Computer Networks 70 (2014) 96–112

Contents lists available at ScienceDirect

Computer Networks

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

Review Article

Location-free boundary detection in mobile wireless sensor networks with a distributed approach

Wei-Cheng Chu [a,*], Kuo-Feng Ssu [b,c]

[a] Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan
[b] Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan
[c] Department of Computer and Information Sciences, University of Delaware, Newark, DE, USA

ARTICLE INFO

Article history:
Received 24 October 2013
Received in revised form 21 April 2014
Accepted 12 May 2014
Available online 22 May 2014

Keywords:
Wireless sensor network
Boundary detection
Location-free

ABSTRACT

Location-free boundary detection is an important issue in wireless sensor networks (WSNs). Detecting and locating boundaries have a great relevance for network services, such as routing protocol and coverage verification. Previous designs, which adopt topology-based approaches to recognizing obstacles or network boundaries, do not consider the environment with mobile sensor nodes. When a network topology changes, a topology-based approach has to reconstruct all boundaries. This study develops a distributed boundary detection (DBD) algorithm for identifying the boundaries of obstacles and networks. Each node only requires the information of its three-hop neighbors. Other information (e.g., node locations) is not needed. A node with DBD can determine whether itself is a boundary node by a distributed manner. The DBD approach further identifies the outer boundary of a network. Performance evaluation demonstrates that DBD can detect boundaries accurately in both static and mobile environments. This study also includes experiments to show that DBD is applicable in a real sensor network.

1. Introduction

Boundaries can be classified into two types: outer and inner boundaries. An outer boundary is the network boundary. An inner boundary is an obstacle boundary in a network. The obstacle is a geographic region without the functionality of sensing or communication. The obstacle could be induced by node failures or a non-uniform deployment of sensor nodes. The obstacle may include a physical obstacle that blocks the propagation of communication signals [1]. Obstacle existence could degrade the performance of sensing applications, such as environment monitoring, and target tracking. Some studies have investigated the dead-end problem in geographic routing caused by obstacle existence [2,3], and they require the knowledge of all obstacle shapes to identify concave areas. Even though the dead-end problem has been solved, delivering data along the boundaries of obstacles yields an unbalanced traffic load and congestion. The unbalanced traffic load further increases the sizes and number of obstacles. To enhance the load balance, several routing protocols construct multiple paths around all inner boundaries [4–6]. Obstacles may also increase the length of a route [7]. Therefore, obstacle detecting and healing mechanisms have been developed to improve network connectivity [8,9]. Identifying all obstacle boundaries is important for all above applications because their performances depend on the recognized shapes of obstacles (e.g., the concave area [3]).

The literature contains several algorithms for detecting communication holes in WSNs [7,10–12]. The algorithms supposed that each node is aware of its own location with high accuracy, so these studies can locate obstacles by

* Corresponding author. Tel.: +886 6 2374532.
E-mail addresses: 491191291@s91.tku.edu.tw (W.-C. Chu), ssu@ee.ncku.edu.tw (K.-F. Ssu).

http://dx.doi.org/10.1016/j.comnet.2014.05.005
1389-1286/

nodes using a coordinate computing approach. Stereovision systems are proposed based on image-processing algorithms for physical obstacle detections [13–16]. The systems enable a robot to navigate a path between two pre-determined end points to avoid obstacles. Signal processing is another approach to detecting physical obstacles [17]. Each node uses additional equipment to compute received signals to depict the shapes of physical obstacles. Some researches used location information of nodes to recognize obstacle boundaries [18]. However, the detection algorithms based on the location information are not preferable because either GPS or other localization mechanisms cannot guarantee the needed accuracy [19]. Location-free approaches thus have become increasingly common [20–31].

Existing location-free approaches required a global or partial topology to construct boundaries. A node cannot determine whether it is a boundary node by itself. In other words, a node is unaware of its erroneous node role caused by a topology change (e.g., a change of its neighbor set). To maintain the correct role of each node, topology-based approaches have to frequently reconstruct all boundaries. The frequent reconstruction produces enormous cost wastes. These approaches are not applicable in a mobile node environment.

Previous design developed a fine-grained boundary recognition method (TTG) to detect all holes [31]. Each hole can be enclosed by exactly one cycle. The TTG approach can also detect the outer boundary of a network. The outer boundary determination is useful for some applications (e.g., the coverage issue [32]). However, TTG did not address the issue of mobile nodes. The solution of the outer boundary determination is not complete, either. For example, in Fig. 1, the length of the outer boundary cycle *abcde f g* is smaller than the length of inner boundary cycle *abcd jihg*. Cycle *abcd jihg* is then erroneously considered as the outer boundary. Chu and Ssu developed a mechanism to construct boundaries locally in mobile sensor networks [33]. The mechanism produces excessive boundary nodes that could affect the performance of related applications (e.g., routing protocols [7,10]).

Dong et al. indicated that a boundary has two properties, *continuity* and *consistency* [31]. Continuity means that all the nodes on a boundary can be connected by themselves and can form a loop. Consistency ensures that a boundary encircles its corresponding obstacle in any graph embedding. Saukh et al. argued that a boundary has various properties: *continuity* and *uniqueness* [28]. Uniqueness indicates that a boundary represents for exactly one polygon in a given embedding of graph *G*. However, the uniqueness is hardly useful in a location-free environment. Fig. 2 shows an example of two unique boundaries for a given embedding of Graph *G* (the thick black line). Vertex *a* is not a real node so it cannot be identified in a location-free network. Therefore, *approximate uniqueness*, described in Section 3.2, is used in this study.

This study introduces a distributed boundary detection (DBD) to accomplish the detection of inner and outer boundaries. The DBD scheme does not need location information or knowledge of the distance among nodes. Additionally, DBD does not enforce the Unit Disk Graph (UDG) constraint. The DBD mechanism reconstructs the boundaries locally and rapidly in mobile environments. Each node collects the three-hop neighboring information to construct its own contours and three-hop neighbor graph. A contour is broken if it cannot enclose the node. A broken contour is caused by encountering an obstacle. The node determines whether it is nearby the obstacle by constructing its contour. If there exists no broken contour, the node uses its own three-hop neighbor graph to check the existence of obstacle again. This study develops a pruning rule, which is similar to the vertex deletion of skeleton extraction phase [31], to reduce the number of boundary nodes. To the best of our knowledge, TTG [31] is the only one approach which guarantees the boundary detection for all obstacles in a static and location-free environment. However, TTG generated excessive flooding processes for boundary detection. In contrast, DBD not only guarantees the correctness but also reduces the control overhead of boundary detection. Thus, DBD is more suitable for a mobile environment. Based on our simulation and experiments, DBD outperforms the previous topology-based boundary detection methods for mobile wireless sensor networks. The contributions of this study are listed as follows:

- **Identifying boundary nodes with low control overhead.** Each node only requires its own three-hop neighboring information to identify whether itself is a boundary node. A global topology information is not needed. When a network topology has changed because of node movements, a node can examine its role by itself again. Therefore, boundaries can be corrected locally without reconstructing all boundaries.
- **Proving the correctness of DBD.** This study provides the correctness proof of DBD that each constructed boundary encloses exactly one obstacle. All obstacles in the network can be detected by DBD. Each boundary follows three properties: continuity, consistency, and approximate uniqueness. The proof also confirms that DBD is a distributed method.

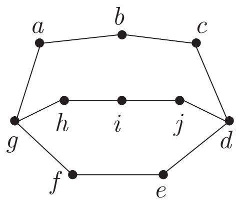

Fig. 1. The length of outer boundary cycle is smaller than the length of one of inner boundary cycles.

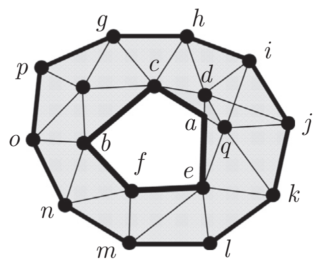

Fig. 2. Real boundaries and obstacle cycles.