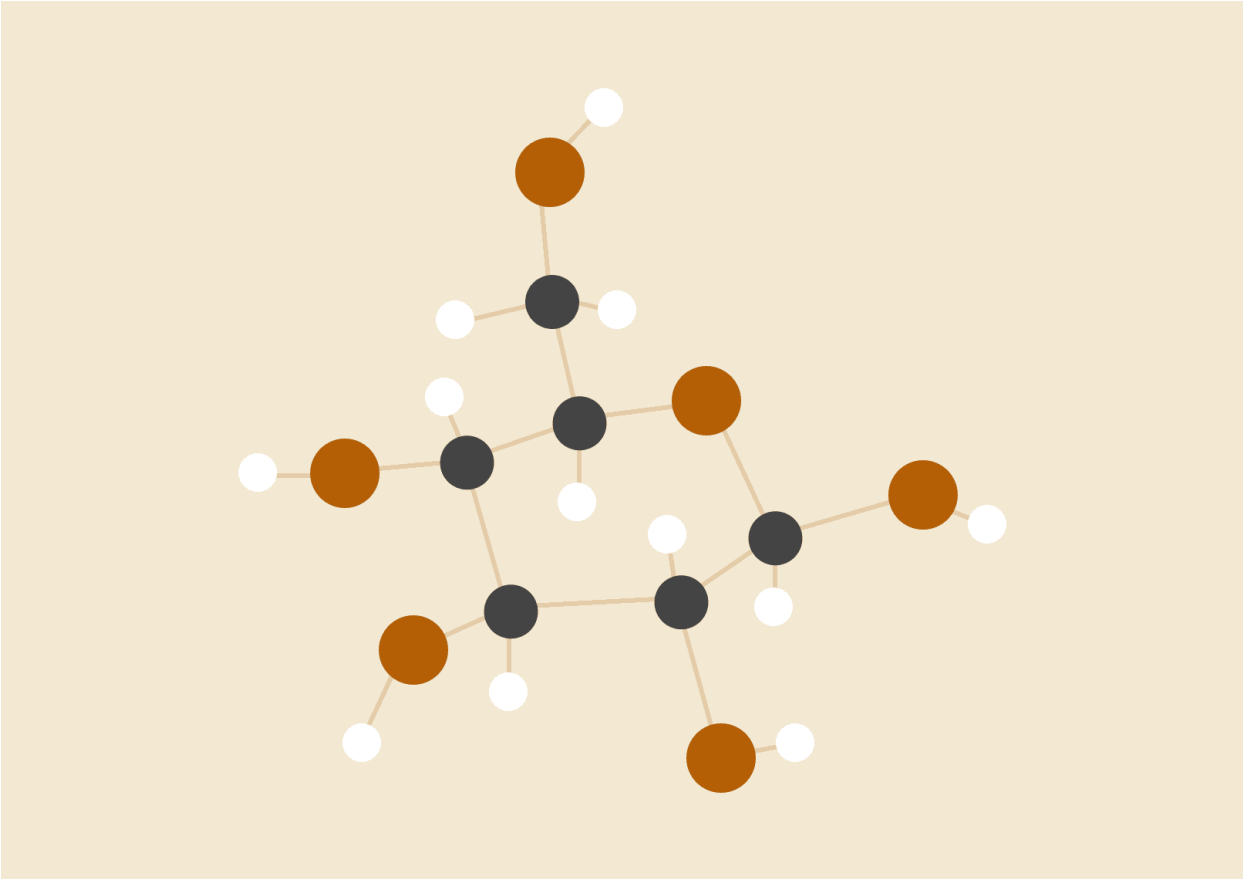
Computer Networks Lab 10



Alisetti Sai Vamsi

19/11/2021

111801002

1. Packet Sniffer sniffing sent and received packets:

Program Name: Q1.c

Compilation: gcc Q1.c -o Q1

Invocation: sudo ./Q1

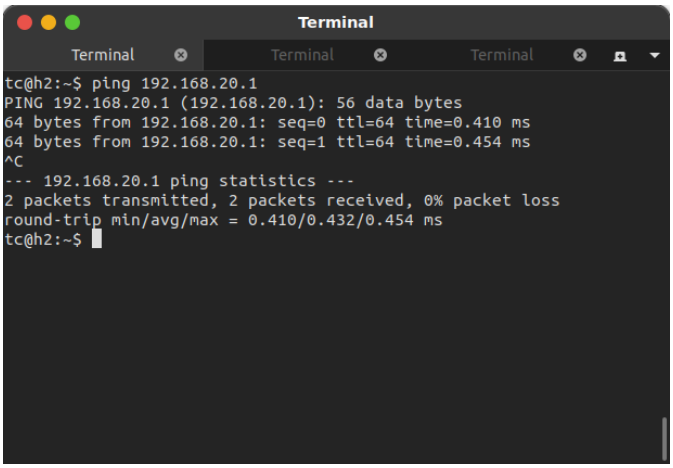
Algorithm Outline:

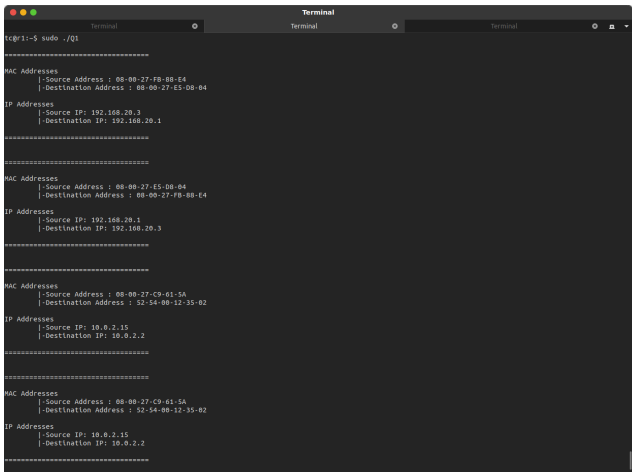
Initialization:

1. **Initialize sockets, buffer and sockaddr_ll structure.**

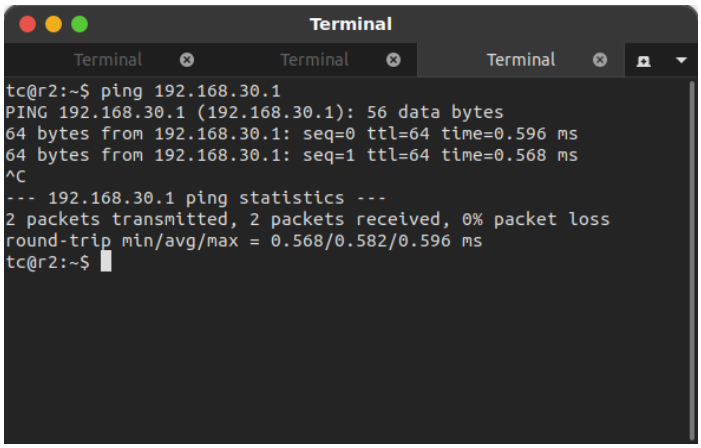
Algorithm:

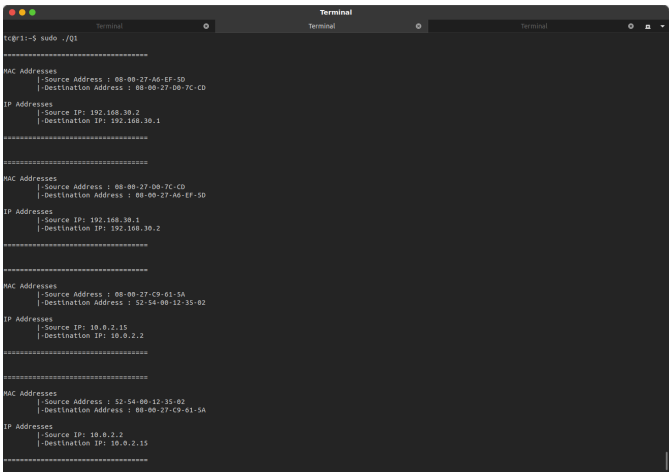
1. **Creating raw sockets with ETH_P_ALL protocol which listens to all activity on the network card.**
2. **Loop**
    a. **Receive from all the interfaces**
    b. **Process the received packets to extract the ethernet header and IP header.**
    c. **From the ethernet header extract source and destination MAC addresses.**
    d. **From the IP header extract source and destination IP addresses.**



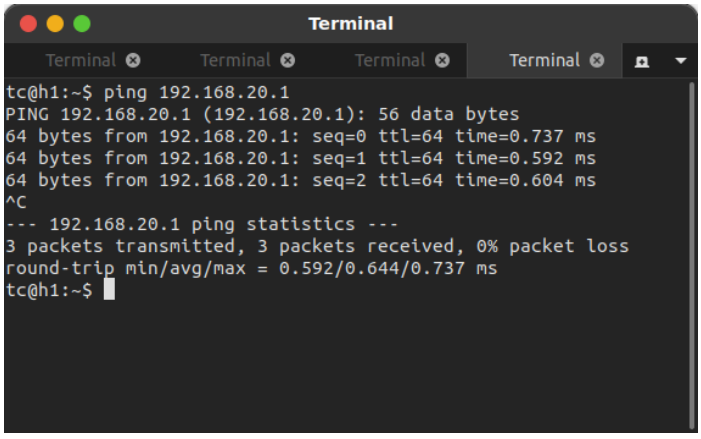


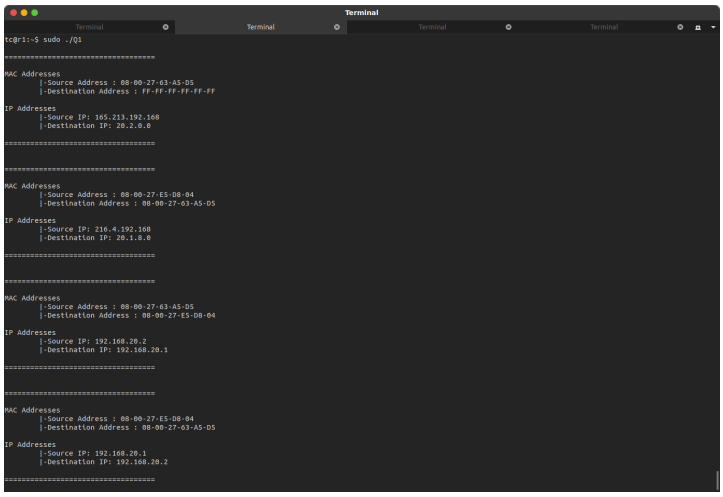
i) Pinging r1 from h2





ii) Pinging r1 from r2





iii) Pinging r1 from h1

2. Packet Sniffer sniffing received packets on eth1:

Program Name: Q2.c

Compilation: gcc Q2.c -o Q2

Invocation: sudo ./Q2

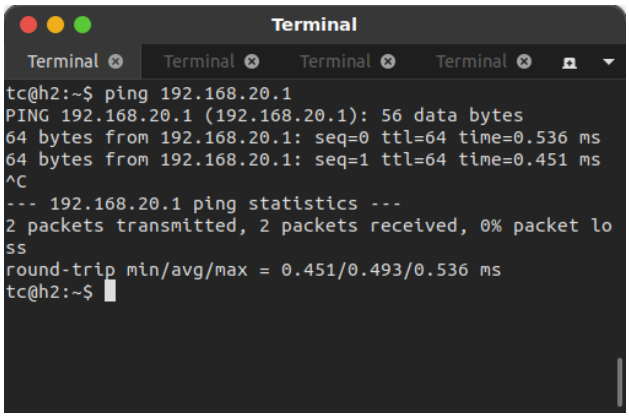
Algorithm Outline:

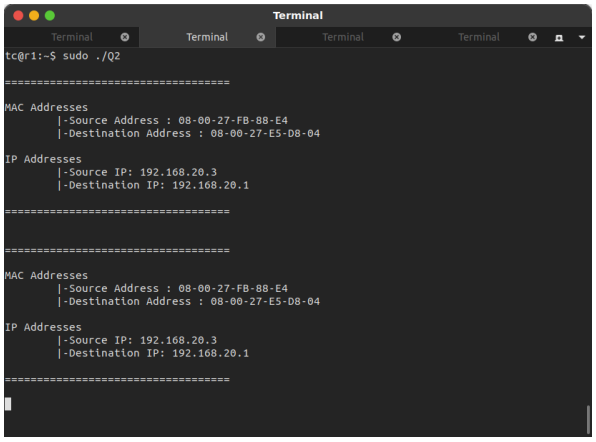
Initialization:

1. **Initialize sockets, buffer and sockaddr_ll structure.**

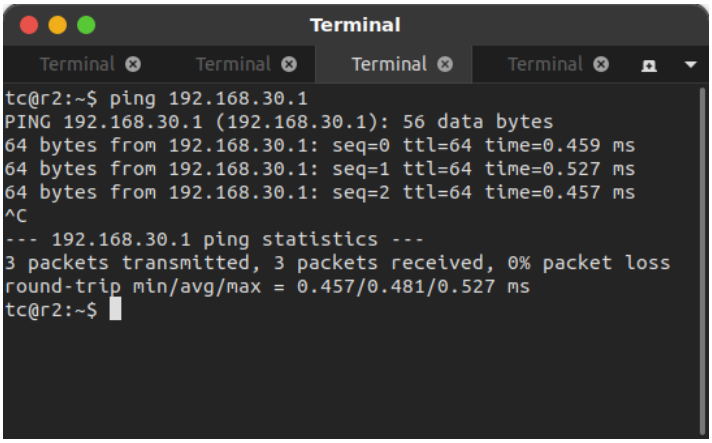
Algorithm:

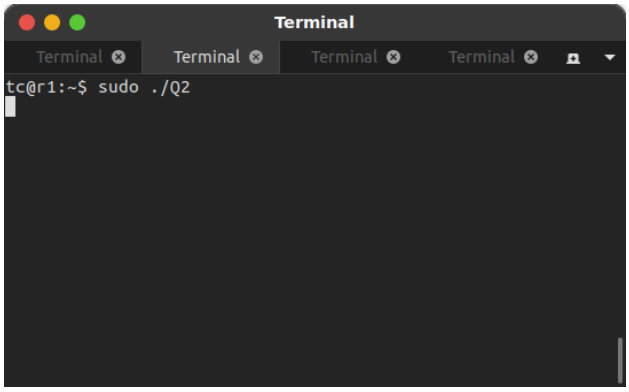
2. **Creating raw sockets with ETH_P_IP protocol which listens to only IP packets on the network card.**
3. **Loop**
    a. **Receive from all the interfaces**
    b. **Filter only received packets using packet type field in sockaddr_ll structure.**
    c. **Process the received packets to extract the ethernet header and IP header.**
    d. **From the ethernet header extract source and destination MAC addresses.**
    e. **From the IP header extract source and destination IP addresses.**



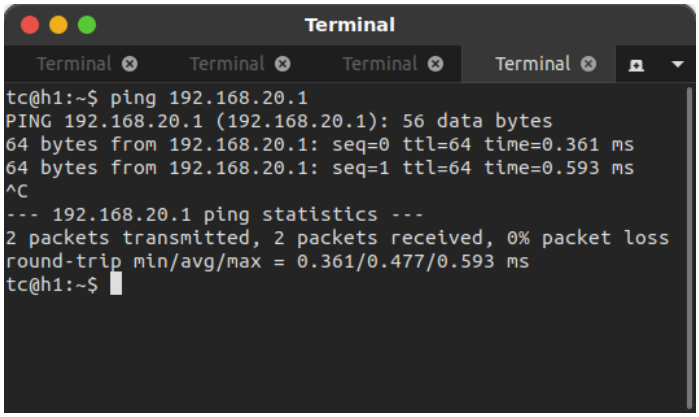


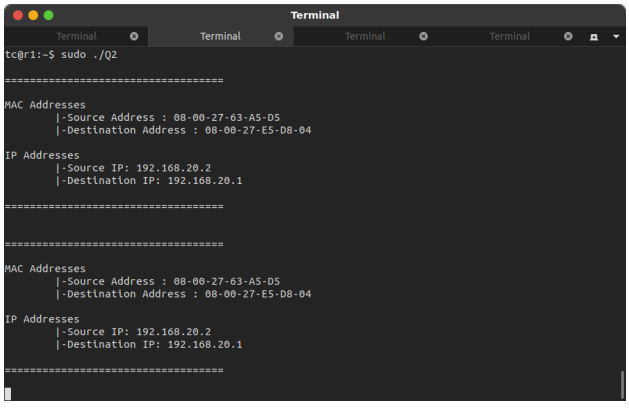
i) Pinging r1 from h2





ii) Pinging r1 from r2 produces no response on r1 since r1 is listening on eth1 not eth2





iii) Pinging r1 from h1

3. Rudimentary Router:

Program Name: Q2.c

Compilation: gcc Q2.c -o Q2

Invocation: sudo ./Q2

Algorithm Outline:

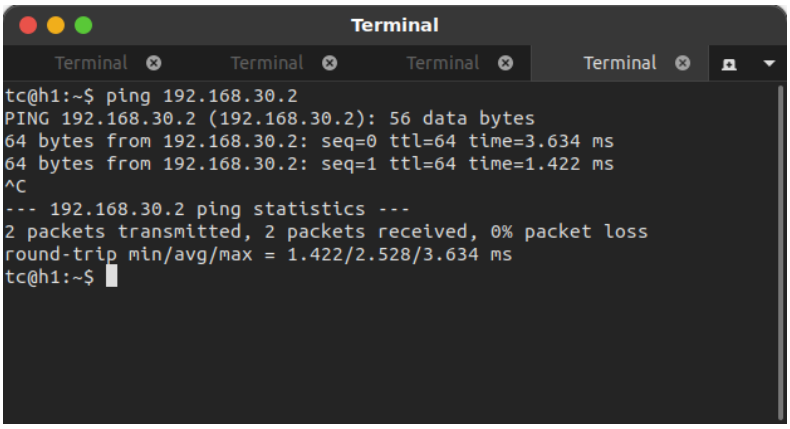
Initialization:

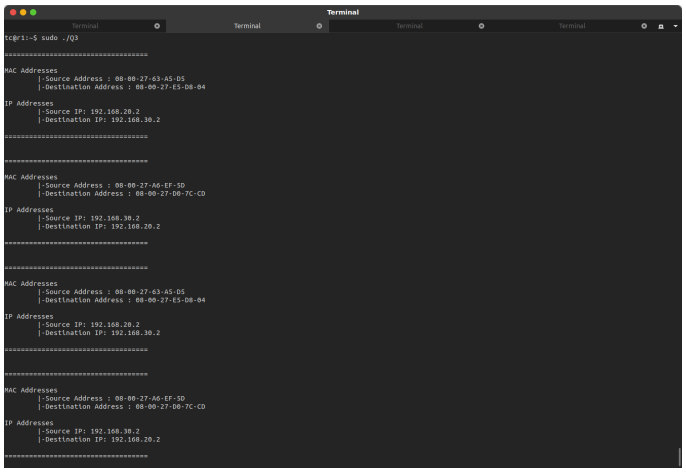
1. **Initialize sockets, buffer and sockaddr_ll structure.**

Algorithm:

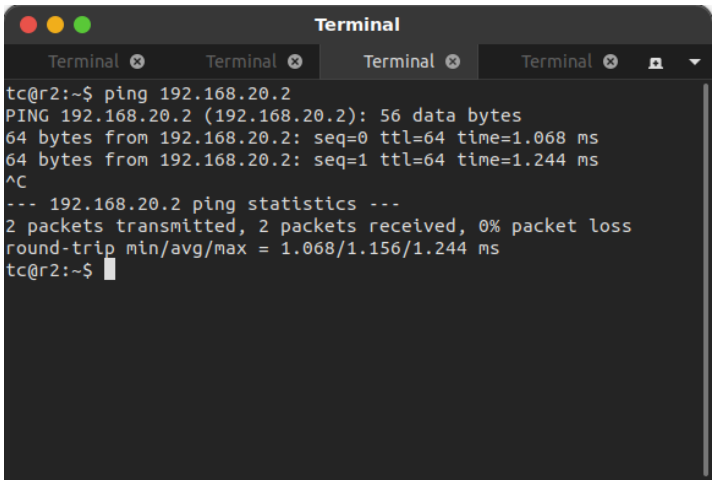
2. **Creating 4 raw sockets with ETH_P_IP protocol which listens to only IP packets on the network card. Two of the sockets are used for receiving packets and are created by using the following arguments socket(AF_PACKET, SOCK_RAW, htons(ETH_P_IP)). The other two sockets are used for sending and are created by using the following arguments socket(AF_INET, SOCK_RAW, IPPROTO_RAW). Here IPROTO_RAW protocol signifies that the data passed into the socket should be an IP packet with an IP header and transport layer header.**
3. **Loop**
    a. **Select on the receiving sockets**
        i. **If eth1 is ready on select**
            1. **Receive from interface eth1**
            2. **Filter only received packets using packet type field in sockaddr_ll structure.**
            3. **Process the received packets to extract the ethernet header and IP header.**
            4. **From the ethernet header extract source and destination MAC addresses.**
            5. **From the IP header extract source and destination IP addresses.**
            6. **Filter the packets with destination ip as different from that of ip address of interface eth1**

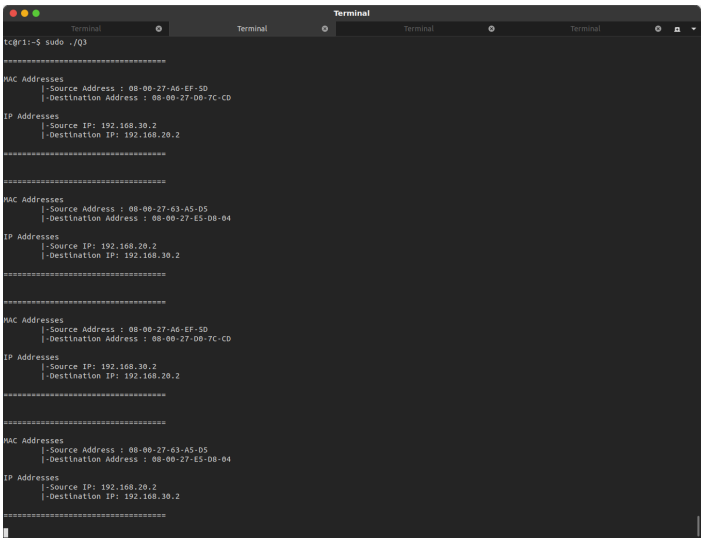
a. **Send the packet to the appropriate destination IP by filling the sockaddr_in structure and passing it to the sendto() function.**



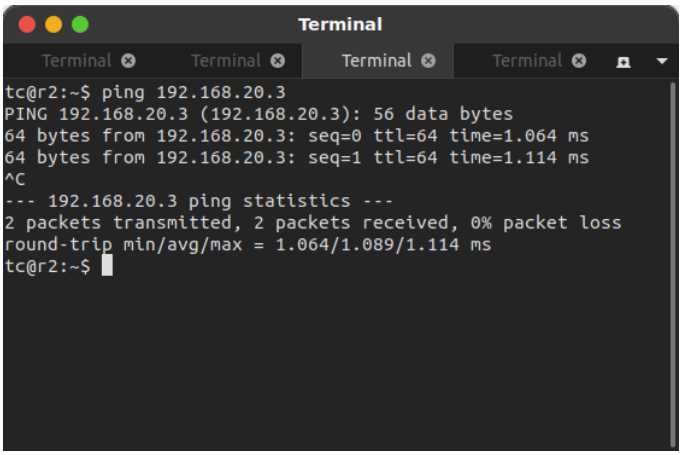


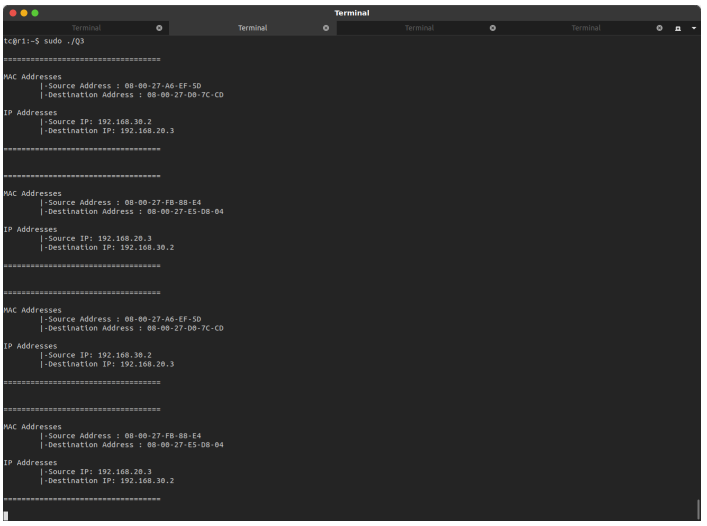
i) Pinging r2 from h1



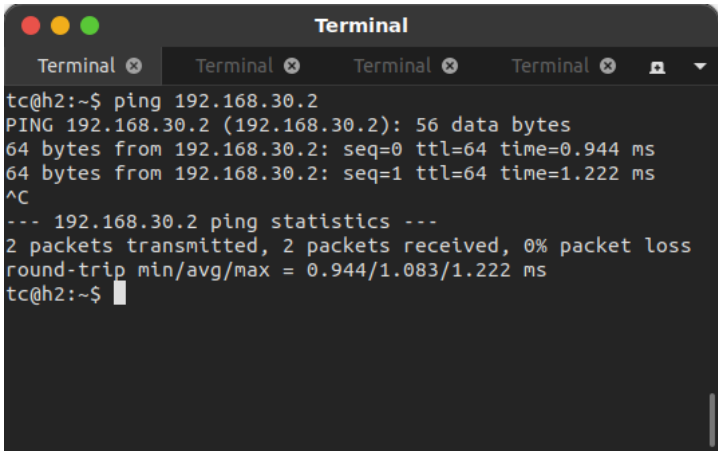


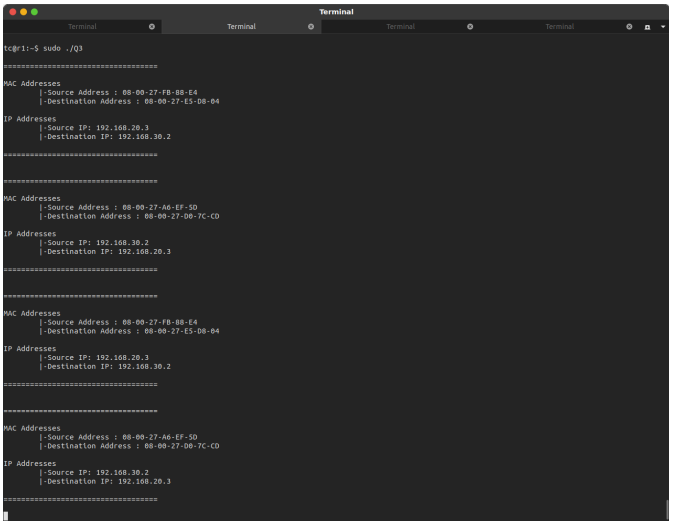
ii) Pinging h1 from r2





iii) Pinging h2 from r2





iv) Pinging r2 from h2

Note:

1. **In Q3 r1 is only printing the received packets.**
2. **In Q2 r1 is only printing the received packets as mentioned in the question.**