Computer Networks 54 (2010) 1630–1645

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Heuristic algorithm for finding boundary cycles in location-free low density wireless sensor networks

Lanny Sitanayah *, Amitava Datta, Rachel Cardell-Oliver

School of Computer Science and Software Engineering, The University of Western Australia, Perth WA 6009, Australia

ARTICLE INFO

Article history:
Received 13 November 2008
Received in revised form 21 October 2009
Accepted 17 January 2010
Available online 25 January 2010
Responsible Editor: X.S. Shen

Keywords:
Wireless sensor networks
Boundary cycles
Location-free
Distributed algorithm

ABSTRACT

Wireless sensor networks (WSNs) comprise a large number of sensor nodes, which are spread out within a region to be monitored and communicate using wireless links. In some WSN applications, recognizing boundary nodes is important for topology discovery, geographic routing, tracking and guiding. In this paper, we study the problem of identifying the boundary nodes of a WSN. In a WSN, close-by nodes can establish direct communications with their neighbors and have the ability to estimate distances to nearby nodes, but not necessarily the true distances. Our objective is to find the boundary nodes by using only the connectivity relation and neighbor distance information without any other knowledge of node locations. Moreover, our main aim is to design a distributed algorithm that works even when the average degree is low. We propose a heuristic algorithm to find the boundary nodes which are connected in a boundary cycle of a location-free, low density (average degree 5–6), randomly deployed WSN. We develop the key ideas of our boundary detection algorithm in the centralized scenario and extend these ideas to the distributed scenario. The distributed implementation is more realistic for real WSNs, especially for sparse networks when all local information cannot be collected very well due to sparse connectivity. In addition, the distributed implementation can tolerate faults by recomputing the boundary locally when a boundary node is faulty. Simulations in ns-2 show that the distributed implementation outperforms the centralized one with higher quality of boundaries.

1. Introduction

Rapid improvement in wireless communication and electronics technologies have enabled the development of small, low-cost, low-power, multifunctional devices, sensor nodes. A sensor node (or mote) is a battery-powered device with integrated sensing, processing and communication capabilities. It can detect and monitor changes in a variety of physical conditions, such as temperature, humidity, light, sound, chemicals, or the presence of certain objects [17]. Nodes can perform simple computations and communicate with each other over short distances using radio. A wireless sensor network (WSN) is composed of hundreds to thousands of unattended sensor nodes and one or more base stations. The sensor nodes are deployed either densely or sparsely, manually or randomly, in a region to be monitored, for example, natural environments, battlefields, hospitals, houses and industries. The arrangement and management of a WSN depends on the application for which it is used, such as military, environmental, health, home and some commercial applications [2]. In some WSN applications, recognizing boundary nodes is necessary for topology discovery [11–13,18], geographic routing [6,13], tracking and guiding [13]. In the applications that involve tracking moving objects, such as in military applications for tracking enemy vehicles and detecting illegal border crossings, and in environmental applications for habitat monitoring [14], boundary

* Corresponding author. Tel.: +61 411 782 558.
E-mail addresses: lanny@csse.uwa.edu.au (L. Sitanayah), datta@csse.uwa.edu.au (A. Datta), rachel@csse.uwa.edu.au (R. Cardell-Oliver).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.01.009

detection is vital in order to detect objects entering or leaving the monitoring region [5,7–9].

During the network's lifetime, nodes on the boundary are required to keep sensing at all times to detect objects that enter the monitoring region, while other nodes sleep to conserve energy. With the assumption that the objects move from outside into the monitoring area and there are no sensing gaps between two neighboring nodes on the boundary, objects can be detected by one or more boundary nodes when they cross the boundary. When an object is detected, a detection message is broadcast by a cluster head to its neighboring nodes to continue tracking the object. The cluster head is elected among nodes that can sense the object at that time and has the highest remaining energy. Any nodes, which receive the detection message and can sense the object, elect among themselves a new cluster head to broadcast the next detection message. This monitoring task of "sense-communicate-sense" is carried out by sensor nodes along the object's track until the object leaves the region.

We illustrate an overview of a WSN for tracking a moving object in Fig. 1a and the corresponding graph model for the WSN in Fig. 1b. A sensor node corresponds to a vertex in the graph and a connectivity relation between nodes corresponds to an edge between the corresponding vertices in the graph. A *cycle* in a graph is a closed curve of edges that consists of m distinct vertices with $m \geqslant 3$. A *boundary cycle* of a WSN is the outer cycle of the network which separates the monitoring region from the outside world. *Boundary nodes* are sensor nodes which lie on the boundary cycle and *non-boundary nodes* are sensor nodes which are not on the boundary.

The problem of detecting boundary nodes is simplified if the exact location of each sensor node is known. Unfortunately, building a large scale WSN with special location hardware such as GPS embedded in the nodes is not practical [16] because it is neither cost effective nor energy efficient. The price of GPS is expensive compared to the nodes themselves and it consumes a considerable amount of nodes' energy that will lead to short lifetime networks. Furthermore, GPS cannot function well in a closed place where the microwave signal from the satellites is blocked by obstacles. In addition, the size of GPS can change the structure of the nodes that are required to be small by many applications.

In this paper, we propose a novel heuristic algorithm to find boundary cycles of randomly deployed WSNs. We have three motivations behind our boundary detection algorithm, which become the contributions of this paper. Firstly, the aim is to design a fault-tolerant algorithm that does not require location information. Secondly, the algorithm must be decentralized that does not rely on one central node. So, it can perform local computation to tolerate faults. Thirdly and the most important motivation is to design an algorithm that works even when the average degree is low (average degree 5–6). This is definitely the case compared to other algorithms in the literature [13,8,11,12,5,18] that assume high average degree. We choose low-degree networks with average degree 5–6, because it is possible to construct data gathering trees in such networks for sensor reporting. Our boundary detection algorithm will be practical to use in location-free and low-degree WSNs as most practical networks are of this kind.

We find boundary cycle where two neighboring boundary nodes are within communication range of each other. So, any processes that need boundary nodes to exchange information along the cycle can do so. In this algorithm, we only assume each node can communicate directly with its neighbors and has a mechanism to estimate distances to nearby nodes, but not necessarily the true distances. We do not assume any geographical information as in [6], our assumption on average degree is not as high as the average degree assumed in [13,8,11,12,5,18] and we do not assume uniform node distribution. Moreover, compared to the related work in [13], our work improves on sparse networks because the algorithm in [13] fails for low density networks.

We develop the key ideas of our boundary detection algorithm in the centralized scenario. Then, we extend these ideas to the distributed scenario, which is more realistic for running in real networks. We consider the need for

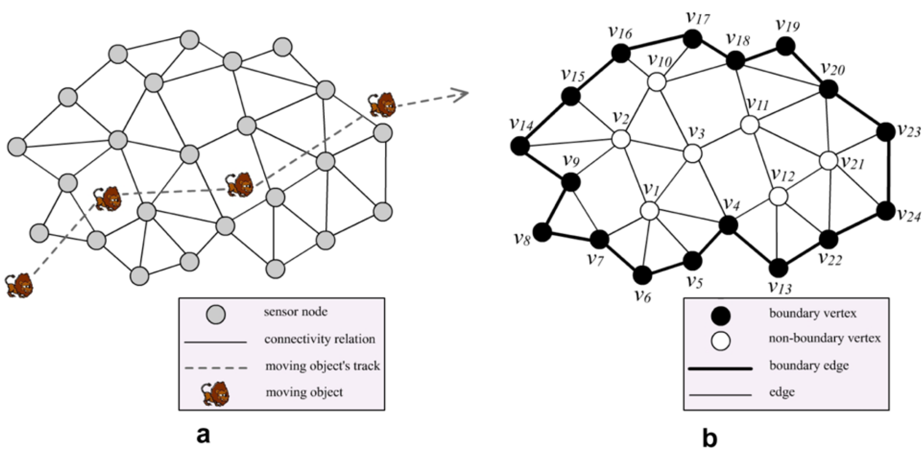

Fig. 1. (a) An overview of a WSN and (b) the corresponding graph model for the WSN.