Comparison Study
(Raymonds vs Suzuki-Kasami)

INDEX

T{I}:   time in milliseconds

[T1, T2] indicates the range of time for which a process waits before requesting for a critical section. This is to simulate the execution (other than the critical section) in a process.

T3 is the time that a process spends in a critical section.

Data collection and measurement

We ran both algorithms for two different sets of values of T1, T2 and T3 with varying number of nodes in the distributed system. The data was collected at each node. This data includes the total number of messages sent and received, total synchronization delay, total wait time, and start and finish times for each critical section.

The coordinator process requests for this data after it has received complete messages from all the processes. Processes respond by sending the collected data to the coordinator.

In the coordinator process, we accumulate all the data from each node and calculate the average number of messages, synchronization delay and wait time corresponding to the number of nodes in the system.
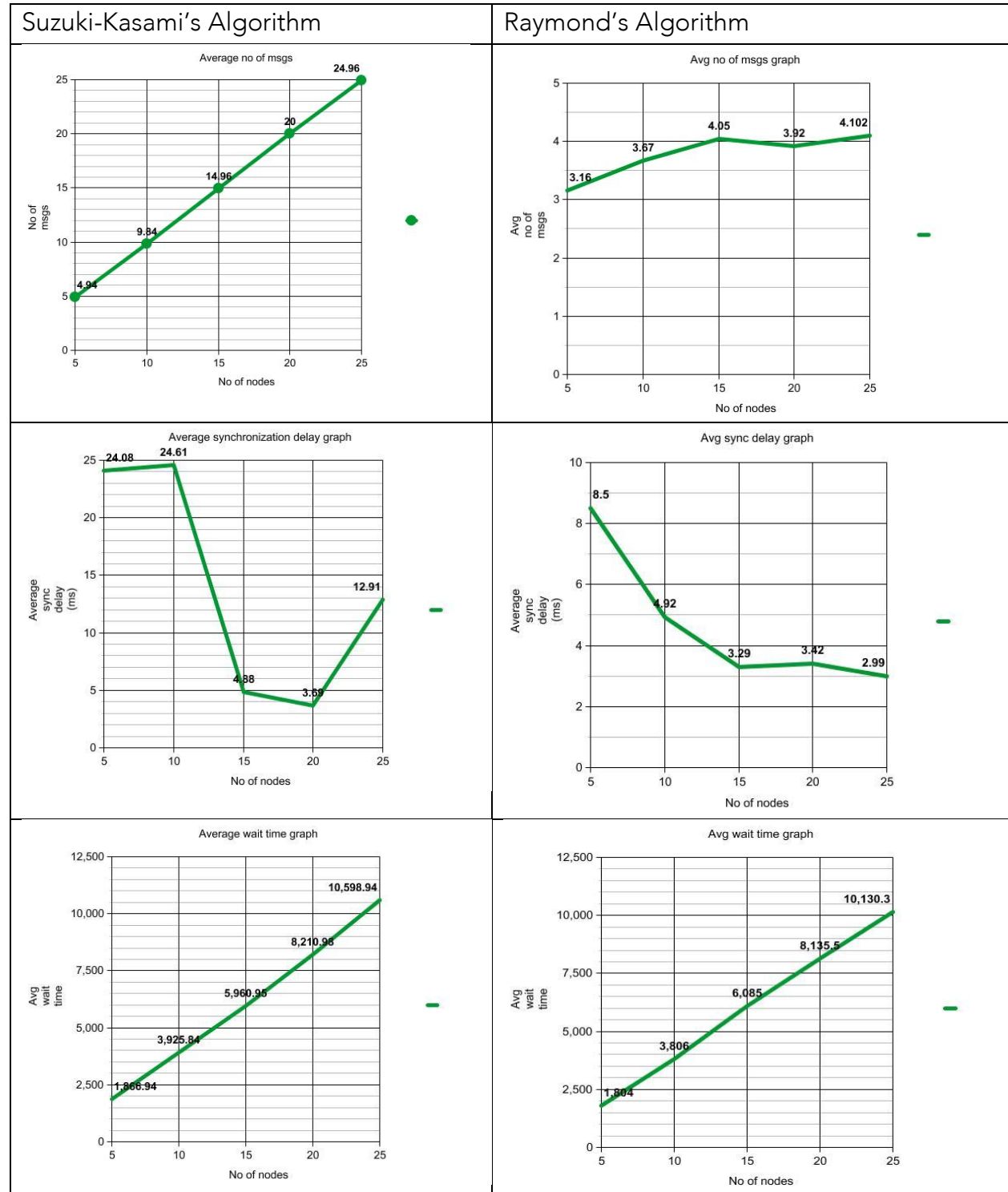
RESULT GRAPHS

The two algorithms have been compared in terms of number of messages sent and received, total synchronization delay, the total wait time, and the start and finish times for each critical section. The respective graphs for each set have been attached.

1st configuration:

T1 = 50

T2 = 200

T3 = 500

Suzuki-Kasami's Algorithm	Raymond's Algorithm
Average no of msgs	Avg no of msgs graph
Average synchronization delay graph	Avg sync delay graph
Average wait time graph	Avg wait time graph

2nd configuration:

T1 = 500

T2 = 2000

T3 = 50

Suzuki-Kasami's Algorithm	Raymond's Algorithm