Experiment 14: Packet Sniffing using RAW Sockets

AIM:

To implement a **packet sniffer** using RAW sockets with Python and Scapy to capture and display **IP packets** along with their protocol type, source IP, and destination IP.

CODE:

```
from scapy.all import sniff

from scapy.layers.inet import IP, TCP, UDP, ICMP


def packet_callback(packet):

    if IP in packet:

        ip_layer = packet[IP]

        protocol = ip_layer.proto

        src_ip = ip_layer.src

        dst_ip = ip_layer.dst


        # Determine the protocol

        if protocol == 1:

            protocol_name = "ICMP"

        elif protocol == 6:

            protocol_name = "TCP"

        elif protocol == 17:

            protocol_name = "UDP"

        else:

            protocol_name = "Unknown Protocol"


        # Print packet details

        print(f"Protocol: {protocol_name}")

        print(f"Source IP: {src_ip}")

        print(f"Destination IP: {dst_ip}")

        print("-" * 50)
```

```
# Capture packets on the default network interface
sniff(iface='Wi-Fi', prn=packet_callback, filter="ip", store=0)
```

SAMPLE OUTPUT:

```
Protocol: TCP
Source IP: 192.168.1.10
Destination IP: 142.250.72.14
Protocol: ICMP
Source IP: 192.168.1.10
Destination IP: 8.8.8.8

Protocol: UDP
Source IP: 192.168.1.10
Destination IP: 192.168.1.1
```

RESULT:

The packet sniffer successfully captured **IP packets** on the network, identifying their **protocol type**, **source IP**, and **destination IP**.
This demonstrates the use of **RAW sockets** for monitoring network traffic in real-time.