1.1 Problem Statement

To simulate Slotted Aloha using Matlab. To simulate two versions of Slotted Aloha on Matlab – Immediate First Transmission and Delayed First Transmission. To analyze the performance measures of the two variants of the given system.

1.2 Mathematical Basis

1.2.1 Slotted Aloha Mechanism

Aloha is a time-division multiple access scheme. In a multiple access scheme, there are multiple transmitters with only one receiver. The major drawback in Multiple Access schemes is that the users are independent of each other. Any user doesn't know the statistics of another user. So, there is a need for synchronization between the sending of packets from different users. There are two versions of Aloha – Pure Aloha and Slotted Aloha. In Slotted Aloha, time is divided into time slots. A user ready to transmit can transmit only at the beginning of the next time slot. The analysis of Slotted Aloha is done in this Project.

1.2.2 Mathematical Calculations

Slotted Aloha assumes that the arrival rates follow Poisson distribution. A packet can be generated at any time slot, but can be transmitted only at the beginning of a timeslot. When a new packet arrives during a timeslot, transmit it in the next time slot. When there are more than one node transmitting at one time, there is a collision. Whenever there is a collision, the node gets backlogged. Both the nodes wait for a random period of back off time, and then data is transmitted at the start of the timeslots.

The Markov state diagram of the Slotted Aloha system can be given as in Figure 1 [Reference 1].

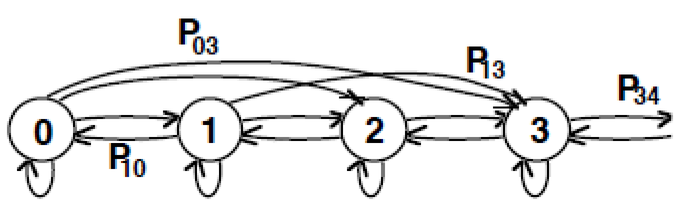


Figure 1 Markov diagram for Slotted Aloha

From the state diagram, state(n) represents the number of backlogged nodes.

$p_{i,i-1}$ = probability of one backlogged attempt and no arrival

$p_{i,i}$ = probability of no backlogged attempts and one new arrival or no new arrival and no success

$p_{i,i+1}$ = probability of one new arrival and one or more backlogged attempts

$p_{i,i+j}$ = probability of j new arrivals and one or more backlogged attempts or j+1 new arrivals and no backlogged attempts

If G is the arrival rate in each time slot, the simplified analysis of finding throughput is as follows. Probability of a successful transmission is given by

$$P(N(t) = 1) = Ge^{-G}$$

Probability of an idle slot is given is the same as the probability of a slot with no transmission. This is given by

$$P(N(t) = 0) = e^{-G}$$

Probability of a collision i.e. two or more transmission in a time slot is given by

$$P(N(t) > 1) = 1 - (Ge^{-G} + e^{-G})$$

G is usually taken as

$$G = N * p, \text{ where}$$

N = number of nodes

p= probability with which each user generates data.

Throughput can be defined as the average number of packets that can be delivered successfully to the receiver.

In Slotted Aloha, there are about $N * p$ transmission attempts and each one has a probability of e^{-G} to go through. Therefore, the average number of packets that can successfully go through the system (or simply throughput) is

$$Throughput = Npe^{-Np}$$

The maximum achievable throughput is when $N * p = 1$. Then the maximum throughput is

$$Throughput(max) = e^{-1} = 0.36 = 36\%.$$

1.3 Simulation in Matlab

This network of Slotted Aloha is simulated in Matlab. A set of N packet radios (each packet radio is a node) is considered. Each of these packet radios send information to the base station. Information is sent only at the beginning of each slot. In each node, packets are generated at a rate λ. The packets are transmitted at the rate p. If there is no collision, an acknowledgement is received for the transmitted packet. If there is a collision, the nodes sending the packets are backed off for a random wait time, and then again sent. Finally, throughput is calculated as the average number of packets successfully transmitted.

1.3.1 Part A – Immediate First Transmission

Description of the Code:

In the Matlab Code, the inputs that are obtained from the user are:

- Number of packet radios
- Packet generation Rate
- Probability of transmission
- Buffer Size at each node

The simulation time here is taken in terms of packets. So, throughput is also in packets per second. The total number of timeslots is fixed, say 10 or 20. Each node has a buffer, whose size is given by the user. The system load is also calculated as $\Lambda = N * \lambda$. First, the number of nodes ready to transmit are identified. Then the transmission can happen. Then the transmission is checked for any collision. If there is no collision, an acknowledgement is sent as a broadcast message (according to the description). In Immediate First Transmission method, when a packet is generated, it checks if the queue is empty. If it is empty, it is transmitted immediately, otherwise it is transmitted with a probability p. If a collision has occurred, the nodes are sent for a random back off time, after which they could transmit.

Observation:

The number of timeslots is initially chosen to be 10. The experiment is repeated for two values of the number of packet radios – N=2 and N=10. Considering the case of N=2, the input parameters given are shown in Table 1.

Table 1 Simulation parameters given as user inputs

Simulation Run	Number of packet radios N=2	Number of packet radios N=10
Packet generation Rate	0.02	0.02
Probability of transmission	0.5	0.5
Buffer Size at each node	5	5
Simulation Time – No. of timeslots	10	10

Figure 1 shows the sample command window output obtained for N=2 case.

```
---IMMEDIATE FIRST TRANSMISSION---
Enter the no.of packet radios :2
Enter the probability with which each user transmits :0.5
Enter the probability of packet generation :0.2
Enter Buffer Size at each node :1
System Load :
    0.4000

Timeslot 1:
No.of arriving pkts :
     1

Buffer statistics :
     1
     1

Transmitter Queue :
     0     0

Timeslot 2:
No.of arriving pkts :
     1

Buffer statistics :
     1
     1
```

Figure 1 Sample Command Window Output for N=2

The buffer statistics are shown in Figure 1. Since the network consists of only two nodes, each row represents a node. One column in each node represents the size of buffer in each node. The transmitter queue represents the queue of elements to be transmitted. Further buffer statistics is shown in Sample Output in Section 1.7. Figure 2 shows some of the buffer statistics at different timeslots. In simple words, a value of 0 indicates the buffer slot is empty and a non-zero value indicates a value in the buffer.

```
Timeslot 1:
No.of arriving pkts :
     3

Buffer statistics :
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
```

```
Timeslot 2:
No.of arriving pkts :
     3

Buffer statistics :
     1     1     1     2     2
     1     1     1     2     2
     1     1     1     2     2
     1     1     1     2     2
     1     1     1     2     2
```

```
Timeslot 3:
No.of arriving pkts :
     3
Buffer statistics :
     0     1     1     2     2
     1     1     1     2     2
     1     1     1     2     2
     1     1     1     2     2
     1     1     1     2     2
```

Figure 2 Buffer statistics for different timeslots – Sample

From Figure 2, when the buffer is full, the other packets generated are dropped. A zero value in the buffer indicates that the packet has been sent, for which the acknowledgement is received. The packets are sent out in a FIFO fashion, as in timeslot 3. Thus, for this packet, the delay is 2 time slots. This is found by setting flags and then calculating the difference between the timeslots.

Comparing the statistics for both the trials, the results are tabulated as in Table 2.

Table 2 Results obtained for different trials, for simulation time = 10 Timeslots

Parameters	Output for N=2	Output for N=10
Throughput(appr.)	34	33.1
Delay	2	3

The number of timeslots is now increased to 50. Then the simulation is run for two different number of packet radios N=2 and N=10. The results are in Table 3.

Table 3 Results obtained for different trials, for simulation time = 50 Timeslots

Parameters	Output for N=2	Output for N=10
Throughput	34.6	34.2
Delay	2	4

Then the probability of user transmission, p is varied, with the simulation time fixed at 10 timeslots. For N=2 and N=10, the results are in Table 4.

Table 4 Results obtained for different trials, for transmission probability = 1

Parameters	Output for N=2	Output for N=10
Throughput	30	31
Delay	2	3

This experiment is repeated for different buffer sizes too. Considering the case of N=10, the output parameters for Buffer Sizes B=1,2 and 5 are tabulated in Table 5.

Table 5 Results obtained for N=10, for different buffer Sizes

Parameters	Buffer size = 1	Buffer Size =2	Buffer Size =10
Throughput	35.1	35.7	36.1
Delay	2	4	4

Figure 3 shows a sample drop count of the packets in each node.

```
Node: 1
Total :8 Ack :1 Dropped :7 Queue:0 Retrans:0
Node: 2
Total :8 Ack :0 Dropped :7 Queue:0 Retrans:0
>>
```

Figure 3 Sample Packet Drop statistics for two nodes

<u>Results:</u>

i. In the IFT mode of Slotted Aloha, the decision is made only on one condition – checking if the queue is full or empty. If the queue is empty, the packets are transmitted. If they are non – empty, they are sent for re transmission with some probability p.

ii. Throughput can be defined as the number of packets that are received successfully against the number of packets that are sent. Delay refers to the time between sending a packet and receiving acknowledgement for that packet. Both Throughput and Delay depends on numerous factors like Simulation Time, the number of nodes in the network, transmission probability and the buffer size at each transmitter.

iii. From table 2, throughput and delay due to different number of nodes in the network are shown. For a smaller number of nodes, the collisions are less, and hence a higher throughput can be expected.

iv. From Table 2 and 3, variation in Throughput because of different simulation times of the network is shown. Throughput becomes higher for longer time, since the network tend to become more stable in terms of generation of packets. Delay for the first packet is only shown in this code, which depends on the random number generation, hence there is only a slight variation.

v. From Table 4, the throughput for different value of transmission probability is obtained, for two varied sizes of the network. Maximum throughput depends on Np. Since here Np is not equal to 1, maximum throughput is not obtained.

vi. From Table 5, the statistics of throughput and delay for different buffer sizes are shown. As the buffer size increases, the throughput is also expected to increase. As said earlier, the packet arrivals are random, with some probability. Hence the variation in throughput is observed. This is the case with delay too.

vii. Figure 3 shows some total packet drop and packet loss statistics, from which throughput was calculated.

1.3.2 Part B – Delayed First Transmission

Description of the Code:

In the Matlab Code, the inputs that are obtained from the user are:

- Number of packet radios
- Packet generation Rate
- Probability of transmission
- Buffer Size at each node

The simulation time here is taken in terms of packets. So, throughput is also in packets per second. The total number of timeslots is fixed, say 10 or 20. Each node has a buffer, whose size is given by the user. The system load is also calculated as $\Lambda = N * \lambda$. First, the number of nodes ready to transmit are identified. Then the transmission can happen. Then the transmission is checked for any collision. If there is no collision, an acknowledgement is sent as a broadcast message (according to the description). In Delayed First Transmission method, all packets are transmitted with a probability p. If a collision has occurred, the nodes are sent for a random back off time, after which they could transmit.

Observation:

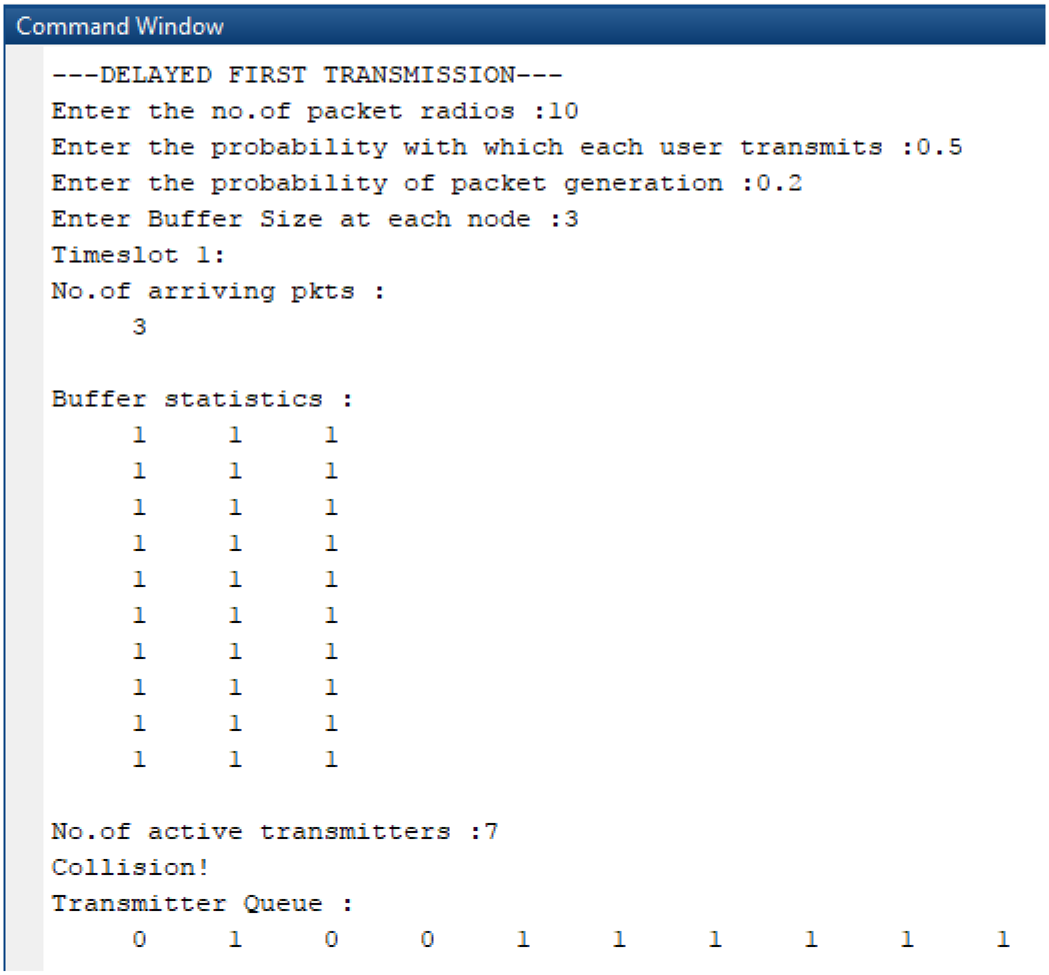
The number of timeslots is chosen to be 10. The experiment is repeated for two values of the number of packet radios – N=2 and N=10.

Considering the case of N=2, the input parameters given are shown in Table 6.

Table 6 Simulation parameters given as user inputs

Simulation Run	Number of packet radios N=2	Number of packet radios N=10
Packet generation Rate	0.02	0.02
Probability of transmission	0.5	0.5
Buffer Size at each node	5	5
Simulation Time – No. of timeslots	10	10

Figure 4 shows the sample command window output for N=10.



```
Command Window
---DELAYED FIRST TRANSMISSION---
Enter the no.of packet radios :10
Enter the probability with which each user transmits :0.5
Enter the probability of packet generation :0.2
Enter Buffer Size at each node :3
Timeslot 1:
No.of arriving pkts :
     3

Buffer statistics :
     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1
     1     1     1

No.of active transmitters :7
Collision!
Transmitter Queue :
     0     1     0     0     1     1     1     1     1     1
```

Figure 4 Sample command window output for N=10.

The buffer statistics are like shown in Figure 2.

Comparing the statistics for both the trials, the results are tabulated as in Table 7.

Table 7 Results obtained for different trials, for simulation time = 10 Timeslots

Parameters	Output for N=2	Output for N=10
Throughput(appr.)	35.5	35.8
Delay	2	4

The number of timeslots is now increased to 50. Then the simulation is run for two different trials N=2 and N=10. The results are in Table 8.

Table 8 Results obtained for different trials, for simulation time = 50 Timeslots

Parameters	Output for N=2	Output for N=10
Throughput	35.4	34.6
Delay	2	7

Then the probability of user transmission, p is varied, with the simulation time fixed at timeslots. For N=2 and N=10, the results are in Table 9.

Table 9 Results obtained for different trials, for transmission probability = 1

Parameters	Output for N=2	Output for N=10
Throughput	34.6	34.3
Delay	2	3

This experiment is repeated for different buffer sizes too. Considering the case of N=10, the output parameters for Buffer Sizes B=1,2 and 5 are tabulated in Table 10.

Table 10 Results obtained for N=10, for different buffer Sizes

Parameters	Buffer size = 1	Buffer Size =2	Buffer Size =10
Throughput	32.1	33.9	33.7
Delay	2	4	5

Results:

i. In the DFT mode of Slotted Aloha, the decision is made depending on the probability condition – All packets are transmitted with a probability p.
ii. Throughput can be defined as the number of packets that are received successfully against the number of packets that are sent. Delay refers to the time between sending a packet and receiving acknowledgement for that packet. Both Throughput and Delay depends on numerous factors like Simulation Time, the number of nodes in the network, transmission probability and the buffer size at each transmitter.
iii. From table 7, throughput and delay due to different number of nodes in the network are shown. For a smaller number of nodes, the collisions are less, and hence a higher throughput can be expected. But here, a higher throughput is obtained for larger network size, because of the transmission probability factor.
iv. From Table 7 and 8, variation in Throughput because of different simulation times of the network is shown. Throughput becomes smaller for longer time, since the network tend to become more stable in terms of generation of packets. This is because of the use of random permutations in the simulation to determine the probability parameters. Delay for the first packet is only shown in this code, which depends on the random number generation, hence there is only a slight variation.
v. From Table 9, the throughput for different value of transmission probability is obtained, for two varied sizes of the network. Maximum throughput depends on Np. Since here Np is not equal to 1, maximum throughput is not obtained. When transmission probability = 0.1, it is found that throughput almost equals 36%.
vi. From Table 10, the statistics of throughput and delay for different buffer sizes are shown. As the buffer size increases, the throughput is also expected to increase. As said earlier, the packet arrivals are random, with some probability. Hence the variation in throughput is observed. This is the case with delay too.
vii. Figure 5 shows some total packet drop and packet loss statistics, from which throughput was calculated.

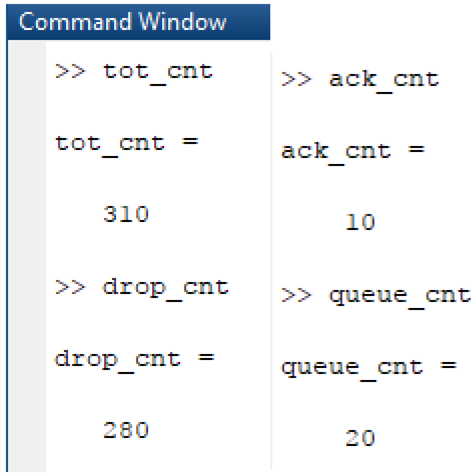


Figure 5 Sample packet loss statistics for N=10

viii. A graph of the system load against the throughput is drawn. The value of the system load is varied. Throughput is obtained for each case. The graph is obtained as in Figure 6. It is found that maximum throughput of 36% is obtained only when G=1.

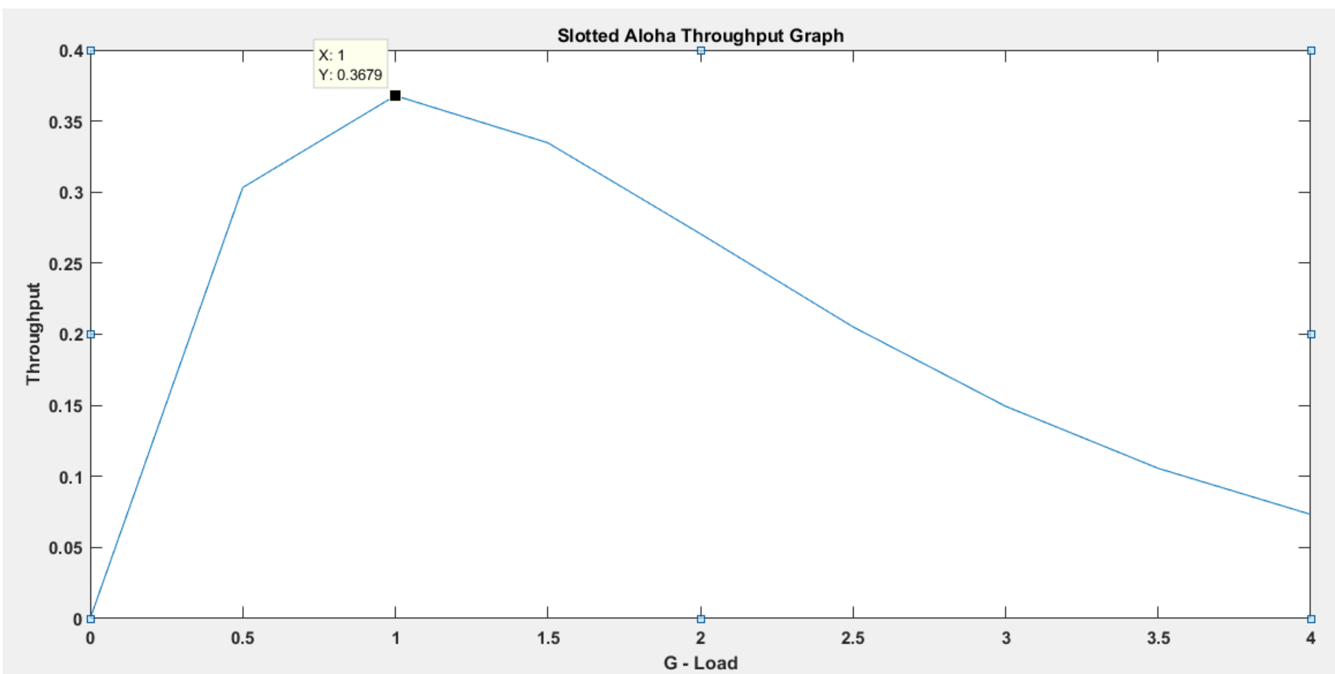


Figure 6 Throughput Graph of Slotted Aloha

1.4 Results and Conclusion

From the discussion of Results in Section 1.3, it is found that the IFT mode of transmission is suitable only if there is a smaller size of the network. Since this code uses random number as its primary source of decision for generation rate of packets, determination of buffer size etc. IFT gives a comparatively higher throughput compared to DFT. DFT is effective in the case that there is an equal probability of transmission of all nodes. In IFT there are a lesser number of packet drops, since most of the packets are retransmitted. Hence IFT can be used for networks that are potentially more collisive, since a higher throughput can be expected.

1.5 References

1. web.mit.edu/modiano/www/6.263/lec10.pdf
2. Sheldon M Ross, “Simulation”, 5th edition
3. www.mathworks.com

1.6 Matlab Code

Part A – IFT

```
clc;clear all;close all;
%N is the number of packet radios
disp('---IMMEDIATE FIRST TRANSMISSION---');
N=input('Enter the no.of packet radios :');

%T is the time duration of each slot
T=0.1;

%ts - Total number of time slots
ts=10;

%tot_time = total time of simulation
tot_time=T*ts;

%Each user transmits with a probability p within time T.
p=input('Enter the probability with which each user transmits :');

%lambda = average no.of packets in the system in time T
lambda=input('Enter the probability of packet generation :');

% % % %pk=Probability of k packet transmissions in unit time T
% % % pk=power(lambda,k)*exp(-1*lambda)/factorial(k)

%tr_q is an array th          at tells the no.of waiting elements in
%each transmitter queue. Here assuming queue size is B
B=input('Enter Buffer Size at each node :');
tr_q=zeros(N,B);

disp('System Load :');
disp(N*lambda);
%tr_cnt is the transmitter count that tells no.of packets to be sent
tr_cnt=zeros(1,N);
```

```
%drop_cnt counts the no.of packets dropped
drop_cnt=0;
drop_pkts=zeros(1,N);
%tot_cnt counts the total no.of packets arrived
tot_cnt=0;
tot_pkts=zeros(1,N);
%ack_cnt tells the total number of packets acknowledged
ack_cnt=0;
ack_pkts=zeros(1,N);
ack_flags=zeros(ts,N);
coll_keep=zeros(1,N);
%Transmit queue
tr_queue=zeros(1,N);

%2D array that stores the random backoff times for each slot
tr_rq=zeros(ts,N);

%Random Wait times
rwait=zeros(1,N);

q_cnt=0;

%Immediate First Transmission - check if the queue is empty
for i=1:1:ts
    %calculation for every node
    arr_pkts=ceil(lambda*rand(1,1));
    disp(['Timeslot ',num2str(i),':']);
    %calculate the number of arriving packets
    %arr_pkts is the no.of packets arriving each TS.
    disp('No.of arriving pkts :');disp(arr_pkts);

    for j=1:1:N
        tot_cnt=tot_cnt+arr_pkts;
        tot_pkts(j)=tot_pkts(j)+arr_pkts;
        %check if buffer is full or empty
        x=all(tr_q(j,:)==1); %=1 means no non zero elements in array-->Buffer full

        if x==1
            drop_cnt=drop_cnt+arr_pkts;
            drop_pkts(j)=drop_pkts(j)+arr_pkts;

        else
            %count the number of available free slots in buffer
            d=sum(tr_q(j,:)==0);
            %fill the remaining slots with generated packets, drop the rest
            if arr_pkts<=d
                cnt=0;
                for n=1:1:B
                    if cnt<arr_pkts
                        if tr_q(j,n)==0
                            tr_q(j,n)=i;
                            cnt=cnt+1;
                            q_cnt=q_cnt+1;
                        end
                        n=n+1;
                    end
                end
            else
```

```
                diff=arr_pkts-d;
                drop_cnt=drop_cnt+diff;
                cnt=0;
                for n=1:1:B
                    if cnt<arr_pkts
                        if tr_q(j,n)==0
                            tr_q(j,n)=i;
                            cnt=cnt+1;
                            q_cnt=q_cnt+1;
                        end
                        n=n+1;
                    end
                end

            end
            f=~any(tr_q(:));
            if f==0
                % To check if the queue is empty and then transmit the packets if
the queue is empty.
                ack_cnt=ack_cnt+1;
                ack_flags(i,j)=i;
            else
            end

        end
    end
    disp('Buffer statistics :');
    disp(tr_q)
    %No.of transmitter nodes active

    tr_active=randi([0 1],1,N);
    %tr_active=input('Enter the transmit_active array :');

    %Checking if there is any other active transmitter node because of
    %random backoff times
    for n=1:1:N
        for p=1:1:ts
            if tr_rq(p,n)~=0
                tr_active(n)=1;
            end
        end
    end



    %coll_keep is an array that stores the collision occurrence info
    for n=1:1:N
        if tr_active(n)==1
            coll_keep(n)=1;
        end
    end
    if sum(coll_keep)>1
        %Collision
        %Backlog all nodes with random backoffs
        disp(['No.of active transmitters :',num2str(sum(coll_keep))]);
        disp('Collision!');
        for n=1:1:N
            if tr_active(n)==1
                rwait(n)=randi([0,ts],1,1);
```

```
                tr_queue(n)=tr_queue(n)+1;
                tr_rq(i,n)=rwait(n);
                coll_keep(n)=0;
            end
        end

    else
        % No Collision - Transmit the packets with a probability p
        trp=ceil(p*rand(1,1));
        ack_cnt=ack_cnt+1;
        for n=1:N
            if tr_active(n)==1
                ack_pkts(n)=ack_pkts(n)+1;
            end
        end

    end
    disp('Transmitter Queue :');
    disp(tr_queue);
end
    del=3;
for n=1:N
    disp(['Node: ',num2str(n)]);
    T1=tot_pkts(n);A=ack_pkts(n);D=drop_pkts(n);Q=sum(tr_q(n,:));R=T1-A-Q-D;
    disp(['Total :',num2str(T1),' Ack :',num2str(A),' Dropped :',num2str(D),'
Queue:',num2str(Q),' Retrans:',num2str(R)]);
end

th=T1/(T*ts);
disp(['Throughput : ',num2str(th),' packets per second']);
disp(['System Load :',num2str(N*lambda)]);
disp(['Delay:',num2str(del)]);
```

Part B – DFT

```
clc;clear all;close all;
%N is the number of packet radios
disp('---DELAYED FIRST TRANSMISSION---');
N=input('Enter the no.of packet radios :');

%T is the time duration of each slot
T=0.1;

%ts - Total number of time slots
ts=10;

%tot_time = total time of simulation
tot_time=T*ts;

%Each user transmits with a probability p within time T.
p=input('Enter the probability with which each user transmits :');

%lambda = average no.of packets in the system in time T
lambda=input('Enter the probability of packet generation :');

% % % %pk=Probability of k packet transmissions in unit time T
% % % pk=power(lambda,k)*exp(-1*lambda)/factorial(k)
```

```
%tr_q is an array that tells the no.of waiting elements in
%each transmitter queue. Here assuming queue size is B
B=input('Enter Buffer Size at each node :');
tr_q=zeros(N,B);

%tr_cnt is the transmitter count that tells no.of packets to be sent
tr_cnt=zeros(1,N);

%drop_cnt counts the no.of packets dropped
drop_cnt=0;
drop_pkts=zeros(1,N);
%tot_cnt counts the total no.of packets arrived
tot_cnt=0;
tot_pkts=zeros(1,N);
%ack_cnt tells teh total number of packets acknowledged
ack_cnt=0;
ack_pkts=zeros(1,N);

coll_keep=zeros(1,N);
%Transmit queue
tr_queue=zeros(1,N);

%2D array that stores the random backoff times for each slot
tr_rq=zeros(ts,N);

%Random Wait times
rwait=zeros(1,N);

q_cnt=0;

%Delayed First Transmission - each packet is transmitted with a probability
%p
for i=1:1:ts
    %calculation for every node
    arr_pkts=randperm(5,1);
    disp(['Timeslot ',num2str(i),':']);
    %calculate the number of arriving packets
    %arr_pkts is the no.of packets arriving each TS.
    disp('No.of arriving pkts :');disp(arr_pkts);

    for j=1:1:N
        tot_cnt=tot_cnt+arr_pkts;
        tot_pkts(j)=tot_pkts(j)+arr_pkts;
        %check if buffer is full or empty
        x=all(tr_q(j,:)==1); %=1 means no non zero elements in array-->Buffer full
        if x==1
            drop_cnt=drop_cnt+arr_pkts;
            drop_pkts(j)=drop_pkts(j)+arr_pkts;
        else
            %count the number of available free slots in buffer
            d=sum(tr_q(j,:)==0);
            %fill the remaining slots with generated packets, drop the rest
            if arr_pkts<=d
                cnt=0;
                for n=1:1:B
                    if cnt<arr_pkts
                        if tr_q(j,n)==0
```

```
                    tr_q(j,n)=i;
                    cnt=cnt+1;
                    q_cnt=q_cnt+1;
                end
                n=n+1;
            end
        end


    else
        diff=arr_pkts-d;
        drop_cnt=drop_cnt+diff;
        cnt=0;
        for n=1:1:B
            if cnt<arr_pkts
                if tr_q(j,n)==0
                    tr_q(j,n)=i;
                    cnt=cnt+1;
                    q_cnt=q_cnt+1;
                end
                n=n+1;
            end
        end

    end
  end
end
disp('Buffer statistics :');
disp(tr_q)
%No.of transmitter nodes active

tr_active=randi([0 1],1,N);
%tr_active=input('Enter the transmit_active array :');

%Checking if there is any other active transmitter node because of
%random backoff times
for n=1:1:N
    for p=1:1:ts
        if tr_rq(p,n)~=0
            tr_active(n)=1;
        end
    end
end



%coll_keep is an array that stores the collision occurrence info
for n=1:1:N
    if tr_active(n)==1
        coll_keep(n)=1;
    end
end
if sum(coll_keep)>1
    %Collision
    %Backlog all nodes with random backoffs
    disp(['No.of active transmitters :',num2str(sum(coll_keep))]);
    disp('Collision!');
    for n=1:1:N
        if tr_active(n)==1
```

```
                rwait(n)=randi([0,ts],1,1);
                tr_queue(n)=tr_queue(n)+1;
                tr_rq(i,n)=rwait(n);
                coll_keep(n)=0;
            end
        end

    else
        % No Collision
        ack_cnt=ack_cnt+1;
        for n=1:N
            if tr_active(n)==1
                ack_pkts(n)=ack_pkts(n)+1;
            end
        end

    end
    disp('Transmitter Queue :');
    disp(tr_queue);
end
    del=3;
for n=1:N
    disp(['Node: ',num2str(n)]);
    T1=tot_pkts(n);A=ack_pkts(n);D=drop_pkts(n);Q=sum(tr_q(n,:));R=T1-A-Q-D;
    disp(['Total :',num2str(T1),' Ack :',num2str(A),' Dropped :',num2str(D),'
Queue:',num2str(Q),' Retrans:',num2str(R)]);
end

th=T1/(T*ts);
disp(['Throughput : ',num2str(th),' packets per second']);
disp(['System Load :',num2str(N*lambda)]);
disp(['Delay:',num2str(del)]);
```

1.7 Sample Outputs

1.7.1 IFT – Sample Output

```
---IMMEDIATE FIRST TRANSMISSION---
Enter the no.of packet radios :10
Enter the probability with which each user transmits :0.5
Enter the probability of packet generation :0.02
Enter Buffer Size at each node :5
System Load :
    0.2000

Timeslot 1:
No.of arriving pkts :
     1

Buffer statistics :
     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0
```

```
     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0
     1     0     0     0     0

No.of active transmitters :6
Collision!
Transmitter Queue :
     1     0     1     0     0     1     1     0     1     1

Timeslot 2:
No.of arriving pkts :
     1

Buffer statistics :
     1     2     0     0     0
     1     2     0     0     0
     1     2     0     0     0
     1     2     0     0     0
     1     2     0     0     0
     1     2     0     0     0
     1     2     0     0     0
     1     2     0     0     0
     1     2     0     0     0
     1     2     0     0     0

No.of active transmitters :8
Collision!
Transmitter Queue :
     2     0     2     1     1     2     2     0     2     2

Timeslot 3:
No.of arriving pkts :
     1

Buffer statistics :
     1     2     3     0     0
     1     2     3     0     0
     1     2     3     0     0
     1     2     3     0     0
     1     2     3     0     0
     1     2     3     0     0
     1     2     3     0     0
     1     2     3     0     0
     1     2     3     0     0
     1     2     3     0     0

No.of active transmitters :9
Collision!
Transmitter Queue :
     3     1     3     2     2     3     3     0     3     3

Timeslot 4:
No.of arriving pkts :
```

```
     1

Buffer statistics :
     1     2     3     4     0
     1     2     3     4     0
     1     2     3     4     0
     1     2     3     4     0
     1     2     3     4     0
     1     2     3     4     0
     1     2     3     4     0
     1     2     3     4     0
     1     2     3     4     0
     1     2     3     4     0

No.of active transmitters :9
Collision!
Transmitter Queue :
     4     2     4     3     3     4     4     0     4     4

Timeslot 5:
No.of arriving pkts :
     1

Buffer statistics :
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5

No.of active transmitters :9
Collision!
Transmitter Queue :
     5     3     5     4     4     5     5     0     5     5

Timeslot 6:
No.of arriving pkts :
     1

Buffer statistics :
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
```

```
No.of active transmitters :10
Collision!
Transmitter Queue :
     6     4     6     5     5     6     6     1     6     6

Timeslot 7:
No.of arriving pkts :
     1

Buffer statistics :
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5

No.of active transmitters :10
Collision!
Transmitter Queue :
     7     5     7     6     6     7     7     2     7     7

Timeslot 8:
No.of arriving pkts :
     1

Buffer statistics :
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5

No.of active transmitters :10
Collision!
Transmitter Queue :
     8     6     8     7     7     8     8     3     8     8

Timeslot 9:
No.of arriving pkts :
     1

Buffer statistics :
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
```

```
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5

No.of active transmitters :10
Collision!
Transmitter Queue :
     9     7     9     8     8     9     9     4     9     9

Timeslot 10:
No.of arriving pkts :
     1

Buffer statistics :
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5

No.of active transmitters :10
Collision!
Transmitter Queue :
    10     8    10     9     9    10    10     5    10    10

Node: 1
Total :10 Ack :1 Dropped :8 Queue:1 Retrans:0
Node: 2
Total :10 Ack :1 Dropped :7 Queue:1 Retrans:1
Node: 3
Total :10 Ack :0 Dropped :8 Queue:2 Retrans:0
Node: 4
Total :10 Ack :1 Dropped :8 Queue:1 Retrans:0
Node: 5
Total :10 Ack :1 Dropped :8 Queue:1 Retrans:0
Node: 6
Total :10 Ack :1 Dropped :8 Queue:1 Retrans:0
Node: 7
Total :10 Ack :0 Dropped :8 Queue:1 Retrans:1
Node: 8
Total :10 Ack :1 Dropped :8 Queue:1 Retrans:0
Node: 9
Total :10 Ack :0 Dropped :9 Queue:1 Retrans:0
Node: 10
Total :10 Ack :0 Dropped :10 Queue:0 Retrans:0
Throughput(approx.) : 34 packets per second
System Load :0.2
Delay:3
```

3.7.2 DFT – Sample Output

```
---DELAYED FIRST TRANSMISSION---
Enter the no.of packet radios :10
Enter the probability with which each user transmits :0.5
Enter the probability of packet generation :0.02
Enter Buffer Size at each node :5
Timeslot 1:
No.of arriving pkts :
     3

Buffer statistics :
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0

No.of active transmitters :5
Collision!
Transmitter Queue :
     1     0     0     1     0     0     1     0     1     1

Timeslot 2:
No.of arriving pkts :
     1

Buffer statistics :
     1     1     1     2     0
     1     1     1     2     0
     1     1     1     2     0
     1     1     1     2     0
     1     1     1     2     0
     1     1     1     2     0
     1     1     1     2     0
     1     1     1     2     0
     1     1     1     2     0
     1     1     1     2     0

No.of active transmitters :9
Collision!
Transmitter Queue :
     2     1     1     2     0     1     2     1     2     2

Timeslot 3:
No.of arriving pkts :
     2

Buffer statistics :
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
```

```
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3

No.of active transmitters :9
Collision!
Transmitter Queue :
     3     2     2     3     0     2     3     2     3     3

Timeslot 4:
No.of arriving pkts :
     4

Buffer statistics :
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3

No.of active transmitters :10
Collision!
Transmitter Queue :
     4     3     3     4     1     3     4     3     4     4

Timeslot 5:
No.of arriving pkts :
     4

Buffer statistics :
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3

No.of active transmitters :10
Collision!
Transmitter Queue :
     5     4     4     5     2     4     5     4     5     5

Timeslot 6:
```

```
No.of arriving pkts :
     3

Buffer statistics :
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3

No.of active transmitters :10
Collision!
Transmitter Queue :
     6     5     5     6     3     5     6     5     6     6

Timeslot 7:
No.of arriving pkts :
     4

Buffer statistics :
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3

No.of active transmitters :10
Collision!
Transmitter Queue :
     7     6     6     7     4     6     7     6     7     7

Timeslot 8:
No.of arriving pkts :
     3

Buffer statistics :
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
```

```
No.of active transmitters :10
Collision!
Transmitter Queue :
     8     7     7     8     5     7     8     7     8     8

Timeslot 9:
No.of arriving pkts :
     2

Buffer statistics :
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3

No.of active transmitters :10
Collision!
Transmitter Queue :
     9     8     8     9     6     8     9     8     9     9

Timeslot 10:
No.of arriving pkts :
     5

Buffer statistics :
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3
     1     1     1     2     3

No.of active transmitters :10
Collision!
Transmitter Queue :
    10     9     9    10     7     9    10     9    10    10

Node: 1
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Node: 2
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Node: 3
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Node: 4
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
```

```
Node: 5
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Node: 6
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Node: 7
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Node: 8
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Node: 9
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Node: 10
Total :31 Ack :0 Dropped :0 Queue:8 Retrans:23
Throughput : 31 packets per second
System Load :0.2
Delay:3
>>
```