

# PACKET SNIFFING USING RAW SOCKETS

## AIM:

To capture and display Ethernet frame details (Destination MAC, Source MAC, and Protocol) from network traffic using raw sockets.

## ALGORITHM:

1. Create a raw socket to capture all incoming network packets.
2. Enable promiscuous mode to capture all frames, not just those directed to the machine.
3. Unpack the Ethernet frame to extract Destination MAC, Source MAC, and Protocol.
4. Continuously print the extracted details.

## 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('Ethernet 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()
```

```
Ethernet Frame:  
Destination MAC: AA:BB:CC:DD:EE:FF  
Source MAC: 11:22:33:44:55:66  
Protocol: 0x0800
```

**RESULT:**

The program continuously prints the Destination MAC, Source MAC, and Protocol of each captured Ethernet frame, running indefinitely until manually stopped.