World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:11, No:5, 2017

# Aggregation Scheduling Algorithms in Wireless Sensor Networks

Min Kyung An

***Abstract*—In Wireless Sensor Networks which consist of tiny wireless sensor nodes with limited battery power, one of the most fundamental applications is *data aggregation* which collects nearby environmental conditions and aggregates the data to a designated destination, called a *sink* node. Important issues concerning the data aggregation are time efficiency and energy consumption due to its limited energy, and therefore, the related problem, named *Minimum Latency Aggregation Scheduling* (MLAS), has been the focus of many researchers. Its objective is to compute the *minimum latency schedule*, that is, to compute a schedule with the minimum number of *timeslots*, such that the sink node can receive the aggregated data from all the other nodes without any collision or interference. For the problem, the two interference models, the *graph model* and the more realistic *physical interference model* known as *Signal-to-Interference-Noise-Ratio* (SINR), have been adopted with different power models, *uniform-power* and *non-uniform power* (with *power control* or without power control), and different antenna models, *omni-directional* antenna and *directional* antenna models. In this survey article, as the problem has proven to be NP-hard, we present and compare several state-of-the-art *approximation* algorithms in various models on the basis of latency as its performance measure.

***Keywords*—Data aggregation, convergecast, gathering, approximation, interference, omni-directional, directional.

Open Science Index, Computer and Information Engineering Vol:11, No:5, 2017 waset.org/Publication/10007096

## I. Introduction

ONE of the most crucial applications of Wireless Sensor Networks (WSNs) is *data aggregation* (also called a *data gathering* or *convergecasting*) which monitors nearby environmental conditions periodically, and aggregates the gathered data from all nodes to a designated destination called a *sink* node (also called a *base station*). When a node sends its data to its receiver, a *collision* or *interference* can occur at the receiver if the transmission is interfered by signals concurrently sent by other nodes, and thus the data should be re-transmitted. Due to its periodic data gathering using limited energy of the tiny nodes, prolonging the network lifetime by reducing energy consumption which can be caused by the unnecessary retransmissions have been focused by researchers. An interesting approach is to assign *timeslots* to nodes to obtain a good *schedule*. Following the schedule, all data can be aggregated without any collision or interference on their way to the sink node. Since the data aggregation occurs periodically, reducing the *latency* of the schedule, that is, the problem of constructing a schedule with a minimum number of timeslots, has been a fundamental issue. This problem is known as the *Minimum Latency Aggregation Scheduling* (MLAS) problem in the literature.

M.-K. An is with the Department of Computer Science, Sam Houston State University, Huntsville, TX, 77341, USA (e-mail: an@shsu.edu).

In this survey article, we investigate the MLAS problem. In Section II, we introduce the network models, and then define the MLAS problem. Section III describes the NP-hardness results of the problem. We describe selected algorithms in Section IV, and review the existing results of several approximation algorithms in various networks models in Section V. Finally, we conclude with some remarks in Section VI.

## II. Network Models and Problem Definition

A wireless sensor network consists of a set $V$ of sensor nodes, each $u \in V$ of which is assigned a *transmission power level* $p(u)$, and equipped with an *omni-directional* or *directional* antenna with a fixed beam-width $\theta \in (0, 2\pi]$ and omni-directional receiving antenna. The *transmission range* $r(u)$ of $u$ is defined as the radius of the *broadcasting sector* $sec(u)$ of $u$. Notice that $sec(u)$ is a circle (when $\theta = 2\pi$) or a sector (when $\theta < 2\pi$) centered at $u$ with radius $r(u)$. Accordingly, a directed edge $(u, v)$ exists from node $u$ to node $v$, if $v$ resides in $sec(u)$.

### A. Antenna Models

There exist two antenna models adopted in WSNs: *omni-directional* and *directional* antenna models (See Fig. 1).

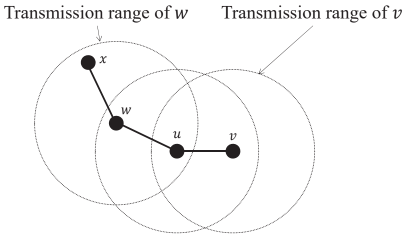
*1) Omni-Directional WSNs:* In omni-directional WSNs, each node is equipped with an omni-directional antenna with the beam-width $\theta = 2\pi$. The omni-directional WSNs are commonly modeled *undirected graphs*, where a undirected edge exists between $u$ and $v$ if $v$ resides in $sec(u)$ and $u$ resides in $sec(v)$.

*2) Directional WSNs:* Recently, new wireless sensor devices equipped with directional antenna whose directions can be collaboratively determined and orientated have been developed. WSNs, consisting of such nodes with a beam-width $\theta \in (0, 2\pi]$ are called directional WSNs, which are commonly modeled as *directed graphs*, where a directed edge exists from $u$ to $v$ if $v$ resides in $sec(u)$.

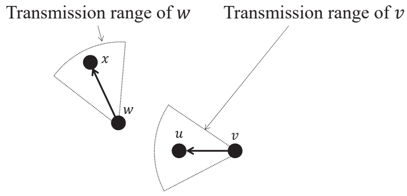
Commercially available directional antennas are typically designed for beam-widths of $\pi$, $2\pi/3$, $\pi/2$, $\pi/3$ and $\pi/4$ [1].

### B. Interference Models

*1) Graph (Protocol) Model:* Let $C_u = \{v | v \in V, d(u, v) \leq r(u)\}$ denote the set of nodes that are covered by $u$'s transmission range (i.e., the set of nodes that reside in $sec(u)$), where $d(u, v)$ denotes the Euclidean distance between $u$ and $v$. Then, two nodes $u$ and $v$ can communicate each other if they




(a) Omni-directional WSN. $v$ and $w$ cannot simultaneously send data to their receivers $u$ and $x$, respectively




(b) Directional WSN. $v$ and $w$ can simultaneously send data to their receivers $u$ and $x$, respectively

Fig. 1 Omni-directional WSN vs. Directional WSN

are covered by each other's transmission range, i.e., $u \in C_v$ and $v \in C_u$. Next, let $I_u = \{v \,|\, v \in V, \, d(u,v) \leq \rho \cdot r(u)\}$ denote the set of nodes that are covered by $u$'s *interference range* $\rho \cdot r(u)$, where $\rho \geq 1$ is the *interference factor*. Then, the *collision* (or *conflict*) is said to occur at a node $w$ if there exist other concurrently sending nodes $u$ and $v$ such that $w \in C_u \cap I_v$, where $\rho = 1$ (i.e., $C_u = I_u$). Also, the *interference* is said to occur at $w$ if there exist other concurrently sending nodes $u$ and $v$ such that $w \in C_u \cap I_v$, where $\rho > 1$ (i.e., $C_u \subset I_u$). The graph model concerning only collision (i.e., when $\rho = 1$) is called the *collision-free* (CF) graph model, whereas the graph model concerning both collision and interference (i.e., when $\rho \geq 1$) is called the *collision-interference-free* (CIF) graph model (See Fig. 2).

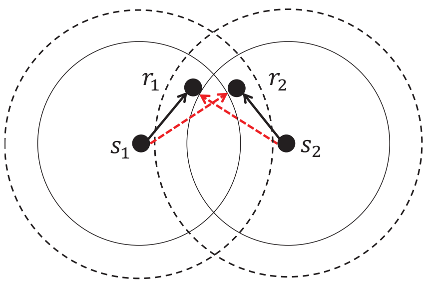



Fig. 2 Collision-Interference-Free Graph Model. Solid circles are broadcasting areas, and dotted circles are interference areas of the senders $s_1$ and $s_2$. For any $\rho > 1$, even there is no collision occurring at any receivers $r_1$ and $r_2$, $s_1$ and $s_2$ cannot send data simultaneously because $r_1$ and $r_2$ are interfered by $s_2$ and $s_1$, respectively

In the graph model, the communication graph is modeled as a directed graph $G = (V, E)$, where $E = \{ \, (u,v) \,|\, u, v \in V, d(u,v) \leq r(u)\}$.

*2) Physical Interference (SINR) Model:* Unlike the graph model, in the more realistic *physical interference* model known as *Signal-to-Interference-Noise-Ratio* (SINR) [2], real world phenomena is adequately captured by considering the cumulative interference caused by all the other concurrently transmitting nodes. In the SINR model, when a node $u$ sends data using its power level $p(u)$, the signal sent by $u$ fades and its receiver $v$ is interfered by the cumulative interference caused by all the other concurrently transmitting nodes, thus the signal sent to $v$ may not be strong enough to be received. The received power at $v$ is defined as $p(u) \cdot d(u,v)^{-\alpha}$, where $\alpha > 2$ is the *path loss exponent*. The receiver $v$ can receive the data transmitted by the sender $u$ without any interference only if the ratio of the received power at $v$ to the total interference caused by the set $X$ of other concurrently transmitting nodes and *background noise* $N > 0$ is beyond an SINR *threshold* $\beta \geq 1$. Formally, a receiver node $v$ can successfully receive data via the communication edge $(u,v)$ from a sender node $u$ only if

$$SINR_{(u,v)} = \frac{\frac{p(u)}{d(u,v)^{\alpha}}}{N + \sum_{w \in X \setminus \{u,v\}} \frac{p(w)}{d(w,v)^{\alpha}}} \geq \beta \geq 1 \qquad (1)$$

In this model, as $u$ can send its data to the nodes within the distance $(\frac{p(u)}{N\beta})^{\frac{1}{\alpha}}$ (i.e., $r(u) = (\frac{p(u)}{N\beta})^{\frac{1}{\alpha}}$) only, the communication graph can be modeled as a directed graph $G = (V, E)$, where $E = \{(u,v) \,|\, u, v \in V, \, d(u,v) \leq r(u)\}$. However, here, if $u$ on link $(u,v)$ of the *maximum link length* $r(u)$ is transmitting, then $u$ can be the only sending node, i.e., none of remaining nodes can transmit concurrently with $u$. Therefore, existing studies in the SINR model consider only links $(u,v)$, where $d(u,v) \leq \delta(\frac{p(u)}{N\beta})^{\frac{1}{\alpha}}$, for some constant $\delta \in (0,1)$ as in [3]. Accordingly, the communication graph is newly modeled as a directed graph $G = (V, E)$, where $E = \{(u,v) \,|\, u, v \in V, \, d(u,v) \leq \delta r(u)\}$.

### C. Power Models

*1) Uniform Power Model:* In this model, each node is assigned a uniform power level $r$, i.e., for each $u \in V$, $p(u) = r$. Thus, determining the right power levels to be assigned (also known as *power control*) is not part of the problem.

*2) Non-uniform Power Model:* In this model, each node $u \in V$ is typically assigned a different power level $p(u)$. The model is further divided into three different models:

- the *bounded power* model, where $u$ is assigned $p(u) \in [p_{min}, p_{max} \neq \infty]$,
- the *unlimited power* model, where $u$ is assigned $p(u) \in [p_{min}, \infty]$, and
- the *discrete power* model, where $u$ is assigned $p(u) \in \{p_1, p_2, ..., p_k\}$, where $k$ is the number of power levels used.

### D. Problem Definition

A *schedule* is a sequence of *timeslots*, at each of which, a set $\{u_{t_1}, u_{t_2}, ..., u_{t_\kappa}\}$ of sender nodes are scheduled to send their aggregated data to one of their neighbors in a set $\{v_{t_1}, v_{t_2}, \cdots, v_{t_\kappa}\}$ of receiver nodes using power levels $p_{t_i} \leq p_{max}$, $1 \leq i \leq \kappa$. Formally, at each timeslot $t$, we have an *assignment vector* $\pi_t = ((u_{t_1}, v_{t_1}, p_{t_1}), (u_{t_2}, v_{t_2}, p_{t_2}), \cdots, (u_{t_\kappa}, v_{t_\kappa}, p_{t_\kappa}))$ in which $u_{t_i}$ is assigned to send data with power level $p_{t_i}$, $1 \leq i \leq \kappa$, and

- Graph model: neither collision or interference occurs at any receiver $v_{t_i}$, or
- SINR model: the SINR inequality (1) is satisfied for every receiver $v_{t_i}$.

A schedule, as a sequence of assignment vectors, is denoted as $\Pi = (\pi_1, \pi_2, ..., \pi_\tau)$, where $\tau$ is its *latency*. The schedule $\Pi$ is *successful* if data from all nodes are aggregated to a sink node. The Minimum Latency Aggregation Scheduling (MLAS) is formally defined as follows:

**Input.** A set $V$ of nodes in a plane, a sink node $c \in V$.
**Output.** A successful minimum latency aggregation schedule.

Note that *determining and orienting antenna directions*, and *power control* are also parts of the problem, in directional networks and in the non-uniform power model, respectively.

TABLE I
NOTATIONS

| Symbol | Definition |
|---|---|
| $G$ | (Un)directed graph $G(V, E)$ |
| $V$ | The set of nodes |
| $E$ | The set of edges |
| $r$ | The uniform power level |
| $p(u)$ | Transmission power level of $u$ |
| $\theta$ | Beam-width of an antenna ($\theta \in (0, 2\pi]$) |
| $r(u)$ | Transmission range of $u$ |
| $sec(u)$ | Broadcasting sector of $u$ |
| $(u, v)$ | (Un)directed edge from $u$ to $v$ (between $u$ and $v$) |
| $C_u$ | The set of nodes covered by $u$'s transmission range |
| $I_u$ | The set of nodes covered by $u$'s interference range |
| $\rho$ | Interference factor ($\rho \geq 1$) |
| $d(u, v)$ | Euclidean distance between $u$ and $v$ |
| $k$ | The number of power levels |
| $\alpha$ | path loss exponent |
| $X$ | Concurrently transmitting nodes |
| $N$ | Background noise ($N > 0$) |
| $\beta$ | SINR Threshold ($\beta \geq 1$) |
| $t$ | A timeslot |
| $\pi_t$ | An assignment vector at timeslot $t$ |
| $\Pi$ | A schedule $(\pi_1, \pi_2, \cdots, \pi_\tau)$ |
| $\tau$ | Latency (i.e., the length of schedule $\Pi$) |
| $T$ | Data aggregation tree |
| $R$ | Network (graph) radius |
| $D$ | Network (graph) diameter |
| $\Delta$ | The maximum node degree |
| $n$ | The number of nodes |

See Table I for notations.

## III. NP-Hardness Results

In this section, we review the NP-hardness results of the MLAS problem in different network models.

### A. Omni-Directional WSNs

*1) Graph Model:* The first NP-hardness result of the MLAS problem was obtained by Chen et al. [4], [5] for the grid topologies in the CF model with uniform power level. They used a reduction from *restricted planar 3-SAT* problem which is known to be NP-complete [6]. In the CIF model with non-uniform power levels, An et al. [7] showed $\Omega(\log n)$ approximation lower bound in the metric model, i.e., there is no approximation algorithm having an approximation ratio better than $\Omega(\log n)$ for the problem unless $NP \subseteq DTIME(n \log \log n)$, by constructing a polynomial-time approximation-preserving reduction from the *Set Cover* problem which is known to be hard to approximate [8], [9].

*2) SINR Model:* Lam et al. [10], [11] was the first to show the NP-hardness of the MLAS problem in the geometric SINR model with the non-uniform power levels by constructing a polynomial time reduction from the *Partition* problem which was proven NP-complete [12]. Recently, the same authors showed its APX-hardness with the uniform power level in [13] by constructing a polynomial-time $L$-reduction from the *Minimum B-K-Set Cover* problem which is known to be APX-complete [14].

### B. Directional WSNs

We observe that the MLAS problem with omni-directional antenna (when $\theta = 2\pi$), whose NP-hardness results are shown in Section III-A, is a special case of the MLAS problem with directional antennas (when $\theta \in (0, 2\pi]$). Thus, we have

**Theorem 1.** *The Minimum Latency Aggregation Scheduling (MLAS) problem with directional antennas is NP-hard.*

## IV. Selected Algorithms

In this section, we study the common approaches used in the literature, and selected algorithms using the approaches.

### A. Common Approaches

Several researchers [3], [7], [10], [11], [15]–[20] have proposed aggregation scheduling algorithms which are divided into two phases: (1) *tree construction phase*, and (2) *scheduling phase*. In this section, we study two interesting methods, *constructing an MIS-based tree* and *network partitioning*, used for successful data aggregation in the phases (1) and (2), respectively. We start by introducing some standard notations [20] (cf. [21]).

- *Graph Center*: Given $G = (V, E)$, we call a node $s$ a *center* node if the hop distance from $s$ to the farthest node from $s$ is minimum.
- *Maximal Independent Set* (MIS): A subset $V' \subseteq V$ of the graph $G$ is said to be *independent* if for any vertices $u, v \in V'$, $(u, v) \notin E$. An independent set is said to be maximal if it is not a proper subset of another independent set.
- *Connected Dominating Set* (CDS): A dominating set (DS) is a subset $V' \subseteq V$ such that every vertex $v$ is either in $V'$ or adjacent to a vertex in $V'$. A DS is said to be *connected* if it induces a connected subgraph.

Next, we describe the two phases in the following.

1) ***Tree Construction Phase***:

One of the common approaches to construct a data aggregation tree is to construct one based on an MIS. The MIS-based tree $T$ is constructed as follows.

a) A breadth-first-search (BFS) tree (cf. [21]) on $G$ rooted at a sink node $c$ is first constructed. Here, $G$ is the initial communication graph constructed under the assumption that each node $u$ is assigned some power level $p(u)$, depending on network models.
b) Then, an MIS is computed level by level on the BFS tree using the algorithm in [22]. Let us call the nodes in the MIS *dominators*, and the remaining nodes *dominatees*. In Fig. 4 (a), the dominators in the MIS are represented by black nodes, and the dominatees are represented by gray nodes. The constructed MIS guarantees that the shortest hop-distance between two sets of a complementary pair, say $A$ and $MIS \setminus A$, where $A \subseteq MIS$, is *exactly two hops*. For example, in Fig. 4 (a), the shortest hop-distance between one complementary pair, $A$ and $MIS \setminus A$, where $A = \{v_1, v_2\}$ and $MIS \setminus A = \{v_3, v_4, v_5, c\}$, on $G$ is exactly two hops.
c) Next, the dominators are connected by dominatees thereby forming a CDS of $G$. The dominatees used to connect dominators are, from now on, called *connectors*. In Fig. 4 (b), the bolded edges represent the CDS.
d) If there exist dominatees not connected to the CDS, then each of them is connected to its neighboring dominator. In Fig. 4 (c), white nodes represent dominatees and the bolded edges represent $T$. This tree $T$ can be used as a data aggregation tree to guide to find an aggregation schedule.

2) ***Scheduling Phase***:
In this phase, algorithms assign timeslots to nodes so that they can send their data without collision or interference.

$T$**-based Scheduling:** Once $T$ is constructed, an algorithm can schedule nodes based on $T$ level by level as follows:

a) Assigning timeslots to dominatees to communicate with their upper level dominators (See Fig. 4(d).)
b) Assigning timeslots to dominators at level $i$ to communicate with their upper level connectors at level $i-1$, where $i = R, (R-2), \cdots, 2$ (See Fig. 4 (e) and 4(g).)
c) Assigning timeslots to connectors at level $j$ to communicate with their upper level dominators at level $j-1$, where $j = (R-1), (R-3), \cdots, 1$ (See Fig. 4( f) and 4(h).)

Every dominatee is scheduled in the phase (a). The phases (b) and (c) need to be repeated level by level until the sink node receives all the aggregated data. For instance, at the first iteration (Fig. 4 (e)), only the dominators at level 4 are selected as senders, then at the second iteration (Fig. 4 (f)), only the connectors at level 3 are selected as senders. Next, at the third iteration (Fig. 4 (g)), only the dominators at level 2 are selected as senders, and lastly (Fig. 4(h)), only the connectors at level 1 are selected as senders. Note that at each phase, nodes scheduled with the same timeslot must not cause any collision or interference when they send their data to their receivers. Here, existing algorithms use different methods to examine any possible collisions or interferences. Next, we review the methods to avoid collision and interference.

**Avoiding Collisions and Interference:** In order to check a possible collision or interference caused by a set of sender nodes which are assigned the same timeslot, *network partitioning* and *coloring* methods have been widely used.

- Network Partitioning: The first method is to partition a network into several cells using a space filling technique. One of the most common techniques is to partition a network into square cells whose side length is $\varphi$. $\varphi$ is set to be $r/\sqrt{2}$ for graph model [7], and set to be $\delta r/\sqrt{2}$ for the SINR model [3], [11] with the uniform power level $r$ so that only one dominator can reside in each square cell. Then, nodes are assigned the same timeslot if they are $K$ cells apart. For instance, in Fig. 4 (a), the two nodes in dashed circles are $K = 3$ cells apart, and thus they can be assigned the same timeslot to send data simultaneously.
- Coloring: Similar to the aforementioned method, some researchers have *colored* each cell so that any two nodes residing different cells but with the same color can be assigned the same timeslot [20], [23], [24]. For instance, An et al. [20] partitions a network into hexagons, and colored the hexagons with $M$-coloring . Fig. 3 (a) shows a 1-coloring, and a 7-coloring is obtained by enclosing the 1-coloring with a layer of hexagons as shown in Fig. 3 (b). Similarly, a 19-coloring obtained as shown in Fig. 3 (c), and recursively $M$-coloring is obtained in general. Fig. 3 (d) shows an example of tessellating a network with hexagons using 19-coloring.

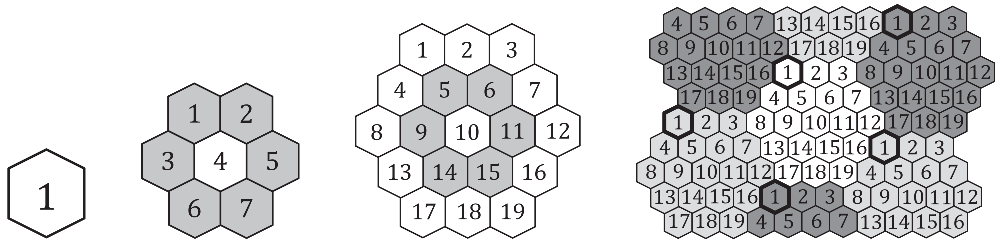
Each paper appropriately chose the values of $K$ and $M$ depending on their network models.

### B. Huang et al.'s Algorithm

In this section, we introduce the algorithm proposed by Huang et al. [15]. The authors studied the problem in the CF graph model (i.e., $\rho = 1$) with the uniform power level $r$ and the antenna beam-width $\theta = 2\pi$. It was the *first* constant-factor approximation algorithm proposed for the network model in the literature.

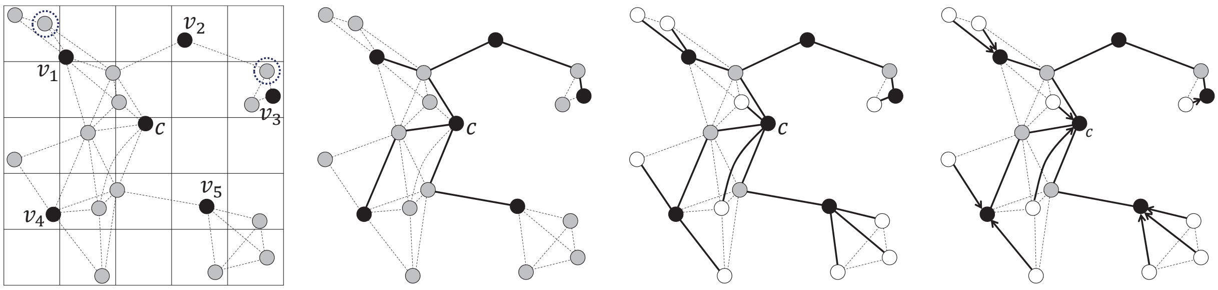
Huang et al.'s algorithm has two phases 1) tree construction phase 2) scheduling phase where *first-fit scheduling* algorithm is used.

1) Tree construction phase: The algorithm constructs the data aggregation tree $T$ as described in IV-A's ***Tree Construction Phase***.
2) Scheduling phase (*First-Fit Scheduling*): In this phase, the algorithm schedules nodes level by level based on $T$ as described in IV-A's $T$***-based Scheduling***.




(a) (b) (c) (d)

Fig. 3 Coloring by [20]: (a) 1-coloring, (b) 7-coloring, (c) 19-coloring, (d) Tessellation with 19-coloring

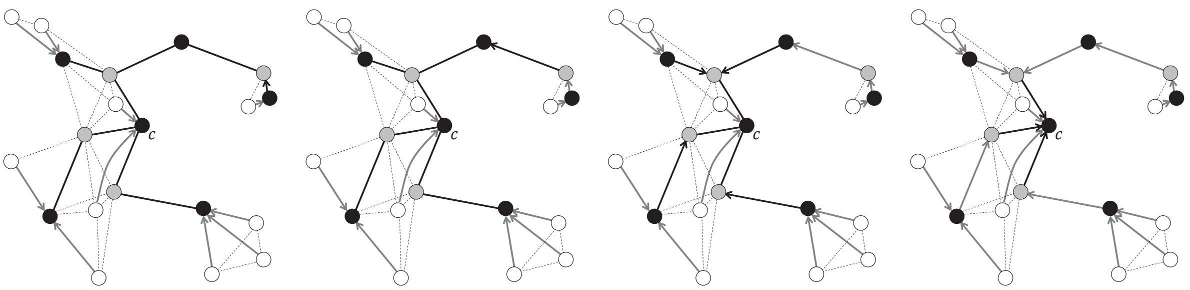



(a) Initial graph $G$ and its MIS represented by black nodes which are called dominators. Gray nodes represent dominatees

(b) CDS represented by nodes connected with bold lines. Black nodes represent dominators, and gray nodes connecting the black nodes represent connectors

(c) Aggregation tree $T$ rooted at $c$ represented by bold lines. Black nodes represent dominators, gray nodes represent connectors, and white nodes represent dominatees

(d) Only domintees are selected as senders



(e) Only the dominators at level 4 are selected as senders

(f) Only the connectors at level 3 are selected as senders

(g) Only the dominators at level 2 are selected as senders

(h) Only the connectors at level 1 are selected as senders

Fig. 4 (a)-(c) Illustration of MIS-based Tree Construction, (d)-(h) Illustration of Scheduling Based on a Data Aggregation Tree

a) Dominatees to dominators: First, a node $s_1$ from a set $S$ of dominatees is selected, and the algorithm examines if it causes any collision. As it is currently the only sender node selected, there is no possible collisions, and thus $s_1$ is added to a temporary set $X$, and is removed from $S$. Next, another node $s_2$ is selected from $S$, and the algorithm examines if it conflicts with any node in $X$ or not. To do so, the parent nodes of all nodes in $X$ in $T$ will be checked. As there is only one node $s_1$ in $X$, only $p_T(s_1)$, the parent of $s_1$ in $T$, is examined, and the algorithm also checks whether $s_2$ is adjacent to $p_T(s_1)$ in $G$. If yes, nothing to do, otherwise, it means that $s_2$ does not conflict with any nodes in $X$, and so $s_2$ is added to $X$ and removed from $S$. This process is repeated until the largest set $X$ is found such that all other nodes in $S$ will conflict at least one node in $X$. Notice that this $X$ is the maximal possible set whose nodes can be assigned the same timeslot $t$. Now the algorithm looks for other set of sender nodes who can be assigned the next timeslot $t+1$, and the process is repeated to find the next maximal possible set. The algorithm repeats the process until all elements in $S$ are scheduled. The details of the algorithm is shown in Algorithm 1.
b) Assigning timeslots to dominators at level $i$ to communicate with their upper level connectors at level $i-1$, and
c) Assigning timeslots to connectors at level $i-1$ to communicate with their upper level dominators at level $i-2$.

These processes (b) and (c) are repeated layer by layer until all data is aggregated to the sink node.

While assigning timeslots, it checks collisions using the *First-Fit Scheduling* algorithm.

**Algorithm 1** *First-Fit Scheduling* algorithm [15]

**Input**:Graph $G$, Sender set $S$, Tree $T$, Starting timeslot $t$
**Output**: Updated schedule $\Pi$, and the next timeslot $t$

```
X ← ∅
repeat
    for ∀s_i ∈ S do
        X ← {u | u is a sender node in π_t}
        if ∀x ∈ X, (p_T(x), u) ∉ E then
            π_t ← π_t ∪ (s_i, p_T(s_i), r)
            S ← S \ {s_i}
        end if
    end for
    Π ← Π ∪ {π_t}
    t ← t + 1
until S = ∅
return Π and t
```

The authors claimed that their algorithm produces schedules whose latencies are bounded by $28R + \Delta - 18$ which gives a nearly constant approximation.

**Theorem 2.** *For CF graph model with the uniform power level and the antenna beam-width $\theta = 2\pi$, Huang et al.'s algorithm [15] produces schedules whose latencies are bounded by $28R + \Delta - 18$, where $R$ is the network radius, and $\Delta$ is the maximum node degree.*

Note that here, $\Delta$ contributes to an *additive* factor instead of a *multiplicative* one unlike Chen at al.'s latency bound $R(\Delta - 1)$ [4], and thus the authors [15] claim that their algorithm has a significantly less latency bound than earlier algorithms especially when $\Delta$ is large.

### C. An et al.'s Algorithm

In this section, we introduce the algorithm proposed by An et al. [7]. The authors studied the problem in the CIF graph model (i.e., $\rho \geq 1$) with the uniform power level and the antenna beam-width $\theta = 2\pi$. The algorithm is based on Huang et al.'s algorithm [15] introduced in IV-B. It starts by partitioning the network into square cells, each of which has side length $r/\sqrt{2}$. This induces a grid where the upper-left corner has coordinates $(1, 1)$. A cell is denoted by Cell-ID $C(x, y)$ if its upper-left corner has coordinates $(x, y)$. It then has two phases 1) tree construction phase 2) scheduling phase as follows.

1) Tree construction phase: The algorithm constructs the data aggregation tree $T$ as described in IV-A's ***Tree Construction Phase***.
2) Scheduling phase: In this phase, the algorithm schedules nodes level by level based on $T$ as described in IV-A's ***$T$-based Scheduling***. At each level, it selects nodes which are $K = \lceil \rho \cdot \sqrt{2} + 2 \rceil$ cells apart to assign the same timeslot $t$, and these sender nodes do not cause any collision or interference at $t$.

The authors claimed that their algorithms produces schedules whose latencies are bounded by $\Delta \cdot K^2 + 21 \cdot K^2 \cdot R = O(\Delta + R)$ which gives a nearly constant approximation ratio (See Theorem 4).

**Algorithm 2** *Assign-Time-Slot* algorithm [7]

**Input**:Sender set $S$, Graph $G$, Tree $T$, Starting timeslot $t$
**Output**: Updated schedule $\Pi$, and the next timeslot $t$

```
for t_1 = 0, ··· , K − 1 and t_2 = 0, ··· , K − 1 do
    Let S' ⊆ S be the set of nodes with C(x, y) such that
    t_1 = x mod K, and t_2 = y mod K.
    for each node v ∈ S' do
        π_t ← {(v, p_T(v), r)}
        S' ← S' \ {v}
    end for
    Π ← Π ∪ {π_t}
    t ← t + 1
    S ← S \ S'
end for
return Π and t
```

**Theorem 3.** *For CIF graph model with the uniform power level and the antenna beam-width $\theta = 2\pi$, An et al.'s algorithm [7] produces schedules whose latencies are bounded by $O(\Delta + R)$, where $R$ is the network radius, and $\Delta$ is the maximum node degree.*

## V. Results of Existing Algorithms

In this section, we review the results of existing approximation algorithms in different interference and power models. We first start go over the lower bounds for the MLAS problem to understand better the current existing algorithms' approximation ratios.

**Theorem 4.** *[Lower Bounds] In order to produce a successful schedule, any data aggregation scheduling algorithm requires*

- *in graph model [19]:*
  - $\geq \max\{\Delta, \log R\}$ *timeslots, for* $\rho = 1$,
  - $\geq \max\{\frac{\Delta}{\phi}, R\}$ *timeslots, where* $\phi = \frac{2\pi}{\lfloor \arcsin \frac{\rho - 1}{2\rho} \rfloor}$, *for* $1 < \rho < 3$, *and*
  - $\geq \max\{\Delta, R\}$ *timeslots, for* $\rho \geq 3$, *and*
- *in SINR model [3]:* $\geq \max\{\frac{\Delta}{\omega}, R\}$, *where* $\omega = \frac{r^\alpha}{\beta} - 1$,

*where $\Delta$ is the maximum node degree of the network, and $R$ is the network radius.*

### A. Graph Model with Uniform Power and Beam-Width $\theta = 2\pi$

*1) Collision-Free Graph Model:* To review the results in this network model, we group the papers based on their way to build data aggregation trees.

1) **Tree-Based Data Aggregation:** [4], [5], [25], [26] have proposed tree-based data aggregation algorithms. Annamalai et al. [25] developed a heuristic algorithm, named *Convergecasting Tree Construction and Channel Allocation Algorithm* (CTCCAA), which constructs a data aggregation tree with timeslots assigned to nodes. The tree is constructed by spanning from a sink node $c$ to its neighbors $sec(c)$, from $sec(c)$ to $sec(sec(c))$, and so on. The algorithm uses several constraints (e.g., given orthogonal codes, distances, etc) when children

choose their parents to which they send their data. [26] introduced the *Latency Bounded Data Aggregation Tree* (LBDAT) algorithm, which is developed from *Light Approximation Shortest-path Tree* (LAST) algorithm [27] that aims at balancing Minimum Spanning Tree (MST) and Shortest Path Tree (SPT), and proved that LBDAT produces schedules whose latency is bounded by $D \cdot \min\{\Delta, n\}$, where $D$ is the network diameter. The authors also claimed that there exists a schedule whose length is at most $\min\{\log_2(\frac{n+2}{3})+1, 3\log_3(\frac{2n+3}{5})+2\}$. Later, [4], [5] introduced an approximation algorithm with the ratio of $(\Delta - 1)$. The algorithm proceeds by incrementally constructing smaller and smaller SPTs rooted at $c$. Their technique forms a data aggregation tree after a schedule is made, not making a schedule after a tree is constructed.

2) **MIS-Tree-Based Data Aggregation:** On the other hand, [15]–[17] used the MIS-based tree $T$ as their data aggregation tree. For the ***Scheduling Phase*** (See Section IV-A), [15]–[17] introduced different algorithms. [15] named the algorithm '*first-fit algorithm*', and [16] named its algorithm '*distributed aggregation scheduling algorithm*' (SCHDL for short). [15], [16], and [17] also proved that their algorithms produce schedules whose latencies are bounded by $23R + \Delta - 18$, $24D + 6\Delta + 16$, and $16R + \Delta - 14$, respectively. Recently, [28] constructed a novel data aggregation tree which is different from the commonly used CDS (Connected 2-hop Dominating Sets-based) approaches. Their tree is built with a balanced Connected 3-hop Dominating Sets (C3DS)-based structure. Furthermore, their algorithm simultaneously constructs a data aggregation tree and schedules. The authors stated that their latency bound, $12R + \Delta - 2$, is currently the best.

*2) Collision-Interference-Free Graph Model:* Considering the both collision and interference (i.e., $\rho \geq 1$), [18] was the first study to introduce approximation algorithms. The authors first proposed three algorithms with $\rho = 1$ which use $T$ as their aggregation tree. Each of the algorithms uses *Sequential Aggregation Scheduling* (SAS), *Piplelined Aggregation Scheduling* (PAS), and *Enhanced Pipelined Aggregation Scheduling* (E-PAS) algorithms, respectively, in their Scheduling Phases. They also proved that the latencies produced by the algorithms are $15R+\Delta-4$, $2R+O(\log R)+\Delta$, and $\left(1+O\left(\log R/\sqrt[3]{R}\right)\right)+\Delta$, with SAS, PAS, and E-PAS, respectively. Authors [18] stated that novel structures like the two connected dominating sets and the canonical inward arborescences used by these three algorithms are of independent interest and are expected to have applications in other communication scheduling. Then, they obtained two aggregation schedules with $\rho > 1$ by expanding SAS and PAS. The expanding algorithm is called the *$\rho$-expansion of a communication scheduling* algorithm. $\rho$-expansions of communication schedules produced with SAS and PAS were proved to have the latencies of $\zeta_{\rho+1}(15R + \Delta - 4)$ and $\zeta_{\rho+1}(2R + \Delta + O(\log R))$, respectively, where $\zeta_\eta = \frac{\pi}{\sqrt{3}}\eta^2 + \left(\frac{\pi}{2}+1\right)\eta+1$. Later, [19] proved that the overall lower bound of data aggregation scheduling under any interference model is $\max\{\log n, R\}$, and obtained lower bounds, $\max\{\Delta/\phi, R\}$ and $\max\{\Delta, R\}$, for the cases $1 < \rho < 3$ and $\rho \geq 3$, respectively, where $\phi = (2\pi)/(\lfloor\arcsin\frac{\rho-1}{2\rho}\rfloor)$ (See Theorem 4). They also proposed an aggregation algorithm that assigns timeslots based on $T$, and uses *Improved data Aggregation Scheduling* (IAS) algorithm in the Scheduling Phase. The latency bound of produced schedules is $16R+\Delta-14$. [7] also proposed a constant factor approximation algorithm, named *Cell Coloring*, whose latency is bounded by $O(\Delta + R)$. The Cell Coloring algorithm uses $T$ as its data aggregation tree, and uses the partitioning technique for scheduling.

### *B. Graph Model with Non-Uniform Power and Beam-Width $\theta = 2\pi$*

*1) Collision-Free Graph Model:* In the CF model, assuming that the maximum transmission range of a node is unbounded, [29] proposed a very simple randomized distributed algorithm whose latency is bounded by $O(\log n)$, where $n$ is the number of nodes in the network. They also showed that the obtained bound is tight, and any algorithm needs $\Omega(\log n)$ timeslots for data aggregation in an arbitrary network.

### *C. SINR Model with Uniform Power and Beam-Width $\theta = 2\pi$*

The first approximation algorithm in the SINR model with uniform power level was introduced by [3]. The algorithm uses $T$ as its data aggregation tree, and partitions the network for scheduling. into cells with $\varphi = r/\sqrt{2}$. Then, any nodes which are $K = \lceil(\frac{4\beta\kappa\cdot P\cdot\varphi^{-\alpha}}{(\sqrt{2})^{-\alpha}P\cdot\varphi^{-\alpha}-\beta N}+1+\sqrt{2})^{\frac{1}{\alpha}}\rceil$, where $\kappa = \frac{\alpha(1+2^{-\frac{\alpha}{2}})}{\alpha-1} + \frac{\pi 2^{-\frac{\alpha}{2}}}{2(\alpha-2)}$, cells apart are assigned the same timeslot. The latency of schedules produced by the algorithm is $O(\Delta+R)$.

### *D. SINR Model with Non-Uniform Power and Beam-Width $\theta = 2\pi$*

In the following, we review the studies done in the SINR model with non-uniform power levels.

*1) Bounded Power Model with Power Control:* While most existing works studied the problem under the uniform power model or the unlimited power model, few researchers investigated the problem assuming a more realistic non-uniform power assignment where the maximum power level is bounded. [30] introduced a distributed algorithm that computes schedules whose latency is bounded by $O(R + \Delta\log n)$. Their algorithm runs on a data aggregation tree which is also MIS-based. When scheduling, the algorithm also controls powers (i.e., assigns appropriate power levels to nodes). Later, [24] proposed a constant factor approximation algorithm with $O(R + \chi)$ timeslots, where $\chi$ is the link length diversity. Under a reasonable assumption about $\chi$, the number of timeslots is bounded by $O(R+\log n)$ which gives a constant approximation ratio. The algorithm partitions a network using the divide-and-conquer approach, applying a multilevel partitioning technique which repeatedly partitions cells into smaller subcells.

*2) Unbounded Power Model with Power Control:* In this model, it is assumed that the transmission power of each node is large enough to cover the maximum node distance in the network. [23], [31] proposed a distributed algorithm that yields $O(\chi)$ timeslots, where $g$ is the logarithm of the ratio between the lengths of the longest and shortest links in a network. They also proposed a centralized algorithm whose latency is $O(\log^3 n)$ which was improved by [32] to $O(\log n)$.

*3) Discrete Power Model without Power Control:* [11] studied the problem in the dual power model, where each node $u \in V$ is assigned either the high power level or the low power level, i.e., $p(u) \in \{p_{min}, p_{max}\}$, and proposed two constant factor approximation algorithms with the latency bounds $O(R + \Delta)$, *Square-Based Aggregation Scheduling* (SBAS) and *Hexagon-Based Aggregation Scheduling* (HBAS) algorithms. Both algorithms schedule nodes based on the MIS-based data aggregation $T$, but use different network partitioning techniques: partitioning the network into *square cells* or *hexagonal cells*.

### *E. Networks with Beat-Width $\theta < 2\pi$*

With the uniform power model, Liu at et. [33] proposed a nearly constant factor approximation algorithm with $0 < \theta \leq 2\pi$. To the best of our knowledge, there have been no studies done with non-uniform power model.

## VI. Conclusion

In this survey article, we have presented a comprehensive survey of approximation algorithms for the MLAS problem which is known to be NP-hard [4], [5], [7], [10], [11]. In both graph and SINR models, several studies provided constant-factor approximation algorithms. One interesting observation is allowing unlimited power significantly effects on reducing time complexity in both the models. Even though the bounded power model is considered to be more realistic than the unbounded power model, studying of data aggregation algorithms with unlimited powers could be still meaningful for this reason.

Most existing algorithms do not consider any possible changes of a network topology which can be caused by the death of nodes or the mobility of nodes. Especially, central algorithms using a pre-built data aggregation tree cannot be directly applied to a network in such environment. Further research of distributed data aggregation algorithms for unknown topology, e.g., a network in which nodes have no knowledge of the number of neighboring nodes within any given radius from themselves [30], holds the promise of providing data aggregation in mobile WSNs.

## References

[1] S. Roy, Y. C. Hu, D. Peroulis, and X.-Y. Li, “Minimum-Energy Broadcast Using Practical Directional Antennas in All-Wireless Networks,” in *INFOCOM*, 2006.

[2] P. Gupta and P. R. Kumar, “The Capacity of Wireless Networks,” *IEEE Transctions on Information Theory*, vol. 46, pp. 388 – 404, 2000.

[3] X.-Y. Li, X. Xu, S. Wang, S. Tang, G. Dai, J. Zhao, and Y. Qi, “Efficient Data Aggregation in Multi-hop Wireless Sensor Networks under Physical Interference Model,” in *MASS*, 2009, pp. 353–362.

[4] X. Chen, X. Hu, and J. Zhu, “Minimum Data Aggregation Time Problem in Wireless Sensor Networks,” in *MSN*, 2005, pp. 133–142.

[5] ——, “Data Gathering Schedule for Minimal Aggregation Time in Wireless Sensor Networks,” *IJDSN*, vol. 5, no. 4, pp. 321–337, 2009.

[6] D. Lichtenstein, “Planar Formulae and Their Uses,” *SIAM J. Comput.*, vol. 11, no. 2, pp. 329–343, 1982.

[7] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen, “Minimum Data Aggregation Schedule in Wireless Sensor Networks,” *IJCA*, vol. 18, no. 4, pp. 254–262, 2011.

[8] C. Lund and M. Yannakakis, “On the Hardness of Approximating Minimization Problems,” *Journal of the ACM*, vol. 41, no. 5, pp. 960 – 981, 1994.

[9] U. Feige, “A Threshold of $\ln(n)$ for Approximating Set Cover,” *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[10] N. X. Lam, M. K. An, D. T. Huynh, and T. N. Nguyen, “Minimum Latency Data Aggregation in the Physical Interference Model,” in *MSWiM*, 2011, pp. 93–102.

[11] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen, “Minimum Latency Data Aggregation in the Physical Interference Model,” *COMCOM*, vol. 35, no. 18, pp. 2175–2186, 2012.

[12] R. Karp, “Reducibility among Combinatorial Problems,” in *Complexity of Computer Computations*. Plenum Press, 1972, pp. 85–103.

[13] N. X. Lam, T. Tran, M. K. An, and D. T. Huynh, “A Note on the Complexity of Minimum Latency Data Aggregation Scheduling with Uniform Power in Physical Interference Model,” *Theoretical Computer Science*, vol. 569, pp. 70–73, 2015.

[14] C. H. Papadimitriou and M. Yannakakis, “Optimization, Approximation, and Complexity Classes,” *J. Comput. System Sci.*, vol. 43, no. 3, pp. 425–440, 1991.

[15] S. C. H. Huang, P.-J. Wan, C. T. Vu, Y. Li, and F. Yao, “Nearly Constant Approximation for Data Aggregation Scheduling,” in *INFOCOM*, 2007, pp. 6–12.

[16] B. Yu, J. Li, and Y. Li, “Distributed Data Aggregation Scheduling in Wireless Sensor Networks,” in *INFOCOM*, 2009, pp. 2159–2167.

[17] X. Xu, S. Wang, X. Mao, S. Tang, P. Xu, and X.-Y. Li, “Efficient Data Aggregation in Multi-hop WSNs,” in *GLOBECOM*, 2009, pp. 3916–3921.

[18] P.-J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, and X. Jia, “Minimum-Latency Aggregation Scheduling in Multihop Wireless Networks,” in *MOBIHOC*, 2009, pp. 185–194.

[19] X. Xu, X. Y. Li, X. Mao, S. Tang, and S. Wang, “A Delay-Efficient Algorithm for Data Aggregation in Multihop Wireless Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 163–175, 2011.

[20] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen, “Minimum Latency Aggregation Scheduling in Interference-Aware 3-Dimensional WSNs,” in *ICWN*, 2013.

[21] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 3rd ed. McGraw-Hill Higher Education, 2009.

[22] P.-J. Wan, K. Alzoubi, and O. Frieder, “Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks,” in *INFOCOM*, 2002, pp. 159–1604.

[23] H. Li, Q. S. Hua, C. Wu, and F. C. M. Lau, “Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks under Physical Interference Model,” in *MSWiM*, 2010, pp. 360–367.

[24] N. X. Lam, M. K. An, D. T. Huynh, and T. N. Nguyen, “Scheduling Problems in Interference-Aware Wireless Sensor Networks,” in *ICNC*, 2013, pp. 783–789.

[25] V. Annamalai, S. K. S. Gupta, and L. Schwiebert, “On Tree-based Convergecasting in Wireless Sensor Networks,” in *WCNC*, 2003, pp. 1942–1947.

[26] H. Du, X. Hu, and X. Jia, “Energy Efficient Routing and Scheduling for Real-time Data Aggregation In WSN,” *COMCOM*, vol. 29, no. 17, pp. 3527–3535, 2006.

[27] S. Khuller, B. Raghavachari, and N. E. Young, “Balancing Minimum Spanning Trees and Shortest-Path Trees,” *Algorithmica*, vol. 14, no. 4, pp. 305–321, 1995.

[28] H. Yousefi, M. Malekimajd, M. Ashouri, and A. Movaghar, “Fast Aggregation Scheduling in Wireless Sensor Networks,” *IEEE Trans. Wireless Communications*, vol. 14, no. 6, pp. 3402–3414, 2015.

[29] A. Kesselman and D. R. Kowalski, “Fast Distributed Algorithm for Convergecast in Ad Hoc Geometric Radio Networks,” *Journal of Parallel and Distributed Computing*, vol. 66, no. 4, pp. 578–585, 2006.

[30] N. Hobbs, Y. Wang, Q.-S. Hua, D. Yu, and F. C. M. Lau, “Deterministic Distributed Data Aggregation under the SINR Model,” in *TAMC*, 2012, pp. 385–399.

[31] H. Li, C. Wu, Q.-S. Hua, and F. C. Lau, "Latency-minimizing Data Aggregation in Wireless Sensor Networks under Physical Interference Model," *Ad Hoc Networks*, vol. 12, pp. 52–68, 2014.
[32] M. M. Halldórsson and P. Mitra, "Wireless Connectivity and Capacity," in *SODA*, 2012, pp. 516–526.
[33] H. Liu, Z. Liu, D. Li, X. Lu, and H. Du, "Approximation Algorithms for Minimum Latency Data Aggregation in Wireless Sensor Networks with Directional Antenna," *Theor. Comput. Sci.*, vol. 497, pp. 139–153, 2013.