Institute of Telematics

Technical Report

Worst-Case Analysis of a Self-Stabilizing Algorithm Computing a Weakly Connected Minimal Dominating Set

Bernd Hauck

October 2008

urn:nbn:de:gbv:830-tubdok-5126

Hamburg University of Technology
Schwarzenbergstraße 95, 21073 Hamburg, Germany
phone: +49 40 42878-3531; Fax:+49 40 42878-2581
email: telematik@tuhh.de
web: http://www.ti5.tu-harburg.de

Worst-Case Analysis of a Self-Stabilizing Algorithm Computing a Weakly Connected Minimal Dominating Set

Bernd Hauck [a]

[a]Hamburg University of Technology, Institute of Telematics, Schwarzenbergstraße 95, 21073 Hamburg, Germany

Recently, Srimani and Xu presented a self-stabilizing algorithm that computes a weakly connected minimal dominating set [2]. They prove an upper bound of $O(2^n)$ until stabilization but they do not provide a lower bound. This paper verifies by giving an example that their algorithm indeed requires $O(2^n)$ moves on a certain graph.

Keywords: Self-Stabilizing algorithms, fault tolerance, distributed computing, graph algorithms

1. Introduction

A distributed system is self-stabilizing if it can start at any possible global configuration and regain consistency in a finite number of steps by itself without any external intervention and remains in a consistent state. Detailed information and a more formal definition of self-stabilization can be found e.g. in [1].

Let $G = (V, E)$ be a connected undirected graph, $|V| = n$ and $|E| = m$. A dominating set S of G is a subset of V such that each $v \in V \backslash S$ has at least one neighbor in S. S is a minimal dominating set if for any node $v \in S$ the set $S \backslash \{v\}$ is not dominating. A dominating set S is called *weakly connected* if the subgraph weakly induced by S, i.e. the graph $(N[S], E \cap (S \times N[S]))$ is connected.

Recently, Srimani and Xu presented the first self-stabilizing algorithm that computes a weakly connected minimal dominating set (WCMDS) [2].

2. Algorithm of Srimani and Xu

The algorithm of Srimani and Xu [2] requires a breadth-first spanning tree in the given graph. A self-stabilizing algorithm that establishes such a tree is also presented in [2]. It assumes unique node identifiers and the node with maximum ID is chosen to be the root of the spanning tree. This algorithm initializes the variables $P(i)$ that stores the parent node of a node i, and $L(i)$ which keeps the distance in hops to the root node. The root node r has $P(r) = r$ and $L(r) = 0$. The boolean variable F_i denotes, if node i is a member of the WCMDS or not. Algorithm 1 shows Srimani and Xu's set of rules.

Via the first rule the root node enters the WCMDS, if it is not included already. The second rule makes a node leave the set if its parent node is included. Otherwise it enters the set itself.

3. Complexity Analysis

Srimani and Xu prove that their Algorithm 1 stabilizes after at most $O(2^n)$ moves. However, they do not perform a worst-case analysis to verify that it in fact requires $O(2^n)$ moves at all. The following example provides a lower bound for Algorithm 1.

Let G_k be a graph that is composed of two nodes, one of them regarded as root, and k circles $C_1, \ldots C_k$ one after another which consist of eight nodes each. Every C_i contains a node $v_{s,i}$ and a node $v_{t,i}$ with distance 4. For all components C_i and C_{i+1}: $v_{t,i} = v_{s,i+1}$. Figure 1 shows G_3.

It is possible to initialize circle C_i in a way that allows node $v_{t,i}$ to make twice as much moves as $v_{s,i}$, if $v_{s,i}$ gets enabled by a node with higher level. Figure 2 shows an execution of Algorithm 1 on C_1 that

Algorithm 1 WCMDS Algorithm of Srimani and Xu

R1: (root node)
if $P_i = i$
then $F_i := true$

R2: (non-root nodes)
if $(P_i \neq i) \wedge (\exists j \in N(i)\ s.t.\ L_j \leq L_i \wedge F_j = true)$
then $F_i := false$
else $F_i := true$

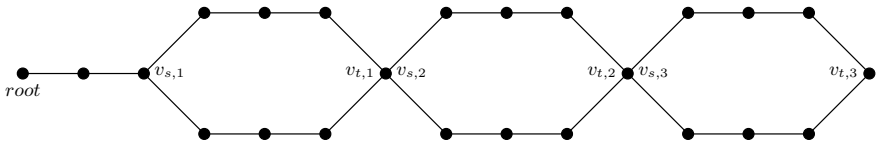

Figure 1. Graph G_3

demonstrates this behavior (the nodes that enable $v_{s,1}$ are not included in the figure). $v_{s,1}$ changes its state twice and $v_{t,1}$ changes its state four times. Nodes with $F_i = true$ are colored black, the others are white.

It is now easy to construct a graph and a initial configuration that leads to an exponential number of moves until stabilization: Consider the initial configuration given in Figure 3: As shown above, $v_{t,i}$ can make twice as much moves as $v_{s,i}$, if $v_{s,i}$ gets enabled by a node with higher level. In the worst case the nodes of a circle C_i do not perform a move if a node of a circle C_j with $i < j$ is enabled. Thus, there is an execution of Algorithm 1 of G_k (consisting of $7k + 2$ nodes) in which node $v_{t,k}$ can make 2^k moves. Hence, $O(2^n)$ is also a lower bound for Algorithm 1.

4. Conclusion

This paper analyzed the worst case complexity of an algorithm by Srimani and Xu that computes a weakly connected minimal dominating set [2]. By giving an example it verifies that $O(2^n)$ is not only an upper bound but also a lower bound for the number of moves until stabilization.

REFERENCES

1. S. Dolev. *Self-stabilization.* MIT Press, Cambridge, USA, 2000.
2. P. K. Srimani and Z. Xu. Self-stabilizing algorithms of constructing spanning tree and weakly connected minimal dominating set. In *Proc. 27th Int. Conf. on Distributed Computing Systems Workshops*, page 3, 2007.

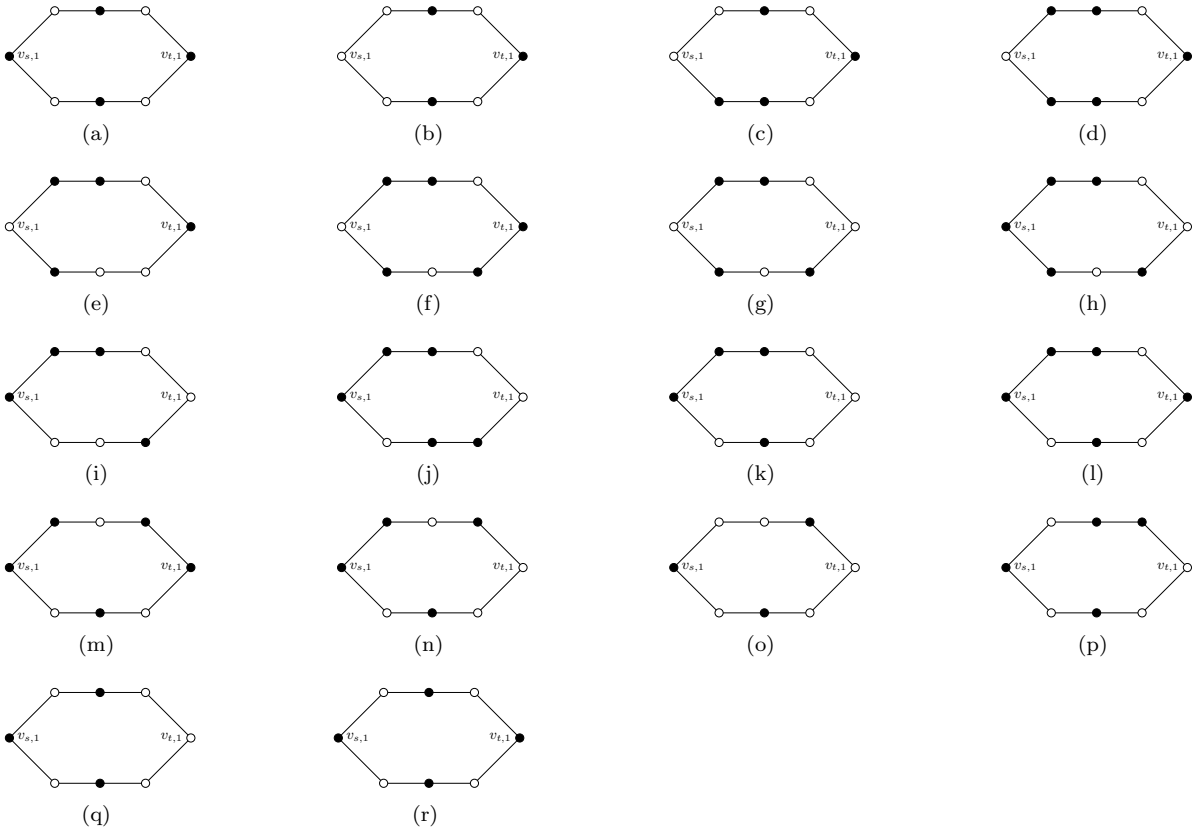

Figure 2. Execution of Algorithm 1 on a circle

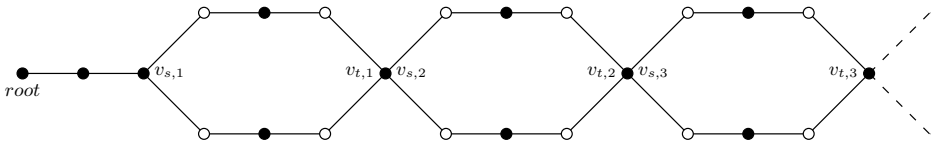

Figure 3. Initial Configuration of the WCMDS algorithm on Graph G_k