ASSIGNMENT 3

EE673A DIGITAL COMMUNICATION NETWORKS
2023-2024 (Odd)

8th October 2023 Deadline: 19th October 2023

Instructions:

1. Submissions are to be made through the HelloIITK portal
2. You are required to submit a single zip file containing well commented and easy to understand codes and a single PDF (containing answers to theory questions, if required). There should also be a readme file with clear instructions on how to run the codes.
3. Kindly name your submission file as <RollNo>_<Name>, eg: 18204269_LavishArora
4. 10% Marks will be deducted for all submissions that do not follow the above guidelines.

Question 1: Wireshark Labs

Attempt the IP and ICMP sections of the Wireshark Labs from the reference material Computer Networking A Top-Down Approach by Kurose and Ross (Links are given below). You are required to submit the answers for the questions that are part of the above Labs.

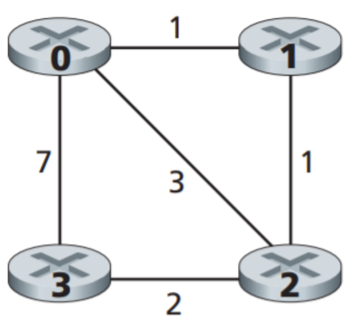
IP: https://t.ly/Syyf [*5 marks*]
ICMP: https://t.ly/R7Vk [*5 marks*]

Question 2: Centralized and Distributed Routing Algorithms

In this exercise you will implement various routing algorithms that we learned in class. You are free to use MATLAB/Python/C/C++ or any other language. For the network graph given below implement:

Link-state routing algorithm: LS routing algorithm is a global routing algorithm, i.e., each node has the complete global knowledge about the network. Since, global information is available at each node, the nodes do not need to share information with other nodes for generating its routing table. Implement LS routing algorithm (also known as Dijkstra's algorithm) for the given graph that runs at each node and outputs the cost and routing table at each node. [*20 marks*]

(***Optional Question***) **Distance-vector routing algorithm**: DV routing algorithm is a distributed algorithm. No node has complete information about the network. Each node begins with only the knowledge of costs of its own directly attached links and communicates with neighbouring nodes to share and gather information about the network. The algorithm is called Distance vector algorithm as each node maintains a vector of estimates of the costs (distances) to all other nodes in the network.



For the basic part of the assignment, you are to write the following routines. For node 0, you will write the routines:

1. `rtinit0()`
   This routine will be called once at the beginning of the emulation.

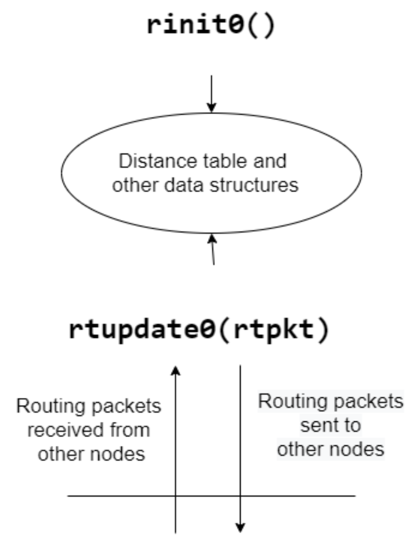
`rtinit0()` has no arguments. It should initialize the distance table in node 0 to reflect the direct costs of 1, 3, and 7 to nodes 1, 2, and 3, respectively. In Figure 1, all links are bi-directional and the costs in both directions are identical. After initializing the distance table, and any other data structures needed by your node 0 routines, it should then send its directly-connected neighbors (in this case, 1, 2 and 3) the cost of its minimum cost paths to all other network nodes. This minimum cost information is sent to neighboring nodes in a routing packet by calling the routine `tolayer2()`, as described below. The format of the routing packet is also described below.

2. `rtupdate0(rtpkt)`
   This routine will be called when node 0 receives a routing packet that was sent to it by one of its directly connected neighbors. The parameter `rtpkt` is the routing packet the format for which is described below. `rtupdate0()` is the "heart" of the distance-vector algorithm. The values it receives in a routing update packet from some other node i contain i's current shortest-path costs to all other network nodes. `rtupdate0()` uses these received values to update its own distance table (as specified by the distance-vector algorithm). If its own minimum cost to another node changes as a result of the update, node 0 informs its directly connected neighbors of this change in minimum cost by sending them a routing packet. Recall that in the distance-vector algorithm, only directly connected nodes will exchange routing packets. Thus, nodes 1 and 2 will communicate with each other, but nodes 1 and 3 will not communicate with each other.

Similar routines are defined for nodes 1, 2, and 3. Thus, you will write eight procedures in all: `rtinit0()`, `rtinit1()`, `rtinit2()`, `rtinit3()`, `rtupdate0()`, `rtupdate1()`, `rtupdate2()`, and `rtupdate3()`. These routines will together implement a distributed computation of the distance tables for the topology and costs shown in the network figure. Also write the main file which implements the algorithm. [*20 marks*]

Format for the communication packet
`rtpkt` {
int `sourceid`; /* id of node sending this pkt, 0, 1, 2, or 3 */
int `destid`; /* id of router to which pkt being sent (must be an immediate

neighbor) */
int `mincost`[4]; /* min cost to node 0 ... 3 */
};



Question 3: HOL blocking

In this exercise, we use an example to illustrate the throughput loss induced by HOL Consider a 2×2 switch, and assume input queues are infinitely backlogged. The destinations (output ports) of packets are randomly and uniformly chosen from {1, 2}, where 1 represents output port 1 and 2 represents output port 2. The destinations are independent across packets. Let $d_i(t)$ $(i = 1, 2)$ denote the destination of the head-of-line packet of input queue i at time t. When $d_1(t) \neq d_2(t)$, both head-of-line packets can be transferred to corresponding output ports; otherwise, one of them is randomly selected and transferred to the corresponding output port.

1. Note that d(t) is a Markov chain with state space {11, 12, 21, 22}. Write down the transition matrix of this Markov chain and compute the stationary distribution of the Markov chain.

2. Based on the stationary distribution, compute the average throughput of this 2×2 switch.

[*20 marks*]

Question 4: Capacity Region and Stability analysis

(a) Consider a cellular wireless network consisting of a base station and two receivers, mobile 1 and mobile 2. The network can be in one of two equally likely channel states: c1 = (1, 2) and c2 = (3, 1). Assume the channel state process is i.i.d. across time and that the base station can transmit to only one mobile in each time slot. Draw the capacity region of this wireless network.

Note: Here, the channel state $c = (c(1), c(2))$ indicates that when the channel is in state c if mobile 1 is selected for transmission it gets a rate of $c(1)$ packets/sec. Similarly, if mobile 2 is chosen for transmission in that slot, it gets a $c(2)$ packets/sec rate. [*10 marks*]

(b) Consider a maximum-weight scheduling algorithm which schedules the user with the highest weight in that time slot, where weight $w_t(i)$ of user i is defined as the product of its current queue length $q_t(i)$ and the rate $c_t(i)$ offered by the channel state c_t at time t. In other words, user $i^*(t) := \arg\max_i q_t(i)c_t(i)$ is scheduled in time slot t.

Use the Foster-Lyapunov Theorem to prove that the maximum-weight scheduling algorithm stabilizes the queues for any arrival rate in the capacity region (derived in the previous question). [*20 marks*]

END OF ASSIGNMENT