BRAINWARE UNIVERSITY
School of Engineering
Department of Computer Science & Engineering—Cyber Security & Data Science
398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

Department of Computer Science & Engineering—Cyber Security & Data Science

Laboratory Manual

Bachelor of Technology in Computer Science & Engineering—Data Science
PCC-CSD 592, Computer Networks Lab

Compiled by
Mr. NILAV DARSAN MUKHOPADHYAY
Assistant Professor
Dept. of CSE- CS & Ds
Brainware University

BRAINWARE UNIVERSITY
School of Engineering
Department of Computer Science & Engineering—Cyber Security & Data Science
398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

Content

Contents

Date	Compiled by	Description
January 2025	Mr. Nilav Darsan Mukhopadhyay	Updated with 15 points for each example
January 2023	Dept. of CSE	Initial release

Table: Revision History

Experiment No. 1: Network Devices

1. Aim / Purpose of the Experiment

Case study on Network Devices: Repeater, hub, switch, router, gateway, Commands for communication: ping, tracert, nslookup, ARP, Learn about RJ45co nnector, different cable and crimping too

2. Learning Outcomes

Upon completion, students will be able to:

- Identify and differentiate between various types of network devices.
- Explain the functionality and use-cases of each device.
- Analyze network topologies and determine the placement of devices.
- Configure basic settings of routers and switches (if simulated or real setup available).
- Evaluate how network devices impact performance and security.

3. Prerequisites

- Basic knowledge of computer networks and OSI/TCP-IP model
- Familiarity with IP addressing and subnetting
- Hands-on experience with any network simulator

4. Materials / Equipment / Apparatus / Software Required

- Network Simulator: Cisco Packet Tracer / GNS3 / Wireshark (for traffic analysis)
- Internet access for research
- Optional: Real networking hardware (router, switch, access point)

5. Introduction and Theory

Network devices are essential components that facilitate communication between computers and other network-enabled devices. These include:

- **Hub:** A basic device that broadcasts data to all connected devices; operates at Layer 1.
- **Switch:** An intelligent device that sends data to the intended recipient using MAC addresses; operates at Layer 2.
- **Router:** Connects different networks and routes data using IP addresses; operates at Layer 3.
- **Access Point (AP):** Allows wireless devices to connect to a wired network.
- **Firewall:** A security device or software that monitors and controls incoming and outgoing network traffic based on security rules.

6. Operating Procedure

1. **Study Phase:**
    - Use theoretical references to understand the features and OSI layer functionality of each device.
    - Identify the input/output ports and LEDs on physical/simulated devices.
2. **Simulation Phase:**

- Open Cisco Packet Tracer or other tool.
- Create a simple network topology including PCs, switches, routers, and APs.
- Configure IP addresses on PCs and routers.
- Test connectivity using ping

3. **Observation Phase:**
   - Observe how data is routed through each device.
   - Use simulation tools to capture packet flow and device behavior.
   - Optionally, use Wireshark to analyze packets if a real network is used.

7. Precautions and/or Troubleshooting

- Ensure correct IP addressing and avoid IP conflicts.
- Do not overload the simulator with too many devices.
- Use secure configurations (e.g., disable unused ports).
- Save configurations frequently.
- Avoid connecting wrong cable types between devices.

8. Observations

Device	Layer	Function	Configuration Done
Hub	Layer 1	Broadcasts to all	N/A
Switch	Layer 2	Directs to MAC	MAC Table Built
Router	Layer 3	Routes by IP	IP Configured
Access Point	Layer 2	Wireless bridge	SSID set
Firewall	Layer 3/4	Blocks/Allows traffic	Rules Set

10. Result & Interpretation

Students successfully observed and understood the roles and behavior of different network devices through simulation and theoretical analysis.

11. Follow-up Questions

1. What is the main difference between a hub and a switch?
2. How does a router decide where to forward a packet?
3. Why is a firewall important in a network?
4. What happens if two routers are connected incorrectly?

5. Can an access point also act as a router? Explain with an example.

13. Assessments

- Quiz on Network devices
- Viva-voce questions onOSI Layer

14. Suggested Readings

- **"Computer Networking: A Top-Down Approach"** by Kurose and Ross – Pearson
- **"Data Communications and Networking"** by Behrouz A. Forouzan – McGraw-Hill
- **Cisco Networking Academy Packet Tracer Labs and Tutorials**

Experiment No 2: Implementing Different Network Topologies

1. Aim / Purpose of the Experiment

To study, design, and implement various network topologies (Bus, Star, Ring, Mesh, and Hybrid) and analyze their behavior using simulation software or hardware setup.

2. Learning Outcomes

After completing this activity, students will be able to:

- Define and distinguish between different network topologies.
- Design logical network topologies using simulation tools.
- Evaluate the strengths and weaknesses of each topology.
- Implement and test network connectivity for different layouts.
- Analyze data transmission and calculate performance metrics such as transmission delay and number of links.

3. Prerequisites

- Basic understanding of networking concepts (nodes, links, IP addresses)
- Familiarity with the OSI and TCP/IP models
- Knowledge of networking commands such as ping, ipconfig, etc.
- Experience with network simulation tools like Cisco Packet Tracer or GNS3

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer / GNS3 / NS2 / NS3 / Wireshark
- Computer with minimum 4 GB RAM

5. Introduction and Theory

A **network topology** is the physical or logical arrangement of nodes and connections in a network. Different topologies influence cost, performance, and fault tolerance.

- **Bus Topology**: All devices connected to a single backbone. Simple, but collision-prone.
- **Star Topology**: All nodes connected to a central switch or hub. Easy to manage.
- **Ring Topology**: Nodes connected in a circular fashion. Each node has exactly two neighbors.
- **Mesh Topology**: Every node connected to every other. High reliability.
- **Hybrid Topology**: A combination of two or more topologies, e.g., star-ring.

Each topology has trade-offs in terms of cabling cost, reliability, and performance.

6. Operating Procedure

- **Launch the simulation tool (e.g., Cisco Packet Tracer).**
- **Design each topology:**
- Add appropriate number of PCs and interconnecting devices (switches, hubs).
- Use correct cable types for connections.
- **Assign IP addresses** to all nodes based on a chosen IP scheme.
- **Test the connectivity** using `ping` from one node to another.
- **Observe packet flow** using simulation mode (if supported).
- **Repeat for each topology**: bus, star, ring, mesh, hybrid.
- **Document** all observations including success/failure of communication.

7. Precautions and/or Troubleshooting

- Avoid IP address conflicts.
- Connect devices using appropriate cables (copper straight-through or crossover).
- Turn on simulation mode after completing topology setup.
- Save your work frequently

8. Observations

Topology	Devices Used	Total Links	Ping Success
Bus	4 PCs + 1 hub	4	Yes
Star	4 PCs + 1 switch	4	Yes
Ring	4 PCs	4	Yes
Mesh	4 PCs	6	Yes

9. Calculations & Analysis

Number of links required:

- Bus: n (where n = number of devices)

- Star: n (each device connects to central switch)
- Ring: n (each connects to 2 neighbors)
- Mesh: n(n-1)/2
  For 4 devices: 4(4-1)/2 = 6 links
- Hybrid: Depends on components used

2. Transmission Delay:

Formula:
Transmission Delay = Data Size / Bandwidth

If data size = 1000 bits and bandwidth = 1000 bits/sec:
Delay = 1000 / 1000 = 1 sec

10. Result & Interpretation

Different network topologies were implemented and tested successfully. Their performance and design trade-offs were analyzed. Mesh showed the highest reliability, while star offered a good balance of performance and simplicity.

11. Follow-up Questions

1. Why is mesh topology rarely used in small networks?
2. In which topology does a single point of failure affect the whole network?
3. How does hybrid topology overcome limitations of basic topologies?
4. What happens when a link breaks in a ring topology?
5. Which topology is most scalable and why?

13. Assessments

- **Quiz/Test:** Multiple choice and short-answer questions on topology characteristics.
- **Lab Task:** Design and simulate a hybrid topology combining ring and star.
- **Viva:** Questions on practical observations and configuration.

14. Suggested Readings

- **"Data and Computer Communications"** by William Stallings – Pearson
- **"Computer Networking: Principles, Protocols and Practice"** by Olivier Bonaventure – Open Source
- **"Networking All-in-One For Dummies"** by Doug Lowe – Wiley

Experiment No 3: Implementation of LAN (Local Area Network) and MAN (Metropolitan Area Network)

1. Aim / Purpose of the Experiment

To understand and implement the basic structure of LAN and simulate a MAN environment using network simulation tools, analyzing their characteristics, performance, and applications

2. Learning Outcomes

After completing this experiment/case study, students will be able to:

1. Define and distinguish between LAN and MAN.
2. Design and implement a LAN using appropriate networking devices.
3. Simulate a basic MAN setup interconnecting multiple LANs.
4. Analyze topology, transmission, and data flow in both network types.
5. Understand practical use cases and performance implications of LANs and MANs.

3. Prerequisites

- Basic understanding of networking concepts (nodes, switches, routers, IP addressing)
- Knowledge of network models (OSI, TCP/IP)
- Familiarity with simulation tools like Cisco Packet Tracer or GNS3
- Understanding of subnetting and routing

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer / GNS3 / NS2 or NS3
- Computer system with required specifications
- Basic networking hardware

5. Introduction and Theory

Local Area Network (LAN):

A LAN is a network limited to a small geographic area such as a room, building, or campus. It enables high-speed communication and resource sharing among computers.

- Devices: PCs, Switches, Routers, Access Points
- Technologies: Ethernet, Wi-Fi
- Common Protocols: TCP/IP, DHCP, DNS

Metropolitan Area Network (MAN):

A MAN spans a city or a large campus, connecting multiple LANs. It is larger than a LAN but smaller than a Wide Area Network (WAN).

- Devices: Routers, Modems, Switches, Fiber links
- Technologies: Fiber optics, Metro Ethernet, DSL
- Use cases: Inter-campus universities, bank branches in a city, city-wide ISPs

6. Operating Procedure

LAN Implementation:

- Open the network simulation tool (e.g., Cisco Packet Tracer).
- Place 4–6 PCs, 1 switch, and a router.
- Connect PCs to the switch using straight-through cables.

- Assign IP addresses to each PC and configure the router.
- Test connectivity using the ping command.
- Observe packet flow and connectivity.

MAN Implementation (Simulation of multiple LANs):

- Create 2 or 3 LANs similar to the above steps.
- Interconnect them using routers or fiber links (simulate using serial connections).
- Assign IP addresses using subnetting.
- Configure static or dynamic routing between routers.
- Test communication between LANs (e.g., PC1 in LAN1 to PC6 in LAN3).
- Observe routing and latency.

7. Precautions and/or Troubleshooting

- Avoid IP address duplication.
- Ensure correct cable types (crossover for router-router, straight-through for PC-switch).
- Turn on all devices and interfaces.
- Save simulation periodically.
- Ensure routing tables are properly configured.

8. Observations

Network Type	Devices	No. of Hosts	Topology Used	Ping Status	Packet Delay	Comments
LAN 1	1 switch, 4 PCs, 1 router	4	Star	Success	Low	Fast local communication
LAN 2	1 switch, 4 PCs, 1 router	4	Star	Success	Low	Independent subnet
MAN (LAN1 + LAN2)	2 routers, serial link	8	Hybrid	Success	Medium	Routing between LANs successful

9. Calculations & Analysis

Subnet Calculation (Example):

- Network: 192.168.10.0/24
- Subnet 1 (LAN1): 192.168.10.0/28 (14 hosts)
- Subnet 2 (LAN2): 192.168.10.16/28 (14 hosts)

Number of Links Required:

- LAN: For n devices, n links to the switch.
- MAN: One serial/fiber link between routers.

Transmission Delay:

- Delay = Data Size / Bandwidth
- For LAN (100 Mbps): 1000 bits / 100×10^6 = 0.00001 sec
- For MAN (10 Mbps): 1000 bits / 10×10^6 = 0.0001 sec

10. Result & Interpretation

- Successfully implemented LAN and simulated MAN connections using multiple LANs.
- Verified interconnectivity using ping and observed latency differences.
- Understood configuration of routing and device interconnections.

11. Follow-up Questions

1. What is the primary difference between LAN and MAN in terms of coverage and performance?
2. Which routing protocols can be used for MAN implementation?
3. Why is it essential to subnet a MAN?
4. How can you enhance the security in a MAN setup?
5. Can MAN be wireless? Give examples

13. Assessments

- **Written Test/Quiz:** Definitions, comparison, routing concepts
- **Practical Task:** Implement 2 LANs and connect them using routers
- **Viva Voce:** Questions on routing, subnetting, latency analysis

14. Suggested Readings

- **Data Communications and Networking"** by Behrouz A. Forouzan – McGraw Hill
- **"Computer Networks"** by Andrew S. Tanenbaum – Pearson
- **Cisco Networking Academy – CCNA Course Materials**

Experiment No 4: Implementation of Wide Area Network (WAN)

1. Aim / Purpose of the Experiment

To design and simulate the implementation of a Wide Area Network (WAN) that interconnects multiple geographically distributed LANs using routers and WAN protocols, and to analyze connectivity and routing behavior

2. Learning Outcomes

By the end of this experiment, students will be able to:

- Define WAN and differentiate it from LAN and MAN.
- Design a WAN using appropriate devices and communication links.

- Simulate WAN using tools like Cisco Packet Tracer/GNS3.
- Configure routing protocols for WAN communication.
- Analyze latency, packet routing, and network performance across long distances.

3. Prerequisites

- Basic knowledge of LAN and MAN topologies
- Understanding of IP addressing, subnetting, and routing
- Familiarity with network simulation tools (e.g., Packet Tracer, GNS3)
- Concepts of static/dynamic routing (e.g., RIP, OSPF)

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer or GNS3
- Computer with minimum 4 GB RAM
- Networking cables

5. Introduction and Theory

A **Wide Area Network (WAN)** spans large geographic areas such as cities, states, or countries. It is used to connect multiple LANs or MANs using long-distance communication technologies like leased lines, satellite links, MPLS, and VPN.Key WAN devices and technologies:

- **Routers**: Forward packets between LANs
- **Modems**: Convert digital signals for analog transmission
- **Serial Links**: Used to simulate leased lines between routers
- **Protocols**: PPP, HDLC, Frame Relay, MPLS, VPN

WANs are essential for organizations with multiple branches, cloud connectivity, and internet backbone infrastructure.

6. Operating Procedure

Designing the WAN:

- Open Packet Tracer.
- Create three separate LANs (e.g., for different cities).
- Add routers to each LAN and connect them using serial connections to simulate WAN links.

IP Address Assignment:

- Assign unique IP subnets to each LAN and WAN link.
- Configure IP addresses on PCs, routers' Ethernet and Serial interfaces.

Configure Routing:

- Use **RIP v2** or **OSPF** routing protocol.
- Ensure routers exchange routing information across the WAN.

Test Connectivity:

- Use ping to test connectivity between PCs in different LANs.
- Use tracert (or simulation mode) to observe the path taken by packets.

Analyze Results:

- Use simulation tools to examine packet delay, route selection, and router behavior

7. Precautions and/or Troubleshooting

- Ensure proper cable connections (serial for WAN, straight-through for LAN).
- Avoid IP conflicts by using proper subnetting.
- Enable and configure routing protocols correctly.
- Activate all router interfaces using no shutdown command.
- Save configurations frequently

8. Observations

Location	LAN Subnet	Router Name	WAN Interface IP	Ping to Other LANs	Routing Table Updated
Delhi	192.168.10.0/24	R1	10.0.0.1	Yes	Yes
Mumbai	192.168.20.0/24	R2	10.0.0.2	Yes	Yes
Kolkata	192.168.30.0/24	R3	10.0.0.3	Yes	Yes

9. Calculations & Analysis

1. **Subnetting Example:**
    - LAN Subnets:
      Delhi: 192.168.10.0/24
      Mumbai: 192.168.20.0/24
      Kolkata: 192.168.30.0/24
    - WAN Subnet (point-to-point links):
      Between R1 and R2: 10.0.0.0/30
      Between R2 and R3: 10.0.0.4/30

2. **Routing Path Trace:**
   - Example: PC in Delhi (192.168.10.2) to PC in Kolkata (192.168.30.2)
   - Route: R1 → R2 → R3
3. **Latency Estimation (Simulated):**

   - For WAN: Delay = Propagation Delay + Transmission Delay
   - Transmission Delay = Data Size / Bandwidth
   - For a 1000-bit packet over 1 Mbps:
     1000 / 1,000,000 = 0.001 seconds

10. Result & Interpretation

The WAN simulation was successfully implemented. Multiple LANs in different locations were connected using routers and serial links. Routing protocols enabled successful communication across the WAN. Latency was observed to be higher compared to LAN, indicating real-world behavior.

11. Follow-up Questions

1. What is the main difference between LAN, MAN, and WAN?
2. Why is routing necessary in WAN implementation?
3. What are the advantages of using dynamic routing protocols over static routing?
4. What challenges might you face in real WAN setups (e.g., security, bandwidth)?
5. How can VPNs be integrated into WAN designs?

13. Assessments

- **Quiz/Test:** Questions on WAN characteristics, routing, and protocols
- **Lab Task:** Simulate a WAN interconnecting 3 LANs using RIP or OSPF
- **Viva:** Configuration steps, routing table analysis

14. Suggested Readings

- **"Data Communications and Networking"** by Behrouz A. Forouzan – McGraw Hill
- **"Computer Networks"** by Andrew S. Tanenbaum – Pearson
- **"CCNA Routing and Switching"** Study Guide by Todd Lammle – Wiley

Experiment No 5: Implementation of Distance and link state Routing-RIP

1. Aim / Purpose of the Experiment

To implement and configure the Routing Information Protocol (RIP) on routers in a simulated or physical network environment, and to analyze dynamic routing behavior and path updates

2. Learning Outcomes

By the end of this lab, students will be able to:

- Understand the purpose and working of RIP in a network
- Configure RIP on routers for automatic route discovery.
- Observe routing table updates and convergence behavior.
- Compare RIP with other routing techniques (e.g., static routing).
- Analyze routing loops and convergence time in RIP.

3. Prerequisites

- Basic knowledge of IP addressing and subnetting
- Understanding of routing concepts and types (static vs. dynamic)
- Familiarity with Cisco Packet Tracer or other network simulators
- Knowledge of router interface configuration and basic commands

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer / GNS3 / Physical routers (optional)
- Computer with required specifications

5. Introduction and Theory

Routing Information Protocol (RIP) is a distance-vector dynamic routing protocol used to determine the best path for data packets in a network. RIP uses **hop count** as the metric and allows a maximum of **15 hops**, making it suitable for small to medium-sized networks.

Key features of RIP:

- Works at Layer 3 (Network layer)
- Uses UDP port 520
- Updates routing tables every 30 seconds
- RIP v1: Classful, no subnet mask info
- RIP v2: Classless, supports VLSM and subnetting
- RIP helps routers share information about networks they can reach, without needing manual route configuration.

6. Operating Procedure

Network Design:

- Open Cisco Packet Tracer.
- Create a network with 2–3 routers and at least one PC in each router's LAN.
- Connect routers using serial links and PCs using Ethernet connections to switches.

IP Configuration:

- Assign IP addresses to PCs and interfaces on routers.
- Ensure subnetting is done properly to avoid overlap.

Basic Router Configuration:

Use CLI to enter configuration mode on each router:

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
```

RIP Configuration:

Enable RIP on each router and define the networks to be advertised:

```
Router(config)# router rip
Router(config-router)# version 2
Router(config-router)# network 192.168.1.0
Router(config-router)# network 10.0.0.0
```

Repeat for all routers with their respective networks.

Verify Configuration:

Use the following commands to check routing tables:

```
Router# show ip route

Router# show ip protocols
Router# show running-config
```

Use ping to test connectivity across the network.

Observe RIP Updates:

1. Switch to simulation mode to view RIP packets.
2. Observe routing table changes after 30 seconds or after a link failure/recovery.
3. Traverse until the starting node is reached again.

7. Precautions and/or Troubleshooting

- Ensure that IP subnets do not overlap.
- Always use the same RIP version on all routers (preferably version 2).
- Use no auto-summary in RIP v2 if using discontiguous networks.
- Avoid unnecessary routing loops; verify correct network advertisement.
- Save configuration using write memory or copy run start.
- Save configuration using write memory or copy run start.

8. Observations

Router	Directly Connected Networks	RIP Networks Advertised	Received Routes	Ping Success
R1	192.168.1.0, 10.0.0.0	Yes	R2, R3 routes	Yes
R2	192.168.2.0, 10.0.0.0	Yes	R1, R3 routes	Yes
R3	192.168.3.0, 10.0.1.0	Yes	R1, R2 routes	Yes

9. Calculations & Analysis

Hop Count:

If PC1 (R1's LAN) sends data to PC3 (R3's LAN) via R2:

- Path: R1 → R2 → R3 → PC3
- Hop count = 3

Subnetting (Example):

- 192.168.1.0/24 for LAN1
- 192.168.2.0/24 for LAN2
- 192.168.3.0/24 for LAN3
- 10.0.0.0/30 and 10.0.0.4/30 for serial links

Routing Table Example:

- R1:

```
C 192.168.1.0/24 is directly connected
```

- `R 192.168.3.0/24 [120/2] via 10.0.0.2`

10. Result & Interpretation

The RIP protocol was successfully configured and implemented across a multi-router network. Routing tables were updated dynamically, and connectivity between PCs in different networks was established through RIP-learned routes.

11. Follow-up Questions

1. What metric does RIP use to determine the best route?
2. What is the maximum number of hops allowed in RIP?
3. What is the difference between RIP version 1 and version 2?
4. How often does RIP update its routing table?
5. What happens when a router or link goes down in a RIP network

13. Assessments

- **Quiz/Test:** MCQs and short answers on RIP characteristics and configuration
- **Practical Task:** Configure RIP for a three-router network and demonstrate routing table updates
- **Viva Voce:** Questions on RIP commands, metrics, and behavior

14. Suggested Readings

- **"Routing Protocols and Concepts – CCNA Exploration"** by Cisco Networking Academy
- **"Computer Networks"** by Andrew S. Tanenbaum – Pearson
- **"Data Communications and Networking"** by Behrouz A. Forouzan – McGraw Hill

Experiment No 6: Implementation of Distance and link State Routing- EIGRP

1. Aim / Purpose of the Experiment

To implement and configure EIGRP on a multi-router network using simulation tools, and to analyze its routing behavior, convergence, and efficiency

2. Learning Outcomes

Upon successful completion, students will be able to:

- Understand the operation and features of EIGRP.
- Configure EIGRP in a network using Cisco IOS commands.
- Analyze EIGRP metrics, convergence, and routing tables.
- Differentiate EIGRP from RIP and OSPF.
- Observe and verify EIGRP neighbor relationships and updates.

3. Prerequisites

- Understanding of IP addressing, subnetting, and CIDR
- Familiarity with router configuration commands
- Basic knowledge of dynamic routing protocols
- Experience with simulation tools like Cisco Packet Tracer or GNS3

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer / GNS3 / Physical routers (optional)
- Computer with minimum 4 GB RAM

- Router simulation with CLI access

5. Introduction and Theory

EIGRP (Enhanced Interior Gateway Routing Protocol) is an advanced distance-vector routing protocol developed by Cisco. It combines features of both distance-vector and link-state protocols, making it more efficient than RIP and easier to configure than OSPF.

Key Features of EIGRP:

- Cisco proprietary (used in Cisco devices)
- Uses **DUAL (Diffusing Update Algorithm)** for loop-free and fast convergence
- Supports **VLSM**, **CIDR**, and **unequal cost load balancing**
- Uses bandwidth, delay, reliability, and load as metrics (default: bandwidth and delay)
- Supports manual summarization

Terminology:

- **Successor**: Primary route to a destination
- **Feasible Successor**: Backup route, already in routing table

6. **Operating Procedure**

1. **Network Design:**
    - Create a topology with 3 routers (R1, R2, R3), each connected to a different LAN.
    - Interconnect routers using serial links.
    - Connect end devices (PCs) to each LAN.
2. **Assign IP Addresses:**
    - Configure IPs for each PC, router interface (Ethernet and Serial).
    - Ensure unique subnets for each LAN and link.
3. **Basic Router Configuration:**

Access each router's CLI and configure hostname, interfaces, and IP addresses:

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
Router(config)# interface FastEthernet0/0
Router(config-if)# ip address 192.168.10.1 255.255.255.0
Router(config-if)# no shutdown
```

4. **Configure EIGRP:**

On each router, enable EIGRP and define the autonomous system number (same across all routers):

```
Router(config)# router eigrp 100
Router(config-router)# network 192.168.10.0
Router(config-router)# network 10.0.0.0
```

5. **Verification:**

Use the following commands:

```
show ip route
show ip protocols
show ip eigrp neighbors
show running-config.
```

7. **Precautions and/or Troubleshooting**

- Use the same autonomous system number on all routers in the same EIGRP domain.
- Avoid mismatched subnet masks or overlapping subnets.
- Ensure router interfaces are not shut down.
- Configure the correct `network` statements.
- Save the configuration using `write memory` or `copy run start`.

8. **Observations**

Router	Interfaces Configured	EIGRP Neighbors	Routes Learned	Ping Success
R1	Fa0/0, S0/0/0	R2	R3's LAN	Yes
R2	Fa0/0, S0/0/0, S0/0/1	R1, R3	All networks	Yes
R3	Fa0/0, S0/0/1	R2	R1's LAN	Yes

9. **Calculations & Analysis**

- **EIGRP Metric Formula (simplified):**
  EIGRP metric = [Bandwidth + Delay] × 256
  - Bandwidth: minimum bandwidth along path (kbps)
  - Delay: sum of delays on all outgoing interfaces (in tens of microseconds)

- Example:
  - Bandwidth = 1000 kbps, Delay = 20000 µs
  - Metric = $(10^7 / 1000 + 20000) \times 256 = (10000 + 20000) \times 256 = 7680000$

10. **Result & Interpretation**

EIGRP was successfully implemented in a multi-router network. All routers learned about other network routes dynamically and maintained neighbor relationships. Packet delivery across different LANs was successful.

11**. Follow-up Questions**

1. What are the advantages of EIGRP over RIP?
2. What is the purpose of the DUAL algorithm in EIGRP?
3. How does EIGRP support unequal-cost load balancing?
4. What is the role of the autonomous system number in EIGRP?
5. What happens if a router interface in EIGRP goes down?

13. **Assessments**

- **Quiz/Test:** Multiple choice and descriptive questions on EIGRP concepts
- **Lab Activity:** Configure EIGRP across 3 routers with 3 LANs and verify convergence
- **Viva Voce:** Explain routing table entries, EIGRP metric components, and neighbor formation
- **Assignment:** Compare EIGRP with OSPF and RIP on the basis of convergence, metrics, scalability, and complexity

14. **Suggested Readings**

- ""**Cisco CCNA Routing and Switching"** by Wendell Odom – Cisco Press
- **"Routing Protocols and Concepts"** – Cisco Networking Academy
- **"Computer Networking: A Top-Down Approach"** by Kurose and Ross – Pearson

Experiment No 7: Implementation of OSPF Protocol

1. Aim / Purpose of the Experiment

To implement and configure the OSPF routing protocol on a multi-router network using simulation tools, and to verify dynamic routing, link-state behavior, and network convergence.

2. Learning Outcomes

Upon successful completion of this lab, students will be able to:

- Understand the OSPF protocol and its features.
- Configure OSPF in single-area networks.
- Verify OSPF neighbor relationships and routing table entries.
- Analyze OSPF metric (cost) and path selection.
- Compare OSPF with RIP and EIGRP in terms of convergence and scalability

3. Prerequisites

- Knowledge of IP addressing, subnetting, and CIDR
- Basic routing concepts (static and dynamic)
- Familiarity with network simulation tools (Packet Tracer, GNS3)
- Understanding of router CLI commands

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer / GNS3 / Physical routers (optional)
- PCs and switches for LAN connectivity
- Router simulation environment (CLI access)

5. Introduction and Theory

OSPF (Open Shortest Path First) is a **link-state routing protocol** used within large enterprise networks. Unlike RIP (which uses hop count), OSPF uses **cost** based on link bandwidth to determine the shortest path. It supports **hierarchical routing** using **areas**, most commonly with a backbone area (Area 0).

Key Features of OSPF:

- Open standard (not vendor-specific)
- Uses Dijkstra's algorithm for shortest path calculation
- Divides large networks into areas for scalability
- Faster convergence than RIP
- Supports VLSM and CIDR
- Sends link-state advertisements (LSAs) rather than full routing table

6. Operating Procedure

Design the Topology:

- Create a network with 3 routers connected via serial interfaces.
- Attach at least one LAN (with PCs and a switch) to each router.

Assign IP Addresses:

- Assign unique IP addresses to all interfaces.
- Subnet appropriately for serial links and LANs.

Configure Routers: On each router, enter global configuration mode:

```
1. Router> enable
2. Router# configure terminal
3. Router(config)# hostname R1
```

Configure OSPF:

```
4. Router(config)# router ospf 1
5. Router(config-router)# network 192.168.1.0 0.0.0.255 area 0
6. Router(config-router)# network 10.0.0.0 0.0.0.3 area 0
```

Verify OSPF Operation:

7. Check routing table:
   ```
   show ip route
   ```
8. View OSPF neighbors:
   ```
   show ip ospf neighbor
   ```
9. Inspect OSPF configuration:
   ```
   show ip protocols
   ```

Test Connectivity:

Ping between PCs in different LANs to ensure dynamic routing is working. Switch to simulation mode to observe OSPF hello packets and LSA exchanges.

7. **Precautions and/or Troubleshooting**

- Ensure all routers are in the same area if not using multi-area OSPF.
- Verify correct wildcard masks in network commands.
- Ensure router interfaces are up and not administratively shut down.
- Use unique router IDs if required (OSPF uses highest IP on active interface by default).

8. **Observations**

Router	Interfaces	OSPF Neighbors	OSPF Routes Learned	Ping Successful
R1	**Fa0/0, S0/0/0**	**R2**	**R3's LAN route**	**Yes**
R2	**Fa0/0, S0/0/0, S0/0/1**	**R1, R3**	**All routes**	**Yes**
R3	**Fa0/0, S0/0/1**	**R2**	**R1's LAN route**	**Yes**

9. **Calculations & Analysis**

1. **Wildcard Mask Calculation:**
   - For subnet 192.168.1.0/24, the wildcard mask is:
     255.255.255.0 → 0.0.0.255
     Command:
     ```
     network 192.168.1.0 0.0.0.255 area 0
     ```
2. **OSPF Cost Calculation (default):**
   - Cost = 100,000,000 / bandwidth in bps
   - For FastEthernet (100 Mbps): Cost = 1
   - For Serial (1.544 Mbps): Cost = 64

3. **Convergence Time:**
    - Measured as time taken from link failure to routing table update
    - Faster than RIP due to link-state awareness

10. **Result & Interpretation**

The OSPF protocol was successfully configured on all routers. Routing tables were dynamically updated, neighbor relationships were established, and communication across all networks was verified. OSPF provided fast convergence and efficient routing

11. **Follow-up Questions**

1. How does OSPF differ from RIP and EIGRP in terms of metric and convergence?
2. What algorithm does OSPF use to calculate shortest path?
3. Why are areas used in OSPF? What is the purpose of Area 0?
4. What are LSAs and how do they function in OSPF?
5. How can you control the cost of an OSPF link?

13. **Assessments**

- **Quiz/Test:** Short answer and multiple-choice questions on OSPF features and configuration
- **Lab Task:** Configure OSPF in a three-router topology and verify dynamic routing
- **Viva Voce:** Explain OSPF states (Down, Init, 2-Way, etc.), LSAs, and neighbor formation
- **Assignment:** Compare OSPF with RIP and EIGRP in terms of performance and scalability

14. **Suggested Readings**

- "**Routing Protocols and Concepts – CCNA Exploration"** by Cisco Networking Academy
- **"CCNA Routing and Switching"** by Todd Lammle – Wiley
- **"Computer Networks"** by Andrew S. Tanenbaum – Pearson

Experiment No 8: Implementation of OSPF using 2 Routers, 2 Switches, and 4 PCs

1. Aim / Purpose of the Experiment

To implement and configure the OSPF routing protocol on a small network comprising two routers, two switches, and four PCs, and to verify routing and connectivity across the network.

2. Learning Outcomes

By the end of this experiment, students will be able to:

- Understand the concept and working of the OSPF routing protocol.
- Configure OSPF on a basic network topology with multiple LANs.
- Observe and analyze OSPF neighbor formation and routing table updates.
- Verify packet delivery between PCs in different networks through OSPF.
- Demonstrate proficiency in using simulation tools for routing protocol configuration.

3. Prerequisites

- Understanding of IP addressing and subnetting
- Familiarity with Cisco IOS router CLI commands
- Basic knowledge of dynamic routing and OSPF fundamentals
- Experience with Cisco Packet Tracer or equivalent simulation tools

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer (or GNS3)
- Two routers (R1, R2)
- Two switches (S1, S2)
- Four PCs (PC0–PC3)
- Ethernet and serial cables (virtual or physical

5. Introduction and Theory

Open Shortest Path First (OSPF) is a link-state, interior gateway routing protocol that uses the **Dijkstra Shortest Path First (SPF)** algorithm to calculate the best path. Unlike RIP (which uses hop count), OSPF uses **cost**, which is derived from the bandwidth of the interface. It supports **hierarchical design** with areas, typically starting with **Area 0** (the backbone area).

Key Points:

- OSPF uses **hello packets** to form neighbor relationships.
- **Converges faster** than RIP and supports **CIDR and VLSM**.
- Routers exchange **Link State Advertisements (LSAs)** to build a l**ink-state database.**

6. Operating Procedure

Network Topology Design:

- Connect:
  - PC0 and PC1 to Switch S1 → S1 to Router R1
  - PC2 and PC3 to Switch S2 → S2 to Router R2
  - R1 and R2 via Serial (or crossover) link

Assign IP Addresses:

- Example configuration:
  - PC0: 192.168.1.2/24 → GW: 192.168.1.1
  - PC1: 192.168.1.3/24 → GW: 192.168.1.1
  - PC2: 192.168.2.2/24 → GW: 192.168.2.1
  - PC3: 192.168.2.3/24 → GW: 192.168.2.1
  - R1 Fa0/0: 192.168.1.1/24
  - R2 Fa0/0: 192.168.2.1/24
  - R1 S0/0/0: 10.0.0.1/30
  - R2 S0/0/0: 10.0.0.2/30

Router Configuration:

R1 Configuration:

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
Router(config)# interface fa0/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config)# interface s0/0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.252
Router(config-if)# no shutdown
Router(config)# router ospf 1
Router(config-router)# network 192.168.1.0 0.0.0.255 area 0
Router(config-router)# network 10.0.0.0 0.0.0.3 area 0
```

R2 Configuration:

```
Router> enable
Router# configure terminal
Router(config)# hostname R2
Router(config)# interface fa0/0
Router(config-if)# ip address 192.168.2.1 255.255.255.0
Router(config-if)# no shutdown
Router(config)# interface s0/0/0
Router(config-if)# ip address 10.0.0.2 255.255.255.252
Router(config-if)# no shutdown
Router(config)# router ospf 1
Router(config-router)# network 192.168.2.0 0.0.0.255 area 0
Router(config-router)# network 10.0.0.0 0.0.0.3 area 0
```

Test Connectivity:

- Use ping from PC0 to PC2 and PC3.
- Use simulation mode to observe OSPF hello packets and LSAs.

Check routing table with:

```
show ip route
show ip ospf neighbor
```

7. Precautions and/or Troubleshooting

- Ensure routers are in the **same OSPF area** (area 0).
- Use correct **wildcard masks** (not subnet masks) in OSPF configuration.
- Confirm that all interfaces are **enabled (`no shutdown`)**.
- Avoid IP overlap and check for correct default gateways on PCs.
- Save configurations after changes (`copy run start` or `write memory`).

8. Observations

Device	Interface	IP Address	OSPF Area	Status	Neighbor
R1	Fa0/0, S0/0/0	192.168.1.1, 10.0.0.1	0	Up	R2
R2	Fa0/0, S0/0/0	192.168.2.1, 10.0.0.2	0	Up	R1
PC0	Ethernet0	192.168.1.2	—	Connected	—
PC2	Ethernet0	192.168.2.2	—	Connected	—

9. Calculations & Analysis
Wildcard Masks:

- 255.255.255.0 → 0.0.0.255
- 255.255.255.252 → 0.0.0.3

OSPF Cost (default formula):
Cost = 100,000,000 / Bandwidth in bps

- FastEthernet (100 Mbps): Cost = 1
- Serial (1.544 Mbps): Cost ≈ 64

Route Path:

- PC0 → R1 → R2 → PC2
- Observed in `show ip route` as an OSPF-learned route (indicated by "O")

10. Result & Interpretation
The OSPF protocol was successfully implemented in a 2-router, 2-switch, and 4-PC topology. Routers established neighbor relationships and dynamically learned routes between LANs. End-to-end communication between PCs on different LANs was successfully verified

11. Follow-up Questions

1. What metric does OSPF use to calculate the best path?
2. How does OSPF detect and establish neighbor relationships?
3. What is the role of Area 0 in OSPF?
4. Why is OSPF considered a link-state protocol?
5. How is OSPF better than RIP for larger networks?

13. Assessments

- **Quiz:** Multiple-choice and short-answer questions on OSPF protocol
- **Practical:** Configure OSPF in the specified topology, verify connectivity and routing
- **Viva Voce:** CLI commands, OSPF areas, cost, neighbor relationships.

14. Suggested Readings

- ""**"Routing Protocols and Concepts"** – Cisco Networking Academy
- **"CCNA Routing and Switching"** by Wendell Odom – Cisco Press
- **"Computer Networks"** by Andrew S. Tanenbaum – Pearso

Experiment No 9:Implementation of NAT (Network Address Translation) Protocol

1. Aim / Purpose of the Experiment

To configure and implement Network Address Translation (NAT) on a router to enable devices in a private network to access external networks (e.g., the Internet), and to understand the different types of NAT.

2. Learning Outcomes

After completing this lab, students will be able to:

- Understand the concept and purpose of NAT in networking.
- Identify different types of NAT (Static, Dynamic, PAT).
- Configure NAT on a router using Cisco IOS.
- Verify NAT operation using real-time simulations.
- Enable private IP address access to external/public networks.

3. Prerequisites

- Basic knowledge of IP addressing and subnetting
- Familiarity with router and switch configuration
- Understanding of public vs. private IP addresses (RFC 1918)
- Basic knowledge of access control lists (ACLs) and interface types

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer or GNS3
- 1 Router
- 2 Switches
- 4 PCs (2 for private network, 2 for public/external simulation)

- Internet cloud (in simulation) or simulated external network

5. Introduction and Theory

Network Address Translation (NAT) is a method used by routers to modify IP address information in packet headers while in transit. NAT is widely used to allow devices with **private IP addresses** to access the **public Internet**.

Types of NAT:

1. **Static NAT** – Maps one private IP to one public IP
2. **Dynamic NAT** – Maps a pool of private IPs to a pool of public IPs
3. **PAT (Port Address Translation)** – Also called NAT Overload, maps many private IPs to one public IP using different port numbers

Why NAT is needed:

- To conserve public IPv4 addresses
- To provide a layer of security by hiding internal IPs
- To enable Internet access from private networks

6. Operating Procedure

Scenario:

Two PCs (PC1, PC2) in the private network use NAT configured on a router to access a public server or simulated external PCs.

1. **Topology Setup:**
   - PC1 and PC2 connected to Switch0 → Router (inside interface)
   - Switch1 with PC3 and PC4 simulates public/external network → Router (outside interface)

2. **IP Address Assignment:**
   Router Configuration (PAT Example):

```
enable
configure terminal
interface fa0/0
ip address 192.168.1.1 255.255.255.0
ip nat inside
no shutdown
exit

interface fa0/1
ip address 203.0.113.1 255.255.255.0
ip nat outside
```

```
12. no shutdown
13. exit
14.
15. access-list 1 permit 192.168.1.0 0.0.0.255
16.
17. ip nat inside source list 1 interface fa0/1 overload
18.
19. exit
```

4. **PC Configuration:**
    - Set default gateway of PC1 and PC2 to 192.168.1.1
    - Set default gateway of PC3 and PC4 to 203.0.113.1
5. **Verification:**
    - Use `ping` from PC1 to PC3

On the router, check NAT translation table: `show ip nat translations`

- `show ip nat statistics`

7. Precautions and/or Troubleshooting

- Ensure interfaces are assigned correct inside/outside roles.
- Use the correct subnet and wildcard mask in the access list.
- Overload (PAT) requires a valid interface with a public IP.
- Verify the PCs' gateway settings point to the router's interface.
- Ensure NAT configuration matches routing setup if any static routes are used.

8. Observations

Action	Expected Result	Verified?
PC1 pinging PC3	Successful ping	Yes/No
NAT table shows translations	Yes (private to public)	Yes/No
Outside interface reachable	Yes	Yes/No

9. Calculations & Analysis

1. **IP Addressing Check:**
   - Inside subnet: 192.168.1.0/24
   - Outside subnet: 203.0.113.0/24
   - NAT allows 192.168.1.x to appear as 203.0.113.1 externally.

2. **NAT Overload Mapping:**
   - PC1 (192.168.1.2) → 203.0.113.1:port1
   - PC2 (192.168.1.3) → 203.0.113.1:port2

3. **Access-list 1:**
   - Allows NAT for IP range 192.168.1.0/24 (ACL 1 permit 192.168.1.0 0.0.0.255)

10. Result & Interpretation

NAT (PAT) was successfully implemented, allowing internal hosts with private IP addresses to access an external host. The NAT router correctly translated and tracked connections through dynamic port mapping.

11. Follow-up Questions

1. WWhat is the difference between static NAT and PAT?
2. What command is used to verify NAT translations?
3. Why is NAT important in IPv4 networks?
4. What are the limitations of NAT?
5. What does the wildcard mask in the access-list represent?

13. Assessments

- **Quiz/Test:** NAT types, translation types, inside/outside terms
- **Lab Task:** Implement PAT using different IP ranges and test translation
- **Viva Questions:** Role of NAT in security, port mapping behavior
- **Assignment:** Compare NAT with IPv6 addressing and discuss how NAT becomes less relevant in IPv6 networks

14. Suggested Readings

- **Cisco CCNA Routing and Switching"** by Todd Lammle – Wiley
- **"Routing Protocols and Concepts"** – Cisco Networking Academy
- **"Data Communications and Networking"** by Behrouz A. Forouzan – McGraw Hill

Experiment No 10: Implementation of DHCP Protocol

1. Aim / Purpose of the Experiment

To implement and configure the Dynamic Host Configuration Protocol (DHCP) on a router or server so that IP addresses and related network configurations are automatically assigned to hosts

2. Learning Outcomes

After completing this experiment, students will be able to:

- Understand the purpose and working of the DHCP protocol.
- Configure a router or server as a DHCP server.
- Assign IP addresses dynamically to end devices.
- Verify DHCP IP allocation using simulation tools or command-line tools.
- Analyze DHCP lease information and address pools

3. Prerequisites

- Basic knowledge of IP addressing and subnetting
- Understanding of static vs. dynamic IP configuration
- Familiarity with router CLI configuration commands
- Experience with Cisco Packet Tracer or GNS3

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer or GNS3
- 1 Router, 1 Switch
- 3 PCs (end devices)
- DHCP-enabled router or external DHCP server

5. Introduction and Theory

Dynamic Host Configuration Protocol (DHCP) is an **application-layer protocol** used to automatically assign IP addresses and other network configuration parameters (like default gateway, DNS server) to client devices on a network.

Without DHCP, each client must be manually configured, which is inefficient in large networks.

Key Concepts:

- **DHCP Server**: Assigns IP addresses from a predefined pool.
- **DHCP Client**: Requests network configuration from the server.
- **Lease Time**: The period for which the IP is assigned
- **DHCP Discover → Offer → Request → Acknowledge (DORA)** process.

6. Operating Procedure

Network Design:

- Connect PC0, PC1, and PC2 to a switch. Connect the switch to a router's interface (e.g., Fa0/0).
- Configure the router to act as a DHCP server.

IP Planning:

- Network: 192.168.10.0/24

- Default Gateway: 192.168.10.1
- DHCP IP Pool Range: 192.168.10.10 to 192.168.10.100

Router Configuration:

```
Router> enable
Router# configure terminal
Router(config)# interface fa0/0
Router(config-if)# ip address 192.168.10.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit

Router(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
Router(config)# ip dhcp pool LAN_POOL
Router(dhcp-config)# network 192.168.10.0 255.255.255.0
Router(dhcp-config)# default-router 192.168.10.1
Router(dhcp-config)# dns-server 8.8.8.8
Router(dhcp-config)# lease 1
Router(dhcp-config)# exit
```

PC Configuration:

On each PC:

- Open IP Configuration
- Set to “DHCP” or “Obtain IP automatically”

Verification:

On the PCs: Run `ipconfig` (in Packet Tracer or CMD on real OS)

On the router:

```
show ip dhcp binding

show running-config
```

7. **Precautions and/or Troubleshooting**

- Exclude the gateway IP from the DHCP pool to avoid conflicts.
- Ensure that only one DHCP server exists on the subnet to avoid IP conflicts.
- Always verify interface status using `show ip interface brief`.
- Verify client devices are set to obtain IP automatically.

8. Observations

PC Name	Assigned IP	Subnet Mask	Default Gateway	DNS Server
PC0	192.168.10.10	255.255.255.0	192.168.10.1	8.8.8.8
PC1	192.168.10.11	255.255.255.0	192.168.10.1	8.8.8.8
PC2	192.168.10.12	255.255.255.0	192.168.10.1	8.8.8.8

9. Calculations & Analysis

1. **DHCP Address Pool Size**
   - Range: 192.168.10.10 to 192.168.10.100
   - Total IPs = 100 – 10 + 1 = **91 addresses**
2. **Subnet Mask**:
   - /24 → 255.255.255.0 → Supports 254 hosts in total
3. **Excluded Range**:
   - 192.168.10.1 to 192.168.10.9 → Reserved for gateway, servers, static hosts

10. Result & Interpretation

The DHCP protocol was successfully implemented on a router. The DHCP clients (PCs) received IP addresses, subnet masks, gateways, and DNS settings automatically from the DHCP server

11. Follow-up Questions

- What is the purpose of DHCP in a network?
- What happens if two DHCP servers are active in the same network?
- What is the difference between static and dynamic IP addressing?
- What is DHCP lease time? What happens when it expires?
- Explain the DORA process used by DHCP.

13. Assessments

- **Quiz/Test:** Fill-in-the-blanks, MCQs, and short answers on DHCP configuration, roles, and address allocation.
- **Practical Task:** Configure DHCP in a custom topology with different ranges.
- **Viva Questions:** DHCP message types, lease management, conflict handling.
- **Assignment:** Compare DHCP with manual IP assignment and analyze pros and cons.

14. Suggested Readings

- **"Routing and Switching Essentials – Cisco NetAcad"**

- **"CCNA Routing and Switching Study Guide"** by Todd Lammle – Wiley
- **"Computer Networks"** by Andrew S. Tanenbaum – Pearson

Experiment No 11: Configuration of Access Control Lists (ACLs) in Router and Basic Switch Configuration

1. Aim / Purpose of the Experiment

To configure standard and extended ACLs on a router to control network traffic and perform basic configuration on a switch including hostname, VLANs, and port assignments.

2. Learning Outcomes

After successful completion of this experiment, students will be able to:

- Understand the purpose and types of ACLs in routers.
- Create and apply standard and extended ACLs to filter traffic.
- Perform basic configuration tasks on a switch (hostname, VLANs, IP address).
- Verify ACL behavior and switch configuration using ping and show commands.
- Enhance network security using ACLs

3. Prerequisites

- Knowledge of IP addressing and subnetting
- Understanding of router and switch architecture
- Familiarity with Cisco IOS CLI commands
- Experience with basic LAN topology in Cisco Packet Tracer or GNS

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer or GNS3
- 1 Router, 1 Switch
- 3 PCs (Client1, Client2, and Server)
- Ethernet cables and patch cords
- Console terminal access (CLI)

5. Introduction and Theory

Access Control Lists (ACLs):ACLs are used in routers to control traffic flow by permitting or denying packets based on IP address, protocol type, port number, etc.

Types of ACLs:

- **Standard ACL**: Filters traffic based only on source IP address.
- **Extended ACL**: Filters traffic based on source/destination IP, protocol, and ports.

Standard ACL Range: 1–99
Extended ACL Range: 100–199

ACLs are applied in two directions:

- **Inbound** (as the packet enters the interface)
- **Outbound** (as the packet exits the interface)

Switch Configuration Basics:

- Assigning hostname
- Assigning VLANs and IP address
- Configuring access ports
- Saving configuration

6. Operating Procedure

A. Basic Switch Configuration:

Set hostname:

```
Switch> enable
Switch# configure terminal
Switch(config)# hostname SW1
```

Create VLAN and assign ports:

```
Switch(config)# vlan 10
Switch(config-vlan)# name STAFF
Switch(config)# interface range fa0/1 - 2
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 10
```

Assign IP address for switch management (SVI):

```
Switch(config)# interface vlan 1
Switch(config-if)# ip address 192.168.1.2 255.255.255.0
Switch(config-if)# no shutdown
```

Set default gateway:

```
Switch(config)# ip default-gateway 192.168.1.1
```

Save configuration:

```
Switch# write memory
```

B. Router ACL Configuration:

Scenario:

- Allow PC1 (192.168.1.2) to access Server (192.168.2.2)
- Deny PC2 (192.168.1.3) from accessing Server

Configure Router Interfaces:

```
Router> enable
```

```
Router# configure terminal
Router(config)# interface fa0/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config)# interface fa0/1
Router(config-if)# ip address 192.168.2.1 255.255.255.0
Router(config-if)# no shutdown
```

Configure Standard ACL:

```
Router(config)# access-list 10 permit 192.168.1.2
Router(config)# access-list 10 deny 192.168.1.3
Router(config)# access-list 10 deny any
```

Apply ACL on interface:

```
Router(config)# interface fa0/0
Router(config-if)# ip access-group 10 in
```

Verify ACL:

```
Router# show access-lists
Router# show running-config
```

7. Precautions and/or Troubleshooting

- Always apply ACLs in the correct direction and interface.
- Place more specific `permit` statements before `deny` statements.
- ACLs are processed top-down; once matched, no further rules are checked.
- Ensure switch VLANs and IPs are properly configured before testing ACLs.

8. Observations

Device	IP Address	Access to Server	ACL Applied	Action
PC1	**192.168.1.2**	**Yes**	**Yes (Permit)**	**Allowed**
PC2	**192.168.1.3**	**No**	**Yes (Deny)**	**Blocked**
Server	**192.168.2.2**	**-**	**-**	**Responded only to PC1**

9. Calculations & Analysis

- **Subnet**: 192.168.1.0/24 and 192.168.2.0/24 = 255.255.255.0
- **Standard ACL List**: Deny 192.168.1.3, Permit 192.168.1.2
- **Wildcard mask for host**: `0.0.0.0` for exact IP match

10. Result & Interpretation

The router successfully filtered traffic using standard ACL, allowing only the permitted IP to access the server. The switch was configured with VLANs and management IP and responded to ping correctly

11. Follow-up Questions

1. What is the difference between standard and extended ACLs?
2. Where should you place standard ACLs: close to source or destination?
3. Can ACLs be used for security? How?
4. What would happen if the ACL ends without a "deny any"?
5. What is the default behavior of an ACL if a packet does not match any line?

13. Assessments

- **Quiz/Test:** Identify correct ACL commands, syntax, and behavior.
- **Practical Task:** Configure both standard and extended ACLs.
- **Viva Questions:** Purpose of wildcard masks, directions (in/out), VLAN importance.

14. Suggested Readings

- "**CCNA Routing and Switching Study Guide"** by Todd Lammle – Wiley
- **"Routing and Switching Essentials"** – Cisco Networking Academy
- **Cisco ACL Configuration Guide** – https://www.cisco.com

Experiment No 12:Implementation of Telnet

1. Aim / Purpose of the Experiment

To configure and examine Telnet (remote login) operation in a network using routers and switches, and understand its working, limitations, and security concerns

2. Learning Outcomes

After completing this lab, students will be able to:

- Understand how Telnet operates and its role in remote management.
- Configure Telnet access on routers/switches.
- Access and manage network devices remotely using Telnet.
- Recognize the security implications of using Telnet.
- Use basic commands to monitor and manage Telnet sessions

3. Prerequisites

- Understanding of TCP/IP model and IP addressing
- Familiarity with command-line interface (CLI) configuration
- Basic knowledge of device access protocols (e.g., Telnet, SSH)
- Experience with Cisco IOS commands

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer or GNS3
- 1 Router, 1 Switch
- 2 PCs (one for configuration, one for remote access)
- Ethernet cables and console access

5. Introduction and Theory

Telnet Characteristics:

- Plain-text protocol (insecure)
- Requires device IP and password for access
- Supports remote management and configuration
- Should be replaced with **SSH** in secure environments

Common Telnet Commands:

- telnet [IP address] — Initiates a session
- show users — Displays active Telnet sessions
- disconnect [line #] — Disconnects a session
- exit — Closes session

6. Operating Procedure

Step 1: Network Topology Setup

- Connect PC0 and Router0 to a switch.
- Assign Router0 an IP address on its interface connected to the switch.
- Connect PC1 to the same switch (for Telnet access).

Step 2: Configure IP on PC0 and PC1

- PC0: 192.168.1.2 /24, Gateway: 192.168.1.1
- PC1: 192.168.1.3 /24, Gateway: 192.168.1.1

Step 3: Configure Router for Telnet

```
Router> enable
Router# configure terminal
Router(config)# interface fa0/0
Router(config-if)# ip address 192.168.1.1 255.255.255.0
Router(config-if)# no shutdown
```

```
Router(config-if)# exit

Router(config)# line vty 0 4
Router(config-line)# password cisco
Router(config-line)# login
Router(config-line)# exit

Router(config)# enable password class
Router(config)# exit
```

Step 4: Test Telnet from PC1

- On PC1:
  Open command prompt → `telnet 192.168.1.1`
    - Enter password: `cisco`
    - You are now remotely accessing the router CLI

7. Precautions and/or Troubleshooting

- Ensure all devices are on the same subnet.
- Telnet must be enabled using VTY lines and a password.
- Verify that the interface is up (no shutdown).
- Use a strong password in real scenarios.
- Remember: **Telnet sends data in plaintext**, use **SSH** for secure environments

8. Observations

Device	Action	Observation
Router	**VTY line configured**	**Telnet access allowed**
PC1	**Telnet command used**	**CLI access granted**
PC1	**Password entered**	**Login successful**
Router	**`show user` executed**	**PC1's session displayed**

9. Calculations & Analysis

This lab involves logical IP planning and session tracking rather than mathematical calculations. However, you may include:

- **IP Planning**: Ensure PCs and Router are in the same subnet
- **Number of Telnet Sessions Allowed**: 5 (VTY 0 to 4

10. Result & Interpretation

Telnet was successfully configured on the router, and remote login was achieved from PC1 using the Telnet client. The session was established over TCP port 23 and demonstrated remote device access.

11. Follow-up Questions

1. What port does Telnet use?
2. How is Telnet different from SSH?
3. Why is Telnet considered insecure?
4. What are VTY lines in router configuration?
5. How many simultaneous Telnet sessions are supported by default?

13. Assessments

- **Quiz/Test:** Identify Telnet commands, port numbers, security issues
- **Practical Task:** Configure Telnet on a router and access it from a PC
- **Viva Questions:** Purpose of `line vty`, `show users`, Telnet vs. SSH.

14. Suggested Readings

- **"Routing and Switching Essentials"** – Cisco Networking Academy
- **"CCNA Routing and Switching Study Guide"** by Todd Lammle – Wiley
- **Cisco IOS CLI User Guide for Telnet and SSH** – https://www.cisco.com

Experiment No 13: Socket Programming – Client-Server Communication using TCP

1. Aim / Purpose of the Experiment

To implement a basic client-server communication system using TCP sockets where the client sends a message to the server and receives a response.

2. Learning Outcomes

By the end of this experiment, students will be able to:

- Understand the concept of socket communication using TCP.
- Write programs to establish a TCP connection between a client and a server.
- Transmit and receive data through a reliable, connection-oriented protocol.
- Understand the importance of socket binding, listening, accepting, and connecting.
- Troubleshoot connection errors and understand TCP port/address dependencies

3. Prerequisites

- Knowledge of Python or Java programming
- Understanding of TCP/IP networking concepts
- Familiarity with client-server architecture
- Basics of socket APIs and function

4. Materials / Equipment / Apparatus / Devices / Software Required

- Python (version 3.x) or Java SDK
- A computer system or two systems connected via LAN
- Any IDE or terminal-based environment (VS Code, IDLE, Eclipse)
- Internet or LAN connection (for testing real network communication)

5. Introduction and Theory

Socket Programming enables communication between two systems or processes using endpoints called **sockets**. A **TCP socket** provides a reliable, connection-oriented communication channel.

Key TCP Socket Functions:

- `socket()` – Create a socket
- `bind()` – Attach socket to IP and port (server side)
- `listen()` – Wait for incoming connections
- `accept()` – Accept a connection from client
- `connect()` – Initiate a connection (client side)
- `send()` / `recv()` – Data transmission and reception
- `close()` – Terminate socket

TCP Communication Process:

1. Server creates a socket and binds to an IP and port.
2. Server listens for incoming connection requests.
3. Client creates a socket and connects to the server's socket.
4. Data is exchanged using send() and recv() (Python).
5. Both ends close the socket after communication.

6. Operating Procedure

1. **import socket**
2. **# Create socket object**
3. **server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)**
4. **# Bind to local IP and port**
5. **server_socket.bind(('127.0.0.1', 12345))**
6. **server_socket.listen(1)**
7. **print("Server is listening...")**

```
# Accept client connection
conn, addr = server_socket.accept()
print(f"Connected with {addr}")
# Receive message from client
data = conn.recv(1024).decode()
print("Received from client:", data)
# Send response to client
conn.send("Message received.".encode())
# Close connection
conn.close()
server_socket.close()
```

2. Client Side Code (Python):

```
import socket
# Create socket object
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Connect to server
client_socket.connect(('127.0.0.1', 12345))
# Send message
client_socket.send("Hello Server!".encode())
# Receive response
data = client_socket.recv(1024).decode()
print("Received from server:", data)
# Close connection
client_socket.close()
```

Steps to Run:

1. Open two terminal/IDE windows.
2. Run the server script first.
3. Then run the client script.
4. Observe the message exchange

7. Precautions and/or Troubleshooting

- Always run the server before the client to avoid connection errors.
- Ensure firewall or antivirus doesn't block TCP ports.
- Use `127.0.0.1` or localhost for local testing.
- Use the same port number on both client and server.
- Don't forget to close sockets to release resources.

8. Observations

Component	Observation	Component	Observation
Server Start	"Server is listening..." printed	Server Start	"Server is listening..." printed
Client Connect	"Connected with ('127.0.0.1', port)" on server	Client Connect	"Connected with ('127.0.0.1', port)" on server
Message Received	Message appears on both client and server	Message Received	Message appears on both client and server
Socket Closed	No errors; both sides close cleanly	Socket Closed	No errors; both sides close cleanly

9. Calculations & Analysis

No mathematical calculations are needed.
However, you can log/measure:

- **Latency**: Time between send() and recv()
- **Buffer Size**: How many bytes are sent in one message (recv(1024))

10. Result & Interpretation

Client and server were able to establish a TCP socket connection successfully. The client sent a message, the server responded, and both sides terminated the connection gracefully

11. Follow-up Questions

1. What is the difference between TCP and UDP sockets?
2. Why must the server listen and accept connections before the client connects?
3. What are the possible errors during socket programming and how to handle them?
4. How does TCP ensure reliable data delivery?
5. Can two clients connect to the same server simultaneously in this model? If not, how to modify?

13. Assessments

- **Quiz/Test:** Functions of `bind()`, `listen()`, `recv()`, port numbers, error handling
- **Practical Task:** Modify server to handle multiple clients (use threads)

- **Viva Questions:** TCP 3-way handshake, differences between TCP and UDP sockets

14. Suggested Readings

- **"Foundations of Python Network Programming"** by John Goerzen – Apress
- **"Computer Networks"** by Andrew Tanenbaum – Pearson
- **Python `socket` module documentation** – https://docs.python.org/3/library/socket.html

Experiment No 14: Socket Programming -UDP Echo Client and UDP Echo Server

1. Aim / Purpose of the Experiment

To implement a UDP-based Echo Client and Echo Server using socket programming, where the server sends back the same message received from the client

2. Learning Outcomes

By the end of this experiment, students will be able to:

- Understand the structure and behavior of UDP socket communication.
- Differentiate between TCP and UDP sockets.
- Write UDP-based client-server programs for message transmission.
- Test message delivery, loss, and delays in connectionless environments.
- Handle data packets using datagram sockets in Python or Java

3. Prerequisites

- Understanding of socket programming basics
- Familiarity with UDP (User Datagram Protocol)
- Programming experience in Python or Java
- Concept of IP addressing and port.

4. Materials / Equipment / Apparatus / Devices / Software Required

- Python (3.x) or Java SDK
- IDE or Terminal (VS Code, IDLE, Eclipse, or Command Prompt)
- Single or two computer systems (optional)
- Working network or localhost setup

5. Introduction and Theory

UDP (User Datagram Protocol) is a connectionless, unreliable transport-layer protocol used for fast data transmission with minimal overhead.

Characteristics of UDP:

- Connectionless and stateless

- No handshaking or session establishment
- Low latency, suitable for real-time applications (VoIP, video)
- No guaranteed delivery or order

UDP Echo Server: Listens for datagrams, then sends back the same data to the sender.

UDP Echo Client: Sends a message to the server and waits for the echoed message.

6. Operating Procedure

Server Side Code (UDP Echo Server – Python):

```
import socket
# Create UDP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
# Bind to local IP and port
server_socket.bind(('127.0.0.1', 12345))
print("UDP Echo Server is ready...")
while True:
    # Receive data and sender address
    data, addr = server_socket.recvfrom(1024)
    print(f"Received from {addr}: {data.decode()}")
    # Send back the same data
    server_socket.sendto(data, addr)
```

2. Client Side Code (UDP Echo Client – Python):

```
import socket
# Create UDP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
# Server details
server_address = ('127.0.0.1', 12345)
# Message to send
message = input("Enter message to send: ")
client_socket.sendto(message.encode(), server_address)
# Receive echoed response
data, _ = client_socket.recvfrom(1024)
print("Received from server:", data.decode())
# Close the socket
client_socket.close()
```

Steps to Run:

1. Run the server script first.
2. Run the client script, enter a message.
3. Observe the message echoed back from server to client.

7. Precautions and/or Troubleshooting

- Ensure both scripts use the same port number and IP address.
- Run the server **before** the client to avoid connection failure.
- UDP does not guarantee delivery — retry if no response.
- Close sockets properly to free up system resources.
- Avoid using firewall-blocked ports

8. Observations

Action	Output/Observation
Client sends message	**Message is sent as a UDP datagram**
Server receives data	**Prints message and client's IP, port**
Server echoes message	**Sends same message back to client**
Client receives echo	**Prints echoed response**

9. Calculations & Analysis

Although UDP is non-deterministic, you can observe and log:

- **Packet Round-Trip Time (RTT)** (using timestamps)
- **Packet Size** in bytes
- **Port numbers** used by client and server

Example (Python enhancement for RTT):

```
import time
start = time.time()
client_socket.sendto(message.encode(), server_address)
data, _ = client_socket.recvfrom(1024)
end = time.time()
print("RTT: ", (end - start), "seconds")
```

10. Result & Interpretation

UDP-based echo client and server were successfully implemented. The server echoed back the client's message, demonstrating connectionless and stateless data transmission

11. Follow-up Questions

1. What is the major difference between UDP and TCP communication?
2. Why doesn't UDP guarantee packet delivery?
3. What happens if the server is not running when the client sends data?
4. How can you simulate UDP packet loss in testing?
5. What are real-world applications of UDP-based communication?

13. Assessments

- **Quiz/Test:** UDP protocol characteristics, port numbers, socket functions
- **Practical Task:** Modify the echo server to reverse the message before echoing.
- **Viva Questions:** Datagram vs. Stream sockets, use cases for UDP, error handling

14. Suggested Readings

- **Python Network Programming Cookbook"** by Dr. M. V. Vishal – Packt
- **"Computer Networking: A Top-Down Approach"** by Kurose and Ross – Pearson
- **Python `socket` module documentation**: https://docs.python.org/3/library/socket.html

Experiment No 15: Implementation of TCP and UDP Protocols using Cisco Packet Tracer

1. Aim / Purpose of the Experiment

To simulate and understand the functioning of TCP and UDP protocols in a network scenario using Cisco Packet Tracer by observing their behavior in client-server communication and packet delivery.

2. Learning Outcomes

After completing this experiment, students will be able to:

- Understand the difference between TCP and UDP protocols.
- Simulate TCP-based services like HTTP, FTP, and Telnet.

- Simulate UDP-based services like DNS and TFTP.
- Analyze packet delivery and protocol behavior using simulation mode.
- Observe protocol characteristics such as reliability, connection setup, and data loss

3. Prerequisites

- Basic knowledge of OSI and TCP/IP models
- Understanding of TCP vs. UDP characteristics
- Familiarity with Cisco Packet Tracer environment
- IP addressing and basic network device configuration knowledge.

4. Materials / Equipment / Apparatus / Devices / Software Required

- Cisco Packet Tracer (version 7.2 or higher)
- 2 or more PCs
- 1 Router
- 1 Switch
- 1 Server (configured for HTTP, FTP, DNS, or TFTP)
- Network cables (Ethernet)

5. Introduction and Theory

Transmission Control Protocol (TCP):

TCP is a **connection-oriented**, **reliable** protocol that ensures ordered delivery of data using acknowledgments and retransmissions. It is used in applications such as **HTTP**, **FTP**, **Telnet**, and **SMTP**.

User Datagram Protocol (UDP):

UDP is a **connectionless**, **unreliable** protocol which sends datagrams without acknowledgment. It is used in applications like **DNS**, **TFTP**, **VoIP**, and **streaming**.

Key Differences:

Feature	TCP	UDP
Connection	Connection-oriented	Connectionless
Reliability	Reliable (acknowledgments)	Unreliable
Speed	Slower (more overhead)	Faster (less overhead)
Use Cases	HTTP, FTP, Telnet	DNS, TFTP, VoIP, Streaming

6. Operating Procedure

Part A: TCP-based Communication (HTTP/FTP/Telnet)

1. **Create a network topology:**
   - 1 Router, 1 Switch, 2 PCs, and 1 Server
   - Assign static IPs to all devices within the same subnet (e.g., 192.168.1.0/24)

2. **Configure the Server:**
   - Click on the Server → Services tab
   - Enable **HTTP**, **FTP**, and **Telnet** services

3. **Configure the PCs:**
   - Use IP Configuration to assign IP address and default gateway
   - Use PC's web browser to access http://[server IP] for HTTP
   - Use Command Prompt to telnet [server IP]
   - Use ftp [server IP] for FTP testing

4. **Observe behavior in Simulation Mode:**
   - Switch to Simulation mode
   - Add TCP filter
   - Send HTTP request or FTP request and observe 3-way handshake

Part B: UDP-based Communication (DNS/TFTP)

1. **Configure the Server:**

   - Enable **DNS** and **TFTP** services
   - In DNS, add a record (e.g., `www.example.com → 192.168.1.10`)

2. **Configure a PC with DNS:**

   - Assign DNS server IP in IP Configuration

3. **Test UDP Services:**
   - Use web browser to access `http://www.example.com`
   - Use TFTP application (optional) to upload/download files from server

4. **Observe packet behavior:**
   - Switch to Simulation mode
   - Add UDP filter

- Generate DNS or TFTP request and observe packet delivery (no ACKs)

7. Precautions and/or Troubleshooting

- Ensure devices are powered on and cables connected correctly.
- IP addresses and gateway must be correctly configured.
- In simulation mode, apply appropriate protocol filters (TCP/UDP).
- Always start server services before generating traffic.
- Use unique port numbers and hostnames in DNS configuration
-

8. Observations

Protocol	Service	Protocol Type	Reliability	Observed Behavior
HTTP	**Web**	**TCP**	**Reliable**	**3-way handshake**
FTP	**File**	**TCP**	**Reliable**	**Auth and file tx**
Telnet	**CLI**	**TCP**	**Reliable**	**CLI remote login**
DNS	**Naming**	**UDP**	**Unreliable**	**Query-response**
TFTP	**File**	**UDP**	**Unreliable**	**Quick transfer**

9. Calculations & Analysis

Though TCP and UDP are protocol-layer functions, here are some conceptual calculations:

1. **TCP Overhead** = IP Header (20 bytes) + TCP Header (20 bytes) = 40 bytes
2. **UDP Overhead** = IP Header (20 bytes) + UDP Header (8 bytes) = 28 bytes
3. **Throughput estimation** = (Data size - overhead) / Total transmission time
4. **RTT (Round Trip Time)** can be approximated in Simulation view

10. Result & Interpretation

TCP and UDP protocols were successfully implemented and simulated using Cisco Packet Tracer. The differences in connection behavior, reliability, and service type were observed clearly through various scenarios

11. Follow-up Questions

1. What is the role of port numbers in TCP and UDP communication?
2. Which services in the OSI model use TCP and which use UDP?
3. What is a 3-way handshake? Why is it not used in UDP?
4. Why is UDP preferred in real-time communication like VoIP?
5. What would happen if the server response is lost in TCP vs. UDP

13. Assessments

- **Quiz/Test:** Identify service-port mappings (e.g., HTTP–80, DNS–53)
- **Practical Task:** Simulate both TCP and UDP failures and discuss recovery
- **Viva Questions:** TCP flags, packet structure, reliability, use cases

14. Suggested Readings

- **"CCNA Routing and Switching Study Guide"** by Todd Lammle – Wiley
- **Cisco Networking Academy Packet Tracer Labs**
- **"Computer Networking: A Top-Down Approach"** by Kurose and Ross – Pearson