A Rigorous Proof of O(n^2) Bound on the Number of Moves for Stabilization of Dijkstra's 3-State Algorithm

Neeraj Mittal and Vijay K. Garg

TR-PDS-2001-005 December 2001

Parallel and Distributed Systems Group
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712

A Rigorous Proof of O($\mathbf{n^2}$) Bound on the Number of Moves for Stabilization of Dijkstra's 3-State Algorithm

Neeraj Mittal
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188, USA
neerajm@cs.utexas.edu http://www.cs.utexas.edu/users/neerajm

Vijay K. Garg*
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084, USA
garg@ece.utexas.edu http://www.ece.utexas.edu/~garg

Abstract

We give a rigorous proof of $O(n^2)$ bound on the number of moves Dijkstra's 3-state self-stabilizing algorithm for token-ring takes to reach a legal state. To our knowledge, ours is the first formal *proof of the $O(n^2)$ bound.*

Keywords: distributed systems, self-stabilization, analysis of algorithms

For the notation used in this paper please refer to [Dij86]. Our proof uses the following three different metrics.

$$x = \text{the number of arrows}$$

$$y = \text{the number of left-pointing arrows} + \text{twice the number of right-pointing arrows}$$

$$z = \sum_{\substack{\text{left-pointing}\\ \text{arrows}}} \text{distance from Bottom} + \sum_{\substack{\text{right-pointing}\\ \text{arrows}}} \text{distance from Top}$$

Note that the first two metrics were used by Dijkstra in [Dij86] to establish that the 3-state algorithm eventually reaches a legal state. The maximum initial value of metrics x,

*supported in part by the NSF Grants ECS-9907213, CCR-9988225, Texas Education Board Grant ARP-320, an Engineering Foundation Fellowship, and an IBM grant.

y and z is n, $2n$ and n^2, respectively. The transformations for each move in terms of arrows [Dij86] and the corresponding change in metrics x, y and z is as follows.

For Bottom:

from	$B \leftarrow R$	to	$B \rightarrow R$	$\Delta x = 0$	$\Delta y = +1$	$\Delta z = n - 1$	(0)

For a normal machine:

from	$L \rightarrow S \quad R$	to	$L \quad S \rightarrow R$	$\Delta x = 0$	$\Delta y = 0$	$\Delta z = -1$	(1)
from	$L \quad S \leftarrow R$	to	$L \leftarrow S \quad R$	$\Delta x = 0$	$\Delta y = 0$	$\Delta z = -1$	(2)
from	$L \rightarrow S \leftarrow R$	to	$L \quad S \quad R$	$\Delta x = -2$	$\Delta y = -3$	$\Delta z = -n + 1$	(3)
from	$L \rightarrow S \rightarrow R$	to	$L \quad S \leftarrow R$	$\Delta x = -1$	$\Delta y = -3$	$\Delta z \leqslant n - 2$	(4)
from	$L \leftarrow S \leftarrow R$	to	$L \rightarrow S \quad R$	$\Delta x = -1$	$\Delta y = 0$	$\Delta z \leqslant n - 2$	(5)

For Top:

from	$L \rightarrow T$	to	$L \leftarrow T$	$\Delta x = 0$	$\Delta y = -1$	$\Delta z = n - 1$	(6)
from	$L \quad T$	to	$L \leftarrow T$	$\Delta x = +1$	$\Delta y = +1$	$\Delta z = n - 1$	(7)

Let m_i denote the number of moves corresponding to the transformation (i), where $i \in [0 \dots 7]$, made by the system before it stabilizes. It is proved in [Dij86] that if the algorithm has not yet reached a legal state then y decreases by at least 1 per move of Bottom. As the initial value of y is at most $2n$, $m_0 \leqslant 2n$. It is also proved in [Dij86, Lemma 0] that between two successive moves of Top at least one move of Bottom takes place. Therefore $m_6 + m_7 \leqslant 2n$. We now bound the number of moves made by normal machines. Note that the transformations (3), (4) and (5) decrease x whereas the transformations (0), (1) and (2) do not change x at all. Further, a move by Top increases x by at most 1 and the total number of such moves is upper-bounded by $2n$. Since the maximum initial value of x is n, $m_3 + m_4 + m_5 \leqslant 3n$. Finally, since the transformations (1) and (2) decrease z, the other transformations—which are at most $7n$ in total—increase it by at most n and the maximum initial value of z is n^2, $m_1 + m_2 \leqslant 8n^2$ giving us an upper-bound of $O(n^2)$ on the total number of moves required for the system to stabilize. The following example establishes that it is, in fact, a tight upper bound. Consider the string consisting of $(n-1)/2$ right-pointing arrows followed by $(n-1)/2$ spaces, for some odd $n > 3$. By repeatedly using transformation (1) $(n-1)^2/4$ times, we obtain the string consisting of $(n-1)/2$ spaces followed by $(n-1)/2$ right-pointing arrows; each arrow moves to the right by $(n-1)/2$ places with the system staying in illegal state throughout.

References

[Dij74] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Communications of the ACM (CACM)*, 17(11):643–644, 1974.

[Dij86] E. W. Dijkstra. A Belated Proof of Self-Stabilization. *Distributed Computing*, 1:5–6, 1986.