Advanced Operating Systems
Project 02 Report

Aim
The objective of the project is to check the performance of Lamport's mutual exclusion protocol with respect to Response Time and System Throughput.

Methodology
Lamport's mutual exclusion protocol is implemented for a sample distributed application. The protocol runs for a configuration of the system parameters namely, number of processes (n), the mean delay between two subsequent requests by a process (d), the mean execution time of a critical section (c) and the max number of critical section requests made by a single process (r).

Implementation
Lamport's mutual exclusion protocol are implemented in C++ using TCP socket programming in a single project. All mutual exclusion related communication is handled by the service module. Every process has (n+1) threads; 1 for the Application module and n Receiver threads (1 for each node). The application module interacts with the service module only through csEnter() and csLeave() function calls.

Configuration File
The configuration file is the input to the program and is a plain-text formatted file. Lines which are not valid are ignored. The first valid line of the configuration file contains four tokens. The first token is the number of nodes in the system. The second token is the mean value for d (in milliseconds). The third token is the mean value for c (in milliseconds). The fourth token is the number of requests each node should generate. After the first valid line, the next n lines consist of three tokens. The first token is the node ID. The second token is the hostname of the machine on which the node runs. The third token is the port on which the node listens for incoming connections.

Testing
The mechanism that was used to test the correctness of the protocols was using a central shared file where the processes write logs when they enter and exit a critical section. This way no two enters or no two exits should be adjacent.

Execution Setup
A script (launcher.sh) is created to run the application instance in the machines as mentioned in configuration file. The script passes the necessary arguments for each instance to run successfully in respective machines. Then the cleanup.sh script is run to gather all output data for this run.

Experimental Setup

Following are the set up information used in our experimentation

Operating System:	Linux
Processor:	64-bit
RAM:	16 GB
Programming Language:	C++
Socket Protocol:	TCP

The experiments were run multiple times for various values of n, d and c.

Observations

For all observations, we considered the no of nodes to be 8 where each node generates 1000 Critical Section requests.
We plot 4 graphs

1. Mean Response Time Vs Inter Request Delay
2. Mean System Throughput Vs Inter Request Delay
3. Mean Response Time Vs Mean Critical Section Execution Time
4. Mean System Throughput Vs Mean Critical Section Execution Time

While plotting graphs for Mean Response Time/ Mean System Throughput Vs Inter Request Delay, the value of Inter Request Delay was 0,2,4,6,8,10 ms and critical section execution time was 5 ms.

While plotting graphs for Mean Response Time/ Mean System Throughput Vs mean Critical Section execution time, the values of Critical Section Execution were 0,2,4,6,8,10 ms and Inter Request Delay was 5 ms.

For each value Inter Request Delay/ Critical Section Execution values, multiple iteration was performed and the mean value of iterations was taken to plot the final graph.

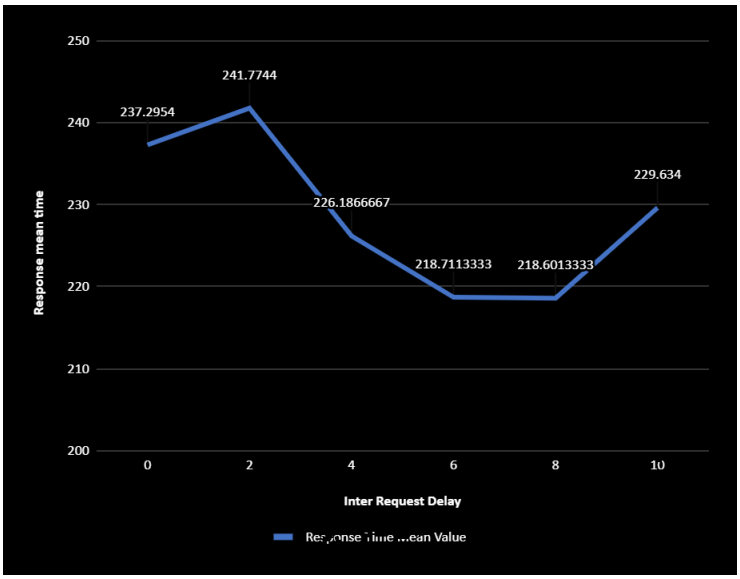
Response Time Mean Value	Inter Request Delay
237.2954	0
241.7744	2
226.1866667	4
218.7113333	6
218.6013333	8
229.634	10

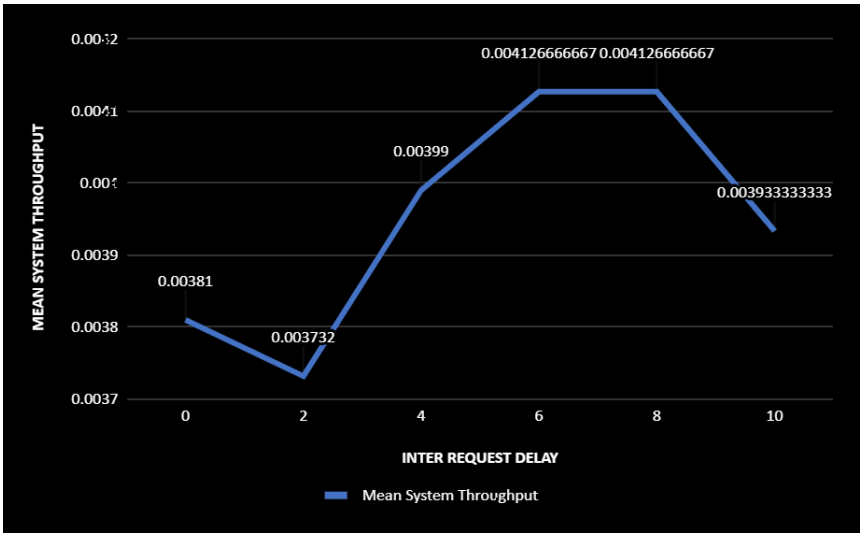
Mean System Throughput	Inter Request Delay
0.00381	0
0.003732	2

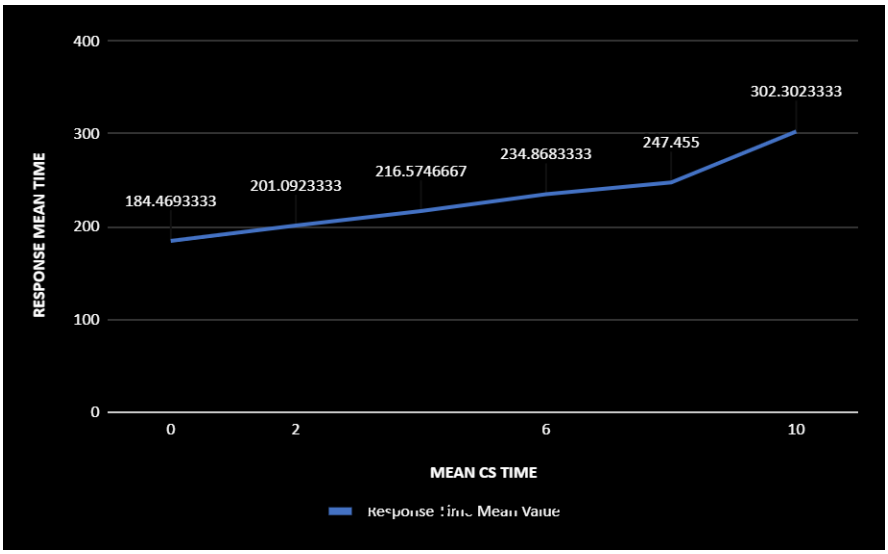
0.00399	4
0.004126666667	6
0.004126666667	8
0.003933333333	10

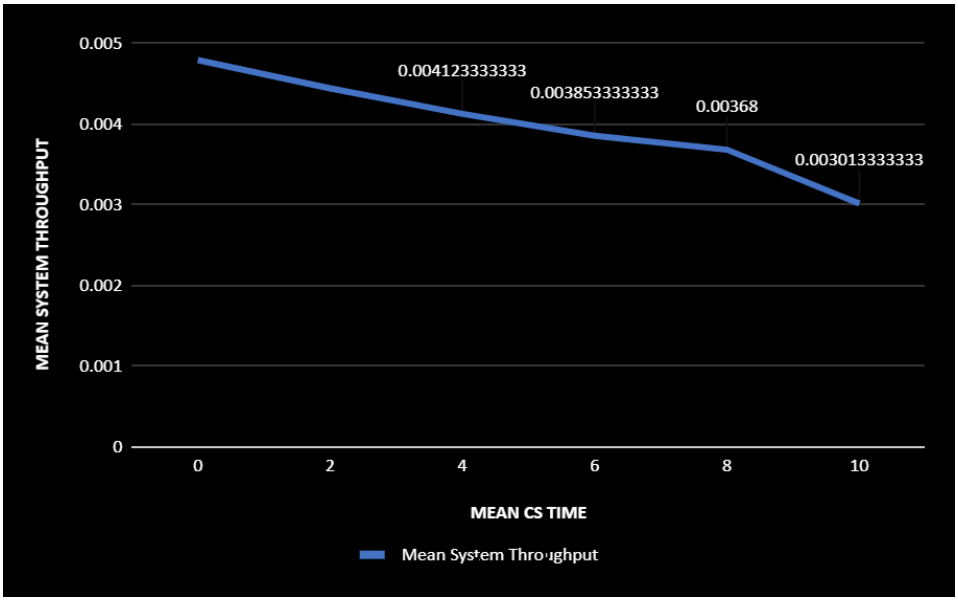
Response Time Mean Value	Mean CS time
184.4693333	0
201.0923333	2
216.5746667	4
234.8683333	6
247.455	8
302.3023333	10

Mean System Throughput	Mean CS time
0.00479	0
0.00444	2
0.004123333333	4
0.003853333333	6
0.00368	8
0.003013333333	10




MEAN SYSTEM THROUGHPUT
0.0042
0.0041
0.004
0.0039
0.0038
0.0037
0.00381
0.003732
0.00399
0.004126666667
0.004126666667
0.003933333333
0
2
4
6
8
10
INTER REQUEST DELAY
Mean System Throughput


RESPONSE MEAN TIME
400
300
200
100
0
184.4693333
201.0923333
216.5746667
234.8683333
247.455
302.3023333
0
2
6
10
MEAN CS TIME
Response Time Mean Value



Conclusion

- Having look at the graphs, The graphs Mean Response Time vs Inter Request Delay and Mean System Throughput vs Inter Request Delay are an inverse of each other
- The Response Day increases as the value of Critical Section Execution time increases
- The System Throughput decreases as the value of Critical Section Execution time increases