Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 29 (2006) 2494–2505
computer communications
www.elsevier.com/locate/comcom

Distributed and energy-efficient target localization and tracking in wireless sensor networks

Jeongkeun Lee *, Kideok Cho, Seungjae Lee, Taekyoung Kwon, Yanghee Choi

School of Computer Science and Engineering, Seoul National University, San 56-1 Shilim-dong, Kwanak-gu, Seoul, Republic of Korea

Available online 3 March 2006

Abstract

In this paper, we propose and evaluate a distributed, energy-efficient, light-weight framework for target localization and tracking in wireless sensor networks. Since radio communication is the most energy-consuming operation, this framework aims to reduce the number of messages and the number of message collisions, while providing refined accuracy.

The key element of the framework is a novel localization algorithm, called Ratiometric Vector Iteration (RVI). RVI is based on distance ratio estimates rather than absolute distance estimates which are often impossible to calculate. By iteratively updating the estimated location using the distance ratio, RVI localizes the target accurately with only three sensors' participation.

After localization, the location of the target is reported to the subscriber. If the target is stationary or moves around within a small area, it is wasteful to report (almost) the same location estimates repeatedly. We, therefore, propose to dynamically adjust a reporting frequency considering the target's movement so that we can reduce the number of report messages while maintaining tracking quality. Extensive simulation results show that the proposed framework combining RVI and the movement-adaptive report scheduling algorithm reduces the localization error and total number of the transmitted messages up to half of those of the existing approaches.

Keywords: Target localization; Tracking; Ratiometric; Energy-efficient; Sensor network

1. Introduction

Wireless sensor networks are systems of small, low-powered networked sensing devices deployed over an interested area to monitor interested events and perform application-specific tasks in response to the detected events. One of the most significant and elementary application is localization and tracking moving targets. The type of interested signals includes temperature, sound, light, magnetism and seismic vibration: a sensing modality is determined based on the types of targets to be tracked.

Regardless of the various types of targets and tracking environments, there are four common procedures involved in distributed (or decentralized) target tracking applications: First, *sensors should be localized* prior to participating in target tracking tasks. The importance of sensor nodes's location information have been emphasized in the literature and many sensor network localization techniques, e.g., [8,9,6,10–13] can be used. We assume that each node is aware of its location and consider only the other procedures in the below.

Second, *target localization* is required. Here, if the target is able to communicate with sensors (e.g., a householder in the intelligent home network), the localization problem gets easier and the same localization technique with sensor network localization can be used. In many cases, however, targets are not cooperative with sensors, e.g., enemy vehicles and unregistered victims in disaster areas. Most of the previous localization algorithms use absolute point-to-point distance estimates. If the target is cooperative with the sensor network, it is possible to know the original signal (interested signal) strength at the target source as a pre-defined parameter or through communication. In the non-cooperative cases, however, the absence of the original signal strength information prevents the use of absolute

* Corresponding author.
E-mail address: jklee@mmlab.snu.ac.kr (J. Lee).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.02.004

distance estimates. Instead, one can estimate the original signal strength by collecting and analyzing a number of sensing data, which often requires non-linear optimization techniques [16,19]. Because of the high computation and communication overhead, however, those techniques are not suitable for low-cost sensors in distributed environments. In order to tackle this problem, we present a light-weight localization algorithm, dubbed *Ratiometric Vector Iteration (RVI)* that is based on relative distance ratio estimates rather than absolute distance estimates. RVI not only achieves high accuracy, but also enables a distributed operation in the low-cost sensors.

Third, collaborative data processing among nodes is desirable because sensing information collected from different sensors may be redundant and contribute to the localization result with different importance. Thus, we need a *sensor collaboration* algorithm considering sensory data as well as constraints on energy consumption, latency and other costs. Therefore, the leader is often selected to manage collaborative data processing and sensor grouping. The leader also maintains the target's state such as location, speed and direction. Because the target is moving, the state information has to be forwarded along with the moving target: *leader selection* is required. Most previous solutions use explicit group management (for sensor selection) and leader selection approaches which incur the control message overhead and/or assume hierarchical node-cluster deployment. In our framework, the leader selection and group management are accomplished implicitly without any additional message overhead. With the help of received-signal-strength(RSS)-based backoff timer, we ensure that only a sufficient number (3 in our case) of sensors broadcast messages to determine the target location with a low collision probability.

Fourth, the tracking system (in particular, the leader) should *report* the location of the target to a sink node in a timely manner. Usually, the sink is a gateway connecting the sensor network to the subscriber. Therefore, reporting to the sink normally requires multi-hop message relaying whose message transmission overhead is proportional to the hop distance between the leader and the sink. We propose a movement-adaptive report scheduling algorithm that reduces the report message overhead, while keeping substantial tracking quality at the subscriber side.

Because of the low cost requirement and small-form factor design of sensor nodes, the resources available to individual nodes are highly limited. Although the limitations of processor bandwidth and small memory are expected to weaken with a development of fabrication techniques, the energy constraint is likely to last for decades. Typical sensor nodes are powered by small batteries that are difficult to replace even if not impossible considering the slow progress in battery capacity [1]. In wireless sensor networks, in particular, communication is the most energy-consuming operation, with each bit transmission costing as much energy as about 1000 instructions processing [2]. With this in mind, the goals of this paper and the proposed solutions are given as follows:

- achieving good localization quality with low communication cost: Ratiometric Vector Iteration
- minimizing the number of collisions among messages: RSS-based backoff timer
- reducing the number of transmitted messages: movement-adaptive report scheduling.

We begin with Section 2 which gives related work. In Section 3, we introduce a novel localization algorithm, RVI, and show its advantages. In Section 4, we give some assumptions and describe the target tracking cycle composed of three steps. The message collision probability is analyzed in Section 5 and we present simulation results in Section 6. Section 7 concludes this paper.

2. Related work

Localization: Many existing localization algorithms for wireless sensor networks employ a centralized operation and/or use an absolute distance estimates. A large volume of sensing data are gathered from sensors and processed in a centralized manner. The gathered data are applied to multidimensional scaling [8], convex optimization [9], maximum likelihood testing [6], and so on. High computational overhead, however, prevents the use of those centralized localization methods in distributed, low-cost, low-power sensor networks. Moreover, the centralized method requires all sensory data to be delivered to the high-end node, which incurs high communication overhead.

Absolute point-to-point distances estimated from RSS or time-of-arrival (TOA) or time-difference-of-arrival (TDOA) information are often used for localization. Cricket [10] is the well-known solution based on absolute distances estimated from TDOA between a radio signal and an ultra sound signal. Some distributed localization algorithms [11,12] do not use the absolute distance estimates but require communication between the target and sensors, which is impractical in non-cooperative scenarios assumed in this paper. Centroid [13] is a distributed solution which is not dependent on absolute distance estimates. Nodes localize themselves to the centroid of reference points (target-detecting sensors in this paper) considering poor distance estimates from the RSS. The proposed Weighted Centroid algorithm in this paper is an advanced version of Centroid.

All of the above solutions are suitable for localizing sensor network itself or cooperative targets. For the non-cooperative target localization, several solutions [16,19] have been proposed in the literature. As will be discussed in Subsection 3.2, the previous solutions have some limitations of high computation overhead and/or hierarchical sensor deployment.

Target tracking: Collaborative data processing and in-network processing have been extensively studied to reduce redundant communication in target tracking applications. Clustering or group-based techniques such as *information-driven sensor query (IDSQ)* [14] and *dynamic convoy tree-based collaboration (DCTC)* [15] are well-known solu-

tions for collaborative data processing. They explicitly select and manage sensors that participate in target tracking. However, the group management and the hierarchical structure maintenance overhead are not negligible and eventually incur additional expenditure of energy for computation and communication. Our approach performs collaborative target tracking implicitly with small message overhead through RVI and RSS-based backoff timer that guarantees low collision probability. A dynamic clustering and Voronoi diagram-based target tracking framework is proposed in [16] for acoustic target tracking. Pre-construction of neighborhood Voronoi diagram and the use of two-phase random backoff timer enable efficient cluster-head and sensor selections. However, it requires a designated powerful cluster heads' deployment and the localization accuracy depends on the nature of the approximation technique's poor accuracy. The proposed approach of this paper supports a fully distributed operation while rendering good accuracy.

Some algorithms such as [17,18] predict the future target position based on the assumption of a linear target trajectory. Such prediction-based approaches, however, are not robust when the prediction is wrong. Our tracking framework does not predict or maintain target history but keeps only the most recently reported target location and the reporting time instance. Also proposes an adaptive protocol that controls the frequency of localization based on the velocity of moving sensors [18]. The idea of reducing message overhead by adapting to the sensor movement is similar to the proposed report scheduling algorithm of this paper. However, [18] considers the localization of moving *sensors* while we deal with the tracking of a moving *target* in the localized sensor network. We also discuss the additional use of report messages for local time synchronization.

3. Target localization

3.1. Sensing model

We use a conventional received-signal-strength (RSS) based sensing model in which RSS decreases exponentially with the propagation distance:

$$r_i = \frac{a}{|X - S_i|^{\alpha}} + n_i, \ 1 \leqslant i \leqslant N, \tag{1}$$

where r_i is the sensed RSS value in the ith sensor, a is the original signal strength at the target source, X is the target location in two-dimensional coordinate system, S_i is the location of ith sensor, $|X - S_i|$ means the Euclidean distance between X and S_i, α is the pathloss exponent and n_i is the White Gaussian noise with $\mathcal{N}(0, \sigma)$. N is the number of sensors around the target. Again, X and S_i are elements of $\mathbf{R}^2$ in this paper. We assume that r_i, S_i, and α are known values, while a and X are unknown. Since a is unknown, it is difficult to estimate an absolute point-to-point distance.

We can use a relative distance ratio, r_i/r_j, $(i \neq j)$. By disregarding the noise, the ratio between RSS in the ith sensor and jth sensor, $r_i^{-1/\alpha}/r_j^{-1/\alpha}$ represents the ratio of the target-to-ith-sensor distance to the target-to-jth-sensor distance as follows:

$$r_i^{-1/\alpha} : r_j^{-1/\alpha} = |S_i - X| : |S_j - X|, \tag{2}$$

where $i \neq j$, $1 \leqslant i, j \leqslant N$.

Before describing the proposed RVI algorithm, we first introduce two existing approaches for the purpose of comparison.

3.2. Previous approaches

Approximation: The location of a sensor with the highest received-signal-strength can be used as the *approximate* location of the target [16]. This is the simplest method, but its localization accuracy is poor: the localization error is about a half of the inter-sensor distance. In [16], a Voronoi diagram is exploited to bound the error of the above approximation in the presence of sensing noise. This voronoi-diagram based approach, however, has some limitations. First, the noise still affects the approximation error when the Voronoi diagram is constructed. Second, this approach requires a static backbone of hierarchically placed high-capability sensors. Third, because a measured RSS of a sensor should be compared with that of all Voronoi neighbors of the sensor, the number of participating sensors (Voronoi neighbors) may increase overwhelmingly as the sensor node density increases, especially in a random sensor distribution.

Nonlinear optimization: Given a pair of sensing data (r_i, r_j) and the locations of both sensors (S_i, S_j), the locus of the potential location of the target X can be shown to be an Apollonius circle: the set of all points whose distances from two fixed points (S_i,S_j) are in a constant ratio $|X - S_i| : |X - S_j|$. For noisy measurements, the unknown target location X is estimated by solving a nonlinear least square problem of the form [16,19]:

$$X = \arg \ \min \sum_{k=1}^{m} \frac{|X - O_k|^2 - p_k^2}{p_k^2},$$

where m is the number of given ratios, O_k is the center coordinates, and p_k is the radius of the Apollonius circle associated with the kth ratio. Newton's method [20] can be used to solve the nonlinear least square optimization problem. This sophisticated method promises good localization results at the expense of high computational complexity: it is not a suitable solution for low-cost sensors.

3.3. Proposed algorithm description

We describe a novel localization algorithm using a distance ratio, Ratiometric Vector Iteration (RVI). Before describing RVI, we first introduce the weighted centroid technique which will be used to obtain an initial guess close to the true target location for RVI's iterative estimations.

Weighted Centroid: Given locations of $k(k \geqslant 3)$ sensors $(S_1,\ldots,S_k)$ and sensing data $(r_1,\ldots,r_k)$ from those sensors, the location of target in a two-dimensional Euclidean plane can be estimated as a weighted centroid of sensor's locations where each weight $(w_1,\ldots,w_k)$ on each sensor's location is characterized by

$$\begin{aligned} & w_1 : w_2 : \ldots : w_k \\ &= \frac{1}{|S_1 - X|^\beta} : \frac{1}{|S_2 - X|^\beta} : \ldots : \frac{1}{|S_k - X|^\beta}, \\ &= r_1^{\beta/\alpha} : r_2^{\beta/\alpha} : \ldots : r_k^{\beta/\alpha}, \end{aligned}$$

where β determines a weight function. That is, $\beta = 1$ means that the weight is an inverse of distance and $\beta = 2$ implies a square inverse function. Note that, if $\alpha = \beta$, sensing data can be used as weights without additional algebraic operation, that is $w_i = r_i$. The target location estimate by Weighted Centroid, X_{WC}, is given by

$$X_{\mathrm{WC}} = \frac{\sum_{i=1}^{k} w_i S_i}{\sum_{i=1}^{k} w_i}.$$

We call this algorithm as a *weighted centroid*.

Ratiometric Vector Iteration: In Fig. 1, if X and X_j represent the true location of the target and the estimated location at the jth iteration respectively, we can move X_j "toward" X such that the difference between a distance ratio $|\overrightarrow{S_1X}| : |\overrightarrow{S_2X}| : |\overrightarrow{S_3X}| = 2:1:2$ and a distance ratio $|\overrightarrow{S_1X_j}| : |\overrightarrow{S_2X_j}| : |\overrightarrow{S_3X_j}| = 1:3:3$ decreases. Note that $|\overrightarrow{S_iX_j}| = |S_i - X_j|$. Vectors $\overrightarrow{S_iX_j}$ are multiplied by the ratio difference and summed up to compose a vector $\overrightarrow{V_j}$ which is eventually added to the X_j. This vector translation of X_j by $\overrightarrow{V_j}$ is repeated until X_j is sufficiently close to the true location X.

As inputs to the RVI algorithm, locations of $k(k \geqslant 3)$ sensors $(S_1,S_2,\ldots,S_k)$ and sensing data $(r_1,r_2,\ldots,r_k)$ from those sensors are given. Then, the distance ratio is represented by using sensing values $r_i^{-1/\alpha}$, and we normalize it by the sum $\sum_{i=1}^{k} r_i^{-1/\alpha}$ for the purpose of comparison:

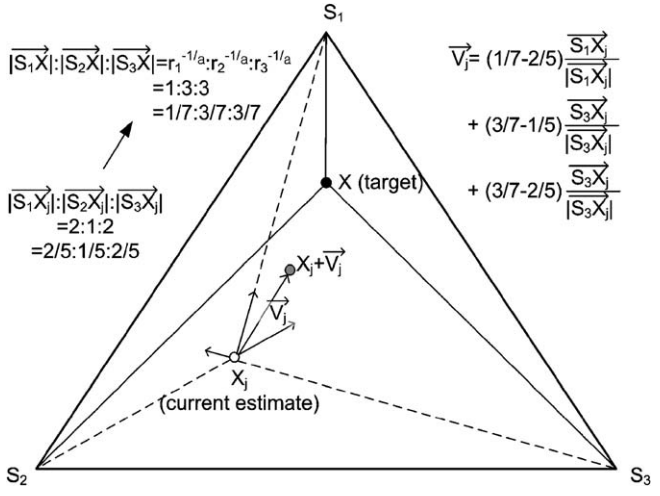

Fig. 1. Ratiometric Vector Iteration.

$$\begin{aligned} |\overrightarrow{S_1X}| : \ldots : |\overrightarrow{S_kX}| &= r_1^{-1/\alpha} : \ldots : r_k^{-1/\alpha} \\ &= g_1 : \ldots : g_k \end{aligned}$$

where $g_i = \dfrac{r_i^{-1/\alpha}}{\sum_{i=1}^{k} r_i^{-1/\alpha}}$.

The goal of RVI is to update X_j such that the difference between $|\overrightarrow{S_1X_j}| : \ldots : |\overrightarrow{S_kX_j}|$ and $g_1 : \ldots : g_k$ decreases. This translation of X_j is iterated until the direction of V_j drastically changes or the normalized moving distance of X_j becomes smaller than a pre-determined threshold size C_{th}. A generalized algorithm description is given as follows:

0. *Initialization*: The target location estimate at the jth iteration is denoted by X_j. The location estimated by weighted centroid, X_{WC} is used as the starting point for the first iteration, i.e., $X_0 = X_{\mathrm{WC}}$. The moving vector is initialized as zero, i.e., $V_0 = \vec{0}$. Set the iteration index $j = 0$.
1. *Normalization*: For comparison with $g_1 : \ldots : g_k$, the ratio $|\overrightarrow{S_1X_j}| : \ldots : |\overrightarrow{S_kX_j}|$ is also normalized by the sum $\sum_{i=1}^{k} |\overrightarrow{S_iX_j}|$ and is given by

   $$|\overrightarrow{S_1X_j}| : \ldots : |\overrightarrow{S_kX_j}| = g_{1,j} : \ldots : g_{k,j},$$

   where $g_{i,j} = \dfrac{|\overrightarrow{S_iX_j}|}{\sum_{i=1}^{k} |\overrightarrow{S_iX_j}|}$.
2. *RVI Move*: The move vector $\overrightarrow{V_j}$ is given by

   $$\overrightarrow{V_j} = \sum_{i=1}^{k} (g_i - g_{i,j}) \frac{\overrightarrow{S_iX_j}}{|\overrightarrow{S_iX_j}|}.$$

   Each element vector $\overrightarrow{S_iX_j}$ is normalized by $|\overrightarrow{S_iX_j}|$ because the vector only indicates the direction and the vector size is determined by the difference between ratios $(g_i - g_{i,j})$.
3. *New Estimate*: Update

   $$X_{j+1} = X_j + \overrightarrow{V_j}. \tag{3}$$

4. *Stationary Point*: The algorithm terminates if

   $$\overrightarrow{V_{j-1}} \cdot \overrightarrow{V_j} < 0$$

   or

   $$\frac{|X_{j+1} - X_j|}{D_{\mathrm{inter\_sensor}}} < C_{\mathrm{th}}$$

   where $D_{\mathrm{inter\_sensor}}$ denotes the pre-determined average inter-sensor distance, and C_{th} is the threshold for a terminating condition. In other words, the algorithm stops when the direction of $\overrightarrow{V_j}$ drastically changes from the direction of the previous iteration vector $\overrightarrow{V_{j-1}}$, or the moving distance at the jth iteration (normalized by the average inter-sensor distance) becomes smaller than C_{th}. The condition $\overrightarrow{V_{j-1}} \cdot \overrightarrow{V_j} < 0$ prevents infinite pingpong repetition around the target.On these termination conditions, RVI stops and returns the target location estimate as the algorithm output:

$X_{RV} = X_{j+1}$.

Otherwise, the index j is incremented, and the algorithm goes to step 1.

Fig. 2 shows the six results of our test program with three sensors ($k = 3$) which forms a regular triangle assuming noiseless sensing. An arrow indicates the true target location and small points represent a convergence of target estimates during iterations. An eyeball-like circle at the beginning indicates the starting point X_{WC}. When the target resides inside the triangle (convex area) formed by three sensors, the RVI algorithm accurately estimates the target location as shown by the case A. Moreover, the case C shows that the target location estimate is accurate even when the target is outside the triangle but inside the large dotted circle. Generally speaking, if the target is located inside the circumscribed circle passing through all three sensors, RVI accurately estimate the target location. If the target is outside the circumscribed circle, as shown by cases B and D, the target tends to be localized at the opposite of the true target location along the line passing the target location and the center of the circumscribed circle. As the target goes farther from the circle, the estimated location becomes closer to the center of the circle, which means the estimation error increases.

As shown in Fig. 2, this approach takes a large number (up to 50) of iterations to accurately point the true location. It is because the ratio difference $(g_i - g_{i,j})$ and, eventually, the step size $|\overrightarrow{V_j}|$ gets smaller as X_j becomes closer to the target location X. Thus, in order to reduce the number of iterations for convergence, we rewrite Eq. (3) to set the step size as a constant, c:

$$X_j = X_{j-1} + \Delta_j,$$

$$\Delta_j = \begin{cases} c\dfrac{\overrightarrow{V_j}}{|\overrightarrow{V_j}|} & \text{if} \quad j > N_{\text{init}} \text{ and } |\overrightarrow{V_j}| < c, \\ \overrightarrow{V_j} & \text{otherwise.} \end{cases} \tag{4}$$

In the first N_{init} number of iterations[1], the moving distance Δ_j is allowed to be smaller than c so that iteration vectors can progressively change its direction toward the target location. If the initial iterations have an inappropriate large vector size, the direction of the move vector would change drastically, and the algorithm will stop by the condition $\overrightarrow{V_{j-1}} \cdot \overrightarrow{V_j} < 0$ before getting closer to the true location.

The algorithm terminates in a finite number of iterations by the termination conditions. With Eq. (4), in particular, the condition $\overrightarrow{V_{j-1}} \cdot \overrightarrow{V_j} < 0$ guarantees the termination of algorithm around the true target location X because the constant step size may not achieve fine-grained convergence, and eventually the direction of the move vector changes drastically (more than 90°) around the true location.

[1] $N_{\text{init}} = 3$ in this paper.

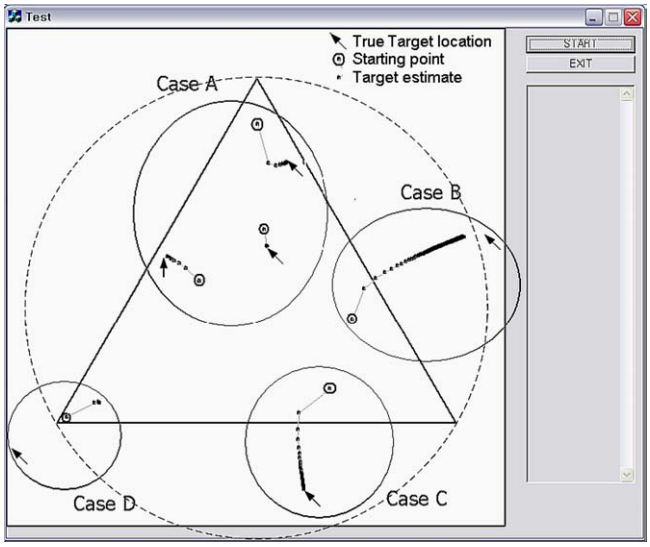

Fig. 2. Examples of Ratiometric Vector Iteration.

There is a tradeoff between the estimation accuracy and the total number of iterations. As the step size c increases, the algorithm converges to the target location with the less number of iterations, and the estimation error $|X_{RV} - X|$ increases, and vice versa. When the new Eq. (4) is applied to the test program with $c = 0.1$ while the inter-sensor distance is 1.0, the maximum number of iterations decreases to 5 and the algorithm typically performs 3–4 iterations. With this configuration, the average error is about 0.05, half of the step size c.

It can be shown that the time complexity of RVI is $O(kl)$ where RSS and location information are given from k sensors, and l is the number of iterations. We can limit the number of participating sensors (in this paper, three sensors) and limit the number of iterations to a constant number (by adjusting the step size c): the complexity becomes $O(1)$.

3.4. Target-in-triangle/target-in-circle issues

In the previous subsection, as long as the target resides within the circumscribed circle, RVI achieves good localization results (half of the step size, on the average). In contrast, if the target is outside the circumscribed circle, the estimation error increases up to the distance between the target location and the center of the circle.

To observe the probabilities that the target is inside the triangle (convex area) and the circumscribed circle of the three closest sensors, an intensive simulation work is performed by using Qualnet network simulator [4]. A total of 400 sensors are deployed in the area of $190 \times 190\ m^2$. To be free from the edge effect, the target moves only in the inner area of $150 \times 150\ m^2$ and the edge area within 20 m from the area boundary are excluded. Table 1 shows the probability of having the target inside the triangle/cir-

Table 1
Probability of target-in-triangle/circle

Sensor placement	Grid	Uni	Rnd
Triangle	1.0	0.41	0.25
Circumscribed circle	1.0	0.82	0.73

cumscribed-circle formed by the three closest sensors from the target. Grid, uniform (Uni)[2], and random (Rnd) node placement strategies are examined. Note that the target location is inside the circumscribed circle with high probability (0.73–1.0) compared to the triangle case. This is the merit of RVI because it can more effectively handle the geometric dilution of precision (GDOP) problem than traditional geometric positioning systems.

By using additional information about target states, such as its trajectory and moving direction, more appropriate convex-area-forming sensors can be selected such that the target resides within the triangle/the circumscribed circle with higher probabilities. The use of Voronoi diagram [16] can find 3 convex-area-forming sensors with the extra computation and communication overhead. Another way to increase the above probabilities is to make a more number of sensors ($k > 3$) to participate in a target localization. However, the increased number of sensors indicates the increased message overhead: in this paper, we confine to the case with three sensors which is the minimum number to localize a target in a two dimensional space.

4. Target tracking framework

The proposed framework consists of three main procedures: (a) sensing and buzzing, (b) leader selection and localization, and (c) reporting to sink nodes. Before elaborating on each procedure, we first introduce network assumptions.

4.1. Assumptions

First, we assume that each sensor is aware of its own location and stationary. These are common assumptions for many sensor network applications. For simplicity and ease-of-description, we assume isotropic signal propagation from the target. Thus, the sensing area of a sensor is shaped as a circle with a radius R centered at the location of the sensor. Likewise, the sensing area for a target also forms a circle centered at the location of the target. Because the original signal strength at a target, a, is unknown in the sensing model of (1), the radius R is also unknown. We assume that a has its upper bound $a_{\max}$ and the ith sensor detects the target when the RSS value r_i exceeds a pre-determined threshold r_{th}. Therefore, a sensing range R also has an upper bound $R_{\max}$ which is derived from (1):

$$r_i = \frac{a}{d_i^{\alpha}} > r_{\text{th}} \Longleftrightarrow d_i < a^{1/\alpha} \cdot r_{\text{th}}^{-1/\alpha}$$
$$\therefore R_{\max} = a_{\max}^{1/\alpha} \cdot r_{\text{th}}^{-1/\alpha}, \tag{5}$$

where $d_i = |X - S_i|$. And we assume that sensors can directly communicate with the neighboring sensors within a radius at least $2R_{\max}$. This assumption is typical and even conservative considering the field data. For example, MICA II Berkeley mote [3] has a transmission range of up to 300 m in open space, while the sensing range is at most 30 m even with long-range infrared sensors. Photoelectric sensors and acoustic sensors have a sensing range of about 10 m.

The second assumption is a coarse-grained time synchronization among neighboring nodes, which can be easily achieved by current millisecond-level synchronized solutions. This level of time synchronization is also required by MAC protocols such as S-MAC [5]. In our framework, the leader periodically reports the estimated target location, which will help the sensors nearby the target to be synchronized.

Third, it is assumed that different targets are far enough apart that each sensor can detect only one target at a time. Detecting the presence of multiple targets and tracking them require additional sensing and signal processing algorithms [6,7] which are beyond the scope of this paper.

4.2. Target tracking cycle

4.2.1. Target sensing and buzzing

Fig. 3 shows that the three procedures compose a tracking cycle. Each sensor that detects a signal whose RSS exceeds the threshold r_{th} broadcasts a buzz to its one-hop neighbors. A buzz from the ith sensor contains the sensor's coordinates (S_i) and the RSS (r_i).

Each target-detecting sensor sets a backoff timer and waits until the timer expires. Specifically, the backoff timer of the ith sensor, B_i, is

$$B_i = W_{\text{size}} \cdot \frac{r_i^{-1/\alpha}}{r_{\text{th}}^{-1/\alpha}}, \tag{6}$$

where W_{size} denote the backoff window size. In summary, the range of the backoff timer value is $[0, W_{\text{size}}]$. Before the timer expires, the ith sensor keeps listening to other sensors' buzzes and if it receives three or more buzzes, it cancels its backoff timer. This ensures that only three sensors generate buzzes. Those sensors will have the highest RSS values among target-detecting sensors with a high probability.

In Section 5, we will show that B_i is linearly proportional to the distance between the target and the ith sensor. This property guarantees that the first three buzzes will be transmitted successfully with a low collision probability (Lemma 1).

4.2.2. Leader selection and localization

After W_{size} time has passed from the time of backoff timer setting, the sensor with the highest RSS volunteers to become

[2] In uniform distribution, the area is equally divided into a number of cells based on the number of sensors. Within each cell, a sensor is placed randomly.

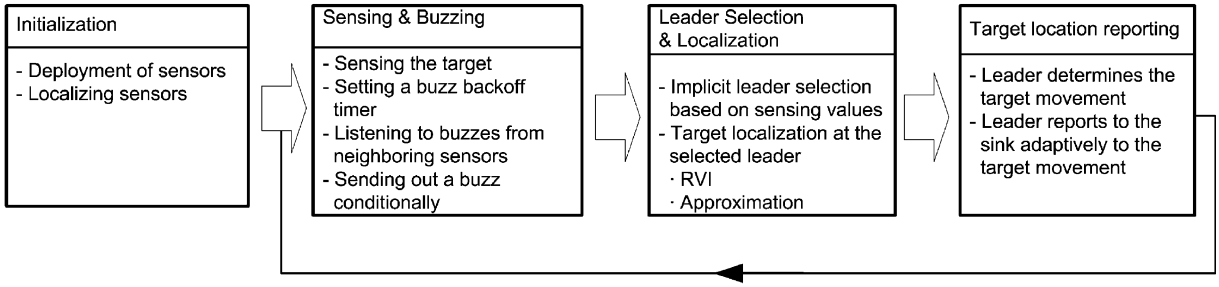

Fig. 3. Target tracking cycle.

a leader. Then, the leader localizes the target using three RSS values including its own RSS, and uses the location of three sensors are obtained. Suppose the total number of target-detecting sensors is less than three or the number of successfully transmitted buzzes without collision is less than three. Then, the leader cannot perform RVI (neither weighted centroid). In this case, the target location is estimated by approximation of Subsection 3.2. Otherwise, at least three RSS values are available: the leader performs RVI.

4.2.3. Movement-adaptive report scheduling

After localizing the target, the leader is obliged to report the estimated location to the sink node. Usually, the sink is a gateway connecting the sensor network to subscribers. Therefore, reporting to the sink normally incurs multi-hop message relaying and multiple transmissions which depends on the hop distance between the leader and the sink. If the target is stationary or moves around within a small area, the estimated location does not change notably as time goes by and the same location estimate (or possibly with small deviation) is repeatedly reported to the sink. In these cases, it is not necessary and even wasteful to report (almost) same location estimates repeatedly.

Rather than reporting every time leader performs localization, we propose to dynamically schedule a reporting period considering the target's movement (e.g., velocity). Specifically, the current leader compares the current location estimate X_{current} with the previously reported location estimate X_{previous} and report to the sink if the distance between them $|X_{\text{current}} - X_{\text{previous}}|$ is longer than D_{th}. If the target has been stationary for a long time, the sink would have received no report and may conclude that the target is lost or disappeared. To prevent this, we set an upper bound of the reporting period: if more than T_{th} time has passed after the previous-report-reception time T_{previous}, the leader reports to the sink regardless of the above distance condition. In summary, the leader reports if and only if

$$|X_{\text{current}} - X_{\text{previous}}| > D_{\text{th}}$$

or

$$T_{\text{current}} - T_{\text{previous}} > T_{\text{th}}, \tag{7}$$

where T_{current} denotes the current time. Note that both T_{current} and T_{previous} are not absolute time but relative time instance according to the leader's clock.

If the condition (7) is satisfied, the leader sends out the report when W_{size} time has passed from the backoff-timer-setting time in order to avoid collisions between the report and the buzzes. However, there is a possibility of collisions among report messages. Suppose a collision occurs between the first two buzzes with the highest and the second highest RSS values. Then, three more buzzes will be broadcast by sensors which have detected the third, fourth, and fifth highest RSS values. Because the sensors with the first and second highest RSS values do not know the collision, both of them deem their RSS values as the highest and volunteer as a leader. Moreover, the sensor with the third highest RSS also deem itself as a leader because it does not receive the first and second buzzes. Consequently, the three sensors will try to transmit reports simultaneously. This case only happens when the first buzz collides (with the second buzz), and the first buzz collision probability is derived in the next section. One solution for this problem is to use a report backoff timer although we do not include the report backoff timer in the simulation. Because only three reports contend, the report backoff timer may have a small window size.

4.3. Tracking cycle timing

Fig. 4 shows a timing diagram of the tracking cycle described in the previous subsection. The total cycle time T_{cycle} is

$$T_{\text{cycle}} = W_{\text{size}} + T_{\text{tx}} + T_{\text{gap}},$$

where T_{tx} is the one-hop transmission time of a report message (we ignore the processing time for localization and the propagation delay), and T_{gap} is the time between the report-reception time and the backoff-setting time.

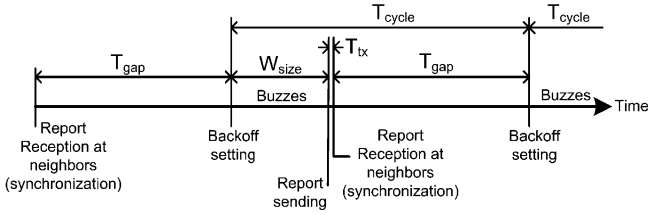

Fig. 4. Traking cycle timing.

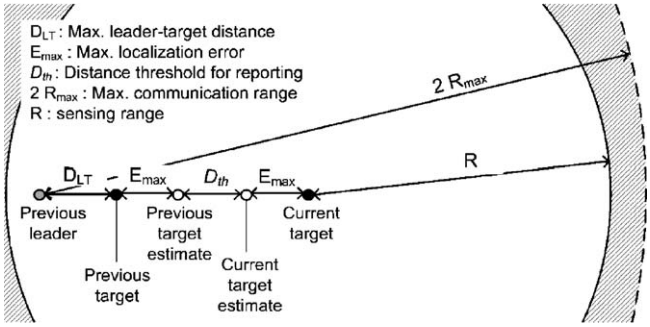

Fig. 5. The limitation of D_{th} for time synchronization.

Due to the assumption of $2R_{\max}$ communication range, all target-detecting sensors are assumed to have received a report message from the previous leader. Therefore, the time when sensors receive (actually overhear) the report message from the leader can be used for time-synchronization among target-detecting sensors. However, this assumption limits D_{th} as illustrated by Fig. 5. D_{LT} denotes the maximum distance between the previous leader (that sent the previous report) and the previous target location, and $E_{\max}$ is the maximum localization error. In the case of a grid sensor distribution, D_{LT} can be approximated using the inter-sensor distance: $D_{\text{LT}} = \frac{\sqrt{2}}{2} D_{\text{inter\_sensor}}$. The solid circle and the dotted circle denote the target's sensing range and the previous leader's communication range, respectively. Fig. 5 shows the worst case when the distance between the current target location and the previous leader is the longest. For the purpose of time synchronization, all sensors that detect the current target should reside within the communication range of the previous leader:

$$D_{\text{LT}} + E_{\max} + D_{\text{th}} + E_{\max} + R \leqslant 2R_{\max},$$

$$\therefore D_{\text{th}} \leqslant 2R_{\max} - R - \frac{D_{\text{inter\_sensor}}}{\sqrt{2}} - 2E_{\max}.$$

Suppose that $R = R_{\max} = 25$ m (worst case), $D_{\text{inter\_sensor}} = 10$ m and $E_{\max} = 5$ m. Then, approximately, D_{th} must be 7.9 m or less.

Because the reporting interval can be extended up to $T_{\text{th}} (> T_{\text{cycle}})$, the leader may not send a report at every T_{cycle} interval if the target is stationary. Therefore, a target-detecting sensor may have to send buzzes without receiving a report up to $\lfloor \frac{T_{\text{th}}}{T_{\text{cycle}}} \rfloor$ times.

5. Analysis of buzz collision probability

In this section, we analyze the probability of successful buzz transmission and show how many buzz transmissions are needed to receive three buzzes successfully. Random sensor distribution is assumed and the sensor density is given by ρ.

According to Eq. (6), the backoff timer, B_i, is deterministically related to the sensing RSS r_i. Because the RSS r_i is larger than the threshold value r_{th} by Eq. (5), the term $\frac{r_i^{-1/\alpha}}{r_{\text{th}}^{-1/\alpha}}$ of Eq. (6) ranges $(0, 1)$, and from Eq. (5), this term can be expressed as

$$\frac{r_i^{-1/\alpha}}{r_{\text{th}}^{-1/\alpha}} = \left(\frac{a_{\max}}{a}\right)^{1/\alpha} \cdot \frac{1}{R_{\max}} \cdot d, \tag{8}$$

where a is the original signal strength at the target source, $a_{\max}$ is the upper bound of a, $R_{\max}$ is the upper bound of the sensing range R, and d denotes the distance between the target and a sensor. Note that a, $a_{\max}$, R and $R_{\max}$ are constants, while d is a variable which ranges $(0, R)$. Eq. (8) implies that the backoff timer, B_i is proportional to d. By the definition of Eq. (6) and the implication of Eq. (8), the range of backoff time window $(0, W_{\text{size}})$ *linearly maps* to the range of distance $(0, R)$, where the sensing range R is determined by setting $r_i = r_{\text{th}}$ and $\alpha = 2$ as follows:

$$R = d_{(r_i = r_{\text{th}})} = \sqrt{\frac{a}{a_{\max}}} \cdot R_{\max}. \tag{9}$$

We assume that a collision occurs if the transmission times for two buzzes overlap. Due to the carrier sensing capability, a sensor can withhold its transmission if it senses other sensor's transmission for a carrier sensing time or more. In other words, if one buzz starts transmission at time t, then any other transmission starting between $t - T_{\text{cs}}$ and $t + T_{\text{cs}}$ will cause a collision, where T_{cs} is the carrier sensing time (duration). Due to the linear mapping of the backoff time range $(0, W_{\text{size}})$ and the distance range $(0, R)$, T_{cs} is translated to D_{cs}, which is the minimum difference of distances between the target and two sensors required to avoid a collision: $D_{\text{cs}} = R \frac{T_{\text{cs}}}{W_{\text{size}}}$. Also, the time instance t (from the timer-setting time) is mapped to the locus of positions whose distance from the target is d: $d = R \frac{t}{W_{\text{size}}}$.

Then, the expected number of transmissions that collide with a transmission started at time instance t is given by

$$E_{\text{col}}(d) = \rho \cdot \int_{\max(0, d - D_{\text{cs}})}^{\min(R, d + D_{\text{cs}})} 2\pi x \, \mathbf{d}x. \tag{10}$$

Then, the probability that a transmission started at time instance t is successful without a collision is given by

$$P_{\text{suc}}(d) = \begin{cases} 1 - E_{\text{col}}(d) & \text{if } E_{\text{col}}(d) < 1, \\ 0 & \text{if } E_{\text{col}}(d) \geqslant 1. \end{cases} \tag{11}$$

$P_{\text{suc}}(d)$ monotonically decreases as d increases. From Eq. (11), Lemma 1 provides an important observation.

Lemma 1. Early buzzes (possibly from the sensors close to the target) have a higher successful transmission probability than late buzzes. Therefore, the first three buzzes will be transmitted successfully with a high probability.

Finally, the expected number of attempted transmissions in the range of $(0, t)$ is

$$H(t) = H_d(d) = \rho \cdot \int_0^d 2\pi x \, \mathbf{d}x = \rho \cdot \pi \cdot d^2 \tag{12}$$

and the expected number of the successful transmissions in the range of $(0, t)$ is

$$G(t) = G_d(d) = \rho \cdot \int_0^d (P_{\text{suc}}(x) \cdot 2\pi x)\ \mathbf{d}x. \tag{13}$$

Fig. 6 plots $H(t)$ (dotted line) and $G(t)$ (solid line) when parameters are set as specified in Table 2, $a_{\max} = 1$, and 400 sensors are randomly placed in the area of 200×200 square meters, i.e., $\rho = \frac{400}{200^2}$ sensors/m^2. As the two lines get closer, the collision probability becomes ignorable. The expected number of buzz transmissions until receiving three successful buzzes is $H(t')$ when $G(t') = 3$. As shown in Fig. 6, with $a = 0.2$, $H(t')$ is about 3.09 when $G(t') = 3$. Thus, 3.09 buzz transmissions are needed in order to receive three buzzes successfully. With $a = 1.0$, $H(t')$ is about 3.22 when $G(t') = 3$.

With $a = 0.2$, $G(t') \cong 1$ and $H(t')$ is about 1.017 which means that the first buzz collision probability is $\frac{1.017-1}{1.017} \approx 0.017$. Likewise, with $a = 1.0$, $H(t')$ such that $G(t') = 1$ is about 1.04 which means that the first buzz collision probability is about 0.038.

6. Numerical results

In this section, we evaluate the proposed tracking framework which is implemented on QualNet simulator [4] and Matlab [21]. The surveillance area is 190×190 m^2. Four hundred sensors including a sink are deployed in this area with three topologies: grid, uniform, and random distribution. A static routing path forming a tree rooted at the sink is used in the network layer and IEEE 802.11b is adopted as the MAC layer protocol. The sensing range of a sensor device is 25 m and the transmission range is 50 m (twice the sensing range). We adopt the random-waypoint model as a mobility model for the target, which is moving around within the surveillance area. The maximum and minimum speed of the target movement are 20 and 1 m/s, respectively. The default pause time is set to 20 s unless a specific comment is given. Throughout the simulation experiments, T_{cycle} is one second. Each distance estimate from a corresponding RSS measurement contains a random noise component which follows the White Gaussian noise model of $\mathcal{N}(0, \sigma)$, and noise magnitude (σ) has a default value of 1.6 m. The default sensor topology is a uniform distribution. Other default parameter settings are given by Table 2.

The performance metrics are as follows:

1. Localization error: the difference between the true target location and the estimated location at the leader.
2. Tracking error: the difference between the true target location and the most recently reported location at the sink.
3. The total number of reports sent by the leaders.
4. The total number of transmitted messages: the sum of the buzz messages, the report messages sent by the leaders, and the report-forwarding messages relayed by intermediate sensors between the leaders and the sink.
5. The number of iterations until RVI converges.

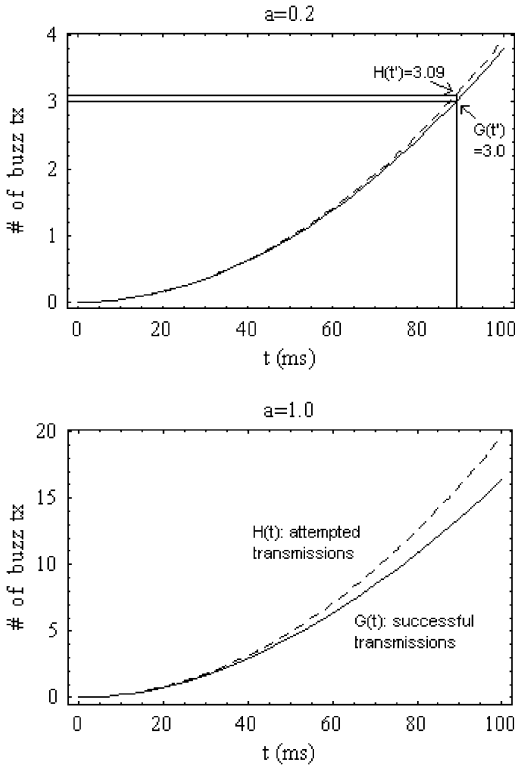

Fig. 6. The average numbers of the attempted transmissions and the successful transmissions by the time t.

Four different combinations of localization and report scheduling algorithms are compared:

- approximation + non-movement-adaptive (Appr),
- Newton's method + non-movement-adaptive (Newton),
- RVI + non-movement-adaptive (RVI + Non), and
- RVI + movement-adaptive (RVI + Adap).

Here, Newton's method refers to the nonlinear optimization approach in Section 3.2.

The performance comparison of the four combinations are shown in Table 3, which comes from ten runs of the simulation experiments. Overall, with 1.6 m noise magnitude, RVI shows about half of the localization error compared to the approximation localization algorithm and reduces the standard deviation. Moreover, the localization error and standard deviation of RVI

Table 2
Parameters for the proposed tracking framework

Notation	Parameter	Default
W_{size}	Window size (ms)	100
T_{cycle}	Cycle time (ms)	1000
T_{cs}	Carrier sensing time (ms)	0.32
$R_{\max}$	Maximum sensing range (m)	25
c	RVI step size (m)	1.5
D_{th}	Distance threshold for reporting (m)	5
T_{th}	Time threshold for reporting (ms)	5000

The first four parameters are for general configurations; the fifth one is specific to RVI; the remaining parameters are specific to movement-adaptive reporting.

Table 3
Performance comparison under uniform sensor distribution and 1.6 m noise magnitude

	Localization error (m)		Tracking error (m)		Total reports	Total messages
	Average	SD	Average	SD		
Appr	4.35	2.12	4.43	2.33	1000	10,097
Newton	2.21	1.54	2.29	1.76	1000	10,228
RVI + non_adap	2.41	1.61	2.48	1.84	1000	10,252
RVI + adap	2.41	1.61	2.83	2.27	439	6,444

Other parameters follow Table III.

are close to the performance of the sophisticated Newton's method. Also, the adaptive reporting framework reduces the total report messages from the leaders down to less than a half of that of the non-adaptive reporting framework. However, this reduction of the report messages slightly increases the tracking error compared to the localization error.

Fig. 7 plots the tracking error versus the noise magnitude. Since the approximation localization algorithm has a coarse-grained localization error, the effect of small noise magnitude is not noticeable. However, as the noise magnitude increases, all of the four combinations show increasing tracking errors. Also, the difference of the localization error between the approximation algorithm and the RVI algorithm declines as the noise magnitude increases.

Fig. 8 compares the number of the total messages (buzz messages, report messages, and report-forwarding messages) of non-movement-adaptive reporting and movement-adaptive reporting. Obviously, as the pause time (between two moving periods) increases, the number of the total messages falls off while slightly decreasing tracking errors (that is not shown in the graph). Note that even when the pause time is zero, the total number of messages of the adaptive reporting framework is notably less than that of the non-adaptive reporting framework.

Fig. 9 shows two kinds of outcomes. First, it plots the tracking error as the step size of the RVI algorithm increases in three topologies: random, uniform, grid. As also hinted

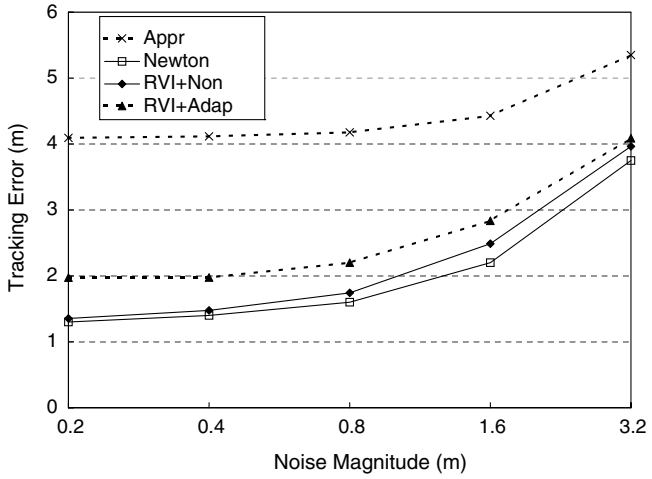

Fig. 7. Effect of the noise magnitude on the tracking error.

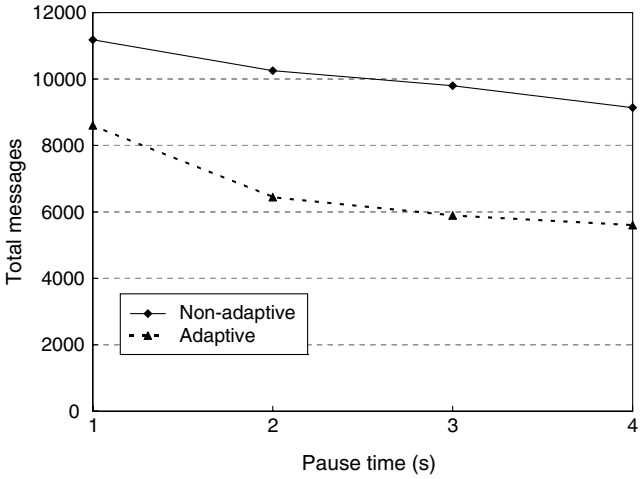

Fig. 8. Effect of the pause time of the target on the number of the total messages.

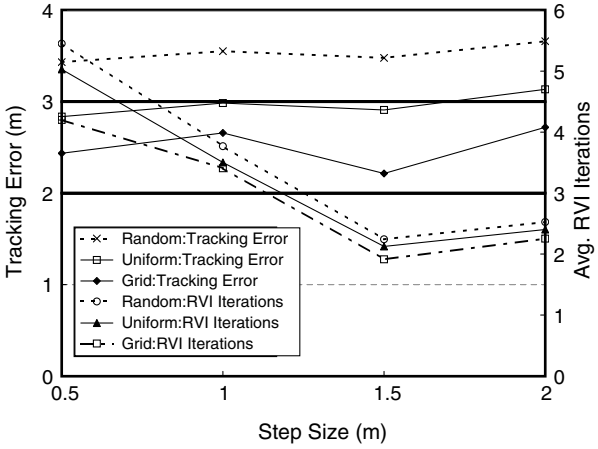

Fig. 9. Effect of the RVI step size on the tracking error and the number of RVI iterations.

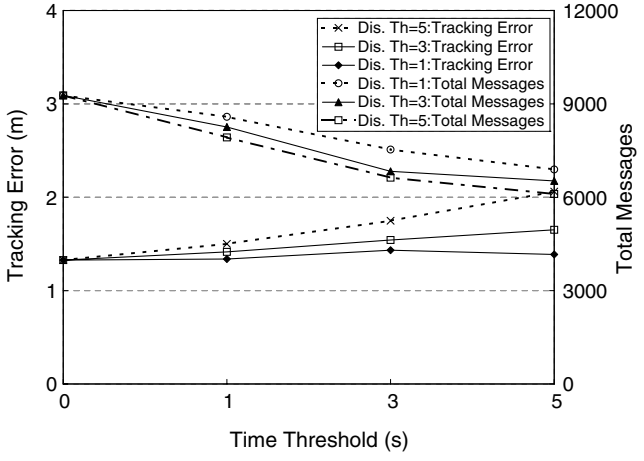

Fig. 10. Effect of the distance and time threshold of the adaptive report scheduling algorithm on the tracking error and the number of total messages.

in Table 1, this figure indicates that the more regularly the sensor nodes are deployed, the less becomes the localization error (and hence the tracking error). Second, as the step size of RVI increases, the number of RVI iterations decreases with a small turnover at the step size 1.5.

Fig. 10 shows the tracking error and the number of the total messages with different distance thresholds (D_{th}) as the time threshold (T_{th}) varies. The noise magnitude is 0.2 m. The time threshold of zero second indicates the non-movement-adaptive reporting. Note that there is a tradeoff between the tracking error and the message overhead. As the time threshold and/or the distance threshold increases, the tracking error increases and the message overhead decreases. The message overhead is more sensitive in response to the change of the time threshold than the tracking error because the tracking error can not be less than the localization error that depends on the noise magnitude and the step size.

7. Conclusions

The proposed framework consists of three main components: RVI localization, 3-neighbor sensor discovery, and the movement-adaptive reporting mechanism. Ratiometric Vector Iteration (RVI) is based on distance ratio estimates rather than absolute distance estimates which are often impossible to calculate. In the RVI algorithm, only three closest sensors (to the target) broadcasts their received signal strengths, which are called buzzes. The proposed RSS-based backoff timer enables successful buzz transmissions of the three closest sensors with a low collision probability. Also, a reporting algorithm that is adaptive to the target's movement is introduced to further reduce the energy consumption. That is, the energy efficiency is achieved by reducing the number of messages, while maintaining the accuracy required by applications. Extensive simulation results show that the proposed RVI algorithm renders less than half of the localization error of the approximation algorithm and this result is close to that of nonlinear optimization. Moreover, the proposed report scheduling algorithm reduces the total number of transmitted messages substantially compared to the baseline framework which is not adaptive to the target movement.

Acknowledgements

This work was supported in part by the Brain Korea 21 project of the Ministry of Education, 2005, Korea.

References

[1] R. Powers, Batteries for low power electronics, Proc. IEEE (1995) 687–693.
[2] D. Culler, D. Estrin, M. Srivastave, Overview of sensor networks, IEEE Comput. Mag. (2004).
[3] Crossbow Technology, Inc., <http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf>, 2005.
[4] Network Simulator – Qualnet, <http://www.scalable-networks.com>.
[5] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: Proc. of IEEE INFOCOM 2002, June 2002.
[6] X. Sheng, Y. Hu Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks, in: IEEE Transactions on Signal Processing, vol. 53, No. 1, January 2005.
[7] J. Liu, J. Liu, J. Reich, P. Cheung, F. Zhao, Distributed group management for track initiation and maintenance in target localization applications, in: Proc. Second Workshop Information Processing in Sensor Networks, April 2003.
[8] Y. Shang, et al., Localization from mere connectivity, in: Proc. ACM MOBIHOC 2003, June 2003.
[9] L. Doherty, L.E. Ghaoui, K.S.J. Pister, Convex position estimation in wireless sensor networks, in: Proc. IEEE INFOCOM 2001, April 2001.
[10] D. Moore, J. Leonard, D. Rus, Robust distributed network localization with noisy range measurements, in: Proc. ACM SenSys 2004, November 2004.
[11] T. He, et al., Range-free localization schemes in large scale sensor networks, in: Proc. ACM MobiCom 2003, September, 2003.
[12] D. Niculescu, B. Nath, Ad hoc positioning system (APS), Proc. IEEE GLOBECOM (2001).
[13] N. Bulusu et al., GPS-less low cost outdoor localization for very small devices, IEEE Pers. Commun. Mag. vol. 7 (no. 5) (2000).
[14] F. Zhao, J. Shin, J. Reich, Information-driven dynamic sensor collaboration for tracking applications, IEEE Signal Process. Mag. (2002).
[15] W. Zhang, G. Cao, DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks, IEEE Trans. Wireless Commun. (2004).
[16] W. Chen, J.C. Hou, L. Sha, Dynamic clustering for acoustic target tracking in wireless sensor networks, IEEE ACM/USENIX Conference Mobile System on Mobile Computing vol. 3 (no. 3) (2004).
[17] H. Yang, B. Sikdar, A protocol for tracking mobile targets using sensor networks, in: Proc. IEEE Workshop Sensor Network Protocols and Applications, (in conjunction with IEEE ICC), May 2003.
[18] S. Tilak, V. Kolar, N.B. Abu-Ghazaleh, K.D. Kang, Dynamic localization control for mobile sensor networks, in: IEEE International Workshop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks (IEEE IWSEEASN'05), April 2005.
[19] D. Li, K. Wong, Y. Hu, A. Sayeed, Detection, classification, tracking of targets in micro-sensor networks, IEEE Signal Process. Mag. (2002) 17–29.
[20] D.P. Bertsekas, Nonlinear Programming, second ed., Athena Scientific, Belmont, Mass, 1999.
[21] Matlab, <http://www.mathworks.com>.

Jeongkeun Lee. He received a B.S. degree in computer engineering from Seoul National University in 2001. He is an Ph.D. candidate in the Multimedia and Mobile Communications Lab., School of Computer Science and Engineering, Seoul National University. He was a visiting researcher at University of California at Santa Cruz and Fraunhofer FOKUS, Germany in 2002 and 2005, respectively. His research interests include wireless mesh networks, RFID systems, and user localization and tracking in wireless networks.

Kideok Cho. He received a B.S. degree in computer science and engineering from Seoul National University in 2004. He is a master course student in the Multimedia and Mobile Communications Lab., School of Computer Science and Engineering, Seoul National University. His research interests include wireless sensor networks, RFID systems, and mobility management.

Seungjae Lee. He received a B.S. degree in computer science and engineering from Seoul National University in 2005. He is a master course student in the Multimedia and Mobile Communications Lab., School of Computer Science and Engineering, Seoul National University. His research interests include internet identifier system and distributed hash table.

Taekyoung Kwon. He is an assistant professor in Multimedia and Mobile Communications Lab., School of Computer Science and Engineering, Seoul National University. He received his Ph.D., M.S., and B.S. degrees in computer engineering from Seoul National University in 2000, 1995, and 1993, respectively. He was a visiting student at IBM T.J. Watson Research Center in 1998 and a visiting scholar at the University of North Texas in 1999. His recent research areas include radio resource management, wireless technology convergence, mobility management, and sensor network.

Yanghee Choi. He received B.S. in electronics engineering from Seoul National University, M.S. in electrical engineering from Korea advanced Institute of Science, and Doctor of Engineering in Computer Science from Ecole Nationale Superieure des Telecommunications (ENST) in Paris, in 1975, 1977 and 1984, respectively. Before joining the School of Computer Engineering, Seoul National University in 1991, he has been with Electronics and Telecommunications Research Institute (ETRI) during 1977–1991, where he served as director of Data Communication Section, and Protocol Engineering Center. He was research student at Centre National d-Etude des Telecommunications (CNET), Issy-les-Moulineaux, during 1981–1984. He was also Visiting Scientist to IBM T.J. Watson Research Center for the year 1988–1989. He is now leading the Multimedia Communications Laboratory in Seoul National University. He was editor-inchief of Korea Information Science Society journals. He was chairman of the Special Interest Group on Information Networking. He has been associate dean of research affairs at Seoul National University. He was president of Open Systems and Internet Association of Korea. His research interest lies in the field of multimedia systems and high-speed networking.