Problem Statement

We need to simulate a packet radio network based on slotted aloha mechanism with different versions viz.

1. Immediate first transmission (IFT): When a packet is generated and finds the queue empty it is transmitted immediately, otherwise it is transmitted with probability *p.*
2. Delayed first transmission (DFT): All transmissions are treated equally, i.e. they are always attempted with probability *p.*

We will also simulate for protocol like that in ethernet standard like the first time a packet is transmitted, it is transmitted with probability 1; if it must be retransmitted for the kth time, it transmits with probability 1/(2^k).

Theoretical Background

In our simulation model, we consider one source which generates packets with probability λ such that Λ=N* λ varies between 0 and 1. So, the value of λ varies from 0 to 1/N. N is the number of packet radios. Each packet radio has a buffer for storing packets. Whenever packet is generated, it is stored in the buffer of the radio. In IFT, if there is only one packet in buffer then it is transmitted with probability 1, else it is transmitted with probability p. In DFT, each packet is transmitted with probability p independent of queue size.

Simulation

For simulating this model, we take time slots as variable *currentSlot* and increment it at end of every slot. At beginning of the slot, we check for the new packets generated for each packet radio and add them to the buffer. If the buffer is full then packet generated is lost. We also store the value to *currentSlot* variable as timestamp for generation of packet. We are also giving each packet a id, so that later on we can find the timestamp of generation of packet for calculating the Delay. Now, the packets are present in buffer and radio packet will try to transmit the packet with some probability based on version of slotted aloha i.e. IFT or DFT. We are keeping track of status of packet radio for detecting collision. If more than one radios try to transmit the packet then collision will occur and packet will stay in buffer for retransmission in next slot. If collision is not occurred then packet will be transmitted to the base station and acknowledgement is received. Also, the buffer is left-shifted by one to make room for new packet. At the end, we calculate the throughput which is equal to packets acknowledged per number of slots, mean delay which is the average value of number of slots required for transmission of one packet and number of packets lost due to buffer overflow.

Code:

```
clc
clear all
close all

method=input('Enter the method number: ');
% 1:IFT, 2:DFT, 3:Ethernet

buffersize=5;
N=2; % Number of packet radios
p=0.057; % Probability that radio will transmit the packet
```

```
B=zeros(N,buffersize); % Matrix for queue of packet radios
lamda=0.3; % Anything such that N*lamda varies from 0 to 1 so lamda is in
range 0 to 1/N
packetsId=zeros(N,buffersize); % Matrix for storing the packet ids
noOfRetransmissions=zeros(N,1); % Vector for storing value of
retransmissions

simulationTime=5000; % Total number of slots to simulate
currentSlot=0; % Value of current slot
radioStatus = zeros(1,N);
% 0: not transmitting
% 1: transmitting with probability p
packetLossCount=0;
packetsGenerated=0;
packetsAcknowledged=0;
packetsCollisionCount=0;

while currentSlot<simulationTime
    currentSlot=currentSlot+1; % Increment the slot value

    % Check if source generates a new packet for each radio
    for i=1:N
        if rand(1)/N <= lamda
            packetsGenerated=packetsGenerated+1;
            % Store the value of currentslot as timestamp
            packetGenerateTimeStamp(packetsGenerated)=currentSlot;
            temp=find(B(i,:)==0);
            if isempty(temp) % means buffer is overflown
                packetLossCount=packetLossCount+1;
            else
                B(i,temp(1))=1; % store the packet in buffer
            end
            temp=find(packetsId(i,:)==0);
            if ~isempty(temp)
                packetsId(i,temp(1))=packetsGenerated;
            end
        end
    end

    % Check which radio packet will transmit packet with probability p
    for i=1:N
        if sum(B(i,:)==0)~=5
            if method==1
                if(B(i,1)==1 && B(i,2)==0 && B(i,3)==0 &&B(i,4)==0
&&B(i,5)==0  ) % means there is only one packet in queue and it will be
transmitted immidiately
                    radioStatus(1,i)=1;
                else
                    if rand(1) <= p % packet will transmitted with
probability p
                        radioStatus(1,i)=1;
                    end
                end
            end

            if method==2
                if rand(1) <= p %packet will transmitted with probability p
                    radioStatus(1,i)=1;
                end
            end
```

```
            if method==3
                if rand(1) <= (1/2^(noOfRetransmissions(i,1))) % packet
will transmitted with probability (1/2^k)
                    radioStatus(1,i)=1;
                end
            end
        end
    end

    % Check for collision of transmitting radios
    for i=1:N
        if sum(radioStatus==1)==1 && radioStatus(i)==1 % means there is no
collision and base stations sends achnolegment
            packetsAcknowledged=packetsAcknowledged+1;
            packetAcknowledgeDelay(packetsAcknowledged)=currentSlot-
packetGenerateTimeStamp(packetsId(i,1)); % this calculated the number of
slots required for complete transmission for particular packet
            B(i,:)=[B(i,2:end),0]; % left shift the buffer and make room
for new packet
            packetsId(i,:)=[packetsId(i,2:end),0];
            if method==3
                noOfRetransmissions(i,:)=[noOfRetransmissions(i,2:end),0];
            end
        end
    end
    if sum(radioStatus==1) ~= 1 && sum(B(1,:)==0)~=5 && sum(B(2,:)==0)~=5 %
means collision has occured
        packetsCollisionCount=packetsCollisionCount+1;
        if method==3
            for i=1:N
                if radioStatus(i)==1
                    noOfRetransmissions(i,1)=noOfRetransmissions(i,1)+1; %
increment number of retransmissions
                end
            end
        end
    end
    radioStatus = zeros(1,N);
    bufferOccupancyStat(currentSlot)=sum(sum(B==1));
end

% Calculate throughput
throughput = packetsAcknowledged / currentSlot
% Calculate mean delay for packet transmission
meanDelay=mean(packetAcknowledgeDelay)
% Calculate number of packets lost
packetlosscount=packetLossCount
% Calculate buffer occupancy stats
averageBufferOccupancy=mean(bufferOccupancyStat)/N
```

Output:

For Buffer size=5, p=0.05, λ=0.3

		Throughput	Delay	Packets Generated	Packets Lost	Average Buffer Occupancy
N=2	IFT	0.0983	103.60	6034	5559	4.9074
	DFT	0.1016	95.96	6036	5510	4.9058
	Ethernet	0.6026	8.93	5985	2930	2.5307

For Buffer size=5, p=0.05, λ=0.03

		Throughput	Delay	Packets Generated	Packets Lost	Average Buffer Occupancy
N=10	IFT	0.3056	155.88	15098	13522	4.8779
	DFT	0.3370	141.90	14934	13200	4.8752
	Ethernet	0.4892	60.40	14891	12401	4.4366

For Buffer size=2, p=0.05, λ=0.3

		Throughput	Delay	Packets Generated	Packets lost	Average Buffer Occupancy
N=2	IFT	0.4016	6.77	6024	4012	1.3212
	DFT	0.0930	40.99	5985	5516	1.9082
	Ethernet	0.6094	1.83	6004	2955	1.0560

For Buffer size=2, p=0.05, λ=0.03

		Throughput	Delay	Packets Generated	Packets lost	Average Buffer Occupancy
N=10	IFT	0.3978	44.19	15084	13077	1.7805
	DFT	0.3136	59.49	14979	13393	1.8904
	Ethernet	0.4460	21.44	14886	12637	1.7719

For Buffer size=1, p=0.05, λ=0.3

		Throughput	Delay	Packets Generated	Packets lost	Average Buffer Occupancy
N=2	IFT	0	5000	6041	6039	0.9996
	DFT	0.092	19.68	5955	5487	0.9188
	Ethernet	0.5802	1.37	6000	3098	0.5232

For Buffer size=1, p=0.05, λ=0.03

		Throughput	Delay	Packets Generated	Packets lost	Average Buffer Occupancy
N=10	IFT	0	5000	14998	14988	0.9997
	DFT	0.3040	29.37	14912	13383	0.8988
	Ethernet	0.4112	14.57	15079	13016	0.8665

The simulation data obtained for IFT, DFT and Ethernet for different buffer sizes 1,2 and 5 is shown in tables above. The throughput for both IFT and DFT increases as value of N increases until one threshold and then it decreases as value of N increases. The packets lost is dependent on the buffer size. When the buffer size increases the packet lost count is less. The relation between the throughput and number of packet radios is shown in figure 1.

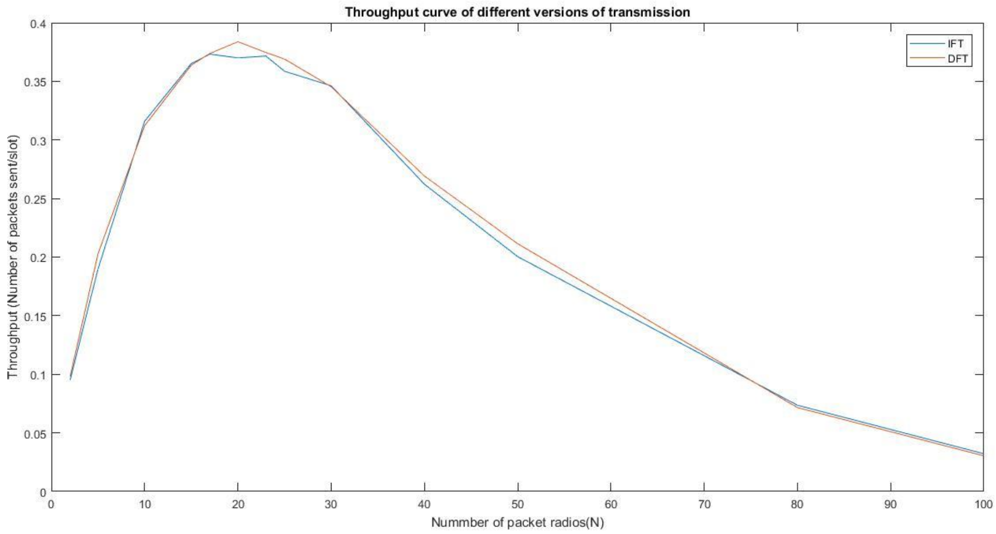

Figure 1 Relation between throughput and # packet radios for IFT and DFT

The relationship between the number of packets lost and probability of packet generation is shown in figure 2. From the plot, we can clearly see that the number of packets lost is independent of probability of packet generation.

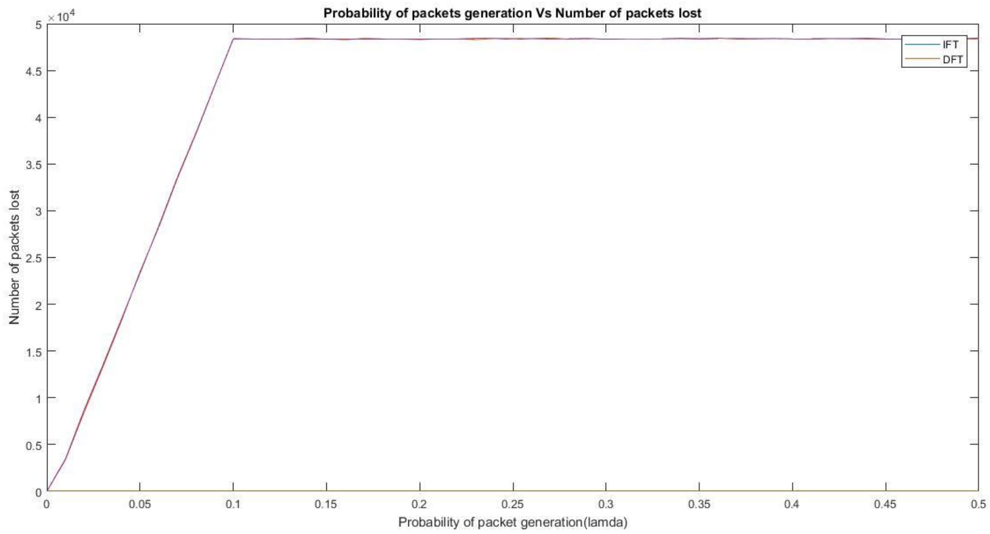

Figure 2 Relationship between Probability of packets generation and Number of packets lost

The relationship between the throughput and probability of packet generation p for N=2 and N=10 is shown in figure 3 and 4 respectively. For N=2, the throughput increases till one value and then decreases. For N=10, the throughput increases exponentially and then till decreases exponentially. For ethernet protocol, the throughput is independent of transmission probability p because the transmission probability for ethernet is function of number of retransmissions.

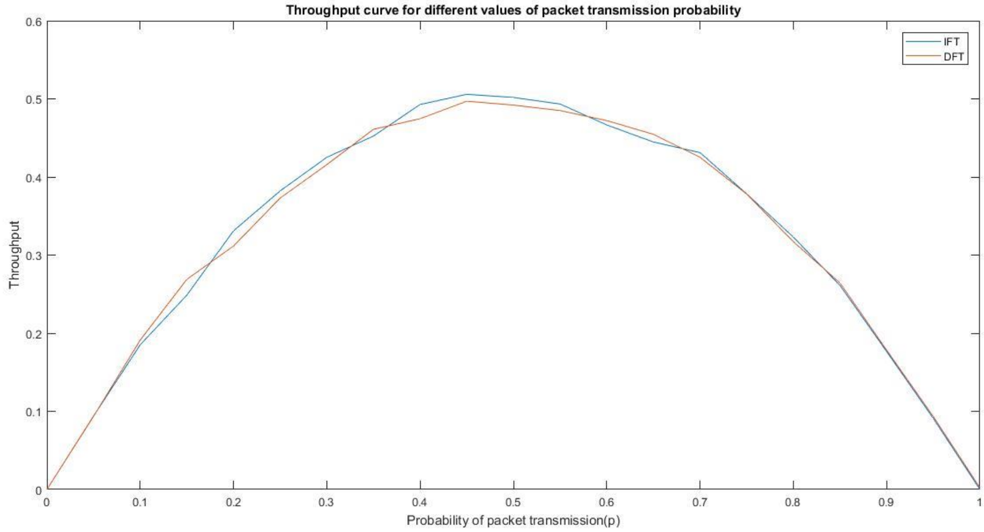

Figure 3 Relationship between throughput curve and packet transmission probability p for N=2

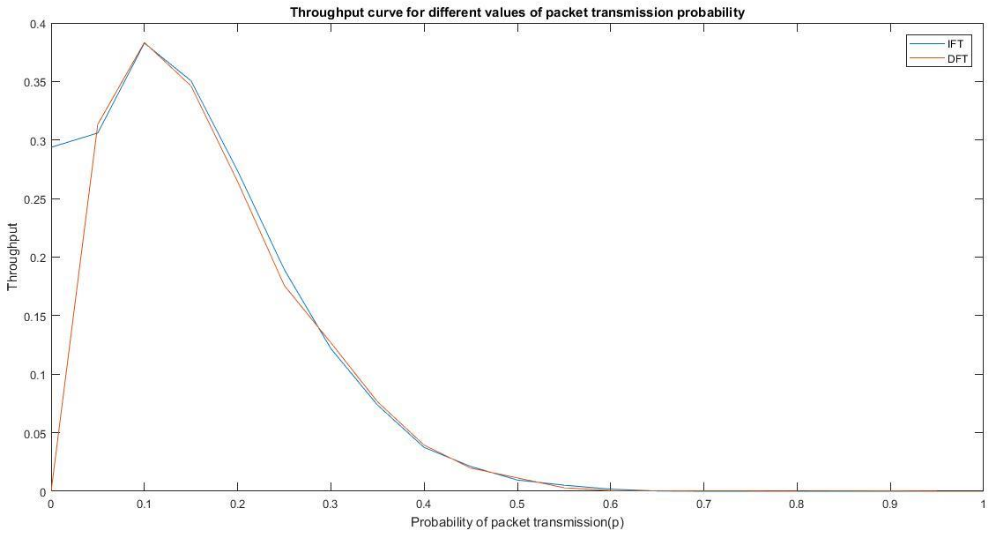

Figure 4 Relationship between throughput curve and packet transmission probability p for N=10