Herd-Based Target Tracking Protocol in Wireless Sensor Networks

Xiaofei Xing[1], Guojun Wang[1,2,*], and Jie Wu[2]

[1] School of Information Science and Engineering, Central South University, Changsha 410083, China
[2] Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA
* Corresponding author: csgjwang@mail.csu.edu.cn

Abstract. Target tracking is a killer application in wireless sensor networks (WSNs). Energy efficiency is one of the most important design goals for target tracking. In this paper, we propose a herd-based target tracking protocol (HTTP) with the notions of *node state transition* and *herd-based node group* for target tracking. A sensor node has three states, namely, sleeping state, sensing state, and tracking state. Each sensor node is associated with a weight to be used to make a state transition among the three states. When a target moves into a monitoring area, a cluster node is selected as the herd head that is responsible for reporting the target information to the sink in the network. The sensor node can adjust the frequency of data reporting according to the velocity of the target. Simulation results show that HTTP not only improves the energy efficiency, but also enhances the tracking accuracy.

Key words: wireless sensor networks, target tracking, energy efficiency, node state transition, herd-based node group

1 Introduction

Wireless sensor networks (WSNs) [1], [2] consist of a set of sensor nodes, each of which is a self-contained unit with a low-speed processor, one or multiple sensors, a radio module, and a battery module. The sensors in a WSN system are deployed over an area in an attempt to sense and monitor interesting events or to track the mobile targets or people as they move through the area. So it is widely used in military battlefields, environmental monitoring, traffic transportation, medical diagnosis, and many other fields. Target tracking is one of the killer applications in WSNs. Because of the uncertain movement of the target, it becomes a challenging issue to position and track the moving target effectively and efficiently.

WSNs have the advantage in high accuracy, real-time, and low cost in tracking the target by using random and uniform distribution and mutual cooperation of sensor nodes. Because of energy constraints of sensor nodes, which are closely related to the lifetime of the network, the designed target tracking protocol should be able to track the target accurately with low energy consumption. This study can be applied to many target tracking scenarios, such as military tank movement, vehicle movement, wild animal environments, and so on.

We study the issue of tracking mobile targets using wireless sensor networks. The whole tracking process is divided into the positioning stage and the tracking stage. In the process of target tracking, a lot of factors [3] affect energy consumption, which includes the number of moving targets, the speed of moving targets, data reporting frequency, tracking data accuracy, data collection frequency, and so on. Obviously, as design goals, the sensor nodes surrounding the moving target should be able to promptly provide reliable status information about the moving target and the area around it in an energy efficient way, and the sensor nodes should report this information to the sink in a fast and energy efficient way.

In this paper, we propose a herd-based target tracking protocol (HTTP) for single-target tracking in WSNs. In summary, the key contributions of this paper are as follows.

- The herd-based node group which consists of sensor nodes in the tracking state can track the target dynamically by using the node state transition mechanism and weight calculation mechanism.
- An appropriate threshold can be obtained to decide whether the sensor nodes should participate in tracking target or not.

- The data reporting frequency can be adjusted according to the velocity of the target to reduce unnecessary data transmission and improve the energy efficiency.

The rest of the paper is organized as follows: Section 2 summarizes some existing target tracking protocols in WSNs. Section 3 discusses the basic ideas behind the proposed HTTP protocol, including node state transition, node weight calculation, and the initialization and reconstruction of the herd-based node group. An extended HTTP is discussed in Section 4. We simulate and evaluate the proposed HTTP protocol in Section 5. Section 6 concludes the paper.

2   Related Work

Target tracking in wireless sensor networks has been investigated extensively. As one of the fundamental problems with target tracking, energy efficiency has been researched from different aspects. Lee et al [4], Niyogi et al [5], and Wang et al [6] propose prediction mechanisms to restrict the amount of transmitted messages and to select the nodes that participate in target tracking. A dual-prediction-based data aggregation scheme to decrease the communication overhead is proposed in [7]. Considering the tradeoff between tracking quality and energy efficiency and guaranteeing an acceptable tracking quality, some strategies are proposed in [8], [9] to keep the sensor nodes in sleeping state as long as possible.

Guo et al [10] propose a target tracking protocol based on energy efficiency which includes two energy efficient algorithms, i.e., RARE-Area and RARE-Node algorithms. The RARE-Area algorithm reduces the emergence of low quality data and the amount of involved tracking sensors. So only sensor nodes that can generate high quality data are permitted to track the target. The RARE-Node algorithm considers the spatial relationship among sensors in order to determine whether the data generated by a node is redundant or not.

He et al [11] present the design of an analysis of VigilNet, a large-scale outdoor WSN which detects, classifies, and tracks a target in a timely and efficient manner. Through simulation and experiments, the authors demonstrate that their system can meet the real-time requirement and their tradeoffs are validated. On the basis of the deadline partition method and theoretical derivations to guarantee each sub-deadline, they make a guided engineering decision to meet the end-to-end tracking deadline.

Lee et al [12] propose a distributed energy-efficient target positioning and tracking algorithm, and study the RVI positioning issue based on distance ratio. Moreover, they propose a scheme that dynamically adjusts the lead nodes' reporting frequency, so that the amount of the status information of the target and energy consumption can be reduced.

A dynamic convoy tree-based collaboration scheme is introduced in [13], [14]. The moving target can be tracked effectively by the reconfiguration convoy tree and node expansion and pruning. However, it incurs a significant amount of processing data and system overhead. Our work is most closely related to this work but our proposed scheme drastically decreases the overhead, while still keeping a good accuracy for tracking.

3   Basic HTTP (B-HTTP)

In this section, a basic herd-based target tracking protocol (B-HTTP) is proposed, including node state transition, node weight calculation, and the initial construction and reconstruction of the herd-based node group.

3.1   Basic Ideas

In B-HTTP, each sensor node has three states, namely, sleeping state, sensing state, and tracking state. We set a weight for each sensor node. The node calculates its weight to decide whether it should participate in target tracking or not according to the target's situation. When a target moves into a monitoring area, sensor nodes that are awake and closer to the target can detect the target. There will be more sensor nodes being transformed into the tracking state. Thus, these sensor nodes construct a dynamic tracking group surrounding the target. We call this dynamic tracking group a *herd-based node group*. One of the cluster heads is selected as a *herd head*, which is responsible for data reporting and also for managing the membership of the herd-based node group. At the same time, we select another cluster head in the herd-based node group to be a *backup herd head*. Both the herd head and the backup herd head store the same
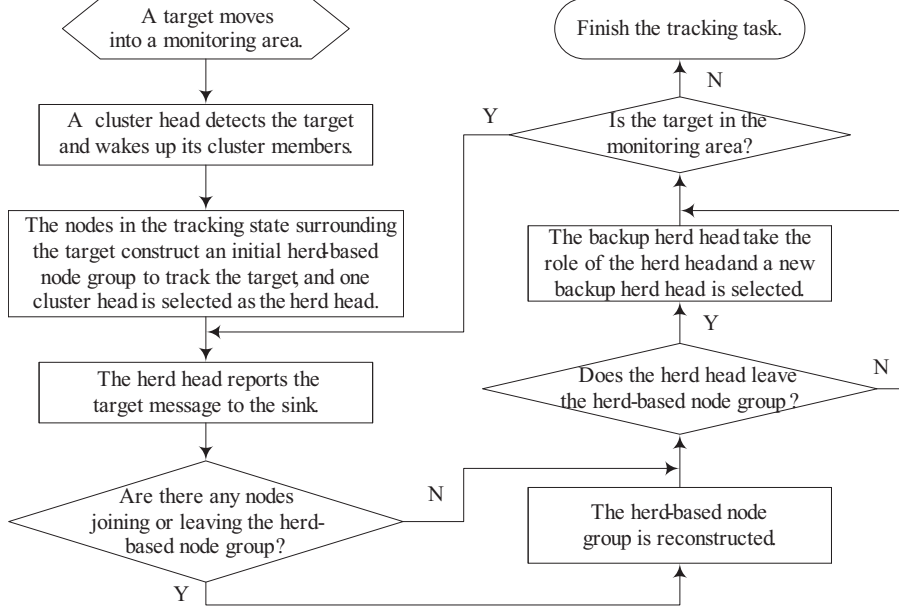
Fig. 1. Basic procedure of B-HTTP.

information, such as membership information. The difference between them is that the backup herd head needs not to report the target information to the sink. The herd head sends the information to the backup herd head periodically. When the herd head can not take its role for the herd-based node group for some reason, for example if it runs out of energy, the backup herd head can change to take the role of the herd head by also reporting the data to the sink, that is, the backup herd head becomes the herd head. At the same time, a new backup herd head needs to be selected.

In this scheme, the membership of the herd-based node group is dynamic. With the moving of the target, some new nodes join this to track the target and some nodes leave as the target is not within their sensing range. The member nodes of the herd-based herd group send the target information to the herd head with a fixed frequency. Then the herd head is responsible for reporting the target information to the sink. In Section 3.5 and Section 3.6, we will discuss how to construct and maintain the herd-based node group. The basic procedure of B-HTTP is shown in Fig. 1.

3.2 Assumptions

We know that the large part of sensor node energy is dissipated on sensing module, processing module, and wireless communication module. With advances in integrated circuit technology, energy consumption on sensing module and processing module become very low. Most of energy is dissipated on the wireless communication module [16]. So we mainly consider energy consumption dissipated on the communication module in this paper.

We adapt a wireless communication model proposed in [17] to calculate node energy consumption. To transmit and receive a k-bit message at distance d, the communication expenditure can be calculated as follows:

$$E_{T(k,d)} = E_{T-elec}k + \varepsilon_{amp}d^2k \tag{1}$$

$$E_R(k) = E_{R-elec}k \tag{2}$$

Where E_{T-elec} and E_{R-elec} denote energy consumption in the transmitter and receiver circuitry respectively, and ε_{amp} is the transmit amplifier. We assume $E_{T-elec} = E_{R-elec} = 50nJ/bit$, $\varepsilon_{amp} = 100pJ/bit/m^2$.

In this paper, we assume that wireless sensor nodes are homogeneous, that is, all the sensor nodes have the same sensing radius and communicating radius. Each sensor node has a unique *id*. Each node is aware
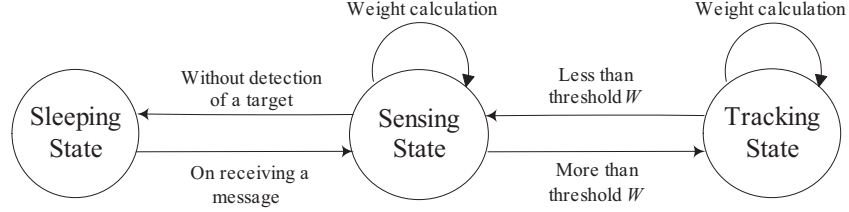
Fig. 2. Node state transition diagram.

of its own geographic location by using the global positioning system (GPS) or trilateration [18]. Here, three range measurements are required to locate the target. The distance information can be obtained from the received signal intensity measurements. A sensor node can get its geographic location information by using the two or more distance measurements which come from its neighbor nodes.

The sensor nodes are deployed densely over an area of interest. Based on a clustering algorithm called geographical adaptive fidelity (GAF) [15], the monitoring area is divided into some virtual *grids*. We call each virtual grid a *cluster*. Each node is assigned to a corresponding grid according to its geographic location. One node is selected periodically as a *cluster head* in each grid. The cluster head can manage its cluster member nodes.

3.3 Node State Transition

Each sensor node has three states, namely, sleeping state, sensing state, and tracking state. In the sleeping state, the communication module of the node is closed to preserve energy. The sensing state means that the node can monitor the surrounding environment, detect whether a target appears or not, and calculate the distance to the target. But the sensor nodes do not send data packets to the cluster head in this state. The most energy is dissipated when the nodes are in the tracking state because they not only monitor the surrounding setting, such as the target moving in or out, and collecting target information, but also transmitting or forwarding the data packets to the herd head. The node state transition mechanism is shown in Fig. 2.

From Fig. 2, we can see that the node in the sleeping state transforms into the sensing state when it receives a beacon message from its cluster head or neighbor nodes. The sensor node in the sensing state calculates its weight periodically. If a node's weight is larger than the threshold after the calculation of weight, it transforms into the tracking state automatically. Otherwise, the node transforms into the sleeping state when there is no target to appear. The node in the tracking state periodically calculates its own weight. If the weight of the node in the tracking state is less than the threshold W, it transforms into the sensing state automatically. The nodes in the tracking state can communicate with other nodes. In addition, the node state can not be transformed directly between the tracking state and sleeping state because we consider some other situations, for example when the target soon moves in the node sensing range again after it leaves.

3.4 Node Weight Calculation

We know that the further away a sensor node is from a target, the weaker the received signal intensity is. Therefore, the nodes near the target have a high qualification to participate in target tracking compared to those far away. So we can use the distance as its weight indicator. In order to reduce low quality redundant data, we should decrease the number of nodes that participate in target tracking when guaranteeing tracking quality. Each node that is not in the sleeping state can calculate its own weight according to the moving target information. Because of the noise signal interference, the effective sensing radius of the node ranges from 0 to $1.2R_s$, where R_s denotes the node's sensing radius. We use W_d and W_{dir} to denote distance weight and direction weight, respectively. The allocated weights for the sensor node are listed in Table 1.

The total weight of node W_{tol} can be calculated below:

$$W_{tol} = W_d + W_{dir} \tag{3}$$

Table 1. The weight allocation for a sensor node

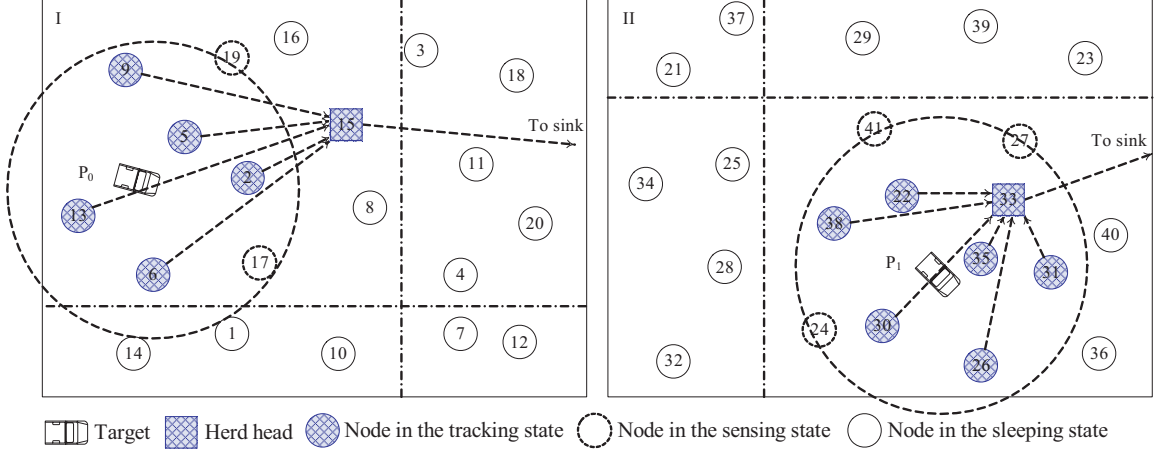| Weight	Description	Value
W_d	$d \leq 1/3R_s$	4
	$1/3R_s < d \leq 2/3R_s$	3
	$2/3R_s < d \leq R_s$	2
	$d > R_s$	1
W_{dir}	Target moves towards sensor	1
	Target is stationary	0
	Target moves away from sensor	-1

Fig. 3. An illustration of the target moving from P_0 to P_1 in the monitoring area.

Obtained from equation (3) and Table 1, we can get a maximal weight value of 5 for the sensor node when the target moves towards the sensor and the distance between the target and the node is less than $1/3R_s$. It means the node owns an optimal sensing capacity. Also, a node would not keep tracking the moving target when its weight is 0. A node can participate in tracking the target when its weight value is more than the threshold. Setting a high threshold decreases positioning accuracy of the target. A greater node threshold means fewer nodes are involved in tracking.

3.5 Initial Construction of the Herd-Based Node Group

When a target moves into a monitoring area, the cluster heads that are awake and closer to the target first detect the target. Then the cluster head sends a beacon message to wake up the cluster member and its neighbor cluster heads. The nodes in the sleeping state transform into the sensing state and calculate their weights according to the target information. When their weight is larger than the threshold, the nodes transform into the tracking state. These sensor nodes in the tracking state form an initial herd-based node group. The cluster member can communicate with the cluster head directly. The amount of clusters that participate in target tracking is only determined by the grid size. For example, there are at least 4 clusters which can detect the target simultaneously if the size of a cluster equals to $\frac{\sqrt{2}}{2}R_s$. As shown in Fig. 3 (left), after the target moves into the monitoring area from grid I, some nodes are woken up by cluster head *node* 15. *Nodes* 9, 5, 2, 15, 6, 13 form an initial herd-based node group and cluster head *node* 15 of grid I is selected as the *herd head* because it first detects the target.

The nodes in the sensing or the tracking state calculate their own weight according to the collected information of the target. When their weights are larger than the threshold, the nodes transform into the tracking state automatically to track the moving target. The nodes report the target information to the herd head. The information, including timestamp, node *id*, and the distance d to the target etc, is sent to its

```
/* Algorithm 1: Node state transition */
Wthr: the threshold value.

1. If (Target moves into a monitoring area) {
2. Switch (Node state) {
3.     Case (Sleeping state) {
4.            If (Receive cluster head's message)
5.                Transform into sensing state;
6.            Else Keep sleeping state;
7.            Break; }
8.     Case (Sensing state) {
9.            If (W>Wthr) Transform into tracking state;
10.           Else If (No target appear || W≤Wthr)
11.                Transform into sleeping state;
12.           Else Keep sensing state;
13.           Break; }.
14.    Case (Tracking state) {
15.           If ( W>Wthr) Keep tracking state;
16.            Else Transform into sensing state;
17.            Break; }
18.    }
19.}
```

Fig. 4. Node state transition algorithm.

cluster head. Then, the cluster heads (excluding the herd head) send the collected target information to the herd head. The herd head reports the target information to the sink after data aggregation processing.

3.6 Reconstruction of the Herd-Based Node Group

In order to track the target effectively, the dynamic herd-based node group needs to be reconstructed dynamically. It mainly includes the herd membership management and the herd head selection mechanism.

We use the node state transition mechanism to decide whether a node should join or leave the herd-based node group. The sensor nodes in the tracking and the sensing states calculate their own weight periodically. When the target moves to a new grid, the nodes in the tracking state are divided into two kinds: some nodes transform into the sensing state when their weight is less than the threshold, and the others still stay in the tracking state to track the moving target. The node state transition is shown in Algorithm 1 (Fig. 4).

The nodes in the sensing state have two types. The first type is then the nodes cannot detect the moving target because the distance between the nodes and the target is beyond the nodes' sensing range. Therefore, these nodes transform into the sleeping state automatically. The second type is that the nodes transform into the tracking state as the nodes' weights are larger than the threshold.

When the target moves into a new grid, the cluster head wakes up its cluster member to track the target. The nodes that just detect the target transform into the tracking state and join the herd-based node group. Other cluster members whose weights are less than the threshold, will leave the herd-based node group. The herd head calculates its weight. It will keep on tracking if the weight is larger than the threshold. When the herd head can not take its role for the herd-based node group for some reasons, the backup herd head can change to take the role of the herd head by also reporting the data to the sink. Due to the reconfiguration cost, the herd-based node group should be reconfigured based on the distance between the herd head and the current location of the target. If this distance is larger than a certain threshold, the herd-based node group should be reconfigured. In order to decrease the overhead on calculations, we specify the cluster head that lastly joins the herd-based node group to act as the backup herd head. So this herd-based node group

is reconstructed according to the target information. As shown in Fig. 3 (right), we can see that *nodes* 38, 22, 33, 35, 31, 30, 26 reconstruct a new herd-based node group and *node* 33 is selected as a new herd head when the target moves from P_0 to P_1.

4 Extended HTTP (E-HTTP)

In the last section, we designed the basic HTTP protocol by which the target information can be sent to the sink. In order to get a better performance for the proposed HTTP, we further design an extended HTTP (E-HTTP) to decrease energy consumption which can adjust the frequency of data reporting dynamically.

The frequency of the nodes reporting the target information to the sink can be determined by specific applications. The nodes periodically report about the moving target information regarding the reporting sensor node's *id*, tracking time, target's location and velocity, moving direction, and so on. As we do not know what speed the moving target is, it would bring some drawbacks as follows if the node adapts a fixed frequency for sending the data to the sink.

- Positioning accuracy decreases when the speed of moving target increases;
- Sensor nodes consume more energy if they report the target information to the sink with a high frequency while the target moves at a low speed;
- Sensor nodes run out of energy prematurely. If the moving target keeps stationary for a long time in the monitoring area, the node's energy would be exhausted as the node keeps tracking the target. Thus, it may cause the monitoring area to not be fully covered by the sensor nodes and the lifetime of the network to be shortened.

Therefore, if the frequency of data reporting can be adjusted according to the target situation, it can improve energy efficiency and data accuracy.

Before the backup herd head takes the role of the herd head, the herd head sends the location L_{pre} of the target at a previous time T_{pre} to the backup herd head. When the backup herd head takes the role of the herd head, we can get a new location L_{cur} of the target at current time T_{cur} using the positioning algorithm [18]. Thus, we get an average velocity of the target from T_{pre} to T_{cur}.

$$\overline{v} = \frac{|L_{cur} - L_{pre}|}{T_{cur} - T_{pre}} \tag{4}$$

Then we can get the frequency of the data reporting sent by the sensor node.

$$f = \lfloor \overline{v}/D_{th} \rfloor + 1 \tag{5}$$

Where D_{th} is a distance threshold for data reporting.

From Lee et al [12] we can get a formula as below.

$$D_{th} \le 2R_{\max} - R_s - \frac{D_{inter-sensor}}{\sqrt{2}} - 2E_{\max} \tag{6}$$

Where R_{max} is the maximal communication radius of sensor node, R_s is the sensing radius of sensor node, and $D_{inter-sensor}$ is the pre-determined average inter-sensor distance. Suppose that $R_{max}=R_s=40$m (worst case), $D_{inter-sensor}=10$m, and $E_{max}=5$m. Then, approximately, D_{th} is 12.9m or less. It denotes that the frequency of sending message to the sink is $f = \overline{v}/12 + 1$.

5 Performance Evaluation

5.1 Simulation Model

In order to test and evaluate the performance of the proposed HTTP protocol, we set up the simulation in a $600{\times}600m^2$ area based on the OMNET++[19] experimental platform.

In our simulation, the communication radius of a sensor node is 40m, and the monitoring area is divided into $14{\times}14m^2$ virtual grids, which can guarantee that one cluster head can communicate with 8 neighbor cluster heads directly. In each experiment, 1,000 or 5,000 sensor nodes are deployed randomly and uniformly to simulate a sparse setting or a dense setting, respectively. The parameters in our simulation are listed in Table 2.

Table 2. Simulation parameters

Parameter	Description	Value
l	Field size(m)	600×600
N	Number of nodes	1,000(sparse setting) or 5,000(dense setting)
R_c	Communication radius(m)	40.0
R_s	Sensing radius(m)	40.0
V	Speed of a moving target(m/s)	5.0-25.0
S_m	Size of message(bit)	1,000
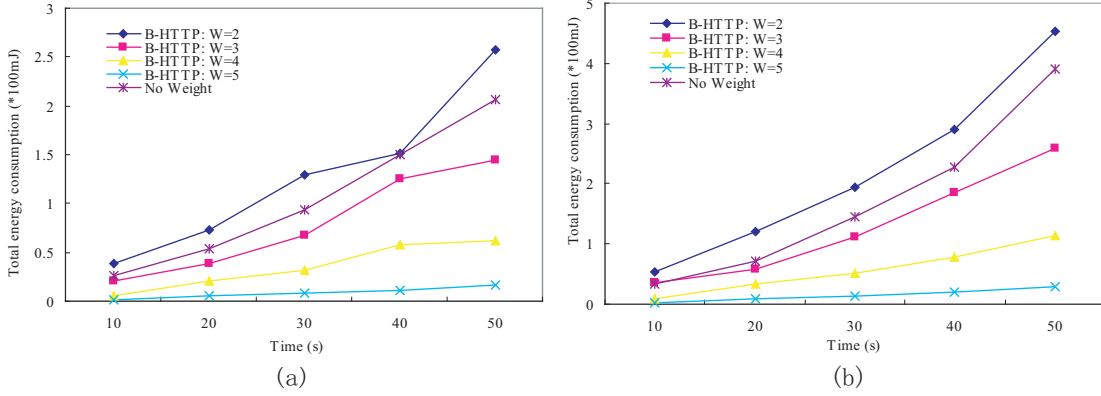
Fig. 5. Total energy consumption comparison of B-HTTP under different weights. (a) Sparse setting; (b) Dense setting

5.2 Simulation Results

Fig. 5 describes the total energy consumption of the B-HTTP protocol under different weights. The two figures clearly show that the weight affects the total energy consumption of sensor nodes. The total energy consumption decreases as the weight increases. It is due to a fact that the larger the weight is, the smaller the amount of the nodes participating in tracking is. We can also see that a large energy consumption gap exists under different weights.

In order to demonstrate the advantage of the proposed B-HTTP protocol on the total energy consumption, we compare the total energy consumption under the sparse setting (as shown in Fig. 5(a)) and the dense setting (as shown in Fig. 5(b)). From the two figures, we can see that the total energy consumption under the dense setting is significantly larger than that under the sparse setting. It is because there are more sensor nodes participating in target tracking. The total energy consumption of the nodes under the dense setting is approximately twice as much as that under the sparse setting. At the same time, we can see that the total energy consumption increases when the node weight is not considered. The reason is that only nodes whose weight is more than the threshold can participate in tracking. So by calculating the node's weight, the number of nodes that participate in tracking is decreased. When the node's weight is not considered, the total energy consumption is almost equal to that of the nodes when the threshold W=2.

Fig. 6 shows the total energy consumption of the E-HTTP protocol under different weights. The total energy consumption decreases as the weight increases. Under both the sparse setting (as shown in Fig. 6(a)) and the dense setting (as shown in Fig. 6(b)), we can see that the total energy consumption under the dense setting is larger than that under the sparse setting and the total energy consumption increases when the node weight is not considered. By Comparing B-HTTP (as shown in Fig. 5), the saved total energy consumption of E-HTTP protocol is more than 5%.

We compare the average energy consumption under the sparse setting (as shown in Fig. 7(a)) and the dense setting (as shown in Fig. 7(b)). The average energy consumption increases as the velocity of the target increases. When the velocity increases, there are more nodes joining the herd-based node group under
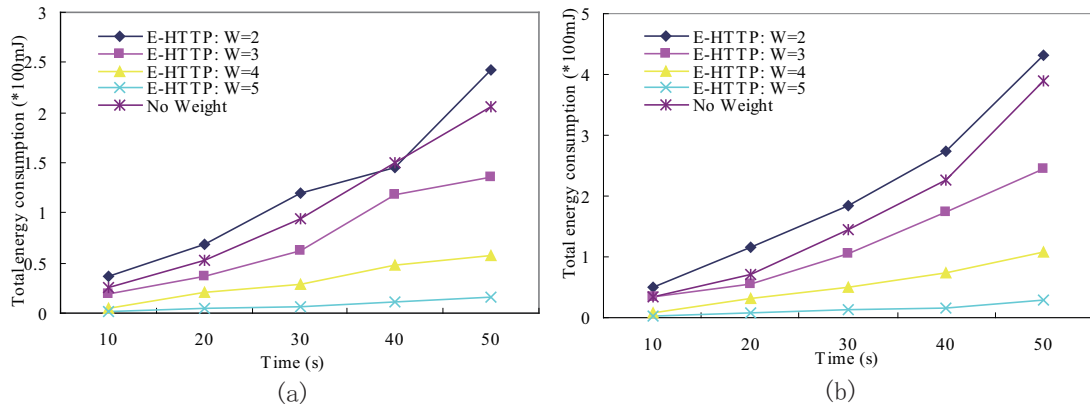
Fig. 6. Total energy consumption comparison of E-HTTP under different weights. (a) Sparse setting; (b) Dense setting
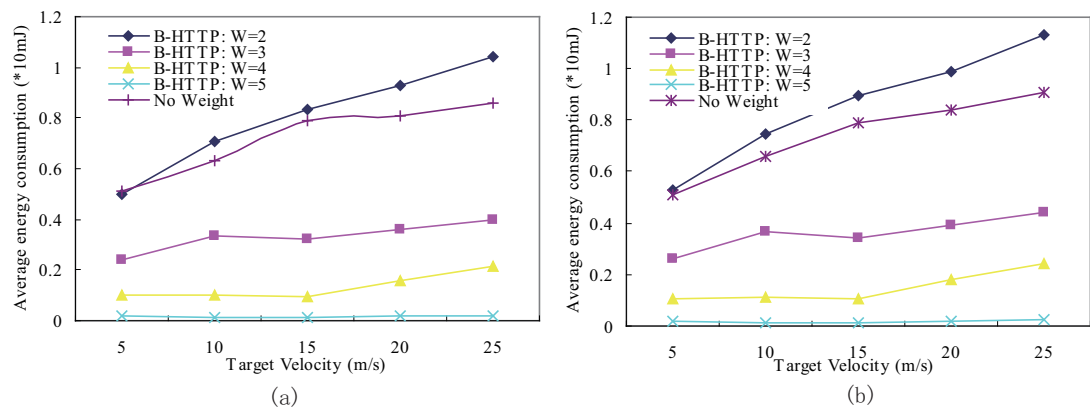

Fig. 7. The average energy consumption under different velocity of the target. (a) Sparse setting; (b) Dense setting

different weights, which consumes more energy. However, the amount of energy consumption is not so large. When the velocity of the target increases from $5m/s$ to $25m/s$, its range increases by 4 times, yet the average energy consumption only increases by 2 times. So the average energy consumption in the network increases with the velocity increment of the target, but the increasing rate is low. When the node's weight is not considered, we can see that the average energy consumption is only less than that of the nodes when the threshold $W=2$. Therefore, HTTP protocol has a distinct advantage in terms of energy consumption. When compared with the sparse setting, the average energy consumption under the dense setting increases slightly. Furthermore, we can see that the gap of the average energy consumption under different weights is relatively large. In summary, HTTP protocol shows an excellent performance on energy consumption.

6 Conclusion

In this paper, we proposed a protocol named HTTP, which can be used to detect and track a moving target in wireless sensor networks efficiently and effectively. The nodes in the tracking state can construct a herd-based node group to track the target dynamically. By setting the weight for each sensor node in the sensing and tracking state, the nodes can calculate their weight to determine whether they should join or leave the herd-based node group. Then the herd head adjusts the data reporting frequency according to the target's

situation. We believe this work opens a new area of research in the near future. The optimization of the tracking group and evaluation of data accuracy will be further investigated in our future work.

7 Acknowledgement

References

1. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey, Computer Networks, 38(4), (2002) 393-422
2. Culler, D., Hong, W.: Wireless Sensor Networks, Communications of the ACM, 47(6), (2004) 30-33
3. Xu, Y., Winter, J., Lee, W.: Prediction-Based Strategies for Energy Saving in Object Tracking Sensor Networks, In: Proceedings of the 2004 IEEE International Conference on Mobile Data Management, (2004) 346-357
4. Lee, W. C., Xu, Y.: On Localized Prediction for Power Efficient Object Tracking in Sensor Networks, In: Proceedings of the 23rd International Conference on Distributed Computers Systems Workshops, (2003) 434-439
5. Niyogi, K., Mehrotra, S., Venkatasubramanian, N., Yu, X.: Adaptive Target Tracking in Sensor Networks, In: Proceedings of the 2004 Communication Networks and Distributed Systems Modeling and Simulation Conference, (2004) 253-258
6. Wang, G., Bhuiyan, M., Zhang, L.: Two-Level Cooperative and Energy-Efficient Tracking Algorithm in Wireless Sensor Networks, To appear in Wiley's Concurrency and Computation: Practice and Experience, (2009)
7. Wang, G., Wang, H., Cao, J., Guo, M.: Energy-Efficient Dual Prediction-Based Data Gathering for Environmental Monitoring Applications, In: Proceedings of the 2007 IEEE Wireless Communications and Networking Conference(WCNC 2007), (2007) 3516-3521
8. Xu, Y., Heidemann, J., Estrin, D.: Geography Informed Energy Conservation for Ad Hoc Routing, In: Proceedings of ACM Mobile Computing and Networking, (2001) 70-84
9. Gui, C., Mohapatra, P.: Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks, In: Proceedings of the Annual International Conference on Mobile Computing and Networking, (2004) 129-143
10. Guo, M., Olule, E., Wang, G., Guo, S.: Designing Energy Efficient Target Tracking Protocol with Quality Monitoring in Wireless Sensor Networks, The Journal of Supercomputing (Springer) available online, DOI 10.1007/s11227-009-0278-5, (2009)
11. He, T., Vicaire, P. A., Yan, T., etc: Achieving Real-Time Target Tracking Using Wireless Sensor Nnetworks, In: Proceedings of the Twelfth IEEE Real-Time and Embedded Technology and Applications Symposium, (2006) 37-48
12. Lee, J., Cho, K., Lee, S.: Distributed and Energy-Efficient Target Localization and Tracking in Wireless Sensor Networks, Computer Communications, Vol.29, (2006) 2494-2505
13. Zhang, W., Cao, G.: DCTC: Dynamic Convey Tree-Based Collaboration for Target Tracking in Sensor Networks, IEEE Transactions on Wireless Communication, 11(5), (2004) 1689-1701
14. Zhang, W., Cao, G.: Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks, In: Proceedings of IEEE InfoCom, (2004) 434-2445
15. Xu, Y., Heidemann, J., Estrin, D.: Geography Informed Energy Conservation for Ad-hoc Routing, In: Proceedings of ACM Mobile Computing and Networking, (2001) 70-84
16. Sun, L., Li, J., Chen, Y., etc: Wireless sensor networks(in Chinese), Tsinghua University Press, Beijing (2005)
17. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy Efficient Communication Protocol for Wireless Microsensor Networks, In: Proceedings of the 33rd Hawaii International Conference on System Sciences, (2000), 3005-3014
18. Niculescu, D., Nath, B.: Ad Hoc Positioning System (APS), In: Proceedings of the 2001 IEEE Global Telecommunications Conference (GLOBECOM 2001), (2001) 2926-2931
19. Varga, A.: The OMNeT++ Discrete Event Simulation System, In: Proceedings of the European Simulation Multiconference (ESM 2001), (2001) 319-324