

Research Article

Datacollection in Event-Driven Wireless Sensor Networks with Mobile Sinks

Harshavardhan Sabbineni and Krishnendu Chakrabarty

Department of Electrical and Computer Engineering, Duke University, Durham, NC 27705, USA

Correspondence should be addressed to Harshavardhan Sabbineni, hvardhan@ee.duke.edu

Received 22 May 2009; Revised 5 December 2009; Accepted 5 July 2010

Copyright © 2010 H. Sabbineni and K. Chakrabarty. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a two-tier distributed hash table-based scheme for data-collection in event-driven wireless sensor networks. The proposed method leverages mobile sinks to significantly extend the lifetime of the sensor network. We propose localized algorithms using a distributed geographic hash-table mechanism that adds load balancing capabilities to the data-collection process. We address the hotspot problem by rehashing the locations of the mobile sinks periodically. The proposed mobility model moves the sink node only upon the occurrence of an event according to the evolution of current events, so as to minimize the energy consumption incurred by the multihop transmission of the event-data. Data is collected via single-hop routing between the sensor node and the mobile sink. Simulation results demonstrate significant gains in energy savings, while keeping the latency and the communication overhead at low levels for a variety of parameter values.

1. Introduction

Wireless sensor networks (WSNs), consisting of large numbers of sensor nodes, have enabled a new monitoring, and sensing paradigm for large geographical areas. WSN applications include detection of forest fires, habitat monitoring and target tracking in the battlefield. In a sensor network, the data generated by individual sensor nodes is forwarded to the sink for further processing.

Typical sensor network applications generate large amounts of data and send that data to the base station using multihop routing. However, transporting large quantities of data to the base station can quickly drain the limited energy resources of the sensor nodes and reduce the lifetime of the sensor network. One way to significantly reduce communication cost in sensor networks is to perform in-network aggregation (e.g., AVG and MIN) [1, 2] or in-network processing (e.g., beamforming [3, 4]). However, due to the inherent loss of detail, these techniques do not provide the fine data granularity desired by several sensor network applications. Often times, all of the data generated by the sensor network is necessary for the purposes of accuracy and modeling in these applications. For instance, it has been noted that in a networked structural health

monitoring application [5], more than 500 samples per second are required to efficiently detect damage. Another example is the Sonoma Redwoods project [6] where the biologists researching the project require as much detailed data from the sensor network as possible to try various physical models and test various hypotheses over the data.

Besides, such multihop routing with static sink nodes results in the early death of the one-hop neighbors of the sinks and make the sensor network unusable. This is expected because most of the data traffic is relayed to the sink by these nodes, thus greatly increasing their energy consumption, resulting in their untimely death and partitioning of the network topology. Consequently, nodes located remotely will be unable to report to the sink, which greatly reduces the lifetime of the WSN. Similarly, in hostile environments, it may not be feasible to replenish the batteries of the sensor nodes.

Recently, mobile data sinks have been proposed as a solution for datacollection to geographically balance the energy consumption among the sensor nodes throughout the network. Mobile sinks, either proactively or reactively, move in the sensor network and collect the data from the sensor nodes. This not only solves early death problem of the one-hop neighbors of the sink but also extends the network

lifetime by distributing the responsibility of relaying data to the sink among many nodes in the sensor network. The sink mobility assumption may be useful for many applications such as target tracking, intrusion detection, and habitat monitoring. It is also possible to use this strategy of mobile sinks in other environments, where mobile sinks can traverse the sensor network area and collect the data based on the protocols.

Even though a mobile sink is desirable, it poses new challenges for designing energy-efficient sensor network protocols. As the sink moves in the sensor field, it leads to high control overhead to find a route to the sink and route packets to it, which may potentially offset the energy saved from a mobile sink strategy. Furthermore, frequent changes to the sink location can result in an increase in packet latency.

The majority of published methods assume periodic sensing, where sensors are continuously monitoring the network and reporting data to the sink. However, there are a large number of applications (e.g., intrusion detection, environmental monitoring, habitat monitoring) where an event-driven approach is more appropriate. In event-driven sensor networks, data is only generated when events occur.

In this paper, we present an energy-efficient datacollection protocol for event-driven sensor networks. This protocol, which is executed in a distributed fashion on every node in the network, uses a two-tier geographic hash table-based scheme for datacollection. The proposed mobility model moves the sink node only upon the occurrence of an event according to the evolution of current events, so as to eliminate the energy consumption incurred by the multihop transmission of the event-data. Data is collected via single-hop routing when the mobile sink is closer to the sensor nodes. Simulation results demonstrate significant gains in energy savings, while keeping the latency and the communication overhead at very satisfactory levels for various parameter values.

The rest of this paper is organized as follows. In Section 2, we describe the related work. In Section 3, we explain the geographic hash table-based datacollection scheme. In Section 4, we provide our analytical framework to evaluate the performance of the proposed protocol. We present other datacollection protocols for comparison in Section 5 and then present our simulation results in Section 6. Finally, we summarize the paper in Section 7.

2. Related Work

Several approaches have been proposed for data collection in static sensor networks. Partial Network Coding is proposed as a tool for continuous data collection in [7]. Parts of data segments are combined using network coding techniques to extend the network lifetime. A cascading data collection mechanism [8] is proposed for periodic data collection in wireless sensor networks. During each round, only a subset of the nodes send data directly to the sink. Each intermediate node combines all the information using network coding and forwards the data to the sink. An adaptive sampling approach to data collection is proposed in [9]. Data is directly

collected from a dynamically changing subset of sampler nodes whereas data for the non-sampler nodes is predicted based on the use of probabilistic models. Although this is an energy-efficient approach for periodic data collection, it is only proposed for static sensor networks. Enhancing data collection by leveraging temporal correlation between data is explored in [10]. Methods for individual and aggregate data collections are proposed in [11]. An offline algorithm to compute the optimal data update strategy as well as online algorithms to cope with message losses have been proposed for static wireless sensor networks. These schemes do not consider mobile sinks for extending the network lifetime.

Several mobile-sink-based approaches are also proposed for data collection in the literature. These approaches can be classified into three categories based on the mobility model of the sink: random mobility, predictable mobility, and controlled mobility of the mobile sink. A sensor network architecture that collects data using Mobile Ubiquitous LAN Extensions (MULEs) that leverages the random mobility of mobile agents has been proposed in [12]. In Sensor Networks with Mobile Agents (SENMA) [13], data is sent directly to the mobile agent that is flying above the sensor field. Cluster-based data collection is described in [14]. The authors assume that the mobile sink is either a helicopter, airplane or an LEO satellite and hence the sensors have direct access to the mobile sink. Mobile sink periodically collects the data from the sensor nodes. This scheme is not optimized for event-based sensor networks like HTDC. In [15], authors propose three protocols for energy-efficient data collection in sensor networks with multiple mobile sinks. Mobile sinks leave an imprint of their movement in the sensor network and coordinate to reduce the overlap area for data collection amongst each other. Mobile sinks continuously move and collect data from the sensor nodes. This strategy can waste a significant amount of energy for the mobile sinks in event-driven sensor networks where events occur at discrete times. Multi-hop operation between the mobile sink and the source nodes is considered in [16]. Forecast algorithms are used to predict the location of the next event. The work [17–21] describe several applications that perform data collection with mobile sinks.

Two decentralized mobility models for data collection are proposed in [22]. In each mobility model, a data collection request is given bids by all mobile sinks and the mobile sink with the lowest bid services the data collection request. However, each sensor node sends an individual data collection request to the mobile sinks and this consumes more energy compared to our proposed scheme. The security of the data collection mechanism using mobile sinks is studied in [23]. The authors propose a secure data collection mechanism based on one-way hash chains in which the sensors must verify the source of the beacon messages sent by the mobile sink before they send the data to the mobile sink. In [17], data collection by multiple mobile sinks in underwater acoustic networks is studied and three algorithms are presented. Some of the problems identified are the same issues we have attempted to solve in this paper. However, the scheme that we present here is more energy-efficient due to the two-tier hash table-based design.

In this paper, we propose a solution that is significantly different from all the above approaches. We assume an event-driven scenario, where sensors that have detected an event send data to the sink node via single-hop routing. Our goal is to control the mobility of the sink so as to ensure an energy-efficient operation of the network. The sink node is notified about the occurrence of events, and its trajectory so as to maximize network lifetime.

3. HTDC Design

Hash Table-based Data Collection (HTDC) is a reactive protocol: each sensor node reactively initiates the data collection process after an event is sensed by the sensor node. It proposes controlled mobility for the mobile sink to improve the energy-efficiency of the mobile sinks in an event-driven sensor network.

HTDC uses a two-level geographic hash table-based mechanism to select the event announcer node locations and the mobile link locations. This is more energy-efficient compared to a random selection of these nodes. If the random selection mechanism were to be used, the location of the mobile sinks and the announcer nodes must be communicated to all the sensor nodes. This location information can then be used by the sensor nodes to route event-announcement data packets to the event announcer nodes and the mobile sinks. The communication cost for broadcasting this information to all the sensor nodes is $O(mn)$ where m is the number of mobile sinks and n is the total number of sensor nodes. The broadcast complexity is prohibitively high. On the other hand, if fixed locations are used for the mobile sinks and the event announcer nodes, the hotspot problem will reduce the sensor network lifetime dramatically.

HTDC binds individual sensor nodes in the network to the cells of a virtual grid based on their geographic location. This limits the cost of event announcement to an individual cell. Each node maintains only the information necessary to determine the location of the Local Event ANnouncer (LEAN) node and no additional state needs to be maintained. This provides fault tolerance to the sensor nodes in the case of the failure of an event announcer node as the underlying geographic routing algorithm routes to the sensor node closest to the location specified as the destination and not to a specific node.

HTDC achieves three goals. First it ensures correctness by attaching each sensor node to a mobile sink. Second, it adaptively selects the positions for the local event announcer nodes in each cell as well as the agent nodes for the mobile sinks. Third, by dividing the sensor field into virtual cells among each mobile sink, it ensures load balancing among available mobile sinks.

HTDC runs above the MAC and routing layers. HTDC utilizes geographic routing protocol as the underlying routing protocol. Control packets from the sensor nodes notifying LEAN nodes of new events are routed to the LEAN nodes and control packets from the LEAN nodes are routed to the mobile sink.

3.1. System Model. Consider a sensor network, where N nodes are randomly deployed in a sensing field. A small number of mobile sink nodes are present in the sensor network and gather data from the sensor nodes. A specific application scenario is a battlefield, where a large amount of sensor nodes are scattered randomly, performing environment monitoring and intrusion detection. A vehicle travels through this field and collects data from sensor nodes. We make the following assumptions about the sensor network architecture.

- (i) All the sensor nodes are fixed and are aware of their location (either through localization techniques or through a GPS receiver). The maximum radio range is the same as r for all nodes.
- (ii) The mobile sink nodes are non-sensing nodes, and are aware of their own locations. However, it is not necessary for mobile sinks to know the locations of other sensor nodes.
- (iii) Mobile sinks can communicate with each other in a single-hop. Mobile sinks communicate with the base station via multihop routing.
- (iv) We assume that there are specific types of events in the sensor network. We recognize that an event has occurred when at least two sensor nodes have generated data packets related to that event.

3.2. Virtual Grid Construction. HTDC divides the sensor network into a virtual grid of cells. Each cell is a square. Each sensor node identifies itself with a single cell. For a particular sensor node at location $L_s = (x; y)$, the bottom left corner of the cell is located at $L_{bl} = (x_i; y_i)$ such that

$$\begin{aligned} x_i &= \left\lfloor \frac{x}{s} \right\rfloor, \\ y_i &= \left\lfloor \frac{y}{s} \right\rfloor, \end{aligned} \tag{1}$$

where s is the length of the side of each cell.

Division of the sensor field into uniform virtual grids is highly energy-efficient compared to other schemes such as Voronoi graph division [24]. The construction of Voronoi graph consumes more energy in resource-constrained sensor nodes. Besides, in uniform grid, each sensor node is implicitly aware of the boundaries of each cell unlike in a Voronoi graph division.

3.3. Geographic Hash Table. The data collection system architecture, which we describe in this section to meet the above-enumerated design criteria, is a two-level geographic hash table (GHT) [25]. GHT hashes a key k into geographic coordinates. A key-value pair is stored at a node in the vicinity of the location to which its key hashes. The hash function is chosen such that it spreads the different key names evenly across the geographic region where the sensor network is deployed. The home node for a GHT packet is the node geographically nearest to the destination coordinates of the packet. Other nodes around the hashed location are called the perimeter nodes. GHT has two modes, a greedy

mode and a perimeter mode. In greedy mode, GHT forwards a packet closer to the destination at each hop. Perimeter mode is used when no next hop neighbor can be found that is closer to the destination than the forwarding node. In perimeter mode, the packet is forwarded on the edge that is counter-clockwise about the forwarding node from the ingress edge. In GHT, the packet enters perimeter mode at the home node, then traverses the entire perimeter that encloses the destination, before returning to the home node. This protocol replicates stored data for a key k at nodes around the location to which k hashes in the perimeter nodes. When a mobile sink moves from its hashed location, the perimeter nodes store the data till the mobile sink returns to its hashed location. In HTDC, mobile sinks always return to the hashed location once the data collection in a cell is completed.

3.4. Hash Functions. HTDC utilizes two hash functions:

Hash1: This hash function determines the location of the event announcer node in a cell. This hash function takes the cell id of the source node as a parameter and outputs the location of the Event Announcer. Since a sensor node may not be present at the exact location as determined by the hash function, any sensor node that is the closest to the hashed location can act as the Event Announcer. Each event announcement message can travel through multiple hops in the cell and finally stops on a sensor node that is the closest to the location specified in the message. Hash1 is evaluated on the sensor nodes in each cell. The hash function is selected such that the location of the LEAN node is within the cell boundaries. The input to the hash function is the coordinates of the center of the cell and hence there will not be any collisions due to using Hash1 hash function.

Hash2: This hash function determines the location of the mobile sink. This hash function takes the mobile sink name as a parameter and outputs the location of the mobile sink. Each mobile sink stays at the location determined by Hash2 until a data collection request arrives. Hash2 is evaluated on the LEAN nodes to determine the location of the mobile sinks. Hash collisions are possible with Hash2 hash function. If there is a collision, location of more than one mobile sink will be the same. However, such a scenario can be avoided by the selection of an appropriate hash function.

3.5. Local Event Announcer Node. Figure 2 shows a sensor network with four mobile sinks where an event has occurred. After the occurrence of an event, all the sensor nodes that are within the area affected by the event, generate sensory data related to that event. Next, these source sensor nodes execute the *AnnounceEvent* procedure. The pseudocode for the *AnnounceEvent* procedure is shown in Figure 1. Each source node determines the location of the Event Announcer using a geographic hash table-based [25] method using the Hash1 hash function (Line 2). The source sensor

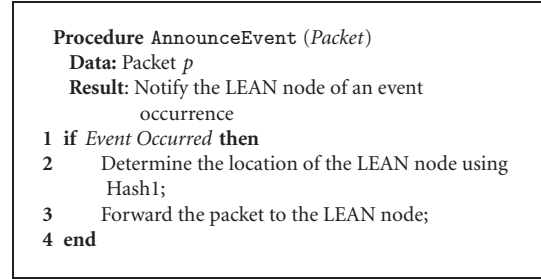


FIGURE 1: Procedure *AnnounceEvent*.

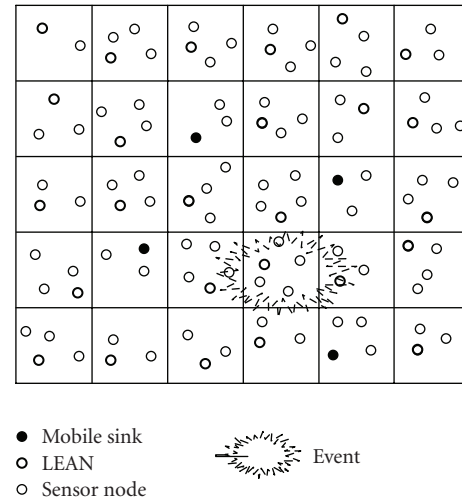


FIGURE 2: An event has occurred in the sensor network with four mobile sinks.

nodes forward an event announcement message called Local Event Announcement Messages (LEAM) to the Local Event Announcer (LEAN) node as shown in Figure 4 (Line 3). Local Event Announcer node is a sensor node present in the same cell as the source node that has data to be collected and whose main responsibility is to notify a mobile sink that there is data to be collected in its cell. There is one Event Announcer per cell in the sensor network except in a special case as described below.

The LEAN node stores the source information including the location coordinates of the source node. It waits for a preset number of event announcement message to arrive from the source nodes in its cell. Once event announcement packets are received, LEAN node executes the *SendDCRPacket* procedure. The pseudocode for the *SendDCRPacket* procedure is shown in Figure 6. The closest mobile sink is determined using the Hash2 hash function with the names of the mobile sinks (Line 1). This two-level hashing mechanism notifies the mobile sink of the event occurrence in the cell specified in the DCR message. Once the location of the mobile sink is determined, LEAN node constructs a Data Collection Request (DCR) packet (Line 2) and forwards the packet to the closest mobile sink (Line 3).

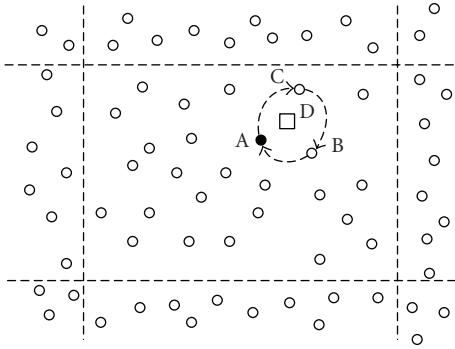


FIGURE 3: Local Event Announcer Node. D is the hashed location of the LEAN node. A is the LEAN node for this virtual cell. B and C are the perimeter nodes for the LEAN node.

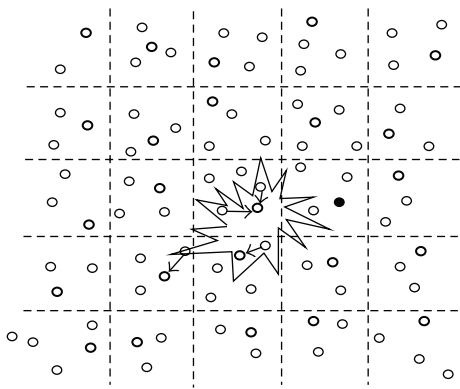


FIGURE 4: Source nodes in each cell send an event announcement message to LEAN.

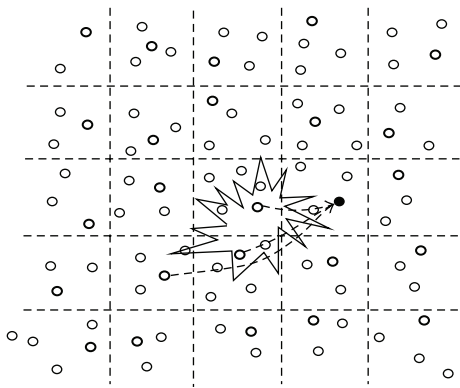


FIGURE 5: LEAN nodes send a DCR packet to the mobile sink to request data collection.

Once the event announcement message reaches it, the mobile sink initiates the data collection process by executing the *DataCollection* procedure as shown in Figure 6.

Each mobile sink also maintains a list of its immediate neighbors and they maintain the state of the mobile sink in turn. A mobile sink periodically refreshes the state of the perimeter nodes (mobile sink agents) by broadcasting beacon messages as shown in Figure 3. They can be in one of

Procedure SendDCRPacket (Packet)

Data: Packet p

Result: Send a data collection request packet to the mobile sink

- 1 Determine the location of the mobile sink node using Hash2;
- 2 Construct the DCRpacket;
- 3 Forward the packet to the Mobile Sinks;

FIGURE 6: Procedure SendDCRPacket.

Procedure DataCollection (Cell ID)

Data: Cell ID

Result: Collect data in the cell specified by the cell ID

- 1 Send beacon every time period;
- 2 Receive packets from sensor nodes;
- 3 Move in a trajectory;

FIGURE 7: Procedure DataCollection.

the three states, namely, “*present*”, “*not present*” and “*failed*”. When a mobile sink is present in its hashed location, the mobile sinks agents have the state of the mobile sink as “*present*”. The cell that contains the mobile sink will not have the Event Announcer node. Instead, nodes present in the cell send their data directly to the mobile sink. However, the mobile sink agents will still be present.

The pseudocode for notifying the mobile sink of an event occurrence by a LEAN node in HTDC is shown in Figure 7.

3.6. Data Collection. The event announcement message contains the location of the cell. The mobile sink moves to the cell where the event has occurred and initiates data collection among the nodes in the cell. The mobile sink traverses the cell in a snake-like fashion in order to cover the complete area of each virtual cell. We chose this mobility model for the mobile sinks as it covers all nodes in the cell for a square-shaped cell. The mobility model of the mobile sink depends on the shape of the cell and node density. It should allow the mobile sink to collect data from all the nodes in the cell. If the form of the cells is different, other mobility models can be considered.

The data is collected passively: sinks periodically broadcast a beacon message at distance r , thus alerting the sensors one-hop away to start sending their data. Sensor nodes that receive a beacon transmit the collected data stored in their memory, to the sender of the beacon. Note that this is strictly a one-hop communication scheme. A sensor node responds to the mobile sink’s message with the sensed data about the event with its data packet. The frequency of the beacon broadcast depends on the node density in each cell and also on the communication range of the sensor node. Once a mobile sink has collected the data from one sensor node and moves into the communication range of the next sensor node, it issues the next beacon. Once all the sensors in the cell are covered and if there are no pending requests, the mobile sink returns to its hashed location and waits for the next event occurrence.

3.7. *HTDC Messages.* HTDC uses three types of messages to perform data collection.

- (i) DCR: data collection request. When a local event announcer node has detected an event in its cell, it can communicate this new event to the mobile sink by transmitting a DCR message.
- (ii) PRQ: Perimeter Request. If a mobile sink agent receives a DCR message and the mobile sink is not at its location, then the mobile sink agent constructs this PRQ request and sends it to the next closest mobile sink node location.
- (iii) LEAM: Local event announcement. This message is sent by a source sensor node upon detecting an event.

3.8. *Load Sharing.* In this section, we discuss how our data collection mechanism enables load sharing. Sensor networks can have events whose event area can be of different sizes. In such cases, events can span multiple cells. To provide load sharing, the location of the mobile sinks is periodically changed. This will ensure that each mobile sinks stays at a different location in every period. This will also reduce the hotspots in the sensor network. Sensor nodes also periodically change the Event Announcer node in each cell to reduce the hotspot problem.

3.9. *Routing of Messages to the Mobile Sink.* The Event Announcer node communicates with the mobile sink using a DCR message as shown in Figure 5. If the mobile sink is currently servicing an event, the state of the mobile sink agent (MSA) nodes is set to “not present”. MSA nodes receive the Event Announcement messages meant for the mobile sink in its absence. Upon receipt of the Event Announcement messages, the mobile sink agent nodes generate a new Event Announcement message and forward it to the next nearest mobile sink. This will ensure that mobile sinks that are currently free from servicing any event can share the workload. In applications such as target tracking, the target continuously moves enabling sensor nodes to generate data continuously. In such a scenario, a sensor node will only report an event occurrence to the LEAN node if there is data in its cache that is yet to be collected and no event occurrence message has been sent to report this data. Once a mobile sink collects the data, the sensed data is removed from the node’s cache. If the same node senses data related to the same event later, the new data must be collected again by the mobile sink. The sensor node sends a new event announcement message to the LEAN node about the event occurrence. As the LEAN node is aware that the mobile sink is in the same cell, it informs the mobile sink that collects the data again.

A “Mobile Sink failed” message will be sent to the base station to inform the base station about the failure of the mobile sink.

4. Analysis

In this section, we analyze the energy consumption of HTDC.

4.1. *Model and Notation.* We consider a square sensor field of area A in which N sensor nodes are uniformly distributed. There are M mobile sinks in the sensor field. Each mobile sink moves at an average speed of v_m/s . When an event occurs at a location (x, y) , all sensors present within range of the sensing radius r_s from (x, y) generate data for the event and continuously generate data packets as long as the event persists.

4.2. *Energy Consumption.* The total energy consumption is the summation of three components:

- (1) Energy consumed during event notification by the source sensor nodes;
- (2) Energy consumed to notify the mobile sink;
- (3) Energy consumed during data collection. This can be divided into two parts: (a) energy consumed by the mobile sink and (b) energy consumed by the sensor nodes.

Now we analyze the energy consumption for each of these parts separately.

4.2.1. *Energy Consumed for Event Notification.* The energy consumed during event notification is directly proportional to the number of sensors that sensed the event. Suppose all the sensor nodes in a single cell sense the event. Each such sensor node sends an event notification message to the LEAN node. The maximum number of hops the packet has to traverse is equal to the diameter of the subnetwork inside the cell.

The number of nodes in each cell is given by

$$n_{\text{cell}} = N \times \frac{c^2}{A}, \quad (2)$$

where c is the side of each cell.

The expected number of transmissions needed to notify the LEAN node of an event by a single source node in the cell is given by

$$E_{\text{cell}}^1 = k \times \sqrt{n_{\text{cell}}}, \quad (3)$$

where k is a constant. The expected total energy consumed due to event notification in a single cell is given by

$$E_{\text{cell}} = E_{\text{cell}}^1 \times n_{\text{cell}}. \quad (4)$$

4.2.2. *Energy Consumed for Mobile Sink Notification.* Once LEAN node receives event notification messages from the source sensor nodes in the cell, it constructs a data collection request packet and sends the packet towards the mobile sink. The average distance between the mobile sink and the LEAN node can be evaluated by computing the average distance between two randomly chosen points in a unit square. Suppose (X_1, Y_1) and (X_2, Y_2) denote the coordinates of two random points which are selected independently and uniformly. The distance between these two points can be written as $D = \sqrt{(|X_1 - X_2|^2 + |Y_1 - Y_2|^2)}$. According to [26],

the probability density function of the random variable D is defined as follows:

$$f_D(d) = \begin{cases} 2\pi d - 8d^2 + 2d^3 & \text{if } 0 \leq d \leq 1 \\ 2(\pi - 2)d + 8d\sqrt{d^2 - 1} \\ -2d^3 - 4d \arccos\left(\frac{2 - d^2}{d^2}\right) & \text{if } 1 < d \leq \sqrt{2}. \end{cases} \quad (5)$$

Since there are M mobile sinks, suppose D_i denotes the distance between a LEAN node and a mobile sink M_i . D_i is a random variable and its probability density function is the minimum of the M i.i.d random variables where the probability density function of each random variable is given by(5)

$$f_{D_{\min}}(d) = M f_D(d) (1 - F_D(d))^{M-1}. \quad (6)$$

Since (6) does not yield a simple closed form solution, we simplify the analysis by assuming that each mobile sink will cover a square of area A/M . The side of the square is given by

$$s = \sqrt{\frac{A}{M}}. \quad (7)$$

Hence the probability density function of the distance between the event announcer node and the mobile sink is:

$$f_{D_{\min}}(d) = \begin{cases} 2\pi d - 8d^2 + 2d^3 & \text{if } 0 \leq d \leq s \\ 2(\pi - 2)d + 8d\sqrt{d^2 - 1} \\ -2d^3 - 4d \arccos\left(\frac{2 - d^2}{d^2}\right) & \text{if } s < d \leq s\sqrt{2} \end{cases} \quad (8)$$

The average distance between the cell and the mobile sink is given by

$$E[D_{\min}] = 0.52 \times s. \quad (9)$$

According to [27], given a geographic routing protocol, we have $H(l) = \zeta l/r$ where $H(l)$ is the number of hops on a path between two arbitrary nodes x and y such that $|x, y| = l$ is the Euclidean distance between the two nodes, r is the communication range, and $\zeta \geq 1$ a scaling factor depending on the spatial node density. Hence, the expected number of hops needed to reach the mobile sink from the event announcer node is given by

$$h_{\min} = \zeta \times \frac{E[D_{\min}]}{r}. \quad (10)$$

Hence the expected total number of transmissions that will be sent in HTDC is given by

$$P_{\text{total}} = h_{\min} + n_{\text{cell}} \sqrt{n_{\text{cell}}}. \quad (11)$$

which is directly proportional to the total energy cost E_{total} needed to notify the mobile sink of an event.

If there are a total of n_E events, then the expected total energy consumed is given by

$$E_{\text{total}}^{n_E} \propto n_E P_{\text{total}}, \quad (12)$$

4.2.3. Energy Consumed for Data Collection. The energy consumed by the mobile sink for data collection is proportional to the distance traveled. The energy consumed by a sensor node in the data collection process is for listening to the beacon messages from the mobile sink and transmission of data.

4.3. Time Complexity. In this section, we determine the delay in collecting the data using HTDC. The delay t_{total} is given by

$$t_{\text{total}} = t_c + t_{\text{msfetch}} + t_{\text{ms}} + t_{\text{dc}} \quad (13)$$

where

- (i) t_c is the amount of time it takes for the Event Announcer node to detect that an event occurred in its cell;
- (ii) t_{msfetch} is the amount of time taken for the DCR message to reach the mobile sink;
- (iii) t_{ms} is the amount of time it takes for the mobile sink to reach the cell;
- (iv) t_{dc} is the amount of time it takes for the mobile sink to collect the data from all the nodes in the cell.

The parameter t_c is the time taken by the LEAN node to detect an event is the time taken to receive the preset number of event detection messages from the source sensor nodes and is $O(d)$ where d is the diameter of the subnetwork in the virtual cell.

The parameter t_{msfetch} is the time taken by the DCR packet to reach the mobile sink from a LEAN node is $O(E[D_{\min}])$.

The parameter t_{ms} is the time taken by the mobile sink to reach the cell is given by the distance between the cell and the mobile sink location and the velocity of the mobile sink, v . This is $O(E[D_{\min}]/v)$.

The parameter t_{dc} is the time taken by the mobile sink to collect all the data in a given cell is proportional to the number of sensor nodes in the cell. Hence the overall time complexity of the HTDC protocol for data collection is $O(n_{\text{cell}})$.

5. Other Data Collection Algorithms

In this section, we describe the three datacollection algorithms against which we will compare the performance of HTDC.

5.1. Reactive Data Collection (RDC). In RDC, a source node that wants its data collected either floods the network with its data packet or if it has multiple data packets that needs to be collected reactively [28], locates the mobile sink by flooding the network with a query about the location of a mobile

sink. Once the query reaches the mobile sink, the sink moves toward the source node and transmits a beacon message to the source node indicating its readiness to collect the data from the source node. The algorithm converges when data from all the source nodes that detected the event is collected by the mobile sink. Reactive data collection consumes the least amount of time in notifying the mobile sink about the occurrence of the event. However, it consumes a vast amount of energy because it floods the network with query requests.

5.2. Continuous Data Collection (CDC). Several approaches such as [15] partition the network based on geography and allocate one mobile sink to each partition. This is a continuous data collection approach where each mobile sink continuously moves in its allocated grid partition in rounds. In each round, a mobile sink moves in a designated trajectory that enables it to reach all the nodes in the grid partition. The mobile sink will broadcast a beacon message querying each sensor node for data as it moves closer to the node. Data is collected from all the source nodes in the grid partition in each round via one-hop routing. If a node does not have data in a certain round, it will not respond to the beacon message of the mobile sink and it is ignored by the mobile sink.

5.3. Ideal Data Collection (IDC). In IDC, each mobile sink immediately knows about the occurrence of an event in its grid partition and which sensor nodes are the source nodes with data about that event. The mobile sink immediately moves towards the source sensor nodes after the occurrence of the event and collect the data. We call this *ideal data collection* since in this case, the sink reaches the source nodes in the shortest possible amount of time and no energy is expended in notifying the mobile sink about the occurrence of the event. We expect this scheme to provide an upper bound for the amount of energy consumed. To simulate ideal data collection, we maintained a centralized database of event locations in our simulator. This database is used to notify the mobile sinks about the location of the event upon event occurrence.

6. Performance Evaluation

We used a 150 node network within a 250×250 m monitoring area. In the simulations, a LEAN node only sends a DCR to a mobile sink after at least two nodes in the cell have reported the occurrence of the event. The results are averaged over 10 simulations runs and are shown with 95% confidence intervals. The vertical lines on the bars for all graphs indicate the confidence intervals. Confidence intervals are calculated using the method of independent replications [29] for these simulations. We obtain a single output variable in each of the simulation runs and its distribution is not known. We can use statistical inference based on normal distribution (because of the Central Limit Theorem) to determine the confidence intervals. However, since the number of simulation runs is small, we have used the Students t distribution and

TABLE 1: Radio characteristics [30].

Radio mode	Power consumption (mW)
Transmit (T_x)	14.88
Receive (R_x)	12.50
Idle	12.36
Sleep	0.016

TABLE 2: Simulation environment parameters.

Bandwidth	30 Kbps
Terrian	250×250 m
Number of nodes	100
Node placement	Uniform
Radio range	20 m

the following formula is used to obtain the $100(1 - \alpha)\%$ confidence interval of the population mean μ :

$$\bar{\theta} - t_{n-1, \alpha/2} \frac{S_n}{\sqrt{n}} < \mu < \bar{\theta} + t_{n-1, \alpha/2} \frac{S_n}{\sqrt{n}}, \quad (14)$$

where $\bar{\theta}$ is the sample mean, S_n is the sample variance and n is the number of simulation runs.

6.1. Energy Model. Each sensor node is assumed to have a radio range of 20 m. The bandwidth of the radio is assumed to be 20 Kbps. The sensor characteristics are given in Table 1. These values are taken from the specifications for the TR1000 radio from RF Monolithics [30].

6.2. Simulation Environment. We have compared the different data collection approaches using the J-SIM sensor network simulator [31]. J-SIM is an open-source network simulator that provides a modeling and simulation environment for wireless sensor networks. To implement the HTDC protocol we used the GPSR extensions to the J-SIM simulator. The general simulation environment is summarized in Table 2.

6.3. Energy-Efficiency of the Virtual Grid. HTDC divides the sensor monitoring area into several virtual cells. The monitoring area can be divided into cells in a number of ways. We compare the performance of virtual grid with the division of the sensor field using voronoi graphs. LEAN are selected randomly to become a voronoi site. Upon occurrence of an event, sensor nodes in the Voronoi site send the event announcement packets to the LEAN node which in turn sends a DCR packet to the mobile sink. The results are shown in Figure 8. As the cell size increases, Voronoi graph construction increases much more than using a uniform grid. We have also compared the election of a LEAN node using clustering and using hashing in a uniform virtual grid. Election of a LEAN node using clustering mechanism consumes energy in cluster-head election. As the size of the cell increases, the cost of electing a cluster head increases. HTDC performs better compared to the other two methods

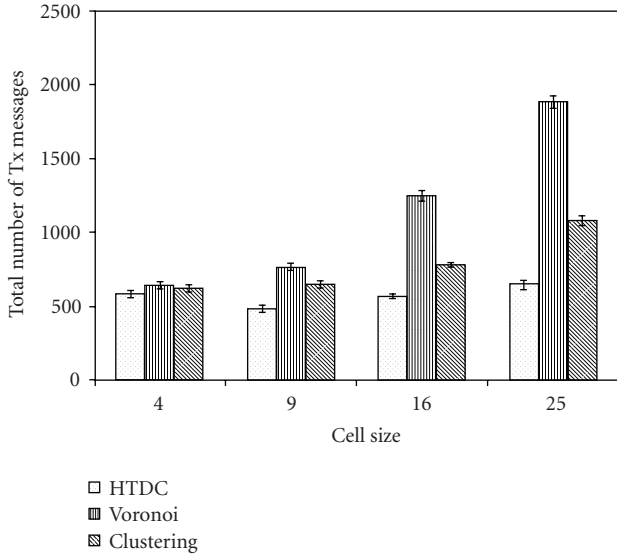


FIGURE 8: Total number of transmissions versus cell size.

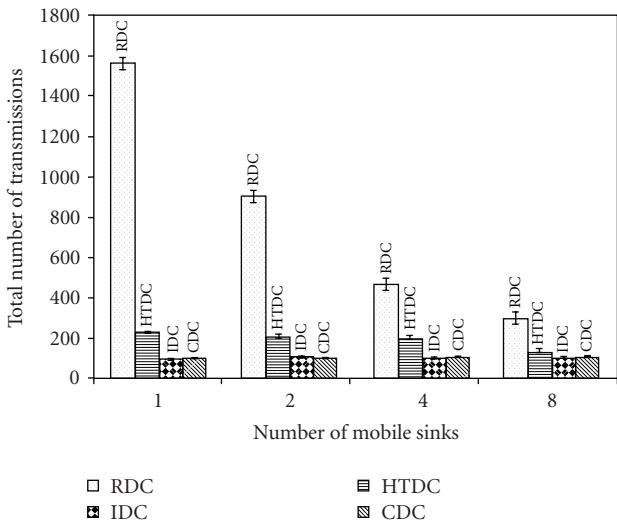


FIGURE 9: Total number of transmissions versus number of mobile sinks.

because a geographic hash table required only a minimal setup.

6.4. Effect of Mobile Sinks. First we show the effect of the number of mobile sinks on the network lifetime as well the average data collection delay. For this simulation, one event is generated for this experiment and the event persists for a period of 100 seconds and can be sensed by sensor nodes in an area of 3600 m² which is approximately equal to one cell. Each cell in the virtual grid contains 9 sensor nodes. Sensor nodes generate data every 10 seconds and hence each sensor node that has sensed the event generates 10 packets for this event. Mobile sinks move at an average speed of 3 m/s [32]. As expected, RDC provides an upper bound in terms of the total number of messages transmitted in the network. This

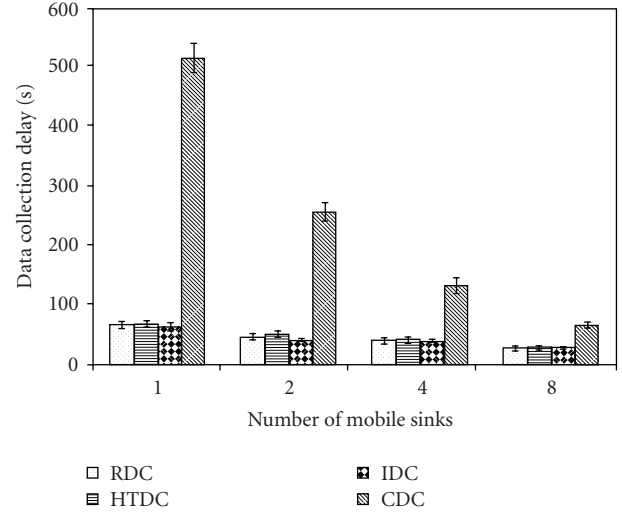


FIGURE 10: Datacollection delay versus the number of mobile sinks.

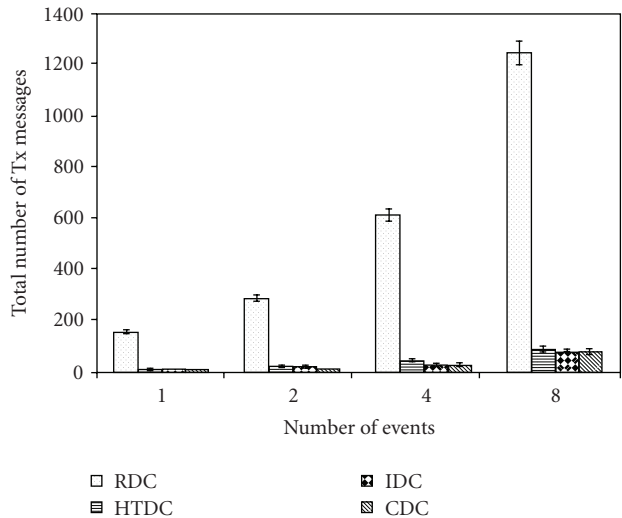


FIGURE 11: Total number of transmissions versus number of events.

measure is inversely proportional to the network lifetime. However, Figure 9 also shows that even the RDC protocol benefits from more mobile sinks in the network.

Next, we show the effect of varying number of mobile sinks on datacollection delay. The data collection delay is defined as the total time taken to collect all the data generated by the events. Figure 10 shows that RDC achieves low datacollection delay as flooding the network usually informs the mobile sink quickly about the event occurrence. HTDC performs equally well in terms of the datacollection delay with only using very few messages, thus improving the network lifetime. CDC performs the worst as the mobile sink continuously traverses the entire grid partition without any knowledge of the event occurrence. This shows that the continuous data collection protocols are unsuitable for event-driven sensor networks.

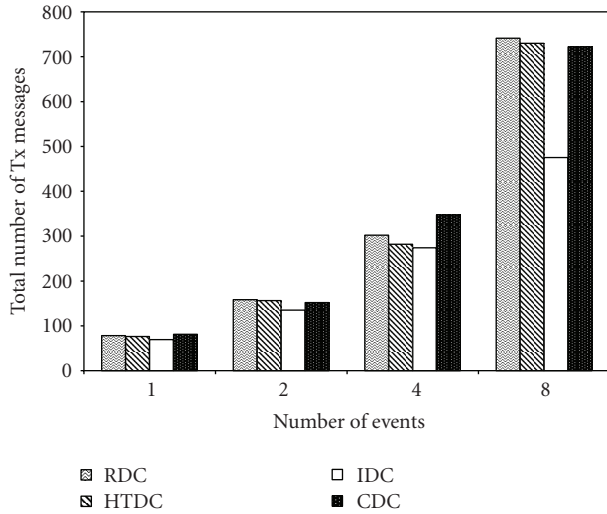


FIGURE 12: Datacollection delay versus number of events.

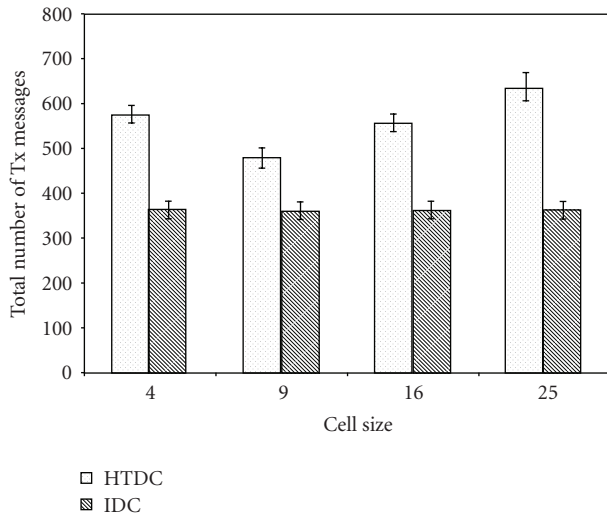


FIGURE 13: Total number of transmissions versus cell size.

6.5. Effect of the Number of Events. We have also investigated the effect of the number of events on the performance of HTDC. Figure 11 shows the total number of transmissions with an increase in the number of events. For this experiment, we placed four mobile sinks in the sensor field. As the number of events increase, the number of transmissions increase for all four protocols. However, RDC performs the worst as source sensor nodes flood the network with data collections requests. HTDC performs as well as the IDC that uses global information for data collection.

Figure 12 shows the effect of increasing the number of events on the data collection delay. The 95% confidence intervals are within 1% of the mean and hence not shown. As the number of events increase, we can see that data collection delay for CDC decreases compared with other methods. This is because as the number of events increase, CDC benefits from the fact that event occurrences are continuous in this simulation and occur sequentially.

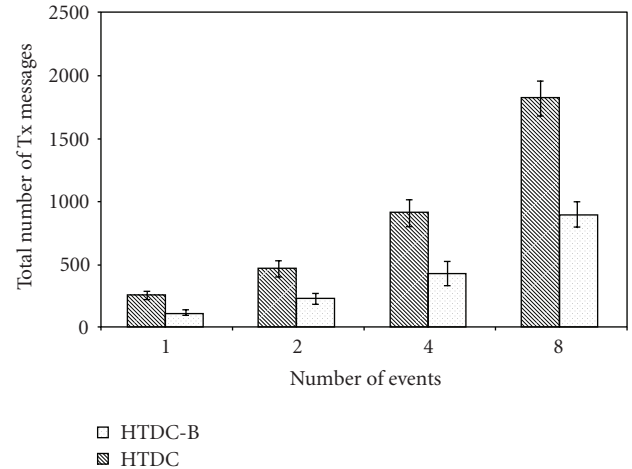


FIGURE 14: Total number of transmissions versus number of events.

6.6. Effect of Cell Size. Figure 13 shows the effect of varying the cell size on the performance of HTDC. The cell size is described in terms of the number of nodes present in the cell. As the cell size increases, the number of transmissions decreases initially but increases from cell size 16 nodes and beyond. This is due to the fact that when the cell size is small, the number of transmissions enroute to the mobile sink dominate the total number of transmissions but as the cell size gets larger, the number of transmissions inside the cell dominate. This can be observed clearly from Figure 13.

6.7. Effect of DCR Broadcast. In HTDC, each LEAN node sends a DCR message to the closest mobile sink as a unicast. If the mobile sink is not present at its location, a PRQ message is sent to the next closest mobile sink. On the other hand, each LEAN node can also broadcast the DCR message to all the mobile sinks. We compared the energy savings of HTDC compared with a modified version of HTDC where we broadcast the DCR, HTDC-B and the results are as shown in Figure 14. We found that HTDC realizes higher energy savings due to fewer message transmissions compared to a broadcast. A major problem with the broadcast approach is that once all mobile sinks receive the message, if all of them attempt to respond to the DCR message and move towards the source cell where the event has occurred, it would be a wastage of resources since only one mobile sink is sufficient. Hence, a handshake mechanism to identify which mobile sink will collect the data is required, leading to further consumption of precious energy resources.

6.8. Impact of Error in Localization. Errors in localization do not have any effect on the performance of HTDC as the geographic-routing methods correctly route even when localization errors are present. A sensor node with an incorrect location estimate can wrongly identify its virtual cell. In such a scenario, the sensor node will forward its event notification announcements to a different LEAN node. This will only lead to a few more additional messages compared to a network with zero errors in location estimation. Hence,

we expect the effect of errors in localization on the energy-efficiency of HTDC to be negligible.

7. Conclusions

Mobile data sinks have been proposed in the literature as a solution for datacollection to balance the energy consumption among the sensor nodes and improve the network lifetime. In this paper, we have presented HTDC, which leverages mobile sinks to significantly extend the lifetime of the sensor network through the use of a two-tier geographic hash table. The proposed mobility-management method moves the sink node only upon the occurrence of an event according to the evolution of current events to minimize the energy consumption incurred by the multihop transmission of the event-data. Data is therefore collected in this method only via single-hop routing. Simulation results demonstrate significant gains in energy savings for various parameter values.

Acknowledgments

The authors thank professor John Board and Dr. John Zachary for helpful comments on an earlier draft of this paper. The author are indebted to professor Kishor Trivedi, who helped improve the results through the use of statistical inference methods.

References

- [1] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: tiny AGgregate queries in ad-hoc sensor networks," in *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, Boston, Mass, USA, 2002.
- [2] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, "TiNA: a scheme for temporal coherency-aware in-network aggregation," in *Proceedings of the 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE '03)*, pp. 69–76, ACM, New York, NY, USA, September 2003.
- [3] H. Ochiai, P. Mitran, H. V. Poor, and V. Tarokh, "Collaborative beamforming for distributed wireless ad hoc sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4110–4124, 2005.
- [4] G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 81–88, ACM, New York, NY, USA, April 2004.
- [5] K. Chintalapudi, T. Fu, J. Paek et al., "Monitoring civil structures with a wireless sensor network," *IEEE Internet Computing*, vol. 10, no. 2, pp. 26–34, 2006.
- [6] G. Tolle, "Sonoma redwoods data," 2005, <http://www.cs.berkeley.edu/get/sonoma>.
- [7] D. Wang, Q. Zhang, and J. Liu, "Partial network coding: concept, performance, and application for continuous data collection in sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, pp. 1–22, 2008.
- [8] H. Li, H. Yu, L. Li, and A. Liu, "On the cascading data collection mechanism in wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 2479–2482, September 2007.
- [9] B. Gedik, L. Liu, and P. S. Yu, "ASAP: an adaptive sampling approach to data collection in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1766–1783, 2007.
- [10] C. Liu, K. Wu, and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 1010–1023, 2007.
- [11] X. Tang and J. Xu, "Adaptive data collection strategies for lifetime-constrained wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 6, pp. 721–734, 2008.
- [12] D. Wang, J. Xu, J. Liu, and F. Wang, "Mobile filtering for error-bounded data collection in sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS '08)*, pp. 530–537, June 2008.
- [13] A. Kinalis and S. Nikolettseas, "Scalable data collection protocols for wireless sensor networks with multiple mobile sinks," in *Proceedings of the 40th Annual Simulation Symposium (ANSS '07)*, pp. 60–69, March 2007.
- [14] M. Lotfinezhad, B. Liang, and E. S. Sousa, "Adaptive cluster-based data collection in sensor networks with direct sink access," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 884–897, 2008.
- [15] I. Chatzigiannakis, A. Kinalis, S. Nikolettseas, and J. Rolim, "Fast and energy efficient sensor data collection by multiple mobile sinks," in *Proceedings of the 5th ACM International Workshop on Mobility Management and Wireless Access (MobiWac '07)*, pp. 25–32, ACM, New York, NY, USA, 2007.
- [16] Z. Vincze, D. Vass, R. Vida, A. Vidács, and A. Telcs, "Adaptive sink mobility in event-driven multi-hop wireless sensor networks," in *Proceedings of the 1st International Conference on Integrated Internet Ad Hoc and Sensor Networks (InterSense '06)*, p. 13, 2006.
- [17] J. Wang, D. Li, M. Zhou, and D. Ghosal, "Data collection with multiple mobile actors in underwater sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 216–221, June 2008.
- [18] X. Zhang, H. Wang, A. K. R. Ansari, and G. Chen, "Distortion analysis for real-time data collection of correlated fields in randomly distributed sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 2211–2215, May 2008.
- [19] D. England, B. Veeravalli, and J. B. Weissman, "A robust spanning tree topology for data collection and dissemination in distributed environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 5, pp. 608–620, 2007.
- [20] M. Vahabi, M. F. A. Rasid, G. M. F. Hossein, and R. S. A. R. Abdullah, "Adaptive MAC protocol for wireless sensor networks in periodic data collection applications," in *Proceedings of the International Symposium on Information Technology (ITSim '08)*, vol. 4, pp. 1–8, August 2008.
- [21] J. M. Glasgow, G. Thomas, E. Pudenz, N. Cabrol, D. Wettergreen, and P. Coppin, "Optimizing information value: improving rover sensor data collection," *IEEE Transactions on Systems, Man, and Cybernetics Part A*, vol. 38, no. 3, pp. 593–604, 2008.

- [22] S. Hanoun, D. Creighton, and S. Nahavandi, "Decentralized mobility models for data collection in wireless sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 1030–1035, May 2008.
- [23] A. Rasheed and R. Mahapatra, "Secure data collection scheme in wireless sensor network with mobile sink," in *Proceedings of the 7th IEEE International Symposium on Networking Computing and Applications (NCA '08)*, pp. 332–340, July 2008.
- [24] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, New York, NY, USA, 1992.
- [25] S. Ratnasamy, B. Karp, L. Yin et al., "GHT: a geographic hash table for data-centric storage," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 78–87, September 2002.
- [26] E. Uysal-Biyikoglu and A. Keshavarzian, "Throughput achievable with no relaying in a mobile interference network," in *Proceedings of the 8th IEEE International Symposium on Computers and Communications (ISCC '03)*, p. 641, 2003.
- [27] S. De, A. Caruso, T. Chaira, and S. Chessa, "Bounds on hop distance in greedy routing approach in wireless ad hoc networks," *International Journal of Wireless and Mobile Computing*, vol. 1, no. 2, pp. 131–140, 2006.
- [28] H. Wada, P. Boonma, and J. Suzuki, "Macroprogramming spatio-temporal event detection and data collection in wireless sensor networks: an implementation and evaluation study," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS '08)*, p. 498, January 2008.
- [29] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, John Wiley & Sons, Chichester, UK, 2002.
- [30] RF Monolithics Inc, "Ash transceiver's designers guide," 2003, <http://www.rfm.com/>.
- [31] A. Sobeih, J. C. Hou, L.-C. Kung et al., "J-Sim: a simulation and emulation environment for wireless sensor networks," *IEEE Wireless Communications*, vol. 13, no. 4, pp. 104–119, 2006.
- [32] J. Rao, T. Wu, and S. Biswas, "Network-assisted sink navigation protocols for data harvesting in sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 2887–2892, April 2008.