Computer Communications 45 (2014) 32–46

Contents lists available at ScienceDirect

Computer Communications

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

A novel localization algorithm for large scale wireless sensor networks

Haidar Safa

American University of Beirut, Department of Computer Science, P.O. Box: 11-0236, Riad El-solh, Beirut 1107 2020, Lebanon

ARTICLE INFO

Article history:
Received 31 May 2013
Received in revised form 13 January 2014
Accepted 17 March 2014
Available online 27 March 2014

Keywords:
Wireless sensor networks
Tracking
WSN localization and monitoring
DV-HOP
Routing

ABSTRACT

Localization has been a major challenge in wireless sensor networks. The data reported from a sensor is only useful when the position of that sensor is found. In this paper, we propose a new algorithm that accurately localizes sensors while minimizing their power consumption and memory requirements. The proposed algorithm splits the task of localization between sensor nodes and the base station (i.e., sink) and does not necessitate the presence of many anchor nodes (nodes with preconfigured positions). It recommends placing the anchors in a circle or a semi-circle around the perimeter of the WSN. Not only this placement strategy leads to more accurate localization, but it is also very convenient for WSNs used for environment monitoring, military surveillance, or tracking applications. The proposed approach can also be tailored to route the sensed data to the base station which alleviates the sensors from the overhead incurred in establishing routes to the base station. The performance of the proposed approach was evaluated and compared to other peer algorithms using the network simulator (NS-2). Results show that significant enhancement is obtained with the proposed algorithm when measuring metrics such as energy and localization error while varying other simulation parameters such as the number of sensors and the area size.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Wireless Sensor Networks (WSNs) are currently used to monitor a wide range of military, environmental, civil, and health-care applications. A WSN is composed of a collection of sensor nodes deployed in a sensor field, each of which collects data and relays them to the base station (also called a sink node) where data can be best-analyzed and used [2]. A sensor node consists, in general, of a sensing unit, a processing unit, a transceiver, and a power unit. Its task may be summarized by three key terms: *sensing*, *computing* and *communicating*. There is a variety of sensor nodes [8] such as: (1) the *acoustic sensors* that use the sounds as a sensing medium and can be used in air or underwater; (2) the *thermal sensors* that are used to measure temperature changes at homes and industries; (3) the *magnetic sensors that* are used to detect metallic objects and other magnetic flux without requiring any physical contact with the sensed phenomenon; (4) the *mechanical sensors* that require direct physical contact with the phenomenon and can be used to detect changes in the resistance in electronic circuits; and (5) the *biosensors* which have a biochemical component that witnesses bio-molecular changes at its proximity and then converts these changes into detectable optical signals.

Finding the location of the sensor where the event occurred is an intrinsic and integral part of any WSN application and represents a major challenge because without finding the position of the sensor that is reporting the sensed data, the latter will not be useful. An approach to find the location of a reporting sensor is to equip it with a built-in GPS receiver. However, this method is totally not suitable for WSNs because GPS receivers are expensive and power-consuming. Other approaches, referred to as collaborative localization algorithms, assume that only a small fraction of sensors have their absolute positions either through manual configuration or using GPS receivers [25,26]. These sensors are called anchors, and their positions can be used as references to estimate the positions of reporting sensors as shown in the deployed WSN of Fig. 1. The remaining of this paper is organized as follows. In Section 2, we survey some of the existing localization techniques in WSNs. In Section 3, we present the proposed algorithm which is an extension to our work in [19]. The proposed localization approach encompasses a routing algorithm; hence there is no need to employ an additional routing algorithm. In Section 4, we evaluate the performance of the proposed algorithm and compare it to other algorithms found in the literature. Finally, Section 5 concludes this paper.

2. Background and related work

WSN localization algorithms can be divided into two categories: range-based algorithms and range-free algorithms [7,21]. Range-based algorithms depend on measuring physical attributes of the

---

E-mail address: haidar.safa@aub.edu.lb

http://dx.doi.org/10.1016/j.comcom.2014.03.020
0140-3664/© 2014 Elsevier B.V. All rights reserved.

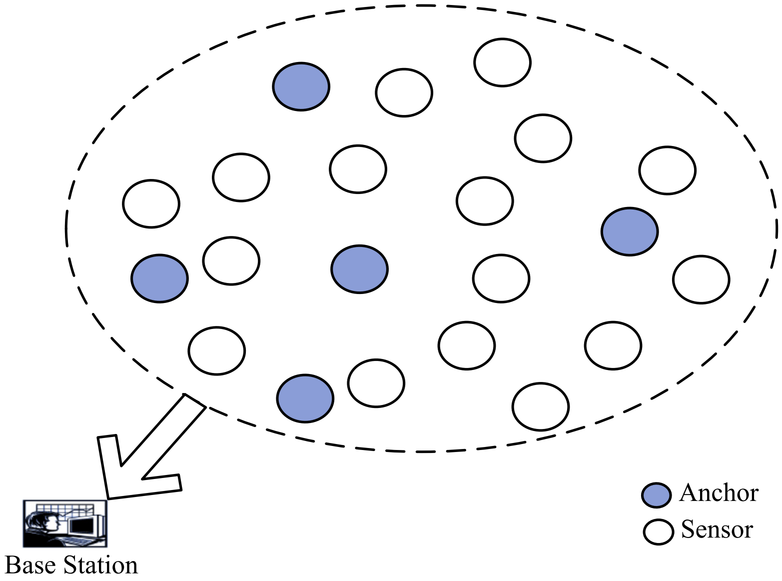

Fig. 1. An example of a deployed WSN used in collaborative localization algorithms.

wireless signals transmitted between antennas [5] such as the received signal strength indicator (RSSI), the time of arrival (ToA) and the angle of arrival (AoA) of the signal. These algorithms, however, require adding extra hardware to sensors; which is expensive and power-consuming. In this paper, we focus on range-free algorithms which can be centralized or distributed. In centralized range-free localization algorithms, each sensor relays information about itself to a base station [29]. The base station uses this information to build a map of the network to estimate the sensor position. These centralized algorithms suffer from communication overhead between the base station and sensors. In distributed range free localization algorithms each sensor localizes itself locally as described next.

2.1. DV-Hop localization algorithm

The Distance Vector by Hop counting (DV-Hop) is the most known distributed algorithm. It was proposed in [15] as an ad hoc positioning system (APS) in which sensors exchange distance vectors that contain *hop counters* that signify the number of hops between the sensor receiving this vector and the sending anchor. DV-Hop assumes that every anchor i has a hop counter, $HopCount_i$, and every sensor in the network stores the hop counter corresponding to every anchor; the value of the $HopCount_i$ that a sensor stores about anchor i represents the minimum number of wireless hops between that sensor and the anchor i. DV-Hop consists of three phases as shown in Fig. 2. In the first phase, an anchor i floods a message containing its ID, coordinates, and the variable $HopCount_i$ initialized to zero. Sensors store and exchange anchor's hop counters. Indeed, every time a sensor receives a message containing a $HopCount_i$ corresponding to an anchor i, it checks the value of the $HopCount_i$ that it maintains about anchor i. If this value is less than the received one, then the latter is ignored; otherwise the receiving sensor increments the value of the received *HopCount*, updates its stored $HopCount_i$, then floods it in the network.

In the second phase, anchor i computes the average hop length from its perspective, $AvgHopLength_i$, using Eq. (1) which is given as:

$$AvgHopLength_i = \frac{\sum_{j=1, j\neq i}^{M} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j=1, j\neq i}^{M} HopCount_{i,j}} \quad (1)$$

where M is the number of anchors in the network, j identifies other anchors, $HopCount_{i,j}$ is the distance in hops between anchor i and anchor j, (x_i, y_i) and (x_j, y_j) represent the coordinates of anchors i and j, respectively.

After computing $AvgHopLength_i$, anchor i floods it in the network for other anchors and sensors. A sensor maintains only the average hop length flooded by the closest anchor to it. Fig. 3 shows an example where each of the three anchors A1, A2 and A3 calculates its average hop length. Then sensor node S with unknown position maintains the one that was received from A2 (i.e., the closest to S). S uses the received $AvgHopLength_i$ to compute the distance d_j between S and every anchor j using Eq. (2) which is given as:

$$d_j = AvgHopLength_i \times HopCount_j \quad (2)$$

where, $HopCount_j$ is the hop counter that S maintains for anchor j and $AvgHopLength_i$ is the average hop length that sensor S obtained from the closest anchor, say i.

In Fig. 3, anchor A1 is 40 meters and 2 hops away from anchor A2, and 100 m and 6 hops away from anchor A3. A1 computes the $AvgHopLength_1$ as per Eq. (1). Hence, $AvgHopLength_1$ is equal to $(100 + 40)/(6 + 2) = 17.5$ m. Similarly, A2 and A3 compute their average hop length, $AvgHopLength_2$ and $AvgHopLength_3$, as $(40 + 75)/(2 + 5) = 16.42$ m and $(75 + 100)/(6 + 5) = 15.90$ m, respectively. Then, each anchor floods its *AvgHopLength* in the network, so other anchors and sensors receive it. A sensor S will maintain only the *AvgHopLength* flooded by the closest anchor to S. When sensor node S gets the $AvgHopLength_2$, it uses Eq. (2) to estimate its distance away from A1, A2, and A3 which are respectively 49.26 m (3×16.42), 32.84 m (2×16.42), and 49.26 m (3×16.42).

In the third phase, a sensor uses the Least Square (LS) technique to trilaterate its position [10]. In basic trilateration, only three distances from anchors are necessary for an unknown sensor to find its location. However, the more distances available leads to better localization accuracy. Fig. 4 illustrates the circle intersection of basic trilateration. Numerically, trilateration is done by solving a system of equations. The LS technique is used in numerical optimization in curve fitting when we seek to find the parameters that best match an existing parameterized model to our experimental data. Our aim is to minimize the objective function which is defined as the sum of square of errors between the calculated values of the model and the measured values of the system. For

Download English Version:

https://daneshyari.com/en/article/448693

Download Persian Version:

https://daneshyari.com/article/448693

Daneshyari.com