Querying Sensor Networks Using Ad-hoc Mobile Devices: A Two Layer Networking Approach[1]

Shourui Tian[†], Sol M. Shatz[†], Yang Yu[‡] and Juzheng Li[†]

[†]Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60644 USA
stian,shatz@cs.uic.edu

[‡]Pervasive Platforms and Architecture Lab
Application Research Center, Motorola Labs
Schaumburg, IL 60196 USA
yang@motorola.com

***Abstract*—An interplay between mobile devices and static sensor nodes is envisioned in the near future. This will enable a heterogeneous design space that can offset the stringent resource and power constraints encountered in traditional static sensor networks by taking advantage of the more powerful mobile devices. We present a systematic framework for end-to-end query processing, using a two-layer architecture that consists of mobile devices at the upper layer and static sensor nodes at the bottom layer. The framework employs a "PULL" query model that contains staged operations including query generation, query routing, query injection, and query result routing. Each of these stages of query processing is discussed with an emphasis on techniques for energy-efficient query injection and query result routing with location-ignorant sensor nodes. The techniques leverage the mobility and transmission flexibility of mobile objects at the upper layer. Numeric and simulation results are provided to support the proposed methods.**

***Index Terms*—mobile objects, opportunistic query processing, sensor networks.**

1 INTRODUCTION

Stringent power constraints and limited computation and communication capabilities are key issues in the development of sensor networks. For example, a Berkeley Mote powered by two AA batteries can operate for about one year in the idle state, but only one week when fully loaded. In addition, large-scale sensor network applications impose a demand for cheap, small, low-power sensor nodes, making it impractical to equip sensor nodes with GPS-receivers. While various localization techniques are evolving [1], localization for very large-scale sensor networks is still in the research stage, especially when there is the need to obtain accurate location information under restrained energy conditions. Thus, we focus on location-ignorant sensor networks, which impose challenges on various aspects for query processing in the network, including both query routing and results gathering.

In contrast to sensor networks, mobile wireless networks have much relaxed power constraints. Various communication technologies, including Cellular, Wi-Fi, WiMAX, and Bluetooth, have been developed for connecting billions of electronic products, such as PDAs, cell phones, laptops, and cars. Moreover, GPS-receivers are reasonably affordable on such mobile devices.

We observe that a heterogeneous design space consisting of both static sensor nodes and mobile devices can successfully offset the stringent resource and power constraints in traditional sensor networks. Such an interplay between mobile (ad hoc) networks and sensor networks also links the mobile devices as query requesters and data consumers directly to sensor networks that are responsible for sensing the physical world. We propose a framework for query processing that is based on the "PULL" query model [2] [3] in a two-layer network structure, including a mobile network at the upper layer and a wireless sensor network at the bottom layer. Hereafter, the term "mobile objects" refers to mobile devices operated by users. In our framework, mobile objects take on four roles: query generator, query carrier, query injector and query result collector. By assigning these roles to mobile objects, we expect to simplify the operations of sensor nodes, thus shifting the energy burden from sensors to mobile objects. Another novel aspect of our approach is that mobile objects can take advantage of each other's independent motion plans to perform opportunistic query processing.

---

[1] This material is based upon work supported by the U.S. Army Research Office under grant number W911NF-05-1-0573.

A preliminary version of this paper appears in the Workshop for Distributed Sensor Systems (DSS) 2007. This paper contains substantially new materials including improved system architecture, analysis of methods for gathering query results, and extensive simulation results

The presented framework is designed for delay-tolerant, end-to-end query processing, which consists of four key phases: query generation, query routing, query injection, and query result routing. Specifically, after being *generated* by a querying mobile object M_s, a query is first *routed* in the mobile network layer in an effort to get the query "close" to the target query region. Geographic routing can be used to serve this purpose. When a mobile object that is "close enough" to the query region receives the query, it analyzes the query to determine *if* and *where* the query shall be injected into the sensor network layer, based on the mobile object's current location and velocity, and the query's expiration time. A mobile object *injects* the query into the sensor network when the mobile object arrives at an expected "injection point" location. We denote the injecting mobile object as M_i. Upon receiving the query, sensor nodes perform the required sensing task, and then attempt to *route* the query results back to M_i (or to some other nearby mobile object), typically by disseminating the results to some "intermediate" sensor nodes. A controlled flooding is implemented to limit the degree of flooding of query results in the sensor network layer. When a mobile object, such as M_i, receives the query results, it opportunistically routes the results back to the querying object M_s, again via geographic routing. We also compensate for the mobility of M_s by performing a controlled flooding when the results are routed to the proximity of M_s's original position, i.e. its position at the time of initial query routing.

An implication of our framework is that by exploiting the wide-area mobility of mobile objects, our research has the potential to integrate far-apart sensor networks and provide a framework for future pervasive computing environments.

The remainder of this paper is organized as follows. In Section 2, related work is summarized. In Section 3, we describe the target applications and key properties of mobile objects and sensor nodes. Our system architecture is presented in Section 4. In Sections 5, 6 and 7, we discuss details of the four key query-processing phases, respectively. In Section 8, numerical and simulation results are presented. Finally, Section 9 provides some conclusions and future work.

2 Related work

To conserve sensor node power and simplify routing, base stations in sensor networks are typically assumed to be fixed [4]. The MetroSense project [5] provides a three-tier architecture: a server tier; a sensor access point tier, and a sensor tier consisting of mobile and static sensors. One of the key aspects of MetroSense is to virtually extend the sensing range of a static node by delegating its sensing task to a passing by mobile sensors. In some recent research, mobility of base stations has also been exploited. The TTDD scheme [6] provides a Two-Tier Data Dissemination approach to reduce battery consumption and transmission collision during frequent location updates from multiple sinks to sensor nodes. Another data dissemination protocol, SEAD [7], is proposed to minimize energy consumption in building the dissemination tree and in disseminating data to mobile base stations. In addition, the approach in [8] focuses on the topology control process for mobile base station and application node, which serves as a type of "super" sensor node – it receives raw data from sensor nodes, creates a comprehensive local-view, and forwards the composite bit-stream toward a base station. The Data MULEs approach [9] presents a three-tier architecture and "MULEs" pick up data from sensors in close transmission range, buffer the data, and then drop off the data to wired access points. However, as the designers of this approach noted, the energy consumed during radio monitoring can be very high because each sensor must continuously listen in order to identify a passing-by MULE. The Tenet approach [10] implements a two-tier network: a lower tier consisting of motes and an upper-tier containing relatively less resource-constrained masters.

In general, previous works assume that users and mobile base stations play two distinct roles in sensor networks, with base stations serving as simple information "senders" or "collectors" for users of an existing network. In contrast, our approach unifies the roles of mobile users and mobile base stations. We also extend their role to behave as query carrier and result collector. This requires cooperation among mobile objects, which are connected via ad hoc wireless networks.

Moreover, the research ideas mentioned above, except MetroSense, are based on an event-driven "PUSH" query model. We adopt an alternative scheme, the "PULL" query model. In this scheme, queries are generated by mobile objects, and then disseminated into the sensor network through other mobile objects. The results extracted from the sensor network are then routed back to the querying objects via the mobile network layer.

Although the MetroSense project and our framework share the idea of exploiting mobile nodes that operate in a sensor environment, there are several fundamental differences. MetroSense and our approach assign opposite roles to static and mobile nodes. In MetroSense, mobile sensors interact with a static communication infrastructure, while our approach emphasizes the integration of mobile devices with static sensor networks. Our approach focuses on conserving sensor node power during query processing by limiting the communication burden of sensor nodes; MetroSense focuses on accomplishing a sensor's predefined sensing task when the sensor's sensing range cannot reach a desired sensing region. Finally, our query-oriented research is based on a two-layer architecture with mobile objects serving as source, carrier, injector and collector of queries, whereas MetroSense uses a three-tier architecture that includes a server layer.

Our approach shares a common trait with Tenet [10] – both ideas seek to take advantage of more powerful nodes in an upper tier to offset constraints associated with sensor nodes/motes in the lower tier. But, the main goals of these approaches are different. Our approach is designed specifically to handle mobile-device generated queries that target some remote region by allowing such queries to be opportunistically aided by other mobile objects. The goal of Tenet is to define an architecture to simplify and standardize the development of sensor network applications by forcing the functionality and complex application logic onto "masters," which are the upper-tier nodes that are relatively less constrained (memory, power, etc.) than the lower-tier motes and

are equipped with higher-bandwidth radios. Tenet does not specifically address mobile masters. A key property of the Tenet architecture is the constraining of multi-node fusion to the master tier.

The work in this paper extends the results in [11] [12] by describing enhancements to the system architecture and further study and analysis on query result routing, and also providing comprehensive numerical and simulation results. In [11], the focus was only on the issue of query injection. In [12], the basic two-layer system architecture was proposed and techniques for some of the query processing phases were presented.

3 TARGET APPLICATIONS AND PROPERTIES

3.1 Applications

The type of two-layer networks we consider in this paper can be applied to a wide array of applications when high density mobile objects exist, such as in urban environments.

Thousands of sensor nodes can be densely scattered over some area for monitoring environmental information and aiding mobile users, who might have mobile devices connecting each other through peer-to-peer network. For example, sensors may be densely dispersed along a highway to help monitor traffic and road conditions. Consider an advanced automotive agent system, such as the WILLWARN system [13] sponsored by European automakers, which might serve as an intelligent driving assistant. In the course of its route planning activity, this agent may be interested in road conditions at some remote region (a region outside the transmission range of the vehicle's on-board radio). In this case the (mobile) agent can transmit a query to other vehicle-based agent systems, which in turn route the query for eventual injection to the target region. Moreover, the query results can be routed back to the querying vehicle again via relay by other vehicle agents. Therefore, the agent system can obtain useful information such as icy road conditions or traffic jams in advance, and plan some alternative route. Note that travel plans of individual mobile objects can be independent of the queries that an object happens to be carrying.

3.2 Mobile Objects

We consider mobile objects with relatively powerful computation and communication capability. We assume mobile objects are supported by a rechargeable battery, and equipped with localization devices that also provide a global clock. We also assume that mobile objects have two radios operating in different frequency bands [14]. For example, one radio may operate in the 915 MHz band to communicate with sensor nodes while the other radio operates in the 2.4 GHZ band to communicate with other mobile objects. In addition, a mobile object is assumed to be able to adjust its transmission power when communicating with sensor nodes [15], so as to tune the number of sensor nodes that must be actively engaged during query injection. We denote the maximum transmission range of a mobile object as Tm.[2]

3.3 Sensor Nodes

To facilitate the design of our system architecture and query processing mechanisms, we assume a large number of static sensor nodes to be deployed with high node density, each with limited battery power and computational capability. Also, for our purpose, we assume that sensor nodes are not aware of their location information since localization techniques are still in the research stage, especially those techniques that can reduce energy cost and improve accuracy for very large-scale networks. However, as localization techniques mature, it may be possible to take advantage of new localization methods (e.g., [16][17]) for large-scale, low-cost sensor networks. The presented framework is extensible to cover cases where location information is available to sensor nodes. We denote the fixed transmission range of a sensor node as Ts.

4 SYSTEM OVERVIEW

In this section we present the query details and system architecture of our two layer network system. The major notations are summarized in Table 1.

4.1 Query Details

For the purpose of this paper, we consider delay-tolerant queries, i.e., the querying mobile object requires the results to be delivered in a relatively relaxed time frame, ranging from seconds to minutes. For simplicity, we assume a circular target region, referred to as the *query region* (R_q). For example, a mobile object MO100 at location (10, 100) may issue a query at time 500: "report the temperature associated with the query region centered at location (45, 90), with a radius of 10 meters, before time unit 530."

[2] Although we acknowledge irregular and dynamic communication range for realistic radio models, we adopt a circular model for tractable analysis, which is quite common in the literature, e.g., [18, 19].

Table 1. Key notations

	Variable	Description
Sensor Nodes	Ts	Transmission range of sensor node
Mobile Objects	Tm	Transmission range of mobile objects
	R_d	Dissemination region
	M_s	Querying (source) mobile object
	M_i	Injecting mobile object
Query	q	Query
	R_q	Query region
	λ	Radius of query region
	R_{cq}	Covered query region
	OIP	Optimal Injection point
	D	Distance between query region center and injection point
	$QRCR$	Query region coverage rate
	$DRCR$	Dissemination region coverage rate
	IER	Injection Effectiveness Ratio
Query Result	r	Query result (result packet)
	r_i	Intended query result
	RHC	Return hop counter
	IFA	Intended flooding area
	$P_{MO\text{-}IFA}$	Probability of at least one mobile object within IFA

A *query q* is a 5-tuple: q = (*q_id, object_info, Rq, q_expiration, and q_type*), where

q_id is a query identifier, locally generated by a mobile object;

object_info = (*object_id*, *object_location*, *max_speed, q_time*) is a container for information regarding the mobile object generating the query q;

object_id is a unique identifier of the querying mobile object, *object_location* is the (x, y) coordinate position at the time the querying object initiated routing of the query, *max_speed* is the maximum speed associated with the querying object, and q_*time* is the time stamp of the query (note that *object_location* and *max_speed* are used in Section 7.3 during query result routing);

Rq = (*S, λ)* is the query region centered at S = (x, y), with radius λ;

q_expiration is the expiration time of the query;

q_type is the supported type of a query, such as temperature, route condition, etc.

Note: the combination of *object_id* and *q_id* is globally unique.

In terms of the above example, assuming the object moves at a maximum speed of 20 m/sec, the query can be expressed as the 5-tuple: q = (*q1,* (*MO100,* (*10, 100*), *20, 500*), ((*45, 90*), *10*), *530, temperature*).

We assume that each mobile object can store either locally-generated queries or routed queries in a local database. Considering the advances in low-cost, high capacity memory, we assume a sufficiently large local storage space, thus not concerning ourselves with storage constraints at this time.

4.2 System Architecture

Based on the properties of mobile objects, sensor nodes and queries, we propose a new two-layer network system consisting of a mobile network layer and a sensor network layer. The basic architecture is depicted in Figure 1. Although mobile objects and sensor nodes are physically located in the same two-dimensional coordinate system, we logically classify mobile objects as being in the "upper" layer, and sensor nodes as being in the "lower" layer, because of their different properties.

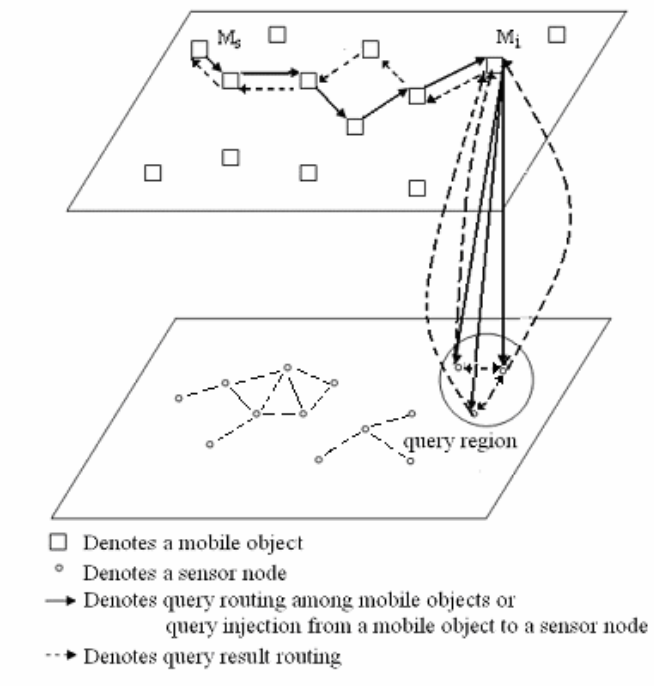

Figure 1. System architecture

Although there may be multiple queries in progress at any time, to simplify our discussion we consider only the processing of a single query since each query is handled using similar but independent processes. Our query processing consists of four sequential phases as illustrated in Figure 2: query generation, query routing, query injection, and query result routing.

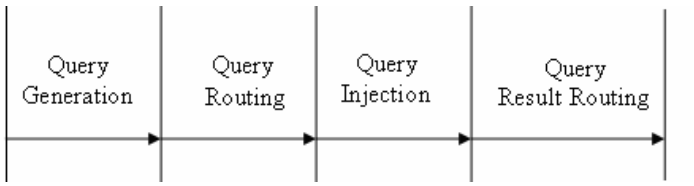

Figure 2. End-to-end query processing

One final comment about our approach is the potentially large conservation of sensor node power due to the fact that a sensor node only reacts when it has received a query from some mobile objects. Thus, sensor nodes can stay in a low-power mode during most of their life, using a radio-triggered wake-up scheme [20].

4.3 Objectives and Obstacles

The ideal objective of our query processing strategy is that any mobile object that generates a query will receive (in response) a set of associated query results, which we can denote as the set QR = $\{r_1, r_2, ... r_n\}$, such that there exists a one-to-one mapping between QR and the set of *n* sensors that reside within the query region associated with the query. However there are practical obstacles that can impede the achievement of this optimal objective and which motivate heuristic techniques that are adopted in the various query processing phases discussed in the following sections.

First, since a source mobile object takes advantage of other mobile objects' independent motion plans to perform opportunistic query routing, there is no guarantee that the mobile object that is selected to inject the query will be able to inject the query such that the query reaches all sensor nodes within the query region. The best we can do is to try and maximize the number of such sensor nodes that are reached (covered) by the query. This issue is the subject of Section 6.1, where the key idea of an "optimal injection point" and an associated query region coverage rate metric are defined. Section 8.2 presents an analysis of this issue. A second obstacle arises due to the fact that we are explicitly not assuming location-aware sensors. Thus, some sensor nodes outside the query region may receive, and therefore respond to, injected queries. While it is not possible to completely eliminate such undesired query results, we can take steps to minimize them. This issue is discussed in Section 6.2, where a second metric (dissemination region coverage rate) is defined; and Sections 8.3 and 8.4 provide associated analysis. As we will see, maximizing "good" query results while at the same time minimizing "bad" query results is an important trade-off. A final obstacle is that it may not be feasible for all query results associated with sensor nodes to be routed back to the source mobile object due to imperfect routing caused by message collisions, limited sensor node transmitter capability, efforts to save sensor node power,

disconnection of sensor nodes or mobile objects from other nodes or mobile objects, etc. Section 7 discusses techniques for query result routing, and various analysis results are given in Sections 8.5 and 8.6.

5 QUERY GENERATION AND ROUTING

A mobile object may generate, or initiate, queries either based on spontaneous user interest, or in a planned fashion. For the former case, for instance, a driver may request traffic information for a particular road segment. For the second case, an intelligent agent-based automotive system, as alluded to earlier, may automatically query road conditions within the next few miles. Once a query is generated by a mobile object, the query is stored locally in preparation for query routing. We refer to the mobile objects that generate queries as "source" mobile objects since they serve as the source of the query.

After a query is generated by a source mobile object M_s, the query is routed to other mobile objects, instead of being distributed to sensor nodes immediately, as in traditional sensor networks. Mobile objects can route the query among themselves via various network connections, including cellular, satellite, and WiFi. We observe that using existing cellular or satellite systems would require extra coordination (e.g., knowing the phone number of a mobile object at the query region) and infrastructure support from carriers (e.g., semantic processing of short messages). Thus, we prefer an infrastructure-free peer-to-peer mechanism, including the ad hoc mode of 802.11 and various Dedicated Short Range Communications (DSRC) compliant technologies for vehicular ad hoc networks (VANETs) [21], to route the queries. Since we consider a region-based query and assume that mobile objects are location aware, standard geographic routing [22], or methods that seek to exploit node mobility [23] to enhance greedy-forwarding, are natural choices. So, a mobile object always selects a neighbor closer to the destination as the next forwarding hop, allowing the query to make progress toward the query region. Existing methods to handle a potential "dead end" can be adopted [24].

A mobile object can store and route queries either locally generated or received from neighbors. For received queries, the mobile object checks the query's validity by comparing the local time to the *q-expiration* in the query record. Expired queries are dropped. Also, to overcome reliability issues for wireless communication and object mobility, a routed query may be re-transmitted if no response is received after a pre-defined timeout.

6 QUERY INJECTION

Since geographic routing is adopted during query routing in the mobile object layer, at any give time only one mobile object is responsible for carrying the query. Future research can investigate techniques for effectively using multiple query carriers and injectors. If a mobile object carrying a query determines that the target region is within the communication range of its radio and that there are no neighbor objects closer to the query region, the mobile object then becomes a potential candidate for injecting the query to the sensor network layer. Since sensor nodes are location-ignorant, the challenge is where and how to inject the query to the sensor network (from a query injection candidate).

6.1 Motion Profile Model and Optimal Injection Point

To avoid blind, energy inefficient flooding by sensor nodes, a mobile object, M_i, should inject a query at an injection point (*IP*) when the query region is (partially) within the object's transmission range. The region that is reached by the query is referred to as the *dissemination region* (R_d) of the query.

Definition: For a query injected at *IP*, the *dissemination region* (R_d) is the region formed by the following set of points:
$R_d = \{P\,(x, y)$, s.t. $|P - IP| \leq Tm\}$.
Recall that *Tm* is the transmission radius for a mobile object.

The dissemination region is simply the region that is covered by an injected query. See Figure 3. The goal of sending (broadcast) a query at some injection point is to have the query directly reach sensor nodes within the query region. Depending on the specific injection point, a query will cover some portion of the query region, denoted as the *covered query region,* R_{cq}, as shown in Figure 3.

Definition: For a query injection, the *covered query region* (R_{cq}) is the region formed by the following set of points:
$R_{cq} = \{P\,(x, y)$, s.t. $|P - S| \leq \lambda$ and $|P - IP| \leq Tm\}$; or equivalently, $R_{cq} = R_d \cap R_q$.
Recall that *S* is the center point of the query region which has radius λ.

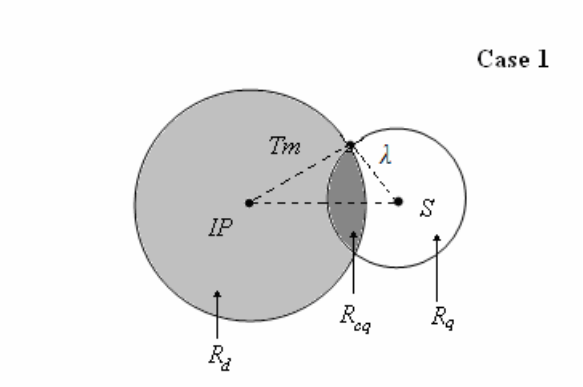

Figure 3. Partial coverage of query region

Based on a mobile object's current location, current velocity, local time and the query's expiration time, the object's short-term motion profile can be generated in real-time. The motion profile is used to predict a mobile object's future trajectory and also pre-select a desired injection point for the carried query. A motion profile for a query is defined by the 3-tuple ($\overline{CE}$, $\overline{C'E'}$, *OIP)*.

$\overline{CE}$ is called the *active-query segment* since it represents points at which the query is still active, i.e., not expired. If M_i changes its velocity, including its motion speed or direction, M_i will recalculate $\overline{CE}$ before the query expires. See Figure 4. Note that other aspects of Figure 4, such as the points C', E', and OIP, will be explained shortly.

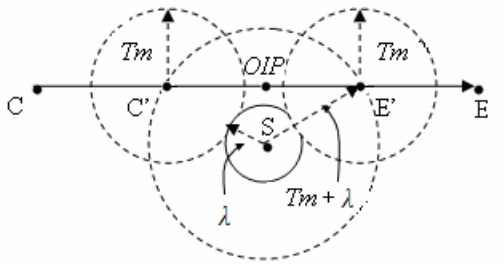

Figure 4. Active-query segment and Injectable-query segment

Definition: For a mobile object with current position C (x, y), carrying a query q and traveling with velocity $\overrightarrow{V(v_x, v_y)}$ at time t, the *active-query segment* $\overline{CE}$ is a segment with end points C (x, y) and E (x, y), where E $(x, y) = \overrightarrow{V(v_x, v_y)} * (q\_expiration - t) + C$ (x, y).

An *injectable-query segment* $\overline{C'E'}$ is a sub-segment of the active-query segment. Figure 4 shows an injectable-query segment $\overline{C'E'}$, which defines the locations during which the distance between the mobile object and the center S (x, y) is less than or equal to $(Tm + \lambda)$.

Definition: For a mobile object that is carrying a query q and has an active-query segment $\overline{CE}$, the *injectable-query segment* $\overline{C'E'}$ is the following set of points:

$\overline{C'E'} = \{P\ (x, y)$, s.t. $P(x, y)$ belongs to $\overline{CE}$ and $|P - S| \leq (Tm + \lambda)\}$.

A mobile object can inject a query to the sensor networks when it reaches an *injection point* (*IP*), which can potentially be any point on $\overline{C'E'}$. This allows the mobile object to choose a desired injection point in order to make a trade-off between coverage of the target query region and energy cost associated with the sensor nodes involved in the query.

Sensor nodes within the entire dissemination region—even those not within the query region—are awakened to perform the sensing task. To conserve energy while ensuring a reasonable query quality, it is crucial to maximize awakened nodes that are within the query region. Intuitively, this requires query injection at a location along M_i's trajectory that is closest to the center of the query region.

A metric called the *query region coverage rate* (*QRCR)* is introduced to measure the effectiveness of query injection.

$$QRCR = \frac{\#\ of\ awakened\ sensors\ in\ query\ region}{\#\ of\ sensors\ in\ query\ region}$$

Assuming a large number of sensors uniformly distributed in the environment, then:

$$QRCR = \frac{area\ of\ covered\ query\ region}{area\ of\ query\ region}$$

To analyze the coverage metric, we can identify four cases and then evaluate the relationship between the coverage metric and the position of the mobile object when it injects a query (at an injection point). Let D denote the distance between the injection point and the center of the query region, i.e., $D = |IP - S|$.

Case 1: The dissemination region (R_d) and query region (R_q) partially overlap, as shown in Figure 3. $R_{cq} \neq \emptyset$ and $R_{cq} \neq R_q$.
Case 2: R_d and R_q do not overlap. $R_{cq} = \emptyset$.
Case 3: R_q is fully contained within R_d, as shown in Figure 5. $R_{cq} = R_q$.
Case 4: R_d is fully contained within R_q, as shown in Figure 6. $R_{cq} = R_d$.

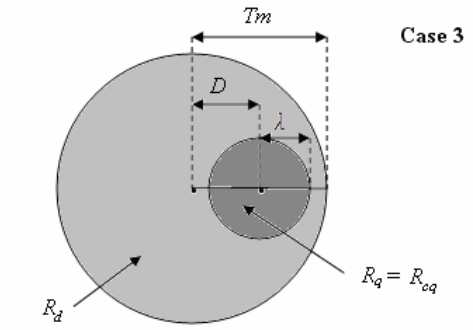

Figure 5. Total coverage of query region

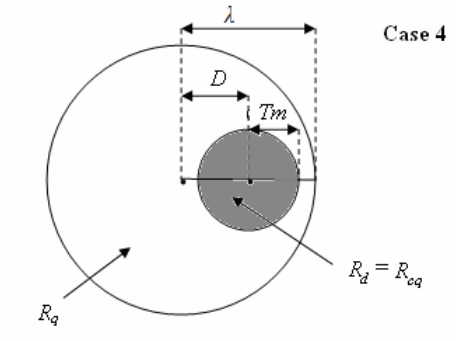

Figure 6. Total coverage of dissemination region

Case 1:

For this case, $Tm - \lambda < D < Tm + \lambda$. The boundaries of the regions R_q and R_d have two intersecting points, labeled as G and H in Figure 7.

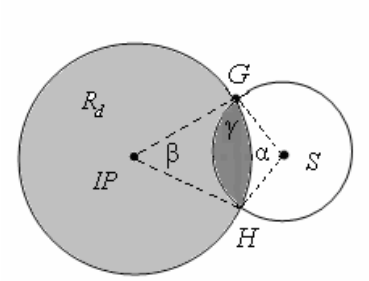

Figure 7. Intersecting boundary points

Assume α, β and γ are the angles of sector (G, S, H), sector (G, IP, H) and angle (IP, G, S), respectively. Given $Tm = |IP - G|$ and $|G - S| = \lambda$, then α, β and γ are all functions of D.

QRCR = (area of sector (G, S, H) – area of triangle (G, S, H) +
Area of sector (G, IP, H) – area of triangle (G, IP, H))/area of R_q

Area of sector (G, S, H) = $\frac{1}{2}\lambda^2 * \alpha$

Area of triangle (G, S, H) = $\frac{1}{2}\lambda^2 \sin(\alpha)$

Area of sector (G, IP, H) = $\frac{1}{2}Tm^2 * \beta$

Area of triangle (G, IP, H) = $\frac{1}{2}Tm^2 * \sin(\beta)$

Area of $R_q = \pi * \lambda^2$

$$QRCR = \frac{\left[\frac{1}{2}\lambda^2 * (\alpha - \sin(\alpha)) + \frac{1}{2}Tm^2 * (\beta - \sin(\beta))\right]}{\pi * \lambda^2} \quad (1)$$

Lemma 1: For Case 1, *QRCR* is a decreasing function with respect to the distance measure *D*. Proof is in the appendix.

For Case 1, the mobile object can inject a query at any point on the injectable-query segment, but by Lemma 1, *QRCR* is maximized when *D* is minimized. This matches intuition – we achieve greater query region coverage when the injection point is close to the center of the query region. So, the optimal injection point *OIP* in the motion profile is the injection point that is closest to the center of the query region. Figure 4 shows an example situation.

Definition: For a mobile object that is carrying a query *q* and has an injectable-query segment $\overline{C'E'}$, the optimal injection point *OIP* can be expressed in two similar ways:

OIP = {*IP* (*x, y*), s.t. *IP* ∈ $\overline{C'E'}$ and Min (|*IP* – *S*|) }; or, *OIP* = {*IP* (*x, y*), s.t. *IP* ∈ $\overline{C'E'}$ and Max (*QRCR*)}

Case 2:

For Case 2, $D > Tm + \lambda$. Thus, *QRCR* = 0, independent of *D*. Since the query region coverage rate is always zero, there is no optimal injection point. *OIP* = NULL.

Case 3:

For Case 3, $Tm \geq \lambda$ and $D \leq Tm - \lambda$. Thus, $R_q = R_{cq}$ and *QRCR* = 1, independent of *D*. In this case, there is complete coverage of the query region. So, any injection point is an optimal injection point; there is a set of optimal injection points.

Case 4:

For Case 4, $Tm < \lambda$ and $D < \lambda - Tm$. Thus, $R_d = R_{cq}$ and $QRCR = (\frac{\pi * Tm^2}{\pi * \lambda^2}) = (\frac{Tm}{\lambda})^2$, independent of *D*. Since *QRCR* is not dependent on *D*, any injection point is an optimal injection point; there is a set of optimal injection points.

Although a mobile object's trajectory can be a curve instead of a straight line, because of changing velocity, Lemma 1 still holds. So, the optimal injection point of the mobile object is the point closest to the center of the query region, as shown in Figure 8.

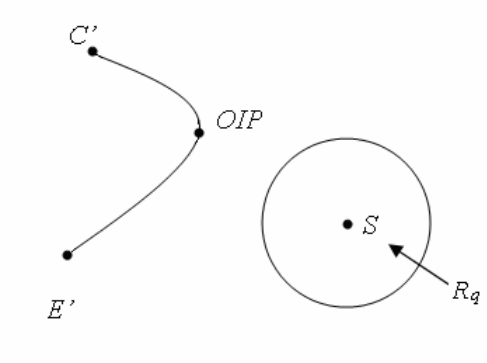

Figure 8. Non-linear trajectory for mobile objects

6.2 Transmission Range Adjustment

From the above analysis, we determined that a mobile object should inject a query when the object reaches the optimal injection point, thus achieving maximum *query region coverage rate.* We refer to those sensor nodes in the query region as "target sensor nodes".

But, maximizing *QRCR* may still result in query reception by some number (possibly many) sensor nodes that are *not* within the query region. For the situation shown in the left-hand side of Figure 9, which represents a case when the coverage rate is optimal (equal to 1), we see many such sensor nodes that are being "reached" inadvertently. We refer to these sensor nodes as "unintended sensor nodes." Those sensor nodes outside the dissemination region are called "uncovered sensor nodes" (they are not covered/reached by an injected query).

Definition: A sensor node located at *P* (*x*, *y*), is a *target sensor node* iff $|P - S| < \lambda$.

Definition: A sensor node located at $P(x, y)$, is an *unintended sensor node* iff $|P - S| > \lambda$ and $|P - IP| < Tm$.

Definition: A sensor node located at $P(x, y)$, is an *uncovered sensor node* iff $|P - IP| > Tm$.

Note: Some sensor nodes can be both target sensor nodes and uncovered sensor nodes.

To rectify this situation, M_i can adjust its transmission range to tune the fraction of awakened "target sensor nodes" and reduce awakened "unintended sensor nodes." This is illustrated in the right-hand side of Figure 9, where the object's transmission range has been reduced, in comparison to the situation in the left-hand side.

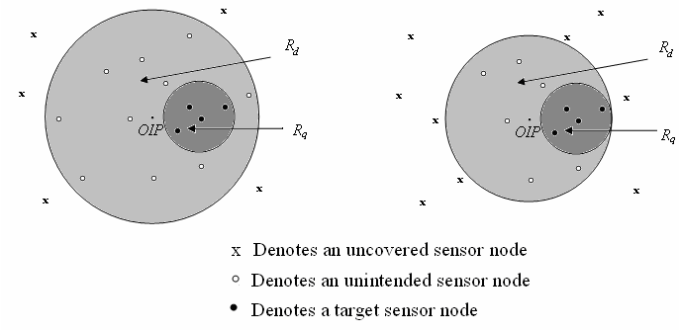

Figure 9. Unintended sensor nodes and reduced transmission range

Figure 10 shows a case where the transmission range has been reduced even further, to the point that there are only two undesired target sensor nodes. But this gain is at the sacrifice of also possibly reducing the number of sensors in the covered query region. Clearly, there is a tradeoff between these two objectives.

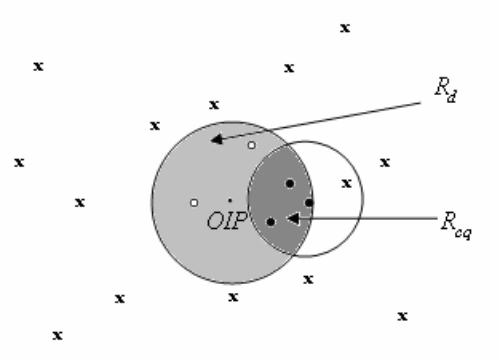

Figure 10. Further reduced transmission range

Thus, we now introduce a new coverage metric, *dissemination region coverage rate* (*DRCR)*, to measure the effectiveness of covering the query region with a reduced object transmission range.

$$DRCR = \frac{\#\ of\ awakened\ sensors\ \ in\ query\ region}{\#\ of\ sensors\ \ in\ \ dissemination\ region}$$

As before, if we assume a large number of uniformly distributed sensors, then

$$DRCR = \frac{area\ of\ covered\ query\ region}{area\ of\ dissemination\ region}$$

Now, *DRCR* can be analyzed in three cases, where we let *Tm* denote the object's (reduced) transmission range at the time of query injection.

Case A: The dissemination region and query region partially overlap, but the optimal injection point is not within the query region. In this case, $R_{cq} \neq \emptyset$ and $|OIP\text{-}S| > \lambda$; thus (similar to the previous analysis for *QRCR*)

$$DRCR = \frac{\left[\frac{1}{2}\lambda^2 * (\alpha - \sin(\alpha)) + \frac{1}{2}Tm^2 * (\beta - \sin(\beta))\right]}{\pi * Tm^2} \quad (2)$$

Although the maximum dissemination region coverage rate with respect to *Tm*, or the optimized transmission range, can be characterized by solving the ordinary differential equation $\frac{d(DRCR)}{d(Tm)} = 0$, we simplify the analysis by using numerical and simulation results, as presented in Section 8. The simulation reveals that *DRCR* increases as *Tm* decreases, but up to a limit.

Case B: The dissemination region and query region do not overlap. In this case, $R_{cq} = \emptyset$; thus *DRCR* = 0, independent of *Tm*.

Case C: The dissemination region and query region overlap, and the optimal injection point is within the query region, as shown in Figure 11. $R_q \cap R_d \neq \emptyset$ and $|OIP - S| < \lambda$. To maximize *DRCR*, the mobile object can reduce its transmission range so that R_d is within R_q, i.e., $Tm \leq \lambda - D$. In this case, $R_{cq} = R_d$ and $DRCR = (\pi Tm^2)/(\pi Tm^2) = 1$. One practical problem that becomes apparent from simulation analysis of this case (See Section 8.3) is that if a mobile object is located too close to the edge of the query region (but still within the query region), the maximization of *DRCR* can define such a small (optimized) transmission range that very few, if any, sensors will be reached. While there can be many ways to prevent this special situation, we adopt the simple strategy of setting a lower-bound threshold on the transmission range of mobile objects. In the simulation studies for query result routing (Section 8.5), we use 50 meters as the threshold since that is the assumed value for the transmission range of sensor nodes.

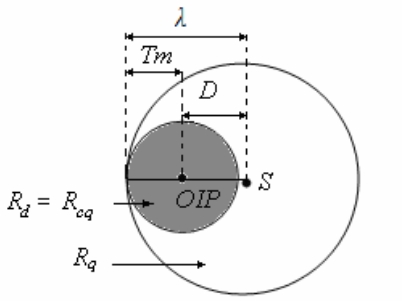

Figure 11. Covered query region, given $D < \lambda$

Before injecting a query, if M_i changes its velocity, including its motion speed or direction, M_i will recalculate its optimal injection point in the motion profile and optimal injecting range.

7 QUERY RESULT ROUTING

Upon receiving a query, sensor nodes perform the required sensing task. Then, a key operation is to efficiently route the query results *r* back to a mobile object.

A query result *r* associated with some sensor is a 5-tuple:

r = (*q_data*, *sensor_id*, *RHC*, $RHC_{initial}$, *q*), where

q_data is the desired sensor data;

sensor_id is the unique id for sensor node;

RHC (*Return Hop Counter*) is a variable that changes as this query result moves through the sensor network. Its precise purpose is discussed shortly;

$RHC_{initial}$ is the initial value of *RHC* that is assigned by the injecting mobile object M_i. $RHC_{initial}$ is used to control query result routing in the mobile layer. Its precise purpose is discussed shortly;

q is the original query generated by the querying mobile object:

q = (*q_id, object_info, Rq, q_expiration*, *q_type*).

7.1 Routing in the Sensor Network Layer

Since the transmission range of sensor nodes is typically much smaller than that of mobile nodes, sensor nodes must typically relay their query results to some intermediate sensor nodes before the query result can be received by a nearby mobile object. Query result routing in the sensor network layer is challenging due to the nature of one-shot queries and the limitations of sensor nodes. Traditional data-centric routing methods such as directed diffusion [25] are not well suited for one-shot queries since these

methods require significant overhead in flooding an interest throughout the network to build gradients that can support longer-term data aggregation. Likewise, routing-tree based protocols are also of limited value because route discovery has high energy cost that is not justified by satisfying a one-shot query. Finally, the location ignorance of sensor nodes prevents us from adopting routing protocols that require location knowledge. So, we use flooding as a base routing technique, despite the fact that flooding is not highly efficient and can cause network congestion and large energy consumption.

To prevent network-wide flooding we use a form of TTL (time-to-live) to restrict the number of transmissions by hops. A *Return Hop Counter (RHC)* is used to implement the controlled flooding. The value $RHC_{initial}$ is pre-calculated by a mobile object during the query injection phase and then attached to the injected query. Therefore, when the injected query reaches a sensor node, this $RHC_{initial}$ value can be inserted into the query result. $RHC_{initial}$ is used as an upper bound on the number of hops that a query result is re-transmitted by sensor nodes. Every re-transmission of the query result decrements *RHC* until *RHC* reaches zero, at which point the query result *r* is dropped. See Figure 12.

There are two ways to think about the outcome of query-result routing within the sensor network: The query result eventually reaches the mobile object that injected the query or the query result reaches some other mobile object that just happens to be within range of some sensor node that is relaying the query result. Both situations are discussed below, with simulation results given in Section 8.

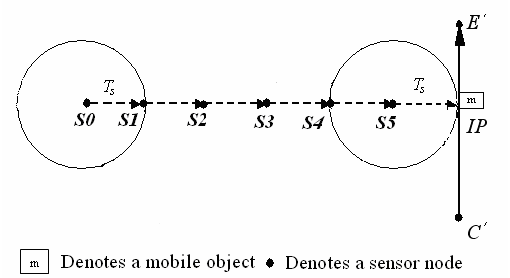

Figure 12. Example of query result routing with $RHC_{initial} = 6$

7.1.1 Restricted Flooding with M_i as Target

To increase the chance that M_i can receive each query result packet in a dense sensor network, $RHC_{initial}$ can be set as RHC_{MAX}:

$$RHC_{MAX} = \frac{Mobile\ Object's\ Transmission\ Range}{Sensor\ Node's\ Transmission\ Range}$$

Note: Theoretically, in a sufficiently dense sensor network, a query result that is generated by a sensor node located at the maximum distance from M_i (but within transmission range of M_i) can reach the injection point using RHC_{MAX} hops. But, this hop count may not be large enough in practice. First, routing paths often consist of zigzag paths rather than straight-ling paths. Second, since the source mobile object is moving during query result routing, the injecting object will be located relatively near, but not at, the injection point. Thus, network dilution[26][3] θ should be applied to RHC_{MAX} to compensate for such cases. Network dilation is defined as the upper bound on the ratio of the hop count between any two nodes to the minimum hop count between them. Thus,

$RHC_{adjusted} = \theta * RHC_{\mathrm{MAX}}$

In Section 8.5, we choose θ as 2 during the simulation of the query result routing from sensor nodes to injecting mobile objects.

7.1.2 Restricted Flooding with Arbitrary Targets

Although M_i can theoretically receive its query results via $RHC_{initial}$ relaying hops, if the mobile object density is sufficiently high, it is likely that other mobile objects may be able to receive a query result through fewer relaying hops than $RHC_{initial}$. This would lead to greater energy savings for the sensor nodes. However, when the density and location of mobile objects in vicinity are unavailable, a reduction in the number of relaying hops may also decrease the probability of having at least one mobile object receive the query result. Thus, a tradeoff needs to be explored between the deliverability of query results and the energy consumed by sensor nodes.

Although all sensor nodes within a dissemination region can sense the environment and return query results, only query results generated by sensor nodes within the covered query region (i.e, target sensor nodes) are the intended results for the query. Recall the definitions of target sensor nodes and unintended sensor nodes from Section 6.2. We denote intended query results as r_i hereafter.

Let us refer to the area that can be covered by the controlled flooding of an r_i (using *RHC* to control the flooding) as the *Intended Flooding Area* (*IFA*). To explain the significance of the IFA, consider the simple case when $RHC_{initial}$ is set as 1. In this

[3] [26] focuses on network dilution in sensing-covered network, which means that every point in a geographic area must be within the sensing range of at least one sensor. Study shows that network dilution can be as low as 2.5 in some circumstance.

case, r_i is broadcast exactly once by queried sensor nodes. Other sensor nodes receiving an r_i message will not rebroadcast the r_i since *RHC* will have dropped to zero. Figure 13 illustrates the *IFA* in this special case, where the *IFA* is the combined area of the darker-shaded R_{cq} area and the lighter-shaded band for sensor nodes that can receive the r_i but did not generate the r_i.

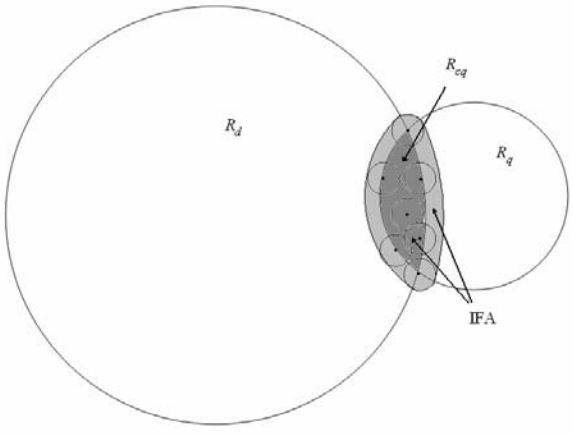

Figure 13. An illustration of Intended Flooding Area (*IFA*)

To simplify our analysis, we approximate this *IFA* by the intersection of an "extended" dissemination region (R_d) and an "extended" query region (R_q), as shown in Figure 14. Let T_S denote the transmission range of sensor nodes. The radius of the extended R_d is $Tm+T_S$ and the radius of the extended R_q is $\lambda+T_S$. Therefore, the intended flooding area is approximated by the covered query region associated with these extended regions. We can denote this as $IFA \approx Area\ of\ R_{cq}(Tm+T_S, \lambda+T_S)$.

When $RHC_{initial} \geq 1$,

$$IFA \approx Area\ of\ R_{cq}(Tm + RHC_{initial} * T_S, \lambda + RHC_{initial} * T_S) \quad (3)$$

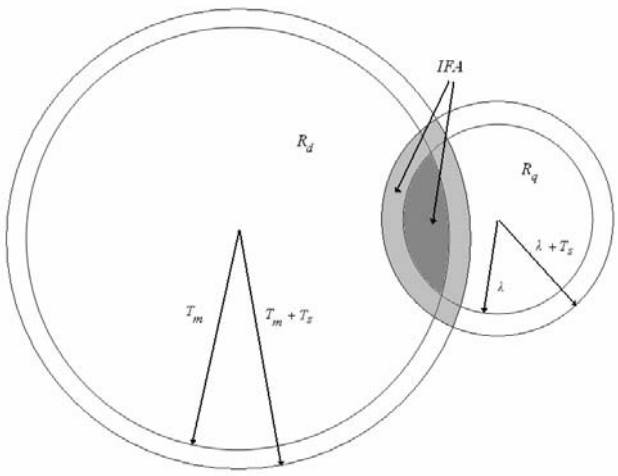

Figure 14. *IFA* approximation

Given an *IFA*, we are interested in the probability of there being at least one mobile object within this area. We denote this probability as $P_{MO\text{-}IFA}$. So $P_{MO\text{-}IFA}$ denotes the deliverability of r_i to at least one mobile object. Let Ω denote the area of the entire operational space of the mobile objects and φ denote the total number of mobile objects in that space. We assume a uniform distribution of mobile objects in the space. Intuitively, $RHC_{initial}$ can be tuned to control the range of *IFA* and the consequent deliverability of r_i. A greater $RHC_{initial}$ results in a larger *IFA* and thus a higher $P_{MO\text{-}IFA}$. When $RHC_{initial}$ is set to RHC_{MAX}, $P_{MO\text{-}IFA}$ will be 1 since each r_i will be received by the injecting mobile object, M_i. Therefore, $P_{MO\text{-}IFA}$ can be theoretically calculated as:

$$\begin{aligned} P_{MO-IFA} &= 1 - prob(no\ mobile\ oject\ is\ within\ IFA) \\ &= 1 - (1 - \frac{IFA}{\Omega})^{\varphi}, \qquad if\ RHC_{initial} < RHC_{MAX} \\ p_{MO-IFA} &= 1, \qquad\qquad\qquad if\ RHC_{initial} = RHC_{MAX} \end{aligned} \quad (4)$$

We now model the expected total energy consumption among sensor nodes due to routing query result r with $RHC_{initial}$. To simplify our model we ignore energy costs that might arise due to collision among radio signals in a geographical area[4] and we assume fixed packet lengths and identical sending (receiving) energy costs for all sensor nodes.

Let E_T denote the energy cost for a node to transmit a query result r, and E_R denote the energy cost to receive a query result r. The total energy consumption (which is a function of the hop count used) is denoted as $EC(RHC_{initial})$ and can be calculated as:

$EC(RHC_{initial})$ = (total # of sensor nodes transmitting any $r * E_T$) + (total # of sensor nodes receiving any $r * E_R$) (5)

Assume sensor nodes are randomly distributed in the environment and the density of sensor nodes is Δ_S. The expected total energy consumption, denoted as EXP($TE(RHC_{initial})$), can be calculated as:

EXP($EC(RHC_{initial})$)
= # of r * (energy cost for routing a r)
= # of r * ((# of sensors transmitting a given $r * E_T$) + (# of sensor nodes receiving a given $r * E_R$)) (6)

where
of r = area of dissemination region * density of sensor nodes
$= \pi * Tm^2 * \Delta s$ (7)

# of sensors transmitting a given r = controlled flooding area of a given r * density of sensor nodes
$= \pi * (T_s * RHC_{initial})^2 * \Delta s$ (8)

# of sensors receiving a given r = # of sensors transmitting a given r * (# of sensors receiving a given r when the given r is transmitted one time)
= # of sensors transmitting a given r * (covered area when r is transmitted one time * density of sensor nodes)
$= [\pi * (T_s * RHC_{initial})^2 * \Delta s] * (\pi * T_s^2 * \Delta s)$ (9)

Based on the formulas (6), (7), (8) and (9),

$$EXP(\ EC(RHC_{initial})\ )$$
$$= [\pi * Tm^2 * \Delta s] * \{[\pi * (T_s * RHC_{initial})^2 * \Delta s * E_T] +$$
$$[\pi * (T_s * RHC_{initial})^2 * \Delta s * (\pi * T_s^2 * \Delta s)] * E_R\} \quad (10)$$

We will present simulation results of energy consumption in Section 8. One finding is that by setting the return hop count to be less than RHC_{MAX} to conserve sensor node energy, query results can still be collected by some mobile object with high probability.

Since a sensor node may receive the same query result multiple times, a sensor node can always cache recently routed packets and check *q-id, object_id* and *sensor-id* to identify and drop duplicate packets.

7.2 Routing From Sensor Node to Mobile Object

As query results are broadcast by sensor nodes to their neighbors, some mobile object is expected to eventually receive the query result r. Any mobile object receiving a r from a sensor node can check whether the r was possibly generated by a target sensor node; this is a query result worthy of further routing. Consider the following two cases.

Case 1: If a mobile object's distance to the query region is greater than the cumulative distance that a received query result was transmitted, it must be true that the query result was generated by some sensor node outside the desired query region. Thus, the query result is not an intended query result and the mobile object should not route the query result in the mobile layer.

Case 2: If a mobile object's distance to the query region is less than or equal to the cumulative distance that a received query result was transmitted, it is possible for a query result to have been generated by some target sensor node (i.e., a sensor node inside the query region). Therefore, the mobile object should initiate the routing of this query result in the mobile layer.

It is important to note that information contained in the query result facilitates the ability for a mobile object to distinguish the above two cases. Specifically, when receiving a query result, a mobile object M_j calculates the distance between itself and the query region, denoted as $|M_j\ (x, y)\text{-}S\ (x, y)| - \lambda$. Recall that S is the center of the query region and λ is the radius of the query region. The cumulative distance that a received query result was transmitted can be calculated as $|RHC_{initial}\text{-}RHC| * Ts$

[4] When the sensor distribution is dense, a straightforward broadcasting can cause serious redundancy and costly collision, known as the broadcast storm problem. Reference [27][28] proposed several schemes to eliminate this problem.

Then, M_j adopts the following strategy to handle the cases listed above:

Case 1: M_j drops the query result, if
$|M_j(x, y)\text{-}S(x, y)| - \lambda > |RHC_{initial}\text{-}RHC| * Ts$

Case 2: M_j distributes the query result, if
$|M_j(x, y)\text{-}S(x, y)| - \lambda \leq |RHC_{initial}\text{-}RHC| * Ts$

7.3 Routing in the Mobile Layer

After a sensor node transmitted a query result *r* to a mobile object, the mobile object adopts a geographic routing scheme to route the query result back to the querying mobile object M_s. Since multiple mobile objects may receive the same query result, mobile objects can cache recently routed query result*s* so that duplicate query results are identified and dropped.

A key challenge is that while query results are routed back toward *object_location*, which is a parameter of the original query, the querying mobile object M_s may have since moved to a new location. To cope with this problem, we propose a controlled flooding of query results once they can be distributed to the location *object_location* but they have failed to reach M_s. The flooding is performed by the last mobile object on the geographic routing path. The flooding area is within a circular region centered at *object_location*, with radius *RF*, where

$$RF = |local\_time - q\_time| * \max\_speed \quad (11)$$

The rational is that the query result is flooded within a region determined by the predicted distance that can be traveled by M_s. Figure 15 illustrates a situation using the controlled flooding with M_s as target. Mobile object M_1 routes a query result *r* to M_2 using geographic routing, but M_2 has no neighboring mobile object closer to *object-location*, which is where mobile object M_s was located when it generated the initial query *q*. Assuming that M_2 cannot directly send the query result to M_s (since M_s has moved since the time that it generated the query) the query result will then be flooded from M_2 to other mobile objects M_3, M_4 and M_5, since those objects are within the flooding region specified in formula (11). Eventually, M_5 routes the query result to M_s since M_s is also within the flooding region.

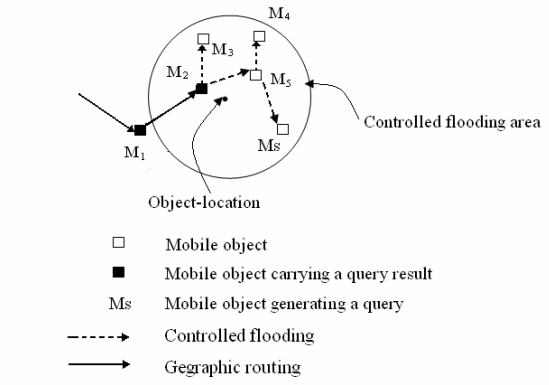

Figure 15. Controlled flooding in mobile layer

7.4 Query-Result Routing Failure

We presented various techniques to address the obstacles identified in Section 4.3. In particular, the *QRCR* metric is used to locate the optimal injection point and thus maximize the number of query results generated by target sensor nodes. But it is still possible for a query to "fail" in the sense that some query results fail to reach the querying mobile object. There can be several reasons for this. First, some query results may not be able to be routed to the injecting mobile object, or other nearby mobile objects, due to inherent deployment limitations of sensor nodes (such as low density in some regions). Second, query results may get dropped due to packet collisions during restricted query-result flooding. Third, the dynamic topology of mobile objects may prevent the construction of a geographic-routing path between a mobile object carrying some query results and the querying mobile object. Finally some query results may simply expire before reaching the querying mobile object. Some of these cases are observed in the simulation studies of the next section. Future research can investigate ways to mitigate the impact of these issues.

8 NUMERICAL AND SIMULATION RESULTS

In this section, we present both numerical and simulation results of the proposed techniques. We focus on several critical stages to validate the feasibility of these critical stages and to examine key performance issues. Implementation and simulation of the complete end-to-end query processing approach remains for future work.

8.1 Simulation Environment and Basic Parameters

Numerical results were generated using MATLAB 7.0. Simulation results were collected using custom simulators for the various stages. The simulation development was done in Visual Studio.NET. Please refer to Table 1 in Section 4 for a reminder of the variables used in the simulations. The simulation area was of size 5000 units by 5000 units. One unit represents one meter. Sensor nodes and mobile objects were randomly placed within the simulation area. The default transmission range of mobile objects (Tm) was 250 meters. The default transmission range of sensor nodes (T_S) was 50 meters. Each simulation result was based on aggregated results from 100 independent simulations with the same parameter settings.

8.2 QRCR Analysis

Query injection is one of the critical stages in the proposed two-layer networking approach. Since the concepts of optimal injection point and transmission range adjustment are key factors during query injection, we analyzed and simulated *QRCR* with respect to these two factors.

Figure 16 shows numerical and simulation results for *QRCR* with varying λ, radius of query region, and fixed D_{min}. A total of 500 sensor nodes are deployed in the environment. The results were consistent with Lemma 1 – *QRCR* was a decreasing function under Case 1 and it was maximized when the mobile object injected a query at the optimal injection point. We can also observe the special cases discussed in Section 6.1. When λ=100 and $D>Tm+\lambda$, *QRCR*=0 for Case 2. When λ=100 and $D<Tm-\lambda$, *QRCR*=1 for Case 3. When λ=400 and $D<\lambda-Tm$, *QRCR* is close to $0.39\ (\frac{Tm}{\lambda})^2$.

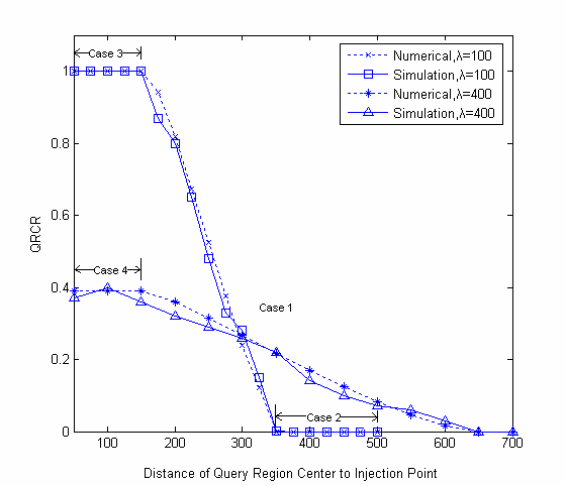

Figure 16. *QRCR*, given *Tm*=250, *Ts*=50, D_{min}=50.

8.3 DRCR Analysis

When a mobile object reaches its optimal injection point, the mobile object adjusts its transmission range to reduce awakened "unintended sensor nodes." We analyzed and simulated *DRCR* with respect to the situations identified and discussed in Section 6.2.

Figure 17 gives simulation results for *DRCR* corresponding to Case A in Section 6.2, given λ=100, D_{min}=150, and *Tm* varies from 250 to 50. Again 500 sensor nodes are deployed in the environment. Based on the numerical results, the optimized transmission range (*Tm*) for a mobile object was 170 units, achieving a dissemination region coverage rate of 19.4%. Based on the simulation results, the optimized transmission range was 160 units, which achieved a dissemination region coverage rate of 21.3%.

Although the numerical and simulation results showed slightly different optimized transmission range values, we could observe a common characteristic on how the coverage rate changed. The coverage rate improved initially, as the transmission range of mobile objects decreased from its initial range value of 250. This was due to a reduction in the number of unintended sensor nodes. But then the coverage rate decreased after this transmission range went below some value. This was due to the fact that the rate at which nodes changed from target nodes to uncovered nodes was greater than the rate at which nodes changed from unintended nodes to uncovered nodes. The peak value for *Tm* was the optimized transmission range to be used for query injection.

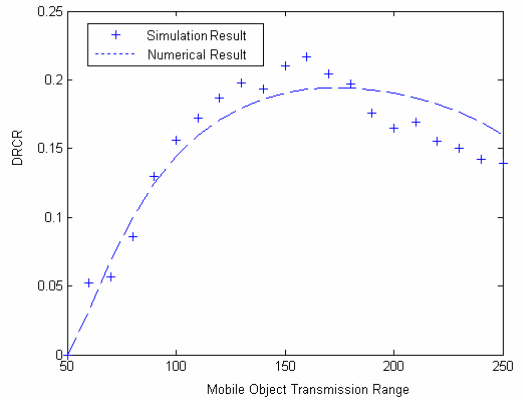

Figure 17. *DRCR*, given *Tm*=250, λ=100, D_{min}=150.

Figure 18 shows simulation results for *DRCR* with varying number of sensor nodes, corresponding to Case C in Section 6.2. As in Figure 17, λ=100 and *Tm* varies from 250 to 0. But, now D_{min}=50, causing the dissemination region and query region to overlap, and the optimal injection point is within the query region (recall Figure 11). For this case, the mobile object can optimize DRCR by reducing its transmission range so that R_d is within R_q, i.e., $Tm \leq \lambda - D$. Thus, $R_{cq} = R_d$ and *DRCR* is expected to be $\pi * Tm^2 / \pi * Tm^2 = 1$. However, this simulation demonstrated some interesting features. First, we observed that the plotted DRCR values in Figure 18, which correspond to averaging of individual simulation runs, will range between 0 and 1. The theoretical peak value of 1 for *DRCR* was not achieved in a small number of simulation runs. This occurred in those situations where there were no sensors that could be reached by the injected query, resulting in a *DRCR* value of 0 for those simulations. Second, we found that *DRCR* decreased dramatically when a mobile object further reduced its transmission range after reaching a peak value for DRCR. Once the transmission range is reduced below this critical point, every simulation run will either produce a *DRCR* value of 1 (if any sensors are reached by the injected query) or a *DRCR* value of 0 (if no sensors are reached). As expected, when the transmission range decreased, the number of cases with *DRCR*=0 increased, pulling the average *DRCR* down. Finally, we can observe that for systems with a higher density of sensor nodes, there is a lower probability of such cases with *DRCR*=0 when the transmission range is below the critical point.

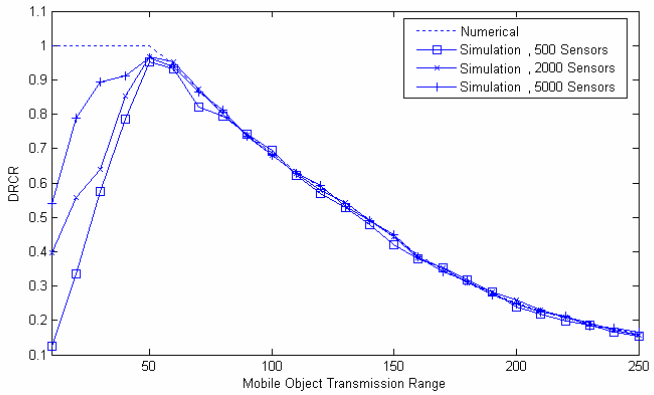

Figure 18. *DRCR*, given *Tm*=250, λ=100, D_{min}=50.

8.4 Injection Effectiveness Analysis

Even when a mobile object injects a query using an optimized transmission range, the query might still reach unintended sensor nodes. Too many such unintended sensor nodes during query injection not only wastes sensor node energy, but also can deteriorate the quality of query results. While the DRCR metric indicates the effectiveness of a particular query injection, it does not reflect the system-level effectiveness due to multiple injections of multiple queries from multiple source mobile objects. Thus, we define another metric, called the Injection Effectiveness Ratio (*IER*), as follows:

$IER = \overline{DRCR}$, i.e., the average value of DRCR computed over a set of injections.

To evaluate the relationship between this system-level metric and system properties such as number of mobile objects and query region size, we performed a set of simulation experiments. We used random-waypoint [29] as the mobility model. A mobile object randomly selects a destination and moves in the direction of the destination with a speed uniformly chosen between 0 and 40 units (meters) per second. After reaching its destination, the node selects another random destination.

Figure 19 shows the aggregate results of the injection effectiveness ratio (*IER*) with varying number of mobile objects, two different query region sizes, and 500 sensor nodes. The query region location was fixed. Each simulation randomly selected a mobile object to serve as the query generator and then located the injecting mobile object based on geographical routing in the mobile object layer. For each injection, the DRCR value was computed. Each plotted point in the graph represents the IER value for 100 simulation runs. We observed that the ratio increased when the density of mobile objects became higher or when the query region size increased. Intuitively, the probability of some mobile objects moving to a position that is within a query region is higher when more mobile objects exist in the environment or when the query region itself is large. Furthermore, when an injecting mobile object is within the query region, there are no unintended sensor nodes involved in the query injection process (except for the special case when the transmission range for query injection falls below some enforced threshold, as was mentioned in Section 6.2, Case C.)

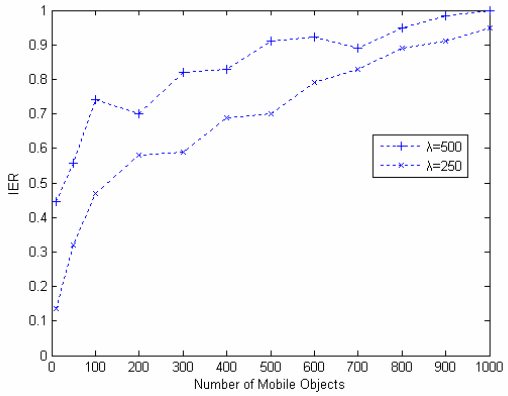

Figure 19. *IER*, given *Tm*=250

8.5 Query Result Routing in the Sensor Network Layer

After a mobile object injects a query into the sensor network layer, sensors within the dissemination region will respond by sending their sensor data. As we discussed earlier (see Section 7.1), query-result data is routed among sensor nodes using a restricted flooding approach based on a return hop counter (RHC). This subsection studies the performance of query result routing for a single query injected under a variety of conditions. Since the simulation experiments only considered a single query at a time, the monitored environment size Ω was reduced to a 1000m by 1000m square, with a fixed query region located at the center (i.e., query region was centered at (500, 500)), with a radius λ=200). The simulation analysis considered four cases, corresponding to four different injection points, with respect to the query region. In each case, the transmission range of the injecting mobile object was computed to optimize *DRCR*, as discussed in Section 6.2, and we established a 95% success rate for the injecting mobile object to reach sensors within its transmission range.

Case 1: The injection point was the center of the query region. This resulted in an optimized transmission range of 200, the same as the radius of the query region. This represented a best case scenario in terms of injection effectiveness.
Case 2: The injection point was within the query region and located at position (400,500). This resulted in an optimized transmission range of 100.
Case 3: The injection point was at the edge of the query region and located at position (300,500). This resulted in an optimized transmission range of 50, due to our adoption of 50m as the lower-bound threshold.
Case 4: The injection point was slightly outside of the query region and located at position (270,500). This resulted in an optimized transmission range of 190.

To evaluate the effectiveness of query result routing we considered those query results that were successfully routed back to the injecting mobile object. Once a sensor generated a query result, it chose a random number between 0 and 10 time units as the waiting time before the sensor-initiated RHC-based flooding of the query result. This aimed to reduce packet collisions for the initial broadcasts. In this case, 1 time unit represented 50 milliseconds, the approximate time to send a packet on the 10 Kbit radio of Motes [30]. We assumed that sensors had a 100% success rate for communicating with a mobile object located within 50 meters, while the sensor-to-sensor communication success rate depended on a selected communication model – two models are described below. After injection, the injecting mobile object moved based on a speed randomly chosen between 0 and 40 m/sec,

and a direction randomly chosen from one of four directions on a grid: east, west, south and north. While this motion could change every few seconds, this was not relevant since no query results required more than one second to reach the mobile object. Even with multi-hop routing, the delay was small since we used a "best-effort" routing approach (no retransmission).

As is standard in the field of information retrieval, we considered both the precision and recall associated with the query results gathered by the mobile object. In this case, precision and recall are similar to *DRCR* and *QRCR*, respectively, as used to study query injection, but with focus on query results that are received rather than on queries that are disseminated.

$$\text{Query result precision (QRP)} = \frac{\#\text{ of query results received from sensors in the query region}}{\#\text{ of query results received from sensors in the dissemination region}}$$

$$\text{Query result recall (QRR)} = \frac{\#\text{ of query results received from sensors in the query region}}{\#\text{ of sensors in the query region}}$$

For each injection-point case, these metrics were impacted by three main factors: 1) density of the sensor nodes; 2) MAC-layer issues (collision handling); and 3) communication characteristics of sensor nodes. For each case, two sets of simulations were conducted using different communication parameters, and in each set of simulations we varied the number of sensor nodes (300, 500, 700, 900 and 1100). The first set of simulations assumed a perfect MAC layer with no collisions and a "binary" sensor-to-sensor communication model, meaning that there was a near-perfect (95%) success rate for communication between nodes locate within 50 meters of each other, but a 0% success rate otherwise. These simulations established an upper-bound result. The second set of simulations assumed a more realistic situation – first, packet collisions were detected at the receiving side, and when there is a collision, those packets were discarded (i.e., we do not employ any retransmission); second, a "decay" sensor-to-sensor communication model was used, meaning that the success rate for communication was a function of the distance between the sending/receiving sensors. We used the link quality model [31], but with the "effective" transmission range scaled to 50 meters, which is appropriate for state-of-the-art hardware platforms such as the TelosB node [32].

8.5.1 Case 1: Injection point is the center of the query region

Figure 20(a) shows the results for a perfect MAC layer with no collisions and a "binary" sensor-to-sensor communication model. The query result precision was 100%, independent of the network density. This was because the dissemination range was optimized so that no sensors outside the query region were activated by the query. For query result recall, we observed that this metric increased with network density. We attributed this to the nature of network connectivity – when network density was low, some sensors may have failed to find a neighbor to relay query results, thus producing a low recall score. As the network density increased, more query results were able to be route back to the mobile object.

Figure 20(b) shows results when packet collisions were allowed and detected at the receiving side, and when using a "decay" sensor-to-sensor communication model. Again, the precision score was 100% since no matter what MAC-layer protocol was used and no matter what sensor radio communication model was used, sensors outside the query region were not activated. But, the recall score was lower than 20%, regardless of sensor density. Again, at low network density this was due to the fact that many sensors simply fail to find a neighbor to relay query results. As density increased, each node had more neighbors to help relay query results; however, there was then a higher probability for collisions from the routing of query results. Since we were using a MAC-layer protocol with receiver-side collision detection and no retransmission, collisions had a significant negative impact on the number of query results that were received at the mobile object, decreasing the recall score. This clearly demonstrated the value of employing more complex MAC-layer protocols, at the cost of increasing sensor node energy consumption and query processing delay.

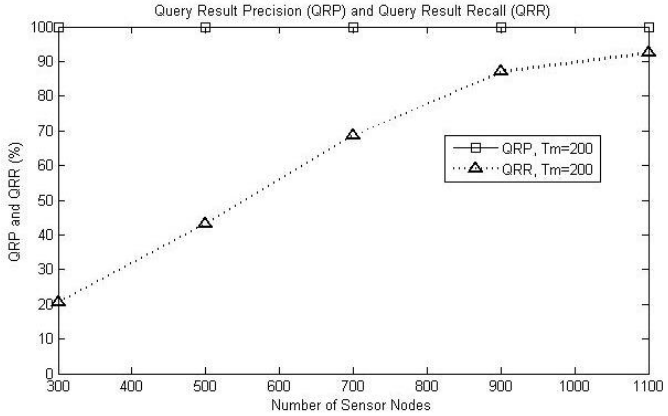

Figure 20(a). Case 1: No-Collision/Binary Communication Model

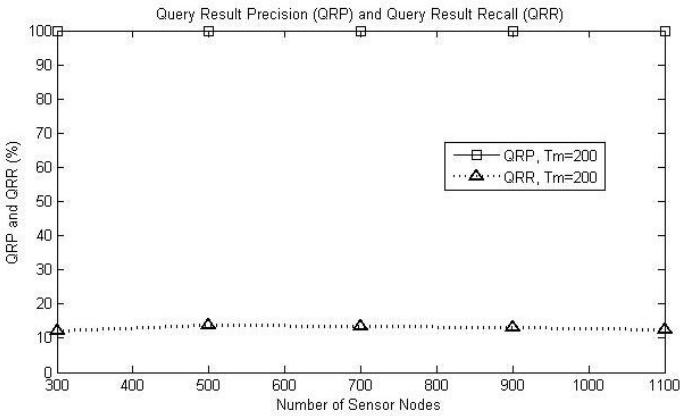

Figure 20(b). Case 1: Collision/Decay Communication Model

8.5.2 Case 2: Injection point is inside the query region

Case 1 is clearly a special case. We now consider a more general case, when the injection point is within the query region, but not at the center. See Figure 21. As in Case 1, the precision score was 100% for both setting of the MAC-layer protocol and sensor radio model. For the most optimistic setting of communication parameters (Figure 21(a)), the recall score was less than 25%. This was due to the reduction of the transmission range at query injection for the purpose of optimizing the precision score. For purposes of comparison, one interesting question is: What is the best recall we can achieve if we ignore precision? In other words what is the recall score if the maximum transmission range is used during query injection (250m). The results showed an 80% recall at the highest sensor density level; but the tradeoff was that the precision score fell to about 60%.

For the analysis that allowed query result collisions (Figure 21(b)), the recall score was less than 20%, as it was in Case 1, and for the same main reasons. Again, for the purpose of comparison, we considered what would happen if we allowed the query injection at maximum transmission range (250m). The result was a very small improvement in recall with a 10% decrease in precision. We also observed that the precision score at maximum transmission range was not affected by network density as significantly as when we used the optimistic setting of communication parameters (Figure 21(a)), and was only slightly improved. This was because query results from sensor nodes outside the query region – which could improve the precision score – were more likely dropped as a result of packet collisions at higher network density.

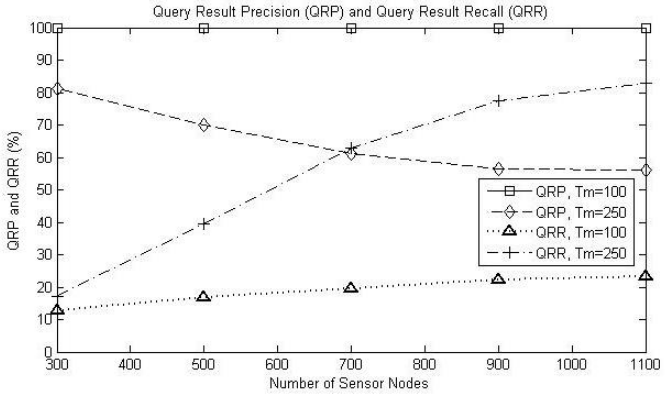

Figure 21(a). Case 2: No-Collision/Binary Communication Model

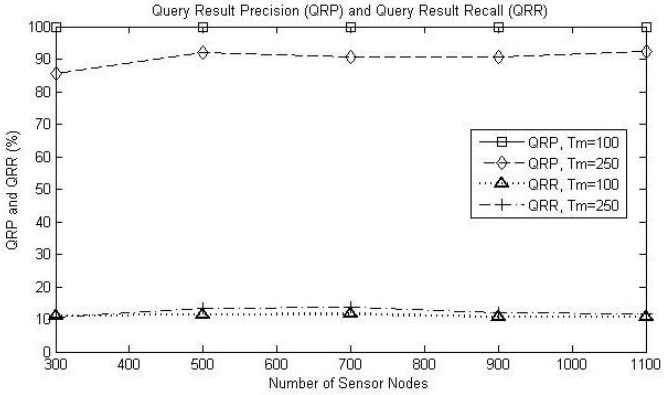

Figure 21(b). Case 2: Collision/Decay Communication Model

8.5.3 Case 3: Injection point is at the edge of the query region

In Section 6.2 we discussed the need for setting a lower-bound threshold on the transmission range of mobile objects for query injection, for cases such as this one. Without this threshold, the transmission range could be reduced to a theoretical value of zero in order to maximize the *DRCR* metric. For our simulation studies, we set the threshold as 50m, the same as the assumed transmission range for sensor nodes. For the optimistic setting of communication parameters (Figure 22(a)), we observed that the recall score was only about 3% and was not significantly affected by network density, due to the fact that (1) a 50m dissemination range could only cover a small portion of the query region and (2) all the activated sensors were located within 50 meters of the mobile object, which allowed all query results to reach the injecting mobile object in one hop. The precision score was a bit less than 50%. When we maximize recall, by letting the query injection transmission range be 250m, we achieved slightly better than 50% recall at the highest network density, but with some smaller decrease in precision.

For the analysis that allowed collisions (Figure 22(b)), we observed that the precision and recall scores, when using the optimized transmission range (50m), were basically the same as in the upper-bound case of Figure 22(a). This is because we assumed the sensor-to-mobile-object communication success rate was 100%, and there was no effective loss of query results due to sensor-to-sensor message collisions or signal strength decay. Remember, in this case all activated sensor nodes could forward their query results directly to the mobile object. When we again maximized the recall score by letting the transmission range at query injection be its maximum, we found that the recall was improved by 3% while the precision dropped by more than 5%.

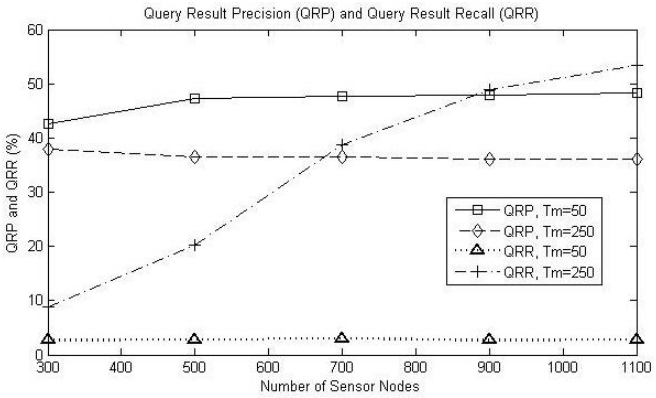

Figure 22(a). Case 3: No-Collision/Binary Communication Model

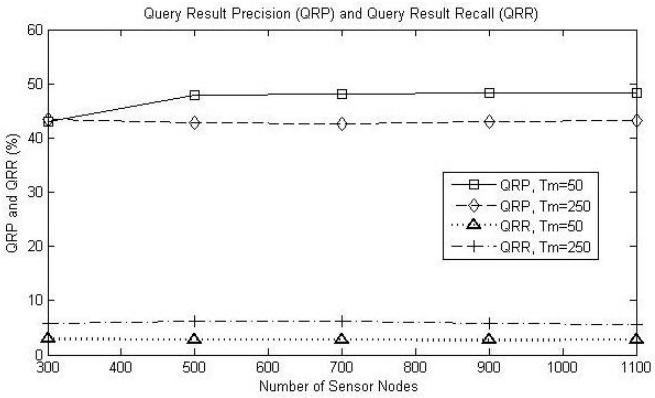

Figure 22(b). Case 3: Collision/Decay Communication Model

8.5.4 Case 4: Injection point is outside the query region

In Section 8.4 we observed that the probability of some mobile objects moving to a position that is within a query region was higher when more mobile objects existed in the environment or when the query region itself was large. As we observed in Cases 1 and 2 above, this resulted in a perfect precision score of 100%. However, there may be cases when no mobile object that is carrying a query can reach such an optimal injection point; thus we now present such a case. As in the previous cases, we first considered the upper-bound results based on an optimistic setting of communication parameters (Figure 23(a)). We found no dramatic changes in the precision score when increasing the injection transmission range, and the precision reached about 30% at higher sensor-node density (700 sensor nodes or more). When we used the maximum injection transmission range, the recall score was improved, but only achieved a score of slightly over 40% at the highest network density. The point is that there was simply an inherent limitation on the quality that could be achieved for these query results due to the physical location of the injecting mobile object.

When we performed the analysis that allowed collisions (Figure 23(b)), we observed that both quality metrics decreased, especially the recall score. The main reason for the low recall score in this case is the same as in the previous cases. Yet in this situation, when we maximized the recall score by letting the query injection transmission range be its maximum, we discovered no drastic changes in either metric. This is because we had only increased the transmission range by 60 meters (from 190 to 250). Furthermore those sensors that were newly activated by this increase in transmission range were located relatively far from the injection point. Thus, those query results had a fairly small possibility of actually reaching the injecting mobile object due to the increasingly likelihood of collisions along the path from the sensor node to the injecting mobile object.

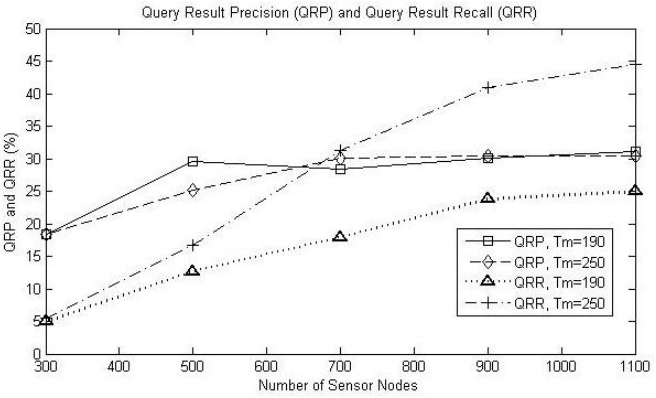

Figure 23(a). Case 4: No-Collision/Binary Communication Model

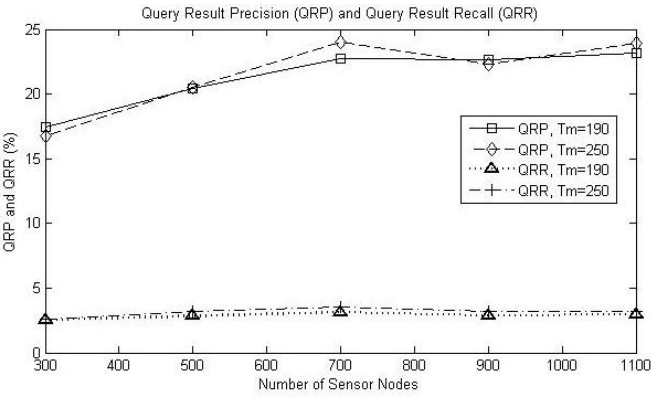

Figure 23(b). Case 4: Collision/Decay Communication Model

8.6 Deliverability of Intended Query Result and Sensor Energy Consumption

In the query result routing phase, we proposed a strategy of restricted flooding with an arbitrary target. This was used to balance the tradeoff between the deliverability of intended query results and energy consumption due to query result routing. Here we present some results that were derived numerically to analyze these two properties.

Figure 24 shows numerical results of $P_{MO\text{-}IFA}$ with $RHC_{initial}$ changing from 1 to $RHC_{\max}$, when varying the number of mobile objects in the environment. The environment size Ω was 5,000 x 5,000, with 500 sensor nodes, Tm = 500, λ =250, D_{min} =500, T_S = 50.

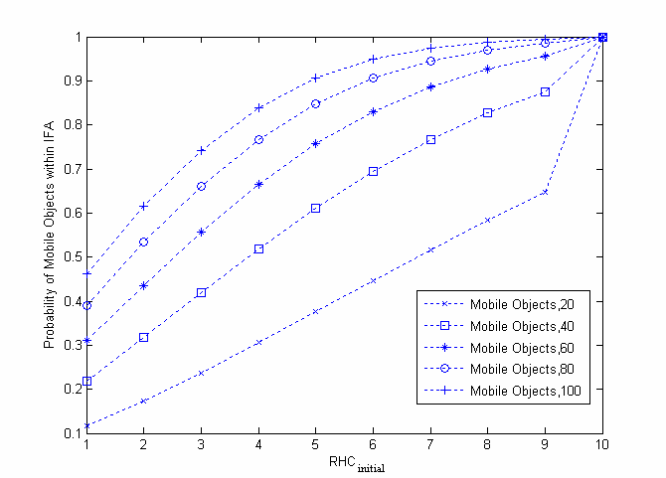

Figure 24. Probability of a mobile object being located within intended flooding region.

The deliverability of r_i was 90.54% when number of mobile objects was 100, $RHC_{initial}$ =5 and $RHC_{\max}$ =10. We can see that the probability of delivery reached the significance level 90% when $RHC_{initial}$ was 5, which was much less than the hop count value of RHC_{MAX} = 10. So, with very high probability, intended query results could reach some mobile objects other than M_i using fewer hops then needed to reach M_i.

Figure 25 plots the expected energy consumption during r routing with variable $RHC_{initial}$. The result was normalized to the maximal energy consumption when $RHC_{initial} = RHC_{MAX}$. It clearly demonstrated the significant energy savings that was achieved by using a small $RHC_{initial}$ value.

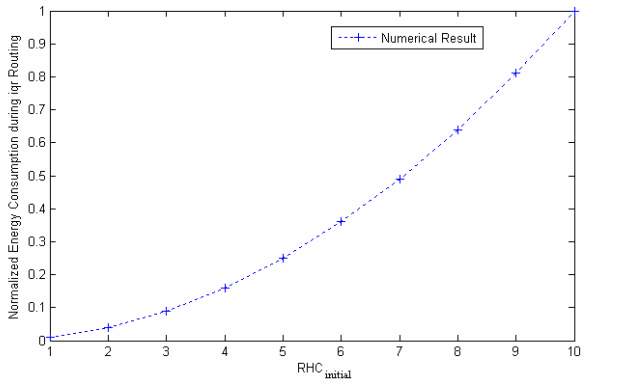

Figure 25. Normalized energy consumption with different $RHC_{initial}$

Table 2 presents the deliverability of r_i and the normalized energy consumption (NEC) for different values of $RHC_{initial}$ (1-10). The results showed that $P_{MO\text{-}IFA}$ increased with a decreasing rate, but energy consumption increased with an increasing rate when we used a higher $RHC_{initial}$. For the example of $RHC_{initial}$ = 5, the normalized energy consumption is 0.25, but $P_{MO\text{-}IFA}$ was 0.9054. This meant that there was a 90.54% chance of some mobile object being located within the intended flooding region and that 75% of the energy consumption was saved in comparison to the case of $RHC_{initial}$= 10.

Table 2. Deliverability of r_i and associated normalized energy consumption (NEC)

$RHC_{initial}$	1	2	3	4	5
$P_{MO\text{-}IFA}$	0.4615	0.6146	0.7419	0.8384	0.9054
NEC	0.01	0.04	0.09	0.16	0.25
$RHC_{initial}$	6	7	8	9	10
$P_{MO\text{-}IFA}$	0.9482	0.9736	0.9874	0.9944	1
NEC	0.36	0.49	0.64	0.81	1

9 Conclusion and Future Research

We presented a query processing architecture and associated techniques for networks composed of mobile objects operating within the context of a sensor rich environment. Key properties of mobile objects are used to offset the constraints associated with sensor nodes, such as the fact that sensor nodes can be location-ignorant. The main contribution of this work is a delay-tolerant, end-to-end query processing approach that consists of four key phases: query generation, query routing, query injection, and query result routing.

One limitation of our approach is that it does not address all potential sources for query failures, where we consider a query failure as the inability to obtain any query result within the specified time and location constraints. For example, with insufficient density or sparse distribution of mobile objects, it may be infeasible to inject queries to a specified query region. In such cases the idea of using opportunistic query routing is not sufficient and other approaches must be considered. Assuming that queries can be effectively injected into a query region, there are other potential sources of query failure associated with query-result delivery. These were discussed in Section 7.4 and can be the subject of future research. Further research can also be conducted on the idea of exploiting multiple query injections, meaning that more than one mobile object might be selected to carry and inject a given query. This might be useful as a fault-tolerance strategy for various forms of mobile object failures, or for optimizing those query retrievals where the coverage of a query region has higher priority than conserving sensor node energy. One first-cut idea for soliciting multiple injecting mobile objects is to revise the geographic routing used for query routing by allowing the last relay mobile object, selected to carry the query, to not only carry the query, but to also broadcast the query to its neighbors. So these neighbors then also become potential query injectors. Some specific research questions that arise with this approach are: How to coordinate among the set of potential query injectors to avoid unproductive use of sensor node energy at injection time? How to merge or evaluate the quality of query results associated with multiple injections? Future research can also include an integrated end-to-end simulation of the four query processing phases. Finally, as the simulation results of Section 8.5 indicate, it can be useful to perform further study on optimizing query result routing techniques and MAC-layer protocols to improve query result precision and query result recall

References

[1] R. Stoleru, T. He, J. Stankovic and D. Luebke, "High-Accuracy, Low-Cost Localization System for Wireless Sensor Networks," *The Third ACM Conference on Embedded Networked Sensor Systems*, San Diego, USA, Nov. 2005.

[2] A. Trigoni, Y. Yao, A. Demers, J. Gehrke and R. Rajaraman, "Hybrid Push-Pull Query Processing for Sensor Networks," *Proceedings of the Global Internet Workshop on Sensor Networks* (WSN), Karlsruhe, Germany, Feb. 2004.

[3] B. Gedik, L. Liu and P. Yu, "ASAP: An Adaptive Sampling Approach to Data Collection in Sensor Networks," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 18, No.12, pp. 1766-1783, Dec. 2007.

[4] Y. Yu, B. Krishnamachari and V. K. Prasanna, "Energy Minimization for Real-Time Data Gathering in Wireless Sensor Networks," *IEEE Trans. on Wireless Communication*, Vol. 5, No. 11, pp. 3087-3096, Nov. 2006.

[5] S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, G. Ahn and A. T. Campbell, "MetroSense Project: People-Centric Sensing at Scale," *The Workshop on World-Sensor-Web* (WSW 2006), Boulder, Oct. 2006.

[6] H. Luo, F. Ye, J. Cheng, S. Lu and L. Zhang, "TTDD: Two-layer Data Dissemination in Large-scale Wireless Sensor Networks," *ACM Wireless Networks*, Vol. 11(1-2), pp. 161-175, Jan. 2005.

[7] H. S. Kim, T. F. Abdelzaher and W. H. Kwon, "Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks," *The* First ACM Conference on Embedded Networked Sensor Systems, Los Angeles, Nov. 2003.

[8] J. Pan, L. Cai, Y. Hou, Y. Shi and X. Shen, "Optimal Base-Station Locations in Two-Layered Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, Vol. 4(5), pp. 458-473, 2005.

[9] R. Shah, S. Roy, S. Jain and W. Brunette, "Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks," *The First IEEE International Workshop on Sensor Network Protocols and Applications*, Alaska, USA, May 2003.

[10] O. Gnawali, B. Greenstein, K. Jang, A. Joki, J. Paek, M. Vieira, D. Estrin, R. Govindan and E. Kohler, "The TENET Architecture for Tiered Sensor Networks," *The 4th ACM Conference on Embedded Networked Sensor Systems*, Boulder, CO., Nov. 2006.

[11] S. Tian and S. M. Shatz, "Optimizing Query Injection from Mobile Objects to Sensor Networks," *The 8th International Symposium on Autonomous Decentralized Systems*, Arizona, USA, Mar. 2007.

[12] S. Tian, S. M. Shatz and Y. Yu, "A Framework for Querying Sensor Networks Using Mobile Devices," *The First International Workshop on Distributed Sensor Systems* (DSS'07), Hawaii, USA, Aug. 2007.

[13] WILLWARN: http://prevent-ip.org/en/prevent_subprojects/safe_speed_and_safe_following/willwarn/

[14] V. Kottapalli, A. Kiremidjian, J. Lynch, E. Carryer, T. Kenny, K. Law and Y. Lei, "Two-Tiered Wireless Sensor Network Architecture for Structural Health Monitoring," *The 10th Annual International Symposium on Smart Structures and Materials*, San Diego, USA, March, 2003.

[15] J. Gomez and A. T. Campbell, "A Case for Variable-Range Transmission Power Control in Wireless Multihop Networks," *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.
[16] T. He, C. Huang, B. M. Blum, J. A. Stankovic and T. F. Abdelzaher, "Range-Free Localization and its Impact on Large Scale Sensor Networks," *ACM Trans. on Embedded Computing Systems* (TECS), Vol. 4, Issue 4, 2005.
[17] Y. Zhu, M. Gao and L. Ni, "Distributed Localization Refinements for Mobile Sensor Networks," *The International Conference on Computer Networks and Mobile Computing*, Zhangjiajie, China, Aug. 2005.
[18] R. Iyengar, K. Kar and S. Banerjee, "Low-coordination Topologies for Redundancy in Sensor Networks," *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing***,** Urbana-Champaign, IL, May 2005.
[19] L. Ma, X. Cheng, F. Liu, F. An and J. Rivera, "iPAK: An In Situ Pairwise Key Bootstrapping Scheme for Wireless Sensor Networks," *IEEE Trans. on Parallel and Distributed Systems*, Vol. 18, No. 8, Aug. 2007, pp. 1174 – 1184.
[20] L. Gu and J. Stankovic, "Radio-Triggered Wake-Up Capability for Sensor Networks," *The 11th IEEE Real-Time and Embedded Technology and Applications Symposium*, Toronto, May 2004.
[21] First ACM Workshop on Vehicular Ad Hoc Networks (*VANET 2004*), Philadelphia, PA, October 2004.
[22] X. Li, G. Calinescu and P. Wan, "Distributed Construction of Planar Spanner and Routing for Ad Hoc Networks," *Proceedings of IEEE INFOCOM*, New York, USA, Jun. 2002.
[23] J. Li and S. M. Shatz,"Toward Using Node Mobility to Enhance Greedy-Forwarding in Geographic Routing for Mobile Ad Hoc Networks," *The International Workshop on Mobile Device and Urban Sensing* (*MODUS 2008*), St. Louis, MO, April 2008.
[24] H. Frey and I. Stojmenovic, "On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad Hoc and Sensor Networks," *The 12th ACM International Conference on Mobile Computing and Networking*, Los Angeles, CA, Sept., 2006.
[25] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Senor Networks," *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, Aug. 2000.
[26] G. Xing, C. Lu, R. Pless and Q. Huang, "On Greedy Geographic Routing Algorithms in Sensing-covered Networks," *The 5th ACM Symp. on Mobile Ad Hoc Networking and Computing* (MobiHoc), Tokyo, Japan, May 2004.
[27] Y. C. Tseng, S. Y. Ni, Y. S. Chen and J. P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *ACM Trans. on Wireless Networks*, Vol. 8, No. 2, pp. 153-167, Mar. 2002.
[28] W. Lou and J. Wu, "Towards Broadcast Reliability in Mobile Ad Hoc Networks with Double Coverage," *IEEE Trans. on Mobile Computing*, Vol 6, No. 2, pp. 148-163, Feb. 2007.
[29] C. Bettstetter, G. Resta and P. Santi, "The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, Vol. 2, No. 3, July 2003, pp. 257-269.
[30] P. Levis and D. Culler, "Maté: A Tiny Virtual Machine for Sensor Networks," *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating systems* (*ASPLOS X*), San Jose, CA, Oct. 2002.
[31] A. Woo, T. Tong and D. Culler, "Taming the Underlying Challenges of Multihop Routing in Sensor Networks," *The 1st ACM Conference on Embedded Networked Sensor Systems* (*SenSys*), Los Angeles, CA, Nov. 2003.
[32] J. Polastre and R. Szewczyk and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," The ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2005), Los Angeles, CA, Apr. 2005.

APPENDIX

Lemma 1: For case 1, *QRCR* is a decreasing function with respect to the distance measure *D*, where

$$QRCR = \frac{\left[\frac{1}{2}\lambda^2 * (\alpha - \sin(\alpha)) + \frac{1}{2}Tm^2 * (\beta - \sin(\beta))\right]}{\pi * \lambda^2}$$

Proof:
First, note that α, β and γ are all functions of *D*. Taking derivative over *D* on both sides:

$$\frac{d(QRCR)}{d(D)} = \frac{\left[\frac{1}{2}\lambda^2 * (1 - \cos(\alpha)) * \alpha' + \frac{1}{2}Tm^2 * (1 - \cos(\beta)) * \beta'\right]}{\pi * \lambda^2}$$

$$= \frac{\left[\lambda^2 * (\sin(\frac{\alpha}{2}))^2 * \alpha' + Tm^2 * (\sin(\frac{\beta}{2}))^2 * \beta'\right]}{\pi * \lambda^2}$$

$$= \frac{\left[\frac{1}{4} * (2 * \lambda * \sin(\frac{\alpha}{2}))^2 * \alpha' + \frac{1}{4}(2 * Tm * \sin(\frac{\beta}{2}))^2 * \beta'\right]}{\pi * \lambda^2}$$

$$= \frac{\left[\frac{1}{4} * (|\ GH\ |)^2 * \alpha' + \frac{1}{4}(|\ GH\ |)^2 * \beta'\right]}{\pi * \lambda^2}$$

$$= \frac{\left[(|\ GH\ |)^2 * (\alpha' + \beta')\right]}{4\pi\lambda^2}$$

$$= \frac{\left[(|\ GH\ |)^2 * (2\pi - 2\gamma)'\right]}{4\pi * \lambda^2}$$

$$= \frac{-\left[(|\ GH\ |)^2 * \gamma'\right]}{2\pi * \lambda^2} \quad (1)$$

Now, consider γ'.

From Figure 7, we observe that $D^2 = Tm^2 + \lambda^2 - 2\lambda * Tm * \cos(\gamma)$.

Taking derivative over *D* on both sides:

$$\frac{d(D^2)}{dD} = \frac{d(Tm^2 + \lambda^2 - 2\lambda * Tm * \cos(\gamma))}{D}$$

$$2D = 2\lambda * Tm * \sin(\gamma) * \gamma' \quad (2)$$

Since γ is one angle of triangle (IP, G,S), $0 \leq \gamma \leq \pi$.

Thus, $\sin(\gamma) \geq 0$. (3)

In (2), *D,* λ and *Tm* are all greater than zero and $\sin(\gamma) \geq 0$ (from (3)). Therefore, $\gamma' > 0$. (4)

Now, (1) can be written as:

$$\frac{d(QRCR)}{d(D)} < 0$$

So, *QRCR* is a decreasing function with respect to the distance measure *D*.