Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Optimal self-stabilizing synchronous mobile Byzantine-tolerant atomic register

Silvia Bonomi [a,*], Antonella Del Pozzo [a,b], Maria Potop-Butucaru [b]

[a] *DIAG - University of Rome "La Sapienza", Rome, Italy*
[b] *Sorbonne Universités, UPMC, LIP6-CNRS 7606, Paris, France*

ARTICLE INFO

Article history:
Received 12 April 2016
Received in revised form 24 July 2017
Accepted 22 August 2017
Available online xxxx

Keywords:
Self-stabilizing atomic storage
Byzantine mobile agents
Round-based synchronous computation

ABSTRACT

This paper addresses for the first time the problem of MWMR atomic memory in a *Mobile Byzantine Agents* model. The register is maintained by n servers and faulty (Byzantine) agents move from one server to another and when they are affecting a server, this one behaves arbitrarily. This paper addresses the round-based synchronous communication model. We focus on four Mobile Byzantine Failure models differing in the diagnosis capabilities at server side. We address the case when servers can diagnose their failure state (that is, servers are aware that the mobile Byzantine agent has left), and the case when servers cannot self-diagnose.

We first prove lower bounds on the number of servers n necessary to construct a safe register tolerant to the presence of $f < n$ Mobile Byzantine Failures for four Mobile Byzantine Failure models. Additionally, we prove that our lower bounds do not change when the system is affected by any number of *transient failures*.

Furthermore, we propose a parametric algorithm that implements an atomic MWMR register algorithm working in all the above models and matches the lower bounds. Additionally, our algorithm is also self-stabilizing. That is, started in an arbitrary state (i.e. after the occurrence of a transient failure) it is able to self-recover a correct behavior in a finite, bounded number of rounds. Our algorithm tolerates (i) any number of transient failures and (ii) up to f Mobile Byzantine Failures.

1. Introduction

To ensure high availability, storage services are usually implemented by replicating data at multiple locations and maintaining such data consistent. Thus, replicated servers represent today an attractive target for attackers that may try to compromise replicas correctness for different purposes. Some examples are: to gain access to protected data, to interfere with the service provisioning (e.g. by delaying operations or by compromising the integrity of the service), to reduce service availability with the final aim to damage the service provider (reducing its reputation or letting it pay for the violation of service level agreements) etc. In addition, another emerging issue in the design of replicated services is the possibility of having *transient failures* as consequence of possible corruption in replicas state due to errors in the communication or in the computation.

[*] Corresponding author.
E-mail addresses: bonomi@dis.uniroma1.it (S. Bonomi), delpozzo@dis.uniroma1.it (A. Del Pozzo), maria.potop-butucaru@lip6.fr (M. Potop-Butucaru).

http://dx.doi.org/10.1016/j.tcs.2017.08.020
0304-3975/© 2017 Elsevier B.V. All rights reserved.

A compromised replica is usually modeled through an arbitrary failure (i.e. a Byzantine failure) that is made transparent to clients by employing Byzantine Fault Tolerance (BFT) techniques.

As pointed out in [1], in addition to classical Byzantine behaviors, it is worth considering *mobile adversaries*. Mobile adversaries have been primarily introduced in the context of multi-party computation and they model an attacker that is able to progressively compromise computational entities but only for a limited period of time. Therefore, tolerating Mobile Byzantine Failures is, in some sense, similar to having a bounded number of compromised entities at any given time but such set changes over time. Such model captures phenomena like virus injection (where viruses start to infect the network but then they are detected and progressively deleted from a set of machines). Without additional corrective actions every process can be compromised in a long lasting execution leaving the system without any correct process able to progress in the computation (i.e., it could be impossible to define an upper bound $f < n$ for long lasting executions that guarantees correctness of the system). Therefore, systems that cope with Mobile Byzantine Failures have to execute programmed maintenance with the aim of restoring potentially infected machines or self-repairing themselves [2].

From a theoretical point of view, mobile adversaries have been formalised in different *Mobile Byzantine Failures* models [3], [4], [5], [6].

In the context of distributed storage implementations (e.g. register abstraction), common approaches to BFT are based on the deployment of a sufficiently large number of replicas to tolerate an estimated number of compromised servers (i.e. BFT replication). However, few efforts have been spent in addressing multiple type of failures i.e., how to cope with both Byzantine and transient failures. In addition, to the best of our knowledge, no storage abstraction has been investigated so far assuming mobile adversaries.

Contribution. In this paper, we address the problem of building a self-stabilizing Multi-Writer/Multi-Reader (ss-MWMR) atomic register in the presence of both Byzantine Mobile Failures and transient failures. Concerning Mobile Byzantine Failures, we considered the four theoretical models introduced by Garay, [3], Buhrman et al. [4], Sasaki et al. [5] and Bonnet et al. [6].

The paper provides two main contributions: (i) it proves a set of lower bounds (one for each of the four considered models) on the number of servers, n, necessary to implement an atomic register in presence of both transient failures and mobile Byzantine failures (i.e., under multi-failure assumption) and (ii) a parametric algorithm implementing a self-stabilizing atomic register in a synchronous round-based message passing system under multi-failure assumption, working in all the four models [3–6].

Let us note that the complexity of the proposed algorithm matches the computed lower bounds. As a consequence, the computed bounds are tight in the considered model and the proposed algorithm is optimal. As far as we know, our construction is the first that builds a distributed self-stabilizing MWMR atomic register in the considered environment.

More in details, we proved that the cost, in terms of minimum number of servers n, of implementing a MWMR atomic register under multi-failure assumption (i.e., f mobile Byzantine Failures and an arbitrary number of transient failures) are: $n \geq 3f + 1$ in the Garay's model, $n \geq 4f + 1$ in the Sasaki et al.'s model and Bonnet et al.'s model and $n \geq 2f + 1$ Buhrman et al.'s model.

Roadmap. The paper is organized as follows. Section 2 discusses Related Works and Section 3 presents the system model and the problem specification. Section 4 shows lower bounds on the number of servers necessary to implement self-stabilizing safe register in the following Mobile Byzantine Failure models: Garay [3], Buhrman et al. [4], Sasaki et al. [5] and Bonnet et al. [6]. In Section 5 we present a generic tight algorithm that implements self-stabilizing MWMR atomic register parametrized in function of the considered mobile Byzantine model. The correctness of the generic algorithm is proved in Section 5.2. Finally, Section 6 concludes the paper and discusses some open research directions.

2. Related work

Computations under Mobile Byzantine Failure Models. Concerning Mobile Byzantine Failures models, there are two main research directions: (i) Byzantines with constrained mobility and (ii) Byzantines with unconstrained mobility. Byzantines with constraint mobility were studied by Buhrman et al. [4]. They consider that Byzantine agents move from one node to another only when protocol messages are sent (similar to how viruses would propagate). In [4], Buhrman et al. studied the problem of Mobile Byzantine Agreement. Intuitively, Mobile Byzantine Agreement, ensures that correct processes agree in a finite time on an initially proposed value. They proved a tight bound for its solvability (i.e., $n > 3f$, where f is the maximal number of simultaneously faulty processes) and proposed a time optimal protocol that matches this bound.

In the case of unconstrained mobility the motion of Byzantine agents is not tied to message exchange. Several authors investigated the agreement problem in variants of this model: [3,5–9]. Reischuk [9] investigates the stability/stationarity of malicious agents for a given period of time. Ostrovsky and Yung [8] introduced the notion of mobile virus and investigate an adversary that can inject and distribute faults. However, none of these works consider the implementation of the shared memories (e.g. register).

Garay [3] and, more recently, Banu et al. [7] and Sasaki et al. [5] or Bonnet et al. [6] consider that processes execute synchronous rounds composed of three phases: *send*, *receive*, *compute*. Between two consecutive rounds, Byzantine agents can move from one host to another, hence the set of faulty processes has a bounded size although its membership can