Minimum Latency Scheduling for Multi-Regional Query in Wireless Sensor Networks

Mingyuan Yan Jing (Selena) He Shouling Ji Yingshu Li
Department of Computer Science
Georgia State University
Atlanta, Georgia, 30303, USA
Email: {myan2, jhe9, sji, yli}@cs.gsu.edu

Abstract

Query scheduling as one of the most important technologies used in query processing has been widely studied recently. Unfortunately, to the best of our knowledge, no previous work focuses on the Minimum Latency Multi-Regional Query Scheduling (ML-MRQS) problem. In this paper, we investigate the ML-MRQS problem in Wireless Sensor Networks (WSNs), which aims to generate a scheduling plan with minimum latency for a more practical query model called Multi-Regional Query (MRQ). A MRQ targets at user interested data from multiple regions of a WSN, where each region is a subarea of the WSN. We claim that the ML-MRQS problem is NP-hard. Therefore, we propose a heuristic scheduling algorithm Multi-Regional Query Scheduling Algorithm (MRQSA) to solve this problem. Theoretical analysis shows that the latency of MRQSA is upper bounded by \( 2A + B + C \) for a MRQ with \( m \) query regions \( R_1, R_2, ..., R_m \), where

\[
A = \max_{i=1}^{m} D_i, \\
B = \max_{i=1}^{m} \{(23D_i + 5\Delta + 21)k_i\}, \\
C = \sum_{i=1}^{m} H_i + 5\Delta - m + 17, \\
m \text{ is the number of regions, } \Delta \text{ is the maximum node degree in the WSN, } D_i \text{ is the diameter of } R_i, k_i \text{ is the maximum overlapped degree of sensor nodes in } R_i, H_i \text{ represents the distance of } R_i \text{ with respect to the sink, and } D_{left}^i \text{ is the diameter of the non-overlapped part of } R_i.
\]

Extensive simulations are conducted to verify the performance of our algorithm, which show that MRQSA significantly reduces the query latency when compared with the most recently published multi-query scheduling algorithm.

1. Introduction

Wireless Sensor Networks (WSNs) consist of a large number of battery-powered sensor nodes whose power are almost impossible to be recharged [4]. Early researches of query processing mainly focus on how to improve the query efficiency while minimizing the energy consumption. With the emergence of real-time query processing, e.g. fire monitoring in forests, minimum latency query scheduling becomes another primary concern when processing queries.

Ever since the query scheduling problem was introduced in WSNs, tons of efforts have been made to construct different scheduling strategies for different query applications. Since power limitation is still a severe constraint in WSNs, the authors in [2] and [3] proposed to schedule sensor nodes into sleep mode while no query request is disseminated in periodic data query processing. This method can efficiently conserve energy by reducing the energy consumption of idle listening. The authors in [5] investigated the energy efficient query scheduling problem for multi-query with temporal and spatial overlapping. The real time query scheduling problem is studied in [6]. The scheduling algorithm proposed in [6] aims to get the query results before a predefined deadline. In [14], [16], [17] and [18], the authors focused on the periodic data aggregation scheduling problem in WSNs. The authors in [7] and [8] investigated the tradeoff between query latency and network throughput when designing a query scheduling. Additionally, the authors in [9] addressed the optimal scheduling problem for queries whose arrival time obeys Poisson distribution and the inter-arrival time obeys the hyperexponential distribution, respectively. In [10], the authors proposed to consider the quality of service (response time and/or query lifetime) and the quality of data (the received data to be more evenly distributed) while dealing with query scheduling.

SELECT \( \mathcal{K}_{\text{maxTemperature}} \), \( \mathcal{K}_{\text{minHumidity}} \) FROM \( \mathcal{K} \), \( \mathcal{K}_0 \) WHERE \( \mathcal{K}_{11^\text{th}}\cup12^\text{th}\) floors and \( \mathcal{K}_{12^\text{th}}\cup13^\text{th}\) floors and time=10:00am

Figure 1. An example of a MRQ
The previous researches made great contributions to the development of query scheduling technologies, however, the aforementioned techniques are inefficient when dealing with the query example shown in Figure 1, which is a query targeting at different data (the maximum temperature of the 11th and 12th floors and the minimum humidity of the 12th and 13th floors) from two specified regions of a WSN at a particular time. The reasons are as follows. First, in most existing works, a user disseminates a query by broadcasting the query request to the entire network even though the user interested data is from a particular small region. That wastes a large amount of energy, especially in large scale WSNs. Second, the scheduling strategies that proposed in existing works construct a single fixed scheduling tree consisted of all the sensor nodes. Avoiding collisions on a fixed global scheduling tree leads to more latency. Hence, It is not suitable for scheduling multi-query without correlations and targeting at multiple regions as shown in Figure 1. Finally, the assumptions on disseminating time of queries with temporal and spatial connections considered in existing works are no longer reasonable when dealing with queries without those characteristics.

In order to overcome the aforementioned three drawbacks, we investigate the Minimum Latency Multi-Regional Query Scheduling (ML-MRQS) problem in this paper. A Multi-Regional Query (MRQ) is defined as a query that targets at user interested data from multiple regions (overlapping may exist) of a WSN, where each region is a subarea of the WSN. Furthermore, for a MRQ, we make no limitation in query disseminating time interval or assumptions on temporal or spatial correlations. The traditional multi-query processing can be taken as a special case of the MRQ processing if the query region is the entire network. Therefore, compared with the traditional query model, the MRQ model is more practical and general. Consequently, we investigate the ML-MRQS problem in this paper which aims to get a collision-free scheduling plan for a MRQ with minimum latency. The contributions of this paper can be summarized as follows:

1. In this paper, we investigate the ML-MRQS problem for WSNs. To the best of our knowledge, we are the first one to study this problem in WSNs. We claim that the ML-MRQS problem is NP-Hard. Hence, a heuristic collision-free scheduling algorithm, named Multi-Regional Query Scheduling Algorithm (MRQSA), is proposed. In MRQSA, we construct a CDS-based scheduling tree for each region. All those scheduling trees form a scheduling forest which is served as the data transmission structure during the data transmission procedure. Furthermore, in order to improve the parallelism of MRQSA, we propose to assign sensor nodes who may cause multiple collisions a higher priority. During the scheduling procedure, transmissions on different trees are scheduled concurrently. In order to avoid collision, priority and interference will be considered in the entire forest.

2. We theoretically prove that the latency of MRQSA is upper bounded by \(23A + B + C\) for a MRQ with \(m\) query regions \(R_1, R_2, ..., R_m\), where \(A = \max_{i=1}^{m} D_{ref}^i\), \(B = \max_{i=1}^{m}\{23D_i + 5\Delta + 21\}k_i\), \(C = \sum_{i=1}^{m} H_i + 5\Delta - m + 17\), \(m\) is the number of regions, \(\Delta\) is the maximum node degree in the WSN, \(D_i\) is the diameter of \(R_i\), \(k_i\) is the maximum overlapped degree of sensor nodes in \(R_i\), \(H_i\) represents the distance of \(R_i\) with respect to the sink, and \(D_{ref}^i\) is the diameter of the non-overlapped part of \(R_i\).

3. We also conduct extensive simulations to validate the performance of MRQSA. The simulation results indicate that the proposed MRQSA has a better performance compared with the most recently published multi-query scheduling algorithm C-DCQS [7]. MRQSA reduces latency by 49.3% on average when the number of query regions increases and 50.7% on average when the network density increases compared with C-DCQS.

The rest of this paper is organized as follows: in Section 2, we introduce the terminologies and define the ML-MRQS problem. In Section 3, we present the detail of our scheduling algorithm followed by theoretical analysis. The simulation evaluations are given in Section 4. Finally, we conclude this work in Section 5.

2. Problem Formulation

2.1. Network Model

We consider a connected dense WSN consisting of \(n\) sensor nodes along with one sink (base station) denoted by \(s\). Each node is equipped with a single, half-duplex radio. During a time slot, a node \(a\) can either send or receive data (not both) to or from all directions. The transmission/interference range of any sensor node is defined as a unit disk with radius 1 centered at the node. Hence, we can use an undirected Unit Disk Graph (UDG) \(G = (V, E)\) to represent the topology graph of a WSN, where \(V\) represents the set of all the sensor nodes and the sink, then \(|V| = n + 1\); \(E\) denotes the bidirectional link relations among all the sensor nodes. \(a, b \in V\), a’s one-hop neighbor set is denoted by \(N(a) = \{b ||a - b|| \leq 1, b \in V, a \neq b\}\). An edge \((a, b) \in E\) if and only if \(b \in N(a)\).

We assume the network time is synchronized and slotted. Within a time slot, a data package can be sent from a sender and received by a receiver successfully as long as there is no
interference. Additionally, we assume the global locations of all the sensor nodes are known.

2.2. Multi-Regional Query

Let $U$ represent the entire monitoring area of a WSN. A Region is defined as a consecutive subarea within $U$, i.e., for a region $R_i$, in a WSN, $R_i \subseteq U$. Since the considering WSN is dense, the sensor nodes deployed in $R_i$ is a connected component of $G$. For $\forall a \in V$, $\forall R_i \subseteq U$, $a$ is called a Regional Node (RN) of $R_i$ if and only if $a$ is deployed in $R_i$. Furthermore, $S(R_i)$ is defined as the set of RNs in $R_i$.

\[
\text{SELECT } R_i, d_s, R_i, d_s, \ldots, R_n, d_s \\
\text{FROM } R_i \cup R_i \cup \ldots \cup R_n \\
\text{[WHERE predicates]}
\]

Figure 2. Formal format of MRQ

Definition 2.1. Multi-Regional Query (MRQ). A MRQ is a query targeting at user interested data from multiple regions in a WSN. The formal expression of a MRQ is given in Figure 2 which shows a MRQ with $m$ query regions. SELECT denotes the query results, where $R_i, d_s$ means the user interested data in region $R_i$. FROM shows the multiple regions that the MRQ should be disseminated. The query conditions are specified in WHERE.

When a MRQ is disseminated to regions specified by the FROM sentence, the RNs deployed in $R_i$ only have to collect $R_i, d_s$. An example of a MRQ is shown in Figure 1. The MRQ targets at two regions with $R_1 = \{11^{th} \cup 12^{th}\}$ floors and $R_2 = \{12^{th} \cup 13^{th}\}$ floors. The user interests in obtaining the maximum temperature of $R_1$ and the minimum humidity information of $R_2$. From this example, firstly, we can see that this MRQ targets at data with no correlations from different regions simultaneously. Secondly, the target regions in a MRQ may overlap, such as $R_1 \cap R_2 = \{12^{th}\}$ floor. Thirdly, since the target regions of a MRQ are known, the MRQ can be directly disseminated to $R_1 \cup R_2 = \{11^{th} \cup 12^{th}\} \cup \{12^{th} \cup 13^{th}\} = \{11^{th} \cup 12^{th} \cup 13^{th}\}$ floors.

In this paper, we investigate how to schedule a MRQ, which is dynamically disseminated into multiple regions of a WSN simultaneously. In order to conserve network resource, query results are transmitted to the sink with aggregation. That is, on the routing tree, parent nodes have to wait for all of its children finishing their data transmissions before starting its own transmission. We make no assumption on the correlations among data collected from different regions. Hence, the data can be aggregated during data transmission only if they are user interested data from the same region.

2.3. Problem Definition

Let $R_i$ represent a region. A Regional Query Tree (RQT) $T_i$ is a tree constructed within $R_i$ satisfies: $\forall a \in S(R_i) \Rightarrow a \in T_i$. The root of $T_i$ is denoted by $r_{T_i}$. Considering a MRQ with $m$ query regions $R_1, R_2, \ldots, R_m$, $\forall 1 \leq i \leq m$, $T_i$ denotes the RQT constructed in $R_i$ with $S(R_i)$. The Multi-Regional Query Forest (MRQF) denoted by $F$ of a MRQ is the set of all the RQTs of this MRQ, i.e. $F = \{T_1, T_2, \ldots, T_m\}$. The set of all the sensor nodes in $F$ is denoted by $S(F)$, i.e. $S(F) = S(R_1) \cup S(R_2) \cup \ldots \cup S(R_m)$. The defined MRQF (RQTs) is served as the routing structure in our proposed algorithm.

The Overlapped Degree (OD) of a RN $a$, denoted by $od_a$, is defined as the number of RQTs it belongs to. For any region $R_i$ with $k$ RNs $n_1, n_2, \ldots, n_k$, let $O = \{a | od_a = \min\{od_{n_1}, od_{n_2}, \ldots, od_{n_k}\}\}$, which is the set of RNs in $R_i$ with the smallest OD. $A$ is the set of RNs in $O$ that are closest to the sink (the distance of each RN to the sink is calculated as their hop-distance). The Accessing Node (AN) is defined as the RN in $A$ with the smallest ID.

Given a MRQF $F = \{T_1, T_2, \ldots, T_m\}$, let $I = \bigcup_{i=1}^{m} \{a | a \in T_i \land \exists b \in T_j, i \neq j, a \in N(b), 1 \leq i, j \leq m\} \cup \{a | od_a = \geq 2\}$ represent the set of RNs that have at least one one-hop neighbors on another RQT and RNs whose ODs are no less than 2. $\forall a \in T_i, D(a)$ is defined as $a$’s descendant set which contains the RNs on a subtree of $T_i$ rooted at $a$. The Overlapped Interference Set (OIS) of a MRQF is denoted by $S(OIS) = I \cup D(I)$, where $D(I)$ is the set of descendants of RNs in $I$. The reason of including the descendants of $I$ relies on the transmission constraint of aggregating data within a region during the data transmission.

Figure 3. An example of a MRQF
Taking Figure 3 as an example, it shows a MRQF with two RQTs \( T_1 \) and \( T_2 \) which are constructed within query regions \( R_1 \) and \( R_2 \), respectively. A RN is represented by a black circle with its ID inside. The ANs are filled with black and the RNs in the overlapped interference set \( S'(Ois) \) are filled with gray. From Figure 3, we can see that \( od_3 = 1 \) since it is only on \( T_1 \), while \( od_{16} = 2 \) because it is on both \( T_1 \) and \( T_2 \). RN 0 is the AN of \( T_1 \) since it has the smallest OD and is closest to the sink and the AN of \( T_2 \) is RN 6. The OIS of the MRQF is \( S'(Ois) = T \cup D(T) = \{5, 9, 10, 15, 16, 17, 21, 22, 25, 29\} \cup \{23, 26, 27, 28, 30, 31\} = \{5, 9, 10, 15, 16, 17, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31\}, \) where each RN in \( T \) has at least one one-hop neighbor on another RQT, and the RNs in \( D(T) \) are descendants of RNs in \( T \).

**Interference Model:** In this paper, we use the classical interference model which is widely used in the previous works [2]-[10][14][16]. Given two simultaneously transmissions \( a \rightarrow b \) and \( c \rightarrow d \), where \( a, c \) are senders and \( c, d \) are receivers. Transmission \( c \rightarrow d \) is said to conflict with transmission \( a \rightarrow b \) if and only if \( d \in N'(a) \) or \( b \in N'(c) \). Under the defined interference model, the definition of a Non-Concurrent Set is given below.

**Definition 2.2. Non-Concurrent Set (NCS).** Given a MRQ with \( m \) regions, its corresponding M-RQF is \( F = \{T_1, T_2, ..., T_m\} \). \( \forall i \in T_i \), \( P_i(a) \) is the parent node of \( a \) on \( T_i \). Then, transmission \( a \rightarrow P_i(a) \)'s NCS is defined as \( NCS(a \rightarrow P_i(a)) = N_F(P_i(a)) \cup \{ \cup_{b \in N_F(a) \cap F}(Ch(b)) \cup \{a, P_i(a)\} \cup \{Ch(b)\} \}, \) where \( N_F(P_i(a)) \) is \( P_i(a) \)'s one-hop neighbors in \( F \), \( N_F(a) \) is \( a \)'s one-hop neighbors in \( F \), and \( Ch(b) \) is \( b \)'s children nodes in \( F \).

Specially, due to the fact that RNs may be on multiple RQTs, for a particular time slot, any RN can participate in only one transmission. Hence, for \( a, Ch(a) \) and \( P_i(a) \) are also needed to be included in \( NCS(a \rightarrow P_i(a)) \), which is different from the NCS defined in traditional case based on a single fixed scheduling tree.

Then, we define the Minimum Latency Multi-Regional Query Scheduling problem (ML-MRQS) as follows.

**Definition 2.3. Minimum Latency Multi-Regional Query Scheduling (ML-MRQS).** Given a MRQ targets at \( m \) regions \( R_1, R_2, ..., R_m \), the RQT constructed in \( R_i \) is represented by \( T_i \), \( F = \{T_1, T_2, ..., T_m\} \) is the MRQF. Let \( Sch_{T_i} \) be the set of all child \( \rightarrow \) parent transmissions on \( T_i \), \( Sch_F = Sch_{T_1} \cup Sch_{T_2} \cup ... \cup Sch_{T_m} \), and \( Sch^t_{T_i} \) represent the set of transmissions on \( T_i \) that are scheduled at time \( t_i \). An ML-MRQS is a sequence of transmission sets denoted by \( SCH = \{Sch^t_{T_1}, Sch^t_{T_2},..., Sch^t_{T_m}\} \), \( \{Sch^{t_1}_{T_1}, Sch^{t_2}_{T_1},..., Sch^{t_{t_m}}_{T_1}\}, \) \( ..., \{Sch^{t_1}_{T_1}, Sch^{t_2}_{T_1},..., Sch^{t_{t_m}}_{T_1}\} \) satisfying the following conditions:

(i). \( Sch^t_{T_i} \cap Sch^t_{T_k} = \emptyset, \forall i \neq j, 1 \leq k \leq m, 1 \leq i, j \leq L \).
(ii). \( \forall a \rightarrow P(a) \in Sch^t_{T_i}, \forall b \rightarrow P(b) \in Sch^t_{T_k}, a \notin NCS(b \rightarrow P(b)) \) and \( b \notin NCS(a \rightarrow P(a)) \), where \( \forall k \neq j, 1 \leq k, j \leq m, 1 \leq i \leq L \).
(iii). \( \bigcup_{i=1}^{m} Sch^t_{T_i} = Sch_{T_i} \), where 1 \( \leq k \leq m \).
(iv). \( \bigcup_{i=1}^{m} \bigcup_{k=1}^{m} Sch^t_{T_k} = Sch_F \).
(v). Data are aggregated from \( \bigcup_{k=1}^{m} Sch^t_{T_k} \) to \( Sch_F \) - \( \bigcup_{i=1}^{m} \bigcup_{k=1}^{m} Sch^t_{T_k} \) at time slot \( t_i \), for all \( L = 1, 2, ..., L \). Data are aggregated to \( \bigcup_{k=1}^{m} r_{T_k} \) in \( L \) time slots, \( L \) is the latency of the MRQS.
(vi). \( L \) is minimized.

In Definition 2.3, Constraint (i) specifies that a particular transmission on a RQT can only be scheduled once. Constraint (ii) guarantees that for a particular time, the transmissions scheduled on different RQTs should without collision. Constraint (iii) forces all the transmissions on a RQT getting their scheduling time slots, while Constraint (iv) guarantees that all the transmissions in a MRQF are scheduled. Constraint (v) denotes the transmission set scheduled at time \( t_i \), and it also guarantees the query result of each region is collected to the AN. Constraint (vi) denotes that the latency should be minimized.

We claim that the ML-MRQS problem under the UDG model is NP-hard. That is based on the fact that the minimum latency data aggregation scheduling problem is NP-hard [1] under the UDG model, which can be considered as a special case of ML-MRQS when the target region of a MRQ is the entire network.

### 3. Multi-Regional Query Scheduling

In this section, we propose a scheduling algorithm for ML-MRQS which consists of two phases. First, a MRQF is constructed which is served as the data transmission structure during the data transmission procedure. Subsequently, a collision-free scheduling plan is generated.

#### 3.1. Construction of MRQF

In order to obtain an efficient routing tree for the data transmission of a ML-MRQS, we use the method proposed in [11] to construct a CDS-based RQT in each region. The RQTs constructed in all the regions form a MRQF, which are used as the data transmission structure of the ML-MRQS.

In a WSN represented by \( G = (V, E) \), a Dominating Set (DS) is defined as a subset of sensor nodes in the WSN with the property that for each sensor node in the WSN, it is either in the subset or adjacent to at least one node in the subset. If the induced subgraph of a DS is connected,
we call this DS a Connected Dominating Set (CDS). Sensor nodes in a CDS are called dominators or connectors and others are called dominatees. Ever since the CDS is introduced, it is widely used in many applications as a virtual backbone for efficient routing [15][19].

The construction of $T_i$ in $\mathcal{R}_i$ can be briefly described as four steps:

1. **Step 1:** Find the AN $a$ of $\mathcal{R}_i$.
2. **Step 2:** Build a Breadth First Searching (BFS) tree rooted at $a$ with all the RNs in $\mathcal{R}_i$. Along with the BFS tree constructing procedure, obtain a Maximal Independent Set (MIS) $\mathcal{M}_i$. The RNs in $\mathcal{M}_i$ colored black are dominators.
3. **Step 3:** Find a set of sensor nodes $\mathcal{C}_i$ to connect the nodes in $\mathcal{M}_i$ to form a CDS. The RNs in $\mathcal{C}_i$ colored gray are connectors and satisfy that, for any RN in $\mathcal{C}_i$, its parent node and children nodes are dominators.
4. **Step 4:** Color RNs in $S(\mathcal{R}_i)\setminus \mathcal{M}_i \cup \mathcal{C}_i$, which are dominates, as white. And pick each dominatee $b$ a dominator from $\mathcal{N}(b) \cap \mathcal{M}_i$ as its parent. This step allocates each dominatee to a dominator, such that every dominatee can transmit data to its allocated dominator during the data transmission process.

Due to the existence of overlapped regions, RQTs in a MRQF may overlap with each other (like Figure 3 shows), which is different from a traditional forest. Therefore, the ML-MRQS problem is much more challenging than the traditional scheduling problem because RNs in $S(Ois)$ may cause collisions more than once.

### 3.2. MRQSA

#### 3.2.1 Scheduling Initialization

At the beginning of initialization, the RNs in $\mathcal{F}$ are divided into two parts $S(Ois)$ and $\mathcal{F}^{\text{conf}}$, where $S(Ois)$ is the OIS of a MRQF, and $\mathcal{F}^{\text{conf}} = \mathcal{F}\setminus S(Ois)$. Let $T_i^{\text{conf}} = T_i \setminus S(Ois)$, then $\mathcal{F}^{\text{conf}}$ can be represented by $\{T_i^{conf}, T_2^{conf}, \ldots, T_m^{conf}\}$ satisfying $\forall i \neq j$, $T_i^{conf} \cap T_j^{conf} = \emptyset$. The behind-the-scene meaning of separating $S(Ois)$ from $\mathcal{F}^{\text{conf}}$ is to improve the parallelism of MRQS. The RNs in $S(Ois)$ may influence the data transmissions of multiple RQTs because their interference range may overlap with multiple regions. Therefore, the RNs in $S(Ois)$ are given a higher priority in MRQSA. It follows that the RNs in $S(Ois)$ may finish their scheduling first. After that, the scheduling among the $T_i^{conf}$s can be paralleled without collision.

The $S(Ois)$ can be specified by checking all the RNs in the MRQF: from $T_1$ to $T_m$. If one RN on $T_i$ has a one-hop neighbor who is on $T_j$ ($i \neq j$) or the OD of the RN is no less than 2, mark it as a RN in $S(Ois)$, i.e., if $\forall a \in T_i, \exists b \in T_j, a \in \mathcal{N}(b), i \neq j$, or $\alpha(b) \geq 2$, put $a$ in $S(Ois)$.

Subsequently, find the descendants of all the elements in $S(Ois)$.

#### 3.2.2 Scheduling Algorithm

After the initialization phase, all information needed for scheduling are prepared. For simplicity, let $Sch^l$ represent the schedule plan in the $l$-th iteration, which is a set of transmissions are scheduled at the $l$-th time slot, $l = 1, 2, ..., L$. $NCS(Sch^l)$ represents the NCS of the transmissions in $Sch^l$. Let $T_i^{Ois} = T_i \setminus S(Ois)$.

A transmission $a \rightarrow b$ on $T_i$ is said ready to be scheduled if RN $a$ is a leaf node or $a$’s children nodes on $T_i$ have already obtained their scheduling time slots and the priority of $a$ is the highest compared with other scheduling candidates on $T_i$. The priority is given to a RN in the order of “RN in the OIS”, “color is white”, “depth is larger”, “ID is smaller”.

We propose a scheduling algorithm, called Multi-Regional Query Scheduling Algorithm (MRQSA), as shown in Algorithm 1.

**Algorithm 1: MRQSA**

**input**: MRQF  
**output**: the schedule plan $S = \{Sch^1, Sch^2, ..., Sch^m\}$, latency $L$

1. $S \leftarrow \emptyset$;  
2. $l \leftarrow 0$;  
3. while $\mathcal{F}^{\text{conf}} \neq \emptyset$ or $S(Ois) \neq \emptyset$ do  
   4. $l \leftarrow l+1$;  
   5. $Sch^l \leftarrow \emptyset$;  
   6. $NCS(Sch^l) \leftarrow \emptyset$;  
   7. Schedule $S(Ois)$ at the $l$-th time slot, find the conflict-free transmissions in $S(Ois)$ and add these transmissions to $Sch^l$ followed by updating $NCS(Sch^l)$ (as shown in Algorithm 2);  
   8. Schedule $\mathcal{F}^{\text{conf}}$ at the $l$-th time slot, find the conflict-free transmissions in $\mathcal{F}^{\text{conf}}$ and add these transmissions to $Sch^l$ (as shown in Algorithm 2);  
   9. Add $Sch^l$ to $S$;  
10. return $S$, $L = l$;

Algorithm 1 shows that MRQSA takes the MRQF as an input. It runs iteratively from line 3 to line 9 and finally output the scheduling plan $S$ and the latency $L$ of $S$. During the $l$-th iteration ($l = 1, 2, ..., L$), the first step is to initialize $Sch^l$ and $NCS(Sch^l)$ to be empty sets (line 5-6). Subsequently, schedule the transmissions whose senders are RNs in $S(Ois)$ and can be scheduled in the $l$-th iteration as line 7 shows. Finally, the transmissions whose senders are RNs in $\mathcal{F}^{\text{conf}}$ can be scheduled in the $l$-th iteration are found as shown in line 8. MRQSA generates a scheduling plan iteratively until all the $child \rightarrow parent$ transmissions in each
RQT are scheduled. The method of scheduling \( S(O_{is}) \) or \( F_{left}^i \) is presented in Algorithm 2.

### Algorithm 2: Schedule \( S(O_{is}) \) or \( F_{left}^i \) in the l-th slot

- **input**: \( O_S \) (Objective Set)-\( S(O_{is}) \) or \( F_{left}^i \), \( NC(Sch^l) \)
- **output**: the schedule plan \( Sch^l \) for time slot \( l \)

1. \( flag = 1; \)
2. **while** \( flag \neq 0 \) **do**
   3. \( flag \leftarrow 0; \)
   4. **for** \( i = 1 \rightarrow m \) **do**
      5. *if* \( a \in O_S \) and a \( \rightarrow P_i(a) \) is ready and
         \( a \notin NC(Sch^l) \) **then**
         6. \( flag \leftarrow 1; \)
         7. \( Sch^l \leftarrow Sch^l \cup a \rightarrow P_i(a); \)
         8. \( NC(Sch^l) \leftarrow NC(Sch^l) \cup NC(a \rightarrow P_i(a)); \)
         9. \( oda \leftarrow -1; \)
      10. *if* \( oda = 0 \) **then**
          11. \( O_S \leftarrow O_S \backslash \{a\}; \)
   12. **return** \( Sch^l, NC(Sch^l); \)

Algorithm 2 takes the \( O_S \) (Objective Set) as an input, where \( O_S = S(O_{is}) \) when scheduling \( S(O_{is}) \) and \( O_S = F_{left}^i \) when scheduling \( F_{left}^i \), respectively. The output of Algorithm 2 is the transmissions whose senders are in the considering \( O_S \) that can be scheduled in the \( l \)-th iteration. From \( T_l \) to \( T_m \) (line 4), Algorithm 2 runs iteratively to find a ready transmission \( a \rightarrow P_i(a) \) with the highest priority and \( a \notin NC(Sch^l) \) as shown in line 5. If such a transmission is found, add it to \( Sch^l \) (line 7). Subsequently, the \( NC(Sch^l) \) is updated to \( NC(Sch^l) \cup NC(a \rightarrow P_i(a)) \) (line 8). If all the transmissions in the MRQF that \( a \) is involved in are scheduled, \( a \) should be removed from the \( O_S \) (line 10-11). The procedure will continue until no more ready transmission that can be scheduled without collision is found in this iteration.

After obtaining the scheduling plan, every transmission can be scheduled in its assigned time slot.

Particularly, the priority strategy guarantees the new added transmissions whose senders are dominators or connectors won’t conflict with the scheduled transmission whose senders are dominators, and it also guarantees the scheduling of transmission whose sender has a smaller depth won’t conflict with a scheduled transmission whose sender has a bigger depth.

### 3.3. Performance Analysis

In this subsection, we analyze the latency performance of MRQSA. Compared with traditional query scheduling methods, a MRQ can be directly disseminated to the query regions instead of the entire network. Intuitively, this procedure will reduce latency and energy consumption. Therefore, we only focus on the latency performance of MRQSA which is consisted of the following three parts: the latency of scheduling \( F_{left}^i \), the latency of scheduling \( S(O_{is}) \), and the latency of scheduling ANs to transmit the final aggregated values to the sink.

For convenience, The **diameter** of a region is defined as the maximum distance between any two RNs in this region. Let \( D_{left}^i \) represent the diameter of the non-overlapped part of region \( R_i (T^i_{left}); D_i \) be the diameter of \( R_i \), \( k_i \) be the maximum overlapped degree of \( T_{Ois}^i \) which is defined as the maximum OD of the RNs in \( T_{Ois}^i \), and \( H_i \) represent the distance of \( R_i \) with respect to the sink, which is the hop-distance from the AN to \( T_i \) to the sink. The following analysis is based on some conclusions given in [12], [14] and [13].

**Lemma 1.** The latency of scheduling \( F_{left}^i \) is no more than \( 23 \max_{i=1}^m D_{left}^i + 5\Delta + 17 \).

**Proof sketch:** Based on the aforementioned illustration, since any two \( T^i_{left} \) and \( T^j_{left} \) (\( i \neq j \)) in \( F_{left}^i \) are pairwise disjoint, the scheduling of \( T^i_{left} \) is independent with others. As a consequence, the latency of scheduling \( F_{left}^i \) is actually depends on the latency of \( T^i_{left} \) whose latency is the largest. Additionally, since the RNs in \( T_{Ois}^i \) are leaves or subtrees, each \( T^i_{left} \) is still a CDS-based tree based on the construction procedure of a MRQF in Section 2.3. According to MRQSA, the scheduling of each \( T^i_{left} \) consists of two phases. Firstly, the scheduling from dominators to dominators, which is at most \( 5\Delta + 20 \). Secondly, the scheduling between dominators and connectors, which is upper bounded by \( 23 \max_{i=1}^m D_{left}^i + 5\Delta + 17 \). Hence, the latency of scheduling \( F_{left}^i \) is upper bounded by \( 23 \max_{i=1}^m D_{left}^i + 5\Delta + 17 \). □

**Lemma 2.** The latency of scheduling \( S(O_{is}) \) is no more than \( \max_{i=1}^m \{23D_{i} + 5\Delta + 21k_i\} \).

**Proof sketch:** According to the definition of \( S(O_{is}) \) (Section 2.3), \( S(O_{is}) \) is a set of RNs whose interference range overlapped with other RN (or RNs) in other region (or regions) and their descendants. \( S(O_{is}) \) can be represented by \( \{T_{1Ois}^i, T_{2Ois}^i, \ldots, T_{mOis}^i\} \). Let \( \{k_1, k_2, \ldots, k_m\} \) represent the maximum OD of each \( T_{Ois}^i \). For each \( T_{Ois}^i \), let \( l_{omin} \) represent the smallest depth of the RNs in \( T_{Ois}^i \) and \( l_{omax} \) represent the largest depth of the RNs in \( T_{Ois}^i \). Since \( T_{Ois}^i \) may not be consecutive, we extend \( T_{Ois}^i \) to a subset consisted of the RNs in \( T_i \) from depth \( l_{omin} \) to depth \( l_{omax} \). Evidently, the latency of \( T_{Ois}^i \) is no more than the latency of extended \( T_{Ois}^i \). It follows that \( l_{omax} - l_{omin} \leq 2D_i \).
We first consider the scheduling of the extended \( T'_i^{Q,i} \) without considering the overlapping. The scheduling of \( T'_i^{Q,i} \) contains two parts. The time needed for the first part that scheduling from dominatees to dominators is no more than \( 5\Delta + 20 \). The second part is the scheduling between dominators and connectors is upper bounded by \( 23D_i + 1 \).

Therefore, the latency of scheduling \( S(O_1) \) when the \( k_i \) overlappings are considered is upper bounded by \( l_{S(O_1)} = \max_{i=1}^{m}\{(23D_i + 5\Delta + 21)k_i\} \).

**Lemma 3.** The latency of scheduling the ANs is no more than \( \sum_{i=1}^{m}(H_i - 1) \).

**Proof:** Since there are only \( m \) RQ instances, the number of ANs is at most \( m \). In the worst case, no two ANs can be scheduled concurrently. Hence, the latency is at most \( l_{ANs} = \sum_{i=1}^{m}(H_i - 1) \).

However, the time for ANs finishing their data collecting may vary. The actual latency may much less than the bound shown in Lemma 4.

Then we can conclude Theorem 1. The proof of Theorem 1 is omitted due to space limitation.

**Theorem 1.** The upper bound of the latency of MRQSA is no more than \( 23A + B + C \), where \( A = \max_{i=1}^{m}D'^{ef}_{i} \), \( B = \max_{i=1}^{m}\{(23D_i + 5\Delta + 21)k_i\} \), and \( C = \sum_{i=1}^{m}H_i + 5\Delta - m + 17 \).

4. Performance Evaluation

In this section, we compare the performance of MRQSA with the centralized version of DCQS denoted by C-DCQS proposed in [7]. C-DCQS is the most recently published algorithm that solves the multi-query scheduling problem. We compare the two algorithms in terms of latency under different conditions.

4.1. Simulation Environment

In our simulations, we deploy sensor nodes in a monitoring area of \( 1000m \times 1000m \), where all the sensor nodes have the same interference/transmission range of \( 50m \). The sink is deployed in the upper left corner. Moreover, the regions in a MRQ are set as squares. During the result collecting procedure of a MRQ, the results from a particular region will be aggregated and the results from different regions cannot be aggregated. Besides, the latency is defined as the time interval between the first RN beginning to transmit data and the last packet of the results being received by the sink. The sensor nodes involved in MRQSA contains RNs and the sensor nodes on the shortest paths from ANs to the sink. To be fair, we make the following improvements to C-DCQS. Firstly, we use the same algorithm in [11] to construct a CDS-based tree as the routing tree for C-DCQS. Secondly, the sensor nodes involved in a MRQ under C-DCQS is defined as a subtree rooted at the sink which has the least number of sensor nodes but contains all the RNs. By this way, the number of sensor nodes involved in a MRQ are minimized under C-DCQS. The results shown in the following are the averaged values by executing the same procedure for 100 times.

4.2 Simulation Results

We examine the latency of C-DCQS and MRQSA when the number of query regions of a MRQ varies as shown in Figure 4 (a). We fix each query region as a square of \( 600m \times 600m \) and the regions are randomly generated in the WSN. The number of query regions increases from 2 to 10 by 1. From Figure 4 (a), we can see that, with the size of a MRQ increases (the size of a MRQ means the number of regions in this MRQ), the latency of both C-DCQS and MRQSA increase. This is because more data transmissions are needed when collecting the results from new added regions to the sink. It follows that more time is needed to guarantee collision-free scheduling. MRQSA has a better performance is due to that MRQSA tries to maximum the parallelism of scheduling by concurrently schedule transmissions of different regions. While C-DCQS calculates the minimum time interval for each query in different regions and schedules transmissions in different regions sequentially according to the single fixed scheduling tree. Thus, more time is required for C-DCQS. Under this condition, MRQSA reduces latency by 49.3% on average when compared with C-DCQS.

We study the influence of network density on the latency of C-DCQS and MRQSA as shown in Figure 4 (b). In this case, the query region is fixed to be a square of \( 600m \times 600m \) and the number of query regions is set to be 5. We increase the number of sensor nodes in the WSN from 600 to 2000 by 200 each time. Figure 4 (b) shows that the latency of both C-DCQS and MRQSA increase with the increasing of node density. This is because the number of RNs in each query region increases in a more dense network. Therefore, more time is required to finish the scheduling. However, the increasing of MRQSA is more stable than C-DCQS. Especially, when the number of the sensor nodes increases from 1400 to 2000, the latency of C-DCQS increases shapely. This is because under C-DCQS, with the increasing of the network density, the intermediate nodes in a fixed single routing tree have to wait more time for collecting results from its children. Hence, the minimum interval time required to avoid collisions between the scheduling of different regions increases. MRQSA reduces latency by
50.7% on average when compared with C-DCQS under different network densities.

5. Conclusion

In this paper, we investigate the ML-MRQS problem in WSNs. We first introduce the concept of a MRQ, and then shows the challenge and importance of studying the ML-MRQS problem. We claim that the ML-MRQS problem is NP-Hard. Hence, a heuristic scheduling algorithm, called MRQSA, is proposed. The theoretical analysis shows that MRQSA has a latency bounded by $23A + B + C$, where $A = \max_{i=1}^m D_{i}^{k_i}$, $B = \max_{i=1}^m \{(23D_i + 5\Delta + 21)k_i\}$, $C = \sum_{i=1}^m H_i + 5\Delta - m + 17$, $m$ is the number of regions, $\Delta$ is the maximum node degree in the WSN, $D_i$ is the diameter of $R_i$, $k_i$ is the maximum overlapped degree of sensor nodes in $R_i$, $H_i$ represents the distance of $R_i$ with respect to the sink, and $D_{i}^{k_i}$ is the diameter of the non-overlapped part of $R_i$. The simulations shows that MRQSA reduces latency by 49.3% on average when the number of query regions increases and 50.7% on average when the network density increases compared with C-DCQS.

References