Program 1: Familiarizing with Windows Network Commands

Aim: To study and execute basic Windows network commands for understanding system and network configurations.

Procedure:
1. Switch on the computer and open the Windows operating system.
2. Go to the Start Menu, search for Command Prompt, and open it.
3. Type ipconfig and press Enter to display the system’s IP address, subnet mask, and default gateway.
4. Use ping (e.g., ping www.google.com) to check connectivity with another host or server.
5. Run tracert to view the route taken by packets to reach the destination.
6. Type netstat to display active network connections and ports in use.
7. Try other commands like nslookup, arp –a, and hostname to understand their functionality.
8. Observe and note the outputs of each command to verify network connectivity and configuration.
9. Conclude the experiment by analyzing how each command helps in troubleshooting and managing a network.

Program 2: Establishing a Local Area Network (LAN)

Aim: To create and configure a LAN using Packet Tracer and test communication between connected devices.

Procedure:
1. Open Cisco Packet Tracer and drag required devices such as PCs and a switch onto the workspace.
2. Connect all PCs to the switch using copper straight-through cables.
3. Assign IP addresses to each PC by going to Desktop → IP Configuration (e.g., PC0: 192.168.1.1, PC1: 192.168.1.2, etc.).
4. Ensure that all PCs have the same subnet mask and belong to the same network.
5. Save the configuration and check physical connections.
6. To test communication, go to Desktop → Command Prompt in one PC and use the ping command to contact another PC.
7. Alternatively, use Add Simple PDU (mail icon) to send test packets between PCs.
8. Verify whether the packets are delivered successfully.
9. If unsuccessful, recheck cabling and IP assignments.
10. Conclude by confirming that the LAN is working and all devices can communicate.

Program 3: Connecting Two LANs using Router with Static Route

Aim: To connect two LANs using a router and configure static routes for communication.

Procedure:
1. Open Cisco Packet Tracer and drag two sets of PCs with switches to form two separate LANs.
2. Place a router in between the two LANs.
3. Connect each switch to a different interface of the router using copper cables.
4. Assign proper IP addresses to PCs in each LAN, ensuring they are in different networks (e.g., LAN1: 192.168.1.0/24, LAN2: 192.168.2.0/24).
5. Configure the router by clicking it, going to CLI mode, and assigning IP addresses to FastEthernet interfaces.
6. Enable the interfaces using the no shutdown command.
7. Add static routes in the router so that it knows the path to reach the other network.
8. Save the configuration and test connectivity between PCs of different LANs using the ping command or Add PDU.
9. If pings fail, verify cabling, IP settings, and routing commands.

10. Conclude the experiment after confirming successful communication between both LANs.

Program 4: Multi-Routing Connection with Static Router

Aim: To connect multiple LANs using routers and configure static routing so that communication can occur between all networks.

Procedure:
1. Open Cisco Packet Tracer and place three or more LANs, each consisting of PCs and a switch.
2. Place routers to connect these LANs in a chain or mesh as per the required topology.
3. Connect the PCs to switches and switches to routers using copper cables.
4. Assign unique IP addresses to each LAN, making sure every LAN belongs to a different subnet.
5. Configure router interfaces by assigning IP addresses to FastEthernet and Serial interfaces.
6. Enable each router interface with the no shutdown command.
7. Add static routes in each router to specify the path to reach all other networks. For example: Router(config)#ip route
8. After configuration, test communication by pinging from a PC in one LAN to PCs in other LANs.
9. If ping fails, verify IP assignment, cabling, and static route entries.
10. Conclude after confirming all LANs are able to communicate through routers with static routing.

Program 5: Implementing Mini Search Engine

Aim: To design and implement a simple web-based mini search engine using HTML and JavaScript that redirects users to specific web pages.

Procedure:
1. Open any text editor such as Notepad, Notepad++, or VS Code.
2. Start writing a basic HTML structure with , , and tags.
3. Inside the , include a JavaScript function that checks the input value entered by the user.
4. The function should redirect to a URL based on the user's choice (for example, Google for "WEB" or SRM website for "SRM").
5. In the , create a form with a text box for entering search keywords and radio buttons for selecting search options.
6. Add a submit button that calls the JavaScript function when clicked.
7. Save the file with the extension .html.
8. Open the file in a web browser.
9. Enter a keyword, select the option, and click the search button to test whether redirection is working.
10. Conclude after verifying that the program correctly redirects to the appropriate search page.

Program 6: Implementing Simple Web Server

Aim: To implement a simple web server in Java that can handle client requests and serve HTML or image files.

Procedure:
1. Open a Java IDE or a text editor such as Notepad++ or VS Code.
2. Create a new Java file, e.g., WebServer1.java.
3. Import necessary Java classes such as ServerSocket, Socket, BufferedReader, PrintStream, and FileInputStream.
4. In the main method, create a ServerSocket object listening on a specific port (e.g., 5555).
5. Continuously accept client requests using the accept() method.
6. Read the client request using BufferedReader. Extract the request type (GET) and requested file

path.
7. Open the requested file from the server directory and determine its content type (text/html, image/jpeg, etc.).
8. Send appropriate HTTP response headers followed by file data using PrintStream.
9. Handle errors by returning a “Bad Request” message if the file is not found.
10. Save, compile, and run the program. Test it by entering http://localhost:5555/ in a browser.

Program 7: Designing Various Topologies using Cisco Packet Tracer

Aim: To design and analyze different network topologies (Star, Bus, Ring, Mesh, Hybrid) using Cisco Packet Tracer.

Procedure:
1. Open Cisco Packet Tracer and select required network devices such as PCs, switches, and hubs.
2. For Star topology, connect all PCs to a central switch.
3. For Bus topology, connect PCs along a single backbone using hubs.
4. For Ring topology, connect PCs in a closed loop.
5. For Mesh topology, connect each PC to all others directly.
6. For Hybrid topology, combine two or more of the above topologies.
7. Assign IP addresses to all PCs in each topology.
8. Test communication between devices using the ping command or Add Simple PDU tool.
9. Record results for each topology and note advantages/disadvantages.
10. Conclude by verifying which topologies provide better connectivity, fault tolerance, and cost efficiency.

Program 8: FTP Server Simulation using Cisco Packet Tracer

Aim: To simulate and configure an FTP server in Cisco Packet Tracer and verify file transfer operations.

Procedure:
1. Open Cisco Packet Tracer and place one server and at least one PC.
2. Connect the PC and server using a switch.
3. Assign IP addresses to both devices.
4. Select the server and enable FTP service in the Services tab. Disable unnecessary services.
5. Create a new FTP user by providing a username, password, and permissions (Read/Write/Delete).
6. On the PC, open the Command Prompt and type ftp to connect to the FTP server.
7. Enter the username and password created earlier.
8. Perform FTP operations such as listing files, uploading, and downloading.
9. Verify successful transfer of files by checking the server directory.
10. Conclude the experiment by confirming that FTP login and file operations were successful.

Program 9: DNS Server Simulation using Cisco Packet Tracer

Aim: To configure and simulate a DNS server in Cisco Packet Tracer that maps domain names to IP addresses.

Procedure:
1. Open Cisco Packet Tracer and place a DNS server, a web server, and at least one client PC.
2. Connect all devices using a switch.

3. Assign IP addresses to all devices.
4. On the DNS server, open Services → DNS and enable DNS.
5. Add a DNS record mapping a domain name (e.g., www.example.com) to the IP address of the web server.
6. Configure the web server with an index.html page.
7. On the client PC, configure the DNS server’s IP in the DNS field.
8. Open a web browser in the client PC and type the domain name.
9. Verify that the page hosted on the web server loads successfully.
10. Conclude by confirming that the DNS successfully resolved the domain name to the server IP.

Program 10: ARP Simulation using Cisco Packet Tracer

Aim: To simulate and understand the working of Address Resolution Protocol (ARP) using Cisco Packet Tracer.

Procedure:
1. Open Cisco Packet Tracer and create a network with a PC, switch, and server.
2. Assign IP addresses to all devices.
3. Go to Simulation Mode to visualize ARP working.
4. On the PC, open Command Prompt and type arp –a to check ARP table (initially empty).
5. From the PC, ping the server using its IP address.
6. Observe that ARP request packets are broadcasted to find the MAC address corresponding to the IP.
7. The server replies with its MAC address, which is added to the ARP table of the PC.
8. Verify updated ARP entries using arp –a again.
9. Repeat the process with multiple PCs to understand ARP table population.
10. Conclude that ARP successfully maps IP addresses to MAC addresses for communication.