# Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks under Physical Interference Model

Hongxing Li
Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong
hxli@cs.hku.hk

Qiang-Sheng Hua
Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong
qshua@cs.hku.hk

Chuan Wu
Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong
cwu@cs.hku.hk

Francis C.M. Lau
Department of Computer Science
The University of Hong Kong
Pokfulam Road, Hong Kong
fcmlau@cs.hku.hk

## ABSTRACT

Minimum-Latency Aggregation Scheduling ($MLAS$) is a problem of fundamental importance in wireless sensor networks. There however has been very little effort spent on designing algorithms to achieve sufficiently fast data aggregation under the physical interference model which is a more realistic model than traditional protocol interference model. In particular, a distributed solution to the problem under the physical interference model is challenging because of the need for global-scale information to compute the cumulative interference at any individual node. In this paper, we propose a distributed algorithm that solves the $MLAS$ problem under the physical interference model in networks of arbitrary topology in $O(K)$ time slots, where $K$ is the logarithm of the ratio between the lengths of the longest and shortest links in the network. We also give a centralized algorithm to serve as a benchmark for comparison purposes, which aggregates data from all sources in $O(log^3 n)$ time slots (where $n$ is the total number of nodes). This is the current best algorithm for the problem in the literature. The distributed algorithm partitions the network into cells according to the value $K$, thus obviating the need for global information. The centralized algorithm strategically combines our aggregation tree construction algorithm with the non-linear power assignment strategy in [9]. We prove the correctness and efficiency of our algorithms, and conduct empirical studies under realistic settings to validate our analytical results.


*MSWiM'10,* October 17–21, 2010, Bodrum, Turkey.


## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*wireless communication, network topolgoy*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*geometric problems and computations, sequencing and scheduling*

## General Terms

Algorithms, Theory, Performance

## Keywords

Minimum Latency, Data Aggregation, Wireless Sensor Networks, Physical Interference Model

## 1. INTRODUCTION

Data aggregation is a habitual operation of practical use in all wireless sensor networks, which transfers data (*e.g.*, temperature) collected by individual sensor nodes to a sink node. The aggregation typically follows a tree topology rooted at the sink. Intermediate sensor nodes of the tree may simply merge and forward all received data or perform certain operations (*e.g.*, computing the sum, maximum or mean) on the data. In a wireless environment, because of the interference among wireless transmissions, transmissions to forward the data need to be meticulously coordinated. The fundamental challenge can be stated as: How to schedule the aggregation transmissions in a wireless sensor network such that no undesired interference may occur and the total number of time slots used (referred to as *aggregation latency*) is minimized? This is known as the *Minimum-Latency Aggregation Scheduling (MLAS)* problem in the literature [1, 5, 15, 16, 17]. Note that we divide the time into time slots, which makes the design and analysis more tractable.

The $MLAS$ problem is typically approached in two steps: (i) data aggregation tree construction, and (ii) link transmission scheduling. For (ii), we assume the simplest mode where every non-leaf node in the tree will make only one transmission which is after all the data from its child nodes have been received. To solve the MLAS problem, we require that no collision of transmissions should occur due to wireless interference. If the above two steps are being carried out simultaneously, we have a "joint" design.

To model the interferences, most existing literature assume the *protocol interference model*. The best results known for the *MLAS* problem or similar ones ([5, 15, 16, 17]) bound the aggregation latency in $O(\Delta + R)$ time slots, where $R$ is the radius of the sensor network counted by hop count and $\Delta$ is the maximal node degree. A more realistic model than the protocol interference model is the *physical interference model* [14]. So far, however, very little research has been done to address the *MLAS* problem under the physical interference model.

The protocol interference model considers only interferences within a limited region, whereas the physical interference model tries to capture the cumulative interferences from all other currently transmitting nodes or links. More precisely, in the physical interference model, the transmission of link $e_i$ can be successful if the following Signal-to-Interference-Noise-Ratio (*SINR*) condition is satisfied:

$$\frac{P_i/d_{ii}^{\alpha}}{N_0 + \sum_{e_j \in \Lambda - \{e_i\}} P_j/d_{ji}^{\alpha}} \geq \beta \tag{1}$$

Here $\Lambda$ denotes the set of links that transmit simultaneously with $e_i$. $P_i$ and $P_j$ denote the transmission powers at the transmitter of link $e_i$ and that of link $e_j$, respectively. $d_{ii}$ ($d_{ji}$) is the distance between the transmitter of link $e_i$ ($e_j$) and the receiver of link $e_i$. $\alpha$ is the path loss ratio, which has a typical value between 2 and 6. $N_0$ is the ambient noise. $\beta$ is the *SINR* threshold for a successful transmission, which is at least 1.

A solution to the MLAS problem can be a centralized one, a distributed one, or something in between. For a large sensor network, a distributed solution is certainly the desired choice. Distributed scheduling algorithm design is significantly more challenging with the physical interference model, as "global" information in principle is needed by each node to compute the cumulative interference at the node. The only work targeting the physical interference model we are aware of is [7] which presents an efficient distributed solution to the *MLAS* problem with latency bound of $O(\Delta+R)$. One of the drawbacks of their work is that no efficiency guarantee can be given for arbitrary topologies.

In this paper, we tackle the minimum-latency aggregation scheduling problem under the physical interference model, by designing both a centralized and a distributed scheduling algorithm. Our algorithms are applicable to arbitrary topologies. Our main focus is on the proposed distributed algorithm; the centralized algorithm is included for the purpose of serving as a benchmark in the performance comparison, which however may be a practical solution for situations where centralization is not a problem. The distributed algorithm we propose, *Cell-AS*, circumvents the need to collect global interference information by partitioning the network into cells according to a parameter called link length diversity ($K$) which is the logarithm of the ratio between the lengths of the longest and the shortest links. Our centralized algorithm, *NN-AS*, has the best aggregation performance with respect to the current literature. It combines our aggregation tree construction algorithm with the non-linear power assignment strategy proposed in [9].

We conduct theoretical analysis to prove the correctness and efficiency of our algorithms. We show that the distributed algorithm *Cell-AS* achieves a worst-case aggregation latency bound of $O(K)$ (where $K$ is the link length diversity), and the centralized algorithm *NN-AS* achieves a worst-case bound of $O(\log^3 n)$ (where $n$ is the total number of sensor nodes). In addition, we derive a theoretical optimal lower bound for the *MLAS* problem under any interference model—$\log(n)$. Given this optimal bound, the approximation ratios of *Cell-AS* and *NN-AS* are $O(K/\log n)$ and $O(\log^2 n)$, respectively. We also compare our distributed algorithm with Li et al.'s algorithm in [7] both analytically and experimentally. We show that both algorithms have an $O(n)$ latency upper bound for their respective worst cases while *Cell-AS* can still be effective in Li et al.'s worst cases. Our experiments under realistic settings demonstrate that *Cell-AS* can achieve up to a 35% latency reduction as compared to Li et al.'s.

The remainder of this paper is organized as follows. We discuss related work in Sec. 2 and formally present the problem model in Sec. 3. The *Cell-AS* and *NN-AS* algorithms are presented in Sec. 4 and 5, with extensive theoretical analysis given in Sec. 6. We report our empirical studies of the algorithms in Sec. 7. Finally, we conclude the paper in Sec. 8.

# 2. RELATED WORK

## 2.1 Data Aggregation

Data aggregation is a prominent problem in wireless sensor networks. There exist a lot of exciting work trying to solve the problem [1, 5, 7, 15, 16, 17]. Minimizing the aggregation scheduling length is one of the most important concerns.

To the best of our knowledge, all except one paper [7] assume the protocol interference model. [1] proposed a data aggregation algorithm with latency bound of $(\Delta-1)R$, where $R$ is the network radius by hop count and $\Delta$ is the maximal node degree. The NP-hard proof of the *MLAS* problem is also presented. The current best contributions [5, 15, 16, 17] bound the aggregation latency by $O(\Delta + R)$.

[5] is the first work that converted $\Delta$ from a multiplicative factor to an additive one. The algorithm builds on the basis of maximal independent set which is also used in [17]. The latter one actually gives a distributed solution.

In [15], the *MLAS* problem is cast in multihop wireless networks with the assumption that each node has a unit communication range and an interference range of $\rho \geq 1$. [16] proposes an aggregation schedule for a distributed solution and proves a lower-bound of $\max\{\log n, R\}$ on the latency of data aggregation under any graph-based interference model; $n$ is the network size.

The only solution for the MLAS problem under the physical interference model is [7] by Li et al. They have proposed a distributed aggregation scheduling algorithm with constant power assignment, which can achieve a latency bound of $O(\Delta+R)$. However, the efficiency of their algorithm cannot be guaranteed in arbitrary topologies, which is a consequence of constant power assignment.

## 2.2 Link Scheduling under the Physical Interference Model

The physical interference model has received increased attention in recent years for its more realistic abstraction of wireless networks [14]. For the physical interference model, some have focused on the maximum achievable network capacity which is primarily determined by the result of the *Minimum Length link Scheduling (MLS)* problem. The MLS problem is closely related to the link scheduling step of our MLAS problem here. Recent results [9, 10, 11] demonstrate that, with the physical interference model, as opposed to the protocol interference model, the network capacity can be greatly increased.

Moscibroda et al. formally propose the problem of link scheduling complexity in [10]. In [11], Moscibroda et al. study topology control for the physical interference model and obtain a theoretical upper bound on the scheduling complexity of arbitrary topologies in wireless networks.

In [9], Moscibroda applies link scheduling to the data gathering tree in wireless sensor networks with an $O(log^2 n)$ complexity. It was the first time a scaling law that describes the achievable data rate in worst-case sensor networks was derived. Goussevskaia et al. [3] make the milestone contribution of proving the NP-completeness of a special case of the *MLS* problem.

# 3. THE PROBLEM MODEL

We consider a wireless sensor network of $n$ arbitrarily distributed sensor nodes $v_0, v_1, \ldots, v_{n-1}$ and a sink node $v_n$. Let directed graph $G = (V, E)$ denote the tree constructed for data aggregation from all the sensor nodes to the sink, where $V = \{v_0, v_1, \ldots, v_n\}$ is the set of all nodes, and $E = \{e_0, e_1, \ ..., \ e_{n-1}\}$ is the set of transmission links in the tree with $e_i$ representing the link from sensor node $v_i$ to its parent.

Our problem at hand is to pick the directed links in $E$ to construct the tree and to come up with an aggregation schedule $S = \{S_0, S_1, \ ..., \ S_{T-1}\}$, where $T$ is the total time span for the schedule and $S_t$ denotes the subset of links in $E$ scheduled to transmit in time slot $t, \forall t = 0, \ldots, T-1$. A correct aggregation schedule must satisfy the following conditions. First, any link should be scheduled exactly once, *i.e.*, $\bigcup_{t=0}^{T-1} S_t = E$ and $S_i \cap S_j = \emptyset$ where $i \neq j$. Second, a node cannot act as a transmitter and a receiver in the same time slot, in order to avoid the *primary interference*. Let $T(e_i)$ and $R(e_i)$ be the transmitter and the receiver of link $e_i$, respectively, and $T(S_t)$ and $R(S_t)$ denote the transmitter and receiver set for the links in $S_t$, respectively. We have $T(S_t) \cap R(S_t) = \emptyset, \forall t = 0, \ldots, T-1$. Third, a non-leaf node $v_i$ transmits to its parent only after all the links in the subtree rooted at $v_i$ have been scheduled, *i.e.*, $T(S_i) \cap R(S_j) = \emptyset$ where $i < j$. Finally, each scheduled transmission in time slot $t$, *i.e.*, link $e_i \in S_t$, should be correctly received by the corresponding receiver under the physical interference model considering the aggregate interference from concurrent transmissions of all links $e_j \in S_t - \{e_i\}$ i.e., the condition $\frac{P_i/d_{ii}^{\alpha}}{N_0 + \sum_{e_j \in S_t - \{e_i\}} P_j/d_{ji}^{\alpha}} \geq \beta$ should be satisfied.

The minimum-latency aggregation scheduling problem can be formally defined as follows:

Definition 1. *Minimum-Latency Aggregation Scheduling: Given a set of nodes $\{v_0, v_1, \ldots, v_{n-1}\}$ and a sink $v_n$, construct an aggregation tree $G = (V, E)$ and a link schedule $S = \{S_0, S_1, \ ..., \ S_{T-1}\}$ satisfying $\bigcup_{t=0}^{T-1} S_t = E$, $S_i \cap S_j = \emptyset$ where $i \neq j$, and $T(S_i) \cap R(S_j) = \emptyset$ where $i \leq j$, such that the total number of time slots $T$ is minimized and all transmissions can be correctly received under the physical interference model.*

Without loss of generality, we assume that the minimum Euclidean distance between each pair of nodes is 1. As our algorithm design targets at arbitrary distribution of sensor nodes, we assume the upper bound of the transmission power at each node to be large enough to cover the maximum node distance of the network, such that no node would be isolated. Each node in the network knows its location. This is not hard to achieve during bootstrapping stage in a network where the sensors are stationary.

# 4. DISTRIBUTED AGGREGATION SCHEDULING

Our main contribution is an efficient *distributed* scheduling algorithm called *Cell Aggregation Scheduling (Cell-AS)* for solving the *MLAS* problem with arbitrary distribution of sensor nodes.

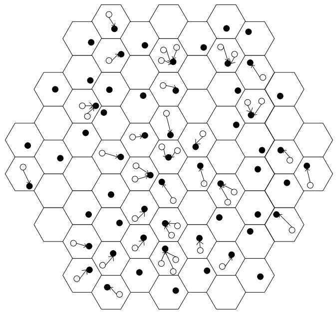
Our distributed algorithm features joint tree construction-link scheduling-power control in a phase-by-phase fashion to achieve minimum aggregation latency; whereas tree construction and link scheduling are separate steps in [7]. We first present the key idea behind our algorithm design and then discuss important techniques to implement the algorithm in a fully distributed fashion.

## 4.1 Design Idea

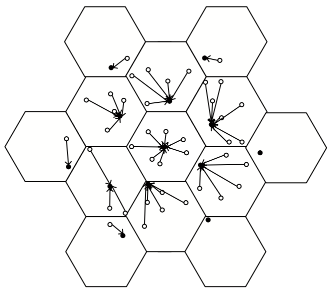
Our distributed algorithm first aggregates data from sensor nodes in each small area with short transmission links, and then further aggregates data in a larger area by collecting from those small ones with longer transmission links; this process repeats until the entire network as the largest area is covered.

We classify the lengths of all possible transmission links in the network into $K+1$ categories: $[3^0, 2\cdot 3^0], (2\cdot 3^0, 2\cdot 3^1], \ldots, (2\cdot 3^{K-1}, 2\cdot 3^K]$, where $K$ is bounded by the network's maximum node distance $D$ with $2\cdot 3^{K-1} < D \leq 2\cdot 3^K$. A link from node $v_i$ to node $v_j$ falls into category $k$ if the Euclidean distance between these two nodes lies within $(2\cdot 3^{k-1}, 2\cdot 3^k]$ with $k = 1, \ldots, K$ or $[3^0, 2\cdot 3^0]$ with $k = 0$. We define $K$ as the *link length diversity* which is proportional to the logarithm of the ratio between the lengths of the longest and the shortest possible links in the network. In our design, aggregation links in category $k$ are treated and their transmissions are scheduled (to aggregate data in the smaller areas) before links in category $k+1$ are processed (to aggregate data in the larger areas).

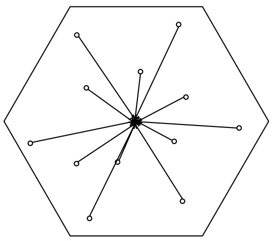
Our algorithm carries out its actions in an iterative fashion: In round $k$ ($k = 0, \ldots, K$), we divide the network into hexagonal cells of side length $3^k$. In each cell, a node with the shortest distance to the sink is selected as the head, responsible for data aggregation; the other nodes in the cell directly transmit to the head with links no longer than $2\cdot 3^k$. In the next round $(k+1)$, only the head nodes in the previous round remain in the picture. The network is covered by hexagonal cells of side length $3^{k+1}$ and a new head is selected for data aggregation in each cell. After $K+1$ rounds of the algorithm, only one node will remain, which should have collected all the data in network, and will transmit the aggregated data to the sink node in one hop. Fig. 1 gives an example of the algorithm in a sensor network with 3 link length categories, in which selected head nodes are in black.



(a) Round 0.



(b) Round 1.



(c) Round 2.

**Figure 1: The iterations of *Cell-AS*: an example with 3 link length categories with sink in the center.**

In each round $k$ of the algorithm, links of length category $k$ are scheduled as follows to avoid interference and to minimize the aggregation latency. We assign colors to the cells and only cells with the same color can schedule their link transmissions concurrently. To bound the interference among concurrent transmissions, cells of the same color need to be sufficiently far apart. We use $\frac{16}{3}X^2+12X+7$ colors in total, such that cells of the same color are separated by a distance of at least $2(X+1)3^k$ with $X=(6\beta(1+(\frac{2}{\sqrt{3}})^\alpha\frac{1}{\alpha-2})+1)^{1/\alpha}$, as illustrated in Fig. 2. (The grey cell in the center represents a landmark cell in Sec. 4.B.) We will show in Sec. 6 that by using these many colors, we are able to bound the interferences and thus prove the correctness and efficiency of our algorithm. Inside each cell, the transmission links from all other nodes to the head are scheduled sequentially.

The *Cell-AS* algorithm is summarized as Algorithm 1 where the scheduling of links in cells of the same color is carried out according to Algorithm 2.

## 4.2 Distributed Implementation

The algorithm can be implemented in a fully distributed fashion. The key is to decide at each peer the following:

### *4.2.1 Location and synchronization*

In the bootstrapping phase, the origin $(0,0)$ is set to a central position in the sensor network. Each node learns its location coordinates $(x,y)$ with respect to the origin, using GPS. In fact, only a small number of nodes need to use

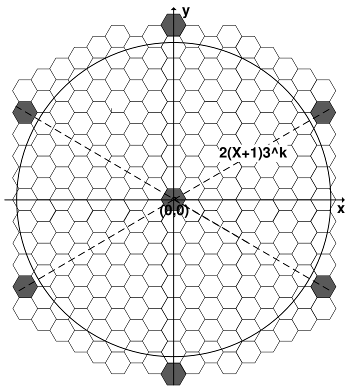




**Figure 2: Link scheduling in one round of *Cell-AS*: cells with the same color are separated by a distance of at least $2(X+1)3^k$.**

**Algorithm 1** Distributed Aggregation Scheduling (*Cell-AS*)

**Input**: Node set $V$ with sink $v_n$.
**Output**: Tree link set $E$ and link schedule $S$.

$k:=0$; $V:=V-\{v_n\}$; $t:=0$;
$X:=(6\beta(1+(\frac{2}{\sqrt{3}})^\alpha\frac{1}{\alpha-2})+1)^{1/\alpha}$;
**while** $|V|\neq 1$ **do**
  Cover the network with cells of side length $3^k$ and color them with $\frac{16}{3}X^2+12X+7$ colors;
  **for** $i:=1$ to $\frac{16}{3}X^2+12X+7$ **do**
    $E_i:=\emptyset$, where $E_i$ is link set in cells of color $i$;
    **for** each cell $j$ with color $i$ **do**
      Select node $v_h$ in cell $j$ closest to sink $v_n$ as head;
      Construct links from all other nodes in cell $j$ to $v_h$;
      Add the links to $E_i$ and $E$;
      Remove all the nodes in cell $j$ except $v_h$ from $V$;
    **end for**
    $S:=S\cup$ Same-Color-Cell-Scheduler($E_i$, $t$);
  **end for**
  $k:=k+1$;
**end while**
$v_h:=$ the only node in $V$; Construct link $e_h$ from $v_h$ to $v_n$;
$E:=E\cup\{e_h\}$; $S:=S\cup\{\{e_h\}\}$;
**return** $E$ and $S$;

**Algorithm 2** Same-Color-Cell-Scheduler

**Input**: Link set $E_i$ and time slot index $t$.
**Output**: Partial link schedule $PS_i$ for links in $E_i$.

$X:=(6\beta(1+(\frac{2}{\sqrt{3}})^\alpha\frac{1}{\alpha-2})+1)^{1/\alpha}$;
Define constant $c:=N_0\beta X^\alpha$;
$PS_i:=\emptyset$;
**while** $E_i\neq\emptyset$ **do**
  $S_t:=\emptyset$;
  **for** each cell $j$ with color $i$ **do**
    Choose one non-scheduled link $e_m$ in cell $j$;
    Assign transmission power $P_m:=c\times d_{mm}^\alpha$;
    $S_t:=S_t\cup\{e_m\}$; $E_i:=E_i-\{e_m\}$;
  **end for**
  $PS_i:=PS_i\cup\{S_t\}$; $t:=t+1$;
**end while**
**return** $PS_i$;

GPS, while the others can obtain their coordinates through relative positioning. (*e.g.*, [13]).

Each node in the sensor network carries out the distributed algorithm in a synchronized fashion—*i.e.*, it knows the start of each round $k$. Such synchronization can be achieved using one of the effective synchronization algorithms in the literature (*e.g.*, [8]).

### 4.2.2 Neighbor discovery

In each round $k$, the network is divided into cells of side length $3^k$ in the fashion as illustrated in Fig. 2. Each node can determine the cell it resides in in this round based on its location. It can then discover its neighbors in the cell via local broadcasting [2]. The broadcasting range is $2 \cdot 3^{k+1}$, such that all nodes in the same cell can be reached.

### 4.2.3 Head selection

The head of a cell in round $k$ is the node in the cell closest to the sink. All the nodes are informed of the sink's location in the bootstrapping stage of the algorithm, or even before they have been placed in the field. Since each node knows the location information of all its neighbors in the same cell, it can infer whether itself is the head, or some other neighbor is the head of the cell in this round.

### 4.2.4 Distributed link scheduling

In each round $k$, coloring of the cells are done as illustrated in Fig. 2. As each node knows which cell it resides in, it can calculate color $i$ of its cell in this round. Cells of the same color are scheduled according to the sequence of their color indices, *i.e.*, cells with color $i$ can schedule their transmissions before those with color $i+1$. The head node in a cell is responsible to decide when the other nodes in its cell can start to transmit, and to announce the completion of transmissions in its cell to all head nodes within $2(X+1)3^k$ distance.

A head node in a cell with color $i+1$ waits until it has received completion notifications from all head nodes in cells of color $i$ within $2(X+1)3^k$ distance. It then schedules the transmission of all the other nodes in its cell one by one, by sending "pulling" messages. For a non-head node in the cell, it waits for the "pulling" message from the head node and then transmits its data to the head.

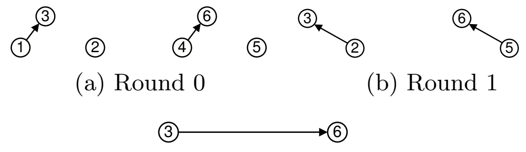
When the algorithm is executed round after round, only the nodes that have not transmitted (the heads in previous rounds) remain in the execution, until their transmission time slots arrive.

## 5. CENTRALIZED AGGREGATION SCHEDULING

When global information is assumed to be available at each sensor, a centralized scheduling algorithm can achieve the best aggregation latency for the *MLAS* problem. We present in the following a centralized algorithm, *Nearest-Neighbor Aggregation Scheduling (NN-AS)*, which does exactly that.

Our centralized algorithm progresses also in a phase-by-phase fashion, with joint tree construction and link scheduling. In each round, we find a nearest neighbor matching among all the sensor nodes that have not transmitted their data, and schedule all the links in the matching.

We start the algorithm with all the sensor nodes in $V - \{v_n\}$. We find for each node $v_i$ the nearest neighbor node




(a) Round 0 (b) Round 1

(c) Round 2

**Figure 3: The iterations of *NN-AS*: an example of 6 sensor nodes.**

**Algorithm 3** Centralized Aggregation Scheduling (*NN-AS*)

**Input**: Node set $V$ with sink $v_n$.
**Output**: Tree link set $E$ and link schedule $S$.

$k := 0$; $E := \emptyset$; $S := \emptyset$; $V = V - \{v_n\}$;
**while** $|V| \neq 1$ **do**
  $M_k := \emptyset$;
  **for** each $v_i \in V$ **do**
    **if** $v_i \notin T(M_k) \cup R(M_k)$ **then**
      Find $v_i$'s nearest-neighbor $v_j \in V$;
      **if** $v_j \notin T(M_k) \cup R(M_k)$ **then**
        Construct link $e_i$ from $v_i$ to $v_j$; $M_k := M_k \cup \{e_i\}$;
      **end if**
    **end if**
  **end for**
  $E := E \cup M_k$; $S := S \cup$ Phase-Scheduler$(M_k)$;
  $V := V - T(M_k)$; $k := k + 1$;
**end while**
$v_i :=$ the only node in $V$; Construct link $e_i$ from $v_i$ to $v_n$;
$E := E \cup \{e_i\}$; $S := S \cup \{\{e_i\}\}$;
**return** $E$ and $S$;

$v_j$, where neither $v_i$ nor $v_j$ has already been included in the matching, and establish a directed link from $v_i$ to $v_j$. For example, in Fig. 3 where a sensor network of 6 sensor nodes is shown, the matching we identify in round 0 contains two links, from 1 to 3 and from 4 to 6, respectively. We then schedule the links in matching $M_0$ (of round 0), using the link scheduling algorithm with non-linear power assignment proposed in [9]. This algorithm schedules a set of links in a network generated as the nearest neighbor matching as in our case, with guaranteed scheduling correctness under the physical model. After all transmissions in round 0 are scheduled, all the nodes that have transmitted are removed, and the algorithm repeats with the reduced node set. In Fig. 3(b), nodes 2, 3, 5, and 6 remain, and two links are generated using the nearest neighbor criterion and scheduled for transmission. The process repeats until only one sensor node remains, which will transmit the aggregate data to the sink node in one hop.

The centralized algorithm is summarized as Algorithm 3, where *Phase-Scheduler* calls upon the algorithm in [9] to generate the schedule for links in matching $M_k$ in round $k$.

**Algorithm 4** Phase-Scheduler

**Input**: Link set $M_k$.
**Output**: Link schedule $S_m$.

For space limitation, please refer to [9] for details.

## 6. ANALYSIS

In this section, we prove the correctness of our distributed and centralized algorithms and analyze their efficiency with respect to the bound of aggregation latency. Due to space limitation, we only present the major results and the outlines of proof to each theorem. The analytical details can be found in technical report [6].

## 6.1 Correctness

Theorem 1 (Correctness of *Cell-AS*). *The distributed* Cell-AS *in Algorithm 1 can construct a data aggregation tree and correctly schedule the transmissions under the physical model.*

Proof. Algorithm 1 guarantees that each sensor node transmits for exactly once and will not serve as a receiver again after transmission. Hence the resulting transmission links constitute a tree.

We further prove that $\frac{16}{3}X^2 + 12X + 7$ colors are enough to separate the cells with the same color by a distance of at least $2(X + 1)d$, where $d = 3^k$ is the side length of cells in category $k$. With the distance of $2(X + 1)d$, we can bound the cumulative interference at each receiver, and prove that each transmission is successful under the physical interference model by satisfying the SINR requirement. $\square$

Theorem 2 (Correctness of *NN-AS*). *The centralized* NN-AS *in Algorithm 3 can construct a data aggregation tree and correctly schedule the transmission under the physical interference model.*

Proof. Algorithm 3 guarantees that each node will be removed from the node set $V$ after selected for transmission and hence will be the transmitter for exactly once. At the end of each round, receivers and other non-scheduled nodes remain in $V$, and all aggregated data resides on the remaining nodes. Therefore, the generated transmission links correctly construct a data aggregation tree.

For link scheduling, Algorithm 3 applies the algorithm in [9], whose correctness under the physical interference model has been proven in [9]. $\square$

## 6.2 Aggregation Latency

We now analyze the latency bound and approximation ratio of the algorithms.

Theorem 3 (Aggregation Latency of *Cell-AS*). *The aggregation latency for the distributed* Cell-AS *in Algorithm 1 is upper bounded by $O(K)$, where $K$ is the link length diversity .*

Proof. We first bound the number of time slots for link schedule in each cell as a constant value. As the cells of the same link length category are colored with $\frac{16}{3}X^2 + 12X + 7$ colors, which is also constant, the aggregation latency for each category is bounded as constant. So the overall aggregation latency has an upper bound of $O(K)$. $\square$

Theorem 4 (Aggregation Latency of *NN-AS*). *The aggregation latency for the centralized* NN-AS *in Algorithm 3 is upper bounded by $O(\log^3 n)$.*

Proof. We first show that each node can be the nearest neighbor of at most 6 other nodes on a plane. Then, at least $\frac{1}{7}|V|$ nodes are removed from node set $V$ in each round of *NN-AS*. So we can prove that the data aggregation tree can be constructed with at most $\lceil \log_{\frac{7}{6}} n \rceil$ rounds in *NN-AS*.

After demonstrating that the link scheduling latency in each round of *NN-AS* is $O(\log^2 n)$, we have that, in total, *NN-AS* schedules the data aggregation in $O(\log^3 n)$. $\square$

Theorem 5 (Optimal Lower Bound). *The aggregation latency for the* MLAS *problem under any interference model is lower bounded by* $\log n$.

Proof. Under any interference model, as a node cannot transmit and receive at the same time, at most $\frac{|V|}{2}$ links can be scheduled for transmission in one time slot. Since each node only transmits for exactly once, at most $\frac{|V|}{2}$ nodes complete their transmissions in one time slot.

Suppose we need $k$ time slots to aggregate all the data. We have $\lceil \frac{n}{2^k} \rceil = 1$, and thus $k = \lceil \log n \rceil$, *i.e.*, the aggregation latency under any interference model is at least $\log n$. $\square$

As compared to the optimal lower bound, our distributed *Cell-AS* achieves an approximation ratio of $O(K/\log n)$, and the centralized *NN-AS* has an approximation ratio of $\frac{O(\log^3 n)}{\log n}$, which is equivalent to $O(\log^2 n)$. Note that $O(K)$ is between $O(\log n)$ and $O(n)$ based on the detailed analysis on the range of $K$ in technical report [6].

## 6.3 Comparison with Li et al.'s Algorithm in [7]

We next analytically compare our distributed *Cell-AS* with the distributed algorithm proposed by Li et al. [7] (referred to as *Li et al.'s algorithm* hereinafter), which is the only existing work addressing the *MLAS* problem under the physical interference model, as far as we are aware of.

*Li et al.'s* algorithm includes four consecutive steps,

—*Topology Center Selection*: the node with the shortest network radius in terms of hop counts is chosen as the topology center.

—*BFS Tree Construction*: using topology center as the root, BFS is executed over the network to build BFS tree.

—*Connected Dominating Set (CDS) Construction*: a CDS is constructed as the backbone of aggregation tree by an existing approach [12] based on BFS tree.

—*Link Scheduling*: the network is separated into grids with side length $l = \delta r/\sqrt{2}$, where $0 < \delta < 1$ is a configuration parameter, which is assigned before execution, and $r$ is the maximum achievable transmission range under the physical interference model with constant power assignment $P$ and $\frac{P/r^\alpha}{N_0} = \beta$. The grids are colored with $\lceil (\frac{4\beta\tau P\cdot l^{-\alpha}}{(\sqrt{2})^{-\alpha}P\cdot l^{-\alpha} - \beta N_0})^{\frac{1}{\alpha}} + 1 + \sqrt{2} \rceil$ colors and links are scheduled with respect to grid color. Here, $\tau = \frac{\alpha(1+2^{-\frac{\alpha}{2}})}{\alpha-1} + \frac{\pi 2^{-\frac{\alpha}{2}}}{2(\alpha-2)}$.

**Aggregation Latency**

*Li et al.'s* algorithm solves the *MLAS* problem in $O(\Delta+R)$ time slots, where $R$ is the network radius counted by node hops and $\Delta$ is the maximum node degree. In the worst case, either $R$ or $\Delta$ can be $O(n)$. And $R = O(\log n)$ in best case. Our *Cell-AS* achieves an aggregation latency of $O(K)$, which also equals to $O(n)$ in the worst case and $O(\log n)$ in the best case. Therefore the two algorithms share the same order of worst-case and best-case aggregation latency.

**Computational and Message Complexity**

*Cell-AS* can have an upper bound of $O(\min\{Kn, 13^K\})$ for both computational complexity and message complexity. Since $K = n$ in worst case, both computational complexity and message complexity are at most $O(n^2)$.

*Li et al.'s* algorithm has a computational complexity of $O(n|E|)$ and message complexity of $O(n + |E|)$. As $|E| =$

$n^2$ in worst case, *Li et al.'s* algorithm's computational and message complexity are $O(n^3)$ and $O(n^2)$ respectively.

We can have that *Cell-AS* has a better computational complexity while sharing the same order of message complexity with *Li et al.'s* algorithm. More detailed analysis by case study in technical report [6] demonstrates that *Cell-AS* outperforms *Li et al.'s* algorithm in its worst cases.

# 7. EMPIRICAL STUDY

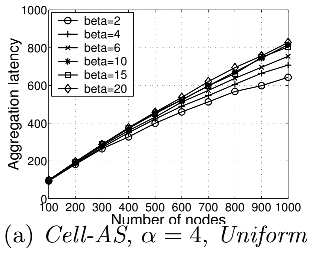
We have implemented our proposed distributed algorithm *Cell-AS*, centralized algorithm *NN-AS*, as well as *Li et al.*'s algorithm, and carried out extensive simulation experiments to verify and compare their efficiency empirically.

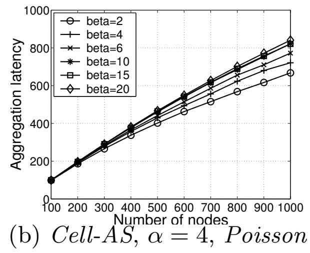
In our experiments, three types of sensor network topologies, namely *Uniform*, *Poisson* and *Cluster*, are generated with $n = 100$ to $1000$ nodes distributed in a square area of 40000 square meters. The nodes are *uniformly* randomly distributed in *Uniform* topologies, and are distributed with the *Poisson* distribution in *Poisson* topologies. In *Cluster* topologies [4], $n_C$ cluster centers are uniformly randomly located in the square and $\frac{n}{n_C}$ nodes are uniformly randomly distributed within the disk of radius $r_C$ centered at each cluster center. We use the same settings as in [4], $n_C = 10$ and $r_C = 20$, in our experiments. We set $N_0$ to the same constant value 0.1 as in [7] (which nevertheless would not affect the aggregation latency). The transmission power in our implementation of *Li et al.*'s algorithm is assigned the minimum value to maintain the connectivity of the respective network, while $\delta$ is set to 0.6 in compliance with the simulation settings in [7]. Since $2 < \alpha < 6$ and $\beta \geq 1$, we experiment with $\alpha$ set to 3, 4 and 5, and $\beta$ to values between 2 to 20, respectively. All our results presented are the average of 1000 trials.

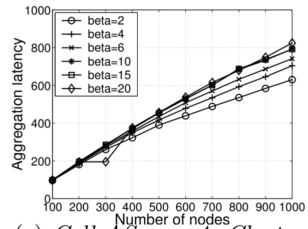
We first compare the aggregation latency among the three algorithms with different combinations of $\alpha$ and $\beta$ values in three types of topologies. The representative results at $\alpha = 4$ are presented in Fig. 4, and the complete sets of plots can be found in our technical report [6] due to space constraint.

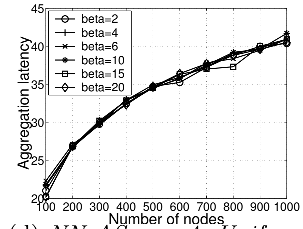
From our plots in Fig. 4 and [6], we observe that with *Cell-AS* algorithm, as expected, the aggregation latency is larger with smaller $\alpha$, which represents less path loss of power and thus larger interference from neighbor nodes, and larger $\beta$, corresponding to higher SINR requirement. However, similar latency performance is observed with *NN-AS*, at different values of $\alpha$ and $\beta$. This shows that network topology is the dominant influential factor to aggregation latency for *NN-AS*, given its *nearest-neighbor* mechanism in tree construction and *non-linear* power assignment [9] for link scheduling.

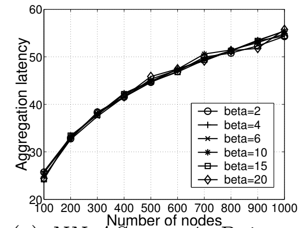
For *Li et al.*'s algorithm, from Fig. 4(g)–(j), we observe that most of the curves produced at different $\beta$ values are linear lines overlapping onto each other, except in the following cases with *Uniform* topologies: $\beta = 2$ when $\alpha = 4$ (Fig. 4(g)), $\beta = 2$, $\beta = 4$ and $\beta = 6$ when $\alpha = 5$ (Fig. 4(h)). The reason behind the linear overlapping lines is that each grid is scheduled one by one without any concurrency with *Li et al.*'s algorithm in cases of the *Poisson* and *Cluster* topologies, as well as the *Uniform* topologies with smaller $\alpha$ and larger $\beta$. The no-concurrency phenomenon can be further explained: Since the number of colors is $\lceil (\frac{4\beta\tau P \cdot l^{-\alpha}}{(\sqrt{2})^{-\alpha} P \cdot l^{-\alpha} - \beta N_0})^{\frac{1}{\alpha}} + 1 + \sqrt{2} \rceil$ with $l = \delta r/\sqrt{2}$, $\tau = \frac{\alpha(1+2^{-\frac{\alpha}{2}})}{\alpha-1} + \frac{\pi 2^{-\frac{\alpha}{2}}}{2(\alpha-2)}$ and $\frac{P/r^\alpha}{N_0} = \beta$ (See Sec. 6.C for description of *Li et al.*'s algorithm), smaller $\alpha$ and larger $\beta$ lead to a larger number of colors needed. On the other hand, in *Poisson* and *Cluster* topologies, the nodes are not evenly distributed, thus requesting a larger $r$ to maintain the network connectivity as well, which leads to a smaller number of grids since the side length of each grid is $\delta r/\sqrt{2}$. In these cases, the number of required colors in the algorithm, as decided by $\alpha$ and $\beta$, is larger than the total number of grids in the network (which is proportional to $1/r$). Therefore, each grid is actually scheduled one by one. In comparison, the number of cells in our *Cell-AS* is only related to the link length diversity but not $r$. Therefore, our algorithm has much more concurrency of link scheduling across different cells, leading to the sublinear curves in Fig. 4(a)–(c).

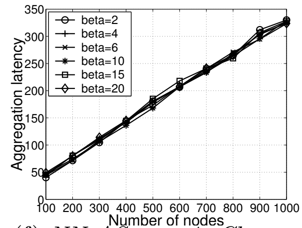

(a) *Cell-AS*, $\alpha = 4$, *Uniform*

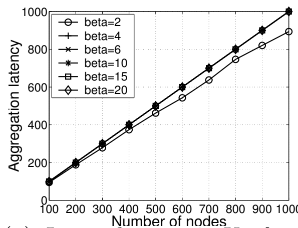

(b) *Cell-AS*, $\alpha = 4$, *Poisson*

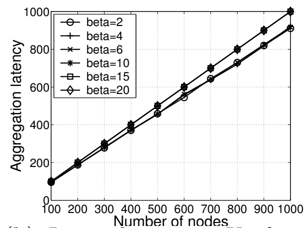

(c) *Cell-AS*, $\alpha = 4$, *Cluster*

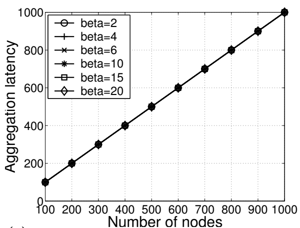

(d) *NN-AS*, $\alpha = 4$, *Uniform*


(e) *NN-AS*, $\alpha = 4$, *Poisson*


(f) *NN-AS*, $\alpha = 4$, *Cluster*


(g) *Li et al.*, $\alpha = 4$, *Uniform*


(h) *Li et al.*, $\alpha = 5$, *Uniform*


(i) *Li et al.*, $\alpha = 4$, *Poisson*

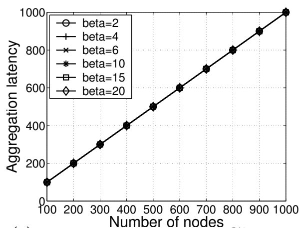
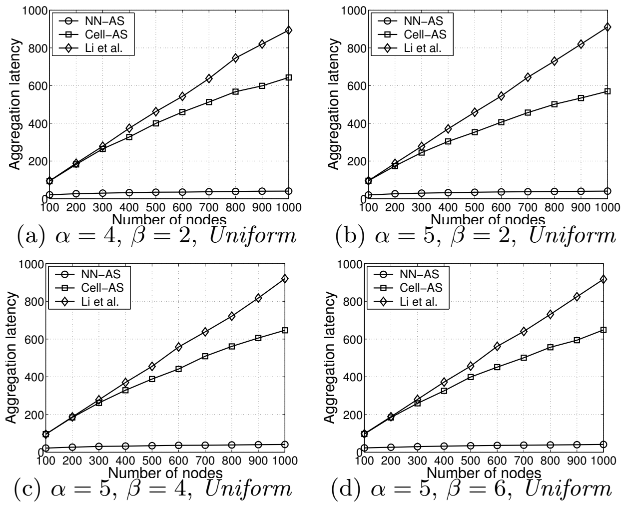

(j) *Li et al.*, $\alpha = 4$, *Cluster*

**Figure 4: Aggregation latency for three algorithms in different topologies.**




(a) $\alpha = 4$, $\beta = 2$, *Uniform* (b) $\alpha = 5$, $\beta = 2$, *Uniform*

(c) $\alpha = 5$, $\beta = 4$, *Uniform* (d) $\alpha = 5$, $\beta = 6$, *Uniform*

**Figure 5: Aggregation latency comparison among three algorithms in selected network settings.**

Fig. 4 shows show that concurrent link scheduling (across different cells/grids) occurs with all three algorithms only in four cases in the *Uniform* topologies: (1) $\alpha = 4$, $\beta = 2$; (2) $\alpha = 5$, $\beta = 2$; (3) $\alpha = 5$, $\beta = 4$; (4) $\alpha = 5$, $\beta = 6$. We next compare the aggregation latencies achieved by the three algorithms in those four cases. Fig. 5 shows that our centralized *NN-AS* achieves a much lower aggregation latency as compared to the other two algorithms, which remains at a similar level regardless of the network sizes. The performance of our distributed *Cell-AS* is similar to that of *Li et al.*'s algorithm where $n \leq 200$, but becomes up to 35% better than the latter when the network becomes larger.

# 8. CONCLUDING REMARKS

This paper tackles the minimum-latency aggregation scheduling problem under the physical interference model. Despite the abundant results on the *MLAS* problem under the protocol interference model, they are much less relevant to real networks than any solution under the physical model which is much closer to the physical reality. The physical model is favored also because of its potential to enhance the network capacity [9, 10, 11]. Although the physical model adds to the difficulty of a distributed solution for the problem, we propose a distributed algorithm to solve the problem in networks of arbitrary topologies. By strategically dividing the network into cells according to the link length diversity ($K$), the algorithm obviates the need for global information and can be implemented in fully distributed fashion. We also present a centralized algorithm that represents the current most efficient algorithm for the problem, as well as prove an optimal lower bound of the aggregation latency for the *MLAS* problem under any interference model. Our extensive analysis shows that the distributed algorithm aggregates all the data in $O(K)$ time slots (with approximation ratio $O(K/\log n)$ with respect to the optimal lower bound), and the centralized algorithm in at most $O(log^3 n)$ time slots (with approximation ratio $O(log^2 n)$). Our empirical studies under realistic settings further demonstrate that, both *Cell-AS* and *NN-AS* outperform *Li et al.*'s algorithm in all three topologies tested.

# 9. REFERENCES


[1] X. Chen, X. Hu, and J. Zhu. Minimum data aggregation time problem in wireless sensor networks. In *Proc. of MSN'05*, Dec. 2005.

[2] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local broadcasting in the physical interference model. In *Proc. of DIAL M-POMC'08*. ACM, Aug. 2008.

[3] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric sinr. In *Proc. of MOBIHOC'07*. ACM, Sept. 2007.

[4] O. Goussevskaia, R. Wattenhofer, M. M. Halldorsson, and E. Welzl. Capacity of arbitrary wireless networks. In *Proc. of INFOCOM'09*. IEEE, Apr. 2009.

[5] S. C.-H. Huang, P. Wan, C. T. Vu, Y. Li, and F. Yao. Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In *Proc. of INFOCOM'07*. IEEE, May 2007.

[6] H. X. Li, Q.-S. Hua, C. Wu, and F. C. M. Lau. Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks under Physical Interference Model. Technical report, Dept. of Computer Science, The Univ. of Hong Kong, TR-2010-07, Aug. 2010.

[7] X.-Y. Li, X. H. Xu, S. G. Wang, S. J. Tang, G. J. Dai, J. Z. Zhao, and Y. Qi. Efficient data aggregation in multi-hop wireless sensor networks under physical interference model. In *Proc. of MASS'09*. IEEE, Oct. 2009.

[8] M. Maroti, B. Kusy, G. Simon, and Á. Ledeczi. The flooding time synchronization protocol. In *Proc. of SenSys'04*. ACM, Nov. 2004.

[9] T. Moscibroda. The worst-case capacity of wireless sensor networks. In *Proc. of IPSN'07*. ACM/IEEE, Apr. 2007.

[10] T. Moscibroda and R. Wattenhofer. The complexity of connectivity in wireless networks. In *Proc. of INFOCOM'06*. IEEE, Apr. 2006.

[11] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Topology control meets sinr: The scheduling complexity of arbitrary topologies. In *Proc. of MOBIHOC'06*. ACM, May 2006.

[12] W. P.-J, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *Proc. of INFOCOM'02*. IEEE, Jun. 2002.

[13] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proc. of MOBICOM'03*. ACM, Sept. 2003.

[14] S. Schmid and R. Wattenhofer. Algorithmic models for sensor networks. In *Proc. of WPDRTS'06*. ACM, Apr. 2006.

[15] P.-J. Wan, S. C.-H. Huang, L. X. Wang, Z. Y. Wan, and X. H. Jia. Minimum-latency aggregation scheduling in multihop wireless networks. In *Proc. of MOBIHOC'09*. ACM, May 2009.

[16] X. H. Xu, S. G. Wang, X. F. Mao, S. J. Tang, and X.-Y. Li. An improved approximation algorithm for data aggregation in multi-hop wireless sensor networks. In *Proc. of FOWANC'09*. ACM, May 2009.

[17] B. Yu, J. Li, and Y. Li. Distributed data aggregation scheduling in wireless sensor networks. In *Proc. of INFOCOM'09*. IEEE, Apr. 2009.