Ex. No. 06 **Packet Sniffing**

Date: 01-09-2025

Aim:

The capture and display ethernet frame information including destination MAC address, source MAC address, and protocol type using raw sockets in python.

Algorithm:

1. Get the host ip address of the current machine.

2. Create a new socket with AF_INET and sock_RAW to capture packets.

3. Bind the scoket options to include IP headers and enable promiscuous continualls form the socket.

4. Receive packets continually from the socket.

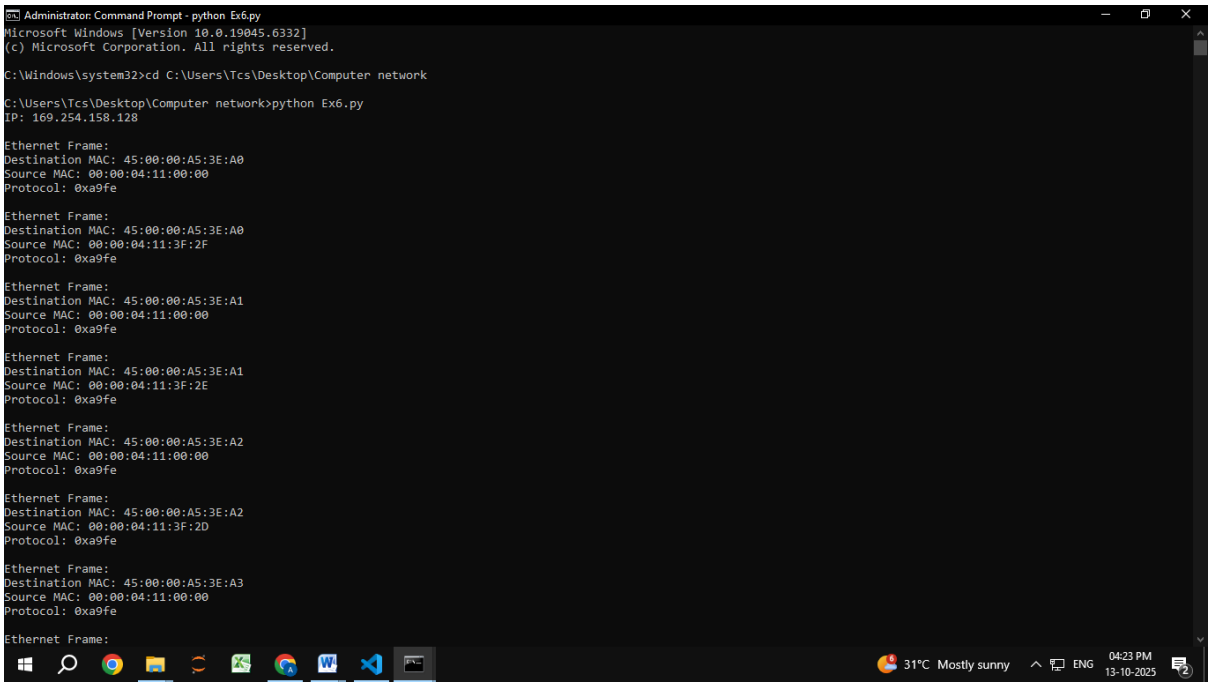
5. Extract ethernet froma data by unpackeong the first 14 bytes.

Program:

```
import socket
import struct

def main():
    # Get host IP address
    host = socket.gethostbyname(socket.gethostname())
    print('IP: {}'.format(host))

    # Create raw socket and bind it
    conn = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_IP)
    conn.bind((host, 0))

    # Include IP headers
    conn.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)

    # Enable promiscuous mode (Windows only)
```

```
    conn.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)

    while True:
        raw_data, addr = conn.recvfrom(65536)
        dest_mac, src_mac, eth_proto, data = ethernet_frame(raw_data)

        print('\nEthernet Frame:')
        print(f'Destination MAC: {dest_mac}')
        print(f'Source MAC: {src_mac}')
        print(f'Protocol: {eth_proto}')

def ethernet_frame(data):
    dest_mac, src_mac, proto = struct.unpack('!6s6sH', data[:14])
    return get_mac_addr(dest_mac), get_mac_addr(src_mac), eth_protocol(proto), data[14:]

def get_mac_addr(bytes_addr):
    mac_addr = ':'.join(format(b, '02x') for b in bytes_addr)
    return mac_addr.upper()

def eth_protocol(proto):
    protocols = {
        0x0800: "IPv4",
        0x0806: "ARP",
        0x86DD: "IPv6",
    }
    return protocols.get(proto, hex(proto))

if __name__ == "__main__":
    main()
```

Output:



Result:

Thus, the program to capture and display Ethernet frame details using raw sockets was successfully implemented and executed.