Theoretical Computer Science 552 (2014) 99–108

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Space-efficient self-stabilizing counting population protocols on mobile sensor networks ☆

Tomoko Izumi [a,*], Keigo Kinpara [b], Taisuke Izumi [b], Koichi Wada [c]

[a] College of Information Science and Engineering, Ritsumeikan University, 1-1-1 Noji higashi, Kusatsu, Shiga 525-8577, Japan
[b] Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, 466-8555, Japan
[c] Department of Applied Informatics, Faculty of Science and Engineering, Hosei University, 3-7-2 Kajino-cho Koganei, Tokyo, 184-8584, Japan

ARTICLE INFO

Article history:
Received 29 March 2013
Received in revised form 16 July 2014
Accepted 31 July 2014
Available online 15 August 2014
Communicated by C. Kaklamanis

Keywords:
Counting problem
Symmetric population protocol
Self-stabilizing algorithm
Mobile sensor networks

ABSTRACT

In this study, we consider a self-stabilizing counting problem for a passively-mobile sensor network with a base station originally proposed by Beauquier et al. [13], where the base station must count the number of sensors in the network. Self-stabilizing counting means that the base station eventually counts the exact number of sensors in the system from the configuration where each sensor has an arbitrary initial state. In this paper, we focus on the space complexity of the self-stabilizing counting problem in terms of the number of sensor states. We propose two self-stabilizing counting protocols. Given a known upper bound P on the number of sensors, the first protocol performs counting using $2P$ sensor states and its convergence time is $O(\log n)$ in fair executions, where n is the actual number of sensors. The second protocol uses only $3 \cdot \lceil P/2 \rceil$ sensor states but assumes the global fairness, which is an assumption stronger than the standard fairness. The best known protocol requires $4P$ states while the corresponding lower bound is P, so our result reduces the gap of the feasibility between P and $4P$.

1. Introduction

1.1. Background

A *passively-mobile* system is a collection of agents that can move within a certain region with no control over their movements. The communication range of each agent is quite small compared with the size of the region, so two agents can only communicate when they are sufficiently close together. Passive mobility is present in many real systems. For example, a network of smart sensors attached to cars or animals. The *population protocols* [6] are one of the models for passively-mobile systems. A population protocol consists of a number of agents, to which some program (protocol) is deployed. According to the deployed protocol, each agent changes its state by *pairwise interactions* with other agents, i.e., two agents come close together in a region and update their states by exchanging information. In the original population protocol model, it is

☆ The conference version of this paper is published in the proceedings of The 14th International Conference on Principles of Distributed Systems (OPODIS2010).

* Corresponding author. Tel.: +81 77 561 3445.
E-mail addresses: izumi-t@fc.ritsumei.ac.jp (T. Izumi), keigo@phaser.elcom.nitech.ac.jp (K. Kinpara), t-izumi@nitech.ac.jp (T. Izumi), wada@hosei.ac.jp (K. Wada).

http://dx.doi.org/10.1016/j.tcs.2014.07.028
0304-3975/

Table 1
Previous results and our results.

♯ States	Station memory	Convergence time	Scheduler	Reference
$<P$	–	impossible	any	[13]
$3 \cdot \lceil P/2 \rceil$	$3P$	finite	globally fair	this paper
$2P$	$3P$	$O(\log n)$	fair	this paper
$4P$	$6P$	3	fair	[13]

assumed that each agent has a constant memory, and was shown that semilinear predicates are the necessary and sufficient class of the computable predicates [3,4].

In this paper, we are interested in a variant of population protocols, where the system is equipped with a base station, and consider a weaker form of the *self-stabilizing counting* problem for that model. The counting problem requires that the base station counts and outputs the exact number of sensors in the system. Self-stabilizing counting protocols must solve the counting problem from any initial configuration. However, in general, it is unsolvable unless the base station is initialized. Therefore, we assume that exceptionally the base station can be initialized, while each sensor can have any state at the initial configuration. The self-stabilizing counting problem is motivated by the following scenario. A group of birds (petrels in [13]) carry small sensors on their bodies and the base station needs to count the remaining live birds. When a bird is close enough to the base station, its sensor interacts with the base station. In addition, the sensors on the birds can also interact with each other when two birds are sufficiently close together. Each sensor is exposed to environmental challenges in nature, so its state can be corrupted by its surroundings. This problem was first posed by Beauquier et al. [13]. In that paper, several protocols working under a number of different assumptions have been proposed. In addition, the lower bound on the number of sensor states has been also shown: For the vast class of the assumptions, there is no protocol using only $P - 1$ states or less. This fact is the motivation of considering the protocols with the number of sensor states depending on P.

1.2. Our contribution

In this paper, we focus on the space complexity of the self-stabilizing counting problem in the symmetric petrels-to-base-station-and-to-petrels (STBTP) model. The STBTP model is a variation of population protocols with a base station, where sensors have no mechanism for breaking symmetry of their states. That is, this model only allows the interactions such that two interacting sensors have the same post-state if their states are the same before the interaction. Since the existence of two or more agents with the same states obviously prevents the correct counting, any protocol for the STBTP model has to provide some technical method for distinguishing sensors with the same state to solve the counting problem. In contrast, a model allowing symmetry breaking between two sensors is also proposed, which is called the asymmetric petrels-to-base-station-and-to-petrels (ATBTP) model. In this model, two sensors with the same state can have different states after their interaction. So, distinguishing two (same-state) agents can be achieved in a relatively simple way.

The weak capability of sensors incurs the restriction of the resources on the sensor side. Thus, our primary interest is the reduction of the memory space requirement of sensor nodes. In the paper originating this problem [13], two protocols, which work for the ATBTP and STBTP models, respectively, are proposed. Given an upper bound P on the number of sensors, the protocols use P and $4P$ sensor states, respectively. The study also showed that there is no protocol that can use $P - 1$ states or less for both models. Thus, the feasibility of self-stabilizing counting using less than whose sensor-side space complexity is between $4P$ and P for the STBTP model remains an open question.

Our primary contribution is to reduce this gap. We present a self-stabilizing counting protocol for the STBTP model which uses $2P$ sensor states with $O(\log n)$ rounds convergence time, where n is the actual number of sensors. This protocol assumes that executions are *fair*, as assumed in the previous study [13]. That is, it is assumed that every sensor interacts with the station and other sensors infinitely often. We also show that a strong fairness assumption, known as the *global fairness*, reduces the requisite number of sensor states for the counting problem. Informally, the global fairness assumes the livelock-freedom of executions. More precisely, if some configuration C appears in a (global-fair) execution infinitely often, any possible transition from C must occur infinitely often. Assuming the global fairness, the second protocol we propose uses only $3 \cdot \lceil P/2 \rceil$ states. Table 1 shows a comparison of the results of the previous and our studies. The convergence time is measured by *asynchronous rounds* (rounds). That is, a round is defined as the shortest fragment of an execution where each sensor interacts with all the other sensors and the station at least once. The finite convergence time in the table means that the protocol is guaranteed to converge eventually, but there is no explicit bound for the time until the convergence. As shown in Table 1, under the fairness assumption, our first protocol needs $2P$ states of a sensor and its convergence time is $O(\log n)$ rounds. In contrast, the protocol proposed by the previous work needs more states (i.e., $4P$) but converges in a constant number of rounds. Moreover, under the assumption of the global fairness, the number of sensor states can be reduced further to $3 \cdot \lceil P/2 \rceil$, but the convergence time in this case is not bounded.