# **LAB 1 - TASK 1**

rng(0)

% parameters

n\_slots =1000; %

p=0.1;

n\_users = 4;

% used to init the simulation of the ALOHA system

initSimulation(n\_users,n\_slots)

% result

n\_succ = 0; % number of successful transmissions

n\_empty= 0; % number of empty slots

n\_coll = 0; % number of slots with collisions

slot\_status = zeros(n\_users,n\_slots); % used for the plot

% simulate the transmission process for n\_slots

for t = 1:n\_slots % Loop for each slot

slot = zeros(1,16); % reset the slot

for id=1:n\_users

% get the current frame and check if it is empty

if isempty(getFrame(id))

%get a new datagram

datagram = getNewDatagram(id);

% create the frame using the user id and the datagram

frame = createFrame(id,datagram);

% save the new frame

updateFrame(id,frame);

end

% determine when to transmit

if ( rand(1)<p ) % generate a Bernoulli random variable with parameter p

% transmission: update the slot using a logical or

slot = or(slot,getFrame(id));

% delete the frame, so that we can transmit a new one

resetFrame(id);

% update the slot\_status for the final plot

slot\_status(id,t) = 1;

end

end

% is there a new message?

if ~isequal(slot,zeros(1,16))

% check the received message

if checkReceivedFrame(slot,n\_users)

n\_succ = n\_succ + 1; % update the counter of frame trasmitted successfully

else

n\_coll = n\_coll + 1; % update the counter for the collisions

end

else

n\_empty = n\_empty +1; % update the number of empty frames

end

end

% % % % Revise the following code % % % %

efficiency = 0;

% % % % Do not change the code below % % % %

display(['Total number of slots: ' num2str(n\_slots) ]);

display(['Empty slots: ' num2str(n\_empty) ]);

display(['Collisions: ' num2str(n\_coll) ]);

display(['Frame transmitted successfully: ' num2str(n\_succ) ]);

% plot

fh = figure(1);

window\_size =50;

plotTraffic(fh,slot\_status,n\_users,window\_size);

63

64

65

66

67

68

69

70

This task provides an overview of the slotted ALOHA system. Your task is to compute the efficiency of the system, which is defined as the ratio between the number of successful transmissions and the number of time slots.

INSTRUCTIONS

In our simulatino of the slotted ALOHA protocol we study here, we assume a population of **n\_users** backlogged nodes (nodes with data to send) access a shared channel by transmitting a frame in each slot with a certain probability 0 < **p** < 1. There is a nonzero probability that that only one of the nodes will use the channel, thus avoiding collision and enabling a successful transmission. Since access to the channel is controlled by the random transmission by the nodes, we call this a random access scheme. This is a slight simplification of the original ALOHA protocol, since we do not consider retransmission, but rather that frames are lost if there are collisions, and we do not explicity detect collisions.

The MATLAB code in the below window simulates a slotted ALOHA system with **n\_users**=4 nodes. Each user is identified by an index **id** ranging from 1 to **n\_users**, and transmits a data frame in each slot with probability **p**=0.1. The code simulates the transmission for **n\_slots**=1000 time slots. At the beginning of each slot, the code initializes the variable **slot**, which is the state of the shared channel in the current slot, to zero. In this code, the length of each slot/frame is 16 bits (see the next task for details). Then, each user checks whether its frame is empty. Its frame will be empty if the user transmited its frame in the previous time slot. If it is empty, it gets a new datagram (function **getNewDatagram**) and then creates and stores the frame (functions **createFrame** and **updateFrame**). Then, the user determines whether to transmit the frame or not by generating a Bernoulli random variable with parameter **p**=0.1. If the user transmits its frame, the code will get the previously stored frame (function **getFrame**), update the variable **slot**, and then resets (empties) its frame (function **resetFrame**) so that it will get a new frame in the next iteration. If the user does not transmit, then its frame is retained for future transmission. In this simulation, we assume a binary channel where the state of the channel, **slot**, is obtained by taking the logical **or** operation among all frames transmitted. In a practical system, this would be implement if each node was connected to a common wire with a pull down resistor to keep the state at zero if no nodes are transmitting, and each node is connected to the wire with a pull up transistor.

The correctness of the received frame is checked using the checksum (function **checkReceivedFrame**), which will be explained in details in Task 3. In this code, errors in the transmission are only due to multiple transmissions in the same time slot (collision), not to noise. If the checksum is correct, we increment the counter of successful transmissions (**n\_succ**). If the checksum is incorrect, we increment the number of collisions (**n\_coll**). If the slot is empty (zero), we increment the counter of empty frames (**n\_empty**).

At the end of the simulation, we display the total number of time slots, empty slots, collisions and successful transmitted frames. We also plot in Fig. 1 the transmission traffic for the first 50 slots to illustrate how ALOHA protocol works in controling the access of multiple users to the same shared channel. For each user, the plot shows a circle at the time it transmits a frame. When we observe more than one circle in a time slot (e.g., slots 4, 14, ...), there is a collision. When there are no circles (e.g., slots 1, 5, ...) there is an empty frame.

Your task is to compute the efficiency of the system and store the result inside the variable **efficiency**. Please revise the code between the lines

% % % % Revise the following code % % % %

and

% % % % Do not change the code below % % % %

Do not change other parts of the code.

## **Your Solution**

[MATLAB Documentation](https://www.mathworks.com/help/)

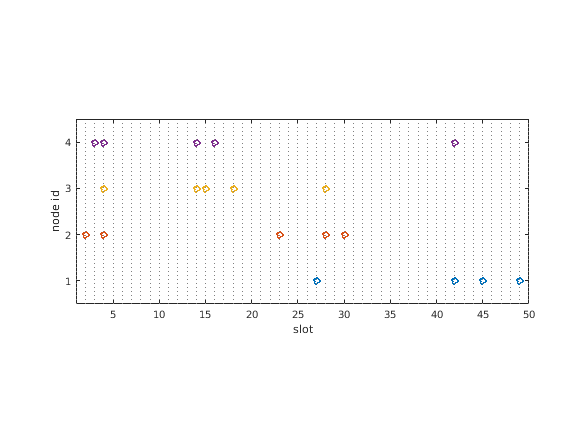
 Reset

 Save



**Output**

Total number of slots: 1000 Empty slots: 636 Collisions: 48 Frame transmitted successfully: 316



**Assessment Tests**

Is problem setup unmodified?

Is the efficiency correct?