COL334-Networks

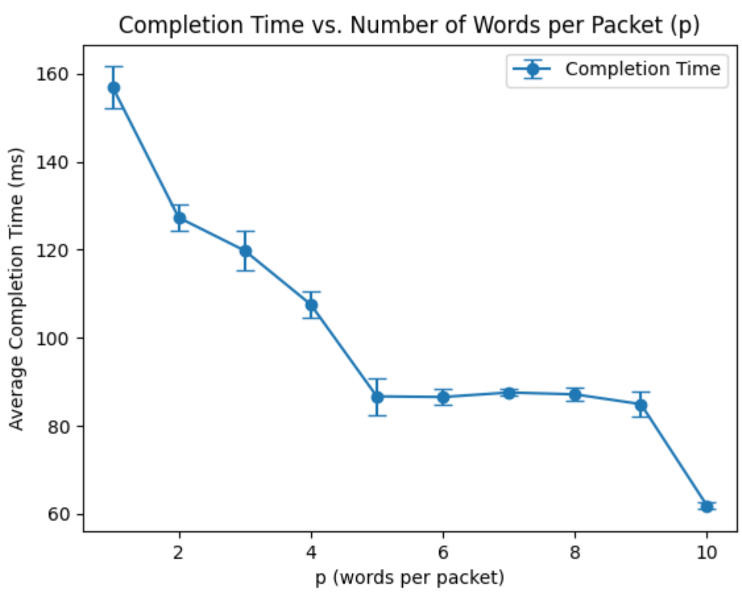
Assignment-2 : Report

Sanyam Garg: 2022CS11078 Aneeket Yadav: 2022CS11116

Introduction:

This report explains the design choices involved in implementation of various protocols and the observations/insights acquired while doing the same as well as their potential explanations.

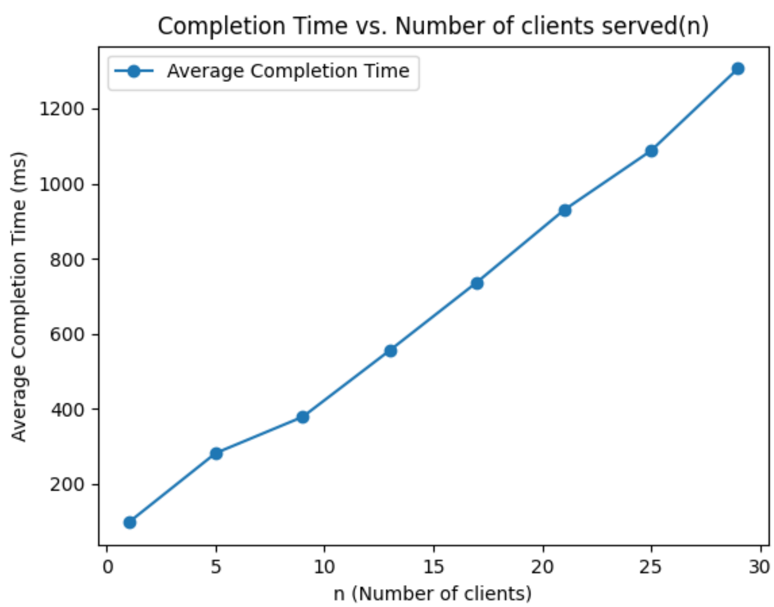
Part-1



We observe a clear decreasing trend indicating demonstrating the effect of packet size on the time taken to transfer the corpus from the server to the client. This is expected as packet size increases, the number of packets required to transmit a fixed amount of information will decrease. Since delay of packet can be considered independent of size in this case (delay would consist of transmission delay and propagation delay in this case, the former being independent of packet size, and if we ignore the case of fragmentation and reassembly, the latter as well), the average completion time for a client decreases. The above analysis can be considered complete as no interference with other clients on the network is assumed.

a. In case of a single client, bandwidth is not expected to be a limiting factor for transmission and thus should not affect completion time.
b. Latency of the network will decrease the propagation delay and thus decrease completion time.
c. Decrease in k would mean that the number of packets required to transmit k words at a time would decrease. Hence, this would decrease the completion time.

Part-2



By case analysis, one of the following must occur:

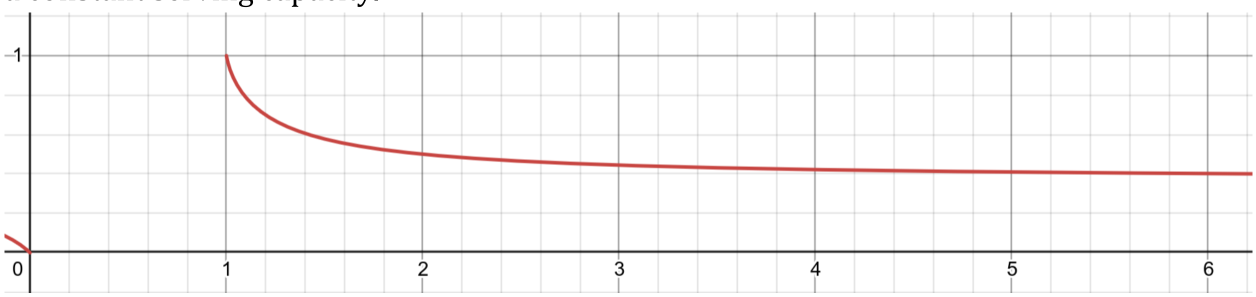
a. Number of threads at server being greater than the number of cores: Then due to context switching, true parallelism cannot be achieved and thus the average completion time increases instead of staying constant.
b. Number of threads at server being less than or equal to the number of cores: Even if this condition is met, the shared memory channel becomes a performance bottleneck. Therefore, more than one client being served parallelly will also delay each other.

In either case, average completion time increases with the number of clients.

Part-3

For comparison, we considered three values of T in our analysis: 1ms, 5ms and 50ms. However, since service times of clients are in the microseconds range, it is not practical to compare them when T is sufficiently large, say 50ms.

a. The efficiency of Slotted Aloha protocol is (1-1/n) ^(n-1) where n nis the number of clients being served. Even for small values of n, the quantity is very near to its asymptote y=1/e. Thus, we may assume that the protocol has approximately constant efficiency i.e. it is able to efficiently serve only a certain number of clients. As n increases, the average completion time is expected to increase as the server has a constant serving capacity.

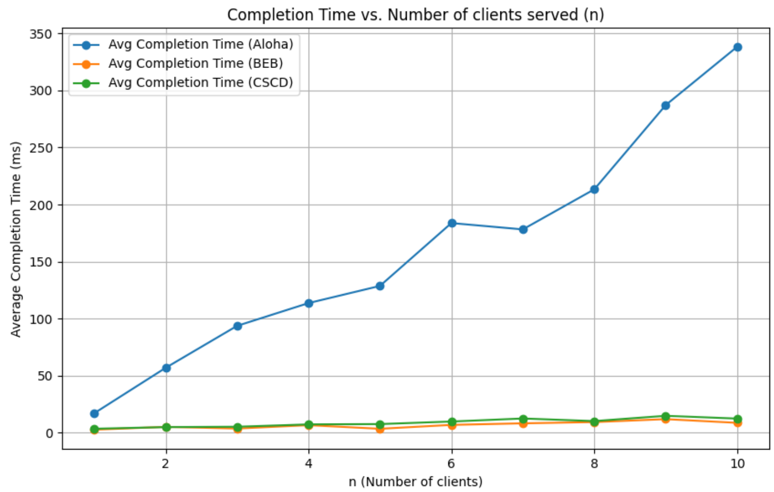


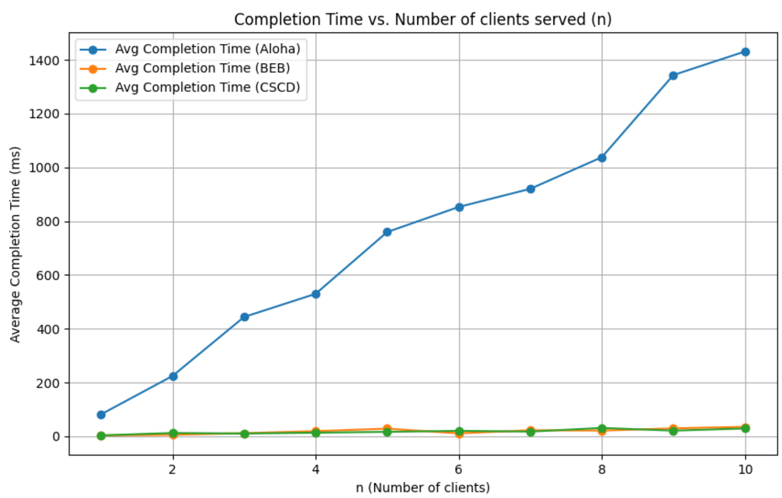
b. Binary Exponential Backoff – By increasing the weight time by at least a factor of 2, each time a collision occurs, BEB greatly reduces the probability of repeated collisions as compared to Slotted Aloha. This adaptive approach helps maintain a relatively stable completion time even as the number of clients increases.

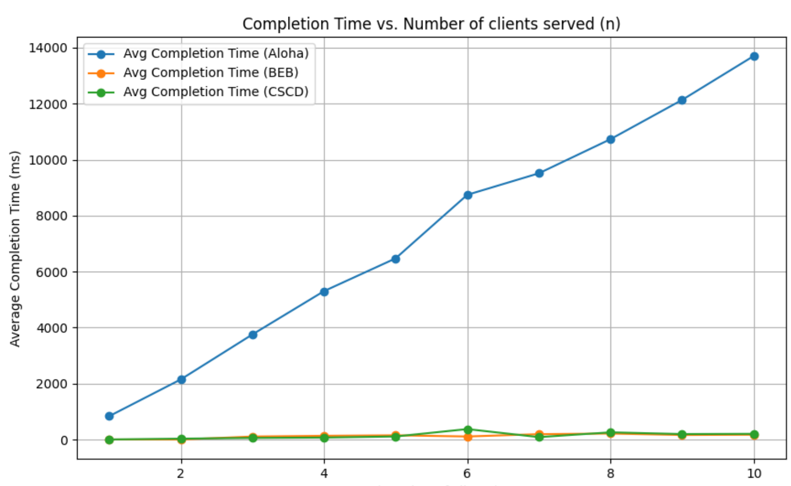
c. BEB with Carrier Sensing: CSCD combines BEB with the ability to detect if the channel is busy before transmitting. Clients only attempt to transmit when the channel is idle, further reducing the likelihood of collisions. This additional layer of collision avoidance helps maintain low and stable completion times.

→ As T increases, empirically we observe that average completion time increases in case of slotted Aloha while for BEB and BEB with CSMA, time required is nearly constant because repeated collisions were being avoid at the scale of T=1ms since service times are of the order of microseconds. Therefore, similar behavior is expected as value of T increases.

→ If *prob* increases, then chances of transmission within a slot will increase. Hence, probability of collision also increases leading to increase in mean completion time.

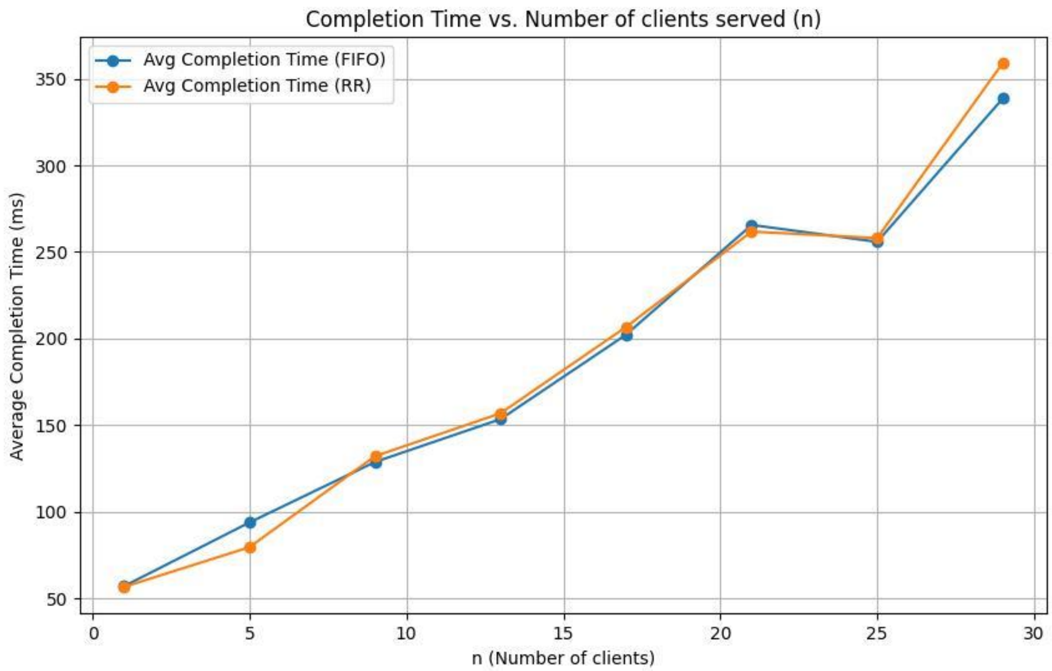

Completion Time vs. Number of clients served (n)
Avg Completion Time (Aloha)
Avg Completion Time (BEB)
Avg Completion Time (CSCD)
Average Completion Time (ms)
n (Number of clients)
0
50
100
150
200
250
300
350
2
4
6
8
10


Completion Time vs. Number of clients served (n)
Avg Completion Time (Aloha)
Avg Completion Time (BEB)
Avg Completion Time (CSCD)
Average Completion Time (ms)
n (Number of clients)
0
200
400
600
800
1000
1200
1400
2
4
6
8
10


Completion Time vs. Number of clients served (n)
Avg Completion Time (Aloha)
Avg Completion Time (BEB)
Avg Completion Time (CSCD)
Average Completion Time (ms)
n (Number of clients)
0
2000
4000
6000
8000
10000
12000
14000
2
4
6
8
10

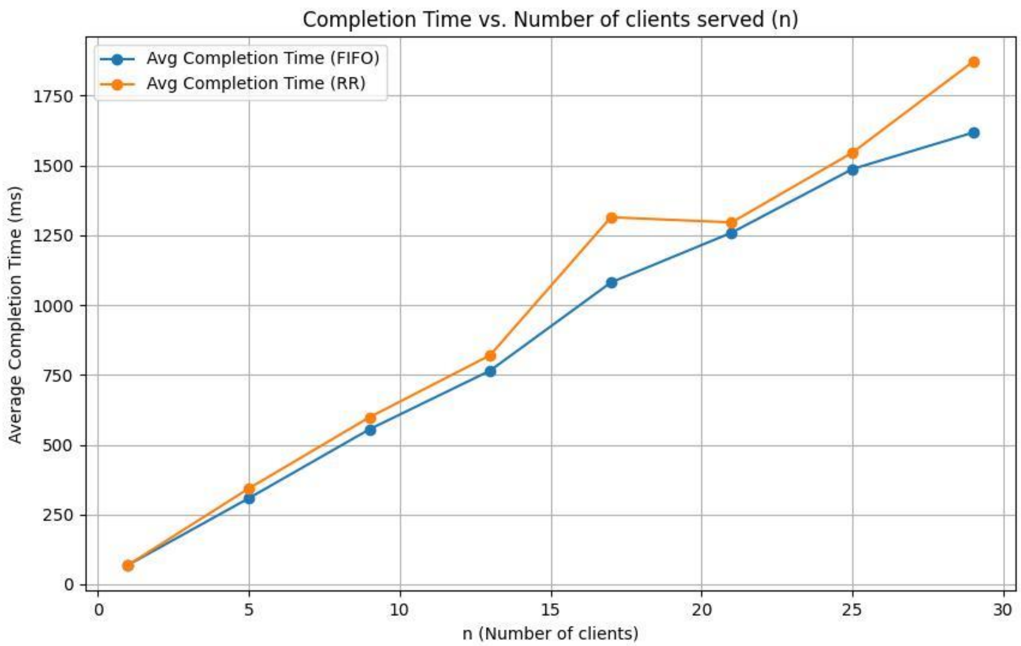
Part –4

a. Comparison of Round Robin and FIFO – Since the server can serve only one request at a time, the queuing policy does not affect the average completion time. Therefore, both Round Robin and FIFO have similar average completion times.



b. Compared to strategies in Part-3, we observe that queuing strategies offer worse average completion time since the overhead involved in waiting in queues is much larger as compared to random requests in which case service time is quite low as the incoming request is not potentially delayed due to head-of-line blocking.

c. In case of a rogue client, though Jain's fairness index is quite variable, two observations are clear. Firstly, Jain's index for Round Robin is always higher than that of FIFO, which is due to its inherently fair nature. Moreover, no for FIFO, this index can be quite close to zero especially when head-of-line blocking is particularly severe. Moreover, we observe that behavior of both queuing models is similar. The corresponding plot is given on the next page.


Completion Time vs. Number of clients served (n)
Avg Completion Time (FIFO)
Avg Completion Time (RR)
Average Completion Time (ms)
0
250
500
750
1000
1250
1500
1750
0
5
10
15
20
25
30
n (Number of clients)