Two-Phase Top-k Query Processing in Mobile Ad Hoc Networks

Yuya SASAKI #1, Takahiro HARA #1 and Shojiro NISHIO #1

Department of Multimedia Engineering Graduate School of Information Science and Technology, Osaka University
Yamadaoka 1-5, Suita, Osaka, 565-0871, Japan
[1]{ sasaki.yuya, hara, nishio } @ist.osaka-u.ac.jp

***Abstract*—In mobile ad hoc networks (MANETs), to acquire only necessary data items, it is effective that each mobile node retrieves data items using a top-k query, in which data items are ordered by the score of a particular attribute and the query-issuing mobile node acquires data items with k highest scores. In our previous works, we proposed query processing methods for top-k query for reducing traffic and also keeping high accuracy of query result. However, those methods have the problem that reduction of candidates included in the top-k result is still not enough because nodes cannot know the accurate k-th highest score. In this paper, we propose a two-phase query processing method for top-k query processing. In this method, the query-issuing node collects the information of scores of data items held by each node in the first phase. Based on the received information, it determines the threshold of the score, i.e., the estimated k-th highest score. Then, in the second round, the query-issuing node transmits a query attached with the threshold, and each node that received the query sends back only its own data items whose scores are equal to or larger than the threshold. In this way, our proposed method can further reduce the traffic and also keep high accuracy of the query result.**

***Index Terms*—MANET; Top-k query; Two-phase;**

I. Introduction

Recently, there has been an increasing interest in *mobile ad hoc networks (MANETs)*[5], [7], which are constructed by only mobile nodes. In MANETs, all mobile nodes play the role of a router. Even if the source and the destination mobile nodes are not within the communication range of each other, data packets are forwarded to the destination mobile node by relaying the transmission through other mobile nodes that exist between the two mobile nodes.

Since the network bandwidth and batteries of mobile nodes are limited in MANETs, it is important for query processing to acquire only necessary data items to reduce data traffic. A possible and promising solution is that each mobile node acquires data items using a top-k query, in which data items are ordered by the score of a particular attribute and the query-issuing node acquires data items with the k highest scores (*top-k result*).

A naive manner to achieve a top-k query is as follows. A query-issuing node floods all mobile nodes in the entire network with a query message, and then, each mobile node that receives the query message transmits its own data items with the k highest score. By doing so, the query-issuing node can acquire all data items included in the top-k result if communication failures do not occur. However, many data items that are not included in the top-k result are sent back to the query-issuing node. This increases unnecessary traffic and causes communication failures, which results in the decrease of the accuracy of the query result.

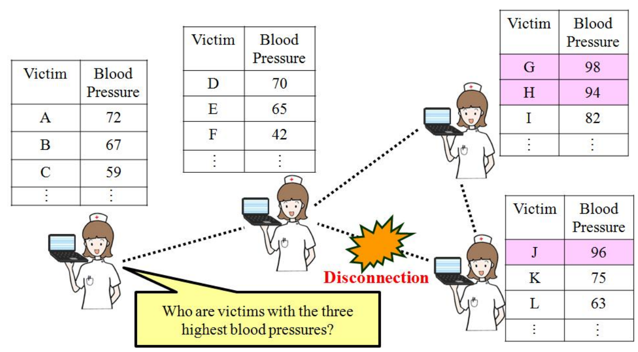


Fig. 1. Example of top-k query in a MANET

This problem can be improved by sending back a fixed number (less than k) of data items with high scores to the query-issuing node. However, setting an appropriate number is very difficult practically. For example, if the number of transmitted data items is large, the query-issuing node can acquire all data items included in the top-k result with high probability. However, the traffic problem remains. On the other hand, if the number of transmitted data items is small, the traffic problem can be solved, but the query-issuing node cannot acquire some data items that are included in the top-k result, i.e., the accuracy of the query result decreases.

In MANETs, there is another big problem. Since mobile nodes move freely, connections and disconnections of radio links frequently occur. If some radio links between mobile nodes are disconnected during the execution of a top-k query, the query-issuing node might not be able to acquire some data items that are included in the top-k result. This also causes the decrease of the accuracy of the query result.

Figure 1 shows an application example of a top-k query assuming a rescue effort at a disaster site where the communication infrastructure, e.g., Internet, has been broken down. Each nurse manages the information on victims and their medical records, and searches for victims with high blood pressure to provide them with medical treatment with a limited number of doctors. When the nurse on the left side wants to find victims with the three highest blood pressures in the entire area, she performs a top-k query where k is set to 3. Here, if each nurse transmits the information on victims with the three highest blood pressures in her responsible region, the query-issuing nurse acquires the information on more victims than necessary. On the other hand, if each nurse transmits the information on only one victim with the highest blood pressure, she cannot acquire the information on the victim with the third highest blood pressure, i.e., H, in the entire area. In addition, if the radio link between portable computers held by the two nurses on the bottom-right and top-left sides is disconnected during the query execution, the query-issuing nurse may not be able to acquire the information on the victim with the highest blood pressure, J.

In our previous works [4], [8], we proposed query processing methods for top-k query for reducing traffic and also keeping high accuracy of query result. Those methods estimate the k-th score based on the scores of data items held by each node on the query path, and reduce the candidates of data items included in the top-k result being sent back to the query-issuing node. In those methods, we adopted the design policy that a top-k query is processed in one round of massage transmissions (query flooding and replies) to reduce the query execution time to avoid the network topology change during the query execution. However, since the estimation is not based on the scores of all data items in the entire network, the accuracy of the estimation is not very high, thus, the reduction of candidates included in the top-k result is still not enough.

In this paper, to solve this problem, we propose a two-phase query processing method for top-k query processing in MANETs. In this method, the query-issuing node collects the information on scores of data items held by each node in the first phase (round). Based on the received information, it determines the threshold of the score, i.e., the estimated k-th highest score. Then, in the second phase (round), the query-issuing node transmits a query attached with the threshold, and each node that received the query sends back only its own data items whose scores are equal to or larger than the threshold. In this way, our proposed method can further reduce the traffic and also keep high accuracy of the query result. Moreover, when a mobile node detects the disconnection of a radio link during the transmission of the reply, it sends the reply to another neighboring node to preserve high accuracy of the query result. We present simulation results to evaluate the performance of our proposed method.

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 data access to acquire only necessary data items to reduce data traffic. In this paper, we proposed an effective top-k query processing method for reducing traffic and also preserving high accuracy of the query result.
- In our proposed method, the query-issuing node can estimate the k-th highest score in the entire network more accurately than the previous methods through the first phase. Therefore, our proposed method can further reduce the traffic compared with the previous methods as well as keeping high accuracy of the query result. To the best of our knowledge, this method is the first approach that assumes two rounds of message transmissions for processing top-k queries in MANETs.
- Through extensive simulations assuming various situations, we show that our method works well in terms of traffic reduction and high accuracy.

The remainder of this paper is organized as follows: In section II, we introduce related works. In section III, we present our proposed top-k query processing method in MANETs. In section IV, we show the results of the simulation experiments. Finally, in section V, we summarize this paper.

II. Related work

In the research field of unstructured P2P networks, many strategies for processing a top-k query have been proposed. In [6], the authors proposed a method in which the query-issuing node floods all nodes with a query message that includes the number of requested data items (k). Then, each node that receives the query message orders its own data items by score, and transmits data items with k highest scores. This method transmits many unnecessary data items that are not included in the top-k result. In [2], the authors proposed a method in which the query-issuing node acquires the data item with the highest score by transmitting a query message and continues the same procedure k times to acquire the top-k result. This method minimizes the number of transmitted data items but takes very long time to acquire the top-k result. Since the network topology dynamically changes in MANETs, it is not effective to apply this method. In [1], the authors proposed a method in which each mobile node selectively transmits query messages and replies to neighboring nodes in order to reduce the network traffic. This method assumes an environment where any pair of two nodes can communicate with each other by (one-hop) unicast, which is impossible in MANETs.

In [4], [8], we proposed message processing methods for top-k query in MANETs for reducing traffic and keeping accuracy of the top-k result. To the best of our knowledge, there is no work addressing top-k query processing in MANETs before us. In the method proposed in [4], each mobile node attaches a query message with scores of some data items held by itself and its ancestor nodes on the query path, which we call $standard\ scores$. Then, when each mobile node sends back a reply, it reduces the number of candidates of data items that are included in the top-k result by using the standard scores. These standard scores are updated (improved) as the query message is forwarded to neighboring nodes. This method can prevent data items that are certainly not included in the top-k result from being transmitted to the query-issuing node. In [8], we proposed a top-k query processing method to further reduce the traffic. In this method, each mobile node constructs a histogram from the scores of its holding data items and estimates the distribution of scores of all data items and the k-th highest score in the entire network. This histogram is attached with query messages and replies so that other mobile nodes can estimate the score distribution and k-th score more accurately. In addition, taking into account that there is a difference between the estimated k-th score and the real one, each mobile node updates the estimated k-th score using a safety margin and the k-th score in the result of the previous top-k query. When transmitting a reply, each mobile node sends back only data items with larger scores than the estimated k-th score to reduce traffic as much as possible. However, since these methods cannot estimate the k-th highest score accurately, reduction of the candidates included in the top-k result is not enough. Moreover, in the method proposed in [8], some data items included in the top-k result might not be sent back to the query-issuing node due to the estimation error. On the other hand, in our proposed method, since the query-issuing node can estimate the k-th highest score more accurately through the first phase, it can further reduce the number of candidates of data items included in the top-k result.

III. Top-k Query Processing method

In this section, first, we describe the design policy of our proposed method and the assumed environment. Then, we describe the overview of our proposed method and how to process a top-k query in detail.

A. Design Policy

In MANETs, since mobile nodes consume the limited communication bandwidth for data transmission, a packet loss and packet retransmission may occur when the network is congested, i.e., some data items cannot be transmitted. Thus, the number of data items transmitted by each mobile node should be reduced as much as possible. For this aim, it is effective for the query-issuing node to estimate the k-th highest scores in the entire network as accurate as possible. Thus, in this paper, we put the first priority to accurate estimation of the k-th highest score, which results in the reduction of traffic. This design policy is different with that in [4] and [8], which put high priority to short query execution time to cope with the network topology change. Specifically, we adopt two-phase top-k query processing in this paper, where information on scores of data items held by all nodes in the first phase and estimate the k-th highest score accurately.

B. Assumptions

The system environment is assumed to be a MANET in which mobile nodes retrieve data items held by itself and other mobile nodes using a top-k query. The query-issuing node transmits a query message with the query condition and acquires data items with the k highest scores among all data items held by mobile nodes in the entire network.

We assign a unique $data\ identifier$ to each data item in the system. The set of all data items in the system is denoted by $D = \{D_1, D_2, \cdots, D_n\}$, where n is the total number of data items and $D_i (1 \leq i \leq n)$ is a data identifier. Each data item is held by a specific node. Data items are not updated, but are newly generated and deleted as time passes, i.e., n dynamically changes. For simplicity, all data items are of the same size and not replicated. The scores of data items can be calculated from the query condition and some scoring functions.

TABLE I
Scores of data items held by each mobile node

rank	M_1	M_2	M_3	M_4	M_5
1	79	96	81	83	74
2	60	87	66	76	61
3	58	77	65	72	52
4	53	64	50	69	47
⋮	⋮	⋮	⋮	⋮	⋮

We assign a unique $node\ identifier$ to each mobile node in the system. The set of all mobile nodes in the system is denoted by $M = \{M_1, M_2, \cdots, M_m\}$, where m is the total number of mobile nodes and $M_i (1 \leq i \leq m)$ is a node identifier. Each mobile node moves freely.

C. Overview of our method

Our proposed method process a top-k query in two phases: $search\ phase$ and $data\ collection\ phase$.

In the search phase, the query-issuing node collects the information on scores of data items held by all nodes in the entire MANET. Then, it checks the k-th highest score among those scores and determines the score as the threshold. In the data collection phase, the query-issuing node collects data items with scores equal to or larger than the threshold. In this way, our proposed method can decrease the number of transmitted data items as much as possible.

Figure 2 shows an example of the search phase when M_1 issues a top-k query for acquiring data items with the three highest scores ($k = 3$). Table I shows the scores (descending order) which each mobile node calculates for its own data items from the query condition using the scoring function. In this figure, the list in each balloon denotes the three highest scores among data items held by the corresponding node and its children nodes. For example, M_3 that receives a query message from M_1 replies a message attached with the three highest scores (81, 66 and 65) since it does not have any child in the query propagation tree. Finally, M_1 acquires the three highest scores in the entire network and determines the threshold.

Figure 3 shows an example of the data collection phase. In this figure, the list in each balloon denotes the scores of the data items which are held by the corresponding node and transmitted to the query-issuing node. Since each node that received the query message sends back its own data items whose scores are equal to or larger than the threshold (83), the number of transmitted data items can be reduced as much as possible.

D. Top-k Query Processing

Message processing in both the search phase and the data collection phase is similar to each other, i.e., flooding a query message and collecting replies. In the following, we explain the detail of the processes.

1) Transmissions of Query Messages: A query message in the search and data collection phases consists of the following elements.

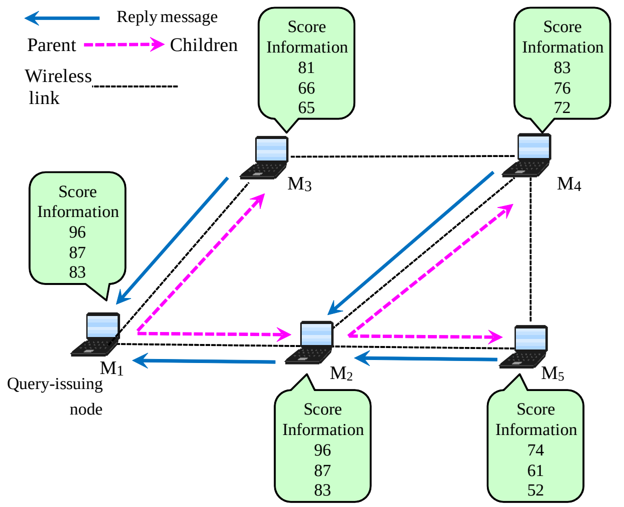


Fig. 2. Example of the search phase

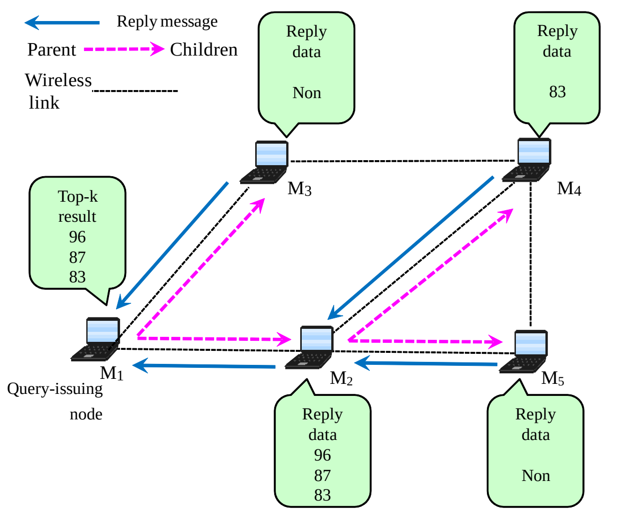


Fig. 3. Example of the data collection phase

- *Query-issuing node's ID*: The node identifier of the query-issuing node.
- *Query ID*: The query identifier of the query (e.g., sequence number).
- *Number of requested data items*: The number of data items, k, that the query-issuing node requests.
- *Query condition*: The query condition specified by the query-issuing node.
- *Query path*: The list of node identifiers of mobile nodes on the path through which the query message is transmitted.
- *Threshold (only for data collection phase)*: The threshold set by the query-issuing node, i.e., the estimated k-th highest score. This is an empty for a query message in the search phase.

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

1) M_p specifies the number of requested data items, k, and the query condition. Then, M_p calculates scores of its holding data items from the query condition using the scoring function.
2) M_p transmits a query message to its neighboring mobile nodes. In the query message, the query-issuing node's ID is set as M_p, the query ID is set as the value determined based on some numbering method (we do not restrict the manner), the number of requested data items is set as k, the query condition is set as the condition specified by the user and the query path is set as $\{M_p\}$. In the case of data collection phase, the threshold determined by the query-issuing node is also attached with the query message.
3) Each mobile node, M_q, that received the query message stores the identifier of the last mobile node in the query path and its hopcount. Here, the hopcount is the distance from the query-issuing node to the last node in the query path. If M_q has already received the same query message in the past in the same phase, the procedure goes to step 5. Otherwise if M_q firstly receives the query message, it sets the mobile node that sent this message as its parent. Then, the procedure goes to step 4.
4) M_q appends its node ID to the query path in the query message. Then, M_q forwards the query message to its neighboring nodes. The procedure goes back to step 3.
5) M_q sets the last mobile node in the query path as its neighboring node. Moreover, if the second last mobile node in the query path is M_q, M_q sets the last mobile node in the query path as its child.

After the above process, each mobile node can recognize its parent and child in the tree whose root is the query-issuing node (query propagation tree), the path from query-issuing node to itself and the hopcounts to its neighboring nodes. In the case of data collection phase, each node can know the threshold for the query. Here, the network topology may change between search phase and data collection phase so the query propagation tree constructs in data collection phase again.

2) Transmissions of Replies: In our proposed method, each mobile node sends back a reply message attached with either the score information (in the search phase) or the data items with scores equal to or larger than the threshold (in the data collection phase). By doing so, in the search phase, the query-issuing node can acquire the score information in the entire network. In the data collection phase, the query-issuing node can acquire data items included in the top-k query result.

A reply consists of the following elements.

- *Query-issuing node's ID*: The node identifier of the query-issuing node.
- *Query ID*: The query identifier of the query.
- *Score list (only for search phase)*: The list of scores of data items.
- *Data list (only for data collection phase)*: The list of pairs of data items whose score are larger than the threshold and their scores.

The behaviors of each mobile node that has completed the transmission of a query message is as follows.

1) Each mobile node, M_r, that has no children transmits a reply to its parent. In the reply, the query-issuing node's ID is set as M_p, the query ID is set as the query ID in the received query message. In the case of search phase, the score list consists of the k highest scores among its own data items. In the case of data collection phase, the data list consists of pairs of data items whose scores are equal to or larger than the threshold among its holding data items and their scores.
2) When a mobile node, M_s, receives replies from all its children or the predetermined time limit has passed, M_s transmits a reply to its parent. In the case of search phase, the score list consists of the k highest scores among scores of its own data items and those in the score lists in the replies received from its children. In the case of data collection phase, the data list consists of pairs of data items with scores equal to or larger than the threshold among its own data items and all data items in the data lists in the replies received from its children and their scores. Here, if the number of data items contained in the data list becomes larger than k, the reply list is updated to contain only data items with the k highest scores.
3) If M_s that has already transmitted the reply to its parent receives a reply from a mobile node which is not its child, M_s behaves differently according to the phase. In the case of data collection phase, M_s re-transmits a reply to its parent in the same way as in step 2. However, data items that have been already transmitted in the previous reply and their scores are deleted from the reply list to prevent the duplication. In the case of search phase, M_s simply relays the received reply to its parent.

In the procedure of transmitting replies, each mobile node sends either the score information or only data items whose scores are equal to or larger than the estimated k-th highest score, and does not send data items that are likely not included in the top-k result. Since the data volume of the information on scores is not very large, our proposed method can reduce traffic and keep high accuracy of the query result, while it may make the query processing time longer due to two rounds of message transmissions.

3) Detection of Radio Link Disconnection: In MANETs, the network topology dynamically changes due to the migration of mobile nodes. If a mobile node cannot transmit a reply to its parent because of the disconnection of the radio link between these nodes, the accuracy of the query result decreases. Therefore, when a mobile node detects the disconnection of the radio link to its parent, it transmits the reply to another neighboring node. The procedure when a mobile node, M_t, detects a radio link disconnection is as follows.

1) When M_t detects a disconnection to its parent, it transmits the reply to a neighboring mobile node. If there are multiple candidates, it transmits the reply to a node with the minimum hopcount to the query-issuing node to reduce traffic and delay as much as possible. If M_t has no neighboring nodes, M_t discards the reply.
2) When a mobile node, M_u, receives a reply from a mobile node which is not M_u's child, it transmits the reply to another neighboring node as follows. If M_t is not included in the path from the query-issuing node to M_u, M_u transmits the reply to its parent as described in clause III-D2. Otherwise, if M_t is included in the path from the query-issuing node to M_u, such as the case where M_t is M_u's parent, transmitting the reply to M_u's parent might cause that the reply will be sent back to M_t. To avoid this, M_u transmits the reply to a neighboring node except for M_u's parent. Here, if M_u's parent is only the neighboring node of M_u, it discards the reply.

By this procedure, the query-issuing node can receive replies from mobile nodes that disconnect from their parents and the accuracy of the query result can be preserved.

We show an example of the procedure in which a mobile node detects the disconnection and find an alternative path. If the radio link between M_2 and M_4 is disconnected in Figure 3, M_4 transmits the reply to M_3 with the minimum hopcount to the query-issuing node.

IV. Simulation Experiments

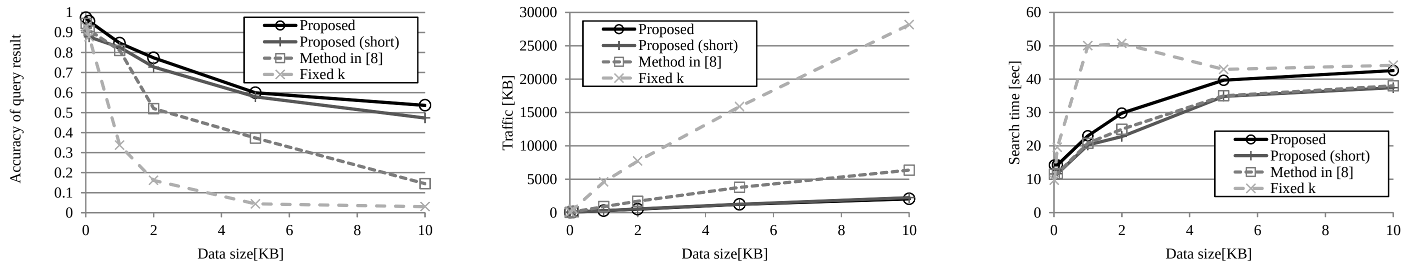
In this section, we show the results of simulation experiments regarding the performance evaluation of our proposed method. For the simulation experiments, we used a network simulator, QualNet4.0 [9].

A. Simulation Model

The number of mobile nodes in the entire system is 50 ($M_1, M_2, \cdots, M_{50}$). These mobile nodes exist in an area of R [m] $\times$ R [m] and move according to the random waypoint model [3] where the movement speed and the pause time are fixed to 1.0 [m/sec] and 60 [sec]. Each mobile node transmits messages and data items 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]. We assume that packet losses and delays occur due to radio interference.

The number of data items that each mobile node initially has is randomly determined from the range between 80 to 120. It randomly increases by 1 to 5 at every 300 [sec] from the start time (0[sec]) to 2,400 [sec], then, randomly decreases by 1 to 5 at every 300 [sec] from 2,400 [sec] to 4,800 [sec]. It then randomly increases again by 1 to 5 at every 300[sec] until the simulation end time. The size of data items is d [KB]. The score distribution of data items held by each mobile node is skewed and different with other nodes. The score distribution in the entire network follows the normal distribution and the total range of scores are set as [80, 320). Each mobile node repeatedly issues a top-k query at time intervals of 3,000 [sec].

We compare the performance of our proposed method with that of the method proposed in [8] and a naive method. In the method proposed in [8], each mobile node constructs a histogram from the scores of its own data items and estimates the distribution of scores of all data items and the k-th highest score in the entire network. In addition, taking into account



(a) Accuracy of query result (b) Traffic (c) Search time

Fig. 4. Effects of d

TABLE II
PARAMETER CONFIGURATION

Symbol	Meaning	Values (Range)
k	Number of requested data items	100 or 10 (1∼100)
d	Size of data items	1 [KB] (0.01∼10)
R	Area size	600 [m] (600∼2000)

that there is a difference between the estimated k-th score and the real one, each mobile node updates the estimated k-th score using a safety margin and the k-th score in the result of the previous top-k query. When transmitting the reply, each mobile node sends back only data items with scores equal to or larger than the estimated k-th score. In [8], two different approaches for estimating the k-th score were proposed. In our simulations, we adopt one of the approaches, which uses the probability density function. In the naive method, the query-issuing node floods the entire network with a query message and each node replies k data items with the k highest scores among its own data items and the data items contained in the replies received from its children.

The query-issuing node waits for α [sec] to acquire replies, i.e., the query deadline is α [sec]. We determine different query deadlines for different methods through the preliminary experiments, so that these methods show the highest accuracy of query result under the deadline. Only for our method, we also adopt another shorter query deadline (denoted as "short"), which is same as that for the method in [8]. Our proposed method with the short query deadline sometimes cannot completely collect the score information and replies.

Table II shows the parameters and their values used in the simulation experiments. The parameters are basically fixed to constant values specified by numbers to the left of the parenthetic values. Each of them is varied in the range specified by the parenthetic element in a simulation experiment.

In the above simulation model, we randomly determine the initial position of each mobile node and evaluate the following criteria during 7,200 [sec].

- *Accuracy of query result:* is defined as the ratio of the numbers of data items acquired by the query-issuing node that are included in the top-k result to the number of requested data items, k.
- *Traffic:* is defined as the average of the total volume of query messages and replies exchanged for processing a query.
- *Search time:* is defined as the average of the elapsed time after the query-issuing node issued a top-k query until it acquired the result.

B. Impact of Data Size

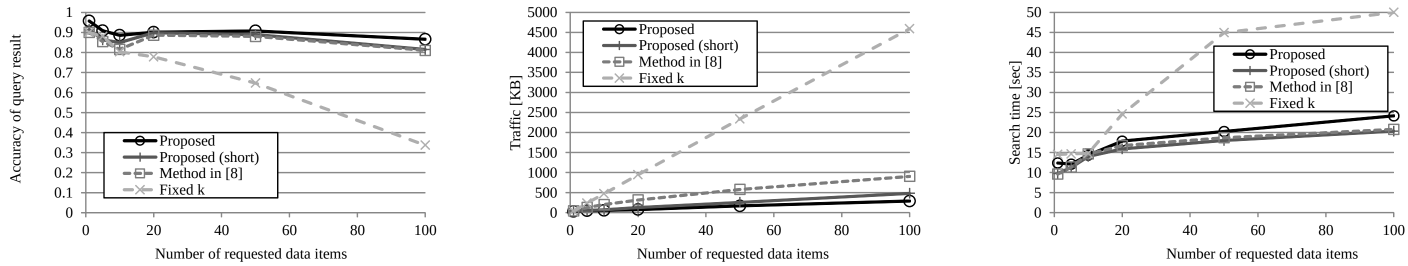
We examine the effects of size of a data item, d, where the number of nodes is 50 and k is fixed to 100. Figure 4 shows the simulation result. In this graph, the horizontal axis indicates d. The vertical axis indicates the accuracy of query result in Figure 4 (a), the traffic in Figure 4 (b) and the search time in Figure 4 (c). In these graphs, "Proposed" denotes proposed method with the proper query deadline, "Proposed (short)" denotes our proposed method with the shorter query deadline, "Method in [8]" denotes the method proposed in [8] and "Fixed k" denotes the naive method.

From Figure 4 (a), as d increases, the accuracy of query result decreases in all methods. This is due to packet losses caused by the increase of traffic. When d is small, the accuracy of query result in all methods is high. However, no methods achieve 100 % of the accuracy because network partitioning occurs. When d is large, our proposed method gives higher accuracy than the method in [8] and the naive method. This is because when the data size is large, reducing the number of transmitted data items is effective to avoid packet losses. Since the query-issuing node can estimate the k-th highest score more accurately by collecting the score information in the first phase, our method can further reduce the number of transmitted data items compared with other methods as shown in Figure 4 (b).

From Figure 4 (c), the search time in our method ("Proposed") is longer than that in the method proposed in [8]. This is because our method requires two rounds of message transmissions. In the naive method, the search time firstly increases, and then decreases. This is because when d is large, packet losses too much occur in the naive method, and thus, the optimal query deadline becomes slightly shorter to avoid too much (useless) retransmissions of queries and replies. Compared between the two cases in our method, the search time in case of the optimal query deadline (in terms of the accuracy of query result) is longer than that in with the short deadline, while the accuracy of query result is slightly higher.

C. Impact of Number of Requested Data Items

Next, we examine the effects of the number of requested data items, k. Figure 5 shows the simulation result. In this graph, the horizontal axis indicates k. The vertical axis indicates the accuracy of query result in Figure 5 (a), the traffic in Figure 5 (b) and the search time in Figure 5 (c).



(a) Accuracy of query result (b) Traffic (c) Search time

Fig. 5. Effects of k

From this result, as k increases, the accuracy of query result increases at first, then it decreases in our proposed method with the short query deadline and the method proposed in [8] while it keeps decreasing in other two cases. When k is very small, our proposed method with the optimal query deadline and the naive method show higher accuracy than other cases. In our proposed method with the short query deadline, the query-issuing node sometimes cannot receive some data items included in the top-k result due to the short query deadline, and this impact is big when k is very small. In the method proposed in [8], the impact of errors in the k-th score estimation is big when k is very small. (Intuitively, when k is very small, e.g., $k = 1$, one missing data items much degrades the accuracy of query result.) As k increases, packet losses occur more frequently, and thus, the accuracy of query result decreases in all methods, which can be confirmed by the increase in traffic as shown in Figure 5 (b). Since the proposed method with the optimal query deadline can reduce the traffic as much as possible, the accuracy of query result is higher than the other methods. From Figures 5 (b) and (c), as k increases, the traffic and the search time increase in all methods since the size of a reply message increases. Since the proposed method collects the score information in the first phase, the number of transmitted data items is reduced, which is lower than that in the method proposed in [8]. However, since query processing requires two rounds of message transmissions, the search time becomes slightly longer. Our method with the short deadline gives almost same search time as the method in [8], while it still gives slightly lower traffic. The naive method gives much higher traffic and search time than other methods because it requires often retransmissions of messages due to packet losses.

D. Impact of Area Size

Finally, we examine the effects of the area size and the data size. Figures 6, 7 and 8 show the simulation results. The area size is 1000 [m] in Figure 6, 1500 [m] in Figure 7, and 2000[m] in Figure 8, and the number of requested data items, k, is 10. Intuitively, the area size affects the number of data items collected by the query-issuing node. Here, as mentioned subsection IV-C, when k is small, the impact of missing data items that should be included in the top-k result is big. Therefore, to make clear the impact of the area size on the system performance, we set k as 10 (i.e., k is smaller than that in above experiments).

In those graph, the horizontal axis indicates d. The vertical axis indicates the accuracy of query result in Figures 6, 7 and 8 (a), the traffic in Figures 6, 7 and 8 (b), and the search time in Figures 6, 7 and 8 (c).

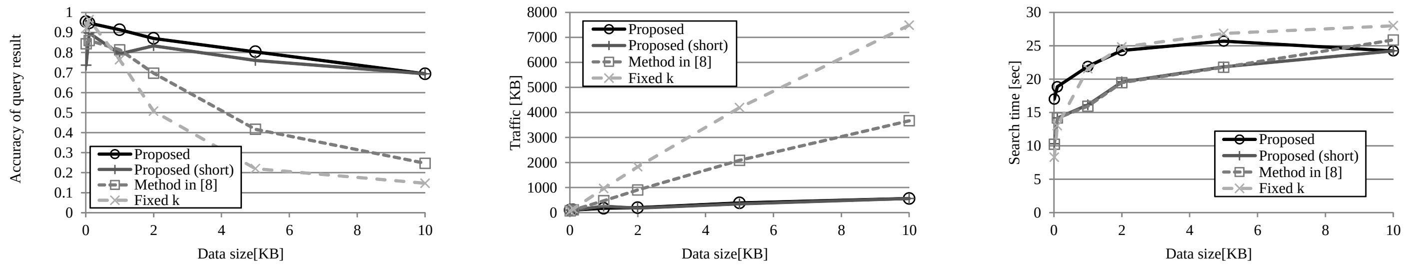
From Figures 6, 7 and 8 (a), as the area size increases, the accuracy of query result decreases in all methods. This is because when the area size is large, network partitioning frequently occurs, thus, the query-issuing node cannot acquire some data items included in the top-k result. When the data size is small, the accuracy of query result in our method with the short query deadline is smaller than that in other cases. This is because our method with the short query deadline cannot get some replies due to the short query deadline. On the other hand, the accuracy of query result in our method with the optimal query deadline is higher than that in other methods. This is because our proposed method with the optimal query deadline waits for replies so that the accuracy of query result becomes high.

From Figures 6, 7 and 8 (b), the traffic is the largest in the case that the area size is 1500 [m] ($R = 1500$ [m]) and the smallest in the case that the area size is 2000 [m] ($R = 2000$ [m]). The traffic in the case of $R = 1500$ [m] increases (compared with the case of $R = 1000$ [m]) because the hopcount increases as the network becomes sparse. The traffic in the case of $R = 2000$ [m] is small because network partitioning often occurs, thus, some data items cannot be sent to the query-issuing node. The search time in the case of $R = 1500$ [m] is the longest because the hopcount is large.

V. Conclusions

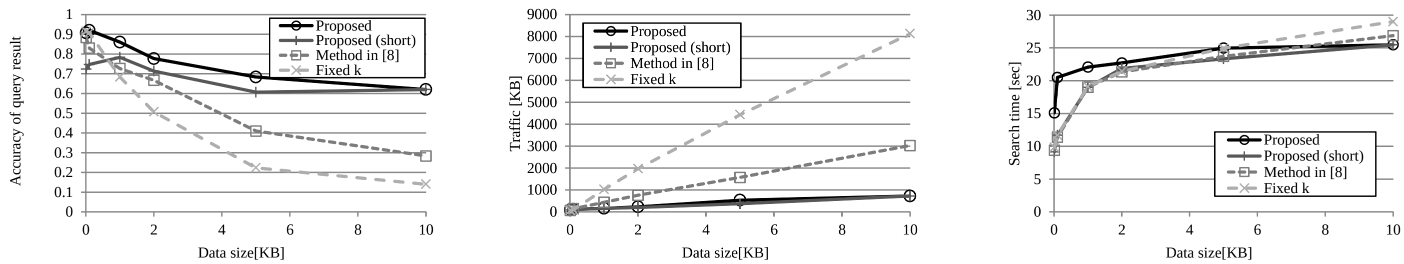
In this paper, we propose a two-phase top-k query processing method in MANETs. In this method, the query-issuing node collects the information of scores of data items held by each node in the first phase. Based on the received information, it determines the threshold of the score, i.e., the estimated k-th highest score. Then, in the second phase, the query-issuing node transmits a query attached with the threshold, and each node that received the query sends back only its own data items whose scores are equal to or larger than the threshold. In this way, our proposed method can further reduce the traffic and also keep high accuracy of the query result. Moreover, when a mobile node detects the disconnection of a radio link during the transmission of a reply, it sends the reply to another neighboring node to preserve high accuracy of the query result.

The simulation results showed that our proposed method reduces traffic and achieves high accuracy of the query result



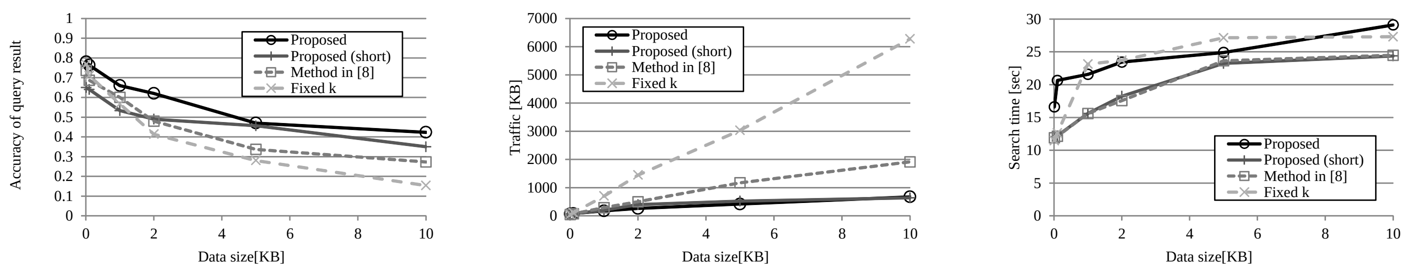
(a) Accuracy of query result (b) Traffic (c) Search time

Fig. 6. Effects of d where the area size is 1000 [m]



(a) Accuracy of query result (b) Traffic (c) Search time

Fig. 7. Effects of d where the area size is 1500 [m]



(a) Accuracy of query result (b) Traffic (c) Search time

Fig. 8. Effects of d where the area size is 2000 [m]

compared with the method in [8] and the naive method. We also compared the performance between the cases of two query deadlines in our method. With the optimal query deadline in terms of the accuracy of the query result, the traffic decreases and the accuracy of the query result increases while the search time increases. On the other hand, with the short query deadline, the search time can be shortened while the accuracy of the query result decreases.

In our proposed method, the query-issuing node collects the information of scores of data items held by all nodes in the entire network in the first phase. This process takes long time when the network size (i.e., the number of mobile nodes and the hopcount) is very large. Therefore, we plan to extend our method so that the query-issuing node estimates the threshold from a part of scores.

REFERENCES

[1] R. Akbarinia, E. Pacitti, and P. Valduriez, "Reducing network traffic in unstructured P2P systems using top-k queries," *Distributed and Parallel Databases*, vol.19, no. 2–3, pp. 67–86, 2006.

[2] W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden, "Progressive distributed top-k retrieval in peer-to-peer networks," in *Proc. Int. Conf. on Data Engineering*, pp. 174–185, 2005.

[3] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing (WCMC)*, vol. 2, no. 5, pp. 483–502, 2002.

[4] R. Hagihara, M. Shinohara, T. Hara, and S. Nishio, "A message processing method for top-k query for traffic reduction in ad hoc networks," *Proc. Int. Conf. on Mobile Data Management*, pp. 11–20, 2009.

[5] D. B. Johnson, "Routing in ad hoc networks of Mobile Hosts,"Proc. IEEE WMCSA'94, pp. 158–163, 1994.

[6] P. Kalnis, W.S. Ng, B.C. Ooi, and K.-L. Tan, "Answering similarity queries in peer-to-peer networks," *Information Systems*, vol. 31, no. 1, pp. 57–72, 2006.

[7] C. E. Perkins, and E. M. Ooyer, "Ad-hoc on-demand distance vector routing," Proc. iEEE WMCSA'99, pp. 90–100, 1999.

[8] Y. Sasaki, R. Hagihara, T. Hara, M. Shinohara, and S. Nishio, "A top-k query method by estimating score distribution in mobile ad hoc networks," Int. Workshop on Data Management for Wireless and Pervasive Communications (DMWPC 2010), pp. 944–949, 2010.

[9] Scalable Networks: makers of QualNet and EXata, the only multi-core enabled network simulation and emulation software.[Online].Available: http://www.scalable-networks.com/.