Target Trajectory Querying in Wireless Sensor Networks

Jin Zheng[§], Weijia Jia[§†*], Guojun Wang[§‡], and Jie Wu[‡]

[§]Schl. of Info. Sci. & Eng.
Central South University
Changsha, P. R. China, 410083
{zhengjin,csgjwang}@mail.csu.edu.cn

[†]Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
[*]wei.jia@cityu.edu.hk

[‡]Dept. of Comp. & Info. Sci.
Temple University
Philadelphia, PA 19122, USA
jiewu@temple.edu

***Abstract*—In this paper, we focus on the query problem of trajectory tracking of moving targets in a wireless sensor network. Our motivation is to design in-network storage and querying protocols to support queries for searching a given target signature in a low query latency, high query reliability, and energy efficient manner. We propose a target trajectory query protocol, which combines index-based query and random query schemes. The query protocol guarantees query reliability and energy efficiency. In an index-based queries scheme, each target trajectory has an index, which is stored at an index node (designated node). Query nodes send query messages to the index node to get results. Naturally, index-based queries can guarantee query reliability, but the disadvantage is the query hotspot problem, which reduces the lifetime of the sensor network. A random queries scheme, which is based on opportunistic target signature propagation, sends query messages in random chosen directions to hit a sensor node that contains the given target signature near the query node. However, queries for targets whose signature propagation is very limited need a large number of random query messages to achieve high query reliability. Our algorithm combines the two query schemes. It takes advantages of both of them, and eliminates each individual's disadvantages. Performance analysis and simulation studies show that our query protocol is superior to index-based queries in terms of the query hotspot problem, and it is superior to random queries in terms of energy efficiency, query latency, and query reliability.**

***Index Terms*—Target signature; linked list; opportunistic propagation; random query; query reliability**

I. Introduction

The development of wireless sensor networks has enabled long-term environmental monitoring. One of the most important areas where the advantages of sensor networks can be exploited is target tracking in an unobtrusive manner, e.g., rare animals in habitat monitoring, or vehicles/humans in security applications [1].

Most target tracking applications require reporting real-time data to a base station. However, some applications only need to query target trajectories or historical data in the monitoring field. For example, in a battlefield, soldiers need to query the trajectory information of an invader to judge the enemy's actions. When a soldier enters the monitoring battlefield, he wants to know the history locations of the invaders to make a motion analysis. The goal of these kind of tracking applications is to create an intelligent environment to support real-time sensing, situation understanding, and knowledge extraction for long-term monitoring of an environment.

The conventional approaches of reporting each individual tracking data to a base station, in this case, are not energy efficient because the communication cost can be high when the number of moving targets is large. In this paper, we focus on how to store and query target trajectories of moving targets in the network to support queries with low query latency and high query reliability in an energy efficient manner.

There are two kinds of target trajectory queries of moving targets [2], namely, existence query, and trajectory reporting query, respectively:

(a) Was there a blue truck entering the field in the past half hour?
(b) If yes, return its trajectory.

There have been a lot of prior works on tracking the moving targets by distributed sensor networks [3], [4], [5], [6], but most of these works focus on how to track the current location of the moving target, and they do not consider the target trajectory query for moving targets in a monitoring field. These works apply an external storage policy, which means sensors actively send all tracking data to an external device, i.e., a base station.

Data-centric storage in a distributed sensor network is a major architectural component and has attracted significant interest in recent years. In data-centric storage, the discovered data is preprocessed and stored in the distributed sensors. The data-centric storage category is fit for those applications, in which a large amount of data is produced, but only a small portion of the data may be queried.

There are two query methods regarding target trajectory, which use the data-centric storage policy to store and process data in a distributed sensor network, namely, IBQ (index-based query) [7] and RQ (random query) [2].

In our previous work, we proposed an index-based query scheme (IBQ) for target trajectory querying [7]. In IBQ, the target's moving data is stored in nodes near the detection sensor, and the relationship of storage nodes is maintained as a linked list along the moving path of the target. The sensor that detects the target first sends an index message, which includes the target ID, the sensor ID, and the timestamp of the target entering the monitoring field to an index node (designated node). Thus, query requests only need to be forwarded to the

index node to get the query results. But, because every query must be forwarded to an index node first, the index node easily becomes a query hotspot, which affects the network lifetime.

Zhou and Gao [2] proposed an opportunistic processing and query schemes, in which sensors exploit existing communication opportunities to propagate their respective signature of previously detected targets. There is no index node, and thus, no index for target trajectory. The query node initiates an RQ (random query) message, which follows a randomly chosen direction until either it visits a sensor node that contains information about the desired target, or it hits the sensor field boundary. If the random query message does not return the target signature, the query node repeats the random query process until it gets the given target signature, or the repeating number of sending random query messages exceeds a predefined threshold.

The random query schemes do not have the hotspot problem, but they can only give users a probability guarantee for successful querying, and they cannot keep the query reliability for users. Furthermore, they may repeat a large number of random query to hit a sensor, which contains target signatures, because the propagation of target signatures is limited. A large number of random query processes result in large energy consumption, so it is very energy intensive.

In this paper, we focus on the problem of storing, processing, and querying trajectories of moving targets in an energy efficient, low latency, and reliable way. We propose a storage and querying protocol of target trajectories, which is based on RQ (random query) and IBQ (index-based query) [2], [7]. The main idea of our storage protocol is: (1) to apply the local storage policy to store the tracking data near detection sensors, and maintain the relationship of sensors, which detect the same target to help the trajectory reporting query, (2) to apply the data-centric storage policy to store the index of each target trajectory, which only includes the target ID and the sensor ID (the sensor ID is the first sensor ID that detected the target), and (3) to apply the opportunistic propagation in the tracking process to make the target signature wider in the the network.

Corresponding to the storage policy, the main idea of our query strategy is that we use random query schemes first in hopes of hitting a sensor, which contains the given target signature near the query node. If the random query message does not return the given target signature, the query node sends query message to the index node to get the results. This query policy can avoid the problem of repeating a large number of random query processes, which consumes a large amount of energy of sensors. The proposed protocol can take advantages of the two query strategies, and eliminate the weaknesses of both of them dramatically.

The rest of this paper is organized as follows: Section II introduces the linked list of trajectory and opportunistic target signature propagation among linked lists. Section III describes the query algorithm combining random and index-based queries. Sections IV and V describe performance analysis and simulation studies. Finally, Section VI concludes this paper.

II. Linked List of Trajectory and Opportunistic Propagation of Target Signature

A. Linked List of Trajectory

A moving target moves arbitrarily in the monitoring field, and in some target tracking applications, users only need to get partial tracking data. In order to be energy efficient, target movement information should be stored near the detected sensor node instead of forwarding it to a central base station.

In our paper, a linked list is constructed to help users retrieve the historical tracking data, or get the target trajectory reporting.

Tracking based on dynamic clustering mechanisms can exploit in-network processing to improve the tracking precision, reduce the missing rate, and tracking data [8]. Sensor nodes in a cluster transmit the sensed data to the cluster head (CH), and then the cluster head records the digested data after data aggregation. When the target moves away, the previous CH will select a sensor node among its communication neighbors to be the next CH, which is the nearest to the target by prediction to continue tracking, and sends notification packets to the new CH. The previous CH will record the ID of the new CH, and the new CH can also record the ID of the previous CH. Using this method, the linked list of a target is formed along its trajectory (moving path) with nearly no additional overhead. The detection of any sensor on the linked list allows us to locally follow the linked list and get the entire movement information of a target, or the target trajectory reporting.

B. Opportunistic Propagation of Target Signature

The detection of any sensor in a linked list allows us to get signatures or to get trajectory reporting of a target. In fact, in many tracking applications, the targets being tracked come in the monitoring field at different times. When a new target moves into the field, some of sensor nodes near the target may have already stored signatures of other previous targets that have passed by. Thus, a sensor node should take existing communication opportunities in the tracking process to propagate signatures it has recorded so far. Naturally, as a target enters the field, if a CH hands over the knowledge it has learned to the next CH in the packet of new CH notifications, the targets signatures, which include target IDs and timestamps, can be propagated in the new linked list (trajectory). By using target signature propagation, a target signature is known not only to nodes along its linked list, but also possibly carried to other linked lists (trajectories) by targets that come in afterwards. The longer a target signature exists in the network, the wider it is propagated in the sensor network because more linked lists (trajectories) will propagate it. Each trajectory has an *age*, which is the number of targets that came in after itself before the query is initiated. The older a trajectory is, the easier it is to be discovered.

The idea of opportunistic propagation of a target signature is, in essence, similar to rumor routing [9]. It allows

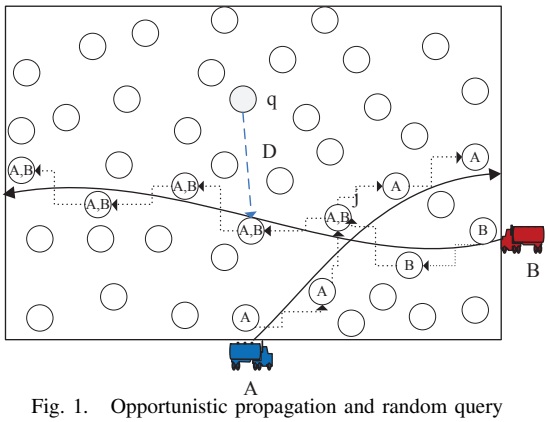

Fig. 1. Opportunistic propagation and random query

information to propagate wider in the network with nearly no additional communication to reduce random query cost.

Suppose there are two targets A and B in the monitoring field, as shown in Fig.1. B is passing by node j after target A, a node sends a random query message q for target A along a random direction R. Then, there are two possibilities for the query to discover the target A, either because q meets the node, which is in the linked list of A directly, or q meets the node, which is in the linked list of B.

For existence queries, the information propagated only includes the target ID and timestamp of the target entering the field. For target trajectory reporting queries, the information propagated should include the target ID, timestamp, and the sensor ID, which is the head of a linked list to help the query retrieve the tracking data. Because the target signature is propagated in the cluster head notification packet, it doesn't need other additional communication, so the propagation cost can be ignored compared with other communication costs throughout in the network.

C. Index of Target Trajectory

When a moving target is entering the monitoring field, the first sensor, which detected the target, sends a message called index, which includes the target ID, the sensor ID, and the timestamp to a predefined sensor node (called index node). The other sensors tracking the target will not send any messages to the index node in the future. The index message can help answer existing queries and to locate the target trajectory for trajectory reporting queries. The index node may be more powerful than other sensors, which may have larger storage space and energy.

III. QUERY ALGORITHM COMBINING RANDOM AND INDEX-BASD QUERIES

A. Query Algorithm Combining Random and Index-based Queries

The idea of random querying is similar in spirit with double rulings schemes [10]. In random query schemes, the query node initiates a query message. The query message follows a randomly chosen direction until either it visits a sensor node that contains information about the desired target, in which case the target signature is returned to the query node, or it hits the sensor field boundary, in which case the query message does not discover the signature of the given target. In order to keep the query reliability, the query node initiates another random query message and repeats the random query process until it gets signature of a desired target, or the repeat count exceeds a predefined threshold.

The success chance of a random query is greatly dependent on the target signature propagation level. If the target signature is propagated wider in the network, a random query has a high probability to hit a sensor that contains the target signature in the process of forwarding the query message along a direction. The simulation of reference [2] shows that in the opportunistic propagation schemes, when either the age or the length of a target trajectory is large enough, the random query number is close to 1. Otherwise, the repeat count of random query processes will be relatively large to discover the given target signature.

In our query protocol, we combine the random query and index-based query schemes to keep the energy efficiency, high query reliability, and low query latency. When a sensor node receives a query request about a target, for example: Was there a red truck entering the field in the past half an hour? This query $q(id, t)$ includes the target id and time restriction t. We approximate the target age a as time restriction t multiplied by the target entering rate u, i.e. $u.t$, or just think t as age a to simplify the implementation. It is reasonable because age a is linear with time t. If the time restriction is small, which means the number of targets entering the field afterwards is small too, thus the age of the desired target trajectory is young. If the age is less than a predefined threshold a_l, the query node initiates an index-based query message and sends it to the index node to get the query results of the existence queries. For trajectory reporting queries, if the index node has the record of the given target, the query message will be forwarded to the start node (head) of the linked list for the given target and get the data following that linked list. And if the age is larger than a_l, the query node initiates a random query message, which follows a randomly chosen direction, searching for a given target or target trajectory. If the random query message does not return the given target signature, it shows that the target's information is not propagated well, thus a large number of random query processes may be needed to keep the high query reliability. A large number of random query processes leads to large communication cost. So, for those queries of the 'young' target trajectory, index-based query schemes are more reasonable than random query schemes. The algorithm, called combining random and index-based queries (CRIQ), combines the random query and index-based query schemes to reduce the burden of the index node, thus the query hotspot problem is solved effectively, and CRIQ reduces query cost for 'young' target trajectories, and targets which do not enter the monitoring fields. It keeps energy efficiency and low query latency.

Algorithm 1 Query algorithm CRIQ (combining random and index-based query schemes)

Input:
id, the target ID of a query request
t, the time restriction of a query request
Output: yes or no

1: **if** $t > a_t$ or the distance from the query node to the index node is larger than a predefined threshold d_u, **then**
2: The query node initiates one random query message, and forwards it along a random chosen direction;
3: **if** the query message returns no successful results, **then**
4: The query node initiates an index-based query message, and forwards it to the index node;
5: **end if**
6: **else**
7: The query node initiates an index-based query message, and forwards it to the index node;
8: **end if**
9: The query node returns query results to the user.

TABLE I
NOTATIONS USED IN ANALYSIS

Parameter	Instruction
a	Age of target trajectory when query happens
n	Number of sensor nodes
r_1	Query rate for target trajectories not existing in a monitoring field
r_a	The rate for target trajectories query at age a ($a > 1$)
u	The rate of targets entering the monitoring field
C_{I-Q}	Cost of an index-based query
C_{I-U}	Cost of index updating
$C_{R-Q.S}(a)$	Cost of a successful random query for a target trajectory at age a
$C_{R-Q.F}$	Cost of a failure random query
n_{max}	The maximum random query messages of a query request

IV. PERFORMANCE ANALYSIS

This section analyzes the performance in message complexity (i.e., the total number of messages, except that of tracking and tracking data transfers in the network), query latency, and query reliability of CRIQ and RQ [2], and the query hotspot problem of CRIQ and IBQ. A summary of the notations used in the analysis is provided in Table1. For the sake of simplicity, data transmission is assumed to be error-free.

A. Message Complexity Analysis

It is easy to see that the message complexity of a target trajectory query using random queries depends on the age of that target trajectory (for existence query). The probability of a target trajectory with age a being discovered by a random query is $(1 - c\ln^2 a/a)$ (c is constant) [2]. Because most applications are more interested in recent target trajectories, we assume the oldest possible age of the target query is k. So, the total message complexity of our proposed protocol for target trajectory queries is given in Formula (1):

$$C_{CRIQ} = u.C_{I-U} + \sum_{a=2}^{k} r_a.((1 - c.\ln^2 a/a)C_{R-Q.S}(a) + c.\ln^2 a/a(C_{R-Q.F} + C_{I-Q})) + r_1(C_{R-Q.F} + C_{I-Q}) \quad (1)$$

Where u is the rate of targets entering the monitoring field, a is the trajectory age of a given target when the queries take place, r_1 is the query rate for target trajectories that are not existent in the monitoring field, r_a is the query rate for target trajectories whose age is a. The first term of Formula (1) is the index update cost. The second term is the query cost for targets whose answers are yes. And the last term is the query cost for targets whose answers are no (target trajectories do not exist in the monitoring field).

In random query schemes, a query node initiates a random query message, which is forwarded along a random chosen direction. If the query message does not return the signature of the given target, another random query message will be initiated. This process is repeated until the query node gets the given target signature, or the repeating number exceeds a predefined threshold n_{max}. So, the message complexity of RQ can be approximated as:

$$C_{RQ} = \sum_{a=2}^{k} r_a.((\frac{1}{1 - c.\ln^2 a/a)} - 1)C_{R-Q.F} + C_{R-Q.S}(a)) + r_1.n_{\max}.C_{R-Q.F} \quad (2)$$

For random trajectories and random queries, from the simulation of reference [2] and our simulation studies, n_{max} being set to 10 is reasonable for keeping random query reliability more than 90%. The first term of Formula (2) is the query cost for target trajectories, whose answers are yes. And the second term is the the query cost for target trajectories, whose answers are no (not existing in the monitoring field).

Assume that the sensor network consists of n uniformly distributed sensor nodes. As in references [11], [12], we use $\sqrt{n}$ to approximate the average cost of sending a message between two nodes in the network. It is easy to see that

$$C_{I-U} = \sqrt{n},\ C_{I-Q} = 2\sqrt{n} \quad (3)$$

For a random query of a target trajectory, whose age is *a*, the average distance to hit a node that has knowledge about that target is $d_{r-q.s} \leq \sqrt{n/3a}$ (Consider trajectories entering the field one by one after a trajectory *T*, and we will divide the process into epochs, such that after epoch 1, we have another trajectory T_1 that intersects T. The union of the trajectories T_1 and T is the dissemination network N_1 after epoch 1. Similarly, after epoch i, we have a trajectory T_i that intersects with only one trajectory of the current trajectory network N_i, and N_i is updated to include T_i as well. Thus, the monitoring field can be divided by p ($p \in [3a - 1, 3a + 1]$ planes at least). So, the average cost of successful queries in random schemes for targets that entered the monitoring field can be

approximated as:

$$C_{R-Q.S}(a) = 2d_{r-q.s} = 2\sqrt{n/3a} \tag{4}$$

But, for a random query that does not hit any node that contains the target signature, it will end up at the field's boundary and return no signature (unsuccessful) to the query node. The average cost for that query message is:

$$C_{R-Q.F} = 2\sqrt{n} \tag{5}$$

In our query protocol CRIQ, if a random query message does not return the signature of the given target, in the next step, the query node may initiate an index-node query message, and forward it to the index node. So, combining (1) through (5), we can rewrite (1) and (2) separately as:

$$\begin{aligned} C_{CRIQ} =&(u + 4r_1 + 2\sum_{a=2}^{k} 2r_a.c.\ln^2 a/a)\sqrt{n} \\ &+ 2\sum_{a=2}^{k} r_a.(1 - c.\ln^2 a/a).\sqrt{n/3a} \end{aligned} \tag{6}$$

$$\begin{aligned} C_{RQ} =&(20r_1 + 2\sum_{a=2}^{k} r_a(\frac{1}{1 - c.\ln^2 a/a} - 1))\sqrt{n} \\ &+ 2\sum_{a=2}^{k} r_a.\sqrt{n/3a} \end{aligned} \tag{7}$$

From our simulation studies and the simulation of reference [2], the constant c of Formulae (6) and (7) can be approximated as 2. We can see that the value of Formula (6) is less than the value of Formula (7) in the case that the ratio r/u (the ratio of query rate and target entering rate) is large enough. For a larger r_1, the value of Formula (7) is large than that of Formula (6). This means that CRIQ is more energy efficient than RQ when the query rate is larger compared with the new target entering rate, or there are large query requests whose answers are *no*. In fact, we are typically interested in querying recent trajectories, so CRIQ is more energy efficient than RQ in many applications, in which the query rate is large enough compared with the target entering rate.

B. Query Latency Analysis

CRIQ is expected to have better performance than RQ in the query latency, especially for 'young' age target trajectory queries. Because the query node initiates a random query message first (if the time restriction is larger than a predefined value), if the target signature is propagated wider in the network, the random query may get it near the query node with low query latency. If the query node gets no successful results in the first step, the query node will initiate an index-based query message, and forward it to the index node to get query results.

In RQ, the query node initiates one random query message, if the query message doesn't return any successful results, the query node initiates another random message, repeating this process until obtaining the given target signature, or the repeat count of random query processes exceeds a predefined threshold n_{max}. It is easy to see that the query latency of our proposed query protocol is not larger than that of RQ in general. Of course, the query node can also initiate n_{max} random query messages, and forward them along different directions in one step in hopes that at least one of them hits a node, which contains the knowledge of the given target. But, the query cost is too high because of the large number of random query messages.

C. Query Reliability Analysis

In our protocol CRIQ, a query node initiates a random query message in hopes that the query message hits a node, which contains the given target signature. If the query node gets no successful results, it sends an index-based query message to the index node to get the query results. But, for RQ, if the query node gets no successful result after repeating n_{max} times of random queries, it just means that the target does not exist in the network with probability $1 - \varepsilon$. Of course, we can increase the query reliability

by increasing the maximum query number n_{max}, but the message complexity will increase at the same time.

D. Query Hotspot Problem Analysis

CRIQ is expected to have better performance than IBQ in the query hotspot problem. Because the query node initiates a random query message first (if the time restriction is larger than a predefined value), if the target signature is propagated wider in the network, the random query may get it. Only if the query node gets no successful results in the random query process, the query node will initiate an index-based query message, and forward it to the index node to get query results. The forwarding load of the index node and its neighbours is reduced, so, the query hotspot problem is relived.

V. SIMULATION STUDIES

We conduct simulations to compare our storage and querying protocol against the random query schemes [2]. The goal of this study is to evaluate the important measure: the volume of messages incurred in the sensor network.

A. Simulation Setup

As similar to reference [2], in the simulation, we generate a network with sensor nodes uniformly and randomly distributed in a circular field with a radius of 25. We use a unit disk graph model with a communication range of 1. The number of sensor nodes is 5000, and the average degree of the sensor network is about 8. One hundred trajectories are randomly generated within the sensor field with the entrance and exit points taken randomly on the sensor field boundary. Random query messages are initiated at nodes randomly chosen inside the sensor field. The maximum query number n_{max} of a query request is 10.

For a particular query, the random query message follows a randomly chosen direction until either it hits the sensor node

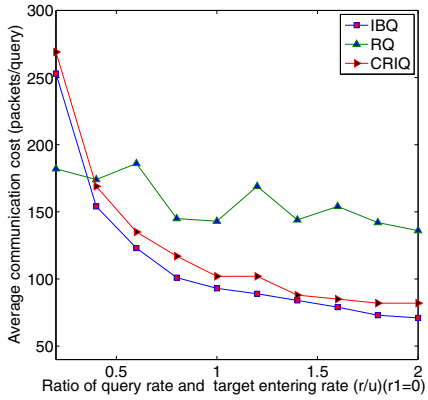

Fig. 2. Performance results on communication cost

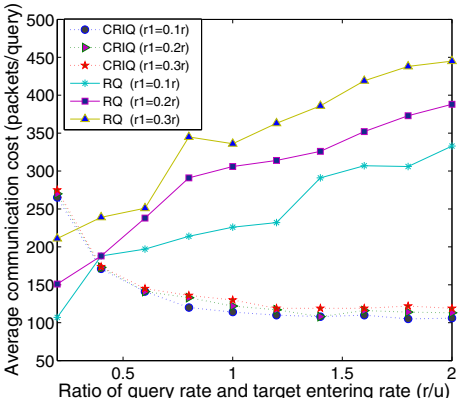

Fig. 3. Performance results on communication cost in different r_1

that contains the signature about the desired target, or it hits the sensor field boundary, in which case the random query is not successful, and another query message is generated. Routing of a query is done by greedy geographical routing.

B. Message Complexity

This section examines the message complexity of protocols CRIQ, RQ, and IBQ. Our simulation results show that CRIQ has better performance than RQ when r/u is large enough. Fig. 2 and Fig. 3 show the message complexity of CRIQ and RQ. As observed in Fig. 2, when there is no queries whose answers are no, we can see that the average communication cost of CRIQ is similar to that of IBQ (index-based query). And when the query rate is large enough compared with the target entering rate, the average communication cost of CRIQ is less than RQ. From Fig. 3, we can see that when there are target trajectory queries whose answers are no, there is no significant difference with the different ratios of CRIQ. But, for RQ, a higher ratio of query requests whose answers are no, more packets need to be forwarded, because the repeat count of random query processes is high for those queries of targets that do not enter the monitoring field.

VI. Conclusion

In this paper, we proposed a target trajectory query protocol, which combines the advantages of random and index-based query schemes. As sensors have scarce energy resources, the use of existing opportunities in the detection process itself is expected to be useful in a more general domain of data management in wireless sensor networks. The success rate of random querying greatly depends on the propagation level of the target signature. It needs a large number of query messages to keep the high query reliability for 'young' targets. The index of trajectories shows good performance with trajectory queries, but it has the query hotspot problem. The proposed query algorithm CRIQ alleviates the query hotspot problem. The average communication cost is lower than RQ when the query rate is large enough, especially when there are significantly more queries of 'young' target trajectories, or there are more target trajectory queries whose answers are no.

Acknowledgment

This work is supported by the Hunan Provincial Natural Science Foundation of China for Distinguished Young Scholars under Grant No. 07JJ1010, the Program for New Century Excellent Talents in University under Grant No. NCET-06-0686, and the Research Grants Council of the Hong Kong SAR, China Nos. CityU 114908, CityU 114609, and CityU R & D Center No. 9681001.

References

[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," *Proceedings of 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA-02)*, New York, NY, USA, pp. 88–97, 2002.

[2] D. Zhou and J. Gao, "Opportunistic processing and query of motion trajectoriesin wireless sensor networks," *Proceedings of INFOCOM 2009*, Rio de Janeiro, Brazil, April 2009.

[3] L. Guibas, "Sensing, tracking and reasoning with relations," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 73–85, 2007.

[4] W. Kim, K. Mechitov, J. Choi, and S. Ham, "On target tracking with binary proximity sensors," *Proceedings of 4th International Symposium on Information Processing in Sensor Networks (IPSN-05)*, Sunset Village, Los Angeles, CA, pp. 301–308, 2005.

[5] G. Wang, M. Z. A. Bhuiyan, and L. Zhang, "Two-level cooperative and energy-efficient tracking algorithm in wireless sensor networks," *Wiley's Concurrency and Computation: Practice & Experience*, DOI: 10.1002/cpe.1503, September 2009.

[6] Y. Xu and W. C. Lee, "DTTC: Delay-tolerant trajectory compression for object tracking sensor networks," *Proceedings of Sensor Networks, Biquitous, and Trustworthy Computing*, Taichung, Taiwan, pp. 436–445, June 2006.

[7] J. Zheng, W. Jia, and G. Wang, "Data management of mobile object tracking applications in wireless sensor networks," *Journal of Computers*, vol. 4, no. 9, pp. 845–852, 2009.

[8] W. Yang, Z. Fu, J. Kim, and M. S. Park, "An adaptive dynamic cluster-based protocol for targettracking in wireless sensor networks," *Proceedings of APWeb/WAIM 2007, LNCS 4505*, pp. 157–167, 2007.

[9] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," *Proceedings of 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA-02)*, Atlanta, Georgia, USA, pp. 22–31, September 2002.

[10] R. Sarkar, X. Zhu, and J. Gao, "Double rulings for information brokerage in sensor networks," *Proceedings of ACM MobiCom'06*, Los Angeles,CA,USA, , pp. 286–297, September 2006.

[11] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensornets with GHT, a geographic hash table," *ACM Mobile Networks and Applications*, vol. 8, no. 4, pp. 427–442, 2003.

[12] W. Zhang, G. Cao, and T. La Porta, "Data dissemination with ring-based index for wireless sensor networks," *Proceedings of International Conference on Network Protocols (ICNP)*, Atlanta, GA, pp. 305–314, november 2003.