SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2628001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

Department of Computer Science and Engineering

Course Code : 22CS503

COMPUTER NETWORKS LABORATORY

Continuous Assessments Record

Submitted by

Name	
Register No	
Year & Semester	
Department	**Computer Science and Engineering**
Section	
Academic Year	**2024-2025 (ODD Semester)**

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2628001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

Department of Computer Science and Engineering

22CS503 – COMPUTER NETWORKS LABORATORY

Submitted by

Register No.:

Name :

Degree : **Branch :**

BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record work by

Mr./Ms……………………………………………………………..……………………

Register No:……………………………………………………… during the

academic year 2024 – 2025.

Faculty In-charge

Submitted for the Autonomous Practical Examination held on

…………….........

INTERNAL EXAMINER **EXTERNAL EXAMINER**

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY
An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2628001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

Department of Computer Science and Engineering

VISION OF THE INSTITUTE

To Produce Globally Competitive Engineers with High Ethical Values and Social Responsibilities.

MISSION OF THE INSTITUTE

- To impart highest quality state-of-the-art technical education by providing impetus to innovation, Research and Development and empowering students with Entrepreneurship skills.
- To instill ethical values, imbibe a sense of social responsibility and strive for societal wellbeing.
- To identify needs of society and offer sustainable solutions through outreach programs.

VISION OF THE DEPARTMENT

To prepare professionals with high technical, research and entrepreneurial skills as well as ethical values who will contribute to the computational world.

MISSION OF THE DEPARTMENT

- Develop human resources with the ability and attitude to adapt to emerging technological changes through academic and research oriented events
- Identify current socio, economic problems of national and international significance and provide solutions through competency centers
- Impart ethics, social responsibilities and necessary professional, entrepreneurial and leadership skills through student lead activities

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

The following Programme Educational Objectives are designed based on the department mission

PEO 1	Be successful in their career in industries associated with Computer Science and Engineering
PEO 2	Comprehend, analyze, design, and create novel products and solutions for the real-life problems
PEO 3	Possess professional and ethical attitude, effective communication skills, team working skills, multi-disciplinary approach, and an ability to relate engineering issues to broader social context.
PEO 4	Exhibit leadership qualities and progress through life-long learning

PROGRAMME SPECIFIC OUTCOMES (PSOs)

On successful completion of Bachelor of Engineering in Electronics and Communication Engineering Programme from Sri Krishna College of Engineering and Technology, the graduate will demonstrate:

PSO 1: Apply the fundamental knowledge for problem solving and analysis as well as conduct investigations in computer science and engineering for sustainable development.

PSO 2: Design and develop the solutions for real time problems and implement them by using modern software tools in lieu of deploying them in the society for its growth.

PSO 3: Communicate effectively, adopt ethics and engage in life-long learning.

PROGRAMME OUTCOMES (POs)

At the time of their graduation students of Electronics and Communication Engineering Programme should be in possession of the following Programme Outcomes

PO**1.Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO**2. Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO**3. Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO**4. Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO**5. Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO**6. The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO**7**. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO**8. Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO**9**. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO**10**. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO**11**. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

SYLLABUS

Ex. No	Description	Course Outcome	Level of Bloom's Taxonomy
1.	Implement and study the performance of CDMA on NS2/NS3 (Using stack called Call net) or equivalent environment.	C503.1	[AN]
2.	Implement Dynamic routing with RIP and OSPF.	C503.1	[AN]
3.	Build simple LANs, perform basic configurations for routers and switches, and implement IPv4 and IPv6 addressing schemes.	C503.2	[AN]
4.	Setup an network with IP address.	C503.1	[AN]
5.	Write a program on a datagram socket for the client/server to display the messages on the client side typed at the server-side	C503.3	[AN]
6.	Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion using packet tracer tool.	C503.5	[AN]
7.	Configure routers, switches and end devices to provide access to local and remote network resources and to enable end-to-end connectivity between remote devices.	C503.5	[AN]
8.	Build two virtual local area networks (VLAN) and communicate them.	C503.4	[AN]
9.	Configuration of DHCP, DNS and Web Server.	C503.4	[AN]
10.	Implement a home or small business network using wireless technology, then connect it to the Internet.	C503.4	[AN]

1. **Text book and Reference book**:

Text book:

1. Behrouz A. Forouzan, Data communication and Networking, 5th Edition, Tata McGraw- Hill, 2017.
2. A S Tanenbaum, DJ Wetherall, Computer Networks, 5th Edition, Prentice-Hall, 2022.

Reference Books:

1. Peterson & Davie, Computer Networks, A Systems Approach, 3rd Edition, Harcourt, 2013.

2. William Stallings, Data and Computer Communications, 9th Edition, PHI, 2006, Bertsekas and Gallagher Data Networks, PHI, 2011.

Web References:

1. https://www.coursera.org/learn/computer-networking
2. https://archive.nptel.ac.in/courses/106/105/106105183/

2. Expected outcome of the course:

Upon successful completion of this course, the student will be able to:

C503.1	Infer the fundamental concepts in networking and system administration.	[AP]
C503.2	Test the different networking protocols and their flow	[AP]
C503.2	Design a wireless network for real time applications.	[AP]
C503.2	Construct the network with cloud connectivity.	[AP]
C503.2	Analyze the various simulators used for network design.	[A]

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2628001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

Department of Computer Science and Engineering

22CS503 – COMPUTER NETWORKS LABORATORY (ODD SEMESTER: 2024-2025)

Components	EX1	EX2	EX3	EX4	EX5	EX6	EX7	EX8	EX9	EX10	EX11	EX12
DATE												
Objective & Components Required (10)												
Program / Connection Diagram (20)												
Simulation & Execution (40)												
Result (20)												
Documentation & Viva (10)												
Consolidated Mark (100)												
Faculty Signature												
Average (100)										Signature		

TABLE OF CONTENTS

Faculty Signature

Continuous Evaluation Rubrics Sheet

22CS503 – COMPUTER NETWORKS LABORTORY

Items	Excellent	Good	Satisfactory	Needs Improvement
Objective & Components Required (10)	**10 Marks**	**(9-7) Marks**	**(6-4) Marks**	**(3-1)Marks**
	The aim and purpose of the experiment are clearly defined. Preliminary questions are to be answered. The equipment's required are to be clearly listed with specifications.	Able to define the aim and purpose of the experiment but not clearly. Preliminary questions are answered but not in detail. The equipment's required are clearly listed but with inappropriate specifications.	The aim and purpose of the experiment are defined but not clear. Few Preliminary questions are unanswered. The equipment's required are clearly listed but without specifications.	Unable to explain the aim and purpose of the experiment. Preliminary questions are unanswered. The equipment's required are not clearly listed.
Program / Connection Diagram (20)	**20Marks**	**(19-17) Marks**	**(16-14) Marks**	**(13-11) Marks**
	Able to draw Connection diagram neatly with no errors. Formulas required are written and used fully in computation. Model graphs are provided with necessary information.	Connection diagram of the experiment is provided with minor mistakes. Formulas required are written. Model graphs are provided with few missing data.	Connection diagram not drawn legibly and with fewer specifications of the components involved Some mistakes in formula. Model graph is not clear.	Connection diagram of the experimental setup are given wrongly. No model graph provided.
Simulation & Execution (40)	**40Marks**	**(39-37) Marks**	**(36-34) Marks**	**(33-31) Marks**
	The components are identified and connections done without error. Experimentation is as required. Readings taken are justified and tabulated.	The components are identified and connections done without error. The method adopted is relevant but the measurements made are partial. Readings are tabulated.	The components are identified but the connection done with few mistakes. The measured values are deviated.	Poorly experimentation and the method adopted is not relevant to the stated objective.
Result (20)	**20Marks**	**(19-17) Marks**	**(16-14) Marks**	**(13-11) Marks**
	Readings/measurements are utilized to draw necessary charts/graphs. The results are interpreted and compared with	Almost all of the results have been correctly recorded and summarized; only minor improvements are	Some mistakes in tables and graph. Conclusions drawn from the results are not clear.	Experimental measurements are incorrect and wrongly interpreted. Not able to take a

	desired values successfully.	needed in post lab discussion.		measurements and proceed further.
Documentation & Viva (10)	**10 Marks**	**(9-7) Marks**	**(6-4) Marks**	**(3-1) Marks**
	The aim, procedure, circuit diagram, experiment details are well documented.	Able to present the aim, procedure, circuit diagram and experiment details to some extent.	Not able to draw and tabulate the content and organize properly.	Incomplete work and Poor writing presentation.

Signature of Evaluator

Experiments based CISCO PACKET TRACER

Experiment No.	: 1
Title of Experiment	: Implement and study the performance of CDMA on NS2/NS3.
Date of Experiment	:

Aim:

To implement Code Division Multiple Access (CDMA) in NS2/NS3 and evaluate its performance based on parameters such as throughput, delay, and packet loss under varying network conditions.

Software Required:

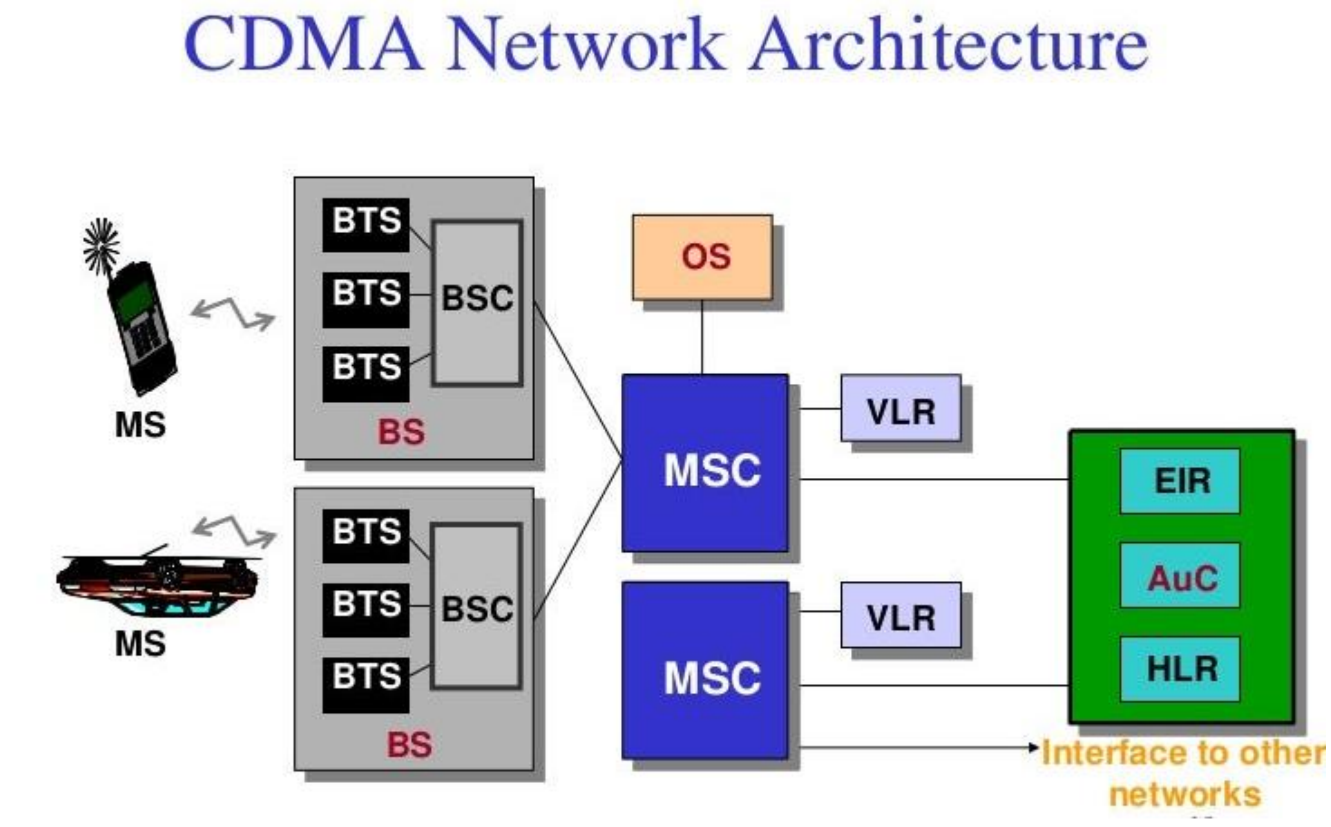
1. **NS3 (Network Simulator 3):**
   - **Version:** Latest stable version.
   - **Download Link:** NS3 Official Site.
   - **Dependencies:** Python, gcc, g++, CMake.
2. **NS2 (Optional):**
   - **Version:** Latest stable version.
   - **Download Link:** NS2 Official Site.
   - **Dependencies:** Tcl, OTcl, Nam.
3. **GNUPlot or Matplotlib:**
   - For plotting and visualizing simulation results.
4. **Wireshark:**
   - For packet-level analysis and detailed examination of network traffic.
5. **Linux OS (e.g., Ubuntu):**
   - A Linux environment is generally recommended for running NS2/NS3 simulations.

Theory:

Code Division Multiple Access (CDMA) is a multiple access technique used in wireless communication, where multiple users share the same frequency spectrum simultaneously. Unlike other access methods like Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA), CDMA assigns a unique code to each user, allowing their signals to coexist without interference. This is achieved through the process of spreading and de-spreading, where each user’s data is multiplied by a high-frequency code unique to them. At the receiver, the original data is extracted by correlating the received signal with the same code. CDMA's robustness against interference and its ability to accommodate many users make it particularly useful in cellular networks, where bandwidth efficiency is crucial.

In a simulated environment like NS2/NS3, CDMA is typically implemented by customizing the physical (PHY) and medium access control (MAC) layers to handle code assignment and signal spreading. Performance analysis involves studying parameters like throughput, delay, and packet loss under varying conditions, such as different user counts or traffic loads. The key challenge in CDMA is managing interference, particularly as the number of users increases, which can lead to code collisions and degraded performance. Properly designed, CDMA enables efficient use of available bandwidth and supports a large number of simultaneous communications.

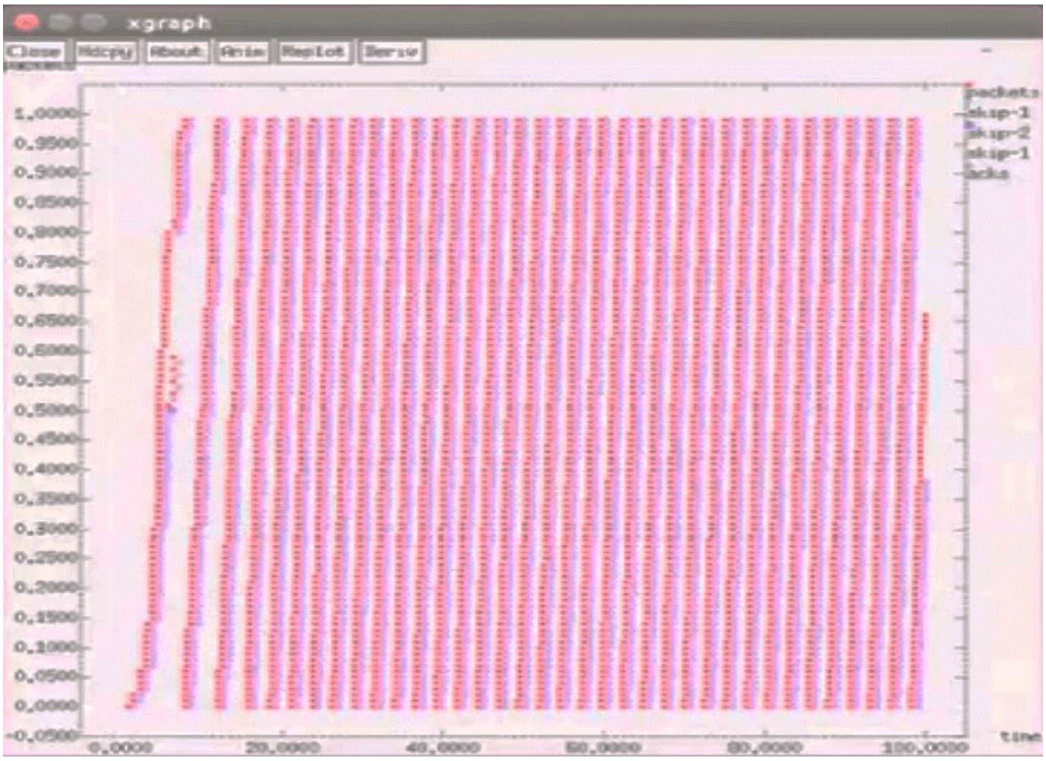
Connection Diagram / Program



```
#set  parameters
  set stop 100    ;# Stop time.

  # Topology
  set type umts ;#type of link

  # AQM parameters
  set minth 30
  set maxth 0
  set adaptive 1 ;# 1 for Adaptive RED, 0 for plain RED

    # Traffic generation.
  set flows 0 ;# number of long-lived TCP flows
  set window 30 ;# window for long-lived traffic

  # Plotting statics.
  set opt(wrap) 100 ;# wrap plots?
  set opt(srcTrace) is ;# where to plot traffic
  set opt(dstTrace) bs2 ;# where to plot traffic

  #default downlink bandwidth in bps
  set bwDL(umts) 384000

    #default downlink propagation delay in seconds
  set propDL(umts) .150
  set ns [new Simulator]
  set tf [open Mlab6.tr w]
  $ns trace-all $tf

  set nodes(is) [$ns node]
  set nodes(ms) [$ns node]
  set nodes(bs1) [$ns node]
  set nodes(bs2) [$ns node]
  set nodes(lp) [$ns node]

  proc cell_topo {} {
  global ns nodes
  $ns duplex-link $nodes(lp) $nodes(bs1) 3Mbps 10ms DropTail
  $ns duplex-link $nodes(bs1) $nodes(ms) 1 1 RED
  $ns duplex-link $nodes(ms) $nodes(bs2) 1 1 RED
  $ns duplex-link $nodes(bs2) $nodes(is) 3Mbps 50ms DropTail
  puts " umts Cell Topology"
  }

  proc set_link_para {t} {
  global ns nodes bwDL propDL
  $ns bandwidth $nodes(bs1) $nodes(ms) $bwDL($t) duplex
  $ns bandwidth $nodes(bs2) $nodes(ms) $bwDL($t) duplex

  $ns delay $nodes(bs1) $nodes(ms) $propDL($t) duplex
  $ns delay $nodes(bs2) $nodes(ms) $propDL($t) duplex

  $ns queue-limit $nodes(bs1) $nodes(ms) 20
  $ns queue-limit $nodes(bs2) $nodes(ms) 20
  }

  # RED and TCP parameters
  Queue/RED set adaptive_ $adaptive
  Queue/RED set thresh_ $minth
  Queue/RED set maxthresh_ $maxth
  Agent/TCP set window_ $window

  #Create topology
  switch $type {
  umts {cell_topo}
  }

  set_link_para $type
  $ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
  $ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]

  # Set up forward TCP connection
  if {$flows == 0} {
  set tcp1 [$ns create-connection TCP/Sack1 $nodes(is) TCPSink/Sack1 $nodes(lp) 0]
  set ftp1 [[set tcp1] attach-app FTP]
  $ns at 0.8 "[set ftp1] start"
  }
```

```
  proc stop {} {
  global nodes opt tf
  set wrap $opt(wrap)
  set sid [$nodes($opt(srcTrace)) id]
  set did [$nodes($opt(dstTrace)) id]

  set a "Mlab6.tr"

  set GETRC "/var/cn/ns-allinone-2.35/ns-2.35/bin/getrc"
  set RAW2XG "/var/cn/ns-allinone-2.35/ns-2.35/bin/raw2xg"

  exec $GETRC -s $sid -d $did -f 0 Mlab6.tr | \
  $RAW2XG -s 0.01 -m $wrap -r > plot6.xgr

  exec $GETRC -s $did -d $sid -f 0 Mlab6.tr | \
  $RAW2XG -a -s 0.01 -m $wrap >> plot6.xgr

  exec xgraph -x time -y packets plot6.xgr &
  exit 0
  }
  $ns at $stop "stop"
$ns run
```

OUTPUT



Procedure:

Procedure:

1. Setup Simulation Environment:

- **Install NS3:**

    - Download and install NS3 following the instructions provided on the official website.
    - Verify the installation by running a sample script provided in the NS3 examples directory.
- **Install Supporting Tools:**
    - Install GNUPlot for graph plotting using the command sudo apt-get install gnuplot.
    - Install Wireshark for packet analysis using the command sudo apt-get install wireshark.

2. CDMA Implementation:

- **Create or Modify PHY and MAC Layers:**
    - Access the NS3 source code related to the physical and MAC layers.
    - Implement CDMA by assigning unique spreading codes to different nodes.
    - Modify the PHY layer to include the spreading and de-spreading functionality.
- **Configure Network Topology:**
    - Design a network topology with multiple nodes (e.g., 5-10 nodes).
    - Assign each node a unique CDMA code for communication.
    - Set up traffic sources (e.g., Constant Bit Rate (CBR), FTP) between nodes.
- **Simulation Script:**
    - Write or modify a simulation script in NS3 to implement the CDMA functionality.
    - Include parameters like simulation time, number of nodes, and traffic patterns.

3. Run the Simulation:

- **Execute the Simulation:**
    - Run the simulation script and monitor the output.
    - Ensure that the simulation runs without errors and captures all necessary data.
- **Data Collection:**
    - Collect data on key performance metrics such as throughput, delay, packet loss, and jitter.
    - Use NS3 trace files and Wireshark for detailed packet analysis.

4. Analyze Results:

- **Plot Results:**
    - Use GNUPlot or Matplotlib to plot the results.
    - Analyze how different variables (e.g., number of users, code length) impact CDMA performance.
- **Interference and Capacity Analysis:**
    - Study how interference between different CDMA codes affects performance.
    - Evaluate the system's capacity by increasing the number of users.

Expected Output - Parameters

- **Throughput:** CDMA should allow multiple users to share the same frequency spectrum with minimal interference, leading to stable throughput as long as the code length and power levels are managed well.
- **Delay:** The delay might increase as the number of users increases due to potential code interference and processing delay.
- **Packet Loss:** With an efficient implementation, packet loss should be minimal, but it may increase in high-interference scenarios.
- **Interference Management:** Proper code assignment should reduce interference, but the performance will degrade as the number of users exceeds the system's capacity.

Viva Questions:

1. Define CDMA

2. What is the primary difference between CDMA and other multiple access techniques like TDMA and FDMA?

3. What challenges might arise when implementing CDMA in a simulated environment like NS2/NS3, and how can they be addressed

4. How does the number of users in a CDMA system affect its performance, and what measures can be taken to mitigate potential issues?

Result:

Thus the CDMA functionality is verified and studied successfully.

Experiment No.	: 2
Title of Experiment	: Implement Dynamic routing with RIP and OSPF using CISCO Packet Tracer
Date of Experiment	:

Aim:

To configure and study the implementation of dynamic routing protocols, RIP (Routing Information Protocol) and OSPF (Open Shortest Path First), in a simulated network environment using Cisco Packet Tracer.

Software Required:

1. **Cisco Packet Tracer:**
   - **Version:** Latest available version.
   - **Download Link:** Cisco Networking Academy.
   - **Operating System:** Windows/Linux/MacOS.
2. **PC or Laptop:**
   - **Operating System:** Any compatible OS with Cisco Packet Tracer.
   - **RAM:** Minimum 4 GB.
   - **Processor:** Minimum Dual-core processor.

Theory:

Dynamic Routing is a method where routers automatically adjust the paths to network destinations based on current network conditions. Unlike static routing, where routes are manually configured, dynamic routing allows routers to share routing information and adapt to network changes such as link failures or topology changes.

RIP (Routing Information Protocol) is a distance-vector routing protocol that uses hop count as a routing metric. It periodically broadcasts its entire routing table to its neighbours every 30 seconds. RIP is simple but limited to smaller networks due to its maximum hop count of 15.

OSPF (Open Shortest Path First) is a link-state routing protocol that uses the Dijkstra algorithm to compute the shortest path to each network. Unlike RIP, OSPF sends routing updates only when there are changes in the network, making it more efficient for larger and more complex networks. OSPF supports hierarchical routing using areas, improving scalability and network management.

In this experiment, RIP and OSPF will be configured on routers within a simulated network environment using Cisco Packet Tracer. The performance and behavior of these protocols will be observed by examining how they populate the routing tables and handle network changes.

Steps to Execute

Routing Information Protocol (RIP) is an active routing protocol that operates hop count as a routing metric to find the most suitable route between the source and the destination network. It is a distance-vector routing protocol that has an AD value of 120 and works on the Network layer of the OSI model.

Steps to Configure and Verify Three Router Connections in Cisco Packet Tracer using RIP Routing:

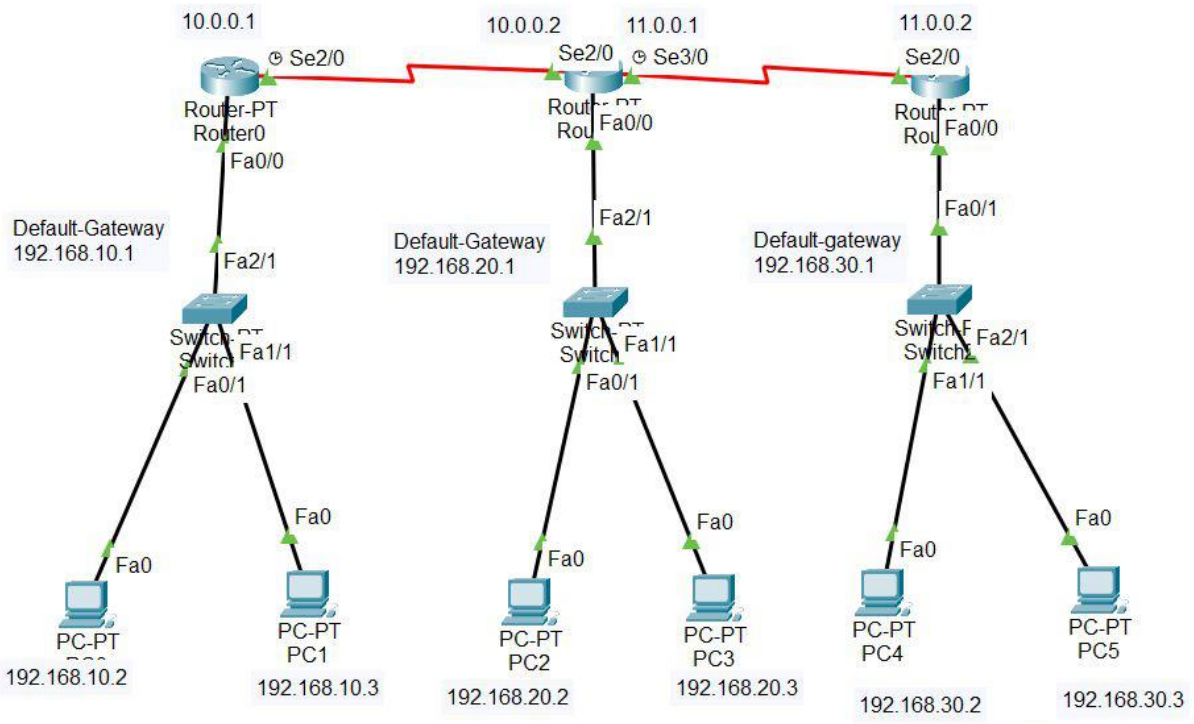
Step 1: First, open the Cisco packet tracer desktop and select the devices given below:

S.NO	Device	Model Name	Qty.
1.	PC	PC	6
2.	Switch	PT-Switch	3
3.	Router	PT-router	3

IP Addressing Table:

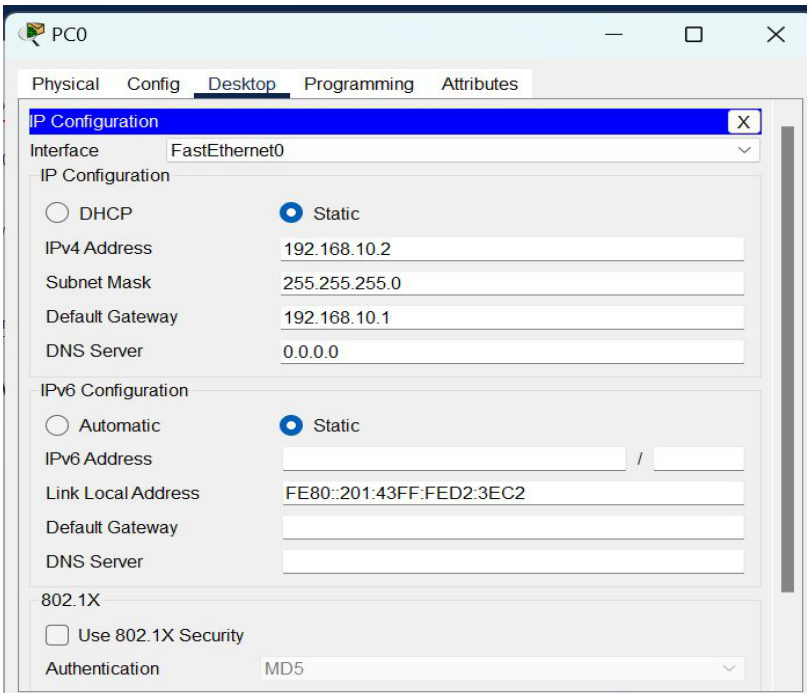
S.NO	Device	IPv4 Address	Subnet mask	Default Gateway
1.	PC0	192.168.10.2	255.255.255.0	192.168.10.1
2.	PC1	192.168.10.3	255.255.255.0	192.168.10.1
3.	PC2	192.168.20.2	255.255.255.0	192.168.20.1
4.	PC3	192.168.20.3	255.255.255.0	192.168.20.1
5.	PC4	192.168.30.2	255.255.255.0	192.168.30.1
6.	PC5	192.168.30.3	255.255.255.0	192.168.30.1

- Then, create a network topology as shown below the image.
- Use an Automatic connecting cable to connect the devices with others.



Step 2: Configure the PCs (hosts) with IPv4 address and Subnet Mask according to the IP addressing table given above.

- To assign an IP address in PC0, click on PC0.
- Then, go to desktop and then IP configuration and there you will IPv4 configuration.
- Fill IPv4 address and subnet mask.



Interface: FastEthernet0

IP Configuration

○ DHCP ● Static

IPv4 Address: 192.168.10.2

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.10.1

DNS Server: 0.0.0.0

IPv6 Configuration

○ Automatic ● Static

IPv6 Address: /

Link Local Address: FE80::201:43FF:FED2:3EC2

Default Gateway:

DNS Server:

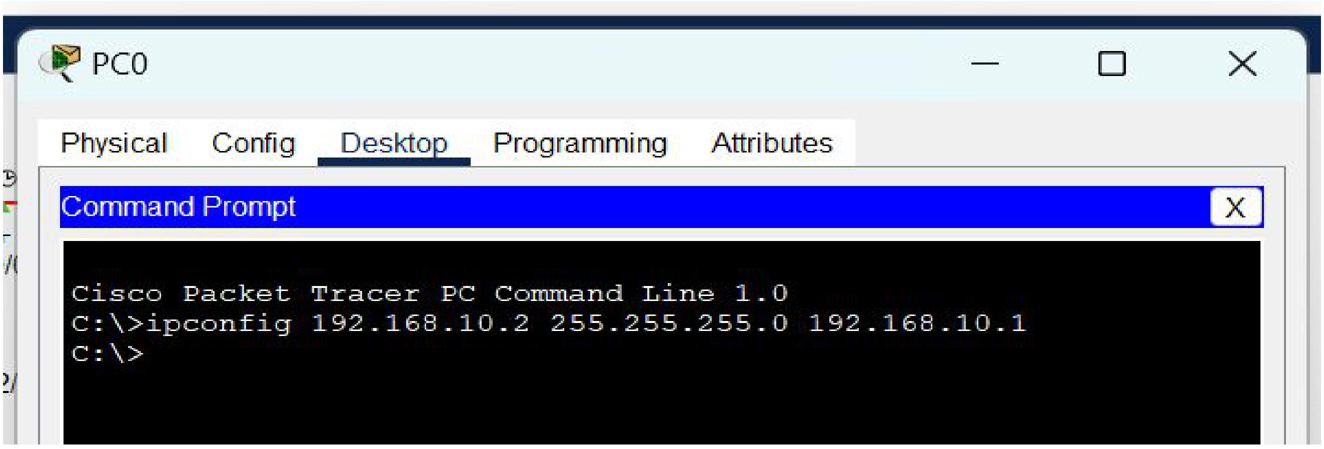
802.1X

□ Use 802.1X Security

Authentication: MD5

- Assigning an IP address using the ipconfig command, or we can also assign an IP address with the help of a command.
- Go to the command terminal of the PC.
- Then, type iPConfig <IPv4 address><subnet mask><default gateway>(if needed)

Example: iPConfig 192.168.10.2 255.255.255.0 192.168.10.1



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ipconfig 192.168.10.2 255.255.255.0 192.168.10.1
C:\>
```

Repeat the same procedure with other PCs to configure them thoroughly.

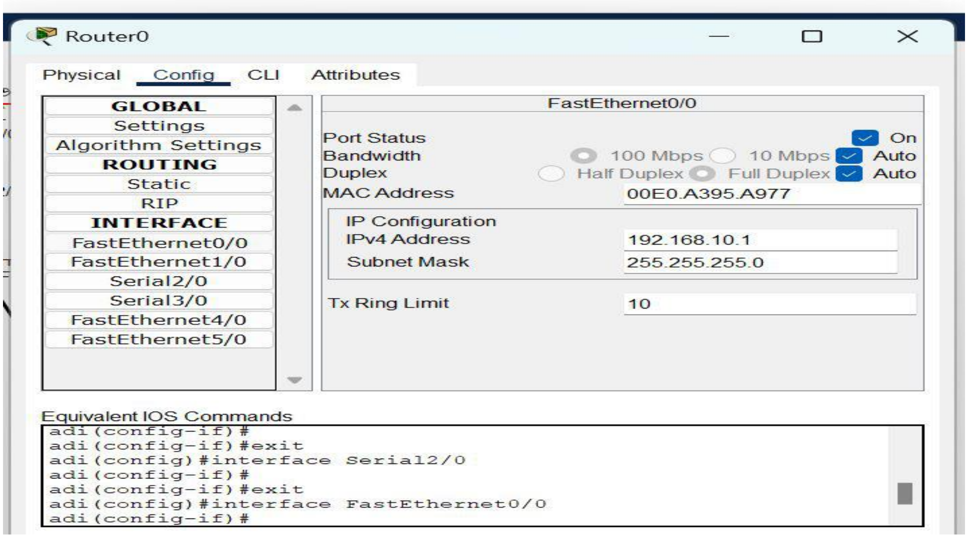
Step 3: Configure router with IP address and Subnet mask.

IP Addressing Table Router:

S.NO	Device	Interface	IPv4 Address	Subnet mask
1.	router0	FastEthernet0/0	192.168.10.1	255.255.255.0
		Serial2/0	10.0.0.1	255.0.0.0
2.	router1	FastEthernet0/0	192.168.20.1	255.255.255.0
		Serial2/0	10.0.0.2	255.0.0.0
		Serial3/0	11.0.0.1	255.0.0.0
3.	router2	FastEthernet0/0	192.168.30.1	255.255.255.0
		Serial2/0	11.0.0.2	255.0.0.0

- To assign an IP address in router0, click on router0.

- Then, go to config and then Interfaces.
- Make sure to turn on the ports.
- Then, configure the IP address in FastEthernet and serial ports according to IP addressing Table.
- Fill IPv4 address and subnet mask.



FastEthernet0/0

Port Status ☑ On

Bandwidth ● 100 Mbps ○ 10 Mbps ☑ Auto

Duplex ○ Half Duplex ● Full Duplex ☑ Auto

MAC Address 00E0.A395.A977

IP Configuration

IPv4 Address 192.168.10.1

Subnet Mask 255.255.255.0

Tx Ring Limit 10

Equivalent IOS Commands

```
adi(config-if)#
adi(config-if)#exit
adi(config)#interface Serial2/0
adi(config-if)#
adi(config-if)#exit
adi(config)#interface FastEthernet0/0
adi(config-if)#
```

Repeat the same procedure with other routers to configure them thoroughly.

Step 4: After configuring all of the devices we need to assign the routes to the routers.

To assign RIP routes to the particular router:

- First, click on router0 then Go to CLI.
- Then type the commands and IP information given below.

CLI command : router rip

CLI command : network <network id>

RIP Routes for Router0 are given below:

Router(config)#router rip

Router(config-router)#network 192.168.10.0

Router(config-router)#network 10.0.0.0

RIP Routes for Router1 are given below:

Router(config)#router rip

Router(config-router)#network 192.168.20.0

Router(config-router)#network 10.0.0.0

Router(config-router)#network 11.0.0.0

RIP Routes for Router2 are given below:

Router(config)#router rip

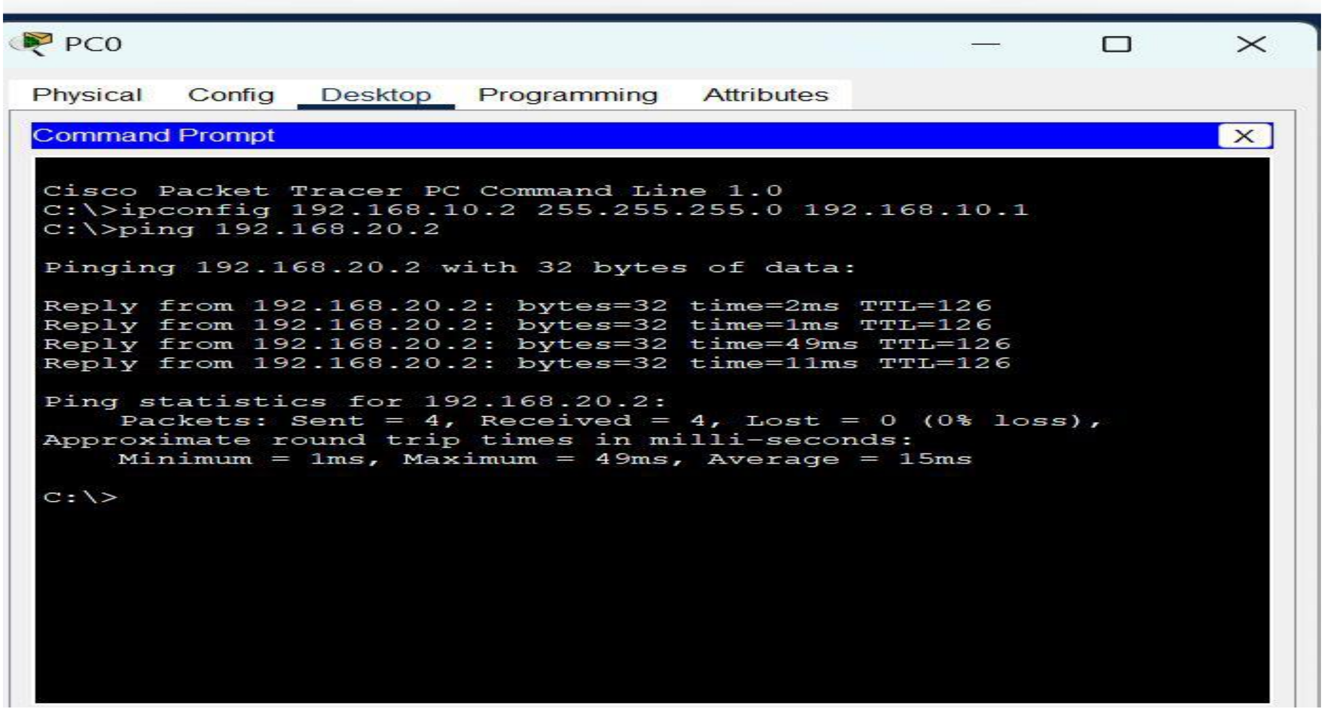
Router(config-router)#network 192.168.30.0

Router(config-router)#network 11.0.0.0

Step 5: Verifying the network by pinging the IP address of any PC.

- We will use the ping command to do so.
- First, click on PC0 then Go to the command prompt.
- Then type ping <IP address of targeted node>.
- As we can see in the below image we are getting replies which means the connection is working properly.

Example : ping 192.168.20.2



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ipconfig 192.168.10.2 255.255.255.0 192.168.10.1
C:\>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2: bytes=32 time=2ms TTL=126
Reply from 192.168.20.2: bytes=32 time=1ms TTL=126
Reply from 192.168.20.2: bytes=32 time=49ms TTL=126
Reply from 192.168.20.2: bytes=32 time=11ms TTL=126

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 49ms, Average = 15ms

C:\>
```

- A simulation of the experiment is given below we are sending PDU from PC0 to PC2 and PC3 to PC5

Procedure

(i) Set Up Network Topology:

- Open Cisco Packet Tracer and create a new project.
- Add three routers, three switches, and six PCs to the workspace.
- Connect the routers to each other and to the switches using serial or Ethernet cables, and connect the PCs to the switches.

(ii) Assign IP Addresses:

- Configure IP addresses on each router's interfaces to ensure proper communication.
- Example: Router 1 with 192.168.1.1/24 for LAN and 10.0.0.1/30 for WAN; similar configurations for Router 2 and Router 3.

(iii) Configure RIP on Routers:

- Access the CLI of each router and enable RIP.
- Use network commands to add the directly connected networks to the RIP process.

(iv) Configure OSPF on Routers:

- Access the CLI of each router and enable OSPF with router ospf 1.
- Use network commands to add networks to OSPF in area 0.

(v) Verify and Test Configurations:

- Use the show ip route command on each router to verify that RIP and OSPF have correctly populated the routing tables.
- Test connectivity by pinging between PCs across different networks.

(vi) Observe Dynamic Routing Behaviour:

- Simulate network changes (e.g., disconnect a link) and observe how RIP and OSPF update the routing tables automatically.
- Reconnect the link and verify that the network recovers and reestablishes the routes.

Viva Questions:

1. Define RIP

2. Differentiate RIP & OSPF

3. Why do we use dynamic routing instead of static routing in larger networks?

4. How does OSPF determine the best path to a destination in a network?

Result:

Thus, the RIP (Routing Information Protocol) and OSPF (Open Shortest Path First) is studied and the dynamic routing protocols implemented in a simulated environment using Cisco Packet Tracer

Experiment No.	: 3
Title of Experiment	: Constructing LAN and perform basic configurations for routers and switches with IPv4 and IPv6 addressing schemes.
Date of Experiment	:

Aim:

To create a simple LAN connection with 5 PC's and 2 Laptops and a Switch and test the connection.

PROCEDURE

WIRED LAN

1. First, we will download Cisco Packet Tracer from netacad.com (latest version).
2. After downloading we will open it and now in this window, we see there are multiplesmall windows where we can select component and create our own particular computer network.
3. Select the components that are listed on the left bottom corner.
4. Select the 2950T switch from the components and place it on the white screen.
5. Place the 5 PC's and 2 laptops from the components and place it on the white screen.
6. Now select the wire from the connections and select copper straight through wire andconnect fast ethernet from PC to the switch.

CONFIGURING THE NETWORK

1. Now assign IP address to each of the PC and laptops and set the subnet mask to255.255.2550.
2. Under fast ethernet tab when you double click on the PC you will able to see fast ethernet and under that set IPv4 Address to the 192.168.1.101, 192.168.1.102, 192.168.1.103, 192.168.1.104, 192.168.1.105 and for laptops 192.168.1.106 and 192.168.1.107.

TESTING THE NETWORK

1. Choose the device you want to test and double click on that and under desktop youwill see the command prompt option
2. Click on that and type the command ping "host ip"(the ip of any other device in thenetwork).
3. The data packets are successfully sent from the source to destination.

WIRELESS LAN

1. First, we will download Cisco Packet Tracer from netacad.com (latest version).
2. After downloading we will open it and now in this window, we see there are multiplesmall windows where we can select component and create our own particular computer network.

3. Select the components that are listed on the left bottom corner.
4. Select the Home router from the components under wireless devices and place it onthe white screen.
5. Place the 5 PC's and 2 laptops from the components and place it on the white screen.

CONFIGURING THE NETWORK

1. Now in each device remove remove the Host NM and replace it with the WMP300N.
2. Now assign IP address to each of the PC and laptops and set the subnet mask to255.255.2550.
3. Under fast ethernet tab when you double click on the PC you will able to see fast ethernet and under that set IPv4 Address to the 192.168.1.101, 192.168.1.102,192.168.1.103, 192.168.1.104, 192.168.1.105 and for laptops 192.168.1.106 and 192.168.1.107.

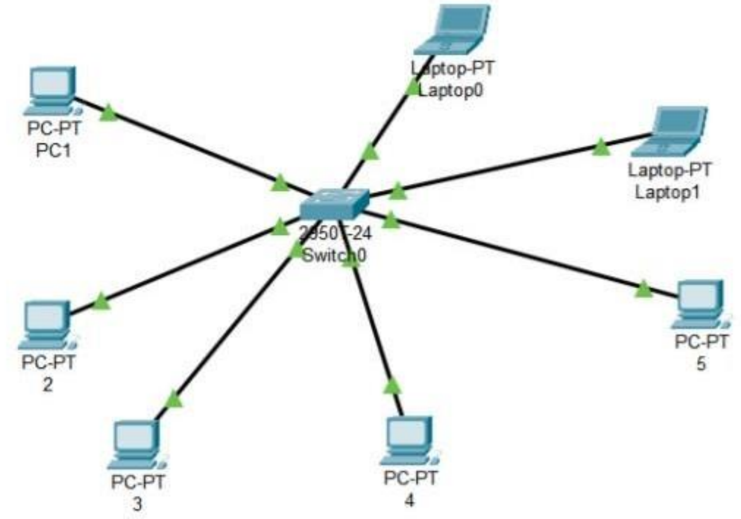
TESTING THE NETWORK

1. Choose the device you want to test and double click on that and under desktop youwill see the command prompt option
2. Click on that and type the command ping "host ip"(the ip of any other device in thenetwork).
3. The data packets are successfully sent from the source to destination.

OUTPUT

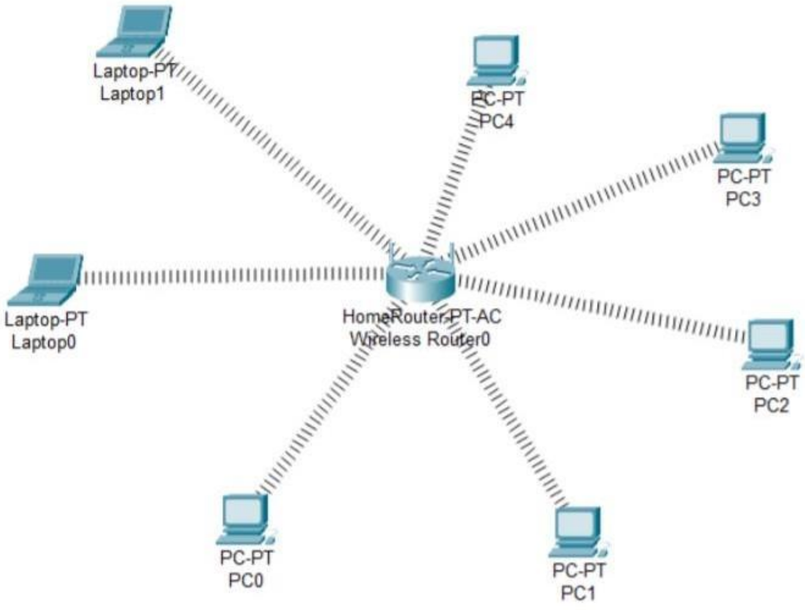
A)WIRED LAN

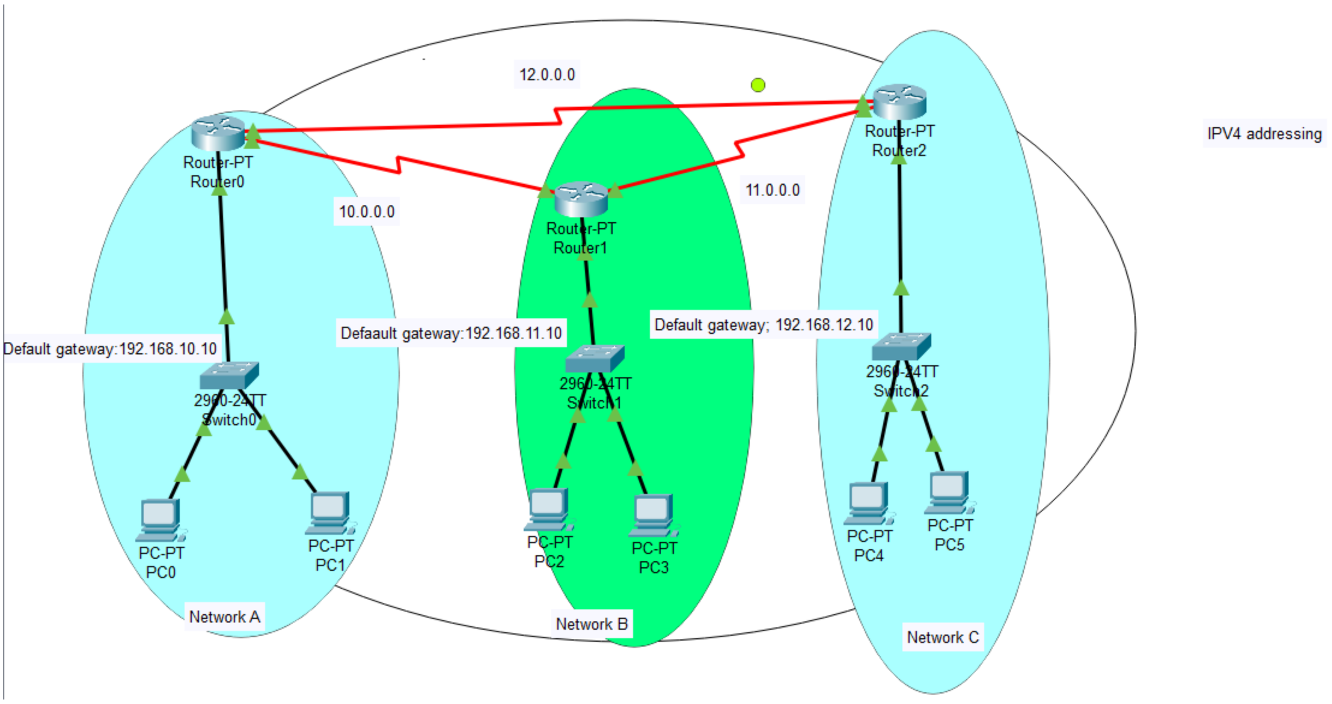
Wired Connection:



B)WIRELESS LAN

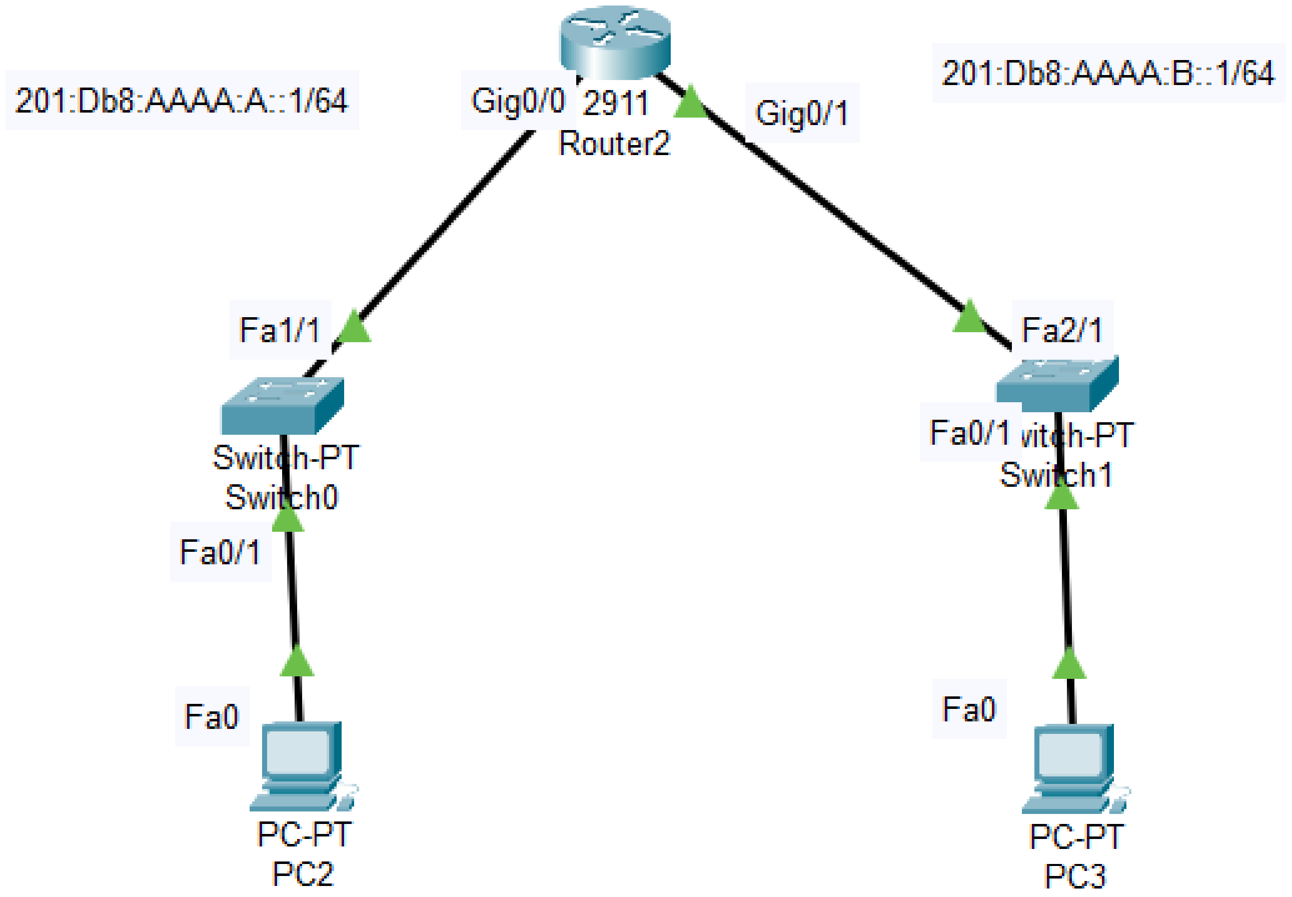
Wireless Connection:





IPV6- Addressing

- An IPv6 (normal) address has the format y:y:y:y:y:y:y:y, where *y* is called a *segment* and can be any hexadecimal value between 0 and FFFF. The segments are separated by colons, not periods. An IPv6 normal address must have eight segments; however, a short form notation can be used in the TS4500 management GUI for segments that are zero, or those that have leading zeros.
- The following are examples of valid IPv6 (normal) addresses:2001:db8:3333:4444:5555:6666:7777:8888
- 2001:db8:3333:4444:CCCC:DDDD:EEEE:FFFF
- :: (implies all 8 segments are zero)
- 2001:db8:: (implies that the last six segments are zero)
- ::1234:5678 (implies that the first six segments are zero)
- 2001:db8::1234:5678 (implies that the middle four segments are zero)
- 2001:0db8:0001:0000:0000:0ab9:C0A8:0102 (This can be compressed to eliminate leading zeros, as follows: 2001:db8:1::ab9:C0A8:102)



VIVA - QUESTIONS

1. What is the difference between IPv4 and IPv6, and why is it important to configure both addressing schemes in modern networks?

2. How do you assign an IPv4 address to a router interface and verify connectivity in a LAN? Can you explain the steps?

3. How do you configure a switch for basic functionality and VLAN creation in a LAN setup?

4. What are the key differences between configuring routing for IPv4 and IPv6 on a router, and how would you enable IPv6 routing?

5. How would you troubleshoot network connectivity in a LAN that uses both IPv4 and IPv6, and which tools or commands would you use?

RESULT

Thus the simple LAN connection with wired and wireless connection are created andtested successfully.

Experiment No.	: 4
Title of Experiment	: Setup an network with IP address.
Date of Experiment	:

Aim:

To setup an simple network with various IP address

PROCEDURE

Step-by-Step Procedure

Step 1: Open Cisco Packet Tracer

1. Launch **Cisco Packet Tracer** on your system.

Step 2: Add Devices to the Workspace

1. **Drag and drop devices** from the device list onto the workspace.
   - **Router:** Select a router (e.g., 1841) from the "Network Devices" section.
   - **Switch:** Drag a switch (e.g., 2960) from the same section.
   - **PCs:** Select two PCs from the "End Devices" section.

Step 3: Connect Devices

1. Use the **"Connections"** tool (cable icon) from the toolbar.
2. Connect the devices using the appropriate cables:
   - Use **copper straight-through** cables to connect:
     - **PC1 → Switch**.
     - **PC2 → Switch**.
   - Use **copper cross-over** cables to connect:
     - **Router → Switch** (choose appropriate ports like FastEthernet 0/0 on the router and FastEthernet 0/1 on the switch).

Step 4: Configure IP Addresses for PCs

1. **Click on PC1**:
   - Go to the **Desktop** tab.
   - Open the **IP Configuration** window.
   - Assign the following IP address:
     - IP Address: 192.168.1.2
     - Subnet Mask: 255.255.255.0
     - Default Gateway: 192.168.1.1
2. **Click on PC2**:
   - Go to the **Desktop** tab.
   - Open the **IP Configuration** window.
   - Assign the following IP address:
     - IP Address: 192.168.1.3
     - Subnet Mask: 255.255.255.0
     - Default Gateway: 192.168.1.1

Step 5: Configure the Router

1. **Click on the Router** and go to the **CLI** tab.
2. Enter the following commands to assign IP addresses to the router's interfaces:

plaintext

Copy code

```
Router> enable
Router# configure terminal
Router(config)# interface fastethernet 0/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# exit
Router# write memory
```

3. The no shutdown command ensures the interface is up and operational.

Step 6: Configure the Switch (Optional)

1. **Click on the Switch** and go to the **CLI** tab.
2. Configure the switch to have a basic IP address for management purposes:

plaintext

Copy code

```
Switch> enable
Switch# configure terminal
Switch(config)# interface vlan 1
Switch(config-if)# ip address 192.168.1.4 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# exit
Switch# write memory
```

Step 7: Verify Connectivity

1. **Click on PC1**:
   - Open the **Command Prompt** from the **Desktop** tab.
   - Type the following command to ping the router:

plaintext

Copy code

```
ping 192.168.1.1
```

   - You should see replies from the router, indicating successful connectivity.
   -

2. **Click on PC2**:
   - Open the **Command Prompt** from the **Desktop** tab.
   - Ping PC1 to ensure they can communicate:

plaintext

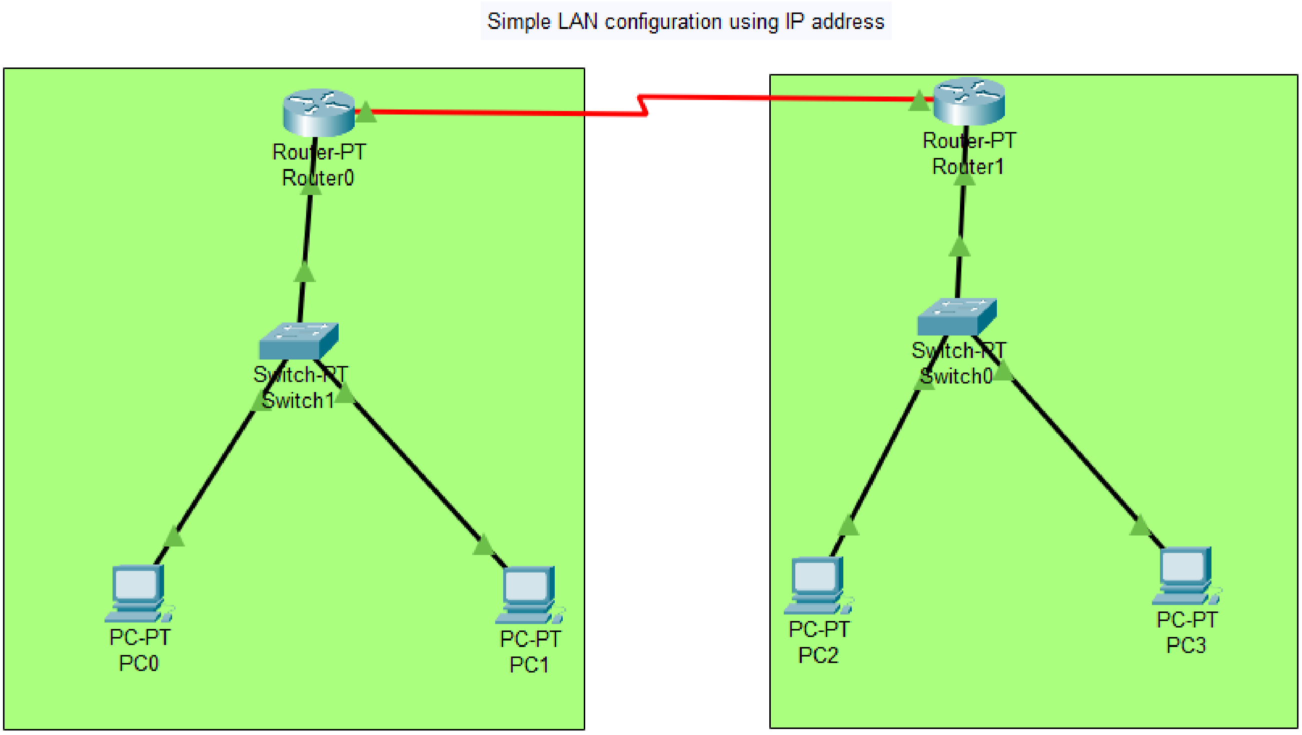
Copy code

```
ping 192.168.1.2
```

Step 8: Save the Configuration

1. **Save the packet tracer file** for future reference by going to **File** → **Save As**.

OUTPUT


Simple LAN configuration using IP address
Router-PT
Router0
Switch-PT
Switch1
PC-PT
PC0
PC-PT
PC1
Router-PT
Router1
Switch-PT
Switch0
PC-PT
PC2
PC-PT
PC3

Experiment No.	: 5
Title of Experiment	: Write a program on a datagram socket for the client/server to display the messages on the client side typed at the server-side
Date of Experiment	:

Aim:

To Write a program on a datagram socket for the client/server to display the messages on the client side typed at the server-side

Theory

In network programming, a Datagram Socket is a type of socket that is used for sending and receiving packets of data using the User Datagram Protocol (UDP). UDP is a connectionless protocol that allows data transmission without establishing a connection, making it faster but less reliable compared to TCP.

Characteristics of UDP:
- Connectionless: No connection is established between the sender and the receiver before communication begins.
- Unreliable: There is no guarantee of data delivery or order. Packets may be lost or delivered out of sequence.
- Fast: Due to its connectionless nature, UDP is faster as it skips the overhead of establishing and maintaining connections.

In a client-server model using Datagram Sockets, the server sends messages to the client, and the client receives and displays them. The interaction is one-way (server to client), and no connection is established between the two.

A network that is connected with two devices as a link to execute two-way communication on the network. It receives and sends data to the devices. The socket address is a combination of IP address and port. In the TCP/IP layer, a socket is bound as a port number which can identify whether the data is to be sent to an application or not. The transport layer in the socket is the core mechanism for managing and establishing communication between the devices.
Sockets are used as a communication device or interaction between the client and server. It receives information from the client and sends information to the client and disconnects it after receiving the data.

Types of Socket:

1. Datagram Sockets: Datagram sockets allow processes to use the User Datagram Protocol (UDP). It is a two-way flow of communication or messages. It can receive messages in a different order from the sending way and also can receive duplicate messages. These sockets are preserved with their boundaries. The socket type of datagram socket is SOCK_DGRAM.

2. Stream Sockets: Stream socket allows processes to use the Transfer Control Protocol (TCP) for communication. A stream socket provides a sequenced, constant or reliable, and two-way (bidirectional) flow of data. After the establishment of connection, data can be read and written to these sockets in a byte stream. The socket type of stream socket is SOCK_STREAM.

3. Raw Sockets: Raw Socket provide user access to the Internet Control Message Protocol (ICMP). Raw sockets are not used for most applications. These sockets are the same as the datagram oriented, their characteristics are dependent on the interfaces. They provided support in developing new communication protocols or for access to more facilities of an existing protocol. Only the superusers can access the Raw Sockets. The socket type of Raw Socket is SOCK_RAW.

4. Sequenced Packet Sockets: Sequenced Packet Sockets are similar to the stream socket, with the exception that record boundaries are preserved in-stream sockets. The given interface in this section is of Network System (NS) that has an abstraction of Sockets and is ordered in all the applications. The Sequenced Packet Sockets enable the user to multiply the sequence packet protocol or some IDP (Internet Datagram Protocol) which heads on the packet or a packet group by writing in the header of the prototype along with the data that has been sent. The socket type of Sequenced Packet Socket is SOCK_SEQPACKET.

Procedure

Step 1: Setup the Server
The server will:
1. Create a DatagramSocket to bind to a specific port.
2. Accept messages typed by the user on the server side.
3. Send these messages to the client via the socket using DatagramPackets.

Step 2: Setup the Client
The client will:
1. Create a DatagramSocket to receive messages from the server.
2. Continuously listen for messages from the server.
3. Display the received messages on the console.

Step 3: Implement the Communication
- The server will read the user input and send it to the client.
- The client will wait for incoming messages from the server and display them.

Java Code

1. Server-side code (DatagramSocket Server)

```java
```

```
import java.net.*;
import java.util.Scanner;

public class DatagramServer {
    public static void main(String[] args) {
        DatagramSocket serverSocket = null;
        try {
            // Create a server socket bound to port 9876
            serverSocket = new DatagramSocket(9876);
            InetAddress clientAddress = InetAddress.getByName("localhost"); // Client address
            Scanner scanner = new Scanner(System.in);

            System.out.println("Server is ready to send messages to the client.");
            while (true) {
                // Read a message from the server user
                System.out.print("Enter message: ");
                String message = scanner.nextLine();

                // Convert the message to a byte array
                byte[] sendData = message.getBytes();

                // Create a datagram packet to send data to the client on port 9875
                DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
clientAddress, 9875);

                // Send the packet
                serverSocket.send(sendPacket);

                // Exit the loop if the user types "exit"
                if (message.equalsIgnoreCase("exit")) {
                    System.out.println("Server exiting...");
                    break;
                }
            }
            scanner.close();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (serverSocket != null && !serverSocket.isClosed()) {
                serverSocket.close();
            }
        }
    }
}
```

2. Client-side code (DatagramSocket Client)

```java
import java.net.*;

public class DatagramClient {
    public static void main(String[] args) {
        DatagramSocket clientSocket = null;
        try {
            // Create a socket to listen on port 9875
            clientSocket = new DatagramSocket(9875);
            byte[] receiveData = new byte[1024]; // Buffer for receiving data

            System.out.println("Client is ready to receive messages.");
            while (true) {
                // Create a datagram packet to receive data
                DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

                // Receive data from the server
                clientSocket.receive(receivePacket);

                // Convert the byte array into a string
                String message = new String(receivePacket.getData(), 0,
receivePacket.getLength());

                // Display the message
                System.out.println("Server says: " + message);

                // Exit if the message is "exit"
                if (message.equalsIgnoreCase("exit")) {
                    System.out.println("Client exiting...");
                    break;
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (clientSocket != null && !clientSocket.isClosed()) {
                clientSocket.close();
            }
        }
    }
}
```

Explanation

1. DatagramSocket: Both the server and client use a DatagramSocket to send and receive data packets. The server sends messages via a socket bound to port 9876, and the client listens for these messages on port 9875.

2. DatagramPacket: This is used to wrap the message into a packet before sending and receiving. The server creates a packet containing the user’s message and sends it to the client. The client receives this packet and extracts the message.

3. Byte Array: The message is converted to a byte array before sending because DatagramPacket operates on byte data.

4. End of Communication: Both the client and server check if the message is "exit". If so, the sockets are closed, and the communication is terminated.

Expected Outcome

1. Server Output:
   - The server prompts the user to enter a message.
   - Once a message is typed, it is sent to the client.
   - The server terminates when "exit" is entered.

2. Client Output:
   - The client waits for a message from the server.
   - When the server sends a message, the client displays it on the console.
   - The client terminates when the message "exit" is received.

Sample Run

Server Side:

```
Server is ready to send messages to the client.
Enter message: Hello Client
Enter message: How are you?
Enter message: exit
Server exiting...
```

Client Side:

```
Client is ready to receive messages.
Server says: Hello Client
Server says: How are you?
Server says: exit
Client exiting...
```

Viva Questions

1. What is the difference between UDP and TCP, and why would you use UDP in this scenario?

2. How does a DatagramPacket work, and what is its role in sending and receiving messages?

3. What happens if a packet is lost during transmission in UDP? How can you handle this in your program?

4. Explain why DatagramSockets are considered connectionless and how that affects communication between the client and server.

5. Can you modify the program to handle multiple clients simultaneously? How would you implement this?

Results

This program demonstrates how to create a simple UDP-based communication system using Datagram Sockets. The server sends messages to the client, which displays them in real-time. Since UDP is connectionless, it provides a fast but unreliable communication mechanism.

Experiment No.	: 6
Title of Experiment	: Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion using packet tracer tool.
Date of Experiment	:

AIM:

To Implement transmission of ping messages/trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion using packet tracer tool.

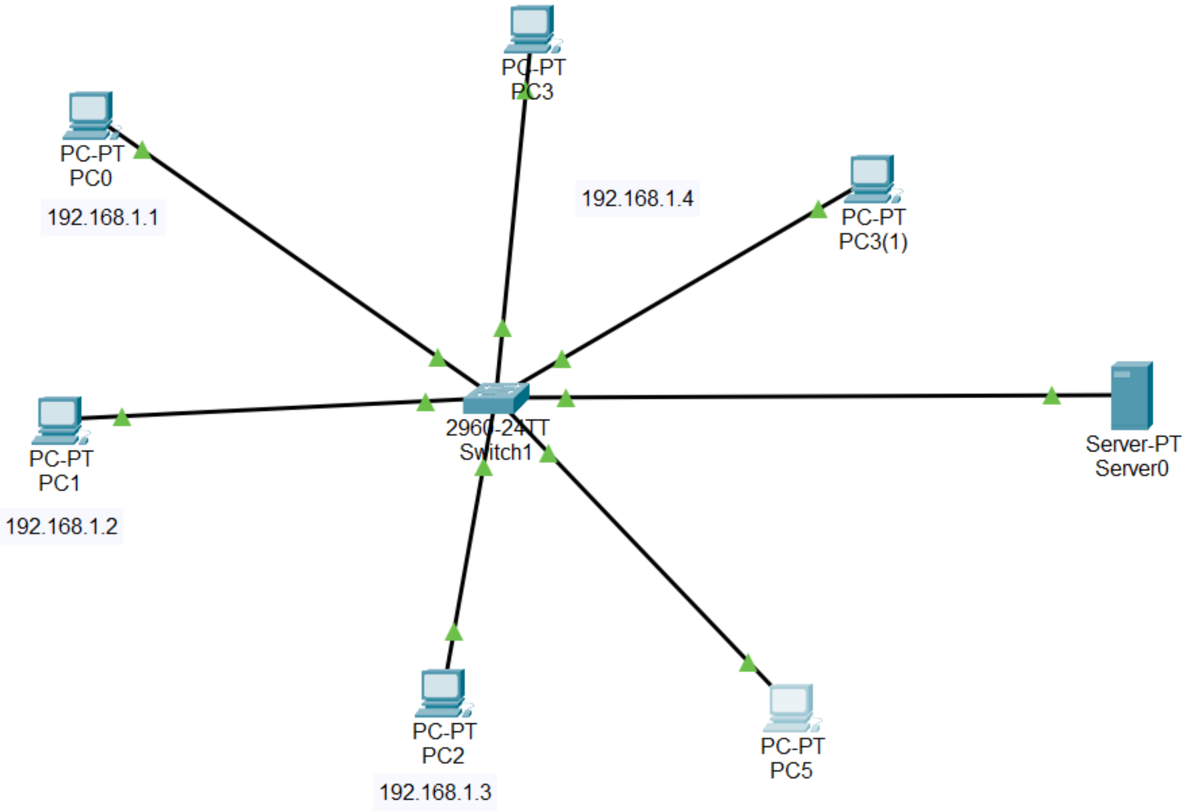
Theory

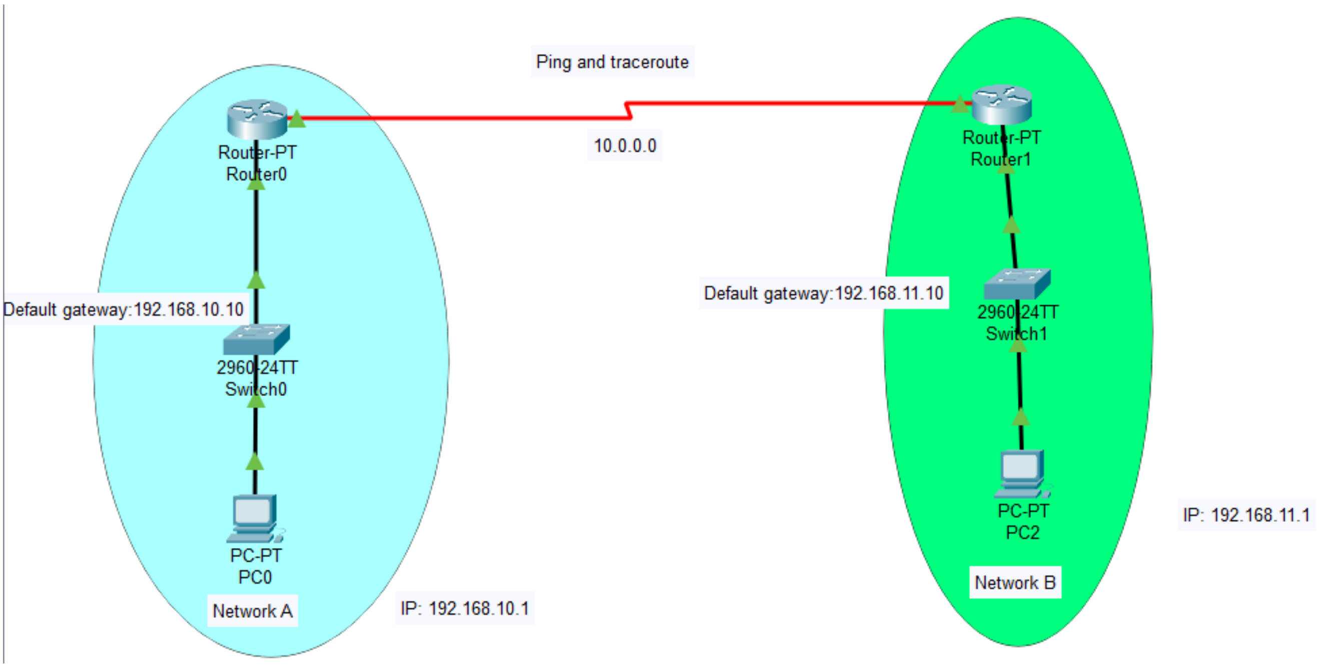
In network diagnostics, ping and traceroute are essential tools for testing network connectivity and analyzing the path packets take across a network. These utilities are vital in evaluating network performance, detecting delays, and diagnosing issues like packet loss due to congestion. Ping uses the Internet Control Message Protocol (ICMP) to send echo request messages to a target and listens for an echo reply. It measures the round-trip time (RTT) between the sender and the receiver, thus testing if the network is operational. Packet loss or delayed responses in ping indicate possible network congestion or failures.

Traceroute, on the other hand, maps the route that packets take through the network. It provides information about each hop along the way, including the router's IP address and the RTT for each segment of the journey. This tool is useful for identifying where congestion or packet drops occur along the route from the source to the destination.

In Packet Tracer, we can simulate a network with six nodes consisting of routers and PCs. The routers handle data forwarding, while PCs act as the endpoints generating traffic. To evaluate congestion, heavy traffic can be introduced by sending multiple ping requests simultaneously or transmitting large data volumes between nodes. This simulates real-world congestion when routers' buffers overflow due to excessive traffic.

Packet Tracer’s simulation mode helps visualize the path packets take across the network and enables us to monitor congestion-related events like packet drops. By analyzing the logs and packet flow, we can determine how many packets were lost due to buffer overflows or delayed responses at any router. This analysis provides insights into network efficiency, helping design more optimized and congestion-resilient networks.





Procedure

1. Network Design:
   - Open Cisco Packet Tracer.
   - Place 6 nodes on the workspace: 3 routers (R1, R2, R3) and 3 PCs (PC1, PC2, PC3).

- Connect the nodes using copper straight-through cables (use appropriate interfaces like Ethernet or FastEthernet).
- Configure IP addressing for each node. Assign IP addresses to the router interfaces and PC nodes.

2. Routing Configuration:
   - Configure **static routes** or use a **dynamic routing protocol** like RIP or OSPF on the routers to ensure all PCs can communicate across the network.
   - Check for basic connectivity using the `ping` command between different nodes to verify that the network is operational.

3. Ping and Traceroute Testing:
   - On PC1, use the ping command to send ICMP echo requests to PC3 (e.g., `ping PC3_IP_address`).
   - Similarly, on PC1, run a traceroute command to trace the route from PC1 to PC3 (e.g., `tracert PC3_IP_address`).

4. Simulate Network Congestion:
   - Introduce network congestion by creating heavy traffic between the routers. This can be done by sending large data packets or initiating multiple simultaneous ping requests from different PCs.
   - Open the simulation mode in Packet Tracer to visualize the flow of packets across the network.

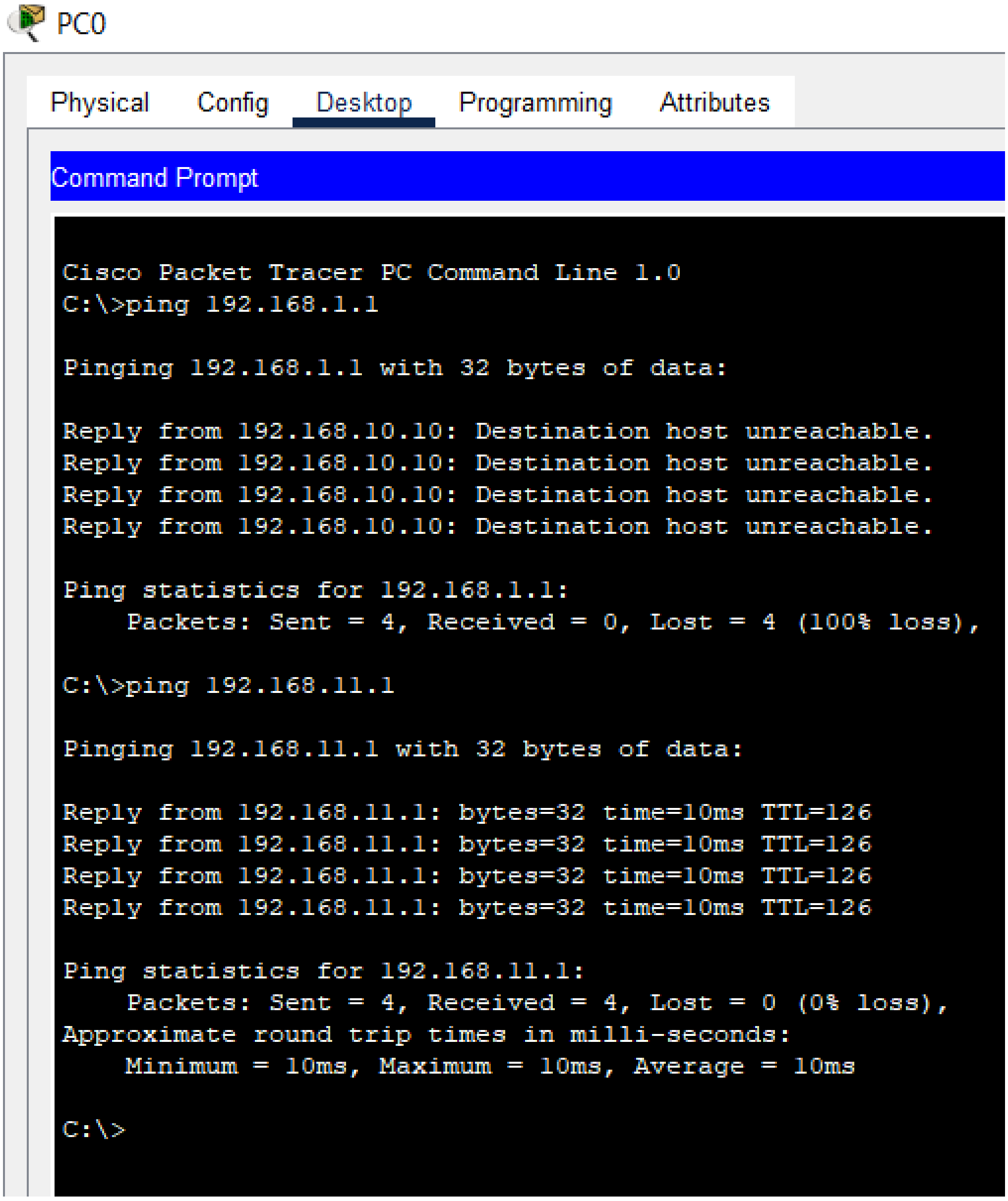
5. Monitor Packet Drop:
   - While running the simulation, observe the network's performance and check for packet drops at different points due to congestion.
   - You can monitor the routers’ interfaces and buffer statistics to see where the congestion is occurring and where the packet loss happens.

6. View Packet Statistics:
   - In the simulation mode, inspect the ICMP packets and traceroute hops. Analyze where packets are getting delayed or dropped due to congestion by viewing Event List and Packet Log.

Expected Outcome

1. Ping Output:

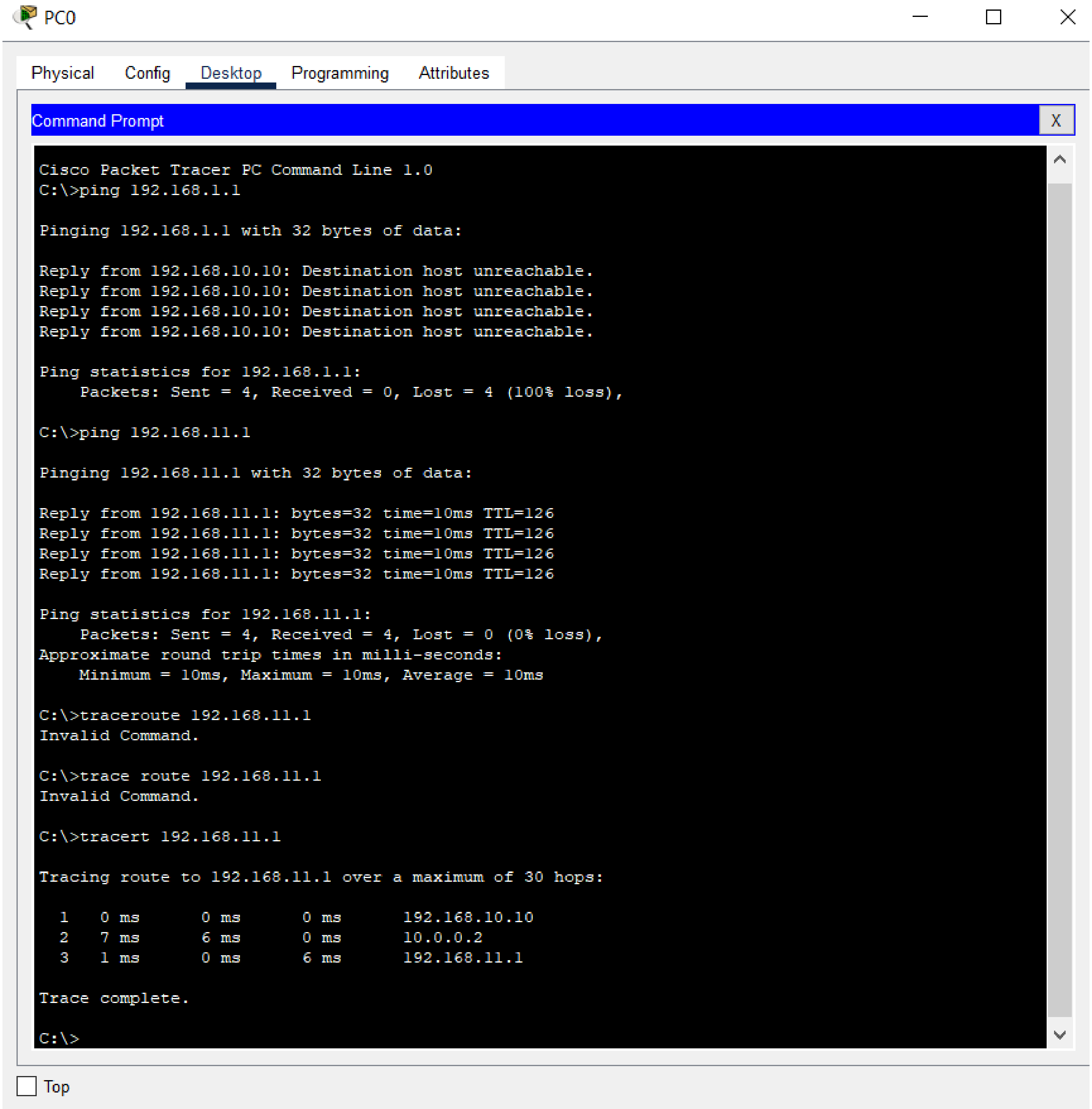
- The ping command will successfully show the round-trip time (RTT) between PC1 and PC3 unless congestion causes packet drops. In the case of dropped packets, the ping will report lost packets.
- You should see some packets being successfully sent and received, while others may be dropped due to congestion.



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.10.10: Destination host unreachable.
Reply from 192.168.10.10: Destination host unreachable.
Reply from 192.168.10.10: Destination host unreachable.
Reply from 192.168.10.10: Destination host unreachable.

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.11.1

Pinging 192.168.11.1 with 32 bytes of data:

Reply from 192.168.11.1: bytes=32 time=10ms TTL=126
Reply from 192.168.11.1: bytes=32 time=10ms TTL=126
Reply from 192.168.11.1: bytes=32 time=10ms TTL=126
Reply from 192.168.11.1: bytes=32 time=10ms TTL=126

Ping statistics for 192.168.11.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 10ms, Average = 10ms

C:\>
```

2. Traceroute Output:

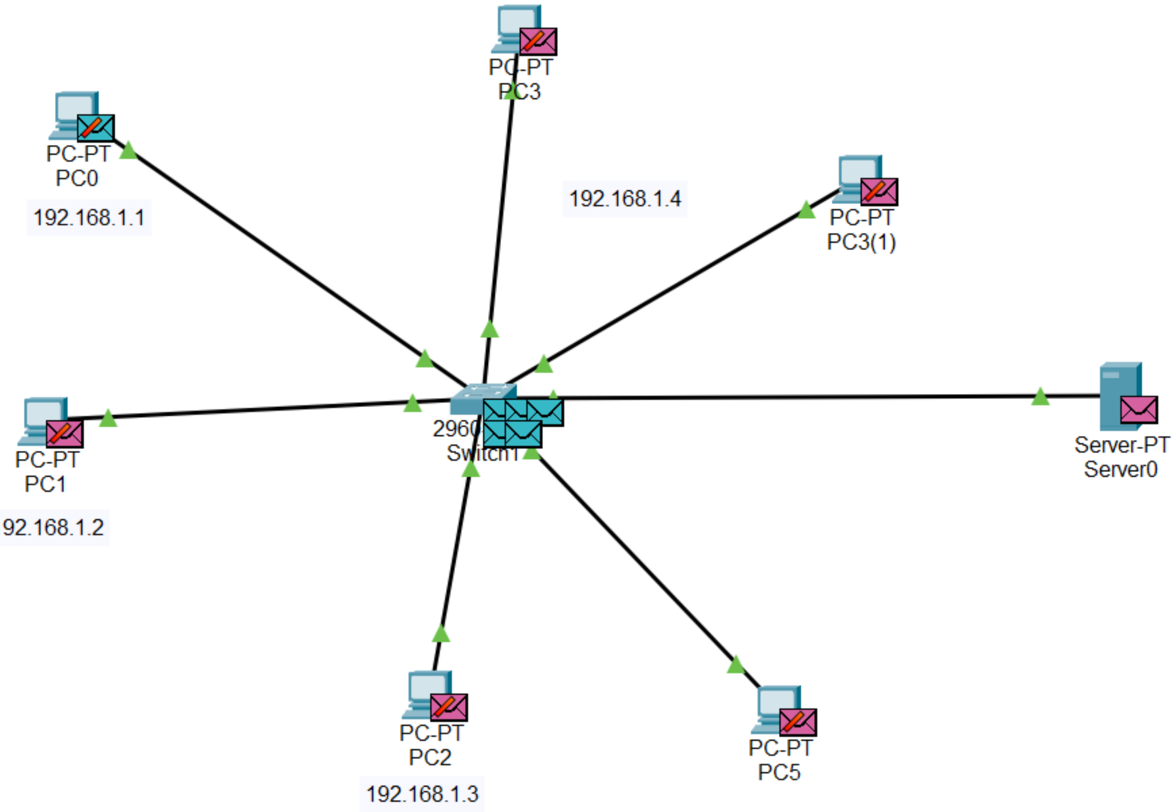
- The traceroute command will display the path taken by the packets from PC1 to PC3, showing the routers (R1, R2, and R3) as the hops.
- If congestion is present at any node, you will observe delays or packet drops, and some hops may be missing from the output.



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.10.10: Destination host unreachable.
Reply from 192.168.10.10: Destination host unreachable.
Reply from 192.168.10.10: Destination host unreachable.
Reply from 192.168.10.10: Destination host unreachable.

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.11.1

Pinging 192.168.11.1 with 32 bytes of data:

Reply from 192.168.11.1: bytes=32 time=10ms TTL=126
Reply from 192.168.11.1: bytes=32 time=10ms TTL=126
Reply from 192.168.11.1: bytes=32 time=10ms TTL=126
Reply from 192.168.11.1: bytes=32 time=10ms TTL=126

Ping statistics for 192.168.11.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 10ms, Average = 10ms

C:\>traceroute 192.168.11.1
Invalid Command.

C:\>trace route 192.168.11.1
Invalid Command.

C:\>tracert 192.168.11.1

Tracing route to 192.168.11.1 over a maximum of 30 hops:

  1   0 ms      0 ms      0 ms      192.168.10.10
  2   7 ms      6 ms      0 ms      10.0.0.2
  3   1 ms      0 ms      6 ms      192.168.11.1

Trace complete.

C:\>
```

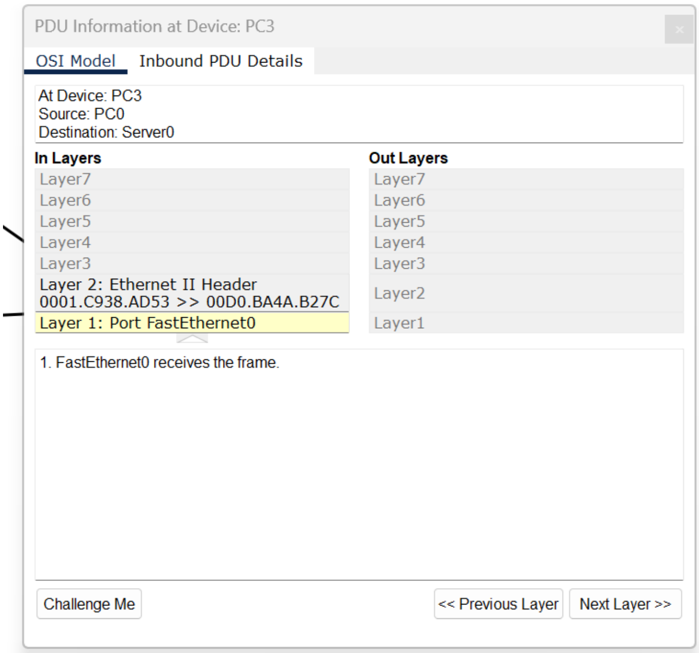
3. Packet Drop Analysis:

- In simulation mode, you should observe some packets being dropped at the router level due to buffer overflow or network congestion.
- Using the Packet Tracer Simulation Panel, you can inspect which packets were dropped, the reason for the drop, and how many packets failed to reach the destination due to congestion.



4. Congestion Statistics:

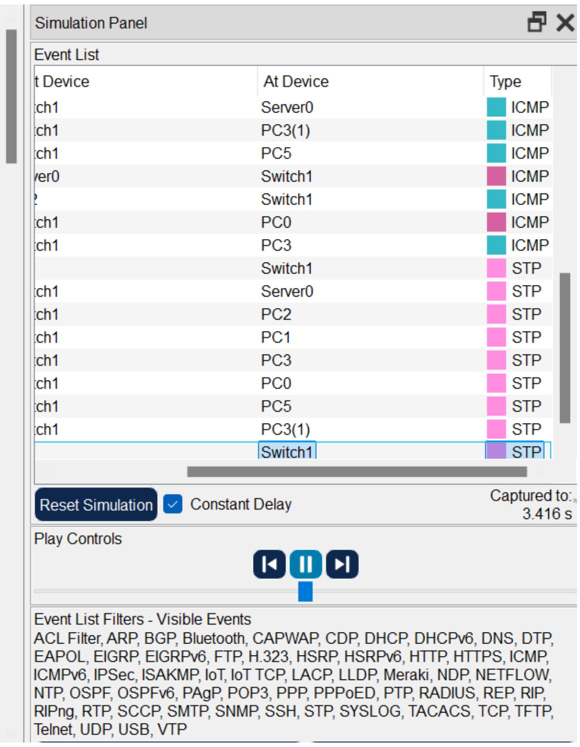
- The number of packets dropped can be calculated by reviewing the packet flow during the simulation and checking router buffer stats.
- As network congestion builds, the number of packet drops increases. This can be observed in the packet drop logs generated in Packet Tracer.



At Device: PC3
Source: PC0
Destination: Server0

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer3	Layer3
Layer 2: Ethernet II Header 0001.C938.AD53 >> 00D0.BA4A.B27C	Layer2
Layer 1: Port FastEthernet0	Layer1

1. FastEthernet0 receives the frame.



Event List

t Device	At Device	Type
ch1	Server0	■ ICMP
ch1	PC3(1)	■ ICMP
ch1	PC5	■ ICMP
ver0	Switch1	■ ICMP
	Switch1	■ ICMP
ch1	PC0	■ ICMP
ch1	PC3	■ ICMP
	Switch1	■ STP
ch1	Server0	■ STP
ch1	PC2	■ STP
ch1	PC1	■ STP
ch1	PC3	■ STP
ch1	PC0	■ STP
ch1	PC5	■ STP
ch1	PC3(1)	■ STP
	Switch1	■ STP

☑ Constant Delay

Captured to:
3.416 s

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

Viva Questions

1. What is the difference between ping and traceroute, and in what situations would you use each?

2. How does network congestion affect packet delivery, and how can it be simulated in Packet Tracer?

3. What role do ICMP messages play in the ping command, and how are they processed in the network?

4. How can you interpret the output of a traceroute command to identify bottlenecks or points of congestion?

5. What are some methods to prevent packet loss due to congestion in a real-world network?

Conclusion

This simulation demonstrates how to configure and evaluate network performance using ping and traceroute commands in Cisco Packet Tracer. By introducing congestion in the network, the simulation illustrates how packet loss occurs due to overwhelmed buffers and provides insight into the importance of network optimization to reduce packet drops.

Experiment No.	: 7
Title of Experiment	: Configure routers, switches and end devices to provide access to local and remote network resources and to enable end-to-end connectivity between remote devices.
Date of Experiment	:

AIM:

The objective is to configure routers, switches, and end devices in a network to ensure local and remote network connectivity, allowing end-to-end communication between devices. This configuration includes setting up routing, IP addressing, and switch operation to enable seamless access to both local and external network resources.

Apparatus/Requirements:

1. Hardware/Software Tools:
   - Cisco Packet Tracer software (for simulation).
   - Routers: Minimum of 2 Cisco Routers (e.g., 2911 series).
   - Switches: Minimum of 2 Cisco Switches (e.g., 2960 series).
   - End Devices: At least 2 PCs or laptops, along with their network interfaces.
   - Ethernet Cables (Copper straight-through or crossover cables).
   - Serial cables (for router-to-router connection, if needed).

2. Configurations:
   - Router Configuration: IP addressing, routing protocols (e.g., RIP, OSPF, static routes).
   - Switch Configuration: VLAN setup, port assignments, trunking (if needed).
   - PC Configuration: IP addressing, gateway settings.

3. Network Services:
   - DHCP (Dynamic Host Configuration Protocol) for automatic IP assignment.
   - DNS (Domain Name System) for resolving domain names.

Theory

Configuring routers, switches, and end devices is a fundamental aspect of networking that ensures access to both local and remote network resources, providing seamless end-to-end connectivity between devices. In a typical network, these devices play critical roles: routers handle traffic between different networks, switches manage communication within the same network (or LAN), and end devices like PCs and servers are the network's primary users, generating and consuming data. A well-configured network enables efficient and reliable data transmission, supporting everything from local file sharing to global internet access.

Routers are the backbone of inter-network communication. Their primary role is to forward data packets between different networks based on IP addressing. Routers operate at Layer 3 of the OSI model (the network layer), making routing decisions based on the destination IP address in each packet. To ensure connectivity between remote devices, routers must be configured with appropriate IP addresses for their interfaces and routing protocols such as RIP, OSPF, or static routes. These protocols allow routers to determine the best path for data to travel through the network. In smaller networks, static routes may be used, where the administrator manually defines the path. In larger, more dynamic environments, dynamic routing protocols like OSPF are more efficient as they automatically update the routing tables based on network conditions.

Switches are critical for managing communication within a Local Area Network (LAN). They operate at Layer 2 of the OSI model (the data link layer) and use MAC addresses to forward data frames to the correct device within the LAN. Switches ensure that multiple devices within a LAN can communicate simultaneously without collisions. When configuring switches, features like VLANs (Virtual Local Area Networks) can be implemented to improve network security and performance by logically segmenting the network into different broadcast domains. This way, traffic from one VLAN cannot directly communicate with another, limiting the scope of broadcast traffic and improving network efficiency. Switches can also be configured for trunking, which allows multiple VLANs to be transmitted over a single physical connection between switches and routers, preserving the network’s segmentation while allowing inter-VLAN routing.

End devices, such as PCs, servers, and printers, are the users of the network and need proper configuration to access both local and remote resources. Each end device must be assigned a unique IP address, which can be done statically or dynamically using DHCP (Dynamic Host Configuration Protocol). Additionally, devices need to have a properly configured default gateway, typically the router's IP address, to send traffic outside their local network. Without the correct default gateway, an end device will be able to communicate within its local network but not beyond it. The DNS (Domain Name System) server should also be configured to resolve domain names to IP addresses, facilitating easier access to remote services.

In real-world scenarios, networks often consist of multiple local and remote segments that must communicate with one another. This requires inter-network communication between different subnets or even across wide geographical areas. A typical enterprise network could involve multiple LANs, each managed by switches, and interconnected through routers. These routers not only handle traffic between different subnets but also serve as gateways to the internet or to remote office networks. By configuring **routing tables**, routers can learn about remote networks and determine the most efficient path for forwarding packets, ensuring end-to-end connectivity.

Procedure:

1. Device Placement:
   - Open **Packet Tracer** and place two **routers**, two **switches**, and two **PCs** on the workspace.

- Connect the devices using the appropriate **Ethernet cables** (use copper straight-through cables between switches, routers, and PCs; crossover if necessary between similar devices).

2. IP Addressing:
   - Assign IP addresses to the routers' interfaces and configure subnets. For example:
     - Router 1 (R1):
       - LAN Interface (e.g., GigabitEthernet0/0): 192.168.1.1/24
       - WAN Interface (e.g., Serial0/0/0): 10.0.0.1/30
     - Router 2 (R2):
       - LAN Interface (e.g., GigabitEthernet0/0): 192.168.2.1/24
       - WAN Interface (e.g., Serial0/0/0): 10.0.0.2/30
   - Assign IP addresses to the PCs, ensuring they belong to the correct subnet (e.g., PC1: 192.168.1.2/24, PC2: 192.168.2.2/24).
   - Configure default gateways for the PCs to match the router interface IP (e.g., PC1 gateway: 192.168.1.1, PC2 gateway: 192.168.2.1).

3. Router Configuration:
   - Access the CLI of **Router 1** and **Router 2** via Packet Tracer.
   - Set up basic configurations:

```bash
Router(config)# hostname Router1
Router1(config)# interface gigabitEthernet 0/0
Router1(config-if)# ip address 192.168.1.1 255.255.255.0
Router1(config-if)# no shutdown
Router1(config-if)# exit

Router1(config)# interface serial 0/0/0
Router1(config-if)# ip address 10.0.0.1 255.255.255.252
Router1(config-if)# clock rate 64000  # Set clock on DCE side
Router1(config-if)# no shutdown
Router1(config-if)# exit
```

   - Perform a similar setup on Router 2, configuring the serial interface and LAN interface:

```bash
Router2(config)# hostname Router2
Router2(config)# interface gigabitEthernet 0/0
Router2(config-if)# ip address 192.168.2.1 255.255.255.0
Router2(config-if)# no shutdown
Router2(config-if)# exit

Router2(config)# interface serial 0/0/0
Router2(config-if)# ip address 10.0.0.2 255.255.255.252
Router2(config-if)# no shutdown
```

   - Enable static routing or dynamic routing (e.g., RIP or OSPF):

- Static Routing (on Router 1 and Router 2):
```bash
Router1(config)# ip route 192.168.2.0 255.255.255.0 10.0.0.2
Router2(config)# ip route 192.168.1.0 255.255.255.0 10.0.0.1
```
- Alternatively, configure dynamic routing such as **RIP** or **OSPF** if desired.

4. Switch Configuration:
- Assign VLANs if needed and configure switch ports for the correct VLAN:
```bash
Switch(config)# interface range fastEthernet 0/1 - 12
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 10
```
- Ensure trunking is configured between switches if connecting them:
```bash
Switch(config)# interface gigabitEthernet 0/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# exit
```

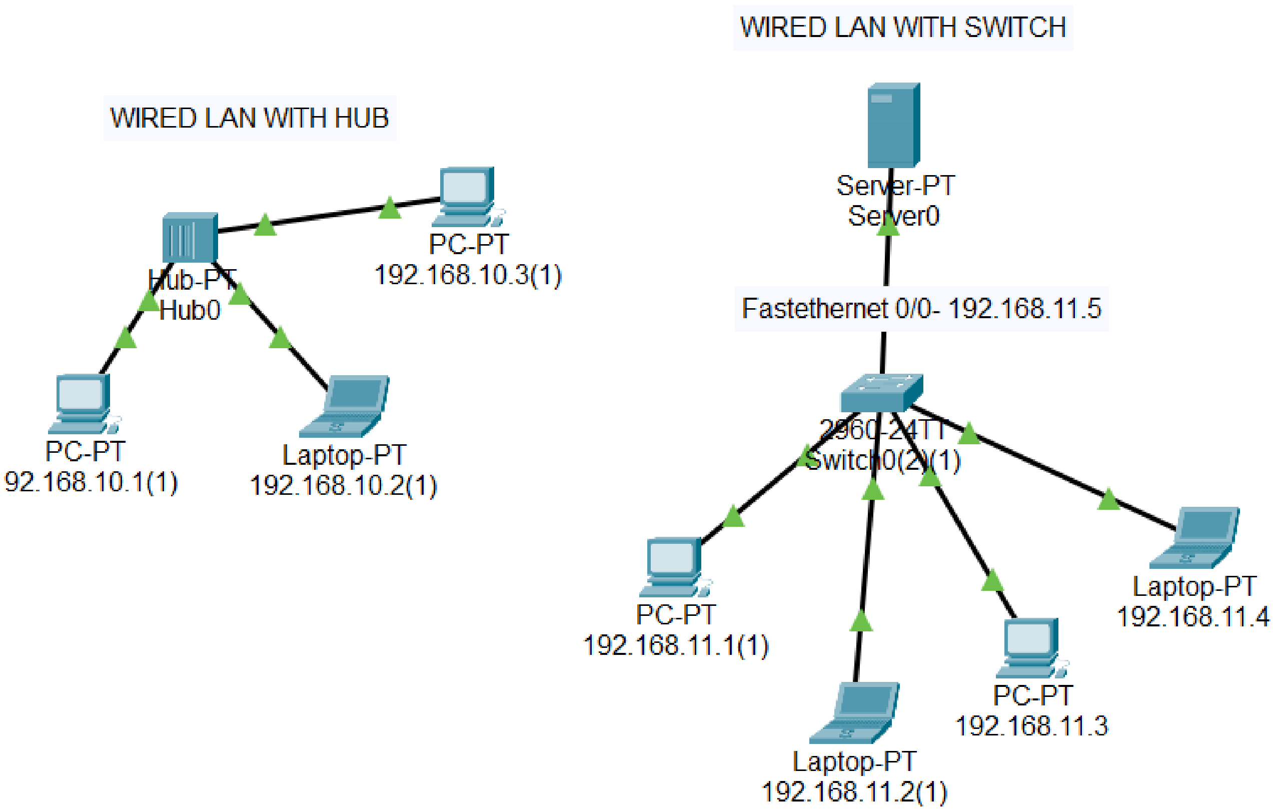
5. Testing Connectivity:
- Use the `ping` command from PC1 to PC2 to ensure end-to-end connectivity. For example:
```bash
PC1> ping 192.168.2.2
```
- Test remote connectivity by pinging the router interfaces from both PCs.
- Use traceroute to verify the routing path taken by the packets across the network:
```bash
PC1> tracert 192.168.2.2
```

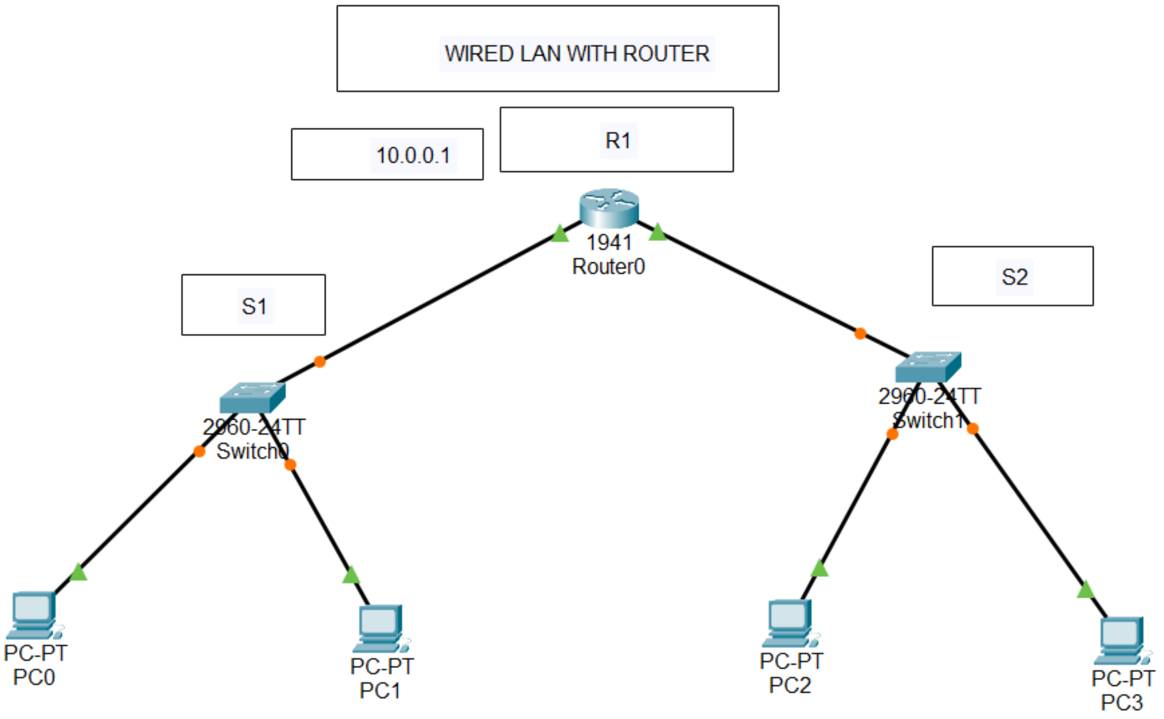
6. Enable DHCP and DNS (Optional):
- DHCP Configuration on the router:
```bash
Router1(config)# ip dhcp pool LAN1
Router1(dhcp-config)# network 192.168.1.0 255.255.255.0
Router1(dhcp-config)# default-router 192.168.1.1
```
- Configure DNS if required to resolve domain names for the PCs.

Expected Outcome:

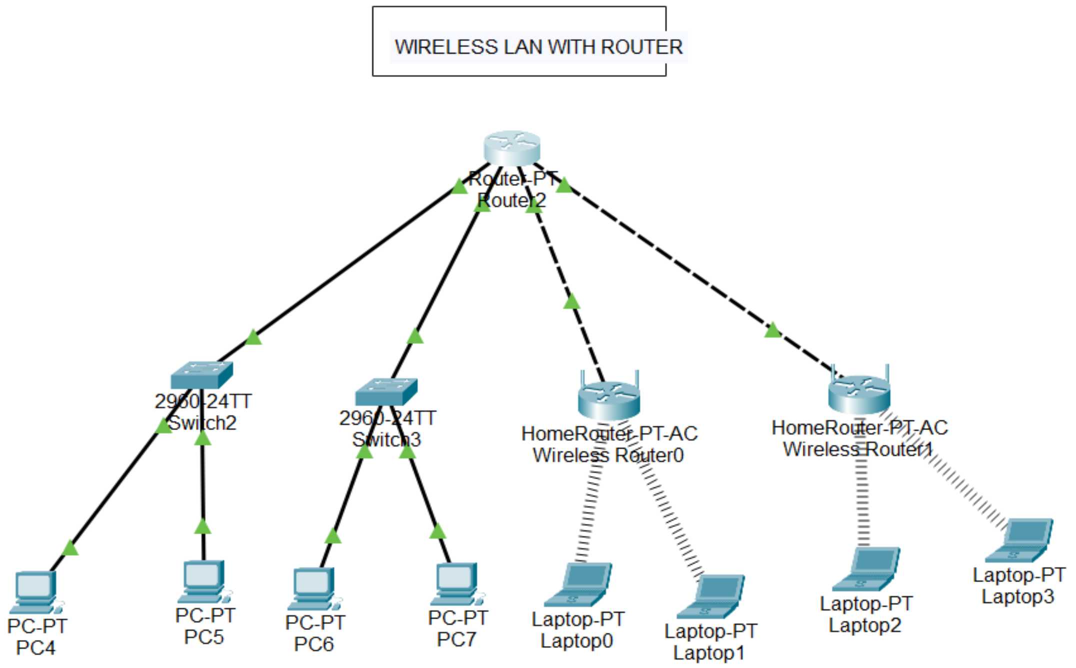
1. Local and Remote Network Access: Both PCs should have local access within their respective LANs (PC1 in the 192.168.1.0/24 network and PC2 in the 192.168.2.0/24 network) and remote access through the routers.

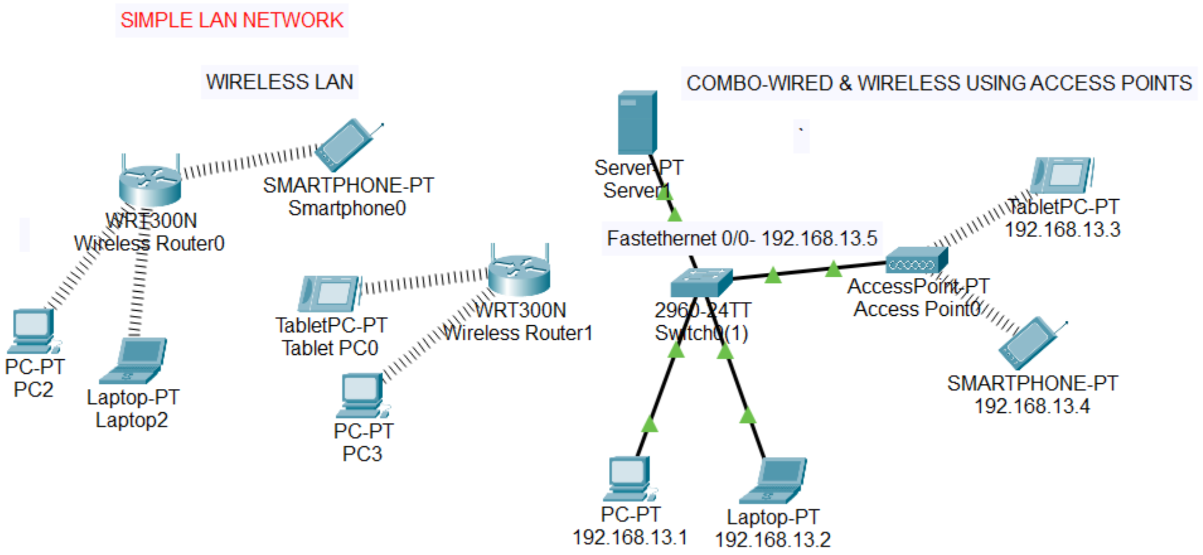
2. End-to-End Connectivity**: The ping command will succeed when testing connectivity between PC1 and PC2, confirming proper router and switch configurations.
3. Routing Path Visibility: Using the traceroute command will display the hops taken between PC1 and PC2, passing through the routers.





3. Packet Flow and Communication: The routers will successfully route packets between different subnets, enabling full communication between local and remote devices in the network.





Viva Questions:

1. Why is a default gateway necessary on end devices, and how does it function in inter-network communication?

2. What is the significance of using static routing versus dynamic routing protocols like OSPF or RIP?

3. How does VLAN configuration on switches improve network security and efficiency?

4. Explain how DHCP and DNS services enhance network management in a real-world network environment.

5. What factors affect the performance of a router in handling traffic between multiple networks?

Conclusion:

The Configure routers, switches and end devices to provide access to local and remote network resources and to enable end-to-end connectivity between remote devices are studied and verified successfully

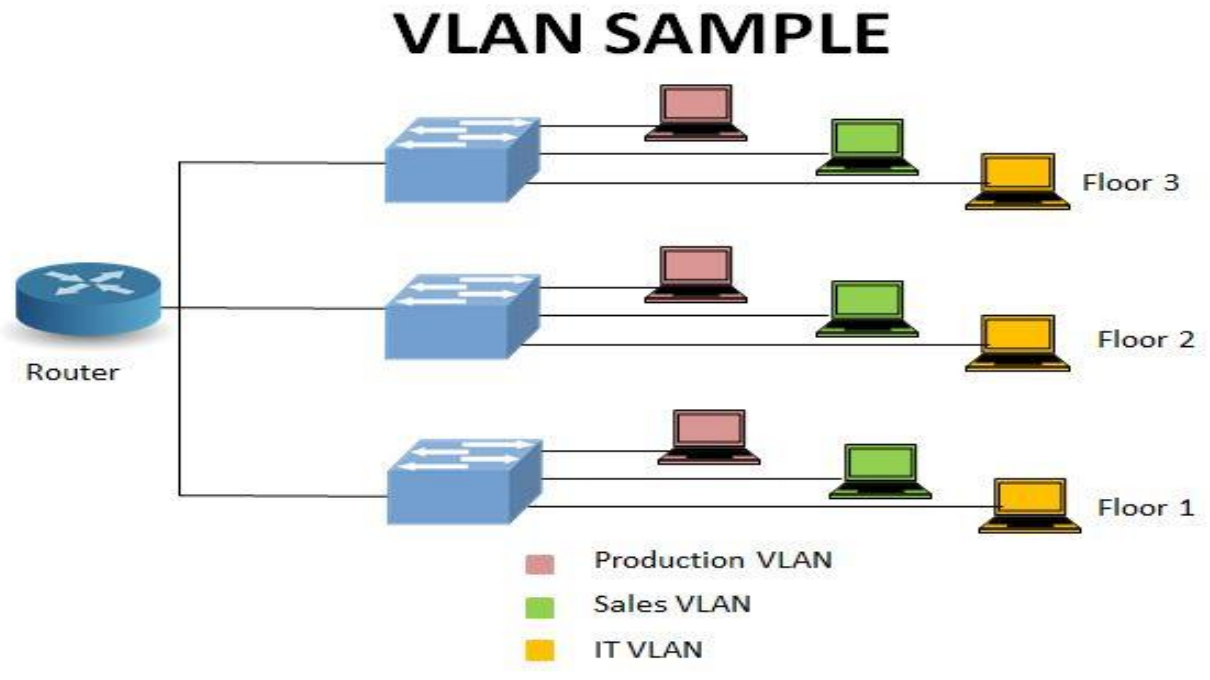
Experiment No.	: 8
Title of Experiment	: Build two virtual local area networks (VLAN) and communicate them.
Date of Experiment	:

AIM:
To design and build two virtual LAN (VLAN) using cisco packet tracer and communicate them.

Theory:

Virtual Local Area Networks (VLANs) are a critical feature of modern switched networks that allow network segmentation, improving performance, security, and manageability. VLANs logically group devices on a network, even if those devices are on different physical switches. This enables devices in the same VLAN to communicate with each other while isolating traffic between different VLANs.

There are two ways to establish a virtual LAN: static and dynamic. Static: This network creation requires virtual LANs to connect to the port manually. It's the most secure way to create a virtual connection as the configurations cannot be altered without the administrator's permission.



In VLANs, broadcast traffic is confined to the specific VLAN, reducing unnecessary traffic across the network. VLANs provide better control over network resources, improved security by restricting access, and enhanced performance by segmenting broadcast domains. Typically, each VLAN is associated with a unique VLAN ID , and devices in different VLANs communicate through a router, often using Inter-VLAN Routing .

In this task, we will use Cisco Packet Tracer to create two VLANs, allowing us to segment the network and improve its performance and security.

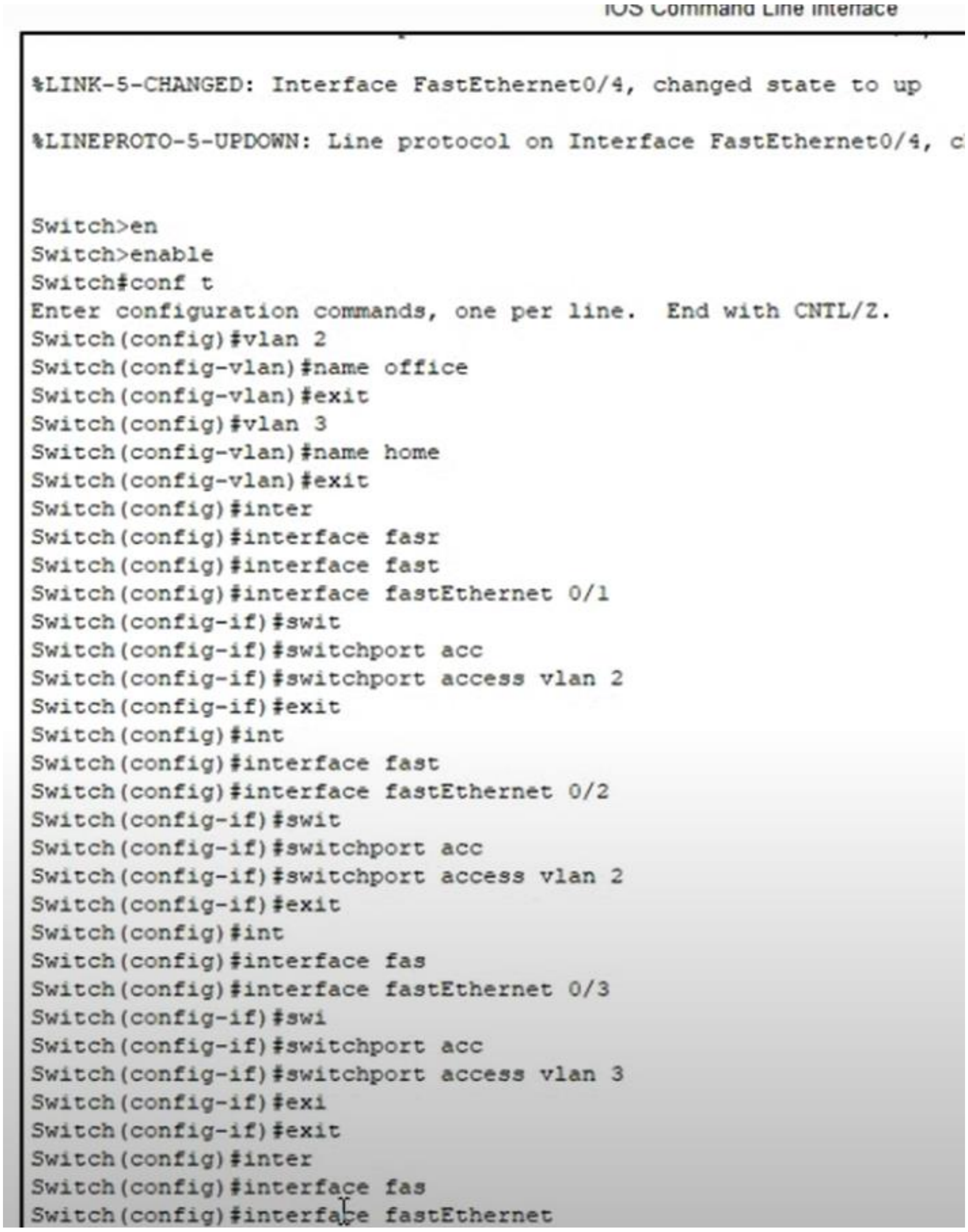
Procedure:

1. Design the Network Topology :
   - Open Cisco Packet Tracer .
   - Place a Cisco 2960 Switch on the workspace.
   - Add two PCs (e.g., PC1 and PC2) to represent devices in the first VLAN, and two additional PCs (PC3 and PC4) to represent devices in the second VLAN.
   - Connect all PCs to the switch using Copper Straight-Through cables.

2. Access the Switch :
   - Click on the Switch to open the CLI (Command Line Interface).

3. Create VLANs :
   - Enter global configuration mode on the switch and create two VLANs (e.g., VLAN 10 and VLAN 20):

```bash
Switch> enable
Switch# configure terminal
Switch(config)# vlan 10
Switch(config-vlan)# name VLAN_10
Switch(config-vlan)# exit
Switch(config)# vlan 20
Switch(config-vlan)# name VLAN_20
Switch(config-vlan)# exit
```



IOS Command Line Interface

```
%LINK-5-CHANGED: Interface FastEthernet0/4, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/4, c


Switch>en
Switch>enable
Switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#vlan 2
Switch(config-vlan)#name office
Switch(config-vlan)#exit
Switch(config)#vlan 3
Switch(config-vlan)#name home
Switch(config-vlan)#exit
Switch(config)#inter
Switch(config)#interface fasr
Switch(config)#interface fast
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#swit
Switch(config-if)#switchport acc
Switch(config-if)#switchport access vlan 2
Switch(config-if)#exit
Switch(config)#int
Switch(config)#interface fast
Switch(config)#interface fastEthernet 0/2
Switch(config-if)#swit
Switch(config-if)#switchport acc
Switch(config-if)#switchport access vlan 2
Switch(config-if)#exit
Switch(config)#int
Switch(config)#interface fas
Switch(config)#interface fastEthernet 0/3
Switch(config-if)#swi
Switch(config-if)#switchport acc
Switch(config-if)#switchport access vlan 3
Switch(config-if)#exi
Switch(config-if)#exit
Switch(config)#inter
Switch(config)#interface fas
Switch(config)#interface fastEthernet
```

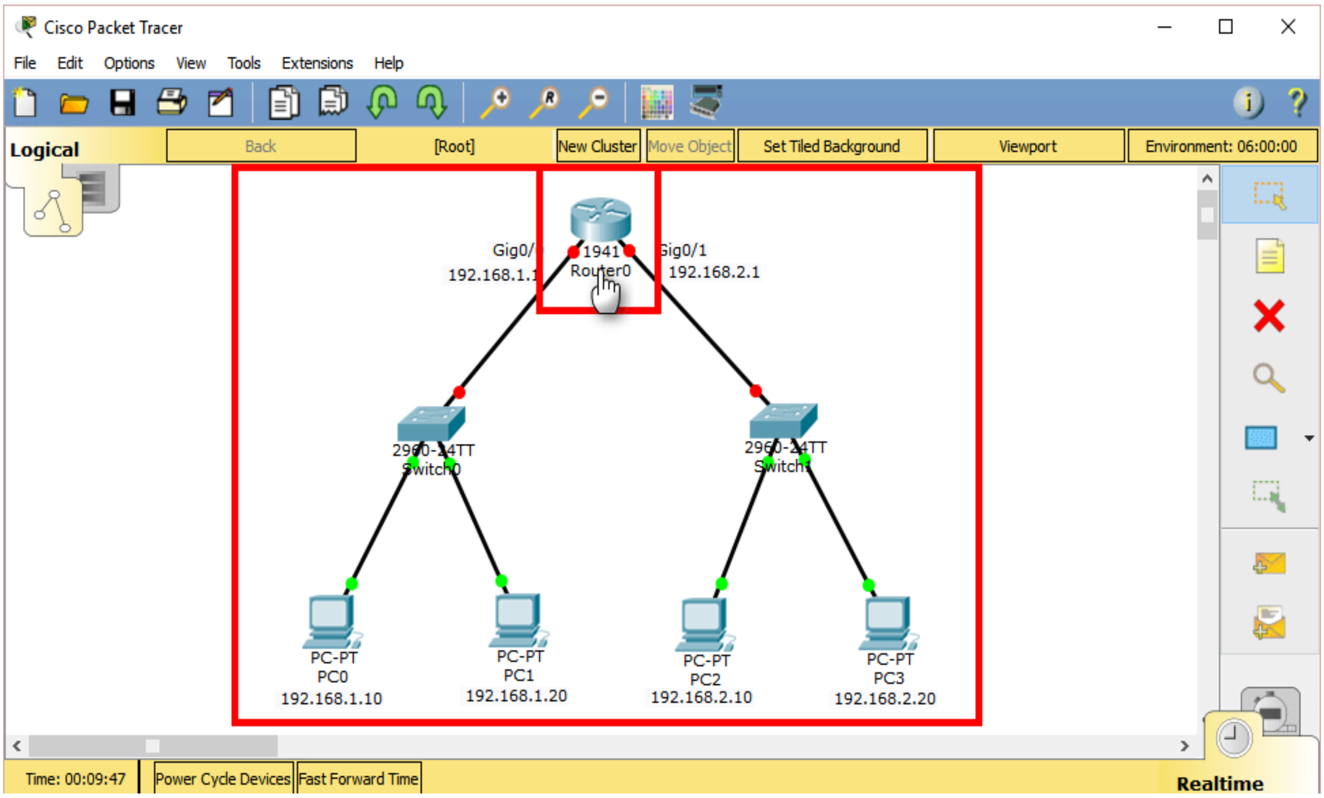
4. Assign Ports to VLANs :

- Assign the switch ports connected to PC1 and PC2 to VLAN 10 :

```bash
Switch(config)# interface range fastEthernet 0/1 - 2
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 10
Switch(config-if-range)# exit
```

- Assign the switch ports connected to PC3 and PC4 to VLAN 20 :

```bash
Switch(config)# interface range fastEthernet 0/3 - 4
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 20
Switch(config-if-range)# exit
```



5. Verify VLAN Configuration :

- Use the `show vlan brief` command to verify that the ports have been correctly assigned to the appropriate VLANs:

```bash
Switch# show vlan brief
```

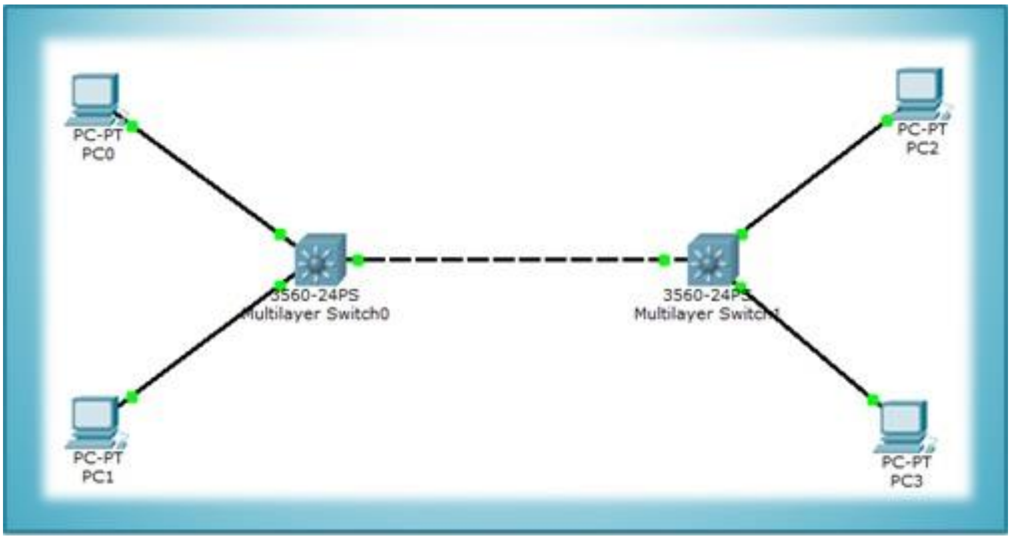
- The output should show VLAN 10 with ports Fa0/1 and Fa0/2 , and VLAN 20 with ports Fa0/3 and Fa0/4 .

6. Assign IP Addresses to PCs :

- Click on each PC and assign an IP address manually:
  - PC1 (VLAN 10) : IP address: 192.168.10.2, Subnet mask: 255.255.255.0
  - PC2 (VLAN 10) : IP address: 192.168.10.3, Subnet mask: 255.255.255.0
  - PC3 (VLAN 20) : IP address: 192.168.20.2, Subnet mask: 255.255.255.0
  - PC4 (VLAN 20) : IP address: 192.168.20.3, Subnet mask: 255.255.255.0

7. Test Connectivity :

- To ensure that the VLANs are isolated, use the `ping` command from PC1 to PC2 (both in VLAN 10). The ping should be successful.
- Ping PC3 from PC1 . The ping should fail , as PC1 is in VLAN 10 and PC3 is in VLAN 20 , and there is no routing between VLANs.

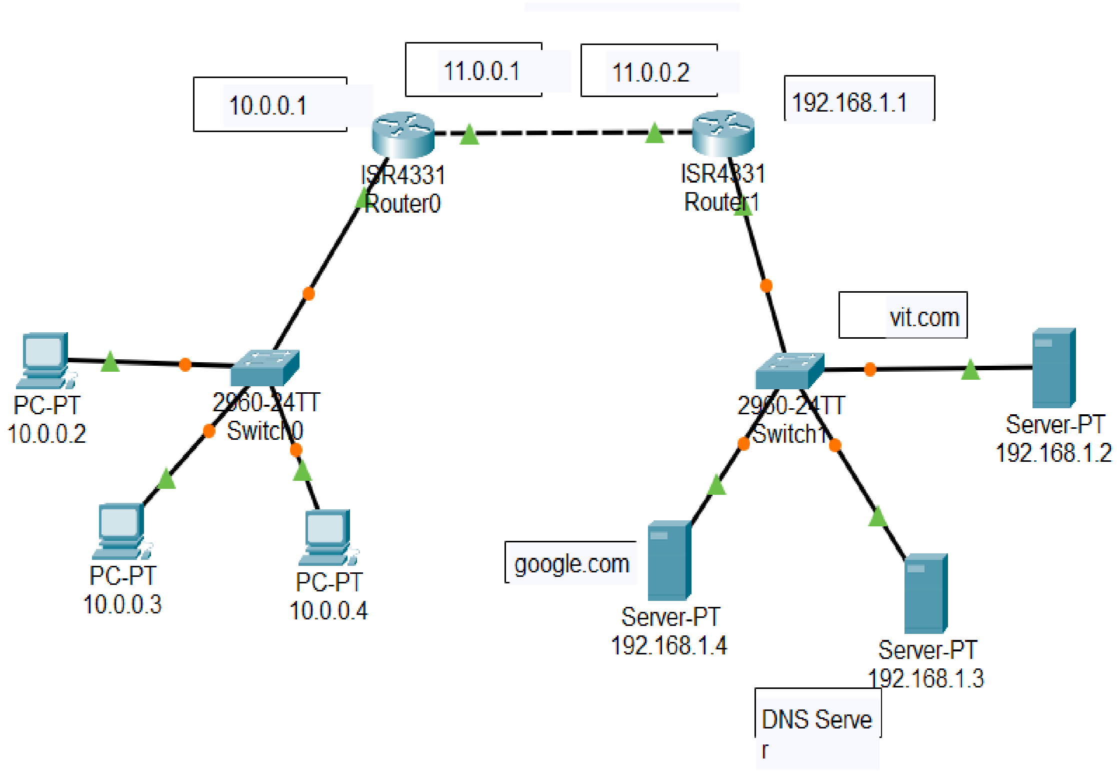


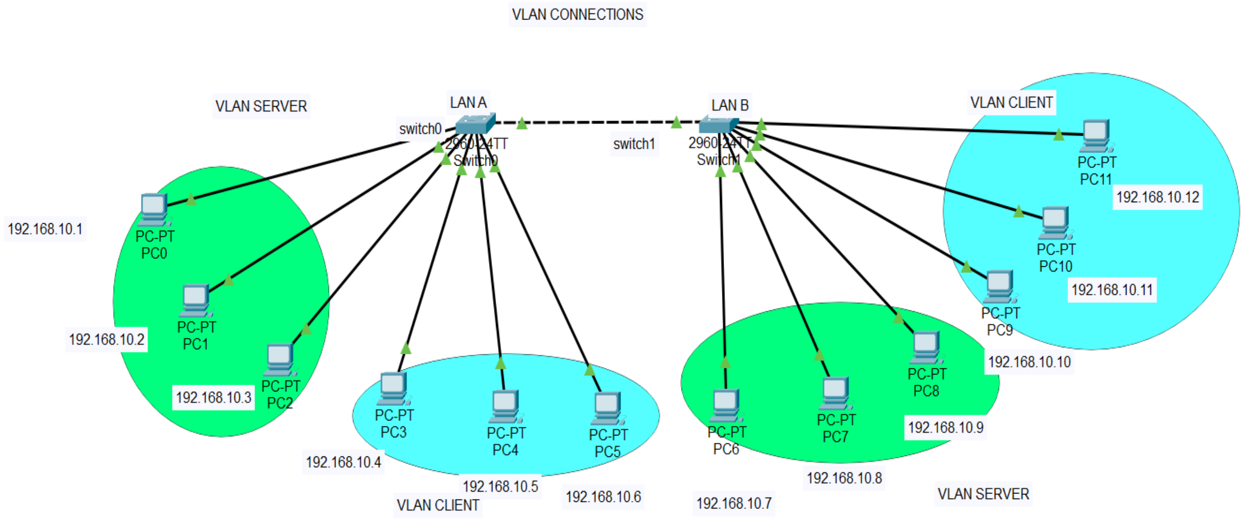
Expected Outcome:

- Devices in the same VLAN (such as PC1 and PC2 in VLAN 10) will be able to communicate with each other.
- Devices in different VLANs (such as PC1 in VLAN 10 and PC3 in VLAN 20) will not be able to communicate, as VLANs are isolated from each other by default.

Expected Results:

1. PC1 can successfully ping PC2 , confirming communication within VLAN 10 .
2. PC3 can successfully ping PC4 , confirming communication within VLAN 20 .
3. PC1 cannot ping PC3 , showing that the VLANs are isolated from each other without routing.





Viva Questions:

1. What is the purpose of VLANs in a network, and how do they improve security and performance?

2. Explain the difference between access ports and trunk ports in a switch.

3. How would you configure Inter-VLAN Routing to allow communication between different VLANs?

4. What happens if two devices in different VLANs attempt to communicate without proper routing?

5. How does the creation of VLANs reduce broadcast traffic in a network?

Results:

Thus the design of two virtual LAN (VLAN) using cisco packet tracer are build and communicated them successfully.

Experiment No.	: 9
Title of Experiment	: Configuration of DHCP, DNS and Web Server.
Date of Experiment	:

AIM:

To configure a DHCP server in Cisco Packet Tracer to dynamically allocate IP addresses to clients in a network.

Software Required:

- Cisco Packet Tracer (version 7.2 or later recommended)

Theory:

DHCP (Dynamic Host Configuration Protocol):

- DHCP is a network management protocol used to automate the process of assigning IP addresses to devices (hosts) on a network.
- Instead of manually assigning static IP addresses, DHCP dynamically allocates IP addresses from a pre-defined range.
- The DHCP server provides IP addresses, subnet masks, default gateways, and DNS information to the clients.
- The primary steps of DHCP are Discover, Offer, Request, and Acknowledge (DORA).

DHCP Process:

1. Discover: Client broadcasts a DHCPDISCOVER message to find available DHCP servers.
2. Offer: DHCP servers respond with a DHCPOFFER message offering IP configuration.
3. Request: Client requests the offered configuration via a DHCPREQUEST message.
4. Acknowledge: Server sends DHCPACK to confirm the lease of the IP address.

Procedure for Configuring DHCP in Cisco Packet Tracer:

Step 1: Network Setup

- Add the devices:
    - Drag and drop the following devices onto the workspace:
        - 1 Router (acting as the DHCP server)
        - 1 Switch
        - 2 or more PCs (to act as DHCP clients)

Step 2: Basic IP Configuration

- Router Setup:
    - Configure the IP address for the router’s interface connected to the switch (use a static IP for the router).
    - Example IP: 192.168.1.1 with a subnet mask of 255.255.255.0.
- PCs Setup:
    - Do not assign static IPs to the PCs. We will configure them to obtain IP addresses dynamically from the DHCP server.

Step 3: Configuring the DHCP Server on the Router

1. Access the CLI of the Router:
    - Click on the router and go to the CLI tab.
    - Enter privileged EXEC mode by typing:

bash

Copy code

```
enable
```

2. Enter Global Configuration Mode:

Copy code

```
configure terminal
```

3. Define the DHCP IP Address Pool:
    - Define the IP address range for clients:

Copy code

```
ip dhcp pool LAN
```

4. Set the Network and Subnet:
    - Assign the network and the subnet for the DHCP pool:

Copy code

```
network 192.168.1.0 255.255.255.0
```

5. Set the Default Gateway (Router’s interface IP):

arduino

Copy code

```
default-router 192.168.1.1
```

6. Set the DNS Server (Optional):

Copy code

```
dns-server 8.8.8.8
```

7. Exclude IP Addresses (If required):
    - Reserve certain IP addresses (such as for servers or static configurations):

css

Copy code

```
ip dhcp excluded-address 192.168.1.1 192.168.1.10
```

8. End and Save Configuration:
    - Exit configuration mode and save the changes:

arduino

Copy code

```
end
write memory
```

Step 4: Configure PCs to Obtain IP via DHCP

- Click on each PC, then go to Desktop > IP Configuration.
- Choose DHCP (instead of Static).

Step 5: Verify the DHCP Operation

- On the PCs, once DHCP is selected, the IP address, subnet mask, default gateway, and DNS server should be automatically populated.

Step 6: Test the Network Connectivity

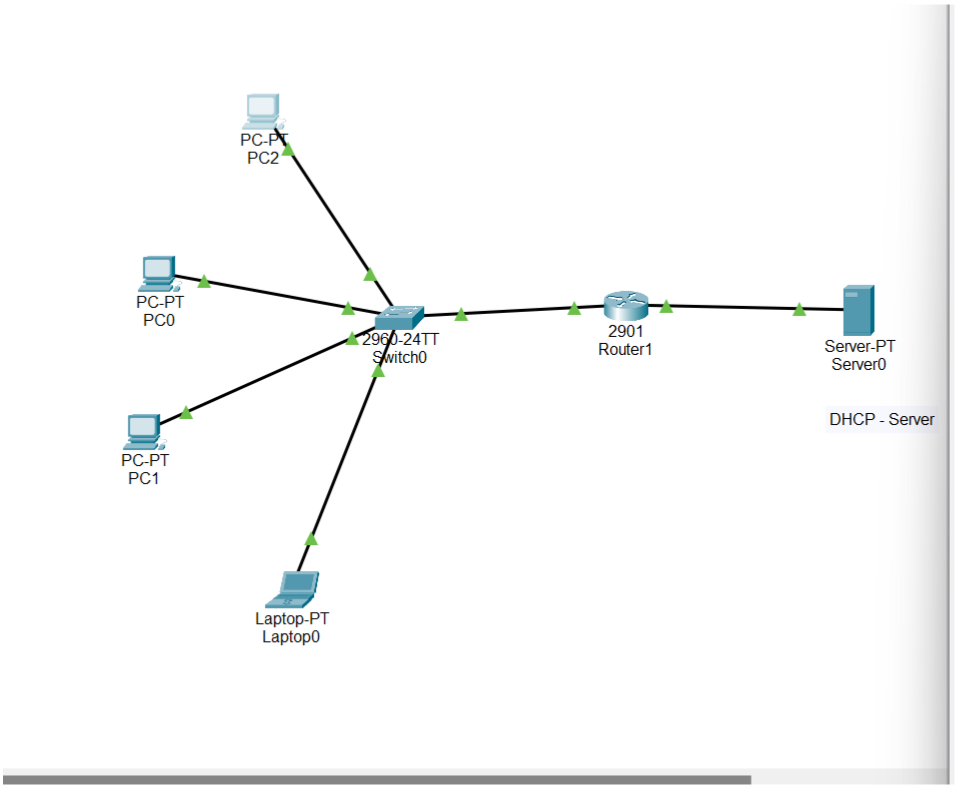
- On a PC, open the Command Prompt and use the ipconfig command to verify the assigned IP address.
- Test connectivity using the ping command:

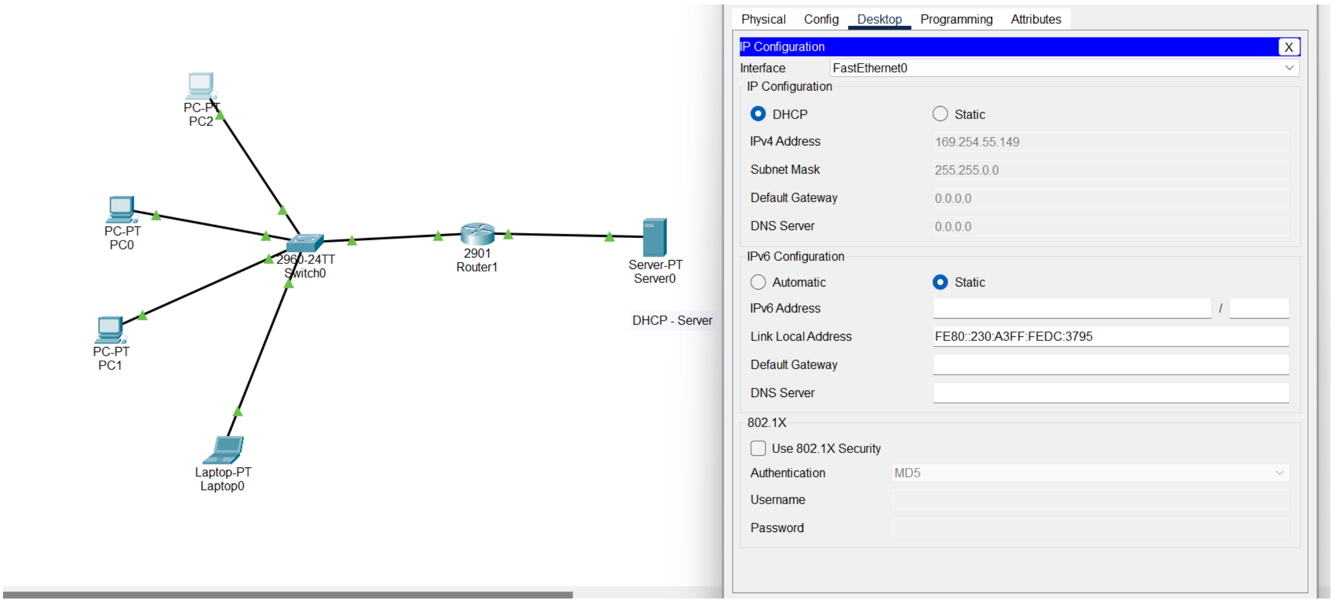
Copy code

```
ping 192.168.1.1
```

Verification and Observation:

- Check if each PC receives a unique IP address from the DHCP server.
- Ensure network connectivity between the PCs and the router using the ping command.
- Observe the automatic IP assignment process as the clients request addresses from the DHCP server.





IP Configuration

Interface: FastEthernet0

IP Configuration

● DHCP ○ Static

IPv4 Address: 169.254.55.149

Subnet Mask: 255.255.0.0

Default Gateway: 0.0.0.0

DNS Server: 0.0.0.0

IPv6 Configuration

○ Automatic ● Static

IPv6 Address: /

Link Local Address: FE80::230:A3FF:FEDC:3795

Default Gateway:

DNS Server:

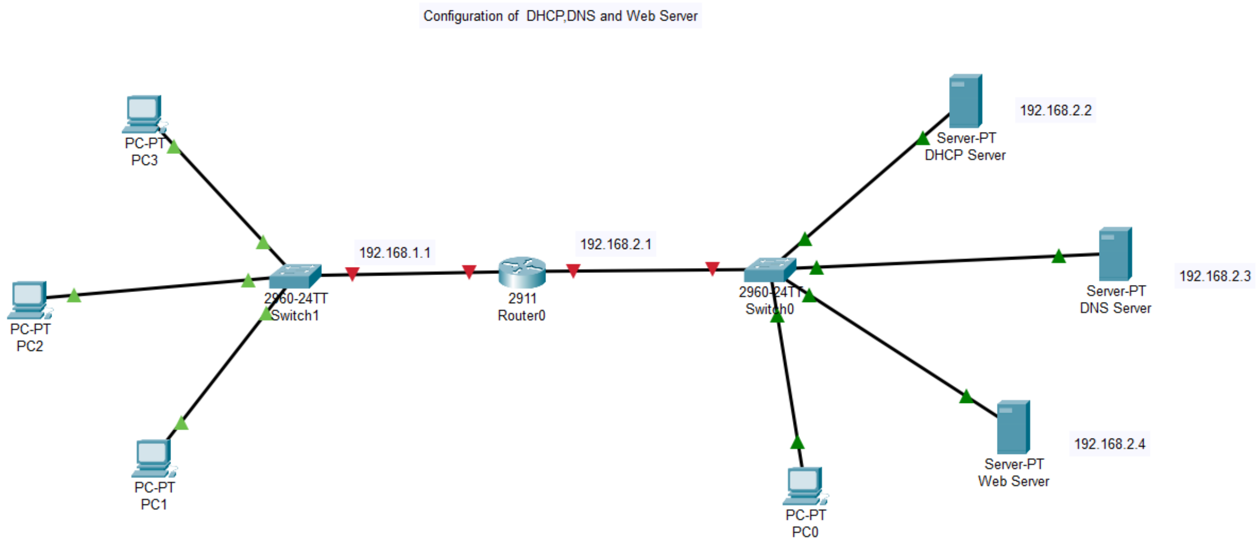
802.1X

□ Use 802.1X Security

Authentication: MD5

Username:

Password:



VIVA Questions

1. What is the main function of DHCP in a network, and how does it differ from static IP addressing?

2. Describe the DORA process in DHCP and the role of each step.

3. What information does a DHCP server provide to a client besides the IP address?

4. How does the DHCP lease mechanism work, and why is it important?

5. What is the purpose of excluding certain IP addresses in a DHCP configuration?

Results:
By configuring the DHCP server on the router, IP addresses are dynamically assigned to the clients, demonstrating how DHCP simplifies network management by automating IP address allocation. Hence DHCP is verified

Experiment No.	: 9)b)
Title of Experiment	: Configuration of DNS.
Date of Experiment	:

AIM:

To simulate DNS on server using Cisco packet tracer

Theory:

The Domain Name System (DNS) is an essential component of the internet that translates human-readable domain names (like `www.example.com`) into machine-readable IP addresses. This translation is crucial because while humans prefer using domain names, computers and network devices communicate using numerical IP addresses. DNS operates as a hierarchical and distributed system, consisting of several layers: the root, top-level domains (TLDs) like `.com`, `.org`, and second-level domains (SLDs) like `example` in `example.com`. When a user enters a domain name into a browser, a DNS query is initiated. This query is first sent to a recursive DNS resolve, which may have cached the required information from a previous request. If not, the resolver queries a root DNS server , which points to the appropriate TLD server (e.g., `.com`). The TLD server then directs the query to the authoritative DNS server responsible for the specific domain, which provides the corresponding IP address. This process enables the user's device to connect to the target server. DNS stores different types of records to manage various network functions. The most common is the A record , which maps a domain to an IPv4 address, while AAAA records map to IPv6 addresses. Other important records include CNAME records , which alias one domain to another, and MX records , which specify mail servers for the domain.

DNS caching, which stores resolved queries for a set duration (defined by the Time-to-Live (TTL) value), improves network performance by reducing repeated lookups. In lab experiments, configuring and troubleshooting DNS is critical for understanding how network communication is facilitated. Proper DNS setup ensures smooth internet connectivity, making it an important concept in networking studies.

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 3PCs and drop them on the screen (PC0, PC1, PC2).

3. Select "Switch 2960-24T logo" from the Network Devices section. Drag 1 switch and dropit 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. TTP Configuration: Inside Services section in the server, turn on HTTP. Edit the index.html file and add your content (for eg.: VIT Chennai)

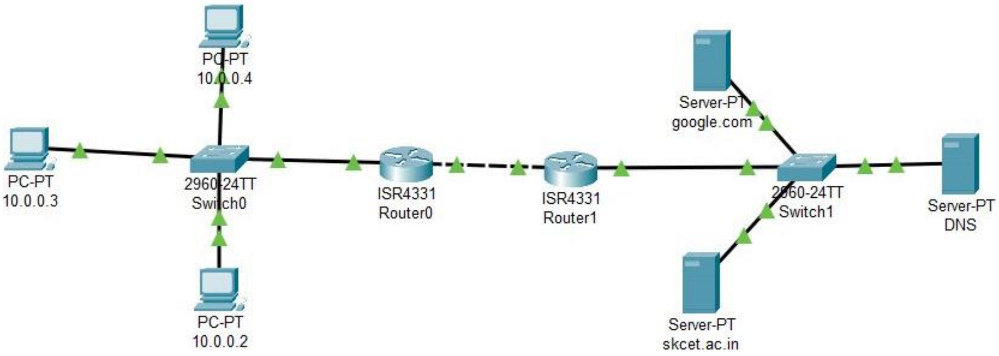
18. Add a server for google.com with IP address 192.168.1.4 and add the google page source content.

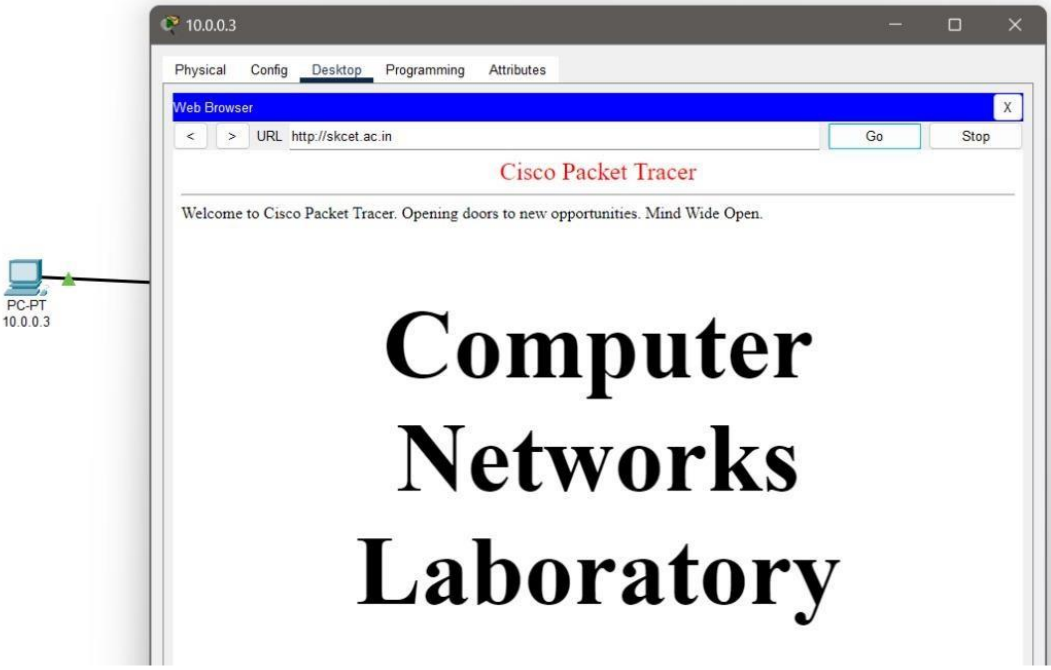
19. DNS Configuration: Add one more server for DNS in the second network. Add its IP address as 192.168.1.3 and add it in the DNS Server section too. In the services section, click on DNS and turn it on. Add the web page name for eg: vit.com along with its IP address (192.168.1.2) and google.com along 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 SCREENSHOTS:



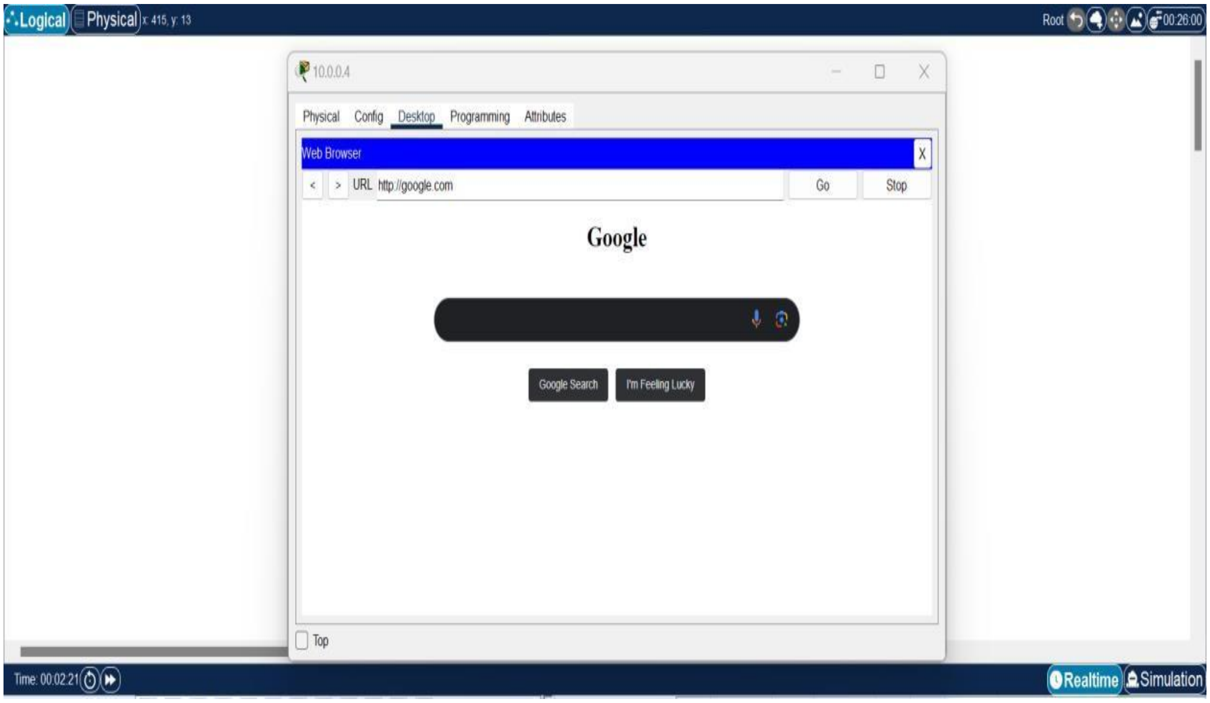


http://skcet.ac.in

Cisco Packet Tracer

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

Computer Networks Laboratory



VIVA Questions

1) What is the purpose of the DNS and how does it work in a network?

2) Explain the difference between an A record and a CNAME record in DNS.

3) What is DNS caching and how does it improve network performance?

4) Describe the role of a recursive DNS resolver in the DNS lookup process.

5) How do DNS zones differ from domains in a DNS hierarchy?

RESULT:

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

Experiment No.	: 9)C
Title of Experiment	: Webserver- HTTPS Protocols
Date of Experiment	:

AIM :

To simulate http protocol using Cisco packet tracer

THEORY

A web server is a software system that hosts websites and delivers web pages to users through the internet. It listens for incoming requests from clients, typically web browsers, and responds by providing the requested content, such as HTML files, images, or other resources. Web servers use protocols like HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure) to communicate with clients. When a user enters a URL in a browser, the browser sends a request to the web server, which processes it and returns the appropriate content. Popular web server software includes Apache, Nginx, and Microsoft's IIS.

HTTP, the fundamental protocol for web communication, operates over the TCP/IP suite and defines how messages are formatted and transmitted. HTTP is stateless, meaning each request from a client to the server is independent and carries no information about previous interactions. However, HTTP itself lacks encryption, meaning data transmitted between the client and server is susceptible to eavesdropping, man-in-the-middle attacks, and data tampering.

To address security concerns, HTTPS was developed, which is an extension of HTTP. HTTPS uses Transport Layer Security (TLS) or its predecessor Secure Sockets Layer (SSL) to encrypt communication between the client and the server. With HTTPS, sensitive data like login credentials, payment information, or personal details are securely transmitted, ensuring confidentiality, integrity, and authentication. The encryption ensures that even if the data is intercepted, it cannot be read without the decryption keys. HTTPS has become the standard for secure web browsing, especially for sites handling personal information or financial transactions, and is a critical part of ensuring user trust and data protection on the internet.

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 3PCs and drop them on the screen (PC0, PC1, PC2).

3. Select "Switch 2960-24T logo" from the Network Devices section. Drag 1 switch

and dropit 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 dropit 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.

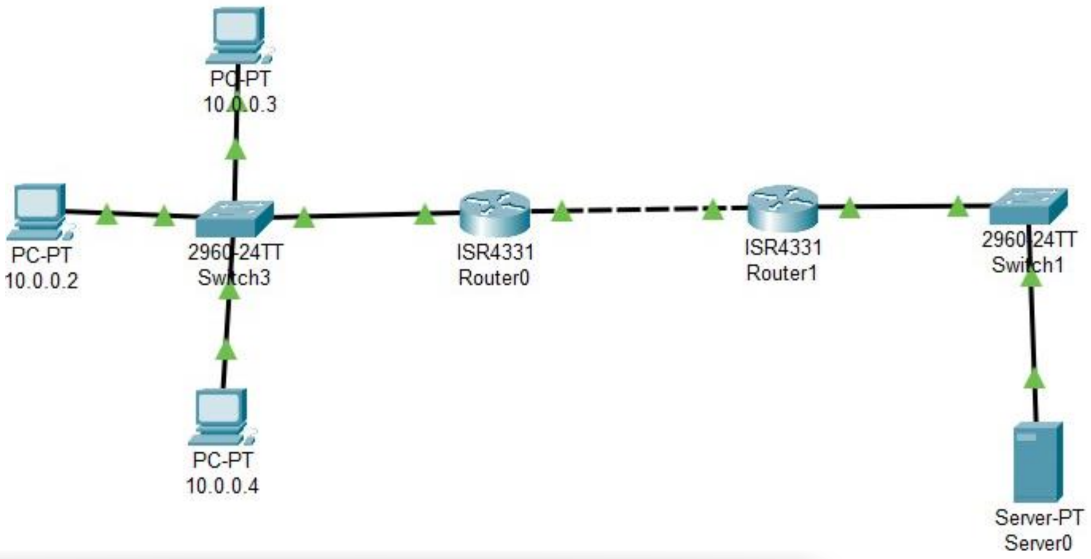
16. Dynamic routing protocol- Add 10.0.0.0, 11.0.0.0 and 192.168.1.0 in the RIP section ofboth the routers.

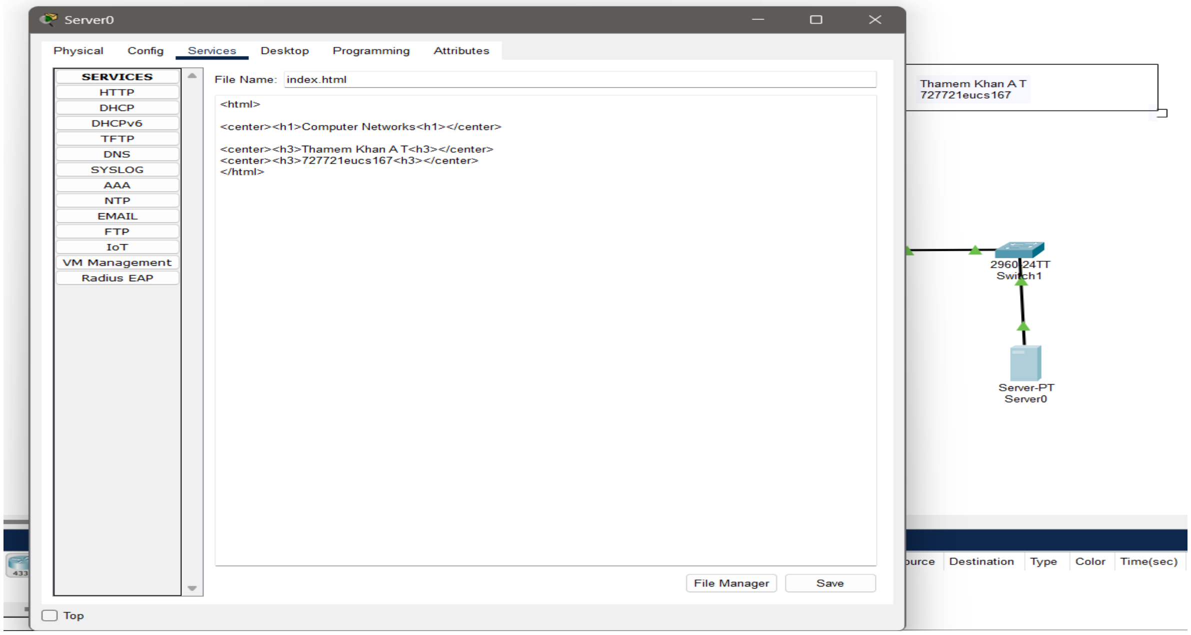
17. HTTP Configuration: Inside Services section in the server, turn on HTTP. Edit theindex.html file and add your content.

18. Check for the web page in the web browser section of any PC by searching for the serverIP address 192.168.1.2

Network Diagram:

HTTP Configuration index.html file:

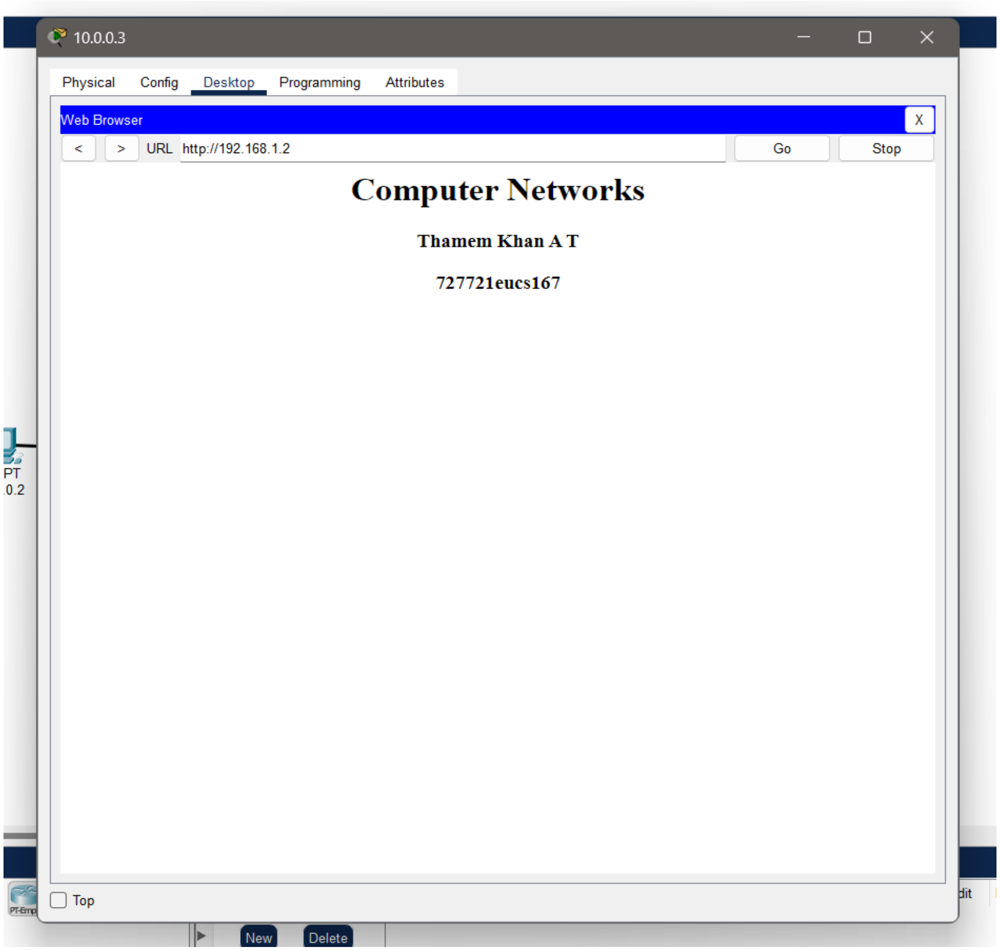




File Name: index.html

```
<html>

<center><h1>Computer Networks<h1></center>

<center><h3>Thamem Khan A T<h3></center>
<center><h3>727721eucs167<h3></center>
</html>
```

OUTPUT: Searching IP address from PC0:



URL http://192.168.1.2

Computer Networks

Thamem Khan A T

727721eucs167

RESULT:

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

Experiment No.	: 10
Title of Experiment	: Implement a home or small business network using wireless technology, then connect it to the Internet
Date of Experiment	:

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 buttonon 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 Providethe 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.

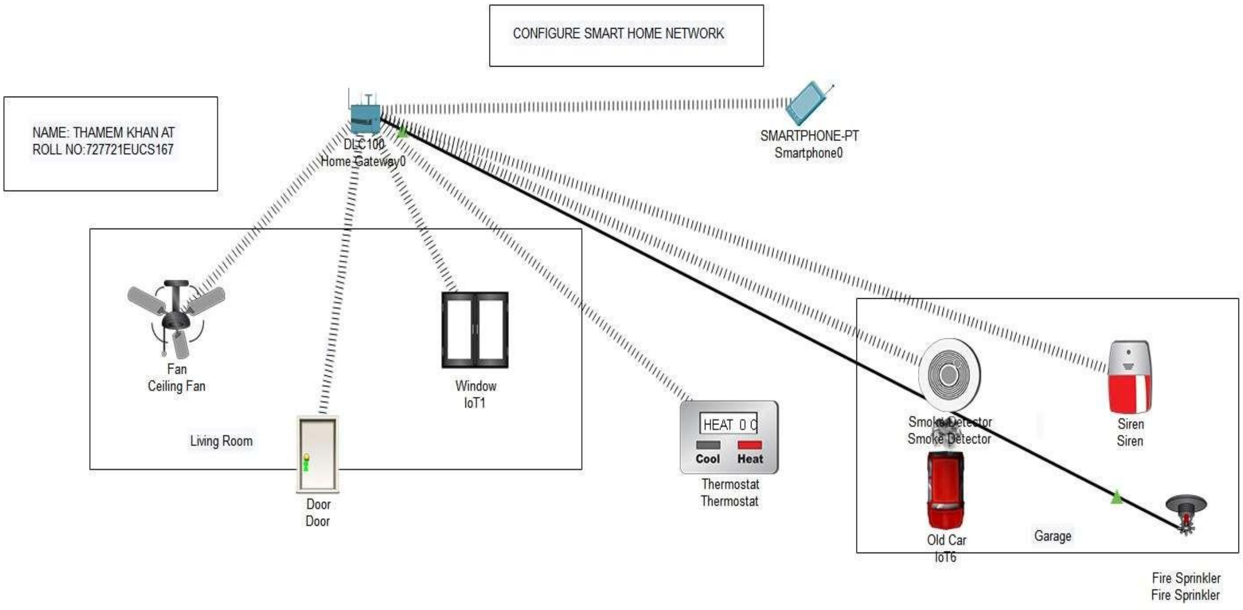
18. Now an old car is chosen to produce smoke effect.

19. Then the conditions are provided in the in IOT Monitor of the smartphone to monitor the devices.

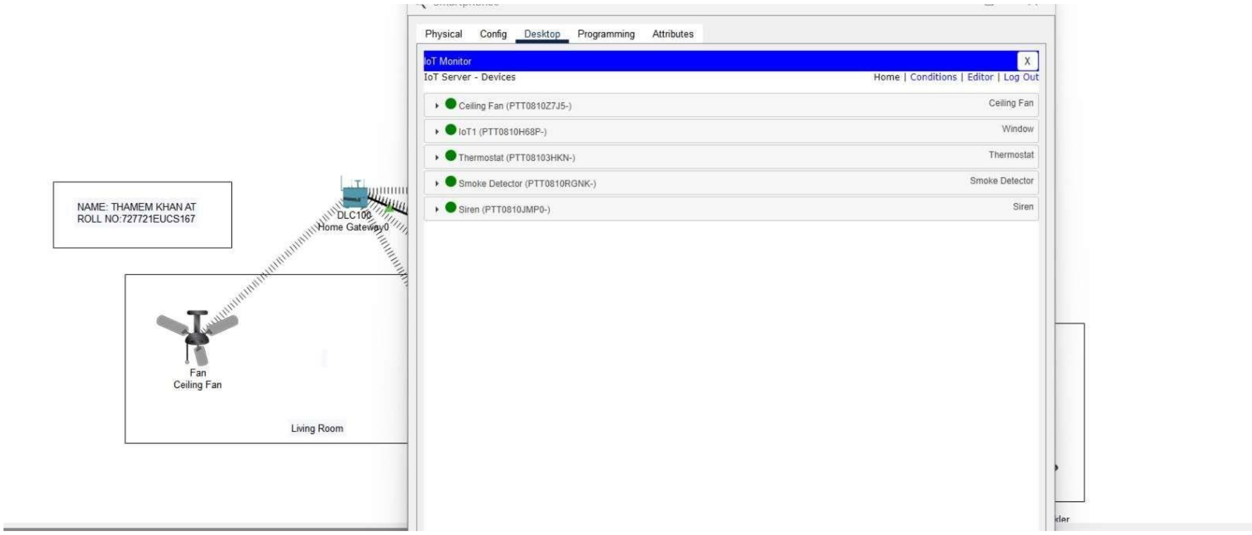
20. Provide conditions like if the smoke detector detects the smoke level > 10 , then the siren alarm is on. Also the fire sprinkler is turned on.

21. Check those by creating smoke level according to the provided conditions.

NETWORK DIAGRAM :



Devices connected

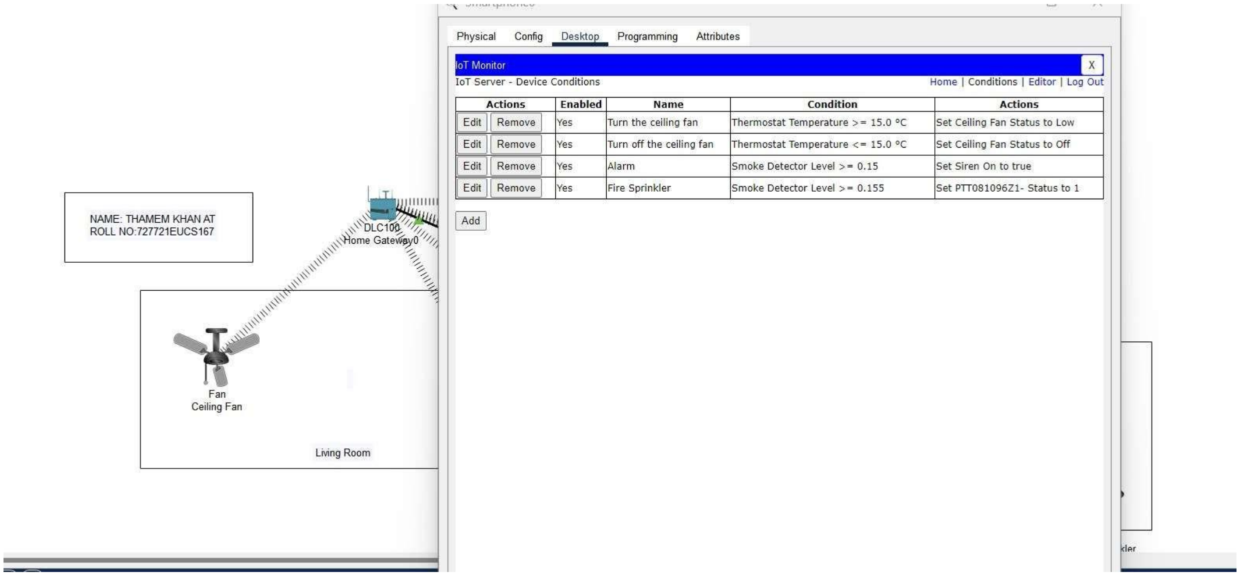


NAME: THAMEM KHAN AT
ROLL NO:727721EUCS167

IoT Server - Devices

- ● Ceiling Fan (PTT0810Z7J5-) Ceiling Fan
- ● IoT1 (PTT0810H68P-) Window
- ● Thermostat (PTT08103HKN-) Thermostat
- ● Smoke Detector (PTT0810RGNK-) Smoke Detector
- ● Siren (PTT0810JMP0-) Siren

Conditions provided



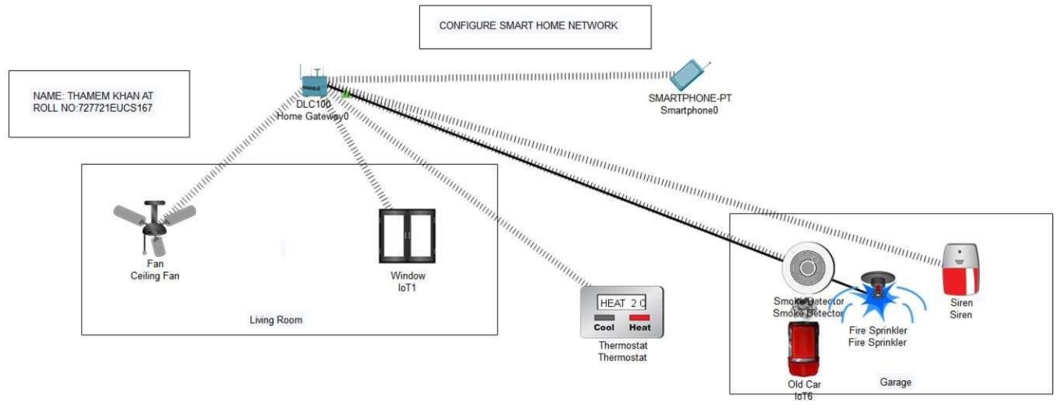
NAME: THAMEM KHAN AT
ROLL NO:727721EUCS167

IoT Server - Device Conditions

Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	Turn the ceiling fan	Thermostat Temperature >= 15.0 °C	Set Ceiling Fan Status to Low
Edit Remove	Yes	Turn off the ceiling fan	Thermostat Temperature <= 15.0 °C	Set Ceiling Fan Status to Off
Edit Remove	Yes	Alarm	Smoke Detector Level >= 0.15	Set Siren On to true
Edit Remove	Yes	Fire Sprinkler	Smoke Detector Level >= 0.155	Set PTT081096Z1- Status to 1

OUTPUT :

Fire Sprinkler and Alarm turned on as the smoke is detected



CONFIGURE SMART HOME NETWORK

NAME: THAMEM KHAN AT
ROLL NO:727721EUCS167

Viva Questions

1. What are the key components required to set up a wireless network for a home or small business?

2. Explain the process of configuring a wireless router for internet access in a home or small business environment.

3. How do you secure a wireless network and what are the different encryption methods available?

4. What is the role of the SSID in a wireless network, and why should it be configured properly?

5. How would you configure a wireless network to extend its range using a repeater or access point?

RESULT:

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

Experiment No.	: 11
Title of Experiment	: Study of Networks Commands
Date of Experiment	:

AIM:

To study the network commands(windows based) by using command prompt and alsofind the IP address for familiar websites.

PROCEDURE:

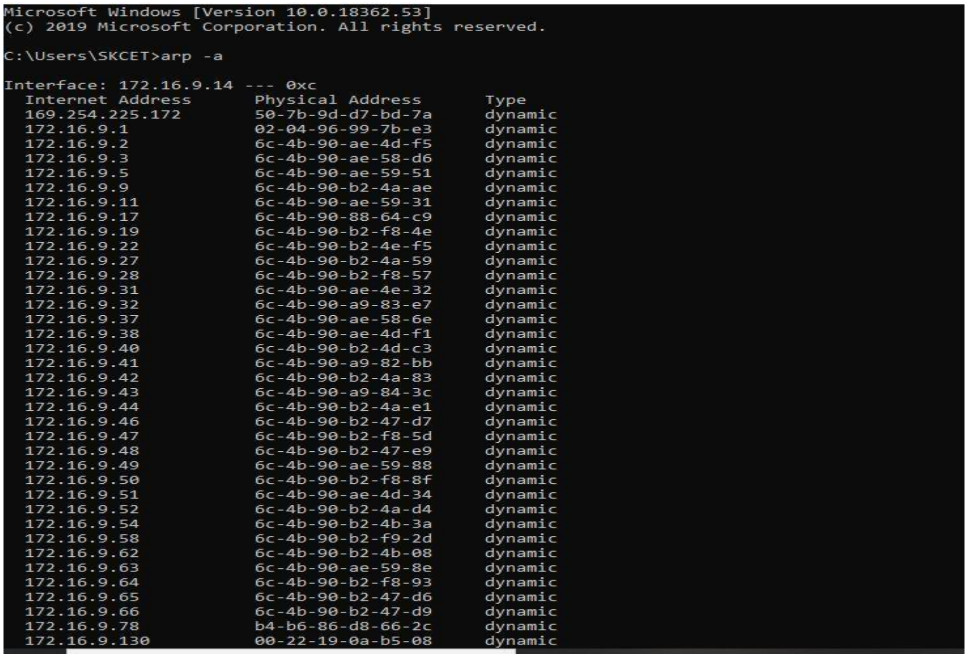
- Open a new terminal.
- Type the below command in the terminal and get the output.

COMMAND AND OUTPUT:

ii) 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:

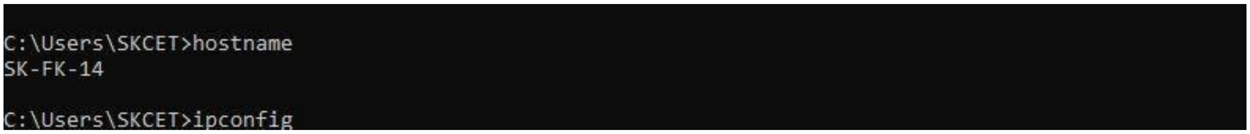


```
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\SKCET>arp -a

Interface: 172.16.9.14 --- 0xc
  Internet Address      Physical Address      Type
  169.254.225.172       50-7b-9d-d7-bd-7a     dynamic
  172.16.9.1            02-04-96-99-7b-e3     dynamic
  172.16.9.2            6c-4b-90-ae-4d-f5     dynamic
  172.16.9.3            6c-4b-90-ae-58-d6     dynamic
  172.16.9.5            6c-4b-90-ae-59-51     dynamic
  172.16.9.9            6c-4b-90-b2-4a-ae     dynamic
  172.16.9.11           6c-4b-90-ae-59-31     dynamic
  172.16.9.17           6c-4b-90-88-64-c9     dynamic
  172.16.9.19           6c-4b-90-b2-f8-4e     dynamic
  172.16.9.22           6c-4b-90-b2-4e-f5     dynamic
  172.16.9.27           6c-4b-90-b2-4a-59     dynamic
  172.16.9.28           6c-4b-90-b2-f8-57     dynamic
  172.16.9.31           6c-4b-90-ae-4e-32     dynamic
  172.16.9.32           6c-4b-90-a9-83-e7     dynamic
  172.16.9.37           6c-4b-90-ae-58-6e     dynamic
  172.16.9.38           6c-4b-90-ae-4d-f1     dynamic
  172.16.9.40           6c-4b-90-b2-4d-c3     dynamic
  172.16.9.41           6c-4b-90-a9-82-bb     dynamic
  172.16.9.42           6c-4b-90-b2-4a-83     dynamic
  172.16.9.43           6c-4b-90-a9-84-3c     dynamic
  172.16.9.44           6c-4b-90-b2-4a-e1     dynamic
  172.16.9.46           6c-4b-90-b2-47-d7     dynamic
  172.16.9.47           6c-4b-90-b2-f8-5d     dynamic
  172.16.9.48           6c-4b-90-b2-47-e9     dynamic
  172.16.9.49           6c-4b-90-ae-59-88     dynamic
  172.16.9.50           6c-4b-90-b2-f8-8f     dynamic
  172.16.9.51           6c-4b-90-ae-4d-34     dynamic
  172.16.9.52           6c-4b-90-b2-4a-d4     dynamic
  172.16.9.54           6c-4b-90-b2-4b-3a     dynamic
  172.16.9.58           6c-4b-90-b2-f9-2d     dynamic
  172.16.9.62           6c-4b-90-b2-4b-08     dynamic
  172.16.9.63           6c-4b-90-ae-59-8e     dynamic
  172.16.9.64           6c-4b-90-b2-f8-93     dynamic
  172.16.9.65           6c-4b-90-b2-47-d6     dynamic
  172.16.9.66           6c-4b-90-b2-47-d9     dynamic
  172.16.9.78           b4-b6-86-d8-66-2c     dynamic
  172.16.9.130          00-22-19-0a-b5-08     dynamic
```

iii)Command: hostname

It will display the name of the

host.Output:

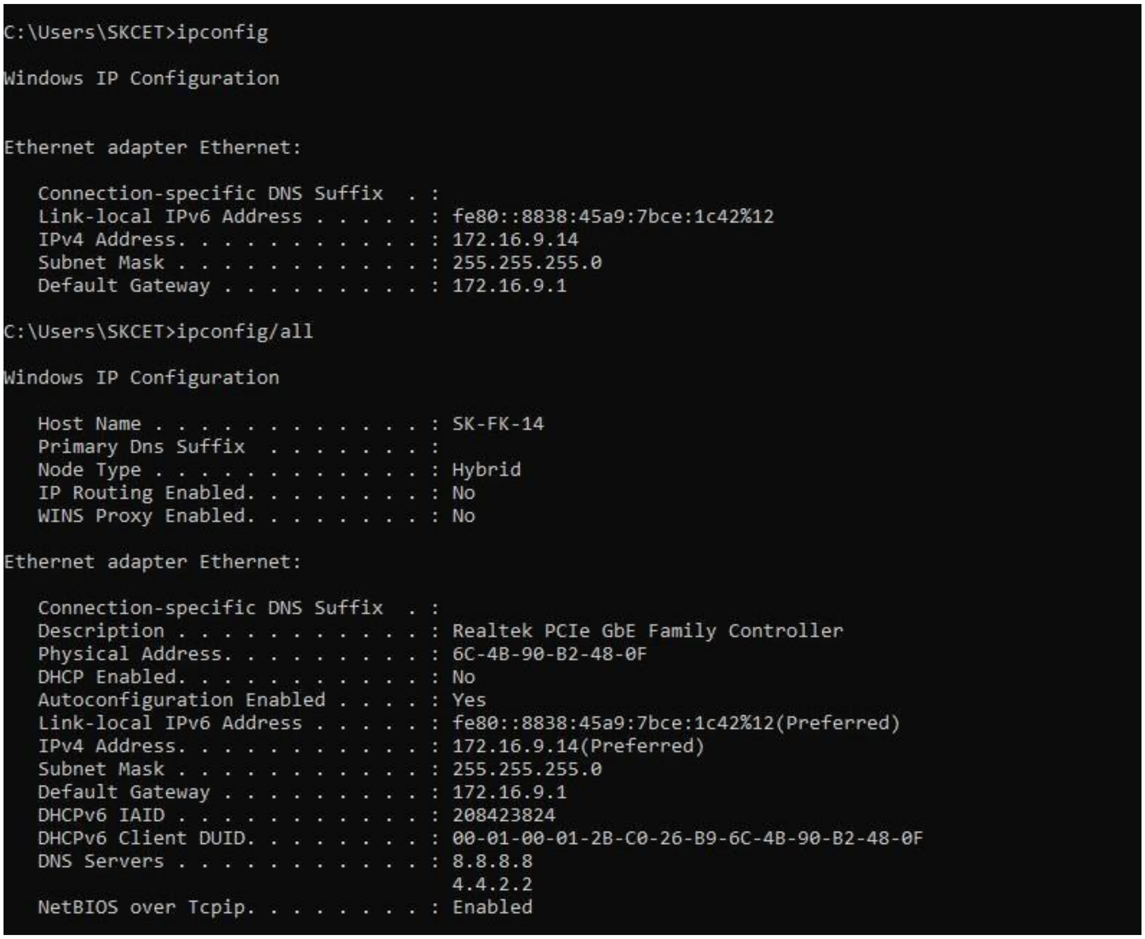


```
C:\Users\SKCET>hostname
SK-FK-14

C:\Users\SKCET>ipconfig
```

iv) 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:

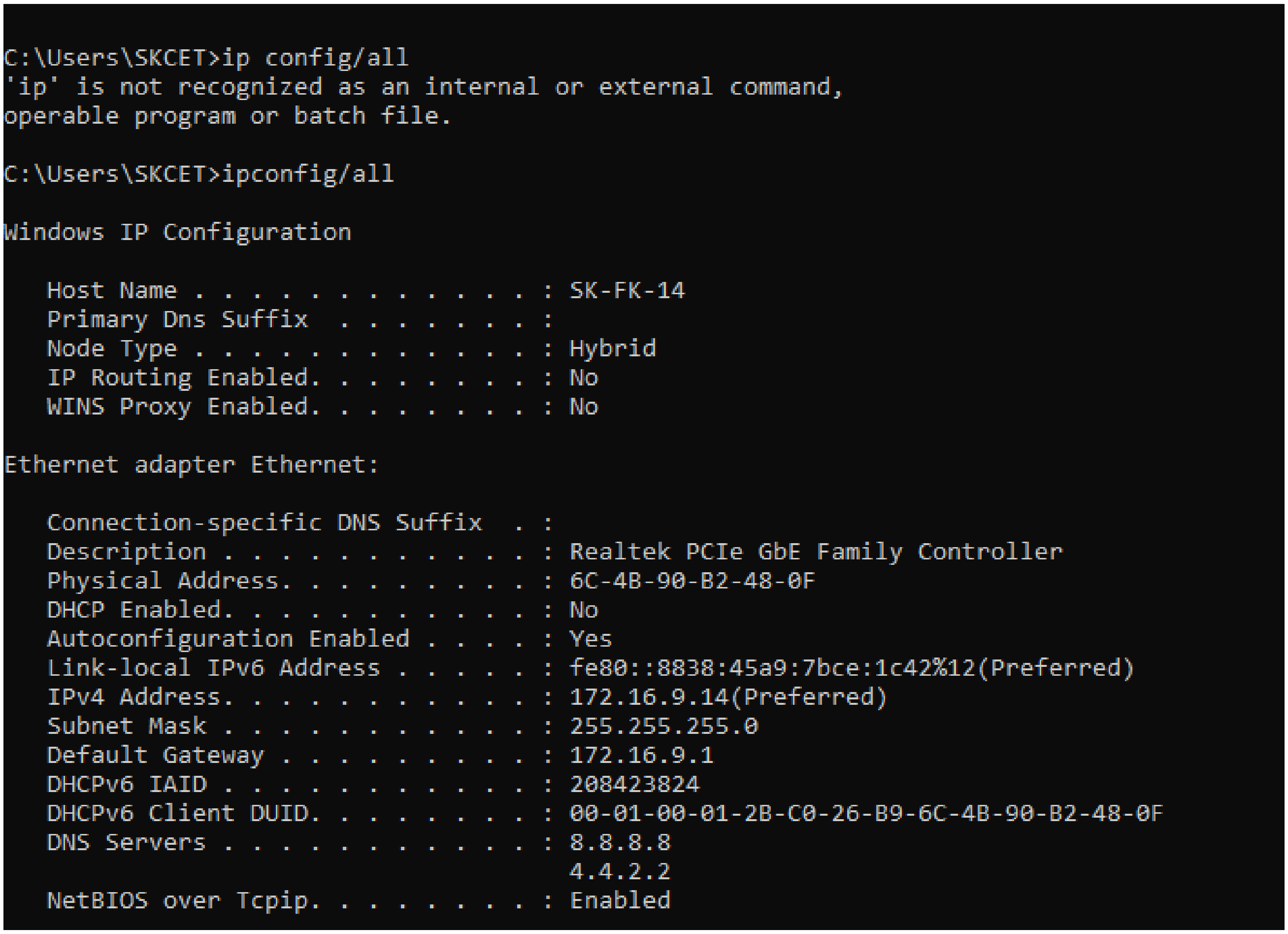


```
C:\Users\SKCET>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8838:45a9:7bce:1c42%12
   IPv4 Address. . . . . . . . . . . : 172.16.9.14
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.9.1

C:\Users\SKCET>ipconfig/all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : SK-FK-14
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physical Address. . . . . . . . . : 6C-4B-90-B2-48-0F
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::8838:45a9:7bce:1c42%12(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.9.14(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.9.1
   DHCPv6 IAID . . . . . . . . . . . : 208423824
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2B-C0-26-B9-6C-4B-90-B2-48-0F
   DNS Servers . . . . . . . . . . . : 8.8.8.8
                                       4.4.2.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

v) Command: ip config/all

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

Output:

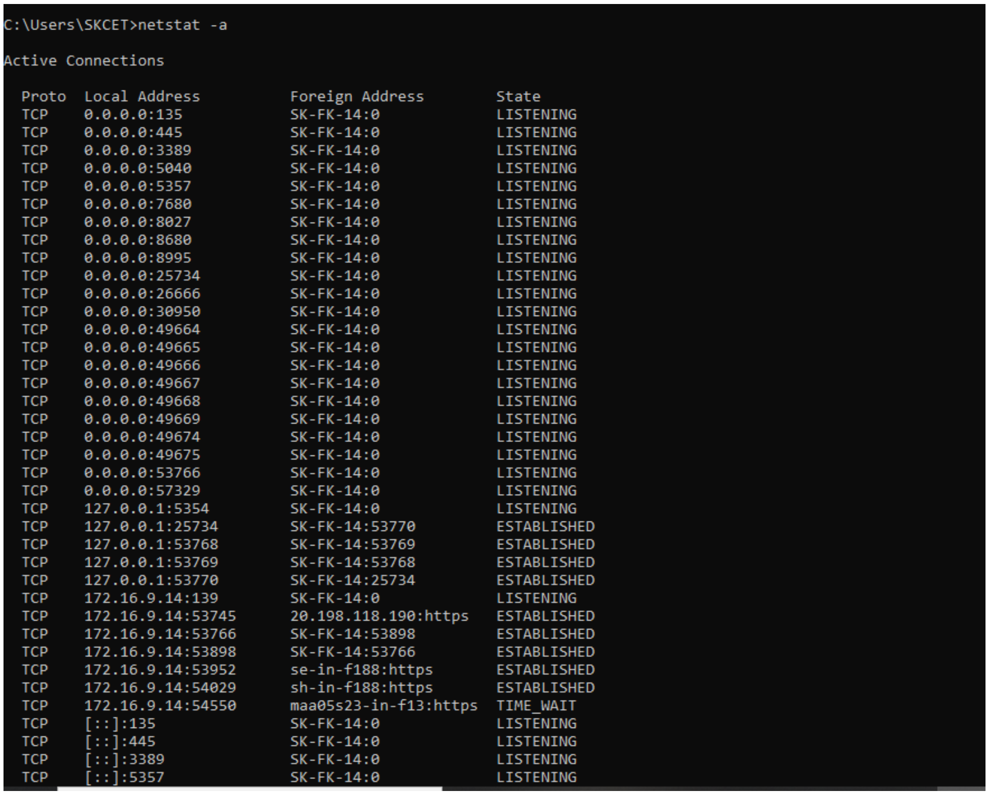


```
C:\Users\SKCET>ip config/all
'ip' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\SKCET>ipconfig/all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : SK-FK-14
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physical Address. . . . . . . . . : 6C-4B-90-B2-48-0F
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::8838:45a9:7bce:1c42%12(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.9.14(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.9.1
   DHCPv6 IAID . . . . . . . . . . . : 208423824
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2B-C0-26-B9-6C-4B-90-B2-48-0F
   DNS Servers . . . . . . . . . . . : 8.8.8.8
                                       4.4.2.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

vi)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:

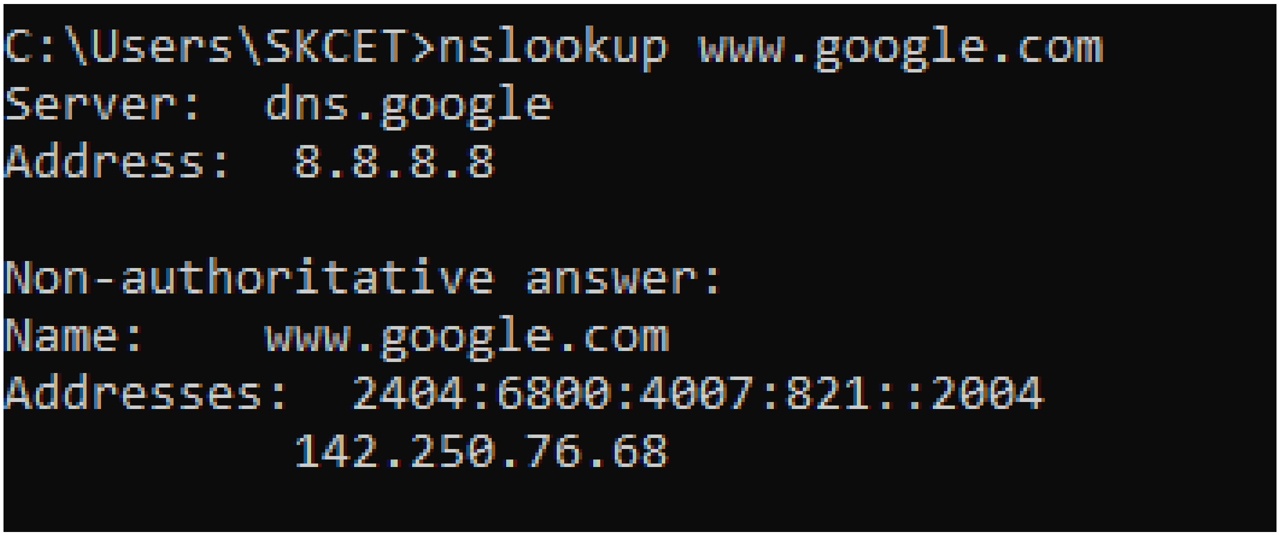


```
C:\Users\SKCET>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            SK-FK-14:0             LISTENING
  TCP    0.0.0.0:445            SK-FK-14:0             LISTENING
  TCP    0.0.0.0:3389           SK-FK-14:0             LISTENING
  TCP    0.0.0.0:5040           SK-FK-14:0             LISTENING
  TCP    0.0.0.0:5357           SK-FK-14:0             LISTENING
  TCP    0.0.0.0:7680           SK-FK-14:0             LISTENING
  TCP    0.0.0.0:8027           SK-FK-14:0             LISTENING
  TCP    0.0.0.0:8680           SK-FK-14:0             LISTENING
  TCP    0.0.0.0:8995           SK-FK-14:0             LISTENING
  TCP    0.0.0.0:25734          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:26666          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:30950          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:49664          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:49665          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:49666          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:49667          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:49668          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:49669          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:49674          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:49675          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:53766          SK-FK-14:0             LISTENING
  TCP    0.0.0.0:57329          SK-FK-14:0             LISTENING
  TCP    127.0.0.1:5354         SK-FK-14:0             LISTENING
  TCP    127.0.0.1:25734        SK-FK-14:53770         ESTABLISHED
  TCP    127.0.0.1:53768        SK-FK-14:53769         ESTABLISHED
  TCP    127.0.0.1:53769        SK-FK-14:53768         ESTABLISHED
  TCP    127.0.0.1:53770        SK-FK-14:25734         ESTABLISHED
  TCP    172.16.9.14:139        SK-FK-14:0             LISTENING
  TCP    172.16.9.14:53745      20.198.118.190:https   ESTABLISHED
  TCP    172.16.9.14:53766      SK-FK-14:53898         ESTABLISHED
  TCP    172.16.9.14:53898      SK-FK-14:53766         ESTABLISHED
  TCP    172.16.9.14:53952      se-in-f188:https       ESTABLISHED
  TCP    172.16.9.14:54029      sh-in-f188:https       ESTABLISHED
  TCP    172.16.9.14:54550      maa05s23-in-f13:https  TIME_WAIT
  TCP    [::]:135               SK-FK-14:0             LISTENING
  TCP    [::]:445               SK-FK-14:0             LISTENING
  TCP    [::]:3389              SK-FK-14:0             LISTENING
  TCP    [::]:5357              SK-FK-14:0             LISTENING
```

vii) Command: nslookup

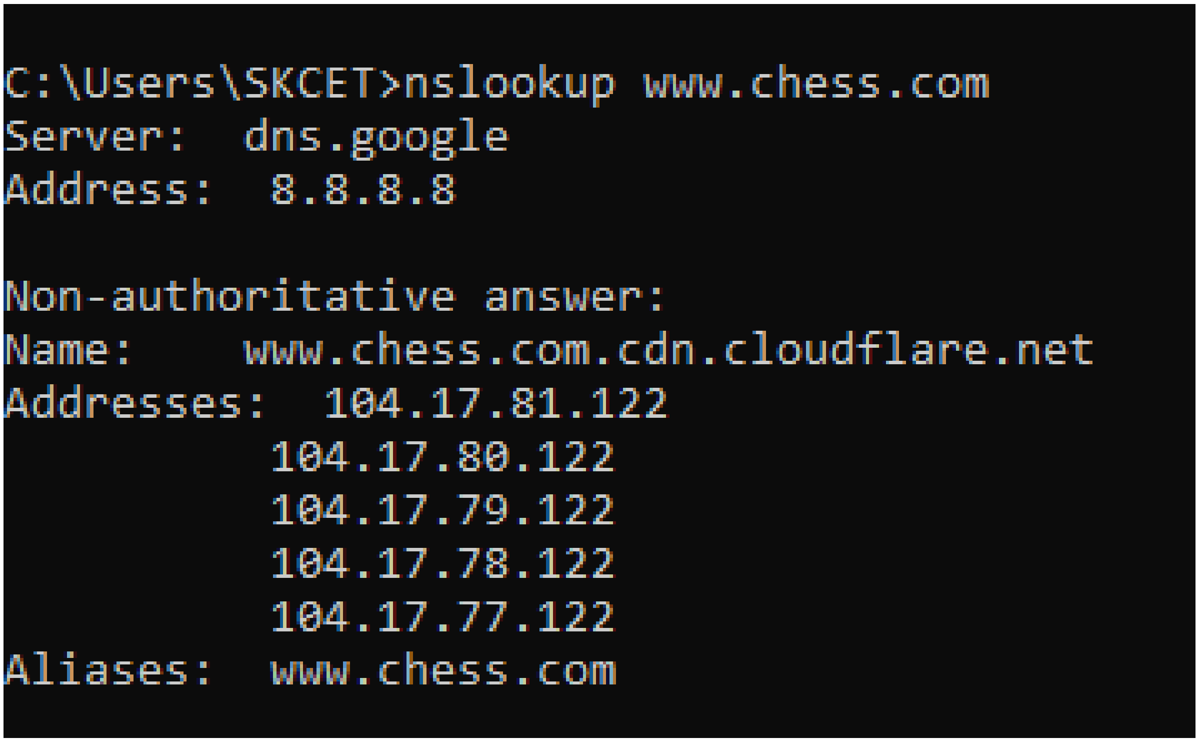
"nslookup" , which stands for "Name Server Lookup," is a command-line tool used forquerying DNS (Domain Name System) servers to obtain domain name or IP address information. It is available on various operating systems, including Windows, macOS, andLinux, and is used to troubleshoot DNS-related issues,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

nslookup www.google.com



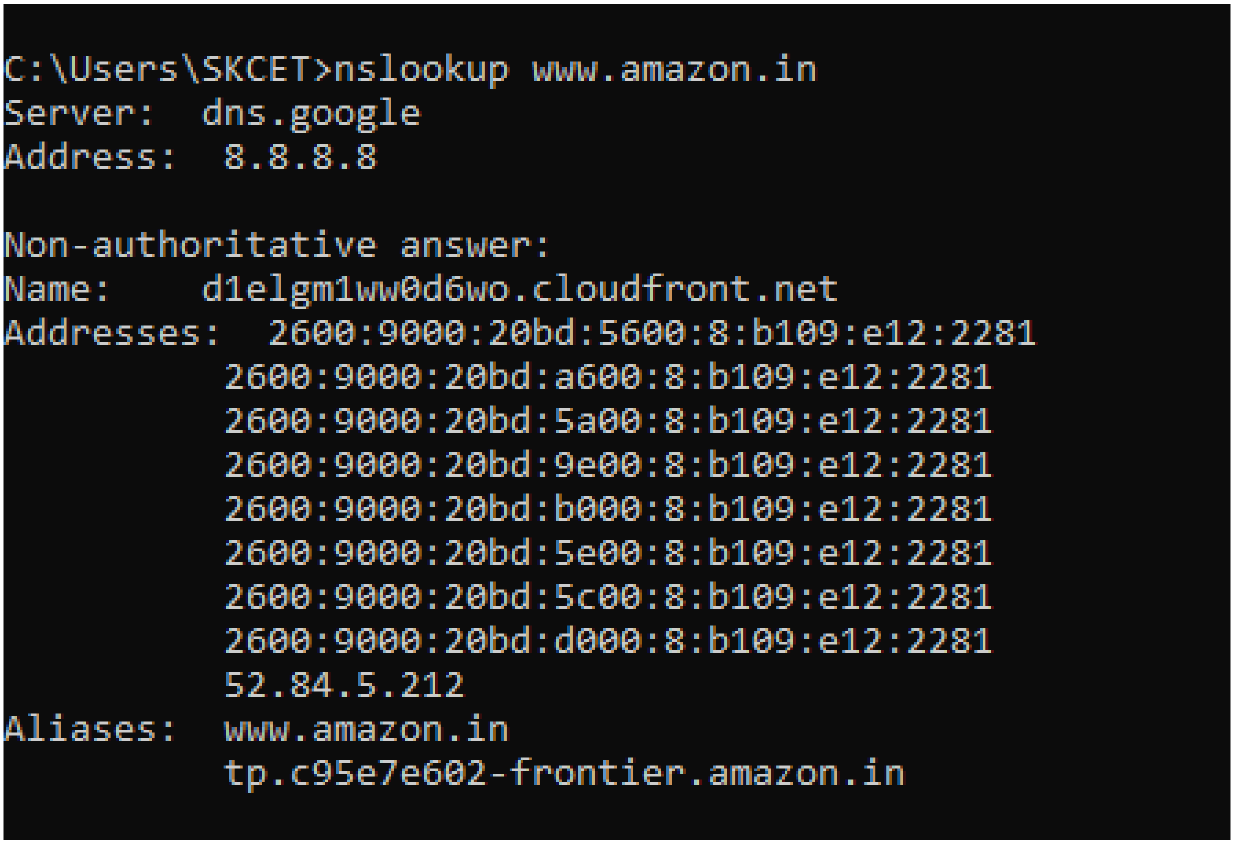
```
C:\Users\SKCET>nslookup www.google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4007:821::2004
          142.250.76.68
```

nslookup www.chess.com



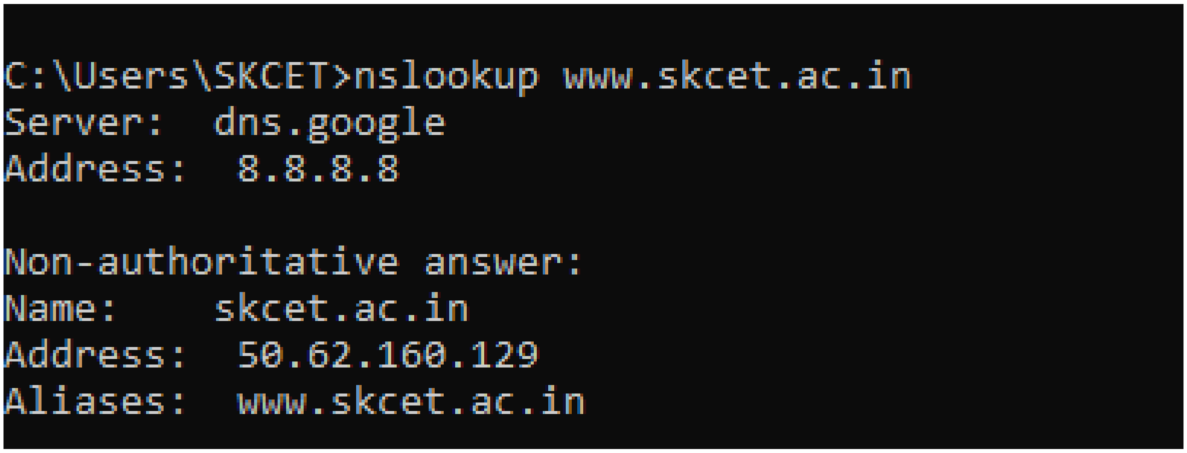
```
C:\Users\SKCET>nslookup www.chess.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    www.chess.com.cdn.cloudflare.net
Addresses:  104.17.81.122
          104.17.80.122
          104.17.79.122
          104.17.78.122
          104.17.77.122
Aliases:  www.chess.com
```

nslookup www.amazon.in



```
C:\Users\SKCET>nslookup www.amazon.in
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    d1elgm1ww0d6wo.cloudfront.net
Addresses:  2600:9000:20bd:5600:8:b109:e12:2281
          2600:9000:20bd:a600:8:b109:e12:2281
          2600:9000:20bd:5a00:8:b109:e12:2281
          2600:9000:20bd:9e00:8:b109:e12:2281
          2600:9000:20bd:b000:8:b109:e12:2281
          2600:9000:20bd:5e00:8:b109:e12:2281
          2600:9000:20bd:5c00:8:b109:e12:2281
          2600:9000:20bd:d000:8:b109:e12:2281
          52.84.5.212
Aliases:  www.amazon.in
          tp.c95e7e602-frontier.amazon.in
```

nslookup www.skcet.ac.in

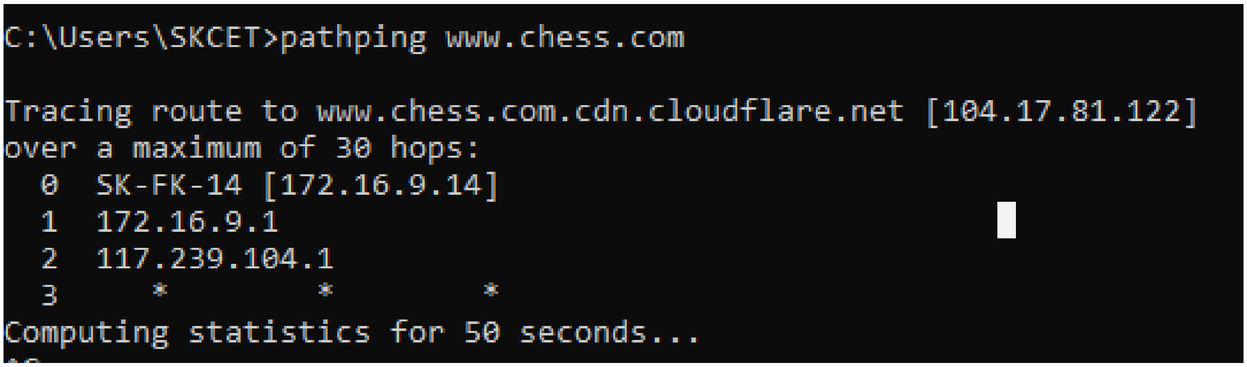


```
C:\Users\SKCET>nslookup www.skcet.ac.in
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    skcet.ac.in
Address:  50.62.160.129
Aliases:  www.skcet.ac.in
```

viii) Command: pathping www.chess.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:

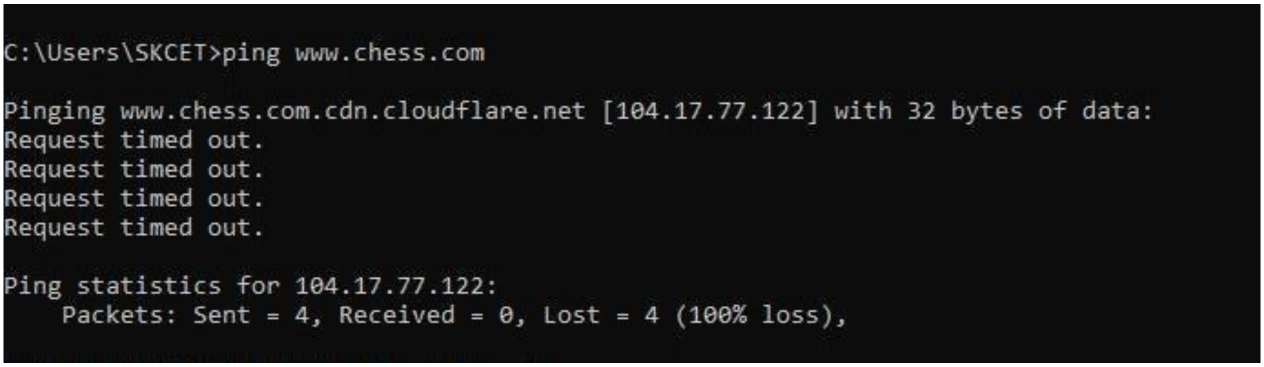


```
C:\Users\SKCET>pathping www.chess.com

Tracing route to www.chess.com.cdn.cloudflare.net [104.17.81.122]
over a maximum of 30 hops:
  0  SK-FK-14 [172.16.9.14]
  1  172.16.9.1
  2  117.239.104.1
  3     *        *        *
Computing statistics for 50 seconds...
```

ix) Command: ping www.chess.com

The ping command is a network utility available on most operating systems, includingWindows, 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 commandsends ICMP (Internet Control Message Protocol) echo requests to the remote host and waitsfor ICMP echo replies. This is a simple and widely used tool for checking network connectivity and latency.

Output:

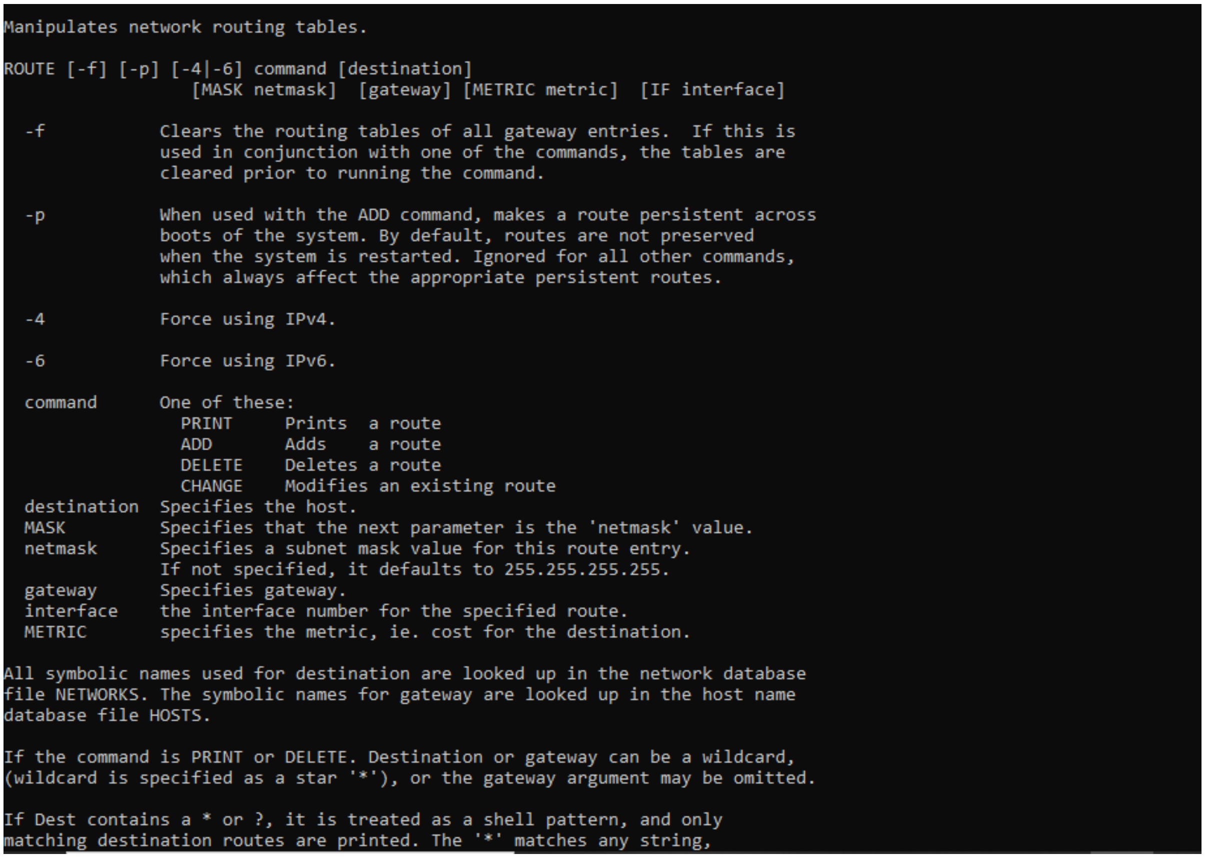


```
C:\Users\SKCET>ping www.chess.com

Pinging www.chess.com.cdn.cloudflare.net [104.17.77.122] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 104.17.77.122:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

x) Command: Route www.chess.com

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:

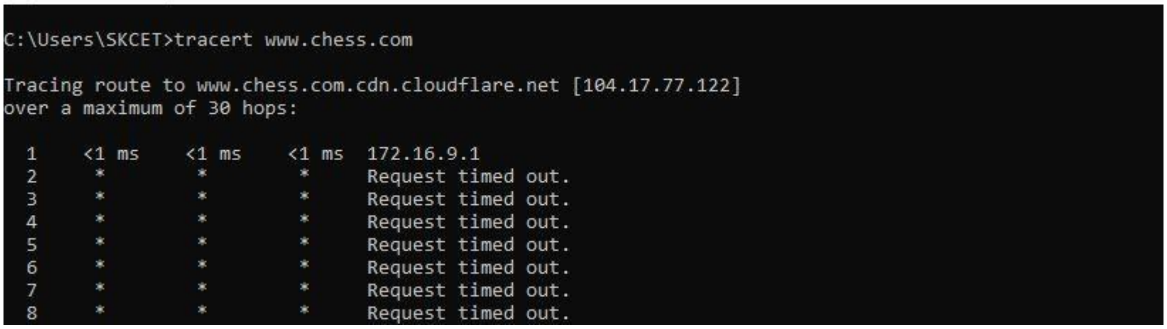


```
Manipulates network routing tables.

ROUTE [-f] [-p] [-4|-6] command [destination]
                  [MASK netmask]  [gateway] [METRIC metric]  [IF interface]

  -f           Clears the routing tables of all gateway entries.  If this is
               used in conjunction with one of the commands, the tables are
               cleared prior to running the command.

  -p           When used with the ADD command, makes a route persistent across
               boots of the system. By default, routes are not preserved
               when the system is restarted. Ignored for all other commands,
               which always affect the appropriate persistent routes.

  -4           Force using IPv4.

  -6           Force using IPv6.

  command      One of these:
                 PRINT     Prints  a route
                 ADD       Adds    a route
                 DELETE    Deletes a route
                 CHANGE    Modifies an existing route
  destination  Specifies the host.
  MASK         Specifies that the next parameter is the 'netmask' value.
  netmask      Specifies a subnet mask value for this route entry.
               If not specified, it defaults to 255.255.255.255.
  gateway      Specifies gateway.
  interface    the interface number for the specified route.
  METRIC       specifies the metric, ie. cost for the destination.

All symbolic names used for destination are looked up in the network database
file NETWORKS. The symbolic names for gateway are looked up in the host name
database file HOSTS.

If the command is PRINT or DELETE. Destination or gateway can be a wildcard,
(wildcard is specified as a star '*'), or the gateway argument may be omitted.

If Dest contains a * or ?, it is treated as a shell pattern, and only
matching destination routes are printed. The '*' matches any string,
```

xi)Command: tracert www.chess.com

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:



```
C:\Users\SKCET>tracert www.chess.com

Tracing route to www.chess.com.cdn.cloudflare.net [104.17.77.122]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  172.16.9.1
  2     *        *        *     Request timed out.
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
  8     *        *        *     Request timed out.
```

RESULT:

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

Experiment No.	: 12)a
Title of Experiment	: Implements of Subnetting using JAVA
Date of Experiment	:

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 helps 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

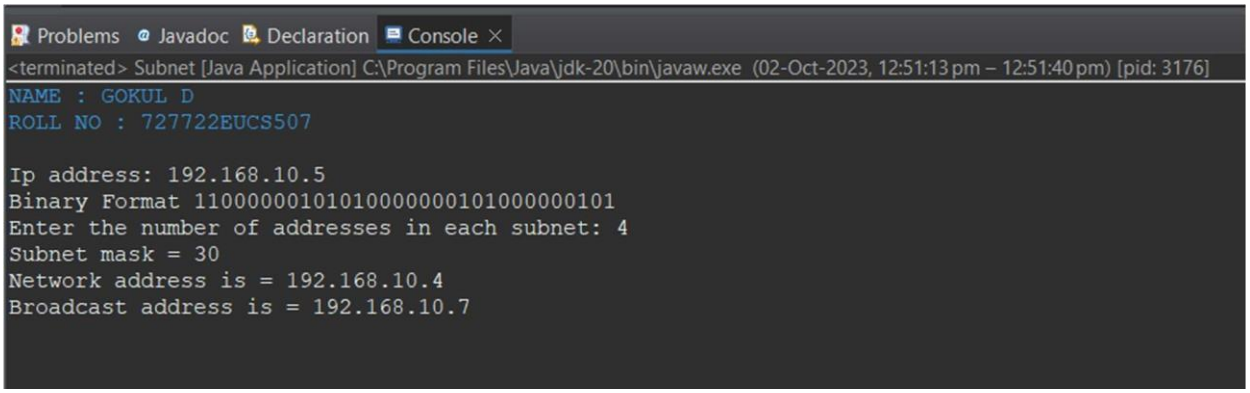
Description on classes, objects, constructors and methods used:

- 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 asanunsigned 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 from0 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(spl
                it_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
        addressint 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 0fbip[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 1lbip[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 :



```
<terminated> Subnet [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe  (02-Oct-2023, 12:51:13 pm – 12:51:40 pm) [pid: 3176]
NAME : GOKUL D
ROLL NO : 727722EUCS507

Ip address: 192.168.10.5
Binary Format 11000000101010000000101000000101
Enter the number of addresses in each subnet: 4
Subnet mask = 30
Network address is = 192.168.10.4
Broadcast address is = 192.168.10.7
```

RESULT:

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

Experiment No.	: 12)b
Title of Experiment	: **APPLICATIONS USING TCP SOCKETS –CONCURRENT SERVER**
Date of Experiment	:

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 threadsdirectly. 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

resourcesServer:

1. Server is listening on port 5057
2. Run infinite loop to get client requests

3. Create socket object to receive incoming client requests
4. Obtain input and out streams
5. Create a new thread object
6. Invoke the start() method
7. Create a ClientHandler class
8. Ask user what he wants
9. Receive answer from the client
10. Create Date Object
11. Write on output stream based on answer from the client
12. 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 thespecified port number on the named host.

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

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

* writeUTF(String str)- Writes a string to the underlying output stream using modifiedUTF-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 andaccepts 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.readUT
                F());String tosend =
                scn.nextLine();
                dos.writeUTF(tosend);
                // If client sends exit,close this connection
                // and then break from the while
                loopif (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();
        }
    }
}
```

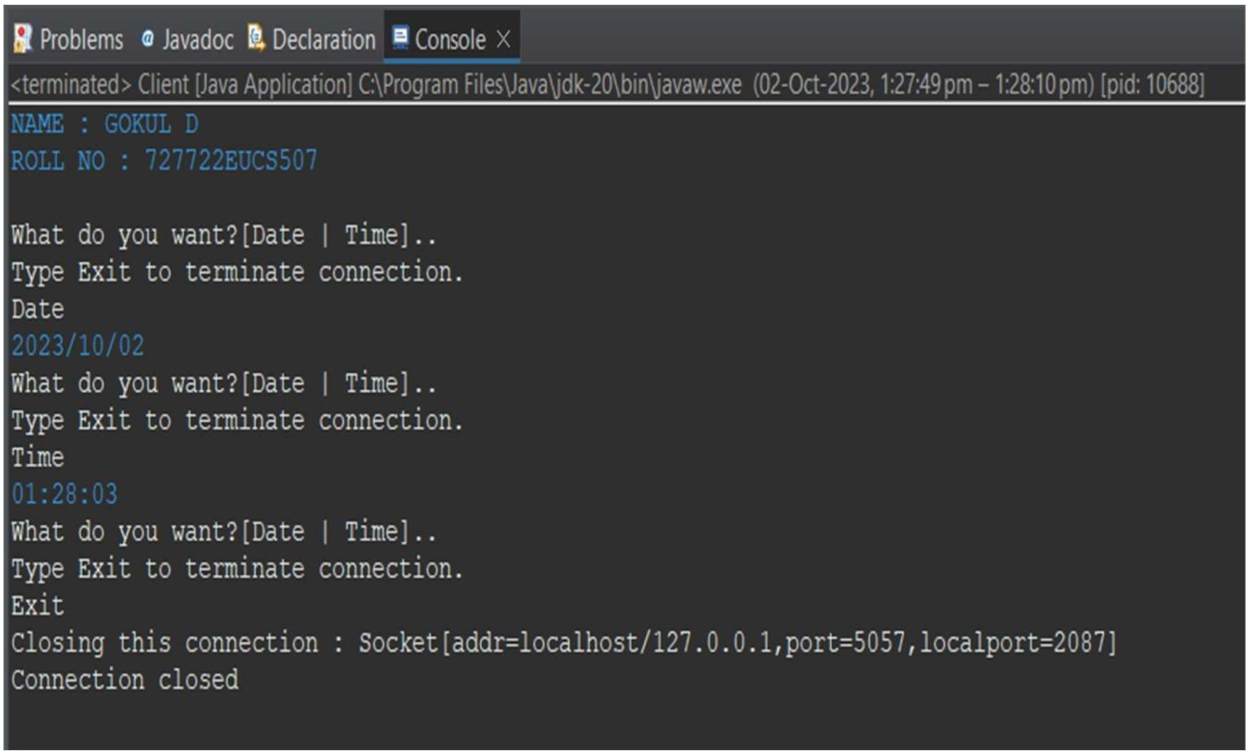
Server.java

```
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();

                switch (received) {
                case "Date":
                    toreturn = cyanColorCode + fordate.format(date) +

resetColorCo
de;                 dos.writeUTF(toreturn);
                    break;
                case "Time":
                    toreturn = cyanColorCode + fortime.format(date) +

                    dos.writeUTF(toreturn);
resetColorCo        break;
de;             default:
                    dos.writeUTF("Invalid input");
                    break;
```

```
                                        }
                                } catch (IOException e)
                                        {
                                        e.printStackTrac
                                        e();



                        }
                        try {
        }

        this.dis.close();

                                this.dos.close();
                        } catch (IOException e) {
                                e.printStackTrace();
                        }
                }
        }
```

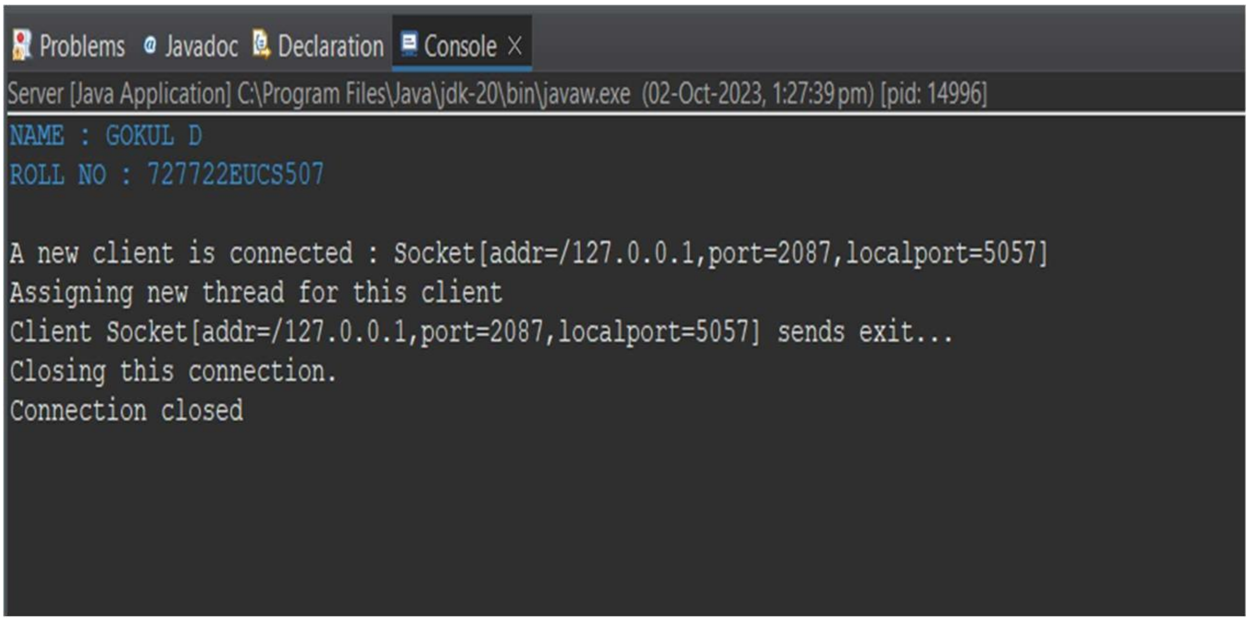
OUTPUT :

Client.java



```
<terminated> Client [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe (02-Oct-2023, 1:27:49 pm – 1:28:10 pm) [pid: 10688]
NAME : GOKUL D
ROLL NO : 727722EUCS507

What do you want?[Date | Time]..
Type Exit to terminate connection.
Date
2023/10/02
What do you want?[Date | Time]..
Type Exit to terminate connection.
Time
01:28:03
What do you want?[Date | Time]..
Type Exit to terminate connection.
Exit
Closing this connection : Socket[addr=localhost/127.0.0.1,port=5057,localport=2087]
Connection closed
```

Server.java



```
Server [Java Application] C:\Program Files\Java\jdk-20\bin\javaw.exe  (02-Oct-2023, 1:27:39 pm) [pid: 14996]
NAME : GOKUL D
ROLL NO : 727722EUCS507

A new client is connected : Socket[addr=/127.0.0.1,port=2087,localport=5057]
Assigning new thread for this client
Client Socket[addr=/127.0.0.1,port=2087,localport=5057] sends exit...
Closing this connection.
Connection closed
```

RESULT:

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