Distributed Real-time Data Aggregation Scheduling in Duty-Cycled Multi-hop Sensor Networks

Xiaohua Xu[1], Yi Zhao[2], Dongfang Zhao[3], Lei Yang[4], and Spiridon Bakiras[5]

[1] Department of Computer Science, Kennesaw State University, USA
[2] Department of Computer Science and Technology, Tsinghua University, China
[3] Department of Computer Science & Engineering, University of Nevada, Reno, USA
[4] School of Software Engineering, South China University of Technology, China
[5] College of Science and Engineering, Hamad Bin Khalifa University, Qatar

Abstract. Wireless sensor network (WSN) systems often need to support real time periodic queries of physical environments. In this work, we focus on periodic queries with sufficiently long time horizon in duty-cycled sensor networks. For each periodic query issued by a control center in a WSN, after the source sensors produced the sensory data, the data are to be sent to the sink via multi-hop data aggregation timely in a periodic fashion. To this end, we propose efficient and effective data aggregation algorithms subject to quality of service constraints such as deadline requirements and interference constraints. We decompose these into three sequential operations: (1) aggregation tree construction (2) node and link-level scheduling and (3) packet scheduling. Inspired by the scheduling algorithms, we identify both sufficient conditions and necessary conditions for scheduling multiple queries. The schedulability analysis under various interference models demonstrate that the proposed algorithms achieve an approximate proportion of the maximum possible load.

Keywords: Real time scheduling · Duty cycle · Data aggregation · Interference.

1 Introduction

In numerous applications of wireless sensor networks (WSNs), we often need to support queries (*e.g.* habitat monitoring, structural health monitoring, queries for assessment of potential damages for earthquakes) formed in a Structured Query Language (SQL). After a user queries regarding a data report, the sensors cooperate to generate a convincing response with the help of in-network aggregation. Data aggregation allows data compressing where the data from different source nodes may be correlated. Thus, data aggregation is recommended for answering queries via using an aggregation function, *e.g.*, $\max$, $\min$, average, associated with each query. As an example, when computing the summation of all data in the network, instead of transmitting all raw data to the sink, we only need to transmit the sum result from corresponding children nodes to the sink. Therefore, this feature of in-network data processing potentialy allows energy efficient information delivery, compared to the raw data collection. This is because an outgoing packet size possibily becomes smaller during the data aggregation process.

In this work, we concentrate on satisfying a set of aggregation queries (*i.e.*, queries for monitoring light, temperature, acoustic and ammonia). Given a sink node and a collection of sensor nodes, assume that a control center releases multiple queries. Some query may ask for average ammonia concentration in certain wastewater processing tank and some query may ask for the temprature data in certain area. Each query has a period and a release time and the data from all source nodes are expected to be aggregated to the sink node for each period. Each query also has an end-to-end latency requirement for getting the answer, thus there is an expected deferred deadline for each instance of the query. Given a query set and a wireless interference model, the objective is to design an efficient aggregation tree and an interference-aware schedule of node, link, and packet-level activities for each query.

We study the problem in a duty-cycled scenario. Duty-cycled is originally proposed to save energy. Nowadays, most WSNs are duty-cycled. Under the uncoordinated duty-cycled model, we assume that the time horizon is divided equally into time-slots. In a single time-slot, a node is allowed to transmit data at any time-slot while the node is only able to receive data at some pre-defined time-slot(s) of every period.

Related Work: Job scheduling has been well studied in the literature for both single node case and multi-nodes case [13, 14]. Considering a group communication pattern of the job scheduling where multiple nodes need to cooperate together to finish a job or task, Chipara *et al.* [5] studied the real time query scheduling in wireless network by assuming that the routing tree is pre-given. Xu *et al.* [21] considered the real time data aggregation scheduling with a bounded end-to-end delay performance. However, the proposed scheduling method is centralized and the method does not consider the duty-cycled constraints. Later on, [19, 22] studied the real time data collection and multicast scheduling respectively. On the other hand, one-shot data aggregation scheduling has received a lot of research interests such as [2, 3, 9, 12, 16]. However, only a few [4, 8, 10, 11, 17, 18, 23, 24] studied fast data aggregation scheduling in duty-cycled scenario.

Our main contributions in this paper are on the efficient duty-cycled scheduling algorithms and schedulability test for periodic data aggregation queries. We design routing and scheduling algorithms for data aggregation queries. Inspired by the proposed algorithms, we present sufficient conditions for scheduling multiple queries in a duty-cycled sensor network. We also identify necessary conditions. The schedulability analysis under various interference models demonstrate that the proposed algorithms achieve an approximate proportion of the maximum possible load.

We organize the remaining of the paper as follows. Section 2 presents the system model and the questions to be studied. Section 3 presents scheduling algorithms for data aggregation queries under variant interference models. Section 4 presents schedulability results queries. Section 5 presents the conclusion and the future work.

2 System Model

Let $G = (V, E, v_s)$ be an aggregation graph that models a WSN where V consists of all nodes, E consists of all communication links, and the node $v_s \in V$ is the distinguished sink node. Suppose all nodes have a uniform communication radius. There is a communication link between two nodes if and only if (iff) their distance is at most the communication radius.

We assume a duty-cycling scenario. Each node i has a period $\mathbf{P}$, and an active time-slot in a period of $\mathbf{P}$ consecutive time-slots. In a duty-cycling network, for any node v, suppose its active time-slot is $k(0 \leq k < \mathbf{P})$, then the node v is active and ready for data reception at time t iff $t \equiv k \mod \mathbf{P}$. We assume that a node u can send a data packet to node v at time t, if and only if $\overrightarrow{uv} \in G$ and the receiver node v is active at time-slot t. In this case, node u should already have the data ready before time t and has been waiting for node v to be active.

Considering that multiple link transmissions may occur simultaneously and possibly cause interference, we address extensively several commonly used interference models.

Protocol Interference Model (PrIM) [7] : Each node has a transmission range of one and an *interference range* ρ. A node u can transmit to another node v iff $\|uv\| \leq 1$ and the distance between v and any other sender node is greater than ρ.

RTS/CTS Model [1] : A node u can transmit to another node v iff $\|uv\| \leq 1$ and both nodes u and v are not interfered. Here a node is interfered if the distance between the node and any other sender node or receiver node is at most one.

Physical Interference Model (PhyIM) [25, 26] : A receiver node v can receive the data from a sender u iff the signal to interference plus noise ratio is above a threshold value β, *i.e.*,

$$\mathit{SINR} = \frac{P \cdot \|uv\|^{-\kappa}}{N_0 + \sum_{w \in I} P \cdot \|wv\|^{-\kappa}} \geq \beta.$$

Here P is node u's transmission power and we assume that all nodes have the same power. $\kappa > 2$ is the path loss exponent, $\|wv\|$ is the distance between w and v, $N_0 > 0$ is the background noise, and I is the set of concurrent transmitting nodes.

Query Model: Assume source nodes generate data reports periodically for some applications (*e.g.* temperature monitoring, assessment of potential damages for earthquakes). In a connected network $G = (V, E)$ with $v_s \in V$ as the sink node, the control center issues a query set $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N\}$. For each query $\mathbf{q}_i$, let $\mathcal{S}_i \subseteq V$ be a collection of source nodes that have data to report. The size of a data unit for a source node $v \in \mathcal{S}_i$ is ℓ_i and χ_i is the time needed to transmit ℓ_i data over a communicaton link. The data need to be convergecasted to the distinguished sink v_s to answer query $\mathbf{q}_i$ periodically.

The period of each query $\mathbf{q}_i \in \mathcal{Q}$ is $\mathbf{p}_i$. Each query $\mathbf{q}_i$ has a release time $\mathbf{a}_i$, thus, the release time of the query's t-th instance is $\mathbf{a}_i + (t-1) \cdot \mathbf{p}_i$. Each query also has an end-to-end latency requirement $\mathbf{d}_i$ for getting the answer, thus the expected time for the data from all source nodes to be aggregated is $\mathbf{a}_i + (t-1) \cdot \mathbf{p}_i + \mathbf{d}_i$ for the t-th instance. Note that due to the network delay and duty-cycled constraints, the latency requirement $\mathbf{d}_i$ is usually in the order of $O(R \cdot \mathbf{P})$ for each query. Here R is the radius of the graph G.

Given a query set $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N\}$ for data aggregation, we mainly address two questions. The first question is to design periodic data aggregation scheduling to answer queries. The second question is to determine whether all queries can be satisfied or not. The sufficient condition and necessary condition are used to test whether a given query set for data aggregation is schedulable. A query set is schedulable *iff* they can be answered.

3 Distributed Duty-Cycled Aggregation Query Scheduling

We assume for each query, the data aggregation routing tree that is used can be varied, *i.e.*, for different queries or different instants of a single query, different routing trees can be used. Under this general assumption, we develop effective routing algorithms, node, link, and packet scheduling to avoid interference, and answer all queries.

3.1 Overall Approach

The proposed scheduling of queries for aggregation consists of four phases:

Phase I: For each query, construct a routing tree in the network.

Given a network with the communication graph G, we first build a Connected Dominating Set (CDS) based spanning tree $\mathbf{T}_{CDS}$. The construction of a CDS is in [6]. Then using a *pruning* method, we can derive a data aggregation routing tree $\mathbf{T}_i$ for each query $\mathbf{q}_i \in \mathcal{Q}$. Note that any data aggregation routing tree $\mathbf{T}_i$ is a Steiner tree connecting $\mathcal{S}_i \cup \{v_s\}$ with node $v_s \in V$ as the sink node. In addition, $\mathbf{T}_i$ is a subtree of $\mathbf{T}_{CDS}$.

Phase II: Identify the load of each node in real time which specifies the data packets to transmit for each node.

Given a data aggregation routing tree $\mathbf{T}_i$, for each node u not in dominating set, its original data first are aggregated to a dominator (a neighbor in dominating set) periodically. After that, all data in the G_{CDS} are routed to the sink periodically. Using this routing method, for each query $\mathbf{q}_i \in \mathcal{Q}$, first every leaf node (must be a source node) of the tree $\mathbf{T}_i$ adds packets to its load for each period, then every non-leaf node u in the tree $\mathbf{T}_i$ adds *one* packet to its load upon receiving all packets from its children in $\mathbf{T}_i$ for the query $\mathbf{q}_i$. Note that node u may receive several packets but generates only one packet by computing an aggregation function on data received possibly with its own data. After processing for all queries, we get a load of each node.

Phase III: Allocate transmission time to cells and nodes in proportion to their load.

For a query set $\mathcal{Q}$ and a set of data aggregation routing trees, under the duty-cycled model, a node u's load is defined as $\mathcal{L}_{G,\mathcal{Q}}(u) = \sum_{u \in \mathbf{T}_i} \frac{\chi_i}{\mathbf{p}_i} \cdot \mathbf{P}$.

We partition the plane into cells. We define the load of a cell g as the summation of the loads of all nodes in the cell, *i.e.*, $\mathcal{L}_{G,\mathcal{Q}}(g_{v,h}) = \sum_{u \in V(g)} \mathcal{L}_{G,\mathcal{Q}}(u)$. Here $V(g) \subseteq V$ consists of all nodes lying in cell g. Thus, the load also accounts for routing data.

Based on the definition of load, a cell or node with more packets (thus more load) need to be allocated with more time to transmit.

Phase IV: Prioritize data packets to transmit when it is a node u's allocated time. We use *rate monotonic (RM)* [14] or *earliest-deadline first (EDF)* method in this phase. Note that both methods are effective to ensure that every packet can catch its deadline.

3.2 Constructing CDS-based Routing Trees

The construction for data aggregation routing trees relies on selecting a CDS G_{CDS} of G first. For every node not in the CDS, we connect it to its neighboring dominator. For a query, the corresponding source nodes may be only a subset of the node set. Thus, for query $\mathbf{q}_i \in \mathcal{Q}$, we prune every node $u \in V$ and the corresponding link $\overrightarrow{uv}$ in $\mathbf{T}_{CDS}$ that does not have any source node in the subtree of $\mathbf{T}_{CDS}$ rooted at u. Thus we can get a data aggregation routing tree $\mathbf{T}_i$ for $\mathbf{q}_i$. Here CDS's sparsity property can ensure that the size of CDS in a cell is bounded by a constant times the area of the cell.

For physical interference model, let $\mathbf{r} = \sqrt[\kappa]{\frac{P}{N_0\beta}}$ be the maximum transmission radius. If a communication link's length is approaching $\mathbf{r}$ in practice, the SINR of the link cannnot exceed the threshold with high probability (*w.h.p.*), then the transmission probably fails. Therefore, a link of length close to $\mathbf{r}$ is prevented from transmissions in this work. Given a parameter δ, we only focus on links of length at most $\delta\mathbf{r}$ in the network as in [20]. We can obtain a parameterized reduced graph, denoted as $G(V, \delta\mathbf{r})$. If $G(V, \delta\mathbf{r})$ is connected, we can perform data transmissions in this subgraph instead. Therefore, under physical interference model, we construct data aggregation trees in a parameterized graph $G(V, \delta\mathbf{r})$.

3.3 Identifying Load of Each Node According to Routing Trees

Before constructing a load of a node, we need to determine which queries the node is involved with. If a node participates in a query $\mathbf{q}_i$, the node adds a packet (either original packet or aggregated one in the corresponding data aggregation routing tree $\mathbf{T}_i$) periodically for $\mathbf{q}_i$ to its load. We store the load of each node in the node's buffer. The details are shown in Algorithm 1. Here, we use $[N]$ to denote $\{1, 2, \cdots, N\}$.

3.4 Allocate Time to Node in Proportion to Load

After we identify loads, each node may store some packets in its load. The next phase is to allocate time to each node in proportion to its load.

We employ a cell partition and coloring to ensure that only nodes far apart could possibly be allocated the same time to transmit. We use vertical lines $a_v : x = v \cdot \mathbf{l}(\mathcal{M})$ where $v \in \mathbb{Z}$ and horizontal lines $b_h : y = h \cdot \mathbf{l}(\mathcal{M})$ where $h \in \mathbb{Z}$ to partition the plane into cells. Here $\mathbb{Z}$ is the set of all integers. Under the protocol interference model, we set $\mathbf{l}(\mathcal{M}) = \rho + 1$; Under the CTS/RTS model, we set $\mathbf{l}(\mathcal{M}) = 3$. The above values of $\mathbf{l}(\mathcal{M})$ can ensure that any two senders of a mutual distance $\mathbf{l}(\mathcal{M})$ do not cause any interference. Under the physical interference model, we set the cell length $\mathbf{l}(\mathcal{M}) = \mathbf{r}$.

After cell partition, we allocate time to cells. We color all cells such that every neighboring cells of the same color are separated apart by exactly $\sqrt{c_2(\mathcal{M})} - 1$ cells. Thus, the number of cell colors used is $c_2(\mathcal{M})$. The value of $c_2(\mathcal{M})$ is given in Lemma 5. After cell coloring, we allocate time to each cell in proportion to its load. Due to the duty-cycled constraints, we set the time allocated to a cell as the load of the cell multiplied by the cycling period.

Algorithm 1: Identifying Load of Each Node According to Routing Trees

```
Input : A network $G = (V, E, v_s)$, a set of queries $\mathcal{Q} = \{\mathbf{q}_i : i \in [N]\}$, a set of routing
        trees $\{\mathbf{T}_i : i \in [N]\}$.
Output: The load of each node: a set of packets for each node.
1  $t \longleftarrow 1$;
2  while TRUE do
3      for each query $\boldsymbol{q}_i \in \mathcal{Q}$ do
4          for each node $u \in \boldsymbol{T}_i$ do
5              if $u$ is a leaf node in $\boldsymbol{T}_i$ then
6                  add the original packet for $t$-th instance of query $\mathbf{q}_i$ to node $u$'s load;
7              else
8                  if $u$ receives a packet for $\boldsymbol{q}_i$ then
9                      store the packet to its buffer;
10                     if $u$ has received packets for $t$-th instance of query $\boldsymbol{q}_i$ from all
                         children in $\boldsymbol{T}_i$ then
11                         aggregate all packets to be one packet;
12                         add the packet to node $u$'s load;
13         $t \longleftarrow t + 1$;
14 return packets for each node for each period.
```

Then, we allocate time to a node from a selected cell to transmit. Suppose a cell is allocated with a time period T, we allocate each node u in a cell g with transmission time $T \cdot \frac{\mathcal{L}_{G,\mathcal{Q}}(u)}{\mathcal{L}_{G,\mathcal{Q}}(g_{v,h})}$.

Note that we can implement the proposed algorithms in a distributed manner as we do not require global coordination for each cell. Another benefit of cell partition is the increasing adaptivity of our method, thus we may allow the network to be more dynamic.

3.5 Packet-Level Scheduling

When it is the transmission time for a node, we determine the packet(s) to transmit from the node's load. We use a *rate monotonic* [15] method for packet scheduling and satisfy the duty-cycled constraints at the same time.

1. A packet of current instance has a lower priority than that of any previous instance.
2. A packet of current instance for a query with a shorter period should be scheduled later than any the packet of current instance for a query with a longer period. Similarly, a packet of previous instance for a query with a shorter period should be scheduled later than any packet of previous instance for a query with a longer period. Ties are broken randomly.

As proved in [15], the rate monotonic method achieves maximum performance for each packet to be transmitted before deadline if each node has a load of at most 0.69.

We can also use EDF scheduling instead if each node has a load of at most one. The details of EDF scheduling are shown in Algorithm 3.

Algorithm 2: Allocate Time to Node in Proportion to Load

```
Input  : The load of each node.
Output: Allocated time to nodes.
1 Perform cell partition and coloring;
2 Allocate time to each cell in proportion to its load and only cells with the same color
    can be allocated to the same time;
3 while TRUE do
4     for $i \in [c_2(\mathcal{M})]$ do
5         for each cell $g_{v,h}, v, h \in \mathbb{Z}$ with the i-th color do
6             for each node $u$ in the cell $g_{g,h}$ do
7                 allocate $u$'s transmission time as: $T \cdot \frac{\mathcal{L}_{G,\mathcal{Q}}(u)}{\mathcal{L}_{G,\mathcal{Q}}(g_{v,h})}$;
8 return allocated transmission time for each node.
```

Algorithm 3: EDF-Based Packet Scheduling of Each Node

```
Input  : A node $u$.
Output: A packet scheduling.
1 while TRUE do
2     if node $u$ is to transmit then
3         select a packet from $u$'s load with the earliest deadline, assume the packet is for
            query $\mathbf{q}_i$;
4         node $u$ transmits the packet to its parent in $\mathbf{T}_i$ at its parent's active time-slot
            under the duty-cycled model;
```

4 Schedulability Analysis

First, we derive **sufficient conditions** for schedulability of queries for data aggregation. Then, we propose **necessary conditions** for schedulability. We verify the schedulability of a given query set by comparing the sufficient conditions and necessary conditions.

4.1 Sufficient Condition on Schedulability

In Section 3, we proposed algorithms to schedule periodic queries for data aggregation. We prove that the proposed algorithms are *feasible*. Here we call an algorithm is feasible for a query set iff by using the algorithm, we can both avoid interference and answer all queries under the duty-cycled model.

Lemma 1. *The proposed algorithms in Section 3 are interference-free.*

Proof. In Algorithm 2, when allocating transmission time to nodes, each time we only select one node from only cells with the same color. By the definition of $c_2(\mathcal{M})$ which is the number of cell colors used for cell coloring, the proposed algorithms can avoid interference under various interference models.

Lemma 2. *The proposed algorithms in Section 3 answer all data aggregation queries.*

Proof. From Algorithm 2, each cell has $1/c_2(\mathcal{M})$ fraction of time to be active. At the same time, the load of each cell is at most $0.69/c_2(\mathcal{M})$. Considering the ratio of the node's load to the fraction of time the node is allocated to, note that each node's relative load is at most one. By using linear time allocation, we can allocate time to each node to transmit as long as each node has enough buffer. Therefore, we can answer the given set of data aggregation queries in time as long as the latency requirement is large enough.

The feasibility verification (Lemma 1, 2) implies schedulability of queries by using the proposed algorithms in Section 3. To sum up, we present a sufficient condition on which a query set is schedulable.

Theorem 1. *There exist scheduling algorithms to satisfy a data aggregation query set $\mathcal{Q}$ under an interference model $\mathcal{M}$, if*

$$\begin{cases} \mathcal{L}_{G,\mathcal{Q}}(g_{v,h}) & \leq \frac{0.69}{c_2(\mathcal{M})\cdot \boldsymbol{P}},\ \forall g_{v,h} \\ \sum_{\boldsymbol{q}_i \in \mathcal{Q}} \frac{\chi_i}{\boldsymbol{p}_i} & \leq \frac{0.69}{\boldsymbol{P}} \end{cases} \tag{1}$$

Here $\mathcal{L}_{G,\mathcal{Q}}(g_{v,h})$ is the load of cell $g_{v,h}$ and $c_2(\mathcal{M})$ is the number of cell colors. The value of $c_2(\mathcal{M})$ is provided in Lemma 5.

4.2 Necessary Condition on Schedulability

Consider the communication graph $G = (V, E)$ and a query set $\mathcal{Q}$, we define the source load of a node $u \in V$ as $\ell_{G,\mathcal{Q}}(u) = \sum_{u \in \mathcal{S}_i \cap \mathbf{q}_i \in \mathcal{Q}} \frac{\chi_i}{\mathbf{p}_i}$. The source load of a cell g is defined as the summation of the source loads of all nodes in the cell: $\ell_{G,\mathcal{Q}}(g_{v,h}) = \sum_{u \in V(g)} \ell_{G,\mathcal{Q}}(u)$ where $V(g)$ is the set of nodes from V lying inside the cell g.

To schedule nodes' transmissions in the worst case, for every set of clique nodes where no two nodes can transmit concurrently, the total load of all nodes can not exceed 1 in duty-cycled networks. Generally, for a cell $g_{v,h}$, in which the maximum number of sender nodes in $g_{v,h}$ that can transmit without interference is $c_1(\mathcal{M})$, the source load of the cell is at most $c_1(\mathcal{M})$.

Moreover, for a query $\mathbf{q}_i \in \mathcal{Q}$, the sink node ($v_s \in V$) needs to receive at least one packet for data aggregation during every period $\mathbf{p}_i$, which takes time χ_i. Given a query set $\mathcal{Q}$, the amount of data received at sink v_s, given by $\sum_{\mathbf{q}_i \in \mathcal{Q}} \frac{\chi_i}{\mathbf{p}_i}$, is at most one if $\mathcal{Q}$ can be answered.

To sum up, we present a necessary condition for a query set to be schedulable in Theorem 2.

Theorem 2. *A set of aggregation queries $\mathcal{Q} = \{\boldsymbol{q}_i : i \in [N]\}$ in a duty-cycled model are schedulable if the following conditions are satisfied.*

$$\begin{cases} \ell_{G,\mathcal{Q}}(g_{v,h}) & \leq c_1(\mathcal{M}),\ \forall g_{v,h} \\ \sum_{\boldsymbol{q}_i \in \mathcal{Q}} \frac{\chi_i}{\boldsymbol{p}_i} & \leq 1 \end{cases} \tag{2}$$

Here $\ell_{G,\mathcal{Q}}(g_{v,h})$ is the source load of an cell $g_{v,h}$. $c_1(\mathcal{M}) \geq 1$ is the maximum possible number of nodes that can transmit currently in a cell. The value of $c_1(\mathcal{M})$ is provided in Lemma 5.

4.3 Sufficiency vs Necessity

In Theorem 1 and 2, we present sufficient conditions and necessary conditions on scheduling queries for data aggregation respectively. Despite of a constant ratio difference for the sink's requirement, their main gap lies in that we use different terms for schedulability. In Theorem 2, we use the term *source load* of a cell to test schedulability, as the routing structure is unknown when testing schedulability, we cannot compute the load of a cell; while in Theorem 1, we use another term *load* to guarantee schedulability. Note that different routing structures (a set of data processing trees) for the query set have vast impact on the load of a cell, even if the source load of the cell is fixed.

To capture the exact difference between necessary and sufficient conditions on schedulability, we need to unify the terms used for comparing. We address the questions for two cases: (1) all nodes are source nodes for all queries, (2) only a subset are source nodes for every query.

Queries on All Nodes When all nodes have data, every node needs to report a packet during each period of a query $\mathbf{q}_i \in \mathcal{Q}$. Then the source load of each node is $\sum_{\mathbf{q}_i \in \mathcal{Q}} \frac{\chi_i}{\mathbf{p}_i}$. On the other hand, for data aggregation, each node only needs one transmission during a period of a given query $\mathbf{q}_i$. Then the load of a node is at most $\sum_{\mathbf{q}_i \in \mathcal{Q}} \frac{\chi_i}{\mathbf{p}_i}$ dispite the routing structure used. Thus the source load of a node is the same as the load of the node when all nodes have data for all queries. As a corollary, for each cell, the load is the same the source load.

Lemma 3. *When all nodes have data, given an aggregation query set $\mathcal{Q}$ with any interference model $\mathcal{M}$, for each cell, the load is the same as the source load.*

Using Lemma 3, we prove that the gap between necessary and sufficient conditions is a constant.

Theorem 3. *When all nodes have data, we can achieve a constant approximation ratio of $c_1(\mathcal{M}) \cdot c_2(\mathcal{M}) \cdot \boldsymbol{P}/0.69$ for scheduling an aggregation query set $\mathcal{Q}$ under various interference models.*

Proof. By Lemma 3, the sufficient conditions on schedulability given in Theorem 1 is equivalent to

$$\begin{cases} \ell_{G,\mathcal{Q}}(g_{v,h}) & \leq \frac{0.69}{c_2(\mathcal{M}) \cdot \mathbf{P}}, \ \forall g_{v,h} \\ \sum_{\mathbf{q}_i \in \mathcal{Q}} \frac{\chi_i}{\mathbf{p}_i} & \leq \frac{0.69}{\mathbf{P}} \end{cases}$$

At the same time, a necessary condition on schedulability given in Theorem 2 is:

$$\begin{cases} \ell_{G,\mathcal{Q}}(g_{v,h}) & \leq c_1(\mathcal{M}), \ \forall g_{v,h} \\ \sum_{\mathbf{q}_i \in \mathcal{Q}} \frac{\chi_i}{\mathbf{p}_i} & \leq 1 \end{cases}$$

By comparing the difference, we can see that the maximum load is at most $c_1(\mathcal{M})c_2(\mathcal{M}) \cdot \mathbf{P}/0.69$ times of the load we can schedule for each cell and the sink. Thus we can achieve an approximation ratio of $c_1(\mathcal{M}) \cdot c_2(\mathcal{M}) \cdot \mathbf{P}/0.69$ for schedulability. Therefore the proof is done.

Queries On Subset of Nodes When a subset of nodes have data for each query, the load can differ from the source load of a cell. In an extreme case, the load may be very large while the source load is zero. We compare necessary and sufficient conditions on schedulability from another perspective.

Lemma 4. *Given an interference model $\mathcal{M}$ and an aggregation query set $\mathcal{Q}$, by using our routing algorithms in Section 3,*

$$\begin{cases} \ell_{G,\mathcal{Q}}(g_{v,h}) & \leq 0.69/\left(2 \cdot c_2(\mathcal{M})\boldsymbol{P}\right), \forall g_{v,h} \\ \sum_{\boldsymbol{q}_i \in \mathcal{Q}} \frac{\chi_i}{\boldsymbol{p}_i} & \leq \frac{0.69}{2 \cdot c_2(\mathcal{M}) \cdot (c_3(\mathcal{M})-1) \cdot \boldsymbol{P}} \end{cases} \tag{3}$$

implies:

$$\begin{cases} \mathcal{L}_{G,\mathcal{Q}}(g_{v,h}) & \leq \frac{0.69}{c_2(\mathcal{M}) \cdot \boldsymbol{P}}, \forall g_{v,h} \\ \sum_{\boldsymbol{q}_i \in \mathcal{Q}} \frac{\chi_i}{\boldsymbol{p}_i} & \leq \frac{0.69}{\boldsymbol{P}} \end{cases}$$

Here $\boldsymbol{P}$ is the cycling period, $c_2(\mathcal{M})$ is the number of cell colors used, and $c_3(\mathcal{M}) > 1$ is the maximum size of a CDS in a cell plus one. The values of $c_2(\mathcal{M})$ and $c_3(\mathcal{M})$ are provided in Lemma 5.

Corollary 1. *Given a query set $\mathcal{Q}$ with $\mathcal{M}$, Equation (3) is a sufficient condition on schedulability.*

Theorem 4. *When a subset of nodes have data for each query, we can achieve an approximation ratio of $\max\{2c_1(\mathcal{M})c_2(\mathcal{M})\boldsymbol{P}/0.69, \frac{2 \cdot c_2(\mathcal{M}) \cdot (c_3(\mathcal{M})-1) \cdot \boldsymbol{P}}{0.69}\}$ on schedulability of an aggregation query set $\mathcal{Q}$ under various interference models.*

Proof. By Lemma 4, the sufficient conditions on schedulability in Theorem 1 is Equation (3),

$$\begin{cases} \ell_{G,\mathcal{Q}}(g_{v,h}) & \leq \frac{0.69}{(2 \cdot c_2(\mathcal{M}))\mathbf{P}}, \forall g_{v,h} \\ \sum_{\mathbf{q}_i \in \mathcal{Q}} \frac{\chi_i}{\mathbf{p}_i} & \leq \frac{0.69}{2 \cdot c_2(\mathcal{M}) \cdot (c_3(\mathcal{M})-1) \cdot \mathbf{P}} \end{cases}$$

while a necessary condition on schedulability is:

$$\begin{cases} \ell_{G,\mathcal{Q}}(g_{v,h}) & \leq c_1(\mathcal{M}), \ \forall g_{v,h} \\ \sum_{\mathbf{q}_i \in \mathcal{Q}} \frac{\chi_i}{\mathbf{p}_i} & \leq 1 \end{cases}$$

By comparing the difference, the approximation ratio is $\max\{2c_1(\mathcal{M}) \cdot c_2(\mathcal{M}) \cdot \mathbf{P}/0.69, \frac{2 \cdot c_2(\mathcal{M}) \cdot (c_3(\mathcal{M})-1) \cdot \mathbf{P}}{0.69}\}$.

Last, we provide the values of $c_1(\mathcal{M}), c_2(\mathcal{M})$, and $c_3(\mathcal{M})$ in Lemma 5 and summarize the notations in Table 1.

Lemma 5. *[19] The values of $c_1(\mathcal{M}), c_2(\mathcal{M}), c_3(\mathcal{M})$ under various interference models are as follows.*

$$c_1(\mathcal{M}) = \begin{cases} \frac{16 \cdot \rho^2}{(\rho-1)^2} \\ 36 \\ \lfloor \frac{2^{\kappa} \cdot P}{N_0 \beta^2} \rfloor \end{cases} \quad c_2(\mathcal{M}) = \begin{cases} 4 \\ 4 \\ O(1) \end{cases} \quad c_3(\mathcal{M}) = \begin{cases} 8 \cdot (\rho+4)^2 & \textit{under PrIM} \\ 200 & \textit{RTS/CTS} \\ 200 & \textit{under PhyIM} \end{cases}$$

Table 1. Notations

$\ell_{G,\mathcal{Q}}(u)$	source load of a node u	$\ell_{G,\mathcal{Q}}(g_{v,h})$	source load of a cell
$\mathcal{L}_{G,\mathcal{Q}}(u)$	load of a node u	$\mathcal{L}_{G,\mathcal{Q}}(g_{v,h})$	load of a cell
$\mathbf{l}$	cell side-length	$\mathcal{S}_i$	the set of source nodes for query i
$\mathbf{T}_i$	routing tree for query i	ℓ_i	size of data unit of query i
$\mathbf{P}$	duty cycling period	χ_i	transmission time of data unit of query i
v_s	sink node	$c_1(\mathcal{M})$	max # of senders transmitting in a cell
$\mathcal{Q}$	query set	$c_2(\mathcal{M})$	# of cell colors
$\mathbf{q}_i$	query i	$c_3(\mathcal{M})$	maximum size of CDS in a cell plus one

5 Conclusions

We designed real-time aggregation scheduling algorithms in duty-cycled sensor networks under various interference models. The proposed algorithms achieve constant approximation bound in terms of schedulability.

Some interesting questions are left for future research. The first one is to consider the aggregation latency of the proposed algorithms. The second one is to consider each query may have multiple sink nodes and the sink nodes of different queries could be different. The last one is to extend the proposed algorithms to deal with a more duty-cycled model. For example, the cycling periods of different nodes may be different.

References

1. Alicherry, M., Bhatia, R., and Li, L. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *ACM MobiCom* (2005), p. 72.
2. Chen, K., Gao, H., Cai, Z., Chen, Q., and Li, J. Distributed energy-adaptive aggregation scheduling with coverage guarantee for battery-free wireless sensor networks. In *IEEE INFOCOM* (2019).
3. Chen, Q., Gao, H., Cai, Z., Cheng, L., and Li, J. Energy-collision aware data aggregation scheduling for energy harvesting sensor networks. In *IEEE INFOCOM* (2018), pp. 117–125.
4. Chen, Q., Gao, H., Cheng, S., Li, J., and Cai, Z. Distributed non-structure based data aggregation for duty-cycle wireless sensor networks. In *IEEE INFOCOM* (2017), pp. 1–9.
5. Chipara, O., Lu, C., and Roman, G. Real-time query scheduling for wireless sensor networks. In *IEEE RTSS* (2007).
6. Du, D.-Z., and Wan, P.-J. Weighted CDS in unit disk graph. In *Connected Dominating Set: Theory and Applications*. Springer, 2013, pp. 77–104.
7. Gupta, P., and Kumar, P. The capacity of wireless networks. *IEEE Transactions on information theory 46*, 2 (2000), 388–404.
8. Ha, N. P. K., Zalyubovskiy, V., and Choo, H. Delay-efficient data aggregation scheduling in duty-cycled wireless sensor networks. In *ACM RACS* (2012), pp. 203–208.
9. He, Z., Cai, Z., Cheng, S., and Wang, X. Approximate aggregation for tracking quantiles and range countings in wireless sensor networks. *Theoretical Computer Science 607* (2015), 381–390.

10. Jiao, X., Lou, W., Feng, X., Wang, X., Yang, L., and Chen, G. Delay efficient data aggregation scheduling in multi-channel duty-cycled wsns. In *IEEE MASS* (2018), pp. 326–334.
11. Jiao, X., Lou, W., Wang, X., Cao, J., Xu, M., and Zhou, X. Data aggregation scheduling in uncoordinated duty-cycled wireless sensor networks under protocol interference model. *Ad Hoc & Sensor Wireless Networks 15*, 2-4 (2012), 315–338.
12. Li, J., Cheng, S., Cai, Z., Yu, J., Wang, C., and Li, Y. Approximate holistic aggregation in wireless sensor networks. *ACM Transactions on Sensor Networks 13*, 2 (2017), 11.
13. Liu, C. L., and Layland, J. W. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM 20*, 1 (1973), 46–61.
14. Liu, J. *Real-time systems*. Prentice Hall, 2000.
15. Shih, W., Liu, J., and Liu, C. Modified rate-monotonic algorithm for scheduling periodic jobs with deferred deadlines. *IEEE Transactions on Software Engineering 19*, 12 (1993), 1171–1179.
16. Wan, P.-J., Huang, S. C.-H., Wang, L., Wan, Z., and Jia, X. Minimum-latency aggregation scheduling in multihop wireless networks. In *ACM MobiHoc* (2009).
17. Xiao, S., Huang, J., Pan, L., Cheng, Y., and Liu, J. On centralized and distributed algorithms for minimizing data aggregation time in duty-cycled wireless sensor networks. *Wireless Networks* (2014), 1–13.
18. Xu, X., Cao, J., and Wan, P.-J. Fast group communication scheduling in duty-cycled multihop wireless sensor networks. In *WASA* (2012), Springer, pp. 197–205.
19. Xu, X., Li, X.-Y., and Song, M. Distributed scheduling for real-time data collection in wireless sensor networks. In *IEEE GLOBECOM* (2013), pp. 426–431.
20. Xu, X., Li, X.-Y., and Song, M. Efficient aggregation scheduling in multihop wireless sensor networks with sinr constraints. *IEEE Transactions on Mobile Computing 12*, 12 (2013), 2518–2528.
21. Xu, X., Li, X.-Y., Wan, P.-J., and Tang, S. Efficient scheduling for periodic aggregation queries in multihop sensor networks. *IEEE/ACM Transactions on Networking 20*, 3 (2012), 690–698.
22. Xu, X., and Song, M. Delay efficient real-time multicast scheduling in multi-hop wireless sensor networks. In *IEEE GLOBECOM* (2015), pp. 1–6.
23. Yan, X., Du, H., Ye, Q., and Song, G. Minimum-delay data aggregation schedule in duty-cycled sensor networks. In *WASA* (2016), Springer, pp. 305–317.
24. Yu, B., and Li, J.-Z. Minimum-time aggregation scheduling in duty-cycled wireless sensor networks. *Journal of Computer Science and Technology 26*, 6 (2011), 962–970.
25. Yu, D., Ning, L., Zou, Y., Yu, J., Cheng, X., and Lau, F. C. Distributed spanner construction with physical interference: constant stretch and linear sparseness. *IEEE/ACM Transactions on Networking 25*, 4 (2017), 2138–2151.
26. Yu, J., Huang, B., Cheng, X., and Atiquzzaman, M. Shortest link scheduling algorithms in wireless networks under the sinr model. *IEEE Transactions on Vehicular Technology 66*, 3 (2017), 2643–2657.