DATA COLLECTION SCHEDULING IN DIRECTIONAL WIRELESS SENSOR NETWORKS

A Thesis

Presented to

The Faculty of the Department of Computer Science

Sam Houston State University

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science

---

by

Ecem Simsek

May, 2019

DATA COLLECTION SCHEDULING IN DIRECTIONAL WIRELESS SENSOR NETWORKS

by

Ecem Simsek

____

APPROVED:

Min Kyung An, PhD
Thesis Director

Hyuk Cho, PhD
Committee Member

Qingzhong Liu, PhD
Committee Member

John Pascarella, PhD
Dean, College of Science and Engineering Technology

DATA COLLECTION SCHEDULING IN DIRECTIONAL WIRELESS SENSOR NETWORKS

by

Ecem Simsek

APPROVED:

Min Kyung An, PhD
Thesis Director

Hyuk Cho, PhD
Committee Member

Qingzhong Liu, PhD
Committee Member

John Pascarella, PhD
Dean, College of Science and Engineering Technology

ABSTRACT

Simsek, Ecem , *Data Collection Scheduling in Directional Wireless Sensor Networks*. Master of Science (Computing and Information Science), May, 2019, Sam Houston State University, Huntsville, Texas.

This thesis studies *Minimum Latency Collection Scheduling (MLCS)* problem in Wireless Sensor Networks (WSNs) whose objective is to obtain collision-free data collection schedules with minimum latencies. Unlike most existing works that explored the problem with the uniform power model in omnidirectional WSNs, this thesis studies the problem with *non-uniform power model* in *directional WSNs*. In this study, *power control*, where power levels of sensor nodes need to be controlled, is also considered. The thesis proposes an algorithm, named *Hierarchical Streaming Collection Scheduling Algorithm (HSCS)*, that produces collision-free data collection schedules where appropriate power levels are assigned, and validates its performance in terms of latency on simulate networks.

KEY WORDS: Data collection, Collision-Free, Latency, Directional Wireless Sensor Network, Non-uniform Power, Power control

ACKNOWLEDGEMENTS

Most of all, I would like to express my sincere gratitude to my supervising professor, Dr. Min Kyung An, for her continuous guidance and persistent help during my master's studies. It was a great opportunity for me to work on this thesis with her.

I would like to thank my committee members, Dr. Hyuk Cho, and Dr. Qingzhong Liu, for their time and valuable suggestion.

Last but not the least, special thanks should be given to my parents for their support, full of love, and powerful encouragement throughout my academic career.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1. Wireless Sensor Network

Wireless Sensor Networks (WSNs) consist of wireless sensor devices whose powers are supplied from their embedded small batteries, which make their energy sources very limited [1]. The small-sized nodes are set to turn on their powers to emit radio signals or to shut them down to conserve energies [2].

The applications of WSNs perform various tasks such as *broadcasting*, *gossiping*, *data collection*, and *aggregation* [3]. Broadcasting is to disperse a data from a base station (or a sink) to all the other nodes in network periodically, whereas gossiping is to distribute data from each node to all the other nodes. Data collection and aggregation perform similarly but the former is to collect raw data from every node to the sink node, while the latter aggregates data as a single data to the sink node.

While performing the tasks, a node emits its signal including data to nodes that reside in its *transmission range*. If the signal is interrupted by other simultaneously emitted signals, a *collision* occurs, and the data should be retransmitted [1]. Due to nodes' limited energies, it is crucial to reduce such unnecessary retransmissions to prolong the lifetime of a network [3].

One of common approaches to complete the tasks avoiding any collisions is to assign *timeslots* to obtain a *minimum latency schedule*. If nodes follow the schedule, then any nodes assigned the same timeslot can send their data simultaneously without causing any collisions, and the tasks can complete using the minimum number of timeslots.

1.2. Minimum Latency Data Collection Scheduling (MLCS) Problem

This thesis studies the *Minimum Latency Collection Scheduling (MLCS)* problem in WSNs whose objective is to obtain collision-free collection schedules with minimum latency. Unlike most existing works that explored the problem with the uniform power model in omnidirectional WSNs, this thesis studies the problem with *non-uniform power model* in *directional WSNs*. In this study, *power control*, where power levels of sensor nodes need to be controlled, is also considered. The thesis proposes an algorithm, named *Hierarchical Streaming Collection Scheduling Algorithm (HSCS)*, that produces collision-free collection schedules where appropriate power levels are assigned, and validates its performance in terms of latency on simulate networks.

1.3. Outline of the Thesis

This thesis is organized as follows. In Section II, various antenna, power and interference models used in WSNs is first introduced. Then, the formal definition of the MLCS problem and the network model used to study the problem are described. Section III summarizes the existing algorithms for the MLCS problem in different antenna and power models, and it also introduces *Hierarchical Agglomerative Aggregation Scheduling (HAAS)* algorithm proposed by An et al. [5] based on which a new data collection scheduling algorithm is proposed. In Section IV, the new data collection scheduling algorithm is described, and its performance is validated through simulations in Section V. Finally, the thesis is concluded with some remarks in Section VI.

CHAPTER II

PRELIMINARIES

When studying the Minimum Latency Collection Scheduling (MLCS) problem, choosing antenna, power and interference models is a crucial step. While a substantial amount of research results has been obtained in omni-directional Wireless Sensor Networks (WSNs) with no power control, this thesis studies the problem in directional WSNs with power control. In this chapter, various antenna, power and interference models are introduced. Then, the formal definition of the MLCS problem and the network model used to study the problem are described.

2.1 Antenna Model

In traditional *omni-directional WSNs*, every node is equipped with an omni-directional sending and receiving antenna with a beam-width $\theta = 360°$. The omni-directional WSNs are modeled as unidirectional graphs, where nodes are connected with each other via an undirected communication edge [2] if they cover each other in their transmission ranges.

Unlike the omni-directional WSNs, in *directional WSNs*, nodes collaboratively determine and orientate sending antennas' directions whose beam-width is $\theta \in (0{,}360°]$. Nodes still are equipped with omni-directional receiving antennas. The directional WSNs are commonly modeled as a directed graph, where nodes are connected via directed communication edge [2] if they cover each other in their transmission ranges.

This thesis adopts the *switch beam directional antenna system* [3] where each u has K fixed *broadcasting sectors*, denoted by $sec_k(u)$, $1 \leq k \leq K$, whose central angle is $\theta \in (0{,}2\pi)$ as in [5]. Each node can switch on one of its sectors for transmission. Let us denote the set of K sectors of u by $S(u) = \{sec_k(u) | 1 \leq k \leq K\}$ [5]. Commercially, available sectored antennas are typically designed for beam-widths of $\pi, 2\pi/3, \pi/2, \pi/3,$ and $\pi/4$ [6].

The motivation for adopting directional WSNs is with an intuition that reducing broadcasting areas would reduce potential collisions. For instance, in Figure 1, if the sender v and w are equipped with omni-directional antennas, they cannot send data simultaneously to their receivers (i.e., the two receivers u and x cannot receive data at the same time due to collisions). However, if they are equipped with directional antenna with narrower broadcasting areas, then the two receivers u and x can receive data at the same time without any collisions.

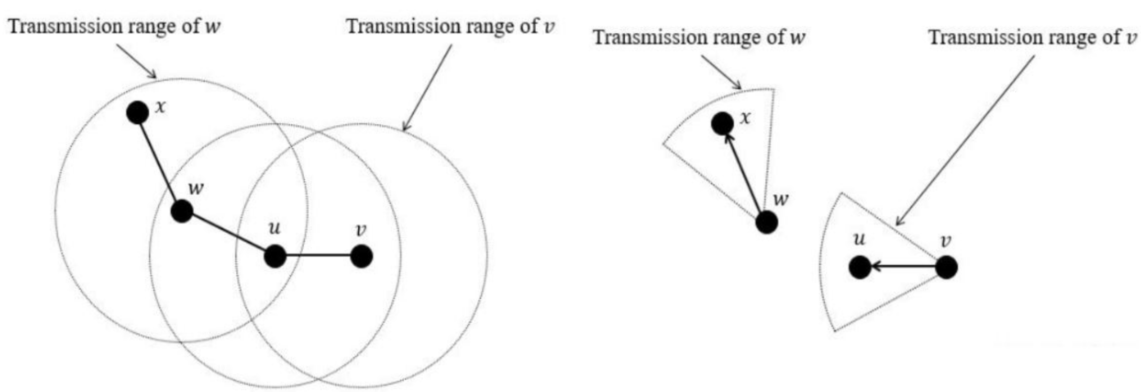

Figure 1. Omni-directional antenna vs. directional antenna [2].

2.2 Power Models

Problems in WSNs have been actively investigated with two different power models: *non-uniform power model*, where each node can be assigned different power level and *uniform power model*, where every node is initially assigned a uniform power level.

Power control in non-uniform power model is to assign an appropriate power level to each node. It is crucial to control powers in WSNs to conserve energies because for instance, if a receiver node is closely located to its sender node, then the sender does not have to use strong power level to send its data to the receiver.

This thesis adopts non-uniform power level with power control.

2.3 Interference Models

Different kinds of interference models have been proposed for the problems in WSNs. The *collision-free model* and *collision-interference-free model*, both together is called *graph models*, are two of the interference models widely used in WSNs. While the collision-free model concerns *collision* only, the collision-interference-free concerns both *collision* and *interference* [1]. Given a *transmission range* (or *broadcasting range*) $r(u)$ for every node u, the *interference range* of u is defined as $\rho \cdot r(u)$, where $\rho \geq 1$ is the *interference factor* [1]. If $\rho = 1$, it is a collision-free model, otherwise if $\rho \geq 1$, it is a collision-interference-free graph model.

2.4 Network Model

In this thesis, a WSN consists of a set V of nodes, each $u \in V$ which is equipped with a switch beam directional sending antenna with a beam-width $\theta \in (0{,}360°]$ and an omni-directional receiving antenna. At a timeslot $t(u)$, a transmission power level $p(u) \in (0, P_{max}]$ and an antenna orientation $w(u)$ are assigned to u, which activates one of antenna sector using $p(u)$. Accordingly, the transmission range $r(u)$ of u is defined as the radius of the broadcasting sector $sec_k(u)$ and this sector is covered by $p(u)$.

A collision is said to occur if there is a node w such that $w \in sec_k(u)$ and $w \in sec_k(u')$, and there are a concurrently sending nodes u and u', where $t(u) = t(u')$ [5].

2.5 Minimum Latency Collection Scheduling (MLCS) Problem

The MLCS problem is defined as follows. Given a set V of nodes, we assign each node timeslots, power levels, antenna orientations so that any nodes assigned the same timeslot can send data to their receivers simultaneously without any collision and data of all nodes are collected to a sink node $s \in V$. Formally, at a timeslot t, we have an *assignment set* $\pi_t = \left\{\left(s_{t_i}, \omega\left(s_{t_i}\right), p\left(s_{t_i}\right)\right) | 1 \leq i \leq l\right\}$, where l denotes the number of nodes scheduled at timeslot t [5]. In each assignment π_t, every sender s_{t_i} can send data simultaneously to their receivers r_{t_i} with power level $p\left(s_{t_i}\right)$ by orienting their antennas to the direction $\omega\left(s_{t_i}\right)$ at the assigned timeslot t. The assignments set produces a *schedule* $\Pi = (\pi_1, \pi_2, \dots, \pi_L)$, where L is the length of the schedule, also called *latency*. The schedule Π is said *successful* if data of every node $v \in V$ is collected to a sink node $s \in V$ [5].

Input: A set V of nodes

Output: A successful schedule Π

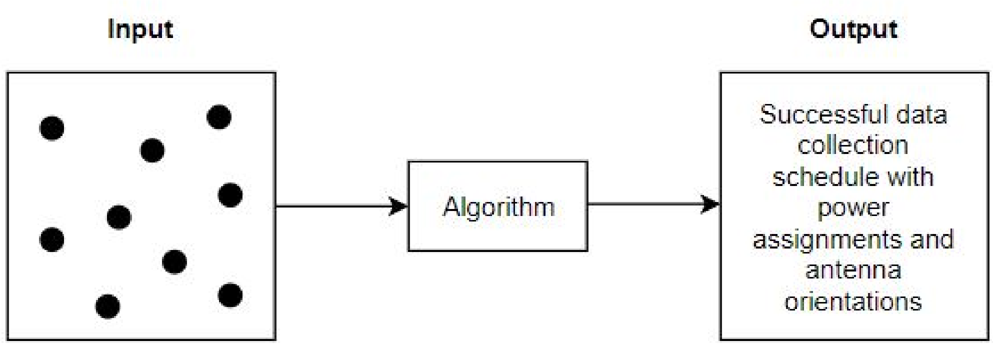

Figure 2. Input and Output diagram.

CHAPTER III

RELATED WORKS

3.1 Minimum Latency Data Collection Problem

The Minimum Latency Collection Scheduling (MLCS) problem has been widely studied in the past years. As the problem was proved NP-hard by Bermond et al. [7], this thesis more focuses on summarizing notable works that propose approximation algorithms rather than heuristics.

Let us first consider the collision-free model, i.e., $\rho = 1$. Both Florens et al. [8] and Bonifaci et al. [9] proposed 3-approximation algorithms, but the former studied the problem for tree networks only. Later, while Bermond et al. [7] proved its NP-hardness, Coleri et al. [10] proved NP-completeness and proposed two heuristic algorithms. Bermond et al. [11] also studied the problem in special grid shaped networks and proposed a 3-approximation algorithm. Bermond et al. [12] addressed the problem on linear topologies and proposed an optimal algorithm with $r(u) = 2, 3,$ and 5. Kowalski et al. [13] also proposed a 2-approximation algorithm in linear topologies.

In the collision-interference-free model, i.e., $\rho \geq 1$, Bermond et al. [7] and Bonifaci et al. [9] proposed 4-approximation algorithms with $\rho \geq 1$ and $\rho > 1$, respectively. Bermond et al. [11] studied the problem in grid shaped networks and proposed a 4-approximation algorithm with $\rho = 2$. The problem was then investigated with $\rho \geq 2$ in tree networks by Bermond et al. [14] and they proposed a closed formula for the data collection of the optimal schedule. An et al. [1, 15] proposed 3-approximation algorithm with $\rho \geq 1$.

Table 1 summarizes the existing algorithms for the MLCS problem in different antenna and power models. Notice that all existing works studied the problem adopting omni-directional WSNs with uniform power model.

Table 1

Existing Works for MLCS Problem in Various Antenna and Power Models

	Uniform Power	**Non-uniform Power**
Omnidirectional WSNs	[1], [15]-[28]	
Directional WSNs		This Paper

3.2 Hierarchical Agglomerative Aggregation Scheduling Problem

The thesis proposes an algorithm based on an existing work by An et al. [5] that studied other application of WSNs, *aggregation*. An et al. [5] investigated the *Minimum Latency Aggregation Scheduling (MLAS)* problem that targets to attain collision-free minimum latency data aggregation schedules adopting non-uniform power model with power control in directional WSNs. Unlike existing works that schedule nodes based on trees, their proposed scheduling algorithm, named *Hierarchical Agglomerative Aggregation Scheduling (HAAS)*, does not create trees. Instead, it repeatedly partitions a whole network into smaller networks, and the smaller networks are systematically agglomerated to achieve aggregate data with no collisions.

CHAPTER IV

HIERARCHICAL STREAMING COLLECTION SCHEDULING ALGORITHM

This section describes the proposed algorithm, named *Hierarchical Streaming Collection Scheduling algorithm (HSCS)*, which is designed based on An et al. [5]'s *Hierarchical Agglomerative Aggregation Scheduling (HAAS)* algorithm. As in [5], it is assumed that every node $u \in V$ is equipped with a switch beam directional antenna with a fixed beam-width $\theta = \pi/2$, and its broadcasting disk is partitioned to have $K = 4$ sectors. Each section is identified as $sec_k(u), k \in \{1, 2, 3, 4\}$ as shown in Figure 3.

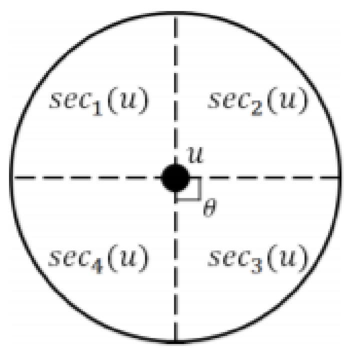

Figure 3. Broadcasting sectors of u with switch beam-width $\theta = \pi/2$ [5].

The pseudocode of the proposed HSCS algorithm is shown in **Algorithm 1**. It starts by setting the first timeslot $t = 1$ (Step 1), and setting each node $v \in V$ except the sink s (Steps 2 – 5) to have $n_m(v)$ with 0, which is the number of messages that v has, and have $n_f(v)$ with 0, which is the number of messages to forward. The difference between $n_m(v)$ and $n_f(v)$ is that $n_m(v)$ is with v's own data, and $n_f(v)$ is with messages that are delivered from other nodes.

Table 2

Algorithm 1: HSCS

Algorithm 1 HSCS (Modified HAAS [5])

Input: A set V of nodes

Output: Schedule II

1: $t \leftarrow 1$

2: **for** each $v \in V \setminus \{s\}$ **do**

3: $n_m(v) \leftarrow 1$

4: $n_f(v) \leftarrow 0$

5: **end for**

6: **repeat**

7: $V' \leftarrow \{v | v \in V$ and $n_m(v) + n_f(v) > 0\} \cup \{s\}$

8: **repeat**

9: Mark all nodes in V' as non-head nodes.

10: Partition the network in $\leq z^2$ square cells such that $0 \leq |C_{i,j}| \leq 4$, where $C_{i,j}$ denotes the set of nodes which reside in the cell at row i and column j $(1 \leq i, j \leq z)$.

11: $GRAY \leftarrow \{C_{i,j} | i\%2 \neq 0$ and $j\%2 \neq 0\} \cup \{C_{i,j} | i\%2 = 0$ and $j\%2 = 0\}$

12: $WHITE \leftarrow \{C_{i,j} | i\%2 \neq 0$ and $j\%2 = 0\} \cup \{C_{i,j} | i\%2 = 0$ and $j\%2 \neq 0\}$

13: $t \leftarrow Scheduling(t, GRAY)$

14: $t \leftarrow Scheduling(t, WHITE)$

15: $V' \leftarrow \{v \mid v \in V, v \equiv head$ or $v \equiv s\}$

16: **until** $z = 1$

17: **until** V' has s only

18: return $\pi \leftarrow (\pi_1, \pi_2, \dots, \pi_L$

After that, the following steps (Steps 6 – 17) are repeated until the sink s collects data from every other node. Steps 8 – 16 is the modified procedure based on An et al. [5]'s HCAS algorithm. In these steps, networks are partitioned into square cells, and the cells are divided into two groups, $GRAY$ and $WHITE$. (See Figure 4.) It then calls

$Scheduling$ (**Algorithm 2**) to schedule the nodes in the cells of $GRAY$ with the starting timeslot t_{start}. Specifically, for each cell $C_{i,j} \in GRAY$, a *head node* is decided to collect all data from the other nodes. At the end of the $Scheduling$ call, every head node collected data from all the other nodes in $C_{i,j}$, and $Scheduling$ returns the next timeslot t_{next} that will be the starting timeslot for group $WHITE$. Again, $Scheduling$ is called to schedule the nodes in the cells of $WHITE$ with the starting timeslot t_{start}. The same procedure mentioned above is repeated to collect data to head node in each WHITE cell. Then, HSCS (Step 15) updates the sender set V' by removing the scheduled nodes. Note that the updated $V' = \{u | u \in C_{i,j}, 1 \leq i,j \leq z, u \equiv head$ or $u \equiv s\}$. Steps 8 – 16 are repeated until every node is scheduled.

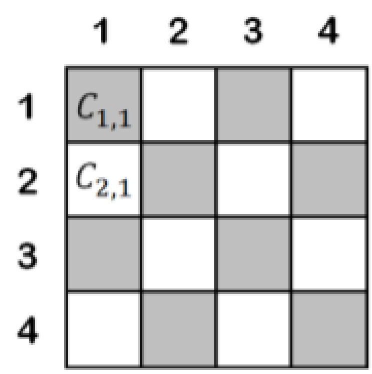

Figure 4. A network partitions. Cells are grouping into $GRAY$ and $WHITE$ [5].

The completion of Steps 8 – 16 does not imply that all data is collected. HSCS repeats these steps until every data is collected to the sink node, i.e., V’ has only s, where $V' \leftarrow \{v | v \in V$ and $n_m(v) + n_f(v) > 0\} \cup \{s\}$ (Steps 6 – 17).

Table 3

Algorithm 2: Scheduling

Algorithm 2: Scheduling

Input: Starting timeslot t_{start}, and a group $GROUP$

Output: Next timeslot t_{next} for the other group

for each cell $C_{i,j} \in GROUP$ **do**
    $t \leftarrow 1$
    Partition $C_{i,j}$ into two $subcells$ C_A and C_B, in each of which at most four nodes reside.
    Label nodes in C_A as a and a', and nodes in C_B as b and b'.
    $t(a) \leftarrow t, p(a) \leftarrow d(a, a'),$
    $w(a) \in \{sec_k(a) | sec_k(a) \in S(a)\},$
    $a \in sec_k(a)$, and $sec_k(a)$ does not cover any other $GROUP$ cells.
    if $n_m(a) = 1$ **then**
        $n_m(\mathrm{a}) \leftarrow 0$
    else if $n_m(a) = 0$ and $n_f(a) > 0$ **then**
        $n_f(a) \leftarrow n_f(a) - 1$
    end if
    $n_f(a') \leftarrow n_f(a') + 1$
    $\pi_t \leftarrow \pi_t \cup \big(a, w(a), p(a)\big), t \leftarrow t + 1$
    $t(b) \leftarrow t, p(b) \leftarrow d(b, b'),$
    $w(b) \in \{sec_k(b) | sec_k(b) \in S(b)\},$
    $b' \in sec_k(b)$, and $sec_k(b)$ does not cover any other $GROUP$ cells.
    if $n_m(b) = 1$ **then**
        $n_m(b) \leftarrow 0$
    else if $n_m(b) = 0$ and $n_f(b) > 0$ **then**
        $n_f(b) \leftarrow n_f(a) - 1$
    end if
    $n_f(b') \leftarrow n_f(b') + 1$
    $\pi_t \leftarrow \pi_t \cup \big(b, w(b), p(b)\big), t \leftarrow t + 1$

Pick node $h \in a', b'$ which is closed to sink s is the shortest as the head node for the cell.

$h' \leftarrow u | u \in a', b', u \not\equiv h$

$t(h') \leftarrow t, p(h') \leftarrow d(h', h)$,

$w(h') \in \{sec_k(h') | sec_k(h') \in S(h')\}$,

$h \in sec_k(h')$ and $sec_k(h')$ does not cover any other $GROUP$ cells

if $n_m(h') = 1$ **then**

  $n_m(h') \leftarrow 0$

else if $n_m(h') = 0$ and $n_f(h') > 0$ **then**

  $n_f(h') \leftarrow n_f(h') - 1$

end if

$n_f(h) \leftarrow n_f(h) + 1$

$\pi_t \leftarrow \pi_t \cup \left(h', w(h'), d(h', h)\right)$

end for

$t_{next} \leftarrow \{\max_t(u) | u \in C_{i,j}, C_{i,j} \in GROUP + 1\}$

return t_{next}

CHAPTER V

SIMULATION

This chapter evaluates the performance of the purposed algorithm, *Hierarchical Streaming Collection Scheduling (HSCS)*, in terms of latency by comparing it with the algorithm by An et al. [1] which studied the MLAS problem. An et al. [1]'s algorithm runs adopting omni-directional WSNs whose nodes are assigned a uniform power level initially, i.e., no power control.

In the simulation, for each number of nodes $n = 100, 200, 300, 400, 500$, 100 random networks are generated in the Euclidean plane of dimension 500×500. Then both HSCS and An et al. [1]'s algorithms are tested on the networks, and the latencies produced are averaged.

As shown in Table 4 and Figure 5, HSCS performs better than An et al. [1]'s algorithm in terms of average latencies.

Table 4

Simulation Results

	Averaged Latencies		# of worse resulted networks	% decrease on latencies
n	HSCS	An et al. [1]		
100	220.37	269.39	6	18.20%
200	441.61	566.38	1	21.50%
300	660.17	685.48	1	23.73%
400	877.4	1164.98	1	24.68%
500	1086.46	1463.67	0	25.78%

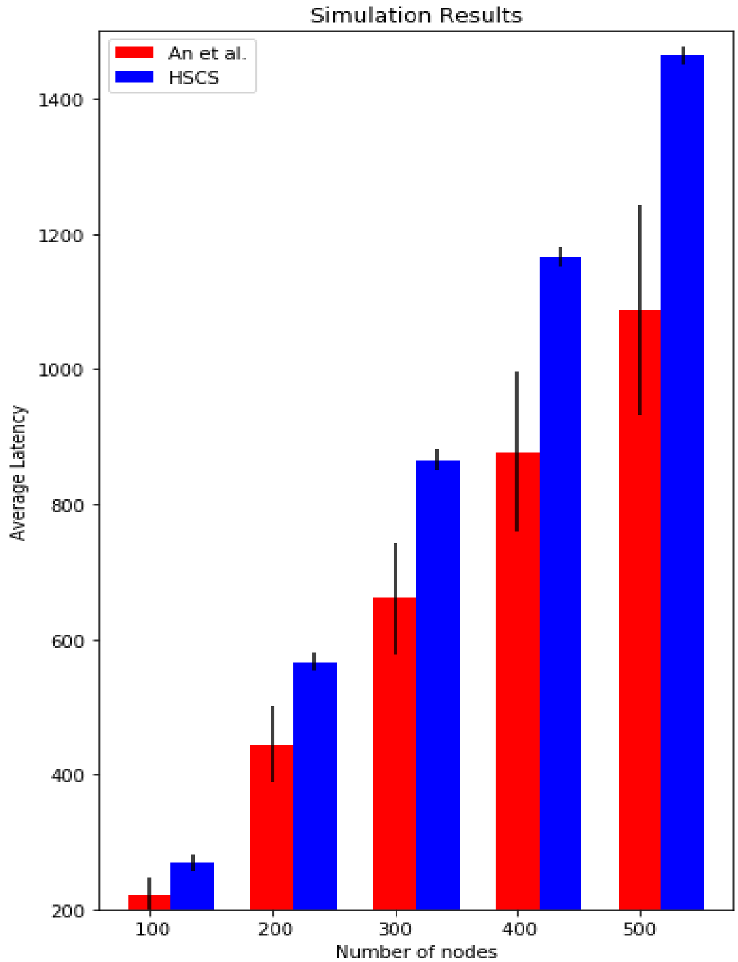

Figure 5. Simulation Result in Grouped Error Bar.

CHAPTER VI

CONCLUSION

In this thesis, the Minimum Latency Collection Scheduling (*MLCS*) problem with non-uniform power model in directional wireless sensor networks (WSNs) unlike the other existing algorithms adopting omni-directional WSNs with uniform power model. The proposed algorithm, named *Hierarchical Streaming Collection Scheduling (HSCS),* does not run on backbone trees where all existing works construct trees first and assign timeslots based on the trees. Instead of using trees, the HSCS algorithm was designed based on an existing aggregation scheduling algorithm that employs hierarchical agglomerative steps, where a whole network is repeatedly partitioned into smaller networks and the smaller networks are systematically agglomerated to assign timeslots to complete a given task. The simulation result shows that HSCS performs better than a recent existing algorithm by An et al [1].

As to the future study, analyzing the complexity of the proposed algorithm and the power consumptions, and studying other related problems such as broadcasting with similar approach are planned.

REFERENCES

[1] An, M. K., & Cho, H. (2016). Efficient Data Collection in Interference-Aware Wireless Sensor Networks. Journal of Networks, 10(12). doi:10.4304/jnw.10.12.658-667

[2] An, M., Cho, H. (2016). 'Efficient Broadcasting in Wireless Sensor Networks'. World Academy of Science, Engineering and Technology, International Science Index 114, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 10(6), 1133 - 1139.

[3] An, M. K. & Cho, H. (2018). "Scheduling Problems in Wireless Sensor Networks and Internet of Things: A Comparative Overview" IEEE International Conference on Computing, Networking and Communication Conference (IEMCON), University of British Columbia, Canada

[4] H. Liu, Z. Liu, D. Li, X. Lu, and H. Du, "Approximation Algorithms for Minimum Latency Data Aggregation in Wireless Sensor Networks with Directional Antenna," Theor. Comput. Sci., vol. 497, pp. 139–153, 2013.

[5] An, M. K., Cho, H., & Chen, L. (2018). Hierarchical Agglomerative Aggregation Scheduling in Directional Wireless Sensor Networks. 2018 International Conference on Computing, Networking and Communications (ICNC).

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

[7] J.-C. Bermond, J. Galtier, R. Klasing, N. Morales, and S. Perennes, "Hardness and Approximation of Gathering in Static Radio Networks," Parallel Processing Letters, vol. 16, no. 2, pp. 165 – 184, 2006.

[8] C. Florens and R. J. McEliece, “Packet Distribution Algorithms for Sensor Networks,” in INFOCOM, 2003

[9] V. Bonifaci, P. Korteweg, A. Marchetti-Spaccamela, and L. Stougie, “An Approximation Algorithm for the Wireless Gathering Problem,” in SWAT, 2006, pp. 328 – 338.

[10] S. C. Ergen and P. Varaiya, “TDMA Scheduling Algorithms for Wireless Sensor Networks,” Wireless Networks, vol. 16, no. 4, pp. 985 – 997, 2010.

[11] J.-C. Bermond, B. Li, N. Nisse, H. Rivano, and M.-L. Yu, “Data Gathering and Personalized Broadcasting in Radio Grids with Interferences,” INRIA, Research Report RR8218, 2013.

[12] J.-C. Bermond, R. Klasing, N. Morales, S. Perennes, and ´P. Reyes, “Gathering Radio Messages in the Path,” Discrete Mathematics, Algorithms and Applications, vol. 5, no. 1, 2013

[13] D. R. Kowalski, E. Nussbaum, M. Segal, and V. Milyeykovski, “Scheduling Problems in Transportation Networks of Line Topology,” Optimization Letters, vol. 8, no. 2, pp. 777 – 799, 2014.

[14] J.-C. Bermond, L. Gargano, S. Prennes, A. A. Rescigno, and U. Vaccaro, “Optimal Time Data Gathering in Wireless Networks with Multidirectional Antennas,” Theoretical Computer Science, vol. 509, pp. 122 – 139, 2013.

[15] M. K. An and H. Cho, “Minimum Latency Data Collection in Interference-aware Wireless Sensor Networks,” in CITS, 2014, pp. 1–5.

[16] M. K. An, N. X. Lam, D. T. Huynh, and T. N. Nguyen, “Minimum Latency Gossiping in Wireless Sensor Networks,” in Proceedings of the International Conference on Wireless Networks (ICWN), 2012, pp. 406–412.

[17] K. Krzywdzinski, “Fast Construction of Broadcast Scheduling and Gossiping in Dynamic Ad Hoc Networks,” in IMCSIT, 2010, pp. 879–884

[18] R. Gandhi, Y.-A. Kim, S. Lee, J. Ryu, and P.-J. Wan, “Approximation Algorithms for Data Broadcast in Wireless Networks,” in INFOCOM, 2009, pp. 2681–2685.

[19] R. Gandhi, Y. Kim, S. Lee, J. Ryu, and P. Wan, “Approximation Algorithms for Data Broadcast in Wireless Networks,” IEEE Trans. Mob.Comput., vol. 11, no. 7, pp. 1237–1248, 2012.

[20] C. Florens and R. J. McEliece, “Packet Distribution Algorithms for Sensor Networks,” in INFOCOM, 2003, pp. 1063–1072.

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

[22] J.-C. Bermond, N. Morales, S. Prennes, J. Galtier, and R. Klasing, “Hardness and Approximation of Gathering in Static Radio Networks,” in PERCOMW, 2006, pp. 75–79.

[23] V. Bonifaci, P. Korteweg, A. Marchetti-Spaccamela, and L. Stougie, “An Approximation Algorithm for the Wireless Gathering Problem,” in SWAT, 2006, pp. 328–338.

[24] J.-C. Bermond, B. Li, N. Nisse, H. Rivano, and M.-L. Yu, “Data Gathering and Personalized Broadcasting in Radio Grids with Interferences,” INRIA, Research Report RR-8218, 2013.

[25] J.-C. Bermond, L. Gargano, S. Prennes, A. A. Rescigno, and U. Vaccaro, “Optimal Time Data Gathering in Wireless Networks with Multidirectional Antennas,” TCS, vol. 509, pp. 122–139, 2013.

[26] J.-C. Bermond, R. Klasing, N. Morales, S. P´erennes, and P. Reyes, “Gathering Radio Messages in the Path,” DMAA, vol. 5, no. 1, 2013.

[27] D. R. Kowalski, E. Nussbaum, M. Segal, and V. Milyeykovski, “Scheduling Problems in Transportation Networks of Line Topology,” Optimization Letters, vol. 8, no. 2, pp. 777–799, 2014.

[28] M. K. An, “Data Collection with Bounded-Sized Messages in Wireless Sensor Networks,” in International Conference on Computing, Control and Networking (ICCCN), 2016, pp. 968–972.

APPENDIX

Table 5

Simulation Result Data

n	HSCS	An et al.	Worse than An et al.[5]
100	242	281	0
100	202	273	0
100	279	274	1
100	219	275	0
100	232	263	0
100	182	278	0
100	223	278	0
100	187	265	0
100	222	279	0
100	243	283	0
100	214	277	0
100	260	277	0
100	203	276	0
100	234	290	0
100	208	268	0
100	229	270	0
100	202	274	0
100	177	254	0
100	236	281	0
100	211	259	0
100	215	264	0
100	234	283	0
100	182	260	0
100	271	275	0
100	209	276	0
100	287	278	1
100	191	288	0
100	183	267	0
100	195	272	0
100	180	265	0
100	198	281	0
100	202	243	0
100	224	269	0
100	257	279	0
100	206	274	0
100	245	261	0

100	234	279	0
100	209	268	0
100	218	282	0
100	230	276	0
100	243	279	0
100	238	268	0
100	248	219	1
100	198	266	0
100	191	279	0
100	245	265	0
100	189	265	0
100	209	270	0
100	229	285	0
100	215	273	0
100	200	266	0
100	187	264	0
100	193	249	0
100	272	269	1
100	234	264	0
100	202	282	0
100	261	273	0
100	241	277	0
100	204	280	0
100	275	260	1
100	244	275	0
100	265	271	0
100	238	264	0
100	232	280	0
100	230	285	0
100	207	263	0
100	217	249	0
100	244	283	0
100	213	258	0
100	206	272	0
100	177	255	0
100	201	272	0
100	220	275	0
100	219	271	0
100	158	261	0
100	190	248	0
100	232	265	0
100	205	253	0
100	253	259	0

100	240	278	0
100	181	262	0
100	230	269	0
100	256	281	0
100	189	276	0
100	258	277	0
100	232	276	0
100	217	279	0
100	242	283	0
100	224	271	0
100	255	268	0
100	192	240	0
100	176	270	0
100	181	214	0
100	190	268	0
100	275	253	1
100	211	262	0
100	209	271	0
100	228	269	0
100	240	269	0
100	211	276	0
average	**220.37**	**269.39**	**6**
			Total
Percentage decrease on latency		**-18.1967**	

n	HSCS	An et al.	Worse than An et al.[5]
200	468	560	0
200	406	575	0
200	490	564	0
200	470	587	0
200	339	567	0
200	461	579	0
200	433	565	0
200	418	561	0
200	380	576	0
200	496	584	0
200	488	575	0
200	443	575	0
200	395	531	0
200	438	582	0
200	402	558	0
200	451	563	0
200	478	557	0
200	403	552	0
200	436	575	0
200	426	555	0
200	470	571	0
200	431	549	0
200	431	569	0
200	373	568	0
200	541	576	0
200	512	580	0
200	401	566	0
200	462	572	0
200	401	572	0
200	445	544	0
200	518	564	0
200	403	564	0
200	478	567	0
200	500	570	0
200	418	554	0
200	438	581	0
200	415	549	0
200	346	558	0
200	504	581	0
200	450	572	0
200	567	587	0

200	488	555	0
200	439	542	0
200	561	579	0
200	377	576	0
200	500	550	0
200	518	557	0
200	517	575	0
200	499	569	0
200	335	571	0
200	445	574	0
200	468	572	0
200	439	540	0
200	408	551	0
200	470	573	0
200	446	574	0
200	460	574	0
200	415	584	0
200	518	580	0
200	406	545	0
200	471	577	0
200	465	558	0
200	450	529	0
200	358	579	0
200	406	578	0
200	393	554	0
200	361	555	0
200	344	582	0
200	404	556	0
200	590	563	1
200	322	578	0
200	460	572	0
200	416	569	0
200	347	559	0
200	428	563	0
200	349	551	0
200	480	561	0
200	426	566	0
200	422	561	0
200	563	588	0
200	457	579	0
200	391	535	0
200	570	586	0
200	450	572	0

200	504	576	0
200	386	564	0
200	406	567	0
200	477	569	0
200	466	572	0
200	465	573	0
200	409	570	0
200	444	559	0
200	367	519	0
200	469	578	0
200	466	563	0
200	513	574	0
200	437	583	0
200	405	555	0
200	554	566	0
200	436	583	0
average	**444.61**	**566.38**	**1**
			Total
Percentage decrease on latency		**-21.4997**	

n	HSCS	An et al.	Worse than An et al.[5]
300	661	873	0
300	622	880	0
300	687	888	0
300	776	880	0
300	493	871	0
300	763	874	0
300	678	859	0
300	568	859	0
300	584	875	0
300	742	855	0
300	585	864	0
300	676	866	0
300	570	867	0
300	659	878	0
300	521	862	0
300	486	882	0
300	646	840	0
300	579	857	0
300	771	846	0
300	595	874	0
300	597	871	0
300	659	850	0
300	685	850	0
300	800	881	0
300	908	873	1
300	688	873	0
300	665	871	0
300	565	865	0
300	708	870	0
300	587	828	0
300	361	836	0
300	660	868	0
300	763	877	0
300	655	860	0
300	666	863	0
300	700	872	0
300	597	822	0
300	612	864	0
300	612	883	0
300	704	873	0
300	803	879	0

300	686	855	0
300	601	846	0
300	720	868	0
300	586	868	0
300	663	853	0
300	617	865	0
300	466	870	0
300	791	880	0
300	784	870	0
300	625	866	0
300	748	877	0
300	785	861	0
300	571	843	0
300	686	863	0
300	664	885	0
300	724	888	0
300	684	870	0
300	742	881	0
300	690	846	0
300	698	869	0
300	811	845	0
300	601	874	0
300	637	873	0
300	751	861	0
300	689	835	0
300	595	847	0
300	676	873	0
300	601	878	0
300	637	856	0
300	674	882	0
300	731	864	0
300	631	884	0
300	602	870	0
300	652	874	0
300	678	866	0
300	602	858	0
300	603	879	0
300	576	861	0
300	737	885	0
300	761	885	0
300	653	840	0
300	694	876	0
300	604	869	0

300	754	874	0
300	554	855	0
300	603	867	0
300	710	866	0
300	691	868	0
300	632	878	0
300	752	866	0
300	574	849	0
300	600	800	0
300	677	866	0
300	743	881	0
300	682	880	0
300	590	878	0
300	712	869	0
300	677	840	0
300	682	873	0
average	**660.17**	**865.48**	**1**
			Total
Percentage decrease on latency		**-23.7221**	

n	HSCS	An et al.	Worse than An et al.[5]
400	951	1167	0
400	924	1179	0
400	823	1189	0
400	767	1185	0
400	618	1165	0
400	959	1173	0
400	872	1141	0
400	918	1142	0
400	859	1161	0
400	754	1136	0
400	834	1168	0
400	936	1168	0
400	882	1162	0
400	772	1173	0
400	767	1170	0
400	954	1180	0
400	819	1136	0
400	794	1157	0
400	862	1160	0
400	744	1170	0
400	905	1161	0
400	858	1165	0
400	904	1147	0
400	1008	1181	0
400	946	1173	0
400	600	1177	0
400	741	1178	0
400	844	1154	0
400	956	1171	0
400	894	1162	0
400	648	1160	0
400	837	1150	0
400	832	1181	0
400	1093	1162	0
400	771	1157	0
400	802	1184	0
400	822	1156	0
400	972	1163	0
400	1039	1187	0
400	905	1160	0
400	934	1189	0

400	795	1159	0
400	916	1180	0
400	958	1158	0
400	814	1159	0
400	850	1142	0
400	1067	1168	0
400	1101	1167	0
400	732	1172	0
400	1071	1163	0
400	811	1168	0
400	1070	1162	0
400	875	1153	0
400	1174	1162	1
400	795	1166	0
400	848	1181	0
400	801	1186	0
400	1035	1178	0
400	817	1178	0
400	947	1135	0
400	1005	1183	0
400	1022	1134	0
400	881	1173	0
400	778	1170	0
400	1079	1170	0
400	817	1153	0
400	665	1131	0
400	813	1166	0
400	950	1170	0
400	1017	1178	0
400	804	1181	0
400	1060	1146	0
400	804	1181	0
400	927	1164	0
400	1033	1168	0
400	906	1162	0
400	873	1145	0
400	873	1172	0
400	866	1134	0
400	1006	1182	0
400	987	1182	0
400	899	1117	0
400	636	1174	0
400	672	1172	0

400	1106	1173	0
400	767	1173	0
400	939	1165	0
400	896	1165	0
400	864	1158	0
400	734	1170	0
400	761	1166	0
400	1018	1146	0
400	841	1167	0
400	753	1153	0
400	1056	1177	0
400	793	1175	0
400	914	1174	0
400	826	1173	0
400	639	1149	0
400	863	1169	0
average	**877.4**	**1164.98**	**1**
			Total
Percentage decrease on latency		**-24.6854**	

VITA

Ecem Simsek

EDUCATION

- M.S. in Computing and Information Science, Expected May 2019
  Sam Houston State University (Huntsville, Texas),
  Thesis Title, Aggregation Scheduling Algorithm for Directed Wireless Sensor Networks
  Supervisor's Dr. Min Kyung An

- Bachelor's Degree in Computer Software Engineering Technology, May 2017
  Sam Houston State University (Huntsville, Texas)

- Bachelor's Degree in Software Engineering, June 2015
  Firat University (Elazig, Turkey)

SKILLS

Technical Skills

- Java: intermediate, NetBeans
- Android: intermediate, Android Studio
- Database in MySQL
- C++ intermediate, OpenGL, Visual Studio
- C# Beginner, Visual Studio
- Python: Beginner, Enthought Canopy

Languages

- Turkish
- English

EXPERIENCE

Graduate Assistant | Sam Houston State University | Huntsville, TX **| 01.2018 – Current**

- Courses: Data Base Management System, Human Computer Interaction, Information System Design and Management, Programming Fundamentals I, Introduction to Computer
- Assisting with the grading exams, quizzes and assignments, Lab tutoring, Tutoring individual students

Internship | Turkish Petroleum | Ankara, TURKEY **| 06.2016 – 07.2016**

- Participated team project and create a project for the institutions in Java

Internship | Iskur | Ankara, TURKEY **| 08.2014 – 09.2014**

- Participated team project and modify the institutions of the programs in .NET

CERTIFICATION

- Occupational health and safety certification dates
- Certificate of Completion, Java 8 - Beyond the Basics
- Certificate of Completion, Python for Beginners
- Certificate of Completion, Practical Data Structures & Algorithms in java

SOCIETIES/MEMBERSHIPS

- Vice President, Turkish Student Organization (TSO) date
- Member, Sam Houston Association of Computer Scientists (SHACS) date

HONORS

- High School Honor Award
- 2016 Spring - 2017 Spring, The Dean’s List of Academic Honors, Sam Houston State University
  2017 Spring, President's List, Sam Houston State University