

k -nearest neighbor search based on node density in MANETs

Yuka Komai*, Yuya Sasaki, Takahiro Hara and Shojiro Nishio

*Department of Multimedia Engineering, Graduate School of Information Science and Technology,
Osaka University, Osaka, Japan*

Abstract. In a k NN query processing method, it is important to appropriately estimate the range that includes k NNs. While the range could be estimated based on the node density in the entire network, it is not always appropriate because the density of nodes in the network is not uniform. In this paper, we propose two k NN query processing methods in MANETs where the density of nodes is ununiform; the One-Hop (OH) method and the Query Log (QL) method. In the OH method, the nearest node from the point specified by the query acquires its neighbors' location and then determines the size of a circle region (the *estimated k NN circle*) which includes k NNs with high probability. In the QL method, a node which relays a reply of a k NN query stores the information on the query result for future queries.

Keywords: MANETs, k NN query, LBS

1. Introduction

A location-based service (LBS) [15] is a typical application in *mobile ad hoc networks (MANETs)* [1, 2, 7, 10, 13, 16, 19, 25]. In an LBS, it is common that a node issues queries to search information on a specific location held by a mobile node in real time. In such a case, it is effective to process the queries as k nearest neighbor (k NN) queries, which search the information on the k nearest neighbors (k NNs) from a specified location (*query point*) [4, 6, 14, 17, 21, 26–28].

In our previous work, we proposed a k NN query processing method, the Explosion (EXP) method, which can reduce traffic and also maintain high accuracy of the query result in MANETs [12]. In the EXP method, the query-issuing node first transmits a k NN query using geo-routing to the nearest node from the query point (the *global coordinator*). Then, the global coordinator floods the k NN query to nodes within a specific circle region (the *estimated k NN circle*) whose center is the query point, which looks like a query message that explodes at the global coordinator. Each node that received the query replies with the information on itself to the global coordinator, and the global coordinator sends back k NNs to the query-issuing node.

It is very important to appropriately determine the size of the estimated k NN circle because it directly affects performance. If the estimated k NN circle is set too small, the information of all k NNs may not be acquired because there may be less than k nodes within the estimated k NN circle. On the other hand, if the estimated k NN circle is set too large, unnecessary traffic increases because nodes that are not

*Corresponding author: Yuka Komai, Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Yamadaoka 1-5, Suita-shi, Osaka, Japan. E-mail: komai.yuka@ist.osaka-u.ac.jp.

included in the result of the k NN query reply with their information. In the EXP method, the estimated k NN circle is determined based on the density of nodes in the entire MANET. However, in a real environment, it is not always easy to know the total number of nodes in the entire MANET and the area size beforehand. Moreover, since the density of nodes is generally not uniform in a MANET, the estimated k NN circle, which is set based on the density in the entire area, is not always appropriate.

In this paper, we propose two extended EXP methods; the One-Hop (OH) method and the Query Log (QL) method, for reducing traffic and also maintaining high accuracy of the query result in MANETs where the density of nodes is not uniform. In the OH method, the global coordinator acquires its neighbors' information (only one-hop nodes' information) by exchanging messages to know the density of nodes near the query point. If the number of neighbors exceeds k , the global coordinator can reply with k NNs to the query-issuing node. If not, the global coordinator sets the radius of the estimated k NN circle based on the density of nodes within its communication range, and acquires the information on nodes within the estimated k NN circle. In the QL method, a node that relays a reply for a k NN query stores the information on the query result to use it for determining the estimated k NN circle for future queries. During query forwarding, the query-issuing and query-relaying nodes attach some of the stored information to the query, which is used to estimate the density of nodes near the query point. The global coordinator then estimates the radius of the estimated k NN circle using some of the attached information, and acquires the information on nodes within the estimated k NN circle. These methods can set the size of the estimated k NN circle more appropriately using the information acquired during the query execution even if each node cannot know the information on the area size and the total number of nodes beforehand, and the density of nodes in the entire network is not uniform.

We also explain some experimental results to verify that our proposed methods can reduce traffic compared with the EXP method and also achieve high accuracy of the query result.

The contributions of this paper are as follows:

- Since the network bandwidth and batteries of mobile nodes are limited in MANETs, it is very important for k NN query processing to reduce unnecessary query messages and replies (i.e., traffic) as much as possible. We propose two effective k NN query processing methods (the One-Hop (OH) and Query Log (QL) method) for reducing traffic and also maintaining high accuracy of the query result.
- The performance of these methods is affected by several factors such as k and network topology. Thus, by adaptively choosing one of the two methods, we can adapt to various system situations.
- Through extensive simulations, we show that our proposed methods work very well in terms of both traffic reduction and high accuracy of the query result.

The remainder of this paper is organized as follows: In Section 2, we introduce related work. In Section 3, we explain our previous work. In Section 4, we present our proposed k NN query processing methods. In Section 5, we show the results of the simulation experiments. Finally, we summarize this paper in Section 6.

2. Related work

In [5,23], the authors proposed infrastructure-free k NN query processing methods. In these methods, a query is first transmitted to the nearest node from the query point [11], adding the information for setting the search range that contains k NNs with high probability. Then, the nearest node from the query point estimates the size of the search range based on the information attached to the query. In [23], a

relaying node sends the query with a list including its location and the number of newly encountered neighbors. The nearest node from the query point determines the size of the search range based on the information on the list. On the other hand, in [5], a relaying node updates the query message including the total area covered by the communication ranges of all relaying nodes and the total number of nodes within the area. Then, the nearest node from the query point determines the search range based on the calculated density of nodes within the area. After that, the search range is partitioned into some sectors. With respect to each sector, a node collects partial results that contain information on nodes within its communication range and propagates the query to the next node along a well devised itinerary structure.

However, these methods basically assume a location-aware sensor network. More specifically, in these methods, each sensor node must precisely know its neighbors (e.g., by frequently exchanging beacon messages), which causes too much overhead in highly dynamic MANETs. On the other hand, in our proposed methods, each mobile node does not have to know the network topology or its neighbors beforehand, which is more suitable for MANETs.

In [24], the authors proposed methods for processing k NN queries in location-aware sensor networks; the GRT, KBT and IKNN algorithms. In the KBT algorithm, a tree infrastructure composed of sensor nodes is constructed and a k NN query propagates along it. The nearest node from the query point determines the search range using for example an approach based on the number of hops during geo-routing. However, in these approaches, because the radius of the search range is set large enough in order not to miss k NNs, unnecessary replies are sent back from nodes that are in the circle but not k NNs. Moreover, a static sensor network is basically assumed with these methods; therefore, they cannot be directly applied to highly dynamic MANETs.

In [9], the authors proposed a k NN query processing method in a 3D sensor network. This method adopts a data collector that efficiently tours k NNs. Because it is assumed that nodes are uniformly distributed in the network and each node knows the area size and number of nodes in the entire network with this method, the search range is set based on the average density of nodes in the network similar to our EXP method. However, as mentioned, in a real environment, it is not always easy to know the total number of nodes in the entire MANET and the area size beforehand. Moreover, the density of nodes is generally not uniform in a MANET.

3. Previous work: EXP method

In this section, we describe the EXP method that we previously proposed [12].

3.1. Geo-routing for forwarding query to query point

In the EXP method, the query-issuing node first forwards a k NN query using our geo-routing method (an extension of the protocol proposed in [8]) to the global coordinator. Our geo-routing method adopts a three-way handshake protocol to send a query to the neighboring node closest to the query point among its neighboring nodes. By repeating this procedure, the query is forwarded to the global coordinator.

More specifically, in our geo-routing method, the query-issuing node first broadcasts a neighbor searching message. Then, when a node receives the neighbor searching message, if it is closer to the query point than the source node, it sets the waiting time for sending a reply. Because nodes closer to the query point transmit a reply message after a shorter waiting time, the nearest node from the query point among the neighbors firstly transmits a reply message to the source node. Then the source node that received reply messages from its neighbors sends a (forwards the) k NN query message only to the

node that firstly sent the reply. The node that received the forwarded k NN query broadcasts a neighbor searching message in the same procedure. Finally, if the node that sent a neighbor searching message does not receive any reply messages when the query point is included in its communication range, it recognizes itself as the global coordinator and starts acquiring k NNs.

Therefore, the query-issuing node forwards a k NN query to the global coordinator with little traffic because this geo-routing method neither uses beacon messages nor constructs multi-paths.

3.2. Forwarding k NN query and replying result

In the EXP method, we assume each node knows the total number of nodes and the size of the entire area in which nodes exist. First, the query-issuing node determines the size of the estimated k NN circle based on the density of nodes in the entire area. After receiving a query transmitted using the geo-routing method described in Section 3.1, the global coordinator floods a local query message to nodes within the estimated k NN circle. Then, each node that received the local query message stores the identifier of the source node as its *EXP parent* and sets the waiting time, WT , for sending a reply. Here, WT gets decreases as the distance between the node and the query point increases. When WT has passed, the node transmits a reply message attached with its information to its EXP parent. Finally, after collecting replies from nodes in the estimated k NN circle, the global coordinator replies with the k NN result to the query-issuing node.

Figure 1 shows an example of executing the EXP method where M_1 is the global coordinator. When M_5 receives a local query message from M_1 , it stores M_1 's ID as its EXP parent and broadcasts a local query message to its neighboring nodes because it is within the estimated k NN circle. In the same way, upon receiving the query message, M_6 stores M_5 's ID as its EXP parent and broadcasts a local query message to its neighboring nodes. On the other hand, upon receiving the query message, M_7 discards the message because it is not within the estimated k NN circle. When WT has passed at M_6 , M_6 transmits a reply message attached with M_6 's information to M_5 . M_5 that received the reply message from M_6 transmits a reply message attached with M_5 's and M_6 's information when WT has passed. Such procedures are performed in the entire MANET. Finally, M_1 acquires the information on all nodes within the estimated k NN circle. When WT has passed at M_1 , M_1 transmits the k NN result to the query-issuing node.

The EXP method can reduce the traffic for collecting the k NN result because a tree structure is dynamically constructed during transmissions of local query messages, and the information of nodes in only a specific region (the estimated k NN circle) is efficiently collected along the tree. In this method, determining the radius of the estimated k NN circle, R , is based on the density of nodes in the entire MANET. However, in a real environment, it is not always easy to know the total number of nodes in the entire MANET and the area size beforehand. Moreover, the density of nodes is generally not uniform in a MANET.

4. KNN query processing methods

In this section, first we describe the design policy of our proposed methods and the assumed environment. Then, we give an overview of our proposed methods. Finally, we describe in detail of how to process a k NN query with our methods.

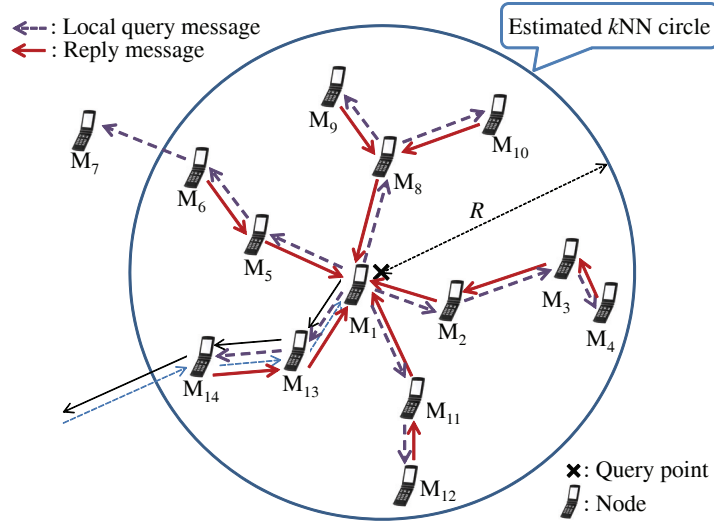


Fig. 1. EXP method.

4.1. Design policy

In MANETs, it is very important to reduce as much traffic as possible due to limitations of network bandwidth and battery of mobile nodes. If each node periodically broadcasts a beacon message even when no node searches k NNs in the network, this causes unnecessary traffic. A method without using beacon messages is suitable for MANETs. However, without exchanging beacon messages, each node cannot know its neighboring nodes' information beforehand. Therefore, we design our methods as beacon-less methods that perform on-demand search.

Moreover, in MANETs, the k nearest nodes from the query point change during the search because a node moves freely. Therefore, the query-issuing node should acquire the query result in a short time. For this aim, we design our methods to execute a query by just one round of message transmissions.

Since mobile nodes consume limited communication bandwidth for data transmission, packet loss and packet retransmission may occur when the network is congested, i.e., some information cannot be transmitted. Thus, the amount of information transmitted by each mobile node should be reduced as much as possible, so the estimated k NN circle should be appropriately set. When each node knows the total number of nodes in the entire network and the area size, and the density of nodes is uniform, the density of nodes in the entire network can be the optimal estimated k NN circle. However, these assumptions are not always true in a real environment. Therefore, we propose the methods without these assumptions.

4.2. Assumptions

The system environment is assumed to be a MANET in which all mobile nodes have the same radio communication facility, in which the communication range is a circle with a fixed size. We assume that the MANET is sufficiently dense so that network partitioning does not occur and geo-routing can be performed between any pair of nodes. In the MANET, mobile nodes retrieve the information on mobile nodes using k NN queries. The query-issuing node transmits a query message associated with the query point and acquires the information on the k nearest nodes from the query point among all nodes in the entire network.

We assign a unique *node identifier* to each mobile node in the system. The set of all mobile nodes in the system is denoted as $M = \{M_1, M_2, \dots, M_n\}$, where n is the total number of mobile nodes and M_i ($1 \leq i \leq n$) is a node identifier. Each mobile node moves freely. Every mobile node knows its current location by using positioning devices such as GPS.

4.3. Overview of our methods

To appropriately set the estimated k NN circle, the node should efficiently know the density of nodes near the query point because there should be k NNs near the query point. When the density of nodes is not uniform in the entire network, it is more effective to know the density of nodes near the query point than the average density of nodes in the entire network. However, it is costly to widely acquire the information on locations of many nodes to calculate the density of nodes. Therefore, the global coordinator first acquires only the number of nodes within its communication range (one-hop) by broadcasting the query to its neighbors. By doing so, the global coordinator can calculate the density of nodes near the query point.

According to the above policy, after a query is transmitted to the global coordinator using geo-routing with the OH method, the global coordinator acquires its neighbors' information (one-hop nodes' information) by exchanging messages to know the density of nodes near the query point. If the number of neighbors exceeds k , the global coordinator can reply with k NNs to the query-issuing node. If not, the global coordinator sets the radius of the estimated k NN circle based on the density of nodes within its communication range and acquires the information on nodes within the estimated k NN circle.

The problem in which a node can estimate the density of nodes only from its (one-hop) neighboring nodes is evident with the OH method. Thus, when k is large or the density of nodes is sparse, i.e., the range where k NNs exist is large, the accuracy of the estimated k NN circle, which is estimated by the information obtained from its neighboring nodes, is expected to decrease. If a node stores the information on the query result when receiving or relaying the query reply, it can use the information to know the density of nodes near the query point in a wider range than with the OH method. Since there are various k NN queries issued in the network, a node can widely determine the density of nodes in the entire network by storing the information on the query result. When a new query is issued, such information on the density of nodes can be used for determining the estimated k NN circle. That information can also be collected from multiple nodes that relay the query message during geo-routing, which is helpful to enhance the quality of the information on the node density. After receiving the query, the global coordinator can determine the radius of the estimated k NN circle based on the density information attached to the query.

According to the above policy, a node that relays a reply for a k NN query in the QL method stores the information on the query result as the *query log*, which includes the density of nodes around the query point, to use it for future queries. During query forwarding for a new query, the query-issuing and query-relaying nodes attach some of the stored information to the query message, which is used to estimate the density of nodes near the query point. Then, the global coordinator estimates the radius of the estimated k NN circle based on some of the attached information. The information on the node density attached to the query message is expected to be more accurate when its query point is closer to that of the current query, and when its query-issuing time is closer to that of the current query. Thus, the QL method takes this fact into account when determining the estimated k NN circle.

4.4. Forwarding *k*NN query and replying with result: One-Hop (OH) method

The behaviors of the query-issuing node, M_s , and mobile nodes that receive the query message are as follows. After step 6 except for step 10, each node mostly behaves in the same way as in the EXP method.

1. M_s specifies the requested number of *k*NNs, k , and the query point. Then, similar to the EXP method, M_s transmits a *k*NN query message to the global coordinator using the geo-routing method described in Section 3.1. In the query message, the query-issuing node's ID and location are respectively set as M_s and its location, the requested number of *k*NN is set as k , and the query point is set as the location specified by the query.
2. Through the procedures described in Section 3.1, the global coordinator, M_p , is selected. Then it broadcasts a *one-hop query message* to its neighboring mobile nodes. In the message, the global coordinator's ID and location are respectively set as M_p and its location.
3. Each mobile node, M_q , that received the one-hop query message replies with a *one-hop reply message* to M_p if the distance between the query point and M_q is shorter than the radius of the communication range. In the message, the source node's ID and location are respectively set as M_q and its location.
4. M_p that received the one-hop reply messages from its neighboring nodes stores the *tentative kNN result* by adding the information on nodes that replied with the one-hop reply messages. If the number of nodes included in the tentative *k*NN result exceeds k , the information on nodes that are not *k*NNs from the query point is removed from the tentative result, and the procedure continues to step 11.
5. M_p determines the radius of the estimated *k*NN circle, R , which contains *k*NNs with high probability, based on the tentative *k*NN result by the following equation:

$$R = \alpha \cdot l \cdot \sqrt{\frac{k}{n'}}. \quad (1)$$

For determining R for the first time, l is the radius of the communication range, n' is the number of nodes included in the tentative *k*NN result (including M_p), and α is a margin for safely setting the estimated *k*NN circle. After the second time, (returned from step 10), l' is the distance from the farthest node to the query point in the tentative *k*NN result and n' and α are same. As Eq. (1) shows, R is set based on the density of nodes acquired by the global coordinator.

6. M_p broadcasts a local query message to its neighboring mobile nodes. In the message, the requested number of *k*NNs is set as k , the radius of the estimated *k*NN circle is set as R , the query point is set as that in the received query message, and the global coordinator's ID and location are respectively set as M_p and its location.
7. Each mobile node, M_q , that received the local query message the first time stores the identifier of the source node as its *EXP parent*. If M_q is within the estimated *k*NN circle, it sets the waiting time, WT , for sending a reply by the following equation:

$$WT = \beta \cdot \left(\frac{R}{r} \right) \cdot \left(1 - \frac{a}{R+b} \right). \quad (2)$$

a is the distance between M_p and M_q , b is the distance between the query point and M_p , β is a parameter decided by a system designer to prevent collisions of messages, and r is the communication range. As Eq. (2) shows, WT decreases as the distance between M_p and M_q increases.

At the same time (without WT), M_q broadcasts a local query message to its neighboring mobile nodes.

If M_q has already received a local query message before or it is not within the estimated k NN circle, it discards the message and does nothing.

8. The node that has set the minimum WT starts to transmit a reply message (after WT) attached with the information on itself including its location to its EXP parent. This attached information is also called the tentative k NN result.
9. Each node that received the reply message (from its EXP child) updates the tentative k NN result attached in the reply message by adding the information on itself. If the number of nodes whose information is included in the tentative k NN result exceeds k , the information on the node which is the farthest from the query point is removed from the tentative k NN result.
When WT has passed, it transmits a reply message attached with the updated tentative k NN result to its EXP parent if it is not the global coordinator. Then, the procedure returns to step 8. Otherwise, if it is the global coordinator, the procedure continues to step 10.
10. When WT has passed, the global coordinator, M_p , behaves as follows. If the number of nodes included in the tentative k NN result exceeds k , or the times of retransmitting the local query exceeds T , the procedure goes to step 11. Otherwise, the procedure returns to step 5, i.e., the global coordinator re-estimates R and re-does the same process.
11. M_p replies with the tentative k NN result as the final result to the query-issuing node using the geo-routing method described in Section 3.1, where the query point is set as the location of the query-issuing node. If M_p or a relaying node incidentally knows a node on the query path from the query-issuing node to the global coordinator, it forwards the k NN result to the node and the k NN result is sent back to the query-issuing node along the query path. If some nodes along the query path do not connect with their parents due to link disconnection, they again transmit the k NN result using the geo-routing method.

By using this method, the radius of the estimated k NN circle can be appropriately estimated because it is calculated based on the density of nodes near the query point. Therefore, unnecessary transmissions of queries and replies can be suppressed.

4.5. Forwarding k NN query and replying with result: Query Log (QL) method

The behaviors of the query-issuing node, M_s , and mobile nodes that receive the query message are as follows.

1. M_s specifies the requested number of k NNs, k , and the query point. If M_s has some query logs, it attaches the query logs (L) that satisfy the following condition:

$$\forall i \{L.t \leq L(i).t \cup L.d \leq L(i).d\}. \quad (3)$$

Here, $L(i)$ is the i -th query log, $L(i).t$ is the time interval between the query-issuing time of the current query and that for $L(i)$, and $L(i).d$ is the distance between the query point of the current query and that for $L(i)$. This condition shows that only query logs of queries issued recently and near the query point of the current query can be used to determine the estimated k NN circle. Because all query logs under this condition are attached to the query, they can be used to select query logs for estimating the radius of the estimated k NN circle. Although the size of the query message is slightly large since some query logs are attached, this condition can prevent many query logs from being attached.

M_s transmits a k NN query message to the global coordinator using the geo-routing method described in Section 3.1. In the query message, the query-issuing node's ID and location are respectively set as M_s and its location, the requested number of k NN is set as k , the query point is set

as the location specified by the query, and the query log list is set as the list of query logs each of which contains the information on the query point, the query-issuing time, the requested number of *k*NN, *k_past*, and the distance from the query point to the *k_past*-th nearest node of the corresponding query.

2. During geo-routing, if M_t that received the query message has some query logs, it adds some of the query logs to the query log list in the query message, which are chosen by using the method described in step 1, and broadcasts the message.
3. Through the process described in Section 3.1, the global coordinator, M_p , is selected. If the query log list in the received query message is empty and M_p does not store any query logs that satisfy condition (3), it performs steps 2 to 9 in Section 4.4, and the procedure continues to step 4. Otherwise, M_p determines the radius of the estimated *k*NN circle by using query logs in the query log list and that are stored on M_p . First, the estimated *k*NN circle, R , for each of the query logs that satisfy condition (3) is calculated by the following equation:

$$R' = l' \cdot \sqrt{\frac{k}{k_past}}. \quad (4)$$

$$R = \sqrt{R'^2 + \frac{d}{\gamma} + \frac{t}{\theta}}. \quad (5)$$

We call the query corresponding to the query log *previous query*. *k_past* is the requested number of *k*NNs in the previous query, l' is the distance from the query point to the *k_past*-th nearest node in the previous query, d is the distance between the query point of the current query and that of the previous query, t is the time interval between the query-issuing time of the current query and that of the previous query, and γ and θ are weighting parameters to adjust the impact of d and t . R' in Eq. (4) is set based on the density of nodes in the query log. When the query point of the previous query is farther than that of the current query, and when the query-issuing time of the previous query is older, as shown in Eq. (5), R increases. This is effective in preventing the accuracy of the query result from decreasing by an error estimation of the estimated *k*NN circle.

Then, the query log that has the smallest R among all the query logs is selected, and its R becomes the radius of the estimated *k*NN circle. Then, steps 6 to 9 in Section 4.4 are performed, and the procedure continues to step 4.

4. When *WT* has passed, the nearest node from the query point, M_p (the global coordinator) behaves as follows. If the number of nodes included in the tentative *k*NN result exceeds k , or the times of retransmitting the local query exceeds T , the procedure continues to step 5. Otherwise, it performs steps 5 to 9 in Section 4.4 again, i.e., the global coordinator re-estimates R and re-does the same process. Here, l' in step 5 is set to R determined in step 3.
5. M_p replies with the tentative *k*NN result as the final result to the query-issuing node using the geo-routing method (same as step 11 in Section 4.4).

During geo-routing, M_s , M_p , and the relaying nodes store the information on the query result as a query log, which contains the query point, the query-issuing time, the requested number of *k*NNs, k , and the distance from the query point to the k -th nearest node.

Since the QL method determines the radius of the estimated *k*NN circle using the stored information on previous queries, it does not require extra message exchanges. Moreover, it can estimate the node density in a wider area than the OH method, which estimate the node density based on the number of one-hop neighbors. To reduce errors in estimation, the QL method gives higher priority to queries in the query log list that are newer and specify the query point closer to that of the current query.

5. Simulation experiments

In this section, we explain the results of simulation experiments regarding the performance evaluation of our proposed methods. For the simulation experiments, we used the network simulator Qual-Net5.2 [22].

5.1. Simulation model

The number of mobile nodes in the entire system is 400 (except in Section 5.7 setting on the number of nodes as 800 nodes). These mobile nodes exist in an area of $800 \times 800 \text{ m}^2$ and their initial positions are randomly selected. These nodes move according to the random walk model [3] where nodes select a random direction and random speed from 0.5 to 1.0 m/sec every minute.¹ We also conducted simulations with other mobility models: the random waypoint and random waypoint models with a home area. In the latter model, the entire area is partitioned into four square regions of equal size, and each node selects its next destination either from the region in which it resides (90% probability) or from another region (10%). The results show that our proposed methods achieve roughly the same performance in all the three mobility models, and the differences in performance between our methods and comparative methods are almost same in the three mobility models. Thus, we only show here the results with the random walk model.

Each mobile node transmits messages using an IEEE 802.11b device whose data transmission rate is 11 Mbps. The transmission power of each mobile node is determined so that the radio communication range becomes about 100 m. Packet losses and delays occur due to radio interference. We assume that each node knows its current location. The query point specified by a k NN query is randomly selected within the entire area, and α in Eq. (1), β in Eq. (2), γ in Eq. (5), and θ in Eq. (5) are respectively set to 1, 1, 0.01 and 10 based on our preliminary experiments. The requested number of k NNs, k , is randomly selected from 1 to 100 by the query-issuing node.

We compare the performance of our proposed methods with that of two different methods. The first method is the EXP method [12]. We assume that each node can know the total number of nodes, n , and the area size. Thus, in the EXP method, R (radius of the estimated k NN circle) is determined by the following equation:

$$R = \sqrt{\frac{k \cdot \text{area}}{\pi \cdot n}}. \quad (6)$$

area is the area size ($\text{area} = 800 \times 800$) and n is the total number of nodes. We adopt three different values of n for the EXP method, $n = 400, 200$, and 800 . It should be noted that the real number of nodes in the simulations is 400 as described above. Here, we assume that each node in the EXP method misunderstands the number of nodes when $n = 200$ and 800 . These two cases respectively represent situations in which nodes overestimate and underestimate the estimated k NN circle because of the difference in node density between the entire area and the target region of k NN queries. From this, we can verify the impact of misestimation of node density in the EXP method. In the graphs of the experimental results, we show the results when $n = 400, 200$, and 800 as “EXP (just)”, “EXP (over)”, and “EXP (under)”, respectively. The other method for comparison is called the “optimal method” (denoted as “Optimal” in the graphs). In this method, we assume that the global coordinator completely knows

¹We change the simulation setting on initial positions and movement of nodes in Section 5.5.

Table 1
Message types and sizes

Type	Size [B]
Neighbor searching (geo-routing)	48
Reply (geo-routing)	8
Query (OH and EXP methods)	56
Query (QL method)	$64+32p$
One-hop query	32
One-hop reply	32
Local query	64
Local reply	$16+16q$
Reply (OH and EXP methods)	$32+16q$
Reply (QL method)	$60+16q$
Ack to a received reply	16

the position of the k -th nearest node from the query point; thus, it can set the optimal search range (i.e., estimated k NN circle) to acquire k NNs with the smallest traffic. After setting the search range, it performs in the same way as the EXP method. Of course, this is an ideal method, which cannot be implemented in reality. We show the performance of this method as the upper-bound of performance, which is just a guideline.

After one minute passes since the simulation started, the query-issuing node is randomly chosen among all nodes and it issues a k NN query. We repeat this process 1,000 times (i.e., 1,000 queries) every 20 seconds and evaluate the following three criteria.

– Traffic

We examine the total volume of query messages and replies exchanged for processing a query. Table 1 lists the size of each message used in our methods and the comparative methods, where p denotes the number of query logs attached to a query and q denotes the number of nodes whose information is included in the reply. We define “traffic” as the average total volumes for all queries issued.

– Response time

We examine the time from transmitting a query message by the query-issuing node until receiving the k NN result. We define “response time” as the average times for all queries issued.

– Accuracy of query result

We examine the ratio of the number of k NNs whose information is included in the k NN result acquired by the query-issuing node to the requested number of k NNs, k . We define “accuracy of query result” as the mean average precision (MAP) value which measures the performance of the result with a ranking [18]. MAP is an average of the average precision (AP) for each query. AP and MAP are determined by the following equations.

$$AP_i = \frac{1}{k} \sum_{j=1}^k \frac{g}{j} \cdot e \quad (7)$$

$$MAP = \frac{1}{querynum} \sum_{i=1}^{querynum} AP_i \quad (8)$$

AP_i is the AP on the i -th issued query, g is the number of nodes that are included in the query result among the top- j nearest nodes, $querynum$ is the total number of issued queries (i.e., 1,000 in this simulation), and e is determined by the following equation:

$$e = \begin{cases} 1 & (j\text{-th nearest node is included in } k\text{NN result}). \\ 0 & (\text{otherwise}). \end{cases} \quad (9)$$

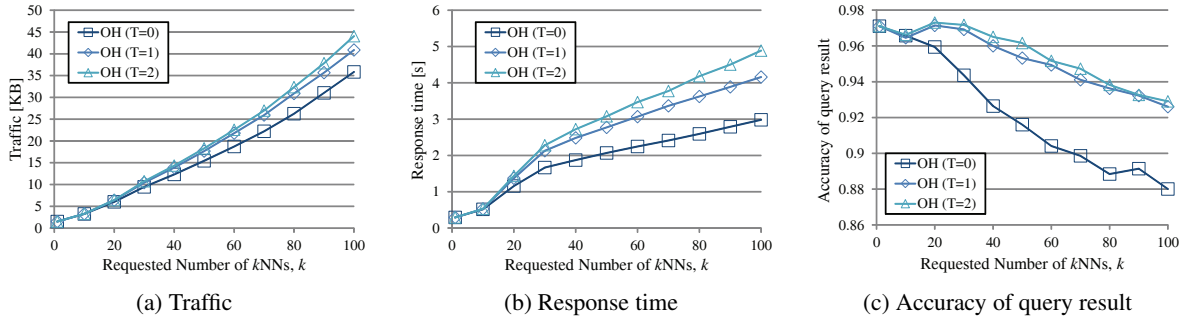


Fig. 2. Impact of number of re-estimations in OH method.

Thus, MAP becomes higher as the query-issuing node obtains the information on nodes closer to the query point.

5.2. Impact of number of re-estimations of R

In our methods, the number of re-estimations of R occurring in processing a k NN query affects the performance. Therefore, we first explain the results of simulations where the maximum number of re-estimations of R (denoted as T) is set to 0 (no re-estimation), 1 (up to one re-estimation) and 2 (up to two re-estimations).

5.2.1. OH method

First, we examine the performance of the OH method when varying the number of re-estimations of R . Figure 2 shows the simulation results. In the graphs, the horizontal axis indicates the requested number of k NNs, k , and the vertical axes indicate the traffic in Fig. 2(a), the response time in Fig. 2(b), and the accuracy of query result in Fig. 2(c).

From Fig. 2(a), when k is smaller than 10, the traffic does not increase rapidly as k increases. This is because the global coordinator can acquire the information on more than k nodes by one-hop replies since it has at least 10 neighboring nodes. This fact can be confirmed from the results where there are no differences in traffic when $T = 0, 1$, and 2 . When k is larger than 20, the traffic for $T = 1$ and $T = 2$ is much larger than that for $T = 0$. This is because retransmissions of queries and replies occur due to misestimation of R , which increase traffic. This suggests a disadvantage with the OH method, which estimates the density of nodes only from the information on one-hop neighbors. When k is large, the search range increases (larger than the communication range); thus, errors in estimation also increase. The traffic for $T = 1$ and $T = 2$ is almost the same, which shows that k NNs can be acquired by re-estimating R only once in most cases.

From Fig. 2(b), when k is smaller than 10, the response time is very short in all cases. This is because the global coordinator can acquire the information on more than k nodes by one-hop replies as mentioned. Moreover, the response times for $T = 1$ and $T = 2$ are longer than that for $T = 0$ because it takes time to acquire the information on remaining k NNs after performing re-estimations of R .

From Fig. 2(c), the OH method where $T = 0$ can maintain high accuracy of the query result when k is small. However, the accuracy decreases as k increases. This is because, as mentioned above, errors in estimation of the estimated k NN circle increase as k increases, and the search range increases. In the OH method, where $T = 1$ and $T = 2$, the accuracy of query result remains high regardless of k (better than that for $T = 0$). This shows the effectiveness of re-estimating R . The accuracy of query result is

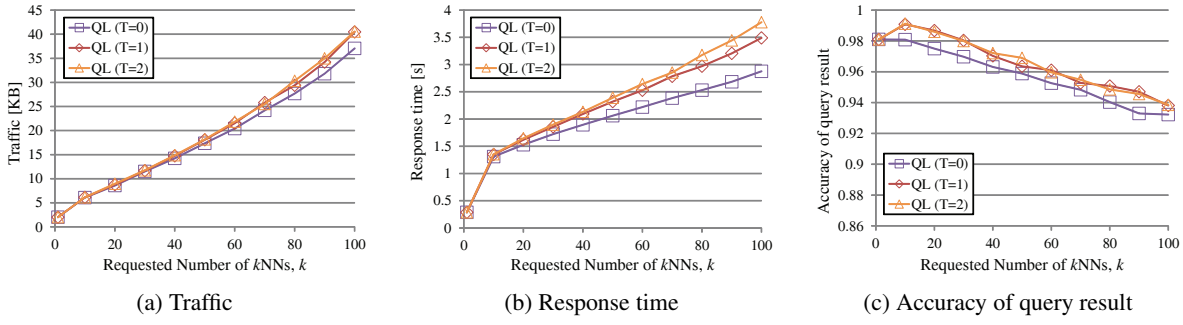
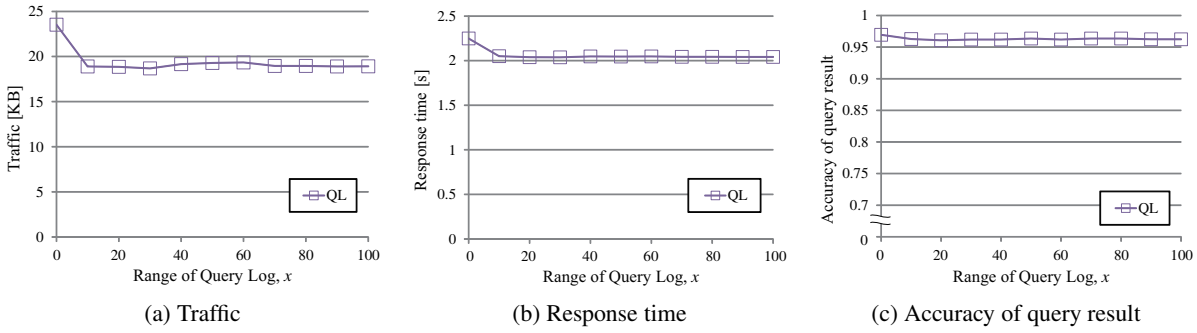


Fig. 3. Impact of number of re-estimations in QL method.

Fig. 4. Impact of k_{past} in query logs.

almost the same for $T = 1$ and $T = 2$. Since the traffic and the response time increase as T increase, we can conclude that the number of re-estimations of R should be at most once.

5.2.2. QL method

We also examine the performance of the QL method when varying the number of re-estimations of R . Figure 3 shows the simulation results. In the graphs, the horizontal axis indicates the requested number of k NNs, k , and the vertical axes indicate the traffic in Fig. 3(a), the response time in Fig. 3(b), and the accuracy of query result in Fig. 3(c).

From Fig. 3(a), the traffic is almost the same when $T = 0, 1$, and 2 . From Fig. 3(b), the response times for $T = 1$ and 2 are longer than that for $T = 0$. This is due to the same reason as that in Section 5.2. However, the differences in response time are smaller than the result in Fig. 2(b).

From Fig. 3(c), the QL method, in which $T = 0$, can maintain high accuracy of query result even when k is large (unlike the OH method). This shows the advantage of the QL method in using the information on node density in a wider area than the OH method, obtained from query logs. The accuracy of query result decreases as k becomes larger. This is because packet losses often occur due to message collisions when the search range increases. In the QL method where $T = 1$ and $T = 2$, the accuracy of query result remains high, which is similar to the case of the OH method. This shows the effectiveness of re-estimating R . The accuracy of query result is almost same for $T = 1$ and $T = 2$. Since the traffic and the response time increase as T increases, we can conclude that the number of re-estimations of R should be at most once in the QL method, similar to the OH method.

Table 2
Number of queries using query logs

x	0	10	20	30	40	50	60	70	80	90	100
# of queries using query logs	402	933	953	961	961	961	962	963	963	964	964

5.3. Impact of k_past in selected query logs

In the QL method, the global coordinator estimates the search range based on query logs attached to a query, which are selected among query logs stored on nodes through which the query is transmitted. This makes the global coordinator know the density of nodes in a wider area with a low traffic. Here, even if two queries which respectively specify the same query point are issued at the same time, the densities of nodes stored in their query logs are basically not same when k (k_past in query logs) specified by the two queries is different. This is because ranges where the k_past nearest neighbors exist are different.

Therefore, in the QL method, k_past in query logs selected for estimation of the k NN circle affects the performance, and we examine the impacts of k_past in selected query logs. For this aim, we conducted a simulation where nodes transmitting a query select only query logs among those with k_past within “the current $k \pm x$ ”. The range of query logs, x , is varied from 0 to 100 in this simulation; when x is 0, query logs are selected among those with the same k_past as the current query, and when x is 100, query logs are selected among all query logs.

Figure 4 and Table 2 show the simulation results. In the graphs, the horizontal axis indicates the range of query logs, x , and the vertical axes indicate the traffic in Fig. 4(a), the response time in Fig. 4(b), and the accuracy of query result in Fig. 4(c). In Table 2, each number in the lower column indicate the number of queries which are processed by using query logs for k NN circle estimation out of 1,000 queries. In this subsection, the accuracy of query result is calculated only for queries processed by using query logs for k NN circle estimation.

From Figs 4(a) and 4(b), the traffic and response time are large when x is 0. This is because the global coordinator often overestimates the search range since it estimates it based on the density of nodes in a few query logs whose k_past is the same as that of the current query. More specifically, since it does not often happen that the query point and query-issuing time of the attached query logs are close to that of the current, the search range tends to be estimated largely for safety especially for the case of $x = 0$, where only a few query logs are available. Table 2 shows that when x is 0, only less than half of all queries (402) are processed by using query logs. This is because it often happens that nodes transmitting a query do not store any query logs whose k_past is the same as that of the current query. Except for the case of $x = 0$, the traffic and response time are nearly constant. This is because the global coordinator can estimate the search range using enough number of query logs in most cases.

From Fig. 4(c), the accuracy of query result is slightly higher when x is 0. This is because the search range tends to be estimated largely for safety. However, improvement of accuracy of query result is less than 1% while the traffic and response time increase. Therefore, we can conclude that the performance of the QL method is not sensitive to x (except for the case of $x = 0$). Based on this, in all simulations, nodes use all query logs (i.e., $x = 100$) in the QL method.

5.4. Impact of requested number of nodes, k

Next, we compare our proposed methods with the optimal and the EXP methods. Figure 5 shows the simulation results. In the graphs, the horizontal axis indicates the requested number of k NNs, k , and the vertical axes indicate the traffic in Fig. 5(a), the response time in Fig. 5(b), and the accuracy of query result in Fig. 5(c). In our proposed methods; the OH method and the QL method, T is set to 0.

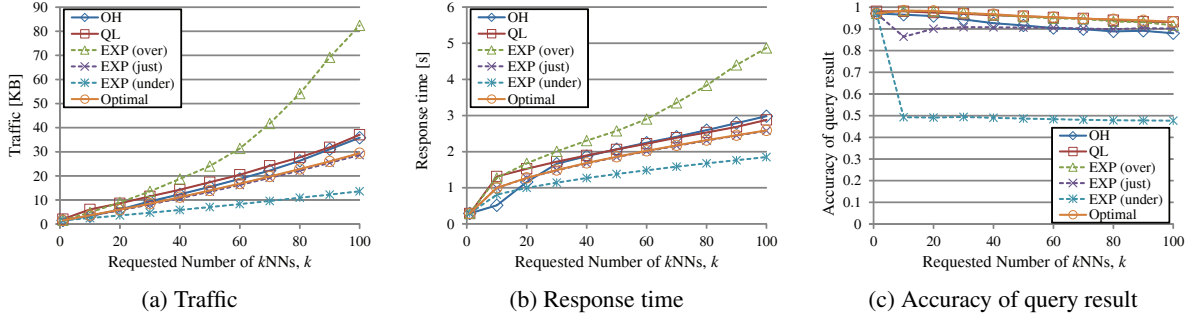
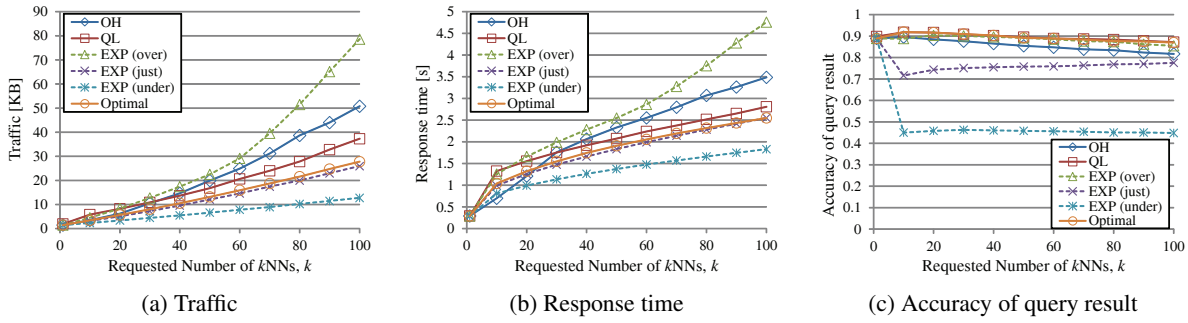
From Fig. 5(a), as k increases, the traffic increases in all methods. This is because the searching area for processing a k NN query and the data volume of the reply increase. The traffic in the EXP method (just) is almost same as that in the optimal method. This is because the estimated k NN circle is appropriately set when the density of nodes near the query point is the same as that in the entire area. However, in the EXP method (over), the traffic becomes higher due to the overestimation of the k NN circle, i.e., the search range is too large. On the other hand, in the EXP method (under), the traffic is very small due to the underestimation of the estimated k NN circle, which can be seen from the result in Fig. 5(c) where the information on only about half of k NNs is acquired. This result suggests that the EXP method cannot appropriately estimate the search range (the estimated k NN circle) when the density of nodes is different from that near the query point. In the OH method, the traffic is slightly larger than that in the optimal method. This is because extra message exchanges are necessary to acquire the neighboring nodes' information. The traffic in the QL method is also slightly larger than that in the optimal method. This is because the estimated k NN circle is sometimes set to much larger than the optimal range for safety (e.g., when new query logs for queries issued near the query point cannot be found).

From Fig. 5(b), the response time in all methods increases as k increases. This is because in all methods, the waiting time, WT , increases as the estimated k NN circle increases. In particular, since the EXP method (over) overestimates the search range, it sets a longer WT than other methods. Meanwhile, the EXP method (under) gives the shortest response time since the search range is the smallest, i.e., WT is the smallest. In the OH method, the response time is very short when k is smaller than 20. This is because the global coordinator can acquire the information on about 20 nodes by one-hop replies. When k is large, the response time of the OH method is longer than that in the optimal method. This is because the OH method requires at least two rounds of message exchanges; (i) to acquire the neighboring nodes' information and (ii) to acquire the information on k NNs. The response time in the QL method is slightly longer than that in the optimal method. This is because the estimated k NN circle is sometimes set to much larger, as mentioned above.

From Fig. 5(c), the accuracy of query result is very high (nearly 1) in the QL, EXP (over), and optimal methods. This suggests that the estimated k NN circle can be appropriately set in the QL method. In the optimal method (though the estimated k NN circle is optimally set), the accuracy of query result slightly decrease as k increases, which also occurs in our proposed methods and EXP method (over). This is because many replies are sent back to the query-issuing node and collisions of messages often occur. In the OH method, the accuracy of query result slightly decreases, but it can remain high with low traffic by re-estimating R , as described in Section 5.2. On the other hand, the EXP method (just) gives lower accuracy of query result. This is because R is sometimes set smaller than the optimal one when the density of nodes near the query point is incidentally lower than that in the entire area. The accuracy of query result in the EXP method (under) is much lower than other methods due to the underestimation of R .

5.5. Impact of requested number of nodes, k , in skewed network

Finally, we change the simulation setting on distribution and movement of nodes and compare our proposed methods with the optimal and EXP methods. We aim to examine the impact of skewed node density on our methods. Specifically, the simulation area is partitioned into four square sub-areas with the same size and 50, 150, 50, and 150 nodes are deployed in top-left, top-right, bottom-left, and bottom-right sub-areas, respectively. Nodes move according to the random walk model [3] in their own sub-areas, where each node selects a random direction and random speed from 0.5 to 1.0 m/sec every minute.

Fig. 5. Impact of requested number of *k*NNs, *k*.Fig. 6. Impact of requested number of *k*NNs, *k*, in skewed network.

This represents a situation where sub-areas with 150 nodes (dense sub-areas, $150/(400 \times 400) 1/m^2$) are much more dense than that with 50 nodes (sparse sub-areas, $50/(400 \times 400) 1/m^2$). In our proposed methods, T is set to 0, and γ and θ in Eq. (5) are respectively set to 0.1 and 1 based on our preliminary experiments. Figure 6 shows the simulation results. In the graphs, the horizontal axis indicates the requested number of *k*NNs, *k*, and the vertical axes indicate the traffic in Fig. 6(a), the response time in Fig. 6(b), and the accuracy of query result in Fig. 6(c).

From Fig. 6(a), in the OH method, the traffic is much larger than that in Fig. 5(a). This is because the estimated *k*NN circle is set larger than necessary due to an error in estimation when the query point is set as a point in a sparse sub-area, which is near the border of a dense sub-area (i.e., the OH method does not take into account the density of nodes in a dense area). On the other hand, the traffic in the QL method is almost the same as that in Fig. 5(a). This shows that the QL method can set the estimated *k*NN circle appropriately even in a skewed network where the density of nodes is not uniform.

From Fig. 6(b), in the OH method, the response time is longer than that in Fig. 5(b) when *k* is large. This is because the waiting time, *WT*, increases as the estimated *k*NN circle increases.

From Fig. 6(c), the accuracy of query result is lower than that in Fig. 5(c) in all methods because geo-routing sometimes does not work well in a sparse area. More specifically, a node that relays a message sometimes cannot find any nodes closer to the query point than itself. In the OH method, the accuracy of query result remains high, while it produces much traffic, as described above. On the other hand, in the QL method, the accuracy of query result remains high, while the traffic is not so large. From these facts, we can confirm that using query logs is effective in estimating the estimated *k*NN circle in a network where the density of nodes is not uniform.

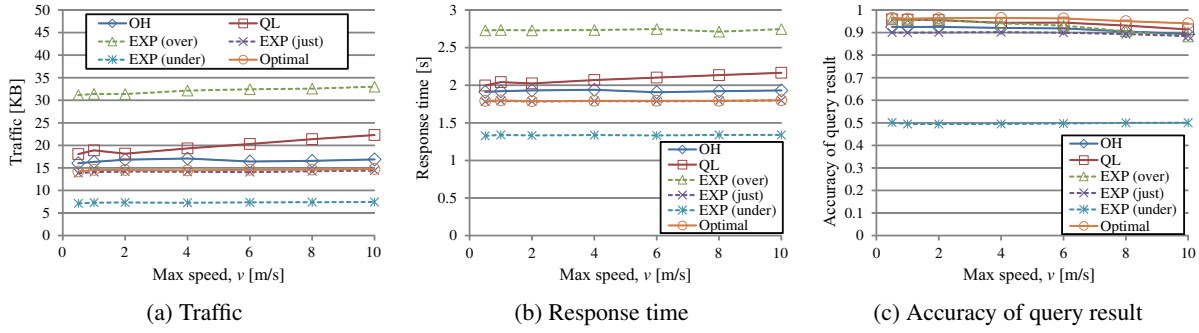


Fig. 7. Impact of node speed.

5.6. Impact of node speed, v

We vary the maximum speed of nodes to examine the impact of node speed. Figure 7 shows the simulation results. In the graphs, the horizontal axis indicates the maximum speed of nodes, v , and the vertical axes indicate the traffic in Fig. 7(a), the response time in Fig. 7(b), and the accuracy of query result in Fig. 7(c). In our proposed methods; the OH method and the QL method, T is set to 0.

From Figs 7(a) and 7(b), in the QL method, the traffic and response time slightly increase as v increases. This is because the search range tends to be overestimated more since the density of nodes more dynamically changes as the node speed increases. In the other methods, the traffic and response time are nearly constant, regardless of nodes' speed. This shows the size of the search range is almost same even if nodes move faster.

From Fig. 7(c), the accuracy of query result slightly decreases due to more packet losses. However, even if the max speed is 10 m/s, our methods still keep high accuracy of query result.

5.7. How to choose a method

As shown above, our proposed methods, the OH and QL methods, show different performance in different situations, e.g., k and the density of nodes, and thus, which is the best method changes depending on a situation. More specifically, when k is small and the density of nodes is high, the OH method outperforms the QL method. This is because the estimation based on the density of the global coordinator's neighboring nodes works well since the area where k nearest nodes exist is relatively small. In addition, since more than k nodes are often neighbors of the global coordinator, the global coordinator can acquire the information on k nearest nodes without setting the search range in the OH method. On the other hand, when k is large and the density of nodes is low, the QL method outperforms because it is useful to estimate the search range based on the density of nodes in a wider area. Therefore, the global coordinator can select either the OH method or the QL method based on the size of k or the density of nodes in order to effectively process a query.

To examine how to partially achieve this, we conducted an experiment where we introduce a new system parameter *switching- k* as the border line of selecting the two methods. If the specified k is less than *switching- k* , the OH method is chosen, i.e., the global coordinator estimates the search range based on the density of the global coordinator's neighbors. Otherwise, the QL method is chosen, i.e., the global coordinator estimates the search range based on query logs.

Only in this experiment, we set the number of nodes as 800 because it shows clearer characteristics than other setting. Figure 8 shows the simulation results. In the graphs, the horizontal axis indicates the

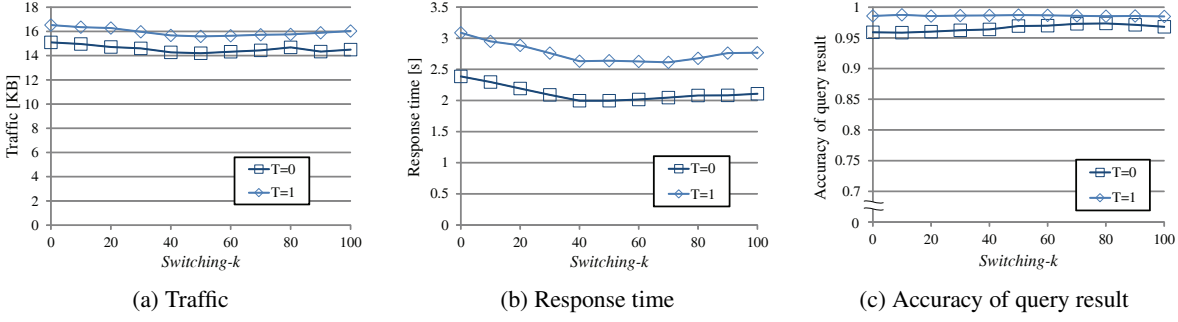


Fig. 8. Impact of method selection.

switching-k, and the vertical axes indicate the traffic in Fig. 8(a), the response time in Fig. 8(b), and the accuracy of query result in Fig. 8(c). In our proposed methods, T is set to 0 and 1.

From Figs 8(a) and 8(b), the traffic and response time are small when *switching-k* is around 50. This shows that it is somewhat effective to switch between the OH and QL methods based on the value of k . The traffic and response time slightly increase as T increases as described in Section 5.2. Here, the traffic and response time show almost the same tendencies in both cases of T ($T = 0$ and 1), i.e., the optimal value of *switching-k* is same regardless of T .

From Fig. 8(c), in the case of $T = 0$, the accuracy of query result is the highest when the *switching-k* is 80. When the *switching-k* is small, the accuracy of query result decreases because the search range is sometimes underestimated based on inaccurate past information on density of nodes in the QL method. This shows that the OH method should be selected when k is small, i.e., the estimation based on the density of the global coordinator's neighbors works well. When the *switching-k* is significantly large, the accuracy of query result also decreases. This is because when k is large, the density of nodes near the global coordinator is no more reliable for estimation, i.e., the QL method outperforms the OH method. The accuracy of query result where $T = 1$ is higher than that where $T = 0$, and is nearly constant regardless of *switching-k*.

In summary, to efficiently reduce the traffic and response time in this simulation setting (e.g., the number of nodes is 800), it is effective *switching-k* is set to 50 and the global coordinator re-estimates the search range once if needed.

6. Conclusions

In this paper, we proposed two k NN query methods; the One-Hop (OH) method and the Query Log (QL) method, for reducing traffic and also maintaining high accuracy of the query result in MANETs, assuming that the density of nodes is not always uniform. In the OH method, the global coordinator acquires its neighbors' information (only one-hop nodes' information) by exchanging messages to know the density of nodes near the query point. If the number of neighbors exceeds k , the global coordinator can reply with the information on k NNs to the query-issuing node. If not, the global coordinator sets the radius of the estimated k NN circle based on the density of nodes within its communication range and acquires the information on nodes within the estimated k NN circle. In the QL method, a node which relays a reply for a k NN query stores the information on the query result to use it for determining the estimated k NN circle for future queries. During query forwarding, the query-issuing and query-relaying nodes attach some of the stored information to the query, which is used to estimate the density of nodes

near the query point. Then, the global coordinator estimates the radius of the estimated k NN circle using some of the attached information and acquires the information on nodes within the estimated k NN circle. These methods can set the size of the estimated k NN circle more appropriately using the information that is acquired during the query execution even if each node cannot know the information on the area size and total number of nodes beforehand, and the density of nodes in the entire network is not uniform.

The experimental results show that our proposed methods produce similar traffic for processing k NN queries as the optimal method and also achieve high accuracy of the query result. The EXP method, which calculates the estimated k NN circle based on the average density of nodes, sometimes cannot appropriately set the estimated k NN circle since the density of nodes near the query point is not always the same as that in the entire area. In the OH method, when k is small, the estimated k NN circle can be appropriately set; however, when k is large, i.e., the search range is large, the accuracy of the estimated k NN circle decreases. In the QL method, the accuracy of the query result sometimes decreases because the QL method estimates R based on the density of nodes using past query logs, which contain some errors.

We also assumed that k NNs are the k nearest nodes from the query point (i.e., nodes are the targets of search). We plan to extend our proposed methods to search general objects associated with locations (e.g., location-based data).

Acknowledgment

This research is partially supported by the Grant-in-Aid for Scientific Research (S) (21220002), (B) (24300037), and (A) (26240013) of MEXT, Japan.

References

- [1] D.J. Baker, J. Wieselthier and A. Ephremides, A Distributed Algorithm for Scheduling the Activation of Links in a Self-Organizing, Mobile, Radio Network, *Proc. IEEE ICC*, 1982, pp. 2F6.1–2F6.5.
- [2] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu and J. Jetcheva, A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *Proc. Int. Conf. on MobiCom*, 1998, pp. 159–164.
- [3] T. Camp, J. Boleng and V. Davies, A Survey of Mobility Models for Ad Hoc Network Research, *Wireless Communications and Mobile Computing* **2**(5) (2002), 483–502.
- [4] M. Demirbas and H. Ferhatosmanoglu, Peer-to-Peer Spatial Queries in Sensor Networks, *Proc. Int. Conf. on Peer-to-Peer Computing* (2003), 32–39.
- [5] T.Y. Fu, W.C. Peng and W.C. Lee, Parallelizing Itinerary-Based KNN Query Processing in Wireless Sensor Networks, *IEEE Transactions on Knowledge and Data Engineering* **22**(5) (2010), 711–729.
- [6] Y. Gao, B. Zheng, G. Chen and Q. Li, Algorithms for Constrained K -Nearest Neighbor Queries over Moving Object Trajectories, *GeoInformatica* **14**(2) (2010), 241–276.
- [7] T. Hara and S.K. Madria, Consistency Management Strategies for Data Replication in Mobile Ad Hoc Networks, *IEEE Transactions on Mobile Computing* **8**(7) (2009), 950–967.
- [8] M. Heissenbüttel, T. Braun, T. Bernoulli and M. Walchli, BLR: Beacon-Less Routing Algorithm for Mobile Ad-Hoc Networks, *Computer Communications* **27**(11) (2004), 1076–1086.
- [9] P.P. Jayaraman, A. Zaslavsky and J. Delsing, Cost-Efficient Data Collection Approach Using K -Nearest Neighbors in a 3D Sensor Network, *Proc. Int. Conf. on Mobile Data Management*, 2010, pp. 183–188.
- [10] D.B. Johnson, Routing in Ad Hoc Networks of Mobile Hosts, *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, 1994, pp. 158–163.
- [11] B. Karp and H.T. Kung, GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, *Proc. Int. Conf. on MobiCom'00*, 2000, pp. 243–254.
- [12] Y. Komai, Y. Sasaki, T. Hara and S. Nishio, A k NN Query Processing Method in Mobile Ad Hoc Networks, *Proc. Int. Conf. on Mobile Data Management*, 2011, pp. 287–288.

- [13] F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger, Geometric Ad-Hoc Routing: Of Theory and Practice, *Proc. Symposium on PODC*, 2003, pp. 63–72.
- [14] W.-C. Lee and B. Zheng, DSI: A Fully Distributed Spatial Index for Location-Based Wireless Broadcast Services, *Proc. Int. Conf. on Distributed Computing Systems*, 2005, pp. 349–358.
- [15] D.L. Lee, M. Zhu and H. Hu, When Location-Based Services Meet Databases, *Mobile Information Systems* **1**(2) (2005), 81–90.
- [16] J. Li, J. Jannotti, D.S.J.D. Couto, D.R. Karger and R. Morris, A Scalable Location Service for Geographic Ad Hoc Routing, *Proc. Int. Conf. on MobiCom*, 2000, pp. 120–130.
- [17] B. Liu, W.-C. Lee and D.L. Lee, Distributed Caching of Multi-Dimensional Data in Mobile Environments, *Proc. Int. Conf. on Mobile Data Management*, 2005, pp. 229–233.
- [18] C.D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, *Cambridge University Press*, 2008.
- [19] C.E. Perkins and E.M. Royer, Ad Hoc On Demand Distance Vector Routing, *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.
- [20] N. Roussopoulos, S. Kelley and F. Vincent, Nearest Neighbor Queries, *Proc. ACM SIGMOD*, 1995, pp. 71–79.
- [21] M. Safar, K Nearest Neighbor Search in Navigation Systems, *Mobile Information Systems* **1**(3) (2005), 207–224.
- [22] Scalable Network Technologies: “Qualnet,” <http://www.scalable-networks.com/>.
- [23] S.H. Wu, K.T. Chuang, C.M. Chen and M.S. Chen, Toward the Optimal Itinerary-Based KNN Query Processing in Mobile Sensor Networks, *IEEE Transactions on Knowledge and Data Engineering* **20**(12) (2008), 1655–1668.
- [24] Y. Xu, T.Y. Fu, W.C. Lee and J. Winter, Processing *K* Nearest Neighbor Queries in Location-Aware Sensor Networks, *Signal Processing* **87**(12) (2007), 2861–2881.
- [25] Y. Xu, J. Heidemann and D. Estrin, Geography-Informed Energy Conservation for Ad Hoc Routing, *Proc. Int. Conf. on MobiCom*, 2001, pp. 70–84.
- [26] B. Xu, F. Vafaei and O. Wolfson, In-Network Query Processing in Mobile P2P Databases, *Proc. Int. Symposium on Advances in Geographic Information Systems*, 2009, pp. 207–216.
- [27] Y. Yao, X. Tang and E.-P. Lim, Localized Monitoring of *k*NN Queries in Wireless Sensor Networks, *VLDB Journal* **18**(1) (2009), 99–117.
- [28] C. Yu, B.C. Ooi, K.-L. Tan and H.V. Jagadish, Indexing the Distance: An Efficient Method to KNN processing, *Proc. Int. Conf. on VLDB*, 2001, pp. 421–430.

Yuka Komai received the B.E. degree in the Multimedia Engineering and the M.E. degree in the Information Science and Technology from Osaka University, Osaka, Japan, in 2011 and 2013, respectively. Currently, she is a Ph.D. candidate in the Information Science and Technology from Osaka University, Osaka, Japan. Her research interests include distributed databases, mobile networks, and mobile computing systems.

Yuya Sasaki received the B.E. degree in the Multimedia Engineering and the M.E. degree in the Information Science and Technology from Osaka University, Osaka, Japan, in 2009 and 2011, respectively. Currently, he is a Ph.D. candidate in the Information Science and Technology from Osaka University, Osaka, Japan. His research interests include data search and replication mechanisms in mobile computing environments.

Takahiro Hara received the B.E., M.E., and Dr.E. degrees in Information Systems Engineering from Osaka University, Osaka, Japan, in 1995, 1997, and 2000, respectively. Currently, he is an Associate Professor of the Department of Multimedia Engineering, Osaka University. He has published more than 300 international Journal and conference papers in the areas of databases, mobile computing, peer-to-peer systems, WWW, and wireless networking. He served and is serving as a Program Chair of IEEE International Conferences on Mobile Data Management (MDM’06 and 10) and Advanced Information Networking and Applications (AINA’09 and 14), and IEEE International Symposium on Reliable Distributed Systems (SRDS’12). He guest edited IEEE Journal on Selected Areas in Communications, Sp. Issues on Peer-to-Peer Communications and Applications. His research interests include distributed databases, peer-to-peer systems, mobile networks, and mobile computing systems. He is a senior member of IEEE and ACM and a member of three other learned societies.

Shojiro Nishio received his B.E., M.E., and Ph.D. degrees from Kyoto University in Japan, in 1975, 1977, and 1980, respectively. He has been a full professor at Osaka University since August 1992, and is currently a distinguished professor of Osaka University. He served as a Vice President and Trustee of Osaka University from August 2007 to August 2011. He also acted as the Program Director in the Area of Information and Networking, Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan from April 2001 to March 2008. His research interests include database systems and multimedia systems for advanced networks such as broadband networks and mobile computing environment. Dr. Nishio has co-authored or co-edited more than 55 books, and authored or co-authored more than 600 refereed journal or conference papers. He served as the Program Committee Co-Chairs for several international conferences including DOOD 1989, VLDB 1995, and IEEE ICDE

2005. He has served as an editor of international journals including IEEE Trans. on Knowledge and Data Engineering, VLDB Journal, ACM Trans. on Internet Technology, and Data & Knowledge Engineering. Dr. Nishio has received numerous awards during his research career, including the Medal with Purple Ribbon from the Government of Japan in 2011, which is awarded to people who have made outstanding and important contributions in academic fields, arts and sports. He is also a fellow of IEEE, IEICE and IPSJ, and is a member of four learned societies, including ACM.

