

Ninad Karkhanis

# Computer Networks Journal

---

Seat number - 31011221031

Roll number - 30

# INDEX

<u>PRACTICAL NUMBER</u>	PRACTICAL NAME	PRACTICAL DATE
1	Finding out IP Address	03/03/23
2	Commands	10/02/23
3	Static Routing Configuration	13/01/23
4	RIP Configuration	17/01/22
5	OSPF Configuration	27/01/23
6	DHCP & DNS Configuration	06/01/23
7	Study of firewall implementation using ACL	03/02/23
8	Wireshark Analysis	17/02/23
9	Socket Programming	10/03/23

## Practical 1: IP Address

### What is an IP Address?

An Internet Protocol address is a numerical label such as 192.0.2.1 that is connected to a computer network that uses the Internet Protocol for communication. An IP address serves two main functions: network interface identification and location addressing.

### **Classes of IP Address, Range, Network Address, Host Address, Binary Notation, Subnet Mask with Example**

<u>Class</u>	<u>Range</u>	<u>IP Address</u>	<u>Network ID</u>	<u>Host ID</u>	<u>Subnet Mask</u>	<u>Binary Notation</u>
A	0-127	110.10.10.2	110.0.0.0	0.10.10.2	255.0.0.0	0110111000001010 0000101000000010
B	128-191	130.100.20.3	130.100.0.0	0.0.20.3	255.255.0.0	1000001000001010 0000101000000011
C	192-223	192.10.15.6	192.10.15.0	0.0.0.6	255.255.255.0	1100000000001010 0000111100000110
D	224-243	235.18.25.8	235.18.25.0	0.0.0.8	255.255.255.0	1110101100010010 0001100100001000
E	244-255	250.21.10.1	250.21.10.0	0.0.0.1	255.255.255.0	111101000010101 0000101000000001

## Practical 2: Commands

### Command 1:

**ipconfig :-** The ipconfig command displays the basic IP addressing information for each network interface on the Windows system. This information includes both the IP address and subnet mask.

```
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ninad>ipconfig

Windows IP Configuration

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . : iball.wifi.net
    Link-local IPv6 Address . . . . . : fe80::efa9:18bc:2890:cfa1%8
    IPv4 Address. . . . . : 192.168.1.210
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1
```

## Command 2 :

**ping ip\_address(192.168.0.29 => IPv4 Address we got from entering ipconfig command in cmd prompt) :** - The ping command sends one datagram per second and prints one line of output for every response received. The ping command calculates round-trip times and packet loss statistics, and displays a brief summary on completion. The ping command completes when the program times out or on receipt of a SIGINT signal.

```
Pinging 192.168.1.210 with 32 bytes of data:
Reply from 192.168.1.210: bytes=32 time<1ms TTL=128
Reply from 192.168.1.210: bytes=32 time<1ms TTL=128
Reply from 192.168.1.210: bytes=32 time<1ms TTL=128
Reply from 192.168.1.210: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.210:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



### **Command 3 :**

**netstat :-** The netstat command shows active network connections and displays information about them. It includes the status of the connection, the networking protocol, local and remote computer IP address, and more.

```
C:\Users\ninad>netstat
```

#### Active Connections

Proto	Local Address	Foreign Address	State
TCP	127.0.0.1:49784	checkhost:49785	ESTABLISHED
TCP	127.0.0.1:49785	checkhost:49784	ESTABLISHED
TCP	127.0.0.1:49789	checkhost:49808	ESTABLISHED
TCP	127.0.0.1:49808	checkhost:49789	ESTABLISHED
TCP	192.168.1.210:49692	ec2-65-2-109-57:https	ESTABLISHED
TCP	192.168.1.210:49695	ec2-65-2-109-57:https	ESTABLISHED
TCP	192.168.1.210:63596	13.107.5.91:https	ESTABLISHED
TCP	192.168.1.210:63612	192.168.1.205:8009	ESTABLISHED
TCP	192.168.1.210:63628	192.168.1.205:8008	ESTABLISHED
TCP	192.168.1.210:63646	20.198.118.190:https	ESTABLISHED
TCP	192.168.1.210:63699	sb-in-f188:5228	ESTABLISHED
TCP	192.168.1.210:63707	192.168.1.205:8009	ESTABLISHED
TCP	192.168.1.210:63763	ec2-65-1-118-171:https	ESTABLISHED
TCP	192.168.1.210:63801	whatsapp-cdn-shv-01-pnq1:https	ESTABLISHED
TCP	192.168.1.210:63924	vip01:https	CLOSE_WAIT
TCP	192.168.1.210:63930	vip01:https	CLOSE_WAIT
TCP	192.168.1.210:63931	ec2-13-49-204-92:https	CLOSE_WAIT
TCP	192.168.1.210:63932	vip01:https	CLOSE_WAIT
TCP	192.168.1.210:63933	104.16.226.10:https	ESTABLISHED
TCP	192.168.1.210:63934	vip02:https	CLOSE_WAIT
TCP	192.168.1.210:63935	104.16.227.10:https	ESTABLISHED
TCP	192.168.1.210:63936	104.16.226.10:https	ESTABLISHED
TCP	192.168.1.210:63937	104.16.226.10:https	ESTABLISHED
TCP	192.168.1.210:63938	104.16.227.10:https	ESTABLISHED
TCP	192.168.1.210:63939	151.101.1.208:https	ESTABLISHED
TCP	192.168.1.210:63945	any-in-2015:https	ESTABLISHED
TCP	192.168.1.210:63948	bom07s45-in-f6:https	ESTABLISHED
TCP	192.168.1.210:63952	a23-212-254-91:https	ESTABLISHED
TCP	192.168.1.210:63953	104.16.227.10:https	ESTABLISHED
TCP	192.168.1.210:63954	104.16.226.10:https	ESTABLISHED
TCP	192.168.1.210:63960	20.50.201.200:https	TIME_WAIT
TCP	192.168.1.210:63962	a23-217-53-76:https	CLOSE_WAIT
TCP	192.168.1.210:63963	a23-217-53-76:https	CLOSE_WAIT
TCP	192.168.1.210:63964	a23-217-53-76:https	CLOSE_WAIT
TCP	192.168.1.210:63965	a23-217-53-76:https	CLOSE_WAIT
TCP	192.168.1.210:63966	a23-217-53-76:https	CLOSE_WAIT
TCP	192.168.1.210:63967	a23-217-53-76:https	CLOSE_WAIT
TCP	192.168.1.210:63973	a23-217-53-31:https	CLOSE_WAIT
TCP	192.168.1.210:63974	a23-217-53-31:https	CLOSE_WAIT
TCP	192.168.1.210:63975	a23-217-53-31:https	CLOSE_WAIT
TCP	192.168.1.210:63976	a23-217-53-31:https	CLOSE_WAIT
TCP	192.168.1.210:63980	a23-205-88-40:https	CLOSE_WAIT
TCP	192.168.1.210:63981	152.195.38.76:http	ESTABLISHED
TCP	192.168.1.210:63983	ec2-52-87-89-140:https	CLOSE_WAIT
TCP	192.168.1.210:63984	51.132.193.104:https	TIME_WAIT

## **Command 4 :**

**arp :-** The arp command displays and modifies the Internet-to-adaptor address translation tables used by the Address in Networks and communication management. The arp command displays the current ARP entry for the host specified by the HostName variable.

```
C:\Users\ninad>arp

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

    -a          Displays current ARP entries by interrogating the current
                  protocol data.  If inet_addr is specified, the IP and Physical
                  addresses for only the specified computer are displayed.  If
                  more than one network interface uses ARP, entries for each ARP
                  table are displayed.
    -g          Same as -a.
    -v          Displays current ARP entries in verbose mode.  All invalid
                  entries and entries on the loop-back interface will be shown.
inet_addr      Specifies an internet address.
-N if_addr     Displays the ARP entries for the network interface specified
                  by if_addr.
-d            Deletes the host specified by inet_addr.  inet_addr may be
                  wildcarded with * to delete all hosts.
-s            Adds the host and associates the Internet address inet_addr
                  with the Physical address eth_addr.  The Physical address is
                  given as 6 hexadecimal bytes separated by hyphens.  The entry
                  is permanent.
eth_addr       Specifies a physical address.
if_addr        If present, this specifies the Internet address of the
                  interface whose address translation table should be modified.
                  If not present, the first applicable interface will be used.

Example:
> arp -s 157.55.85.212 00-aa-00-62-c6-09 .... Adds a static entry.
> arp -a          .... Displays the arp table.
```

## **Utilities of arp command**



**arp -a :-** This command is used to display the ARP table for a particular IP address. It also shows all the entries of the ARP cache or table.

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

Interface: 192.168.1.210 --- 0x8
    Internet Address      Physical Address      Type
    192.168.1.1           a0-47-d7-2a-41-30     dynamic
    192.168.1.205         64-e0-03-b5-f3-05     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
    239.192.152.143       01-00-5e-40-98-8f     static
    239.255.255.250       01-00-5e-7f-ff-fa     static
```

**arp -g :-** This command works the same as the arp -a command.

```
C:\Users\ninad>arp -g

Interface: 192.168.1.210 --- 0x8
    Internet Address      Physical Address      Type
    192.168.1.1           a0-47-d7-2a-41-30     dynamic
    192.168.1.205         64-e0-03-b5-f3-05     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
    239.192.152.143       01-00-5e-40-98-8f     static
    239.255.255.250       01-00-5e-7f-ff-fa     static
```

**arp -d :-** This command is used when you want to delete an entry from the ARP table for a particular interface. To delete an entry, write arp -d command along with the IP address in a command prompt you want to delete.

**arp -s :-** This command is used to add the static entry in the ARP table, which resolves the InetAddr (IP address) to the EtherAddr (physical address). To add a static entry in an ARP table, write arp -s command along with the IP address and MAC address of the device in a command prompt.

## **Command 5 :**

**tracert url :-** This command is used to trace route of different URLs.

```

C:\Users\ninad>tracert google.com

Tracing route to google.com [142.250.67.206]
over a maximum of 30 hops:

  1    14 ms    1 ms    1 ms  192.168.1.1
  2     5 ms    2 ms    2 ms  103.50.76.202
  3     3 ms    2 ms    7 ms  103.50.76.201
  4    28 ms    4 ms    5 ms  103.27.170.11
  5    19 ms    9 ms    5 ms  108.170.248.161
  6     5 ms    5 ms    5 ms  142.250.235.9
  7     4 ms    4 ms    4 ms  bom12s08-in-f14.1e100.net [142.250.67.206]

Trace complete.

```

### **Command 6 :**

**tracert ip\_address :-** This command is used to trace route of ip addresses.

```

C:\Users\ninad>tracert 100.10.20.5

Tracing route to pool-100-10-20-5.prvdr1.fios.verizon.net [100.10.20.5]
over a maximum of 30 hops:

  1    36 ms    1 ms    5 ms  192.168.1.1
  2    12 ms   16 ms   36 ms  103.50.76.202
  3    13 ms    3 ms    3 ms  103.50.76.201
  4     7 ms    4 ms    4 ms  103.248.63.193
  5    14 ms   23 ms   10 ms  59.163.29.81.static.vsnl.net.in [59.163.29.81]
  6    39 ms   14 ms    5 ms  172.28.133.173
  7    14 ms   23 ms   26 ms  ix-ae-0-100.tcore1.mlv-mumbai.as6453.net [180.87.38.5]
  8   200 ms  225 ms  200 ms  if-ae-2-2.tcore2.mlv-mumbai.as6453.net [180.87.38.2]
  9      *      *      *      Request timed out.
 10   206 ms  197 ms  196 ms  if-ae-66-2.tcore3.nto-newyork.as6453.net [80.231.130.106]
 11   201 ms  193 ms  192 ms  et-8-0-5.BR2.NYC4.ALTER.NET [152.179.120.229]
 12   203 ms  223 ms  204 ms  ae204-0.PRVDRI-VFTTP-304.verizon-gni.net [100.41.129.53]
 13   211 ms  224 ms  216 ms  pool-100-10-20-5.prvdr1.fios.verizon.net [100.10.20.5]

Trace complete.

```

### **Command 7 :**

**route Print :-** This command is used to display entire routing table information. The output of the route print command includes the following sections: Interface List, IPv4 Route Table, and IPv6 Route Table. The persistent Routes section shows permanent static routes.

```

C:\Users\ninad>route Print
=====
Interface List
18...12 f4 8d 0d 8b e7 .....Microsoft Wi-Fi Direct Virtual Adapter
17...22 f4 8d 0d 8b e7 .....Microsoft Wi-Fi Direct Virtual Adapter #2
 8...00 f4 8d 0d 8b e7 .....Qualcomm Atheros AR956x Wireless Network Adapter
1.....Software Loopback Interface 1
=====

IPv4 Route Table
=====
Active Routes:
Network Destination        Netmask          Gateway          Interface        Metric
0.0.0.0                    0.0.0.0          192.168.1.1      192.168.1.210    50
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          192.168.1.210    306
192.168.1.210              255.255.255.255  On-link          192.168.1.210    306
192.168.1.255              255.255.255.255  On-link          192.168.1.210    306
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          192.168.1.210    306
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          192.168.1.210    306
=====

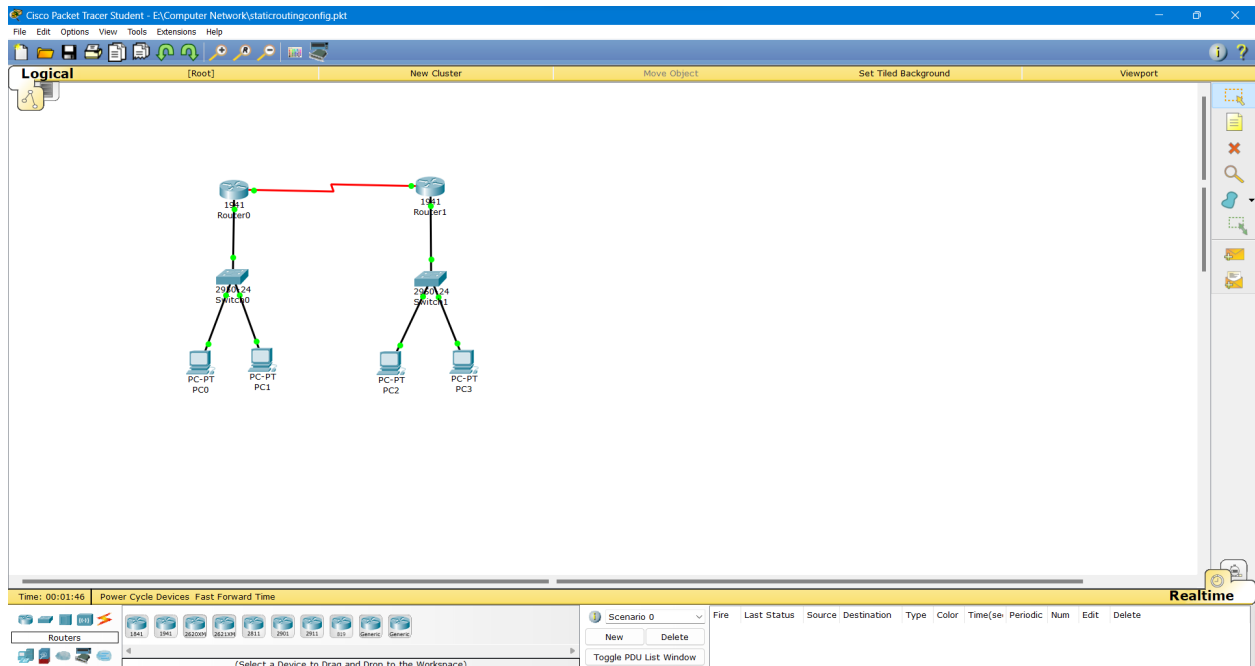
Persistent Routes:
None

IPv6 Route Table
=====
Active Routes:
If Metric Network Destination      Gateway
1    331 ::1/128               On-link
8    306 fe80::/64            On-link
8    306 fe80::efa9:18bc:2890:cfa1/128 On-link
1    331 ff00::/8              On-link
8    306 ff00::/8              On-link
=====

Persistent Routes:
None

```

## Practical 3 : Static Routing Configuration



### ROUTER1

```
>enable
```

```
conf t
```

```
#host r1
```

```
#int serial0/1/0
```

```
#ip add 10.10.0.1 255.0.0.0
```

```
#no shut
```

```
#exit
```

```
#int g0/0
```

```
#ip add 192.168.10.1 255.255.255.0
```

```
#no shut
```



```
#exit
```

```
#ip add 172.10.10.0 255.255.255.0 10.10.0.2
```

```
#^Z
```

```
#show ip route
```

## **ROUTER2**

```
>enable
```

```
conf t
```

```
#host r1
```

```
#int serial0/1/0
```

```
#ip add 10.10.0.1 255.0.0.0
```

```
#no shut
```

```
#exit
```

```
#int g0/0
```

```
#ip add 172.168.10.1 255.255.0.0
```

```
#no shut
```

```
#exit
```

```
#ip add ip route 192.168.10.0 255.255.255.0 10.10.0.1
```

```
#^Z
```

```
#show ip route
```

Router0

Physical Config CLI

### IOS Command Line Interface

```
R1(config)#ip route 172.168.10.0 255.255.255.0 100.10.10.2
R1(config)#^Z
R1#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

      100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       100.0.0.0/8 is directly connected, Serial0/1/0
L       100.10.10.1/32 is directly connected, Serial0/1/0
      172.168.0.0/24 is subnetted, 1 subnets
S       172.168.10.0/24 [1/0] via 100.10.10.2
      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, GigabitEthernet0/0
L       192.168.10.1/32 is directly connected, GigabitEthernet0/0
```

Copy Paste

Router1

Physical Config CLI

### IOS Command Line Interface

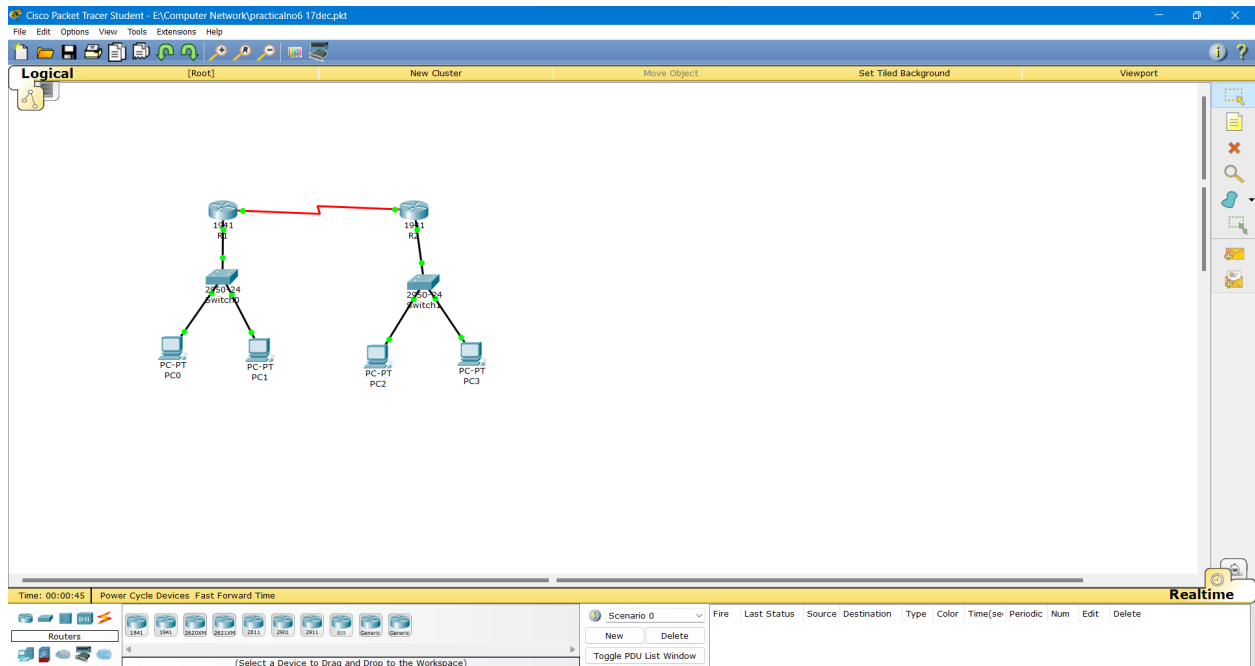
```
R2(config)#ip route 192.168.10.0 255.255.255.0 100.10.10.1
R2(config)#^Z
R2#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

      100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       100.0.0.0/8 is directly connected, Serial0/1/0
L       100.10.10.2/32 is directly connected, Serial0/1/0
      172.168.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.168.0.0/16 is directly connected, GigabitEthernet0/0
L       172.168.10.1/32 is directly connected, GigabitEthernet0/0
S       192.168.10.0/24 [1/0] via 100.10.10.1
R2#
```

Copy Paste

## Practical 4 : RIP Configuration



### ROUTER1

>enable

conf t

#host r1

#int serial0/1/0

#ip add 10.10.0.1 255.0.0.0

#no shut

#exit


#int g0/0

#ip add 192.168.0.1 255.255.255.0

#no shut

#exit

#router rip



```
#network 10.10.0.0
```

```
#network 192.168.0.0
```

```
^Z
```

```
#show ip route
```

## **ROUTER2**

```
>enable
```

```
conf t
```

```
#host r2
```

```
#int serial0/1/0
```

```
#ip add 10.10.0.2 255.0.0.0
```

```
#no shut
```

```
#exit
```

```
#int g0/0
```

```
#ip add 172.116.0.1 255.255.255.0
```

```
#no shut
```

```
#exit
```

```
#router rip
```

```
#network 10.10.0.0
```

```
#network 172.116.0.0
```

```
^Z
```

```
#show ip route
```



R1

PhysicalConfigCLI

IOS Command Line Interface

```
R1(config-router)#network 192.168.0.0
R1(config-router)#^Z
R1#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

      100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       100.0.0.0/8 is directly connected, Serial0/1/0
L       100.10.0.1/32 is directly connected, Serial0/1/0
R       172.10.0.0/16 [120/1] via 100.10.0.2, 00:00:11, Serial0/1/0
        192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.0.0/24 is directly connected, GigabitEthernet0/0
L       192.168.0.1/32 is directly connected, GigabitEthernet0/0
R1#
```

CopyPaste

R2

PhysicalConfigCLI

IOS Command Line Interface

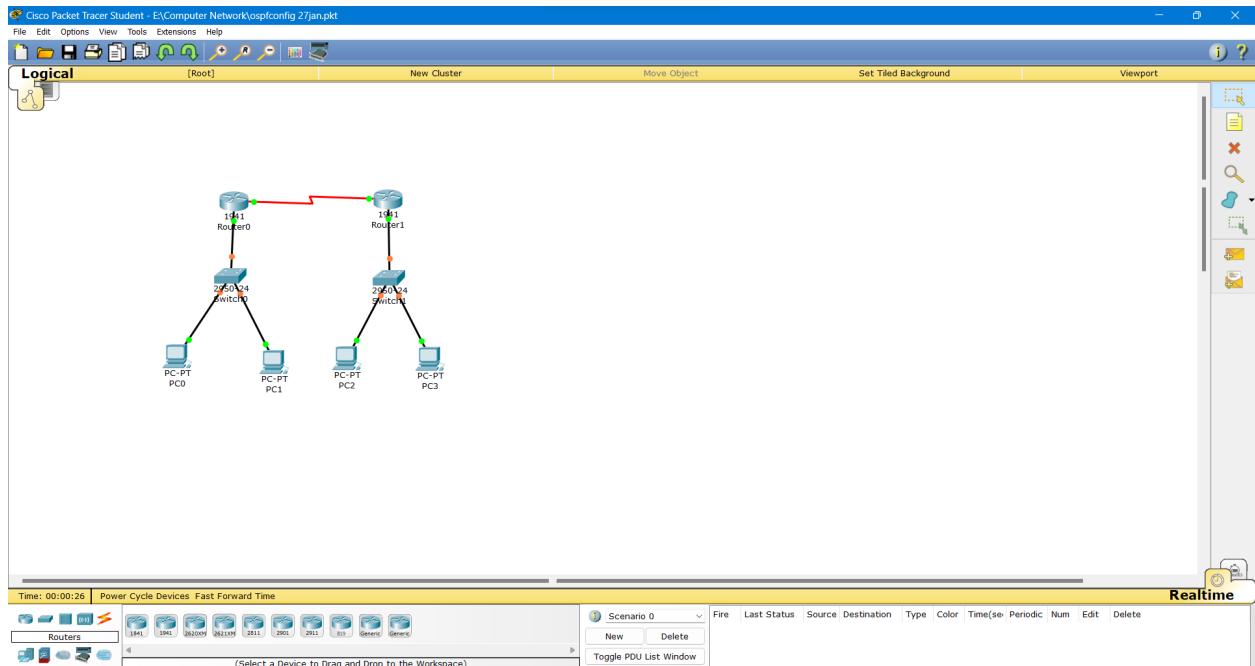
```
R2(config-router)#network 172.10.0.0
R2(config-router)#^Z
R2#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

      100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       100.0.0.0/8 is directly connected, Serial0/1/0
L       100.10.0.2/32 is directly connected, Serial0/1/0
        172.10.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.10.0.0/16 is directly connected, GigabitEthernet0/0
L       172.10.0.1/32 is directly connected, GigabitEthernet0/0
R       192.168.0.0/24 [120/1] via 100.10.0.1, 00:00:01, Serial0/1/0
R2#
```

CopyPaste

# Practical 5 : OSPF Configuration



## ROUTER 1

```
>enable
```

```
Router#conf t
```

```
#host R1
```

```
#int serial0/1/0
```

```
#ip add 198.168.20.1 255.255.255.0
```

```
#no shut
```


```
#exit
```

```
#int g0/0
```

```
#ip add 198.168.10.1 255.255.255.0
```

```
#no shut
```

```
#exit
```



```
#router ospf 1
```

```
#network 198.168.20.0 0.0.0.255 area 0
```

```
#network 198.168.10.0 0.0.0.255 area 0
```

```
^Z
```

```
#show ip route
```

## **ROUTER 2**

```
>enable
```

```
#conf t
```

```
#Host R2
```

```
#int serial0/1/0
```

```
#ip add 198.168.20.2 255.255.255.0
```

```
#no shut
```

```
#exit
```

```
#int g0/0
```

```
#ip add 198.168.30.1 255.255.255.0
```

```
#no shut
```

```
#exit
```

```
#router ospf 1
```

```
#network 198.168.20.0 0.0.0.255 area 0
```

```
#network 198.168.30.0 0.0.0.255 area 0
```

```
^Z
```

```
#show ip route
```

Router0

Physical Config CLI

### IOS Command Line Interface

```
R1(config-router)#network 192.168.10.0 0.0.0.255 area 0
R1(config-router)#^Z
R1#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

      192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, GigabitEthernet0/0
L       192.168.10.1/32 is directly connected, GigabitEthernet0/0
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.20.0/24 is directly connected, Serial0/1/0
L       192.168.20.1/32 is directly connected, Serial0/1/0
O       192.168.30.0/24 [110/65] via 192.168.20.2, 00:08:17, Serial0/1/0
R1#
```

Copy Paste

Router1

Physical Config CLI

### IOS Command Line Interface

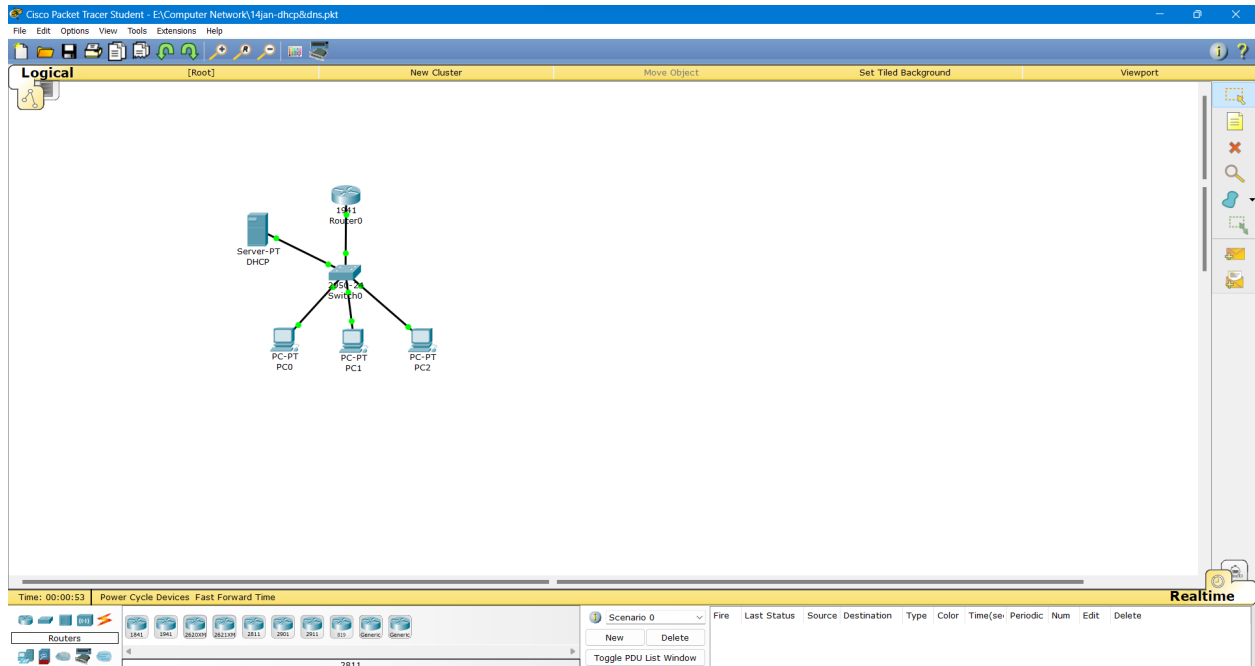
```
R2(config-router)#network 192.168.30.0 0.0.0.255 area 0
R2(config-router)#^Z
R2#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

O       192.168.10.0/24 [110/65] via 192.168.20.1, 00:10:11, Serial0/1/0
      192.168.20.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.20.0/24 is directly connected, Serial0/1/0
L       192.168.20.2/32 is directly connected, Serial0/1/0
      192.168.30.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.30.0/24 is directly connected, GigabitEthernet0/0
L       192.168.30.1/32 is directly connected, GigabitEthernet0/0
R2#
```

Copy Paste

## Practical 6 : DHCP & DNS Configuration



### ROUTER 1

```
>enable
```

```
#conf t
```

```
#Host r1
```

```
#int g0/0
```

```
#ip add 192.168.1.1 255.255.255.0
```

```
#no shut
```

```
#ip dhcp pool Lab4
```

```
#network 192.168.1.0 255.255.255.0
```

```
#default - router 192.168.1.1
```

```
#^ Z
```

## Server

DHCP

Physical

Config

Services

Desktop

Custom Interface

SERVICES

HTTP

DHCP

DHCPv6

TFTP

DNS

SYSLOG

AAA

NTP

EMAIL

FTP

DHCP

Interface 

FastEthernet0

 Service ☒ On ☐ Off

Pool Name 

lab4

Default Gateway 

192.168.1.1

DNS Server 

0.0.0.0

Start IP Address : 

19216811

Subnet Mask: 

2552552550

Maximum number of Users : 

255

TFTP Server: 

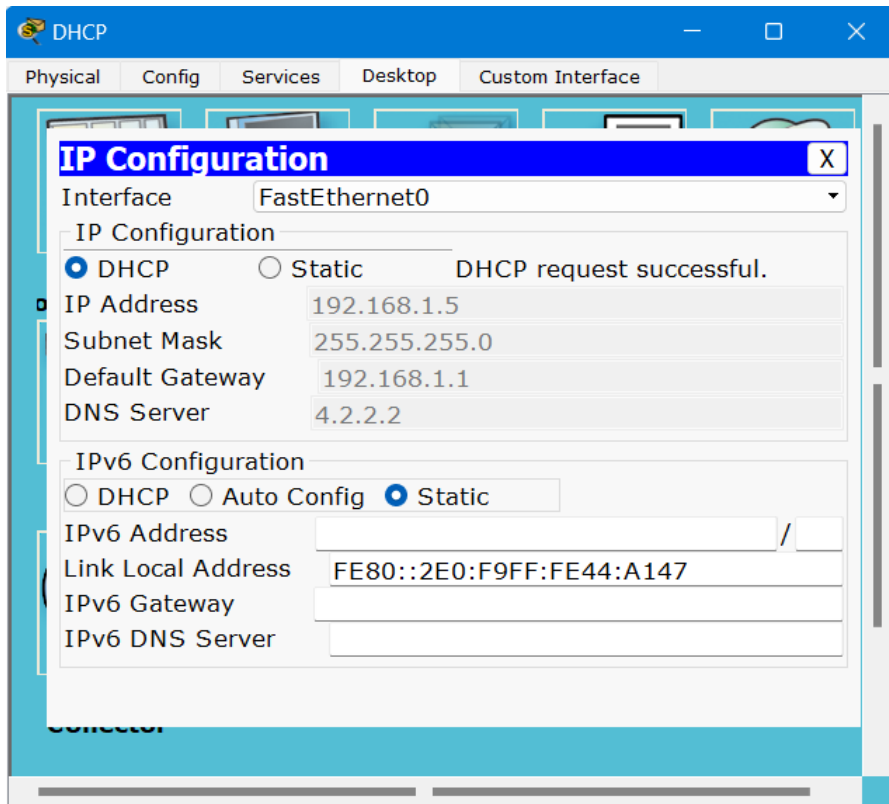
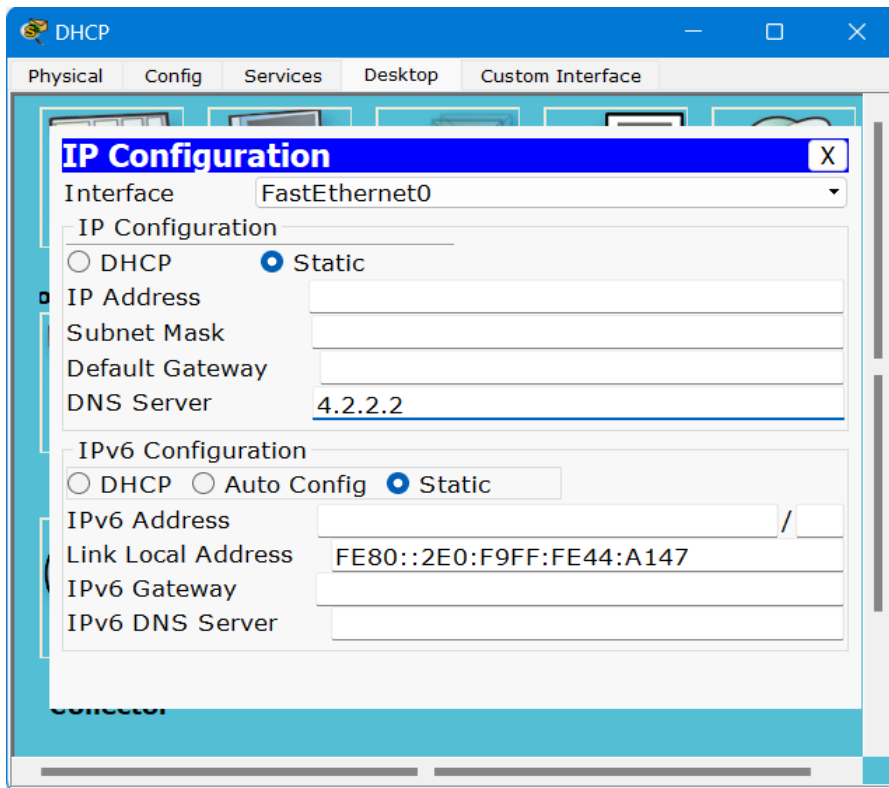
0.0.0.0

Add

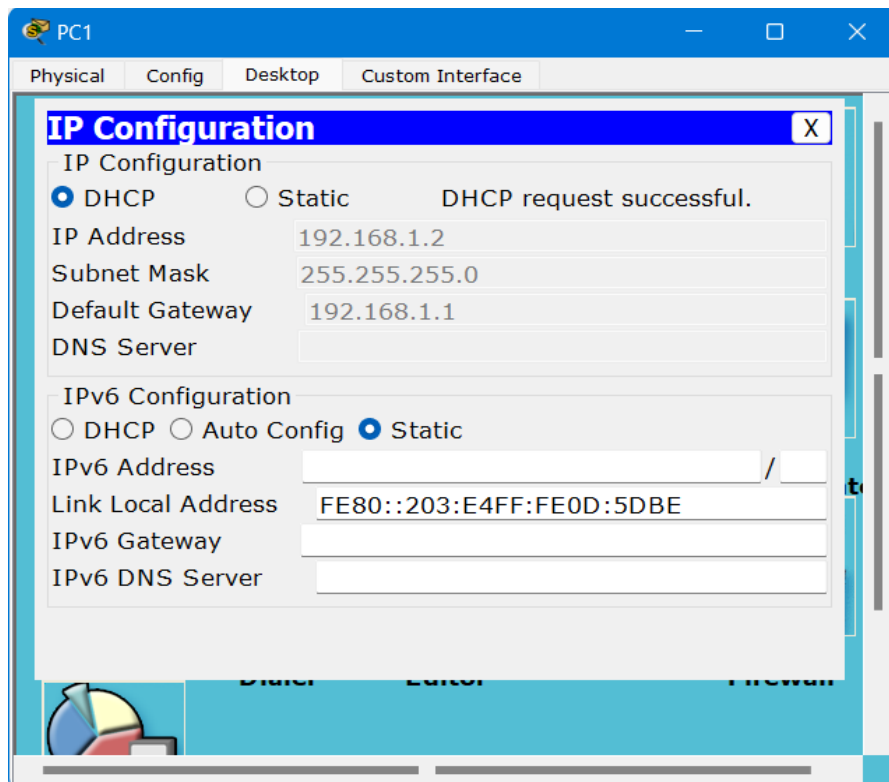
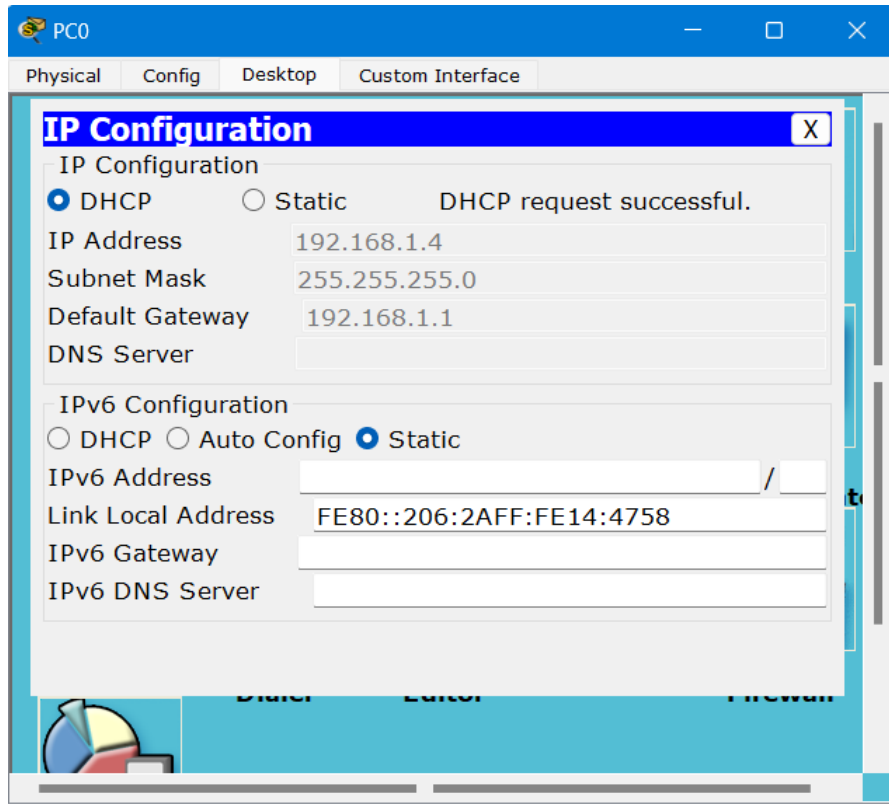
Save

Remove

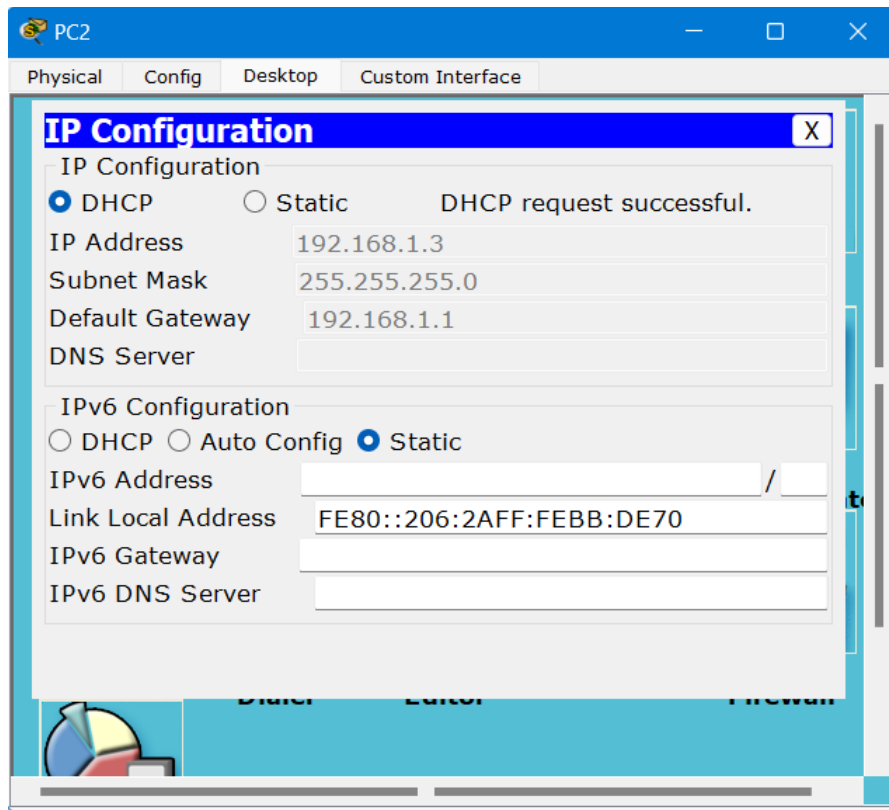
Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max Users	TFTP Server
lab4	192.168.1.1	0.0.0.0	192.168.1.1	255.255.255.0	255	0.0.0.0
serve...	0.0.0.0	0.0.0.0	192.168.1.1	255.255.255.0	255	0.0.0.0

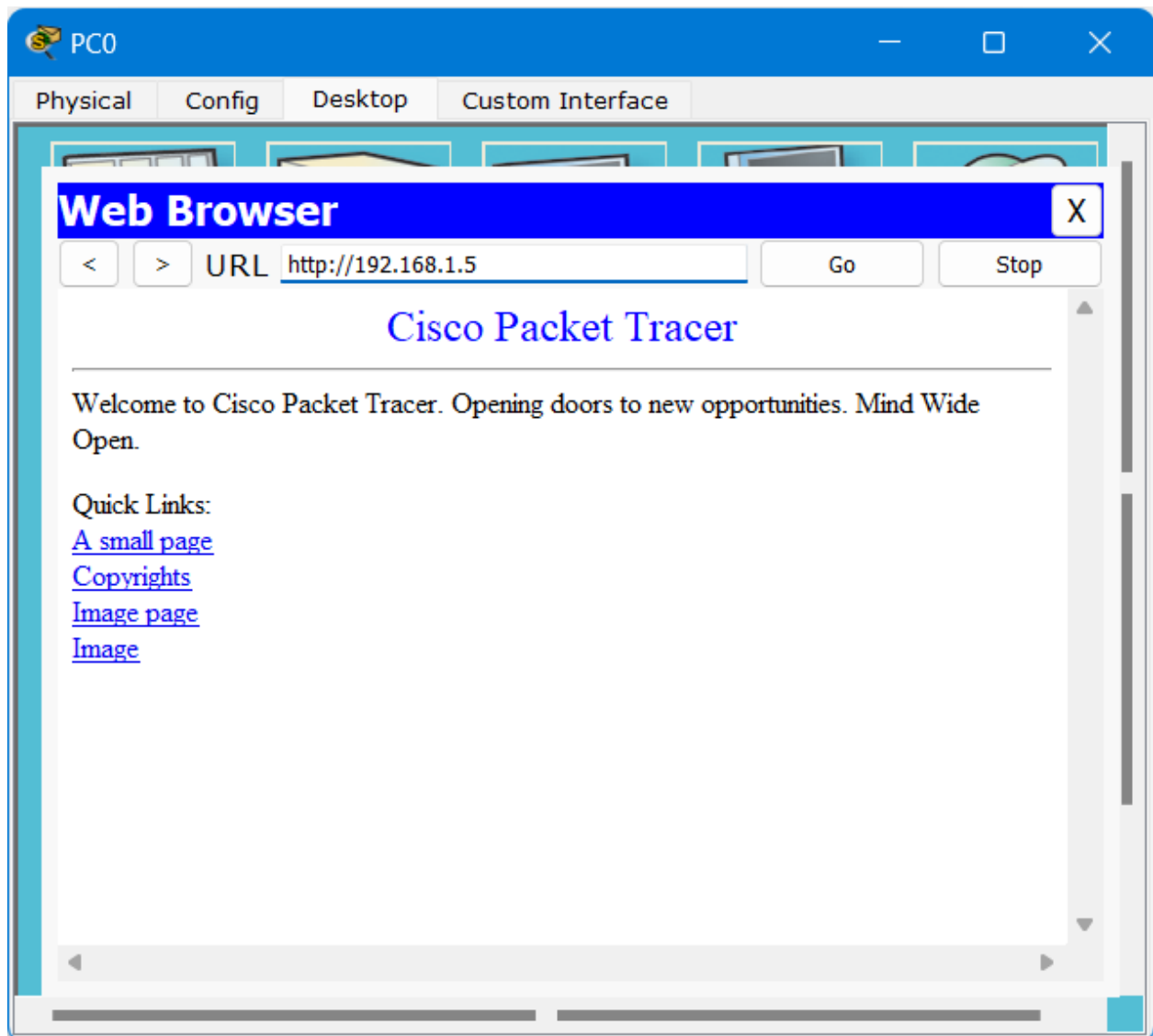


## PC

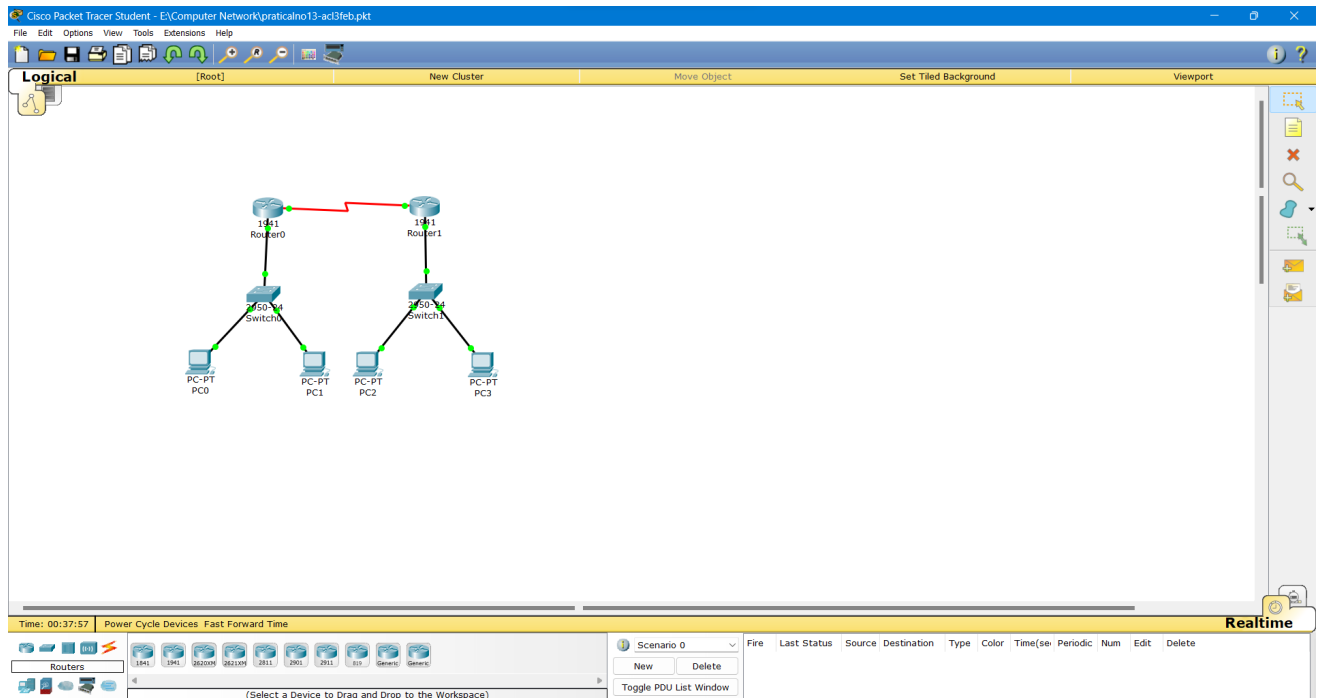








## Practical 7 : Study of firewall implementation using ACL



### ROUTER 1

>enable

#conf t

#host r1

#int serial 0/1/0


#ip add 10.0.0.1 255.0.0.0

#no shut

#exit

#int g0/0

#ip add 11.0.0.1 255.0.0.0



```
#no shut
```

```
#exit
```

```
#ip route 12.0.0.0 255.0.0.0 10.0.0.2
```

```
^z
```

```
#show ip route
```

## **ROUTER 2**

```
>enable
```

```
#conf t
```

```
#host r2
```

```
#int serial 0/1/0
```

```
#ip add 10.0.0.2 255.0.0.0
```

```
#no shut
```

```
#exit
```

```
#g0/0
```

```
#ip add 12.0.0.1 255.0.0.0
```

```
#no shut
```

```
#exit
```

```
#ip route 11.0.0.0 255.0.0.0 10.0.0.1
```

```
#show ip route
```

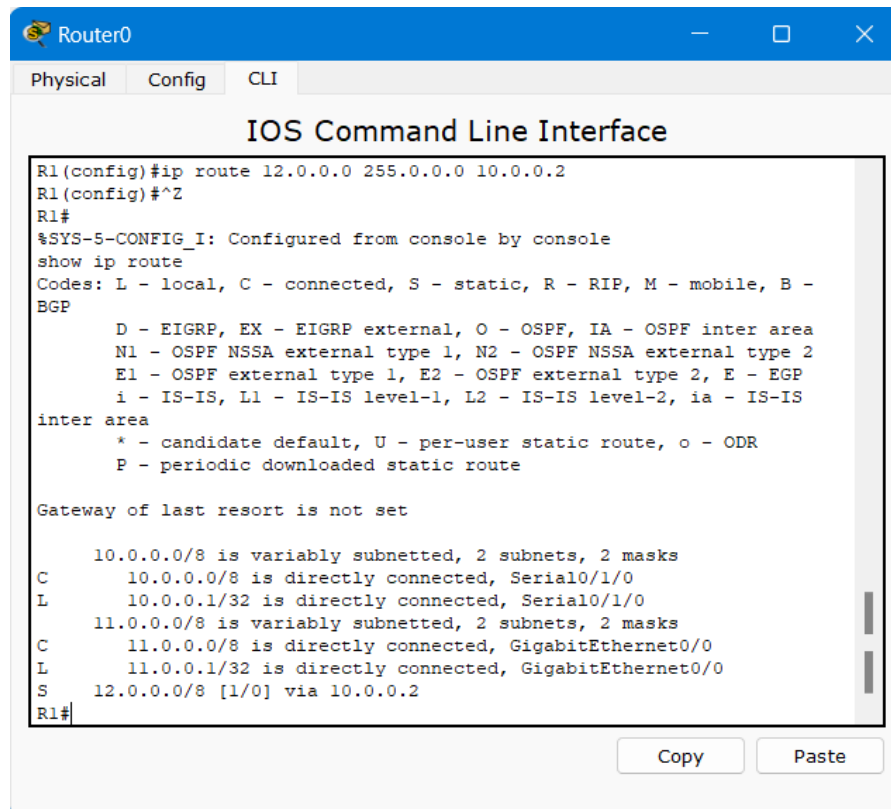
```
#access-list 1 deny 11.0.0.0 0.255.255.255
```

```
#access-list 1 permit any
```

```
#int serial 0/1/0
```

#ip access-group 1 in

#show ip access-list 1



Router0

Physical Config CLI

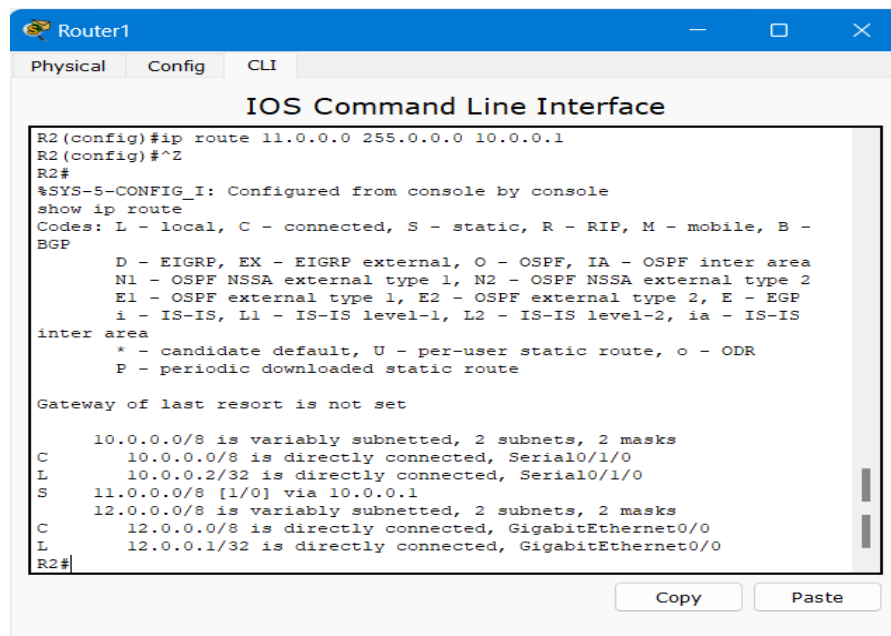
### IOS Command Line Interface

```
R1(config)#ip route 12.0.0.0 255.0.0.0 10.0.0.2
R1(config)#^Z
R1#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/8 is directly connected, Serial0/1/0
L       10.0.0.1/32 is directly connected, Serial0/1/0
      11.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       11.0.0.0/8 is directly connected, GigabitEthernet0/0
L       11.0.0.1/32 is directly connected, GigabitEthernet0/0
S       12.0.0.0/8 [1/0] via 10.0.0.2
R1#
```

Copy Paste



Router1

Physical Config CLI

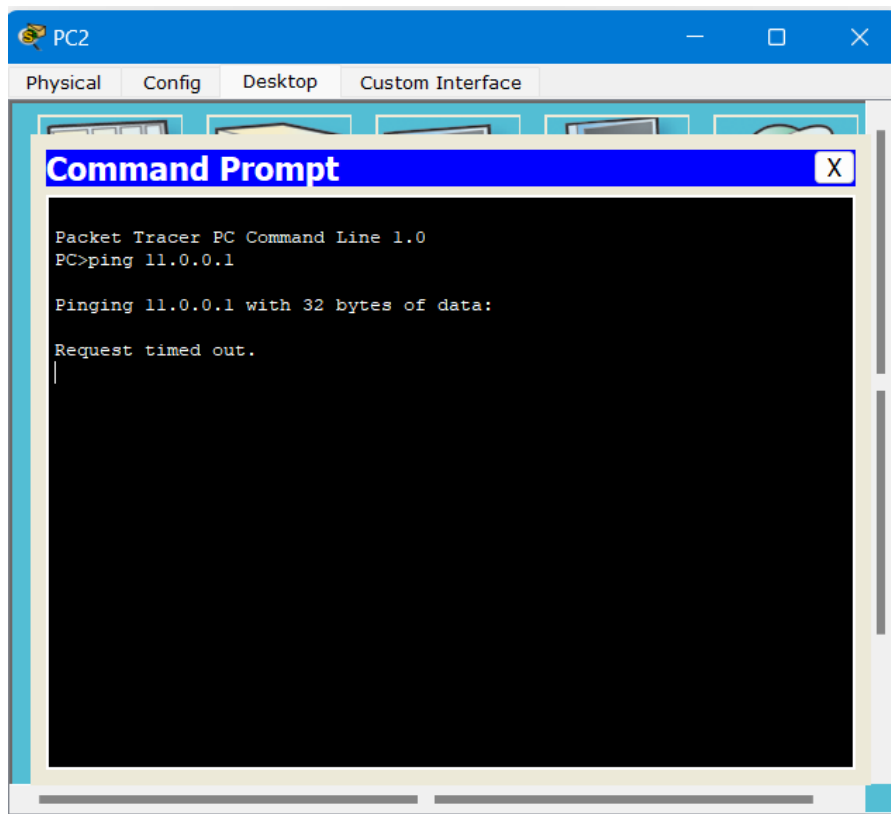
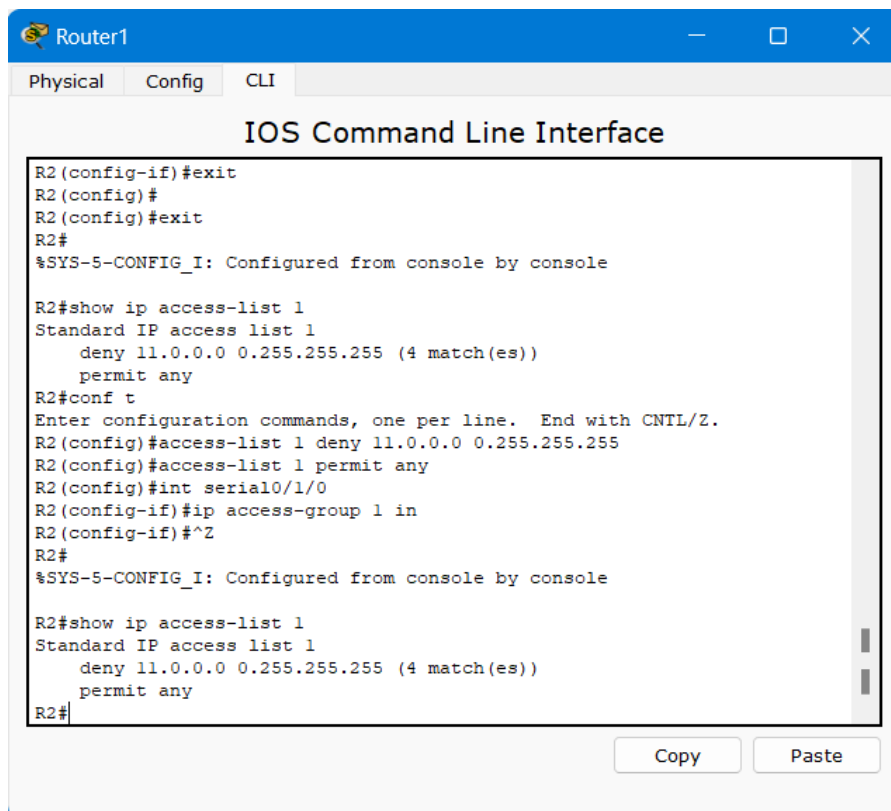
### IOS Command Line Interface

```
R2(config)#ip route 11.0.0.0 255.0.0.0 10.0.0.1
R2(config)#^Z
R2#
%SYS-5-CONFIG_I: Configured from console by console
show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B -
BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
        * - candidate default, U - per-user static route, o - ODR
        P - periodic downloaded static route

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/8 is directly connected, Serial0/1/0
L       10.0.0.2/32 is directly connected, Serial0/1/0
S       11.0.0.0/8 [1/0] via 10.0.0.1
      12.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       12.0.0.0/8 is directly connected, GigabitEthernet0/0
L       12.0.0.1/32 is directly connected, GigabitEthernet0/0
R2#
```

Copy Paste



# Practical 8 : Wireshark Analysis

## What is Wireshark?

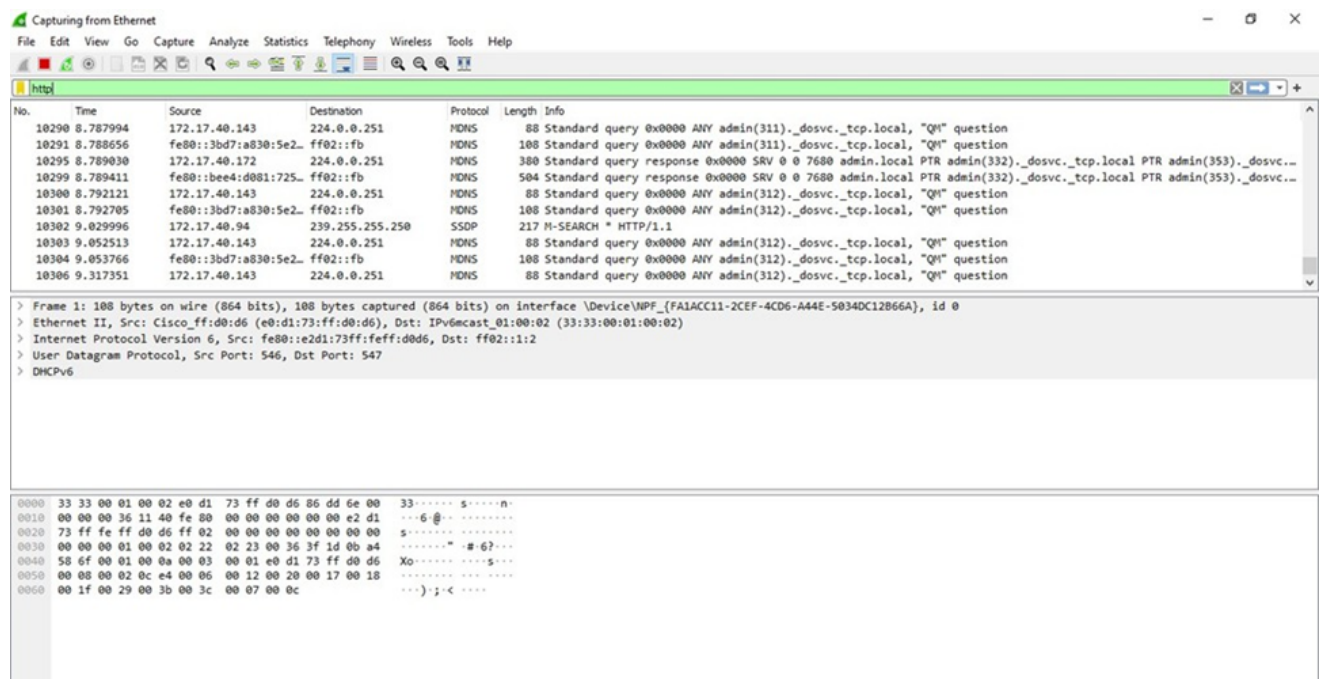
Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting.

It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems.

Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as a free packet sniffer computer application. It puts the network card into an unselective mode, i.e., to accept all the packets which it receives.

## Tracing of different protocols

### HTTP



## SMTP

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

smtp

No.	Time	Source	Destination	Protocol	Length	Info
3055	4.653577	172.17.40.99	224.0.0.251	NDNS	85	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "Q" question
3056	4.655210	fe80::7cc7:53b3:d97...	ff02::fb	NDNS	105	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "Q" question
3057	4.749369	172.17.40.6	172.17.255.255	BROWSER	223	Become Backup Browser
3058	4.759547	172.17.40.130	224.0.0.251	NDNS	355	Standard query response 0x0000 PTR, cache flush admin(316)._dosvc._tcp.local SRV, cache flush 0 0 7680 admin.lo...
3059	4.760294	fe80::239f:5871:2f2...	ff02::fb	NDNS	375	Standard query response 0x0000 PTR, cache flush admin(316)._dosvc._tcp.local SRV, cache flush 0 0 7680 admin.lo...
3060	4.760880	172.17.40.130	224.0.0.251	NDNS	296	Standard query response 0x0000 SRV, cache flush 0 0 7680 admin.local TXT, cache flush A, cache flush 172.17.40...
3061	4.761472	fe80::239f:5871:2f2...	ff02::fb	NDNS	316	Standard query response 0x0000 SRV, cache flush 0 0 7680 admin.local TXT, cache flush A, cache flush 172.17.40...
3065	5.528656	172.17.40.93	172.17.41.255	NDNS	110	Registration NB 16L<00>
3066	5.558332	172.17.40.99	224.0.0.251	NDNS	85	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "Q" question
3067	5.559273	fe80::7cc7:53b3:d97...	ff02::fb	NDNS	105	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "Q" question

> Frame 4: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF\_{FA1ACC11-2CEF-4CD6-A44E-5034DC12B66A}, id 0

> Ethernet II, Src: Micro-St\_8d:10:8f (d8:cb:8a:8d:10:8f), Dst: IPv4mcast\_7f:ff:fa (01:00:5e:7f:ff:fa)

> Internet Protocol Version 4, Src: 172.17.40.7, Dst: 239.255.255.250

> User Datagram Protocol, Src Port: 59387, Dst Port: 1900

> Simple Service Discovery Protocol

```
0000 01 00 5e 7f ff fa d8 cb 8a 8d 10 8f 00 00 45 00  -->E:
0010 00 cb 24 b9 00 00 01 11 d0 56 ac 11 28 07 ef ff  --$-V:
0020 ff fa e7 fb 07 6c 00 b7 e5 8f 4d 2d 53 45 41 52  --1--M-SEAR
0030 43 48 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48  CH * HT P/1.1-H
0040 4f 53 54 3a 20 32 33 39 2e 32 35 35 2e 32 35 35  OST: 239 .255.255
0050 2e 32 35 30 3a 31 39 30 30 0d 0a 4d 41 4e 3a 20  .250:190 0-MAN:
0060 22 73 73 64 70 3a 64 69 73 63 6f 76 65 72 22 0d  "ssdp:discover"
0070 0a 4d 58 3a 20 31 0d 0a 53 54 3a 20 75 72 6e 3a  -MX: 1--ST: urn:
0080 64 69 61 6c 2d 6d 75 6c 74 69 73 63 72 65 65 6e  dial-multiscreen
0090 2d 6f 72 67 3a 73 65 72 76 69 63 65 3a 64 69 61  -org:service:dia
00a0 6c 3a 31 0d 0a 55 53 45 52 2d 41 47 45 4e 54 3a  l:1-USE R-AGENT:
00b0 20 47 6f 6f 67 6c 65 20 43 68 72 6f 6d 65 2f 31  Google Chrome/1
00c0 30 39 2e 30 2e 35 34 31 34 2e 31 32 30 20 57 69  09.0.541.4.120 Wi
```

## TCP

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
7252	27.907147	172.17.40.56	162.159.130.234	TCP	54	52088 → 443 [ACK] Seq=1 Ack=2911 Win=1020 Len=0
7253	27.950197	162.159.130.234	172.17.40.56	TLSv1.2	96	Application Data
7254	27.970676	162.159.130.234	172.17.40.56	TLSv1.2	96	Application Data
7255	27.970741	172.17.40.56	162.159.130.234	TCP	54	52088 → 443 [ACK] Seq=1 Ack=2995 Win=1026 Len=0
7256	28.005471	162.159.130.234	172.17.40.56	TLSv1.2	96	Application Data
7257	28.031098	Zhejiang_bb:74:97	Broadcast	ARP	60	Who has 172.17.40.200? (ARP Probe)
7258	28.031098	Zhejiang_bb:74:97	Broadcast	ARP	60	Who has 172.17.40.200? (ARP Probe)
7259	28.053750	172.17.40.56	162.159.130.234	TCP	54	52088 → 443 [ACK] Seq=1 Ack=3037 Win=1026 Len=0
7260	28.181462	Elitegro_c4:cb:f1	Broadcast	ARP	60	Who has 172.17.40.67? Tell 172.17.40.17
7261	28.683694	fe80::91ed:86c0:585...	ff02::1:2	DHCPv6	163	Solicit XID: 0x36714e CID: 000100011c5bb2d7a41f7267eb9f

> Frame 1: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF\_{FA1ACC11-2CEF-4CD6-A44E-5034DC12B66A}, id 0

> Ethernet II, Src: Dell\_7d:33:08 (a4:1f:72:7d:33:08), Dst: IPv4mcast\_7f:ff:fa (01:00:5e:7f:ff:fa)

> Internet Protocol Version 4, Src: 172.17.40.58, Dst: 239.255.255.250

> User Datagram Protocol, Src Port: 53928, Dst Port: 1900

> Simple Service Discovery Protocol

```
0000 01 00 5e 7f ff fa a4 1f 72 7d 33 08 00 00 45 00  -->E:
0010 00 cb 24 10 00 00 01 11 d0 cc ac 11 28 3a ef ff  --$-V:
0020 ff fa d2 a8 07 6c 00 b7 f5 ac 4d 2d 53 45 41 52  --1--M-SEAR
0030 43 48 20 2a 20 48 54 54 50 2f 31 2e 31 0d 0a 48  CH * HT P/1.1-H
0040 4f 53 54 3a 20 32 33 39 2e 32 35 35 2e 32 35 35  OST: 239 .255.255
0050 2e 32 35 30 3a 31 39 30 30 0d 0a 4d 41 4e 3a 20  .250:190 0-MAN:
0060 22 73 73 64 70 3a 64 69 73 63 6f 76 65 72 22 0d  "ssdp:discover"
0070 0a 4d 58 3a 20 31 0d 0a 53 54 3a 20 75 72 6e 3a  -MX: 1--ST: urn:
0080 64 69 61 6c 2d 6d 75 6c 74 69 73 63 72 65 65 6e  dial-multiscreen
0090 2d 6f 72 67 3a 73 65 72 76 69 63 65 3a 64 69 61  -org:service:dia
00a0 6c 3a 31 0d 0a 55 53 45 52 2d 41 47 45 4e 54 3a  l:1-USE R-AGENT:
00b0 20 47 6f 6f 67 6c 65 20 43 68 72 6f 6d 65 2f 31  Google Chrome/1
00c0 31 30 2e 30 2e 35 34 38 31 2e 31 37 37 20 57 69  10.0.548.1.177 Wi
```



## UDP

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp

No.	Time	Source	Destination	Protocol	Length	Info
7880	4.922870	fe80::db48:f7b:9889_	ff02::fb	NDNS	107	Standard query 0x0000 ANY 16L-(138)._dosvc._tcp.local, "QI" question
7889	4.924053	fe80::20d7:a052:595_	ff02::fb	NDNS	319	Standard query response 0x0000 SRV 0 0 7680 16L-.local PTR 16L-(156)._dosvc._tcp.local PTR 16L-(162)._dosvc._tc
7898	4.925018	172.17.40.96	224.0.0.251	NDNS	60	Standard query response 0x0000 SRV 0 0 7680 16L-.local PTR 16L-(156)._dosvc._tcp.local PTR 16L-(162)._dosvc._tc
7899	4.926802	172.17.40.93	224.0.0.251	NDNS	87	Standard query 0x0000 ANY 16L-(139)._dosvc._tcp.local, "QI" question
7900	4.927118	fe80::db48:f7b:9889_	ff02::fb	NDNS	107	Standard query 0x0000 ANY 16L-(139)._dosvc._tcp.local, "QI" question
7904	4.927118	172.17.40.93	224.0.0.251	NDNS	953	Standard query response 0x0000 SRV 0 0 7680 16L-.local PTR 16L-(159)._dosvc._tcp.local PTR 16L-(170)._dosvc._tc
7908	4.927516	fe80::db48:f7b:9889_	ff02::fb	NDNS	1077	Standard query response 0x0000 SRV 0 0 7680 16L-.local PTR 16L-(159)._dosvc._tcp.local PTR 16L-(170)._dosvc._tc
7909	4.928494	172.17.40.93	224.0.0.251	NDNS	87	Standard query 0x0000 ANY 16L-(140)._dosvc._tcp.local, "QI" question
7910	4.928801	fe80::db48:f7b:9889_	ff02::fb	NDNS	107	Standard query 0x0000 ANY 16L-(140)._dosvc._tcp.local, "QI" question
7914	4.929731	172.17.40.94	224.0.0.251	NDNS	455	Standard query response 0x0000 SRV 0 0 7680 16L-.local PTR 16L-(161)._dosvc._tcp.local PTR 16L-(183)._dosvc._tc

> Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF\_{FALACC11-2CEF-4CD6-A44E-5034DC12B66A}, id 0

> Ethernet II, Src: HP\_b3:eb:1b (c0:18:03:b3:eb:1b), Dst: IPv4mcast\_fb (01:00:5e:00:00:fb)

> Internet Protocol Version 4, Src: 172.17.40.103, Dst: 224.0.0.251

> User Datagram Protocol, Src Port: 5353, Dst Port: 5353

> Multicast Domain Name System (query)

```

0000  01 00 5e 00 00 fb c0 18 03 b3 eb 1b 00 00 45 00  ..^.....E:
0010  00 48 89 2d 00 00 01 11 7b 04 ac 11 28 67 e0 00  .H.....{...g..
0020  00 fb 14 e9 14 e9 00 34 ea 85 00 00 00 00 00 01  .....4.....
0030  00 00 00 00 00 00 08 31 36 4c 2d 28 36 39 29 06  .....1 6L-(69)
0040  5f 64 6f 73 76 63 04 5f 74 63 70 05 6c 6f 63 61  _dosvc._tcp:loca
0050  6c 00 00 ff 00 01 1.....

```

## Practical 9 : Socket Programming

### MyServer.java

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

class MyServer {
    public static void main(String[] args) {
        try {

            Scanner sc = new Scanner(System.in);

            ServerSocket ss = new ServerSocket(6666);
            Socket s = ss.accept();// establishes connection

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

            String str = (String) dis.readUTF();
            System.out.println("message= " + str);

            Thread t = new Thread();
            t.sleep(4000);

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

            System.out.println("Enter message");
            String s1 = new String();
            s1 = sc.nextLine();
            dout.writeUTF(s1);
            dout.flush();
            dout.close();
            ss.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

## **MyClient.java**

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

class MyClient {
    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            Socket s = new Socket("localhost", 6666);
            DataOutputStream dout = new DataOutputStream(s.getOutputStream());
            System.out.println("Enter message");
            String s1 = new String();
            s1 = sc.nextLine();
            dout.writeUTF(s1);
            Thread t = new Thread();
            t.sleep(5000);
            dout.flush();

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

            String str = (String) dis.readUTF();
            System.out.println("message= " + str);
            dout.close();
            s.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
C:\Users\ninad\Downloads>java MyClient
Enter message
Hi Ninad
```

```
C:\Users\ninad\Downloads>java MyServer
message= Hi Ninad
Enter message
Bye Ninad
```

```
C:\Users\ninad\Downloads>java MyClient
Enter message
Hi Ninad
message= Bye Ninad
```