NAME: PRATIBA.S DATE:1.09.25 ROLL NO.:241901080

# EXERCISE 6

**IMPLEMENT PACKET SNIFFING USING RAW SOCKETS IN PYTHON**

**AIM:**

To implement a simple packet sniffer(raw sockets) that prints destination MAC,source MAC and protocol for each ethernet frame.

**ALGORITHM:**

1. Create raw socket, bind to host IP
2. Enable IP address headers inclusion and turn on promiscuous mode
3. Loop recvfrom(), slice first 14 bytes, unpack destination/source MAC (6+6) and proto (2)
4. Format and print MACs and protocol

**CODE:**

import socket

import struct

import binascii

import textwrap

def main():

# Get host

host = socket.gethostbyname(socket.gethostname())

print('IP: {}'.format(host))

# Create a 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

conn.ioctl(socket.SIO\_RCVALL, socket.RCVALL\_ON)

while True:

# Recive data

raw\_data, addr = conn.recvfrom(65536)

# Unpack data

dest\_mac, src\_mac, eth\_proto, data = ethernet\_frame(raw\_data)

print('\nEthernet Frame:')

print("Destination MAC: {}".format(dest\_mac))

print("Source MAC: {}".format(src\_mac))

print("Protocol: {}".format(eth\_proto))

# Unpack ethernet frame

def ethernet\_frame(data):

dest\_mac, src\_mac, proto = struct.unpack('!6s6s2s', data[:14])

return get\_mac\_addr(dest\_mac), get\_mac\_addr(src\_mac), get\_protocol(proto), data[14:]

# Return formatted MAC address AA:BB:CC:DD:EE:FF

def get\_mac\_addr(bytes\_addr):

bytes\_str = map('{:02x}'.format, bytes\_addr)

mac\_address = ':'.join(bytes\_str).upper()

return mac\_address

# Return formatted protocol ABCD

def get\_protocol(bytes\_proto):

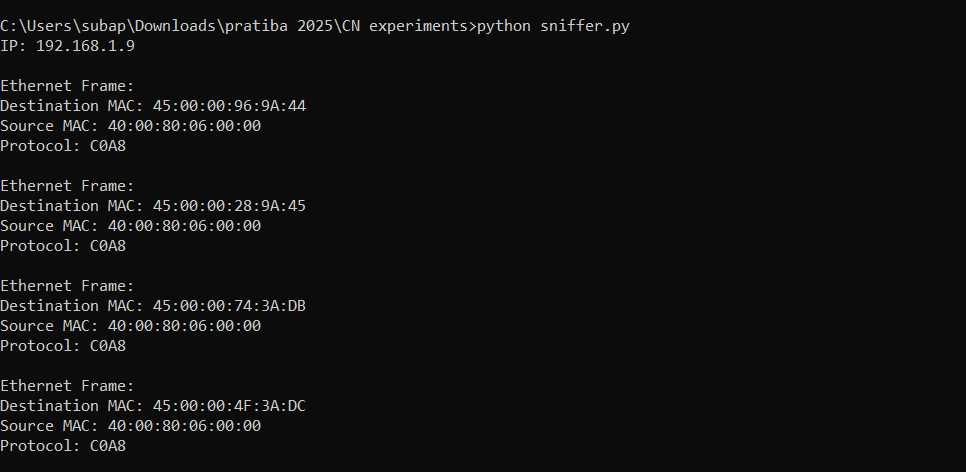
bytes\_str = map('{:02x}'.format, bytes\_proto)

protocol = ''.join(bytes\_str).upper()

return protocol

main()

**OUTPUT:**

****

**RESULT:**

The packet sniffer successfully captured and displayed Ethernet frames, showing the destination MAC, source MAC and protocol type of each packet