.net.Socket

\*Implements client sockets (also called just “sockets”).

\*An endpoint for communication between two machines.

\*Constructor and Methods

-Socket(String host, int port): Creates a stream socket and connects it to the specified port number on the named host.

* DataInputStream class- A data input stream lets an application read primitive Java datatypes from an underlying input stream in a machine-independentway.

\*DataOutputStream class-A data output stream lets an application write primitive Java datatypes to an output stream in a portable way.

* writeUTF(String str)- Writes a string to the underlying output stream using modified UTF-8encoding in a machine-independent manner.
* readUTF()-See the general contract of the readUTF method of DataInput.

# Server:

java.net.ServerSocket

\*Implements server sockets.

\*Waits for requests to come in over the network.

\*Performs some operation based on the request.

\*Constructor and Methods

-ServerSocket(int port)

-Socket Accept(): Listens for a connection to be made to this socket and accepts it. This method blocks until a connection is made.

\*format(Date date)- Formats a Date into a date-time string.

* writeUTF(String str)- Writes a string to the underlying output stream using modified UTF- 8encoding in a machine-independent manner.
* readUTF()-See the general contract of the readUTF method of DataInput.

# PROGRAM:

**Client.java**

package CN; import java.io.\*; import java.net.\*;

import java.util.Scanner; public class Client {

static String blueColorCode = "\u001B[34m"; static String greenColorCode = "\u001B[32m"; static String resetColorCode = "\u001B[0m"; static String cyanColorCode = "\u001B[36m";

public static void main(String[] args) throws IOException { System.**out**.println(cyanColorCode + "NAME : GOKUL D\nROLL NO :

727722EUCS507\n" + resetColorCode); try {

Scanner scn = new Scanner(System.**in**);

InetAddress ip = InetAddress.getByName("localhost"); Socket s = new Socket(ip, 5057);

DataInputStream dis = new DataInputStream(s.getInputStream()); DataOutputStream dos = new

DataOutputStream(s.getOutputStream());

while (true) {

System.**out**.println(dis.readUTF()); String tosend = scn.nextLine(); dos.writeUTF(tosend);

// If client sends exit,close this connection

// and then break from the while loop if (tosend.equals("Exit")) {

System.**out**.println("Closing this connection : " + s); s.close();

System.**out**.println("Connection closed"); break;

}

String received = dis.readUTF(); System.**out**.println(received);

}

scn.close();

dis.close();

dos.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

# Server.java

package CN; import java.io.\*; import java.text.\*; import java.util.\*; import java.net.\*;

public class Server {

static String blueColorCode = "\u001B[34m"; static String greenColorCode = "\u001B[32m"; static String resetColorCode = "\u001B[0m"; static String cyanColorCode = "\u001B[36m";

public static void main(String[] args) throws IOException { System.**out**.println(cyanColorCode + "NAME : SUBIKSHA KR\nROLL NO : 727721EUCS154\n" + resetColorCode);

ServerSocket ss = new ServerSocket(5057); while (true) {

Socket s = null; try {

s = ss.accept();

System.**out**.println("A new client is connected : " + s); DataInputStream dis = new

DataInputStream(s.getInputStream());

DataOutputStream dos = new DataOutputStream(s.getOutputStream());

System.**out**.println("Assigning new thread for this client"); Thread t = new ClientHandler(s, dis, dos);

t.start();

} catch (Exception e) { s.close(); e.printStackTrace();

}

}

}

}

//ClientHandler class

class ClientHandler extends Thread {

static String blueColorCode = "\u001B[34m"; static String greenColorCode = "\u001B[32m"; static String resetColorCode = "\u001B[0m"; static String cyanColorCode = "\u001B[36m";

DateFormat fordate = new SimpleDateFormat("yyyy/MM/dd"); DateFormat fortime = new SimpleDateFormat("hh:mm:ss");

final DataInputStream dis; final DataOutputStream dos; final Socket s;

public ClientHandler(Socket s, DataInputStream dis, DataOutputStream dos) { this.s = s;

this.dis = dis; this.dos = dos;

}

@Override

public void run() {

String received; String toreturn; while (true) {

try {

dos.writeUTF("What do you want?[Date | Time]..\n" + "Type Exit to terminate connection.");

received = dis.readUTF();

if (received.equals("Exit")) {

System.**out**.println("Client " + this.s + " sends exit..."); System.**out**.println("Closing this connection."); this.s.close();

System.**out**.println("Connection closed"); break;

}

Date date = new Date();

resetColorCode;

resetColorCode;

switch (received) { case "Date":

toreturn = cyanColorCode + fordate.format(date) +

dos.writeUTF(toreturn); break;

case "Time":

toreturn = cyanColorCode + fortime.format(date) +

dos.writeUTF(toreturn); break;

default:

dos.writeUTF("Invalid input"); break;

}

} catch (IOException e) { e.printStackTrace();

}

try {

}

this.dis.close();

this.dos.close();

} catch (IOException e) { e.printStackTrace();

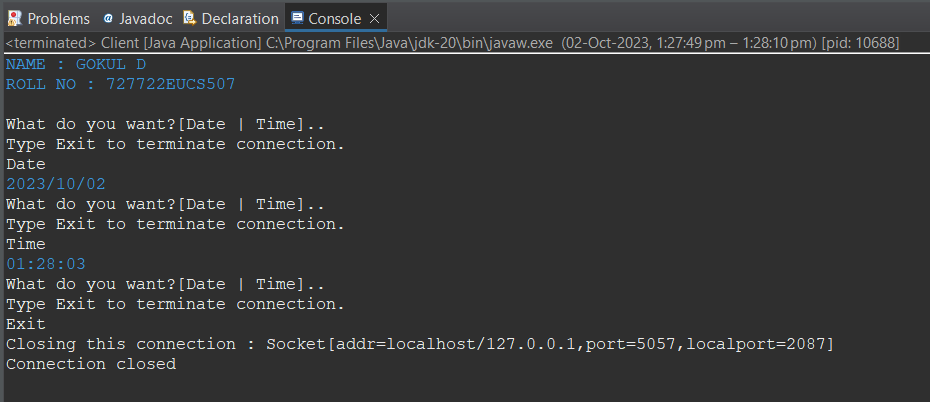
}

}

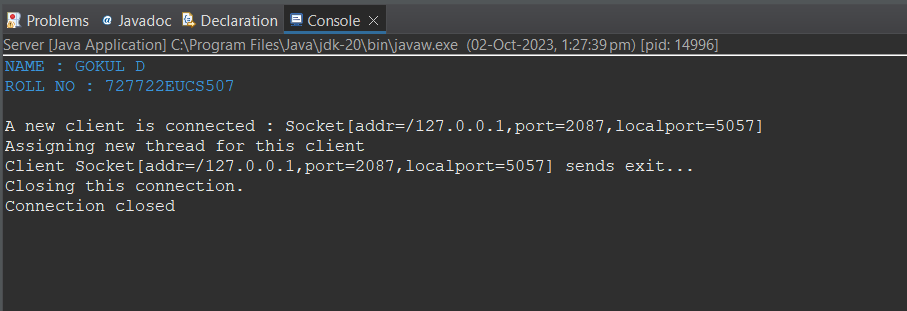
}

# OUTPUT :

**Client.java**

****

# Server.java

****

**RESULT:**

Thus the multiple processes running simultaneously at the server using threads was executed successfully.

|  |  |
| --- | --- |
| EXP NO : 11 | **STUDY OF NETWORK SIMULATOR 3 – NS3** |
| DATE : |

# AIM :

To do a study of Network Simulator 3 and analyse the output using NetAnim.

# DESCRIPTION:

**NS3:**

NS-3 is a discrete-event network simulator, targeted primarily for research and

educational use. NS-3 is free software, licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.

The goal of the ns-3 project is to develop a preferred, open simulation environment for networking research: it should be aligned with the simulation needs of modern networking research and should encourage community contribution, peer review, and validation of the software.

# Simulation Models

The NS-3 project is committed to building a solid simulation core that is well documented, easy to use and debug, and that caters to the needs of the entire simulation workflow, from simulation configuration to trace collection and analysis.

The NS-3 simulation core supports research on both IP and non-IP based networks. However, the large majority of its users focuses on wireless/IP simulations which involve models for Wi-Fi, WiMAX, or LTE for layers 1 and 2 and a variety of static or dynamic routing protocols such as OLSR and AODV for IP-based applications.

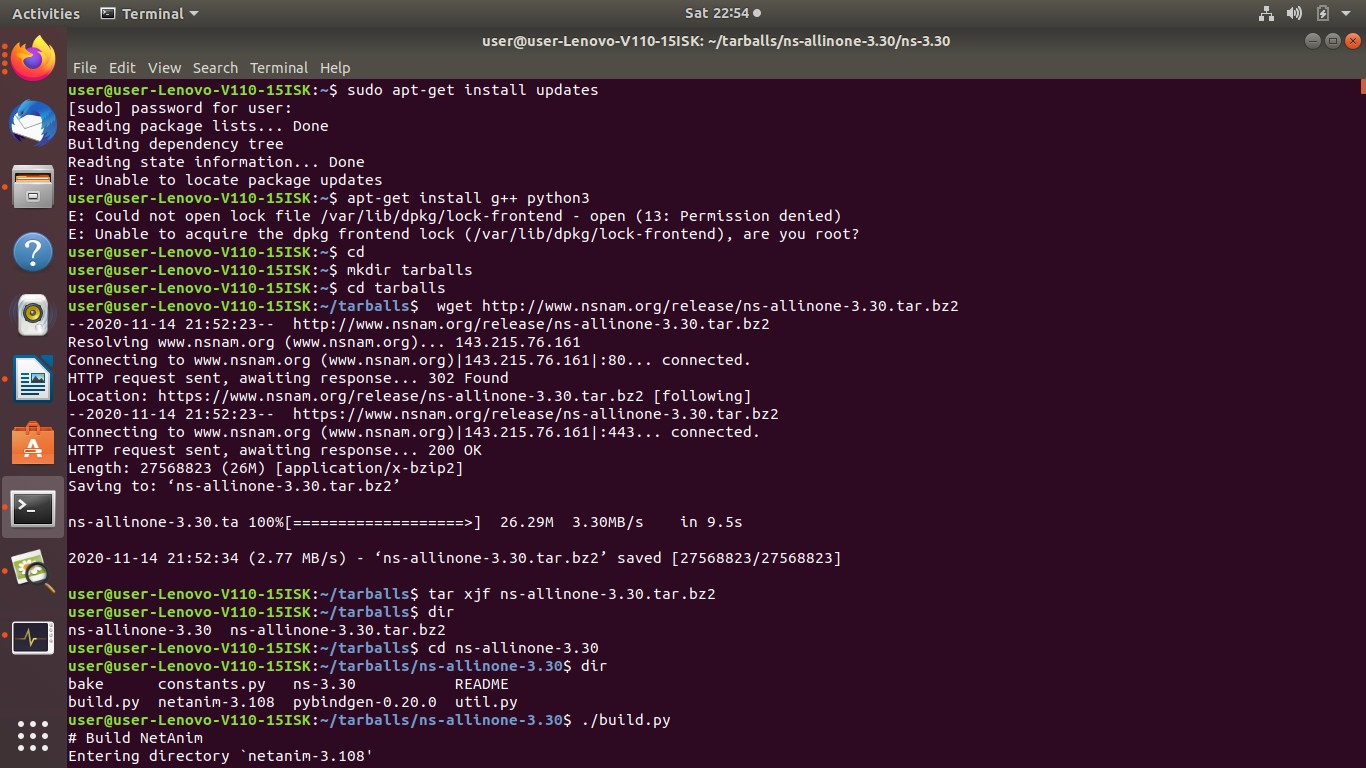
The NS-3 also supports a real-time scheduler that facilitates a number of “simulation- in-the-loop” use cases for interacting with real systems. For instance, users can emit and receive ns-3-generated packets on real network devices, and NS-3 can serve as an interconnection framework to add link effects between virtual machines.

Another emphasis of the simulator is on the reuse of real application and kernel code. Frameworks for running unmodified applications or the entire Linux kernel networking stack within NS-3 are presently being tested and evaluated.

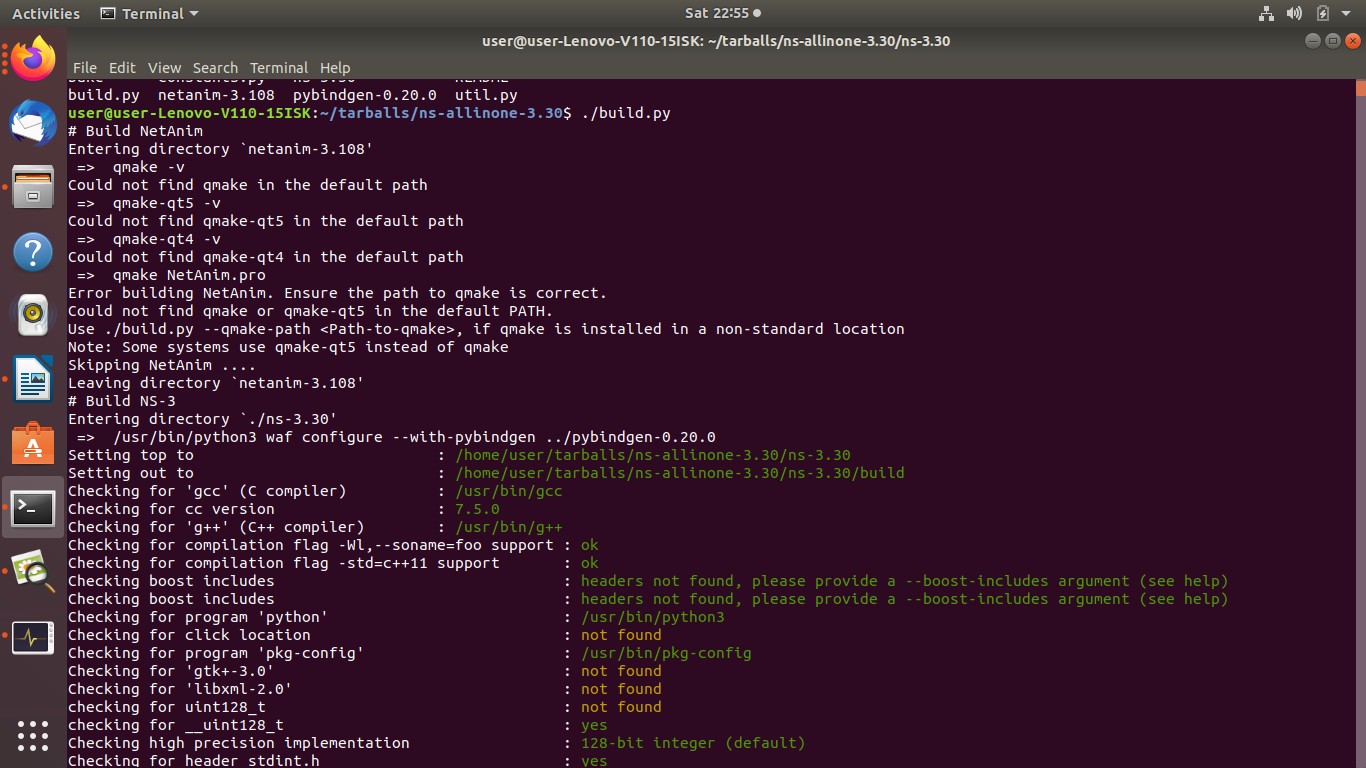
# ANALYSIS:

The analysis is done through the following steps.

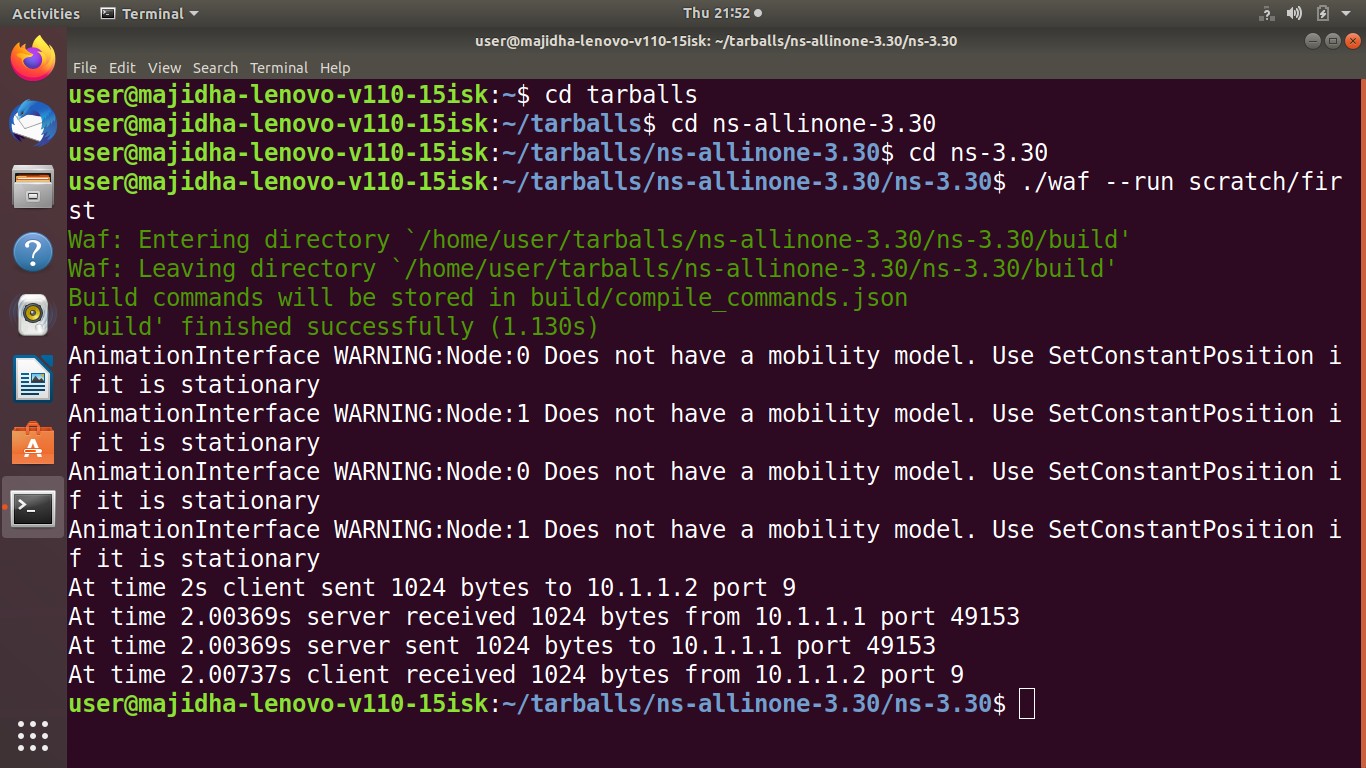
1. Download ns-allinone-3.30 tar file from url “[http://www.nsnam.org](http://www.nsnam.org/)” and extract the file.



1. Install the network simulator by the command “./build.py”

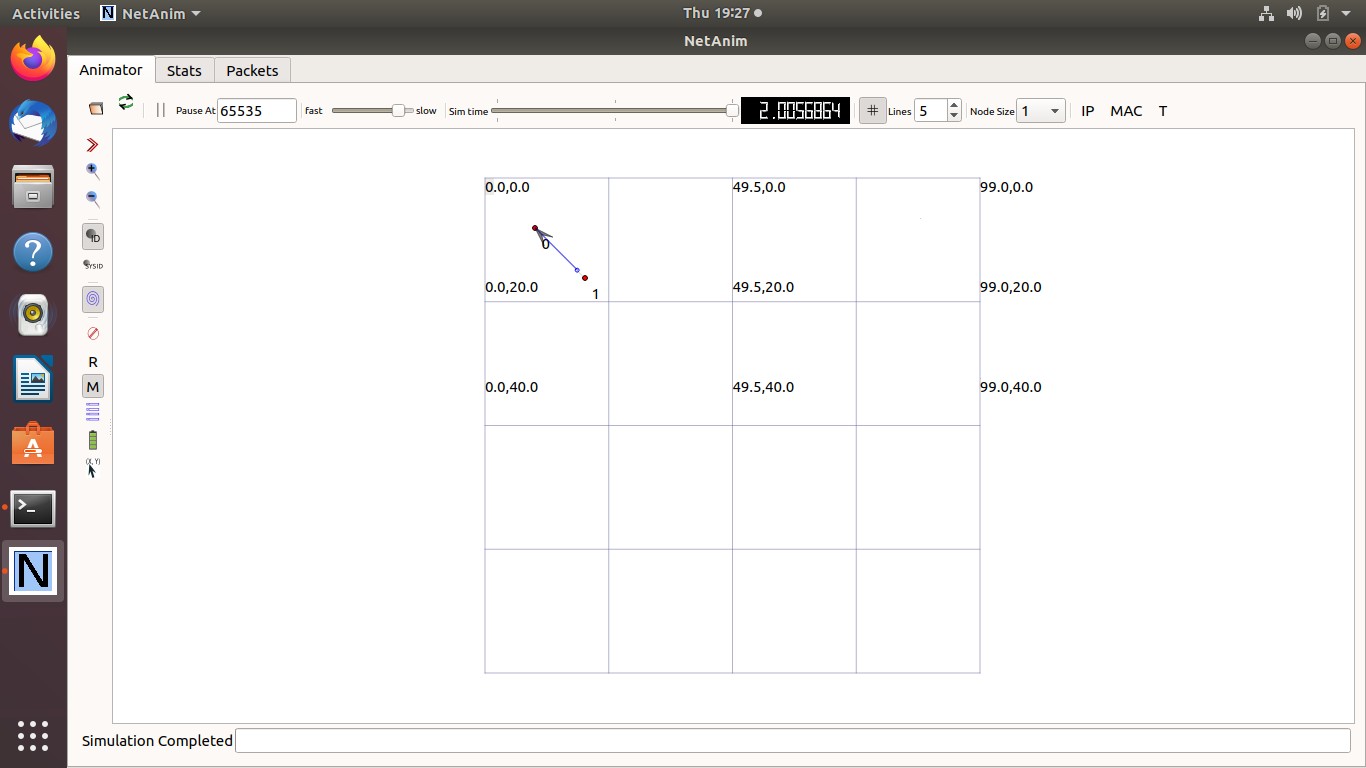


1. Configure using waf and execute the program “first.cc” by the command ./waf --run scratch/first

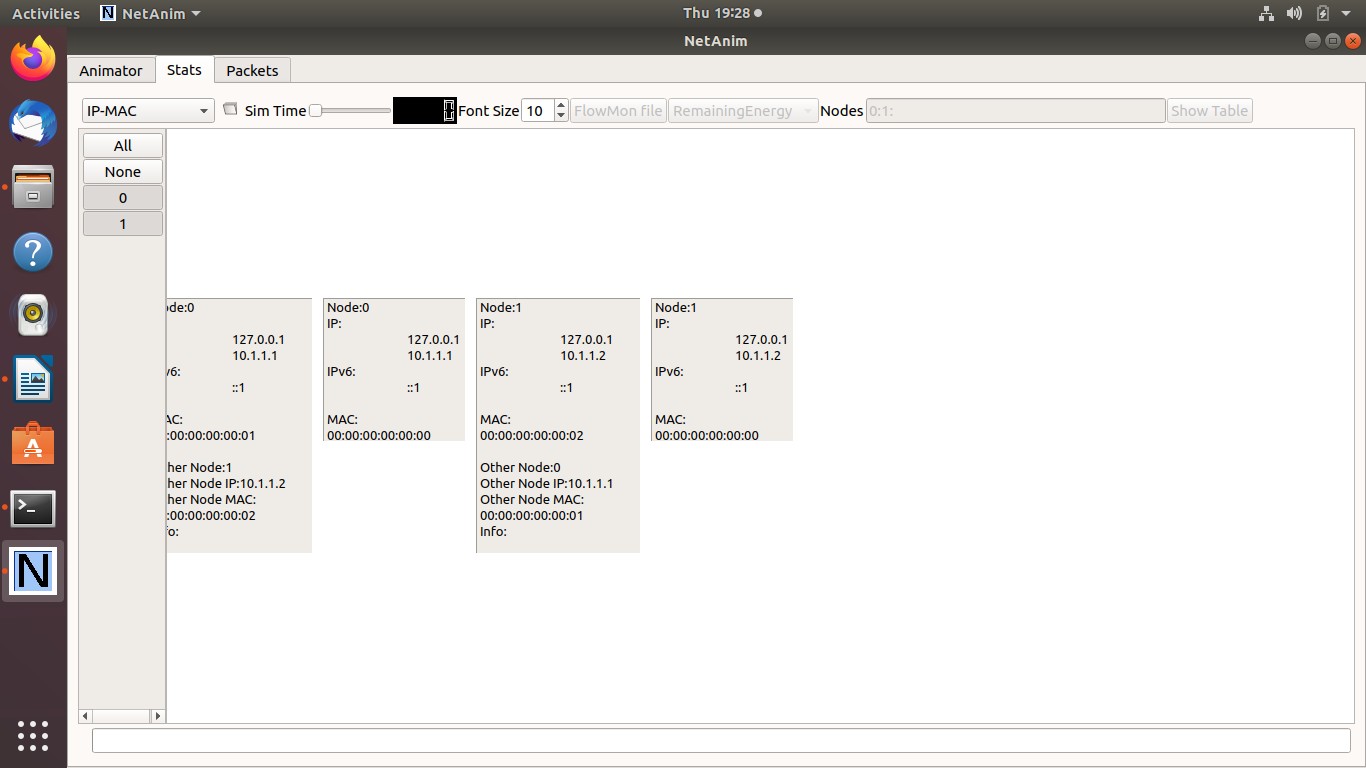


# OUTPUT:

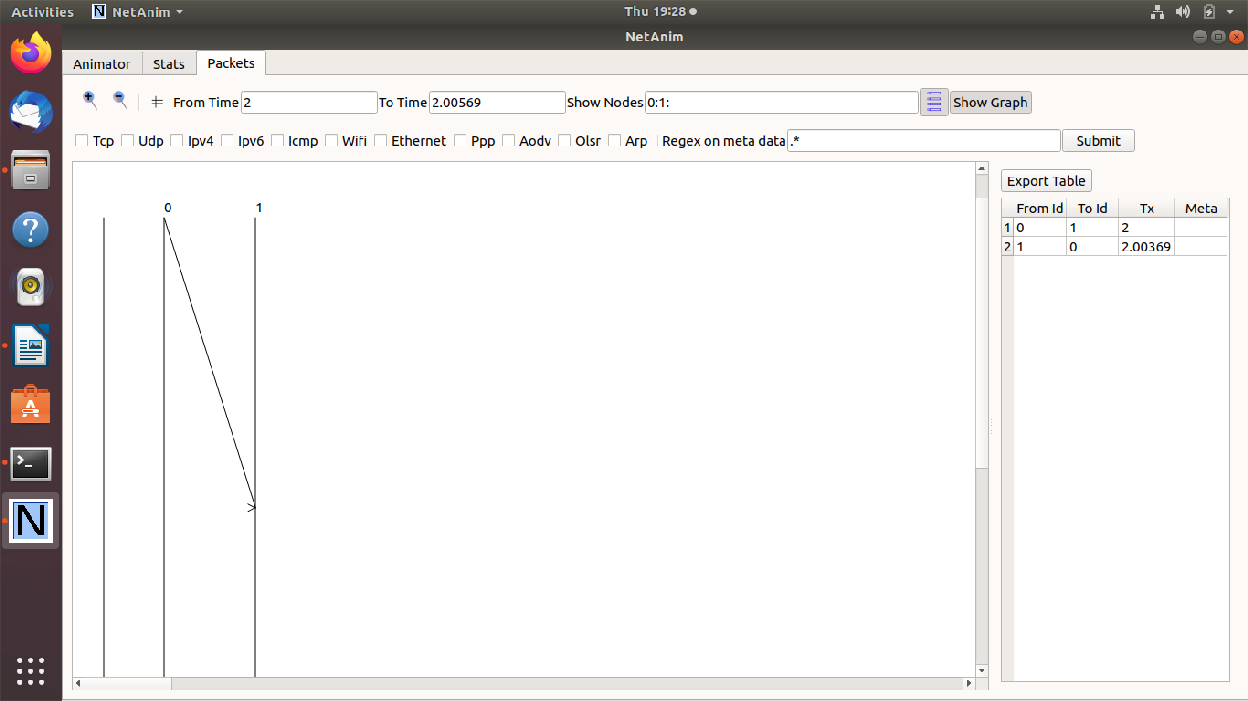
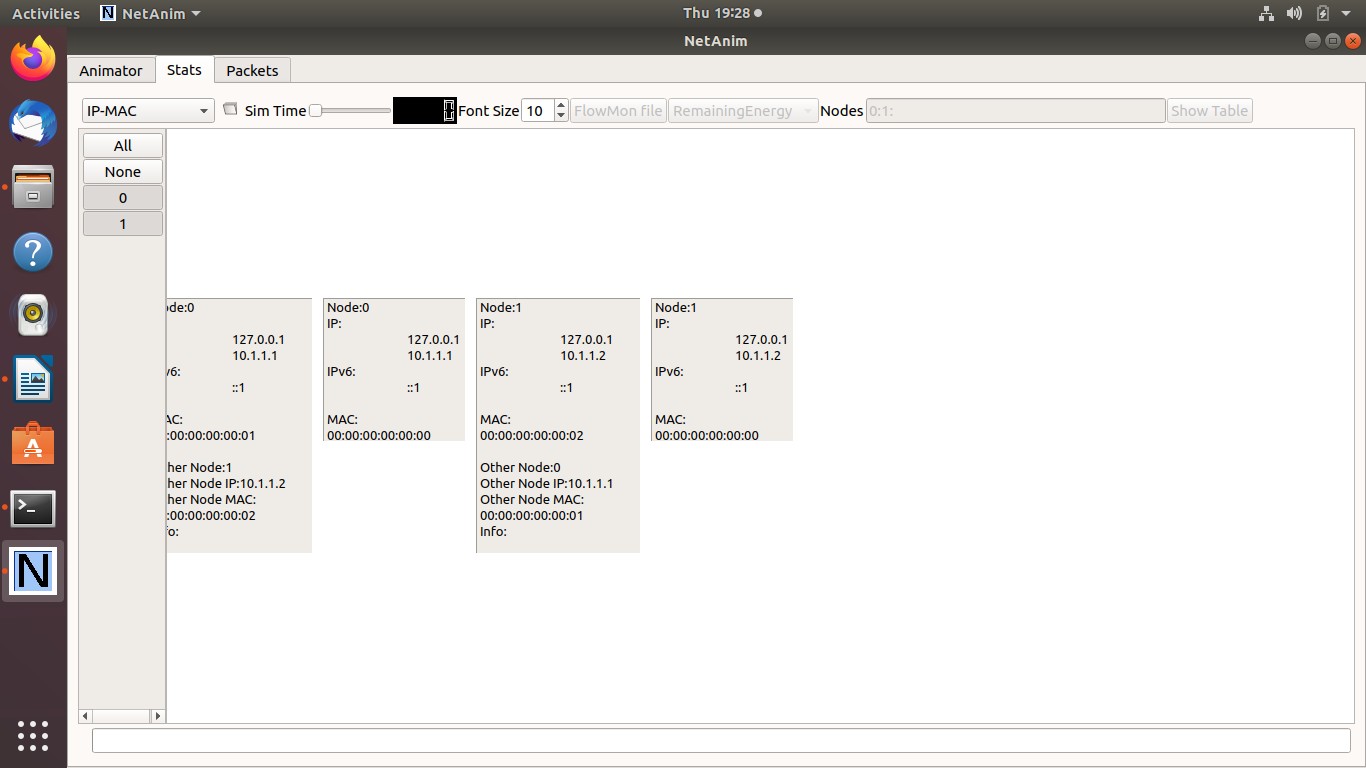
The animator portrays the topology of the peer-to-peer nodes created by the file “first.xml”.



The Statics with IP and MAC addresses of the nodes are shown:



The graph displays the network traffic flow between the nodes



# RESULT:

Thus the study of Network Simulator is performed and a network topology is created and analysed.