Information and Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

A self-stabilizing algorithm for edge monitoring in wireless sensor networks

Brahim Neggazi [a], Mohammed Haddad [a], Volker Turau [b,*], Hamamache Kheddouci [a]

[a] University of Lyon, LIRIS UMR5205 CNRS, Claude Bernard Lyon 1 University, 43 Bd du 11 Novembre 1918, F-69622, Villeurbanne, France
[b] Hamburg University of Technology, Institute of Telematics, Am Schwarzenberg-Campus 3, 21073, Hamburg, Germany

ARTICLE INFO

Article history:
Received 15 January 2015
Available online xxxx

Keywords:
Self-monitoring
Self-stabilization
Wireless sensor networks

ABSTRACT

Self-monitoring is a simple and effective mechanism for surveilling wireless sensor networks, especially to cope against faulty or compromised nodes. A node v can monitor the communication over a link e if both end-nodes of e are neighbors of v. Finding a set of monitoring nodes satisfying all monitoring constraints is called the *edge-monitoring problem*. The minimum edge-monitoring problem is known to be NP-complete. In this paper, we present a novel self-stabilizing algorithm for computing a minimal edge-monitoring set under the unfair distributed scheduler. For sparse networks the time complexity of this new algorithm is much lower than the currently best known algorithm.

1. Introduction

A wireless sensor network (WSN) is an ad-hoc network with a large number of nodes that are micro-sensors to collect and transmit locally measured data autonomously in a multi-hop style to a gateway. Sensor networks have many applications such as facility surveillance, intrusion detection, industrial process control, or machine health monitoring.

WSNs usually operate in an autonomous mode without a human in the loop. Hence, reliability and fault tolerance are of high importance. The boundary conditions such as extreme resource limitations, high failure rates and ad hoc deployment combined with the high number of nodes preclude dependence on manual control. The dynamic and lossy nature of wireless communication caused by the primitive, low-power radio transceivers used in WSNs can lead to situations, where nodes lose synchrony and programs reach arbitrary states. Furthermore, due to the unattended nature of WSNs, they are prone to security problems. Inevitably, unattended WSNs must self-organize in response to node failures or addition of new nodes, and must adapt to changing environmental conditions [1]. Hence, there is a need for lightweight mechanisms to online monitor and diagnose network problems. As stated in [2], optimized local monitoring schemes provide a possible choice to gather and analyze network traffic, as well as building self-diagnosis and self-monitoring mechanisms while maximizing the network lifetime.

A particular form of self-monitoring is called *local monitoring*. The basic idea is to assign monitoring roles to some of the nodes in the network. Monitors exploit the convenience of overhearing due to the broadcast nature of wireless communication and dense deployments of WSNs. For this purpose, monitors are placed somewhere in the intersection of

* Corresponding author.
E-mail addresses: neggazi.brahim@gmail.com (B. Neggazi), mohammed.haddad@liris.cnrs.fr (M. Haddad), turau@tuhh.de (V. Turau), hamamache.kheddouci@liris.cnrs.fr (H. Kheddouci).

http://dx.doi.org/10.1016/j.ic.2016.09.003
0890-5401/

Please cite this article in press as: B. Neggazi et al., A self-stabilizing algorithm for edge monitoring in wireless sensor networks, Inf. Comput. (2016), http://dx.doi.org/10.1016/j.ic.2016.09.003

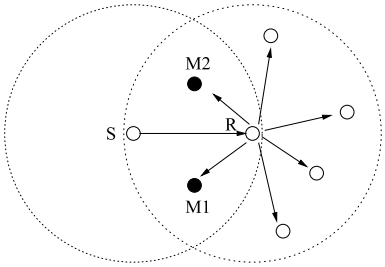

Fig. 1. Local monitoring.

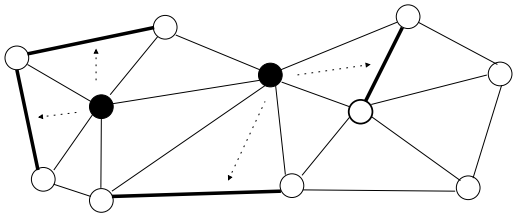

Fig. 2. The two black nodes can together monitor the four bold communication links.

the communication ranges of a sending S and a receiving node R. In the scenario sketched in Fig. 1 nodes $M1$ and $M2$ can monitor the communication from S to R, by analyzing the traffic between S and R. If $M1$ and $M2$ know the expected communication pattern between S and R, they can detect compromised or faulty nodes.

In [3] these monitoring nodes are called watchdogs. They monitor nodes by listening promiscuously to the transmissions of both nodes. When node S forwards a packet to R, the watchdog of this link verifies that node R also forwards the packet. If R does not forward the packet, then it is misbehaving. Similar to this, monitoring nodes are able to detect malicious actions such as delaying, dropping, modifying, or even fabricated packets [2,4].

The communication structure of a WSN is represented as an undirected graph $G = (V, E)$ with $n = |V|$ and $m = |E|$. A node $v \in V$ can monitor an edge $e = \langle u, w \rangle \in E$ if u and w are both neighbors of v. Let $E_M \subseteq E$ be a set of edges that require monitoring. The edge monitoring problem consists of finding a minimal set M of nodes, such that for each edge $e \in E_M$ there exists a node $v \in M$ that can monitor e. Consider for example the deployment depicted in Fig. 2. The black nodes can monitor all communication links depicted in bold. In [2], Dong et al. proved that deciding whether there exists a monitoring set with less than c nodes is an NP-complete problem.

This paper considers edge monitoring as a tool to protect a WSN from general transient faults such as memory corruption, communication errors, temporary disconnection from the network, overflow of input buffer in wireless interface, etc. and not from malicious behavior. To determine minimal monitoring sets we apply the concept of self-stabilization pioneered by Dijkstra [5]. A distributed system is self-stabilizing if after transient faults, regardless of their cause, it returns to a legitimate state in a finite number of steps regardless of the initial state, and the system remains in a legitimate state as long as no new fault occurs [6,7].

We assume that identifiers are not corrupted, e.g. they are stored in ROM as opposed to RAM. Our algorithm only determines the set of nodes that can monitor the edges, the actual monitoring task is a different subject and not part of this work. In this sense we assume that nodes execute their protocol as stated. Corruption of code, as a consequence of a fault or by a deliberate action, is clearly beyond the scope of this paper.

In 2012 Hauck proposed the first self-stabilizing algorithm to compute a minimal edge monitoring set [8]. The contribution of this paper is a self-stabilizing algorithm that improves the work of Hauck. The time complexity for sparse networks is considerably lower.

The rest of this paper is organized as follows. After reviewing related work we formally introduce the edge monitoring problem and describe our assumptions. In Section 4 we prove that it is impossible for a deterministic self-stabilizing algorithm to compute a minimal edge monitoring set. Our self-stabilizing algorithm is presented in Section 5. The proofs of correctness and termination are contained in Section 6 and 7 respectively. Section 8 concludes the paper.

2. Related work

In sensor and ad-hoc networks, the concept of local monitoring was introduced by Marti et al. in [3] for detecting malicious nodes. In the literature, two types of local monitoring are distinguished: *edge-monitoring* [2] and *self-protection* [9]. Edge monitoring is a mechanism that uses nodes to monitor links while self-protection uses nodes to provide protection to nodes themselves. Let k be a positive integer, a sensor network is k-self-protecting if each sensor (active or sleeping) is covered by at least $k-1$ active sensors [10]. This concept can be modeled as a k-tuple total dominating set problem [11].

Please cite this article in press as: B. Neggazi et al., A self-stabilizing algorithm for edge monitoring in wireless sensor networks, Inf. Comput. (2016), http://dx.doi.org/10.1016/j.ic.2016.09.003