Efficient Query-Based Data Collection for Mobile Wireless Monitoring Applications

LONG CHENG1, YIMIN CHEN2, CANFENG CHEN2, JIAN MA3, LEI SHU4, ATHANASIOS V. VASILAKOS5 AND NAIXUE XIONG6,*

1State Key Lab of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China
2Department of Electrical Engineering, Stanford University, Standford, CA 94305, USA
3Nokia Research Center, Beijing, China
4Department of Multimedia Engineering, Osaka University, Osaka, Japan
5Department of Computer and Telecommunications Engineering, University of Western Macedonia, Kozani, Greece
6Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA
*Corresponding author: nxiong@cs.gsu.edu

Considering sensor nodes deployed densely and uniformly a mobile sink moving through the sensing field queries a specific area of interest for monitoring information. The Query packet, injected by the mobile sink, is routed to the specific area and the corresponding Response packet is expected to return via multi-hop communication. In this paper, we analyze such a network model to address the problem of efficient data collection for mobile wireless monitoring applications. We first propose a meeting position-aware routing (MPAR) protocol for routing the Response packet efficiently and then propose an efficient query-based data collection scheme (QBDCS) for mobile wireless monitoring applications based on the MPAR. In order to minimize the energy consumption and packet delivery latency, the QBDCS chooses the optimal query time of injecting the Query packet and tailors the routing mechanism for sensor nodes forwarding packets. Simulation study has verified the analysis and demonstrated that the QBDCS can significantly reduce the energy consumption and end to end delivery latency.

Keywords: wireless sensor networks; mobile sinks; data collection; wireless monitoring

Received 26 August 2009; revised 2 December 2009
Handling editor: Yu-Chee Tseng

1. INTRODUCTION

In recent years, wireless sensor networks (WSNs) for wireless monitoring have been designed and developed in both military and civil areas for a wide variety of applications, including smart battlefield, environment and habitat monitoring, health care, home automation, traffic control etc. [1]. Some of these applications, for example, environment and habitat monitoring, require a large number of tiny sensors while these sensors usually operate on limited non-rechargeable battery power. Furthermore, a WSN is always expected to last over several months or years with limited power supply. Thus, energy saving becomes one of the most critical design issues for WSNs, while other constraints, such as throughput and fairness, become relatively less important.

To extend the lifetime of a WSN for wireless monitoring application, one common approach is to dynamically schedule each sensor’s work/sleep cycle (or duty cycle) [2]. When a sensor is in the sleep mode, the sensor’s processor will be turned off, but a low-power timer or some other triggering mechanism may be running to wake up the sensor at a later time, therefore it consumes only a tiny fraction of the energy, compared to that consumed in the active mode. More specifically, a radio will be turned off when it is not actively sending and receiving, and it will be turned on when communication is expected [3].
The duty-cycling operation in WSNs can be loosely categorized into two main types: (1) random duty-cycling, where sensors are turned on and off in a random fashion independent of each other; and (2) coordinated duty-cycling, where sensors coordinate via communication and information exchange to collectively achieve an on/off schedule followed by multiple sensors. With regard to random duty-cycling, its advantage lies in its simplicity and without any additional communication overhead for coordination among neighbors. However, randomly turning off the wireless radio inevitably results in a larger delay for each hop [4].

Energy efficient data collection in wide-deployed WSNs is a well-studied problem. Aiming at solving this problem, researchers have proposed to exploit mobility of sinks for energy efficient data collection in WSNs [5]. It is believed that this can balance the energy consumption of each sensor node and reduce the possibility of routing hot spots, which are introduced by fixed sinks due to the nearby heavy data flow.

The mobile sinks can be generally categorized into two kinds: (1) using dedicated mobile robots which are specially designed for collecting sensed data; and (2) employing existing mobile objects such as mobile handsets, vehicles or ships, on which the data collection devices are usually mounted.

For the dedicated mobile robots, the motion trajectory of mobile sinks is always carefully designed and controlled to improve the performance of WSNs, such as [6, 7]. However, in real-world scenarios, mobile sinks are more likely to be the combination of data collection devices and existing mobile objects with random mobility, e.g. mobile handsets, notebook computers or vehicles. Some studies have been done for exploiting such mobile objects present in wireless monitoring fields to collect and deliver sensed data, such as Data MULEs [8], TSA-MSSN [9] and [10]. In this case, the mobility of sinks may be uncontrollable, but predictable in a short period to some degree [9]. For example, when a vehicle moves along the road with a digital map available, the prediction of its future position can thus be predicted based on the continuity of mobility.

The specific contributions of this paper mainly come from the following two points:

(i) a meeting position-aware routing (MPAR) protocol for routing the Response packet is proposed. MPAR predicts the position of the mobile sink at the time of meeting the Response packet. Thus, the Response packet can be directly forwarded toward this predicted position from the source node.

(ii) an efficient query-based data collection scheme (QBDCS) for mobile wireless monitoring applications is proposed based on MPAR. The QBDCS finds the optimal time to inject the Query packet, in order to complete a query-based data collection cycle with minimum energy consumption and delivery delay.

The rest of this paper is organized as follows. Section 2 surveys the related work. Section 3 presents the system model and problem statement. The detailed MPAR protocol and QBDCS design are presented in Sections 4 and 5, respectively. Simulation results are presented in Section 6. Finally, Section 7 concludes this paper.

2. RELATED WORK

Recently, the demand for WSNs in many monitoring applications has grown tremendously in both military and civil areas, such as surveillance system [15], habitat monitoring [16], environmental observation and forecasting [17], health monitoring [18] and intelligent traffic [12].

Geographic routing is considered to be an efficient and scalable data delivery scheme and is quite commonly adopted for information delivery in large-scale WSNs. It is assumed that the sensor nodes are location aware and the source node
knows where the destination is. Sensor nodes may have global positioning system (GPS), or an ad hoc way to figure out their coordinates. The basic idea for geographic routing is greedily forwarding data packets to the neighbor geographically closest to the destination. A number of research works on geographic routing in WSNs have been conducted, e.g. GPSR [19], GEAR [20], TPGF [21] and [22]. In [19], a packet is forwarded to a one-hop neighbor that is closer to the destination than the current node. This process is repeated until the packet reaches the destination, or the packet is blocked at a node whose one-hop neighbors are all farther away from the destination, which is defined as a hole. Some variants of geographic routing, such as [23–26], propose different routing metrics for different applications and network models.

Many hole-bypassing mechanisms have been studied. As proposed in GPSR [19], when a packet gets stuck at a node, it is routed by the right-hand rule counter-clockwise along a face of the planar graph until either it can be routed by greedy forwarding again, or it goes back to where it enters the perimeter mode, which means that no route to the destination is available. Another approach is to identify hole boundary nodes first, then use the knowledge of these identified boundary nodes to facilitate the hole-bypassing routing [27].

The problem of efficient data collection has been investigated by many researchers. Most researches are based on the assumption that data collection involves all nodes of a WSN. A typical representative is the LEACH protocol [28], which selects a small number of clusters in a self-organized manner. A node in each cluster designated as the cluster head, collects and fuses data from other nodes in its cluster and transmits the result to the base station. In [29], the authors proposed to balance the energy consumption of the entire network, instead of load balance within each cluster in LEACH. In [30], a two-tier data dissemination approach was proposed to provide data delivery to multiple mobile sinks. Each data source constructs a backbone structure (partitioning the whole network into several small grids.) for query and data transmission. When the sink requests any data, the query will only be flooded reaching the data source. However, the overhead associated with maintaining and recalculating the grid would be very high [31].

In fact, there is a number of queries that select only a subset of nodes in a network. Upon receiving a user’s query message, each sensor node that is corresponding to the query delivers the sensed data to the sink node. The authors in [32] argued that selective queries (query-based data collection) are an important class of queries that can benefit from algorithms that are tailored for partial node participation of a WSN. In [33], an on-demand localized data collection scheme is proposed. According to the fixed location of the sink node, packet propagation and data collection are localized in a small area. The query message is efficiently unicasted to the forwarding node in the queried area, and is forwarded into the network within a localized area. Sensed data is collected at the forwarding node, at which data is aggregated. The aggregated data is then sent to the fixed sink node. In contrast to the sink with fixed position in [33], we focus on the query-based data collection with mobile sinks.

The network model in this paper, which is similar to that of [34], as shown in Figure 1, is inspired by the recent research on WSNs applied in the area of mobile wireless monitoring, collecting data such as temperature, sound, pressure or pollutants. The Query packet is sent by the mobile sink to a WSN and the corresponding Response will be delivered from the queried area to the mobile sink. However, as a case study, the proposed network model in [34] is too application-specific and the algorithm is restricted to the conditions that are initially assumed. For example, some active sensor nodes, or Vise Sink in their work, need to be deployed along the mobile sink trajectory. The introduction of Vise Sinks is not only constraining a lot of applications, but also introducing significant delivery latency when they are too widely separated. In this paper, the network model considers a more general case without such assumptions. The application scenario can be that sensor nodes are densely and uniformly deployed in the sensing field where no Vise Sinks exist. Possible mobile sinks are different kinds of roving vehicles moving through the sensing field to query the WSN. Considering the continuity of mobile sink mobility and for simplicity, the analysis in this paper only focuses on a mobile sink moving along a straight path.

In [34], the authors have proposed an adaptive routing algorithm based on the instantaneous position estimation, which can avoid frequent broadcasting of the mobile sink’s position. However, they only considered the Response Propagation phase, in this paper we consider a complete query-based data collection cycle, including choosing the right moment to inject

![Figure 1](image-url)

**FIGURE 1.** Example of the considered application scenario in [34]. The mobile sink injects a *Query* in the WSN through the closest vice sink, which is then forwarded to the queried region. The *Response* packet is finally delivered through a Vice Sink to the mobile sink. However, the *Response* packet is forwarded toward the mobile sink’s instantaneous position, where the routing path as a whole deviates from the optimum (if directly forwarding toward the predicted packet-sink meeting position). Besides, the authors in [34] only considered the Response Propagation phase, and did not point out when to inject the *Query* packet.
the Query packet. Besides, their method was based on the mobile sink instantaneous position prediction. Each relaying sensor node estimates the instantaneous mobile sink position and then routes the data packet toward the virtual destination node according to a given metric (for instance Euclidean distance or energy-aware distance) calculated by the relaying sensor node. However, the routing path as a whole deviates from the optimum compared with the proposed MPAR protocol, because MPAR can forward the Response packet directly toward the predicted packet–sink meeting position.

3. SYSTEM MODEL AND PROBLEM STATEMENT

3.1. Common assumptions

We make the following common and reasonable assumptions.

(i) Sensor nodes are homogeneous, immobile, energy-constrained and expected to run for a long time. They are able to wirelessly communicate with neighbors in a short range.

(ii) Each sensor node has a duty cycle $D_c \in [1\%, 100\%]$. It periodically opens the radio to transmit sensed data, but at other times it sleeps to save energy.

(iii) Each sensor node can identify its geographic location and maintains a neighbor table for routing packets. The location information can be gained by running a localization algorithm [35].

(iv) The mobile sink has an estimation of its current mobility (velocity, direction and position), which can be obtained from the GPS etc. The mobile sink operates on a rechargeable battery and has much higher computation and communication capabilities than sensor nodes.

(v) The mobile sink can obtain the sensor node’s information, such as transmission range and duty cycle. This can be achieved by the initial negotiation procedure.

3.2. Sensor deployment strategy

For simplifying the analysis and without loss of generality, we consider that sensor nodes are deployed in the sensing field densely and uniformly, forming a two-dimensional grid [36], where inside each cell a sensor node is placed allowing a little deviation. For example, sensor nodes are manually placed at selected locations and the deployment tolerates a slight deviation, as shown in Fig. 2.

3.3. Hierarchical network structure

Considering the wireless monitoring applications with sensor nodes densely and uniformly deployed in the sensing field, a mobile sink moving through the sensing field with random speed queries a specific point or an area of interest for monitoring information from the physical world. The hierarchical network structure consists of three tiers: sensor nodes, on-demand cluster head and mobile sink. Each Query packet contains location information of the interested area. The sensor node closest to the center of the interested area elects itself as the cluster head, which is called the ‘on-demand cluster head’, and is responsible for gathering and aggregating data in the interested area. The aggregated data (Response packet) is then sent back to the mobile sink via multi-hop communication. As shown in Fig. 3, a vehicle is moving along a straight road, and it queries a WSN deployed around the road for wireless monitoring. The deployed WSN enables vehicles to obtain the conditions of the surrounding environment, e.g. the status of parking slots in a specific parking ground along both sides of the street.

3.4. Problem statement

A complete query-based data collection cycle is composed of three phases: (1) Query Propagation phase, where the initially injected Query packet is propagated to the on-demand cluster head within the interested area; (2) Data Aggregation phase, where the queried sensed data is aggregated and the Response packet is generated by the on-demand cluster head; and (3) The subsequent Response Propagation phase, where the
corresponding Response packet is forwarded and then delivered to the mobile sink.

The query-based data collection problem is conceptually formulated by the following question: how can the WSNs complete the transmission of the Query and Response packets with minimum energy consumption and delivery delay? The basic technique adopted is the geographical location-aware routing; a scheme of bypassing holes is also adopted during packet transmission.

To minimize the energy consumption and delivery latency for transmitting the Query and Response packets, the following specific approaches are taken: (1) the packet delivery velocity is estimated and the position of the mobile sink at the time it meets the Response packet is predicted; thus, the Response packet can be forwarded toward the meeting position via multi-hop wireless transmission directly with geographical location-aware routing; and (2) the optimal time for the mobile sink to inject the Query packet to a WSN is estimated.

4. MPAR PROTOCOL

In this section, we first briefly introduce the on-demand cluster head selection in the Data Aggregation phase. Then, a MPAR protocol is presented for forwarding the Response packet in the Response Propagation phase.

4.1. On-demand cluster head selection

Once a Query packet is sent by the mobile sink, which includes the location information of the interested sensing field (radius and center), it will be routed toward the specific area of interest until reaching a node which we call the on-demand cluster head, i.e. the sensor node closest to the center of the interested area.

At each hop, the current forwarding node calculates its distance from the center of the queried area. If the distance is less than a predefined threshold and there is no other neighboring node that has a closer distance, the node will elect itself as the on-demand cluster head. It then broadcasts a data aggregation request within the queried area. Finally, it aggregates the sensed data and generates a Response packet. Note that we use the Euclidean distance as the metric in cluster head selection and the routing process. The metric can also be combined with residual battery energy of nodes for the load balance consideration.

4.2. Packet delivery velocity estimation

The packet delivery velocity \( V \) is defined as the delivered distance a packet travels divided by its end-to-end delay. The end-to-end delay of a message propagating in WSNs is related to the node processing speed, overhead, congestion, collision, interference, duty cycling operation and numerous other factors [37].

Let \( L \) be the length of the packet, \( R \) the data transmission rate of each sensor node, \( d \) the distance between the data source and destination and \( r \) the communication range of each sensor node. Let \( d_{\text{hop}} \) be the average distance per hop (\( d_{\text{hop}} \leq r \)). Intuitively, the number of hops between a source and a destination is roughly linear with the distance \( d \) between them and the variance depends on the actual sensor nodes deployment. Taking the duty-cycled \( D_\text{C} \) operation into account, which introduces an additional delay at each hop, we have the estimated packet delivery velocity under the ideal condition as follows:

\[
V = \frac{d}{(L/R \cdot D_\text{C}) \cdot [d/d_{\text{hop}}]} \approx \gamma \cdot r \cdot R \cdot D_\text{C} / L, \quad 0 < \gamma \leq 1,
\]

(1)

where \([d/d_{\text{hop}}]\) is the estimated hop count and \( \gamma \) (\( \gamma = d_{\text{hop}} / r \)) is the modification coefficient in the packet delivery velocity estimation by taking into account the actual deployment of sensor nodes.

4.3. Average distance per-hop estimation

The Query packet is basically composed of the following elements:

(i) interested types of the sensed data (temperature, humidity etc.);
(ii) geographic location information of the interested sensing field (radius and center);
(iii) mobile sink mobility status and the timestamp;
(iv) routing hop counts.

During the Query Propagation phase, each relaying node increases the hop count. Thus, given each propagation distance and the hop counts, \( d_{\text{hop}} \) can be estimated at the on-demand cluster head.

4.4. Meeting position estimation

Let \( t_0 \) denote the query time point, when the mobile sink injects a Query packet to a WSN, let \( t_1 \) denote the time point of the corresponding Response packet being sent and \( t_2 \) denote the estimated packet-sink meeting time point. Let \( L_s \) and \( V_s \) be the length and estimated delivery velocity of the Response packet, respectively. Given the starting point \((x_1, 0)\) of the Query packet, the center of the interested sensing area \((a, b)\) and the velocity of the mobile sink \( V_s \), as depicted in Fig. 4, we have the estimated meeting position (EMP) \((x_2, 0)\) of the Response packet and the mobile sink.

\[
(t_1 - t_0) + \frac{\sqrt{(x_2 - a)^2 + b^2}}{V_t} = \frac{|x_2 - x_1|}{V_s},
\]

(2)

where

\[
V_t = \gamma \cdot \frac{r \cdot R \cdot D_\text{C}}{L_t},
\]

(3)
4.5. MPAR design

Based on the packet delivery velocity estimation and packet-sink meeting position estimation, the MPAR protocol for the Response Propagation phase is designed as follows.

4.5.1. Adaptive EMP modification

After the queried sensed data is aggregated and a Response packet is then generated by the on-demand cluster head, the cluster head first calculates the initial EMP. It then attaches this information to the packet and sends it to the next relaying node by adopting a geographic routing algorithm.

For each relaying sensor node, before forwarding the Response packet, it first recalculates the EMP and updates the corresponding fields in the Response packet, which is also calculated under the condition that the Response packet is forwarded toward the estimated destination in a straight line with speed \( V_r \), as shown in Fig. 5. The recalculated EMP may deviate from the initial EMP; possible reasons are that the propagation delay exceeds the initial estimation (e.g. due to bypassing ‘holes’), or the propagation delay is overestimated. Then the Response packet will be forwarded toward the new calculated EMP by adopting geographic routing algorithm. This recursive process will last until the Response packet arrives at or goes beyond the ‘notifying node’, as shown in Fig. 6. The ‘notifying node’ is referred as the sensor node nearest to the latest calculated EMP. When the Response packet bypasses a hole, it may arrive at a sensor node close to the road and beyond the ‘notifying node’, as shown in Fig. 7.

4.5.2. Chasing mode

When the Response packet arrives at the ‘notifying node’ and finds that the mobile sink has not passed and is not in its radio range, the ‘notifying node’ waits for the mobile sink to come and then delivers the packet. However, it is possible that by the time the Response packet arrives at the ‘notifying node’, the mobile sink has already passed, possibly because the mobile sink’s mobility varies significantly or the packet delivery velocity is overestimated. In this case, the packet enters into the Chasing Mode as depicted in Fig. 6. The packet follows the sink’s moving trajectory until it catches up with the mobile sink.

4.5.3. Bypassing holes

When a relaying sensor node finds that its one-hop neighbors are all farther away from the EMP, it means that this sensor...
node is facing a hole. A hole is likely to appear when a set of sensor nodes are dead due to energy exhaustion or damage.

The **Response** packet cannot progress toward the EMP by greedily examining its local neighborhood when facing a hole. In this case, the relaying sensor node will apply the recovery routing mechanism to route the **Response** packet to the one-hop neighbor that has the smallest positive angle formed between the EMP direction and next hop direction, as shown in Fig. 7. When the **Response** packet arrives at any node close to the road, although that node is not the ‘notifying node’, it just checks whether the mobile sink has passed or not. If has not passed, the node waits for the mobile sink to enter its radio range before delivering the **Response** packet. Otherwise, the **Response** packet enters into the Chasing Mode following the sink’s moving trajectory until it catches up with the mobile sink.

### 4.5.4. Deadline awareness

In order to avoid the case where the packet chases the mobile sink endlessly, there usually exists a limit on the period of propagation time that the user allows between sending out the **Query** and receiving the **Response** packet. This time limit is denoted as $T_{\text{deadline}}$. Similar to the TTL (time to live) in TCP/IP protocols, if the **Response** packet cannot arrive at the destination within $T_{\text{deadline}}$, it should be discarded.

However, determining the remaining time to $T_{\text{deadline}}$ at each intermediate node is not trivial due to the lack of a globally synchronized clock. To handle this problem, we borrow the mechanism in [38] by piggybacking the elapsed time to the packet so that the following forwarding node can determine the remaining time to deadline without a globally synchronized clock.

### 4.6. Summary of MPAR

The flowchart of the MPAR protocol is presented in Fig. 8. The on-demand cluster head is responsible for aggregating the queried sensed data and generating a **Response** packet. It then calculates the EMP for the **Response** packet. After that, it forwards the **Response** packet to the next relaying node. For a relaying node, once receiving a **Response** packet, it first checks whether it exceeds the time limitation $T_{\text{deadline}}$. If yes, the **Response** packet will be discarded. Then it checks whether it is the ‘notifying node’. If yes, the **Response** packet either enters into the Chasing Mode or waits for the mobile sink if the mobile sink is not in the relaying node’s radio range. If the relaying node is not the ‘notifying node’, it first recalculates the EMP and then forwards the **Response** packet to the next relaying node recursively. This process will last until the **Response** packet arrives at the ‘notifying node’ or a sensor node close to the road and beyond the ‘notifying node’.

MPAR is more efficient compared with the existing routing protocol [34] for similar mobile wireless monitoring applications. The feature of MPAR is that it predicts the position of the mobile sink at the time of meeting the data (**Response**) packet. Thus, the **Response** packet is then directly forwarded toward this predicted position from the source (on-demand cluster head) during the Response Propagation phase. However, in [34], each individual hop is toward the virtual mobile sink instantaneous position calculated by the relaying sensor node. The routing path in [34] is expected to be an arc line theoretically. The comparison of MPAR with the routing algorithm in [34] is shown in Fig. 9. It is desirable that the routing path of the **Response** packet in MPAR is shorter than the referenced routing protocol, which may lead to lower energy consumption and consequently shorter delivery latency.

### 5. QBDCS

Intuitively, there must be an optimal query time, at which sending a **Query** packet is energy efficient and has shorter delivery latency theoretically. To complete a query-based data collection cycle with minimum energy consumption and delivery delay, the QBDCS finds the optimal time to inject the **Query** packet based on the proposed MPAR protocol.
The two objectives of the QBDCS are to minimize the energy consumption and at the same time shorten the delivery latency when the mobile sink is located at (x_{optimal}, 0), where the objective function attains a global minimum, we get the optimal query time which can minimize the overall energy consumption to complete the transmission of Query and corresponding Response packets.

5.2. Optimal query time

5.2.1. A result

Generally, the Query packet is much shorter than the Response packet in length. In this case, the optimal query time is when the mobile sink is located at (x_{optimal}, 0), which the mobile sink can meet with the Response packet at (a, 0), as shown in Fig. 11.

5.2.2. Proof

Let \( L_t/L_q = k \); we have \( E_t/E_q = k \). To simplify the analysis, we assume that \( t_g \approx 0 \), which will not affect the...
final result. The objective function can be written as

$$E_d = \frac{E_{d1}}{d_{hop}} \cdot \min\{d_1 + k \cdot d_2\},$$  \hspace{1cm} (12)$$

where \( E_{d1} / d_{hop} \) can be regarded as a constant. Let \( V_q / V_s = m \); the constraint is

$$d_1 + k \cdot d_2 = m \cdot (\sqrt{d_1^2 - b^2} + \sqrt{d_2^2 - b^2}).$$  \hspace{1cm} (13)$$

Applying the Lagrange multipliers method yields

$$k \cdot \frac{d_1}{\sqrt{d_1^2 - b^2}} = -\frac{d_2}{\sqrt{d_2^2 - b^2}} = 0,$$

$$d_1 + k \cdot d_2 = m \cdot (\sqrt{d_1^2 - b^2} + \sqrt{d_2^2 - b^2}) = 0.$$  \hspace{1cm} (14)$$

Let \( d_1 = b \sec \alpha \) and \( d_2 = b \sec \beta \), where \( \alpha, \beta \in (0, \pi/2) \). Then (14) can be written as

$$\frac{k}{\sin \alpha} - \frac{1}{\sin \beta} = 0,$$

$$\sec \alpha + k \sec \beta - m \tan \alpha - m \tan \beta = 0.$$  \hspace{1cm} (15)$$

which has boundary values when \( \alpha = 0 \) or \( \beta = 0 \). Solving (15), when \( k \neq 1 \), we have

$$-2mk \sin^3 \beta + (m^2 + k^2 + 1) \sin^2 \beta = 0.$$  \hspace{1cm} (16)$$

Thus,

$$\sin \beta = \frac{m^2 + k^2 + 1}{2mk}, \quad \beta \in \left(0, \frac{\pi}{2}\right).$$  \hspace{1cm} (17)$$

Obviously, (17) has no solution. Therefore, the objective function attains the optimal value when \( \alpha = 0 \) or \( \beta = 0 \). When \( k > 1 \), it can be proved that the objective function has optimal value when \( \beta = 0 \), which is equivalent to the mobile sink and the Response packet meeting at \( (a, 0) \).

Based on the above analysis, when the Query packet is shorter than the Response packet in length, the QBDCS finds the optimal query time when the mobile sink is located at \((x_{optimal}, 0)\), where

$$\sqrt{(a - x_{optimal})^2 + b^2} + t_q + b \frac{D}{V_s} - \frac{a - x_{optimal}}{V_q} = 0.$$  \hspace{1cm} (18)$$

FIGURE 11. Theoretical optimal query time based on MPAR.

FIGURE 12. Two cases \((k > 1 \text{ and } k < 1)\) of the optimal query time.

5.2.3. The general result
The theoretical optimal query time depends on \( L_r / L_q = k \), which is the ratio of the length of the Response and Query packets. As shown in Fig. 12, when \( k > 1 \), the optimal query time is when the mobile sink can just meet the Response packet at \((a, 0)\); when \( k < 1 \), the optimal query time is just when the mobile sink is located at \((a, 0)\). When \( k = 1 \), the optimal query time is when the mobile sink is located at such a position where the Response and Query packets can have symmetrical routes (\( \alpha = \beta \)).

5.3. Summary of QBDCS
The QBDCS is summarized in Table 1. It is noteworthy that the \( d_{hop} \) is estimated by the on-demand cluster head in MPAR. However, the estimation of \( d_{hop} \) is conducted by the mobile sink when injecting the Query packet in the QBDCS, which can be estimated after the initial negotiation procedure. A scheme of bypassing holes is adopted when designing the MPAR; in fact, this scheme will be adopted during packets transmission, not limited to the Response Propagation phase.

6. SIMULATION
To evaluate the performance of the proposed QBDCS, we simulate the QBDCS in ns-2 simulator. We first consider regular network topology (without ‘holes’), where sensor nodes are deployed in a 1000 m × 500 m sensing field densely and uniformly. A rectangular grid (50 × 25) is generated, with the side length of each cell being 20 m. Then we test QBDCS for random topologies, which are all generated by the setdest tool in ns-2.

6.1. Simulation settings
The key communication parameters generally applies to IEEE 802.15.4, as listed in Table 2. The velocity of mobile sinks varies from 20 to 30 m/s. For simplicity, we assume that the data aggregation time could be neglected. Sensor nodes communicate at a rate of \( R = 250 \) kbps, the transmission range is set at \( r = 40 \) m. The CSMA/CA mechanism for collision management is adopted. Taking the data-link layer
TABLE 1. Summary of the QBDCS for mobile wireless monitoring applications.

Given $V_s, R, r, L_q, L_r, D_C, (a, b), d_{hop}$,
Calculate the EMP and the optimal query time ($X_{optimal}, 0$)
by (18).

Step 1: Querying at the right time
The mobile sink checks whether it is at the optimal query time. If yes, it injects the Query packet toward the interested area.

Step 2: Routing the Query packet
The Query packet is routed toward the interested area by adopting geographical location-aware routing.

Step 3: Data aggregation
Once the Query packet arrives at the targeted sensing area, the sensor node closest to the center of the interested sensing area elects itself as the cluster head, gathers and aggregates the sensed data, and then generates a Response packet.

Step 4: Routing the Response packet
The Response packet is then routed toward the estimated packet–sink meeting position until it arrives at the ‘notifying node’ or a sensor node close to the road and beyond the ‘notifying node’. The sensor node checks whether the mobile sink has passed or not. If it has passed, the Response packet enters into the Chasing Mode until it catches up with the mobile sink. Otherwise, the sensor node waits for the mobile sink to enter its radio range before delivering the Response packet. During this process, if exceeding the time limit $T_{deadline}$, the Response packet should be discarded.

Step 5: Response packet delivery
The Response packet is delivered to the mobile sink.

TABLE 2. Communication parameters setup.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Unit</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>m</td>
<td>40</td>
</tr>
<tr>
<td>$R$</td>
<td>kB/s</td>
<td>250</td>
</tr>
<tr>
<td>$L_q$</td>
<td>bytes</td>
<td>30</td>
</tr>
<tr>
<td>$L_r$</td>
<td>bytes</td>
<td>120</td>
</tr>
<tr>
<td>$C_T$</td>
<td>s</td>
<td>1</td>
</tr>
<tr>
<td>$D_C$</td>
<td>%</td>
<td>1%</td>
</tr>
</tbody>
</table>

CSMA/CA mechanism into consideration, about 40% overhead is introduced [34]. Thus, the data transmission rate under saturation condition is $R' = 150$ kB/s. Sensor nodes have a cycle time $C_T = 1$ s and $D_C = 1%$. The lengths of the Query and Response packets are 30 bytes and 120 bytes, respectively. Therefore, the estimated delivery velocities $V_q$ ($V_q = \gamma \cdot (r \cdot R' \cdot D_C/L_q)$) and $V_r$ ($V_r = \gamma \cdot (r \cdot R' \cdot D_C/L_r)$) are $250 \cdot \gamma$ m/s and $62.5 \cdot \gamma$ m/s, respectively. Let $D_q$ and $D_r$ denote the maximum delay of the Query and Response packets, respectively; we have $D_q \approx 160$ ms and $D_r \approx 640$ ms.

6.2. Performance metric
6.2.1. End-to-end delivery delay
This metric measures the amount of time from injecting a Query packet by the mobile sink to the corresponding Response packet returning to the mobile sink.

6.2.2. Normalized sending cost
The cost of a transmission consists of the sending cost of the sender, and the receiving cost of one-hop neighbors. Given a fixed number of neighbors, the transmission cost is proportional to the sending cost. Hence, the sending cost metric measures the communication overhead and energy efficiency. We define the normalized energy cost for sending a Response packet as one cost unit. Therefore, the energy cost for sending a Query packet is 0.25 cost unit.

6.2.3. Deviation from the initial EMP
This metric measures the deviation of the final meeting position from the initial EMP.

6.2.4. Packet delivery ratio
This is the ratio of the successful end-to-end transmissions (the mobile sink receives the Response packet) to the total number of tests.

6.3. Evaluating QBDCS
When evaluating the proposed QBDCS, three other schemes are also simulated for comparison as follows: (1) injecting the Query 4s earlier than the QBDCS; (2) injecting the Query 4s later than the QBDCS; and (3) the ‘Naive’ scheme, which the mobile sink sends the Query packet when locating at the nearest point from the interested area, as shown in Fig. 13. All schemes adopt the MPAR protocol to forward the Response packet. The simulation starts from the mobile sink injecting a Query packet.
toward the interested area, and ends when the mobile sink receives the Response packet or when exceeding the limit of the predefined propagation time. The vertical distance from the interested area to the road where the mobile sink is moving along is defined as the query distance (QD).

### 6.3.1. Choosing $\gamma$

The estimation of the packet delivery velocity has a direct effect on the performance of the QBDCS. An overestimate may lead to sending the Query packet later than the optimal query time. An underestimate may result in sending the Query packet earlier than the optimal query time. For both cases, the Response packet has to frequently change the estimated packet–sink meeting position, and consequently deviates from the initial EMP.

We simulate a mobile sink querying the area with $QD = 500$ m. Fig. 14 depicts the change of end-to-end delivery delay by increasing the value of $\gamma$ with different mobile sink velocities. From Fig. 14, it is shown that the optimal value for the modification coefficient $\gamma$ is around 0.95, at which the QBDCS shows the least end-to-end delivery delay.

Figure 15 shows the changes of deviation from the initial EMP as the $\gamma$ changes. The deviation increases as the $\gamma$ decreases when it is smaller than 0.95, where the packet delivery velocity is underestimated. When $\gamma$ is larger than 0.95, where the packet delivery velocity is overestimated, the QBDCS will send the Query packet later than the optimal query time, which also causes the deviation.

It is found that the optimal value of the modification coefficient $\gamma$ depends on the sensor deployment strategy. In our simulations, the optimal value of $\gamma$ is about 0.95, whereas when the side length of each cell in the deployed sensing filed is 30 m and the transmission range is set at 50 m, the optimal value of $\gamma$ is about 0.7.

### 6.3.2. The impact of QD

We vary the QD from 260 to 500 m, comparing the differences when the mobile sink velocity is set at 25 m/s. The modification coefficient $\gamma$ is taken to be 0.95. Obviously, the end-to-end delivery delay and energy consumption depend on the QD. The longer the QD, the larger is the delivery delay and more the energy consumption.

As shown in Fig. 16, with an increase of the QD, the advantage of the QBDCS becomes more significant. Compared with other querying times, the QBDCS outperforms these schemes. This is because the QBDCS can guarantee the least hop counts during the Response Propagation phase, which is the bottleneck of the end-to-end delivery delay and the energy consumption. From Figs 17 and 18, we observe that the QBDCS has the minimum Response packet hop count. Although the number of Query packet hop counts in the QBDCS is more than the ‘Naive’
scheme and the ‘4s later’ scheme, the QBDCS achieves the minimum end-to-end delivery delay.

6.3.3. The impact of mobile sink velocity

We then vary the mobile sink velocity from 20 to 30 m/s with QD = 500 m. The modification coefficient γ is taken as 0.95. Figs 19 and 20 depict the effect of increasing mobile sink velocity on end-to-end delivery delay and normalized sending cost, respectively. We can see that the performance of the QBDCS does not show much dependence upon the mobile sink velocity. This is because of the fact that completing a query-based data collection mainly depends on the hop counts of routing the Response packet. The QBDCS always provides a minimum Response packet hop count during the Response Propagation phase compared with other schemes. For the ‘Naive’ scheme, as the mobile sink velocity increases, the number of Response packet hop counts increases as well. Thus, the performances of the ‘Naive’ scheme decreases rapidly with the increase of the mobile sink velocity.

Next, we look into the impact of the variation of mobile sink velocity for the QBDCS. We set a mean mobile sink velocity and a variation range; the mobile sink moves with the mean velocity, and changes its speed every 2 s to another random value within the range; the results have been averaged over 100 runs. Figs 21–23 show the end-to-end delivery delay, normalized sending cost and deviation from the initial EMP as the variation of the mobile sink velocity increases, respectively.

We can see that even when the variation of mobile sink velocity is significant, for example, the variation ranges in \([-15 \text{ m/s}, 15 \text{ m/s}]\), it has little influence on the end-to-end delivery delay and normalized sending cost performance, with 12% end-to-end delivery delay increase and only a 4% normalized sending cost increase. This is partially because of the sensor node deployment in our simulation. From Fig. 23 we can see that the distance of routing the Response packet...
increases a lot, however, the number of Response packet hop counts does not increase much due to the variation of mobile sink velocity. It is notable that the changing trend of Figs 21 and 22 are not completely the same, the reason being that due to the mobile sink velocity variation, there are some cases of the Response packet waiting for the mobile sink, which only increases the end-to-end delivery delay.

6.3.4. Practical consideration
In this test, we consider more realistic scenarios with irregular network topologies (even with holes) to better investigate the QBDCS. Ten different random topologies are generated by the setdest tool in ns-2. The modification coefficient γ is taken as 1. The value of the T_{deadline} is set at 20 s, which means that the Response packet will be discarded if it cannot be delivered to the mobile sink within 20 s. Fig. 24 reports the average end-to-end delivery delay and the packet delivery ratio of different schemes as the mobile sink velocity increases. It is shown that the ODBCS outperforms the ‘Naive’ scheme even with irregular network topologies. For a random topology, the ODBCS may experience larger end-to-end delivery delay due to the existence of ‘holes’; however, the ODBCS provides a higher packet delivery ratio, shorter delivery latency and, consequently, lower energy consumption on average.

6.4. Discussion
In this paper, we assume that the MAC layer is not optimized (no synchronized duty cycles). Thus, the one-hop delay would be relatively large, in which our proposed scheme has significance. In our future work, we are planning to investigate the duty-cycle scheduling problem in query-based WSNs. Besides, considering the limitation of sensor devices on processing power, the computation load of sensor nodes should be as low as possible. The investigation of efficient data aggregation for the on-demand cluster head is also our future work.
7. CONCLUSION

This paper deals with the problem of query-based data collection for mobile wireless monitoring applications, which is inspired by the recent research on WSNs applied in the area of wireless monitoring. It has two unique characteristics: (1) only partial sensor nodes are involved in a query-based data collection cycle; and (2) the mobile sink position is continuously changing during packet propagation. These two characteristics lead to the fact that existing routing protocols are not suitable for the mobile wireless monitoring applications. To address this problem, we tailored the routing protocol so as to minimize the energy consumption and delivery latency in a complete query-based data collection cycle. We made use of the predictable mobile sink mobility, and adopted the idea of estimating the velocity of packets and the packet–sink meeting position. We presented an MPAR protocol which can route the Response packet toward the predicted meeting position directly. We then proposed an efficient QBDCS based on MPAR. The QBDCS makes the decision on when to inject the Query packet for the mobile sink. Extensive simulations have been carried out for performance evaluation. The experimental results show that the proposed QBDCS achieves the least energy consumption, delivery latency and higher packet delivery ratio to complete a query-based data collection cycle.

FUNDING

This work was supported by National Natural Science Foundation of China (NSFC) under grants No. 60673178 and No. 60873241, and Hi-Tech Research and Development Program of China under grants No. 2008AA01Z217 and No. 2009AA01Z210. The work of L.S. was partially supported by the Lion project supported by Science Foundation Ireland under grant no. SFI08/CE11380 (Lion-2), and partially supported by the Grant-in-Aid for Scientific Research (S)(21220002) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

REFERENCES


