A Data Aggregation Scheduling Algorithm with Long-lifetime and Low-latency in Wireless Sensor Networks

Zhengyu Chen 1,2,3 *, Geng Yang 1,2, Lei Chen 1,2, Jin Wang 4, Jeong-Uk Kim 5

1 Key Laboratory of Broadband Wireless Communication & Sensor Networks Technology of Ministry of Education, Nanjing University of Posts & Telecommunications, Nanjing 210003, China
2 College of Computer Science & Technology, Nanjing University of Posts & Telecommunications, Nanjing 210046, China
3 School of Information Technology, Jinling Institute of Technology, Nanjing 211169, China
4 School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China
5 Department of Energy Grid, Sangmyung University, Seoul 110-743, Korea
zych@jit.edu.cn, {yangg, chenlei}@njupt.edu.cn, wangjin@nuist.edu.cn

Abstract. In scenarios of real-time data collection in long-term deployed Wireless Sensor Networks (WSNs), low-latency data collection and long network lifetime become key issue. We propose a Long-Lifetime and Low-Latency Data Aggregation Scheduling algorithm (L4DAS) in wireless sensor networks. Firstly, we formally formulate the problem of long-lifetime and minimum-latency aggregation scheduling as a constrained optimization problem, and then propose an approximation algorithm for this problem by constructing a degree-bounded minimum height spanning tree as aggregation tree and designing a maximum interference priority scheduling scheme to schedule the transmission of nodes in aggregation tree. Finally, through the simulation and comparisons, we prove the effectiveness of the algorithm.

Keywords: WSNs, Data aggregation, Network lifetime, latency

1 Introduction

Wireless Sensor Networks have been used for many long-term and real-time applications which require networks to operate long durations, as well as to transmit the sensed data to sink as soon as possible. Therefore, both maximizing lifetime and minimizing delay are the fundamental requirements. However, these two requirements are usually conflict with the limited battery power and communication bandwidth of sensor node. Sleep-wake scheduling and data aggregation are the effective mechanisms to prolong the lifetime of energy-constrained sensor networks. However, both sleep-wake scheduling and data aggregation can also lead to additional data collection delay. So, it is critical to research a problem of data aggregation scheduling with long-lifetime and low-latency in WSNs.

In WSNs, the Minimum-Latency Aggregation Schedule (MLAS) problem is to find
the schedule that routes data appropriately and has the shortest time for all requested
data to be aggregated to sink. Chen et al. in [1] proved that the MLAS problem is NP-
hard. [1, 2] proposed the centralized scheduling algorithms and proved the latency
upper-bound. Yu et al. [3] and Xu et al. [4] proposed the distributed scheduling
method with delay at most $24D + 6\Delta + 16$ and $16R + \Delta - 14$ time-slots respectively.
However, we note that the upper-bounds of these schemes are far too pessimistic
compared to the typical practical behavior of their algorithms. Malhotra et al.[5]
proved the latency lower-bound and got the better performance than the previous
algorithms by constructing a balanced shortest path tree (BSPT) and using a ranking-
based heuristic scheduling.

Energy efficiency is the biggest challenge in designing long-living sensor networks.
Wu et al. [6] proved that finding a maximum lifetime arbitrary tree is NP-complete,
and proposed an approximation algorithm that produces a sub-optimal tree. Malhotra
et al. [5] constructed a BSPT to prolong network lifetime. However, the network
lifetime in [5] determined by network architecture and the number of nodes.

In this paper, we propose a Long-Lifetime and Low-Latency Data Aggregation
Scheduling (L$^4$DAS) algorithm in WSNs. Our main contributions are as follows: (1)
we construct a degree-bounded minimum height spanning tree as aggregation tree
which provides a long network lifetime and is conducive to reduce scheduling
length and, (2) we propose a maximum interference priority scheduling algorithm to
schedule the transmission of nodes such that the latency is approximately minimized,
and (3) we carry out extensive simulations to verify our algorithms, and the results
show that our algorithm greatly outperforms the state-of-art schemes.

2 System Model and Problem Statement

2.1 System Model and Definitions

We consider a WSN consisting of $N$ sensor nodes $v_1, v_2, ..., v_N$ and sink node $v_s$.
All nodes have the same transmission range $r$ and interference range $r_i$. We use a
undirected graph $G(V, E)$ to represent this WSN, where $V = \{s, v_1, v_2, ..., v_N\}$
denotes the set of nodes and $E$ denotes the set of edges, i.e. there is an edge
$(v_i, v_j) \in E$ whenever their Euclidean distance $|v_i - v_j| \leq r$.

We consider the protocol interference model in which concurrent transmissions on
two edges $u \rightarrow v$ and $p \rightarrow q$ conflict with each other if and only if $v = q$,
$\|p - v\| \leq r_i$ or $\|q - u\| \leq r_i$ [4,5,7]. If link $u \rightarrow v$ conflict with link $p \rightarrow q$ , we call
that sender $u$ conflicts with sender $p$.

All sensors are homogeneous which have the same energy $E_{node}$ and consume energy $E_t$ and $E_r$ for transmitting and receiving one bit data respectively. We adopt
a perfect data aggregation model and TDMA-based scheduling protocol. In order to
facilitate the description, we give the following definitions:

Definition 1. Round. A Round is defined as the process of gathering data from all
nodes to the sink which is equivalent to a TDMA schedule period consisting of $T$ time slots. The duration of each round is called the scheduling latency. At each time slot, all senders and their corresponding parent nodes are scheduled in active state while the remaining nodes in sleep state.

**Definition 2. Network Lifetime.** The network lifetime is defined as the lifetime of the first dead node in the network. In data aggregation scheduling, we usually need to construct a data aggregation tree, so the network lifetime will be the lifetime of data aggregation tree. The lifetime $L(T_{ds})$ of the data aggregation tree $T_{ds}$ is

$$L(T_{ds}) = \min_{i=1,2,...,N} \left[ \frac{E_{node}}{KE_n, D(T_{ds}, v_i) + KE_n - E_{r}} \right]$$

where $D(T_{ds}, v_i)$ is the degree of node $v_i$ in $T_{ds}$, $K$ is the number of bits generated by each node in a round.

### 2.2 Problem Statement

We formulate the long-lifetime and low-latency data aggregation scheduling problem as a constrained optimization problem. Table 1 lists the symbols used in this optimization problem.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{v_i}$</td>
<td>The set denotes the neighbors of node $v_i$.</td>
</tr>
<tr>
<td>$f'(v_i, v_j)$</td>
<td>The decision variable of link $(v_i, v_j)$ at time slot $t$: $f'(v_i, v_j) = 1$ if link $(v_i, v_j)$ is scheduled to transmit at time slot $t$; $f'(v_i, v_j) = 0$ otherwise.</td>
</tr>
<tr>
<td>$R(v_i)$</td>
<td>The recipient corresponds to the sender $v_i$.</td>
</tr>
<tr>
<td>$t(v_i)$</td>
<td>The time slot is assigned for the transmission by node $v_i$.</td>
</tr>
<tr>
<td>$CHD(v_i)$</td>
<td>The set denotes the children nodes of $v_i$ in aggregation tree.</td>
</tr>
<tr>
<td>$D$</td>
<td>The maximum permitted degree of each node in aggregation tree.</td>
</tr>
</tbody>
</table>

The objective function of this problem becomes minimize scheduling latency $T$ subjects to the following constraints,

(a) $\sum_{v_i \in N_{v_i}} \sum_{i=1}^{T} f'(v_i, v_j) = 1 \quad i \in \{1, 2, ..., N\}$

(b) $\sum_{v_i \in N_{v_i}} \sum_{i=1}^{T} f'(v_i, v_j) = 0 \quad i \in \{1, 2, ..., N\}$

(c) $\sum_{v_i \in N_{v_i}} f'(R(v_i), v_j) = 0 \quad i \in \{1, 2, ..., N\}$

(d) $\sum_{v_i \in N_{v_i}} \sum_{i=1}^{T} f'(v_i, v_j) + \sum_{v_i \in N_{v_i}} \sum_{j=1}^{T} f'(v_i, v_j) = 0 \quad i \in \{1, 2, ..., N\}$

(e) $\sum_{v_i \in CHD(v_i)} f'(v_i, v_j) \leq D - 1 \quad i \in \{1, 2, ..., N\}$
Constraint (a) enforces a single transmission per node in each round. Constraint (b) ensures that, once a node transmits, it can no longer receive data from its children nodes in the same round. Constraint (c) guarantees that node can not transmit and receive simultaneously at the same time slot, i.e. half-duplex operation. Constraint (d) ensures requirement that there can be no interference at the recipient node. Constraint (e) guarantees that each node at most has \( D - 1 \) children nodes. This scheduling problem has been proved as a NP-hard problem [1]. We propose an approximation algorithm for this problem. We first construct a degree-bounded minimum height spanning tree as aggregation tree, and then design a maximum interference priority scheduling scheme to schedule the transmission of nodes in aggregation tree.

3. Main Design

3.1 Data Aggregation Tree Construction

We construct a degree-bounded minimum height spanning tree as data aggregation tree through two phases. First, we break up the graph into clusters with the diameter equaling to transmission range \( r \) and construct a cluster spanning tree respecting the degree constraint \( D \) in each cluster. Then we construct a global tree over the clusters and connect the spanning tree in each cluster to the global tree.

![Fig.1. An example of constructing a data aggregation tree](image)

Given \( G(V, E) \), depending on the size of the deployment area and transmission radius, we first partition \( V \) into pairwise disjoint sets [8]:

1. \( V = V_1 \cup V_2 \cup \ldots \cup V_m \), for \( \forall i, j \in \{1, 2, \ldots, m\} \), \( V_i \cap V_j = \emptyset \);
2. \( \forall v_i, v_j \in V_i, \ i \in \{1, 2, \ldots, m\}, \|v_i - v_j\| \leq r \);

Where \( V_i \ (i \in \{1, 2, \ldots, m\}) \) is the set of nodes in cluster \( i \). We partition \( V \) by tessellating the deployment area into a set of hexagonal clusters each of side length \( r/2 \) and assigning each node to a unique cluster whose center is closest to the node. We then choose a representative for each cluster to constitute a set \( R = \{u_1, u_2, \ldots, u_m\} \) where \( u_i \in V_i, \ i \in \{1, 2, \ldots, m\} \). The global tree will be constructed by these representatives. If some representatives can not connect with each other, we choose some connecting nodes \( C = \{c_1, c_2, \ldots, c_n\} \) to connect them. The global tree
should be constructed satisfying the following constraint,
\[
\min_{u \in R} \max_{v \in sR} \text{hops}(u, v) \\
\text{s.t.} \quad \max_{u \in R} \deg(u) \leq D - 1
\]  
(3)

where \( \text{hops}(u, v) \) is the hop distance from \( u \) to sink \( v \), \( \deg(u) \) is the degree of \( u \).

As shown in Fig.1, each circle represents a cluster. The nodes from \( v_1 \) to \( v_6 \) are representatives which correspond to 6 clusters respectively. \( c_1 \) and \( c_2 \) are connecting nodes. Dotted lines with arrows connect these nodes together to form the global tree. In each cluster, we construct a cluster spanning tree rooted at representative node while respecting the degree constraint \( D \). The Algorithm of constructing data aggregation tree is shown in Algorithm 1.

**Algorithm 1** Constructing Data Aggregation Tree

**Input:** \( G(V, E), \) sink, \( D \geq 2 \);  
**Output:** aggregation tree \( DA_T \).

**Step 1:** Partition \( V \) into pairwise disjoint sets \( V_1, V_2, \ldots, V_m \).

**Step 2:** Choose representatives \( R = \{u_1, u_2, \ldots, u_n\} \) and connecting nodes \( C = \{c_1, c_2, \ldots, c_s\} \) satisfying formula (3);

**Step 3:** Construct global tree \( G_T \) rooted at sink by connecting nodes in both \( R \) and \( C \);

**Run Step 4 to Step 6 for each cluster, construct \( m \) cluster spanning trees:**

**Step 4:** For cluster \( i \), \( V_i = \{v_{1i}, v_{2i}, \ldots, v_{ni}\} \) and representative node \( u_i \), initialize spanning tree \( T_i = (V_{ni}, E_{ni}) \), \( V_{ni} \leftarrow \{u_i\} \), \( E_{ni} \leftarrow \emptyset \);

**Step 5:** Choose \( n \) nodes (\( \{v_{ni}, \ldots, v_{ni}\} \)) from \( V_i \) as children nodes of \( u_i \), where \( n = D - \deg^{G_T}(u_i) \) (\( \deg^{G_T}(u_i) \) is the degree of node \( u_i \) in the global tree \( G_T \));

**Step 6:** Choose nodes from \( \{v_{ni}, \ldots, v_{ni}\} \) as parent nodes, then choosing \( D - 1 \) children nodes from \( V_i / \{v_{ni}, \ldots, v_{ni}\} \) for each parent node, repeat this process until all nodes in \( V_i = \{v_{1i}, v_{2i}, \ldots, v_{ni}\} \) are scheduled;

**Step 7:** Combine global tree \( G_T \) with all cluster spanning trees, \( DA_T = \bigcup_{1 \leq i \leq m} T_i \).

### 3.2 Aggregation Scheduling

A data aggregation schedule with delay \( T \) can be defined as a sequence of sender sets \( S_1, S_2, \ldots, S_r \) satisfying the following conditions:

1. \( S_i \cap S_j = \emptyset, \forall i \neq j \);
2. \( \forall v_i, v_j \in S_i, l = 1, 2, \ldots, n \), \( v_i \) and \( v_j \) do not conflict with each other;
3. At time slot \( k \), each sender in \( S_i \) transmits data to its parent node in \( V - \bigcup_{j=1}^{i-1} S_j \).
In this subsection, we design a maximum interference priority scheduling scheme to schedule the nodes in aggregation tree. At a time slot, nodes are eligible to be scheduled as senders if they are leaf-nodes not being scheduled at earlier time slot or intermediate nodes whose all children nodes have been scheduled at earlier slots. We define a set $F$ of such nodes as an eligible schedule set. When a node is scheduled to transmit, the number of receivers which can be interfered by this node is defined as the interference intensity of this node. For an eligible schedule set $F_i$ and the sets $S_1, S_2, ..., S_{i-1}$ that has been scheduled at earlier $i-1$ slots, we can calculate the interference intensity $DI(u)$ of node $u$ (for $\forall u \in F_i$) as

$$DI(u) = \left| \text{Nei}_u \cap \left( V - \bigcup_{j=1}^{i-1} S_j \right) - \text{Nei}_u \cap F_i - \{\text{par}_u\} \right|$$

where $\text{Nei}_u$ is the neighbors of node $u$ in graph $G$, $\text{par}_u$ is the parent node of node $u$ in aggregation tree, and $V - \bigcup_{j=1}^{i-1} S_j$ represents the set of nodes that have not been scheduled at time slot $i$.

For the eligible schedule set $F_i$ at time slot $i$, we first assign the node with the largest interference intensity to sender set $S_i$, then choose the node that has the largest interference intensity in $F_i - S_i$ and also do not conflict with the nodes in $S_i$. We continue this process until there do not exist node in $F_i - S_i$ satisfying conflict-free schedule. Algorithm 2 gives the processes of schedule.

**Algorithm 2** Aggregation Scheduling

**Input:** $G = (V, E)$, data aggregation tree $T_{da}$, sink;

**Output:** Sets of sender $S_1, S_2, ..., S_T$ ;

**Initialize:** $t = 1$, $i = 1$;

Repeat executing Step 1 to Step 4 until all nodes have been scheduled.

**Step 1:** Calculate the eligible schedule set $F_i$ at time slot $i$;

**Step 2:** $v = \arg \max_{u \in F_i} DI(u)$, $S_i = \{v\}$;

**Step 3:** if $p = \arg \max_{u \in F_i - S_i} DI(u)$ and node $p$ do not conflict with the nodes in $S_i$, $S_i \leftarrow \{p\}$; repeat executing Step 3 until there do not exist node in $F_i - S_i$ satisfying conflict-free schedule;

**Step 4:** Output $S_i$, $i = i + 1$

### 4. Simulation Results

We evaluate the performance of our algorithm using simulations. We randomly deploy sensor nodes in a $200m \times 200m$ field with a sink located at $(100m, 100m)$. All sensor nodes have the same transmission range and interference radius. The
energy consumptions for transmission \( E_t \) and reception \( E_r \) are 0.6 and 0.2 J/bit respectively. The initial energy \( E_{node} \) of each node is 3J. The length of data is 5 Bytes. The degree constraint \( D \) in L\(^4\)DAS is 4. We generate 30 random networks and present the averaged results for performing comparisons.

**Fig. 2.** Latency with different number of nodes

For aggregation latency, we compare L\(^4\)DAS with WIRES [5] and LDAS [7]. Transmission range \( r \) is fixed to 25m, the number of nodes varies from 400 to 1000 with an increment of 50. As can be seen from **Fig.2**, both L\(^4\)DAS and LDAS outperform WIRES. When the number of nodes is less than 650, LDAS outperforms L\(^4\)DAS. The reason is that when node density is not high, the aggregation tree of L\(^4\)DAS has higher height that can increase the low-bound of latency [5]. However, with the increasing of node density, the degree of nodes (especially sink) in LDAS increases rapidly which causes the low-bound of latency increases. So, when the node density exceeds a certain value, L\(^4\)DAS will always outperform LDAS.

**Fig.3.** Latency with different number of nodes and transmission range

In **Fig.3**, the number of nodes varies from 500 to 800 with an increment of 100, while transmission range \( r \) takes three values 25 and 30 respectively. It is indicated from the histogram that with the increment of the number of nodes and the transmission range, the improvement of our algorithm will be larger. These results indicate that our algorithm is greatly preferred for large scale and high density WSNs.
5. Conclusions and the future work

In this paper, we have investigated the data aggregation problem and considered its latency and network lifetime for WSNs in scenarios of real-time and long-term applications. We formulated the problem as a constrained optimization problem. Then, we proposed an approximation algorithm for this problem. Finally, through the simulation and comparisons, we proved that our algorithm outperforms the start-of-art schemes. In the future, we will research the distributed algorithms for constructing aggregation tree and scheduling the transmission of nodes.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) under Grant No. 2011CB302903, the National Science Foundation of China under Grant No. 61272084, the Innovation Project for postgraduate cultivation of Jiangsu Province under Grant No. CXZZ11_0402, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No. 11KJA520002. This work was also supported by the Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Knowledge Economy (MKE) Korea, and by the Natural Science Foundation of Jiangsu Province (No. BK2012461).

References