Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

HierTrack: an energy-efficient cluster-based target tracking system for wireless sensor networks*

Zhi-bo WANG[1], Zhi WANG[‡1], Hong-long CHEN[1,2], Jian-feng LI[1], Hong-bin LI[1], Jie SHEN[1]
([1]*Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China*)
([2]*College of Information and Control Engineering, China University of Petroleum, Qingdao 266580, China*)
E-mail: {zbwang, wangzhi, hlchen, jfli, hbli, shenjie}@iipc.zju.edu.cn
Received Nov. 9, 2012; Revision accepted Feb. 19, 2013; Crosschecked May 13, 2013

Abstract: Target tracking is a typical and important application of wireless sensor networks (WSNs). Existing target tracking protocols focus mainly on energy efficiency, and little effort has been put into network management and real-time data routing, which are also very important issues for target tracking. In this paper, we propose a scalable cluster-based target tracking framework, namely the hierarchical prediction strategy (HPS), for energy-efficient and real-time target tracking in large-scale WSNs. HPS organizes sensor nodes into clusters by using suitable clustering protocols which are beneficial for network management and data routing. As a target moves in the network, cluster heads predict the target trajectory using Kalman filter and selectively activate the next round of sensors in advance to keep on tracking the target. The estimated locations of the target are routed to the base station via the backbone composed of the cluster heads. A soft handoff algorithm is proposed in HPS to guarantee smooth tracking of the target when the target moves from one cluster to another. Under the framework of HPS, we design and implement an energy-efficient target tracking system, HierTrack, which consists of 36 sensor motes, a sink node, and a base station. Both simulation and experimental results show the efficiency of our system.

Key words: Wireless sensor networks, Cluster, Energy efficiency, Target tracking, Scalability, Real-time data routing
doi:10.1631/jzus.C1200318 **Document code:** A **CLC number:** TP393

1 Introduction

Wireless sensor networks (WSNs), consisting of a large number of sensor nodes, have attracted extensive attention in recent years in both industry and academia. With the fast development of micro-electro-mechanical system (MEMS) and digital signal processing, sensor nodes are becoming increasingly smaller and less expensive (Akyildiz *et al.*, 2002). Therefore, it is possible to deploy large-scale sensor networks in areas of interest. Compared with wired networks, WSNs have many unique features, such as easy deployment, self-organization, and wireless communication. However, the resources of sensor nodes are quite limited, such as energy, computation, and communication (Akyildiz *et al.*, 2002; Karaca and Sokullu, 2012), due to the distributed nature and unattended deployment.

Target tracking is the basis of many important applications, such as intruder detection, disaster response, emergency rescue, and battlefield surveillance. A good example is in the case of intruder detection where sensors are deployed along the boundary of countries. Target tracking makes it possible to

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 61273079), the State Key Laboratory of Industrial Control Technology (Nos. ICT1206 and ICT1207), the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA06020300), and the Key Laboratory of Wireless Sensor Network & Communication of Chinese Academy of Sciences (No. 2011001)

detect and track every intruder entering the country at all times.

In general, in target tracking one needs to detect targets and estimate their locations when they move within the network, and also forward the estimated locations in a timely manner to the base station. Since energy is the most limited resource in WSNs, many algorithms have been proposed for energy-efficient target tracking in WSNs. Zhao *et al.* (2002) proposed an information-driven dynamic sensor collaboration mechanism for target tracking, in which the next tasking sensor is selected based on maximizing information utility and minimizing resource cost. An entropy-based greedy heuristic approach was proposed in Wang *et al.* (2004) to sequentially select tasking sensors to reduce localization uncertainty. These algorithms select only one tasking node at each time step. Zhang and Cao (2004) developed a dynamic convoy tree-based collaboration (DCTC) framework to detect and track the mobile target. DCTC relies on a tree structure called the convoy tree to benefit sensor collaboration among multiple nodes surrounding the target. Yang and Sikdar (2003) proposed a distributed prediction tracking protocol which activates three sensors at each time step to form a cluster for target tracking. Yang *et al.* (2007) proposed an adaptive dynamic cluster-based tracking (ADCT) protocol in which cluster heads (CHs) are selected dynamically and sensor nodes surrounding the target are waken up to construct clusters as the target moves in the network.

Most existing solutions focus on accurate and energy-efficient tracking of targets, but they do not consider how to forward the estimates of target locations to the base station in a timely manner, which is another important objective of target tracking. For example, dynamic clustering algorithms, such as ADCT, are energy-efficient for target tracking, but the localization results cannot be sent back to the base station in real time. Although many solutions have been proposed for target tracking, few of them have been tested and implemented on real systems. It is necessary to propose algorithms that can be used for target tracking in large-scale sensor networks and whose performance can be evaluated on a real target tracking system.

In this paper, we present a scalable cluster-based target tracking framework, namely the hierarchical prediction strategy (HPS), for energy-efficient and real-time target tracking in large-scale WSNs. As shown in Fig. 1, sensor nodes are organized into clusters according to any suitable clustering protocol (e.g., the popular Leach protocol), which provides a hierarchical structure that benefits network management and data routing. Each cluster consists of one CH and many member nodes. By constraining communications within each cluster, network congestion can be reduced and wireless bandwidth can be saved so that packets can be delivered with better guarantee. As a target moves in the network, CH predicts the trajectory of the target and selectively activates nearby nodes to perform sensor collaboration for accurate target localization. The estimated locations of the target can be routed to the base station via the backbone composed of the CHs. A soft handoff scheme is also proposed to smoothly track the target when it moves from one cluster to another. Based on the HPS, we further design and implement a real target tracking system, HierTrack, which consists of a sink node, a base station, and 36 sensor motes.

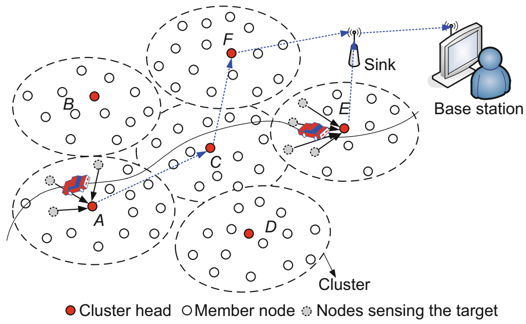


Fig. 1 Demonstration of our cluster-based target tracking algorithm for wireless sensor networks

The main contributions of this paper are summarized as follows:

1. We propose a cluster-based target tracking framework for large-scale sensor networks. The cluster-based hierarchical architecture benefits not only sensor management and collaboration for target tracking, but also real-time data routing to the base station.

2. We propose a soft handoff scheme that guarantees smooth tracking of the target when it moves from one cluster to another.

3. We design and implement a real target tracking system, HierTrack, for WSNs.

2 Related work

The problem of target tracking for WSNs has received considerable attention from various perspectives, and many protocols have been proposed. Karakaya and Qi (2011) proposed a distributed target localization algorithm using a progressive certainty map. Zhang (2011) proposed an adaptive control and reconfiguration scheme for multi-target tracking. He *et al.* (2006a; 2006b) designed and implemented a VigilNet system for energy-efficient and real-time target tracking. Chen *et al.* (2011) and Zhang *et al.* (2012) proposed distributed scheduling algorithms for binary sensor networks and acoustic sensor networks, respectively. Recently, Fayyaz (2011) and Demigha *et al.* (2012) summarized the target tracking techniques in WSNs.

Existing work on target tracking focuses mainly on improving energy efficiency. Demigha *et al.* (2012) reviewed the work on improving energy efficiency in collaborative target tracking in WSNs. You *et al.* (2008) developed a model to predict the mobility estimation error and proposed an energy-saving method by utilizing additional sensors on mobile targets. In Yeow *et al.* (2007), a hierarchical Markov target tracking algorithm was proposed to conserve energy through dynamic spatial and temporal management of sensors based trajectory prediction. Ling *et al.* (2011) used a Kalman filter to predict the next state of the target and select a subset of sensors in advance to track targets. Xu and Qi (2008) proposed to migrate a mobile agent to track targets based on a linear prediction model.

Recently, cluster structure has gradually been used for sensor collaboration in target tracking. Yang *et al.* (2007) dynamically selected sensor nodes surrounding the target to form a cluster to estimate the location of the target. In Chen *et al.* (2004), the active CH wakes up all sensor nodes around it to form a cluster for local sensor collaboration and target location estimation. However, this protocol suffers from the problem caused by the failure of CHs. Hence, once a CH does not work due to mechanical failure or power outage, the protocol cannot work properly and may lose the target during the tracking process. Prediction techniques are usually embodied in tree- and cluster-based tracking protocols to predict the next state of the target.

While the previous target tracking algorithms are quite useful, they are not suitable for target tracking in large-scale WSNs. They focus mainly on energy efficiency, and do not emphasize the other two important objectives of target tracking, network management and real-time data routing.

3 System model

In this section, we present the system model including the network model, the target motion model, and the localization model.

3.1 Network model

We assume that a large-scale WSN is deployed in a two-dimensional terrain where sensor nodes are randomly deployed following Poisson distribution with density of λ. Given an area of A in the area of interest, the probability of having k sensor nodes in this area is

$$\Pr(N(A)=k)=\frac{(\lambda A)^k}{k!}\cdot \mathrm{e}^{-\lambda A},\quad k=0,1,..., \tag{1}$$

where $N(A)$ is the number of sensor nodes in area A.

A sensor node usually has several transmission power levels, which correspond to different communication ranges. For example, the sensor motes used in our system, IRIS, have 16 transmission power levels. In this work, we assume that the sensor network is homogeneous and that each sensor node normally works at a predefined transmission power level with the communication range of r_c. The sensor nodes within the communication range of a sensor node are called its neighbor nodes. $N(v_i)=\{v_j|d(l_i,l_j)\le r_\mathrm{c}\}$ is the set of neighbor nodes of sensor node v_i, where $l_i=(x_i,y_i)$ is the location of v_i, $l_j=(x_j,y_j)$ is the location of v_j, and $d(l_i,l_j)$ is the Euclidean distance between v_i and v_j. A sensor node can also work at different transmission power levels in some cases. For example, a sensor node acting as a CH can choose a higher transmission power level to route data to the sink node.

In general, the strength of the signal of a sensor node received from the target reduces when the distance between the sensor node and the target becomes larger. The attenuated disk sensing model is adopted to capture the sensing quality:

$$r_i=\begin{cases}\dfrac{\beta}{d^\alpha(l_i,L)}, & d(l_i,L)\le r_\mathrm{s}, \quad (2\mathrm{a})\\ 0, & d(l_i,L)> r_\mathrm{s}, \quad (2\mathrm{b})\end{cases}$$

where r_i is the strength of the received signal of sensor node v_i, β is the original strength of the emitted signal of the target, α is the path attenuation exponent, r_s is the sensing range, l_i is the location of v_i, L is the target location, and $d(l_i, L)$ is the Euclidean distance between sensor node v_i and the target.

Note that the sensing region of a sensor node v_i, denoted by $R(v_i, r_s)$, is a disk with center l_i and radius r_s. A target will be detected by sensor v_i when it appears in the sensing region $R(v_i, r_s)$. Conversely, only sensor nodes within the distance of r_s of the target can detect the target. Therefore, we call the disk centered at the target and with radius r_s the monitoring region of the target, as any sensor within this region can monitor the target.

Each node is aware of its own location by using the APIT localization scheme (He *et al.*, 2003) or DV-HOP (Niculescu and Nath, 2003). Each node can operate in three states. It can transmit packets, receive packets, and sense the target in active state. In sensing state, it can perform only sensing operation. In sleep state, it sleeps for most of the time and wakes up periodically to sense the target and listen to messages. Since communication operation dominates energy consumption, we consider mainly the energy consumption for communication in this study.

3.2 Target motion model

Assuming the target moves in a two-dimensional plane, the target motion model is described as

$$\boldsymbol{X}_{k+1} = \boldsymbol{F}_k \boldsymbol{X}_k + \boldsymbol{w}_k, \tag{3}$$

where $\boldsymbol{X}_k$ is the target state at the kth time step, $\boldsymbol{F}_k$ is the state transition matrix, and $\boldsymbol{w}_k \sim N(0, \boldsymbol{Q}_k)$ is the process noise following Gaussian distribution with zero mean and variance of $\boldsymbol{Q}_k$. Define $\boldsymbol{X}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]$, where ($x_k$, y_k) is the location of the target, and $\dot{x}_k$ and $\dot{y}_k$ are the speeds in x and y directions, respectively. Since we are considering a constant velocity model, $\boldsymbol{F}_k$ can be described as

$$\boldsymbol{F}_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where T is the sampling time interval.

3.3 Target localization model

Target localization is an important problem for target tracking and many techniques have been proposed in WSNs. Localization, however, is not the main concern of this study. Any localization algorithm suitable to our framework can be adopted. In general, a target can be detected by its nearby sensors. Therefore, we can use the simple centroid algorithm to estimate the location of the target, which is described as follows:

$$\begin{cases} \bar{x} = \dfrac{1}{n_{\mathrm{d}}} \displaystyle\sum_{i=1}^{n_{\mathrm{d}}} x_i, \\ \bar{y} = \dfrac{1}{n_{\mathrm{d}}} \displaystyle\sum_{i=1}^{n_{\mathrm{d}}} y_i, \end{cases} \tag{5}$$

where $(\bar{x}, \bar{y})$ is the estimated location of the target, (x_i, y_i) is the location of sensor node s_i detecting the target, and n_{d} is the number of sensor nodes detecting the target. This localization algorithm is simple and works for all kinds of sensors.

4 Hierarchical prediction strategy

In this section, we present a hierarchical prediction strategy (HPS) to realize energy-efficient and real-time target tracking in large-scale WSNs.

Fig. 2 shows an overview of HPS. The basic idea is described as follows. To support efficient network management and real-time data routing, we use a cluster-based hierarchical structure for a sensor network. To balance the energy consumption on sensor nodes and prolong network lifetime, CHs change periodically and each sensor node has the possibility of becoming a CH. When sensor nodes detect a target, they report sensing results to their CH which estimates the present location, predicts the target trajectory, and selectively activates sensors around the predicted location in advance for sensing the target. To efficiently track a target from one cluster to another, we propose a soft handoff algorithm in HPS, which hands over the tracking task smoothly and avoids wasting energy for unnecessary handoff. We will mainly describe the key components of HPS, initialization, trajectory prediction and sensor selection, and target tracking handoff, in the following subsections. Note that all sensor nodes are involved in initialization and only activated sensor nodes are responsible for target detection. The CHs are

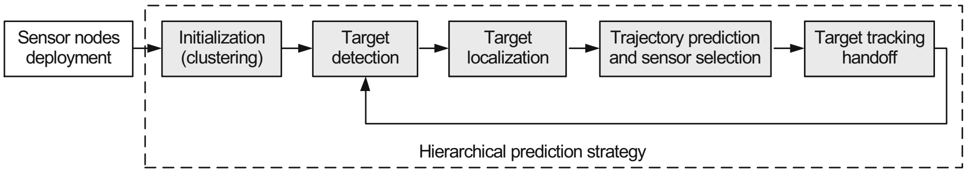


Fig. 2 Framework of the hierarchical prediction strategy (HPS) for wireless sensor networks

responsible for target localization, trajectory prediction and sensor selection, and target tracking handoff.

4.1 Initialization

HPS uses cluster-based network structure to benefit network management and data routing. In order to prolong the network lifetime for target tracking, we have the following requirements for cluster formation:

1. Sensor network is organized into clusters where each cluster consists of one CH and many sensor nodes, and each sensor node belongs to only one cluster.

2. CHs rotate periodically and each sensor node has the possibility of becoming a CH.

This paper is not intended to propose a new clustering algorithm. Any clustering algorithm that satisfies the cluster formation requirements can be used. We find that the Leach protocol, an application-specific clustering protocol, satisfies the cluster formation requirements. Detailed descriptions of the Leach protocol can be found in Heinzelman *et al.* (2002).

In the Leach protocol, CHs directly send data to the sink node, which is not energy-efficient since long range communication consumes much energy. Besides, although a sensor node has multiple transmission power levels, sometimes it is too far away to reach the sink node in one hop. Therefore, to balance energy consumption and real-time data routing, in this study, we organize CHs into a high-level backbone for the sensor network. Hence, CHs in neighboring clusters can communicate with each other directly. For example, as shown in Fig. 1, CHs in clusters A, B, C, D, E, and F form a high-level backbone which can be used to route data quickly to the sink node. The algorithm for constructing the backbone is described as follows.

Each CH maintains a hop count value to the sink node and a neighbor table containing the hop count of its neighboring CHs. The hop count of each CH to the sink node is initialized to a large value, say 1000. The sink node sets its hop count to 0, and broadcasts a BBC message containing its hop count to the sensor network. Let hop_i and hop_j denote the hop counts of sensor nodes s_i and s_j, respectively. Once a CH, say s_i, receives a BBC message from a neighboring CH, say s_j, if $\text{hop}_i > \text{hop}_j + 1$, s_i updates its hop count, hop_i, to $\text{hop}_j + 1$, updates the hop count of s_j, hop_j, in its neighbor table, and broadcasts a new BBC message with the updated hop_i to its neighboring CHs; otherwise, s_i updates only hop_j in its neighbor table. Upon completion of the process, each CH is aware of its hop count to the sink node and also that of its neighboring CHs. Therefore, CHs form a high-level backbone for the sensor network, and data can be routed to the sink node based on the hop count value. However, without a well designed route selection mechanism, a CH may always choose the same neighboring CH to route data, which can quickly deplete the energy of the selected CH. To balance the traffic load and prolong the network lifetime, we leverage the residual energy information to help the route selection to the sink node. Each CH periodically broadcasts its residual energy to its neighboring CHs. When a CH wants to send data, the neighboring CH with a smaller hop count to the sink node and also the maximal residual energy will be selected as the next hop to the sink node.

4.2 Trajectory prediction and sensor selection

Generally speaking, each sensor node works at sleep state and wakes up periodically. In this case, sensor nodes are passive for target sensing. A sensor node may not detect the target as the target moves through its sensing region since it is in sleep state. To solve this problem, we propose a proactive method to predict the target trajectory and activate nodes in the neighborhood of the predicted location in advance for accurate and energy-efficient target tracking.

Since target velocity approximately remains constant within a short period, Kalman filter, commonly used for target tracking, is adopted to predict the target trajectory. The motion model is given in Eq. (3). The measurement model is formulated as follows:

$$\boldsymbol{Z}_k = \boldsymbol{H}_k \boldsymbol{X}_k + \boldsymbol{v}_k, \tag{6}$$

where $\boldsymbol{Z}_k$ is the measured value of the target at the kth step, $\boldsymbol{H}_k$ is the measurement matrix, and $\boldsymbol{v}_k \sim N(0, \boldsymbol{R}_k)$ is the noise of measurement following Gaussian distribution with zero mean and variance $\boldsymbol{R}_k$.

The predicted (a priori) state of the target is

$$\hat{\boldsymbol{X}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{X}}_{k|k}, \tag{7}$$

with the predicted covariance matrix of

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{F}_k \boldsymbol{P}_{k|k} \boldsymbol{F}_k^{\mathrm{T}} + \boldsymbol{Q}_k. \tag{8}$$

The Kalman gain is given by

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k} \boldsymbol{H}_{k+1}^{\mathrm{T}} (\boldsymbol{H}_k \boldsymbol{P}_{k+1|k} \boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k)^{-1}, \tag{9}$$

and the target state is updated as

$$\boldsymbol{X}_{k+1|k+1} = \boldsymbol{X}_{k+1|k} + \boldsymbol{K}_{k+1}(\boldsymbol{Z}_{k+1} - \boldsymbol{H}_k \boldsymbol{X}_{k+1|k}), \tag{10}$$

with the covariance matrix updated as

$$\boldsymbol{P}_{k+1|k+1} = (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{H}_k) \boldsymbol{P}_{k+1|k}. \tag{11}$$

Given $\hat{\boldsymbol{X}}_{k|k} = [x_{k|k}, \dot{x}_{k|k}, y_{k|k}, \dot{y}_{k|k}]^{\mathrm{T}}$, where $(x_{k|k}, y_{k|k})$ is the target location at the kth time step, and $\dot{x}_{k|k}$ and $\dot{y}_{k|k}$ are the target velocities in the horizontal and vertical directions, respectively, and considering a constant velocity model, the state transition matrix is given in Eq. (4), and the observation matrix is given as $\boldsymbol{H}(k) = \boldsymbol{I}_{4\times 4}$. T is the sampling time interval between the sequential time steps. Given the linear motion model, the next location of the target is given as

$$\hat{x}_{k+1|k} = x_{k|k} + T\dot{x}_{k|k}, \tag{12}$$

$$\hat{y}_{k+1|k} = y_{k|k} + T\dot{y}_{k|k}. \tag{13}$$

Let $\hat{L}_{k+1|k} = (\hat{x}_{k+1|k}, \hat{y}_{k+1|k})$ denote the predicted target location. The present working CH selectively activates sensor nodes within a predefined activation range of $\hat{L}_{k+1|k}$. Suppose C_j is the present working cluster. Let r_{a} denote the activation range. Note that only sensor nodes in the present cluster can be activated. Therefore, the set of sensor nodes to be activated in advance is

$$S_{k+1}^{\mathrm{p}} = \{s_i | d(l_i, \hat{L}_{k+1|k}) \le r_{\mathrm{a}}, s_i \in C_j\}, \tag{14}$$

where $l_i = (x_i, y_i)$ is the location of sensor node s_i.

4.3 Tracking handoff

In this subsection, we present our soft handoff scheme which smoothly tracks the target from one cluster to another.

The target may move from one cluster to another. Obviously, the current CH which is responsible for tracking the target should hand over the tracking task to a new CH accordingly. A possible hard handoff scheme is that the tracking task will be handed over to a new CH once the target is predicted to move into another cluster. For example, as shown in Fig. 3, the current location of the target is at point 1 and the predicted location with high probability is at point 2. The head of cluster A hands over the tracking task to the head of cluster B if we use the hard handoff scheme.

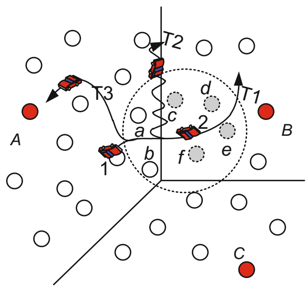


Fig. 3 Illustration of different moving scenarios of the target

However, the hard handoff scheme has several drawbacks. First, the hard handoff scheme may result in frequent handoff and even lose the target in some cases. The target may move randomly in the network. It is possible that the target moves into a new cluster as we predict, and it is also possible that the target moves around in the same cluster. Fig. 3 shows three different movement cases of the target. When the target follows trajectory T2, frequent handoff will take place since the locations of the target are predicted at clusters A and B repeatedly. This unnecessary handoff will waste much energy. When the target follows trajectory T3, target loss may occur because no sensor node in cluster

B can detect the target and all nodes in cluster A are in sleep state. Second, the hard handoff scheme may increase the uncertainty of target localization. For example, when the target moves to location 2 from location 1 following trajectory T1 as we expect, there are six nodes in the sensing region of the target. However, all nodes sensing the target (e.g., c, d, e, f) are at one side of the target, which increases the uncertainty for the centroid localization algorithm. To solve these problems, we propose a soft handoff scheme for smoothly handing over the tracking task from one CH to another.

The soft handoff scheme is described as follows:

Step 1: The cluster head CH_a of the current cluster a predicts the next location of the target and the cluster to which the predicted location belongs. If the predicted cluster, say b, is different from a, go to the next step; otherwise, no handoff is needed.

Step 2: The cluster head CH_b of the predicted cluster b wakes up sensors in the neighborhood of the predicted location and forwards the sensing measurements to CH_a.

Step 3: CH_a collects sensing measurements from its members and CH_b. Based on the measurements, it then estimates the new target location and predicts a new location at the next time step.

Step 4: If the new target location is still in cluster a and the new predicted location is still in b, go to step 3; if the two locations are both in cluster a, go to the next step; otherwise, go to step 6.

Step 5: If the event that both locations are in cluster a occurs thrice continuously, CH_a sends a sleep message to CH_b. CH_b then broadcasts a sleep message to notify its members to go to sleep state to save energy. No handoff is needed.

Step 6: CH_a sends a handoff message to CH_b with the historical locations of the target. After receiving the handoff message, CH_b sends a confirmation message back to CH_a. CH_a then broadcasts a sleep message to notify its members to go to sleep state.

Note that, in step 5, the cooperation between CH_a and CH_b does not break down immediately when both the estimated location and the predicted location are in cluster a. Instead, to avoid frequent cooperation construction and dismissal, the cooperation breaks down only when this event occurs thrice continuously.

5 Simulation results

In this section, we use simulations to evaluate the effectiveness of HPS and compare it with the other two typical target tracking protocols: ADCT (Yang *et al.*, 2007) and DCTC (Zhang and Cao, 2004). The parameters of the simulation setup are listed in Table 1.

Table 1 Simulation setting

Parameter	Value
Area size, A	100 m×100 m
Number of deployed sensor nodes, n	200
Sensing range, r_{s}	10 m
Activation range, r_{a}	15 m
Communication range, r_{c}	20 m
Size of a control message, s_{c}	10 bytes
Size of sensing results reported to CHs, s_{r}	40 bytes

Sensor nodes are randomly deployed and organized into clusters by the Leach protocol. The movement of the target follows the commonly used random waypoint model. In addition, we adopt the energy consumption model proposed in Heinzelman *et al.* (2002). The following metrics are involved in showing the performance results:

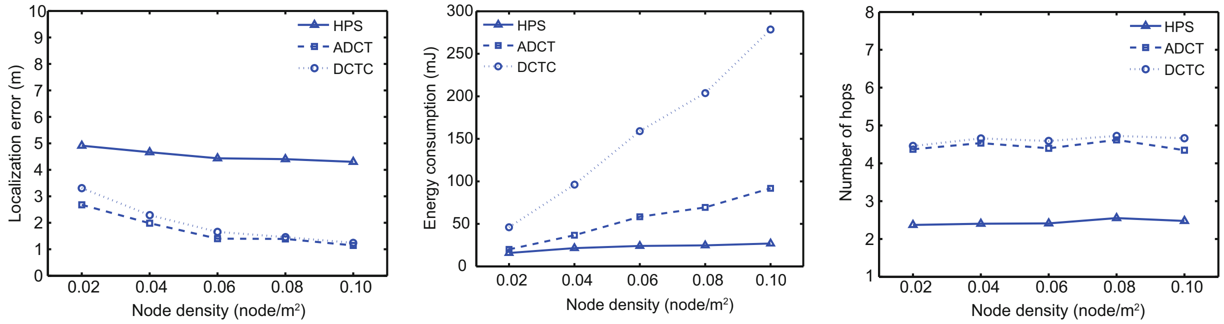
1. Localization error: the average relative distance between the estimated location and true location.

2. Energy consumption: the overall energy consumed for target tracking.

3. Latency: the average number of hops from sources to the sink node for all estimated locations during target tracking.

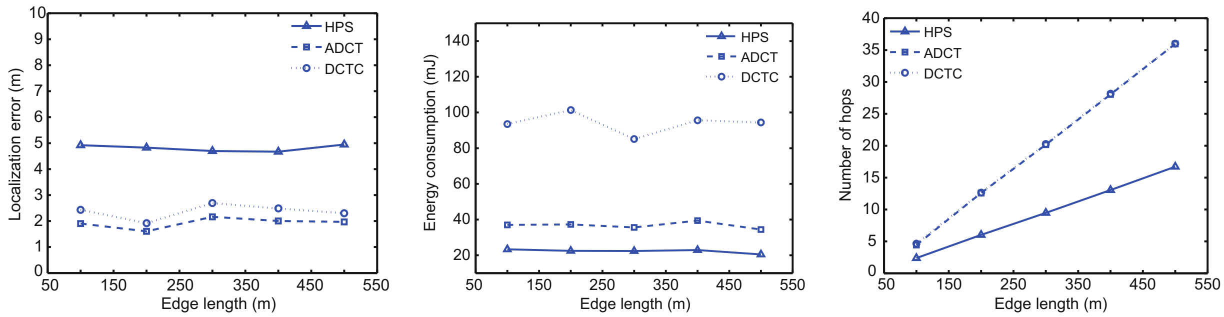
5.1 Effect of node density

Fig. 4 shows the performance comparison among HPS, ADCT, and DCTC when node density changes. The area size is 100 m × 100 m, and the node density changes from 0.02 nodes/m^2 to 0.1 node/m^2 with an interval of 0.02 nodes/m^2. The sink node is at the upper right corner of the area. Fig. 4a shows that the localization errors decrease as the node density increases. This is because by increasing the node density more nodes are involved for localizing the target at each time step, and the more the nodes involved, the smaller the localization error. Note that HPS has a larger localization error compared with the other two protocols. This is due to two facts: first, the number of nodes activated at each



(a) (b) (c)

Fig. 4 Effects of node density on the performance of all algorithms: (a) localization error; (b) energy consumption; (c) latency



(a) (b) (c)

Fig. 5 Effects of network scale on the performance of all algorithms: (a) localization error; (b) energy consumption; (c) latency

time step is usually smaller than those of ADCT and DCTC since the nodes activated in HPS are restricted by the cluster structure; second, the centralized localization algorithm inherently has the characteristic that the localization accuracy improves if more nodes are involved. Even if HPS does not have as good localization accuracy as ADCT and DCTC, the energy consumption and latency performances of HPS are exceptional.

Fig. 4b shows that HPS is more energy-efficient than ADCT and DCTC, and also we can observe that HPS behaves better in terms of scalability. When node density increases, the energy consumption of HPS increases slightly, while the energy consumption of ADCT and DCTC increases greatly. The latency of HPS, as shown in Fig. 4c, is much smaller than that of ADCT and DCTC, which means that the estimated locations of the target can be routed to the base station in a real-time manner by using HPS. We can observe that our algorithm affords a nice tradeoff among energy consumption, latency, and the localization error.

5.2 Effect of network scale

Fig. 5 shows the performance comparison among HPS, ADCT, and DCTC when the network scale changes. The network scale varies from 100 m $\times$ 100 m to 500 m $\times$ 500 m, and the node density remains 0.04 nodes/m^2 at all times.

We can observe that the localization errors and energy consumptions of all algorithms fluctuate slightly when the network scale increases. The performance of HPS is more stable than that of ADCT and DCTC when the network scale increases. Fig. 5c shows that when the network scale increases, the latencies of all algorithms increase accordingly. The reason is that the distance from the source to the sink node increases when the network scale increases. The latency of HPS, however, increases much slower than that of ADCT and DCTC.

It can be concluded that our algorithm is more scalable to node density and network scale. The localization error really depends on the localization algorithm applied. Due to the feature of the centralized localization algorithm, the localization accuracy of our algorithm is not as good as that of the other two typical tracking algorithms, but our algorithm behaves much better with regards to energy efficiency and latency. Therefore, our algorithm affords a nice tradeoff among these metrics and is more suitable for real-time target tracking in large-scale WSNs.

6 System implementation and experiments

Based on our proposed algorithm, we implement a real target tracking system, HierTrack. Our system consists of three parts: a WSN consisting of many sensor nodes, a sink node, and a base station. To evaluate the tracking performance of HierTrack, we use a camera calibration system developed by our lab (Li *et al.*, 2008) to record the real-time ground truth locations of the target.

Fig. 6 shows a snapshot of HierTrack which is built on a 5 m × 5 m platform. A total of 36 sensor nodes are deployed at approximately 0.7 m spacing in a 6 × 6 grid. These sensor nodes are organized into one cluster. Each node is put on an upside down paper cup, which enlarges the distance between the antenna of the node and the platform surface, and then reduces signal reflection of each node. A small automatic robot with one sensor node embodied is called the target, which can move automatically and broadcast messages periodically. The sink node performed by a sensor node and the base station represented by a computer are deployed on the desk close to the platform. A user-friendly interface is designed on the base station to show the status of the network. Users can easily obtain the status of the sensor network, such as which node is sleeping, which node is active, and which node runs out of energy; also, they can see the target tracking results on the user-friendly interface. A camera calibration system deployed on top of the platform can record the real-time locations of the target.

6.1 Platform

In our system, new wireless sensor motes, IRIS, developed by Crossbow, a leading company in wireless sensor platforms, are used as the sensor nodes. It features a low-power microcontroller RF230 and a 250 kb/s, 2.4 GHz, IEEE 802.15.4 wireless communication standard.

Fig. 6 Testbed of the HierTrack system

We program the sensor nodes with NesC and run them on TinyOS, an event-driven operating system developed for wireless sensor platforms. We use the version of TinyOS 2.x to support the IRIS platform. The user-friendly interface written in C# is programmed on the base station.

6.2 Experimental results

We first investigate the localization performance of the system for a static target and then study the tracking performance of the system.

6.2.1 Localization performance

We first investigate the localization performance of the system. As shown in Fig. 7a, three representative grids of grids 1, 2, and 3 are selected. Small squares denote sensor nodes. Grid 1 is surrounded by the least sensor nodes while grid 3 the most.

The target node broadcasts target messages at the center of each grid for 5 min during a period of 60 ms. Experiment at each point is independent of each other. Experiments show that our system behaves very well at communications since no more than three packets are lost at each point. Fig. 7b shows the localization errors and localization standard deviations of these three grids. The localization errors of grids 1, 2, and 3 are 26.4, 22.9, and 18.4 cm, respectively. The standard deviations of results measured in grids 1, 2, and 3 are 15.2, 13.9, and 13.7 cm, respectively. Fig. 7b shows that the localization errors and the standard deviations decrease gradually from grid 1 to grid 3. Hence, the more the nodes by which a target is surrounded, the higher

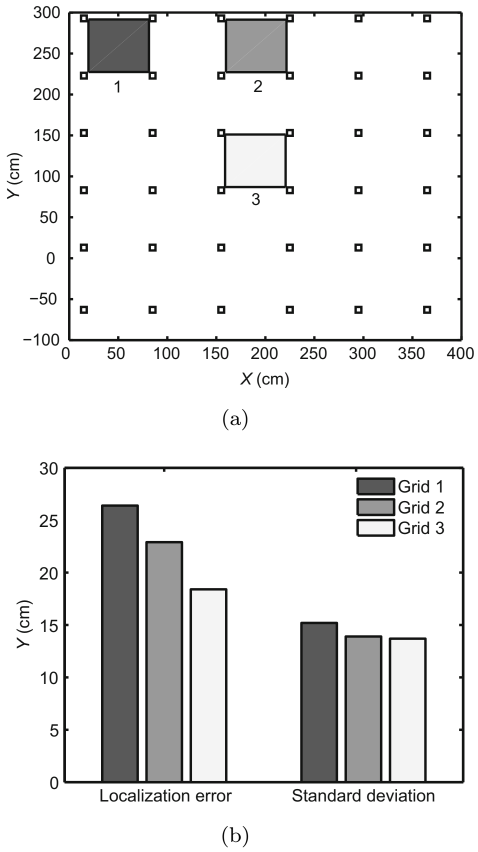


Fig. 7 Localization (a) and localization results (b) of static targets

the accuracy of target localization. Therefore, as a target moves to the center from the boundary, the localization results become increasingly credible and reliable. Although we use the binary sensing model and only two bits of observation of every node to estimate the target location, the system still achieves desirable localization accuracy.

6.2.2 Tracking performance

We also study the tracking performance of the system when the target automatically moves in the network. More than 50 experiments have been performed when the system employs HPS and Naive, respectively. In the Naive scheme, all sensor nodes are working in active state at all times. We find that our system can track the target all the time with a zero miss rate.

Fig. 8 shows the performance comparison between HPS and Naive. We can observe that the localization errors of HPS are close to those of Naive. Fig. 9a shows the tracking results of HPS and Naive in one experiment. Although their real-time tracking results are different, as shown in Fig. 9b, an amazing observation is that we can obtain the same tracking results using the Kalman filter to smooth the original results, which means that users obtain the same target trajectory on the base station and thus the localization accuracy of HPS is the same as that of the Naive method. At the same time, as shown in Fig. 8b, HPS consumes much less energy compared with the Naive scheme. Therefore, we can conclude that HierTrack achieves the same tracking accuracy as the Naive method in a much more energy-efficient way.

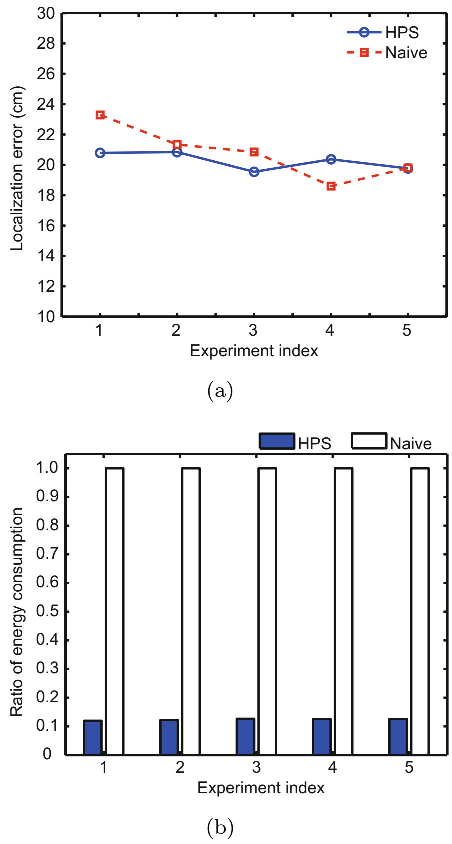


Fig. 8 Comparison of localization errors (a) and energy consumption (b) between HPS and Naive on the HierTrack system

7 Conclusions and future work

Target tracking for large-scale WSNs requires energy efficiency, scalability, and real-time data routing from source nodes to the base station. Most existing target tracking protocols focus on energy

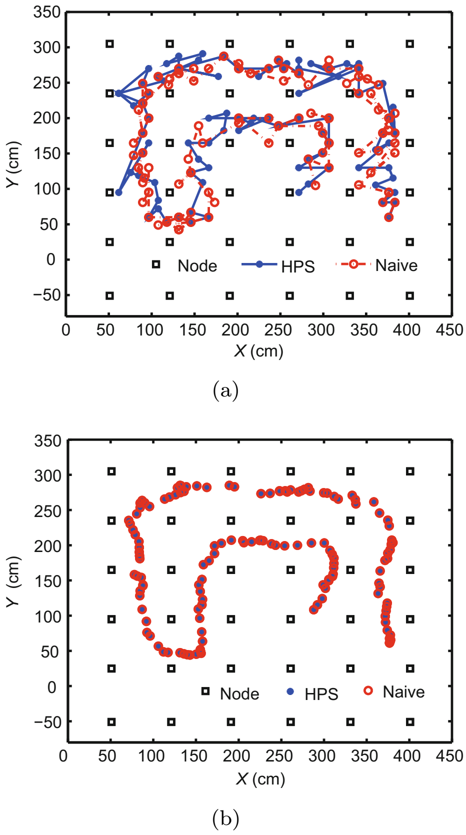


Fig. 9 Comparison of tracking performance before (a) and after (b) smoothing between HPS and Naive

efficiency and little effort has been put into scalability and real-time data routing. In this paper, we presented a scalable cluster-based target tracking framework for energy-efficient and real-time target tracking in large-scale sensor networks. Based on the proposed strategy, we designed and implemented a real target tracking system, HierTrack, consisting of 36 sensor motes, a sink node, and a base station. Simulation results showed that our algorithm exemplifies the tradeoff among energy efficiency, localization accuracy, and latency. Experimental results of the HierTrack target tracking system further validated the efficiency of our algorithm.

While the results presented here represent the latest development on single target tracking, more work needs to be done in this area. The system designed to have good scalability could be extended to more clusters. Therefore, we intend to expand the network size (so that more clusters coexist in the system) to achieve good scalability and cope with the challenges raised by multiple targets tracking.

References

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E., 2002. A survey on sensor networks. *IEEE Commun. Mag.*, **40**(8):102-114. [doi:10.1109/MCOM.2002.1024422]

Chen, J.M., Cao, K.J., Li, K.Y., Sun, Y.X., 2011. Distributed sensor activation algorithm for target tracking with binary sensor networks. *Clust. Comput.*, **14**(1):55-64. [doi:10.1007/s10586-009-0092-0]

Chen, W.P., Hou, J.C., Sha, L., 2004. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Trans. Mob. Comput.*, **3**(3):258-271. [doi:10.1109/TMC.2004.22]

Demigha, O., Hidouci, W.K., Ahmed, T., 2012. On energy efficiency in collaborative target tracking in wireless sensor network: a review. *IEEE Commun. Surv. Tutor.*, **99**:1-13. [doi:10.1109/SURV.2012.042512.00030]

Fayyaz, M., 2011. Classification of object tracking techniques in wireless sensor networks. *Wirel. Sens. Network*, **3**(4):121-124. [doi:10.4236/wsn.2011.34014]

He, T., Huang, C.D., Blum, B.M., Stankovic, J.A., Abdelzaher, T., 2003. Range-Free Localization Schemes for Large Scale Sensor Networks. 9th Annual Int. Conf. on Mobile Computing and Networking, p.81-95. [doi:10.1145/938994.938995]

He, T., Krishnamurthy, S., Luo, L., Yan, T., Gu, L., Stoleru, R., Zhou, G., Cao, Q., Vicaire, P., Stankovic, J.A., *et al.*, 2006a. VigilNet: an integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sens. Networks*, **2**(1):1-38. [doi:10.1145/1138127.1138128]

He, T., Vicaire, P., Yan, T., Luo, L.Q., Gu, L., Zhou, G., Stoleru, R., Cao, Q., Stankovic, J.A., Abdelzaher, T., 2006b. Achieving Real-Time Target Tracking Using Wireless Sensor Networks. 12th IEEE Real-Time and Embedded Technology and Applications Symp., p.37-48. [doi:10.1109/RTAS.2006.9]

Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H., 2002. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel. Commun.*, **1**(4):660-670. [doi:10.1109/TWC.2002.804190]

Karaca, O., Sokullu, R., 2012. A cross-layer fault tolerance management module for wireless sensor networks. *J. Zhejiang Univ.-Sci. C (Comput. & Electron.)*, **13**(9):660-673. [doi:10.1631/jzus.C1200029]

Karakaya, M., Qi, H.R., 2011. Distributed target localization using a progressive certainty map in visual sensor networks. *Ad Hoc Networks*, **9**(4):576-590. [doi:10.1016/j.adhoc.2010.08.006]

Li, Z.N., Li, H.B., Zhang, F., Chen, J.M., Sun, Y.X., 2008. An Indoor Sensor Network System for Pursuit-Evasion Games. 4th IEEE Int. Conf. on Mobile Ad-hoc and Sensor Networks, p.227-233. [doi:10.1109/MSN.2008.32]

Ling, Q., Fu, Y.F., Tian, Z., 2011. Localized sensor management for multi-target tracking in wireless sensor networks. *Inf. Fus.*, **12**(3):194-201. [doi:10.1016/j.inffus.2011.01.003]

Niculescu, D., Nath, B., 2003. DV based positioning in ad hoc networks. *Telecommun. Syst.*, **22**(1-4):267-280. [doi:10.1023/A:1023403323460]

Wang, H., Yao, K., Pottie, G., Estrin, D., 2004. Entropy-Based Sensor Selection Heuristic for Target Localization. 3rd ACM/IEEE Int. Conf. on Information Processing in Sensor Networks, p.36-45. [doi:10.1109/IPSN.2004.1307321]

Xu, Y.Y., Qi, H.R., 2008. Mobile agent migration modeling and design for target tracking in wireless sensor networks. *Ad Hoc Networks*, **6**(1):1-16. [doi:10.1016/j.adhoc.2006.07.004]

Yang, H., Sikdar, B., 2003. A Protocol for Tracking Mobile Targets Using Sensor Networks. Proc. 1st IEEE Int. Workshop on Sensor Network Protocols and Applications, p.71-81. [doi:10.1109/SNPA.2003.1203358]

Yang, W.C., Fu, Z., Kim, J.H., Park, M.S., 2007. An Adaptive Dynamic Cluster-Based Protocol for Target Tracking in Wireless Sensor Networks. 8th Int. Conf. on Advances in Data and Web Management, p.156-167. [doi:10.1007/978-3-540-72524-4_19]

Yeow, W.L., Tham, C.K., Wong, W.C., 2007. Energy efficient multiple target tracking in wireless sensor networks. *IEEE Trans. Veh. Technol.*, **56**(2):918-928. [doi:10.1109/TVT.2007.891480]

You, C.W., Huang, P., Chu, H.H., Chen, Y.C., Chiang, J.R., Lau, S.Y., 2008. Impact of sensor-enhanced mobility prediction on the design of energy-efficient localization. *Ad Hoc Networks*, **6**(8):1221-1237. [doi:10.1016/j.adhoc.2007.11.007]

Zhang, F., Chen, J.M., Li, H.B., Sun, Y.X., Shen, X.M., 2012. Distributed active sensor scheduling for target tracking in ultrasonic sensor networks. *Mob. Networks Appl.*, **17**(5):582-593. [doi:10.1007/s11036-011-0311-9]

Zhang, W.S., Cao, G.H., 2004. DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks. *IEEE Trans. Wirel. Commun.*, **3**(5):1689-1701. [doi:10.1109/TWC.2004.833443]

Zhang, X., 2011. Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking. *IEEE Trans. Autom. Control*, **56**(10):2429-2444. [doi:10.1109/TAC.2011.2163862]

Zhao, F., Shin, J., Reich, J., 2002. Information-driven dynamic sensor collaboration. *IEEE Signal Process. Mag.*, **19**(2):61-72. [doi:10.1109/79.985685]