Auction-based adaptive sensor activation algorithm for target tracking in wireless sensor networks

Jin Zheng\textsuperscript{a}, Md Zakirul Alam Bhuiyan\textsuperscript{a}, Shaohua Liang\textsuperscript{a}, Xiaofei Xing\textsuperscript{b}, Guojun Wang\textsuperscript{a,}\textsuperscript{*}

\textsuperscript{a} School of Information Science and Engineering, Central South University, Changsha, Hunan Province, 410083, PR China

\textsuperscript{b} School of Computer Science and Educational Software, Guangzhou University, Guangzhou, Guangdong Province, 510006, PR China

**HIGHLIGHTS**

- We present an auction mechanism for the problem of target tracking in WSNs.
- We propose an adaptive sensor activation algorithm for the target tracking.
- We improve prior trilateration algorithm for low target localization errors.
- The proposed algorithms provide high energy-efficiency and tracking quality.

**ABSTRACT**

Due to the severe resource constraints in wireless sensor networks (WSNs), designing an efficient target tracking algorithm for WSNs in terms of energy efficiency and high tracking quality becomes a challenging issue. WSNs usually provide centralized information, e.g., the locations and directions of a target, choosing sensors around the target, etc. However, some ready strategies may not be used directly because of high communication costs to get the responses for tracking tasks from a central server and low quality of tracking. In this paper, we propose a fully distributed algorithm, an auction-based adaptive sensor activation algorithm (AASA), for target tracking in WSNs. Clusters are formed ahead of the target movements in an interesting way where the process of cluster formation is due to a predicted region (PR) and cluster members are chosen from the PR via an auction mechanism. On the basis of PR calculation, only the nodes in the PR are activated and the rest of the nodes remain in the sleeping state. To make a trade-off between energy-efficiency and tracking quality, the radius of PR and the number of nodes are adaptively adjusted according to current tracking quality. Instead of fixed interval (usually used in existing work), tracking interval is also dynamically adapted. Extensive simulation results, compared to existing work, show that AASA achieves high performance in terms of quality of tracking, energy efficiency, and network lifetime.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The rapid development in low-power micro-electro-mechanical system (MEMS) technology, microprocessors, and wireless communications has brought tremendous attention to the research in wireless sensor networks (WSNs). WSNs consist of hundreds of low-cost, low-power, multi-functional micro-sensor nodes with the capabilities of sensing, processing, and wireless communications \cite{1,2}. Sensor nodes have been deployed to play significant roles in traffic control, battlefield, habitat monitoring, and intruder tracking in recent years \cite{1–6}.

Target tracking with WSNs has gained much attention in recent years. The main aim of target tracking is to find out the location, velocity, and direction of a target instantaneously. Accurate location is the most important information for a target tracking system. For example, we can monitor and track a target like a person or a vehicle that is moving under the radio coverage of a WSN, and sensor nodes report the location information of the target to a control center (or the sink node) periodically. Here, the sink is the node that collects the information from the nodes and processes them, and makes an aggregated tracking information for end users. However, there are difficulties in achieving such information from WSNs as WSNs have inherent limitations, e.g., the unattended sensor nodes, the strict power constraint, and the limited computational capability of sensor nodes \cite{3}. Energy depletion of sensor nodes in a region of interest causes the emergence of the connectivity and coverage holes, which finally leads to the network failure. Therefore, designing an efficient target tracking algorithm with energy efficiency, high tracking quality, and low computational complexity becomes a highly challenging problem.
Various target tracking algorithms for WSNs have been proposed recently [1,2,7–13]. The sensor activation problem as a research area has received considerable attention in recent years. Some prior algorithms have been shown to outperform other solutions for the problem, namely, SARA (sensor activation and radius adaptation), DLM (distributed lifetime maximization), VRSC (variable radii connected sensor cover), DSA (distributed sensor activation algorithm) [14–17]. For a specific application, such as target tracking in WSNs, sensor activation in a region of interest of a WSN is a critical task. On the one hand, some of the existing sensor activation algorithms have high computation complexity and high communication costs to get responses for tracking tasks from the sink, such as filter based tracking algorithms [18,19,11], while others have crucial requirements on the capabilities of sensors or the settings of the WSN, such as the algorithms presented in [20–22]. On the other hand, a set of algorithms (e.g., [23]) activates a large number of sensors for tracking, which is also not efficient for a tracking system. Overall, the quality of tracking is not sufficient in these algorithms. To make every step of tracking operation energy-efficient while ensuring the quality of tracking, the following concerns should be addressed adequately:

(Q1) Which subset of sensor nodes should be activated to form a cluster before the target moving toward the nodes?

(Q2) How is a target located when considering practical deployment issues, such as measurement noises or obstacles?

(Q3) How are sensor nodes scheduled in an energy-efficient manner, while guaranteeing the tracking quality?

In response to the similar concerns above, cluster-based tracking algorithms normally used [24,25]. In a target tracking application, both cluster member nodes and cluster head (CH) node can detect the target. The burden on the CH is much heavier than on the cluster member nodes, for example, the CH has to collect the data from the members and finish the data fusion and communicates with the sink. Thus, the CH consumes more energy than the member nodes. Selections of CH and cluster members have important effects on both the quality of tracking and the network lifetime. In prior work, clusters are either formed at the network initialization or at the time of tracking. Both have effects in tracking quality, e.g., the cluster may not be exactly found as it was formed at the initialization, there can be delays caused by cluster formation as a target is detected.

To address those concerns above to a great extent, specifically, energy-efficiency and tracking quality, we come up with the following ideas and algorithms.

(a) We introduce an “auction mechanism” to form a cluster ahead of a target as it moves. The auction mechanism is employed to form a cluster ahead of the target moving toward the cluster. In each iteration of tracking, a CH in the current cluster compute a predicted region (PR) using a prediction method, where the target is likely to move based on the current and previous movement information of the target. At the end of each iteration, the current CH broadcasts an auction message to activate the nodes in the PR. Then, the CH acts as the auctioneer and the nodes in the PR act as bidders. Each bidder evaluates the tracking task and responds the CH by a bid. The CH ranks the bids to choose appropriate sensor nodes for tracking. The sensor node with the highest bid is selected as the next CH, and the other appropriate sensor nodes are chosen to be the members of predicted cluster. Our proposed auction mechanism requires few computations and reduces communications between the nodes in the PR. The nodes with a lower bid do not need to participate in tracking. This mechanism does not consume significant energy of the network compared to many existing clustering mechanisms, such as LEACH [26], TopDisc [27], and GAF [28].

(b) We propose an auction-based adaptive sensor activation algorithm (AASA) to make a trade-off between energy efficiency and tracking quality. The algorithm is composed of two parts: (1) the adaptive radius of the PR and the number of members in a cluster; (2) the adaptive tracking time interval. In part (1), both the radius and the number of members are adaptive, which are dynamically adjusted, according to the current tracking quality. This implies that the number of sensors that should be activated for achieving high quality of tracking can be adapted according to the system user. In this paper, we use prediction error as the metric for assessing tracking quality. In part (2), the tracking interval is an important setting to a tracking system. Although the tracking interval is fixed in most of the existing tracking algorithms, we strongly believe that the tracking interval should be related to the velocity of the target considering for energy saving in a WSN. Because a high velocity of the target requires a short interval to capture the target and a low velocity allows a long interval. We found that tracking quality is also greatly affected in those algorithms by the fixed interval. In AASA, the tracking interval is dynamically adapted based on the instantaneous velocity of the target, which saves more energy and keeps the required tracking quality.

(c) In order to locate the target in the current region and compute the PR, we propose an improved trilateration algorithm to locate the target approximately. An original trilateration algorithm is available in the ideal sensor network that does not handle the detection errors occurred in the localizations, caused by node unavailability. In a practical deployment, sensing noises and obstacles are inevitable. As a result, using the trilateration algorithm directly in tracking operation is infeasible. Our improved trilateration algorithm takes the detection errors into account.

In summary, the contributions of this paper are as follows.

1. We present an auction mechanism to form a cluster in a predicted region before the target arrives in the region.

2. We propose an adaptive sensor activation algorithm (AASA) to activate a subset of appropriate sensors in the predicted region.

3. We improve an existing trilateration algorithm to provide target localization with low detection errors.

4. We conduct extensive simulations to evaluate our tracking algorithm. Compared to existing work [7,23], our algorithm shows that it is energy efficient and provides prolonged lifetime and high tracking quality.

The rest of this paper is organized as follows. Section 2 reviews the related work. Preliminaries of the approach is given in Section 3. The auction mechanism and the localization algorithm are presented in Section 4. The proposed AASA algorithm is proposed in Section 5. Section 6 discusses the simulation studies. Section 7 concludes this paper.

2. Related work

The research of target tracking using WSNs has been gaining a lot of attention in recent years [1,2,7–13]. Various tracking algorithms have been presented in the literature that can be classified into five categories of schemes, which are tree-based tracking, cluster-based tracking, prediction-based tracking, mobicast message-based tracking, and hybrid methods [29,25]. In this paper, we concentrate on a sensor activation algorithm (based on an auction mechanism) for tracking, for which we need to discuss both the cluster-based and prediction-based methods. More specifically, we need to find a combination of both methods in sensor activation as accurately as possible and in an energy-efficient manner.

2.1. Cluster-based and prediction-based target tracking

A prediction-based clustering algorithm (PBCA) is presented in [7]. In this algorithm, two parameters, distance from predicted

...
location, and remaining energy of sensor nodes, are used for selection of tracking nodes, which causes the nodes with lower energy to stay longer. At each measurement period, the nodes need to exchange extra information, e.g., information about cost of communication between nodes, residual energy, and their location to form a cluster. We think that tracking delays may appear due to the extra packet transmissions, which results in an amount of energy consumption in a WSN.

A multi-point surveillance scheme is proposed in [30]. The authors first examines the relationship between the tracking probability and the sensor density, and then derived an approximate expression for easy estimation of tracking probability from this relationship. Suganya et al. proposes a cluster-based system architecture [31], where the detection algorithm and localization algorithm are performed in different sets of selected sensor nodes. Alaybeyoglu et al. [24] investigates and evaluates the tracking performance of dynamic and static cluster based target tracking approaches against various mobility models. The general conclusion is that the dynamic clustering based target tracking is favorable for accurate tracking, but consumes more energy than static tracking based clustering.

Considering the energy management problem of target tracking applications in WSN, a dynamic energy management mechanism based on dynamic adaptive clustering with intra-cluster optimal routing (DACIOR) is presented in [23]. A communication framework is defined by distributed adaptive clustering (DAC). Integrating the advantage of LEACH [26], the cluster head choosing approach is improved to form more uniform cluster distribution. Clustering is performed in a distributed manner and the cluster size is adjustable. In addition, the optimal paths are obtained by Dijkstra's algorithm in each cluster to reduce intra-cluster communication cost. In this approach, the clustering should be formed initially and be maintained during the system run, and all of the inter–clusters should communicate for tracking. The tracking interval is fixed. The number of overlapping nodes between two clusters is large. Thus, it may have significant energy wasting in the WSN.

The algorithm proposed in [20] chooses the appropriate cluster members that have the best data and lowest cost by an optimal choice mechanism. The algorithm assumes that each node can estimate the cost of communications with its one-hop neighboring nodes. In [21], optimal nodes are selected by a defined selection function, which considers the residual energy and communication cost of nodes. The algorithm requires that each node knows the location and residual energy of its neighbor nodes. Rad et al. [22] proposes an adaptive power optimization scheme, which dynamically modifies the tracking time interval according to the current average velocity of the target, but the algorithm requires high density of sensor nodes to guarantee that at any given location at least three sensors are able to sense the target. In a sparse network setting, the algorithm may not guarantee the quality of tracking.

### 2.3. The use of the auction mechanism

We use the auction mechanism for sensor activation in this work. The auction mechanism has been used in various fields in different contexts, specially, in robotics, resource allocations, spectrum management in sensor networks [33–35,11,36,37]. For robot–robot coordination, a market-based approach [33] is considered. It is based on an auction organized by robots or the sink (central unit) collecting the tasks, the cost of performing tasks by each robot and potential benefit to a team of robots. Robots participating in the auction decide on whether or not to ‘invite’ more robots to the auction, as the invitations themselves cause communication overhead. In [35], the market-based mechanism is used to solve the problem of bandwidth sharing and resource allocation in WSNs in event-driven scenarios, which is different from our sensor activations for tracking. Chen et al. [38] show that the auction allocation mechanism for congestion control performs better than equal allocation or mission priority proportional allocation.

Auction-based congestion management for target tracking in WSNs is presented in [3,39]. Chen et al. [39] used auction mechanism to locally manage network bandwidth allocation, where the congested node acts as the auctioneer, and the packets carrying target updates act as the bidders. In [3], the auction scheme is introduced to the coalition formation process. The coalition leader candidate broadcasts the tracking tasks to recruit coalition members. Each node that received the tasks responds to the leader candidate with a bid. The leader candidate ranks the bids to choose appropriate coalition members, but how to evaluate the bid is not mentioned in their paper.

Comparing with the tracking algorithms mentioned above, our proposed auction-based adaptive sensor activation algorithm (AASA) is different and more efficient. First, the communication energy of AASA is very low. No extra information of neighboring nodes is required to exchange in AASA, such as the sensor node location, cost of communication, and residual energy. Intuitively, there will be a low possibility of tracking delays in AASA, caused by packet delays and delays by the target when it stays idle somewhere for an unspecified amount of time. This is because forming the cluster before the target moves to the cluster in a predicted region reduces the tracking delays and the chance of target missing. Second, AASA does not require high computational complexity, so it is suitable for the WSNs to execute. Third, there are few requirements on the network topology by using AASA, such as sensor density and capabilities of sensor nodes.

### 3. Preliminaries

In this section, we first state our problem and then make assumptions and definitions of some important aspects in our work.

#### 3.1. Problem statement

For simplicity, we consider the problem of tracking a single target moving in a two-dimensional planar field covered by a wireless sensor network (WSN). The target can be a mobile object, e.g., a tank, a vehicle. The WSN is composed of randomly deployed sensor nodes and one control center (or a sink node). The sensor nodes communicate and work collaboratively for the mobile target tracking with an adjustable sensing interval, while the sink node gathers the information sensed by the sensor nodes and forwards the tracking information to an end user who is interested in tracking the target.

Two possible constraints are the energy constraint and computational complexity constraint. We consider various metrics to measure the performance of our approach, including tracking quality, energy consumption, and network lifetime.
3.2. Assumptions

We make the following assumptions and definitions in our work.

- Assume that a WSN consists of $N$ sensor nodes and is deployed for the purpose of tracking operation. The WSN can be either homogeneous or heterogeneous. In this paper, we consider the homogeneous WSN, where all of the sensor nodes have the same capabilities and energy, which offers advantages for employing a dynamic cluster based tracking algorithm. However, we assume that our algorithm can also be employed in a heterogeneous WSN. In this case, the predicted region (PR) should be adjusted according to an adjustable communication and sensing range. Besides, only the planar area is considered in this paper; because our localization technique is not valid in spatial area. We assume that the auction mechanism and the algorithm based on it can be easily applied to the spatial situation.

- Each node knows its own location by the self-localization algorithm introduced in [40]. After the deployment of the WSN, each node computes its own location. During the tracking process, locations of nodes are regarded as a priori. To make a decision on the target location accurately, time synchronization has been achieved in the whole network [41].

- When a sensor node detects a target, it has the ability of computing the Euclidean distance between itself and the target. A multi-sensor target detection model is used to detect the target [42,23]. One of the energy-based target measurements is RSSI (received signal strength indicator), which converts the received signal energy into the distance between the target and the sensor according to the decay model [32]. We assume that such a distance estimation from a sensor to the target may have errors, since the measurement suffers from secular biases due to effects of shadowing or multi-path propagation, radio occlusions, and decalibration, as well as large unbiased errors due to measurement noise.

- We consider that each sensor has three states of operation: active, listen, and sleep [43]. Sensors in the active state (active sensors) can sense the target and communicate with the neighboring nodes. Sensors in the listening state (listen sensors) can receive any messages from the neighboring nodes. Sensors in the sleeping state (sleep sensors) can neither sense nor communicate anything. The most energy-efficient state is the sleep state. Sensors wake up at a predefined period of time and change their sleep state to a listening state to listen to the communication channel. If it receives a message about an approaching target, it becomes active.

Important notations used in this paper are given in Table 1.

### Table 1

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{pr}$</td>
<td>The radius of the predicted region (PR)</td>
</tr>
<tr>
<td>$N_{ch}$</td>
<td>The number of members in the PR</td>
</tr>
<tr>
<td>$s_i$</td>
<td>$i$th sensor node</td>
</tr>
<tr>
<td>$p_i$</td>
<td>The distance between $i$th sensor and a target</td>
</tr>
<tr>
<td>$p_{ch}$</td>
<td>The distance between a CH and the previous predicted location of the target</td>
</tr>
<tr>
<td>$e_i$</td>
<td>The residual energy of $i$th sensor</td>
</tr>
<tr>
<td>$e_{ch}$</td>
<td>The residual energy of a cluster head</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>The weighted coefficient to energy and distance</td>
</tr>
<tr>
<td>$S$</td>
<td>The number of sensor nodes that have detected the target</td>
</tr>
<tr>
<td>$PE$</td>
<td>The prediction error</td>
</tr>
<tr>
<td>$PE_{pre}$</td>
<td>The previous prediction error</td>
</tr>
<tr>
<td>$PE_{cur}$</td>
<td>The current prediction error</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>A tracking interval</td>
</tr>
</tbody>
</table>

4. Auction mechanism and localization

In order to support collaborative data processing, clusters are dynamically formed along the target’s trajectory. Most clustering algorithms consist of two phases [7,31,24,9,20,21,23]. In the first phase, the CH which consumes more energy in a tracking iteration is selected. The method of selecting the CH affects not only the energy efficiency, but also the tracking quality. Simply selecting the node that is the closest to the predicted location, without considering its residual energy, energy efficiency may be reduced. Conversely, if a node with the highest residual energy but far away from the target is selected as a CH, tracking quality cannot be guaranteed. In the second phase, the CH broadcasts a message to recruit cluster members, which faces the same problem as in the first phase.

How to choose appropriate sensor nodes for tracking is a NP-hard problem [3]. In [21], sensor nodes are chosen to form a cluster by an optimal selection function, which considers the residual energy and communication cost of nodes, but the algorithm requires that each node has the residual energy information of its neighbor nodes. However, the communication cost caused by exchanging residual energy information is considerable.

The concept description. We present an “auction mechanism” to form a cluster ahead of a target as it moves. In each iteration of tracking operation, a CH in the current cluster computes a predicted region (PR) using a prediction method, where the target is likely to move. At the end of each iteration, the current CH issues an auction message to activate the nodes around the PR. Then, the CH acts as the auctioneer and the nodes in the PR act as bidders. Each bidder evaluates the tracking task and responds to the CH with a bid. The CH ranks the bids to choose appropriate sensor nodes for tracking. The sensor node with the highest bid is selected as the next CH, and the other appropriate nodes are chosen to be the members of a predicted cluster.

**Detailed process of the auction mechanism.** In the beginning of tracking, a cluster and a CH are formed by an existing algorithm [7]. Then, during the whole tracking operation, our auction mechanism is employed to form clusters, where each cluster is formed before the target moving toward the region of the cluster. The CH predicts the next location of the target by a prediction method and broadcasts a message $msg\_auction$ to its neighbor nodes. Each sensor node, denoted by $s_i$, receives the message and calculates the distance $p_i$ between itself and the predicted location. Then, the CH acts as the auctioneer and the nodes matching condition $p_i \leq R_{pr}$ act as the bidders. Each bidder evaluates the tracking task and responds to the CH with a bid. The CH ranks the bids to choose appropriate sensor nodes for tracking. The sensor node with the highest bid is selected as the next CH, and the other appropriate sensor nodes are chosen to be the members of the next cluster. The bid is calculated by (1) as follows:

$$f(e_i, p_i) = \alpha \cdot \left( \frac{e_i}{e_{ch}} \right) + (1 - \alpha) \cdot \left( \frac{p_{ch}}{p_i} \right)$$

where $p_{ch}$ is the distance between the CH and the previous predicted location of the target, $e_{ch}$ is the residual energy of the CH, $e_i$ is the residual energy of $s_i$, $p_i$ is the distance between $s_i$ and the predicted location of the target, and $\alpha$ is the weighted coefficient to energy and distance. A high value of $\alpha$ implies that residual energy of a sensor node is considered more important than the distance between the node and the predicted location. The result is the decreasing of localization accuracy, because the farther the node is from the target, the lower the localization accuracy is. Contrarily, a low value of $\alpha$ may cause the selection of nodes with short distance but low residual energy, which causes early deaths of some nodes and shortens the network lifetime. Because the unit of energy and distance is incomparable, simply adding of them is not a feasible
metric for the selection of sensor nodes. Here, we adopt relative concept in function $f(e_i, p_i)$. Instead of adding $e_i$ and $p_i$ directly, we first compare them with $e_{th}$ and $p_{th}$ respectively. The quotients are comparable, and the unit is one now. Apparently, the higher the value of $f(e_i, p_i)$ is, the more appropriate $s_i$ is.

### 4.1. Prediction method

Based on an approximate distance measurement using the RSSI method, we use a prediction method to estimate the next location of the target based on the current and $n$-th ($n = 1, 2, \ldots$) previous locations. If one wishes, all of the previous locations can be stored with a little space overhead, but they can be used to improve the prediction in a real-world scenario. For improving energy saving, only the nodes in the PR are activated and the rest of the nodes remain in the sleeping state. To simplify the calculation, we use a linear prediction method, which has sufficient precision for most applications. Given the present detected location $L_i(x_i, y_i)$ and previous detected location $L_{i-1}(x_{i-1}, y_{i-1})$ of the target, we can estimate the target’s speed $v$ and the direction $\theta$ [43] as:

$$v = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \quad (2)$$

$$\theta = \cos^{-1} \frac{x_i - x_{i-1}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \quad (3)$$

Then, we can calculate the predicted location of the target after a given time $t$ by:

$$\begin{align*}
x_{i+1} &= x_i + vt \cos \theta \\
y_{i+1} &= y_i + vt \sin \theta.
\end{align*} \quad (4)$$

In a real scenario, observations of acoustic signal processes are disturbed, due to obstacles and noise. The approximate distance between the target and a detecting sensor depends on the target’s approximate location measurement by the sensor. Thus, approximate location information is estimated with an adjustment on error covariance, $C_e$.

We characterize the approximate target localization (described in the next subsection) by using a covariance bound that is similar to the formulation of the Cramér–Rao lower bound (CRLB) of the variance [44]. To estimate the target location, it is necessary to define an information measuring utility, denoted by $\phi(\cdot)$. CRLB is defined as the inverse of the Fisher information matrix (FIM) [45,44]. We can derive the utility based on the covariance, $c_i$, of the distribution. As explained in [22], the maximization of the determinant of the target location is equivalent to the entropy-based selection approach introduced in [45] for Gaussian measurement errors and measurement equations. The determinant, $\text{det}(c_i)$, is proportional to the size of the region [45]. The information utility function for this approximation is chosen as $\phi(\cdot) = -\text{det}(c_i)$. Hence, the FIM for the target location can be estimated. More details can be found in [44,46].

Furthermore, to be more precise, it can be shown that $t$’s next location obeys a two-dimensional Gaussian distribution with $(x_i + 1; y_i + 1)$ as the mean. For $t$’s speed, we assume that it differs within a range $[0, v_{\text{max}}]$. The accuracy of the “prediction” is very important for the PR calculation and the quality of tracking. Note that in each location, $L_i$ is estimated by the median of several observations. We also adopt higher order prediction, which predicts the $n$th location information based on previous $(n - 1)$th approximate locations for accuracy. We consider finding an approximate prediction of the target rather than using very accurate prediction. We think that once the radius of the PR is identified by a number of sensors, the target should be inside the PR. Achieving an approximate location of the target after the CRLB process is expected to be enough for the PR calculation.

### 4.2. Improved trilateration method

We can obtain approximate target locations considering the above method, however, we still need to localize the target at a location. We allow the sensor nodes to localize the target in collaboration, since the accuracy of localization is very important for tracking. Approximate localization information is expected in this work to provide good tracking quality (e.g., by decreasing the missing rate of the target), but also prolongs the network lifetime. If there is a target missing in PR, the sensor nodes near the region need to actively communicate and compute to locate the target. In some cases, if the target is not detected in a given tracking interval, the network may need to revert to the initial state to relocate the target. These processes cause significant energy consumption in the network. We rework on the existing trilateration localization method. The original method is shown in Fig. 1.

In Fig. 1, the shaded inner circle stands for a sensor node. With the sensor nodes as the centers and the corresponding detected distances as the radii, we draw three circles. The circles intersect at the target’s location whose coordinate can be worked out by relation (6):

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2, \quad (i = 1, 2, 3). \quad (6)$$

In this relation, $(x, y)$ is the coordinate of the target, $(x_i, y_i)$ is the coordinate of sensor node $i$, and $d_i$ is the distance between $s_i$ and the target. However, Fig. 1 is the ideal situation, which ignores the detected noise. In practice, (6) usually has no real solution due to the detection errors. We present the practical situations in Figs. 2 and 3.

In Fig. 2, any two circles have two intersections that determine a straight line. Then, we have three straight lines altogether. We choose any two of them randomly and let the intersection be the target’s location. In Fig. 3, if at least one circle has no intersection with the other two circles, we cannot guarantee to estimate the target’s location as in Fig. 2. In this case, after receiving the distance measurement messages from two selected nodes, the CH usually calculates the current location of the moving target. In some cases of tracking, the CH receives the data from its members and forwards the data to the sink; however, it may not be the third node in three formed relations in the Trilateration algorithm.
case, the original algorithm also does not work well. This results in detection errors in the target tracking. In addition, in a practical WSN deployment, sensing noise and obstacles are inevitable. As a result, using the trilateration algorithm directly in a target tracking is infeasible.

For simplicity, in order to solve these problems, the center of the triangle determined by the three sensor nodes can be regarded as the approximate location of the target. However, in order to keep the localization accuracy as high as possible, we let weighted means of the three nodes’ locations be the target’s location, according to formulas (7)-(9):

\[ w_i = \frac{1}{d_i}, \quad (i = 1, 2, 3) \]

\[ x = \frac{\sum_{j=1}^{3} 1/d_j}{\sum_{j=1}^{3} 1/d_j} \]

\[ y = \frac{\sum_{j=1}^{3} w_i \cdot x_j}{\sum_{j=1}^{3} w_i} \]

Because of the weighted means, if one of the three nodes does not participate in localization, or the node fails during the localizations, the remaining two nodes are enabled to provide the localization information. Since we need an approximate location of the target inside a certain PR, such the absence of the third node does not bring the event of target missing in our work. We have three situations in terms of the number of participating nodes in the localization algorithm. Let S be the number of nodes that have detected the target. We summarize the situations as follows:

(a) If \( S = 1 \), the unique sensor node’s location is considered as the target’s location. If there are no neighbors discovered, there is a hole or obstacle (so that a node cannot connect the neighboring nodes), or all of its neighboring nodes are failed/dead, only this node needs to localize the target.

(b) If \( S = 2 \), the weighted means of the two nodes’ is considered as the target’s location. If there are no more neighbors discovered, there is a hole or obstacle (so that the two nodes cannot connect other neighboring nodes), or all of their neighboring nodes are failed/dead, only these two nodes need to localize the target.

(c) If \( S \geq 3 \), we choose three sensor nodes randomly and work out the target’s location by distinguishing the cases shown in Figs. 2 and 3. In some cases, more than three nodes may participate in the localization. If more than 3 nodes participate in the localization, it does not affect the localization accuracy, but the energy-efficiency.

5. Auction-based adaptive sensor activation algorithm

In this section, we first give the basic idea of the proposed AASA, and then explain each component of the algorithm. Finally, the overall consecutive procedure of the algorithm is described.
5.2. Adaptive tracking time interval

Tracking interval is the time interval between two consecutive tracking iterations. Tracking quality and energy efficiency depend deeply on the tracking interval. There is no doubt that lessening the tracking interval causes the missing rate of the target to decrease, but the cost is the increasing energy consumption due to the frequent operations and status switching of sensor nodes. In most of the existing tracking algorithms, tracking interval is a fixed value. We think that the quality of target tracking can greatly be affected by the fixed interval, if one of the following reasons happens during tracking.

(1) The target is faster than its normal speed.
(2) There is a hole that the target is lost inside.
(3) There is an obstacle; the nodes in the current cluster fail to detect the target and calculate the PR.
(4) The target stays idle (no movement is detected) at some point in time.

However, we believe that it should be related to the velocity of the target. High velocity requires a short tracking interval to capture the target, and low velocity allows a long tracking interval for energy saving. In order to save more energy while keeping the required quality, we adjust the tracking interval dynamically according to the instantaneous velocity of the target. The strategy is given as follows:

$$\Delta t = \frac{S_i}{|v|}$$  \hspace{1cm} (13)

where $\Delta t$ is the tracking interval, $v$ is the instantaneous velocity of the target and $S_i$ is the predetermined value of distance that the target may move during $\Delta t$.

By using (13), we shorten the tracking interval when the instantaneous velocity of the target is high, which helps avoid the situation of target missing; otherwise, we enlarge the tracking interval to save energy.

5.3. Algorithm design

The details of adaptive sensor activation are described by Algorithm 1. In Algorithm 1, some global variables are firstly described, then, the pseudo-code of the algorithm is given.

5.3.1. Algorithm description

Our proposed algorithm, AASA, considers the WSN as a homogeneous network, in which all of the sensor nodes have the same communication range and energy. For convenience of narration, we first explain some terms in Tables 1 and 2, which are used in the algorithm description.

At the beginning, all of the sensor nodes in the network are in the sleeping state. After receiving a command from the base station (or the sink) for a tracking operation, all the nodes are woken up at a predefined time and turn into the active state. Those who can detect the target form a cluster to track the target; others that do not detect the target turn into the sleep state after a given period of time.

The specific steps of AASA are described as follows:

(1) The CH adjusts $R_{pr}$, $N_{mb}$ and $\Delta t$.
(2) The CH calculates the PL via a prediction method, and then broadcasts a message msg_auction to its neighboring nodes.
(3) Each node $s_i$ that receives the message msg_auction calculates the distance $p_i$ between itself and PL. If $p_i \leq R_{pr}$, $s_i$ evaluates the tracking task by (1), and then responds with a message msg_bid to the CH.
(4) The CH receives the msg_bid messages and ranks the bids to choose appropriate sensor nodes for tracking. The number of chosen nodes is equal to or less than $N_{mb}$. If the number of bidders is less than $N_{mb}$, all of the bidders are chosen; otherwise, the number of node chosen is equal to $N_{mb}$. The sensor node with the highest bid is selected as the Next_CH, and the other chosen nodes are the members of the next cluster.
(5) The CH sends a message msg_head to the Next_CH, and sends msg_member messages to the other selected members. When the CH has finished its work in the current tracking iteration, it turns to sleeping state.
(6) After an interval $\Delta t$, the Next_CH becomes CH. All of the cluster members begin to detect the target.
(7) If a node $s_i$ detects the target, it sends a message msg_detect to the CH. The CH calculates the target’s location by the improved trilateration localization method (see Table 2).

The proposed algorithm repeats the above steps and forwards the locations of the target to the sink node. No matter how highly efficient an algorithm is, missing the target sometimes is inevitable, so the failure recovery mechanism is necessary for a tracking algorithm. To be coherent, we did not state the failure recovery in the algorithm steps above. However, a two-step failure recovery mechanism is applied to the tracking algorithm. If the CH did not receive any msg_bid message or no sensor node detected the target, the CH first informs all of its neighbor nodes to take part in the tracking task. If the target is still missed, the timer in the sink will expire and all the nodes of the network begin to sense the target further. We need to mention that the AASA algorithm is a fully distributed algorithm. In the algorithm, there is no intervention between the sink and nodes in the network during clustering and no need to connect the sink for every single tracking information forwarding. The nodes in the network are set to compute the target location information locally. A subset of nodes in a cluster finds another subset of nodes in a cluster in the predicted region. The CH only connects to the sink when it needs to forward the tracking information to the sink, or there is an obvious failure that the nodes in the current and predicted cluster cannot find the target in two consecutive tracking periods of time.

5.3.2. Computational complexity

In designing the AASA algorithm, we assume that the communication capability of sensor nodes (such as MICAz, Imote2) is constrained. The computational complexity levels of the AASA algorithm under the measurement noise process case are of the order $O(k N_{mb} M)$ to $O(k N_{mb} M^2)$, where $k$ is the number of targets, $N_{mb} \leq M$ is the number of nodes activated in a PR, and $M(\leq N)$ is the number of nodes in the region around the PR that is communicated to construct the PR. In this work, we consider $k = 1$. Specifically, the computational complexity involves $\sum_{i=1}^{M} \left( \sum_{j=i+1}^{M} (kl + (N_{mb} - i)) \right)$ iterative steps.

6. Simulation studies

In this section, the performance of our proposed algorithm is evaluated through simulations. We use the OMNet++ simulation
tool [47]. We first describe simulation settings and give the performance metrics. Then, we evaluate the system performance under different network and parameter settings.

The monitoring surveillance area is 200 m × 200 m, where 600 sensor nodes are randomly deployed. To compute the energy consumption and the network lifetime, we use a well-accepted energy model for packet transmission [48] in our evaluation. The initial energy on all the nodes is 20 J. Thesensing energy model for packet transmission[48] in our evaluation. The consumption and the network lifetime, we use a well-accepted

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field size</td>
<td>200 m × 200 m</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>600</td>
</tr>
<tr>
<td>Initial energy</td>
<td>20 mJ</td>
</tr>
<tr>
<td>Sensing energy</td>
<td>50 µJ</td>
</tr>
<tr>
<td>Receiving energy</td>
<td>50 µJ</td>
</tr>
<tr>
<td>Computation energy</td>
<td>5 µJ</td>
</tr>
<tr>
<td>Sensing range</td>
<td>15 m</td>
</tr>
<tr>
<td>Communication range</td>
<td>30 m</td>
</tr>
<tr>
<td>Speed of the target</td>
<td>(0–10) m/s</td>
</tr>
<tr>
<td>Maximal acceleration</td>
<td>3 m/s²</td>
</tr>
</tbody>
</table>

In the following, we list some related metrics to observe the performance of the proposed algorithm.

Average predicted radius (APR): The average of $R_{pr}$ of all the tracking iterations. $R_{pr}$ is adaptively modified based on $PE$, we also calculate the average value of $R_{pr}$.

Average member number (AMN): The average value of $N_{mb}$ of all the tracking iterations.

Average sensor number in PR (ASNPR): The average number of valid sensor nodes in the predicted region. Valid sensor nodes are the nodes that still have energy. Dead nodes are invalid nodes. At time goes on, more and more sensor nodes become invalid. We use this metric to evaluate how many valid sensor nodes are in the predicted region.

Average tracking sensor number (ATSN): The average number of sensor nodes that practically participate in tracking in each iteration. The proposed algorithm chooses $N_{mb}$ sensors from all of the sensors in PR to track the target. The real number of chosen nodes is equal to or less than $N_{mb}$, which is determined by the number of valid sensor nodes in PR.

Average predicted error (APE): This is the average of predicted errors of all the tracking iterations. It reflects the tracking quality of an algorithm.

Average tracking error (ATE): This is the average of tracking errors of all the tracking iterations. It is the most primitive metric for evaluating the tracking quality. Since we cannot get the tracking error in practical applications, this metric is available only in simulations.

Average tracking interval (ATI): This is the average of tracking intervals of all the tracking iterations. By means of this metric, we know the running situation of adaptive tracking interval.

We run the simulations until either the whole period of tracking over or 75% of all sensor nodes are dead. All of the results are averaged over 50 runs for high confidence. To observe the benefit of the AASA, we compare it with two existing approaches, namely, the prediction-based clustering algorithm (PBCA) [7] and dynamic energy management mechanism based on dynamic adaptive clustering with intra-cluster optimal routing (DACIOR) [23]. We analyze the results in the following subsections.

6.1. Study of $\alpha$ in (1)

In (1), $\alpha$ is a weighted coefficient to the remaining energy of a sensor node and the distance from the target. In this section, we study the influences of $\alpha$ on the performance of the proposed algorithm. Five representative values of $\alpha$ are chosen for the simulations, which are 0.0, 0.3, 0.5, 0.7 and 1.0. All the user parameters are listed in Table 4, and the results are shown in Figs. 4, 5 and Table 5.
Fig. 4 shows the number of dead nodes as time goes on when $\alpha$ varies from 0 to 1. From the figure, we summarize the simulation results as follows: the network lifetime using AASA algorithm is about one hour; only a few nodes die during the early twenty minutes; most nodes die in the last ten minutes. Therefore, we use 5 min as the observation interval in all of the simulations. In order to observe the results clearly, we only show the number of dead nodes from 5th to 10th observations as shown in Fig. 4.

As shown in Fig. 4, the number of dead nodes is inversely proportional to the value of $\alpha$. When $\alpha$ is close to 0, the distance parameter is more important than the residual energy in (1), which causes the unbalance energy consumption and early death of some nodes with low energy. The higher value of $\alpha$ implies that distant nodes with high energy can be selected to track the target sometimes, while the closest nodes with low energy remain in the sleeping state. However when $\alpha$ gets close to 1, detection errors and localization errors will increase greatly, which causes the high missing rate and the high PE. The high PE requires the high value of $R_{pr}$ and $N_{nb}$. As a result, the energy consumption increases. That is the reason why the number of dead nodes increases rapidly in later times when $\alpha$ is set to 1.

With the different values of $\alpha$, we provide further analysis of our simulation results in terms of energy consumption and missing rate. In Fig. 5, we count the missing rate over time. As expected, a high $\alpha$ lets sensor nodes with low energy remain longer at the cost of causing a higher missing rate. By analyzing Figs. 4 and 5, we can find out that when $\alpha$ is set to 0.3, AASA not only rescues more sensor nodes, but also reduces the missing rate.

The performance of tracking considering different metrics is summarized in Table 5. From Table 5, we have the following statistics: the average value of tracking interval is about 2.2 s, which is much longer than the fixed interval 1 s; the average number of tracking sensors is about 4; except for ATI, all the metrics are directly proportional to $\alpha$.

### 6.2. Study of $t_{\text{up}}$ in Algorithm 1

In Algorithm 1, $t_{\text{up}}$ is the maximal permitted value of the tracking interval. The influence of $t_{\text{up}}$ on the performance of the proposed algorithm is studied in this section. The need of $t_{\text{up}}$ is due to the acceleration of the target. The distance that the target moves during a tracking interval may be farther than $S_t$. Thus, we use $t_{\text{up}}$ to limit the PE to an acceptable value. A low value of $t_{\text{up}}$ provides good tracking quality, but consumes more energy from the nodes. A high value of $t_{\text{up}}$ saves more energy but the tracking quality decreases. We find a suitable value of $t_{\text{up}}$ for the proposed algorithm based on the current network settings. The given parameters are listed in Table 6, and the simulation results are shown in Figs. 6, 7, and 8.

Fig. 6 shows the number of dead nodes as time goes on when $t_{\text{up}}$ varies from 1 to 4 s. When $t_{\text{up}}$ is set to 1 s, the tracking interval loses the adaptation feature and is fixed as 1 s. From Fig. 6, we can see that high value of $t_{\text{up}}$ extends the lifetime of sensor nodes, which finally prolongs the network lifetime significantly.

The missing rate over the network lifetime is shown in Fig. 7. As to our expectation, a low value of $t_{\text{up}}$ offers a low missing rate, but the cost is the high energy consumption, which shortens the network lifetime. This means that a careful parameter settings in AASA can provide better tracking results.

Fig. 8 shows the energy consumption as time goes on. We can see that when $t_{\text{up}}$ is about 1, the energy consumption is higher than $t_{\text{up}}$ is more than 1. When $t_{\text{up}}$ is from 3 to 4, the energy consumption increases gradually. The metrics related to $t_{\text{up}}$ are summarized in Table 7, which are used when analyzing the performance on the energy consumption.

Analyzing by synthesizing Figs. 6–8 and Table 7, it is inevitable that high $t_{\text{up}}$ provides better performance in saving energy and prolonging the network lifetime. However there is an upper limit to $t_{\text{up}}$. If the value of $t_{\text{up}}$ exceeds the upper limit, the performance would not increase but decrease because of the high missing rate and high PE. In our simulations, when $t_{\text{up}}$ is set to 3 s, the proposed algorithm has the best trade-off between energy efficiency and tracking quality.

### Table 4
User settings.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{pr}$</td>
<td>$3R/4$</td>
</tr>
<tr>
<td>$R_{pr}$</td>
<td>$3R/2$</td>
</tr>
<tr>
<td>$S_t$</td>
<td>10 m</td>
</tr>
<tr>
<td>$t_{\text{low}}$</td>
<td>1 s</td>
</tr>
<tr>
<td>$t_{\text{up}}$</td>
<td>3 s</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>$0.0, 0.3, 0.5, 0.7, 1.0$</td>
</tr>
</tbody>
</table>

### Table 5
Further metrics.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>APR</th>
<th>AMN</th>
<th>ASNPR</th>
<th>ATSN</th>
<th>APE</th>
<th>ATE</th>
<th>ATI</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>14.8</td>
<td>5.1</td>
<td>6.5</td>
<td>3.8</td>
<td>6.9</td>
<td>5.7</td>
<td>2.23</td>
</tr>
<tr>
<td>0.3</td>
<td>14.7</td>
<td>5.2</td>
<td>6.5</td>
<td>3.8</td>
<td>7.1</td>
<td>5.6</td>
<td>2.25</td>
</tr>
<tr>
<td>0.5</td>
<td>14.8</td>
<td>5.3</td>
<td>6.9</td>
<td>4.0</td>
<td>7.1</td>
<td>5.7</td>
<td>2.19</td>
</tr>
<tr>
<td>0.7</td>
<td>14.9</td>
<td>5.3</td>
<td>6.9</td>
<td>4.0</td>
<td>7.2</td>
<td>5.7</td>
<td>2.21</td>
</tr>
<tr>
<td>1.0</td>
<td>15.6</td>
<td>5.8</td>
<td>7.2</td>
<td>4.3</td>
<td>8.2</td>
<td>6.4</td>
<td>2.17</td>
</tr>
</tbody>
</table>

### Table 6
User settings.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{pr}$</td>
<td>$3R/4$</td>
</tr>
<tr>
<td>$R_{pr}$</td>
<td>$3R/2$</td>
</tr>
<tr>
<td>$S_t$</td>
<td>10 m</td>
</tr>
<tr>
<td>$t_{\text{low}}$</td>
<td>1 s</td>
</tr>
<tr>
<td>$t_{\text{up}}$</td>
<td>1 s, 2 s, 3 s, 4 s</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.0, 0.3, 0.5, 0.7, 1.0</td>
</tr>
</tbody>
</table>

### Table 7
Usersettings.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{pr}$</td>
<td>$3R/4$</td>
</tr>
<tr>
<td>$R_{pr}$</td>
<td>$3R/2$</td>
</tr>
<tr>
<td>$S_t$</td>
<td>10 m</td>
</tr>
<tr>
<td>$t_{\text{low}}$</td>
<td>1 s</td>
</tr>
<tr>
<td>$t_{\text{up}}$</td>
<td>1 s, 2 s, 3 s, 4 s</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.0, 0.3, 0.5, 0.7, 1.0</td>
</tr>
</tbody>
</table>
6.3. Comparison with existing algorithms

We compare AASA with PBCA [7] and DACIOR [23]. The key point of PBCA is in (14).

\[ \text{selection}_i = \frac{\text{energy}_i}{\text{distance}_i} \]  

where \( \text{distance}_i \) is the distance between sensor node \( s_i \) and the predicted location, and \( \text{energy}_i \) is the remaining energy of \( s_i \). The CH selects these nodes that have the maximum selection parameter as tracker nodes in the next iteration. In order to evaluate the performance of the adaptive sensor activation method, we also run AASA-fixed-\( t \) with 1 s as a fixed tracking interval. In both AASA and AASA-fixed-\( t \), \( \alpha \) is set to 0.3. Here, we consider a fixed \( t \) to observe the performance of an adjustable \( t \) in AASA.

The performance comparison results are shown in Figs. 9, 10, and 11. In Fig. 9, we observe that the missing rate is 4% to 10% (on average) in several tracking intervals in the beginning of tracking in DACIOR. However, in the rest of the tracking intervals, the missing rate is about 25% more than the missing rate in AASA. In PBCA, the missing rate is more than 10% in the beginning, but it drastically increases with the times. There are several reasons for the higher increasing rate in both DACIOR and PBCA than AASA:

(1) Activation range and cluster size are constant all the time in both PBCA and DACIOR.

(2) Both DACIOR and PBCA use a fixed tracking interval.

(3) The sensor nodes need to communicate with all of the nodes around the target to construct a cluster. Thus, the number of communicated nodes becomes large.

The nodes in the inter-clusters also need to communicate frequently. It is also difficult to relocate the target in a fixed time interval if the target is lost once in both PBCA and DACIOR. We observe the event of target missing in a total of 50 simulation runs. We found that the event of target missing occurs frequently in all of simulation runs in DACIOR. It is important to mention that once the target is lost, it is not discovered by any of the sensors of the network in 5 simulation runs among the 50 simulation runs. All of these issues result in tracking delays. The frequent events of target missing and recovering from the missing events require significant energy consumption in the network. AASA outperforms both DACIOR and PBCA in terms of missing rate.

In Fig. 10, the number of dead nodes in both PBCA and DACIOR increases as the tracking delays increase. We can see that the tracking operation stops in 19 simulations out of the 50 simulations in PBCA before the given tracking interval finishes. In DACIOR, the tracking operation stops in 17 simulations. In AASA, it is 3 simulation runs in total out of 50 simulations. This implies that the whole network lifetime in both PBCA and DACIOR is lower than AASA. Among the 19 simulations for PBCA, the target missing event occurs in 11 simulations, which is about 20% of the total simulations.

Different from PBCA, AASA-fixed-\( t \) dynamically adjusts \( N_{mb} \) and \( R_{pr} \) according to current tracking quality. When the tracking quality is high, AASA-fixed-\( t \) decreases the number of Active nodes to save energy; otherwise, it increases the number of Active nodes to avoid missing the target. Meanwhile, function (1) is more flexible than (14), because we can set a suitable value for \( \alpha \) according to current network settings. In our simulations, the suitable value of \( \alpha \) is 0.3. Comparing to both PBCA and DACIOR, AASA-fixed-\( t \) prolongs the network lifetime without decreasing the tracking quality. The energy consumption is greatly affected by the tracking quality in both PBCA and DACIOR. In DACIOR, integrating the advantage of LEACH, the cluster head choosing approach is improved to form more uniform cluster distribution. The cluster size is adjustable. Although the clusters are formed in a distributed manner, the number of nodes activated in each cluster is large. In addition, resizing the cluster further also requires extra energy consumption.

Being more adaptive than AASA-fixed-\( t \), AASA also modifies the tracking interval dynamically according to the instantaneous velocity of the target. The simulation results show that AASA has better performance on decreasing energy consumption and prolonging the network lifetime compared to PBCA and DACIOR, while minimizing the missing rate.
Fig. 10. The number of dead nodes as time goes on.

Fig. 11. Energy consumption.

7. Conclusion

In this paper, we proposed a distributed algorithm about adaptive sensor node activation to monitor and track a moving target in wireless sensor networks. To balance the energy consumption in the network, an auction mechanism was introduced to the cluster formation process, which helps the sensor nodes with low energy to prolong their lifetimes. Considering the detection errors, an improved trilateration method was used to gain high localization accuracy for the target tracking. We evaluated the performance of our proposed algorithm through extensive simulations. The results showed that our algorithm can greatly decrease the energy consumption and increase the network lifetime, while ensuring the quality of tracking. This work still needs further improvements in the future. As the auction mechanism in this work, we need to find suitable values for the user parameters to get an optimal network performance by simulations. In addition, we will study the relationship between the user parameters and the network parameters (such as sensor density, communication range of sensor nodes, target’s velocity, and so on) and will set up a real-world experiment to validate our algorithm.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant Numbers 61272496 and 61272151.

References


