CS5320

ASSIGNMENT-2
TERMINATION DETECTION

SHAMIK KUNDU (CS16MTECH11015)

Introduction

A distributed system of several processes is said to be terminated if all the processes are idle and there are no messages in the channel. In this assignment, two termination detection algorithm is implemented with the assumption that the underlying graph is not complete, rather it's a spanning tree. Also, initially there can be multiple active processes.

Few salient features of the implementation:

1. Use of socket as an interprocess communication.
2. Use of threads and mutex to properly mimic process behaviour and maintain synchronization

Few modifications have been made in the original algorithm to make it useful for non-complete graph also. And to make sure, there is a definite termination, it is assumed that a process won't become active once it has sent all of it's messages.

The modifications to the Huang's algorithms are:

1. The condition for termination is that, maximum weight of the initiator(root) in the end should be equal to the number of initially active processes. Also the initiator should be passive.
2. If a passive process receives an application message, it becomes active only if already hasn't sent all of it's application messages. Otherwise, it will only add the weights of the application message but won't become active.
3. If a passive process receives a control message, it directly sends it back to it's parent and do not add this weight to it's own weight.

Explanations:

Both the algorithms were run on increasing number of processes (1 to 4) and with different topology. The average number of control messages sent in each case were plotted in the following graph.

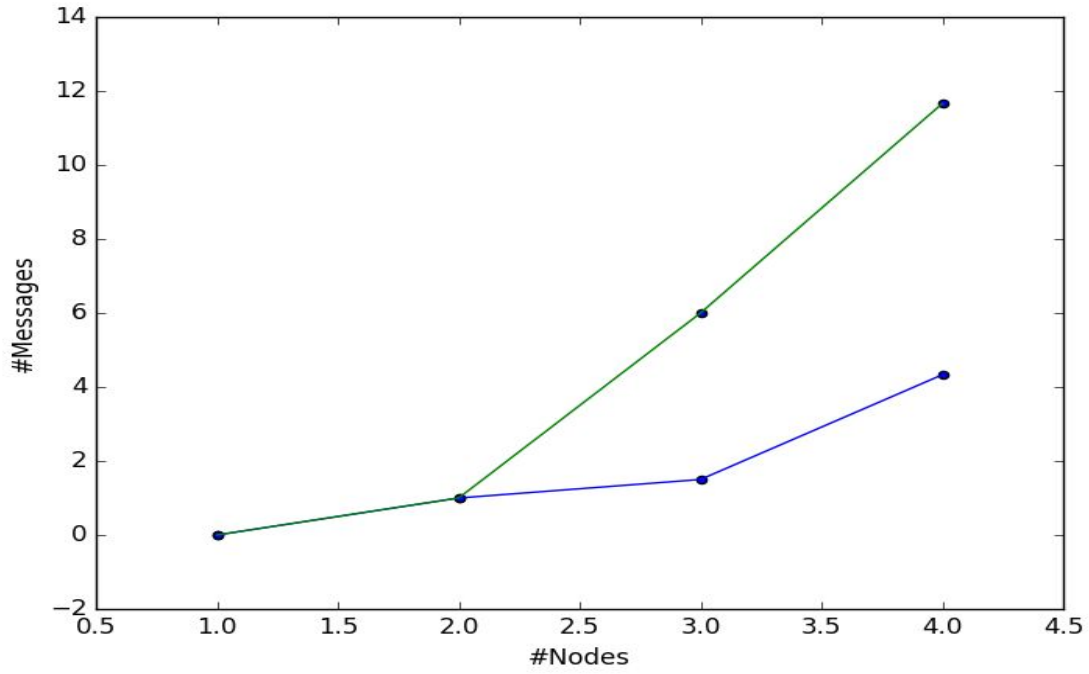
Huang's algorithm:

#Node	1	2	3	4
#Messages	0	1	1.5	4.33

Spanning tree based algorithm:

#Node	1	2	3	4
#Messages	0	1	6	11.66

The blue graph represents huang's algorithm and the green graph represents spanning tree based algorithm.



It is clearly evident that Huang's algorithm needs less number of control messages than Spanning tree based algorithm. The main reason behind is that, in Huang's algorithm, weights are piggybacked with application messages. Only when a process becomes permanently passive, it send a control message to the initiator. On the other hand, in spanning tree based algorithm, there are two kinds of control message and in each time, both the messages traverse the entire height of the tree. It leads to more number of control messages.