Available online at www.sciencedirect.com
SCIENCE DIRECT®

Electronic Notes in Theoretical Computer Science

ELSEVIER Electronic Notes in Theoretical Computer Science 128 (2005) 53–68
www.elsevier.com/locate/entcs

Simplifying Itai-Rodeh Leader Election for Anonymous Rings

Wan Fokkink[1]

Department of Software Engineering, CWI, Amsterdam, The Netherlands
Department of Computer Science, Vrije Universiteit Amsterdam, The Netherlands

Jun Pang[2]

Department of Software Engineering, CWI, Amsterdam, The Netherlands

Abstract

We present two probabilistic leader election algorithms for anonymous unidirectional rings with FIFO channels, based on an algorithm from Itai and Rodeh [14]. In contrast to the Itai-Rodeh algorithm, our algorithms are finite-state. So they can be analyzed using explicit state space exploration; we used the probabilistic model checker PRISM to verify, for rings up to size four, that eventually a unique leader is elected with probability one.

Keywords: Distributed computing, leader election, anonymous networks, probabilistic algorithms, model checking.

1 Introduction

Leader election is the problem of electing a unique leader in a network, in the sense that the leader (process) knows that it has been elected and the other processes know that they have not been elected. Leader election algorithms require that all processes have the same local algorithm and that each computation terminates, with one process elected as leader. This is a fundamental

[1] Email: wan@cwi.nl; wanf@cs.vu.nl
[2] Email: pangjun@cwi.nl

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2005.04.004

problem in distributed computing and has numerous applications. For example, it is an important tool for breaking symmetry in a distributed system. By choosing a process as the leader it is possible to execute centralized protocols in a decentralized environment. Leader election can also be used to recover from token loss for token-based protocols, by making the leader responsible for generating a new token when the current one is lost.

There exists a broad range of leader election algorithms. These algorithms have different message complexity in worst and/or average case. Furthermore, they vary in communication mechanism (*asynchronous vs. synchronous*), process names (*unique identities vs. anonymous*), and network topology (e.g. *ring, tree, complete graph*).

A first leader election algorithm for unidirectional rings was given by Le Lann [17]. It requires that each process has a unique identity, with a total ordering on identities; the process with the largest identity becomes the leader. The basic idea of Le Lann's algorithm is that each process sends a message around the ring bearing its identity. Thus it requires a total of n^2 messages, where n is the number of processes in the ring. Chang and Roberts [7] improved Le Lann's algorithm by letting only the message with the largest identity complete the round trip; their algorithm still requires in the order of n^2 messages in the worst case, but only $n \log n$ on average. Franklin [10] developed a leader election algorithm for bidirectional rings with a worst-case message complexity of $\mathcal{O}(n \log n)$. Peterson [18] and Dolev, Klawe, and Rodeh [8] independently adapted Franklin's algorithm so that it also works for unidirectional rings. All the above algorithms work both for asynchronous and for synchronous communication, and do not require a priori knowledge about the number of processes.

Sometimes the processes in a network cannot be distinguished by means of unique identities. First, as the number of processes in a network increases, it may become difficult to keep the identities of all processes distinct; or a network may accidentally assign the same identity to different processes. Second, identities cannot always be sent around the network, for instance for reasons of efficiency. An example of the latter is FireWire, the IEEE 1394 high performance serial bus. A leader election algorithm that works in the absence of unique process identities is also desirable from the standpoint of fault tolerance. In an *anonymous network*, processes do not carry an identity. Angluin [2] showed that there does not exist a terminating algorithm for electing a leader in an asynchronous anonymous network. According to this result, a *Las Vegas* algorithm (meaning that the probability that the algorithm terminates is greater than zero, and all terminal configurations are correct) is the best possible option.