Theoretical Computer Science 561 (2015) 134–143

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Sublinear bounds for randomized leader election

Shay Kutten [a,1], Gopal Pandurangan [b,c,2], David Peleg [d,3], Peter Robinson [b,*,4], Amitabh Trehan [a,1]

[a] Information Systems Group, Faculty of Industrial Engineering and Management, Technion – Israel Institute of Technology, Haifa 32000, Israel
[b] Division of Mathematical Sciences, Nanyang Technological University, 637371 Singapore
[c] Department of Computer Science, Brown University, Box 1910, Providence, RI 02912, USA
[d] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel

ARTICLE INFO

Available online 22 February 2014

Keywords:
Distributed algorithm
Leader election
Randomization
Lower bound

ABSTRACT

This paper concerns *randomized* leader election in synchronous distributed networks. A distributed leader election algorithm is presented for complete n-node networks that runs in $O(1)$ rounds and (with high probability) uses only $O(\sqrt{n}\log^{3/2} n)$ messages to elect a unique leader (with high probability). When considering the "explicit" variant of leader election where eventually every node knows the identity of the leader, our algorithm yields the asymptotically optimal bounds of $O(1)$ rounds and $O(n)$ messages. This algorithm is then extended to one solving leader election on any connected non-bipartite n-node graph G in $O(\tau(G))$ time and $O(\tau(G)\sqrt{n}\log^{3/2} n)$ messages, where $\tau(G)$ is the mixing time of a random walk on G. The above result implies highly efficient (sublinear running time and messages) leader election algorithms for networks with small mixing times, such as expanders and hypercubes. In contrast, previous leader election algorithms had at least linear message complexity even in complete graphs. Moreover, super-linear message lower bounds are known for time-efficient *deterministic* leader election algorithms. Finally, we present an almost matching lower bound for randomized leader election, showing that $\Omega(\sqrt{n})$ messages are needed for any leader election algorithm that succeeds with probability at least $1/e + \varepsilon$, for any small constant $\varepsilon > 0$. We view our results as a step towards understanding the randomized complexity of leader election in distributed networks.

---

[*] Corresponding author.

E-mail addresses: kutten@ie.technion.ac.il (S. Kutten), gopalpandurangan@gmail.com (G. Pandurangan), david.peleg@weizmann.ac.il (D. Peleg), peter.robinson@ntu.edu.sg (P. Robinson), amitabh.trehaan@gmail.com (A. Trehan).

[1] Supported by the Israeli Science Foundation and by the Technion TASP Center.

[2] Research supported in part by the following grants: Nanyang Technological University Grant M58110000, Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 2 Grant MOE2010-T2-2-082, and a grant from the US–Israel Binational Science Foundation (BSF).

[3] Supported in part by the Israel Science Foundation (Grant 894/09), the United States–Israel Binational Science Foundation (Grant 2008348), and the Israel Ministry of Science and Technology (infrastructures grant).

[4] Research supported in part by the following grants: Nanyang Technological University Grant M58110000, Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 2 Grant MOE2010-T2-2-082.

http://dx.doi.org/10.1016/j.tcs.2014.02.009
0304-3975/

1. Introduction

1.1. Background and motivation

Leader election is a classical and fundamental problem in distributed computing. It originated as the problem of regenerating the "token" in a local area *token ring* network [16] and has since then "starred" in major roles in problems across the spectrum, providing solutions for reliability by replication (or duplicate elimination), for locking, synchronization, load balancing, maintaining group memberships and establishing communication primitives. As an example, the content delivery network giant Akamai uses decentralized and distributed leader election as a subroutine to tolerate machine failure and build fault tolerance in its systems [21]. In many cases, especially with the advent of large scale networks such as peer-to-peer systems [25,26,31], it is desirable to achieve low cost and scalable leader election, even though the guarantees may be probabilistic.

Informally, the problem of distributed leader election requires a group of processors in a distributed network to elect a unique leader among themselves, i.e., exactly one processor must output the decision that it is the leader, say, by changing a special *status* component of its state to the value *leader* [18]. All the rest of the nodes must change their status component to the value *non-leader*. These nodes need not be aware of the identity of the leader. This *implicit* variant of leader election is rather standard (cf. [18]), and is sufficient in many applications, e.g., for token generation in a token ring environment. This paper focuses on implicit leader election (but improves the upper bounds also for the explicit case, by presenting a time and message optimal randomized protocol).

In another variant, all the non-leaders change their status component to the value *non-leader*, and moreover, every node must also know the identity of the unique leader. This formulation may be necessary in problems where nodes coordinate and communicate through a leader, e.g., implementations of Paxos [5,15]. In this variant, there is an obvious lower bound of $\Omega(n)$ messages (throughout, n denotes the number of nodes in the network) since every node must be informed of the leader's identity. This *explicit* leader election can be achieved by simply executing an (implicit) leader election algorithm and then broadcasting the leader's identity using an additional $O(n)$ messages and $O(D)$ time (where D is the diameter of the graph).

The complexity of the leader election problem and algorithms for it, especially deterministic algorithms (guaranteed to always succeed), have been well-studied. Various algorithms and lower bounds are known in different models with synchronous/asynchronous communication and in networks of varying topologies such as a cycle, a complete graph, or some arbitrary topology (e.g., see [9,18,22,27,30] and the references therein). The problem was first studied in context of a ring network by Le Lann [16] and discussed for general graphs in the influential paper of Gallager, Humblet, and Spira [6]. However, leader election in the class of complete networks has come to occupy a special position of its own and has been extensively studied [1,8,10,12,13,28]; see also [4,17,29] for leader election in complete networks where nodes have a sense of direction.

The study of leader election algorithms is usually concerned with both message and time complexity. For complete graphs, Korach et al. [11] and Humblet [8] presented $O(n\log n)$ message algorithms. Korach, Kutten, and Moran [10] developed a general method decoupling the issue of the graph family from the design of the leader election algorithm, allowing the development of message efficient leader election algorithms for any class of graphs, given an efficient traversal algorithm for that class. When this method was applied to complete graphs, it yielded an improved (but still $\Omega(n\log n)$) message complexity. Afek and Gafni [1] presented asynchronous and synchronous algorithms, as well as a tradeoff between the message and the time complexity of synchronous *deterministic* algorithms for complete graphs: the results varied from an $O(1)$-time, $O(n^2)$-messages algorithm to an $O(\log n)$-time, $O(n\log n)$-messages algorithm. Singh [28] showed another trade-off that saved on time, still for algorithms with a super-linear number of messages. (Sublinear time algorithms were shown in [28] even for $O(n\log n)$ messages algorithms, and even lower times for algorithms with higher messages complexities.) Afek and Gafni, as well as [11,13] showed a lower bound of $\Omega(n\log n)$ messages for *deterministic* algorithms in the general case. One specific case where the message complexity could be reduced (but only as far as linear message complexity) was at the expense of also having a linear time complexity, see [1]. Multiple studies showed a different case where it was possible to reduce the number of messages to $O(n)$, by using a *sense of direction* – essentially, assuming some kind of a virtual ring, superimposed on the complete graph, such that the order of nodes on a ring is known to the nodes [4]. The above results demonstrate that the number of messages needed for deterministic leader election is at least linear or even super-linear (depending on the time complexity). In particular, existing $O(1)$ time deterministic algorithms require $\Omega(n^2)$ messages (in a complete network).

At its core, leader election is a symmetry breaking problem. For anonymous networks under some reasonable assumptions, deterministic leader election was shown to be impossible [2] (using symmetry arguments). Randomization comes to the rescue in this case; random rank assignment is often used to assign unique identifiers, as done herein. Randomization also allows us to beat the lower bounds for deterministic algorithms, albeit at the risk of a small chance of error.

A randomized leader election algorithm (for the explicit version) that could err with probability $O(1/\log^{\Omega(1)} n)$ was presented in [24] with time $O(\log n)$ and linear message complexity.[5] That paper also surveys some related papers about

---

[5] In contrast, the probability of error in the current paper is $O(1/n^{\Omega(1)})$.