Name: Gunjan Jhawar

Subject code: EE 511

Simulation of Scholastic System

USCID: 1857534264

Project: 1

Aim:

Consider a set of N ***packet radios*** (i.e. radios that transmit data in fixed size packets), that share a common radio channel, sending information to a base station. There is a data source (perhaps a sensor, or a human) connected to each packet radio that generates packets from time to time. Time can be considered to be slotted (where a slot corresponds to the time it takes one nodes to transmit one packet of data.) The radios have no way of controlling who should access the channel in any slot and consequently the protocol is that a radio that has a packet waiting to be sent will transmit that packet in the next slot with some probability p . If exactly one radio transmits in a slot we can assume that the base station will receive the packet

of data and will indicate that it did so by broadcasting out an ***acknowledgement*** packet (over a different channel) so that the source will know its transmission was successful and it can delete that packet from it queue of waiting packets. If more than one radio transmits, the base station will receive a garbled signal not be able to decode any of the packets that were transmitted - this is called a ***collision***.

This mechanism is called ***Slotted Aloha*** and versions of it are used on the control channels of cellular telephone systems. There are two main versions: IFT (Immediate first transmission) and DFT (delayed first transmission.) In IFT, when a packet is generated and finds the queue empty it is transmitted immediately (in the next slot),

otherwise it is transmitted with probability p . In DFT, all transmissions are treated equally, i.e. they are
always attempted with probability p . Try both.

Parameters

N - the number of packet radios

p - the probability that a packet radio that has a non-empty buffer will transmit (suggested values – try a range of values and see what happens.) (Depends on the version of ALOHA)

B - buffer size, i.e., the number of packets that can be stored awaiting transmission at each packet radio. The packets are transmitted from the buffer in a FIFO fashion.

Performance Statistics to Measure

Throughput □ in packets per second
Delay T in slots. Time from generation of a packet until it is successfully transmitted Buffer occupancy statistics
Packet Loss due to buffer overflows.

Software Used:

MATLAB

Procedure:

A. In DFT transmission are send with same probability. We create two array using rand. Length of the array depends on the number of radios. We also define value of 'lambda' and 'probability of transmission'. We define an array of N zeros used for counting the buffer for each time slot. For transmitting to be successful, we have various conditions to fulfill:

1. When no packet is generated and some audios are transmitted successfully, if buffer is not zero for some, it transmits and buffer decrements by 1. Otherwise buffer will stay the same.
2. When one packet is generated and audio is transmitted successfully, buffer decrements by 1. Otherwise buffer size increments by 1.
3. Multiple packets are generated, only one is transmitted. Buffer size will decrement by 1.
4. No matter how many packets are generated, if more than one want to transmit, collision occurs. Buffer increments by 1.

Creating for loops with if-else conditions to fulfill all above.
Defining parameters as given:
packet =Number of packet radios
p = Probability that nonempty buffer will transmit
B = Buffer size
sim_time = Simulation time
lambda = Probability that new packet is generated

B. The second one is Immediate First transmission. In it, when a packet is generated it finds the queue empty and immediately transmits, otherwise it transmits according to the probability of the packet. The difference is the buffer conditions. For this method, we will pick some p values and lambda to inspect the performance of the network.

Observation and Analysis:

A) DFT Observation

1)From the result that we get, we draw the following table: (N=2)

P	0.1	0.2	0.3
Throughput	150	320	329
Delay	6.2	3.22	2.5
Lost	441	279	230

As P increases, throughput increases, lost decreases and average delay decreases which means performance improvement of the network.

2)From the result that we get, we draw the following table: (N=2)

Lambda	0.1	0.2	0.3
Throughput	167	320	398
Delay	5.2	3.22	2.5
Lost	10	59	230

As lambda increases, throughput increases, average delay decreases and lost increases. As when transmitting probability is more than generating probability whatever packets are generated will be sent successfully which validates the result we get.

3)From the result that we get, we draw the following table: (N=10)

P	0.1	0.4	0.5

Throughput	198	32	19
Delay	4.9	25.88	115
Lost	41	211	230

When P increases, throughput decreases, lost increases, buffer value increases and average value increases. From the earlier result, we increased the value of N which means collision increases in this case causing the decline of the performance of the network.

4)From the result that we get, we draw the following table: (N=10)

lambda	0.01	0.02	0.03
Throughput	4	11	8
Delay	200	91	109
Lost	86	140	220

Throughput is small and average delay is high. Lost is increasing thus performance of the system is bad. As more and more packets are generated more collision occurs. Packets would not be send out and will stay in buffer and then dropped.

5)From the result that we get, we draw the following table: (N=10)

B	2	3	4
Throughput	9	15	20
Delay	110	100	88
Lost	291	179	133

When buffer increases, lost value drops, throughput increases and average delay drops. As there is space for packets to stay rather being dropped like earlier case.

B.Comparison of both method: DFT and IFT

When N B and lambda are fixed but change P(N =2)

Method	DFT	IFT
Throughput	150	215
Average_Delay	6.2	4.2
Lost	441	320

When N B and P are fixed but change lambda (N =10)

Method	DFT	IFT
Throughput	4	16
Average_Delay	200	125.2
Lost	86	21

Result:

A)

1. When N B and lambda are fixed but change P(N =2)

Input:

```
Number of packet radios 2
Probability that nonempty buffer will transmit 0.1
Buffer size2
Simulation time500
Probability that new packet is generated0.3
```

Output:

```
lost_value = 441
Through = 150
Average_Delay = 6.2
```

Input:

```
Number of packet radios 2
Probability that nonempty buffer will transmit 0.2
Buffer size2
Simulation time500
Probability that new packet is generated0.3
```

Output:

```
lost_value = 279
Through = 320
Average_Delay = 3.22
```

Input:

```
Number of packet radios 2
Probability that nonempty buffer will transmit 0.3
Buffer size2
Simulation time500
Probability that new packet is generated0.3
```

Output:

```
lost_value = 230
Through = 329
Average_Delay = 2.5
```

2. When N B and P are fixed but change lambda (N =2)

Input:

```
Number of packet radios 2
Probability that nonempty buffer will transmit 0.4
Buffer size2
```

Simulation time500
Probability that new packet is generated0.1

Output:

lost_value = 10
Through = 167
Average_Delay = 5.2

Input:

Number of packet radios 2
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.2

Output:

lost_value = 59
Through = 320
Average_Delay = 3.22

Input:

Number of packet radios 2
Probability that nonempty buffer will transmit 0.2
Buffer size2
Simulation time500
Probability that new packet is generated0.3

Output:

```
lost_value = 230
Through = 398
Average_Delay = 2.5
```

3. When N B and lambda are fixed but change P(N =10)

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.1
Buffer size2
Simulation time500
Probability that new packet is generated0.01
```

Output:

```
lost_value = 41
Through = 198
Average_Delay = 4.9
```

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.01
```

Output:

```
lost_value = 211
Through = 32
Average_Delay = 25.88
```

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.5
Buffer size2
Simulation time500
Probability that new packet is generated0.01
```

Output:

```
lost_value = 230
Through = 19
Average_Delay = 115
```

4. When N B and P are fixed but change lambda (N =10)

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.01
```

Output:

```
lost_value = 86
Through = 4
Average_Delay = 200
```

Input:

Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.02

Output:

lost_value = 140
Through = 11
Average_Delay = 91

Input:

Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.03

Output:

lost_value = 220
Through = 8
Average_Delay = 109

5. When N lambda and P are fixed but change B (N =10)

Input:

Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500

```
Probability that new packet is generated0.03
```

Output:

```
lost_value = 291
Through = 9
Average_Delay = 110
```

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size3
Simulation time500
Probability that new packet is generated0.03
```

Output:

```
lost_value = 179
Through = 15
Average_Delay = 100
```

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size5
Simulation time500
Probability that new packet is generated0.03
```

Output:

```
lost_value = 133
Through = 20
Average_Delay = 88
```

B)

1. When N B and lambda are fixed but change P(N =2)
Input:

Number of packet radios 2
Probability that nonempty buffer will transmit 0.1
Buffer size2
Simulation time500
Probability that new packet is generated0.3
Output:

lost_value = 320
Through = 215
Average_Delay = 4.2

Input:

Number of packet radios 2
Probability that nonempty buffer will transmit 0.2
Buffer size2
Simulation time500
Probability that new packet is generated0.3

Output:

lost_value = 389
Through = 376
Average_Delay = 2.45

2. When N B and P are fixed but change lambda (N =2)

Input:

```
Number of packet radios 2
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.1
```

Output:

```
lost_value = 239
Through = 312
Average_Delay = 3.2
```

Input:

```
Number of packet radios 2
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.2
```

Output:

```
lost_value = 389
Through = 352
Average_Delay = 2.278
```

3. When N B and lambda are fixed but change P(N =10)

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.1
Buffer size2
Simulation time500
Probability that new packet is generated0.01
```

Output:

```
lost_value = 211
Through = 67
Average_Delay = 15.209
```

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.2
Buffer size2
Simulation time500
Probability that new packet is generated0.01
```

Output:

```
lost_value = 218
Through = 58
Average_Delay = 16.65
```

4. When N B and P are fixed but change lambda (N =10)

Input:

```
Number of packet radios 10
```

```
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.01
```

Output:

```
lost_value = 211
Through = 16
Average_Delay = 125.2
```

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
Probability that new packet is generated0.02
```

Output:

```
lost_value = 340
Through = 16
Average_Delay = 122.56
```

5. When N lambda and P are fixed but change B (N =10)

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size2
Simulation time500
```

```
Probability that new packet is generated0.03
```

Output:

```
lost_value = 231
Through = 7
Average_Delay = 187.2
```

Input:

```
Number of packet radios 10
Probability that nonempty buffer will transmit 0.4
Buffer size3
Simulation time500
Probability that new packet is generated0.03
```

Output:

```
lost_value = 231
Through = 16
Average_Delay = 99
```

Conclusion

We see that IFT has better network performance than DFT because IFT sends packets right away and not need to work on whether to send the package or not as in DFT.