CS5320 – Assignment #3
Implementing Distributed Mutual Exclusion Algorithm

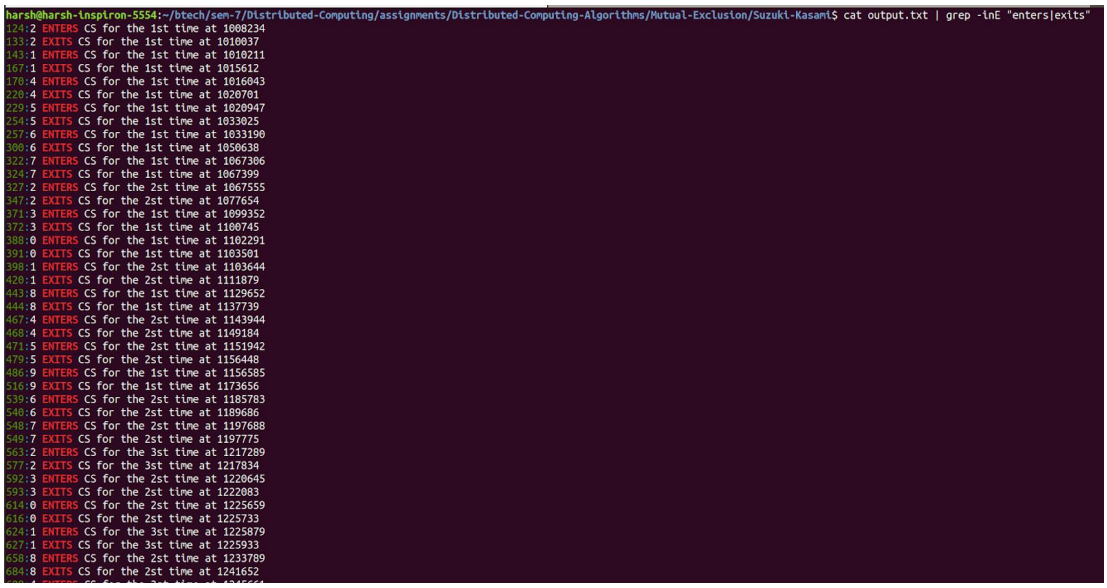
By Harsh Agarwal
cs15btech11019

Implementation Details

Each node is represented by a thread.
Every node creates another thread for receiving messages at a port number.
All nodes are running on different port numbers on localhost.
The port number for every node identified using *(PORT_STARTING_RANGE + nodeID)*.

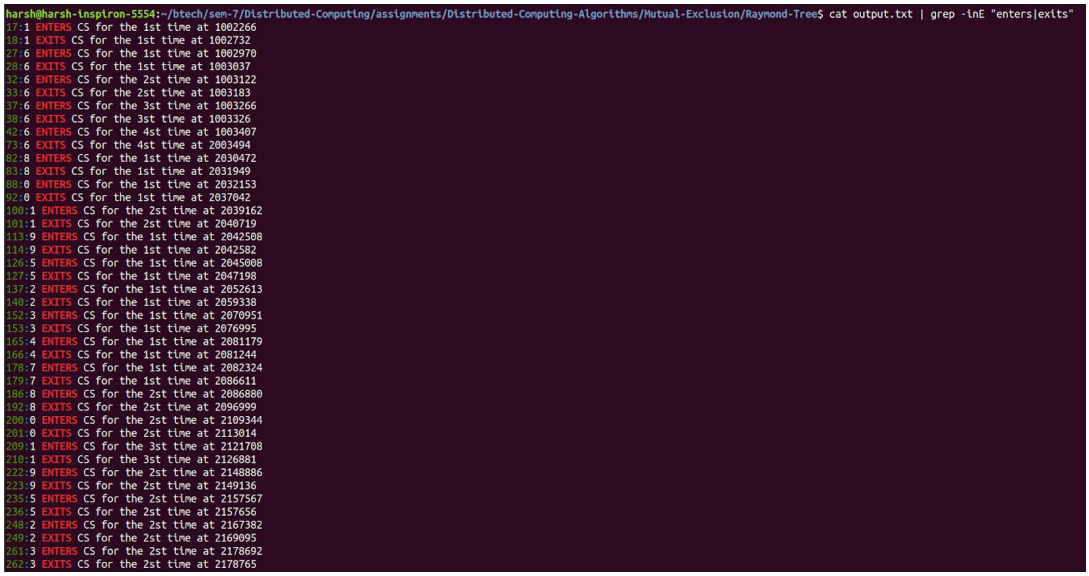
Mutual Exclusion Property

Suzuki Kasami Algorithm



All ENTER events are followed by an EXIT event before the next ENTER.
Thus mutual exclusion is satisfied.

Kerry Raymond Algorithm



All ENTER events are followed by an EXIT event before the next ENTER. Thus mutual exclusion is satisfied.

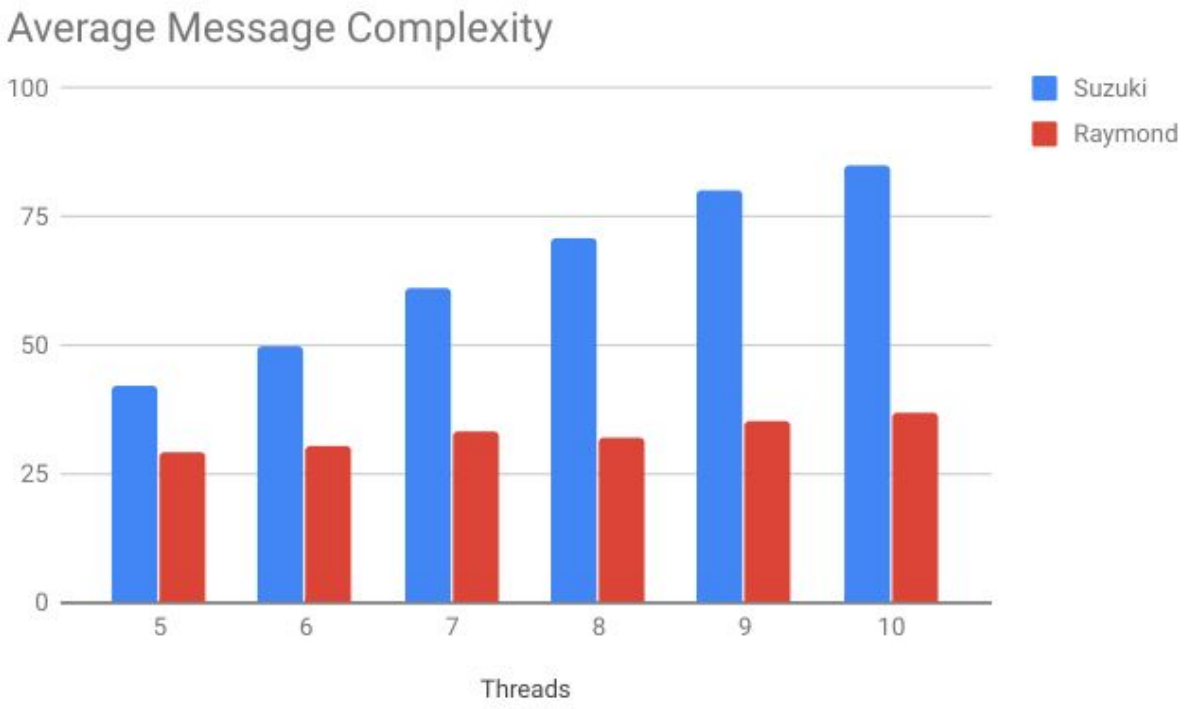
Graphs Section

Topology - Complete Graph
Number of CS executions - 8
Intial Node With Token - 0
Alpha - 2
Beta - 1

Average Message Complexity

READINGS

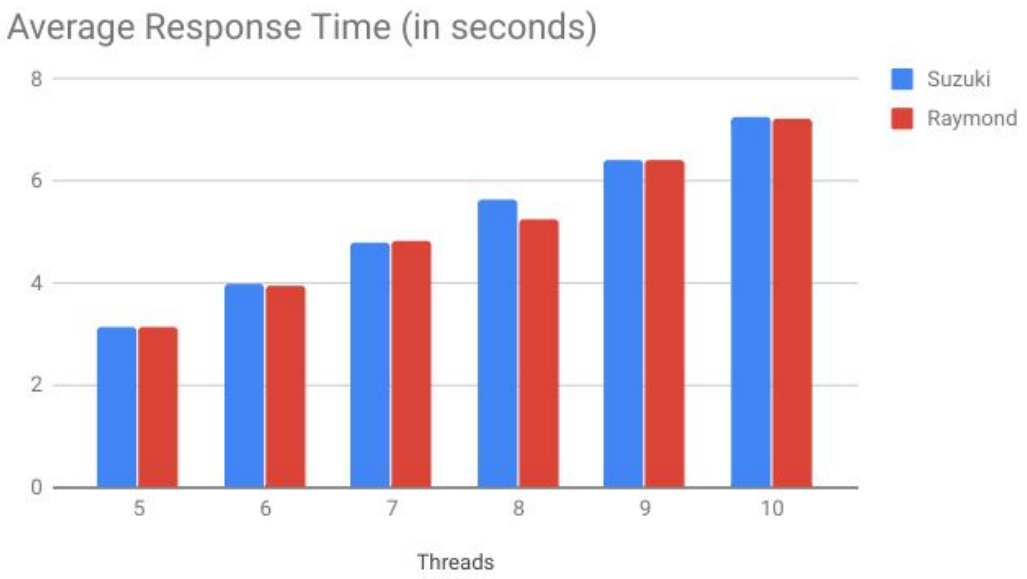
Threads	Suzuki	Raymond
5	42	29.2
6	50	30.6667
7	61	33.1429
8	71	32.25
9	80	35.3333
10	85	36.8



Average Response Time (in seconds)

READINGS

Threads	Suzuki	Raymond
5	3.143	3.152
6	3.972	3.964
7	4.797	4.825
8	5.639	5.249
9	6.422	6.410
10	7.275	7.226



OBSERVATIONS

- Raymond Algorithm has much lesser message complexity than Suzuki because Raymond sends REQUEST & PRIVILEGE messages just to its neighbors in a spanning tree.
  The added message complexity of forwarding PRIVILEGE message along spanning tree edges in Raymond is superseded by the benefit of sending REQUEST just to HOLDER node.(There are much more REQUEST messages in the system than PRIVILEGE messages.)

  But Suzuki Algorithm sends REQUEST messages to all other nodes. Due to this, a lot of messages need to be sent.

- On dividing Raymond message complexity with the number of iterations, we get the average message complexity per CS execution to be almost equal to 4.
  This matches with the fact explained in book
  “as the number of nodes requesting the privilege increases, the number of messages exchanged per critical section entry decreases. In fact, it requires the exchange of only four messages per CS execution.”
  That’s why the message complexity increases very slowly with the

number of nodes.

- Suzuki algorithm message complexity increases at a faster rate with the number of nodes. This is because more REQUEST messages need to be sent along new N-1 edges.

- Response times are almost the same for both the algorithms.
  It increases with time.
  This is because:
  For Raymond, more edges need to be traversed to pass on the token from a privileged node to a requesting node.
  For Suzuki, more REQUEST messages need to be sent along new N-1 edges.