

Target Tracking with Monitor and Backup Sensors in Wireless Sensor Networks

Md. Zakirul Alam Bhuiyan¹, Guojun Wang^{1,2,*}, Jie Wu²

¹ School of Information Science and Engineering
Central South University, Changsha, 410083, China

*Corresponding author: csgjwang@mail.csu.edu.cn

² Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431, USA

Abstract—Over the past few years, developing an efficient target tracking system in wireless sensor networks (WSNs) has become a hot research topic. We propose Target Tracking with Monitor and Backup sensors in WSNs (TTMB) to increase the energy efficiency of the network and decrease target capturing time while considering the effect of a target’s variable velocity and direction. The approach is based on a face routing and prediction method. We use a state transition strategy, a dynamic energy consumption model, and a moving target positioning model to reduce energy consumption by requiring only a minimum number of sensor nodes to participate in communication, transaction, and sensing for target tracking. Two sensor nodes, namely, ‘Monitor’ and ‘Backup’, are employed for target tracking for each period of time. For the whole time of target tracking, a linked list of monitor and backup sensors are formed. If either monitor or backup sensor fails, this approach can still be survivable. Simulation results compared with existing protocols show better tracking accuracy, faster target capturing speed, and better energy efficiency.

Index Terms - Wireless sensor networks, target tracking, wakeup mechanism, planarization, energy efficiency.

I. INTRODUCTION

A sensor is a device that detects changes in the environment and records the changes. Many sensors can collaborate to form a wireless sensor network (WSN) which can be used to monitor large areas effectively. One of the most important areas where the advantages of WSNs can be exploited is for tracking mobile targets. Typical examples include establishing survivable military surveillance systems, environmental and industrial monitoring, personnel and wildlife monitoring systems requiring tracking schemes, capable of deducing kinematic characteristics such as position, velocity, and acceleration of single or multiple targets of interest [1].

We propose a novel lightweight approach to implementing target tracking in WSNs. So far, many protocols have been proposed in the literature, while some protocols are designed to increase the lifetime of target tracking wireless sensor networks [1], some others enable good localization methods with low energy consumption [2] and put forward some critical issues such as system robustness, scalability, and fault tolerance. Many of the protocols employ a lot of sensors for target detection and data transmission at the same time. If the number of active sensors is large, it means the tracking accuracy can be high; however, with high energy consumption. Our approach relies on accumulated information from a small number of sensor nodes. The approach can locate and track the target in a

sensing area effectively. They can be achieved by employing low complexity prediction based cooperative tracking that compares the data received from different nodes. The comparative results with existing protocols show better tracking accuracy, faster target capturing speed, and better energy efficiency.

The basic idea of our tracking approach is as follows. An entity that intends to track a target is called a *tracker*. A tracker is assumed to be a single generic source such as a mobile user or a respective authority. A target can be an enemy vehicle, an intruder, or a moving fire. Each sensor in the network has the capability of sensing, communicating, and computing. One of the active and working sensors is elected as a *monitor*, and another one is elected as a *backup* for fault tolerance concern. In the case that the monitor has any problem due to any reason, the backup will take the role of the monitor.

The monitor can work at request of the tracker. When the tracker intends to follow a target, it queries the sensor network. We assume that each sensor knows all its neighbors in a spatial neighborhood – known as a *face* in face routing [3-4]. All the sensors in the network are periodically clock synchronized to be awake, active, or inactive. When a sensor receives a query request, it checks whether it is close to the target, if it is, it then becomes a *monitor* and informs the tracker. The tracker then moves toward the monitor and queries for the target information. If the target is still within the face, the monitor keeps tracking the target; at the same time, the monitor elects the next possible monitor as the new monitor by using our proposed prediction mechanism. The monitor also elects a new backup for fault tolerance concern. The new monitor and the new backup are two common sensors of both the current face and one of its adjacent faces. If the target has already moved out of the area, the monitor informs the tracker about the new monitor, and the tracker moves toward the new monitor.

When the monitor finishes its task, it changes its state. This is also true for the backup. In this way, a special linked list of monitor and backup sensors will be formed as time goes on. If both the monitor and the backup are viewed as one *logical node* at each time step of tracking, this special linked list is simply a linear link of logical nodes, which stands for the tracking route.

To sum up, this research contributes and improves a number of situations as follows.

- The monitor locally chronicles tracking information in its stack and waits for the tracker. The tracker does not need to wait for the tracking information.

- The monitor does not need to transmit information to all neighbors in a face if there is no event of a target missing.
- After computation, the monitor does not need to send the location information to the tracker. It informs the tracker only when it gets a request.
- By directly using predicted-next-location, we aim to simplify the sensor's calculation and minimize the volume of messages exchanged between the monitor as well as the sensors and the tracker.

The main motivation of our work is to shorten target capturing time and prevent the chance of target missing if there is any incident of node failure, routing failure, or loss of tracking. In real WSNs, node or link failure is often possible. The failures may be caused by software or hardware faults, environmental impairment, or battery depletion. In general, a node can be unreachable when such a failure event happens, which may cause partitioning of the network, but can be overcome, for example by avoiding the link and establishing a new link to another node. In this paper, we propose the Target Tracking with Monitor and Backup sensors (TTMB) protocol in WSNs to avoid unpredicted failures through the cooperation of sensors. More specifically, if there is a missing or failure event, we allow neighbor sensors close to the monitor in the face to cooperate. Even in this state, if the target is not sensed, we allow all the neighbors in the face to relocate the target. If the target is still not sensed, we allow all the neighbor sensors in close proximity of the face to relocate the target. Even when all the neighbors fail, TTMB reverts to the initial state.

The rest of the paper is structured as follows. We review the related work in Section II. In Section III, we give our proposed system model. Section IV illustrates the overview and design of the proposed protocol. Performance evaluation is conducted in Section V. Finally, Section VI concludes this paper and discusses our future work.

II. RELATED WORK

Tracking moving target using WSN technology is a thought-provoking and well-established research area [1] [5]. Although extensively researched in the past, this topic still has some important challenges that are unaddressed. In this paper, we concentrate on prediction-based cooperative target tracking in sensor networks. Some prediction-based methods [5-7] are used to predict the location of mobile targets and to allow a limited number of sensors to track a target, and the use of mobile agents for tracking [8]. Mobicast routing [3] for sensor networks is mainly designed for predicting the target's moving direction. Tracking maneuvering mobile targets using a network of cooperative sensor nodes has attracted substantial research interest. To the best of our knowledge, this is the first work in target tracking research combining geographic routing and prediction methods. For instance, Tsai, Chu and Chen [9] have presented a target tracking protocol using sensor networks for mobile users. It is assumed that a mobile target may move in any direction with a constant speed, so the sensors need to be active in all directions. As a result, the number of active sensors is large, leading to high energy consumption. It also induces larger information collection delay, meaning that it incurs wakeup delay, resulting in a large message delay.

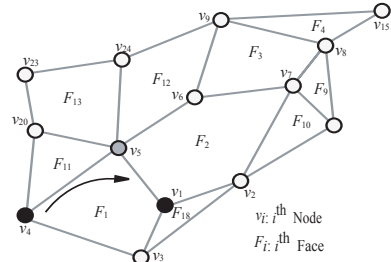


Figure 1. An example of planarized network showing faces

While all the aforementioned research efforts represent valuable contributions in managing tracking accuracy, capturing speed, and energy efficiency tradeoffs, they assume using a large number of active sensors and lack of sensor cooperation in tracking, and the constant motion of a target. Our proposed scheme for fast maneuvering and energy-aware target tracking addresses these shortcomings by considering sensor cooperation, which allows a minimal number of sensors near the target to work in both communication and sensing, and to maximize the lifetime by conserving the energy of the sensor network. Early estimation, various motion, sudden change current motion, appropriate position and velocity, and energy management are addressed by our target tracking scheme.

III. SYSTEM MODELS AND OBJECTIVES

A. Planarization Model

We assume the underlying network graph can be planarized using existing algorithms. A sensor network can be modeled as a graph $G = (V, E)$ by utilizing two well-known distributed planarization algorithms, Gabriel graph (GG) [4] and relative neighborhood graph (RNG) [11]. In GG or RNG, $u \in V$ and $v \in V$ represent sensor nodes, and there is an edge uv between u and v when they are within each other's communication range. All edges $uv \in E$ such that there is no vertex or point w where $uw \in E$, $wv \in E$, and $\|uw\| < \|uv\|$ and $\|wv\| < \|uv\|$. We can obtain a connected planar subgraph $G' = (V, E')$ that maintains connectivity with fewer edges in both GG and RNG. The G' has no intersecting edges. When a network graph has no crossing edges, and it is not unidirectional and disconnected, the graph is planar. A planar graph consists of faces, which are enclosed polygonal regions. In equational form,

$$\forall u, v \in V : (u, v) \in E' \text{ iff} \\ \exists w \in V : \max\{d(u, w), d(v, w)\} < d(u, v) \quad (1)$$

More generally, in any dimension, the GG connects any two points forming the endpoints of the diameter of an empty sphere. In equational form,

$$\forall u, v \in V : (u, v) \in E' \text{ iff} \\ \exists w \in V : d^2(u, w) + d^2(v, w) < d^2(u, v) \quad (2)$$

In order to describe face strategy in the TTMB protocol, we can see from Figure 1 that node v_1 corresponds to 3 adjacent faces, namely, F_1 , F_2 , and F_{18} . Suppose a target is presently in F_2 and v_1 is a monitor node, then F_1 and F_{18} are called *neighbor faces*. So v_1 stores information about 3 faces that are adjacent to it in the planar subgraph - $\{v_1, v_3, v_4, v_5\}$, $\{v_1, v_5, v_6, v_7, v_2\}$ and $\{v_1, v_2, v_3\}$. Node v_1 has only 3 neighbor nodes v_2, v_3 and v_5 , but here we only consider the neighbor nodes with respect to the target position. Thus, v_1 has 2 neighbor nodes, v_5 and v_2 in F_2 , called *immediate neighbors*. While the rest of v_1 's neighbor nodes, v_6

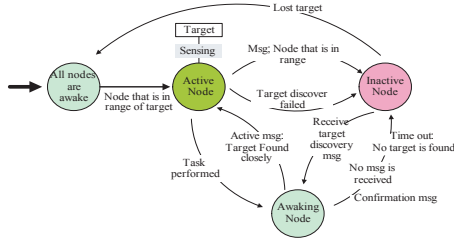


Figure 2. The state transition diagram with three-state model

and v_7 in F_2 are called *distant neighbors*. Similarly, node v_5 has 5 *neighbor faces* with 11 neighbors. If we consider all the faces and nodes corresponding to a node, the network communication cost and energy consumption will be higher. Every node is aware of its own location by using global positioning system (GPS) or other techniques [12-13]. Every node in the graph completely knows all its neighbors in the faces. This concept is inspired by geographic routing [14] and face routing [3-4] in particular.

B. State Transition and Energy Consumption Model

Putting sensors into an inactive state is the most widely-used and cost-effective technique [15-16] to prolong the application lifetime. In this work we presume a sensor node has three different states of operation, i.e. active (s_0), awaking (s_1) and inactive (s_2). Figure 2 gives a general idea of how nodes in the network operate. The most energy efficient state is s_2 in which the sensor turns its service off.

A sensor in state s_2 periodically awakes at a predefined period and changes its state to s_1 . A sensor can only change state to s_0 when it is in state s_1 and it is ensured that it is closest to the target. If it is closest, it then sends a message to the current node to be in state s_1 . If the sensor does not change its state to s_0 in this period, or it has not found the target, its state returns to s_2 . Assume a tracking event is occurred by a sensor node at some time t_0 ; processing is finished at time t_1 ; and the next tracking event occurs at time $t_2 = t_1 + t_i$. According to the state transition diagram in Figure 2, each state s_k has a power consumption P_k , and the transition time to a state and return from the state is given by $\tau_{d,k}$ and $\tau_{u,k}$, respectively. Typically, in different node states, $P_j > P_i$, $\tau_{d,i} > \tau_{d,j}$ and $\tau_{u,i} > \tau_{u,j}$ for any $i > j$, and $\Delta P = P_0 - P_k$ and $\Delta P^+ = P_0 + P_k$. When the node changes state from state s_0 to, say, state s_k , individual components such as the radio, memory, and processor are powered down with time. The energy savings, $E_{s,k}$, because of state transition given by the difference in the face and sleep thresholds $T_{th,k}$ corresponding to the states s_k are computed as follows:

$$E_{s,k} = \Delta P \cdot t_i - \frac{1}{2} (\Delta P \tau_{d,k} + \Delta P^+ \tau_{u,k}) \quad (3)$$

$$T_{th,k} = \frac{1}{2} \left[\tau_{d,k} + \frac{\Delta P^+}{\Delta P} \tau_{u,k} \right] \quad (4)$$

This state transition, as shown in Figure 2, takes advantage of the energy saving feature in sensor networks while the monitor and backup sensors are in active state one by one, and the other nodes typically stay in a periodic awaking or inactive state. To save energy, an energy evaluation model is followed for target detection and positioning [10]. The energy used for communication between nodes and the monitor can be categorized into two types, E_b and E_c . Where E_c is the energy consumed by a sensor node for communication with the

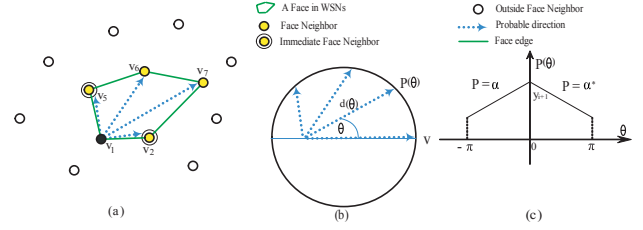


Figure 3. A monitor in a face indicating next probable target's direction

monitor. E_b is the energy needed for broadcasting data from the monitor to the node. For the case of a target moving in the sensor field during the time interval $[t_s, t_e]$, $E(t)$ denotes the corresponding instantaneous energy consumption, E denotes the total energy consumption in the wireless sensor network, and E^* denotes the energy consumption for target positioning. Assume that the number of sensors in a face or neighbor faces requested by the monitor to inform an event is k_q , and the maximum number of sensors is k_{max} . The difference in energy consumption is $\Delta E = E - E^*$.

$$\Delta E = \sum_{t=t_s}^{t_e} E(t) \quad (5)$$

With the proper selection of k_{max} ($k_q < k_{max}$), energy consumption can be greatly reduced as time goes on.

C. Mobile Target Positioning and Movement

One of the most crucial assumptions of the proposed TTMB approach is that each node in the network can locally estimate the cost of sensing, and communicating data to another node for all the tracking behaviors completed around the target without any central intervention.

The monitor can find out a target's position, velocity and direction. Assume the target's present location in oL_i is (x_i, y_i) at time t_i and (x_{i-1}, y_{i-1}) in previous location oL_{i-1} at time t_{i-1} . Then we can estimate the target's speed v and the direction as

$$\Delta E = \sum_{t=t_s}^{t_e} E(t) \quad (6)$$

$$\theta = \cos^{-1} \frac{x_i - x_{i-1}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}} \quad (7)$$

Using these values, the predicted location for the target (x_{i+1}, y_{i+1}) after a given time t is given by

$$x_{i+1} = x_i + v t \cos \theta, \quad y_{i+1} = y_i + v t \sin \theta \quad (8)$$

To be more precise, it can be shown that the target's next location obeys a two dimensional Gaussian distribution with (x_{i+1}, y_{i+1}) as the mean. For the target's speed, we assume that it differs within a range $[0, v_{max}]$. To keep prediction error to a minimum, for the target's direction, we use a linear model based on the value of θ that the target has a higher probability to keep the current direction than to change to another direction, and turning around (making a U-turn) has the least probability. The probability is denoted by p . Figure 3(b) shows the probability that the target moves to different moving directions in a 2D field. Figure 3(c) is the probability of the model, where $\theta \in [-\pi, \pi]$ and the direction of $\theta = 0$ is the instantaneous direction at the current time point. The probability p decreases on the side directions linearly and reaches the minimum value at the direction of $\theta = \pi$, $\alpha = x_{i+1}\theta + y_{i+1}$, $\alpha^* = -x_{i+1}\theta + y_{i+1}$. We use the following equation to describe it.

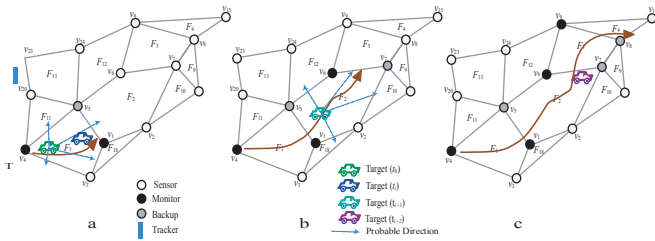


Figure 4. Step by step target tracking in TTMB.

$$p = \begin{cases} \alpha, & \theta \in (-\pi, 0] \\ \alpha^*, & \theta \in (0, \pi] \end{cases} \quad (9)$$

The accuracy of the ‘‘prediction’’ is very important if the monitor is to be identified accurately and the overall tracking protocol is to be effective. While many prediction mechanisms are possible, currently we have chosen a linear predictor, which uses the previous two locations to linearly predict the third location. We also adopt higher order prediction, which predicts the n^{th} location information based on previous $n-1$ actual locations. It gives more accurate results, though, at the cost of much more energy consumption.

IV. PROTOCOL OVERVIEW AND DESIGN

This section discusses the interactions between the tracker T , the target o , and the monitors and backups. The process is illustrated in Figure 4. T sends a flooding request into the sensor network at time t_i , requesting the position of o . Since o is in Face F_1 , T is informed of o ’s current location and that the closest sensor is v_4 (see Figure 4(a)) thus, v_4 becomes the monitor. In F_1 , v_4 has 3 face neighbors that are v_5 , v_1 , and v_3 , where its immediate neighbors are v_5 and v_3 .

If o moves in the direction of v_3 , v_4 is able to easily determine it. As shown in Figure 4(a), if the monitor v_4 estimates that o is moving toward v_1 , v_4 then communicates with v_1 and v_5 . Since T arrives at the vicinity of v_4 after o has moved away, v_4 informs T to follow the same route i.e. go to v_1 (o ’s current position). When o moves from F_1 to F_2 along the route indicated in Figure 4 (b) at time t_{i+1} , only the monitor v_1 or both monitor v_1 and backup v_5 are already aware of o ’s route. We would like to mention here that when a monitor indicates the predicted location where T will be after a given period of time, it determines the most likely node as a new monitor that o is approaching; it also determines an immediate neighbor of the new monitor as backup and informs them in advance about oncoming o .

Node v_1 is now in awaking state and detects o as it moves from F_1 to F_2 . Node v_1 receives the message from v_4 and forwards it in F_2 (v_5), and then checks itself to confirm if it is the new monitor. As a monitor, v_1 senses, and observes o in F_2 . Meanwhile, T reaches the position of v_1 and requests for o ’s status. Monitor v_1 works in the same way and elects new monitor v_6 and backup v_7 compared with o ’s future movements at time t_{i+2} . The sequence (for example $F_1 \rightarrow F_2 \rightarrow F_3 \rightarrow F_4$ and so on) of monitor nodes represents o tracks and T ’s routes.

A detection failure or loss of tracking can occur either due to an event of link failure, node damage, or topological change. TTMB usually avoids a single node failure that causes the partition of a sensor network. When a new monitor fails to detect or is not close to o , the backup takes up the role of the

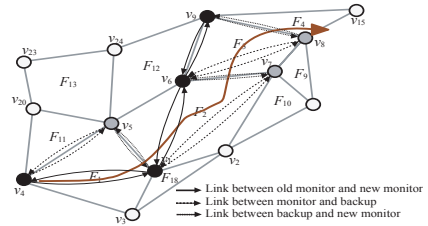


Figure 5. Illustration of communication linked list

monitor. The relationship between the monitor, the new monitor, and the backup is maintained through a low cost implicit linked list among them, as shown in Figure 5. When o moves across the sensing field, the monitor can construct a linked list automatically. We can see from the linked list as shown in the Figure that there is a linear link between the monitor v_4 and the new monitor v_1 , and another link between v_4 and v_5 . We also show a link between v_1 and its backup v_5 . In order to discuss detection failure, suppose v_6 is currently elected as the new monitor and v_7 is the backup; if o is not in the sensing range of v_6 , v_6 may fail to detect o , but in our work we assume if v_6 fails to detect, v_7 cooperates to detect o . If v_7 also fails to detect due to the fact that o may change direction or because of failures, v_7 and v_6 send a detection failure message to v_1 . Old monitor v_1 then sends a message to all its face neighbor nodes to cooperate in tracking o . These neighbors include v_5 , v_6 , v_7 and v_2 , except itself in F_2 .

If the monitor v_1 gets a message about the presence of T , it replies with o ’s detection failure. If o is not detected in F_2 , then v_1 sends a message to all sensors in the neighbor faces corresponding to its immediate neighbors and distant neighbors through them. After that, if o is not sensed in a face, the monitor v_1 sends a request to T to reposition o , T also sends a message to the sensor network to relocate o . This mechanism prevents the chance of routing failure, node failure, as well as loss of tracking. Additionally, if o unfortunately stays in a void region while T gets an o missing message, it means T cannot get the next destination information and then it can hear from the sensor, and relocate o if it receives a request after a predefined period. When o leaves the void region, it can be detected in a face, and T can obtain its location. The proposed TTMB target tracking protocol consists of two levels, one runs at each monitor and another one runs at sensor node. Both levels of protocol are message driven, which means all operations are triggered by messages received from the outside. TTMB uses a number of message types which are exchanged between the sensor, the monitor, as well as the tracker.

TTMB is derived from the cooperation of sensors. Whenever an o is detected, an *election process* will be conducted by the monitor and backup sensors to choose a sensor, on which a tracker will be initiated to observe the movement and behavior of o . The following steps describe how the sensor nodes operate with reference to the diagram in

TABLE I. SIMULATION PARAMETER SETTINGS

Description : Values	Description : Values
Field size: 300 m x 300 m	Communication range: 40m
Total number of nodes: 90	Sensing range: 20m (approx.)
Processor sleep (mW): 0.8	Velocity of the target (m/s): 0 – 20
Transmission power (mW): 720	Simulation per run: 1500s
Sensor activate (mW): 23	Receiving power (mW): 360

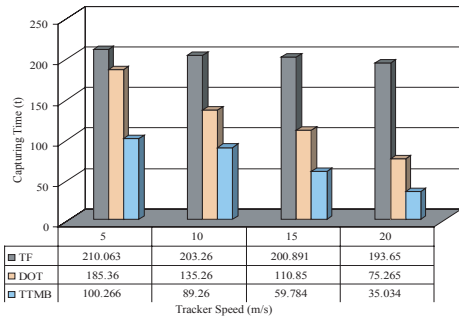


Figure 6. The time target being captured first time

Figure 3: 1) Initialization, 2) Target detection, 3) Tracker movement, 4) Location and direction estimation and 5) Monitor election process.

V. SIMULATION STUDIES

A. Simulation Settings

We conducted extensive simulations using OMNet++ software [17] and the Mobility Framework fw2.0p3 (<http://mobility-fw.sourceforge.net/>). The version of OMNet++ is 3.32. We set up a basic network of 300m x 300m, where nodes are randomly deployed. The initial energy on all the nodes is 40J. The target node velocity is from 0.5 m/s to 20 m/s followed by the target movement model. Tracker node speed varies between 2 m/s, 5 m/s, 10 m/s, 15 m/s, and 20 m/s. Each experiment is run for 1,500 seconds. The settings of the simulation environment are shown in TABLE I. We design four experiments to evaluate the effects of different parameters on the protocol's performance, keeping energy efficiency in mind. These parameters include the target speed, distance, direction, and the number of nodes.

B. Comparison with Other Protocols

In the simulations we compare our proposed protocol with dynamic object tracking (DOT) [9]. We also compare our proposed protocol with flooding-based target tracking. However, the flooding-based query method for target tracking involves broadcasting raw messages as soon as a node senses a target; then each message is broadcast in the entire network. In this way, whenever a node needs to send a message, it broadcasts the message to the entire sensor network. In our work we allow the flooding method in the initial state only for target discovery. The DOT protocol has better performance and energy efficiency than the flooding-based target tracking. There are three kinds of flooding query methods, i.e. threshold flooding (TF), schedule flooding, and schedule updating. Our proposed algorithm is compared with TF.

In DOT and TF protocols, the target discovery process is similar. When a tracker obtains the location of a moving target, it moves toward the obtained location and carries out the target discovery process repetitively, however the TTMB algorithm has no repetition of the target discovery process, except in the special case of target missing. In DOT, a mobile target may move in any way, so in all the ways the sensors need to be active, and it consumes energy and incurs wakeup delay, which can lead to long message delays when a large volume of messages are exchanged. When the distance between the target and the tracker is longer, DOT uses face-track to reach the face

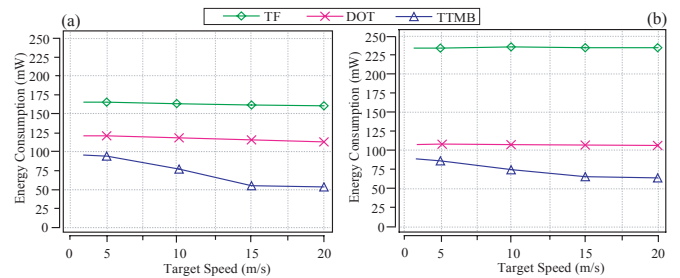


Figure 7. Average energy consumption (a) before the target is captured (b) after the target is captured

where the target is in. The term *face-track* is a path shortening method for adjusting a number of faces when the tracker positioning face is not immediately after the target positioning face, but can be at least two or three faces away. In that case DOT uses an optimal route to reach the location of the face where the target is. The drawback of the face-track is that it lets the tracker wait for a long time, though the used face-track in DOT is not the optimal track. In addition, there are more chances of target missing. In DOT, there is no consideration of when a network failure happens, although a lot of sensors are occupied to track a target at a timestamp. We have found a significant improvement for all these situations using TTMB.

C. Simulation Results

1) *Study of the target capturing time (t)*: First, we compare the performance of TTMB protocol with DOT protocol and flooding-based tracking based on target capturing time. We calculate the time for when the target is captured for the first time. We can see from Figure 6 that the tracker can follow the target comparatively faster. When the tracker's velocity is fast, it can capture the target in a short time. This is because the tracker does not need to use flooding often. It also does not need to adjust the face-track. The tracker does not need to wait long because the monitor predicts the next location in advance and sends the target's information timely by means of an efficient *linked list*. Thus, the tracker can follow the mobile target quickly.

2) *Study of the target moving speed (v)*: We take the target's variable moving speed and direction into consideration by introducing target moving direction probability mentioned in the model. In order to analyze the energy consumption, we observe on the difference between our TTMB protocol and others in terms of energy consumption, as shown in Figure 7.

In the experiments, we observe the target moving with a changing speed. We change the target speed v from 2 m/s to 20 m/s. We see when the target speed is about between 5 m/s and 15 m/s, energy consumption is drastically reduced, where DOT and TF are typical. It can be seen that the TTMB approach consumes less energy than DOT and flooding based tracking, because fewer sensors are used in TTMB protocol by using the energy consumption model for tracking, and also flooding does not need to be used in a repetitive manner for discovering the target. When the velocity of the target is faster than that of the tracker, it is difficult for the tracker to capture the target. The energy consumption is averaged by the data gathered from 50 uninterrupted simulations with different parameters. When the tracker velocity increases, average energy consumption

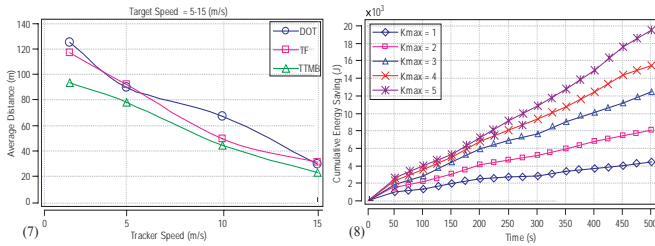


Figure 8. Relative distance between target and tracker considering speed

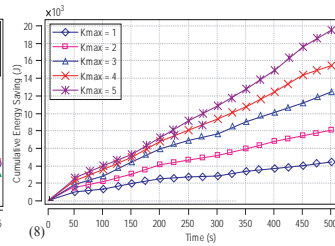


Figure 9. Cumulative energy consumption as time goes

decreases in TTMB as shown in Figure 7, but the missing rate increases (not shown here).

3) *Study of the relative distance (m)*: Figure 8 illustrates the difference in distance between the target and the tracker with respect to different speeds. As with prediction error (not shown here), the distance between target and tracker is the greatest at turns or a change in direction. The results of the average distance are received from 0 ~ 600s. It can be seen from Figure 8, TTMB has better performance than DOT and TF in both slow and high speeds. It is because the tracker gets the target location information from the monitor immediately and the monitor informs the new monitor and backup through the linked list in advance by using the prediction method and movement model when the target changes its moving direction. When the monitor is not acknowledged by the new monitor, the backup becomes a monitor. Hence the distance between the tracker and the target can be longer, but after the target is sensed, the tracker's speed can be increased to reach the new monitor.

4) *Study of the energy conservation (AE)*: Figure 9 shows the value of the cumulative energy savings as the target moves along in the sensor field. It is clear from Figure 9 that a large amount of energy is saved during target tracking. Nevertheless, a considerable amount of energy is saved in target positioning, even when the maximum number of used sensors, $k_{max} = 4$ or 5. When a target is not in the sensing area, the monitor queries all of the neighbors in a face or neighbor faces. When there is any event of link failure or node failure, the monitor elects a new monitor and establishes a new link. In that case, the value of k_{max} is increased. When we try the value of $k_{max} = 3$ to 5, the proposed scheme works well. Figure 9 presents the cumulative energy consumption as time goes on. It is important to say that DOT has neither prediction method nor description of faulty node avoidance, so there is no energy consumption for prediction error and linked list formation, but when the tracker is not close to the target or the target is out of sensing range of the monitor, the nodes in the neighbor faces are always used. In that case, the value of k_{max} can be 8, 10, or more, though DOT has no definition for the neighbor face. The Figure clearly shows that TTMB saves much more energy than others.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new scheme to detect and track a mobile target efficiently in wireless sensor networks. Most of the existing works were on how to track a target accurately, but used a lot of sensors to track the target. Taking the energy constraints into account, we used several practical implementation methods to exploit the energy management issues in target tracking sensor networks. Simulation results

showed that our scheme greatly contributed to energy conservation by keeping good tracking accuracy and still achieve fast target tracking. Failure of one or a few nodes does not affect the operation of the network during target tracking due to its fault tolerance capability. This work has room for further improvements in some areas, as follows: 1) Finding a better technique for position estimation while considering error avoidance and a deeper analysis of the measurement uncertainties. 2) Investigating the issue of quality tracking vs. energy consumption of the entire network and the target missing probability vs. sensing range or moving speed and 3) Evaluation of the cost of fault tolerance in the algorithms with complex scenarios, and effect of localization errors on face routing for target tracking.

REFERENCES

- [1] T. He, P. Vicaire, T. Yan, L. Luo, L. Gu, G. Zhou, R. Stoleru, Q. Cao, J. Stankovic, and T. Abdelzaher, "Achieving real-time target tracking using wireless sensor networks," *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2006, pp. 37-48.
- [2] K. T. Soe, "Increasing lifetime of target tracking wireless sensor networks," *World Academy of Science Engineering and Technology*, Aug. 2008, Vol. 32(32), pp. 440-446.
- [3] Q. Huang, S. Bhattacharya, C. Lu, and G.-C. Roman, "FAR: Face-aware routing for mobicast in large-scale sensor networks," *ACM Transaction on Sensor Networks*, 2005, pp. 240-271.
- [4] B. Leong, S. Mitra, and B. Liskov, "Path vector face routing: geographic routing with local face information," *13th IEEE International Conference on Network Protocols*, 2005, pp. 147-158.
- [5] Y. Xu, J. Winter, and W.-C. Lee, "Prediction-based strategies for energy saving in object tracking sensor networks," *IEEE International Conference on Mobile Data Management*, 2004, USA, pp. 346-357.
- [6] X. Wang, J.-J. Ma, S. Wang, and D.-W. Bi, "Prediction-based dynamic energy management in wireless sensor networks," *IEEE Sensors 2007*, Atlanta, USA, Vol. 7, pp. 251-266.
- [7] K. Niyogi, S. Mehrotra, N. Venkatasubramanian, and X. Yu, "Adaptive target tracking in sensor networks," *Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2004, pp. 253-258.
- [8] H. Qi, and Y. Xu, "Mobile agent migration modeling and design for target tracking in wireless sensor networks," *Ad Hoc Networks*, (Elsevier Journal)-2008, Vol. 6, pp. 1-16.
- [9] H.-W. Tsai, C.-P. Chu, and T.-S. Chen, "Mobile object tracking in wireless sensor networks," *Computer Communications*, Elsevier Journal, 2007, Vol.30, pp.1811-1825.
- [10] Y. Zou, K. Chakrabarty, "Energy-aware target localization in wireless sensor networks," *IEEE International Conference on Pervasive Computing and Communication*, USA, 2003, pp.60-66.
- [11] Toussaint GT. "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, 1980, Vol. 12(4), pp. 261-268.
- [12] H.-C. Chu and R.-H. Jan, "A GPS-less self-positioning method for sensor networks," *11th International Conference on Parallel and Distributed Systems*, 2005, Vol. 2, pp. 629-633.
- [13] R. Stoleru, J. A. Stankovic, and S. Son. "Robust node localization for wireless sensor networks," *4th workshop on Embedded networked sensors*, 2007, pp. 48 - 52.
- [14] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," *Symposium on Network System Design & Implementation*, USENIX Association, USA, 2005, Vol. 2, pp. 217 - 230.
- [15] O. Yang and W. Heinzelman, "A better choice for sensor sleeping," *European Conference on Wireless Sensor Network*, Ireland, February 2009. in press.
- [16] H. Wang, W. Wang, D. Peng, and H. Sharif, "A route-oriented sleep approach in wireless sensor network," *IEEE Singapore International Conference on Communication System*, 2007, Vol. 7(4), pp. 1-5.
- [17] Discrete Event Simulation System, <http://www.omnetpp.org>.