Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 351–358

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

Electing a Leader in Dynamic Networks using Mobile Agents and Local Computations

Mouna Ktari[a], Mohamed Mosbah[b], Ahmed Hadj Kacem[a]

[a]ReDCAD Laboratory, University of Sfax, FSEGS, 3018 Sfax, Tunisia
[b]LaBRI, Bordeaux INP, CNRS, University of Bordeaux, F-33405 Talence, France

Abstract

In dynamic distributed systems, the topology of networks changes over time which makes difficult the design and much harder the proof of distributed algorithms. These unavoidable changes of the topology make the election and the maintenance of the elected leader a complex task. The maintaining problem is not considered in a static context. To encode distributed algorithms, we adopt the local computation model. Distributed algorithms are formally presented by transitions systems. Based on both, the mobile agent paradigm and the local computation model, we present in this paper a distributed algorithm that elects a leader in a tree. A set of topological events that may affect the structure of the tree: we focus on the appearance and the disappearance of places as well as the communication channels. Our goal is to maintain always a tree with a single leader or a forest of trees where each one has his own leader.

Keywords: Distributed Algorithms; Local computations; Election; Mobile Agents; Dynamic Networks.

1. Introduction

Leader election is a classical problem in distributed systems and applications. This problem aims to elect one entity among a set of connected entities and to break the symmetry of distributed environments. There are many variants of leader election algorithms in distributed networks. They consider the different models of the distributed systems, different communication systems, different parameters related to the network topology and the initial knowledge[1,2,3].

Several variants of leader election algorithms in distributed systems are proposed in the literature. The majority of proposed algorithms to solve this problem are based on message passing systems[4,5,6,7,8]. However, with the success of mobile agents based on applications[9], regards are switched from classical models based on stationary process (message passing, remote procedure call, shared memory, etc) towards this new paradigm. A mobile agent is an

* Mouna Ktari. Tel.: +2-169-416-8968.
E-mail address: mouna.ktari@redcad.org

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.363

autonomous entity which moves through a network under its own control and interacts with resources and local environments. In computer software technology, this program is a computing technique oriented to solve a large scale of problems in computer science. In a practical level, agents perform operations independent of the processors, thereby lending to more flexible algorithms supporting different types of networks. For such a network, they led to the considerable reduction of network load. Contrariwise, in communication models based on message passing systems, a distributed algorithm must be run on any entities (nodes) of computing systems. The application of a computing step must be preceded by a synchronization, but this synchronization not only delays the execution of the algorithm but also uses a great amount of communication resources [11].

Based on mobile agent paradigms, Deugo [10] has proposed a technique for mapping a distributed algorithm to a set of homogeneous agents solving the election problem in a ring. Such a mobile agent does not openly communicate with other agents via messages. Instead, agents exchange information through briefings, posted to the environments. The comparison between the number of messages sent and the number of agents used is done. In [12], a catalog of solutions of the leader election problem in several static network topologies is proposed. Distributed algorithms are modeled using the high level encoding of mobile agents in a local computation model. The proposed solutions, contrary to classical ones, do not need a great amount of computational and communicational resources. In [13], self-stabilizing mobile agents are used to execute the leader election among processors in a distributed network. Leader election in terms of self-stabilization methodology can be interpreted as a system that is unstable when it has no leader and attains stability when a processor is elected as a leader among the processors in the distributed network.

Towards mobile agents, the distributed computing community is presenting an increasing interest due to their considerable reduction of network load and their overcoming of the network latency [9]. In fact, the implementation of local computations using mobile agents in a static context has confirmed this interest [12,14]. In the same context of local computations, it remains a challenge to prove the mobile agent computational power in dynamic networks. Our work is a step in this direction. We are motivated by the increasing need to develop distributed algorithms executed by mobile agents and using local computation model in dynamic networks. Abstracted by graphs, these networks are characterized by a constant evolution over time and they change at any time through the addition or removal of nodes and/or edges (communication links). Consequently, distributed computing becomes much harder to design and to prove than a static context. We will present in this paper an electing algorithm of a leader in a tree where the underlying graph is dynamic. The unavoidable changes of the topology makes the election and the maintenance of the elected leader a complex task. Our goal is to maintain a tree with a single leader or a forest of trees where each one has his own leader.

To model distributed algorithms, we have proposed a framework for designing, proving and simulating distributed algorithms in dynamic networks [15]. This model gets benefits from both mobile agent paradigm and local computations theory. To illustrate our model, we have proposed a catalog of distributed algorithms. A solution to the spanning tree problem has been proposed in [16]. A second algorithm will be presented in this paper, electing a leader in a tree. Our algorithm is divided in two steps, the election of a leader and its maintenance following the detection of topological events (appearance or disappearance of places and communication links). We present a description of this algorithm as a transition system.

To model dynamic networks, several models exist in the literature. Erdos and Renyi [17] is the mother of all models, it is known by a “random graph model”. In general, modeling dynamic networks based on this model requires a fully acquaintance of the evolution of the dynamic graph. To get closer to real networks, we have adopted another model that covers a variety of dynamic networks. This model is known as the evolving graph model. It is characterized by a lack of prior knowledge of the evolution of dynamic graphs [18]. This paper is organized as follows. We present in the second section an overview of the adopted model to formally present distributed algorithms. A solution for the leader election problem is presented in the third section. The last section gives a conclusion and a plan for future works.