

Dijkstra-Scholten and Shavit-Francez termination algorithms

Swapna Kalidindi
Department of Computer Science
Kent State University
Kent, OH 44242
skalidin@kent.edu

ABSTRACT

The Dijkstra-Scholten algorithm detects the termination of a centralized basic computation. This algorithm was generalized to decentralized basic computations by Shavit and Francez. Where the computation graph is a forest, of which each tree is rooted at an initiator of the basic computation. The aim of this project is to study how the particular topology impacts the overall message complexity of the system.

1. INTRODUCTION

A computation of a distributed algorithm terminates when the algorithm has reached the terminal configuration that is when no further steps of the algorithm are applicable and no messages in transit. Such termination detection is significant as no process will have knowledge about the global state of the computation.

The approach to this problem is by designing an algorithm that can consider message termination after which the algorithm is made process-termination using one of the methods which contain two other algorithms that interact with the given message terminating algorithm and with each other. One of these algorithms detect that the computation has reached the terminal configuration and then it calls the second algorithm that floods terminate message to all processes making them to

enter critical state. Based on the above approach Dijkstra and Scholten has proposed an algorithm to detect the global termination of a centralized basic computation. In the detection algorithm, the initiator of the basic computation plays an important role. In this approach detection algorithm maintains a computation tree, initiator being the root. Tree is expanded when a basic message is sent or when a process not in the tree becomes active. The basic message is deleted after it is received and after termination to ensure the progress of the detection algorithm the tree should collapse after finite number of states both the cases involves removal of nodes from tree. Leaves of the tree are messages and every process keeps the count of their sons.

For some process P the deletion of its son might occur in a different process Q , either due to the receipt of a message or by the deletion of son of process Q . A control message is sent to P to inform it, about the

deletion of its son and process P decrements to its son count on the receipt of control message. When the tree seizes such that it contains only an initiator, at this point the computation algorithm has detected terminal configuration and then the initiator calls the another routine, Announce that explicitly terminates all the processes.

Although Dijkstra-Scholten is a correct termination –detection algorithm, it was limited to diffusing computation. In 1986, N. Shavit and N.Francez generalized Dijkstra-Scholten to decentralized computation. In the Shavit – Francez algorithm, computation tree is a forest where each tree is rooted at the initiator of the basic computation. Unlike the Dijkstra-Scholten algorithm, each initiator in the forest observes only the emptiness of its own tree which does not prevent the initiator to become active again. Initiator becomes active again when it is inserted in the tree of another initiator. Verification that all trees are collapsed is done by running wave algorithm. A process participates in wave only if the tree was collapsed and when the wave algorithm decides, Announce is called. In the Shavit – Francez algorithm for every basic message sent, a control message then wave algorithm is run which calls Announce routine. The message complexity of this algorithm is $M+W$ control messages. For termination detection of decentralized computation, this is a worst-case optimal algorithm.

2. EXPERIMENTAL SETUP

To measure the message complexity of different topologies, an experimental setup was made by the simulation on different number of initiators by varying topologies. The experiments were carried by varying the initiators on different number of nodes ranging from 5 to 50 in multiples of 5. The

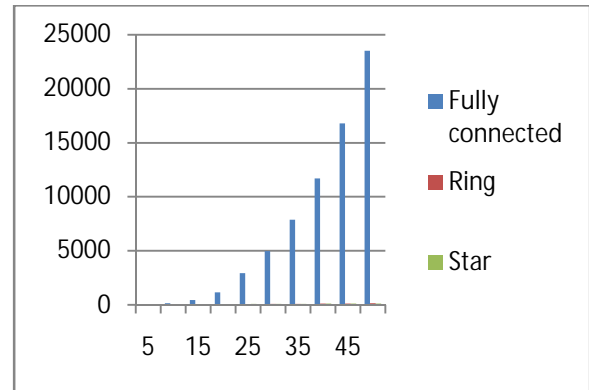
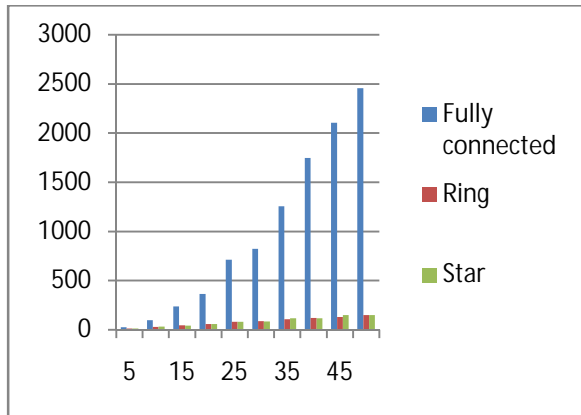
number of messages generated were recorded for each of the five runs and average values were taken. . Echo wave algorithm is used on the top of termination detection algorithm.

Simulation engine uses three random number generators, and there purpose are: for each node to select the number of neighbors to send message, the neighbors to which message need to be sent and to decide whether a node participates in communication or not. Data results were collected on Fully connected, Ring and Star topologies. The simulation machine begins by the exchange of basic messages among processes and it halts when announce is called by the echo algorithm i.e.; when it receives reply from all the initiators

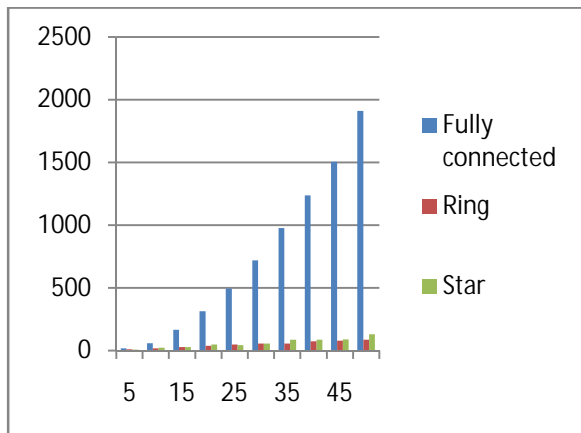
3. RESULTS

The experiments were carried out on three different types of topologies. They are fully connected, star and ring by considering the cases when there is single initiator, half of the nodes being initiators and all the nodes acting as initiators on Fully connected, Star and Ring topologies by varying the number of nodes on a scale of 5. The graphs below show the number of messages generated according to the chosen topology.

The below graph depicts the number of messages generated in case of fully connected, star and ring topologies when one initiator is considered.



The below graph depicts the number of messages generated in case of fully connected, star and ring topologies when half of the processes act as initiators.



The below graph depicts the number of messages generated in case of fully connected, star and ring topologies when all the processes act as initiators.

4. CONCLUSION AND FUTURE WORK

From the data analyzed, results obtained were the function of the chosen parameters. It can be concluded that overall message complexity of the algorithm varies according to the topology and number of initiators. In my class presentation there was deviation in the graph when 45 nodes participated in communication, in a fully connected topology has I have mistakenly taken the value of other simulation which I corrected in this report. Future work will include observing the performance of the algorithm for different wave algorithms and also when random nodes participate in computation.

5. REFERENCES

- [1] Edsger W. Dijkstra, C. S. Scholten "Termination Detection for Diffusing Computations" 1980.
- [2] Gerard Tel "Introduction to Distributed Algorithms", Cambridge University Press, 1st edition, 1994.
- [3] Nir Shavit, Nissim Francez "A new approach to detection of locally indicative stability" 1986.