

DYNAMIC MODULATION SCALING ENABLED REAL TIME  
TRANSMISSION SCHEDULING FOR WIRELESS SENSOR NETWORKS

by

Arda Gumusalan  
A Dissertation  
Submitted to the  
Graduate Faculty  
of  
George Mason University  
In Partial fulfillment of  
The Requirements for the Degree  
of  
Doctor of Philosophy  
Computer Science

Committee:

\_\_\_\_\_ Dr. Robert Simon, Dissertation Director  
\_\_\_\_\_ Dr. Hakan Aydin, Committee Member  
\_\_\_\_\_ Dr. Parth Pathak, Committee Member  
\_\_\_\_\_ Dr. Jill Nelson, Committee Member  
\_\_\_\_\_ Dr. Sanjeev Setia, Department Chair  
\_\_\_\_\_ Dr. Kenneth Ball, Dean, The Volgenau School  
of Engineering

Date: \_\_\_\_\_ Spring Semester 2019  
George Mason University  
Fairfax, VA

DMS Enabled Real Time Transmission Scheduling for Wireless Sensor Networks

A dissertation submitted in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy at George Mason University

By

Arda Gumusalan  
Master of Science  
George Mason University, December 2017  
Bachelor of Science  
State University of New York, May 2012

Directors: Dr. Robert Simon, Professor, Dr. Hakan Aydin, Professor  
Department of Computer Science

Spring Semester 2019  
George Mason University  
Fairfax, VA

Copyright © 2019 by Arda Gumusalan  
All Rights Reserved

## Dedication

I dedicate this dissertation to my parents and my wife.

## Acknowledgments

Never thought the day to write these lines would have come. What can I say, it was a long journey... Long but fun with lots of ups and downs.

I would like to start with thanking to my advisors; Dr. Robert Simon, and Dr. Hakan Aydin for their guidance and patience during my stay at GMU. I consider myself to be fortunate for having them as my mentors. I will never forget their ability to calm me down whatever the reason was. Somehow they were always able to show me a way out during my quandaries.

Also my parents... They provided me endless support with everything. This will sound like a cliché but I couldn't have done this without them. Not just the PhD, everything that brought me up to this point. I felt their support everyday which gave me all the strength I needed and I will forever be grateful for that. I hope to be able to do the same for my kid(s) one day...

My wife Akemi... Thanks for being there for every critical moment of my PhD. You were probably the one who I reflected all the stress that I had and somehow you agreed to stick with me and even marry me. Muito obrigado.

Lastly but not the least, my friends who I have shared countless moments of joy and despair. I will never forget Sven Brehmer who I met the first day of my PhD and we were friends till the very end. Also Erhan Uyar who were with me the whole time. And of course I would like to acknowledge my lab mates who are either doctors and soon-to-be-doctors; Maryam Bandari, James Pope, Kevin Andrea, Abhishek Roy, Haoliang Wang, Ann Wang, Li Lui, Mohammed Hassan, Tim Balint (we accepted you as one of us), Mengbai Xiao, Ming Zhang, Mansour Abdulaziz and Yoon who joined us later. My colleagues; Raven Russell and Huangxin Wang... I will never forget our curious discussions almost about anything. I have learned a lot from each and every one of you. I hope our friendships last long after I leave GMU. I wish everyone of you best of luck in life.

# Table of Contents

	Page
List of Tables . . . . .	vii
List of Figures . . . . .	viii
Abstract . . . . .	xi
1 Introduction . . . . .	1
1.1 Road Map . . . . .	8
2 Background and Related Work . . . . .	9
2.1 Inter-node Power Management Techniques . . . . .	9
2.2 Node Level Power Management Techniques . . . . .	13
2.3 Summary . . . . .	19
3 Architectural Model . . . . .	20
3.1 Cluster Model . . . . .	21
3.1.1 Communication . . . . .	22
3.2 Multi-hop Model . . . . .	24
4 Minimizing Expected Energy Consumption of Time-Critical Cluster WSNs with Probabilistic Workloads . . . . .	27
4.1 Dynamic Superframe . . . . .	28
4.2 Joint Deadline-Energy Optimization problem . . . . .	34
5 Performance Evaluation . . . . .	39
5.1 Proposed Algorithms . . . . .	39
5.2 Simulation Settings . . . . .	42
5.3 Analysis of the Ideal Case . . . . .	45
5.4 Analysis of the effect of traditional LPL with no interference . . . . .	46
5.5 Analysis of the impact of neighborhood/interference . . . . .	49
5.6 Effect of different probability distribution functions . . . . .	51
5.7 Analysis of scalability . . . . .	52
5.8 Impact of Radio Hardware Variations . . . . .	52
5.9 Hardware Test Results . . . . .	54
5.10 Conclusions . . . . .	56

6	A Software Defined Radio Analysis of the Impact of Dynamic Modulation Scaling within Low Power Wireless Systems . . . . .	63
6.1	Experimental Methodology . . . . .	64
6.1.1	Experimental results . . . . .	69
7	DMS-enabled Multi-hop Topology Control for Time Critical WSNs . . . . .	84
7.1	Optimization Problem Formulation . . . . .	85
7.2	Polynomial Time Heuristics . . . . .	90
7.2.1	Finding the Minimum Delay Path . . . . .	90
7.2.2	Interference Aware Ordering . . . . .	90
7.2.3	Computing Minimum Delay . . . . .	91
7.2.4	Aggressive . . . . .	92
7.2.5	Explorer . . . . .	93
7.3	Performance Evaluation . . . . .	96
7.3.1	Network Topology Setup . . . . .	97
7.3.2	Results . . . . .	98
8	Compatibility of My Dissertation with the Existing Real-Time WSN Protocols .	107
8.1	Zigbee . . . . .	107
8.2	WirelessHART . . . . .	108
8.3	IEEE 802.15.4-2015 . . . . .	110
9	Conclusion . . . . .	114
	Bibliography . . . . .	117

## List of Tables

Table	Page
3.1 Scaling functions . . . . .	23
4.1 List of symbols . . . . .	35
5.1 Packet Loss Rate at Different Modulation Levels . . . . .	56
7.1 The variables used in the problem formulation and their description. . . . .	86

## List of Figures

Figure	Page
1.1 Slotframe structure. . . . .	4
1.2 Multiple simultaneous slotframes. . . . .	5
1.3 Low Latency Deterministic Network superframe. . . . .	6
3.1 The lay-out of superframe . . . . .	21
3.2 Delay bound . . . . .	25
3.3 An example network. Each link has at least a predetermined PDR rate. . .	26
4.1 Illustration of static and dynamic slack reallocation . . . . .	29
4.2 Parameters of the Sleep-Listen Cycle . . . . .	29
4.3 Possible intersection of listening period and preamble . . . . .	30
4.4 The steps of HLPL . . . . .	32
4.5 Comparison of traditional LPL and RLPL . . . . .	33
5.1 An example for dynamic algorithms . . . . .	41
5.2 Energy consumption in the ideal case. . . . .	46
5.3 Simulation results with no interference. . . . .	48
5.4 Impact of interference . . . . .	58
5.5 Energy consumption of smart-HLPL with different probability distribution functions . . . . .	59
5.6 Scalability in terms of number of nodes . . . . .	60
5.7 Scalability in terms of number of packets . . . . .	60
5.8 The effect of sleep power consumption. . . . .	61
5.9 Energy consumption with CC2420 based power consumption settings . . . .	62
6.1 Implementation of transmitter and receiver in GNURADIO . . . . .	65
6.2 Steps of signal decoding . . . . .	65
6.3 An example PSK signal recovery process . . . . .	66
6.4 Constellation display of received QPSK signal before and after recovery process.	67
6.5 Faraday cage inside look . . . . .	67
6.6 Faraday cage outside look . . . . .	68

6.7	SNR to PDR comparison of empirical measurements and theoretical calculations for MPSK . . . . .	70
6.8	SNR to PDR comparison of empirical measurements and theoretical calculations for MDPSK . . . . .	71
6.9	The test area that was used for distance measurements. Sender and receiver are 100 meters apart. . . . .	74
6.10	Packet delivery rates of different modulation levels using 2.4 GHz center frequency with 70dB output amplifier gain in terms of distance. . . . .	76
6.11	Packet delivery rates of different modulation levels using 915 MHz center frequency with 70 dB output amplifier gain in terms of distance. . . . .	76
6.12	Packet delivery rates of different modulation levels using 915 MHz center frequency with 60 dB output amplifier gain in terms of distance. . . . .	77
6.13	2DPSK packet delivery rate given a transmitter-receiver distance and an output power level. . . . .	78
6.14	4DPSK packet delivery rate given a transmitter-receiver distance and an output power level. . . . .	79
6.15	8DPSK packet delivery rate given a transmitter-receiver distance and an output power level. . . . .	79
6.16	Test area that was used for elevation tests. Receiver is placed on the ground and the sender is elevated. . . . .	81
6.17	The height of the sender and the distance of the sender from the receiver. . . . .	81
6.18	Packet delivery rates of difference modulation levels using 915 MHz center frequency with 60 dB output amplifier gain in terms of elevation difference. . . . .	82
6.19	Packet delivery rates of difference modulation levels using 2.484 GHz center frequency with 70 dB output amplifier gain in terms of elevation difference. . . . .	83
7.1	Optimization problem demonstration from a single node's perspective. . . . .	86
7.2	Hidden node scheduling problem. . . . .	89
7.3	Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes. . . . .	99
7.4	Comparison of optimal solution against heuristics when the base-station is placed at the <i>center</i> with 10 nodes. . . . .	102
7.5	Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes. . . . .	103

7.6	Comparison of optimal solution against heuristics when every modulation level has the same interference range of modulation level 1. Area size is 500x500 m <sup>2</sup> and network size is 10 nodes plus the coordinator placed at (0,0).	105
7.7	Scalability analysis in terms of number of nodes in a 500x500 m <sup>2</sup> area and the coordinator is placed at the coordinates (0,0) with a deadline factor of 0.5.	105
8.1	Zigbee Network . . . . .	108
8.2	WirelessHART Network . . . . .	109
8.3	IEEE 802.15.4-2015 extended superframe structure as taken from the official standard description. . . . .	111
8.4	How the Optimal Solution (see Section 7.1) might look like. . . . .	112
8.5	After the Optimal Solution has edited to comply with IEEE 802.15.4-2015.	112

## Abstract

DMS ENABLED REAL TIME TRANSMISSION SCHEDULING FOR WIRELESS SENSOR NETWORKS

Arda Gumusalan, PhD

George Mason University, 2019

Dissertation Director: Dr. Robert Simon

Dissertation co-Director: Dr. Hakan Aydin

Real-time low-power wireless monitoring are increasingly being used in applications such as: Industrial Internet-of-Things, Smart City technologies, and critical infrastructure monitoring. Creating a deadline driven scheduling while considering energy management creates complex optimization problems. My research integrates energy saving mechanisms with real-time scheduling for time critical WSNs to save energy while meeting the desired quality of service requirements. I investigate and improve the energy consumption of real-time wireless sensor network (WSN) protocols utilized in industrial control systems. My contributions are in three distinct areas.

First, I focus on single cluster real-time WSNs specifically on improving time-slotted superframe based techniques. The current wireless standards for Industrial Control Networks (ICNs) employ static slots for their superframe structure. I study the concept of dynamic readjustment of time-slots to minimize the overall energy consumption and combine real-time performance with novel energy conservation methods by describing a set of dynamic modulation scaling (DMS) based adaptive packet transmission scheduling algorithms that reclaim unused slot times.

To support my reclaiming method in a wireless environment, I introduce a novel low-power listening technique called Hybrid Low-Power Listening (HLPL) protocol. I evaluate my algorithms using Castalia simulator against an oracle-based approach, and show that my dynamic slot reclaiming approach, coupled with HLPL, can introduce substantial power savings without sacrificing real-time support. In order to further expand applicability scope of my solution, my dissertation work considers non-deterministic workloads. Next, I take a deeper look into DMS and its effect on low power wireless communication. In recent years a number of studies have suggested that DMS techniques can reduce energy consumption in low-power wireless transmission technologies. These studies tend to rely on theoretical or simulation DMS models to predict network performance metrics. However, there is little, if any, work that is based upon empirically verified network performance outcomes using DMS. My dissertation fills that gap. First, by using GNU Radio and SDR hardware I show how to emulate DMS in low power wireless systems. Second, I measure the impact of varying Signal-to-Noise levels on throughput and delivery rates for different DMS control strategies. Third, I quantify the impact of distance by using DMS and finally, I measure the impact of different elevations between sender and receiver on network performance. My results provide an empirical basis for future work in this area.

For the third part, I investigate transmission scheduling of multi-hop time critical WSNs. Previous work has shown that connection driven topology control has tremendous potential to decrease energy consumption and/or latency [1–5]. DMS changes transmission energy levels and has a direct impact on packet loss rate and propagation distance. However, current work does not provide any multi-cluster communication solutions which incorporate DMS into already managed transmission energy level control. I address this gap by first formulating Mixed Integer Nonlinear Formulation (MINLP) of DMS enabled transmission scheduling for deadline driven networks. Next, I present two polynomial time heuristics. I compare them against the optimal solution by integrating the empirical measurements obtained from my SDR tests and present how DMS can be applied to multi-hop WSNs to save substantial amount of energy.

## Chapter 1: Introduction

Wireless Sensor Network (WSN) is the network of autonomous sensor devices which are distributed among a target area in order to sense and monitor physical and environmental conditions. These devices, also called nodes, are typically capable of sensing sound, temperature, humidity, and pressure. It is also possible to extend their capabilities to be tailored for specific tasks. As a result, WSNs have vast application areas such as structural health monitoring, intrusion detection, highway monitoring, smart grid monitoring, and power delivery systems monitoring [6, 7]. These nodes are typically expected to operate for a long duration of time on AA batteries and hence have limited processing power and memory.

Industrial Control Networks are among the many application areas of Wireless Sensor Networks. Industrial network automation systems were traditionally installed with wires connecting communicating devices. Potential drawbacks to purely wired systems are higher costs for cables and maintenance and inflexibility in terms of deploying new nodes or reconfiguring existing systems. As a result industrial automation and control systems are increasingly being supported by wireless networks [8]. Wireless industrial systems are now appearing in application domains such as manufacturing, electrical generation, and chemical refining [9], along with Smart City and environmental monitoring applications [10]. Currently deployed industrial wireless protocols includes IEEE 802.15.4e, WIA-PA, WirelessHART and ISA100.11a [11]. WirelessHART has been deployed by more than 24 thousand networks with at least 5 billion operating hours according to their website.

Low-power real-time wireless protocols typically work by organizing nodes in cluster or star topologies, and sometimes in multi-hop topologies. Variations of time division multiple access (TDMA) based scheduling for link access is the most widely used method to provide real-time guarantees on WSN [12]. TDMA systems generally have a coordinator that is

in charge of distributing time slots to the nodes. Nodes in the system therefore share a logical superframe that is divided into timeslots. Each node has a pre-assigned time slot where it is allowed to transmit so that collisions are prevented. The work presented in [12] concludes that TDMA improves the performance of basic CSMA/CA protocols. Most current standards either use a fixed size or varying size but pre-computed slot lengths for their superframes. This may lead to an inefficient or inflexible use of resources in the form of unused timeslots, especially if the workload is not fully predictable.

First part of my dissertation studies dynamically readjusting time slot lengths in a superframe as a method to reduce overall energy cost and provide tighter real-time guarantees to address these issues. As noted, many existing protocols assume that the workload is fully deterministic and known in advance, which is not the case for many newer applications that can be supported by real-time wireless protocols [13]. My dissertation addresses these limitations through the design and analysis of adaptive, superframe based techniques designed to maintain real-time performance guarantees while minimizing *expected* energy consumption. In order to accomplish this goal, I adopted a well-known and widely-studied technique called Dynamic Modulation Scaling (DMS) [14, 15], also known as Adaptive Modulation. DMS is a technique that exploits the trade off between time and energy consumption of a given modulation level. Higher modulation levels consumes more energy to transmit and receive the data but the transmission takes less time. It has been showed that DMS technique leads to reduced energy consumptions [15]. It is also commonly used to increase throughput in hostile and time varying wireless situations [16]. For instance, in tactical military environments one current mobile handheld standard is called JTRS. Mobile handheld radios such as the Harris AN/PRC-15 implement adaptive modulation within the JTRS standard. For Low-Power and Lossy Networks, the TI CC1200 supports 2-FSK, 2-GFSK, 4-FSK, 4-GFSK, MSK, OOK modulation whereas CC2500 supports 2-FSK, 2-GFSK, MSK and OO. The application of DMS within WSNs has been studied in [17–20].

The underlying idea of the first part of my dissertation is to assign each node a time slot in order to meet their communication transmission deadlines, but allow them to *proactively*

wake up and determine if other nodes have transmitted all of their packets and no longer require some portion of their time slots. If this is the case, a node can begin packet transmission before its scheduled time, and conserve energy by transmitting at reduced modulation levels. In order to accomplish this, I designed a new low-power listening protocol called *Hybrid Low-Power Listening*. The algorithmic and protocol challenge is to schedule packet transmissions in a manner that reduces energy while maintaining real-time performance, as compared to the traditional static TDMA approach.

Next, I investigate the application of DMS on time-critical multi-hop low-power networks. Multi-cluster star-topologies are among the most common deployment topologies by ICNs for data aggregation. It has been shown that topology control has tremendous potential to decrease energy consumption and/or latency. DMS changes transmission energy levels and has a direct impact on packet loss rate and propagation distance. However, current work in literature does not provide any multi-cluster communication solutions which incorporate DMS into already managed transmission energy level control. To address this, my dissertation invests the affect and applicability of DMS on multi-cluster/multi-hop networks. It has been shown that there exists a threshold distance  $d$  where DMS does not function as expected for the lower values of  $d$  [21, 22]. This threshold value significantly depends on hardware design specifically the underlying sensor node's circuitry energy consumption. However, this is as far as it goes in terms of understanding the effects of DMS on transceiver distances. An empirical measurement on this subject as well as DMS based energy, packet loss, distance aware routing protocol *was* yet to be developed.

Next in my dissertation, I look into incorporating DMS into multi-hop networks. The first part of my work has focused on the superframe structure described in 802.15.4-2006 but a more recent amendment (802.15.4e-2012) has extended this structure tailored for multi-hop network. The challenge comes from both interference aware scheduling as well as incorporating DMS into this structure. Interference aware multi-hop scheduling has been studied in literature and proven that a throughput optimal TDMA schedule is an NP-complete problem even for linear topologies [23] and [21] proves that interference and

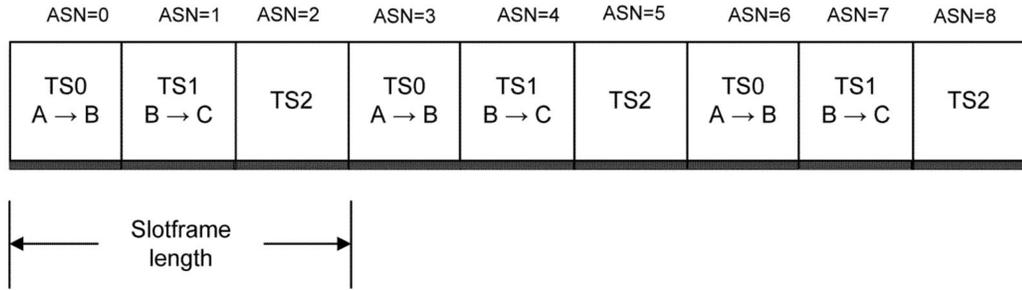


Figure 1.1: Slotframe structure.

energy aware speed assignment problem is NP-hard.

802.15.4e-2012 extend 802.15.4-2006 by including bi-directional communication and channel-hopping. All hopping sequences are identified by an id. Each id refers to pseudo-randomly shuffled list of all of the channels. This amendment also introduces *slotframe* concept as depicted in Figure 1.1. Slotframe timeslots can be assigned to a specific pair of nodes to communicate, unlike superframe which assigns time slots between nodes and the coordinator. Usually these slots are selected to be long enough to exchange a data packet and an ACK packet but allowed to be of shorter length. Timeslots, slotframe length and which frequency to use is assigned by a higher layer. In Figure 1.1 *Absolute Slot Number (ASN)* 0 is assigned for node B to receive data from node A. It is possible to have multiple simultaneous slotframes (Figure 1.2) which may share the same nodes. The interference can be handled in either the time or the frequency domain. My dissertation handles interference in the time domain.

A new *Low Latency Deterministic Networks (LLDN)* superframe structure is also introduced as shown in Figure 1.3. This new structure adds management time slots, directional (uplink, downlink, bidirectional) time slots, retransmission time slots.

The new LLDN superframe combined with slotframe structure aims to extend the 802.15.4-2006 superframe structure for multi-hop/multi-cluster communication. The frequency hopping and division needs to be adjusted such that interference and multi-hop fading will be reduced. Uplink time slots allows a specific cluster to form a group and

	ASN=0	ASN=1	ASN=2	ASN=3	ASN=4	ASN=5	ASN=6	ASN=7	
Slotframe 1 5 slots	TS 0	TS 1	TS 2	TS 3	TS 4	TS 0	TS 1	TS 2	...
Slotframe 2 3 slots	TS 0	TS 1	TS 2	TS 0	TS 1	TS 2	TS 0	TS 1	...

Figure 1.2: Multiple simultaneous slotframes.

transmit back to back followed by a *group acknowledgement* (GACK) where a single bit is used to indicate the success of a group member’s transmission. Management time slots are used to collect health reports from the network as well as to impose new rules on the network by the network manager. The primary purpose of superframe is clock synchronization among the nodes. It also contains information about the current operating mode and the ACKs from previous superframe. Similar properties also exist in WirelessHART standard. In addition to backup slots and frequency hopping, WirelessHART requires at least one alternative routing path per node which may be utilized in case the a transmission fails in primary and backup time slots. Time synchronization mechanisms of WirelessHART and 802.15.4e-2012 are extremely similar where each time slot has predefined TX and RX offset values and the gap between expected arrival time of a packet and the actual arrival time is used to synchronize sender and receiver clocks. My dissertation assumes the existence of this time synchronization mechanism.

My dissertation integrates DMS into these real-time WSN protocols. Not only DMS aware multi-hop routing is understudied, to my best knowledge, there does not exist any protocol which considers the effect of the network topology and channel quality on the applicability of DMS for multi-hop time critical low power networks. It is known that, the effectiveness of DMS depends on the communication distance [21, 22]. However, there does not exist a work that takes effective modulation level ranges into account, possibly in a network initialization phase, based on the communication distances. Moreover, the effect

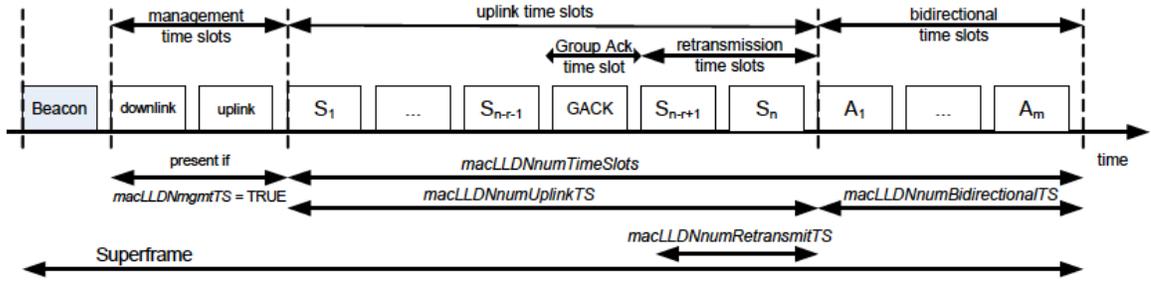


Figure 1.3: Low Latency Deterministic Network superframe.

of DMS on wireless interference is also not studied. As it was discussed previously, the key components of a real-time WSN protocol is the time division and DMS will have an impact on it. Increasing the transmission output power in order to use a higher modulation level may increase the interference range of nodes and hence effect their co-schedulability.

The second part of my dissertation investigates these issues by conducting tests on Software Defined Radios (SDRs). There does not exist a sensor node which has advanced DMS capability. My goal is to fill the gap in the empirical analysis of DMS when applied in the WSNs. In this domain, IEEE 802.15.4 is one of the standards that define physical and data link layers for low-power networks. It is widely adopted by Wireless Sensor Networks (WSNs). Despite its inherent potential for energy savings and hence increased sustainability, DMS is not part of IEEE 802.15.4 and there does not exist any off-the-shelf commercially available IEEE 802.15.4 compliant DMS-capable radio. This lack of available hardware creates a bottleneck for moving forward with the applicability of DMS in low-power networks. To solve this problem I have programmed Ettus B210 Software Defined Radios according to IEEE 802.15.4 base-band, symboling rate, and samples per second settings. I have obtained empirical results by varying Signal-to-Noise-Ratios (SNRs) for  $\{2, 4, 8, 16\}$  PSK and DPSK modulations and observed how the packet delivery rates (PDRs) are affected inside a noise controlled environment. I have compared my observation with existing general mathematical models. Next, I conducted various distance tests to get the empirical measurements of how propagation distance affects PDR for DMS.

Third and the last part of my dissertation consists of applying DMS to multi-hop time-critical WSNs by using the empirical measurements from the second part of my dissertation. Until now, no empirical measurements were available to understand how DMS would affect parent node selection process for real-time multi-hop WSNs. This part of my dissertation first presents the Mixed Integer Nonlinear Programming (MINLP) formulation of DMS enabled parent node selection process. The goal of this MINLP formulation is to minimize the network's overall energy consumption while satisfying a given deadline constraint. Later, I explain how this MINLP formulation can be converted into Mixed Integer Linear Programming (MILP). Generally speaking, MILP problems are known to be intractable. Hence for the next step, I have developed two polynomial time heuristics and compared them against the optimal solution.

My dissertation makes the following contributions. I first build a basic real-time superframe model, and then use DMS to opportunistically save energy. I formulate the joint real-time and energy minimization as an optimization problem under the assumption of a workload that can only be known probabilistically. Since solving the optimization problem may be computationally complex, I then propose a set of polynomial-time algorithms to address the joint real-time energy optimization problem. To avoid excessive energy expenditures during the times nodes proactively wake up to see if they can prematurely transmit, I propose a novel Hybrid Low-Power Listening (HLPL) protocol. HLPL incorporates a new technique called *reverse-low-power listening* (RLPL). RLPL is a twist on traditional low-power listening (LPL) protocols, which are well-known methods used in low data rate duty cycling wireless sensor networks [24]. LPL protocols yield significant energy savings. To my best knowledge this is the first usage of a hybrid LPL technique in a joint TDMA-based real-time energy savings protocol, as well as the first one that combines the beacon-enabled superframe concept with low power listening. Next, I show how a signal recovery process and DMS can be implemented on SDRs. I then show how PDR is affected by various environmental conditions such as different SNRs, transceiver distances and elevation differences. I provide a detailed insight into how output power can be managed to achieve desired

PDRs for low-power WSNs. This is the first work that has been done to provide empirical measurements on the effect of DMS on WSNs. This is particularly important because the applicability of these measurements is not for only this dissertation but for the future researches on this domain as well. Lastly, I present the energy-optimal MINLP formulation of DMS enabled parent node selection and propose two polynomial time heuristics. This work is especially novel for its extended DMS model. The work in literature typically assumes a discrete power level for each modulation level. My dissertation extends this model and shows how power levels can be selected from a range for each modulation level according to the given environmental conditions. My MINLP formulation computes the time-slot assignment of the nodes and assigns a modulation level to use while the interference is handled in the time-domain. Moreover, this interference aware scheduling takes the different interference ranges and packet delivery rates per modulation level into account. Due to the NP-hard nature of the problem [21], I propose two polynomial time heuristics and compare them against the optimal solution in various network setups.

## 1.1 Road Map

This dissertation is organized as follows; in Chapter 2, I review current energy management techniques widely used WSN domain in terms of communication and computation. Moreover, I divide them into node level and inter-node level techniques. In Chapter 3, the architectural model is explained, Section 3.1 of this chapter explains the architectural of my single cluster model and Section 3.2 extends it to multi-hop networks. I present my dynamic superframe concept coupled with my novel HLPL protocol for single hop cluster networks in Section 4. Chapter 6 presents my empirical measurements of packet-delivery-rates of different modulation levels in terms of various noise levels, elevation differences and transceiver distances. In Chapter 7, I show how DMS can be applied to multi-hop real-time WSNs by first presenting the optimal solution formulation and two polynomial time heuristics and analyzing their performance by incorporating the data obtained from Chapter 6. Finally, I conclude my dissertation in Chapter 9.

## Chapter 2: Background and Related Work

This chapter reviews the literature and provides the background information concerning my research. Particularly, gives a comprehensive summary of MAC layer protocols and energy management techniques widely used in WSN domain. I divide them into two categories namely inter-node and node-level techniques to be discussed in Section 2.1 and Section 2.2 respectively.

### 2.1 Inter-node Power Management Techniques

My primary focus is industrial control protocols which generally incorporate time division multiple access (TDMA) mechanisms. A beacon-enabled superframe technique is presented as an amendment to IEEE 802.15.4 standards and is included in 802.15.4e [25]. Aimed at supporting real-time industrial systems, 802.15.4e provides real-time guarantees for wireless sensor networks (WSNs). This standard defines the contention-access-period, contention-free-period and guaranteed-time-slot formats. This basic approach is incorporated in industrial standards such as WirelessHART, ISA 100.11a and WIA-PA [11, 26].

WirelessHART made TDMA based scheduling an industrial standard using a formulation similar to the approach presented in [25]. It defines a superframe as the organization of fixed size time slots controlled by the network manager [26]. In order to avoid interference, it uses a frequency hopping spread spectrum mechanism across the 16 channels of 2.4 GHz ISM band [27]. ISA100.11a is another industrial standard for wireless networks. It also has a similar superframe concept as WirelessHART [11]. WIA-PA is a widely adopted standard for system architecture and communication protocols for wireless networks. WIA-PA uses the beacon enabled superframe design as introduced in 802.15.4e [11]. In this dissertation,

I use a generic beacon enabled superframe model that can be applied to any of these industrial standards. These standards emphasize supporting real-time performance in wireless networks. My work enhances these standards by adopting dynamic time slot allocation and on-the-fly adjustment for the generic superframe structure. Moreover, I propose a novel protocol called Hybrid Low-Power Listening (HLPL) as an efficient technique to eliminate the impact of neighborhood for superframe structures by combining two seemingly contradicting ideas such as superframe and low-power listening.

My work incorporates TDMA, CSMA, and Low Power Listening (LPL) MAC layer protocols; hence, it can be categorized as a hybrid MAC layer protocol. Hybrid MAC-layer approaches have been studied for a number of years [28]. Some well-known examples are Z-MAC, HyMAC, H-MAC, ER-MAC, and Queue-MAC [29–33]. Z-MAC [29] uses a randomized two-hop setup mechanism to avoid collisions where each node is assigned a time slot. In the case of no transmission after a predefined time interval, the nodes can start communicating using CSMA for the duration of the time slot. A hybrid bandwidth aware mechanism is presented in [34]. The work in [35] used a Markov Decision Process in a hybrid environment to resolve congestion issues and minimize energy consumption, while other authors consider approaches for reducing queue length [36]. HyMAC combined TDMA and FDMA where a base station assigns time slots and frequencies [30]. H-MAC is another hybrid MAC protocol that combines CSMA with the Aloha protocol [31]. ER-MAC, on the other hand, [32] reduces the energy consumption of Z-MAC by allowing CSMA in only *emergency* situations. Queue-MAC [33] also combines TDMA with CSMA in conjunction with variable slot length. In Queue-MAC, each node transmits its load to its parent which in return adjusts the slot distribution in advance. One possible weakness in this protocol is that it assumes the load of the nodes are known prior to their transmission, which is not practical in many real life scenarios [13].

A more recent hybrid MAC protocol called MMSMAC is proposed in [37]. This protocol operates in synchronous, asynchronous, and hybrid modes. MMSMAC groups the nodes into clusters according to their per-hop distances to the cluster head. In synchronous

mode, the nodes are grouped into odd or even based on their cluster numbers and work in periodic active and sleep cycles. Only the nodes in active state are allowed to receive or transmit data (one node can transmit during an active state per cluster). This mode reduces energy consumption but increases delay. In asynchronous mode, the nodes compete for the channel which reduces delay but increases energy consumption. In hybrid mode, the nodes are set up just as synchronous mode but sensor nodes of the active cluster follow the asynchronous operation mode. The hybrid mode's performance is between asynchronous and synchronous modes. Another recent hybrid MAC protocol is TAHMAC [38] which combines CSMA/CA, TDMA, and FDMA. This protocol uses CSMA/CA for lower traffic levels and TDMA/FDMA for high traffic and only TDMA for medium traffic. An adaptive TDMA scheduling for multi-cluster networks is proposed in [39]. This system divides time slots into three categories — IntraSend, InterComm, and IntraRecv — and requires each node to know its interference information and workload. The adjustment of time-slots are done accordingly and are static during the superframe interval. [40] also proposed a distributed slot scheduling algorithm for hybrid CSMA-TDMA MAC layer. In this protocol, each node randomly selects a slot from the available slots list then broadcasts its request to own it. If no objection arises from a neighboring node, the slot is assigned. CSMA comes into play during slot re-allotment process where the system tries to reduce the number of allocated time slots. This re-allotment process may cause non-slot-owner nodes to collide and hence, CSMA is used by non-owners for channel assessment.

My dissertation differs from the above through my introduction of the Hybrid Low-Power Listening (HLPL) concept that allows efficient on-the-fly slot adjustments, even in the presence of interference where the nodes may overhear each others' communication, and my use of Dynamic Modulation Scaling (DMS) to save energy while maintaining transmission deadlines.

Low-power listening (LPL) is a commonly used MAC-layer protocol that reduces the energy consumption caused by idle listening to the channel for an activity. In LPL, nodes periodically wake up to detect the activities in the channel. LPL techniques are generally

categorized as either *sender-initiated*, *receiver-initiated*, or *hybrid*. Another classification divides LPL protocols into *synchronous* or *asynchronous* solutions. The HLPL protocol that I designed is a sender-initiated, asynchronous LPL and includes a new technique to address high false-alert rates caused by overhearing observed traffic in the traditional LPL protocol frameworks [41].

The current wisdom in these scenarios is to use either TDMA to give QoS guarantees or LPL otherwise [42]. To my best knowledge, my work is the first to show that these can be combined in order to give QoS guarantees such as real-time performance and energy saving.

Other energy conservation schemas include topology control, data management, and mobility [1]. Topology control is a term used for both hierarchy imposing techniques and for finding the optimal subset of the nodes which will ensure the connectivity. My focus will be for the latter. Location-driven topology control protocols divides the network into grids where every node within a grid can communicate with the neighboring grids. Then, a subset of the nodes in a grid route the packets and the remaining nodes go to idle state. Data-reduction approaches aims to reduce the amount of data being transmitted in a network. These include data aggregation, data compression, and data prediction. In my research data-prediction is of interest. In the first part of my work, I assume probabilistic workloads which is a stochastic approach to data prediction. This typically involves training of probability density function [43]. Other data driven approaches include adaptive-sampling, hierarchical sampling, and model-based active sampling for efficient data acquisition process. Adaptive sampling aims to reduce the amount of data reported by examining the correlation between current sample with the previously obtained ones. Hierarchical sampling typically assumes multiple sensors where a low powered one is used to sample coarse-grained samples and high powered sensors are activated in case of an event is detected. Lastly, model-based active sampling combines data prediction with a computed model of data correlation.

## 2.2 Node Level Power Management Techniques

This section provides an overview of power management techniques of hardware components of the sensor motes. These techniques typically focus on reducing radio and CPU power consumption. Such techniques further reduce the energy consumption in addition to the network level energy aware protocols discussed in Section 2.1.

The basic Device Power Management (DPM) for WSNs started with exploiting active and sleeping energy consumption. In most cases the devices can turn most its components off and hence consume close to zero energy. However, there are two conditions to check. First one is the transition energy consumption. Typically turning a hardware component on requires significantly more power compared to its operating power level. Hence, there exists a break-even point  $p$  where it only saves energy if the device hardware component is put to sleep for more than  $p$  seconds. Several hardware components might also be linked such that it may require to put all of them into sleep simultaneously. In that case, the break even point needs to be calculated accordingly. Second one is the transition time consumption. Typically putting a device into sleep is instant but requires a significant amount of time to put it back to operating mode. This transition time needs be taken into account for real-time scheduling. Furthermore, devices might have multiple sleep levels which have different operating energy and transition energy/time consumption. Later advancement in technology has led to development of hardware components with multiple operating power levels. In WSN domain processor and radio typically have these capabilities.

Dynamic Voltage Scaling (DVS) is a well-studied energy management techniques applicable to processing units. DVS is the ability to change a CPU's supply voltage which leads to a change in the power level. It exploits the trade-off between the execution frequency (speed) vs execution time. By increasing (decreasing) the supply voltage, the frequency and its corresponding power level increases (decreases) and the execution time decreases (increases). For CMOS processors, power consumption is a convex function of voltage and processing frequency and is applicable to both high and low power microprocessors [44].

CMOS circuits have *static* and *dynamic* components where *static* is assumed to be constant necessary to keep the basic circuits on . On the other hand, *dynamic* changes with the supply voltage and consists of frequency *dependent* and *independent* subcomponents. It is typically assumed that *dynamic* is the dominant source of energy consumption and scales quadratically to the supply voltage  $E \propto V^2$  [45]. Increasing the computation time by reducing the frequency might lead to an increase of the frequency *independent* subcomponents' energy consumption. As a result, depending on the ratio of the underlying system, there may exist a threshold speed such that further reduction of speed may increase the overall energy consumption. Hence, for each system energy efficient frequency  $f_{ee}$  needs to be calculated which will represent the minimum frequency which further scaling will hurt overall energy consumption.

My research focuses on communication power consumption which is the dominating one. Transmitting a single bit of information is approximately equivalent to processing a few thousand operations in a typical sensor node [46]. Zolertia Z1 motes which I have used for my research is equipped with a CC2420 radio and a MSP430 microprocessor. CC2420 draws 17.4mA when receiving and 18.8mA when transmitting whereas MSP430 draws  $330\mu\text{A}$  current while active.

Similar energy conservation mechanisms of CPU are also applicable to radio modules. Dynamic Modulation Scaling (DMS), Dynamic Code Scaling (DCS), and Dynamic Modulation Code Scaling (DMCS) are among the widely used approaches [14,22]. I have applied DMS in my research which is a mechanism to manage the number of bits carried per signal. It requires higher energy to encode more bits in a single symbol and this increase in energy depends on the underlying modulation technique that is used. As a result, the transmission needs less symbol to transmit and hence reduces the transmission time. Similar to DVS, DMS also has *static* and *dynamic* power consumption components and effectiveness of DMS depends of *static* to *dynamic* ratio. *Static* component consumes energy independently from the used modulation level and is necessary to keep the basic circuit components on. *Transmission* power levels on the other hand are dynamic and can be manipulated by changing

the modulation levels. For shorter transmission distances, sender can use lower transmission output power for an error free communication which means less potential to save energy using lower modulation levels. As a result, the range of modulation levels which successfully preserves energy vs latency tradeoff depends on the underlying circuit architecture, modulation schema used and transmission distance. One fundamental different between DMS and DVS is the scheduling. Unlike DVS, DMS requires both communicating parties to agree on a modulation level which introduces additional overhead and synchronization challenges. In addition to that, computation task scheduling can take advantage of preemptive scheduling whereas communication tasks cannot be preempted. Finally, DMS is more susceptible to changing environmental conditions which effects the channel link quality.

The pioneering work presented in [14] led many researchers to consider DMS as an important leverage tool for low-power wireless communication. The authors presented the underlying principals of DMS and demonstrated how it can be used as an energy management technique. Some of these techniques are presented in Section 3.1.1. In the same paper, the authors also propose packet scheduling algorithms for real-time and non-real-time communication using DMS.

The work in [47] proposes modulation optimization algorithms for MQAM and MFSK schemas. The authors first break down the overall energy consumption of DMS into each hardware component's energy consumption and then analyze energy consumption per bit for different modulation levels. This paper was also one of the first to point out that the effectiveness of DMS depends on transmission distances. It has been shown that for sufficiently large distances, the lowest modulation level is the most energy efficient. However, for shorter distances this *may not* be the case, considering the increased weight of electronic circuitry consumption compared to that of the wireless radio.

DMS has also been studied in Wireless Sensor Network (WSN) domain, and this is the main focus of my dissertation. Authors in [48] studied integration of DMS into real-time data gathering for tree-topology based WSNs. They proposed an offline optimization algorithm as well as a distributed online algorithm for efficient modulation level assignments

and their simulation results have shown up to 90% energy savings. However, the proposed solutions were not applicable to TDMA-based protocols and assumed very minimal collision detection delay with light-weight traffic scenarios.

The work in [49] also incorporated DMS into tree-topology based WSNs. The novelty of the paper came from combining energy harvesting with optimal modulation level selection. The authors proposed a centralized optimal as well as a distributed near-optimal solution.

The authors of [18] proposed a slack generation and reclamation mechanism for real-time WSNs. They generated slack by analyzing data redundancies and by eliminating redundant data transmissions which in return created slack time. They then used this slack time to lower modulation levels to reduce energy consumption while meeting deadlines. The same research group later formulated joint scheduling of computation and communication tasks with DVS and DMS respectively as Mixed Integer Linear Programming Problem in [21]. They also proposed a polynomial-time heuristic and compared it against the optimal solution.

In [50], the authors applied DMS into cluster based WSNs with probabilistic workloads. For this scenarios, they formulated an optimal modulation level assignment to minimize overall energy consumption while meeting the deadlines. The same group later formulated optimal modulation level assignment for cluster based topologies while taking energy harvesting information into account. They have also proposed several polynomial-time algorithms and compared against the optimal solution.

The above papers, while important, exhibit a number limitations. Most crucially, they all assume discrete output power levels for given modulation levels. Although a higher power level is necessary to increase the modulation levels, the distance and current noise in the channel will also affect the exact output level which needs to be used for each pair of transceivers. Next, the modulation schema that have been used in the performance evaluations is primarily and mainly Quadrature Amplitude Modulation (QAM). However, only 16QAM was recently added to IEEE 802.15.4-2015 standard. It is possible that the selection of QAM in the vast majority of the DMS papers followed from the use of theoretical

energy scaling equations in QAM in pioneering research articles [14,47] papers, coupled with the absence of DMS capability in off-the-shelf commercially available low-power radios. In Section 6.1, I show the real world test results and how they differ from what the generalized mathematical formulas suggest.

IEEE 802.15.4 standard specifies many modulations for its physical layer, including DSSS-OQPSK, MSK, FSK, ASK, GFSK, SUN-OFDM, and MPSK. More recently, at the sub-GHZ level there are a number of commercially available radios starting to be offered with support to different modulation methods. These include GFSK, FSK, OOK, and MSK modulations [51]. Among this variety of modulations, only a few of them are commonly available in commercial IEEE 802.15.4 compliant radios. For instance, one commercially available data sheet shows that there are seven immediately or soon available IEEE 802.15.4 compliant radios listed in Texas Instruments CC radio series. All of the radios in this list are using 2.4GHz band and DSSS-OQPSK modulation. On the Sub-1 GHz side, twenty six currently or soon available radios are listed, which are not specified as IEEE 802.15.4 compliant but some of them can be used as one. These radios support GFSK, FSK, OOK, and MSK modulations [51]. Microchip-Atmel has a total of thirty IEEE 802.15.4 compliant radios listed with either 2.4GHz or Sub-1GHz base radios and which are all using DSSS-OQPSK [52]. This dissertation aims to address this gap by presenting empirical results obtained by using Software Defined Radios (SDRs) while complying with IEEE 802.15.4 specifications.

One important work currently exists in the literature is [53] which also uses SDR to analyze the impact of DMS however, in another domain namely *Inter-Vehicle Communication*. This work has analyzed BPSK, QPSK, 16QAM, and 64QAM in 802.11p stack. The authors reported the signal-to-noise ratios and corresponding packet-delivery-ratios for each modulation schema used. These tests were set up according to IEEE 802.11p standard where the underlying modulation is OFDM and the aforementioned modulation schemas are used as modulation mapping.

Perhaps the most closely related work to the third and the last part of my dissertation

is [21]. This paper studied DMS and DVS aware multi-hop scheduling. The authors first formulated interference and energy aware joint scheduling problem (IEJS) and proved that it is NP-hard by reducing Multiple Choice Knapsack Problem to IEJS. Although the solution presented in [21] did not consider any specific real-time protocol, the solution is generic enough that can be applied to both WirelessHART and 802.15.4e-2016. Here, the authors then propose a heuristic by creating a *mixed graph* of the given network topology such that the nodes that are within their interference range are connected with an undirected edge and precedence constraint is represented with a directed edge. The heuristic continues by scheduling the nodes with the larger depth nodes first and breaks the ties by giving priority to the nodes with higher undirected edges. In this initial scheduling step, the heuristic only assigns highest speed levels for both communication and computation. Due to the nature of this scheduling, a gap between the given deadline and the end of all transmissions might exist. This gap is utilized by lowering the computation frequencies and communication modulation levels according to their energy gain values. Other similar approaches also exist in literature [49]. My work differs from [21] because of the extended DMS model that I use. The authors of [21] assumes same interference and communication range for every modulation level. Also, each modulation level has a single discrete output power. I show in my research that for every modulation level, there is a range of output power levels that taken be chosen based on current noise and transceiver distances. More importantly, just because node<sub>a</sub> can talk to node<sub>b</sub> using modulation level  $m$ , that does not necessarily mean that they will still be able to conduct a successful communication by using modulation level  $m + 1$  given the power constraint of the hardware. Therefore, my research specifically chooses output power levels based on low-power radio energy consumption characteristics and limitations.

## 2.3 Summary

I have summarized node and inter-node level energy management techniques commonly used in wireless sensor networks. I have also stated the widely adopted wireless industrial control protocols. Inter-node level energy management techniques mostly based on duty cycling where the nodes transit between idle and active states. Node level approaches focuses on the hardware component energy management mostly by considering processing and communication units. These techniques include Dynamic Voltage Scaling, Dynamic Modulation Scaling, and Dynamic Power Management, and Dynamic Coding Scaling. Among these techniques, duty cycling, dynamic power management and dynamic modulation scaling are used in my dissertation.

## Chapter 3: Architectural Model

My research is looking into power management techniques for deadline driven scenarios. Specifically a shared deadline among the nodes where all the participating nodes of the network has to be done communicating before the deadline reaches.

As explained in Chapter 2, power management techniques can be designed for either node level and/or network level. My research combines both and design a cross-layer energy management solution. My preliminary work can also be seen as a combination of node and network layer where each node reduces its own energy consumption by lowering the modulation levels. On the other hand, each node is aware of cluster wide communication and opportunistically waiting for available slack time to claim under-utilized times.

My research primarily focuses on deadline driven, real-time communication systems. Two options are in place here, either a node level or a cluster level deadline. As more details are provided in Section 3.1, I focus on the applications where the nodes in a cluster shares a deadline. However, the algorithms I have designed respects node level deadlines and also applicable to those cases. This deadline is treated a hard deadline where every communication and speed levels are chosen accordingly.

DMS is the energy/latency tuning knob in my research. My proposed communication structure can be applied to other tuning knobs as well such as DVS. I have considered DMS because communication is the dominating energy consumption in wireless sensor networks. Transmitting one bit of information is estimated to be equivalent to a few thousand cycles of computation [46].

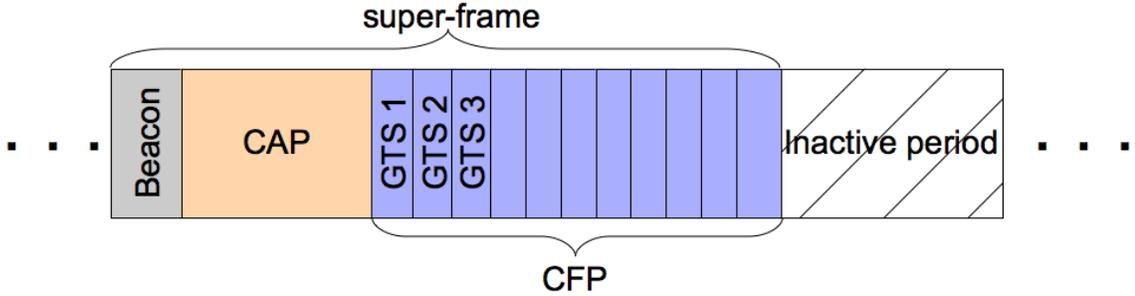


Figure 3.1: The lay-out of superframe

### 3.1 Cluster Model

In the first part of my work, I consider *probabilistic* workload as opposed to *deterministic* workloads. Designing algorithms for *probabilistic* workloads is more challenging and typically require a run time solution whereas *deterministic* workloads often has an offline solution which can be computed prior to the deployment of the network. Knowing the probability distribution of the workload allows us to optimize *expected* energy consumption.

My cluster model focuses on nodes that form single-hop communication clusters. Each node is assumed to periodically generate some number of packets that it must transmit by a specific deadline. The actual number of packets changes over time, and is only known probabilistically.

In order to meet these requirements I use a generic, beacon-enabled superframe architecture for real-time communication. I assume every node participating in the cluster can directly communicate with the coordinator. I also assume every node only wants to talk to and listen to the coordinator. This is a very realistic assumption because the main application model of the WSN is to collect data and transmit to a base station. Most of the routing protocols used by WSNs such as RPL [54] are mainly designed according to this principle. The energy consumption of the coordinator is not a concern in my system model, I assume the coordinator is a base station which has high computation power and has unlimited power supply.

Superframe architecture is shown in Figure 3.1. As seen, at the beginning of each superframe the beacon is transmitted by the coordinator. The beacon contains management information such as TDMA slot assignments, and is received by all the nodes in the cluster. This is followed by a contention access period (CAP), allowing each node to talk to the coordinator via CSMA. During this phase a node may send future workload information, or may ask to join or leave the cluster. The contention free period (CFP) starts right after the CAP. The CFP consists of a series of Guaranteed Time Slots (GTSs), which are assigned to specific nodes. In order to provide real-time communication guarantees, each node is assigned a number of GTSs equal to its worst-case traffic workload.

The coordinator manages GTS assignments to avoid collisions and provide real-time guarantees. The CAP can be followed by an inactive period during which the nodes can sleep. I assume that the coordinator manages the duty cycle and, without loss of generality, do not consider its impact in my work. Studies have shown that reducing collision saves energy tremendously [12]. The end of superframe is a hard deadline where all the communication has to be completed before the next superframe starts.

A traditional approach is to have a node sleep during the GTSs assigned to other nodes. In this research, I introduce a Hybrid Low-Power Listening protocol that allows nodes to *proactively* remain awake during time slots assigned to other nodes to attempt to opportunistically transmit their packets at a reduced energy level.

### 3.1.1 Communication

I assume that each node is equipped with a DMS-enabled radio capable of dynamically adjusting the modulation levels. I adopt the basic energy model presented in [14]. Specifically, the radio power consumption is divided into two parts. The first is *transmission power*, denoted as  $p_s$ , and the second is *electronic circuitry power*, denoted as  $p_e$ . These values can be expressed as  $p_s = C_s \times \phi(b) \times R_s$  and  $p_e = C_e \times R_s$ , respectively. Here,  $R_s$  is the number of symbols transmitted per second and  $b$  is the modulation size. The values  $C_e$  and  $C_s$  are radio-specific; but  $C_s$  can be affected by the current environmental conditions,

Table 3.1: Scaling functions

Modulation scheme:	$2^b$ -QAM	$2^b$ -PSK	$2^b$ -PAM
$\phi(b)$ :	$2^b - 1$	$(\sin \frac{\pi}{2^b})^{-2}$	$\frac{2^{2b}-1}{3}$

such as atmospheric noise, transmitter-receiver distance and temperature. In practice  $C_e$  and  $C_s$  can both be approximated as constants. Finally  $\phi(b)$  is the convex scaling function of the modulation used, depending on the modulation scheme. For QAM,  $\phi(b)$  function is  $2^b - 1$  [14] for even modulation levels and a close approximation for odd modulation levels, which shows the exponential increase in power consumption in terms of the modulation level ( $p_e$  is assumed to be constant). Table 3.1 shows the scaling function of PSK and PAM modulations.

The transmission time, on the other hand, is  $\frac{1}{b \times R_s}$  which decreases linearly in terms of modulation level. As a result, the tradeoff of DMS becomes an exponential increase (decrease) of transmission power with a linear decrease (increase) of transmission time for QAM [14].

My HLPL protocol uses two schemes commonly used in asynchronous duty-cycled low-power MAC protocols, namely low-power listening (LPL) and embedding information in short preamble (physical layer) packets. A typical LPL protocol such as B-MAC [24] requires a sender to transmit a long preamble. Receivers wake up, sense the preamble, and stay awake to receive data. The duration of listening and sleeping schedules can be adjusted to, for example, maximize energy savings, maximize throughput, or minimize delay. Protocols such as X-MAC [55] extend this idea by using shorter preambles with embedded address information of the target node. The advantage of using preamble addressing is a reduction in the number of bits that need to be transmitted and a flexible, user configurable information that follows the preamble. X-MAC also showed embedding address information

into preamble avoids the overhearing problem and saves energy on the non-target devices.

In order for the parties to successfully communicate, they need to agree on the modulation levels prior to the beginning of the transmission. In my communication model, I achieved this goal by using link-layer *preambles* which is broadcasted by the coordinator prior to each GTS. These preamble messages contain the address of the receiver and the modulation levels for the communication. This way both the coordinator and the sender can agree on the modulation levels. One other commonly used technique to achieve the same goal is to use physical layer headers [56].

### 3.2 Multi-hop Model

In my data monitoring model, I assume there are 3 types of nodes; sensing-nodes, sense-and-relay-nodes and a base-station. Sensing nodes are located on the edge of the network whose only duty is to sense and report. Sense-and-relay-nodes, on the other hand, not only sense the environment but also receive packets from sensing-nodes and forward it to other sense-and-relay-nodes or to the base-station. Here sensing duty of sense-and-relay nodes is optional. I also assume each node generates a constant amount of data after each sensing operation. Base-station is the node which is the final destination for all the information collected from the network. Sensing and sense-and-relay-nodes are assumed to be battery powered whereas base-station has unlimited power supply. Moreover, I also assume each node is equipped with a radio that supports  $\{2, 4, 8\}$  PSK modulations. The modulation schema can be replaced with any other schemas as long as the PDR are known given environmental conditions.

My system model also assumes the application which is using this network is time-critical which means there has to be a delay bound where all the sensed data from every sensing-node needs to arrive to base-station within a specific delay bound depicted as  $D$  in Figure 3.2. This effectively implies that each sensing-node's packet generation rate,  $\lambda$ , is  $1/D$ . Also, the nodes are time-synchronized which is a viable assumption especially for Wireless Industrial Control Networks running WirelessHART protocol due to its built in

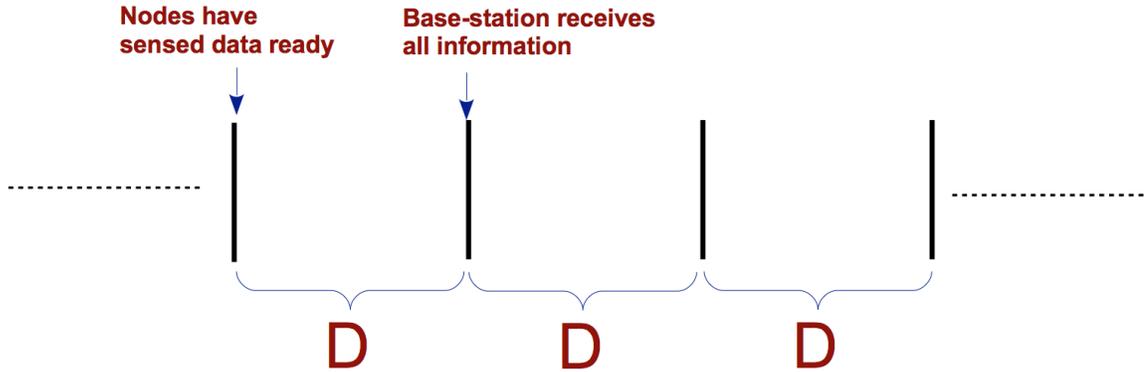


Figure 3.2: Delay bound

time synchronization mechanism. More importantly, optimization formulation presented in Section 7.1 assumes we have a network such that the information of a set of parent nodes to forward the packets given a node and a modulation level (potentially different for each  $\{2, 4, 8\}$  PSK modulations) is known. Also, I assume the interference range given a modulation level (once again different for each  $\{2, 4, 8\}$  PSK modulations) and the set of nodes within that interference range is known. These are also viable assumptions due to the all-knowing nature of Industrial Control Networks. WirelessHART for example requires a centralized *network manager* who has the global information about each link quality and network topology. Each node submits *health reports* every 15 minutes. These health reports contain neighbor table with observed RSSI values as well as traffic rates, priorities and deadlines [57]. It is network manager's duty to compose a routing graph (**offline**) according to these collected reports. In case of a change in topology or significant changes in network link qualities, network manager is expected to recompute this routing graph.

Figure 3.3 shows an example network setup. Here  $\{n_1, n_5\}$  are sensing-nodes.  $\{n_2, n_3, n_4, n_6, n_7\}$  are sense-and-relay-nodes,  $n_8$  is the base-station. The arrows show the directional links with a given modulation level. Please note that these interference ranges are drawn as unit-disk models for ease of demonstration but this may not be the case for real-world deployments. According to this setup,  $n_1$  can talk to  $n_3$  using modulation levels

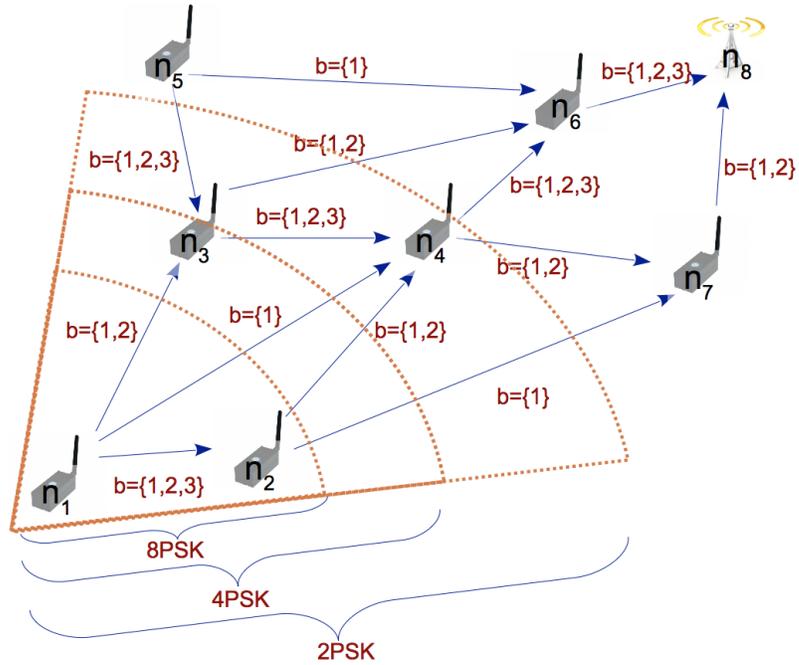


Figure 3.3: An example network. Each link has at least a predetermined PDR rate.

1 and 2. The Figure also shows the interference range of  $n_1$  for each modulation level. For 8PSK  $n_1$  has the interference range set of  $\{n_2\}$ , for 4PSK  $\{n_2, n_3\}$ , for 2PSK  $\{n_2, n_3, n_4\}$ .

Lastly, I would like to add that, it is possible to apply the same system model to multi-cluster networks such that each sensing-node and sense-and-relay nodes can be replaced with a cluster where a cluster head can report the aggregated sensing information from its own cluster and forward it to other cluster heads.

## Chapter 4: Minimizing Expected Energy Consumption of Time-Critical Cluster WSNs with Probabilistic Workloads

This chapter of my work focuses on 802.15.4 beacon enabled MAC layer protocol designed for deadline driven applications and utilized the superframe structure discussed in Section 3.1. This work addresses node level communication power management for nodes with probabilistic workload. This approach requires to minimize expected energy consumption where I assume probabilistic workload models are known.

To this aim, I first formulate the binary integer programming problem of computing the optimal modulation levels given the workload probability distribution function which minimizes the expected energy consumption while meeting the deadline. The solution to this problem can be computed offline and outputs a *speed scheduling* solution. Under speed scheduling, the nodes starts transmitting with low communication speeds and gradually increases the current speed as the deadline approaches. Next, I propose dynamic superframe structure where the guaranteed time slots (GTSs) can be adjusted online according to the current communication workload. To this end, the nodes need to be aware of the current communications on the channel which requires additional energy consumption. In order to overcome and reduce this overhead, I present a new novel low power listening protocol called *Hybrid Low Power Listening*.

By enabling HLPL, I design several online algorithms which outperforms the aforementioned optimal offline solution in certain cases. The evaluation of the dynamic superframe technique and HLPL are done with both a custom built Matlab simulation and by using Castalia framework of Omnet++ Simulator. Here, Castalia based results are presented. Additional hardware based results are also included.

## 4.1 Dynamic Superframe

The advantage of DMS is that the nodes can minimize overall energy consumption by using lower modulation levels. The drawback is that lower modulation levels require longer periods of time to transmit the same number of bits. In my environment this means that more GTSs are required. I assume that the nodes can bound their worst-case workload in terms of the number of packets to send, but the actual distribution is only known probabilistically. This means that the nodes may send fewer packets than their worst case estimate. Hence, a node might not use all of the GTSs assigned to it. It is therefore desirable to devise *dynamic* algorithms capable of assigning these unused slots to other nodes. Further, it is possible to use DMS techniques to extend transmission over unused slots, in order to reduce their energy expenditures, as long as the deadlines are maintained. I call the extra time available from unused slots "slack time".

Figure 4.1 shows how different algorithms may behave when slack time is available. I can generalize these slack reallocation algorithms as static and dynamic. Assume the initial GTS assignments are shown in the superframe labeled *A*. In this example *node*<sub>1</sub> has been assigned 3 packet-length GTS but it only transmits a single packet. Superframe *B* shows what happens under the traditional static approach. The GTS assignments for *node*<sub>2</sub> and *node*<sub>3</sub> do not change, and the available slack time remains un-utilized. The superframe *C* shows a possible dynamic approach where these slots are reallocated to *node*<sub>2</sub>, which can lower its modulation level but still meet the deadline. Another possible dynamic approach is *dynamic\_fair*, shown in superframe *D*. In this case the slack could be allocated among *node*<sub>2</sub> and *node*<sub>3</sub> equally. Detailed descriptions of these algorithms along with methods for determining modulation levels for the new GTS distribution are provided in Section 7.3.

For dynamic algorithms to succeed, the nodes need to be aware when the currently scheduled node prematurely finishes transmission. My approach works by having the coordinator broadcast a relatively short **preamble** that contains the address of the next node to transmit and the modulation levels that the node will use. Nodes hear this preamble by using low-power listening. The selected node may in fact be granted permission to transmit

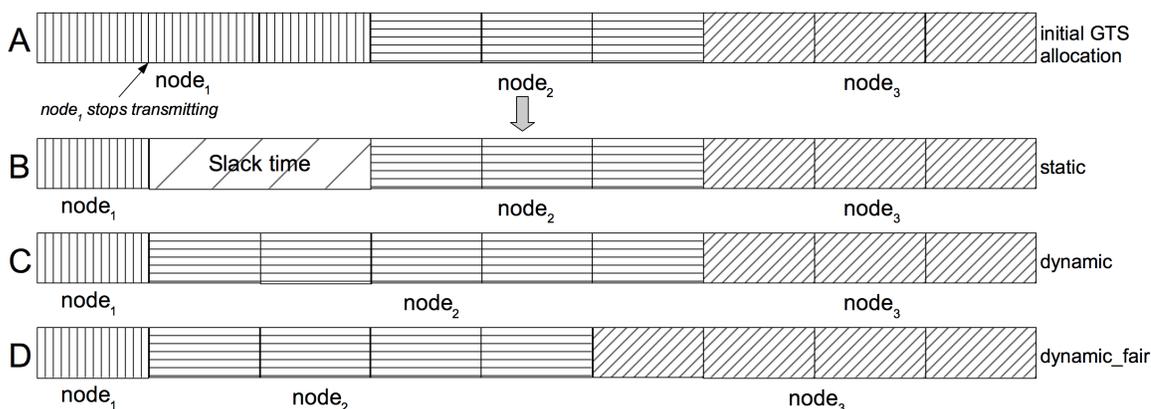


Figure 4.1: Illustration of static and dynamic slack reallocation

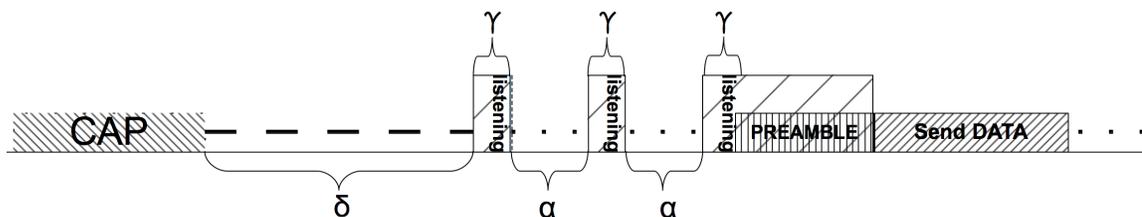


Figure 4.2: Parameters of the Sleep-Listen Cycle

early using the available slack time.

Figure 4.2 shows the parameters of the listen-sleep cycle from the perspective of a single node. Here  $\delta$  shows the *wait-duration* before the node starts its low-power listening (LPL) cycle. The node is entirely asleep during the period  $\delta$ . Initial intuition is to set  $\delta$  value to zero which means the nodes will start performing LPL as soon as the CAP period ends to ensure no preamble will be missed. As I will show in Section 7.3, performance improvements can be achieved with a careful choice of a non-zero  $\delta$  value. However, this is possible only in the cases where I have a priori information about the packet workloads of the nodes. For the LPL phase, I use parameters  $\alpha$  and  $\gamma$ , referred to as the *sleep-duration* and *listening-duration*, respectively.  $\gamma$  is the time during which the node is listening to the medium whereas  $\alpha$  is the time during which the node is in sleep mode.

The length of the coordinator's preamble has to be greater than the LPL period of  $\alpha + \gamma$ .  $L_{preamble} \geq \alpha + \gamma$  guarantees that the receiver will hear a portion of the preamble. However,

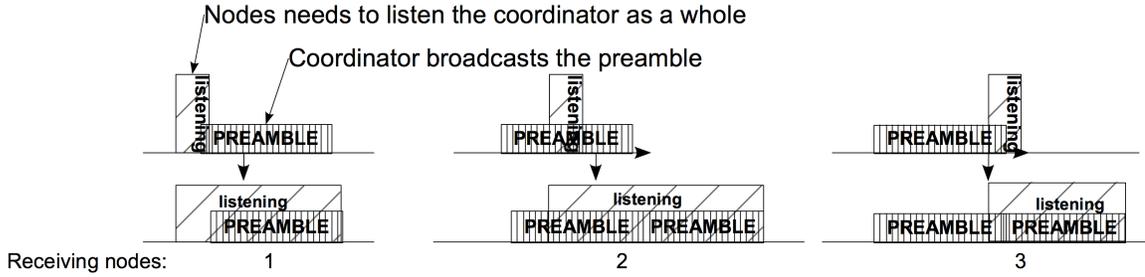


Figure 4.3: Possible intersection of listening period and preamble

it does not guarantee that the receiver will listen to the preamble as a whole.

Figure 4.3 shows the possible intersections of the LPL period and preamble. Among these three possibilities, the first one is desired, since the receiver receives the entire preamble. In the second and third cases, the receiver will hear the preamble; however, it will not know who the preamble is addressed to. As a result, the receiver will need to keep listening even after the preamble message is over, in order to learn the address of the preamble. This adds to the power consumption of the receiver. The coordinator needs to make sure that after sending the preamble message, the intended node starts transmitting. If not, the coordinator needs to re-send the preamble. I evaluate the effect of these parameters in Section 5.

An additional problem exists in that the absence of transmission activity being detected by a node does *not* necessarily mean that nodes are not transmitting in the cluster. Two nodes may be entirely out of each others' radio range. Another possibility is that a node may be in another node's interference range, but not its transmission range. This means that a node can hear another node transmit but cannot decode the transmission. Also, when a node is in the transmission range of another node, it overhears the communication. However, nodes are only interested in transmissions from the coordinator. Listening to the other nodes in addition to the coordinator increases the energy consumption. For my scheme, the practical impact is that a node may not be able to hear another node that is in the process of sending a packet to the coordinator, and therefore cannot tell if the slot

is used or idle. Further, it may overhear the unintended communications with additional energy cost. I refer to these issues as the *neighborhood* problem.

In order to overcome the *neighborhood* problem, I combine preamble addressing with my newly defined hybrid-low-power listening (HLPL) protocol. As described earlier HLPL is a combination of traditional low-power listening and a new scheme called *reverse-low-power listening* (RLPL). On the receiver side, the node needs to decide which LPL mode it needs to be in. In HLPL, if a node receives a preamble and learns that it is not scheduled next **and** it senses any transmission during its *first* wake-up after this preamble, the node goes into the RLPL (described below) stage. For non-zero wait-duration values, when the node wakes up, it listens to the channel for a preamble. If it senses any transmission and this transmission is not a preamble then the node goes into the RLPL stage. However, if this transmission is a preamble that is not addressed to itself and if it does not hear any transmission during its first wake-up after the preamble, it goes into the traditional LPL stage.

For zero wait-duration values the nodes (except for the first scheduled node) start with traditional LPL. During listening phases if they do not hear any transmission, they stay in the tradition LPL stage. However, when a node senses a transmission after a preamble, it goes into the RLPL stage. The logic behind this process is the fact that hearing a transmission during the first wake-up right after the "false" preamble indicates that there is an interference since the sleep-duration is smaller than length of a single packet. If the node does not hear any transmission after a preamble addressed to another node means there is no interference. Meanwhile, the coordinator waits for sleep-duration amount of time before it broadcasts the preamble, unlike in traditional LPL, where a sender broadcasts the preamble as soon as the current node stopped transmitting. Waiting for *sleep-duration* amount of time ensures that all the nodes that are in RLPL mode are currently in the *wait-for-preamble* stage. Figure 4.4 shows the flow chart for HLPL.

RLPL differs from traditional LPL in its conditions to transit between listening and sleeping stages. In RLPL when the node wakes-up and hears a transmission, it goes back

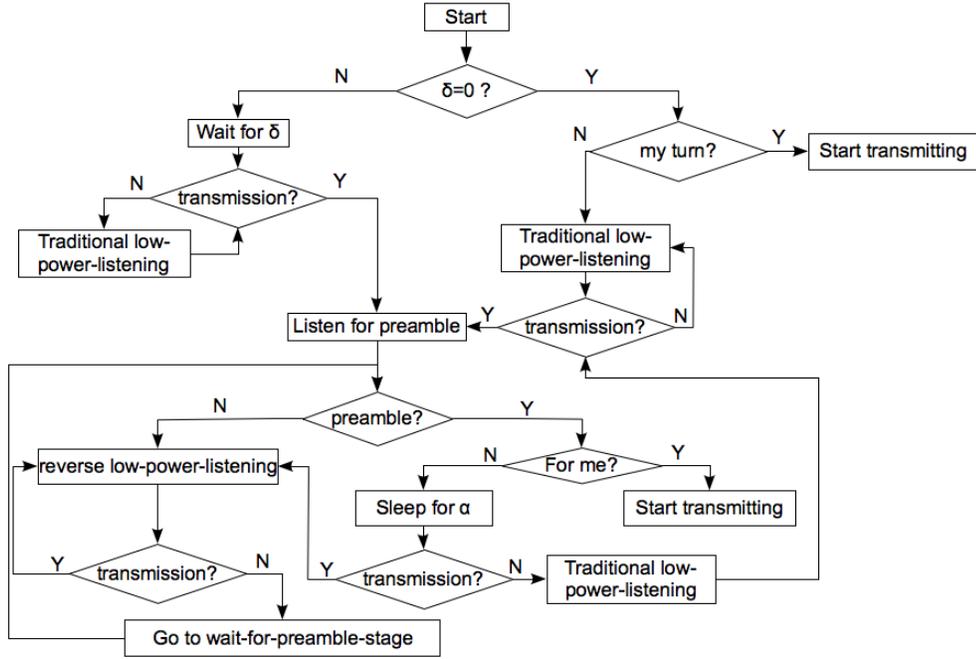


Figure 4.4: The steps of HLPL

to sleep. However, if it does not hear any transmission, it starts listening, which is different than the traditional low-power listening. In RLPL, this listening phase is called *wait-for-preamble* stage. As explained, when a node stops transmitting, the coordinator waits for *sleep-duration* amount of time before it broadcasts the preamble. Hence, *wait-for-preamble* stage can last at most for the sleep-duration time. *Wait-for-preamble* guarantees that when the coordinator broadcasts the preamble, the nodes will be listening to the channel.

The core idea of HLPL is to save energy when there is *constant* traffic in the network. In the absence of this, HLPL behaves very similar to traditional LPL. Figure 4.5 aims to clarify the difference between traditional LPL and RLPL during a constant traffic. Under traditional LPL, the node wakes up periodically and tries to sense a transmission. Then it stays awake long enough to conclude that the transmission is not from the coordinator. On the other hand, under RLPL, the node first listens to the channel, realizes that it is not a preamble, and goes to the RLPL stage. With RLPL, a node still periodically wakes up but stays awake enough to detect that there is *some* transmission. If so, the node goes back

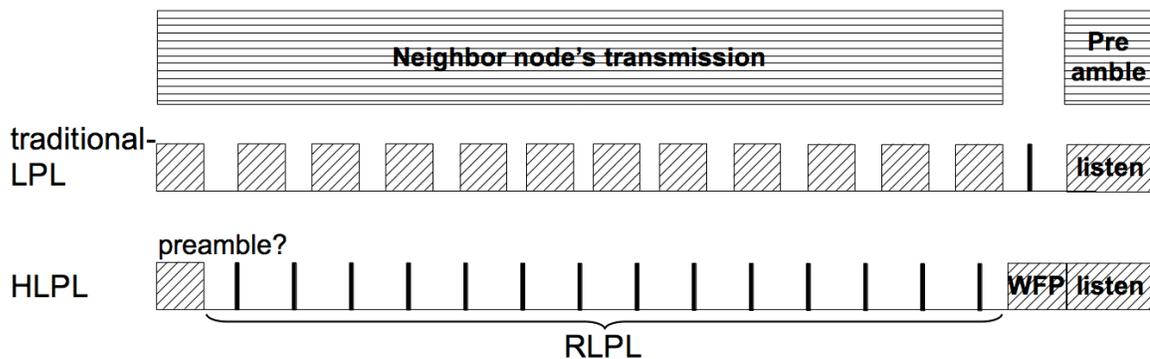


Figure 4.5: Comparison of traditional LPL and RLPL

to the sleep mode. Otherwise, the RLPL stage concludes, and the node transitions to the wait-for-preamble stage. It stays in that stage until it receives the preamble.

Neighborhood size creates a big overhead in my application due to the high frequency of communication during a superframe. Low-power listening traditionally adopted to the cases where the nodes need to listen the channel for a long duration of time where the sample rate (packets/sec) of the system is low. A state-of-art low-power listening enabled MAC protocol B-MAC [24] assumes the initial sampling rate for its performance evaluation settings is 1/300 packets per second in its experiment settings. Low-power listening in similar scenarios saves tremendous amount of energy compared to idle listening the channel instead. However, a TDMA based superframe approach is typically adopted when the network has high sample rate where it is almost certain that there will be many collisions at any time in the duration of superframe. Typically a superframe has a full utilization where there is always a node transmitting at any given time.

What is required in my case is that in the lights of what has been described in the previous paragraph, adopting traditional low-power listening into a superframe can be thought of as combining two fundamentally contradicting ideas. As described previously, my proposed *dynamic* algorithms need to know when the coordinator broadcasts an information-rich preamble. In the presence of *neighbors*, when a node wakes up to check for a preamble it

overhears the communication from its neighbors where it needs to listen to this communication up to two preamble times to make sure that it is not receiving a preamble. In an environment where there is always two parties communicating, the overhead of overhearing neighbors' communication cancels out the energy saving that are obtained from *dynamic* algorithms.

In RLPL, In this stage, whenever the node senses a transmission when it wakes up, it goes back to sleep right away. However, when the node wakes up and there is no transmission, it will go into *wait-for-preamble* stage and remain awake till it gets a preamble. In *traditional* low-power listening, the coordinator was broadcasting the preamble as soon as the current node stopped transmitting. However, in RLPL, the coordinator waits for sleep-duration before it broadcasts the preamble. This ensures that all the nodes in the system is currently in *wait-for-preamble stage*. The reason why this scheme is called *reverse* is because when the node hears a transmission, it goes back to sleep and when the node does not hear any transmission, it starts listening in contrast to *traditional* low-power listening.

Note that during the course of the superframe, the nodes have to use both *traditional* and *reverse* low-power listening. The condition of hearing a transmission during the first wake-up right after the "false" preamble indicates that there is an interference since the sleep-duration is smaller than length of a single packet. If the node does not hear any transmission after the "false" preamble that means there is no interference and it continues with *traditional* low-power listening. Figure 4.4 shows the flow chart of the decision process for reverse and traditional low-power listening.

## 4.2 Joint Deadline-Energy Optimization problem

Based upon the number of nodes, the real-time constraints, and the actual workload, the question remains how to set the modulation levels to achieve all deadlines and conserve energy. I now show how to formulate this question as an optimization problem.

Earlier research in DMS has shown that there exists a *constant* optimal modulation

Table 4.1: List of symbols

symbol	description
$n$	Number of nodes
$m_i$	Upper limit on the number of packets $node_i$ can send
$D$	length of superframe
$p_i(k)$	Probability that $node_i$ 's workload is $k$ packets
$y_i(k)$	Probability that $node_i$ sends $k$ or more packets
$b_i(k)$	Modulation level used by $node_i$ to send its $k^{th}$ packet
$L$	Maximum transmission unit of the underlying communication protocol in bits
$t_{bit}$	Time to send a bit
$t_{symbol}$	Time to send a single symbol
$R_s$	Symboling rate
$b_{min}$	Minimum modulation level that a node can use
$b_{max}$	Maximum modulation level that a node can use
$\beta_l^{i,k}$	A binary indicator that equals 1 for the selected modulation level $l$ for $node_i$ 's $k^{th}$ packet
$p_e$	Power consumption of the electronic circuitry
$p_s$	Power consumption from transmission

level that minimizes the energy consumption while meeting all deadlines under *deterministic* workloads [14]. However, the work in [58] observed that under probabilistic workloads, this is not the case. Instead, the optimal solution to minimize the *expected* energy consumption consists in transmitting the first packets at low speed (modulation), and increasing the speed gradually for the subsequent packets when approaching the deadline. This is based on the observation that in the more likely scenarios where the actual workload deviates from the worst-case, low modulation levels are sufficient to meet the deadline while saving significant energy. However, as more packets are transmitted, the modulation level is gradually increased to meet the deadline. The framework to find the optimal modulation levels given a deadline and probabilistic workload profile is called *speed scheduling* in [58] and I also adopt this approach.

For my targeted applications each node has a varying communication workload determined by a known probabilistic distribution. The  $node_i$  can have from 1 to  $m_i$  packets to transmit in a given superframe.  $p_i(k)$  represents the probability distribution function of

$node_i$ 's workload. Specifically,  $p_i(k)$  denotes the probability that  $node_i$  will transmit exactly  $k$  packets during a superframe.

The energy needed to transmit a single packet,  $e_{packet}$ , is the product of time to send a single bit ( $t_{bit}$ ), the length  $L$  of the maximum transmission unit (in bits), and the total power ( $p_s + p_e$ ). Moreover,  $t_{bit} = \frac{1}{b \cdot R_s}$  where  $b$  indicates the modulation level and  $R_s$  is constant. A typical value for  $R_s$  is 62500 symbols/second for 802.15.4 [59]. By using the radio power consumption formula, I get:

$$e_{packet} = L \cdot (p_s + p_e) \cdot t_{bit} = \frac{L \cdot (C_s \cdot \phi(b) + C_e)}{b} \quad (4.1)$$

Define  $y_i(k)$  as the probability that node  $i$  will actually transmit the  $k^{th}$  packet. Then  $y_i(k) = \sum_{x=k}^{m_i} p_i(x)$ . The total *expected* energy consumption is the sum of expected energy consumption of  $n$  nodes:

$$e_{expected} = \sum_{i=1}^n \sum_{k=1}^{m_i} e_{packet} \cdot y_i(k) \quad (4.2)$$

By denoting the modulation level of the  $k^{th}$  packet of the node  $i$  by  $b_i(k)$ , I obtain:

$$e_{expected} = \sum_{i=1}^n \sum_{k=1}^{m_i} \frac{L \cdot y_i(k)}{b_i(k)} \cdot [C_s \cdot \phi(b_i(k)) + C_e] \quad (4.3)$$

Note that the  $k^{th}$  packet of node  $i$  can potentially be transmitted with any of the discrete modulation levels in the range  $[b_{min}, \dots, b_{max}]$ . Let  $\beta_l^{i,k}$  be a binary indicator variable  $\in \{0, 1\}$  to represent whether the  $k^{th}$  packet of node  $i$  is transmitted using the modulation level  $l$  or not. Then an integer programming formulation to minimize the expected energy can be obtained as:

$$\text{minimize} \quad \sum_{i=1}^n \sum_{k=1}^{m_i} \sum_{l=b_{min}}^{b_{max}} \beta_l^{i,k} \cdot \frac{L \cdot y_i(k)}{b_l} \cdot [C_s \cdot \phi(b_l) + C_e] \quad (4.4a)$$

$$\text{subject to} \quad \sum_{l=b_{min}}^{b_{max}} \beta_l^{i,k} = 1 \quad \forall i, k \quad (4.4b)$$

$$\sum_{i=1}^n \sum_{k=1}^{m_i} \sum_{l=b_{min}}^{b_{max}} \beta_l^{i,k} \cdot \frac{L}{b_l \cdot R_s} \leq D \quad (4.4c)$$

$$\beta_l^{i,k} \in \{0, 1\} \quad \forall i, k \quad (4.4d)$$

The objective function gives the sum of the energy consumption of all the packets over all the nodes, by considering their probability of being transmitted and all possible modulation levels. The constraints (4.4b) and (4.4d) indicate that exactly one modulation level will be assigned to each packet in the workload. The constraint (4.4c) enforces that all the modulation levels must be selected in a way that all the transmissions will be completed before the deadline (the end of the superframe). The selected modulation levels can only be integer values where  $b_{min} \leq b_{selected} \leq b_{max}$ . Hence, this formulation corresponds to an Integer Programming problem where the main constraint is to complete the communication tasks before the deadline. Although integer programming problems are known to be intractable in the general case, moderate-size instances can be solved using the existing optimization tools such as CPLEX.

Recall that communication energy consumption has two components. The circuitry energy consumption increases with the communication time whereas the transmission energy consumption decreases. As I keep decreasing modulation levels and increasing transmission times, the circuitry energy consumption may dominate the transmission energy consumption. The work done in [22][60] both verifies that for distances greater than 25 meters, transmission energy consumption is the dominating factor. Hence, in my preliminary work

I assumed that the distance between the communicating parties are greater than 25 meters to better simulate the benefits of DMS. However, my preliminary work can easily be extended for any distances by simply setting the first derivative of the radio energy consumption function to zero and calculate the minimum modulation level that this scheme works [58].

## Chapter 5: Performance Evaluation

### 5.1 Proposed Algorithms

Section 4.1 showed how static and dynamic algorithms may behave when slack time is available. Section 4.2 showed how to optimally calculate the modulation levels to achieve energy minimization and required performance. Now I will discuss the algorithms that combine both.

#### **Static**

In this algorithm, assuming the worst-case workload (i.e., 10 packets) for every node, the smallest possible modulation level with which the deadline can be met is assigned to all the nodes in uniform manner statically. The modulation levels are precomputed and broadcasted with the beacon message at the beginning of each superframe. The assigned modulation levels do not change for the duration of the superframe, even though the actual workload of a node may deviate from the worst-case (i.e., slack is not reclaimed).

#### **Static\***

This is similar to *Static* algorithm in the sense that the modulation levels are computed statically and slack is not reclaimed. However, instead of assigning a constant modulation level to every node, the nodes use a speed schedule that gradually increases the modulation levels by exploiting the probabilistic workload profile. This is computed by solving the integer programming problem with the objective function given in Equation (4.4a). The modulation levels are distributed with the beacon message.

#### **Dynamic**

This algorithm makes initial modulation level assignments as in *Static* but then dynamically adjusts the modulation levels in order to take advantage of the available slack time after the end of each node's transmission and allows only the following node to reclaim this slack

time by adaptively reducing the modulation level. At slack reclamation times, each node uses the smallest feasible modulation level to use the duration of its originally allocated slots and reclaimed slots. Recomputed modulated levels are distributed with the preamble messages.

### **Dynamic\***

This algorithm enables dynamic reclaiming of the unused slots by adaptively reducing the modulation level at run-time. However, the initial modulation levels are computed using *Static\** and the node that reclaims the slack uses the speed scheduling solution to re-assign possibly different modulation levels to each of its packets. Recomputed modulated levels are distributed with the preamble messages.

### **Dynamic\_f**

The *fair* version of the *Dynamic* algorithm in the sense that the available slack time is distributed evenly among *all* subsequent nodes rather than being assigned entirely to the next node. The modulation levels of all the subsequent nodes are dynamically adjusted after the end of each node's transmission to the lowest feasible modulation level.

*Static*, *Dynamic*, and *Dynamic\_f* are polynomial-time algorithms. They only iterate over each modulation level (from 2 to 8) once and select the minimum feasible one. *Static\** solves the Binary-Integer Programming Problem introduced in Section 4.2 but it is executed offline by the coordinator and only once unless the probability distribution changes. *Dynamic\** also solves the same Binary Integer Problem but only for a single node. As a result, it runs extremely fast due to small problem size and also due to known good performance of Branch and Bound technique on Binary-Integer-Problems. In practice, a look-up table can be constructed with the pre-computed modulation levels as a function of available slack.

Figure 5.1 shows an example of possible slack reclamation of dynamic algorithms. In this example there are 5 nodes with maximum workload of 10 packets. Initially, each node is assigned slots with total length equal to 10 packets with the modulation level  $b$  where  $b > b_{min}$ . When it is  $node_1$ 's turn, it sends 6 packets using modulation level  $b$  which yields a slack time of four packets long. *Dynamic* and *Dynamic\** allocate this slack time to the

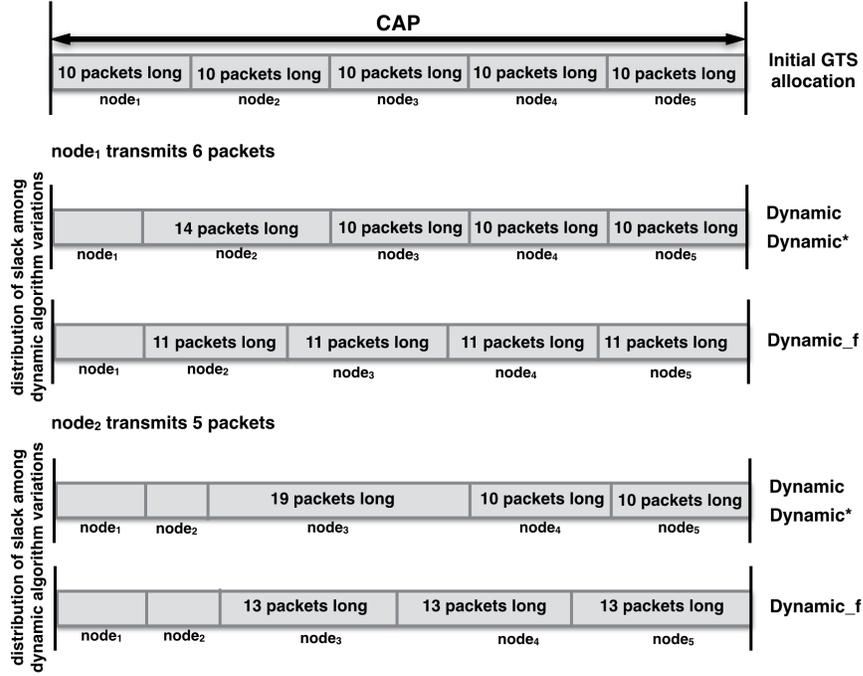


Figure 5.1: An example for dynamic algorithms

next scheduled node, namely  $node_2$ . Now,  $node_2$  has effectively additional slots, giving a transmission time equal to 14 packets. However,  $node_2$  will transmit at most 10 packets so it can reduce its modulation levels. In the case of *Dynamic*,  $node_2$  uses the lowest feasible modulation,  $b_D$ , where  $b_D < b$  for each of its 10 probable packets. *Dynamic\** uses the optimal modulation levels computed by solving Equation (4.4a) only for its probabilistic workload and slot length. When it is  $node_2$ 's turn, it ends up transmitting 5 packets implying there is a slack time of 9 packets with modulation  $b$ . Similarly, *Dynamic* and *Dynamic\** assign this slack time to the next scheduled node, namely  $node_3$ . In the *Dynamic* case,  $node_3$  uses the lowest feasible modulation level,  $b_{D'}$ , where  $b_{D'} \leq b_D \leq b$ .  $node_3$  uses the optimal solution computed for its own packets with its own deadline. In the *Dynamic-f* case, the 4-packet long slack time after  $node_1$ 's transmission is distributed among  $node_2$ ,  $node_3$ ,  $node_4$ , and  $node_5$ . These nodes have 11-packet long slack time with the modulation level  $b$ . The lowest feasible modulation level,  $b_{D_f}$ , that will meet with the deadline with 40 possible

packets is computed where  $b_{D_f} \leq b$ . After  $node_2$  stops transmitting, the 9-packet long slack is distributed among  $node_3$ ,  $node_4$ , and  $node_5$ . The new lowest feasible modulation level  $b_{D'_f}$  for all 30 possible packets to meet the deadline is computed where  $b_{D'_f} \leq b_{D_f} < b$ .

## 5.2 Simulation Settings

To evaluate the performance of the several variants of the proposed algorithms (described below) under different workload conditions, I simulated the system on Castalia framework of Omnet++ simulator. I simulated a system with a coordinator and 10 nodes arranged in star topology, and with communication range set to 30m. The work done in [21] shows that DMS is effective for distances greater than 25 meters. Each node’s workload in a superframe varies between 1 to 10 packets and is derived from a probability distribution. I assumed DMS-capable systems (with QAM modulation) where the modulation levels can vary from 2 to 8.

The purpose of my simulation is to quantify, from an algorithmic perspective, the difference between DMS-aware and DMS-oblivious approaches in energy aware super-frame management. In order to achieve this, I ran various simulations for different superframe lengths (deadlines) to analyze how the energy consumption varies. Furthermore, I have ran my proposed algorithms against an *Oracle* algorithm which is the *yardstick* algorithm where the exact number of packets that each node will transmit is known in advance, at the beginning of each superframe. As a result, it does not need to assume the worst-case workload. *Oracle* does not require any LPL because it knows the exact time each node will stop transmitting. Hence, the overhead of LPL is also omitted. Although it is not a feasible algorithm in practice, it provides the minimum energy consumption that is theoretically possible for a given experiment.

The minimum deadline  $D_0$  is assumed to be the superframe length necessary to allow the transmission of the worst-case workload (10 packets) by each node at the maximum modulation level, considered to be equal to 208 ms<sup>1</sup>. The actual deadline for a given

---

<sup>1</sup>As in low-power listening mode each node can miss up to 2 preambles before it can start transmitting,

experiment is then computed as  $D = \frac{D_0}{load}$  where the system's *load* is in the range [0.1, 1.0].

For each *load* value, the simulator generated 600 workload instances for 10 nodes using a uniform distribution function. I also repeated the experiments with Normal, Pareto and Flipped-Pareto distributions for workload generation. Detailed simulation results for the Uniform distribution and underline the trends and relative ordering of the schemes for the remaining distributions will be presented.

Castalia is a high fidelity simulator with an advanced channel model that incorporates log normal path loss with temporal variations [61]. It is not platform-specific which allows reliable and realistic validation of wide range of algorithms and platforms [62]. The packet loss is computed according to collisions as well as comparing the energy level of the received packet to the noise power of the environment. My simulation implementation complies with 802.15.4-2006 standard, and allows data transfer during CAP period where nodes only use slotted CSMA/CA. In the slotted CSMA/CA, a node needs to wait for a random backoff slots to transmit data packet but the acknowledgement packet does not have to use slotted CSMA/CA. During the CFP period, the coordinator sends ACK packets after each successful data packet transfer. IEEE 802.15.4-2006 standard describes how the superframe intervals must be calculated. The active + inactive period must be equal to  $BaseSuperframeDuration = NumberOfSuperframeSlots \times symbolTime$ . *SymbolTime* is calculated as

$$\frac{1}{physicalDataRate \times 1000 / physicalBitsPerSymbol}$$

and then *BeaconInterval* is calculated as  $BaseSuperframeDuration \times 2^{BeaconOrder}$ . Here, *BeaconOrder* is a constant and is equal to 6 in my simulations. The active portion of the superframe is  $ActiveInterval = BaseSuperframeDuration \times 2^{ActiveOrder}$ . I have chosen 4 as my *ActiveOrder* constant. Also, the number of time slots assigned to CAP period needs to be specified in order for the CAP length to be calculated. I set the CAP period to 2 GTS 

---

 this duration as well as the transmission delay are included in  $D_0$  to ensure feasibility.

long. The transition cost in terms of energy and delay between RX, TX, and Sleep states are also included.

- $MaximumNumberOfTries\_CAP = 4$ ,  
 $MaximumNumberOfTries\_CFP = 2$ ,  
 $guardTime = 1ms$
- Clear Channel Assessment: IEEE 802.15.4-2006 specifies three modes of performing CCA. Castalia's radio module is built to provide *Mode 1* which checks whether the measured energy is above a threshold value or not. The default time duration to measure the energy level is set to 0.000128 s which is independent of the radio that is being used. The default value of energy threshold is -95 dBm.
- Transition costs: For the most part, I only consider the lightest sleep level which consumes 0.5mW. The list of transition costs are; RX to TX = 32 mW, TX to RX = 32 mW, RX to Sleep = 1.4 mW, Sleep to RX = 1.4 mW, TX to Sleep = 1.4 mW, and Sleep to RX = 0.5 mW. Other sleep levels are also considered and reported in Section 5.8.
- Transition delays: Once again only with the lightest sleeping mode the transition delays are; RX to TX = 0.01 ms, TX to RX = 0.01 ms, RX to Sleep = 0.05 ms, Sleep to RX = 0.194 ms, TX to Sleep = 0.05 ms, Sleep to RX = 0.194 ms are taken from CC2420 radio.
- Modulation level parameters:  $DataRate(kbps)$  is calculated as  $symbolRate * bitsPerSymbol$  where  $symbolRate$  is constant and 62500 for 2450 MHz radios such as CC2420. Bandwidth,  $noiseBandwidth$ ,  $noiseFloor$ , and Sensitivity values are taken from CC2420 radio and are 20 MHz, 194 MHz, -100 dBm, and -95 dBm respectively. TX\_dBm levels which effects packet loss, CCA, and neighboring problem (see Section 4.1) are 5, 8, 11, 14, 17, 20, 23 dBm respective for modulation levels 2 to 8.

All the proposed algorithms run on the coordinator. The computed modulation levels are

transmitted to the nodes using beacons and preambles. Hence, the computation overhead at the nodes is minimal. I set the preamble size to 14 bytes, which includes a 1 byte for sender address, a 1 byte for receiver address, 10 bytes for the calculated modulation levels, and 2 bytes for the CRC footer. For the beacon messages, the generic 802.15.4 beacon structure is adopted. *Static*, *Static\**, and *Oracle* algorithms uses 100 bytes for the assigned modulation levels for each possible packet and an additional 10 bytes for time slot allocation in *GTS fields* of the beacon message. *Dynamic*, *Dynamic\**, and *Dynamic-f* algorithms use 10 bytes for the scheduling order, 10 bytes for the modulation levels for the first scheduled node in the same field.

All the dynamic algorithms require same amount of signaling. Furthermore, the coordinator is to use the maximum modulation level to broadcast the preamble. In my simulation settings, the *listening-duration*  $\gamma = 0.000128$  s and the *sleep-duration*  $\alpha = t_{preamble} - \gamma$ . For the  $C_e$  and  $C_s$  values described in Section 4.2,  $15 \times 10^{-9}$  and  $12 \times 10^{-9}$  Joules are selected, respectively, and  $b_{min} = 2$ ,  $b_{max} = 8$ , after [14, 58]. All the simulation results are presented at 95% confidence level. In all the plots presented, the energy consumption values of various schemes are normalized with respect to the energy consumption of *Static* at *load* = 1.0.

### 5.3 Analysis of the Ideal Case

This section evaluates the proposed algorithms' ideal case performances. In *ideal case*, the nodes have exact knowledge about the time at which they need to wake up, in advance. The static algorithms can incorporate this information in the beacon message. For dynamic algorithms the same beacon message structure is also assumed. Obviously ideal case is not realistic and the need for low-power listening disappears. Still, the analysis of this case reveals some important patterns because it points to the upper bounds on the energy savings that each algorithm can provide with zero-overhead low-power listening.

Figure 5.2 shows the normalized energy consumption of the proposed algorithms. It is observed that on higher load values the *Dynamic*, *Dynamic\**, and *Dynamic-f* algorithms give

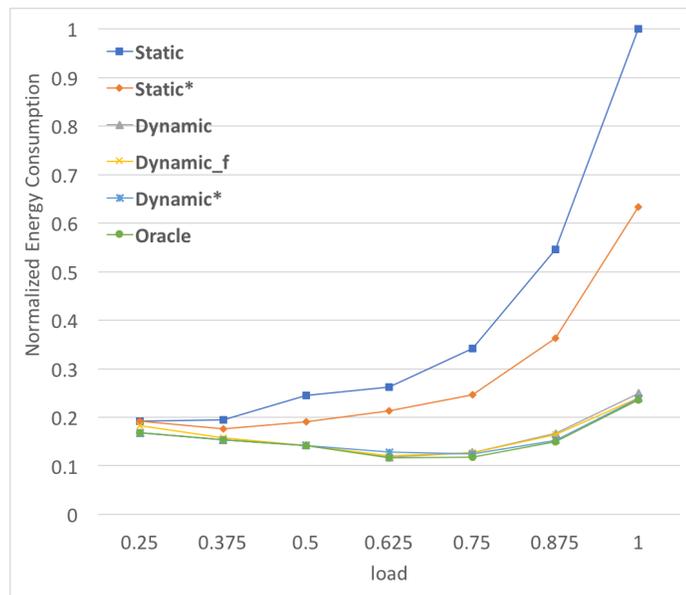


Figure 5.2: Energy consumption in the ideal case.

Here, I assume dynamic algorithms hypothetically know the exact time they need to wake up in order to start transmitting. Hence, LPL is not needed.

significant energy savings compared to *Static* and *Static\** algorithms. Moreover, *Dynamic* and *Dynamic\** perform better than *Dynamic\_f*. However, at lower load values, the dynamic algorithms provide only limited gains; this is because even the static algorithms are able to assign low modulation levels when the system has ample time to finish the workload.

It is also seen that the energy consumption minimizes for the load value 0.625; for the load values smaller than 0.625 the sleeping energy consumption becomes dominant and for the load values greater than 0.625 the transmission and reception energy consumptions becomes dominant.

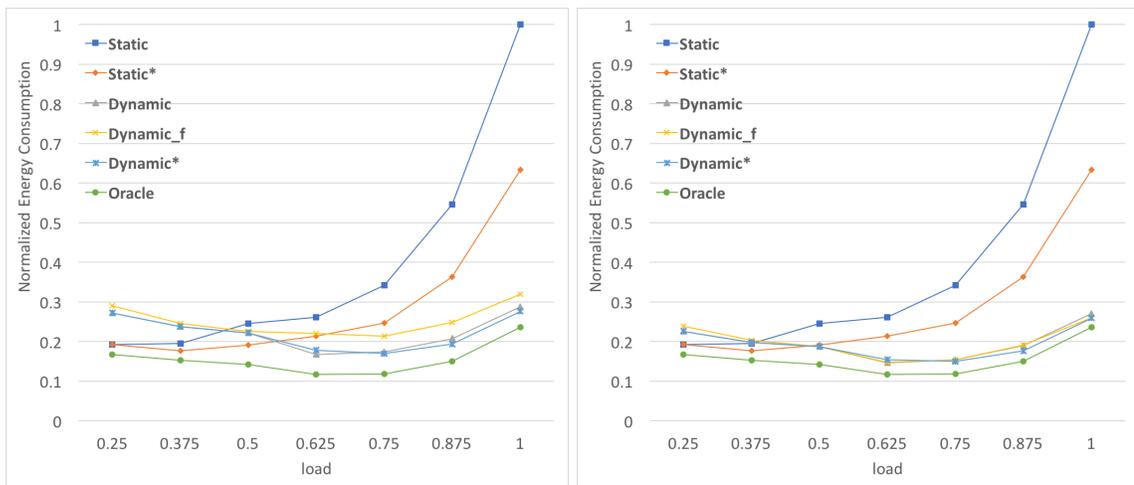
## 5.4 Analysis of the effect of traditional LPL with no interference

This section shows the effect of low-power listening on the proposed algorithms. The ideal case where the nodes know exactly when the previously scheduled node stops transmitting

cannot be implemented in real-life scenarios. The nodes need to listen for a preamble from the coordinator to see when they can start transmitting. One possibility is to use the traditional low-power listening (without the HLPL enhancement described in Section 4.1) and the results in this section consider this case, by further assuming that the cross-node interference is negligible. In Section 5.5, I will re-analyze these settings within the HLPL framework by considering the impact of the interference.

Figure 5.3a shows the normalized energy consumption of greedy low-power listening enabled algorithms. The compared algorithms are *greedy* in the sense that they use traditional low-power listening with the wait-duration  $\delta$  set to zero. Please recall that only *Dynamic*, *Dynamic\**, and *Dynamic-f* require low-power listening. The remaining algorithms have pre-determined wake-up times. We can see that the dynamic algorithms perform poorly compared to static algorithms when  $load \leq 0.6$ . This is due to the fact that for lightly loaded systems, the gain from dynamic reclamation of the slack times is offset by the additional energy consumption due to energy overhead of traditional low-power listening activity. The dynamic algorithms' energy performance improves only when  $load$  approaches and exceeds 0.6 (this threshold is slightly larger for *dynamic-f*) – this is when the overhead of low-power listening (necessary to implement the reclaiming mechanism) becomes reasonably low compared to the gains of adaptive modulation downscaling at run-time. It is also observed the energy consumption gap between dynamic algorithms becomes more significant where *Dynamic* and *Dynamic\** performs very closely and outperforms *Dynamic-f*.

Another possibility for the implementation of the traditional low-power listening in these settings is to have each node wait for a time duration  $\delta$  equal to the *expected time needed for the completion of the packet transmissions by the previous nodes*. The idea is to take advantage of the known probability distribution. Rather than letting nodes start low-power listening as soon as the collision-avoidance-period starts, the nodes calculate the expected number of packets that will be transmitted by the previously scheduled nodes based on the known probability distribution function. I call this scheme *smart-LPL*. The expected-number-of-packets before  $node_i$  can start to transmit is  $\sum_{k=1}^{i-1} \sum_{l=1}^{m_i} p_k(l) \times l$ . The



(a) Energy consumption of greedy-LPL where  $\delta = 0$

(b) Energy consumption of smart-LPL where  $\delta = \text{expected-wait-time}$

Figure 5.3: Simulation results with no interference.

It is assumed each node only hears the coordinator and none of its neighbors. Hence, the neighborhood problem described in Section 4.1 hypothetically does not exist in this settings.

scheduling order is embedded into the beacon message. Two observations are in order here: *i*) if the node starts low-power listening before its actual turn then the node spends more energy for low-power listening but does not miss any of its slack time. However, if the node wakes up after its turn starts then the node loses some portion of the given slack time (the node could not reduce its modulation levels as much as it could have) but spends less energy on low power listening. Hence, there is a trade-off between the gain from low-power listening and loss from smaller slack times. *ii*) The modulation level assumed in the calculation of the expected-wait-time for the previous nodes is another critical variable. The nodes know the expected number of packets to be sent before their turn, but they do not know what modulation levels have been used by the previous nodes (they cannot accurately map the expected number of packets to the expected amount of waiting time). In my simulation settings, the modulation level calculated by *Static* is used to compute the expected-wait-time values.

Figure 5.3b illustrates the normalized energy consumptions with smart-LPL. This analysis reveals that with smart-LPL, the energy consumption of the dynamic algorithms is reduced compared to the greedy-LPL case. One important observation here is the effect of low-power listening on the dynamic algorithms. The increased gap between dynamic algorithms observed in Figure 5.3a becomes less significant in Figure 5.3b. This is a result of two factors: *i)* shorter GTS slots means longer duration to listen for a preamble and hence leads to higher overhead created by low-power listening; *ii)* For the case shown in Figure 5.3b, the expected-wait-time values are calculated using the modulation level given by the *Static* algorithm. However, higher ideal case performance implies that the previously scheduled nodes have used smaller modulation levels than initially computed. This leads to less accuracy in predicting expected-wait-time and as a result, a longer duration for traditional low-power listening. This is a crucial result that shows how dynamic modulation levels can affect low-power listening and becomes one of the fundamental reasons necessitating the use of HLPL protocol.

## 5.5 Analysis of the impact of neighborhood/interference

In this section, the aim is to evaluate the effect of neighborhood/interference on traditional low-power listening and also include my newly proposed HLPL in the comparison. In real settings when a node wakes up to check for a preamble, it has to listen to its neighbors' communications to make sure that the communication it is sensing is not a preamble. In order for a node to make sure that it is not receiving a preamble, it may need to listen the channel for up to two preamble transmission times as shown in Figure 4.3.

Figure 5.4a shows the normalized energy consumption with traditional low-power listening and possible interference. A striking observation is the significantly increased energy consumption of the dynamic algorithms for most of the spectrum, due to the prohibitive energy consumption of *false alerts* induced by the interference due to the naive application of the traditional low-power listening framework. In this case, the nodes receive *false alerts* from their neighbors and they need to verify the content of these transmissions. The

length of preamble message is 14-byte long whereas the MTU of 802.15.4 is 127 bytes. This indicates that even for a single packet with the highest modulation level, the node has to consume an additional energy of listening up to 2/3 of a packet (which is 84 bytes) to see if there is or there is not a preamble addressed to itself. In a neighborhood of size 4, this may create an additional overhead up to 26 packets per node as can be seen from Figure 5.4a.

The overhead created by the interference also depends on the values used for sleep-duration and *preamble size*. In my simulations, I have observed that larger *sleep-duration* values tend to decrease the overhead induced by the interference. However, longer sleep-duration has other consequences such as longer superframe lengths and larger losses in the available slack times. In order to ensure the deadlines, the maximum time a node can miss before it hears a preamble has to be accounted for. This maximum time needs to be added to the minimum feasible superframe length to ensure the feasibility of the system. Longer superframe lengths improve the performances of the static algorithms as well as those of the dynamic ones so the gap between both classes of algorithms remains more or less the same. Also, longer superframe lengths cause a larger delay between the two consecutive turns of the same node.

Some optimal values of preamble length and sleep-duration values that will minimize this overhead may exist. However, I believe even this minimized overhead it will still be undesirable especially for lower utilization factors where *offline* algorithms perform well. Finding this minimized overhead value is left as a future work.

Figure 5.4b shows the simulation results obtained after adopting greedy-HLPL. Comparing to Figure 5.4a, one can see the drastic energy savings provided by the greedy-HLPL. *Dynamic* and *Dynamic\** outperform the *Static* algorithm for load values higher than 0.52. For load value 0.6 and higher, It is observed that *Dynamic* and *Dynamic\** have less energy consumption than *Static\**. *Dynamic\_f* outperforms *Static\** for load values roughly after 0.91. If we compare Figure 5.4b with Figure 5.3a, we can see that the performance of *Dynamic* and *Dynamic\** algorithms in the presence of interference is rather close to the one in the no-interference case where *Dynamic\_f* results in a more significant increase. Figure 5.4c

shows the normalized energy consumption of smart-HLPL when wait-duration is equal to expected-wait-time. This case further reduces the overall energy consumption of dynamic algorithms. In this case, *Dynamic* outperforms the static algorithms for load values roughly larger than 0.45.

As it can be seen, HLPL successfully addresses the neighborhood/interference problems and yields significant energy savings. Finally, the comparison of Figure 5.2 with Figure 5.4c shows that the results of HLPL are reasonably close to the ideal case, showing the potential of the framework.

## 5.6 Effect of different probability distribution functions

All the results presented so far were obtained under Uniform probability distribution for the packet workload. The simulations are also repeated with Normal ( $\mu = 5$  and  $\sigma = 2$ ), Pareto and Flipped-Pareto distributions  $k = 10$  (shape parameter),  $\sigma = 3$  (scale parameter), and  $\theta = \frac{10}{3}$  (threshold value).

An important difference is in terms of the average energy consumption of different distributions. The simulation results show that the Pareto distribution has the lowest average energy consumption followed by Normal, Uniform, and Flipped-Pareto distributions. This is expected due to the fact that each distribution function has different expected workload figures which are 3.22, 5.04, 5.5, 7.78 for Pareto, Normal, Uniform, and Flipped-Pareto distributions, respectively.

Figure 5.5 shows the energy consumptions for each probability distribution function. Several conclusions can be drawn here: *i)* Distribution functions had limited impact on the results presented in previous sections; the ordering of the algorithms is still the same for each of the tested probability distribution functions. *ii)* For all the cases analyzed in previous sections: the gap between the average energy consumption values of the algorithms got smaller for Pareto case. The *dynamic* algorithms have performed very close for these load values greater than 0.5. Fewer number of packets led to limited difference in transmission

energy consumption. For similar reasons, this gap became larger for Flipped-pareto distribution function. We can say that when the nodes have higher workloads, the performance gaps between *dynamic* algorithms get larger and for the cases where the nodes have lower workloads, the gap between *Dynamic\_f* and *Dynamic\** as well as the gap between *Dynamic\** and *Dynamic* get smaller. *iii)* In the Pareto distribution case, *Dynamic*, *Dynamic\** and *Dynamic\_f* outperformed *Static\** for the load value roughly 0.5. These values are slightly lower than the results presented in previous sections.

## 5.7 Analysis of scalability

The scalability of HLPL in terms of number of nodes, and number of packets is analyzed. For the number of nodes case, I have conducted simulations from 1 to 20 nodes each with uniformly distributed workload of 10 packets and for the number of packets case, 10 nodes with uniformly distributed 1 to 20 packets of probabilistic workload. Figures 5.6 and 5.7 show the scalability in terms of number of nodes and packets. The linear regression analysis shows that for all of the scheduling algorithms except *Dynamic* and *Dynamic\** in Figures 5.7a- 5.7b, the average energy consumption grows linearly with respect to number of nodes and number of packets where each of the regression analysis had an R-squared value of 0.96 or higher. For the mentioned *Dynamic* and *Dynamic\** results, a 6<sup>th</sup> degree polynomial regression had R-squares value of 9.95 or higher. The general ordering of the algorithms has not changed in terms of number of nodes. However, I observed that *Static\** outperforms *Dynamic\_f* for 14 packet workloads.

## 5.8 Impact of Radio Hardware Variations

This section discusses the effect of different values of radio unit power consumption. First a series of experiments with sleeping-power consumption of 0mW and 3mW was run. In the previously reported results, this value was set to 1.4mW. Figure 5.8 shows the effect of sleep

power consumption with 10 nodes, maximum of 10 packets workload with uniform distribution. Figure 5.8a is the ideal case (described in Section 5.3) with sleep power consumption is set to 0. Here, I observe a strict increase in energy consumption with increasing load value. This result is different that the one shown in Figure 5.2 which has the minimum energy consumption at the load value of 0.625. This is because lower load values mean longer superframe duration and hence increased energy consumption from sleeping. When we take sleeping energy consumption out of the equation (recall that ideal case does not require low-power-listening), higher load values *strictly* results in higher energy consumptions due to higher modulation levels. Figure 5.8b shows the energy consumption with greedy-HLPL when sleep power consumption is set to 0. Comparing this with Figure 5.4b, we see the energy consumption gap between the highest and the lowest load levels increases. Figure 5.8c shows the case where sleep power consumption is set to 3 mW. The lowest load level results in the highest energy consumption except for *Static*. For the other algorithms, the sleeping energy consumption is higher than the energy savings from using lower modulation levels. This case also shows that the *Dynamic* and *Dynamic\** outperforms static algorithms for every load value which further emphasizes the effectiveness of HLPL.

Next, a set of experimented with CC2420 based power consumption values was conducted. The results are presented in Figure 5.9. CC2420 only has a single modulation level of 4, which consumes 62 mW. If we assume the same exponential increase of initial test values, the transmission/reception power consumption can be estimated to be 15, 31, 62, 124, 248, 496, 992 mW for modulation levels 2, 3, 4, 5, 6, 7, 8 respectively. Figure 5.9a shows the energy consumption results for greedy-HLPL. These settings shows using *Static\** consumes less energy for every load value hence, the best option. This is due to the very expensive clear channel assessment performed by the dynamic algorithms exceeding the energy savings from lowering modulation levels. However, Figure 5.9b gives significantly different results with smart-HLPL with only the lightest low-power-mode called *idle*. In this mode, CC2420 turns off its frequency synthesizer and draws 426  $\mu$ A of current. The results showed that *dynamic* and *dynamic\** outperforms *static\** for load values of 0.8 or higher

and *dynamic\_f* does so for the load value of roughly 0.9. Moreover, CC2420 has two more low-power-modes namely *power down* and *power off*. In *power down* the crystal oscillator is turned off in addition to frequency synthesizer and draws 20  $\mu\text{A}$  of current, and in *power off* the voltage regulator is also turned off with a total of 0.02  $\mu\text{A}$  current draw. It takes 0.6 msec to transition from *power off* to *power down* and 1.0 msec from *power down* to *idle*. I have assumed during these transitions there is a current draw equal to the level with the higher current draw value. All the algorithms are adjusted to consider all low-power-modes and put the nodes into the deepest one wherever expected-wait-time exceeds the respective break-even point. The results shown in Figure 5.9c indicates a significant improvement. The dynamic algorithms outperform the static ones for the load value of roughly 0.4 and greater. This further shows the benefits of smart-HLPL.

## 5.9 Hardware Test Results

To analyze the packet loss probability in an actual deployment, I implemented an *emulation* of DMS on Zolertia Z1 motes (that do not have DMS capability) using ContikiOS. Specifically, the transmission times that correspond to individual modulation levels are computed by scaling the transmission time that corresponds to the maximum modulation level which is 8. In general, the transmission time increases linearly with the modulation level, because the time to send a single bit is  $\frac{1}{b \times R_s}$  where  $b$  is the modulation size and  $R_s$  is a constant denoting the number of symbols transmitted per second. Based on this relationship, the packet loss rate is analyzed for different modulation levels in a series of experiments.

I have used 10 Z1 motes with a coordinator mote connected to a desktop computer which formed a star topology. I have implemented the emulated version of proposed *Static* and *Static\** algorithms on the conventional superframe structure as explained in Section 3. At the beginning of each superframe, the coordinator broadcasts a beacon which has the CAP start and end times. During the CAP, each node is allowed to request a GTS from the coordinator by registering their workload probability. The coordinator then computes

the modulation levels according to the corresponding static algorithm. These modulation levels and GTS start and end times were embedded into the beacon of the next superframe for every node. I have emulated the Normal probability distribution of the workload as presented in Section 5.

I had to do minor adjustments to the superframe length that was used in my simulation settings. I had to increase the superframe length for few milliseconds per node because the following bullets were more severe than the Castalia simulation settings: (1) the recipient time of superframe which acts as the basis for GTS start and end times were not exactly the same for every node, (2) the hardware delay between sending consecutive packets, (3) the timer skewness between motes. Hence, I had to wake the mote 2 ms before the start of the next superframe.

It is expected to have low packet loss during CFP due to minimized collisions. In my experiment, I have disabled the low-power listening for the coordinator mote. I aimed to analyze the packet loss as a function of transmission time which varies with the modulation level. In these experiments, retransmission during CFP is disabled and when the motes did not receive an ACK after their transmission, they stopped transmitting. I observed the probability of losing packet during CFP as 1.3%. Theoretically this ratio should be the same for any modulation level since in DMS we increase the energy to noise ratio according to the scaling function. However, I have observed a slight increase on the packet loss at reduced modulation levels, which shows that increased transmission times have a higher (and increasing) impact on the packet loss rate. The packet loss rates that are observed at different modulation levels are summarized in Table 5.1. In case of a packet loss, the motes add the missed packets into their future workload while making sure their workload do not exceed maximum workload to ensure the feasibility. This fact does not change the modulation levels for the static algorithms and the energy consumption increases proportionally by the corresponding packet loss rate.

If we directly map transmission time into modulation levels, we can conclude that the probability of losing a packet with modulation 8 is 1.3%, 7 is 1.34%, 6 is 1.4%, 5 is 1.48%, 4

Table 5.1: Packet Loss Rate at Different Modulation Levels

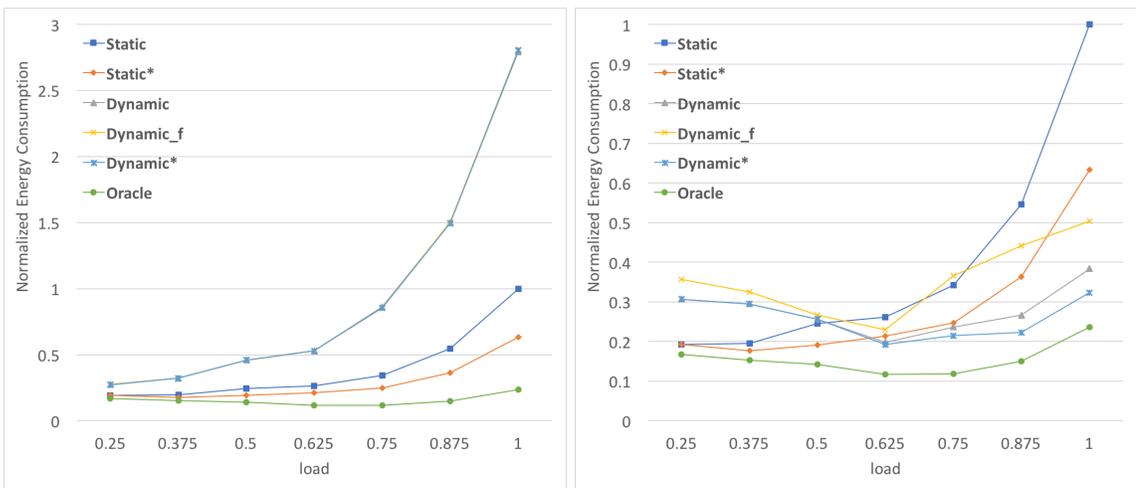
Modulation Level	2	4	6	8
Packet Loss Rate	0.017	0.016	0.014	0.013

is 1.6%, 3 is 1.66%, and 2 is 1.717. Same formula applies to *Static\** but the increase in the energy consumption is slightly less since it recomputes the modulation levels according to new probability distribution. The effect of packet loss is different on the proposed *dynamic* algorithms. When a packet loss occurs, the nodes has to update their workload probability distribution just as the *static* algorithms which increases the energy consumption for the current node. However, when a packet loss occurs the nodes stop transmitting which increases the slack time of upcoming nodes. This will lead to reduced energy consumption for the upcoming nodes. As a result, I do not expect packet loss to alter my results for *dynamic* algorithms under these assumptions.

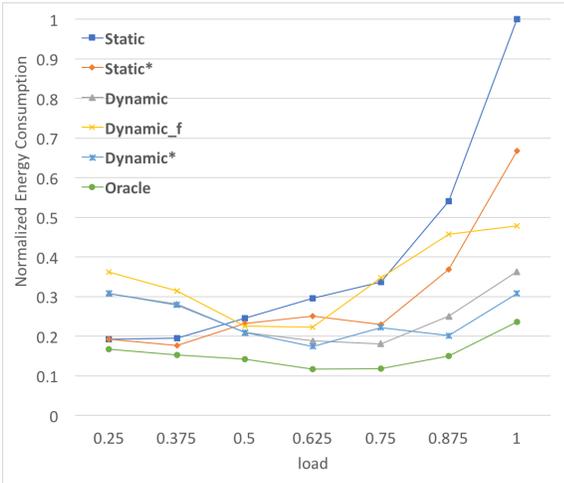
## 5.10 Conclusions

My preliminary work addressed the problem of ensuring real-time guarantees while minimizing the overall energy consumption in wireless sensor networks. I focused on cluster-oriented superframe communication, the most widely adopted method for providing real-time guarantees in industrial wireless networks. Using Dynamic Modulation Scaling I studied static and dynamic algorithms for reallocation of slack times of a superframe. I analyzed the effect of interference and dynamic modulation levels on low-power listening. I also introduced a new low-power listening protocol called *hybrid-low-power listening* (HLPL) in order to overcome the interference problem caused by neighborhood. Using the Castalia Simulator, I empirically assessed the performance improvements of DMS slack reclaiming and HLPL. My experiments show that dynamic slot readjustment saves a significant amount of energy under highly loaded systems. They also indicate that HLPL overcomes the interference caused by other nodes in the cluster and significantly reduces the overall energy consumption of

the system.

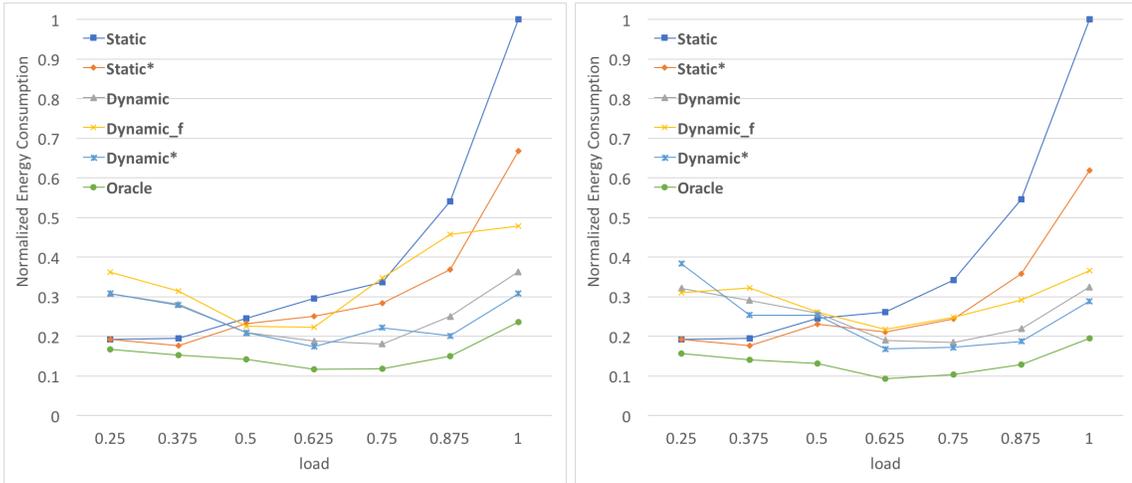


(a) Energy consumption with greedy-LPL and interference where  $\delta = 0$  (b) Energy consumption with greedy-HLPL and interference where  $\delta = 0$



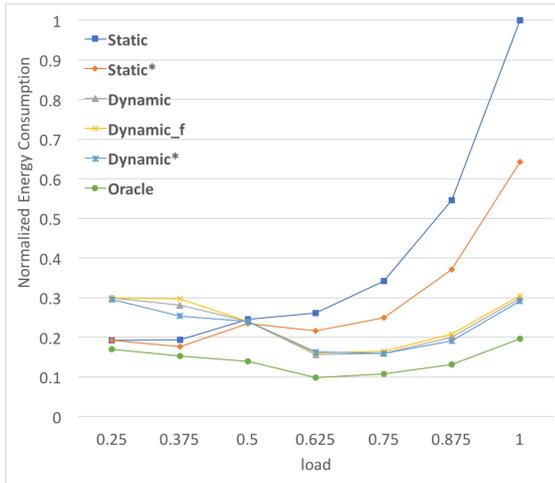
(c) Energy consumption with smart HLPL and interference where  $\delta = \text{expected-wait-time}$

Figure 5.4: Impact of interference



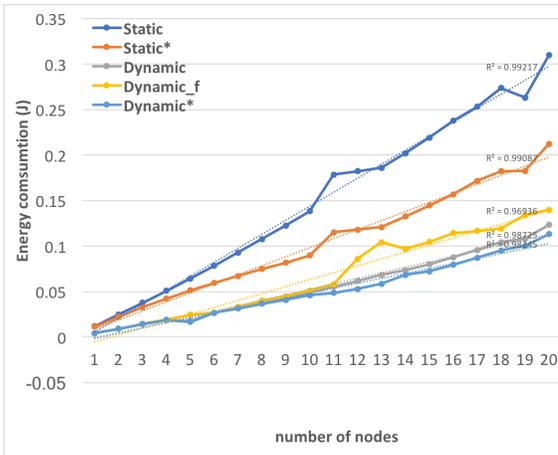
(a) Flipped-Pareto distribution

(b) Normal distribution

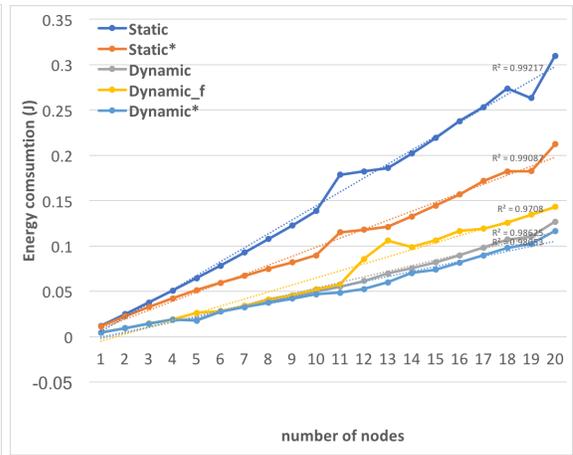


(c) Pareto Distribution

Figure 5.5: Energy consumption of smart-HLPL with different probability distribution functions

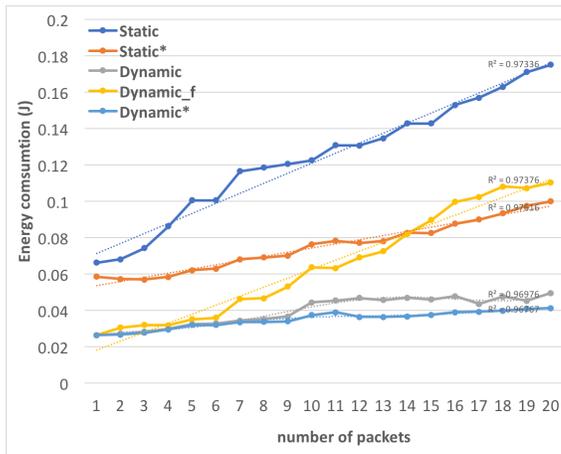


(a) number of nodes smart-HLPL

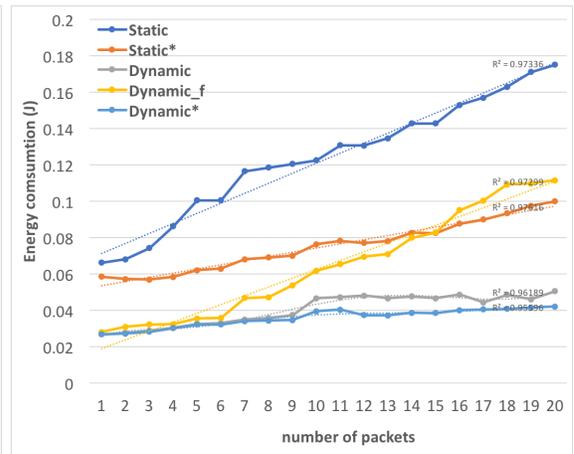


(b) number of nodes greedy-HLPL

Figure 5.6: Scalability in terms of number of nodes

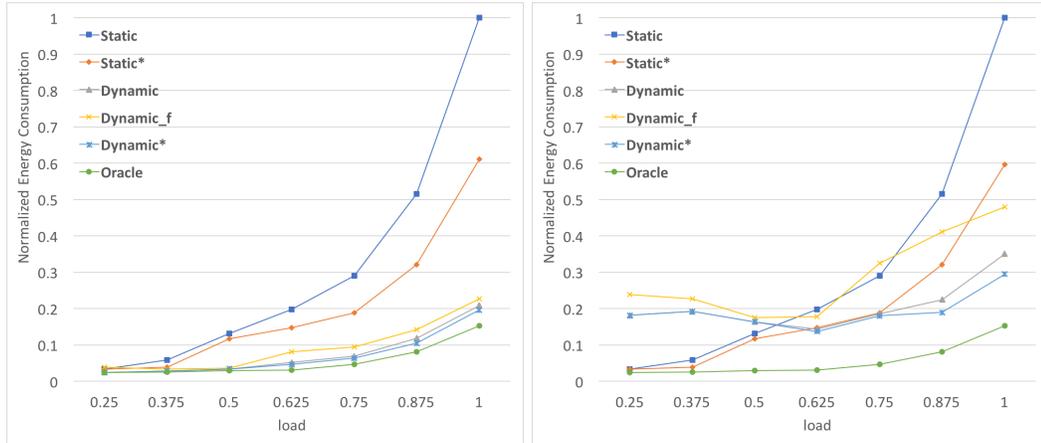


(a) number of packets smart-HLPL

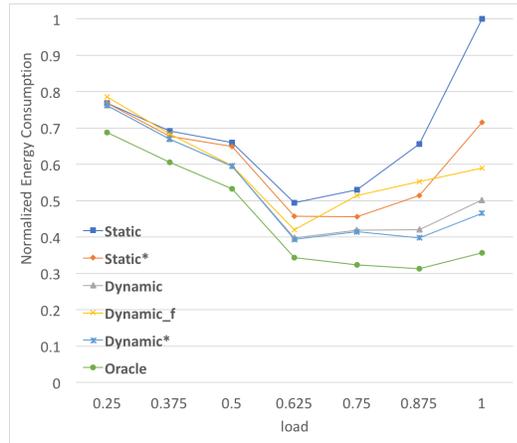


(b) number of packets greedy-HLPL

Figure 5.7: Scalability in terms of number of packets

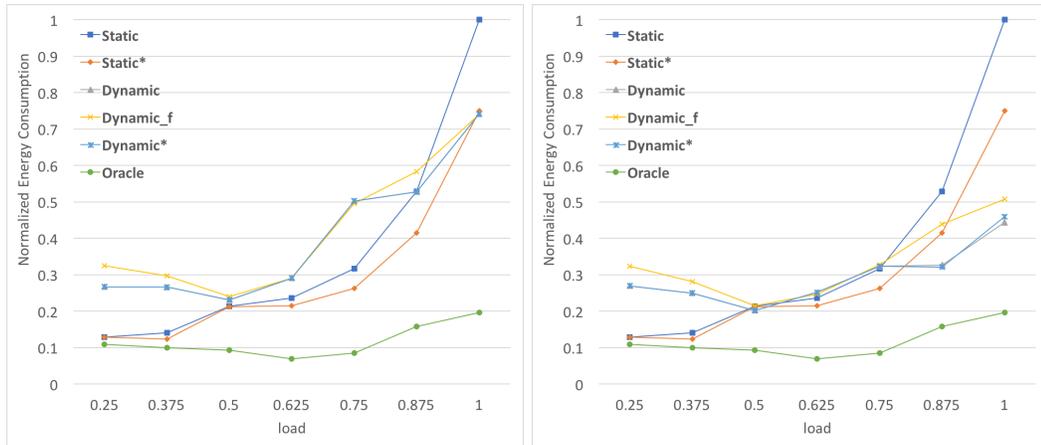


(a) Ideal case with sleep power = 0 mW. (b) Greedy-HLPL with sleep power = 0 mW.

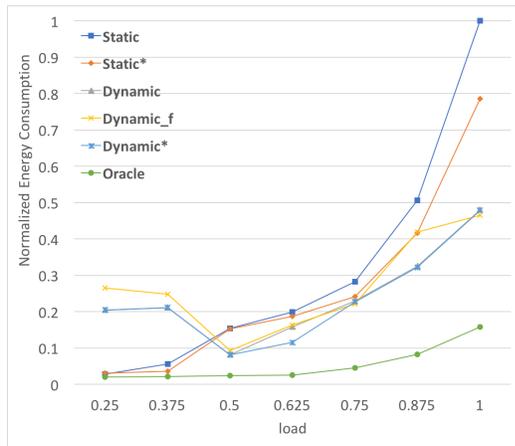


(c) Greedy-HLPL with sleep power = to 3 mW

Figure 5.8: The effect of sleep power consumption.



(a) Greedy-HLPL with lightest sleep level (b) smart-HLPL with lightest sleep level



(c) smart-HLPL with multiple sleep-levels

Figure 5.9: Energy consumption with CC2420 based power consumption settings

## Chapter 6: A Software Defined Radio Analysis of the Impact of Dynamic Modulation Scaling within Low Power Wireless Systems

On the medium-to-low data rate applications, IEEE 802.15.4 is one of the standards that define physical and data link layers for low-power networks. It is widely adopted by Wireless Sensor Networks (WSNs). Despite its inherent potential for energy savings and hence increased sustainability, DMS is not part of IEEE 802.15.4 and there does not exist any off-the-shelf commercially available IEEE 802.15.4 compliant DMS-capable radio..

A detailed description of existing work on DMS is presented in Section 3.1.1. In particular, DMS has been the subject of several research articles in the Wireless Sensor Network (WSN) domain. However, due to the lack of commercially available hardware, these evaluations were typically conducted via simulation and/or using the pre-supposed closed-form energy scaling formulas. **My dissertation aims to fill the gap in the empirical analysis of DMS when applied in the WSN domain.** I am particularly interested in investigating the following question: Given a set of environmental conditions, modulation techniques and power levels, how do the packet delivery rates change with DMS?

I have used the Ettus B210 Software Defined Radios [63] (SDRs) and configured them according to the IEEE 802.15.4 base band, symboling rate, data rate and samples per symbol settings. Working on this configuration, I make the following contributions. First, I show how to emulate DMS in low-power networks using existing GNU Radio blocks and SDR hardware. I also provide a detailed look into the signal recovery process which may be used for future researchers to extend my work. The next three contributions model various types of environmental transmission conditions which may affect wireless systems. I vary the Signal-to-Noise-Ratios (SNRs) of channel within  $\{2, 4, 8, 16\}$ -PSK and DPSK

modulations to assess how packet delivery rates (PDRs) change. This experiment gives a detailed insight on how the output power can be managed to achieve the desired PDR. These tests required a controlled noise environment with a noise generator, and therefore was conducted using Faraday Cages. I compare my findings with what existing general theoretical models suggest. I conclude that the necessary energy increase is greater than what the models suggest when I increase the modulation levels. Furthermore, I repeat the same experiments with Differential-PSK and report a very significant increase in performance. I then conduct a set of distance tests for  $\{2, 4, 8\}$ -DPSK and measured PDRs for distances up to 100 meters. Distance tests are of utmost importance for multi-hop networks and provide great insight on how it can be used to set up more energy- and/or latency-aware topology control. Finally, I test the impact of elevation difference between transmitter and receiver which is a common scenario for applications such as residential sensor monitoring, where some nodes are placed on tall objects.

## 6.1 Experimental Methodology

Software Defined Radio (SDR) experiments are performed on the Ettus USRP B210 radio [63]. SDRs differ from traditional radios by implementing various components such as amplifiers, modulators/demodulator, filters on a software layer rather than hardware. This offers higher configurability and flexibility.

The software I used is built using the GNU Radio, an open source platform that comes with a number of reusable *blocks*. Figure 6.1 shows our transmitter and receiver implementation. PSK modulation and demodulation blocks were already implemented and are explained below. The data source is a text file consisting of randomly generated ASCII characters. The subsequent blocks shown in Figure 6.1 are needed to packetize the data from the source file. I have chosen 128 bytes as the packet payload length in order to comply with the IEEE 802.15.4 standard. Packetizing and tagging the input flow allow the receiver to distinguish between data from noise and other communications. I have also

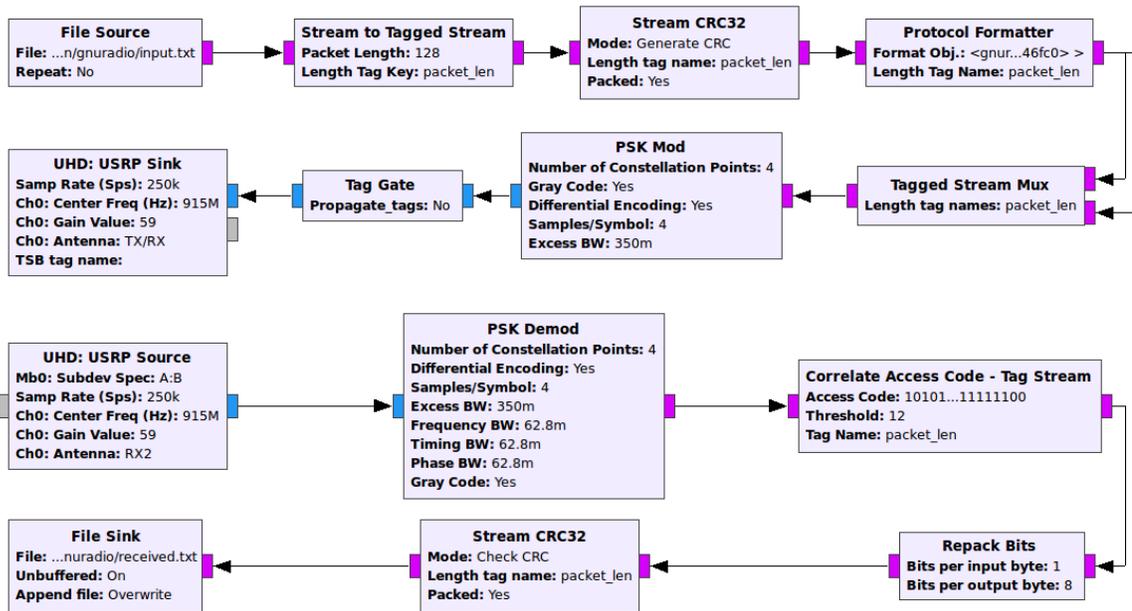


Figure 6.1: Implementation of transmitter and receiver in GNURADIO



Figure 6.2: Steps of signal decoding

appended a 32-bit CRC to make sure we consider only successfully received packets after the demodulation phase.

In general, in order to successfully demodulate a signal with PSK, there has to be a signal recovery process preceding demodulation as demonstrated in Figure 6.2. The PSK demodulation block already has signal recovery embedded in it with adjustable parameters. However, it only outputs the demodulated data, not the recovered signal. For the demonstration purposes, I have created a signal recovery process as described in [64].

Figure 6.3 shows the implementation of a signal recovery process with the existing GNU Radio blocks and only one of the many possible techniques. Figure 6.4 shows the

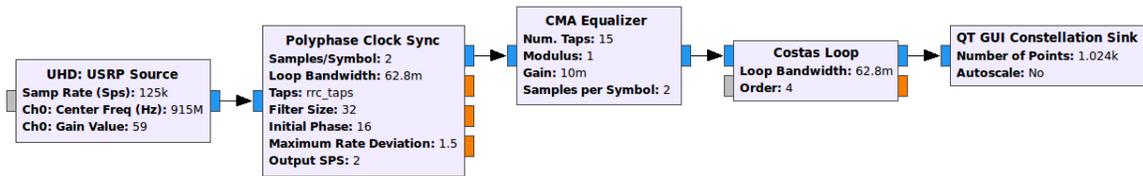
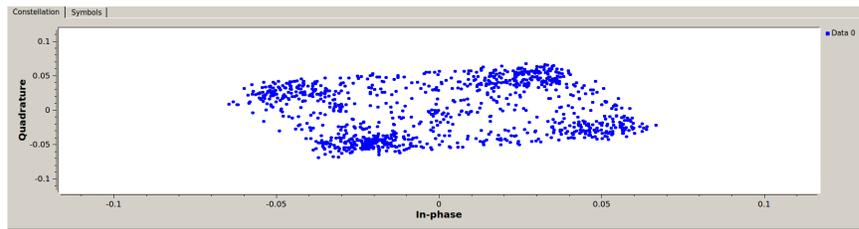


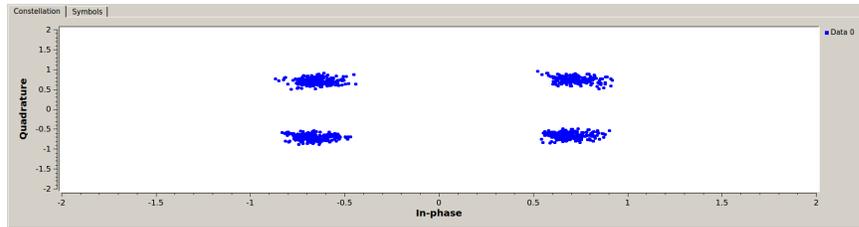
Figure 6.3: An example PSK signal recovery process

constellation display of QPSK-encoded signal before and after the recovery process. Here *Polyphase Clock Sync* is used for timing recovery, inter-symbol-interference, and down-sampling of sampling rate from 4 samples per second to 1 sample per second. The *CMA Equalizer* block is used to react to multipath fading and as the final step the *Costas Loop* block is used to correct phase and frequency offset. Even though this signal recovery process could successfully lock the constellation points, it may have locked them with a 0, 90, 180 or 270 degrees difference, a problem going forward with demodulation. There are two ways to fix this. The first is to have a predetermined *access code* and try each of the possible phase difference to see which one decodes successfully. The second is to use *Differential PSK* (DPSK) instead. DPSK only transmits the phase shift difference as reference to the previously submitted signal. Hence, no matter what the phase difference between recovered and transmitted signal is, the relative phase difference between consecutive phase shifts still holds.

Before I present the results, I would like to elaborate more on my experience with the aforementioned PSK demodulation block's performance, as potential hints for future testbed implementations. First, in order to have successful communication, I always had to start the sender first and then the receiver; doing the the other way around with significant delays prevented the receiver from recovering the signal. I have also observed that it is not always advisable to set the transmitter and receiver channel gain values equally. In some cases, especially for higher modulation levels, I have observed a lower receiver gain (compared to the transmitter gain) yielded higher performance especially for noisy environments. Last but not least, it is very helpful to set the PSK Demod block parameters as real-time configurable



(a) Received signal



(b) Recovered signal

Figure 6.4: Constellation display of received QPSK signal before and after recovery process.

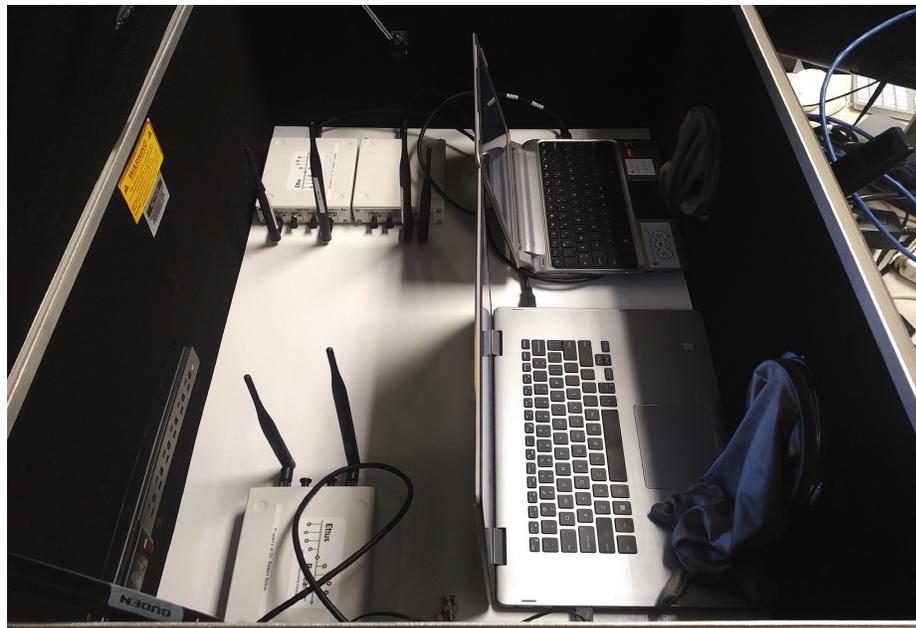


Figure 6.5: Faraday cage inside look

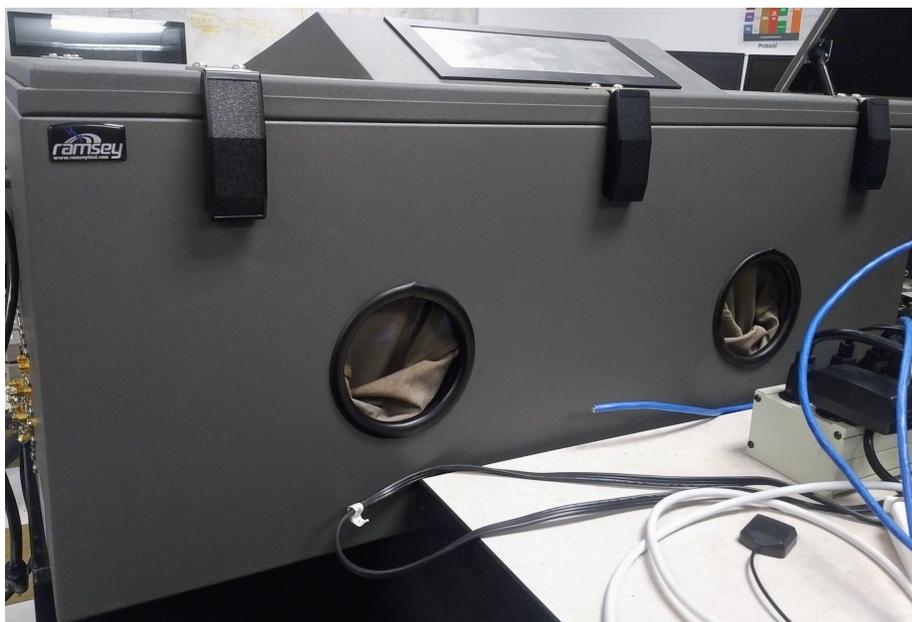


Figure 6.6: Faraday cage outside look

variables which can be tested on the fly and adjusted accordingly for each modulation level and current environmental conditions.

For my testing, I have configured our Ettus B210 Radios to use 915 MHz center frequency and 62,5 ksymbol/second symbol rate. I have also used 4 samples/second which gave a total of 250 kb/second data rate (for BPSK and QPSK 2 samples/second also worked well). The 915 MHz ISM band was chosen due to its higher output power levels compared to 2.4 GHz center frequency with the same output amplifier gain.

My primary objective was to obtain empirical indicators for energy consumption at different modulation levels. To achieve this, I have measured the necessary transmission power output with different noise levels to establish reliable communication. In order to have a noise-controlled environment and eliminate the effect of moving objects, I used Faraday Cages shown in Figure 6.5.

I have used a total of three Ettus B210 SDRs, one sender, one receiver, one noise generator. I have placed the sender and noise generator at equal distances from the receiver

to make sure signal and noise powers are equally affected by the distance (Figure 6.5).

### 6.1.1 Experimental results

In this section I explain the experiment setup and present the results. Using these results, I also provide some suggestions for future work in reliability and power management.

#### Signal-to-Noise-Ratio Testing

Most of the previous work involving DMS with low-power wireless communication tends to assume constant energy output levels per each modulation level and does not take into account the fact that output energy consumption can be controlled independent of the modulation level adjustments, in particular by varying the output amplifier gain. To better understand this issue, I have measured signal-to-noise-ratios (SNRs) of MPSK modulation and the corresponding packet-delivery-rate (PDR), for a given modulation level by varying the amplifier gain. Here higher SNR values correspond to higher transmission power consumption. My primary goal in this set of experiments is to analyze the energy increase required to have a reliable communication with different modulation levels and various output power levels using the MPSK modulation.

Each test involved sending 300KB of payload. In order to make sure the results were minimally affected by my design, I created a *control case* by averaging how much data I can decode without any noise using the second modulation level. Hence, I have compared the amount of the data received to this control case value to compute the reported PDR values. The control case was never able to receive all of the 300KB of data. The highest I could decode was 99.9%. This is because the signal recovery process takes a while and loses some of the initial data. This can be fixed by sending a data link layer preamble before the data packets or creating a more robust and optimized demodulation process. I believe the latter will be the case when DMS-compliant low-power radios are commercially available. Each PDR point is averaged over 100 experiments. Lastly, I have adjusted the noise level to at 10% of the maximum analog gain of Ettus B210 radios (max gain is 89.8

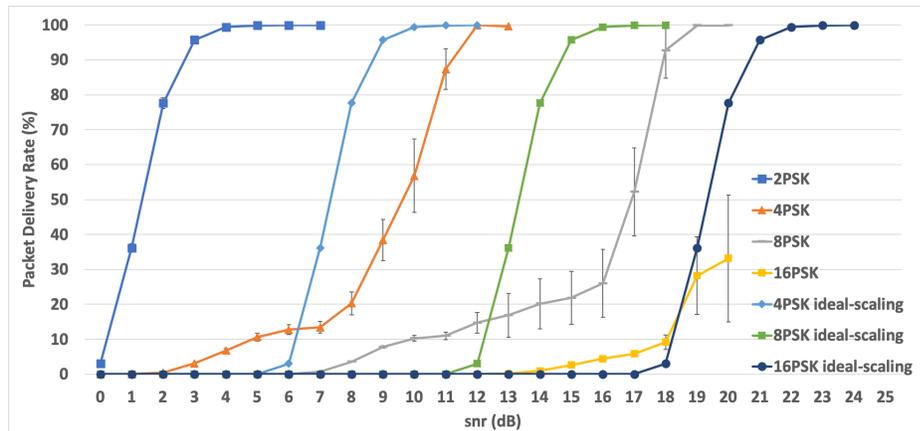


Figure 6.7: SNR to PDR comparison of empirical measurements and theoretical calculations for MPSK

dB) and adjusted transmission gains accordingly to create the desired SNR ratio. For the SNR ratios greater than 10 times, I have reduced the noise level. The receiver gain values varied across the different settings. Hence, the reported increases in transmission power are only for the transmitter but a close approximation for the receiver.

Figure 6.7 shows the results obtained for  $\{2, 4, 8, 16\}$  PSK. In addition, I have also plotted the theoretical values obtained for  $\{4, 8, 16\}$  PSK using the scaling function reported in Table 3.1 by using 2PSK as the base case. We can see a marked difference between the theoretical predictions and empirical measurements. In order to have at least an average PDR of 99% 2PSK required an SNR of 4. For 4PSK, this value increased by a factor of 6 (to 12). However, the scaling function suggests only an increase of 4 times. 8PSK required an SNR of 16, an increase of 5 times over 4PSK, which is also higher than what the scaling function suggests. Unfortunately for 16PSK case, the best I could received was an average of 33% PDR (I was unable to pass 35% even in the absence of noise). This may be due to not having enough output power level and as well as low performance of signal recovery process of built in PSK demodulation for higher constellation sizes also the noise generated by the transmitting SDR and the internal noise of the receiving SDR's hardware. However, if we compare this 33% PDR value to 8PSK, a 2.5 times increase is observed, once again a

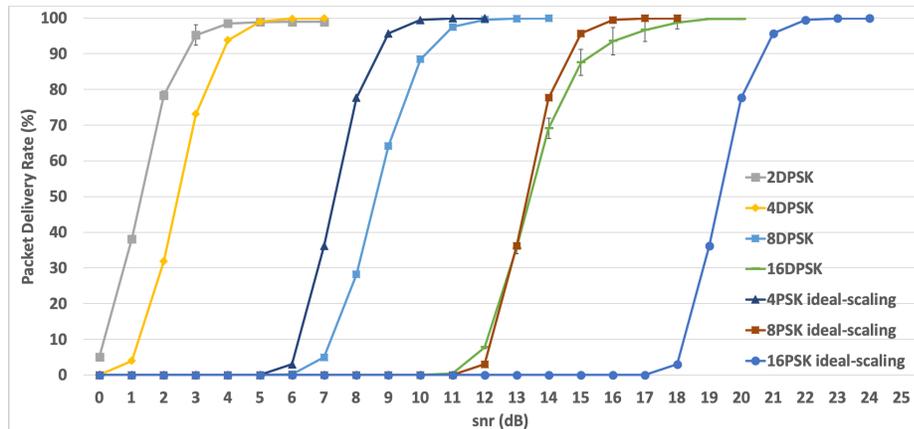


Figure 6.8: SNR to PDR comparison of empirical measurements and theoretical calculations for MDPSK

value lower than what the scaling function suggests.

Next, I repeated the same experiments with Differential-PSK (DPSK) modulation. It has been shown that in communications over fading channels DPSK is preferable over PSK. DPSK uses the previously transmitted symbol interval's phase as a reference to decode the current symbol interval. In channels with slow condition variations compared to the symboling rate, DPSK gives good performance [65] [66].

Figure 6.8 validates the better performance of DPSK over PSK. The results of 2DPSK and 2PSK were very close and both required an SNR value of 4 to have at least a 99% delivery rate. The major difference is observed for  $\{4, 8, 16\}$  DSPK. 4DPSK achieved 99% PDR for SNR = 5 and required roughly 1.25 times more output gain compared to 2DPSK. This value was 6 for 4PSK. 8DPSK achieved the same PDR for of 12 which is 5 times the output gain compared to 4DPSK. However, it only required 6 times the output gain of 2DPSK. Lastly, 16DPSK also required 5 times more output gain compared to 8DPSK.  $\{4, 8, 16\}$  DSPK all performed better than the theoretical PSK results as expected; 31%, 39%, 50% of the theoretical results, respectively. A final conclusion can be drawn from both Figure 6.7 and 6.8; we can see that the scaling ratios between consecutive modulation levels vary and are not a constant value.

Translation from the results reported in Figure 6.7 and Figure 6.8 to the exact energy consumption of an 802.15.4 compliant radio will depend on the hardware design (the constants mentioned in Section 3.1.1), efficiency of the demodulation technique, as well as the channel access algorithm. As with many wireless standards, the 802.15.4 data link layer has two operating modes: The *beacon* mode and *non-beacon* mode. *Beacon* mode consists of two phases; *contention allowance period* (CAP) and *contention free period* (CFP). During CAP the nodes use slotted-CSMA to get an access to channel and in CFP the nodes have guaranteed time slots where the nodes take turn for channel access in time division multiple access fashion. *Non-beacon* mode is a simple CSMA access mechanism. For both cases the nodes assess the current noise level in the channel before transmitting. Since our reported results are based on SNR values, lower noise levels will require lower output power. However, the increase in power consumption between different modulation levels will remain valid for a given noise level.

It is now time to give a real-world use case where reported SNR based energy consumption is quite useful. Protocols for Wireless Industrial Control Networks (ICNs) such as WirelessHART or ISA 100.11a require real-time packet delivery with application specific packet delivery rate requirements. Wireless ICN protocols typically uses time-division-multiple-access based channel access mechanism with primary and backup time slots for reliability [25–27]. In an application where a 98% PDR is the goal, SNR-based output power control provides a very fine tuned energy control mechanism. In this scenario rather than aiming for 98% or above delivery rate with a single transmission, aiming for a lower PDR rate and retransmitting in case of a packet loss may give a lower expected energy consumption. For example, if we are aiming for at least a 98% PDR using 16DPSK, we can either use an SNR value of 18 and transmit only once or use 15 (88% PDR) with a retransmission if fails. In the latter, we will have to retransmit with a probability of 12% which will give a total of 98.56% PDR. Using an SNR of 18 with a single transmission will consume 1.78 times more energy as oppose to the expected energy consumption of using 15 with retransmission. Integrating DMS into this scenario gives the application the ability to

not only fine tune its output power level but also take advantage of available transmission delay. An application can use a lower modulation level as long as the time slot length is large enough. Let us assume the primary time slot requires modulation level 16 whereas the backup slot can compensate for 8 (twice as long). Once again aiming for 88% PDR in both time slots for an expected PDR of 98% will consume 52% of a single transmission energy consumption.

I also suggest a slightly different perspective to the application of DMS in the low-power wireless domain. The primary focus of DMS-enabled low-power protocols has been energy management, and DMS is utilized as a tradeoff between time and energy while keeping PDR constant. In other words the goal is to manage output power while achieving the same PDR. However, it is also possible to view DMS as a *reliability* mechanism that provides a tradeoff between time and reliability. In other words, DMS can be used to keep the output power constant while reducing the modulation level in order to achieve the same or higher PDR. For example, let us assume we have a sensor mote which is currently using 8DPSK and achieving 97% PDR with a SNR value of 11. Now assume that there is a sudden change in channel condition which reduced our SNR value to 7. Instead of increasing the output power to restore SNR value back to 11, the mote can decrease its modulation level to 4 and still achieve 97% and above PDR while using the same output power level as long as the latency requirements are not violated. Even in non-changing channel conditions where SNR ratio is fairly constant, if the current quality-of-service requirements of the application-layer protocol changes and requires a higher PDR, it is still achievable with the same transmission power by decreasing the modulation level rather than increasing output gain. I believe this type of DMS application is just as important and a viable direction for the future research of DMS in low-power wireless networks.

### **Distance Testing**

In the next set of experiments, I tested the PDR of M-DPSK in terms of distance. Distance testing is important due to its impact on topology control. In wireless domains, topology



Figure 6.9: The test area that was used for distance measurements. Sender and receiver are 100 meters apart.

control is mainly used to find an optimal subset of the nodes in the network to ensure connectivity. Notice that connectivity may have slightly different meaning in the low-power domain, so here I am referring to the ability that each sensing node can successfully communicate with the coordinator with or without using relay nodes. One important advantage of this approach is that nodes can better manage their sleep schedules [1]. DMS gives the ability to control the transmission distance without increasing the output power consumption but with a penalty of increased transmission delay.

Figure 6.9 shows the picture of the test area. I chose this location because it was very isolated and I measured very weak WiFi signals. Also, it allowed me to test up to 100 meters. The location of the receiver was chosen in such a way that the multi-path fading was minimal and it was located in the exact same spot for each test. The location of the sender was adjusted accordingly for each desired distance on a straight line.

My first experiment was conducted in the 2.484-GHz center frequency with the same parameters used for the SNR tests. Low-power radios such as CC2420 has 0dBm maximum transmission power. Hence, my goal was to set my SDRs to use 0dBm output power as well. However, SDR only allows us to configure its output amplifier gain (in dB) which produces different absolute transmission power levels depending on the center frequency being used. In order to get very accurate mapping of output amplifier gain to the absolute transmission output power, spectrum analyzer and highly sensitive power monitors are needed. I identified a forum post on the official National Instruments website by a SDR Product Manager of the company, which provided their internal test results of mapping amplifier gain to specific output power for different center frequencies using the Ettus B200 series SDRs [67]. According to their internal test results an amplifier gain between 61dB and 75dB corresponds to 0dBm output power. In order to pinpoint a specific gain value within this range, I have measured RSSI values with Zolertia Z1 motes set as sender and receiver equipped with CC2420 radios transmitting at 0dBm. Next, I have repeated the same test with the same Z1 receiver instead of Ettus B210 transmitter. An output gain of 70dB gave the closest RSSI values to match 0 dBm absolute transmission output power.

Figure 6.10 shows the results I obtained for  $\{2, 4, 8\}$  DPSK. I was unable to decode any signal using 16DPSK with 70 dB output amplifier gain. The error bars seen on the graph are 95% confidence levels. From these tests, I can conclude that increasing modulation levels decreases PDR at a specific distance.

I then repeated the same test with the 915-MHz center frequency. National Instruments internal tests indicate 70 dB amplifier gain corresponds to a little less than 10 dBm output power. Similarly, Texas Instruments sub-1-Gig low power radios have around 10 dBm maximum output power as oppose to 0 dBm of CC2420 2.4 GHz radio. As expected PDR has increased for each modulation level for each distance value significantly. So much that both 2 and 4 DPSK has maintained above 90% delivery rate for up to 100 meters. This increase is due to aforementioned increase in the output power level. For a good measure of comparison I ran tests with 915 MHz but this time with 60 dB amplifier gain. This gain

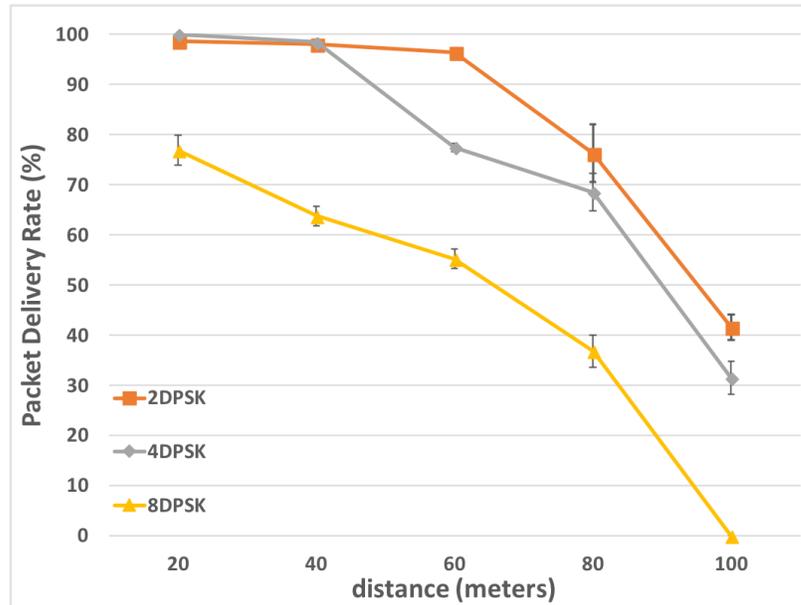


Figure 6.10: Packet delivery rates of different modulation levels using 2.4 GHz center frequency with 70dB output amplifier gain in terms of distance.

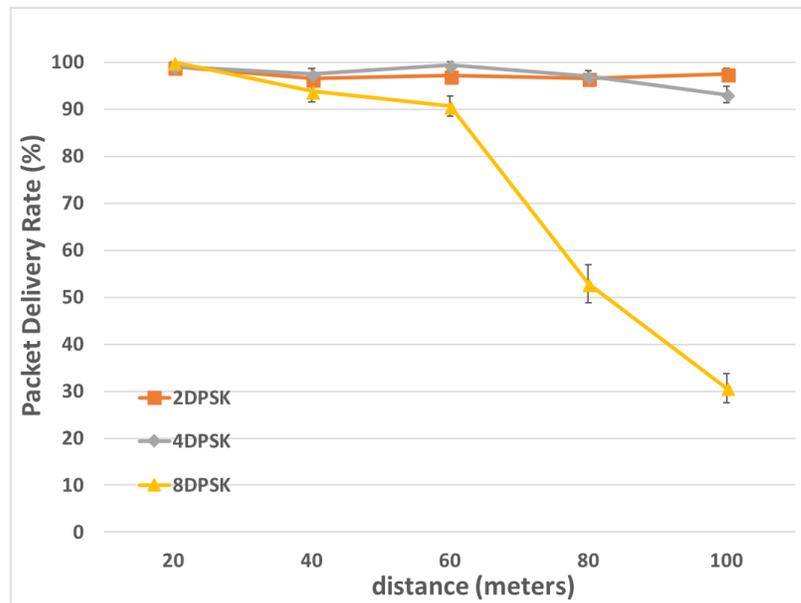


Figure 6.11: Packet delivery rates of different modulation levels using 915 MHz center frequency with 70 dB output amplifier gain in terms of distance.

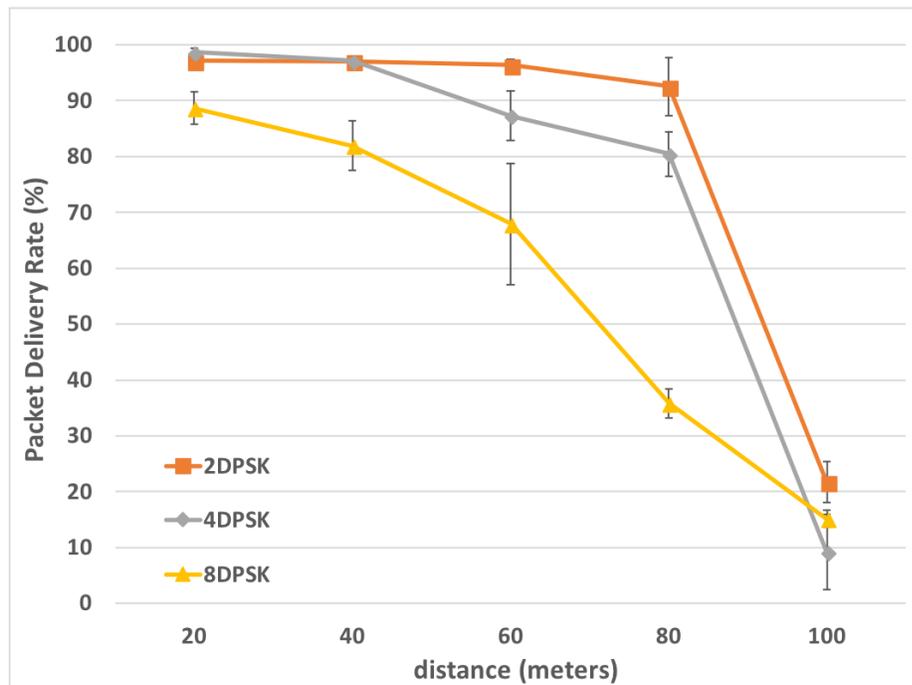


Figure 6.12: Packet delivery rates of different modulation levels using 915 MHz center frequency with 60 dB output amplifier gain in terms of distance.

value corresponded approximately to 0 dBm according to National Instruments internal tests [67]. However, this time I did not have a sub-1gig radio to compare RSSI values. I have observed similar but slightly better results compared to 2.4 GHz. This was likely due to a combination of slightly higher output power level created by 60 dB gain 915 MHz center frequency as oppose to 70 dB 2.4 GHz and lower free space propagation loss of 915 MHz band as oppose to 2.4 GHz band [68].

Lastly, I have expanded my distance testing to include wide variety of output amplifier gain values. My optimization problem, as you will see later in Section 7.1, assumes the energy consumption per modulation level, distance and output power level is known. In order to set realistic parameters for my simulation setup, I have run these additional SDR tests. These tests aim to find the PDR given a modulation level, sender-receiver distance and output power level. To accomplish this, I have used various output power levels, specifically -10, -7, -5, -3, -1, 0 dBm levels which CC2420 supports[69]. More precisely, I have set the

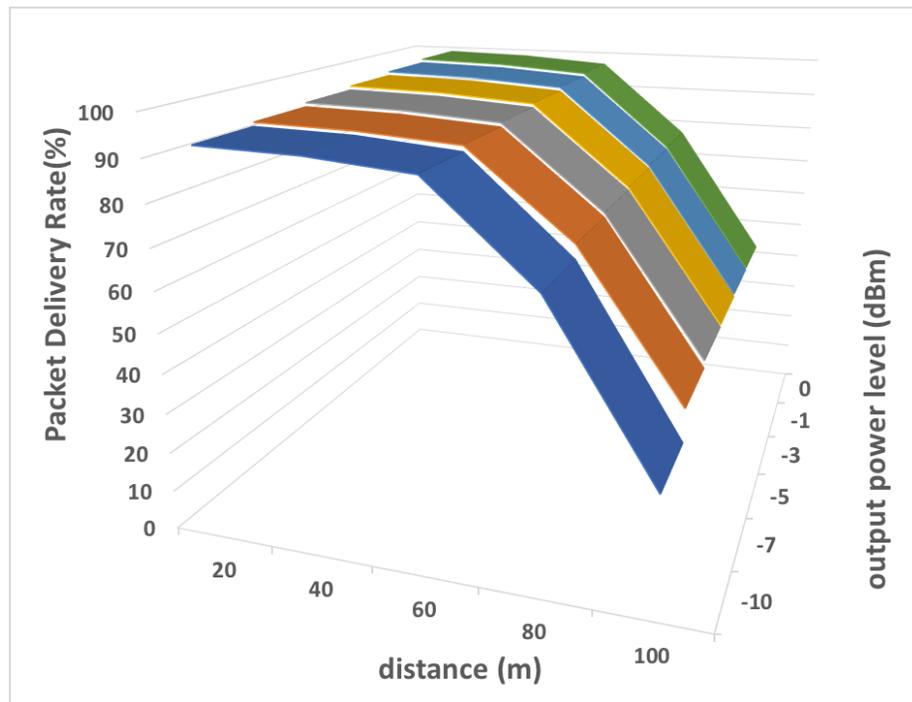


Figure 6.13: 2DPSK packet delivery rate given a transmitter-receiver distance and an output power level.

output and receiver amplifier gains to 60, 63, 65, 67, 69, 70 dB respectively to achieve the desired power output levels using 2.485 GHz center frequency. Figures 6.13, 6.14, and 6.15 shows the obtained results for 2DPSK, 4DPSK, 8DPSK respectively. In these Figures, you can see distance matrix on the x-axis, output power levels on z-axis and corresponding PDR values on y-axis.

These results have important implications for topology control. Connection driven topology control protocols such as Span [2], ASCENT [3] aim to find the minimum number of relay nodes necessary to forward packets to the coordinator. The goal is to reduce the number of active nodes and hence the number of hops. Lowering modulation level allows approaches that work by enabling nodes to broaden their communication range while reducing their energy consumption. As a result, these similar topology control protocols can further reduce the number of active relay nodes. Assume an example where a sensing node

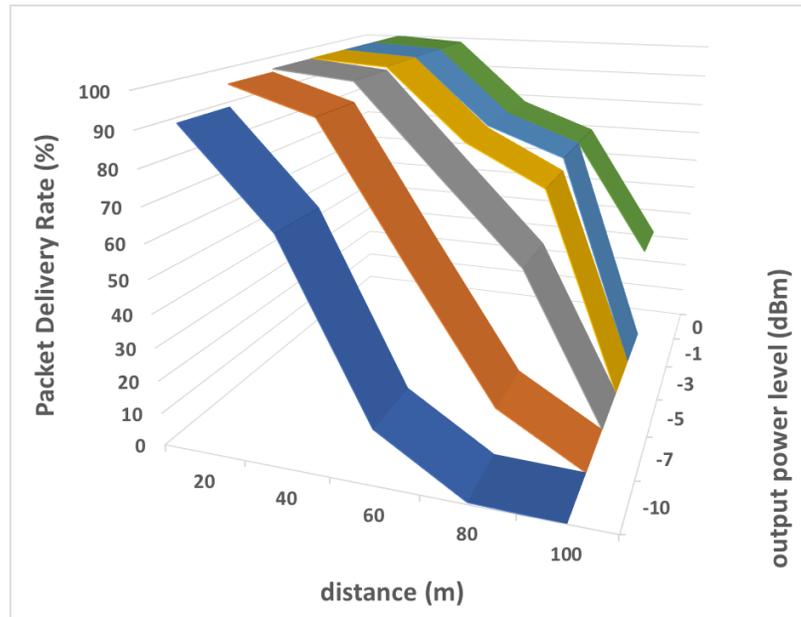


Figure 6.14: 4DPSK packet delivery rate given a transmitter-receiver distance and an output power level.

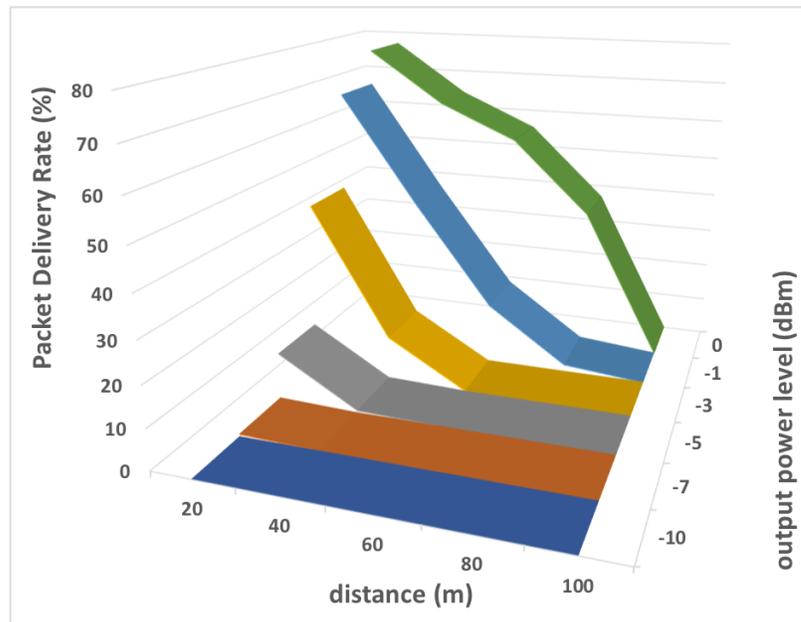


Figure 6.15: 8DPSK packet delivery rate given a transmitter-receiver distance and an output power level.

is 120 meters away from the coordinator and needs 2 relay nodes positioned 40 meters apart to forward its packets using modulation level 4. According to our tests, reducing modulation level to 2 will increase the reliable communication range up to 60 meters and only a single relay node will be sufficient to forward the packets. Here, the total energy savings will be the sum of *energy saving from the sensing node, relay node and coordinator by reducing their modulation levels* and *energy savings by putting a relay node into deep sleep*, a figure which is quite significant.

### **Elevation Testing**

In my last set of experiments, I have measured PDR of {2, 4, 8} DPSK in terms of elevation difference between sender and receiver. I have placed the sender outside of a building at 1.08m, 8.9m, and 13.6m high in reference to the receiver. The receiver is placed on the ground level (slightly elevated). Figure 6.16 shows the tests area and Figure 6.17 shows the height of the sender and the distance between sender and the receiver. In applications such as residential automatic meter reading, it is possible for sender and receiver to have elevation difference such as a transmitter can be placed on a current-meter higher of the ground or an application where an elevated pipeline equipped with sensors broadcasting the current pressure inside the pipe.

Figure 6.18 shows the PDR of {2, 4, 8} DPSK with 915 MHz center frequency and 60 dB output amplifier gain as I have used in Section 6.1.1. Some conclusions we draw can be listed as follows: (i) Higher elevations cause higher packet loss for both 915 MHz and 2.484 GHz center frequency although 2.484 GHz center frequency with 70 dB output amplifier gain had higher PDR for both 4 and 8 DPSK as shown in Figure 6.19. I should mention here that I was using different antennas for 915 MHz and 2.484 GHz tests that are both omni-directional vertical antennas (Ettus Research VERT900 and VERT2450 respectively). Different antennas may have different polarization characteristics typically either vertical, horizontal or circular in x-y axis. Elevation difference requires the signals to be transmitted in the z axis where VERT900 and VERT2420 antennas likely to have different characteristics



Figure 6.16: Test area that was used for elevation tests. Receiver is placed on the ground and the sender is elevated.

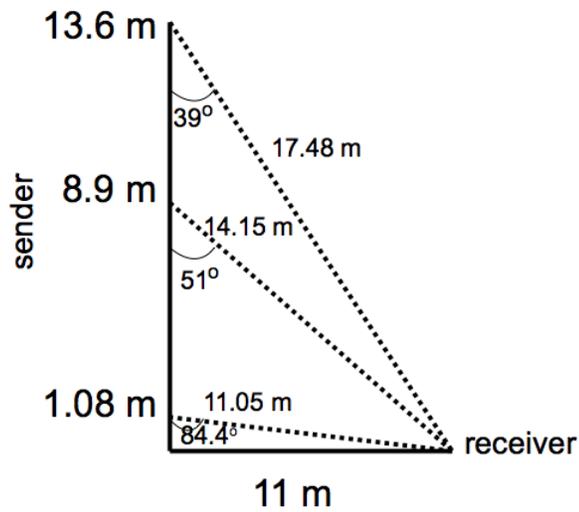


Figure 6.17: The height of the sender and the distance of the sender from the receiver.

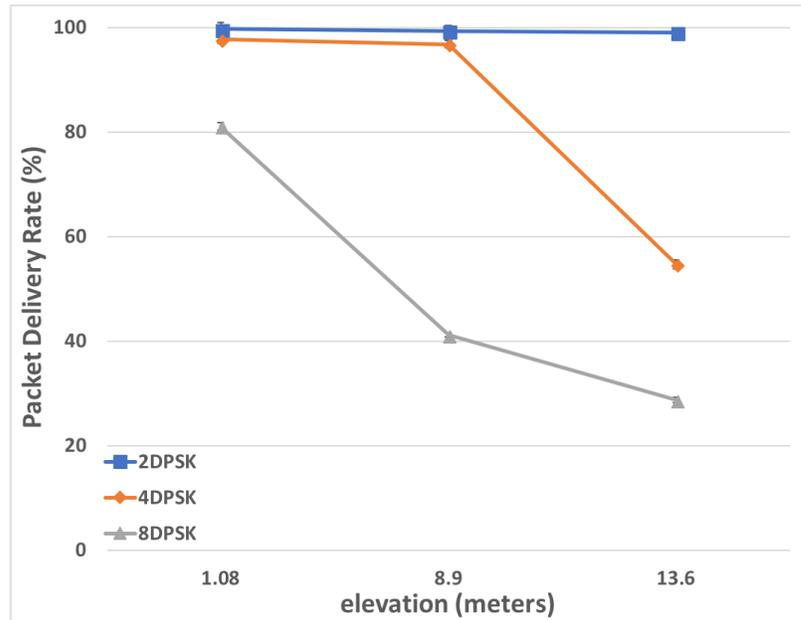


Figure 6.18: Packet delivery rates of difference modulation levels using 915 MHz center frequency with 60 dB output amplifier gain in terms of elevation difference.

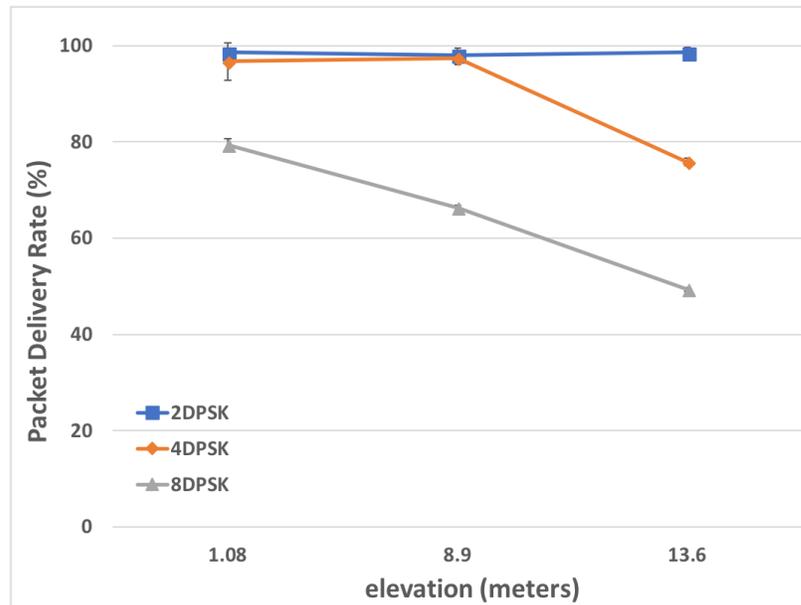


Figure 6.19: Packet delivery rates of difference modulation levels using 2.484 GHz center frequency with 70 dB output amplifier gain in terms of elevation difference.

in this direction. (ii) 2DPSK seems to be not effected by the elevation difference with the tested output gain values making it the best candidate for the aforementioned types of applications. (iii) If we compare the results from Figures 6.18 and 6.19 with Figures 6.10 and 6.12, it shows us that elevation difference has a higher impact on PDR compared to distance difference with the same elevation. Figures 6.10 and 6.12 indicates that all the modulation levels have 80% or above PDR for distances up to 20 m ( with no elevation difference) whereas when the transmitter is elevated 13.6m and placed 17.48 m apart from the receiver, the PDR reduces dramatically for 4 and 8 DPSK as low as 54% and 28% respectively.

## Chapter 7: DMS-enabled Multi-hop Topology Control for Time Critical WSNs

This chapter aims to solve the problem of optimal interference aware time-slot allocation for DMS-enabled, time-critical, multi-hop WSNs. I am particularly interested in answering the following question: *how does empirical measurements from Chapter 6 affect parent node selection process for time critical multi-hop WSNs?*

To answer this question, I first present the Mixed Integer Nonlinear Programming (MINLP) formulation of the problem; *a given a set of connections between nodes per modulation level, choose a modulation level and a parent for each node to talk to while minimizing the total network energy consumption given a maximum delay bound of when the base station has to collect all the information from the network.* The system model that I adapt for this problem is explained in Section 3.2. This problem description assumes the links between the nodes *per modulation level* are known, in other words, it assumes given a node, let's say node<sub>a</sub>, the set of nodes that node<sub>a</sub> can talk to using modulation levels 1, or 2, or 3,... is known. These set of nodes per modulation level are not necessarily identical. Generally speaking, MINLP problems are known to be intractable which led me to provide two polynomial time heuristics. I have compared my heuristics against the optimal solution using the simulation I built by incorporating the data I obtained from our Chapter 6.

In summary this chapter has the following contributions; *i)* I describe a new perspective on how DMS can be applied to WSNs. The papers in literature typically assume discrete and single power levels for each modulation level. While this is valid, it is very limited and not necessarily the only valid model. It is possible to adjust power output levels independently from the modulation levels that is used. To my best knowledge, this is the first work to use this extended model in DMS enabled WSN domain. *ii)* I show the optimal MINLP

formulation of DMS enabled parent node selection problem for time critical WSNs and *iii*) I propose two polynomial time heuristics and compare them against the optimal solution.

## 7.1 Optimization Problem Formulation

Here, I present the optimization problem formulation of *a given a network topology with known potential parent nodes per modulation level, choose a parent and a modulation level for each node while minimizing the total network energy consumption given a maximum delay bound of when the base station has to collect all the information from the network.*

Figure 7.1 gives an example case from a single node's perspective, node<sub>4</sub> in this case. The arrows show the connection between nodes with possible modulation levels and corresponding delay and energy consumption values. The goal is to find a path from node<sub>4</sub> to the base-station, denoted with b, bounded by 3t. There are three possible routes in this case;  $r_1 : n_4 \rightarrow n_2 \rightarrow n_1 \rightarrow b$ ,  $r_2 : n_4 \rightarrow n_3 \rightarrow b$ , and  $r_3 : n_4 \rightarrow n_1 \rightarrow b$ . The lowest delay  $r_1$  can provide is 3t with 12e of corresponding total energy consumption. For  $r_2$  and  $r_3$ , these values are 4t with 2e and 3t with 5e respectively. Only  $r_1$  and  $r_3$  satisfies the given delay bound where  $r_3$  has the lower corresponding energy consumption. Hence, in this example  $n_4$  should use  $r_3$ .

My optimization formulation finds these paths for each node and their transmission start times such that the total energy consumption of the whole network is minimized while the time it takes to collect data from each node is bound by the given deadline. Table 7.1 is the list of the variables and their definitions used in our problem formulation. The integer variables in our formulation are the  $\beta_l^{i,k}$  and  $I^{i,k}$  binary integer variables. The remaining variables are real numbers (except for  $IR_l^i$  which is a set). When  $\beta_l^{i,k}$  equal to 1, that means node  $i$  will transmit to node  $k$  using modulation level  $l$ . The same notation is also used for the remaining variables. The value of  $e_l^{i,k}$  might differ for each link even for the ones which are using the same modulation level depending on the current noise level of the channel as well as the distance between transmitter and the receiver. Energy consumption between

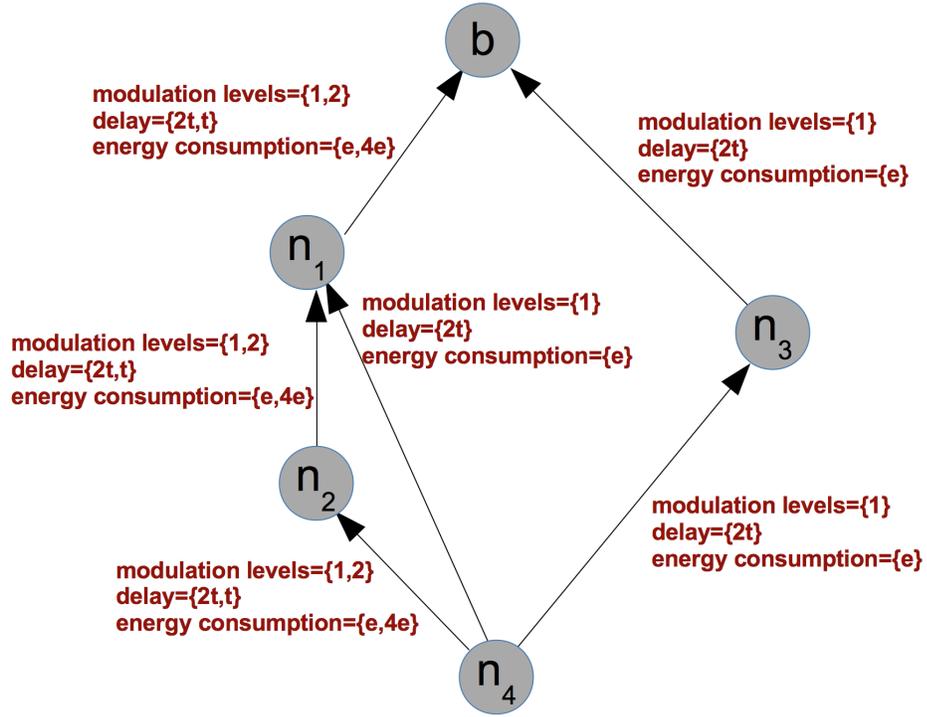


Figure 7.1: Optimization problem demonstration from a single node's perspective.

Table 7.1: The variables used in the problem formulation and their description.

$e_l^{i,k}$	transmission plus computation energy consumption when node $i$ sends a packet to node $k$ via modulation level $l$ .
$d_l^{i,k}$	transmission and computation delay when node $i$ sends a packet to node $k$ via modulation level $l$ .
$\beta_l^{i,k}$	binary integer variable to ensure node $i$ only picks a single parent node $k$ to send its packets.
$S_i$	the time when node $i$ starts transmitting.
$E_i$	the time when node $i$ stops transmitting.
$I^{i,k}$	binary integer variable to ensure only node $i$ or node $k$ is scheduled to transmit at any given time, never both.
$n$	total number of nodes in the tree.
$IR_l^i$	modulation $l$ interference range of node $i$ .
$D$	deadline constraint.
$Q$	a very large number

node  $i$  and all non-reachable nodes are set to  $\infty$ . Similarly, delay variable  $d_l^{i,k}$  is also set to  $\infty$  if node  $i$  cannot reach node  $k$  via modulation level  $k$ .

$$\min. \sum_{i=1}^n \sum_{k=1}^n \sum_{l=b_{min}}^{b_{max}} \beta_l^{i,k} e_l^{i,k} \quad (7.1a)$$

$$\text{s.t.} \sum_{k=1}^n \sum_{l=b_{min}}^{b_{max}} \beta_l^{i,k} = 1 \quad \forall i \neq k \quad (7.1b)$$

$$\beta_l^{i,k} (S_i + d_l^{i,k}) = \beta_l^{i,k} E_i \quad \forall i, k, l \quad (7.1c)$$

$$\beta_l^{i,k} E_i \leq S_k \quad \forall i, k, l \quad (7.1d)$$

$$\beta_l^{i,k} E_i \leq \beta_l^{i,k} (S_j + I^{i,j} Q) \quad \forall i, k \forall j \in IR_l^i \quad (7.1e)$$

$$\beta_l^{i,k} E_j \leq \beta_l^{i,k} (S_i + (1 - I^{i,j}) Q) \quad \forall i, k \forall j \in IR_l^i \quad (7.1f)$$

$$\beta_{i_1}^{i,k} \beta_{l_2}^{m,t} E_i \leq \beta_{i_1}^{i,k} \beta_{l_2}^{m,t} (S_m + I^{i,m} Q) \quad \forall m \notin IR_{l_1}^i, \forall k \in IR_{l_2}^m \quad (7.1g)$$

$$\beta_{i_1}^{i,k} \beta_{l_2}^{m,t} E_m \leq \beta_{i_1}^{i,k} \beta_{l_2}^{m,t} (S_i + (1 - I^{i,m}) Q) \quad \forall m \notin IR_{l_1}^i, \forall k \in IR_{l_2}^m \quad (7.1h)$$

$$S_i \geq 0 \quad \forall i \quad (7.1i)$$

$$S_{\text{basestation}} \leq D \quad (7.1j)$$

$$\beta_l^{i,k} \in \{0, 1\} \quad \forall i, k, l \quad (7.1k)$$

$$I^{i,k} \in \{0, 1\} \quad \forall i, k \quad (7.1l)$$

Equation 7.1a is the objective function which is to minimize the total sum of each node's energy consumption. When  $\beta_l^{i,k}$  is 1, we sum the corresponding energy consumption of node  $i$  transmitting to node  $k$  using modulation level  $l$ .

Constraint 7.1b forces each node to select a single modulation level and a parent node

to transmit its data.

Constraint 7.1c sets the transmission end time of each node,  $E_i$ , to its transmission start time plus transmission delay of the chosen parent and chosen modulation level.  $E_i$  is an auxiliary variable to ease the readability of the formulation. When  $\beta_l^{i,k}$  is 1, the transmission start time of node  $i$  ( $S_i$ ) plus the transmission delay of node  $i$  transmitting to node  $k$  using modulation level  $l$  ( $d_l^{i,k}$ ) equals to the transmission end time of node  $i$ .

Constraint 7.1d makes sure if node  $i$  chooses node  $j$  as its parent then node  $j$  can start transmitting only after node  $i$  stops transmitting. If there are multiple nodes who transmits to node  $k$ ,  $S_k$  has to be greater than the largest  $E_i$  value which implies that a node can start transmitting only after *all* of its children stops transmitting. When  $\beta_l^{i,k}$  is 0,  $S_k$  has to be greater than or equal to 0 which is trivial.

Constraints 7.1e and 7.1f ensure the nodes which are interfering with each other do not transmit simultaneously. More specifically, either the start time of node  $i$  is after the end time of node  $j$  or the start time of node  $j$  is after the end time of node  $i$ . When  $\beta_l^{i,k}$  is 1, for all the nodes which are in node  $i$ 's interference range of modulation level  $l$  the following holds;  $E_i \leq S_j + I^{i,j}Q$  and  $E_j \leq S_i + (1 - I^{i,j})Q$ . If  $I^{i,j}$  equals to 1 then it becomes  $E_i \leq S_j + Q$  and  $E_j \leq S_i$ . Since  $Q$  is a very large number  $E_i \leq \infty$  is trivially satisfied and node  $i$  is allowed to transmit only after node  $j$  stops transmitting. If  $I^{i,j}$  equals to 0 then the constraints become  $E_i \leq S_j$  and  $E_j \leq S_i + Q$ . The second constraint is trivially satisfied again and node  $j$  is allowed to start transmitting only after node  $i$  stops transmitting.

Constraints 7.1g and 7.1h are to prevent collision due to hidden node problem. Even though I have prevented interfering nodes from co-scheduling (Constraints 7.1e and 7.1f), it is still possible to have colliding nodes. In Figure 7.2 nodes  $i$  and  $m$  are not within each other's interference range. However, if they were to transmit to nodes  $k$  and  $t$  respectively, node  $k$  may not have been able to recover  $i$ 's signal since  $k$  is within interference range of  $m$ . Hence, node  $k$  shouldn't be scheduled to receive any packets while node  $m$  is transmitting. When node  $i$  transmitting to node  $k$  and any node which node  $k$  is in its interference range is

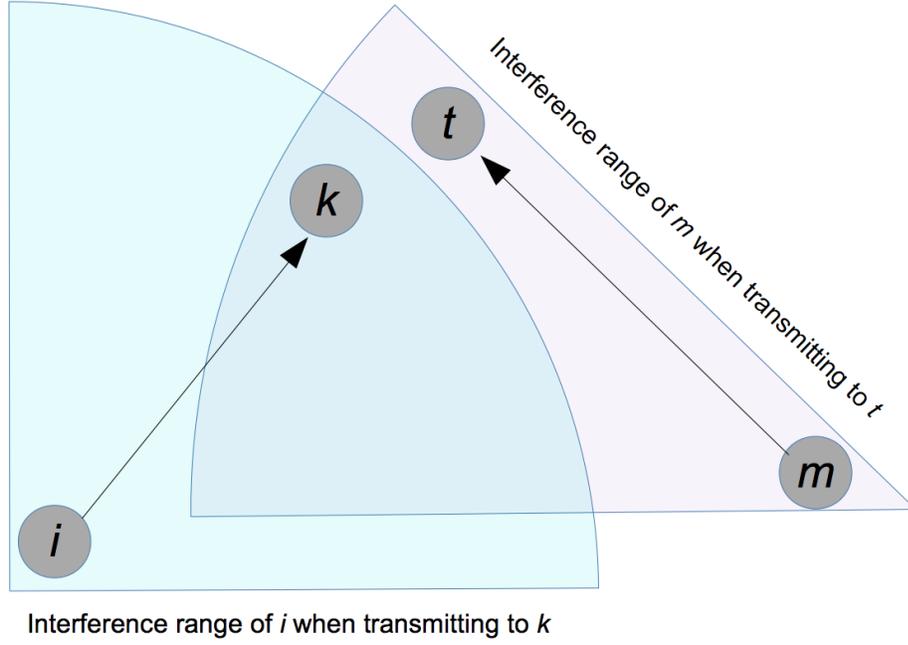


Figure 7.2: Hidden node scheduling problem.

also transmitting, the product  $\beta_{l_1}^{i,k} \beta_{l_2}^{m,t}$  becomes 1. Here I used two different variables for the modulation levels  $l_1$  and  $l_2$  to emphasize that the nodes  $i$  and  $m$  might be using different modulation levels and node  $k$  has to be in the interference range of node  $m$ 's currently chosen modulation level  $l_2$ . When this holds, the constraints becomes  $E_i \leq S_m + I^{i,m}Q$  and  $E_m \leq S_i + (1 - I^{i,m}Q)$ . Using the same logic I have demonstrated for the Constraints 7.1e and 7.1f, these conditions enforces either the transmission start time of node  $m$  to be after node  $i$ 's transmission end time or vice versa.

The remaining constraints are straight forward. Constraint 7.1i is to make sure nodes do not start transmitting prematurely whereas Constraint 7.1j is to enforce the deadline which is to make sure base-station receives all the packets in at most  $D$  seconds. Lastly, Constraints 7.1k and 7.1l defines the variable  $\beta$  and  $I$  as binary integers.

## 7.2 Polynomial Time Heuristics

Generally, integer programming problems are known to be intractable, hence to evaluate the performance of the optimization problem described in Section 7.1, I have designed two polynomial time heuristics which are explained in this section. However, there are some components which are shared by both and I wanted to explain these shared components here first.

### 7.2.1 Finding the Minimum Delay Path

First shared step is finding the minimum delay path. This step involves running Dijkstra's shortest path algorithm from each node until the shortest path to the base-station is found. I assigned the delay of each edge as its weight. Let's call the total number of nodes in the network including the base-station  $n$ , and the total number of modulation levels  $b$ . Complexity of Dijkstra's algorithm implemented with a binary heap is  $O(|E| + |V|\log|V|)$ [70]. In my set up, the number of edges in the worst case is  $b \cdot n^2$ . Hence each Dijkstra's algorithm run is  $O(b \cdot n^2 + n \log n)$ . I run Dijkstra's algorithm from each node so the total is  $O(b \cdot n^3 + n^2 \log n)$ . In practice  $O(b \cdot n^3 + n^2 \log n)$  is not a very tight bound. After a single round of Dijkstra's algorithm from a node, let's call it  $node_a$ , the shortest paths of all the nodes in  $node_a$ 's shortest path are also computed. Hence, there is no need to run Dijkstra's algorithm from these nodes. Nevertheless, the time complexity of this step is  $O(n^3)$ .

After the end of this process, each node now knows their shortest path parent and which modulation level to use when sending to the base-station.

### 7.2.2 Interference Aware Ordering

The next shared step is interference aware ordering. This step aims to find a schedule among all the nodes such that no interfering nodes will be co-scheduled as well as no parent will be scheduled before its children. In order to accomplish this, I run a *topological sort* on the directed-acyclic-graph (dag) generated after the first step described above. Topological

sort of a dag is the ordering of its vertices such that if there is an edge  $(u, v)$ ,  $u$  appears before  $v$  in the ordering. After the topological sort of the dag is known and I start assigning orders to the nodes as the following. I initialize each node's order to -1. Following the topological sort, each node is assigned the first available order in the range  $[maximum\ order\ of\ its\ children+1, maximum\ order\ among\ all\ the\ nodes\ in\ its\ interference\ range\ of\ every\ modulation\ level\ and\ their\ minimum\ delay\ parents+1]$ . Moreover, when a node is assigned an order, its parent node is also temporarily assigned the same order to prevent interference due to hidden node. Topological sort makes sure no parent is scheduled before its children and my ordering mechanism makes sure no interfering nodes are co-scheduled. Time complexity of topological sort is  $O(|V| + |E|)$ , the worst case number of edges in our setup is  $b \cdot n^2$  where  $b$  is number of modulation levels and  $n$  is the number of nodes in the network including the base-station. The time complexity of assigning orders is  $O(n^2)$ . All together, the total time complexity of this step is  $O((b + 1) \cdot n^2 + n)$  which is  $O(n^2)$ .

### 7.2.3 Computing Minimum Delay

The third and the last shared step is computing minimum delay. Since now we know each node's minimum delay parent, minimum delay modulation level and its order, in this step I compute each node's minimum delay such that the minimum time it takes for that node to send its packet to the base-station. This is calculated by first computing *time-slot lengths*. There are same number of time-slots as the maximum order value (the order of the base-station). Each time slot length is equal to the maximum delay of the nodes which has the same order as that time slot number. Minimum delay of a node is then computed by summing up the time slot lengths from its order till (and including) the order of the node in its shortest path which is directly connected to the base-station. The calculated minimum delay of a node means this is the shortest time it would take for this node to send its packet to the base-station according to the path and ordering chosen by the previous steps. The time complexity of computing time slot lengths is  $O(n)$ . Computing minimum delay of each node can be done recursively such that  $mindelay(i) = sum\ of\ time\ slot\ lengths\ from\ i\ to$

its minimum delay parent  $p + \text{mindelay}(p)$ . And if we *memoize* the previously computed minimum delay values, each node is only visited once and hence the time complexity is  $O(n)$ . The total time complexity of this step is  $O(2n)$  which is  $O(n)$ .

The total time complexity of all shared steps is  $O(b \cdot n^3 + n^2 \log n + (b + 1) \cdot n^2 + 3n)$  where  $b$  is a constant and equals to the number of modulation levels and  $n$  is the number of nodes in the network including the base-station. Hence, the total complexity is  $O(n^3)$ .

Next, I describe my heuristics.

#### 7.2.4 Aggressive

My first heuristic starts with computing minimum delay of each node by following the steps described in the beginning of this section. When given a maximum delay bound, a.k.a. deadline, *Aggressive* first checks to make sure the minimum delay of each node is at least as long as the deadline. If all the nodes can meet the deadline, it calculates the *slack* which is the difference between the delay of node with the longest minimum delay and the deadline. As long as the slack is large enough to compensate for the additional delay by a reduction of a modulation level, the time-slot which has the most number of nodes with minimum delay modulation levels greater than the lowest modulation level is chosen. The modulation levels of these nodes are reduced by one level. The slack is then recomputed and the process is repeated till we no longer have a large enough slack. Algorithm 1 is the pseudocode of *Aggressive* for a clearer top down view.

Comparing minimum delay of each node against the deadline can be done in  $O(n)$  time complexity. Finding the time-slot with the with the highest number of nodes with modulation levels greater than  $b_{min}$  can be done in  $O(2n)$ ; first by going over the each node and updating the corresponding time-slot information and then by iterating over the time-slots to return the maximum. In the worst case each node will occupy a different time-slot and hence it will take  $O(n)$  time. Lowering each node's modulation level can also be done in  $O(n)$  in the worst case which in when all the nodes share a single time-slot and all are eligible for a modulation level reduction. The time complexity of computing

---

```

for each node in the network do
  | Check if minimum delay  $\leq$  deadline;
  | return no answer if the check is false;
end
slack = deadline - max of the min delays of each node;
while slack  $\geq$  increase in delay by reducing modulation levels by one do
  | slot := most occupied time-slot by nodes with modulation levels greater than
  |  $b_{min}$ ;
  | updated := false;
  | for node := nodes in slot do
  | | if modulation level of node  $\geq b_{min}$  then
  | | | reduce modulation level of node by one;
  | | | updated = true;
  | | end
  | end
  | if updated is false then
  | | break the while loop;
  | end
  | recompute min delays of the nodes;
  | slack = deadline - max of the min delays of each node;
end

```

---

**Algorithm 1:** The algorithm of Aggressive Heuristic

minimum delays of the nodes are explained in the third bullet of the shared steps explained previously which is  $O(2n)$ . The *while loop* can at most loop for  $(b - 1)n$  times since each node can at most reduce its modulation level  $b - 1$  times. Hence the total complexity of *Aggressive* without the initial shared steps is  $O(5(b - 1)n^2 + n)$  if we include these steps the time complexity becomes  $O(bn^3 + n^2 \log n + (6b - 4)n^2 + 4n)$  which is  $O(n^3)$ .

### 7.2.5 Explorer

My second heuristic is an improved version of Aggressive where it also starts with computing the minimum delay of each node by following the shared steps with one addition: while I am running Dijkstra's algorithm to compute the minimum delay paths, I also store the **cumulative energy consumption** of each node. A node's, let's say  $node_a$ 's, cumulative

energy consumption equals to the sum of each node's individual energy consumption which are on that node<sub>a</sub>'s shortest path including node<sub>a</sub>.

Using the collective information of both *cumulative energy consumption* and *minimum delay*, the nodes now *explores* more options to forward their packets other than their minimum delay parents. Instead of *aggressively* forwarding packets to their minimum delay parents, now the nodes look among all the nodes within its reach with higher order number and picks the one with the (minimum cumulative energy consumption + energy to send to it) among the nodes which have short enough (minimum delay + delay to send to it) value to meet the deadline. Algorithm 2 shows the pseudocode of the Explorer Heuristic.

Explorer keeps track of each node's individual deadline requirement separately. This is because each node's individual deadline will change according to the nodes' individual deadline which pick it as their parent. Algorithm 2 starts with initializing each node's individual deadline requirement to the given application's deadline requirement. Then, it proceeds node by node from the one with the smallest order to the largest. Let's assume Explorer currently picked node<sub>i</sub> and will try to find a parent for it. The other nodes need to satisfy two conditions to be qualified as a candidate; i) its order has to be greater than the node<sub>i</sub>'s, ii) its minimum delay has to be less than or equal to node<sub>i</sub>'s individual delay. By iterating over the nodes according to their orders and picking the nodes with greater order ensures that the minimum delay computed initially still holds because the nodes with greater order still has the potential to pick their minimum delay parents if necessary. Node<sub>i</sub> picks the node with the lowest cumulative energy consumption among the set of candidates, let's call this node<sub>j</sub>. Next the individual delay of node<sub>j</sub> has to be updated. Node<sub>j</sub> now has to be make sure it can delivery the packet in the worst case of (individual delay of node<sub>i</sub> - transmission delay from node<sub>i</sub> sending to node<sub>j</sub> by using the chosen modulation level). Only updating the individual delay of node<sub>j</sub> is not enough. Not only node<sub>j</sub> but also all the nodes with greater order than node<sub>i</sub>'s has to have at least the individual delay of node<sub>i</sub>. This is because if any of the nodes with greater order than node<sub>i</sub>'s, delivers its packet with a longer delay than of node<sub>i</sub>'s individual delay then there is a chance for node<sub>i</sub> to miss the

---

**Input:** deadline  
initialize each node's individual deadline to deadline;  
initialize each order's minimum requirement to deadline;  
sorted := sort the nodes according to their order;  
**for**  $i$  *in sorted* **do**  
    update  $i$ 's individual deadline requirement to the updated minimum delay requirement from previous orders;  
    min :=  $\infty$ ;  
    **for**  $j$  *within  $i$ 's reach per modulation level* **do**  
        **if**  $j$ 's order >  $i$ 's order AND  $j$ 's minimum delay  $\leq$   $i$ 's individual deadline AND  $j$ 's cumulative energy consumption + energy consumption from  $i$  to  $j$   $\leq$  min **then**  
            choose  $j$ ;  
            choose modulation level;  
            update min;  
        **end**  
    **end**  
    update  $j$ 's individual deadline to MINIMUM of  $j$ 's individual deadline OR  $i$ 's individual deadline - delay from  $i$  transmitting to  $j$ ;  
    update the minimum requirement of the order ( $i$ 's order + 1) to MINIMUM of minimum requirement of ( $i$ 's order + 1) OR  $i$ 's individual deadline - delay from  $i$  transmitting to  $j$ ;  
**end**  
recompute interference aware ordering;  
recompute minimum delay;  
**if** *all the nodes meet deadline* **then**  
    run Aggressive Heuristic with the new edges;  
**end**  
**else**  
    run Aggressive Heuristic with Dijkstra's shortest path edges;  
**end**

---

**Algorithm 2:** The algorithm of Explorer Heuristic

deadline.

After Explorer picks a parent for each node, it recomputes the interference aware ordering with the new set of edges. Please recall that interference aware ordering was topological sort based, now with the new set of edges, it is possible for a node to be a leaf node whereas it was not according to initially computed shortest paths based on Dijkstra's algorithm, this has the potential to lower its minimum delay because now it does not have to wait for its children and grandchildren to finish before start transmitting. On the down side, the nodes which has the same order with different minimum delay parents might now be interfering if Explorer was to choose them the same parent. Because of the potential changes in ordering, Explorer recomputes the orders and checks if the deadline is still met. If so it runs the Aggressive Heuristic with the new edges chosen by Explorer. Otherwise, the Aggressive Heuristic is run with the edges chosen by Dijkstra's algorithm as if Explorer did not even execute to begin with.

Sorting the nodes according to their orders take  $O(n \log n)$  time. There can be at most  $b(n - 1)$  nodes with the reach to be a parent candidate per node. Hence the for loop of Algorithm 2 has the time complexity of  $O(n \log n + bn(n - 1))$ . I have already discussed the time complexity of interference aware ordering and computing minimum delay which are  $O((b + 1) \cdot n^2 + n)$  and  $O(2n)$  respectively. The time complexity of the last step which is to run Aggressive Heuristic without the shared steps is  $O(5(b - 1)n^2 + n)$  and the time complexity of the shared steps were  $O(b \cdot n^3 + n^2 \log n + (b + 1) \cdot n^2 + 3n)$ . altogether the time complexity of Explorer becomes  $O(bn^3 + n^2 \log n + 8bn^2 - 3n^2 + n \log n + 7n)$  which is  $O(n^3)$ .

### 7.3 Performance Evaluation

In order to evaluate the optimal solution and the heuristics, I have created a custom Java simulator to simulate various network setups. The implementation of optimization problem is done by using IBM ILOG CPLEX Optimization Studio libraries [71]. There is one

detail I need to mention about the CPLEX implementation. In the MINLP formulation, the Constraints 7.1c, 7.1e, 7.1f, 7.1g and 7.1h are not necessarily convex functions which is a problem for the CPLEX software because CPLEX will throw an exception and stop execution when faced with non-convex constraints. To overcome this issue and linearize this formulation into a MILP form, I needed to create an additional auxiliary variable,  $z$ , for each integer variable ( $\beta$ ) and a continuous variable ( $S$  and  $E$ ) product such that:

$$z \leq \text{deadline} \cdot \beta$$

$$z \geq 0$$

$$z \leq S$$

$$z \geq S - \text{deadline} \cdot (1 - \beta)$$

Here if  $\beta$  is 0 then  $z \leq 0$  and  $z \geq 0$  forces  $z$  to be equal to 0. If  $\beta$  is 1 then  $z \leq S$  and  $z \geq S$  forces  $z$  to be equal to  $S$ . This as a workaround that I needed to use for the CPLEX implementation of the MILP problem.

### 7.3.1 Network Topology Setup

My simulation starts with the network topology setup process with the following parameters:

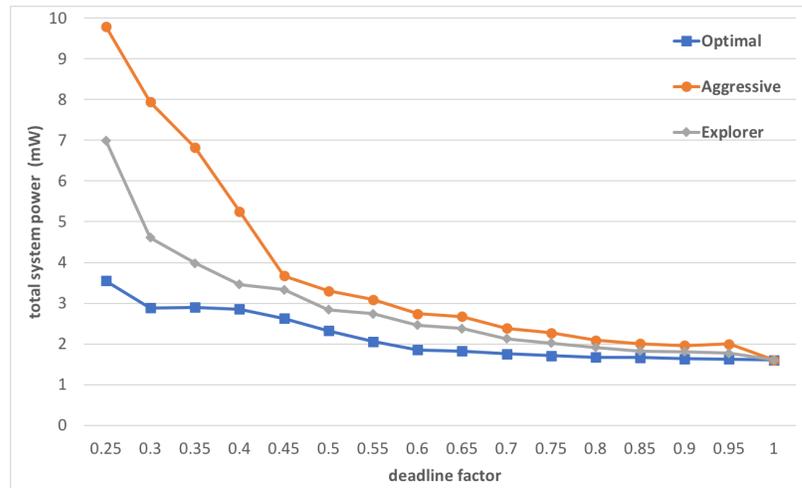
- x-y plane size: The size of x axis and the y axis of the setup area.
- number of nodes: The total number of nodes in the network in addition to the base station.
- energy consumption: The energy consumption of each modulation level per transmitter and receiver distance obtained from Figures 6.13, 6.14, and 6.15.
- delay: The time it takes to send a packet per modulation level.
- deadline factor: the percentage of the longest delay which is when each node is using the lowest modulation level. For example when *deadlinefactor* is 0.6 that means the delay bound  $D$  of the Constraint 7.1j is (number of nodes x lowest modulation level's delay x 0.6).

I preset the location of the base-station and x,y positions for the remaining nodes are randomly generated while making sure no two nodes are assigned the same coordinates. Next step is to setup the connections. For each pair of nodes (including the base-station), their straight line distance to each other is computed and the energy consumption for this pair of nodes per each modulation level is assigned as the following; for each curve that is plotted in Figures 6.13, 6.14, and 6.15, I have fitted the curves with 5<sup>th</sup> degree polynomial trend-lines with the coefficient of determination values of 0.99 and above. I have then assigned the corresponding output power levels while making sure there is at least 75% PDR. This corresponds to the  $e_l^{i,k}$  parameter of my MINLP where the pair of nodes are  $i$  and  $k$  and the modulation level is  $l$ . If this pair of nodes are unreachable for a modulation level or has a PDR of less than 75%, the energy consumption and the delay of that modulation level is assigned to infinity. Moreover, I have assumed the interference range is 1.2 times of the communications range [72]. If after this process, there exists a node such that it is not within the reach of any other node, then we reassign a new location to that node, once again randomly. This process is repeated till every node is connected.

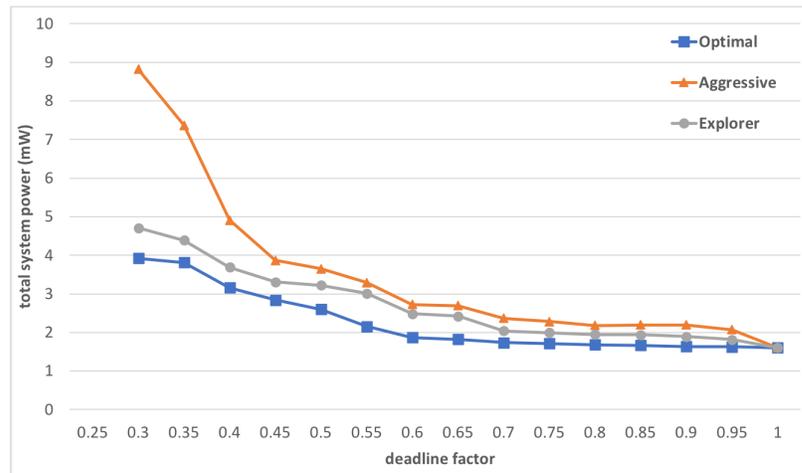
### 7.3.2 Results

In my first set of tests, I have placed the coordinator at the coordinate (0,0) and positioned the remaining 10 nodes by following the procedure described in Section 7.3.1. Each data point on Figures are the averaged results over 100 repetitions.

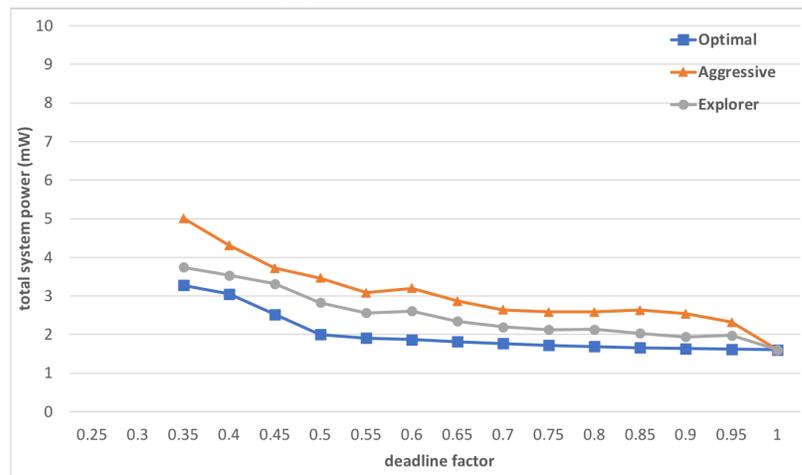
I have experimented with 250x250, 500x500, and 750x750 m<sup>2</sup> areas with deadline factor from 0.25 to 1. Figure 7.3 shows the total sum of each node's power consumption (except for the base-station) for a given deadline factor and the maximum size of the area which the nodes are spread. It is seen that for each density level and deadline factor, there is a strict ordering of power consumption as Optimal < Explorer < Aggressive except for when the deadline factor equals 1. At this point, the deadline is large enough for all the nodes to use the lowest modulation level to meet the deadline. Keen eyed readers must have realized that Figure 7.3b is missing data points for deadline factor 0.25 and Figure 7.3c



(a) area = 250x250 m<sup>2</sup>



(b) area = 500x500 m<sup>2</sup>



(c) area = 750x750 m<sup>2</sup>

Figure 7.3: Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes.

is missing for 0.25 and 0.30. This shows that when the density of the network decreases, it is harder to meet the shorter deadlines. If you recall our distance test results based on various power output levels (Figures 6.10, 6.11, 6.12, 6.13, 6.14, 6.15), higher modulation levels have lower successful communication ranges which forces sparse networks to use lower modulation levels to stay connected which in turn increases transmission delay.

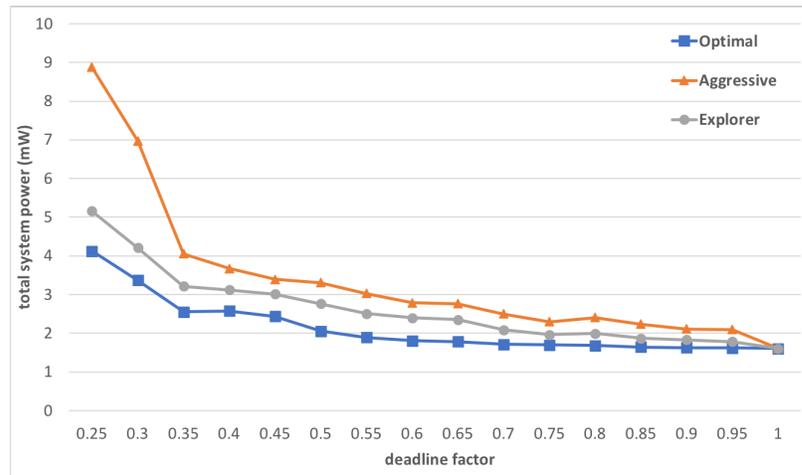
Another conclusion I need to derive from these Figures is the effect of network density. Denser network decreases the Optimal's as well as the Heuristics' total power consumption for tighter deadlines which is actually intuitive. Denser networks will have more potential paths to choose from. In fact a network will preserve all of its potential paths if it was to get *squeezed*. Why? Because any connection provided by any modulation level will still remain valid when the distances between sender and receiver pairs are decreased. Perhaps, there will even appear new potential paths. However, the co-schedulability of the nodes will be effected due to the increased interference. Another observation is the performance difference between my algorithms grows as the network gets more sparse. This also due to decrease in the number of potential paths. Higher number of potential paths allows Optimal solution to perform exhaustive search over more potential solutions. However, due to the polynomial time nature of our Heuristics, Aggressive and Explorer does not benefit from higher number of potential paths as much as Optimal does.

The rate of which the total power consumption decreases as the deadline factor increases is not constant. We can see that there are slight reductions in total power consumption when the deadline factor goes from 0.6 to 1 for all density levels. Although from the Figures the line looks relatively flat, there are roughly 14%, 35%, and 42% reduction in power consumption between deadline factors of 0.6 and 1 for Optimal, Explorer, and Aggressive respectively for each case. My SDR based experimental results have indicated that the power consumption difference between 2DPSK and 4DPSK is significantly lower than the difference between 4DPSK and 8DPSK. Hence, the reduction in total power consumption when more and more nodes are allowed to use 2DPSK rather than 4DPSK as the deadline factor approaches 1 is less than that of when more and more nodes are allowed to use 4DPSK

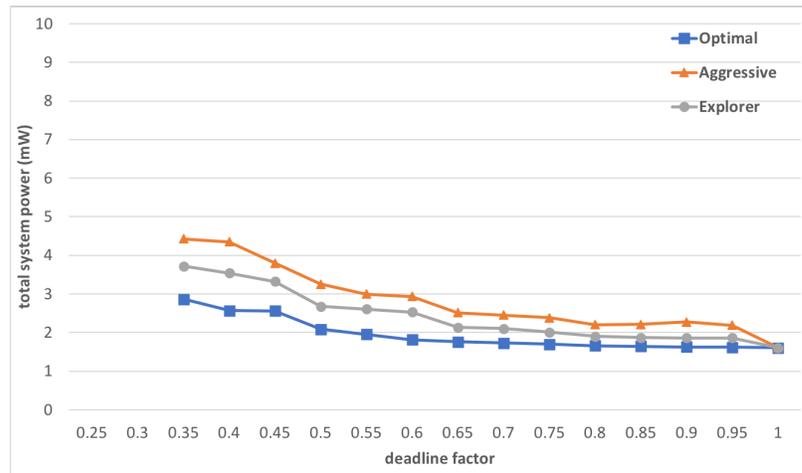
as opposed to 8DPSK when the deadline factor approaches 0.5. This further emphasizes the benefit of DMS on wide ranges of topologies for multi-hop networks and its relatively low scarifies of additional power consumption for a significant reduction in total transmission delay.

Similar traits are also observed when I repeat the tests but this time with the base-station placed in the center of the grid as oppose to corner. Placing the base-station in the center will reduce the average height of the tree and the nodes will have less hops to reach to it. This in return will improve co-schedulability due to less number of nodes affected by parent-children constraint (Section 7.1, Optimization Formulation 7.1, Constraint 7.1d). Since the size of the area is kept the same, the co-schedulability due to interference constraints will be unaffected (Section 7.1, Optimization Formulation 7.1, Constraints 7.1e 7.1f 7.1g 7.1h). The results are shown in Figure 7.4. If we compare the results shown in Figures 7.3a to 7.4a, 7.3b to 7.4b, and 7.3c to 7.4c, we can see that total power consumption has been reduced in each case. This is due to the improved co-schedulability as explained. Moreover, we can see that as the network area increase, so does the total power consumption just as we have seen in Figure 7.3.

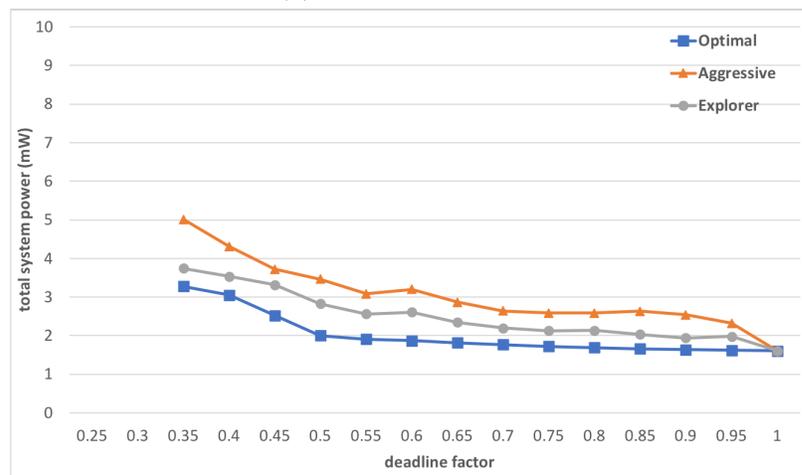
Next, I have conducted tests by significantly reducing the test area by forcing the nodes to be placed on a line with 250, 500, and 750 m length. Reducing the test area size to  $\frac{1}{250}$ ,  $\frac{1}{500}$ , and  $\frac{1}{750}$  of the previous test areas substantially increased the interference and reduced co-schedulability. Figure 7.5 shows the results. Once again I have observed that as the network area gets larger, the total power consumption increases. The reason is the reduced number of potential paths as we discussed previously. If we compare these results with Figure 7.4, we can see that the power consumption in each case has increased, that is for each area size and deadline factor. This comparison also gives us a good insight into the co-schedulability performance of our heuristics. I observe that the performance difference between our algorithms has increased as I reduced the test area size. This is due to the suboptimal ordering mechanism used by our heuristics (See Section 7.2) which has a negative impact on co-schedulability.



(a) area = 250x250 m<sup>2</sup>

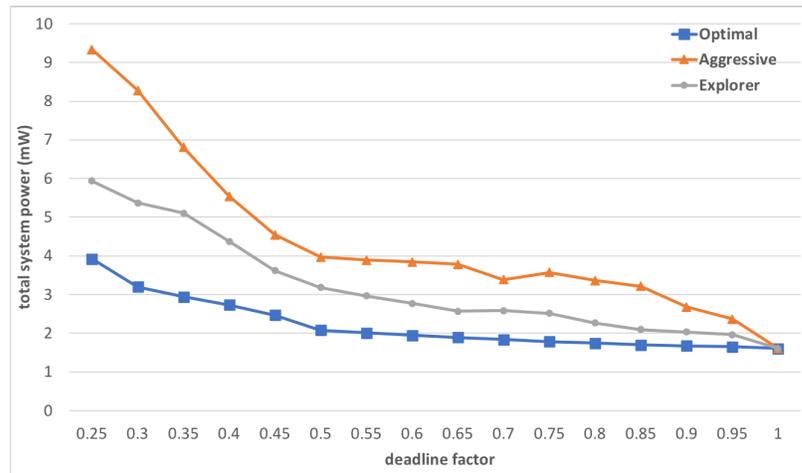


(b) area = 500x500 m<sup>2</sup>

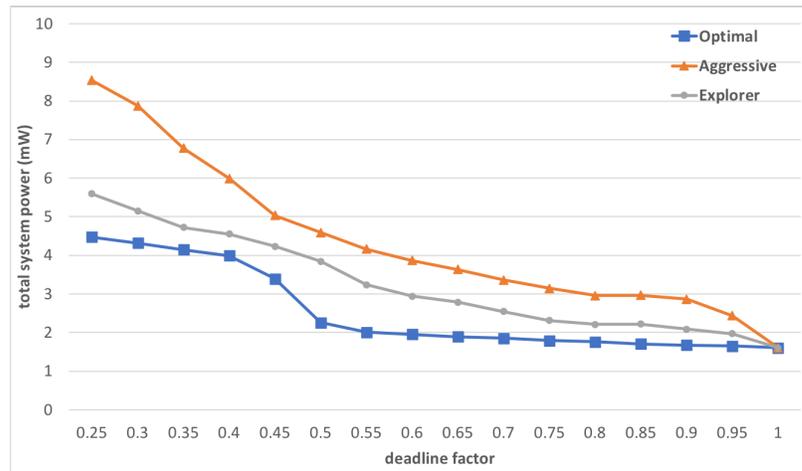


(c) area = 750x750 m<sup>2</sup>

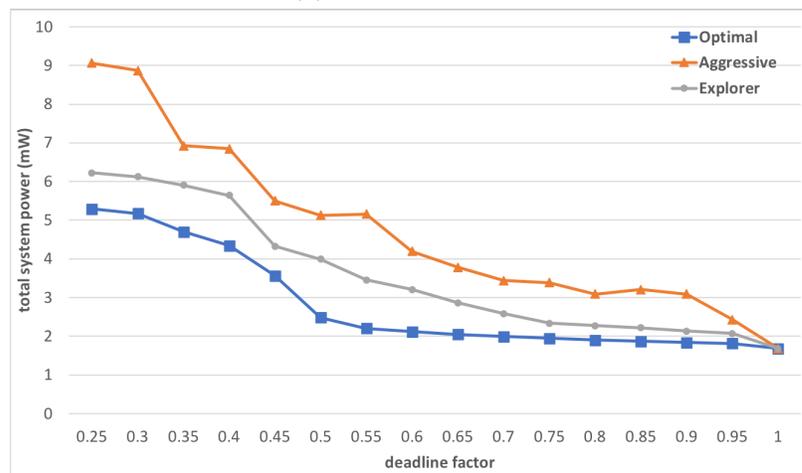
Figure 7.4: Comparison of optimal solution against heuristics when the base-station is placed at the *center* with 10 nodes.



(a) area = 250x1 m<sup>2</sup>



(b) area = 500x1 m<sup>2</sup>



(c) area = 750x1 m<sup>2</sup>

Figure 7.5: Comparison of optimal solution against heuristics when the base-station is placed at (0,0) with 10 nodes.

The interference model I have adapted in my dissertation is first and big step forward towards a more realistic analysis of DMS on WSNs. In my interference model, each modulation level has its own interference range which is 1.2 times of their real work communication range measurements obtained from our SDR tests [72]. The work in literature typically assumes equal interference ranges for different modulation levels. In order to see the effect of my new interference model, I conducted the following test; 10 nodes plus the coordinator placed at (0,0) on a 500x500 m<sup>2</sup> area. The same layout used for generating Figure 7.3b. However, this time the interference range of all the modulation levels equal to that of modulation level 1. In other words, 4DPSK and 8DPSK also has the same interference range of 2DPSK. This is a hypothetical assumption since the communication ranges of all the modulation levels were kept the same. Figure 7.6 shows the results. On average there were 5.7%, 2.46%, and 4.11% increase on power consumption for Optimal, Aggressive, and Explorer respectively when compared to the results shown in Figure 7.3b. The increase in power consumption was more pronounced for smaller deadline factors. For the deadline factor of 0.35, the performance difference was 21.95%, 5.92%, and 19.52% for Optimal, Aggressive, and Explorer respectively. This shows the significance of co-schedulability especially for shorter deadlines.

I want to mention another observation from comparing Figures 7.3, 7.4, and 7.5. The performance difference between these various tests becomes relatively smaller when the deadline factor is 0.5 or more. For Optimal at deadline factor of 0.5, the difference between the maximum and the minimum observed power consumption is 22%. This value is 33% and 37% for Explorer and Aggressive respectively. For the deadline factor of 0.6, these differences shrink to 14%, 25%, and 35% for Optimal, Explorer, and Aggressive. The reason behind this is the relatively close performance of 2DPSK and 4DPSK. At these deadline factors, the nodes are dominantly using either 2DPSK or 4DPSK. The percentage of each changes from case to case but the differences in these percentages create a relatively small difference in performance.

To analyze the scalability of our algorithms, I have run a set of experiments with 5 to

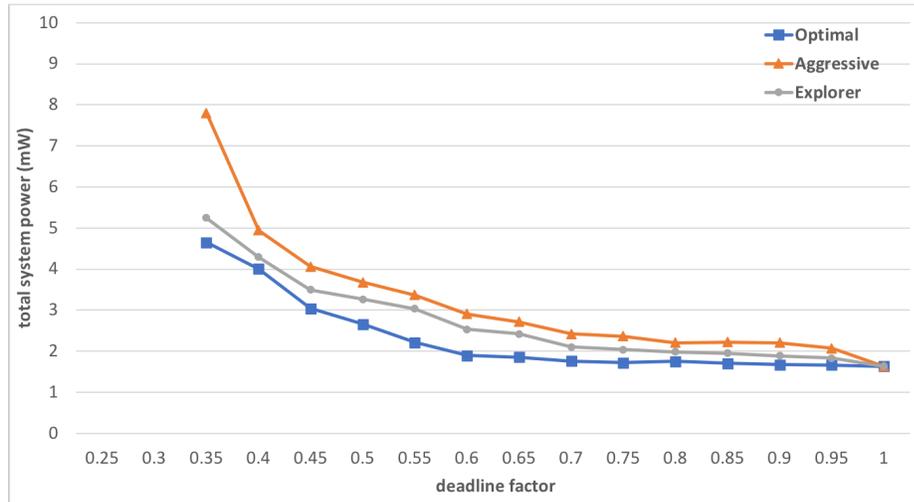


Figure 7.6: Comparison of optimal solution against heuristics when every modulation level has the same interference range of modulation level 1. Area size is 500x500 m<sup>2</sup> and network size is 10 nodes plus the coordinator placed at (0,0).

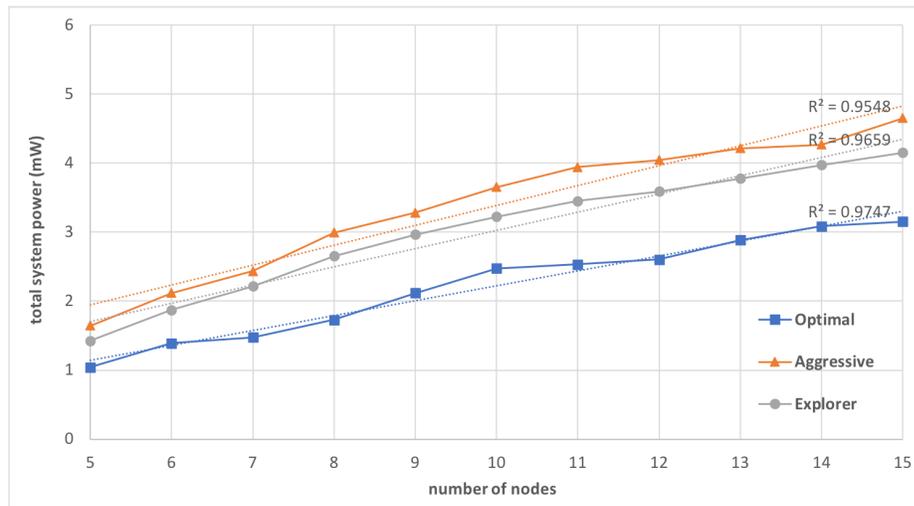


Figure 7.7: Scalability analysis in terms of number of nodes in a 500x500 m<sup>2</sup> area and the coordinator is placed at the coordinates (0,0) with a deadline factor of 0.5.

15 nodes in a 500x500 m<sup>2</sup> area and a coordinator is placed at the coordinates (0,0) with a deadline factor of 0.5. I chose 500x500 m<sup>2</sup> and a deadline factor of 0.5 to have relatively low computational load (smaller area sizes significantly increases the time optimal solution is computed due to increased interference among nodes) and still have a connected network (as the deadline factor decreases so does the connectivity of the network). I could only go up to 15 nodes because after that point, I was unable to find an optimal solution even after 5 days of running. Figure 7.7 shows the results fitted with linear regression lines to each algorithm. Coefficient of determination ( $R^2$ ) values of linear regression analysis were above 95% for all of the algorithms. This is a strong indication that the energy consumption of the particular setup we had grows linearly in terms of number of nodes. Moreover, the aims of the linear regression lines were also similar for the algorithms, 0.29, 0.26, and 0.22 for Aggressive, Explorer and Optimal respectively. That means although the performance difference between the algorithms diverges as the number of nodes increases, it is with a slow rate. So one can expect similar performances from all the algorithms as the number of nodes increases.

## Chapter 8: Compatibility of My Dissertation with the Existing Real-Time WSN Protocols

The main contribution of my dissertation is the integration of DMS into real-time WSNs. While I was doing so, I made sure that my research was generic enough to be tailored individually for wide adopted existing real-time WSN protocols. In this chapter I am going to give more details about how to do so.

Single cluster WSNs are setup such that the coordinator is within reach of every node in the network without relying on any other node to forward its packets typically as a star topology. IEEE 802.15.4 defines this topology layout and other widely used protocols such as Zigbee also adopts this model for its clusters [73]. The details of this topology is explained in Section 3.1. My work on single clusters fits this model one to one as explained in Chapter 4. More details need to be given for multi-hop/multi-cluster topologies.

### 8.1 Zigbee

Zigbee is an alliance which aims to connect a wide variety of devices to form a single network. Its specifications are built on top of IEEE 802.15.4 standard and adds networking and security layers. Network layer is of particular interest for the scope of this dissertation.

Figure 8.1 shows the Zigbee topology. There are three types of device annotated by their first letters; coordinator, router, and end device. If I compare these devices with the ones I used in my dissertation (See Architectural Model (Section 3.2), they correspond to base-station, sense-and-relay nodes, and sensing-nodes respectively. Coordinator is the root of the tree and is the most capable device of the network. The information about the current condition in network is stored in this device. Router node's duty is to forward the packets it receives from end devices to the coordinator. It is also possible for end devices to

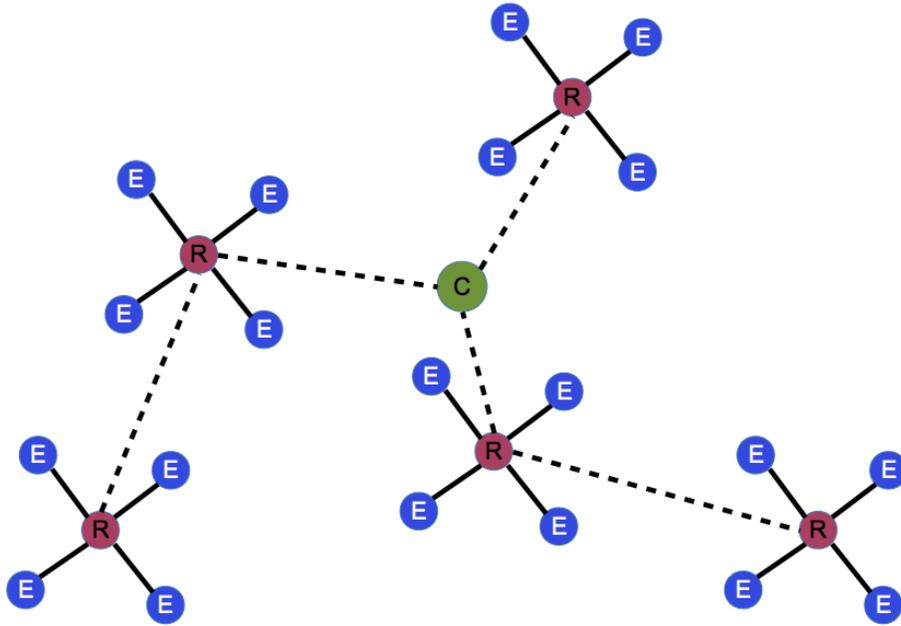


Figure 8.1: Zigbee Network

be directly connected to the coordinator. End devices are low powered which the collects information [74].

Each cluster you see in Figure 8.1 is an IEEE 802.15.4 network which may be configured to beacon enabled mode. My work on cluster networks fits exactly here, Zigbee's clusters can use my dynamic superframe concept to save energy. If we remove the end devices annotated by 'e' from Figure 8.1, the network we left with can be directly applied to my work on multi-cluster networks to have further energy savings.

## 8.2 WirelessHART

WirelessHART is one of the most widely used monitoring protocol for industrial control networks. It based on Time Division Multiple Access mechanism and each device is time synchronized. The physical layer is defined by IEEE 802.15.4 physical layer with frequency hopping spread spectrum built on top.

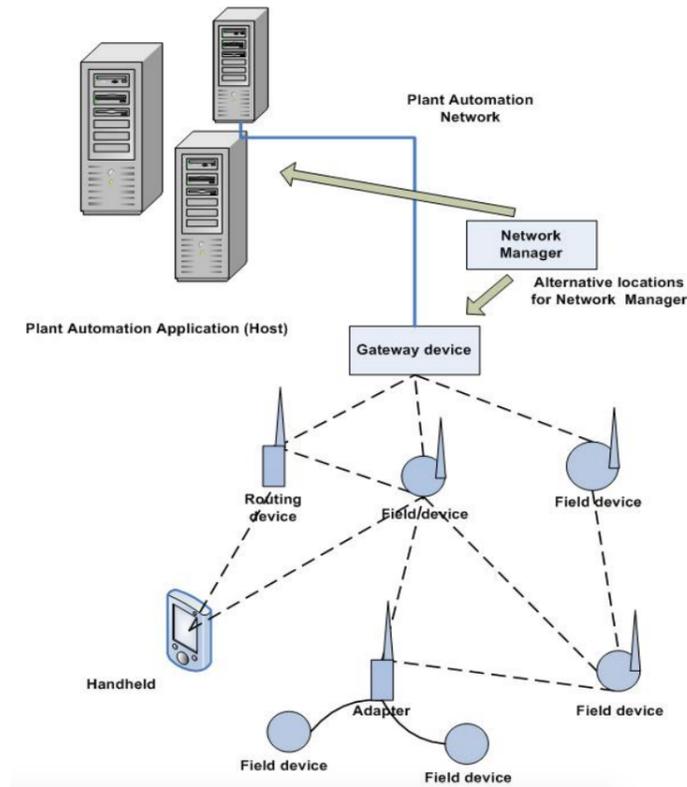


Figure 8.2: WirelessHART Network

Figure 8.2 shows the WirelessHART topology (the picture is borrowed from [27]). Field devices in WirelessHART topology corresponds to sensing-and-relaying nodes of my Architectural Model (Section 3.2), routing device is be considered as a sensing-and-relaying node which do not perform sensing operation. Adapter is needed to convert wired field devices into wireless ones and handheld device is used for configuration, monitoring, and maintenance of network nodes. Gateway device corresponds to the base-station of my model which functions as a bridge between the network manager and the network. Network manager is a centralized entity which is responsible for computing the scheduling for the nodes. Network manager also constantly receives health data from the network such as RSSI values, current packet loss rates, and list of neighbors etc. to maintain healthy time slots allocation for the nodes.

My work on multi-hop networks is a tool which can be used by the network managers to compute routing tables and the time-slots for the nodes. However, a few modifications need to be made for it to comply with the standard. First, WirelessHART has specific requirements for each time-slot. Both transmitter and receiver have specific offset times for clock synchronization and also to compensate for the delay of turning on radios at the beginning of each time slot. An attempt of transmission is followed by another offset time and an acknowledgment message. Moreover, each pair of transmitters are assigned back to back time-slots in case the transmission fails during the first one [75]. The delay variable of my optimal MILP formulation (See Section 7.1) needs to be computed according to these constraints for each possible modulation level such that  $2 \times (\text{packet offset} + \text{time it takes to send a packet using modulation level } m + \text{acknowledgement offset} + \text{time it takes to send an ACK using modulation level } m)$ . Another requirement of WirelessHART protocol is to assign primary and secondary parent nodes for each node to further increase reliability. Primary nodes are the nodes chosen by the optimization problem presented in Section 7.1, Equation 7.1. Network manager can choose the secondary parents by rerunning the optimal formulation with a slight modification; whenever there is a term in the constraints in the form of  $X_l^{i,k}$  (here  $X$  can be replaced with any of the variables used in the Equation), the scope of  $k$  needs to be changed to  $\forall k \in N - p_1^i$  where  $N$  is the set of nodes in the network and  $p_1^i$  is the primary parent of node  $i$ .

### 8.3 IEEE 802.15.4-2015

At the time of writing this dissertation 2015 is the latest version of IEEE 802.15.4 standard. This version preserves the beacon enabled single cluster model of 2006 version and adds new support for multi-hop/multi-cluster networks. It resembles to Zigbee and WirelessHART in many ways [76].

Single cluster model of IEEE 802.15.4-2015 matches one to one with the single cluster model I adapted for my dissertation (see Section 3.1). Hence my dynamic superframe

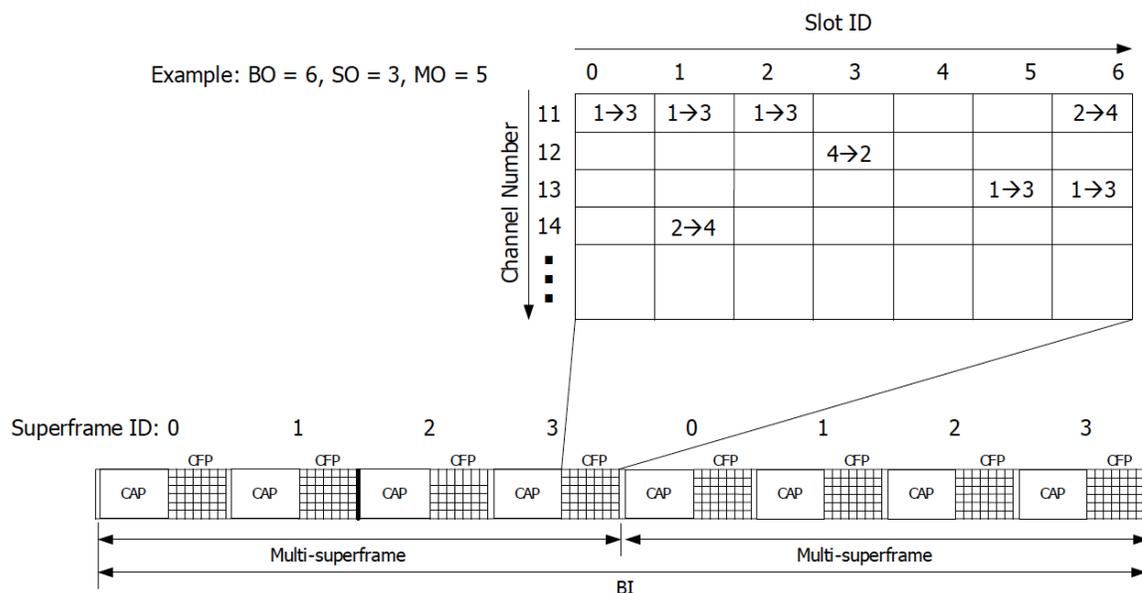


Figure 8.3: IEEE 802.15.4-2015 extended superframe structure as taken from the official standard description.

structure (see Chapter 4) can be applied to IEEE 802.15.4-2015 single cluster networks without a modification.

Multi-cluster network topology is similar to that of Zigbee, depicted in Figure 8.1. In IEEE 802.15.4-2015 terminology, the node annotated as  $C$  is *Super PAN Coordinator*,  $R$  is *PAN Coordinator*, and  $E$  as *Device*. Figure 8.3 shows the extended superframe for multi-hop/multi-cluster networks called *slotframe*. We can see that now at any given time more than one pair of transceivers can communicate simultaneously. For example during slot id 1, node 1 is transmitting to 3 while node 2 is transmitting to 4. The interference has to be handled in time domain and/or frequency domain. My dissertation handles interference in time domain and assumes the existence of frequency hopping to fight with multi-path fading. However, extending my work to include frequency domain is also a rather important future work.

The output of my optimization problem and heuristics (see Chapter 7) can be used to create slotframe scheduling. The *order* value computed by my heuristics (see Section 7.2)

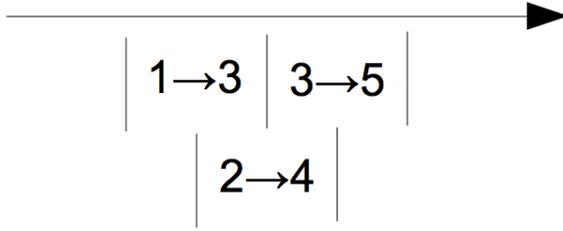


Figure 8.4: How the Optimal Solution (see Section 7.1) might look like.

corresponds to slot id of slotframe. The nodes that have the same order will share the slot with the same slot id. There is however, one minor change that needs to be done to my optimization problem formulation (see Section 7.1) to be fully compatible with slotframe concept. The optimal solution computes start and end times of each node's transmission times from a continuous range of values such that the optimal solution may look as Figure 8.4. However, IEEE 802.15.4-2015 wants the slots to start from a discrete set of values. My heuristic solutions already satisfies such request since each node can only start after all the nodes with lower order number stops transmitting. When we look at the example shown in Figure 8.4, the orders of nodes will be such that  $1 < 2 < 3$  hence node 2 will only be able to start transmitting after node 1 stops and node 3 will do the same after 2. However, this is not necessarily the optimal solution.

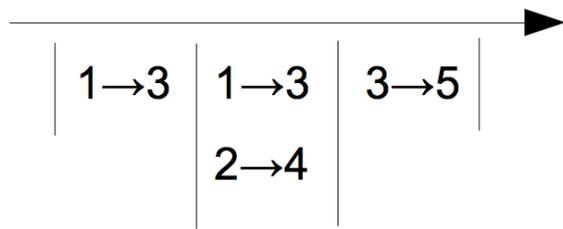


Figure 8.5: After the Optimal Solution has edited to comply with IEEE 802.15.4-2015.

In order to fully comply with the standard, a new constraint such as  $S_i \in \{f_0, f_1, f_2, \dots, f_n\}$  needs to be added to formulation where  $f_x$  is the predefined start of  $x^{th}$  slot. Moreover, the slots has to be length of the transmission duration using the highest modulation level

the system allows. If a node was to use a lower modulation level, it would have to require multiple consecutive slots as shown in Figure 8.5.

## Chapter 9: Conclusion

Applicability of Dynamic Modulation Scaling (DMS) in low-power wireless systems, including IEEE 802.15.4 networks, as an energy management technique has been widely studied in literature and shown that DMS can achieve significant energy savings. In my dissertation I have addressed the problem of applying DMS to real-time WSNs.

My research first addresses the problem of ensuring real-time guarantees while minimizing the overall energy consumption in wireless sensor networks. First, I focused on cluster-oriented superframe communication, the most widely adopted method for providing real-time guarantees in industrial wireless networks. Using Dynamic Modulation Scaling, I studied static and dynamic algorithms for reallocation of slack times of a superframe. I analyzed the effect of interference and dynamic modulation levels on low-power listening. I also introduced a new low-power listening protocol called *hybrid-low-power listening* (HLPL) in order to overcome the interference problem caused by neighborhood. Using the Castalia Simulator I assessed the performance improvements of DMS slack reclaiming and HLPL. My experiments show that dynamic slot readjustment saves a significant amount of energy under highly loaded systems. They also indicate that HLPL overcomes the interference caused by other nodes in the cluster and significantly reduces the overall energy consumption of the system.

Next, I have researched the application of DMS into multi-hop/multi-cluster real-time WSNs. I first presented the Mixed Integer Nonlinear Programming (MINLP) formulation of minimizing total network energy consumption while satisfying the deadline requirements for a DMS enabled time critical network. This formulation picks the most energy efficient parent node and a modulation level for each node in the network. I later showed how to linearize this MINLP formulation to convert it into Mixed Integer Linear Programming

(MILP) problem which is typically a requirement for problem solving engines such as IBM's CPLEX.

However, much the work that has been done so far lacks substantial empirical analysis of DMS on Wireless Sensor Networks (WSNs). As a result, only a generalized mathematical evaluation of DMS were available to the researchers. These evaluations were not enough to for me to move forward with the solution to the problem I have formulated. Hence, my dissertation addressed that gap and provided empirical results that can be used by not just us but future researchers as a complement to currently available mathematical techniques. To accomplished this, I used Ettus B210 Software Defined Radios (SDRs) and configured them according to the 2015 IEEE 802.15.4 standard. I experimented with various Signal-to-Noise-Ratios and corresponding Packet-Delivery-Rates for {2, 4, 8, 16}-PSK as well as DPSK modulations. My results show that increasing PSK modulation levels requires more transmission power increase compared to theoretical scaling functions (such as those shown in [14]). On the other hand, DPSK requires significantly *less* power increase compared to the same scaling function. I have also shown that the scaling ratios between consecutive modulation levels are not equal. Next, I measured the PDRs for {2, 4, 8}-DPSK for distances up to 100 meters. The results have shown that lower modulation levels can achieve higher communication distances. I have also shown the performance difference between 915 MHz and 2.484 GHz and observed 915 MHz has higher packet delivery rate with the same output amplifier gain. Lastly, I have shown that elevation difference between transmitter and receiver affects these results tremendously. 4-DPSK and 8-DPSK had as low as 54% and 28% PDR for elevation difference of 11.6m and 17.48m apart. The same modulations had 99% and 76% PDR respectively for distances up to 20 m with no elevation difference.

Lastly, I designed two polynomial time heuristics and compared them against the optimal solution obtained by CPLEX software. In order to have an accurate network model, I have incorporated the empirical measurements obtained from my SDR tests into my custom simulator. I ran various tests with different network area sizes and base-station locations and

showed our polynomial time heuristics perform very closely to the optimal solution especially for larger deadlines. I have also shown that there is relatively less energy consumption penalty if the deadline of the system were to be halved that is if more and more nodes needed to use 4DPSK rather than 2DPSK. However, as the deadline further reduced and more nodes are required to use 8DPSK to meet the deadline, there is a significant increase in total energy consumption.

## Bibliography

## Bibliography

- [1] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” *Wireless networks*, vol. 8, no. 5, pp. 481–494, 2002.
- [3] A. Cerpa and D. Estrin, “Ascent: Adaptive self-configuring sensor network topologies,” *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 1, pp. 62–62, 2002.
- [4] O. Dousse, P. Mannersalo, and P. Thiran, “Latency of wireless sensor networks with uncoordinated power saving mechanisms,” in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2004, pp. 109–120.
- [5] Z. Kong and E. M. Yeh, “Distributed energy management algorithm for large-scale wireless sensor networks,” in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007, pp. 209–218.
- [6] R. A. Len, V. Vittal, and G. Manimaran, “Application of sensor network for secure electric energy infrastructure,” *IEEE Transactions on Power Delivery*, vol. 22, no. 2, pp. 1021–1028, 2007.
- [7] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis, “Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea,” in *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005, pp. 64–75.
- [8] V. C. Gungor and G. P. Hancke, “Industrial wireless sensor networks: Challenges, design principles, and technical approaches,” *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258–4265, 2009.
- [9] B. Galloway and G. Hancke, “Introduction to industrial control networks,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 860–880, 2013.
- [10] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, “Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors,” in *Proceedings of Fourth IEEE International Conference on Sensor Technologies and Applications*, 2010.

- [11] W. Liang, X. Zhang, Y. Xiao, F. Wang, P. Zeng, and H. Yu, "Survey and experiments of wia-pa specification of industrial wireless network," *IEEE Wireless Communications and Mobile Computing*, vol. 11, no. 8, pp. 1197–1212, 2011.
- [12] V. Cionca, T. Newe, and V. Dadarlat, "Tdma protocol requirements for wireless sensor networks," in *Proceedings of Second IEEE International Conference on Sensor Technologies and Applications*, 2008.
- [13] R. Szabo and et. al., "Framework for smart city applications based on participatory sensing," in *Proceedings of 4th IEEE International Conference on Cognitive Infocommunications*, 2013.
- [14] C. Schurgers, V. Raghunathan, and M. B. Srivastava, "Power management for energy-aware communication systems," *ACM Transactions on Embedded Computing Systems*, vol. 2, no. 3, pp. 431–447, 2003.
- [15] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.
- [16] C. Dong, L.-L. Yang, and L. Hanzo, "Performance of buffer-aided adaptive modulation in multihop communications," *Communications, IEEE Transactions on*, vol. 63, no. 10, pp. 3537–3552, Oct 2015.
- [17] T. Hamachiyo, Y. Yokota, and E. Okubo, "A cooperative power-saving technique using dvs and dms based on load prediction in sensor networks," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*. IEEE, 2010, pp. 7–12.
- [18] B. Fateh and M. Govindarasu, "Energy minimization by exploiting data redundancy in real-time wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 6, pp. 1715–1731, 2013.
- [19] M. H. Anisi, G. Abdul-Salaam, M. Y. I. Idris, A. W. A. Wahab, and I. Ahmedy, "Energy harvesting and battery power based routing in wireless sensor networks," *Wireless Networks*, pp. 1–18, 2015.
- [20] A. Gumusalan, R. Simon, and H. Aydin, "Flexible real-time transmission scheduling for wireless networks with non-deterministic workloads," *Ad Hoc Networks*, 2018.
- [21] B. Fateh and M. Govindarasu, "Joint scheduling of tasks and messages for energy minimization in interference-aware real-time sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 14, no. 1, pp. 86–98, 2015.
- [22] S. Cui, A. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization for coded systems," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 1, Dec 2003, pp. 372–376 Vol.1.
- [23] M. Adler, R. K. Sitaraman, A. L. Rosenberg, and W. Unger, "Scheduling time-constrained communication in linear networks," in *Proceedings of the tenth annual ACM symposium on Parallel algorithms and architectures*. ACM, 1998, pp. 269–278.

- [24] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of 2nd ACM International Conference on Embedded Networked Sensor Systems*, 2004.
- [25] I. 802.15.4e Standard. <https://standards.ieee.org/findstds/standard/802.15.4e-2012.html>. Last accessed on 2017-May-22. [Online]. Available: <https://standards.ieee.org/findstds/standard/802.15.4e-2012.html>
- [26] J. Song, S. Han, A. K. Mok, D. Chen, M. Lucas, and M. Nixon, “Wirelesshart: Applying wireless technology in real-time industrial process control,” in *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008.
- [27] T. Lennvall, S. Svensson, and F. Hekland, “A comparison of wirelesshart and zigbee for industrial applications,” in *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*. IEEE, 2008, pp. 85–88.
- [28] C. Na, Y. Yang, and A. Mishra, “An optimal {GTS} scheduling algorithm for time-sensitive transactions in {IEEE} 802.15.4 networks,” *Computer Networks*, vol. 52, no. 13, pp. 2543 – 2557, 2008.
- [29] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, “Z-mac: a hybrid mac for wireless sensor networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 3, pp. 511–524, 2008.
- [30] M. Salajegheh, H. Soroush, and A. Kalis, “Hymac: Hybrid tdma/fdma medium access control protocol for wireless sensor networks,” in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*. IEEE, 2007, pp. 1–5.
- [31] S. Mehta and K.-S. Kwak, “H-mac: a hybrid mac protocol for wireless sensor networks,” *ITC-CSCC: 2007*, pp. 755–756, 2007.
- [32] L. Sitanayah, C. Sreenan, and K. Brown, “Er-mac: A hybrid mac protocol for emergency response wireless sensor networks,” in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, July 2010, pp. 244–249.
- [33] S. Zhuo, Y.-Q. Song, Z. Wang, and Z. Wang, “Queue-mac: A queue-length aware hybrid csma/tdma mac protocol for providing dynamic adaptation to traffic and duty-cycle variation in wireless sensor networks,” in *Factory Communication Systems (WFCS), 2012 9th IEEE International Workshop on*, May 2012, pp. 105–114.
- [34] Y. K. Rana, B. H. Liu, A. Nyandoro, and S. Jha, “Bandwidth aware slot allocation in hybrid mac,” in *Proceedings of 31st IEEE Conference on Local Computer Networks*, 2006.
- [35] B. Shrestha, E. Hossain, and K. W. Choi, “Distributed and centralized hybrid csma/ca-tdma schemes for single-hop wireless networks,” *IEEE Wireless Communications*, vol. 13, no. 7, pp. 4050–4065, 2014.
- [36] M. H. S. Gilani, I. Sarrafi, and M. Abbaspour, “An adaptive csma/tdma hybrid {MAC} for energy and throughput improvement of wireless sensor networks,” *Ad Hoc Networks*, vol. 11, no. 4, pp. 1297 – 1304, 2013.

- [37] M. Guerroumi, A.-S. K. Pathan, A. Derhab, N. Badache, and S. Moussaoui, "Mmsmac: A multi-mode medium access control protocol for wireless sensor networks with latency and energy-awareness," *Wireless Personal Communications*, pp. 1–38, 2016.
- [38] A. Rajasekaran and V. Nagarajan, "Adaptive intelligent hybrid mac protocol for wireless sensor network," in *Communication and Signal Processing (ICCSP), 2016 International Conference on*. IEEE, 2016, pp. 2284–2289.
- [39] G. Ali, K. H. Kim, and K.-I. Kim, "Adaptive tdma scheduling for real-time flows in cluster-based wireless sensor networks." *Computer Science & Information Systems*, vol. 13, no. 2, 2016.
- [40] M. R. Lenka, A. R. Swain, and M. N. Sahoo, "Distributed slot scheduling algorithm for hybrid csma/tdma mac in wireless sensor networks," in *Networking, Architecture and Storage (NAS), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–4.
- [41] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "A-mac: a versatile and efficient receiver-initiated link layer for low-power wireless," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 4, p. 30, 2012.
- [42] C. Cano, B. Bellalta, A. Sfaïropoulou, and M. Oliver, "Low energy operation in wsns: A survey of preamble sampling mac protocols," *Computer Networks*, vol. 55, no. 15, pp. 3351–3363, 2011.
- [43] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*. IEEE, 2006, pp. 48–48.
- [44] J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic voltage scaling on a low-power microprocessor," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, pp. 251–259.
- [45] T. D. Burd and R. W. Brodersen, "Energy efficient cmos microprocessor design," in *System Sciences, 1995. Proceedings of the Twenty-Eighth Hawaii International Conference on*, vol. 1. IEEE, 1995, pp. 288–297.
- [46] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000. [Online]. Available: <http://doi.acm.org/10.1145/332833.332838>
- [47] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE transactions on wireless communications*, vol. 4, no. 5, pp. 2349–2360, 2005.
- [48] Y. Yu, V. K. Prasanna, and B. Krishnamachari, "Energy minimization for real-time data gathering in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, 2006.
- [49] B. Zhang, R. Simon, and H. Aydin, "Harvesting-aware energy management for time-critical wireless sensor networks with joint voltage and modulation scaling," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 514–526, 2013.

- [50] M. Bandari, R. Simon, and H. Aydin, "On minimizing expected energy usage of embedded wireless systems with probabilistic workloads," *Sustainable Computing: Informatics and Systems*, vol. 11, pp. 50–62, 2016.
- [51] T. Instruments. Sub1ghz radios. Last accessed on 2018-March-5. [Online]. Available: <http://www.ti.com/wireless-connectivity/simplelink-solutions/sub-1-ghz/products.html>
- [52] Microchip-Atmel. Ieee 802.15.4 compliant radios. Last accessed on 2018-March-5. [Online]. Available: <https://www.microchip.com/ParamChartSearch/Chart.aspx?branchID=30041>
- [53] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "Towards an open source ieee 802.11 p stack: A full sdr-based transceiver in gnu radio," in *Vehicular Networking Conference (VNC), 2013 IEEE*. IEEE, 2013, pp. 143–149.
- [54] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.
- [55] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of 4th ACM international conference on Embedded networked sensor systems*, 2006.
- [56] I. 802.15.4g Standard. <http://standards.ieee.org/findstds/standard/802.15.4g-2012.html>. Last accessed on 2015-May-22. [Online]. Available: <http://standards.ieee.org/findstds/standard/802.15.4g-2012.html>
- [57] O. Khader and A. Willig, "An energy consumption analysis of the wireless hart tdma protocol," *Computer Communications*, vol. 36, no. 7, pp. 804–816, 2013.
- [58] M. Bandari, R. Simon, and H. Aydin, "Energy management of embedded wireless systems through voltage and modulation scaling under probabilistic workloads," in *Proceedings of IEEE Green Computing Conference*, 2014.
- [59] F. Easdy, *Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers*. Newnes, 2007.
- [60] B. Fateh and M. Govindarasu, "Joint scheduling of tasks and messages for energy minimization in interference-aware real-time sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 14, no. 1, pp. 86–98, Jan 2015.
- [61] A. Boulis *et al.*, "Castalia: A simulator for wireless sensor networks and body area networks," *NICTA: National ICT Australia*, vol. 83, 2011.
- [62] D. Peditakis, Y. Tselishchev, and A. Boulis, "Performance and scalability evaluation of the castalia wireless sensor network simulator," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010, p. 53.
- [63] E. Research. Ettus usrp b210. Last accessed on 2018-Feb-22. [Online]. Available: <https://www.ettus.com/product/details/UB210-KIT>

- [64] K. D. S. Bandara, A. Melaragno, D. Wijesekera, and P. Costa, "A case study of cognitive radio networks: Secure spectrum management for positive train control operations," in *Spectrum Access and Management for Cognitive Radio Networks*. Springer, 2017, pp. 121–152.
- [65] P. Ho and D. Fung, "Error performance of multiple symbol differential detection of psk signals transmitted over correlated rayleigh fading channels," in *Communications, 1991. ICC'91, Conference Record. IEEE International Conference on*. IEEE, 1991, pp. 568–574.
- [66] F. Edbauer, "Bit error rate of binary and quaternary dpsk signals with multiple differential feedback detection," *IEEE Transactions on Communications*, vol. 40, no. 3, pp. 457–460, 1992.
- [67] <https://forums.ni.com/t5/USRP-software-radio/how-to-set-the-trasmit-power-of-usrp/td-p/2202190>. [Online]. Available: <https://forums.ni.com/t5/USRP-Software-Radio/How-to-set-the-trasmit-power-of-USRP/td-p/2202190>
- [68] M. Park, "Ieee 802.11 ah: sub-1-ghz license-exempt operation for the internet of things," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 145–151, 2015.
- [69] T. Instruments, "Cc2420: 2.4 ghz ieee 802.15. 4/zigbee-ready rf transceiver," 2006.
- [70] M. Barbehenn, "A note on the complexity of dijkstra's algorithm for graphs with weighted vertices," *IEEE transactions on computers*, vol. 47, no. 2, p. 263, 1998.
- [71] I. ILOG, "Cplex optimizer," *En ligne*. Available: <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer>, 2018.
- [72] G. Zhou, T. He, J. Stankovic, T. Abdelzaher *et al.*, "Rid: radio interference detection in wireless sensor networks," in *Proceedings of 24th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, Mar 2005, pp. 891–901.
- [73] J. T. Adams, "An introduction to ieee std 802.15. 4," in *Aerospace Conference, 2006 IEEE*. IEEE, 2006, pp. 8–pp.
- [74] N. A. Somani and Y. Patel, "Zigbee: A low power wireless technology for industrial applications," *International Journal of Control Theory and Computer Modelling (IJCTCM) Vol*, vol. 2, pp. 27–33, 2012.
- [75] A. Saifullah, D. Gunatilaka, P. Tiwari, M. Sha, C. Lu, B. Li, C. Wu, and Y. Chen, "Schedulability analysis under graph routing in wirelesshart networks," in *Real-Time Systems Symposium, 2015 IEEE*. IEEE, 2015, pp. 165–174.
- [76] M. Köstler, F. Kauer, T. Lübker, V. Turau, J. Scholz, and A. von Bodisco, "Towards an open source implementation of the ieee 802.15. 4 dsme link layer," *Proceedings of the 15. GI/ITG KuVS Fachgespräch Sensornetze, J. Scholz and A. von Bodisco, Eds. University of Applied Sciences Augsburg, Dept. of Computer Science*, 2016.

## **BIOGRAPHY**

My biography here will be outdated even within a year after I submit my thesis so please web search Arda Gumusalan and find my LinkedIn (or the current professional networking platform) for most recent bio.