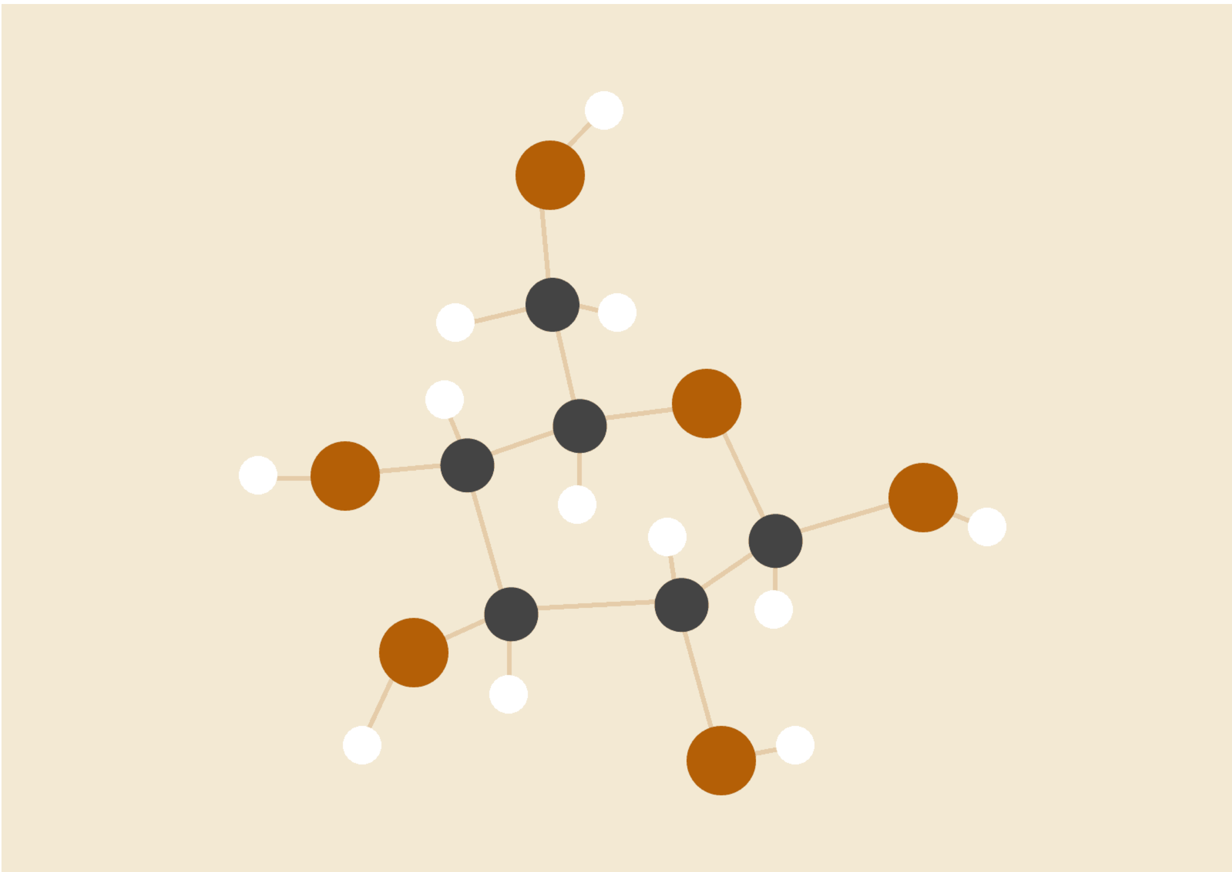
Computer Networks Lab 3



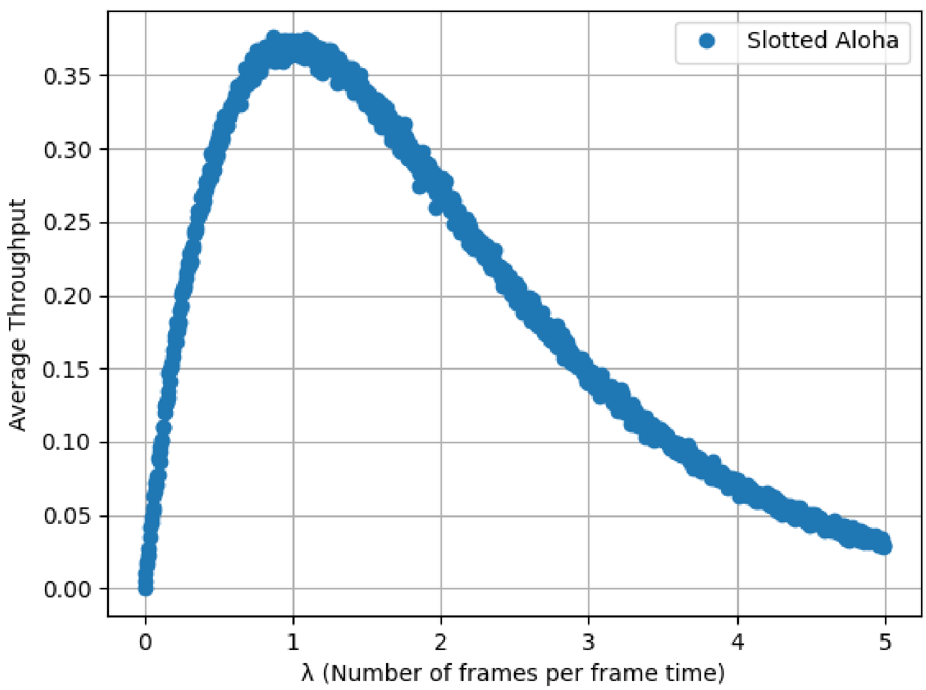
Alisetti Sai Vamsi

10/08/2022

111801002

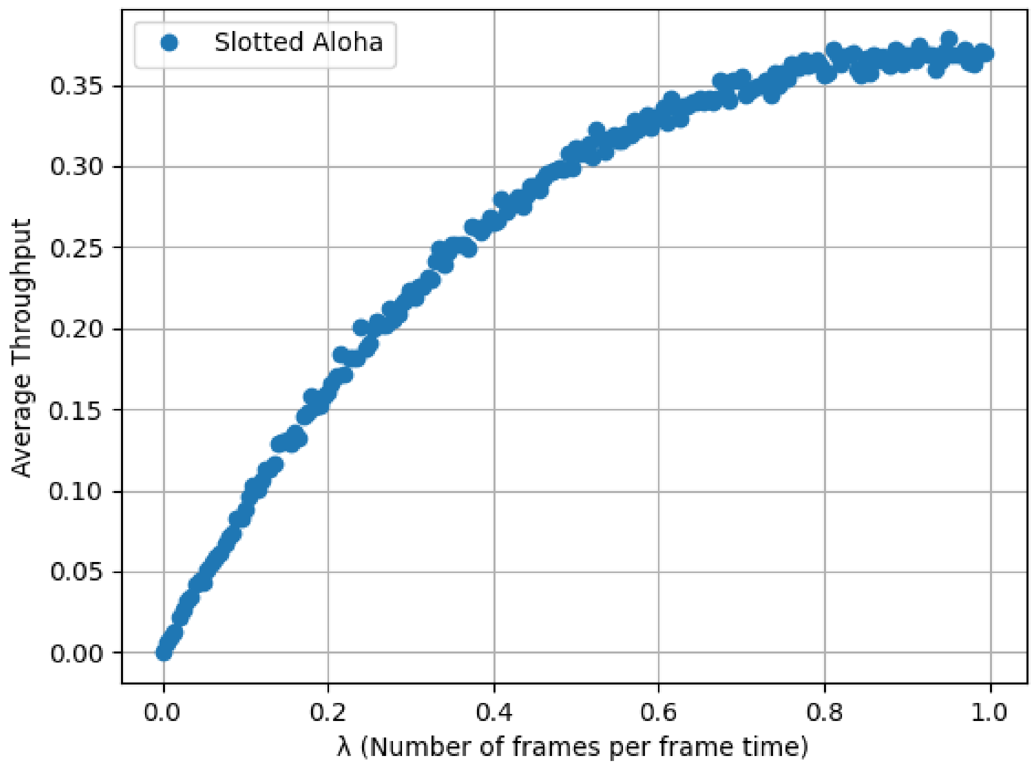
1. Consider a slotted system with n ≥ 1. In each slot, each user generates a new ethernet frame with probability λ/n, where λ ∈ [0, 1]. Assume that slot length is τ , and ethernet frame transmission time is T. All users use a common channel and have to compete with each other for an opportunity to successfully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

a. 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.



i) Figure 1: Throughput vs Lambda (0 - 5)

ii) Figure 1: Throughput vs Lambda (0 - 1)



Simulation Variables:

Number of time slots = 10000

Number of users = 100

Lambda Spacing = 0.005

Algorithm:

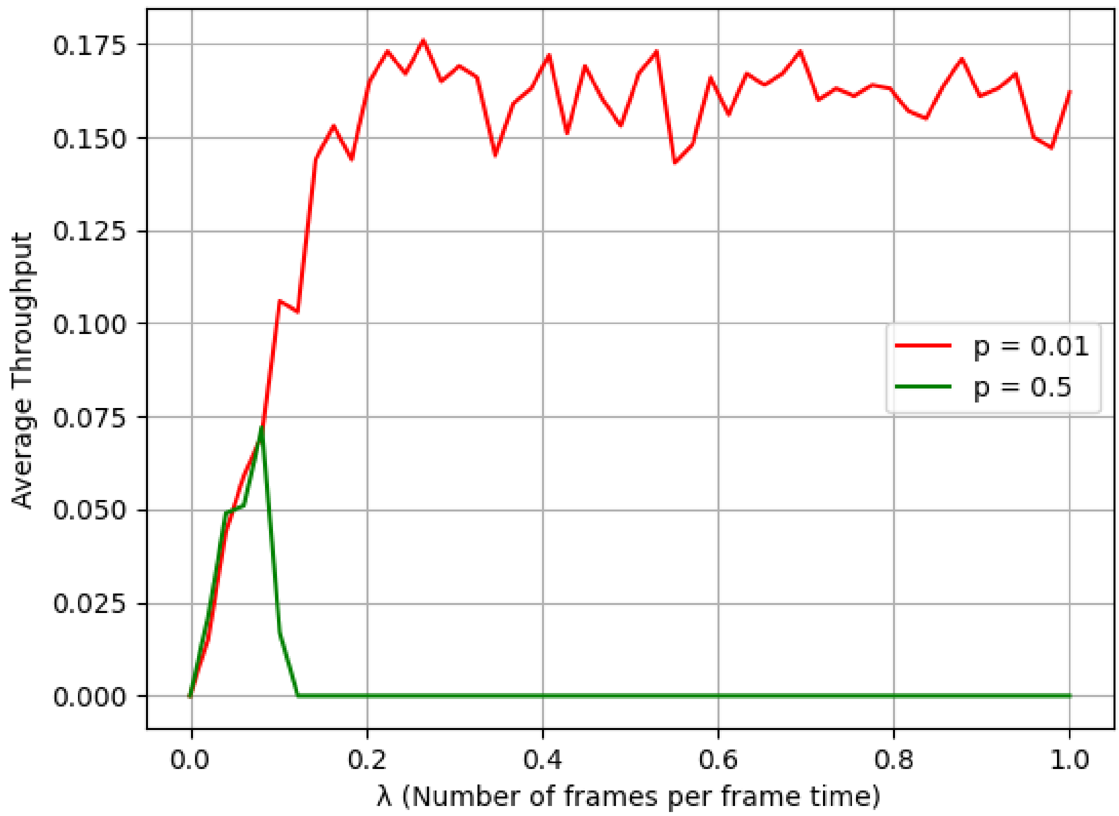
1. Variables Initialized
    a. n = 100 - 'n' is the number of User stations
    b. num_slots = 10000 - 'num_slots' is the number of time slots
    c. lamdas = np.arange(0, 1, 0.005) - 'lamdas' are the list of $\lambda \in (0, 1)$
    d. average_throughput = [] - The final output list of throughputs

2. For each lambda, set successful_transmission counter to 0, loop through the time slots, and on each loop calculate the probability of creation of a frame on each of the n stations. This can be done by creating 100 random numbers between [0,1] and finding the numbers that are less than lambda/n. Upon doing this we will get a list of 100 numbers each 0 or 1 where 1 indicates the creation of frame and 0 indicates not creating the frame.

3. From this list, if we have more than one 1 in the list, then it means more than 1 station created a frame. This means that there is a collision and we have to discard these. Hence if the sum is equal to 1 then it means that only 1 station created a frame and there are no collisions. This means that this can be transmitted and hence we increase the successful transmission counter.

4. Append these successful_transmisson values for each lambda into the average_throughput list and finally plot it with lamdas as x-axis and the average_throughput values as y-axis.

b. **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**



iii) Figure 1: Throughput vs Lambda (0 - 1)

Simulation Variables:

Number of time slots = 1000

Number of users = 100

Lambda Spacing = 0.02

Algorithm:

1. Variables Initialized
    a. np.random.seed(142) - Random Seed for better random values
    b. n = 100 - 'n' is the number of User stations
    c. lam = np.linspace(0, 1, 50) - 'lamdas' are the list of $\lambda \in (0, 1)$
    d. num_slots = 1000 'num_slots' is the number of time slots
    e. average_throughput = [] - The final output list of throughputs
    f. persistence = [0.01, 0.5] - p - Persistence Values
    g. Successful_transmissons - Keeping track of the number of successful transmissions for each lambda value.
    h. Busy_flag - Flag used for indicating whether the channel is busy or not
    i. Wait_flag - Flag used for waiting for 3-time slots.

2. For each persistence value, for each lambda, loop through all the time slots. On each loop of the lambda loop, set successful_transmissons counter = 0, wait_flag = 0, busy_flag = 0. On each loop of time slot,

    a. Create new frames on each station with a probability of (lambda/n) and append it to the respective queues on each station.
        i. In my code, I have managed this queue by creating a list of 100 binary numbers (0 or 1) where 1 indicates the presence of a frame and 0 indicates the absence of a frame. So this indicates the top of the queue for each station.
        ii. And the number of items in the queue is managed by the "counts list" which has "n" numbers that stores the frequency of created frames on each station.
    b. Check if the line is busy by checking if the wait flag is a multiple of 3 or not. If the wait flag is a multiple of 3, it means that it is not waiting on any collision or transmission which means that it is ready to transmit. Otherwise, if it is not a multiple of 3 then it is waiting on a collision or transmission.

c. If it is ready for transmission, then call the “transmit function” which takes in two parameters (top of the frames of all the n stations, persistence value p), and takes all the created frames, and runs a probability check of p on each of these. With a probability of p, we keep the created frame as it is, and with a probability of 1-p, we change the created frame to not created i.e changing a 1 to a 0.
    i. If the sum of the “transmit function” output is 1 which means that there are no collisions meaning only 1 station transmitted successfully then increment the successful_transmission by one and wait for three-time slots by setting the busy_flag to 1.
    ii. If the sum of the“transmit function” output is 0 which means that there are no transmissions at all, in that case, the channel is idle, just skip the time slot.
    iii. If the sum of the “transmit function” output is greater than 1 which means that there is a collision and we have to wait for three more time slots which is done by setting the busy_flag to 1.
    iv. Once the busy_flag is set, then we are skipping three time slots, and on the 4th time slot, the transmit function is called again.

d. Finally, append the successful_transmissons to the average_througput list and plot the graph.