An Efficient Target Tracking Mechanism for Guaranteeing User-Defined Tracking Quality in WSNs

Cuijuan Shang, Guilin Chen, Chengchao Ji, and Chih-Yung Chang, *Member, IEEE*

***Abstract*— Target tracking is one of the most important issues in wireless sensor networks (WSNs). In a densely deployed WSN, all sensors participating in the monitoring task can provide full coverage but consume their energies inefficiently. Some approaches pursue better network lifetime at the cost of tracking accuracy, while some other mechanisms aim to achieve the best tracking accuracy without consideration of network lifetime. However, it is not necessary to achieve the best tracking accuracy, especially for those applications that users have defined their requirement of tracking quality. This paper proposes an efficient target tracking (ETT) mechanism, which dynamically wakes up sensors in a distributed manner, aiming to minimize the number of working sensors while the user-defined tracking quality can be satisfied. Performance results show that the proposed ETT outperforms existing work in terms of energy consumption of sensor nodes and the tracking quality.**

***Index Terms*— Energy conservation, quality guaranteed, target tracking, user-defined tracking quality, wireless sensor network.**

I. Introduction

Wireless sensor network (WSN) composes of many sensor nodes each of which is capable of sensing, data processing, and communication. It has been widely applied in many applications, including environment monitoring, scientific observation, and tracking [1]–[8]. Since sensor nodes are battery powered, they are constrained in energy. Boundary detection and location tracking is one of the most important issues in WSNs, aiming to detect the crossing behavior of a specific target, such as intruder or an interested moving target [2], and obtain the target's trajectory.

In literature, the target tracking issue in WSNs has received much attention. Plenty of mechanisms have been proposed to obtain the trajectory of moving targets in the monitoring area [9]. These studies can be mainly divided into two categories: the centralized approaches [10], [11] and the distributed approaches [12]–[16].

Manuscript received March 19, 2015; revised May 21, 2015; accepted May 23, 2015. Date of publication June 2, 2015; date of current version July 24, 2015. This work was supported in part by the Natural Science Foundation of Educational Government of Anhui Province under Grant KJ2011ZD06 and in part by the National Natural Science Foundation of China under Grant 61472057. The associate editor coordinating the review of this paper and approving it for publication was Prof. Okyay Kaynak.

C. Shang, G. Chen, and C. Ji are with Chuzhou University, Chuzhou 239000, China (e-mail: shangcuijuan@chzu.edu.cn; glchen@chzu.edu.cn; jcc@chzu.edu.cn).

C.-Y. Chang is with Tamkang University, Taipei 25157, Taiwan (e-mail: cychang@mail.tku.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2015.2440295

In the centralized class, most proposed tracking schemes [10], [11] assumed that the sink node is aware of the locations of all sensors. Most of existing tracking approaches are executed by sink. Therefore, the sensor nodes need to report to the sink node as soon as they detect the target events. Then the sink node makes decision associated with each reported event. The common challenge of the centralized approaches is the inefficient energy consumptions and poor real-time performance.

To improve difficulties usually found in the centralized tracking mechanisms, many distributed algorithms [12]–[16] are proposed. Most the proposed distributed approaches allow each sensor executing the tracking task locally. Upon detecting the target, the detected sensor wakes up some neighboring sensors for cooperatively estimating the target location. A handoff operation is required as a part of the proposed approach to cope with the target mobility problem. In general, the tracking accuracy is increased with the number of working sensors. However, it is still a big challenge to improve the energy efficiency while maintaining a high tracking quality.

This paper proposes an efficient target tracking mechanism (*ETT*) for a single target to dynamically wake up the minimal number of working sensors in a distributed manner while the user-defined tracking quality is satisfied. The contributions of this paper are itemized as follows.

(1) *Energy Conservation When the Target Not Entering the Monitoring Region:* The initialization phase of the proposed *ETT* only wakes up boundary sensors. All the other sensors stay in sleeping state for energy conservation. In addition, the boundary sensor that monitors the longer boundary segment has the higher priority to participate in the monitoring task to minimize the number of working boundary sensors. Most related works did not consider the situation that target does not enter the monitoring region.

(2) *Energy Conservation When the Target Entering the Monitoring Region:* The target detection phase of the proposed *ETT* aims to wake up the minimal number of the neighboring sensors of the target detected sensor. All the other sensors stay in sleeping state for energy conservation.

(3) *Satisfying the User Defined Requirement of Tracking Quality:* The proposed *ETT* guarantees the tracking quality which meets the user-defined quality requirement.

(4) *Distributed Operations and High Network Scalability:* The operations designed in the proposed *ETT* are totally distributed. This helps the tracking algorithm work even though some sensors are failure or newly deployed. As a result, the work of WSN has higher network scalability.

The remaining part of this paper is organized as follows. Section II reviews the related works. Section III presents the network environment and problem formulation of this paper. Section IV details the operations of the proposed three phases while Section V presents the architecture of the proposed *ETT* mechanism. The performance of the proposed mechanism is investigated in Section VI. Finally, Section VII offers a conclusion.

II. Related Works

In literature, a number of studies have been proposed for target tracking in wireless sensor networks. Solutions of mobile target tracking can be mainly classified into two types, including centralized [10], [11] and distributed [12]–[16] target tracking approaches.

Kim *et al.* [10] proposed the BPS-Tracking scheme to improve the weight calculation for each binary proximity sensor that detects the target and estimates the target location based on the estimated velocity. However, the method assumes that the target moves at a constant velocity on a linear trajectory. Furthermore, they only considered the sensor nodes that detect the target. Actually, the absence of detection can also provide additional implicit information which can be used to improve the tracking accuracy.

Study [11] proposed the UNS-Tracking scheme with unreliable node sequences that removes the constraint of constant velocity of the target and proposed a mobile target tracking framework. The mechanism converts the tracking problem to the problem of finding the shortest path in a graph. However, it is not suitable for WSNs since the complexity of the mechanism increases with the number of deployed sensors. Additionally, information broadcasted by tracking sensors from all regions is collected, aiming at computing the estimated trajectory. This indicates that the tracking lacks scalability and consumes time and energy for information collection. In addition, the proposed mechanism cannot guarantee a specific user-defined tracking quality.

In the past few years, some distributed computing algorithms [12]–[16] were proposed to cope with the target tracking problem. With the purpose of improving the accuracy of the mobile target tracking, Chen *et al.* [12] proposed the target tracking scheme with adaptive sensor activation (ASA) to predict the moving path of the target and then selected sensors along the predicted path to join the target tracking task. The center of gravity of working sensors, which detected the target, is regarded as the location of the target, improving the accuracy of the target's trajectory. However, only a small part of working sensors can detect the target, resulting in low energy efficiency and network lifetime.

In the two-phase sleep scheduling (TPSS) protocol [13], the monitoring region is divided into several small grids. In each grid, one sensor is chosen and responsible for detecting the target. The proposed mechanism can significantly reduce the unnecessary energy consumption. Once a working sensor detects the target, it will wake up all neighboring sensors located in the same grid to improve the tracking accuracy. However, the complexity of the algorithm increases with the number of sensors deployed in the WSN, resulting in high tracking delay.

To further reduce the number of working sensors, Wang *et al.* [14] employed sensors with directional antenna and limited measurement capabilities in the proposed tracking protocol with directional binary sensors, named DBS-tracking. The sensors have ability to recognize not only target's appearance but also the sector where the target located. This mechanism achieves a better performance in terms of energy consumption because the rich information of the target can be obtained only using a small number of working sensors. Although the number of sensors staying awake is reduced, the directional antennas or limited measurement capabilities are very costly. The extra information containing presence/absence of a target is also broadcasted, significantly increasing network overheads.

For reducing the sensor cost in advance, Wang *et al.* [15] considered the binary sensor networks and proposed a real-time and distributed target tracking (RTDT-Tracking) algorithm. They assume that the target located on a partial arc of the sensing range of the sensor that firstly detects the target's presence or absence. Each sensor collects the detecting information of its neighbors and then calculates the arc by intersecting the circles of neighbors. Although the tracking accuracy of the mobile target is improved in a dense deployed WSN, the network lifetime is extremely low. This occurs because that they did not consider the sleep scheduling of the sensors. This also implies that the sensor keeps awake even the target far away from it until the sensor exhausts its energy.

In [16], Wang *et al.* employed polygon (face) to detect the movement of the target, aiming to improve the tracking accuracy and the energy efficient. Once the target appeared in a face, sensors belong to the face are awakened and all the other sensors stay in the inactive state. Therefore, the energy consumption is reduced. The mechanism further estimates the target's moving toward another face such that sensors belong to the estimated face are awakened before the target's presence. This not only reduces the response time required for executing the face handoff procedure and but also could timely recover from the loss of tracking. However, the mechanism using faces to represent the quality of tracking leads to low tracking accuracy.

All of the algorithms mentioned above emphasize the improvement of either tracking accuracy or energy efficiency but could not guarantee the predefined tracking quality. This paper proposes an efficient target tracking mechanism (*ETT*) that dynamically wakes up sensors in a distributed manner, aiming to minimize the number of working sensors while the user-defined tracking quality is satisfied. Table 1 summarizes the characteristics of the related studies and the proposed *ETT*.

TABLE I
COMPARISON BETWEEN THE PROPOSED ETT AND EXISTING MECHANISMS

Studies	Distributed	Binary Sensor	Varible Velocity	Real-Time
BPS-Tracking [10]	X	○	X	X
UNS-Tracking [11]	X	○	○	X
ASA [12]	○	○	○	○
TPSS [13]	○	○	○	X
DBS-Tracking [14]	○	X	○	○
RTDT-Tracking [15]	○	○	○	○
Face Tracking [16]	○	○	○	○
The proposed *ETT*	○	○	○	○

III. NETWORK ENVIRONMENT

This section initially introduces the network environment and assumptions of the given WSN. Then, the problem formulations of our study are proposed.

A. Network Environment and Assumptions

Assume that a given monitoring region M has been randomly deployed with n sensors $S = \{s_1, s_2, \ldots, s_n\}$ that are capable of supporting full coverage. Each sensor s_i is aware of its physical location $p_i = (x_i, y_i)$ as well as the boundary of M. Assume that all sensors are time synchronized and are operated at a predefined time after they have been deployed in M.

B. Problem Statements

This paper aims to develop a distributed target tracking approach to awake the appropriate sensors for monitoring and tracking the mobile target. The reported trajectory of the mobile target should satisfy the requested *Quality of Tracking* (*QoT*). To prolong the network lifetime of the given WSN, another goal of the target tracking problem is to minimize the number of the sensors staying awake and the times switching between working and sleeping states.

The following presents the problem formulation of this work. The quality of the target tracking algorithm is mainly determined by two parameters: tracking accuracy and energy consumption. The tracking accuracy is measured by the location deviation between the estimated location and the target location. The energy consumption is measured by the total energy cost of the sensors.

Let $E = \{Entering, Moving, Exiting\}$ denote the event types of target entering of monitoring area, target moving out the restricted region formed by the working sensors, and target exiting the monitoring area, respectively. Initially, the whole network starts to work at time t_0. Assume that the time between t_0 and a particular time t_m is divided into m unequal time segments $[T_0, T_1, \ldots, T_m]$ according to the occurrence of some events $[e_0, e_1, \ldots, e_m]$, respectively, where $e_z \in E$ for $0 \le z \le m$. Assume that the event e_z of target g occurs at t_z. The time segment T_z is represented as the time period $[t_z, t_{z+1}]$.

Let $p_g^t\left(x_g^t, y_g^t\right)$ denote the real location of target g at time point $t \in T_z$. Let A_g^t denote an existing area of mobile target g estimated by working sensors at t. Let $p^t\left(x_p^t, y_p^t\right)$ be any point in A_g^t at t. Let ε^t denote the location deviation between the real location of target g and the estimative location given by the algorithm at t. Notation ε^t can be calculated according to Equ. (1).

$$\varepsilon^t = \max_{p_g^t, p^t \in A_g^t} \left\{ \left(\left[x_g^t - x_p^t\right]^2 + \left[y_g^t - y_p^t\right]^2 \right)^{1/2} \right\}, \ t \in T_z \tag{1}$$

Let $Q_{ETT}^t = 1/\varepsilon^t$ denote the tracking accuracy of the proposed ETT algorithm at t. Let Q_{req} denote the user defined requirement of tracking quality. The user defined tracking quality can be satisfied if Exp. (2) holds, where T is the total time the target located in the monitoring area.

$$\forall t \in T, \quad Q_{ETT}^{T_z} \ge Q_{req} \tag{2}$$

In addition, the energy conservation is an important key of prolonging the network lifetime. A sensor node consumes energy when it stays in working, sleeping, and even switching states between sleeping and working. The energy consumption of a sensor in working state includes sensing for detecting the target and communicating with neighbors. The energy consumption of a sleeping sensor is used for maintaining the ability of receiving broadcast message from the working sensors. Let $e^{working}$, $e^{sleeping}$, and $e^{switching}$ denote the energy consumptions of a sensor for working, sleeping, and state switching operations for a unit time, respectively. Let $\delta_i^{t,working}$, $\delta_i^{t,sleeping}$, and $\delta_i^{t,switching}$ denote the Boolean variables, representing whether or not sensor s_i stays in working, sleeping or state switching at t. Equ. (3) presents the constraint that each sensor can only stay in one state at any given time.

$$\delta_i^{t,working} + \delta_i^{t,sleeping} + \delta_i^{t,switching} = 1 \tag{3}$$

Let $|T_z|$ represent the value of the time segment T_z from the time point t_z to the time point t_{z+1}. Exp. (4) aims to minimize the energy consumption of all sensors deployed in the given monitoring region.

$$\mathbf{min} \left\{ \sum_{i=0}^{n} \sum_{z=0}^{m-1} \left(\delta_i^{t,working} * e^{working} + \delta_i^{t,sleeping} * e^{sleeping} + \delta_i^{t,switching} * e^{switching} \right) * |T_z| \right\} \tag{4}$$

IV. THE PROPOSED ETT MECHANISM

The goal of this paper is to prolong the network lifetime under the condition that the tracking quality defined by user can be guaranteed. The proposed *ETT* mechanism mainly consists of three phases: *Network Initialization Phase*, *Target Detection Phase*, and *Target Tracking Phase*. The *Network Initialization* phase aims to wake up some sensor nodes such that any target will be caught once it enters the monitoring region. These nodes are called *working boundary sensors*. When the target is caught by a *working boundary sensor*, the *Target Detection* phase is launched to estimate the target location such that the estimative location accuracy satisfies the user requirement while the number of awaken sensors is as few as possible. Moreover, the *Target Tracking* phase is designed to cope with the target movement problem.

A. Network Initialization Phase

This section presents the details of the Network Initialization Phase. The main purpose of this phase is to find a minimal number of working boundary sensors such that any target entering region M can be detected by at least one sensor. This phase begins at time point t_0 and we assume that it will be accomplished during time segment T_0. Each sensor will stay in one of the following three states: *working*, *checking*, and *sleeping* states. Initially, each sensor stays in *checking* state, aiming to make a decision whether or not it should participate in the monitoring task. A sensor stays in *working* state if it joins the monitoring task. On the contrary, a sensor stays in *sleeping* state for prolonging its own lifetime if it abandons the monitoring task.

Initially, each sensor stays in *checking* state. Let $S^{boundary}$ denote the set of sensors, called *boundary sensors*, whose sensing ranges cover the boundaries of M. Let notation $\mathcal{W}_B$ denote the set of *working boundary sensors* that are selected to participate in the boundary monitoring task. Each boundary sensor $s_i \in S^{boundary}$ in checking state evaluates its contribution c_i which is measured by the length of boundary segment covered by s_i. The boundary sensors will wake up and broadcast their IDs and location information in a distributed manner, according to their own contributions. More specifically, the boundary sensor that has larger contribution to the border monitoring is included in $\mathcal{W}_B$ earlier and then broadcasts the working message including its ID and location. Upon receiving the working message, all the other boundary sensors need to recalculate their own contributions and reset their own waiting time. Let l_i denote the shortest distance between the sensor $s_i \in S^{boundary}$ and the boundary of the monitoring region. Let r denote the sensing radius of each sensor deployed in monitoring region. The contribution c_i of sensor s_i can be derived by applying Equ. (5).

$$c_i = 2(r^2 - l_i^2)^{1/2} \tag{5}$$

We assume that the given monitoring range is much larger than the sensing range of a sensor node. This guarantees that a sensor deployed at the corner of M intersects with at most two edges of M.

After obtaining the value of contribution c_i, the waiting time of each sensor can be calculated according to Equ. (6).

$$w_i = \frac{r}{c_i} \tag{6}$$

As shown in Fig. 1, the initial contributions of boundary sensors s_i and s_j are c_i and c_j, which are represented by red and blue segments, respectively. Since the value of c_i is larger than that of c_j, the waiting time of sensor s_i is shorter than that of sensor s_j, according to the Equ. (6). When sensor s_i finishes its waiting time, it broadcasts the working message containing its own ID and physical location. Each of the other boundary sensors, named s_j, that has the overlapped boundary coverage with s_i needs to recalculate the contribution c_j and updates the waiting time w_j. The new contribution of sensor s_j should be recalculated by using the original contribution with subtraction of the common contribution of sensors s_i and s_j. Let $d(i, j)$ denote the distance between sensors s_i and s_j. Let $l_{i,j}^{overlap}$ denote the length of boundary segment that are commonly covered by s_i and s_j. The value of $l_{i,j}^{overlap}$ can be derived by applying Equ. (7).

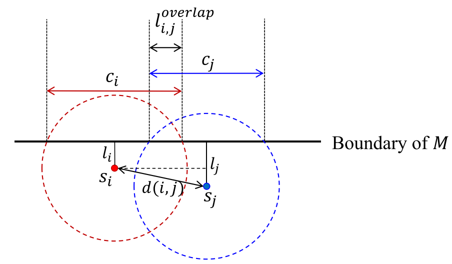

Fig. 1. Overlap boundary between sensors s_i and s_j causes sensor s_j to recalculate its contribution and update its waiting time.

$$l_{i,j}^{overlap} = c_i/2 + c_j/2 - (d(i,j)^2 - (l_j - l_i)^2)^{1/2} \tag{7}$$

The new contribution c'_j of s_j can be further derived according to Equ. (8).

$$c'_j = c_j - l_{i,j}^{overlap} \tag{8}$$

The new waiting time w'_j of sensor s_j is also reset by applying Equ. (9).

$$w'_j = 2r/c'_j - w_i \tag{9}$$

After waiting for the backoff time of w'_j, sensor s_j wakes up, switches to the working state, and then broadcasts the working message. All the other boundary sensors should recalculate their own waiting time based on Equ. (9). The network initialization phase should be repeated executing until the boundary of M has been full covered by *working boundary sensors* in $\mathcal{W}_B$. All the boundary sensors whose new contributions equal to zero obtained by applying Equ. (8) will directly switch from checking to sleeping state, where sensors only need to periodically listen to the *state change message*. Once any boundary sensor runs out of its energy, the initialization process will be restarted to pick up a new group to take over the job of the failure boundary sensor.

Fig. 2 depicts the detailed steps designed for *Network Initialization* phase. In Step 1, the boundary sensors are selected in the set $S^{boundary}$. Next, the sensor with the shortest waiting time is identified in Steps 2 and 3. Then, this sensor is added in the set $\mathcal{W}_B$ and is removed from $S^{boundary}$ in Step 4. Step 5 tends to check whether or not the boundaries of M are fully covered by the sensors in $\mathcal{W}_B$. If it is not the case, Step 6 will be executed, aiming to select new *working boundary sensor*. Otherwise, Steps 7 and 8 will be carried out and the procedure is terminated.

```
Network Initialization Procedure
Input:
1. A set of sensors S = {s_1, s_2, ..., s_n}, where m is the number of sensors.
2. Each sensor s_i ∈ S is aware of its physical location p_i = (x_i, y_i) as
well as the boundary of M.
Output: W_B
      /* Boundary sensors selection */
1.    for (i=1, i ≤n, i++){
          compute l_i;
          if l_i ≤ r then S^boundary ← s_i;}
      min_w=∞;
2.    sen_id=0;
      /* Calculation of contribution and waiting time */
      for (j=1, j ≤ | S^boundary | , j++){
3.        compute c_j according to Equ. (8)
          compute w_j according to Equ. (9)
          if w_j < min_w then{
              min_w ← w_j;
4.            sen_id←j; }}
      W_B←S^boundary(sen_id);
5.    S^boundary=S^boundary - S^boundary(sen_id);
6.    While the boundary of M is not fully covered by W_B
7.        Go to 3
8.    end while
      Return W_B
```

Fig. 2. Network Initialization Procedure.

B. Target Detection Phase

This phase will be initiated when any working boundary sensor detected the target. This phase aims to select a minimal number of sensors in a distributed manner to monitor the target such that the user requirement of *Quality of Tracking* (*QoT*) can be satisfied. Once the target enters the monitoring region M, time segment T_0 ends. A new event e_1 occurs because that the target g enters the monitoring region and hence a new time segment T_1 begins. This event also triggers the execution of the target detection phase, which aims to select some appreciate sensors to participate in the monitoring work. We assume that the Target Detection Phase will be triggered once the event e_z occurs, where $z \in [1, \ldots, m]$. Without loss of generality, the time parameter T_z will be omitted in the following descriptions.

As soon as the target passes through the boundary of monitoring region, it must be detected by at least one working boundary sensor. The working boundary sensor that detects target is called *Master Node* that is responsible to notify its neighbors for executing the operations designed in the target detection phase. If there are two working boundary sensors catching the target at the same time, a negotiation is required between them such that the one with larger remaining energy should play the role of Master Node. If they have the same remaining energy, the one with larger ID will be selected as the Master Node. Let s_{master} denote the *Master Node*.

Let the *restricted area* of target g, denoted by A_g, represent the area where the movement of target g cannot be detected by any sensor. Initially, the A_g can be considered as the sensing range of s_{master}. The sensor s_{master} should further select some neighbors to participate in the monitoring task since the size of the current A_g is still larger than the accuracy requirement as defined by user.

Initially, the sensor s_{master} broadcasts the detecting message containing its ID and its own physical location. Upon receiving the detecting message, all neighbors of s_i switch from *sleeping* state to *checking* state, intending to check if they should join the monitoring task. The sensors that determine to participate in the monitoring task will switch from checking state to working state; otherwise, they will return to *sleeping* state. In this phase, the sensors that stay in working state are called *working sensors*. Let notation $\mathcal{W}_D = \{s_1^{working}, \ldots, s_k^{working}\}$ denote the set of *working sensors*. The following presents how to construct $\mathcal{W}_D$ in a distributed manner such that the number of sensors in $\mathcal{W}_D$ is as few as possible while the user defined QoT can be achieved.

In order to simply the description for our algorithm, herein we formally give some definitions. Let R_i denote the sensing range of sensor s_i and $|R_i|$ represent the area size of R_i, where $R_i = \{x | d(x, s_i) \leq r_s\}$. Let α_i be a Boolean variable, which is defined as follows.

$$\alpha_i = \begin{cases} 1, & s_i \text{ is in working state} \\ 0, & s_i \text{ is in sleeping or checking states} \end{cases}$$

Let S_{master}^N denote the set of neighbors of node s_{master}. The S_{master}^N will be classified into two types, according to the condition whether or not the sensors detect the target. Let $S_{master}^{N,D}$ and $S_{master}^{N,ND}$ denote the sets of sensors that detected and not-detected the target, respectively. Each sensor $s_j \in S_{master}^N$ should evaluate its contribution to check if it will be included in $\mathcal{W}_D$.

To select neighbors of s_{master} for target detection, this phase proposes three strategies, including *Target-Not-Detected Neighbor Selection* (*TND-NS*), *Target-Detected Neighbor Selection* (*TD-NS*) and *Integrated Neighbor Selection* (*I-NS*) strategies. The *TND-NS* and *TD-NS* strategies aim to find appropriate sensors to participate in monitoring task from s_{master}'s neighbors in $S_{master}^{N,D}$ or $S_{master}^{N,ND}$, respectively. The *Integrated Neighbor Selection* (*INS*) strategy further integrates *TN-DNS* and *TD-NS* algorithms and takes advantages from them for obtaining better performance of tracking accuracy.

1) Target-Not-Detected Neighbor Selection (TND-NS) Strategy: This subsection illustrates the *Target-Not-Detected Neighbor Selection* (*TND-NS*) strategy in detail. The *TND-NS* strategy aims to select sensors from s_{master}'s neighbors that do not detect target g in a distributed manner. Once neighbors of s_{master} receive the detecting message from s_{master}, they will switch from *sleeping* state to *checking* state, intending to check if they should be included in $\mathcal{W}_D$.

Let Φ^{active} denote the set of sensors $s_j \in S_{master}^{N,ND}$ which have already determined to participate in the monitoring task. That is, $\Phi^{active} = \{s_j | \alpha_j = 1\}$. Initially, set Φ^{active} is a null set. In the conceptual level, each sensor s_j calculates its contribution c_j according to the improvement of the tracking accuracy of target g. Based on the value of c_j, each sensor waits for a random backoff time. The sensor with larger contribution will wait for a shorter time and hence it will have higher priority to participate in the monitoring task. Then the winner joins the active set Φ^{active} and announces

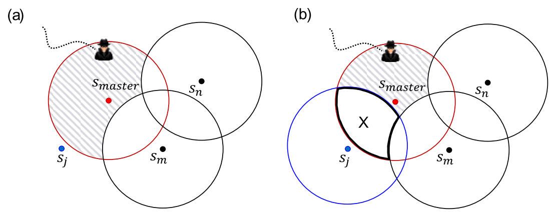

Fig. 3. A_g subject to the set Φ^{active}. (a) $\Phi^{active} = \{s_m, s_n\}$. (b) $\Phi^{active} = \{s_m, s_n, s_j\}$.

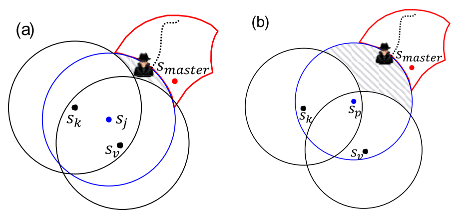

Fig. 4. Future potential weight. (a) Sensor s_j, which is closed to neighbors s_k and s_v, constructs a small restricted area that provides a high future potential weight. (b) Sensor s_p, which is far from neighbors s_k and s_v, constructs a large restricted area that provides a low future potential weight.

its participation. Upon receiving the participating announcement from any neighboring sensor, each sensor should recalculate its contribution and reset its backoff time. The procedure will be repeatedly executed until the working sensors can form a monitoring area which restricts the target with satisfied accuracy. The backoff time mainly depends on the contribution c_j of sensor s_j, which consisted of two weights, including the current potential weight and future potential weight. The current potential weight of s_j represents the reduced size of A_g caused because that sensor $s_j \in S^{N,ND}_{master}$ participate in the monitoring task. If the target is restricted in a small region, a high tracking accuracy can be achieved. Finally, the set $\mathcal{W}_D$ can be obtained by setting $\mathcal{W}_D = \Phi^{active}$.

As shown in Fig. 3(a), assume that sensors s_m and $s_n \in S^{N,ND}_{master}$ have been included in set Φ^{active}. The restricted area A_g of target g can be represented as the shadow region. Each neighbor of s_{master} will calculate its contribution to the reduction of A_g. As shown in Fig. 3(b), the shadow area represents the new A_g if sensor s_j participates in the monitoring task. Therefore, the reduced size of A_g by newly adding sensor s_j in Φ^{active} is identical to the size of region X which is surrounded by bold edge. This implies the value of current potential weight of sensor s_j, denoted by the notation c^{curr}_j, will be the size of region X.

Another important factor to evaluate the contribution of s_j is its future potential weight. Since the target might move into the sensing region of working sensor s_j, the future potential weight of s_j represents how well the cooperation among neighbors of s_j such that these neighbors can form a smaller restricted area A_g for the target g.

As shown in Fig. 4, assume that the target does enter the sensing range of sensor $s_j \in \Phi^{active}$ in the near future. Let s_k and s_v be the two neighbors of s_j. The shadow regions depicted in Fig. 4(a) and 4(b) represents the restricted areas of target g. In fact, if the target moves out the shadow region, one of sensors s_j, s_k or s_v will detect the movement of target g. Therefore, the shadow region represents the restrict area of target g. A good property of sensor s_j is that it has closer neighbors such that a small enough restricted area can be formed. For example, Fig. 4(a) has a smaller size of shadow region than Fig. 4(b). This occurs because sensors s_k and s_v in Fig. 4(a) are much closer to sensor s_j, as compared with those sensors in Fig. 4(b). The good property of s_j shown in Fig. 4(a) will benefit to the restricted area only if the target really enters the sensing range of sensor s_j in the near future. Therefore, the contribution is treated as future potential weight. Let notation c^{future}_j denote the future potential weight of sensor s_j. The value of c^{future}_j depends on the size of the new A_g formed by sensor s_j and its neighbors s_k and s_v.

The current potential weight c^{curr}_j and future potential weight c^{future}_j will be integrated as single contribution indicator which will be used to determine the backoff time of s_j. If sensor s_j is the first one that finishes its waiting time, it announces a participating message to all neighbors that it has been included in Φ^{active}. It is obvious that one additional sensor participating in Φ^{active} can further reduce the size of A_g and hence improve the tracking accuracy of target g. In what follows, the current potential factor is further discussed.

a) Evaluation procedure of current potential weight: The following presents the calculation of the current potential weight in detail. Recall that R_i denote the sensing range of sensor s_i and notation A_g denotes the *restricted area* which is constructed by all sensors $s_k \in \Phi^{active}$ and sensor s_{master}. Equ. (10) represents the calculation of A_g.

$$A_g = R_{master} \setminus \bigcup_{s_k \in \Phi^{active}} R_k \tag{10}$$

As soon as sensor s_j participates in the monitoring task, the new restricted area can be calculated by Equ. (11).

$$A^{new}_g = R_{master} \setminus \left[R_j \cup \left(\bigcup_{s_k \in \Phi^{active}} R_k \right) \right] \tag{11}$$

As discussed above, the current potential weight of s_j represents the reduced size of A_g by adding sensor s_j in Φ^{active}. Threrefore, sensor $s_j \in S^{N,ND}_{master}$ in *checking state* evaluates its current potential weight by applying Equ. (12) in a distributed manner.

$$c^{curr}_j = \frac{\left| A_g - A^{new}_g \right|}{|R|} \tag{12}$$

where $|R|$ denote the size of sensing range of each sensor. If sensor s_j participates in Φ^{active}, the current A_g should be replaced by A^{new}_g. The current potential weight will be one important factor for determining the priority of s_j to participate in the monitoring task. Another important factor is the future potential weight which will be discussed in the next subsection.

b) Evaluation procedure of future potential weight: The discussion of future potential weight is based on an assumption that sensor s_j will detect the target in the near future. Recall that the notation c^{future}_j denotes the good property that

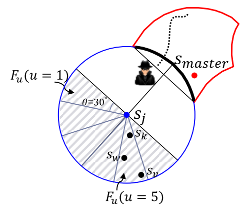

Fig. 5. Calculation of future potential weight.

sensor s_j has closer neighbors such that a small enough restricted area can be formed. Let the notation $d(s_j, s_k)$ denote the distance between sensor s_j and its neighboring sensor s_k. As shown in Fig. 4, the restricted region in Fig. 4(a) is smaller than that in Fig. 4(b). This occurs because the neighbors of sensor s_j are closer to s_j such that they can construct a smaller restricted region. Since a smaller restricted region represents a higher accuracy for target tracking, sensor s_j has higher future potential weight c_j^{future} than sensor s_p. Therefore, the future potential weight c_j^{future} can be evaluated simply by the distances between sensor s_j and some representative neighbors.

The following presents the selection procedure which select some representative neighbors $s_k \in S_j^{N,ND}$ for calculating the value of c_j^{future}. As shown in Fig. 5, if target g moves in the sensing range of sensor s_j, it should cross the bold edge and therefore it is detected by sensor s_j. The neighbors of sensor s_j that fall in the shadow region shown in Fig. 5 have high impact on the size of restricted area of target g. Therefore, the shadow semicircle region of sensor s_j will be equally partitioned into several sectors with angle $\theta = 30°$ and one representative sensor will be selected from each sector. Let F_u denote the u-th sector along with the anticlockwise direction, where $1 \le u \le 6$, as shown in Fig. 5. Let $\Phi_{s_j}^{F_u}$ denote the set of neighbors which falls in the region of F_u but does not detect target g. The sensor in set $\Phi_{s_j}^{F_u}$ which has the shortest distance with sensor s_j will be selected to represent the good property of sector F_u. Let s_{j,F_u}^{best} represent the selected sensor in set $\Phi_{s_j}^{F_u}$. Equ. (13) depicts the selection criteria of s_{j,F_u}^{best}.

$$s_{j,F_u}^{best} = \underset{s_k \in \Phi_{s_j}^{F_u}}{\arg\min} \left(d\left(s_k, s_j\right)\right) \tag{13}$$

For example, as shown in Fig. 5, there are three sensors s_w, s_v and s_k in $\Phi_{s_j}^{F_u}$, where $u = 5$. According to Equ. (13), sensor s_k will be selected to represent the good property of sector F_5.

The selection procedure will select one sensor from each sector by applying Equ. (13). After that, a set of the best representative sensors, denoted by Φ_{j,F_u}^{best}, will be obtained as shown in Equ. (14).

$$\Phi_{j,F_u}^{best} = \left\{s_{j,F_u}^{best} | 1 \le u \le 6\right\} \tag{14}$$

Herein, the weakest one of set Φ_{j,F_u}^{best} will be selected to represent the good property of sensor s_j. Equ. (15) presents the calculation of $s_j^{weakest}$.

$$s_j^{weakest} = \underset{s_{j,F_u}^{best} \in \Phi_{j,F_u}^{best}, 1 \le u \le 6}{\arg\max} \left(d\left(s_{j,F_u}^{\mathrm{best}}, s_j\right)\right) \tag{15}$$

The future potential contribution of sensor s_j in $S_{master}^{N,ND}$ can be obtained by applying Equ. (16).

$$c_j^{future} = 1 - d\left(s_j^{weakest}, s_j\right)/r \tag{16}$$

where r denote the sensing radius of each sensor.

c) Selection procedure based on $S_{master}^{N,ND}$ set: Next, both of two factors, including the current and future potential weights discussed above, will be integrated into consideration for determining the priority of s_j to participate in the monitoring task. Equ. (17) presents the calculation of the backoff time of sensor $s_j \in S_{master}^{N,ND}$. The impact of β will be observed in the performance experiments.

$$t_{s_j}^{backoff} = \beta \times \frac{1}{c_j^{curr}} + (1-\beta) \times \frac{1}{c_j^{future}} \tag{17}$$

Sensors with larger contribution will have shorter backoff time and thus have higher opportunity to take part in the monitoring work. Upon completing the waiting of backoff time, sensor s_j announces its participation. All the other sensors that stay in checking state but receive the message will recalculate its new contribution and reset its waiting time. Assume that s_j firstly announces its participation. Since sensor s_j is the new working sensor and joins in Φ^{active}, the restricted area A_g should be updated based on Equ. (10).

As described earlier, this phase aims to satisfy the user's tracking accuracy requirement by selecting a minimal number of working sensors. After A_g has been updated, it also should be checked if the size of A_g is small enough such that it meets the tracking accuracy requirement as requested by user. Exp. (1) will be applied for calculating the deviation between the real location of target g and the estimated one while Exp. (2) will be applied for checking if the tracking accuracy meets the requirement. For example, assume that $Q_{req} = 1/3$. This indicates that the maximal deviation accepted by user is $\varepsilon^t = 3$ meters. Whenever a working sensor is selected, the value of Q_{ETT}^t is calculated and Exp. (2) is checked. If the value of Q_{ETT}^t equals to 1/4, Exp. (2) is not satisfied. Therefore the Master Node should continue to execute the selection procedure which further selects additional working sensor. If the value of Q_{ETT}^t equals to 1/2, the Master Node will terminate the selection procedure. This indicates that the maximal deviation is 2 meters. The tracking accuracy achieved by *ETT* obviously satisfies the user defined one.

If it is the case, the selection of working sensors will be terminated immediately. Otherwise, each neighbor of sensor s_{master}, say s_k, should recalculate its new contributions c_k^{curr} and c_k^{future} based on Equs. (11), (12) and (13)–(17). Then the next new working sensor will be selected. These operations will be repeated executed until the size of restricted area is smaller than the required tracking accuracy. In summary, this subsection has presented a mechanism for master sensor to select its neighbors that are not detected target for constructing a restricted region. The next subsection further considers the s_{master}'s neighbors that detected the target g.

Fig. 6 depicts the procedure designed for *TND-NS strategy*. The relevant variables are initialized in Steps 1 and 2. In Step 3, the $t_{s_j}^{backoff}$ of each sensor s_j in $S_{master}^{N,ND}$ is computed.

***TND-NS* Procedure**	
Inputs:	
1. A set of neighbor sensors $S_{master}^{N,ND}$ of s_{master}	
2. The *QoT* requirement Q_{req} requested by user	
Output: Φ^{active}	
1.	Φ^{active} is null;
	$A_g = R_{master}$;
2.	$min\_t=\infty$;
	sen_id=0;
	/* Calculation of contribution and backoff time */
3.	for (j=1, $j \le \lvert S_{master}^{N,ND} \rvert$, j++) {
	computes c_j^{curr} according to Equ. (12)
	computes c_j^{future} according to Equ. (16)
	computes $t_{s_j}^{backoff}$ according to Equ. (17)
	if $t_{s_j}^{backoff} < min\_t$ then{
	$min\_t \leftarrow t_{s_j}^{backoff}$;
4.	$sen\_id \leftarrow j$; }}
	/* Update Φ^{active} and A_g */
5.	$\Phi^{active} \leftarrow S_{master}^{N,ND}(sen\_id)$;
6.	$S_{master}^{N,ND} = S_{master}^{N,ND} - S_{master}^{N,ND}(sen\_id)$;
7.	computes A_g^{new};
8.	$A_g \leftarrow A_g^{new}$;
9.	**While** the size of A_g is not satisfied with Q_{req}
	Go to 3;
	End while
	Return Φ^{active}

Fig. 6. *TND-NS* Procedure.

The sensor with the shortest $t_{s_j}^{backoff}$ will be included in Φ^{active} and is removed from $S_{master}^{N,ND}$ as shown in Step 4. In Step 5, the *restricted area* A_g is updated. In Step 6, the size of A_g should be checked if it meets the requirement defined by user. If it is not the case, Step 7 will be executed in order to select another *working sensor.* Otherwise, Steps 8 and 9 will be carried out and the procedure is terminated.

2) Target-Detected Neighbor Selection (TD-NS) Strategy: The *Target-Detected Neighbor Selection* (*TD-NS*) strategy is proposed to select sensors from s_{master}'s neighbors that have detected target g in a distributed manner. Similar to *TND-NS*, each sensor in $S_{master}^{N,D}$ will switch state from *sleeping* to *checking* upon receiving the detecting message. Furthermore, each sensor in $S_{master}^{N,D}$ will also calculate its contribution and its backoff time. Similar to sensors in $S_{mater}^{N,ND}$, the sensor with larger contribution has higher priority to announce that it should be included in $\mathcal{W}_D$. Let $\tilde{\Phi}^{active} \subset \mathcal{W}_D$ denote the set of sensors $s_j \in S_{master}^{N,D}$ which have already been included in $\mathcal{W}_D$. That is, $\tilde{\Phi}^{active} = \{s_i | \alpha_i = 1\}$. Initially, $\tilde{\Phi}^{active}$ is a null set. Likewise, each sensor $s_j \in S_{master}^{N,D}$ stays in *checking state*, aiming to evaluate its contribution. Based on its contribution, each sensor setups a random backoff time in a distributed manner. The sensor with the shortest backoff time will join the set $\tilde{\Phi}^{active}$ and announce its participation. Upon receiving the participating announcement from any neighbor, sensors stayed in checking state will recalculate its contribution and reset its backoff time. The abovementioned operations will be repeatedly carried on until the working sensors can form a restricted area which restricts the target with satisfied accuracy.

The *TD-NS* strategy mainly consists of two procedures, including *Evaluation Procedure of Contribution* and *Selection*

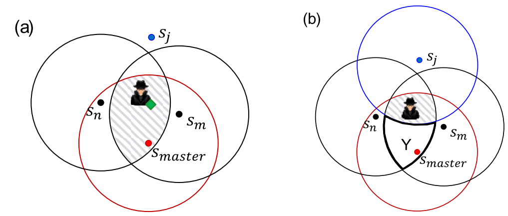

Fig. 7. $\tilde{A}_g$ subject to the set $\tilde{\Phi}^{active}$. (a) $\tilde{\Phi}^{active} = \{s_m, s_n\}$. (b) $\tilde{\Phi}^{active} = \{s_m, s_n, s_j\}$.

Procedure based on $S_{master}^{N,D}$ Set. The two procedures will be introduced in detail.

a) Evaluation procedure of contribution: This subsection presents the contribution evaluation of sensor $s_j \in S_{master}^{N,D}$. Let notation $\tilde{c}_j$ denote the contribution of sensor $s_j \in S_{master}^{N,D}$. Let $\tilde{A}_g$ denote the *restricted area* of target g that is formed by sensor s_{master} and the working sensors $\in S_{master}^{N,D}$. Herein, the contribution $\tilde{c}_j$ of sensor $s_j \in S_{master}^{N,D}$ represents the reduced size of $\tilde{A}_g$ by adding sensor s_j in $\tilde{\Phi}^{active}$.

As shown in Fig. 7(a), assume that sensors s_m and $s_n \in S_{master}^{N,D}$ have been included in set $\tilde{\Phi}^{active}$. The restricted area $\tilde{A}_g$ of target g can be represented as the shadow region. Equ. (18) represents the calculation of A_g.

$$\tilde{A}_g = R_{master} \cap \left(\bigcap_{s_k \in \tilde{\Phi}^{active}} R_k \right) \tag{18}$$

As soon as the sensor s_j participates in the monitoring task as shown in Fig. 7(b), the new restricted area can be calculated by applying Equ. (19).

$$\tilde{A}_g^{new} = R_{master} \cap \left[R_j \cap \left(\bigcap_{s_k \in \tilde{\Phi}^{active}} R_k \right) \right] \tag{19}$$

As discussed above, the current potential weight of s_j represents the reduced size of $\tilde{A}_g$ by adding sensor s_j in $\tilde{\Phi}^{active}$. Threrefore, sensor $s_j \in S_{master}^{N,D}$in *checking state* evaluates its current potential weight by applying Equ. (20) in a distributed manner.

$$\tilde{c}_j = \frac{\left| \tilde{A}_g - \tilde{A}_g^{new} \right|}{|R|}, \tag{20}$$

where |R| denotes the size of sensing range of each sensor.

b) Selection procedure based on $S_{master}^{N,D}$ set: This subsection presents the selection procedure which aims to select a minimal number of sensors in set $S_{master}^{N,D}$ to participate in the monitoring task. Since each sensor determines its participation decision by its own in a distributed manner, a backoff scheme is locally applied by each sensor. The length of backoff time mainly depends on the contribution $\tilde{c}_j$ discussed above. Equ. (21) presents the calculation of backoff time of sensor $s_j \in S_{master}^{N,D}$. Based on the value of $\tilde{t}_{s_j}^{backoff}$, sensors will wake up one by one to join the active set $\tilde{\Phi}^{active}$ in a distributed manner by complying the similar criteria until the requested

***TD-NS* Strategy**	
Inputs: 1. A set of neighbor sensors $S_{master}^{N,D}$ of s_{master} **2.** The *QoT* requirement Q_{req} requested by user	
Output: $\tilde{\Phi}^{active}$	
1.	$\tilde{\Phi}^{active}$ is null;
2.	$\tilde{A}_g = R_{master}$; *min_t*=∞; *sen_id*=0;

3.	/* Calculations of contribution and backoff time */
	for (j=1, $j \leq \lvert S_{master}^{N,D} \rvert$, j++) {
	compute $\tilde{c}_j$ according to Equ. (20)
	compute $\tilde{t}_{s_j}^{backoff}$ according to Equ. (21)
	if $\tilde{t}_{s_j}^{backoff} < min\_t$ then {
	$min\_t \leftarrow \tilde{t}_{s_j}^{backoff}$;
	sen_id←*j*; }}

4.	/* Update $\tilde{\Phi}^{active}$ and $\tilde{A}_g$ */ $\tilde{\Phi}^{active} \leftarrow S_{master}^{N,D}(sen\_id)$;
5.	$S_{master}^{N,D} = S_{master}^{N,D} - S_{master}^{N,D}(sen\_id)$;
6.	compute $\tilde{A}_g^{new}$;
7.	$\tilde{A}_g \leftarrow \tilde{A}_g^{new}$;
8.	**While** $\tilde{A}_g$ is not satisfied with Q_{req} Go to 3; **End while**
9.	**Return** $\tilde{\Phi}^{active}$

Fig. 8. *TD-NS* Strategy.

tracking quality is satisfied.

$$\tilde{t}_{s_j}^{backoff} = 1/\tilde{c}_j \tag{21}$$

Assume that sensor s_j is the new working sensor to be included in Φ^{active}. The restricted area $\tilde{A}_g$ should be updated based on Equ. (19). Similarly, the size of the new restricted area $\tilde{A}_g$ will be checked if it is small enough to satisfy the tracking requirement defined by user, as shown in Equ. (1) and Exp. (2). If the tracking requirement of user is achieved, the selection procedure of working sensors will be terminated. Otherwise, sensors in checking state will recalculate their own contributions and compete to play the role of working sensor until the quality of tracking meets the requirement of users.

Fig. 8 depicts the procedure designed for *TD-NS* Strategy. In Steps 1 and 2, the relevant variables are initialized. In Step 3, the $\tilde{t}_{s_j}^{backoff}$ of each sensor s_j in $S_{master}^{N,D}$ is computed. The sensor with the shortest $\tilde{t}_{s_j}^{backoff}$ will be picked up. Then this sensor is added in $\tilde{\Phi}^{active}$ and is removed from $S_{master}^{N,D}$ in Step 4. In Step 5, the *restricted area* $\tilde{A}_g$ is updated. Then the size of $\tilde{A}_g$ should be checked if it meets the requirement defined by user as shown in Step 6. If it is not the case, Step 7 will be executed in order to select another *working sensor*. Otherwise, Steps 8 and 9 will be carried out and the procedure is terminated.

3) Integrated Neighbor Selection (I-NS) Strategy: The following illustrates the Integrated Neighbor Selection (*I-NS*) strategy which integrates *TND-NS* and *TD-NS* strategies. Both of *TND-NS* and *TD-NS* aim to select sensors to construct a small enough restricted area for target g such that the tracking accuracy satisfies the user's requirement. The *TND-NS* strategy considers the impact of sensors in $S_{master}^{N,ND}$ while the *TD-NS* strategy pays attention to the impact of sensors in $S_{master}^{N,D}$. The *I-NS* strategy is proposed to take advantage of all neighbors of s_{master}.

Upon receiving the detection message from s_{master}, each neighbor, say s_j, of s_{master} should calculate its contribution and backoff time. The calculations of contribution and backoff time of sensor s_j depend on the situation whether or not sensor s_j detects target g. If a neighbor does not detect target g, it applies Equs. (12) and (16) to calculate its c_j^{curr} and c_j^{future}, respectively. Otherwise, it applies Equ. (20) to calculate its contribution. It's observed that the values of contribution calculated in both cases are normalized in Equs. (12), (16) and (20). Therefore, all neighbors of s_{master}, including $S_{master}^{N,ND}$ and $S_{master}^{N,D}$, can fairly compete with each other for participating in the monitoring task. The sensor with larger contribution will set up a shorter backoff time which implies that it has a higher priority to be included in $\mathcal{W}_D$. Once a sensor finishes its backoff time, it announces its participation. Upon receiving the participation announcement from neighbor, each sensor recalculates its own contribution and backoff time. Herein, we notice that the restricted area of target g will be the intersection of A_g and $\tilde{A}_g$. The selection procedure will be repeated executed until the restricted area satisfies the user requirement of the tracking accuracy.

C. Master Handoff Phase

The *Master Handoff Phase* is proposed to deal with the problem of target movement and aims to obtain the trajectory of mobile target. The following presents three different strategies applied to *Master Handoff Phase*. The proposed three handoff strategies correspond to three *Target Detection* strategies as presented in Section 4.2. In the handoff procedure, we will discuss two issues: the condition that triggers the *Master Handoff Phase* and the selection of a new *Master Node*.

1) Handoff Procedure by Applying TND-NS Strategy: Recall that if *TND-NS* strategy is employed in *Target Detection Phase*, target g will be restricted in a small area A_g constructed by *Master Node* s_{master} and working sensors in Φ^{active}. Once target g moves out of the *restricted area* A_g, it will be detected by at least one working sensor in Φ^{active}. In order to continuously track the target with a satisfactory accuracy, the *Handoff Procedure* by applying *TND-NS* strategy should be triggered. Let $s_j \in \Phi^{active}$ be the one that detects the movement of target g. Sensor s_j should broadcast a *Change_Master* message and play the role of *Master Node*. If there are two working sensors detected the movement of target g, a negotiation is required between them such that the one with larger remaining energy should play the role of Master Node. In case that they have the same remaining energy, the one with larger ID will be chosen as the Master Node. Meanwhile, *TND-NS* strategy as proposed in the *Target Detection Phase* will be invoked to find a new group of working sensors.

2) Handoff Procedure by Applying TD-NS Strategy: Different from *TND-NS* strategy, the *TD-NS* strategy wakes up a group of s_{master}'s neighbors in $S_{master}^{N,D}$ to form a small *restricted area* $\tilde{A}_g$ for target g. As shown in Fig. 5(b), if target g moves out $\tilde{A}_g$, some sensors, say s_j^{exit}, might detect that target g exits its sensing range. This detection will invoke the *Handoff Procedure* which applies *TD-NS* strategy to find a new *Master Node*. Sensor s_j^{exit} should broadcast the *Exit* message. Let $\tilde{\Phi}^{active,new} = \tilde{\Phi}^{active} \backslash s_j^{exit}$ denote the set of nodes that have target g in their sensing range. Upon receiving the *Exit* message, sensors in $\tilde{\Phi}^{active,new}$ will recalculate the new *restricted area* of target g in a distributed manner. Let $\tilde{A}_g^{new}$ denote the new restricted area. Each sensor in $\tilde{\Phi}^{active,new}$ and s_{master} will measure the distance between itself and the geometric gravity of $\tilde{A}_g^{new}$. The working sensor with the shortest distance will be the new *Master Node* and announce the *Change_Master* message which notifies its neighbors about this change. If there are two working sensors satisfy the Master criteria, they should have a negotiation. The one with larger remaining energy should play the role of Master Node. If they have the same remaining energy, the one with larger ID should play the role of Master Node. Then, a new group of working sensors for monitoring task will be selected by applying the operations as described in *Target Detection Phase* with *TND-NS* strategy.

Fig. 7(b) gives an example for illustrating the handoff procedure. Assume that target g moves out $\tilde{A}_g$ which is marked as the shadow region, and then enters region Y which is marked with the bold edges as shown in Fig. 7(b). Since sensor s_j detects that target g exits its sensing range, it broadcasts the *Exit* message. Upon receiving the *Exit* message, the set of new working sensors $\tilde{\Phi}^{active,new} = \{s_m, s_n\}$ and s_{master} will recalculate the new restricted area for target g, which is marked as a shadow region in Fig. 7(a). Each sensor in $\tilde{\Phi}^{active,new}$ or s_{master} will compute the distances between working sensors and the shadow region's geometric gravity as marked with a green diamond. In this example, sensor s_m has the shortest distance and hence it announces the *Change_Master* message which indicates that s_m will be the new master. Then, a new group of working sensors for monitoring task will be selected through the *TND-NS* algorithm in the *Target Detection Phase*.

3) Handoff Procedure by Applying I-NS Handoff Strategy: The handoff procedure by applying the I-NS strategy mainly integrates the two handoff strategies proposed in previous subsections, aiming to take advantages from the two handoff procedures such that the number of working sensor can be further reduced. As soon as the target passes through the border of the*restricted area*, it will be detected by a sensor node either in Φ^{active} or $\tilde{\Phi}^{active}$ and hence the *I-NS Handoff* strategy is invoked. If the sensor that detects target movement, belongs to Φ^{active}, it plays the *Master Node* similar to the *TND-NS Handoff* Strategy. Otherwise, the new *Master Node* will be selected by applying the criteria of *TD-NS Handoff* strategy. Similarly, the new *Master Node* will announce a *Change_Master* message to notifies its neighbors about this change and the *I-NS* strategy of the *Target Detection Phase* is further applied to select a new group of sensors to be included in $\mathcal{W}_D$.

It's observed that, when the target g moves in the monitoring region, the proposed *Target Detection Phase* and *Master Handoff Phase* will be executed in turn in an event driven manner. The execution of the proposed *ETT* will be terminated until target g moves out the monitoring region. Whenever the *Target Detection Phase* is executed, a *restricted area* of target g will be formed to monitor target g. A trajectory that consists of continuous restricted areas of mobile target g will be obtained.

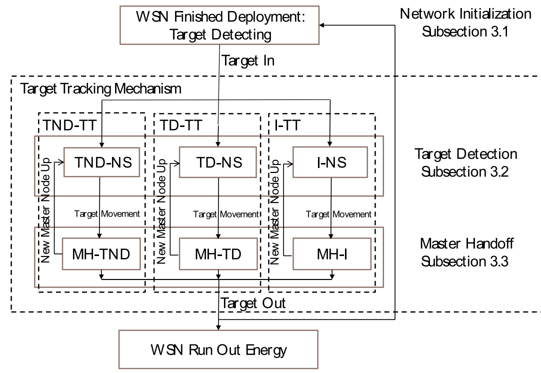

Fig. 9. Architecture of the proposed *ETT* mechanism.

V. Architecture of the Proposed ETT Mechanism

As presented in previous sections, the *ETT* mechanism consists of three phases: *Network Initialization*, *Target Detection* and *Master Handoff* phases. Fig. 9 gives the architecture of the proposed *ETT* mechanism. The *ETT* algorithm will be applied when the WSN has been deployed. Each sensor applying *ETT* cooperatively monitor the target in a distributed manner. The network initialization phase aims to detect the event of target entering monitoring region. The boundary sensor that firstly detects the target would play the master role. In *Target Detection* phase, the master sensor aims to select some sensors participating in the tracking task for satisfying the user defined tacking accuracy. Here in, as shown in Fig. 9, three strategies are proposed, including *Target-Not-Detected Neighbor Selection* (*TND-NS*), *Target-Detected Neighbor Selection* (*TD-NS*) and *Integrated Neighbor Selection* (*I-NS*) strategies. Then the Master Handoff phase aims to cope with the target movement problem. As shown in Fig. 9, three strategies are also presented in the *Master Handoff* phase. The *ETT* mechanism can be treated as three independent algorithms, according to the particular strategies that are applied to the *Target Detection* and *Master Handoff* phases. As shown in Fig. 9, in conceptual level, a vertical combination of the proposed strategies, each chosen from the horizontal level, can treated as an independent algorithm of *ETT* mechanism.

The first algorithm, called *Target-Not-Detected Neighbor Target Tracking (TND-TT)* algorithm, mainly applies the *TND-NS* strategy in *Target Detection* phase and *MH-TND* strategy in *Master Handoff* phase. The second and third algorithms, called *Target-Detected Neighbor Target Tracking* (*TD-TT*) and *Integrated Target Tracking (I-TT)* respectively. However, as shown in Fig. 9, the *TD-TT* algorithm mainly applies the *TD-NS* strategy in *Target Detection* phase and *MH-TD* strategy in *Master Handoff* phase, while the *I-TT* algorithm mainly applies the *I-NS* strategy in *Target Detection* phase and *MH-I* strategy in *Master Handoff* phase.

TABLE II
SIMULATION PARAMETERS

Parameter	Description
Monitoring area	400m×400m
Number of Sensor Nodes	1000~2000
Sensing Range	20m
Communication Range	40m
required tracking accuracy	1/10, 1/8, 1/6,1/4, 1/2
Target Velocity	2~10m/s
Working Energy Cost	70.01mJ/s
Switching Energy Cost	11.2mJ/time
Sleeping Energy Cost	42μJ/s

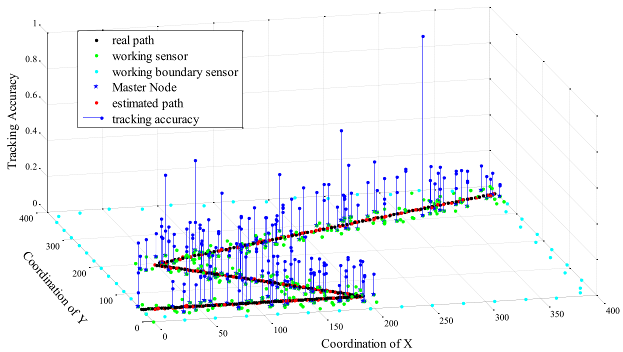

Fig. 10. Tracking result by applying *I-TT* in Scenario 1 with zigzag path.

VI. SIMULATION

This section compares the performance of the related work Face Tracking [16] and the proposed *ETT* algorithm in terms of tracking accuracy and energy consumption. The proposed *ETT* associated with three different strategies are called *TND-TT, TD-TT* and *I-TT*.

A. Simulation Environment

The following illustrates the parameters considered in the simulation environment. The sensing and communication ranges of each sensor node are set by 20 and 40 meters, respectively. In our simulation, sensors are deployed with a guarantee of full coverage in the monitored area sized 400m × 400m. The initial location of the mobile target is generated randomly and the velocity of the mobile target is set between 2∼10m/s. Recall that, the tracking deviation is defined by the max distance between the points of restricted area according to the Equ. (1). The user defined quality of tracking, as denoted by Q_{req}, is set by 1/10, 1/8, 1/6, 1/4. For example, if the user requested quality of tracking is 10m, then the Q_{req} will have value 1/10. Table 2 lists the parameters of simulation environment. To further investigate the performance of the four mechanisms, two scenarios were considered. In the first and second scenarios, the target moves along zigzag and snake-like trajectories, respectively.

B. Simulation Results

Fig. 10 investigates the tracking accuracy of the proposed *I-TT* approach. The number of sensors in the region is 1250 nodes and the Q_{req} is set by 1/8. In Fig. 10, the proposed

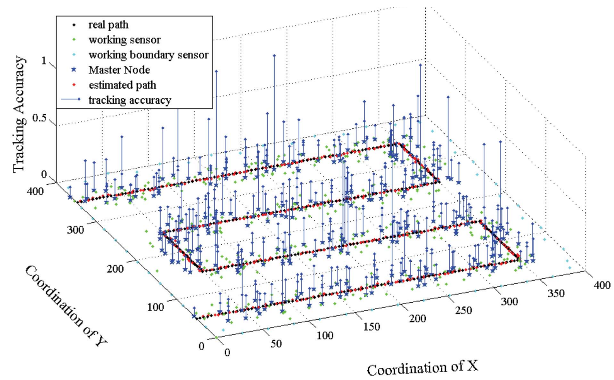

Fig. 11. Tracking result by applying *I-TT* in Scenario 2 with snake-like path.

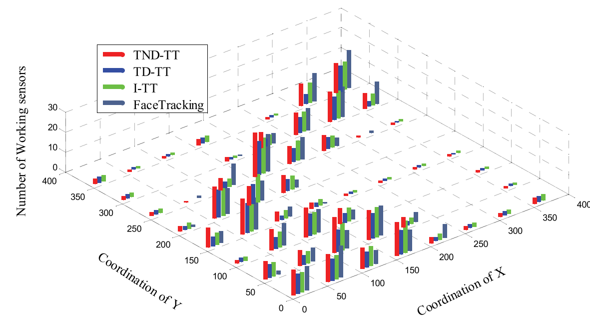

Fig. 12. Number of working sensors with zigzag path.

I-TT mechanism is applied and a lot of sensors which are closed to the target locations play the role of working sensors. These working sensors contribute tracking accuracies and their contributions are represented by the master nodes. That is, a blue point in Fig. 10 represents the master node while a blue line represents the tracking accuracy obtained from the master node and its corresponding working sensors. In Fig. 10, it is observed that all tracking accuracies are higher than the user required value $Q_{req} = 1/8$. This snapshot of the execution of *I-TT* also indicates that the proposed *I-TT* guarantees the required tracking quality.

Similar to Fig. 10, Fig. 11 gives the snapshot of the execution of the proposed *I-TT* mechanism. The major difference between Figs. 10 and 11 is that the target moves along the snake-like trajectory in Fig. 11. As shown in Fig. 11, all tracking accuracies are higher than the user required value $Q_{req} = 1/8$. This snapshot also verifies that even the target moves everywhere of the monitoring region, the proposed *I-TT* mechanism always guarantees the required tracking quality.

Fig. 12 compares the number of working sensors awakened by the proposed *TND-TT*, *TD-TT* and *I-TT* mechanisms, and the Face Tracking mechanism using the zigzag scenario. Compared with the other two mechanisms, the *TND-TT* and Face Tracking wake up a higher number of working sensors. Although the proposed *TND-TT* mechanism has higher number of working sensors for guaranteeing the required tracking quality, it has low total energy consumption. It occurs because the tracking accuracy obtained by applying *TND-TT* algorithm, which satisfies the user's requirement, is lower than that

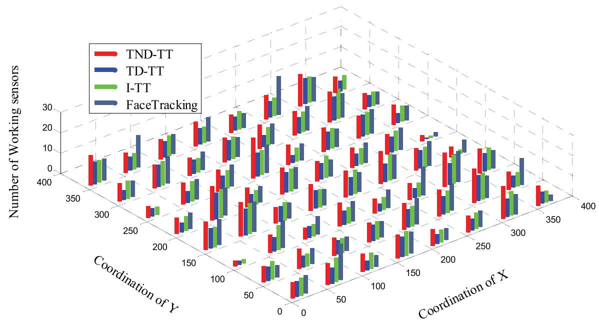

Fig. 13. Number of working sensors with snake-like path.

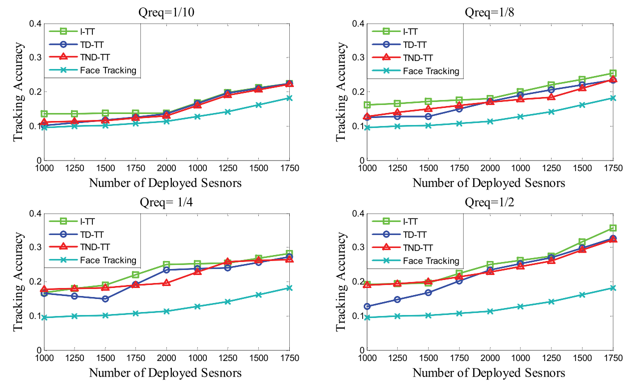

Fig. 14. Performance comparison of the four mechanisms, in scenario 2, in terms of tracking accuracy. The number of deployed sensors ranges from 1000 to 2000.

TABLE III
EFFICIENCY INDEX AND FAIR INDEX

	Efficiency Index		Fair Index	
	Scenario 1	Scenario 2	Scenario 1	Scenario 2
TND	0.03956	0.03833	0.05890	0.06122
TD	0.04341	0.03714	0.06016	0.06100
I-TT	0.05882	0.06002	0.06312	0.06541
Face Tracking	0.03875	0.03786	0.05640	0.05989

by applying *TD-TT* and *I-TT*. Therefore, *TND-TT* has lower frequency of executing Hand-off operations than *I-TT* and *TD-TT*, reducing the energy consumption for communicating and switching state. This fact can be verified in the later experiment.

On the other hand, the proposed *TD-TT* has the lowest number of working sensors than the other three mechanisms. This occurs because that the proposed *TD-TT* mechanism selects working sensors from the neighboring set $S_{master}^{N,D}$ which has a smaller number of elements than the set $S_{master}^{N,ND}$. It further results in a small number of working sensors.

Similar to Fig. 12, Fig. 13 compares the number of working sensors awakened by the four mechanisms in scenario 2. It's observed that the trajectory of the mobile target has little impact on the performance of the four mechanisms in term of the number of working sensors. As shown in Fig. 13 with the snake-like path, *TND-TT* and Face Tracking mechanisms have a higher number of working sensors while *TD-TT* has the lowest one. Furthermore, we can observe that the working sensors have a relatively uniform distribution in the region, compared with working sensors in scenario 1.

From the experiments, we observed that the tracking accuracy is increased with the number of working sensors. However, a large number of working sensors consume more energy for executing sensing and communication tasks. Table 3 further investigates the efficiency index which considers the two parameters, including tracking accuracy and energy consumption. The tracking efficiency, denoted by $I_{efficiency}$, is defined as the tracking accuracy obtained by consuming each unit of energy. Let ζ denote the total energy consumption of all Master Nodes and working sensors, for executing detecting, communication and state switching. Let QoT_i denote the tracking accuracy that was achieved by the ith Master Node. The tracking efficiency was measured according to the Efficiency Index depicted in Equ. (22).

$$I_{efficiency} = \frac{\sum_i^n qot_i}{\zeta} \tag{22}$$

A high efficiency index of a mechanism indicates that the mechanism can obtain better tracking quality with little energy consumption. As shown in Table 3, the proposed mechanism *I-TT* outperforms the other approaches. It occurs because applying the *I-TT* results in higher tracking accuracy and smaller number of working sensors obtained of each Master Node, as compared with the other three mechanisms.

In addition to the comparison of efficiency index, Table 3 also investigates the fairness index values of the compared four mechanisms. The lifetime of a WSN significantly depends on the energy balance which can be measured by the *Fairness Index*. Let x_i denote the energy consumption of sensor s_i. The Fairness Index, denoted by $I_{fairness}$, can be calculated by Equ. (23).

$$I_{fairness} = \frac{\left(\sum_{i=1}^n x_i\right)^2}{n\sum_{i=1}^n x_i^2} \tag{23}$$

As shown in Table 3, all mechanisms applied in scenario 2 have higher Fairness Index values than those applied in scenario 1. This occurs because the snake-like path almost covers the whole monitoring area, which results in more sensors participating in tracking task. Therefore, energy consumptions of all sensors are relatively fair. In the same scenario, mechanism *I-TT* outperforms the other mechanisms in terms of the Fairness Index. This occurs because the Handoff procedure of *I-TT* mechanism was triggered frequently for achieving higher tracking accuracy. Therefore, the proposed *I-TT* prefers to select different working sensors, which can avoid some sensors remaining in working state for a long time period.

Fig. 14 compares the tracking accuracy of the four mechanisms with different number of deployed sensors and various values of Q_{req}. As shown in Fig. 14, the proposed *I-TT* mechanism outperforms the other three mechanisms in all cases. It occurs because that *I-TT* integrates advantages of *TND-TT* and *TD-TT* and the awakened sensors have more contributions than the other three mechanisms in terms of

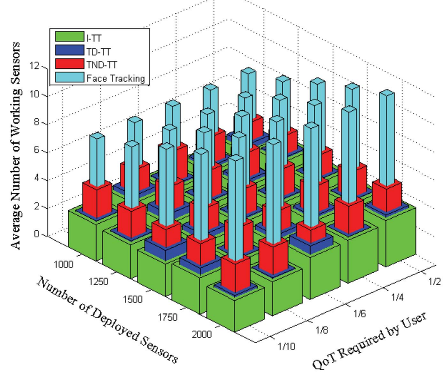

Fig. 15. Average number of working sensor versus number of deployed sensors and Q_{req}. The scenario 2 is applied.

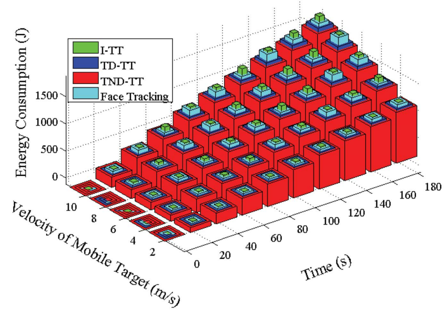

Fig. 16. Energy cost versus velocity and time ($N = 1250$, $Q_{req} = 1/8$).

tracking accuracy. On the other hand, the Face Tracking mechanism has the lowest tracking accuracy because that it uses faces to represent the tracking accuracy. Meanwhile, Face Tracking does not take into consideration the tracking quality required by user. Therefore, the tracking accuracy achieved by the Face Tracking mechanism does not change with the Q_{req}, as shown in Fig. 14. In case of small numbers of deployed sensors, the proposed *TND-TT* mechanism has better performance than the proposed *TD-TT*. On the contrary, the proposed *TND-TT* mechanism has lower tracking accuracy than *TD-TT* in case of a big number of deployed sensors. This occurs because the *TND-TT* mechanism will have a larger number of sensors in neighboring set $S_{master}^{N,ND}$ to be selected for improving the tracking accuracy, when the number of deployed sensors is small. However, when the number of deployed sensors is large, *TND-TT* does not obtain significant benefit from the number of candidates. It's also observed that, as shown in Fig. 14, when the number of deployed sensors is small, a high Q_{req} cannot be achieved by the four mechanisms, which indicates that tracking accuracy largely depends on the number of deployed sensors.

Fig. 15 compares the four mechanisms in terms of the average number of working sensors that are required for achieving the Q_{req}. As shown in Fig. 15, when the number of deployed sensors increases, the numbers of working sensors obtained by applying the proposed three mechanisms decrease. On the contrary, the Face Tracking mechanism requires more working sensors when the number of deployed sensors grows. This occurs because that the average number of sensors in a face increases with the number of deployed sensors. In general, *I-TT* and *TD-TT* outperform *TND-TT* and Face Tracking mechanisms in terms of average number of working sensors.

Fig. 16 compares four mechanisms in terms of total energy consumption. In this experiment, we aim to observe the energy cost dedicated for the time period that the target has entered the monitoring region. That is, the energy cost in the time period that the target does not appear is ignored. As shown in Fig. 16, the four mechanisms have similar results that the total energy consumption is increased with the target velocity. In particular, the *TND-TT* outperforms the other two mechanisms in terms of total energy consumption. This occurs because *I-TT* and *TD-TT* always have higher tracking accuracy and smaller restricted area, which lead to a higher frequency of triggering the Master Handoff Phase. In addition, the *TND-TT* has lower energy consumption than Face Tracking because that the Face Tracking wakes up a higher average number of working sensors.

VII. Conclusion

This paper proposed a target tracking mechanism *ETT*, which aims at prolonging the network lifetime of WSN under the criteria that the supported tracking quality satisfies user requirement. The proposed *ETT* consists of three phases. The first phase wakes up as few as possible the boundary nodes such that they can construct a barrier for the boundary surveillance. As a result, when there is no target in the region, only a few boundary sensors stay in working state. These sensors can detect the target crossing the boundary and entering the monitoring region. The detected boundary sensors will initiate the second phase, which aims to further wake up as few as possible neighboring sensors but guarantee the user defined tracking quality. As the target moves out the restricted area formed in second phase, the Mater Handoff Phase will be triggered to cope with the movement problem. Since the proposed *ETT* is operated in a distributed manner, the target tracking operations can still be maintained even though some sensors are failure or newly deployed. Compared with the existing related work, the proposed *ETT* significantly reduces the energy consumptions of sensors and hence prolongs the network lifetime of WSNs while the requirement of tracking quality defined by user can be satisfied.

References

[1] C.-Y. Lin, W.-C. Peng, and Y.-C. Tseng, "Efficient in-network moving object tracking in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 8, pp. 1044–1056, Aug. 2006.

[2] T. He *et al.*, "Achieving real-time target tracking using wireless sensor networks," in *Proc. 12th IEEE RTAS*, San Jose, CA, USA, Apr. 2006, pp. 37–48.

[3] H. Zhu, M. Li, Y. Zhu, and L. M. Ni, "HERO: Online real-time vehicle tracking," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 5, pp. 740–752, May 2009.

[4] P. Vicaire *et al.*, "Achieving long-term surveillance in VigilNet," *ACM Trans. Sensor Netw.*, vol. 5, no. 1, 2009, Art. ID 9.

[5] G. Wang, M. Z. A. Bhuiyan, and L. Zhang, "Two-level cooperative and energy-efficient tracking algorithm in wireless sensor networks," *Concurrency Comput., Pract. Exper.*, vol. 22, no. 4, pp. 518–537, 2010.

[6] E. Amaldi, A. Capone, M. Cesana, and I. Filippini, "Design of wireless sensor networks for mobile target detection," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 784–797, Jun. 2012.

[7] G. Y. Keung, B. Li, and Q. Zhang, "The intrusion detection in mobile sensor network," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1152–1161, Aug. 2013.

[8] R. Sarkar and J. Gao, "Differential forms for target tracking and aggregate queries in distributed networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1159–1172, Aug. 2013.

[9] K. Ramya, K. P. Kumar, and V. S. Rao, "A survey on target tracking techniques in wireless sensor networks," *Int. J. Comput. Sci. Eng. Surv.*, vol. 3, no. 4, pp. 93–108, 2012.

[10] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, "On target tracking with binary proximity sensors," in *Proc. 4th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Los Angeles, CA, USA, Apr. 2005, pp. 301–308.

[11] Z. Zhong, T. Zhu, D. Wang, and T. He, "Tracking with unreliable node sequences," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1215–1223.

[12] J. Chen, K. Cao, Y. Sun, and X. Shen, "Adaptive sensor activation for target tracking in wireless sensor networks," in *Proc. IEEE ICC*, Dresden, Germany, Jun. 2009, pp. 1–5.

[13] Q. Ren, J. Li, and H. Gao, "TPSS: A two-phase sleep scheduling protocol for object tracking in sensor networks," in *Proc. IEEE Int. Conf. MASS*, Macau, China, Oct. 2009, pp. 458–465.

[14] Z. Wang, E. Bulut, and B. K. Szymanski, "Distributed target tracking with directional binary sensor networks," in *Proc. IEEE GLOBECOM*, Honolulu, HI, USA, Nov./Dec. 2009, pp. 1–6.

[15] Z. Wang, E. Bulut, and B. K. Szymanski, "Distributed energy-efficient target tracking with binary sensor networks," *ACM Trans. Sensor Netw.*, vol. 6, no. 4, 2010, Art. ID 32.

[16] G. Wang, M. Z. A. Bhuiyan, J. Cao, and J. Wu, "Detecting movements of a target using face tracking in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 939–949, Apr. 2014.

Cuijuan Shang received the B.S. degree from Zhengzhou University, China, in 2008, and the M.S. degree from the Shandong University of Technology, in 2011.

She is currently an Assistant with the School of Computer and Information Engineering, Chuzhou University, Anhui, China. Her current research interests include wireless sensor networks, healthcare, and Internet of Things.

Guilin Chen received the B.S. degree from Anhui Normal University, China, in 1985, and the M.S. degree from the Hefei University of Technology, in 2007.

He is currently a Professor with the School of Computer and Information Engineering, Chuzhou University, Anhui, China. His current research interests include cloud computing, wireless networks, healthcare, and Internet of Things.

Chengchao Ji received the B.S. degree in computer science from Southwest University for Nationalities, China, in 2001, and the M.Sc. degree in computer science from Anhui University, China, in 2008. He is currently pursuing the Ph.D. degree at the School of Telecommunication and Information Engineering, Nanjing University of Posts and Telecommunications, China. He is also an Associate Professor at the School of Computer and Information Engineering, University of Chuzhou, Anhui, China. His current research interests include medium access control, routing, and roadside unit placement in vehicular ad hoc networks.

Chih-Yung Chang (M'05) received the Ph.D. degree in computer science and information engineering from National Central University, Zhongli, Taiwan, in 1995.

He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan. His current research interests include Internet of Things, wireless sensor networks, ad hoc wireless networks, and long-term evolution broadband technologies. He served as an Associate Guest Editor of several SCI-indexed journals, including the *International Journal of Ad Hoc and Ubiquitous Computing* from 2011 to 2014, the *International Journal of Distributed Sensor Networks* from 2012 to 2014, *IET Communications* in 2011, *Telecommunication Systems* in 2010, the *Journal of Information Science and Engineering* in 2008, and the *Journal of Internet Technology* from 2004 to 2008.

Dr. Chang was an Area Chair of the IEEE International Conference on Advanced Information and Applications in 2005, the Taiwan Academic NETwork Conference in 2000 and 2010, the IEEE International Symposium on Wireless IP in 2005, the IFIP International Conference on Embedded And Ubiquitous Computing in 2005, the IEEE International Conference on Information Technology: Research and Education in 2005, and the IEEE AINA in 2008, the Program Co-Chair of the IEEE International Workshop on Multimedia Network Systems and Applications in 2005, the Mobile and Ubiquitous Technologies Enhanced Learning Conference in 2006, the Workshop on Wireless, Ad Hoc, and Sensor Networks in 2007, the ACM International Workshop on Sensor, Ad Hoc, and Mesh Networks in 2008, the IEEE International Workshop on Ad Hoc and Ubiquitous Computing in 2008, the International Conference on University Basic Computers Education and e-Learning in 2010 and 2011, the Workshop Co-Chair of the International Workshop on Mobile Systems, E-commerce and Agent Technology in 2003 and 2004, the IEEE International Workshop on Information Networking and Applications in 2005, the International Computer Symposium in 2008, the National Computer Symposium in 2009, and the IEEE International Workshop on Vehicular Communications, Networks, and Applications in 2009, and the Publication Chair of MSEAT in 2005 and the International Conference on Sharable Content Object Reference Model in 2006.