Query-Driven Data Collection and Data Forwarding in Intermittently Connected Mobile Sensor Networks

Wei Wu
School of Computing
National University of Singapore
wuw@nus.edu.sg

Hock Beng Lim
Intelligent Systems Centre
Nanyang Technological University
limhb@ntu.edu.sg

Kian-Lee Tan
School of Computing
National University of Singapore
tankl@comp.nus.edu.sg

ABSTRACT

In sparse and intermittently connected Mobile Sensor Networks (MSNs), the base station cannot easily get the data objects acquired by the mobile sensors in the field. When users query the base station for specific data objects, the base station may not have received the necessary data objects to answer the queries. In this paper, we propose to use a Mobile Data Collector (MDC) to collect the data objects from the mobile sensors that the base station needs for answering queries. To facilitate the MDC's data collection, we design a location-based data forwarding protocol that exploits the location metadata of data objects and uses caching to improve data availability in the MSNs. Results of performance study show that our solution can reduce query response times on the base station.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Distributed databases*

General Terms

Algorithms, Experimentation

Keywords

Mobile sensor network, data collection, query processing

1. INTRODUCTION

Mobile sensor networks (MSNs) are very useful in reconnaissance, disaster rescue, and environment monitoring tasks. For examples, mobile sensors can be used to gather information in battlefields and earthquake areas.

In most applications of MSNs, users (such as commander or rescue personnel) will want to access the data objects acquired by the mobile sensors. They query the base station for the data objects they want. Answering these queries timely is very desirable (sometimes even critical).

DMSN '10, September 13, 2010, Singapore

Unfortunately, in sparse MSNs where the sensors and the base station are only intermittently connected, many queries may only be answered after a quite long time. Due to the intermittent connections, it is difficult for the mobile sensors to send data objects to the base station and for the base station to pull data objects from the sensors.

Mobile data collectors (MDCs) [4], [10], [12] [3] are widely used to collect data objects in sensor networks. A MDC collects data objects by moving in the sensor networks, getting data objects from sensors within wireless communication range, and moving back to the base station.

In this paper, we propose to use a MDC to do *query-driven* data collection in sparse MSNs to reduce the average query answering time. When the base station receives a query but does not have the data object that the query requests for, it lets the MDC to collect the data object. We focus on data collection for answering *spatial queries* that explicitly request for data objects that are acquired at specific locations by the mobile sensors.

The challenge of query-driven data collection in intermittently connected MSNs is that the base station and the MDC do not know which mobile sensor has the data object they want. Due to disconnections, they cannot simply flood a message to all the sensors and find this out.

We design a query-driven data collection solution called `F4C` (Forwarding for Collection). Our main idea is to use spatial information of data objects to direct data forwarding and use spatial information of queries to direct data collection so that the MDC will have a good chance of meeting a mobile sensor that carries the data object that it needs.

In `F4C`, when the mobile sensors forward a data object acquired at location l to the base station, they keep (when possible) that data object in a region along the shortest physical path from l to the base station. When a MDC needs to collect a data object that was acquired at l, it simply moves towards l along the shortest path from the base station to l. The mobile sensors make data forwarding decisions that reduce the distance the MDC needs to move before it can get the data it wants. Furthermore, caching is used to increase the data availability among the mobile sensors. Through results from simulation, we show that `F4C` can reduce the average query answering time on the base station.

The remainder of the paper is organized as follows. We describe the system model in Section 2, and present `F4C` in Section 3. Results of performance study are shown in Section 4. Related work is briefly discussed in Section 5. We conclude this paper and list the directions of future work in Section 6.

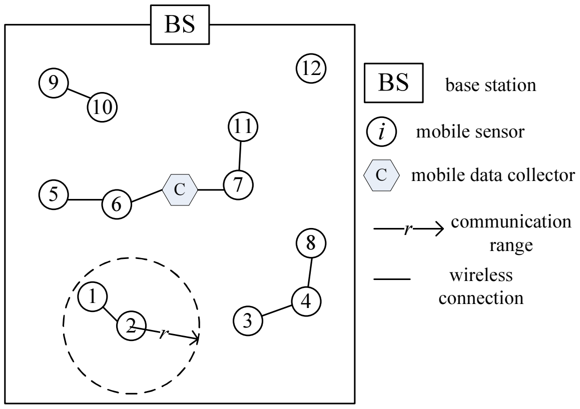

Figure 1: Example of a sparse MSN where a MDC is used to do data collection.

2. SYSTEM MODEL

2.1 Mobile Sensor Network (MSN)

The system consists of a stationary base station BS, n mobile sensors (s_1,s_2,...,s_n), and a mobile data collector (MDC) in a task field A. They use wireless technology such as WiFi for communication. Two nodes (we use "node" to refer to a sensor, the BS, or a MDC) can communicate *directly* only if the distance between them is smaller than the wireless technology's communication range r.

The mobile sensors move in the task field following a certain mobility model. Their move speed is v. We assume that all mobile nodes have GPS equipped so they always know their own locations. The BS is stationary and its location is known to all mobile nodes.

This work assumes a mobile sensor network that is sparse and only intermittently connected due to low sensor density (and/or short communication range) and sensor movement. Figure 1 shows an example of an intermittently connected MSN where there are twelve mobile sensors.

2.2 Data Objects

The mobile sensors acquire data objects when they move in the task field. The location where a data object is acquired is kept as a part of the data object's metadata. We use D_p to refer to a data object that is acquired at location p. In addition, a spatial region is also associated with the data object. It is determined by the location p and the sensor's sensing range, and is the geographical region whose feature is captured in the data object. For example, if the data object is an image, then the spatial region associated with it is the area captured in the image.

2.3 Queries

Users of the system query the base station for data objects by spatial predicates. For simplicity, we assume each query asks for one data object acquired at a specific location. We use *query location* to refer to the location specified in a query.

A data object can be used to answer a query if the data object's spatial region covers the query location. The time duration from the base station gets a query to the base station answers the query is the query's *response time.*

2.4 Data Forwarding

The mobile sensors always try to forward their data objects to the base station. They forward data objects in a carry-and-forward fashion [2], because the sensor network is only intermittently connected. When a sensor acquires a data object or receives a data object from a neighbor (a *neighbor* refers to a node within communication range) but has no suitable neighbor to forward it to, the sensor carries the data object and tries to forward it later.

In sparse MSNs, the sensors have only limited communication opportunities to forward data objects. For this reason, we assume that a mobile sensor does not forward a data object multiple times.

The mobile sensors make their data forwarding decisions based on a data forwarding algorithm. We will describe our location-based data forwarding algorithm in Section 3.

2.5 Data Collection

The MDC's job is to collect data objects from the mobile sensors. When there is no pending queries on the base station, the MDC moves in the task field to collect data objects from mobile sensors and periodically returns to the BS. After returning to the BS, the MDC sends the data that it has collected to the BS.

If the BS has pending queries when the MDC returns to it, the BS sends a query to the MDC and lets it collect a data object for the query. We call this *query-driven data collection*, and use *mission query* to refer to the query that the base station sends to the MDC.

In this paper we look at the problem of data collection for *one* query. In future work, we will study the problem of data collection for *multiple* queries.

We assume that the MDC also has sensing capability. If the MDC failed to get a data object from the sensors for the mission query, it can move to the query location to acquire a data object for the query.

3. F4C: FORWARDING FOR COLLECTION

`F4C` (Forward for Collection) is specially designed for query-driven data collection in intermittently connected MSNs. Its goal is to reduce the distance that the MDC needs to move before it gets a data object for the mission query.

We define two terms in `F4C`: a data object's *collection path* and *forwarding region.* The MDC collects a data object by moving on the data object's collection path. The forwarding region is an area along the collection path. The mobile sensors keep a data object in its forwarding region when they forward the data object towards the base station.

3.1 Collection Path and Forwarding Region

For a data object D_p acquired at location p, we define the shortest physical path in the field from the base station to p as D_p's *collection path*, and the union of the points in the field whose distances to D_p's *collection path* are shorter than r (the wireless communication range) as D_p's *forwarding region.* We will use $Path(D_p)$ to denote D_p's collection path, and $Region(D_p)$ to denote D_p's forwarding region.

Figure 2 shows an illustration of a data object's collection path and forwarding region in a field where there are no obstacles. If there are obstacles in the filed, the collection path may not be a straight line. For simplicity, in this paper we will use examples where there are no obstacles in the field. Note that `F4C` also applies to fields where there are obstacles.

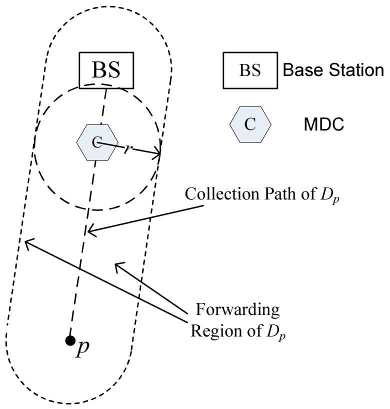

Figure 2: Illustration of a data object's Collection Path and Forwarding Region.

Note that D_p's collection path is determined by the location of BS and p (the location metadata of D_p), and D_p's forwarding region is determined by its collection path and the wireless communication range r. The idea is that when the MDC moves along D_p's *collection path* it will encounter the mobile sensors in D_p's *forwarding region*.

A data object's collection path and forwarding region are fixed and they are independent of the mobile sensor that is currently carrying it. Since the location of the BS is known to all the mobile nodes, and data objects have location metadata, the mobile sensors can compute their data objects' collection paths and forwarding regions by themselves. Given a query, the MDC can also immediately compute the collection path of the data object that the query requests for.

3.2 Query-Driven Data Collection

When the BS has a pending query that requests for a data object acquired at location l and the MDC is connected to the BS, the BS sends the query to the MDC and lets it collect a data object for the query.

In F4C, the MDC's process of query-driven data collection is very simple. It simply moves from the BS towards l on $Path(D_l)$. When the MDC encounters a mobile sensor, it queries the mobile sensor for a data object that can answer the mission query. If the mobile sensor has such a data object, it sends the data object to the MDC. After receiving the data object, the MDC moves back to the BS. If none of the sensors that the MDC encountered has a data object that is an answer to the mission query, the MDC will arrive at l. The MDC acquires a data object by itself at l, and moves back to the BS.

Note that although the MDC's main task in a query-driven data collection mission is to collect a data object for the mission query, it also collects other data objects when it encounters the mobile sensors. We will elaborate on this in Section 3.4.2.

The time cost of collecting a data object for a query is roughly twice the time from it got the query to it gets a data object that can answer the query. In the worst case, the MDC arrives at the query location and then acquires such a data object. Reducing the distance that the MDC needs to move before it gets a data object for the mission query will be an effective way to reduce the time cost of query-driven data collection.

3.3 Location-based Data Forwarding

The general idea of data forwarding in F4C is keeping the data objects in their own forwarding regions when the mobile sensors forward the data objects towards the base station. The objective is to maintain a data object's availability in its forwarding region so that the MDC can easily get the data object by moving on its collection path.

Suppose sensor s_i currently carries a data object D_p. s_i computes D_p's forwarding region using D_p's location metadata and the location of the BS (which is known to all the sensors). s_i knows its own location and exchanges location information with neighbors periodically. s_i makes different forwarding decisions for D_p based on (1) whether it is inside D_p's forwarding region and (2) neighbors' location.

- If s_i is in D_p's forwarding region, s_i forwards D_p to a sensor that is also in D_p's forwarding region but is nearer to the base station.

- If s_i is not in D_p's forwarding region, s_i forwards D_p to a sensor that is in D_p's forwarding region. If none of s_i's neighbors is in D_p's forwarding region, s_i forwards D_p to a sensor that is closer to D_p's forwarding region.

In both cases, if none of s_i's neighbors satisfies the conditions, s_i carries the data object and tries to forward it later.

3.4 Prioritize Data Objects in Forwarding

In sparse MSNs, each mobile sensor will be carrying many data objects, because the sensor keeps acquiring data objects in the field but has limited opportunities to forward data objects to the BS or the MDC. Therefore, when a sensor encounters a neighbor, it will have more data objects than it can send to the neighbor during the short connection duration with the neighbor. The sensor has to decide what data objects should be forwarded to the neighbor. Based on whether the neighbor is a mobile sensor or the MDC, different algorithms are used to select the data objects for forwarding. Both algorithms are optimized to help reduce the distance that the MDC needs to move before getting a data object for the mission query.

3.4.1 Forwarding to a Neighboring Sensor

To help the sensors decide which sensor should carry what data object, we define a measure called *collection-distance* and let the sensors make data forwarding decisions based on their collection-distance of the data objects. Given a data object, a sensor's collection-distance is the distance that a DMC needs to move to get the data object if the sensor carries that data object.

Let $Circle(s_i, r)$ denote the circle centered at the location of s_i with radius r which is the wireless communication range. Given a data object D_p, if $Circle(s_i, r)$ does not intersect with $Path(D_p)$, s_i's collection-distance for D_p is defined as the length of $Path(D_p)$; otherwise, let I be the intersection of $Circle(s_i, r)$ and $Path(D_p)$ that is closer to the BS, s_i's collection-distance for D_p is the distance from the BS (along $Path(D_p)$) to I.

Figure 3 illustrates the definition of collection-distance with two examples. $Circle(s_1, r)$ does not intersect with $Path(D_p)$, so s_1's collection-distance for D_p is the length of $Path(D_p)$. $Circle(s_2, r)$ intersects with $Path(D_p)$ at I_1 and

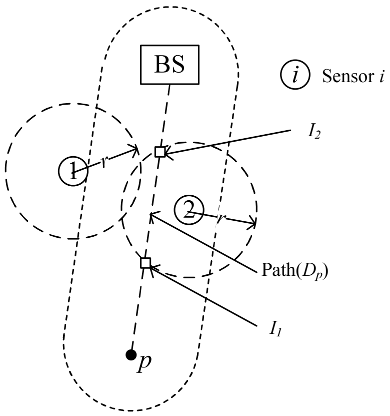

Figure 3: Illustration of forward distance definitions.

I_2. I_2 is closer to the BS, so s_2's collection-distance for D_p is the distance from the BS to I_2 along $Path(D_p)$.

Intuitively, a sensor's collection-distance for a data object is the distance that the MDC needs to move alone the data object's collection path before it can get the data object from the sensor. When a sensor is outside a data object's forwarding region, its collection-distance for the data object is the length of the data objet's collection path.

Given two sensors s_i and s_j, a data object D_p, and let $cd(s_i, D_p)$ and $cd(s_j, D_p)$ denote s_i and s_j's collection-distances for D_p, we define ($cd(s_i, D_p)$ - $cd(s_j, D_p)$) as the *delta-collection-distance* between s_i and s_j for D_p.

When s_i encounters s_j, s_i uses delta-collection-distance for the data objects (that s_i carries) to prioritize the forwarding of the data objects. First, the data objects whose delta-collection-distances are positive are considered for forwarding. s_i keeps sending to s_j the data object for which the delta-collection-distance between s_i and s_j is the largest. This process goes on until s_i and s_j are not connected any more or no data object has a positive delta-collection-distance.

For the data objects whose delta-collection-distance is zero, the ones for which s_i is outside their forwarding regions are considered for forwarding. Since s_i and s_j have the same collection-distance for these data objects, s_j must also be outside the objects' forwarding regions. s_i forwards a object to s_j if s_j is closer to its forwarding region. Due to space limitation, we do not further elaborate on this.

Note that we are considering single-path forwarding where a sensor forwards a data object only once. Once s_i sends a data object to s_j, s_i may remove the data object from its storage. Also note that s_j may also forward data objects to s_i. Our design is at the application layer and assume that the allocation of communication slots is controlled by a MAC (Media Access Control) layer protocol.

3.4.2 *Forwarding to the MDC*

During the MDC's query-driven data collection missions, the MDC collects not only data objects for the mission queries but also other data objects. When the MDC encounters a mobile sensor, the mobile sensor can send data objects to the MDC as long as they are connected.

A mobile sensor prioritizes the data objects for forwarding to the MDC by the lengths of their collection paths. The data object whose collection path is the longest gets first forwarded to the MDC. For example, suppose sensor s_i carries data object D_a and D_b, and the collection path of $Path(D_b)$ is longer than $Path(D_a)$, and s_i can only forward one data object to the MDC due to limited connection time. s_i will forward D_b to the MDC.

The rationale behind this design is that the data objects with longer collection paths are more difficult for the MDC to collect if there is a query in the future requesting for it. Recall that in the worst case the MDC has to move to the query location to acquire a data object for the query. By forwarding the data objects with longer collection paths to the MDC, the base station will get (from the MDC) these data objects and will be able to answer the queries that request for such data objects. The queries requesting for data objects closer to the BS are easier to be answered because it is easier for the MDC to collect these data objects even if they are not in their forwarding regions.

3.5 Caching

In `F4C`, the sensors do their best to keep a data object in its forwarding region, but sometimes the sensor carrying the data object may move out from the data object's forwarding region and none of its neighbors is in the data object's forwarding region. To improve the chance that the MDC encounters a sensor that has the data object which the MDC is looking for, we use caching to further improve the data availability among the mobile sensors.

After a sensor s_i forwards a data object to a neighboring sensor s_j, s_i does not delete the local copy of the data object but keeps it as a caching in local storage. s_i will not forward the copy of the data object to any other sensor (because we are considering single-path data forwarding rather than multiple-path data forwarding), but can send it to the MDC if the MDC needs this data object for answering its mission query. Recall that when the MDC encounters a sensor, the MDC will check whether the sensor has a data object for the query.

4. PERFORMANCE STUDY

We study the performance of `F4C` through simulation. The aim of the experiments is to investigate whether `F4C` can help reduce the average query answering time at the base station and how the system parameters (such as the number of sensors, size of data object, etc) will affect `F4C`'s performance.

Since we are not aware of any existing query-driven data collection scheme for sparse MSNs, we compare `F4C` to a solution where the MDC moves towards the query location (as in `F4C`) and the mobile sensors use geographical greedy routing [9] in data forwarding. We choose geographical greedy routing for comparison because it has been regarded as a very effective data forwarding algorithm in sensor network. In experiments we call this method GeoGreedy. In GeoGreedy, only the locations of neighbors are considered and a sensor always forwards data objects to a neighbor that is closer to the BS (no matter whether the neighbor is in the data objects forwarding regions).

We designed a simulation package for mobile sensor networks and implemented it in Java. In our simulation experiments, n mobile sensors are initially randomly placed in a 600 Meters * 600 Meters field and they move in the field at speed v according to a random waypoint mobility model. The move speed of the MDC is $2 * v$. Each sensor acquires a data object every T_s seconds. The size of a data object

Table 1: System Parameters

Parameter	Unit	Default	Range
number of sensors n		30	20 - 50
move speed v	Meters/s	2	1 - 8
data size D	KB	500	100 - 1000
sense interval T_s	seconds	20	10 - 60
query interval T_q	seconds	20	10 - 60
communication range r	Meter	100	50 - 150

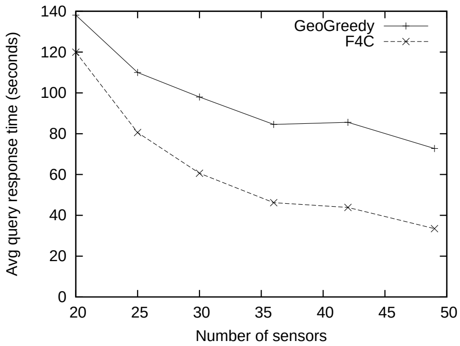

Figure 4: Effect of the number of sensors on average query response time.

is D KB. The BS receives a query every T_q seconds. 100 queries with random query locations in the field are issued to the base station. The wireless communication bandwidth is 2Mbps and the communication range is r Meters. Table 1 lists the parameters and their values.

The performance measure is the average query answering time on the base station. Both the queries that are answered by the base station right away and the queries answered through the MDC's query-driven data collection are accounted for in the calculation of the average query answering time.

In each set of experiments we vary the value of one parameter and study its effect on F4C and GeoGreedy. Due to space limitation, we will present only some representative experiment results.

4.1 Effect of the Number of Sensors

The number of mobile sensors immediately affect the density of sensors in the field and the connectivity of the sensor network. Figure 4 shows the effect of the number of sensors on the performance of F4C and GeoGreedy. We see that F4C reduces the average query response time and the improvement over GeoGreedy is between 15% to 50%. When there are very few (e.g., 20) sensors moving in the field, the sensor network is very sparse and most of the times a sensor has no neighbor to forward its data objects to. The sensors cannot keep data objects in their forwarding regions and the MDC needs to move to the query locations to get data objects for most of the mission queries. As the number of sensors increases, F4C can effectively direct the sensors to forward data objects to neighbors in data objects' forwarding regions, so F4C starts to clearly outperform GeogGreedy.

F4C outperforms GeoGreedy because in F4C a data objects' availability along its collection path is generally higher than that in GeoGreedy.

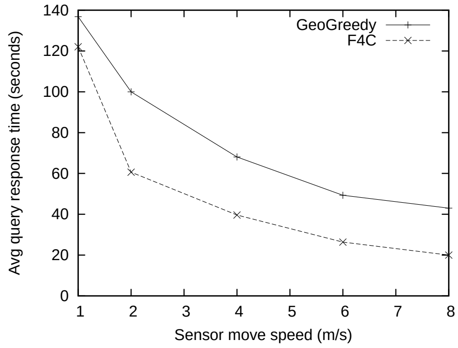

Figure 5: Effect of sensor's move speed on average query response time.

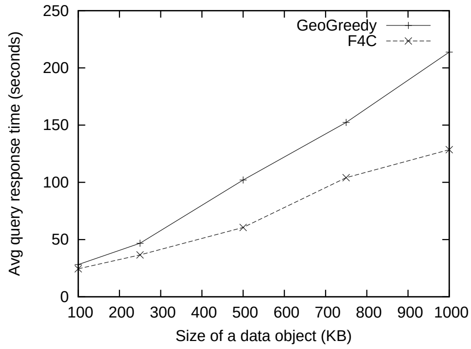

Figure 6: Effect of data object size on average query response time.

4.2 Effect of Sensors' Move Speed

The effect of sensors' move speed on the performance of F4C and GeoGreedy is shown in Figure 5. We observe that F4C consistently outperform GeoGreedy, and the average query answering times decrease as the sensors' move speed increases. Sensors' move speed affects the number of neighbors that a sensor will encounter during a period of time. When the sensors and the MDC move at a faster speed, they encounters new neighbors more frequently but have shorter connection time with the neighbors, and the MDC can arrive at the query locations in shorter time. In F4C, the mobile sensors exploit the encounters and keep data objects in their forwarding regions by sending carefully selected (through prioritization) data objects to the neighbors.

4.3 Effect of the Size of a Data Object

Figure 6 shows how the size of a data object will affect the performance of F4C and GeoGreedy. The average query answering times in both F4C and GeoGreedy are longer when the data objects are bigger. This is because as the data object size increases a sensor can forward a smaller number of data objects to a neighbor during their limited connection time. This not only means that a smaller number of data objects can be kept in their forwarding regions but also means that the MDC will receive a smaller number of data objects from the sensors. As a result, the BS will get less data objects from the MDC and more queries need to be answered through the MDC's data collection in the field.

4.4 Effect of Other Parameters

Due to space limitation, we will not present in detail the experiment results on communication range, sense interval, and query interval. All experiment results show that F4C results in shorter average query answering time when compared to GeoGreedy. Longer communication range makes the sensor network better connected and lets the nodes have more time for communication, and thus has a positive effect on average query answering times. Longer sense interval and longer query interval also have positive effects on average query answering times. Longer sense interval means the sensors will gather smaller amount of data and it will be easier for them to keep the data objects in forwarding regions or send to the MDC. Longer query intervals means that the BS will receive more data before it receives a new query so it is more likely that the query can be answered right away.

5. RELATED WORK

Several routing protocols designed for mobile ad-hoc networks (MANET) and wireless sensor networks make use of mobile nodes' location information. The well-known examples include *compass routing* [7], DREAM (distance routing effect algorithm for mobility) [1], LAR (location-aided routing) [6], GPSR (greedy perimeter stateless routing) [5], and GEAR (Geographical and Energy Aware Routing) [11]. [9] investigated the performance of geographic greedy routing algorithms in sensing-covered networks and showed that simple greedy geographic routing is an effective routing scheme in many sensing-covered networks.

The data forwarding method proposed in this paper differs from existing location-based routing protocols in that it exploits not only the location of the mobile nodes but also the location information of the data objects. Furthermore, our data forwarding method is specially designed to facilitate a MDC's query-driven data collection.

Studies on mobile data collectors in sensor networks are also related to this work. In existing work on mobile data collection [8], [13], [4], [10], [12] [3], however, the focus is to minimize the energy consumption of the sensors. The mobile data collectors in existing work either have fixed collection routes or move randomly in the field. The main difference between this work and existing work on mobile data collectors is that we focus on data collection for query answering on the base station and use query location to direct the movement of the mobile data collectors.

6. CONCLUSION AND FUTURE WORK

In sparse and intermittently connected mobile sensor networks, the base station cannot easily get data objects from the mobile sensors to answer the queries received from the users. We propose to use a mobile data collector (MDC) to do query-driven data collection so that the base station can answer the queries more timely. We present a data forwarding and data collection solution called F4C (Forwarding for Collection) that reduces the time that the MDC needs to move before it gets data objects for queries. In F4C, a MDC collects a data object for a query by simply moving from the BS towards the query location, and the mobile sensors keep data objects available in regions along the data objects' collection paths. The mobile sensors make data forwarding decisions based on their locations and the data objects' location metadata. The algorithms that the mobile sensors use to prioritize the data objects for forwarding are designed to help reduce average collection time. We did simulation to study the performance of our proposed solution. Experiment results show that F4Ccan help reduce average query processing time on the base station.

This preliminary work only considers spatial information of the query and data objects. A future work is to take query and data objects' temporal information also into account. In F4C, single-path data forwarding is assumed. Another direction of future work, therefore, is to consider multi-path forwarding and study the trade-off between data availability and data traffic. Last but not the least, we believe that it will be interesting to design an algorithm for the MDC to collect data for multiple queries.

7. ACKNOWLEDGMENTS

This work is partially supported by a research grant R-252-000-352-232 (TDSI/08-001/1A) from the Temasek Defense Systems Institute.

8. REFERENCES

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). In *MobiCom '98*, 1998.

[2] K. W. Chen. Cafnet: A carry-and-forward delay-tolerant network. MIT Master Thesis, 2007.

[3] M. D. Francesco, K. Shah, M. Kumar, and G. Anastasi. An adaptive strategy for energy-efficient data collection in sparse wireless sensor networks. In *EWSN*, 2010.

[4] Y. Gu, D. Bozdağ, R. W. Brewer, and E. Ekici. Data harvesting with mobile elements in wireless sensor networks. *Comput. Netw.*, 50(17), 2006.

[5] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MOBICOM*, 2000.

[6] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4), 2000.

[7] E. Kranakis, S. O. C. Science, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Canadian Conference on Computational Geometry*, 1999.

[8] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *IEEE SNPA Workshop*, 2003.

[9] G. Xing, C. Lu, R. Pless, and Q. Huang. On greedy geographic routing algorithms in sensing-covered networks. In *MobiHoc*, 2004.

[10] G. Xing, T. Wang, W. Jia, and M. Li. Rendezvous design algorithms for wireless sensor networks with a mobile base station. In *MobiHoc*, 2008.

[11] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report, UCLA, 2001.

[12] M. Zhao and Y. Yang. Bounded relay hop mobile data gathering in wireless sensor networks. In *MASS*, 2009.

[13] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc*, 2004.