A Prediction-based Energy-conserving Approximate Storage and Query Processing Schema in Object-Tracking Sensor Networks

Yi Xie, Weidong Xiao, Daquan Tang, Jiuyang Tang and Guoming Tang
Science and Technology Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
[e-mail: xieyi310@163.com]

Received March 2, 2011; revised April 25, 2011; accepted May 16, 2011; published May 31, 2011

Abstract

Energy efficiency is one of the most critical issues in the design of wireless sensor networks. In object-tracking sensor networks, the data storage and query processing should be energy-conserving by decreasing the message complexity. In this paper, a Prediction-based Energy-conserving Approximate StoragE schema (P-EASE) is proposed, which can reduce the query error of EASE by changing its approximate area and adopting predicting model without increasing the cost. In addition, focusing on reducing the unnecessary querying messages, P-EASE enables an optimal query algorithm to taking into consideration to query the proper storage node, i.e., the nearer storage node of the centric storage node and local storage node. The theoretical analysis illuminates the correctness and efficiency of the P-EASE. Simulation experiments are conducted under semi-random walk and random waypoint mobility. Compared to EASE, P-EASE performs better at the query error, message complexity, total energy consumption and hotspot energy consumption. Results have shown that P-EASE is more energy-conserving and has higher location precision than EASE.

Keywords: Data storage; data dissemination; location query; message complexity; object-tracking sensor networks.

1. Introduction

Wireless Sensor Networks (WSNs) which consists of small nodes with sensing, computing, and wireless communicating capabilities has become an important means of information's collection, storage and transmission [1]. A typical application of wireless sensor networks in military and many civilian areas is tracking moving objects, such as monitoring wildlife in remote areas and detecting the intrusion of illegal targets in important military areas [2]. As a result, wireless sensors are always deployed randomly or deliberately in some area to accomplish the task of monitoring, including event/target detection, localization and tracking with or without cooperation from the object itself [1][3], and according to some storage schema, queries are able to get real-time object location which is the way of tracking objects.

However, energy efficiency is a critical consideration in the design of large-scale sensor networks. Some significant researches have been carried out on energy-conserving object-tracking sensor networks (for example, [4], [5] and [6]). Most of these studies aimed at reducing the number of sensor nodes activated for tracking an object and/or reducing the location update traffic in providing accurate answers to location queries.

In the object-tracking application, the user just wants to know "where the object is" rather than "what the object's accurate coordinate is", which gives rise to an error tolerance of user's query, Inspired by this, the Energy-conserving Approximate StoragE (EASE) scheme is proposed in [7], which takes a different approach to improve energy efficiency by exploiting the trade-off between data quality and energy conservation. However, this scheme suffers from some location error caused by the difference between the object's real location and its approximate location, as well as some extra querying cost due to choosing improper querying destination. So EASE can be further optimized.

In this paper, we propose a Prediction-based Energy-conserving Approximate StoragE (P-EASE) whose approximate area is a sector with the centre angle being $2\pi / 3$ (1/3 circle) is introduced. EASE schema has been designed based on the assumption that the movement trace of an object is random. However, the mobility of the objects in many applications (e.g., the movement traces of an enemy soldier on instructional operation in a battle field or a firefight on rescue operation in the disaster area) can be determined in advance since they move in accordance with the decided operation or the drill regulation to perform their task. So the objects' movements are goal-driven, and a probable moving direction should be existed. We adopt a sector with the centre angle $2\pi / 3$ (1/3 circle) to be the approximate area and make the axis of the sector to be the direction of the object moving. In P-EASE, a predicting model is used, when a approximate query takes place, the predicting model calculates the probable location of object and return, not just return some static location simply (the static centre of the circle is returned as the querying result in EASE). We found that with prediction though the approximate area becomes smaller, the traffic overhead of the network does not increase ultimately, and the location precision is increase significantly.

It is known that the centric storage node is generated by some algorithm according to the object id, GHT e.g., and the object is moving arbitrarily in the field and the query request may come from everywhere. However, if the approximate query request is always sent to the centric storage node directly, it may bring in great message complexity. Hence, the optimal query schema is proposed to query the proper storage node, that is the nearer storage node of the centric and local storage node, and we illuminate the condition when the optimal query schema should be used to make sure to decrease the network load instead of bringing in extra messages resulted from notifications.

We summarize the contributions made in this study as follows:

- A Prediction-based Energy-conserving Approximate StoragE (P-EASE) is brought up, which improves the trade-off between the query precision and the energy conserving by using prediction method to return the approximate querying result dynamically to cut down the location error.
- The performance of the schema is analyzed theoretically and the optimal radius of the sector with the centre angle is $2\pi / 3$ (1/3 circle) is given under different situations.
- An adaptive radius adjusting algorithm is proposed for the unknown mobility of objects, which can adjust the radius of the approximate area in order to make the system work well.
- An optimal query schema is adopted to send the querying request send to the proper storage node instead of the centric node. Besides, theoretical analysis is given to determine the condition of using this schema. Thus, the unnecessary cost of query transmit to a far destination is eliminated.
- Extensive simulation experiments are given to validate the theoretical analysis.

The rest of the paper is organized as follows: Section 2 reviews related work firstly. Then the system model is described and the P-EASE scheme is proposed in detail in Section 3. Section 4 presents the performance of proposed P-EASE scheme by analyzing the optimal value of the radius of the approximate area and giving an adaptive radii adjusting algorithm. Section 5 proposes an optimal query schema to cut down the unnecessary transmitting messages and its occurring condition. Section 6 presents the results of simulation experiment. Finally, Section 7 concludes the paper.

2. Related Work

Improving the lifetime of an object-tracking sensor network is really one of the keystones in large-scale sensor networks. There are two research directions. One is to reduce energy consumption in sensor component. In [6] and [8], the sensor nodes sleep and wake-up periodically to make sure that only the object is within its radio range, it is waken-up to response for tracking, otherwise, it keeps sleeping for energy conserving. In [9], a low-cost preamble is used to wake nodes up for tracking. In [10][11][12][13][14], the sensor nodes are organized into a cluster-based architecture so that only the cluster head calculates object locations based on signal readings from its slave nodes.

The other direction is to improve energy efficiency by reducing network traffic in disseminating location updating. [4] proposed a prediction-based approach that a base station collects sensor readings and periodically generates predictions to send back to the sensor nodes. A sensor node reports a location update only when its reading differs from the predicted one. [15] suggested a dual-prediction scheme, namely, the predicting model in the base-station and the sensor node is the same, so that if the sensor nodes found the detecting locations is different from the predicted ones, the updating is occurred. [16] investigated continuous location queries and proposed a publish-and-subscribe tracking method. [17] proposed an efficient strategy for monitoring and tracking a single or multiple moving targets by reducing communication among the sensors for predicting the next location of moving targets. In [5] and [15], the trade-off between energy conservation and tracking quality has been investigated which inspired the [7] to adopted this method in object-tracking sensor networks.

To best of our knowledge, the EASE schema proposed by [7] is the first study on data dissemination in object-tracking sensor networks that attempts to address the issue of energy efficiency by exploiting the trade-off between data quality and energy conservation. However, the approximate method investigated in EASE can be improved at two aspects: First, combining with a predicting model, a sector approximate area with $2\pi / 3$ (1/3 circle) centre angle is used to cut down both the location error and the updating message complexity compared to EASE. Second, an optimal query schema is raised to eliminate the unnecessary querying messages resulted from choosing inopportune querying destination. Theoretical analyses and simulation experiments show that our schema is more precise and energy-conserving than EASE.

3. P-EASE: Prediction-based Energy-conserving Approximate StorgaE

3.1. System Model

A sensor network is formed by a large number of sensor nodes deployed in an area. Each node can obtain its own location by a GPS or other localization algorithms. The sensor network is organized into clusters, and every cluster has a more powerful cluster head which is equipped with local storage to store data and is also capable of communicating with other cluster heads to exchange data. For simplicity, in this paper a heterogeneous sensor network is constructed based on voronoi diagram according to [10]. However, our approach is also fit for other clustering strategies, e.g. the clustering strategies in [12][14][19][20][21] [22].Sensor nodes in a cluster can identify and track the object cooperatively with some localization algorithm, e.g. [10][13][19]. We also consider the sampling rate of sensor node is ultimately fixed and any object has a unique identifier. An object's location can be obtained by a real-time location query. Hence, continuous location query can track the object effectively.

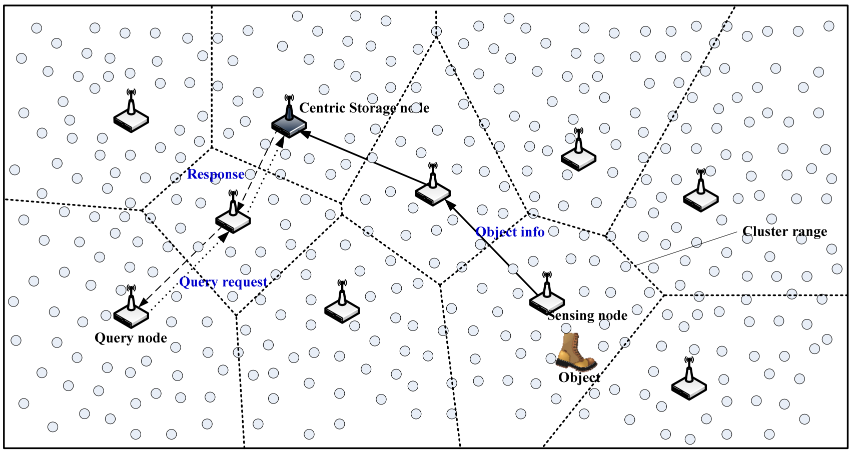


Fig. 1. Overview of the system model

Construct the clusters. There are two kinds of nodes in our sensor network: (a) a static backbone of high-capability sensors which will assume to be *cluster head* with more power and more transmission range; (b) proper densely populated low-end *member sensor nodes* whose function is to provide sensing information to cluster heads. Thus, let each cluster head to be a site of voronoi diagram, the field can be divided into several voronoi cells, sensors in each voronoi cell form a cluster [10]. Focusing on reducing communication among cluster heads, unless explicitly specified, a sensor node refers to a cluster head in the rest of this paper [7]. (As shown in Fig. 1)

Approximate location query. The location queries from users can be made by a sensor node from anywhere in the network. We assume that each query can be approximate and can tolerate an error *p* to an object with an identified *object id*, which is specified by a tuple <*object id; p*>.

Object Prediction. The object information contains the previous position (x_p, y_p) , the moving direction φ and the velocity v of the object which are used to predict its current location by:

$$\begin{cases} x_c = x_p + v \cdot \Delta t \cdot \cos(\varphi) \\ y_c = y_p + v \cdot \Delta t \cdot \sin(\varphi) \end{cases} \tag{1}$$

So after Δt , the current location of the object (x_c, y_c) can be predicted by the centric node or local storage node [2].

Hybrid Storage. Local storage (LS) and data centric storage (DCS) are both used to store the information of object in different situation as the EASE has shown to us, which can achieve a good balance between the querying cost and the updating cost for approximate location queries.

3.2. P-EASE Schema

1) Approxmate area of P-EASE:

In EASE, the CIRCLE is the given approximate area, let *r* be the radius of the CIRCLE which also stands for the max error tolerance of the users. At the next sampling time instant, if the object is still in the circle, the centre of the circle which is the previous location of the object is returned as the approximate location.

So Fig. 2-(a) shows that the distance from the object to its approximate location (the centre of the circle in EASE) distributes in [0, *r*].

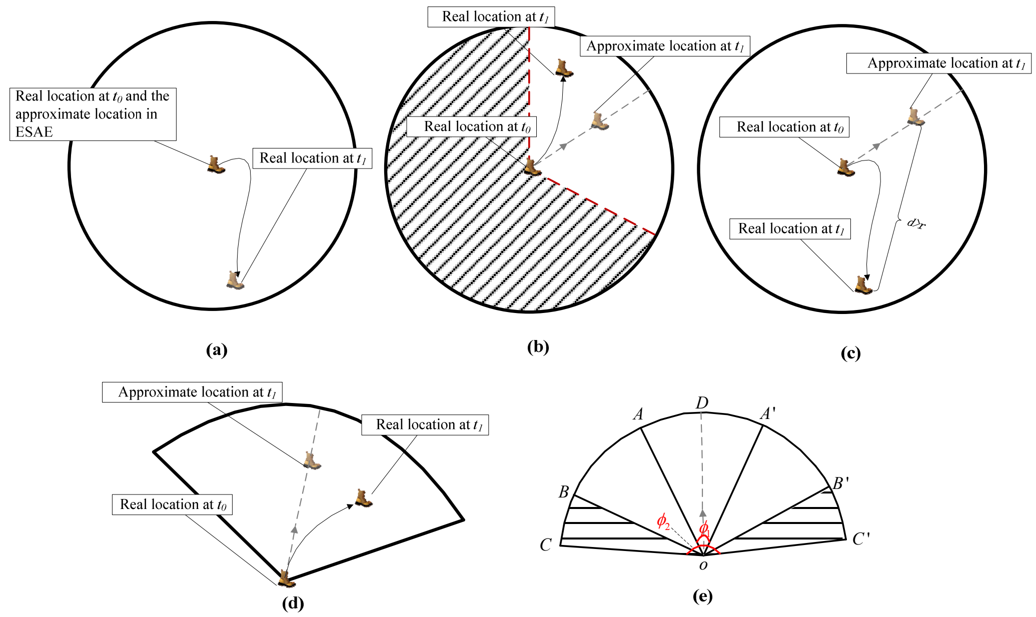


Fig. 2. Selecting Approximate Area

It is known that the EASE schema proposed a tradeoff between the querying precision and the updating message complexity, in order to achieve an energe-efficent storage and query processing schema. As we discussed the **Object Prediction** in Section 3.1, it is obvious that if the location of the object is known as well as the velocity and its moving direction, the location of this object at next time instant can be predicted by (1). Thus, we consider using object prediction to complement the error brought by the approximate storage and querying. However, if directly adding predicting model to the CIRCLE-like approximte area, there comes two problems:

Problem 1. **Some of the approximate area is useless.** Since we use predicion to get the approximate location, the returning location will be located on the radius of object's moving direction. Besides, its real location is propable in some area near the direction axis, as shown in **Fig. 2-(b)**. On one hand, other area of the CIRCLE will be useless. Thus, the redundant area should be inavailable. On the other hand, the object appear in other area will also lead the location error to be great, there comes ***Problem 2***.

Problem 2. **The location error may be greater than tolerance.** We assume the distance from the real location of the object to the returning location is *d*. As shown in **Fig. 2-(c)**, if at some time instant the object turns around and moves in an opposite direction, it will be at some location far away from the predicting location. When the distance is longer than the radius which is considerd to be the upper bound of error ($d>r$), the location error becomes greater than we setup. Hence, appdending prediction to the CIRCLE-like approximte area directly may cause even worse results and greater error than without appending it.

Since CIRCLE-like approximte area is not fit for object prediction, another approximate area is needed. Thus, we summarize 3 principles of choosing a proper approximate area to conquer both problems.

Principle 1: The new approximate area should be ***an axial symmetric graph instead of a central symmetric graph***, the axis is the object's moving direction and the points on its axis should be the approximate locations.

Principle 2: We should guarantee that ***returning location is no more distant than r to the object's real location***, as this will bring in some approximate points beyond the user's error tolerance.

Principle 3: The approximate area should be set ***as large as possible for a given r*** in order to avoid frequent remote updatings resulted from the tendency that object is continuously moving out of the area.

After appending prediction, the returning location will not be one fixed point any more, but some point on the radial of the object's moving direction. The ***Principle 1*** implicates us that the central symmetric graph, e.g.CIRCLE, should be changed into an axial symmetric one with its symmetric axis the same as the object's moving direction. Thus, the returning location can be any point on the symmetric axis, while the real location of the object is some point in the approximate area. Besides, in order to facilitate the tolerance of error with radius, adopting a SECTOR to be the approximate area is conceivable, as shown in **Fig. 2-(d)**. Thus, for the SECTOR-like approximate area, the points on its axis, which is the moving direction of objects, will be the predicted location returned to the querying node.

The centre angle can be determined according to ***Principle 2*** and ***Principle 3***.

Firstly, as shown in **Fig. 2-(e),** supposing $\angle AoA' = \phi_1$ and $\angle CoC' = \phi_2$, $\angle BoB' = 2\pi/3$, it is obvious that $\phi_1 < 2\pi/3 < \phi_2$. When the central angle of the sector is larger than $2\pi/3$, e.g. ϕ_2, some point in the sector BoC and the sector $B'oC'$ will be more distant than *r* to some point on the axis, which is against the ***Principle 2.*** So only a SECTOR whose centre angle is less than $2\pi/3$ can ensure that the max location error is less than *r*.

Secondly, the central angle of the sector is less than $2\pi/3$,e.g. ϕ_1, is not the optimal either. Because the ***Principle 3*** shows that the approximate area is expected to be as large as possible to limit occurrence of the remote update, the centre angle of the SECTOR is set to be the upper bound $[0, 2\pi/3]$,and the predicted location located on the axis, which make the error stays distribute in [0, *r*].

As we see that the max value of location error of P-EASE is equal to EASE. But the average error should be the more important to the location precision. In addition, since the area of SECTOR is only 1/3 of the CIRCLE, it is really another problem whether the remote updating of object information would become more frequent. The two problems will be discussed in Section 4.

2) Location query and update protocol.

There are four kinds of nodes in P-EASE schema:

Detecting node: the nodes that detect objects

Local storage node*:* the nodes that first detect objects and store the object info locally.

Centric storage node*:* the nodes that store approximate object info according to DCS schema (e.g., Geographic Hash Table, GHT [23]).

Query node: the nodes that send location query requests.

As mentioned before, the detected object information is stored in two places with two versions: an accurate version at a local storage node and an approximate version at the centric storage node. Actually, P-EASE is a tradeoff between the location precision and the energy cost. We can divide the P-EASE protocol into two parts:

A. Location querying.

If a node wants to query the location of an object, it will send an approximate querying request <*object id; p*> to the centric storage node according to the DCS schema. When the centric storage node receives the request, it compares the user's error tolerance *p* to the radius of approximate area *r* to decide how to response the query.

If the stored object info in the centric storage node satisfies the precision requirement ($p \geq r$) which is colored blue in **Fig. 3**, there are two querying steps:

1. The **querying node** sends an approximate querying request to the centric storage with $p \geq r$.
2. The predicted location is computed by the predicting model according to the approximate object information stored in the **centric storage node** itself, and returns the approximate result to the querying node immediately.

Else if $p < r$ which is colored red in **Fig. 3**, three steps are needed:

1. The **querying node** sends an approximate querying request to the centric storage with $p < r$.
2. The **centric storage node** should forward the querying to the local storage node which has the accurate info of the object.
3. When the **local storage node** gets the request, it will predict the object location according to the accurate object info and returns the accurate results to the querying node directly.

However, if $p \geq r$ and the predicted location is beyond the approximate area, the returned location will be the midpoint of the arc, which should be the most distant predicted location (as shown in **Fig.4**).

B. Object information updating.

Once a node has detected the object and does not have the object information notified by geocast [7], it would be the local storage node and would send a geocast notification to the nodes in the approximation area.

At each subsequent sampling instance, if the detecting node in the current approximation area (that has be notified by geocast) detects the object, it only sends a *local update* including the new object location, moving direction and velocity to the local storage node(as shown in **Fig.5**).

Otherwise, the detecting node which is out of the approximate area elects itself to be the new local storage node. If the new and old local storage node can communicate with each other directly (as shown in **Fig.6**), the procedure of object information updating is as follow:

1. The **new local storage node** sends a *remote update* to the centric storage node.
2. The **new local storage node** sends an *invalidate message* to the old local storage node directly;
3. The **new local storage node** sends a *geocast* to the new approximate area about the object info;
4. When the **old local storage node** has received the invalidation message sent from the new local storage node, it sends an *invalidation geocast* to the old approximate area.

However, if the new local storage node cannot communicate with the old one directly, it has to send the invalidating message to the centric storage node firstly which knows the old local storage node. The old local storage node can only receive the invalidation message forwarded by the centric storage node (as shown in **Fig.7**). Hence, the procedure contains one more step:

1. The **new local storage node** sends a *remote update* to the centric storage node.
2. The **new local storage node** sends an *invalidate message* to the centric storage node;
3. The **centric storage node** forwards the *invalidate message* to the old local storage node;
4. The **new local storage node** sends a *geocast* to the new approximate area about the object info;
5. When the **old local storage node** has received the Invalidation message sent from the new local storage node, it sends an *invalidation geocast* to the old approximate area.

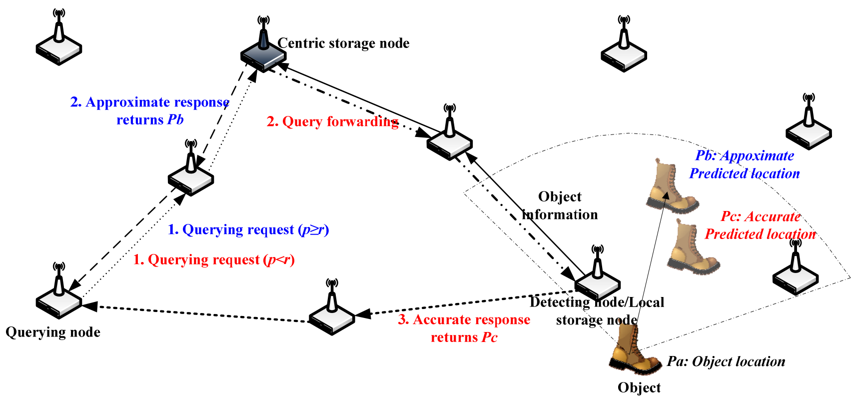


Fig. 3. Location querying (a)

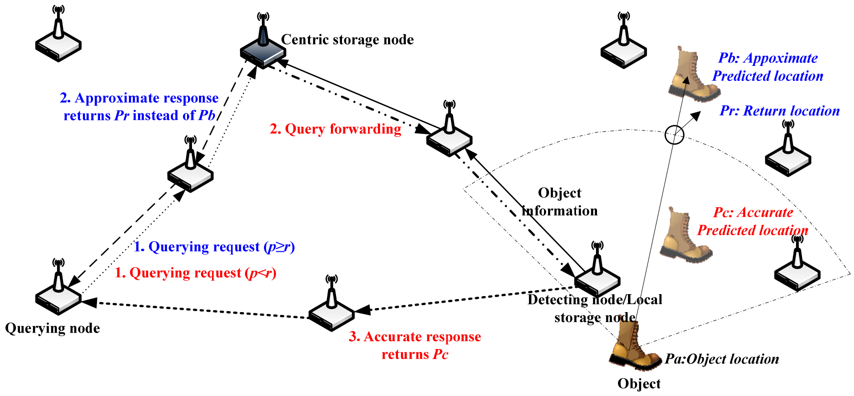


Fig. 4. Location querying (b)

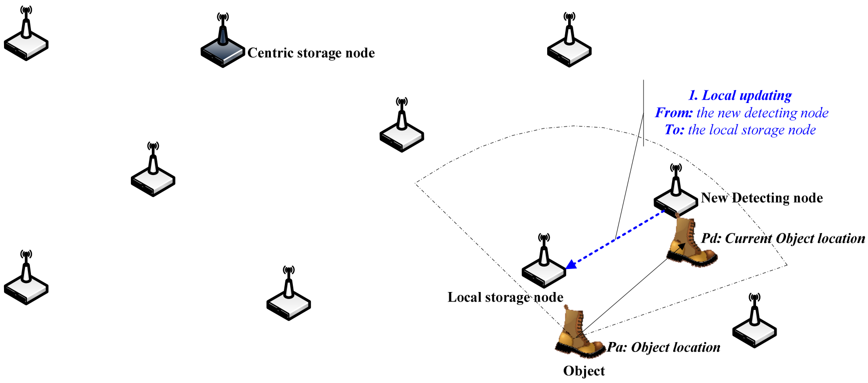


Fig. 5. Local update protocol

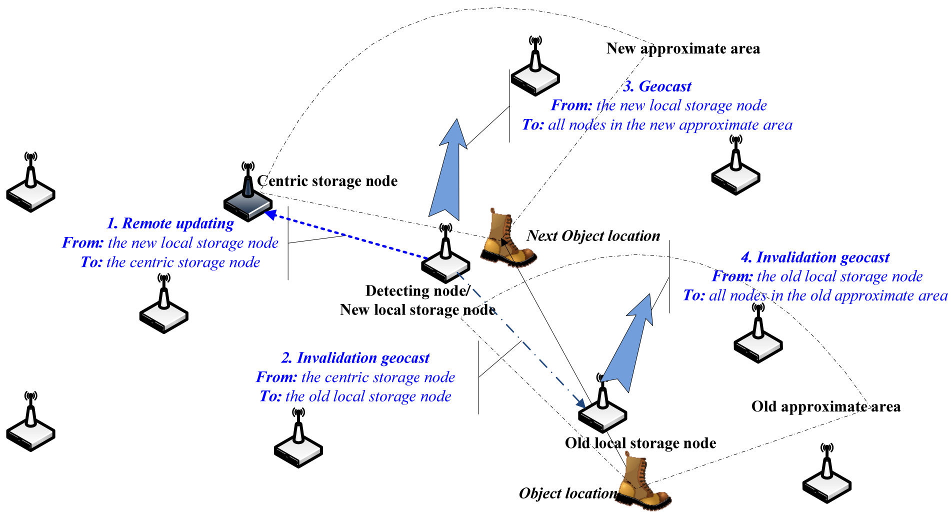


Fig. 6. Remote update protocol(a)

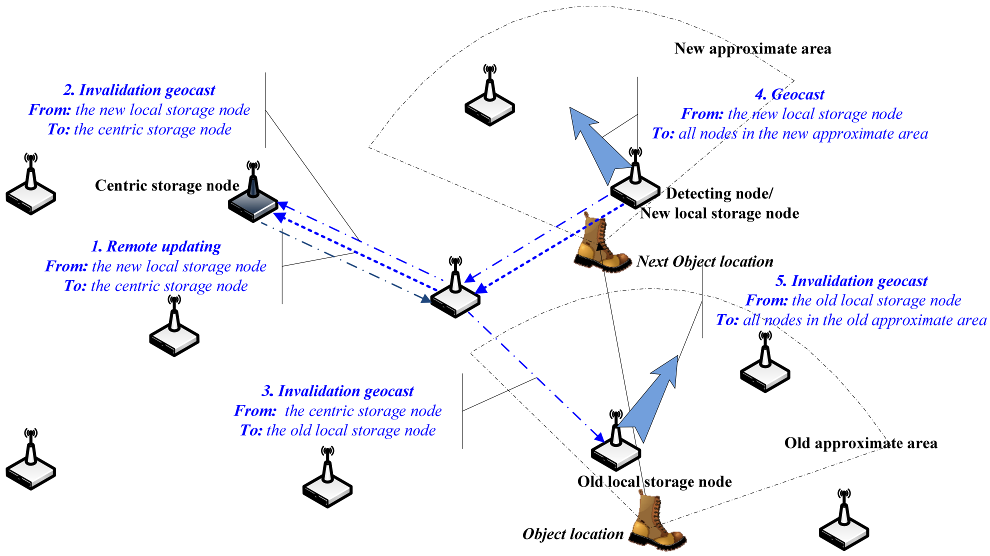


Fig. 7. Remote update protocol(b)

3) Algorithm.

The centric storage nodes, the local storage nodes and the detecting nodes have their respective process protocols which are given as follow:

Algorithm 1: Protocol executed at the centric storage node

if receiving a query $.< object\ id; p >$ **then**

  if $p \geq r$ then

    set $now\_x \leftarrow target\_x + v \cdot \Delta t \cdot \cos(direct)$;

    set $now\_y \leftarrow target\_y + v \cdot \Delta t \cdot \sin(direct)$;

    if $v \cdot \Delta t > R$

      set $now\_x \leftarrow target\_x + R \cdot \cos(direct)$;

      set $now\_y \leftarrow target\_y + R \cdot \sin(direct)$;

    end if;

    return the stored location to the querying node;

  end if;

end if;

if receiving a remote location update message **then**

  if the old approximation area is unknown at the local storage node **then**

    send an invalidation geocast message to the old approximation area;

  end if;

  store the new location and local storage node of the object;

end if;

Algorithm 2: Protocol executed at the local storage node

if receiving a forwarded query $< object\ id; p >$ then

  set $now\_x \leftarrow target\_x\_new + v\_new \cdot \Delta t \cdot \cos(direct\_new)$;

  set $now\_y \leftarrow target\_y\_new + v\_new \cdot \Delta t \cdot \sin(direct\_new)$;

  return the location (now_x, now_y) by forecast to the querying node;

end if;

if receiving a local location update message then

  store the new location of the object;

8:**end if** ;
9: **if** receiving a invalid geocast message then
10: send an invalidation geocast message to approximation area A ;
11: **end if;**

Algorithm 3: Protocol executed at the detecting node
1: **if** object o is in the valid approximation area and the local storage node is known then
2: send a target info update $T =< target\_x, target\_y, v, direct >$ to the local storage node;
3: **else**
4: send a target info update $T =< target\_x, target\_y, v, direct >$ to the centric storage node and fetch the information of the previous local storage node;
5: elect itself as a new local storage node;
6: set the new approximation area A' a sector centered at o with central angle $\theta = 2\pi / 3$ and radius R and whose axis of symmetry is the line of direct.
7: **if** the previous local storage node is known then
8: send an invalidation geocast message to the previous local storage node;
9: **end if**;
10: send a notification geocast message to A';
11: **end if**;

4. Performance Discussion

Since P-EASE is also a tradeoff schema between the location precision and the energy cost like EASE, this section will first discuss the message complexity and location error of the P-EASE schema and then compare them to EASE. The notations used in the discussion are provided in **Table 1**.

Table 1. Notation used in analysis

Notation	*Description*	*Notation*	*Description*
n	number of sensor nodes	r	radius of the approximate sector
λ	query rate	$p_{\max}$	max of user error tolerance
μ	sampling rate	$C(r)$	overall message complexity
C_{qn}	cost of answer a query by centric storage node.	$p_{qf}(r)$	probability of forwarding query
C_{qf}	cost of answer a query by local storage node.	$\psi(r)$	remote update rate
C_{ur}	cost of remote updating	d	step distance of random walk
C_{ul}	cost of local updating	l	time of each step in random walk
f	density of sensor nodes		

4.1. Message Complexity

We assume that the number of nodes is n, the cost of answering a query by centric storage node is C_{qn}, the cost of answer a query by local storage node is C_{qf}, the cost of remote updating is C_{ur} and the cost of local update is C_{ul}, then it is easy to get the overall message complexity of the network:

$$C(r) = (1 - p_{qf}(r)) \cdot \lambda \cdot C_{qn} + p_{qf}(r) \cdot \lambda \cdot C_{qf} + \psi(r) \cdot C_{ur} + \mu \cdot C_{ul} \qquad (2)$$

As proved in [8] and [19], the cost of flooding is approximate n and the cost of sending message between two nodes is $\sqrt{n}$, so we can get . $C_{qn} = 2\sqrt{n}$ and $C_{qf} = 3\sqrt{n}$. Supposing the max level of user's error tolerance is $p_{\max}$, the probability of forwarding query which means that the approximate querying is not satisfied the user's error tolerance we obtain: $p_{qf}(r) = \min\{r / p_{\max}, 1\}$

The cost of a remote updating consists of three parts:

1. The cost of invalidating the old local storage node, which is given by $\sqrt{n}$;
2. The cost that the old local storage node sends invalidating geocast in the old approximate area. Since the cost of flooding is approximate n, the cost can be estimated by the number of nodes in the approximate area which is $f \cdot \pi r^2 / 3$;
3. The cost of geocast to the new approximate area for notification which is $f \cdot \pi r^2 / 3$ as well;

Thus, the cost of a remote updating is $C_{ur} = \sqrt{n} + 2\pi r^2 f / 3$.

As a local updating is that just detecting node sends object information to the local storage node, the cost can be evaluated by the average number of sensor nodes encountered when message traveling. The traveling distance is given by Appendix I which is $0.4255r$, so the cost of the local updating is $C_{ul} = 0.4255\mu r\sqrt{f}$.

Combining all of these, the overall message complexity of the network becomes:

$$C(r) = \lambda \cdot 2\sqrt{n} + \lambda \cdot \sqrt{n} \cdot \min\{r / p_{\max}, 1\} + \psi(r) \cdot (\sqrt{n} + 2\pi r^2 f / 3) + 0.4255\mu r\sqrt{f} \tag{3}$$

Compare to the overall cost of EASE [7] that is:

$$C'(r) = \lambda \cdot 2\sqrt{n} + \lambda \cdot \sqrt{n} \cdot \min\{r / p_{\max}, 1\} + \eta(r) \cdot (\sqrt{n} + 2\pi r^2 f) + 2\mu r\sqrt{f} / 3 \tag{4}$$

We can obtain the difference by:

$$C(r) - C'(r) = \psi(r) \cdot (\sqrt{n} + 2\pi r^2 f / 3) - \eta(r) \cdot (\sqrt{n} + 2\pi r^2 f) - 0.2411\mu r\sqrt{f} \tag{5}$$

$\eta(r)$ is the remote updating rate of EASE schema and $\psi(r)$ is that of P-EASE. To be simple, we suppose the object's moving keeps to a two-dimensional random walk model, then the $\eta(r) = d^2 / lr^2$ in EASE [7]. However, in P-EASE the reason why the object moves out of the approximate is twofold: the one is the ***displacement*** from the approximate location stored by the centric storage node to the current location detected by the detecting node is greater than the radius of the sector-shaped approximate area; the other is the line passing the current location and the approximate location makes an ***angle*** which is larger than $\pi / 3$ with the axis of the sector. (As shown in **Fig. 8**)

According to Appendix II, if the object moves out of the approximate area because of displacement, the average time is given by $T_1(r) = r^2 \cdot l / d^2$, which is the same as that of EASE. Else if it is because of the angle, the average time is $T_2(r) = 7r \cdot l / 3d$. Hence, we can easily get the remote updating rate, that is $\psi(r) = 1 / \min(T_1, T_2)$.Therefore, when $T_1(r) \geq T_2(r)$, namely $r / d \geq 7/3$, the object is considered to be moving slowly, then the remote updating rate is $\psi(r) = 1 / T_1(r)$, otherwise, the object is moving fast which leads the remote updating rate to be $\psi(r) = 1 / T_2(r)$. We can get:

$$\psi(r) = \begin{cases} 3d / 7rl & r / d \geq 7/3 \\ d^2 / r^2 l & r / d < 7/3 \end{cases} \tag{6}$$

It is obvious that for the random walk model, when $r / d < 7/3$, the overall cost of P-EASE is less than EASE, while $r / d \geq 7/3$, the cost of P-EASE will change by:

$$C(r) - C'(r) = \frac{(3r\sqrt{n} + 2\pi r^3 f) \cdot d - (7\sqrt{n} + 14\pi r^2 f) \cdot d^2}{7r^2 \cdot l} \tag{7}$$

Choosing some network parameter, we can get the difference which is a function like:

$$f(d,l) = (B \cdot d - A \cdot d^2) / (C \cdot l) \qquad A, B, C > 0$$

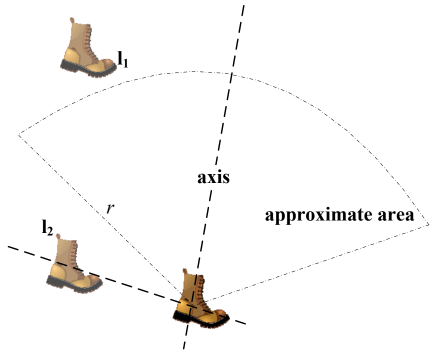


Fig. 8. Two situation that object moves out of approximate area

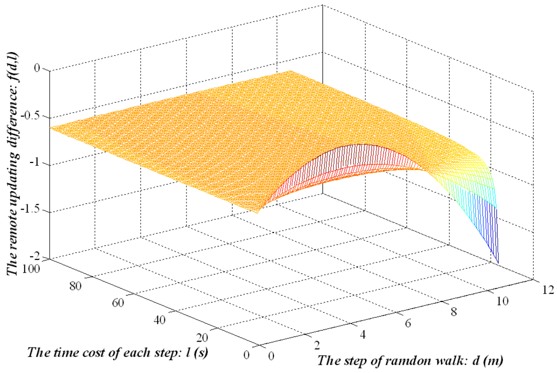


Fig. 9. Ploting the Function $\boldsymbol{f(d,l)}$

Equation (5) shows that when the $r/d < 7/3$, the value of the difference between EASE and P-EASE is a constant $-0.2411\mu r\sqrt{f}$. However, when $r/d \geq 7/3$, function $f(d,l)$ is shown by **Fig. 9** which stands for the difference of cost between EASE and P-EASE. For a given d/l which can be considered the average velocity of the object, there are two situations. When d and l are both great, the object's stepping distance is long and the taken time is long as well, so the probability of moving out of the area is also ultimately due to the displacement, hence, the difference is a small constant. In contrast, the probability of moving out of the area resulting from the angle is dominating, thus, the difference can be a little greater but has a maximum. This demonstrates, compared to EASE, the remote updating of P-EASE does not become more frequent.

So we summarize that when the object is moving fast, the cost of P-EASE is less than EASE; when the object is moving slowly and stepping long distance, the cost is of P-EASE is much less than EASE; and when the object is moving slowly and stepping short distance, the cost of P-EASE is no more than EASE which is dependent on network parameter. The problem will be detailed in our following simulation.

4.2. Optimal Approximation Setting with Known Mobility

As (3) and (6) show, the radius of the sector is a key factor of the tradeoff between the location error and the remote updating rate. Plugging (6) into (3), and making: $\frac{\partial C(r)}{\partial r}=0$, then we can obtain the optimal settings of r_l^* (when $r/d \geq 7/3$) and r_h^* (when $r/d < 7/3$) both in two cases:

$$\text{when } r/d \geq 7/3, \quad r_l^* = \begin{cases} \sqrt{\dfrac{3d\sqrt{n}}{2\pi df + 2.9785\mu l\sqrt{f}}} & \text{when } r \leq p_{\max} \\ \sqrt{\dfrac{3p_{\max}d\sqrt{n}}{7\lambda l\sqrt{n} + 2.9785p_{\max}\mu l\sqrt{f} + 2p_{\max}\pi df}} & \text{when } r > p_{\max} \end{cases} \tag{8}$$

$$\text{when } r/d < 7/3, \quad r_h^* = \begin{cases} \sqrt[3]{\dfrac{2d^2\sqrt{n}}{0.4255\mu l\sqrt{f}}} & \text{when } r \leq p_{\max} \\ \sqrt[3]{\dfrac{2p_{\max}d^2\sqrt{n}}{l\cdot(\lambda\sqrt{n} + 0.4255p_{\max}\mu\sqrt{f})}} & \text{when } r > p_{\max} \end{cases} \tag{9}$$

We can observe that the optimal setting of r^* is affected by many factors such as the network size, query rate, sensor sampling rate, mobility pattern, and precision requirement. it is easy to see that when $r/d < 7/3$ (the object moving fast), the optimal setting of the radius is the same as the EASE schema because the remote update is mainly caused by displacement of the object. However, for a slower movement of the object, the remote updating is ultimately caused by the angle between the object and the centre of the sector, so the optimal setting of P-EASE is different from the EASE schema now. In a real system, this result of optimal setting of radius in P-EASE shown in (8) and (9) can be applicable to other mobility models as long as $\psi(r)$ can be estimated.

4.3. Adaptive Approximation Setting with Unknown Mobility

It is known that in real case, the mobility of objects always cannot be described by mobility models; hence the P-EASE calls for a method that can determine the optimal radius of the sector (approximate area) dynamically. We observe that the value of r is given at the initialization of the network, and for the dynamic mobility of object, the r should be adjusted continuously in order to guarantee the optimization of the system. When $r > p_{\max}$, the storage schema is the same as the local storage schema, thus the query request should be forwarding to the local storage node to get the object info, which means $p_{qf}(r)=1$. Then when $r \le p_{\max}$, the probability of not forwarding query (the centric storage node returns the results directly) is

$$p_{qf}(r) = \frac{r}{p_{\max}}$$

When $r = r^*$, and we denote $\rho_1 = \lambda\sqrt{n} + 0.4255 p_{\max}\mu\sqrt{f}$, $\rho_2 = \frac{2}{7} p_{\max}\pi f$ and ρ_1, ρ_2 are constants. It is not difficult to observe that:

$$\frac{p_{qf}(r)}{\psi(r)} = \begin{cases} \dfrac{\sqrt{n}}{\lambda\sqrt{n} + 0.4255 p_{\max}\mu\sqrt{f} + \frac{2}{7} p_{\max}\pi f \cdot (\frac{d}{l})} = \dfrac{\sqrt{n}}{\rho_1 + \rho_2 \cdot (\frac{d}{l})} & \text{when } r/d \ge 7/3 \\ \dfrac{\sqrt{n}}{\lambda\sqrt{n} + 0.4255 p_{\max}\mu\sqrt{f}} = \dfrac{2\sqrt{n}}{\rho_1} & \text{when } r/d < 7/3 \end{cases} \quad (10)$$

We can also find out that if the object is moving fast, e.g., $r/d < 7/3$, the ratio of $p_{qf}(r)$ to $\psi(r)$ is a constant of $2\sqrt{n}/\rho_1$, if the object has a slower movement, $r/d \ge 7/3$, the ratio is a function of d/l, which is just the average velocity v of the object. Though in real case, the average velocity cannot be found for a unknown mobility, the dynamic velocity at every sampling time can be denote by the "real-time" d/l. So we can derive the ratio as $\sqrt{n}/(\rho_1 + \rho_2 \cdot v)$.

Motivated by this observation, we propose an adaptive algorithm to dynamically adapt approximation radius r .The basic idea is that we adjust the radius of approximate area (the sector) every time the remote updating take place according to the ratio of $p_{qf}(r)$ to $\psi(r)$. The current modification of the remote updating rate can be calculated by the time interval from the previous remote updating to the current one, and let α be a factor weighing the importance of the current update against past updates. We can get:

$$\psi(r)^{new} = \alpha\psi(r)^{old} + (1-\alpha)/\Delta T \quad (11)$$

Besides, we can obtain the current velocity of object v, and according to $r/(v \cdot l) < 7/3$ or $r/(v \cdot l) \ge 7/3$, the ratio of $p_{qf}(r)$ to $\psi(r)$ can be determined. As we get the ratio as the reference we can adjust the radius r to optimize the P-EASE schema. The adaptive adjustment of radius is given as:

Algorithm 4. Adaptive adjustment of r (at the centric storage node).

1: **for** each remote update do

```
    get the current moving speed of the object
    if r/(v·l) < 7/3 then
        set ρ ← 2√n / ρ1;
    else
        set ρ ← √n / (ρ1 + ρ2·v);
    end if;
    update the rate of remote updates ψ(r) according to: ψ(r)^new = αψ(r)^old + (1−α)/ΔT
    if p_qf(r)/ψ(r) > ρ·(1+ε) then
        set r' ← r/(1+δ);
    else if p_qf(r)/ψ(r) < ρ·(1−ε) then
        set r' ← r/(1−δ);
    end if;
    return r' to the new local storage node;
end for;
```

4.4. Location Error

In P-EASE and EASE, a tradeoff between the energy and precision is used in order to cut down the energy cost which definitely brings location error. When the error tolerance of user is greater than the radius of the approximate area, namely $p_{\max} \geq r$, the response to the querying node is an approximate result. So the location error is the difference between the response location and the real location of the object.

In EASE, the approximate area is a CIRCLE and there is no predicting, so the approximate response to the querying node is the centre of a circle, thus we can easily obtain the average distance from any point in the circle to the centre that is $2r/3$. While in P-EASE, the approximate area is a SECTOR (1/3 circle, centric angle is $2\pi/3$) and predicting is used to improve the precision. Hence, the responding querying result is some point on the axis of the sector. According to Appendix I, supposing two radiuses make an angle with θ, the average distance, from the point on one radius whose distance to the centre is x, to the other radius is given by:

$$\left(\frac{x^2\cos^2\theta}{2}+\frac{(r-x\cos\theta)^2}{2}\right)/r=\frac{x^2\cos^2\theta}{2r}+\frac{(r-x\cos\theta)^2}{2r} \tag{12}$$

Due to symmetry, the average distance from the point on axis whose distance to the centre is x to other radiuses is:

$$\left(\int_0^{\pi/3}\frac{x^2\cos^2\theta}{2r}+\frac{(r-x\cos\theta)^2}{2r}\right)/(\pi/3)=\left(\frac{\pi r}{6}+\frac{\pi x^2}{6r}+\frac{\sqrt{3}x^2}{8r}-\frac{\sqrt{3}x}{2}\right)/(\pi/3)=\frac{r}{2}+\frac{x^2}{2r}+\frac{3\sqrt{3}x^2}{8\pi r}-\frac{3\sqrt{3}x}{2\pi} \tag{13}$$

For any responding location located on the axis, supposing the distance from the location to the centre is considered x which is between range $[0,r]$.The average location error of P-EASE is laid:

$$\int_0^r \frac{r}{2}+\frac{x^2}{2r}+\frac{3\sqrt{3}x^2}{8\pi r}-\frac{3\sqrt{3}x}{2\pi}\,\mathrm{d}x=\frac{16\pi-15\sqrt{3}}{24\pi}r\approx 0.3220r \tag{14}$$

Therefore, we conclude that location error of P-EASE is less than that of EASE.

5. Optimal query schema

For any object, it is known that its centric storage node is generated by some algorithm according to the object id, e.g., GHT. The object is moving arbitrarily in the field and the query request may come from everywhere, which means the centric node's location has its randomicity and so does the querying node, the local storage node as well as the object itself. If the querying node is near the local storage node but far from the centric storage node, the querying node should send querying request to the ***far*** centric storage node according to approximate query schema. Obviously, this will cause much querying traffic no matter the centric node responds to the querying node or forwards it to the local storage node. Whereas, in this case, the query request should be sent to the ***near*** local storage node directly (as shown in **Fig. 10**).

So we should manage the querying request according to the notification from the centric storage node. In detail, when the centric storage node has received the remote updating, it should determine whether the local storage node is nearer to the querying node than itself. Then the centric storage node sends a notification message to tell the querying node the proper querying destination. The subsequent querying request from the same querying node will send to the destination notified by the centric node until the querying process ends or new notification from the centric node comes. And when the query ends, the query node should inform the centric node (as shown in **Fig. 11**).

The optimal query schema has two advantages: Firstly, it can avoid the unwanted traffic when the distance from the querying node to the centric storage node is far, while to the local storage node is comparatively near. Secondly, it will decrease the load of the centric node which is the hotspot of storage as well as querying.

5.1 Discussion

We suppose the field of the network is a circle which is just used in this paper as a case study for simplicity. The optimal schema presented here is applicable to other shaped field as long as long as their remote updating rate and querying rate can be estimate.

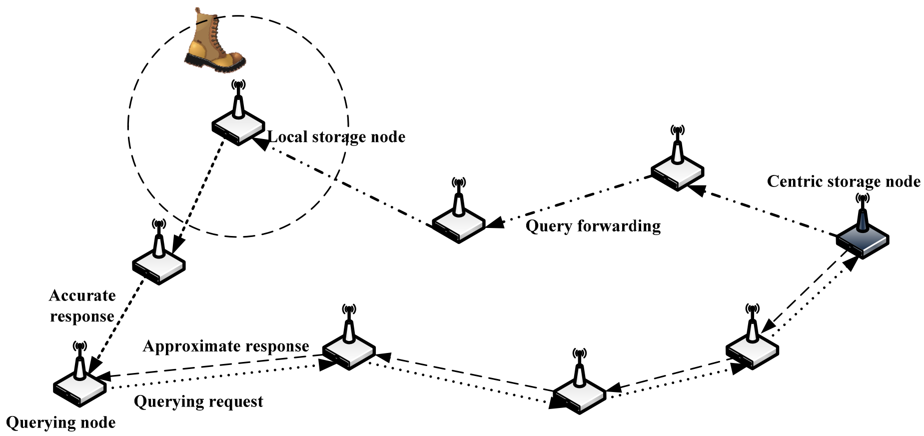


Fig. 10. Optimal query schema (a)

As shown in **Fig.12**, if the querying node gets the approximate response from the centric storage node, the travelling distance is $l_1 + l_1 = 2l_1$, else if accurate response returns from the local storage node, the travelling distance of is $l_1 + l_2 + l_3$. When $l_1 \geq l_3$, the approximate querying request is replaced by the accurate ones whose the traveling distance is $2l_3$, it is obvious that $2l_3 \leq 2l_1 \leq l_1 + l_2 + l_3$.

We observed that the optimal query schema does not necessarily reduce the network load, because the notification sent from centric storage node and the query-end informing sent from the local storage node would increase network traffic, and if the querying in the network is not frequent , then it might increase network load. Actually, whether the method used is determined by the remote update rate and the query rate. We assume that every querying request can be satisfied by the centric storage node, in other words, only approximate response is returned. The average distance of any two points within a circle is $2R/3$, therefore, the querying request from any querying node to any centric storage node will travel $\bar{l}_1 = 2R/3$ averagely. In addition, any querying node and local storage node should be on a radius of the circle, and the angle between the two radiuses is θ. According to Appendix I, the average distance between the querying node and the local storage node is $l_3 = R(2\cos^2\theta - 3\cos\theta + 3)/6$, summing up both cases the traveling distance of a querying request is given by $\min(\bar{l}_1, l_3)$. Hence, the cost cut down by the optimal query schema after a querying request is:

$$C_{s1} = \int_0^{2\pi} \frac{1}{2\pi} \cdot \max\left(\frac{R(2\cos^2\theta - 3\cos\theta + 3)}{6} - \frac{2}{3}R, 0\right) \mathrm{d}\theta \cdot \sqrt{f} \tag{15}$$

Let $f(\theta) = R(2\cos^2\theta - 3\cos\theta + 3)/6$, as shown in **Fig.13** we obtain $C_{s1} = 0.1662 \cdot R\sqrt{f}$ and the cost saved after a querying request and response is $C_s = 2C_{s1} = 0.3324 \cdot R\sqrt{f}$.

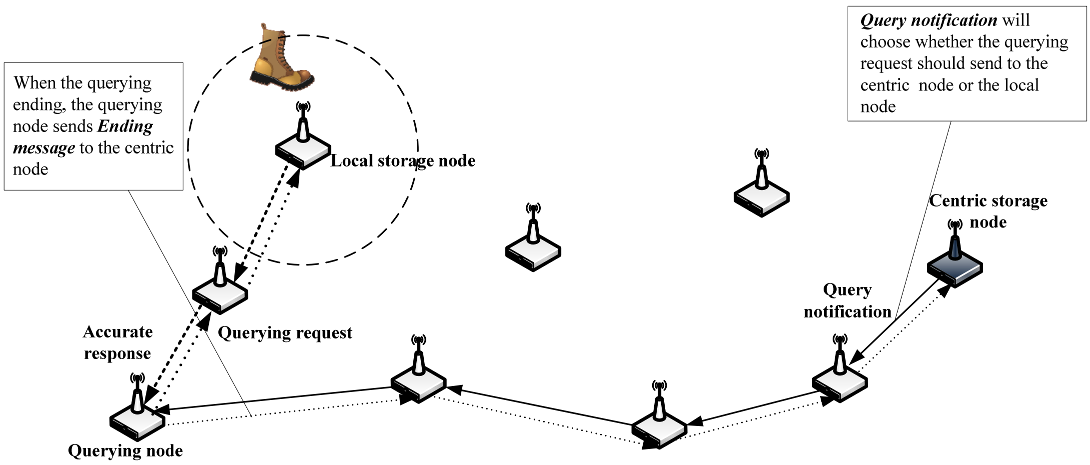


Fig. 11. Optimal query schema (b)

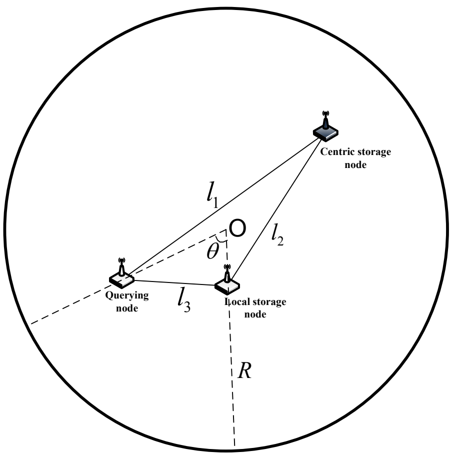


Fig. 12. Analysis of optimal query

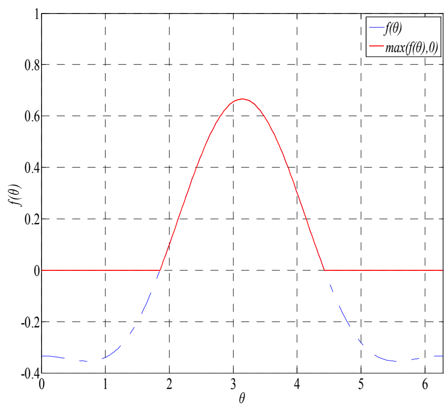


Fig. 13. The curve of Function $f(\theta)$

If the querying forwarding and accurate response sent from the local storage node is considered, the cost of a querying forwarding is $\sqrt{f} \cdot 2R/3$, as shown is fig… it is known that the probability the centric storage node is more distant to the querying node than the local storage node is 0.41, then the cost cut down by the optimal query schema after a querying forwarding and accurate response is:

$$C_{sj} = (\frac{2}{3}R\sqrt{f} + 0.1662\cdot R\sqrt{f})\cdot\frac{r}{p_{\max}}\cdot 0.41 \approx 0.3415\cdot R\sqrt{f}\cdot\frac{r}{p_{\max}} \tag{16}$$

So the overall cost saved is:

$$C_s = 0.3415\cdot R\sqrt{f}\cdot\frac{r}{p_{\max}} + 0.3324\cdot R\sqrt{f} = 0.1927\cdot\sqrt{n}\cdot\frac{r}{p_{\max}} + 0.1875\cdot\sqrt{n} \tag{17}$$

However the extra cost brought by notification is $\psi(r)\sqrt{n}$ (let α to be a factor weighing the importance of the current update against past updates and ΔT to be the time-interval of two updates, the real-time remote updating rate is given by $\psi(r)^{new} = \alpha\psi(r)^{old} + (1-\alpha)/\Delta T$), thus we can determine when the optimal querying schema. We conclude:

When $\psi(r)\cdot\sqrt{n} < (0.1927\cdot\sqrt{n}\cdot r/p_{\max} + 0.1875\cdot\sqrt{n})\cdot\lambda$, namely, $\psi(r) < (0.1927\cdot r/p_{\max} + 0.1875)\cdot\lambda$, the optimal querying schema is used, otherwise not.

5.2. Algorithm

The processing algorithms of querying node and the centric storage node are as follow:

Algorithm 5: query process of querying nodes

if receiving a notify n from the centre node **then**
    set $query\_dest \leftarrow n.node$, $query\_x \leftarrow n.x$ *and* $query\_y \leftarrow n.y$;
end if;
when ending query
    send a query end info i to centre node;

Algorithm 6: query process of the centric storage node

if receiving a query $< object\ id; p >$ from node q **then**
    if $\psi(r) < (0.1927\cdot r/p_{\max} + 0.1875)\cdot\lambda$ **and** *distance(centre, query)* ≥ *distance(local, query)* **then**
        send *nofiy*= <*local, local _x, local _y*> to query node;
        set *notify_flag[q]* == 1;
    end if;
    call Algorithm 1;
end if;
if receiving a remote location update message **then**
    foreach q in notify_flag
        if *notify_flag[q]* == 1
            if *distance(centre, q)* ≥ *distance(local, q)* **then**
                send *nofiy*= < *local, local_x, local_y*> to q;
            else
                send *nofiy*= < *centre, centre_x, centre_y*> to q;
            end if;
        end if;
    end foreach;
    call Algorithm 1;
end if;
if query end info i from query node q **then**
    notify_flag[q] == 0;
end if;

6. Performance Evaluation

In this section, we conduct simulation experiments to compare the proposed P-EASE scheme with the EASE schemes. We implemented our simulation based on OPNET 14.5. **Table 2** summarizes the system parameters and their settings used in our experiments. 144 sensor nodes are deployed in a $240 \times 240m^2$ field. We divided the field into 144 $20 \times 20m^2$ cells, and the radio range of each node is $30m$.

Table 2. System Parameters and Settings

Parameter	***Setting***	***Parameter***	***Setting***
Filed Size	$240 \times 240m^2$	GPSR Beacon Interva	5sec
Number of Nodes (n)	144	GPSR Beacon Expriation	15sec
Node Density (f)	$1/20 \times 20m^2$	Sensor Sampling Rate (μ)	2/sec
Radio Range	$30m$	Querying Rate (λ)	0.1-10/sec
Power Consumption (Sending Msgs)	$60mW$	Querying Msg Payload Size	10bytes
Power Consumption (Receiving Msgs)	$45mW$	Updating Msg Payload Size	56bytes
Power Consumption (Idel Listening)	$15.5mW$	Result Msg Payload Size	56bytes
Power Consumption (Sleeping)	$0.09mW$	Geocast Msg Payload Size	10bytes
B-MAC Preamble Length	271bytes	Notification Msg Payload Size	10bytes
B-MAC Wake-up Interval	100msec	Query Start Time	15sec
B-MAC Sampling Time	2.55msec	Simulation Time	360sec

In order to save energy like EASE, we adopted B-MAC [9] to be the MAC protocol of nodes and use an energy model to measure the energy consumption. Querying node sends the querying request according to the querying rate and the error tolerance distribute in $[0, p_{max}]$ uniformly. The GPSR [23] is employed to be the routing protocol. We take two representative mobility models which are the Random Waypoint and the Semi-random Walk [16] to simulate object's movement.

For the sake of comparing the message complexity, the cost of a scheme should be normalized, that is, defined as the ratio of the measured cost of the scheme to that of DCS.

The simulation takes 360 seconds, and the first 15 seconds is a warm-up, after which the performance evaluation begins.

6.1. Optimal Setting of the Approximation Radius

With the analysis in Section 4.2, the initial optimal radius of the approximation sector in P-EASE schema can be determined by (8) and (9) according to the parameter of the network. Besides, the approximation radius of EASE can also be worked out according to [7]. We assume the maximum error tolerance is set at 50m by default, and query rate is set at 1/sec. The speed of the object under Random waypoint mobility is uniformly distributed between [1m/s, 10m/s], while the speed under Semi-random walk mobility varies from 10m/s to 20m/s. While the approximate radius changes from 0m to 50m, we get the message complexity of P-EASE and EASE both under Random waypoint and Semi-random walk mobility. The average speed of the object is obtained by: $v = d / l$, where d is the distance the object moves at each step, and l is the time it takes (duration). We can easily get that under our Random waypoint mobility, it satisfies $r / d \geq 7 / 3$, so we compute the optimal radius by (8), while under our Semi-random walk mobility, we'd better determine the optimal radius by (9).

As **Fig. 14** shown, when the radius is less than 15m, the message complexity of P-EASE is similar to that of EASE under both mobility. However, when the approximate radius is longer than 15m, compared to EASE, under Random waypoint mobility, the message complexity of

P-EASE is reduced by 12%-20%, while under Semi-random walk mobility; the message complexity of P-EASE is reduced by 12.5%-28%.

Additionally, we can also find out that the analytical optimal approximation radius is close to our actual radius in the optimal performances, which validates our analysis in Section 4.2.

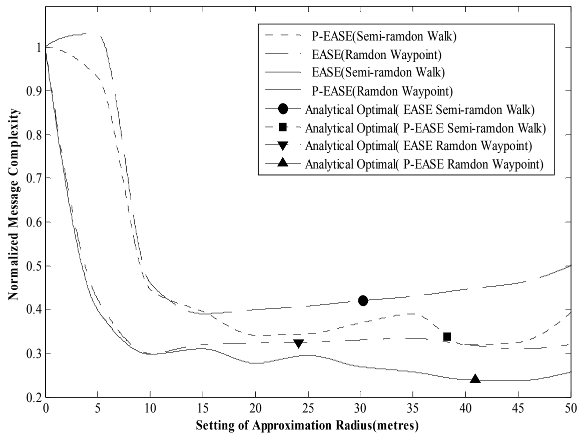


Fig. 14. Analysis and simulation results of optimal setting of the approximation radius(λ =1/sec and p_{max}=50m)

6.2. Message Complexity

This section focuses on message complexity of P-EASE and EASE under both Random waypoint and Semi-random walk mobility. Firstly, the object is under Semi-random walk mobility, query rate λ is set at 1/sec, and the maximum error tolerance $p_{\max}$ is set at 50m, we let the radius of approximate area vary from 0m to 50m, compare the number of different kinds of messages; After that we normalized the message complexity to examines how the normalized message complexity is changed along with the optimal radius under various error tolerances and querying rate, and compare it to EASE; Then, we adopted the adaptive approximation setting to observe the average message complexity; Finally, we showed the advantage result from the optimal query schema discussed in Section 5.

1) Number of Messages.

Firstly, the maximum error tolerance is set at 50m and querying rate is set at 1/sec, and we let the approximate radius changes from 0m to 100m. **Fig. 15** shows that when the radius r is small, the number of messages of P-EASE is similar to which of EASE, however, the difference of the number of messages between the two schema increases along with the increasing of r . Especially, when $r > 40m$, the total number of messages of P-EASE is reduced by 30% compared to EASE, which means that the number of messages of P-EASE is much less than EASE with a proper approximate area.

Secondly, under Semi-random walk mobility, maximum error tolerance is set at 50m, querying rate is set at 1/sec, and for each schema the radius of approximate area is set at its optimal setting according to **Fig. 16** (P-EASE: 42.81m, EASE: 35.25m, and the following paper). Fig… shows that compared to EASE, the query request of P-EASE is reduced by 43%, query result is reduced by 35%, and query forwarding is reduced by 72%. The local updating, geocast and remote updating of P-EASE is similar to that of EASE, while P-EASE brings in some extra notification messages. Overall, the total number of messages of P-EASE is 31% less than EASE.

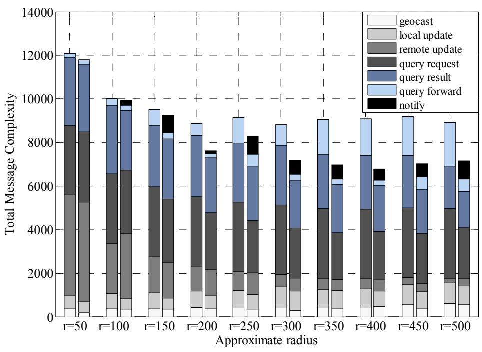


Fig. 15. Breakdown of message complexity versus approximate radius (Semi-random walk, p_{max} =50m and λ =1/sec)

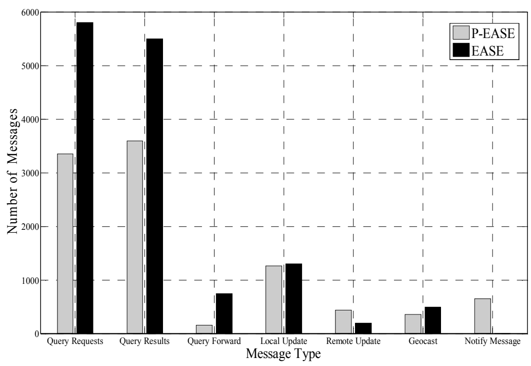


Fig. 16. Breakdown of message complexity (Semi-random walk, p_{max} =50m and λ =1/sec)

2) Message Complexity.

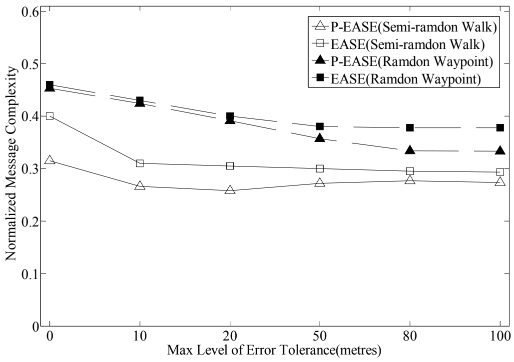
The message complexity is one of the key performances we must concern about. The querying rate is set at constant 1/sec and the max error tolerates changes from 0m to 100m. As **Fig. 17-(a)** shows that under the assumption of random waypoint mobility, when the max error tolerance is less than 50m, the message complexity of EASE is close to that of P-EASE, this is because the reduction of querying message, local updating and geocast messages are counteracted along with the increasing of remote updating and notification. However, when $p_{\max} > 50m$, P-EASE decreased the message complexity by 7.5%-12.5%. For the semi-random walk mobility, P-EASE will not bring extra remote updating because of the orderliness of object moving. Hence, the message complexity of P-EASE will be reduced by 6.25%-25% compared to EASE.

Then we let the max error tolerance to be constant 50m, and the querying rate changes from 0.1/sec to10/sec follows the logarithm. As shown in **Fig. 17-(b)**, we can find that under random waypoint mobility when $\lambda > 0.33 / \mathrm{sec}$, the message complexity is reduced by 9%-28% in P-EASE compared to EASE, and under the semi-random walk mobility, the message complexity in P-EASE decreased by 10%-23%.

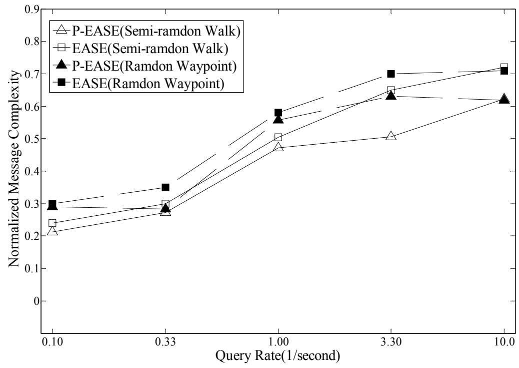
3) Average Message Complexity.

Then we obverse how the message complexity changes over time under both random waypoint and semi-random walk mobility in P-EASE compared to EASE. The error tolerance is set to 50m and querying rate is 1/sec. As shown in **Fig. 18-(a)**, it is obvious that under semi-random walk mobility the cost of P-EASE is 32.3% less than EASE. On 15s，30s，50s-70s，75s，105s，115s and 125s the extra messages in P-EASE comes from the notification when using optimal querying schema. **Fig. 18-(b)** shows that under random waypoint mobility, the message complexity is reduced by 58.2%.

Hence, we can conclude that the message complexity in P-EASE is substantially reduced compared to EASE under both mobility models.

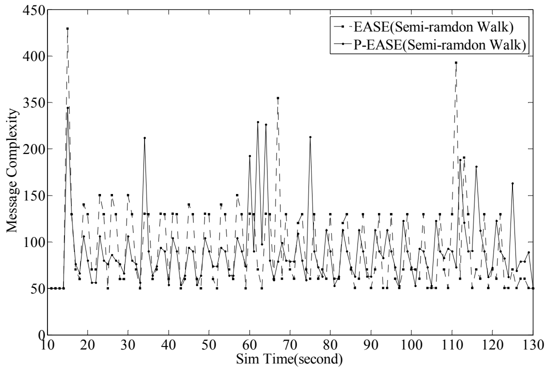


(a) Normalized message complexity versus query precision requirement(λ =1/sec)

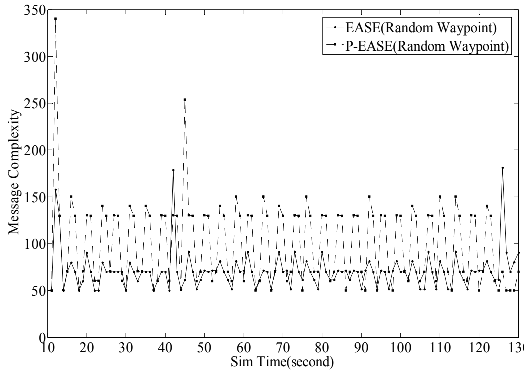


(b) Normalized message complexity versus query rate(p_{max} =50m)

Fig. 17. Normalized message complexity



(a) Message complexity along time series (semi-random walk, λ =1/sec, p_{max} =50m)

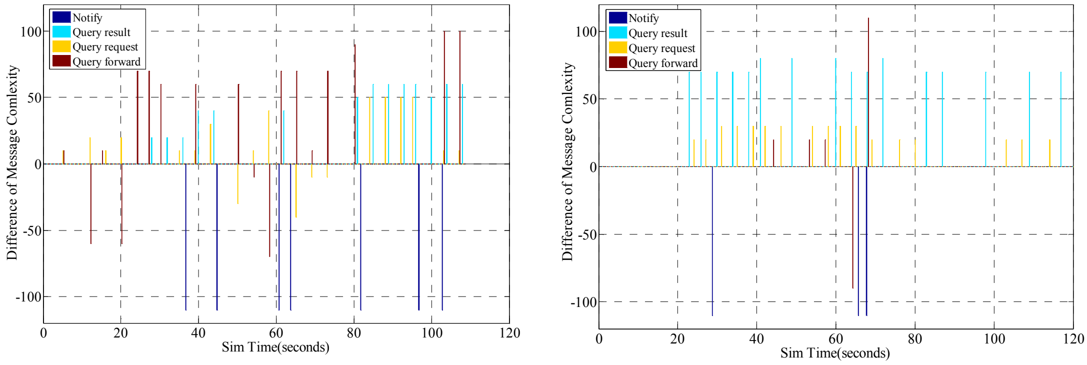


(b) Message complexity along time series (random waypoint, λ =1/sec, p_{max} =50m)

Fig. 18. Message complexity along time series

4) Advantage of Optimal Query Schema.

As **Fig. 18-(a)** and **Fig. 18-(b)** show, optimal query schema will decrease the cost of querying greatly, which directly cut down the message complexity of the network. From the analytical result in Section 4.2 and Section 5, the system can obtain the remote updating rate by $\theta(r)^{new} = \alpha\theta(r)^{old} + (1-\alpha)/\Delta T$, and only if $\theta(r) < (0.1927 \cdot r / p_{\max} + 0.1875) \cdot \lambda$, the optimal query should take place. **Fig. 19-(a)** and **Fig. 19-(b)** show the differences between P-EASE and EASE under both Random waypoint and Semi-random walk mobility along time series. There are 4 types of messages, which are query request, query forwarding, query result and notification. In **Fig. 19-(a)** and **Fig. 19-(b)**, compared to EASE, the plus part means how much the message complexity of P-EASE has been cut down, while the minus part means extra message complexity in P-EASE. Simulation results show that proper use of the optimal query schema can decrease the cost of the network largely.



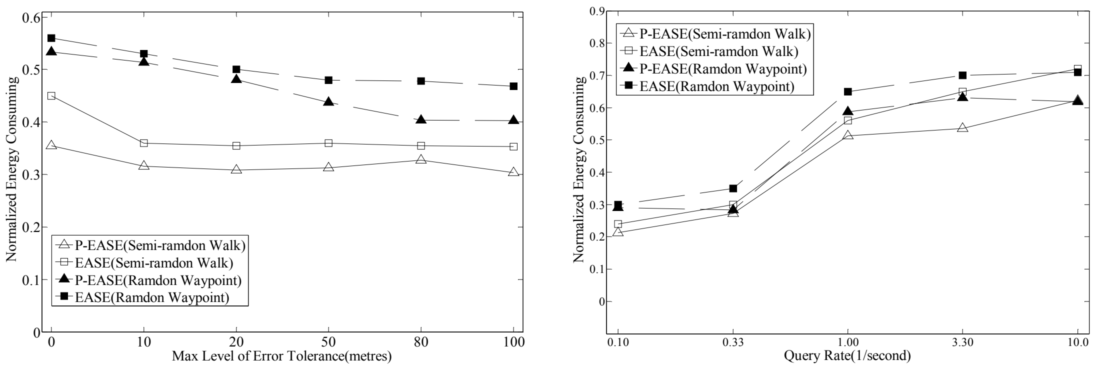
(a) Message difference between P-EASE and EASE along time series (Semi-random Walk)

(b) Message difference between EASE and P-EASE along time series (Random Waypoint)

Fig. 19. Message difference between P-EASE and EASE along time series

6.3. Energy Consumption

The energy consumption of the network is mostly due to the communication traffic, **Fig. 20-(a)** and **Fig. 20-(b)** show the normalized energy consumption under various error tolerances and various querying rate in log scale, which is ultimately inosculate with the normalized message complexity.



(a) Normalized total energy consuming versus query precision requirement(λ=1/sec)

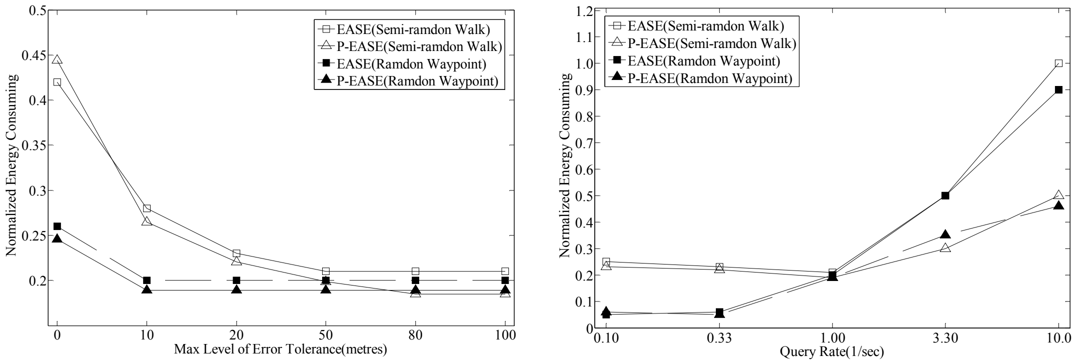
(b) Normalized total energy consuming versus query rate(p_{max} =50m)

Fig. 20. Normalized total energy consuming

Besides, the energy consumption of the hotspot should be discussed here as well. When the querying rate is set to 1/sec, the max error tolerance varies from 0m to 100m, as shown in **Fig. 21-(a)**, we found that the energy consumption of the P-EASE is close to which of the P-EASE. This is because the energy consumption of the hotspot mostly results from the querying process and remote updating, and if the querying rate is low, the optimal querying schema will not be used frequently, which leads to the result. However, when the max error tolerance is constant 50m, and the querying rate changes from 0.1/sec to 100/sec in log scale, as shown in **Fig. 21-(b)**, the energy consumption of the hotspot in P-EASE will be reduced by 50% compared to EASE.

6.4. Location Error

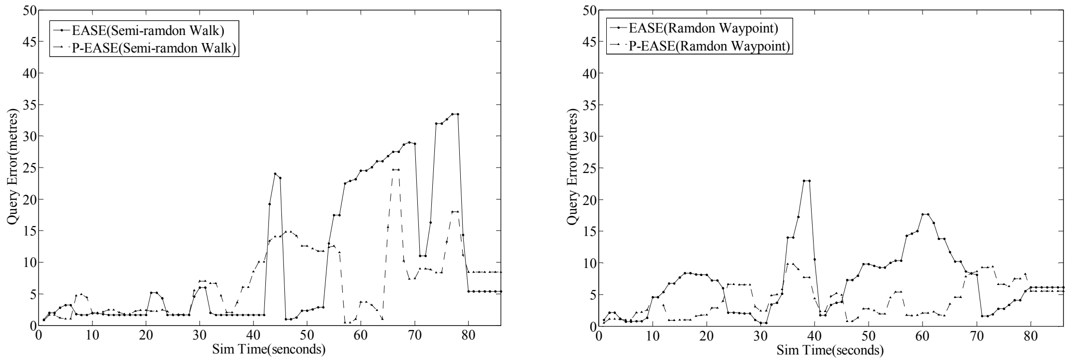
P-EASE is expected to increase the location precision compared to EASE; actually we are indicated from **Fig. 22-(a)** and **Fig. 22-(b)** that no matter under the random waypoint or the semi-random walk mobility, the location error of P-EASE is evidently lower than that of EASE. Under the random waypoint mobility, the P-EASE reduced the error by 39.9%, while under semi-random walk mobility, it cut down it by 28.9%, which is ultimately consistent with our analysis.



(a) Normalized energy consuming of most consuming node versus query precision requirement(λ =1/sec)

(b) Normalized energy consuming of most consuming node versus query rate(p_{max} =50m)

Fig. 21. Normalized energy consuming of most consuming node



(a) Query error along time series (Semi-random Walk)

(b) Query error along time series (Random Waypoint)

Fig. 22. Query error along time series

7. Conclusion

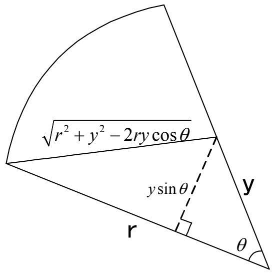
In this paper, a prediction-based energy-conserving approximate storage schema (P-EASE) is proposed to reduce the location error resulted from the approximate querying in EASE firstly. In P-EASE, the SECTOR whose central angle is $2\pi / 3$ (1/3 circle) is adopted instead of CIRCLE to adapt to the predicting method, which can reduce the location error effectively without increasing the message complexity of the network. Furthermore, the optimal query schema is introduced to reduce the additional cost caused by choosing improper destination of querying request in EASE, and its working conditions are determined by the parameters like querying rate, max tolerance error and so on. The optimal querying schema can reduce the hotspot's overhead at a certain extent as well. Theoretical analysis and simulation experiment results show that P-EASE's query precision and energy efficiency should be significantly higher than that of EASE.

Our future work will firstly consider the problem of load balance, because the centric storage node is the hotspot of both storage and querying, and the HDC method in [25] may help to balance the load. In addition, predicting model in the P-EASE can be replaced by others (e.g., [26]) in order to adapt to more complicated object mobility. Furthermore, the inherent relationship between the location error and the updating rate should be discussed, which is determined by the network parameters such as the radius or the centre angle of the approximate area.

Appendix

Appendix I: Analysis of the average distance between two points in a sector with the centre angel is $2\pi/3$.

Firstly, we need to calculate the average distance between the points on two different radiuses of the sector that meet with the angle of θ. As shown in the figure, we can see that when a point is on a radius of the sector that is y distant to the centre, then the average distance between the point and another radius is given by:



$$\int_{y\sin\theta}^{y} x\mathrm{d}x + \int_{y\sin\theta}^{\sqrt{r^2+y^2-2ry\cos\theta}} x\mathrm{d}x = \frac{y^2\cos^2\theta}{2} + \frac{(r-y\cos\theta)^2}{2}$$

So the point on the two radiuses is:

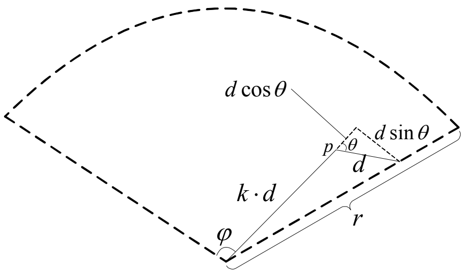
$$\left(\int_0^r \frac{y^2\cos^2\theta}{2} + \frac{(r-y\cos\theta)^2}{2}\mathrm{d}y\right) / (r\cdot r) = \frac{r(2\cos^2\theta - 3\cos\theta + 3)}{6}$$

And any point in the sector must be on a radius, so that the average distance between two points in a sector with the centre angel is $2\pi/3$ we obtain:

$$\int_0^{2\pi/3} \frac{3}{2\pi}\cdot\frac{r(2\cos^2\theta-3\cos\theta+3)}{6}\mathrm{d}\theta = \frac{r(32\pi-21\sqrt{3})}{48\pi} \approx 0.4255r$$

Appendix II: Analysis of the remote updating rate because of the angle.

Let d be the moving distance of each step, and Each step takes a duration of l, [7] investigated under the random walk mobility the remote updating rate $\eta(r)$ can be approximated by $d^2/(l\cdot r^2)$. However, this remote updating rate is caused by the ***displacement*** from centre of the sector to the current location detected by the detecting node that is greater than the radius of the approximate area. We should analyze the remote updating rate because of the ***angle*** here, denote by $\eta'(r)$.



We assume that at any time instant the object is $k\cdot d$ distant to the centre, $k>0$. As shown in the figure right, we can get:

$$\varphi = \frac{2\pi}{3} - \arctan\left(\frac{\sin\theta}{k+\cos\theta}\right)$$

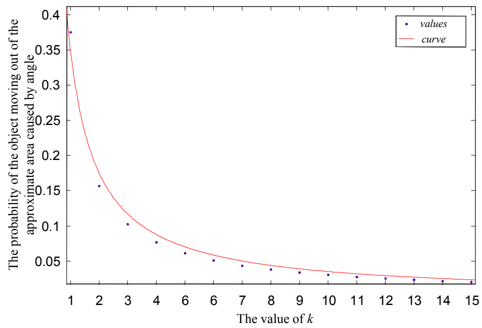
So, for each given k and θ, the probability of the object moving out of the approximate area caused by angle $P(s)$ is:

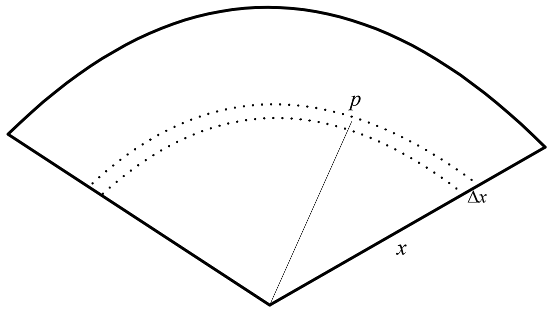
$$P_{k,\theta}(s) = \frac{\varphi}{2\pi/3} = 1 - \frac{3}{2\pi}\arctan(\frac{\sin\theta}{k+\cos\theta})$$

And for any k, θ is various in $[0,\pi]$,which gives that:

$$P_k(s) = \int_0^{\pi} 1 - \frac{3}{2\pi}\arctan(\frac{\sin\theta}{k+\cos\theta})\, d\theta$$

From the curve of the function $P_k(s)$ below, $P_k(s)$ can be approximated by $P_k(s) = 0.35/k$





The probability that a point p locate in a ring enclosed by two circles with radii x and $x+\Delta x$ is given by:

$$\frac{\pi x^2 - \pi(x+\Delta x)^2}{\pi r^2} = \frac{2x\Delta x + (\Delta x)^2}{r^2} \approx \frac{2x\Delta x}{r^2}$$

For a given, x, the time that an object moves out of the area because of the angle is :

$$\frac{2x\Delta x}{r^2}\cdot\frac{x^2 l}{d^2}\cdot\frac{0.35d}{x} = \frac{7lx^2\Delta x}{dr^2}$$

So we get the average time that an object moves out of the area because of the angle: $\int_0^r \frac{7lx^2}{dr^2} x dx = \frac{7rl}{3d}$ which lead the remote updating rate to be $\eta'(r) = \frac{3d}{7rl}$.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci , "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 392-422, Mar. 2002. Article (CrossRef Link)

[2] Euisin Lee, Soochang Park, Fucai Yu, Younghwan Choi, Min-Sook Jin and Sang-Ha Kim. "A Predictable Mobility-based Data Dissemination Protocol for Wireless Sensor Networks," in *Proc. of IEEE 22nd Int'l Conf. on Advanced Information Networking and Applications*, pp. 741-747, Mar. 2008. Article (CrossRef Link)

[3] Z. Zhong, T. Zhu, D.Wang and T. He, "Tracking with Unreliable Node Sequences," in *Proc. of 28th IEEE Conf. on Computer Communications*, pp. 1215–1223, Apr. 2009, Rio de Janeiro, Brazil. Article (CrossRef Link)

[4] S. Goel and T. Imielinski, "Prediction-Based Monitoring in Sensor Networks: Taking Lessons from MPEG," in *Proc. of ACM SigComm Computer Communication Rev.*, vol. 31, no. 5, pp. 82-98, Oct. 2001. Article (CrossRef Link)

[5] C. Gui and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," in *Proc. of ACM MobiCom '04*, pp. 129-143, Oct. 2004. Article (CrossRef Link)

[6] W. Zhang and G. Cao, "Optimizing Tree Reconfiguration for Mobile Target Tracking in Sensor Networks," in *Proc. of IEEE InfoCom'04*, vol. 4, pp. 2434-2445, Mar. 2004. Article (CrossRef Link)

[7] J. Xu, X. Tang, W. Lee. "A New Storage Scheme for Approximate Location Queries in Object-Tracking Sensor Networks," *IEEE Trans.on Parallel and Distribute Systems*, vol. 19, no. 2, pp. 262-275, Feb. 2008. Article (CrossRef Link)

[8] H. Liu, X. Jia, P. Wan, C.-W. Yi, S. Makki and P. Nik "Maximizing Lifetime of Sensor Surveillance Systems," *IEEE ACM Trans. Networking*, vol. 15, no. 2, pp. 334-345, Apr. 2007. Article (CrossRef Link)

[9] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Proc. of ACM 2nd Int'l Conf. on Embedded networked sensor systems*, pp. 95-107, Nov. 2004. Article (CrossRef Link)

[10] Q.X. Wang, W.P. Chen, R. Zheng, K. Lee and L. Sha, "Acoustic Target Tracking Using TinyWireless Sensor Devices," in *Proc. of 2nd Int'l Workshop Information Processing in Sensor Networks (IPSN '03)*, pp. 642-657, Apr. 2003. Article (CrossRef Link)

[11] H. Yang and B. Sikdar, "A Protocol for Tracking Mobile Targets Using Sensor Networks," in *Proc. of IEEE Workshop Sensor Network Protocols and Applications*, pp.71-78, May 2003. Article (CrossRef Link)

[12] S. Suganya, "A Cluster-based Approach for Collaborative Target Tracking in Wireless Sensor Networks," in *Proc. of the First International Conference on Emerging Trends in Engineering and Technology (ICETET'08)*, pp. 276-281, Jul. 2008. Article (CrossRef Link)

[13] X. Wang, J.J. Ma, S. Wang and D.W. Bi, "Cluster-based Dynamic Energy Management for Collaborative Target Tracking in Wireless Sensor Networks," *Sensors*, vol. 7, no. 7, pp. 1193-1215, 2007. Article (CrossRef Link)

[14] T. Kaur and J. Baek, "A Strategic Deployment and Cluster-Header Selection for Wireless Sensor Networks," *IEEE Transactions on Consumer Electronics (IEEE TCE)*, vol. 55, no. 4, pp. 1890–1897, Nov. 2009. Article (CrossRef Link)

[15] Y. Xu, J.Winter and W.-C. Lee, "Dual Prediction-Based Reporting Mechanism for Object Tracking Sensor Networks," in *Proc. of First Ann. Int'l Conf. Mobile and Ubiquitous Systems (MobiQuitous '04)*, pp. 154-164, Aug. 2004. Article (CrossRef Link)

[16] H.T. Kung and D. Vlah, "Efficient Location Tracking Using Sensor Networks," in *Proc. of IEEE Wireless Comm. and Networking Conf. (WCNC '03)*, vol.3, pp.1954-1961, Mar. 2003. Article (CrossRef Link)

[17] M. Abhishek, M.S. Kami, O. Lawrence and S. Bo, "An Intelligent Energy Efficient Target Tracking Scheme for Wireless Sensor Environment," in *Proc. of 5th Int'l Symposium on Wireless Pervasive Computing (ISWPC)*, pp. 93-97, May 2010. Article (CrossRef Link)

[18] S. Pattem, S. Poduri and B. Krishnamachari, "Energy-Quality Tradeoffs for Target Tracking in Wireless Sensor Networks," in *Proc. of Second Int'l Workshop Information Processing in Sensor Networks (IPSN '03)*, pp. 32-46, Apr. 2003. Article (CrossRef Link)

[19] D. Smith and S. Singh, "Approaches to Multisensor Data Fusion in Target Tracking: A Survey," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 12, pp. 1696-1710, Dec. 2006. Article (CrossRef Link)

[20] V.P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar and N. Shroff, "A Minimum Cost Heterogeneous Sensor Network with a Lifetime Constraint," *IEEE Trans. On Mobile Computing*, vol. 4, no. 1, pp. 4-15, Jan./Feb. 2005. Article (CrossRef Link)

[21] L. Qing, Q.X. Zhu and M.W. Wang, "Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks," *Computer Communications*, vol. 29, no. 12, pp. 2230-2237, Aug. 2006. Article (CrossRef Link)

[22] V.Mhatre and C. Rosenberg, "Homogeneous vs heterogeneous clustered sensor networks: a comparative study," in *Proc. of IEEE Int'l Conf. on Communications*, vol. 6, pp. 3646–3651, Jun. 2004. Article (CrossRef Link)

[23] 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/Kluwer Mobile Networks and Applications*, vol. 8, no. 4, pp. 427-442, Aug. 2003. Article (CrossRef Link)

[24] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Sensor Networks," in *Proc. of ACM MobiCom '00*, pp. 243-254, Aug. 2000. Article (CrossRef Link)

[25] T. Zijin, G. Zhenghu and L. Zexin, "Two New Push-Pull Balanced Data Dissemination Algorithms for Large-Scale Wireless Sensor Networks," *Journal of Computer Research and Development* , China, vol. 45, no. 7, pp. 1115~1125, 2008. Article (CrossRef Link)
[26] J. Teng, H. Snoussi and C. Richard, "Prediction-based Proactive Cluster Target Tracking Protocol for Binary Sensor Networks," in *Proc. of IEEE Int'l Symposium on Signal Processing and Information Technology*, pp.234-239, Dec. 2007. Article (CrossRef Link)

Yi Xie received the B.S. and M.S. degree from National University of Defense Technology (NUDT) in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree in the Science and Technology Information Systems Engineering Laboratory of NUDT, Changsha, China. He is a professional member of ACM and
Chinese Computer Federation (CCF). His current research interests are in target-tracking wireless sensor networks, data management in sensor networks, mobile sensor networks and information resource management in wireless networks.

Weidong Xiao received the B.S. and M.S. degree from National University of Defense Technology (NUDT) in 1990 and 1998, respectively and obtained his Ph.D. there in 2004. He is currently an Professor with Information resource management and also a supervisor of Ph.D. candidates in the Science and Technology Information Systems Engineering Laboratory of NUDT. He is a director of Chinese Institue of Manament Science and Engineering. His current research interests are in data management in sensor networks, information management in ubiquitous networks and Internet of Things (IoTs).

Daquan Tang received the B.S. and M.S. degree from National University of Defense Technology (NUDT) in 1994 and 2002, respectively. He is currently an Professor at College of Information System and Management of NUDT, where he obtained his Ph.D. in 2009. He is the senior member of the Chinese Institue of Electronics. His current research interests are in data management in sensor networks, spatio-temporal data management and target-tracking wireless sensor networks.

Jiuyang Tang received the B.S. degree from National University of Defense Technology (NUDT) in 2000. He is currently an Associate Professor at College of Information System and Management of NUDT, where he obtained his Ph.D. in 2006. His current research interests are in topology control in sensor networks, P2P, ubiquitous networks and Internet of Things (IoTs).

Guoming Tang received the B.S. degree from National University of Defense Technology (NUDT) in 2010. He is currently pursuing the M.S. degree in the Science and Technology Information Systems Engineering Laboratory of NUDT, Changsha, China. His current research are mainly on target-tracking wireless sensor networks, and information resource management in wireless networks.