Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2016

Prof. R. Wattenhofer
Darya Melnyk

Principles of Distributed Computing
Exercise 12

1 Self-stabilizing Spanning Tree

In this exercise, we are searching for efficient, self-stabilizing spanning tree algorithms. Thus, the spanning tree will be adapted if the topology changes or nodes and/or links fail.

a) Prove an (asymptotically) optimal lower bound on the time complexity of any self-stabilizing spanning tree algorithm!

b) If the transformation from the lecture is applied to the Bellman-Ford BFS algorithm (see Algorithm 1), is the time complexity of the resulting self-stabilizing algorithm optimal? On average, how much more information per round must be transmitted compared to the original algorithm?

Algorithm 1 Bellman-Ford BFS

Each node u stores an integer d_u which corresponds to the distance from u to the root. Initially $d_{\text{root}} = 0$, and $d_u = \infty$ for every other node u.
The root starts the algorithm by sending "1" to all neighbors.
if a node u receives a message "y" with $y < d_u$ from a neighbor v **then**
  node u sets $d_u := y$
  node u sends "$y + 1$" to all neighbors (except v)
end if

c) Assume that the diameter D is known to the leader (i.e., the root of the constructed tree) Can you give an algorithm stabilizing—up to constant factors—in the same time, but sending not more information in each round than the unmodified Bellman-Ford algorithm? If yes, can your approach be generalized to other algorithms? If no, prove a corresponding lower bound!