Data Communications and Systems Lab File

Name: Kshitij Chandrakar
SAP: 500124827
Batch: 5
Submitted To: Richa Kumari

	Date	Content	Page. No	Attendance	Signature
22/01/25	30/1/25	Experiment 1	1 - 5	Present	
29/01/25	30/1/25	Experiment 2	6 - 8	Present	
3/02/25	3/2/25	Experiment 3	9 - 14	~~Present~~ Absent.	
5/02/25	6/2/25	Experiment 4	15 - 18	Present	
5/02/25	6/2/25	Experiment 5	19 - 22	Present	
19/02/25	20/2/25	Experiment 6	23 - 30	Present	
12/03/25	13/3/25	Experiment 7	31 - 37	Present	
12/03/25	20/3/25	Experiment 8	38 - 39	Present	
26/03/25	3/4/25	Experiment 9	40 - 53	Present	
02/04/25	27/3/25	Experiment 10	55 - 62	Present	
09/4/25	10/4/25	Experiment 11	63 - 77	~~Present~~ Absent	
16/4/25	17/4/25	Experiment 12	78 - 87	Present	
23/4/25	24/4/25	Class Test and Viva	88 - 91	Present	

Rich 30/4/25

Progress — 12/14

Page No:

Assignment - 1

Q1 5-Networking Devices :-

1. Routers :- it is used to connect multiple networks and Route packets Between them. Provides Security features like firewalls, NAT, and VPN Support.

2. Switch :- Used to connect multiple Devices in a LAN; Uses MAC Addresses to forward data frames and Reduces network congestion by Sending data only to the Intended Recipient. It Supports VLANs.

3. Modem :- Converts Digital Signals from a computer into analog Signals for transmission over telephone lines. Connects to an ISP to provide Internet Access Integrated with Routers in modern Setups.

4. Wireless Access point :- Allows wireless Devices to connect to a wired network via wifi, Extends Network coverage without more cabling.

5. firewalls :- Monitors and controls Incoming/outgoing Network traffic based on Security Rules. Blocks Unauthorised access while permitting legitimate communication and protects against Cyber Attacks.

Page No :

Q8:- Difference Between a Switch and Hub.

	Hub	Switch
1.	Works on the Physical Layer	Works on Data-Link Layer.
2.	Broadcasts Data to all connected Devices	forwards only to Intended Recipient
3.	Slower Due to collisions and causes Congestion	faster Due to No collisions and full Duplex connection
4.	All Devices can see Traffic and is Therefore, less Secure	Isolation of Traffic leads to more Security.

Page No :

Q2 A. Difference Between Routes and Modem

	Modem	Routes.
1.	Connects to ISP and converts Signal	Routes Data Between Devices and Network and Internet
2.	Physical layer	Network layer.
3.	Converts Digital Signals to Analog and vice vesca	Works with Digital Data only
4.	Gets a public IP from ISP	Assigns private IP Addresses and used NAT to Share a single public IP
5.	Doesn't include Security features	Includes firewall, VPN, Port forwarding and Access control.
6	Doesn't include ~~WLAN~~ Wifi.	Includes Wifi

B. NIC :- a Network Interface Card allows a computer to connect to a Network. its Primary Role is to facilitate communication between the computer and the Network Devices, it Assigns a MAC Address, and Supports Network protocols.

Page No :

Q3 A. How to connect a computer to a Switch Using an Ethernet Cable.

1. Gather the Hardware
   - Ethernet cable
   - Switch
   - computer.
2. Power on the Switch Using its Adapter
3. Plug in one End of the Ethernet cable into the computer Ethernet port
4. Plug other End into an Available port on the Switch
5. Check Status Light on the Switch to check its Status.

B. Configure IP on ~~Linux~~ windows
Use the following to check IP

PS> ipconfig

Configure IP

PS> netsh interface ip set address name="Ethernet" static 192.168.1.100 255.255.255.0 192.168.1.1

192.168.1.100 — Set IP
255.255.255.0 — Subnet Mask
192.168.1.1 — Default Gateway

Q5 A. Modem - Connects to the ISP and converts into Digital connection

Switch :- connects Multiple devices together

WAP :- Provides Wifi for Portable Devices

Firewall :- filters Incoming and Outgoing traffic.

VPN Server :- Encrypts Remote connections.

B. What is a firewall?

A firewall acts as a Digital Barrier Between a trusted Internal Network and External Connections. It does the following

1. Blocks unauthorized Access
2. filters Malicious traffic
3. Prevents Data Breaches
4. Controls Network Access

How Does It Work?

It uses various techniques like

1. Packet filtering
2. Stateful Inspection
3. Application - level Inspection
4. Next Generation firewall like deep packet Inspection, Intrution prevention

Page No :

Assignment-2

Q1 What is Bit Stuffing?

Bit Stuffing is used in Data communication protocols to ensure Reliable transmission by inserting extra bits into the Data Stream, it helps the recievers Distinguish Between Data and control Signals.

Why is it Neccessary?

1. Prevents frame delimiter conflicts
2. Ensures Synchronization
3. Avoids Control Signal Ambiguity
4. Improves Error Detection.

Q2 Difference Between Bit Stuffing and Byte Stuffing.

1. Bit Stuffing :- Used in Bit Oriented protocols
   → Inserts an Extra bit to prevent confusion with Reserved Bit Patterns

   for Ex :- Using HDLC Protocol
   Original Data - 01111101
   Stuffed Data - 011111001

Page No :

Q2 Byte Stuffing :- Inserts an Extra Byte before reserved bytes like ESC, STX etc and is used in Byte oriented Protocols.

for Example :-

Original Data :- Flag 7E

~~Flag~~ Stuffed Data :- ESC 5D

Q3 Given Data

01111110 1101 1111 1101 1110

Stuffed Data

[Insert 0 after 5 '1's

01111101 0 10110111110 0 11011101 0

Stuffed Bits

Destuffed Data :-

01111110 11011111 11011101 0

Page No :

~~Assignment 3~~

Q4 Advantages and Disadvantages of Bit Stuffing.

Advantages

1. Prevents Delimiter Conflicts
2. Simple Hardware Implementation
3. Low overhead
4. Works in continuous Bit Stream
5. Improves Synchronization

Disadvantages

1. Adds small overhead because transmission time is slightly increased
2. Not Human Readable
3. Limited to Bit Oriented Protocols
4. Inefficient for Random Data

Page No.

Assignment-3

Q1 Compute CRC for 1110 001 with polynomial 1001

Polynomial 1001 Represents x^3+1, which is degree 3. Therefore padded Bits = 3

Padded Message = 111 001 000

Binary Division

```
1001 | 1110 0100 0 |
     ⊕1001
     -----
      0111 0
       1001
       ----
       0111 1
        1001
        ----
        0110 0
         1001
         ----
         0101 0
          1001
          ----
          0011
```

Remainder = 011

∴ CRC = 011

Transmitted Message = 111 001 011

Page No:

Q2 Given Generator Polynomial = 111
Message = 1001101

find CRC and transmitted Message.

Generator polynomial = 111 Represents x^2+x+1 of degree 2
∴ CRC length = 2

∴ Padded Message = 100 110 100

Binary Division

```
111 | 100  110  100
      111
      ---
      011 1
      111
      ---
      000 101
          111
          ---
          010 0
            11
            --
            011        Remainder = 011
```

CRC = 11

Transmitted Message = 100 110 111

Q3

Given Message = 101 001

Polynomial = 11011

Recieved Message = 101 001 1001

find CRC and verify Recieved Message.

Polynomial is 11011 which is of Degree 4

∴ CRC bits = 4

Padded Message = 101 001 00 00

Division

```
11011 | 101 001 0000
        1010 11
        -------
        011 11
        11011
        -------
        10100
        11011
        -------
        01111
        11011
        -------
        10100
        11011
        -------
        01110
```

Remainder = 1010

∴ CRC = 1010

Transmitted Message = 101 001 1010

Verification of Recieved Message

Division

```
11011 | 101001 1001
        11011
        -------
        01111
        11011
        -------
        10100
        11011
        -------
        01111
        11011
        -------
        10100
        11011
        -------
        0111
```

Remainder = 0111

∴ Remainder is Non zero

Error is Detected.

Page No :

Q4 Data = 1011
Generate Hamming code

Bit position	1	2	3	4	5	6	7
Type	P_1	P_2	D	P_4	D	D	D
Value			1		0	1	1

$P_1 = P_1 \oplus 1 \oplus 0 \oplus 1 = 0$

$\therefore P_1 = 0$

$P_2 = P_2 \oplus 1 \oplus 1 \oplus 1 = 0$

$\therefore P_2 = 1$

$P_4 = P_4 \oplus 0 \oplus 1 \oplus 1 = 0$

$P_4 = 0$

$\therefore$ Transmitted Message is
0 1 1 0 0 1 1

Q3 Recieved Message 1011101

Bit position	1	2	3	4	5	6	7
Type	P_1	P_2	D	P_4	D	D	D
Recieved	1	0	1	1	1	0	1

To verify Errors and Even parity

$P_1 = P_1 \oplus D_3 \oplus D_5 \oplus D_7 = 0$

→ No Errors

$P_2 = P_2 \oplus D_3 \oplus D_6 \oplus D_7 = 0$

→ No Error

$P_4 = P_4 \oplus D_5 \oplus P_6 \oplus D_7 = 1$

→ Errors Detected

Error position is $P_4 P_2 P_1$ in Binary

∴ Error at $(100)_2$ at Position 4

∴ correct Message by flipping bit 4

Correct Message = 1010101

Q6 Given Data 11001

Using Even Parity

Bit Position:	1	2	3	4	5	6	7	8	9
Type:	P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9
Data:			1		1	0	0		1

$P_1 = P_1 \oplus D_3 \oplus D_5 \oplus D_7 = 0$

$\therefore P_1 = 1$

$P_2 = P_2 \oplus D_3 \oplus D_4 \oplus D_5 = 0$

$P_2 = 1$

$P_4 = D_2 \oplus D_3 \oplus D_4 = 0$

$P_4 = 0$

$P_8 = D_5$

$P_8 = 1$

∴ Transmitted Message is

11100 1101

Assignment 4

Theory

1. **IP Addressing Basics** IPv4 and IPv6 Addressing

- IPv4 (Internet Protocol Version 4): ◦ Uses a 32-bit addressing scheme ◦ allows around 4.3 billion unique addresses ◦ written in decimal format, separated by dots (For Ex: 192.168.1.1)
- IPv6 (Internet Protocol Version 6): ◦ Uses a 128-bit addressing scheme ◦ written in hexadecimal format, separated by colons (For Ex: 1011:09pa:00a2:0000:0000:9fe3:9102:4332). Public vs. Private IP Addresses
- Public IP Addresses: ◦ Routable on the internet and assigned by ISPs (For Ex 8.8.8.8).
- Private IP Addresses: ◦ Used within local networks and not routable on the internet. Their Address Ranges are: Class A: 10.0.0.0 – 10.255.255.255 Class B: 172.16.0.0 – 172.31.255.255 Class C: 192.168.0.0 – 192.168.255.255 Subnet Masks
- A subnet mask defines the network and host portions of an IP address.
- For Ex: 255.255.0.0 indicates that the first two octets represent the network1 of 5 portion, and the last two octet represents hosts.

2. **Subnetting** CIDR Notation and Its Role in Subnetting:

- CIDR: Classless Inter-Domain Routing
- A method to allocate IP addresses efficiently by eliminating the rigid class-based addressing system.
- Expressed using a suffix (For Ex: 192.168.1.0/24 means 24 bits for the network, 8 bits for hosts).
- Subnetting allows networks to be divided into smaller subnetworks, improving security and reducing IP wastage.
- Example: Splitting 192.168.1.0/24 into four subnets: ◦ 192.168.1.0/26 ◦ 192.168.1.64/26 ◦ 192.168.1.128/26 ◦ 192.168.1.192/26

3. **Supernetting** Definition and Advantages:

- Supernetting: The process of combining multiple smaller networks into a larger one by modifying the subnet mask.
- Reduces the number of routing table entries, improving efficiency.
- Helps ISPs aggregate multiple customer networks. Real-World Applications:
- Used in ISP route aggregation to reduce the number of advertised routes.
- Helps in enterprise network management by consolidating multiple subnets.
- Example: Combining four /26 networks (192.168.1.0/26, 192.168.1.64/26, 192.168.1.128/26, 192.168.1.192/26) into a single /24 network (192.168.1.0/24). ## Practical

1. **Subnetting a Corporate Network** Departments:
    1. HR: 50 Hosts
    2. IT: 100 Hosts
    3. Finance: 30 Hosts Subnets Sizes Subnets Are Sized by Powers of 2. Therefore nearest power of 2 for each department. Subnets are determined by the formula: Subnet = 32 - Number of Bits
    4. HR: 64 [6 Bits, /26 Subnet]
    5. IT: 128 [7 Bits, /25 Subnet]
    6. Finance: 32 [5 Bits, /27 Subnet] Subnet Assignment: (Allocating Sequentially We Have)
2. IT (/25)

- Network Address: 192.168.1.0/25
- Broadcast Address: 192.168.1.127
- Subnet Mask: 255.255.255.128
- Usable IPs: 192.168.1.1 – 192.168.1.126

2. HR (/26)

- Network Address: 192.168.1.128/26
- Broadcast Address: 192.168.1.191
- Subnet Mask: 255.255.255.192
- Usable IPs: 192.168.1.129 – 192.168.1.190

3. Finance (/27)

- Network Address: 192.168.1.192/27
- Broadcast Address: 192.168.1.223
- Subnet Mask: 255.255.255.224
- Usable IPs: 192.168.1.193 – 192.168.1.222

2. Subnetting an ISP Network Supernetting combines multiple contiguous networks into a single network. This reduces the number of routing table entries, Thus improving efficiency. Given Networks:

- 192.168.10.0/24
- 192.168.11.0/24
- 192.168.12.0/24
- 192.168.13.0/24 Finding Common Bits in the Network Address Since the subnet is /24, the third octet changes and is relevant. Thus, we find common bits in the third octet. Given Values in 8 Bits:
- 10 : 00001010
- 11 : 00001011
- 12 : 00001100
- 13 : 00001101 We can See here, that only the last 6 bits remain constant. thus, to include all 4 networks we need a /22 Mask. (16 + 6 = 22) 16 for the first 2 octets, 6 for the 3rd one New Network addresses Since the new Subnet mask is /22, The Network Address 192.168.10.0/22 would cover all the four networks. Advantages

- For the router, it reduces the number of entries in the routing table. (from 4 /24 Entries to 1 /22)
- The lower entry count improves efficiency and simplifies the network Management for Larger Networks.

3. Subnetting a University Network Departments:
    1. Engineering: 2,000 hosts
    2. Medical: 1,500 hosts
    3. Management: 1,000 hosts
    4. Library: 500 hosts
    5. Admin: 300 hosts Subnets Sizes Subnets Are Sized by Powers of 2. Therefore nearest power of 2 for each department. Subnets are determined by the formula: Number of Bits are the nearest larger power of 2. Subnet = 32 - Number of Bits
4. Engineering: 2048 [11 Bits, /21 Subnet]
5. Medical: 2048 [11 Bits, /21 Subnet]
6. Management: 1024 [10 Bits, /22 Subnet]
7. Library: 512 [9 Bits, /23 Subnet]
8. Admin: 512 [9 Bits, /23 Subnet] Subnet Assignment: (Allocating Sequentially We Have)
9. Engineering (/21)

- Network Address: 172.16.0.0/21
- Broadcast Address: 172.16.7.255
- Usable IPs: 172.16.0.1 – 172.16.7.254
- Subnet Mask: 255.255.248.0

2. Medical (/21)

- Network Address: 172.16.8.0/21
- Broadcast Address: 172.16.15.255
- Usable IPs: 172.16.8.1 – 172.16.15.254
- Subnet Mask: 255.255.248.0

3. Management (/22)

- Network Address: 172.16.16.0/22
- Broadcast Address: 172.16.19.255
- Usable IPs: 172.16.16.1 – 172.16.19.254
- Subnet Mask: 255.255.252.0

4. Library (/23)

- Network Address: 172.16.20.0/23
- Broadcast Address: 172.16.21.255
- Usable IPs: 172.16.20.1 – 172.16.21.254
- Subnet Mask: 255.255.254.0

5. Admin (/23)

- Network Address: 172.16.22.0/23

- Broadcast Address: 172.16.23.255
- Usable IPs: 172.16.22.1 – 172.16.23.254
- Subnet Mask: 255.255.254.0

Assignment 5

SAP: 500124827
Name: Kshitij Chandrakar
Batch: 5

---

ifconfig: Deprecated, would not run on my machine but displays information about the connected networks. **ip addr:** Displays information about the connected networks.

```
asus@debian-asusrog:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qle
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eno2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group
    link/ether 04:d4:c4:eb:74:97 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group defa
    link/ether 28:7f:cf:e5:6b:e5 brd ff:ff:ff:ff:ff:ff
    altname wlp0s20f3
    inet 192.168.0.102/24 brd 192.168.0.255 scope global dynamic noprefixroute wlo1
       valid_lft 6538sec preferred_lft 6538sec
    inet6 fe80::4d56:ece7:7abb:abe4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: CloudflareWARP: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1280 qdisc mq state
    link/none
    inet 172.16.0.2/32 scope global CloudflareWARP
       valid_lft forever preferred_lft forever
    inet6 2606:4700:110:8865:6078:763e:be97:f2f3/128 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::d28f:8bef:b8ff:28f1/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

hostname: displays the hostname of the machine.

```
asus@debian-asusrog:~$ hostname
debian-asusrog
```

ping: pings the given address

```
asus@debian-asusrog:~$ ping google.com
PING google.com(bom12s08-in-x0e.1e100.net (2404:6800:4009:813::200e)) 56 data bytes
64 bytes from bom12s08-in-x0e.1e100.net (2404:6800:4009:813::200e): icmp_seq=1 ttl=59
64 bytes from bom12s08-in-x0e.1e100.net (2404:6800:4009:813::200e): icmp_seq=2 ttl=59
64 bytes from bom12s08-in-x0e.1e100.net (2404:6800:4009:813::200e): icmp_seq=3 ttl=59
64 bytes from bom12s08-in-x0e.1e100.net (2404:6800:4009:813::200e): icmp_seq=4 ttl=59
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 49.602/107.369/167.864/49.303 ms
```

ip route: Shows the route a given packet takes.

```
asus@debian-asusrog:~$ ip route
default via 192.168.0.1 dev wlo1 proto dhcp src 192.168.0.102 metric 600
169.254.0.0/16 dev wlo1 scope link metric 1000
192.168.0.0/24 dev wlo1 proto kernel scope link src 192.168.0.102 metric 600
```

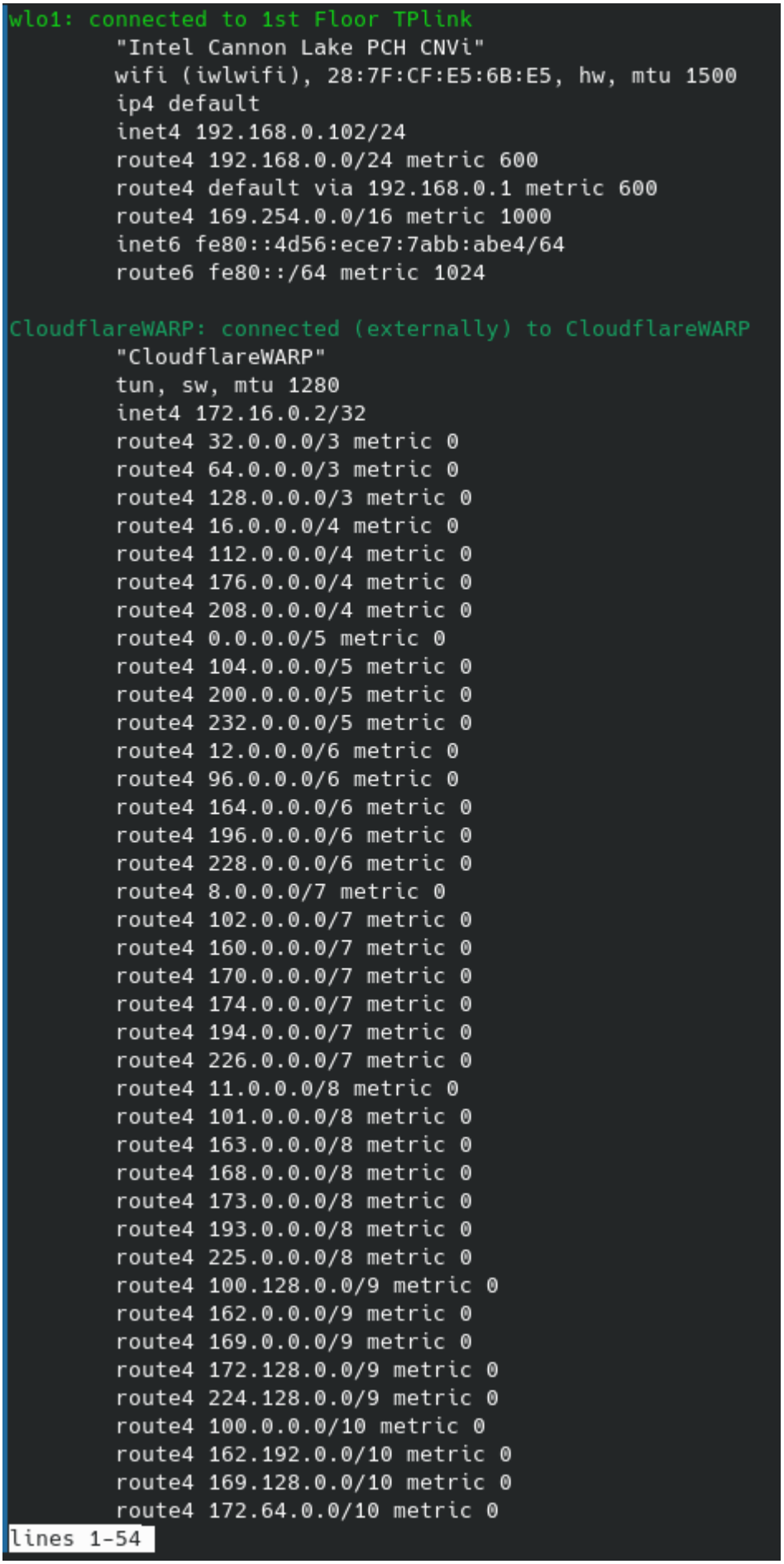
netstat: Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

```
asus@debian-asusrog ~ % netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 debian-asusrog:38442    lb-140-82-112-25-:https ESTABLISHED
tcp        0      0 debian-asusrog:51868    ec2-52-89-11-84.u:https ESTABLISHED
tcp        0     25 debian-asusrog:48074    162.159.192.1:https     LAST_ACK
tcp        0      0 debian-asusrog:35158    server-18-67-195-:https ESTABLISHED
tcp        0      0 debian-asusrog:59318    93.243.107.34.bc.:https TIME_WAIT
tcp        0      0 debian-asusrog:54006    140.227.186.35.bc:https ESTABLISHED
tcp        0      0 debian-asusrog:45268    a23-38-59-250.depl:http ESTABLISHED
```

traceroute: Shows the route a given packet takes.

```
asus@debian-asusrog:~$ traceroute google.com
traceroute to google.com (142.250.192.110), 30 hops max, 60 byte packets

 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

nmcli: cli tool to manage the network.



dhclient: a tool to manage network interfaces using DHCP.

```
asus@debian-asusrog:~$ sudo dhclient
RTNETLINK answers: File exists
asus@debian-asusrog:~$
```

ssh: used to ssh into other machines using the ssh protocol

```
asus@debian-asusrog:~$ ssh
usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command [argument ...]]
asus@debian-asusrog:~$
```

scp: Secure File Copy over SSH in a given network.

```
asus@debian-asusrog:~$ scp
usage: scp [-346ABCOpqRrsTv] [-c cipher] [-D sftp_server_path] [-F ssh_config]
           [-i identity_file] [-J destination] [-l limit] [-o ssh_option]
           [-P port] [-S program] [-X sftp_option] source ... target
```

EXPERIMENT - 6

Theoretical Understanding

1) Define Network Topology.

→ Network topology refers to the physical or logical arrangement of network devices and connections. It defines how computers, cables, switches, and other devices are interconnected to facilitate communication and data exchange.

2) Advantages & disadvantages of -

- **Bus Topology**

Advantages -

- Cost-effective as it requires less cabling.
- Easy to set up and extend.
- Works well for small networks.

Disadvantages -

- A single point of failure (main cable) can bring down the entire network.
- Performance degrades as more devices are added.
- Data collisions can occur, reducing efficiency.

- **Ring topology**

Advantages -

- No data collisions due to the unidirectional or bidirectional data flow.
- Predictable data transmission due to sequential passing of data.

Disadvantages -

- Failure of a single node or cable can disrupt the entire network.
- Troubleshooting is difficult due to the dependency on all devices in the ring.
- Adding or removing devices requires temporarily shutting down the network.

- Star topology

Advantages -

- High fault tolerance; failure of one device does not affect the entire network.
- Easy to manage, scale, and troubleshoot.
- Centralized control provides better performance.

- Disadvantages -

- Expensive due to the requirement of a central switch or hub.
- If the central hub/switch fails, the entire network is affected.

- Mesh topology

Advantages -

- High redundancy; failure of one connection does not disrupt communication

- Offers excellent reliability & security.
- Data transmission is faster due to multiple paths.

Disadvantages-

- Expensive due to expensive cabling and hardware requirements.
- Complex installation and maintenance.

3) Which topology is most fault-tolerant and why?

→ Mesh topology is the most fault-tolerant because it has multiple redundant paths for data transmission. Even if one link or device fails, alternative paths ensure uninterrupted communication.

4) Which topology is cost-effective & why?

→ Bus topology is the most cost-effective because it requires minimal cabling and no expensive central devices, making it suitable for small network

5) Compare Mesh and star Topology in terms of scalability & reliability.

→ Scalability -

Mesh topology is difficult to scale due to high wiring and configuration complexity.

Star topology is easier to scale by adding more devices to the central switch.

Reliability - Mesh topology

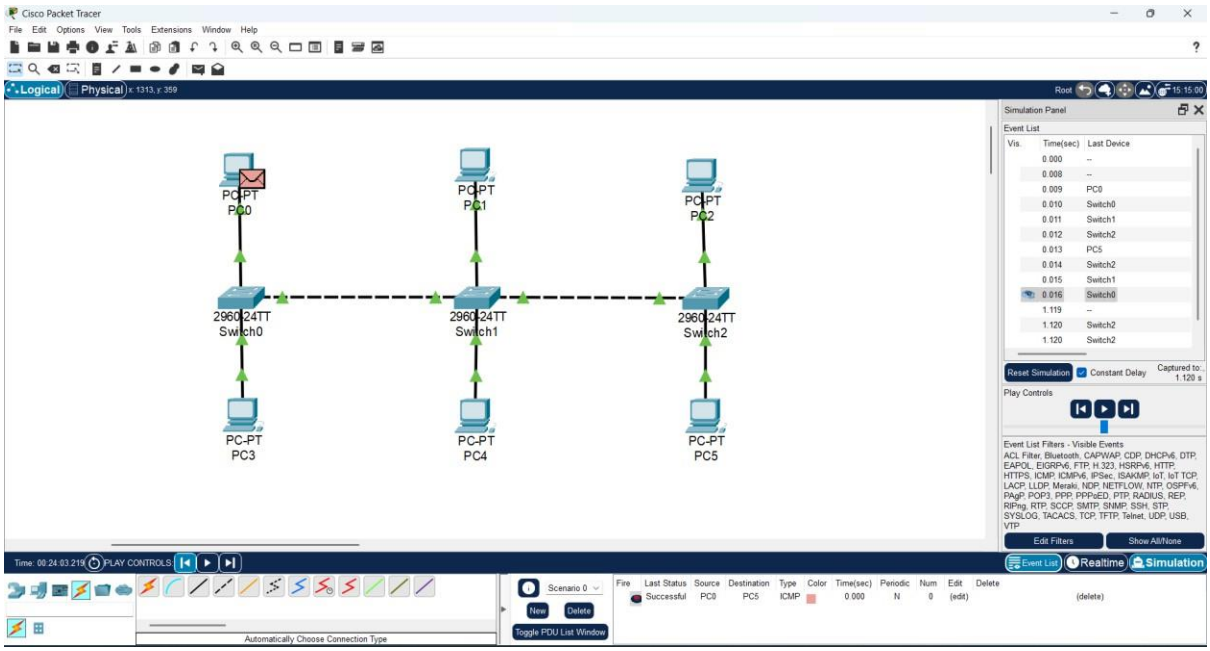
Highly reliable due to multiple redundant paths.

Star topology is also reliable but dependent on the central hub/switch, which is a single point of failure.

Task 2: Network Topology Design in Cisco Packet Tracer

Design and simulate the following network topologies using Cisco Packet Tracer:

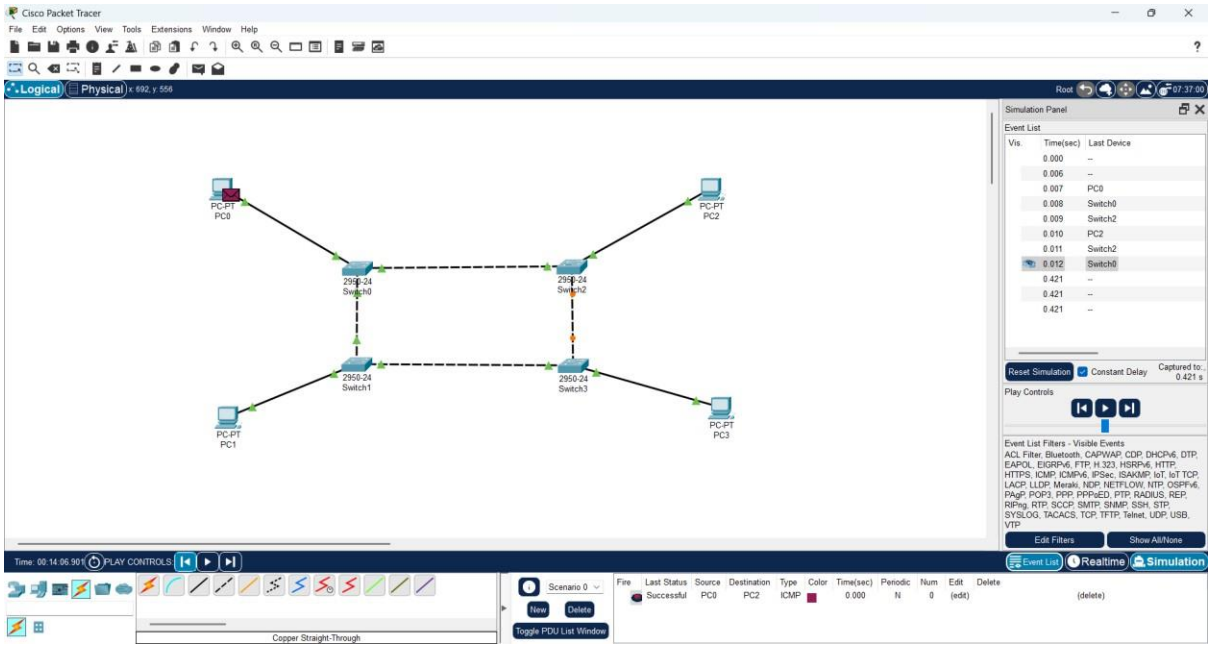
- Bus Topology (Using a single backbone cable)



Data Transmission Flow:

- Data travels along the backbone in both directions.
- Only the intended recipient processes the data.
- If two devices send data simultaneously, a collision occurs, requiring a retransmission.

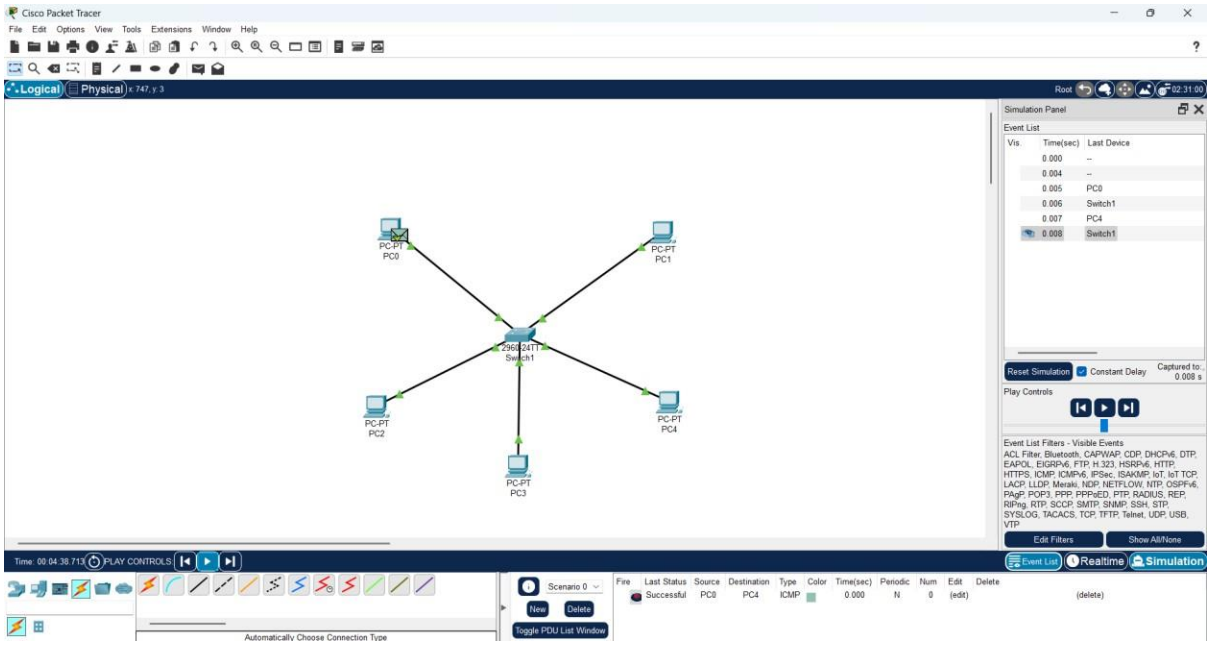
- Ring Topology (Connecting all devices in a loop)



Data Transmission Flow:

- Data moves in one direction (unidirectional) or both directions (bidirectional in modern implementations).
- A token-passing mechanism is used to prevent collisions.

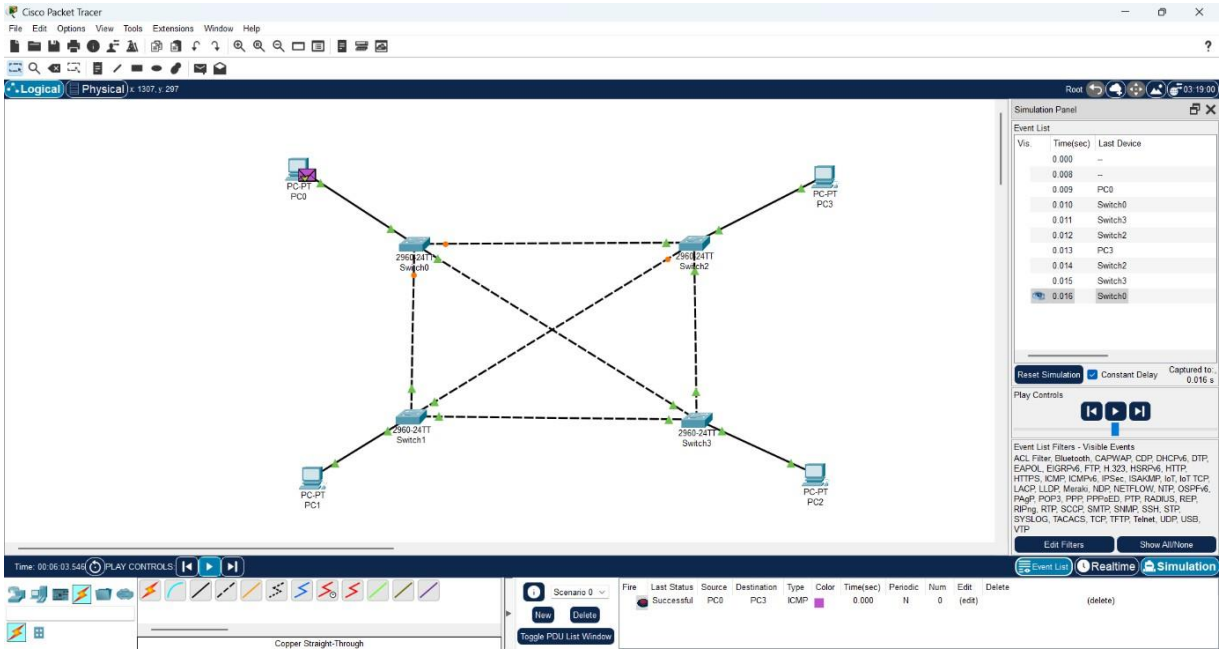
- Star Topology (Using a central switch or hub)



Data Transmission Flow:

- Data travels from the sender to the switch/hub, then forwarded to the recipient.
- Switches use MAC addresses to forward packets, while hubs broadcast data to all devices.

- Mesh Topology (Full or partial, as per requirement)



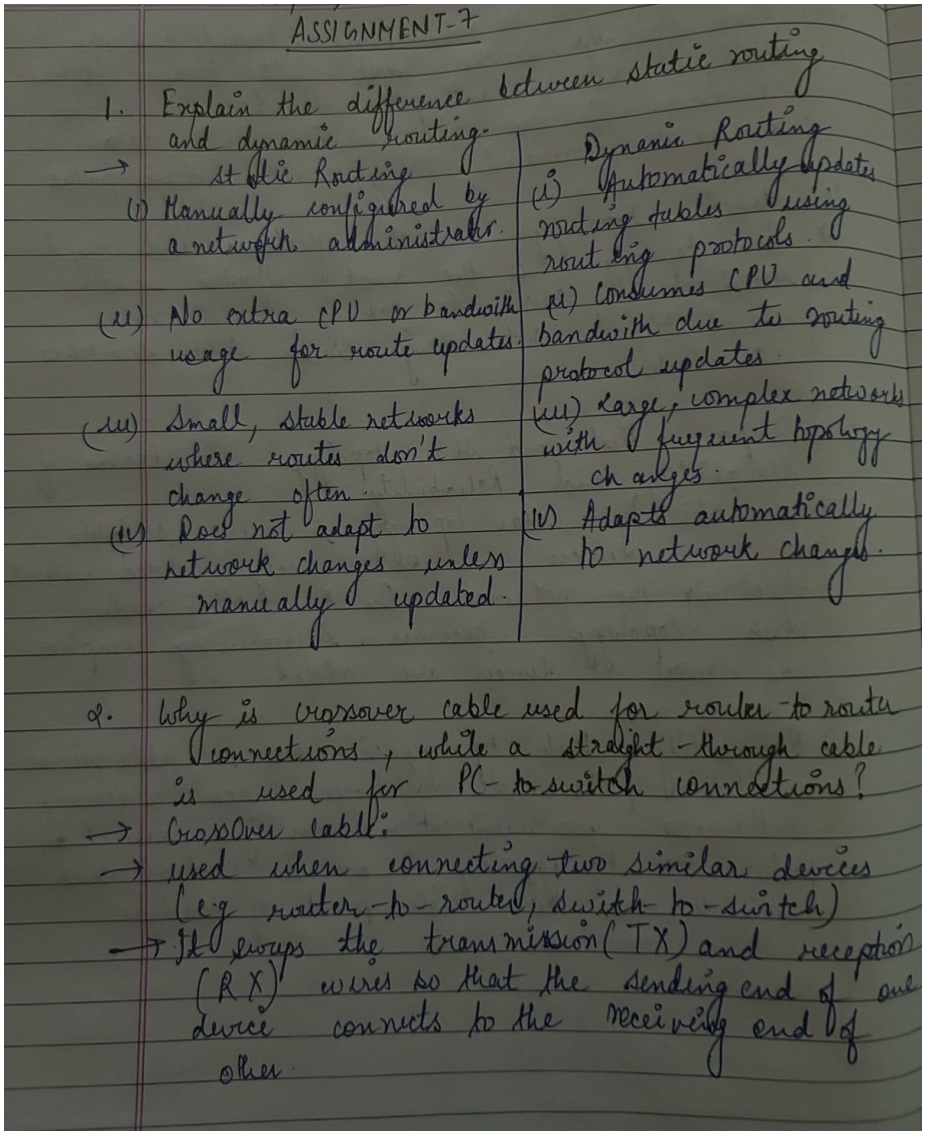
Data Transmission Flow:

- Data follows multiple paths, ensuring redundancy and fault tolerance.
- Routing protocols like OSPF or RIP can be used for dynamic path selection.

Assignment 7

Name: Kshitij Chandrakar SAP: 500124827

Theory Section



ASSIGNMENT-7

1. Explain the difference between static routing and dynamic routing.

→

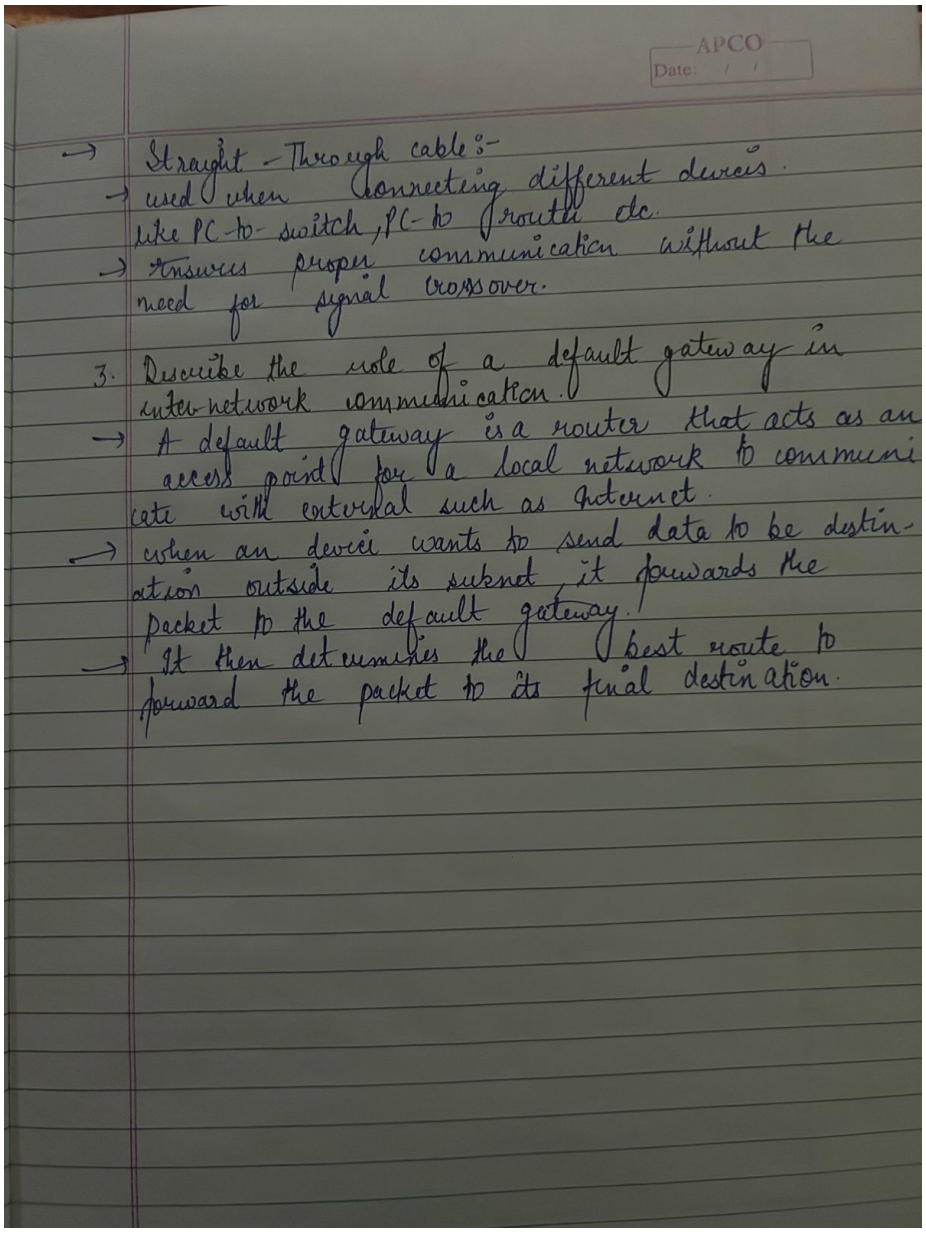
Static Routing	Dynamic Routing
(i) Manually configured by a network administrator.	(i) Automatically updates routing tables using routing protocols.
(ii) No extra CPU or bandwidth usage for route updates.	(ii) Consumes CPU and bandwidth due to routing protocol updates.
(iii) Small, stable networks where routes don't change often.	(iii) Large, complex networks with frequent topology changes.
(iv) Does not adapt to network changes unless manually updated.	(iv) Adapts automatically to network changes.

2. Why is crossover cable used for router-to-router connections, while a straight-through cable is used for PC-to-switch connections?

→ Crossover cable:

→ used when connecting two similar devices (e.g. router-to-router, switch-to-switch)

→ It swaps the transmission (TX) and reception (RX) wires so that the sending end of one device connects to the receiving end of other.

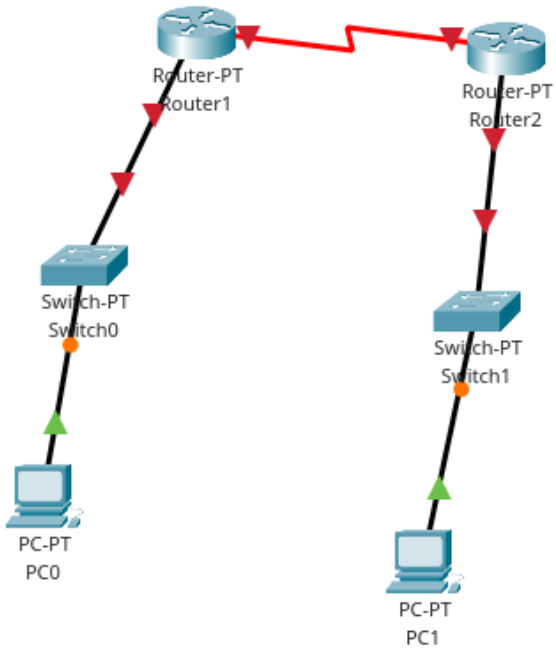


→ Straight - Through cable :-
→ used when connecting different devices like PC-to-switch, PC-to router etc.
→ ensures proper communication without the need for signal crossover.

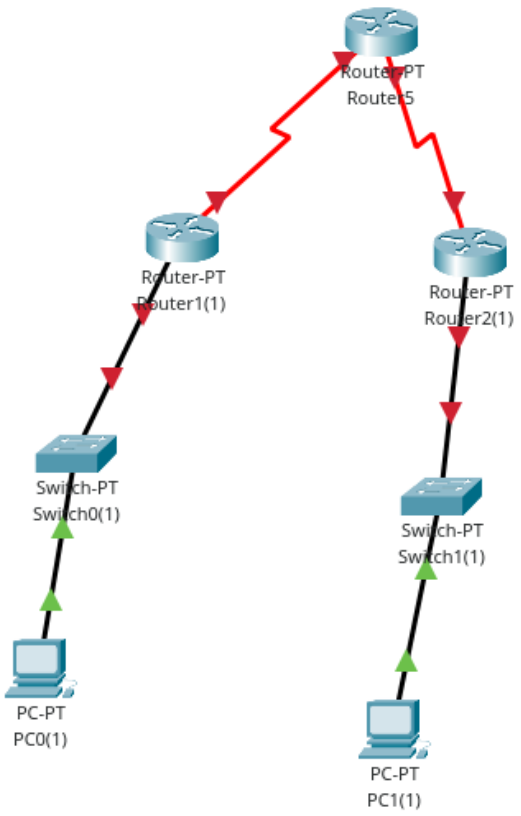
3. Describe the role of a default gateway in inter-network communication.
→ A default gateway is a router that acts as an access point for a local network to communicate with external such as Internet.
→ when an device wants to send data to be destination outside its subnet it forwards the packet to the default gateway.
→ It then determines the best route to forward the packet to its final destination.

Practical Section

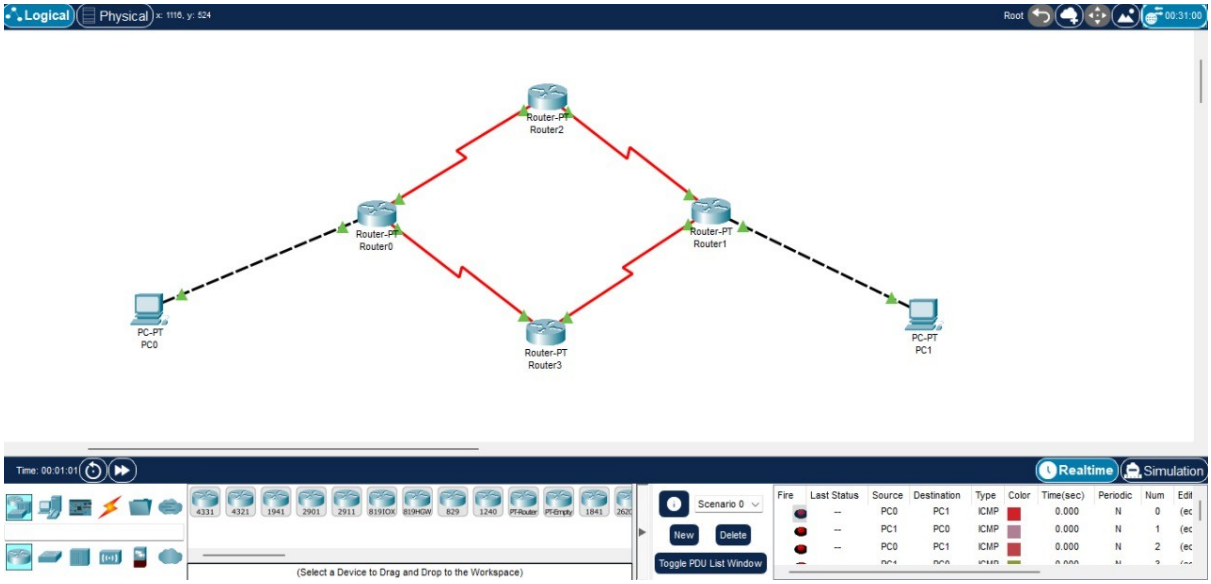
Question 1



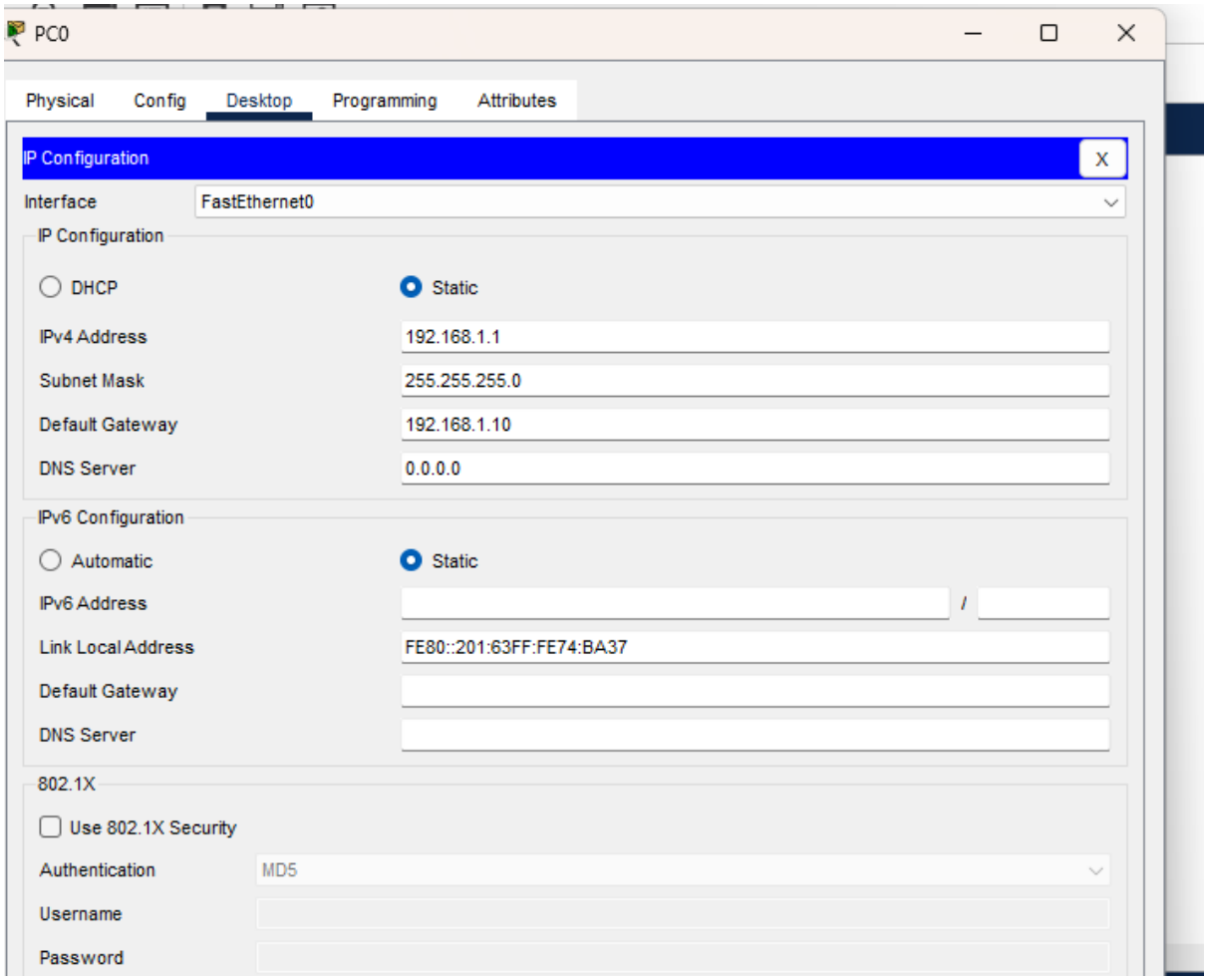
Question 2

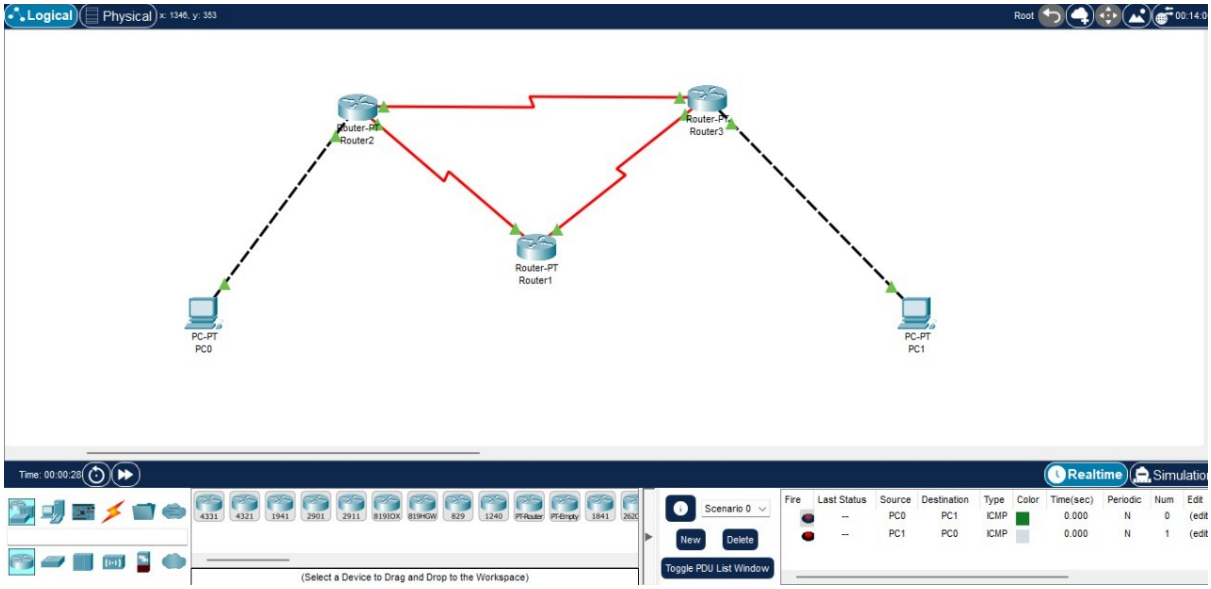

Router-PT
Router5
Router-PT
Router1(1)
Router-PT
Router2(1)
Switch-PT
Switch0(1)
Switch-PT
Switch1(1)
PC-PT
PC0(1)
PC-PT
PC1(1)

Distance Vector Routing (DVR) protocol using Packet Tracker

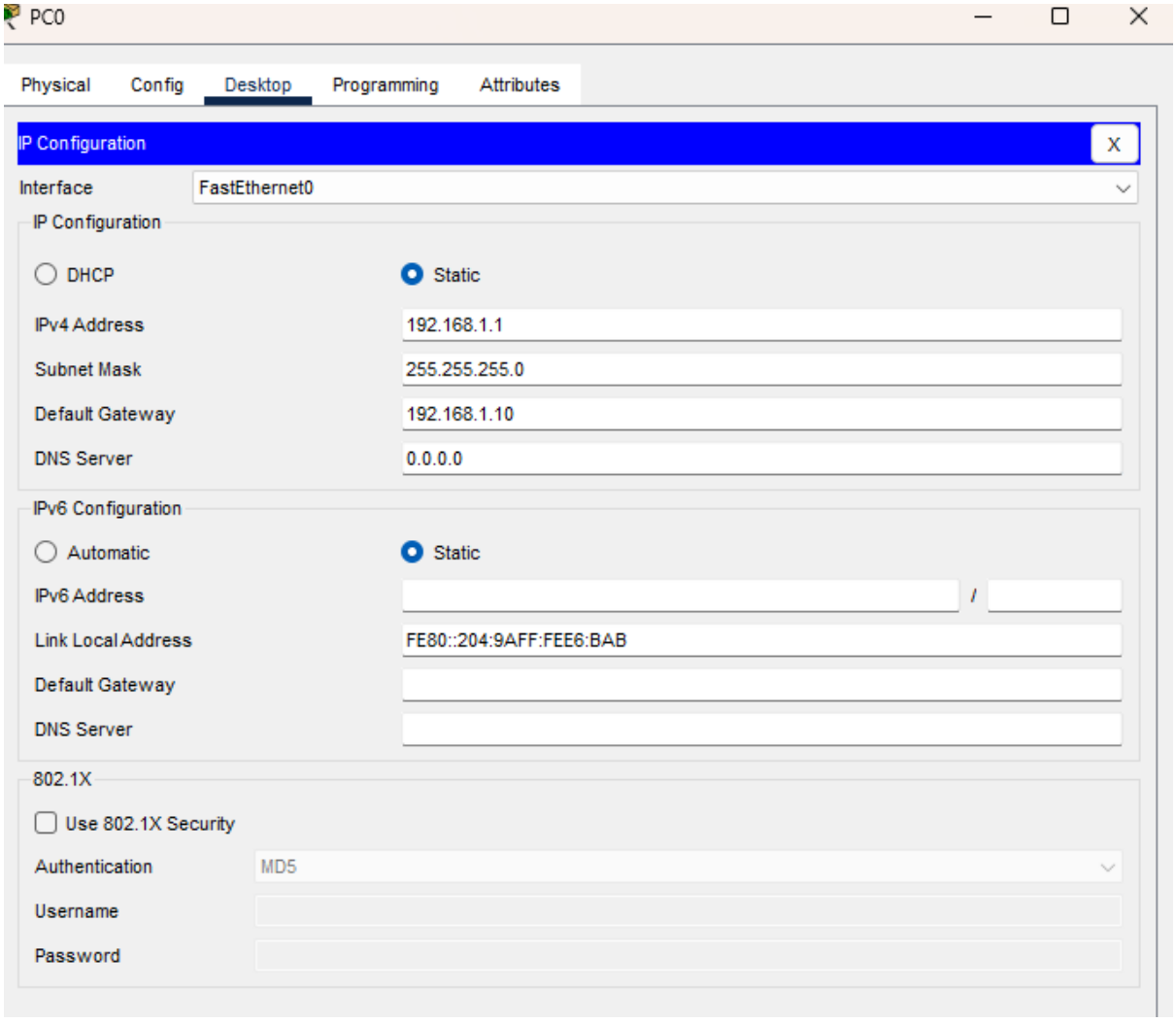


IP CONFIGURATION

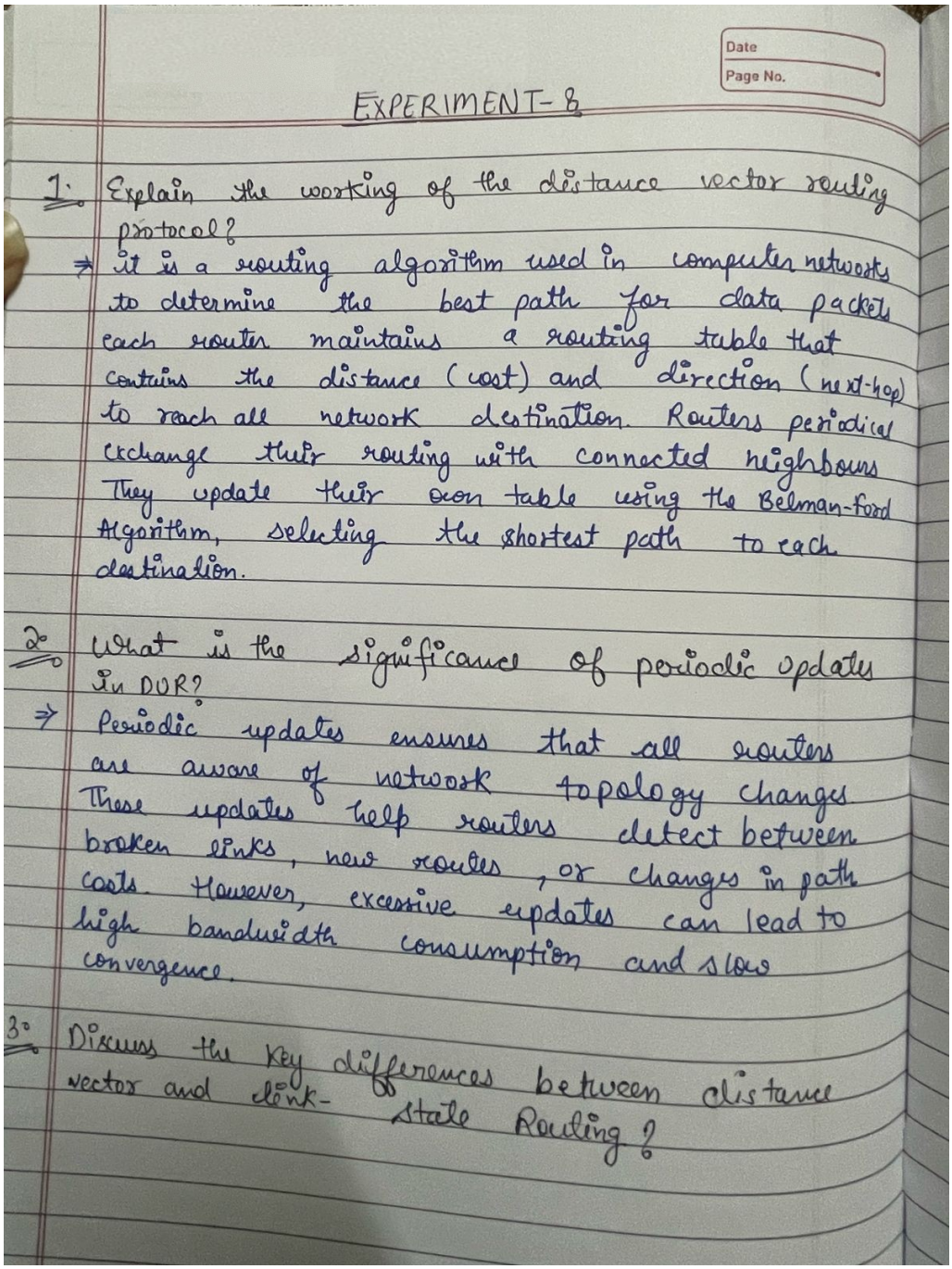




IP CONFIGURATION



THEORY QUESTIONS



Date
Page No.

EXPERIMENT-8

1. Explain the working of the distance vector routing protocol?

⇒ It is a routing algorithm used in computer networks to determine the best path for data packets each router maintains a routing table that contains the distance (cost) and direction (next-hop) to reach all network destination. Routers periodical exchange their routing with connected neighbours They update their own table using the Belman-ford Algorithm, selecting the shortest path to each destination.

2. What is the significance of periodic updates in DVR?

⇒ Periodic updates ensures that all routers are aware of network topology changes These updates help routers detect between broken links, new routes, or changes in path costs However, excessive updates can lead to high bandwidth consumption and slow convergence.

3. Discuss the key differences between distance vector and link-state Routing?

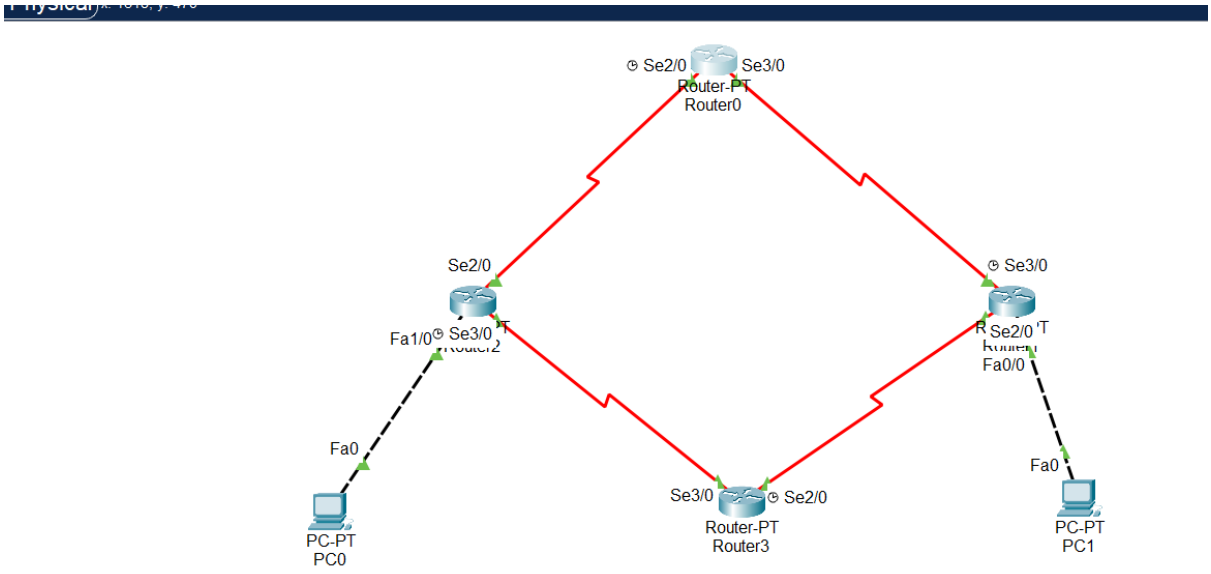
Date
Page No.

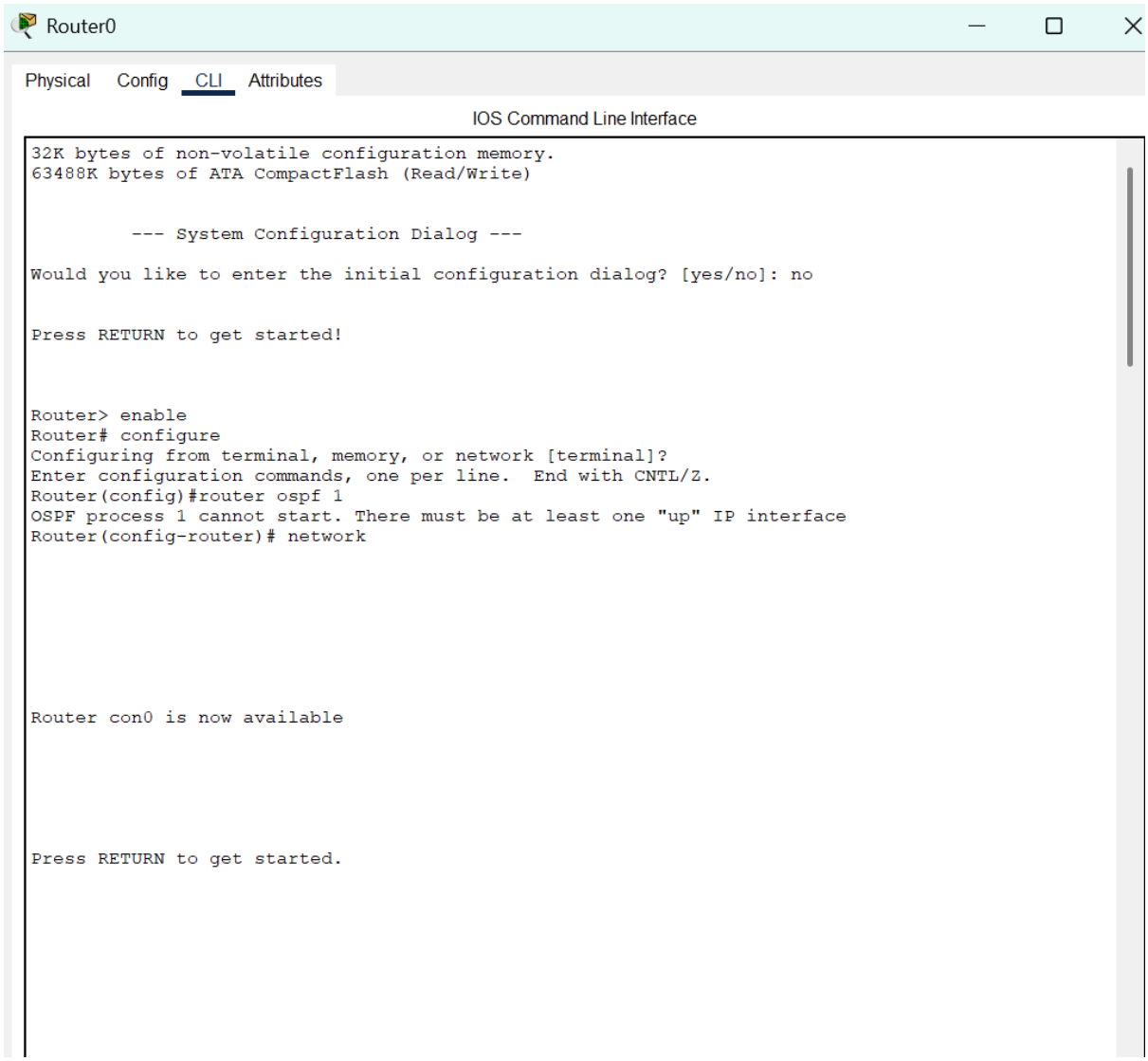
→ Distance vector routing	Link-state routing
i) Entire routing table shared with neighbours.	i) Only link-state information is shared
ii) Slower due to periodic updates	ii) Faster due to triggered updates.
iii) Computational overhead is low.	iii) Computational overhead is low.
iv) It is less scalable due to high bandwidth usage.	iv) It is more scalable for large networks.

4. What are the limitations of the DUR protocol, and how can they be mitigated?

i) It takes time to adapt to network changes, causing delays.
ii) Incorrect routes may loop indefinitely.
iii) Periodic updates consume excessive network resources
iv) Becomes inefficient in large networks due to frequency updates.

Assignment 9: Link-State Vector Routing Protocol Using Cisco Packet Tracer




Router0
Physical
Config
CLI
Attributes
IOS Command Line Interface
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]: no
Press RETURN to get started!
Router> enable
Router# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
OSPF process 1 cannot start. There must be at least one "up" IP interface
Router(config-router)# network
Router con0 is now available
Press RETURN to get started.

```
Router>enable
Router# configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#network 40.0.0.0 0.255.255.255 area 0
Router(config-router)#exit
Router(config)#interface serial2/0
Router(config-if)#ip address 30.30.30.2 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
Router(config-if)#interface serial2/0
Router(config-if)#
Router(config-if)#
```

```
Router con0 is now available
```

```
Press RETURN to get started.
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
```

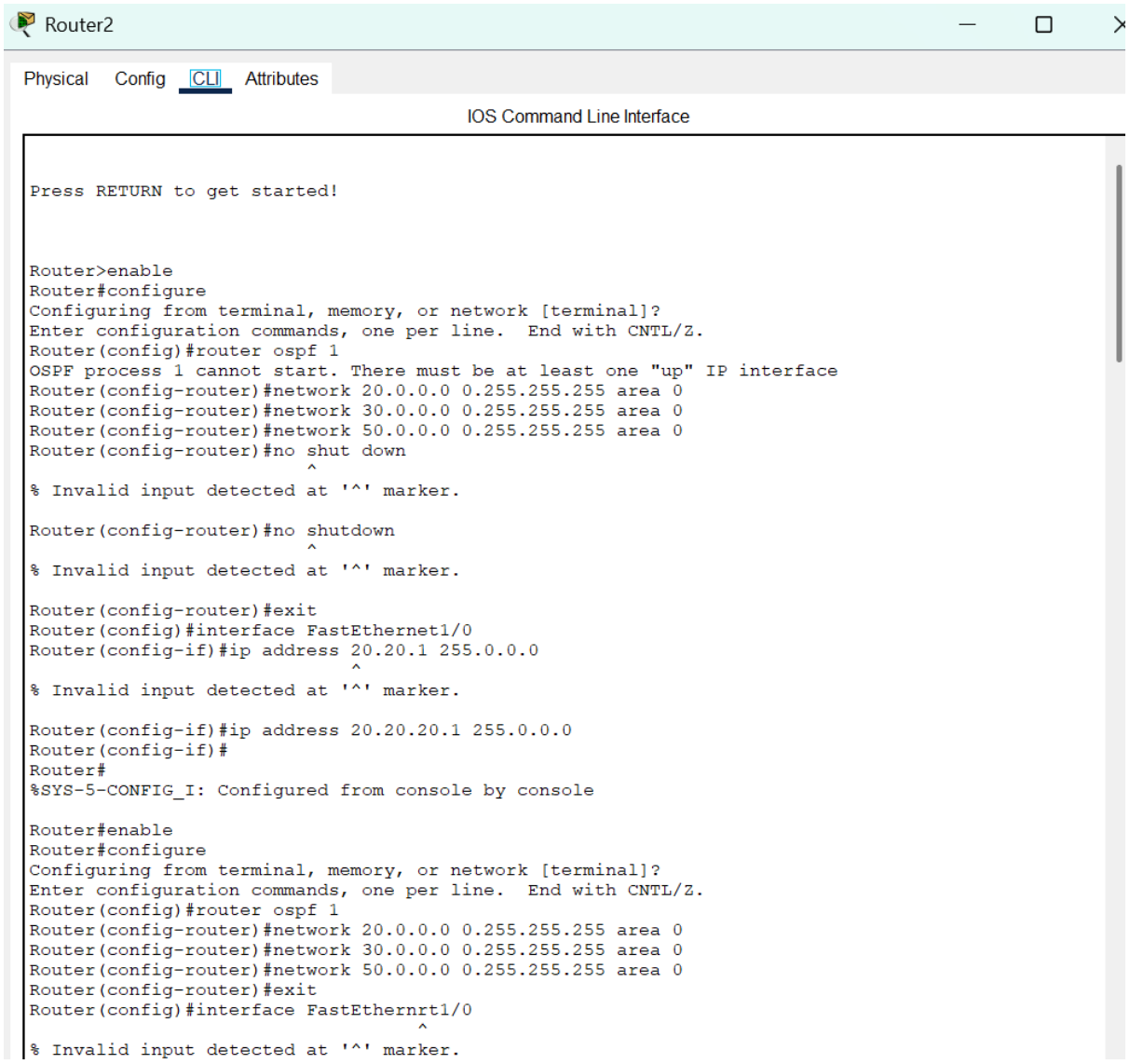
Router1

Physical Config CLI Attributes

IOS Command Line Interface

```
Router(config-if)#exit
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)#interface FastEthernet1/0
Router(config-if)#
%SYS-5-CONFIG_I: Configured from console by console

Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address
% Incomplete command.
Router(config-if)#ip address
% Incomplete command.
Router(config-if)#Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface FastEthernet0/0
Router(config-if)# Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#no ip address
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
```

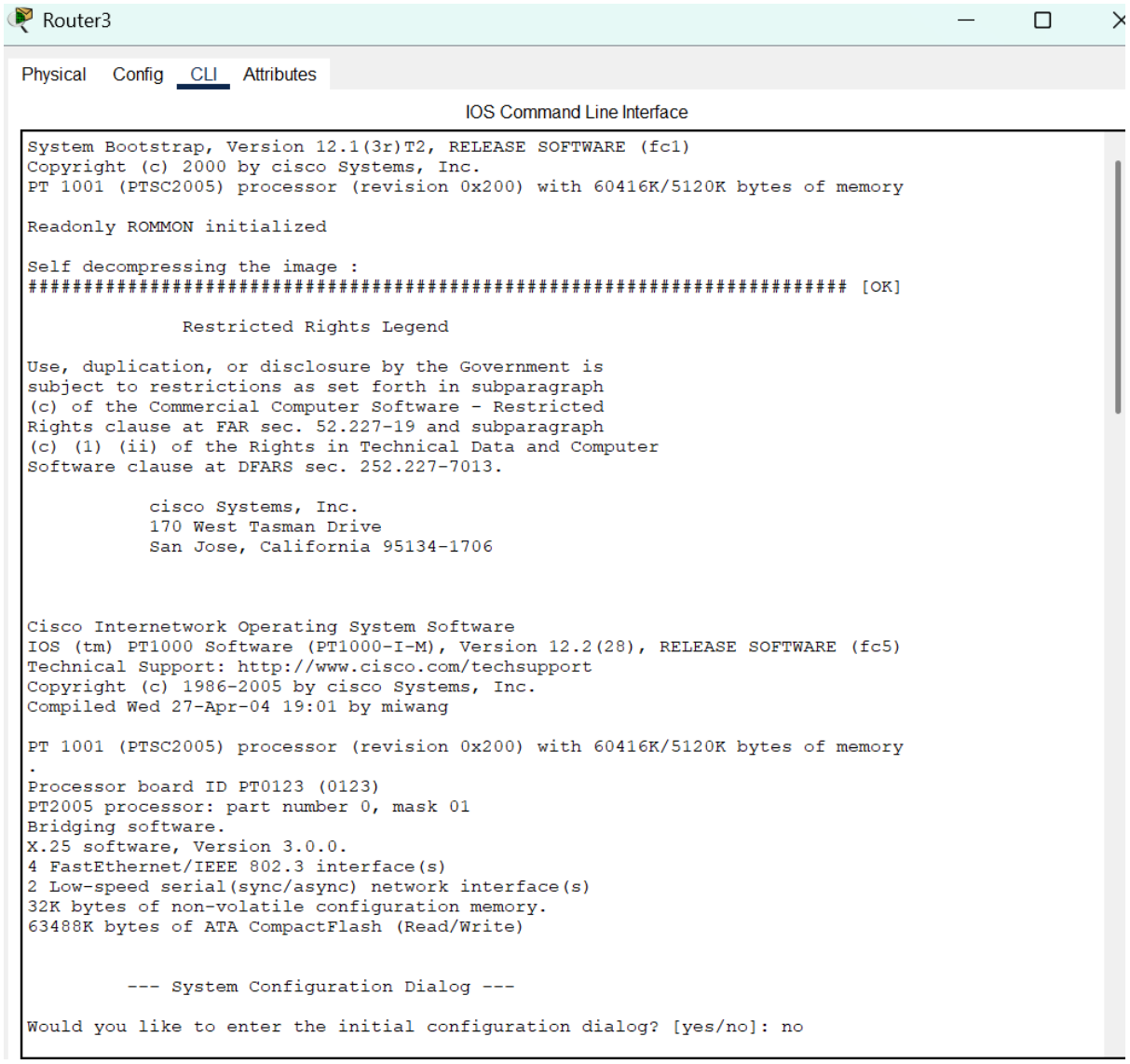
```
Router(config)#interface Serial3/0
Router(config-if)#Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface Serial3/0
Router(config-if)#ip address 30.30.30.1 255.0.0.0
Router(config-if)#ip address 30.30.30.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#no ip address
Router(config-if)#ip address 40.40.40.1 255.0.0.0
Router(config-if)#ip address 40.40.40.1 255.0.0.0
Router(config-if)#
Router(config-if)#exit
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)#interface Serial2/0
Router(config-if)#
%SYS-5-CONFIG_I: Configured from console by console

Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 50.0.0.0
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface Serial2/0
Router(config-if)#
%SYS-5-CONFIG_I: Configured from console by console

Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 30.30.30.1 255.0.0.0
Router(config-if)#
Router(config-if)#exit
```


Router2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Press RETURN to get started!
Router>enable
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
OSPF process 1 cannot start. There must be at least one "up" IP interface
Router(config-router)#network 20.0.0.0 0.255.255.255 area 0
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#network 50.0.0.0 0.255.255.255 area 0
Router(config-router)#no shut down
^
% Invalid input detected at '^' marker.
Router(config-router)#no shutdown
^
% Invalid input detected at '^' marker.
Router(config-router)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 20.20.1 255.0.0.0
^
% Invalid input detected at '^' marker.
Router(config-if)#ip address 20.20.20.1 255.0.0.0
Router(config-if)#
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#enable
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 20.0.0.0 0.255.255.255 area 0
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#network 50.0.0.0 0.255.255.255 area 0
Router(config-router)#exit
Router(config)#interface FastEthernrt1/0
^
% Invalid input detected at '^' marker.

```
Router(config-if)#ip address 20.20.20.1 255.0.0.0
Router(config-if)#Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 20.20.20.1 255.0.0.0
Router(config-if)#n0 shutdown
                  ^
% Invalid input detected at '^' marker.

Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interfaceEnter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface FastEthernet0/0
Router(config-if)#interface serial3/0
Router(config-if)#ip address 30.30.30.1 255.0.0.0
Router(config-if)# interface serial2/0
Router(config-if)# ip address 50.50.50.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up









Router con0 is now available
```

```
Press RETURN to get started.



Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface FastEthernet0/0
Router(config-if)#Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface FastEthernet0/0
Router(config-if)#no ip address
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
```

```
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
%IP-4-DUPADDR: Duplicate address 20.20.20.1 on FastEthernet1/0, sourced by 0060.2F11.364E

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up









Router con0 is now available




Press RETURN to get started.
```


Router3
Physical
Config
CLI
Attributes
IOS Command Line Interface
System Bootstrap, Version 12.1(3r)T2, RELEASE SOFTWARE (fc1)
Copyright (c) 2000 by cisco Systems, Inc.
PT 1001 (PTSC2005) processor (revision 0x200) with 60416K/5120K bytes of memory
Readonly ROMMON initialized
Self decompressing the image :
[OK]
Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.
cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
Cisco Internetwork Operating System Software
IOS (tm) PT1000 Software (PT1000-I-M), Version 12.2(28), RELEASE SOFTWARE (fc5)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:01 by miwang
PT 1001 (PTSC2005) processor (revision 0x200) with 60416K/5120K bytes of memory
.
Processor board ID PT0123 (0123)
PT2005 processor: part number 0, mask 01
Bridging software.
X.25 software, Version 3.0.0.
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]: no

```
Press RETURN to get started!



Router>enable
Router#configure
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 1
OSPF process 1 cannot start. There must be at least one "up" IP interface
Router(config-router)#network 40.0.0.0 0.255.255.255 area 0
Router(config-router)#network 50.0.0.0 0.255.255.255 area 0
Router(config-router)#exit
Router(config)#interface serial3/0
Router(config-if)#ip address 40.40.40.1 255.00.0
                                        ^
% Invalid input detected at '^' marker.

Router(config-if)#40.40.40.1 255.0.0.0
                  ^
% Invalid input detected at '^' marker.

Router(config-if)#ip address 40.40.40.1 255.0.0.0
Router(config-if)#interface serial2/0
Router(config-if)#ip address 50.50.50.1 255.0.0.0
Router(config-if)#Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
```

```
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#no shutdown
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

02:50:56: %OSPF-5-ADJCHG: Process 1, Nbr 30.30.30.1 on Serial2/0 from LOADING to FULL, Loading
Done

%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up









Router con0 is now available




Press RETURN to get started.
```

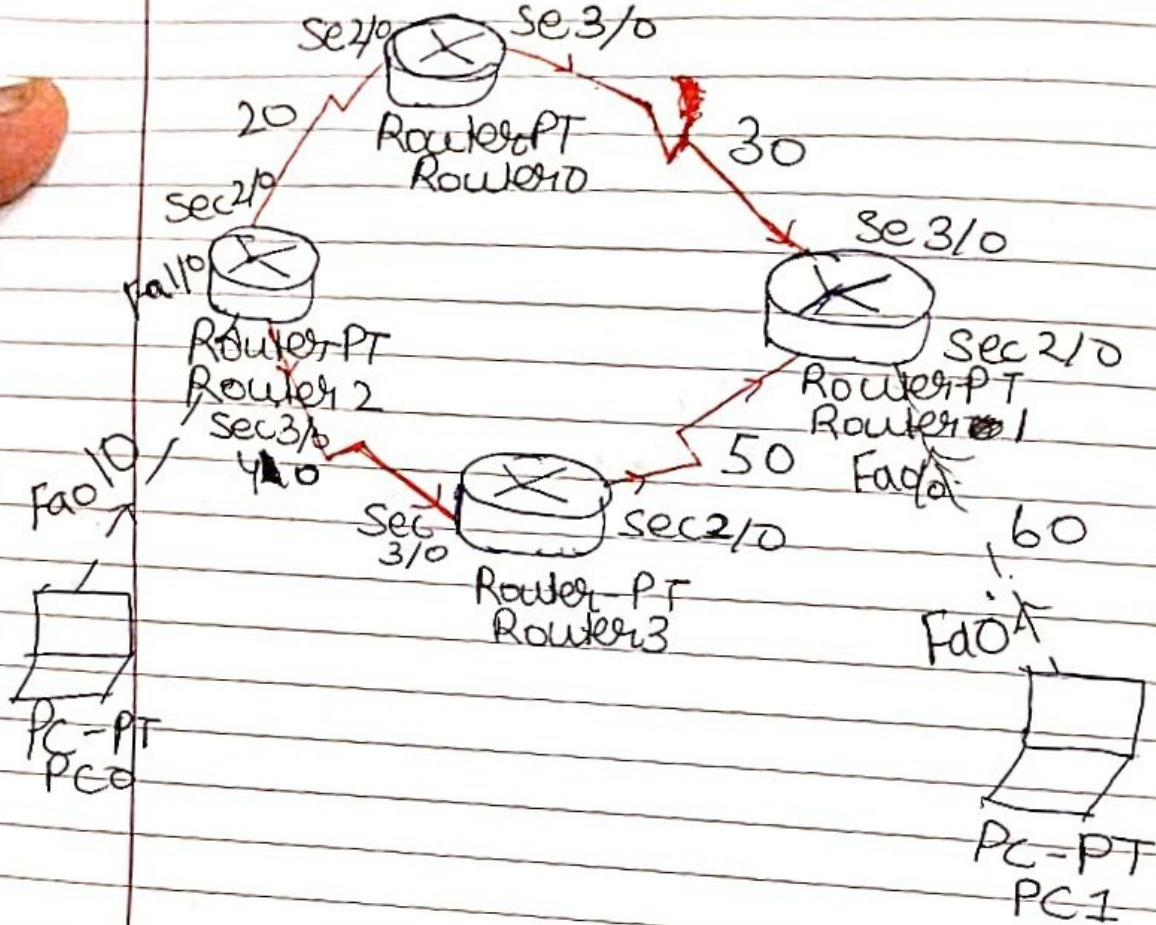
```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down

03:23:57: %OSPF-5-ADJCHG: Process 1, Nbr 30.30.30.1 on Serial2/0 from FULL to DOWN, Neighbor
Down: Interface down or detached

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

03:24:17: %OSPF-5-ADJCHG: Process 1, Nbr 30.30.30.1 on Serial2/0 from LOADING to FULL, Loading
Done
```

```
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
%SYS-5-CONFIG_I: Configured from console by console

Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
%SYS-5-CONFIG_I: Configured from console by console

Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
02:51:17: %OSPF-5-ADJCHG: Process 1, Nbr 50.50.50.1 on Serial2/0 from LOADING to FULL, Loading
Done

Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
```

6/4/25

Date
Page

Assignment 9 : Link-state vector Routing Protocol using cisco Packet Tracer



```
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up









Router con0 is now available




Press RETURN to get started.
```

Section 1: Nmap (Network Mapper)

1. What is Nmap, and what is its primary purpose in network monitoring?

- Nmap (Network Mapper) is an open-source network scanning tool used for network discovery and security auditing. Its primary purpose is to identify active devices, detect open ports, discover services running on a network and assess security vulnerabilities.

2. Explain the different types of scans available in Nmap. Provide examples.

- TCP Connect Scan (nmap -sT): Performs a full TCP three-way handshake to determine open ports.
- SYN Scan (nmap -sS): A stealthy scan that sends SYN packets and detects open ports without completing the handshake.
- UDP scan (nmap -sU): Scans for open UDP ports, useful for detecting services like DNS and SNMP.

APCO
Date: / /

3. How does Nmap help in identifying open ports and services on a network?
- Nmap sends specially crafted packets to target systems and analyzes the responses to determine which ports are open, closed or filtered. It also uses detection (-sV) to identify the specific applications running on open ports.

4. What is OS detection in Nmap and how does it work?
- OS detection (-O flag) allows Nmap to determine the operating system of a target device by analyzing factors such as TCP/IP stack responses, packet TTL values and specific fingerprinting techniques.

5. Discuss the significance of the -A option in Nmap scanning.
- The (-A) option enables aggressive scanning which includes OS detection, version detection, script scanning, and traceroute. It provide comprehensive information about the target network but is more intrusive.

6. How can Nmap be used for security auditing and penetration testing?
→ It identifies open ports and running services, exposing potential vulnerabilities.
→ It detects outdated software versions with known exploits.

→ Uncovers misconfigurations and weak security controls.

7. What are the limitations of Nmap in network Monitoring?

→ It cannot detect all vulnerabilities - relies on known signatures.

→ Slow performance on large network or when scanning with deep analysis options.

→ Limited effectiveness against encrypted traffic.

Section 2: Wireshark.

1. What is Wireshark, and how it used in network analysis?

→ Wireshark is powerful open-source packet analyser used for real-time network traffic analysis. It captures and inspects data packets travelling over a network, helping network administrators and security professionals, detect intrusions.

2. Explain the different components of a captured packet in Wireshark.

→ Frame Header - Contains metadata such as timestamp, frame length and source/destination MAC addresses.

APCO
Date: / /

→ Ethernet Header: Includes source and destination MAC addresses and the Ether Type field indicating the next protocol layer

→ IP Header: shows the source and destination IP addresses, protocol type (TCP, UDP, ICMP) and packet fragmentation details.

→ Transport layer: Display port numbers, sequence/acknowledgment numbers and flags.

3. How does Wireshark help in troubleshooting network issues?

→ It identifies latency and network congestion by analyzing packet delays.

→ Detects dropped packets, retransmissions and high error rates.

→ Analyzes DNS, HTTP traffic to pinpoint misconfigurations.

4. Discuss the importance of filtering packets in Wireshark?

→ Capture filters: Limit the data collected during packet capture.

→ Display filters: Narrow down displayed packets for easier analysis.

→ Filtering enhances performance, reduces noise, speeds up troubleshooting.

APCO
Date: / /

5. What security concerns should be considered when using Wireshark?

→ Unauthorised packet capture can expose sensitive data such as passwords or confidential communications.

→ Wireshark does not decrypt encrypted traffic but plaintext data remains visible.

→ Running Wireshark with administrative privileges can pose security risks.

6. How does Wireshark differentiate between various network protocols?

→ Wireshark use protocol dissectors, which analyze packet headers and payloads to classify traffic based on predefined rules. It recognizes protocols by inspecting fields such as Ethernet, port numbers and protocol identifiers (IP, ARP).

7. Why is Wireshark widely used in cybersecurity investigations?

→ It detects suspicious network traffic, such as malware communication and unauthorised access attempts.

→ Monitors encrypted traffic patterns for anomalies without decrypting data

→ It helps in analyse distributed attacks by identifying malicious packet floods.

Section 3: Comparitive Analysis

1. Compare and contrast Nmap and Wireshark in terms of their functionalities and use cases.

→

Nmap	Wireshark
(i) Network Scanning and discovery.	(i) Packet capture and analysis
(ii) It identifies open ports, services and hosts on a network.	(ii) It inspects the real-time network traffic at a granular level.
(iii) It sends packets to gather information.	(iii) It listens to network traffic without sending packets.
(iv) It identifies network structure, weak points and misconfigurations	(iv) It Analyses detailed network communications and detecting anomalies.

2. Which tool would you use for detecting open ports and why? → Nmap is the best tool because:

→ It actively probes target hosts and check port statuses (open, closed, filtered).

→ It provides detailed service

→ It is designed specifically for network scanning and security auditing

APCO
Date: / /

3. How can both Nmap and Wireshark be used together for network security analysis?

→ Step 1: Use Nmap to scan for vulnerabilities
- Identify active hosts.
- Detect open ports and services.
- Find OS details.

→ Step 2: Use Wireshark to analyze suspicious traffic
- Capture network traffic to examine packet behaviour.
- Filter specific traffic.
- Detect anomalies such as excessive failed login attempts or unusual data transfers.

→ Step 3: Correlate findings
- If Nmap detects an open port running a vulnerable service, Wireshark can confirm if it's being exploited.
- If Wireshark detects suspicious traffic, Nmap can scan for unprotected devices that may be the source.

4. Discuss a real-world scenario where Nmap and Wireshark would help in identifying a network attack.

APCO
Date: / /

— Scenario – Detecting a Brute-force Attack on a SSH Server

→ Step 1: Use Nmap to identify Open SSH Ports
- Run nmap -p 22 -sV target_ip to check if SSH is open.
- If detected, further analyse version details to check for vulnerabilities.

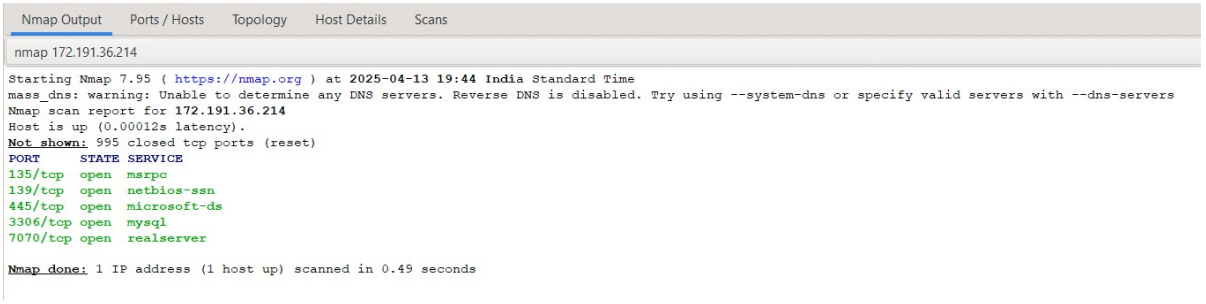
→ Step 2: Use Wireshark to Monitor SSH traffic
- Apply a filter (tcp.port == 22) to capture all SSH related packets
- Detect multiple failed login attempts from the same IP, indicating a brute-force attack.

→ Step 3: Take Action
- Block the attacker's IP using a firewall or intrusion prevention system (IPS)
- Enforce stronger authentication mechanisms
- Monitor further traffic for other attack patterns

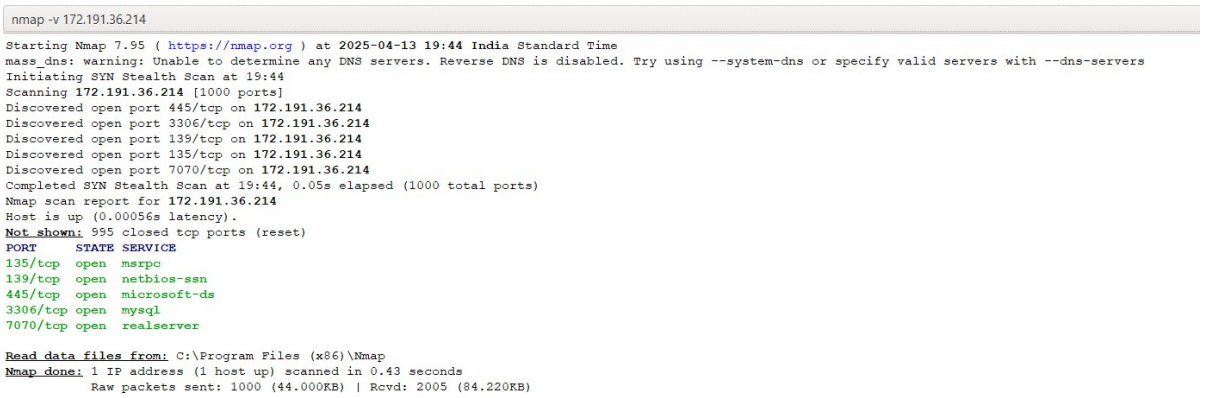
Assignment-11

1. Basic Nmap Commands

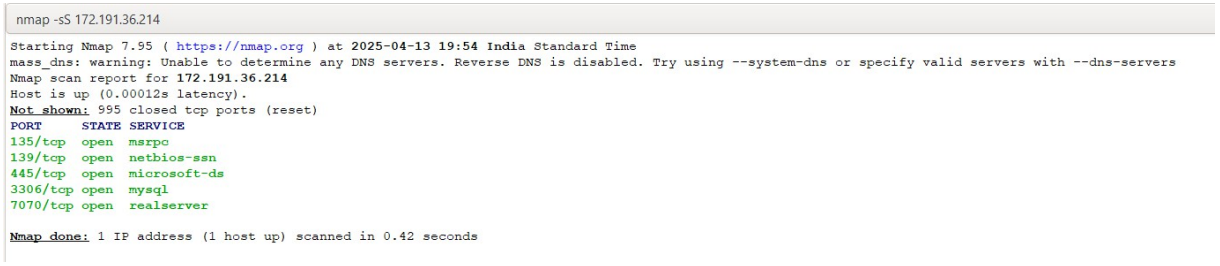
nmap<target> **Basic scan of a target (IP or hostname)**



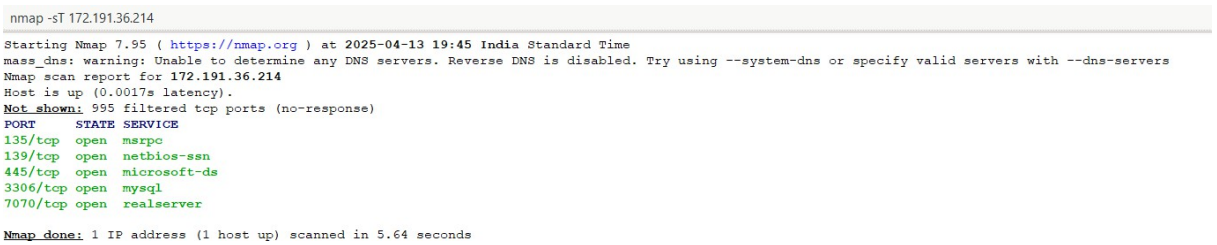
nmap – v <target> **Verbose output of the scan**



nmap -sS <target> **SYN (Stealth) scan**



nmap -sT <target> **TCP Connect scan (less stealthy)**



nmap -sU <target> **UDP scan**

nmap -sU 172.191.36.214

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 19:45 India Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 172.191.36.214
Host is up (0.00031s latency).
Not shown: 990 closed udp ports (port-unreach)
PORT     STATE         SERVICE
80/udp   open|filtered http
123/udp  open|filtered ntp
137/udp  open|filtered netbios-ns
138/udp  open|filtered netbios-dgm
443/udp  open|filtered https
1900/udp open|filtered upnp
4500/udp open|filtered nat-t-ike
5050/udp open|filtered mmcc
5353/udp open|filtered zeroconf
5355/udp open|filtered llmnr

Nmap done: 1 IP address (1 host up) scanned in 181.33 seconds
```

nmap -p <target> Scan all 65535 ports

nmap -p - 172.191.36.214

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 19:46 India Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 172.191.36.214
Host is up (0.00034s latency).
Not shown: 65516 closed tcp ports (reset)
PORT      STATE    SERVICE
135/tcp   open     msrpc
137/tcp   filtered netbios-ns
139/tcp   open     netbios-ssn
445/tcp   open     microsoft-ds
3306/tcp  open     mysql
5040/tcp  open     unknown
7070/tcp  open     realserver
7680/tcp  open     pando-pub
8885/tcp  open     unknown
8886/tcp  open     unknown
33060/tcp open     mysqlx
49664/tcp open     unknown
49665/tcp open     unknown
49668/tcp open     unknown
49669/tcp open     unknown
49672/tcp open     unknown
49688/tcp open     unknown
57272/tcp open     unknown
57621/tcp open     unknown

Nmap done: 1 IP address (1 host up) scanned in 4.22 seconds
```

nmap -O <target> Try to detect OS

nmap -O 172.191.36.214

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 19:47 India Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 172.191.36.214
Host is up (0.00042s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
7070/tcp open  realserver
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.95%E=4%D=4/13%OT=135%CT=1%CU=34769%PV=N%DS=0%DC=L%G=Y%TM=67FBC7
OS:84%P=i686-pc-windows-windows)SEQ(SP=103%GCD=1%ISR=109%TI=I%CI=I%II=I%SS=
OS:S%TS=A)SEQ(SP=104%GCD=1%ISR=109%TI=I%CI=I%II=I%SS=S%TS=A)SEQ(SP=106%GCD=
OS:1%ISR=108%TI=I%CI=I%II=I%SS=S%TS=A)SEQ(SP=F2%GCD=1%ISR=10C%TI=I%CI=I%II=
OS:I%SS=S%TS=A)OPS(O1=MFFD7NW8ST11%O2=MFFD7NW8ST11%O3=MFFD7NW8NNT11%O4=MFFD
OS:7NW8ST11%O5=MFFD7NW8ST11%O6=MFFD7ST11)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFF
OS:F%W5=FFFF%W6=FFFF)ECN(R=Y%DF=Y%T=80%W=FFFF%O=MFFD7NW8NNS%CC=N%Q=)T1(R=Y%
OS:DF=Y%T=80%S=O%A=S+%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=
OS:0%Q=)T3(R=Y%DF=Y%T=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S
OS:=A%A=O%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R
OS:=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=
OS:AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=Z%RUCK=G%
OS:RUD=G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.70 seconds
```

nmap -A <target> Aggressive scan (OS, version, script scanning, traceroute)

nmap -A 172.191.36.214

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 19:48 India Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 172.191.36.214
Host is up (0.00051s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE          VERSION
135/tcp  open  msrpc            Microsoft Windows RPC
139/tcp  open  netbios-ssn      Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3306/tcp open  mysql            MySQL (unauthorized)
7070/tcp open  ssl/realserver?
| ssl-cert: Subject: commonName=AnyDesk Client
| Not valid before: 2023-07-27T16:15:58
|_Not valid after:  2073-07-14T16:15:58
|_ssl-date: TLS randomness does not represent time
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.95%E=4%D=4/13%OT=135%CT=1%CU=32230%PV=N%DS=0%DC=L%G=Y%TM=67FBC7
OS:CA%P=i686-pc-windows-windows)SEQ(SP=101%GCD=1%ISR=10B%TI=I%CI=I%II=I%SS=
OS:S%TS=A)SEQ(SP=106%GCD=1%ISR=108%TI=I%CI=I%II=I%SS=S%TS=A)SEQ(SP=FC%GCD=1
OS:%ISR=FF%TI=I%CI=I%II=I%TS=A)SEQ(SP=FE%GCD=1%ISR=10C%TI=I%CI=I%II=I%SS=S%
OS:TS=A)SEQ(SP=FF%GCD=1%ISR=108%TI=I%CI=I%II=I%SS=S%TS=A)OPS(O1=MFFD7NW8ST1
OS:1%O2=MFFD7NW8ST11%O3=MFFD7NW8NNT11%O4=MFFD7NW8ST11%O5=MFFD7NW8ST11%O6=MF
OS:FD7ST11)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFFF)ECN(R=Y%DF=Y
OS:%T=80%W=FFFF%O=MFFD7NW8NNS%CC=N%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S+%F=AS%RD=0%Q
OS:=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%T=80%W=0%S=Z%
OS:A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=0%Q=)T5(R=Y%D
OS:F=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O
OS:=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=80
OS:%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=Z%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=80%CD=Z)

Network Distance: 0 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-04-13T14:18:39
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.82 seconds
```

nmap -sV <target> Detect service versions

nmap -sV 172.191.36.214

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 19:49 India Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 172.191.36.214
Host is up (0.00091s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE          VERSION
135/tcp  open  msrpc            Microsoft Windows RPC
139/tcp  open  netbios-ssn      Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3306/tcp open  mysql            MySQL (unauthorized)
7070/tcp open  ssl/realserver?
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.96 seconds
```

nmap -T4 <target> Faster scan timing (T0 to T5, T4 is common for speed)

nmap -T4 172.191.36.214

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 19:50 India Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 172.191.36.214
Host is up (0.000071s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
7070/tcp open  realserver

Nmap done: 1 IP address (1 host up) scanned in 0.50 seconds
```

nmap -Pn <target> Skip host discovery (assume host is up)

nmap -Pn 172.191.36.214

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 19:50 India Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 172.191.36.214
Host is up (0.00016s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
7070/tcp open  realserver

Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds
```

nmap --script vuln <target> **Run common vulnerability scripts**

```
nmap --script vuln 172.191.36.214
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 19:52 India Standard Time
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 172.191.36.214
Host is up (0.00034s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3306/tcp open  mysql
7070/tcp open  realserver

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
|_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Nmap done: 1 IP address (1 host up) scanned in 29.05 seconds
```

2. Capturing and Analyzing Network Traffic Using Wireshark

1. Download and Install Wireshark:

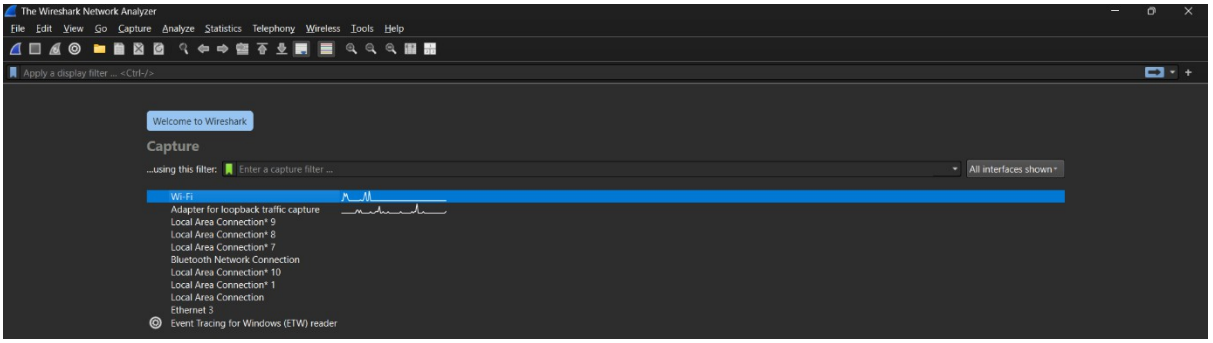
o Visit https://www.wireshark.org/

o Download and install the latest version for your operating system.

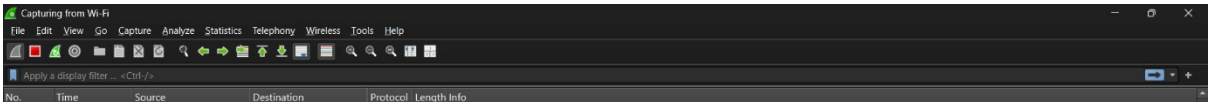
2. Start a Packet Capture Session:

o Open Wireshark.

o Select your active network interface (e.g., Wi-Fi or Ethernet).



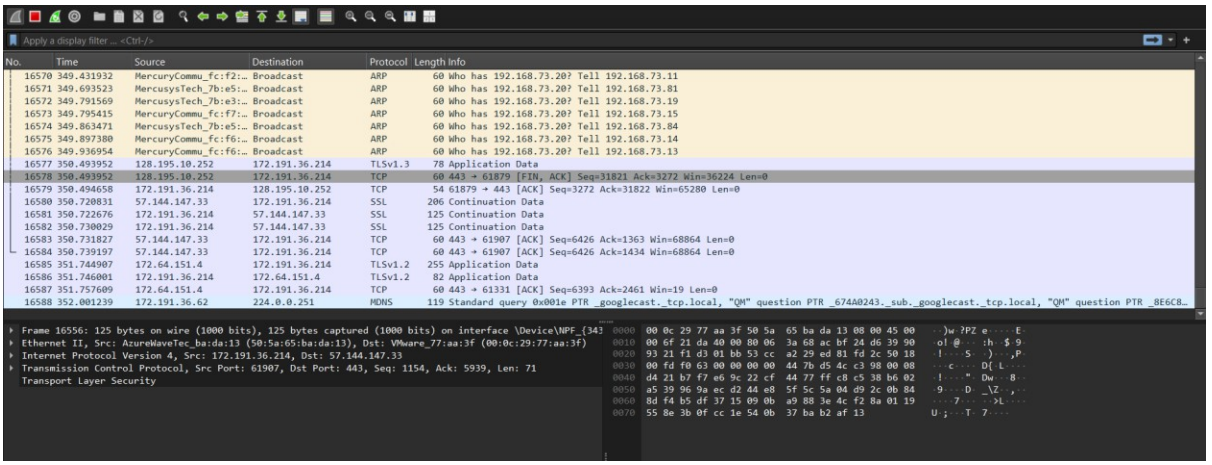
o Click the Start Capturing Packets button (shark fin icon).



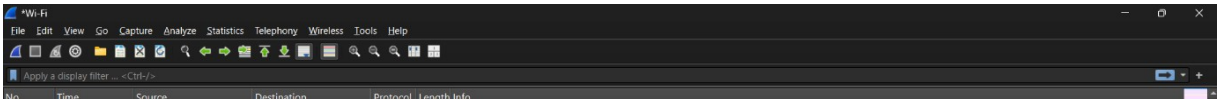
3. Generate Some Network Traffic:

o Open a web browser and visit a few websites.

o You may also open apps that use the internet like email or messaging tools.

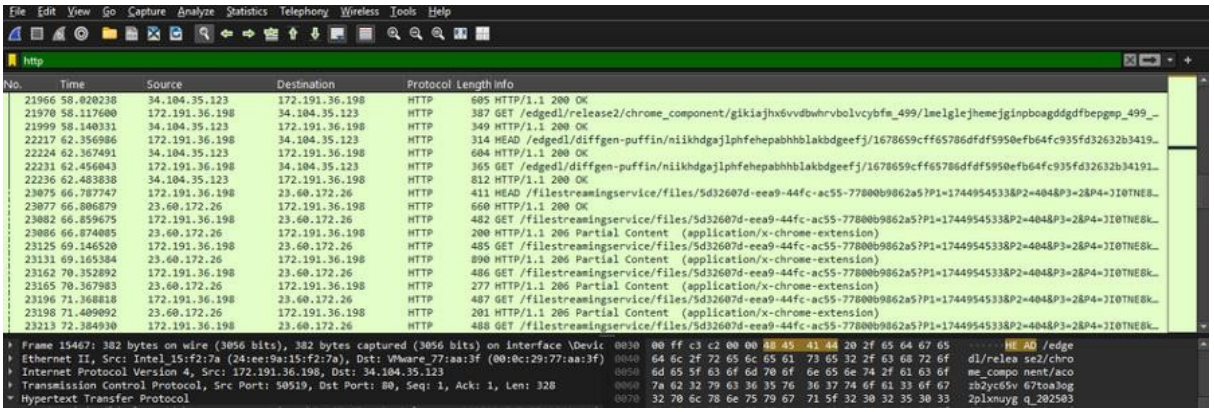


4. **Stop the Capture:**
   o After about 2–5 minutes, stop the capture using the red square icon in Wireshark.

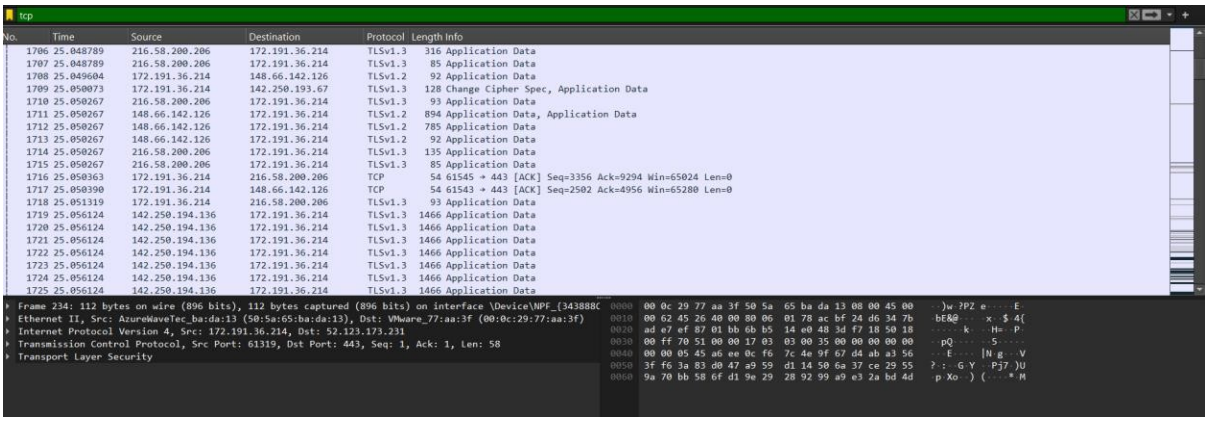


5. **Analyze the Traffic:**
   o Use filters Simple filter HTTP, DNS, or TCP to view specific types of traffic.

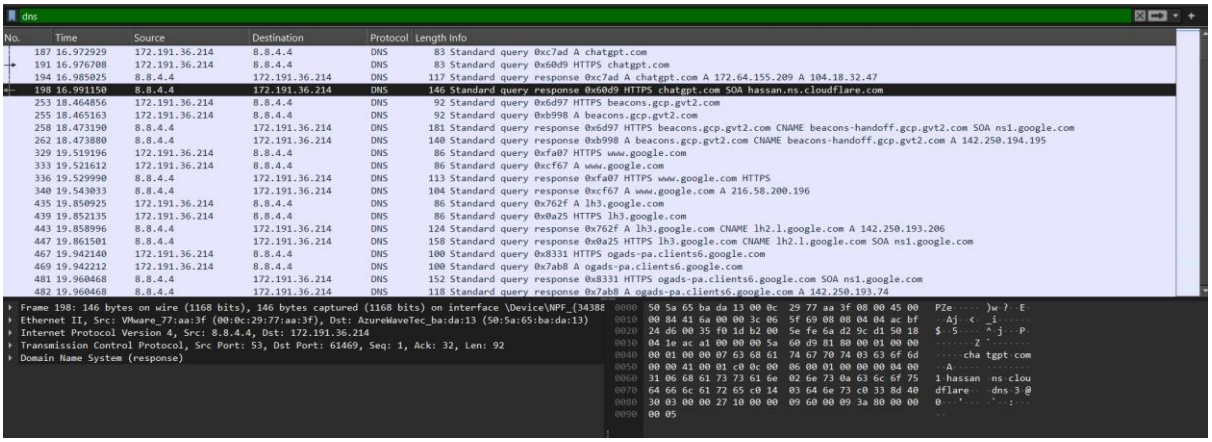
HTTP :



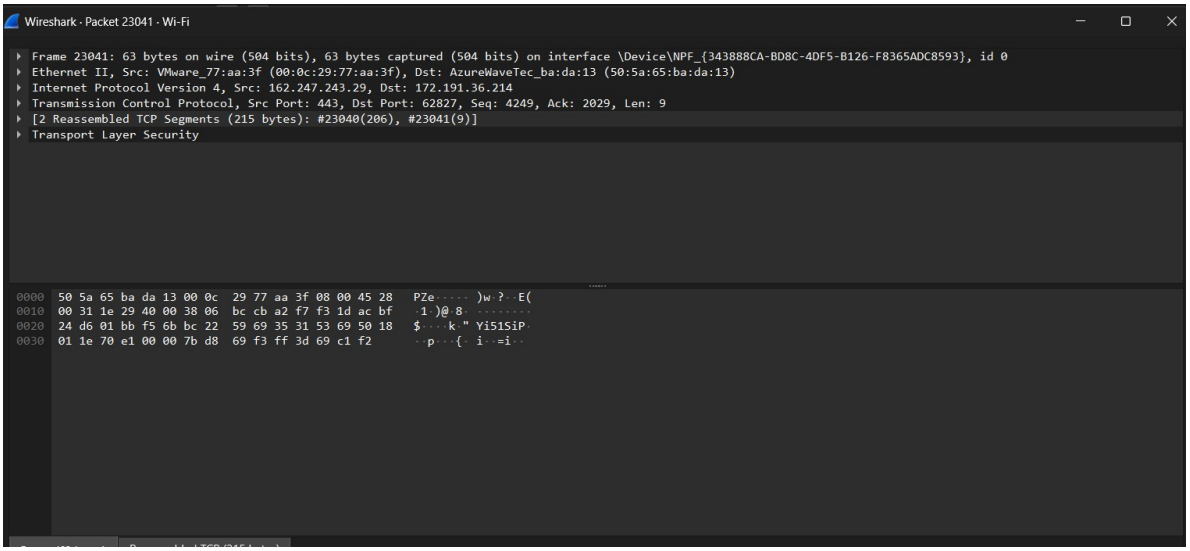
TCP :

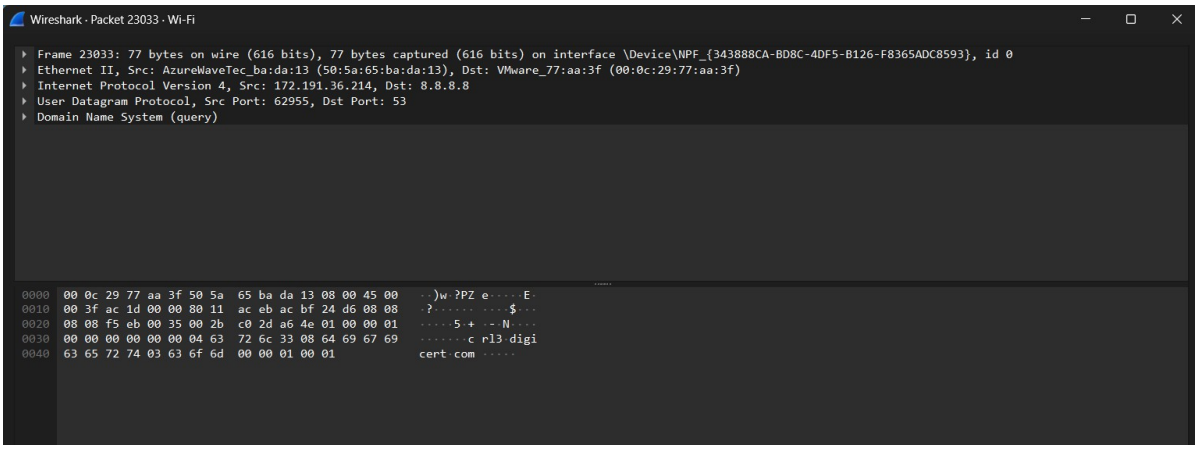


DNS :



o Select a few packets and inspect their details in the packet details pane.





6. **Export Your Capture: Save the capture as a .pcapng file for later review.**

   https://drive.google.com/drive/folders/1aHY89uyl8rcGuUdVozl8XzqsG5Se3KVf?usp=drive_link

Theory:-

Assignment-11

1) Basic Nmap Commands

	Command	Description
a)	nmap <target>	Basic scan of a target
b)	nmap -v <target>	verbose output of the scan
c)	nmap -sS <target>	SYN (stealth) scan
d)	nmap -sT <target>	TCP connect scan (less stealthy)
e)	nmap -sU <target>	UDP scan
f)	nmap -p <Port> <target>	scan specific port(s)
g)	nmap -p <target>	scan all 65535 ports
h)	nmap -O <target>	try to detect OS
i)	nmap -A <target>	aggressive scan (OS, version, script scanning)
j)	nmap -sV <target>	detect service versions
k)	nmap -T4 <target>	faster scan timing (T0 to T5, T4 is common for speed)
l)	nmap -Pn <target>	skip host discovery (assume host is up)
m)	nmap -iL target.txt	scan multiple targets from a file.
n)	nmap -oN output.txt <target>	save output in normal format
o)	nmap --script <script-name> <target>	run specific NSE script
p)	nmap --script vuln <target>	Run common vulnerability scripts.

2. Capturing and analyzing network traffic using wireshark.

Q1 What interface did you select for capturing packets?

Soln For capturing packets, I selected Wi-Fi interface because my computer is connected to the internet via a wireless network. In wireshark all available network interfaces are listed when the application is launched, and selecting the correct one is essential to monitor the traffic effectively. The Wi-Fi interface typically shows ongoing data transmission in the form of spikes which indicates active network traffic. This interface allows the capture of all incoming and outgoing packets from the wireless connection in real-time.

Q2 List 3 protocol you observed during the capture session?

Soln I observed the three commonly used network protocols.

1) HTTP (Hyper text transfer protocol) used for transferring web page over the internet. I noticed several HTTP requests when I opened different websites.

2) DNS (Domain name system)
this protocol translates domain names into IP addresses whenever I typed a URL, DNS packet generated to resolved the IP address.

3) TCP (Transmission control protocol)
A connection-oriented protocol that ensures reliable communication between devices. TCP packets were visible in every connection, especially during communications.

Q3 Pick one HTTP packet. What was the host website being accessed?

⇒ I selected an HTTP GET request packet during the session. The host accessed in that particular packet was:
www.example.com (Note: Replace this with actual domain name you saw during my session such as www.google.com, www.wikipedia.org etc.
In the packet details, under the HTTP section the 'Host' field displays the exact domain being requested. The host field shows which server my browser was typing to reach using the HTTP protocol.

Q4 What is the IP address of your system and the destination IP for the selected packet?

1) Ethernet II. This is the data link layer that includes the MAC addresses of the source and destination devices within the local network.

2) Internet Protocol version 4 (IPV4)
This is the network layer, responsible for identifying source and destination IP address.

3) Transmission control protocol. (TCP)
This is the transport layer, ensuring reliable transmission and providing ports to identify applications.

4) Hypertext Transfer Protocol (HTTP)
This is the application layer, where the actual web request (such as GET or POST) is visible.
Each layer wraps the data from the layer above, which is called encapsulation when the packet is received the layers are de-encapsulated in reverse order.

Q) What did you learn from this activity about how data travels over the internet?

⇒ This activity offered a hands-on understanding of how data travels through the internet using different Protocol layers.

Some Key takeaways :-

1) layer architecture - Data is encapsulated in layer (ethernet, IP, HTTP, TCP) during transmission and de-encapsulated at the receiver's end.

2) Protocol role :- Each protocol serves a specific function - for example, DNS resolves domain names, IP handles addressing and TCP ensures reliable delivery.

3) Traffic visibility - wireshark allows us to inspect the raw data being transmitted including requests to websites, responses from servers, and system interactions with network.

4) Security considerations: observing how easily HTTP data can be read underscores the importance of using HTTPS for secure communication.

5) Packet-level understanding - viewing each packet's structure gives a deeper appreciation of the technical details involved in something as simple as opening a web page.

1) What is the purpose of using filter in Wireshark?

Ans: Filters helps you focus on specific types of network traffic, making it easier to analyze and troubleshoot issue without being overwhelmed by all the data.

2) Which network interface did you use for capturing packets? Why?

An You would typically use the network interface that is connected to the internet or the local network you want to analyze. This allows you a capture relevant traffic.

Q3) Name at least 3 different protocols you observed during the capture.

→ Common protocols you might observe include
- HTTP (for web traffic)
- DNS (for domain name resolution)
- TCP (for reliable data transmission)

Q4) What is the IP address for your system (Source IP)?

Ans: You can find your system's IP address in Wireshark by looking at the source IP of any packet capture packet capture. It usually starts with "192.168." or "10.x" for local network.

Q5) What is one destination IP you communicate with during the capture?

Looks for any packet where your system's IP is the source. The destination IP will be the address

of the server or device your communicate with (eg: a website's IP)

6) Find a DNS query in your capture. What domain name was being looked up?

An Look for packets with the DNS protocol. You can find a query that shows a domain name like "example.com" being looked up.

07) Pick one HTTP packets and describe its source...

Ans Select an HTTP packets and check the details. The source IP is your system's IP, the destination IP is the web server's IP, and ~~most~~ the ~~non~~ Host name is usually found in the HTTP header.
eg: "Host: www.example.com"):

8) Which filter showed the most traffic? Why do you think that is?

Ans The HTTP filter often shows the most traffic because web browsing generates a lot of request and responses. Many user access website frequently leading to high HTTP ~~trapp~~ traffic.

9) Did you observe any encrypted traffic...?
Yes, you can tell if traffic is encrypted by looking for packets with the "TLS" or "SSL" protocol. These packet usually indicate secure connection, like HTTPS website.

10) What's the difference between 'ip.addr == x.x.x.x' and ip.src == x.x.x.x?

ip.addr == x.x.x.x : This filter shows packets where the IP address is either the source or destination.

ip.src == x.x.x.x : This filter shows only packets where the IP address is the source

11) FIND A TCP handshake (SYN, SYN-ACK, ACK)

→ A TCP handshake involve three steps:

1. SYN : your system sends a SYN packets to initiate a connection.
2. SYN - ACK : The server respond with a SYN-ACK packets to acknowledge the request.
3. Ack : your system sends an Ack packets to confirm the connection is established

12) What's one interesting thing you noticed in your network traffic?

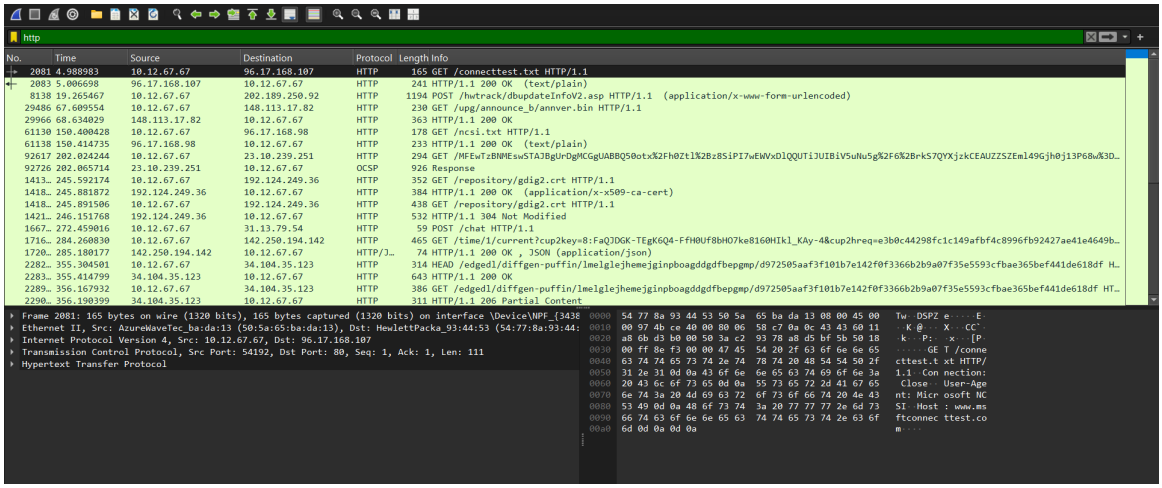
→ You might notice unusual traffic patterns, such as a high number of DNS queries, unexpected IP addresses communicating with your system, or a lot of encrypted traffic, which could indicate secure browsing or potential security concerns.

Assignment-12

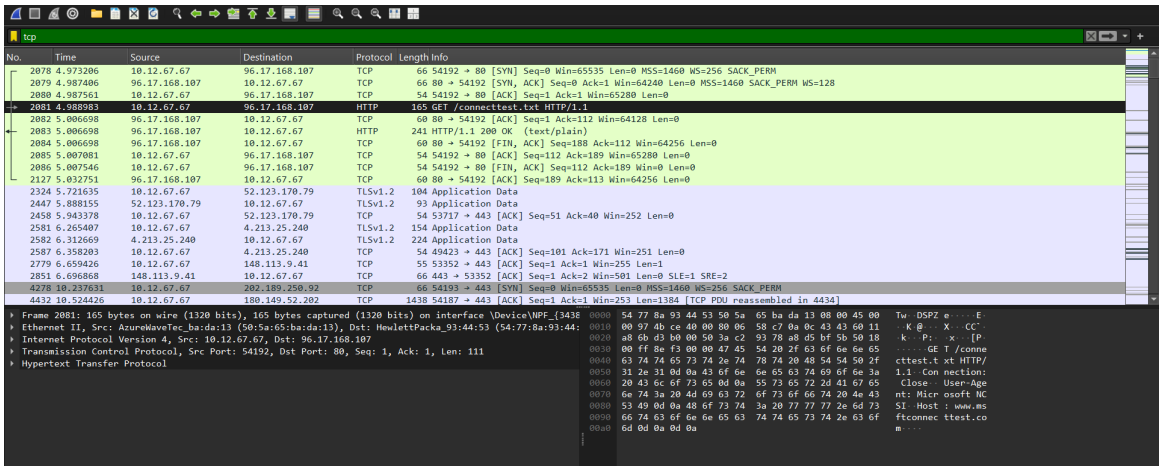
Wireshark display filters :-

1. Protocol-Based Filters

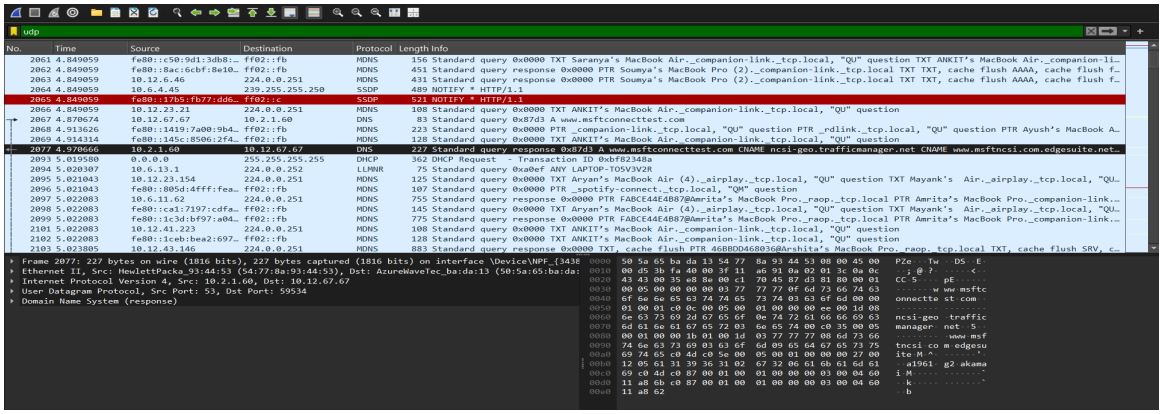
- **http – Shows only HTTP traffic**



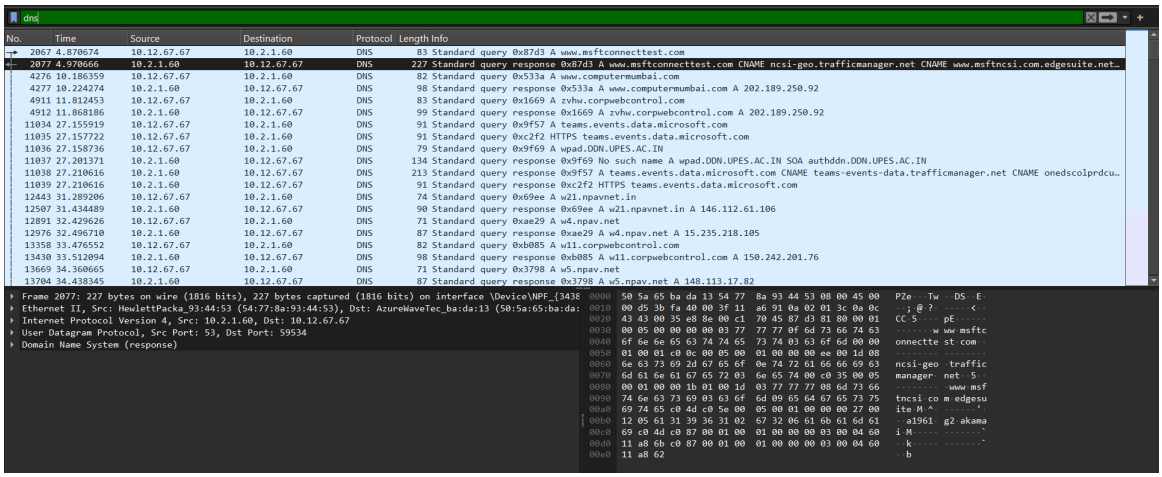
- **tcp – Displays only TCP packets**



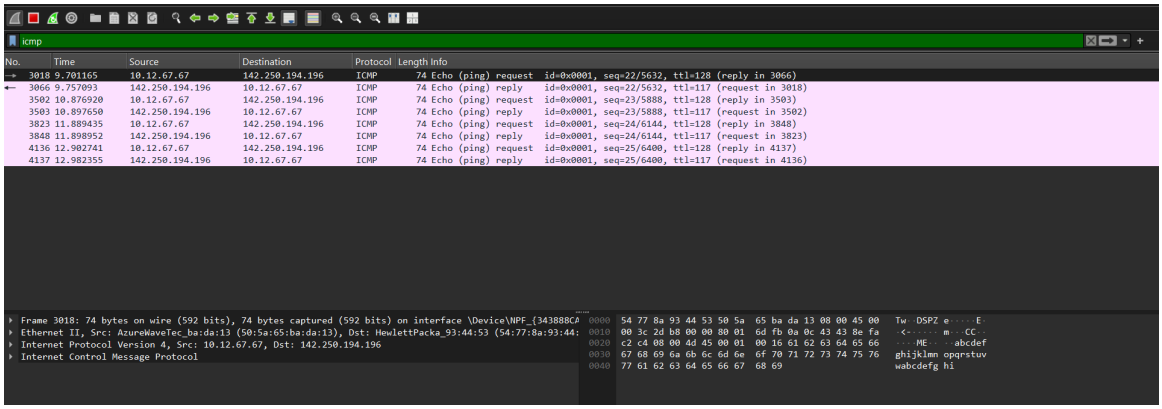
- **udp – Displays only UDP packets**



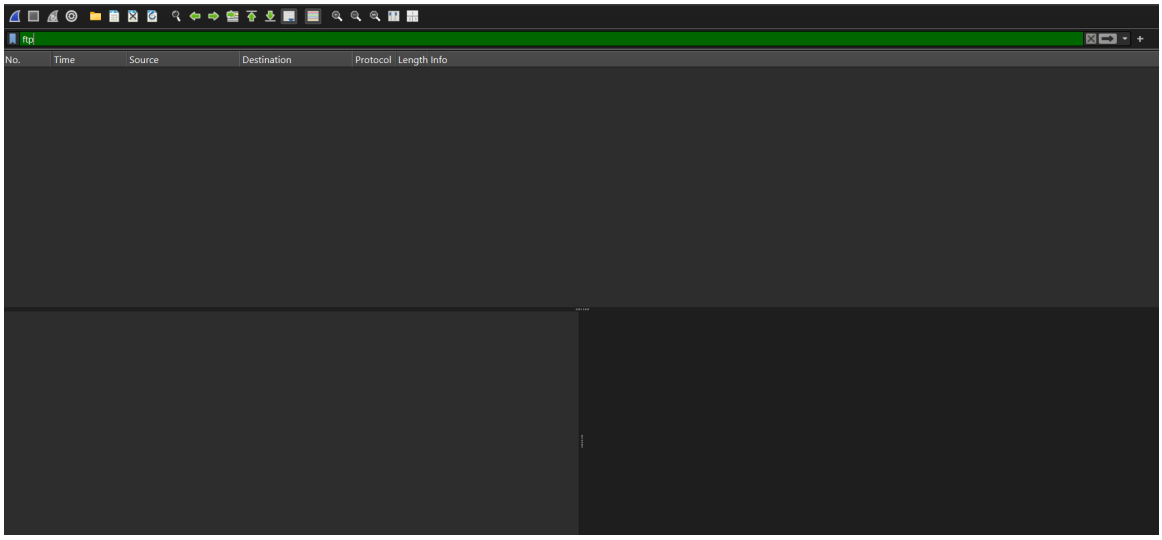
- **dns – Shows only DNS traffic**



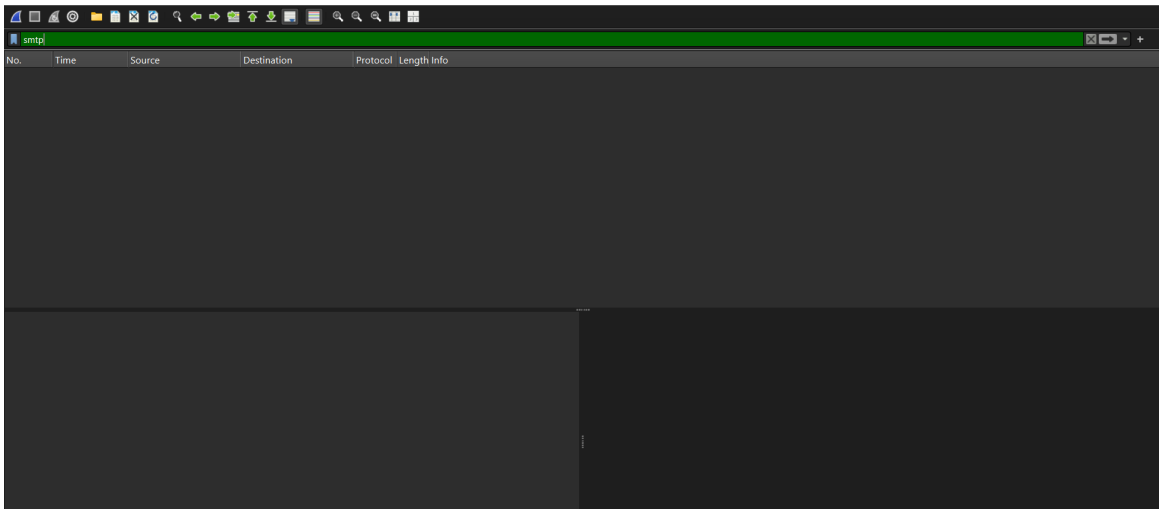
- **icmp – Filters ICMP packets (e.g., ping)**



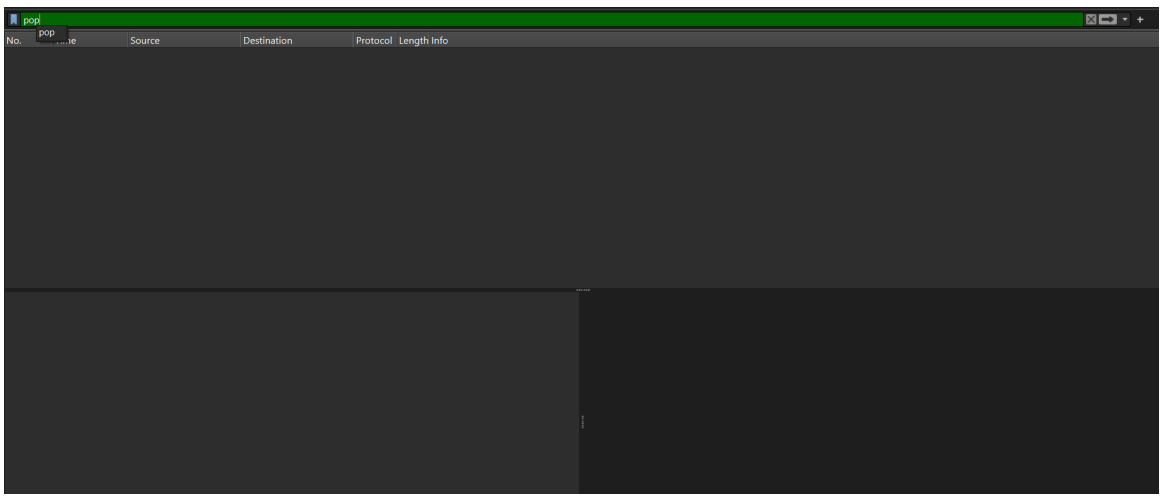
- **ftp – Filters FTP traffic**



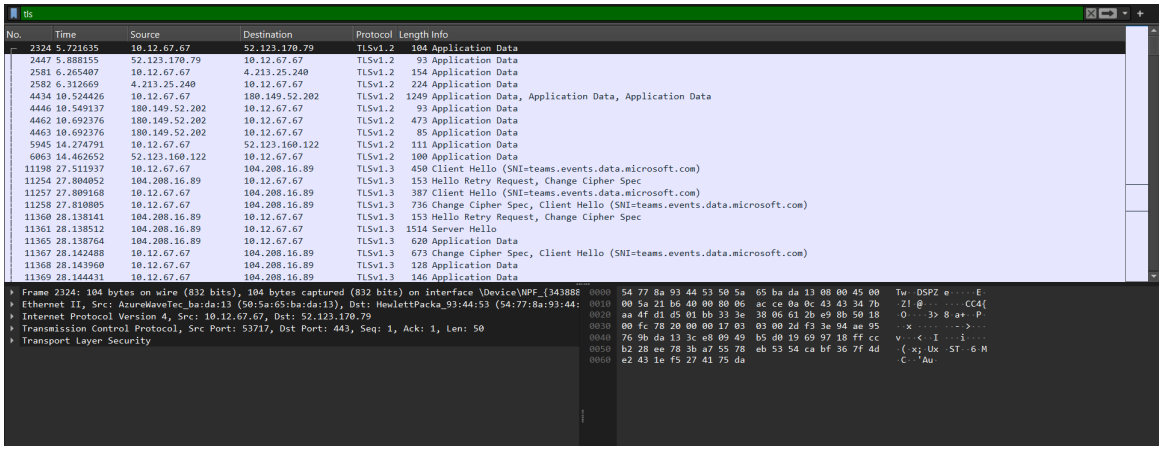
- **smtp – Filters SMTP (email sending)**



- **pop – Filters POP3 (email receiving)**

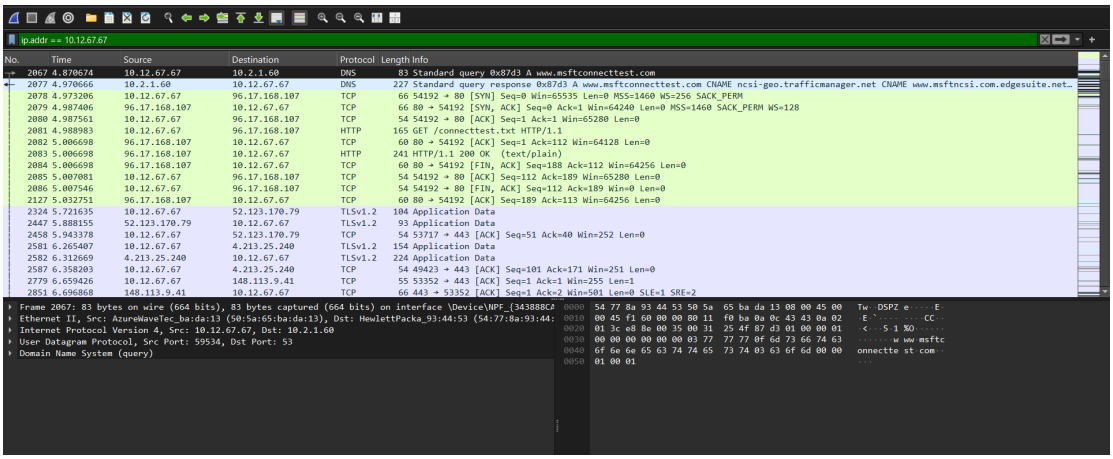


- **tls or ssl – Filters encrypted traffic (HTTPS)**

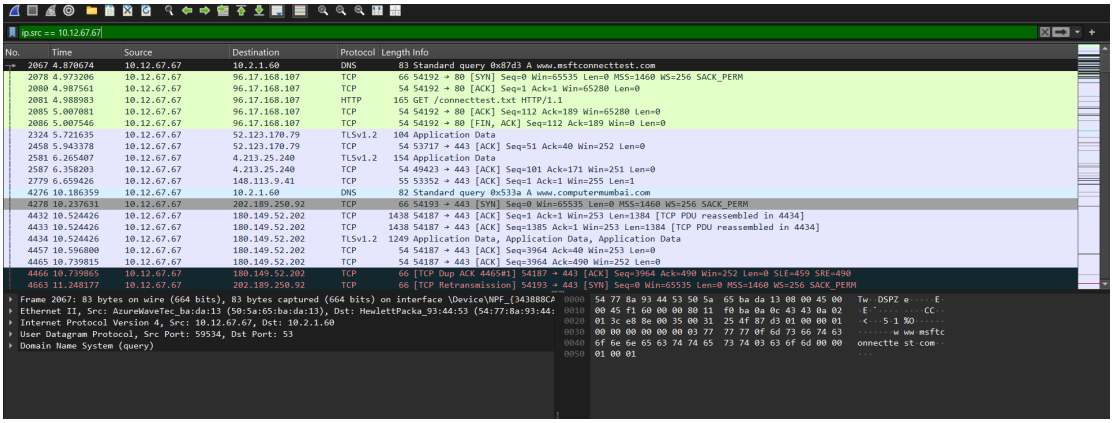


2. IP Address Filters

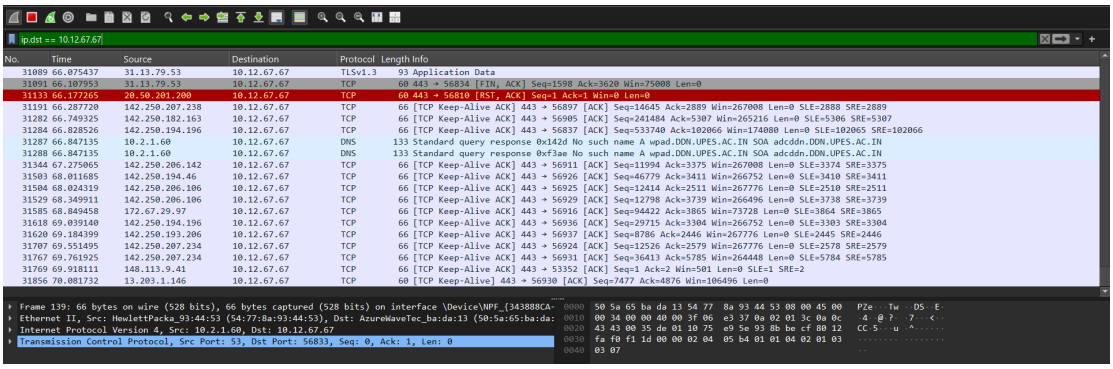
- **ip.addr == 192.168.1.1 – Shows all traffic to/from a specific IP**



- **ip.src == 192.168.1.1 – Traffic from a specific IP**

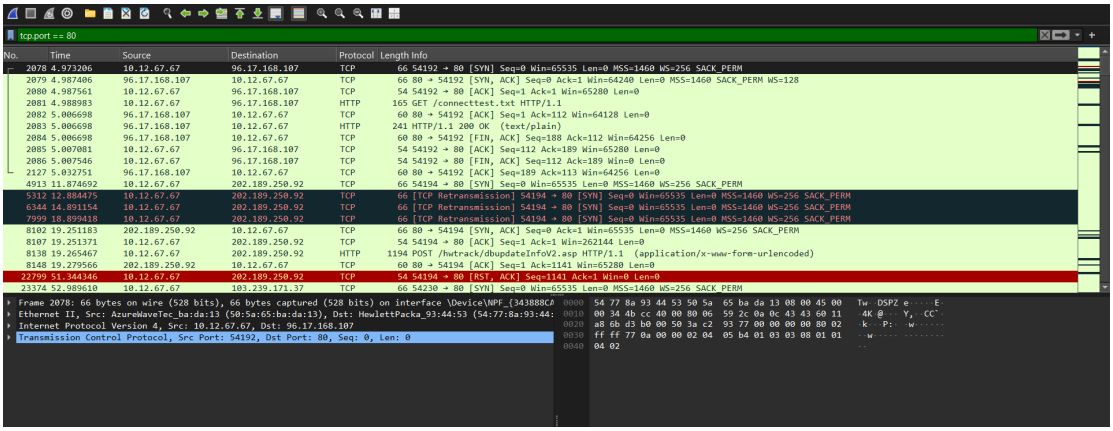


- **ip.dst == 8.8.8.8 – Traffic to a specific IP**

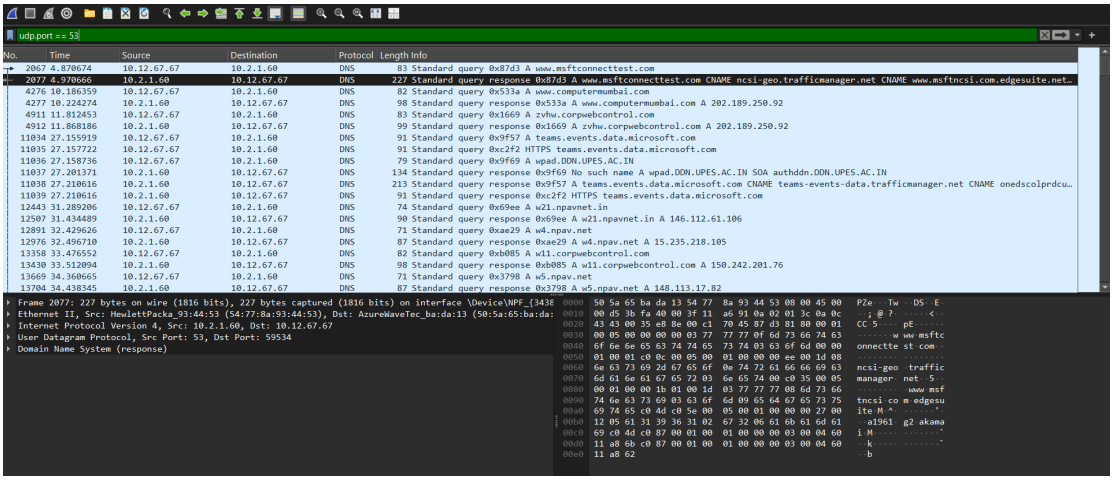


3. Port Filters

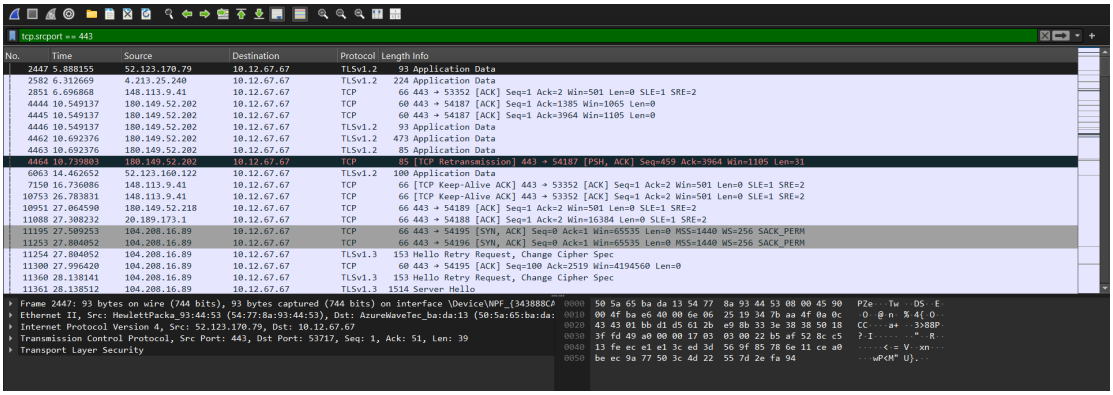
- **tcp.port == 80 – Shows HTTP traffic on port 80**



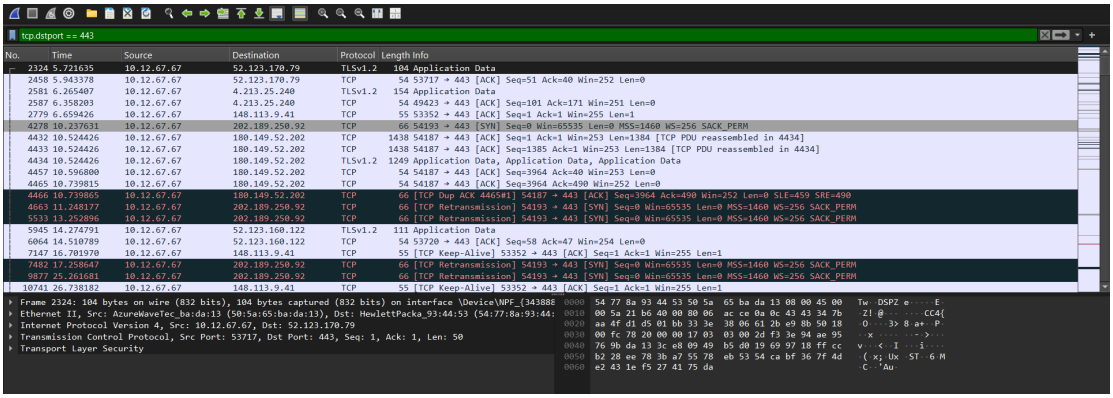
- **udp.port == 53 – Shows DNS traffic over UDP**



- **tcp.srcport == 443 – HTTPS traffic from server**

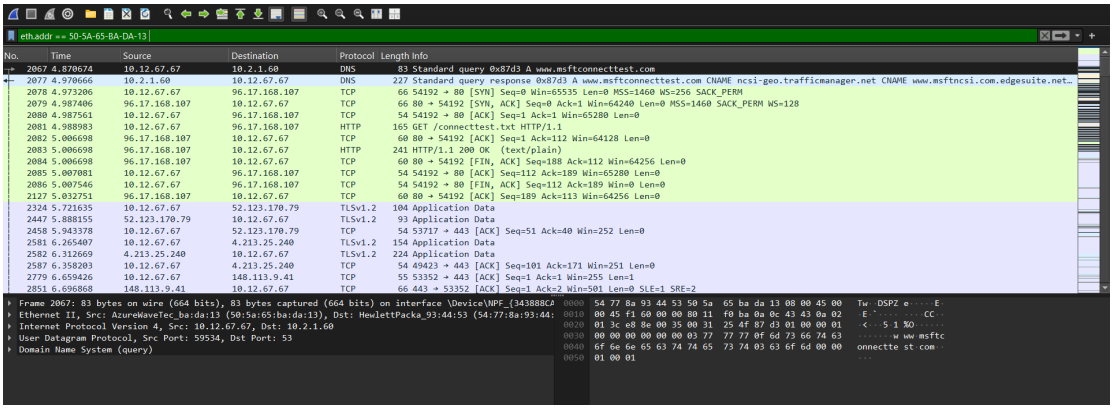


- **tcp.dstport == 443 – HTTPS traffic to server**

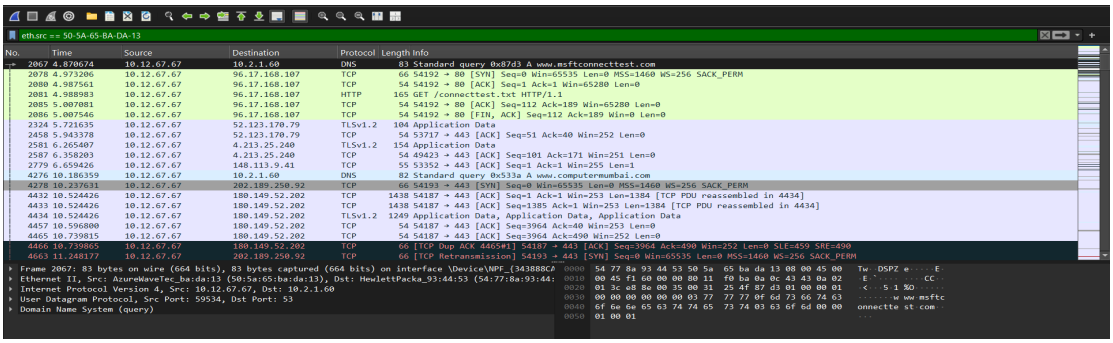


4. **MAC Address Filters**

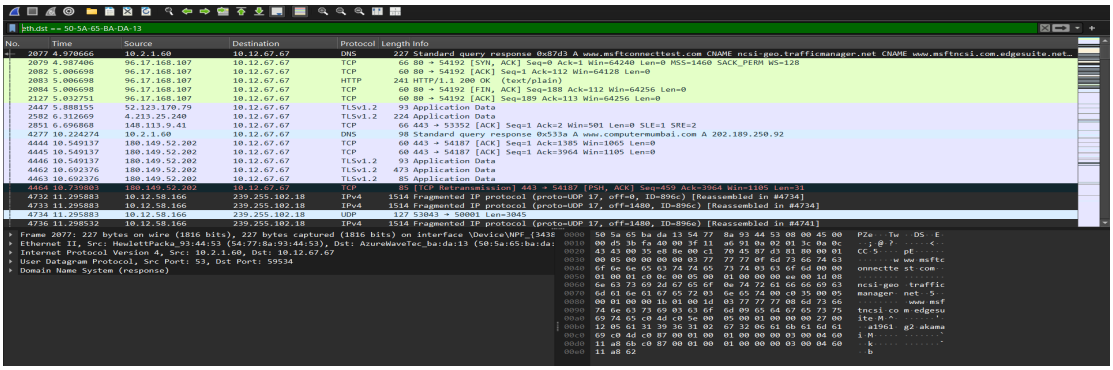
- **eth.addr == 00:11:22:33:44:55 – Traffic to/from a specific MAC address**



- **eth.src == 00:11:22:33:44:55 – Traffic from MAC**

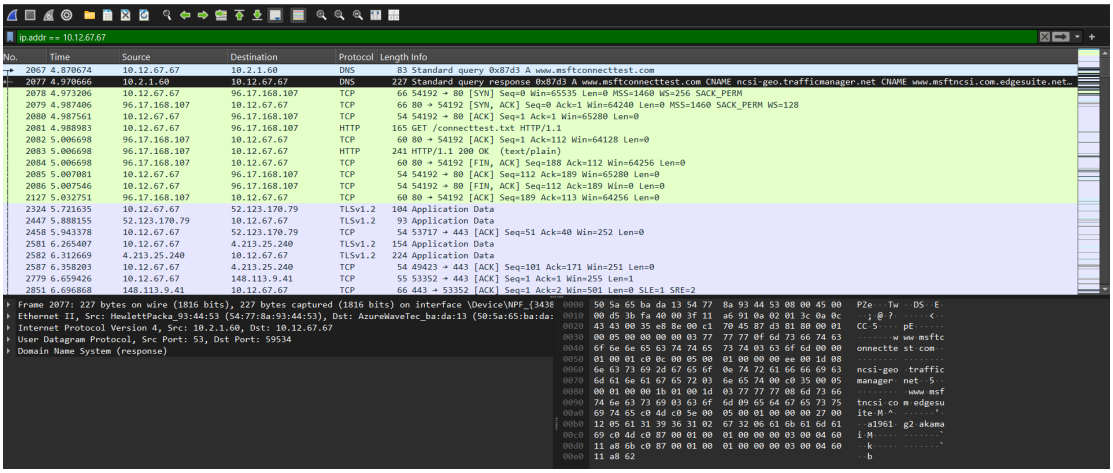


- **eth.dst == 00:11:22:33:44:55 – Traffic to MAC**



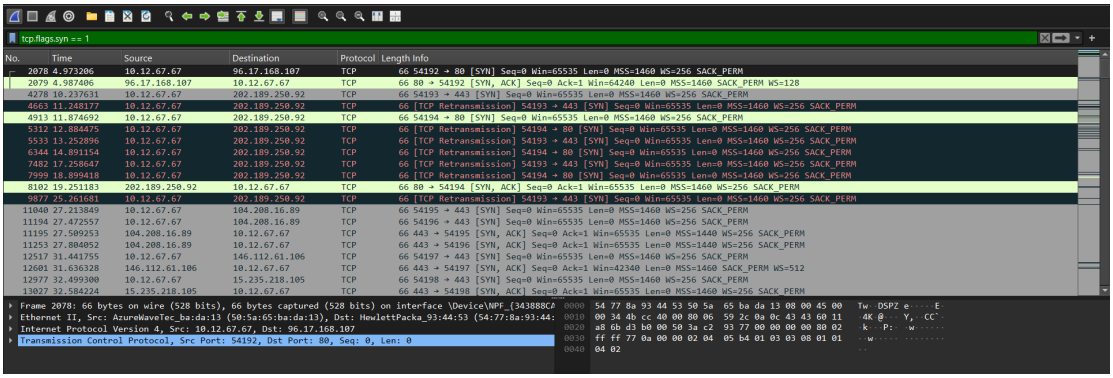
5. **Network Filters**

- **ip.addr == 192.168.0.0/24 – Traffic within a subnet**

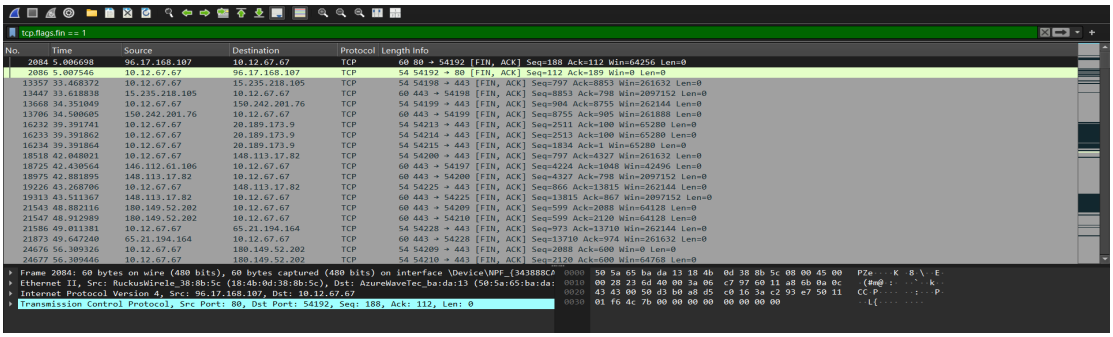


6. TCP Flags

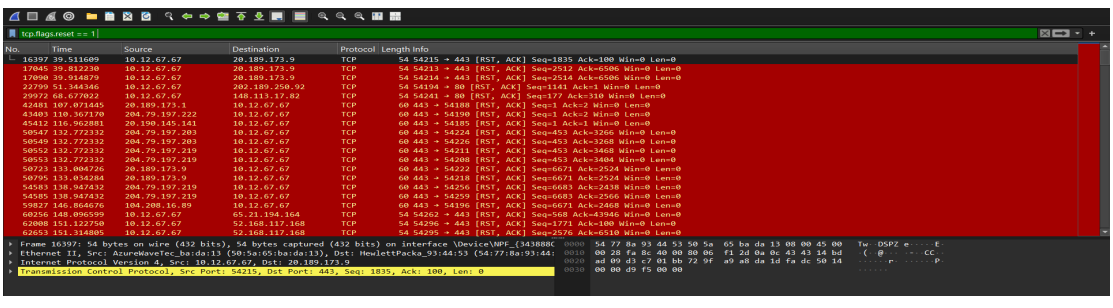
- **tcp.flags.syn == 1 – SYN packets (start of TCP handshake)**



- **tcp.flags.fin == 1 – FIN packets (end of session)**

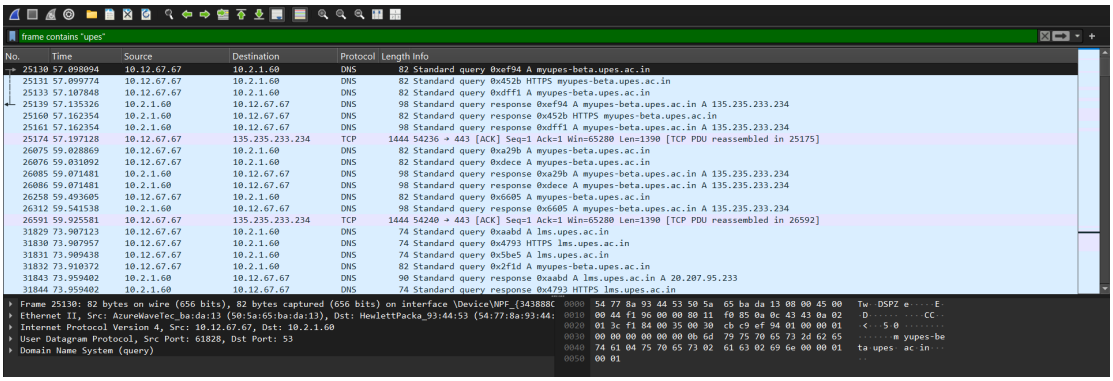


- **tcp.flags.reset == 1 – RST packets (reset connection)**

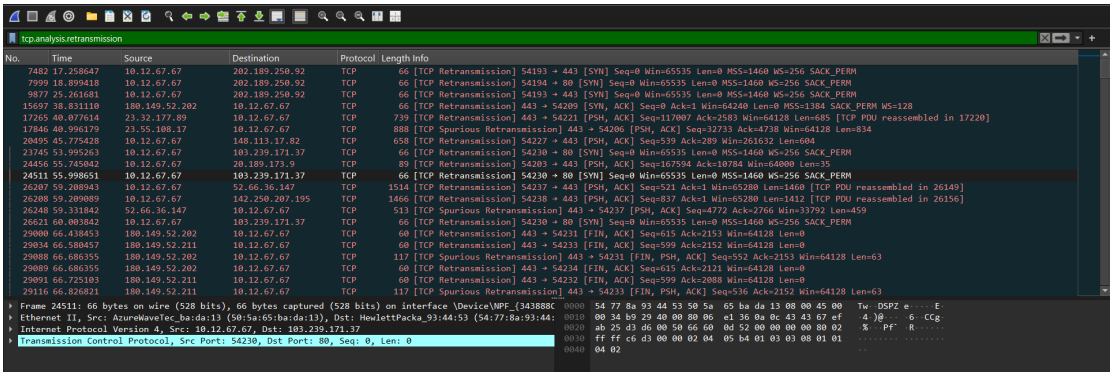


7. Other Useful Filters

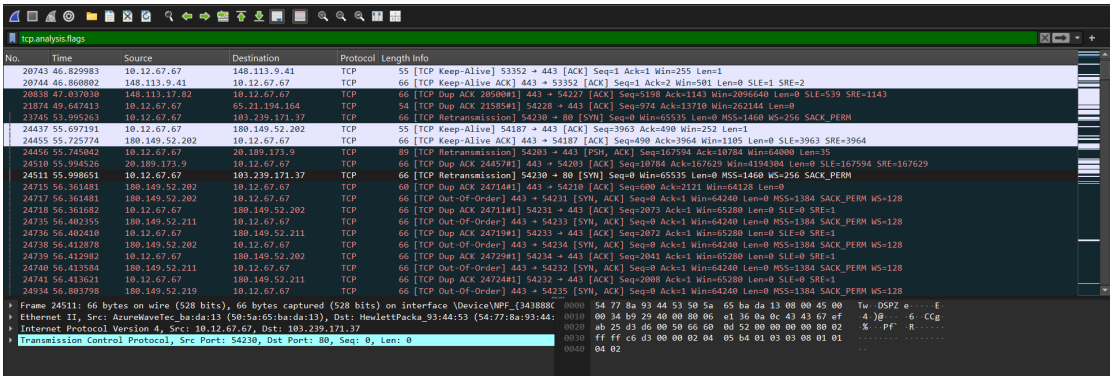
- **frame contains "text" – Packets containing specific text**



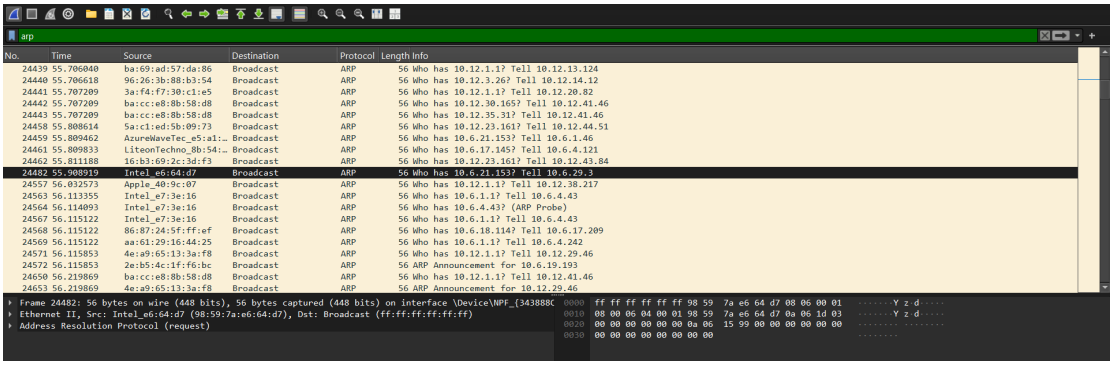
- **tcp.analysis.retransmission – Shows retransmitted packets**



- **tcp.analysis.flags – Flags abnormal TCP behavior**



- **arp – Filters ARP requests/replies**


arp
No. Time Source Destination Protocol Length Info
24439 55.706040 ba:69:ad:57:da:86 Broadcast ARP 56 Who has 10.12.1.1? Tell 10.12.13.124
24440 55.706618 96:26:3b:88:b3:54 Broadcast ARP 56 Who has 10.12.3.26? Tell 10.12.14.12
24441 55.707209 3a:f4:f7:30:c1:e5 Broadcast ARP 56 Who has 10.12.1.1? Tell 10.12.20.82
24442 55.707209 ba:cc:e8:8b:58:d8 Broadcast ARP 56 Who has 10.12.30.165? Tell 10.12.41.46
24443 55.707209 ba:cc:e8:8b:58:d8 Broadcast ARP 56 Who has 10.12.35.31? Tell 10.12.41.46
24458 55.808614 5a:c1:ed:5b:09:73 Broadcast ARP 56 Who has 10.12.23.161? Tell 10.12.44.51
24459 55.809462 AzureWaveTec_e5:a1:… Broadcast ARP 56 Who has 10.6.21.153? Tell 10.6.1.46
24461 55.809833 LiteonTechno_8b:54:… Broadcast ARP 56 Who has 10.6.17.145? Tell 10.6.4.121
24462 55.811188 16:b3:69:2c:3d:f3 Broadcast ARP 56 Who has 10.12.23.161? Tell 10.12.43.84
24482 55.908919 Intel_e6:64:d7 Broadcast ARP 56 Who has 10.6.21.153? Tell 10.6.29.3
24557 56.032573 Apple_40:9c:07 Broadcast ARP 56 Who has 10.12.1.1? Tell 10.12.38.217
24563 56.113355 Intel_e7:3e:16 Broadcast ARP 56 Who has 10.6.1.1? Tell 10.6.4.43
24564 56.114093 Intel_e7:3e:16 Broadcast ARP 56 Who has 10.6.4.43? (ARP Probe)
24567 56.115122 Intel_e7:3e:16 Broadcast ARP 56 Who has 10.6.1.1? Tell 10.6.4.43
24568 56.115122 86:87:24:5f:ff:ef Broadcast ARP 56 Who has 10.6.18.114? Tell 10.6.17.209
24569 56.115122 aa:61:29:16:44:25 Broadcast ARP 56 Who has 10.6.1.1? Tell 10.6.4.242
24571 56.115853 4e:a9:65:13:3a:f8 Broadcast ARP 56 Who has 10.12.1.1? Tell 10.12.29.46
24572 56.115853 2e:b5:4c:1f:f6:bc Broadcast ARP 56 ARP Announcement for 10.6.19.193
24650 56.219869 ba:cc:e8:8b:58:d8 Broadcast ARP 56 Who has 10.12.1.1? Tell 10.12.41.46
24653 56.219869 4e:a9:65:13:3a:f8 Broadcast ARP 56 ARP Announcement for 10.12.29.46
Frame 24482: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF_{343888C
Ethernet II, Src: Intel_e6:64:d7 (98:59:7a:e6:64:d7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
0000 ff ff ff ff ff ff 98 59 7a e6 64 d7 08 06 00 01 ·····Y z·d·····
0010 08 00 06 04 00 01 98 59 7a e6 64 d7 0a 06 1d 03 ·····Y z·d·····
0020 00 00 00 00 00 00 0a 06 15 99 00 00 00 00 00 00 ···· ····
0030 00 00 00 00 00 00 00 00 ····

Name - Kshitij Chandrakar
SAP - 500124827

14

Page No :

Set-B

Q1 What is purpose of CRC in Error Detection Explain with Ex.

Q2 Write a Program to calculate CRC for the Bitstream 100100 with poly 1011

Q1 => CRC (Cyclic Redundancy check) is a Method Designed to transmit Data over the Internet. It does so by making the BitStream Exactly Divisible by the Generator Polynomial by padding the Remainder of the Division to the Bitstream.

=> the layout is BitStream CRC

=> the length of the CRC is Equal to the Degree of the Polynomial

=> It can only Detect Single Bit Errors.

for Ex. the CRC of 100100 with Generator Poly 1011 is

Generator Poly = $x^3 + x + 1$

∴ length of CRC = 3

Padded Message = 100100 000

Page No:

Binary Division

```
1011 | 100100.000
       1011
       ----
       001000
        1011
        ----
        001100
         1011
         ----
         01110
          1011
          ----
          0101
```

Remainder is 101
∴ CRC is 101

Transmitted Message = 100100.101

Consider this Message is Recieved with an Error (Single Bit)

Let Recieved Message = 100101 101
↑
Erroneous Bit

To check if Error is present
we do xor Division with the Generator polynomial
If the Remainder is Non-zero then the Data is Erroneous.

Page No :

XOR Division :-

```
1011 | 100 10ϕ :101
       101  1
       ----
       0010 0ϕ
            1011
            ----
            00 1010
               1011
               ----
               00 01 1   ← Non zero Remainder
```

∴ the Recieved Bit is Errored.

Code

```
def xor(a,b):
    result = []
    for i in range(len(a)):
        if a[i] == b[i]:
            result.append("0")
        else
            result.append("1")
    return "".join(result)

def div(divd, divs):
    c = len(divs)
    tmp = divd[0:c]

    While (c < len(divd)):
        if temp[0] == '1':
            temp = xor(temp, divs) + divd[
```

```
        Else:
            temp = xor('0'*c, temp) + divd[c:]
        c += 1
    if temp[0] == '1':
        temp = xor(temp, divs)
    Else
        temp = xor('0'*c, divs)

    return temp

def Encoding(data, Poly):

    Padded_Data = data + '0'*(len(Poly)-1)
    Remainder = div(data, Poly)
    Encoded = data + Remainder
    Print("Data is", Data)
    Print("CRC is", Remainder)
    Print("Encoded is", Encoded)

data = "100100"
Poly = "1011"
Encoding(data, Poly)
```