Solving Sensor Void Problem in Uncontrolled Mobile Sensor Networks
Development of Mobile Sensor Database Systems

Soo Seok Park[1], Chong Sok Lim[2] and Soon J. Hyun[1]
[1]*Department of Computer Science, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea*
[2]*Department of Information and Communication Engineering, KAIST, Yuseong-gu, Daejeon 305-701, Republic of Korea*

Keywords: Sensor Database System, Mobile Sensor Networks, Query Language, Database.

Abstract: Due to the random movement of sensor nodes in uncontrolled mobile sensor networks, the distribution of sensor nodes may not be uniform over the area of network coverage. Querying a target space in which mobile sensors are unavailable at the time of the data request will lead to no return of the result. In this paper, this problem is referred to as sensor void problem, which leads to difficulties in data management and application design activities. We address the sensor void problem and offer solutions with a location-aware time-constrained query processing strategy. The key idea of the proposed strategy is to allow mobile sensor nodes to keep track of time validity and target location matching checks for the execution of the queries received from the base station. Programmers can specify a certain degree of time sensitivity in their queries according to their applications' semantics. We implemented the proposed mobile query processing scheme into our previously developed sensor database system. A performance evaluation shows how the proposed query processing strategy effectively handles the sensor void problem with various parameters of an uncontrolled mobile sensor network.

1 INTRODUCTION

With the rapid technical advancements in sensor hardware designs and wireless communications, a variety of smart sensing functions are increasingly equipped virtually in all types of objects, including smart phones, vehicles, wearable devices, unmanned flying objects, and others. A wireless sensor network (WSN) is composed of a large number of sensor nodes to monitor the physical world. Based on the mobility of the sensor node, there are two types of WSNs, the stationary sensor network and the mobile sensor network (MSN). Application services using the former include disaster management, precision agriculture, health care and traffic management. Recently, a variety of new application services, such as land, ocean and air exploration and monitoring; automobile applications; habitant monitoring; and a wide range of other scenarios have been developed (De Zoysa et al., 2007; Nittel et al., 2007). They demand data collection from mobile sensor nodes (i.e., sensors on moving objects) and form a new class of WSN known as a mobile sensor network (MSN).

There are a few different types of mobility in MSNs (Di Francesco et al., 2011). We focus on the type of MSN known as an uncontrolled mobile sensor network, which is composed of randomly moving mobile sensor nodes. This type is illustrated in Figure 1. In this type of mobile sensor network, sensors are typically attached to autonomous objects, such as vehicles, unmanned flying objects, or smart phones. A few recent studies tried to leverage these types of sensor networks to support environmental monitoring and urban monitoring applications (Abdelzaher et al., 2007; Campbell et al., 2006).

The typical problem in uncontrolled MSNs is, due to their random movement, that there may exist some target regions (e.g., grid cells) which do not have any sensor node to execute the given data collection query at the time of the application request. We refer to this situation as sensor voidance in a region. From time to time, any cell may become a sensor void region in an uncontrolled MSN, as sensor nodes move randomly. In other words, although a cell region may be a sensor void at the time of querying, some sensor nodes may come into

Seok Park S., Sok Lim C. and J. Hyun S..
Solving Sensor Void Problem in Uncontrolled Mobile Sensor Networks - Development of Mobile Sensor Database Systems.
DOI: 10.5220/0005228400470055
In *Proceedings of the 4th International Conference on Sensor Networks* (SENSORNETS-2015), pages 47-55
ISBN: 978-989-758-086-4

the target region and thus execute the associated query.

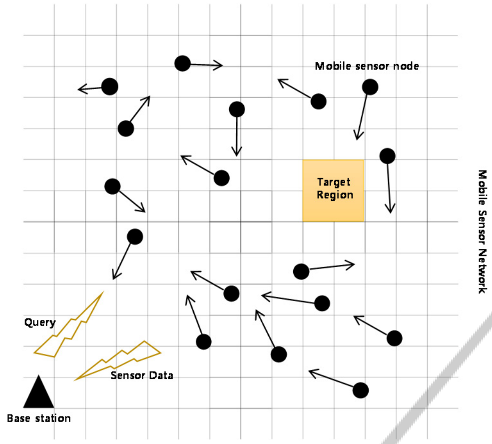

Figure 1: Mobile sensor network with randomly moving sensor nodes.

In this paper, we address the sensor void problem and offer a solution with a location-aware time-constrained query processing strategy. The proposed technique aims to maximize the probability of data acquisition from the target region for a given query. The key idea of the proposed strategy is that it allows mobile sensor nodes to check and keep track of the time validity and target location matching for the execution of the queries received from the base station. Programmers can specify a certain degree of time sensitivity in their queries according to their applications' semantics. The benefits are two-fold. First, this would give applications a greater probability of a data return from the designated target cell region by minimizing the rate of data no-returns. Second, by avoiding possibly unnecessary query executions and the collection of incorrect sensor data, mobile sensor nodes may reduce their battery consumption levels.

In our location-aware time-constrained query processing strategy, the overall procedure can be briefly described in phases, as follows. First, the base station broadcasts the application's queries to all mobile sensor nodes in the sensor network. Second, sensor nodes store them into a query table if the current location does not match the target region of the query. Third, mobile sensor nodes check the time validity and target matching for all of the queries stored in its query table. This comparing procedure is triggered in the sensor node whenever the node moves into a new cell region. Then, the node in the target region will finally fire a matching query and transmit the sensor data to the base station.

The remainder of this paper is organized as follows. We introduce the sensor database system and its major related works in Section 2. In Section 3, we propose a location-aware time-constrained query processing technique to manage the sensor void problem of an uncontrolled mobile sensor network. A performance evaluation of our work is given in Section 4, and the conclusion of this paper is given in Section 5.

2 DATA MANAGEMENT IN MOBILE SENSOR NETWORKS

In WSNs, database management is an important issue when coping with the excessive and yet mostly useless amount of streaming data and the energy drain of battery-operated sensor nodes. In sensor databases, sensor data are collected by querying the target sensors and the sensors of the target region within the sensor network. There have been many sensor database systems designed with various query languages and query processing strategies for a wide range of sensor network applications services (Diallo et al., 2013; Madden et al., 2004).

Recently, database management in mobile sensor networks (MSNs) has attracted a considerable amount of attention which has motivated a wide range of new applications (De Zoysa et al., 2007; Nittel et al., 2007). The mobility of sensor nodes is a major issue in MSNs (Zeinalipour-yazti and Chrysanthis, 2009). Valkanas et al. (2001) designed declarative queries for an in-network data analysis, which can be efficiently optimized to allow for the automatic deployment of executable code for a mobile setting. They presented a watchdog as a future direction; this is a software module that tracks sensor network changes. Andreou et al. (2011) proposed SenseSwarm, a framework for the acquisition and storage of spatio-temporal events in MSNs. SenseSwarm detects physical phenomena using a swarm of sensor nodes that are dynamically organized in the perimeter and core nodes. The two sensor database systems above assume sensor nodes with controlled mobility or sensor nodes with low mobility. In contrast to the above mobile sensor database systems, our work looks into MSNs in which mobile sensors have fully random mobility and where application queries seek sensor data from any target region.

A few recent studies leveraged human-carried or vehicle-mounted mobile sensor networks using short/mid-range radios (e.g., ZigBee, Wi-Fi, Bluetooth) for the purpose of data collection to support a wide range of new applications (Abdelzaher et al., 2007; Campbell et al., 2006). Database management in uncontrolled MSNs will frequently encounter the absence of sensor nodes in the target area at the time of the application's query request. This sensor void problem results in a no-return (or empty return) of data to the application.

The sensor void regions are never predetermined and may exist randomly over modeled cell regions. Thus, if some sensor nodes near a sensor void cell region hold requested query for a certain period of time, some of them may happen to enter the sensor void target cell region and hence execute the query. No existing works conducted an investigation of this subject by assuming that some mobile (or fixed) sensor nodes exist in all cell regions (Diallo et al., 2013; Madden et al., 2004).

3 LOCATION-AWARE TIME-CONSTRAINED QUERY IN SNQL

3.1 Time Window Operation

The situation of sensor voidance in uncontrolled MSNs is a prevalent problem due to the random mobility of the mobile sensor nodes. It will result in no return of the sensor data. In an effort to manage this problem, we designed query operations and processing techniques, known collectively as a location-aware time-constrained query processing strategy, as an extension to our previous development of a sensor network query language and processing system (Changbai et al., 2008; Lim et al., 2014). Below, we briefly explain how this strategy manages query processing tasks which take place in a sensor void target region.

The application specifies the time validity in terms of a time window in its sensor database query, and the query remains valid at sensor nodes during the specified time. Even if no sensor nodes are currently available at the time of the application's query request in the target region (i.e., sensor voidance), the query will be given a chance to be fired when any mobile node enters the target region and transmits the sensor data back to the application. To do this, the query processor will use both the location information of the mobile nodes and the time specification facility of the query language according to the query processing procedure.

Figure 2 shows the EBNF format of the location-aware time-constrained query expression added to our previous SNQL construct. The application specifies the types of sensor data to collect in the SELECT clause (1); is specifies the information of the target region in the FROM clause (2); and it sets several conditions of the sensor data with a WHERE clause (3).

```
<time-constrained-query> ::=
SELECT <select-attribute-list>                              (1)
FROM <region>                                               (2)
[WHERE <predicate>]                                         (3)
[WITHIN <percentage> | CASE WHEN … THEN … ]
COLLECTION VALID FOR <time-window>                          (4)

<time-window> ::=
<time-interval> | <time-window> <time-interval>

<time-interval> ::=                                         (5)
<lower-time-bound> | NOW,
<upper-time-bound> | NOW

<lower-time-bound> ::= <time>
<upper-time-bound> ::= <time>

<time> ::= <yyyy-MM-dd hh:mm>                               (6)
```

Figure 2: Time-window query in SNQL.

The target region can be defined as any shape using the spatial specifications and sophisticated spatial operations in SNQL (Lim et al., 2014). The spatial operator in SNQL is based on openGIS (Herring, 2011), which describes a common architecture for simple feature geometry. SNQL supports spatial assignment facilities for specifying irregularly shaped regions by name, and it provides spatial operators such as the union, intersection and minus operators to extend the expressiveness of a query language.

The COLLECTION VALID FOR clause specifies the time validity of a query in terms of the time window (4). A time window can have one or more time intervals, and a time interval is expressed by the lower-time bound and upper-time bound (5). In our present implementation of the location-aware time-constrained query processing strategy, we employ only the data type yyyy (year), MM (month), dd (day), hh (hour) and mm (minutes) (6). The NOW parameter denotes the time the query is issued. Other temporal expressions and operations can be utilized for better support of time window operations, such as after, before, intersect, difference, and union (Mkaouar et al., 2011). The design of temporal operations is beyond the scope of this paper.

3.2 Query Processing Technique

In this section, we introduce our query processing scheme, which efficiently handles the sensor void problem in uncontrolled MSNs. We describe the procedure of the proposed location-aware time-constrained query processing technique from the query dissemination to the data collection stages.

3.2.1 Network Setup

Query routing and data aggregation methods in stationary WSNs are typically found based on various types of routing trees. In MSNs, on the other hand, such routing trees can scarcely be constructed and maintained due to the mobility of the sensor nodes, such that the network topology is very likely to be transient (Zeinalipour-yazti and Chrysanthis, 2009). The routing in MSNs can be modeled in consideration of the scalability and energy efficiency. The two-tier network model is known to be feasible for both scalability and energy efficiency (Gupta and Younis, 2003). The main theme of a two-tier network model is clustering. In a cluster, there is a cluster head and a number of member nodes. The cluster head is responsible for data collection and aggregation from other member nodes. Cluster heads communicate with each other for data aggregation and to route the sensor data to the base station. Our network model uses the two-tier architecture and divides the entire sensor network into grid cell regions as uniquely identifiable query targets. Our network model is described below.

- All the sensor nodes are equal and have mobility, and they move randomly within the sensor network space N . They have maximum transmission range (MTR) which can directly communicate with other neighboring sensor node.
- Mobile sensor nodes are location-aware using GPS or some localization mechanisms (Bulusu *et al*., 2000). The location data will be evaluated against the target data of the query whenever the node enters a new cell region of grid.
- Sensor nodes are aware of the entire cell regions of the network space N, given as $2^k \times 2^k$; where $\mathrm{k} = \log_2(N)$. This information is stored as meta-data.
- The base station is stationary and its MTR covers the overall sensor network space N.
- A target cell region of a query is equal to one cell region of the grid.

Different types of mobility can significantly impact the query processing phases in MSNs (Di Francesco et al., 2011; Zeinalipour-yazti and Chrysanthis, 2009). In general, the mobility is modeled in three categories: the deterministic, partially deterministic, and fully random categories. Deterministic mobility implies that the movement of sensor nodes is predefined or can be perfectly scheduled. The partially deterministic mobility implies that the mobility patterns of mobile nodes can be predicted by analyzing the traces of sensor nodes. In the random mobility model, on the other hand, the mobility of sensor nodes is entirely unpredictable and is independent of the previous movement and, thus, the sensor void situation is most likely to arise. Modeling our uncontrolled MSNs, we assume a random mobility model, in that the sensor nodes randomly select the direction, speed and travel time (Johnson and Maltz, 1996).

The density of sensor nodes is an important parameter for the sensor void problem. The sparser they are, the more probable the sensor voidance problem becomes. We presume that high degrees of sparseness prevail in most real-world applications such that applications will encounter a high probability of the sensor void problem. We conducted an experiment to learn their co-relations, as outlined in Section 4. It should be noted that the density will not be even among the cell regions due to the random movement of sensor nodes and that the difference in the density among the cell regions may change from time to time.

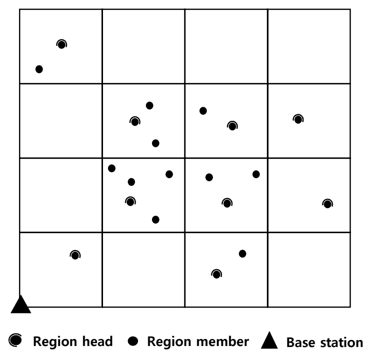

Figure 3: Sensor network topology model.

Our network model uses a two-tiered architecture, as shown in Figure 3, and each cell region may have a head node or none, and each may have zero or several member nodes. A cell region head (RH) is responsible for the collection and aggregation of sensor data from the cell region members (RM). We employ the scheme for head selection introduced in earlier work (Nasser et al., 2012). In that the mobility factor (1) is introduced to select an RH, each node keeps track of its mobility and records the number of movements it has made as well as the amount of residual energy it has. A node may change its cell region as a result of movement and join a new cell region as an RM if there is an RH in the new cell region. If the cell region is empty, the node becomes an RH in the new cell region. The number of times a node has moved over cell regions and the number of location changes along with the remaining energy are used to calculate the mobility factor (*M.F*) of the mobile nodes, as given below.

$$M.F = \frac{Z_t}{M_t} \times \frac{1}{e'} \qquad .e' > 0 \qquad (1)$$

Z_t is the total number of region changes, M_t is the total number of moves made during t seconds, and e' is the remaining energy. Each node keeps a record of its mobility factor. A lower value of the mobility factor indicates that the node is less mobile and more of a candidate to become the RH. In contrast, a higher value depicts frequent movements and renders a node inconsistent as a RH. The region head selection procedure starts at each node by broadcasting its *M.F*. This broadcast is intended for the members of the same region and is discarded by others. Initially, each node keeps its own *M.F* as the region head *M.F*. Once a broadcast is received, the node compares the region head *M.F* with the one received. If the received value is lower than the current value, the region head *M.F* and region head identifier are appropriately updated. At the end of the broadcast phase, each node has knowledge of the node with the lowest mobility factor; hence, the node is considered as the region head. The lowest value of the *M.F* ensures that the node will serve as the region head for a longer duration. Therefore, the number of procedures for the head selection process can be minimized.

In our network model, there are three different sensor node states as shown in Figure 4: idle, sensing, and discovery. A state transition is triggered by a cell region change of a sensor node or by the execution of a query. When an RM leaves its current cell region, it changes its status to discovery and broadcasts a discovery message to the RH of the new cell region. If it receives an acknowledgement from the existing RH of the new cell region, it sends its reference information, including its node identification number and network information to the RH. On the other hand, if the incoming sensor node does not receive any acknowledgement of the RH discovery message, this indicates that there is no RH in the new cell region. Thus, it must take the role of the RH of the new cell region. When an RH leaves its current cell region, it broadcasts a beacon to notify its members of its status. The RM after receiving the broadcast changes its status to discovery and elects a new RM following the same procedure defined in the 'region head selection' procedure described above. The RH is in charge of the collection and aggregation of the sensor data from the RM. The sensor node is in sensing status during the query execution phase and changes its status to the idle status after finishing the execution.

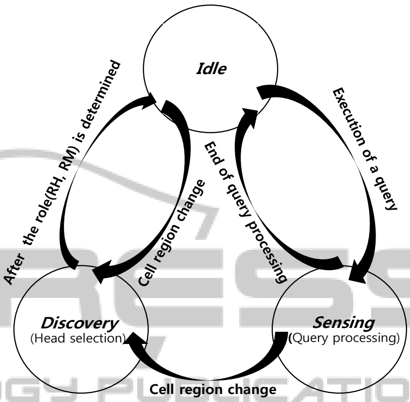

Figure 4: State transitions of a mobile sensor node.

3.2.2 Query Processing Technique

Based on the uncontrolled MSN topology and head election strategies given in Section 3.2.1, we propose a location-aware time-constrained query processing technique. The proposed query processing strategy is composed of three phases: (1) query dissemination, (2) query execution, and (3) data return. The key idea is to give the mobile sensor nodes the capability of holding queries for certain durations that are specified in the time window of the queries. As mentioned earlier, the time window can be programmed by an application designer using the SNQL query language construct for MSNs, as described in Section 3.1.

1) Query dissemination: For a query request from an application, the base station broadcasts the query to the MSNs. We assume that the maximum transmission range (MTR) of the base station is wide enough to cover the entire sensor network space. All of the sensors in the network receive the transmitted query and check its target specification and the validity of the time in the time window specification. If the target cell region of a query matches the current location of the sensor node, the RH immediately executes the query and aggregates the sensor data from the RMs. Otherwise, sensor node stores it into its query table (QT) for later evaluations. Figure 5 shows an example description of a QT.

Query ID	Region ID	Time window
0000150	754	02-09-2014 02:00 02-09-2014 03:15
0004033	367	02-08-2014 04:20 02-09-2014 04:30
. . .	. . .	. . .
0000117	019	02-09-2014 02:45 02-09-2014 03:25

Figure 5: Example of a query table (QT).

The stored queries in the QT are evaluated for possible execution in the matching target cell region at some time. By doing so, the probability of the sensor void problem (i.e., the data no-return rate) can be greatly diminished. The mobile sensor node checks the queries in the QT when it enters another cell region.

2) Query execution: When a sensor node enters a new cell region, it checks whether there is an RH in the new region. If there is an RH, the sensor node registers itself to the RH. If not, the sensor node by itself serves as the RH in the new region and checks the time validity of the query by comparing its time window specification and the current time. As the result, all of the timed-out queries will be deleted from the QT, after which target region matching is conducted by comparing the target data (i.e., the region ID in the query table) and the current location in the cell region just entered. When the time validity and region matching are found to be true, the query is executed for data collection. Figure 6 shows how the aforementioned query processing procedure is implemented.

```
Algorithm

Sensor node moves to a new cell region
Check whether there is an ZH in the new cell region
if there is an ZH in the new region then
    Register itself to the ZH in the new region
else
    Take the role of an ZH in the new cell region
    Check the time window of queries in the QT
    if there is a timed-out query then
        Delete the query from the QT
    end if
    Conduct target region matching of queries in the QT
    if there is a matching query
   (both the time validity and the region matching)
        Execute the query
        Remove the executed query from the QT
    end if
end if
```

Figure 6: Execution of queries in the QT.

3) Data return: In contrast to the query transmission procedure, the data return procedure relies on a multi-hop fashion due to the limited MTR of the sensor nodes. Furthermore, the random mobility of the sensor nodes will cause frequent link failure problems and the communication void problems (i.e., a node with no neighbor closer to the destination than itself) due to possible sensor void cell regions during the data return processes. Our data return strategy is based on a stateless geographic routing protocol for mobile ad-hoc networks (Karp and Kung, 2000). We designed the data return procedure, as follows.

In our network model, only the RHs participate in the routing process for returning sensor data to the base station. When an RM finishes its query execution, it sends the sensor data to the RH in the same cell region. The RH which receives the sensor data from the RM undertakes aggregation and starts to send the data to the base station. When an RH as the data holder returns sensor data to the base station, it uses a stateless geographic routing protocol by selecting the next data holder from among the neighboring RHs which has the highest positive forwarding cell region hop count (i.e., the neighboring RH which is closest to the base station is the next data holder). For a communication void region in which the data holder itself has the highest positive forwarding cell region hop count within the maximum transmission range, it broadcasts a beacon and holds the sensor data until there is a response message from a neighboring RH which has a higher positive forwarding cell region hop count.

At this stage, Figure 7 shows a situation in which the data holder is in a communication void situation and is moving out of the current cell region. In this situation, the data holder (RM) sends the sensor data to the RM in the new cell region. If the new cell region is empty, the sensor node becomes an RH in the new cell region. It then broadcasts a beacon and holds the sensor data until there is a response message from a neighboring RH which has a higher positive forwarding cell region hop count.

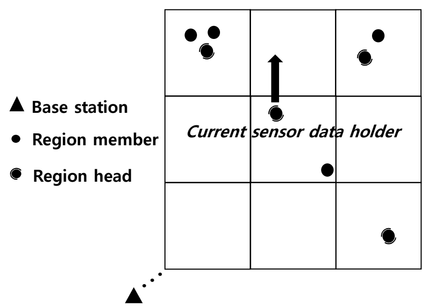

Figure 7: Example of a data holder (RM) movement in communication void situation.

4 EVALUATION

In this section, we show the experimental results of the proposed location-aware time-constrained query processing technique based on a grid-based two-tier network model. Our experiment includes a measurement of the occurrence ratio of the sensor void problem. We also examined the ratio of successful query executions in association with the number of sensor nodes in a sensor network (i.e., the node density), the velocity of the sensor nodes, and the amount of validity time in terms of the time window sizes.

For the experiment with the proposed location-aware time-constrained query processing strategy, we used MobiSim to generate a mobility trace (Mousavi et al., 2007). MobiSim is a java-based mobility management utility which was mainly designed to investigate mobility issues in mobile ad-hoc networks. We used a grid-based two-tier network model and analyzed the mobility traces of the sensor nodes.

Table 1: Experiment settings.

Parameters	Values
Mobility pattern	Random waypoint
Network size	1000 x 1000
Region size	100 x 100
Simulation(s)	10,000
Number of trials	50
Min speed(m/s)	1
Time window interval (s)	Simulation dependent
Number of nodes	Simulation dependent
Max speed(m/s)	Simulation dependent

Table 1 shows the MobiSim experimental parameters used here. The sensor network size was set to 1000m * 1000m with a cell region size of 100m * 100m. We assume that the sensors have random mobility. We set the minimum speed of the sensor node to 1m/s with a maximum pause time of 100 seconds. In the experiment, we vary the number of sensor nodes, the maximum speed of the sensor node and the time window intervals. This result was produced by averaging 50 experiments for each parameter variable.

First, we examined the occurrence of a sensor void situation depending on the number of sensor nodes in the sensor network. We set a single cell region as the target region to be queried. The target region is randomly selected in this experiment. As shown in Figure 8, as the number of sensor nodes increases, the ratio of sensor voidance decreases. This is obvious, because the sensor void situation is quite proportional to the average density of mobile sensor nodes, that the size of the cell region and the size of the network space will show similar ratios.

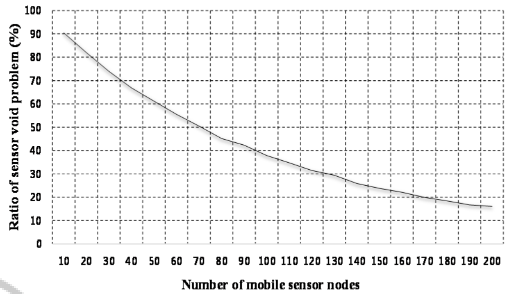

Figure 8: Sensor voidance ratio on network density.

Second, we examined the average time of sensor void recovery depending on the size of the time-window interval. This experiment shows that the rate of successful query execution depends on the amount of time validity specified in the time windows, as shown in Figure 9. For this experiment, we set the maximum speed of the sensor node to 5m/s and number of sensor nodes to 10. We verified that a longer validity time led to a higher the rate of successful query executions, as the time window interval becomes large and thus the probability of the inflow of the sensor node in the target region is increased.

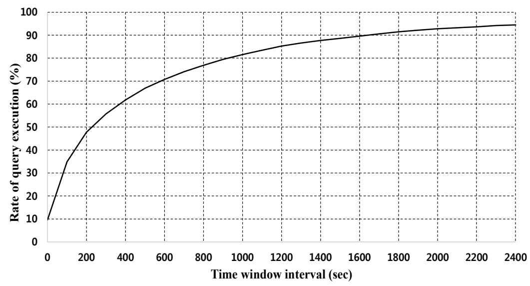

Figure 9: Query execution rate on time window intervals.

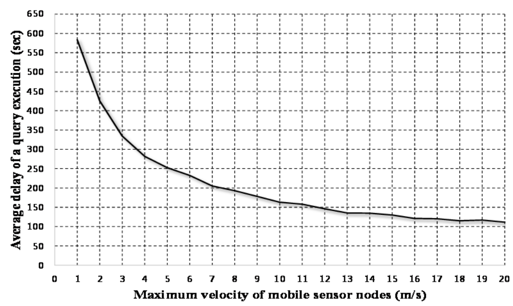

Figure 10: Average delay of a query execution on the maximum velocity of mobile sensor nodes.

Finally, we conducted an experiment on how the maximum velocity of mobile sensor nodes can affect the recovery rate of the sensor void situation. We increased the maximum velocity of the sensor node from one to twenty and checked the average delay in the query execution. Figure 10 shows that a faster mobile sensor node leads to a lower average delay for query executions. This is derived from the fact that the sensor is more likely to flow into the target region if the sensor has higher velocity.

5 CONCLUSIONS

In this paper, we propose a sensor database system for MSNs with uncontrolled mobile sensor nodes. In contrast to a stationary sensor network, the availability of sensor nodes in the target region is not guaranteed in a mobile sensor network, and the random mobility issue gives rise to difficulties in sensor database management, which is referred to as the sensor void problem in this paper. In this paper, we have proposed a location-aware time-constrained query processing technique which is highly effective for handling the sensor void situation of sparse MSNs with uncontrolled mobile sensor nodes. We have demonstrated the proposed query processing procedure in operational phases for query dissemination, execution, data collection and aggregation, and data return. Our experiments show that various mobility parameters are correlated with the occurrence rates of sensor void situations. Finally, we plan to develop our query processing strategy further and implement more functions into our mobile sensor network database management system.

ACKNOWLEDGEMENTS

This work was supported by ICT R&D program of MSIP/IITP. [B0101-14-0334, Development of IoT-based Trustworthy and Smart Home Community Framework]

REFERENCES

Abdelzaher, T., Anokwa, Y., Boda, P., Burke, J. A., Estrin, D., Guibas, L., Reich, J., 2007. *Mobiscopes for Human Spaces*. Pervasive Computing, 6(2), pp.20–29.

Andreou, P., Zeinalipour-Yazti, D., Chrysanthis, P. K., Samaras, G., 2011. *In-network data acquisition and replication in mobile sensor networks.* Distributed and Parallel Databases, 29(1-2), pp.87–112.

Bulusu, N., Heidemann, J., Estrin, D., 2000. *GPS-less low-cost outdoor localization for very small devices.* Personal Communications, IEEE, 7(5), 28-34.

Campbell, A. T., Eisenman, S. B., Lane, N. D., Miluzzo, E., Peterson, R. A., 2006. *People-centric urban sensing*. Proceedings of the 2nd annual international workshop on Wireless internet - WICON '06, p.18–es.

Changbai, C., Jaehyoung, L., Juyeon, H., Insung, J., Minsoo, K., Hyun, S. J., 2008. *SNQL: A Query Language for Sensor Network Databases.* In Proceedings of the 7th WSEAS International Conference on Telecommunications and Informatics, pp. 114-119.

De Zoysa, K., Keppitiyagama, C., Seneviratne, G. P., Shihan, W. W. A. T., 2007. *A public transport system based sensor network for road surface condition monitoring.* Proceedings of the 2007 workshop on Networked systems for developing regions - NSDR '07, p.1.

Di Francesco, M., Das, S.K., Anastasi, G., 2011. *Data Collection in Wireless Sensor Networks with Mobile Elements.* ACM Transactions on Sensor Networks, 8(1), pp.1–31.

Diallo, O., Rodrigues, J. J. P. C., Sene, M., Lloret, J., 2013. *Distributed database management techniques for wireless sensor networks.*

Gupta, G., Younis, M., 2003. *Load-balanced clustering of wireless sensor networks.* IEEE International Conference on Communications, 2003. ICC '03., 3, pp.1848–1852.

Herring, E.J.R., 2011. *Open Geospatial Consortium Inc . Status : Corrigendum Category : OpenGIS ® Implementation Standard OpenGIS ® Implementation Standard for Geographic information - Simple feature access - Part 1 : Common architecture.*

Johnson, D. B., Maltz, D. A., 1996. *Dynamic source routing in ad hoc wireless networks.* In Mobile computing, pp. 153-181, Springer US.

Karp, B., Kung, H.T., 2000. *GPSR : Greedy Perimeter Stateless Routing for Wireless Networks.* In Proceedings of the 6th annual international conference on Mobile computing and networking, pp. 243-254.

LIM, C. S., LEEb, J. H., PARK, M., HYUN, S. J., 2014. *Design and Implementation of Spatial Operators and Energy-efficient Query Processing Strategy in Wireless Sensor Network Database System.*

Madden, S.R., Franklin, M.J., Hellerstein, J.M., 2004. *TinyDB : An Acquisitional Query Processing System for Sensor Networks* 1. , V(212).

Mkaouar, M., Bouaziz, R., Moalla, M., 2011. Querying and manipulating temporal databases.

Mousavi, S. M., Rabiee, H. R., Moshref, M., Dabirmoghaddam, A., 2007. *Mobisim: A framework for simulation of mobility models in mobile ad-hoc networks. In Wireless and Mobile Computing, Networking and Communications*, 2007. WiMOB 2007, pp. 82-82.

Nasser, N., Al-Yatama, A., Saleh, K., 2012. *Zone-based routing protocol with mobility consideration for wireless sensor networks.* Telecommunication Systems, 52(4), pp.2541–2560.

Nittel, S., Trigoni, N., Ferentinos, K., Neville, F., Nural, A., Pettigrew, N., 2007. *A drift-tolerant model for data management in ocean sensor networks.* Proceedings of the 6th ACM international workshop on Data engineering for wireless and mobile access - MobiDE '07, p.49.

Valkanas, G., Gunopulos, D., Galpin, I., Gray, A. J., Fernandes, A. A., 2011. *Extending query languages for in-network query processing.* Proceedings of the 10th ACM International Workshop on Data Engineering for Wireless and Mobile Access - MobiDE '11, p.34.

Zeinalipour-Yazti, D., Chrysanthis, P. K., 2009. *Mobile sensor network data management*. In Encyclopedia of Database Systems (pp. 1755-1759), Springer US.