A Top-k Query Method by Estimating Score Distribution in Mobile Ad Hoc Networks

Yuya Sasaki, Ryo Hagihara, Takahiro Hara, Masako Shinohara and Shojiro Nishio
Dept of Multimedia Engineering Graduate School of Information Science and Technology, Osaka University
{sasaki.yuya,hagihara.ryou,hara,sinohara.masako,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 this paper, we propose a query processing method for top-k query for reducing traffic and also keeping high accuracy of the query result. 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 k-th 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. When transmitting a reply, each mobile node sends back only data items whose scores that are larger than the estimated k-th score to reduce traffic as much as possible. We also present simulation results to evaluate the performance of our proposed method.

1 Introduction

Recently, there has been an increasing interest in $mobile$ $ad\ hoc\ networks$ $(MANETs)$, which are constructed by only 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 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 node acquires data items with the k highest scores $(top\text{-}k\ result)$.

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 high score. 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

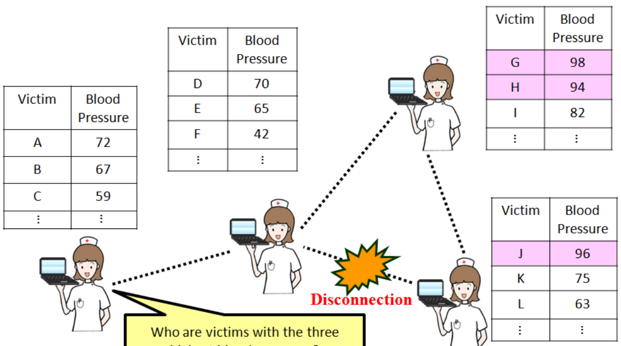


Figure 1: Example of top-k query in a MANET

move freely in MANETs, connections and disconnections of radio links between mobile nodes frequently occur.

In Figure 1, 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 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, the query-issuing nurse cannot acquire the information on the victim with the highest blood pressure, J.

In this paper, we propose a query processing method for top-k query for reducing traffic and also preserving high accuracy of the query result. 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 k-th 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. When transmitting the 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.

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 remainder of this paper is organized as follows: In section 2, we introduce related works. In section 3, we present our proposed query processing method for top-k query. In section 4, we show the results of the simulation experiments. Finally, in section 5, we summarize this paper.

2 Related work

In the research field of unstructured P2P networks, many strategies for processing a top-k query have been proposed. In [4], 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, leaving the problem of the increase of traffic described in section 1 unsolved. In [1], 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 a very long time to acquire the top-k result. Since the network topology dynamical changes in MANETs, it is not effective to apply this method.

In [3], we proposed a message processing method for top-k query for reducing traffic and keeping accuracy of the top-k query. In this method, each mobile node estimates data items with the k highest scores and sets a part of those scores as $standard\ scores$. When a mobile node transmits query messages and replies, 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. However, since this method does not consider the score distribution,reduction of the candidates included in the top-k result is not much. On the other hand, since each mobile node estimates the score distribution from scores of data items held by other nodes on the query route, the method proposed in this paper can further reduce the number of the candidates included in the top-k result. However, if the score distribution is not estimated accurately, the query-issuing node might not be able to acquire some data items included in the top-k result.

3 Top-k Query method

3.1 Assumptions

The system environment is assumed to be a MANETs 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 $D_i (1 \le i \le n)$ is a data identifier. Each data item is held by a specific node. For simplicity, all data items are the same size and not replicated. The scores of data items can be calculated from the query condition and some scoring functions.

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 $M_i (1 \le i \le m)$ is a node identifier. Each mobile node moves freely.

Every mobile node knows the total number of mobile nodes (m), the maximum score (upper bound), the minimum score (lower bound).

3.2 Histogram Construction

Each mobile node calculates scores of its holding data items from the query condition and constructs a histogram from those scores. A histogram represents the score distribution, in which the total score range is divided uniformly into multiple sub-ranges and each sub-range is called a $class$. A histogram is constructed by counting the number of scores classified into each class.

In this paper, we denote a histogram constructed from scores of data items held by M_i as H_i where the total score range, $[MIN, MAX)$, is uniformly divided into C sub-ranges. We also denote each class as $c_j (1 \le j \le C)$, where, c_j denotes the j-th class.

3.3 Top-k Query Processing

3.3.1 Transmissions of Query Messages

When each mobile node receives a query message from another mobile node, it integrates its own histogram and the histogram in the received query message. Then, it attaches the integrated histogram with the received query message and forwards the message to its neighboring nodes.

A query message consists of the following elements.

- $Query\text{-}issuing\ node's\ ID$: The node identifier of the query-issuing node.
- $Query\ ID$: The query identifier of the query.
- $Number\ of\ requested\ data\ items$: 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 through the path which the query message is transmitted.
- $Query\ histogram$: The histogram attached with the query message.

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 some scoring functions and constructs its own histogram, H_p, as described in subsection 3.2.

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, the query path is set as $\{M_p\}$ and the query histogram is set as H_p.
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, the procedure goes to step 5. Otherwise if M_q first 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 constructs its own histogram, H_q, in the same way as step 1 and integrates it and the histogram in the received query message. Moreover, M_q appends its node ID to the query path in the query message and the query histogram is set as the integrated histogram. 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.

During the transmission of the query message, each mobile node integrates histograms of mobile nodes on the path from the query-issuing node to itself. Moreover, according to the query path, each mobile node can recognize its parent and child in the tree whose root is the query-issuing node, the path from query-issuing node to itself, and the hopcounts to its neighboring nodes.

Figure 2 shows an example of transmitting a query message when M_1 issues a top-k query. In this figure, the histogram in a balloon denotes the query histogram that the corresponding mobile node attaches with the query message. Histograms of mobile nodes, M_1, M_2, M_3, M_4 and M_5, are shown as yellow, green, pink, blue and purple bars respectively for easy understanding of the histogram integration process.

3.3.2 Transmissions of Replies

In our proposed method, each mobile node can estimate the score distribution and k-th score more accurately, as the number of scores included in the histogram gets higher. Therefore, each mobile node attaches the histogram with a reply. Moreover, a mobile node that sends a reply does not transmit data items that are not included in the top-k result to reduce data traffic. A reply contains the following elements.

$Query\text{-}issuing\ node's\ ID$: The node identifier of the query-issuing node.

$Query\ ID$: The query identifier of the query.

$Reply\ list$: The list of pairs of data items whose score are larger than the estimated k-th score and their scores.

$Reply\ histogram$: The histogram attached with the reply.

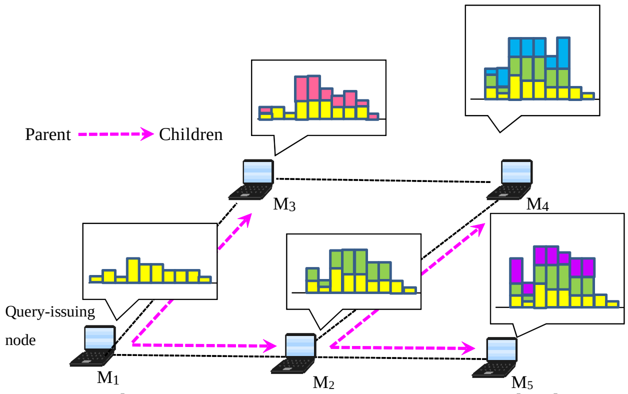


Figure 2. Query message transmission

The behavior 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 estimates the score distribution and k-th score in the entire network as described in subsection 3.4 . Then, M_r 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 query message, the reply histogram is set as H_r and the reply list contains data items whose scores are larger than the estimated k-th score 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, it integrates its query histogram and reply histograms attached with all received replies. Then, using the integrated histogram, M_s estimates the score distribution and k-th score in the same way as in step 1.
3. M_s transmits a reply to its parent. In the reply, the reply list contains data items whose scores are larger than the estimated k-th score among its holding data items and the data items contained in the replies received from its children and their scores. Then, if the number of data items contained in the reply list becomes larger than k, the reply list is updated to contain only data items with the k highest scores. Moreover, the reply histogram is set as the histogram integrated from H_s and all histograms received from its children.
4. If M_s that has already transmitted the reply receives a reply from a mobile node which is not its child, it integrates its reply histogram and the received histogram in the same way as step 2. It re-estimates the score distribution and k-th score. Then, M_s re-transmits a reply to its parent in the same way as step 3. 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 procedure of transmitting replies, each mobile node sends only data items whose scores are larger than the estimated k-th score and does not send data items that are likely not included in the top-k result. In addition, each

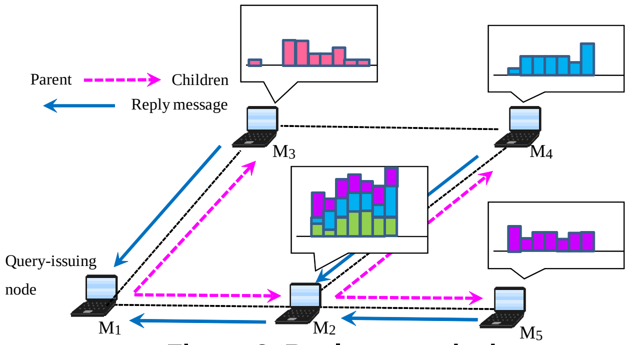


Figure 3. Reply transmission

mobile node transmits a reply with the reply histogram integrated from its own histogram and histograms of its descendants. As a result, each mobile node can estimate the score distribution more accurately because the number of scores included in the histogram becomes larger. Therefore, our proposed method can reduce traffic and preserve high accuracy of the query result.

Figure 3 shows an example of transmitting a reply. In this figure, the histogram in a balloon denotes the reply histogram that the corresponding mobile node attaches with the reply. In this example, M_4 that has no children estimates the k-th score from its own query histogram where it integrates yellow, green and blue histograms. Then, it transmits a reply to its parent, M_2, attached with the reply histogram and the list of pairs of data items whose scores are larger than the estimated k-th score and their scores.

3.3.3 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, an accuracy of the query result decreases. Therefore, when a mobile node detects the disconnection of a 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 3.3.2. 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.

In the procedure of detecting a radio link disconnection, if a mobile node detects that the radio link to its parent has been broken, it transmits the reply to another neighboring node. Thus, the query-issuing node can receive replies from the mobile nodes and the accuracy of the query result is prevented from decreasing as much as possible.

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 4, M_4 transmits the reply to M_3 with the minimum hopcount to the query-issuing node.

3.4 Estimation of score distribution and k-th score

In our proposed method, when each mobile node transmits a reply, it estimates the score distribution and k-th score in the entire network from the reply histogram that is constructed by integrating histograms of mobile nodes on the path from the query-issuing node to itself and its descendants.

Our approach assumes that the score distribution in the entire network is the same as that in the constructed histogram. First, the class (a-th class) that includes the k-th score in the entire network is estimated by using the following inequality and equation:

$$\sum_{l=a}^{C} \frac{n}{n_{EH}} \cdot e_l \geq k, \qquad (1)$$

$$n_{EH} = \sum_{l=1}^{C} e_l. \qquad (2)$$

Here, n denotes the number of data items in the entire network, k denotes the number of requested data items, C denotes the total number of classes in the histogram, $e_j(1 \leq j \leq C)$ denotes the number of data items whose scores are classified in j-th class, and n_{EH} denotes the total number of data items, i.e., scores, in the constructed histogram. Thus, in inequality (1), the number of data items classified in each class in the entire network is assumed to be proportional to the ratios of the numbers of data items in that class in the constructed histogram. Therefore, a denotes the class number where the summation of the number of data items classified in classes $c_l(a \leq l \leq C)$ firstly becomes equal to or larger than k.

Then, the mobile node estimates the k-th score, S_k, by using the following equation.

$$S_k = MIN + \frac{(a-1)(MAX - MIN)}{C}. \qquad (3)$$

Here, MIN and MAX respectively denote the minimum and maximum scores in the entire network, and S_k denotes the lower bound of c_a.

Figures 4 and 5 show an example of estimating the score distribution using the ratios of the numbers of data items in classes in the constructed histogram. Figure 4 shows the histogram constructed by M_3, where $C = 10$. n_{EH} becomes 40 according to equation (2). Here, if we assume

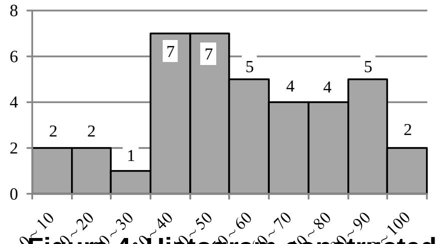


Figure 4. Histogram constructed by M_3

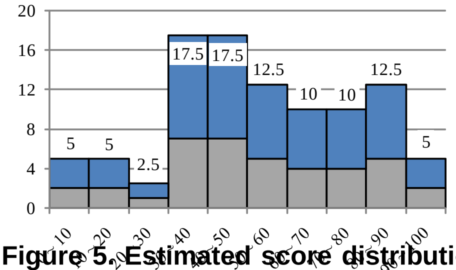


Figure 5. Estimated score distribution using the numbers of data items in constructed histogram

n is 100, M_3 estimates the score distribution in the entire network by increasing the number of scores in each class constructed histogram (Figure 4) by one and half. Figure 5 shows the estimated score distribution in the entire network. Here, let us assume $k = 10$. In this case, M_3 estimates the k-th score as the lower bound of c_9, i.e., 80, because the left part of inequality (1) first exceeds $k(= 10)$ when a is 9.

4 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 [5].

4.1 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 1,000 [m] $\times$ 1,000 [m] and move according to the random waypoint model [2] where the movement speed and the pause time are fixed to 0.5 [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.

Each mobile node holds 100 data items. The size of data items is d [KB]. The score distribution 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 which are randomly determined from 1,000 to 2,000 [sec]. Each mobile node estimates the score distribution and k-th score using the methods described in subsection 3.4. Each query-issuing node wait for 60 [sec] to collect replies, i.e., the query deadline is 60 [sec].

We compare the performance of our proposed method with that of a simple method where each mobile node that receives a query message sends back a fixed number, R, of its holding data items with R highest scores. We examine three different cases for R; $R = k$/50 and k.

In a simulation experiment, the number of requested data items, k, is basically 100 and is varied from 1 to 200 in subsection 4.2. The data size is basically 1[KB] and is varied from 0.01 to 5 in subsection 4.3. We determine the number of classes, C, as 20 based on our preliminary experiments, which showed this number is appropriate.

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: We examine 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: We examine the total volume of query messages and replies exchanged for processing a query.

4.2 Effects of k

We examine the effects of the number of requested data items, k, on the performance. Figure 6 shows the simulation result. In this graph, the horizontal axis indicates k. The vertical axis indicates the accuracy of query result in Figure 6 (a) and the traffic in Figure 6 (b).

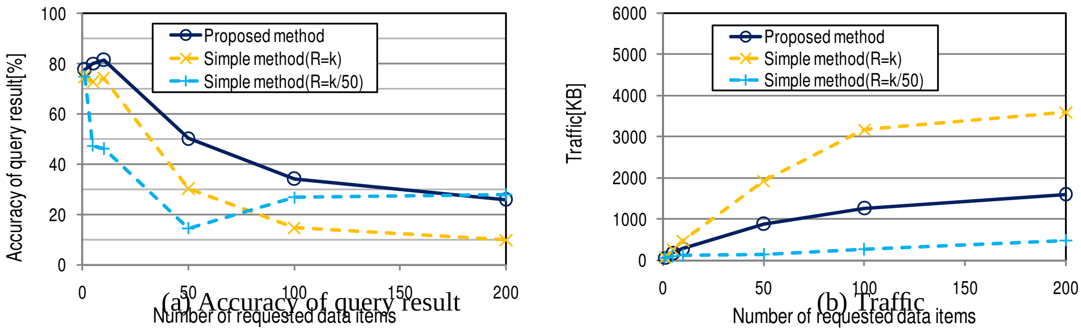
From this result, as k increases, the data items transmitted become larger and thus traffic becomes larger in both our method and the simple method. Therefore, the accuracy of query result decreases due to packet losses. In our method, the traffic is larger than simple method where R is 50/k. However, decreasing the accuracy of query result is slower than simple method.

As for the simple method, when k is less than 50 and R is set as k or k/2, the accuracy of query result is high. This is because the query-issuing node can acquire more data items than necessary although the number of data items transmitted by each mobile node is small. When k is equal to 50 and R is set as k/50 (=1), the accuracy of query result is very low. This is because each mobile node transmits only one data item with the highest score even if it holds multiple data items included in the top-k result.

4.3 Effects of d

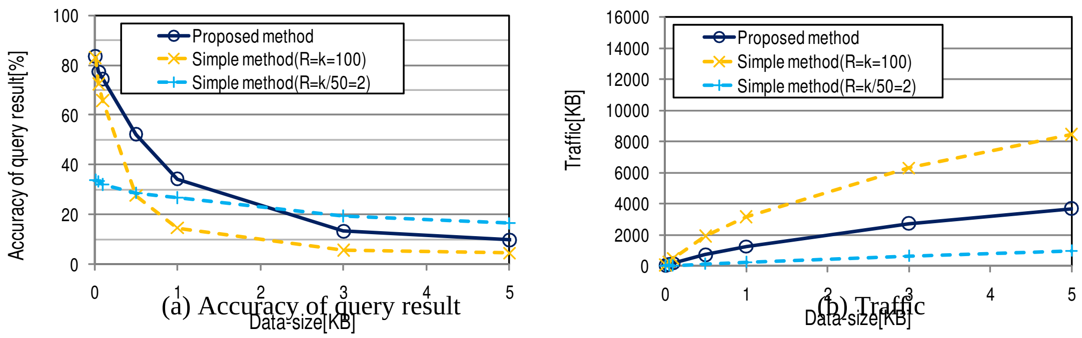
We examine the effects of the size of data items, d, on the performance. Figure 7 shows the simulation result. In this graph, the horizontal axis indicates d. The vertical axis indicates the accuracy of query result in Figure 7 (a) and the traffic in Figure 7 (b).

From this result, as d increases, the traffic becomes larger because reply size increases. Moreover, the accuracy of query result decreases due to packet losses same as the subsection4.2. When d is small, the accuracy of query result in our method is almost same as that in the simple method where R is k (=100) and the traffic is very small. However, when d is large, the accuracy of query result in our method



(a) Accuracy of query result (b) Traffic

Figure 6. Effects of k (normal distribution)



(a) Accuracy of query result (b) Traffic

Figure 7. Effects of data size (normal distribution)

becomes smaller than that in simple method. This is because the number of transmitted data items is large in our proposed method than that in simple method.

5 Conclusions

In this paper, we have proposed a query processing method for top-k query for reducing traffic and also preserving high accuracy of the query result. In our proposed method, each mobile node constructs a histogram from scores of its holding data items and estimates the distribution of scores of all data items and k-th 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. When transmitting the 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. 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. The simulation results showed that our proposed method reduces traffic and achieve high accuracy of the query result.

In this paper, we did not assume a particular type of distribution, e.g. normal distribution, Zipf distribution etc, however many data items follows particular type of distribution. If data items follows a type of distribution, it is possible to estimate the score distribution more accurately by directly estimating the parameters of the probability density function of the distribution. Moreover, we assumed that each mobile node knows the total number of data items in the entire network. However, this assumption is not always true and sometimes difficult to achieve in a real environment. Therefore, we plan to address these issues in our future work.

References

[1] 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.

[2] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing (WCMC)*, 2(5): 483–502, 2002.

[3] 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.

[4] 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.

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