Peer-to-Peer Networking and Applications (2019) 12:1041–1060
https://doi.org/10.1007/s12083-019-00744-0

A type of energy-efficient target tracking approach based on grids in sensor networks

Chao Sha[1,2] • Lian-hua Zhong[1,2] • Yao Bian[3] • Dan-dan Song[1,2] • Chun-hui Ren[1,2]

Received: 23 January 2018 / Accepted: 12 March 2019 / Published online: 26 April 2019

Abstract

To enhance the reliability as well as the value of sensing data in Wireless Sensor Networks (WSNs), a type of Energy-efficient Target Tracking Approach (ETTA) is proposed in this paper. The sensor network is divided into several virtual grids for distributed tracking and three kinds of states (tracking state, prepared-tracking state and preparing-tracking state) of these grids are also proposed to reduce energy consumption and enhance the accuracy of node localization. Moreover, a tracking recovery strategy is also described in this paper that effectively enhance the robustness of the tracking system. Experiment results show that ETTA has a good performance on target tracking in sensor networks compared to BPS and EMTT.

Keywords Sensor networks · Target tracking · Virtual grids · Energy-efficient · Network lifetime

1 Introduction

Wireless Sensor Networks (WSNs) is a special type of ad-hoc network that is composed of a large number of inexpensive sensors. They are often powered by portable energy supplies and can sense physical events to collect environmental information [1, 2]. Target tracking, whose goal is to detect and track the moving targets, has been one of the most typical applications in WSNs [3, 4]. In some applications of the modern military field, the randomly deployed sensor nodes have been adopted for target tracking. For example, in the military base, with the help of a large number of cheap nodes as well as other precise equipment, it is better to detect, locate and track illegal intrusion targets in real time. In addition, this WSN-based target tracking technology is also widely used in the fields of biological habits monitoring and urban road traffic monitoring. For example, as early as 2005, the researchers from UC Berkeley put 30 sensor nodes on the Great Duck Island to track the behavior of wildlife in real time. Finally, they obtained the most comprehensive data on animal life habit.

In WSNs, the accuracy of localization directly affects the effect of target tracking. Nowadays, there are three types of localization strategies in sensor networks, those are range-based, range-free and hybrid localization methods. In the range-based method, some additional hardware should be installed on nodes, so they are expensive and not suitable for large-scale densely deployed networks. On the other hand, the location accuracy based on the range-free methods are often not high enough. Relatively speaking, the hybrid localization algorithm achieves a compromise between the above two, but its overhead on time and space is much higher. Thus, it is also not suitable for target tracking system with high real-time requirements. Therefore, how to design a lightweight target localization method with high accuracy is the first problem to be solved in this paper.

Different from the discrete event detection method [5], the WSN-based target tracking system often needs continuous monitoring. That is to say, there should always exist nodes that can detect the target along its trajectory. Usually, when the targets move into the network, many sensor nodes have to remain in active mode to track these targets in all potential directions. Nevertheless, nodes are powered by low-cost batteries and should work for a long time in an unattended manner. So, how to reduce the energy consumption of nodes so as

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s12083-019-00744-0) contains supplementary material, which is available to authorized users.

✉ Chao Sha
shac@njupt.edu.cn

[1] School of Computer Science, Software and Cyberspace Security, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, China

[2] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210023, Jiangsu, China

[3] School of Oversea Education, Nanjing University of Posts and Telecommunications, Nanjing 210023, Jiangsu, China

to ensure the network runs for a long time becomes more and more important [6]. In our previous work, we have designed several types of energy replenishment schemes by using some mobile charging vehicles [7, 8]. However, it is undeniable that the wireless charging rate is still slow nowadays, and it is not realistic to charge all nodes periodically. Therefore, in order to reduce the energy consumption of nodes in the process of target tracking, current researches mainly focus on the energy saving strategy of the active nodes.

In general, the locality characteristics of the target tracking process are obvious. So, it is sometimes unnecessary to awaken all the nodes. In most cases, it only needs to wake up some nodes in the area where the target is expected to arrive [9, 10]. On the other hand, if the neighbor node which works in sleeping mode couldn't be waked up in time, the target may be missing. Thus, how to design a reliable sleep scheduling strategy for nodes in the process of target tracking is of great significance to the energy consumption as well as the running effect of the whole network.

In order to solve the above problems, a virtual grid based target tracking strategy is proposed in this paper. Contributions of our work can be concluded as follows.

Firstly, with the help of the collaboration between grids, not all the nodes are required to participate in target tracking. This not only enhances the energy efficiency of the whole network but also reduces the cost of communication.

Secondly, the position of the target can be accurately and timely calculated out by the improved APIT (Approximate Point-In-Triangulation test algorithm) localization algorithm. Thus, the predicted trajectory of the target is almost the same as its real trajectory.

Finally, a tracking recovery mechanism is also proposed that enhances the robustness of our method. Moreover, the target may also be detected in real time when it moves out of the network or moves into the network again.

The remainder of this paper is organized as follows. The related works are described in section 2. And the virtual grids based network model as well as the target localization method are described in section 3 and 4 respectively. In addition, the target tracking strategy based on moving direction prediction is proposed in section 5. Finally, experimental results of ETTA are shown in section 6 and the conclusion is provided in the last section.

2 Related works

As mentioned above, target tracking is a significant application in WSNs so that a lot of scholars have carried out related research [11, 12]. Lots of researches on target tracking are committed to looking for an optimum energy consumption strategy, enhancing the tracking accuracy and reducing the time spending on calculation [13]. Generally, the target tracking methods in WSNs could be classified into three categories: tree-based tracking strategy [14–18]; cluster-based tracking strategy [2, 10, 19–23] and prediction-based tracking strategy [24–30].

2.1 Tree-based tracking strategies

In this type of strategies, sensor nodes are often organized as a tree or graph structure, in which the vertices represent the nodes and the edges represent the links between nodes that can communicate directly. Kung [14] and Liu [15] organized the network as a distributed database by the tree-based structure, which is also called "message-pruning tree". In this structure, each sensor node should register with each other when the target enters or leaves its detection range. Thus, the target can easily be detected. However, the cost on communication of this strategy is much higher. Zhang et al. [16] proposed a dynamic tree-based collaboration method, which was formalized as a multiple moving target optimization problem. A sequence based on some trees with low energy consumption and high coverage was selected out to enhance the accuracy of tracking. In addition, Mehta et al. [17] researched on the location privacy issues in target tracking under this tree-base method. They calculated out a lower bound on the communication overhead of location privacy and discussed the trade-offs between communication cost and location privacy. In addition, Alaybeyoglu et al. [18] proposed an approach to awaken nodes in a real time target tracking system. It formed a tree-based structure to decrease the missing rate of the target alone the predicted trajectory of the target dynamically.

2.2 Cluster-based tracking strategies

In cluster-based strategies, nodes are organized as some clusters, with each cluster consisted of several cluster member nodes and one cluster head. When the cluster members detect the target, they immediately send data to their cluster heads. After collecting the information from all its members, the cluster head could predict the position of the target. The cluster-based cooperation in target tracking is used to promote collaborative data processing, which greatly reduces the energy consumption by managing the scarce resources of the whole networks.

Liao et al. [5] proposed a distributed information compression method by cluster-based structure, which described the measurement uncertainty of target tracking in sensor networks. This kind of leader-based information processing scheme is proved to improve the tracking accuracy. Moreover, Wang et al. [10] proposed an energy saving mechanism in target tracking, which based on dynamic and distributed adaptive clustering with reasonable routing. An intra-cluster optimal routing framework with Particle Filter (PF) was established in this algorithm to predict the position and trajectory of the target. Bernabe et al. [19] proposed a novel and efficient cluster selection approach in target-tracking, which was able to track multiple targets accurately in real-time applications by the camera activation mechanism. Bhatti et al.

[20] proposed a cluster-based target tracking mechanism, which constructed static clusters at the time when the network was firstly deployed. In this algorithm, the size and cluster members of each cluster were fixed, but the cluster head must be in regular rotation among nodes of each cluster. So, the better distribution of energy consumption about all nodes can be ensured in the network. Teng et al. [21] employed a general state evolution model of target tracking to describe the dynamical system without precise location information of the sensors. Compared with the centralized approaches, this algorithm reduced energy consumption as well as the bandwidth occupancy. Similarly, Enayet et al. [22] also proposed a clustering target tracking mechanism, where the cluster heads coordinated with their cluster members and determined the target location precisely by aggregating sensing data from these member nodes. This mechanism was proved to minimize the sensing redundancy of sensor nodes and to maximize the number of sleeping nodes in the networks. Furthermore, Fu et al. [23] proposed an efficient cluster head selection scheme for target tracking, in which the cluster members were dynamically chosen by adopting a greedy on-line decision method. In order to balance the tracking accuracy and energy consumption in Wireless Camera Sensor Networks (WCSNs), this greedy on-line decision method also considered the limited energy of all camera nodes.

2.3 Prediction-based tracking strategies

Prediction-based methods, with a prediction the target trajectory and its next location, only activate special nodes of networks for tracking and rest of nodes remain in sleeping mode for energy saving. Typical target prediction methods include kinematics-based prediction [24–26], dynamics-based prediction [27], and Kalman Filter prediction [28–30]. Kinematics and dynamics are two branches of the classical mechanics. Kinematics describes the target motion without considering circumstances that cause the motion, while dynamics studies the relationship between target motion and its causes.

Jiang et al. [24] presented a prediction-based and sleep scheduling protocol which was an example of kinematics-based prediction. Another example of this type of prediction is the Prediction-based Energy Saving scheme (PES) introduced in [25]. It only used simple models to predict a specific location without considering the detailed moving probabilities. Turgut et al. [26] predicted the next position of the target by the linear predictor, in which the previous location and current location of the target were all taken into account. The prerequisite of their study is that the target has a linear mobility with constant velocity. However, this is often unrealistic because the movement of the target is often uncontrollable. Moreover, Taqi et al. [27] proposed a dynamic prediction-based protocol. They predicted the position that the target was probably moving to by calculating yaw rate and the side force, instead of estimating all the possibilities.

The Kalman Filter model allows the elaboration of an algorithm to estimate the optimal state vector values. It's possible to generate a sequence of state values in each time unit, predicting future states using the current state, and allowing the creation of systems with real-time updates. J.M. Hsua et al. [28] tracked the mobile target by the regression-based prediction and Kalman filter in sensor networks. Similarly, Olfati-Saber et al. [29] presented a distributed version of the Kalman Filter that is applicable to sensor networks with different observation matrices. This enables the sensor network to act as a collective observer. Experiments with target tracking applications show that the distributed version is similar to the centralized version in terms of performance. Feng et al. [30] proposed a dynamic approach for target tracking by an improved Kalman filter, in which the cluster head received the estimations of the target from their respective cluster members, and then, it applied the modified Kalman filter (KF) based on Fisher information matrix for filtering in target tracking.

All of these researches cited above performed target tracking in a continuous observation sensor networks, which estimate the target position by complex computations and estimating the distance from nodes to the target. Although the prediction-based methods introduced above performed target tracking more accurately, they usually resulted in high-energy consumption.

In view of the fact that the current target tracking system consumes a lot of energy, in the proposed strategy, the network is firstly divided into some grids. With the help of the collaboration among different grids, most of the nodes can be in sleeping mode during the target tracking process. This ensures that the network runs steadily for a long time. In addition, to enhance the accuracy of target tracking, an improved APIT localization method is also proposed without additional hardware. Furthermore, most of the current researches do not consider how to discover the target again when it is lost. In our algorithm, the lost target can be found again in time, as long as they are still in the network or re-enter the network. It is worth noting that the proposed method is not only applicable to rectangular networks, but also to networks with irregular shape.

3 Network model

In large-scale sensor networks, the network is often divided into several rectangular grids for cluster building, topology maintenance and routing. For example, "Geographical Adaptive Fidelity" (GAF) [5, 31] is one of the classic grid-based routing algorithms. On the other hand, using several cooperative sensor nodes in grids for target tracking can not only enhance the performance on real-time and accuracy but also reduce energy consumption and data redundancy. Therefore, in the proposed model, the rectangular network with the length of M and width of L is divided into several grids whose lengths and widths are m and l respectively (both M/m and L/l are integers). It is assumed that $(M/m) \times (L/l) = k$. As shown in Fig. 1, if the shape of the network area is irregular, its external rectangle is divided into grids according to the method described above. What's more, some definitions are listed as follows.

3.1 Boundary grid

If the network boundary goes through grid G_u, G_u is a boundary grid, such as G_1 in Fig. 1.

3.2 Neighborhood grid

If G_u and G_v have common edges or vertices, they are neighbor grids of each other, such as G_1 and G_2 in Fig. 1.

N nodes are randomly deployed in the network, and it is assumed that each node knows its geographical coordinate. r_s and r_t are defined as the length of the sensing radius and communication radius respectively. In order to ensure the real-time accuracy of target tracking, each node should be able to communicate with any node located in the neighbor gird directly. That is, $4l^2 + 4\ m^2 \leq r_t^2$.

Obviously, if the shape of the network is irregular, the number of nodes in the boundary grid may be much less than those in other regions. Take the boundary grid G_u as an example, if $Num(G_u) < N/k$, it is regarded that nodes in G_u are not suitable to be clustered separately (e.g., grid G_3 in Fig. 1). $Num(G_u)$ is defined as the total number of nodes in G_u. In this case, for each node S_i in G_u, the average distance between S_i and nodes in the neighborhood grid G_v can be calculated out by formula (2) (the value of $Num(G_v)$ should be no less than N/k). Then, the neighborhood grid (e.g., G_v) with the smallest value of $P(S_i) = \delta E(S_i) + \eta/\sigma(S_i, G_u)^2$ is selected out. Thus, S_i is allocated to this specific neighborhood grid, and the value of $Num(G_v)$ is updated to $Num(G_v) + 1$. As shown in Fig. 1, nodes S_3, S_4, S_5 and S_6 join in the clusters of G_1 and G_4 respectively. G_1 and G_4 are the neighborhood of G_3. Moreover, CH_1 and CH_4 are defined as the cluster headers.

$$\overline{d(S_i, G_v)} = \frac{\sum_{j=1}^{Num(G_v)} \sqrt{(x(u,i)-x(v,j))^2 + (y(u,i)-y(v,j))^2}}{Num(G_v)} \tag{1}$$

In formula (1), $x(u,i)$, $y(u,i)$, $x(v,j)$ and $y(v,j)$ are defined as the vertical and horizontal coordinates of S_i and S_j respectively. Subsequently, nodes in each grid begin to form clusters separately. For each node S_i in G_u, the priority of being the cluster head can be defined by formula (2).

$$P(S_i) = \delta E(S_i) + \eta/\sigma(S_i, G_u)^2 \tag{2}$$

δ and η are the adjustable parameters which vary from 0 to 1. $E(S_i)$ represents the residual energy of S_i. $\sigma(S_i,G_u)^2$ is defined as the variance about the distance between S_i and other nodes in G_u. That is,

$$\sigma(S_i, G_u)^2 = \frac{\sum_{j=1}^{Num(G_u)-1} (x(u,i)-x(u,j))^2 + (y(u,i)-y(u,j))^2}{Num(G_u)-1} - \left(\overline{d(S_i, G_u)}\right)^2 \tag{3}$$

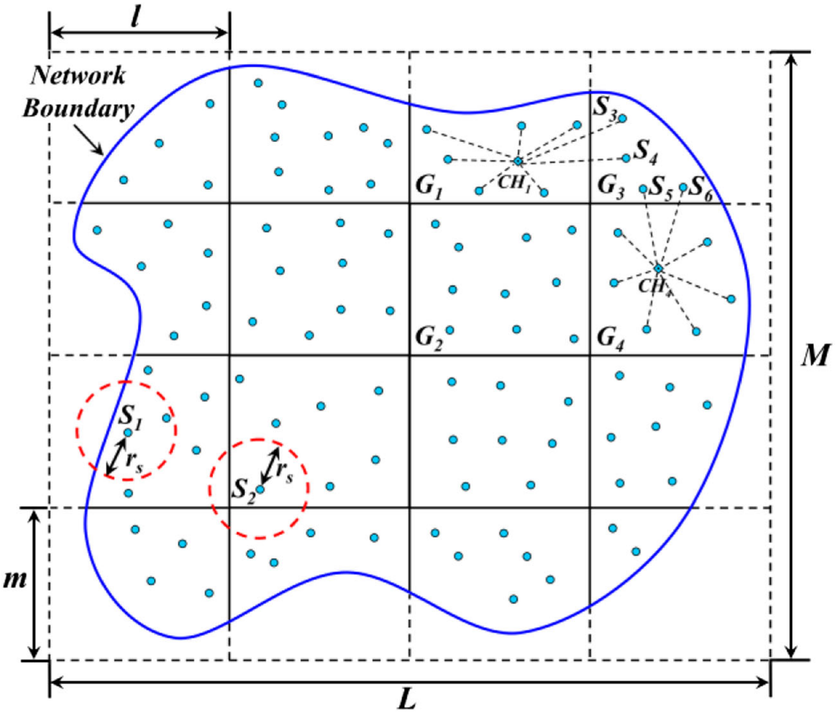


Fig. 1 Nodes Distribution and Clustering based on Grids

Thereinto,

$$\overline{d(S_i, G_u)} = \frac{\sum_{j=1}^{Num(G_u)-1} \sqrt{(x(u,i)-x(u,j))^2 + (y(u,i)-y(u,j))^2}}{Num(G_u)-1} \tag{4}$$

To ensure energy balance between nodes and to minimize the cost of communication, the node with the maximum value of *P(S$_i$)* is selected as the cluster head of G_u. In the following process of target tracking, if the residual energy of the cluster head is less than 15% of its initial energy, it will no longer serve as a cluster head. In this case, the grid will select a new cluster head according to the same method so that the network can run well.

4 Target localization

At present, RSSI (Received Signal Strength Indicator) is still one of the most widely used methods for distance measurement in sensor networks. However, a large number of researches have shown that the wireless signal transmission between nodes is easily influenced by the environment which results in reflection, refraction, diffraction et al [32]. To a certain degree, it reduces the accuracy of target tracking. Therefore, it is necessary to improve the tracking accuracy of nodes for the moving target. As mentioned before, in ETTA, each node knows its position. Thus, the average ranging error Δd can be defined as follows.

$$\Delta d = \frac{1}{N} \sum_{i=1}^{i=N} \left(\frac{\sum_{j=1}^{j=Num\_nei(S_i)} \left(d(S_i, S_j)' - d(S_i, S_j) \right)}{Num\_nei(S_i)} \right) \tag{5}$$

d(S$_i$,S$_j$)' is the distance measured by RSSI between S_i and S_j, and *d(S$_i$,S$_j$)* represents the real distance between them. It is not difficult to know that *d(S$_i$,S$_j$)* is less than *d(S$_i$,S$_j$)'*. Moreover, *Num_nei(S$_i$)* is defined as the number of neighbors of S_i.

Thus, when the node detects the target *O*, the estimated distance will be revised by formula (6).

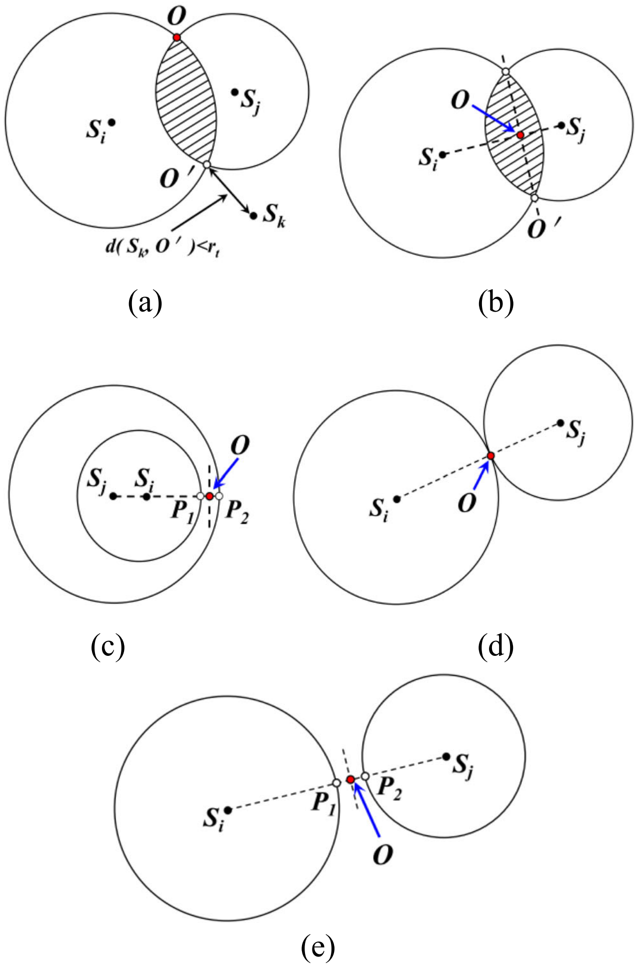
$$d(S_i, O)' = d(S_i, O) - \Delta d \tag{6}$$

d(S$_i$, O) is the measured distance between S_i and the target.

In the two-dimensional plane network, to get its position, the sensor needs to obtain at least three non-collinear anchor nodes' coordinates. However, in ETTA, due to the boundary effect as well as the distribution of nodes, the target may still be able to located even if it is only detected by two nodes.

As shown in Fig. 2, assuming that only S_i and S_j in the network can find the target, *O*. The radius of circle S_i is the corrected value of the measured distance between S_i and *O*. Similarly, the radius of circular S_j can be obtained. Next, considering the following conditions.

1) If circle S_i and S_j intersect, there are two possibilities for the estimated position of the target, as *O* and *O'* show in Fig. 2(a). If another active node S_k appears near both S_i and S_j and *d(S$_k$, O')* is smaller than r_t, it is certain that the target's real position is *O* because S_k did not detect the target. If there is no such node like S_k in the network, we can use the convex programming idea to take the intersection of the line S_iS_j and the straight-line *OO'* as the estimated position of the target, as shown in Fig. 2(b).
2) If there exists an inclusion relation between the circle S_i and the circle S_j, the straight line S_iS_j intersects the narrowest of the rings formed by these two circles. The midpoint of the two intersections will be set as the estimated position of the target, as shown in Fig. 2(c).



(a) (b)

(c) (d)

(e)

Fig. 2 Target's Position Estimation by Two Nodes only

3) If the circle S_i is tangent to the circle S_j, the tangent point will be taken as the estimated position of the target, as shown in Fig. 2(d).
4) When the circle S_i and the circle S_j are separated, which means that the straight line S_iS_j will intersect the two circles at two points, the midpoint of the two intersection points will be chosen as the estimated position of the target, as shown as demonstrated in Fig. 2(e).

When the target is detected by three or more nodes, the traditional localization method can be used to achieve the positioning of the target. In this case, APIT (Approximate Point-in Triangulation Test) [4, 5] is a good choice because it is not only range-free but also an iterative refinement based method. That is to say, the node which is to be positioned can choose three beacons (black points shown in Fig. 3) to test whether it is in the triangle formed by them. After testing all the potential combinations, APIT calculates out the overlapped triangle region that contains the unknown node. Thus, the centroid of the overlapped region is regarded as the estimated position of this node.

For example, in Fig. 3, there are 12 beacon nodes (S_1, S_2,…S_{12}). It is not difficult to know that any three non-collinear nodes can form a triangle. Then, the unknown node uses the "Point-in Triangulation Test" method [33] to judge which triangles it is possible to located in. Supposing that there are four such triangles (e.g., $\Delta S_1S_6S_8$, $\Delta S_2S_9S_{10}$, $\Delta S_3S_7S_{12}$, $\Delta S_4S_5S_{11}$ in Fig. 3), it is regarded that the common area of the four triangles (e.g., the yellow polygon in this figure) is the area where the unknown node is most likely to exist. Without loss of generality, the centroid of this polygon is defined as the estimated position of the node to be located, just like the red point shows in Fig. 3.

However, when the unknown node is not inside the triangle or when the beacons around the unknown node cannot form a triangle, the localization accuracy of APIT will be greatly decreased. The above situation may occur in ETTA because of the random distribution of the nodes (especially when the target is firstly detected by the nodes near the boundary of the network). So, the localization algorithm in ETTA should be improved based on the traditional APIT algorithm. As shown in Fig. 4, if the relationship among S_i, S_j, S_k and O could meet formula(7), it is possible to basically determine that the target is in the triangle formed by these three nodes, as shown in Fig. 4(a). S_i, S_j and S_k are the nodes who can detect the target.

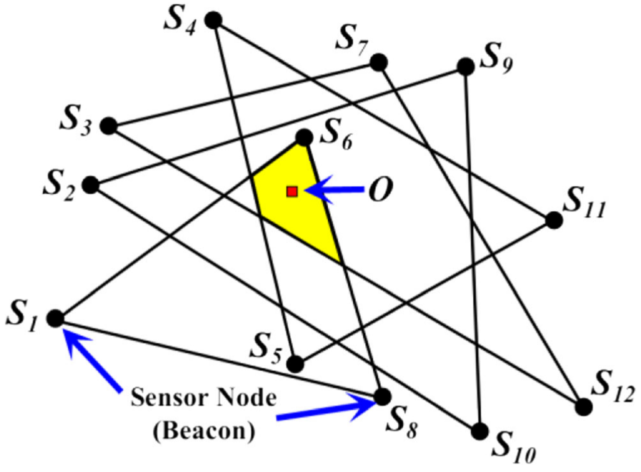


Fig. 3 Target Position Estimation by APIT [33]

$$S(\Delta S_iS_jS_k)\approx S(\Delta OS_jS_k)+S(\Delta S_iOS_k)+S(\Delta S_iS_jO) \quad (7)$$

If the value of $S(\Delta OS_jS_k)+S(\Delta S_iOS_k)+S(\Delta S_iS_jO)$ is far greater than that of $S(\Delta S_iS_jS_k)$, it is regarded that the target is out of the scope of $\Delta S_iS_jS_k$, as shown in Fig. 4(b).

Thus, if the target is detected by three or more nodes, the solution is to determine whether there are triangles formed by these nodes can meet the condition of formula (7). If so, the traditional APIT method is used to estimate the position of O, otherwise, these nodes are combined with each other to estimate the position of O according to the method shown in Fig. 3. The mean value of these estimated positions is regarded as the final position of O.

5 Target tracking based on moving direction prediction

5.1 Target discovery

In order to describe the target tracking process more clearly, hereby we make the following definitions.

5.1.1 Boundary node

For one node S_i in current network (grid), if its sensing range is beyond the boundary of the network (grid), S_i is regarded as the boundary node. As node S_1 shows in Fig. 1, it is not only the boundary node of the network but also the boundary node of the grid. In addition, S_2 is only the boundary node of the grid.

5.1.2 Working mode of node

Active mode In this mode, the sensor node can continuously monitor the surrounding environment, and it also has the ability of both data forwarding and computing.

Monitoring mode In this mode, to save energy, the sensor node can only receive data.

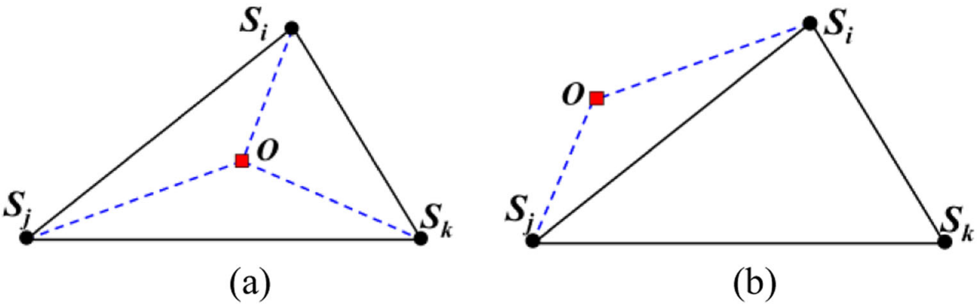


Fig. 4 Find out the Region where the Target is Located at

5.1.3 State of grid

Tracking state If the target is now in the current grid, this grid is in the tracking state. Meanwhile, all nodes in this grid should be in active mode.

Prepared-tracking state When the possibility of the target's departure from the current grid is beyond a certain threshold, each neighborhood grid calculates the possibility of target's coming into itself at the next moment. Then, the grid which has the maximum possibility turns into the prepared-tracking state, and nodes in this grid keep in active mode. The purpose of setting this state is to awaken those nodes which are near the position where the target is most likely to pass. So, the accuracy and the real-time performance about target tracking will be improved.

Preparing-tracking state The grid which is the common neighborhood of the grids in tracking state and prepared-tracking state is set as the preparing-tracking state. In this type of grid, the cluster head and boundary nodes are all in active mode and other nodes are set to the monitoring mode. Thus, even if the target moves into a grid which is not in the prepared-tracking state, it can still be detected. This greatly enhances the fault tolerance of the system.

At the beginning, all the boundary nodes of the network are in active mode. Due to this, the target can be detected in time as soon as it enters into the network. Meanwhile, other nodes work in the monitoring mode, as shown in Fig. 5.

If there are more than one network boundary nodes have detected the target, the position of this target is calculated out according to the method mentioned in section 4. Subsequently, each network boundary node who finds out the target sends the message, "target is close to the network", to its cluster head. The cluster head that receives this message changes its mode from the monitoring to active, and then it broadcasts this message to its members. The cluster members then turn into active mode, as shown in Fig. 6.

Once the target moves into the network, it must be located in one of the grids. It can be known from the above description that at this moment, nodes in this grid are already in active mode and the grid is also in the tracking state. Meanwhile, to

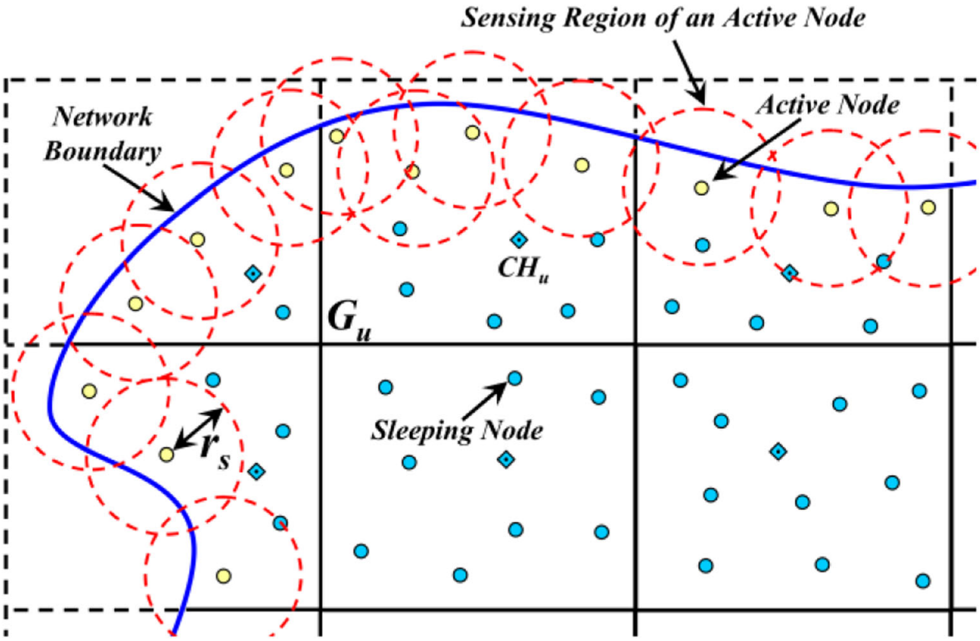


Fig. 5 The Working States of Nodes before Target's Coming

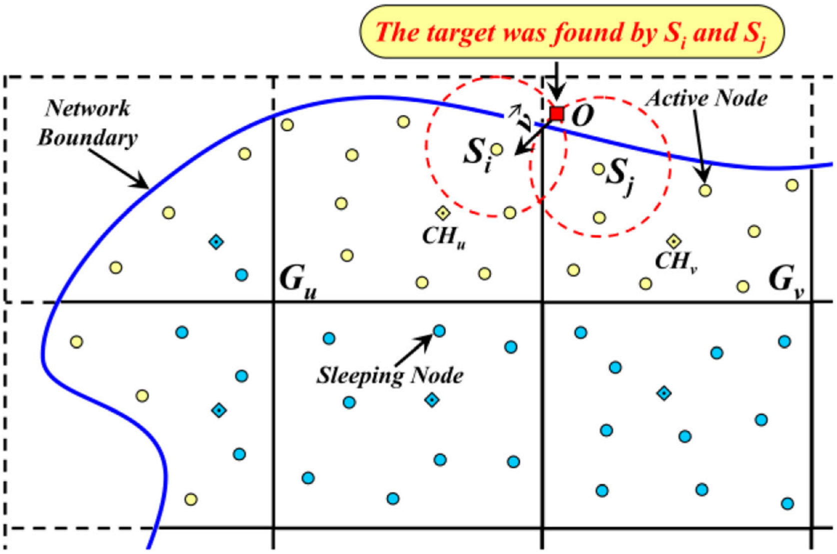


Fig. 6 Working States of Nodes when the Target is Found out by the Boundary Nodes

reduce energy consumption, the cluster head in this grid broadcasts a message, "target has been found out", to other cluster heads. The cluster head who receives the message then makes all its member nodes as well as itself into monitoring mode, as shown in Fig. 7.

5.2 Target's position prediction

At moment t, for a grid G_u which is in tracking state, it can predict the region that the target may arrive to at moment $t+\Delta t$.

5.2.1 Case 1. G_u is a boundary grid

As shown in Fig. 8, when the area of the circle O which is outside the boundary of the network (the green part in this figure) exceeds a certain threshold, it indicates that the target probably moves out of the network at the next moment. O is the target and the radius of circle O is $v \times \Delta t$. Then, all nodes in G_u who can detect this target predict the position of O at the next moment by Kalman filter algorithm. So, the general non-linear equation of state is

$$X_{t+\Delta t|t} = IX_{t|t} + U_{t+\Delta t} \tag{8}$$

And, the observation equation is

$$\mathrm{Z}_{t+\Delta t} = HX_{t+\Delta t|t} + V_{t+\Delta t} \tag{9}$$

$X_{t+\Delta t|t}$ is the predicted value of the system at $t+\Delta t$. In other words, the real coordinates of the target at moment t is used to predict the possible position of the target at $t+\Delta t$. $X_{t|t}$ is the optimal state vector at moment t, that is, the real position of the target at this time. I is the system parameter. In formula (9), $Z_{t+\Delta t}$ is the measurement value at $t+v\Delta t$, and H is the parameter of the measurement system. Moreover, $U_{t+\Delta t}$ and $V_{t+\Delta t}$ represent the process and measurement noise

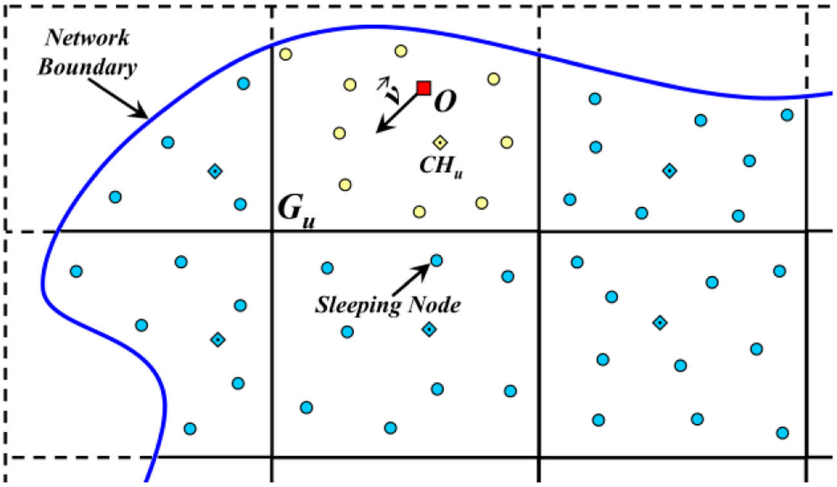


Fig. 7 Working States of Nodes after the Target Entering into the Network

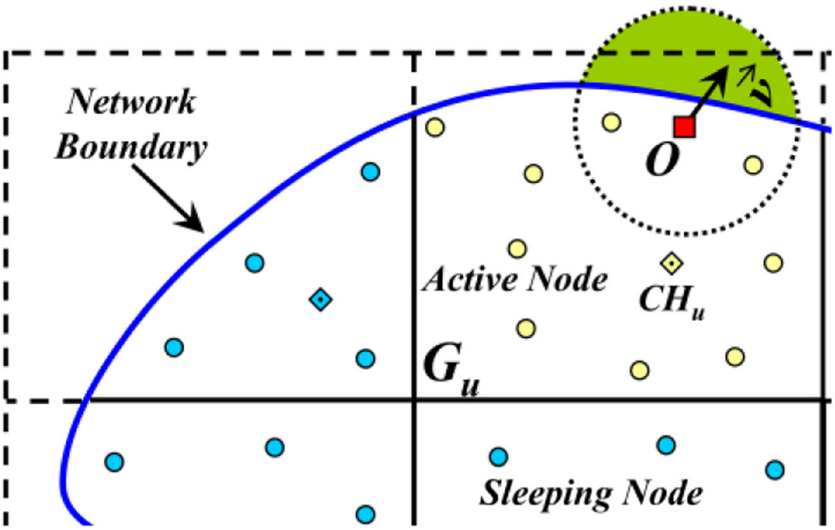


Fig. 8 Target's Position Prediction with the help of Nodes in Boundary Grids

respectively. It assumes that they are both Gaussian white noise, and the covariance is Q and R. They do not change the state of the system.

Furthermore, it is assumed that $X_t = \left[x_t, v_{x_t}, y_t, v_{y_t}\right]^T$ and $Z_t = \left[x_{z_t}, y_{z_t}\right]^T$. X_t and Z_t are defined as the horizontal and vertical coordinates of the target at moment t. While, v_{x_t} and v_{y_t} are the velocity component of the target in the X and Y directions. In addition, x_{z_t} and y_{z_t} represent the horizontal and vertical coordinates of the target detected by the system. When predicting the optimal state of the system, it is necessary to update the covariance W at each moment. That is,

$$W_{t+\Delta t|t} = IW_{t|t}I^T + Q \tag{10}$$

In formula (10), $W_{t+\Delta t|t}$ is the covariance for $X_{t+\Delta t|t}$, while $W_{t|t}$ is the covariance for $X_{t|t}$. I^T is the transpose matrix of the system parameter, I. In addition, Q is the covariance of the process noise. Combined with the predictive value at $t+\Delta t$ as well as the measurement value $Z_{t+\Delta t}$, it is not difficult to calculate out the optimal estimate value at $t+v\Delta t$. This is the position where the target will most likely to arrive to at moment $t+v\Delta t$.

$$X_{t+\Delta t|t+\Delta t} = X_{t+\Delta t|t} + K_{t+\Delta t}\left(Z_{t+\Delta t} - HX_{t+\Delta t|t}\right) \tag{11}$$

$K_{t+\Delta t}$ is the Kalman Gain. That is,

$$K_{t+\Delta t} = W_{t+\Delta t|t}H^T / \left(HW_{t+\Delta t|t}H^T + R\right) \tag{12}$$

In order to keep the Kalman filter running until the network lifetime is over, the covariance $W_{t+\Delta t|t+\Delta}$, that is corresponding to the system optimal state $X_{t+\Delta t|t+\Delta t}$ at moment $t+v\Delta t$, needs to be updated. That is,

$$W_{t+\Delta t|t+\Delta t} = (I - K_{t+\Delta t}H)W_{t+\Delta t|t} \tag{13}$$

Values of the system parameter I and the measurement parameter H in the above formula are

$$I = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

If the predicted position of O is indeed outside the network (as the pink square shows in Fig. 9), the cluster head in G_u will broadcast a message to the cluster heads in all the boundary grids. The content of this message is "the target is likely to leave the network". The cluster head that receives the message will change the network boundary node within its cluster to the active mode. So, when the target re-enters into the network, it is possible to find it in real-time.

If the area of circle O which is outside the boundary of the network does not exceed the threshold, the prediction about the region in which the target may arrive to will be carried out according to the method described in case 2.

5.2.2 Case 2. G$_u$ is a non-boundary grid

When the area of the circle outside the grid (defined as $S_{out}(O)$) exceeds a certain threshold (the green area in Fig. 10), it is possible that the target will move into the neighborhood grid at moment $t+\Delta t$. $W_{out}(G_u)$ is defined as the possibility about the target leaving from the current grid G_u.

$$W_{out}(G_u) = S_{out}(O)/\pi(v\times\Delta t)^2 + \lambda \times Max(\cos\theta_k) \tag{14}$$

θ_k is the angle between the velocity vector of the target and the vertical axis of the target to the boundary of the grid (e.g., θ_1 and θ_2 in Fig. 10). λ is an adjustable parameter. This formula considers both the arrival possibility of the target at the

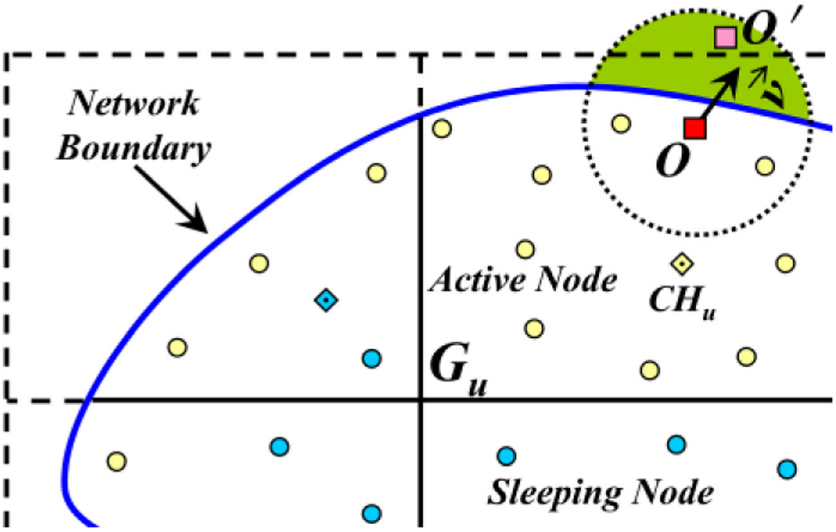


Fig. 9 The Predicted Position of the Target is outside the Network

next moment and the direction of the current velocity. To a certain extent, it improves the accuracy of judging the moving trend of the target.

If the value of W_{out} (G_u) is larger than the threshold W_{out} ‘, the target is likely to leave from the current grid. In order to ensure the real-time and the accuracy of target tracking, the nodes located at the vicinity of the area where the target may move to at moment $t+\Delta t$ need to be in active mode. Moreover, these nodes should deduce that which grid the target is most likely to move to.

In this case, each grid, e.g., G_v, covered by circle O calculate the value of $W_{in}(G_v)$ according to formula (15). $W_{in}(G_v)$ is regarded as the possibility of target's moving into the region of G_v at the next moment. It is easy to know that, the number of this type of grids is at most three, as G_i, G_j and G_k show in Fig. 10.

$$W_{in}(G_v) = S(O, G_v)/\pi(v \times \Delta t)^2 + \lambda^{'} \cos\mu_k \tag{15}$$

In formula (15), $S(O, G_v)$ is the size of the area where circle O is overlapped with G_v. μ_k is the angle between the velocity vector of the target and the connection vector from the target to the center of G_v (such as μ_1, μ_2 and μ_3 in Fig. 10). λ' is an adjustable parameter.

The grid with the maximum value of $W_{in}(G_v)$ then goes into the prepared-tracking state (G_i in Fig. 11). Otherwise, it will be in the preparing-tracking state (G_j and G_k in Fig. 11).

5.3 Grid's state updating

At the moment of $t+\Delta t$, the target tracking strategy is described as follows.

1) If the target moves into G_i, the state of G_i is changed from prepared-tracking to tracking. At the same time, to reduce the unnecessary energy consumption, the cluster head in G_i broadcasts a message to G_u and other grids that in

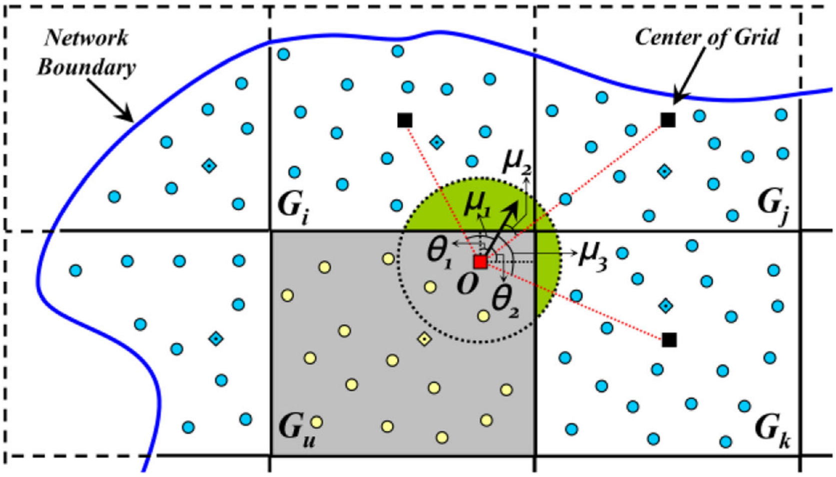


Fig. 10 Target's Position Prediction with the help of Nodes in Non-boundary Grids

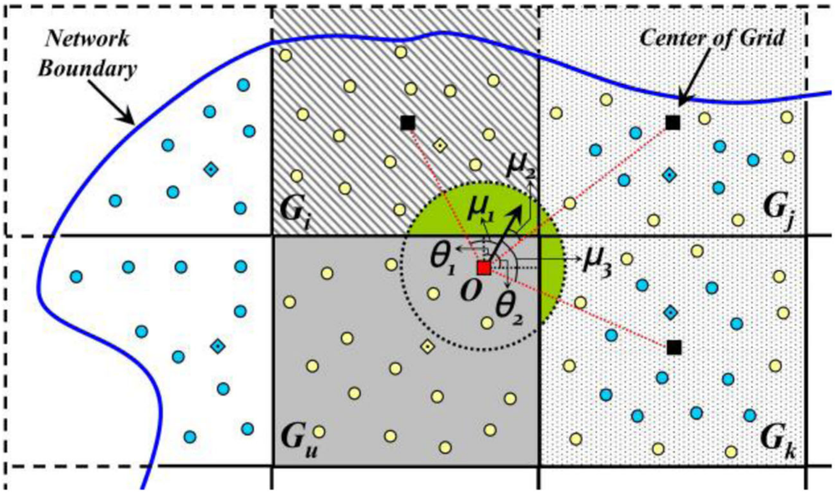


Fig. 11 States Updating of the Non-boundary Grids after Target's Position Prediction

preparing-tracking state (e.g., G_j and G_k in Fig. 11). Nodes in these grids will then change its working mode into monitoring, as shown in Fig. 12.

2) If the target moves into G_k (G_k is now still in the preparing-tracking state), the cluster head of G_k shall notify all nodes in its area to change their working mode into active. That is to say, the state of G_k is changed from preparing-tracking to tracking, as shown in Fig. 13. Similarly, the cluster head in G_k broadcasts a message to G_u and other grids that in preparing-tracking or prepared-tracking state (e.g., G_i and G_j in Fig. 11). Nodes in these grids will then change its working mode into monitoring, as shown in Fig. 13.

3) Under the conditions of $W_{out}(G_u) > W_{out}'$, if the target is still in G_u at the moment of $t + \Delta t$ (although this is unlikely to happen), the cluster head in G_u broadcasts a message to its neighbor grids that is in the preparing-tracking or prepared-tracking state. Therefore, nodes in these grids will then change its working mode into monitoring, as shown in Fig. 14.

It is worth mentioning that, in case 1) and case 2), if G_u and G_v are respectively the boundary and non-boundary grids, and if all the boundary nodes are in active mode at $t + \Delta t$, the cluster head of G_v needs to broadcast another message to the cluster heads in all the boundary grids. The content of this message is "the target is still in the network". All the cluster heads that receive this message then revert the nodes in their own cluster to monitoring mode.

5.4 Tracking process recovery

In ETTA, if there are the following circumstances, it may briefly appear the condition that the target is missing.

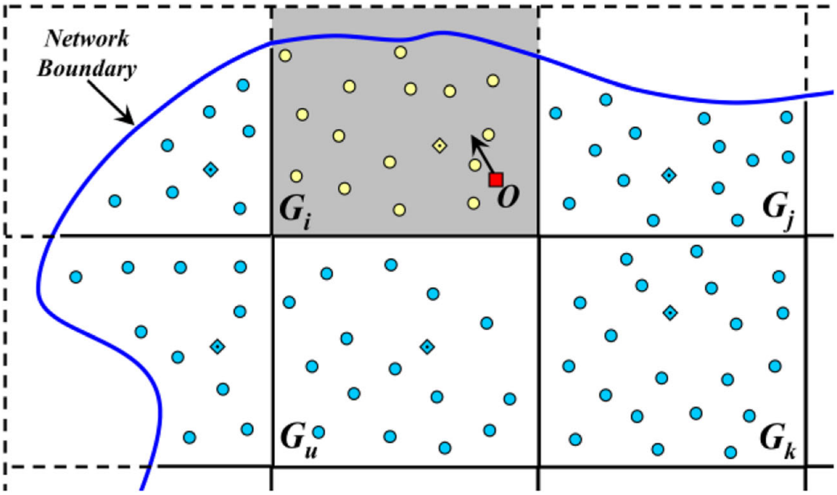


Fig. 12 The Target Enters into the Grid that is in Prepared-tracking State

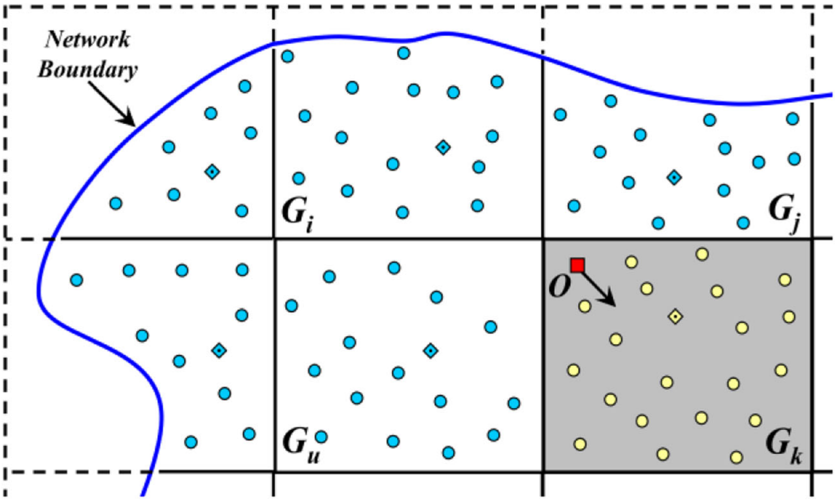


Fig. 13 The Target Enters into the Grid that is in Prepared-tracking State

1) Node failure or energy depletion leads to the failure to discover the target in time.
2) There is an error in predicting the position of the target, which means no node is active in the grid where the target moves to so that the system cannot detect the target.
3) The communication links between nodes are disturbed, so that the corresponding nodes in the neighborhood grids cannot be awakened to active mode, then losing the ability to track the target in real time.

The tracking recovery strategy in ETTA is described as follows.

Step 1: When the target is lost, all nodes which have completed target tracking at moment t send the message "$Error(ID, t, v_t, a_t)$" to their cluster head. In this message, ID is the serial number of the node itself, and v_t and a_t are the coordinates of the velocity as well as position of the target at moment t, respectively.

Step 2: When the cluster head receives this "*Error*" message, it immediately delivers the message to the cluster head in all the neighborhood grids.

Step 3: When the cluster head in the neighborhood grid receives this "*Error*" message, it wakes up all its member nodes to carry out the search for the target. If the system still cannot find the target, the "*Error*" message will continue to be forwarded to the other cluster heads, so as to wake up more nodes to search for the target.

Step 4: Once a node finds out the target, it immediately sends a response packet "$Find(ID, t', v_{t'}, a_{t'})$" to its cluster head. $v_{t'}$ and $a_{t'}$ are the coordinates of the velocity and position of the target at moment t'. If the target cannot be positioned, $a_{t'}$ can be empty. In the meantime, the grid which the cluster head lies in changes into tracking mode.

Step 5: The cluster head in the grid which is in tracking state transmits the "*Find*" message to the cluster head in all its neighborhood grids to indicate that it has rediscovered the target. The cluster heads that received

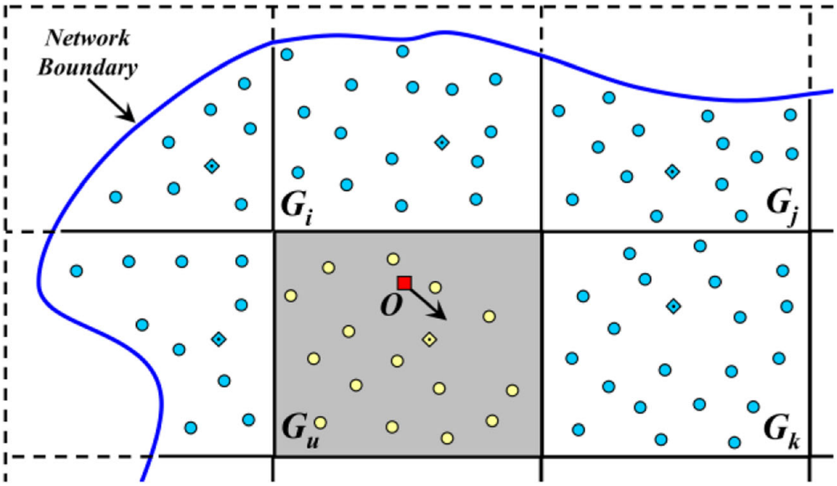


Fig. 14 The Target is still in G_u at the Moment of $t + \Delta t$

the "*Find*" message then continue to send this message to other grids that are still searching for the target. Simultaneously, the cluster head restores itself and its member nodes to monitor mode.

6 Simulation results and analysis

In order to verify the efficiency of ETTA in terms of target tracking accuracy, target missing rate, tracking delay and network energy consumption, experiments were carried out under the Origin platform. Meanwhile, it is compared with two typical target tracking algorithms (BPS [34] and EMTT [35]) in sensor networks.

BPS (target tracking with Binary Proximity Sensors) [34] is a type of tracking method which employed binary proximity sensors to track moving target in WSN. Such sensors provided only 1-bit information regarding a target's presence or absence in their vicinity, albeit with less than 100% reliability [34]. This method utilized the sensor outputs to estimate individual positions in the path of the target in the near past and found the line which best fitted the path points. This line was then used to estimate the target's current position. Different target tracking system can be established by different weight calculation methods in this tracking algorithm framework, whose performance depended on the algorithms used in weight calculation.

In addition, EMTT (Energy efficient Moving Target Tracking) [35] is another typical target tracking algorithm in sensor networks. It used a linear dynamic system with multiple sensors to track the target and monitor its surrounding area, which established the fuzzy model for measurement condition estimation. Firstly, the measurement possibility was calculated based on the probability-possibility transformation. Then, the measurements with high possibility and low possibility were considered as the L-sensor linear dynamic system measurements for position calculation via neighborhood function. Finally, GKF (Generalized Kalman Filter) was utilized to produce the optimization of smoothed position estimates.

We randomly deploy *600–1000* nodes in *100 m* × *100 m* and *200 m* × *200 m* networks, and it supposes that the locations of all nodes are known. The initial energy of the nodes is *40* unit, the sensing radius is between *5 and 10 m.*

6.1 Target tracking accuracy

6.1.1 Tracking accuracy under different trajectories

In the *100 m* × *100 m* network, the target is in accordance with the two different trajectories for the uniform movement respectively, and the speeds are set to *4 m/s* and *8 m/s*. The number of nodes in the network is set to *800*, and the length of its sensing radius is *8 m*. Figures 15 and 16 are the predicted trajectory and the real trajectory comparison results based on ETTA, BPS and EMTT. In order to match the actual scene of the moving target as far as possible, two types of moving trajectories have been used in this experiment. One is the relatively smooth trajectory (Fig. 15) and the other is the trajectory with a greater degree of curvature and closer to the edge of the network (Fig. 16). It can be seen that in the case of different trajectories of the target, the predicted results of the three algorithms are all close to the real one. However, the degree of coincidence between the ETTA and the real trajectory is higher than that of both BPS and EMTT, especially when the direction of the velocity of the target is changed greatly. This is because ETTA uses the virtual grid division and self-clustering way to achieve distributed tracking. What's more, by setting the "prepared-tracking state" and "preparing-

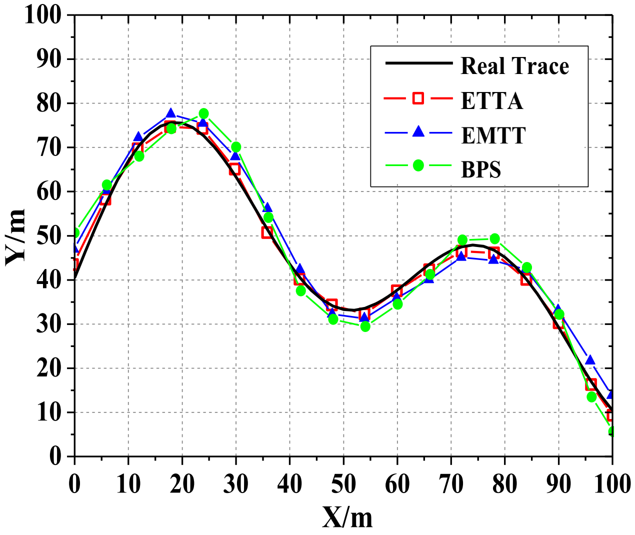


Fig. 15 Target Tracking Accuracy (*v* = *4 m/s*)

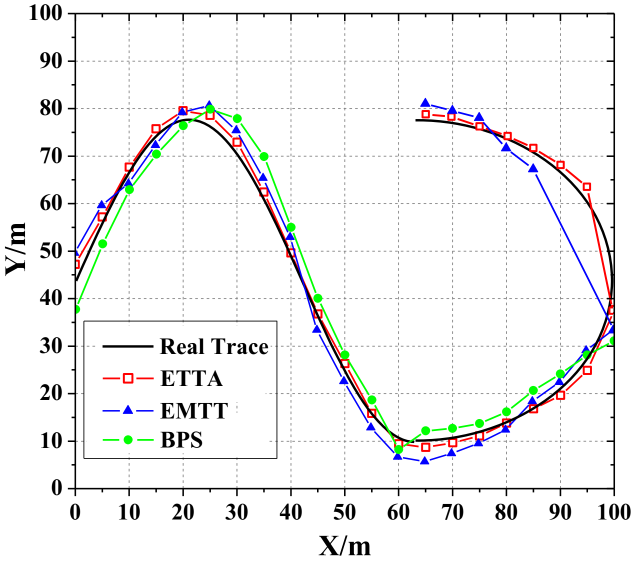


Fig. 16 Target Tracking Accuracy (*v* = *8 m/s*)

tracking state" of the grid, this algorithm further ensures the continuity of target tracking. According to section 5.3, it is not difficult to know that regardless of which neighbor grid the target will move to at the next moment, ETTA can continue to track the target and update the grid's status, that further improves the tracking accuracy. In contrast, the BPS and EMTT methods mainly use the historical trajectory of the target to perform the curve fitting, and thus predict the next possible reach. Therefore, when the direction of the velocity of the target changes greatly, the reference value of the historical track point will be greatly reduced, resulting in a larger prediction error.

Moreover, through the comparison of Figs. 15 and 16, we can find that the tracking accuracy of the three algorithms decrease when the target velocity increases. However, the accuracy decline range of ETTA is relatively low. This is also because the "tracking state", "prepared-tracking state" as well as the "preparing- tracking state" have ensured that wherever the target goes, it can always be found by active nodes. In addition, ETTA also proposed the improved APIT localization algorithm, which enhances the positioning accuracy of the target to a certain extent. Even if the target moves faster, the target tracking effect will not show a significant decline.

In addition, it also can be seen from Fig. 16 that when the target moves to the vicinity of the network boundary, the accuracy of the three methods for trajectory prediction has decreased. For example, in Fig. 16, when the target moves from one grid (the four vertexes of this grid is (90, 30), (100, 30), (90, 40), (100, 40)) to the upper grid along the right side of the network, BPS has been unable to predict the position of this target at the next moment, that is, it mistakenly considered "target lost". Although the target tracking effect of EMTT is slightly better than that of BPS, it also once appeared on the "target lost" misjudgment. In contrast, for the target's trajectory near the boundary, ETTA has higher tracking accuracy than the other two methods. It is not difficult to know from section 5.2 that, in ETTA, for the boundary grid, the Kalman filter method is used to predict whether the target is moving out of the boundary or not. Furthermore, ETTA will wake all the boundary nodes into active mode when the predicted result is "out of boundary". This ensures the real-time, accurate tracking of the target after it enters the network again. However, it should be pointed out that in real physical scenarios, the effect of these three target tracking methods is also affected by terrain, interference sources and obstacles. We will study it in depth in the future work.

6.1.2 Target tracking errors

For the convenience of explanation, the target tracking error is defined as the Euclidean distance between the target's real coordinate and the predicted coordinate at a certain monitoring time. Experiments are carried out in the *100 m* × *100 m* and *200 m* × *200 m* network respectively and the internal grid sizes are set to *10 m* × *10 m* and *13 m* × *13 m*. What's more, the node's sensing radius is set to *8 m*, and the target movement speed is *4 m/s*.

As can be seen from Fig. 17, regardless of the size of the network, the target tracking error of ETTA decreases as the number of nodes in the network increases. This is because in the improved APIT localization algorithm, the more the number of beacon nodes, the more the ideal triangle produced. Thus, the localization accuracy is improved. Furthermore, the size of the grid also has a certain impact on target tracking error. That is, the larger the size of the grid, the smaller the error. As mentioned above, in ETTA, when the target enters into a grid, the cluster head will wake up all the nodes in this grid. In the case where the nodes are randomly and uniformly distributed, more and more nodes are awakened to participate

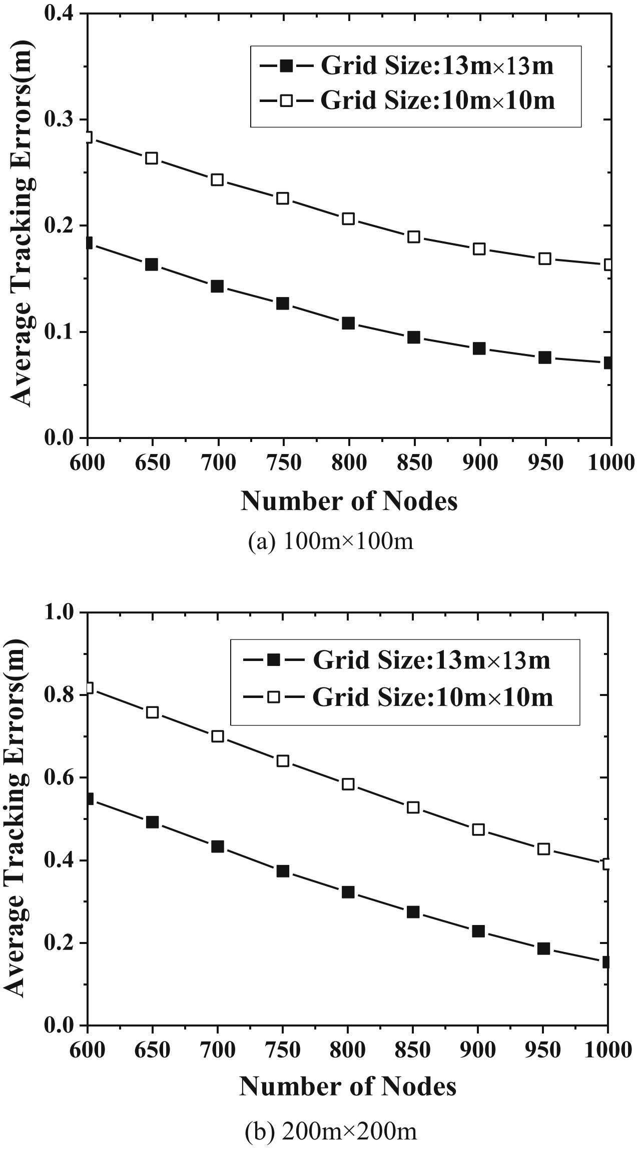


Fig. 17 Average Tracking Error under Different Grid Sizes

in target localization and tracking with the increase of the size of the grid.

Besides, Fig. 17 also shows that when the number of nodes and the size of grid are constant, the larger the network, the higher the target tracking error. This is due to the decrease of the number of active nodes that participate in target tracking per unit area. However, in the larger network, the tracking error caused by the increase of the grid's size is more obvious. For example, in the case of the same number of nodes, when the length of the grid increases from *10 m* to *13 m*, the mean value of the error in Fig. 17(b) (about *0.25 m*) is significantly larger than that in Fig. 17(a) (about *0.1 m*).

Figure 18 shows the maximum target tracking error under different number of nodes. Moving speed of the target is *4 m/s*. It is easy to see from the figure that the maximum error in ETTA is less than that in the other two methods, especially when the number of nodes is small. This is because ETTA uses grids to achieve distributed target tracking which ensures that a significant number of nodes are active in the vicinity of the target. Moreover, with the help of the improved APIT method, the accuracy of target localization is also improved. In EMTT, the target tracking node needs a certain number of neighbor nodes to assist in locating the target, and the topology dependency between nodes is very strong. While, the BPS method can only determine whether or not the target is within the sensing range of the node and cannot measure the distance between the target and the node. Thus, the localization errors of these two methods are greater than that of ETTA.

In order to further refine the target tracking effect at different times, we give the experimental results of Fig. 19 (moving speed of the target is *4 m/s* and the number of nodes in the network is *800*). When the target is discovered for the first time, the tracking error of ETTA is significantly lower than the other two methods. This is because the network boundary nodes are always active and they can accurately locate the target. When the target moves into the network, the corresponding grids immediately go into tracking state, which ensures the real-time performance of the target tracking (as shown in Figs. 5 and 6). In addition, it can be found that the target tracking error in the three algorithms is increased by a certain amplitude at the three monitoring time points from the 21st to 23rd second. This is because the direction of the movement of the target changes greatly during this period of time. However, due to the prediction mechanism of ETTA, nodes in the "prepared-tracking state" and "preparing-tracking state" grids are awakened in advance, so that the tracking error can be reduced. Thus, the increment of error of ETTA from the 21st to 23rd second is less than that of BPS and EMTT.

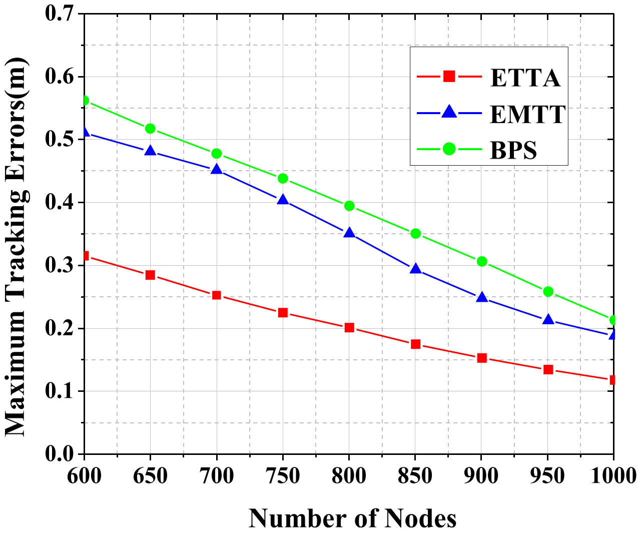


Fig. 18 Maximum Target Tracking Error (*100 m* × *100 m*)

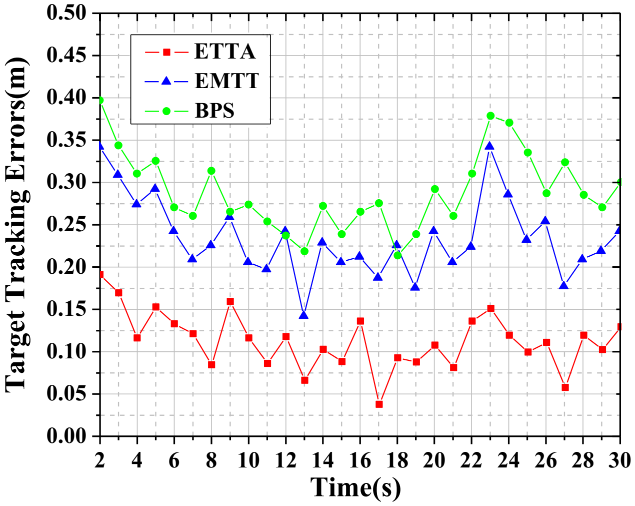


Fig. 19 Target Tracking Error at Different Time Points (*100 m* × *100 m*)

Of course, it is worth noting that both the experiments in Figs. 18 and 19 are only simulated in a 100 m × 100 m network. In real physical scenarios, the size of the network may be much larger than that. In this case, the scattering, refraction and diffraction phenomena of radio waves may be more obvious, which will cause greater interference to the communication between nodes. So, the target tracking error is likely to increase considerably. This problem may be solved by increasing the transmission power or the deployment density of nodes.

6.2 Target missing rate

In order to analyze the fault tolerance of these target tracking algorithms, the target missing rate is further studied. The network size and the target moving speed are set to *100 m* × *100 m* and *4 m/s* respectively. Here, the target missing rate is defined as the ratio of the duration during which the target is not monitored by any node to the total time period in which the target stays in the network.

As can be seen from Fig. 20, when the monitoring time interval Δt increases, the target missing rate in ETTA also

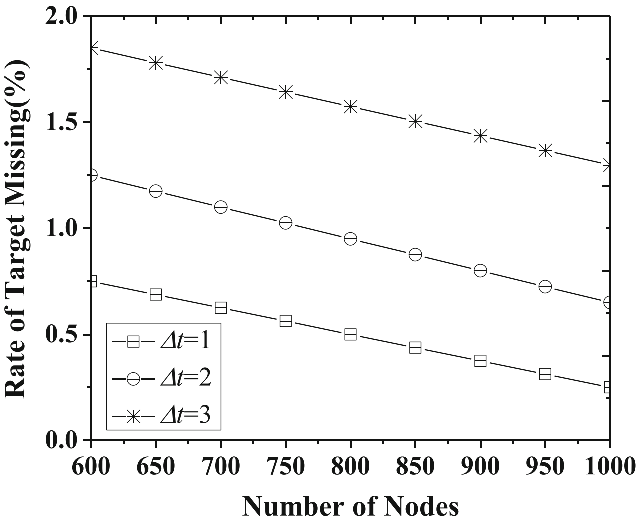


Fig. 20 Target Missing Rate of ETTA under Different Monitoring Intervals

increases. This is because at this time, the grid in "tracking state" may not be able to predict the "target is about to leave the grid", which failed to make the corresponding grids change their states to "prepared-tracking state" or "preparing-tracking state". So, the temporary interruption of target tracking occurs. However, it can be seen from section 5.4 that ETTA has established a recovery mechanism, which ensures that target tracking can be restored in a short time. So, the target missing rate is still acceptable. On the other hand, it also can be seen from this figure that the more the nodes in the network, the lower the target missing rate. The reason of this is similar to that of Figs. 17 and 18.

What's more, it can be seen from Fig. 21 that in the case where the number of nodes in the network is constant, the larger the sensing radius of the node, the lower the missing rate of the target. In the case of a certain radius of sensing, the greater the number of nodes, the lower the target missing rate. That means the greater the value of node density and sensing radius, the better the effect of target tracking. However, whether it is to increase the number of nodes or increase the length of its sensing radius, it will undoubtedly consume more energy. Therefore, it is often needed to consider the balance between the two in the real-world.

In the real-world, in order to save energy, the monitoring interval of active nodes tends to be larger than the value set in Fig. 20. Similarly, the sensing range of nodes is also limited in most cases. In this case, the target missing rate is bound to higher than the experimental results in Figs. 20 and 21. To this end, we can deploy more nodes in the network (of course, the premise is not to cause excessive coverage redundancy). In addition, we can appropriately reduce the size of the grid, so that more grids can participate in target tracking.

Figure 22 shows the experimental results of the target missing rate under the different moving speed of the target. The target's velocity increases from 0 to 16 m/s, which can basically match most of the current real target tracking scenarios. The network size is still *100 m* × *100 m*, and the number of nodes is *1000*. The sensing radius and monitoring interval are set to *10 m* and *1 s*, respectively. It can be seen from Fig. 22 that as the target's moving speed continues to increase, target missing rate of these three algorithms is increasing as well. In BPS, when the moving speed of the target increases from *8 m/s* to *10 m/s*, the missing rate is increased by 1%. While in EMTT, when moving speed of the target grows from *10 m/s* to *12 m/s*, the range of the target missing rate rising by more than 1%. But in ETTA, only the target moving speed increased from *14 m/s* to *16 m/s*, the target missing rate increases by more than 1%. So, ETTA is more adaptable to the velocity change.

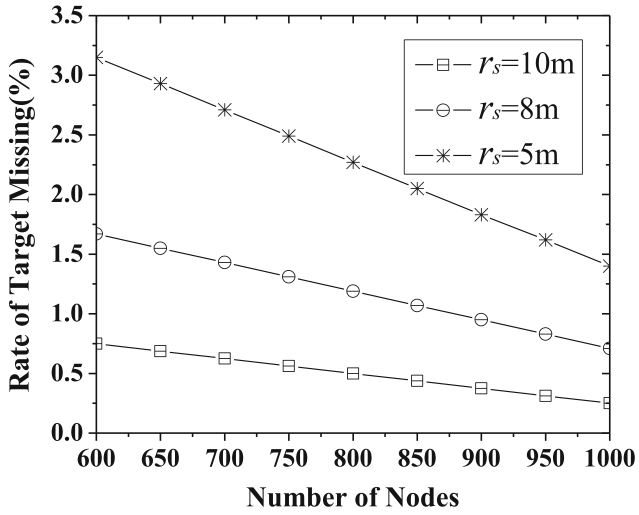


Fig. 21 Target Missing Rate of ETTA under Different Length of Sensing Radius

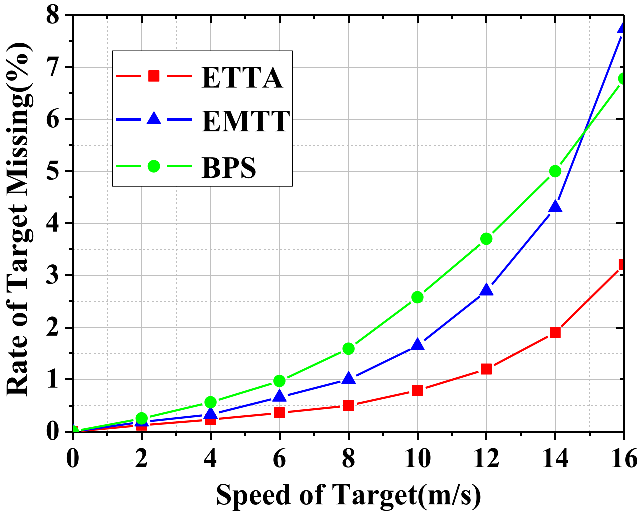


Fig. 22 Target Missing Rate under Different Target Moving Speeds

6.3 Target tracking delay

To verify the real-time performance of the three algorithms, comparisons about target tracking delay are shown in Figs. 23 and 24. The “target tracking delay” is defined as the interval from the time when the target is captured by at least one node to the first time it is successfully localized.

It can be seen that the target tracking delay in ETTA is between 0.5 s and 1.6 s when the sensing radius is 5 m–10 m, which shows good real-time performance. Moreover, no matter what the number of nodes is, the tracking delay in ETTA is always less than that in BPS and EMTT. This is because when ETTA starts to run, the network boundary nodes are active which ensures that the target can be quickly captured after appearing, especially in the case where the sensing radius is longer or the number of nodes is larger. For example, when r_s = 10 m or the number of nodes is 1000, the tracking delay in ETTA is only 0.5 s. While in BPS, only a few nodes could locate the target, and the network boundary nodes are not always active. So, the real-time performance of target tracking in BPS is not very good, especially when the density of nodes is lower. In addition, EMTT needs to establish the neighborhood function and the target motion model to locate the target, so that both communication and computing space-time overhead are larger. Therefore, both BPS and EMTT are inferior to ETTA in tracking real-time, but the tracking delay of the two is also significantly reduced when the sensing radius of node or the number of nodes increases.

6.4 Energy consumption on target tracking

Figure 25 shows the total energy consumption of nodes after performing the target tracking process under a different number of nodes and different monitoring time intervals. It is easy to see that the longer the monitoring interval, the lower the total energy consumption. Obviously, this is caused by the reduction of the number of times on localization and the state switching. In the case of a certain monitoring time interval, the total energy consumption of nodes is proportional to the number of them, but the growth rate is slower. This is because ETTA uses a grid-based approach to achieve distributed target tracking. The increase of the number of nodes will only increase the number of the active nodes, but other nodes are still in the listening mode. Therefore, the energy efficiency of ETTA is higher.

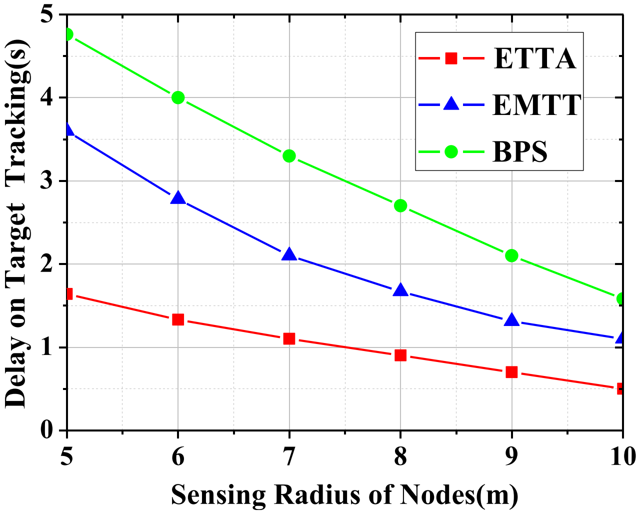


Fig. 23 Target Tracking Delay under Different Sensing Radius

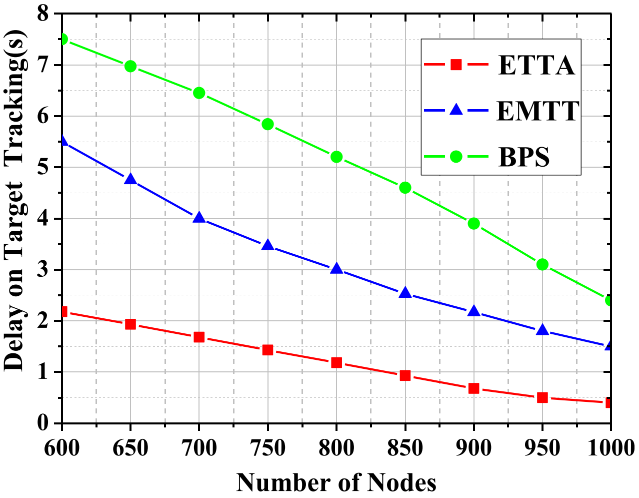


Fig. 24 Target Tracking Delay under Different Number of Nodes

Figure 26 shows the average residual energy of nodes after completing a target tracking process. The network size is set to *200 m* × *200 m* and the target moving speed is *4 m/s*. With the increase of the number of nodes, the average residual energy of nodes in the three algorithms is increased, because all of them use localized target tracking strategies, that is, not all nodes in the network are involved in target tracking. So, the more the number of nodes is, the higher the average residual

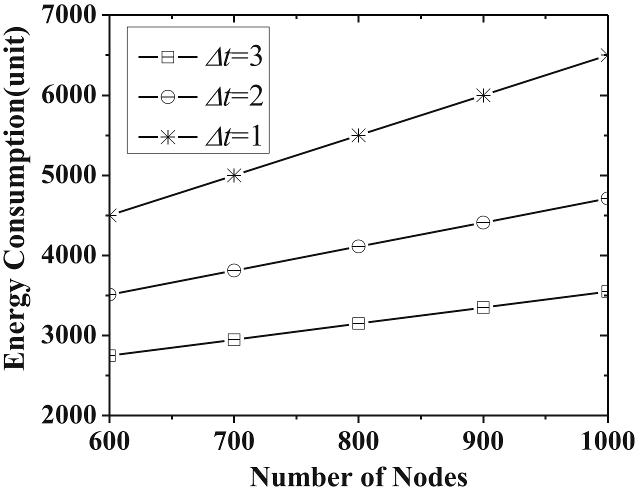


Fig. 25 Energy Consumption of the Whole Network after One Round of Target Tracking

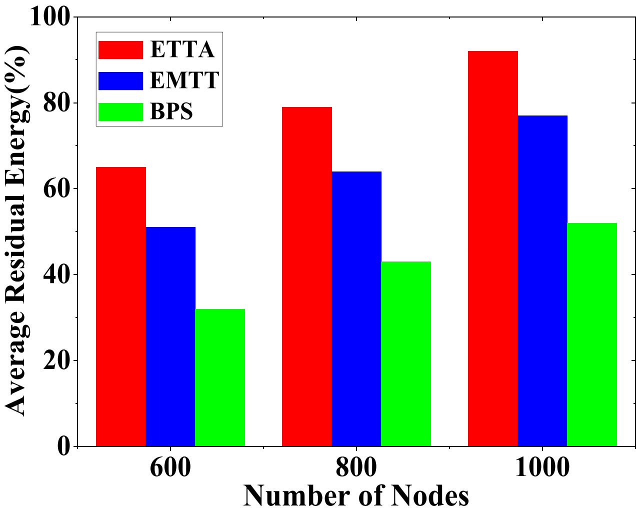


Fig. 26 Mean Value of Residual Energy of Nodes

energy is. Moreover, ETTA not only uses the virtual grid to further refine the tracking area of the target but also selectively changing the working state of the node by accurately predicting the moving position of the target. For example, the grid in preparing-tracking state requires only the boundary nodes to be active, without requiring all nodes to be active. Thus, the average residual energy of ETTA is the highest. When the number of nodes in ETTA is up to 1000, after a round of target tracking, the average residual energy is higher than 90% of the initial energy, which ensures the network's long-lasting, stable operation.

In the real physical world, the energy saving strategies can only prolong the network lifetime to a certain extent. Once the number of dead nodes exceeds a certain value, the whole target tracking system will not work. In response to this potential problem, we have carried out some research [7, 8]. In future, we will research on the more reasonable energy supplement strategy as well as the energy consumption optimization algorithms according to the actual characteristics of the target tracking system.

7 Conclusion

A virtual grids based energy-efficient target tracking strategy for Wireless Sensor Networks is proposed in this paper. States of the virtual grids are set to tracking state, prepared-tracking state and preparing-tracking state that not only enhances the accuracy of target tracking but also prolongs network lifetime.

However, in real-world scenarios, ETTA algorithm still has some limitations, which need to be further improved in the future.

The first problem is how to effectively reduce the complexity of target localization. Although the improved APIT method improves the accuracy, it is undeniable that when the node density is large, there may be more "ideal triangles". Thus, the time complexity of calculation will increase.

Secondly, the proposed algorithm does not take into account the diversity of the actual network environment for the time being. However, in real physical scenarios, the communication effect between nodes may be affected by terrain, interference sources and obstacles. Moreover, in large-scale networks, the connectivity between nodes may also be reduced. All of these may affect the real-time and robustness of target tracking.

Finally, the energy consumption rate of the target tracking system is higher than that of the WSN which only senses. This is because the former requires a lot of computation and communication between nodes. Although the ETTA algorithm reduces energy consumption to some extent by setting the grids to different states, how to maintain the long-term stability of the target tracking system is still one of the key problems that need to be solved in the future.

Funding The subject is sponsored by the National Natural Science Foundation of P.R. China (61872194), Jiangsu Natural Science Foundation for Excellent Young Scholar (BK20160089), Six Talent Peaks Project of Jiangsu Province (JNHB-095), "333" Project of Jiangsu Province, Qing Lan Project of Jiangsu Province, Innovation Project for Postgraduate of Jiangsu Province (KYCX17_0796, KYCX17_0797, SJCX17_0238, SJCX18_0295) and 1311 Talents Project of Nanjing University of Posts and Telecommunications.

References

1. Jiang W, Chen SR, Cai BG et al (2018) A multi-sensor positioning method-based train localization system for low density line[J]. IEEE Trans Veh Technol 67(11):10425–10437
2. Kim S, Kim DY (2018) Efficient data-forwarding method in delay-tolerant P2P networking for IoT services[J]. Peer-to-Peer Networking and Applications 11(6):1176–1185
3. Wang T, Peng Z, Liang J et al (2016) Following targets for Mobile tracking in wireless sensor networks[J]. ACM Transactions on Sensor Networks 12(4):1–24
4. Cui J, Shao LL, Zhong H et al (2018) Data aggregation with end-to-end confidentiality and integrity for large-scale wireless sensor networks[J]. Peer-to-Peer Networking and Applications 11(5): 1022–1037
5. Wu W, Xiong N, Wu C (2017) Improved clustering algorithm based on energy consumption in wireless sensor networks[J]. IET Networks 6(3):47–53
6. Chen T, Chen JJ, Wu CH (2016) Distributed object tracking using moving trajectories in wireless sensor networks[J]. Wirel Netw 22(7):2415–2437
7. Sha C, Wang QW, Zhang L, Wang RC (2018) A high-efficiency data collection method based on maximum recharging benefit in sensor networks[J]. Sensors 18(9):2887–2920
8. Sha C, Liu Q, Song SY, Wang RC (2018) A type of annulus-based energy balanced data collection method in wireless rechargeable sensor networks[J]. Sensors 18(9):3150–3178
9. Satish RJ, Rajkumar SD (2019) Kalman filtering framework-based real time target tracking in wireless sensor networks using generalized regression neural networks[J]. IEEE Sensors J 19(1):224–233

10. Engin M, Abdulkadir K (2018) A proportional time allocation algorithm to transmit binary sensor decisions for target tracking in a wireless sensor network[J]. IEEE Trans Signal Process 66(1):86–100
11. Banaezadeh F, Haghighat AT (2015) Evaluation ARIMA modeling-based target tracking scheme in wireless sensor networks using statistical tests[J]. Wirel Pers Commun 84(4):1–13
12. Guo YN, Cheng J, Liu HY, Gong D, Xue Y (2017) A novel knowledge-guided evolutionary scheduling strategy for energy-efficient connected coverage optimization in WSNs[J]. Peer-to-Peer Networking and Applications 10(3):547–558
13. Souza FL, Pazzi RW, Nakamura EF (2015) A prediction- based clustering algorithm for tracking targets in quantized areas for wireless sensor networks[J]. Wirel Netw 21(7):2263–2278
14. Kung, H. T.; Vlah, D. Efficient location tracking using sensor networks[C]. In Proceedings of the 57th Wireless Communications and Networking Conference, New Orleans, USA, 16–20 March, 2003, 1954–1961
15. Liu BH Effective reconstruction of the message pruning trees in wireless sensor networks[C]. In: Proceedings of the 4th international conference on genetic and evolutionary computing, Shenzhen, China, 13–15 December 2010, pp 695–698
16. Zhang W, Cao G (2004) DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks[J]. IEEE Transactions on Wireless Communication 3(5):1689–1701
17. Mehta K, Liu D, Wright M (2012) Protecting location privacy in sensor networks against a global eavesdropper[J]. IEEE Trans Mob Comput 11(2):320–336
18. Alaybeyoglu A, Kantarci A, Erciyes K (2014) A dynamic look ahead tree based tracking algorithm for wireless sensor networks using particle filtering technique[J]. Computers & Electrical Engineering 40(2):374–383
19. Alberto de SB, Jose RMD, Anibal O (2015) Efficient cluster-based tracking mechanisms for camera-based wireless sensor networks[J]. IEEE Trans Mob Comput 14(9):1820–1832
20. Bhatti S, Xu J, Memon M (2011) Clustering and fault tolerance for target tracking using wireless sensor networks[J]. IET Wireless Sensor Systems 1(2):66–73
21. Teng J, Snoussi H, Richard C et al (2012) Distributed variational filtering for simultaneous sensor localization and target tracking in wireless sensor networks[J]. IEEE Trans Veh Technol 61(5):2305–2318
22. Enayet A, Razzaque MA, Hassan MM et al (2014) Moving target tracking through distributed clustering in directional sensor networks[J]. Sensors 14(12):24381–24407
23. Fu P, Cheng Y, Tang H, Li B, Pei J, Yuan X (2017) An effective and robust decentralized target tracking scheme in wireless camera sensor networks[J]. Sensors 17(3):639–662
24. Jiang B, Ravindran B, Cho H (2013) Probability-based prediction and sleep scheduling for energy-efficient target tracking in sensor networks[J]. IEEE Trans Mob Comput 12(4):735–747
25. Xu Y, Winter J, Lee W-C (2004) Prediction-based strategies for energy saving in object tracking sensor networks[C]. In: IEEE international conference on Mobile data management, Berkeley, CA, USA, 19–22 Jan, pp 346–357
26. Turgut D, Bölöni LIVE Improving the value of information in energy-constrained intruder tracking sensor networks[C]. In: 2013 IEEE international conference on communications, Budapest, Hungary, 9–13 June 2013, pp 6360–6364
27. Taqi RM, Hameed MZ, Hammad AA et al (2008) Adaptive yaw rate aware sensor wakeup schemes protocol (A-YAP) for target prediction and tracking in sensor networks[J]. IEICE Trans Commun 9(11):3524–3533
28. Hsua JM, Chenb CC, Li CC (2012) An efficient object tracking strategy based on short-term optimistic predictions for face-structured sensor networks[J]. Computers & Mathematics with Applications 63(2):391–406
29. Olfati-Saber, R. Distributed Kalman filtering for sensor networks[C]. 46th IEEE Conference on Decision and Control, New Orleans, LA, 12–14 December, 2007, 5492–5498
30. Wang F, Bai X, Guo B (2016) Dynamic clustering in wireless sensor network for target tracking based on the fisher information of modified Kalman filter[C]. In: International conference on systems and informatics, Shanghai, China, 19–21 November, pp 696–700
31. He J, Xiong N, Xiao Y et al (2010) A reliable energy efficient algorithm for target coverage in wireless sensor networks[C]. In: IEEE 30th international conference on distributed computing systems workshops, Genova, Italy, 21–25 June, pp 180–188
32. Chen YR, Lu SY, Chen JJ et al (2017) Node localization algorithm of wireless sensor networks with mobile beacon node[J]. Peer-to-Peer Networking and Applications 10(3):795–807
33. Yuan YL, Huo LW, Wang ZX et al (2018) Secure APIT localization scheme against Sybil attacks in distributed wireless sensor networks[J]. IEEE Access 6:27629–27636
34. Kim W, Mechitov K, Choi JY, Ham S (April 2005) On target tracking with binary proximity sensors[C]. Fourth international symposium on information procession in sensor networks, Boise, Idaho. USA 15-16:301–308
35. Wen Y, Gao R, Zhao H (2016) Energy efficient moving target tracking in wireless sensor networks[J]. Sensors 16(1):1–11

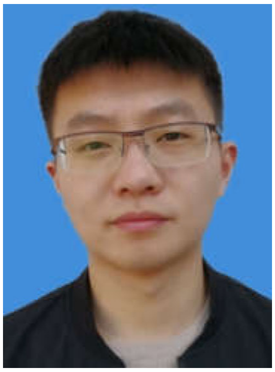
Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Chao Sha was born in Nanjing, Jiangsu Province, China, on July 26, 1983. He received his M.S. degrees and Ph.D. degrees from Nanjing University of Posts and Telecommunications in 2007 and 2010, respectively. He is now an associate professor in Computer Science at Nanjing University of Posts and Telecommunications, China. His research concentrates on mobile sink based data collection and distributed computing architecture in sensor networks. Chao Sha is the corresponding author and can be contacted at: shac@njupt.edu.cn

Lian-hua Zhong received her B.S. degree from Nanjing University of Posts and Telecommunications in 2015. She is now a postgraduate student of Nanjing University of Posts and Telecommunications. Her research interests are data collection algorithm and target tracking technology in sensor networks.

Dan-dan Song received her B.S. degree from Liaocheng University in 2017. She is now a postgraduate student of Nanjing University of Posts and Telecommunications. Her research interests are data collection algorithm and recharging technology in mobile sensor networks.

Yao Bian is an undergraduate student of Nanjing University of Posts and Telecommunications. His main research interest is energy balance technology in sensor networks.



Chun-hui Ren received his B.S. degree from Nanjing University of Posts and Telecommunications in 2017. He is now a postgraduate student of Nanjing University of Posts and Telecommunications. His research interests are sleep scheduling technique and target tracking technology in sensor networks.