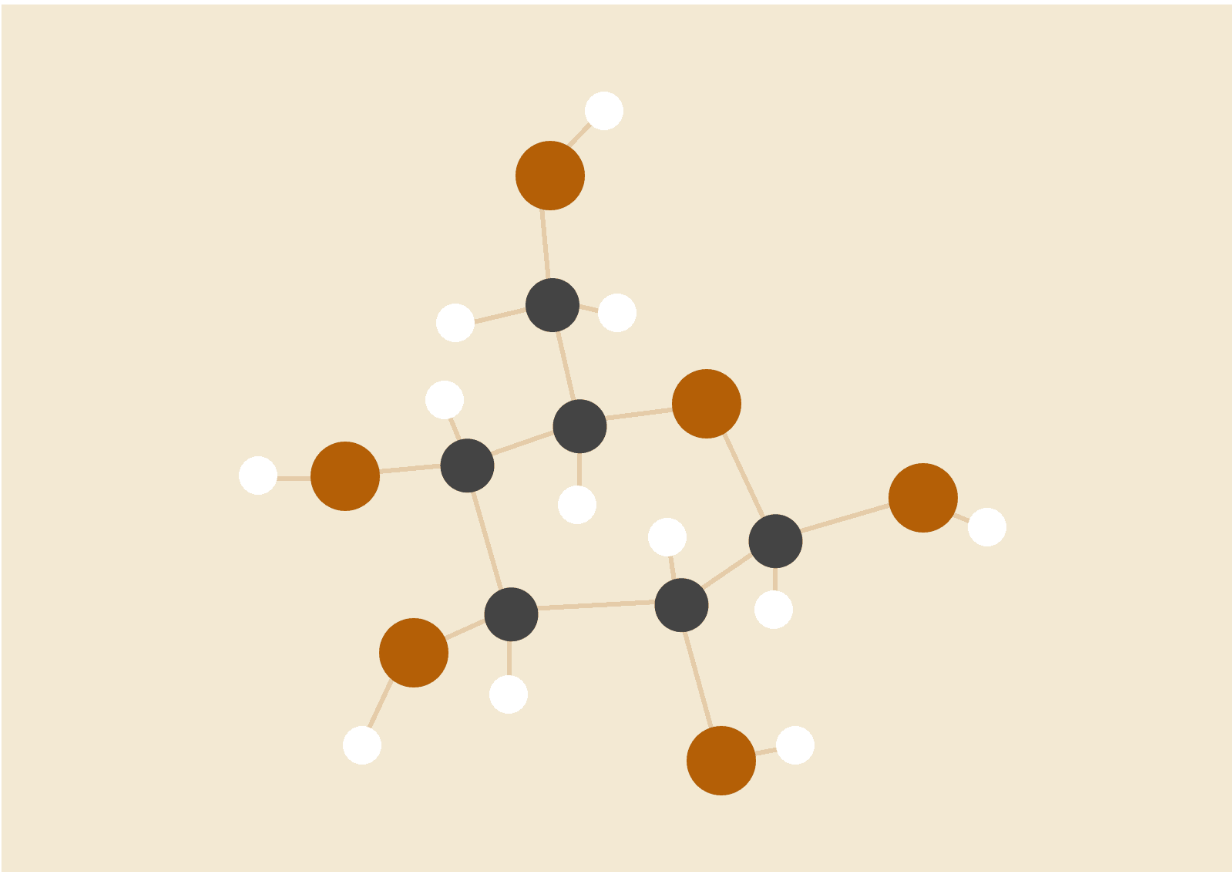
NETWORKING LAB REPORT

CLASS *BCSE III* ***SEM*** *FIFTH* ***YEAR*** *2021*



NAME Neeladri Pal

ROLL 001910501015

GROUP A1

ASSIGNMENT - 3

PROBLEM STATEMENT

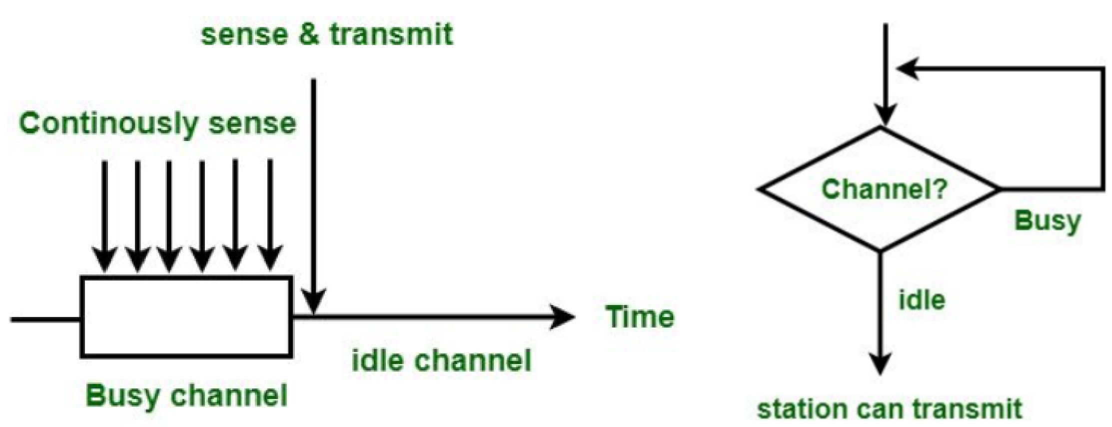
Implement 1-persistent, non-persistent and p-persistent CSMA techniques.

In this assignment, you have to implement 1-persistent, non-persistent and p-persistent CSMA techniques. Measure the performance parameters like throughput (i.e., average amount of data bits successfully transmitted per unit time) and forwarding delay (i.e., average end-to-end delay, including the queuing delay and the transmission delay) experienced by the CSMA frames (IEEE 802.3). Plot the comparison graphs for throughput and forwarding delay by varying p. State your observations on the impact of performance of different CSMA techniques.

CSMA METHODS

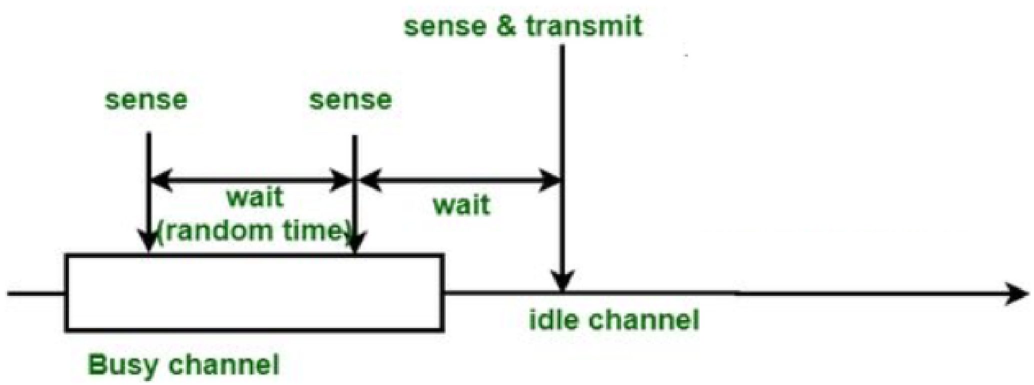
One-Persistent CSMA:

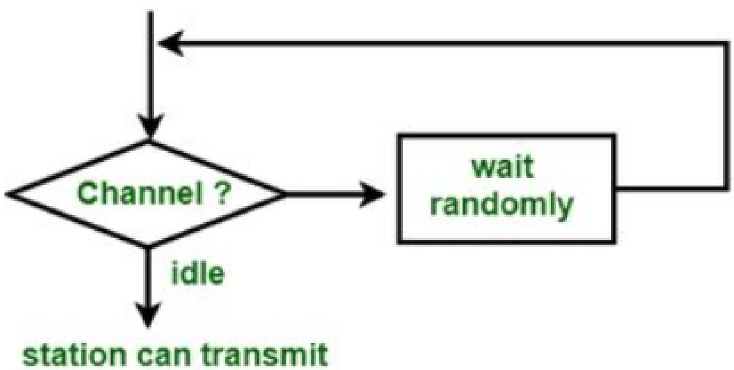
In this method, the station continually senses the channel and after it finds the line idle, it sends its frame immediately (with probability I). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately. Ethernet uses this method.



Non-Persistent CSMA:

In this method, a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. The nonpersistent approach reduces the chance of collision because it is unlikely that two or more stations will wait the same amount of time and retry to send simultaneously. However, this method reduces the efficiency of the network because the medium remains idle when there may be stations with frames to send.

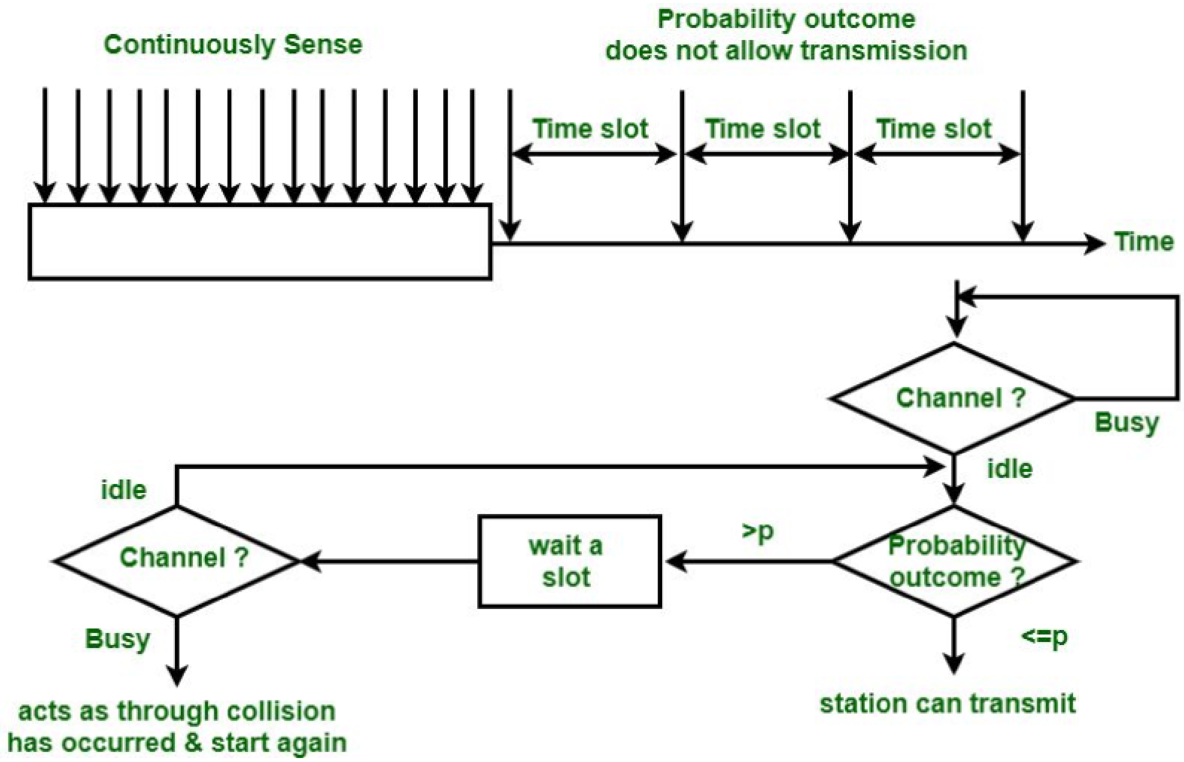




p-Persistent CSMA:

The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves efficiency. In this method, after the station finds the line idle it follows these steps:

1. With probability p, the station sends its frame.
2. With probability q = 1 - p, the station waits for the beginning of the next time slot and checks the line again.
    a. If the line is idle, it goes to step 1.
    b. If the line is busy, it acts as though a collision has occurred and uses the back- off procedure.



DESIGN

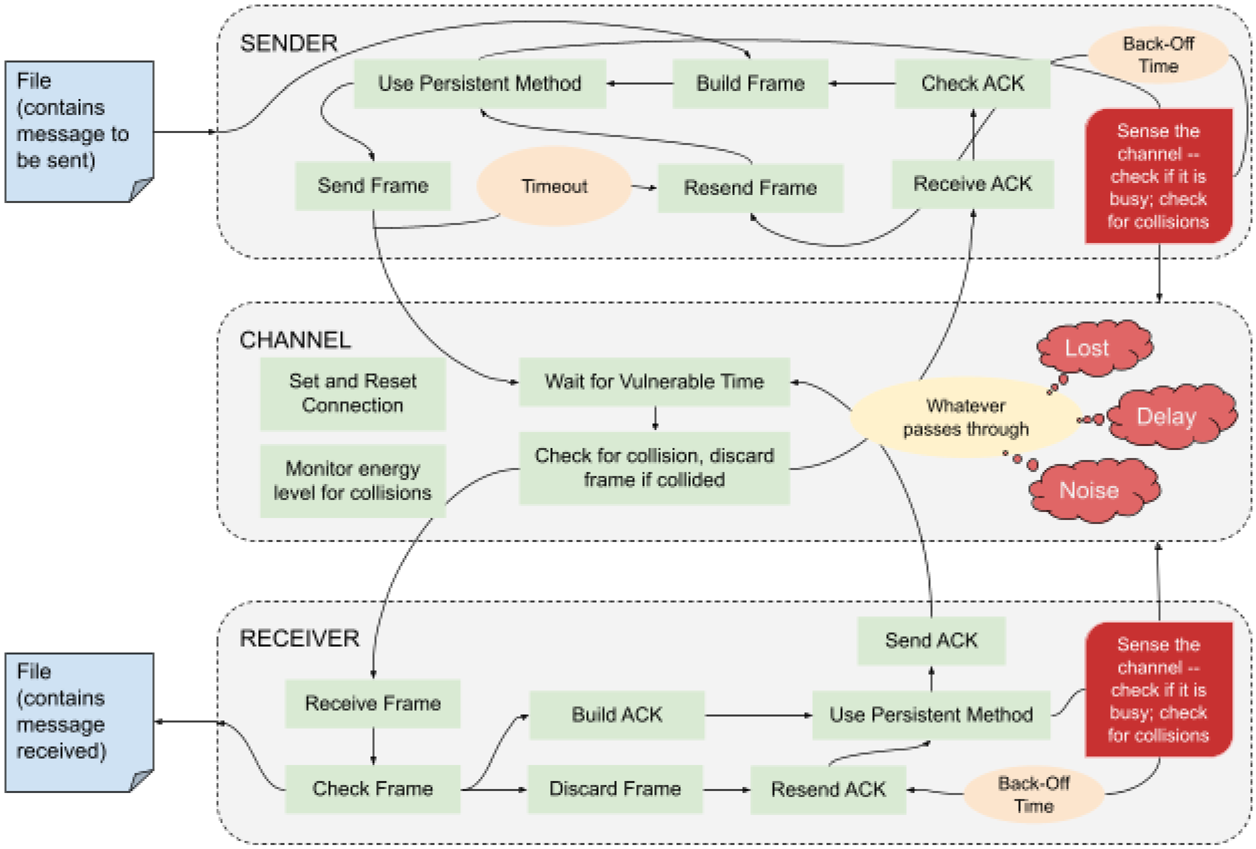
IEEE 802.3 Ethernet Frame Format								
Preamble	Frame Delimiter	Destination Address	Sender Address	Type	Sequence Number	Data	CRC Check Bits	Total
7	1	6	6	1	1	46	4	72

*all sizes are in bytes

CRC key used - 100000100110000010001110110110111 (CRC - 32)

Preamble and Frame Delimiter - used for synchronization purposes
Type - 0 (Data), 1 (ACK)
Data - contains the contents of the message

CSMA WITH STOP-N-WAIT MECHANISM



Sender reads in the **message** to be sent from the **input file**, makes a **packet** out of it, adds **redundant bits**, converts it into **frame** and sends it to the channel. While sending the frame, a suitable **persistence method** is chosen to prevent multiple frames from colliding. The sender keeps a **copy** of the sent frame, lest needed later. The **noisy channel** introduces error, delay or the frame may even be lost.

If anyhow the frame lands up in the receiver's arena, the receiver checks it for any sort of **error**. If everything is fine, it sends an **ACK**. The ACKs/NAKs pass through the channel and reach the sender.

In spite of checks, the frames may **collide** and get corrupted. So the stations must check for collisions continually after sending and undergo suitable **back-off procedure**.

The sender then checks it and either **sends** the next frames or **resends** the previous (lost in transit) frames. Also, the frames may be resent by the sender on **timeout**. The receiver, when it receives the correct frames, writes them to an output file. At the end, this file contains the total **message received**.

RESULTS

Bandwidth - 1000 bps

Throughput - Average number of successful transmissions per frame slot

Average Delay - Average end-to-end delay for a frame to be transmitted

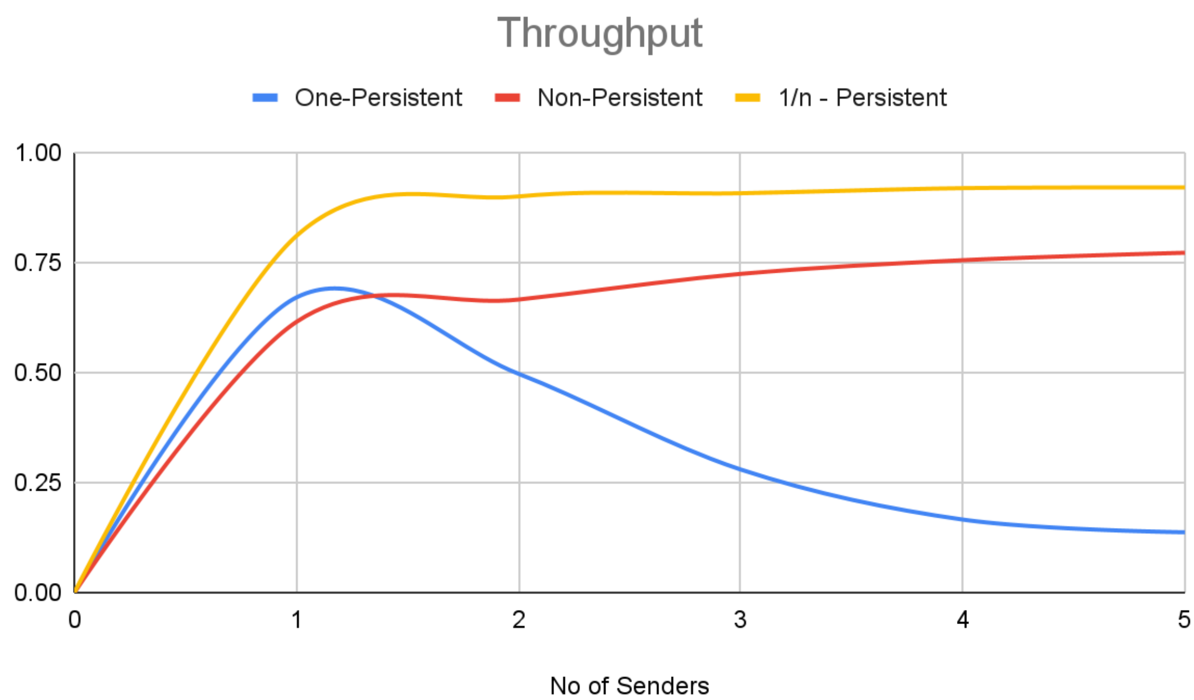
One-Persistent Method						
No of senders	Frames sent	Effective Frames sent	Total time taken (in sec)	Collision	Throughput (bits / slot)	Average Delay (sec / frame)
1	20	14	24	1	0.67032	1.714285714
2	28	14	32	7	0.49677	2.285714286
3	50	14	57	13	0.27943	4.071428571
4	84	14	97	21	0.16573	6.928571429
5	102	14	117	31	0.13688	8.357142857

Non-Persistent Method						
No of senders	Frames sent	Effective Frames sent	Total time taken (in sec)	Collision	Throughput (bits / slot)	Average Delay (sec / frame)
1	21	14	26	1	0.61478	1.857142857
2	23	14	24	5	0.66541	1.714285714
3	23	14	22	7	0.72375	1.571428571
4	21	14	21	7	0.75481	1.528782999
5	21	14	20	7	0.77167	1.428571429

n - no of senders

1/n - Persistent Method						
No of senders	Frames sent	Effective Frames sent	Total time taken (in sec)	Collision	Throughput (bits/slot)	Average Delay (sec/frame)
1	16	14	19	1	0.81005	1.357142857
2	17	14	17	2	0.89957	1.214285714
3	20	14	17	5	0.90683	1.214285714
4	18	14	17	4	0.91839	1.214285714
5	18	14	17	4	0.91998	1.214285714

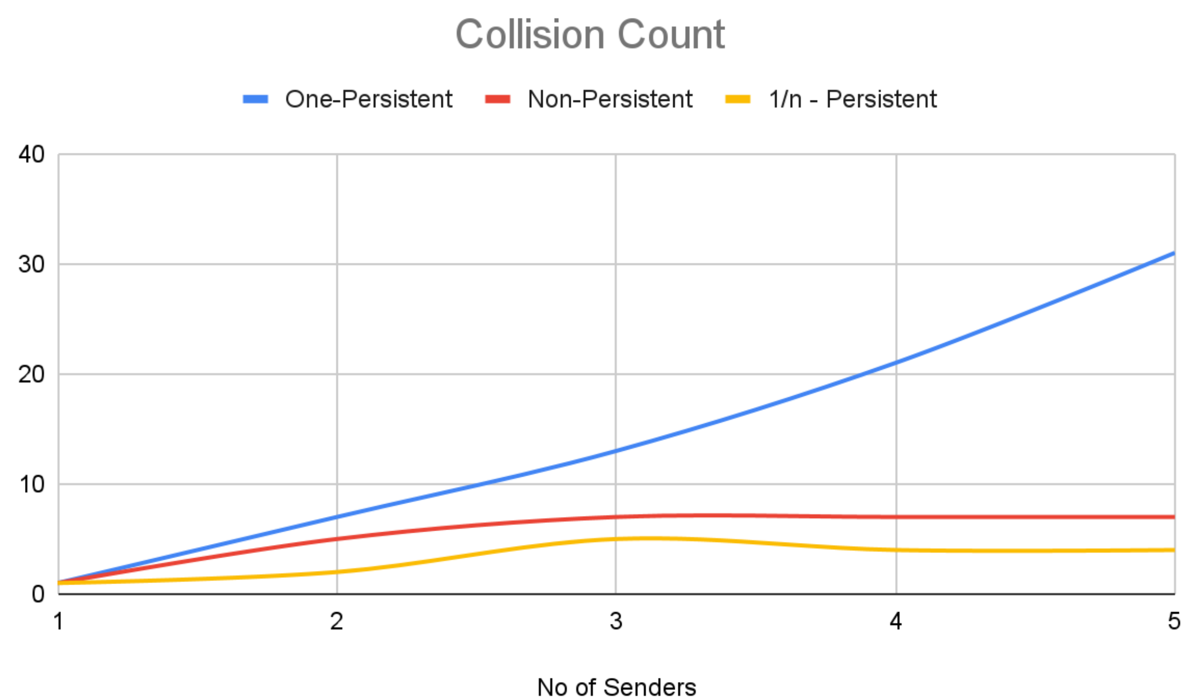
ANALYSIS



The throughput of one-persistent method is the lowest among all and with increasing number of senders it decreases and approaches 0. Non persistent is far better than this and 1/n-persistent is the best. At a higher number of sending stations throughput of both non persistent and 1/n -persistent become almost constant.

- In one-persistent method, the number of collisions increases rapidly with the number of senders, so throughput drops significantly.
- In non-persistent method, collisions decrease due to random channel sensing. It is very less probabilistic to get two frames to collide even if the number of senders are more. So throughput increases clearly.
- In the 1/n-persistent method, the probability to send the packet depends on the number of senders. A sender thus gets a fair share of access to the channel. So even when the number of senders increases, there is no effect on throughput. It remains constant.

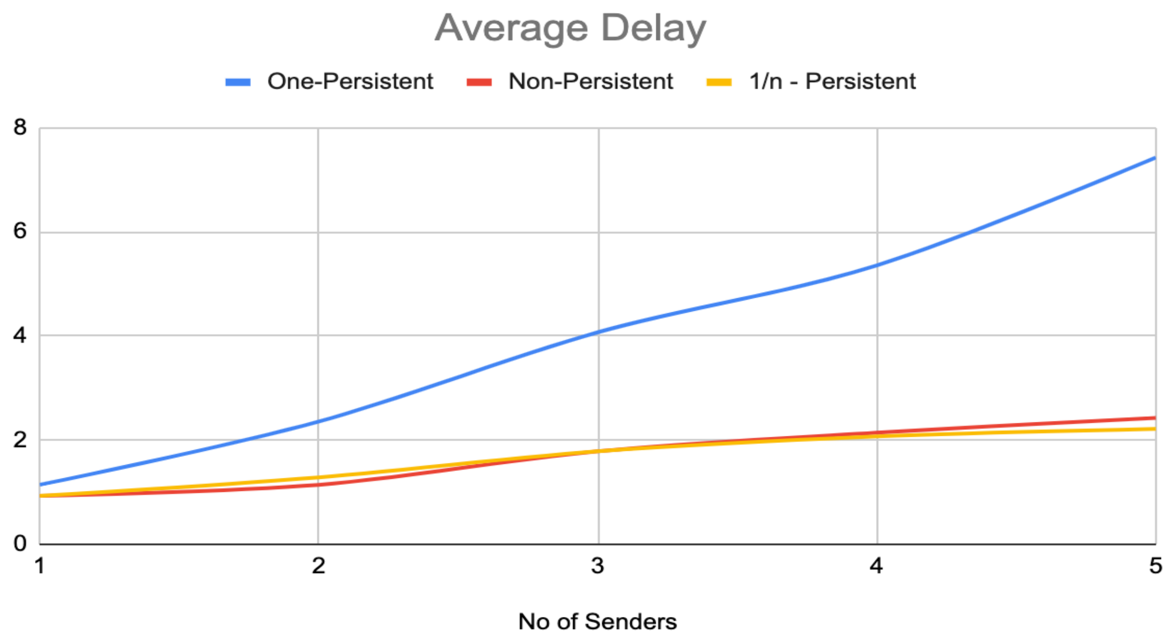
Throughput : 1/n-Persistent > Non-Persistent > One-persistent



The number of collisions per sender increases rapidly with an increasing number of sending stations for one persistent method. But for non persistent and p persistent methods the number of collisions increases very slowly. For 1/n - persistent, the growth in the number of collisions is very low.

- As in one persistent method a frame is transmitted immediately after it senses the channel idle, it has the maximum chances of getting collided in the channel. When the number of senders increases, collisions increase exponentially.
- In non persistent method, the sender waits for a random time when the channel is found to be busy. As they wait for a random time, it reduces the chances of sensing the idle channel simultaneously thus reducing the number of collisions. Average number of collisions remains almost the same with an increasing number of senders.
- 1/n- persistent method uses combination of above two methods. When it senses an idle channel, it doesn't transmit immediately. It generates a random value 'x' which must be less than p (= 1/n) to transmit the frame. If 'x' exceeds p, then it waits for a timeslot then again senses the channel and repeats the above process. It is unlikely for different senders to generate 'x' (<p) in the same slot which reduces collision probability.

Collisions : 1/n-Persistent < Non-Persistent < One-persistent



- In one persistent method, as collisions increase with an increasing number of sending stations, delay increases exponentially.
- In non persistent method, with an increasing number of sending stations, delay increases slowly.
- 1/n-persistent method gives the best throughput. Here successful transmission time increases linearly with an increasing number of sending stations. It takes almost the same time as a non persistent method as with increasing number of stations the value of p decreases, so in spite of the channel being idle no sender may send its frame which indeed increases the time unnecessarily.

Average Delay - 1/n-Persistent ≈ Non-Persistent < One-persistent

COMMENTS

The assignment gives insights into different CSMA techniques used in a noisy shared channel. It helps to understand how collisions are resolved in a channel.

The code can be tested for more analysis with more senders to generate more accurate results, by varying the noise on channel, and by varying frame times.

The code is very hard to implement and even harder to debug. Waiting hours to debug a synchronization issue, changing the design all over again to make it scalable is normal. At last when everything works, it works like magic.