Efficient Aggregation Scheduling in Multihop Wireless Sensor Networks with SINR Constraints

Xiaohua Xu, Xiang-Yang Li, *Senior Member, IEEE,* and Min Song, *Senior Member, IEEE*

Abstract—We study delay efficient data aggregation scheduling in wireless sensor networks subject to Signal to Interference-Plus-Noise Ratio (SINR) constraints. We construct a routing tree and propose two scheduling algorithms that can generate collision-free link schedules for data aggregation. We prove that the delay of each algorithm is $O(R + \Delta)$ time-slots, where R and Δ are respectively the graph radius and the maximum node degree in a reduced communication graph of the original network; the proposed algorithms are asymptotically optimum on delay in random wireless sensor networks. We evaluate the performances of the proposed algorithms and the simulation results corroborate our theoretical analysis.

Index Terms—Wireless sensor networks, aggregation scheduling, delay, SINR.

1 Introduction

A *wireless sensor network (WSN)* consists of small-sized and low-powered wireless nodes spreading over a geographical area which can collaborate with each other for control applications. In each application, there is usually a control center from which end users can query on sensory data within the network. The control center, having more computational ability than other wireless nodes, needs to gather sensory data from the network. In the process of data gathering, data may be compressed within the network to save energy. *Data aggregation* [5], [12] is a process in which information can be gathered and expressed in a summary form according to some aggregation function such as maximum, and/or sum. Data aggregation introduces a possibility of a new energy or time efficient method to gather data, in contrast to raw data gathering.

In most control applications, the time of utilizing the data are critical and a data aggregation task often comes with a stringent delay constraint imposed by applications. Here, the *delay* of data aggregation is the duration from the time when the first wireless node transmit for the task, to the time when the control center receives all (possibly aggregated) data. One promising way of minimizing the delay is to maximize the throughput while data transmissions in WSNs face a fundamental challenge, *i.e.*, the wireless interference. Previous work often focused on graph-based interference models in their protocol design for data aggregation [5], [12], [22], [27], [29]. Graph-based models serve as a useful abstraction of WSNs; they facilitate the process of designing protocols and proving their efficiency, while they cannot reflect the superimposed effect of wireless interference. Although the interference from one transmitter may be relatively small, the accumulated interference of several nodes can be sufficiently high to corrupt a transmission. Additionally, graph-based models are localized interference models and they simply neglect interference of nodes beyond a certain range. On the other hand, the physical model [8], [28] represents wireless interference more realistically and practically. Under this model, a signal is received successfully if the Signal to Interference-plus-Noise Ratio (SINR) is above a hardware-defined threshold. This definition of a successful transmission, accounts for interference generated by distant transmitters, thus can capture the interference between links more accurately.

For data aggregation with SINR constraints, we have to take care of superimposed interference. The effect of potential interference from far-away nodes makes it difficult to ensure that all active links satisfy the SINR constraints. Moreover, the notion of an interference edge is not a binary relation anymore. The SINR at each receiver node depends on which transmissions are being scheduled concurrently in each time slot. Thus, a simple conflict graph cannot be constructed without knowing the solution beforehand (actually the interference graph is a hyper-graph). This makes the analysis of algorithms more challenging than in graph-based models. In this work, we will study delay efficient data aggregation scheduling with SINR constraints. Given is a set of wireless nodes distributed in a plane, each node contains some data to report, the objective is to design routing and a node transmission schedule for data aggregation with SINR

- *X. Xu, M. Song are with the EECS department at the University of Toledo. E-mail: xiaohua.xu0@gmail.com; Min.Song@utoledo.edu*
- *X.-Y. Li is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616 USA. E-mail: xli@cs.iit.edu*

- *An earlier version [16] of this paper appeared in IEEE MASS 2009.*

constraints.

Our Contributions: We propose two algorithms which will rely on a concept of *reduced graph* of the original communication graph. In a reduced graph, we only consider short links and perform localized and interference-aware scheduling. Specifically, let r denote the maximum transmission range. We will only consider links $\overrightarrow{uv}$ with length at most δr, where $0 < \delta < 1$ is a constant. We analytically prove that the proposed algorithms can achieve a constant approximation ratio on the delay of data aggregation where the optimum is also computed using the reduced graph $G(V, \delta r)$.

We present a lower-bound on the delay of any algorithm in the reduced graph $G(V, \delta r)$. Notice that using links longer than δr, we may be able to reduce the delay of data aggregation (it is an open question at the current stage whether we can reduce the delay by more than a constant factor). In this work, we are able to prove that our method is asymptotically optimum for random WSNs where nodes are distributed uniformly at random.

In a random network, we prove that, by using only the reduced network $G(V, \delta r)$, our method can achieve a delay that is within a constant factor of the optimum delay achievable using all links from the original $G(V, r)$. On the negative side, we show that, given any constant $\delta < 1$, there is a deployment (or distribution) of n nodes V such that *any* algorithm for data aggregation scheduling using only links in $G(V, \delta r)$ has a delay of at least n, while the optimum delay using all links in $G(V, r)$ is only $O(\sqrt{n})$.

Finally, we perform extensive evaluations to show that our methods perform well in practice. To further maximize the throughput, we present a greedy method called *compressive scheduling*, where we schedule as many links as possible in each time-slot. We find that the method will almost halve the delay achieved by our first algorithm for sparse networks (with maximum node degree $\Delta \leq 25$).

The rest of the paper is organized as follows. Section 2 describes the system model and outlines the related work. Section 3.2 presents our two scheduling algorithms. Section 4 analyzes the performances. Section 5 discusses the overall lower-bound under our model. Furthermore, Section 6 provides the analytical results in randomly WSNs. Section 7 presents the simulation results. Section 8 concludes the paper.

2 Problem Definition, Related Work

Consider a set V of n nodes distributed in a two-dimensional plane where $v_s \in V$ is the sink node. Each node can send (receive) data to (from) all directions. Under the **physical interference model** [2], we assume all nodes have fixed transmission power P, and define $P_v(u) = P \cdot g(u, v)$ as the received power at the node v of the signal transmitted by the node u. Here $g(u, v) \leq 1$ is called the path-loss from node u to v. Set path-loss as $g(u, v) = \eta \cdot \|uv\|^{-\alpha}$ (this setting is justified in [26]), where the constant $\alpha \geq 2$ is the path-loss exponent, and $\|uv\|$ is the Euclidean distance between node u and v. A receiver node v can successfully receive a packet from a sender node u *iff* the Signal to Interference-plus-Noise Ratio (SINR) at node v is above a certain threshold $\beta > 0$:

$$\frac{P_v(u)}{\xi + \sum_{w \in S_u} P_v(w)} \geq \beta \tag{1}$$

Here $\xi \geq 0$ denotes the ambient noise, and S_u denotes the set of senders transmitting concurrently with sender u. In a given network, η is fixed. We can adjust the value of background noise (by setting ξ to be ξ/η) and omit η in the expression of path-gain.

We can compute the maximum transmission range under the physical interference model as $r = (\frac{P}{\xi\beta})^{\frac{1}{\alpha}}$. r is the maximum possible length of a communication link that can transmit alone successfully. From now on, for any $l \leq r$, we use $G(V, l)$ to denote the induced subgraph of $G(V, r)$ that has all edges (u, v) of length at most l. Observe that a long link with length comparatively close to r is not a good candidate in practice for transmission since the SINR at the receiver is very small. Even worse, it prevents many possible concurrent transmissions. Thus, we will only use some links that are smaller than r to some extent. Specifically, considering a small constant $\delta \in (0, 1)$, we can derive a subgraph (denote as $G(V, \delta r)$) with only links with length at most δr. We call this subgraph as a reduced graph (or network). Generally, the larger the value of δ, the reduced graph is connected with higher probability. r has an extremely large value. This means that the node degree is very large in the original graph $G(V, r)$. As long as the δ value is not too small, the reduced graph is still connected. In the case that the reduced graph $G(V, \delta r)$ is not connected, then $G(V, \delta r)$ consists of multiple connected components. We will remove the number of connected components by adding a shortest edge between a pair of connected components. We will continue such operations until there is exactly one connected component in $G(V, \delta r)$, consisting of the whole graph, this means that $G(V, \delta r)$ is connected.

In a reduced graph $G(V, \delta r)$, we review the definition of data aggregation scheduling [27], [29]. We assume a synchronous message passing model in which time is divided into slots; in each time-slot, a node $v \in V$ can send a message (data unit) to one of its neighboring nodes. Given a reduced graph $G(V, \delta r)$, each node has one data unit to send, assume A, B are two disjoint subsets of nodes in V, then data can be aggregated from A to B in one time-slot *iff* there exists a set of communication links between A and B from $G(V, \delta r)$ that can transmit concurrently, and at the same time, each node from A serves as a sender of some link incident to a node in B. Then, a valid aggregation schedule in $G(V, \delta r)$ with delay L can be defined as a sequence of sender sets $S_1, S_2, \cdots, S_L$ satisfying the following conditions:

1) $\cup_{i=1}^{L} S_i = V \setminus \{v_s\}$;
2) $S_i \cap S_j = \emptyset$, $\forall i \neq j$;
3) Data can be aggregated from S_i to $V \setminus \cup_{j=1}^{i} S_j$ in $G(V, \delta r)$ in one time-slot.

Notice that here $\cup_{i=1}^{L} S_i = V \setminus \{v_s\}$ is to ensure that all data will be gathered to sink v_s; $S_i \cap S_j = \emptyset$ ($\forall i \neq j$) is to ensure that every node participate in aggregation by using some *aggregation function* at most once. To simplify our analysis, we will relax the requirement that $S_i \cap S_j = \emptyset$, $\forall i \neq j$. When the sets S_i, $1 \leq i \leq L$ are not disjoint, in the actual data aggregation, a node v, that appears multiple times in $S_1, S_2, \cdots, S_L$, will participate in the data aggregation only once (say the smallest i when it appears in S_i), and then it will only serve as a relay node in later appearances.

Given a network consisting of n nodes V, each node has one unit of data to send to the sink node, the problem **Data Aggregation Scheduling** seeks a valid data aggregation schedule $\{S_1, S_2, \cdots, S_L\}$ in the graph $G(V, \delta r)$ with minimum delay L. Here, we assume the network $G(V, \delta r)$ is connected, otherwise the disconnected nodes cannot send their data to the sink node via the reduced graph. For simplicity, we define $\Delta(G), D(G), R(G)$ as the maximum degree, network diameter, network radius of the reduced graph $G(V, \delta r)$ respectively. Here the network diameter is the greatest hop distance between any pair of nodes. The network radius is the minimum eccentricity of any node where the eccentricity of a node $u \in V$ is the greatest hop distance between u and any other node.

Concept of Aggregation Function: Data aggregation functions can be classified into three categories: distributive (*e.g., maximum, minimum, sum, count*), algebraic (*e.g., minus, average, variance*) and holistic (*e.g., median, k^{th} smallest or largest*). Here we only focus on the distributive or algebraic aggregation functions. The detailed definition of aggregation function is available in [27].

Since Connected Dominating Set (CDS) will be used in our algorithm design, we briefly review its definition as follows. Please refer to a recent survey [1] and references therein for more details on CDS.

Concept of Connected Dominating Set: In a graph $G = (V, E)$, a subset V' of V is a dominating set (DS) if each node in V is either in V' or adjacent to some node in V'. Nodes in V' are called *dominators*, whereas nodes not in V' are called *dominatees*. A subset C of V is a CDS if C is a dominating set and C induces a connected subgraph. Consequently, the nodes in C can communicate with each other without using nodes in $V \setminus C$.

2.1 Related Work

The minimum delay data aggregation problem has been proved to be NP-hard [5]. Under the protocol interference model, the distributed aggregation scheduling (DAS) algorithms were proposed in [29]. DAS can generate a collision-free schedule that has a delay bound of $24D + 6\Delta + 16$, where D is the network diameter when the conflict-graph is the original UDG communication graph. This was recently improved to $16R + \Delta$ model by Xu *et al.* [27] and $\left(1 + (\log R/\sqrt[3]{R})\right) R + \Delta$ by Wan *et al.* [22] under the same network model. Li *et al.* also proposed a distributed implementation in [12] under protocol interference model. Wan *et al.* [23] considered group communication scheduling including aggregation scheduling under the physical interference model. Li *et al.* [15] studied the complexity of data aggregation in wireless networks.

We then review the related work of scheduling for link activities. There has been extensive work on developing efficient approximation algorithms for link scheduling to maximize throughput under the physical interference model [4], [8], [9], [21], [25]. The approximation algorithm in [21] can achieve a constant approximation bound for link scheduling with uniform power control under the physical interference model, *i.e.*, the number of scheduled links is at least a constant factor of the optimum. Link scheduling under other wireless interference models has also been studied such as [24], which considered the problem of max-throughput (or max-fairness) routing and an interference-aware link scheduling in wireless networks.

Recently, there are a series of work on stable link scheduling to maximize the throughput region such as [10], [11]. This work differs from theirs in two respects. First, we will focus on both the routing path construction and a very careful design of node activities in the routing structure. We can guarantee that the delay is at most a constant factor of network radius, and avoid the interference at the same time. Second, their work assumes that the communication requests along each link arrive continuously while we assume that each node has only one data unit to transmit.

3 Algorithm Design

In our algorithm design for data aggregation scheduling, we first construct a routing tree, and then design a schedule of link activities to minimize the delay.

3.1 Routing

The basic idea of constructing a routing tree relies on a *breadth-first-search (BFS)* tree in the reduced graph $G(V, \delta r)$. The routing tree has the following properties. First, the depth of this tree is within a small constant factor of the network radius $R(G)$. Second, each internal node will be connected to a constant number of other internal nodes. Similar to [27], we use the *topology center* of $G(V, \delta r)$ as the root of our BFS tree. Here a node v_0 is called the topology center in a graph G if $v_0 = \arg\min_v\{\max_u d_G(u, v)\}$, where $d_G(u, v)$ is the hop distance between nodes u and v in graph G. The usage of topology center can enable us to reduce the delay to a function of the network radius $R(G)$, instead of the network diameter $D(G)$. After the topology center gathers the data from all nodes, it relays the aggregated data to sink node via the shortest path (the bold line path in Fig. 1). This will incur an additional delay $d_G(v_0, v_s)$ of at most $R(G)$.

The routing tree construction begins with finding the topology center v_0 of $G(V, \delta r)$. We calculate the hop distances of shortest paths between every pair of nodes, and then for each node, we find the hop distance of shortest path to the farthest node. The node for which this value is minimal is the topology center of the graph $G(V, \delta r)$. We then perform BFS over $G(V, \delta r)$ to build the BFS tree T_G. The BFS begins at the root node and inspect all its neighboring nodes. Then each of the neighbor nodes, inspects their neighboring nodes which were unvisited, and so on. We next select a maximal independent set (MIS) of T_G by an existing approach described in [20].

We interconnect MIS by using some additional nodes to form a CDS. Note that in each layer of the BFS tree, there are some dominator(s) and some dominatee(s); each dominatee has at least one neighboring dominator in the same or upper level. Thus, every dominator (except the root) has at least one dominator in the upper level within two hops. Based on our observations, if every dominator transmits its data to some dominator in upper level within two-hops, all the data in the dominators can reach the root finally. From another point of view, considering dominators in the decreasing order of their levels, a dominator u in level i aggregates data from all dominators in level $i+1$ or $i+2$ that are within two-hops of u. This can ensure that all the data will be aggregated to the root (topology center) as well. We will interconnect MIS like follows, for each dominator, we use minimum additional nodes to interconnect all its two hop neighbors. Those additional nodes are called *connectors*. All above operations result in a CDS tree $\mathbf{G_c}$.

After that, for each dominatee v not in $\mathbf{G_c}$, we connect it to $\mathbf{G_c}$ by adding a link from v to one of its neighboring dominators. The resulted tree is the final routing tree T.

In the example in Fig. 1, assume v_s is the sink node. We first find the topology center as v_0. Then we construct a BFS tree with v_0 as the root node. In the BFS tree, assume $\{v_1, v_2, v_3\}$ lie in level 1; $\{v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_s\}$ lie in level 2. We then

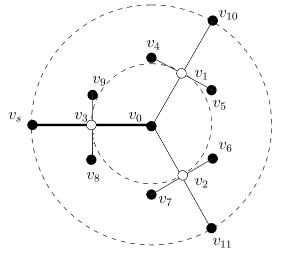


Fig. 1: Illustration of constructing a routing tree.

select a MIS as $\{v_0, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_s\}$. After that, we select additional nodes $\{v_1, v_2, v_3\}$ as connectors. Finally, all other dominatees (omitted in this figure) are connected to its nearest dominators. The above operations result in a routing tree.

3.2 Distributed Aggregation Scheduling

In this section, we present a distributed aggregation scheduling algorithm based on the routing tree T constructed in Section 3.1. Our algorithm consists of two phases: (1) dominators gather data from dominatees, (2) data gathering towards the sink node v_s.

In the first phase, we need to schedule single-hop data transmissions from dominatees to dominators; we will split this phase into several rounds. In each round, every dominator u selects a link $\overrightarrow{vu}$ from one of its neighboring dominatees to itself; we need to transmit the set of selected links (denoted as $\mathcal{L}$). Since each dominator has at most Δ neighboring dominatees, there are at most Δ rounds. Let

$$K = \lceil (\frac{4\beta\tau P \cdot \ell^{-\alpha}}{(\sqrt{2})^{-\alpha} P \cdot \ell^{-\alpha} - \beta\xi})^{\frac{1}{\alpha}} + 1 + \sqrt{2}\rceil$$

Here $\tau = \frac{\alpha(1+2^{-\frac{\alpha}{2}})}{\alpha-1} + \frac{\pi 2^{-\frac{\alpha}{2}}}{2(\alpha-2)}$, $\ell = \delta r/\sqrt{2}$. Next, we will show that each round costs at most K^2 time-slots, *i.e.*, all links from $\mathcal{L}$ can transmit successfully within K^2 time-slots.

Grid Partition and Coloring: To avoid the interference of data transmissions, a pair of links transmitting concurrently need to be separated well apart. Towards this end, we employ a grid partition of the deployment plane. The vertical lines $x = i \cdot \ell$ for $i \in \mathbb{Z}$ and horizontal lines $y = j \cdot \ell$ for $j \in \mathbb{Z}$ partition the planes into half-open and half-closed grids of side ℓ (here $\mathbb{Z}$ represents the integer set):

$$\{[i\ell, (i+1)\,\ell) \times [j\ell, (j+1)\,\ell) : i, j \in \mathbb{Z})\}.$$

Next, we color the grids such that one among every K^2 grids is assigned with the same color. We want to ensure that when at most one node from every grid with a monotone color transmits simultaneously, the transmissions are interference-free. We then index the colors used and denote σ_g as the color of grid g ($\sigma_g \in \{0, 1, \cdots, K^2-1\}$). For each link $\overrightarrow{up}$ in $\mathcal{L}$, let σ_g be the color index of the grid g where p lies, we then assign

Algorithm 1: Distributed Aggregation Scheduling

```
Input : sink $v_s$, topology center $v_0$, routing tree $T$, parameter $K$.
1  Partition the deployment plane into cells, each with side length $\delta r/\sqrt{2}$;
2  Color the grids such that one among every $K^2$ grids has the same color;
3  Each node $u$ initializes the value $\mathrm{NoC}[u]$, and $\mathrm{Type}[u]$ and $\mathrm{Level}[u]$;
4  Each node $u$ computes $\mathrm{Color}[u]$ from its location;
5  $\forall u \in V : \mathrm{TST}[u] \leftarrow 0$;
6  Each dominator $u$ sends a different number from $\{1, 2, \cdots, \mathrm{NoC}[u]\}$ to each leaf child in $T$;
7  if a leaf node $u$ received a number $N$ then
8      $\mathrm{TST}[u] \leftarrow (N-1) \cdot K^2 + \mathrm{Color}[u]$;
9  Each dominator sends a different number from $\{1, 2, \cdots, 12\}$ to each connector child in $T$;
10 Each connector sends a different number from $\{1, 2, 3, 4\}$ to each dominator child in $T$;
11 if a node $u$ received a number $N$ then
12     if $u$ is a dominator then
13         $TST[u] \leftarrow \Delta \cdot K^2 + 16K^2 \cdot \mathrm{Level}[u] + K^2 \cdot (N-1) + \mathrm{Color}[u]$;
14     if $u$ is a connector then
15         $TST[u] \leftarrow \Delta \cdot K^2 + 16K^2 \cdot \mathrm{Level}[u] + K^2 \cdot (N+3) + \mathrm{Color}[i]$;
16 Each node $u$ transmits data at time-slot $TST[u]$;
17 $v_0$ relays aggregated data to the sink $v_s$ via the shortest path.
```

a time-slot σ_g to transmit. Since each cell contains at most one dominator, we can finish the transmissions of $\mathcal{L}$ within K^2 time-slots.

We will proceed the second phase in the bottom-up manner, *i.e.*, from lower level to upper level. Every node will remain silent until the level where it locates begins running; when its turn comes, the dominator will try to gather all the data from other dominators in lower levels that have not been aggregated. This process consists of two steps: (1) every dominator aggregates its data to its corresponding connectors; (2) every connector transmits its data to the dominator in the upper level. To avoid the interference, we will use grid partition and coloring as well. We only let links with the same color transmit in a time-slot.

We present the details of our distributed aggregation scheduling in Algorithm 1. Each node u maintains some local variables in its buffer:

- Type: $\mathrm{Type}[u] \in \{L, D, C\}$, to indicate the type of the node u. The character 'L' represents leaf node in the routing tree T, 'D' represents dominator, and 'C' represents connector.
- Level: $\mathrm{Level}[u] \in \mathbb{N}$, to indicate the level of the node u in the BFS tree.
- Color: $\mathrm{Color}[u] \in \{0, 1, \cdots, K^2 - 1\}$, to indicate the color of the grid where the node u lies.
- Number of Children: $\mathrm{NoC}[u]$, which is the number of children nodes of u in tree T.
- Time Slot to Transmit: $\mathrm{TST}[u]$, which is the assigned time-slot that node u indeed sends its data to its parent.

The TST of all nodes are initialized to 0. By running the algorithm, the TST of all nodes are set gradually. If each node transmits at the time-slot equal to TST, the sink node will receive the aggregated data of all nodes correctly.

3.3 Improved Aggregation Scheduling

In this section, we will present a scheduling algorithm (noted as "Improved Algorithm") that will greatly improve the delay of our distributed algorithm. Similar to our distributed algorithm, Improved Algorithm consists of two phases based on the routing tree T constructed in Section 3.1. In the first phase, every dominator aggregates the data from all its dominatees, we will use the same method as that of our distributed algorithm. In the second phase, dominators aggregate their data towards the sink node v_s via the routing tree; we will propose a new method for the second phase.

The main idea of the second phase is to proceed data transmissions level by level in the routing tree. For each level, the dominators at this level will try to gather all the data from other dominators in lower levels that have not been aggregated. This process consists of two steps: (1) every dominator aggregates its data to its corresponding connectors. We apply grid partition and coloring method which has already been presented in detail in Section 3.2. For each dominator, we will assign its time-slot to transmit based on its own color (the color of the grid containing this dominator). Observe that each grid contains at most one dominator, thus two dominators of the same color can transmit concurrently, irrespective of their receives. We will prove that this step costs at most $(K+3)^2$ time-slots; (2) every connector transmits its data to the dominator in the upper level. In this step, for each connector, we will assign its transmission time based on the color of one of its children instead of itself. We will prove that this step costs at most $(K+3)^2$ time-slots as well. Thus, each level costs at most $2\,(K+3)^2$ time-slots.

Following our main idea, the assignment of transmission time can be described as follows. For each dominator u, let i be the level of the node u, and let σ_g be the color index of the grid g where u lies, we assign the time-slot $(K+3)^2\,\big(2(R-i)\big) + \sigma_g$ to transmit; For each connector u, let i be the level of u, we assign the time-slot of $(K+3)^2\,\big(2(R-i)+1\big) + \sigma_g$ to transmit. Here σ_g is the color of the connector u's one child. If the connector u has more than one children, we

Algorithm 2: Improved Aggregation Scheduling

```
Input : sink $v_s$, topology center $v_0$, routing tree $T$, parameter $K$.
1  Partition the deployment plane into cells, each with side length $\delta r/\sqrt{2}$;
2  Color the grids such that one among every $(K+3)^2$ grids has the same color;
3  Unmark all dominatees;
4  while unmarked dominatee(s) exist do
5      for each dominator $u$ do
6          select one link from $u$'s unmarked neighboring dominatee to $u$;
7      All the selected links form a set $\mathcal{L}$;
8      for $j = 1, \cdots, (K+3)^2$ do
9          all links in $\mathcal{L}$ of the $j$-th color transmit;
10     Mark all dominatees incident to links in $\mathcal{L}$;
11 for each node $u$ in CDS do
12     $i \leftarrow$ the level of $u$ in tree $T$;
13     if $u$ is a dominator then
14         $\sigma_g \leftarrow$ the color of grid $g$ containing $u$;
15         $u$ transmits at time-slot
16         $(K+3)^2\big(2(R-i)\big) + \sigma_g$ ;
17     if $u$ is a connector then
18         $v \leftarrow$ one child of $u$ in tree $T$;
19         $\sigma_g \leftarrow$ the color of grid $g$ containing $v$;
20         $u$ transmits at time-slot
           $(K+3)^2\big(2(R-i)+1\big) + \sigma_g$;
21 $v_0$ relays the aggregated data to the sink $v_s$ via the shortest path.
```

select only one as u's *representative child* and find its color index as σ_g. This finishes the second phase. The details are shown in Algorithm 2. We will prove in Section 4.2 that this assignment of transmission time-slots achieves better bounded delay compared with our distributed algorithm (Algorithm 1).

4 Upper bound on Latency

First, we prove that both of our algorithms can avoid interference. Theorem 1 formally gives a feasible value of K to guarantee the correctness of simultaneous transmissions of at most one node in each cell with the same color.

Theorem 1: After the plane is divided into cells, we can set $K = \lceil (\frac{4\beta\tau P\cdot\ell^{-\alpha}}{(\sqrt{2})^{-\alpha}P\cdot\ell^{-\alpha}-\beta\xi})^{\frac{1}{\alpha}} + 1 + \sqrt{2}\rceil$ to ensure that if at most one node in each cell with the same color transmit, the transmissions are interference-free. Here $\tau = \frac{\alpha(1+2^{-\frac{\alpha}{2}})}{\alpha-1} + \frac{\pi 2^{-\frac{\alpha}{2}}}{2(\alpha-2)}$, $\ell = \delta r/\sqrt{2}$ (The proof is available in the conference version [16] of this paper).

4.1 Analysis of Distributed Algorithm

Lemma 1: Algorithm 1 can avoid interference.

Proof: In the first phase, in each time-slot, since at most one node in each cell with the same color transmit, the transmissions are interference-free by Theorem 1.

In the second phase, if u is a dominator, its transmission time is $\Delta\cdot K^2 + 16K^2\cdot \text{Level}[u] + K^2\cdot(N-1) + \text{Color}[u]$; if u is a connector, its transmission time is $\Delta\cdot K^2 + 16K^2\cdot\text{Level}[u] + K^2\cdot(N+3) + \text{Color}[i]$. If two nodes transmit at the same time-slot, they must lie in the same level. If u is a dominator, $K^2\cdot(N-1) < 4K^2$; if u is a connector, $K^2\cdot(N+3) \geq 4K^2$, thus the transmissions of dominators will not interleave with the transmissions of connectors. If two dominators transmit at the same time, they must receive the same number N and have the same color $\text{Color}[u]$, then they can transmit successfully according to Theorem 1. If two connectors transmit at the same time, they must receive the same number N and have the same color, then they also can transmit successfully according to Theorem 1. □

Next, we prove that the delay achieved by Algorithm 1 is $O(R+\Delta)$ where R is the network radius and Δ is the maximum node degree of $G(V,\delta r)$.

Lemma 2: The first phase of Algorithm 1 (and Algorithm 2) costs at most $K^2\cdot\Delta$ time-slots.

Proof: Since the maximum degree of each dominator is Δ, there are at most Δ rounds for aggregating data to dominators. We only need to prove that each round cost at most K^2 time-slots. Since the diameter of each cell is δr, at most one dominator falls in each cell. In each round, according to Theorem 1, every cell can be scheduled at least once in every K^2 time-slots. □

We then bound the number of connectors that a dominator u will use to connect to all dominators within two-hops. Our proof is based on a technique lemma implied from lemmas proved in [19].

Lemma 3: [19] Suppose that dominator v and w are within 2-hops of dominator u, v' and w' are the corresponding connectors for v and w respectively. Then either $|wv'| \leq 1$ or $|vw'| \leq 1$ if $\angle vuw \leq 2\arcsin\frac{1}{4}$.

Lemma 4: In the routing tree T, the following geometric facts are true:

1) each dominator has at most 12 neighboring connectors.
2) each connector has at most 4 children.

Proof: Let us verify the first fact first.

We first delete all the redundant connectors. We then claim that for each connector, there exists at least one dominator which has only this connector as its adjacent connector. This claim can be proven by contradiction as follows: assume there exists a connector u such that each of its adjacent dominators is also a neighbor of some other connector, then we can always delete this redundant connector u. This implies that if there are 13 connectors for one dominator, we must have at least 13 non-sharing dominators. By Lemma 3, we know that there are at least two dominators will share a same connector which contradicts to fact that all of those 13 dominators should be non-sharing with each other.

We then verify the second fact. This lemma can be proven by contradiction. Assuming there exists a connector u which has at least six neighboring dominators, then there exist two dominators v and w such that $\angle vuw \leq 60°$, this contradicts to the fact that no two dominators are adjacent to each other. Then each connector u has at most five neighboring dominators; as one neighboring dominator is the connector u's parent in the routing tree T, u has at most 4 children in the routing tree T. This finishes the proof. □

Lemma 5: In the second phase of Algorithm 1, the sink can receive all the aggregated data in at most $16K^2 \cdot R + R$ time-slots.

Proof: For each node u, its time to transmit is either $\Delta \cdot K^2 + 16 \cdot \text{Level}[u] + K^2 \cdot (N-1) + \text{Color}[u]$ (u is a dominator) or $\Delta \cdot K^2 + 16 \cdot \text{Level}[u] + K^2 \cdot (N+3) + \text{Color}[i]$ (u is a connector). Thus delay is upper-bounded by $16 \cdot R + 16K^2$ Since the root v_0 will relay the result to the sink via the shortest path, this costs an additional time of at most R time-slots. □

Lemma 5 implies the following theorem.

***Theorem** 2:* The delay of Algorithm 1 is at most $K^2 \cdot \Delta + 16K^2 \cdot R + R$.

4.2 Analysis of Improved Algorithm

Next, we upper-bound the delay achieved by Algorithm 2.

Lemma 6: Algorithm 2 can avoid interference.

Proof: In the first phase, in each time-slot, since at most one node in each cell with the same color transmit, the transmissions are interference-free.

In the second phase, the transmissions of dominators will not interleave with that of connectors. Thus, the transmission of any dominator will not interfere with the transmission of any connector. If two dominators transmit at the same time-slot, we have $(K+3)^2\left(2(R-i)\right) + \sigma_g = (K+3)^2\left(2(R-i')\right) + \sigma'_g$. Here i and i' are the levels of these two dominators respectively, σ_g and σ'_g are the colors of these two dominators respectively. Then these two dominators lie either in two different cells with the same color, or in the same cell. In the former case, they can transmit concurrently. In the latter case, these two dominators lie in the same cell with diameter one, which causes contradiction.

If two connectors transmit at the same time-slot, we have $(K+3)^2\left(2(R-i)+1\right) + \sigma_g = (K+3)^2\left(2(R-i')+1\right) + \sigma'_g$. Here i and i' are the levels of these two connectors respectively, σ_g and σ'_g are the colors of these two connectors' representative children respectively. Then these two connectors' children lie either in the same cell or in different cells with the same color. The former case causes contradiction immediately as these two connectors' children are both dominators. In the latter case, these two connectors' children must be separated by at least $(K+3)$ grids, then these two connectors must be separated by at least K grids, they can transmit according to Theorem 1. □

We then upper-bound the number of time-slots needed for the second phase.

Lemma 7: In the second phase of Algorithm 2, the sink can receive all the aggregated data in at most $2(K+3)^2 \cdot R$ time-slots.

Proof: The maximum value of $2((K+3)^2(R-i) + \sigma_g) + 1$ is $(K+3)^2 \cdot R$. Since the root v_0 will relay the result to the sink via the shortest path, this costs at most R time-slots. The second phase costs at most $2(K+3)^2 \cdot R + R$ time-slots. □

***Theorem** 3:* The delay of Algorithm 2 is at most $(K+3)^2 \cdot \Delta + 2(K+3)^2 \cdot R + R$.

5 Overall lower-bound on delay

In this section, we discuss the overall lower-bound on the delay for data aggregation under the physical interference model, for a reduced graph $G(V, \delta r)$. An overall lower-bound is the minimum time-slots (which may not be achievable) needed to finish this data aggregation communication task by any possible algorithm.

Lemma 8: Under the physical interference model, in a reduced graph $G(V, \delta r)$, it requires at least R time-slots for any algorithm to finish the aggregation transmission. Here R is the maximum hop distance for any node to the sink node in $G(V, \delta r)$.

Proof: We can see that the scheduled links can only be selected from the edges in $G(V, \delta r)$. It requires at least R time-slots for the farthest node v to transmit its data to the sink node v_s. This finishes the proof. □

Lemma 9: Under the physical interference model, in a reduced graph $G(V, \delta r)$, there are at most $\omega = \frac{(3\delta r)^\alpha}{\beta} - 1 = 3^\alpha \delta^\alpha \frac{\xi}{P} - 1$ senders transmitting concurrently which are all neighbors of a single node.

Proof: Assume by contradiction that there exist ω' senders transmitting simultaneously which are all neighbors of a single node u. Here $\omega' > \omega$, thus $\omega' \geq \frac{(3\delta r)^\alpha}{\beta}$. Consider the sender s_i whose incident link l_i is the shortest among all links of the senders, we assume its corresponding receiver is r_i, we have

$$\begin{aligned}\|s_j r_i\| &\leq \|s_j s_i\| + \|s_i r_i\| \leq \|s_j u\| + \|u s_i\| + (\delta r)\\ &\leq (\delta r) + (\delta r) + (\delta r) = 3(\delta r).\end{aligned}$$

Then, the interference experienced by node r_i is at least $SINR_{S_{\omega'}}(l_i) = \frac{\frac{P}{\|l_i\|^\alpha}}{\sum_{s_j \in S_{\omega'}} \frac{P}{\|s_j r_i\|^\alpha} + \xi} \leq \frac{P}{\sum_{s_j \in S_{\omega'}} \frac{P}{\|s_j r_i\|^\alpha} + \xi} < \frac{P}{\sum_{s_j \in S_{\omega'}} \frac{P}{\|s_j r_i\|^\alpha}} \leq \frac{1}{\sum_{s_j \in S_{\omega'}} \frac{1}{\|s_j r_i\|^\alpha}} \leq \frac{1}{\omega' \cdot \frac{1}{(3\delta r)^\alpha}} \leq \beta$. This contradicts the fact that l_i can transmit subject to SINR constraints. □

Corollary 1: Under the physical interference model, in a reduced graph $G(V, \delta r)$, it requires $\frac{\Delta}{\omega}$ time-slots for any algorithm to finish the aggregation transmission.

Proof: We consider the node v whose degree reaches maximum value Δ (if there are multiple such nodes, choose one randomly). Since for every neighboring node of v, it needs to report its data to the sink. This means every neighboring node of v needs to transmit at least once. By Lemma 9, at each time-slot, at most ω neighboring node can send concurrently, thus it requires at least $\frac{\Delta}{\omega}$ time-slots to finish aggregation scheduling. $\square$

Lemma 8 and Corollary 1 together imply an overall lower-bound for data aggregation scheduling as follows.

***Theorem** 4:* Under the physical interference model, for any $\delta < 1$, in a reduced graph $G(V, \delta r)$, it requires $\max\{R, \Delta/\omega\}$ time-slots for *any* algorithm to finish the aggregation transmission. Here R is the maximum hop distance from any node to the sink node in the graph $G(V, \delta r)$ and Δ is the maximum node degree of $G(V, \delta r)$.

Theorem 4 means that our algorithm can achieve the asymptotically optimum delay for data aggregation when we can only use links with length at most δr.

6 Approximability of Reduced Graph

In this section, we compare the performance of our algorithm with the optimum solution when all links in the network could be used for data aggregation. We first study the case when all wireless nodes V are randomly placed (random network), which happens in most scenarios for real wireless sensor networks. We then study the case when V are arbitrarily placed (arbitrary network).

6.1 Random Network

For a random network [13], [14], we assume that all n nodes are randomly deployed in a square region of side-length A. It is well-known that to ensure that the random network is connected with high probability, the degree of each node should be in the order of $\Omega(\log n)$. Then the side-length A is assumed to be in the order $A = O(\frac{n}{\log n})$ here.

We first review the following lemma.

Lemma 10: [13] Assume there is a graph with n nodes randomly deployed in a square region with side-length $O(\frac{n}{\log n})$. We partition the deployment square into cells, each of side-length a constant a. Then there is a sequence of $\delta(n) \to 0$ such that

$$\Pr\Big(\text{every cell contains a node}\Big) \geq 1 - \delta(n)$$

In a topology graph G, the hop distance $h_G(u, v)$ between a pair of nodes u and v is defined as the smallest number of hops between them. In a random network, if we connect any pair of wireless nodes with distance smaller than δr and r respectively, we get two topology graphs $G(V, \delta r)$ and $G(V, r)$.

Lemma 11: For any pair of nodes u and v, *w.h.p*, we have $h_{G(V,\delta r)}(u, v) \leq \rho \cdot h_{G(V,r)}(u, v)$, where ρ is a constant.

Proof: In a graph $G(V, \delta r)$, any pair of nodes u and v from two adjacent cells (sharing a common side) can communicate with each other directly. Then by choosing one node from each cell which is crossed by segment uv, we can connect u and v using $\Theta(\|uv\|/r)$ hops.

Similarly, if we partition the network into squares with side-length $\delta r/\sqrt{5}$. Then together with the assumption that the random network is connected with high probability, it follows that by only using the links with length δr, we can connect any two vertices u and v within $\Theta(\|uv\|/\delta r)$ hops. Notice that $\Theta(\|uv\|/\delta r) = \Theta(\|uv\|/r)$. It implies that in a random network, the minimum number of hops used to connect any two vertices in $G(V, r)$ is in the same order of the minimum number of hops used in $G(V, \delta r)$. $\square$

***Theorem** 5:* If there exists an algorithm for aggregation scheduling in $G(V, \delta r)$ with approximation ratio a, then we can achieve a solution for the original topology $G(V, r)$ with approximation ratio $\rho \cdot a$ *w.h.p.*

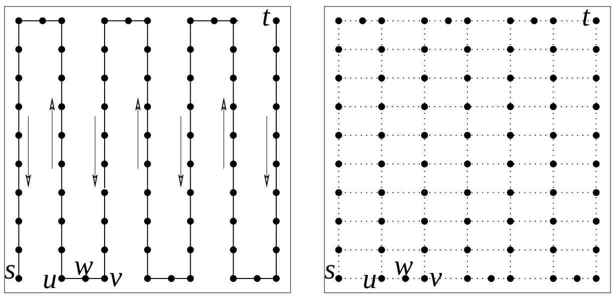
Proof: Let $OPT(G(V, r)), OPT(G(V, \delta r))$ be the optimum schedules for $G(V, r)$ and $G(V, \delta r)$. We overload the terms as the delay achieved using the schedules correspondingly.

By Lemma 11, the hop distance in the graph $G(V, \delta r)$ is at most ρ times of the hop distance in the original graph $G(V, r)$. So $R_{\delta r} = \max_{u \in V} \min h(u, v_s)$ of the graph $G(V, \delta r)$ is at most ρ times of $R_r = \max_{u \in V} \min h(u, v_s)$ of the original graph $G(V, r)$. Here $h(u, v_s)$ is the number of hops between the node u and the sink node v_s. By Lemma 8, R is a lower bound of the data aggregation under the physical interference model. At the same time, the maximum degree $\Delta_{\delta r}$ of graph $G(V, \delta r)$ is at most the maximum degree Δ_r of graph $G(V, r)$. In Corollary 1, we have proved that Δ is another lower bound of the data aggregation under physical interference model. Then $OPT(\delta r) \leq \rho \cdot OPT(r)$, which implies $ALG(\delta r) \leq a \cdot OPT(\delta r) \leq \rho \cdot a \cdot OPT(r)$. $\square$

6.2 Arbitrary Network

Theorem 5 shows that for random wireless sensor networks, the optimum solutions for the reduced graphs $G(V, \delta r)$ and the original communication graph $G(V, r)$ are in the same order. One may conjecture that this nice property holds for an arbitrary network. Unfortunately, Fig 2 gives a counter-example where the delay of data aggregation by using these two graphs will be very different when $0 < \delta < 1$ is a given constant.

In Fig 2, there are n nodes deployed in a rectangle region. There are $\sqrt{n}$ vertical evenly spaced lines with



Graph $G(V, \delta r)$ Graph $G(V, (\delta + \epsilon)r)$

Fig. 2: A network example: the optimum solutions for $G(V, r)$ and $G(V, \delta r)$ are not in the same order. The distance d_h between consecutive vertical lines is $(\delta + \epsilon)r$. The distance between two consecutive nodes on a vertical line is $d_v = \delta r$.

Euclidean distance $d_h = (\delta + \epsilon)r$ between consecutive lines. Here $\delta < 1$ is a given constant and $0 < \epsilon < 1-\delta$. For example, we can set $\epsilon = \min(\delta, \frac{1-\delta}{2})$. For each such line, we evenly place $\sqrt{n}$ nodes with Euclidean distance $d_v = \delta r$ between consecutive nodes. Additionally, $\sqrt{n} - 1$ nodes, like w between u and v in the example, act as the bridges to guarantee the network connectivity when doing aggregation in graph $G(V, \delta r)$. Distance between w and u (or v) is smaller or equal to δr, and distances between w and other nodes are larger than δr. Thus, in graph $G(V, \delta r)$, if we want to aggregate the data from s to t, we should strictly follow the dashed path as shown in Fig 2. The delay of data aggregation in $G(V, \delta r)$ thus is at least n. On the other hand, in graph $G(V, r)$, the red solid path may be a possibility for getting data from a source node s to the sink node t. It is easy to show that for graph $G(V, r)$, the network radius is in the order of $\Theta(\sqrt{n})$. On the other hand, the graph $G(V, r)$ contains the graph $G(V, (\delta+\epsilon)r)$ as a subgraph. Notice that, for the graph $G(V, (\delta + \epsilon)r)$, the maximum node degree is $\Delta \leq 10$ and the radius of the network is at most $R \leq 2\sqrt{n}$. Then using our scheduling algorithm, for graph $G(V, (\delta + \epsilon)r)$, the delay of data aggregation is at most $O(\Delta + R) = O(\sqrt{n})$. Consequently, the delay of data aggregation in the original network $G(V, r)$ is at most $O(\sqrt{n})$. Thus, we have the following lemma.

Lemma 12: For any constant $0 < \delta < 1$, there are examples of networks with n nodes V in two-dimensional such that the delay using only links in $G(V, \delta r)$ is at least n while the delay using all links in $G(V, r)$ is only $O(\sqrt{n})$ using Algorithm 1. In other words, there is no universal constant δ that ensures a constant approximation ratio for the delay of data aggregation by *any* algorithm that only considers links in $G(V, \delta r)$.

We then show that the ratio of the delay for data aggregation in $G(V, \delta r)$ over the delay for data aggregation in $G(V, r)$ is at most $O(n^{2/3})$ for any network of n nodes V in a two-dimensional space. Let us consider any n nodes V distributed in a two-dimensional region. Let Δ and R be the maximum node degree and radius for network $G(V, r)$ respectively. Obviously, we have $\pi R^2 \cdot \Delta \geq n$. This implies that $\max(R, \Delta) \geq \pi^{1/3} n^{1/3}$. Thus, the delay of data aggregation in $G(V, r)$ is at least $\pi^{1/3} n^{1/3}$. On the other hand, the delay of data aggregation in $G(V, \delta r)$ is at most n. Then we have the following theorem.

***Theorem** 6:* The ratio of the delay for data aggregation in $G(V, \delta r)$ over the delay for data aggregation in $G(V, r)$ is at most $n^{2/3}/\pi^{1/3}$ for any network of n nodes V in a two-dimensional space.

This implies that our algorithm achieves a delay that is at most $O(n^{2/3})$ factor of the delay by the optimum algorithm for $G(V, r)$. Lemma 12 implies that there are network examples such that the delay achieved by our algorithm is at least $\Omega(n^{1/2})$ factor of the delay by the optimum algorithm for $G(V, r)$. Note that there is a gap between the performance bound of our algorithm for network $G(V, r)$. We conjecture that

Conjecture 1: The delay achieved by our algorithm is $\Theta(n^{1/2})$ factor of the delay by the optimum algorithm for $G(V, r)$.

7 Performance Evaluations

In this section, we evaluate the performances of our proposed algorithms, *i.e.*, to measure the delay for the sink node to compute the aggregation result (*max, sum or average*) of all required data by using our proposed data aggregation scheduling algorithms. Note that here *delay* is defined as the number of time-slots needed to aggregate all required data from within the network to the sink node. We will compare the performances of Algorithm 1 and 2 with another method *Compressive Scheduling* which is described as follows.

Compressive Scheduling: This is in fact a greedy algorithm and it tries to schedule as many links as possible in every single time-slot. It consists of 5 steps:

1) find all leaf nodes, say N, in the same routing tree constructed in Section 3.1.
2) partition the deployment region into cells using the same method in Section 3.2.
3) find a subset $N_1 \subseteq N$ such that each cell contains at most one node from N_1. If a cell contains multiple nodes from N, we choose the one with the largest level in the BFS tree.
4) for each node $u \in N_1$, we find the corresponding link from u to u's parent in the routing tree. All the links found form a link set $\mathcal{L}$.
   - color the links in $\mathcal{L}$ by using the same method in Section 3.2, find a subset $\mathcal{L}_1$ of links of monotone color with the largest size. We then try to insert links into $\mathcal{L}_1$ greedily subject to SINR constraints (Equation (1));
   - apply Algorithm 1 in [25] to find a feasible solution $\mathcal{L}_2$ which is a subset of $\mathcal{L}$;

parameter	value	parameter	value
β	1	α	2.5
ξ	0.1	P	15

TABLE 1: Parameters of physical interference model.

- select one set from $\mathcal{L}_1$ and $\mathcal{L}_2$ with a larger size and let all links in this set transmit concurrently. Delete all transmitters which are leaf nodes in the routing tree. Update the routing tree accordingly.

5) repeat step 1) – 4) until there is no leaf node in the routing tree.

Simulation Setup: We deploy a set of nodes $\{v_1, \cdots, v_n\}$ uniformly (n varies from 200 to 2000 with step 100) in a square region with size $500meter \times 500meter$ (note that we always keep connectivity of the networks). We assume that every node in the network knows its own geometric information.

We will focus on physical interference model with the parameter setting listed in Table 1. Based on the parameters, we can derive the value of the maximum transmission radius r. For any pair of nodes u and v, we draw a feasible link (u, v) if $|uv| \leq \delta r$, where $|uv|$ is the distance between u and v. Note that we use a crucial parameter δ here. Generally, the smaller the value of δ, the less links is contained in the corresponding reduced network $G(V, \delta r)$, which implies that:

1) each node has less incident links (edges), thus the maximum node degree Δ becomes smaller.
2) the maximum length of links in G becomes smaller, thus the network radius R becomes larger.

Since the time spent (denoted as l_1) for collecting data to dominators increases monotonically with Δ. Thus l_1 becomes smaller as δ becomes smaller. The time spent (denoted as l_2) for collecting data from dominators to dominators level by level increases monotonically with R. Thus l_2 becomes larger as δ becomes smaller. To sum l_1 and l_2 as the total delay, when δ become smaller, it is not clear whether the total delay will decrease or increase. Thus, it is not straightforward for the effect of δ. However, δ cannot be too small or too large (close to 1). If δ is too small, the reduced graph may not be connected, thus some un-connected node cannot send its datum to the sink node. On the contrary, if δ is close to 1, then the network is strongly connected by a lot of long links. These long links prevent the simultaneous transmissions, which potentially increases the delay. From this perspective, a link with smaller length is preferred for scheduling.

The link scheduling mechanism is based on the TDMA manner. Basically, each node will locally broadcast TDMA scheduling information for each link it has to all its local neighboring nodes through beacon messages.

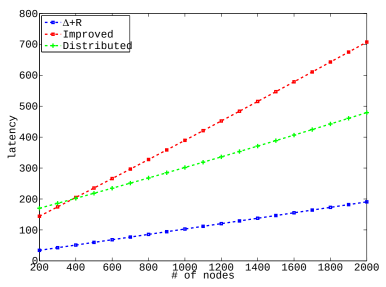
Performance Comparisons: We will call Algorithm 1 and 2 as Distributed Algorithm and Improved Algorithm respectively. To evaluate the effect of maximum node degree Δ, we fix the network deployment area as $(500m \times 500m)$ and vary the network size (# of nodes) from 200 to 2000 with step 100. By connecting every pair of nodes with distance no larger than δr, we obtain a reduced graph $G(V, \delta r)$. Here we set $\delta = 0.6$ and ensure that the corresponding reduced graph is connected. Later we will consider different values of δ. By fixing the size of network deployment area, we find that the network radius R is nearly fixed (around 25). At the same time, the maximum node degree (Δ) increases monotonously with network size. We measure the worst case performances of our proposed algorithms under this condition, the average result is shown in Figure 3(a). Observe that the delay increases monotonously with Δ.

To evaluate the effect of network radius R, we consider a network instance similar to the first one with two modifications: we set $\delta = 0.1$ and vary the network size from 200 to 2000 while fix the maximum node degree in the network deployment area. We find that Δ is nearly fixed (around 25) as well in the corresponding reduced graph. At the same time, R increases monotonously with the network size. We measure the worst case performances of Improved and Distributed algorithms under this condition, the average result is shown in Figure 3(b). Observe that the delay of Distributed Algorithm increases faster.

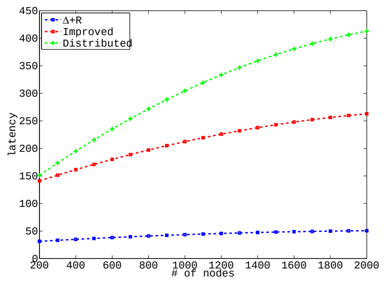
For average performance comparisons of different methods, we run simulations and compare their delays. Note all comparisons are fairly conducted in the same reduced network, and in each reduced graph, data are aggregated from the same set of nodes to the same sink node. Figure 4 illustrates the average delays of different methods. From Figure 4(a) to Figure 4(b), we can see that compressive scheduling has remarkable effect on the delays of data aggregation in sparse networks. It significantly reduces the delays.

Theoretically, the delay of compressive scheduling algorithm for data aggregation should be at least that of our improved and distributed algorithms. In practice, we show that our algorithm can be further improved, although the theoretical performances remain the same here. The proposed *compressive scheduling* method can further reduce the delay by merging the links scheduled in different slots to a single time-slot without violating the SINR constraints.

In previous subsections, we all set δ to be some default values, which may not be an optimum choice to minimize the delays. Here we conduct simulations to see the effect of δ in different scenarios. Let the network size (# of nodes) be 1000 and the deployment area be $(500m \times 500m)$. Fig 5(a) compares the delays of Distributed Algorithm and Improved Algorithm with varied values of δ. Fig 5(b) shows the best choices of δ with varied network size in a fixed deployment area.

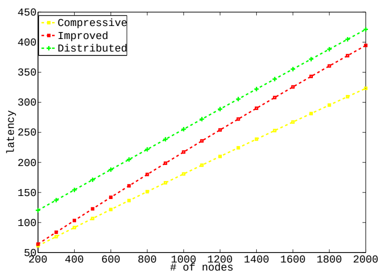


(a) Fixed R

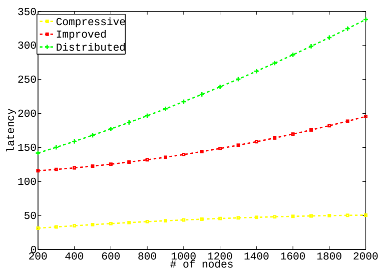


(b) Fixed Δ

Fig. 3: Worst case.

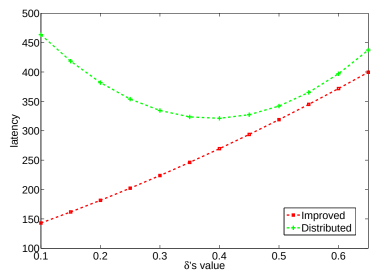


(a) Fixed R

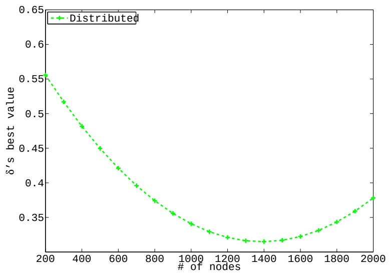


(b) Fixed Δ

Fig. 4: Average case.



(a) Latency with δ



(b) The best choice of δ

Fig. 5: The effect of δ.

Here a best choice means that the delay of Distributed Algorithm reaches minimum when δ is set as this value. We conjecture that the smallest δ which can ensure the connectivity of $G(V, \delta r)$ is an optimum choice.

8 Conclusion

We proposed two delay efficient aggregation scheduling algorithms under the physical interference model in wireless sensor networks. Some interesting questions are left for future research. The first one is to improve the approximation ratio of the proposed algorithms. The second one is to design efficient data aggregation method that has the asymptotically optimum performance guarantee compared with the optimum delay using $G(V, r)$. The third one is to extend the proposed algorithms to deal with a more general path loss model.

9 Acknowledgement

The research of Xiaohua Xu is supported in part by NSF grant CNS-1248092.

The research of Xiang-Yang Li is partially supported by NSF CNS-0832120, NSF CNS-1035894, NSF ECCS-1247944, National Natural Science Foundation of China under Grant No. 61170216, No. 61228202, China 973 Program under Grant No.2011CB302705.

The research of Min Song is supported in part by NSF IPA Independent Research and Development (IR/D) Program. Any opinion, finding, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

[1] Blum, J., Ding,M., and Cheng, X. Applications of Connected Dominating Sets in Wireless Networks. *Handbook of Combinatorial Optimization*. (Editors D.-Z. Du and P. Pardalos), pp. 329-369, 2004, Kluwer Academic Publisher

[2] Brar, G., Blough, D., and Santi, P. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *ACM MobiCom* (2006).

[3] Chafekar, D., Kumar, V., Marathe, M., Parthasarathy, S., and Srinivasan, A. Cross-layer latency minimization in wireless networks with SINR constraints. In *ACM MobiHoc* (2007).

[4] Chafekar, D., Kumar, V., Marathe, M., Parthasarathy, S., and Srinivasan, A. Approximation Algorithms for Computing Capacity of Wireless Networks with SINR Constraints. In *IEEE INFOCOM* (2008).

[5] Chen, X., Hu, X., and Zhu, J. Minimum data aggregation time problem in wireless sensor networks. *Mobile Ad-hoc and Sensor Networks* (2005).

[6] Cruz, R., and Santhanam, A. Optimal routing, link scheduling and power control in multihop wireless networks. In *IEEE INFOCOM* (2003).

[7] ElBatt, T., and Ephremides, A. Joint scheduling and power control for wireless ad hoc networks. *IEEE Transactions on Wireless Communications* (2004).

[8] Goussevskaia, O., Oswald, Y., and Wattenhofer, R. Complexity in geometric SINR. In *ACM MobiHoc* (2007).

[9] Goussevskaia, O., W. R. H. M., and Welzl, E. Capacity of Arbitrary Wireless Networks. In *IEEE INFOCOM* (2009).

[10] Joo, C. and Lin, X. and Shroff, N.B. Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-hop Wireless Networks. In *IEEE INFOCOM* (2008).

[11] Le, L.B. and Modiano, E. and Joo, C. and Shroff, N.B. Longest-queue-first scheduling under SINR interference model. In *ACM MobiHoc* (2010).

[12] Li, Y., G., L., and Prasad., S. An Energy-Efficient Distributed Algorithm for Minimum-Latency Aggregation Scheduling in Wireless Sensor Networks. In *IEEE ICDCS* (2010).

[13] Li, X. Multicast Capacity of Wireless Ad Hoc Networks. *IEEE/ACM Transaction on Networking* (2008).

[14] Li, X., Liu, Y., Li, S.,Tang, S. Multicast capacity of wireless ad hoc networks under Gaussian channel model. *IEEE/ACM Transaction on Networking* (2010).

[15] Li, X., and Wang, Y. and Wang, Y. Complexity of data collection, aggregation, and selection for wireless sensor networks. *IEEE Transaction on Computers* (2011).

[16] Li, X.Y., and Xu, X., and Wang, S., and Tang, S., and Dai, G., and Zhao, J., and Qi, Y. Efficient data aggregation in multi-hop wireless sensor networks under physical interference model. *IEEE MASS* (2009).

[17] Moscibroda, T., and Wattenhofer, R. The Complexity of Connectivity in Wireless Networks. In *IEEE INFOCOM* (2006).

[18] Moscibroda, T., Wattenhofer, R., and Zollinger, A. Topology control meets SINR: the scheduling complexity of arbitrary topologies. In *ACM MobiHoc* (2006).

[19] Shang, W., Wan, P., and Hu, X. Approximation algorithm for minimal convergecast time problem in wireless sensor networks. *Wireless Networks* (2009).

[20] Wan, P., Alzoubi, K., and Frieder, O. Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications* (2004).

[21] Wan, P., Jia, X., and Yao, F. Maximum Independent Set of Links under Physical Interference Model. *Wireless Algorithms, Systems, and Applications*, (2009).

[22] Wan, P., Scott, C., Wang, L., Wan, Z., and Jia, X. Minimum-latency aggregation scheduling in multihop wireless networks. *ACM MobiHoc* (2009).

[23] Wan, P., Wang, L., and Frieder, O. Fast Group Communications in Multihop Wireless Networks Subject to Physical Interference. *IEEE MASS* (2009).

[24] Wang, Y., Wang, W., Li, X., Song, W. Interference-aware joint routing and TDMA link scheduling for static wireless networks In *IEEE Transactions on Parallel and Distributed Systems* (2006), 19 (12), 1709-1726

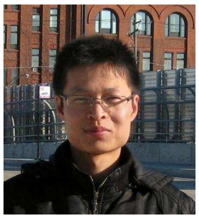
[25] Xu, X., and Tang, S. A Constant Approximation Algorithm for Link Scheduling in Arbitrary Networks With Physical Interference Model In *ACM MobiHoc FOWANC workshop* (2009).

[26] Xu, X., Tang, S., and Wan, P. Maximum weighted independent set of links under physical interference model. *Wireless Algorithms, Systems, and Applications* (2010).

[27] Xu, X., Li, X., Mao, X., Tang, S., and Wang, S. A Delay Efficient Algorithm for Data Aggregation in Multi-hop Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems* (2011).

[28] Xu, X., Li, X., Wan, P., Tang, S. Efficient scheduling for periodic aggregation queries in multihop sensor networks. *IEEE/ACM Transactions on Networking* (2012).

[29] Yu, Bo, L., J., and Li, Y. Distributed Data Aggregation Scheduling in Wireless Sensor Networks. In *IEEE INFOCOM* (2009).



Dr. Xiaohua Xu is currently a Research Assistant Professor in the Electrical Engineering and Computer Science Department at the University of Toledo. He received the BS degree from ChuKochen Honors College at Zhejiang University, P.R. China, in 2007, the Ph.D. degree in Computer Science from Illinois Institute of Technology in 2012. His research interests and experience span a wide range of topics from theoretical analysis to practical design in wireless networks.

Dr. Xiang-Yang Li is currently a Professor of Computer Science at the Illinois Institute of Technology. He hold MS (2000) and PhD (2001) degree at Computer Science from University of Illinois at Urbana-Champaign. He received B.Eng. at Computer Science and Bachelor degree at Business Management from Tsinghua University, P.R. China in 1995. His research interests span wireless networks, cyber physical systems, security and privacy, social networks, and algorithms, and has published over 200 papers and edited 4 books on these fields. He is an editor of IEEE TPDS, IEEE TMC, Networks: An International Journal, and was a guest editor of several journals, such as ACM MONET, IEEE JSAC. In 2008, he published a monograph ”Wireless Ad Hoc and Sensor Networks: Theory and Applications”. He is a senior member of the IEEE and a member of ACM.

Dr. Min Song is a Professor in the Electrical Engineering and Computer Science Department at the University of Toledo. He received his Ph.D. in Computer Science from the University of Toledo in 2001. Dr. Song is the recipient of NSF CAREER Award. Dr. Song’s professional career is comprised of a total 22 years of work experience in academia, government, and industry. He was the Founding Director of a Networking System Division in an IT company, and launched an international journal and served as the Editor-in-Chief. Dr. Song has acted as an Editor or Guest Editor of 13 international journals, and served as a General chair, Technical Program Committee chair, and Session chair for numerous international conferences. Dr. Song is an IEEE Senior member.