EDA387 / DIT660 Computer Networks

Lab 2.4: Self-stabilizing algorithm for finding the centers with the tree topology

Project Group 29

Name: Kashif Shabir and Osita Christopher Nwanu

Date: October 08, 2020

Solution 1

$Max(N_h^-(i))$ is the missing expression in line 12 of Algorithm 1.

Solution 2

We defined a set of legal executions for Algorithm 1 using lines 11 and 12 of the code. In line 11, a processor P_i is a leaf node in the tree when the condition $h_i = 0$ holds, while in line 12, P_i is not a leaf node in the tree when the condition $h_i = 1 + max(N_h^-(i))$ holds.

Solution 3

Assumptions:

I. C and C_{safe} represent a configuration and safe configuration in Algorithm 1, respectively.
II. We will use floating distance [h-values] to represent the height of the nodes towards the center.
III. In C_{safe}, two conditions apply: first, the smallest floating distance $h_i = 0$ holds for all leaf nodes in the tree [line 11 of the code]. Second, the maximum floating distance $h_i = 1 + max(N_h^-(i))$ holds for all non-leaf nodes in the tree as per line 12 of the code.
IV. Convergence happens within k + 1 asynchronous rounds.

Proof of Convergence:

In line 11 of the code, when i = 0, then $h_0 = 0$ within the first asynchronous round. This proves that the smallest distance for all leaf nodes in the tree is at least 0 in C_0.

We then prove that the maximum floating distance $h_i = 1 + max(N_h^-(i))$ holds for all non-leaf nodes in the tree.

Since P_i can read the h-values or floating distance of all their neighbors, it follows that within k+1 asynchronous rounds, Algorithm 1 will reach a safe configuration C_{safe}, wherein all the leaf nodes were eliminated, remaining only the non-leaf node with minimum eccentricity whose value does not change. This proves the convergence of Algorithm 1.

Solution 4

P_i reads the calculated h-values of its neighbor. Therefore, at C_{safe}, P_i would have known if it met the criteria for a center set, that's P_i value must be greater than or equal to its neighbor.

The above answer also applies when there is more than one node in the center set.

Solution 5

There's no need to modify Algorithm 1 because, at C_{safe}, every node reads the h-value of its neighbors to determine their position towards the center set. For example, if a node determines that it's h-value is less than its neighbor, then that node is not part of the center set.