*Ad Hoc & Sensor Wireless Networks*, Vol. 34, pp. 191–220
Reprints available directly from the publisher
Photocopying permitted by license only


# A Lateration-localizing Algorithm for Energy-efficient Target Tracking in Wireless Sensor Networks

VINH TRAN-QUANG[1], THU NGO-QUYNH[2] AND MINHO JO[3,*]

[1]*School of Electronics and Telecommunications,*
*Hanoi University of Science and Technology, Hanoi, Vietnam*
*E-mail: vinhtq@hust.edu.vn*
[2]*School of Information and Communication Technology,*
*Hanoi University of Science and Technology, Hanoi, Vietnam*
[3]*Department of Computer and Information Science,*
*Korea University, South Korea*

*Received: November 22, 2015. Accepted: January 25, 2016.*

Target tracking systems, such as battlefield surveillance, wildlife monitoring and border security, need to meet certain real-time constraints in response to transient events, such as fast-moving targets. In addition, a tracking system also needs to address other important properties, such as energy consumption (due to limited power and the processing capability of sensor nodes) and tracking accuracy. In this work, we first propose a Lateration-localizing algorithm to estimate the trajectory of a dynamic target. Based on this Lateration scheme, we propose an energy efficient tracking method that achieves high tracking accuracy and low end-to-end delay. Finally, we present an analytical model and implement an extensive simulation to evaluate the performance of the proposed system. The results confirmed that our system associated with the simple linear estimator Lateration achieves better energy consumption and real time property compared to conventional Extended Kalman Filter while maintaining reasonable tracking accuracy.

*Keywords:* Energy efficiency, target tracking, lateration, estimation, Kalman filter, wireless sensor networks

## 1 INTRODUCTION

Wireless sensor networks (WSNs) and their applications have become one of the most promising emerging technologies in recent years. Sensor nodes

* Contact author: E-mail: minhojo@korea.ac.kr

are equipped with sensing, communication, and data processing units, which allow the sensor nodes to collect, exchange, and process information about ambient environments and to intelligently react to those environments [1]. Among the wide range of WSN applications, target tracking has received great attention from researchers in different fields. This application needs to achieve energy efficiency due to limited power, low processing capability of sensor nodes and dynamic topology of large scale networks. In addition, most tracking applications address various types of *real-time* constraints in response to the physical world. Unlike detection that aims to study discrete events [2], a target tracking system is often required to ensure continuous monitoring and to report the location of the target to subscribers (usually remote controllers) accurately and in a timely manner [3, 4]. Therefore, the most stringent criterion of target tracking is to track with low delay and high accuracy. All these issues pose as challenges to designing a target tracking system that can manage not only the energy efficiency of sensor nodes but also tracking accuracy in real-time.

Generally, there are three common procedures when designing a target tracking application, regardless of the various types of target and tracking environments. The first procedure is to localize the position of targets or sensors prior to participating in target tracking tasks. Many location techniques have been emphasized in the literature. Cricket [5] is a well-known solution based on absolute distances estimated from the TDOA (time difference of arrival) between a radio signal and an ultrasound signal. In contrast, the Received Signal Strength (RSS) technique [6] requires only an appropriate signal attenuation model. According to the target behavior, most of the previous works on target localization can be categorized into two classes, cooperative [7] and non-cooperative [8]. A cooperative target, as part of the network, can emit certain forms of physical signals that reveal its presence or report its own identification. The localization mission becomes easier, and the same technique as with sensor network localization mentioned above can be used. Unfortunately, such cases are very limited. In general, targets are not cooperative with sensors, i.e., enemy vehicles and unregistered victims in disaster areas. In these non-cooperative scenarios, the target will not cooperatively provide any information of its own existence to the network. Moreover, the targets tend to occur randomly and occasionally, and their movements are normally unpredicted. To overcome this problem, many estimation and prediction algorithms have been suggested in the literature, such as the Bayesian [9] and Kalman filtering methods [10,11]. The Kalman filter (KF) is a classical method that can estimate the original state of a target by collecting and analyzing a number of different types of sensing data.

Because the sensing information collected by sensors may be redundant, correlated, and/or inconsistent, it is desirable to have sensors collaborate on processing the data and providing a concise digest to subscribers [12, 13]. Thus, the second problem that we need to consider is the collaborative data processing among sensor nodes. This technique reduces not only the number of packets to be transported but also the probability of collision and interference in the shared media. Localized and collaborative data processing also aids in reducing the power consumed in communication activities and, hence, prolonging the lifetime of sensor networks [14]. Once a target is detected and tracked by a group of collaborative sensors, the final important activity is to concisely transmit the tracking information to subscribers [15, 16]. A main technique, introduced by tracking in large-scale networks, is *clustering*, in which the network is divided into many clusters. Within each cluster, a node is selected as a cluster head (CH) and functions as a local proxy to gather and aggregate data from group-based tracking sensors; to maintain the object's state, such as location, speed, and direction; and to report to the base station periodically.

For localizing target position, different filters (BF, KF, EKF and PF) have been widely studied before. For estimating and predicting target position, these filters require maintaining information at previous moments. Thus, it is necessary to broadcast this information to the nodes that will participate lately at the sensing area. This characteristic leads to inefficient energy consumption and degraded tracking accuracy, especially when information of previous moments cannot be maintained. In order to overcome this problem, we investigate Lateration, a simple linear energy-efficient estimator that is able to localize the target without broadcasting and maintaining information of previous moments. Based on Lateration algorithm, we design an energy efficient target tracking system that provides good tracking accuracy and low delay.

The main contributions of our work can be summarized as follows.

- Instead of implementing complicated filters for localizing target, simple energy efficient Lateration-localizing algorithm is investigated to estimate the trajectory of a dynamic target.
- We also propose a collaborative data processing algorithm and dynamic clustering scheme that cooperate with Lateration algorithm for providing energy efficiency and low end-to-end delay while maintaining good tracking performance. The energy efficiency of our system is enhanced by reducing the number of proactively awakened nodes and controlling their active time in an integrated manner with limited loss and low delay of the tracking performance. These techniques allow the system to track targets with low energy consumption, prevent certain nodes from over-consuming energy.
- To deal with the constraints on energy consumption, latency, and accuracy factors, we propose a mechanism to select a wake-up zone by

controlling user-predefined threshold and the sampling interval between two successive tracking. Taking the on-demand collaborative tracking and the dynamic wake-up zone determination into account, the proposed method achieves the energy balance on sensor nodes and prolongs the network lifetime.

The remainder of this article is organized as follows. In section 2, we briefly review the most relevant works that address localization, estimation, and collaboration techniques for target tracking WSNs. Section 3 describes the system model, target movement model, measurement model, and assumptions used in our article. Section 4 presents the conventional EKF tracking for reference and comparison. Section 5 describes the proposed algorithms. Delay analysis and its calculation are presented in Section 6. The simulation results are presented in Section 7. The final section presents the conclusion and future work.

## 2 RELATED WORK

During the past few years, tracking and monitoring applications using WSNs have attracted much attention from the research communities of many fields.

The most important mission of this application is to localize a moving target. For this purpose, an optimal or linear filtering technique used in tracking is Kalman filter (KF) [2, 10, 11]. An estimate of the state of the process is performed recursively so that each time mean of squared error is reduced while tracking a target with random trajectory. The KF assumes linear system dynamics, but for nonlinear system dynamics, the tracking performance of Kalman filter is degraded. To address this problem, researchers developed suboptimal extension of KF, namely, extended Kalman filter (EKF) [4], which is the most widely used approximate filter for tracking [4,17]. The EKF linearizes the transformation by substituting Jacobian matrices for the linear transformations of the KF equations assuming that all the transformations are quasilinear. However, the series approximations of the EKF algorithm can lead to poor representations of the nonlinear functions and probability distributions, so this filter may diverge. To overcome these drawbacks Particle Filter [4, 5] is devised for nonlinear and non-Gaussian system dynamics. In PF any probability distribution function (PDF) can be represented as a set of samples known as particles. Each particle has one set of values for the state variables, and the density of samples in the area of the state space depicts the probability of that region. Hence, in PF the posterior distribution is represented by a set of weighted samples. These are updated recursively in time using the importance sampling method [4]. For estimating and predicting target position, these filters require maintaining information at previous moments. Thus, it is necessary to broadcast this information to the nodes that will participate lately at the sensing area. This characteristic leads to inefficient energy consumption and degraded tracking accuracy, especially when information of previous moments can not be maintained.

Finally, the energy efficiency can be improved by reducing the number of nodes involved in communication. Therefore, various efforts were conducted based on sleep scheduling and collaborative target tracking [12, 18–22]. In [12], collaborative event detection and target tracking algorithms are proposed for heterogeneous wireless sensor networks to detect the presence of targets. In [23], the authors consider border nodes to detect the target when the number of nodes used for target detection is high. However, most existing efforts for proactive wake-up simply awaken all the neighboring nodes in the area where the target is expected to arrive, without any differentiation. In fact, it is sometimes unnecessary to awaken all the neighboring nodes.

Lateration [24, 25], a simple linear energy efficient estimator, is able to localize the target without requiring information about target in the past. The input of the algorithm is a set of at least three measurement vectors in two dimensional plane or at least four measurement vectors in three-dimensional space. This simple linear estimator, has been developed previously in literature but its influence on tracking application is rarely investigated. In addition, the design of a Lateration-based tracking system that assures energy efficiency, low delay and good tracking accuracy is also not realized.

## 3 SYSTEM MODEL AND ASSUMPTION

We use typical notations that $\mathcal{S}$ denotes a set, $x$ a scalar, $\mathbf{x}$ a vector, and an upper case $\mathbf{X}$ random variable/vector. The symbols $i$, $j$, and $k$ refer to the sensor node identification, target identification, and tracking time step, respectively. The estimated trajectory of a mobile target $j$ is performed at discrete time $t_j(k)$. The sampling interval $\Delta t_j(k)$ of the target $j$ at time $t_j(k)$ is the time between two successive tracking, i.e., $\Delta t_j(k) = [t_j(k+1) - t_j(k)]$.

### 3.1 Target Movement Model

In this paper, we consider 2-D constant velocity model in which the true state vector of a target at the time step $k$ is modeled with $\mathbf{x}(k) = [x(k)\ \ \dot{x}(k)\ \ y(k)\ \ \dot{y}(k)]^{\mathrm{T}}$, where $x(k)$ and $y(k)$ are the $x$- and $y$-coordinates of the target, and $\dot{x}(k)$ and $\dot{y}(k)$ are the velocities of the target along the $x$- and $y$-directions at the $k$-th time step. Consider a WSN with a set $\mathcal{S} = \{i\}_{i=1}^{L}$ of $L$ sensor nodes being randomly deployed. A dynamic target $j$ that moves according to an independent and identical distribution over time is modeled using the discrete-time white noise acceleration model described by the evolution of state sequence,

$$\mathbf{x}_j(k+1) = \mathbf{A}_j(k)\mathbf{x}_j(k) + \mathbf{w}_j(k), \tag{1}$$

$$\mathbf{A}_j(k) = \begin{bmatrix} 1 & \Delta t_j(k) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t_j(k) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

$\mathbf{A}_j(k)$ denotes a state-transition matrix, and $\mathbf{w}_j(k)$, where $\mathbf{E}[\mathbf{w}_j(k)\mathbf{w}_j(k)^{\mathrm{T}}] = \mathbf{Q}_j(k)$, denotes the process noise that models the target velocity variations and is assumed to possess a zero-mean white Gaussian distribution with $\mathbf{Q}_j(k)$ covariance matrix,

$$\mathbf{Q}_j(k) = q_j \begin{bmatrix} \frac{\Delta t_j^3(k)}{3} & \frac{\Delta t_j^2(k)}{2} & 0 & 0 \\ \frac{\Delta t_j^2(k)}{2} & \Delta t_j(k) & 0 & 0 \\ 0 & 0 & \frac{\Delta t_j^3(k)}{3} & \frac{\Delta t_j^2(k)}{2} \\ 0 & 0 & \frac{\Delta t_j^2(k)}{2} & \Delta t_j(k) \end{bmatrix} \tag{3}$$

where $q_j$ is a known scalar representing the intensity of the process noise. The goal is to estimate the target state $\mathbf{x}_j(k)$ at the time step $t_j(k)$.

### 3.2 Node Deployment Model

We assume that sensor nodes are deployed uniformly over a two-dimensional area [26], that the number of nodes is sufficient to adequately monitor the field, and that each sensor node i has a unique identifier (such as a MAC address). We further assume that each node knows its location using various positioning methods proposed in [5, 6, 27]. For simplicity, we assume that every sensor node knows its location in terms of $(x, y)$ coordinates.

### 3.3 Sensor Measurement Model

The sensor nodes are capable of measuring acoustic, magnetic, spatial, or seismic data, and the wireless network enables the performance of distributed computations over the gathered data by individual sensor nodes. Source localization using acoustic signals requires little data to be transmitted via the wireless communication channel. Furthermore, acoustic signal intensity measurement is more efficient to detect the presence of a target and to track this target than using other modalities such as seismic or radar. For simplicity, in this article, we assume that each sensor node is able to measure the target range using an acoustic signal that is emitted from the target [11, 28, 29]. The target $j$ is assumed to be an isotropic sound source where its acoustic power intensity received by sensor node $i$ at time $t_j(k)$ can be calculated according to the following measurement model:

$$P_i(k) = \frac{S_j(k)}{R_i^{\alpha}[k,\, j]} \tag{4}$$

where $S_j(k)$ is the emitted acoustic density from the target $j$, which is assumed to be known, $R_i^{\alpha}[k,\, j]$ is the noisy geometric distance between the sensor $i$ and the target $j$, and $\alpha$ is the attenuation decay factor typically between 2 and 5 according to the environment and atmospheric conditions [28].

By measuring $P_i[k]$, therefore, $R_i^{\alpha}[k,\, j]$ can be calculated using Equation (4). Suppose that there is a group $S_{g(j)} = \{1, 2, ..., m\}$ of $m$ activated sensors to track the target $j$, the measurement model that relates the noisy target measurements with the true target state at time $t_j(k)$ is given by the following equation,

$$\mathbf{z}_j(k) = \mathbf{h}_j[k, \mathbf{x}_j(k)] + \mathbf{v}_j(k) \tag{5}$$

where $\mathbf{v}_j(k)$ is the measurement noise that is assumed a zero-mean white Gaussian distribution with $\mathbf{R}_j(k)$ covariance matrix, where $\mathbf{E}[\mathbf{v}_j(k)\mathbf{v}_j(k)^{\mathrm{T}}] = \mathbf{R}_j(k)$, and the noisy target measurement vector $\mathbf{z}_j(k)$ and measurement function (generally nonlinear) $\mathbf{h}_j[k, \mathbf{x}_j(k)]$ are given as

$$\mathbf{z}_j(k) = [R_1[k,\, j] \quad R_2[k,\, j] \quad ... \quad R_m[k,\, j]]^{\mathrm{T}}, \tag{6}$$

$$\mathbf{h}_j[k, \mathbf{x}_j(k)] = \begin{bmatrix} \sqrt{[L_j(k) - L_1]^{\mathrm{T}}[L_j(k) - L_1]} \\ \sqrt{[L_j(k) - L_2]^{\mathrm{T}}[L_j(k) - L_2]} \\ \cdots \\ \sqrt{[L_j(k) - L_m]^{\mathrm{T}}[L_j(k) - L_m]} \end{bmatrix}, \tag{7}$$

where $\mathbf{L}_j(k) = \left[x_j(k) \quad y_j(k)\right]^{\mathrm{T}}$ and $\mathbf{L}_i = [x_i \quad y_i]^{\mathrm{T}}$ are the true target location vector at time $t_j(k)$ and the sensor node $i$'s location vector, respectively.

### 3.4 Radio Model

In this article, we use the simple first-order radio model described in [30], where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics. The most basic model of channel propagation typically found in WSN environments involves direct or free-space waves. It is observed that the radio signal strength falls as some power $\alpha$ of the distance between the transmitter and the receiver, which is called the power-distance gradient or the path-loss gradient [31]. For relatively short distances, the propagation loss can be modeled as inversely proportional to $d^2$, whereas for longer distances, the propagation loss can be modeled as inversely proportional to $d^4$. Power control can be used to invert this loss by setting the power amplifier to ensure a certain power at the receiver. Thus, to transmit a $b$-bit message over a distance $d$, the radio expends

$$\begin{aligned} E_{Tx}(b,d) &= E_{Tx-elec}(b) + E_{Tx-amp}(b,d) \\ &= \begin{cases} b\varepsilon_{elec} + b\varepsilon_{fs}d^2,\ d < d_0 \\ b\varepsilon_{elec} + b\varepsilon_{mp}d^4,\ d \geq d_0, \end{cases} \end{aligned} \tag{8}$$

and to receive this message, the radio expends

$$E_{Rx}(b) = E_{Rx-elec}(b) = b\varepsilon_{elec}. \tag{9}$$

where $d_0$ is the reference distance $\varepsilon_{elec}$ is energy for the transceiver electronics, $\varepsilon_{fs}$ is energy for transmission in the free space model, and $\varepsilon_{mp}$ is energy for transmission in the multi-path model. The residual energy of the sensor node after one data communication is calculated, as follows:

$$E_i(k) = E_{current} - (E_{Tx} + E_{Rx}). \tag{10}$$

The electronics energy $\varepsilon_{elec}$ depends on factors such as the digital coding, modulation, and filtering of the signal before sending data to the transmit amplifier. For the experiments described in this article, we set the energy dissipated per bit in the transceiver electronics to $\varepsilon_{elec} = 50nJ/bit$, $\varepsilon_{fs} = 10\,pJ/bit/m^2$, and $\varepsilon_{mp} = 0.0013\,pJ/bit/m^4$.

## 4 EXTENDED KALMAN FILTER (EKF)

The target state shown above is discretely and non-linearly observed via the nonlinear measurement model. We use the EKF, a mathematical model based on the linearization of the non-linearity in the dynamic and/or measurement models to minimize the mean square error of the estimated parameters. The predicted target state at time $t_j(k+1)$ can be calculated from state equation (Equation (1)), if we have collected the measurements $\mathbf{z}_j(k+1)$ up to and including $t_j(k+1)$. The target state estimation at any $t_j(k+1)$ as well as target tracking can be implemented as follows.

## 4.1 State Estimation

The EKF relies on a state estimation $\hat{\mathbf{x}}_j(k|k)$—the estimate of $\mathbf{x}_j(k)$—to obtain the state predictor $\hat{\mathbf{x}}_j(k+1|k)$—the estimate of $\mathbf{x}_j(k+1)$—the given measurement $\mathbf{z}_j(k+1)$ as

$$\hat{\mathbf{x}}_j(k+1|k) = \mathbf{A}_j(k)\hat{\mathbf{x}}_j(k|k). \tag{11}$$

The predicted measurement is calculated as follows:

$$\hat{\mathbf{z}}_j(k+1|k) = \mathbf{h}_j[k+1, \hat{\mathbf{x}}_j(k+1|k)]. \tag{12}$$

In the update stage, the measurement residual, which is the difference between the actual and predicted measurements, is calculated as follows:

$$\mathbf{r}_j(k+1) = \mathbf{z}_j(k+1) - \hat{\mathbf{z}}_j(k+1|k). \tag{13}$$

The update state estimation is given as

$$\hat{\mathbf{x}}_j(k+1|k+1) = \hat{\mathbf{x}}_j(k+1|k) + \mathbf{K}_j(k+1)\mathbf{r}_j(k+1) \tag{14}$$

where the filter gain $\mathbf{K}_j(k+1)$ is defined below in the state covariance estimation.

## 4.2 State Covariance Estimation

The EKF relies on an error covariance matrix $\mathbf{P}_j(k|k)$—the covariance matrix of $\mathbf{x}_j(k)$—to find the covariance matrix of the state innovation (the prediction error) $\mathbf{P}_j(k+1|k)$—the covariance matrix of $\mathbf{x}_j(k+1)$—the given measurement $\mathbf{z}_j(k+1)$. The state prediction covariance is described as follows:

$$\mathbf{P}_j(k+1|k) = \mathbf{A}_j(k)\mathbf{P}_j(k|k)\mathbf{A}_j^{\mathrm{T}}(k) + \mathbf{Q}_j(k). \tag{15}$$

The measurement prediction or residual covariance matrix is described as follows:

$$\mathbf{S}_j(k+1) = \mathbf{H}_j(k+1)\mathbf{P}_j(k+1|k)\mathbf{H}_j^{\mathrm{T}}(k+1) + \mathbf{R}_j(k+1). \tag{16}$$

Finally, the updated covariance matrix of the state at time $t_j(k+1)$ is calculated by the following equation,

$$\mathbf{P}_j(k+1|k+1) = \mathbf{P}_j(k+1|k) - \mathbf{K}_j(k+1)\mathbf{S}_j(k+1)\mathbf{K}_j^{\mathrm{T}}(k+1) \tag{17}$$

where $\mathbf{K}_j(k+1)$ is the weighting matrix or the filter gain from the estimation equation (17),

$$\mathbf{K}_j(k+1) = \mathbf{P}_j(k+1|k)\mathbf{H}_j^{\mathrm{T}}(k+1)\mathbf{S}_j^{-1}(k+1). \tag{18}$$

The evaluation of Jacobian matrix of $\mathbf{h}_j$ at $\mathbf{x}_j(k+1) = \hat{\mathbf{x}}_j(k+1|k)$ is calculated as follows:

$$\mathbf{H}_j(k+1) = \frac{\partial \mathbf{h}_j[k+1, \mathbf{x}_j(k+1)]}{\partial \mathbf{x}_j(k+1)} = [\mathbf{H}_{ij}] \tag{19}$$

with $1 \leq i \leq m$, $1 \leq j \leq 4$. The detailed derivation of the EKF mathematical equations is given in [11, 32].

## 5 REAL-TIME ENERGY-EFFICIENT TARGET TRACKING

### 5.1 Target Detection

In a typical target tracking scenario, a moving target tends to occur randomly and occasionally. Continuous activation of sensor nodes in this case wastes a significant amount of energy, and therefore all sensor nodes are designed to cyclically turn off their radio transmission or to go into an inactive stage in the absence of a target. Each node is scheduled to remain in the inactive stage for a period $T$. A duty cycle parameter $\beta$ ($0 \leq \beta \leq 1$) is defined as the probability of a node being awaken within the period $T$. To reduce missed detections, each node chooses a wakeup point $T_{start}$ that is independently and evenly distributed between 0 and $T$. The node performs sensing during a time interval $\beta T$ and then returns to the inactive stage until the next duty cycle. Within the duty cycle, if the acoustic power intensity emitted from a target at a particular acoustic sensor exceeds the predetermined threshold, the sensor determines that a target is in the network. Then, the detected sensor broadcasts an alarm message to proactively awaken neighboring nodes to prepare for the approaching target.

### 5.2 Dynamic Clustering

Measurements $R_i^{\alpha}[k, j]$ collected from sensors $i$ may be redundant, and the quality of the measurements degrades as the distance between the target and the sensor increases. Without a doubt, these physical limitations mean that one sensor alone cannot finish the task of target tracking. Furthermore, in a typical target tracking scenario, a moving target tends to occur randomly and occasionally, and only the sensors near the target can detect and perform sensing on it. It is desirable to make sensors collaborate with each other for the processing of measurements. Therefore, to facilitate collaborative data processing in the target tracking task, we design a dynamic distributed cluster architecture to select a cluster and a CH as follows. If the measured signal of sensors exceeds the predefined threshold, resulting from evaluation via Equation (4), the sensor node will be fully activated. It then broadcasts a cluster forming invitation message to its neighbors during time $t_1 = timeStep$:

$$\textbf{REQ\_CLUSTER}\{ID_i, E_i(k), P_i(k)\} \tag{20}$$

where $ID_i$ denotes the sensor identifier, $E_i(k)$ is the value of residual energy of sensor $i$, and $P_i(k)$ is the sensed data resulting from Equation (4). When a node receives the REQ_CLUSTER messages from another node, it sets timer $t_2 = T_{timeout}$. When timer $t_2$ expires, it executes the cost factor function, Equation (21), to determine whether it is a CH or cluster member.

$$F_{CH-selection-factor}(i) = E_i(k) \times P_i(k). \tag{21}$$

The node that has the maximum value of the cost function sets itself as a CH, stores the IDs of its neighboring members, and then creates a TDMA schedule to assign a time slot to each node. According to this TDMA schedule, all members can transmit their sensed data to the CH in order to avoid collision. Algorithm 1 presents the pseudo-code of this mechanism.

By using the dynamic clustering, each node can make its decision without any centralized control. With the dynamic target-based clustering method, clusters are formed based on only when and where targets appear in the sensing field. Sensor nodes outside the expected target approach area stay in the sleeping state to conserve energy. The clusters involved in the tracking task are called the active cluster. Nodes in the active cluster sense the target and report the sensed data to the CH for localization estimation. After receiving a sufficient number of measurements from sensors, the CH applies the proposed localization method to estimate the location of the target, predict the next approaching target's area, extract the signature of the target, and send a report to the subscribers. If the expected area is outside its cluster, the CH node creates and tasks new active clusters. The procedure to create new clusters repeats to continue the tracking task. This collaborative data processing strategy within each cluster results in minimizing the control information exchanged between sensors while still maintaining reasonable tracking accuracy and real-time achievement.

**Algorithm 1** Create a cluster and select the CH

**Require:** measurement parameter $R_i^{\alpha}[k, j]$ from sensor $i$
**Ensure:** tracking cluster $S_{g(j)}$ and CH

$S_{g(j)} \leftarrow \emptyset,\ timer \leftarrow \beta,\ CH \leftarrow 0$
calculate $P_i(k)$ based-on Equation (4)
**for** each sensor $i$ at time step $k$ **do**
  **if** $P_i(k) \geq threshold$ **then**
    $S_{g(j)} \leftarrow \{i\}$
    broadcast $P_i(k)$ and $E_i(k)$ to one-hop
    **if** *timer=0* and $max\{P_i(k), E_i(k)\}$ **then**
      $CH \leftarrow \{i\}$
      CH broadcast TDMA
    **else**
      **if** *timeslot* == *TS* **then**
        send data to CH
      **else**
        wait for time slot TS to send data to CH
      **end if**
    **end if**
  **end if**
**end for**

### 5.3 Target Localization

The lateration [25] is a classical algorithm that can be used when we want to calculate the coordinates of the target if the positions of several sensors, as well as the distance to them, are known a priori. In this method, $(x(k), y(k))$ are the coordinates of the target at time $k$, and $(x_i, y_i)$ are the coordinates of sensor $i$. Let $r_i(k)$ be the precise distance, with no noise, between the unknown positions and each sensor at time $k$, with $i = 1, 2, \ldots, n$, $n \geq 3$. From the Pythagorean Theorem, we obtain a system of three equations as follows:

$$r_i(k) = \sqrt{(x_i - x(k))^2 + (y_i - y(k))^2}. \tag{22}$$

If the coordinates of at least three sensors are known a priori ($i = 1, 2, 3$), the $(x(k), y(k))$ coordinate of the target can be obtained through the following equations:

$$\begin{cases} (x_1 - x(k))^2 + (y_1 - y(k))^2 = r_1^2(k) \\ (x_2 - x(k))^2 + (y_2 - y(k))^2 = r_2^2(k) \\ (x_3 - x(k))^2 + (y_3 - y(k))^2 = r_3^2(k) \end{cases} \tag{23}$$

The equations of (23) can be rewritten in matrix form as follows:

$$2\begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix}\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} (r_1^2(k) - r_3^2(k)) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ \left(r_2^2(k) - r_3^2(k)\right) - \left(x_2^2 - x_3^2\right) - (y_2^2 - y_3^2) \end{bmatrix} \tag{24}$$

In reality, as noise exists in the environment, the range measurements are not perfect and contain some errors. To take this measurement error into account, we can represent the imprecise range measurement in the form $\tilde{r}_i(k) = r_i(k) + v_i(k)$, where $r_i(k)$ represents a perfect measurement and $v_i(k)$ is an unknown measurement error at time $k$. We assume that the noisy error follows a Gaussian distribution with a mean of zero and variance of $\delta_\nu(k)$, i.e., $v_i(k) \sim (0, \delta_\nu^2(k))$. The Gaussian noise model is adopted in many sensed data processing works, such as [33]. In real applications, the parameter $\delta_\nu^2(k)$ is obtained from training data. We consider sensor range measurements and noise to be independently and identically distributed across sensor nodes. In this case, the trilateration does not always give an exact solution. In fact, it is possible that three circles intersect at more than three points. In this case, more than three sensors are used, and that results in an overdetermined system of equations to take into account individual measurement errors. The multilateration for target position evaluation is shown in Figure 1 to address

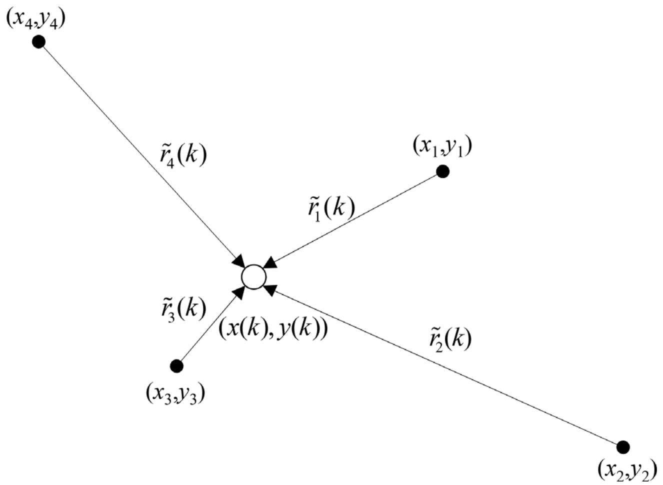


FIGURE 1
Multilateration for determining target's location.

the problem of noisy range measurements in which the shaped circles, for finding the unknown position, do not intersect at all. This overdetermined system of equations can be written in matrix form as follows [24, 25]:

$$2\begin{bmatrix} x_n - x_1 & y_n - y_1 \\ \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix}\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} (\tilde{r}_1^2(k) - \tilde{r}_n^2(k)) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ \vdots \\ \left(\tilde{r}_{n-1}^2(k) - \tilde{r}_n^2(k)\right) - \left(x_{n-1}^2 - x_n^2\right) - (y_{n-1}^2 - y_n^2) \end{bmatrix} \tag{25}$$

For this overdetermined system of equations, a solution can be computed by minimizing the mean square error $||\mathbf{Ax} - \mathbf{b}||^2$, where $0.5\mathbf{A}$ is the left-hand matrix, $\mathbf{x} = (x(k), y(k))$ is the vector describing the unknown target and $\mathbf{b}$ is the right-hand vector of the equation. The term $||\mathbf{Ax} - \mathbf{b}||^2$ is the Euclidean norm squared of the residual $\mathbf{Ax} - \mathbf{b}$, which is equal to:

$$\begin{aligned} ||\mathbf{Ax} - \mathbf{b}||^2 &= (\mathbf{Ax} - \mathbf{b})^{\mathrm{T}}(\mathbf{Ax} - \mathbf{b}) \\ &= \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{Ax} - 2\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\mathbf{b}, \end{aligned} \tag{26}$$

where $(\mathbf{Ax} - \mathbf{b})^{\mathrm{T}}$ is the transpose of the matrix $(\mathbf{Ax} - \mathbf{b})$. Regarding the above equation as a function of $\mathbf{x}$, minimizing this expression means that the gradient of this function should be set equal to zero, which leads to $2\mathbf{A}^{\mathrm{T}}\mathbf{Ax} - 2\mathbf{A}^{\mathrm{T}}\mathbf{b} = 0$. This is equivalent to $\mathbf{A}^{\mathrm{T}}\mathbf{Ax} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$. This equation is called the normal equation for the linear square problem. The coordinates of the target are estimated via the following equations:

$$\mathbf{x} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}, \tag{27}$$

where $(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}$ is the inverse matrix of $(\mathbf{A}^{\mathrm{T}}\mathbf{A})$.

### 5.4 Collaborative Target Tracking Algorithm

We assume that all sensor nodes are homogeneous and static. The coordinates of sensors are known by using GPS or anchor nodes [6, 34]. The objectives of our research are (1) to reduce energy consumption by managing the behavior of sensor nodes, (2) to increase the quality of tracking through target prediction and node pre-activation, and (3) to prolong the network lifetime by balancing the lifetime of all the sensor nodes in the network. The following techniques are proposed to achieve these objectives.

### *5.4.1 Select Tracking Group and Group Leader*

We design an on-demand distributed algorithm to select a tracking group and a group leader, where the sensed attributes of a target can be used to form the group. Every node can make its decision without any centralized control. With the dynamic target-based grouping method, groups are formed based on when and where targets appear in the sensing field. This method can save energy because only a portion of the network becomes active in response to the target.

Initially, all nodes in the sensor network are in listening mode to save battery power. If the measured signal exceeds the user-predefined threshold, resulted from evaluating Equation (4), the sensor node will be activated. It broadcasts a forming cluster invitation message. This message includes its residual energy level $E_i(k)$, the geometric distance to the target $R_i^\alpha[k, j]$, and a $timer_j$. If a nearby sensor node receives the message, it should reply with a joint cluster message including its energy level and the geometric distance to the target if it is available. When the $timer_j$ is expired, each sensor node evaluates the following equation to determine itself as a group leader (having high residual energy and located near the target) or group member.

$$leader_j \xleftarrow{\max} \arg_i\{R_i^\alpha[k, j],\ E_i(k)\} \tag{28}$$

The pseudo code of the algorithm to select a tracking group and a group leader is presented in Algorithm 2.

### *5.4.2 Tracking and Updating*

Most target tracking systems require continuous monitoring, i.e., there always exists nodes that can track the target along its trajectory with low

**Algorithm 2** Select Tracking Group and Group Leader

**Require:** measurement from sensor $i$
**Ensure:** tracking group $S_{g(j)}$ and $leader_j$
$S_{g(j)} \leftarrow \emptyset, timer_j \leftarrow \beta, leader_j \leftarrow 0$
**for** each sensor node $i$ at time step $k$ **do**
  **if** $\|P_i[k]\| \geq threhold_j$ **then**
    $S_{g(j)} \leftarrow \{i\}$ //sensor $i$ is in tracking group
    broadcast $R_i^\alpha[k, j]$ and $E_i(k)$ to one-hop neighbors
    **if** $timer_j = 0$ **and** $max\{\arg_i\{R_i^\alpha[k, j],\ E_i(k)\}\}$ **then**
      $leader_j \leftarrow \{i\}$ //sensor $i$ is leader
    **end if**
  **else**
    sensor $i$ $\leftarrow$ inactive node
  **end if**
**end for**

tracking delay and report its state to a base station in response to transient events, such as fast-moving targets. While the real-time performance is a major concern in these applications, it should be reconciled with other important system properties, such as energy consumption and accuracy. In reality, incorrect predictions unavoidably occur due to unexpected directional changes of non-cooperative moving targets at random times. Hence, prediction errors cannot completely be avoided even if an accurate prediction method is applied. When errors occur, the tracking process is no longer functional because the sensor nodes located near the moving target were likely to turn off their sensors and radios to conserve energy. In the worst case, the failure of sensor nodes can occur at any time due to power depletion or severe natural environments; hence, a moving target is able to slip away from the estimated future location without detection. In this case, the effort both to conserve energy and to obtain acceptable performance leads to failure.

To keep the real-time and energy-efficient requirements in mind and to address prediction error in the target localization process, we use the wake-up zone control mechanism as follows. Given the measurements from the cluster members, the CH uses the lateration to estimate the target state, such as the location, speed, and direction, and then report to the base station. The CH also predicts the next expected target approach area. If the measurement signal falls below the user-predefined threshold, the CH sends an activation message to awaken the next suitable sensors corresponding to the estimated approaching target. We denote the area containing the pre-activated nodes as a wake-up zone. The pseudo-code of the algorithm for tracking and updating is presented in Algorithm 3.

Finally, to evaluate the performance of the algorithms, we use the ARPEES [30] as routing protocols to transmit tracking data from the CH to the base station in a multi-hop relay manner.

**Algorithm 3** Tracking and Updating

**Require:** measurement from $S_g : \{\tilde{r}_1(k),\ \tilde{r}_2(k) \ldots \tilde{r}_n(k)\}$ and sensors' coordinates $(x_i, y_i)$ with $i = 1, 2 \ldots n\ \ (n \geq 3)$

**Ensure:** obtain trajectory of target

1: **for** each time step $k$ **do**
2: estimate trajectory of target based on Equation (27)
3: send state of target to BS
4: update $S_g : \{\tilde{r}_1(k),\ \tilde{r}_2(k) \ldots \tilde{r}_n(k)\}$
5: **if** $\|\tilde{r}_i\ (k)\| \leq$ threshold $+\ \varepsilon$ **then**
6: send message to trigger the next tracking group (Algorithm 2)
7: **end if**
8: **end for**

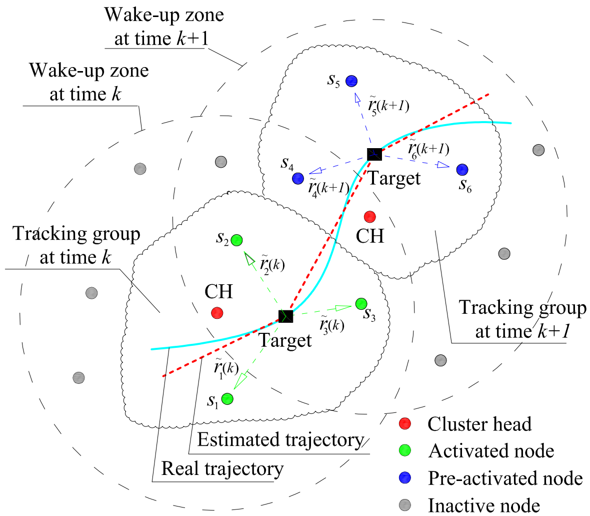


FIGURE 2
An example of collaborative tracking and dynamic wake-up zone determination using the Lateration.

### 5.4.3 *Algorithm Adjustment*

Due to the resource constraints, an accurate but yet complicated prediction mechanism is hard to implement to the real hardware. Instead, the concept of the *wake-up zone* [35] has often been used to approach the energy-efficient tracking problem. In reality, the effective size of the wake-up zone is usually influenced by many factors, such as the target velocity, elapsed time, acceptable accuracy rate, and so on. To deal with constraints on energy consumption, latency, and accuracy factors, we consider the trade-off relationship between the wake-up zone decided by controlling user-predefined threshold ($threshold_j$) and the sampling interval ($\Delta t_j(k)$) between two successive tracking. Taking the on-demand collaborative tracking and the dynamic wake-up zone determination into account, the proposed method achieves the energy balance on sensor nodes. The Figure 2 shows an example of collaborative tracking and dynamic wake-up zone determination using the Lateration. In this mechanism, when the target moving toward the edge of a current cluster head's coverage, the cluster head will send an activation message to activate suitable sensor nodes along the estimated target's trajectory.

## 6 DELAY ANALYSIS AND CALCULATION

In reality, most tracking applications, such as battlefield surveillance and disaster and emergency response, address various types of real-time constraints in response to the physical world. For example, surveillance may require a sensor node to detect and classify a fast moving target within 1 second before it moves out of the sensing range [3]. In other words, for tracking applications, a delay bound is required. In this section, we analyze all components that contribute to the delay. The analysis is verified in the extensive simulation presented in Section 7.

The operation of our tracking system includes five phases. Each phase contributes a sub-delay component to the total end-to-end delay:

**Initial activation phase:** An initial activation delay, denoted as $T_{initial}$, is the period before the first active node starts to sense the incoming target. If sensors are activated by the CH to ensure tracking coverage, $T_{initial} = 0$. Otherwise, $T_{initial} = T_{start}$.

**Initial Target Detection phase:** After the initial activation, it requires a certain delay, denoted as $T_{detect}$, for the first active node to confirm the detection. Specifically, $T_{detect}$ is the period of time for a detection algorithm to recognize a target signature.

**Cluster formation phase:** Once awake, all nodes that detect the same target join a cluster. Each cluster is represented by a CH to maintain operation of the cluster. We denote $T_{cluster}$ as the time value for the cluster formation.

$$\begin{aligned} T_{cluster} &= \sum(T_{BO} + T_{CCA}) \\ &+ T_{frame}(\text{STATUS\_INFO}) \\ &+ T_{LIFS} + T_{timeout}, \end{aligned} \tag{29}$$

where the $\sum(T_{BO} + T_{CCA})$ is the count for the last sensor that sent the STATUS_INFO message, $T_{BO}$ is the back-off period in seconds, $T_{CCA}$ is the clear channel assessment delay, $T_{frame}(x)$ is the transmission time for a payload of $x$, and $T_{LIFS}$ is the IFS time used when the packet size is more than 18 bytes. The component $T_{timeout}$ is the maximum value of wait time for receiving the STATUS_INFO message, which can be calculated as:

$$\begin{aligned} T_{timeout} &= \left(2^{aMaxBE} - 1\right) \times T_{BOslots} \\ &+ T_{frame}(\text{STATUS\_INFO}) \\ &+ T_{CCA} + T_{detection}. \end{aligned} \tag{30}$$

**Data aggregation phase:** The CH gathers measurements from its member and executes tracking algorithm, Algorithm 3, to track the target and then report to the BS. We use $T_{process}$ to denote the cluster aggregation delay, which is the time required to collect measurements from member nodes and process the estimation.

$$\begin{aligned} T_{process} &= (T_{BO} + T_{CCA} + T_{fame}(\text{TDMA}) \\ &+ T_{LIFS})_{TDMA} \\ &+ maxTS \times T_{frame}(\text{DATA\_TO\_CH}). \end{aligned} \tag{31}$$

**Relay selection phase:** After the cluster formation phase, all information from non-CH members is forwarded to the CH, and this CH, in turn, forwards it to the BS. To do this, the CH selects a relay node by broadcasting request relay messages to its neighbors, waits for acknowledgments from all candidates and selects a relay node according to the relay node function shown in [30]. This phase obviously causes a sub-delay to the total end-to-end latency of the system, called $T_{relay}$.

$$\begin{aligned} T_{relay} &= (T_{BO} + T_{CCA} + T_{LIFS} \\ &+ T_{frame}(\text{RELAY\_REQ})) \\ &+ (\textstyle\sum(T_{BO} + T_{CCA}) + T_{timeout} \\ &+ T_{fame}(\text{ACK\_RELAY}) + T_{LIFS}), \end{aligned} \tag{32}$$

where the first component is the delay for broadcasting and processing the RELAY_REQ message at the CH, the second component is the delay for receiving and processing the ACK_RELAY message of the last relay candidate that sent ACK_RELAY, and the component $T_{timeout}$ is the maximum value of wait time for receiving the ACK_RELAY message, which can be calculated as:

$$\begin{aligned} T_{timeout} &= \left(2^{aMaxBE} - 1\right) \times T_{BOslots} \\ &+ T_{CCA} + T_{frame}(\text{ACK\_RELAY}). \end{aligned} \tag{33}$$

**Report phase:** After the relay node selection phase, the CH reports the target state to the BS. The delay of this process is denoted as $T_{report}$. It is also important to recall that two processes (relay node selection and reporting) can be repeated many times until the CH reaches the BS. In other words, the sum of $T_{relay}$ and $T_{report}$ is multiplied several times based on the number of hops along the path towards the BS. The delay for reporting the target state to a relay node or the BS is:

$$\begin{aligned} T_{report} &= T_{BO} + T_{CCA} \\ &+ T_{frame}(x) + T_{LIFS}. \end{aligned} \tag{34}$$

Now we can evaluate the total end-to-end delay in our system as:

$$\begin{aligned} T_{e2e} &= T_{cluster} + T_{process} \\ &+ N \times (T_{relay} + T_{report}), \end{aligned} \tag{35}$$

where $N$ is the number of hops to reach the BS. Among these sub-delays, we realize that the initial delay can be considered a physical element of the sensor's hardware for switching from an inactive to an active state. More concretely, it depends mainly on the type of sensors used in the system. The second delay element is the target detection delay and, as previously analyzed, it consists of the hardware response delay, discrete sampling delay, and other components. In other words, these components depend on the node's type. Therefore, we focus on the calculation and evaluation of the last three sub-delays: cluster formation, relay selection, and transmission. Those sub-delays reflect the performance of our tracking system.

## 7 EXPERIMENTS

The main goal of the experiments described in this article is to measure and compare the energy consumption, delay, and lifetime of the proposed system as a function of varying input parameters and the algorithms we have implemented.

### 7.1 Experimental Setup

We developed an extensive computer simulation, implemented in the OMNeT++ simulator [36], to evaluate the performance of the proposed method. In the simulation, we use the same input parameters as mentioned in [12,30]. A total of 950 sensor nodes are uniformly and randomly deployed within a square field of $640 \times 540$ square meters. The base station is located at $(320, 540)$. A screen-shot of the simulation is shown in Figure 3. Each sensor node is provided an energy level of 100 mJ at the start. We assume that one target is moving in the sensing field with an average speed of 10 m/s. Measurement noise is $5 - 10\%$ of the sensing range. Its real trajectory, which is pre-determined as shown in Figure 4(a), is divided into steps. In fact, the acoustic intensity measured by sensor nodes depends on the sensitivity of the acoustic microphone. The detection threshold of each acoustic microphone is adjusted so that the acoustic detection range is 20, 25, or 30 meters. For simplicity, we use the sensing range of the sensors to simulate the threshold of measurement. Therefore, we can vary both the sensing range of sensor nodes and the sampling interval to control the update estimation sequence.

| **Parameter** | **Value** |
|---|---|
| Network surveillance | $640 \times 540\,m$ |
| Number of nodes | 950 |
| Initial energy ($E_{ini}$) | 0.1 $Joule$ |
| Data Packet Size | 256 $bytes$ |
| Broadcast Packet Size | 30 $bytes$ |
| Sensing Range | *20, 25, and 30 m* |
| Max transmission range | 75 $m$ |
| Average target speed | *10 m/s* |
| Energy for the transceiver electronics ($\varepsilon_{elec}$) | 50 $nJ/bit$ |
| Energy for transmission in the free space model ($\varepsilon_{fs}$) | 10 $pJ/bit/m^2$ |
| Energy for transmission in the multi-path model ($\varepsilon_{mp}$) | 0.0013 $pJ/bit/m^4$ |

TABLE 1
Simulation parameters

Each simulation is repeated 100 times with different network topologies, and the final results present their average.

In this section, we examine and evaluate the performance of our system using the lateration localizing scheme against a traditional solution, EKF, which utilizes the ARPEES [30] routing protocol. The simulation finishes

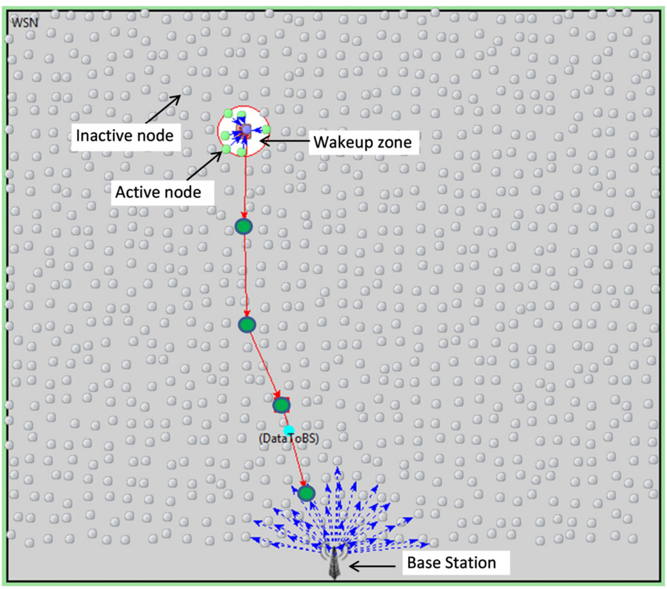


FIGURE 3
A simulation screenshot in OMNet++.

when all sensor nodes that have a shorter distance to the BS than their max transmission range run out of energy.

## 7.2 Results Analysis

### *7.2.1 Tracking Accuracy*

Figure 4 plots the true trajectory of the target and its estimated trajectories received by the BS from our tracking system using lateration and EKF methods. We evaluate the accuracy of the tracked trajectories of the moving target when we assign different values for the sampling interval parameter, $\Delta t$ = 1, 2/3, or 1/3 seconds. The results are shown in Figure 4 (b), (c), and (d). As plotted in these figures, when the sampling interval has the smallest value, the tracking system achieves the best accuracy. Depending on the quality-of-service requirement of the application and network resources, users can vary these parameters to acquire better performance in the aspect of network lifetime or tracking accuracy.

For an accurate evaluation, we compare the true trajectory and the tracking trajectories by lateration and the EKF method. This metric is presented

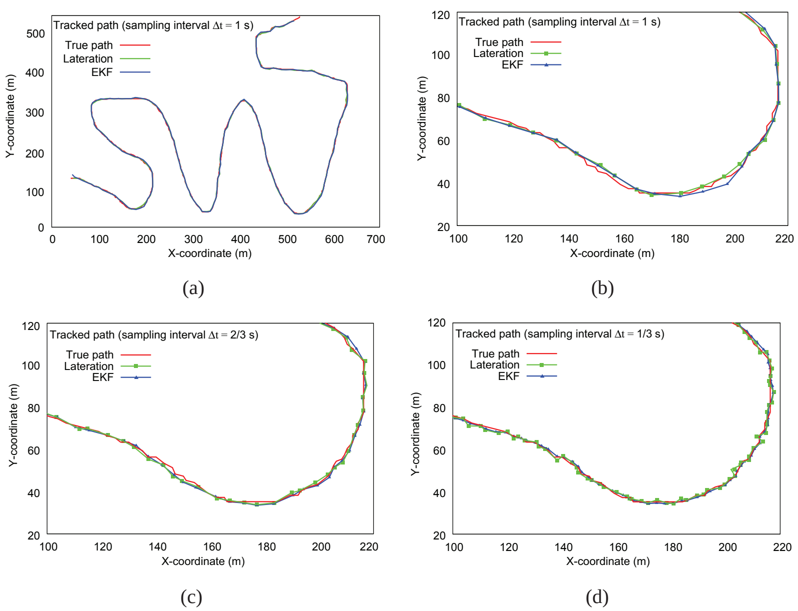


FIGURE 4
Comparison of the true trajectory of the target and its estimated positions based on lateration and EKF at the sampling interval (a) $\Delta t = 1$ second. Subfigures (b), (c), and (d) are the first part of the trajectory with $\Delta t$ = 1, 2/3, and 1/3 seconds, respectively.

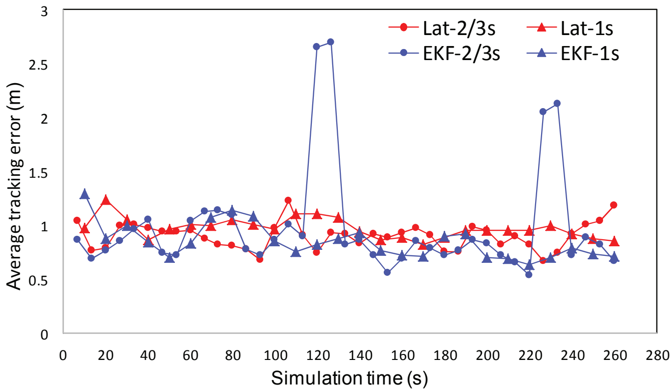


FIGURE 5
Comparison of the average tracking error between lateration and the EKF method.

in Figure 5. From this figure, we can easily find that the EKF prediction method has better quality than the lateration. In other words, the trajectory plotted by EKF is more identical to the true trajectory than that by the lateration. However, in EKF, when the CH does not have information of the target position in the previous step, the position of the CH is utilized for approximating the target position, and that causes a sudden change in the tracking accuracy. This sudden mutation can be observed in this figure. In conclusion, EKF achieves better accuracy than lateration but can cause a sudden change, whereas lateration can still achieve a reasonable tracking accuracy with less energy consumption than EKF does. This is because lateration does not require information of the target position in the previous step.

We use the cumulative probability functions (CDF) of the distance error for measuring the precision of a system. In Figure 6, the EKF has a localization precision of 65% within 0.9 m (the CDF of distance error of 0.9 m is 0.65). Lateration has a localization precision of 55% within 0.9 m. In this case, the EKF is better than Lateration if the distance error requirement is within 1 m. But, if the distance error is within 1.5 m, the precision of Lateration is better than EKF because of its higher precision (90% vs. 80%). This property is consistent with the original objective of this study is to balance energy efficiency and tracking accuracy.

### *7.2.2 Energy Efficiency*

Another important metric is the residual energy of sensors because there is always a trade-off between tracking performance and energy consumption. In this subsection, we evaluate the variation of total residual energy in both cases: using Lateration and EKF filter.

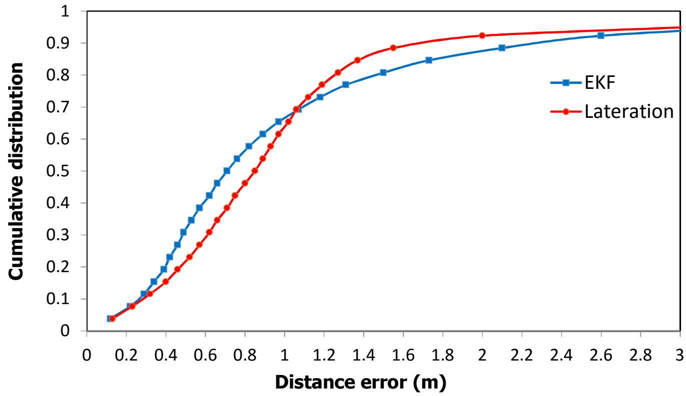


FIGURE 6
The CDF of localization error for tracking algorithms using EKF and Lateration.

Figure 7 shows a comparison of the total residual energy of sensor nodes with varying sampling intervals of 1, 1/3 and 2/3 seconds. Clearly, a longer sampling interval results in a higher residual energy level of sensor nodes while tracking performance becomes lower. Alternately, with the same sampling interval, the lateration method helps sensors to consume less energy than the EKF method.

This important feature can be explained as follows:

- EKF is not only an estimation but also a prediction method. After estimating and predicting the location of target by using EKF, it is necessary to broadcast the state estimation and the updated covariance matrix of the

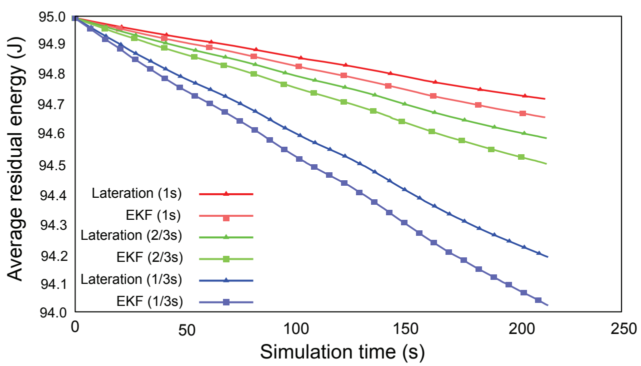


FIGURE 7
Total residual energy of Lateration and EKF methods as a function of simulation time.

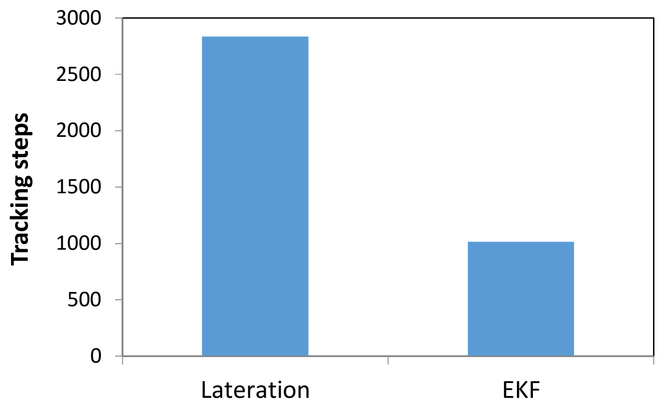


FIGURE 8
Comparison of network lifetime with different tracking methods.

state $(P(k \mid k), \hat{x}(k|k))$ to one-hop neighbors. This information will be used by next CH in order to predict and estimate the target's information.

- Lateration needs only the sensed information of sensors for estimating the location of the target at the present time and transmits this location information towards BS. Unlike EKF, it is not necessary to broadcast any further information to next CH. That is why the tracking system using Lateration consumes less energy than the EKF method.

To evaluate network lifetime, we simulate a network with 30 mJ for each node as the initial energy and stop the simulation when no nodes are able to transmit directly to the BS. The network lifetime is presented in Figure 8. This figure plots the number of tracked points that can be transmitted to the BS. From the figure, the lateration method has a longer network lifetime than EKF.

### *7.2.3 End-to-end Delay*

To assure the real-time property of our system, the tracking tasks of target detection, target localization, and target state reporting need to be completed within each sampling interval. In other words, the end-to-end delay needs to be smaller than the sampling interval. The sampling interval is set at 1 second for a target speed of 10 m/s. The sensing range of sensor nodes is 25 m, the active period $T = T_{sampling} = 1$ second, and the duty cycle $\beta = 0.4$. For simplicity, we ignore the delay caused by hardware response. From Figure 9, we realize that the end-to-end delay of the system in two cases is always smaller than 0.5 seconds, and thus the real-time property of the system is assured.

### 7.2.4 System Flexibility

In this subsection, we measure the end-to-end delay and target loss ratio if the target speed is 10 m/s, 15 m/s, and 20 m/s while the sensing range is 20 m, 25m and 30 m. The experiments are executed 10 times with different network topologies. Figure 10 plots the end-to-end delay as a function of target speed with the tracking algorithm using lateration. We see that, if the sensing range is increased, the end-to-end delay also increases because the number of nodes in the cluster increases. However, when the sensing range is 20 m, the target loss ratio increases significantly (21%) because measurement readings from the awake node do not provide enough target signatures for performance of the estimation algorithm. For a better target lost ratio, the intensity of nodes needs to be augmented, or it is necessary to deploy more than 950 nodes in an area of $640 \times 540\ m^2$.

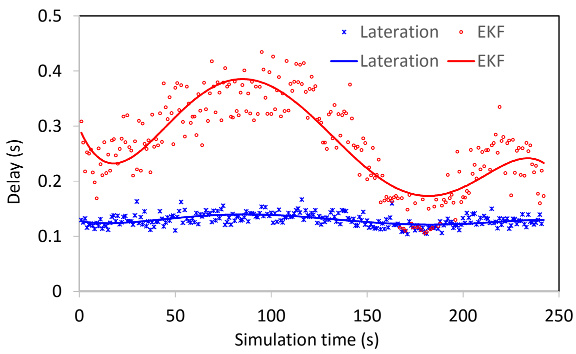


FIGURE 9
Comparison of the end-to-end delay of the tracking system with different tracking methods.

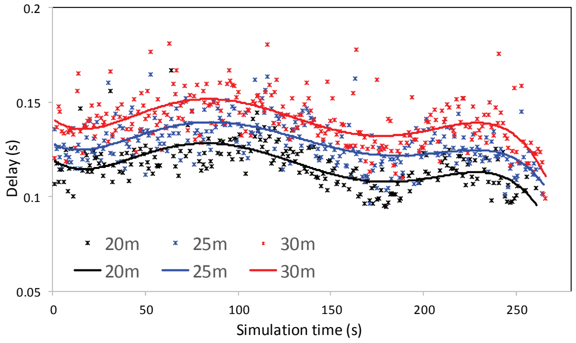


FIGURE 10
Comparison of the end-to-end delay as a function of sensing range.

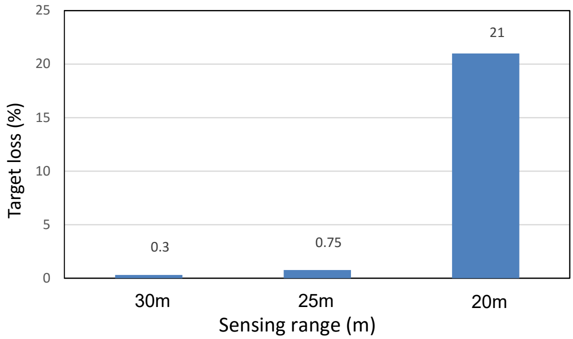


FIGURE 11
Comparison of the target loss with different sensing ranges.

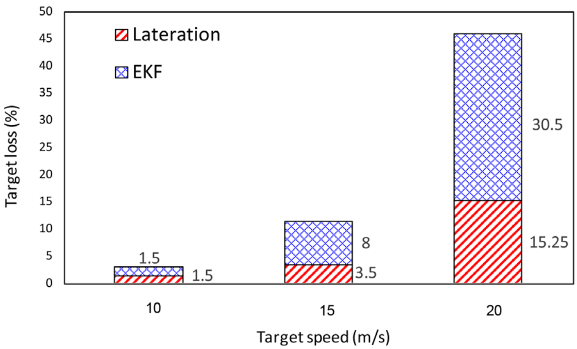


FIGURE 12
Comparison of the target loss with different target speeds.

Next, we evaluate the target lost ratio. In this experiment, the dynamic target movement is according to an independent and identical distribution over a time period of 200 seconds with the speed varying among 10, 15, and 20 m/s as shown in Figure 11 and 12. The lost rates are 1.5%, 3.5%, and 15.25% in the first case and 1.5%, 8%, and 30.5% in the second case. We can conclude that the lateration method achieves a lower target lost ratio than EKF.

## 8 CONCLUSIONS AND FUTURE WORK

In this article, we proposed a new method using lateration for estimating the target trajectory. We also designed the selecting tracking group and tracking algorithm based on the estimated trajectory of the target. Simulation results confirmed that our new method can track the target in real-time with reasonable accuracy while achieving improved energy consumption compared to EKF estimation. In a future work, we will address the problem of multi-target tracking in WSNs.

## ACKNOWLEDGMENTS

This research is funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2012.06.

## REFERENCES

[1] Y. Touati, H. Aoudia, and A. Ali-Cherif, "Intelligent wheelchair localization in wireless sensor network environment: A fuzzy logic approach," in *5th IEEE Int'l Conf. Intelligent Systems (IS 2010)*, pp. 408–413, July 2010.

[2] G. Wittenburg, N. Dziengel, C. Wartenburger, and J. Schiller, "A system for distributed event detection in wireless sensor networks," in *9th ACM/IEEE Int'l Conf. Information Processing in Sensor Networks*, IPSN '10, pp. 94–104, 2010.

[3] X. Chen, N. C. Rowe, J. Wu, and K. Xiong, "Improving the localization accuracy of targets by using their spatial-temporal relationships in wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 72, pp. 1008–1018, Aug. 2012.

[4] A. Hassani, A. Bertrand, and M. Moonen, "Tracking of a rotating object in a wireless sensor network using fuzzy based adaptive imm filter," in *Commun. and Vehicular Technology in the Benelux (SCVT), 2012 IEEE 19th Symposium on*, pp. 1–6, 2012.

[5] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *2nd Int'l Conf. Embedded Networked Sensor Systems, SenSys '04*, pp. 50–61, Sept. 2004.

[6] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *IEEE INFOCOM 2000*, pp. 775–784, March 2000.

[7] K. Qureshi and A. Abdullah, "Localization-based system challenges in vehicular ad hoc networks: Survey," *Smart Computing Review*, vol. 4, no. 6, pp. 515–528, 2014.

[8] S.-J. Kim and D.-J. Park, "A self-calibrated localization system using chirp spread spectrum in a wireless sensor network," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 2, pp. 253–270, 2013.

[9] D. Fox, J. Hightower, L. Liao, and D. Schulz, "Bayesian filtering for location estimation," *IEEE Pervasive Comput.*, vol. 2, pp. 24–33, July 2003.

[10] V. Tran-Quang, H. Nguyen-Khanh, and T. Ngo-Quynh, "Target tracking system using lateration estimation method in wireless sensor networks," in *5th Int'l Conf. Ubiquitous and Future Networks (ICUFN 2013)*, pp. 264–269, July 2013.

[11] Y. Hamouda and C. Phillips, "Adaptive sampling for energy-efficient collaborative multi-target tracking in wireless sensor networks," *IET Wireless Sensor Syst.*, vol. 1, pp. 15–25, March 2011.

[12] V. Tran-Quang, P. Nguyen-Huu, and T. Miyoshi, “A collaborative target tracking algorithm considering energy constraint in wsn,” in *19th Int’l Conf. Software, Telecommun. and Comput. Netw. (SoftCOM2011)*, pp. 1–5, Sept. 2011.

[13] K. Kim and H. Y. Youn, “An energy-efficient and scalable routing protocol for distributed wireless sensor networks,” *Ad Hoc & Sensor Wireless Networks*, vol. 29, no. 1-4, pp. 195–212, 2015.

[14] B. Jiang, B. Ravindran, and H. Cho, “Probability-based prediction and sleep scheduling for energy-efficient target tracking in sensor networks,” *IEEE Trans. Mobile Comput.*, vol. 12, pp. 735–747, Apr. 2013.

[15] H. Tang and H.-Z. Wang, “A multi-chain based hierarchical topology control algorithm for wireless sensor networks,” *KSII Transactions on Internet and Information Systems*, vol. 9, no. 9, pp. 3468–3495, 2015.

[16] S. Chatterjee, A. Das, and J. Sing, “An enhanced access control scheme in wireless sensor networks,” *Ad Hoc & Sensor Wireless Networks*, vol. 21, no. 1-2, pp. 121–149, 2014.

[17] F. Chan and H. So, “Efficient weighted multidimensional scaling for wireless sensor network localization,” *Signal Process., IEEE Trans.*, vol. 57, no. 11, pp. 4548–4553, 2009.

[18] J. Feng, B. Lian, and H. Zhao, “Adaptive energy optimization for object tracking in wireless sensor network,” *KSII Transactions on Internet and Information Systems*, vol. 9, no. 4, pp. 1359–1375, 2015.

[19] G. Han, Y. Dong, H. Guo, L. Shu, and D. Wu, “Cross-layer optimized routing in wireless sensor networks with duty-cycle and energy harvesting,” *Wireless Communications and Mobile Computing*, vol. 15, no. 16, pp. 1957–1981, 2015.

[20] D. Li, J. Lee, and S. Park, “Minimum gradient forwarding for wireless sensor networks,” *Smart Computing Review*, vol. 4, no. 5, pp. 418–425, 2014.

[21] G. Zheng, L. Fu, J. Li, and M. Li, “A large-scale multi-track mobile data collection mechanism for wireless sensor networks,” *KSII Transactions on Internet and Information Systems*, vol. 8, no. 3, pp. 857–872, 2014.

[22] P. Perumal, V. Uthariaraj, and V. Christo, “Wsn lifetime analysis: Intelligent uav and arc selection algorithm for energy conservation in isolated wireless sensor networks,” *KSII Transactions on Internet and Information Systems*, vol. 9, no. 3, pp. 901–920, 2015.

[23] A. Rafiei, M. Abolhasan, D. Franklin, and F. Safaei, “Boundary node selection algorithms in wsns,” in *36th IEEE Conf. Local Computer Networks*, pp. 251–254, Otc. 2011.

[24] K. Holger and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. New York: Wiley, 2005.

[25] L. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Wireless Commun. Mag.*, vol. 11, pp. 6–28, Dec. 2004.

[26] X. Liu, J. Li, and G. Kang, “A smart energy-efficiency deployment scheme for lifetime enhancement in large-scale wireless sensor networks,” *Smart Computing Review*, vol. 5, pp. 591–601, dec 2015.

[27] I. Boulanouar, S. Lohier, A. Rachedi, and G. Roussel, “Dta: Deployment and tracking algorithm in wireless multimedia sensor networks,” *Ad Hoc & Sensor Wireless Networks*, vol. 28, no. 1-2, pp. 115–135, 2015.

[28] C. Meesookho, U. Mitra, and S. Narayanan, “On energy-based acoustic source localization for sensor networks,” *IEEE Trans. Signal Process.*, vol. 56, pp. 365–377, Jan. 2008.

[29] K. Deb, S. Imtiaz, and P. Biswas, “A motion region detection and tracking method,” *Smart Computing Review*, vol. 4, no. 1, pp. 79–90, 2014.

[30] V. Tran-Quang and T. Miyoshi, "Adaptive routing protocol with energy efficiency and event clustering for wireless sensor networks," *IEICE Trans. on Commun.*, vol. E91-B, pp. 2795–2805, Sept. 2008.

[31] T. Rappaport, *Wireless Commun.: Principles and Practice*. New Jersey, USA: Prentice-Hall, Inc., 1996.

[32] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. New York: Wiley, 2001.

[33] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C.-W. Yi, "Data fusion improves the coverage of wireless sensor networks," in *15th Int'l Conf. Mobile comput. and networking (MobiCom '09)*, pp. 157–168, Sept. 2009.

[34] M. McGuire, K. Plataniotis, and A. Venetsanopoulos, "Location of mobile terminals using time measurements and survey points," *IEEE Trans. Vehicular Technol.*, vol. 52, pp. 999–1011, July 2003.

[35] S. Lee, H. Cha, and R. Ha, "Energy-aware location error handling for object tracking applications in wireless sensor networks," *Computer Commun.*, vol. 30, pp. 1443–1450, 2007.

[36] A. Varga, "Omnet++: A discrete event simulation system, ver. 4.3," in *http://www.omnetpp.org/*, 2010.