Homework 1 (due 10/20)

Consider a network with N=10 devices. The network is synchronized in the sense that the slot boundaries of all devices are aligned. Assume that each device follows the same protocol and generates a data frame with a probability q per slot. The size of each frame is fixed to L=10 slots. The protocol used by the devices include slotted aloha, non-persistent CSMA and p-persistent CSMA. These protocols can be found in Section 4.2 while the needed parameters are summarized below. Unless explicitly specified, a transmission is erroneous only when the transmissions of two (or more) stations overlap. For simplicity, it is assumed that each station knows if its transmission is successfully --- right at the end of its own transmission.

1. slotted Aloha: A sending station will wait for a random time drawn from [0,15] slots in case of an erroneous transmission.
2. non-persistent CSMA: If a station has a data frame for transmission and senses the channel (for one slot) is busy, it will wait for a random time drawn from [0,15] slots, and repeat the protocol. In case of an erroneous transmission, it will wait for a random time drawn from [0,15] slots at the end of its own transmission and repeat the protocol.
3. p-persistent CSMA: p=0.5. In case of an erroneous transmission, it will wait for a random time drawn from [0,15] slots at the end of its own transmission and repeat the protocol.

Please apply different values of q while the network is stable in the sense that no device has an ever-growing queue. Note that at certain time point, a device may have some data frames in its own queue since. However, no data frame should be dropped at any circumstance.

Please plot the same figure as Figure 4-4 using Matlab simulation. Please note that N\*q is not equal G in the figure as G is the total number of arrival including retransmission. In your figure, the x-axis must specify G and its corresponding q.

Please compare your result with Figure 4.4 and evaluate your results.

(Here is how you simulate different protocol using Matalb)

N=10;

L=10;

q=; % you need to try a small value here and then increase it gradually

buffered\_number =zeros(N,1);

% An array to store for the number of remaining slots a sending station j (j=1 to N) has to wait until the end of its transmission;

blocked\_time=zeros(N,1);

% An array to store for the number of remaining slots a sending station j (j=1 to N) has to wait until the wait is over;

wait\_time=zeros(N,1) ;

% An array to store the state of station j (j=1 to N)

0 : idle (no data), 1: transmission, 2: collision, 3: wait

state=zeros(N,1)

TOTAL\_SLOT\_NUMBER= 100000;

t=1;

while (t<=TOTAL\_SLOT\_NUMBER)

{

% new data frame arrival for each station

for id=1:1:N

{ is\_arrival=random('unif,0,1);

if is\_arrival<=q

% record the number of data frames in the queue

buffered\_number (id)= buffered\_number (id);

end

}

% below is how you implement the protocol using the state variables "blocked\_time", "wait\_time" and "state"

for id=1:1:N

switch(state(id))

case 0 % the station is idle

if buffered\_number>0

% transmit a frame based on the protocol and update state variables

end

case 1

blocked\_time(id)= blocked\_time(id)-1;

% check if the wait is over and follow the protocol

case 2

blocked\_time(id)= blocked\_time(id)-1;

% check if the wait is over and follow the protocol

case 3

wait\_time(id)= blocked\_time(id)-1;

% check if the wait is over and follow the protocol

end

end

}