5-2015

DESIGN OF RELIABLE AND SUSTAINABLE WIRELESS SENSOR NETWORKS: CHALLENGES, PROTOCOLS AND CASE STUDIES

Fan Yang
Clemson University, fany@g.clemson.edu

DESIGN OF RELIABLE AND SUSTAINABLE WIRELESS SENSOR NETWORKS: CHALLENGES, PROTOCOLS AND CASE STUDIES

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Electrical Engineering

---

by
Fan Yang
May 2015

Accepted by:
Dr. Kuang-Ching Wang, Committee Chair
Dr. Yong Huang, Committee Co-Chair
Dr. Harlan Russell
Dr. Haiying (Helen) Shen
Dr. Hongxin Hu

ABSTRACT

Integrated with the function of sensing, processing, and wireless communication, wireless sensors are attracting strong interest for a variety of monitoring and control applications. Wireless sensor networks (WSNs) have been deployed for industrial and remote monitoring purposes. As energy shortage is a worldwide problem, more attention has been placed on incorporating energy harvesting devices in WSNs.

The main objective of this research is to systematically study the design principles and technical approaches to address three key challenges in designing reliable and sustainable WSNs; namely, communication reliability, operation with extremely low and dynamic power sources, and multi-tier network architecture. Mathematical throughput models, sustainable WSN communication strategies, and multi-tier network architecture are studied in this research to address these challenges, leading to protocols for reliable communication, energy-efficient operation, and network planning for specific application requirements. To account for realistic operating conditions, the study has implemented three distinct WSN testbeds: a WSN attached to the high-speed rotating spindle of a turning lathe, a WSN powered by a microbial fuel cell based energy harvesting system, and a WSN with a multi-tier network architecture. With each testbed, models and protocols are extracted, verified and analyzed.

Extensive research has studied low power WSNs and energy harvesting capabilities. Despite these efforts, some important questions have not been well understood. This dissertation addresses the following three dimensions of the challenge.

First, for reliable communication protocol design, mathematical throughput or energy efficiency estimation models are essential, yet have not been investigated accounting for specific application environment characteristics and requirements. Second, for WSNs with energy harvesting power sources, most current networking protocols do not work efficiently with the systems considered in this dissertation, such as those powered by extremely low and dynamic energy sources. Third, for multi-tier wireless network system design, routing protocols that are adaptive to real-world network conditions have not been studied.

This dissertation focuses on these questions and explores experimentally derived mathematical models for designing protocols to meet specific application requirements. The main contributions of this research are 1) for industrial wireless sensor systems with fast-changing but repetitive mobile conditions, understand the performance and optimal choice of reliable wireless sensor data transmission methods, 2) for ultra-low energy harvesting wireless sensor devices, design an energy neutral communication protocol, and 3) for distributed rural wireless sensor systems, understand the efficiency of realistic routing in a multi-tier wireless network. Altogether, knowledge derived from study of the systems, models, and protocols in this work fuels the establishment of a useful framework for designing future WSNs.

DEDICATION

To my parents, Geng Yang and Feng Mei, for their unconditional love and support.

To my wife, April Li, for her help, support, and encouragement.

ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Kuang-Ching Wang and Dr. Yong Huang, for their guidance and support throughout my Ph.D. studies at Clemson. In the past few years, I have grown significantly both personally and academically, and I am extremely grateful for their commitment and patience. Their dedication and attention to my dissertation work resulted in profound improvements.

Thanks, too, to the rest of my committee: Dr. Harlan Russell, Dr. Haiying Shen, and Dr. Hongxin Hu. I greatly appreciate their support and encouragement on my work, and also their willingness to bend their schedules to accommodate the meetings and defenses. I am also grateful to Dr. Lei Tang, Dr. Tsutomu Shimotori, Dr. Daxing Zhang, Dr. Vamsi Gondi, and Dr. Samuel Esswein for their valuable help and comments on my research work. Special thanks are given to great technicians in both Department of Mechanical Engineering and Department of Electrical and Computer Engineering at Clemson University.

In addition, I would like to thank all my colleagues in my research groups who discussed with me many valuable ideas. They include Ke Xu, Ryan Izard, Aaron Rosen, Bob Strecansky, Ben Garrison, Glenn Evans, Jr., Sajindra Pradhananga, Changxue Xu, Jingyuan Yan, and Leigh Herran. We had a good time together.

Finally, I am grateful to my family and friends who stand behind me all the time. Their love and friendship have provided all inspiration, motivation, and direction in life.

TABLE OF CONTENTS

Table of Contents (Continued)

Page

Page

Table of Contents (Continued)

LIST OF TABLES

LIST OF FIGURES

List of Figures (Continued)

Page

List of Figures (Continued)

Page

CHAPTER ONE

INTRODUCTION AND RESEARCH MOTIVATION

Wireless sensor technology has been increasingly considered as an alternative for various monitoring purposes, because wireless devices are flexible for deployment, and suitable for mobile subsystems, making setup and data acquisition easy [Willig2005]. The purpose of this study is to understand wireless sensor data transmission performance and propose new designs and technical solutions to address three key challenges in designing sustainable WSNs for industrial and environmental monitoring purposes; namely, communication reliability, operation with extremely low and dynamic power sources, and multi-tier network architecture. Such an understanding is expected to enable reliable and energy efficient wireless data communication in monitoring wireless sensor networks (WSNs), and provide sustainable sensing for monitoring WSNs powered by renewable energy harvesting devices that have low and dynamic power output.

1.1 Wireless Sensor Network in Industrial Applications

Advances in wireless communications and electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate wirelessly in short distances [Akyildiz2002]. They are promising for replacing wired sensors in industrial monitoring systems. Communication reliability is an important facet of dependability and quality of service in most applications of wireless sensor networks. Before industrial or environmental monitoring can take place, it is vital that the communication among sensors be reliable and dependable [Luo2011]. Any network outage, loss of transmitted data, or failure to capture important data decreases the quality of the system. Communication reliability is, therefore, a critical challenge for the successful monitoring operations of WSN.

It's known that some link reliability mechanisms (e.g. MAC layer automatic repeat request – ARQ) can significantly reduce the end-to-end packet loss ratio [Wu2010], while some other WSN applications (e.g. manufacturing monitoring applications systems [Dzapo2009]) require high or even total end-to-end reliability, demanding novel reliable strategies. These studies have not adequately assessed and addressed wireless transmission errors that can occur in such environments of known harsh radio channel properties. The focus of this part of dissertation is to enable reliable industrial adoption of wireless sensor network solutions. Furthermore, such monitoring systems often demand high reliability, high data throughput, and low energy consumption. Existing commercial wireless sensor networks based on standards such as IEEE 802.15.4 have been found to have high probability of transmission errors [Wang2007; Tang2008]. It is of crucial importance that such degradation be properly modeled to assess and improve a sensing system's data transmission performance.

Experimental results identified the dominant cause of transmission errors being multipath effects inside the metallic rotating structure. In [Tang2008], such errors were further found to recur in deterministic regions around the rotation cycle, and the temporal and spatial distributions of such error regions were identified from bit error patterns in transmitted probes. The challenge for applying statistical models in such an environment is that these models are not fundamentally designed to capture sensor networks' 1) more deterministic channel property and 2) more deterministic communication pattern. Moreover, when energy consumption is taken into consideration for wireless network design, most channel estimation algorithms will no longer work as they usually assume unlimited power availability and high processing capability of sensor nodes [Devarajan2012; Zhang2013]. Special transmission protocols are needed to deliver both high data quality and transmission energy efficiency.

It has been extensively studied by previous works that wireless data transmission performance is a function of packet error rate measured by receiver [Khan2012; Chen2013]. Therefore, it is straightforward to capture the key properties of error profiles, e.g., peak and average error rate at different locations, for performance prediction and transmission protocol selection. It should be noted that the error profile for a certain spatial trajectory is an important property that needs to be modeled but has not been studied before. In this dissertation, a simple mathematical model based on surface roughness to capture the key characteristics of specific wireless channels is introduced. Furthermore, mathematical models are derived for selected transmission protocols under typical industrial environment. With these models, throughput can be estimated to guide the selection of data transmission protocol under different conditions. Such protocols, which are also presented in this dissertation, are used to provide throughput and energy-efficiency improvements.

1.2 Wireless Sensor Network with Energy Harvesting Sources

As we speed up and scale up the pervasive deployment of monitoring systems into our critical infrastructures and environment, assuring the sustainability of such systems is an utmost important problem. The utilization of renewable energy sources for wireless sensor systems is useful for sustainable monitoring systems [Kansal2007; Niyato2007; Hasenfratz2010], but very limited today due to a range of reasons. Solar cells and wind turbine generate more energy but are not accessible in many environments and their exposed installations are more obtrusive and vulnerable to weather, theft, or vandalism.

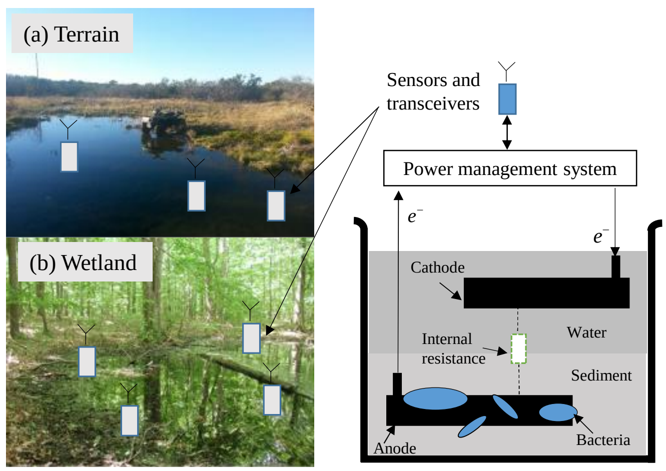

Figure 1.1: Sustainable monitoring of critical infrastructure and environment with MFC-powered wireless sensor networks

Emerging microbial fuel cells (MFCs) based power sources can be enclosed and embedded into terrains, wetlands, and structures. MFCs convert chemical energy in organic compounds to electrical energy through catalytic reactions of microorganisms [Allen1993; Logan2006; Du2007]. They have been envisioned as one of promising green energy sources for sustainable sensing applications [Chen2011]. The advantage of using MFCs as power sources is that they can be seamlessly deployed in natural environments and civil infrastructures as they do not need to be exposed as the solar panels or wind turbines, making them more applicable for pervasive deployment and less susceptible to damages. Figure 1.1 shows a sustainable monitoring infrastructure with MFC-powered wireless sensor networks.

Unfortunately, due to MFCs' high internal resistance and low voltage/power output, MFCs are unable to directly drive most commercial electronic devices [Yang2012]. Various attempts, such as serial stacking and DC (direct current)-DC boosting have been made to increase the MFCs voltage. Serial stacking of MFCs has been proved difficult or ineffective [Meehan2011] to implement in open water environments and may lead to even lower outputs due to voltage reversal [Oh2007]. Using a DC-DC converter to boost the output voltage is a very common approach; however, it is not applicable to MFCs since the MFC output is too low to directly drive the DC-DC converter. In addition, enough energy should also be stored to power the load after the DC-DC converter boosting voltage is achieved [Meehan2011]. As so, a power management system (PMS) is usually required to interface the MFC with the load. The PMS for MFC should be able to raise the voltage to a certain level and accumulate enough power to drive the load intermittently. Pilot studies so far have only managed to demonstrate transmission over one wireless link instead of a network with multiple nodes [Yang2012; Donovan2011].

There is need for designing new PMSs to maximally harvest energy from MFCs. Moreover, despite the wealth of wireless sensor network literature, MFCs present very different and challenging charging properties that are beyond the typical assumptions in the state of the art. For example, the long and highly variable charging time of MFC-based sensor nodes would challenge the feasibility of any existing duty cycling based communications protocols [Ye2004; Polastre2004; Fafoutis2011]; neighboring nodes might never be able to wake up at a common time to initiate communications. There is need to design protocols that are made aware of the energy source's charging properties and regulate their communication strategies in structured ways that can guarantee communication success for such systems to be even considered practical.

1.3 Wireless Sensor Network in Environmental Applications

Beside communication reliability and sustainability, another challenge that most WSNs need to address is network architecture design. Wireless infrastructure design process and architecture can be complex if the system is aimed at monitoring different types of environments. Therefore, well-designed network architecture is critical for fulfilling such requirements [Pakzad2008].

The emergence of wireless mesh networking (WMN) as a supporting infrastructure for sensing networks provides promise to meet the challenges in environmental monitoring network architecture design as it provides flexibility while offering cost-effective solutions. In general, wireless mesh networks consists of three types of nodes: Internet gateways, mesh nodes (also simply known as nodes), and mesh clients. It should be noted that a physical node might function as an Internet gateway and mesh node at the same time. Mesh clients are usually wireless sensors that collect and transmit local environmental data to mesh nodes. Mesh nodes communicate with each other in a multi-path and multi-hop fashion via the wireless links to form the backbone of the infrastructure and relay information packets collected by mesh clients. The network coverage is adjustable by controlling the number of mesh nodes. The mesh backbone can be built using different types of radio technologies. Lastly, the Internet gateway is an Internet connected device responsible for routing all the information packets to a remote server. WMNs provide a cost-effective way to deploy a network and offer services such as Internet connectivity.

Prior work has focused on finding optimal throughput paths in WMNs [Gungor2006; Jain2003] and to this end, various routing metrics such as ETX [Couto2003] and ETT [Draves2004] have been proposed. Typical shortest path routing using hop-count or any of the above metrics can lead to load imbalance and because these metrics are load-agnostic, shortest path routing can lead to situations where some Internet gateways or mesh nodes are overloaded while others are under-utilized. Imbalanced loads and inefficient use of network capacity can lead to throughput degradation. Therefore, a routing protocol is essential to utilize all available paths to the destination and prevent overloading certain nodes in the network. To maintain network utilization while preserving fairness requirements, an efficient routing scheme is critical.

To address this need, this dissertation presents a simple and effective routing scheme based on multi-tier wireless infrastructure. The new network architecture is able to manipulate real-time data flow paths based on link information. To validate the new routing scheme, simulation tools and a special-designed wireless network system are introduced to showcase the improved performance. The architecture and protocol design presented in this dissertation can serve as a useful and pragmatic wireless monitoring framework with centralized control.

1.4 Dissertation Scope and Overview

This dissertation is aimed to 1) understand reliable wireless sensor data transmission performance under fast-changing but repetitive mobile conditions, 2) present an energy neutral communication protocol based on the underlying ultra-low and dynamic charging capability of Microbial Fuel Cells (MFCs), so that such systems can be pervasively deployed and sustainably operated, and 3) study a complete operational wireless system to learn the potential benefits of centralized routing control.

This dissertation is organized as follows. Chapter 2 reviews background and related work on the three challenges that need to be addressed to better understand practical applications of various monitoring WSNs. Chapter 3 addresses the first challenge—communication reliability. Reliability control methods, their performance estimation models and experimental validations are studied and discussed. Chapter 4 addresses the second challenge—operation with low and dynamic power sources. MFCs' output properties are systematically evaluated to serve as the basis for the sustainable network communication protocol design. Chapter 5 addresses the third challenge—routing protocol in multi-tier network architecture. How to provide realistic routing within wireless mesh infrastructure is the focus. The dissertation summarizes in Chapter 6 with suggestions on future work.

CHAPTER TWO

BACKGROUND AND RELATED WORK

This chapter studies wireless sensor network by reviewing three major challenges when implementing reliable and sustainable wireless sensor data transmission: 1) communication reliability, 2) operation with extremely low and dynamic power sources, and 3) multi-tier network architecture.

2.1 Communication Reliability

This section reviews wireless sensor communication performance, wireless channel modeling, and reliability control methods in industrial monitoring applications.

2.1.1 Wireless Data Transmission in Industrial Monitoring

Wireless communication in industrial monitoring is long known to be difficult ever since wireless radios were used in such environments. The extent a radio can tolerate radio propagation effects depends on its circuitry and protocol features. To achieve low cost, IEEE 802.15.4 radios rely on the spread spectrum modulation to provide some inherent tolerance to multipath effects. Radio signals are prone to blockage, multipath fading due to stationary or moving metallic structures near the communicating devices, and radio interferences from machinery and other sources [Rappaport1989]. This brings some challenges for reliable data transmission under harsh industrial environments. For industrial monitoring, throughput and timeliness for transmitting the sensed data are two key performance metrics. For sensor radios, early field tests have revealed that, in industrial settings, multipath propagation, structure obstruction and interferences caused location-dependent, channel-specific, and time-varying communication errors [Tang2010].

In [Werb2005], the authors conducted measurements in a machine room and a compressor house, using six IEEE 802.15.4-compliant radios to measure received signal power and packet loss rate between any two radios. Their results revealed that the received signal power and the packet loss rate were both channel-dependent. Using IEEE 802.15.4-compliant radios, Tang *et al*. presented an experimental study in a university machine shop to measure radio strength and PER to evaluate potential factors for wireless communications, such as distance, stationary and moving obstacles [Tang2006]. In [Tang2007], the same sensor radios were utilized for extensive measurements following a fine grid of locations in a machine shop to investigate the spatial and temporal characteristics of radio channels. In [Tang2012], IEEE 802.15.4 sensor-based measurements were conducted to investigate the effect of stationary and moving forklift on the data transmission performance of wireless sensors. These works confirmed the received power's complex dependency on multipath effects and blocking caused by the surrounding structures.

Other standardized wireless technologies, such as Wi-Fi/IEEE 802.11 and Bluetooth/IEEE 802.15.1, have also been assessed previously [Vohra2012; Nagaonkar2013]. While mature techniques could have been utilized to combat these issues, the simplistic design philosophy of sensor radios has precluded many such options that involve more complex circuitry and higher costs. Given such constraints, sensor radios may not guarantee network connectivity and error-free communications at all times and locations.

2.1.2 Wireless Channel Modeling and Reliability Control Methods

As reviewed by the previous section, the root cause for data communication in sensor networks to be unreliable is the low signal to interference and noise ratio (SINR) at the receiver end due to path loss and signal multipath propagation [Goyal2010; Genender2010]. Wireless channel usually refers to the medium between the transmitting and the receiving antenna, the characteristics of wireless signal changes as it travels from transmitter to receiver. These characteristics depend upon the distance between two nodes, the path(s) taken by the signal, and the environment (building and other objects) around the path.

In traditional channel characterization based on statistical models, the profile of received signal can be obtained from that of the transmitted signal if we have a model of the medium between the two. The power profile of the received signal is usually calculated by convolving the power profile of the transmitted signal with the impulse response of the channel. How to accurately measure the impulse response of wireless channels has been a challenge in most wireless communication systems due to the complex measuring process, not to mention that most of the channel conditions are time-variant. Some works solve the problem by adopting empirical models, e.g. the Friis equation, some works investigate the path loss and the root mean square delay spread of the RF signal [Luo2009]. But these models immediately fail to work in underground or underwater environments. There is a need for simple and basic mathematical models to capture the key characteristics of the specific channel environment.

Besides channel models, transmission protocol is another aspect for improving data transmission performance. A straightforward method to increase SINR and thus increase data throughput is to increase sender's transmit power. The success of avoiding errors by increasing the transmit power suggests that reliable communication can be assured if the minimum required transmit power can be accurately estimated for the sensor location under different conditions. However, the associated time delay and energy consumption during transmit power control process are not desirable in most industrial monitoring applications, especially for those with tight sensor energy constraints.

A classic technique to improve the reliability of data transmission provided by WSN is to increase the redundancy, by either waiting for reports from multiple neighboring sensor motes (spatial redundancy) [Quan2006; Fink2012], by waiting for acknowledgements from the recipient sensor mote (temporal redundancy) [Schmidt2009; Antonopoulos2013], or by including alternative checking data like parity bits in in-network messages (information redundancy) [Schmidt2009; Naderi2012; Guo2012]. For example, channel-coding method computes encoded forms of the data packets at the transmitter such that the receiver can correctly decode them even in the presence of a number of error bits caused during transmission. Typical encode and decode operations are computationally costly. Channel coding methods in essence inject redundancy into transmitted packets in encoded forms. When coding methods are infeasible or undesirable, simpler forms of redundancy can be achieved by simply transmitting multiple copies of the same packet; Automatic repeat request (ARQ) is one such method that relies on receiver acknowledgement to request on demand retransmission of the packet. The simpler computation leads to more bits transmitted than channel coding schemes. Studies have also considered hybrid ARQ schemes, which essentially combine channel coding and ARQ to overcome their individual drawbacks [Tang2009a; Wu2010]; nevertheless, there remain tradeoffs between energy consumption and data transmission error rate. Another form of transmission redundancy can be achieved with adding multiple groups of relay radios between the transmitter and receiver to forward packets in multiple disjoint relay paths that have independent channel characteristics [Quan2006].

If the channel condition can be known, errors can be reduced by allowing the radio to transmit only when the channel is in a good state, which is referred as error avoidance method. In [Deb2001], good and bad channel periods are defined based on transmission error rates being lower or higher than a specified threshold. The work modeled the radio link as a Markov random channel, such that the channel state can be predicted based on prior transmission success history with a statistical Markov model. For the Markov random channel, the error avoidance method yielded 40% to 50% throughput improvement. The model, however, cannot be applied to deterministic channels observed by wireless sensors on rotating structures [Wang2009]. Instead, based on the fact that channel variation on rotating structures is periodical and deterministic, good and bad channel periods can be modeled with a deterministic model. Furthermore, as we speed up and scale up the pervasive deployment of monitoring systems into our critical infrastructures and environment, assuring the sustainability of such systems is an utmost important problem. Many studies utilize renewable energy sources for wireless sensor systems [Hasenfratz2010; Yang2012]. However, energy sources such as solar cells and wind turbine are usually not accessible in many industrial-monitoring environments. The energy consumption problem needs to be solved from system level, that is, the transmission protocols developed from the system models needs to improve throughput without trading off too much energy efficiency. This is another challenge that most existing works fail to address.

In this dissertation, the focus is on identifying challenges of improving wireless transmission performance in a manufacturing monitoring WSN, which is exposed to harsh environments with special wireless channel properties. Specific communication reliability strategies need to be applied. Moreover, it is of crucial importance that performance of these strategies under deterministic harsh radio channel conditions be properly modeled to assess a sensing system's data transmission. However, there are no such existing analytical channel and performance models that consider the typical systems explored in this dissertation.

2.2 Operation with Low and Dynamic Power Sources

In this section, existing microbial fuel cell (MFC) and its power management system (PMS) designs are firstly introduced. The section then focuses on reviewing energy efficient network communication protocol designs.

2.2.1 Microbial Fuel Cell and Power Management System

Wireless sensor networks consist of multiple small and inexpensive devices that can sense, measure, and gather information from the environment. More and more WSNs are harvesting energy from the environments to provide sustainable network operation. PMS, an electronic system aiming to adjust voltage or power, is widely used with various energy harvesting systems with a low-power output. Such energy harvesting systems include MFC, small-size solar cells, and vibration and thermal energy generators, to name a few. A functional PMS should be able to boost a low-voltage input to a high-voltage output to drive commercial electronic devices such as 3.3V for wireless data transmission applications, continuously or intermittently. While commercial DC/DC converters are available for use, most of them are designed to regulate charging at significantly higher currents rather than a few mA currents harvested from low-power energy sources such as MFC. Furthermore, they are inefficient, if still operable, under such low currents.

Usually, rechargeable batteries and super-capacitors are typical devices used in energy harvesting systems to power the sensor nodes [Shantaram2005; Donovan2008; Meehan2011]. Compared with super-capacitor, rechargeable battery has a higher energy density and lower self-discharge rate. However, many energy harvesting systems don't need high energy density devices, and self-discharge rate will not affect the performance of energy harvesting systems too much once the energy storage device is fully charged. Instead, super-capacitor is especially suitable for applications where a large amount of power is often needed for fractions of a second to several minutes, which is the case of

MFC-based applications. The charging voltage of super-capacitor can also be continuously set to meet the optimal rectifier voltage requirement of energy harvesting systems [Buchmann2001]. What is more, a super-capacitor has a longer lifetime than that of a rechargeable battery [Buchmann2001]. A super-capacitor has become a more widely-used energy storage device in energy harvesting systems.

As frequent recharging of these super-capacitors becomes a significant maintenance burden in WSN, the scavenging of energy from environment has been actively researched to provide possible solutions to resolve this problem. How to harvest energy efficiently from environment and make WSNs work reliably in low and discontinuous energy powered condition has emerged as recurring themes in WSN research. In this dissertation, when capacitors are used as energy storage devices, the output power and duty cycle of PMS are affected by the capacity of capacitors [Dewan2010]. Determining the optimal capacitance of energy storage capacitors for MFC applications has been of great interest in order to maximize the usable MFC output power for a given duration. Once a PMS circuit is selected, the PMS performance is largely determined by the selection of associated super-capacitors. The optimization of super-capacitors has usually been conducted through laborious experimental investigations [Dewan2010], which is effective but time-consuming. There are numerous studies on optimizing the charging process of super-capacitors [Peretz2009]; however, most of these optimization methods need additional circuits or devices to manage the charging process, which is not feasible for MFC due to the low power output of MFC. An effective and efficient optimization approach is of great need to optimize the PMS performance, which is also the subject of this work. It should be noted that the comprehensive study of the MFC PMS output performance serves as the basis of the sustainable network communication protocol design with nodes powered by MFCs.

2.2.2 Energy Efficient Communication Protocols

Due to the varying and extremely low power supply, the nodes of WSNs must work in a discontinuous duty-cycling mode. The ideal picture is that nodes wake up when transmitting data and sleep if no data transmission needed. How to determine the duty-cycling protocol is a significant effective actor for the performance of WSNs. While topology control based duty cycling protocols are favored in certain cases, the knowledge of network topology (location driven) is usually not available for all the nodes and the vast energy consumption of coordinators (coordinator driven) makes it even impossible for ultra-low power devices such as MFCs. The major energy components of a wireless sensor include: CPU, listening, transmission, receiving, sleeping and sensing energy. The major sources of sensor node energy waste are packet collisions, idle listing, overhearing and overhead. Therefore, low power design for wireless sensor networks must be accounted for device design, communication protocols, and sensing algorithms. Most wireless sensor devices today adopt IEEE 802.15.4 standard compliant radios [IEEE2003]. One important provision in the standard is the definition of a “sleep mode” that allows power saving by selectively turning off the radio. Radio chipset vendors can provide more than one sleep mode levels. For example, the Texas Instrument CC2420 radio [CC2420] features four different low power levels that incrementally turn off the radio, the processor/clock, and finally the entire chip. Table 2.1 summarizes the power consumption of these sleep modes and the normal operating mode.

Table 2.1: CC2420 operation modes

<table>
<tr><th>CC2420 Modes</th><th>Referred as in Chapter 4</th><th>Power</th><th>Off Components</th></tr>
<tr><td>TX</td><td>Probe transmitting (PB)</td><td>57 mW</td><td>All on</td></tr>
<tr><td>RX</td><td>Idle listening (IL)</td><td>62 mW</td><td>All on</td></tr>
<tr><td>IDLE</td><td>Power down (PD)</td><td>1.4 mW</td><td>Radio off</td></tr>
<tr><td>Power down</td><td rowspan="2">Deep sleep (DS)</td><td>66 µW</td><td>Radio, CPU/crystal off</td></tr>
<tr><td>OFF</td><td>0.066 µW</td><td>All off</td></tr>
</table>

Medium access control is one category of the approaches that have a great impact on the performance of WSNs where the limited resources need to be shared in an efficient and effective manner. In [Yoshida2011], probabilistic retransmission protocol was proposed to improve transmission reliability in sensor networks by having each node calculating the probability of receiving a packet based on its own operating time. However, this type of probabilistic data collection protocol is only useful in dense wireless sensor networks. For most WSNs, transmitters and receivers need to negotiate certain timing protocols to provide reliable and energy-efficient data transmission.

Synchronized methods such as TDMA based MAC protocols and WiseMAC [Hoiydi2004] provide energy efficient network performance by assigning different time frames for node transmitting/receiving and node sleeping, but synchronized wireless sensor networks are usually difficult to implement, especially with limited energy source. Therefore, MAC protocols such as B-MAC [Polastre2004], S-MAC [Dam2003], SpeckMAC [Wong2006] and On-Demand MAC [Fafoutis2011], etc. were proposed to deal with non-synchronized wireless sensor networks, where reliable system performance was demonstrated in their specific environments.

Table 2.2: Drawbacks of low power WSN communication protocols

<table>
<tr><th colspan="3">Protocol category</th><th>Drawbacks</th></tr>
<tr><td rowspan="6">Duty cycling</td><td rowspan="2">Topology control</td><td>Location driven</td><td>Topology knowledge or neighbor information is required.</td></tr>
<tr><td>Coordinator driven</td><td>Coordinators are energy-consuming and how to choose coordinators fairly is a challenging problem.</td></tr>
<tr><td rowspan="4">Power consumption control</td><td>On-demand</td><td>Wakeup radio needed.</td></tr>
<tr><td>Scheduled</td><td>Strict time synchronization.</td></tr>
<tr><td>Asynchronous</td><td>Energy waste on idle listening.</td></tr>
<tr><td>Contention-based</td><td>Backoff schemes with energy waste on ACKs and retransmissions</td></tr>
</table>

Beside the link-layer MAC protocols, scheduling policies are also critical to sensor network performance. In [Pantelidou2008; Joeph2009; Khouzani2011], joint power control, scheduling and routing polices were proposed for networks with multiple transmitters, where linear/nonlinear programming problems were formulated to solve for optimal nodes' sleep/wake-up strategies. There are some other studies that formulated abstracted resource allocation problems to analyze the stability and capacity of renewable energy networks [Moser2010]. The sensing applications can also adopt algorithms that retrieve data from "just enough" sensor nodes to minimize unnecessary redundancy, provided that the total number of deployed nodes is denser than the needed resolution of the monitored physical field [Audet2011]. Table 2.2 categorizes the drawbacks of current low power discontinuous wireless sensor network communication protocols.

There remains a major knowledge gap to realize such systems in practice. In particular, existing protocol solutions based on heuristics have not provided a structured paradigm for assessing and assuring the sustainable network operation. The purpose of this part of research is to introduce a sustainable protocol for WSNs powered by extremely low and dynamic energy supplies, and present duty cycle modeling for the proposed protocol to offer guidance for practical system design.

2.3 Multi-tier Network Architecture

In general, wireless sensor network architecture consists of three major parts: data acquisition, data transmission and data analysis. The focus of this part of dissertation is data transmission, which involves relaying data to remote servers across a wireless transit system. In this section, network architectures for remote monitoring and efficient routing strategies in wireless mesh network are reviewed.

2.3.1 Network Architecture for Remote Monitoring

Existing wireless systems often use a star topology with a single base station; the base station in turn provides communicating with multiple sensors over long-range data links [Wark2007; Werner2005]. More detailed reviews of wireless data transmission networks for remote environmental monitoring can be found in [Rogan2004; Glasgow2004]. These networks share a similar architecture, comprising a data acquisition unit that collects information from the environment and a server unit that receives data over a communication link. Therefore, coverage scalability is the main issue for such monitoring systems. It should be noted that one of the most important characteristics that differentiates existing systems is the distance between the data acquisition units and the server unit. Several studies describe monitoring systems focused on indoor, short-range monitoring [Zhou2007; Junnila2010]; others describe monitoring and control systems for civil telemetry applications [Ciubotaru2006; Pines1998]—still relatively small in scale monitoring compared to large-scale, wide area environmental monitoring programs. The designs used in these systems are not readily applied in remote environmental monitoring systems, as the strategies for designing long-range communication systems are significantly different [Celandroni2013]. Most existing long-range monitoring systems adopt satellite communication systems for reliable packet transmission [Cardell2004] [Xue2010]. A detailed review of such systems can be found in [Celandroni2013]. While satellite communication is a suitable strategy due to the large covered area it supports and the reliable communication performance it provides, deploying satellite base stations is not scalable; the base stations are cumbersome and costly. Therefore, wireless mesh network is largely used for remote monitoring purposes as 1) its system is kept simple to avoid the complexity of developing and managing wireless infrastructure solutions for different applications; 2) its multi-tier network architecture provides flexibility for using different radio technologies to implement data uplink transit. Energy efficient routing protocol for wireless mesh network is the research focus in this section.

2.3.2 Routing in Wireless Mesh Network

There are a significant number of research works that focus on management of WMNs and numerous routing algorithms have been proposed [Baumann2008; Singh2011]. Some of the routing schemes for wireless networks with monitoring are focused on specific environments, whose typical objective is to maintain communication links between mobile stations. Providing reliable connectivity, however, is not sufficient for most WMNs as mesh network users demand more services and sustainable network operation.

Other routing schemes have been proposed to maximize throughput in WMNs and are generally divided into proactive routing, reactive routing and hybrid routing [Rao2011]. These protocols increase the bandwidth allocation of individual mesh nodes and typically achieve high overall network utilization. However, they do not consider the issue of fairness and mesh nodes that are several hops away from the Internet gateway may suffer from a decrease in bandwidth. One way to improve mesh network performance in WMNs is to exploit multiple channels by adding more radios to a single mesh node. Kyasanur et al. investigated the network capacity region on random multi-hop, multi-radio, and multi- channel wireless networks [Kyasanur2005], while Li et al. studied a similar problem with more complex interference model [Li2008]. However, adding more radios involves additional hardware, an unviable option in many practical monitoring site deployments.

Additionally, studies are conducted on joint channel assignment and routing [Alicherry2005], joint routing and link scheduling [Joseph2009], and throughput optimal MAC designs [Rad2007]. In [Yang2005], a novel routing scheme for interference-aware load balancing in WMNs was proposed by formulating a theoretical load-balancing problem. While all these works provide valuable insight on the design of routing protocols for optimal mesh network performance, their protocol implementations and analysis are based on theoretical calculations or simulations and lack a demonstration in a real-world application. Other works propose techniques to balance the network traffic load [Kashanaki2012; Panicker2013]. These solutions require the exchange of network load information between Internet gateways using distributed algorithms to sending notification packets to some sources, increasing overhead traffic in the network. Moreover, only the traffic going through Internet gateways are considered and balanced in these works, neglecting the fairness issues with regard to the network's mesh nodes.

Furthermore, some studies exploit the advantage of multi-path routing. In [Jain2003] the authors consider the problem of optimal multi-path routing and a similar problem is addressed in [Kodialam2003]. This study deals with the joint problem of routing and scheduling of multi- path flows and assumes that each wireless station is equipped with a single radio and that the stations use orthogonal channels to avoid interference. These studies have shown that multi-path routing maximizes overall traffic flow while providing fair service and bandwidth guarantees. However, these methods face difficulties in flow management, since the single flow between each source and destination pair may be divided into multiple small flows to multiple routes and thus packets may arrive out of order. Managing a large number of small flows in traditional routers and maintaining the order of the packets at the aggregation point is not a pragmatic approach. They also generate additional communication and computation overhead on the network nodes [Ganjali2004]. It should be also noted that in [Ganjali2004], the authors claim that, in practice, the load distribution obtained by multi-path routing is essentially similar to the single path routing, unless a very large number of paths are used (which is not practically feasible). Table 2.3 summarizes the different routing protocols for wireless sensor network.

Table 2.3: Routing protocols for wireless sensor network

Routing Scheme	*Protocol*	*Characteristics*	*Advantage*	*Disadvantage*	*Complexity*	*Prototype*
Reactive routing	Ad-Hoc On-demand Distance Vector Routing Protocol; Temporally-Ordered Routing Algorithm routing protocol; Load-Balancing Curveball Routing	Routes established on demand	The connection setup delay is lower	Intermediate nodes may have stale entries	It varies by protocol. Optimization complexity increases exponentially with number of established routes	Distance vector routing is largely used in traditional wired network; other protocols are based on ns2 simulations
Proactive routing	OLSR; Link Quality Source Routing; Witness Aided Routing	Routes to all destinations within the network are known and maintained before use	There is no route discovery delay associated with finding a new route	It uses power and network resources in order to propagate data about possibly unused routes	Simple. They work with thousands of nodes with very little CPU power on 200 MHz embedded devices.	Link state routing is largely used in traditional wired network; other protocols are based on ns2 simulations
Geographic routing	Location-Aided Routing protocol; Distance Routing Effect Algorithm for Mobility; Blind Geographic Routing	Utilize location information for forwarding in large mesh networks	Quick route discovery	Accurate location information is needed	Simple. Involve route request, route reply, and route error packets	Simulations and some GPS-enabled testbeds
Hierarchical routing	Cluster Based Routing Protocol; Core Extraction Distributed Ad-Hoc Routing; Dynamic Address Routing	Divide network into clusters and perform routing for better scalability	Easy to configure on large networks and is more intuitive at selecting the best route	Cluster head consumes more power; how to select cluster head is a challenge	It varies. Additional resource is needed for cluster head	Simulations and some IEEE802.15.4 based testbeds
Multi-radio and multi-channel routing	Various multichannel routing protocols	Utilize multiple channels to increase throughput and find better routes	Increased throughput and more prone to interferences	Additional hardware; It needs to consider channel assignment constraints and switching cost	Number of neighbors n; number of channels Ch; Then local time complexity is O(n*Ch)	Only simulations
Multi-path routing	Split Multi-path Routing; On-demand Multi-path Routing; Routing On-demand Acyclic Multi-path	Simultaneous, parallel transport over multiple carriers	Better utilization of available bandwidth by creating multiple active transmission queues	Moving to the alternative path will incur a potentially disruptive period during which the connection is re-established	Must perform iterative linear programming process. Complexity grows exponentially with available paths	Most simulations; some alternative TCP enabled wireless testbeds
Multi-cast routing	On-Demand Associativity-Based Multi-cast; Ad-Hoc QoS Multi-cast; Content Based Multi-cast	The router generates a routing table with the multicast group of which it has knowledge with corresponding distances. When a multicast packet is	Easy to share information between routers to facilitate the transportation of IP multicast packets among networks	Periodic reflooding necessary to detect new hosts	Simple	Most used in traditional Internet backbone network

		received by a router, it is forwarded by the router's interfaces specified in the routing table.				
Centralized routing	Centralized QoS scheduling; Centralized channel assignment and routing; Logic based distributed routing	Controller calculates optimal routes	Easy for mesh nodes to setup optimal routes	Implementation complexity in controller	Simple at mesh nodes	Only simulations

From Table 2.3, it can be seen that the challenges of designing realistic routing protocols for WMN infrastructures in environmental monitoring lie in: 1) Simplified mesh node, rendering the feasibility of multi-radio routing protocols and other protocols with high computational overhead. 2) Assumptions of accurate node locations, which are not available in most remote monitoring applications. 3) Assumptions of multi-path packet traffic, but most of existing works failed to provide pragmatic solutions. Lack of prototype demonstrations in existing works raises the demand for simple protocol implementations. Therefore, a simple wireless infrastructure and efficient routing scheme are needed.

2.4 Challenges

The research objectives of this dissertation are: 1) understanding reliable wireless sensor data transmission performance under fast-changing but repetitive mobile conditions, 2) presenting an energy neutral communication protocol based on the underlying ultra-low and dynamic charging capability of energy harvesting devices, and 3) studying a multi-tier architecture that provides efficient and realistic routing, To achieve these objectives, wireless sensor network design problem needs to address various challenges: modeling of wireless channels, performance modeling of reliable data transmission, the engineering of new network protocols for sustainable energy neutral operation, and the realistic design of centralized routing protocol in multi-tier wireless network. In particular, this research work addresses:

1. How to design reliability control methods, and present analytical wireless channel models and data transmission performance models under harsh and dynamic wireless environments.
2. How to design network protocols that provide sustainable sensing for wireless sensors powered by low and dynamic energy sources and how can the proposed network protocol support sustainable communication performance.
3. How to design efficient and realistic routing protocol within the multi-tier wireless network architecture.
4. How to design and conduct testbed-based experimental studies given the proposed protocols.

The ultimate goal of this research is to conduct studies on designing reliable and sustainable wireless sensor networks. To account for realistic operating conditions, the studies will implement three distinct WSN testbeds: a WSN attached to the high-speed rotating spindle of a turning lathe, a WSN powered by a microbial fuel cell based energy harvesting system, and a WSN with a multi-tier network architecture. With each testbed, models and protocols are extracted, verified and analyzed.

CHAPTER THREE

SENSOR DATA TRANSMISSION MODELING AND PREDICTING WITH RELIABLE COMMUNICATION STRATEGIES

Wireless sensors have various applications such as smart cities, smart environment monitoring and industrial control. However, before industrial or environmental monitoring can take place, it is vital that the communication among sensors be reliable and dependable. Hence, there are needs to identify individual channel environments for specific wireless sensor network applications and improve data transmission performance in terms of throughput and energy efficiency.

In this chapter, a novel protocol selection process for wireless data transmission considering sensor locations and deterministic channel properties is introduced. Two of the most widely-used protocols to improve communication reliability--ARQ and error avoidance are studied due to their implementation simplicity and better performance under the special channel properties, respectively. A novel link characterization method based on surface roughness is adopted to compare and analyze performance improvements.

3.1 Wireless Data Transmission Performance Modeling Framework

The proposed wireless data transmission performance modeling framework is composed of three parts: 1) model constraints, 2) model metrics, and 3) model parameters. Model constraints include channel environment, communication system, and energy consumption. In this work, data transmission performance is measured by two important metrics: throughput and energy efficiency. For specific channel environments, the channel parameters of mathematical model contain location dependent information that can be used to predict performance. In this section, model metrics and channel parameters for deterministic environments are firstly presented. A novel channel characterization method based on surface roughness parameters is then introduced. Surface roughness is used to characterize the shape of the error profile for a specific wireless channel. Furthermore, data transmission performance is improved by the protocol parameters that are developed from the deterministic channel information. It's worth noting that the protocol parameters should be protocol specific. Figure 3.1 illustrates the diagram of the modeling framework.

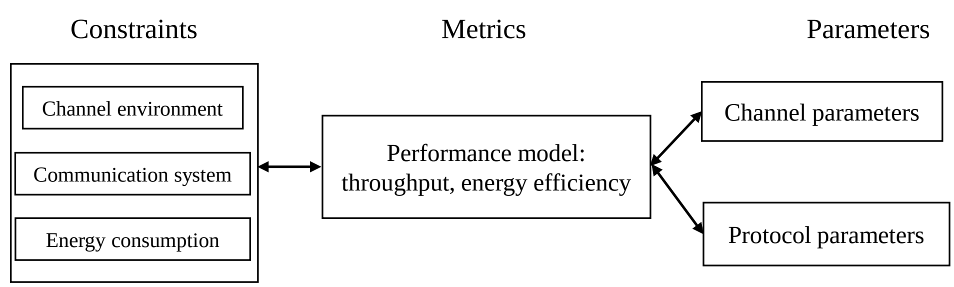

Figure 3.1: Wireless sensor data transmission performance modeling framework

3.1.1 Model Metrics

The most important metrics used for measuring wireless system performance are throughput and energy efficiency. In wireless data communication, throughput and energy efficiency heavily depend on the transmitter's transmission timing pattern. Many sensor systems report sensor data at a constant rate, while there are other systems aiming at transmitting as much data as possible in a short time so that the wireless channel is nearly saturated. In this section, two types of most widely-used wireless data transmission schemes are considered: constant packet rate scheme and saturated scheme. Both schemes are studied for throughput and energy efficiency metrics.

A. Throughput

Throughput is defined as the average rate of successful data transmission over the wireless link. In a constant packet rate transmission scheme, the transmitter transmits one packet in each packet interval. As long as all packets are eventually received successfully within the packet interval, the throughput will be equal to S/I, where S is the packet size in bits and I is the packet interval. On the other hand, if any packet transmission fails, the throughput will degrade proportionally to the probability of such events. Let the probability of any packet successfully completing transmission be defined as the average successful delivery ratio (SDR), then the constant packet rate throughput must be

$$TP_{CPR} = \frac{S}{I} SDR\,. \tag{3.1}$$

For the data transmission system to be stable and the throughput model to be valid, the packet interval must be no shorter than the average time needed to successfully complete a packet’s transmission. Otherwise, the transmitter can become persistently backlogged and achieve only the saturated throughput. When transmitting in saturation, throughput equals to the ratio of packet size S to the average time T_{ave} for the receiver to receive a packet successfully. For example, if retransmission is permitted, T_{ave} is the time needed to complete the average number of transmission attempts until it is received successfully. The saturated throughput is calculated as

$$TP_{Saturated} = \frac{S}{T_{ave}}. \tag{3.2}$$

B. Energy Efficiency

For modeling energy efficiency, it is assumed in this dissertation that every packet has the same size and each transmission takes equal energy. Energy efficiency is typically measured as the average amount of successfully delivered information per unit of consumed energy. Therefore, the energy efficiency can be defined as

$$Eff = \frac{B}{\mu C_T} = \frac{TP \cdot t}{\mu C_T} \tag{3.3}$$

where B is the total successfully delivered information; μ is one unit of consumed energy for transmitting one packet, C_T is the total packet transmission count needed to transmit all data, hence μC_T denotes the total consumed energy for completing all data transmissions, which numerically equals to C_T as μ is normalized to 1; t is the time interval for the whole measurement; TP is the average throughput, which equals to B/t. For constant packet rate and saturated transmission schemes, TP is equal to TP_{CPR} and $TP_{Saturated}$, respectively.

3.1.2 Channel Parameters

For a general wireless communication channel, its error profile depends on channel conditions at different locations where packets are transmitted. Therefore, channel error profile is essential for deriving other channel parameters, such as surface roughness parameters.

A. Channel Error Profile

It is generally considered that the various bits of a same packet may experience different channel conditions, therefore, the packet error sequence may include consecutive error bursts and error-free bursts. It has been verified by many wireless communication researches that bit error rate (*BER*) at a specified location is a function of signal amplitude [He2013; Talha2011]. It is also shown that the wireless signal amplitude at a certain location is affected by the constructive and destructive effects of multi-path signals [Tang2010]. Both the throughput and energy efficiency depends on the wireless link's packet error rate (*PER*), which is a location-dependent property. Packet error rate at any location θ, defined as the *local packet error rate*, depends on the local bit error rate *BER* at location ¯ for both static and moving systems.

While these findings explored the characteristics of an error profile from a physical model point of view, the methodology is not straightforward to be applied for various environments as determining the effects of multi-path wireless signals can be tedious for most applications. However, the physical model findings also indicate that the error profile of a wireless link has the following fundamental properties:

1) Error profile is location-dependent;

2) The shape of error profile has spatial characteristics, such as peaks and valleys.

B. Surface Roughness Parameters

Surface roughness, as a measure of surface textures, is of great importance for various manufacturing and tribological applications, such as friction, contact deformation, and positional accuracy. Generally, surface roughness is quantified by the vertical deviations of a real surface from its datum surface, and the surface profile $R(\bar{\ })$ is used to characterize the shape of surfaces as follows

$$R(\bar{\ }) = f_R[h(\bar{\ })] \tag{3.4}$$

where $h(\bar{\ })$ is the relative height of the surface at individual location $\bar{\ }$. $f_R[\cdot]$ is a given function to characterize a certain feature of the surface texture. Depending on the function being utilized, surface roughness parameters can be classified as amplitude, spacing, and hybrid parameters [Gadelmawla2002].

Considering the spatial characteristics similarity between channel error and surface roughness profiles, the approach of surface roughness characterization is proposed to represent a physical communication link with deterministic channel property between the transmitter and receiver by the shape of its error profile. The characterization, which is based on surface roughness parameters, is used to capture the spatial characteristics of an operating environment, such as error profile peaks, valleys, and their spacing. In this dissertation, we introduce two important surface roughness parameters, namely the amplitude and spacing parameters, to capture the key properties of a wireless channel: amplitude and spacing parameters.

The amplitude parameter: arithmetic average height (R_a)

This parameter is defined as the average absolute deviation of the error profile irregularities from the mean line over one sampling length as shown in Fig. 3.2. This parameter is easy to define, easy to measure, and gives a good general description of error profile height variations. The mathematical definition of the arithmetic average height parameter is

$$R_a = \frac{1}{l}\int_0^l \left|y(\text{¯})\right| d\text{¯} = \frac{1}{l}\int_0^l \left|\text{PER}(\text{¯})\right| d\text{¯} \tag{3.5}$$

where l is the sampling length, y is the error deviation and x is individual location. The mean line is the line that divides the error profile so that the sum of the deviations of the error profile height from it is equal to zero.

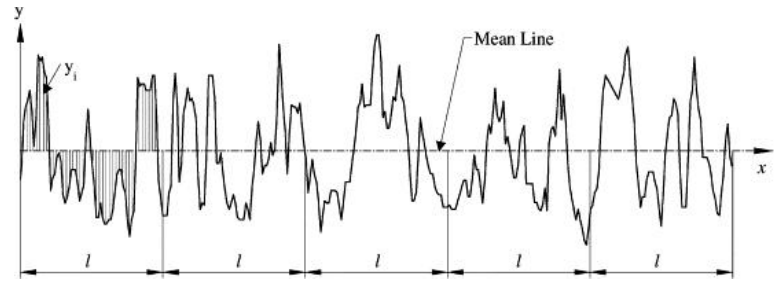

Figure 3.2: Definition of the arithmetic average height (R_a) [Gadelmawla2002]

The spacing parameter: mean spacing at mean line (R_s)

This parameter is defined as the mean spacing between error profile peaks at the mean line and is denoted as R_s. The error profile peak is the highest point of the error profile between upwards and downwards crossing the mean line. The spacing measurement starts from the intersection of the downwards curve and mean line. It ends at the intersection of the immediate upwards curve and mean line. The width of the shaded regions in Fig. 3.3 shows an example of such measurements. This parameter can be calculated from the following equation

$$R_s = \frac{1}{m}\sum_{i=1}^{m} R_s(i) = \sum \text{¯}_j,\ \text{PER}(\text{¯}_j) \le \text{P}_{mean} \tag{3.6}$$

where m is the number of error profile peaks at the mean line. In some applications R_s can be also calculated as the mean spacing at a selected line that is in parallel to the mean line. For example, there are many environments where an error threshold is used to distinguish low and high error regions of a physical wireless link. It is straightforward to calculate mean spacing based on the error threshold line, as the effects of low and high error regions can be taken into account using the alternated calculation.

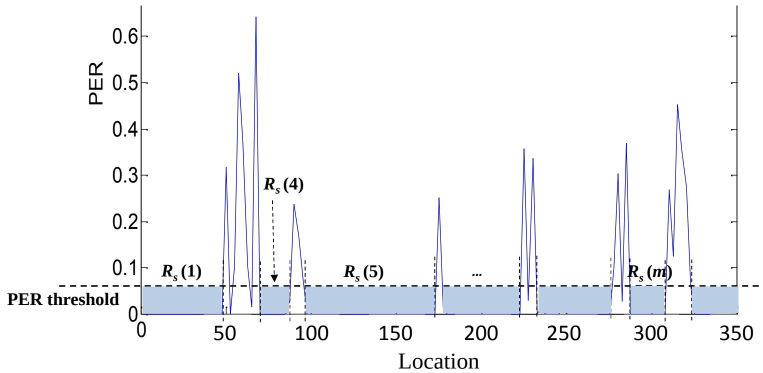

Figure 3.3: The spacing parameter: mean spacing at PER threshold line

In summary, using surface roughness to characterize the error profile has the following advantages:

1. The characterization is easy to define, measure, and calculate. It provides simple abstraction of an error profile.

2. The amplitude and spacing parameters provide key properties of an error profile, such as error heights and low error regions. This characterization along with performance models can be used together to easily identify what transmission protocol should be adopted under certain wireless environments and how much performance improvement it can provide.

3. Without using surface roughness parameterization, two things are needed to select a transmission protocol: *A*) performance models with *BER*/*PER* profile as the input, and *B*) performance comparison among all protocol candidates. It's usually

complicated to derive performance models for different protocols. Fortunately, using the approach of surface roughness characterization, transmission protocol decision can be made based on the parameters extracted from the error profile. This process is simple and deriving complex performance models to determine a transmission protocol is avoided once we utilize this framework.

It is important to note that although the proposed channel characterization can be applied to any wireless communication system with deterministic channels, the derivation of the channel parameters is highly dependent on the specific communication protocol and wireless environment. Mathematical derivation of the channel parameters will be shown in the case study at following sections.

3.1.3 Protocol Parameters

To improve wireless data transmission performance, some link reliability mechanisms (e.g. MAC layer automatic repeat request - ARQ) are used to significantly reduce the end-to-end packet loss ratio [Wu2010]. With ARQ, the transmitter detects a failed transmission when an acknowledgement packet is not received within an ACK timeout period after it finishes transmitting a packet. Upon each failed transmission, the ARQ function schedules an immediate retransmission. A single packet can be retransmitted for a specified number of times, referred as the total transmission count limit. If that packet still cannot be correctly received, the packet is discarded and a next packet's transmission is scheduled. ARQ is one of the transmission protocols that are investigated in this work. It's noted that ACK timeout period t_{ack} and total transmission count limit N are two protocol parameters that will affect data transmission performance.

It’s easy to notice that ARQ improves throughput at the cost of more retransmissions, indicating more energy consumption. If intelligence is added into packet retransmissions, throughput and energy efficiency can be improved at the same time. Error avoidance, where transmitter avoids transmitting packets in a high-error region and resumes transmission in the next feasible low-error region, is one of such “intelligent retransmission” protocols. The error avoidance algorithm that will be investigated in this work, has two phases: the training phase and the operational phase. In the training phase, the transmitter transmits probe packets to the receiver so that it collects sufficient transmission success and failure history to establish an *error location profile*. In the operation phase, the transmitter avoids transmitting packets in high-error regions. It can be seen that error avoidance is most suitable for moving-transmitter environment. Such environment is widely available in many wireless sensor network applications, e.g. rotating industrial monitoring systems. The low-error region threshold *Th* is the protocol parameter, which will greatly affect data transmission performance.

After identifying the protocol parameters, channel parameters can be derived based on specific channel conditions. The error profile of a wireless link is easily measured or calculated. Surface roughness parameters are easily calculated using Eq. (3.5) and (3.6). Data transmission performance in terms of throughput and energy efficiency can be calculated using Eq. (3.1-3.3). Surface roughness parameters are then combined with performance models to identify what transmission protocol should be adopted under certain wireless environments and how much performance improvement it can provide. It can be seen that this framework enables easy capture of the spatial characteristics of general wireless channels and provides easy guidance on protocol parameter selection.

3.2 Throughput and Energy Efficiency Estimation Models

Previous section has shown how the simple framework can be used for performance analysis. In this section, an example based on rotating machine use case and detailed implementations of two reliable transmission control methods: ARQ and error avoidance algorithms are presented. While these methods are not constrained to a specific radio technology, their practical implementation must certainly be tailored to the target platform's radio technology and parameters. For accuracy and clarity of presentation, descriptions in this dissertation assume the MICAz radio based on the IEEE 802.15.4 protocol [MICAz]. Throughput and energy efficiency estimation models under a specific rotating structure environment are then discussed.

3.2.1 Reliable Transmission Control Methods

A. ARQ

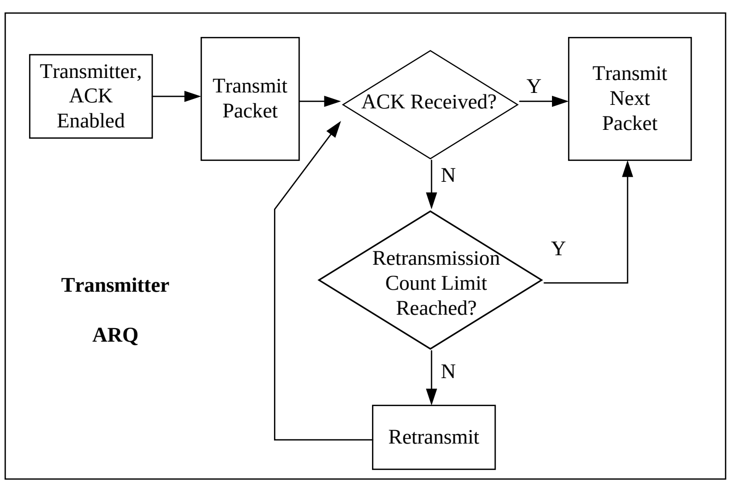

Figure 3.4: Flowchart of implementing ARQ

The ARQ implementation flowchart is shown in Figure 3.4. With ARQ, the transmitter detects a failed transmission when an acknowledgement packet is not received within an ACK timeout period t_{ack} after it finishes transmitting a packet. According to [IEEE2003], t_{ack} = 54 symbol durations = 864 microseconds. Upon each failed transmission, the ARQ function schedules an immediate retransmission. A single packet can be retransmitted for a specified number of times, referred as the total transmission count limit. If that packet still cannot be correctly received, the packet is discarded and a next packet's transmission is scheduled.

B. Error Avoidance

In error avoidance algorithm, the transmitter and receiver clocks are synchronized prior to entering the training phase. In the training phase, the transmitter sends probe packets continuously at a specified interval. Each probe packet carries an incrementing sequence number, and the receiver records the timestamps of all received probe packets. Since the probes are sent in fixed intervals, the receiver can detect lost probes by detecting gaps in received probes' sequence numbers and calculate lost packets' timestamps by interpolation. Given the rotation speed, the received and lost probes' timestamps can be mapped correspondingly to locations on the rotation circumference with respect to a common origin determined as the transmitter location at the time when the first probe is transmitted. Thus, a histogram of received and lost probes at all locations around the circumference (i.e., the number of received and lost packets at each location i on the circumference) can be created with a specified spatial resolution. Let the histogram of received packets be $H_R(i)$, the histogram of lost packets be $H_L(i)$, for

i=0°~360°, the corresponding packet error rate profile can be found as $E(i) = H_L(i)/(H_R(i)+H_L(i))$, for i=0°~360°.

Consider the example in Figure 3.5. Let the first probe's received timestamp be t_0 and the k_{th} probe's received timestamp be t_k, the instantaneous location of the transmitter at t_0 can be defined as the reference origin for the packet error rate profile and the transmitter may have travelled less than one round or more to reach the instantaneous location at t_k. It is therefore apparent that each time the training phase is repeated, the reference origin is different, and the packet error rate profile must be interpreted on the latest origin. If probe k is correctly received by the receiver, the transmitter instantaneous location will be calculated as

$$i = \frac{[\mu \times (t_k - t_0)] \bmod 360^\circ}{\bar{\Delta}} \tag{3.7}$$

where $\bar{\Delta}$ is the degree resolution which equals to 1° in this case, μ is the rotation speed in degrees per second and the timestamps are in seconds.

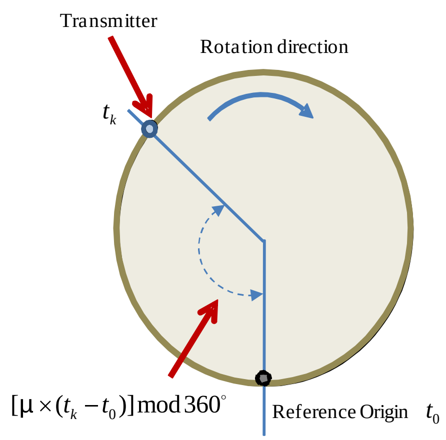

Figure 3.5: Example of transmitter location calculation from probe timestamps

On the other hand, if probe k is lost, its timestamp and location must be determined using interpolation. Let probe m and n be the correctly received probes before and after probe k, then probe k's timestamp is

$$t_k = t_m + \frac{(t_m - t_n)}{(m-n)} \times (k-n) \tag{3.8}$$

and the corrsponding location can be found similarly as Eq. (3.7). With sufficient probes observed at uniformly random locations, the histograms H_L, H_R, and thereby the packet error rate profile E can be found. At the end of the training phase, the receiver identifies all regions with packet error rates less than or equal to the specified threshold Th as low-error regions. A list of the starting and ending positions of low-error regions L_R and the reference origin timestamp t_0 are sent to the transmitter. Upon receiving this information, the transmitter enters the operation phase and starts sending packets based on the error avoidance algorithm.

Error avoidance and ARQ can be combined together, such that the transmitter retransmits a packet when it detects a failure. Note that error regions are defined based on a finite packet error rate threshold; transmission errors may still occur in a low-error region. The combination can be done in two ways. If the retransmission occurs immediately at the end of an ACK timeout, the method is referred to as *partial error avoidance*. On the other hand, if the retransmission is also scheduled with the error avoidance algorithm, the method is referred to as *complete error avoidance*.

In the operation phase, the transmitter avoids transmitting packets in high-error regions. When the packet queue is not empty, the transmitter calculates its current location using its current clock time. A data transmission is permissible only when its expected transmission duration does not overlap with any high error regions; otherwise, the transmission will be delayed until the beginning of an earliest low error region that is sufficient for the transmission to complete. If error avoidance is combined with ARQ, either the partial error avoidance or complete error avoidance procedure will be incurred to handle retransmissions.

3.2.2 Constant Packet Rate Models

In wireless data communication, throughput and energy efficiency heavily depend on the transmitter's transmission timing pattern. Many sensor systems report sensor data at a constant rate, while there are other systems aiming at transmitting as much data as possible in a short time so that the wireless channel is nearly saturated. In this chapter, two types of most widely-used wireless data transmission schemes are considered: constant packet rate scheme and saturated scheme. Both schemes are studied for throughput and energy efficiency metrics.

In a constant packet rate transmission scheme, the transmitter transmits one packet in each *packet interval*. With ARQ, if the first transmission of a packet fails, the transmitter can retransmit it for up to N times until either it is correctly received or the retransmission limit is reached. For the data transmission system to be stable and the throughput model to be valid, the packet interval must be much longer than the average time needed to successfully complete a packet's transmission, including all necessary retransmissions. Otherwise, the transmitter can become persistently backlogged and achieve only the saturation throughput as derived in the next section.

The throughput is derived based on the following observation: with constant packet rate traffic, as long as all packets are eventually transmitted successfully with no more than N-1 retransmission attempts, the throughput will be equal to S/I, where I is the packet interval. On the other hand, if any packet can fail for N transmission attempts, the throughput will degrade proportionally to the probability of such events. Let the probability of any packet successfully completing transmission with no more than N transmission attempts be defined as the average successful delivery ratio (SDR), then the constant packet rate throughput must be

$$TP_{CPR}(S, \mu, N) = \frac{S}{I} SDR(\mu, N) \tag{3.9}$$

Note that SDR and, as a result, TP both depend on μ and N. As the packet error rate differs for different locations, the average successful delivery ratio, $SDR(\mu,N)$, must be determined by accounting for the successful delivery ratio for individual locations. Let $SDR_{\mathrm{L}}(¯,\mu,N)$ be defined as the local successful delivery ratio at location ¯. This is the probability for a packet being transmitted for the first time at location ¯ to eventually complete successfully after potentially a number of retransmissions. Assuming transmissions can be initiated at any random location with a uniform probability,

$$SDR(\mu, N) = \int_{0^\circ}^{360^\circ} SDR_L\left(¯, \mu, N\right) d¯ \tag{3.10}$$

According to Eq. (3.3), energy efficiency is calculated based on throughput and total transmission count. If a packet starts transmitting at location ¯, the probability for that packet to be successfully received after i times of transmissions is $P_{¯,i}$, the local mean transmission count for location ¯ will be:

$$C_{M,L}(¯, \mu, N) = \sum_{i=1}^{N} P_{¯,i}\, i \qquad (3.11)$$

Assuming transmissions can be initiated at any random location with a uniform probability, the mean transmission count for the rotation cycle is:

$$C_M(\mu, N) = \int_{0^\circ}^{360^\circ} C_{M,L}\left(¯, \mu, N\right) d¯ \qquad (3.12)$$

Therefore, the energy efficiency can be expressed as:

$$Eff_{CPR}(S, \mu, N) = \frac{TP_A t}{C_T} = \frac{TP_{CPR}(S, \mu, N)t}{n_P C_M(\mu, N)} \qquad (3.13)$$

where n_p denotes the total transmitted packets number.

With ARQ, the first transmission occurs at location ¯; the probability for it to fail is the packet error rate for the location and let it be p_1. If the first transmission fails, the retransmission must take place at location (¯ + d) mod 360°, where mod is the modulo operation which finds the remainder of division of one number by another, and d is the transmitter's traveled distance during D_{cycle}, the time it transmits one packet and waits for its ACK to return or time out; ie., $D_{\text{cycle}} = S/r + t_{\text{ack}}$. Let the probability to fail for this retransmission be p_2. Hence, subsequent retransmissions, if needed, will take place at (¯ + 2d) mod 360°, (¯ + 3d) mod 360°, etc., and the resulting local successful delivery ratio for a transmission starting at ¯ is

$$SDR_L(¯) = (1 - p_1) + p_1(1 - p_2) + p_1 p_2 (1 - p_3) + (\cdots) + p_1 p_2 \cdots p_{N-1}(1 - p_N) \qquad (3.14)$$

where $p_i = PER_L((¯ + (i\text{-}1)d) \bmod 360^\circ, \mu)$, i=1~N. It is thus apparent that for ARQ with any given N,

$$SDR_L(\text{¯}, \mu, N) = 1 - \prod_{i=1}^{N} PER_L((\text{¯} + (i-1)d) \bmod 360°, \mu) \tag{3.15}$$

Moreover, if successfully transmitting one packet takes only one transmission, that probability will be:

$$P_{\text{¯},1} = 1 - PER_L(\text{¯}, \mu) \tag{3.16}$$

If it takes more than one transmission for that packet to be successfully received, that means all the previous transmissions fails and only the last transmission succeeds. Therefore, the following general form can be used to describe $P_{\text{¯},i}$:

$$P_{\text{¯},i} = \begin{cases} 1 - PER_L(\text{¯}, \mu), & (i = 1) \\ (1 - PER_L((\text{¯} + (i-1)d)) \bmod 360^{\circ}, \mu) \cdot \prod_{k=1}^{i-1} PER_L((\text{¯} + (k-1)d) \bmod 360^{\circ}, \mu), & (1 < i \le N) \end{cases} \tag{3.17}$$

The complete error avoidance scheme uses the error avoidance algorithm to allow each packet's all transmission attempts to take place only in a low-error region that has sufficient time for the transmission to complete in it. For a packet ready for its i_{th} transmission attempt, it must be delayed for u_{i} degrees to start transmission in a low-error region. Table 3.1 illustrates the deferral of each packet's ready location. For $1 \leqslant j \leqslant i$, u_{i} can be found as:

$$u_i = \arg\min_{v \ge 0} \left\{ \begin{array}{l} \left((\text{¯} + \sum_{j=1}^{i-1} u_j + (i-1)d + v) \bmod 360^{\circ} \in L_R \right) \cap \\ \left(\forall w \in [0, Q], (\text{¯} + \sum_{j=1}^{i-1} u_j + (i-1)d + v + w) \bmod 360^{\circ} \in L_R \right) \end{array} \right\} \tag{3.18}$$

where Q is the transmitter's traveled distance in degree during one packet transmission time and $L_R = \{\theta \mid PER_L(\theta) \le Th\}$. The intuitive explanation for (3.18) is that a packet should be transmitted when it enters the nearest low error region and the remaining width of the region is sufficient to fit in the packet.

Table 3.1: Derivation of packet ready location for complete error avoidance

Index	*Packet Ready Location*	*Transmission Location*
1	¯	$(\text{¯} + u_1) \bmod 360^\circ$
2	$(\text{¯} + u_1 + d) \bmod 360^\circ$	$(\text{¯} + u_1 + d + u_2) \bmod 360^\circ$
…	...	...
i	$(\text{¯} + \sum_{j=1}^{i-1} u_j + (i-1)d) \bmod 360^\circ$	$(\text{¯} + \sum_{j=1}^{i} u_j + (i-1)d) \bmod 360^\circ$

The local SDR for the complete error avoidance scheme can thus be expressed as

$$SDR_L(\text{¯}, \mu, N) = 1 - \prod_{i=1}^{N} PER_L((\text{¯} + \sum_{j=1}^{i} u_j + (i-1)d) \bmod 360^\circ, \mu) \tag{3.19}$$

The probability that takes i transmissions for a packet to be successfully received is:

$$P_{\text{¯},i} = \begin{cases} 1 - PER_L((\text{¯} + u_1) \bmod 360^\circ, \mu), & (i = 1) \\ (1 - PER_L((\text{¯} + \sum_{j=1}^{i} u_j + (i-1)d) \bmod 360^\circ, \mu) \cdot \prod_{k=1}^{i-1} PER_L((\text{¯} + \sum_{j=1}^{k} u_j + (k-1)d) \bmod 360^\circ, \mu), & (1 < i \le N) \end{cases} \tag{3.20}$$

Note that partial error avoidance is a special case of complete error avoidance. In partial error avoidance, only each packet's first transmission attempt must take place in a low-error region that has sufficient time for the transmission to complete in it while all retransmissions will take place immediately. Therefore, replacing u_i with u=u_1 in Eq. (3.18-3.20) leads to the SDR_L and $P_{\text{¯},i}$ expressions for partial error avoidance.

3.2.3 Saturated Throughput Models

In a saturated throughput transmission scheme when packets are generated at an interval shorter than the average successful completion time of a packet transmission (average packet delay), the wireless link becomes saturated such that the transmitter queue always has packets to send. If the packet interval is persistently shorter than the average packet delay, a portion of packets will even be dropped due to a full queue. In this section, an analytical model is derived to calculate saturated throughput and energy efficiency with the three transmission schemes.

The wireless sensor radio (MICAz) used in this dissertation does not provide back-to-back packet transmission; A ΔT time is needed for the radio to initiate a new packet transmission. ΔT can be measured by experiments. Figure 3.6 shows the timing relationship between D_{cycle}, t_{ack} and ΔT . In this section, all the calculations are based on IEEE 802.15.4 protocols.

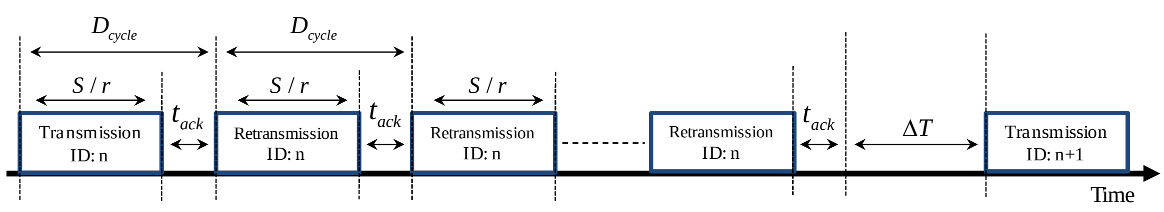

Figure 3.6: Packet timing schematic

When transmitting in saturation, the average time for the receiver to receive a packet successfully is the time needed to complete the average number of transmission attempts until it is received successfully. Generally, if a packet starts transmitting at location ¯, the probability for that packet to be successfully received after *i* times of transmissions is $P_{\theta,i}$, the time delay for i times of transmissions is $T_{\theta,i}$, then the average delay for location θ will be:

$$Delay(\theta, \mu, N) = \sum_{i=1}^{N} P_{\theta,i} T_{\theta,i} \tag{3.21}$$

Here N is the maximum transmission count limit. The overall average delay can then be calculated in the following way:

$$Delay(\mu, N)_{Overall} = \int_{0^\circ}^{360^\circ} Delay(\theta, \mu, N) d\theta = \int_{0^\circ}^{360^\circ} (\sum_{i=1}^{N} P_{\theta,i} T_{\theta,i}) d\theta \tag{3.22}$$

The saturated throughput is:

$$TP_{Saturated}(S, \mu, N) = \frac{S}{\int_{0^\circ}^{360^\circ} (\sum_{i=1}^{N} P_{\theta,i} T_{\theta,i}) d\theta} \tag{3.23}$$

Similar to Eq. (3.23), the saturated energy efficiency can be calculated as:

$$Eff_{Saturated}(S, \mu, N) = \frac{TP_A t}{C_T} = \frac{TP_{Saturated}(S, \mu, N) t}{n_P C_M(\mu, N)} \tag{3.24}$$

The derivation for $P_{\theta,i}$ is the same as Eq. (3.17) and (3.20). The following of this section derives the forms of $T_{\theta,i}$ for each scheme. In ARQ for i times of transmissions, the time delay of successfully transmitting one packet beginning at a particular location θ is:

$$T_{\theta,i} = i(t_{ack} + S / r) + \Delta T \tag{3.25}$$

If using complete error avoidance, every transmission will first check whether it is in a low-error region; if not, it will wait until it is in the nearest low-error region and then transmit immediately. Hence besides the packet transmission time and ACK timeout, additional time for finding the nearest low-error region is needed for each transmission attempt. Using the same notation shown in Eq. (3.18), the time delay of successfully transmitting one packet for a particular location ¯ is:

$$T_{-,i} = \sum_{i=1}^{N} \frac{u_i}{\mu} + i(t_{ack} + S / r) + \Delta T \qquad (3.26)$$

Similarly, partial error avoidance is a special case of complete error avoidance, hence Eq. (3.26) can be modified by having $u=u_1$, in replace of u_i.

3.3 Experimental Study

The constant packet rate and saturated models are validated with two experimental rotating structure setups running error avoidance and ARQ algorithms on a MICAz radio sensor. The following introduces the radio configuration, experimental setups, procedures and model validation results.

3.3.1 Radio Configuration

It is assumed that the rotating system has no specialized mechanism to determine the absolute position of the sensor at any time instant. The introduced approach addresses this challenge by establishing a relative coordinate system based on periodic probing with timestamps. The rotation speed is assumed to be readily available in most modern rotational systems.

To achieve precise control of packet transmission time for an error avoidance scheme, the radios adopt the IEEE 802.15.4 non-beacon transmission mode with the optional acknowledgement **enabled**, clear channel assessment (CCA) **disabled**, and random backoff **disabled**. When error avoidance is employed, the transmitter and receiver radios must be synchronized to determine the transmission time. In this dissertation, it assumes the availability of a third radio, referred to as a base station, to synchronize the transmitter and receiver using the reference broadcast synchronization (RBS) algorithm [Elson2002]. Base station broadcasts a single pulse to transmitter and receiver periodically, allowing them to estimate their relative phase offsets, that is:

1. Base station broadcasts a refernece packet to transmitter and receiver.
2. Each one records the time that the reference packet is received, according to its local clock.
3. Receiver sends its observation to transmitter.
4. Transmitter compares the phase offset between its own observation and receiver's observation. It modifies its own system time based on that offset.
5. Transmitter's systme time is synchronized with receiver's system time.

As base station is close to transmitter and receiver, the signal travel time is neglected. By following the above steps and with base station periodically sending out synchronization pulses, system time between transmitter and receiver can be synchronized. The choice of radio synchronization method, though, is independent of the error avoidance algorithm. The synchronization process can be initiated periodically to account for the sensor nodes' clock drift. For a wireless sensor monitoring system for rotating structures, the dissertation considers a sensor node attached to the rotating surface, referred to as *the transmitter*, for making measurements and transmitting the data to a receiver node, referred to as *the receiver*, which is placed on a stationary surface. The *base station* communicates with both nodes to control their operations and facilitate their time synchronization.

3.3.2 Experimental Setups

A. Rotating Apparatus Experiment Setup

The algorithms implementation and experimental measurements are firstly done on a rotating apparatus. This apparatus uses a permanent magnet DC motor to drive an aluminum rotary plate. The motor model is DAYTON 1F800, and its nameplate speed is 1750 rpm. A controller has been equipped with the motor to tune rotation speed and switch rotation direction. The motor and the rotary plate are supported by brackets on a bench. Figure 3.7 shows the rotation system setup for sensor radios. In this setup, transmitter is mounted somewhere on the rotary plate, it will rotate during the measuring process. Receiver is placed stationary in another place, which is 0.7 meters away from this apparatus.

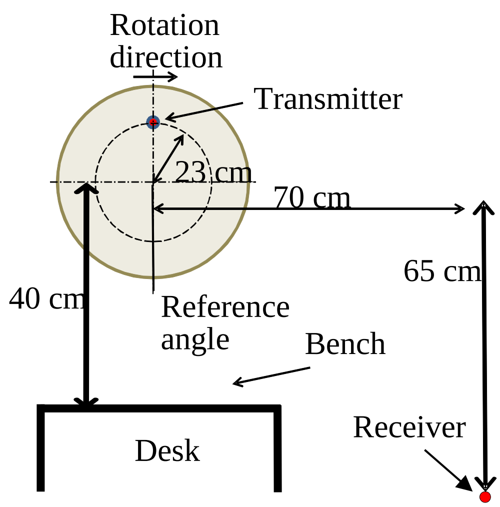

Figure 3.7: Rotating apparatus setup and placement schematic

B. CNC Lathe Experiment Setup

The CNC lathe used in the experiments is a Talent 6/45 CNC lathe by Hardinge Inc. The dimension of the CNC lathe and the placement of the sensor radios are the same in a previous study [Tang2009b]. The lathe is equipped with a 60~6000 rpm speed-controllable spindle in a metallic enclosure space and speed display on its front panel. The transmitter is safely mounted on the lathe spindle by duct tape, and the receiver with MIB510 board (Moog Crossbow, Milpitas, CA, USA) is deployed on the inner-wall of the lathe. During the experiments, the machine window is closed, with no people or other objects allowed in the room. Fig. 3.8 shows the diagram of the MICAz radio sensor mounted on the rotating spindle and example error regions along the spindle periphery.

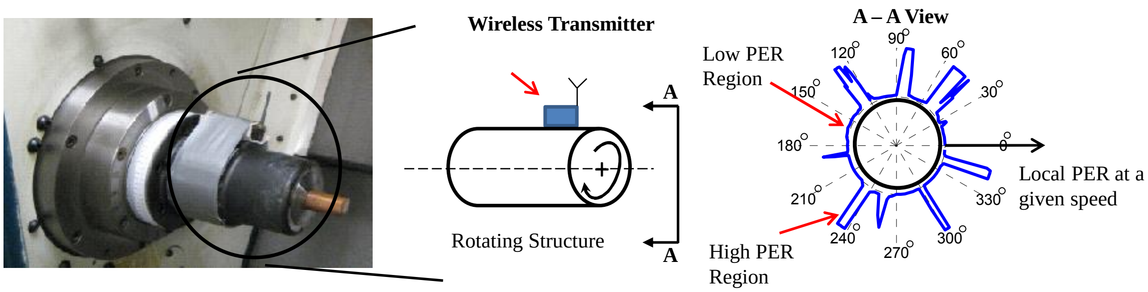

Figure 3.8: Diagram of rotating spindle and example error regions

The functional structure of the measuring system for both experimental setups including transmitter, receiver and base station is illustrated in Figure 3.9. A personal computer records data from the receiver through UART (universal synchronous receiver/transmitter), which is a piece of computer hardware that translates data between parallel and serial forms.

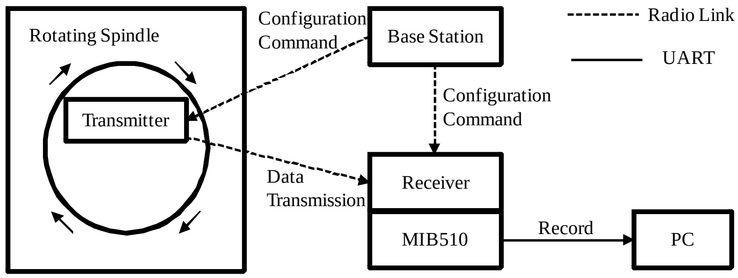

Figure 3.9: Functional structure of sensor measurement system

3.3.3 Experimental Procedure

Before each experiment, speed is measured and saved in the base station's configuration packet. To begin an experiment, the base station sends the configuration packet to both the transmitter and the receiver. Upon receiving configuration packet, the transmitter enters the training phase and starts sending probe packets, the receiver then calculates the PER profile based on the received packets. When 5000 probe packets have been sent out, the transmitter stops transmitting and the receiver sends the high error regions list to the transmitter; upon receiving this packet, the transmitter enters the operation phase, and starts sending data with the chosen ARQ and error avoidance scheme. The measurement is finished when 5000 data packets have been sent. Throughout the whole measurement process, the base station periodically (every 5 seconds) sends synchronization packets to both transmitter and receiver. The synchronization packet has a probability to collide with data packets. Enabling the CCA function at the base station so that it will wait to send out a synchronization packet until there is no other packets transmitting in the channel can effectively decrease packet collision probability. As the synchronization packet will not affect both the transmitter and the receiver's timestamps, enabling the CCA function at the base station is valid in the experiments to utilize error avoidance algorithm.

Total packet transmission count is measured throughout the experiment by recording packet sequence number. Energy efficiency is obtained by dividing the total received information bits with measured total transmission count. As the time interval *t* for the whole measurement is the same for the schemes, energy efficiency is normalized against *t*. In this dissertation, constant packet rate (CPR) model experiments are conducted in the CNC lathe setup. Table 3.2 lists the configuration commands that the base station sends to the transmitter and the receiver.

Table 3.2: Configurations sent by base station (CPR experiment, CNC lathe)

Configuration	*Setup*
Frequency	2478.5 ~ 2481.5 MHz
Transmit power	-15 dBm
Packet interval	0.05 second
Packet size	30 bytes
Maximum transmission count limit	5
Number of packets (Training or operation phase)	5000
PER threshold	0.2
Rotation speed (RPM)	242, 544, 845, 1027 1324, 1510, 1812

The saturated model experiments are conducted on the rotating apparatus setup. The packet size specified is 50 bytes, the PER threshold is set as 0.2 and the rotation speed is chosen from 97.7, 201.4 and 281.4 rpm. Other experimental procedures are identical to the CNC lathe experiments.

3.3.4 Model Validation

CPR throughput is measured in the CNC lathe experiment, where packet size is fixed. At this experimental setup, throughput decreased slightly as rotation speed increased. The reason is that higher rotation speed increases the probability of a packet hitting an error region, which causes a throughput drop. Using error avoidance does not provide significant throughput increase compared to using ARQ at this setup, indicating ARQ is enough to recover most of the packet errors. By comparing the measured and calculated results in Figure 3.10, it is found that throughput difference between measurement and calculation is less than 5 bps, which is only 0.1% of the measured results, demonstrating the effectiveness of analytical throughput model.

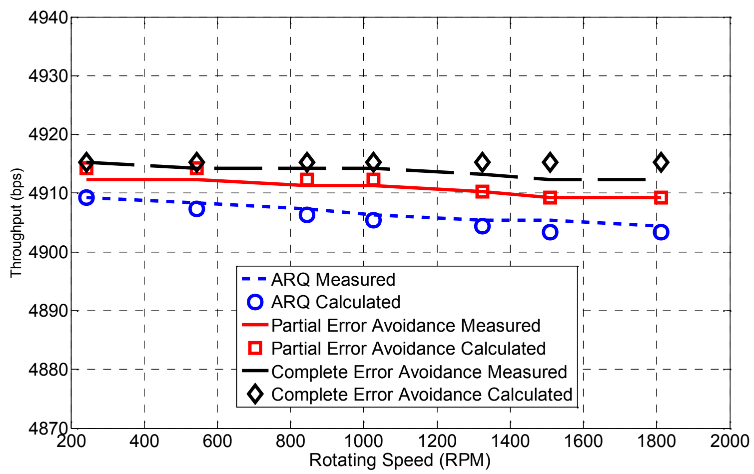

Figure 3.10: CPR throughput model validation

It can be seen in Figure 3.11 that using error avoidance provided nearly 20% energy efficiency improvement than using ARQ. Using error avoidance significantly reduces total transmissions count, therefore saving transmission energy. It is also worth noting that measured energy efficiency results match the model calculation well, with less than 0.87% difference, showing the correctness of the energy efficiency models.

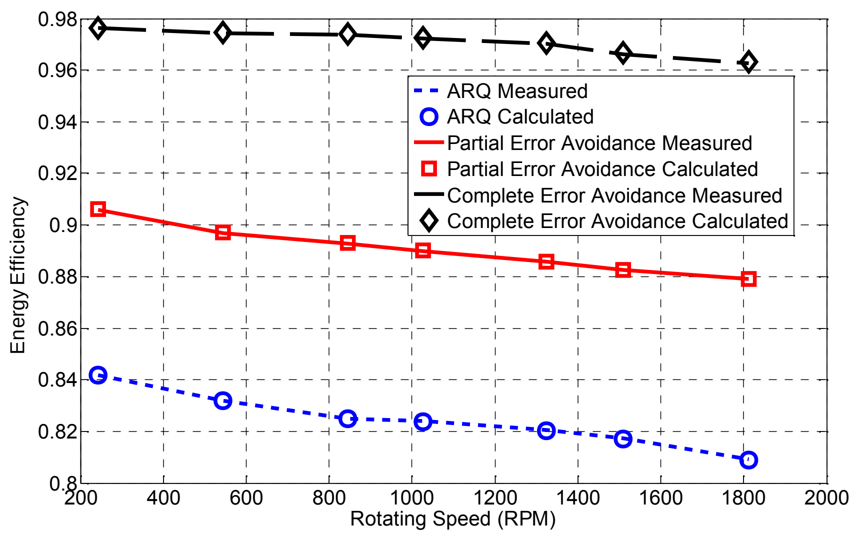

Figure 3.11: CPR energy efficiency model validation, normalized against t

During the saturated throughput experiments, the minimal time interval $\Delta T = 2.398\ ms$ is measured. Figure 3.12 shows the comparison plots between calculated and measured throughputs at different rotation speeds for each scheme. From the plots, calculation results match the measured results well, the largest difference is only within 0.5%, demonstrating the accuracy of the saturation model. For the rotation speed range (100~300 rpm) and PER threshold (0.2) under this experimental setup, it is observed that: 1) saturated throughput decreases as rotation speed increases (from 100 rpm to 300 rpm), and 2) using error avoidance hurts saturated throughput performance. This indicates that PER threshold and rotation speed will affect saturated throughput performance under different schemes. Their effects will be discussed comprehensively in later sections.

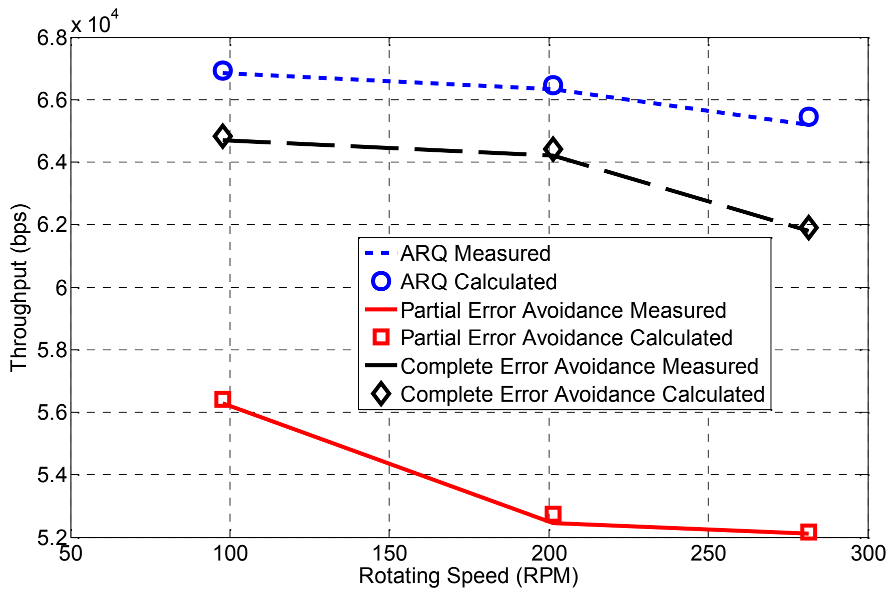

Figure 3.12: Saturated throughput model validation

Energy efficiency for saturated models are measured and compared in Figure 3.13. Although ARQ has better throughput performance than error avoidance in this setup, energy efficiency of using complete error avoidance is still the best, indicating much more retransmissions were utilized to recover lost packets when using ARQ. It is also notable that using complete error avoidance provides nearly 31.6% energy efficiency improvement than using ARQ in this case.

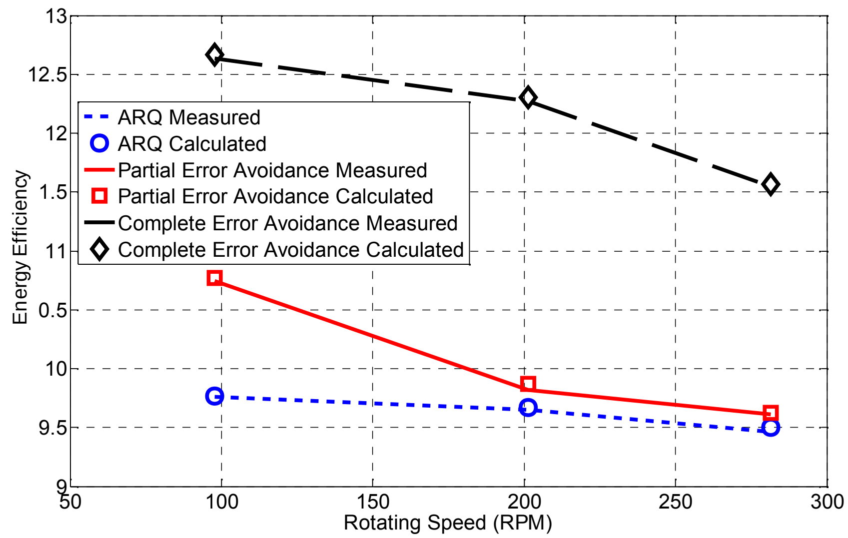

Figure 3.13: Saturated energy efficiency model validation, normalized against *t*

3.4 Reliable Transmission Protocol Selection

Although the deterministic characteristics of the rotating structure provide opportunities for error avoidance algorithms to significantly improve data transmission performance, the extra waiting time on finding the next good region may degrade performance under particular scenarios. In some channel environments, the packet interval in constant packet rate transmission scheme is much longer than the average time needed to successfully complete a packet's transmission, ARQ is able to improve throughput by means of more retransmissions. Using error avoidance may not be able to provide significant throughput improvements.

In this work, the challenge is how to make easy transmission protocol selection given a specific wireless channel, which can be addressed by our framework, using surface roughness parameters. We show theorems that characterize channel using surface roughness parameters. The simple relationship between those parameters reveals under what kind of environment error avoidance will have better throughput improvements. While rotating structure monitoring is presented herein, the proposed theorems can be applied to other wireless communication systems with location-dependent channel properties. Such scenarios and applications include: industrial monitoring applications where sensor nodes are deployed at different point-of-interest in a factory setting with harsh wireless channel conditions, and environmental monitoring applications where sensor nodes are deployed in forests, riverbank, etc. with specific wireless channel conditions.

3.4.1 Theorems

Theorem: *Error avoidance will have better throughput improvements than ARQ in a rotating structure if* $R_{s,max} > Q$*, where* $R_{s,max}$ *is the maximal value of* $R_s(i), 0<i<m+1$*,* Q *is the transmitter's traveled distance in degree during one packet transmission time*.

Proof:

Here Q is the transmitter's traveled distance in one packet transmission, given S is the packet size, r is the link rate and μ is the rotation speed, it is straightforward to obtain:

$$Q = \frac{6\mu S}{r} \tag{3.27}$$

Denote $R_{s,max}$ to be the maximal value of $R_s(i)$, $0<i<m+1$

$$R_{s,\max} = \max\{R_s(i), 0 < i < m+1\} \tag{3.28}$$

If we have:

$$R_{s,\max} > Q \tag{3.29}$$

then it indicates that for $i \in [1, m]$ and $Q>0$, we have

$$\exists i \in [1, m] : R_s(i) > Q \tag{3.30}$$

The location set of intersections of downwards curves and PER threshold line is denoted as **D**. Apparently, $\mathbf{D} \in L_R$. Therefore:

$$\exists(\text{¯} \bmod 360^\circ) \in \mathbf{D} : \forall w \in [0, Q], (\text{¯} + w) \bmod 360^\circ \in L_R \tag{3.31}$$

Moreover, according to Eq. (3.18), a packet should be transmitted when it enters the nearest low error region and the remaining width of the region is sufficient to fit in the packet, by combining Eq. (3.18) and (3.31),

$\exists u$ defined by Eq. (3.18):

$$PER_L((\text{¯} + u + (i-1)d) \bmod 360^\circ, \mu) \le Th < PER_L((\text{¯} + (i-1)d) \bmod 360^\circ, \mu) \tag{3.32}$$

Consequently, for N times of transmission attempts:

$$\prod_{i=1}^{N} PER_L((\text{¯} + \sum_{j=1}^{i} u_j + (i-1)d) \bmod 360^\circ, \mu) < \prod_{i=1}^{N} PER_L((\text{¯} + (i-1)d) \bmod 360^\circ, \mu) \tag{3.33}$$

Slightly rewriting both sides of Eq. (3.33),

$$1 - \prod_{i=1}^{N} PER_L((\text{¯} + \sum_{j=1}^{i} u_j + (i-1)d) \bmod 360^\circ, \mu) > 1 - \prod_{i=1}^{N} PER_L((\text{¯} + (i-1)d) \bmod 360^\circ, \mu) \tag{3.34}$$

It should be noted that the left side of Eq. (3.34) is the average successful delivery ratio (*SDR*) when using error avoidance, while left side of is the *SDR* when using ARQ. Finally, by combining Eq. (3.1), (3.10) and (3.34), it can be concluded that error avoidance will have better throughput performance than ARQ if $R_{s,max} > Q$. Note the above proof process is for constant packet rate throughput models. Similar proof process can also be derived for saturated throughput models. □

Corollary 1: *Denote k to be the number of i such that $R_S(i) > Q$. When error avoidance is used, larger k leads to better throughput improvement performance.*

Proof:

Larger k indicates there is more number of low error regions that are able to fit complete packet transmissions. According to Eq. (3.32), there will be more number of u such that:

$$PER_L((\text{¯} + u + (i-1)d) \bmod 360^\circ, \mu) \leq Th \,. \tag{3.35}$$

Therefore, better throughput improvement will be obtained by combining Eq. (3.1), (3.10) and (3.35). □

Corollary 2: *Given two PER profiles $PER_{L,1}$ and $PER_{L,2}$. If $R_{a,1} < R_{a,2}$, the resultant throughput performance will be: $TP_1 > TP_2$ when error avoidance is used.*

Proof: The proof process is straightforward. According to the definition, If $R_{a,1} < R_{a,2}$, then:

$$\int_{0^\circ}^{360^\circ} PER_{L,1}(\text{¯})\, d\text{¯} < \int_{0^\circ}^{360^\circ} PER_{L,2}(\text{¯})\, d\text{¯} \tag{3.36}$$

When error avoidance is used, combining Eq. (3.1), (3.10), (3.35), and (3.36):

$$SDR_1(\mu, N) = \int_{0^\circ}^{360^\circ} SDR_{L,1}(\text{¯}, \mu, N)\, d\text{¯} > SDR_2(\mu, N) = \int_{0^\circ}^{360^\circ} SDR_{L,2}(\text{¯}, \mu, N)\, d\text{¯} \tag{3.37}$$

Finally, combining Eq. (3.1) and (3.37),

$$TP_1 > TP_2 \tag{3.38}$$

The proof is complete. □

3.4.2 Validation

From the previous section, the throughput and energy efficiency models for different schemes were validated. In this section, validated models and surface roughness based channel parameters are utilized to analyze the effects of different environments on error avoidance and ARQ performance improvements.

Figure 3.14 shows an example PER profile where the maximal value of R_s is 562% larger than Q and the number of i such that $R_s(i) > Q$ equals to 85% of the total number of locations along the peripheral. It is obtained from [Tang2010], where experiments were done using the same rotating apparatus setup in an anechoic chamber that is designed to stop reflections of electromagnetic waves.

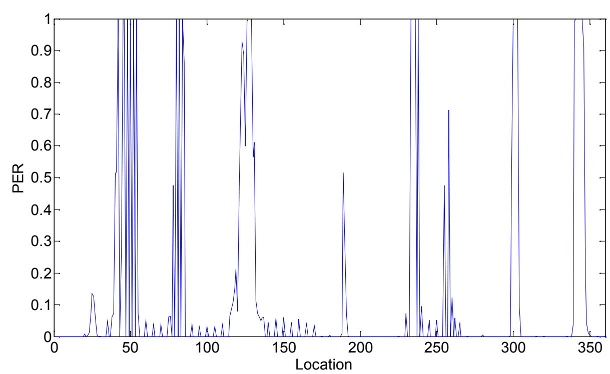

Figure 3.14: Example PER profile [Tang2010]

Based on this PER profile, the throughput performance under different schemes and rotation speeds by calculation from the model is shown in Figure 3.15. It can be seen that using complete error avoidance has almost 100% more throughput than using ARQ when rotation speed is larger than 1400 rpm. Traditional ARQ has poor performance in such environments since the possibility of a packet hitting a high-error region is high, especially when rotation speed grows larger, while error avoidance can have all transmissions of packets happen only in low-error regions, hence the possibility of packet loss greatly decreases.

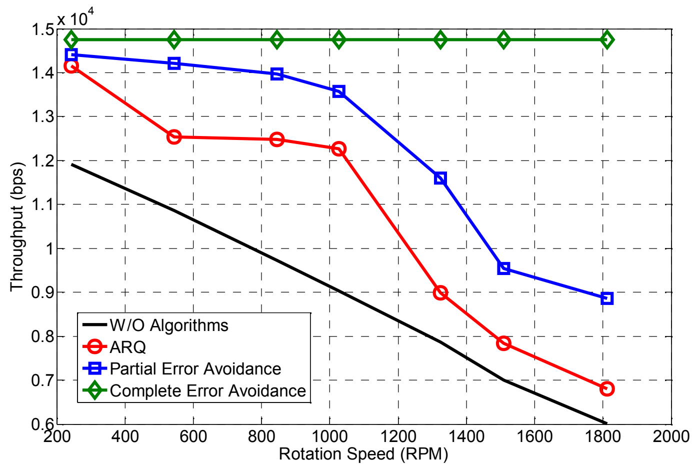

Figure 3.15: Calculated CPR throughput

Furthermore, 100 random local PER profiles with different R_s and R_a values are created to analyze the statistical results when both models are applied. Results successfully prove that as long as the calculated maximum value of R_s is larger than Q at a certain speed, throughput performance using error avoidance is always better than using ARQ. Moreover, Fig. 9 shows R_a and k value plots with throughput being the x-axis. It proves that 1) when error avoidance is used, larger k values lead to better throughput improvements, and 2) when error avoidance is used, smaller R_a values lead to better throughputs. It is also observed from simulation results that if the throughput is low, total packet transmission count is high and thus the energy efficiency becomes much lower.

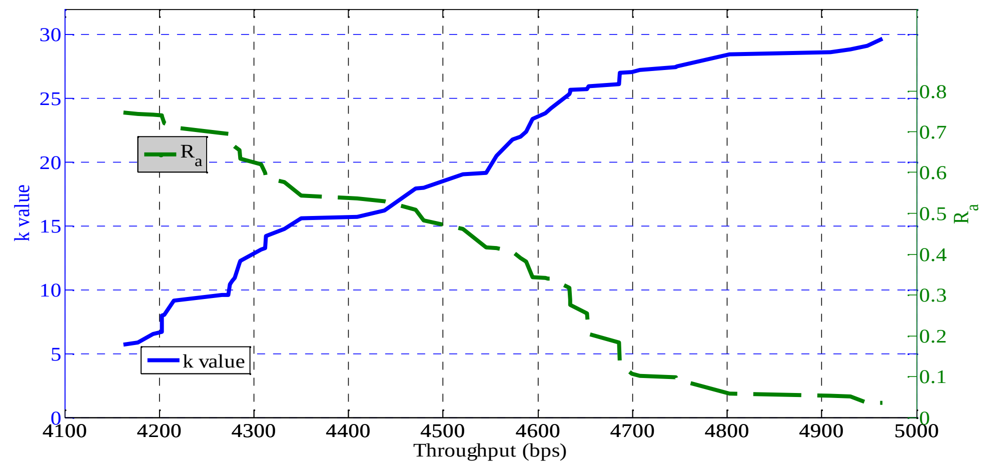

Figure 3.16: R_a (right y-axis) and k value (left y-axis) Plots

Based on each PER profile, the relative energy efficiency improvement using error avoidance compared to ARQ is calculated using the following equation

$$Eff_{relative} = \frac{Eff_{EV} - Eff_{ARQ}}{Eff_{ARQ}} \times 100\% \tag{3.39}$$

where Eff_{EV} and Eff_{ARQ} are the energy efficiency when error avoidance and ARQ are utilized, respectively.

A 95% confidence interval for relative energy efficiency improvement is shown in Table 3.3 for different schemes at different rotation speed (200, 1000, 2000 rpm). PER threshold is fixed to be 0.2 for all the calculations. Demonstrated by the results, complete error avoidance always has better energy efficiency performance than partial error avoidance when $R_s > Q$. Moreover, as higher rotation speed makes a packet find a low error region more quickly, better energy efficiency performance improvement is expected at higher rotating speed.

Table 3.3: Confidence interval for relative energy efficiency improvement

Model	*Scheme*	*Speed (rpm)*	*Confidence Interval (95%)*	
			Lower Bound	*Upper Bound*
Constant Packet Rate	Partial Error Avoidance	200	5.832%	9.367%
		1000	6.752%	10.443%
		2000	6.775%	11.243%
	Complete Error Avoidance	200	10.952%	18.673%
		1000	12.875%	22.876%
		2000	12.998%	24.043%
Saturated	Partial Error Avoidance	200	15.432%	23.789%
		1000	18.221%	28.543%
		2000	19.009%	29.863%
	Complete Error Avoidance	200	29.568%	35.789%
		1000	31.182%	38.665%
		2000	31.986%	39.784%

3.5 Summary

This chapter studies a simple framework for modeling wireless data transmission with spatially deterministic properties. Throughput and energy efficiency models are calculated based on channel parameters. The spatial property of wireless channels is characterized using surface roughness parameters such as the amplitude and spacing parameters, which better represent the profile of channel errors. ARQ and error avoidance transmission protocols are introduced and experimentally implemented in a case study. Furthermore, surface roughness parameter-enabled theorems are presented to illustrate the process of transmission protocol selection under different environments.

Experimental results show that:

1) Throughput and energy efficiency are location-dependent for specific environments, and they depend on the transmission protocol adopted;

2) The amplitude and spacing parameters can be extracted from a given wireless channel with spatially deterministic properties, and then be used effectively for protocol selection, demonstrating a good match between the model predictions and the experimental measurements.

CHAPTER FOUR

SUSTAINABLE COMMUNICATION MODELING AND PROTOCOL DESIGN WITH LOW AND DYNAMIC POWER SOURCES

The utilization of renewable energy sources for wireless sensor systems is useful for sustainable monitoring systems [Niyato2007], but very limited today due to a range of reasons. Solutions with microbial fuel cells based power source are newly emerging techniques which can be enclosed and embedded into terrains, wetlands and structures. How to utilize MFC power output by power management system (PMS) needs to be firstly studied. Despite the wealth of wireless sensor network literature, MFCs present very different and challenging charging properties that are beyond the typical assumptions in the state-of-the-art. How to design communication protocols in structured ways that can guarantee sustainable communication success for such MFC based wireless sensor networks is the major problem that needs to be addressed.

In this chapter, energy neutral data transmission scheduling and protocol design in MFC powered multi-hop WSN are studied and verified in simulations.

4.1 Energy Neutral Transmission Scheduling

The advantage of using MFCs as power sources is that they can be seamlessly deployed in natural environments and civil infrastructures as they do not need to be exposed as the solar panels or wind turbines, making them more applicable for pervasive deployment and less susceptible to damages. How to model and improve MFC output performance using power management system can be found in Appendix A. The extremely limited and dynamic nature of MFCs presents a big challenge in designing communication protocols for MFC-powered WSNs.

In this section, a sensor network that is composed of many MFC-powered nodes deployed in a multi-hop fashion is considered. A sensor/mesh node can either have data (e.g. sensor data) generated locally or receive data packets from other nodes when it acts as a relay node. Energy harvesting rate of each node is modeled after typical MFC-PMS designs as presented in Appendix A, which changes randomly within a certain range with uniform probability.

Each node joins the network randomly and has a fixed number of data packets to send (noted as M). Data packets (size noted as D) contain environmental information such as temperature, humidity and moisture, etc. They only occupy several bits of a packet. There is one sink with unlimited power supply to process and store sensor data from all the nodes. Most packet forwarding will be between nodes as peers, rather than to the sink. A node at a certain hop will relay data packet to its peer node at the next hop in the direction to the sink. In [Kansal2007], a sensor network with Energy Neutral Operation (ENO) indicates that long-term energy consumption by wireless sensors in the network is less than total energy harvested from the environment. In this section, the node-to-node duty cycle communication strategy and its mathematical models are presented.

4.1.1 Duty Cycle Design

The first step towards an energy neutral protocol is design of a duty cycling strategy as shown in Figure 4.1.

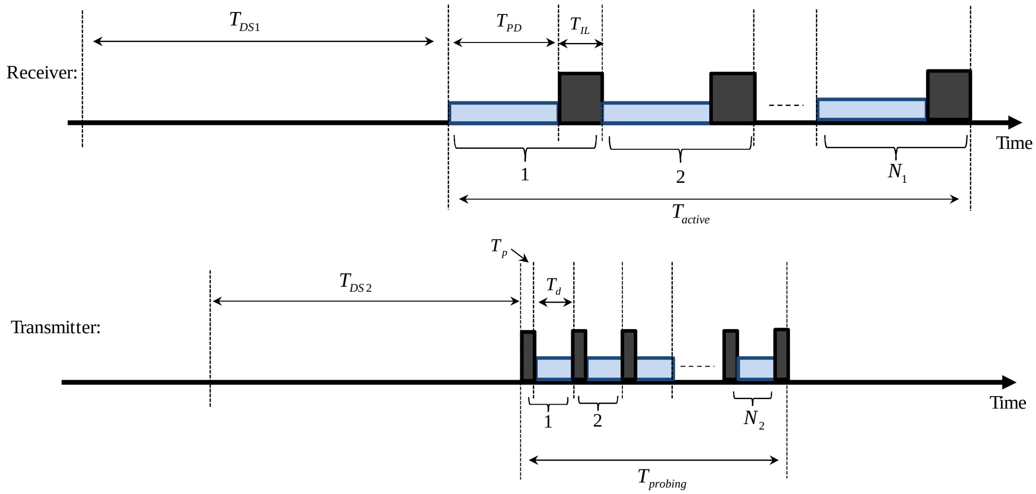

Figure 4.1: Energy neutral duty cycle design. Note the receiver and transmitter diagram are of very different time scales. For general reference consider $T_{probing} = T_{PD} + T_p$.

Each node has four operating modes modeled after the CC2420 radio as summarized in Table 2.1: probe transmitting (PB), idle listening (IL), power down (PD), and deep sleep (DS). A sensor node can be in either the transmitter or receiver mode at a given time operating with the respective duty cycling scheme. The extremely low power consumption in DS period allows fast charging of a node's battery until it reaches the target operating voltage (e.g., 3.3 volt), where the node enters the PD mode to resume its processor operation. After T_{PD} time the node enters the PB or IL mode: for a transmitter (in its PB mode), it transmits a probe of duration T_p if it has data to transmit; for a receiver (in its IL mode), it receives a probe if transmitter's PB mode overlaps with its IL mode. Once a receiver receives a probe successfully, the receiver replies with an acknowledgement (ACK) and the transmitter proceeds to transmit the data. The following analysis focuses on the probing process and leaves out the data transmission stage. A receiver node repeats the PD-IL periods until a probe is received or N_1 cycles have been reached; a transmitter node repeats the PD-PB periods until a probe is received by the receiver node and an acknowledgement is returned or N_2 cycles have been reached. Either node must return to DS when its voltage reaches a lower operating threshold (e.g., 2.5 volt, it may differ for transmitter and receiver) and remain in DS until the target operating voltage is reached.

The duty cycling scheme in Figure 4.1 reveals a number of important considerations shared by any duty cycle based energy neutral protocols. First, how "deep" a sleep mode to adopt is a tradeoff decision concerning recharging time, protocol synchronization, and the network's achievable throughput and latency. Second, the time a node has to spend in DS is a function of the recharging rate, battery capacity, and intended length of active period in each cycle. Therefore, the DS periods for any two nodes should be considered different in practice due to their different charging conditions, and such differences directly impact the probability of successful communication (probing) in their overlapping PB and IL periods. Prior studies often assume such probabilities to be acceptably high with a high enough node density or that nodes can be synchronized to wake up at agreed times. With MFCs' dynamic charging rates and the need to turn clocks off for sufficient power saving, the two assumptions would be impractical. Nevertheless, in the following it is demonstrated that even under such conditions, the network's stochastic performance can still be assessed and optimized based on constraints to guaranteed probing success, guaranteed energy neutrality, and additional application expectations.

The following five constraints represent the conditions for guaranteed probing success (Equation (4.1)-(4.3)) and energy neutral operation (Equation (4.4)-(4.5)).

$$T_{probing} = T_{PD} + T_p \tag{4.1}$$

$$T_d + 2T_p \le T_{IL} \tag{4.2}$$

$$T_{DS2} + T_p \le T_{active} - T_{PD} \tag{4.3}$$

$$T_{DS1}P_e \ge N_1(T_{PD}P_{PD} + T_{IL}P_{IL}) + N_1E_{s1} + E_{s2} + E_x \tag{4.4}$$

$$T_{DS2}P_e \ge (N_2 + 1)T_pP_t + N_2T_dP_{PD} + (N_2 + 1)E_{s1} + E_{s2} + E_x \tag{4.5}$$

All the timing parameters are as shown in Figure 4.1; P_e is the average recharging rate, P_{PD} and P_{IL} are the power consumption rates in PD and IL periods, P_t is the probing packet transmit power, E_{s1} and E_{s2} are power costs for switching from PD to PB/IL modes and switching from DS to PD/PB/IL modes, and E_x is the energy for transmitting and receiving a data packet. Equation (4.1) guarantees that probing packet must be received successfully if the transmitter's first probing packet arrives anywhere in the receiver's power down period (T_{PD}). Figure C.1 in Appendix C shows an example of the timing relationship. Equation (4.2) guarantees that probing packet must be received successfully if the transmitter's first probing packet arrives anywhere in the receiver's idle listening period (T_{IL}). $2T_p$ indicates that the probe packet is considered reception only if the entire packet is heard by the receiver. Equation (4.3) guarantees that a probing packet must be received successfully at the receiver's immediate next active period if the transmitter's first probing packet arrives anywhere in the receiver's DS period (T_{DS1}). With the five constraints, a nonlinear program can be formulated to optimize a range of potential objectives. As data packet size D is considered as a constant in this dissertation, finding the maximum throughput between two given nodes is equivalent to finding the minimum average successful probing time T_{ave}. The derivation of T_{ave} is presented in Appendix C. Therefore, for any given N_1 and N_2, the nonlinear program can identify values of T_d, T_{PD}, T_{IL}, T_{DS1}, and T_{DS2} such that maximal throughput TH_{opt} can be achieved as shown in Equation (4.6).

$$\text{Minimize: } T_{ave}$$

$$\text{s.t.: Equation (4.1-4.5)}$$

$$TH_{opt} = D/T_{ave} \tag{4.6}$$

It should be noted that due to the asynchronous nature of the protocol, the proposed separate periodic probing step is important for assuring a receiver node to be ready for receiving, which is similar to RTS-CTS. This 2-step approach is essential to enable feasible and sustainable network operation with the ultra-low and dynamic power source. As already reviewed in the background, a single step transfer mode would need all nodes to be synchronized, which is not applicable in such systems.

Table 4.1: Example calculated parameter results with P_e = 0.5 mW

Term	*Description*	*Unit*	*Value*
T_{DS1}	Receiver deep sleep (recharging) time	s	178.2
T_{PD}	Receiver power down time	s	0.9
T_{IL}	Receiver idle listening time	ms	19.6
T_{DS2}	Transmitter deep sleep (recharging) time	s	29.1
T_d	Transmitter power down time	ms	18.9

T_{ave}	Average successful probing time	s	77.4

Figure 4.2 is an example showing that node-to-node throughput can drop significantly if T_{DS2} deviates from the optimal value. In this example, average charging rate P_e was selected to be 0.5 mW. The rest of parameters involved in calculation from Equation (4.1-4.6) are listed in Table 4.2 (shown in later sections). The nonlinear program was solved in AMPL [AMPL] with SNOPT nonlinear optimization solver [SNOPT]. The calculated parameter results are listed in Table 4.1.

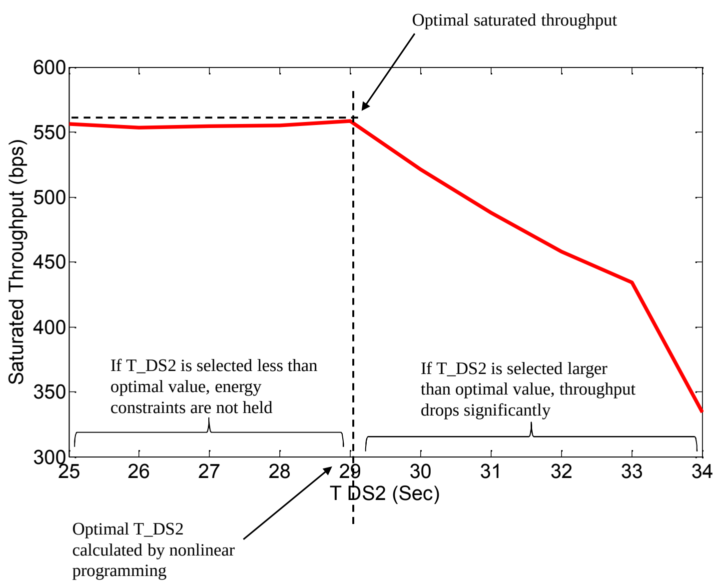

Figure 4.2: Maximal throughput occurs with the timing parameters solved by the nonlinear program

With P_e = 0.5 mW, the resultant T_{DS2} was solved to be 29.1 seconds and TH_{opt} to be 566.8 bps. As shown in Figure 4.2, if other values instead of 29.1 were chosen for T_{DS2} and the rest of parameters were kept unchanged, the recalculated saturated throughput would drop below 566.8 bps. This example demonstrates that the optimization formulation can serve as a useful framework for identifying the theoretical maximal performance as well as for finding optimal protocol parameters for an adaptive network protocol.

4.1.2 Duty Cycle Modeling

The transmitter or receiver duty cycle can be represented as a list of events (e) that occur sequentially in time. Each event is defined as *deepsleep*, *powerdown* or *active*, where the *active* event is defined to be either *probing* (for transmitter) or *idlelistening* (for receiver) such that:

$$active = probing \cup idlelistening \tag{4.7}$$

At any given time, the state of a node is defined by $S(t(q), e(q))$, where q indicates that the node is in the q_{th} duty cycle; the current event of the node is e and t is its event number. Define $n(q)$ as the total event number for the q_{th} duty cycle:

$$n(q) = \max(t(q)) \tag{4.8}$$

If D, P, and A denotes the set of t when e is *deepsleep*, *powerdown*, and *active* within a duty cycle, respectively, then $D = \{t_i \,|1 \le i \le k_D, t_i \in \mathrm{Z}\}$, $P = \{t_i \,|1 \le i \le k_P, t_i \in \mathrm{Z}\}$, and $A = \{t_i \,|1 \le i \le k_A, t_i \in \mathrm{Z}\}$, where k_D, k_P, and k_A is the total number of *deepsleep*, *powerdown*, and *active* events within a duty cycle. It should be noted that $Z = D \cup P \cup A$.

Definition (q-extention): *Given two positive integers n and q, for a set* $A = \{a_i|1 \le i \le k, a_i \in \mathrm{Z}\}$*, the q-extension of A is defined as* $A^q = \{a_i + j * n|1 \le i \le k, 0 \le j \le q - 1, a_i \in \mathrm{Z}\}$*, where n is the total number of elements in set Z.*

Consequently, sets D, P, and A at the q_{th} duty cycle can be represented as D^q, P^q, and A^q:

$D^q = \{t_i + j * n_i(q) \,|\, 1 \le i \le k_D,\ 0 \le j \le q\text{-}1,\ t_i \in \mathrm{Z}\}$;

$P^q = \{t_i + j * n_i(q) \,|\, 1 \le i \le k_P,\ 0 \le j \le q\text{-}1,\ t_i \in \mathrm{Z}\}$;

$A^q = \{t_i + j * n_i(q) \,|\, 1 \le i \le k_A,\ 0 \le j \le q\text{-}1,\ t_i \in \mathrm{Z}\}$, where $n_i(q) = k_D + k_P + k_A$.

Probing interval T_p is the smallest time interval in the protocol, which serves as the time duration of a single event. Therefore, in the transmission scheduling during a given duty cycle q, for a node in transmitter mode:

$$D_{TX} = \{i \,|\, 1 \le i \le k_D,\ k_D = \frac{T_{DS2}}{T_p}\} \tag{4.9}$$

$$P_{TX} = \{k_D + 1 + i + (1+\frac{T_d}{T_p})h \mid 1 \le i \le \frac{T_d}{T_p},\ 0 \le h \le (N_2\text{-}1),\ k_P = \frac{T_d}{T_p} N_2\} \tag{4.10}$$

$$A_{TX} = \{k_D + 1 + (1+\frac{T_d}{T_p})h \mid 0 \le h \le (N_2\text{-}1),\ k_A = N_2 + 1\} \tag{4.11}$$

For a node in receiver mode:

$$D_{RX} = \{i \,|\, 1 \le i \le k_D,\ k_D = \frac{T_{DS1}}{T_p}\} \tag{4.12}$$

$$P_{RX} = \{k_D + i + (\frac{T_{IL} + T_{PD}}{T_p})h \mid 1 \le i \le \frac{T_{PD}}{T_p},\ 0 \le h \le (N_1\text{-}1),\ k_P = \frac{T_{PD}}{T_p} N_1\} \tag{4.13}$$

$$A_{RX} = \{k_D + \frac{T_{PD}}{T_p} + i + (\frac{T_{IL} + T_{PD}}{T_p})h \mid 1 \le i \le \frac{T_{IL}}{T_p},\ 0 \le h \le (N_1\text{-}1),\ k_A = N_1\} \tag{4.14}$$

For a node in either mode:

$$n(q) = k_D + k_P + k_A \tag{4.15}$$

Definition (reachable nodes): *Given a node A in receiver mode for the* $q_{a\text{-}th}$ *cycle and a node B in transmitter mode for the* $q_{b\text{-}th}$ *cycle, node A is reachable by node B if:* $A_{RX}^{q_a} \cap A_{TX}^{q_b} \neq \emptyset$

Theorem: *Given a node A in receiver mode and node B in transmitter mode, node A is guaranteed to be reachable by node B if Equations (4.1-4.3) are satisfied.*

The proof of the theorem is divided into two scenarios: *theoretical case* (charging rate is constant) and *realistic case* (due to the slow changing property of MFCs, it's assumed that the charging rate is always constant within a duty cycle. In *realistic cases*, charging rate is changing randomly within a certain range at different cycles)

A. Proof: Theoretical Case

In this part, node B is considered to start its transmitter mode randomly during node A's *deepsleep* events. If not, according Equation 4.3, the start of node B's transmitter mode at the next cycle will definitely fall in the *deepsleep* events of node A's next cycle. Therefore, for the sake of simplicity, denote ¢ $(1 \leq ¢ \leq \frac{T_{DS1}}{T_p})$ to be the event number when node B starts its transmitter mode, it needs traverse at least $[T_{DS1} + T_{PD} - ¢]/T_p$ *deepsleep* events of node A to reach its *active* events. However, this requirement can be easily relaxed if another duty cycle of node A is considered.

Note that according to Equation 4.1, $T_{PD} = T_{probing} - T_p = N_2(T_d + T_p)$. Let

$$\begin{aligned} M &= [\frac{T_{DS1}}{T_p} + \frac{T_{PD}}{T_p} - ¢]\bmod[\frac{T_{DS2}}{T_p} + 1 + N_2(\frac{T_d}{T_p} + 1)] \\ &= [k_D^{RX} + N_2(\frac{T_d}{T_p} + 1) - ¢]\bmod[k_D^{TX} + N_2(\frac{T_d}{T_p} + 1) + 1] \end{aligned} \tag{4.16}$$

and

$$q = ceil(\frac{[k_D^{RX} + N_2(\frac{T_d}{T_p}+1) - \text{¢}\,]}{[k_D^{TX} + N_2(\frac{T_d}{T_p}+1)+1]}) \tag{4.17}$$

where *ceil*(*x*) indicates the smallest following integer of *x*.

Consequently,

$$M \le k_D^{TX} + N_2(\frac{T_d}{T_p}+1)+1 \tag{4.18}$$

and

$$(q-1)[k_D^{TX} + N_2(\frac{T_d}{T_p}+1)+1] + \text{¢} + M = k_D^{RX} + N_2(\frac{T_d}{T_p}+1) \tag{4.19}$$

Dividing T_p at both sides of Equation 4.3,

$$k_D^{TX} + 1 \le N_1(\frac{T_{IL}+T_{PD}}{T_p}) - \frac{T_{PD}}{T_p} \tag{4.20}$$

Combining Equation (4.18) and (4.20):

$$M - N_2(\frac{T_d}{T_p}+1) \le k_D^{TX} + 1 \le N_1(\frac{T_{IL}+T_{PD}}{T_p}) - \frac{T_{PD}}{T_p} \tag{4.21}$$

Combining Equation (4.19) and (4.21):

$$k_D^{RX} - (q-1)[k_D^{TX} + N_2(\frac{T_d}{T_p}+1)+1] \le k_D^{TX} + 1 \le N_1(\frac{T_{IL}+T_{PD}}{T_p}) - \frac{T_{PD}}{T_p} \tag{4.22}$$

Rewriting Equation (4.22),

$$
\begin{aligned}
k_D^{RX} &\le k_D^{TX} + 1 + (q-1)[k_D^{TX} + N_2(\frac{T_d}{T_p} + 1) + 1] \\
&\le (q-1)[k_D^{TX} + N_2(\frac{T_d}{T_p} + 1) + 1] + N_1(\frac{T_{IL} + T_{PD}}{T_p}) - \frac{T_{PD}}{T_p}
\end{aligned} \tag{4.23}
$$

It can be derived from Equation (4.19) that:

$$(q-1)[k_D^{TX} + N_2(\frac{T_d}{T_p} + 1) + 1] - \frac{T_{PD}}{T_p} \le k_D^{RX} \tag{4.24}$$

Combining Equation (4.23) and (4.24):

$$k_D^{RX} \le k_D^{TX} + 1 + (q-1)[k_D^{TX} + N_2(\frac{T_d}{T_p} + 1) + 1] \le k_D^{RX} + N_1(\frac{T_{IL} + T_{PD}}{T_p}) \tag{4.25}$$

Equation (4.25) indicates that the start of node B's active events at the q_{th} duty cycle will be in the range of nodes A's active events. Combining Equation (4.19) and (4.25), it suggests that *there exist* $0 \le h_2 \le N_2\text{-}1$, $1 \le i \le T_{IL}/T_p$, *and* $0 \le h_1 \le N_1\text{-}1$, *such that*:

$$k_D^{TX} + 1 + (q-1)[k_D^{TX} + N_2(\frac{T_d}{T_p} + 1) + 1] + h_2(1 + \frac{T_d}{T_p}) = k_D^{RX} + \frac{T_{PD}}{T_p} + i + h_1(\frac{T_{IL} + T_{PD}}{T_p}) \tag{4.26}$$

Equation (4.26) is equivalent to:

$$A_{TX}^{q} \cap A_{RX}^{1} \ne \varnothing \tag{4.27}$$

The proof is complete. □

B. Proof: Realistic Case

According to Equation (4.4) and (4.5), only T_{DS1} and T_{DS2} will be affected if charging rate is changing. These two parameters will be changing within a certain range:

$$T_{DS1} \in [T_{DS1}^{\min}, T_{DS1}^{\max}],\ T_{DS2} \in [T_{DS2}^{\min}, T_{DS2}^{\max}]$$

Consequently,

$$(q_2-1)\frac{T_{DS2}^{\min}}{T_p} \le \sum_{m=1}^{q_2-1} k_D^{TX}(m) \le (q_2-1)\frac{T_{DS2}^{\max}}{T_p} \tag{4.28}$$

$$\begin{aligned}&(q_1-1)\frac{T_{DS1}^{\min}}{T_p}+(q_1-1)\frac{T_{IL}+T_{PD}}{T_p} \le \sum_{m=1}^{q_1-1} k_D^{RX}(m)+(q_1-1)\frac{T_{IL}+T_{PD}}{T_p}\\ &\le (q_1-1)\frac{T_{DS1}^{\max}}{T_p}+(q_1-1)\frac{T_{IL}+T_{PD}}{T_p}\end{aligned} \tag{4.29}$$

As $T_{DS1}^{\min}, T_{DS1}^{\max}, T_{DS2}^{\min}, T_{DS2}^{\max}, T_{IL}, T_{PD}$, and T_p are constant parameters, there exist integers $q_1 > 1$ and $q_2 > 1$ such that:

$$\sum_{m=1}^{q_2-1} k_D^{TX}(m) = \sum_{m=1}^{q_1-1} k_D^{RX}(m)+(q_1-1)\frac{T_{IL}+T_{PD}}{T_p} \tag{4.30}$$

Combining Equation (4.26) and (4.30), *there exist* $q_1 > 1$, $q_2 > 1$, $0 \le h_2 \le N_2-1$, $1 \le i \le T_{IL}/T_p$, *and* $0 \le h_1 \le N_1-1$, *such that*:

$$\begin{aligned}&k_D^{TX}+1+\sum_{m=1}^{q_2-1} k_D^{TX}(m)+(q-1)[k_D^{TX}+N_2(\frac{T_d}{T_p}+1)+1]+h_2(1+\frac{T_d}{T_p})\\ &=k_D^{RX}+\frac{T_{PD}}{T_p}+i+h_1(\frac{T_{IL}+T_{PD}}{T_p})\sum_{m=1}^{q_1-1} k_D^{RX}(m)+(q_1-1)\frac{T_{IL}+T_{PD}}{T_p}\end{aligned} \tag{4.31}$$

Equivalently, Equation (4.31) is:

$$A_{TX}^{q_2} \cap A_{RX}^{q_1} \neq \varnothing \tag{4.32}$$

The proof is complete. □

Corollary: *When the charging rate is constant, the upper bound of probe reception time is two receiver-mode cycles; when the charging rate is changing randomly within a certain range, the upper bound of probe reception time is q_1 receiver-mode cycles, and q_1 can be determined by Equation (4.31).*

According to the theorem, if Equation (4.3) cannot be guaranteed due to the varying charging rate, then node A may not be reachable by node B. However, as $T_{DS2} \in [T_{DS2}^{\min}, T_{DS2}^{\max}]$, Equation (4.3) can be slightly modified so that the protocol can still guarantee probe packet reception for the worst case scenario:

$$T_{DS2}^{\max} + T_p \leq T_{active} - T_{PD} \tag{4.33}$$

4.2 Energy Neutral Protocol in Multi-hop MFC Networks

As demonstrated in Section 4.1, guaranteed duty cycle based communication can be approached as a nonlinear optimization problem with energy neutral and guaranteed probing constraints. The formulation was shown for a pair of transmitter and receiver nodes. In a practical MFC sensor network, multi-hop topologies are reasonably expected and must be addressed with a joint routing and scheduling solution.

4.2.1 Protocol Basics

It's noted in this chapter that if a node has packets to send (either it receives a new packet from another node or it generates a new packet), it will go to TX mode; otherwise it will stay in RX mode waiting for the probe packet. When a node is either in TX or RX mode, it will follow the energy neutral transmitter or receiver duty cycling as shown in Figure 4.1. There are two issues that the protocol needs to address.

A. Packet Collision

Collision avoidance is a basic task of Medium Access Control (MAC) protocols. In this study, there are two types of packet collisions: two probe packets collide; one probe packet collides with an on-going data packet transmission. Data packet size is considered to be at the same order of the probe packet, which is much less than the recharging time. In this study, it's assumed that transmitter's probe packet can start anywhere of another node's RX mode cycle with equal probability.

Utilizing the proposed protocol, the probability of packet collision is extremely low. Assume the number of nodes at current hop is H_1, the number of nodes at the next hop is H_2, the number of nodes at the previous hop is H_3. At a given time, a node can be either in TX mode or RX mode, with equal probability ($P_{TX} = P_{RX} = 0.5$). Packet collision happens when there are at least two nodes (among all the nodes at three hops) in their TX mode and they are transmitting probe packets or data packets at the same time.

At a given time, a node in TX mode is sending a probe packet with probability:

$$P_{probe|TX} = \frac{(N_2 + 1)T_p}{T_{DS2} + (N_2 + 1)T_p + N_2 T_d + T_{data}} \tag{4.34}$$

A node in TX mode is sending a data packet with probability:

$$P_{\mathrm{data}|TX} = \frac{T_{data}}{T_{DS2} + (N_2 + 1)T_p + N_2 T_d + T_{data}} \tag{4.35}$$

Therefore, the collision probability for two nodes transmitting at the same time can be expressed as:

$$P_c = \binom{2}{H_1 + H_2 + H_3} (P_{TX} P_{probe|TX})^2 + (P_{TX} P_{\mathrm{data}|TX})^2 + P_{TX}^2 P_{probe|TX} P_{\mathrm{data}|TX} \tag{4.36}$$

For example, if the typical values are selected: H_1+H_2+H_3=10, T_p=0.32 (msec), N_2=50, T_{DS2}=29.1 (sec), T_d=18.9 (msec), T_{data}=3T_p, the resulting packet collision probability would be 3.62×10^{-13}, which can be considered negelected.

B. Overlapping of TX Mode

As stated previously, each node in this study must alternate between the roles of a transmitter (TX mode) and a receiver (RX mode), so that it can receive data transmitted by a neighbor and later transmit the data to another node closer to the sink. It has been guaranteed by timing constraints that: if one node is in TX mode and the other one is in RX mode, the probe packet will be heard by the receiver. Therefore, if a node does not receive an ACK from the destination node at the end of its TX mode, the TX modes of the two nodes must overlap. This will cause the peer nodes never hearing each other.

In order to solve this problem, the node which does not receive an ACK at the end of TX mode will choose to be in RX modes for a random number of cycles, in the interval defined by the binary exponential backoff contention window, i.e. $[0, 2^l-1]$, where l is the number of cycles when TX modes overlap. That is, after l overlaps, a random number of RX mode cycles between 0 and $2^l - 1$ are chosen. For example, each node will be in RX mode for 0 or 1 cycle and then transmits again after the first overlap. After the second overlap, the nodes will wait anywhere from 0 to 3 RX mode cycles (inclusive) before going back to TX mode. After the third overlap, the nodes will wait anywhere from 0 to 7 RX mode cycles (inclusive), and so forth.

Similar to other truncated backoff algorithms, the exponentiation stops after a certain number of increases, i.e. the maximum window size reaches a ceiling (noted as C in this study), and thereafter does not increase any further. This implies that the maximum value of l can be calculated as $l_{\max} = floor(\log_2(C+1))$. The derivation of average successful probing time when overlapping of TX mode is considered is presented in

Appendix D. It can be seen from the derivation that average successful probing time is a function of l_{max} and the ceiling C, whose effects will be evaluated in Section 4.3.

4.2.2 Minimal Node-to-Sink Delay

In order to have sustainable performance in a multi-hop network, each node needs to carefully select its peer node at the next hop so that the node-to-sink delay is minimal [Noh2008]. As derived in Appendix D, the average probing time for node x to successfully send data to node u at the next hop is noted as $T_a(x,u)$, then

$$T_a^S(x,u) = T_a(x,u) + T_a^S(u,u') \tag{4.37}$$

where $T_a^S(x,u)$ is the node-to-sink delay of node x to reach the sink with node u as the next hop, u' is the node that node u selects as the peer node at the next hop when it starts transmitting. Each node in the network keeps a routing table as shown in Table 4.2. It's worth noting that the routing table takes only 8 bits and therefore is embedded in the probe packets.

Table 4.2: Routing table of node *x* in the network

My geographical location (2 bits)
My charging/discharging information (2 bits)
Minimal average node-to-sink delay $T_a^S(x,u)$ (the initial value is ∞) (2 bits)
Next hop node ID *u* (2 bits)

Consider node x is in TX mode and starts broadcasting probe packets. $N(x)$ is the set of neighbors at the next hop of node x. After another node u in $N(x)$ hears the probe packet, node u will evaluate the routing table and decide whether to send back an ACK packet to node x. Note that node u could be any node in the neighbors of node x. 1) If node u is the next-hop destination node, it will send back an ACK to node x to initiate a data packet transmission; 2) if node u is not the destination node but it results in a less $T_a^S(x,u)$ value than that kept by node x, node u will also send back an ACK notifying that node u should be selected as the next-hop node, the routing table of node x should be updated and a data packet transmission should be initiated afterward; 3) for all other conditions, node u simply ignores the probe packet and continues duty cycling as if the medium had been idle, thus resulting less energy waste on unnecessary actions of unwanted probe packets.

After the data packet transmission, node u will follow the same protocol to find its own next hop receiver with minimal node-to-sink delay. Finally, the shortest path in terms of the minimum node-to-sink delay for node x is found. It is determined by comparing and updating the value of $T_a^S(x,u)$ for each node at next hop of node x, which is described by Equation (4.38).

$$NextNode(x) = \left\{ u \mid \min_{u \in N(x)} \{T_a^S(x,u)\} \right\} \tag{4.38}$$

It is worth noting that $T_a^S(x,u)$ is evaluated to account for all nodes' impacts, either explicitly or implicitly. The disadvantage of this method is that it may take some time for a certain node to find its best route. The worst case is that a new node boots up and it has to compare all of the possible receivers. However, once a node finds the best route, it only needs to maintain the routing table unless a better route is found and therefore the table will be updated.

Furthermore, there may be packet losses due to channel variations. For example, if a probe packet is lost, transmitter will assume receiver does not receive the probe packet and will transmit probe packet again after its power down time T_d. The throughput performance of the proposed protocol will not be affected if the probe is successfully received within T_d. If an ACK packet (for the probe) from the receiver is lost, transmitter nodes will have to increase their backoff time and result in the receiver's energy waste when the receiver awaits the data packet transmission. This will result in longer node-to-sink delay. Finally, if a data packet is lost, the lost data packet will have to be retransmitted in a later cycle, which will result in lower throughput and longer node-to-sink delay. The numerical evaluation in the paper was carried out without assuming channel-induced errors to provide an upper bound performance assessment for such systems. The impact of transmission errors on the performance is an independent issue that is part of the future work.

4.3 Performance Evaluation and Numerical Results

4.3.1 Evaluation Environments

In order to evaluate and demonstrate the correctness and benefits of the proposed protocols, a network simulator written in Matlab was implemented. The simulator is developed based on prowler, a probabilistic wireless network simulator capable of simulating wireless distributed systems, from the application to the physical communication layer [Prowler]. Table 4.3 summarizes environmental parameters used in simulation, which runs on each node. The network model is simulated as stated in Section 4.1 and shown in Figure 4.3. The distances between the nodes are placed accordingly so that the nodes at each hop can only reach directly the nodes at its neighboring hop(s) and the nodes at its own hop.

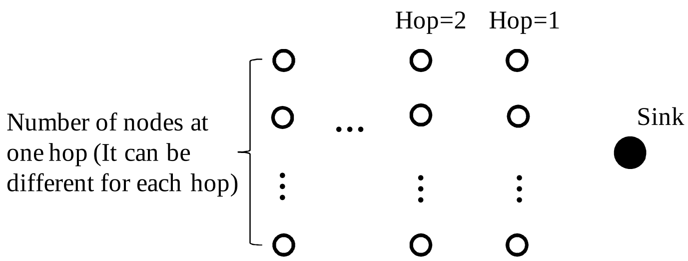

Figure 4.3: The simulated topology

Table 4.3: Environmental parameters adopted for simulation [CC2420; Wong2006; Jurdak2010]

Term	*Description*	*Unit*	*Value*
P_e	Average energy harvesting power	mW	0.25~0.75
N_1	Receiver maximum number of PD/IL cycles	number of cycles	30 (Pre-Determined)
N_2	Transmitter maximum number of PD/PB cycles	number of cycles	50 (Pre-Determined)
E_x	Data packet transmission/receiving energy	mJ	10
T_p	Transmitter probing packet interval (time needed to sample the channel for activity, defined by [Kumar2013])	mSec	0.32
P_t	Packet transmit power	mW	57
P_{PD}	Power down mode power consumption rate	mW	1.4
P_{IL}	Idle listening mode power consumption rate	mW	62
E_{s1}	Switching energy from PD mode to PB/IL modes	uJ	42.3
E_{s2}	Switching energy from DS mode to PD/PB/IL modes	uJ	85.7
M	Total number of packets sent by each node		1000
R	IEEE 802.14.4 raw data rate	kbps	250
D	Data packet size	bits	$\frac{D}{R}P_t = E_x$
T_{DS1}	Receiver deep sleep (recharging) time	sec	Variable (100~3800)

T_{PD}	Receiver power down time	sec	Variable (0.3~10)
T_{IL}	Receiver idle listening time	msec	Variable (4.6~50.6)
T_{DS2}	Transmitter deep sleep (recharging) time	sec	Variable (15~800)
T_d	Transmitter power down time	msec	Variable (5~50)
T_{active}	Receiver maximum PD/IL cycles interval	sec	$N_1(T_{PD}+T_{IL})$
$T_{probing}$	Transmitter maximum PD/probing cycles interval	sec	$N_2(T_p+T_d)+T_p$

Numerical results obtained from simulations are shown and discussed in the following sections. It's worth noting that in this study, MFC has a slow-varying charging property and thus its average energy harvesting power P_e is considered constant during one cycle time. Variable parameters listed in Table 4.3 are solved using a nonlinear program following the procedure presented in Section 4.1 before each duty cycle for different P_e. The range of each variable is verified with simulation and listed in Table 4.3 for reference.

4.3.2 Numerical Results and Discussion

A. Performance under the Worst Scenario

In this part, performance of different aspects are shown for the worst case scenario, that is, the energy harvesting rate is varied within the largest range (0.25~0.75 mW).

Backoff window size affects minimal average successful probing time

Average successful probing time is an important metric to evaluate as it indicates how fast a node can find its peer node. Its performance with respect to maximum backoff window size C at different number of hops was simulated as presented in Figure 4.4. The number of nodes at each hop was set as 4. In this simulation, the major energy consumption component is the wireless radio [MICAz]. Due to the ultra-low and dynamic MFC energy harvesting rate, the simulated average RX mode (cycle) time of all nodes after a complete simulation run is 1.05 hour. The average successful probing time in Figure 4.4 is presented as normalized to this value.

The large error bar and high average successful probing time when *C* equals to one are mainly due to the fact that the probability of nodes choosing the same window size is high. When *C* equals to three, nodes can choose to backoff 0, 1, 2 and 3 RX mode time before entering TX mode again, hence the probability of two nodes choosing the same backoff window size significantly decreases, resulting small error bar and average successful probing time. However, when *C* keeps increasing, nodes may choose large backoff window size before transmitting again, which is not beneficial at all due to the big waiting time. It is recommended by simulation that a maximum backoff window size of four is optimal for the designed environment. The number of hops has no significant effect on the average successful probing time performance.

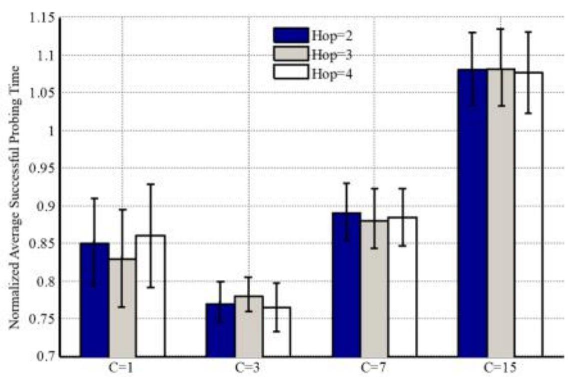

Figure 4.4: Average successful probing time vs. maximum backoff window size. Note that the y-axis is normalized to the simulated average RX mode (cycle) time.

Proposed protocol consumes lowest average power

Average power consumption is another important metric that is worth evaluating. Due to the energy neutral nature of the proposed protocol, the average power consumption rate per node is always less than the average power harvesting rate. Under the same condition, MAC protocols such as B-MAC [Polastre2004], On-Demand (OD-) MAC [Fafoutis2011] and X-MAC [Buettner2006] are not able to keep energy neutral operation. In B-MAC, receiver nodes will periodically wake up and listen for long preambles sent by transmitter. X-MAC breaks the long preamble into several short preambles with target address information. This saves some energy by letting nodes sleep between short preambles, but the overall preamble time is still long when nodes spend long time charging themselves. In OD-MAC, nodes with no packets in the queue will periodically send beacons and nodes with packets to send will listen for beacons first. The long preamble energy waste from transmitter is saved in OD-MAC, but how to optimize the node-to-sink delay is not mentioned. Figure 4.5 shows the per-node average power consumption performance with different number of nodes for all the protocols. Other MAC protocols such as WiseMAC [Hoiydi2004] and MiX-MAC [Merlin2009] were not compared in this study for the following reasons: 1) WiseMAC is not applied in an ad-hoc network topology but an infrastructure network where access point have unlimited power supply; 2) MiX-MAC is generally a mixture of X-MAC, SpeckMAC [Wong2006] (variant of B-MAC) and CSMA-MPS [Mahlknecht2004] (variant of B-MAC), whose performance will be bounded by the best and worst scenarios of the pool of MAC protocols. The simulation parameters were carefully tuned so that the network average throughput as seen by the sink was roughly the same. The number of nodes at each hop was fixed at two and the placement was identical for all the protocols.

It can be clearly seen that when reaching the same average throughput with the ultra-low power input, B-MAC, X-MAC and OD-MAC were not able to guarantee energy neutral operation while the proposed protocol (MFC-MAC) kept the network alive. Note that the average power consumption increased significantly as the number of nodes increased for B-MAC. This is because in B-MAC, nodes wake up when they hear preamble packets from other senders as they do not know if the transmission is intended for them or not. There are slight per-node power consumption increases with higher number of nodes for other protocols, but their influences are not significant.

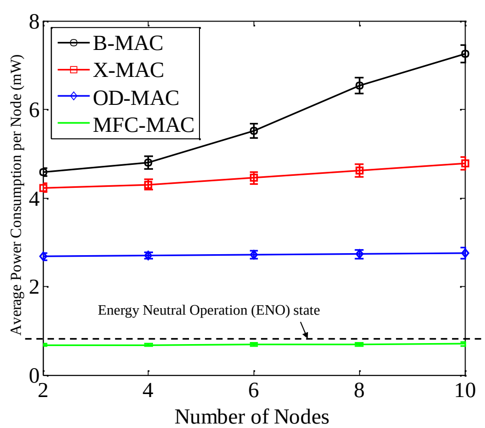

Figure 4.5: Per-node average power consumption with different number of nodes

Average node-to-sink delay is mainly determined by the number of hops

The node-to-sink delay measures the total latency of a packet transmitting from its source node to the sink. Section 4.2 introduces a method to obtain minimal node-to-sink latency. While it does need a long time for each node to update their information tables, the method proves to be effective even at the worst scenario. Figure 4.6 shows the average node-to-sink latency performance under different number of hops. The number of nodes at each hop was fixed at 4. The simulation was repeated for different values of maximum backoff window size *C*. As stated previously, the simulated latency was also normalized to the average RX mode (cycle) time.

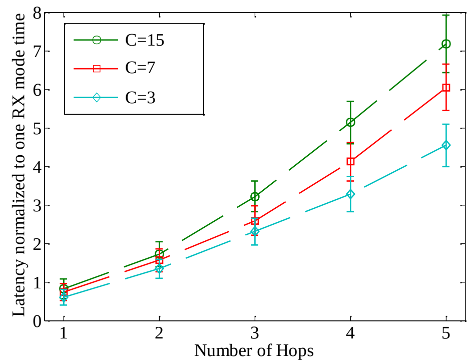

Figure 4.6: Average node-to-sink latency vs. number of hops

Due to the varying energy harvesting rate, nodes that need more hops to reach the sink tend to have long latency with large deviation. The node-to-sink latency is a linear function of the number of hops in the network. As the backoff window size increases, the average successful probing time increases, so does the slope of the latency. It is worth noting that such asynchronous protocol's power consumption is bounded by the cost of nodes' active periods, thus power consumption is usually reduced at the expense of increasing latency. How to better handle the tradeoff between power consumption and latency remains a challenge for duty-cycle based protocols.

B. Performance with Different Harvesting Rate Variations

In this part, protocol performance at different harvesting rate variations were simulated and compared. The number of nodes at each hop was fixed at 4. The maximum backoff window size *C* was fixed at 3 and the results were normalized to the average RX mode (cycle) time. Figure 4.7 shows the average node-to-sink delay and successful probing time performance when power harvesting rate was constant (0.25 mW and 0.75 mW), which represents the best case scenario. Both the probing time and node-to-sink delay dropped when the harvesting rate was selected higher. The deviation of average successful probing time significantly decreased when there were no harvesting rate variations. This is because each node kept the timing information of its next hop peer node in its routing table, the constant harvesting rate made their TX or RX mode time unchanged. Moreover, it's interesting to note that the probing time was independent of the number of hops. The probing process happened between nodes at two adjacent hops and thus it was solely determined by the timing information of peer nodes.

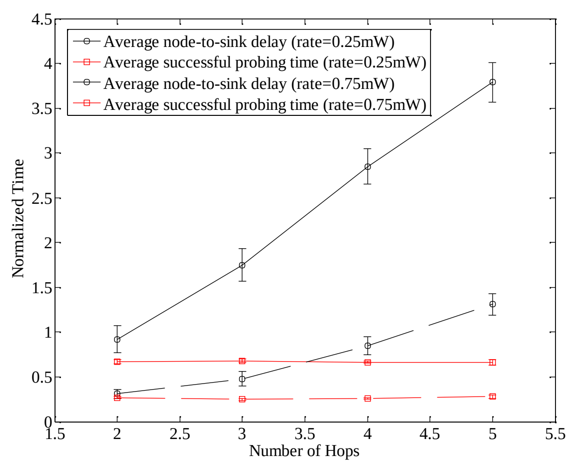

Figure 4.7: Average node-to-sink delay and successful probing time performance when power harvesting rate is constant: 0.25 mW and 0.75 mW

It is also shown that the average node-to-sink delay increased linearly with the number of hops. This observation is consistent with Figure 4.5 except that the deviation was much smaller which was due to the constant harvesting rate. However, as node started their data transmission randomly, the TX mode overlapping problem still existed and thus small deviation of node-to-sink delay was expected. The protocol performance at different harvesting rate variations is presented in Figure 4.8. For each variation, nodes randomly chose a charging rate value within the variation range at each simulation run before data transmission operation. The variation ranges selected for simulation were: 0.25 mW~0.45 mW, 0.45 mW~0.65 mW and 0.25 mW~0.65 mW.

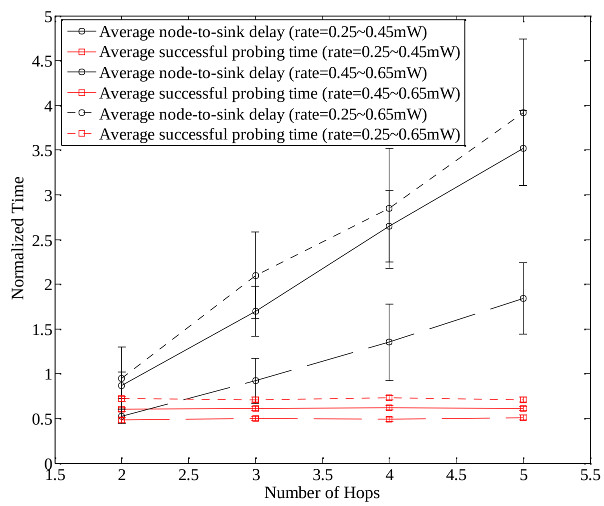

Figure 4.8: Average node-to-sink delay and successful probing time performance at different harvesting rate variations: 0.25 mW~0.45 mW, 0.45 mW~0.65 mW and 0.25 mW~0.65 mW

With both the lower and upper bound of variation range being the highest (0.45 mW~0.65 mW), protocol performance was shown to be the best. Figure 4.8 also indicates that a large variation range would result in poor average node-to-sink delay and successful probing time performance. Although node has the possibility to harvest more power at a given cycle to shorten its charging time when the harvesting rate variation range is large, it's also likely that the peer node is expecting a longer charging time and thus the two nodes will take longer time for probe packet reception, resulting in longer network latency.

This suggests that when designing a wireless network with ultra-low power sources like MFCs, optimal deep sleep time derived from the scheduling protocol must be selected to account for performance degradation caused by MFC's power harvesting rate variations.

4.4 Summary

This chapter presents the design and performance evaluation through simulations of a communication protocol for wireless sensor networks with dynamic and limited energy harvesting source—MFC to be more specific. The protocol includes the following key properties. Firstly, a duty-cycle based energy neutral transmission sleep/wakeup policy between peer nodes was proposed to guarantee packet reception. Secondly, an energy neutral joint scheduling and routing protocol in multi-hop MFC networks was introduced, aiming to provide network-wide packet communications under extremely limited node resource such as ultra-low power supply. Finally, a simulation tool was developed to evaluate the protocol performance. Simulation results show that proposed communication protocol is able to provide sustainable and reliable data transmission with ultra-low and dynamic power input.

CHAPTER FIVE

MESH DATA TRANSMISSION ROUTING AND IMPLEMENTATION WITH MULTI-TIER NETWORK ARCHITECTURE

Wireless monitoring infrastructure usually involves deploying wireless sensors among the environment and analyzing those data which is wirelessly transmitted from local sensors to a remote server via wireless transit system. Most of current transit infrastructures are wireless technology dependent. A multi-tier network architecture based on wireless mesh network is widely used to monitor various environments. How to efficiently route sensor data traffic within the infrastructure is the main challenge. The goal of this section is to study a complete operational system with setup of algorithms to learn the potential benefits of centralized routing control.

In this chapter, the focus is on wireless data communication in wide area environmental monitoring application. The mesh-based multi-tier network architecture is introduced first. A simple routing protocol is then implemented and evaluated based on two sets of experimentations: Matlab simulation and embedded X86 devices based demonstration.

5.1 Multi-Tier Mesh Network Modeling

5.1.1 Wireless Mesh Network

A backbone mesh network is formed by mesh nodes communicating with each other in a multi-hop fashion via wireless links. Data from mesh clients is relayed by mesh routers, and the network coverage is adjusted by controlling the number of mesh routers.

The mesh backbone can be built using various types of radio technologies, with IEEE 802.11 being the dominant standard for most applications [Prasad2006; Bemmoussat2012]. The Internet gateway is the Internet access device, which routes all information packets to a remote server. Figure 5.1 illustrates the general architecture of a wireless mesh backhaul network. Wireless mesh networks consist of three types of nodes: Internet gateways, mesh routers, and mesh clients. It should be noted that a physical node may work as an Internet gateway and mesh router at the same time. Mesh clients are usually wireless sensors connecting to mesh routers. Mesh routers are usually resource-rich nodes, while mesh clients have limited memory and computational power.

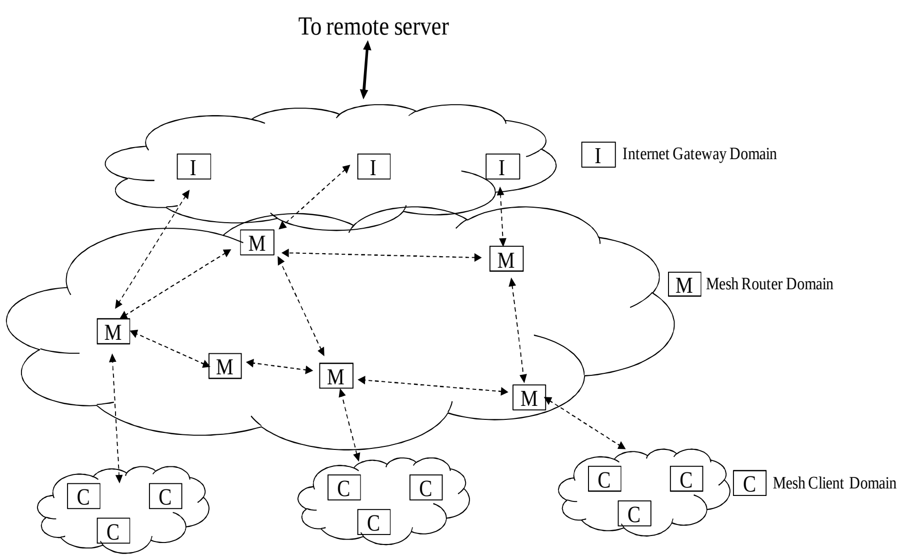

Figure 5.1: Multi-tier wireless mesh network

5.1.2 Mesh Network System Model

In the following sections, the focus is on routing efficiency in the mesh router and Internet gateway domains as shown in Fig. 5.1. Mesh routers are referred as mesh nodes in the proposed system model. Consider that the nodes in the wireless mesh network are single radio mesh nodes. The mesh network is modeled by a directed graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ is a set of n nodes and E is a set of possible directed communication links between any two nodes $u, v \in V$. Among the n mesh nodes, k of them will have Internet access (Internet gateways). For simplicity, let $A = \{a_1, a_2, \ldots, a_k\}$ be the set of k Internet gateway nodes, where a_i is node v_i, for $1 \leqslant i \leqslant k$. The set of non-gateway nodes is thus denoted as V-A, and the number of non-gateway nodes is $m = n - k$. For each node $i \in V$, there exists a set of one-hop neighbors, B. The packet from a source $s \in V$ to a destination d is transmitted along a link $(s, d) \in E$, if d is also an element of B; otherwise, it is transmitted along a multi-hop route made up of a link (s, j) and a route $[j, d]$, where j is a node in B, and (s, j) is a link in E. The route $[j, d]$ represents a route from node j to node d through a subnet $S = (V - \{s\}, E - \{(s, j): j \in B\})$. Furthermore, each node $i \in V$ has a capacity constraint $\lambda_{i,max}$, which guarantees a maximal load on each node. The current load through source node s to destination node d is denoted as λ_{sd}. It is measured in terms of throughput. The system model diagram is shown in Fig. 5.2.

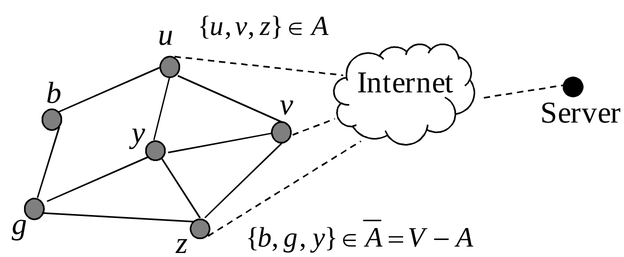

Figure 5.2: Mesh network system model diagram

5.2 A Centralized Routing Policy and Implementation

From the previous section, we can see that the key component in the multi-tier network architecture is wireless mesh network (WMN) which connects local sensors with a remote server. How to efficiently route data traffic in WMN remains a challenge. In this section, a simple centralized routing policy for WMN is presented. It computes a path between each mesh node and an Internet gateway, and prevents allocating overloaded bandwidth for certain nodes. By performing such protocols in the remote server, the network management overhead is offloaded from mesh nodes and Internet gateways, and enables the deployment of simple wireless routers.

5.2.1 Routing Policy

As the links are wireless channels, there is probability that certain links are in bad transmission quality such that poor packet reception rates are obtained at some receiver nodes. Consider the problem of determining the best route between two nodes in the mesh network using path quality. Let P be a candidate path. Let link e belong to path P. The link quality, LQ, is calculated as the fraction of successful packets that are received by a neighbor within a window period. When interference from other nodes is assumed negligible, LQ should only be a function of packet success rate of the specified link. In particular, for the adopted ACK-based transmission method, let $p_{e,f}$ and $p_{e,r}$ be the packet success probabilities of link e in the forward and reverse directions, respectively, the expected link quality of link e is given by:

$$LQ(e) = p_{e,f}\, p_{e,r} \tag{5.1}$$

The path quality, $f(P)$, is then defined as the probability a packet is not dropped across any link along the path, i.e.,

$$f(P) = \sum_{e \in P} LQ(e) \tag{5.2}$$

and the routing policy is to choose the path with the highest path quality $f(P)$ while maintaining the maximal link flow capacity.

5.2.2 Protocol Implementation

The final destination for every data packet is a remote server, denoted by S, which also hosts the central controller. Packets must first reach Internet gateways, which have Ethernet connections to route packets to the remote server. The central controller executes the algorithm shown in Table 5.1 to find routes. The algorithm is heuristic and deliberately simple to reduce execution time at the remote server. The objective is to find a route with the best path quality between a source node (s) and the remote server (S), while guaranteeing that no mesh nodes are overused. To find the best route, it locates the next-hop neighbor through which traffic can be best routed to reach the remote server using Eq. (5.2). It is executed for each node along the route until the traffic reaches Internet gateway, where it is then directed to the remote server through the Internet backbone.

Table 5.1: Routing algorithm

Step 1:	consider a source node s in V-A, identify all neighbor nodes in set B, **if** a route [s, S] is found, **do** return

	else, do go to Step 2
Step 2:	identify all links (s, j), for $j \in B$ go to Step 3

Step 3:	initialize MaxLQ = 0
	for each node j in B, **do**
	if $j \in A$, **do**
	if LQ_{sj} > MaxLQ, **do**
	MaxLQ = LQ_{sj}
	Let $\boldsymbol{g} = \boldsymbol{j}$
	else, **do**
	go to Step 4
	end for
	go to Step 5

Step 4:	find node $\boldsymbol{j}$ and $\boldsymbol{g}$ that result in the best path quality, calculated by Eq. (5.2)
Step 5:	**if** current load on node j $\lambda_j < \lambda_{max}$, **do** node $\boldsymbol{j}$ is node s's next-hop neighbor, and node $\boldsymbol{g}$ is the designated Internet gateway node route [s, S] is found return **else**, **do** eliminate node $\boldsymbol{j}$ in B go to Step 3

In this algorithm, the first step is to identify the neighbor nodes of source node s. If the best path is already found, the algorithm returns; otherwise, it identifies all the links from node s to its neighbors. At the next step, two conditions are checked: if the neighbor node happens to be an Internet gateway, it locates the gateway that gives the best link quality; otherwise, it uses Eq. (5.2) to locate neighbor node and corresponding Internet gateway that result in the best path quality. Next, the algorithm checks if the resulting data traffic through the selected neighbor is less than the maximum load requirement. This guarantees that each relay node is traffic balanced. If the identified neighbor would carry more traffic than allowed, the algorithm eliminates the node from the neighbor list and identifies a new node by looping back through the same steps. Figure 5.3 shows the flowchart of the routing policy.

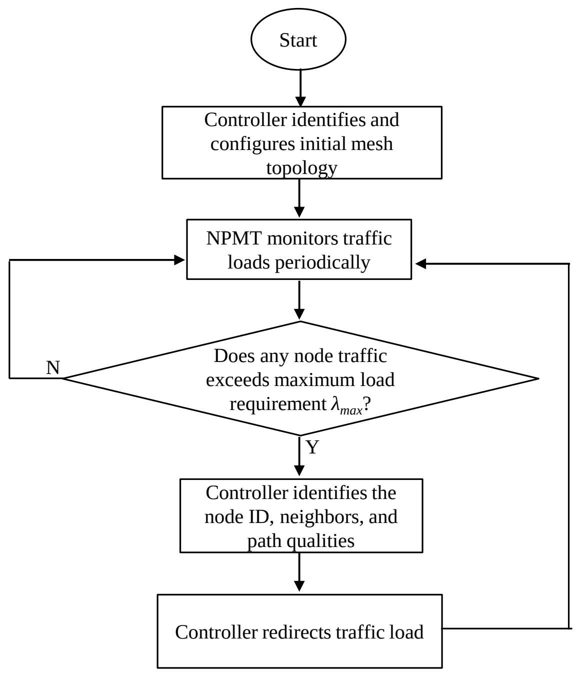

Figure 5.3: Flowchart of the centralized routing policy

The algorithm can be implemented as follows. Each node transmits beacon messages to inform neighboring nodes about its existence. Figure 5.4 shows the format of a beacon message. Besides physical layer header and checksum, each beacon message contains the address of the originator, the address of the node transmitting the packet, a sequence number, and the link quality information of its one-hop neighbors. Each neighbor changes the sending address to its own address, updates the link quality information, and re-broadcasts the message to inform its neighbors about the existence of the initiator, and so on. Upon receiving its own message, the originator performs a bidirectional link check to verify that the detected link can be used in both directions. Further, to estimate the packet reception probabilities in Eq. (5.1), the periodic beacon packets can be used as sentinels, as in [Padhye2004; Couto2003]. Traffic load between a source node and the remote server is measured periodically by a custom network performance measurement tool (NPMT). When a new mesh node joins the network, its initial route to the remote server is identified by broadcasting. Its location, in terms of number of hops to an Internet gateway, is recorded by the controller. Once flow paths are setup based on the initial route, the NPMT is able to monitor the load going through the new node.

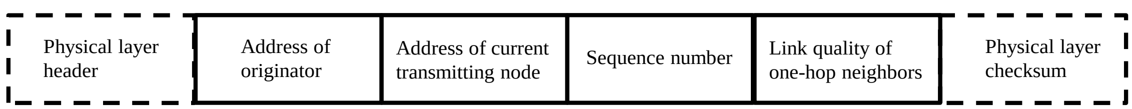

Figure 5.4: Packet format of a beacon message

The wireless infrastructure in this work has several advantages over the existing methods for WMNs that are typically based on distributed solutions. In contrast to the existing solutions that mostly focus on maintaining connectivity, the solution of this dissertation considers the issue of not overloading certain nodes in WMN. Moreover, since the routing and bandwidth allocation decisions are made at the controller, the mesh nodes need only modest computation capabilities.

5.3 Routing Performance Evaluation: Simulation

As different performance metrics have unique goals/formations for specific scenarios, it is not feasible to compare performance theoretically using mathematical models between the proposed protocol and existing solutions. Instead, in this section, Matlab simulation based experiments are designed to compare the performance of the centralized routing policy with three widely-used mesh routing protocols.

5.3.1 Matlab Simulation Model

Figure 5.5 presents the flow of Matlab simulation blocks. First a packet generator is programmed to generate traffic and this traffic is output to mesh routers. Each packet is identified by a destination IP address. The input to packet generator includes packet size definitions and packet arrival rate. In order to focus on the effects of different routing protocols, packet arrival rate is preset as a constant number.

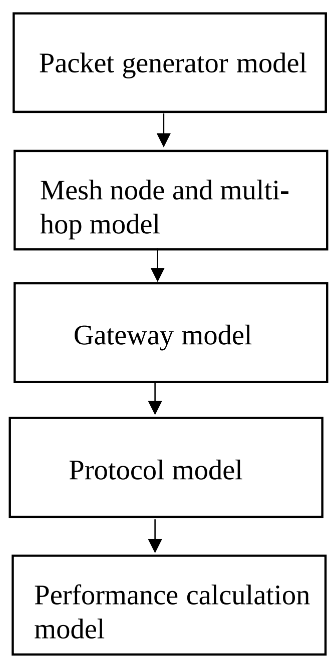

Figure 5.5: Matlab simulation blocks

Mesh node and multi-hop model takes the output from the packet generator block. It is responsible for: 1) routing packets, 2) node placement and neighbor labeling, 3) traffic flow check, and 4) executing protocol model when needed. In wireless sensor networks, most of nodes identify its neighbors by checking received signal strength indicator (RSSI). In order to simplify simulation process, the simulation program developed in this section automatically assign nodes locations and label their neighbors. The model also checks traffic flow of each mesh node: if it exceeds certain threshold (overloaded), protocol model is executed to re-route the traffic. In the meantime, the function of gateway model is for packets reception. Packet labeled as received is considered as a successful transmission, and is output to protocol and performance calculation models.

The function of protocol model is basically to execute the routing protocols of interests. In this dissertation, the routing protocols of interest are:

1) TDMA scheduling and Bellman-ford routing algorithm [Capone2010]. It is a classic routing protocol in wireless mesh network.

2) Link scheduling and expected transmission time based routing [Draves2004]. Expected transmission time is the most-used metric in many mesh routing papers.

3) Joint scheduling and optimized routing [Ning2012]. Due to the optimization process, this protocol shows better performance than most widely-used routing protocols in terms of fairness.

4) The centralized routing policy (CR) presented in this dissertation.

Finally, the function of performance calculation model is to generate results. The metrics of interests are run time and fairness. Protocols and metrics will be discussed in the following section.

Parameters considered for simulation are provided in Table 5.2. Number of gateways is fixed at 5: routing protocols running on mesh nodes are the focus of this simulation. Mesh nodes are randomly placed in terms of coordinates. The number of mesh nodes are 9, 35, and 100. The purpose is to show run time differences when there are different number of nodes in the network.

Table 5.2. Simulation parameters

Parameters	*Value*
Number of gateways	5
Number of mesh nodes	9, 35, 100
Mesh nodes placement	Random
Packet size	100 bytes
Mesh node capacity	10 KBps
Gateway capacity	Unlimited
Packet arrival rate	0.1s (10 packets/s)

5.3.2 Simulated Routing Protocols

In this section, three algorithms are presented. The first algorithm is a traditional algorithm combining TDMA scheduling and Bellman–Ford routing protocol (TSBF). After that, a heuristic algorithm based on link scheduling and expected transmission time

(ETT) routing metric (HLSE) is presented. At last, a cross-layer optimization protocol based on joint scheduling and routing optimization algorithm (JSRO) is presented.

A. TDMA Scheduling and Bellman-Ford routing Algorithm

In MAC layer, TDMA scheduling algorithm traverses from the first node to the last node. In network layer, this algorithm require the knowledge of the links that exist in the network. It utilizes the classical Bellman–Ford algorithm for shortest path routing. This algorithm can solve the shortest monophyletic point problem that is similar with Dijkstra algorithm. However, solving negative edge together with positive edge is a strong advantage where Bellman–Ford algorithm outperforms Dijkstra algorithm. This algorithm is simple to implement, however, it requires time synchronization, which is difficult to achieve in practical sensor network deployment.

B. Heuristic Algorithm based on Link Scheduling and ETT Routing Metric

- Scheduling Metric Design

Eq. (5.3) is used to define the priority of each link, and slots are assigned according to link priority. The MAC layer metric is defined as:

$$L(\mathrm{i, j}) = \mathrm{a}\frac{1}{1+Q_{ij}} + b\frac{Q_{blocked}}{1+Q_{blocked}} \tag{5.3}$$

In Eq. (5.3), $Q_{blocked} = \sum_{(k,l):\text{blocked by }(i,j)} Q_{kl}$, and Q_{ij} means the length of packet prepared to transmit in link (i,j). A link (k,l) is blocked by (i,j) if $k = i$ or j, or, $l = i$ or j. The weight factors a and b vary between 0 and 1, satisfying $a + b = 1$. The smaller computed metric is the higher priority this link has. The first term gives higher priority to the links with larger queues. The second term cares about the effects of congestion developing on neighboring links that are blocked by the activated links. As shown by previous study [Ning2012], a=b=0.5 gave the system an optimal performance, and these are the numbers used for simulation in this section.

- ETT Routing Metric

Hop count (shortest path) routing metric is the basic routing metric in Ad-Hoc network. However, there may be packets drop at both transmitting and receiving nodes. If the transmitted packets are not received successfully at the receiver, then MAC layer needs to adopt the retransmit mechanism. Therefore, expected transmission times (including retransmission times) is an improved routing metric. After k times of transmissions, the successful transmission probability from source node to destination node is:

$$S(k) = p^{k-1}(1-p) \tag{5.4}$$

where p is the packet loss rate. Expected transmission time (ETT) can be obtained by:

$$ETT = \sum_{k=1}^{\infty} k \times S(k) \tag{5.5}$$

It should be noted that Eq. (5.4) assumes that the packet loss rate is independent with the network size and network characters are symmetrical (i.e., the packet loss probability from the source to destination or the destination to source are the same).

C. Joint Scheduling and Routing Optimization Algorithm

The joint algorithm conducts the cross-layer optimization between MAC layer and network layer. To simplify simulation process, the scheduling algorithm in MAC layer is the same as in HLSE algorithm. The routing metric is based on optimized Bellman-Ford algorithm, which is shown below:

$$D(i,j) = d\left(\frac{Q_{block}}{Q_{ij}}\right) + e\left(\frac{R_{ij}}{R_{\max}}\right)^{¥} \tag{5.6}$$

where d and e are weight factors satisfying $d + e = 1$. R_{ij} is the distance of link (i,j), and $R_{\max}$ is the maximal transmission distance. ¥ is a distance coefficient. These parameters are obtained by mesh node and multi-hop block in the Matlab simulation models.

The smaller value of the computed metric is, the higher priority transmission the link has. The first item gives higher priority to those links with less congestion and larger queue of packets, which may benefit for throughput improvement. The second item encourages the short distance link transmitting first to reduce power consumption and delay. Eq. (5.6) makes a compromise between throughput and power consumption. As throughput is the main focus in this section, d = 0.9 and e = 0.1 were chosen for simulation.

5.3.3 Simulated Routing Protocols

In this section, the above protocols are simulated using the developed Matlab tools. Simulation programs were verified by comparing results shown in the cited papers [Capone2010; Draves2004; Ning2012]. Mesh nodes are randomly placed, three different configurations for total number of mesh nodes are investigated: 9, 35 and 100. Each node randomly starts to send packets and needs to finish sending 1000 packets (received by one of the gateways). Two performance metrics: protocol run time and fairness index are evaluated. Numerical results and analysis are presented in this section.

A. Protocol Run Time

Protocol run time is defined as the execution time duration between the start and end of simulation. It is a very useful metric to show time complexity of each protocol. During simulation, the working station is a Windows 7 laptop with Core i5 CPU clocking at 2.6GHz and 16G RAM. The Matlab tool version number is R2012a. It should be noted that the following results are based on the particular working station configuration. 10 simulations were performed for each protocol with 9, 35 and 100 nodes in the network, respectively. The average protocol run time is shown in Fig. 5.6.

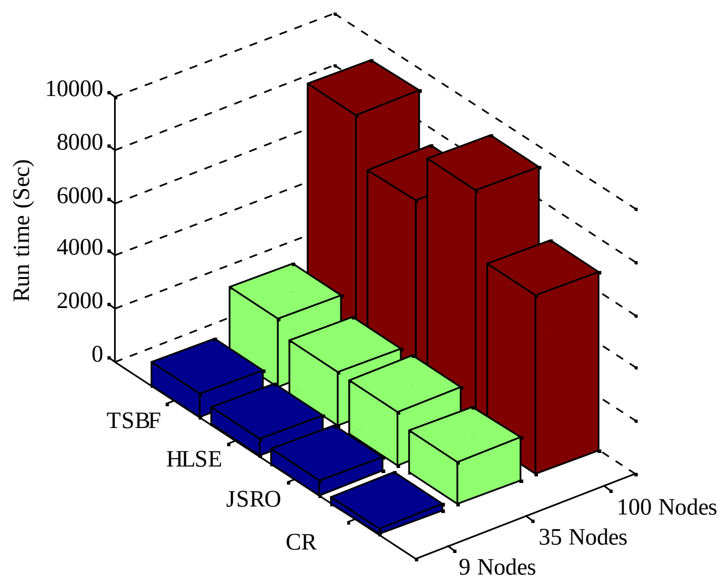

Figure 5.6: Average protocol run time

From simulation, it shows that when the number of nodes increases, the performance of all routing protocols decrease sharply. This is due to the fact that when number of nodes increases, packets will expect an increasing average number of hops to reach gateways. TSBF has the worst performance when there are 100 nodes in the network, resulting averagely more than 2.5 hours to complete a simulation. HLSE improves the run time performance by 19.3% compared to TSBF. The improvement is because of the ETT routing metric. Compared to the shortest hop count metric used in TSBF, expected transmission time provides more routing benefit when there are congestions at certain links.

It can also be seen from simulation that compared to TSBF and HLSE, JSRO takes less time to complete a simulation when there are 9 and 35 nodes in the network, but it takes more time when network size scales up to 100. While the routing optimization algorithm used in JSRO does provide improvement in link scheduling and identifying better route, the complexity of optimization also increases along with network size. There is a tradeoff between routing protocol complexity and performance. It can be derived that for random topology scenarios, time complexity of optimized Bellman–Ford algorithm is $O(V*E)$, where V stands for vertex number and E is defined as link edge number. For traditional Bellman–Ford algorithm, time complexity is estimated as $O(1.39*E)$. This also explains why JSRO has worse performance than TSBF and HLSE when there are a large number of nodes in the network.

Finally, the centralized routing has the best performance in all scenarios, resulting a 25.9% decrease in average protocol run time compared to TSBF when there are 100 nodes in the network. As only the central controller runs the optimization algorithm, in contrast to the fact that all other three protocols run algorithms locally at each node, there is a huge boost in overall performance. It should be noted that the setting up time of controller and initial routes are not considered in simulation. This is a fair assumption as controller and initial routes setup are only one-time process before algorithm is running and the simulation in this section focuses on the protocol performance after nodes start sending packets. Figure 5.7 shows protocol run time standard deviation. It can be seen that performance of TSBF and JSRO fluctuate significantly when there are 100 nodes in network. In the centralized routing scheme, network size has the least impact over its run time.

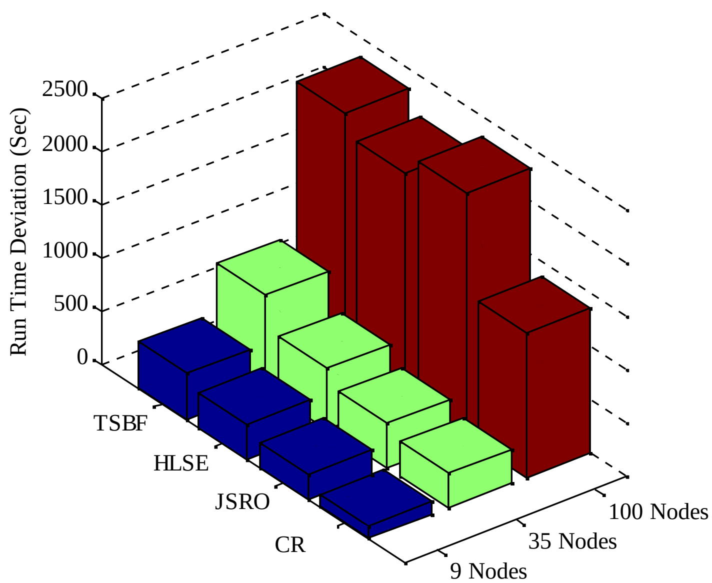

Figure 5.7: Run time standard deviation

B. Fairness Index

Since the transmission between two nodes is fulfilled by link transmission, fairness obtained by the link transmission opportunity is the main concern in this section. Generally, traffic distribution can be weighed among different mesh nodes and an index which can generalize traffic diversity with statistical characteristics is defined as:

$$F(N) = \frac{1}{2N^2 u} \sum_{k=1}^{N} \sum_{n=1}^{N} |u_k - u_n| \quad (5.9)$$

Herein, N stands for the total number of mesh nodes in the network, and u_n is the network traffic coming from the n_{th} mesh node, and u is the arithmetic average of all individuals u_n, that is

$$u = \sum_{i=1}^{N} u_i / N \quad (5.10)$$

It is suggested in [Dianati2005] that, when index F is lower than 0.2, the network traffic distribution is extremely fair. When index F locates between 0.2 and 0.3, it means the network traffic distribution is fair. When index F locates between 0.4 and 0.5, it means there is a big gap for the network traffic distribution. When index F is larger than 0.5, it means there is a great disparity among traffic distribution. Table 5.3 shows the average fairness index calculated from simulation results. Figure 5.8 plots the average fairness index curve with error bars.

Table 5.3: Calculated average fairness index

Protocols	*9 Nodes*	*35 Nodes*	*100 Nodes*
TSBF	0.2837	0.3533	0.5287
HLSE	0.1973	0.3328	0.4672
JSRO	0.1272	0.2258	0.4102
CR	0.0998	0.1165	0.1977

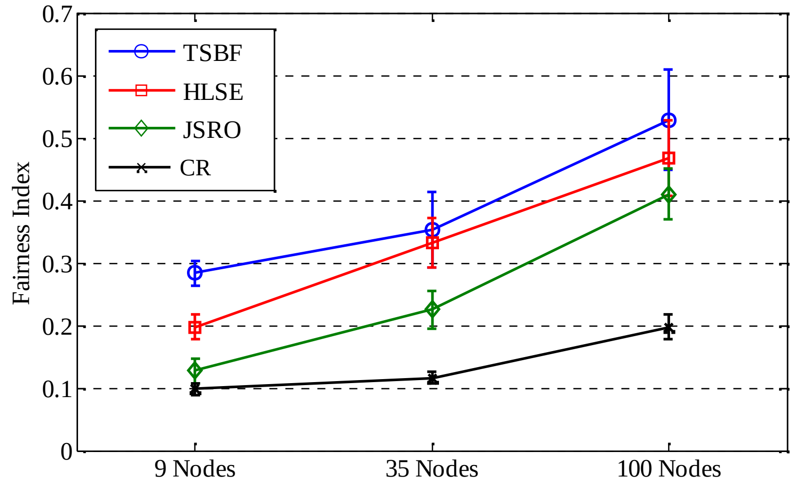

Figure 5.8: Fairness index at different number of nodes

When network size grows, all protocols suffer decreased performance in fairness. This result is as expected as the goal of all protocols is to alleviate traffic load in the bottleneck node and encourage edge links to transmit packets. When network size grows, there will be more "center" nodes suffering overload traffic due to inadequate edge links available. However, even when there are 100 nodes in the network, fairness index for the centralized routing scheme is still less than 0.2, demonstrating an extremely fair network traffic distribution. In this case, its fairness index drops 62.6%, 57.7%, and 51.8%, respectively, compared to TSBF, HLSE, and JSRO. Central controller finds the best routes for packet transmissions with a maximal traffic load threshold at each mesh node, which effectively prevents the network from overloading single node.

5.4 OpenFlow Based Wireless Mesh Infrastructure

To practically implement protocol, special network architecture is needed. Current architecture is tied to one behavior based on the underlying routing protocol. One cannot easily change the routing behavior for various network operations. By applying one of the Software Defined Network (SDN) implementations—OpenFlow, on top of the wireless mesh network, one can more easily install new routing logic by using a centralized controller. In this section, OpenFlow protocol is introduced to help implement traffic redirection. OpenFlow based wireless infrastructure are presented next. With the help of OpenFlow, real-time packet redirection becomes feasible by inserting customized flow rules. The OpenFlow controller is able to make centralized decisions on how to optimally route traffic so that the computational burden at each node is minimized.

5.4.1 Overview

The OpenFlow based wireless infrastructure was inspired by the following observation: WMNs are mainly used as relay networks for sending and receiving information to and from mesh clients via mesh nodes. The OpenFlow controller on the remote server is able to monitor mesh nodes in the WMN and acquire information from them through a secure tunnel. It is possible to shift the route calculation tasks that are typically done by mesh nodes in most distributed protocols to the central controller. In our approach, the controller determines the routes between mesh nodes and Internet gateways and allocates appropriate bandwidth for each traffic flow. To this end, the controller must be aware of the network topology, path quality, node traffic, and distance from each node to every Internet gateway. Distance is measured as the length of the shortest path, with respect to a given routing metric (e.g. hop count). The central controller can easily obtain this information because WMN nodes are typically static (or quasi-static) when there are no node failures. In this work, such information is retrieved by node discovery protocol and NPMT. Based on the collected information, the controller determines optimal routes and configures the mesh nodes accordingly.

Our wireless infrastructure has several advantages over existing WMNs based on distributed solutions. In contrast to existing solutions that focus on maintaining connectivity, our solution also considers the issue of traffic fairness among nodes in the WMN. Moreover, since the routing and bandwidth allocation decisions are made at the controller, the mesh nodes need only modest computational capabilities.

5.4.2 OpenFlow Protocol

OpenFlow is a new industry specification for an open protocol to program flow tables in networking devices. OpenFlow controllers can eliminate the need of programming each networking device independently. The OpenFlow specification defines the interfaces between external controllers and the networking devices as well as many of the external controller functions in terms of setting up flows based upon service profile definitions.

In legacy switch/router structures, there will be a control domain for making packet forwarding decisions between different ports and a forwarding domain where the actual packet forwarding happens, based on the forwarding/routing tables. As there will be no longer control domain residing at an OpenFlow enabled switch/router, the forwarding domain can be kept simple to only contain certain rules for traffic flow processing. The functionality of the control domain is now moved to a network control server, which is also referred as an OpenFlow controller [OpenFlow]. It can intercept and manipulate the packet headers when the incoming traffic belong to the pre-set rules and output those packets to any port(s) as needed [McKeown2008]. It is one of this dissertation's contributions that a practical application of OpenFlow enabled wireless mesh network infrastructure for environmental monitoring purposes is demonstrated. Figure 5.9 shows the structure comparison between legacy and OpenFlow enabled routers.

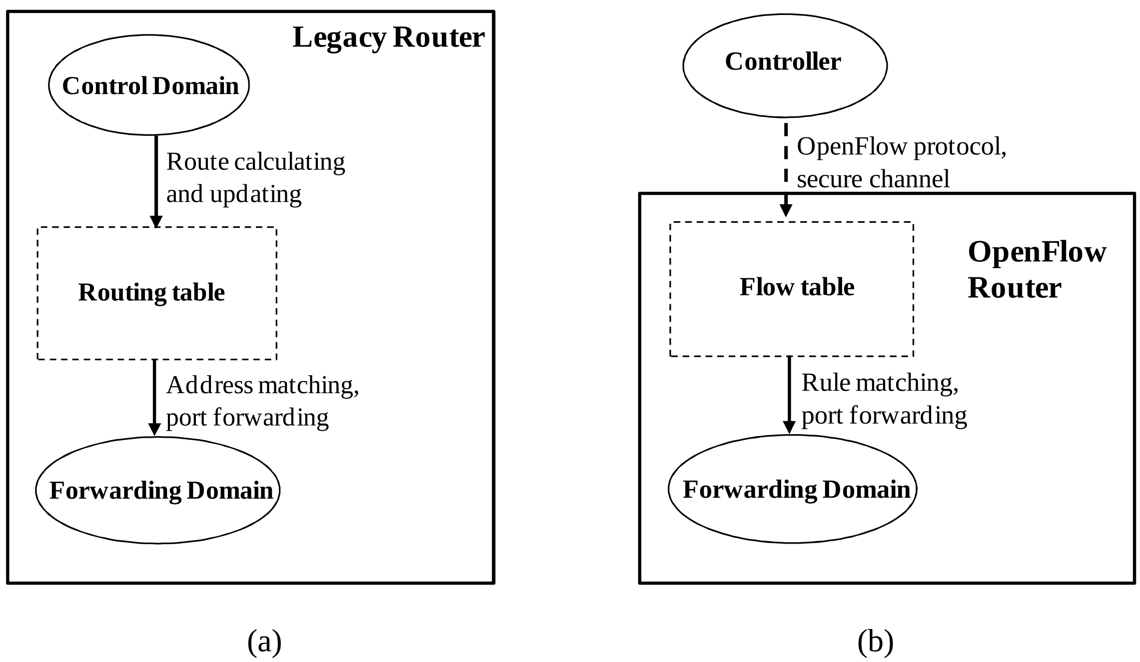

Figure 5.9: Structure comparison between (a) a legacy router and (b) an OpenFlow router.

5.4.3 OpenFlow Enabled Wireless Mesh Nodes

There are generally four components in a typical mesh based wireless infrastructure mesh clients, mesh nodes, Internet gateways, and remote servers. In this dissertation, each mesh node (or Internet gateway) is OpenFlow-enabled and responsible for mesh connectivity and traffic routing and is split into two virtual planes, a data plane and a control plane. The control plane no longer resides only on the switch, but is implemented on a remote server that runs a network-wide controller. The data plane is abstracted as a set of rules, composed of a flow table for flow-processing. Network virtualization is achieved by installing OpenFlow switch software at each device [OVS]. Data traffic relies on the OpenFlow data plane to reach remote servers through data interfaces.

A. In-band control network

A fundamental requirement of an OpenFlow deployment is that IP connectivity needs to be assured for the communication between the OpenFlow switches and controller, which runs over TCP (or SSL). Using the OpenFlow terminology, this control can happen "in-band", if the same network is used to transfer both data and OpenFlow control traffic, or "out-of-band" if different networks are used. Normally, the control and data interfaces of a single mesh node are two different physical interfaces. Control traffic is usually referred as out-of-band traffic because data traffic does not share the same path with control traffic. In this work, however, there is only one physical interface on each mesh node for multi-hop connectivity and traffic flow (the Internet gateway has two interfaces, one is for Internet connectivity only).

A solution to create an OpenFlow wireless mesh with out-of-band control was proposed in [Dely2011], using different SSIDs for the control and data network. But it relies on the capability of the wireless driver to support multiple SSIDs. Differently, in this paper an architecture which uses a single SSID, in-band control strategy is presented. A secure virtual private network (VPN) tunnel is created as the control path for each mesh node or Internet gateway. A VPN client thus resides on each mesh node for VPN connection. Sockets are used by data path components to communicate locally with control path components.

In aim to deploy an in-band control, the control-rules need to be locally setup to forward OpenFlow control packets, which are packets with destination IP address belonging to the control-subnet. To this aim, BATMAN routing protocol is utilized to learn the topology of the control-subnet and then exploit this knowledge to setup the control-rules. Accordingly, a BATMAN routing instance runs on each mesh node and the IP address of the controller is also advertised by BATMAN.

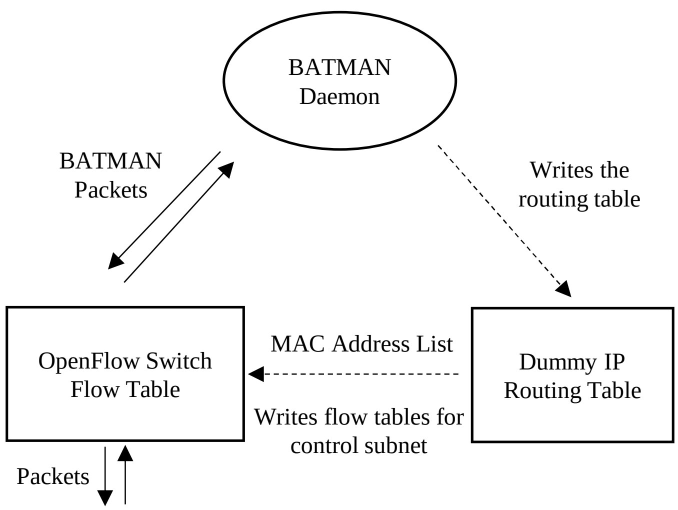

Figure 5.10: OpenFlow and BATMAN interaction

Figure 5.10 reports the main entities of a mesh node involved in the interplay between BATMAN and OpenFlow. The control rules used by OpenFlow message are configured by inspecting an IP routing table handled by the BATMAN daemon. This IP routing table configured is a “dummy” one, i.e. not actually used by the operating system when forwarding IP packets. In the Linux case this is a user defined routing table, different from the kernel main one, and never referenced in the Routing Policy Data Base.

An entry of the dummy routing table has the form <control subnet IP address/32, next-hop, output interface>; it is then converted in a rule of the OpenFlow table whose match is “IP destination == control subnet IP address” and whose action is “change source MAC address with the MAC address of outgoing interface and the destination MAC address with the MAC address of the next-hop”. Therefore BATMAN daemon needs to know the MAC addresses of the mesh nodes; this IP-to-MAC translation can be provided offline or can be distributed by a BATMAN plug-in, so that each mesh node can learn the MAC addresses of all other mesh nodes. To follow topology change, a timeout (e.g. 60s) is set to the inserted control-rules and at the timeout expiration the dummy IP table is dumped again on the OpenFlow Table. In addition to the control-rules used to route OpenFlow traffic, the flow tables are also filled with other control-rules needed to support the BATMAN operations. These rules are used to forward the incoming BATMAN packets to the BATMAN daemon in the mesh node and to let the outgoing BATMAN packets exit from the proper interfaces.

B. Data Network

The diagram of an OpenFlow-enabled wireless mesh node is shown in Fig. 5.11. It includes: one wireless interface belonging to the Wireless Mesh Network (wlan0); an optional wired interface towards sensor data access networks (eth0); an optional wired interface used as a gateway to the Internet (eth1); a virtual interface br0, which is a software bridge using OpenFlow switching logic, e.g. Open vSwitch. A generic “real” mesh node may have additional wireless or wired interfaces towards sensor data access network and additional wireless interfaces can be bridged to br0 if a multi-channel mesh nodes is used.

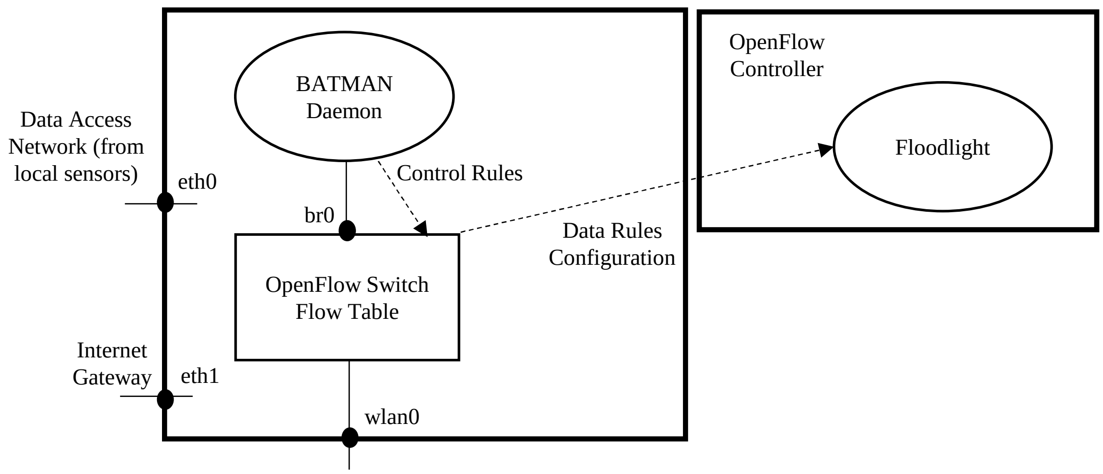

Figure 5.11: OpenFlow-enabled wireless mesh node

The br0 interface has an IP address belonging to the control-subnet, wlan0 does not have an IP address, eth0 has an address of the sensor data access networks subnet and eth1 of the subnet connected to the Internet. BATMAN daemon is connected to br0, and br0 is used as destination for any packets generated by the node and directed towards the mesh nodes. To this aim, a trick of inserting in the main routing table of Linux a fake IP address (e.g. 10.0.254.254) is used as gateway of all the routes whose outgoing interface is br0 (i.e. of the routes directed toward the WMN). To avoid ARP generation, a fake MAC address for the fake IP address is inserted in the ARP table.

The data packet will be received by a mesh node on its sensor data access network interface. Then a match is searched in the flow table. In case a match is found, the related action is carried out. Otherwise, the IP packet is embedded in an OpenFlow *packet-in*, which is transferred to the controller using the in-band control network. When the controller receives the *packet-in*, it applies the programmed routing logic, e.g. the one presented in this dissertation.

To support controller operations, the IP subnets of the sensor data access networks are advertised by mesh nodes and gateways by BATMAN. Moreover, gateways also advertise the default route 0.0.0.0/0. In doing so, each mesh node knows the full network topology and the controller can inquiry the connected mesh nodes to learn this information, which is fundamental to implement traffic engineering logic for data traffic

C. Wireless Mesh Network Architecture

As illustrated in Fig. 5.12, OpenFlow creates a secure channel between the data plane and network controller with a set of rules, including several properties, an expiration time and a list of actions. The properties specify packet source, original header values, and switch ports. When a packet arrives at the switch network interface but no matching rule is found, the packets are encapsulated and forwarded to the network controller. Otherwise, actions allow us to manipulate packet headers, outputing packets to a particular switch port based on the rules. The network controller runs applications that are aware of the flow a packet belongs to and the flow properties through a control plane path, while determining the route for packets through data plane path. In this work, BATMAN is implemented on top of the physical layer to allow node discovery and multi-hop connectivity between the virtual control interfaces. The centralized routing process is implemented on top of the IP layer with the help of the OpenFlow controller by redirecting traffic flows along the determined routes. Connectivity to the remote server is achieved using one or more Internet gateways. A network performance measurement client resides at each mesh node and Internet gateway providing information as needed.

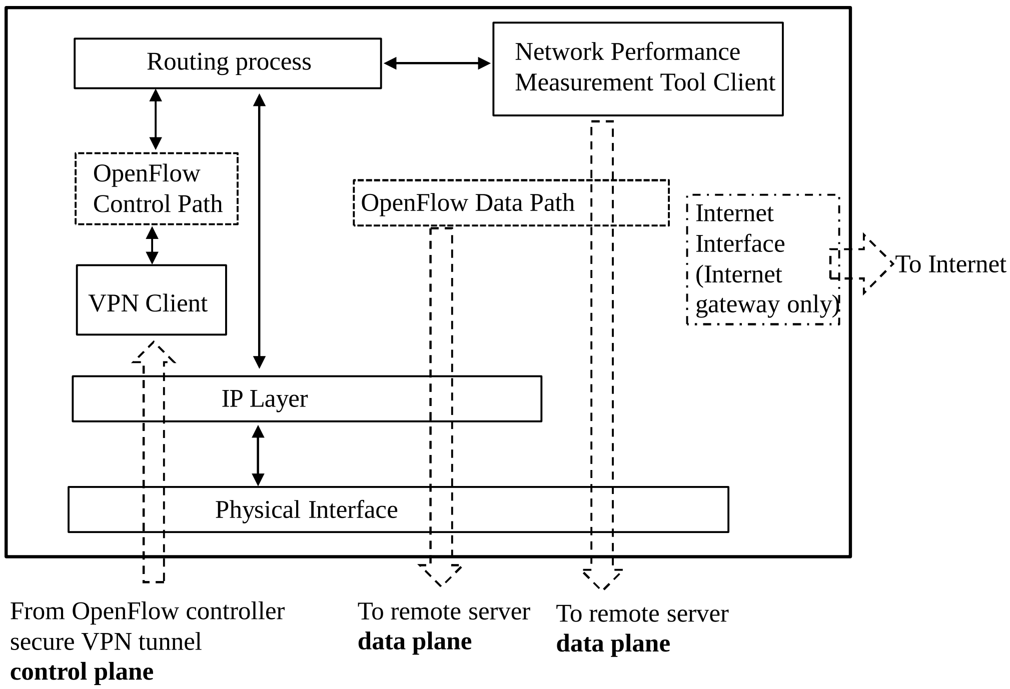

Figure 5.12: Architecture of an OpenFlow enabled mesh node (or an Internet gateway, with an additional Internet interface)

As shown in Fig. 5.13, the architecture of the remote server consists of a network performance measurement tool server, an OpenFlow controller, a VPN server, and a memory space for data reception. The performance server queries information from mesh nodes and Internet gateway(s) and builds a database used for calculating the routing process. The database also contains a network graph built from connectivity information provided by the discovery protocol. The main purpose of the OpenFlow controller is to perform basic load balancing tasks, such as redirecting traffic flows and managing network addresses. The controller has full access to the database, thus it will control traffic according to (soft) real-time routing information.

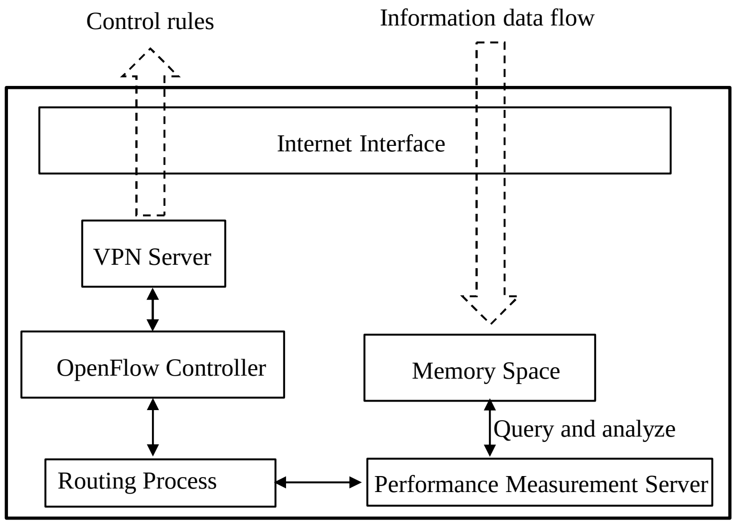

Figure 5.13: Architecture of the remote server

Scenarios where nodes fail and new nodes are added are also discussed in this dissertation: the OpenFlow controller is informed of network topology changes and it updates routes based on the new network graphs; when a new node joins the network, changes in the database trigger recalculation of the routing process, and new flow rules will be installed accordingly. How to technically implement data flow path setup and redirections is explained in detail at Appendix E.

5.5 Routing Performance Evaluation: Demo

In this section, experimental demonstration of routing performance is presented. The protocol is implemented in embedded X86 based devices with the aid of OpenFlow to utilize centralized control.

5.5.1 Experimental Setup

We built a testbed to show the capabilities of centralized routing using various hardware and software components. The testbed is comprised of a server, Internet gateways and OpenFlow-enabled mesh nodes. We used a Dell Optiplex server with Intel R CoreTM i7-3770 CPU and 16 GiB memory for calculating the routes, operating an OpenFlow network controller and the management server. We used Floodlight as the OpenFlow network controller. An embedded X86 device - PCEngine, was chosen to work as mesh nodes and Internet gateways, due to its extensible memory, radio adaptability via PCI cards, (relatively) low power consumption, and low cost. IEEE 802.11 standard based mini-PCI cards was used for communication between individual nodes. The mesh nodes were built on the Debian (6.0) operating system using the OpenFlow reference implementation for control and data paths. Internet gateways were connected to the server through Ethernet and their wireless PCI interface was bridged to create the subnet for communicating with mesh nodes. It is worth noting that route computations are performed at the server, this centralized protocol greatly decreases the computational burden on individual nodes.

The test network is composed of nine OpenFlow-enabled mesh nodes deployed in a multi-hop fashion. A mesh node can either produce data (from locally connected sensors) or receive data packets from other nodes when it acts as a relay node. There are two Internet gateways that relay data to the remote server, where sensor data from all the nodes is processed and stored. Node at a certain hop will relay data packet to its peer node at the next hop in the direction to the remote server. The test network topology is deployed in office as shown in Fig. 5.14.

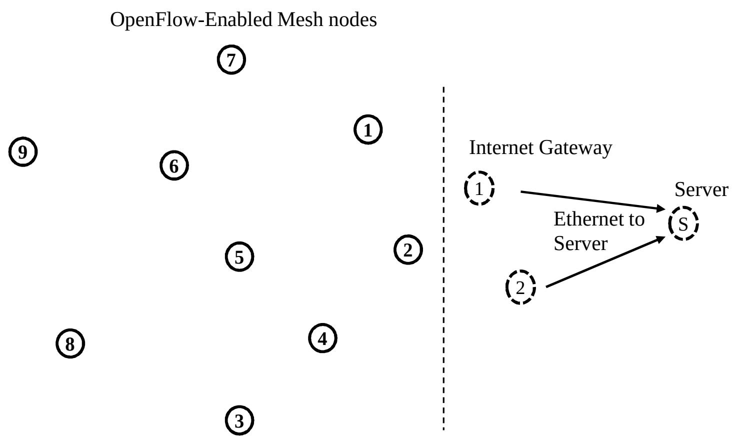

Figure 5.14: Test network topology

The deployment is implemented using the wireless infrastructure introduced in previous section. Transmit power and antenna gain are tuned carefully to simulate different distances of mesh nodes, such that each node can only reach mesh nodes within a certain transmit range. The resulting next-hop neighbors of each mesh node are listed in Table 5.4.

Table 5.4: List of next hop neighbors

Node ID	*Next Hop Neighbor Node ID*
1	Internet Gateways 1, 2, mesh node2
2	Internet Gateways 1, 2
3	Mesh nodes 4, 5

4	Internet Gateways 1, 2, mesh nodes 1, 2
5	Mesh nodes 1, 2, 3, 4
6	Mesh nodes 1, 2, 5, 7
7	Mesh nodes 1, 6, 9
8	Mesh nodes 3, 5, 6, 9
9	Mesh nodes 5, 6, 7, 8

5.5.2 Performance Evaluation

In this section, the control traffic overhead raised by OpenFlow is firstly investigated. Two important performance metrics are then evaluated to analyze the performance of proposed routing protocol: 1) accumulated data traffic at mesh nodes, and 2) network response time to topology changes.

A. Control Traffic Overhead

OpenFlow creates control traffic when new rules are installed, statistics are queried and through the heart beat signal that mesh nodes send to the network control server. The amount of control traffic that is created when installing rules at different rates at random nodes in the network depicted in Fig. 5.14. As traffic is relayed over multiple hops it is counted each time it is transmitted wirelessly. In Fig. 5.15, the OpenFlow control traffic rates is compared with the control traffic created by BATMAN, which provides the basic routing infrastructure. As expected, the OpenFlow control traffic increases as the rule installation rate increases, while BATMAN traffic stays constant. With 19 new rules per second, the additional control traffic introduced by OpenFlow is about 26 kbit/s and the total control traffic is about 5 times higher compared to a case where only BATMAN is used. However, compared to the achievable throughput, the control traffic is still low and for certain scenarios, much lower rule installation rates can be anticipated. Compared to a pure BATMAN network, OpenFlow adds some extra control traffic, but the amount is relatively small.

Figure 5.15: Aggregated traffic at mesh nodes

However, scalability is a major concern when using centralized schemes such as OpenFlow. As the network size increases, more heart beat signals are generated and potentially more rules need to be installed. The results from this small test network show that the amount of control traffic generated by each mesh router is in the order of a few kbit/s. Scalability issues should be investigated further in later work.

B. Accumulated Data Traffic at Mesh Nodes

Accumulated data traffic is calculated by counting the number of received data packets at each mesh node. Under this test scenario, mesh nodes 8 and 9 are source nodes that continuously send UDP data traffic bound for the remote server that is relayed by other mesh nodes (1-7) and Internet gateways (1-2). The packet size of UDP datagram is set at 200 Bytes and the UDP traffic bandwidth is set at 4 Mbps. The accumulated data traffic at each mesh node when centralized routing is used and not used is compared. Figure 5.16 shows the results.

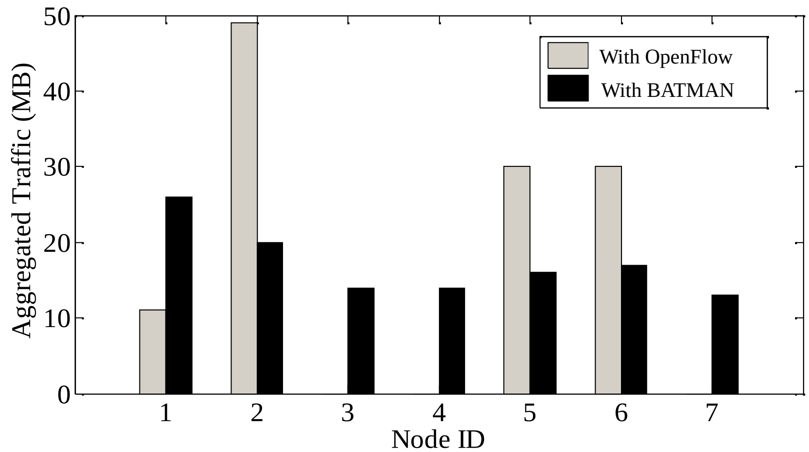

Figure 5.16: Aggregated traffic at mesh nodes

When centralized routing is not used, a mesh node selects its neighbor and routes by querying nodes within transmission range. Once the neighbor is identified, the traffic flow is unicast between mesh node and neighbor. As shown in Fig. 5.16, only mesh nodes 1, 2, 5 and 6 have data traffic, but nodes 3, 4 and 7 are idle. This is not an ideal solution for uplink data transit in wireless mesh networking.

When centralized routing is used, data traffic is balanced among all of the mesh nodes. It allows mesh nodes to transmit traffic flow in multiple paths. Therefore, node 9 splits its traffic to nodes 6 and 7 while node 8 splits to nodes 3 and 5. These neighbor nodes split their relay traffic in the same way to reach the Internet gateways.

C. Network Response Time to Topology Changes

Network response time is defined as the duration from the time when there is a change in network topology (e.g. a new node joins the network or an existing node fails) to the time when the server has set up the best routes for the updated topology. When a new node joins the network in multi-hop wireless network, it sends broadcast packets to locate its immediate neighbors at the next hop (neighbor discover). Nodes that receive the broadcast beacon respond with an acknowledgement packet indicating that a neighbor has been found. In a traditional mesh network, the new node needs to evaluate responses of all its neighbors to identify the best route (route discover). In an OpenFlow-based mesh network, the new nodes location and path quality information are sent to the server by its nearest neighbor using the existing route (server notification). The controller residing at the server is able to recalculate the best routes based on the routing process (route setup). When an existing node fails and disappears from the topology, mesh nodes in a traditional network must check with their next hop peers to form best routes, while the controller in OpenFlow based network notices nodes failure instantly.

The centralized control methodology provides short response times and robust network performance, beneficial attributes for many wireless mesh applications. The performance of network response time is also evaluated under different testing scenarios: when a new node joins the network, and when an existing node fails and leaves the network. For each scenario, network response time is calculated with OpenFlow and BATMAN. The testing scenarios and how network responds at each scenario are summarized in Table 5.5. For example, when a new node joins a network with BATMAN, the network identifies the new node by going through “neighbor discover”, “route discover”, and “server notification” events. However, if a new node joins a network with OpenFlow, it goes through “neighbor discover”, “server notification”, and “route setup” events. As “route discover” takes much longer time than “route setup”, a much less network response time is expected. The same methodology applies to the scenarios when an existing node fails.

Table 5.5: Network response time calculation under different testing scenarios

Testing Scenarios		*Response Time Calculation (Events)*
New node joins	With BATMANM	Neighbor discover, route discover, server notification
	With OpenFlow	Neighbor discover, server notification, route setup
Existing node fails	With BATMANM	Neighbor discover, route discover, server notification
	With OpenFlow	Server notification, route setup

Moreover, each scenario is tested at different number of mesh nodes in the network. The corresponding network configuration is listed in Table 5.6.

Table 5.6: Network configuration with different number of mesh nodes

Number of Nodes in Network	*Network Topology Change*	
Two nodes in network: nodes 1~2	Node 5 joins network	Node 1 fails and leaves network
Four nodes in network: nodes 1~4		
Six nodes in network: nodes 1~6	Node 9 joins network	
Eight nodes in network: nodes 1~8		

Each mesh node reports its status every second. The transmit packet interval is also set as one second for the sake of easy implementation. Network response time is measured by checking event timestamps. It can be clearly seen in the evaluation results in

Fig. 5.17 that network responds to topology changes much faster when using OpenFlow. In particular, the average response time decrease is 48.2% when a new node joins the network and 82.7% when an existing node fails and leaves. Because the server instantly notices a nodes failure, it removes the time for node neighbor discovery events, resulting the lowest network response time. It's worth mentioning that network response time increases as the number of mesh nodes increases at all scenarios. This phenomenon is as expected when there are more nodes in the network because: 1) if a new node joins, it needs to go through a greater number of hops to notify the server, and 2) if a node fails, it takes longer for mesh nodes to relocate to their new neighbors and notify the server. Using centralized routing saves time on relocating new neighbors and setting up new routes, thus demonstrating its efficiency at managing topology changes in wireless mesh networking.

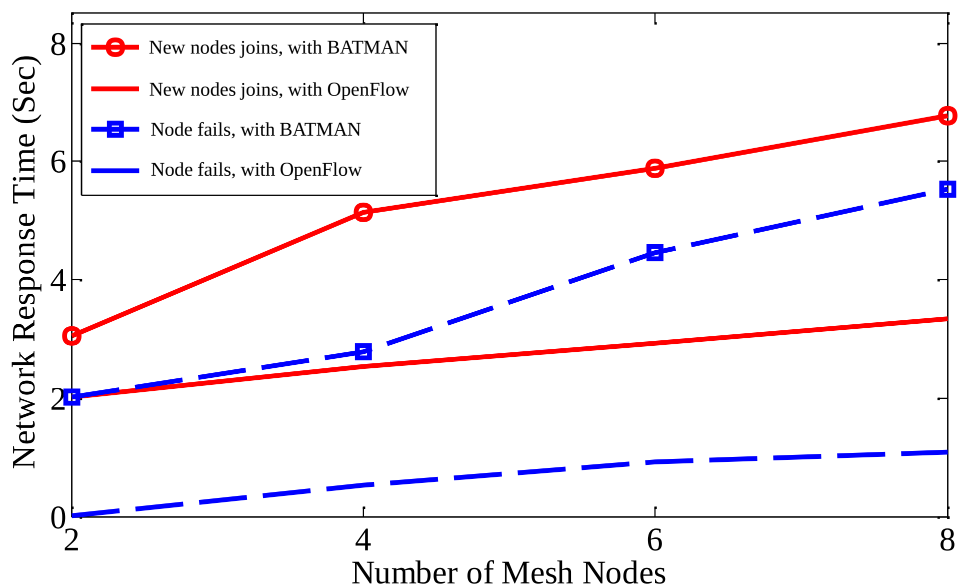

Figure 5.17: Network response time under different testing scenario

5.6 Summary

The multi-tier network architecture based on wireless mesh nodes presented in this chapter provides a base platform that can be adapted to various deployment environments. However, traditional mesh networks and routing protocols have limitations in managing network topology changes and maintaining network utilization. A simple routing policy based on central controller is studied in this chapter to address the challenges. The principle, architecture, simulation, and technical implementation of the centralized routing algorithm are explained in detail. The architecture and protocol design presented in this chapter can serve as a useful and pragmatic wireless monitoring framework with centralized control. It can foster innovation in advanced routing and traffic engineering, and more in general, the optimization of the use of limited communication resources of wireless mesh networks.

CHAPTER SIX

CONTRIBUTIONS AND FUTURE WORK

6.1 Contributions

As an attempt to help understanding wireless sensor networks from aspects of communication reliability, network architecture and operations with low and dynamic power sources, this dissertation offers approaches of evaluating/modeling performance of reliable and sustainable wireless sensor data transmissions for various WSN applications and provides solutions for sustainable wireless sensor network operations based on low power energy harvesting devices. The main contributions of the dissertation are summarized as follows.

6.1.1 Communication Reliability

A simple framework design for modeling wireless data transmission in deterministic channels is presented in this section. Mathematical models to capture the key characteristics of specific wireless channels based on surface roughness parameters are introduced. Two transmission error control methods: ARQ and error avoidance for wireless sensors in rotating environments are then studied. Models for constant packet rate and saturation throughputs and energy efficiency calculation are derived. In order to validate models, experiments were carried out to measure performance under different schemes: ARQ, partial error avoidance and complete error avoidance. Important results are summarized as follows:

- The use of ARQ provides a way of implementing reliable transmission: packets that are not acknowledged are retransmitted immediately after timeout. However, there will be energy wasted on retransmissions. Error avoidance algorithm implemented in this dissertation can greatly decrease retransmission counts, as it avoids packet transmission if it is traversing a high-error region.
- Throughput and energy efficiency calculated from the model match well with experimental measurements.
- Wireless channel's spatial properties can be characterized by surface roughness. It is shown that under different packet error rate profiles, using error avoidance and ARQ have different performance improvements. The preference of selecting reliability control methods can be guided by the surface roughness parameters extracted from packet error rate profiles.

6.1.2 Operation with Low and Dynamic Power Sources

Based on the systematic study of MFC and PMS output properties as shown in Appendix A, this part of dissertation presents the design and performance evaluation through simulations of a communication protocol for wireless sensor networks with dynamic and limited energy harvesting source—MFC to be more specific. The protocol includes the following key properties.

- Firstly, a duty-cycle based energy neutral transmission sleep/wakeup policy between peer nodes is proposed to guarantee packet reception.
- Secondly, an energy neutral joint scheduling and routing protocol in multi-hop MFC networks is introduced, aiming to provide network-wide packet

communications under extremely limited node resource such as ultra-low power supply.

- Finally, a simulation tool is developed to evaluate the protocol performance. Simulation results show that the proposed communication protocol is able to provide sustainable and reliable data transmission with low and dynamic power input.

6.1.3 Multi-tier Network Architecture

Due to the complexity of remote environmental monitoring, there are many factors to consider when deploying a sensor network. The wireless mesh node based multi-tier network architecture presented in this dissertation provides a base platform where new routing logics can be easily added for different application objectives. Key contributions are summarized as follows:

- A new network architecture that allows for the flexible and efficient use of OpenFlow in wireless mesh network is presented. By integrating the centralized routing protocol, the architecture allows for an assessment of the link quality and self-configuration for better resource distribution fairness in wireless mesh networks.
- To demonstrate the usefulness of the proposed approach, simulation tools were developed to compare performance and realistic test cases were implemented to enable data path setup and redirecting. The presented platform serves as a useful and pragmatic wireless monitoring framework with centralized control.

6.2 Future Work

The knowledge derived from study of the systems, models and protocols in this work provides a useful framework for designing future WSNs. There are a number of promising research areas that could be pursued to address the limitations of this dissertation and extend it further. A discussion of the directions in which current dissertation can be extended is listed in this section.

6.2.1 Accurate Event Detection

The presented protocol in this dissertation aiming to achieve communication reliability ignores the importance of an accurate event identification mechanism by assuming that the event of interest has been accurately identified by the sensor node(s). Thus, one of the major challenges for reliability protocols is an accurate event identification mechanism, which is an essential characteristic of an event driven wireless sensor network. Once the events are accurately identified, then only a single packet carrying information about the event would be enough for the sink instead of receiving all the packets from the involved sensor nodes. This approach requires the sensor nodes to suppress the transmission of packets that carry duplicate information about an event, typically those that have been generated by sensors in the same sensing region, by exploiting spatio-temporal correlation among the closely located sensor nodes.

6.2.2 Adaptive Transmit Power Control

An adaptive power control and management mechanism can significantly save the power consumption of transmitters and extend life time. In many sensor radio applications on manufacturing machines, the physical distance between the transmitter and the receiver is not necessarily large. This type of radio placement ensures, at the majority of the transmitter locations on the rotation circular route, the radio channel enjoys low power attenuation with occasional high power attenuation occurrences. Given the precise information of the time-varying power attenuation, the transmitter is able to intelligently adjust the transmit power to achieve an optimum tradeoff between power saving and transmission success.

6.2.3 Protocol Development for MFC Network with Wireless Errors

In the communication protocol design with MFC WSNs, if a data packet is lost, the lost data packet will have to be retransmitted in a later cycle, which will result in lower throughput and longer node-to-sink delay. The numerical evaluation in the dissertation was carried out without assuming channel-induced errors to provide an upper bound performance assessment for such systems. As future work, the impact of transmission errors on the performance can be taken into consideration to provide more accurate performance modeling and communication scheduling designs.

6.2.4 Mesh Infrastructure for Dense Area Monitoring

In the future, the infrastructure studied in this dissertation is expected to be deployed for large and dense areas to generate more performance test results. The monitoring stations will be deployed in three different locations, covering 5 to 10 square miles. The infrastructure consists of a dedicated backhaul wireless hybrid network, using Wi-Fi based traditional and mesh network configurations which will operate at radios frequencies of 2.4 GHz and 900 MHz. The deployed wireless networks will be connected to the Internet using public and private networks. Moreover, to control and monitor hundreds of deployed nodes, utilizing OpenFlow more efficiently to manage deployed network and load balancing of the data generated from the sensor networks will be another interesting to pursue.

6.2.5 Improving OpenFlow-Enabled Wireless Network

As indicated in the dissertation, there are still several limitations of using OpenFlow in wireless networks. For example, if in-band control path is utilized, control messages could be lost due to the unreliability of wireless links. How to prevent in-band control paths from being interrupted by wireless link errors requires further study. Control overhead and scalability is another issue in OpenFlow-enabled wireless networks. When the network scale increases, the amount of control messages significantly increases. It may be a serious problem if the wireless link bandwidth is limited. Adding additional radios is a possible solution, but how to solve the problem in a cost-effective way remains an open challenge.

6.2.6 Security and Privacy Considerations

The deployments of WSNs for monitoring purposes present an easy target for various security and privacy attacks. The existing communication protocols and platforms are confined only to the reliable transmission of sensor data information. They ignore the cases of information manipulation (e.g., injecting false information) or corruption of sensor nodes for collection of unauthorized information (e.g., misrouting packets). It is thus desirable to consider the reliability of the collected information by addressing the security and privacy risks in WSNs.

APPENDICES

Appendix A

Microbial Fuel Cell and Power Management System

In this section, an experimental MFC testbed and a transformer-based PMS are designed and implemented. Optimal capacitor value is selected so that PMS output is maximized given the MFC's charging properties. MFC and PMS output performance is systematically evaluated to provide fundamental basis for the sustainable communication protocol design.

A.1 Experimental MFC and PMS Setup

MFC Testbed Setup

A single-chamber air-cathode MFC reactor was constructed as shown in Figure A.1, and the reactor had an anodic chamber volume of 316 ml. The anode was made of 0.381 mm thick carbon cloth (CCP30CM, Fuel Cell Earth, Stoneham, MA, USA). The cathode was made of 0.28 mm thick carbon paper (TGP-H-090, Fuel Cell Earth, Stoneham, MA , USA), which was covered with Pt-catalyst (XC-72, Fuel Cell Store, Boulder, CO, USA). Each of the electrodes, apart by 2.5 cm and separated by a 0.18 mm thick Nafion membrane (N117, DuPont, Wilmington, DE, USA), had a diameter of 12.7 cm. The Nafion membrane was sequentially soaked in H_2O_2 (3%) at 50°C, deionized water at 80°C, 0.5M H_2SO_4 at 50°C, and deionized water at 80°C (each for 1 hour) immediately prior to the experiment. A piece of stainless steel mesh was used to prevent the deformation of the cathode.

Domestic wastewater bacteria have been shown to be suitable biocatalysts for MFC-based electricity production [Park2003]. Anaerobic domestic wastewater

(Greenville Wastewater Treatment Plant, Greenville, SC, USA) was used as both the inoculum and the substrate during the inoculation process. Additional nutrient medium was prepared to mix with the anaerobic wastewater at the ratio of 1.5:1. The deionized water-based medium, around PH 7, was prepared as follows (per liter): 310 mg NH_4Cl, 130 mg KCl, 2,690 mg $NaH_2PO_4{\cdot}H_2O$, 4,330 mg Na_2HPO_4, 1,510 mg NaCl, 0.15 mg $FeCl_2{\cdot}4H_2O$, 0.006 mg H_3BO_3, 0.009 mg $MnSO_4{\cdot}H_2O$, 0.012 mg $Co(NO_3)_2{\cdot}6H_2O$, 0.0132 mg $ZnSO_4{\cdot}7H_2O$, 0.0025 mg $NiCl_2{\cdot}6H_2O$, 0.0014 mg $CuCl_2$, 0.0025 mg $Na_2MoO_4{\cdot}2H_2O$, 0.00088 ml 37% HCl and 4 ml sodium acetate (Sigma-Aldrich, St. Louis, MO, USA).

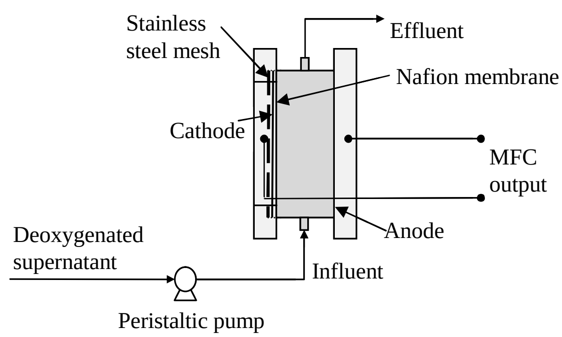

Figure A.1: Single-chamber air-cathode MFC reactor

During inoculation, the MFC reactor was first filled with the anaerobic domestic wastewater and nutrient medium solution and was operated under the batch mode. With an external 1 kΩ resistor connected, the MFC output voltage increased from zero and surpassed 0.605 V, and a rapid current increase was observed. The inoculation process was considered finished after the output voltage had stayed around 0.605 V for five days.

After inoculation, the inoculation solution was replaced with deoxygenated wastewater supernatant (Clemson Wastewater Treatment Plant, Clemson, SC, USA), 10 mM sodium acetate, and 30 mM NaCl. Acetate was added to avoid the substrate limitation on current in that the study of such limitations is beyond the scope of this study. The resulting supernatant-based substrate was approximately PH 6.4. This substrate was continuously instilled into the MFC reactor using a peristaltic pump (Cole-Parmer, Chicago, IL, USA) at 0.4 ml/min (equivalent a hydraulic retention time (HRT) of 13.2 h in this study). The entire experiment was performed with a 1 kΩ resistor connected under room temperature.

Transformer-Based Power Management System for MFC

Different PMS' for MFC power harvesting have been studied for intermittent load driving applications such as intermittent wireless transmission of sensed information. The simplest PMS can be implemented by directly connecting a super-capacitor to an MFC to increase and control the electrical potential generated by an MFC [Shantaram2005]; with this setup, however, the highest achievable output voltage is equal to the MFC output voltage, meaning that only very little energy can be stored and utilized per cycle. A simple improvement can be made by adding a charge pump before the super-capacitor and a DC/DC converter after the super-capacitor, such that the load is driven only when a predefined threshold voltage is reached [Meehan2011]. Such a PMS is good for MFCs'low-voltage outputs, but it only works if the MFC output voltage is high enough to drive the charge pump (e.g., 0.3 V). Its charging/discharging cycle is also much longer since the charging speed is limited by the charge pump.

To lower the input voltage requirement without sacrificing the charging speed, this study proposes to use a transformer to replace the charge pump as shown in Figure

A.2. This capacitor-transformer-converter PMS circuit utilizes two super-capacitors: one is connected to the MFC, and the other is placed after the DC/DC converter. The first super-capacitor (C_1), which is to be optimized, accumulates energy from the MFC and drives the following DC/DC converter. The second super-capacitor (C_2), whose capacitance must be determined based on the target load, stores power from the DC/DC converter and drives the load in bursts with high voltage and current as demanded by different applications.

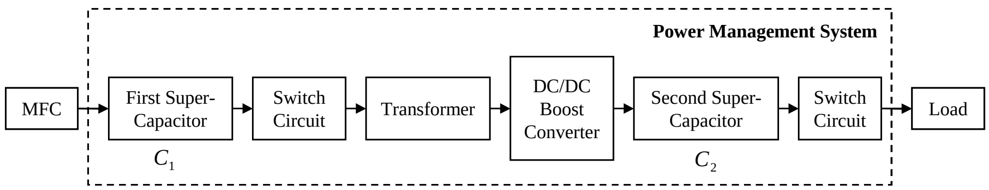

Figure A.2: Transformer-based power management system

PMS Circuit Implementation

The detailed design of the proposed PMS circuit is shown in Figure A.3. The voltage step-up converter used was LTC3108 (Linear Technology, Milpitas, CA, USA) [LTC3108]. The first super-capacitor (C_1) is first charged by the low-voltage output of the MFC, and Switch 1 is utilized to prevent the transformer and LTC3108 from drawing current from the first super-capacitor while it is being charged. Switch 1 is comprised of a Germanium transistor and an ALD110800 MOSFET (Advanced Linear Device, Sunnyvale, CA). The selected MOSFET has a typical zero gate threshold voltage.

The discharging voltage V_d is the rising threshold voltage of the Germanium transistor, and the charging voltage V_c is the falling threshold voltage of the transistor.

When the voltage of the super-capacitor reaches the turn-on voltage of the switch (0.41 V, discharging voltage V_d), the transistor functions and a voltage difference is applied at MOSFET, which makes MOSFET work and Switch 1 close. After Switch 1 closes, the first super-capacitor works as the energy source to power the rest of the PMS system and load. The voltage of the first super-capacitor is then amplified by the transformer (LPR6235-752SML, Coilcraft, Cary, IL, USA). The amplified voltage is then rectified by an internal rectifier circuit. While the first super-capacitor discharges, the second super-capacitor (C_2) then begins the charging cycle.

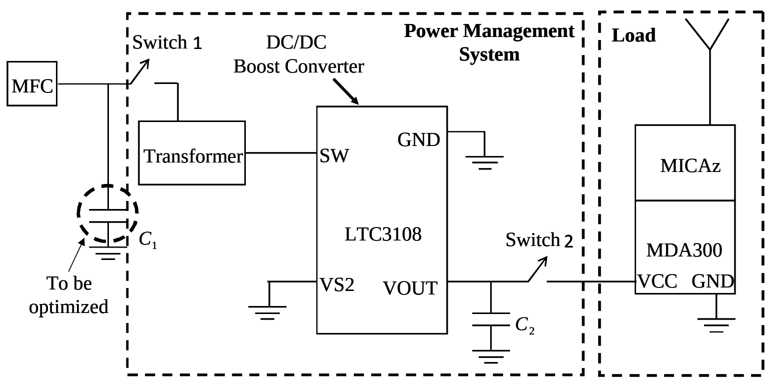

Figure A.3: PMS detailed circuit implementation

Switch 1 remains closed until the input voltage drops to V_c (0.18 V), which makes the transistor completely shut. Once Switch 1 is open, the first super-capacitor begins being charged again. This process iterates until the voltage of the second super-capacitor reaches a target voltage specified by the application (such as 3.3 V for a typical wireless sensor device). Then Switch 2 closes to power the application load. Switch 2 is used to prevent the load from drawing current from the second super-capacitor before it reaches the target voltage.

To demonstrate the feasibility of the proposed transformer-based PMS, the energy harvested from the MFC was used to power the MDA300 sensor board and the MICAz wireless sensor node (Moog Crossbow, Milpitas, CA, USA) [MDA300; MICAz] using the PMS as shown in Figure A.3. Embedded with temperature and humidity sensors, MDA300 can measure environmental data such as temperature and humidity. MICAz is used for wireless transmission of the data collected by MDA300. The total power consumption is approximately 65 mW for MICAz transmission at 0 dBm transmit power and 30 mW for MDA300 sensing. The proposed PMS stores the MFC harvested energy to achieve an intermittent 3.3 V output sufficient for driving the wireless sensor node. To support transmission of three 90-bytes packets in each charging cycle, the second super-capacitor was selected as 0.05 F. It should be noted that the capacitance of the second super-capacitor is dependent on the power need of loads. An LTC3108 converter with a peripheral circuit as described in the LTC3108 datasheet [LTC3108] was utilized in the test setup.

A.2 Identification of Optimal Capacitance in PMS

Capacitor Optimization

In order to identify the optimal capacitance for the first super-capacitor, an MFC is modeled using an equivalent circuit as shown in Figure A.4. Numerous equivalent circuit models have been developed in prior studies to model the MFC behavior when an MFC is connected with an external load. The fuel cell internal resistance can usually be modeled as a combination of anodic resistance, cathodic resistance, and ohmic resistance [Liang2007], while the anodic and cathodic resistances may be current-dependent. There is a capacitance at the interface between the electrode and its surrounding electrolyte/substrate when the charges in the electrode are separated from those in the electrolyte [Larminie2003]. All these effects should be included to model an MFC electrically. Herein, an MFC is modeled as an ohmic resistor in series with a parallel combination of the electrode capacitor and the associated charge transfer resistors, which are related to the activation and concentration energy of the anode and cathode [Wang2005]. In Figure A.4, R_{act}, R_{con} and R_{ohmic} account for the equivalent resistance of the MFC activation loss, concentration loss, and ohmic loss, respectively, E_{OCV} is the MFC open circuit voltage, C_e represents the electrode capacitor, and R_{line} is the resistance of the peripheral circuitry.

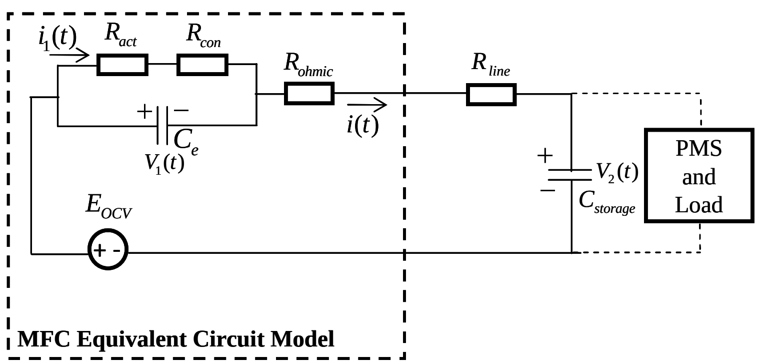

Figure A.4: Equivalent circuit model for MFC and circuit schematic during charging

As shown in the Appendix B, the MFC polarization curve (voltage-current curve) can be described using an equivalent circuit:

$$V_{cell} = E_{OCV} - IR_{ohmic} - aT - bT\ln(I) + c\ln(1 - \frac{I}{I_{\text{limit}}}) \tag{A.1}$$

where I_{limit} is the MFC limiting current (also known as the maximum MFC current), T is the absolute temperature, and I is the current flowing through R_{ohmic}.

Since commercial DC/DC converters are commonly used to build a PMS, it is not convenient to modify the parameters of commercial converters for PMS performance optimization. Under such circumstances, only super-capacitors are considered reconfigurable for optimized average power generation. Using the equivalent circuit shown in Figure A.4, the following equations can be established based on the Kirchhoff and Ohmic laws:

$$\begin{cases} C_e \dfrac{dV_1(t)}{dt} = i(t) - i_1(t) \\ E_{OCV} - V_1(t) - i(t)R_{ohmic} = i(t)R_{line} + V_2(t) \\ C_{storage} \dfrac{dV_2(t)}{dt} = i(t) \\ i_1(t) = \dfrac{V_1(t)}{R_{act} + R_{con}} \end{cases} \tag{A.2}$$

where $V_1(t)$ is the voltage of the electrode capacitor, $V_2(t)$ is the voltage of the super-capacitor ($C_{storage}$, corresponding to C_1 of Figure A.3), $R_{act} = T\left[a + b\ln(i_1(t))\right]/i_1(t)$, $R_{con} = -c\ln\left(1 - \dfrac{i_1(t)}{I_{\lim it}}\right)/i_1(t)$, and a, b, c, I_{limit} and R_{ohmic} are five unknown parameters to be experimentally determined. R_{act} and R_{con} can be estimated based on the MFC steady-state analysis. The initial values of $V_1(t)$ and $V_2(t)$ are both zero before the super-capacitor starts drawing energy from an MFC. When $V_2(t)$ reaches the discharging voltage V_d, the super-capacitor begins to discharge and drive the PMS and load until $V_2(t)$ drops to the charging voltage V_c for charging to begin again. During the steady state, the voltage on the super-capacitor $V_2(t)$ varies between V_d and V_c, and the average power stored in this voltage range is of interest in this work.

By solving Equation (A.2), the charging time (T_c) for the super-capacitor to be charged from V_c to V_d can be obtained, with the average power input to the PMS and load being a function of $C_{storage}$:

$$P_a = \frac{1}{2T_c} C_{storage} (V_d^2 - V_c^2) \tag{A.3}$$

and the optimal capacitance is determined by maximizing P_a.

Identification of MFC Equivalent Circuit and Optimal Capacitance

Once the MFC reactor reached the steady state after three days under a continuous mode, the 1 kΩ resistor was disconnected to measure the MFC open circuit voltage (OCV), which was around 0.79 V in this study. Then a series of resistors were connected to the MFC reactor with the resistance progressively changed from 10 kΩ to 50 Ω every 10 minutes to obtain the MFC polarization curve. Ten minutes were sufficient for the MFC reactor to reach the steady state after each resistance change in this study. The resulting voltage was continuously recorded using a multimeter (34410A, Agilent, Santa Clara, CA, USA) during this process, and the associated current was calculated by dividing the recorded voltage by the external resistance.

For the MFC polarization curve, represented by Equation (A.1), E_{OCV} was measured once the MFC reactor was functional, the absolute temperature was 297 K, and the five unknown parameters were estimated by curve fitting based on Equation (A.1). The parameter estimates are: $R_{ohmic} = 221.52$ Ω, $I_{\text{limit}} = 0.002$ A, $a = 3.6e^{-4}$, $b = 2.7e^{-5}$, and $c = 0.3568$. The identified R_{ohmic} is close to that of a similar MFC study [Liu2005], which was found to be 218 Ω. The value of C_e can be found by fitting the equivalent model to the Nyquist plot of the dynamic impedance measurement or the MFC voltage response during the transient period after the step-down change of current [Ha2010] in addition to the cyclic voltammetry [Logan2006]. For convenience, the capacitance of the electrode capacitor C_e was estimated using the transient voltage response method proposed in [Ha2010]. During the estimation process, a 5 kΩ external resistor was first connected to the MFC until the steady state was reached. The voltage was recorded as the initial voltage, and the 5 kΩ resistor was replaced with a 10 kΩ resistor. The voltage of the external resistor increased (compared with the initial voltage) and finally reached the steady state about 20 minutes later. The resulting transient voltage responses were recorded accordingly, and the voltage difference curve was fitted with the function proposed in [Ha2010]. The identified capacitance (C_e) is 0.0042 *F*. It should be noted that the curve fitting method is not a reliable method to accurately identify each single unknown parameter while it does work in identifying MFC parameters [Ha2010]. For accurate determination of electrochemical parameters, traditional electrochemical techniques, such as the current interrupt test and the cyclic voltammetry [Logan2006], should be adopted.

The optimal capacitance of the first super-capacitor was found as 1.5 F by maximizing P_a based on the equivalent MFC parameters and Equation (A.3). Further experiments were conducted to validate this estimated optimal capacitance. The capacitance of the first super-capacitor was changed from 0.1 F to 6 F (at 0.1, 0.47, 0.67, 0.87, 1, 1.5, 2, 3, 4, 5 and 6 F). At each value, the capacitor was connected to the MFC directly, and then the charging process began. In the steady state, when the voltage of the capacitor (V_2) was raised from the charging voltage (V_c) to the discharging voltage (V_d), the charging time T_c was recorded. The average power P_a harvested during a single charging cycle was then calculated based on Equation (A.3). The same procedure was repeated three times for each capacitance. The experimental results are shown in Figure A.5, and the optimal capacitance is the one that results in the maximum average power. From Figure A.5, the power estimation curve matches the measurement results very well. Both the analytical and experimental studies have found that the 1.5 F super-capacitor works the best for maximum average power harvesting for the MFC reactor in this study.

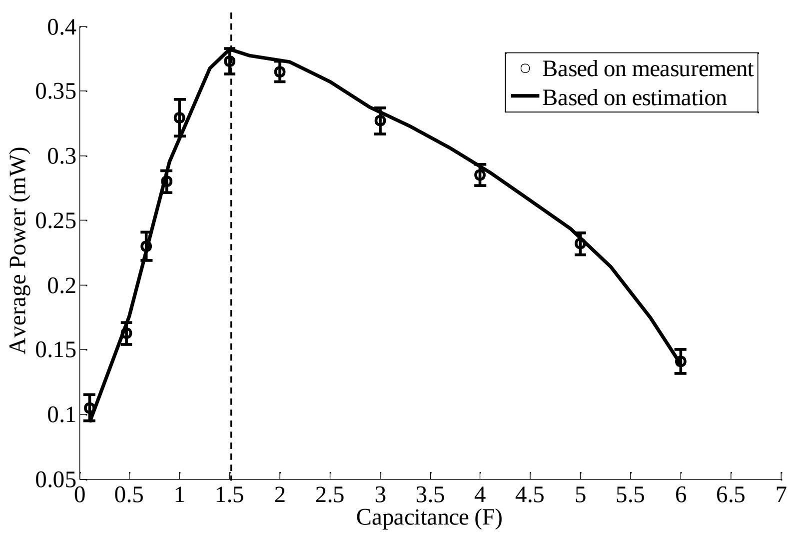

Figure A.5: Measured and estimated average stored power at different capacitance

A.3 MFC and PMS Output Performance Evaluation

Output Voltage and Power

During the experiment, the charging and discharging voltage history was studied based on the wireless sensing setup. When the first super-capacitor was charged to the discharging voltage, Switch 1 closed and the PMS began to charge the second super-capacitor. After the voltage of the first super-capacitor dropped to the charging voltage, it was charged by the MFC again. This charging and discharging process was repeated until the second 0.05 F capacitor was charged from 0V to 3.3V, which was equivalent to about 2.28 hours. After that Switch 2 closed and the power accumulated in the second super-capacitor was discharged to drive the load. This discharging process continued until the voltage on the second capacitor dropped below 2.5 V, under which the load (MICAz with MDA300) was no longer working. The minimum input voltage for MICAz with MDA300 sensor board shown in [MICAz] is 2.7 V, however, 2.5 V was the minimal input voltage value that was measured during the experimental tests herein. After the load was no longer working, the second super-capacitor was again charged until its voltage reached 3.3V. Figure A.6 shows the voltages on the capacitors and output of the PMS in the first three cycles. It took about 0.99 hour to charge the second capacitor from 2.5 V to 3.3 V and this charging period is called the steady state period, which is repeated as charging/discharging cycles. As the initial voltage on the second super-capacitor is 0 V, the period when its voltage increases from 0 V to 2.5 V is referred as the initiation period.

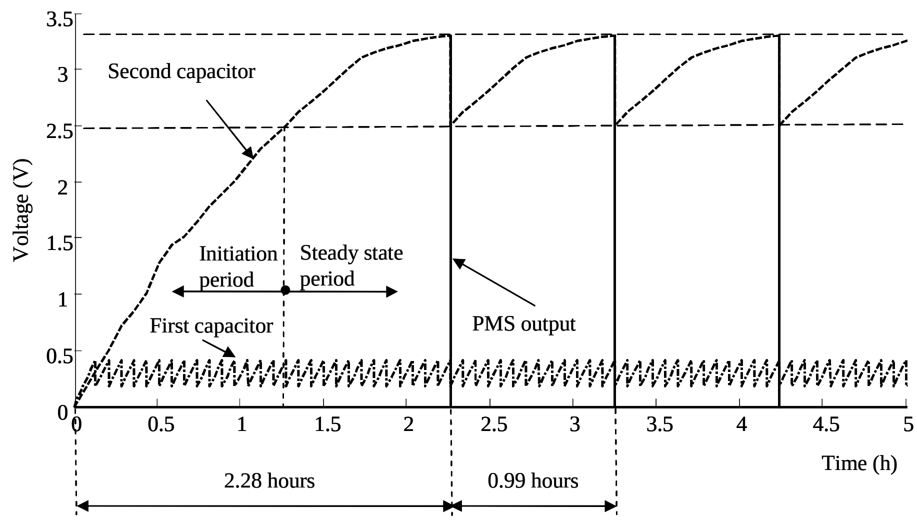

Figure A.6: Voltage on the capacitors and output of PMS

In addition, the power stored in the capacitors and delivered in the first three cycles were also measured and shown in Figure A.7 based on the power information. The first super-capacitor drew less than 0.4 mW from the MFC while the wireless transmitter and sensor board drew up to 95mW power from the PMS when Switch 2 was closed.

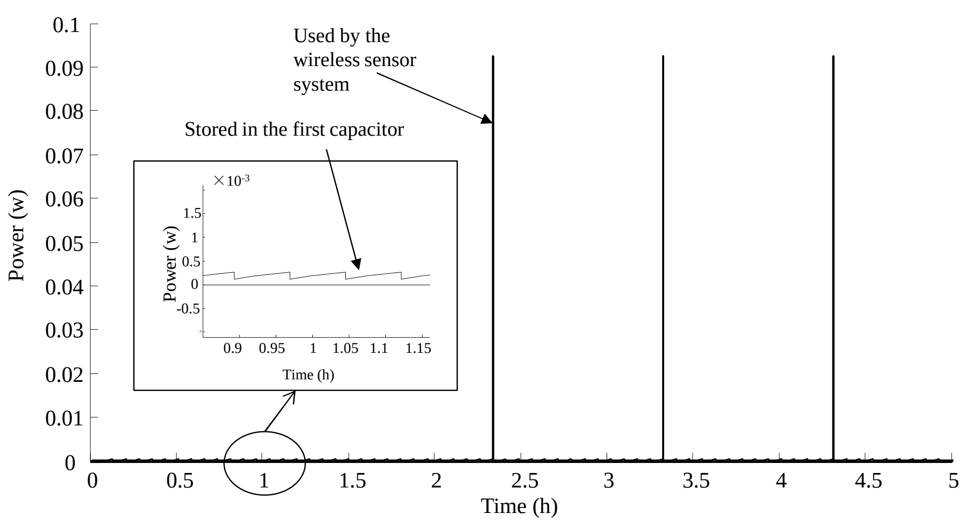

Figure A.7: Power stored in the capacitors and delivered to the loads

Wireless Sensing Application

Wireless transmission of temperature and humidity information using the PMS-assisted MFC was studied as an experimental application. As aforementioned, MICAz and MDA300 were attached together so that the temperature and humidity data sampled by MDA300's internal sensors can be included in MICAz's transmitted packets. When the voltage of the second capacitor reached 3.3 V and thus Switch 2 closed, MICAz was able to start up and transmit three packets. The reason to only transmit three packets is that for typical wireless sensor network applications, at least 2 packets are required for notification and handshake between neighbor sensors and one packet is needed to transmit information data. As shown in Figure A.6, it took about more than two hours for the second super-capacitor to reach 3.3 V if its initial voltage was 0 V. However, the time interval during the steady-state charging/discharging cycles was about 0.99 hour as the voltage of the second super-capacitor decreased to 2.5 V instead of 0 V, which greatly shortened the charging time of the second super-capacitor.

Three representative cycles of wireless humidity and temperature measurements at the Clemson Advanced Manufacturing & Systems Integration Laboratory (Clemson, SC, USA) are shown in Figure A.8. One humidity data and one temperature data were transmitted in a single charging/discharging cycle. The time interval between charging/discharging cycles was measured as 0.99 hour.

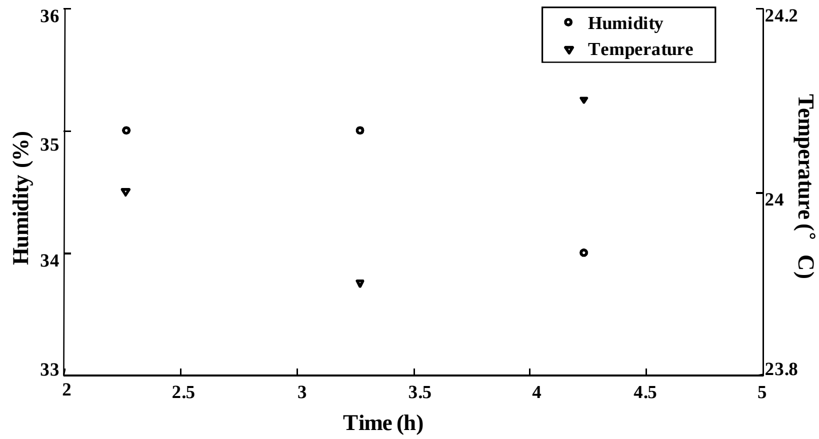

Figure A.8: Wireless sensor measurement results

The second super-capacitor of 50 mF was picked to provide sufficient energy to transmit three packets in this study. It should be noted that this capacity can be varied according to load requirements. Generally speaking, a larger capacity of the second super-capacitor results in a larger number of packets which can be transmitted during a charging/discharging cycle. Of course, the charging time also increases accordingly. The second super-capacitor was varied in terms of its capacity (50, 54.2, 83.3, 100, 200, 300, and 400 mF) to test the number of packets transmitted and the charging time during the steady state period. The experimental measurements are shown in Figure 4.9, and it can be seen that both the number of packets transmitted and the charging time increase linearly with the capacity of the second super-capacitor. It means that once a load is identified, the second super-capacitor can be selected based on the energy need of the load.

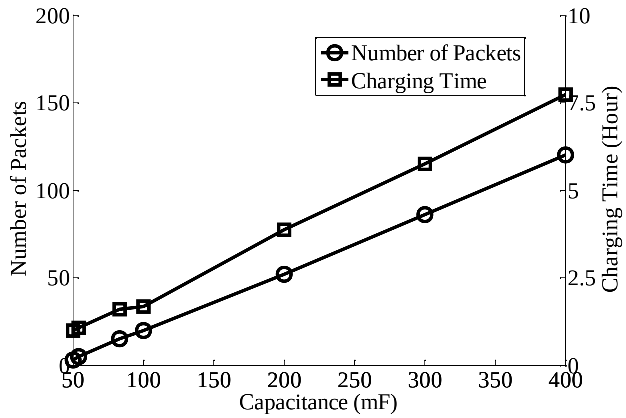

Figure A.9: Capacitance vs. number of packets and charging time

Performance Comparison with Other PMSs

During PMS operation, the 1.5 F (optimal value) super-capacitor C_1 was charged to 0.41 V and discharged to about 0.18 V. The measured average charging time was about 273 (±5) sec. This charging time is comparable to that of other similar studies. For example, Shantaram et al. [Shantaram2005] charged a 4 F capacitor from 0 to 0.5 V in two minutes using an MFC with a 265 cm^2 surface area sacrificial anode. Since a smaller surface area (127 cm^2) MFC was used in this study, a longer charging time was expected. Moreover, in [Donovan2008], a 10 F capacitor needed approximately 10 minutes to charge from 0 to 0.5 V using a sediment MFC with a 0.2 m^2 projected surface area.

Similar wireless data transmission has also been tested in [Donovan2008]. The wireless transmitter chosen was a Madgetech wireless temperature sensor (RFTC4000A, MadgeTech, Warner, NH, USA). Its RF carrier frequency is 418 MHz and its output power consumption is less than 1 mW. There were seven cycles of data transmission in 160 minutes, which indicates a shorter charging/discharging cycle than the proposed

PMS in this study. However, as the load in this work consumed more than 90 mW power, 0.99 hour charging/discharging cycle was already comparable. Moreover, in [Meehan2011], a hydrophone with similar power consumption (95 mW) was used as the load and its charging/discharging cycle was around 9.3 hour, indicating a great charging/discharging cycle performance of the power management system.

Generally, main MFC PMS designs can be categorized as capacitor-converter type [Shantaram2005], charge pump-capacitor-converter type [Meehan2011], and capacitor-transformer-converter type [Yang2012]. The characteristics of the aforementioned three PMS designs are summarized in Table A.1.

Table A.1: Characteristics of main MFC PMS designs

PMS design	*Location of super-capacitor(s)*	*Advantages*	*Disadvantages*
Capacitor-converter type	One super-capacitor which is directly connected with an MFC	Simple setup	It cannot drive the DC/DC converter if the open-circuit-voltage (OCV) is lower than the required input voltage of the converter
Charge pump-capacitor-converter type	One super-capacitor which is connected between the charge pump and the DC/DC converter	Higher discharging voltage of the super-capacitor	It cannot charge the super-capacitor if the MFC output is lower than the required input voltage of the charge pump
Capacitor-transformer-converter type (*design in this work*)	Two super-capacitors: one is directly connected with an MFC and the other is connected between the DC/DC converter and the load	Lower PMS input voltage	Additional super-capacitor is needed after the converter to improve the load driving performance

The latter two rows in Table A.1 are advanced PMS designs and they share some similarities: 1) having simple super-capacitor-based circuits; and 2) being able to boost a low MFC output voltage to 3.3 V. However, there are some distinct performance differences between these two PMSs. Table A.2 summarizes their differences. Some key points are listed as follows:

1. Capacitor-transformer-converter type PMS can accommodate lower input voltages depending on the design of Switch 1;
2. The charging speed of the capacitor-transformer-converter type PMS is not limited by the charge pump as the charge pump-capacitor-converter type PMS, resulting in a shorter charging/discharging cycle. This means that the capacitor-transformer-converter type PMS is more capable in harvesting more energy from an MFC even the charge pump-capacitor-converter type PMS has a higher power efficiency. The charge pump-capacitor-converter type PMS may not be able to utilize all energy produced by an MFC depending on the charge pump selected.

Table A.2: Key differences between two advanced PMSs

	Charge pump-capacitor-converter PMS	*Capacitor-transformer-converter PMS*
Minimum acceptable input voltage (V)	0.3	0.18
Capacitance selection	Capacitance is selected based on the need of load	The first super-capacitor is optimized to achieve a maximized MFC power output while the second super-capacitor is selected based on the need of load
Charging time needed to transmit 5 packets using MICAz	11.3 hours	1.06 hours

Therefore, due to its lower acceptable input voltage and shorter charging time, capacitor-transformer-converter PMS is adopted for MFC based wireless sensor network communication protocol study.

Appendix B

MFC Equivalent Circuit Model

Under practical operating conditions, the MFC output voltage V_{cell} is less than its open circuit voltage (E_{OCV}), mainly due to the activation loss, ohmic loss and concentration loss [Wang2005]:

$$V_{cell} = E_{OCV} - V_{act} - V_{ohmic} - V_{con} \tag{B.1}$$

where V_{act} , V_{ohmic} , and V_{con} represent the activation, ohmic and concentration voltage drops, respectively, and are functions of the MFC current as seen from the MFC polarization curve, which is also known as the voltage-current curve (U-I curve). The activation loss is usually estimated using the Tafel equation [Larminie2003]:

$$V_{act} = T[a + b\ln(I)] \tag{B.2}$$

where T is the absolute temperature, a and b are the coefficients to be identified, and I is the current. Since the estimation of R_{con} , R_{act} and R_{ohmic} is based on the steady-state analysis, there is no current flowing through the electrode capacitor C_e . As such, the current flowing through these resistors is identical, and the Ohmic loss can be estimated as follows:

$$V_{ohmic} = IR_{ohmic} \tag{B.3}$$

and the concentration loss can be represented [Larminie2003]:

$$V_{con} = -\frac{RT}{nF}\ln\left(1 - \frac{I}{I_{\text{limit}}}\right) \tag{B.4}$$

where R is the universal gas constant, n is the number of electrons per reaction mol, F is the Faraday constant, and I_{limit} is the limiting current of an MFC. Then the polarization curve can be simplified as follows:

$$V_{cell} = E_{OCV} - IR_{ohmic} - aT - bT\ln(I) + c\ln(1 - \frac{I}{I_{\text{limit}}}) \tag{B.5}$$

Appendix C

Derivation of Average Successful Probing Time (One TX, one RX)

Probability that first probing packet arrives in T_{active} period:

$$\Pr_{active} = T_{active} / (T_{DS1} + T_{active}) \tag{C.1}$$

Probability that first probing packet arrives in T_{DS1} period:

$$\Pr_{DS1} = T_{DS1} / (T_{DS1} + T_{active}) \tag{C.2}$$

Given in T_{active} period, probability that first probing packet arrives in idle listening period:

$$\Pr(IL \mid active) = T_{IL} / (T_{IL} + T_{PD}) \tag{C.3}$$

Given in T_{active} period, probability that first probing packet arrives in power down period:

$$\Pr(PD \mid active) = T_{PD} / (T_{IL} + T_{PD}) \tag{C.4}$$

C.1 First probing packet arrives during transmitter's active period

C.1.1 Idle Listening

If first probing packet arrives in idle listening period, only 1 probing packet is needed and the probing time is T_p.

C.1.2 Power Down

If first probing packet arrives in power down period, two extreme cases are:

Ⅰ. First probing packet arrives at the end of power down period

It will take $N_{\min 1} = 2$ probing packets to catch the idle listening period and the probing time is:

$$T_{\min 1} = T_d + 2T_p \tag{C.5}$$

Ⅱ. First probing packet arrives at the start of power down period

According to Equation (4.4), if the first probing packet arrives at the start of power down period, it will take $N_{\max 1} = N_2 + 1$ probing packets to catch the idle listening period and the probing time is:

$$T_{\max 1} = T_{probing} = N_2(T_p + T_d) + T_p \tag{C.6}$$

C.2 First probing packet arrives during transmitter's deep sleep period

Two extreme cases are:

Ⅲ. First probing packet arrives at the end of T_{DS1} period

Timing diagram for Case Ⅲ:

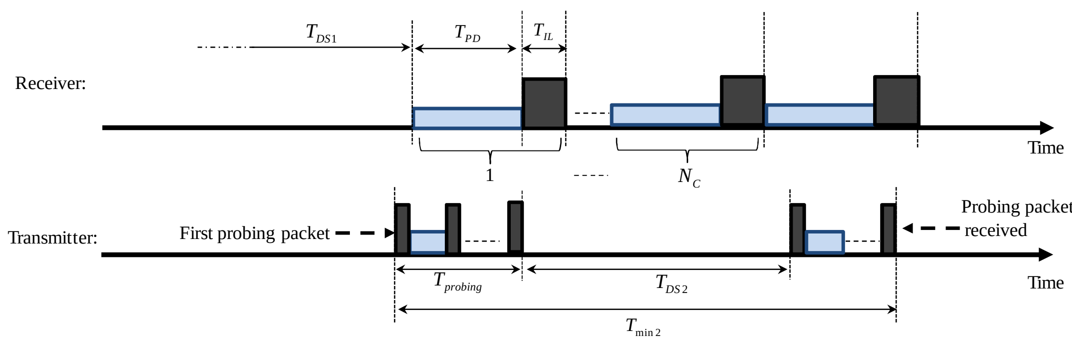

Figure C.1: Timing diagram for Case Ⅲ

Transmitter has to go to deep sleep mode once in order to catch the idle listening period. The first probing packet of the second round of probing/power down cycle may arrive at receiver's power down or idle listening period. The number of receiver's power down/idle listening cycles before the successful reception of probing packet is (N_C shown in the figure):

$$N_C = floor(\frac{T_{probing} + T_{DS2} - T_p}{T_{PD} + T_{IL}}) \tag{C.7}$$

Therefore, if:

$$F_C = floor(\frac{T_{probing} + T_{DS2} - T_p - N_C(T_{PD} + T_{IL})}{T_{PD}}) \tag{C.8}$$

equals to 0, then the first probing packet of the second round of probing/power down cycle arrives at receiver's power down period, it will need

$$N_2 + ceil(\frac{T_{PD} - (T_{probing} + T_{DS2} - T_p - N_C(T_{PD} + T_{IL}))}{T_p + T_d}) + 2 \tag{C.9}$$

probing packets to catch the idle listening period and the total probing time will be:

$$T_{probing} + T_{DS2} + ceil(\frac{T_{PD} - (T_{probing} + T_{DS2} - T_p - N_C(T_{PD} + T_{IL}))}{T_p + T_d})(T_p + T_d) + T_p \tag{C.10}$$

On the contrary, if F_C equals to 1, which means the first probing packet of the second round of probing/power down cycle arrives at receiver's idle listening period, it will then need N_2+2 probing packets and the total probing time will be: $T_{probing}$ + T_{DS2} + T_p. To summer up, the total probing packet number is:

$$\begin{aligned} N_{\min 2} &= F_C(N_2 + 2) \\ &+ (1 - F_C)(N_2 + ceil(\frac{T_{PD} - (T_{probing} + T_{DS2} - T_p - N_C(T_{PD} + T_{IL}))}{T_p + T_d}) + 2) \end{aligned} \tag{C.11}$$

and the total probing time is:

$$T_{\min 2} = F_C(T_{probing} + T_{DS2} + T_p)$$
$$+(1-F_C)(T_{probing} + T_{DS2} + ceil(\frac{T_{PD} - (T_{probing} + T_{DS2} - T_p - N_C(T_{PD} + T_{IL}))}{T_p + T_d})(T_p + T_d) + T_p) \quad \text{(C.12)}$$

Ⅳ. First probing packet arrives at the start of T_{DS1} period

Timing diagram for Case Ⅳ:

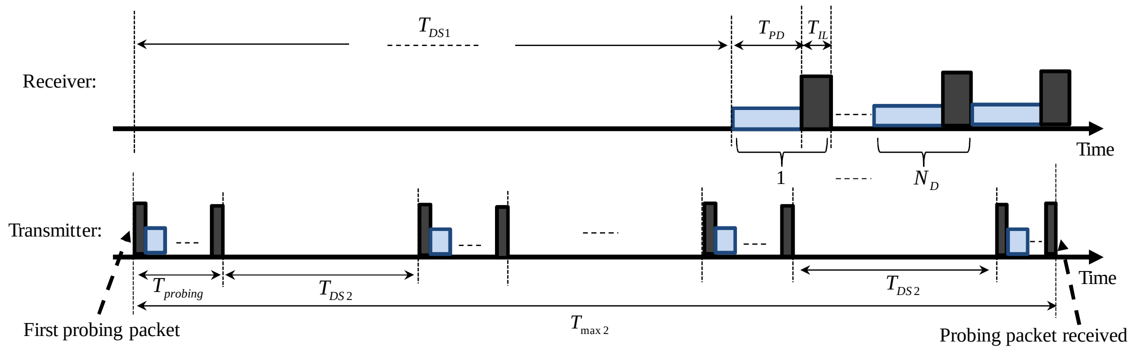

Figure C.2: Timing diagram for Case Ⅳ

Transmitter needs to do several ($ceil[\frac{T_{DS1}}{T_{probing} + T_{DS2}}]$) probing/deep sleep cycles so that its probing packet can catch the idle listening period of the receiver. The number of receiver's power down/idle listening cycles before the successful reception of probing packet is (N_D shown in the figure):

$$N_D = floor[\frac{ceil[\frac{T_{DS1}}{T_{probing} + T_{DS2}}](T_{probing} + T_{DS2}) - T_{DS1}}{T_{PD} + T_{IL}}] \quad \text{(C.13)}$$

Therefore, similar to the above analysis, if:

$$F_D = floor(\frac{ceil(\frac{T_{DS1}}{T_{probing} + T_{DS2}})(T_{probing} + T_{DS2}) - T_{DS1} - N_D(T_{PD} + T_{IL})}{T_{PD}}) \quad \text{(C.14)}$$

The total probing packet number is:

$$
\begin{aligned}
N_{\max 2} = {} & F_D(ceil(\frac{T_{DS1}}{T_{probing}+T_{DS2}})(N_2+1)+1) \\
& +(1-F_D)(ceil(\frac{T_{DS1}}{T_{probing}+T_{DS2}})(N_2+1) \\
& +ceil(\frac{T_{PD}-(ceil(\frac{T_{DS1}}{T_{probing}+T_{DS2}})(T_{probing}+T_{DS2})-T_{DS1}-N_D(T_{PD}+T_{IL}))}{T_p+T_d})+1)
\end{aligned}
\tag{C.15}
$$

The total probing time is:

$$
\begin{aligned}
T_{\max 2} = {} & F_D(ceil(\frac{T_{DS1}}{T_{probing}+T_{DS2}})(T_{probing}+T_{DS2})+T_p) \\
& +(1-F_D)(ceil(\frac{T_{DS1}}{T_{probing}+T_{DS2}})(T_{probing}+T_{DS2}) \\
& +ceil(\frac{T_{PD}-(ceil(\frac{T_{DS1}}{T_{probing}+T_{DS2}})(T_{probing}+T_{DS2})-T_{DS1}-N_D(T_{PD}+T_{IL}))}{T_p+T_d})(T_p+T_d)+T_p)
\end{aligned}
\tag{C.16}
$$

In conclusion, if the first probing packet may arrive anywhere with equal probability, the resulting average probing packet number is:

$$
N_{ave} = \Pr_{active}(\Pr(IL\,|\,active)+\Pr(PD\,|\,active)(\frac{N_{\min 1}+N_{\max 1}}{2}))+\Pr_{DS1}(\frac{N_{\min 2}+N_{\max 2}}{2}) \tag{C.17}
$$

The resulting average successful probing time is:

$$
T_{ave} = \Pr_{active}(\Pr(IL\,|\,active)t+\Pr(PD\,|\,active)(\frac{T_{\min 1}+T_{\max 1}}{2}))+\Pr_{DS1}(\frac{T_{\min 2}+T_{\max 2}}{2}) \tag{C.18}
$$

Appendix D

Derivation of Average Successful Probing Time (TX mode overlapping)

Given node A in TX mode and node B in RX mode, the average successful probing time is shown as in Equation (C.18). When their TX modes overlap, node A and B will choose to be in RX mode for c_A and c_B times, respectively. If $c_A<c_B$, node A will reach TX mode earlier than node B, the average probing time is the backoff time of node A plus T_{ave}; if $c_A>c_B$, node B will be in TX mode earlier than node A. As the data packet flow is from node A to node B, node A will turn into TX mode immediately after node B finds node A, the average probing time is the backoff time of node B plus $2T_{ave}$; if $c_A=c_B$, their TX modes overlap again and the backoff contention window size increases.

In summary, the average successful probing time after l ($1 \le l \le l_{\max}$) times of TX modes' overlapping is:

$$T_{overlap}(l) = \begin{cases} T_{TX} + \frac{3}{4}T_{ave} & (l=1) \\ (1+\frac{1}{2}+\frac{1}{2^l})T_{TX} + \sum_{i=1}^{2^l-1} \frac{2^l - i}{2^l(2^l+1-i)}[2(i-1)+3T_{ave}] & (1<l \le l_{\max}) \end{cases} \tag{D.1}$$

where $T_{TX}=T_{RX}=T_{DS1}+T_{active}$.

The probability of having the l_{th} time TX modes' overlapping is $½^{(l-1)}$, therefore, the overall average successful probing time when TX modes' overlapping happens is:

$$T_{ave\_overlap} = \sum_{l=1}^{l_{\max}} \frac{1}{2^{l-1}} T_{overlap}(l) \tag{D.2}$$

Finally, given two nodes operating under the proposed scheduling and one node in TX mode, the average successful probing time is:

$$
\begin{aligned}
T_a &= 0.5(T_{ave} + T_{ave\_overlap}) \\
&= 0.5[T_{ave} + \sum_{l=1}^{l_{\max}} \frac{1}{2^{l-1}} T_{overlap}(l)]
\end{aligned}
\tag{D.3}
$$

Appendix E

Data Flow Path Setup and Redirection

In this section, two scenarios are considered: 1) basic setup of data flow paths, and 2) data flow redirection between mesh nodes. The goal of this section is to provide sample flow entry configurations.

Basic setup of data flow paths

Basic setup of the data flow paths is the first step before redirecting data traffic among nodes. OpenFlow provides such a capability by inserting flow entries at the controller. For nodes connecting to the controller, they behave like a virtual “switch”, and any incoming traffic that matches the flow rules will be redirected using the flow actions. Figure E.1 shows the schematic of setting up a data flow path between two nodes.

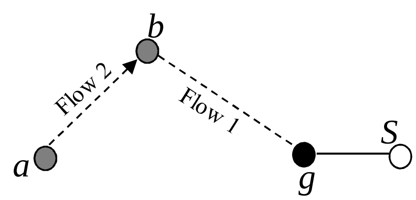

Figure E.1: Basic setup of data flow path

In Fig. E.1, nodes *a* and *b* are mesh nodes, node *g* is an Internet gateway with Ethernet connection to remote server *S*. In this section, the goal is to set up a connection between node *a* and remote server *g*. Assume node *b* already has a connection to node *g*. Any packet originated from node *b* arrives at node g first, which is transported to remote server *S* through Ethernet. Flow 1 in Fig. E.1 shows the original flow path without

OpenFlow. Now consider a new node *a* joins the network, it needs to find its next-hop neighbor so that packets originated from node *a* can be transported to the remote server. In this setup, node *b* is the neighbor and Flow 2 represents the added flow path.

To implement this scenario, traditional solutions require coordination between nodes *a* and *b*, as well as routing table updates, requiring relatively high local computation capability at each physical node. How to add Flow 1 data path using OpenFlow is shown in this section. OpenFlow addresses this by inserting flow entries at the controller, as shown in Table E.1. Thus no more action is required at local mesh nodes.

Table E.1: Flow entries (basic setup of data flow paths)

Flow Rules	'switch':"*data_path_id_node_a*" 'ingress-port':"*virtual_port_node_a*"	'switch':"*data_path_id_node_b*" 'ingress-port':"*physical_port_node_b*"
Flow Actions (push)	set-dst-ip = *ip_node_b* set-dst-mac = *mac_node_b* output = *physical_port_node_a*	set-dst-ip = *ip_node_S* set-dst-mac = *mac_node_S* output = *physical_port_node_b*

For node *a*, its virtual OpenFlow switch identification number is *data_path_id_node_a*, and packets originated from node *a* have the ingress-port number labeled as *virtual_port_node_a*. When the virtual OpenFlow switch receives such data flow matching the flow rules, packet headers will be manipulated to follow the flow actions. In this scenario, the destination of node *a*'s packets is node *b*, hence packets from node *a* must have their destination IP and MAC addresses rewritten to be node *b*. The "set-dst-ip" and "set-dst-mac" fields in the flow actions are used to rewrite packet headers. *ip_node_b* and *mac_node_b* are the IP and MAC address of node *b*, respectively. The modified packets must be output through the wireless radio interface, which is defined as *physical_port_node_a* in the flow entries. After node *b* receives the matching flows, flow actions take effect and it forwards redirected data traffic to gateway *g* via the original link. As the virtual OpenFlow switch takes over node *b* as well, corresponding flow entries need to be set. The destination IP and MAC addresses must be the remote server (*ip_node_S* and *mac_node_S*), the data flow must be output through node *b*'s wireless radio interface (*physical_port_node_b*).

Flow entries are pushed by the controller via a Flooglight module, the *Static Flow Pusher* API, which allows a user to manually insert flows into an OpenFlow network. After flow entries are pushed, the connection between node *a* and gateway *g* is established. This scenario serves the foundation for the following scenario.

Data flow redirection between mesh nodes

Redirecting data traffic between mesh nodes is essential for implementing centralized network control. Figure E.2 shows the corresponding schematic.

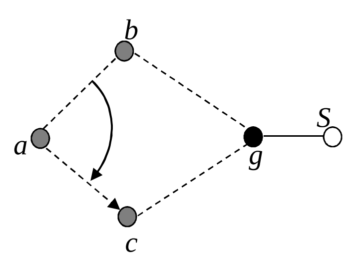

Figure E.2: Redirecting data Traffic between mesh nodes

Nodes *a*, *b*, and c are mesh nodes, node *g* is an Internet gateway with Ethernet connection to remote server *S*. Assume a link is already established from node *a* to gateway *g* via node *b* by pushing the flow entries shown in Table E.1. If node *b* experiences unexpected conditions, e.g. significant packet losses due to a full buffer or unstable wireless channel, node *a* will redirect its data flow to node *c* to avoid the degraded performance. This is a simple application model for traffic redirection, and the OpenFlow controller is able to make the adjustments based on reports from the installed performance measurement tool. In this scenario, we need to delete an existing connection from node *a* to server *S* via node *b* and set up a new connection to server *S* via node *c*.

Flow entries required for this scenario are shown in Table E.2. The previous flow actions for the *a*-*b*-*g* link must be removed before the new flow actions for the *a*-*c*-*g* link are pushed. For node *a*, the destination IP and MAC addresses must be modified to be node *c* (*ip_node_c* and *mac_node_c*). For node *c*, it receives incoming data flow from its wireless interface *physical_port_node_c* with matching virtual OpenFlow identification number *data_path_id_node_c*. It then modifies the destination IP and MAC addresses of packet headers to be the remote server *S* (*ip_node_S* and *mac_node_S*).

Table E.2: Flow entries (data flow redirection between mesh nodes)

Flow Rules	'switch':"*data_path_id_node_a*" 'ingress-port':"*virtual_port_node_a*"	'switch':"*data_path_id_node_c*" 'ingress-port': "*physical_port_node_c*"
Flow Actions (remove)	set-dst-ip = *ip_node_b* set-dst-mac = *mac_node_b* output = *physical_port_node_a*	
Flow Actions (push)	set-dst-ip = *ip_node_c* set-dst-mac = *mac_node_c* output = *physical_port_node_a*	set-dst-ip = *ip_node_S* set-dst-mac = *mac_node_S* output = *physical_port_node_c*

REFERENCES

[Akyildiz2002] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey”, Computer Networks, 38(4), pp.393–422, 2002.

[Alicherry2005] M. Alicherry, R. Bhatia, and E. L. Li., “Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks”, ACM MOBICOM, pp. 58-72, 2005.

[Allen1993] R.M. Allen and H.P. Bennetto, “Microbial fuel cells-electricity production from carbohydrates,” Applied Biochemistry and Biotechnology, 39(40). Pp. 27–40, 1993.

[AMPL] AMPL: A Modeling Language for Mathematical Programming. Available online at http://www.ampl.com/

[AMQP] Advanced Message Queuing Protocol. Available on line at http://www.amqp.org/.

[Antonopoulos2013] A. Antonopoulos, C. Verikoukis, C. Skianis, O. B. Akan, “Energy efficient network coding-based MAC for cooperative ARQ wireless networks”, in Ad Hoc Networks, 11(1), pp. 190-200, 2013.

[Audet2011] D. Audet, L.C. Oliveira, N. MacMillan, D. Marinakis, W. Kui, “Scheduling recurring tasks in energy harvesting sensors”, in Computer Communications Workshops, 2011 IEEE Conference on, 2011.

[Baumann2008] R. Baumann, S. Heimlicher, and B. Plattner, “Routing in large-scale wireless mesh network using temperature fields”, IEEE Network, 22, pp. 25-31, 2008.

[Bemmoussat2012] C. Bemmoussat, F. Didi, and M. Feham, “Efficient routing protocol to support QOS in wireless mesh network”, in International Journal of Wireless & Mobile Networks, 4(5), pp. 89-104, 2012.

[Bertsekas1991] D. P. Bertsekas, “Linear Network Optimization: Algorithms and Codes”, The MIT Press, 1991.

[Buchmann2001] I. Buchmann, “Batteries in a portable world: a handbook on rechargeable batteries for non-engineers, 2nd edition”, Cadex Electronics Inc, Richmond, Canada, 2001.

[Buettner2006] M. Buettner, G. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, pp. 307–320, Boulder, Colorado, USA, 2006.

[Capone2010] A. Capone, G. Carello, I. Filippini, S. Gualandi, and F. Malucelli, “Routing, scheduling and channel assignment in wireless mesh network: optimization models and algorithms,” in Ad-Hoc Network, 8(6), pp. 545–563, 2010.

[Cardell2004] R. Cardell, K. Smettem, M. Kranz, and K. Mayer, “Field testing a wireless sensor network for reactive environmental monitoring”, in Proceedings of the International Conference on Intelligent Sensors, pp.7-12, 2004.

[CC2420] A.S. Chipcon, “SmartRF CC2420 PRELIMINARY Datasheet,” rev. 1.2, February 2004

[Celandroni2013] N. Celandroni, E. Ferro, F. Davoli, and K. Xu, "A survey of architectures and scenarios in satellite-based wireless sensor networks: system design aspects," International Journal of Satellite Communications and Networking, 31(1), pp. 1-38, 2013.

[Chen2011] Y. Chen, C. M. Twigg, O. A. Sadik, and Shiqiong Tong, “A self-powered adaptive wireless sensor network for wastewater treatment plants,” Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshop (PERCOM), pp. 356-359, 2011.

[Chen2013] F. Chen, F. Lim, O. Abari, A. Chandrakasan, and V. Stojanovic, “Energyaware design of compressed sensing systems for wireless sensors under performance and reliability constraints,” IEEE Transactions on Circuits System I, Regular Papers, 60(3), pp. 650–661, 2013.

[Ciubotaru2006] B. Ciubotaru, R. Cioarga, and D. Stanescu, “Wireless solutions for telemetry in civil equipment and infrastructure monitoring,” 3rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, May 25-26, 2006.

[Couto2003] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in Proceedings of ACM MobiCom 2003, San Diego, CA, USA, 2003.

[Dam2003] T. Dam, and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks”, in Proceedings of ACM SenSys’03, pp. 171-180, 2003.

[Deb2001] S. Deb, M. Kapoor, A. Sarkar, “Error avoidance in wireless networks using link state history”, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, 2, pp. 786-795, 2001.

[Dely2011] P. Dely, A. Kassler, and N. Bayer, “OpenFlow for wireless mesh networks,” In Computer Communications and Networks, 2011 Proceedings of 20th International Conference on, pp. 1-6. 2011.

[Devarajan2012] R. Devarajan, S. C. Jha, U. Phuyal, and V. K. Bhargava, “Energyaware resource allocation for cooperative cellular network using multiobjective optimization approach,” IEEE Transactions on Wireless Communication, 11(5), pp. 1797–1807, 2012.

[Dewan2010] A. Dewan, C. Donovan, D. Heo, and H. Beyenal, “Evaluating the performance of microbial fuel cells powering electronic devices,” Journal of Power Sources, 195, pp. 90-96, 2010.

[Dianati2005] M. Dianati, X. Shen, and S. Naik, “A new fairness index for radio resource allocation in wireless networks,” in IEEE Wireless Communications and Networking Conference, 2, pp. 712-717, 2005.

[Donovan2008] C. Donovan, A. Dewan, D. Heo and H. Beyenal, “Batteryless, wireless sensor powered by a sediment microbial fuel cell,” Environmental Science Technology, 42, pp. 8591-8596, 2008.

[Donovan2011] C. Donovan, A. Dewan, H. Peng, D. Heo and H. Beyenal, “Power management system for a 2.5 W remote sensor powered by a sediment microbial fuel cell,” Journal of Power Sources, 196, pp. 1171-1177, 2011.

[Draves2004] R. Draves, and J. Padhye, “Routing in multi-radio, multi-hop mesh networks,” in MobiCom2004, pp. 114–28, 2004.

[Du2007] Z. Du, H. Li and T. Gu, “A state of the art review on microbial fuel cells: a promising technology for wastewater treatment and bioenergy,” Biotechnology Advances, 25, pp. 464–482, 2007.

[Dzapo2009] H. Dzapo, “Digital measuring system for monitoring motor shaft parameters on ships”, IEEE Transaction on Instrumentation and Measurement, 58(10), pp. 3702-3712, 2009.

[Elson2002] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts”, In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation, Boston, MA, USA, 2002.

[Fafoutis2011] X. Fafoutis and N. Dragoni, “OD-MAC: an on-demand MAC protocol for energy harvesting wireless sensor networks,” in Proceedings of the 8th ACM Symposium on Performance Evaluation of Wireless AD-Hoc, Sensor, and Ubiquitous networks, pp. 49-56, 2011.

[Fink2012] J. Fink, A. Ribeiro, and V. Kumar, “Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams,” in Proceedings of IEEE, 100(1), pp. 150–163, 2012.

[Gadelmawla2002] E.S. Gadelmawla, M.M. Koura, I.M. Elewa, and H.H. Soliman, "Roughness parameters." Journal of Materials Processing Technology, 123(1), pp. 133-145, 2002.

[Ganjali2004] Y. Ganjali, A. Keshavarzian, “Load balancing in ad hoc networks: single-path routing vs. multi-path routing”, Proceedings of IEEE INFOCOM, Hong Kong, 2004.

[Genender2010] E. Genender , C. L. Holloway , K. A. Remley , J. Ladbury , G. Koepke and H. Garbe "Simulating the multipath channel with a reverberation chamber: Application to bit error rate measurements", IEEE Transactions on Electromagnetic Compatibility, 52(4), pp. 766-777, 2010.

[Glasgow2004] H.B. Glasgow, J.M. Burkholder, R.E. Reed, A.J. Lewitus, and J.E. Kleinman, “Real-time remote monitoring of water quality: a review of current applications, and advancements in sensor, telemetry, and computing technologies”, in Journal of Experimental Marine Biology and Ecology, 300, pp. 409-448, 2004.

[Goyal2010] M. Goyal, S. Prakash, W. Xie, Y. Bashir, H. Hosseini, and A. Durresi, “Evaluating the impact of signal to noise ratio on IEEE 802.15.4 PHYlevel packet loss rate,” in Network-Based Information Systems (NBiS), 13th International Conference on, Sept. 2010, pp. 279–284.

[Gungor2006] V.C. Gungor and F. Lambert, “A survey on communication networks for electric system automation,” Journal of Computer Networks, 50, pp. 877-897, 2006.

[Guo2012] Z. Guo, J. Huang, B. Wang, S. Zhou, J.-H. Cui, and P. Willett, “A practical joint network-channel coding scheme for reliable communication in wireless networks,” in IEEE Transactions on Wireless Communications, 11(6), pp. 2084-2094, 2012.

[Ha2010] P.T. Ha, H. Moon, B.H. Kim, H.Y. Ng, and I.S. Chang, “Determination of charge transfer resistance and capacitance of microbial fuel cell through a transient response analysis of cell voltage.” Biosensors and Bioelectronics, 25, pp. 1629-1634, 2010.

[Hasenfratz2010] D. Hasenfratz, A. Meier, C. Moser, J.J. Chen and L. Thiele, "Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks," IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2010.

[He2013] Y. He, O. S. Salih, C.-X. Wang, and D. Yuan, "Deterministic process-based generative models for characterizing packet-level bursty error sequences," Wireless Communication and Mobile Computing, 2013, doi:10.1002/wcm.2356, to be published.

[Hoiydi2004] A. El-Hoiydi and J-D. Decotignie, "WiseMAC: an ultra-low power MAC protocol for the downlink of infrastructure wireless sensor networks," in Proceedings of the Ninth International Symposium on Computers and Communications, 2, Washington, DC, USA, pp. 244–251, 2004.

[IEEE2003] IEEE, 2003, IEEE Standard 802 Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks

[Jain2003] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in Proceedings of ACM MOBICOM, San Diego, CA, 2003.

[Joeph2009] V. Joseph, V. Sharma, U. Mukherji, "Joint power control, scheduling and routing for multihop energy harvesting sensor networks", in Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, 2009.

[Joseph2009] V. Joseph, V. Sharma and U. Mukherji, "Joint power control, scheduling and routing for multihop energy harvesting sensor networks," Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, 2009.

[Junnila2010] S. Junnila , H. Kailanto , J. Merilahti , A.-M. Vainio , A. Zakrzewski , M. Vehkaoja and J. Hyttinen "Wireless, multipurpose in-home health monitoring platform: two case trials," IEEE Transactions on Information Technology in Biomedicine, 14(2), pp. 447-455, 2010.

[Jurdak2010] R. Jurdak, A.G. Ruzzelli and G. O'Hare, "Radio sleep mode optimization in wireless sensor networks," IEEE Transactions on Mobile Computing, 9(7), pp. 955-968, 2010.

[Kansal2007] A. Kansal, J. Hsu, S. Zahedi and M.B. Srivastava, "Power management in energy harvesting sensor networks," ACM Transactions on Embedded Computing Systems, 6(4), Article: 32, 2007.

[Kashanaki2012] M. Kashanaki, Z. Beheshti, and M.R. Meybodi., "A distributed learning automata based gateway load balancing algorithm in wireless mesh networks", In Instrumentation Measurement, Sensor Network and Automation (IMSNA), 2012 International Symposium on, 1, pp. 90-94, 2012.

[Khan2012] J. Y. Khan, M. R. Yuce, G. Bulger, and B. Harding, "Wireless body area network (WBAN) design techniques and performance evaluation," Journal of Medical System, 36(3), pp. 1441–1457, 2012.

[Khouzani2011] M. Khouzani, S. Sarkar, and K. Kar, "Optimal routing and scheduling in multihop wireless renewable energy networks", in Proceedings of the 6th Information Theory and Applications Workshop, 2011.

[Kodialam2003] M. Kodialam, T. Nandagopal, “Characterizing the achievable rates in multihop wireless networks”, Proceedings of ACM MOBICOM, San Diego, CA, USA, pp. 42-54, September 2003.

[Kumar2013] D. Kumar, S. Rajasegarar, and M. Palaniswami, “Automatic sensor drift detection and correction using spatial kriging and kalman filtering,” Distributed Computing in Sensor Systems, IEEE International Conference on, pp. 183-190, 2013.

[Kyasanur2005] P. Kyasanur, and N.H. Vaidya, “Capacity of multi-channel wireless networks: impact of number of channels and interfaces”, In Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, New York, NY, USA, pp 43–57, 2005.

[Larminie2003] J. Larminie, and A. Dicks, Fuel Cell Systems Explained, 2nd Ed. John Wiley&Sons Ltd, 2003.

[Li2008] F. Li, X.Y. Li, A. Nusairat, and Y. Wu, “Gateway placement for throughput optimization in wireless mesh networks”, in Mobile Network Application, 12, pp. 198-211, 2008.

[Liang2007] P. Liang, X. Huang, M.Z. Fan, X.X. Cao, and C. Wang, “Composition and distribution of internal resistance in three types of microbial fuel cells,” Applied Microbiology and Biotechnology, 77(3), pp. 551-558, 2007.

[Liu2005] H. Liu, S. Cheng, and B.E. Logan, “Production of electricity from acetate or butyrate using a single-chamber microbial fuel cell,” Environmental Science Technology, 39, pp. 658-662, 2005.

[Logan2006] B.E. Logan, B. Hamelers, R. Rozendal, U. Schroder, J. Keller, S. Freguia, P. Aelterman, W. Verstraete and K. Rabaey, “Fuel cells: methodology and technology,” Environmental Science and Technology., 40(17), pp. 5181–5192, 2006.

[LTC3108] Ultralow Voltage Step-Up Converter and Power Manager, Linear Tech. Corp. 2010.

[Luo2009] S. Luo, N. Polu, Z. Chen, and J. Slipp, “RF channel modeling of a WSN testbed for industrial environment,” in Proceedings of IEEE Radio Wireless Symposium, Phoenix, AZ, USA, pp. 375–378, 2009.

[Luo2011] H. Luo, H. Tao, H. Ma, and S. K. Das, "Data fusion with desired reliability in wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems, 22(3), pp. 501 - 513, 2011.

[Mahlknecht2004] S. Mahlknecht and M. Bock, “CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks,” in Proceedings of the IEEE International Workshop on Factory Communication Systems, pp. 73-80, 2004.

[McKeown2008] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Computer Communication Review, 38, pp. 69–74, 2008.

[MDA300] Crossbow Technology, Inc. Available online at http://www.cens.ucla.edu/~mhr/daq/datasheet.pdf

[Meehan2011] A. Meehan, H.W. Gao, and Z. Lewandowski, “Energy harvesting with microbial fuel cell and power management system,” IEEE Transactions on Power Electronics, 26(1), pp. 176-181, 2011.

[Merlin2009] C. Merlin and W. Heinzelman, “Schedule adaptation of low-power listening protocols for wireless sensor networks,” IEEE Transactions on Mobile Computing, 9(5), pp. 672–685, 2009.

[MICAz] MICAz Wireless Measurement System, Crossbow Technology, Inc. Available online at http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf

[Moser2010] C. Moser, J.J. Chen, and, L. Thiele, “Dynamic power management in environmentally powered systems”, in Design Automation Conference, 15th Asia and South Pacific, pp. 81-88, 2010.

[Naderi2012] M.Y. Naderi, H.R. Rabiee, M. Khansari, and M. Salehi, “Error control for multimedia communications in wireless sensor networks: A comparative performance analysis,” in Ad Hoc Networks 10(6), pp. 1028–1042, 2012.

[Nagaonkar2013] J. Nagaonkar, and U. Srija, "Implementation and evaluation of performance parameters of Zigbee transreceiver." In Advances in Computing, Communication, and Control, pp. 484-491, Springer Berlin Heidelberg, 2013.

[Ning2012] Z. Ning, L. Guo, Y. Peng, and X. Wang, “Joint scheduling and routing algorithm with load balancing in wireless mesh network,” in Computers and Electrical Engineering, 38(3), pp. 533-550, 2012.

[Niyato2007] D. Niyato, E. Hossain, and A. Fallahi, “Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks: performance analysis and optimization,” IEEE Transactions on Mobile Computing, 6(2), 221–236, 2007.

[Noh2008] D. Noh, I. Yoon and H. Shin, “Low-latency geographic routing for asynchronous energy-harvesting WSNs,” in Journal of Networks, 3(1), pp. 78-85, 2008.

[Oh2007] S.E. Oh, and B.E. Logan, “Voltage reversal during microbial fuel cell stack operation,” Journal of Power Sources, 167, pp. 11–17, 2007

[OpenFlow] OpenFlow Technology. Available online at http://www.openflow.org

[OVS] Open vSwitch. Available online at http://openvswitch.org/

[Padhye2004] J. Padhye, R. Draves, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in Proceedings of ACM MobiCom 2004, Philadelphia, PA, USA, 2004.

[Pakzad2008] S.N. Pakzad, G.L. Fenves, S. Kim, and D.E. Culler, “Design and implementation of scalable wireless sensor network for structural monitoring”, Journal of Infrastructure System, 14(1), pp. 89-101, 2008.

[Panicker2013] A.S. Panicker, S. Seetha, and J. Sharmila., “Survey on various load balancing techniques in wireless mesh networks”, International Journal of Advanced Research in Electronics and Communication Engineering, 3, 2013.

[Pantelidou2008] A. Pantelidou, and, A. Ephremides, “What is optimal scheduling in wireless networks?” in Proceedings of the 4th Annual International Conference on Wireless Internet, 2008.

[Park2003] D.H. Park, and J.G. Zeikus, “Improved fuel cell and electrode designs for producing electricity from microbial degradation,” Biotechnology and Bioengineering, 81(3), pp.348-355, 2003.

[Peretz2009] M.M. Peretz, and S.B. Yaakov, “A heuristic digital control method for optimal capacitor charging,” in Energy Conversion Congress and Exposition, pp. 118-1125, 2009.

[Pines1998] D.J. Pines, and P.A., Lovell, “Conceptual framework of a remote wireless health monitoring system for large civil structures,” in Smart Materials and Structure, 7(5), pp. 627–636, 1998.

[Polastre2004] J. Polastre, J. Hill and D. Culler, “Versatile low power media access for wireless sensor networks,” in Proceedings of the Second ACM Conference Embedded Networked Sensor Systems, pp. 95–107, 2004.

[Prasad2006] R. Prasad, and H. Wu, “Gateway deployment optimization in cellular Wi-Fi mesh networks”, in Journal of Networks, 1(3), pp. 31-39, 2006.

[Prowler] Prowler Probabilistic Wireless Network Simulator. Available online at http://w3.isis.vanderbilt.edu/projects/nest/37/index.html

[Quan2006] B. Quan, W.-J. Huang, “Trade-off between reliability and energy consumption in transport protocols for wireless sensor networks”, International Journal of Computer Science and Network Security, 6(8B), 2006.

[Rad2007] A. H. M. Rad and V. W. S. Wong., “Joint channel allocation, interface assignment and mac design for multi-channel wireless mesh networks”, In Proceedings of IEEE INFOCOM, pp. 1469-1477, 2007.

[Rao2011] S. Rao, Y.K Sundara, and K. Nageswara, “A survey: routing protocols for wireless mesh networks”, International Journal of Research and Reviews in Wireless Sensor Networks, 1(3), 2011.

[Rappaport1989] T. S. Rappaport, “Indoor radio communications for factories of the future,” IEEE Communications Magazine, 27(5), pp. 15–24, 1989.

[Rogan2004] J. Rogan, and D.M. Chen, “Remote sensing technology for mapping and monitoring land-cover and land-use change”, Progress in Planning, 61, pp. 301–325, 2004.

[Schmidt2009] D. Schmidt, M. Berning, and N. Wehn, "Error correction in single-hop wireless sensor networks - a case study," In Proceedings of the IEEE Conference on Design, Automation, and Test in Europe (DATE'09), pp. 1296-1301, 2009.

[Shantaram2005] A. Shantaram, H. Beyenal, R. Raajan, A. Veluchamy, and Z. Lewandowski, “Wireless sensors powered by microbial fuel cells,” Environmental Science Technology, 39, pp. 5037-5042, 2005.

[Singh2011] B. Singh, and G. Singh, “Performance evaluation and optimization of DSR routing algorithm over 802.11 based wireless mesh network”, International Journal on Computer Science and Engineering, 3(5), pp. 1980-1985, 2011.

[SNOPT] P. Gill, “User’s Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming”. Available online at: http://www.stanford.edu/group/SOL/guides/sndoc7.pdf

[Talha2011] B. Talha and P. Matthias, “Channel models for mobile-to-mobile cooperative communication systems: A state of the art review,” IEEE Vehicular Technical Magazine, 6(2), pp. 33–43, 2011.

[Tang2006] L. Tang, K. C. Wang, Y. Huang, and F. Gu, “Radio channel characteristics of ZigBee wireless sensors in machine shop for plant floor process monitoring,” in Proceedings of ASME International Manufacturing Science and Engineering Conference (MSEC), Ypsilanti, MI, pp. 1–8, 2006.

[Tang2007] L. Tang, K. C. Wang, Y. Huang, and F. Gu, “Channel characterization and link quality assessment of IEEE 802.15.4-compliant radio for factory environment,” IEEE Transactions on Industrial Informatics, 3(2), pp. 99– 10, 2007.

[Tang2008] L. Tang, K.-C. Wang, M. Liu, Y. Huang, and F. Gu, “Multipath effect and design considerations for wireless sensors in plant floor environments”, Proc. Of 2008 ISFA, 2008 International Symposium Flexible Automation, Atlanta, GA, pp. 1-7, June 23-26, 2008.

[Tang2009a] X. Tang, R. Liu, P. Spasojevic, and H. Vincent, "On the throughput of secure hybrid-ARQ protocols for Gaussian block-fading channels," IEEE Transactions on Information Theory, 55(4), pp. 1575-1591, 2009.

[Tang2009b] L. Tang, K.-C. Wang, and Y. Huang, “Performance evaluation and reliable implementation of data transmission for wireless sensors on rotating mechanical structures”, in Structural Health Monitoring, 8(2), pp.113–124, 2009.

[Tang2010] L. Tang, “Packet error rate predictive model for sensor radios on fast rotating structures”, Ph.D Thesis, Clemson University, SC, USA, August, 2010.

[Tang2012] L. Tang, F. Yang, D. Zhang, K.-C. Wang, and Y. Huang, "Investigation of the effect of moving forklift on data transmission of IEEE 802.15. 4 wireless sensor radio," Journal of Manufacturing Science and Engineering, 134(4), pp. 1-7, 2012.

[Vohra2012] R. Vohra, S.N. Sawhney, and M. Sunandika, "WLAN performance improvisation by fine tuning IEEE 802. 11 parameters," International Journal of Computer Applications, 43(6), pp. 16-19. 2012.

[Wang2005] C. Wang, M.H. Nehrir, and S.R. Shaw, “Dynamic models and model validation for PEM fuel cells using electrical circuits,” IEEE Transactions on Energy Conversion, 20(2), pp. 442-451, 2005.

[Wang2007] K.-C. Wang, L. Tang, and Y. Huang, “Wireless sensors on rotating structures: performance evaluation and radio link characterization”, Proceedings of ACM WinTECH Workshop, pp. 3–10, 2007.

[Wang2009] K.-C. Wang, J. Jacob, L. Tang, and Y. Huang, “Transmission error analysis and avoidance for IEEE 802.15.4 wireless sensors on rotating structures”, International Journal of Sensor Networks, 6(3) 2009.

[Wark2007] T. Wark, P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P.Valencia, D. Swain, and G. Bishop-Hurley, “Transforming agriculture through pervasive wireless sensor networks,” IEEE Pervasive Computing, 6(2), pp. 50–57, 2007.

[Werb2005] J. Werb, M. Newman, V. Berry, S. Lamb, D. Sexton, and M. Lapinski, “Improved quality of service in IEEE 802.15.4 mesh networks,” in Proceedings of International Workshop Wireless and Industrial Automation, San Francisco, CA, USA, pp. 1-6, 2005.

[Werner2005] G. Werner, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, “Monitoring volcanic eruptions with a wireless sensor network,” in Proceedings of Wireless Sensor Networks, pp. 108–120, 2005.

[Willig2005] A. Willig, K. Matheus, and A. Wolisz, “Wireless technology in industrial networks,” Proceedings of the IEEE, 93(6), pp. 1130-1151, 2005

[Wong2006] K.-J. Wong and D.K. Arvind, “SpeckMAC: low power decentralized MAC protocols for low data-rate transmissions in Specknets,” in Proceedings of the Second International Workshop Multi-hop Ad Hoc Networks: From Theory to Reality, pp. 71–78, 2006.

[Wu2010] P. Wu and N. Jindal, “Performance of hybrid-ARQ in block-fading channels: a fixed outage probability analysis,” IEEE Transaction on Communication, 58(4), pp. 1129–1141, 2010.

[Xue2010] Y. Xue, B. Ramamurthy, and M. Burbach, “A two-tier wireless sensor network infrastructure for large-scale real-time groundwater monitoring”, in 5th IEEE International Workshop on Practical Issues in Building Sensor Network Applications, Denver, Colorado, 2010.

[Yang2005] Y. Yang, J. Wang, and R. Kravets, “Interference-aware load balancing for multihop wireless networks”, Technical report, UIUCDCS-R-2005- 2526, UIUC, 2005.

[Yang2012] F. Yang, D. Zhang, T. Shimotori, K.-C. Wang and Y. Huang, “Study of transformer-based power management system and its performance optimization for microbial fuel cells,” Journal of Power Sources, 205, pp. 86-92, 2012.

[Ye2004] W. Ye, J. Heidemann and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” IEEE/ACM Transactions on Networking, 12(3), pp. 493 – 506, 2004.

[Yoshida2011] M. Yoshida, T. Kitani, M, Bandai, T. Watanabe, P. Chou, W.K.G. Seah, “Probabilistic data collection protocols for energy harvesting sensor networks”, in Local Computer Networks, 2011 IEEE 36th Conference on, pp. 366-373, 2011.

[Zhang2013] X. Zhang, W. Liang, H. Yu, and X. Feng, “Reliable transmission scheduling for multi-channel wireless sensor networks with low-cost channel estimation,” IET Communications, 7(1), pp. 71–81, 2013

[Zhou2007] S.Q. Zhou, W. Ling, and Z.X. Peng, “An RFID-based remote monitoring system for enterprise internal production management,” International Journal of Advanced Manufacturing Technology, 33(7–8), pp. 837–844, 2007.