CS4150 Computer Networks Laboratory – Assignment 3

Subhash S – 111801042
Computer Science and Engineering, IIT Palakkad

1. **Consider a slotted system with n ≥ 1. In each slot, each user generates a new Ethernet frame with probability λ/n, where λ 2 [0; 1]. Assume that slot length is τ, and Ethernet frame transmission time is T. All users use a common channel and have to compete with each other for an opportunity to successfully transmit their frames. For a long enough simulation run, average throughput is defined as number of successful transmissions upon number of slots.**

1.1. Use slotted ALOHA with T = τ and n = 100, and plot the average throughput as a function of λ. Plot the theoretical prediction and compare with the simulation results. Assume that frames generated are not queued.

ALOHA is one the first multiple access protocols. It has two variants namely pure and slotted. This implementation is of the slotted ALOHA where the simulation time is divided into slots of equal time intervals.

1. The code contains an enum named *AlohaState* which stores the state a node can be in. It can transmit, re-transmit (on collision) and stay idle when it has no frames to send. These three states are captured by the enum.
2. The *simulate_slotted_aloha* function performs the simulation. It runs for a long enough time specified by the *n_slots* parameter.
3. For each slot,
    a. It loops through each node and checks if the node will generate a frame based on the probability λ/n and changes to state of the node to transmit if it generates a frame in the slot.
    b. If there are no frames generated, then the channel is idle in this slot. The count of number of idle slots is incremented.
    c. If there is exactly one frame generated, then the frame is transmitted in the channel and the node is set to idle so that it can transmit again. The count of frames sent is incremented.
    d. If there is more than one frame generated, all of them lead to collision so I change the state of that node to re-transmit which again gets a change to transmit after a random amount of time due to the randomness in 3.a.. The count of number of colliding slots is incremented.
    e. Steps (a) through (d) are performed for each time slot in the simulation.
4. The function simulate_slotted_aloha is called with various values of λ and the average throughput for each of them is stored in a list named success_avgs.
5. The plot contains λ on x-axis and the throughput for that λ on the y-axis.

1.2. Frames generated are queued. Use p-persistent CSMA with T = 3τ and n = 100, and plot the average throughput as a function of λ for p = 0:5 and p = 0:01

This code has two classes named *Channel* and *Node* representing the communication channel and a single node respectively. The *ChannelState* enum tells the state of the channel i.e. at any point in time, the channel can be *IDLE* or *BUSY* transmitting a frame.

Each *Node* maintains a queue which is nothing but a counter on number of frames in the queue. It also has a field named *ready_to_transmit* which says if the node has frames and has decided to transmit the frame after testing the probability condition.

The *Channel* class represents a channel. It has parameters like state (BUSY or IDLE) and time_step which keeps track the number of time steps for which a frame is being transmitted.

1. The simulate_slotted_aloha functions performs the simulation. It runs for a long enough time specified by the n_slots parameter.
2. For each slot,
    a. It loops through each node, if the node has items in the queue, and if it decides to transmit a frame based on the probability condition, the *ready_to_transmit* property of the node is set to true. If the frame can generate a frame based on the probability check, then it generates a frame and decides if it can transmit the frame based on the previous condition.
    b. If the channel is busy, there are two possible cases. The current slot can be last time slot of the frame after which it finishes transmitting, in which case we stop the transmission, and move to 2.a. The number of successful transmissions are also incremented. If the current slot is not the last slot, we increment the *time_step* in the channel and continue because the other *Nodes* will sense that the channel is BUSY and they defer their transmission.
    c. If the channel is IDLE, there are no frames to transmit, so continue.
    d. If the channel is IDLE, and there is exactly one frame ready to be transmitted, we start transmitting the frame for the next 3 time slots (since $T = 3\tau$).
    e. If the channel is IDLE, and there are two or more frames ready to be transmitted, this is a collision and the frame retries transmission in the next slot. During this re-transmission the channel stays busy for 3 slots since it has to recover from collission.
    f. Steps (a) through (e) are performed for each time slot in the simulation.
3. The function is called with various values of λ and the average throughput for each of them is stored in a list named success_avgs.
4. The plot contains λ on x-axis and the throughput for that λ on the y-axis.

Figure 1: Simulated vs Theoretical slotted ALOHA

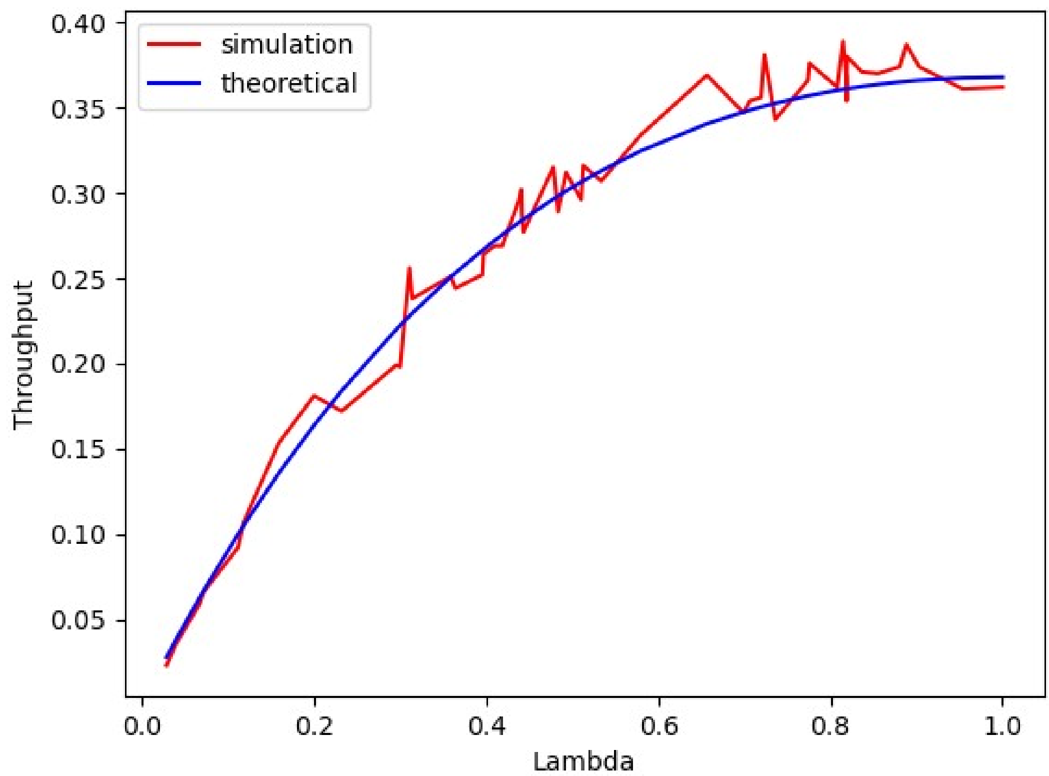

Figure 2: p-persistant CSMA for p=0.5 and p=0.01