List of CN experiment

1. Use basic networking commands in Command Prompt

- ping google.com
- tracert google.com
- nslookup
- netstat -ano

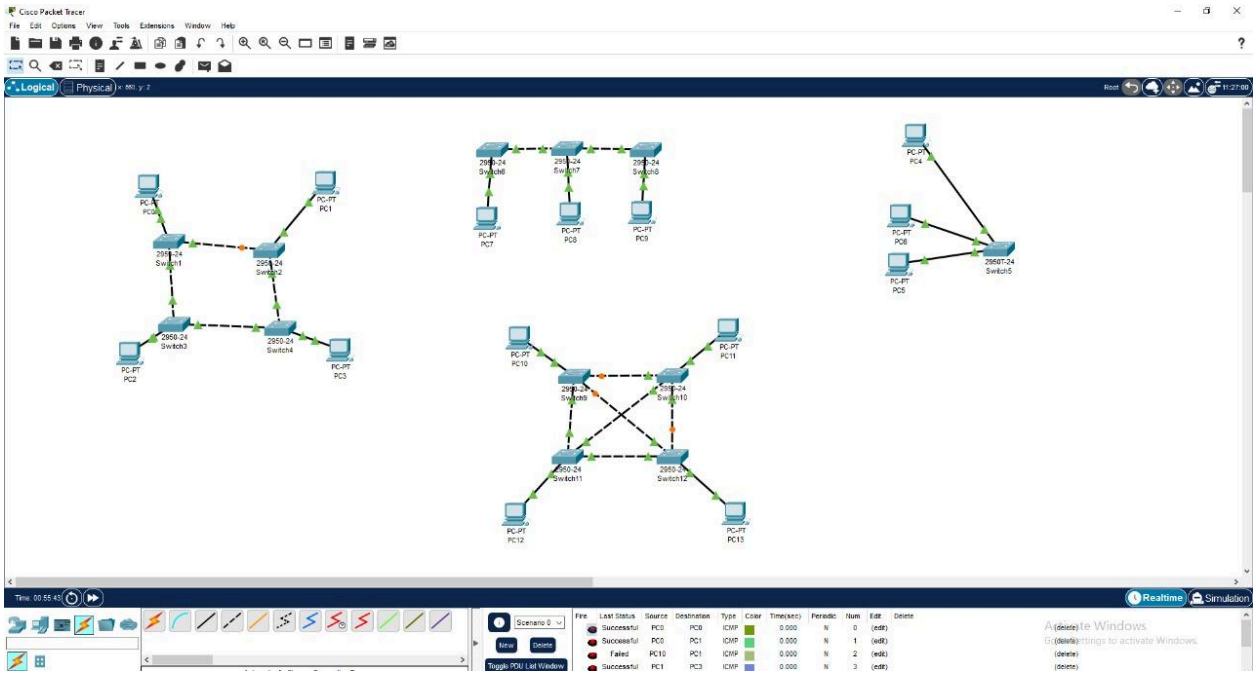
[**Windows Command:** `netstat -ano` (`-a` for all connections, `-n` for numerical addresses, `-o` to show the process identifier)]

- ipconfig /all
- route print

[**Windows Command:** `route print` (displays the routing table), `route add`, `route delete`, `route change` (to modify the routing table).]

2. Build a simple network topology and configure it for static routing protocol using packet
tracer. Setup a network and configure IP addressing, subnetting, masking.

Use packet Tracer with the photo



3. Use Wire shark to understand the operation of TCP/IPlayers:
● Ethernet Layer: Frame header, Frame size etc.
● Data Link Layer: MAC address, ARP (IP and MAC address binding)
● Network Layer: IP Packet (header, fragmentation),ICMP (Query and Echo)
● Transport Layer: TCP Ports, TCP handshake segments etc.
● Application Layer: DHCP, FTP, HTTP header formats

Install Wireshark:

- Your theory section already gives you the Linux commands. Since you're on Linux, open your terminal.
- Type: `sudo su` and press Enter. This will ask for your password and give you administrator privileges (needed for Wireshark to capture network traffic).
- Type: `sudo apt-get install wireshark` and press Enter. This will download and install Wireshark. You might be asked to confirm during the installation – just say "yes" (usually by pressing 'Y' and then Enter).
- Once installed, you can usually find Wireshark in your applications menu.

Start Wireshark:

- Open Wireshark. You should see a window similar to "Figure 1.1" in your notes, listing your network interfaces (like your Wi-Fi or Ethernet connection).

Choose Your Network Interface:

- Decide which network connection you want to monitor. If you're using Wi-Fi, select that interface. If you're connected with a cable, select

the Ethernet interface (it might be called `eth0`, `enp0sX`, or something similar).
- **Click** on the interface you want to capture traffic from.

Start Capturing:

- Once you've selected an interface, click the **shark fin icon** (usually in the top left corner) to start capturing network traffic. You'll see a stream of packets appearing in the main Wireshark window (like "Figure 1.2").

Generate Network Traffic (This is Key!):

- To see the different protocols in action, you need to *use* them. Here are some things you can do *while Wireshark is capturing*:
    - **Browse a website:** Open your web browser (like Chrome, Firefox) and go to a website (e.g., `www.google.com`). This will generate HTTP traffic.
    - **Download a file (small):** Download a small file using your browser or an FTP client (if you have one set up). This will generate HTTP or FTP traffic.
    - **Check your email:** If you use an email client, sending or receiving emails will generate traffic (likely using protocols like SMTP, POP3, or IMAP).
    - **Use the `ping` command:** Open another terminal and use the `ping` command to send echo requests (ICMP). For example: `ping google.com`.
    - **Let DHCP happen:** If you recently connected to a network, you might have seen a DHCP request. If not, you could try disconnecting and reconnecting to your Wi-Fi to trigger a DHCP exchange.

Stop Capturing:

- After you've generated some network activity, click the **red square icon** in Wireshark to stop the capture.

Examine the Captured Packets:

- Now you'll see a list of captured packets. Each row represents a single network packet.
- **Selecting a Packet:** Click on any packet in the top window to see its details in the lower two panes.
    - **Middle Pane (Packet Details):** This pane shows you the different layers of the selected packet (Ethernet II, Internet Protocol Version 4/6, TCP/UDP, Application Layer Protocol). Click on the small arrows next to each layer to expand and see the header fields.
    - **Bottom Pane (Packet Bytes):** This pane shows the raw data of the packet in hexadecimal and ASCII format.

Focus on Each Layer (as per your aim):

- **Ethernet Layer:**
    - Select any packet.
    - In the middle pane, look for the "Ethernet II" section.
    - **Frame Header:** You'll see information like the Destination MAC Address, Source MAC Address, and the EtherType (which indicates the next layer protocol, like IP).
    - **Frame Size:** While not explicitly a field, the total length of the packet as displayed in the top window gives you an idea of the frame size.
- **Data Link Layer (ARP):**
    - To see ARP (Address Resolution Protocol) in action, you might need to filter your capture. In the filter bar at the top, type `arp` and press Enter.

    - Look for ARP Request and ARP Reply packets.
    - Examine the ARP packet details. You should see fields related to the Sender MAC address, Sender IP address, Target MAC address, and Target IP address. This shows how devices find the MAC address associated with an IP address on the local network.
- **Network Layer (IP):**
    - Remove the `arp` filter (or start a new capture and generate traffic again).
    - Select a packet that uses IP (look for "Internet Protocol Version 4" or "Internet Protocol Version 6" in the middle pane).
    - **IP Packet Header:** Expand the IP layer details. You'll see fields like Source IP Address, Destination IP Address, Protocol (TCP, UDP, ICMP), Time to Live (TTL), and potentially "Fragmentation" related fields if a large packet was broken down.
    - **ICMP:** Filter for `icmp`. You should see "Echo (ping) request" and "Echo (ping) reply" packets if you used the `ping` command. Examine the ICMP header, noting the Type and Code fields.
- **Transport Layer (TCP):**
    - Filter for `tcp`.
    - Look for the **TCP Three-way Handshake:** This is the sequence of SYN, SYN-ACK, and ACK packets that establish a TCP connection. You should see packets with the "Flags" field indicating SYN, SYN-ACK, and ACK.
    - **TCP Ports:** Examine the "Source Port" and "Destination Port" fields in the TCP header. These identify the applications or services communicating. Common ports for HTTP are 80 and 443.

    - **TCP Handshake Segments:** The SYN, SYN-ACK, and ACK packets are the handshake segments. Look at their sequence numbers and acknowledgment numbers.
- **Application Layer:**

    - **DHCP:** Filter for `dhcp`. Look at the DHCP Request, DHCP Offer, DHCP Acknowledge packets. Examine the details to see information like the IP address being offered, the DHCP server, and lease duration.
    - **FTP:** If you generated FTP traffic, filter for `ftp`. Look at the commands (like `USER`, `PASS`, `RETR`) and responses in the packet details.
    - **HTTP:** Filter for `http` or `tcp.port == 80` or `tcp.port == 443`. Look at the HTTP GET and POST requests and the HTTP responses (like 200 OK). Expand the "Hypertext Transfer Protocol" section to see the HTTP header fields (e.g., Host, User-Agent, Content-Type).

4. Implement the Hamming code using C++

Code:

```
#include<iostream>
using namespace std;
int main() {
int data[10];
int dataatrec[10],c,c1,c2,c3,i;
cout<<"Enter 4 bits of data one by one\n";
cin>>data[0];
cin>>data[1];
cin>>data[2];
cin>>data[4];
//Calculation of even parity
data[6]=data[0]^data[2]^data[4];
```

```
data[5]=data[0]^data[1]^data[4];
data[3]=data[0]^data[1]^data[2];
cout<<"\nEncoded data is\n";
for(i=0;i<7;i++)
cout<<data[i];
cout<<"\n\nEnter received data bits one by one\n";
for(i=0;i<7;i++)
cin>>dataatrec[i];
c1=dataatrec[6]^dataatrec[4]^dataatrec[2]^dataatrec[0];
c2=dataatrec[5]^dataatrec[4]^dataatrec[1]^dataatrec[0];
c3=dataatrec[3]^dataatrec[2]^dataatrec[1]^dataatrec[0];
c=c3*4+c2*2+c1 ;
if(c==0) {
cout<<"\nNo error while transmission of data\n";
}
else {
cout<<"\nError on position "<<c;
cout<<"\nData sent : ";
for(i=0;i<7;i++)
cout<<data[i];
cout<<"\nData received : ";
for(i=0;i<7;i++)
cout<<dataatrec[i];
cout<<"\nCorrect message is\n";
//if errorneous bit is 0 we complement it else vice versa
if(dataatrec[7-c]==0)
dataatrec[7-c]=1;
else
dataatrec[7-c]=0;
for (i=0;i<7;i++) {
cout<<dataatrec[i];
}
}
return 0;
}
```

Output me put :
1
0
1
0
Then put
1
0
1
0
0
1
0
[No transmission in error is msg]

5. Perform Remote login using Telnet server using Cisco Packet Tracer

Perform in Packet Tracer
1 Router (or Switch)

1 PC

Straight-through cable (PC ↔ Router/Switch)
Click on Router/Switch → CLI.

enable
configure terminal

interface gigabitethernet 0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit

line vty 0 4
password cisco
login
exit

enable password class

service password-encryption

#Click the PC → Desktop → IP Configuration:
#IP Address: 192.168.1.2

#Subnet Mask: 255.255.255.0

#Default Gateway: 192.168.1.1

On the PC → Desktop → Command Prompt:
telnet 192.168.1.1

any problem :
[https://chatgpt.com/share/680fd3af-eb98-8003-a264-5a2444731e1d]

6. To perform basic configuration tasks on Cisco routers and switches using the IOS CLI, including setting hostnames, configuring and encrypting enable passwords, verifying configurations, and saving them to startup configuration. (DAY 4 -LAB YOUTUBE)

Perform in Packet Tracer

7. To analyze network traffic using Packet Tracer's simulation mode by observing various protocol data units (PDUs) across different

layers of the OSI and TCP/IP models, including STP, OSPF, and DHCP operations. (DAY 3 -LAB YOUTUBE)

Perform in Packet Tracer

8. To analyze ARP and ICMP message flow during network communication, learn MAC address table population in switches, and practice clearing dynamic MAC address entries using simulation and real-time modes in Packet Tracer. (DAY 6 – LAB YOUTUBE)

Perform in Packet Tracer

9. To design and implement a network topology using appropriate copper (straight-through and crossover) and fiber-optic connections between PCs, switches, and routers, considering transmission standards and cable selection based on device types and distance limitations. (DAY2 LAB-youtube)

Perform in Packet Tracer

10. Set up multiple IP addresses on a single LAN. Using netstat and route commands of Linux, do the following:
• View current routing table
• Add and delete routes
• Change default gateway Perform packet filtering by enabling IP forwarding using IPtables in Linux.
ip addr show

ifconfig

ip addr show ens33

sudo sysctl -w net.ipv4.ip_forward=1 net.ipv4.ip_forward=1

sudo nano /etc/sysctl.conf

sudo sysctl -p net.ipv4.ip_forward=1

sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE

sudo iptables -A FORWARD -s [ip address] -j ACCEPT

sudo iptables -L -v -n

ping [any new ip]

netstat -rn

sudo ip route add [old ip] via [new ip] dev ens33

ip route show

sudo ip route delete [old ip] via [new ip] dev ens33

ip route show

sudo ip route add default via [new ip] dev ens33

ip route show

sudo ip route del default via [new ip] dev ens33

ip route show

ping 8.8.8.8

traceroute 8.8.8.8

11. Perform File Transfer and Access using FTP.

```
sudo -i

echo 1 > /proc/sys/net/ipv4/ip_forward

sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE

sudo iptables -A FORWARD -s 192.168.253.128/24 -j ACCEPT

sudo iptables -L -v -n

sudo ip route add [ip address dalo via'ip a' - inet wala] dev ens33

sudo apt install iptables-persistent

sudo netfilter-persistent save
```