LAB: 1

AIM: Study about Network Cabling

Theory:

In this exercise, we used **Cat6 cables**, which are a type of twisted-pair cable designed to support higher bandwidths than Cat5e cables. The cable has eight individual wires twisted into four pairs. **Color coding** is important to ensure proper connectivity and data transmission. There are two main wiring standards for Ethernet cables:

- **T568A**
- **T568B**

Each of these standards determines the order in which the eight wires are arranged. Consistency in wiring standards is crucial for proper network functioning.

Color Coding for T568B Standard:

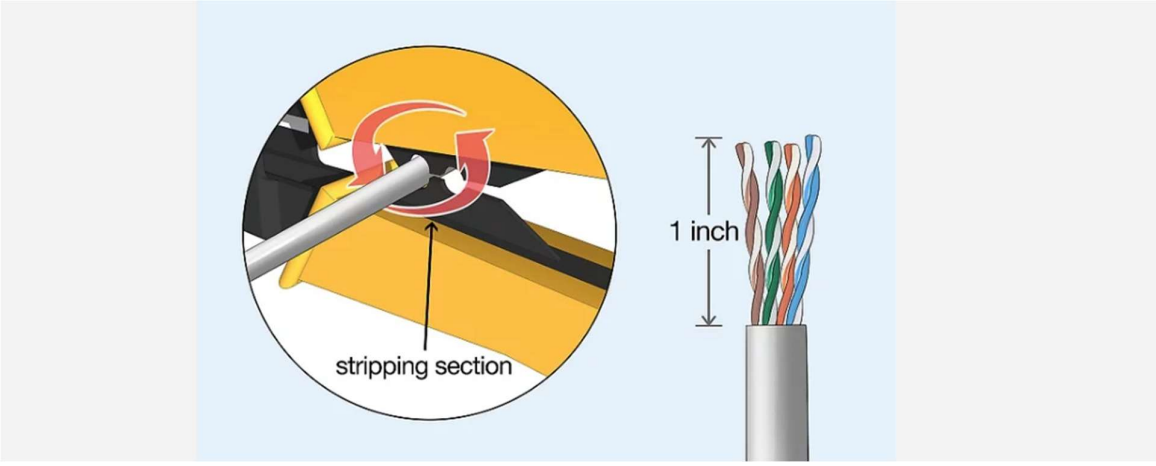
1. White/Orange
2. Orange
3. White/Green
4. Blue
5. White/Blue
6. Green
7. White/Brown
8. Brown

Color Coding for T568A Standard:

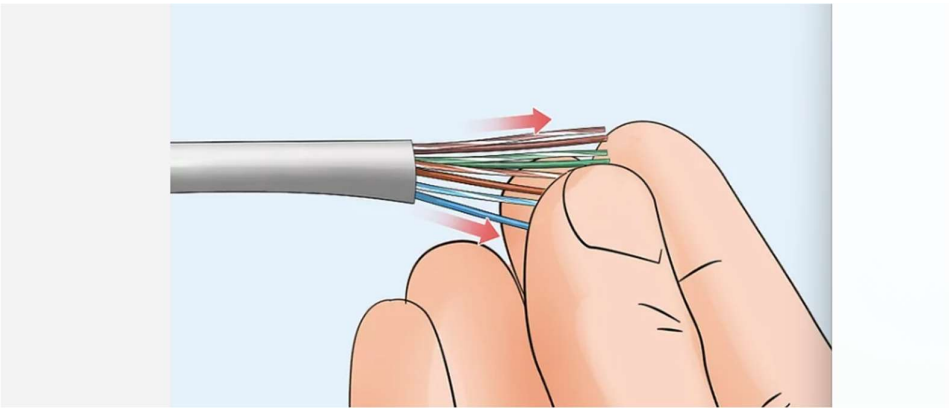
1. White/Green
2. Green
3. White/Orange
4. Blue
5. White/Blue
6. Orange
7. White/Brown
8. Brown

Procedure:

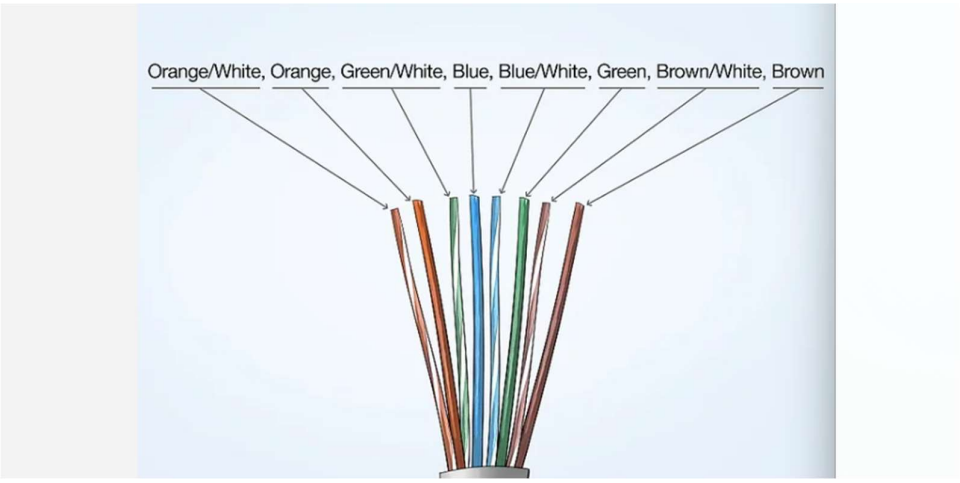
1. **Uncovering the Cable**:
    - Strip about 1-2 inches of the outer insulation from the Cat6 cable using a cable stripper or cutter, revealing the four twisted pairs of wires inside.



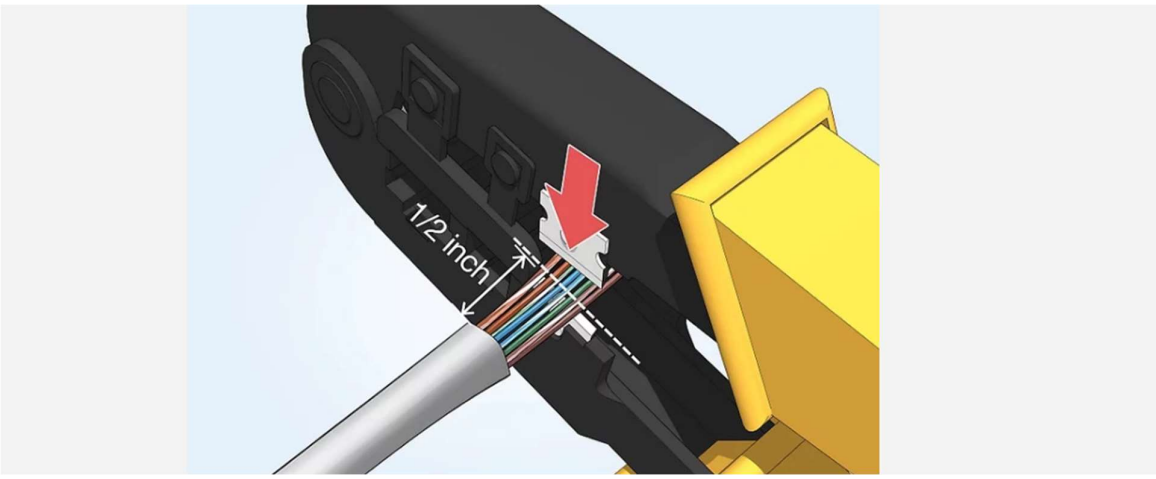
2. **Sorting and Straightening**:
    - Untwist the pairs of wires and straighten them as much as possible for easy insertion into the RJ45 connector.



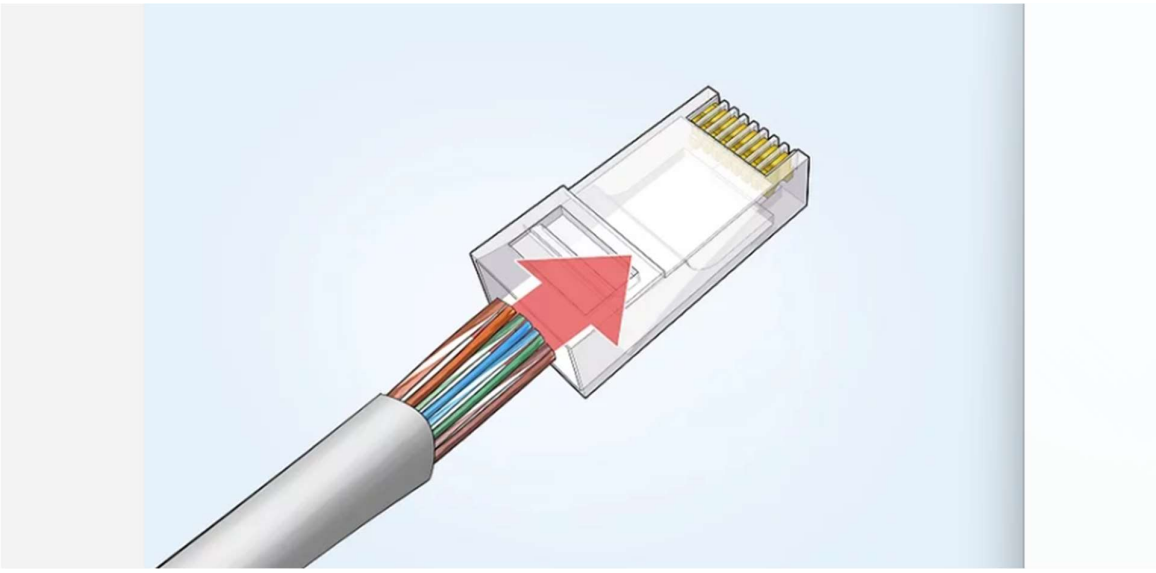
3. **Color Coding**:
    - Arrange the wires according to the **T568B** color code (or T568A if chosen).
    - Make sure to align the wires in the correct order before insertion.



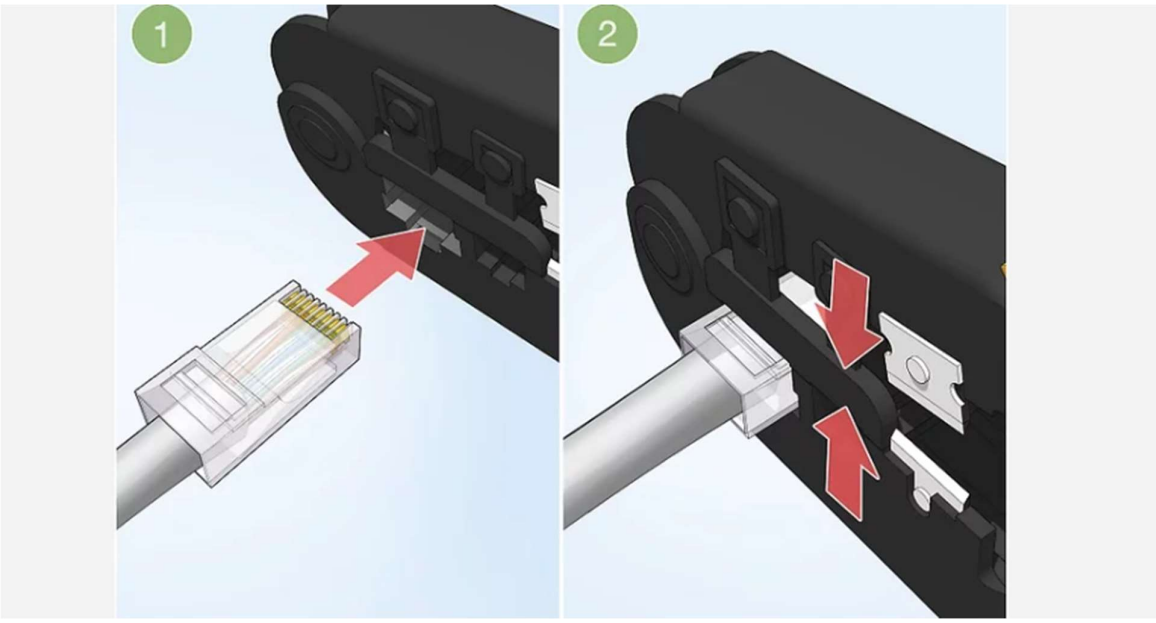
4. **Trimming the Wires**:
    - Trim the ends of the wires evenly to ensure they fit neatly into the RJ45 connector.



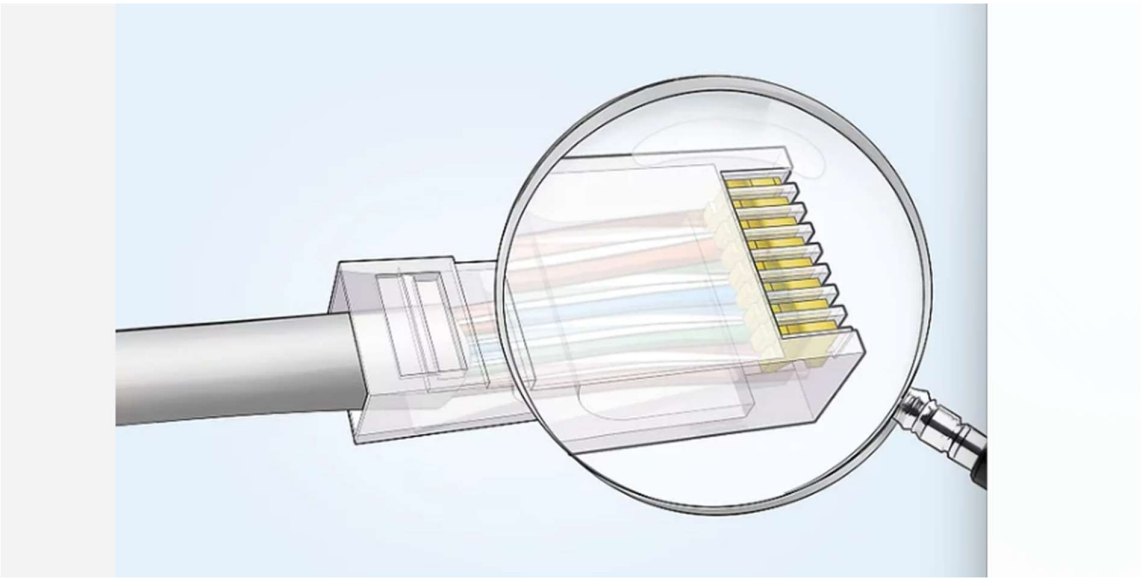
5. **Inserting into the RJ45 Connector**:
    - Insert the sorted wires into the RJ45 connector, ensuring that each wire goes into the correct channel and reaches the metal contacts.



6. **Clamping (Crimping)**:
    - Place the RJ45 connector into the crimping tool and apply firm pressure to clamp the connector onto the cable, securing the wires and making electrical contact.



7. **Testing the Cable**:
    - Use a cable tester to verify that the cable has been properly wired and that all connections are working correctly.



Conclusion:

In this exercise, we learned how to correctly identify the color coding for Cat6 cables and apply the T568B wiring standard. We successfully built a functional Ethernet cable by following the correct procedure, from uncovering the cable to crimping the RJ45 connector.

Discussion:

During the process, one of the main difficulties we faced was **untwisting and straightening the wires**, as they tend to tangle or become difficult to align. Sorting the wires according to the color coding standard also required careful attention to avoid mistakes. Another challenge was **clamping the connector**; ensuring all the wires were fully inserted and maintaining alignment while using the crimping tool was crucial. We overcame these difficulties by taking extra care to straighten the wires properly and double-checking the color arrangement before clamping. This careful approach helped us complete the task successfully.

LAB: 2

Theory:

Networking commands are essential tools used by system administrators and network engineers to troubleshoot, configure, and monitor network-related issues. These commands help in viewing the network configuration of a machine, testing network connectivity, identifying issues, and more. Whether it's determining the IP address, checking connectivity to a remote server, or resolving domain names, these commands are invaluable for managing both local and wide-area networks.

Basic Networking Commands:

1. **ipconfig** (Windows) / **ifconfig** (Linux/macOS)
    - **Description**: Displays the network configuration of the system, including IP address, subnet mask, and gateway.
    - **Syntax**:
        - Windows: ipconfig
        - Linux/macOS: ifconfig

```
C:\Users\Dell>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2ee6:2f29:975c:cb%20
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : worldlink.com.np
   IPv6 Address. . . . . . . . . . . : 2400:1a00:b040:f08b::2
   IPv6 Address. . . . . . . . . . . : 2400:1a00:b040:f08b:7a1d:91d0:9304:90de
   Temporary IPv6 Address. . . . . . : 2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de
   Link-local IPv6 Address . . . . . : fe80::10a1:1c54:6fd0:d05c%7
   IPv4 Address. . . . . . . . . . . : 192.168.1.112
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::1%7
                                       192.168.1.254

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

2. **ping**
    - **Description**: Sends ICMP echo requests to a specific host to test network connectivity and measure the round-trip time.
    - **Syntax**: ping [hostname/IP address]
        - Example: ping www.google.com

```
C:\Users\Dell>ping www.google.com

Pinging www.google.com [2404:6800:4002:816::2004] with 32 bytes of data:
Reply from 2404:6800:4002:816::2004: time=38ms
Reply from 2404:6800:4002:816::2004: time=33ms
Reply from 2404:6800:4002:816::2004: time=40ms
Reply from 2404:6800:4002:816::2004: time=53ms

Ping statistics for 2404:6800:4002:816::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 33ms, Maximum = 53ms, Average = 41ms
```

3. **tracert** (Windows) / **traceroute** (Linux/macOS)
    - **Description**: Traces the route packets take from your computer to a destination. It shows each hop along the way and helps identify where delays or failures occur.
    - **Syntax**:
        - Windows: tracert [hostname/IP address]
        - Linux/macOS: traceroute [hostname/IP address]
        - Example: tracert www.youtube.com

```
C:\Users\Dell>tracert www.youtube.com

Tracing route to youtube-ui.l.google.com [2404:6800:4002:81a::200e]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  2400-1A00-B040.ip6.wlink.com.np [2400:1a00:b040:f08b::1]
  2     3 ms     8 ms     7 ms  2400-1A00-B1A4.ip6.wlink.com.np [2400:1a00:b1a4:0:8100:0:589f:55f7]
  3     *        *        *     Request timed out.
  4     9 ms    13 ms    30 ms  2400:1a00:0:41::170
  5     7 ms     7 ms    13 ms  2400:1a00:0:41::128
  6    12 ms     6 ms     7 ms  2400:1a00:dccc:1:72:9:128:67
  7     *        *        *     Request timed out.
  8    36 ms    39 ms    37 ms  2001:4860:1:1::126a
  9    37 ms    37 ms    39 ms  2404:6800:811f::1
 10    51 ms    38 ms    38 ms  2001:4860:0:1::54e4
 11    35 ms    38 ms    36 ms  2001:4860:0:1::54fb
 12    36 ms    38 ms    38 ms  del11s15-in-x0e.1e100.net [2404:6800:4002:81a::200e]

Trace complete.
```

4. **nslookup**
    - **Description**: Queries DNS to find the IP address of a domain or the domain associated with an IP address.
    - **Syntax**: nslookup [hostname/IP address]
        - Example: nslookup www.gmail.com

5. **netstat**
    - **Description**: Displays active network connections, listening ports, and network statistics. Useful for identifying open connections and troubleshooting.
    - **Syntax**: netstat

```
C:\Users\Dell>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:52942        DESKTOP-E6HN8H5:52943  ESTABLISHED
  TCP    127.0.0.1:52943        DESKTOP-E6HN8H5:52942  ESTABLISHED
  TCP    127.0.0.1:52944        DESKTOP-E6HN8H5:52945  ESTABLISHED
  TCP    127.0.0.1:52945        DESKTOP-E6HN8H5:52944  ESTABLISHED
  TCP    127.0.0.1:52948        DESKTOP-E6HN8H5:52949  ESTABLISHED
  TCP    127.0.0.1:52949        DESKTOP-E6HN8H5:52948  ESTABLISHED
  TCP    127.0.0.1:52950        DESKTOP-E6HN8H5:52951  ESTABLISHED
  TCP    127.0.0.1:52951        DESKTOP-E6HN8H5:52950  ESTABLISHED
  TCP    127.0.0.1:52952        DESKTOP-E6HN8H5:52953  ESTABLISHED
  TCP    127.0.0.1:52953        DESKTOP-E6HN8H5:52952  ESTABLISHED
  TCP    127.0.0.1:52967        DESKTOP-E6HN8H5:52968  ESTABLISHED
  TCP    127.0.0.1:52968        DESKTOP-E6HN8H5:52967  ESTABLISHED
  TCP    127.0.0.1:52969        DESKTOP-E6HN8H5:52970  ESTABLISHED
  TCP    127.0.0.1:52970        DESKTOP-E6HN8H5:52969  ESTABLISHED
  TCP    192.168.1.112:52975    ec2-44-192-202-19:4244  ESTABLISHED
  TCP    192.168.1.112:55158    20.212.88.117:https    ESTABLISHED
  TCP    192.168.1.112:59125    server-54-192-142-70:https  ESTABLISHED
  TCP    192.168.1.112:59177    ec2-18-210-105-110:https  ESTABLISHED
  TCP    192.168.1.112:59190    ec2-54-86-18-244:https  ESTABLISHED
  TCP    192.168.1.112:59196    ec2-54-86-18-244:https  ESTABLISHED
  TCP    192.168.1.112:59250    1drv:https             TIME_WAIT
  TCP    192.168.1.112:59254    20.189.173.3:https     ESTABLISHED
  TCP    192.168.1.112:59255    20.189.173.3:https     ESTABLISHED
  TCP    192.168.1.112:59259    1drv:https             TIME_WAIT
  TCP    192.168.1.112:59267    ec2-3-211-112-147:https  ESTABLISHED
  TCP    192.168.1.112:59271    210.148.85.47:http     TIME_WAIT
  TCP    192.168.1.112:59272    1drv:https             TIME_WAIT
  TCP    192.168.1.112:59273    183:https              ESTABLISHED
  TCP    192.168.1.112:59274    1drv:https             TIME_WAIT
  TCP    192.168.1.112:59275    1drv:https             TIME_WAIT
  TCP    192.168.1.112:59276    ec2-54-243-121-108:https  ESTABLISHED
  TCP    192.168.1.112:59277    1drv:https             TIME_WAIT
  TCP    192.168.1.112:59283    103.211.150.137:https  ESTABLISHED
  TCP    192.168.1.112:59284    a96-17-150-209:https   ESTABLISHED
  TCP    192.168.1.112:59285    a96-17-150-209:https   ESTABLISHED
  TCP    192.168.1.112:59286    a-0003:https           TIME_WAIT
  TCP    192.168.1.112:59289    1drv:https             TIME_WAIT
  TCP    192.168.1.112:59292    1drv:https             TIME_WAIT
  TCP    192.168.1.112:59293    ec2-52-207-103-181:https  ESTABLISHED
  TCP    192.168.1.112:59294    1drv:https             TIME_WAIT
```

```
TCP    192.168.1.112:59292    1drv:https             TIME_WAIT
TCP    192.168.1.112:59293    ec2-52-207-103-181:https  ESTABLISHED
TCP    192.168.1.112:59294    1drv:https             TIME_WAIT
TCP    192.168.1.112:59302    210.148.85.47:http     ESTABLISHED
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:49468  [2603:1040:a06:6::2]:https  ESTABLISHED
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:52267  [2603:1040:a06:6::2]:https  ESTABLISHED
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:55434  sg-in-f188:5228        ESTABLISHED
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:57508  [2620:1ec:29:1::37]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:57927  [2400:1a00:4:c150::67d3:9699]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:57928  [2400:1a00:4:c150::67d3:96aa]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:57929  [2400:1a00:4:c150::67d3:96aa]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:57930  [2400:1a00:4:c150::67d3:96aa]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:57931  [2400:1a00:4:c150::67d3:96aa]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:57932  [2400:1a00:4:c150::67d3:96aa]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59034  whatsapp-cdn6-shv-01-del2:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59035  whatsapp-cdn6-shv-02-del2:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59036  [2400:1a00:4:13a:face:b00c:3333:7020]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59037  whatsapp-cdn6-shv-01-bom2:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59038  [2400:1a00:4:139:face:b00c:3333:7020]:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59039  whatsapp-cdn6-shv-02-del1:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59040  whatsapp-cdn6-shv-03-bom2:https  CLOSE_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59150  [2606:4700:4400::6812:202f]:https  ESTABLISHED
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59256  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59257  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59258  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59260  [2603:1046:1406::5]:https  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59261  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59262  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59263  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59264  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59265  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59266  [2603:1046:1406::5]:https  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59268  [2603:1046:1406::5]:https  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59278  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59279  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59290  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59291  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59295  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59296  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59297  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59298  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59299  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59304  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59305  vip6-safenet-kmd01:domain  TIME_WAIT
TCP    [2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de]:59306  vip6-safenet-kmd01:domain  TIME_WAIT
```

6. **arp**
   - **Description**: Displays or modifies the ARP (Address Resolution Protocol) table, which maps IP addresses to physical MAC addresses.
   - **Syntax**: arp -a (to view the ARP table)

```
C:\Users\Dell>arp -a

Interface: 192.168.1.112 --- 0x7
  Internet Address      Physical Address      Type
  192.168.1.254         04-75-f9-a1-0c-60     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.56.1 --- 0x14
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

7. **route**
    - **Description**: Displays and manipulates the IP routing table, which determines the path that network traffic takes.
    - **Syntax**: route print (to display the routing table)

```
C:\Users\Dell>rout print
'rout' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Dell>route print
===========================================================================
Interface List
  9...60 18 95 1d e3 61 ......Realtek PCIe GbE Family Controller
 20...0a 00 27 00 00 14 ......VirtualBox Host-Only Ethernet Adapter
 10...a4 42 3b 5c 31 96 ......Microsoft Wi-Fi Direct Virtual Adapter
 17...a6 42 3b 5c 31 95 ......Microsoft Wi-Fi Direct Virtual Adapter #2
  7...a4 42 3b 5c 31 95 ......Intel(R) Wireless-AC 9462
 15...a4 42 3b 5c 31 99 ......Bluetooth Device (Personal Area Network)
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    192.168.1.254    192.168.1.112     35
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
      192.168.1.0    255.255.255.0         On-link     192.168.1.112    291
    192.168.1.112  255.255.255.255         On-link     192.168.1.112    291
    192.168.1.255  255.255.255.255         On-link     192.168.1.112    291
     192.168.56.0    255.255.255.0         On-link      192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link      192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link     192.168.1.112    291
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link      192.168.56.1    281
  255.255.255.255  255.255.255.255         On-link     192.168.1.112    291
===========================================================================
Persistent Routes:
  None
```

```
IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  7   4131 ::/0                    fe80::1
  1    331 ::1/128                 On-link
  7   4131 2400:1a00:b040:f08b::/64 On-link
  7     51 2400:1a00:b040:f08b::/64 fe80::1
  7    291 2400:1a00:b040:f08b::2/128
                                   On-link
  7    291 2400:1a00:b040:f08b:7a1d:91d0:9304:90de/128
                                   On-link
  7    291 2400:1a00:b040:f08b:ad18:3a25:ac1c:f7de/128
                                   On-link
 20    281 fe80::/64                On-link
  7    291 fe80::/64                On-link
  7    291 fe80::10a1:1c54:6fd0:d05c/128
                                   On-link
 20    281 fe80::2ee6:2f29:975c:cb/128
                                   On-link
  1    331 ff00::/8                 On-link
 20    281 ff00::/8                 On-link
  7    291 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```

8. **hostname**
    - **Description**: Displays the current hostname of the computer or allows the hostname to be set.
    - **Syntax**: hostname

```
C:\Users\Dell>hostname
DESKTOP-E6HN8H5
```

9. **telnet**
    - **Description**: Establishes a Telnet connection to a remote machine. Telnet is a protocol for remote command-line access.
    - **Syntax**: telnet [hostname/IP address] [port]
        - Example: telnet 192.168.1.1 80

```
C:\Users\Dell>telnet 192.168.1.1 80
Connecting To 192.168.1.1...Could not open connection to the host, on port 80: Connect failed
```

10. **curl**

- **Description**: Transfers data from or to a server using various protocols. It's often used for testing APIs and web servers.
- **Syntax**: curl [URL]
    - Example: curl http://www.example.com

```
C:\Users\Dell>curl http://www.example.com
<!doctype html>
<html>
<head>
    <title>Example Domain</title>

    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style type="text/css">
    body {
        background-color: #f0f0f2;
        margin: 0;
        padding: 0;
        font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;

    }
    div {
        width: 600px;
        margin: 5em auto;
        padding: 2em;
        background-color: #fdfdff;
        border-radius: 0.5em;
        box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
    }
    a:link, a:visited {
        color: #38488f;
        text-decoration: none;
    }
    @media (max-width: 700px) {
        div {
            margin: 0 auto;
            width: auto;
        }
    }
    </style>
</head>

<body>
<div>
    <h1>Example Domain</h1>
    <p>This domain is for use in illustrative examples in documents. You may use this
    domain in literature without prior coordination or asking for permission.</p>
    <p><a href="https://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>
```

Conclusion:

By understanding and using these basic networking commands, we can perform crucial tasks such as diagnosing connectivity issues, analyzing network performance, and resolving DNS queries. These tools are fundamental in managing and troubleshooting both simple and complex network environments.

LAB: 3

AIM: To study packet Tracer and basic LAN setup.

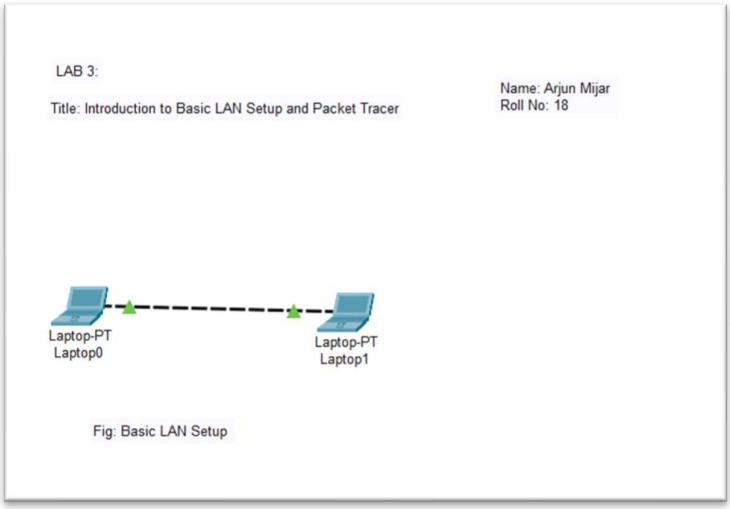
Theory:

Packet Tracer is a simulation tool developed by Cisco, widely used to visualize and simulate networking environments. It allows users to create network topologies, configure devices, and test networking protocols in a virtual setting.

A Local Area Network (LAN) is a network infrastructure that enables devices within a limited geographical area, such as an office or home, to connect and communicate with each other. The basic LAN setup is fundamental in networking, providing the foundational understanding needed for more complex network configurations. A basic LAN can be setup using hubs and switches in Cisco Packet Tracer, a network simulation tool that allows users to create, configure, and analyze network topologies.

Hub: A hub is a basic networking device that operates at the physical layer (Layer 1) of the OSI model. It acts as a central connection point, broadcasting data to all connected devices. This leads to inefficient data transmission and potential collisions, as it cannot filter traffic or identify specific destinations. Hubs are largely outdated in modern networks.

Switch: A switch operates at the data link layer (Layer 2) and efficiently forwards data to the specific device based on its MAC address. Unlike hubs, switches reduce traffic and collisions by directing packets only to the intended destination. They also support full-duplex communication, improving network speed and performance.



Working Procedure:

Basic LAN with Hub:

1. **Setup**: Connect multiple devices (such as PCs) to a central hub using Ethernet cables. Each device is plugged into one of the hub's ports.
2. **Data Transmission**: When a device sends data, the hub receives this data and broadcasts it to all other connected devices.
3. **Data Handling**: The hub does not differentiate between devices; it simply relays incoming data to every connected port. This means all devices on the network receive the data, regardless of whether they are the intended recipient or not.

4. **Network Traffic**: Due to the broadcast nature of the hub, network traffic is high and the likelihood of data collisions increases, especially when multiple devices attempt to send data simultaneously.
5. **IP Configuration**: Ensure each device is configured with a unique IP address within the same network range. This allows the devices to recognize each other and communicate through the hub.
6. **Testing Connectivity**: Use network testing commands such as "ping" to verify device connectivity. Successful pings indicate that devices can communicate through the hub.

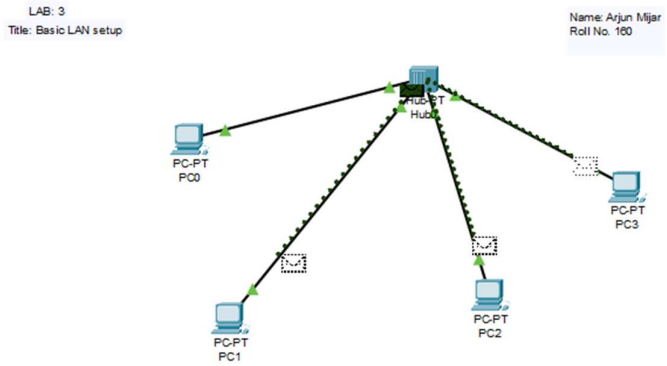


Fig: Basic LAN setup with HUB

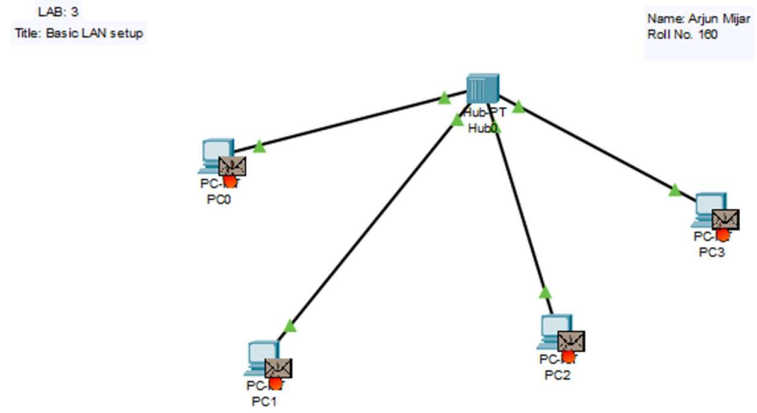


Fig: Basic LAN setup with HUB

Basic LAN with Switch:

1. **Setup**: Connect multiple devices (such as PCs) to a central switch using Ethernet cables. Each device is plugged into one of the switch's ports.
2. **Data Transmission**: When a device sends data, the switch examines the destination device's MAC address and forwards the data directly to that specific device.

3. **Data Handling**: The switch maintains a MAC address table that maps each device's MAC address to its corresponding port. This allows the switch to send data only to the intended recipient, rather than broadcasting it to all devices.
4. **Network Traffic**: Because the switch only forwards data to the intended recipient, network traffic is reduced, and the likelihood of data collisions is minimized.
5. **IP Configuration**: Assign a unique IP address to each device within the same subnet. This ensures devices can communicate effectively within the network.
6. **Testing Connectivity**: Use network testing commands such as "ping" to verify device connectivity. Successful pings confirm that devices can communicate through the switch without issues.

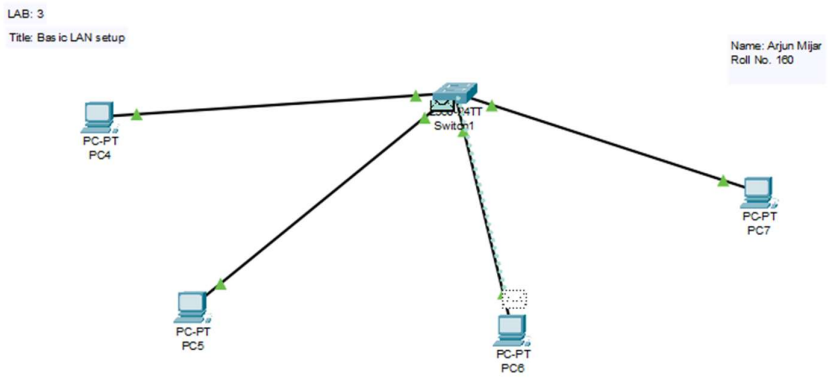


Fig: Basic LAN setup with SWITCH

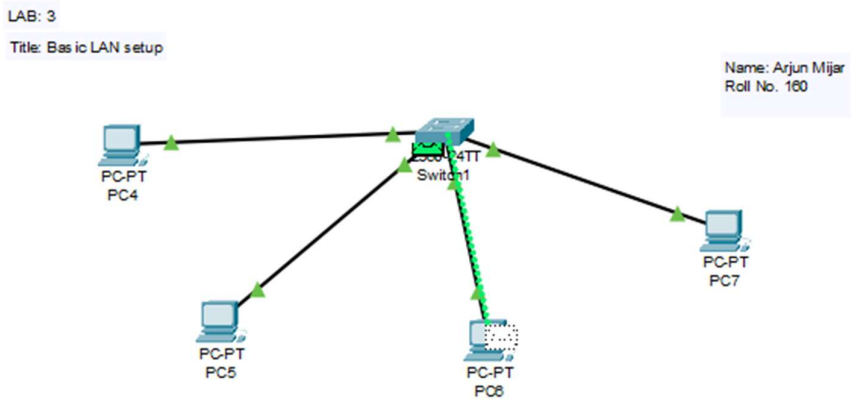


Fig: Basic LAN setup with SWITCH

Conclusion:
This lab showed that hub-based networks are inefficient due to increased collisions, while switch-based networks improve performance by efficiently directing data. Using Cisco Packet Tracer, we gained practical insight into LAN setups and the importance of selecting the right device for network efficiency.

LAB: 4

AIM: To study DHCP, DNS, and Server Setup

Theory:

DHCP (Dynamic Host Configuration Protocol) is a network protocol that automatically assigns IP addresses and other network configuration parameters to devices on a network, allowing them to communicate efficiently. Without DHCP, each device must be manually configured with an IP address, which is time-consuming and prone to errors.

- **Key Functions of DHCP**:
  - Assigns **IP addresses** dynamically from a pool of available addresses.
  - Provides essential information like **subnet mask**, **default gateway**, and **DNS server**.
  - Manages **IP address leases**, ensuring that IP addresses are reassigned or renewed as necessary.

How DHCP Works (4 Steps):

1. **Discover**: A device (client) broadcasts a DHCP request to find a DHCP server.
2. **Offer**: The DHCP server responds with an offer of an IP address.
3. **Request**: The client selects an IP from the offered options and requests it from the server.
4. **Acknowledge**: The DHCP server acknowledges the request and assigns the IP the client uses for a predefined lease time.

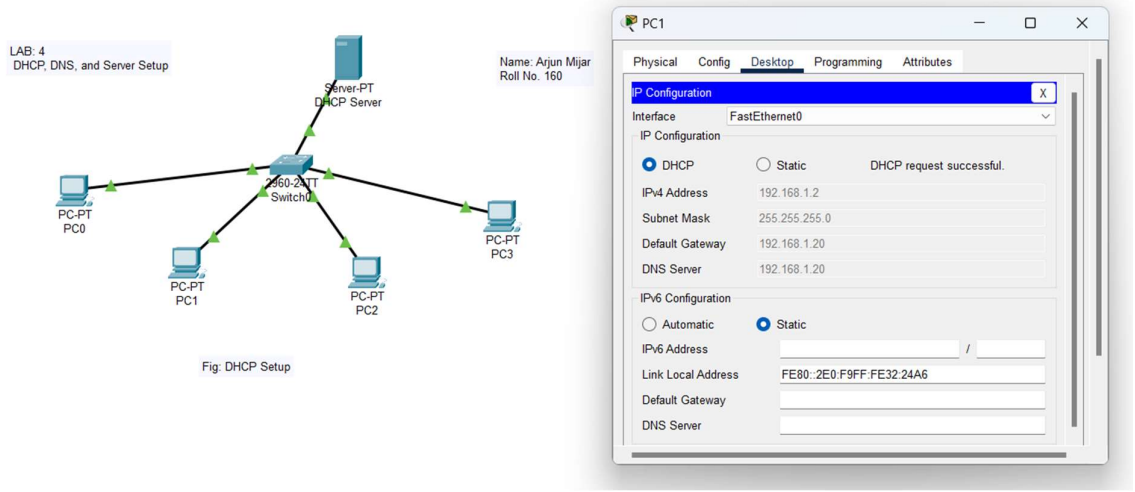


Fig: DHCP Setup

DNS (Domain Name System) is a system that translates human-readable domain names (like www.example.com) into machine-readable IP addresses (like 192.168.1.1). This is essential because while humans can easily remember names, computers require IP addresses to locate and communicate with each other over the internet or a network.

- **Key Roles of DNS**:
    - **Name Resolution**: It converts domain names into IP addresses.
    - **Simplifies Network Access**: Instead of remembering IP addresses, users can type in easy-to-remember domain names.
    - **Hierarchy**: DNS uses a hierarchical naming structure, allowing efficient and scalable management of names across the internet.

How DNS Works

1. **User Query**: When a user types a domain name (e.g., www.example.com), the computer requests the corresponding IP address.
2. **Local Cache**: The computer first checks its local DNS cache. If the IP isn't found, the query goes to the ISP's DNS resolver.
3. **Root DNS Server**: If the resolver doesn't have the answer, it queries a **Root DNS Server**, which points it to the relevant **TLD server** (e.g., .com).
4. **TLD DNS Server**: The **Top-Level Domain (TLD)** server directs the query to the **authoritative DNS server** for the specific domain.
5. **Authoritative DNS Server**: This server provides the IP address of the domain.
6. **Caching and Response**: The resolver caches the IP address and sends it back to the user's browser, which then connects to the web server.

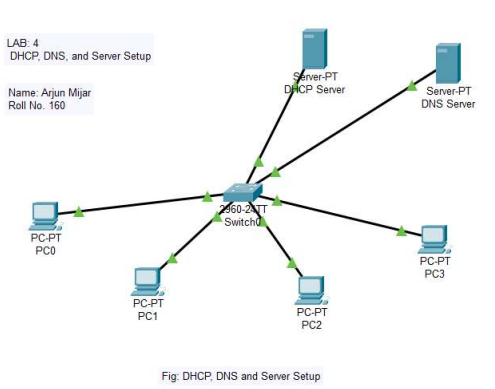
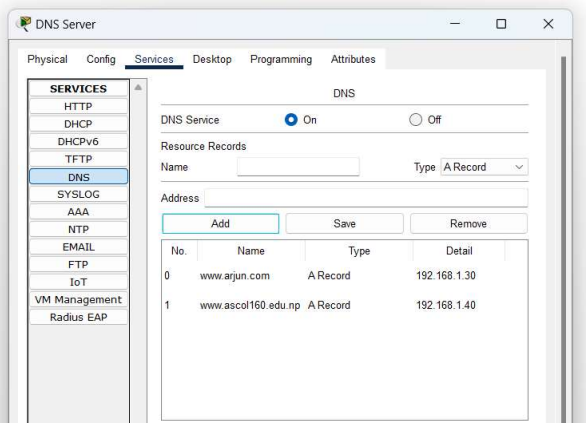


Fig: DHCP, DNS and Server Setup



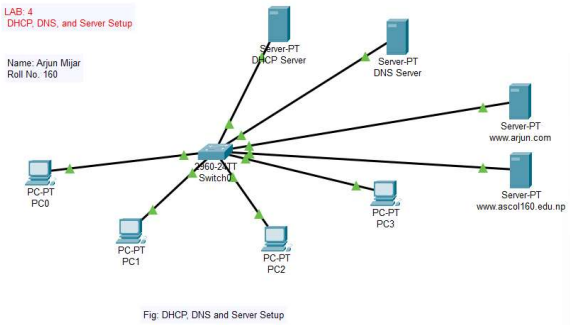
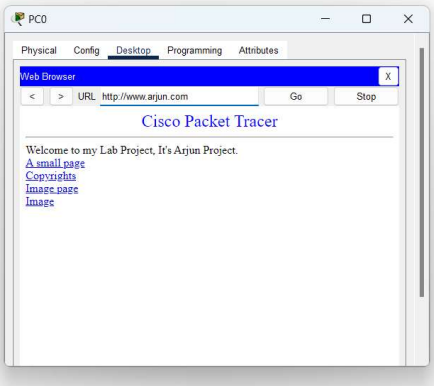


Fig: DHCP, DNS and Server Setup



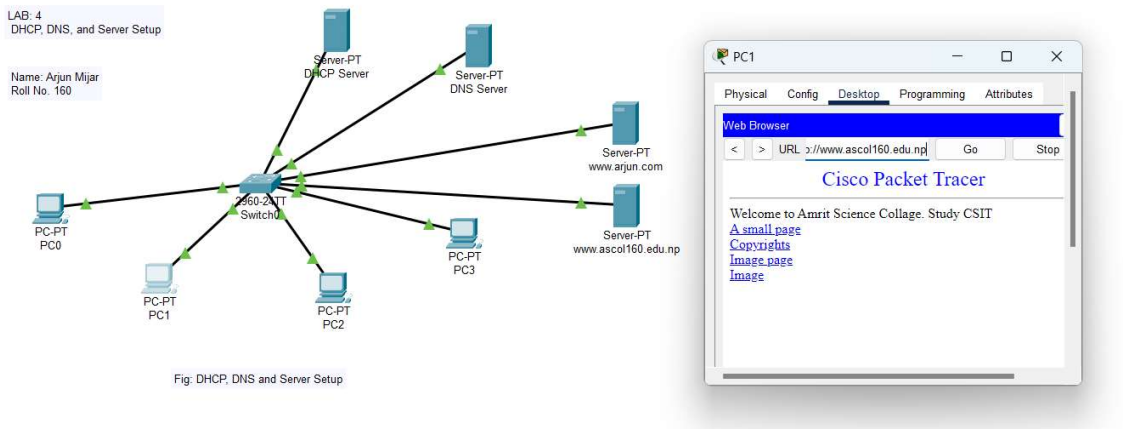


Fig: DHCP, DNS and Server Setup

Conclusion:

This lab demonstrated how IP addresses can be assigned efficiently through DHCP, providing practical insight into configuring network services like DNS and setting up servers effectively. Using Cisco Packet Tracer, we explored the automated assignment of IP addresses statically and learned how DNS resolves domain names to IP addresses. Additionally, the exercise highlighted the importance of configuring servers with static IP addresses for consistent accessibility, ensuring that networks function smoothly and without conflicts. Overall, this hands-on experience deepened our understanding of the dynamic interaction between DHCP, DNS, and server configuration in modern networks.

LAB: 5

AIM: Basic Router Configuration with DHCP

Introduction:

Dynamic Host Configuration Protocol (DHCP) automates the assignment of IP addresses to devices on a network, ensuring ease of network management and reducing the chance of address conflicts. In a basic router configuration with DHCP, the router functions as a DHCP server, dynamically allocating IP addresses to devices within a specified range. This lab focuses on configuring a router to use DHCP to automatically assign IP addresses to connected devices, simplifying the setup of local area networks (LANs).

Theory:

DHCP is a protocol used to automatically assign IP addresses, subnet masks, default gateways, and DNS server addresses to client devices (such as PCs, laptops, and printers) on a network. A router configured with DHCP can dynamically distribute these parameters to devices when they join the network. The DHCP process involves four key stages:

1. **Discover**: A client device sends a DHCP Discover broadcast message to identify available DHCP servers.
2. **Offer**: The DHCP server responds with a DHCP Offer, providing an IP address and configuration details.
3. **Request**: The client sends a DHCP Request message to accept the offered IP address.
4. **Acknowledge**: The DHCP server sends an Acknowledgment (ACK), confirming the assignment.

By automatically assigning IP addresses, DHCP reduces administrative overhead and ensures that each device receives a unique IP address without manual configuration.

Working Procedure:

Step 1: Setting Up the Network in Cisco Packet Tracer

1. **Add Devices**:
   - Place a router, a switch, and several PCs on the workspace.
   - Connect the PCs to the switch using Ethernet cables.
   - Connect the router to the switch using an Ethernet cable as well.

Step 2: Configure Router Interfaces

1. **Access the Router**:
   - Click on the router and open the **CLI** tab.
   - Enter **privileged EXEC mode** by typing enable.
2. **Configure the Router's GigabitEthernet Interface**:
   - Enter global configuration mode by typing configure terminal.
   - Assign an IP address to the router's interface connected to the LAN:

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

Step 3: Configure DHCP on the Router

1. **Enable DHCP**:
   - Configure the DHCP pool that will assign IP addresses to the devices:

```
Router(config)# ip dhcp pool LAN
Router(dhcp-config)# network 192.168.1.0 255.255.255.0
Router(dhcp-config)# default-router 192.168.1.1
Router(dhcp-config)# dns-server 8.8.8.8
```

2. **Exclude Specific IP Addresses**:
   - Exclude IP addresses from the DHCP pool that will not be assigned to clients (e.g., for the router or network servers):

```
Router(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.10
```

Step 4: Verify DHCP Configuration

1. **Test on the Client PC**:
   - On one of the PCs, go to the **Desktop tab** and open the **IP Configuration**.
   - Set the **IP Configuration** to **DHCP**.
   - The PC should automatically receive an IP address from the router's DHCP pool.
2. **Check Router DHCP Assignments**:
   - To view the IP addresses assigned by the router, use the following command:

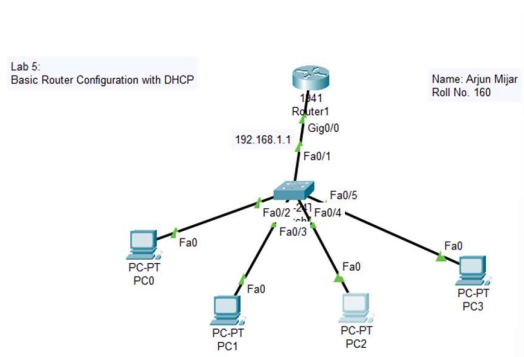
```
Router# show ip dhcp binding
```

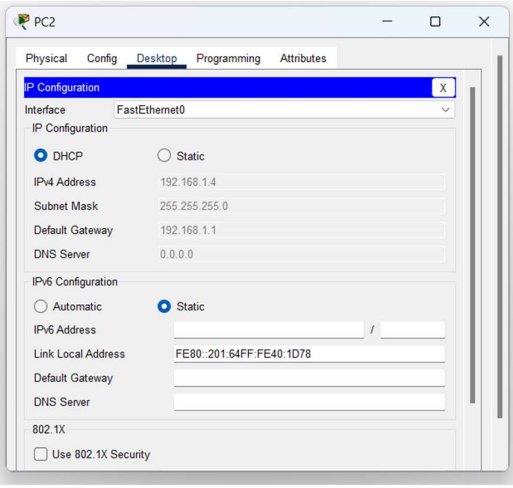
Step 5: Verify Connectivity

1. **Test Network Connectivity**:
   - From the PC, open the **Command Prompt** and test network connectivity by using the ping command:

```
ping 192.168.1.1
```

   - This should successfully ping the router's IP address, confirming that DHCP is working and the devices are connected.






```
arjun#show running-config
Building configuration...

Current configuration : 775 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname arjun
!
!
!
enable secret 5 $1$mERr$/Q/mbs3O9oHsKR7rNG4e81
!
!
!
ip dhcp pool lan1
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.1
!
```

```
interface GigabitEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
!
ip flow-export version 9
!
!
!
!
!
!
!
line con 0
 password Arjun12
 login
!
line aux 0
!
line vty 0 4
 login
!
!
!
end
```

Conclusion:

By configuring a router to act as a DHCP server, we successfully automated the IP addressing process for the devices in the network. DHCP significantly simplifies network management, reducing manual configuration and the likelihood of IP conflicts. The lab demonstrated how the router dynamically assigns IP addresses to connected devices, ensuring efficient and effective network operation.

LAB: 6

AIM: Implementation of Static Routing

Introduction:

Static routing is a fundamental aspect of network design and management, where network administrators manually set up routes within the routing table of routers. Unlike dynamic routing, static routing doesn't rely on algorithms or protocols to update paths; instead, it provides precise control over data flow within the network. This lab focuses on understanding the configuration and operation of static routing in a controlled environment. By implementing static routes, network administrators can optimize network performance, enhance security, and ensure reliable data delivery in networks with predictable traffic patterns.

Static routing is often used in small or specialized networks where the network paths are consistent and predictable. It is also commonly used in edge networks or in scenarios where security is a priority, as it eliminates the possibility of routing updates being intercepted or manipulated.

Theory:

Static routing is a type of network routing technique where routes are manually added to the routing table. Unlike dynamic routing, static routing requires no routing protocol, which makes it more secure but less scalable. It is useful in small networks where the route doesn't change frequently. Static routing involves the manual configuration of the routing table, which contains the routes that the router uses to forward packets to their destination. Each entry in the routing table typically includes the destination network, subnet mask, and the next hop (the IP address of the next router in the path). When a packet arrives at a router, the router examines the destination IP address and forwards the packet based on the routing table entry that best matches the destination.

The key characteristics of static routing include:

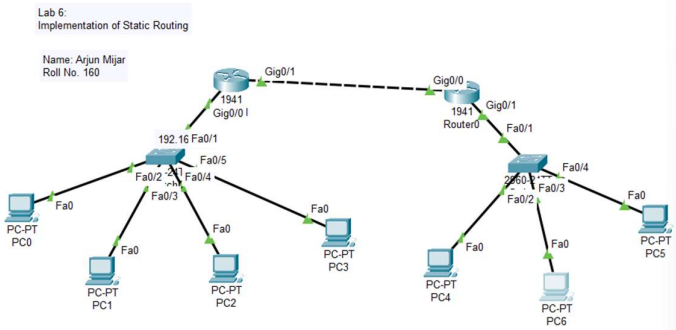
1. **Manual Configuration:** Network administrators manually add and remove routes, which means changes in the network topology require manual updates to the routing tables.
2. **No Routing Overhead:** Since there is no need for routing updates or protocol overhead, static routing reduces the processing burden on the router, making it efficient for small networks.
3. **Predictability and Security:** Static routes provide a predictable path for network traffic and can be more secure than dynamic routes because they are not susceptible to route poisoning or spoofing attacks.
4. **Scalability:** Static routing is not suitable for large or complex networks, as maintaining the routing table becomes cumbersome and error-prone with an increasing number of routes.
5. **Failover:** Static routes do not automatically adjust to changes in the network, so if a link fails, the static route will continue to attempt to use the failed path unless an alternate route is manually configured.

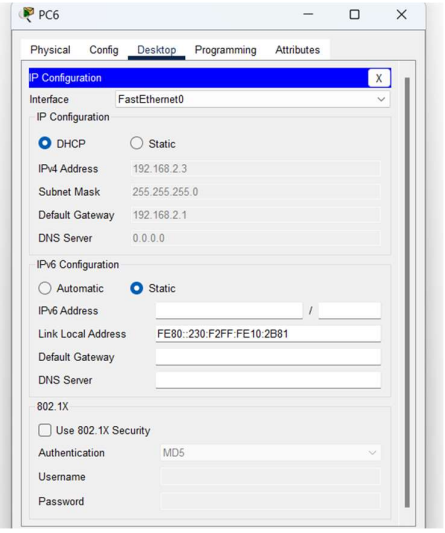
Working Procedure:

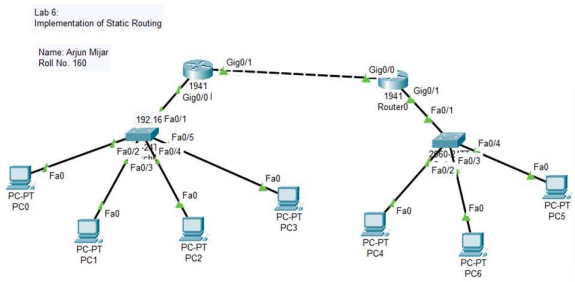
- Create a network topology: Design a network with at least three routers and several networks connected to each router.
- Assign IP addresses: Assign appropriate IP addresses to each interface on the routers and end devices.
- Configure static routes: Manually configure static routes on each router to ensure all networks can communicate with each other.
- Verify the configuration: Use the `ping` and `tracert` commands to ensure the network communication is correctly configured.

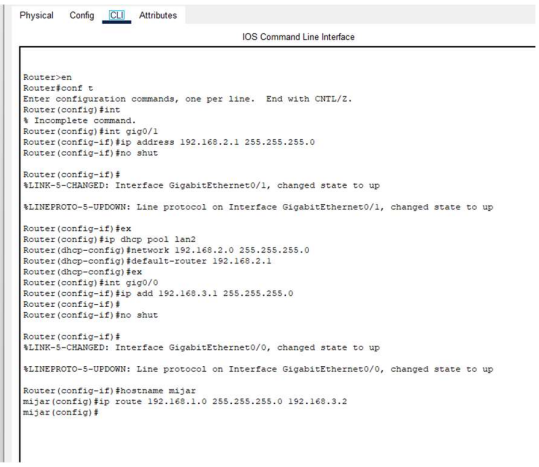
Setup Steps in Cisco Packet Tracer:

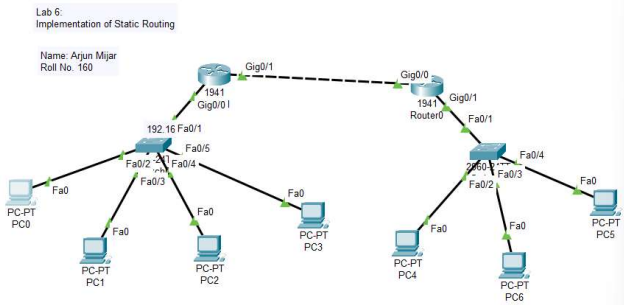
1. Create the Network Topology:
   - Add three routers, two switches, and four PCs to the workspace.
   - Connect the routers using serial connections and the PCs using Ethernet cables to the switches.
2. Assign IP Addresses:
   - Configure IP addresses for each PC and router interface. Ensure all devices are on different networks.
3. Configure Static Routing:
   - On each router, configure static routes using the following command:
     Router(config)# ip route [destination network] [subnet mask] [next hop IP address]
   - For example, to route traffic from Router1 to Router2's network, use:
     Router1(config)# ip route 192.168.2.0 255.255.255.0 192.168.1.2
4. Verify Connectivity:
   - Test the network by pinging from one PC to another across the routers. All pings should be successful if the static routes are correctly configured.
   - Use the `tracert` command to observe the path taken by packets across the network.

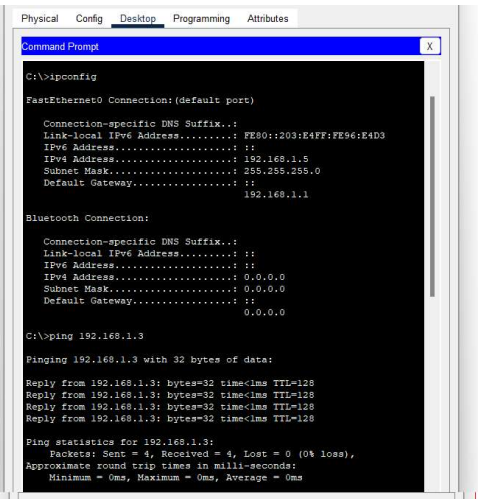


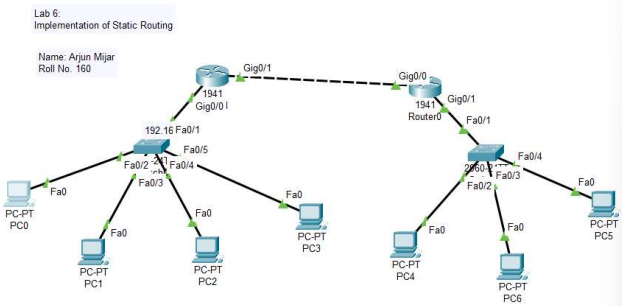


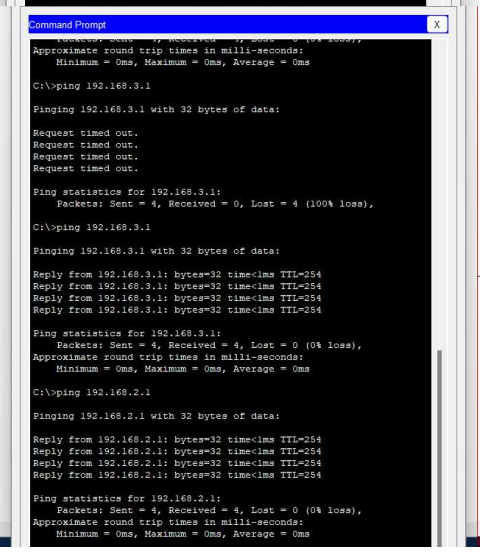

Lab 6:
Implementation of Static Routing
Name: Arjun Mijar
Roll No. 160
Gig0/1
1941
Gig0/0
Gig0/1
1941
Router0
Fa0/1
192.16
Fa0/1
Fa0/5
Fa0/2
Fa0/4
Fa0/3
Fa0/4
Fa0/3
Fa0/2
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC6


Physical
Config
CLI
Attributes
IOS Command Line Interface
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int
% Incomplete command.
Router(config)#int gig0/1
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
Router(config-if)#ex
Router(config)#ip dhcp pool lan2
Router(dhcp-config)#network 192.168.2.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.2.1
Router(dhcp-config)#ex
Router(config)#int gig0/0
Router(config-if)#ip add 192.168.3.1 255.255.255.0
Router(config-if)#
Router(config-if)#no shut
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
Router(config-if)#hostname mijar
mijar(config)#ip route 192.168.1.0 255.255.255.0 192.168.3.2
mijar(config)#


Lab 6:
Implementation of Static Routing
Name: Arjun Mijar
Roll No. 160
Gig0/1
1941
Gig0/0
Gig0/1
1941
Router0
Fa0/1
192.16
Fa0/1
Fa0/5
Fa0/2
Fa0/4
Fa0/3
Fa0/4
Fa0/3
Fa0/2
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC6


Physical
Config
Desktop
Programming
Attributes
Command Prompt
C:\>ipconfig
FastEthernet0 Connection:(default port)
Connection-specific DNS Suffix..:
Link-local IPv6 Address.....: FE80::203:E4FF:FE96:E4D3
IPv6 Address.....: ::
IPv4 Address.....: 192.168.1.5
Subnet Mask.....: 255.255.255.0
Default Gateway.....: ::
192.168.1.1
Bluetooth Connection:
Connection-specific DNS Suffix..:
Link-local IPv6 Address.....: ::
IPv6 Address.....: ::
IPv4 Address.....: 0.0.0.0
Subnet Mask.....: 0.0.0.0
Default Gateway.....: ::
0.0.0.0
C:\>ping 192.168.1.3
Pinging 192.168.1.3 with 32 bytes of data:
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms


Lab 6:
Implementation of Static Routing
Name: Arjun Mijar
Roll No. 160
Gig0/1
1941
Gig0/0
Gig0/1
1941
Router0
Fa0/1
192.16
Fa0/1
Fa0/5
Fa0/2
Fa0/4
Fa0/3
Fa0/4
Fa0/3
Fa0/2
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC-PT
PC6


Command Prompt
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.3.1
Pinging 192.168.3.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.3.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 192.168.3.1
Pinging 192.168.3.1 with 32 bytes of data:
Reply from 192.168.3.1: bytes=32 time<1ms TTL=254
Reply from 192.168.3.1: bytes=32 time<1ms TTL=254
Reply from 192.168.3.1: bytes=32 time<1ms TTL=254
Reply from 192.168.3.1: bytes=32 time<1ms TTL=254
Ping statistics for 192.168.3.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.2.1
Pinging 192.168.2.1 with 32 bytes of data:
Reply from 192.168.2.1: bytes=32 time<1ms TTL=254
Reply from 192.168.2.1: bytes=32 time<1ms TTL=254
Reply from 192.168.2.1: bytes=32 time<1ms TTL=254
Reply from 192.168.2.1: bytes=32 time<1ms TTL=254
Ping statistics for 192.168.2.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

Conclusion:

The implementation of static routing allows for precise control over the paths that data packets take through a network. Unlike dynamic routing, static routes are manually configured, which gives administrators greater control but also increases the need for careful planning and management, especially in larger networks. In this lab, we successfully implemented static routes between routers, ensuring that each router knew the next hop for forwarding packets to different network segments. This approach works well in smaller, stable networks where the topology does not frequently change. However, for more complex and dynamic environments, static routing can become cumbersome, and dynamic routing protocols may be preferred. Through this exercise, we gained a deeper understanding of how routing tables function and the importance of proper route configuration to ensure efficient and correct packet delivery across the network.

LAB: 7

AIM: Dynamic Routing Implementation using RIP

Introduction:

Dynamic routing automates the process of route calculation and distribution within a network, enabling routers to adapt to changes in the network topology. The Routing Information Protocol (RIP) is one of the earliest and most straightforward dynamic routing protocols, ideal for smaller, less complex networks. In this lab, you will explore the implementation of RIP, learning how routers dynamically share information and adjust routes based on network changes.

Dynamic routing protocols like RIP are crucial in modern networks, where the topology can frequently change due to various factors, such as link failures, changes in bandwidth, or the addition of new routers and networks. By using RIP, routers automatically share information about network topology with their neighbors, allowing them to maintain an up-to-date view of the network.

Theory:

RIP is a distance-vector routing protocol that uses hop count as its primary metric to determine the best path to a destination network. The "hop count" refers to the number of routers a packet must pass through to reach its destination. The protocol has a maximum hop count of 15, meaning any network that is 16 or more hops away is considered unreachable.

Key characteristics of RIP include:

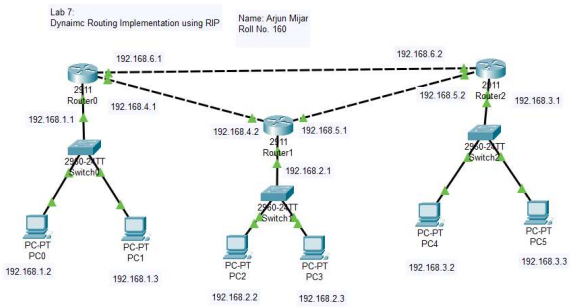
1. **Distance Vector Algorithm:** RIP uses the Bellman-Ford algorithm to determine the shortest path to a destination based on hop count. Each router periodically sends out its entire routing table to its immediate neighbors, who then update their tables based on this information.
2. **Hop Count Limit:** The hop count limit of 15 makes RIP unsuitable for large networks, but this limitation also prevents routing loops in large networks.
3. **Periodic Updates:** RIP routers send out routing updates every 30 seconds, which can cause slow convergence in larger networks but is sufficient for small to medium-sized networks.
4. **Simple Configuration:** RIP is easy to configure, making it a good choice for smaller networks or networks where ease of setup is a priority over performance.
5. **Broadcast Updates:** RIP uses broadcast rather than multicast for its updates, which can result in unnecessary traffic on network segments not running RIP.
6. **Routing Loops and Split Horizon:** RIP implements techniques like split horizon and hold-down timers to prevent routing loops and to ensure stability in the network.

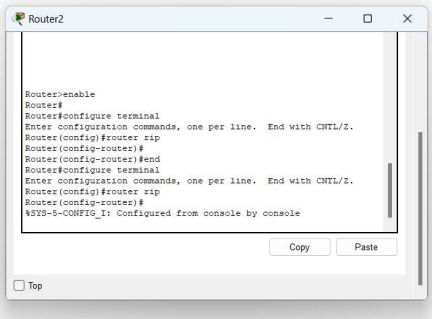
By implementing RIP, network administrators can ensure that the network remains flexible and adaptive to changes, reducing the need for manual intervention when the network topology changes.

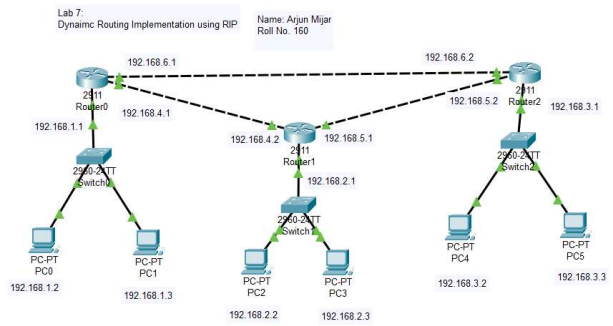
Setup Steps in Cisco Packet Tracer:

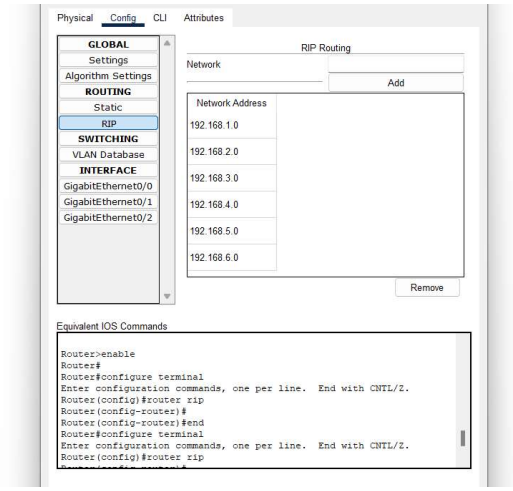
1. Create the Network Topology:
   - Add three routers, three switches, and several PCs to the workspace.
   - Connect routers using serial connections, and connect the PCs to the switches using Ethernet cables.
2. Assign IP Addresses:
   - Configure IP addresses for all router interfaces and PCs.
3. Enable RIP:
   - On each router, enter the following commands to enable RIP:
     Router(config)# router rip
     Router(config-router)# network [network ID]
   - For example, if Router1 is connected to network 192.168.1.0/24, use:
     Router1(config-router)# network 192.168.1.0
   - Repeat the process for all connected networks on each router.
4. Verify Connectivity:
   - Use the `show ip route` command to view the routing table and ensure RIP has populated the routes.
   - Test connectivity by pinging between PCs on different networks. All pings should be successful.
   - Use `tracert` to see the path taken by packets

```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#
%SYS-5-CONFIG_I: Configured from console by console
```









Conclusion:

RIP simplifies the management of routing in a dynamic network by automatically adjusting routes based on network changes. Its simplicity and ease of configuration make it ideal for small to medium-sized networks. Although RIP has limitations, such as a maximum hop count of 15 and slower convergence times compared to more advanced protocols like OSPF, it provides a great foundation for understanding the principles of dynamic routing. By implementing RIP, network administrators can reduce manual intervention and allow the network to adapt to topology changes automatically, making it a useful tool in less complex environments.

LAB: 8

AIM: Dynamic Routing Implementation using OSPF

Theory:

Open Shortest Path First (OSPF) is a dynamic routing protocol used for routing packets within a large or complex network. It is a link-state routing protocol that operates at Layer 3 of the OSI model (the network layer). OSPF dynamically learns and distributes routing information among routers, enabling them to calculate the shortest path to each destination using a metric called **cost**, which is usually based on link bandwidth.

Key features of OSPF include:

1. **Link-State Algorithm**: OSPF uses Dijkstra's shortest path first (SPF) algorithm to calculate the best path to each network.
2. **Hierarchical Structure**: OSPF divides a network into areas to optimize routing and reduce overhead. All areas must connect to a central **backbone area** (Area 0).
3. **Faster Convergence**: OSPF responds more quickly to network changes compared to distance-vector protocols like RIP.
4. **Classless**: OSPF supports Classless Inter-Domain Routing (CIDR), which allows for efficient use of IP address space by supporting variable-length subnet masks (VLSM).
5. **Multicast Updates**: OSPF sends updates using multicast addresses rather than broadcast, reducing unnecessary traffic.

OSPF is widely used in large enterprise networks because of its scalability, efficiency, and adaptability in dynamic network environments.

Working Procedure:
Step 1: Create the Network Topology

1. **Add Devices**:
   - Use a network simulator like Cisco Packet Tracer.
   - Add multiple routers, switches, and PCs to the workspace.
   - Connect the routers using serial or Ethernet links, and connect the PCs to the switches using Ethernet cables.
2. **Assign IP Addresses**:
   - Assign IP addresses to the interfaces of routers and PCs. Ensure that all devices are part of different network segments.
   - Example configuration:
     - **Router1** (connected to 192.168.1.0/24)
     - **Router2** (connected to 192.168.2.0/24)

Step 2: OSPF Configuration on Routers

1. **Access the Router**:
   - Click on a router and open the **CLI**. Enter **privileged EXEC mode**:

     enable

2. **Configure OSPF Routing**:
    - Enter global configuration mode and configure OSPF. Use **process ID 1** (or any valid ID):

      Router(config)# router ospf 1

3. **Assign Networks to OSPF**:
    - Assign each connected network to the OSPF process using the **network** command. Define the network and the area it belongs to. All routers must share the same area to exchange routing information.
        - For Router1 (connected to network 192.168.1.0/24):

          Router(config-router)# network 192.168.1.0 0.0.0.255 area 0

        - For Router2 (connected to network 192.168.2.0/24):

          Router(config-router)# network 192.168.2.0 0.0.0.255 area 0

        - The **0.0.0.255** is the **wildcard mask**, which is the inverse of a subnet mask.
4. **Configure OSPF on All Routers**:
    - Repeat this process for all routers in the network, assigning the appropriate networks to OSPF and using the same area ID (e.g., **area 0**).

Step 3: Verify OSPF Configuration

1. **Verify OSPF Neighbors**:
    - To check if OSPF has established neighbor relationships (adjacency) with other routers:

      Router# show ip ospf neighbor

2. **View OSPF Routing Table**:
    - Check the routing table to ensure that OSPF has successfully shared routes between routers:

      Router# show ip route

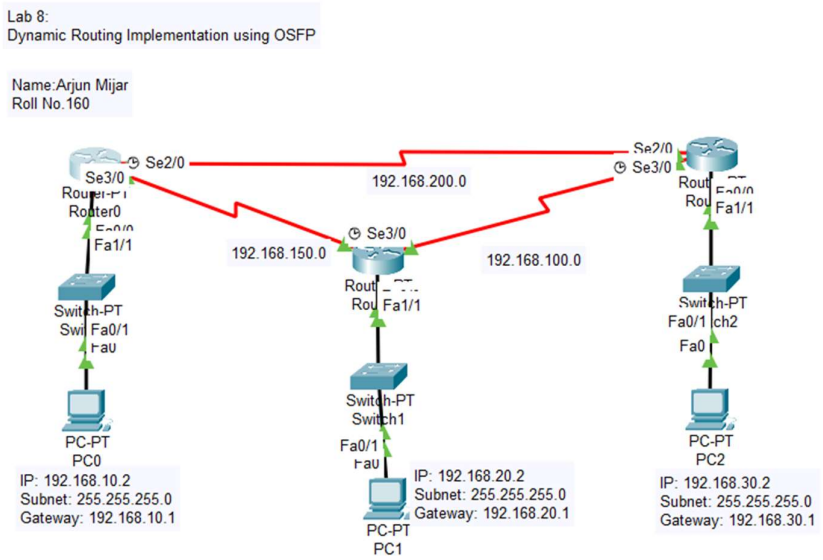
    - The output should display routes learned via OSPF, marked by the **O** symbol.

Step 4: Test Network Connectivity

1. **Ping Between PCs**:
    - Test connectivity by pinging from one PC (in a network connected to one router) to another PC (in a different network connected to another router).
    - Successful pings confirm that OSPF has enabled routing between different network segments.
2. **Traceroute**:

- Use the **tracert** command from a PC to trace the path taken by packets across the network to confirm correct routing:

  tracert <destination IP>



Conclusion:

OSPF offers a scalable and efficient solution for dynamic routing in large or complex networks. By implementing OSPF, routers can automatically share routing information and adjust to network changes without manual intervention. Its faster convergence, support for CIDR, and hierarchical structure make it an ideal choice for large-scale networks. In this lab, we successfully configured OSPF to dynamically route traffic between different network segments, demonstrating the flexibility and adaptability of this protocol in dynamic environments. Through OSPF, network administrators can ensure reliable and optimized packet delivery, even in rapidly changing network topologies.

LAB:9

AIM: VLAN Setup and Inter-VLAN Routing

Theory:

A **Virtual Local Area Network (VLAN)** is a logical grouping of devices in a network, enabling segmentation of the network at Layer 2 of the OSI model. VLANs allow devices on different physical LANs to communicate as if they were on the same network. This segmentation improves network performance, security, and manageability by reducing broadcast domains and isolating different types of network traffic.

However, devices on different VLANs cannot communicate directly with each other by default. **Inter-VLAN Routing** is required to enable communication between devices on different VLANs. This is achieved through a router or a Layer 3 switch that can route traffic between VLANs. By implementing VLANs, administrators can effectively separate user groups, improve network security, and optimize traffic flow.

***Key Concepts**:*

- **VLAN**: A virtual LAN that logically segments a physical network into smaller networks.
- **Inter-VLAN Routing**: The process of routing traffic between VLANs using a router or Layer 3 switch.
- **Trunk Ports**: Ports that allow traffic from multiple VLANs to pass through. Trunks carry VLAN tagging information, allowing VLANs to be identified as traffic passes through network devices.

Working Procedure:

Step 1: Create the Network Topology

1. **Add Devices**:
    - In Cisco Packet Tracer, add one Layer 2 switch, a router (or Layer 3 switch), and several PCs.
    - Connect the PCs to the switch using Ethernet cables.
    - Connect the switch to the router using a trunk link (Ethernet cable).

Step 2: VLAN Configuration on the Switch

1. **Access the Switch**:
    - Click on the switch and open the **CLI**. Enter **privileged EXEC mode** by typing enable.

2. **Create VLANs**:
    - Define multiple VLANs to segment the network. For example, VLAN 10 for the sales department and VLAN 20 for the IT department:

      Switch(config)# vlan 10

```
Switch(config-vlan)# name Sales
Switch(config)# vlan 20
Switch(config-vlan)# name IT
```

3. **Assign Ports to VLANs**:
   - Assign specific switch ports to the VLANs. For example, assign **Fa0/1** to VLAN 10 and **Fa0/2** to VLAN 20:

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

Switch(config)# interface fastEthernet 0/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 20
```

4. **Configure Trunk Port**:
   - Configure the switch port that connects to the router as a trunk port to carry traffic for multiple VLANs:

```
Switch(config)# interface fastEthernet 0/24
Switch(config-if)# switchport mode trunk
```

Step 3: Inter-VLAN Routing on the Router

1. **Access the Router**:
   - Open the **CLI** of the router. Enter **privileged EXEC mode** by typing enable.
2. **Configure Subinterfaces**:
   - Since each VLAN needs a gateway, configure subinterfaces on the router for each VLAN using **Router-on-a-Stick**:

```
Router(config)# interface gigabitEthernet 0/0.10
Router(config-subif)# encapsulation dot1Q 10
Router(config-subif)# ip address 192.168.10.1 255.255.255.0

Router(config)# interface gigabitEthernet 0/0.20
Router(config-subif)# encapsulation dot1Q 20
Router(config-subif)# ip address 192.168.20.1 255.255.255.0
```

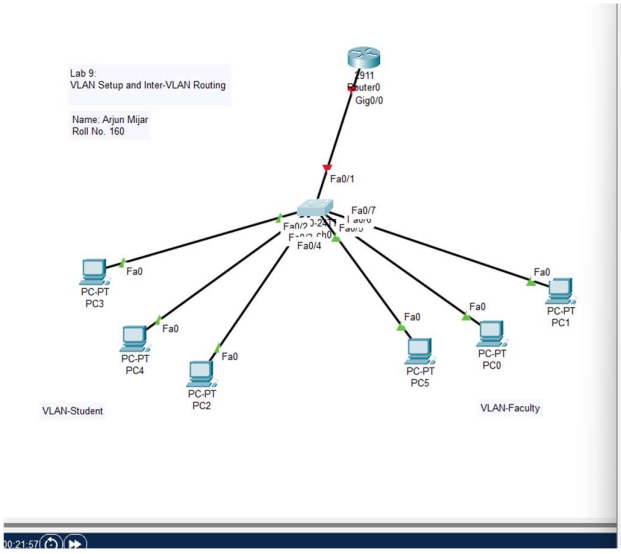
   - This configures **subinterfaces** for each VLAN, allowing the router to route traffic between the VLANs.

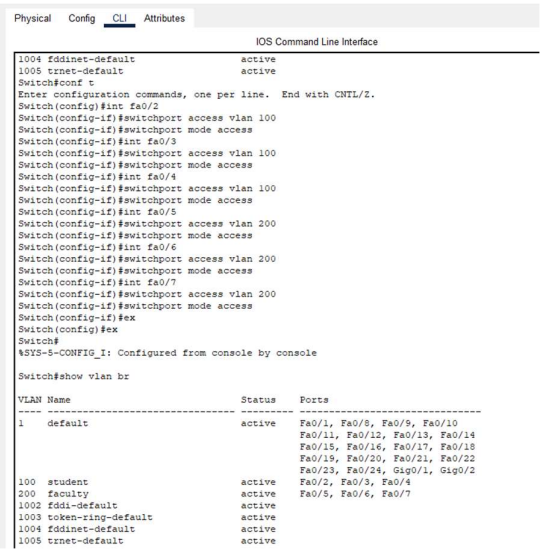
3. **Verify Routing**:
    - Ensure routing between the VLANs by checking the routing table with:

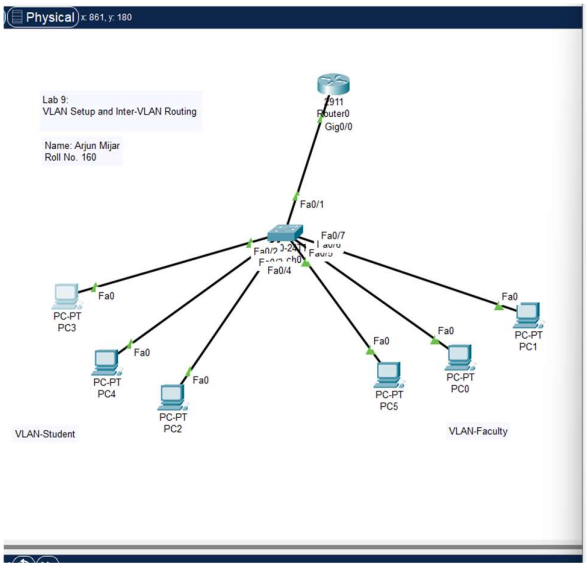
      Router# show ip route

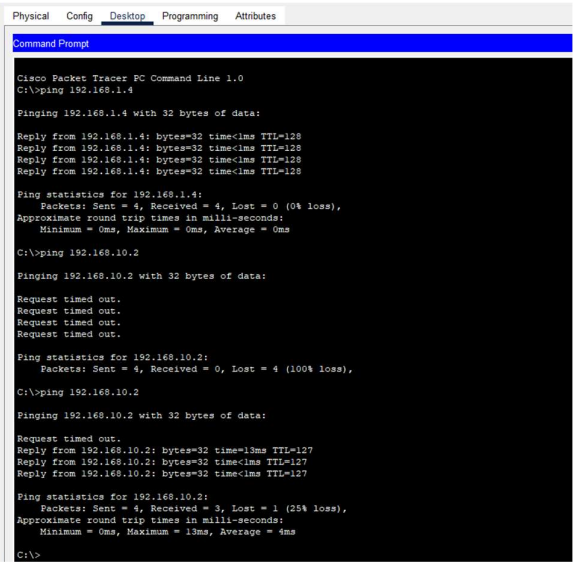
Step 4: Test Connectivity

1. **Verify VLAN Setup**:
    - Test connectivity between devices in the same VLAN by using the **ping** command. Devices in the same VLAN should communicate successfully.
2. **Verify Inter-VLAN Routing**:
    - Test connectivity between devices in different VLANs. If Inter-VLAN routing is correctly configured, PCs in VLAN 10 should be able to communicate with PCs in VLAN 20 using their respective gateways.









Conclusion:

By setting up VLANs and configuring Inter-VLAN routing, we were able to segment the network into multiple logical networks while still allowing communication between them when necessary. VLANs improve network performance and security by reducing broadcast traffic and isolating user groups. Inter-VLAN routing ensures that these isolated networks can communicate through a Layer 3 device, such as a router or Layer 3 switch. This setup demonstrates the flexibility and scalability that VLANs and Inter-VLAN routing bring to modern network designs, making it easier to manage complex network infrastructures.

LAB:10

AIM: Router Access List Configuration

Theory:

Access Control Lists (ACLs) are a fundamental security mechanism used in networking to control traffic flow, ensuring that only authorized packets are permitted to pass through a router or switch. ACLs are used to filter traffic based on defined criteria such as source/destination IP addresses, protocols, and ports.

There are two main types of ACLs:

1. **Standard ACL**: Filters traffic based only on source IP addresses. These ACLs are simpler but provide less granular control over traffic.
2. **Extended ACL**: Filters traffic based on multiple factors such as source/destination IP addresses, protocols (TCP, UDP, ICMP), and port numbers, offering more precise traffic control.

ACLs can be applied to inbound or outbound traffic on router interfaces, helping to improve network security, manage traffic, and control access to network resources. When properly configured, ACLs can prevent unauthorized access to sensitive parts of the network, block malicious traffic, and optimize bandwidth usage by blocking unwanted traffic.

Working Procedure:

Step 1: Set Up the Network

1. **Create the Topology**:
   - Add routers, switches, and PCs in Cisco Packet Tracer (or any network simulator).
   - Use Ethernet cables to connect the routers to the switches, and PCs to the switches.
2. **Assign IP Addresses**:
   - Configure IP addresses for the router interfaces and the PCs.

Step 2: Configure Standard ACLs

1. **Access the Router**:
   - Open the **CLI** on the router and enter **privileged EXEC mode**:

     enable

2. **Create a Standard ACL**:
   - Standard ACLs use numbers between **1–99**. To create a basic standard ACL, use the following commands:

     Router(config)# access-list 10 permit 192.168.1.0 0.0.0.255

This allows traffic from the **192.168.1.0/24** network.

3. **Apply the Standard ACL to an Interface**:
   - ACLs must be applied to an interface in either an inbound or outbound direction:

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

This applies the ACL to inbound traffic on the router's interface.

Step 3: Configure Extended ACLs

1. **Create an Extended ACL**:
   - Extended ACLs use numbers between **100–199**. They offer more control over traffic by allowing filtering based on protocol and port numbers:

```
Router(config)# access-list 110 deny tcp 192.168.1.0 0.0.0.255 any eq 80
```

This denies all HTTP (port 80) traffic from the **192.168.1.0/24** network.

2. **Permit All Other Traffic**:
   - After denying specific traffic, it's important to allow other traffic:

```
Router(config)# access-list 110 permit ip any any
```

3. **Apply the Extended ACL to an Interface**:
   - Apply the extended ACL to the appropriate router interface:

```
Router(config)# interface gigabitEthernet 0/1
Router(config-if)# ip access-group 110 out
```

This applies the ACL to outbound traffic.

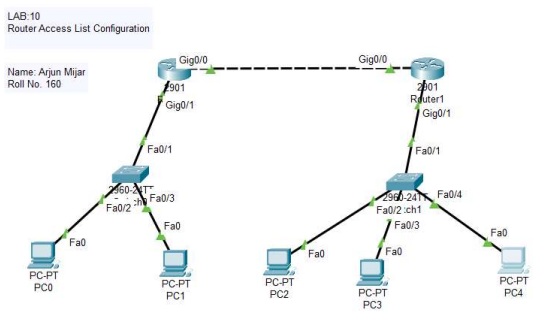
Step 4: Verify the Configuration

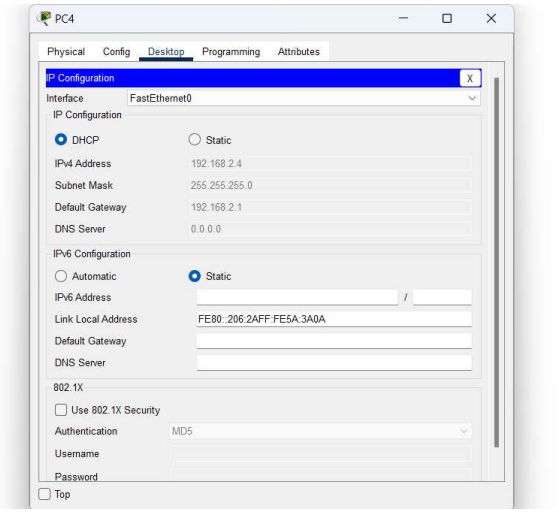
1. **Use the show Commands**:
   - To view the ACL configuration, use the following command:

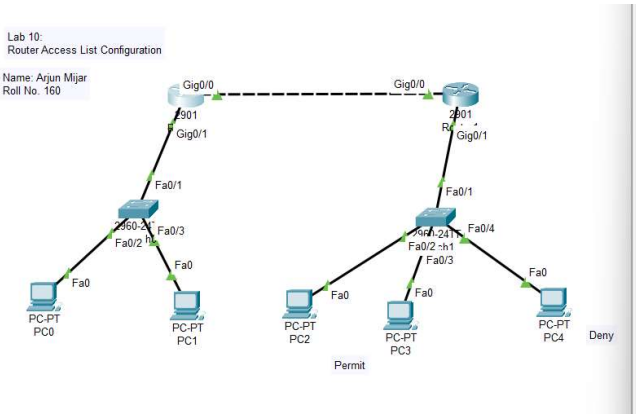
```
Router# show access-lists
```

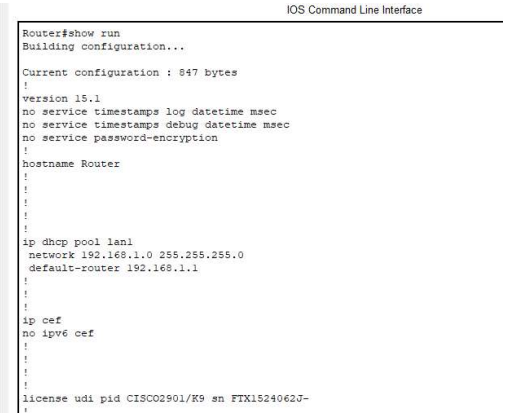
2. **Test Connectivity**:

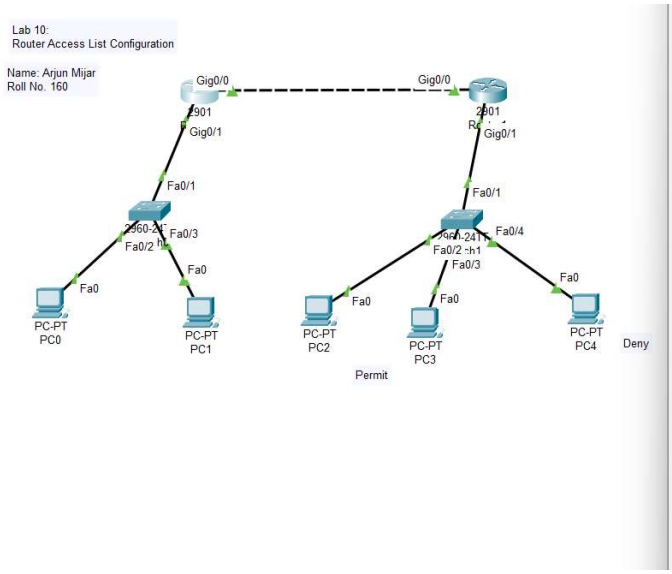
- Use the **ping** command from PCs to test the network. Traffic that matches the ACL rules will be either blocked or permitted according to the ACL configuration.

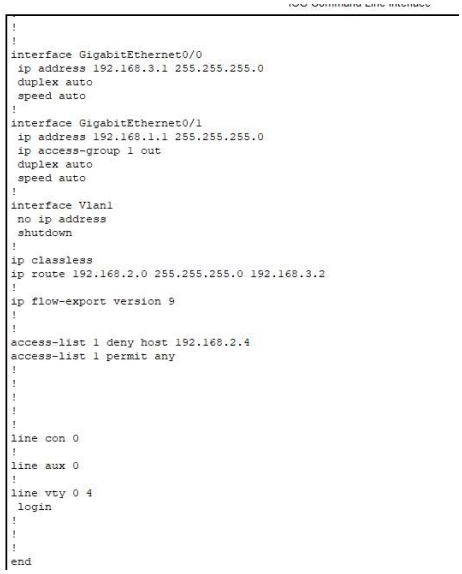




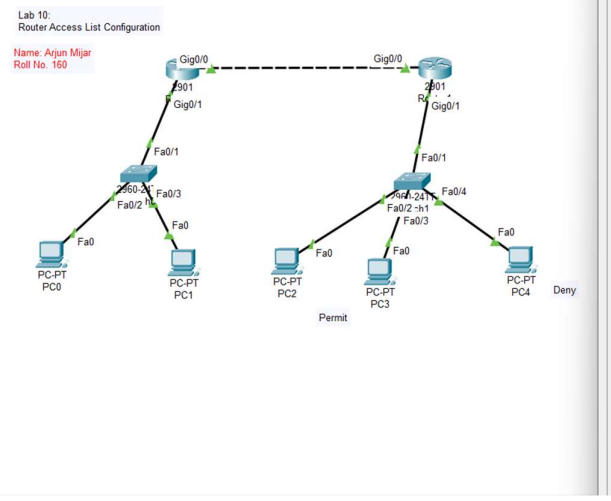


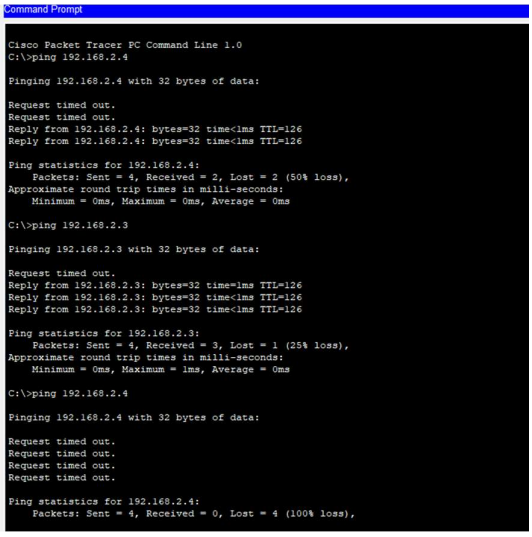






```
!
!
interface GigabitEthernet0/0
 ip address 192.168.3.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 ip access-group 1 out
 duplex auto
 speed auto
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
ip route 192.168.2.0 255.255.255.0 192.168.3.2
!
ip flow-export version 9
!
!
access-list 1 deny host 192.168.2.4
access-list 1 permit any
!
!
!
!
!
line con 0
!
line aux 0
!
line vty 0 4
 login
!
!
!
end
```





```
Command Prompt

Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.2.4

Pinging 192.168.2.4 with 32 bytes of data:

Request timed out.
Request timed out.
Reply from 192.168.2.4: bytes=32 time<1ms TTL=126
Reply from 192.168.2.4: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time<1ms TTL=126
Reply from 192.168.2.3: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 192.168.2.4

Pinging 192.168.2.4 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Conclusion:

In this lab, we successfully configured both **Standard** and **Extended Access Control Lists (ACLs)** on a router. By using ACLs, we were able to control which types of traffic were allowed or denied, based on criteria such as IP addresses, protocols, and ports. This allows for greater security and traffic management in network environments. While Standard ACLs provide basic filtering capabilities, Extended ACLs allow for more precise control, which is critical in more complex networks. Overall, router ACL configuration is essential for network security, ensuring that only authorized traffic flows through the network.