EE 673 Assignment 3

Akshit Verma 200091

1. **Question 1**
    a. **IP Lab**
        i. 192.168.1.46
        ii. ICMP (0x01)
        iii. Total Payload*(56 bytes)* - IP header*(20 bytes)* = IP datagram *(36 bytes)*
        iv. Not fragmented, as fragment bit is 0.
        v. Identification, TTL and Header checksum always change.
        vi. The same and different packets are
            1. The fields that stay constant are:
                a. Version *(As IPv4 is used)*
                b. Header length *(as these are all ICMP packets)*
                c. sourceIP *(same source always)*
                d. Destination IP *(sending to same dest)*
                e. Upper layer Protocol *(all ICMP packets)*
                f. Differentiated Service *(All are ICMP packets so they use same type of service class)*
            2. The fields that change are:
                a. Identification *(IP packets have different ID)*
                b. TTL *(traceroute increments each subsequent packet)*
                c. Header Checksum *(as headers change checksum changes)*
        vii. IP header identification fields increment with each ICMP *(echo)* request.
        viii. Identification: 30767, TTL: 64
        ix. Because the identifying field is a unique value, it varies for all ICMP TTL-exceeded responses. When two or more IP datagrams have the same identification value, it indicates that they are fragments of a larger IP datagram.

            TTL remains unchanged as TTL for first hop router is always the same.
        x. Yes, the packet is fragmented.
        xi. The fragmentation of the datagram is indicated by the flag bit being set. Since the fragment offset is 0, this is the first fragment. The first datagram is of total length 1500 as limited by the network card of the device on which the packet was captured.
        xii. The offset value of *1480* states that this is not the first datagram packet. There are no more fragments as the fragment bit is not set.
        xiii. Total length, flags, fragment offset and checksum are the IP headers that change between the two fragments.
        xiv. 3 packets are created after the packet size was switched to 3500.

xv. Fragment offset and checksum change for each packet. The first 2 packets also share a common header for the length*(1500)* and more fragments bit*(1)* which is 540 and 0 respectively for the third packet.

b. ICMP Lab

i. Host: 192.168.1.101, Destination: 143.89.14.34

ii. Because the ICMP packet was meant to send network-layer information between hosts and routers rather than between application layer processes, it lacks source and destination port numbers. A "Type" and a "Code" are assigned to each ICMP packet. The Type/Code combination indicates the particular message received. Because all ICMP signals are interpreted by the network software, no port numbers are required to route an ICMP message to an application layer process.

iii. ICMP type: 8, Code number: 0.
The packet also has checksum*(2 bytes)*, identifier*(2 bytes)*, sequence number*(2 bytes)*, and data fields.

iv. ICMP type: 0, Code Number: 0.
The packet also has checksum*(2 bytes)*, identifier*(2 bytes)*, sequence number*(2 bytes)*, and data fields.

v. Host IP: 192.168.1.101 Destination IP: 138.96.146.2

vi. Np, if ICMP sent UDP packets, the IP protocol number should be *0x11*.

vii. Same fields as the ping query packets.

viii. Those extra fields include both the IP header and the first 8 bytes of the original ICMP packet the error is for.

ix. The last 3 ICMP packets are message type 0 *(echo reply)* rather than 11 *(TTL expired)*. They are different as datagrams were able to reach the destination before the TTL expired.

x. The link between steps 11 and 12 have a significant delay. This is a transatlantic link from New York to Aubervilliers, France. In figure 4 from the lab, the link is from New York to Pastourelle, France.

2. Question 2

a. Link State Routing Algo

Following are the distances from each node to the other node after implementing Dijkstra's algorithm also called the link state algo.

Node 0		
Dest	Cost	Next Hop
1	1	1
2	2	1
3	4	1

Node 1		
Dest	Cost	Next Hop
0	1	0
2	1	2
3	3	2

Node 2		
Dest	Cost	Next Hop
0	2	1
1	1	1
3	2	3

Node 3		
Dest	Cost	Next Hop
0	4	2
1	3	2
2	2	2

b. Distance Vector Routing Algo

The implementation can be found in the file *question2/part2.py*. Here is the solution for the network provided. As the matrix is symmetric, the cost from 0 to 2 for example is equal to 2 to 0.

	0	**1**	**2**	**3**
0	0	1	2	4
1	1	0	1	3
2	2	1	0	2
3	4	3	2	0

3. **Question 3**
    a. For the given scenario of a 2x2 switch, we have a total of 4 possible states.
        i. **State 11**: Queue1: 1, Queue2: 1.
        ii. **State 12:** Queue1: 1, Queue2: 2.
        iii. **State 21:** Queue1: 2, Queue2: 1.
        iv. **State 22:** Queue1: 2, Queue2: 2.

    As the destinations are chosen uniformly and independently, the transition probabilities are as follows:

    - **State 11** (Possible transitions)
      There's a 50% chance queue 1 gets its packet transferred and gets a new packet wanting to go to port 1. Similarly, there's a 50% chance it gets a new packet wanting to go to port 2. The same probabilities apply for queue 2. This gives us transition probabilities of:
        - 0.50 to state 11.
        - 0.25 to state 12.
        - 0.25 to state 21.
        - 0.00 to state 22.
    - **State 12** (Possible Transitions)
      Both packets can be transferred. The new destinations are chosen independently and uniformly:
        - 0.25 to state 11.
        - 0.25 to state 12.
        - 0.25 to state 21.
        - 0.25 to state 22.
    - **State 21** (Possible transitions)
      This is symmetric to state 12, so the transition probabilities are the same:
        - 0.25 to state 11.
        - 0.25 to state 12.
        - 0.25 to state 21.
        - 0.25 to state 22.
    - **State 22** (Possible transitions)
      This is kind of symmetric to state 11, so the transition probabilities are the same:
        - 0.00 to state 11.
        - 0.25 to state 12.
        - 0.25 to state 21.
        - 0.50 to state 22.

This gives the following Transition Matrix

	11	**12**	**21**	**22**
11	0.50	0.25	0.00	0.00
12	0.25	0.25	0.25	0.25
21	0.25	0.25	0.25	0.25
22	0.00	0.25	0.25	0.50

The code *question3.py* computes the stationary distribution of a Markov chain. It starts by defining a transition matrix P, representing state transition probabilities. It then calculates the left eigenvector associated with eigenvalue 1 of the transpose of P, which corresponds to the stationary distribution. This eigenvector is normalized to ensure the probabilities sum to 1, providing the long-term state probabilities. **[0.25 0.25 0.25 0.25]** is the outcome of solving the markov chain to get the steady state. Which displays the independence and uniformity of the events.

b. Average Throughput

Given the problem, the throughput can be computed as follows:

i. *Both packets can be transferred:* This happens when both are destined to different output ports, i.e states 12 and 21.

ii. *Only one packet can be transferred:* This happens when both the packets are destined to one port, i.e state 11 and 22.

Therefore total throughput is calculated by:

$T = P_{12} \times R_{12} + P_{21} \times R_{21} + P_{11} \times R_{11} + P_{22} \times R_{22}$

On calculation we get:

$T = 0.25 \times 2 + 0.25 \times 2 + 0.25 \times 1 + 0.25 \times 1$

$T = 1.5$

4. Capacity Region

a. We have 2 channel states, namely c1 = (1,2) and c2 =(3,1)

If state is c1:

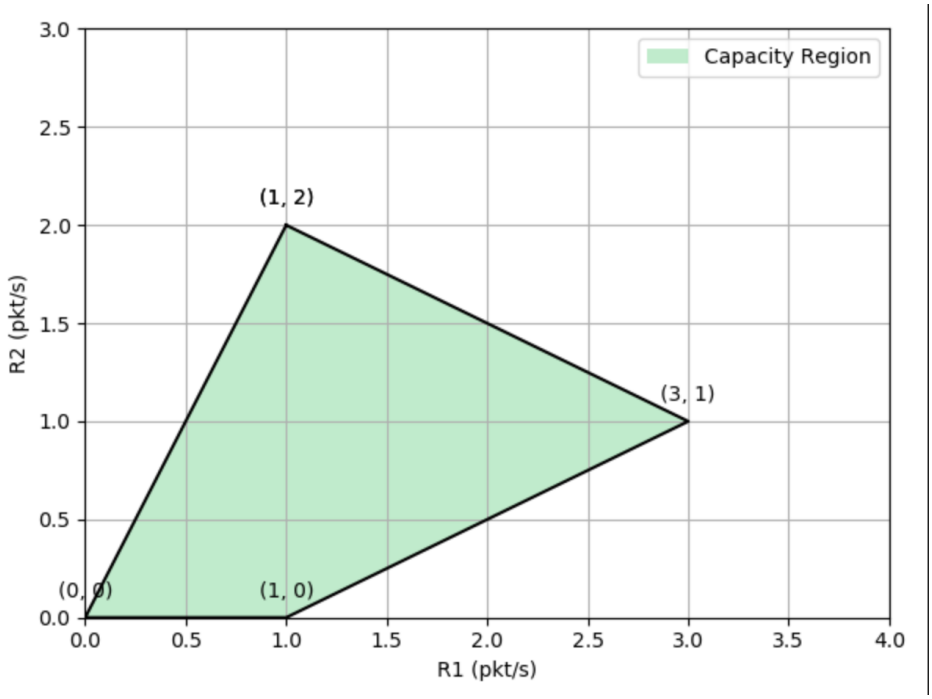
i. If mobile 1 selected, rate: 1 pkt/s.

ii. If mobile 2 selected, rate: 2 pkt/s.

If state is c2:

iii. If mobile 1 selected, rate: 3 pkt/s.

iv. If mobile 2 selected, rate: 1 pkt/s.



This is the capacity region of this wireless network. Code to plot this quadrilateral is present in *solution4.py*.

b. Given the provided system, let's denote the Lyapunov function as:

$$V(t) = q_1(t)^2 + q_2(t)^2$$

where $q = [q_1, q_2]$ is the vector of queue lengths. On analyzing the Lyapunov drift, we get:

$$E[V(t+1) - V(t)|q(t)]$$
$$= E[(q_1(t+1)^2 + q_2(t+1)^2) - (q_1(t)^2 + q_2(t)^2)|q(t)]$$
$$= E[(q_1(t) + a_1(t) - d_1(t))^2 + (q_2(t) + a_2(t) - d_2(t))^2 - (q_1(t)^2 + q_2(t)^2)|q(t)]$$
$$= E[2(q_1(t) + a_1(t) - d_1(t))(a_1(t) - d_1(t)) + 2(q_2(t) + a_2(t) - d_2(t))(a_2(t) - d_2(t)) + (a_1(t) - d_1(t))^2 + (a_2(t) - d_2(t))^2|q(t)]$$
$$= 2E[(q_1(t) + a_1(t) - d_1(t))(a_1(t) - d_1(t)) + (q_2(t) + a_2(t) - d_2(t))(a_2(t) - d_2(t))] + E[(a_1(t) - d_1(t))^2 + (a_2(t) - d_2(t))^2]$$

Analyzing the term $(q_i(t) + a_i(t) - d_i(t))(a_i(t) - d_i(t))$, we can break it down into two cases:

1. If $a_i(t) \leq d_i(t)$, then $(q_i(t) + a_i(t) - d_i(t))(a_i(t) - d_i(t)) \leq 0$
2. If $a_i(t) \geq d_i(t)$, then $(q_i(t) + a_i(t) - d_i(t))(a_i(t) - d_i(t)) = (q_i(t) + a_i(t) - d_i(t))(a_i(t) - d_i(t))$

This implies that the Lyapunov drift is less than or equal to:

$$2E[(q_1(t) + a_1(t) - d_1(t))(a_1(t) - d_1(t)) + (q_2(t) + a_2(t) - d_2(t))(a_2(t) - d_2(t))] + E[(a_1(t) - d_1(t))^2 + (a_2(t) - d_2(t))^2]$$

Now, since we are using the maximum-weight scheduling algorithm, user j(t) is selected in each time slot, which means $q_j(t) + a_j(t) - d_j(t) \geq q_i(t) + a_i(t) - d_i(t)$ for all i.
As a result, we can bound the Lyapunov drift as:

$$E[V(t + 1) - V(t)|q(t)] \leq 2E[(q_j(t) + a_j(t) - d_j(t))(a_j(t) - d_j(t))] + E[(a_j(t) - d_j(t))^2]$$

This shows that the Lyapunov drift is bounded, and by the Foster-Lyapunov theorem, the system is stable using the Lyapunov function $V(t) = q_1(t)^2 + q_2(t)^2$.