Evaluation Experiment

n = Number of Nodes
d = Inter-request Delay
c = Critical Section (CS) Execution time

M = message complexity
R = response time
T = throughput
Number of requests per node = 500

Varying Number of Nodes (n):

n = 10; d = 20; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	16.3482	16.2538	16.4513	16.6551	16.6008	16.46184
R	688.3864	680.8876	745.3445	890.1075	926.7708	786.29936
T	13.227513227513228	13.35113484646195	12.345679012345679	10.504201680672269	10.080645161290322	11.9018348

n = 15; d = 20; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	26.517733333333332	26.560533333333332	26.5856	26.6484	26.620533333333334	26.58656
R	1375.7277333333334	1240.8622666666668	1225.8805333333332	1223.5345333333332	1248.4418666666666	1262.88939
T	10.359116022099448	11.52073732718894	11.627906976744185	11.700468018720748	11.450381679389313	11.331722

n = 20; d = 20; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	36.7184	36.4565	36.5719	36.7216	36.5731	36.6083
R	1600.7679	1537.8352	1493.4126	1559.3557	1488.9605	1536.06638
T	12.062726176115802	12.484394506866417	12.88659793814433	12.391573729863692	12.936610608020699	12.5523806

n = 25; d = 20; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	46.21416	46.05968	46.10936	46.03968	45.94912	46.0744
R	1666.11604	1573.386	1568.74112	1578.75504	1575.5024	1592.50012
T	12.493753123438282	15.262515262515263	15.299877600979192	15.188335358444714	15.206812652068127	14.6902588

n = 30; d = 20; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	56.620733333333334	57.0444	57.305733333333336	57.5518	57.31773333333334	57.16808
R	2467.6307333333334	2757.1617333333334	3337.9674	4930.528333333334	3528.1682	3404.29128
T	11.848341232227488	10.668563300142248	8.849557522123893	6.026516673362797	8.379888268156424	9.154573399

Varying Inter-Request Delay (d):

n = 10; d = 20; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	16.3482	16.2538	16.4513	16.6551	16.6008	16.46184
R	688.3864	680.8876	745.3445	890.1075	926.7708	786.29936
T	13.227513227513228	13.35113484646195	12.345679012345679	10.504201680672269	10.080645161290322	11.9018348

n = 10; d = 25; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	15.9272	16.0558	15.9434	16.0958	16.0106	16.00656
R	591.3272	611.3054	593.4764	611.7132	613.5514	604.27472
T	15.015015015015015	14.619883040935672	15.060240963855422	14.662756598240469	14.534883720930232	14.7785559

n = 10; d = 30; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	16.1002	15.9964	16.1086	16.0172	16.1692	16.07832
R	605.4436	592.2104	597.899	605.7252	589.8908	598.2338
T	14.577259475218659	14.836795252225519	14.749262536873156	14.534883720930232	14.880952380952381	14.7158307

n = 10; d = 35; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	16.0794	16.2522	16.3486	16.0428	16.3572	16.21604
R	595.22	652.4018	657.6038	644.5798	663.5182	642.66472
T	14.619883040935672	13.513513513513514	13.477088948787062	13.58695652173913	13.477088948787062	13.73490619

n = 10; d = 40; c = 10

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	16.5272	16.535	16.6414	16.523	16.336	16.51252

R	764.8376	846.1236	782.82	723.6604	704.335	764.35532
T	11.547344110854503	10.526315789473685	11.363636363636363	12.165450121654501	12.376237623762377	11.5957968

<u>Varying Critical Section Execution Time (c):</u>

n = 10; d = 20; c = 10

16.461847 86.29936 11.9018348

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	16.3482	16.2538	16.4513	16.6551	16.6008	16.46184
R	688.3864	680.8876	745.3445	890.1075	926.7708	786.29936
T	13.227513227513228	13.35113484646195	12.345679012345679	10.504201680672269	10.080645161290322	11.9018348

n = 10; d = 20; c = 15

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	15.564666666666668	15.55888888888889	15.547333333333333	15.543555555555555	15.504444444444445	15.5437778
R	1256.7095555555557	1196.3194444444443	1141.928	1126.9602222222222	1283.0593333333334	1200.99531
T	6.901840490797546	6.168608636052091	7.563025210084033	7.666098807495741	6.756756756756757	7.01126598

n = 10; d = 20; c = 20

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	15.566	15.575555555555555	15.620222222222223	15.536222222222221	15.63711111111111	15.5870222
R	1337.2575555555557	1200.1248888888888	1189.7553333333333	1230.3433333333332	1262.5877777777778	1244.01378
T	6.502890173410405	7.223113964686998	7.281553398058253	7.03125	6.870229007633588	6.9818073

n = 10; d = 20; c = 25

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	15.616444444444445	15.651555555555555	15.625333333333334	15.604	15.584666666666667	15.6164
R	1259.2591111111112	1259.227111111111	1352.111777777778	1295.0415555555555	1289.1322222222223	1290.95436
T	6.891271056661562	6.901840490797546	6.428571428571429	6.696428571428571	6.736526946107785	6.7309277

n = 10; d = 20; c = 30

	Test 1	Test 2	Test 3	Test 4	Test 5	AVG
M	15.610222222222223	15.618666666666666	14.851555555555555	14.981111111111112	15.05711111111111	15.2237333
R	1314.0166666666667	1315.5542222222223	1316.1813333333333	1139.59	1117.6302222222223	1240.59449
T	6.607929515418502	6.5982404692082115	6.202453987730062	7.487520798668885	7.627118644067797	6.90465268

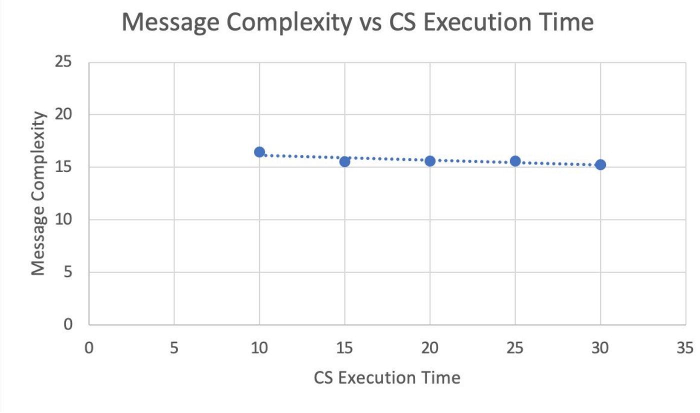
Graphs

Varying Critical Section Execution Time (c):

Message Complexity

Constants: n = 10; d = 20; requests # = 500

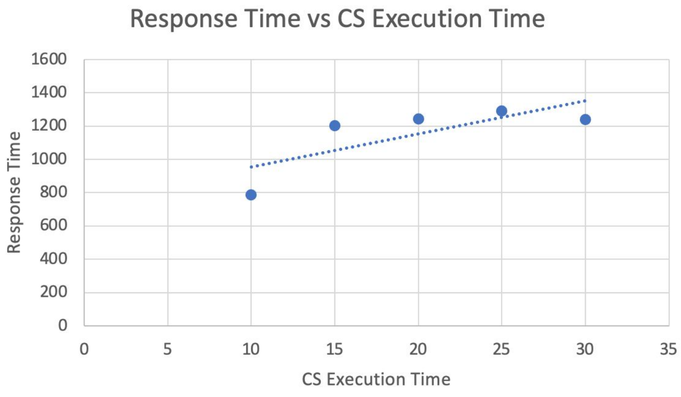
CS Execution time	10	15	20	25	30
Message complexity	16.46184	15.5437778	15.5870222	15.6164	15.2237333



Response Time

Constants: n = 10; d = 20; requests # = 500

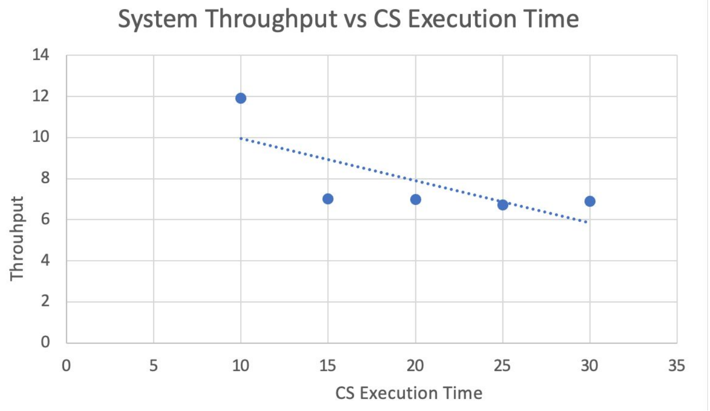
CS Execution time	10	15	20	25	30
Response Time	786.29936	1200.99531	1244.01378	1290.95436	1240.59449



System Throughput

Constants: n = 10; d = 20; requests # = 500

CS Execution time	10	15	20	25	30
Throughput	11.9018348	7.01126598	6.9818073	6.7309277	6.90465268

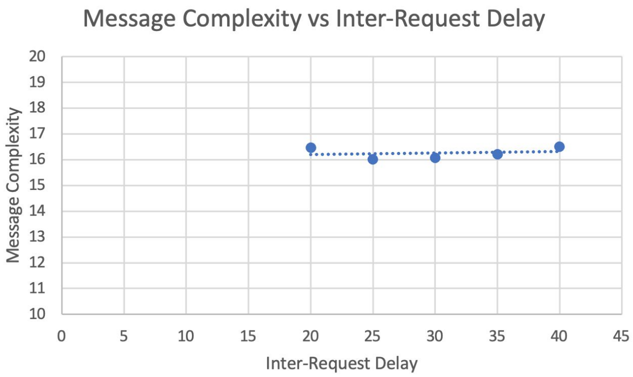


Varying Inter-Request Delay (d):

Message Complexity

Constants: n = 10; c = 10; requests # = 500

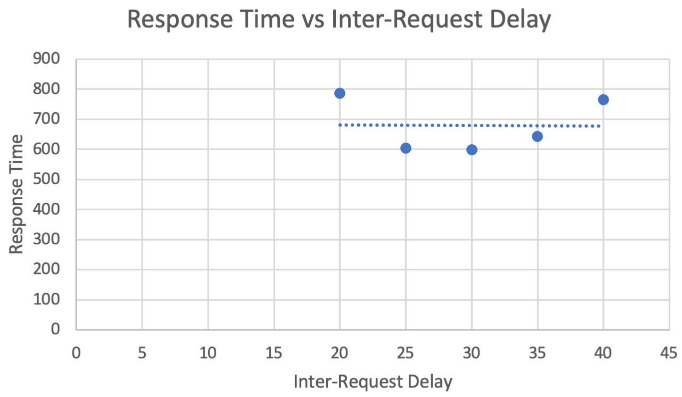
Inter-request Delay	20	25	30	35	40
Message complexity	16.46184	16.00656	16.07832	16.21604	16.51252



Response Time

Constants: n = 10; c = 10; requests # = 500

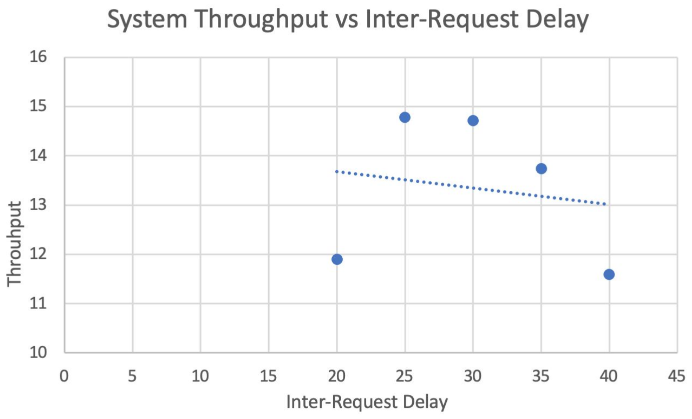
Inter-request Delay	20	25	30	35	40
Response Time	786.29936	604.27472	598.2338	642.66472	764.35532



System Throughput

Constants: n = 10; c = 10; requests # = 500

Inter-request Delay	20	25	30	35	40
Throughput	11.9018348	14.7785559	14.7158307	13.73490619	11.5957968

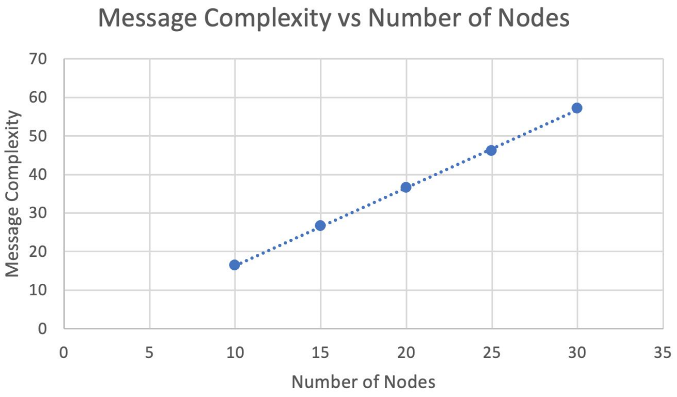


Varying Number of Nodes (n):

Message Complexity

Constants: d = 20; c = 10; requests # = 500

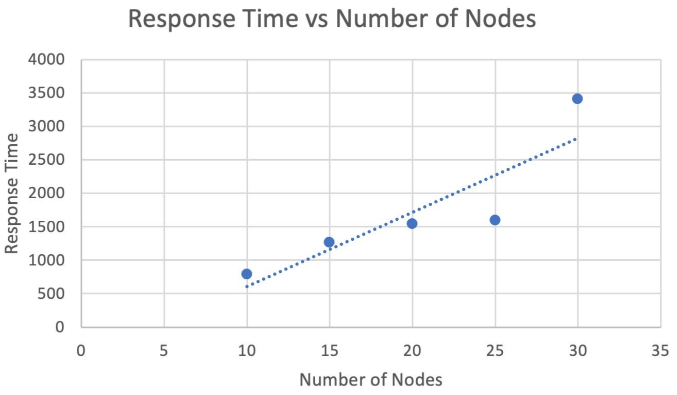
Inter-request Delay	10	15	20	25	30
Message complexity	16.46184	26.58656	36.6083	46.0744	57.16808



Response Time

Constants: d = 20; c = 10; requests # = 500

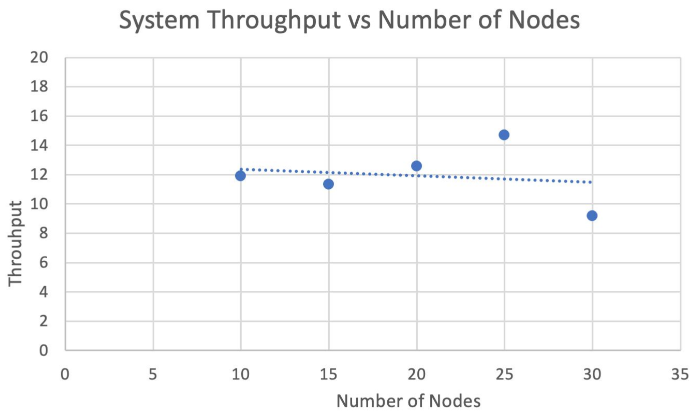
Inter-request Delay	10	15	20	25	30
Response Time	786.29936	1262.88939	1536.06638	1592.50012	3404.29128



System Throughput

Constants: d = 20; c = 10; requests # = 500

Inter-request Delay	10	15	20	25	30
Throughput	11.9018348	11.331722	12.5523806	14.6902588	9.154573399



Analysis:

The results of the tests as well as the corresponding graphs and their trends demonstrate the effect of each type of parameter on the performance metrics of message complexity, response time, and system throughput. Message complexity is the ratio of the total number of messages sent to the number of requests made. Response time is the time spent in csEnter() collecting all the keys and the time spent executing its critical section. System throughput is the ratio of the number of requests to the total time spent from requesting to enter its first critical section to handling all pending requests after leaving its last critical section. In this analysis, the expected results are compared to the experimental results shown by the trend lines on the graphs.

Varying Critical Section Execution Time (c):

By increasing c, more time is spent in each process's critical section. Increasing c causes the system throughput to decrease because the execution of all critical sections for each process will increase which means less requests are processed per unit time. It also causes the response time to increase because response time includes the time spent in csEnter() and the time spent executing the critical section (c). Therefore, if c is increased then response time must also be increased. Finally, increasing c causes the message complexity to remain about the same because this value is dependent on the number of processes in the system. It is likely that if c continued to increase by a larger and larger amount that the message complexity would begin to increase because the more time that processes spend in their critical sections, the more likely it is that a process's requests will be preempted which will cause the process to send a release message and then a request message.

Varying Inter-Request Delay (d):

By increasing d, each process spends more time in the state between critical section execution and requesting to enter its critical section. Increasing d causes the system throughput to decrease because the time spent in between requesting and entering the critical section increases which means less requests are processed per unit time. It also causes the response time to slightly decrease because it is more likely that requests will be satisfied in csEnter() more quickly since each process will be spending more time in between execution and requesting. Finally, increasing d causes the message complexity to remain about the same because this value is dependent on the number of processes in the system. It is likely that if d continued to increase by a larger and larger amount that the message complexity would also begin to decrease because the more time that processes spend not requesting or executing their critical section, the less likely it is that a process's requests will be preempted.

Varying Number of Processes (n):

By increasing n, the total number of processes in the system increases. Increasing n causes the system throughput to decrease because more processes means that each process will wait a longer amount of time to receive all its keys and execute its critical section so less requests are processed per unit time. It also causes the response time to increase because csEnter() will take a longer amount of time since it will have to wait for more processes to finish executing their critical sections. Finally, increasing n causes the message complexity to increase because message complexity is dependent on the number of processes in the system as this determines how many messages are sent between each critical section.