HEURISTIC ALGORITHMS FOR SCHEDULING MULTIPLE TIMESLOTS AND CHANNELS IN INDUSTRIAL WIRELESS SENSOR NETWORKS

BY

YANG GYUN KIM

A dissertation submitted to the Graduate Faculty in Electrical Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy
The City University of New York

2013

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy

September 09, 2013	
Date	Chair of Examining Committee Dr. Myung J. Lee, Professor, Department of Electrical Engineering, The City College of The City University of New York
September 09, 2013	
Date	Executive Officer Associate Professor Ardie D. Walser
Dr. Tarek N. Saadawi	Professor, Department of Electrical Engineering, The City College of The City University of New York
Dr. M. Ümit Uyar	Professor, Department of Electrical Engineering, The City College of The City University of New York
Dr. Yi Sun	Assistant Professor, Department of Electrical Engineering, The City College of The City University of New York
Dr. Sang-Woo Seo	Assistant Professor, Department of Electrical Engineering, The City College of The City University of New York

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

HEURISTIC ALGORITHMS FOR SCHEULDING MULTIPLE TIMESLOTS AND CHANNELS IN INDUSTRIAL WIRELESS SENSOR NETWORKS

By
Yang Gyun Kim

Advisor: Professor Myung Jong Lee

Over the last decade, wireless technologies have made substantial advancements and provided users with quick and convenient connectivity regardless of time and space limitations. Wireless technologies not only enable users to set up a network quickly and conveniently, but also allow them to set up a network. One of the prominent technologies in the wireless communication is Wireless Sensor Network (WSN). WSN has been the focus of research due to its practicalities in the wide range of applications. WSN is a highly attractive alternative to current wired networks, which are bulky, expensive, complex, and difficult to deploy and maintain. Industrial applications such as factory monitoring, control, and automation are newly emerging areas exploiting benefits of wireless sensor network technology. Similar to other wireless technologies, wireless sensor networks provide a new paradigm for factory automation with remarkable impacts on control, tracking, monitoring, and diagnostics of the manufacturing processes and equipment. As delayed information is meaningless and may cause problems in controlling processes and equipments, sensor readings from a process or a machine should be delivered in a

timely manner. Therefore, guaranteed end-to-end delay is the most critical requirement in real-time monitoring and control. A wireless sensor network, as with any other digital network, comprises multiple devices working in conjunction with each other to achieve a specific task for which the WSN is designed. In order for the multiple devices in a network to effectively communicate with each other, all of the devices must follow a common set of rules, known as Medium Access Control (MAC), which governs the complex and intricate communication processes among all devices. This dissertation focuses on Medium Access Control (MAC) protocol that provides guaranteed end-to-end delay for industrial applications, and two heuristic scheduling algorithms: one involving metaheuristics, i.e., Simulated Annealing (SA) and Particle Swarm Optimization (PSO), and the other involving a greedy heuristic for wireless industrial sensor networks with multi-channel and multi-timeslot that can achieve a guaranteed end-to-end delay.

First, we study one of the new MAC protocols in IEEE 802.15.4e standard called Deterministic and Synchronous Multichannel Extension (DSME) mode. DSME enhances the existing IEEE 802.15.4 GTS (Guaranteed Time Slot) to provide robust and timely data delivery for industrial wireless mesh sensor networks. DSME also provides two channel diversity techniques, namely channel hopping and channel adaptation, to cope with dynamic channel conditions.

Second, we implement the two metaheuristic algorithms, namely the SA and the PSO, to schedule multiple channels and timeslots in a multi-hop wireless sensor network. Timely communication in wireless multi-hop sensor networks requires high throughput and low delay, both of which can be achieved by effectively exploiting multiple channels and timeslots. Efficient scheduling becomes indispensable if multiple channels and timeslots are utilized. Optimum scheduling of multiple channels and timeslots in multi-hop networks is an NP-

complete problem. We apply metaheuristic approaches to solve the scheduling problem because of the fact that the solution we are seeking is not the global optimal and that a sub-optimal solution would suffice to guarantee a given end-to-end delay bound.

Third, we introduce a novel scheduling algorithm using multi-channel and multi-timeslot with the objective of minimizing the end-to-end delay in a tree topology-based wireless sensor network. In a tree topology, the data traffic always flows from a child (transmitter) to a parent (receiver) towards the coordinator. Since interference occurs at the receiver end, i.e., the parent, in order to cope with interference, the channel of each parent node that experiences interference is scheduled starting with the parent with the most number of interfering nodes. The algorithm exploits a staggered sequential timeslot allocation in terms of end-to-end paths to minimize end-to-end delay rather than minimizing the total number of the timeslots required by the network in terms of individual branches.

Acknowledgements

I would like to express my deepest appreciation towards the members of my doctoral committee, Professor Myung Jong Lee, Professor Tarek N. Saadawi, Professor M. Ümit Uyar, Professor Yi Sun, and Professor Sang-Woo Seo for their time and guidance. I have been privileged to have Professor Lee as my research and academic advisor during my doctoral study; he is undoubtedly the most supportive advisor anyone could ask for. He has always encouraged me to move forward in the exciting research topics, and to explore unfamiliar knowledge and complex engineering issues. I am sincerely grateful for the energy, time, and patience that he has generously provided to me. I am deeply thankful for his unreserved guidance and support over the years.

I also want to acknowledge the help and support of the fellows in Advance Wireless Networks Laboratory, particularly two of my most valued colleagues and friends, Dr. June Seung Yoon and Dr. Gahng-SeopAhn. Extensive discussions with them have always been full of inspiration and motivation. In addition, I would like to express my gratitude for the additional support and assistance I have received from other colleagues in Advance Wireless Networks Laboratory, namely Dr. Taerim Park, Dr. Rui Zhang, Mr. Muhammad Anwar Hussain, Mr. Yanchen Liu and Mr. Kazi J. Ahmed, who have always been willing to share their knowledge with me and discuss our research issues. In addition, I would like to express my special thanks to Prof. Byoung Seob Park in Inha Technical College for providing useful comments, companionship, and support that enabled me to complete this work.

I must also thank my parents whose love and support have motivated me to succeed throughout

my life. My father always stood as a living example of integrity and hard worker showing his respect towards the others. My mother devoted herself to raising her five children in spite of financial difficulty. I always consider myself to be blessed to have had such great parents who are regretfully no longer with us.

I also would like to express my deepest appreciation for my loving family, my wife, Gee-Soon, and my daughter, Zoey. In spite of all the challenges and difficulties we have been through, Gee-Soon has never lost her smiles and fully supported me even when I felt as though my goals were no longer within reach.

Although I have not been able to be the father that my warm-hearted beautiful daughter, Zoey, deserves, she has always treated me with full respect, and I thank her for helping me to grow my career as she grew up to be a lovely lady.

Finally, my greatest gratitude is towards our God, who always answers the prayers of an undeserving servant such as myself. Without the strength and courage He provides me with, I would not have been able to even dream of pursuing a doctorate degree.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Communication among people has continued to move forward ever since humans started to communicate with each other over remote distances. Over time, communication technologies, such as telephone, fiber optics, and local area networks (LANs) have emerged. One revolutionary transition of the technology converts long and cumbersome wired communication into wireless communication, which has quickly become an essential part of everyday living for almost everyone as it is being utilized in ubiquitous devices such as mobile phones, laptop computers, tablets, and portable gaming devices.

One of the most prominent technologies in the wireless communication is Wireless Sensor Networking, a still-evolving technology for various ubiquitous applications, such as habitat monitoring, environmental observation, patient monitoring, disaster warning, and intruder detection. Wireless sensors can potentially be deployed everywhere in the order of hundreds, thousands, or even tens of thousands.

Wireless sensor network (WSN) technology has seen sustained research and development efforts in a wide range of scenarios and applications over the last decade. Industrial applications, however, began employing WSN technology only recently because of their stringent Quality of Service (QoS) requirements in terms of latency and reliability. Industral applications such as factory monitoring, control, and automation are newly emerging applicable areas exploiting benefits of wireless sensor network technology. The main beneficiaries of wireless sensor

network technology are the applications in monitoring and control areas, including HVAC control, inventory tracking, data gathering, as well as home and factory automation. The most recent emerging area that welcomes wireless sensor network technology is industrial wireless sensor networks since the sensors attached to the surface or inside of the machines, conveyor belt, or utility pipelines form a tree-based network and report sampled data to a central controller periodically. Many industrial alliances such as wireless HART [1], BACnet [2] over ZigBee, and ISA100 [3] have been specifying wireless solutions for their target industries. These solutions operate in IEEE 802.15.4 [4] non-beacon mode, which provides only probabilistic performance in terms of latency as it is a contention-based MAC.

The most critical performance measure in the industrial application is deterministic and guaranteed end-to-end delay. Delayed information is meaningless and can cause serious problems in controlling processes and equipment. In order to achieve deterministic and guaranteed end-to-end delay, each sensor node would have flexbility in utilizing multiple channels and multiple timeslots to provide substantially increased throughput and expedited simultaneous data delivery. Moreover, the sensor node would need to have a repeated superframe structure in medium access control to allow the node to repeatedly use the already allocated timeslots, i.e., once a node is designated one timeslot with the guranteed delay, it continues to use that same timeslot. In addition to a superframe, each node has the capability to employ multiple channels for each individual timeslot while also increasing simultaneous data transmission within two-hop node distance.

This dissertation first introduces a medium access control (MAC) protocol for industrial wireless sensor networks to achieve determinitic and guranteed end-to-end delay. In our industrial MAC design, the allocation of multiple channels and timeslots is performed by three-

way handshaking algorithm on the fly. Second, in order to efficiently schedule multiple channels and timeslots, optimazation for the end-to-end delay using meta-heuristic algorithms, SA and PSO, has been studied to minimize the maximum end-to-end delay. Last, a greedy heuristic algorithm is studied to schedule multlpe channels at the receiver side and to schedule mulple timeslots at the sender side, minimizing the end-to-end delay for industrial wireless sensor networks.

1.2 Summary of Approaches

1.2.1 DSME

This chapter introduces a new MAC protocol in IEEE 802.15.4e standard [5], called Deterministic and Synchronous Multichannel Extension (DSME) protocol mode. While numerous MAC protocols for wireless sensor network have been designed [11]-[22], IEEE802.15.4 could be the most suitable candidate for wireless sensor network. Although it was designed for low-rate applications in wireless personal area networks (PAN), it is well-fitted for wireless sensor networks due to its low-rate, low-power, and short range characteristics as well as its use of Carrier Sensing Medium Access (CSMA). It also has the capability to guarantee Quality of Service (QoS) via Guaranteed Time Slot (GTS). Moreover, its flexible superframe structure can cope with various requirements of wide range of applications. However, adoption of IEEE802.15.4 to certain wireless sensor networks is admittedly a challenge because different scenarios and applications have distinctive requirements and levels of QoS. The new DSME enhances the existing IEEE 802.15.4 GTS to provide robust and timely data delivery in industrial wireless mesh sensor networks. First, DSME introduces fortified superframe structure, which

extends GTS in space, time, and channels while increasing a combination of using channel and timeslot within a two-hop distance. Second, DSME performs enhanced timeslot allocation mechanism in conjunction with channel allocation through collaborative three-way handshaking that results in fast and reliable data delivery to allow simultaneous transmissions with less low interference and deterministic low end-to-end delay. Third, DSME provides two modes, namely channel adaptation and channel hopping, to enhance robustness to dynamic channel conditions in noisy channel environment where the industrial devices reside. DSME can provide guaranteed QoS (Quality of Service) for real-time sensor network in harsh industrial environment by utilizing dynamically changing channels. The performance of DSME is numerically analyzed and then verified through extensive simulations to understand how its parameters should be configured. The simulations with various metrics and scenarios show that the proposed DSME attains dramatic improvement in delay, throughput, jitter and traffic capacity, making it suitable for industrial applications.

1.2.2 Metaheuristics

In the IEEE Standard 802.15.4 [4] for WSNs, a superframe structure consists of both Contention Access Period (CAP) and GTS Guaranteed Time Slot (GTS). Our proposed algorithm utilizes this superframe structure, but each timeslot is extended to accommodate multiple channels as in IEEE Standard 802.15.4e currently in development for industrial applications to guarantee end-to-end (e2e) delay [5]. The channels and timeslots available to a node vary because each node’s selection of channels and timeslots imposes a set of constraints on the channels and timeslots available to its neighbors. It is possible that some of the channel and timeslot combinations are only feasible at certain locations based on the status of each

node’s neighors. Therefore, our proposal affords each node the freedom to choose both the optimal timeslot and channel in establishing commmunication links to its neighbors, resulting in high throughput and low delay.

Scheduling is a critical process for virtually all resource-allocation problems, especially to meet certain quality of service (QoS) requirements. Scheduling channels and timeslots for all nodes constituting an end-to-end path to meet certain delay bounds (e.g., 10 msec end-to-end delay in factory automation [5]) in a WSN is a challenge because each node in the end-to-end path has different path-length and encounters dissimilar channel environments. A simple approach is to schedule channels and timeslots in a sequenced and staggered fashion from a source to a destination. This approach can reduce the end-to-end delay by assigning the first node in a path with the earliest available timeslot and the second node with the earliest available timeslot counted from the timeslot of the first node, assuming all network nodes are synchronized. However, this approach cannot be sustained when there are many simultaneous and overlapping end-to-end paths coexisting at crossing nodes because the scheduling of timeslots to meet individual path delay bounds becomes overly complicated. An example of this approach is found in the Distributed Synchronous Multi-Channel Extension (DSME) mode of IEEE 802.15.4e standard. However, optimizing the scheduling of the channels, timeslots, and nodes in wireless sensor networks subject to constraints such as interference, would allow all nodes to achieve the guaranteed minimum end-to-end delay.

End-to-end delay optimization in a multi-channel, multi-timeslot, and multi-hop wireless sensor network becomes a significant challenge in a wide area network with a large number of nodes because it is an NP-complete problem, and polynomial algorithm cannot be applied directly. Furthermore, there is no algorithm that can guarantee optimality except for an

exhaustive probe of all possible combinations of nodes/hops, timeslots, and channels across the entire search space. Considering its complexity, such probing algorithm would not be feasible in practice. As the scale of the network grows, the total number of possible combinations of nodes/hops, timeslots, and channels would increase exponentially, i.e an exhaustive search is infeasible for our problem.

A popular approach for solving the optimization problem models the network as a graph with the nodes and the links represented by the vertices and the edges of the graph, respectively. The links are colored with different colors depending on the constraints imposed on the timeslots and the channels. However, this graph coloring algorithm for scheduling optimization problem is not a suitablec choice for a practical application due to the following three issues associated with it. First, the input for each node in the graph theoretic approach would be unique because each node has different constraints. In addition, each node except for leaf node allows a multiple combination of timeslot and channel for transmitting and forwarding that produces corresponding multiple edges. Finally, applying the graph coloring algorithm to a practical problem is unreasonable due to its complexity and inflexibility.

Metaheuristic computational methods are logical choices for solving the complex combinatorial optimization problem because the solution quality and the computaion complexity are compromized to obtain a suboptimal solution rather than the optimal solution. Although meta-heuristics do not guarantee optimality due to the nature of heuristics, they provide an acceptable approximation to the global optimum within a reasonable time and measure of quality.

In the community of optimization, PSO is one of the most well-known representations of the evolutionary computation algorithms, and SA is the most well-known representation of stochastic computation algorithms. The primary reason for selecting PSO metaheuristic is due to

its directionality characteristic. The concept of directionality could be explained by the bird feeder example; if there is a bird feeder, its location becomes an indicator of the direction towards which the birds would fly. Although the exact flight path of a specific bird cannot be predicted, i.e., it takes a random path, the general direction of its path can be predicted. The e2e delay involves the sense of directionality in that in order to minimize delay, timeslots are allocated in sequence, and multiple timeslots required by a single path are assigned as close to each other as possible. Although there is no such directionality in SA because the particle movement is completely random, it is still feasible to employ directionality by truncating the movement space. For example, if a particle is restricted from moving in certain direction based on the location of the destination, we can predict that it will not move in that restricted direction. Another reason for selecting the PSO and SA metaheuristics is their relatively low complexity to implement in the discrete domain, i.e., PSO and SA provide a simple solution to a complex problem rather than a complex solution to a complex problem. Therefore, we propose to use PSO and SA to schedule collision-free multi-channel and contention-free multi-timeslot allocations in order to optimize the end-to-end delay in a tree topology environment.

1.2.3 Heuristics

Wireless sensor networks (WSN) have been gaining attention due to their various application-capable characteristics. Wireless sensors, which were initially developed to simply detect and react, have become communication-capable smart sensors, collecting and forwarding sensed data towards a data collector, namely the coordinator. Data collection in large-scale applications, such as bridge monitoring, environment monitoring, and intruder detection would require multi-hop communication because the sensors are deployed in large areas. Time-critical applications, such as patient monitoring, disaster warning, and intruder detection, require expedited delivery of data from a specific source, which may be obtained by minimizing the timeslots [7], [8]. However, industrial applications, such as process monitoring, factory automation, and factory control, require both prompt data delivery and guaranteed end-to-end delay for all possible sources of data because belated data from just one source of data may render the entire system useless.

The traffic in the industrial applications mentioned above is usually periodic, making Time Division Multiple Access (TDMA) approach feasible. TDMA allocates exclusive collision-free timeslot(s) to each node, and minimizes end-to-end delay (e2e) between a source node and the coordinator. Each TDMA timeslot is extended to accommodate multiple channels as in the IEEE Standard 802.15.4e [5] to guarantee e2e delay. Then, each node has the flexibility of utilizing a combination of channel and timeslot in order to achieve expedited data packet transmission towards the coordinator, resulting in minimum e2e delay. Distributed Synchronous Multi-channel Extension (DSME) of IEEE802.15.4e standard [5], which has been released as a new IEEE Media Access Control (MAC) standard in 2012, is one of the MAC operation modes that guarantee e2e delay for the industrial applications. There are inherently two factors that hinder

efficient data collection in WSNs: half-duplex nature of transceiver and interference [8]. In WSNs, each sensor node is typically equipped with a single half-duplex radio transceiver, i.e. each node cannot transmit and receive simultaneously, nor can it function on different channel at the same time due to their resource-constrained attributes. To cope with interference, we assign different channels at a receiver side on interfering links (parents) which are defined by interfering parents through the distances of parent and child [8]. After determining interfering links in an arbitrary network, all the interfering links are eliminated by assigning different channel a pair of interfering links as long as there are an enough number of available channels. For example, there are 16 available channels in IEEE Standard 802.15.4e, and literally, 16 pairs of interfering links are eliminated. The most interfered receiver is assigned with the first available channel, and the next interfered receiver is assigned with next available channel, and so on until available channels are all used. If available channels are not enough, the remaining interfering links can be considered by allocating timeslots with interference constraints at the sender side, thereby transmitting data packet on the designated timeslot using that channel. The problem of formulating collision-free TDMA schedules even under the simple graph-based interference model has been proved to be NP-complete [8]. The adoption of multiple channels into each TDMA makes the scheduling even more challenge. Therefore, we propose a greedy heuristic scheduling algorithm for channel and TDMA allocations to minimize guaranteed e2e delay.

Chapter 2

DMSE

2.1 Introduction

Wireless sensor network (WSN) technology has seen sustained research and development efforts in a wide range of scenarios and applications over the last decade. Industrial applications, however, began employing WSN technology only recently because of their stringent Quality of Service (QoS) requirements in terms of latency and reliability. Many industrial alliances such as wireless HART [1], BACnet [2] over ZigBee, and ISA100 [3] have been specifying wireless solutions for their target industries, e.g., factory monitoring, control, and automation. These solutions operate in IEEE 802.15.4 [4] non-beacon mode, which provides only probabilistic performance in terms of latency as it is a contention-based MAC. Deterministic low latency is one of the most important and challenging requirements in real-time monitoring and control. Belated information is meaningless and it can also cause serious problems in controlling processes and equipments.

IEEE 802.15.4 is considered as the first global standard for WSN which specifies the physical layer and the MAC sub-layer for low rate, low power wireless communications. It has two MAC operation modes: beacon-enabled and non-beacon modes. IEEE 802.15.4 beacon-enabled mode provides contention free access using Guaranteed Time Slot (GTS) scheme to provide QoS guarantee. However, its advantage is limited as IEEE 802.15.4 beacon-enabled mode supports

only up to seven slots, which is not sufficient for medium to large scale networks. GTS is supported only within one-hop from the PAN (Personal Area Network) coordinator and thus it is not applicable for multi-hop networks. Moreover, GTS utilizes a single channel, which is vulnerable to dynamic channel conditions expected in many factories, warehouses, buildings, and etc. These indoor environments can pose severe shadowing and multi-path effect caused by many artifacts which have different reflectivity and permissibility [9].

IEEE 802.15.4e [5] has been released as a new IEEE MAC standard in the year 2012 to address, among others, the deficiencies of GTS scheme in IEEE 802.15.4 beacon-enabled mode. DSME (Deterministic and Synchronous Multichannel Extension) is one of the MAC operation modes in IEEE 802.15.4e, aimed at industrial WSN applications, including process automation, factory automation, smart logistics, and smart grid. Major features of DSME are briefly summarized below.

First, DSME introduces a new versatile superframe structure which extends GTS in space, time, and channel. DSME supports multi-hop mesh connectivity to extend GTS in space. For time extension, DSME provides Multi-superframe extension and Contention Access Period (CAP) reduction to increase the number of available timeslots. DSME also allows timeslots in multiple channels to increase the number of timeslots, and enhance robustness in dynamic channel conditions.

Secondly, DSME performs enhanced timeslot allocation mechanism in conjunction with channel allocation through collaborative Three-Way-Handshaking (TWH). The TWH informs the neighbors of the sender and the receiver of the allocated channel and timeslot to avoid interference while allowing spatial reuse of the slot. The scheduling with the knowledge of channel and timeslot occupancy guarantees low end-to-end delay.

Finally, DSME offers two channel diversity techniques, namely channel hopping and channel adaptation, to enhance robustness in the face of dynamic channel conditions. Channel hopping and channel adaptation in guaranteed timeslots would prevent packet loss caused by channel quality degradation due to environmental effects.

Although DSME is designed with the requirement and challenges of industrial applications, its concepts and performance have not been studied comprehensively. The analysis in the literature [10] is very limited as it is based on a square-grid topology for multi-hop and a star topology for single-hop without evaluating the effects of channel diversity in dynamic channel conditions.

In this paper, we present OPNET simulation studies to validate the effectiveness of DSME's approaches in providing robust and timely data delivery in multi-hop industrial wireless sensor networks. We provide our insight on configuring the parameters of DSME to meet the requirements of delay-sensitive industrial applications. Furthermore, we compare the performances of channel hopping and channel adaptation under dynamic channel conditions, using two state Markov channel model.

2.2 Literature Review

In general, a MAC in WSN domain, has garnered a lot of attention from researchers and markets for more than a decade, and there are numerous on-going developments on medium access protocols, such as SMAC, TMAC, BMAC, XMAC, and so on [11]-[14]. These MAC protocols focus on a single transceiver along with a single channel, and the current trend of the design goal is energy efficiency, which is the primary concern for energy-constrained sensor devices. When WSNs were first developed, a single channel was utilized in order to facilitate low power operation in devices with highly limited resources. Furthermore, the radio bandwidth

is very limited compared to that of Wireless ad hoc network (IEEE 802.11) families. In WSNs, energy efficiency is the primary concern because its nodes operate autonomously on small batteries with highly limited power capacity, and minimizing delay, maximizing bandwidth utilization, and maximizing throughput are of secondary concern. In recent years, a single transceiver along with multiple channels concept has reshaped WSNs, and this concept is widely utilized in commercial devices such as MICA2, Telos, CC1000 and CC2420, all of which support multiple channels. A typical sensor device in WSNs is equipped with a single half-duplex radio transceiver, which is incapable of simultaneous transmission and reception. If the channels are orthogonal, i.e. the channels do not overlap with each other, simultaneous transmissions can take place on multiple channels without incurring interference.

A multi-channel MAC in Ad hoc networks [15] employs two half-duplex transceivers: control transceiver for control channel and data transceiver for data channel. During the control channel, RTS/CTS/RES transaction is performed to reserve a data channel for the sender and the receiver. It is based on CSMA/CA, IEEE 802.11, and the channel negotiation method is similar to the three-way handshaking EGTS but its components and functionalities are quite different. One of the components multi-channel MAC uses is the channel usage time, which notifies the other nodes of the beginning time and the end time of a node's transmission, but it might be difficult for the nodes to coordinate these times accurately in a combined-application network (a network that contains nodes performing different functions using different applications) because of clock drifts in a large number of nodes and varying packet sizes. Clock drift is especially critical in a WSN because its value (40us per second [TMCP]) is non-negligible. In an overlapping (combined-application) network, NAV(Network Allocation Vector) of IEEE 802.11 MAC access protocol must know the accurate values of the beginning and the end time, but if data

packet lengths vary, it would be difficult to accurately calculate the beginning and the end time. If the accurate beginning and the end time are not known, loose NAV causes overall system collapse. The TWH-EGTS enables two-dimensional negotiation for channel and slot allocation while multi-channel MAC in Ad hoc networks negotiates only for the channels. In addition, the channel assignment in RTS/CTS/RES is based on the receiver, in which the transmitting node tunes its channel to the intended receiver's channel.

TMMAC [16] utilizes a single radio transceiver, but it divides each frame into a common channel and multiple data channels. It is based on [15], which means one unique slot and one unique channel is assigned to each node. The frame structure is adopted from IEEE 802.11 PSM (Power Saving Mode), which composes of two windows after a beacon: one for ATIM (Ad hoc Traffic Indication Messages) and the other one for Communication. During the ATIM window, channel and slot negotiation performs via the common channel for data transmission in subsequent communication window. Channel and slot negotiation procedures in TMMAC are similar to those in [15] and THW-EGTS, and both ATIM/ATIM-ACK/ATIM-RES in TMMAC and THW-EGTS utilize the two-dimensional negotiation. However, in THW-EGTS, slot selection is given a higher priority than channel selection while TMMAC gives the same priority to channel selection and slot selection based on the number of available slots and channels. In TMMAC, the slot and the channel combination is selected at random without considering the location of the slot whereas in THW-EGTS, a node is first assigned the earliest possible slot before a channel is assigned because choosing the earliest possible slot minimizes delay. TMMAC is not suitable for a larger number of nodes in close proximity because increasing the number of slots causes delay and leads to difficulty in time synchronization. Furthermore, IEEE 802.11 PSM does not support multi-hop communication owing to the attribute of MAC and

beacon collision. TMMAC is not suitable for a larger number of nodes in close proximity because increasing the number of slots causes delay and leads to difficulty in time synchronization. Furthermore, IEEE 802.11 PSM does not support multi-hop communication owing to the attribute of MAC and beacon collision.

MMAC [17] is contention-based, and a single transceiver with a common channel for channel negotiation is utilized. The channel negotiation process is performed by ATIM/ATIM-ACK/ATIM-RES, and the frame structure is adopted from 802.11 PSM. Once the channel assignment has been completed, the switching of channel is bounded by beacon interval. If a specific application uses a large beacon interval, bandwidth waste becomes considerable, and if the amount of data is low, there is a long idle period that could deprive other nodes of accessing that channel. Once the channel negotiation process has been completed, RTS/CTS procedure is executed prior to real data exchange. Since a WSN has relatively small data packet size, the overhead of using RTS/CTS packets becomes prominent especially when the data packet size is small [18].

MC-LMAC [19] is defined by a timeslot that consists of a control period and a data transmission period. All the nodes wake up at the beginning of each slot and change their channels into a common channel during a common-frequency period, which causes frequent channel switching even if a node does not have any data to send. During the common period, a sender informs the intended receiver about its channel usage, and the receiver switches its transceiver to a receive mode during the data transmission period. According to the authors of [20], in order to avoid the overhead, a control period is much smaller than the data period for transmitting multiple packets. However, if the duration of the data period is increased for multiple packets (not for a single packet), then the time wasted by a node waiting for its next turn

could be lengthy and problematic as the number of nodes within a two-hop area increases. In addition, assignment of the same number of slots to each node regardless of the number of packets a node has leads to unfairness to nodes with relatively large number of packets to send, and underutilization of bandwidth by nodes with relatively small number of packets. For time slot allocation, the whole duration of a frame, which depends on the number of nodes in a two-hop area, is needed for obtaining a list of the free slots. A node chooses one of the free slots, and notifies its neighbors of its slot. By doing so, a time slot conflict can incur because if there are two nodes in close-enough proximity collecting the same slot information, the probability of conflict is significant because each node chooses its own slot independently without negotiating with its neighbors, and a few control frames would be needed in order to overcome this slot selection conflict.

In MMSN [21], a time slot consists of a broadcast contention period and a transmission period. During each time slot, each node competes for medium access using a contention-based procedure since each node shares the same channel. The behavior of the medium channel access depends on whether a node has a broadcast or unicast packet. For a broadcast packet, the designated channel is available, and for a unicast packet, two channels are assigned for transmitting (destination channel) and receiving. Before a node transmits a packet, it alternates between the intended-destination channel and its own receiving channel, and this method is called a snooping mechanism. In order to transmit packet, a node must change channel three times per slot because it must access the beacon channel, the destination channel, and the receiving channel within each slot, causing channel switching delay and excess energy consumption. Even after the transmission, the node switches the channel back to the beacon channel. Furthermore, at the beginning of the transmission period, a node having data must

perform a backoff. When the traffic rate is low, the backoff yields extra delay, and when the traffic is high, chance of transmission gradually gets low, and the number of retransmissions increases. In IEEE 802.11, a node in the backoff period is able to sense any activity that could occur, but in IEEE 802.15.4, a node in the backoff period can only backoff. The authors claim that the preamble bytes of the physical layer protocol data unit (PPDU) could be transmitted. However, from a practical point of view, it is unclear whether this transmission is feasible or not.

TMCP [22] is a tree-based multi-channel protocol to support convergecast traffic as a data collection application. It divides a whole network into multiple subtrees by considering interference among subtree, in order to assign different channels to different subtrees, instead of nodes. TMCP could work well with a small number of available channels which are caused by the interference from co-existing ISM band device, 802.11 networks. When the number of nodes increases, contention within substree which uses on the same channel become problematic and broadcast communication is hard because of the partition of subtree.

2.3 DSME Mode

We studies one of the new MAC protocols in IEEE 802.15.4e standard, called DSME (Deterministic and Synchronous Multichannel Extension) mode. DSME enhances the existing IEEE 802.15.4 GTS (Guaranteed Time Slot) to provide robust and timely data delivery for industrial wireless mesh sensor networks. DSME provides two channel diversity techniques, namely channel hopping and channel adaptation, to cope with dynamic channel conditions. Through simulations, we evaluate the performance of DSME and determine the optimal configuration for DSME parameters. Performance comparison of the two channel diversity

techniques is also presented.

2.3.1 IEEE 802.15.4

IEEE 802.15.4 protocols have already been specified as a guideline for the industry to enable global compatibility among devices. Numerous companies are already in the business of manufacturing IEEE 802.15.4 commercial products due to a promising market demand for low rate applications, such as wireless sensor network applications, which have recently been gaining momentum because of the feasibility of diverse applications, ranging from low data rate event driven monitoring application to high data rate real-time industrial application.

To provide global availability, the IEEE 802.15.4 devices use the 2.4 GHz industrial-scientific-medical (ISM) unlicensed band. IEEE 802.15.4 devices are exposed to a high level of interference because of the free access to the ISM band. On the other hand, the ISM 868 MHz and 915 MHz bands are available in Europe and North America respectively. A total of 27 channels with three different data rates are defined for the IEEE 802.15.4: 16 channels with data rate of 250kbps at the 2.4GHz band, 10 channels with a data rate 40kbps at the 915MHz band, and 1 channel with a data rate of 20kbps at the 868 MHz band. The choice of the data rate depends on the energy and cost efficiency requirements of the application. For example, while 250kbps is required for some computer peripherals (multimedia) and interactive games, a lower data rate like 20kbps can satisfy the requirement of many other envisioned devices, such as sensors, actuators, smart tags, and some consumer electronic products. The three types of topologies are specified in the IEEE 802.15.4: [3] a one-hop star, a multi-hop peer-to-peer topology, and a hybrid of both: cluster-tree network topology. The logical structure of the peer-to-peer topology, however, is defined by the network layer, i.e. the Zigbee Alliance, which

specifies high-level-layer communication protocol stacks on top of IEEE 802.15.4 MAC.

The IEEE 802.15.4 MAC emerges in either beacon-enabled mode or non-beacon-enabled mode. In beacon-enabled mode, a coordinator broadcasts beacons at the very first slot of each superframe periodically in order to synchronize the associated devices. A device performs slotted CSMA/CA prior to transmitting packets in the beacon-enabled mode while a device performs non-slotted CSMA/CA in the non-beacon-enabled mode. In non-beacon-enabled mode, a coordinator does not broadcast beacons periodically. Instead, the coordinator just transmits a unicast beacon to soliciting devices or for constructing topology.

A superframe structure is utilized in beacon-enabled mode and the format of the superframe is determined by the coordinator. From Fig 2.1, a superframe consists of an active part and an optional inactive part, and the size of both parts can be bounded by beacon period, which is the time-difference gap between two consecutive beacons. The length of a superframe (i.e. beacon interval, BI) and the length of its active part (i.e. superframe duration, SD) are determined by the beacon order (BO) and superframe order (SO), respectively. BI and SD are defined as: BI = aBaseSuperframeDuration*2BO, SD = aBaseSuperframeDuration*2SO. Then, the duty cycle of the superframe is bounded by SD/BI =2SO-BO. The active part of a superframe is divided into sixteen equally-sized aNumSuperframeSlots, regardless of the size of BO and SO. In other words, the ratio of each slot size is proportional to the size of BO and SO while the number of slots remains constant. The first slot of sixteen is reserved for beacon frame wherever beacon frame is necessary.

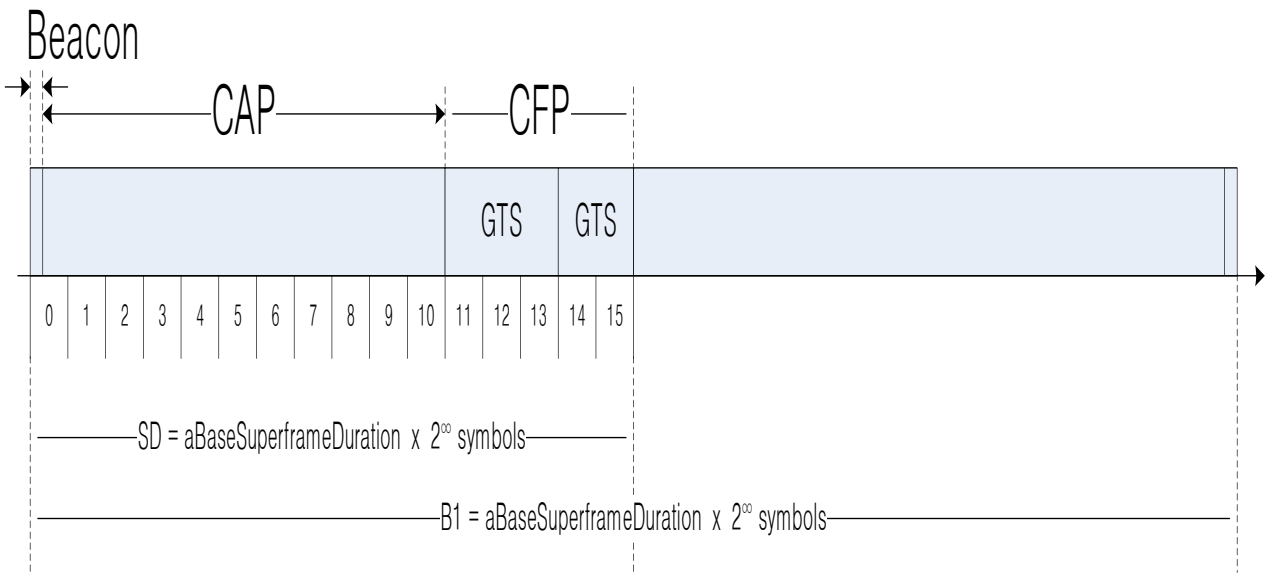


Figure 2. 1 IEEE 15.4 Superframe structure

The active part can be further classified into two periods: a contention access period (CAP) and an optional contention-free period (CFP). The optional CFP may accommodate up to seven Guaranteed Time Slots (GTSs) to provide QoS and a GTS may occupy more than one slot or starving node may request all slots. A sufficient portion of the CAP should remain available for contention-based access of other networked devices or new devices wishing to join the network(aMinCAPLength,440 symbols). All contention-based transactions should be completed before the CFP begins. Moreover, all transactions using GTSs should be finished prior to the time of the next GTS or the end of the CFP.

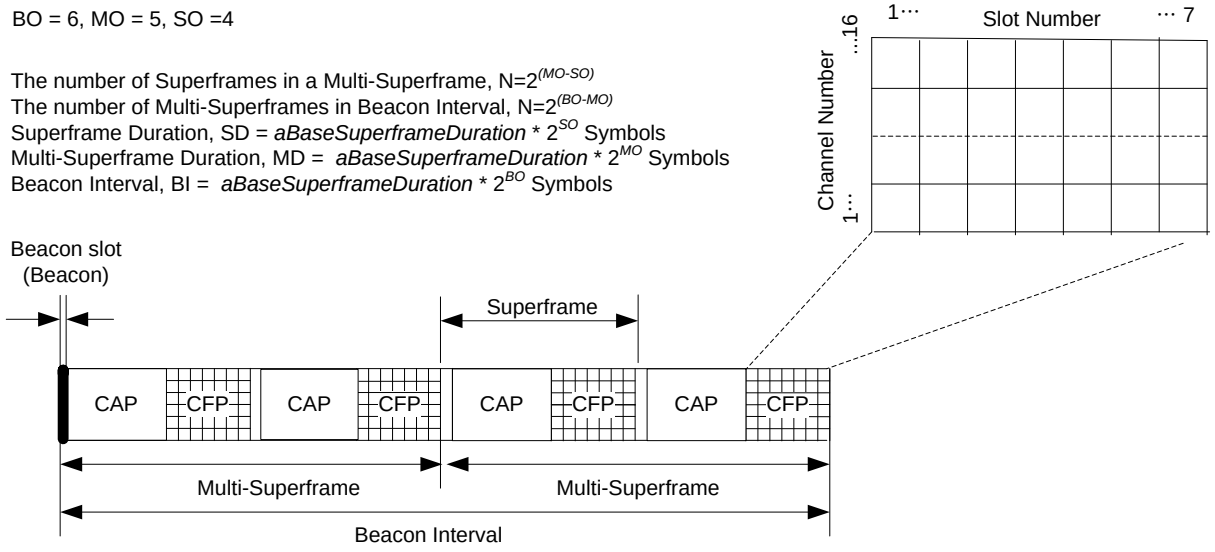
In order to serve time-critical-data transfers, IEEE 802.15.4 MAC provides a Guaranteed Time Slot (GTS) mechanism in a single-hop. A node that is already associated with PANC (PAN Coordiantor) requests GTS slot from PANC during CAP via GTS request command frame. Once PANC receives the request for GTS slot, it first determines if all seven GTS slots have been allocated, and if *aMinCAPLength* is valid. If both of these conitions are satisfied, the PANC

selects GTS slot from the reverse sequence number slot that is currently available (15,14,13….), and assigns the GTS slot based upon the number of slots being requested. The specific GTS slot number and the number of GTS slots being requested is conveyed on beacon frame at a subsequent beacon period. If either one of the above two conditions fail, then PANC stores the related information of GTS request command frame in its GTS allocation table for four consective beacon periods (aGTSDescPersistenceTime). If one of the GTS slots becomes available during the four consective beacon periods, PANC assigns the slot to the node that requested GTS slots, and the specific GTS slot number and the number of GTS slots is conveyed by the beacon frame. If none of the slots become available during the four consective beacon periods, PANC will disregard the request, and the node still having GTS data should re-transmists the GTS request command frame.

2.3.2 Multi-Hop, Multi-Superframe, and Multi-Channel Extension

DSME relieves the limitations of the IEEE 802.15.4 GTS scheme by introducing a new multi-superframe structure using multiple channels and supporting multi-hop mesh topology. Note that the GTS in DSME mode is called DSME-GTS. To support DSME-GTS in multi-hop mesh topology, DSME specifies beacon scheduling and time synchronization methods. In DSME beacon scheduling, each device schedules its beacon slot in one of the superframe in a beacon period and notifies the neighbors via broadcasting. If two neighboring devices schedule the same beacon slot, they reschedule the beacon slot to resolve the conflict. The time synchronization is performed by checking the timestamp in the beacons from the time synchronization parent device, which is determined at the upper layer. In IEEE 802.15.4, a superframe consists of beacon, contention access period (CAP) and contention free period (CFP) in sequence. In CFP,

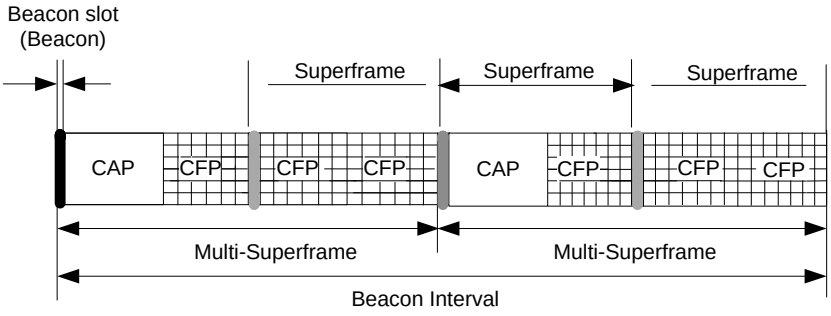
only up to seven slots are allowed. DSME extends the number of available slots by allowing multi-channel use. A multi-superframe in DSME is a cycle of repeated superframe, each one of which consists of a beacon frame, a CAP and a CFP period in sequence as shown in Fig. 2.2a. The new multi-superframe enables a node to connect with more than seven neighbors by using repeated superframes and/or multiple channels. The cost of extending the multi-superframe is the delay because the interval between two transmission opportunities for a reserved slot is determined by the size of a multi-superframe. Given the delay requirement and slot allocation requests, DSME allows the network administrator to configure the size of multi-superframe.



2.2 a).Structure of multi-superframe and of multi-channel

BO = 6, MO = 5, SO =4

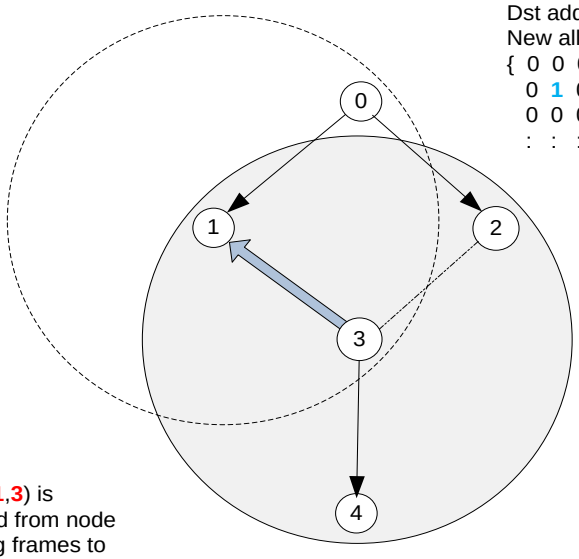
The number of available DSME-GTS slots in a Multi-Superframe
Without CAP reduction = 16 *channels* * (7 * $2^{(MO-SO)}$) *timeslots*
With CAP reduction = 16 *channels* * (7 + ($2^{(MO-SO)}$ -1) * 15) *timeslots*



2.2 b).Structure of CAP reduction

Slot = Tuple (timeslot, channel)

Every node that hears the broadcasts updates its ABT



Assuming slot (**1**,**3**) is already assigned from node 4 for transmitting frames to node 3

Node 1 assigns slot (**2**,**2**) for Node 3

2. DSME-GTS reply, *broadcast*
Payload:
Dst addr (3)
New allocated ABT sub-block
{ 0 0 0
0 **1** 0
0 0 0
: : : }

1. DSME-GTS request, *unicast*
Payload:
Number of slots
ABT sub-block
{ 0 0 **1**
0 0 0
0 0 0
: : : }

3. DSME-GTS notify, *broadcast*
Payload:
Dst addr (1)
New allocated ABT sub-block
{ 0 0 **1**
0 **1** 0
0 0 0
: : : }

2.2 c). TWH for DSME-GTS algorithm

Figure 2.2 Diverse Characteristics of DSME MAC Mode

In order to accommodate an increase in slot allocation requests, DSME can increase the number of timeslots by reducing CAP. The CAP reduction is achieved by replacing the CAP with GTS slots in the superframes while the very first CAP remains in the first superframe as illustrated in Fig 2.2b. Therefore, many nodes requiring timely delivery can easily be accommodated.

2.3.3 Three-Way-Handshaking (TWH) Channel and Timeslot Assignment

Three command frames are required for Three-Way-Handshaking (TWH) algorithm to negotiate for the timeslot and the channel: DSME-GTS request, DSME-GTS reply, and DSME-notify. Each node maintains a local Allocation Bitmap Table (ABT) that contains current timeslot information. The rows and the columns in the ABT represent timeslots and channels, respectively. For instance, when a single bit is set to 1, the specified slot is already allocated by a node itself or its neighbors. On the other hand, when this bit is set to 0, the specified slot is available for use. Fig. 2.2c briefly illustrates an example of the TWH procedure. Three channels are used here because in practice, there are three non-overlapping channels for IEEE 802.15.4 devices in 2.4 Ghz ISM band when coexisting with IEEE 802.11 networks. In this example, we assume that the timeslot number 1 and the channel number 3 are already assigned for the transmission from node 4 to node 3. As shown in Fig. 2.2c, the communication range of node 3 includes node 4, node 2, and node 1. Node 3 has data to transmit to node 1, and it transmits DSME-GTS request by unicast containing its ABT that is marked as 1 (1, 3). After receiving an DSME-GTS request, node 1 rearranges its ABT by performing an ‘OR’ operation on the ABT from node 3 and its own ABT, and selects a new ABT (2, 2) that is marked in the second row for

the timeslot and the second column for the channel. Node 1 replies with DSME-GTS reply that conveys the new ABT specifying the newly allocated slot information via broadcast. After receiving the ABT from node 1, node 3 rearranges its own ABT by performing an 'OR' operation on the new ABT (2, 2) and its current ABT (1, 3) prior to broadcasting the DSME-GTS notify frame containing the ABT information it received from node 1 (2, 2) in order to allow its neighbors to update their own ABTs.

2.3.4 Channel Diversity

For industrial applications, DSME provides two types of channel diversity schemes, namely channel hopping and channel adaptation, to enable reliable communication in a noisy environment since there is a possibility that some of the channels are still available.

2.3.4.1 Channel Hopping

If a large number of sensor devices is deployed densely or these devices are in close proximity of heterogeneous RF devices, such as Bluetooth and IEEE 802.11 which use the same spectrum, channel interference could adversely affect the network operation. Channel interference may be mitigated by receiver-based channel hopping. If a channel goes to deep fading and some other channels are in good condition, channel hopping would prevent consecutive loss of packets by switching to another channel according to the hopping sequence, which is pre-determined for each node by a different starting channel number to hop channels [5]. For example, assume there are three nodes, each of which has 15 channels available and channel offset is one. Node 1 hops channels starting from channel number 1 through 15 to channel number 1 while node 2 starts from 2 through 15 and 1 to 2, and node 3 starting from 3 through 15, 1, and 2 to 3. We utilize

this channel hopping TWH phase and ABT for allocation of unique combination of timeslot and channel offset value within two-hop region. Every transmitter follows a hopping sequence by its unique channel offset in the corresponding timeslot, which determines transmission frequency. Thus, in the allocated timeslot of every multi-superframe, the node transmits a packet via a different channel of the hopping set governed by the channel offset, and repeats after every 15 multi-superframes as we have 15 channels.

2.3.4.2 Channel Adaptation

In the channel hopping, the transmitter or the receiver has no knowledge about the perceived quality of a channel due to hoping from one channel to another according to the known sequence in every consecutive multi-superframe. Therefore, the transmitter or the receiver may not be able to avoid low quality channels and experience channel degradation [5]. In channel adaptation, the node can determine the quality of the channels by calculating the success rate of the transmitting channel. For example, if the success rate falls below a certain threshold, the channel will be switched to another one. In order to adapt channels efficiently, three tables for counting transmitted packets, received packets, and exponentially weighted moving average (EWMA) history of success rate of a channel in the corresponding timeslot are required [23]. The EWMA keeps track of current statistics in a measurement window as well as weighted older statistics. We use window mean with EWMA, which has two tuning parameters [23] $\alpha \in (0,1)$ and the observation period. The success rate is updated after the observation period (e.g., after every 5 transmissions). If the success rate falls below the threshold, new DSME-GTS request is triggered. In this request, the existing slot (timeslot + bad channel) is provided in command payload in addition to the ABT.

2.4 Delay Bound Analysis

Deterministic delay bound is the most important requirement of industrial wireless sensor networks. The delay bound of IEEE 802.15.4 GTS in one-hop star topology has been analyzed in [24] using Network Calculus theory [25], [26]. Now, we analyze the delay bound of DSME-GTS in multi-hop mesh topology using this theory.

In industrial wireless sensor networks, the sensor devices typically generate a periodic data flow with slight jitter. Following the arrival model in [24], let us consider a data flow represented as a cumulative arrival function R(t) that is upper bounded by the linear arrival curve.

$$A(t) = b + r \cdot t \tag{2.1}$$

The arrival curve A(t) represents the maximum number of bits that can be generated by a sensor device in interval [0,t] as illustrated in Fig. 2.3.

Note that only the first hop allows bursty arrival while the next consecutive hops does not. Hence, the latency of the consecutive hops are deterministic. We will first derive the delay bound of the first hop transmission, and then add the deterministic delay of remaining hops to derive entire end-to-end delay.

The delay bound of first hop can be derived by comparing the arrival curve of the data flow and the service curve of DSME-GTS. The arrival curve is presented as (2.1) and the service curve is the minimum number of bits that are assured to be transmitted in time interval [0, t].

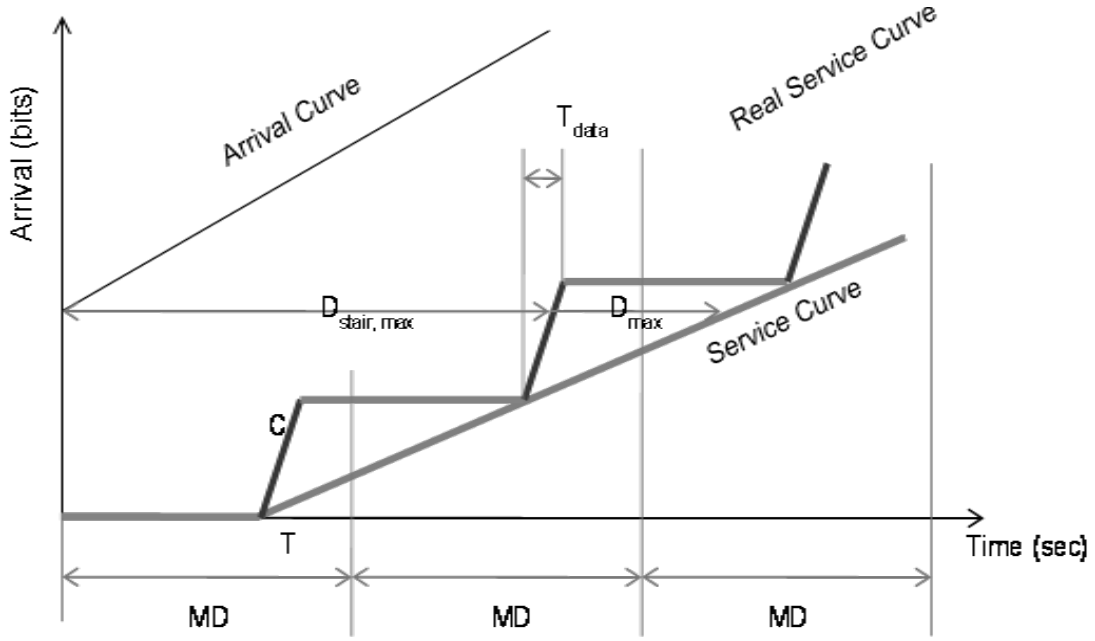


Figure 2.3. Graphical interpretation of arrival and service curves of DSME-GTS

First, we need to define the actual service time during a multi-superframe. More than one data frame can be transmitted in each DSME-GTS duration. If the remaining time is not enough to complete a transmission of another data frame, the transmission of the data frame is postponed until the next allocated DSME-GTS in the next multi-superframe. We denote the duration of a time slot as T_{slot} and the actual service time in an DSME-GTS as T_{data}. Note that T_{data} is always smaller than T_{slot}.

Secondly, we should define the maximum latency T that a data frame may wait before the transmission. The maximum latency happens when the data frame arrives just after the end of the assigned DSME-GTS. Hence, the maximum latency is calculated as

$$T = MD - T_{slot} \tag{2.2}$$

where MD is the time duration of a multi-superframe.

Let us define the k-th multi-superframe as the time interval [(k-1)MD, k·MD]. Then the service curve in the first multi-superframe is:

$$S^1(t) = \sup[C\{t-(MD-T_{slot})\}] \quad \text{if } 0 \le t \le MD - (T_{slot} - T_{data})$$

$$= 0 \quad \text{otherwise}$$

The service curve in the second multi-superframe is:

$$S^2(t) = C\cdot T_{data} + \sup[C\{t-(2\cdot MD-T_{slot})\}] \quad \text{if } MD \le t \le 2\cdot MD - (T_{slot} - T_{data})$$

$$= 0 \quad \text{otherwise}$$

The service curve in k-th superframe can be derived by recurrence as:

$$S^k(t) = (k-1)\cdot C\cdot T_{data} + \sup[C\{t-(k\cdot MD-T_{slot})\}]$$

$$\text{if } (k-1)\cdot MD \le t \le k\cdot MD - (T_{slot} - T_{data}) \qquad (2.3)$$

$$= 0 \quad \text{otherwise}$$

Then, the overall service curve $S_{stair}(t)$ is derived as:

$$S_{stair}(t) = \sum S^k(t) \quad \text{if } (k-1)\cdot MD \le t \le k\cdot MD - (T_{slot} - T_{data}) \qquad (2.4)$$

The overall service curve $S_{stair}(t)$ is a stair function as shown in Fig. 3.5.

For simplicity, the real service curve $S_{stair}(t)$ can be approximated as a linear function $S_{approx}(t)$ with a slope parameter R which is expressed as

$$R = T_{data}\cdot C / MD \qquad (2.5)$$

The slope parameter R represents the amount of bits that are transmitted in each multi-superframe.

According to Network Caculus theory, the delay bound D_{max} for a data flow with the arrival curve A(t) and the service curve $S_{approx}(t)$ is the maximum horizontal distance between A(t) and $S_{approx}(t)$:

$$D_{max} = \sup[\inf\{\tau \geq 0: A(s) \leq S_{approx}(S+\tau)\}] \quad (2.6)$$

as illustrated in Fig. 3.5.

The first hop delay bound can be computed as:

$$D^1_{max} = b / R + T = b \cdot MD / T_{data} \cdot C + T \quad (2.7)$$

Finally, the end-to-end delay bound can be derived simply by adding the constant D^C which represents the total delay of the consecutive hops. Hence the end-to-end delay is bounded by

$$D^E_{max} = b \cdot MD / T_{data} \cdot C + T + D^C \quad (2.8)$$

2.5 Performance Evaluation

We evaluate the end-to-end delay (e2e) performance of DSME in terms of its average, standard deviation, and cumulative distribution function (CDF) with CBR and Poisson traffic models for tree topologies. The simulation is carried out on a 100m x 100m field with 50 nodes distributed randomly, and their communication range is 13m. Simulation runs for 900 sec along

with a 3Kbits buffer size at each node. In order to ensure a reliable data transaction, a buffer-overflow issue is disregarded for a time-critical/reliable application due to large enough buffer size to handle data rate. A single packet for DSME-GTS data corresponds to a timeslot considering transmission time, propagation time, and channel switching time. The actual packet size is 35 bytes and the length of a timeslot is 1.92ms. The values of BO and SO are 4 and 1, respectively.

We select sources randomly, and their destination converges into the PAN coordinator, which is the typical scenario for data collection in WSNs. Since any node in the topology can generate data and forward them to PAN coordinator, it should perform a TWH for itself when it is a source and for its attached children when it is a forwarder. The number of hops from the sources to PAN coordinator ranges 2 to 6. The formation of tree topology is performed with breadth-first search, the order of which is random [27]. The beacon scheduling and DSME-GTS allocation are performed as described in DSME overview section.

2.5.1 Supported Number of Paths

In the simulation, we can see the inherent limitation of IEEE 802.15.4, which uses a single channel and supports only seven GTS slots in a superframe. In a tree topology, any node is a source and the destination of all sources is the PAN coordinator; therefore, the number of source nodes corresponds to the number of paths to a single destination, the PAN coordinator. Each source node requires one GTS slot, while each intermediate node requires two slots: one for receiving data from a previous node and the other for forwarding the data to the next node.

In order to overcome the above two deficiencies for IEEE802.15.4, we study Multi-superframeOrder(MO), which allows multi-superframe in a beacon interval that can extend the total number of GTS slots beyond seven.

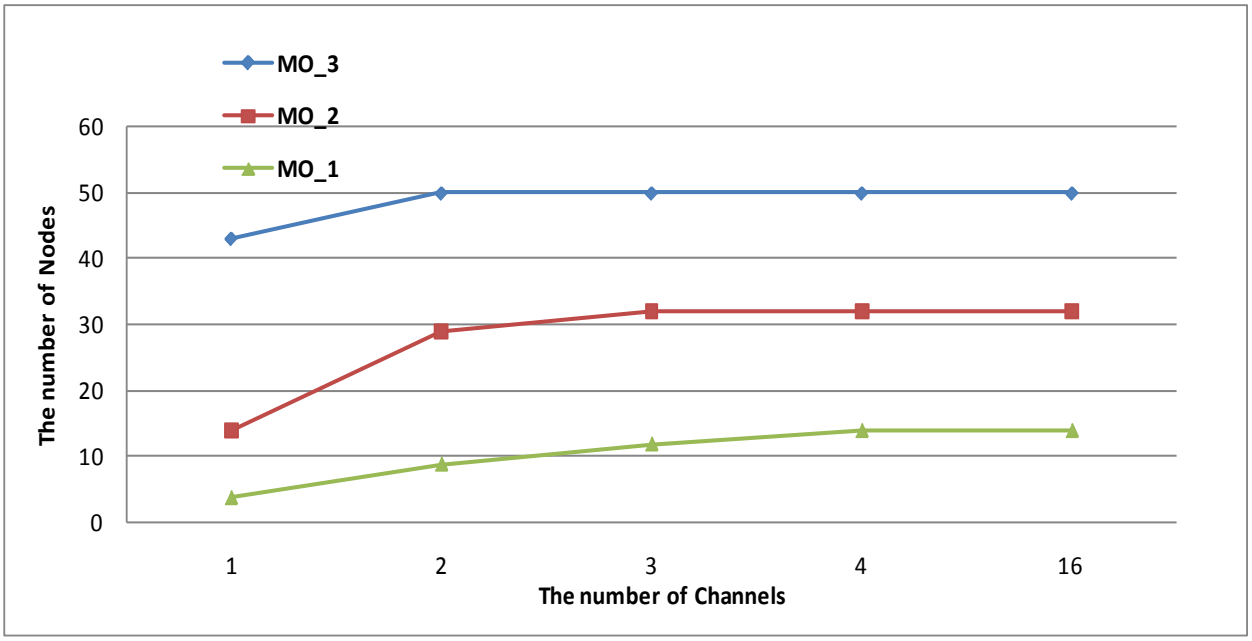


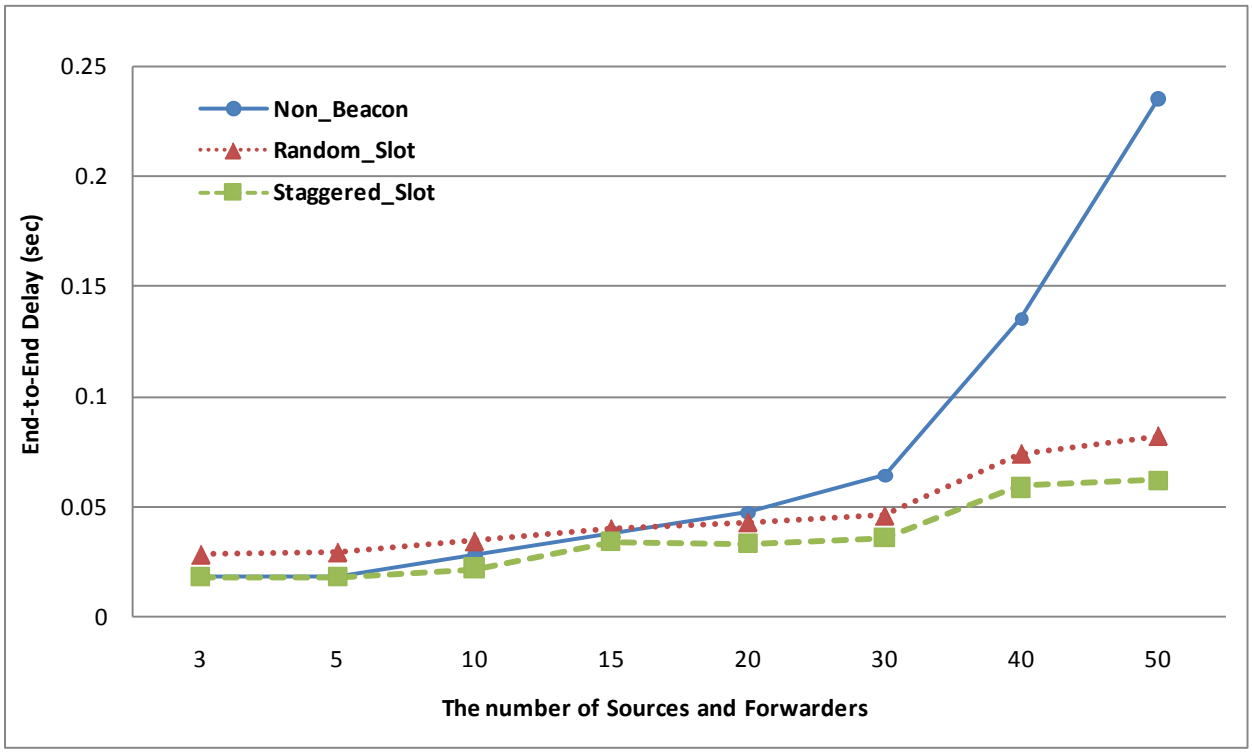
Figure 2.4 The number of supported paths with different MOs in DSMEs

Fig. 2.4 shows that if a single channel is used when MO = 1, which represents the case of legacy IEEE 802.15.4, only four e2e paths are supported. Fig. 2.4 also shows that more paths are supported with multi-channel DSME-GTS than with single channel GTS. The advantage for using multi-channel, however, terminates at a four-channel scheme, which is a saturation point when MO =1. The maximum number of source nodes is 14 because the benefit stays constant when more than four channels are used. This limitation basically stems from only allowing seven GTS slots. According to the results, the overall performance of the current scenario depends solely on the number of available timeslots, instead of the number of available channels.

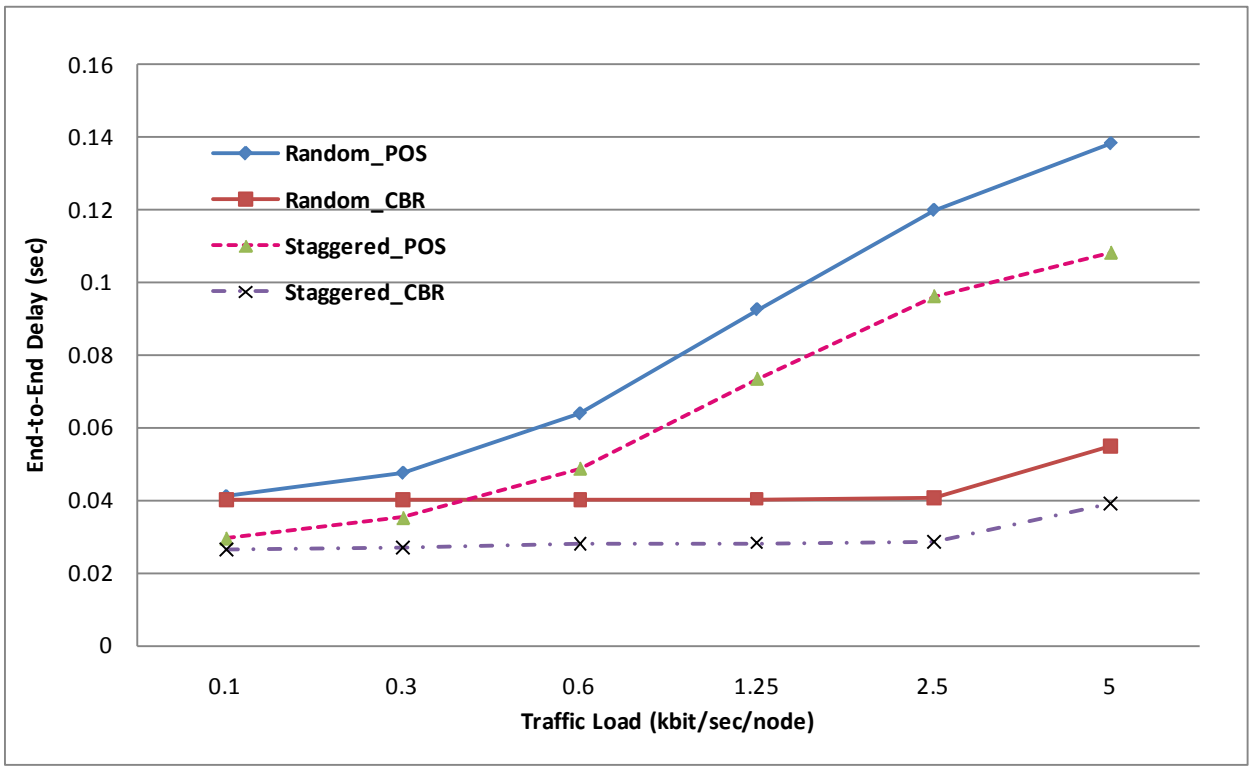
In Fig. 2.4, the number of supported paths reaches up to around 30 among 50 source nodes when MO = 2. Even with 16 channels, the network cannot support more than 35 paths when MO = 2. When MO = 3, all 50 paths are supported with more than two channels.

2.5.2 End-to-End Delay

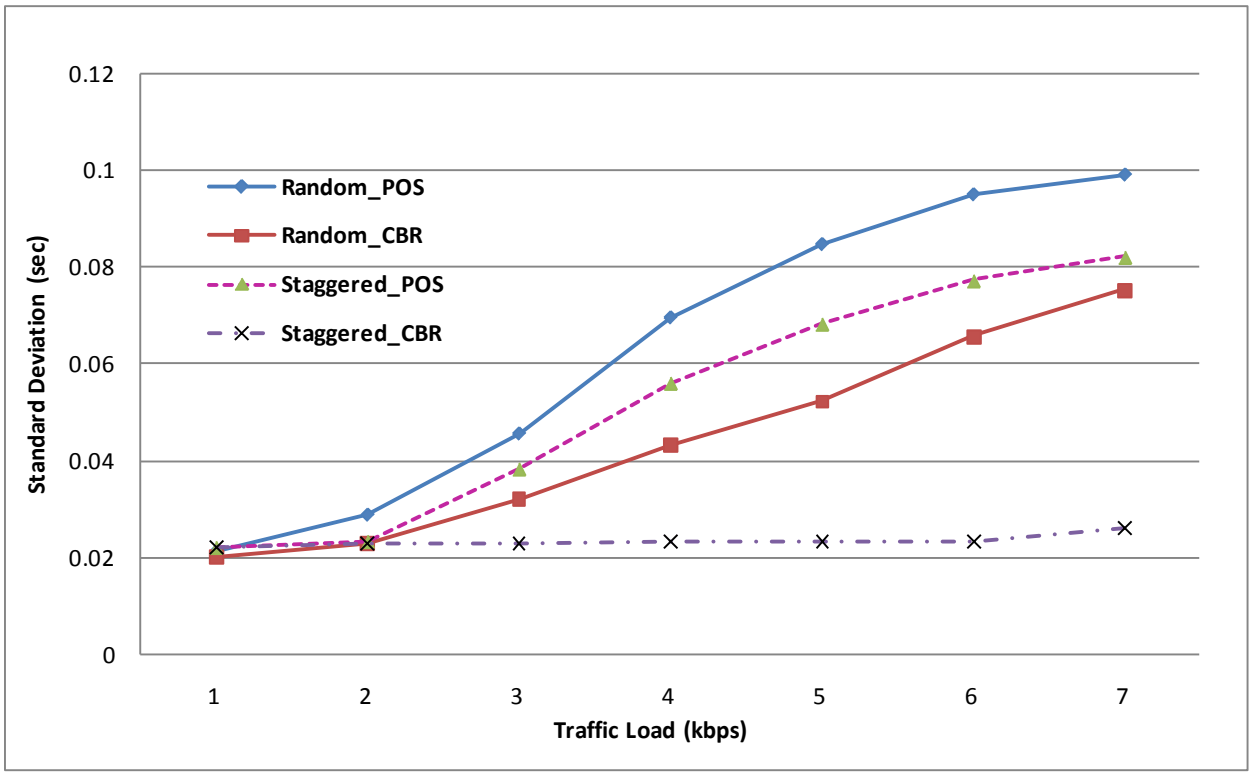
Given the results in Fig. 2.4, we run simulations with four channels and we set the value of MO to 1 if the number of source nodes is less than 14. If the number of source nodes is between 14 and 32, we set the value of MO to 2. If the number of source nodes is more than 32, we set the value of MO to 3.



2.5 a).e2elay vs. the number of sources for



2.5 b).e2edelay vs. traffic



2.5 c). Standard Deviation for CBR and Poisson

Figure 2.5e2e delay in terms of the number of nodes and their traffics, and Standard Deviation for both CBR and Poisson

We implement the random slot allocation and the staggered slot allocation. In the random slot allocation, to ensure avoidance of collision and interference, a slot allocation is unique for each node within one-hop distance and spatially-reused beyond a two-hop distance. The node chooses from the ABT the first available timeslot with at least one vacant channel to avoid a null selection, and then chooses a channel that is available within that same timeslot. In the staggered slot allocation, a node that has time-critical data flags for the staggered slot when the node requests DSME-GTS request. After receiving the DSME-GTS request, a node selects the very first available timeslot before choosing the channel, and then it specifies its allocated timeslot in the DSME-GTS request, which is relayed to the next node. The next node then chooses the next first available timeslot and an available channel before relaying its own DSME-GTS request to its parent node. This pattern is repeated until the final destination node is reached.

In Fig. 2.5a, we run simulations with 6 difference topologies with 6 different random seed due to CBR traffic, and thus we run 36 simulation runs in total, and traffic rate for source node is 0.1 Kbit/sec. Fig. 2.5a shows that the delay for IEEE 802.15.4 non-beacon mode with less than 30 sources is comparable to the staggered slot allocation but the delay increases exponentially with more than 30 source nodes as anticipated. The staggered slot allocation performs better than the random slot allocation and non-beacon mode. In the staggered slot allocation, when a node receives data by the very first slot of the available DSME-GTSs, it forwards the data at the next available DSME-GTS slot within the same multi-superframe. On the other hand, in the random slot allocation, there are many occasions where the next available DSME-GTS slot does not exist

within the same multi-superframe. Hence, the e2e delay for the random slot allocation is greater than the staggered slot allocation. However, the e2e delay for the random slot allocation is still significantly less than the non-beacon mode, which is a contention-based MAC.

Fig. 2.5b shows the average e2e delay with 6 different topologies and 14 randomly chosen sources. We run simulations with varying the traffic load and the x-axis of Fig. 2.5b indicates the traffic load per each source node. In the staggered slot allocation for CBR data, its e2e delay remains constant because the capacity between sources and the PAN coordinator established by DSME-GTS TWH exceeds the total source rate and the aggregate data rate for a superframe duration is constant. These phenomena also occur in the random slot allocation for CBR traffic models. However, the delay for Poisson data in the staggered slot gradually increases along with an increase in the traffic load. As traffic load increases, Poisson data generation eventually exceeds the assigned timeslot's capacity, and the data accumulates in the buffer, leading to a gradual increase in delay. These phenomena also occur in the random slot allocation for Poisson traffic models. The delay increases exponentially as traffic workload increases. As shown in Fig. 2.5b, the e2e delay of the staggered slot allocation is at least 70% less than that of the random slot allocation for both CBR and Poisson data.

In Fig. 2.5c, the standards deviation (jitter) for CBR traffic in the staggered slot allocation remains constant, which implies that the e2e delay remains constant as well, just as what is shown in Fig. 2.5b. However, the jitter for CBR traffic in the random slot allocation increases as the traffic load increases because the probability of CBR data to be backlogged in buffer prior to transmission increases due to the designated slot being assigned at the random slot allocation. If a data frame is backlogged, it should wait until the next multi-superframe to be transmitted. The jitter for CBR traffic in the random slot scheme increases to approximately three times that for

CBR in the staggered allocation scheme. More importantly, the jitter of Poisson traffic becomes prominent when compared to CBR in the staggered slot allocation due to the random statistical nature of Poisson traffic regardless of whether the slot allocation is staggered or random. When traffic load becomes heavy, the random slot allocation for both CBR and Poisson as well as the staggered slot allocation for Poisson experience a significant increase in jitter while the staggered slot allocation for CBR shows a slight increase. The simulation results show that the staggered slot scheme for CBR traffic is well fitted for a deterministic data application.

2.6 Performance Comparison of Channel Hopping and Channel Adaption

2.6.1 Two-State Markov Model for Channel Behavior

A Markov approximation for the block error statistics on fading channel has been well documented in the literature [28]. According to this model, the sequence of data block success and failure can be approximated as two state Markov chain as depicted below.

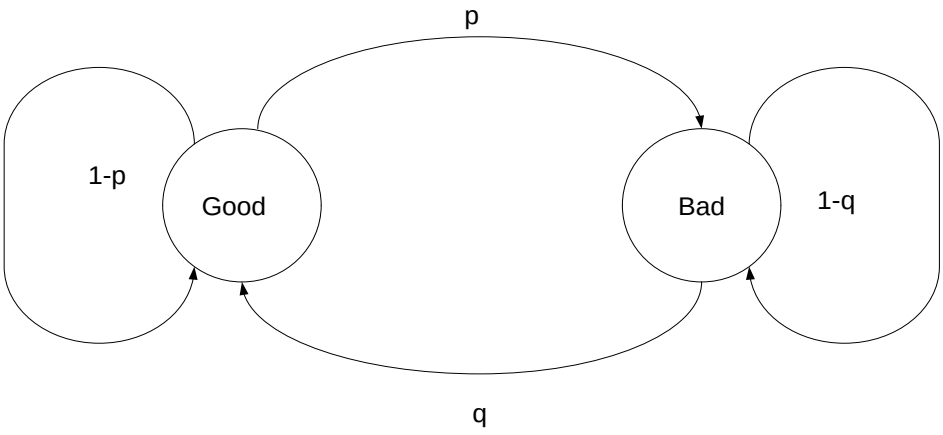


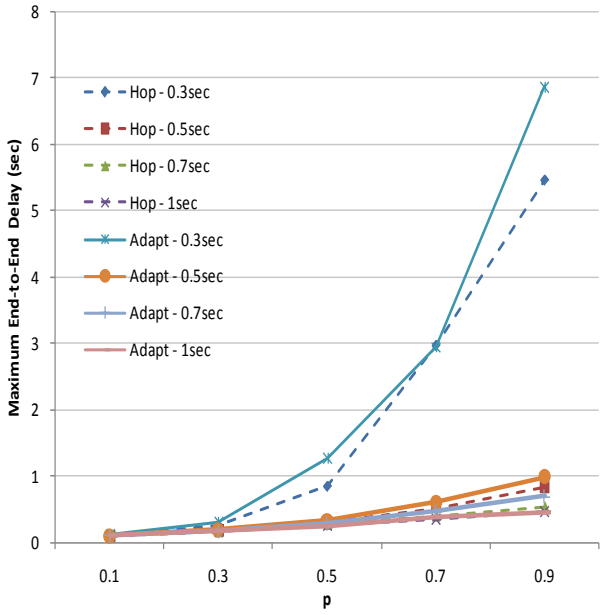
Figure 2.6 Two state Markov chain channel dynamics model

The transition probability matrix is: $\begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}$ where p represents the transition probability from "good" to "bad" state, and q represents vice versa. For example, higher p with the lower q describes worse channel condition while lower p with the higher q describes better channel condition. To reflect diverse channel conditions such as frequency selective fading among different channels, we modeled the channel i with pi and qi drawn from Normal Distributions N(p, σ) and N(q, σ). For example, if $\sigma^2 = 0$, all $p_i = p$ and all $q_i = q$, where all channels become statistically identical. The variance v controls the diversity in channel conditions, i.e., higher v makes channel states more diversified.

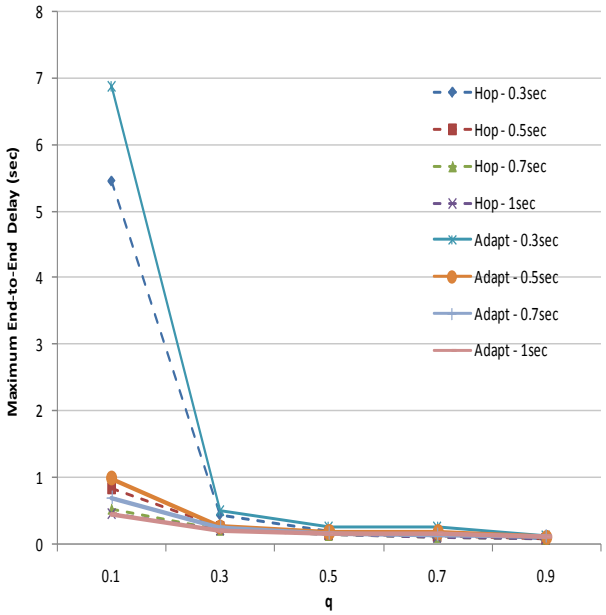
2.6.2 Simulation Results

We create three scenarios by varying the number of source nodes, which generate CBR data. In scenario 1, 6 nodes are data sources among 50 nodes. In scenario 2 and 3, there are 10 and 16 data sources, respectively, which can represent light, medium and heavy traffic situations. If BO = 4, MO = 2, SO = 1 with CAP reduction mode, there would be two superframes and (7 + 15) = 22 DSME-GTS slots. The input parameters are the number of source nodes, packet inter-arrival rate, two state Markov model parameters, the variance of the transition probability, and the mode of channel diversity.

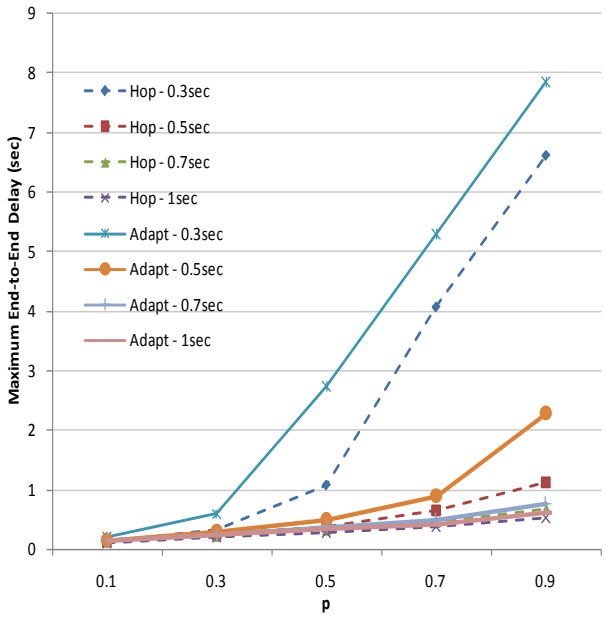
We first look at the situation when all channels are experiencing identical condition, i.e., $\sigma^2 = 0$. In the following graphs we plot maximum e2e delay. We present graphs for p with the maximum delay from all q's because we are interested in the worst case, i.e., simulation result usually shows the lowest q.



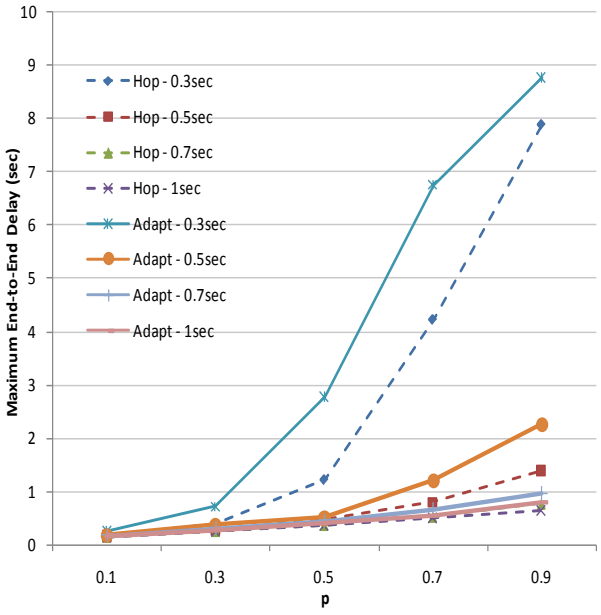
a). Max e2e delay vs. p for 6 nodes



b). Max e2e delay vs. q for 6 nodes



c). Max e2e delay vs. p for 10 nodes

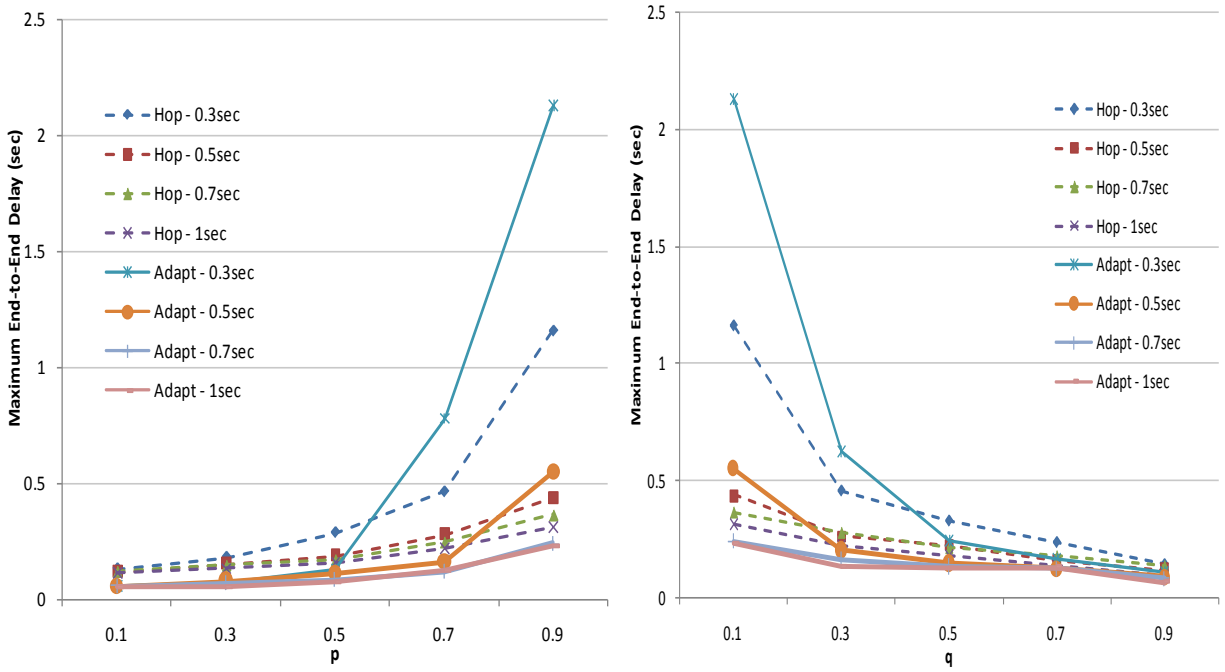


d). Max e2e delay vs. p for 16 nodes

Figure 2.7 Maximum e2e delay with fixed p, and q varies from 0.1 to 0.9

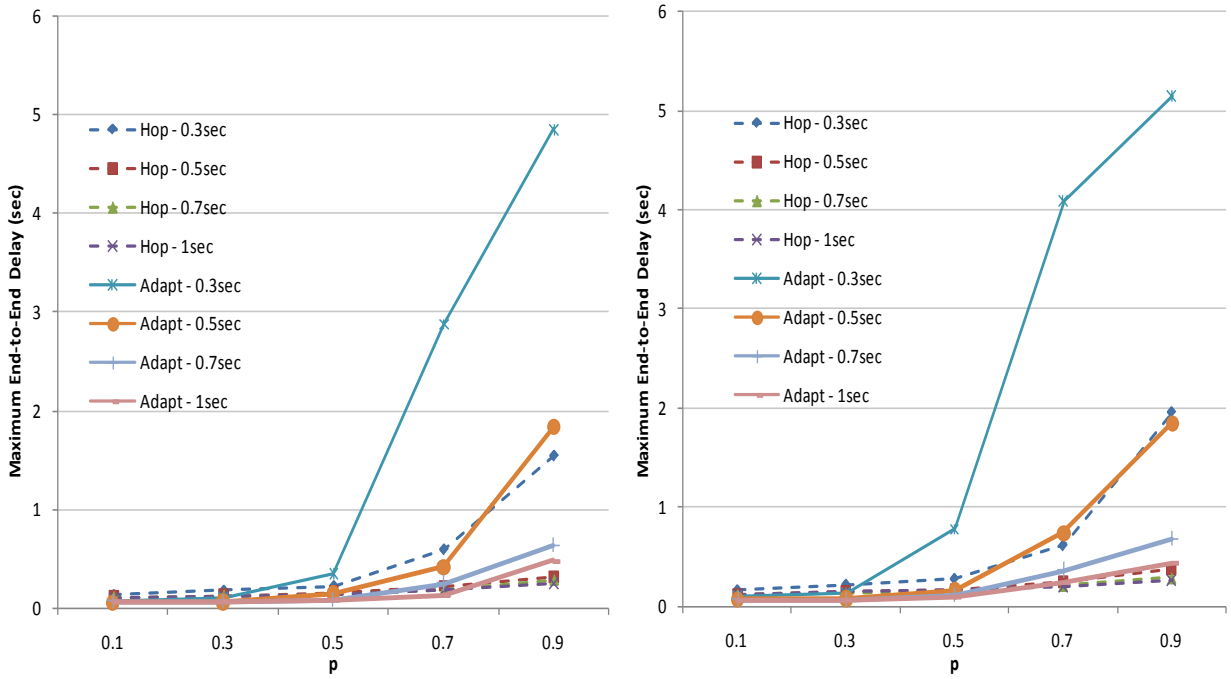
In Fig 2.7, with the variance $\sigma^2 = 0$, we can see that channel hopping technique provides better performance in almost all cases. As every channel is experiencing the same channel quality, switching to a different channel does not produce any benefit. As channel hopping allows a node to use the entire channels periodically, the existence of some good channels at some time duration can be utilized more frequently than channel adaptation which can not readily use the channel due to the observation window and TWH delay.

Now, we look at cases in which we diversify the channels by introducing $\sigma^2 = 0.2$ while keeping the value of the transition probability the same as in the previous cases. We choose variance 0.2 among 0.1, 0.3, and 0.4 because it shows the increased diversity in channel.



a). Max e2e delay vs. p for 6 nodes

b). Max e2e delay vs. q for 6 nodes



c). Max e2e delay vs. p for 10 nodes

d). Max e2e delay vs. p for 16 nodes

Figure 2.8 Maximum e2e delay with fixed p, and q varies from 0.1 to 0.9

In Fig. 2.8, when diversities of channel conditions are present, channel adaptation exploits the use of good channels. However, we observe cross-over points when channel diversity and traffic load increase. Fewer good channels are available to be allocated for channel adaptation in such conditions. Moreover, high network traffic triggers frequent TWH, which causes delay as this commend frame contends during CAP to be successfully transmitted. Although channel adaptation parameters can be tuned by changing the window size, triggering threshold, and α, it would be very difficult to determine the exact combination for a specific scenario. According to the simulation results, the channel adaptation would achieve better performance by checking the quality of the available channels. On the other hands, the channel hopping could be utilized where the fading condition of every channel is identical or when most channels are in bad condition with high traffic.

2.7 Conclusion

In this chapter, we study IEEE 802.15.4e DSME mode that enhances the GTS mechanism of IEEE 802.15.4 beacon-enabled mode to satisfy deterministic latency, high reliability, and flexibility for industrial applications. DSME provides guaranteed latency for both the staggered slot allocation and the random slot allocation in a multi-hop wireless sensor network. Unlike IEEE 802.15.4, DSME shows low end-to-end latency and jitter for CBR and Poisson traffic models, making it a suitable MAC protocol for applications with stringent latency requirements. Moreover, DSME supports robustness over frequency selective fading conditions. The channel adaptation adapts to the high quality channel as long as there is a reasonable number of available channels. The channel hopping becomes more prominent than the channel adaptation during all channels experiencing identical condition or severe discrepancy in channel behavior with high traffic condition.

Chapter 3

METAHEURISTICS

3.1 Introduction

Wireless Sensor Networking (WSN) is a continuously evolving technology for various applications, such as environment monitoring, patient monitoring, and many industrial applications. Wireless sensors can potentially be deployed in a large geographical area via multi-hop communications.

Unlike delay-tolerant applications, patient monitoring, disaster warning, intruder detection, and many industrial applications require timely responses. However, it is challenging to provide timely and reliable communication in WSNs, mainly due to the fact that conventional WSNs operate on a single channel. Sensor nodes must compete with other nodes to access a single channel medium of limited bandwidth. If a transceiver operates on multiple channels, multiple simultaneous transmissions and receptions are feasible on wireless media without interfering with the others, and the bandwidth limitation can be relieved. Therefore, using multiple channels and timeslots facilitates timely communication.

In the IEEE Standard 802.15.4 [4] for WSNs, a superframe structure consists of Contention Access Period (CAP) and Guaranteed Time Slot (GTS). Our proposal utilizes this superframe structure, but each timeslot is extended to accommodate multiple channels as in the IEEE Standard 802.15.4e [5] to guarantee end-to-end delay. The channels and timeslots available to a node vary because each node’s selection of channels and timeslots imposes a set of constraints

on the channels and timeslots available to its neighbors. Our proposal affords each node the freedom to choose the optimal timeslot and channel in establishing communication links to its neighbors, resulting in high throughput and low delay.

3.1.1 Scheduling

Scheduling is a critical process for virtually all resource-allocation problems, especially to meet quality of service (QoS) requirements. Scheduling channels and timeslots for all nodes constituting an end-to-end (e2e) path to meet certain delay bounds is challenging because each node has a different remaining path-length to the destination and encounters dissimilar channel environments. Assuming the channels and timeslots are integer-numbered from 1 to some arbitrary number, a simple approach would be to schedule them in a sequenced and staggered fashion from the source to the destination, i.e., each node chooses the smallest number out of available timeslots and channels, and this channel-timeslot combination becomes unavailable to its children, parent, and their neighbors. This approach can reduce the e2e delay by assigning the first node in a path with the first available timeslot and the next node with the next earliest available timeslot, assuming all nodes are synchronized. However, this approach does not scale when there are many simultaneous and overlapping e2e paths coexisting at crossing nodes. An example of this approach is found in the Distributed Synchronous Multi-Channel Extension (DSME) mode of IEEE Standard 802.15.4e.

Optimal scheduling of multiple channels and timeslots for a large-scale WSN is an NP-complete problem [6] due to the scale of the search space $O((T \cdot C)^{\sum_i h_i})$, where h_i is the number of hops in *i-th* path, and *T* and *C* the number of timeslots and the number of channels for each node, respectively.

3.1.2 Metaheuristic

The metaheuristic optimizations have recently seen much interest in combinatorial optimization problems, especially when the goal is to quickly find a near-optimal solution. The essence of the metaheuristic approaches lies in balancing the exploitation of a promising region for a good solution and the exploration of entire search space [6]. We use the metaheuristic optimization for our scheduling problem because any suboptimal solution that results in an e2e delay lower than the required delay bound is acceptable. Furthermore, fast computation time is desired because scheduling is done not only when the network is first set up, but also whenever a network encounters situations such as a link failure or a topology change. Whenever such situations arise, the optimization algorithm must be rerun in the middle of the operation period.

3.1.3 SA and PSO

As powerful tools for combinatorial optimization problem, the metaheuristic computational methods recently found many applications for which the exhaustive search had not been feasible. The metaheuristic computational methods seek to substantially lessen the computational burden often at the expense of the solution quality.

The Simulated Annealing (SA) was originally introduced in 1983 by Kirkpatrick et al., who were inspired by the physical annealing process in metallurgy [29]. A material heated to a higher temperature than normal is cooled down gradually to increase the size of its crystals, thereby reducing its defects. During high temperature, its atoms move extremely fast in random directions while generating high kinetic energy. When the material is cooled gradually, the atoms have strong tendency to re-crystallize by themselves to achieve minimum energy state (e.g.,

optimal state).

The Particle Swarm Optimization (PSO) with real-valued problems was initially proposed by Kennedy and Eberhart in 1995 [30], who subsequently designed a binary PSO [31]. In PSO, each particle moves through a multidimensional search space towards the desired target by adjusting its velocity according to its own experience (cognitive) and others' experiences (social) [30], [31]. The difference between the real-valued PSO and the binary PSO is that the former represents each particle's position and velocity as a real number while the latter does so as a binary number.

A hybrid HPSO of PSO and SA in [32] reduces the total number of slots and energy consumption. However, minimizing slot usage does not result in minimum e2e delay because each path is neither weighed nor treated equally in terms of delay. Therefore, it may not guarantee a bounded delay for each path. Furthermore, computation time increases when binary PSO and SA are combined [32]. Wang et al. in [33] runs the binary PSO in each cluster separately, and the results from a cluster are completely independent from those of other clusters. In a real network, the members of a cluster can interfere with the members of another cluster if they are within each other's communication range. Currently, [33] disregards this issue by assuming that there is no interference among clusters at all. It runs a PSO algorithm in one cluster, and the results from that cluster are completely independent from those of other clusters, which may suffer in solution quality compared to a centralized approach. Wang et al. [32] and [33] are based on TDMA and channel scheduling, respectively, while our scheduling algorithm accounts for both channel and timeslot. A recent work [8] considers both timeslot and multi-channels and minimizes the number of timeslots required to deliver the data to the coordinator, but did not consider the e2e delay.

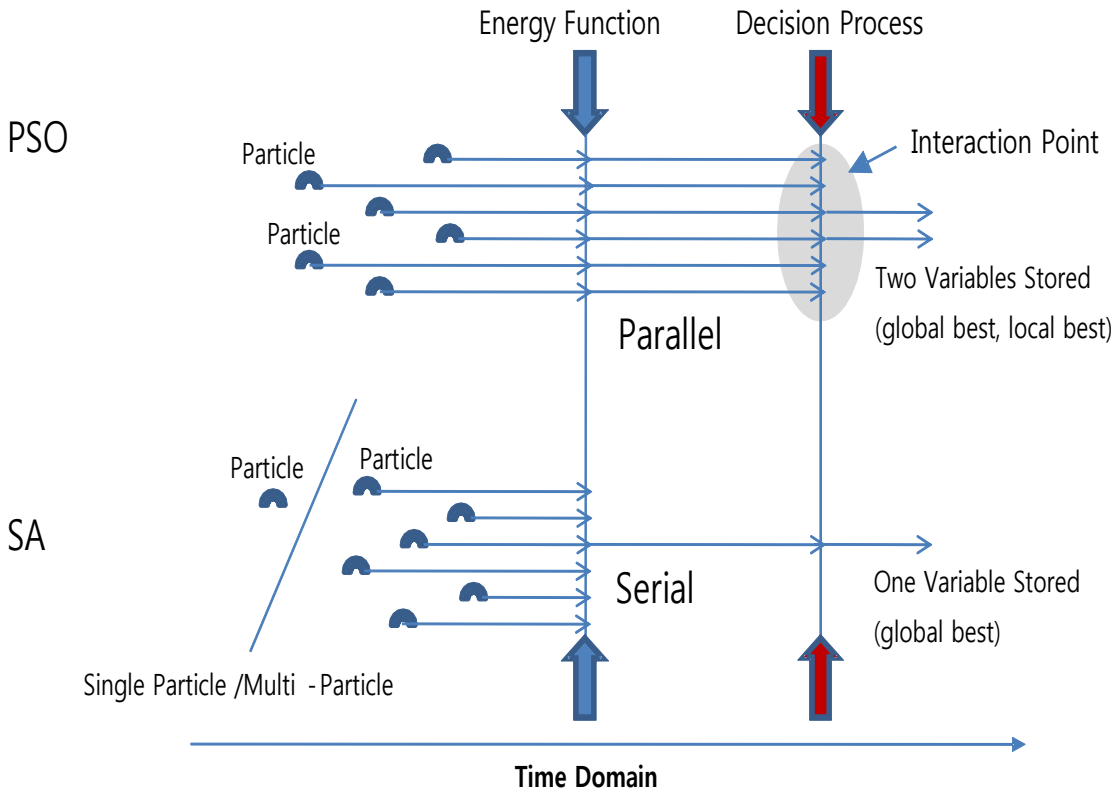


Figure 3.1 A conceptual comparison of PSO and SA, where a particle is a solution

Fig. 3.1 shows a conceptual difference between PSO and SA. In general, SA is implemented using either one particle or multiple particles. As shown, SA has a serial attribute that passes only one of the multiple particles through to the energy function regardless of how many particles there are. In our scheduling problem, the number of particles maps to the number of paths in the networks; therefore, SA is implemented using multiple particles before adjusting temperature to fairly compare its performance with PSO.

3.1.4 Adoption of SA and PSO

The selection of SA is for the simplicity afforded by a discrete domain implementation [36]. If a row and a column in a matrix represent a channel and a timeslot, respectively, a potential solution is obtained by searching for row-column pairs to meet the desired e2e delay bound. On the other hand, choosing PSO for our problem is for its directionality feature. The concept of directionality can be explained by the birdfeeder example. Given a birdfeeder, its location becomes an indicator of the direction towards which the birds would fly. The primary reason for selecting PSO metaheuristic is due to its directionality characteristic. Although the exact flight path of a specific bird cannot be predicted, i.e., it takes a random path, the general direction of its path can be predicted. The e2e delay involves the sense of directionality in that in order to minimize delay, timeslots are allocated in sequence, and multiple timeslots required by a single path are assigned as close to each other as possible. Although there is no such directionality in SA because the particle movement is completely random, it is still feasible to employ directionality by truncating the movement space. For example, if a particle is restricted from moving in certain direction based on the location of the destination, we can predict that it will not move in that restricted direction. Another reason for selecting the PSO and SA metaheuristics is their relatively low complexity to implement in the discrete domain, i.e., PSO and SA provide a simple solution to a complex problem rather than a complex solution to a complex problem. In view of the fast computation time, the real-valued PSO is better-suited for our large-scale domain than binary PSO because the latter requires greater computation time than the former. In binary PSO, each pair of row and column stores each pair's velocity vector. The velocity must be transformed into the change of probability because the movement has to be expressed by probabilities. However, in real-valued PSO, only the pairs of rows and columns that actually

contain particles store each pair's velocity vector.

3.2 Literature Review

A flowshop scheduling problem solved via PSO is proposed by [34] in which the number of jobs is equivalent to the number of particles. Particles fly in a discrete domain in rows (job) and columns (position) to search for an optimal position in terms of the provided performance measure. A timetable scheduling in a discrete domain is studied by PSO [35] to solve the scheduling problem in a continuous domain.

In the channel assignment problem (CAP [36]), SA is applied to assign the channel to the cellular radio Base Station (BS) in terms of the traffic rate because BS's in different locations have diverse traffic attributes, and channel assignment depends on the amount of traffic a BS must handle. In order to reduce the search space, CAP introduces a flip-flop for a neighbor generation in neighborhood structure, in which a used channel is excluded from the list of the channels that are available for the next neighbor generation. The benefit of the flip-flop is quite insignificant because the number of used channel is small compared to the whole search space when there are a large number of search spaces. In [37], channel selection by SA is performed at an Access Point (AP), which interferes with other APs within a certain communication range in multi-AP 802.11b/g networks. The cooling schedule, which is a major contributing factor to SA performance (e.g. solution quality and computation time), was studied broadly, but their system was not complex enough to warrant analysis of numerous cooling schedule behaviors because the attributes of different cooling schedules could be diverse at higher degrees of system complexity. TDMA slot allocation in Satellite system is fulfilled by two-run SA with single cost function [38]. They intentionally divided the SA into two in terms of number of operators. The

first SA assigns timeslots to the operators that require half of the maximum bandwidth, and the rest of the operators rely on the second-run SA in order to take advantage of high initial cooling schedule in each run. High initial temperature helps to reduce the local minimum at the expense of extended computation time. In summary, [36] and [37] focus on channel allocation with single design variable, and [38] centers on timeslot allocation with single design variable. In addition, none of the three models consider multi-hop environments. Our model on the other hand focuses on multi-hop environments, as well as both channel and timeslot allocations. In [39], the network localization problem is formulated by running SA twice with different cost functions to raise the overall accuracy of the location estimate. In the first phase, it tolerates massive search for each iteration period in order to increase the solution quality (accuracy), but this massive search requires high computation complexity. During the second phase, only the nodes that encounter flip ambiguity, in which nodes are placed in a collinear manner, are allowed to perform SA to further increase the accuracy of the location estimate. Sensor placement problem is solved by SA with the cost limitation constraint of relatively small total number of sensors in the network [40]. Optimal sensor placement for a target location is determined based on whether a sensor deployment is permitted at a grid point of the sensor field subject to the cost limitation in order to detect the potential target which can be positioned in any one of the grid points. The distance error of two indistinguishable grid points is formulated for the objective function to minimize the maximum distance error value. In order to ensure accurate positioning, [40] also allows extra annealing processing for each iteration period as in [39]. In conclusion, there has not been any other researches on the instant topic that placed equal importance on both timeslot and channel assignments in a wireless multi-hop sensor network in order to optimize end-to-end delay, treating all nodes with equal importance.

3.3 Network Model and Problem Formulation

We consider a tree-based multi-hop WSN with each node supporting multiple channels and timeslots. Each node transmits data periodically to the coordinator on an assigned channel and timeslot within a superframe, and all channels and timeslots are of equal bandwidth and size, respectively. We assume that the timeslots are synchronized, and that all control packets are transmitted over a common channel. Each node in the tree topology generates a single packet, and each intermediate node would have two roles: a sender and a forwarder. The packets from all the nodes are destined to the coordinator, which is the typical scenario for data collection in WSNs. The coordinator is assumed to have all the required information to schedule all nodes, such as the number of available channels and timeslots, the number of nodes on a route, connectivity, and the relationship between the parent and children nodes. In order to optimize end-to-end delay in a tree topology, there are three distinct issues to consider. First, the required number of timeslots at each node depends on its tree depth and the degree of connectivity. Scheduling in each required timeslot and channel is relayed from the source node to the coordinator in sequence. Second, since each node in the end-to-end path has the different number of timeslots required, the order of schedule becomes an important factor. For example, a path that requires the most number of timeslots will be allocated the first available timeslots in a superframe, and the path requiring the least number of timeslots will be assigned its timeslots after all other paths have been assigned earlier timeslots, which results in an unequal treatment in terms of e2e delay. Third, each node has different number of available timeslots and channels because each node has different constraints imposed on timeslots and channels due to the connectivity and interference. Taking advantage of the global information collected at the coordinator, the proposed algorithm is implemented as a centralized architecture.

Notations:

$$\begin{cases} R_{n,\tilde{n}}: \text{a set of all potential routes } (r_{n,\tilde{n}}) \\ d_{r_{n,\tilde{n}}}: \text{the delay of a route between source } n \text{ and destination } \tilde{n} \\ l, \tilde{l}: \text{the link between node } l \text{ and } \tilde{l} \text{ in a route } r_{n,\tilde{n}} \\ d_{l,\tilde{l}}: \text{the link delay for } l, \tilde{l} \\ \text{The destination of all paths is } \tilde{n}, \text{the coordinator in this paper} \end{cases}$$

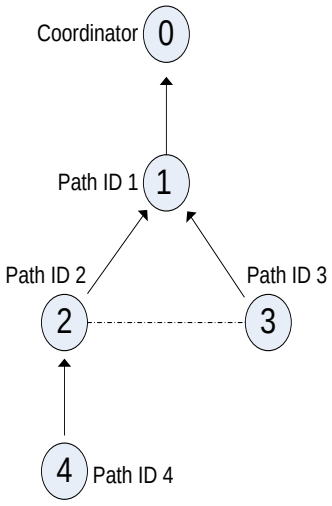
To formulate the optimization problem, we define an energy function, $\varepsilon = d_{r_{n,\tilde{n}}}$ which is an internal energy state prior to the execution of the decision process in each iteration.

The objective of the optimized scheduling is to minimize the maximum e2e delay, which is formulated as:

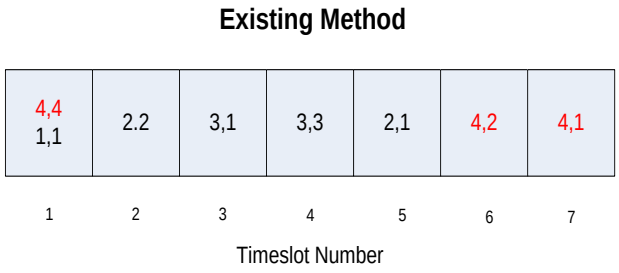
$$D_{\text{e2e delay}} = \min \left(\max_{r_{n,\tilde{n}} \in R_{n,\tilde{n}}} \left(\sum_{l,\tilde{l} \in r_{n,\tilde{n}}} d_{l,\tilde{l}} \right) \right)$$

Constraints *(c.1 – c.3):*

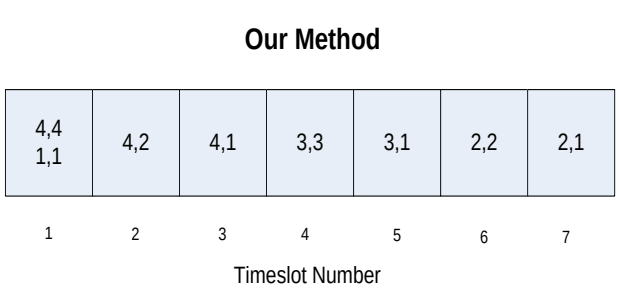
(c.1): the transmission timeslot of node n must be different from those of its one-hop neighbors except when two connected nodes have different parent nodes, a half-duplex transceiver. (c.2): the transmission channel of node n should be different from those of the two-hop neighbors except when two-interfering nodes have different parents, interference avoidance. (c.3): each node would enable multiple channel-timeslot combinations. In addition, node n has at least one channel-timeslot combination except for the coordinator.



Path	Node			Hop
1	1			1
2	2	1		2
3	3	1		2
4	4	2	1	3



Path	e2e delay
1	1
2	4
3	7
4	7



Path	e2e delay
1	1
2	2
3	2
4	3

a) A tree topology and its path vectors

b) The scheduling comparison for existing and our methods

Figure 3.2 Illustration of a topology and its e2e delay comparison

We construct a tree in which each node has its own node vector (path ID, node ID) for the transmission of a packet to the coordinator. For example, path ID 4 consists of node 4, node 2, and node 1. Note that the node ID of each node is the same as the path ID of the path starting from that node and ending at the coordinator. We denote the path length in terms of “hop”. Then, the summation of all path lengths equals total number of hops for a given topology. The total number of hops for the network in Fig. 3.2 is 8. The search space for all one-hop transmissions comprises a matrix with its row representing the channel number and its column for the timeslot

number. The sizes of row and column are *C* and *T*, respectively. Node vector (path ID, node ID), i.e., a particle, in the matrix pertains to a channel and timeslot allocation. The number of particles corresponds to the total number of hops because each hop requires one channel-timeslot. Since the energy function is based on the e2e delay, i.e., a time entity, we take advantage of the explicit "direction" resulting from timeslot arrangement. By this arrangement, the timeslots of all the hops on a particular path are eventually squeezed as close to each other as possible within the constraints (c.1 to c.3) imposed by the other hops in other paths. From the above illustration, the e2e delay in [7], [8] may end up being a whole frame length, i.e., the e2e delay is 7 for a path 4 at the top of Fig. 3.2b. But in our work (the e2e delay is 3), this does not occur even in the worst-case scenario because the timeslots in a path are squeezed within a frame as close as possible. Note that parallel transmission happens at the first timeslot for node 1 and 4. There is no connection between node 2 and node 3 (dotted line in Fig. 3.2), and they must be assigned different timeslots due to the same parent.

3.4 Algorithms of SA and PSO

3.4.1 Simulated Annealing

There is a solution x and an energy function $\mathcal{E}$, which calculates $\mathcal{E}(x)$ for each solution x. X denotes the total number of solutions. The determination of the neighbors $\tilde{x}$ of x is affected by neighborhood structure [29], [30], and we employ the directionality by truncating the movement space in our scheduling. If a solution is restricted from moving in certain direction, we can predict that it would not move in that restricted direction. At each iteration, $\mathcal{E}(\tilde{x})$ of $\tilde{x}$ is calculated and x is replaced by $\tilde{x}$ if and only if $\mathcal{E}(\tilde{x}) \leq \mathcal{E}(x)$, i.e., there is a "downhill move",

which may result in a local minimum. To avoid that, SA probabilistically allows an "uphill move" that produces a higher value according to the so-called Metropolis criterion [29], [30]. If the $\mathcal{E}(\tilde{x})$ is greater than $\mathcal{E}(x)$, the decision to accept $\tilde{x}$ is determined by min { 1, exp (-($\mathcal{E}(\tilde{x})$- $\mathcal{E}(x)$) / T) } > rand [0,1), which is called acceptance probability. The temperature T is controlled by a certain cooling schedule during the annealing process [29], [30]. In an annealing process, during high temperature, its atoms move extremely fast while generating high kinetic energy, i.e., the probability of accepting a frequent uphill move increases, and the SA exhibits stochastic behavior. When the material is gradually cooled, the atoms have strong tendency to re-crystallize to achieve minimum energy state, i.e., the acceptance probability decreases, and infrequent uphill move is accepted. Thus, the SA tends to a deterministic process. Generally speaking, when the temperature is dropped rapidly, the convergence time will be shortened, but the solution quality will suffer. On the other hand, a slow cooling schedule that extends the convergence time will produce a better solution quality. Determining the right cooling schedule is critical since the temperature adjustment impacts SA significantly. In order to determine the right cooling schedule, we investigate four frequently-used cooling schedules: exponential, Boltzmann, fast, and linear. Exponential, Boltzmann, fast, and linear schedules update the temperature vector by exponential decrease, logarithmic decrease, division, and subtraction of the current annealing parameter starting from the initial temperature, respectively. Once the cooling schedule has been determined, figuring out how much to lower the temperature by for iteration during the annealing process is also a challenge, i.e., a large size of temperate drop step falls down a local minimum while a small size of temperature drop causes long computation time.

3.4.2 Particle Swarm Optimization

The notation of real-valued PSO is as follows. N_a denotes the total number of particles. *Let* X_a^i = $(x_{a1}^i, x_{a2}^i, \ldots\ldots, x_{aD}^i)$, where $x_{ad}^i \in \Re^2$, be the particle a in D (two dimensions) at iteration *i*. Each particle is represented by a position in the search space as X_a^i, which is a potential solution. Denote the velocity as $V_a^i = (v_{a1}^i, v_{a2}^i, \ldots\ldots, v_{aD}^i)$, where $v_{ad}^i \in \Re^2$. Let $P_a^i = (p_{a1}^i, p_{a2}^i, \ldots\ldots, p_{aD}^i)$ be the personal best that particle a has obtained until iteration *i*, and $P_g^i = (p_{g1}^i, p_{g2}^i, \ldots\ldots, p_{gD}^i)$ be the global best obtained from p_a^i at iteration *i*. The movements of the particles in the real-valued PSO are governed by the following [31].

$$v_{ad}^i = w^i * v_{ad}^{i-1} + c_1^i * \mathrm{r}_1 * \left(p_{ad}^{i-1} - x_{ad}^{i-1}\right) + c_2^i * \mathrm{r}_2 * \left(p_{gd}^{i-1} - x_{ad}^{i-1}\right) \quad (3.1)$$

$$x_{ad}^i = x_{ad}^{i-1} + v_{ad}^i \quad (3.2)$$

There are three important parameters in Eq. 2 directly affecting the particle behaviors: w^i , c_1^i and c_2^i .The w^i represents inertia weight, which provides the global search ability (exploration) at the beginning and then the local search ability (exploitation) at the end of the process. Thus, w^i varies as *i* progresses because the particle initially moves fast, and then slows down as it approaches the target to avoid overflying [32]. The "cognitive" coefficient c_1^i and the "social" coefficient c_2^i define how fast each particle moves towards p_{ad}^{i+1} and p_{gd}^{i+1} positions. Therefore, as with w^i, varying c_1^i and c_2^i not only promotes exploration of a remote target, but also encourages exploitation at a nearby target. If p_{gd}^i for a particle is selected more often than others, the likelihood of the particle being on the right track towards the global best solution increases, so c_1^iincreases, and the other particles would be more likely to follow that direction. In contrast, if p_{gd}^i for a particle is selected less often, c_2^i increases because the solution quality of that particle is poor compared to those of other particles, and that particle follows another

direction. Consequently, each particle updates its velocity and position depending on the frequency of p_{gd}^{i}. r_1 and r_2 are random numbers uniformly distributed within [0,1). A maximum velocity $\pm V_{max}$ (½ of *D*) is necessary, not only to prevent a particle from escaping the search space, but also to provide the particle a high rate of self-mutation.

SA	PSO
Initialize T_0 for a cooling schedule Scheduling toward (1st channel, 1st timeslot) Initialize $\mathcal{E}_g$ 1) Solutions *X* randomly assigned into a row-column matrix 2) Update current solutions' locations towards the direction 3) if $(\mathcal{E}(\tilde{x}) - \mathcal{E}(x) \leq 0)$, $\mathcal{E}_g = \tilde{x}$ Else apply the *acceptance probability* in SA of Ⅲ 4) Move solutions from the location of $\tilde{x}$ within the constraints *c.1-c.3* 5) Evaluate a path with e2e delay bound 6) Decrease T 7) Loop to step 2 until all paths satisfy e2e delay bound	Initialize $w^i, c_1^i, c_2^i, \pm V_{max}^i, p_a, p_g$ Scheduling toward (1st channel, 1st timeslot) 1) Particles randomly assigned into a row-column matrix 2) Update current particles' locations towards the direction 3) If $\mathcal{E}$ is better than p_a, set $p_a = x_a^i$ and if $\mathcal{E}$ is better than p_g, set $p_g = x_a^i$ 4) Move particles by Eq. 2 and Eq. 3 within the constraints c.1-c.3 5) Evaluate a path with e2e delay bound 6) Update c_1^i and c_2^i accordingly and update w^i 7) Loop to step 2 until all paths satisfy e2e delay bound

Figure 3.3 The pseudo codes of SA and PSO

3.5 Performance Evaluation of FCFS and SA

First Come First Serve (FCFS) timeslot-channel allocation, which has been utilized as an alternative to a scheduling algorithm, is developed for a comparison with SA in a wireless multi-hop tree-based sensor network.

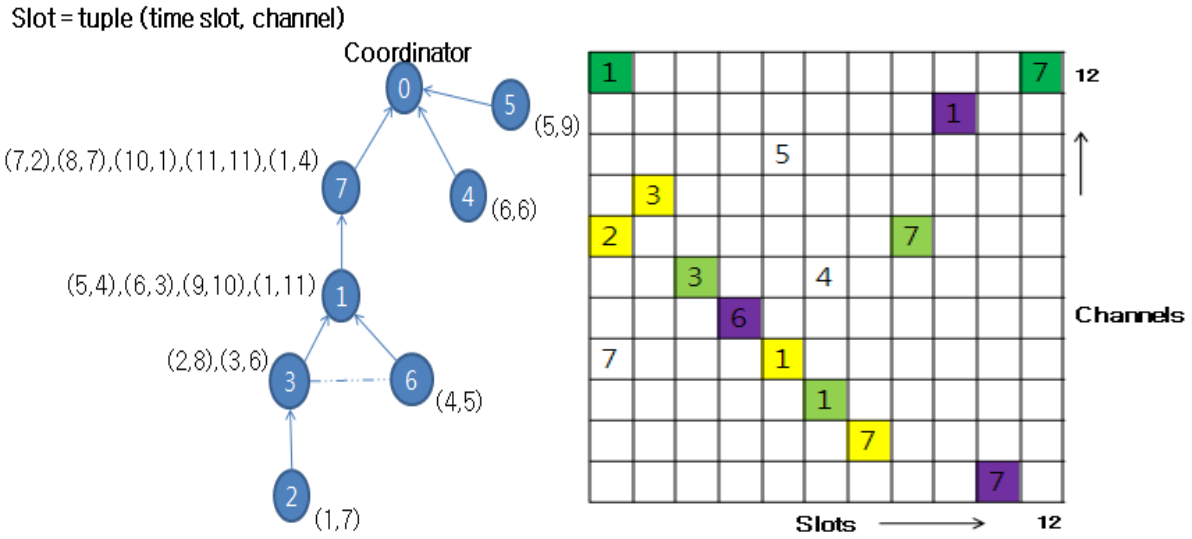


Figure 3.4 FCFS-based slot scheduling

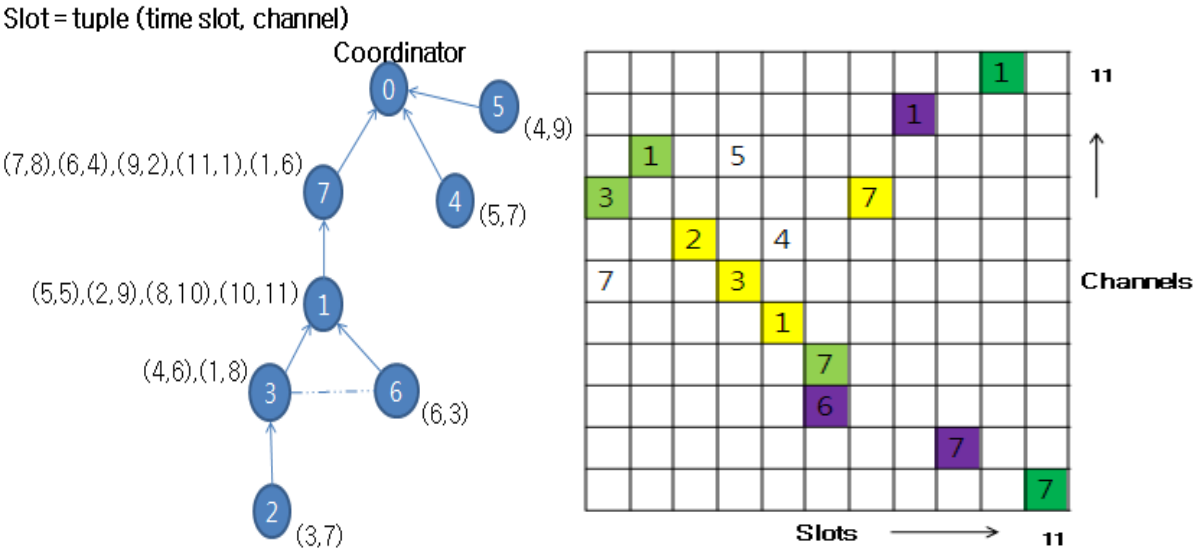


Figure 3.5 SA-based slot scheduling

We conduct a preliminary simulation of FCFS-based and SA-based schedules to investigate their characteristics in terms of the end-to-end delay. From the provided topology, there are fifteen timeslots required in total for each node to deliver to the final destination. After simulating in both FCFS-based and SA-based, only eleven timeslots (minimum number of slots) are needed (slot = tuple (timeslot, channel)) due to space reuse, which reduces the number of required time slots. The above Fig. 3.4 exhibits that the maximum end-to-end delay occurs at source 1 through 7 to the final destination (10), i.e. the node 1 that is treated unequally suffers in significant delay (10). The maximum delay in Fig. 3.5, on the other hands, shows at source 3 through 1, 7 to the coordinator (5), i.e. all nodes being treated equally enable to achieve the guaranteed minimum end-to-end delay (5). In terms of end-to-end delay, the delay in FCFS-based schedule is twice that in SA-based schedule (>>100%). Due to instantaneous channel characteristics, instead of finding a global optimal value at the cost of fully overhead (4 in figures), SA might determine a suboptimal value (5 in Fig. 3.5), i.e. an extremely large system could not be employed in a global optimization because of extremely computation time.

We perform further the simulation to verify the performance of the SA-based schedule to result in an interference-free multi-channel and contention-free slot algorithm. The simulation is carried out on a 500m x 500m field within which there are (∈[25,50]) and on a 1Km x 1Km field within which there are∈([75,100]) nodes being distributed randomly along with communication range 180. Each node calculates its distance for Distance matrix, in which connectivity matrix is constructed by a communication range for one-hop relationships. Furthermore, interference matrix is constructed as well for two-hop distances. We run simulations with 10 difference topologies with four different network sizes, for a total of thirty

simulation runs. The reason for using different fields for different nodes is to maintain the number of hops in the networks. The ratio of connectivity matrix between nodes in diverse fields remains in static, forming a multi-hop environment.

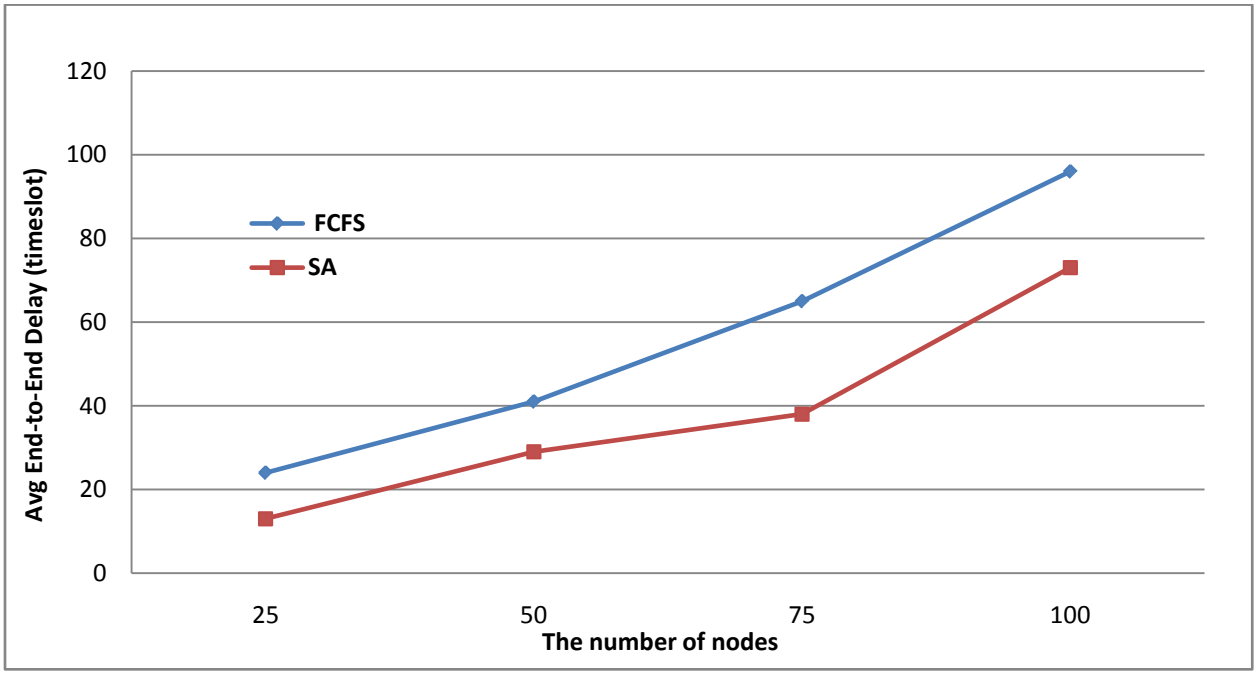


Figure 3.6 Maximum end-to-end delay vs. the number of nodes

The Fig. 3.6 shows the e2e delay of FCFS increases proportionally to the number of nodes so as to imply that the delay is dependent on the number of nodes with a static increase of the number of hops. As a result, FCFS is not suitable for multi-hop environment scenarios due to the lack of a multi-hop view. SA-based algorithms, on the other hand, relates the e2e delay to both the number of hops and the number of nodes so as to reduce the delay significantly, even when the number of nodes is increased.

In conclusion, the timeslot-channel allocation of FCFS-based schedule is performed without considering factors such as other nodes' timeslot-channel allocation and a multi-hop concept,

leading to an unequal-importance node in scheduling timeslot and channel, i.e. it becomes a selfish schedule in terms of e2e delay. The timeslot-channel allocation of SA-based schedule, on the other hand, is consistently weighted for timeslot-channel allocation of the other nodes during the simulation due to the equal-importance node in scheduling timeslot and channel, i.e. it becomes an unselfish (fair) schedule in terms of e2e delay.

3.6 Performance Evaluation of SA and PSO

The performance evaluation tests the feasibility of using SA and PSO in scheduling channels and timeslots to meet a given delay bound, and compares the performances of these two metaheuristic approaches (without any given e2e delay bound). With these two objectives in mind, it is necessary to tune several parameters used for SA and PSO in order to obtain the algorithms' best results for each network size.

3.6.1 Calibration of SA and PSO

First, we calibrate initial temperatures for SA algorithm and observe the performances of the four cooling schedule functions (exponential, Boltzmann, fast and linear) for different network sizes (25, 50, 75 and 100). In order to account for the standard deviation of the energy functions ($\sigma_{\mathcal{E}}$), we set up four different initial temperatures: 50, 100, 150, and 200 to determine the optimal starting temperature, T_0. According to the simulations, the fast cooling schedule produces the smallest $\sigma_{\mathcal{E}}$ with T_0 of 50 in the 25-node network and T_0 of 100 in the 50-node network, respectively. In the 75-node network, the exponential cooling schedule produces the smallest $\sigma_{\mathcal{E}}$with T_0 of 150. In the 100-node network, $\sigma_{\mathcal{E}}$ significantly increases in all cooling schedules and

exponential cooling produces the smallest $\sigma_{\mathcal{E}}$on average, and T_0 of 200 is suitable for 100 nodes.

The w in PSO performs as a linearly decreasing function from 1 to 0.1 over the first 1000 iterations, and remains at 0.1 for the remainder of the simulation of all the network sizes. The c_1 and c_2 depend on the search space, in which the range of movement for a particle would be adjusted based on the network sizes. Thus, the c_1 and c_2 decrease linearly from 2 to 1, from 3 to 1, from 4 to 1, and from 5 to 1, over the first 1000 iterations and remains at 1, for the 25-node, 50-node, 75-node, and 100-node networks, respectively.

3.6.2 Simulation Setting

To compare the performances of SA and PSO for each of the four network sizes, we test them against three different stopping conditions: delay bound, iteration, and iteration threshold for stalled function. A stalled function is the energy function that is invariant for a certain number of iterations. The simulation is carried out on a 500 m x 500 m field for 25 and 50 nodes, and on a 1 km x 1 km field for 75 and 100 nodes. All nodes are distributed uniformly. The number of channels used for our simulations is three because in practice there are three non-overlapping channels for IEEE 802.15.4 devices in 2.4 Ghz ISM band when coexisting with IEEE 802.11 networks. Communication range (CR) is 180 m. The tree is constructed so that the path from any node to the coordinator is the shortest path based on the hop count, e.g., a red line in Fig 3.7 below. Each node calculates the distances to its neighbors, and the connectivity relationship is constructed within CR for one-hop neighbors, e.g., a solid line in Fig 3.7. below. Furthermore, the interference relationship is constructed within CR for two-hop neighbors, who are not in one-hop neighbors, e.g., a dot line in Fig 3.7 below. The two-hop neighbors of a certain node are not able to establish a connection with the node, but they interfere with the node's connection to its

one-hop neighbors. The simulation is conducted with 3 different topologies and 10 different seeds along with 4 different network sizes.

3.6.3 Feasibility of using SA and PSO

We simulate four network sizes: 25-node, 50-node, 75-node, and 100-node networks. Table 1 shows that the total number of timeslots (hops) required for each node to deliver data to the coordinator is 39, 85, 150, and 313 timeslots, respectively. The different topologies would lead to the different total number of timeslots required. In order to achieve consistent evaluation, we ensure that the total number of connections (hops) in any two topologies differs by less than 5 percent. Assume the length of a timeslot is 1 ms and a single packet corresponds to a timeslot. For example, if the e2e delay requirements are arbitrarily given as 15 ms, 33 ms, 64 ms, and 130 ms for the 25-node, the 50-node, the 75-node, and the 100-node networks, respectively, the suboptimal values of SA and PSO up to 15, 33, 64, and 130 delay bound timeslots (in Table 3.1) for the 25-node, the 50-node, the 75-node and the 100-node networks satisfy the e2e delay requirements. Thus, scheduling channels and timeslots via SA and PSO can be considered feasible. If a path satisfies the e2e delay bound, the metaheuristic algorithm excludes that path from the search, and continues to check the rest of the paths until all paths satisfy the e2e delay bound. Some particles are already settling down to the target (early arrival to the given e2e delay bound) and the others are still continuing to search. Note that neither algorithm is able to determine the global optimal values for all network sizes due to the involved computation time, but they both achieve suboptimal values that still satisfy the e2e delay requirement in all network sizes. As expected, networks with longer paths require more computation time than networks with shorter paths due to the fact that the shorter paths are able to quickly satisfy a given delay

bound.

25 Nodes

Delay Bound (timeslot)	3	5	9	12	15	18	21	24	27	33	36	39
SA (iteration)	Fail	112.9	63.38	50.92	41.68	26.93	15.43	11.31	7.69	5.88	4.64	2.52
PSO (iteration)	2115	108.6	84.27	61.69	52.63	23.28	16.85	10.52	8.75	7.96	5.38	2.71

50 Nodes

Delay Bound (timeslot)	6	12	19	26	33	40	47	54	61	67	74	85
SA (iteration)	Fail	Fail	383.5	117.4	78.23	52.79	32.04	28.15	14.73	8.72	6.43	4.39
PSO (iteration)	3273	216	137.4	85.18	57.39	42.31	28.74	16.83	15.92	9.96	7.34	5.22

75 Nodes

Delay Bound (timeslot)	21	32	42	53	64	75	86	97	108	119	130	150
SA (iteration)	Fail	Fail	615.9	177.4	105.4	92.46	74.23	57.49	38.51	23.96	11.63	9.48
PSO (iteration)	4687	1298	1168	304.1	81.4	58.59	57.67	34.01	33.81	18.73	9.6	7.71

100 Nodes

Delay Bound (timeslot)	38	61	75	107	130	153	176	199	224	248	272	313
SA (iteration)	Fail	Fail	985.6	245.9	176.7	118.6	94.57	78.42	56.39	46.81	24.39	18.72
PSO (iteratiion)	6404	4644	1388	344.7	207.1	184.1	159.4	81.24	63.17	54.2	37.09	16.26

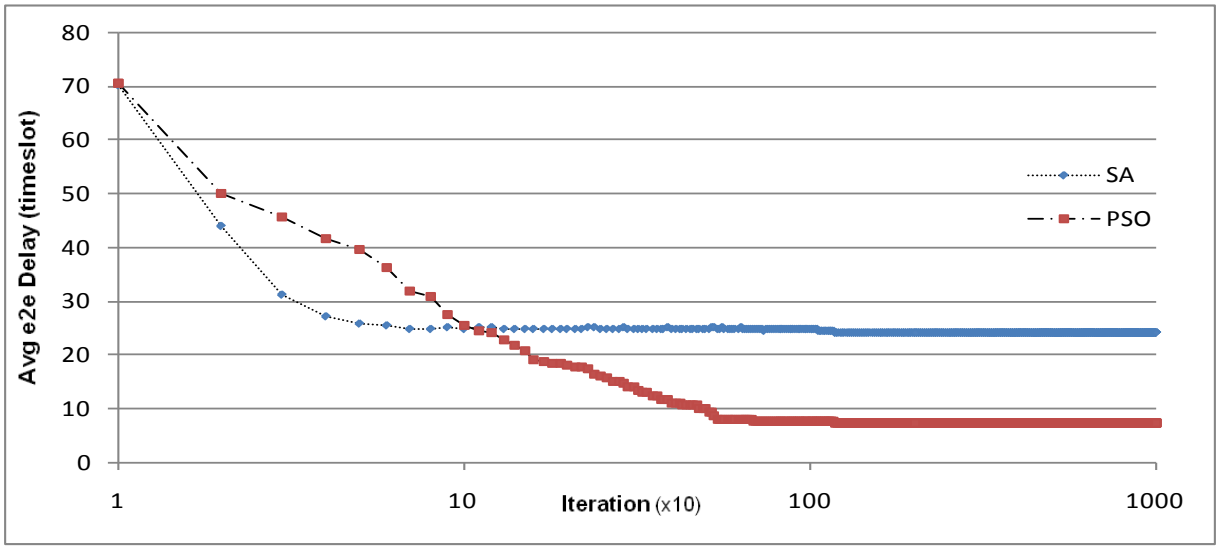
Table 3.1The feasibility of scheduling by SA and PSO

3.6.4 Performance comparison between SA and PSO

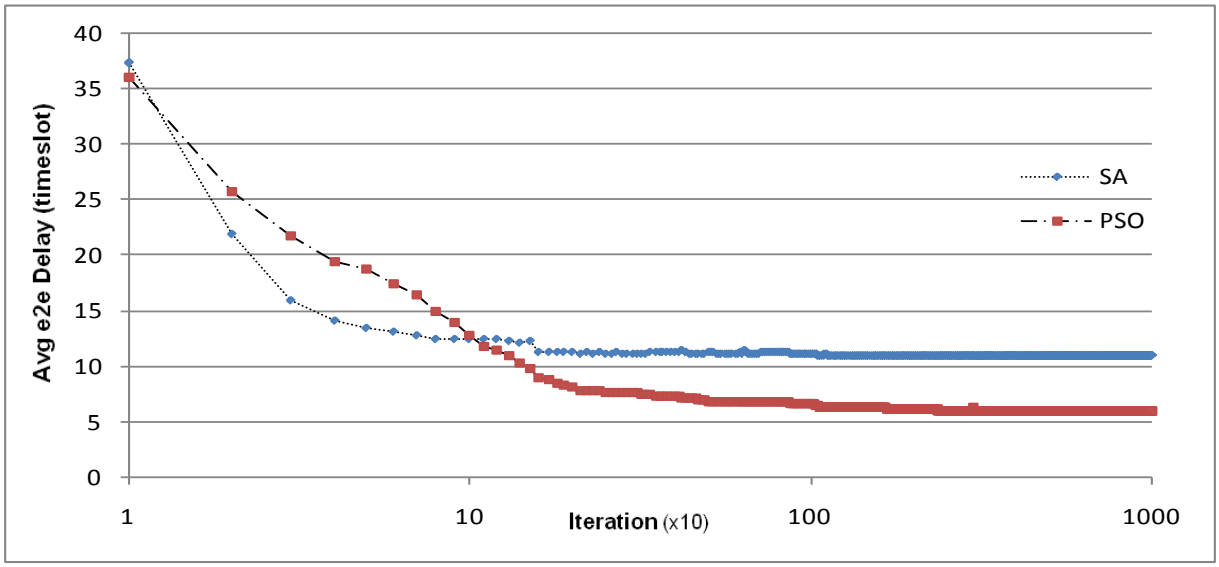
In Fig. 3.8, the maximum e2e delay is presented with respect to the number of iteration with a

log scale. Two curves are crossing in all four network sizes as we expected. SA searches for a solution faster and gets trapped in local minimum because of infrequent movement (e.g., low temperature) while PSO searches for a solution constantly and exceeds the quality of the SA's solution by continuously sharing information about the target towards the global optimum.

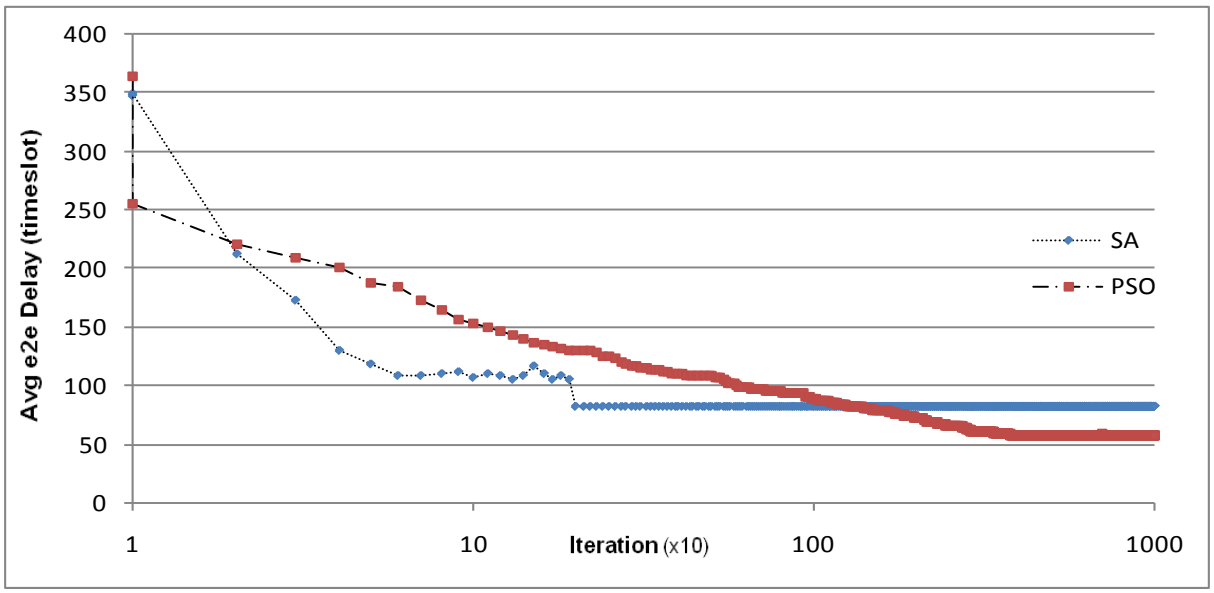
25 Nodes



50 Nodes



75 Nodes



100 Nodes

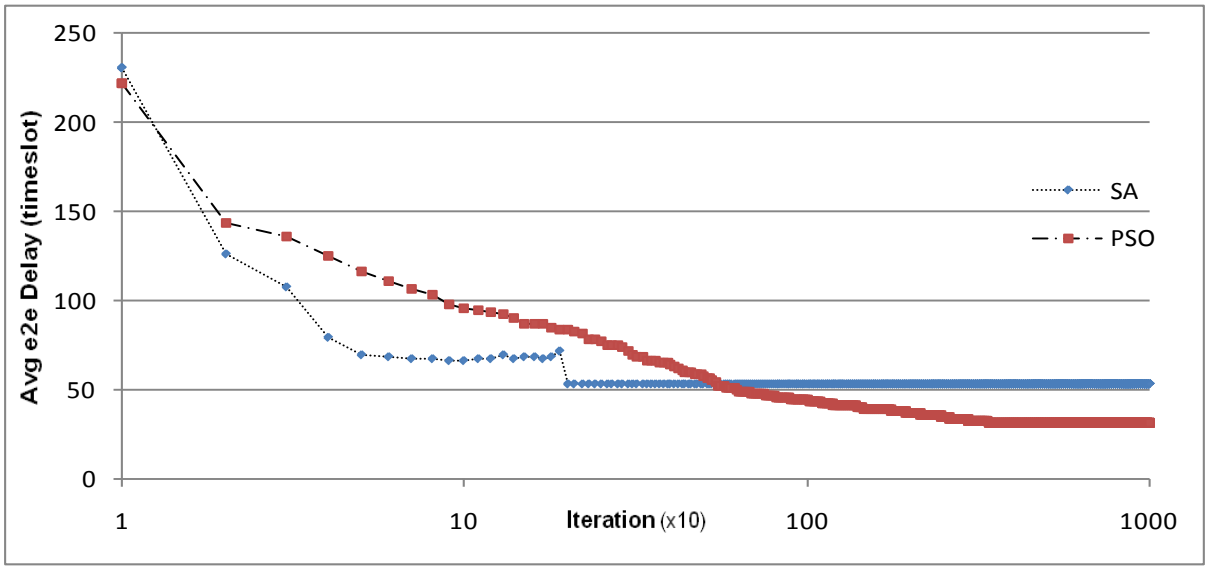


Figure 3.8 Maximum e2e delay in terms of iteration

In the 25-node network, the max e2e delay of PSO is approximately a half of SA in Table 3.2, which is also shown in Fig. 3.8. More prominently, in the 50-node network, the delay of PSO is less than one third of SA. In the 75-node and 100-node network, the noticeable advantage of PSO is reduced, but its delay is still approximately a half of SA's. In terms of standard deviation, SA maintains consistently smaller values in all four network sizes because of its annealing and governing behavior.

25 Nodes

	Mean (Avg)	Mean (Std)
SA	10.3	0.3
PSO	5.5	0.7

50 Nodes

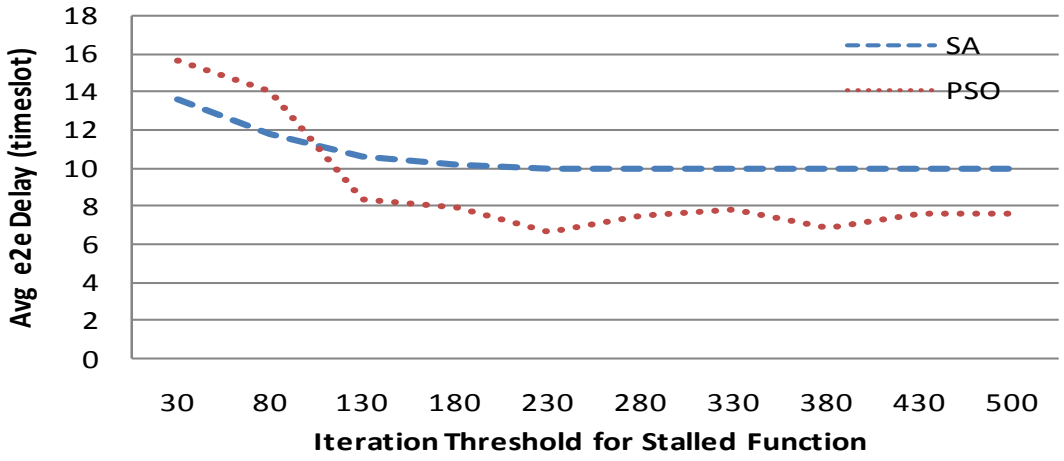
	Mean (Avg)	Mean (Std)
SA	22.9	0.7
PSO	7.1	1.6

	75 Nodes			100 Nodes	
	Mean (Avg)	Mean (Std)		Mean (Avg)	Mean (Std)
SA	53.5	4.3	SA	83.9	6.8
PSO	28.6	6.9	PSO	54.8	10.9

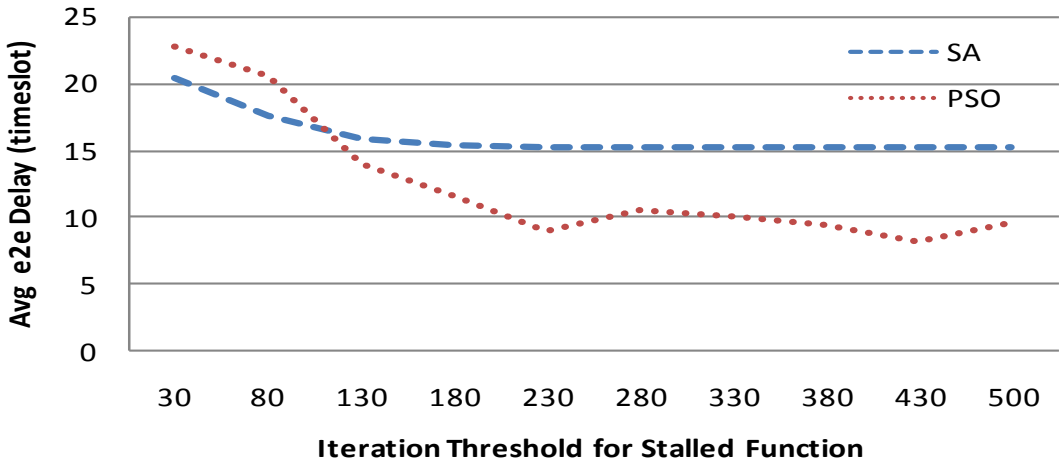
Table 3.2 Comparion of maximum e2e delay statistics between SA and PSO after 10000 iterations

In Fig. 3.9, the quality of the initial solution of SA exceeds that of PSO, but again, a crossover happens as the iterations progress. When the values of the stalled function are 80 and 180 iterations for the 25-node network, the energy function is not changed during the last 80 and 180 iterations with corresponding max e2e-delays of SA and PSO at 12 and 14 (80 iterations), and 10 and 8 (180 iterations).

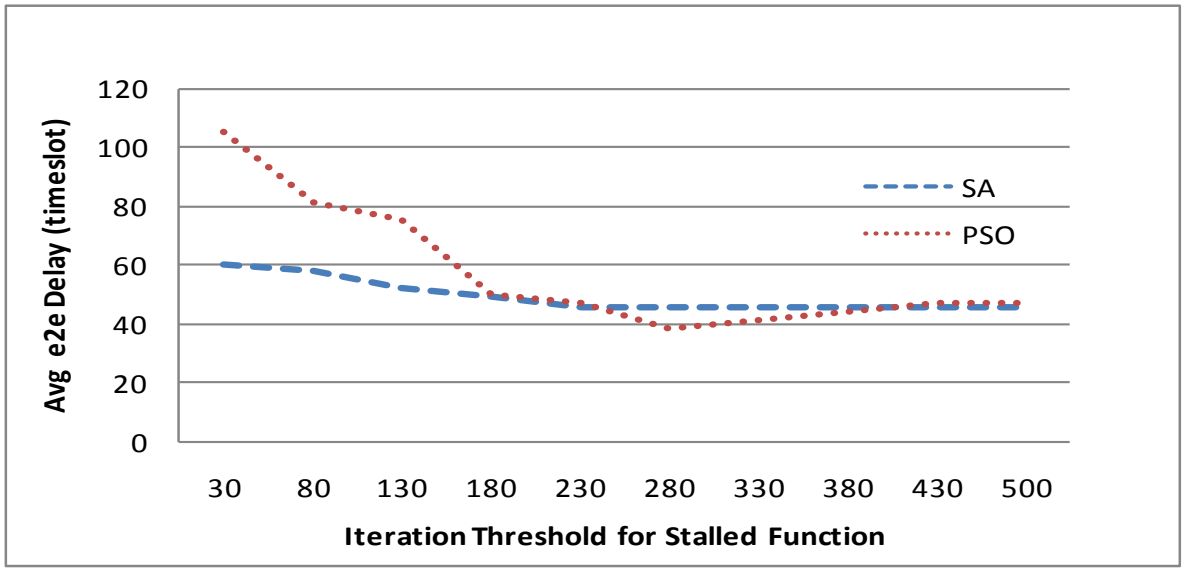
25 Nodes



50 Nodes



75 Nodes



100 Nodes

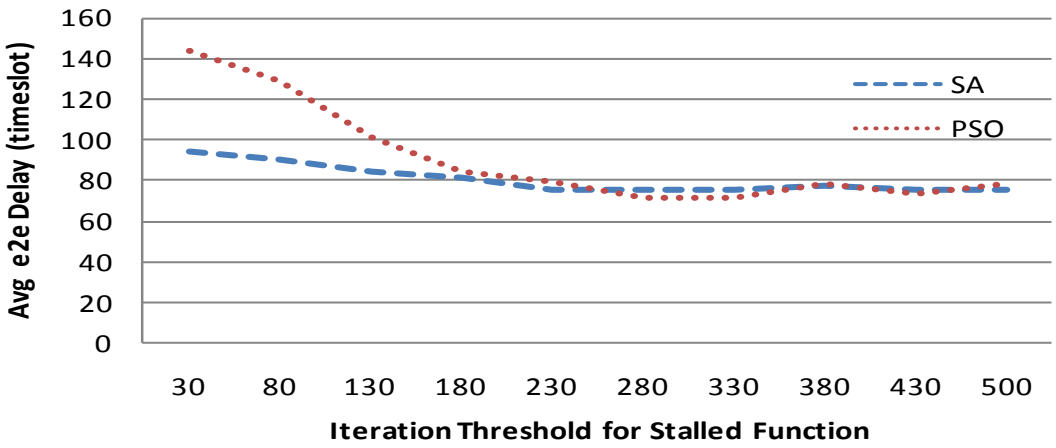


Figure 3.9 Maximum e2e delay in terms of iteration threshold for stalled function

In summary, SA searches blindly for the target, ultimately discovering it. On the other hand, in PSO, if one of the particles knows the target and has biased information at each iteration, that

specific particle at that specific iteration is assumed to know the location of the target, and all other particles follow its direction. However, the deviated direction is accumulated at each iteration, and if it is not moving in the right direction, the particles eventually end up at a wrong location, i.e., a local minimum. This may be one of motivations that a hybrid approach can exploit.

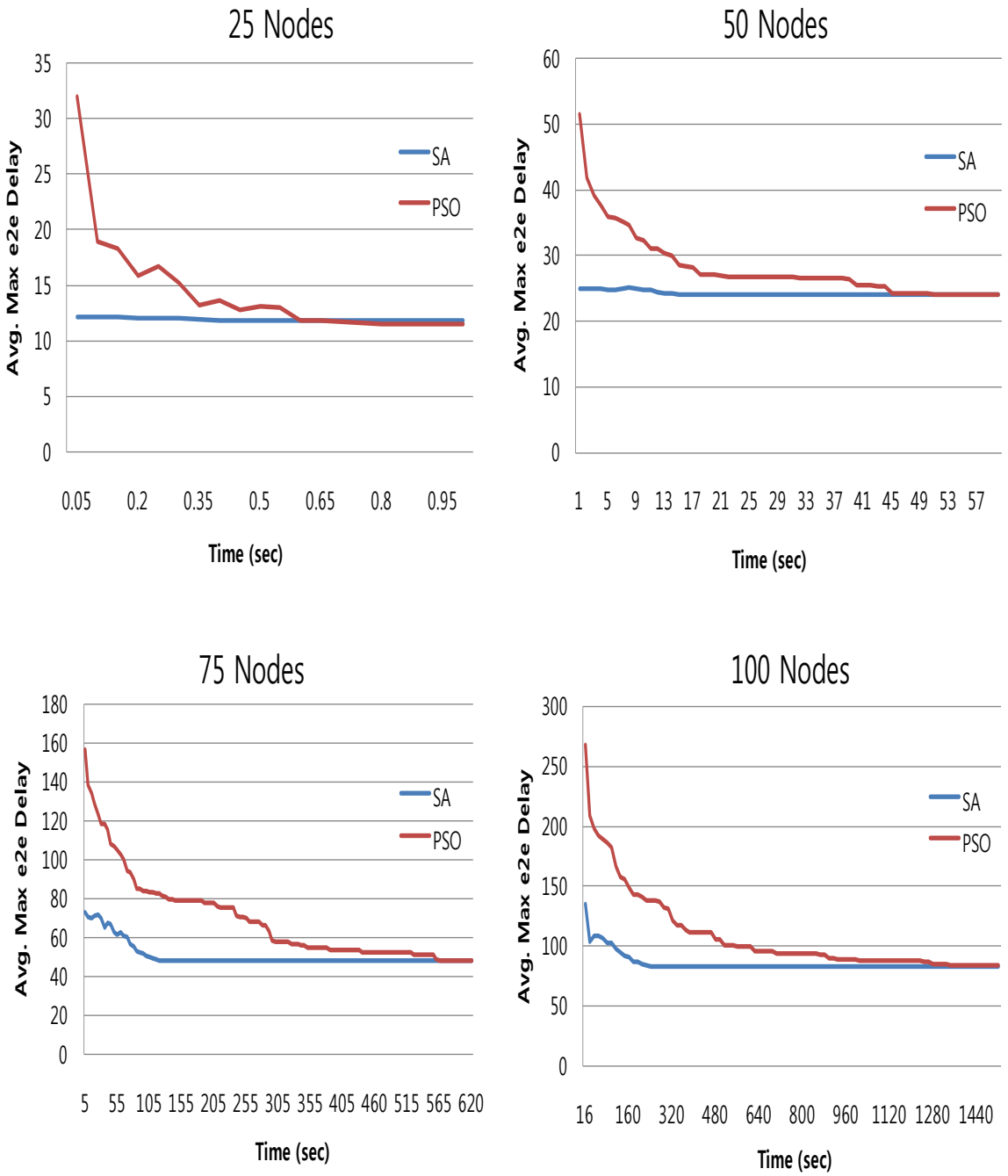


Figure 3.10 Maximum e2e delay in terms of a desired time

The time spent on scheduling timeslots and channels would be a fair and reasonable measure for comparing the performances of SA-based scheduling and PSO-based scheduling due to the following two factors. First, comparing the two solely based on a single iteration would be an unfair comparison since SA might be able to run 60 iterations with relatively low per-iteration solution quality while PSO might only be able to run three iterations with high per-iteration solution quality within one minute in a 100-node network, depending on the computer specification. Therefore, it would be fairer to compare the two over the same time period. Second, comparing the two in terms of the stalled function would be unreasonable since the number of particles/solutions is different for SA and PSO. The SA has one particle/solution while the PSO has multiple particles. For example, if there is a desired time of 10 seconds, the SA is performed in more iterations than the PSO in order to compensate for having just one particle, i.e., frequent iteration is referred to the frequent decision process. In terms of the time, SA in Fig. 3.10 is superior to PSO because the SA is able to quickly determine a stable suboptimal e2e delay due to its relative simplicity, which affords large initial exploration, while the PSO requires heavy initial exploitation due to its complexity.

3.6.5 Summary of Results

For a fair comparison, we implement the two schedules at the best-possible performance levels. SA-based annealing is based on stalled function while PSO based annealing is based on iteration. In terms of Standard Deviation, SA-based scheduling exhibits significantly higher stability than PSO-based scheduling because of its annealing nature. However, the solution quality of SA-based scheduling is relatively poor when compared to that of PSO-based scheduling due to the following two factors. First, at a nearby optimal solution, a particle of PSO moves frequently so

as to exploit diligently while a particle of SA is unwilling to move around once it settles down. Second, PSO has an interactive attribute of particles sharing suboptimal directions that not only lets particles adopt merits of other particles, but also forces particles to move. However, PSO-based scheduling often falls into local minimum in a large search space (in the 50-and-100 node network). For example, if one of the particles has biased information at each iteration, that specific particle at that specific iteration is assumed to know the location of the target, and all others follow its direction. However, the deviated direction is accumulated at each iteration, and if it is not the right direction, the particles eventually would end up at a wrong location (e.g., true negative). On the other hand, SA-based scheduling searches blindly for the target, ultimately discovering it (e.g., true positive). In terms of the computation time, SA is significantly faster than PSO because of its relative simplicity.

3.7 Computation Comparison between SA and PSO

The computation time (how much time is spent in a single iteration) depends on the computer specification. We use Intel® Xeon® CPU W3530 @ 2.80GHz Processor, 4GB memory, and 32-bit Operating System. The following is summarized the computation time comparison of the two algorithms for a single iteration in terms of network sizes.

Algorithm Node	25	50	75	100
SA	0.0119 sec	0.2328 sec	0.8259 sec	1.4726 sec
PSO	0.0439 sec	0.5412 sec	4.3051 sec	15.1903 sec

Table 3.3 computation comparison between SA and PSO

The computation of PSO increases exponentially while that of SA increases linearly as the number of nodes increases. The complexity of PSO relies on the number of particles (their velocity vector) and the number of nodes, while the complexity of SA solely depends on the number of nodes in the network. For example, SA in the 100-node network takes 1.4726 sec and 1.4726 x 10,000 iterations equals 4.0906 hrs, while PSO takes 15.1903 sec and 15.1903 x 10,000 iterations equals 42.1953 hrs (1.58 days). There would be a trade-off between the solution quality and computation time, e.g., fast computation time is desired at the expense of the solution quality whenever the algorithm has to rerun in the middle of the operation. Thus, the network administer is supposed to deal with it in terms of the posed wireless sensor system.

3.8 Conclusion

In this article, we formulate a combinatorial optimization for a scheduling problem in time-constrained WSNs using a mixed-integer programming, focusing on suboptimal values rather than the optimal value because our goal is to satisfy a tolerable range of end-to-end delays. We adopt two metaheuristic algorithms: SA and PSO to schedule both channels and timeslots in a tree topology to satisfy the end-to-end delay bound. The directionality in the arrangement of timeslots to minimize the end-to-end delay is exploited to better fit the operations of PSO and SA. Through simulations, the end-to-end delay of the PSO-based scheduling is shown to achieve either a half or one-third of that achieved by the SA-based scheduling. However, the computation of PSO increases exponentially while that of SA increases linearly as the number of nodes increases. There would be a trade-off between the solution quality and computation time, e.g., fast computation time is desired at the expense of the solution quality whenever the algorithm has to rerun in the middle of the operation. We will further study distributed metaheuristics for a

mesh-based peer-to-peer network environment, where no centralized coordinator exists.

Chapter 4

HEURISTIC

4.1 Introduction

Wireless sensor network (WSN) becomes a de facto wireless communications technology in various practical applications, such as environment monitoring, patient monitoring, and many industrial applications. Wireless sensors can potentially be deployed in a large geographical area via multi-hop communication. A widely deployed sensor node establishes a tree-route towards a data collector, namely the coordinator, and transmits sensing data or other information through the forwarding nodes.

Unlike delay-tolerant applications, patient monitoring, disaster warning, intruder detection, and many industrial applications require a timely data delivery. Deterministic low latency is one of the most important and challenging requirements in real-time monitoring and control. Moreover, industrial applications, such as process monitoring, factory automation, and factory control, require both prompt data delivery and guaranteed end-to-end delay for all possible sources of data because belated data from just one source of data may render the entire system useless. Belated information is meaningless and it can also cause serious problems in controlling processes and equipments.

The traffic in the industrial applications is usually periodic, making Time Division Multiple Access (TDMA) approach feasible. TDMA allocates exclusive collision-free timeslot(s) to each node, and minimizes end-to-end delay (e2e) between a source node and the data center. Each TDMA timeslot is extended to accommodate multiple channels as in the IEEE Standard

802.15.4e [5] to guarantee e2e delay. Then, each node has the flexibility of utilizing a combination of channel and timeslot in order to achieve expedited data packet transmission towards the coordinator, resulting in minimum e2e delay.

There are inherently two factors that hinder efficient data collection in WSNs: half-duplex nature of transceiver and interference [7], [8]. In WSNs, each sensor node is typically equipped with a single half-duplex radio transceiver, i.e. each node cannot transmit and receive simultaneously, nor can it function on different channel at the same time. In a tree topology, the data traffic always flows from a child (bottom) to a parent (upside) towards the coordinator. To cope with interference, one assigns different channels for receivers with interfering links (parents) because interference occurs at a receiver, rather than a transmitter.

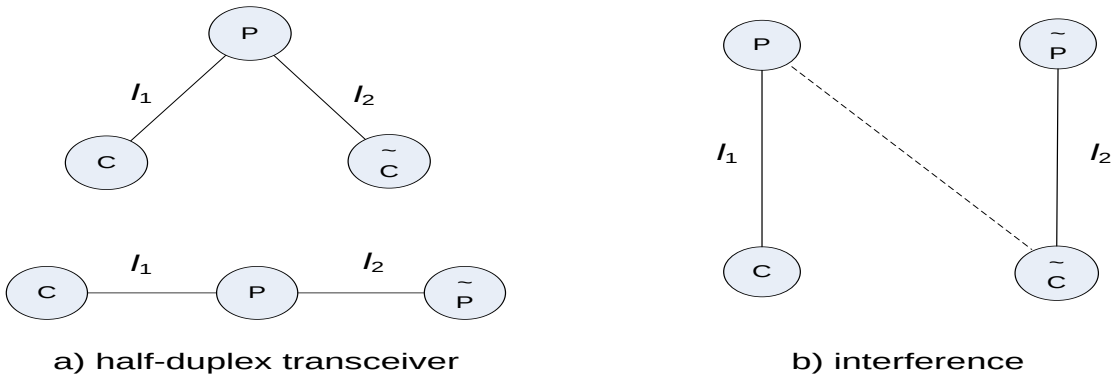


Figure 4.1 The definition of constraints (solid lines are communication links while a dotted line is interference link)

Interfering parents are defined as a pair of parents p and $\tilde{p}$ where the transmission from the child $\tilde{c}$ to its parent $\tilde{p}$ interferes with the transmission from the child c to its own parent p , and vice versa in Fig. 4.1b. All the interfering links are eliminated by assigning different channel a pair of interfering links as long as there are an enough number of available channels [7], [8]. If

available channels are not enough, the remaining interfering links can be considered by allocating timeslots with constraints at the sender side, thereby transmitting data packet on the designated timeslot using that channel.

Formulating collision-free TDMA schedules has been proved to be an NP-complete problem [8]. The adoption of multiple channels into each TDMA makes the scheduling even more challenging. Then, we propose a greedy heuristic scheduling algorithm for channel and TDMA allocations to minimize guaranteed e2e delay.

4.2 Network Model and Problem formulation

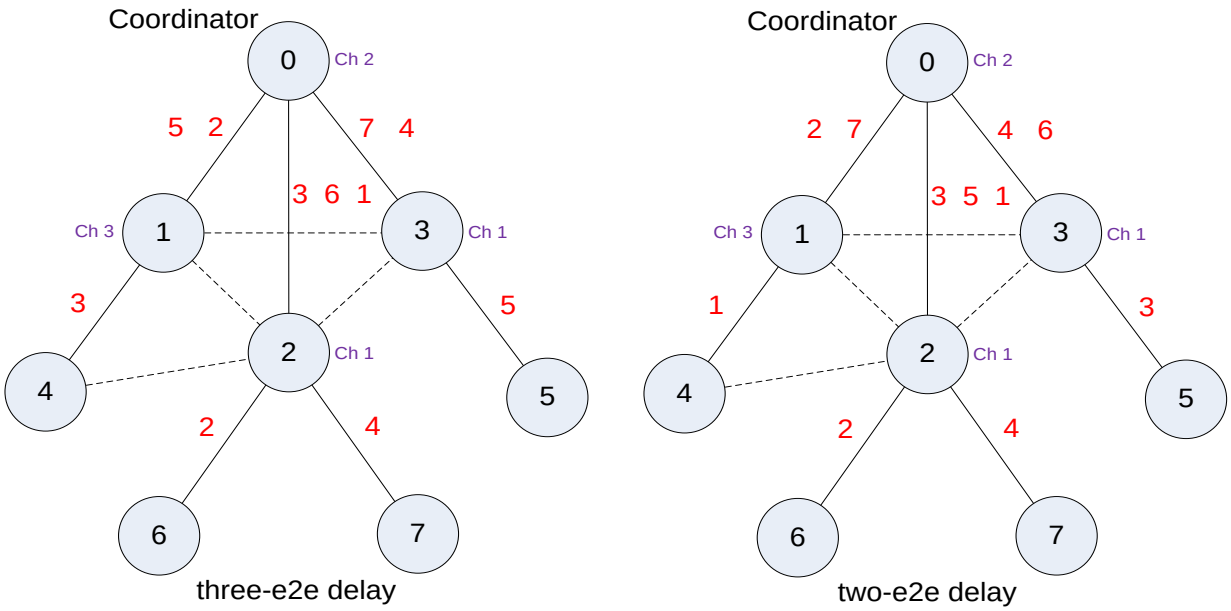
We model the multi-hop WSN as a graph $G = (V, E)$, where V is the set of nodes, and $E = \{(i, j) \mid i, j \in V\}$ is the set of edges representing the wireless links. A designated node $s \in V$ denotes the sink. The Euclidean distance between two nodes, i and j is denoted by d_{ij}. Each node in the tree topology generates a single packet, and each intermediate node would have two roles: a sender and a forwarder. The packets from all the nodes are destined to the coordinator, which is the typical scenario for data collection in WSNs. Our objective is to minimize e2e delay for each node generating and forwarding packets towards the coordinator while considering the following two constraints [8]

- *Adjacency constraint*: two edges $(i, j) \in E$ and $(k, l) \in E$ cannot be scheduled in the same timeslot if they are adjacent to each other, i.e., $\{i, j\} \cap \{k, l\} \neq \emptyset$ due to the half-duplex transceiver in Fig. 4.1a.
- *Interfering constraint*: two edges $(i, j) \in E$ and $(k, l) \in E$ cannot be scheduled simultaneously if (i, j) or (k, l) is an interference link in Fig. 4.1b.

4.3 End-two-end Delay-based Channel and Timeslot Scheduling

4.3.1 Existing approach (non-e2e delay-based):

For simplicity, Incel et al. [7] studies two separate scheduling problems: scheduling for multi-channel at a receiver side and multi-timeslot at a sender side. The receiver-based channel scheduling assigns the channels to the receivers (i.e., parents) in order to eliminate all the interfering links at a receiver side [7]. All potential receivers calculate amount of interference. If the SINR value of a pair of parent and child (a link) is less than that of that child which belongs to other parent, the interfering parent is interfered with the other parent of that child node. After each interfering parent has a list of other interfering parents, the algorithm conducts channel assignment. The most interfered parent who has the largest amount of interference is assigned a channel from the available channels. If a channel is not available, the link of parent and child remains the same channel.



a) non-e2e delay-based b) e2e delay-based

Figure 4.2 The delay comparison of the non-e2e delay-based and e2e delay-based algorithms (the numbers on the links represent timeslots)

Once the channel assignment has been completed at a receiver side to remove potential interferences, timeslot allocation is performed at the sender side to increase parallel transmission along multiple branches. In a given timeslot, if a branch with the highest number of remaining nodes/packets is scheduled, (e.g., the middle branch in Fig. 4.2) [7], timeslot allocation occurs at a node with consecutive timeslot numbers, assuming timeslot is an integer from 1 to some arbitrary number. For example, in Fig. 4.2, the node 2 requires three timeslots. Then, the node 2 would be assigned three consecutive timeslots, 1, 2 and 3, which would increase the e2e delay for the node 6 and 7, because their paths are neither weighed nor treated equally in terms of e2e delay.

4.3.2 Proposed approach (e2e delay-based):

As with the receiver-based channel scheduling [7], [8], the algorithm first determines interfering parents with RSSI due to simplicity according to the definition of interfering parent. We assume each node has eight different transmission power levels (P), and the value of RSSI refers to the distance between two nodes (d_{ij}) due to RSSI$\propto d_{ij}$. Note here that the node ID of each node is the same as the path ID of the path starting from that node and ending at the coordinator (e.g., the node 6 has a path, which starts from node 6 via node 2 to the coordinator). Once the channel assignment has been completed at a receiver side [7], timeslot allocation is performed at the sender side to increase parallel transmission along multiple paths, rather than branches. In a given timeslot, there would be more than one eligible path in a tree topology

because the coordinator can receive data packet from one path among many paths due to the half-duplex transceiver. Then, a path with the longest path-length should be scheduled first because each node has a different remaining path-length to the coordinator. The e2e delay-based algorithm schedules timeslots starting from leaf node (child node) for the first available timeslot to the intermediate node for next available timeslot in a staggered way towards the coordinator that ends up minimizing the e2e delay. The major distinction between the non-e2e delay-based [7] and the e2e delay-based algorithms is that the former schedules timeslots in terms of branches while the latter does so by paths because each node has its own path towards the coordinator.

4.3.3 E2E Delay-based Algorithm:

In Fig. 4.2, there are eight nodes including the coordinator and three available channels. The parent nodes are coordinator, node 1, 2, and 3 while the child nodes are node 4, 5, 6, and 7. After calculating interfering parents for each parent node, the coordinator has the most interfered parent and then it is assigned channel number 2 and the next interfered one is the node 1, which is assigned channel number 3. Node 2 and node 3 remain with channel number 1 as defaults since there are three channels available. In view of the e2e delay, non-e2e delay-based algorithm [7] produces three units of delay while e2e delay-based algorithm produces two units of delay for each path except for the node 1, 2, and 3 due to their one-hop distance from the coordinator. Note that both algorithms achieve maximum timeslot number 7.

Notations:

- $R_{n,\tilde{n}}$: a set of all potential paths $(r_{n,\tilde{n}})$
- $d_{r_{n,\tilde{n}}}$: the delay of a path between source n and destination $\tilde{n}$
- $l, \tilde{l}$: the link between node l and $\tilde{l}$ in a path $r_{n,\tilde{n}}$
- $d_{l,\tilde{l}}$: the link delay for $l, \tilde{l}$
- The destination of all paths is $\tilde{n}$, the coordinator in this paper

The goal of the scheduling is to minimize the e2e delay, which is formulated as:

$$D_{\text{e2e delay}} = d_{r_{n,\tilde{n}}} = \min_{r_{n,\tilde{n}} \in R_{n,\tilde{n}}} \left(\sum_{l,\tilde{l}} d_{l,\tilde{l}} \right)$$

Receiver-based Channel Scheduling Algorithm

P:set of parents, *ch*: the number of available channels
CH: channels assigned to the nodes in *P*
for all $p \in P$do // Create list of interfering parents
C: set of children of *p*
$P'(p)$: set of interfering parents of *p*
AC(*p*): set of available channels for parent *p*
$P'(p) \Leftarrow \emptyset$, AC(*p*) $\Leftarrow \{1, 2, ..., ch\}$
for all $c \in C$ and $c' \notin C$ do
if (*RSSI* (*c*, *p*) $< P'(p)$), $P'(p) \Leftarrow$ parent of c'
end for
end for
while $P \neq \emptyset$ do // Channel assignment
$p \leftarrow$ next most interfered parent from *P*
$CH(p) = i$, $i \in AC(p)$
for all $p' \in P'(p)$ do
$P'(p') = P'(p') \setminus p$
$AC(p') = AC(p') \setminus i$
end for
$P'(p) = \emptyset$
$P \leftarrow P \setminus p$
end while

Transmitter-based Timeslot Scheduling Algorithm

P_n :set of nodes of path n (node n)
N_p: timeslots required for all paths
$t \leftarrow 1$
while $N_p > \text{ø}$ for all paths do
  E: set of paths eligible for scheduling at t
  $l_p = \arg\max_{p \in E}\{N_p\}$ // find the longest path
  while $l_p \neq \text{ø}$
    allocate$t$ to that node n of l_n
    $N_p \leftarrow N_p$-1
    $P_n = P_n \setminus$ n
    For all $n \notin N$do
    S_p: set of nodes for parallel transmission
      while $S_p \neq \text{ø}$ do
        allocate t to that node n of S_p
        $N_p \leftarrow N_p$-1
        $P_n = P_n \setminus$ n
        $S_p = \text{ø}$
      end while
    end for
  $t \leftarrow t + 1$
  $l_p = \text{ø}$
  end while
end while

Figure 4.3 The pseudo codes of scheduling algorithms

4.4 Performance Evaluation

We compare the performances of the non-e2e delay-based and e2e delay-based algorithms for each of the 5 network sizes, 25, 50, 75, and 100 nodes. The simulation is carried out on a 500 m x 500 m field for 25 and 50 nodes, and on a 1 km x 1 km field for 75 and 100 nodes. All the nodes are distributed uniformly within the above fields, respectively. The number of channels used here is three and communication range (CR) is 180 m.

The tree is constructed so that the path from any node to the coordinator is the shortest path based on the hop count. Each node calculates the distances to its neighbors, and the connectivity

relationship is constructed within CR for one-hop neighbors. Furthermore, the interfering-parent relationship is constructed through the definition of interfering parent as in [8]. The simulation is conducted with 6 different tree topologies along with 4 different network sizes. The different topologies would lead to the different total number of timeslots required. In order to achieve consistent evaluation, we ensure that the total number of connections (hops) in any two topologies differs by less than 5 percent. Fig. 4.4-4.7 shows an example of 25-node, 50-node, 75-node, and 100-node network topology.

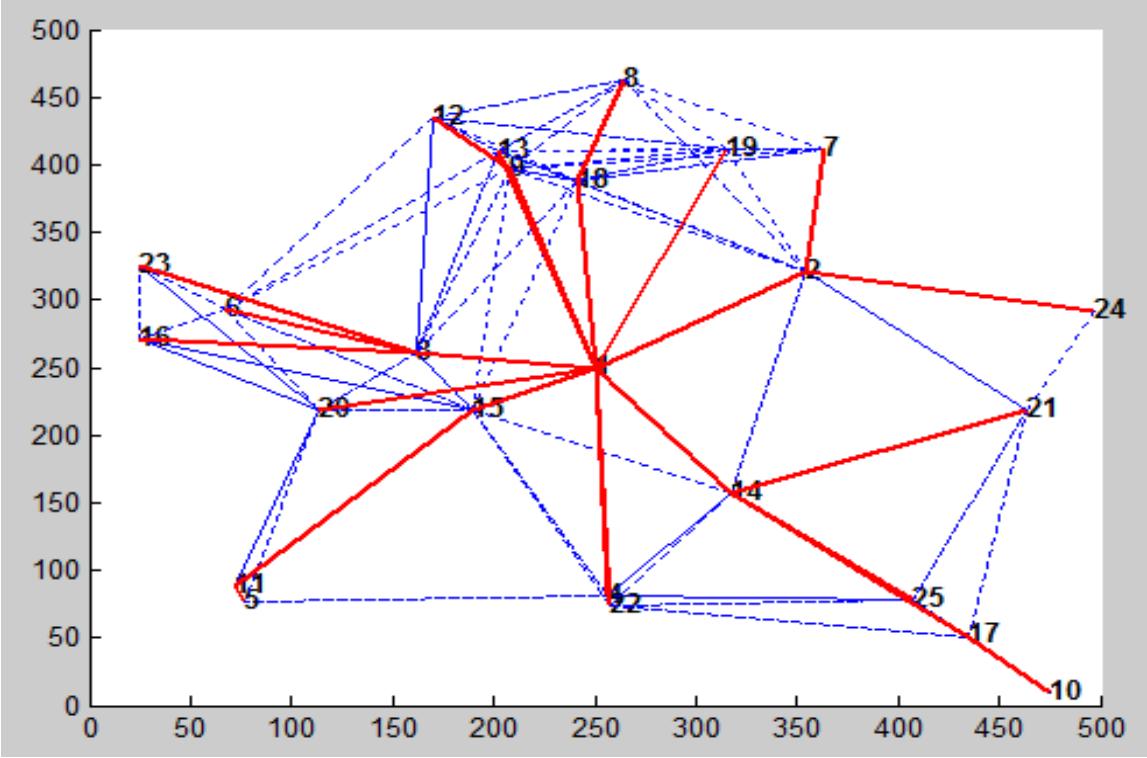


Figure 4.4 A network topology example of 25 nodes

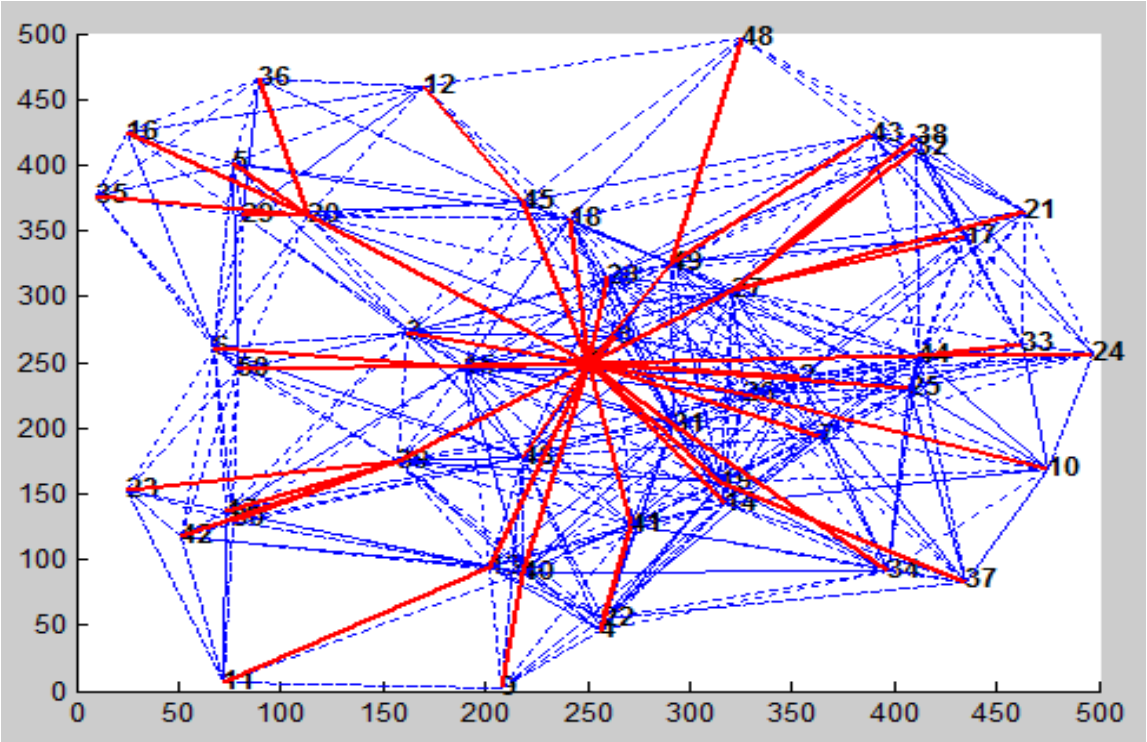


Figure. 4.5 A network topology example of 50 nodes

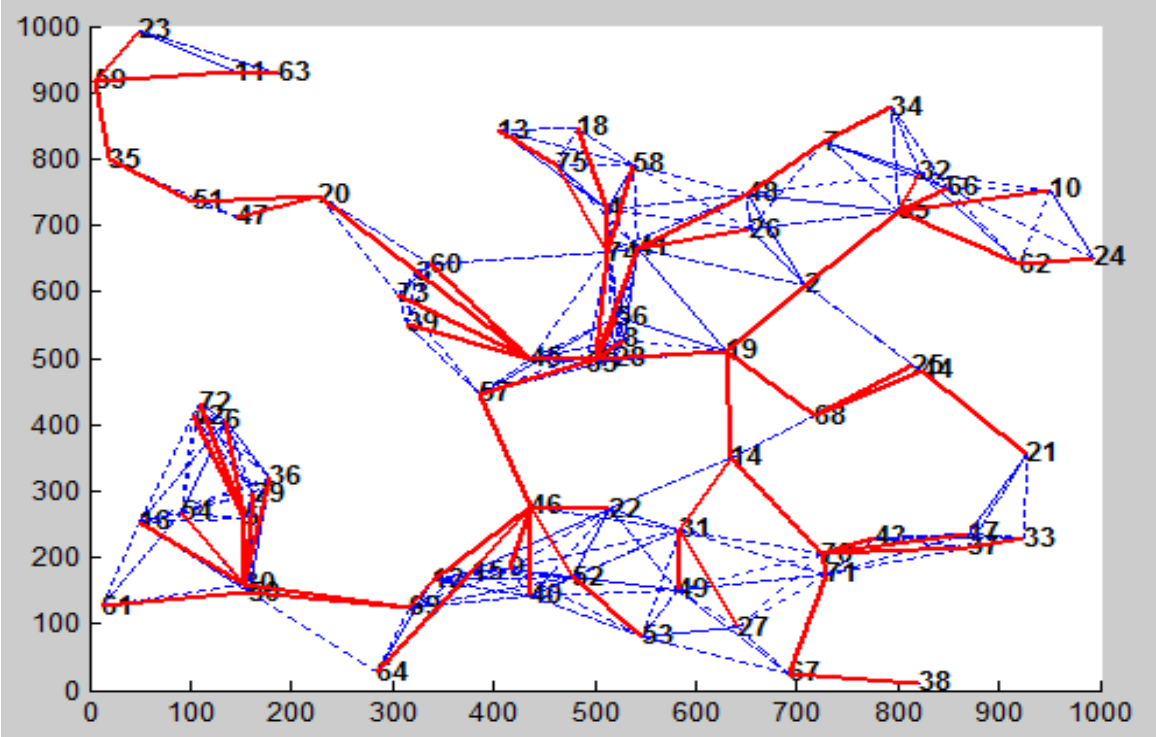


Figure 4.6 A network topology example of 75 nodes

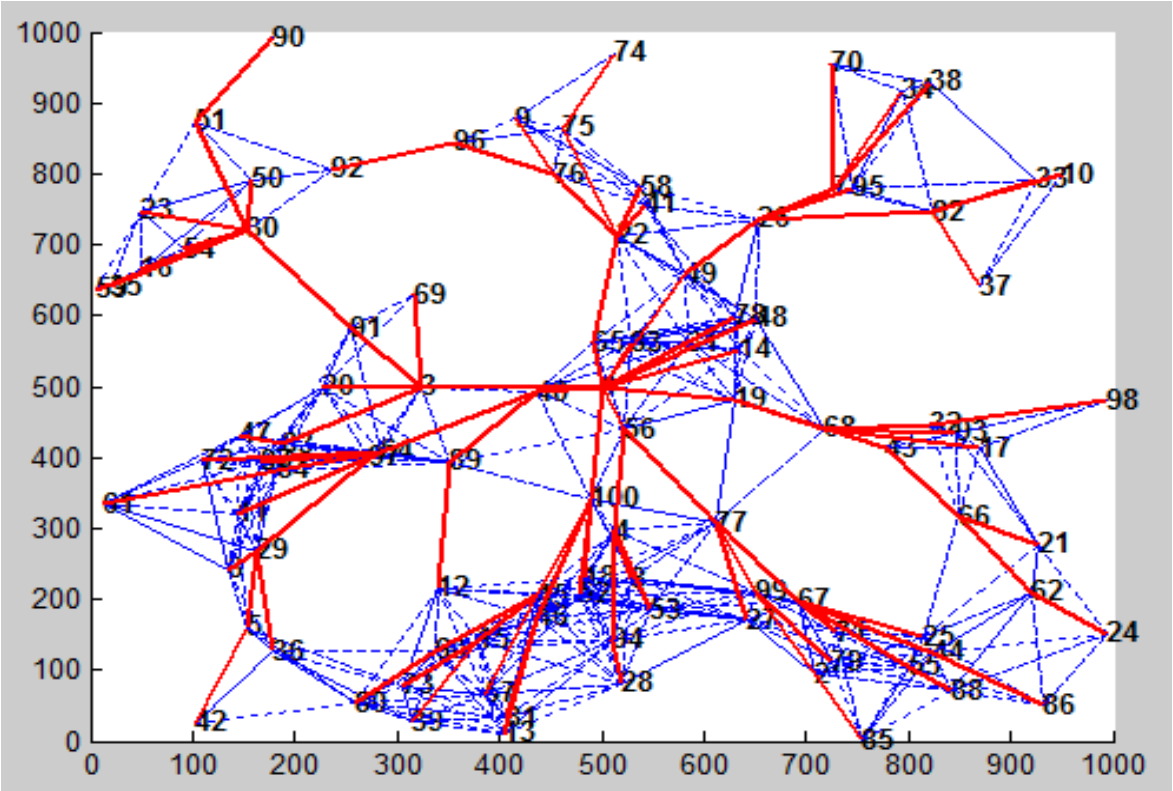


Figure 4.7 A network topology example of 100 nodes

In Fig. 4.8, as we can see, the e2e delay-based algorithm shows guaranteed timeslot 7 (maximum) while the non-e2e delay-based algorithm shows 10 timeslots. For example, if the values of Beacon Order (BO) and Superframe Order (SO) are 2 and 1, respectively, in the IEEE Standard 802.15.4e [5], then the length of a timeslot is 1.92 msec. If the e2e delay bound requirement is 15 msec [5] for all paths in the 25-node network, the achieved delay for the e2e delay-based would be 13.44 msec (7 timeslot x 1.92 msec), while that for the non-e2e delay-based would be 19.2 msec (10 timeslots x 1.92 msec). All paths in the given 25-node network topology satisfy the e2e delay bound requirement with the e2e delay-based algorithm.

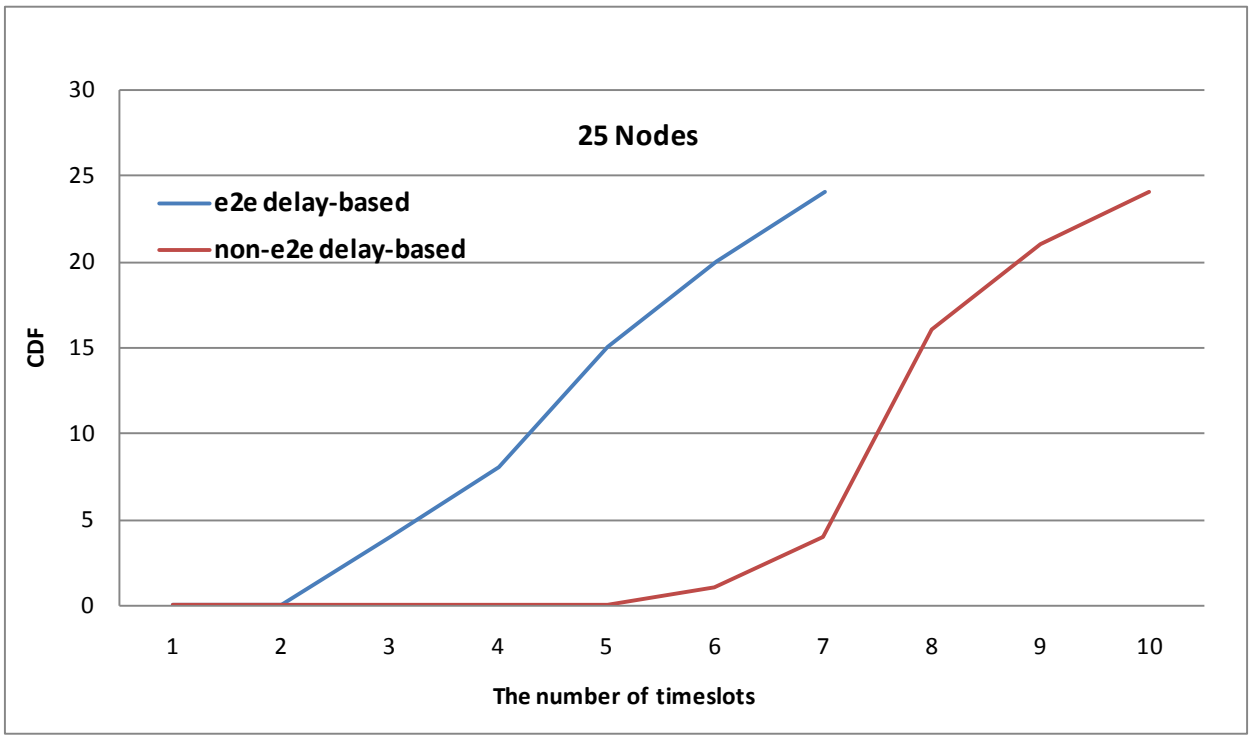


Figure 4.8 Cumulative Distributions of e2e delay-based and non-e2e delay-based for 25 nodes

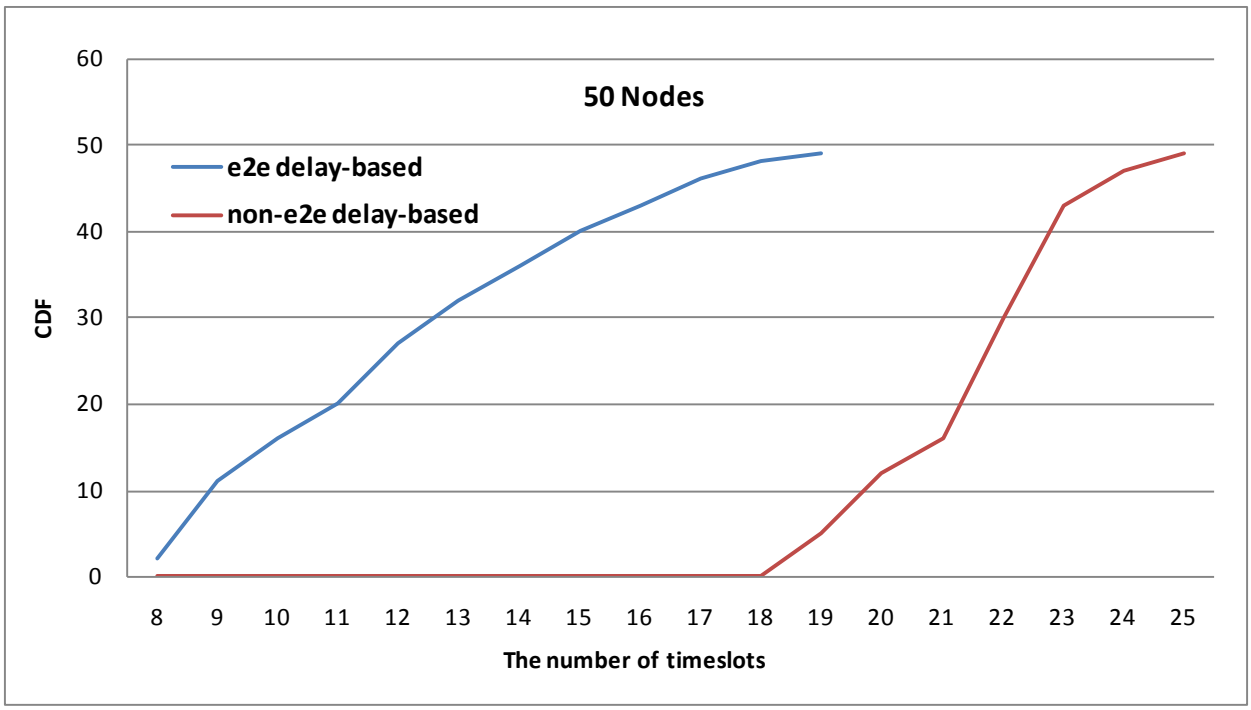


Figure 4.9 Cumulative Distributions of e2e delay-based and non-e2e delay-based for 50 nodes

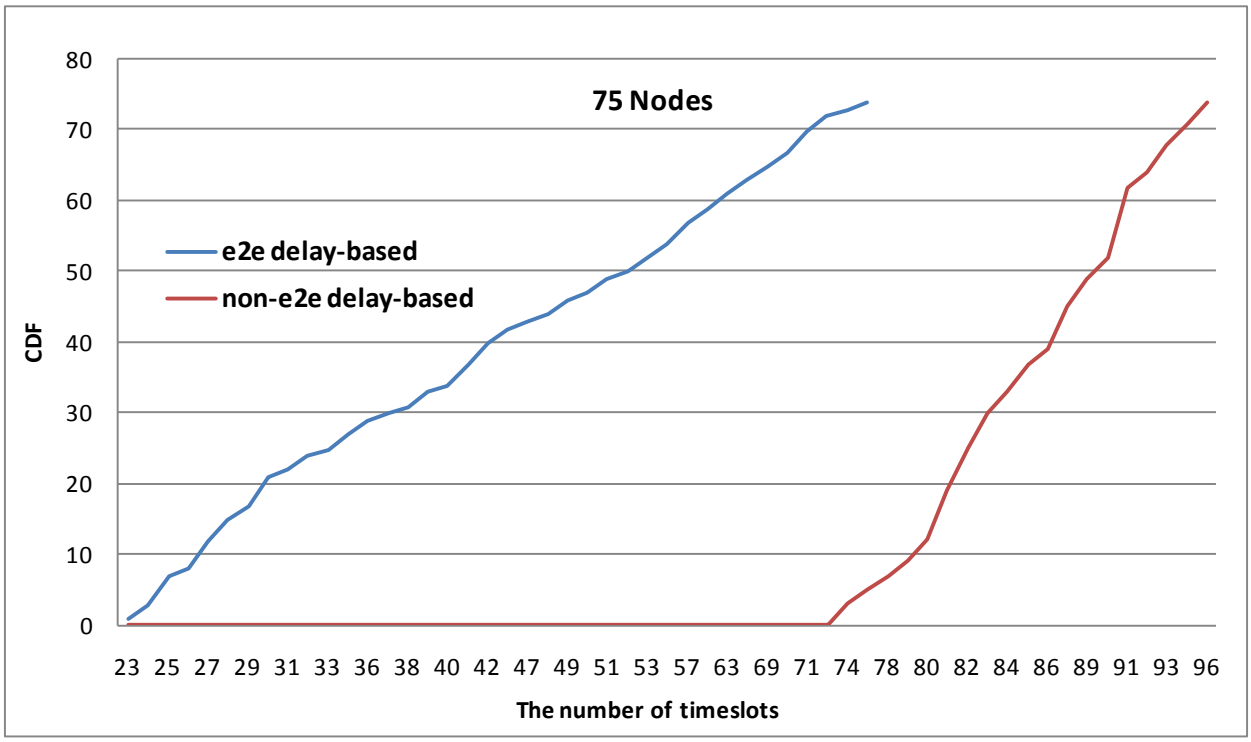


Figure 4.10 Cumulative Distributions of e2e delay-based and non-e2e delay-based for 75 nodes

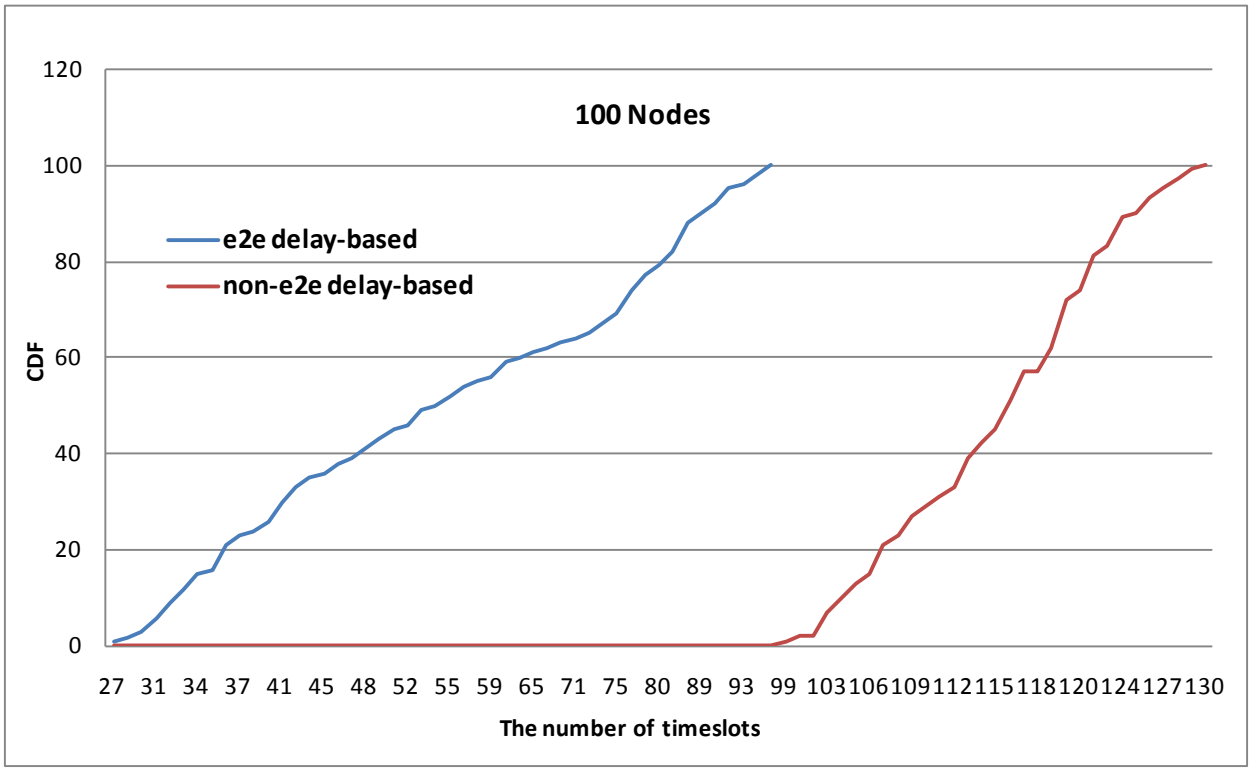


Figure 4.11 Cumulative Distributions of e2e delay-based and non-e2e delay-based for 100 nodes

In Fig. 4.8 and Fig. 4.11, we compare the e2e delay of our e2e delay-based algorithm to [7], which is non-e2e delay-based. We plotted the Cumulative Distribution with the maximum e2e delay for both algorithms among the 6 different topologies. The Cumulative Distributions show that our approach reduces the e2e delay in all network sizes. Although the reduction in the maximum e2e delay achieved by our approach is just 3 timeslots for 25-node network, the reduction in the e2e delay is more than 30 timeslots when the network size is increased to 100. Through simulation, we observe that although simultaneous transmissions reduce the number of timeslots by allowing other nodes to reuse the same timeslots, they also deteriorate the e2e delay because simultaneous transmissions prevent timeslots from being allocated in a staggered sequence.

4.5 Conclusion

In this article, we propose a noble scheduling algorithm using multi-channel and multi-timeslot with the objective of minimizing the end-to-end (e2e) delay in a tree-based wireless sensor network. The algorithm exploits a staggered timeslot allocation in terms of e2e paths, rather than the one with individual branches. It also increases simultaneous data transmissions. The proposed algorithm shows substantial improvement on the e2e delay when compared to current studies that focus on minimizing the number of timeslots, rather than the e2e delay.

Chapter 5

Conclusion and Future Work

This dissertation presents a new MAC protocol, namely the DSME, for industrial wireless sensor networks applications, including smart grid, telemedicine, process automation, and factory Automation. The DSME is a multi-hop, multi-channel, and multi-superframe-based wireless communication protocol for industrial applications with deterministic latency, high reliability, and flexibility requirements. DSME enhances the GTS mechanism of IEEE 802.15.4 to satisfy these requirements of the large scale industrial wireless sensor networks, and facilitates a multi-hop long range industrial network via multiple forwarder nodes that collectively form a multi-hop extention. Multi-hop extention realizes data delivery among the nodes located multiple hops away, which is not supported by IEEE 802.15.4.

In order to support guaranteed end-to-end delay, DSME adopts not only TDMA but also multiple channels over a TDMA timeslot. Multi-channel extention allows multiple simulaneous transmissions by adopting multiple channels in a single DSME-GTS slot, leading to substantially increased throughput and expedited simultaneous data delivery. Thus, DSME is quite appropriate for industrial applications that have large data traffic requiring deterministic and guaranteed end-to-end. Moreover, multi-superframe extention can overcome timeslot limitation of GTS in IEEE802.15.4. Seven GTS slots of IEEE805.15.4 are not enough for large-scale industrial networks, which mainly deal with periodic/deterministic time-sensitive data. Multi-superframe extension reiterates superframes with the configurable parameter MO. It flexibly controls the number of DSME-GTS slots based on the demands from the member nodes. With the extension

of multi-hop and multi-superframe, the utilization of multiple channels and TDMA timeslots would be desirable to increase the number of simultaneous reliable data transmissions by dynamically switching to the most reliable channel while keeping the same timeslot. However, scheduling multiple channels and TDMA timeslots for all nodes in a multi-hop industrial wireless sensor network is a significant challenge. DSME performs scheduling based on a first-come first-served basis by exchanging ABT table of each node using the three-way-handshaking (DSME-GTS request, DME-GTS reply, and DSME-GTS notify) in a distributed way.

In chapter 2, we study metaheuristics in combinatorial optimization for scheduling nodes, channels, and timeslots. The main purpose of metaheuristics is for optimal scheduling that minimizes the maximum end-to-end delay in multi-hop networks, such as wireless industrial sensor networks. TDMA access scheme in DSME provides deterministic end-to-end delay, but the delay may not be optimal since DSME schedules multiple timeslots and channels simply on a first-come, first-served basis whenever a node requests timeslot and channel. Thus, no optimality is considered in DSME. However, the scheduling for multi-hop networks is a combinatorial NP-complete problem, i.e., there are no specific solutions. Hence, we take a metaheuristic approach that provides a fast and efficient optimization tool for the complex combinatorial problem. SA is the most well-known representation of stochastic computation algorithms while PSO is one of the most well-known representations of the evolutionary computation algorithms. In SA, we investigated four frequently-used cooling schedules: exponential, Boltzmann, fast, and linear. Exponential, Boltzmann, fast, and linear schedules update the temperature vector by exponential decrease, logarithmic decrease, division, and subtraction of the current annealing parameter starting from the initial temperature, respectively. In PSO, if a particle is selected more often than the others, the likelihood of that particle being on the right track towards the global best

solution increases, so the cognitive coefficient increases, and the other particles would be more likely to follow that direction. In contrast, if a particle is selected less often than the others, the social coefficient increases because the solution quality of that particle is poor compared to those of the other particles, and that particle follows another direction. Consequently, each particle updates its velocity and position depending on the frequency of global best value. The directionality in the arrangement of timeslots to minimize the end-to-end delay is exploited to better fit the operations of PSO and SA. Through simulation, the e2e delay of the PSO-based scheduling is shown to be either half or one-fourth of the e2e delay achieved by the SA-based scheduling while requiring longer total computation time. In terms of the solution quality and the computation time, there is a tradeoff between PSO-based scheduling and SA-based scheduling. We further studied distributed metaheurisitcs because in ad-hoc applications, such as disaster areas, battlefields, and spontaneous networks, distirbuted metaheuristics provide improved scalability, reduced complexity, and flexibility for a mesh-based peer-to-peer network environment, where no centralized coordinator exists. Other metaheuristics, such as Ant Colony Optimization (ACO) and Harmony Search (HS), will be investigated in future studies.

In chapter 3, we propose a greedy heuristic scheduling algorithm for channel and TDMA timeslot allocations to minimize guaranteed e2e delay while studying two separate scheduling problems: scheduling multiple channels at the receiver side and multiple timeslots at the sender side. First, we determine all potential receivers, which are parent nodes in a tree topology because all data flow occurs from the bottom to the top. Each potential receiver calculates all possible interferences from any child to any parent, and the receiver constructs a list of interfering receivers. If there are enough available channels at the receivers, all of its interferences could be eliminated simply by allocating different channels to a pair of interfering

receivers. However, if there are not enough available channels, timeslot allocation computation is performed at the sender side based on the list of interfering receivers. Once a channel allocation algorithm has been performed at the receiver side, timeslot allocation algorithm is performed to allocate timeslots at the sender side in terms of paths in a staggered way.

Other studies have shown that increasing the number of parallel transmissions by utilizing multiple channels do not always result in a small schedule length [41], and that the structure of the routing tree affects not only the schedule length [7] but also the e2e delay. In [7], the routing tree is constructed by balancing the number of nodes on branches in order to minimize the number of required *timeslots*. However, in a future study, we will propose to construct the routing tree by balancing the number of nodes on paths rather than branches in order to reduce the *e2e delay* even further than the reduction achieved by the heuristic algorithm alone. The balanced tree construction is defined as Capacitated Minimal Spanning Tree Problem and is proved to be an NP-complete [42]. [43] uses a greedy heuristic in solving the balanced tree construction problem by employing a cost function according to the load that a node may bring to a branch. However, the authors do not consider the growth possibilities of the branch and the node. The growth possibility is a measure of the likelihood of increasing the total number of branches in a tree topology [44], and such information is important in determining the final structure of a tree network. We may propose a heuristic algorithm that considers such growth possibilities, and breaks down the tree construction into two parts. First, every node collects information about the potential paths that it can connect to by starting from the sink, and all the nodes propagate information about their hop-count to the sink and the potential path ID's. In the second part, the tree is constructed with a growth set of path including the unconnected neighbors of the nods that are already on the path. Weight of a path is the number of nodes that

are already connected to the path.

References

[1] Wireless HART, http://www.hartcomm.org

[2] BACnet, http://www.bacnet.org

[3] ISA, http://www.isa.org

[4] IEEE Std. 802.15.4-2006, “Part 15.4: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)”, 2006.

[5] IEEE P802.15.4e-2012, “Part 15.4: Low-Rate Wireless Personal Area Networks (WPANs), Amendment 1: MAC sub-layer”, Feb. 2012.

[6] Garey, M.R, and Johnson, D.S. "Computers and Intractability: A Guide to the Theory of NP-Completeness." New York: W.H.Freeman, 1979.

[7] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, “Multi-Channel Scheduling for Fast Convergecast in Wireless Sensor Networks,” USC CENG Technical Report CENG-2008-9.

[8] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, “Fast Data Collection in Tree-Based Wireless Sensor Networks,” IEEE Transactions on Mobile Computing, vol. 11, no. 1, Jan., 2012.

[9] Doherty, L.; Lindsay, W.; Simon, J. “Channel-Specific Wireless Sensor Network Path Data”, ICCCN 2007. Proceedings of 16th International Conference Aug. 2007.

[10] W. Jeong, and J. Lee, “Performance evaluation of IEEE 802.15.4e DSME MAC Protocol for Wireless Sensor Networks”, The first IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012.

[11] J. So and N. Vaidya, “Multi-Channel MAC for Ad-Hoc Networks: Handling Multi-Channel Hidden Terminal Using A Single Transceiver,” in ACM MobiHoc 2004, May 2004.

[12] O. DurmazIncel, P. G. Jansen and S. J. Mullender, “MC-LMAC: A Multi-Channel MAC Protocol for Wireless Sensor Networks,” under review for the Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2009.

[13] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. F. Abdelzaher, “MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks,” in Proceedings of IEEE Infocom, 2006.

[14] Y. Wu, J. Stankovic, T. He, and S. Lin, “Realistic and efficient multi-channel commnications in wireless sensor networks,” in Proceedings of IEEE INFOCOM 2008, 2008, pp. 1193-1201.

[15] S.-L. Wu, C.-Y. Liu, Y.-C. Tseng, and J.-P. Shen, “ A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks,” in I-SPAN, 2000.

[16] J. Zhang, G. Zhou, C. Huang, S. Son, and J. A. Stankovic, “TMMAC: An Energy Efficient Multi-Channel MAC Protocol for Ad Hoc Networks,” in IEEE ICC, 2007.

[17] J. So and N. Vaidya, “Multi-Channel MAC for Ad-Hoc Networks: Handling Multi-Channel Hidden Terminal Using A Single Transceiver,” in ACM MobiHoc 2004, May 2004.

[18] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. F. Abdelzaher, “MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks,” in Proceedings of IEEE Infocom, 2006.

[19] O. DurmazIncel, P. G. Jansen and S. J. Mullender, “MC-LMAC: A Multi-Channel MAC Protocol for Wireless Sensor Networks,” under review for the Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2009.

[20] J. Polastre, J. Hill, and D. Culler, “Versatile Low Power Media Access for Wireless Sensor Networks,” in SenSys04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems. New York, NY, USA: ACM, 2004, pp.95-107.

[21] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. F. Abdelzaher, “MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks,” in Proceedings of IEEE Infocom, 2006.

[22] Y. Wu, J. Stankovic, T. He, and S. Lin, “Realistic and efficient multi-channel commnications in wireless sensor networks,” in Proceedings of IEEE INFOCOM 2008, 2008, pp. 1193-1201.

[23] HolgerKarl, and Andreas Willig. “Protocols and Architectures for Wireless Sensor Networks”, WILEY, 2005.

[24] J.-Y. Le Boudec and P. Thiran, “Network Calculus: A Theory of Deterministic Queuing Systems for the Internet”, Springer, Lecture Note CS, Vol. 2050, 2001.

[25] R. L. Cruz, “A Calculus for Network Delay, Part I: Network Elements in Isolation, Part II: Network Analysis”, IEEE Transactions on Information Theory, Vol. 37, No. 1, January 1991.

[26] Gang Lu, BhaskarKrishnamachari, Cauligi S. Raghavendra, “An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks”, Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International

[27] Chlamtac and S. Kutten, “Tree-Based Broadcasting in Multihop Radio Networks,” IEEE Trans. Comp., vol. 36, no. 10, 1987, pp. 1209-23.

[28] M. Zorzi, R. R. Rao, and L. B. Milstein, “On the accuracy of a first order Markov Model for data transmission on fading channels”, Proc. IEEE ICUPC’95, pp. 211-215, November 1995.

[29] Kirkpatrick, S., Gelatt, C.D. and Vecchi, P. M., “Optimization by Simulated Annealing,” Science, vol. 220, pp. 610-680, 1983.

[30] J. Kennedy, R. Eberhart, “Particle Swarm Optimization,” Proc. IEEE Int. Conf. Neural Networks, 1995, vol. 4, pp. 1942 - 1948.

[31] J. Kennedy, R. Eberhart, “A discrete binary version of the particle swarm algorithm,” Proc. IEEE Int. Conf Systems, Man, Cybernetics, Oct. 1997, vol. 5, pp. 4104 - 4108.

[32] T. Wang, Z. Wu, and J. Mao, “PSO-based Hybrid Algorithm for Multi-objective TDMA Scheduling in Wireless Sensor Networks,” in Second international Conference on Communication and Networking, 2007, pp. 850 – 854.

[33] J. Wang, H. Choi, and C. Jung, “A Distributed Wireless Channel Assignment Algorithm with Collision Reduction,” in International Symposium on Collaborative Technologies and Systems, 2009, pp. 543 – 541.

[34] Liao C, Tseng C, and Luarn P. “A Discrete Version of Particle Swarm Optimization for Flowshop Scheduling Problems,” Computers and Operations Research 2007; 34: 3099-3111.

[35] SC Chu, YT Chen and JH Ho. "Timetable Scheduling using Particle Swarm Optimization," Innovative Computing, Information and Control, 2006. ICICIC'06. First INTERNATIONAL CONFERENCE.

[36] Duque-Anton, M., Kunz, D., and Ruber, B., "Channel assignment for cellular radio using simulated annealing." Vehicular Technology, IEEE Transactions Volume: 42 Issue:1 On page(s): 14 - 21 on Feb 1993.

[37] J. Chen, S. Olafsson, X. Gu, and Y. Yang, “A fast channel allocation scheme using simulated annealing in scalable wlans,” in Proc. of 5thInternational Conference on Broadband Communications, Networks and Systems, London, UK., September 2008, pp. 205–211.

[38] T. Toker, E. Altman, J. Galtier, C. Touati, I. Buret, B. Fabre, and C. Guiraud, "Slot Allocation In a TDMA Satellite System: Simulated Annealing Apporach." 21st International Communications Satellite Systems Conference and Exhibit, 2003.

[39] Anushiya A Kannan, Guoqiang Mao and BrankaVucetic, "Simulated Annealing based Wireless Sensor Network Localization." JOURNAL OF COMPUTERS, VOL. 1, NO. 2, MAY 2006.

[40] Frank Y. S. Lin and P. L. Chiu,"A Near-Optimal Sensor Placement Algorithm to Achieve Complete Coverage/Discrimination in Sensor Networks." IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 1, JANUARY 2005.

[41] H. Choi, J. Wang, and E. Hughes, “Scheduling for information gathering on sensor network,” Wireless Networks, 2007.

[42] C. H. Papadimitriou, “The complexity of the capacitated tree problem,” Networks, vol. 8, no. 3, pp. 217–230, 1978.

[43] L. R. Esau and K. C. Williams, “On teleprocessing system design part ii: A method for approximating the optimal network,” IBM Systems Journal, vol. 5, no. 3, pp. 142–147, 1966.

[44] H. Dai and R. Han, “A node-centric load balancing algorithm for wireless sensor networks,” in Proceedings of IEEE GLOBECOM ’03, vol. 1, Dec. 2003, pp. 548–552 Vol.1.