ABSTRACT

Title of Document: Design and Analysis of Free Space Optical Sensor Networks for Short-Range Applications

Navik Agrawal, Doctor of Philosophy, 2010

Directed By: Professor Christopher C. Davis
Department of Electrical and Computer Engineering

Free space optical communication (FSOC) systems using direct detection and line of sight (LOS) laser links can provide spatially efficient and physically secure connectivity for wireless sensor networks. The FSOC system can be developed with low power microcontrollers so that the entire sensor system can be implemented on a single printed circuit board. Available data rates can range from kb/s to hundreds of Mb/s with the complete system consuming power only in the tens of mW. These features are advantageous for low-power communication networks over short distances in environments where LOS is available, and where radio frequency (RF) connectivity must be avoided because of interference or security issues. In particular, the faster data acquisition rates of FSOC systems are extremely attractive in applications where the sensor systems, or "motes", remain in sleep mode most of the time and need to transmit large amounts of data in extremely short bursts when they wake up. However, in order for directional FSO sensor networks to become

viable short-range solutions, the networks must provide signal coverage over a wide field of view without strict optical alignment requirements, operate with efficient media access protocols that can handle network traffic in an efficient manner, and minimize random access times for the independent transmitting motes within the network. These challenges are the focus of this dissertation.

In general, narrow optical beams used for FSOC require precise and complex pointing, acquisition, tracking and alignment methods. This dissertation addresses the challenge of alignment for FSO-based nodes by designing optical transceiver architectures with multiple narrow field of view (FOV) transmitters and a single, wide angle receiver. The architecture consists of rings of multiple transmitters surrounding a photodiode for light collection. Each ring is tilted at a different angle so that a wide transmission FOV can be obtained, thereby allowing point-multipoint communication. Depending on the number of transmitters and the transmitter's divergence angle, different FOVs can be tailored to fit the requirements of the target application. The developed transmitter design requires only a few milliwatts of transmission power from each transmitter to cover its respective FOV, which is sustainable with drive currents up to 10 milliamps using vertical cavity surface emitting lasers (VCSELs), making it a more practical strategy for a compact battery driven device.

The other major challenge is designing the proper media access control (MAC) protocol, which provides nodes with addresses and channel access capability so that

directional links between multiple nodes can be formed. The challenge lies in the fact that most nodes are blind to other nodes' transmissions because of their relatively narrow directional links. Because of this blindness, packet collisions are inevitable. Therefore, an efficient multiple access protocol needs to be designed for the FSOC system to ensure successful directional communication between the motes and cluster heads for data collection and relaying. While there are many protocols that allow multiple access and provide collision avoidance for traditional RF systems, these protocols are not optimized for FSOC systems consisting of multiple narrow FOV transmitters. Instead, a directional MAC (DMAC) protocol is developed from existing RF protocols, but modified for FSOC technology. It overcomes the limitations in FSOC communication resulting from directionality by setting up a master-slave network architecture where communication takes place between a sensing system, "mote", and a central control station, or "cluster head", which is designed with a multiple VCSEL transmitters. In this way, the physical transmitter sources of the cluster head become an integral part of the FSOC DMAC protocol. In this type of architecture, the master node, or cluster head, has the dual functionality of coordinating network traffic and aggregating data from all the slave nodes, or motes, that are within its field of view (FOV). Multiple cluster heads can form a directional network backbone, and can relay signals collected from a mote through other cluster heads, until the signal is delivered to its destination.

In summary, this dissertation provides: 1) the design and implementation of small and inexpensive short-range FSOC systems that can be implemented using standard

“off the shelf” components including a microcontroller and sensor device to form a complete standalone package; 2) development of a DMAC protocol that is optimized for the implemented FSOC system and target network applications; 3) network performance evaluation and optimization for the combined FSOC hardware, network architecture, and DMAC protocol. This is done through a series of hardware tests on an experimental prototype FSOC sensor network consisting of 10 motes and 1 cluster head and simulations of larger network sizes.

Design and Analysis of Free Space Optical Sensor Networks for Short-Range Applications

By

Navik Agrawal

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor Christopher C. Davis, Chair
Professor Thomas E. Murphy
Professor Neil Goldsman
Professor Martin Peckerar
Professor Gregory B. Baecher, Dean's Representative
Professor Stuart D. Milner

Dedication

To my wonderful mom, dad, sister, and dog (Laila)

for their constant love, guidance, and inspiration.

Acknowledgements

The path to the finish line was not as straightforward as I would have hoped for it to be, and I have to thank my advisor, Professor Christopher C. Davis, for helping me find my way. I am especially grateful for his constant guidance, encouragement, and unwavering support. I also want to thank Dr. Stuart D. Milner for all of his counseling and advice in the field of networking. I would also like to thank my dissertation committee members for their time and service.

I was fortunate to work with an amazing cast of coworkers. I want to thank John, Jaime, Mohammed, and Ehren for all their advice, constructive criticisms, and professionalism that fostered a wonderful working environment. In particular, I am grateful for all the help John Rzasa provided in the lab and at the machine shop. I would also like to thank my undergraduate research assistants Jolyon Zook and Asaf Kaya for their hard work and helpful comments.

I cannot thank my family enough for their mental and loving support throughout this entire program. Without the support from my mother, Seema Agrawal, father, Satish Agrawal, sister, Raani Agrawal, and pet, Laila Agrawal, completing this program would have been an extremely difficult task. Finally, I would like to thank all my friends for the wonderful times that allowed me to mentally unwind. In particular, I'd like to thank Sanjay Varma and Zach Wilkes for making life at the University of Maryland an enjoyable one.

List of Publications

[1] N. Agrawal, and C. C. Davis, "Design of free space optical omnidirectional transceivers for indoor applications using non-imaging optical devices," *Proc. SPIE, Free-Space Laser Communications VIII,* vol. 7091, 2008.

[2] N. Agrawal, S. Milner, and C. C. Davis, "Free space optical sensor network for fixed infrastructure sensing," *Proc. SPIE, Free-Space Laser Communications IX,* vol. 7464, 2009.

[3] C. C. Davis, M. Eslami, and N. Agrawal, "Channel modeling for FSO communications and sensor networking inside structures," *Proc. SPIE, Free-Space Laser Communications IX,* vol. 7464, 2009.

[4] N. Agrawal, C. C. Davis, and S. Milner, "Free space optical sensor networking for underwater sensing applications," *Proc. ISSNIP 2009, Fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing,* (pp. 475-480).

[5] N. Agrawal, S. Milner, and C. C. Davis, "Free space optical sensor network for short-range applications," *Proc. SPIE, Free-Space Laser Communications X,* vol. 7814, 2010.

Table of Contents

List of Figures

Chapter 1 – Introduction to Short-Range Free Space Optical Sensor Networks

1.1 – Preface

In the last 10 years (2000-2010) the consumer market has seen rapid advancements in the functionality of portable data communication devices, which has led to a continuous growth in the user demand for wireless content. With more wireless content in the air now than ever before, wireless networks are becoming increasingly congested and the interference generated by signals from wireless devices is becoming more of a concern when designing new wireless devices. Thus, it is necessary to investigate alternative technologies to conventional radio frequency (RF) wireless devices, which can potentially handle the increasing network demand and generate less interference. At this point in time, the main alternative technology in wireless communications is free space optics (FSO). FSO has the potential to provide extremely fast data rates and very secure communication links as compared to RF, but the ideal environment for these links is point to point, where line of sight (LOS) exists [1] [2]. At first, this LOS requirement may seem to limit the applicability of FSO technology in networks where multiple nodes must be interconnected, but it has been shown that FSO directional networks can be useful for certain network applications [3]. Examples of these applications include structural health monitoring within a fixed infrastructure [4], real-time performance feedback from sensors placed inside airplanes, and rapid upload/download links

within various terminals in settings such as indoor offices or houses [5] [6], airplane cabins [7], and even along the surface under shallow waters [8]. In order for FSO networks to become viable alternatives to RF based networks, efficient communication components and network protocols tailored for directional FSO must be designed. This dissertation specifically focuses on the design of low power and short-range wireless sensor networks where multiple sensor devices need to download and upload information from a central processing station. With increased data rates and spatial diversity techniques, these low power optical networks can be scalable from b/s to hundreds of Mb/s, allowing for the transfer of large amounts of data in a very physically secure manner. In general, there are situations where an optical wireless network is advantageous over a RF wireless network, but the challenge remains in making these networks practical, efficient, and cost effective when compared to traditional radio frequency technologies, so that an alternative wireless technology implementation can become realistic.

1.2 – Defining a Low Power and Short-Range Sensor Network

This dissertation focuses on the design, implementation, and performance of free space optical communication (FSOC) systems for low power and short-range sensor networks. The FSOC system consists of several subsystems: sensor, transmitter, receiver, and central processor. All these subsystems are implemented together on

1 printed circuit board, which is then called the short-range (SR) FSOC system. Some examples of sensors are temperature, pressure, and acceleration sensors.

Typical wireless sensor networks operate with data rates in the hundreds of kb/s range [9]. These data rates are limited by the low power components and central processing hardware used to implement the system, which typically consists of a low-power microcontroller (MCU). Currently, commercially available low power MCUs from companies such as Atmel and Microchip operate at clock frequencies up to around 80 MHz. Therefore, data rates of 160 MHz would potentially be the upper limit, but since processing communication involves many computations, and each computation requires many processor clock cycles, the data rate is usually much less than the main clock speed of the device. Thus, the limitation on data rates comes from the processor clock speed.

A FSO-based sensor network consists of multiple FSOC systems that can transmit data to and receive data from an end destination that is physically separated from the FSOC systems by a few meters. The actual range depends on the complexity of the transmitter and receiver architecture, which will be discussed in greater detail later. But for now, it can be assumed that the communication range is on the order of meters, rather than 10s of meters or 100s of meters. Due to the short-range (SR) nature of the links, the main loss in the optical channel results from the beam spreading, or diverging, as it propagates. The divergence is a property that can be controlled (up to the diffraction limit) by lenses. In the developed architecture, a

typical beam divergence is 10°. As the beam diverges, the energy gets spread out over a wider area, which results in a smaller intensity across the beam. Since the received power at the detector is proportional to the intensity multiplied by the area of the detector, the detector receives less power as the beam energy gets spread out over a wider area. One complication in creating a short-range FSO network arises from the fact that optical direct detection receivers are less sensitive than RF receivers [10]. Since direct detection is the only practical modulation technique for low power applications [1], optical communication requires a more complex transmitter and an increase in transmission power to provide the same amount of signal coverage as an RF transmitter.

1.3 – Advantages of Free Space Optics

There is growing interest in employing optical communications technology to create SR-FSOC sensor networks because such networks have the potential to increase bandwidth and provide more secure data communication compared to conventional RF based communication networks. Besides the potential for FSO to have superior data rates than RF, wireless security is probably the biggest advantage that optical communication has over radio frequency technology. Physical security and connectivity are critical components for a successful implementation of a wireless network, but there is an obvious tradeoff between the two: as connectivity increases, physical security decreases. This is simply due to the fact that to increase

connectivity, the network needs to broadcast the signal over a wider field of view making it easier for any device to detect the signal, which decreases the physical security of the signal. Wireless local area networks (WLANs), which use conventional RF technology, provide a great example of this tradeoff. When individuals turn on their personal laptop's network card, depending on how densely populated the area is, the individual will pick up multiple wireless network access points because RF communication can propagate over great distances and through walls and many other objects. With the communication being broadcast over a large area instead of only inside the individual's house, much of the signal transmission only serves to "pollute" the air space with RF signal and allows for potential unauthorized access onto the network. As a result, complex coding schemes are generally used to encrypt signals to prevent unauthorized access to the information travelling between the two communicating nodes. In contrast, optical signals cannot penetrate through most objects and walls, and even if the optical signal was to escape a room via a window, the signal will be lost within a short distance due to sunlight and other ambient lighting. Therefore, in a given infrastructure, where many rooms or compartments are adjacent to each other, each section can potentially sustain its own wireless network without signal interference from the other networks. Further data encryption can make each adjacent optical network extremely secure.

More specifically, FSOC systems are appropriate for applications where sensors need to transmit large amounts of data in short bursts. Because of the directional nature

of FSOC, spatial diversity schemes can be easily implemented allowing FSOC sensor networks to have the potential to provide applications with data rates that are scalable from b/s to hundreds of Mb/s. The increase in data rates is an important performance difference between an FSO and RF wireless network because it will allow the transfer of the same amount of information in shorter periods of time. Therefore, the sensors can provide faster real-time updates to the end destination. This is advantageous for applications where multiple sensors wake up simultaneously at critical moments and need to transfer data to the end destination. Furthermore, the increased data rates make short-range wireless transfer of high definition content more practical and attractive [3].

1.4 – Research Areas in Point to Multipoint SR-FSOC Designs

This section provides a general overview of the types of research and development underway in the realm of short-range FSOC network design. Much of the ongoing SR-FSOC research aims at solving the following major design challenges:

1. Connectivity – Creating and maintaining optical links in a dynamically changing environment
2. Sensitivity and Tracking– Minimizing transmission power and still maintaining high SNR at the receiver regardless of the receiver's orientation with respect to the transmitter

3. Scalability and Cost – Miniaturizing and integrating transceiver optics and electronics to create scalable multi-transmitter and multi-receiver architectures

4. Networking - Design and implementation of proper directional media access control protocols that are designed specific for SR-FSOC systems, rather than using RF-optimized protocols.

1.4.1 – Connectivity

In order for an indoor-based FSO network to have a high degree of connectivity, the FSOC system needs to provide signal coverage over a wide field of view (FOV) so that multiple devices can connect to the network regardless of their position in the room. There are two general ways to provide signal coverage over a FOV [1] [2] [11]: non line of sight (non-LOS) and line of sight (LOS) communication. In the non-LOS method, a single high power wide FOV laser beam or multiple low power narrow FOV laser beams reflect from walls and objects to cover the entire region with light. While this method can provide connectivity even if the receiver is not in LOS of the transmitter, the system requires large transmission power [12]. Furthermore, multi-path fading effects, which occur when transmitted light arrives at the receiver at different times, will decrease communication bandwidth by distorting the received signal [13]. This effect occurs because light that reflects off different objects and walls may arrive at the detector at different times due to the different travel paths. In the LOS method, multiple directional laser beams are pointed in different

directions to cover a large FOV. The transmission power requirements for such systems are much lower than the non-LOS of method since the light is being directly sent to the receiver without reflecting off a wall. However, high connectivity is not always guaranteed as links can be potentially blocked by a LOS obstruction. Both methods have been studied through numerous simulations and experiments to determine the received power at the detector and to model the infrared wireless channel in different room shapes and sizes [13] [1] [2] [10] [14] [15] [16] [17] [18].

1.4.2 – Sensitivity and Tracking

On the receiver side**,** there has been interest in the research and design of angularly diverse receivers (ADRs) [13] [1] [2] [19]. These receiver systems use multiple non-imaging optical concentrators pointed in different directions to concentrate light onto a photodiode placed at the back end of each concentrator. Each concentrator has a narrow FOV which helps to decrease multipath fading effects, since only a small number of signal paths can be detected by the concentrator's FOV. The non-imaging concentrator typically used in simulations and prototypes is the compound parabolic concentrator (CPC) [20] [21] because it has been shown to be beneficial in optical communication systems [22]. Specifically, light that is incident at the front aperture at angles smaller than the designed CPC maximum concentration angle, ϑ_{CPC}, will be concentrated and transmitted through the exit aperture. Transmission will abruptly drop to zero for light incident at angles greater than ϑ_{CPC} [20]**.** Because

of the CPC's insensitivity to angular fluctuations within its FOV, the CPC is an ideal candidate for an ADR design.

In [17], Carruthers and Khan demonstrated communication between a prototype multiple transmitter base station and a prototype angularly diverse receiver mobile station that achieves a data rate of 70 Mb/s up to 4.2 m in range. While the system performed up to its design parameters, the authors noted that the overall size of the ADR becomes a problem, because the size of a conventional CPC can become very large depending on the design parameters, such as the entrance aperture, exit aperture, and ϑ_{CPC} [20] [21]. Therefore, there has been considerable interest in trying to reduce the size of ADRs by investigating different shapes and sizes of non-imaging concentrators [23] and by replacing the multiple non-imaging concentrators with a single imaging concentrator, or imaging ADR [13].

In [13], Djahani and Kahn presented an analysis on link performance for an imaging ADR used in optical communications compared to the conventional photodiode receiver. The imaging ADR is an optical imaging concentrator that focuses light onto an imaging plane a focal distance behind the concentrator. This imaging plane is divided into an array of photodiodes, where each photodiode can be thought of as a pixel. Figure 1.1, which is taken directly from [13], illustrates the pixel arrangement of the ADR. The number of pixels is determined by the size of the photodiode.

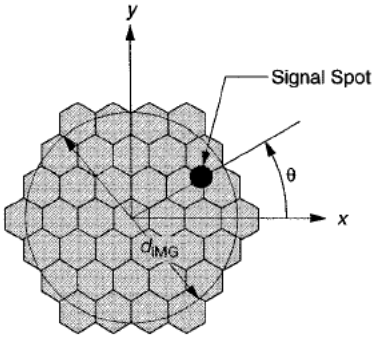


Figure 1.1- The image plane is divided into a number of photodiodes, or pixels, large enough to cover the area of the image spot. (Image taken directly from [13])

ADRs can also be used to implement an electronic tracking architecture so that a narrow LOS optical beam can remain focused upon its intended target [16] [24]. In [16], the authors presented a base station transceiver design that can switch between a highly directional tracked data link when LOS is available between the base station transmitter and terminal receiver, and a wide FOV diffuse link when LOS is not available. Their receiver architecture consists of an array of photodiodes, where each photodiode is located at a different spot in the image plane of the wide angle lens. From ray tracing, it can be shown that incident light will be focused to a specific spot on the image plane based on the angle of incidence. Therefore, each photodiode, or pixel, can only receive light from a certain FOV, which the authors define as a sector. If the incoming incident light shifts from one sector to another, there will be a corresponding shift in received intensity in the image plane. By monitoring which sector the incident light is coming from, the imaging ADR can provide the correct pointing information to the transmitter so that the link between the base station and the terminal remains intact. The transmitter architecture for their base station consists of an array of vertical-cavity surface emitting laser (VSCEL)

diodes. The output of each diode is sent through beam shaping optics so that the output profile from the diffuser array matches the sector defined by the receiver of the base station. With this setup, the transmitter can aim and transmit a single VCSEL directional beam to the same sector from which the incident light is coming from. If all the VSCELs transmit simultaneously, then the base station can provide signal coverage to all sectors in a non-directed non-LOS scheme.

In [25], a single photodiode receiver for electronic tracking was proposed. This receiver, called the single channel imaging receiver (SCIR), is different from an ADR in that there is only 1 photodetector in the image plane. Instead of using an array of photodiodes to cover the image plane, the single photodiode rests on a mechanical motion stage that provides translations in 2 dimensions. The concept behind this receiver is that strong and weak signals and ambient light spots will be imaged onto the imaging plane. By employing a search algorithm via a digital signal processor, the photodetector will be translated to the spot of maximum signal light. Once this signal spot is found on the imaging plane, any gradual changes in the angle of incidence can be tracked by the photodetector. The size of the photodetector can be chosen to be very small so that the field of view is also very small. In this case, multipath distortion, which is light collected from different paths, becomes negligible.

In efforts to minimize transmission power further, there has been interest in using retro-reflectors [26] [27]. In these network architectures, only the base station is

equipped with a transmitter. The other mobile nodes have retro-reflectors, which are devices that reflect incident light from the base station back to the base station. By modulating the retro-reflectors, a signal can be transmitted back to the base station. Therefore, the mobile devices do not need a transmitter at all, which reduces the total power consumption of the device. However, a downside is that these mobile devices cannot initiate communication since they can only transmit by reflecting incident light.

1.4.3 – Scalability

It can be inferred from the above descriptions that the design of short-range optical transceivers can involve many sophisticated components. In [27] [28] [29], O'Brien *et. al.* addressed the issue of whether there is a need to create integrated narrow FOV transceivers so that wide FOV transceiver systems can be constructed by simply scaling the number of integrated narrow FOV transmitters. This idea of scalability could lead to wide angle optical transceivers that are no larger than the sizes of standard surface mount electrical components, making them more attractive for potential applications. To this end, O'Brien *et. al.* designed a prototype integrated optical communication device with all electrical components implemented at the CMOS level to operate at 100 Mb/s [29]. The receiver contains seven detectors flip-chip bonded to arrays of custom CMOS receivers. The detectors are closely packed

in a hexagonal structure, shown in figure 1.2, to provide electronic tracking by monitoring the received signal from each detector.

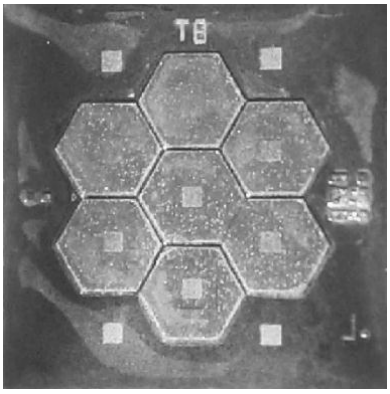


Figure 1.2- Seven receiver hexagonal structure. (Image taken directly from [29]**)**

While O'Brien *et. al.* demonstrated a fully integrated design, they noted that scalability is still a technical challenge. For instance, in their design each detector provides approximately 4.5° FOV coverage, so to cover a 90° FOV, it would take approximately 200 detectors (discussed in greater detail in chapter 2). Each detector requires its own amplification and detection circuitry, which when implemented has a slightly larger area than the detector itself. Therefore, increasing the number of detectors increases the size of the device considerably.

1.4.4 – Networking

Traditionally, FSO links have been used as point to point communication links over long distances. When moving towards point to multipoint communication, the dynamics change considerably. Now, different devices need to be in LOS of one another and proper communication protocols need to be implemented so that

devices do not interrupt one another. However, because FSO links have mainly been used as point to point communication over long distances, there has not been a lot of research and development into FSO-based point to multi-point protocols. Instead many of the FSOC networks that are being proposed and developed today use a collection of RF protocols to handle network communication [4] [6], which makes some sense because RF wireless technology has been around for longer than optical wireless. Furthermore, since many FSOC networks are in development stages, there have not been many experimental results or testing of network protocols in a FSOC networks.

The most common optical protocol is provided by the Infrared Data Association (IrDA), and is used in over 40 million new devices each year [30]. The IrDA 1.x protocol is designed to maintain point to point links at speeds up to 16 Mb/s over a 1 meter distance, with a cone half angle of 15°. One limitation of the IrDA 1.x standard is that it is not designed to handle point to multi-point links. Instead, IrDA has developed a multiple access control standard, called Advanced Infrared (AIr), which aims to provide multiple access functionality, as well as increase the field of view and range of signal coverage [31]. The AIr MAC uses RF-based protocols for multiple access control. A more in depth discussion on RF protocols will be presented in chapter 3.

1.5 – Dissertation Contributions

In order for directional FSOC sensor networks to become viable short-range solutions, the designed FSOC hardware must be inexpensive and capable of providing signal coverage over a wide field of view without strict optical alignment requirements. In addition, the networks need to operate with efficient media access protocols that are optimized for directional links to minimize random access times for the independent transmitting motes within the network. These challenges are the focus of this dissertation. In terms of the 4 major research areas discussed in section 1.4, this dissertation touches on connectivity and scalability but heavily focuses on the networking aspect of FSOC sensor networks.

This dissertation outlines the design and implementation of inexpensive SR-FSOC hardware capable of providing wide FOV signal coverage without using any sophisticated optical components. The designed hardware consists of multiple narrow beam transmitters, each pointed in a different direction. The transmitter sources are VCSELs, which can be driven by a 10 mA current to output 2.5 mW of power. The low current requirements allow for low-power microcontrollers to act as the modulator and driver for the transmitter.

In general, narrow optical beams used for FSOC require precise and complex pointing, acquisition, tracking and alignment methods. However, this dissertation addresses the challenge of alignment by using multiple narrow field of view (FOV) transmitters and a single, wide angle receiver. The developed optical transmitter

architecture consists of rings of multiple transmitters surrounding a photodiode for light collection. Each ring is tilted at a different angle so that a wide transmission FOV can be obtained. Depending on the number of transmitters and the transmitter's divergence angle, different FOVs can be tailored to fit the requirements of the target application.

The other major design challenge addressed in this dissertation is the networking aspect of FSOC sensor networks. Specifically, the networking aspect can be divided into two main parts: the network architecture and the media access control (MAC) protocol, which is designed based on the developed network architecture.

This dissertation presents a network architecture where communication takes place between a sensing device, or "mote", and a central control station, or "cluster head." In this type of architecture, the cluster head is designated as the master node and has the dual functionality of coordinating network traffic and aggregating data from all the slave nodes, or motes, that are within its FOV. The FOV of the cluster heads are not designed to be very wide so there may be motes outside of a cluster head's FOV; however, because cluster heads can form directional links and relay information between one another, the concept is that as long as a mote is connected to one cluster head, it will be connected to all. Figure 1.3 shows a 3d rendering of the developed network architecture.

Figure 1.3- 3D rendering of the network architecture.

The challenge in MAC protocol design lies in the fact that most nodes are blind to other nodes' transmissions because of their relatively narrow directional links. In relation to the developed architecture, if multiple motes try to transmit information to the same cluster head, collisions will occur. To minimize the chances of packet collisions, a directional MAC (DMAC) protocol designed for SR-FSOC sensor networks is presented and experimentally tested on a network of 10 motes and a cluster head. Specifically, the DMAC protocol directs motes to transmit their media access control (MAC) addresses to the cluster head whenever a mote needs to transmit data. The cluster head then reads the different MAC addresses and organizes time slots for the

different motes to transmit their data. The number and length of the time slots can be adjusted in real-time by adding and removing MAC addresses of different motes that need to transmit data. In this manner, the data throughput can remain near its peak value regardless of how many motes are transmitting.

In summary, this dissertation provides: 1) the design and implementation of small and inexpensive short-range FSOC systems that can be implemented using standard "off the shelf" components including a microcontroller and sensor device to form a complete standalone package; 2) development of a DMAC protocol that is optimized for the implemented FSOC system and target network applications; 3) network performance evaluation and optimization for the combined FSOC hardware, network architecture, and DMAC protocol. This is done through a series of hardware tests on an experimental prototype FSOC sensor network consisting of 10 motes and 1 cluster head and simulations of larger network sizes.

In general, much of the previous research in this field consists of numerical simulations of link budgets and preliminary prototype designs testing certain components of the FSOC hardware. There has not been as much work done in providing actual network implementations and in experimentally testing DMAC protocols for FSO based point to multi-point networks consisting of multiple nodes.

1.6 – Organization of Dissertation

The dissertation is arranged as follows. Chapter 2 discusses the transceiver architecture designed for the cluster head, and how the design can be altered to create different FOVs for different types of applications. Chapter 3 explains the network architecture and developed DMAC protocol for FSOC sensor networks in great detail. Chapter 4 presents the design and implementation of the developed SR-FSOC hardware. Chapter 5 shows experimental performance results of a SR-FSOC sensor network consisting of 1 cluster head and 10 motes.

Chapter 2 – Directional SR-FSOC Transceiver Design

2.1 – Introduction

This chapter presents the theoretical rationale and system design for constructing compact and simple FSOC transceivers for short-range sensor networking applications. It will be shown that these systems are practical communication solutions for today's compact and energy efficient portable devices and sensors.

In the field of SR-FSOC, there have been several proposed transceiver architectures, some of which use multiple transmitters aimed in different directions to broadcast a signal over a wide FOV. For this research project, a multi-transmitter architecture was selected, but the design aims to reduce the transmitter circuit complexity and size of the overall system. The reason for this is that portable electronics are becoming smaller, more energy efficient, and less expensive, and therefore, it is necessary to design the communication components in the same manner. The transceiver architecture can consist of a series of multi-transmitter rings, where each ring provides signal coverage over a specified FOV. By adding additional rings, the signal coverage FOV increases. The total number of rings depends on the size constraint of the transceiver and the angular tolerance for the intended application.

The following sections will provide the motivation for wide FOV signal coverage, the design challenges and choices, and the transceiver architecture for the SR-FSOC

system. The chapter concludes with experimental performance results from an implemented transceiver architecture.

2.2 – Multi-Transmitter Architecture

In order for FSOC to become viable short-range solutions in network applications, the FSOC transceiver must be able to broadcast and receive optical signals over a wide FOV without strict alignment requirements. This wide FOV requirement stems from practical considerations. First, the FSOC system should be designed so that deployment and network setup does not require any manual alignment; the system should automatically be able to pick up an incoming signal, determine which direction the signal is coming from, and then transmit another signal back in that direction. Second, the system needs to be designed to achieve point-to-multipoint communication so that multiple motes can be spread out over an area and still optically communicate with the cluster head. This necessitates that the cluster head be able to receive and transmit signals over a large FOV.

2.2.1 – Wide FOV Signal Coverage

In FSO communications, there are two general methods to provide signal coverage over a wide FOV. One method is to use a high-powered laser beam that relies on diffused reflections from walls and objects to cover the entire region with light.

Such a system requires 100s of milliwatts of transmitter power to provide useful amounts of power at the receiver (Rx). To generate this amount of power, the laser transmitters need to be driven by 100s of mA of current, which is impractical for a battery operated communication device since the required current is more than compact batteries can sustain. Another method is to use multiple narrow beam transmitters, each pointed in a different direction, to cover a large FOV. In this method, each transmitter covers a narrow field of view, so only a few milliwatts of transmission power would be required from each transmitter. If each transmitter is a vertical cavity surface emitting laser (VCSELs), these smaller amounts of transmission power can be sustained with drive currents less than 10 milliamps, making this a more practical strategy for a compact battery driven device. Because energy efficiency is an important parameter for a successful sensor network, the FSOC transmitter architecture described in this thesis consists of multiple VCSEL transmitters.

The major design consideration with this multi-transmitter approach is keeping the overall size small and electrical complexity of the transceiver simple. Each additional transmitter requires its own drive circuitry as well as extra physical space on a printed circuit board (PCB). However, each additional transmitter provides better signal coverage and reduces the need for manual alignment upon deployment of the system. The total number of transmitters depends on how much FOV is required for a specific application, as well as how much the laser beam's divergence angle, θ_B, which is defined as the angle where the laser beam's intensity distribution falls to

$1/e^2$ from its central (0˚) intensity. Let's assume that a hemisphere of signal coverage is required for a specific application. It turns out that this is a practical design choice because the FSO transceiver architecture will most likely be placed onto some sort of device, which would block the transceiver's FOV from behind the device. Thus, the transceiver would really only have a FOV normal to the device's surface.

$$N_{trans} = \frac{FOV}{2\pi(1 - \cos\theta_B)} \tag{2.1}$$

Equation 2.1 shows the relationship between the number of transmitters required, N_{trans}, to cover a hemisphere FOV (in steradians) based on θ_B. Figure 2.1 shows how the number of transmitters required to provide a FOV of 2π steradians decreases as θ_B increases. In this plot, there is no overlapping of the transmitter FOVs, which means that the N_{trans} value shown for a particular value of θ_B is the minimum number of transmitters required to cover the entire FOV. A more practical implementation would provide slight overlapping to ensure that no region between transmitter FOVs is left uncovered. Obviously, a larger FOV would provide better signal coverage and require less manual alignment, so selecting transmitters with large divergence angles would help to reduce the required numbers of transmitters. The drawback is that as θ_B increases, the range of the optical link is reduced because the energy of the laser beam spreads out over a larger area perpendicular to the

beam's propagation path. The diverging energy reduces the intensity of the beam, which decreases the amount of incident power collected at the receiver.

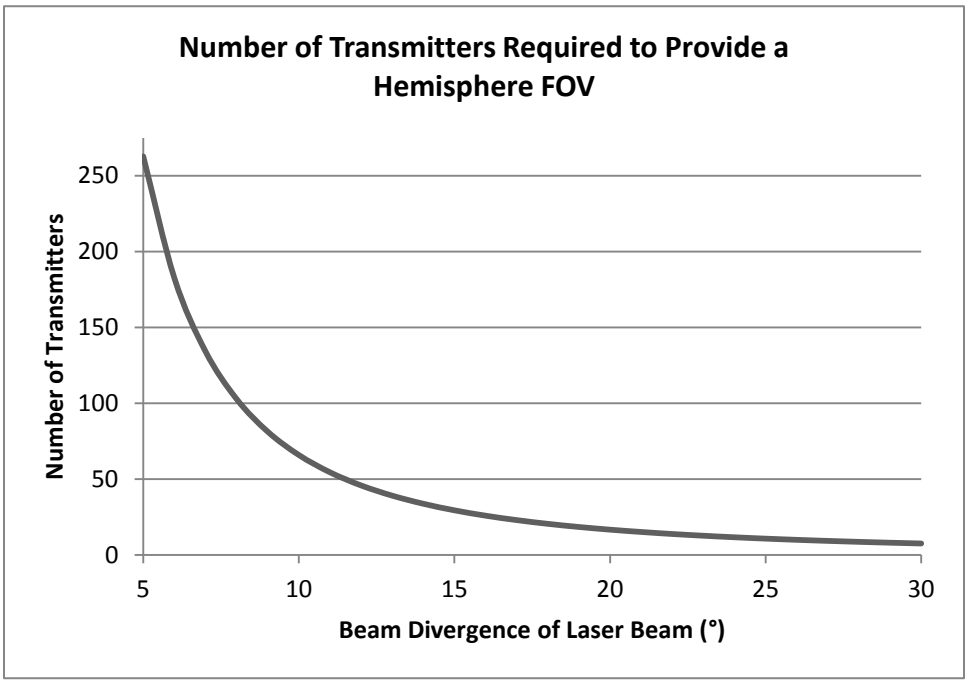


Figure 2.1 - Number of transmitters required to provide a hemisphere FOV.

2.2.2 – Numerical Computation of Link Range

This decrease in received power can be modeled in the following fashion. Let's consider a transmitter source placed above a photodetector, as shown in figure 2.2, where the Tx source and the photodetector are aligned along the *z* axis. The Tx source is transmitting power to the photodetector's surface. The intensity distribution of the laser beam can be written as,

$$\Psi(\theta, z) = I_o(z) * I(\theta) \tag{2.2}$$

where θ is the angle between the vector (shown in orange) from the Tx source to an infinitesimal element on the photodiode's surface and the surface normal vector of the Tx source (shown in white). The z-axis represents the distance along the optical axis normal from the Tx source of transmission. Equation 2.2 assumes that the angular variation in intensity is symmetric around the beam propagation axis, which is the *z* axis. The beam divergence angle is defined by

$$\theta_B = \frac{\lambda}{\pi w_o}. \tag{2.3}$$

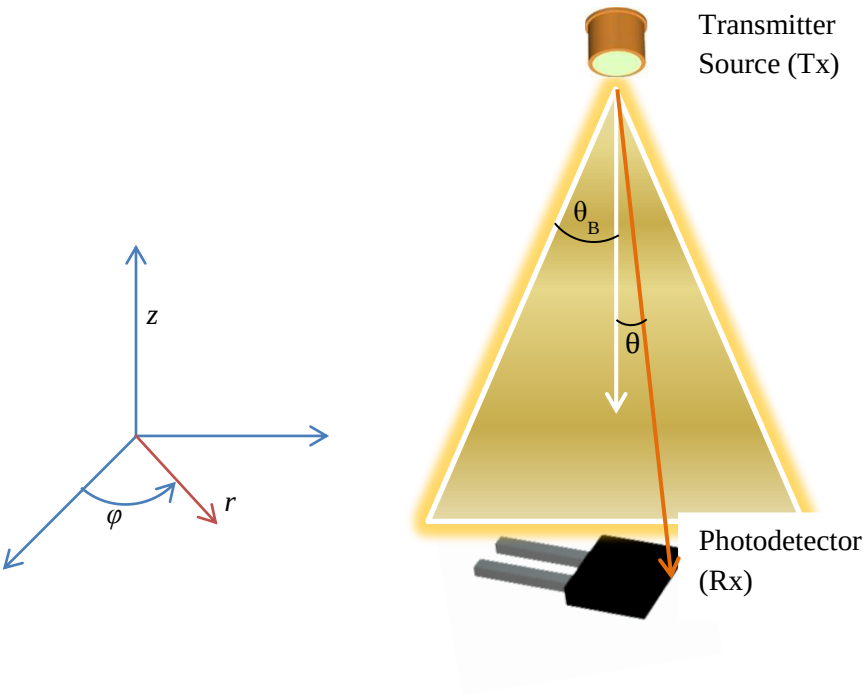


Figure 2.2 - Transmitter source with a beam divergence angle θ_B placed above a photodetector.

The parameters w_o and w are the minimum spot size of the transmitted beam and the spot size of the transmitted beam at a certain distance z from the laser output,

respectively. I_o is the axial intensity of the laser and depends on the distance *z*, because the beam diverges as it propagates.

For the following calculations the intensity distribution from each laser diode in the multi-beam transmitter system is modeled as a single mode Gaussian distribution [32],

$$\Psi(r,z) = I_o(z)e^{-\frac{2r^2}{w^2}}. \tag{2.4}$$

At a given distance, *z*, normal to the transmitter, there will be a spot size *w* and radius *r* which can be related to θ and *z* by

$$w = z\tan\theta_{B.} \tag{2.5}$$

and

$$r = z\tan\theta. \tag{2.6}$$

Substituting for *w* and *r* in terms of θ, *yields*

$$\Psi(\theta,z) = I_o(z)I(\theta) = I_o(z)e^{-\frac{2\tan^2\theta}{\tan^2\theta_B}}. \tag{2.7}$$

The intensity distribution can be approximately related to the laser diode power, P_t, by the function,

$$P_t = \iint \Psi(\theta, z) dA. \tag{2.8}$$

In this analysis, the curved phase front of the optical beam is approximated to be planar. From cylindrical coordinates,

$$dA = r dr d\varphi. \tag{2.9}$$

Differentiating equation 2.6 with respect to θ,

$$dr = z \sec^2 \theta \, d\theta \tag{2.10}$$

and substituting 2.10 and 2.6 into 2.9,

$$dA = z^2 \tan\theta \sec^2 \theta \, d\theta d\varphi. \tag{2.11}$$

Substituting 2.11 into 2.8 and integrating over φ yields the integral,

$$P_t = 2\pi I_o(z) z^2 \int_0^{\frac{\pi}{2}} I(\theta) \tan\theta \sec^2 \theta d\theta \, . \tag{2.12}$$

Since P_t is a known quantity, $I_o(z)$ can be numerically computed for a particular z value once the intensity distribution, $I(\theta)$, is known. However, for the case of a single mode Gaussian beam intensity distribution shown in equation 2.7, a closed form solution for I_o can be derived.

$$P_t = 2\pi I_o(z) z^2 \int_0^{\frac{\pi}{2}} e^{-\frac{2\tan^2\theta}{\tan^2\theta_B}} \tan\theta \sec^2\theta d\theta \,. \tag{2.13}$$

Using the following substitutions and then carrying out the required integration,

$$u = \tan^2\theta \tag{2.14}$$

$$du = 2\ \tan\theta \sec^2\theta\, d\theta \tag{2.15}$$

$$P_t = \pi I_o(z) z^2 \int_0^{\frac{\pi}{2}} e^{-\frac{2u}{\tan^2\theta_B}} du \tag{2.16}$$

$$I_o(z) = \frac{2P_t}{\pi z^2 \tan^2\theta_B} \tag{2.17}$$

The intensity distribution used to model the link between a transmitter and receiver can be written as

$$\Psi(\theta, z) = \frac{2P_t}{\pi z^2 \tan^2\theta_B} e^{-\frac{2\tan^2\theta}{\tan^2\theta_B}} \tag{2.18}$$

The power received at the detector is

$$P_{in} = \iint \Psi(\theta, z) dA \tag{2.19}$$

$$P_{in} = \frac{4P_t}{\tan^2 \theta_B} \int_0^{\theta_{max}} e^{-\frac{2\tan^2 \theta}{\tan^2 \theta_B}} \tan\theta \sec^2 \theta \, d\theta \qquad (2.20)$$

where θ_{max} is the largest angle of detected light between the incident light from the transmitter beam and the surface normal of the photodetector.

Figure 2.3 shows the results of numerically integrating equation 2.20 for various beam divergence angles using a photodetector with a circular detection area of 26.408 mm^2 (radius 2.9 mm) and a 2.5 mW transmitter. The radius was set to 2.9 mm so that the overall area of the simulated photodetector would match the overall area of the actual photodetector used, which had a rectangular detection area of 26.4 mm^2 (5.5 mm x 4.8 mm).

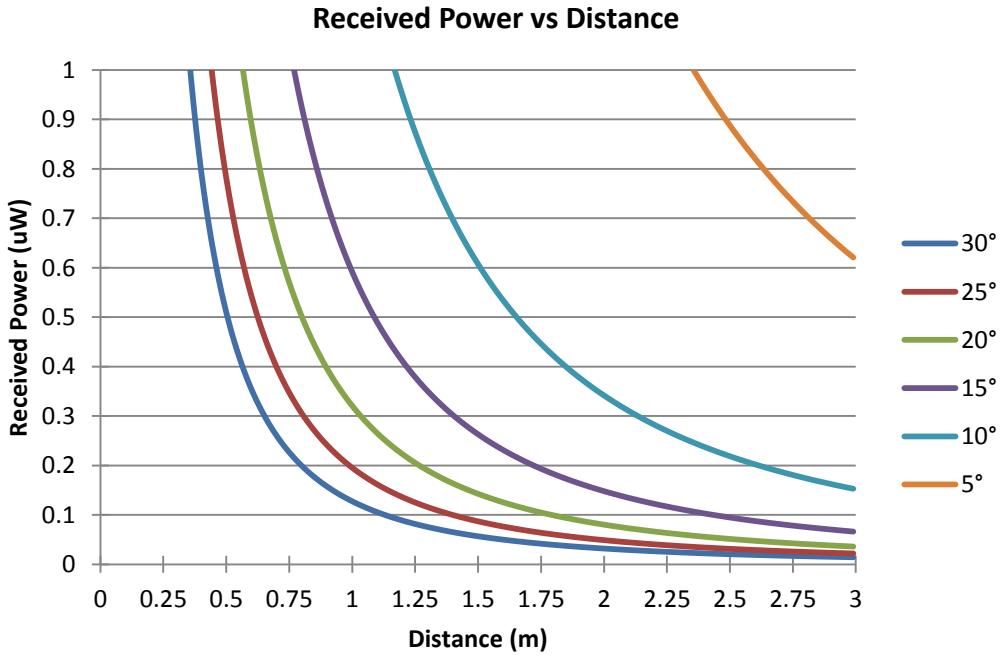


Figure 2.3 - Received power vs. distance from a 2.5 mW Tx source with different divergence angles.

Figure 2.3 shows how the received power decreases for increasing z distances between the transmitter and receiver for several different beam divergence angles, θ_B, which are shown in the right hand side of figure 2.3. In this plot, the transmitter is transmitting 2.5 mW of power. Due to the divergence angle of the beam, the beam spreads as it propagates and at distances of 1-3 m, only uW are detected at the receiver. The beam that has the smallest divergence angle provides more at the receiver for all link distances, because the beam spreads out the least as it propagates. As the divergence angle increases, the power received at the distance decreases. However, as mentioned in section 2.2.1, with larger Tx divergence angles, fewer transmitters are required to cover wide fields of view. Thus, there is a tradeoff between beam divergence and link distance.

2.2.3 – Multi-Transmitter Design

As discussed in chapter 1, the objective in designing a short-range transceiver is to reduce the complexity of the system, while still demonstrating a versatile system capable of handling point to multipoint communications for specific applications over link distances of 1-2 m. Using Tx sources with smaller than 10° beam divergences would essentially require many transmitters and drive circuitries to cover a wide field of view, as shown in figure 2.1. Conversely, using Tx sources with larger than 10° beam divergence angles drastically reduces the received power at the detector at longer ranges, as shown in figure 2.3. With these considerations in

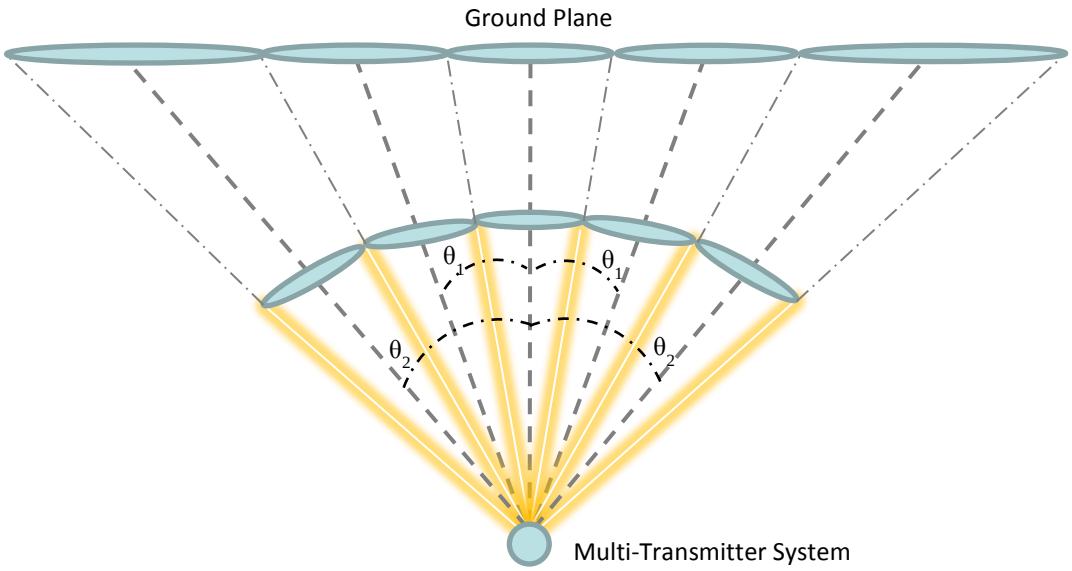
mind, the multi-transmitter architecture was designed for laser beams with beam divergence angles close to 10°.

To provide a wide field of view of signal coverage, a multi-transmitter system based on concentric rings was developed. In this architecture, an array of transmitters is positioned within each ring. Each transmitter is tilted by the same angle but pointed in a different direction to provide signal coverage over a different FOV. Each additional concentric ring provides additional transmitters that can increase the overall transmission signal FOV. In this manner, based on the divergence angle, tilt angle of the ring, and number of concentric rings, differently sized signal coverage fields of view can be designed. For applications that do not need a full semisphere of signal coverage, fewer rings would be needed. Conversely, if a large field of view is needed, more rings can be added. Each additional ring will require more transmitters than the previous ring, so larger fields of view will require more complex and larger transmitter architectures.

To generate the multi-transmitter configuration, an algorithm was implemented in MATLAB that generates ring-arrays of transmitters and then graphically displays the spot sizes of the different transmitter beams on a ground plane 1 meter away. The spot sizes correspond to the width between the $1/e^2$ intensity points of a single mode Gaussian laser profile. The variable input parameters to the algorithm are the number of concentric rings, the tilt angles of each ring, and the physical dimensions that each ring would require in order to provide space for commercially available Tx

sources to be positioned in. The preset input parameters are the Tx source beam divergence angle, which is set to 10°, and the perpendicular distance between the Tx source and the ground plane, which is set to 1 m. Ensuring full signal coverage at the 1 m ground plane guarantees full signal coverage up to 2-3 meters over the entire FOV of the multi-ring transmitter. Figure 2.4 graphically shows this by displaying the projections of the different spot sizes onto the ground plane. If the ground plane is covered, then the signal coverage is guaranteed over the entire FOV for a range of 2-3 meters as long as enough power is transmitted by the Tx source.

Ensuring Spot Size Signal Coverage on Ground Plane

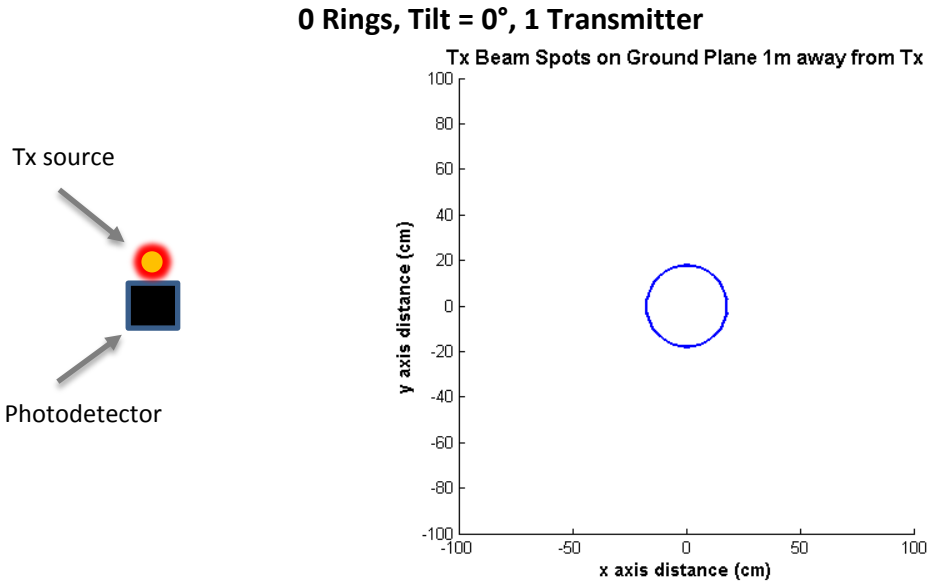


Cross Section of a 2 Ring System
(5 Transmitters Shown)

Figure 2.4 - Graphical display of the projected Tx spot sizes onto the ground plane.

The following shows an example of how a 2-ring multi-transmitter system is generated. A discussion of how an optimized *n*-ring configuration is designed will follow in section 2.2.4.

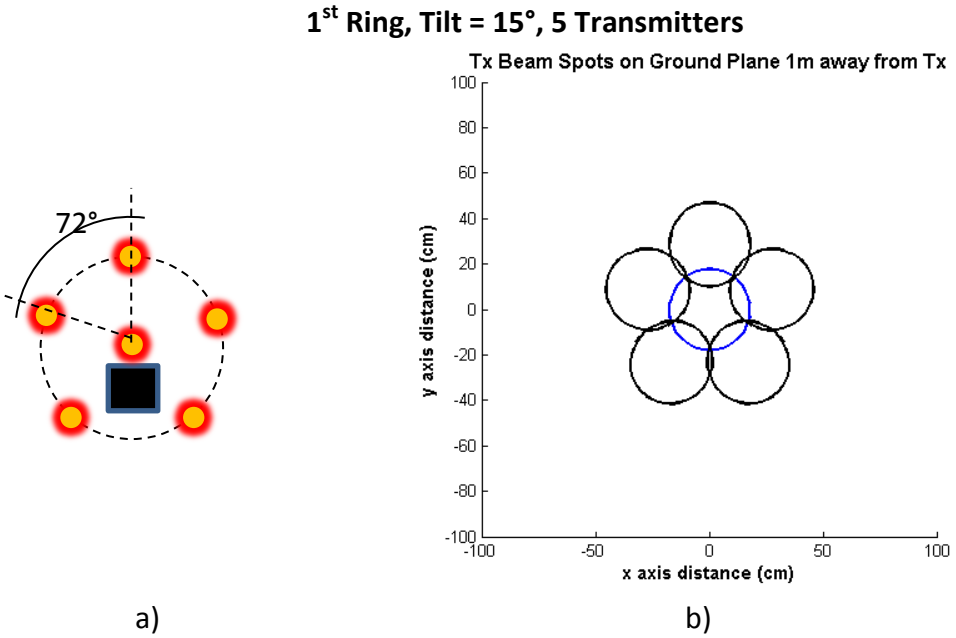
1. An initial transceiver is created with a single Tx source and photodiode. They are positioned next to each other, as shown in figure 2.5(a). Figure 2.5(b) shows the single spot size created on a ground plane 1 meter away from the Tx source. The Tx source is not tilted at all.



a) b)

Figure 2.5 - a) Single transmitter transceiver architecture. b) Single spot size projected onto ground plane.

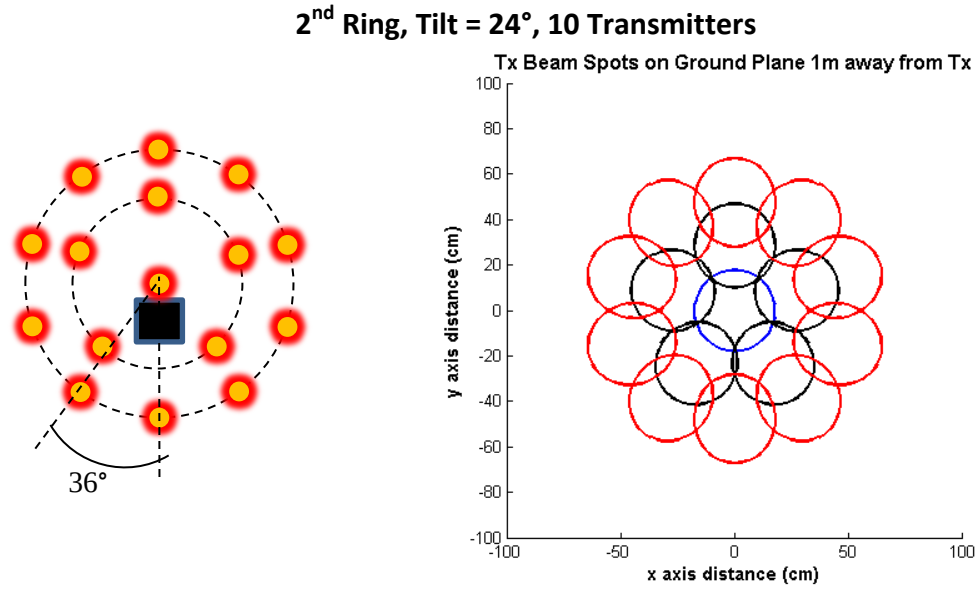
2. A new ring is added around the existing Tx source and photodiode. Each transmitter is tilted by a certain angle and pointed in a different direction so that its spot size covers a different area on the ground plane. The transmitters within this new ring are spaced out by a specific arc length to minimize overlapping in their spot sizes. Figure 2.6(a) shows the updated transceiver configuration, while figure 2.6(b) shows the increase in signal coverage by the transmitter spot sizes.



a) b)

Figure 2.6 - a) 1-ring transmitter ring transceiver architecture with tilt angle 15°. b) Projected spot sizes onto ground plane.

3. Once again, another ring is added in the same manner as described in step 2. Figure 2.7(a) shows the updated transceiver configuration, while figure 2.7(b) shows the increase in signal coverage by the transmitter spot sizes.



a) b)

Figure 2.7 - a) 2-ring transmitter transceiver architecture with tilt angles 24°, 15°. b) Projected spot sizes onto ground plane.

4. The algorithm continues to add additional rings of transmitters in the same fashion until the desired FOV is obtained.

As can be seen from figure 2.7(b), the ring of transmitters generates overlapping spot sizes whose centers lie along a circle. Increasing the number of transmitters within a ring causes more overlap between adjacent transmitter's spot sizes, which increases the overall signal coverage area, but there comes a point where the increase in transmitter complexity does not significantly increase signal coverage. At this point, the amount of transmitters within the ring constitutes a sufficient configuration, even if the overall spot size area contains a few "dead" spots. A "dead" spot is an area on the ground plane that is not contained within any of the transmitters' spot sizes. Thus, for this multi-transmitter design, the optimal number of transmitters for a specific ring is found by examining the percent increase in signal coverage area for each additional transmitter.

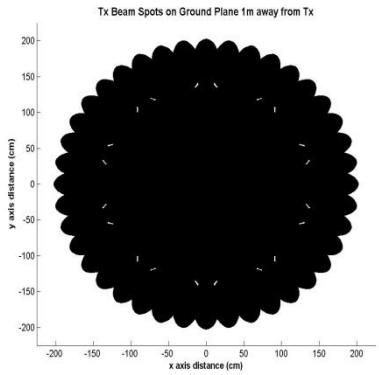
2.2.4 – Optimizing Multi-Transmitter Design

A second MATLAB algorithm was developed to optimize the *n*-ring transmitter configuration. In general, the algorithm worked by computing the optimum number of transmitters and tilt angle for the 1st ring. With the first ring in place, the algorithm worked to compute the optimum number of transmitters and tilt angle for the 2nd ring. This method continued until the desired FOV was obtained. Specifically, the algorithm computed the number of pixels within the ground plane

spot sizes of each *n*-ring structure for different numbers of transmitters within each ring. Once computed, the rate of signal coverage increase per additional transmitter could be analyzed, and based on this rate, an optimum configuration was selected.

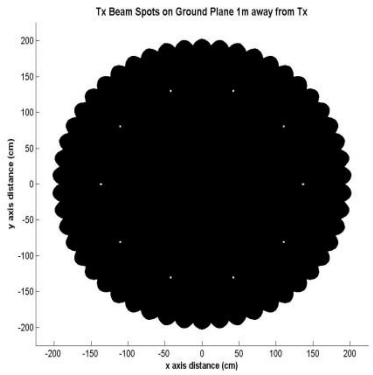
Figure 2.8(a-d) shows a few of the ground plane signal coverage areas from a 5-ring transmitter design where the 5th ring contains 40, 50, 55, and 60 transmitters. In the arrays of 40, 50, and 55 transmitters, several "dead" spots can be seen on the ground plane 1 meter away*, but none for the array of 60 transmitters. Each image in MATLAB consists of 1,080,000 pixels (1200x900).

5th Ring: 40 Transmitters



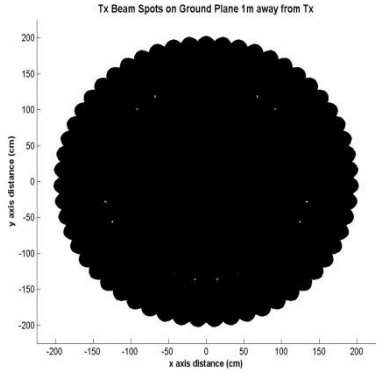
a)

5th Ring: 50 Transmitters



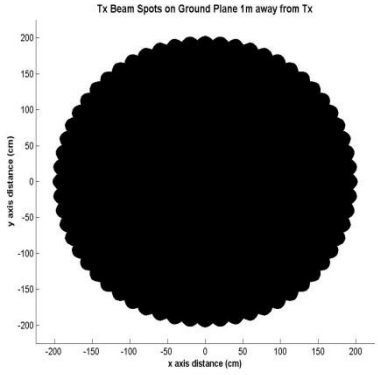
b)

5th Ring: 55 Transmitters



c)

5th Ring: 60 Transmitters



d)

Figure 2.8 - Ground plane signal coverage of a 5-ring transceiver architecture with different numbers of transmitters. a)40 b)50 c)55 d)60

To determine the optimal number of transmitters for this 5th ring, the percent increase in signal coverage for additional transmitters was plotted, as shown in figure 2.9. A threshold of at least .25% signal coverage increase per additional transmitter was used for the 5th ring. From figure 2.9, it can be seen that percent increase in signal coverage significantly drops after 45 transmitters. At 45 transmitters, the value is ~.279%. Based on this information, a ring design with 45 transmitters was decided to be optimal in terms of architecture complexity and signal coverage for a 5-ring configuration. Using this similar procedure for all other rings, the optimal parameters for a multi-transmitter structure up to 6 rings were determined.

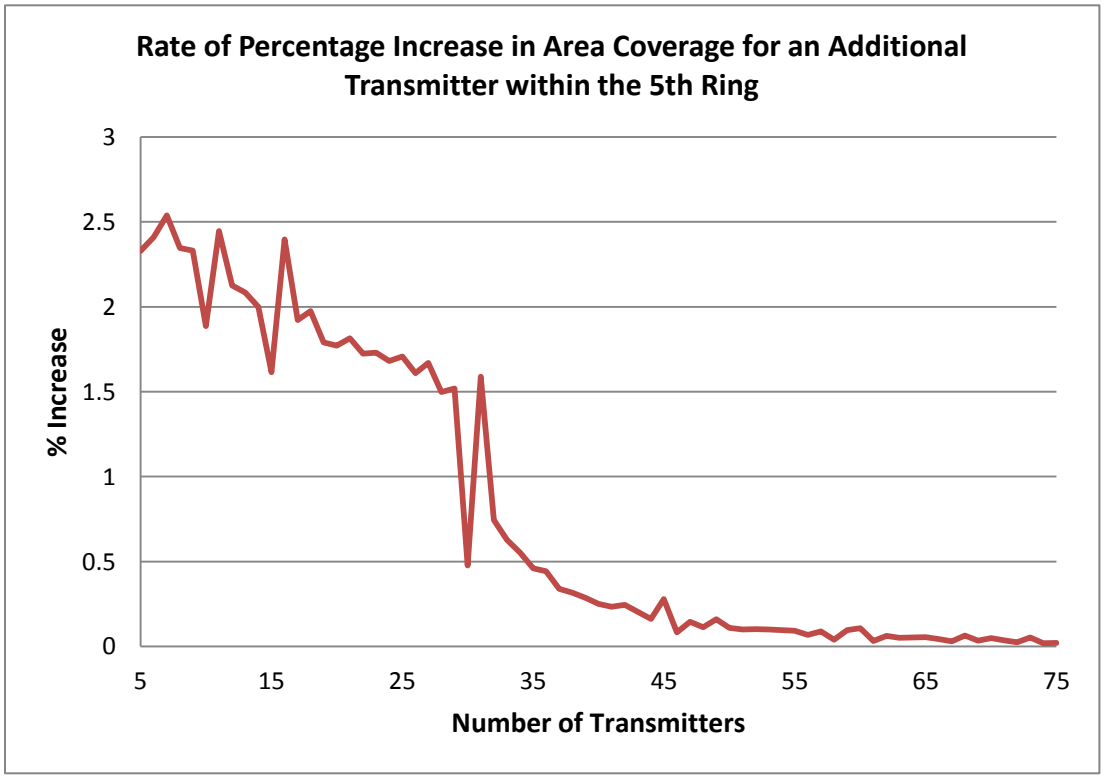


Figure 2.9 - Rate of percentage increase in ground plane area coverage for an additional transmitter within the 5th ring.

2.2.5 – Optimal Ring Structure

The optimal tilt angle and number of transmitters for the different rings of a 6-ring transmitter structure are shown in figure 2.10. The y-axis corresponds to the tilt angle in degrees with respect to the blue curve and to the number of transmitters with respect to the green curve. The reason for not generating structures with greater than 6 rings is because the transmitter count increased too much. Even for a 6 ring structure, the amount of transmitters is significantly high. Since this design is intended for use with commercially available semiconductor lasers that have a 5.4 mm diameter, the overall size of such a multi-transmitter design would be rather large and impractical for a small and low-complexity short-range FSO transceiver. Therefore, if a wide FOV is needed for an application, a better solution would be to use several 2-ring or 3-ring structures, and have them point in different directions. In this method, since both ring structure have their own photodetector, 2 optical channels could exist at once allowing for higher peak data rates with the use of spatial modulation.

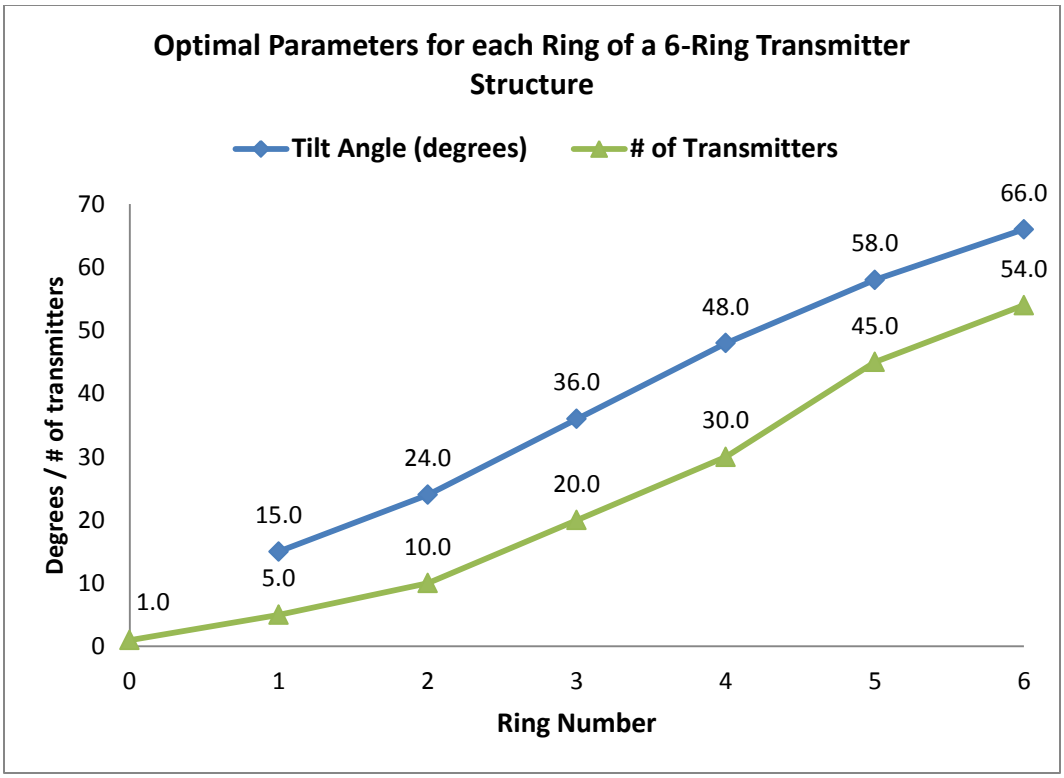


Figure 2.10 - The optimal tilt angle and number of transmitters for the different rings of a 6-ring transmitter structure.

To get a sense of how effective each ring structure is, an effective half cone angle, Ω_{EFF}, was calculated for each ring structure and is given by

$$\Omega_{EFF} = \tan^{-1}\frac{r_{eff}}{h}, \tag{2.21}$$

where r_{eff} is the largest radius circle (neglecting tiny "dead" spots) that is completely contained within the ground plane spot sizes, and h is the 1 meter distance between the ground plane and transmitter rings. The Ω_{EFF} for different rings of a 6-ring transmitter are shown in figure 2.11, along with the total number of transmitters required for that structure. The y-axis corresponds to the half-angle FOV in degrees with respect to blue curve and to the total number of transmitters with respect to the red curve. In terms of FOV coverage vs. required transmitters, multi-transmitter

designs with up to 3 rings are the most cost effective. Using more than 3 rings requires a large amount of transmitters and does not provide a substantial increase in signal FOV coverage.

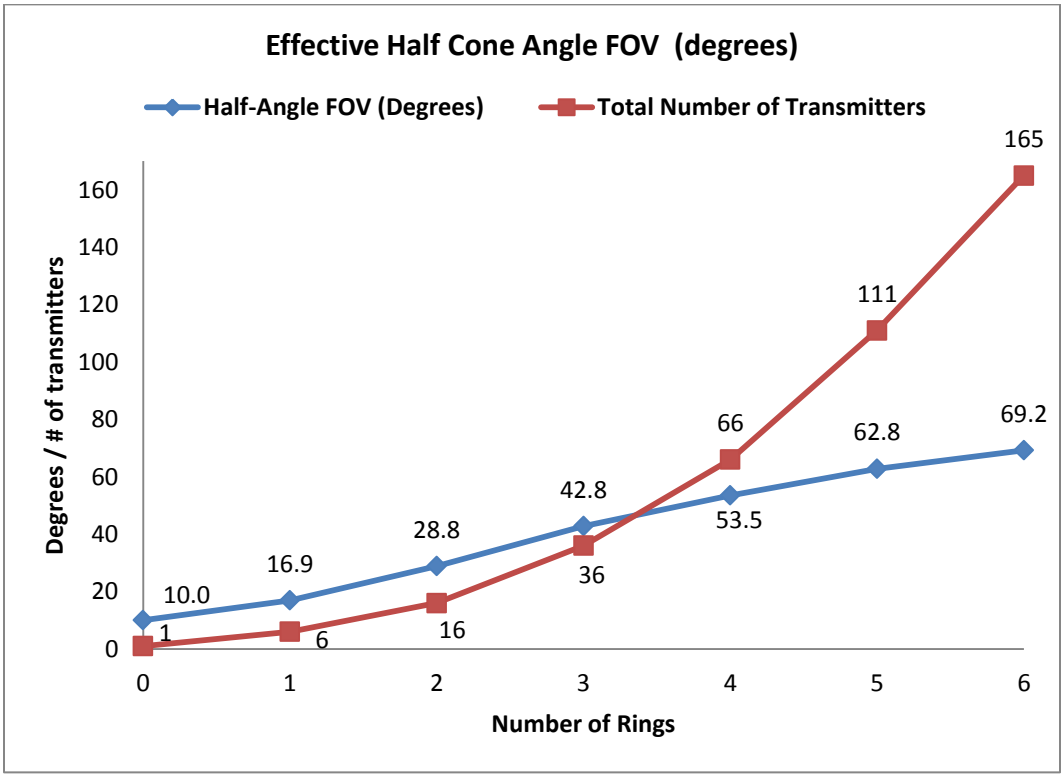


Figure 2.11 - The effective half cone angle FOV for different rings of a 6-ring transceiver.

2.3 Multi-Transmitter Experimental Performance

To test the signal coverage of this multi-transmitter design concept, a single ring transceiver was constructed. Five 980 nm VCSEL transmitters were used as the Tx sources. The VCSELs were purchased from Thorlabs, and were rated to have a beam divergence close to 10°. The ring, shown in figure 2.12, was constructed with a 15° slope, which provided a 15° tilt for each transmitter. The material used was Teflon.

Figure 2.12 - Picture of the actual machined transmitter ring holder. The tilt angle was machined to approximately 15°.

Figure 2.13 shows the schematic and renderings of the top and side views of the developed transceiver. Each VCSEL was separated by 72° arc length from one another. In this design, the center VCSEL was not included so that all spot size measurements at a ground plane could be attributed to the 1st ring of transmitters.

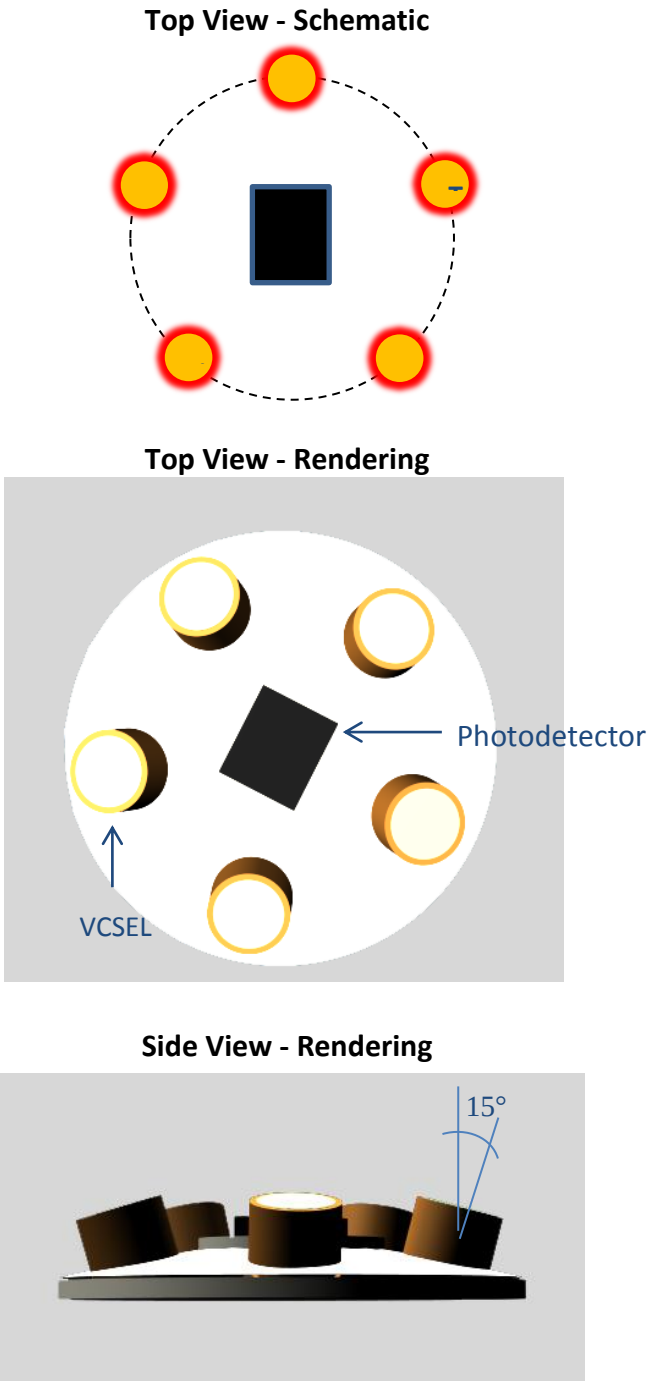
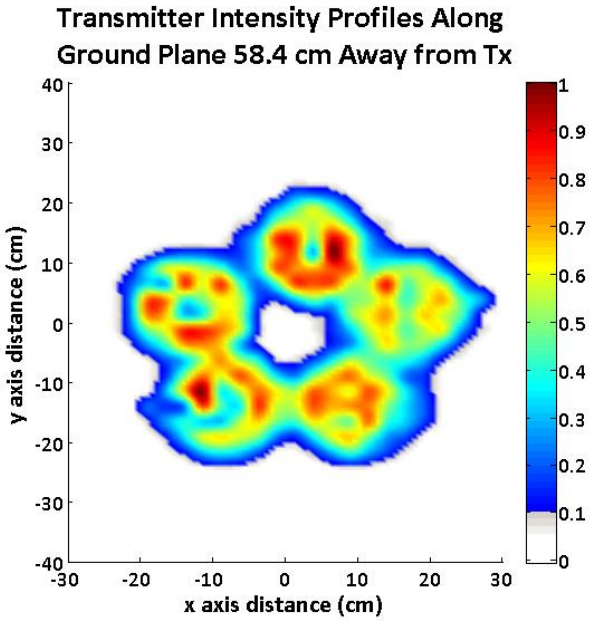


Figure 2.13 - Schematic and renderings of the top and side views of the developed transceiver.

The transceiver was positioned at a height of .584 m above the ground plane. It was pointed directly downward to the ground plane where a 5.5 mm x 4.8 mm photodiode measured the received intensity within a 55 cm x 55 cm square area in ~1.27 cm increments. Figure 2.14(a) shows the measured intensity profiles along

the ground plane, figure 2.14(b) lists the actual spot sizes measured for each intensity profile, and figure 2.15 shows the simulated spot sizes, which were in close agreement to the experimentally measured spot sizes.



Spot Sizes

Tx	X axis (cm)	Y axis (cm)
1	21.0	20.3
2	21.0	19.7
3	20.3	20.3
4	19.1	20.3
5	18.6	19.1

a) b)

Figure 2.14 - a) Picture of measured intensity fluctuations 58.4 cm away from designed single-ring transmitter source. b) Measured spot sizes.

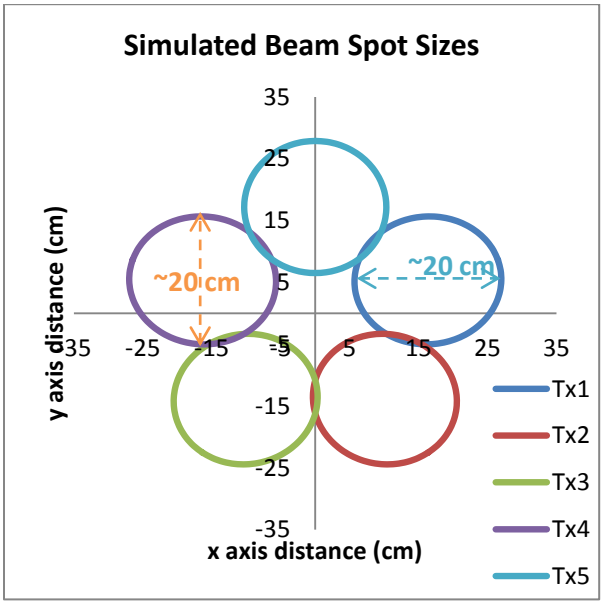


Figure 2.15 - Simulated transmitter beam spot sizes from a single-ring transmitter architecture on a ground plane 58.4 cm away from the transmitter.

The maximum measured intensity from the VCSEL did not occur in the center, but around the center. This is believed to have occurred because the VCSEL is multimode and because it is transmitting through a cylindrical metal housing. To get a sense of the distribution, images of the VCSEL intensity profile were captured at increasing distances. Figure 2.16 shows that the VCSEL intensity profile is weaker at the center as it propagates over a distance.

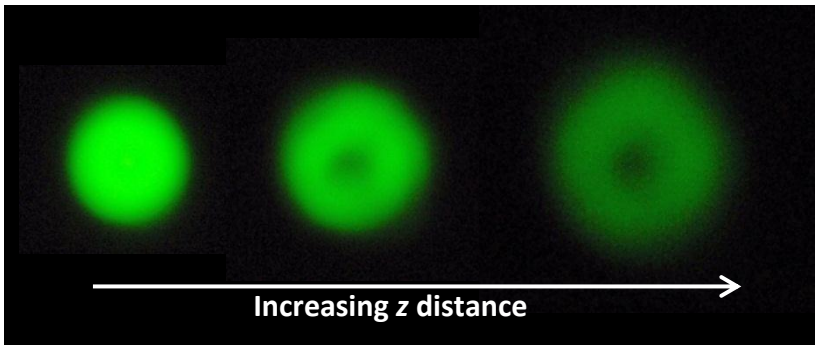


Figure 2.16 - Photographed VCSEL intensity profile as a function of distance.

Even though the VCSEL did not have a single mode Gaussian distribution, the measured and simulated spot sizes were in close agreement. Thus, the experimental result provided a confirmation that the simulated coverage and spot size results were accurate, and that when designing 2-ring or 3-ring structures, the results from figures 2.10 and 2.11 can be applied.

Chapter 3 – Directional SR-FSOC Network Architecture Design

3.1 – Introduction

The focus of this chapter is to present a network architecture that is designed and optimized for a short-range FSO directional wireless network. The objective of the architecture is to provide a framework that allows directional optics to become a practical solution for point to multipoint communication, instead of remaining a technology that is mostly used for long distance point to point communication [References]. Because directional links are being used, most nodes within a network will be unaware of any communication. Therefore, an optimized FSOC media access control (MAC) protocol has been developed to provide unique addresses to all nodes along with optimized multiple access protocols to handle channel access and contention. The multiple access protocol developed within this architecture is tailored to fit directional optics, so that it can maximize the effective data rate, reduce the number of packet collisions, reduce the amount of power consumption, and provide autonomous configuration capability.

The chapter is organized as follows: section 3.2 depicts the configuration of a short range directional FSO network and discusses how the network architecture handles communication between the different nodes; section 3.3 discusses different types of multiple access protocols that have been developed for omnidirectional

communication; and section 3.4 presents the multiple access protocol modified and tailored for directional optics.

3.2 – Defining the SR-FSOC Directional Network

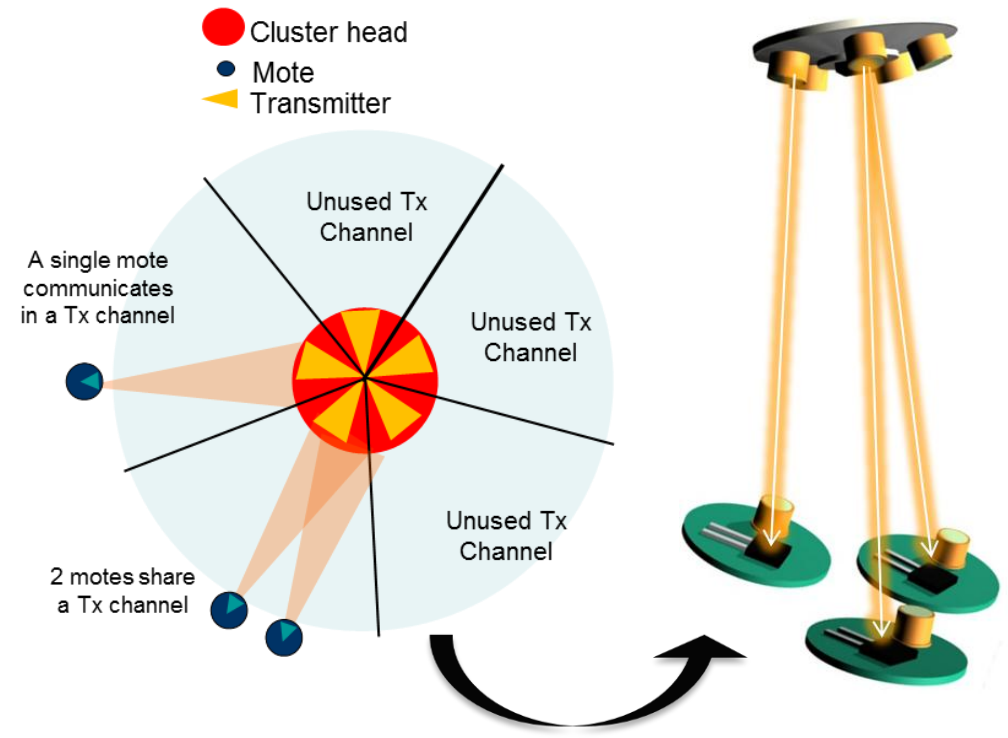
3.2.1 – Network Architecture: Cluster Heads and Motes

In this short-range FSO directional network, there are two distinct types of nodes: cluster heads and motes. Both the cluster heads and motes are modeled to transmit directionally using beams with 10° beam divergence angles; however, the cluster head has a larger FOV than the mote because it is modeled with a *n*-ring transmitter structure, while the mote is modeled with a single transmitter (0-ring structure). The larger FOV allows the cluster head to obtain LOS links with multiple motes that are within its FOV. Because the motes transmit over a narrower field of view, their placement in the network plays a role in order for them to obtain a LOS link. In this network architecture, it will be assumed that in deployment of such a network, the motes will be pointed towards a cluster head. Based on the target applications discussed in chapter 1, this is a reasonable assumption.

Based on these physical parameters of the nodes, a master-slave network architecture was developed for this FSO directional network. The cluster head serves as the master node because it has a larger field of view and can communicate to all the motes at once, and the motes serve as the slave nodes.

The roles of these two types of nodes are very different. The cluster heads coordinate network traffic between the different motes to minimize cross talk and aggregate data from the different motes within their field of view. The motes serve as the data nodes. They are assumed to be connected to a sensor or portable electronic device that needs to transmit data. When data transfer is not needed, the motes enter sleep mode to conserve energy. The cluster heads are assumed to be located on a base station of some sort that serves as the end destination for the data.

Figure 3.1 shows a schematic representation of a cluster head and group of motes and a 3d rendering of how the schematic correlates to the actual hardware developed. The cluster head is shown with 5 transmitters pointing in different directions, which corresponds to a 1-ring transmitter structure as discussed in chapter 2 with the center transmitter missing. The motes have a single transmitter, corresponding to the 0-ring transmitter structure. Because the cluster head has a 1-ring transmitter structure, it can maintain a larger transmission FOV than the motes, allowing multiple motes to be positioned anywhere within this FOV. In this network configuration, 3 motes are connected with LOS links to the cluster head, forming a network cluster.



Cluster head (Master node)
-Coordinates network traffic
-Aggregates data from all motes

Mote (Slave node)
-Transmits data to cluster head
-Enters sleep mode when not transmitting

Figure 3.1 - Schematic and 3d rendering of a cluster head and group of motes.

3.2.2 – FSO Directional Backbone

Because the motes have a narrow field of view, there may be situations in certain networks where some motes are not able to be pointed in the general direction of the cluster head, so they are unable to connect to the network cluster. In these cases, additional cluster heads can be deployed in specific locations, so that they can relay the signal from these motes to the other cluster head via a directional backbone link formed between the two cluster heads. Figure 3.2 illustrates this

concept. In this example, there are 2 network clusters connected by a backbone link.

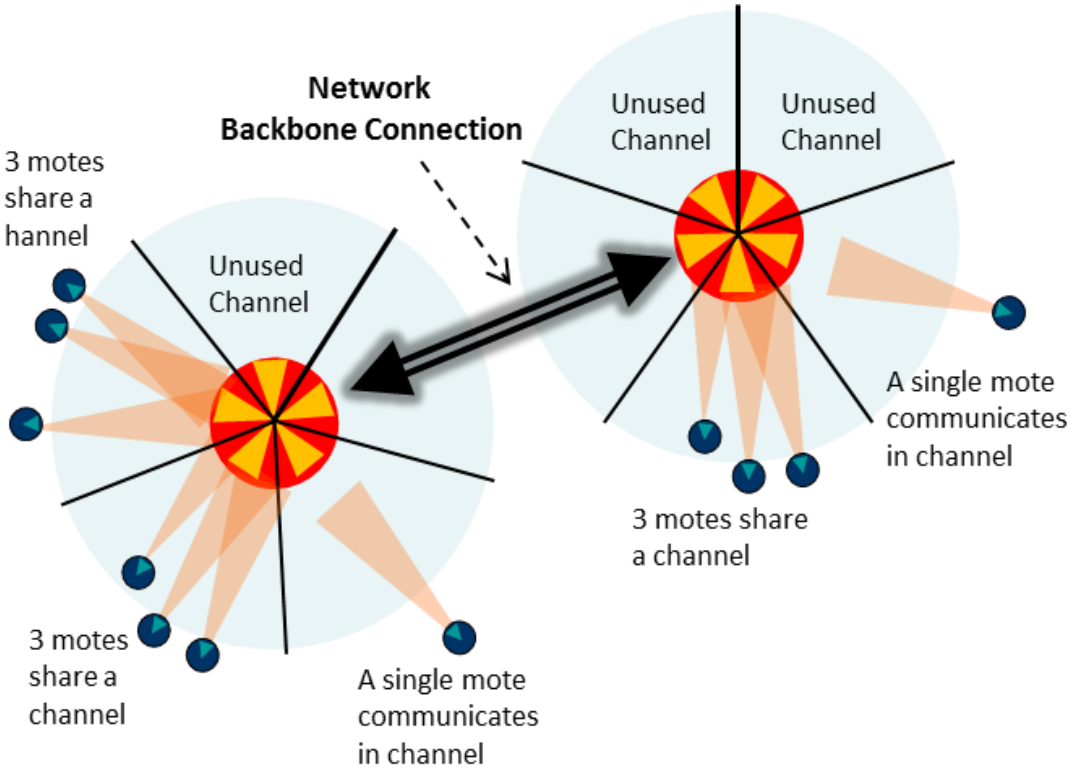


Figure 3.2 - Illustration of two network clusters connected via a backbone link.

Because of transceiver limitations, the master-slave links in this network can only extend a few meters. This backbone concept can be used to extend the range of the network so that motes far from the main cluster head can transmit their data through a series of hops between cluster heads that are linked via backbone links. Figure 3.3 displays an example configuration of such a network where several cluster heads are all connected via backbone links.

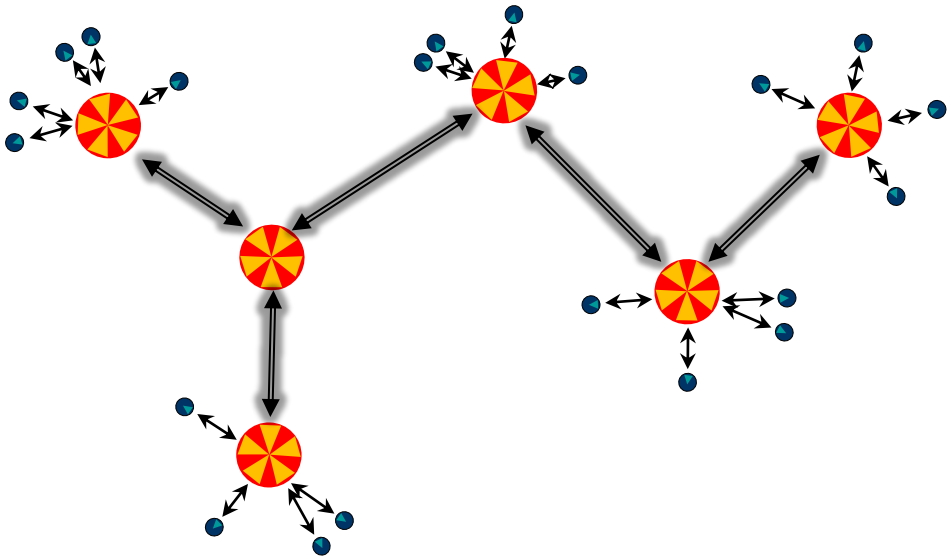


Figure 3.3 - Network configuration example where several cluster heads are connected via backbone links.

3.3 – Background Information on Existing Multiple Access Protocols

In point-to-multipoint networks, there are many nodes that may need to transmit at any given time, so proper protocols that allow multiple nodes to communicate are required to ensure successful communication. These types of protocols are called multiple access protocols. These protocols operate by providing an organized manner in which nodes can communicate with one another. However, sometimes several nodes may communicate simultaneously to another node causing packet collisions at the destination node. In these situations, a more specialized class of protocols that can handle random access is required. These protocols are known as random access (RA) protocols, and are a class of protocols within multiple access protocols. In RF communication, there have been many types of RA protocols

proposed and studied [33] [34] [35] [36] [37] [38] [39]. Many of these protocols are developed from a common set of underlying protocols. These underlying protocols are important to this research and are presented in the following sections.

3.3.1 – Time Division Multiple Access

Time division multiple access (TDMA) [33] [39] is a class of multiple access protocols where a control node sets up separate time slots for all nodes in the network to communicate via a synchronization signal. Whenever a node needs to transmit data, it sends its data within its designated time slot. A drawback to TDMA is that the overall data rate will fall if any node, or mote, does not need to transmit during its interval. During the course of communication in a practical network, different nodes may cease transmission and enter sleep mode while other nodes may wake and begin transmission. Therefore, a more efficient scheme would be one where the control node can dynamically reconfigure the time slots in real-time to add nodes that need to transmit and drop nodes that are finished transmitting. In this manner, this dynamic TDMA scheme (D-TDMA) can allow a system to maintain its overall peak data rate. Figure 3.4 illustrates the differences between TDMA and D-TDMA.

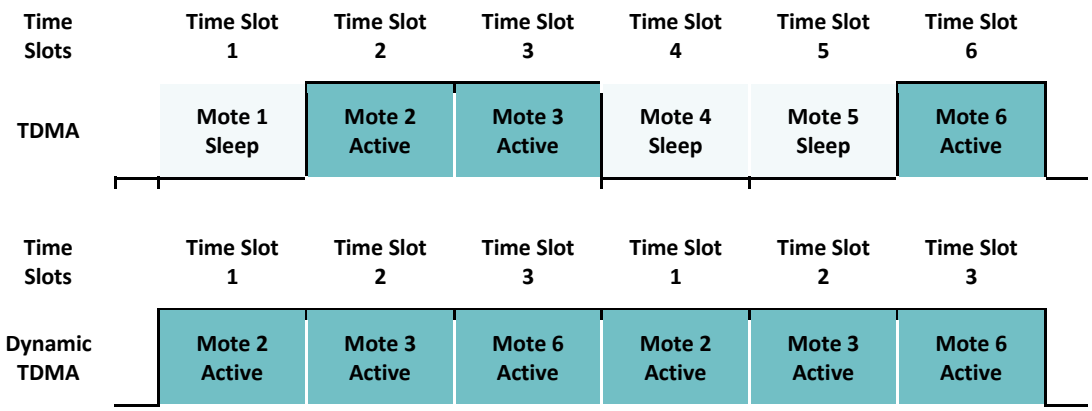


Figure 3.4 - Differences between TDMA and dynamic TDMA.

3.3.2 – Carrier-Sense Multiple Access with Collision Avoidance Protocol

Carrier-sense multiple-access with collision avoidance protocol (CSMA/CA) is a type of random access protocol, which is used in several existing communication standards such as the Zigbee specification (IEEE 802.14.5) for low power RF networking [39] [37]. In this class of protocols, a transmitting node checks, or "senses", if the channel is free before transmitting. If the channel is free, the node begins transmitting. All other nodes in range will sense this transmission and wait for the channel to become free before they transmit. If the transmitting node detects a signal from another node while transmitting, then the node terminates transmission and retries after a random interval.

3.3.3 – Slotted ALOHA

Slotted ALOHA is another type of random access protocol [36] [39] that handles channel contention with a probabilistic approach. In this protocol, a synchronizing signal is present to clearly define time slots. At the beginning of each time slot, any node within the network can transmit data. If only one node transmits data, the transmission will be successful, and an acknowledgement (ACK) will be transmitted back from the receiving node. If multiple nodes transmit, the transmitted signals will interfere, or "collide", with one another, causing the transmission to be unsuccessful. The result of this event is called a collision, and no ACK will be transmitted back. In this situation, each node that transmitted in the previous timeslot will retransmit its data with a probability ***p*** at the beginning of the next time slot. If multiple nodes transmit and a collision occurs again, then in the next time slot there will be a chance that some of those same nodes will not retransmit, but if a collision occurs again, this probabilistic retransmission will continue. Figure 3.5 illustrates an example of this protocol simulating communication from multiple motes during different time slots. In the first time slot, both nodes 1 and 2 transmit, so a collision occurs at the receiving node. In the second time slot, because no ACK was transmitted back to the transmitting nodes, both nodes will attempt to retransmit their respective data with a probability *p*. In this simulation, only note 1 retransmitted its data, so the communication was successful and an ACK was received. In the meantime, node 2 tries to retransmit its data in the third time slot, but since another node also initiated communication in the third time slot, a

collision occurred. This process continues to repeat itself until all nodes have successfully transmitted their data and received their corresponding ACK from the receiving node.

Key	
T	Transmit
-	No Transmission
C	Collision
S	Success

Time Slot	1	2	3	4	5
Mote 1	T	T	-	-	-
Mote 2	T	-	T	T	T
Mote 3	-	-	T	T	-
Result	**C**	**S**	**C**	**C**	**S**

Figure 3.5 - Example of how the slotted ALOHA protocol works.

3.3.4 – Directional MAC Protocols for mm Waves

In the RF world, there has been considerable interest in 60 GHz networks. At this frequency, communication systems have to use narrow-beam antenna systems to efficiently transmit power from node to node [40] [41]. Hence, these systems have similarities to narrow-beam optical systems, and face similar challenges such as LOS. However, these systems are slightly different from true narrow FOV optical systems because they still have the ability to sense the channel in all directions by circularly scanning their narrow beam antenna system [42].

Many of the currently proposed and developed directional MAC algorithms for these 60 GHz communication systems involve some sort of omnidirectional broadcast for obtaining channel information [43] [41] [42]. There are some new protocols that are being proposed and developed for 60 GHz communication systems that operate solely with directional transmissions, and these directional protocols involve using different electric field polarizations to transmit position information. The idea behind this is that linear and circular polarizations have different responses to reflections, so a receiving antenna system can detect whether or not there is a LOS path by comparing received signal strengths [40] [42]. In the case of FSOC systems, building coherent systems that can detect the electric field strengths requires a complex transceiver design. Direct detection optical receivers are less complex and more effective for small-size FSOC transceivers, but they detect intensity only and get no phase information.

3.4 – Modified MAC Protocol for SR-FSOC

In this short-range directional FSO network, communication can only take place between nodes that are in line of sight of each other. In addition, the master-slave network architecture allows communication to only take place between a cluster head and a mote, and between adjacent cluster heads. During communication between a cluster head and another node, the cluster head will only use the transmitter that contains the specific node within its field of view. While this

method helps ensure energy efficiency, it prevents most nodes from detecting whether or not channel to the cluster head is free. Consequently, a major design challenge is implementing the proper media access control protocol optimized for FSO communication to ensure successful directional communication between cluster heads and other nodes. As discussed in section 3.2, several types of these protocols exist, but they are mostly tailored and optimized for the radio frequency (RF) domain, not for directional FSO. The major reason being that in directional FSO, most nodes within the network cannot detect if the channel is free or not. Thus, a combination of the above three mentioned protocols with the proper modifications for FSO technology forms the basis for the developed directional MAC (DMAC) protocol used in this short-range low power directional FSO network.

3.4.1 – Basic Network Communication Protocol

A network cluster is formed if at least 1 mote connects to a cluster head. Upon powering on, the mote transmits a signal periodically. This signal informs any cluster head within its FOV that the mote needs to be connected to the network. When the cluster head detects the signal, it transmits a unique MAC address back to the mote. Once the mote receives the signal and updates its internal memory, it stops transmitting the signal periodically and is now part of the network. From this point on, every packet that the mote transmits contains its MAC address, and this allows the cluster head to detect which mote is transmitting data.

3.4.2 – Dynamic Time Division for Multiple Access

The shorter wavelengths of FSO technology provide the potential for extremely fast data rates, and with faster data rates, a lot of information can be moved from one point to another. Dynamic time division multiplexing provides an organized system of communication that can maintain these extremely fast data rates throughout network communication by dynamically adding and removing time slots. D-TDMA can even adjust the width of the time slots in case a certain node requires longer time slot intervals than the others. In addition, because of these extremely fast data rates, data rates between individual motes and the cluster head would not suffer as much as RF wireless networks even if there were many time slots active with different motes communicating with the cluster head. These features make D-TDMA an excellent protocol to gather data from multiple motes within a network cluster.

The cluster head assigns a time slot to a mote after it receives a request from a mote. Once time slots are issued, the cluster head transmits a synchronization signal to the specific mote whose time slot is beginning, alerting it to transmit data back to the cluster head. After that time slot is over, the cluster head transmits a synchronization signal to the next mote in the D-TDMA time slot queue. Once the cluster head gets to the end of the queue, it begins from the beginning. This process continues until there are no time slots left in the queue. A time slot is dropped once the mote lets the cluster head know it has no more data to transmit. At this point,

the mote can enter sleep mode to converse energy. One issue that must be dealt with when using D-TDMA is that when motes enter sleep mode and have their time slots removed from the D-TDMA queue, they will be unable to communicate with the cluster head on their previously assigned time slots when they are re-awaken. Because most motes have no ability to detect if the channel is free, a random access protocol will be necessary to handle possible channel contention when motes wake up and need to transmit data. Channel contention occurs when multiple motes or an adjacent cluster head try to communicate with the cluster head at the same time or when the cluster head is already busy communicating with a different node.

3.4.3 – Theoretical Rationale for FSOC Random Access Protocol Design

The objective in protocol development for a master-slave directional network is minimizing packet collisions at the master node, or cluster head, because this node is the gateway between other clusters and the end destination. The major design challenge comes from the fact that all motes within a network cluster have no ability on their own to sense whether or not the channel is free, which is the factor that ultimately leads to packet collisions at the cluster head.

Figure 3.6 depicts a network configuration of a single network cluster where 5 motes are connected to a cluster head through 5 of its 5 transmitters. Three motes are currently sleeping, while two motes are actively transmitting data via D-TDMA time slots. Let's consider the situation where the mote within the FOV of transmitter 5

($mote_5$) wakes and wants to transmit to the cluster head. The FOV of $mote_5$ cannot detect the transmission from $mote_1$ or $mote_4$, so it does not know that the channel is currently busy.

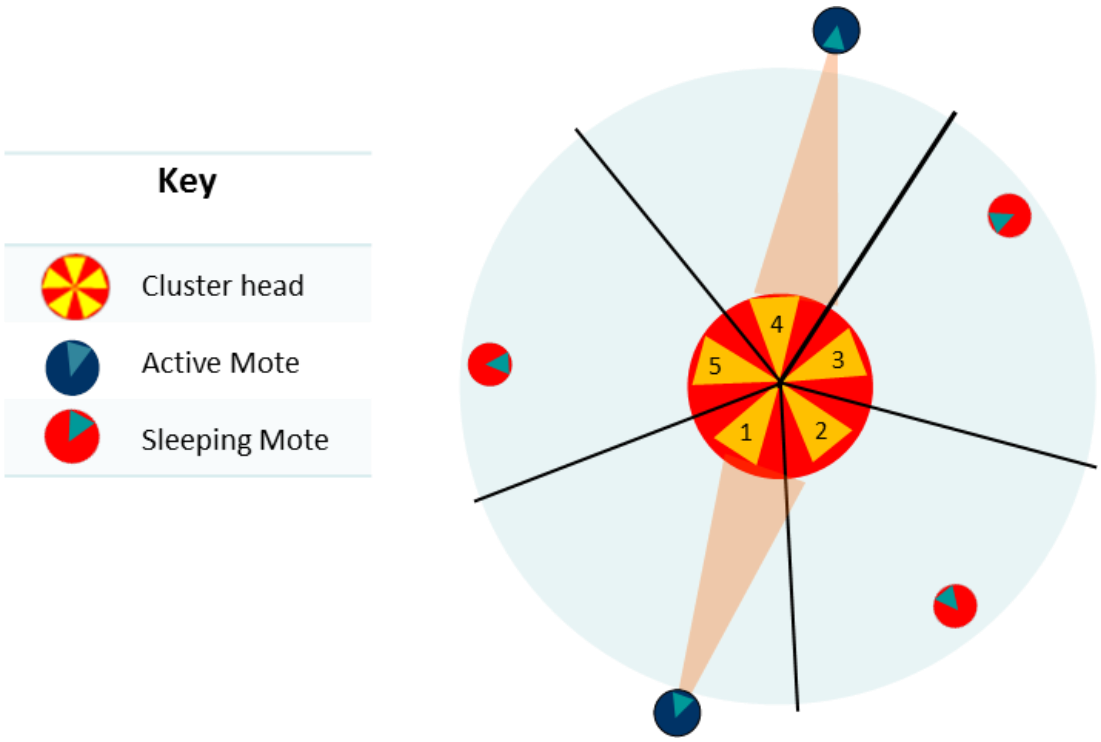


Figure 3.6 - Single cluster network of 2 active and 3 sleeping motes.

The only possible way for $mote_5$ to detect that the channel is busy is if the cluster head broadcasts the D-TDMA synchronization signal over the entire FOV via all of its transmitters while it was collecting data from $mote_1$ and $mote_4$. In this manner, if either $mote_2$ or $mote_3$ was to wake up as well, it would also be able to sense whether or not the channel is free by detecting the synchronization signal. The problem with this method is that it requires the cluster head to constantly use all of its transmitters to broadcast the synchronization signal while collecting data from the motes that have been assigned time slots within the D-TDMA queue, when only 1 transmitter is actually needed to broadcast the synchronization signal

communicate and retrieve data from 1 specific mote during a time slot. In essence, the cluster head would be consuming 5 times the amount of energy to maintain a synchronization signal over the entire FOV so that motes waking up could sense if the channel is free. This is an inefficient use of energy and would not be practical for a low-power network.

Instead, it would be more practical for $mote_5$ to attempt to access the channel regardless of whether the channel was free. If the channel is free, $mote_5$ gets assigned a time slot without incident; however, if the channel is not free, then a packet collision would be detected at the cluster head. By detecting these packet collisions, the cluster head is able to predict that at least 1 other mote is trying to access the network, and that it needs to update its D-TDMA queue. Therefore, packet collision detection becomes the cluster head's mechanism to updating its D-TDMA queue.

From an energy perspective this method would be more practical than broadcasting the synchronization signal over the entire field of view, because ideally, it would require less energy to re-transmit the lost data during the packet collisions than maintaining a 5 transmitter signal over every time slot. In addition, from a latency perspective this method would be more practical for real-time sensitive applications. For example, even if the cluster head was broadcasting the synchronization signal over the entire FOV while it was communicating with $mote_1$ and $mote_4$, it could take a long time before both motes are finished communicating with the cluster head.

Thus, the channel may not be free for a lengthy period of time. If the information from this newly awaken mote is critical, a lengthy delay before the channel becomes free would not be optimal. For these applications, it would be essential that $mote_5$ gets the cluster head's attention and onto a time slot in the D-TDMA queue as soon as possible. The quickest way would be by simply trying to access the channel regardless of whether the channel is free or not. Packet collisions would occur if the channel is not free. By detecting these collisions, the cluster head can assume that at least 1 other mode is trying to access the channel. At this point, the cluster head would run an efficient random access protocol optimized for directional FSO networks to resolve the channel contention and successfully update the D-TDMA time slots in the queue to incorporate $mote_5$. Once the channel contention has been resolved, the network can resume communication via the time slots in the D-TDMA queue.

3.4.4 – Modified FSO Random Access Protocol for a Single Cluster Network

The random access protocol is triggered by detecting packet collisions. Upon detecting packet collisions, the protocol runs a random access protocol to successfully resolve the channel contention. The developed RA protocol is a combination of slotted ALOHA and CSMA that has been modified to fit the needs of a master-slave directional FSO network.

3.4.4.1 – Packet Collision Detection

Whenever a mote tries to access the channel to either retrieve a time slot from the cluster head or to transmit data to the cluster head, the mote transmits its MAC address. If the channel to the cluster head is free, the cluster head will detect the MAC address and transmit an acknowledgement (ACK) back the mote, as well as a time slot for it to transmit in if it requested one. If the cluster head detects an incoming transmission but cannot detect a MAC address, the incoming transmission is labeled as a packet collision.

Since the random access protocol is triggered by packet collisions, the cluster head needs a way to separate the packet collisions that are caused by channel contention from the packet collisions that are caused by random degradations in the optical channel to ensure that the random access protocol only runs when nodes are trying to gain access to the network.

In order to help separate these types of collisions, the protocol is designed so that whenever a mote that is not on a time slot tries to access the channel, it will transmit a certain number of requests within a certain amount of time. If the cluster head is communicating with another node during this time, it will detect a certain number of collisions at a certain rate. The idea is that collisions from channel degradation will be a random process, while these forced collisions are at a predetermined rate. Thus, based on the sequence and frequency of packet collisions, the cluster head will determine if in fact channel contention is occurring or

if the link is simply fading. If channel contention is determined, the cluster head will initiate the random access protocol, resolve channel contention, and update the time slots of the TDMA queue.

3.4.4.2 – Modified Directional FSO Random Access Protocol Algorithm

Once the cluster head determines that it must update its D-TDMA queue, it runs its random access protocol. The specifics are as follows:

1. The cluster head broadcasts a stop transmission request (STR) signal to the entire FOV telling all motes that network traffic will be temporarily halted for the purpose of rebuilding the D-TDMA scheme. Upon detecting the STR signal, all motes that were already communicating via a time slot stop transmitting and enter sleep mode. The motes that were not on time slots and were trying to access the channel stop transmitting as well but remain awake.
2. Immediately after transmitting the STR signal, the cluster head waits for a specific time interval to pass. During this interval, the cluster head does not expect to receive any signal because all motes are supposed to stop transmitting. If it does detect a signal during this interval, then it immediately rebroadcasts another STR signal to the entire field of view and once again waits for the specific time interval to pass. This process continues

until the specified time interval passes without the cluster head detecting any signal.

3. Next, the cluster head broadcasts a random access synchronization (RAS) signal to the entire FOV. Within the RAS signal, the cluster head transmits a probability, *p,* to all the motes. The motes use this value of *p* to generate a possible reply back to the cluster head. Therefore, after the cluster head transmits a RAS signal, it waits for a specified amount of time for a possible reply back from the motes.
4. If only 1 mote replies, then the cluster head successfully detect its MAC address and transmits back a time slot acknowledgement to the mote. This mote now has a time slot and can enter sleep mode. If multiple motes reply, then the cluster head detects a collision, and if no motes reply, then the cluster head detects a timeout. In either of these latter cases, no time slot is transmitted back to the motes. This process continues until all motes have been given a time slot. The cluster head determines that all motes have been given time slots when it transmits a value of p=1 within a RAS signal and detects a timeout. Therefore, within this protocol, the value of *p* must be updated in an efficient manner during each RAS signal to try to minimize the chance of detecting a collision or timeout and maximize the chance of detecting a successful transmission. This topic will be discussed in greater detail in chapter 5.

5. The cluster head updates its D-TDMA queue and resumes communication with all motes in the queue using the specific transmitter whose FOV contains the target mote.

In the above described protocol, a synchronization frame consists of a cluster head's RAS signal and the ensuing "timeout", "collision", or "success" detection based off of the possible reply from any number of motes. Figure 3.7 displays a flowchart outlining the above described steps.

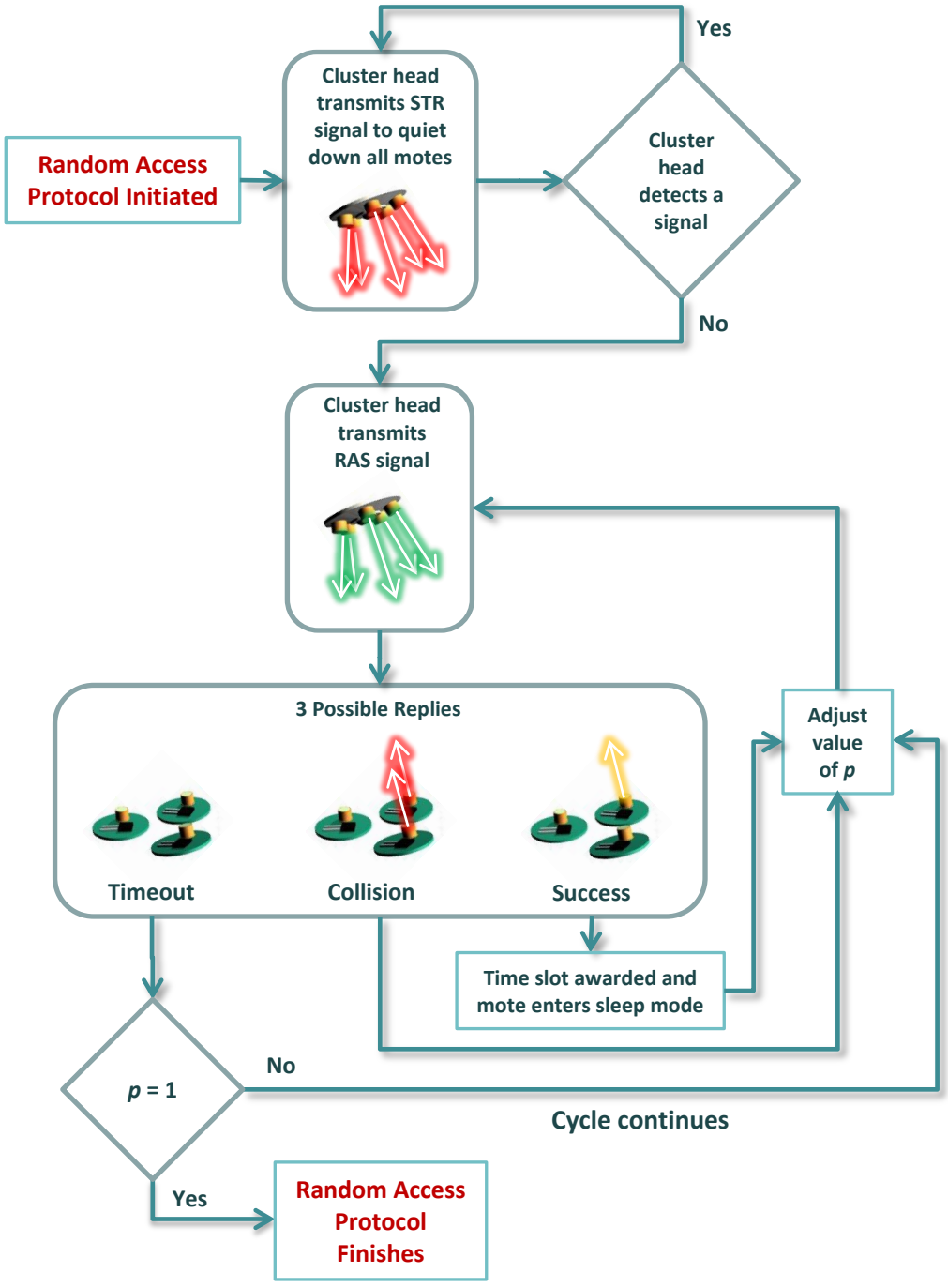
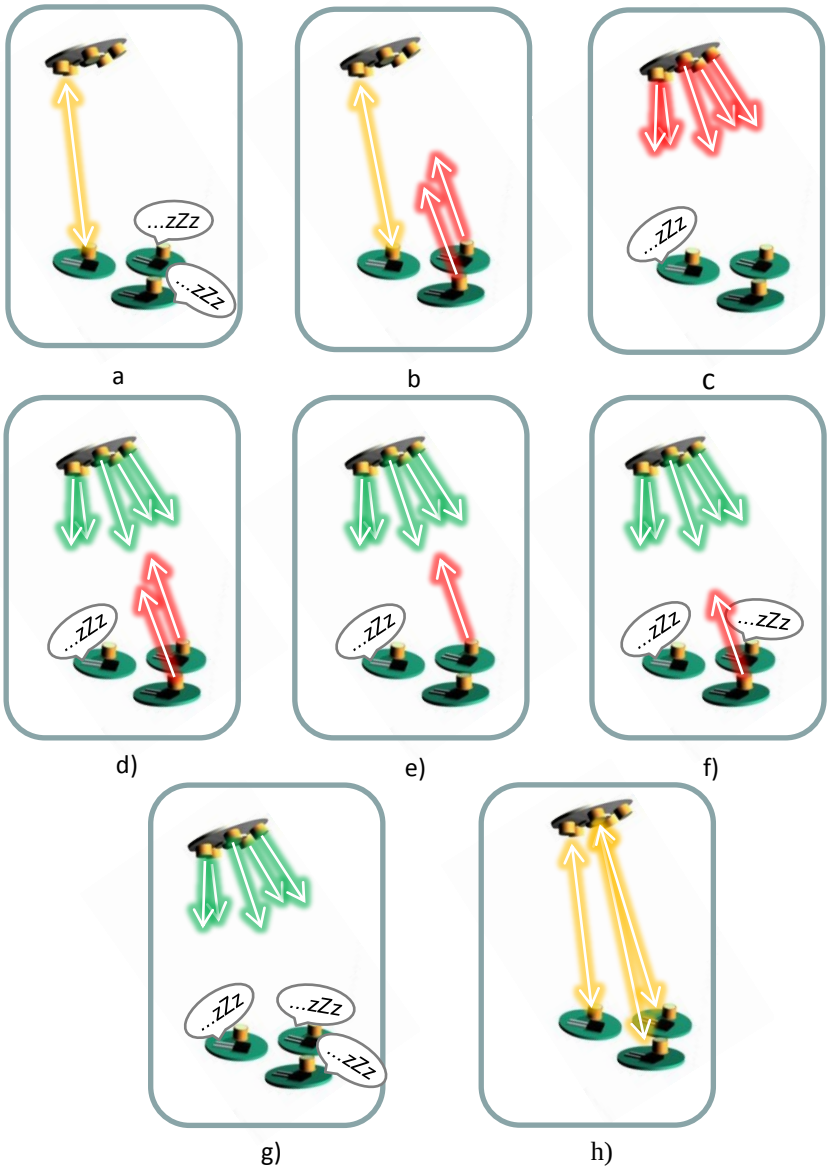


Figure 3.7 - Flowchart of the modified FSOC RA protocol algorithm.

Figure 3.8 illustrates how the MAC protocol handles the situation where one mote (m_1) is communicating to the cluster head on a single time slot and two motes (m_2, m_3) wake up and try to access the channel. In this example, it will be assumed that the cluster head transmits *p* values in the following repeating sequence: [1, .5, .5, 1]. This means that in the first RAS signal, it transmits a value of $p = 1$. In the 2^{nd}, 3^{rd}, and 4^{th}, RAS signals, it transmits $p = .5$, $p = .5$, and $p = 1$, respectively. If more RAS signals are needed, it repeats the sequence. Also, M_1, M_2, and M_3 correspond to the left-most, center, and right-most motes.

In frame a), the cluster head is collecting data from m_1 on its specific timeslot using the transmitter whose FOV contains m_1. In frame b), the 2 motes wake up and try to access the channel to acquire a timeslot, but since the channel is not free, the cluster head detects a packet collision. In frame c), the cluster head initiates the RA protocol and transmits the STR (Stop Transmission Request) signal. The signal quiets all three motes and puts m_1 into sleep mode because it already has a time slot. The other 2 motes, m_2 and m_3 quiet down, but remain awake and await further instructions. In frame d), the cluster head begins transmitting RAS (random access synchronization) signals over the entire FOV. In the first RAS signal, the cluster head transmits $p = 1$. This causes both m_2 and m_3 to generate a reply causing a packet collision at the cluster head. In frame e), the cluster head transmits another RAS signal with $p = .5$. In this case, only 1 mote, m_3, generates a reply. The cluster head successfully detects the MAC address and assigns m_3 a time slot. M_3 now enters sleep mode, leaving only 1 mote, m_2, that still has to acquire a time slot. In frame f),

the cluster head transmits the 3rd RAS signal with p = .5, and m_2 generates a reply. The cluster head assigns a time slot to m_2, and m_2 enters sleep mode. In frame g), the cluster head transmits another RAS signal with p=1. Since there are no motes remaining without a time slot and all other motes are in sleep mode, the cluster head detects a timeout. Furthermore, since it transmitted a value of p = 1, it knows that there are no more motes waiting for a time slot. Therefore, it finishes the random access protocol and switches back to D-TDMA with an updated queue. Now that it can resume communication via time slots, the cluster head only needs to use the transmitters whose FOV contains the motes with time slots. This is shown in frame h), where only 2 transmitters are needed to communicate with the three different motes.



	Cluster head		Motes		
Figure	**Transmit**	**Receive**	**M_1**	**M_2**	**M_3**
a)	Slot 1	Data M_1	Transmit	Sleep	Sleep
b)	Slot 1	Collision	Idle	Wakes	Wakes
c)	STR	Timeout	Sleep	Idle	Idle
d)	RAS, p = .1	Collision	Sleep	Transmit	Transmit
e)	RAS, p = .5	Success	Sleep	Idle	Transmit
f)	RAS, p = .5	Success	Sleep	Transmit	Sleep
g)	RAS, p = 1	Timeout	Sleep	Sleep	Sleep
e)	Slot 1	Data M_1	Transmit	Idle	Idle

Figure 3.8 - Illustrative example of how the MAC protocol handles network comm.

3.5 – Transmission Probability Value, *p*

The probability of transmission, *p*, transmitted within the RAS signal affects the total number of synchronization frames (SF) required to assign all timeslots. The total number of SFs determines the network's random access times. Shorter random access times mean faster mote acquisition. Therefore, an optimal value of *p* must be transmitted to each SF to minimize random access times.

3.5.1 – Calculating Optimal Value of *p*

Let's consider a network cluster where 3 motes are contending for channel access. If a "0" denotes that a mote did not reply and a "1" denotes that the mote did reply, the total possible outcomes for the 3 mote channel contention are shown in figure 3.9.

Possibilities ($m_1m_2m_3$)	Probability	Outcome
000	$(1-p)^3$	**Timeout**
001	$\mathbf{p\,(1-p)^2}$	**Success**
010	$\mathbf{p\,(1-p)^2}$	**Success**
011	$p^2\,(1-p)$	**Collision**
100	$\mathbf{p\,(1-p)^2}$	**Success**
101	$p^2\,(1-p)$	**Collision**
110	$p^2\,(1-p)$	**Collision**
111	$\mathbf{p^3}$	**Collision**

Figure 3.9 - Possible outcomes and probabilities for 3 motes contending for channel access.

A successful outcome occurs when only 1 mote replies within the synchronization frame. If S_n is defined as the probability of a successful packet transmission for a network cluster with *n* motes contending for channel access, then

$$S_3 = P(001) + P(010) + P(100) = 3p(1-p)^2 \quad (3.1)$$

A similar deduction can be made for the case where 10 motes are contending for channel access,

$$S_{10} = 10p(1-p)^9 \quad (3.2)$$

Figure 3.10 shows how the probability of a successful transmission for the contending 10 motes changes for different *p* values.

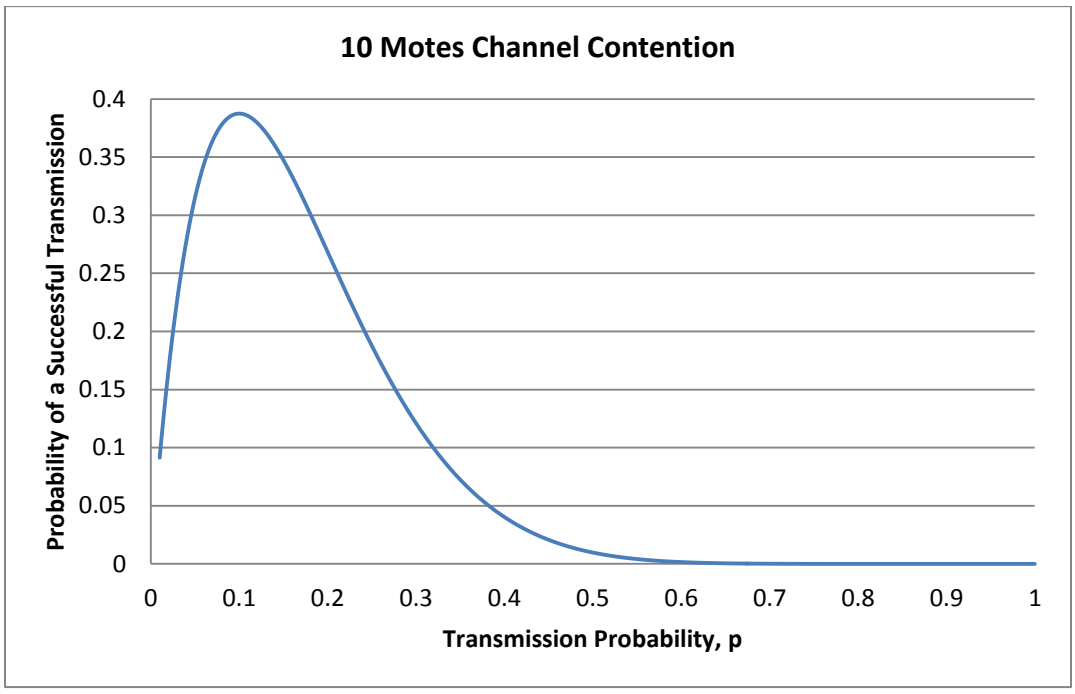


Figure 3.10 - Probability of a successful transmission vs. transmission probability for 10 mote contention.

The largest probability for a successful transmission occurs at $p = 0.10$.

In the general case for *n* contending motes,

$$S_n = np(1-p)^{n-1} \tag{3.3}$$

The maximum probability for a successful transmission occurs when

$$\frac{dS_n}{dp} = 0 \quad \text{,so}$$

$$\frac{dS_n}{dp} = -n(n-1)p(1-p)^{n-2} + n(1-p)^{n-1} = 0 \tag{3.4}$$

$$(n^2 - n)p(1-p)^{n-2} = n(1-p)^{n-1} \tag{3.5}$$

$$(n^2 - n)p = n - np \tag{3.6}$$

$$p = \frac{1}{n} \tag{3.7}$$

In general, the optimal value is $p = \frac{1}{n}$ where *n* denotes the number of motes contending for channel access. Figure 3.11 plots this relationship on a log-scale for the number of motes contending.

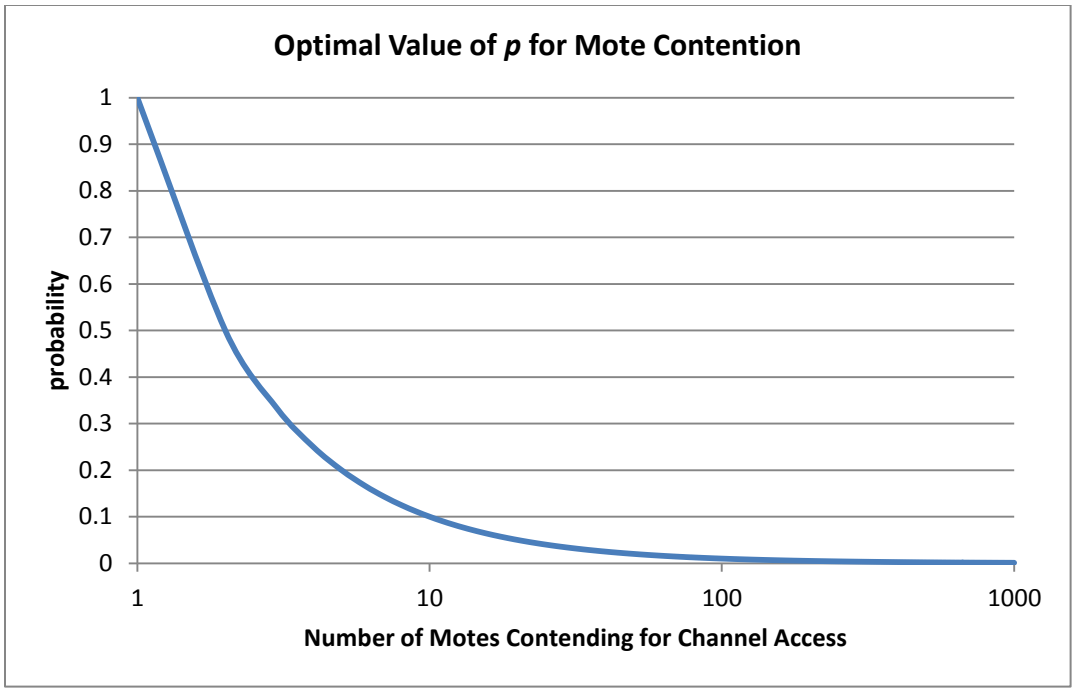


Figure 3.11 - Optimal value of transmission probability vs. number of motes contending for channel access.

The plot shows that when 10 or fewer motes are contending for channel access, the *p* value changes quite dramatically as compared to the rest of the plot. At lower levels of mote contention, the value of *p* will noticeably affect the random access times, but at larger levels of mote contention, the value of *p* shouldn't affect the random access time much as long as it is very small.

3.5.2 –Expected Total Number of Synchronization Frames

If we assume that the optimal value of *p* is used at all times during the random access protocol, then the expected total number of synchronization frames ($<T_{SF}>$) required to resolve *n*-mote channel contention can be calculated from S_n by,

$$\langle T_{SF} \rangle = \frac{1}{S_n} + \frac{1}{S_{n-1}} + \frac{1}{S_{n-2}} + \cdots + \frac{1}{S_1} \tag{3.8}$$

$$\langle T_{SF} \rangle = \sum_{j=1}^{n} \frac{1}{S_j} \tag{3.9}$$

Figure 3.12 plots the relationship between the $<T_{SF}>$ and a small number of motes contending for channel access, while figure 3.13 shows a plot of the $<T_{SF}>$ for a very large range of motes. Each plot shows a linear relationship between the required SFs and the number of contending motes. In figure 3.13, a linear trend line was added to the plot with the following equation,

$$< T_{SF} >= 2.7034x \tag{3.10}$$

where *x* is the number of contending motes. In essence, each contending mode adds an additional ≈2.7 SFs to the RA protocol. This is assuming that the optimal probability is being transmitted within each RAS signal. The random access time can then be computed by multiplying the $<T_{SF}>$ by the time interval of a single SF. The time interval depends on the optical throughput of the system and the packet structure of the SF.

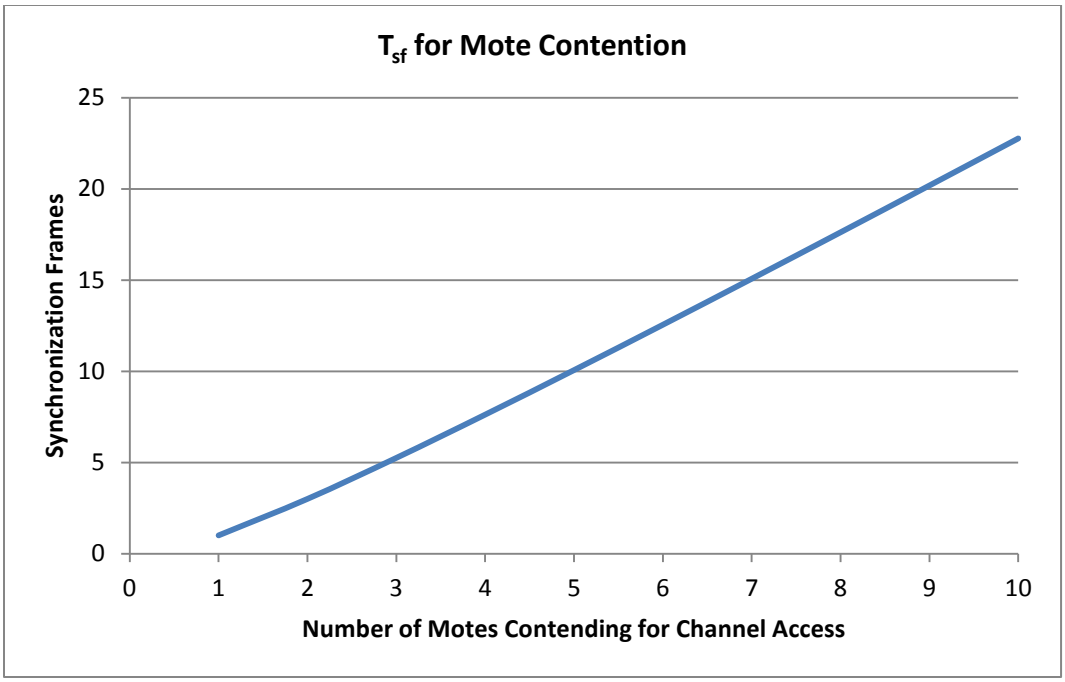


Figure 3.12 - Total number of SFs needed to resolve channel contention when 0-10 motes are contending for channel access.

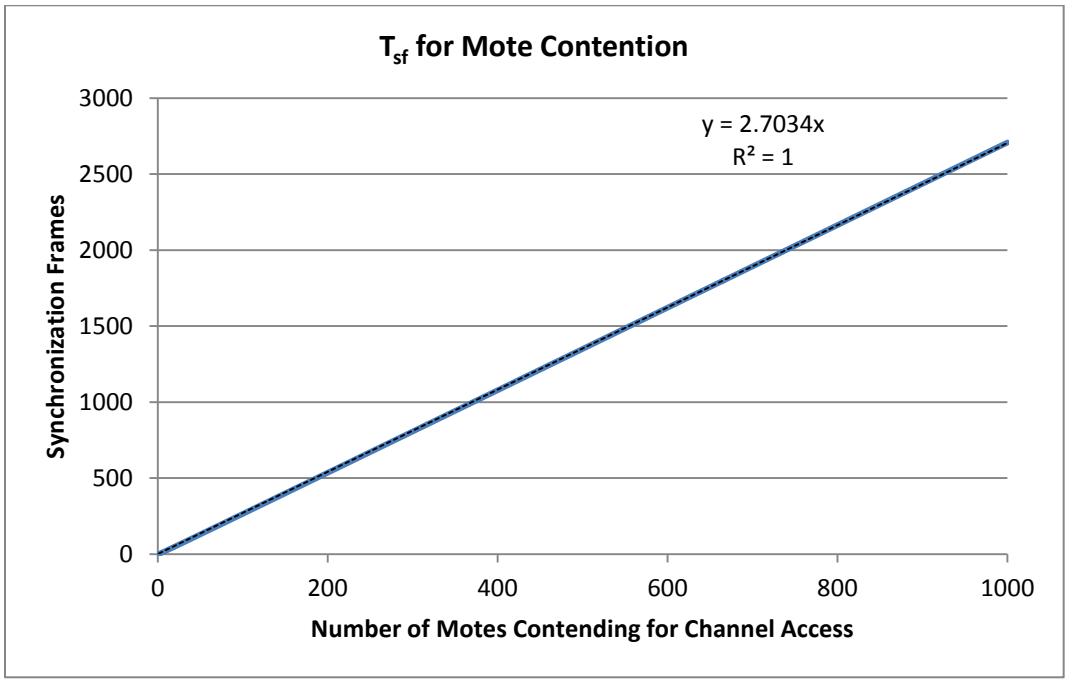


Figure 3.13 - The total number of SFs needed to resolve channel contention when 0-1000 motes are contending for channel access.

3.5.3 – Absolute Best Performance

The fastest the RA protocol can resolve channel contention is when a mote is acquired after every RAS signal (within every SF). This would lead to the shortest number of SFs for a given network. However, based on probability, this is a very unlikely event. If we consider a network cluster with *n* motes contending for channel access and denote the probability of achieving the shortest number of SFs as P_n^{Best}, the probability can be found by the equation,

$$P_n^{Best} = S_n * S_{n-1} * S_{n-2} * \dots * S_1 \tag{3.11}$$

$$P_n^{Best} = \prod_{j=1}^{n} S_j \tag{3.12}$$

Figure 3.14 plots this relationship on a log scale for the probability. The probability of this event happening is 1 when only 1 mote is trying to access the channel, and that is obvious since there would be no channel contention. After 6 mote contention, the probability drops below .01, and continues to drop at a nonlinear rate as channel contention increases. (The rate appears linear because the y-axis is plotted on a log scale.)

Figure 3.14 - The probability of the abs best performance occurring vs. number of motes contending for channel access.

Given the unlikely probability of this event, the expected outcome of the RA protocol will be the $<T_{SF}>$ with some statistical variance. The actual statistical distribution will be measured from the hardware experiments.

3.6 – Extending DMAC Protocol to Multiple Network Clusters

When multiple network clusters are present, adjacent cluster heads can form backbone links with one another. The backbone link allows motes from one network cluster to communicate with another. This is important in situations where only one network cluster is connected to an end destination. In this case, motes from other clusters have to relay information from one cluster head to another to reach the final destination. With multiple network clusters present, cluster heads

can potentially cause collisions with other cluster heads. This new dynamics needs to be accounted for to ensure systematic and controllable communication between adjacent network clusters. Consequently, a few modifications need to be made to the D-TDMA scheme and RA protocol. First, the D-TDMA scheme will be discussed to illustrate how time slots work to ensure a collision free communication process in section 3.6.1. Then, the possible breakdown mechanisms in communication will be discussed in section 3.6.2.

3.6.1 – Static Network: Organizing the Multi-Cluster D-TDMA Queue

Let's consider a network of 3 network clusters, as shown in figure 3.15. Let's further assume that cluster head of network cluster A (CH_A) is linked to an end user station, while CH_B and CH_C are simply acting as relay stations to CH_A. This makes CH_A the "master" cluster head in relationship to CH_B and CH_C. CH_B and CH_C become "slaves" in relationship to CH_A but have no relationship with each other.

In this network configuration, it will be assumed that all motes within the entire multi-cluster network have already been added onto the D-TDMA queue. From this perspective the network is static, meaning that no more will be added onto the queue or dropped from the queue.

To retrieve data from this queue, CH_A must determine how many motes in the queue are from different clusters. In this case, 1 mote is attached to CH_C, 6 motes

are attached to CH_B, and 1 mote is attached to CH_A. Therefore, CH_B will require 6 D-TDMA slots to collect the data from its 6 motes. Similarly, it will take both CH_A and CH_C 1 D-TDMA slot to collect data from their mote. A D-TDMA slot is the synchronization signal that cluster heads use to alert the mote of its time slot.

To maximize the throughput, on the 1st D-TDMA slot, CH_A will transmit a command to CH_B that triggers CH_B to begin collecting data from all motes within its cluster that are in the queue. Then, on the 2nd slot, CH_A transmits a similar command to CH_C while at the same time CH_B begins collecting data from its motes. On the 3rd slot, CH_A collects data from its mote, CH_C collects data from its mote, and CH_B collects data from its 2nd mote. On the 4th slot, CH_A is done collecting data from its mote and also knows that CH_C will be done collecting data from its mote, so CH_A uses this slot to retrieve the data from CH_C. Meanwhile, CH_B is still collecting data from its motes. Now, on the 5th slot, CH_A and CH_C have free slots because they have no more motes on the queue. They can use these free slots to collect more data from their respective motes or to keep their channel free in case of other motes wake up within their network cluster.

If CH_C had wanted to use the free slots to collect more data from its mote, then it would have conveyed that to CH_A on the 4th time slot when they were communicating with each other. This is possible because the master cluster head shares the D-TDMA queue with its slave cluster heads when they communicate. In

this case, the initial D-TDMA queue that was developed by CH_A was then transferred to CH_B and CH_C on the 2nd and 3rd slots.

In this queue, CH_A has free slots until the 8th slot, when it has to begin retrieving data from CH_B. Because CH_B had 6 motes to collect data from, it will take CH_A 6 slots to acquire the data. Meanwhile, CH_C has free slots all the way until slot 3 repeats. The total queue in this cycle is 13 slots long.

Figure 3.16 summarizes the actions of each cluster head in regards to the D-TDMA slots for the network shown in figure 3.15.

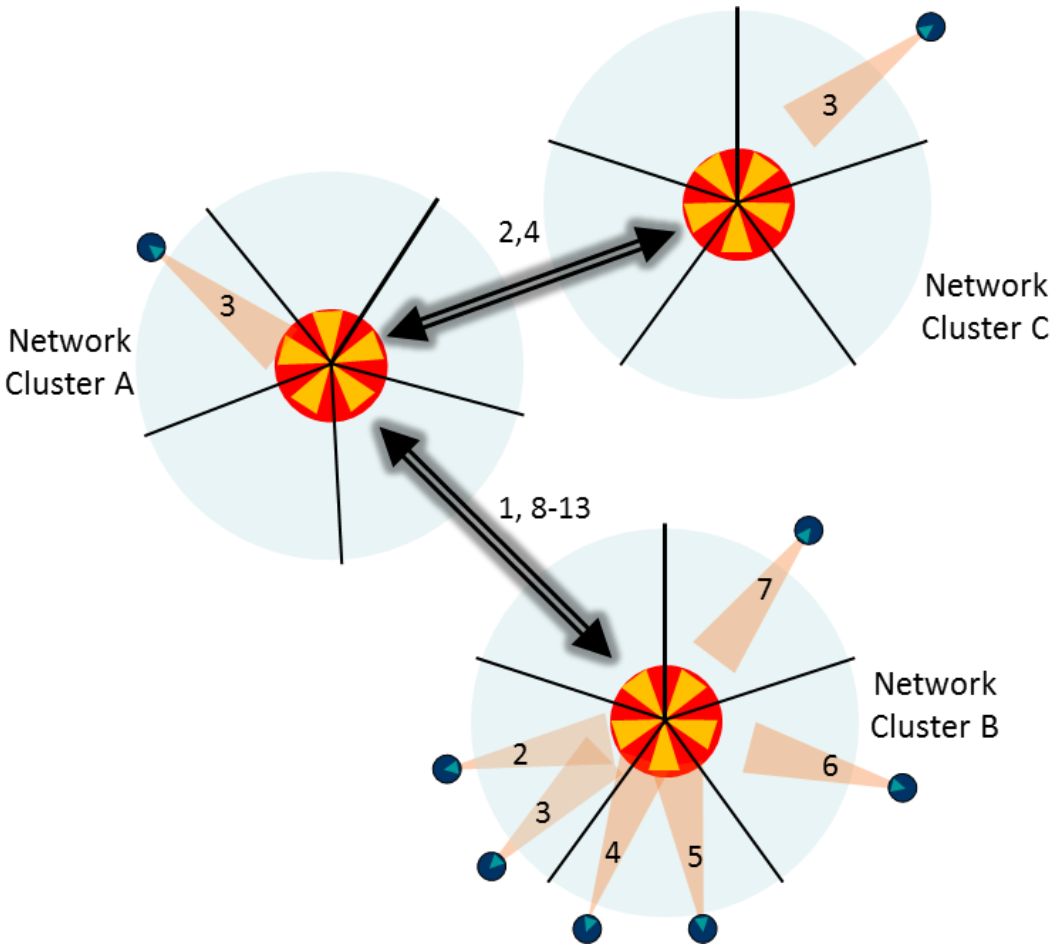


Figure 3.15 - Network configuration of 3 network clusters.

Slot #	Cluster Head A Action	Cluster Head B Action	Cluster Head C Action
1	Relays D-TDMA to B	Relates D-TDMA to A	
2	Relays D-TDMA to C	Collects Data	Relays D-TDMA to A
3	Collects Data	Collects Data	Collects Data
4	Retrieves Data from C	Collects Data	Transmits Data to A
5	Free Slot	Collects Data	Free Slot
6	Free Slot	Collects Data	Free Slot
7	Free Slot	Collects Data	Free Slot
8-13	Retrieves Data from B	Transmits Data to A	Free Slot

Figure 3.16 - The D-TDMA queue for each cluster head shown in the network configuration of figure 3.15.

3.6.2 – Dynamic Network: Avoiding Multi-Cluster Head Interference

The reason that these cluster heads might interfere with each other is because the RA protocol handles channel contention by transmitting RAS signal frames across the entire FOV. Because the MCH and SCH have a direct FOV link, any broadcast from that specific transmitter can interfere with each other's network communication. Because of this new dynamic, several potential issues must be handled by the protocol to ensure that the cluster heads do not interfere with each other's network communications.

There are two different network situations that will lead to multi-cluster head interference. Both network situations can be understood by looking at a single master-slave cluster head link, which is a link between two adjacent cluster heads

where one is the master and one is the slave. In figures 3.17 and 3.19, the master cluster head (MCH) is CH_A and the slave cluster head (SCH) is CH_B.

3.6.2.1 – Active Master-Slave Backbone Link

The first network situation is when a backbone link is active and both cluster heads are in the D-TDMA queue. This means that both cluster heads are actively collecting data from their motes and then the SCH is relaying the data to the MCH on the proper time slots. The network configuration is shown in figure 3.17 and the D-TDMA queue is shown in figure 3.18. The cluster heads are shown as having an active backbone link through the transmitters with the FOV shown in yellow.

While these cluster heads are operating on the D-TDMA queue, if any sleeping mote wakes up, there will be collisions that will force the RA protocol to run. To avoid interfering with each other's networks, the cluster head that initiates the RA protocol will not transmit the STR or RAS signals across the transmitter that generates the backbone link (shown in yellow). Instead, the cluster head resolves the channel contention using the transmitters that are not connected to the adjacent cluster head.

Running the RA protocol desynchronizes the two cluster heads in their current D-TDMA queue because the RA protocol takes a finite amount of time to run. Therefore, both cluster heads need an efficient way to resynchronize and share the

new queue. Regardless if both or one of the cluster head runs the RA protocol, the MCH will always resynchronize the cluster heads, and it will do so by waiting a specified amount of time before trying to broadcast to the SCH. This specific amount of time is the average random access time of the network. If a resynchronization does not happen after this interval, then it waits for an additional amount of time corresponding between the average time and 1 standard deviation above the average time, and so forth if additional time is needed because the RA protocol is taking longer than normal.

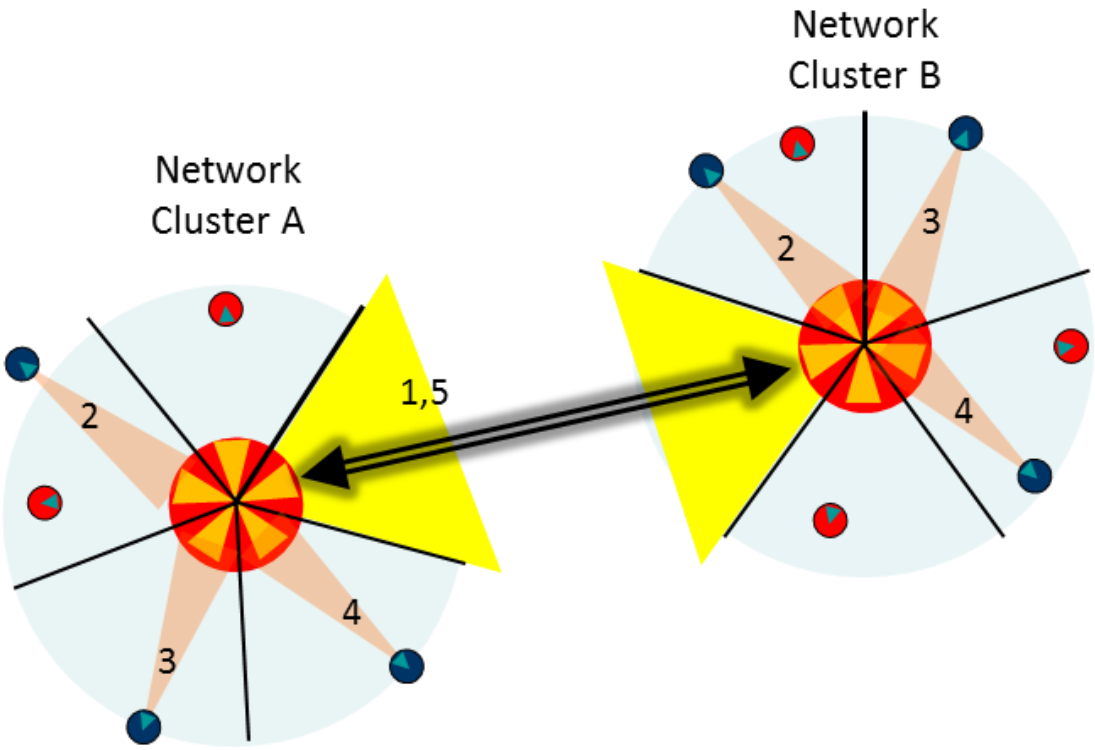


Figure 3.17 - Network configuration of 2 network clusters connected by an active master-slave backbone link.

Slot #	Cluster Head A Action	Cluster Head B Action
1 (Backbone Link)	Relays D-TDMA to B	Relays D-TDMA to A
2	Collects Data	Collects Data
3	Collects Data	Collects Data
4	Collects Data	Collects Data
5 (Backbone Link)	Retrieves Data from B	Transmits Data to A

Figure 3.18 - The D-TDMA queue for the network shown in figure 3.17.

3.6.2.2 – Inactive Master-Slave Backbone Link

The second configuration is when the master-slave backbone link is inactive, and only the MCH is collecting data from active motes. The configuration is shown in figure 3.19. This means that no motes from the SCH network are active. Therefore, the SCH has no queue. Now, suppose m_{1B} wakes up and tries to access the channel. The SCH will pick it up immediately since there is no channel contention, and then it will try to gain access to the MCH by forcing a packet collision. The forced packet collision causes the MCH to initiates its RA protocol. The complication arises if any other mote in the SCH network wakes up and tries to access the channel the same time the MCH begins transmitting STR and RAS signals. If this situation occurs, then the SCH will detect nothing but collisions when it was expecting to receive RAS signals, and this would prevent it from getting a time slot. Furthermore, it effectively wasted communication time because the RA protocol was run and no motes new nodes were added to the queue.

To prevent this from occurring, whenever the SCH attempts to access the channel to the MCH but begins receiving collisions, it will transmit an STR signal across the entire FOV to quiet down the motes within its own cluster, and to let the MCH know that it still needs channel access. The MCH will detect the STR signal, and rebroadcast its own STR signal. Once the MCH receives a timeout, it knows that the SCH was able to quiet down all of its motes, and that it can proceed with the normal RA protocol.

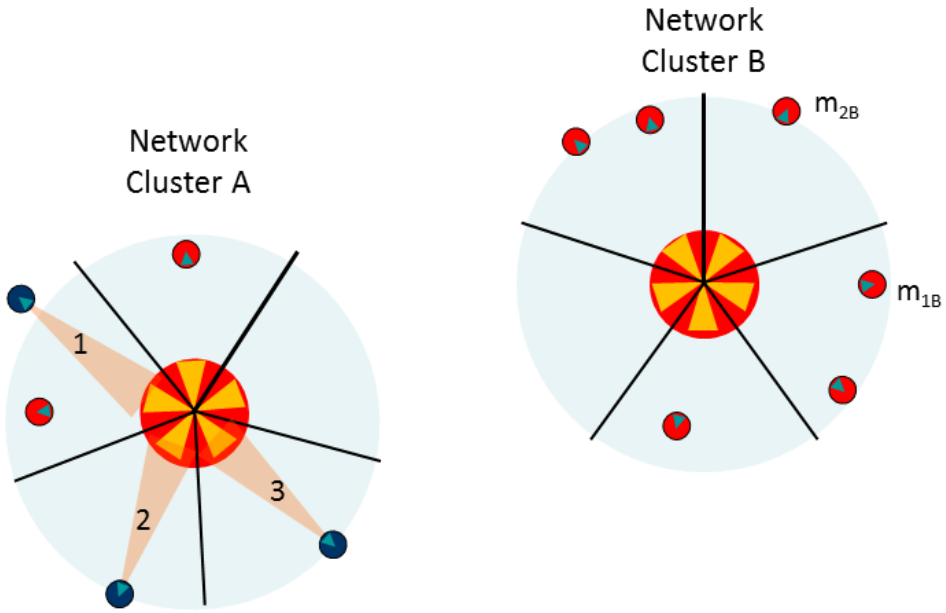


Figure 3.19 - Network configuration of 2 network clusters connected by an inactive master-slave backbone link.

Chapter 4 – SR-FSOC System Design and Implementation

4.1 – Introduction

This chapter focuses on the design rationale for the developed SR-FSOC hardware. Two types of optical communication systems were developed: the cluster head and the mote. Both the cluster heads and the motes are self-contained systems that can generate, modulate, transmit, and receive packets. All the modulation, demodulation, and control algorithms were custom developed in the "C" language and implemented on low power Atmel ATmega644p microcontrollers (MCU). This chapter discusses how the systems were designed and implemented. The hardware is broken down into three main components: the Rx, the Network Controls, and Tx. The inner workings of each component are explained in detail as well as how each component interfaces with one another to create an entire system. The chapter also presents the implementation of these design concepts into fully functional PCB systems using standard off the shelf chips along with custom developed control software. With the hardware fully described, a data throughput analysis of how fast a cluster head and mote can communicate with one another is presented. Finally, the chapter concludes with a discussion of how photovoltaics could be added to the hardware to increase energy efficiency.

4.1.1 – Background Information on ATmega644p Microcontroller

The ATmega644p MCU [44] was selected because it provided features that would greatly help in implementing low-power and compact-sized cluster heads and motes. The main features of the MCU that were used in the system implementation were hardware counters, hardware interrupt service requests and the on-board analog comparator. Other features used were the on-board analog to digital converter, multiple input/output pins, 8 kb of SRAM, and on-board 8 MHz RC oscillator, which served as the main MCU clock. With these features inside the MCU, fewer external components are required, allowing for a compact implementation, where the overall size of the cluster head and motes can be limited by the transceiver architecture. This section briefly explains the main features of the MCU that were used in the implementation.

4.1.1.1 – Counters

The MCU has two 8-bit counters and one 16-bit counter. These counters can be set to operate at different clock rates, based off a scaling factor from the main MCU clock. Each counter is implemented in hardware, so they automatically increment whenever the number of appropriate clock cycles has passed. For example, if a counter is set to operate at $1/8^{th}$ the frequency of the main MCU clock, then after every 8 main MCU clock cycles, the counter increments its value by 1. The maximum values an 8-bit and 16-bit counter can hold are 256 and 65535, respectively. After

the counters reach their maximum value, they reset to 0 and begin counting again, which is known as an overflow.

4.1.1.2 – Hardware Interrupt Service Requests

Hardware interrupt service requests are hardware events that trigger the MCU to temporarily stop performing the main program instructions, after the current instruction is processed. At this point, the MCU performs a specific set of program instructions that correspond to the particular hardware service request. After finishing these program instructions, the MCU resumes performing the instructions from the main program from the point where it previously stopped. These interrupts can be triggered by numerous different hardware events, such as an input pin switching from a logic low to a logic high position (or vice versa), from counters reaching certain values or overflowing, or from certain 8-bit registers being set to a logic high or low.

4.1.1.3 – On-board Analog Comparator

The MCU has an on-board analog comparator (AC). The AC operates by comparing input signals on a positive input pin and a negative input pin. If the positive input pin has a larger voltage then the negative input pin, the AC triggers a flag, or 8-bit

register, to go to its logical “high” state, which is $+V_{CC}$.. This flag can be set to trigger a hardware interrupt service request.

4.2 – SR-FSOC System Design

The SR-FSOC sensor network consists of a series of bi-directional communication links between a cluster head and another mote, as shown in figure 4.1. The propagation medium is the free space between the two devices. Since these networks consist of short-range links, the main loss due to the channel is the spreading, or divergence, of the beam, which is nearly 10° (half-angle).

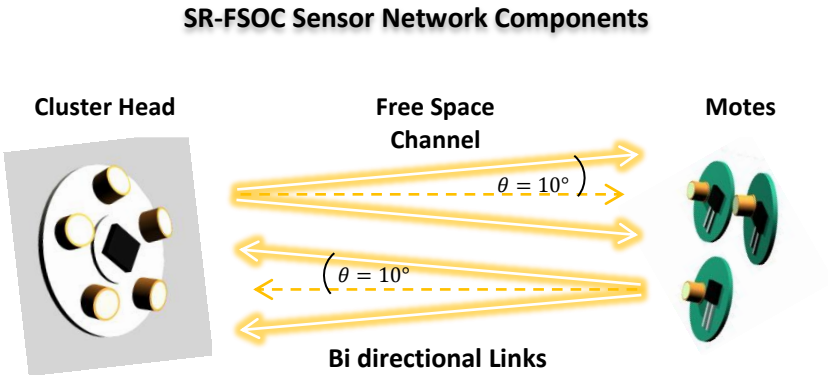


Figure 4.1 - Components of a SR-FSOC bi-directional communication link.

Figure 4.2 shows the system diagram of the developed FSOC mote and cluster head systems. The main components of the system are the receiver (Rx), the data processing and controls logic (DPCL), and the transmitter (Tx). The general overview of the system is as follows. An incoming optical signal is first converted into a very small electrical current signal at the photodiode, which is then amplified and

conditioned before being fed into the AC. The AC takes the analog input voltage and generates a series of hardware interrupt service requests whenever the input signal is greater than the reference voltage. At each of these hardware interrupt service requests, the decision logic generates a stream of how many logic “1”s and “0”s have been received. This stream is then demodulated by the data recovery logic and then processed by the network communication control logic so that the system can interpret the message and respond if necessary. If a response is necessary, the proper packet is generated through the transmission logic and transmitted to the laser driver circuitry via an output pin on the MCU. The laser driver circuitry provides the necessary current for the VCSEL to lase and transmit the outbound packet. In this implementation, the decision logic, data recovery logic, network communication control logic, and transmission logic were all developed in “C” and embedded in the MCU. The network communication control logic acts as the central brain of the system by administering the proper network communications, processing all data sent and received, and controlling the logical states of the system. It oversees the flow of data between the Rx and Tx of the system.

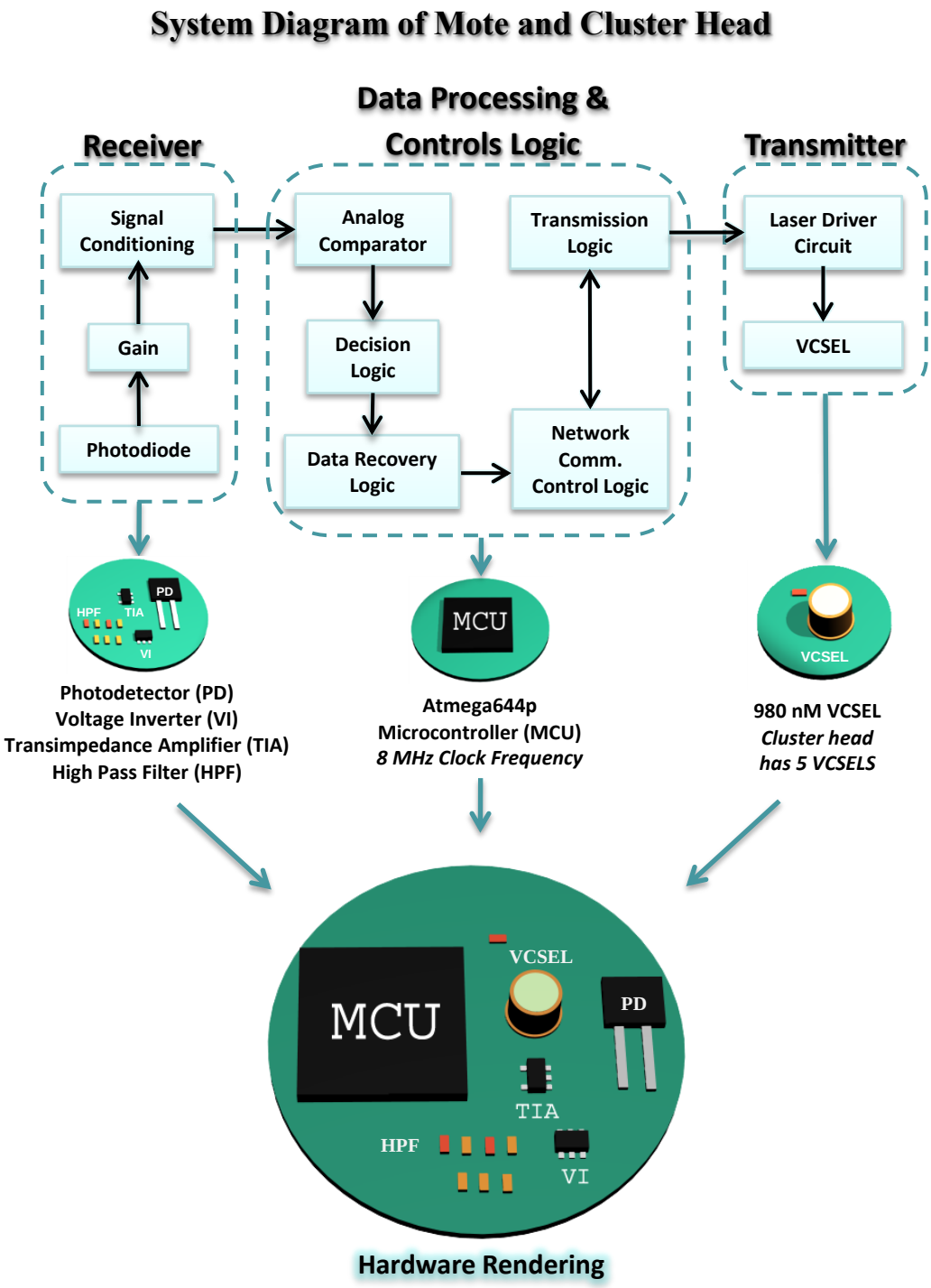


Figure 4.2 - System diagram of the developed FSOC cluster head and mote systems.

4.3 – Receiver Component

The Rx component converts an optical signal into an electrical current at the photodiode so that it can be electrically amplified and conditioned before being input to the AC. Figure 4.3 shows the circuit schematic of the detection, amplification, and conditioning circuit. The photodiode generates a photocurrent I_{ph} whenever an optical signal is incident at the detector's surface. The photodetector is reverse biased to reduce the depletion region capacitance so that the RC time constant of the detector circuit can be reduced [45]. C_i is the total input capacitance at the negative input terminal of the op-amp, which is a combination of the photodiode's depletion capacitance and the op-amp's input capacitance. By reverse biasing the photodiode, the photodiode's depletion capacitance can be reduced, which directly reduces C_i. C_f is a feedback capacitor that helps maintain stability around the feedback loop. R_F is the feedback resistance and its value determines the magnitude of the amplification. Because the op-amp has a narrow bandwidth, it also acts as a low-pass filter. The combination of this "internal" low-pass filter with the external high-pass filter provided sufficient signal conditioning.

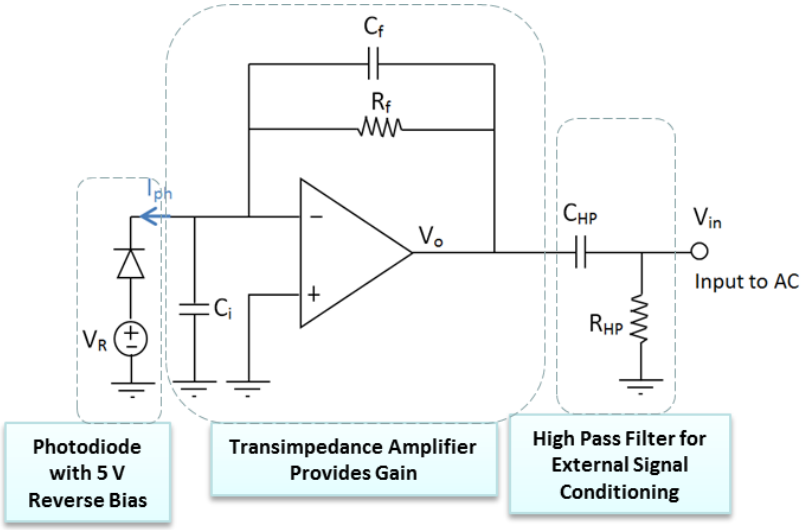


Figure 4.3 - Circuit schematic of the receiver component (detection, amplification, signal conditioning circuits).

4.3.1 – Photodiode

The receiver component was designed with a Silicon p-i-n photodiode, manufactured by Hamamatsu (Part# S6775-01). The detector has a relatively large area of 5.5 x 4.8 mm^2 and was rated to have a terminal capacitance of 40 pF at a 10 V reverse bias [46].

The photodiode generates a photocurrent, I_{ph}, from an incident optical signal according to

$$I_{ph} = R_\lambda * P_{in} * A_{Rx}. \quad (4.1)$$

R_λ is the detector's responsivity and is wavelength dependent. It provides a measure of how much current is generated per watt of incident power, P_{in}, on the detector's surface area, A_{Rx}. Specifically,

$$R_\lambda = \frac{\eta e}{hv} \tag{4.2}$$

where η is the quantum efficiency of generating an electron-hole pair per incident photon, e is the charge of an electron, and the product term hv is the energy of the incident photons.

The detector's responsivity as a function of wavelength is shown in figure 4.4. The spectral response range is from 700 nm to 1100 nm. The photodiode has a peak responsivity of .68 A/W at 960 nm, which is very close to the 980 nm VCSEL transmitter source used in this research. The spectral response range acts like a narrow band filter, as it filters out most of the visible light spectrum. Therefore, incident power from indoor visible light sources will generate a small ambient lighting noise current, i_{AN}.

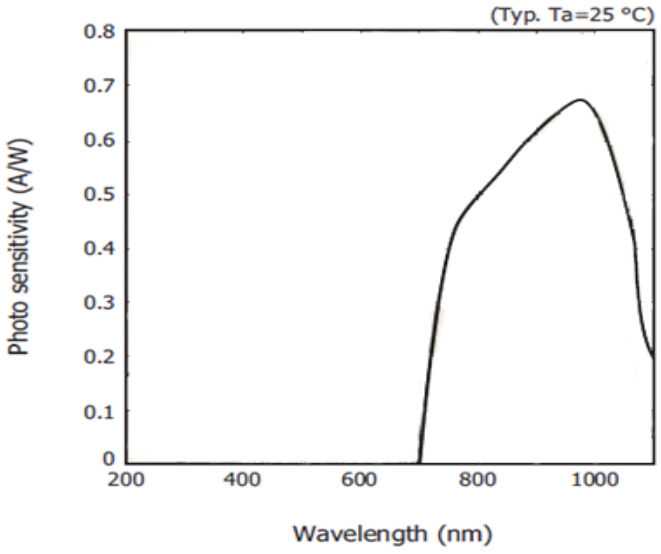


Figure 4.4 - Responsivity of S6775-01 Hamamatsu photodiode as a function of wavelength. *(Picture taken and edited from Datasheet* [46]*)*

4.3.2 – Transimpedance Amplifier Transfer Function

The transfer function, $T(s)$, relates the voltage at the input of the AC, V_{in}, to the generated photocurrent I_{ph}. The gain is shown as a function of s, which is the complex frequency jw. It can be derived as follows.

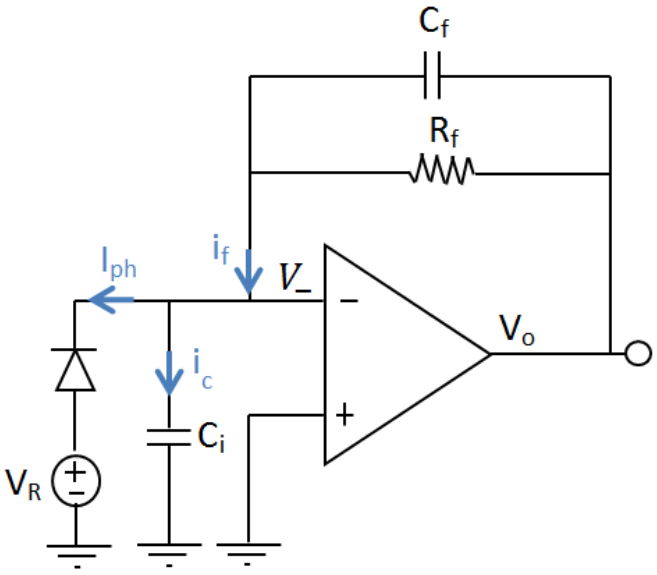


Figure 4.5 - Circuit schematic of transimpedance amplifier.

Referring to figure 4.5, the feedback current is

$$i_f = (V_o - V_-)\left(\frac{1}{R_f} + sC\right)$$

The three currents are related by

$$I_{ph} + i_c = i_f$$

$$I_{ph} + sC_iV_- = (V_o - V_-)\left(\frac{1}{R_f} + sC_f\right) \quad (4.3)$$

V_o and V_- are related together by the op-amp open loop gain function $A(s)$,

$$A(s)\,(V_+ - V_-) = V_0$$

Since V_+ is tied to ground,

$$V_- = -\frac{V_0}{A(s)} \tag{4.4}$$

Substituting 4.4 into 4.3 and solving for $a(s) = \frac{V_o}{I_{ph}}$ yields

$$T(s) = \frac{V_o}{I_{ph}} = \frac{A(s)R_f}{A(s) + sR_fC_fA(s) + 1 + sR_f(C_f + C_i)}. \tag{4.5}$$

If the op-amp gain is assumed to be infinite, 4.4 simplifies into

$$T_{ideal}(s) = \left(\frac{R_f}{1 + sR_fC_f}\right) \tag{4.6}$$

which is the more familiar forward transfer function that neglects non-ideal op-amp gain attenuation at high frequencies. The effects of an ideal and non-ideal op-amp gain are shown in figure 4.6.

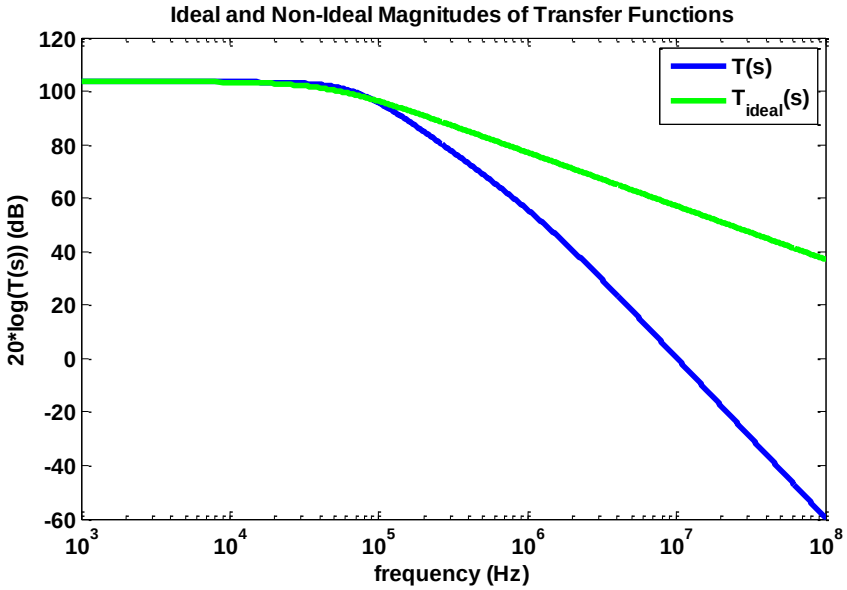


Figure 4.6 - Magnitude plots of the ideal and non-ideal transfer functions. In this simulation, C_i=67.5 pF, $C_f = 22$pF, $R_f = 150$kΩ, and $A(s)$ is the transfer function of the Analog Devices AD 8603 Op-Amp *(modeled in figure 4.7)*

$A(s)$ depends on which operational amplifier is being used, and in this research, the Analog Devices AD8603 operational amplifier was selected. $A(s)$ was modeled as a double pole transfer function,

$$A(s) = \frac{A_0}{\left(1 + \frac{s}{p_1}\right)\left(1 + \frac{s}{p_2}\right)} \tag{4.7}$$

where $A_o = 4 * 10^4$, $p_1 = 62.8319$, and $p_2 = 9.4248 * 10^6$. The transfer function was modeled based on experimental data from the datasheet [47]. Figure 4.7 shows the modeled open loop gain and phase of the AD8603 op-amp. Two important quantities are the gain bandwidth, which is the frequency bandwidth up to the point where the op-amp provides unity gain in amplitude (0 dB), and the phase margin, which is the phase difference between 180° and the phase of the transfer function at the unity gain amplitude point.

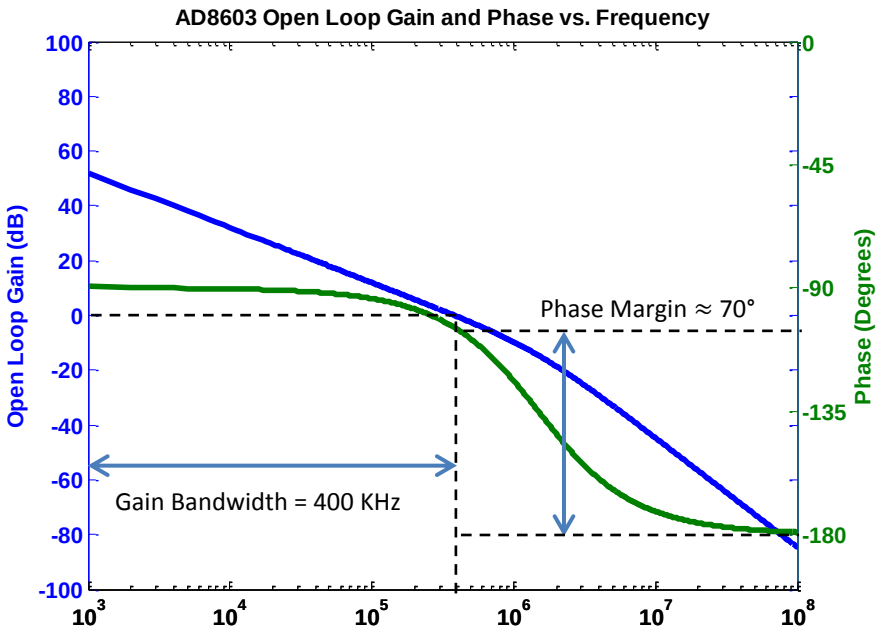


Figure 4.7 - Modeled AD8603 open loop-gain and phase vs. frequency plots.

4.3.3 – Transimpedance Amplifier Stability

Equation 4.8 can be algebraically manipulated into the form

$$T(s) = \frac{\frac{R_f}{1+sR_fC_f}}{1+\frac{1+sR_f(C_f+C_i)}{A(s)(1+sR_fC_f)}} = \frac{T_{ideal}(s)}{1+\frac{N(s)}{A(s)}} \tag{4.8}$$

where

$$N(s) = \frac{1+sR_f(C_f+C_i)}{1+sR_fC_f}. \tag{4.9}$$

$N(s)$ is referred to as the noise gain of the circuit [48] [49]. This representation of the transfer function provides information about the stability of the circuit. The

circuit is stable when $A(s) \gg N(s)$. When $N(s) \geq A(s)$, stability depends on the phase difference between the two functions. If $N(s) \geq A(s)$ and there is a 180° phase difference between these two functions, then

$$\frac{N(s)}{A(s)} \leq -1 \tag{4.10}$$

and this causes the denominator of the transfer function to become negative. When this happens, the circuit's output is prone to oscillate. The way to prevent instability is by making sure that at the frequency where $\left|\frac{N(s)}{A(s)}\right| = 1$, there is a phase margin close to 45° [50]. This can be done by selecting a proper C_f value, which effectively increases the phase margin of $\frac{N(s)}{A(s)}$. The effect of C_f on phase margin is shown in figure 4.8. As C_f increases, the phase margin of $\frac{N(s)}{A(s)}$ is increased over a wider bandwidth.

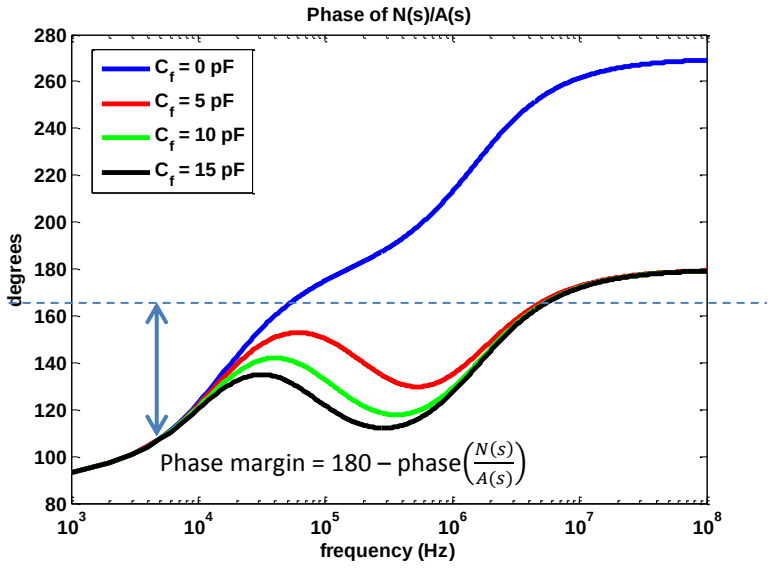


Figure 4.8 - Effect of three different C_f values on the phase of $\frac{N(s)}{A(s)}$

However, by increasing C_f, the 3-dB cutoff frequency (f_{3dB}) gets smaller, which reduces the overall effective gain bandwidth of the system. These effects are shown in figure 4.9.

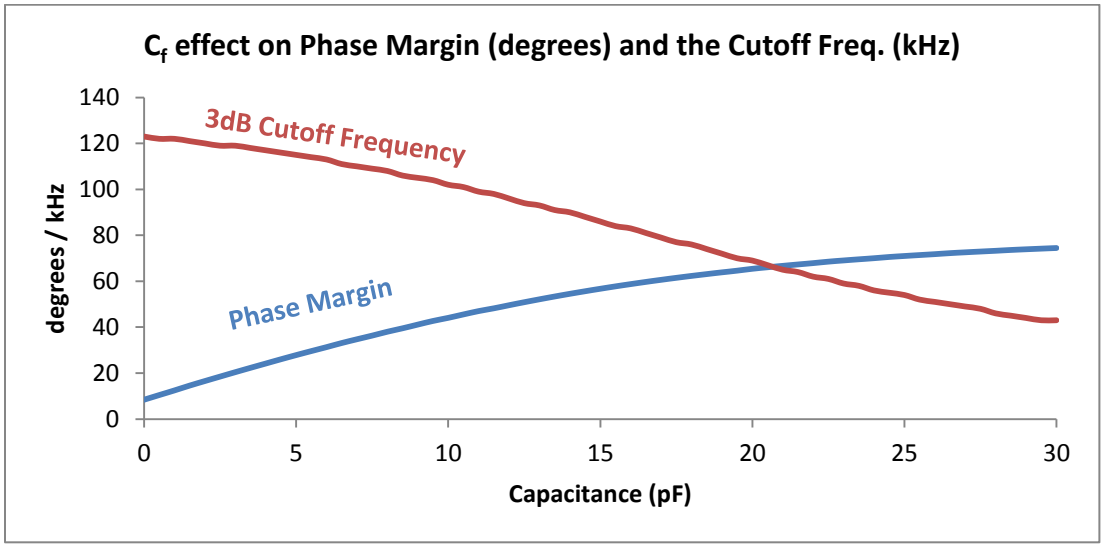


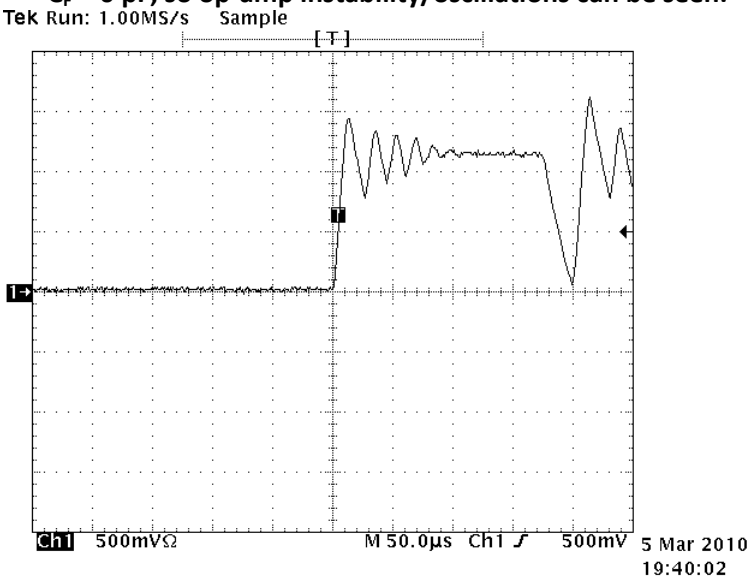
Figure 4.9 - Effect of C_f on phase margin (degrees) and cutoff frequency (Khz).

Figure 4.10 shows an example of transimpedance amplifier instability. In this experiment, an optical signal was transmitted to the photodiode, which converted it into an electrical signal. The waveforms shown are the outputs of the transimpedance amplifier and external high pass filter. Because the op-amp instability originates from high frequency amplification effects, the oscillations are visible near the edge of the input pulses. This is because a sharp rising pulse edge in the time domain contains many high frequencies. Each of these high frequencies is amplified and exhibits oscillation if the phase margin is near 0 degrees.

In direct detection optical communication links, data detection circuits rely on sampling the received voltage to determine if a logic "1" or "0" is being received. Op-amp oscillations can pose a problem for these circuits if the received signal

power is close to the threshold voltage. In situations like these, a logic "1" might be read incorrectly because of the oscillations crossing the threshold voltage. Thus, it's important to mitigate instability effects.

C_F = 0 pF, so op-amp instability/oscillations can be seen.



C_F = 15 pF, so the instability reduced even further.

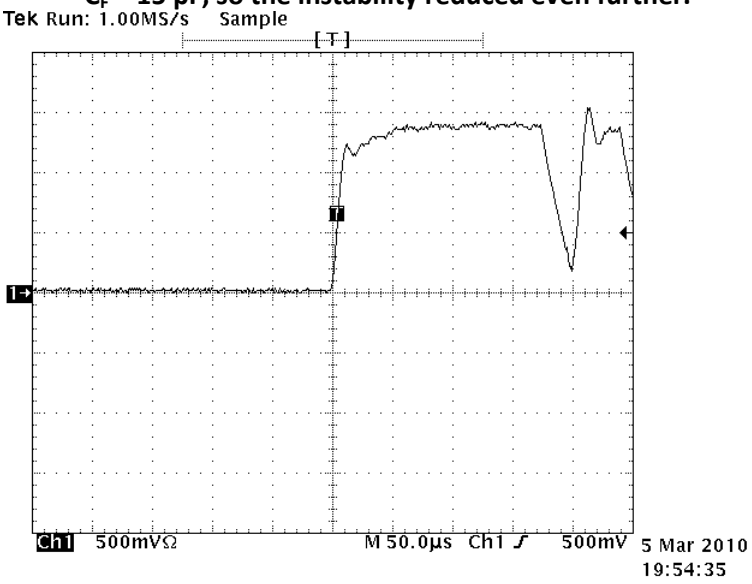


Figure 4.10 - Demonstration of transimpedance amplifier instability. In this demonstration, C_i=67.5 pF, $R_f = 120\text{k}\Omega$, and $A(s)$ is the Analog Devices AD 8603 op-amp transfer function.

4.3.4 – Signal Conditioning

The external high pass filter shown in figure 4.11, has a transfer function of the form,

$$F_{HP}(s) = \frac{V_i}{V_o} = \frac{sR_{HP}C_{HP}}{1 + sR_{HP}C_{HP}}. \tag{4.11}$$

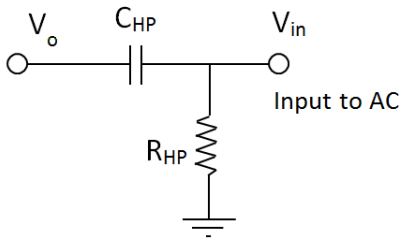


Figure 4.11 - Circuit schematic of external high-pass filter.

Combining this transfer function with the transimpedance amplifier transfer function $T(s)$, yields the total transfer function of the Rx circuit $T_{Rx}(s)$,

$$T_{Rx}(s) = \frac{V_i}{I_{ph}} = \frac{T_{ideal}(s)}{1 + \frac{N(s)}{A(s)}} * F_{HP}(s). \tag{4.12}$$

and is shown in figure 4.12.

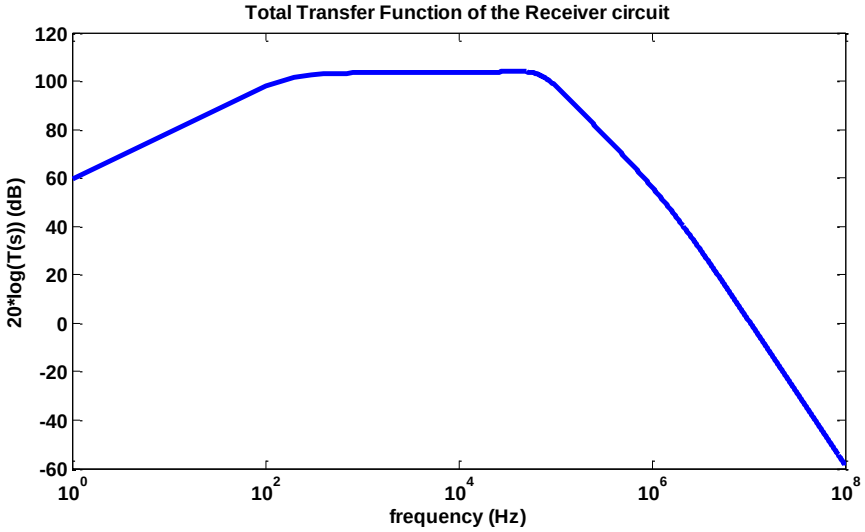


Figure 4.12 - The theoretical total transfer function of the receiver circuit.

The input to the system is the photocurrent I_{ph}. From the transfer function shown in figure 4.12, the system provides strong gains of 103.5218 dB (amplitude gains of 150,000) at frequencies near 10 kHz. Gains at frequencies lower than 10 KHz roll off at 10 db/decade, and the DC gain is 60 dB (amplitude gain of 1000), which is significantly less than 103.5218 dB. The higher frequency gain rolls off at 40 dB/decade.

4.3.5 – Noise Sources

A proper link analysis is fundamental in the design of any optical system to understand the potential noise sources and ensure enough optical power is present at the detector over the specified communication range. The developed model is based on experimental results and theoretical modeling. The major noise sources are listed in figure 4.13.

Noise Type	Noise Source
Ambient Lighting	Indoor visible light sources increase incident optical power at the photodetector
Amplifier Noise	Electrical amplification adds additional a noise current to the system
Thermal Noise	Random current motion within resistor elements add a noise current to the system
Shot Noise	Fluctuations in the photon rate of arrival add noise currents to the system

Figure 4.13 - Major noise sources present in FSOC system.

Ambient lighting increases the incident optical power at the photodetector. Based on experimental data, the ambient lighting noise was determined to have both low and high frequency components. Therefore, the low frequency noise was modeled

as a DC bias, while the high frequency noise was modeled as an AC noise source that had a frequency bandwidth equivalent to the frequency bandwidth of the peak gain of the total transfer function shown in figure 4.12. In this model, the DC bias ambient noise current I_A^{DC} can be approximated by measuring the voltage at the output of the transimpedance amplifier V_M when no optical signal is present, using the following expression

$$I_A^{DC} = \frac{V_M}{G(0)}. \tag{4.13}$$

Since the measurement is made at the output of the transimpedance amplifier and before the high pass filter, $G(0) = 103.5218\ dB$. V_M was measured to be 40 mV inside the laboratory, so $I_{A-DC} = 2.6667 * 10^{-7}$ A.

The AC ambient noise current $I_A^{AC}$was measured at the input to the analog comparator, after the external high pass filter and was modeled with the expression

$$I_A^{AC} = \frac{v_m}{G(10\ kHz)} \tag{4.14}$$

where $G(10\ kHz) = 103.5218\ dB$, and v_M, which was the peak AC ambient noise voltage detected at the input to the analog comparator, was measured to be 12 mV. Together, these values computed a value for the AC ambient noise current to be $I_A^{AC} = 8.33 * 10^{-8}$ A.

The current due to the shot noise can be modeled as [32] [51] [52]

$$< i_S >= \sqrt{2e(I_S + I_A + I_D)\Delta f_{Eff}} \quad (4.15)$$

where I_S is the received signal current, I_D is the photodiode's dark current and is approximated to be 1 nA from the data sheet [46], and Δf_{Eff} is the system's effective bandwidth. Using figure 4.12, Δf_{Eff} is approximated to be the bandwidth over which the gain is above 40 dB. The value of 40 dB was selected based on the assumption that the noise levels are spectrally constant, so an attenuation of 60 dB or more would render these high frequency noise spectral components irrelevant compared to the low and mid frequency noise spectral components. Thus, Δf_{Eff} is approximated to be 1 MHz.

The thermal noise current can be modeled as [32] [51] [52]

$$< i_T >= \sqrt{\frac{4K_B T}{R}\Delta f_{Eff}} \quad (4.16)$$

where K_B is the Boltzmann constant, T is the temperature in Kelvin, and R is the load resistance, which is 150 kΩ.

Finally, using the AD8603 data sheet [47], the amplifier noise current an be written as

$$< i_{Amp} >= .05e - 12\frac{pA}{\sqrt{Hz}} \quad (4.17)$$

The signal current I_S can be written as

$$I_S = R_\lambda * P_{in}. \tag{4.18}$$

All the noise currents and signal currents are shown in figure 4.14 with respect to the incident optical power.

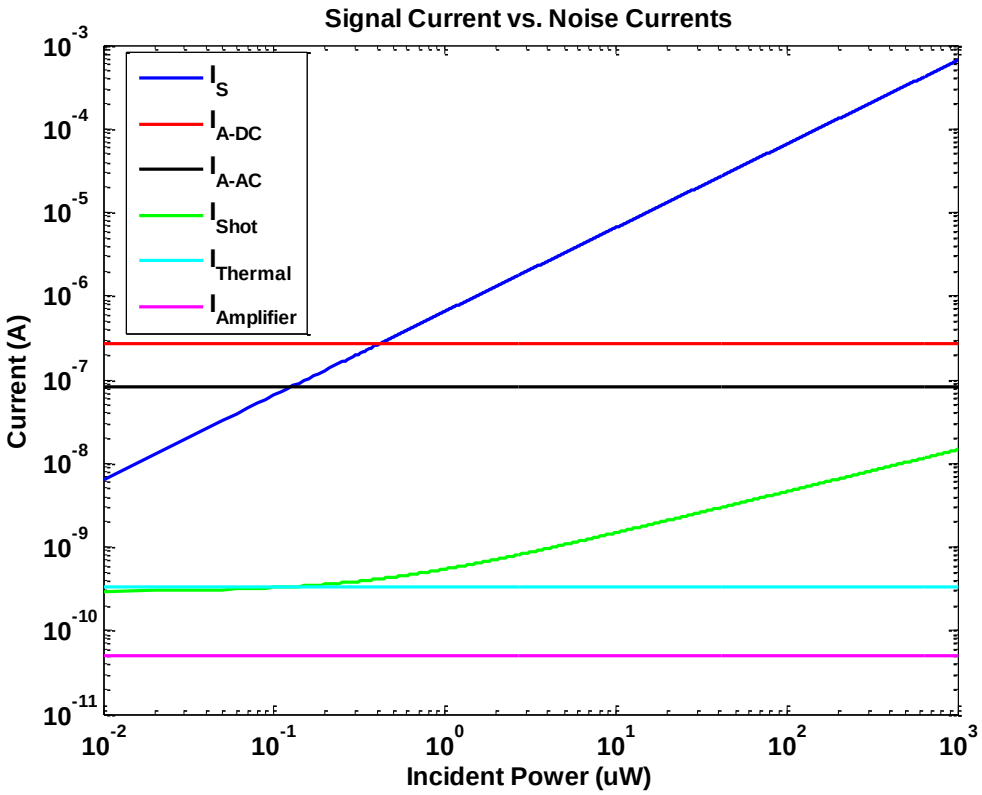


Figure 4.14 - Plot of signal and noise currents vs. incident optical power.

Figure 4.14 shows that as the incident optical power decreases, the DC and AC ambient noise currents become larger than the signal current. The DC ambient noise current is almost entirely removed with the external high pass filter, but the AC ambient noise current is not. The other noise currents (shot, thermal, amplifier) are not as large as the AC ambient noise current at these incident optical powers. Using this model, a threshold voltage of 25 mv, twice the peak AC noise voltage, was selected as the reference voltage input to the analog comparator. All received

voltage signals would be compared to this reference value in order to determine whether or not a logic “1” or “0” was being received.

4.4 – Data Processing and Controls Logic

The core of the system was implemented with an Atmel ATmega644p microcontroller (MCU). The core can be broken down into four main components: the decision circuit, which consists of the analog comparator and decision logic, demodulation logic, network controls, and modulation logic. All the software was custom designed to test all developed network protocols and algorithms. The decision circuit measured the pulse widths of logic “1”s and “0”s of the analog signal V_i and converted the measurements into a digital vector, which it stored in the MCU’s SRAM. The demodulation software processed the digital vector to recover the data bytes from the received signal V_i. Once the data bytes were received, the network controls processed the data and determined what action the hardware had to take next. If transmission was necessary, the modulation component generated the outbound packet, and then transmitted it to the laser driver circuit. Because the VCSEL transmitters required low threshold currents, the MCU could directly drive and modulate the VCSELs from its output pins.

4.4.1 – Decision Circuit

The inputs to the decision circuit are the conditioned input voltage V_i from the Rx component, and the threshold reference voltage, V_R which was set to ~25 mV. Whenever the input signal $V_i > 25$ mV, the AC triggers the AC interrupt service request (ISR) to run. At each AC-ISR run, a time stamp is generated and stored in a data register. Figure 4.15 shows an example where an incident signal V_i causes the AC-ISR to run 8 times and create 8 time stamps. The "black" time stamps correspond to rising edges while the "red" time stamps correspond to falling edges. The differences between the rising edges and falling edges are the durations of logic "1"s, and the differences between the falling edges and rising edges are the duration of logic "0"s. These time stamp differences are stored in a vector, which is then processed by the demodulation component.

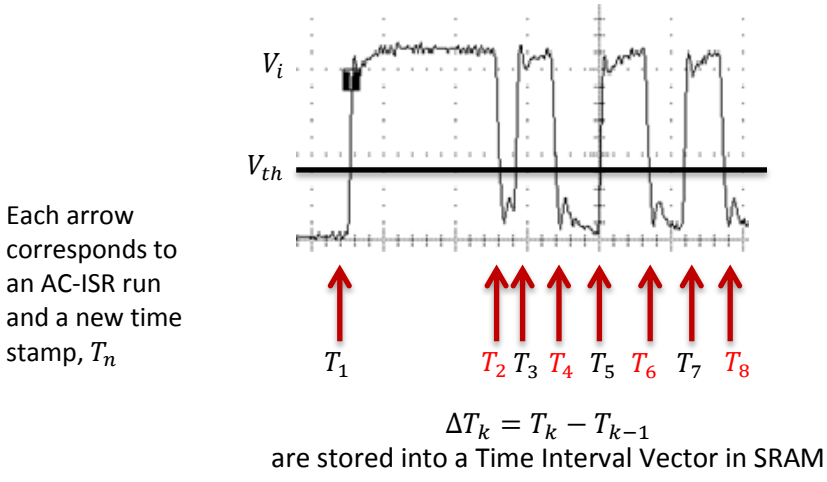


Figure 4.15 - Generated time stamps from an incident optical signal.

The decision circuit was implemented using a combination of hardware and software components. The hardware components are the analog comparator, and

two 8 bit counters: counter 0 (C_0) and counter 2 (C_2). C_0 and C_2 operate at 1/1024th and 1/8th of the main MCU clock rate, respectively. The purpose of C_2 is to provide a time stamp at each time the AC triggers the AC-ISR to run. A time stamp consists of the number of clock cycles that C_2 contains in its 8 bit register at the time that C_2 is read by the MCU processor.

The purpose of C_0 is to count the elapsed time between successive AC-ISR runs. Each time the AC-ISR runs, C_0 is reset back to 0. The idea is that once the last falling edge of the incident signal V_i is time stamped, AC-ISR will not need to run again, so C_0 would no longer be reset. Therefore, the value of C_0 would continue to increment to larger values than it could before when it was being reset by subsequent AC-ISR. The MCU reads this value in the main code (outside of the AC-ISR), and if the value of C_0 is greater than a preset threshold, then the MCU can assume that the last falling edge of the input signal V_i has been time stamped, and that the digitally stored vector represents the information of an entire packet of received data.

A functional diagram of the decision circuit is shown in figure 4.16. At the beginning of every AC-ISR run, a new time stamp, T_n, is generated by C_2. Before this new value is read from C_2 and stored into data register B (R_B), R_B loads its current value, T_{n-1}, which corresponds to the previous time stamp, into data register A (R_A). Then, C_2 loads T_n into R_B. In this manner, the values T_n and T_{n-1} can be used to extract the time interval of the received pulse.

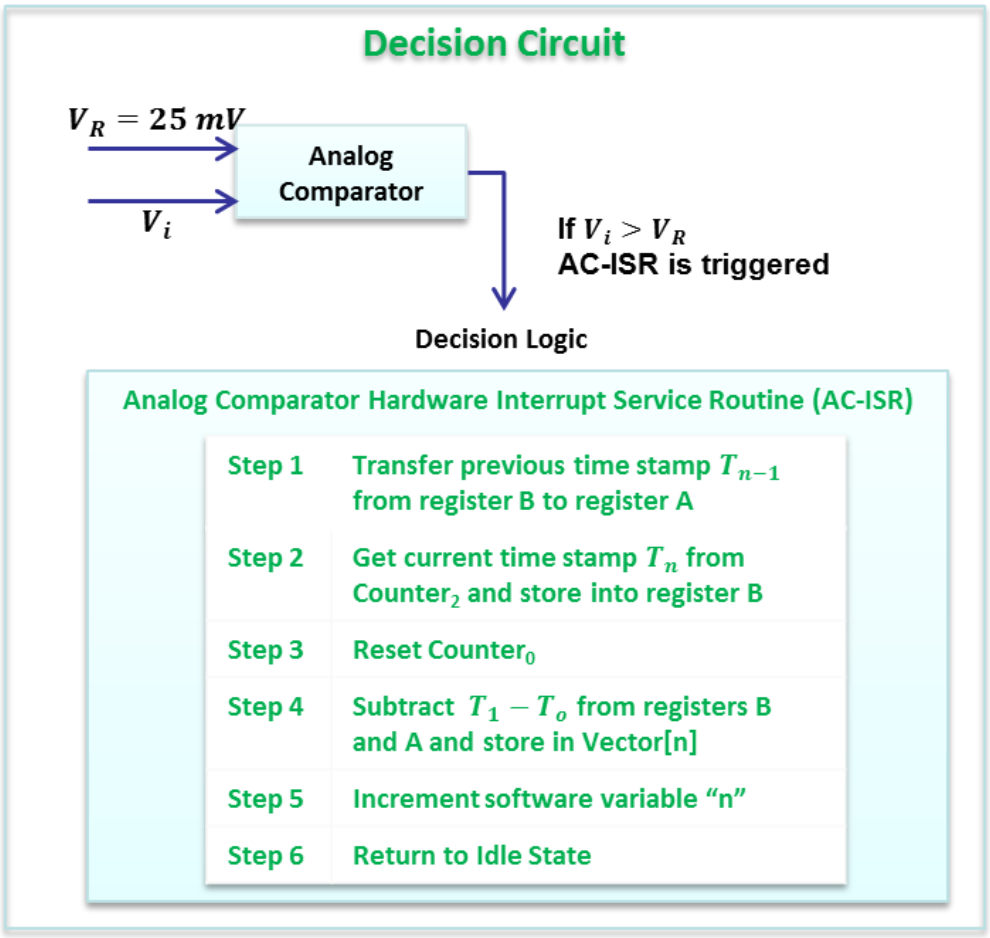


Figure 4.16 - Functional diagram of the decision circuit.

4.4.2 – Demodulation and Modulation

The input to the demodulation software is the vector generated by the decision circuit. The demodulation algorithm processes the vector in 3 steps. First, it recovers the clock rate information. Second, it uses the clock rate information to convert the pulse widths into binary "1"s and "0"s. Third, it converts the binary "1"s and "0"s into useful network commands, which it then stores in SRAM, so that the network controls component can process accordingly.

The packet structure plays an important part in the demodulation algorithm and is shown in figure 4.17. The start pulse signifies the beginning of the packet, and is all "1"s in a sequence of 16 bits. The synchronization frame allows the receiver system to extract the clock rate of the transmitted packet. In this way, the data rate can change and the demodulation algorithm can still detect the packets properly. The separator frame acts as a placeholder so that the receiver system knows that the demodulation algorithm knows the synchronization frame is complete. The header, data, and footer bytes are all mote-specific bytes that can dynamically change during network communications. Finally, the stop byte is an 8 bit all "1"s sequence that signifies the ending of the packet. A received packet waveform is shown in figure 4.18.

Frame Type	#of Bits	Binary Values
Start Pulse	16	**{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}**
Synchronization	20	**{00,1111,00000,11111,0000}**
Separator	8	**{1111,0000}**
Header Byte	8	**MAC ID – Mote specific**
Data Byte	8	**Data Byte 1 – Data specific**
Data Byte	8	**Data Byte 2 – Data specific**
Data Byte	8	**Data Byte 3 – Data specific**
Footer Byte	8	**MAC ID – Mote specific**
Stop Byte	8	**{1,1,1,1,1,1,1,1}**
Total Number of bits	**92**	

Figure 4.17 - Packet structure.

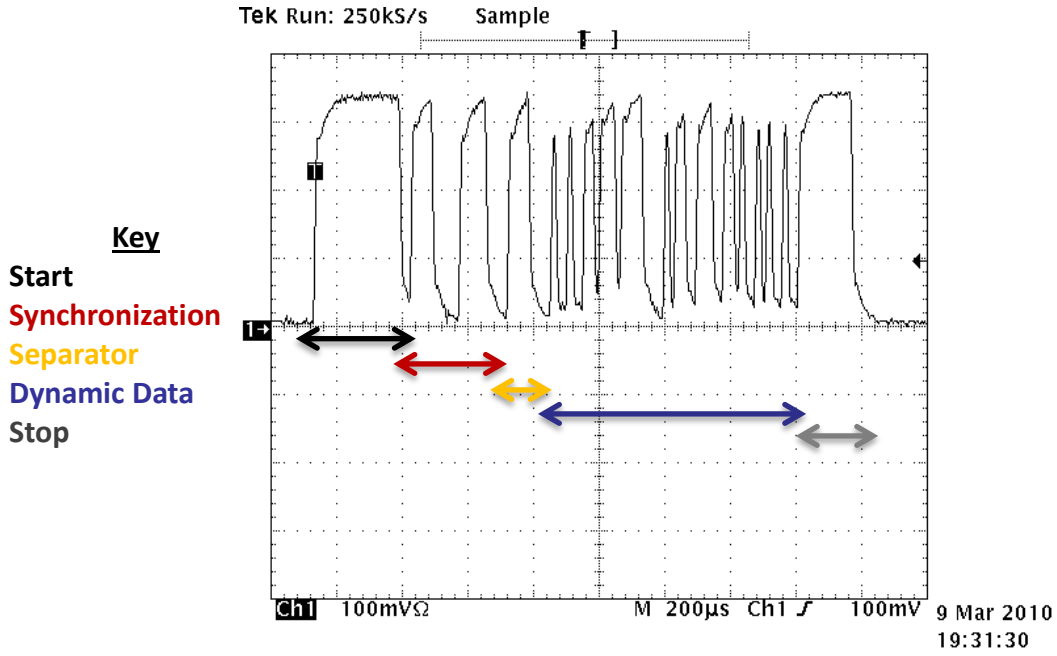


Figure 4.18 - Waveform of a received packet.

The demodulation algorithm works in the following manner. The algorithm reads the time interval vector (TIV) from the decision circuit. The first entry should be a large value, corresponding to the 16 bit all "1"s "Start" pulse. The synchronization frame is next, and the algorithm knows *a priori* that it should be expecting a pattern of {00,1111,00000,11111,0000}. Therefore, by matching values from the TIV to the synchronization sequence, the algorithm can determine the clock rate for logic "1"s and "0"s. With the clock rate discovered, the rest of the TIV values can be converted into the proper number of "1"s and "0"s, and these binary bits get stored in 8 bit SRAM so that the MCU can read and process the data at a later time.

The modulation logic component is nearly the inverse process of the demodulation circuit, in the sense that it converts 8 bit values stored in SRAM into logic "1" and "0" pulse widths, which it sends to the output pin. The output pin is set to +5 V when a

logic “1” pulse is being sent, and is set to ground when a logic “0” pulse is being sent. The modulation flow is shown in figure 4.19. The first step is generating the packet structure with the appropriate data. Once generated, the OC-ISR interrupt gets activated and begins streaming the pulse widths to the output pin.

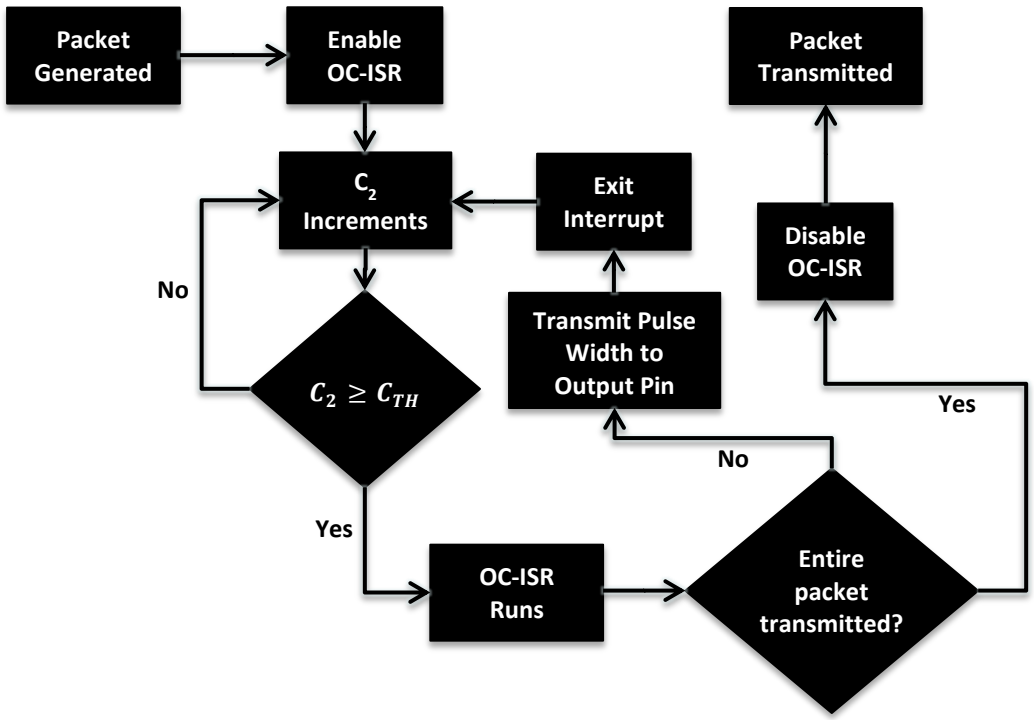


Figure 4.19 - Flow chart of the modulation algorithm.

Whenever C_2 gets larger than C_{TH}, the OC-ISR runs. During every OC-ISR run, only a single bit is transmitted, which means that the value of C_{TH} corresponds to the period of a transmitted bit, T_B. The bitrate, B, is then related by $B = \frac{1}{T_B}$, where $T_B = C_{TH} * T_{C2}$. T_{C2} is the period of a single clock cycle on C_2, and since C_2 operates at 1/8th the speed of the main MCU clock, $T_{C2} = 1/(\frac{8e6}{8}) = 1\ us$. Therefore, the optical communication bit rate is determined by

$$B = \frac{1}{C_{TH} * T_{C2}}. \quad (4.19)$$

From experimental verification, the fastest bitrate achievable using ISR routines with a main MCU clock of 8 MHz was ~100 kb/s, corresponding to $C_{TH} = 10$.

4.4.2.1 – Modulation Formats

The SR-FSOC systems were designed to operate with direct detection receivers. The possible types of modulation formats are on/off keying (OOK), pulse position modulation (PPM), and pulse duration modulation (PDM) [51] [52]. These schemes are shown in figure 4.20 representing three different decimal values in their respective modulation format. OOK is used in the SR-FSOC system hardware.

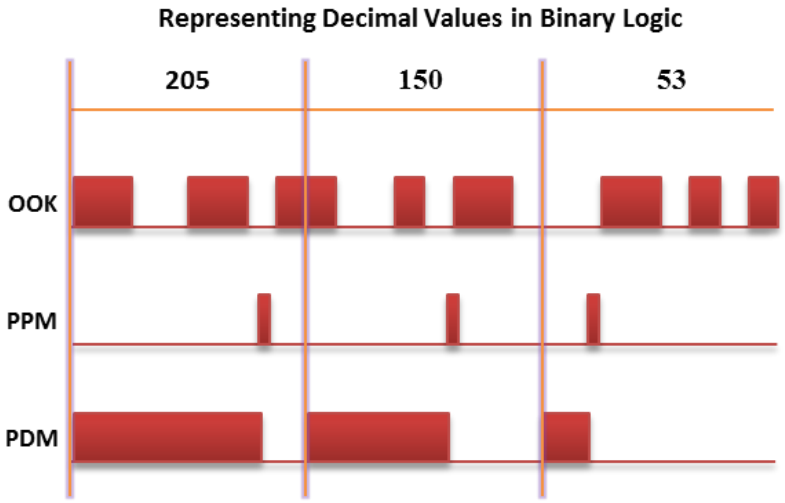


Figure 4.20 - Different direct detection modulation schemes. OOK: On-off keying; PPM: Pulse position modulation; PDM: Pulse duration modulation

The differences between these formats are how they encode decimal values. Assuming 8 binary bits need to be encoded, that would render 256 possible decimal

values. In a given time interval, OOK transmits the 256 possible decimal value by breaking the interval (ΔT) into 8 slots, where each slot corresponds to a binary bit. In this manner, the required bit rate becomes $B_{OOK} = \frac{8}{\Delta T}$. PDM encodes the information by varying the width of the optical pulse. Therefore, PDM requires 256 different pulse widths to encode 8 binary bits in a given ΔT. This implies that the shortest pulse width, which corresponds to the maximum bitrate, for PDM is $B_{PDM}^{max} = \frac{256}{\Delta T}$. PPM encodes the information by varying the position of a very short pulse within ΔT. This implies the condition that $B_{PPM} = \frac{256}{\Delta T}$, because the width of the pulse must remain within ΔT in order to maintain synchronous detection. Thus, in order for all three modulation formats to maintain the same throughput within the same ΔT, the optical bit rate bandwidth for PPM and PDM must be larger than the OOK format. Figure 4.21 shows the ratio of how much larger the bandwidth of PPM must be to maintain the same throughput as OOK for an increasing number of digital bits to encode an analog sample.

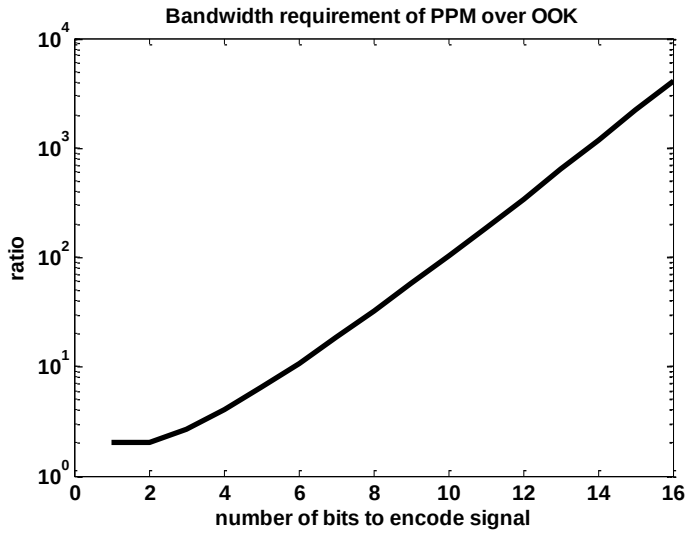


Figure 4.21 - Ratio of bandwidth requirements between PPM and OOK.

4.4.3 – Network Controls

After a signal has been detected and demodulated by either a mote or cluster head, the system processes the data to determine an appropriate response. Since the motes can only communicate to the cluster head, the cluster head controls the network communication flow. Specifically, the cluster head has the responsibility to identify which motes are requesting to transmit data along a time slot, determine whether or not multiple motes are contending for channel access, and to update the TDMA queue. The motes transmit a RTS command when they need to transmit data. Once an RTS command is understood by the cluster head, the cluster head transmits the synchronization signal to the mote, so that the mote can transmit back to the cluster head with its data in the proper time slot.

4.4.3.1 – Network Initialization and Initial Power On

Before any communication can take place, a network cluster has to be setup. This means that there must be at least 1 cluster head and 1 mote within range and field of view of each other. When the cluster head powers up, it enters into idle mode, where it awaits any incident optical signal. When a mote is initially powered up, it periodically transmits a "Request MAC" signal to the cluster head. Once the cluster head interprets the signal, it assigns a unique MAC address to the mote and adds the mote onto the network cluster. Upon receiving a MAC address, the mote stops transmitting a "Request MAC" signal and enters into its sleep state. This method

allows the cluster head to autonomously identify and assign motes unique MAC addresses whenever the motes are initially powered on within the network cluster. In this manner, if any mote needs to be replaced, it can simply be switched with another identical mote, and the cluster head can autonomously reassign the new mote.

4.4.3.2 – Cluster Head Network Flow

Figure 4.22 shows the flow for the cluster head's network processes. After receiving and demodulating the signal, the system verifies whether or not a packet has been received successfully. It does this by verifying that all data bytes have been read and that the start and stop pulses have been successfully detected. If the packet has been successfully received, the cluster head determines whether or not the received MAC address has been added to the network cluster. If it is not part of the network, the cluster assigns the device a unique MAC address and adds the device to the network, but if the received MAC address was already part of the network, then the cluster head processes the received data and selects the appropriate action. The possible actions could be any of the following:

1. Assigning a mote to a time slot
2. Removal of a mote from a time slot
3. Generating a backbone connection to an adjacent cluster head
4. Retrieving data from an adjacent cluster head

Instead, if the packet was not received successfully, the cluster head runs its packet collision detection algorithm to try to determine whether or not the failed packet was caused by channel contention. (This process was explained in chapter 3). If the algorithm determines that channel contention was the reason for the failed packet, the cluster head runs its random access protocol algorithm to assign time slots to the other contending motes, but if channel contention was not the reason, then the cluster head reprocesses its last action.

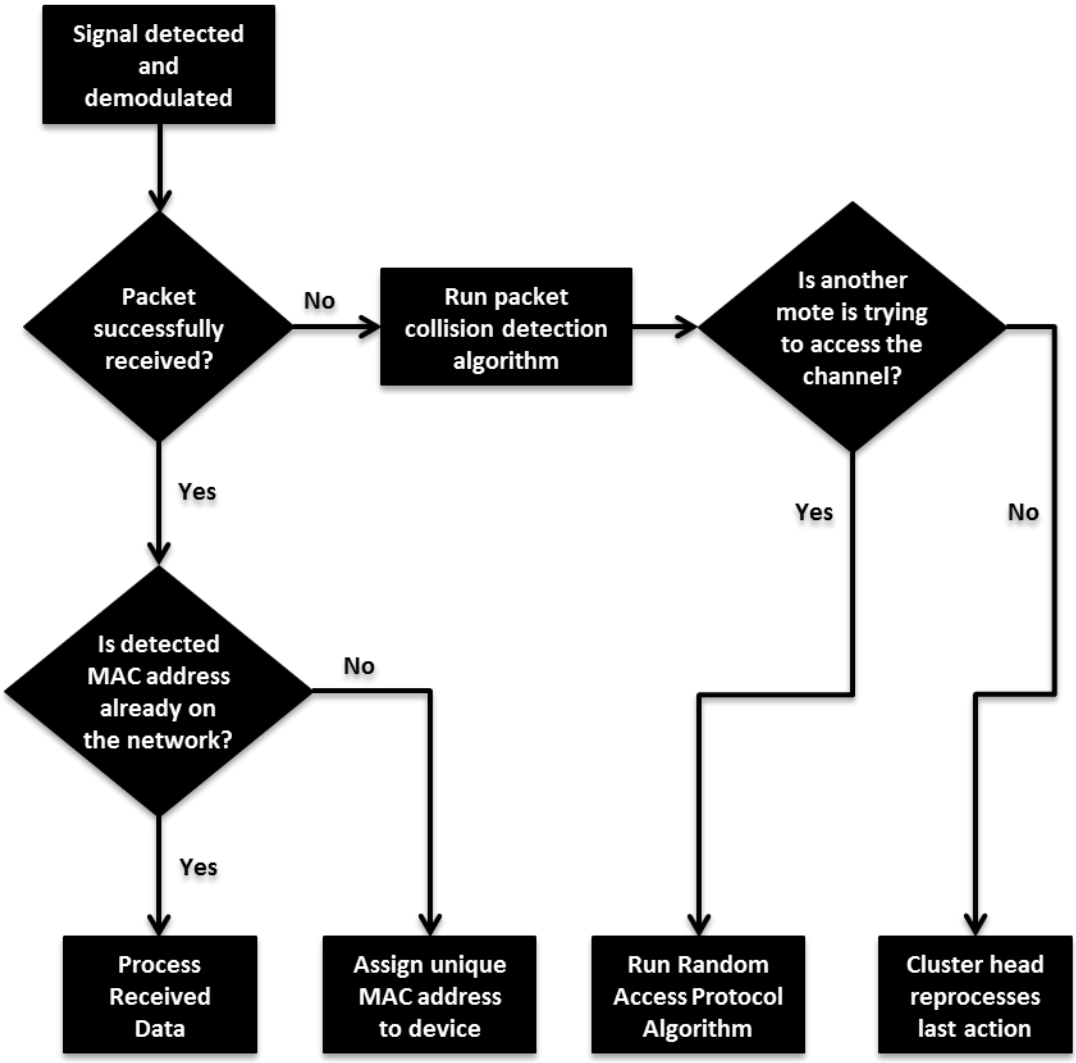


Figure 4.22 - Flow chart of the cluster head's network processes.

4.4.3.3 – Cluster Head Data Processing

To streamline the data processing, all outbound and received communications with a particular device gets stored in a channel vector corresponding to that particular device's MAC address. The vector is shown functionally in figure 4.23. The number of channels is equivalent to the number of active time slots. When a device no longer needs a time slot, the channel is erased. Based on the byte values stored in Rx1, Rx2, and Rx3, the cluster head determines its response and updates the outbound byte values, Tx1, Tx2, and Tx3.

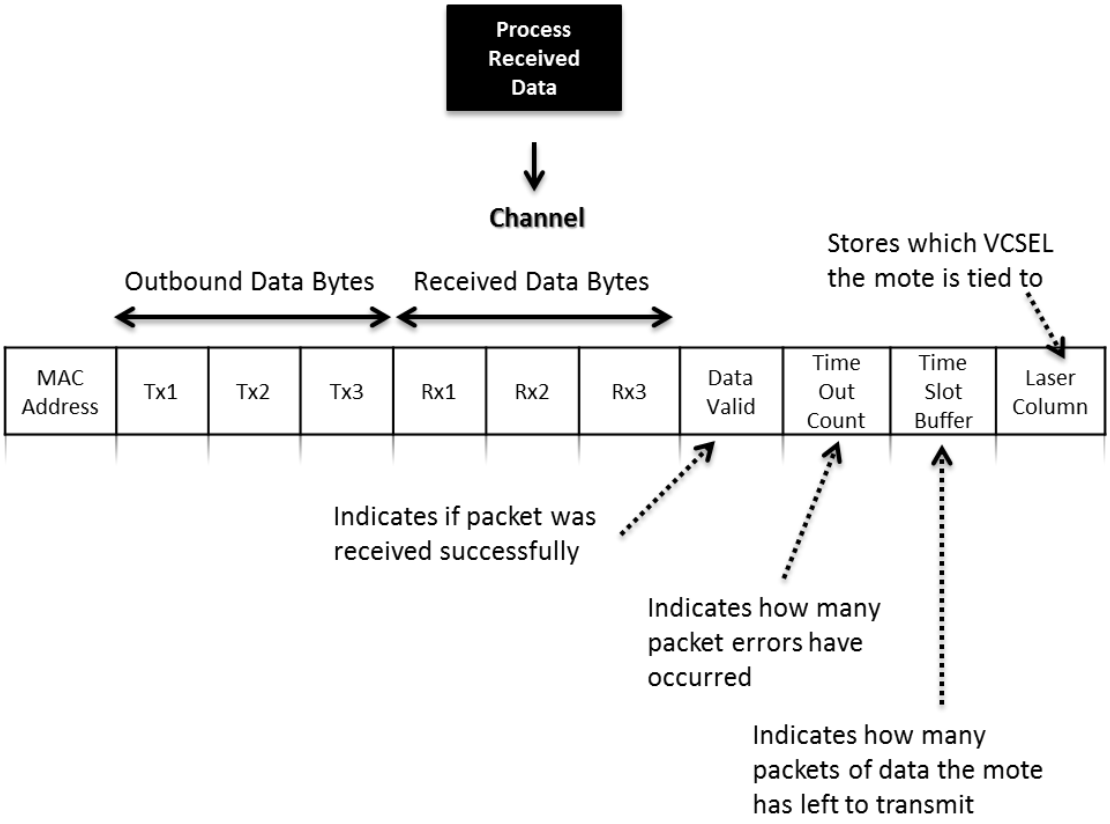


Figure 4.23 - Illustration of the cluster head's channel vector, where all network is stored.

4.4.3.4 – Random Number Generation

The random access protocol algorithm requires the motes to generate probabilities, which are used to determine whether or not the motes will reply to a cluster head's RAS signal. The probabilities were generated using a standard macro from the <stdlib.h> header file. The pseudo random number generator works in the following manner. The function call "ran_var = *rand()",* generates a sequence of pseudo-random numbers and returns a pseudo random number to the variable "ran_var". Subsequent function calls of "ran_var = *rand()"* extract different numbers from that same sequence of pseudo-random numbers. Because the sequence is finite, eventually the pattern will repeat, hence the name pseudo-random. To reduce this effect of repetition, the function call "*srand(*Change_Sequence*)"* can change the sequence that is being used by the *rand()* function call based on the input variable "Change_Sequence". Therefore, if the function "*srand(*Change_Sequence*)"* is called before "ran_var = *rand()"*, the returned pseudo random number can be extracted from different sequences, rather than from the same sequence. This can potentially lessen the chance of "repetitions" in returned pseudo random numbers as long as the "Change_Sequence" variable is being varied at each "*srand(*Change_Sequence*)"* function call. To do this, the system was setup so that whenever a pseudo random number was needed, the motes set their "Change_Sequence" variable to the value of the C_2. The idea is that since communication is taking place at random times, the C_2 value would always present a different value, and therefore "*srand-*

*(*Change_Sequence*)"* function calls could randomly vary the sequences of pseudo random numbers that the function call "ran_var = *rand()"* would extract from.

These random numbers would then be converted into a probability by

$$p_{gen} = mod(ran\_var, C(p_{trans})) \tag{4.20}$$

where *mod* is the modulo operation and *C* is an integer that depends on what p_{trans} value the cluster head transmitted during the RAS signal, and is given by,

$$C(p_{trans}) = floor(p_{trans} * 256). \tag{4.21}$$

Since all possible p_{trans} values are known *a priori*, these values can be computed at initial power up and stored in SRAM for quick processing.

Each "ran_var = *rand()"* function call is a random event. This means that only the average values of the results of the function call will reflect the desired generated probability. Figure 4.24 shows the averaged generated probabilities from the motes' pseudo-random number generator algorithm averaged over 1000 function calls.

Mote #	Probability (Analytical)	Probability (Experimentally Measured)	Relative Error (%)
1	1.00	1.000	0
2	.500	.496	-.80
3	$.33\overline{3}$	.356	6.91
4	$.25$	.237	-5.20
5	.200	.194	-3.00
6	$.16\overline{6}$	.142	-14.46
7	$.\overline{142857}$	.140	-1.81
8	.125	.126	.80
9	$.11\overline{1}$	.126	13.51
10	.100	.117	17.00

Figure 4.24 - MCU generated probabilities averaged over 1000 random generation function calls.

4.5 – Transmission Component

Vertical cavity surface emitting lasers (VCSELs) were used as the transmission sources. VCSELs have low threshold currents, allowing them to be directly driven and modulated by a low power MCU. In this manner, a separate current driver circuit is not necessary for each transmitter. The maximum modulation rate of the VCSELs is in the GHz range, while low-power MCU clock speeds are in the 1-200 MHz range, so the modulation rate is MCU dependent.

Because of the low threshold currents, the output pin of the microcontroller was able to directly drive and modulate the VCSEL transmitters. The circuit is shown in figure 4.25. The value of R_D is selected based off the desired VCSEL operating current. In this research, the VCSELs had threshold currents near 1.5 mA [53] and were driven at ~9.5 mA. The absolute maximum current the MCU was rated to source per pin was 40 mA, which was well above the VCSEL driving current.

$$R_D = \frac{V_{CC} - V_D}{I} = \frac{5.2 - 1.8}{9.5} \approx 358\ \Omega \tag{4.22}$$

where V_{CC} is the voltage at the MCU output pin, and V_D is the approximate voltage drop across the p-n junction. 374 Ω resistors were used in the actual hardware.

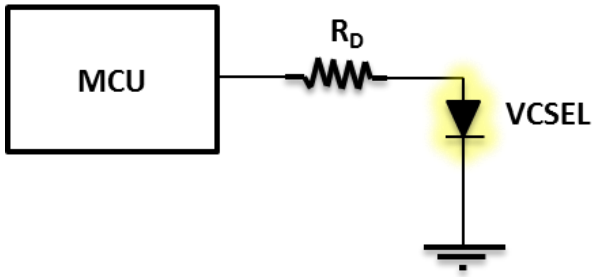


Figure 4.25 - Schematic of the transmission circuit. Cluster head has 5 such circuits, with each circuit connecting to a different output pin on the MCU.

4.6 – Hardware Implementation

In order to create functioning SR-FSOC hardware capable of integrating all developed network algorithms and protocols, a series of prototype devices were designed, implemented, and experimentally tested to evaluate the different components of the proposed optical communication system. These components were the data recovery circuitry, communication protocols, network algorithms, and transceiver architecture. With each successful implemented component, a final version was developed.

The SR-FSOC hardware was implemented onto printed circuit boards (PCBs). The schematic and PCB layouts were drawn using the software "PCB Artist" supplied by Advanced Circuits [54]. Once the layouts were finalized, the schematics were sent

out to Advanced Circuits, who supplied the PCB boards. Figure 4.26 shows the final version of the developed schematic for the cluster head using the "PCB Artist" software. The mote schematic is the same except there is only 1 laser transmitter connected to the MCU, so the "Tx" component has only 1 laser diode drawn instead of the 5 seen in figure 4.26.

Figure 4.27 a) and b) show the final version of the developed PCB layout using the "PCB Artist" software and actual PCB, respectively. The PCB layout consists of 4 planes. The surface plane and back planes are used for signal tracks, while the inner 2 planes serve as ground and $+5V_{CC}$, respectively. The red lines and light blue lines correspond to signal tracks along the surface and back plane, respectively. The yellow holes are "vias", which are gateways that allow signal tracks to connect from one plane to another. The red squares are the surface mount pads that the surface mount chips get soldered to, and are known as PCB footprints. Each footprint has a name that corresponds to the particular component that will be soldered there. These components and values are shown in figure 4.28.

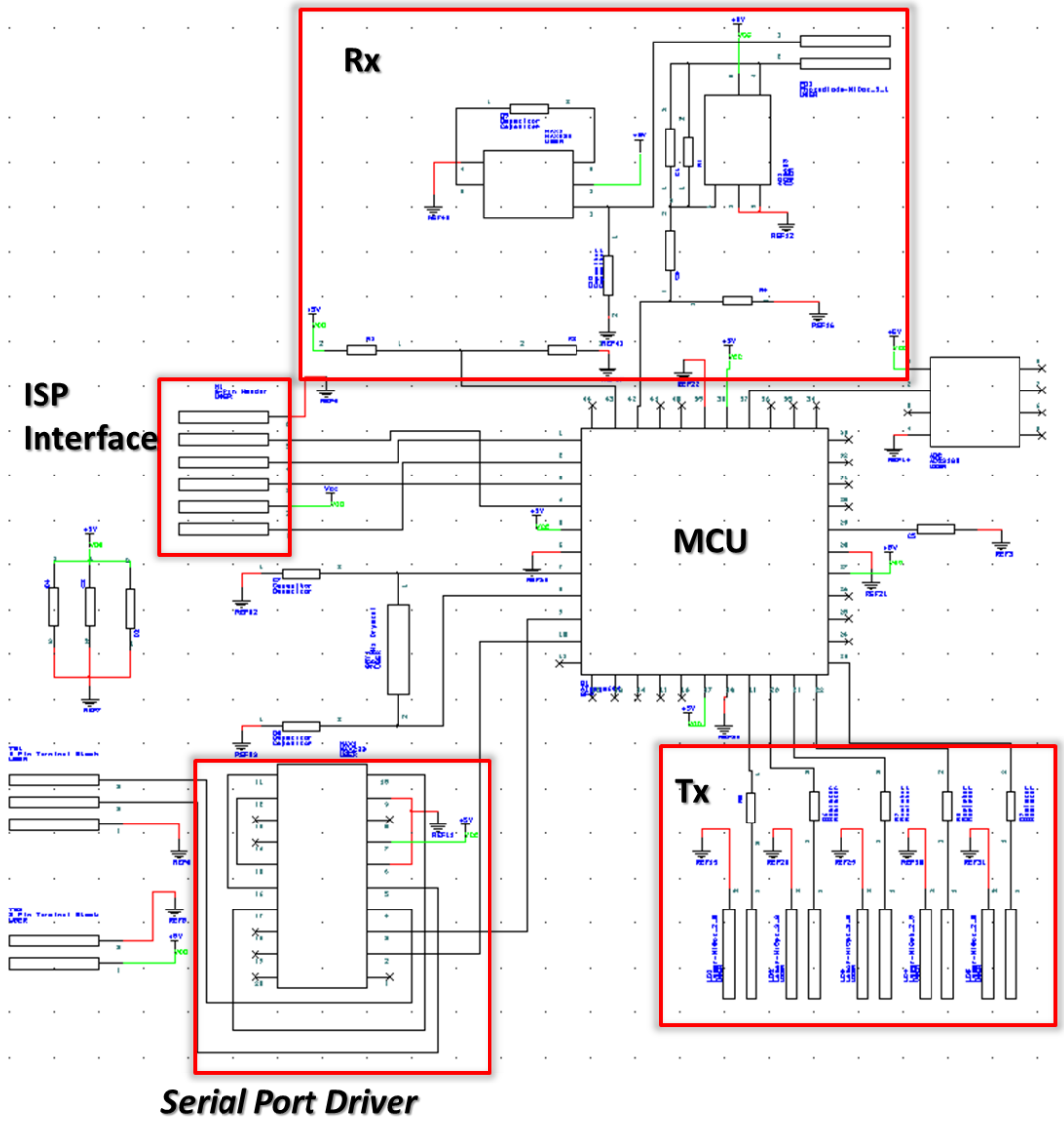
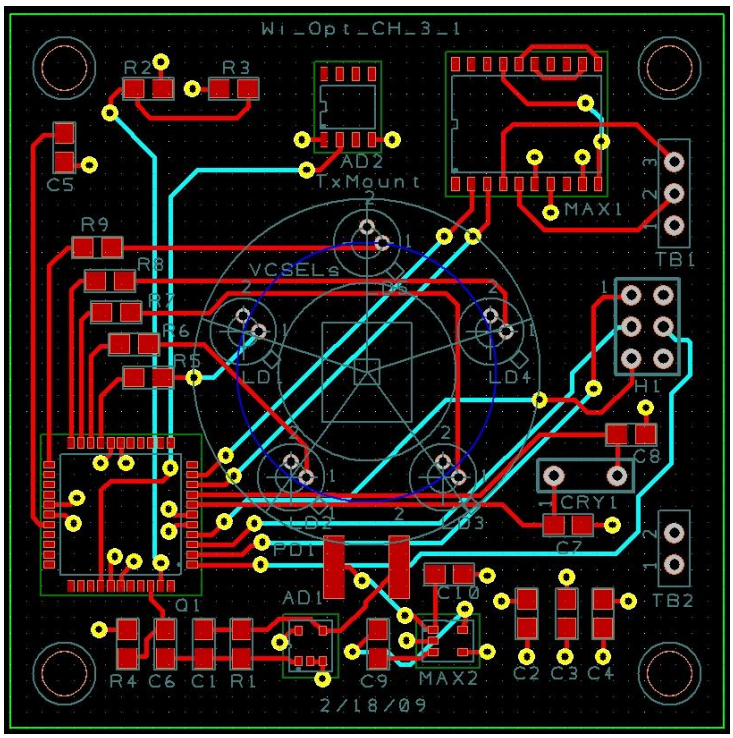
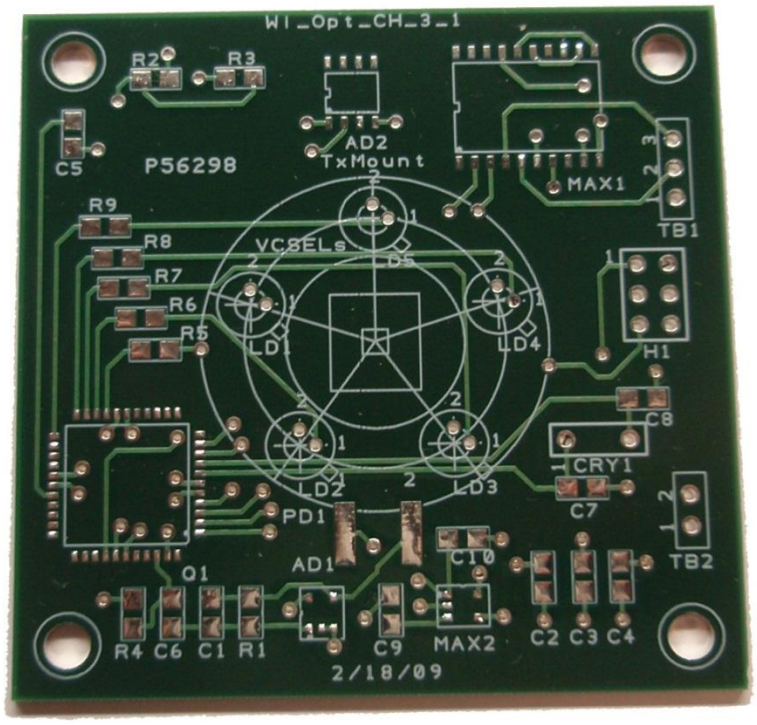


Figure 4.26 - Schematic Layout of Entire Cluster Head Circuit *(ISP Interface: In System Programming Interface)*

Since the main focus of these devices was to test the performance of the algorithms, the circuit designs were laid out with several debugging interfaces, including a serial port to interface with a PC and an in system programming interface (ISP) so that the MCU could be reprogrammed after being soldered onto the board.



a)



b)

Figure 4.27 - a) PCB layout developed using “PCB Artist”. b) Actual cluster head PCB

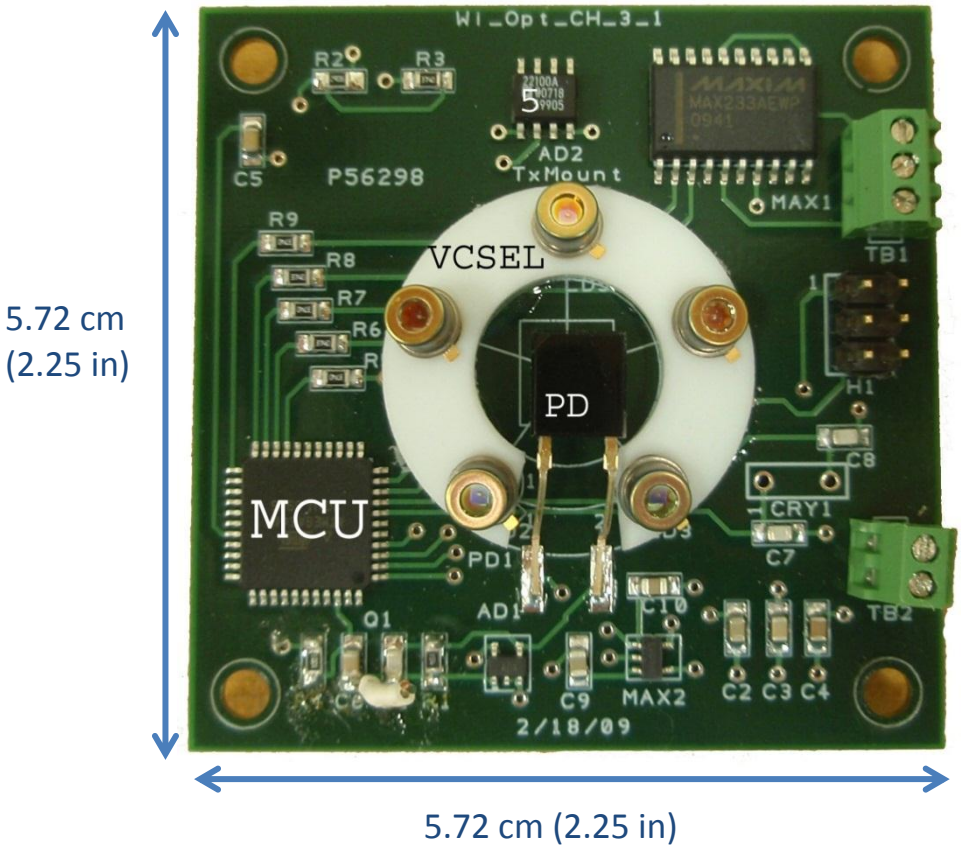
PCB Footprint	Chip	Vendor	Type
MAX1	MAX233A	Maxim-IC	Serial Port Driver
MAX2	MAX829	Maxim-IC	Voltage Inverter
AD1	AD8603	Analog Devices	Op - Amp
AD2	AD22100	Analog Devices	Temperature Sensor
Q1	ATmega644p	Atmel	Microcontroller
LD	VCSEL 980 nm	Thorlabs	Laser Diode
PD	S6775-01	Hamamatsu	Photodiode
CRY1	HC-49US	ECS	20 MHz Crystal Osc.

Connector	Type
H1	ISP Interface
TB1	Serial Port Interface
TB2	System Power Interface

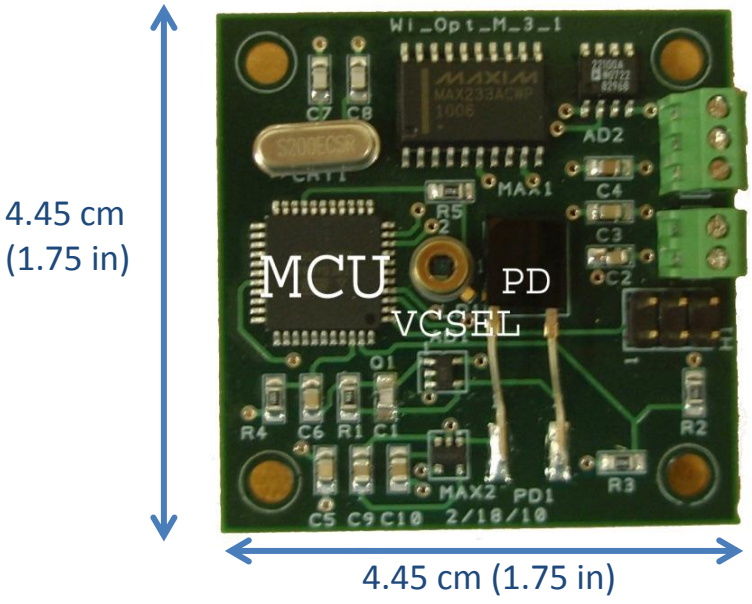
Resistors	Value	Capacitors	Value
R1	150 kΩ	**C1**	15 pF
R2	80 kΩ	**C2**	1 uF
R3	374 Ω	**C3**	.1 uF
R4	1 kΩ	**C4**	.01 uF
R5	374 Ω	**C5**	.1 uF
R6	374 Ω	**C6**	1 uF
R7	374 Ω	**C7**	15 pF
R8	374 Ω	**C8**	15 pF
R9	374 Ω	**C9**	3.3 uF
		C10	3.3 uF

Figure 4.28 - Components and their values of the cluster head system.

A PCB stencil was used to place solder paste onto the PCB. Then, the surface mount chips were laid onto the PCB using the "ezPick vacuum SMT Pick and Place System" [55], which is a manually controlled machine that picks the chips up using suction from a vacuum pump. Figure 4.29 shows the fully laid out cluster head and mote hardware, and figure 4.30 shows the system specifications.



a)



b)

Figure 4.29 - a) Cluster head system. b) Mote system.

Hardware Specifications			
MCU Clock Speed	8 MHz	Photodiode Area	5.5 x 4.8 mm^2
VCSEL Power	2.6 mW @~9.5 mA	Wavelength	980 nm
Beam Divergence	~10º	Quantum Efficiency	.68
Modulation Format	On/Off Keying	Trans-impedance Gain	150 kΩ
Data Rate	100 kb/sec	Range	Up to 2 m

Figure 4.30 - The hardware specifications of the cluster head and mote systems.

4.7 – Throughput Analysis

The optical data rate is 100 kb/s, but it takes a finite amount of time to demodulate, process, modulate, and transmit a signal, so the actual throughput is less than 100 kb/s. The throughput was experimentally measured by reading the value of C_1 after the signal propagated through various components of the SR-FSOC hardware, and then multiplying the value of C_1 by its appropriate time scale. Figure 4.31 shows the throughput diagram showing the lengths of time required for the various components. These times are based off the MCU running at 8 MHz.

The detection circuit adds a delay because it takes a finite amount of time to determine whether or not the entire packet was received. Furthermore, the detection circuit delay is longer on the cluster head (.896 ms) as compared to the motes (.640 ms). The reason for this is because the cluster head may receive signals from more than 1 mote at slightly different times, so that the combined signal from the motes exceeds the average time for a single packet. It was experimentally found

that increasing the detection circuit delay greatly increased the accuracy of the packet collision detection algorithm.

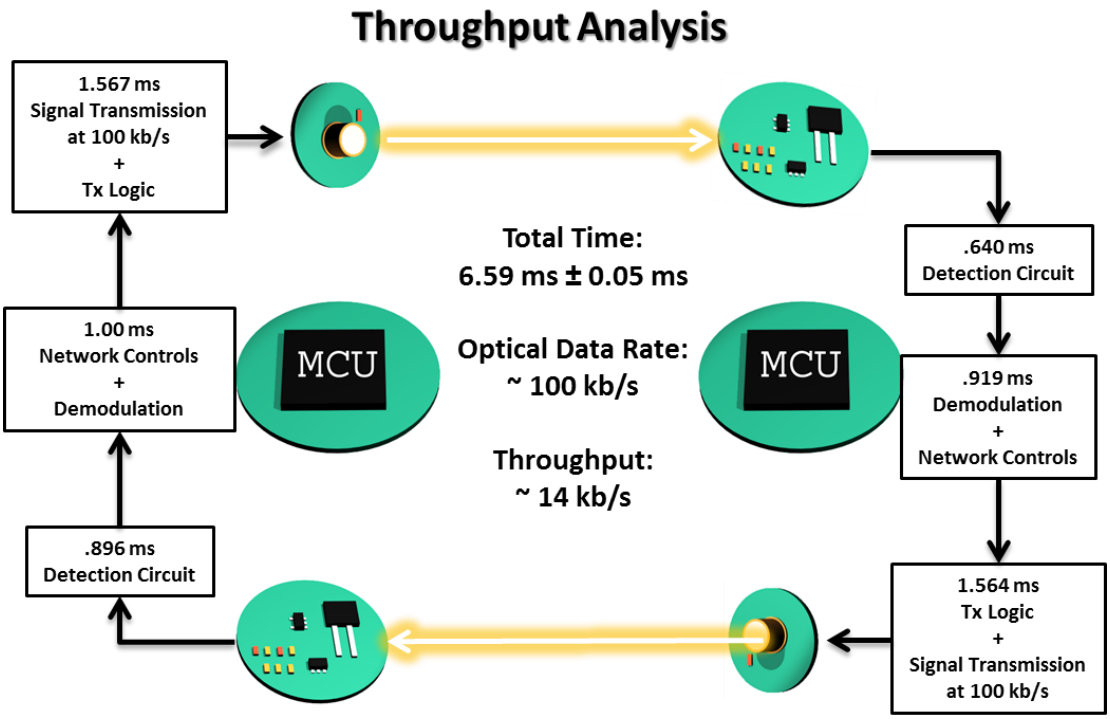


Figure 4.31 - Analysis of time required for the cluster head to transmit a packet to a mote, receive a signal back from the mote, process the received signal, and prepare to transmit another signal to the mote.

Figure 4.31 depicts the amount of time it takes for a cluster head to transmit a command to a mote, receive a reply back from the mote, and then process the reply to select its next action. The total time takes approximately 6.59 ms, providing a data throughput of ~14 kb/s. This process corresponds to the total time interval for a synchronization frame, which plays an important role in the random access protocol algorithm.

4.8 – Photovoltaics

Solar/laser recharging capability can be added to the system by incorporating photovoltaic cells to produce photocurrents [45] and by having super capacitors store the generated charge. The rate of charge depends on the incident light, area of the solar cell, and responsivity at each wavelength. The incident light can be the input optical beam as well as the ambient lighting. Once the super capacitors are charged, a controlled discharge circuit can allow the MCU to transmit a packet without sourcing current from its output pin to the laser diode circuit. Figure 4.32 shows an example of such a circuit. In this circuit, the MCU uses the signal to open and close an IC switch which provides the pathway from the super capacitor to the laser diode. Assuming the IC switch has a large input resistance, the MCU would not need to source much current to drive it. The linear regulator ensures that a constant current source is maintained.

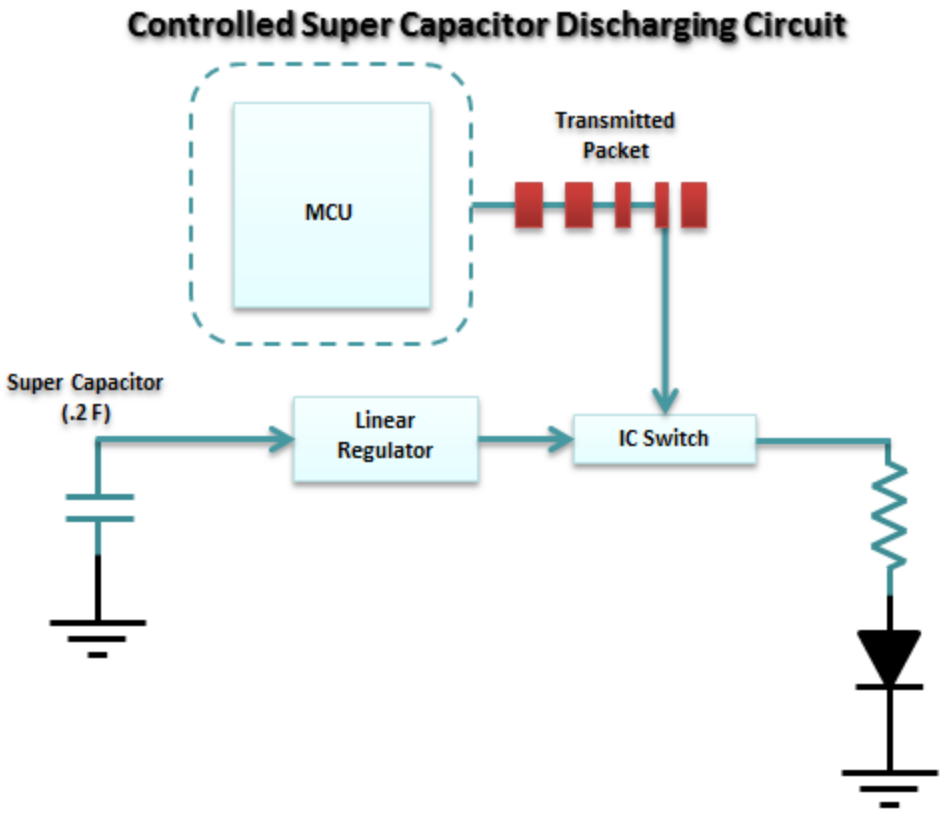


Figure 4.32 - Schematic of a MCU controlled super capacitor discharging circuit to drive the VCSEL transmitter.

The controlled super capacitor discharge circuit shown in figure 4.32 was tested with a fully charged super capacitor (~5V). The mote was able to transmit 63408 packets at 100 kb/s, when the super capacitor discharged from 5.021 V to 2.88 V. That translates into approximately 5.8 megabytes of information ($63408\ packets * 92\ bits/packet$) on a single discharge.

Chapter 5 – SR-FSOC Network Performance

5.1 – Introduction

This chapter presents the experimental performance results of the SR-FSOC sensor network. All communication protocols and network control algorithms were directly implemented into the SR-FSOC hardware, via the ISP interface. More specifically, the FSO DMAC protocol was implemented and embedded into both the cluster heads and the motes. The cluster head uses dynamic TDMA to accumulate data from the different motes and the developed FSO-based RA-protocol to handle the randomness of when the different motes wake up and transmit to the cluster head.

Section 5.2 shows the experimental setup and initial experiment that verified that system hardware and network communications were working as intended. The experiment's objective was to verify that collisions were detected and channel contention was resolved as analytically predicted for a known network configuration. With the network communications working as intended, section 5.3 discusses how the FSOC DMAC algorithm was modified so that it could deal with more practical network configurations. The section presents the algorithm's parameters and how they affect network performance. Section 5.4 presents experimental results of the implemented RA protocol, and discusses a method of optimizing the algorithm parameters based on minimizing latency and energy

consumption. The experiment's objective focused on examining how well the RA-protocol could resolve channel contention.

Section 5.5 presents experimental results of the implemented FSOC DMAC algorithm. In this experiment, the network performance of a single network cluster is evaluated by having different numbers of motes wake up at different times and transmit packets to the cluster head. The experiment focused on examining how much of a delay the RA-protocol inserted into the overall D-TDMA traffic queuing. Furthermore, the section provides an analysis of how the cluster head can optimize the RA-protocol's parameters in real-time based on detected network traffic.

Finally, the chapter concludes with a demonstration of how the RA-protocol can be modified to handle larger network cluster sizes.

5.2 – Single Network Cluster Experimental Setup

The experimental setup of the SR-FSOC system is shown in figures 5.1 – 5.7. A single cluster network is setup with 10 motes placed within the field of view of the cluster head. Figures 5.1-5.3 show the 10 motes from several different views. Figure 5.4 shows the group of 10 motes facing the cluster head. Some motes were slightly tilted and others were raised off the table in order to point at the cluster head, which was located 10 inches above the optical table. The motes are spread out over a circular area with a radius of ~30.5 cm and are positioned .8 m to 1 m in front of

the cluster head. The cluster head is shown in figure 5.5. Figures 5.6-5.7 show the entire network cluster. In these figures, the white and black twisted pair wires are power lines. For all experiments, a 5V power supply powered each mote and cluster head. All random access protocol measurements were made on the cluster head, and then transmitted to a PC via a serial cable. These measurements consisted of:

- Number of SFs required to connect all 10 motes
- Number of packet collisions detected
- Number of timeouts
- Number of packets successful received

The room was lit with standard office lighting producing a noisy bias voltage of 40 mV at the cluster head's detector and other assorted high frequency noise (see chapter 4). Figure 5.8 displays a schematic of the network showing the number of motes contained within each transmitter FOV.

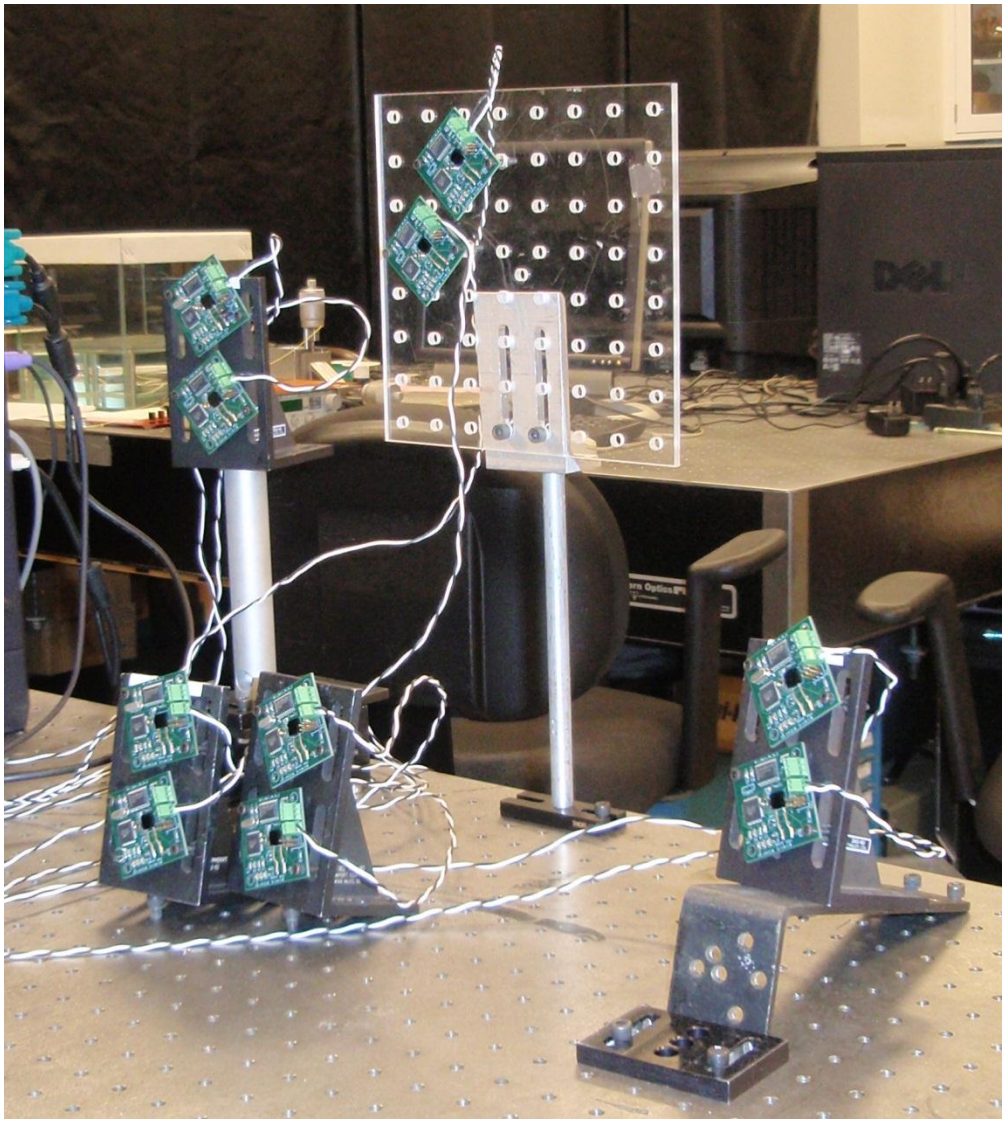


Figure 5.1 - Picture of 10 motes. The view is slightly angled.

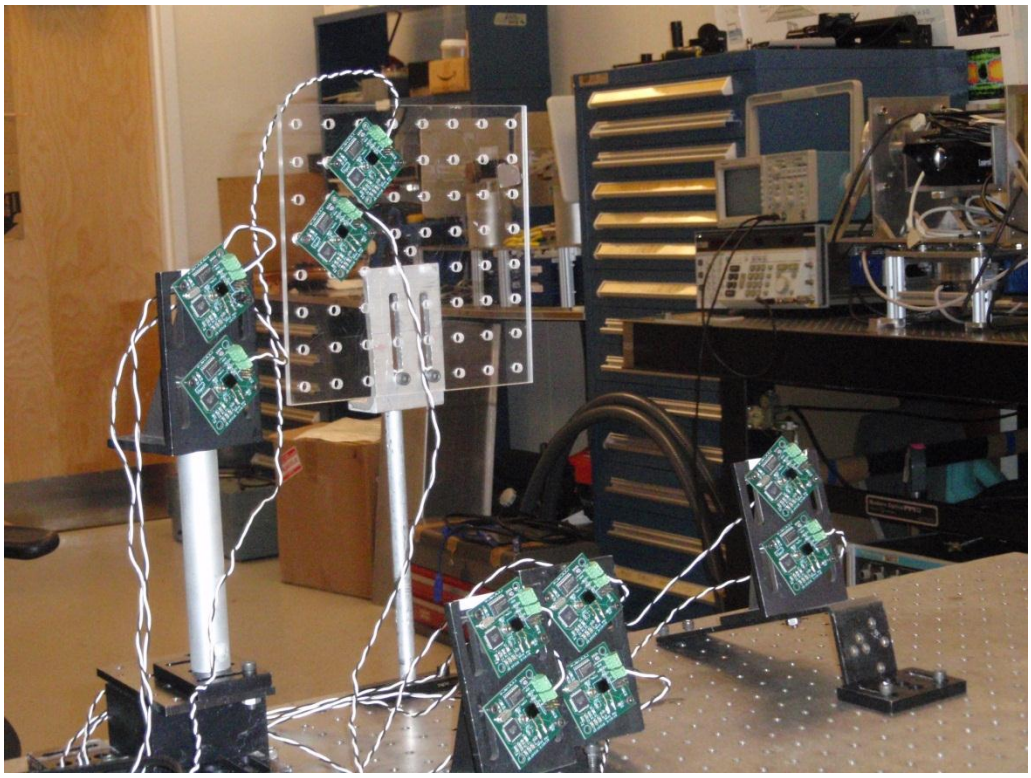


Figure 5.2 - Picture of 10 motes.

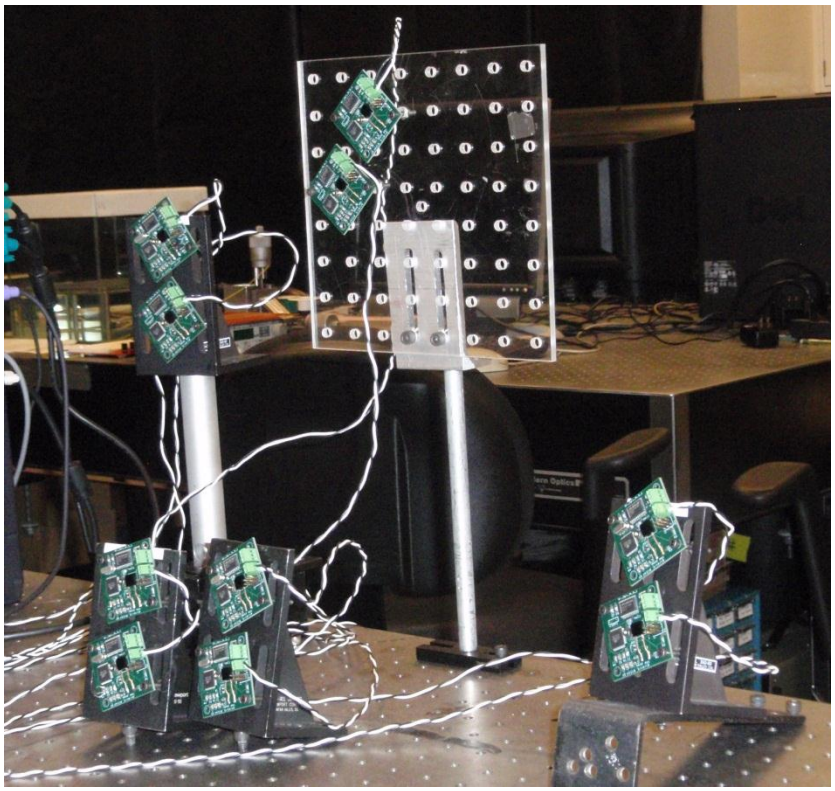


Figure 5.3 - Picture of 10 motes.

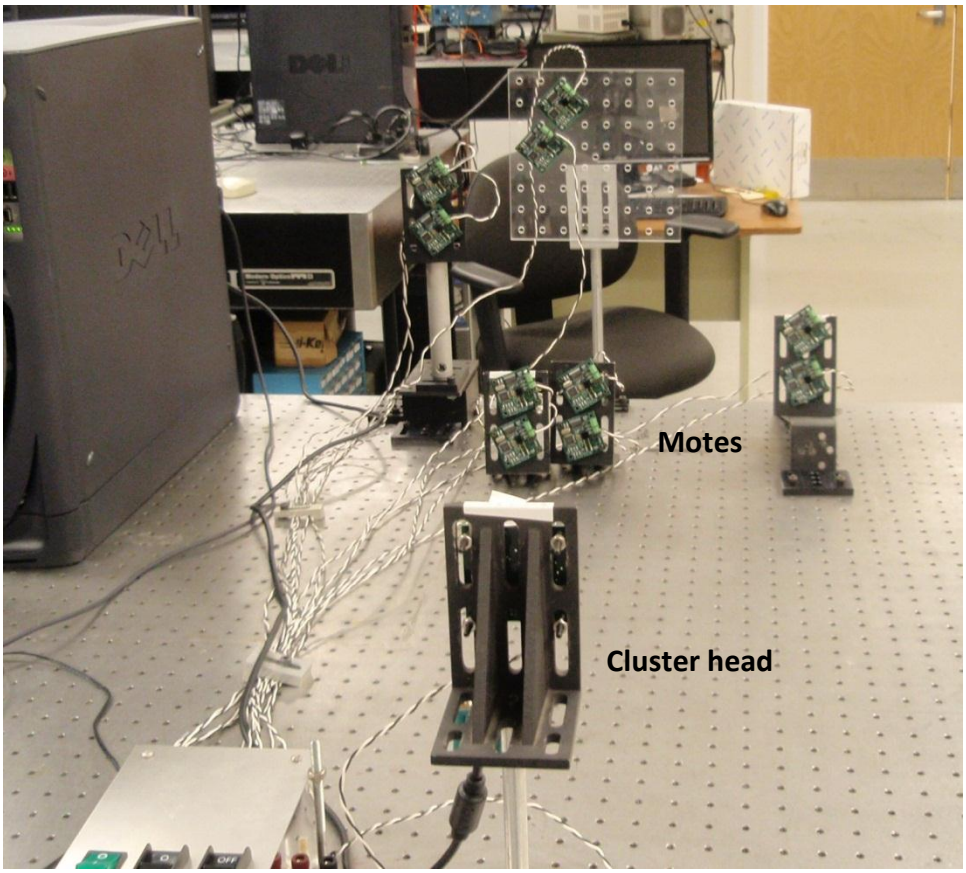


Figure 5.4 - Picture of 10 motes facing the cluster head.

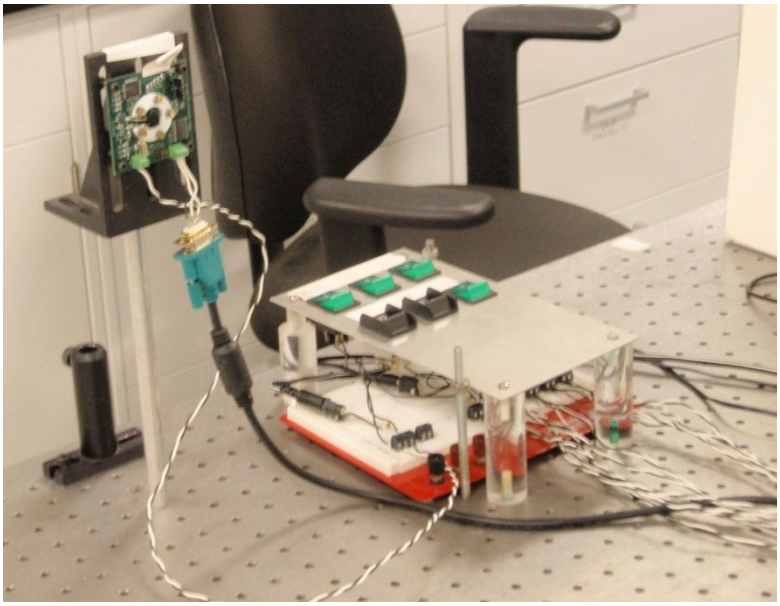


Figure 5.5 - Picture of cluster head. (Power supply switches are to the right).

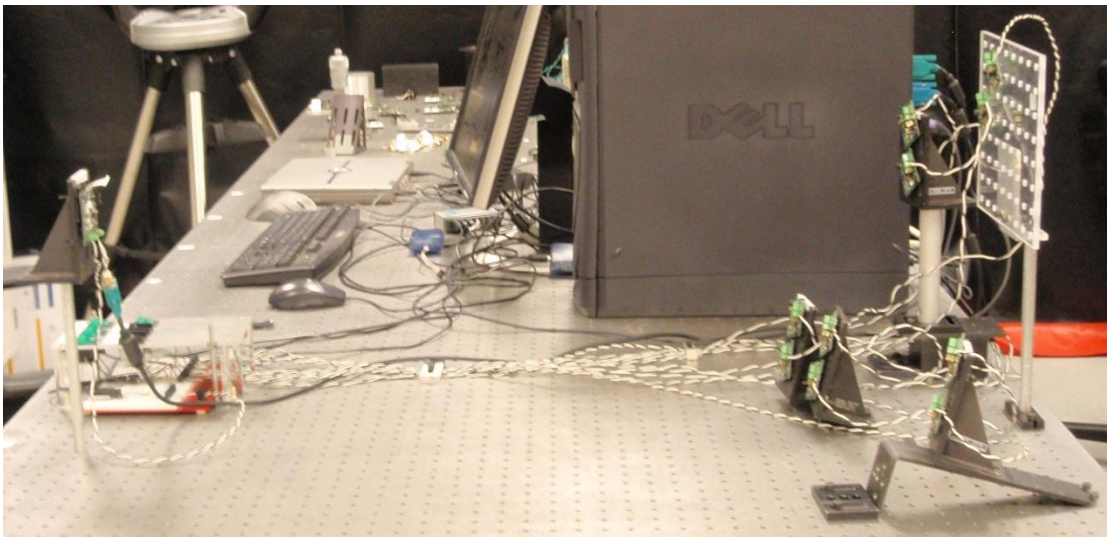


Figure 5.6 - Picture of network cluster. Cluster head (left) and 10 motes (right).

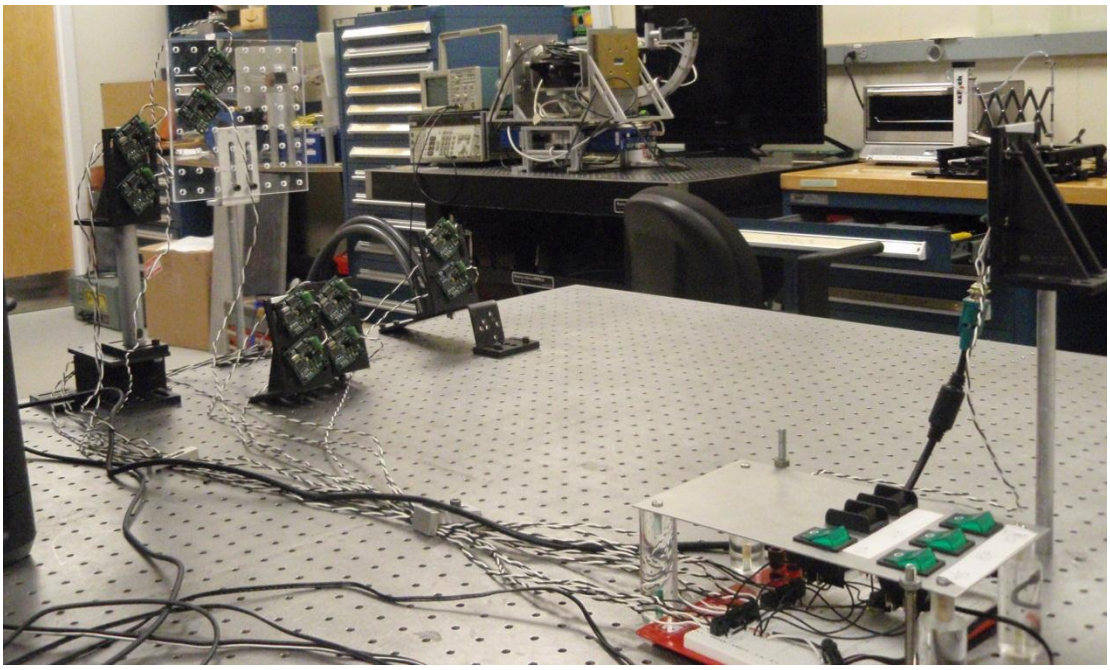


Figure 5.7 - Picture of network cluster. Cluster head (right) and 10 motes (left).

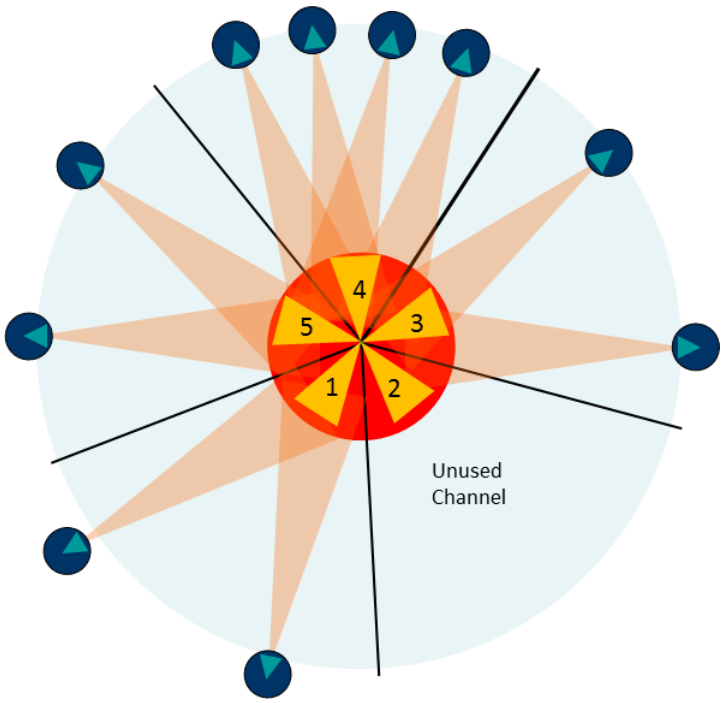


Figure 5.8 - Schematic showing the number of motes within the each of the cluster head's transmitters FOVs.

5.2.1 - Network Protocols and Hardware Verification

An initial experiment was conducted to test the developed network protocols, software controls, and system hardware. The focus was to ensure that the SR-FSOC system was detecting packet collisions and that the MAC algorithm was successfully assigning time slots to all motes contending for channel access.

It was shown (in chapter 3) that the optimal value for *p* depends on the number of motes contending for time slots. In this experiment, the cluster head was aware that initially 10 motes would be trying to acquire time slots. Therefore, the cluster head was pre-programmed to initially transmit a *p* value of $\frac{1}{10}$, and then to adjust

the p value to $\frac{1}{9}, \frac{1}{8}, \frac{1}{7} \ldots \frac{1}{1}$ after it successfully assigned a time slot to a contending mote. Thus, the value of p within each SF reflected the number of motes trying to acquire time slots.

The experiment begins with the cluster head transmitting a wake up signal across all of its transmitters to the motes. Upon detecting the signal, the motes wake up and try to access the channel to the cluster head. The simultaneous transmissions cause collisions at the cluster head, triggering the RA protocol to run and handle the channel contention. The experiment was run 5000 times. Figure 5.9 shows a plot of the average number of SFs required to connect each mote. The averaged experimental results, (shown in red) matched the calculated $<T_{SF}>$ values, which were derived in chapter 3.5.2. This shows that the implemented system hardware and network protocols were operating as intended. The absolute best results, which were discussed in chapter 3.5.3, are shown in blue to provide a relative comparison between expected results, experimental results, and the best possible results. The exact values of the experimental results are shown in figure 5.10.

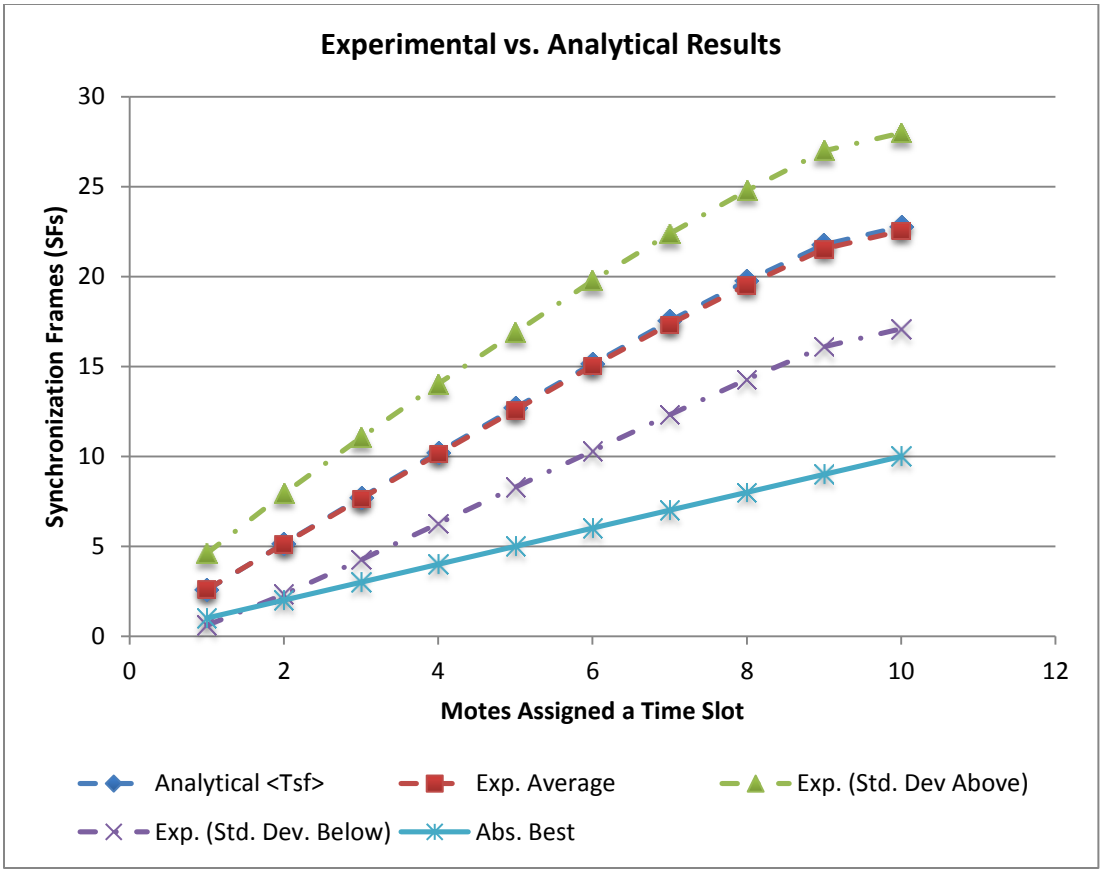


Figure 5.9 - Plot of the experimental and analytical results of the average number of SFs required to connect to each mote. The analytical results were derived in chapter 3.5.2. (Exp: Experimental)

Mote#	Calculated SFs (Theoretical)	Measured SFs (Experimental)	Std. Deviation (Experimental)
1	2.58	2.60	2.0
2	5.15	5.13	2.80
3	7.69	7.65	3.40
4	10.22	10.14	3.89
5	12.70	12.58	4.30
6	15.14	15.03	4.74
7	17.52	17.34	5.03
8	19.77	19.53	5.25
9	21.77	21.54	5.45
10	22.77	22.54	5.45

Figure 5.10 - Table of exact values of the experimental and analytical results.

Figure 5.11 shows the statistical distribution of the experimentally measured $<T_{SF}>$ required to assign time slots to all motes over the 5000 experiment runs. The distribution is slightly skewed to the right. The absolute best performance occurred three times in the 5000 runs, which equated to a probability of .0006. From figure 3.14, the $<P_{10}^{Best}>$ for 10 contending motes is 0.0036288, which corresponds to the event happening approximately once in 2756 trials. The slight deviation in probabilities can be attributed to the microcontroller's inability in generating precise probabilities, as discussed in chapter 4.

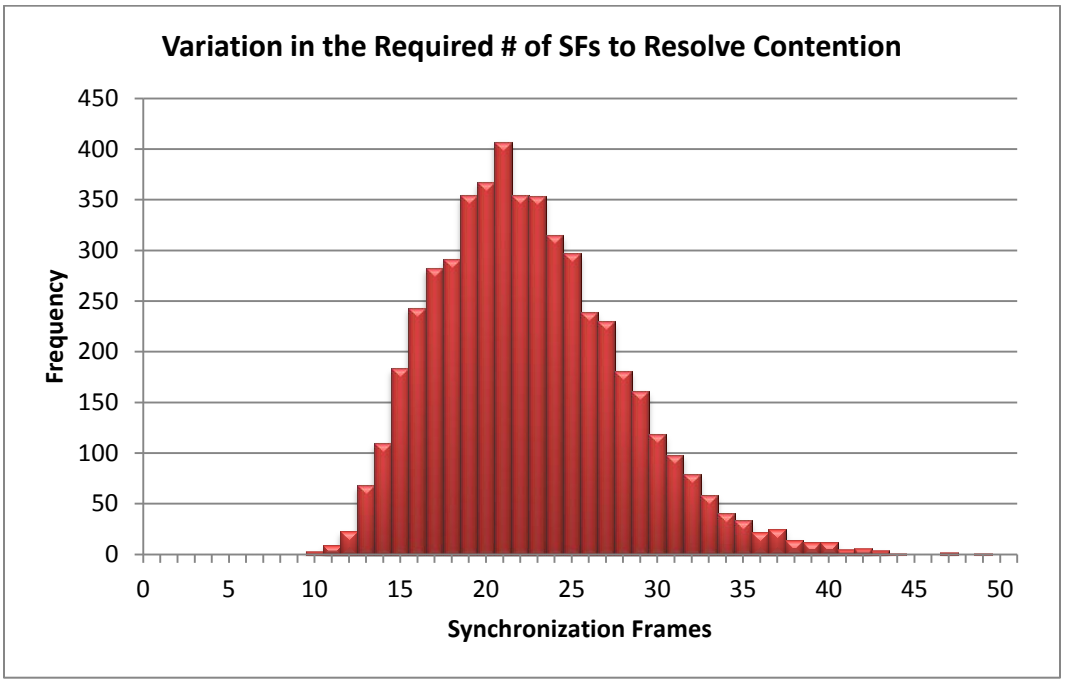


Figure 5.11 - Frequency plot showing the variation in how many total synchronization frames are required to resolve channel contention of 10 motes. The values come from 5000 experimental runs where 10 motes try to connect to the cluster head simultaneously.

5.3. – Implemented Random Access Protocol Algorithm

In the previous experiment, the cluster head was transmitting the optimal probability (p_{opt}) during each RAS signal. In order for that to be possible, the cluster head would need to know the exact number of motes trying to access the channel. However, in a practical network this information will not be available to the cluster head because different numbers of motes will try to access the channel at different times based on the sensor usage. Instead, the cluster head will need to estimate a proper p value to transmit (p_{trans}) each RAS signal. Depending on the difference $|p_{trans} - p_{opt}|$, the number of SFs required to handle channel contention will deviate from the expected $<T_{SF}>$ values. Therefore, the cluster head needs an efficient algorithm that can minimize the difference, $|p_{trans} - p_{opt}|$, so that the total number of SFs required will be close to the calculated $<T_{SF}>$ value.

The objective of the algorithm is to resolve channel contention in the fewest number of SFs regardless of the network load. To this end, an algorithm was developed to dynamically adjust the value of p based on the number of collisions (C), successful (S) transmissions, and timeouts (T) detected at the cluster head. Initially, the RA protocol selects a p_o value. After detecting S, T, and C, the algorithm begins to shift the p_{trans} value accordingly. The objective is to eventually have the cluster head transmit p_{trans}=1 and detect a timeout so that the cluster head can assume that all motes have been assigned time slots and that the network communication should resume via D-TDMA time slots. It will be shown that this approach leads to a simple

MCU implementation that does not require advanced mathematical operations, which drastically reduces the number of clock cycles for a low-power MCU.

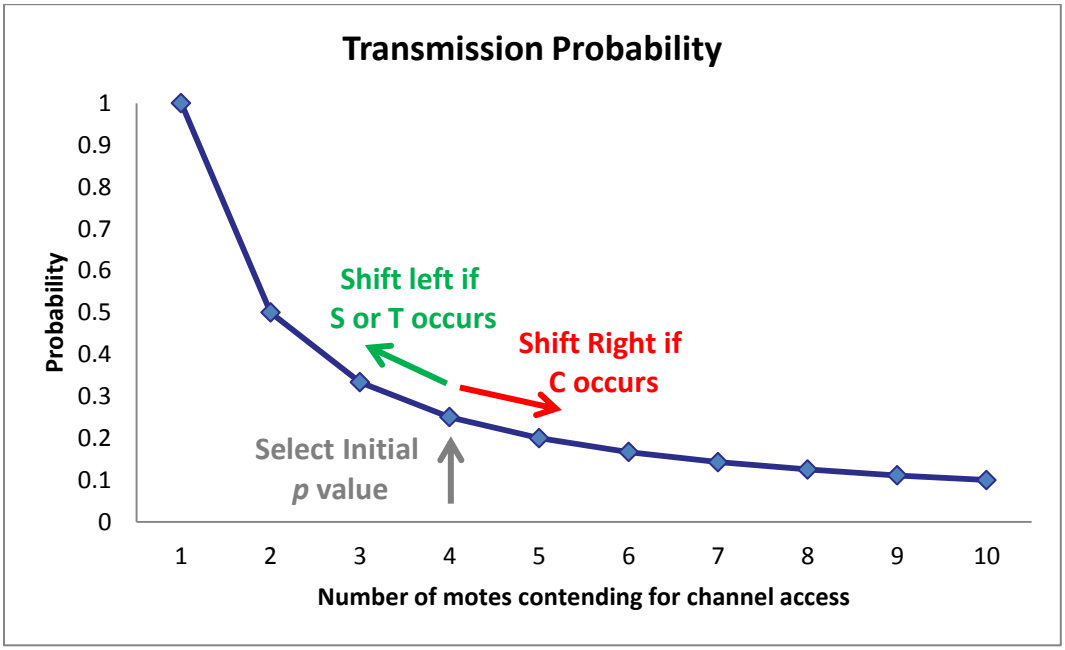


Figure 5.12 - Illustration of how the transmission probability can be adjusted.

5.3.1 – Algorithm Parameters

The developed algorithm has several parameters that need to be initialized. Different initialization values will alter the performance of the algorithm, so a method of optimizing the parameters will be detailed in the following sections. These parameters are the window size, quantity factor, and initial p value (p_o).

Window Size: This value determines how many SFs are required before the cluster head can potentially adjust the current value of p to a new value of p, based on the number of packet collisions, successful transmissions, and timeouts. Increasing the

window (W) size leads to more accurate estimations because more data is aggregated, but in doing so can also increase the overall number of SFs.

Quantity Factor: The quantity factor $\hat{Q}$ is a vector containing the number of timeouts (Q_T), successful transmissions (Q_S), and packet collisions (Q_C) that have occurred within the specified W, $\hat{Q} = (Q_S \ Q_T \ Q_C)$. When the required SFs is equal to W, $\hat{Q}$ is checked to see if any of its 3 elements is greater than the other 2. If that is true, then cluster head adjusts the value of *p* by an adjustment value

$$M = \pm\left|max(\hat{Q})\right| \tag{5.1}$$

where the sign depends on whether the max is Q_S, Q_T, or Q_C. For S and T, the sign of M is negative and for C it is positive.

The adjusted value of *p* is calculated in the following manner. Based on the initial value of *p'*, the initial estimated number of motes (N') is generated, $N' = \frac{1}{p'}$ After a certain number of synchronization frames, as required by the window size, a new estimated number of motes (N) is generated, $N = N' + M$, where the value of M is determined by the algorithm. This value then generates the new value $p = \frac{1}{N}$ with the new value of *p* becoming the adjusted value of *p'* in the next iteration. This process repeats itself until the algorithm determines that there are no more motes contending for channel access.

If there is no maximum value in $\hat{Q}$ within the specified W, then additional SFs are added to W until a maximum value in $\hat{Q}$ occurs. In this situation,

$$M = \pm\left|\max\left(\hat{Q}\right) - \gamma\right| \quad (5.2)$$

where γ is the number of SFs added to the W size in order to find a maximum.

Based on the size of the window, the adjustment value can become very large if a string of S, T, or C occurs.

The range of possible *p* values follows the sequence $\frac{1}{N}, \frac{1}{N-1}, \frac{1}{N-2} \ldots \frac{1}{1}$, where *N* is the total number of motes in the network. If the M factor adjusts *p* to a value below or above its limits, *p* is set back to the appropriate limit. To save computational time for the MCU, all possible *p* values are generated at initial power-on and stored in SRAM.

Initial *p* value: The algorithm can be tuned to start with a different value of p_o. This is the value that the algorithm uses before making any sort of adjustment. The value can range from 1 to $\frac{1}{10}$. For network configurations with many motes contending for channel access, a smaller p_o value would potentially provide a shorter RA-time than a larger p_o value because the smaller value would lessen the chances of collision. Similarly, a larger p_o value is better for situations where only a few motes are contending for channel access because the larger p_o value could decrease the chance of detecting timeouts. But as stated before, since the exact channel contention will not be known *a priori*, the cluster head will not always be able to set

a proper p_o value. Instead, a p_o value that can resolve channel contention the fastest regardless of the network load needs to be set.

5.3.2 – Implemented Algorithms

Numerous simulations were conducted to test the algorithm with different W sizes and different p_o values on a network load with 10 motes contending for channel access. Figure 5.13 shows the simulated $<T_{SF}>$ for the different W sizes for 4 different p_o parameters, p_o = 1.0, p_o = 0.5, p_o = 0.2, and p_o = 0.1. The simulations show that the number of SFs required to resolve contention increases monotonically for increasing *W* after *W* = 3. When $W \leq 3$, the RA-protocol finishes in a shorter number of SFs as compared to larger W sizes for each of the tested p_o values. Therefore, it was determined that using a W size less than 3 for a 10 mote configuration would provide the shortest RA-times, so algorithms with W sizes of 1, 2, and 3 were coded "C" and imbedded within the MCU's flash memory. The implemented algorithms are shown in figure 5.14.

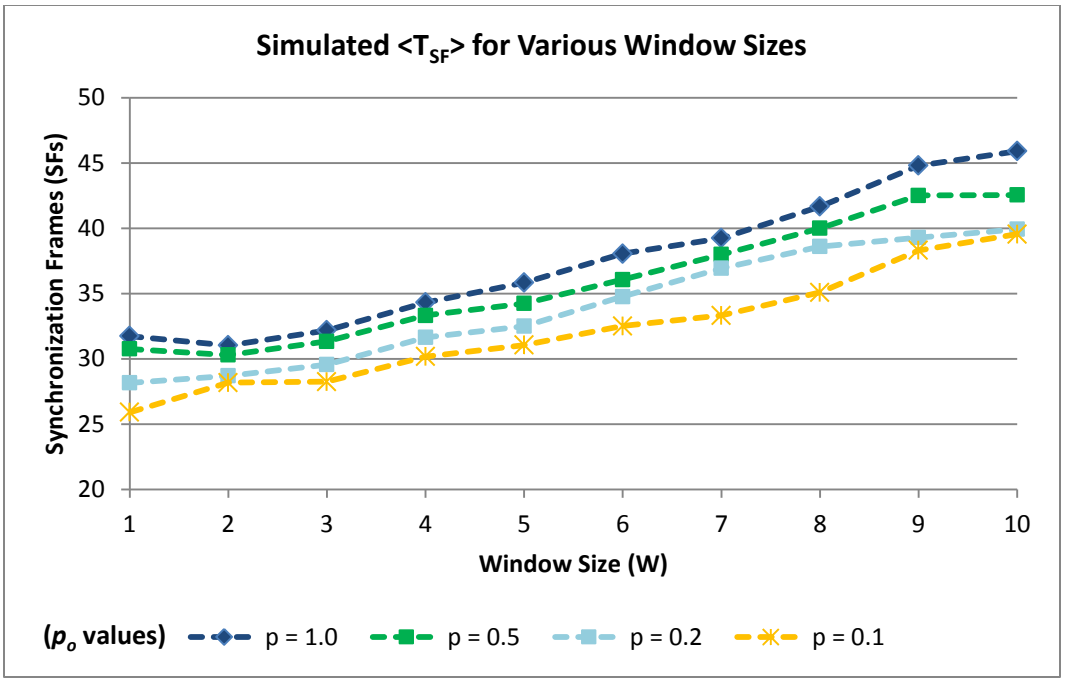


Figure 5.13 - The simulated $<T_{SF}>$ for the different W sizes for 4 different p_o parameters.

Algorithm: W = 1
(A1)

Logic	Output
if $Q_C = 1$	M = +1
if $Q_S = 1$	M = -1
if $Q_T = 1$	M = -1

Algorithm: W = 2
(A2)

Logic	Output
if $Q_C = 2$	M = +2
if $Q_S = 2$	M = -2
if $Q_T = 2$	M = -2

(Extra SFs)

Logic	Output
if $Q_C = 2$	M = +1
if $Q_S = 2$	M = -1
if $Q_T = 2$	M = -1

Algorithm: W = 3
(A3)

Logic	Output
if $Q_C = 3$	M = +3
if $Q_S = 3$	M = -3
if $Q_T = 3$	M = -3
if $Q_C = 2$	M = +2
if $Q_S = 2$	M = -2
if $Q_T = 2$	M = -2

(Extra SF)

Logic	Output
if $Q_C = 2$	M = +1
if $Q_S = 2$	M = -1
if $Q_T = 2$	M = -1

Figure 5.14 - Specifics of each implemented algorithm. Q is the quantity factor. The subscript denotes collisions (C), successes (S), and timeouts (T). M is the modification factor.

5.4 – Random Access Protocol Experiments

Each algorithm was experimentally tested in 2000 runs. Each run consisted of:

1. Cluster head triggering all motes to wake up and to access the channel
2. Cluster head resolving channel contention and assigning time slots

The other input parameter to the algorithm is the p_o value. Figure 5.15 shows how the MCU does not precisely generate probabilities smaller than $\frac{1}{5}$. Taking this into account, each algorithm was tested using p_o values of $\frac{1}{10}$, $\frac{1}{5}$, $\frac{1}{2}$, and 1. These values were selected because they are either separated in value by a distinct amount or they are separated by an M value of 2 or more. In this manner, the RA-protocol could be tested from different initial points and any visible effect on the experimental results could be clearly attributed to that particular p_o value.

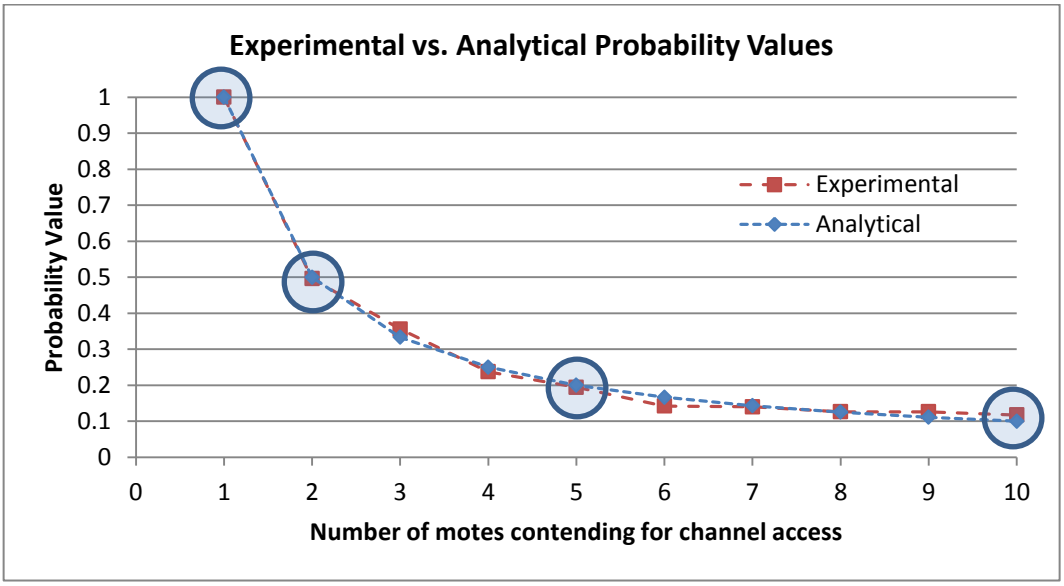


Figure 5.15 - Experimental vs. analytical probability values. The MCU probabilities were averaged over 1000 samples. The circled points indicate which probabilities were experimentally tested as initial transmission probabilities.

All random access protocol measurements were made on the cluster head, and then transmitted to a PC via a serial cable. These measurements consisted of:

- Number of SFs required to connect all 10 motes
- Number of packet collisions
- Number of timeouts
- Number of packets successful received

Each algorithm was tested 4 times (4 different p_o values), on three different network loads of 10 motes, 6 motes, and 2 motes (all contending for channel access simultaneously) for a total of 12 tests per algorithm. These three different network loads were chosen to represent different types of "network burst rates" (NWBR). The NWBR value corresponds to the number of motes that wake up and access the channel simultaneously. For instance, $NWBR_{10}$ corresponds to a network load where 10 motes are trying to access the channel simultaneously. Similarly, $NWBR_6$ and $NWBR_2$ correspond to network loads where 6 motes and 2 motes try to access the channel simultaneously, respectively. These NWBR values set the "collision congestion" of the network. Higher values of NWBR generate more collisions and longer random access times than lower values. In each of these configurations, a total of 10 motes are present within the network, so the possible p_{trans} values are $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \ldots \frac{1}{10}$. The idea is that each algorithm will perform differently on its 12 tests based on the NWBR and p_o values, so an optimal algorithm will have to be selected based on minimizing both the random access times and energy consumption of the

RA-protocol for all potential loads and NWBR values. Figure 5.16 shows the exact placement of the different motes in regards to the cluster head's 5 transmitters for the different NWBR experiments, while figures 5.1-5.7 show the actual experimental setup. The experimental results are shown first and then the mathematical analysis for selecting an optimal algorithm is detailed.

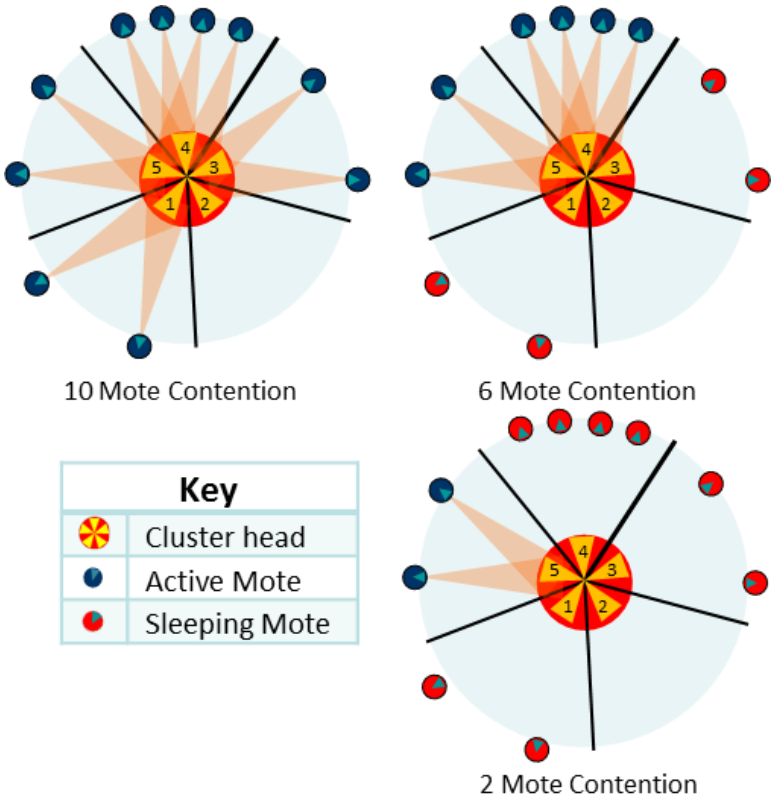


Figure 5.16 - Schematics of three different network configuration experimental tests. The schematic shows the number of active and sleeping motes within the FOV of the cluster head's transmitters.

5.4.1 – Random Access Protocol Results

The results consist of the total number of synchronization frames required to resolve network contention and the total number of collisions and timeouts that occurred during each RA protocol run.

5.4.1.1 – Total Number of Synchronization Frames

Short random access times are important so that the network can spend most of its energy and time transmitting sensor data to the end user/destination rather than transmit MAC communication commands between motes. Therefore, a minimal number of SFs to resolve channel contention is an important aspect of the protocol.

Figures 5.17 – 5.20 and figures 5.21-5.24 show the average number of SFs required to assign each mote a time slot for 10-mote and 6-mote contention, respectively. On the x-axis, the "finish" label corresponds to when the RA protocol terminates, which happens when the cluster head transmits an RAS signal with p_{trans} = 1 and detects a "timeout". This adds 1 extra SF to each algorithm after all motes are assigned time slots. The calculated $<T_{SF}>$ values are plotted along with the different algorithm results so that comparisons in performance could be made. As discussed in chapter 3, the $<T_{SF}>$ values are the expected total number of SFs required to resolve channel contention if p_{opt} is transmitted every RAS signal. The "Abs. Best" line plots the performance of the event where a single mote replies in each RAS

frame, which leads to the shortest number of SFs possible. As mentioned in chapter 3, this is an unlikely event, but it is the best possible outcome.

For both network loads, the algorithms approached the calculated $<T_{SF}>$ values as the initial value of *p* was set closer to the ideal initial value for the network, which is $p = \frac{1}{N}$, where *N* equals 10 and 6 for the 2 different networks.

For 10-mote contention, each algorithm performed relatively similarly when they were initialized with the same p_o value. The fastest RA-times were generated with $p_o = 0.1$ and $p_o = 0.2$. At $p_o = 0.1$, each algorithm produced nearly identical results with the $<T_{SF}>$ values, which shows that the algorithm is capable of producing near optimal results.

For 6-mote contention, there was a bit more variance in each algorithm's performance. Overall, algorithm A1 produced the shortest RA-times when initialized to $p_o = 0.1$ and $p_o = 0.2$.

For 2-mote contention, there was also some variance among each algorith's performance. The fastest RA-times were measured when the algorithms were initialized with $p_o = 1.0$ and $p_o = 0.5$, which is expected since there are a few number of motes accessing the chnanel. Once again, algorithm A1 provided the fastest RA-times.

10-Mote Contention RA-Time Plots

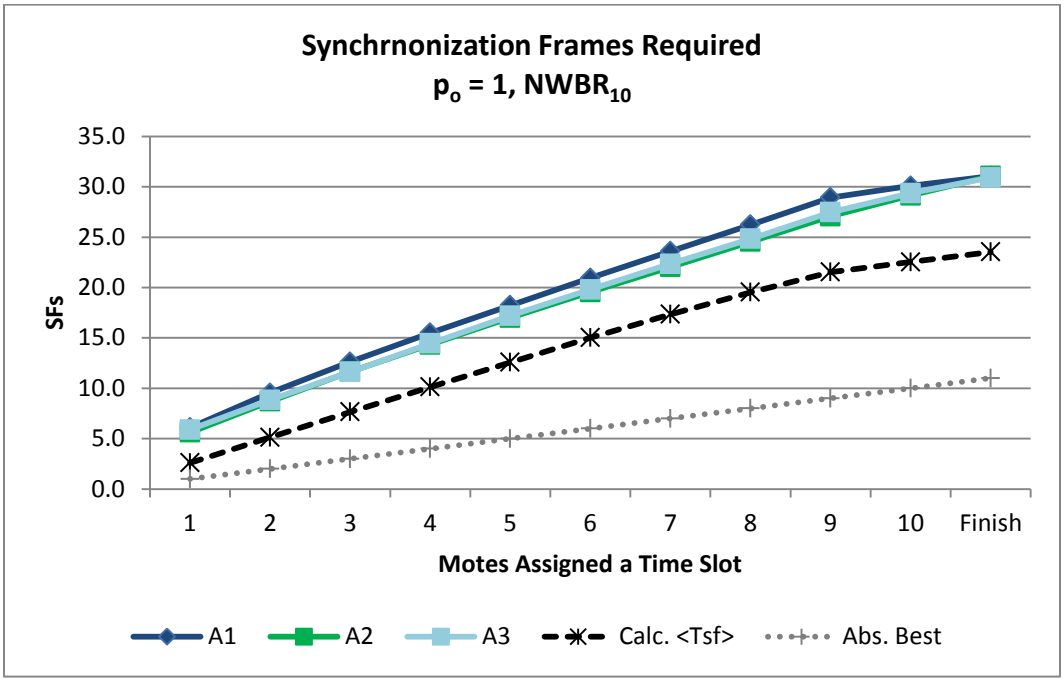


Figure 5.17 - Experimental Results: 10-Mote Contention, p_o=1.0, NWBR = 10

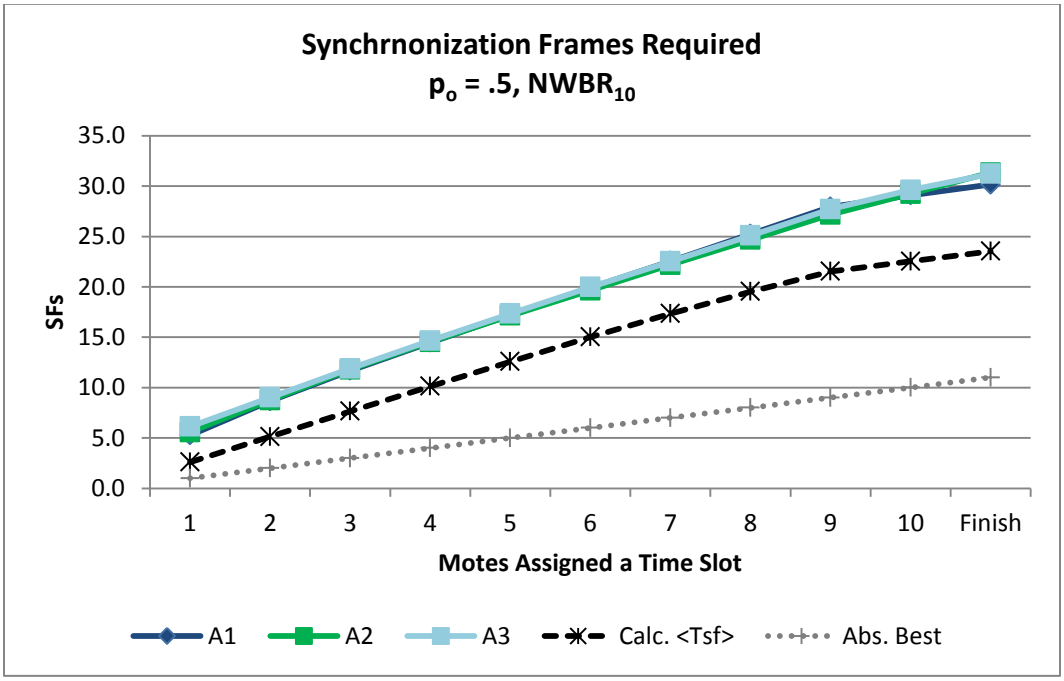


Figure 5.18 - Experimental Results: 10-Mote Contention, p_o=0.5, NWBR = 10

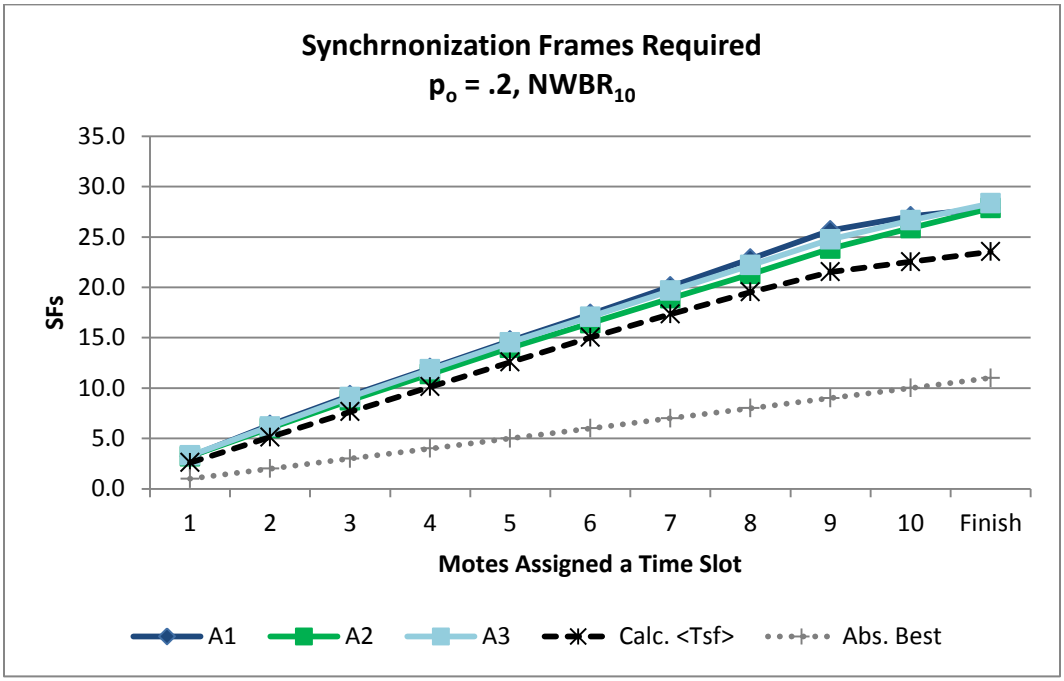


Figure 5.19 - Experimental Results: 10-Mote Contention, p_o=0.2, NWBR = 10

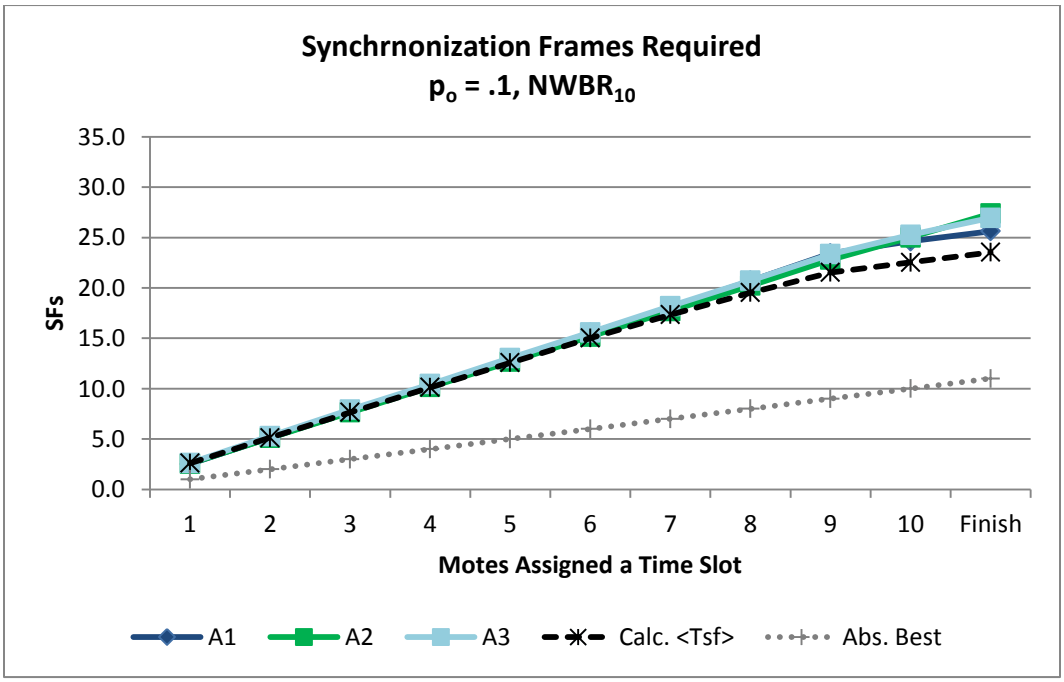


Figure 5.20 - Experimental Results: 10-Mote Contention, p_o=0.1, NWBR = 10

6-Mote Contention RA-Time Plots

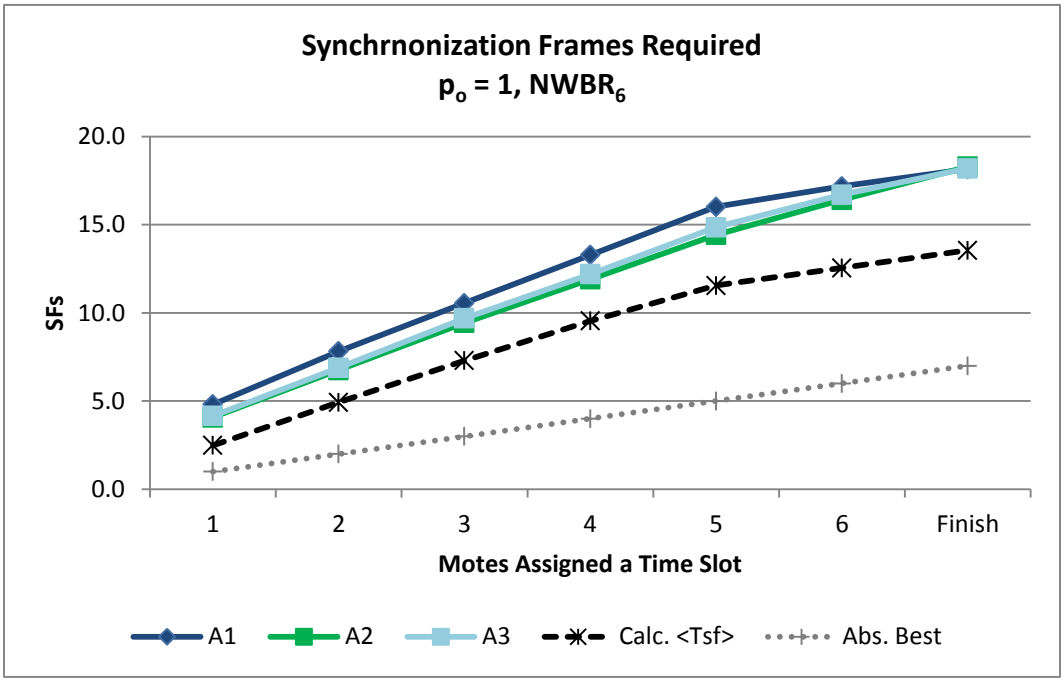


Figure 5.21 - Experimental Results: 6-Mote Contention, p_o=1.0, NWBR = 6

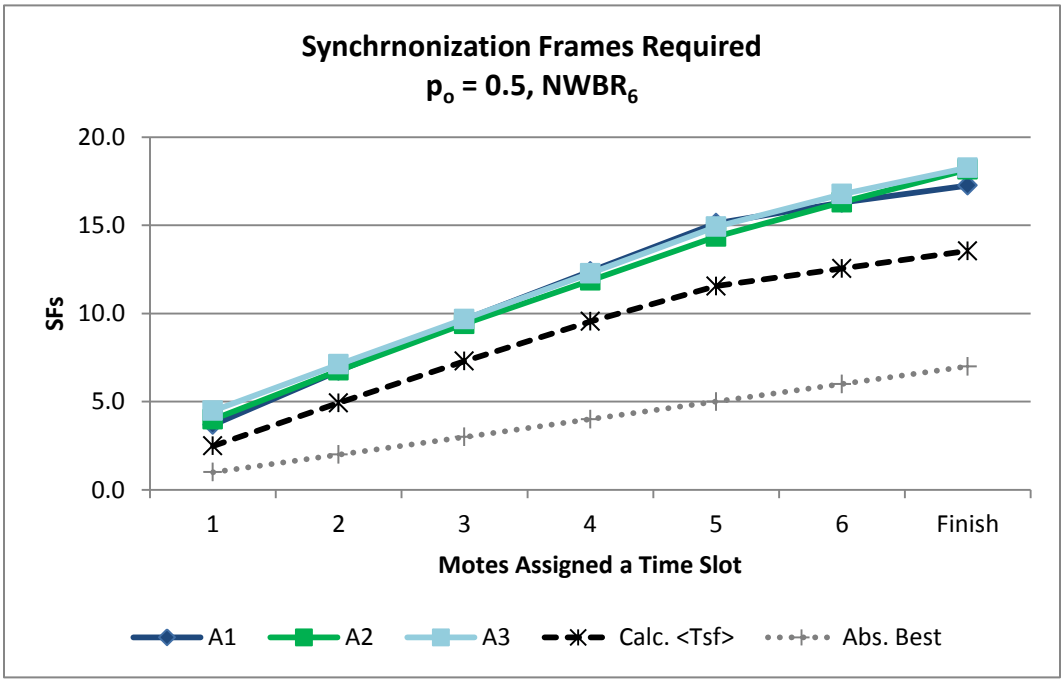


Figure 5.22 - Experimental Results: 6-Mote Contention, p_o=0.5, NWBR = 6

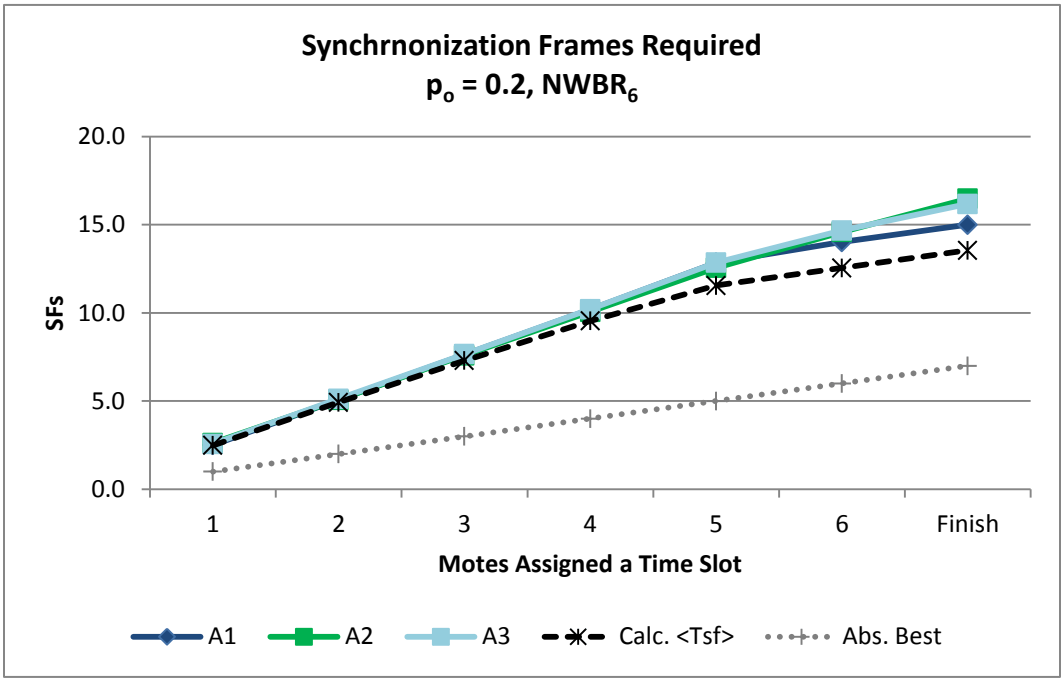


Figure 5.23 - Experimental Results: 6-Mote Contention, p_o=0.2, NWBR = 6

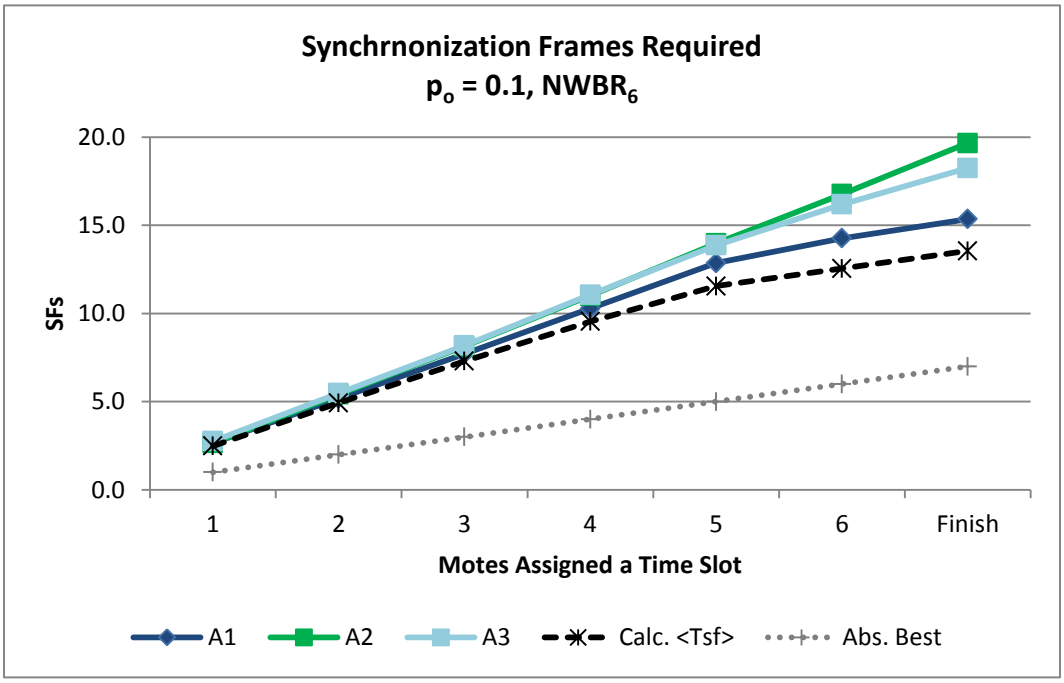


Figure 5.24 - Experimental Results: 6-Mote Contention, p_o=0.1, NWBR = 6

Figure 5.25 provides a summary of the results. The number of SFs required for each algorithm and p_o value are shown against the expected $<T_{SF}>$ and Abs. Best values. Essentially, p_o values of 1.0 and 0.5 are best for 2-mote contention but are the worst for 6-mote and 10-mote contention. Instead, p_o values of 0.2 and 0.1 are best for 10-mote and 6-mote contention, but are the worst for 2-mote contention. The Abs. best values serve as an indicator of the absolute minimum number of SFs required. For $NWBR_{10}$ and $NWBR_6$ there was not a large variance in the total number of SFs between the different tested algorithms and p_o values. Since the total number of SFs is equal to all the timeouts, successes, and collisions during an RA protocol run, and the total number of successes for each tested NWBR value is equal for each algorithm, then it can be assumed that some of the tested algorithms and p_o values caused more collisions while others caused more timeouts. This turns out to be the case, as will be shown in the next section.

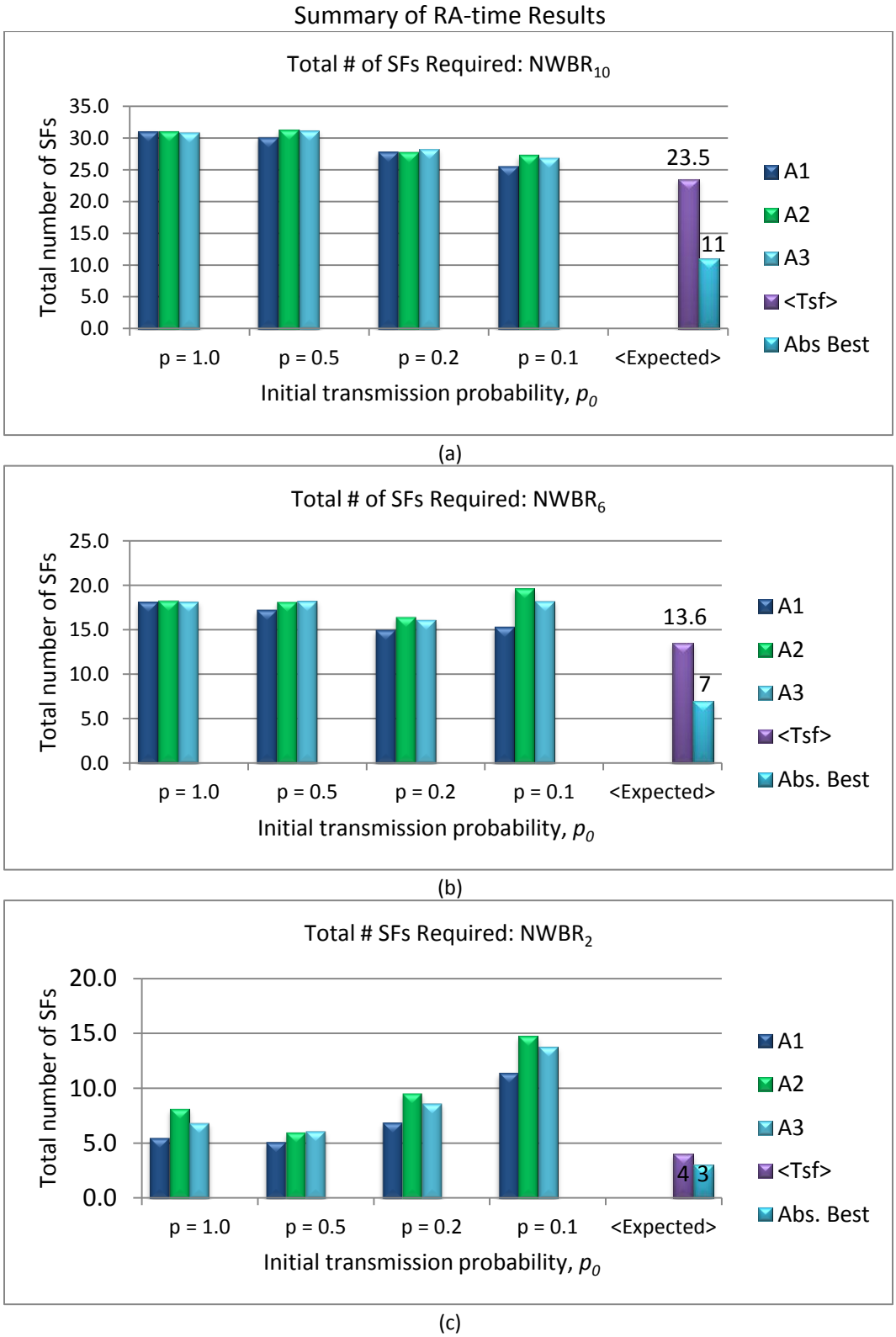


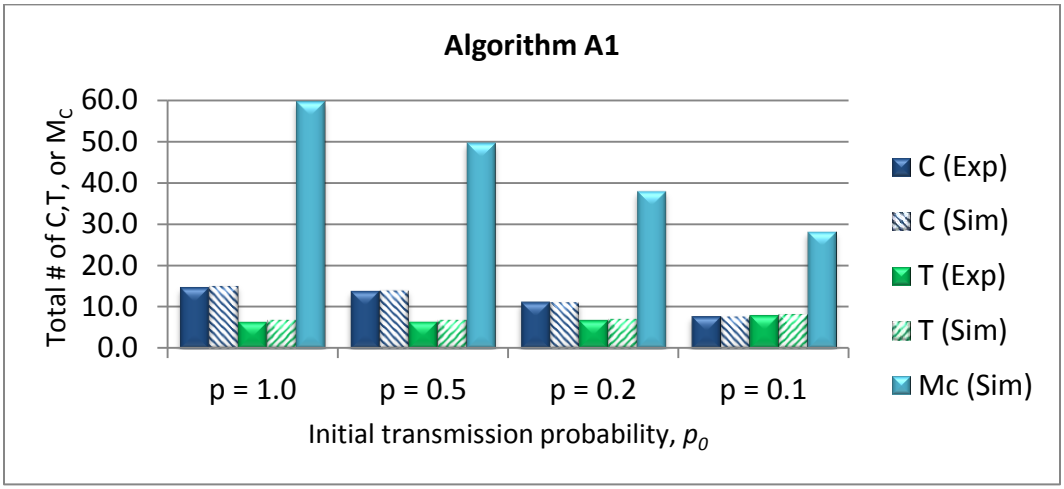
Figure 5.25 - Summary of the Different Algorithm RA-time Results. The *p* values shown on the x-axis are all initial *p* values. The label "Expected" refers to the analytical $<T_{SF}>$ and Abs. Best calculations.

5.4.1.2 – Total Number of Collisions and Timeouts

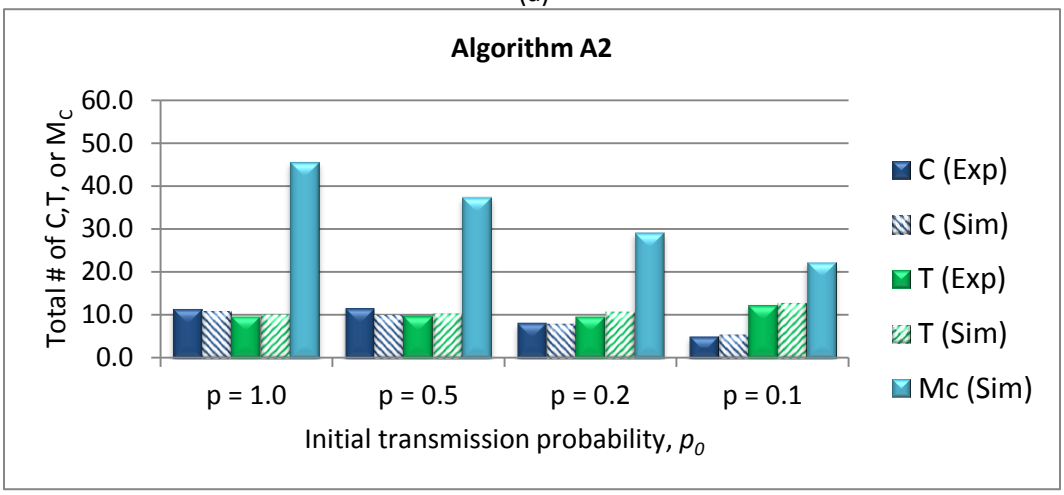
During the RA-protocol, energy is wasted whenever a collision or timeout is detected at the cluster head. For instance, if the cluster head transmits an RAS signal across its entire FOV during a SF and detects either a T or C, then the energy the cluster head used to transmit the signals was wasted since no mote was assigned a time slot. When a T occurs, only the cluster head wastes energy, but when a when a C is detected, then both the cluster head and the motes that transmitted a reply wasted their energy. M_C denotes the total number of transmissions from the motes that are involved with causing a collision at the cluster head.

Since all data was collected via the cluster head, T and C were measured, but M_C was not since it must be measured from the motes. Instead, M_c was recorded by simulating each of the algorithms. Based on the cluster head's p_{trans} value in each RAS signal, the exact number of simultaneous transmissions from the motes was simulated and recorded. Each algorithm that was experimentally tested was simulated in MATLAB using the same p_o and network load. The simulated results for T and C are shown next to the experimental results for T and C in figures 5.26-5.28. As can be seen from the plots, these results match very closely for 10-mote contention and 6-mote contention. For 2-mote contention, the simulated and experimental results do not match exactly in value, but they match in showing the relative difference between collisions and timeouts. Overall, these results provide validity for using the M_C simulated values in the ensuing analysis.

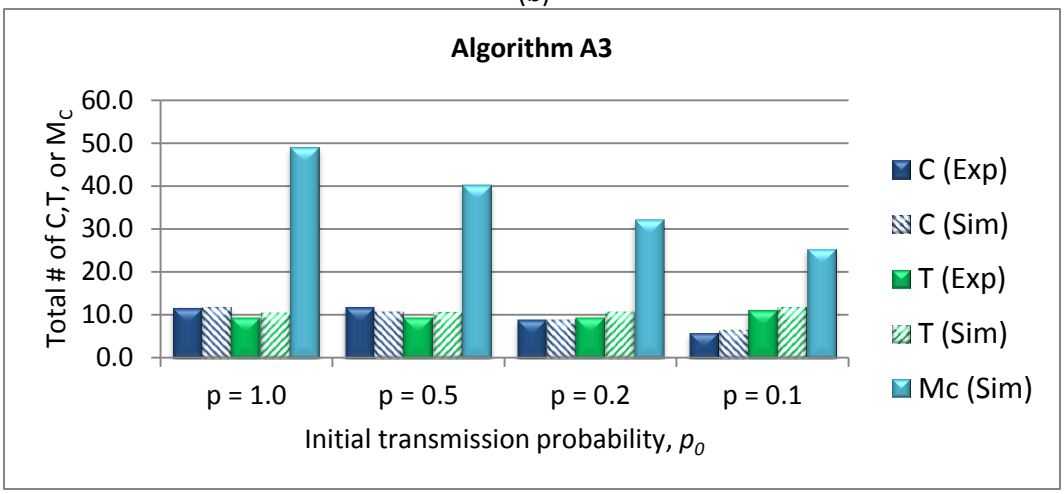
Experimental and Simulated Results of C, T, and M_C for $NWBR_{10}$



(a)



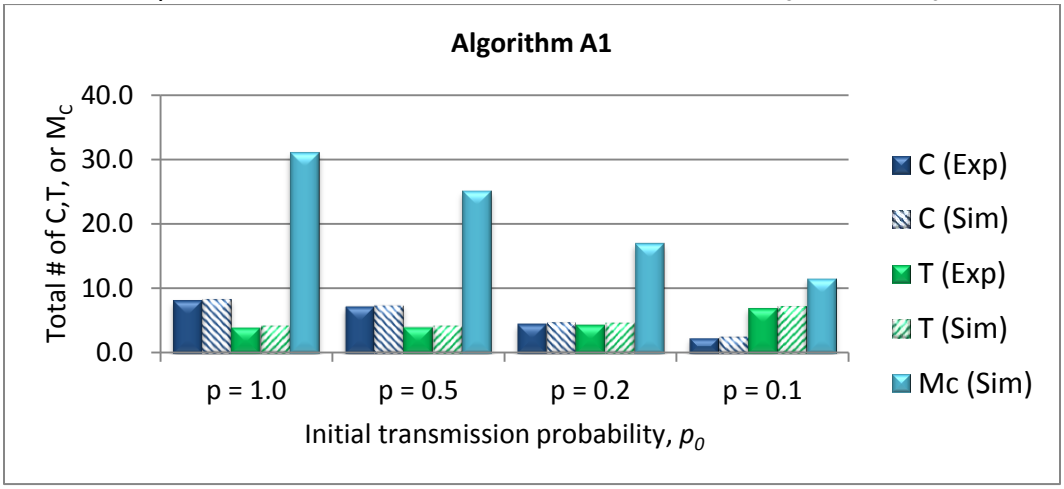
(b)



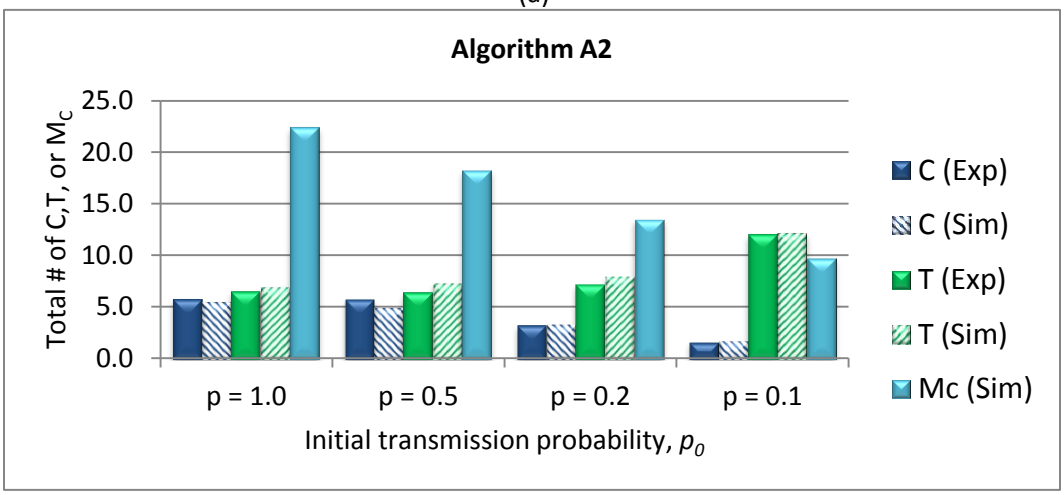
(c)

Figure 5.26 - Experimental and Simulated Results of C, T, and M_C for $NWBR_{10}$ Algorithm: a)A1; b)A2; c)A3

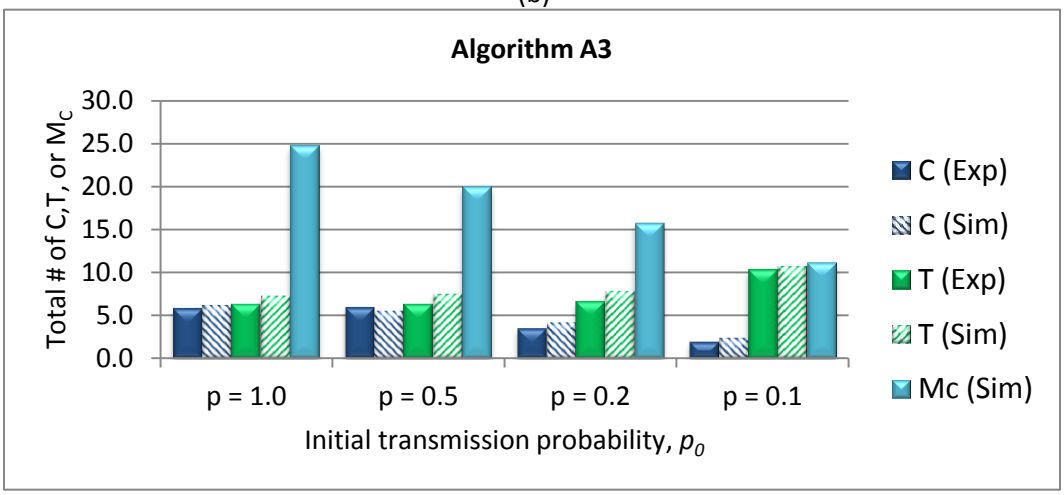
Experimental and Simulated Results of C, T, and M_C for $NWBR_6$



(a)



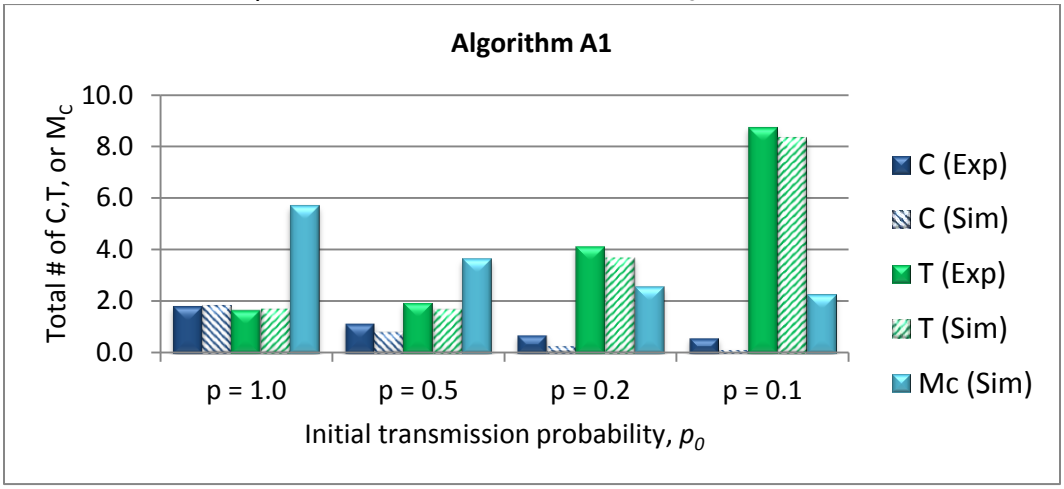
(b)



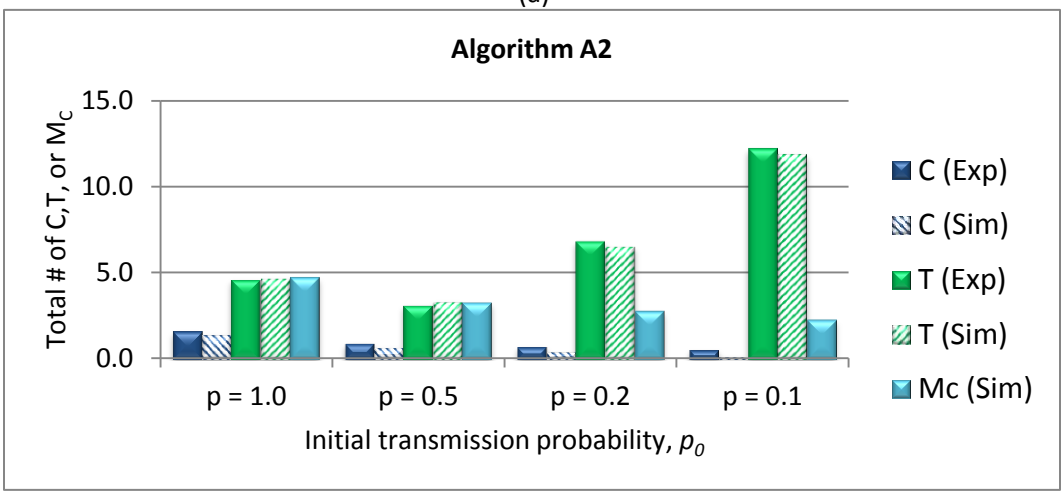
(c)

Figure 5.27 - Experimental and Simulated Results of C, T, and M_C for $NWBR_6$. Algorithm: a)A1; b)A2; c)A3

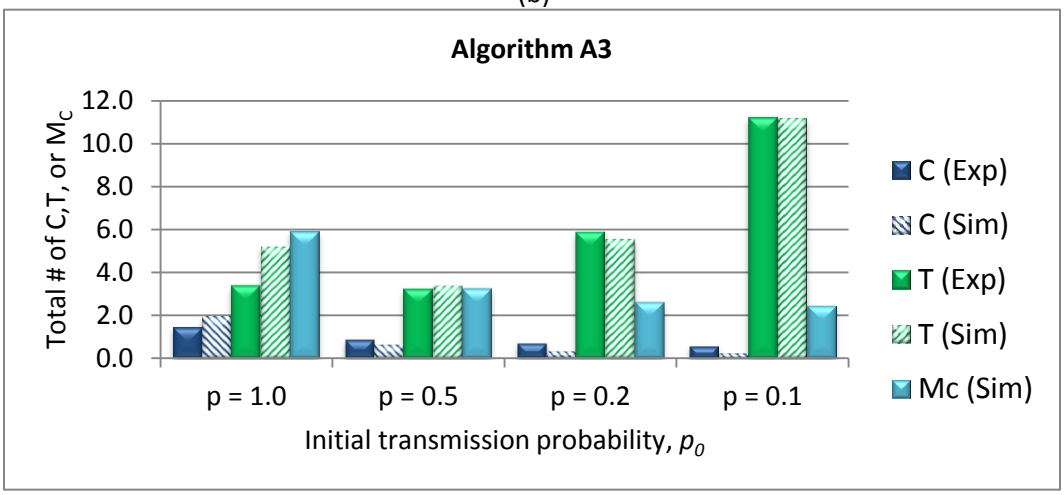
Experimental Results of C, T, and M_C for $NWBR_2$



(a)



(b)



(c)

Figure 5.28 - Experimental Results of C, T, and M_C for $NWBR_2$ Algorithm: a)A1; b)A2; c)A3

In each experiment, algorithms with larger p_o values caused the most collisions. The A_1 algorithm caused the most collisions in general. For $NWBR_{10}$ experiments, collisions were generally higher than timeouts, especially when $p_o = 1$. For $NWBR_2$ experiments, when $p_o = 0.1$, the timeouts were higher than collisions.

5.4.2 – Random Access Protocol Analysis and Optimal Settings

For each network type, the simulated $<T_{SF}>$, <C>, <T>, and $<M_C>$ values provide the expected optimal performance values. These results occur when the cluster head transmits p_{opt} in every SF. The values are shown in figure 5.29. The implemented algorithms are trying to achieve averaged results as close as possible to these optimal results.

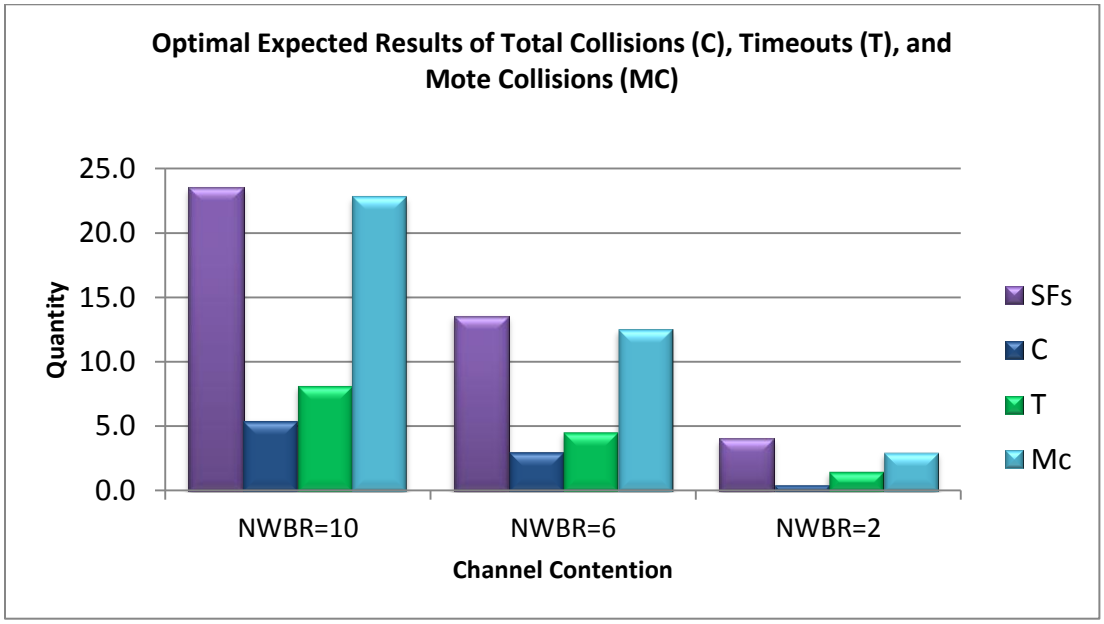


Figure 5.29 - Expected optimal performance results.

A cost function was developed to assess the energy lost to collisions and timeouts during the SFs. The algorithm that minimizes the energy lost provides an energy efficient algorithm for the 10 mote network cluster.

5.4.2.1. – Development of Random Access Protocol Performance Equation

The analysis begins with the construction of a cost function that aids in the minimization process. The cost function relates all the T, C, and M_C of the experimental results into energies and compares them with the expected optimal values of each network type.

The energy used to transmit a packet can be calculated by,

$$E_p = I * V * \frac{\langle \#1\ bits \rangle}{B} \tag{5.3}$$

where *I* is the current through the VCSEL, *V* is the voltage across the laser driver circuit, *<#1 bits>* is the average number of "1" bits within a packet, and *B* is the bitrate of the communication signal. For this system, *<#1 bits>* = 57, *V* = 5V, *I* = 9mA, and *B* is 100 kb/s. With these values, $E_p = 25.7\ \mu J$.

The average energy wasted at the cluster head depends on the algorithm and NWBR and can be calculated by,

$$E_X^{CH}(A_W(p_o)) = 5E_p(T + C), \tag{5.4}$$

where W is the window size of the algorithm, p_o is the initial p value, and X refers to the NWBR value (number of motes contending for channel access). The factor of 5 is present because the cluster head has 5 transmitters. Depending on p_o and the NWBR, each algorithm resolves channel contention in a different number of SFs, which will deviate from the optimal $<T_{SF}>$ values for the particular NWBR value. For each additional SF required over the optimal $<T_{SF}>$, the cluster head is wasting energy through a collision or a timeout.

The average energy wasted by all the motes can be calculated by,

$$E_X^M(A_W(p_o)) = E_p M_c \tag{5.5}$$

The average total wasted energy to T and C is then

$$E_X^T(A_W(p_o)) = E_X^{CH}(A_W(p_o)) + E_X^M(A_W(p_o)) \tag{5.6}$$

With equation 5.6, the energy loss functions for each NWBR value can be combined, and the general form of the RA-protocol total energy cost function, *H,* can be defined as

$$H\left(\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_N, A_W(p_o)\right) = \sqrt{\left(\sum_{X=1}^{N} \lambda_X (E_X^T - O_X^T)^2\right)} \tag{5.7}$$

where N is the total number of motes in the network cluster, O_X^T is the expected energy consumption based off the optimal T and C values for a specific NWBR value, and $\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_N$ are weighting coefficients with a value of 0 or 1. The units of H

are *uJ*. The λ coefficients set the NWBR value for which the cost function *H* is being optimized. For a single cluster network size of *N* motes, there are *N* distinct NWBR values and 2^N-1 different combinations of NWBR values that the network can be optimized for. The 2^N-1 combinations result from different combinations of the *N* λ coefficients, each of which can take on a 0 or 1 value, and the $\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_N = 0$ case is trivial so it can be omitted. Once the distinct NWBR values have been measured, all possible combinations can be formed by simple superposition. Based on specific λ coefficients, the optimal parameters can be found by finding the *minimum of H.* Therefore, each $\lambda_X(E_X^T - O_X^T)^2$ product term will be referred to as a NWBR minimizing term, or min-term.

At this point it is important to make a note of 2 things.

1) From this point on, all NWBR values, whether they are distinct or a combination, will be referred to as a NWBR value. For example, $NWBR_2$, $NWBR_4$, and $NWBR_6$ are distinct NWBR values and $NWBR_{2\text{-}4\text{-}6}$ is a linear combination of NWBR values. Both the distinct and the combination are possible NWBR values for the algorithm to be optimized for.
2) A NWBR value is different than a min-term. The NWBR value refers to the number of motes trying to access the channel simultaneously, while the min-term refers to the $\lambda_X(E_X^T - O_X^T)^2$ product term used in the cost function for optimizing the network based on the targeted NWBR value.

In general, the NWBR value for a network can be treated as a random process with a probability distribution function (PDF) that is compiled by the cluster head from previous recorded network traffic patterns. When cluster heads detect a network traffic patterns from its PDF, the cluster head can re-optimize the RA protocol by updating the H function based on the observed NWBR values and then selecting the appropriate minimum p_o value.

For some applications, all NWBR values might be equally likely to occur ($NWBR_{ALL}$), which corresponds to the case $\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_N = 1$. In these situations, the cost function will have to minimize energy across all min-terms. For the case of a 10 mote cluster network, equation 5.7 would become

$$H\left(\{\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_{10} = 1\}, A_W(p_o)\right) = \sqrt{\left(\sum_{X=1}^{10} (E_X^T - O_X^T)^2\right)}. \quad (5.8)$$

5.4.2.2 – Energy Optimization Example Using 3 Min-Terms

Experimentally, three distinct NWBR values were tested. Using the three energy loss functions for these 3 NWBR values, the total energy cost function, *H*, can be defined as $H(\lambda_2, \lambda_6, \lambda_{10}, \lambda_T, A_W(p_o))$, and is given by

$$H = \sqrt{(\lambda_2(E_2^T - O_2^T)^2 + \lambda_6(E_6^T - O_6^T)^2 + \lambda_{10}(E_{10}^T - O_{10}^T)^2)}, \quad (5.9)$$

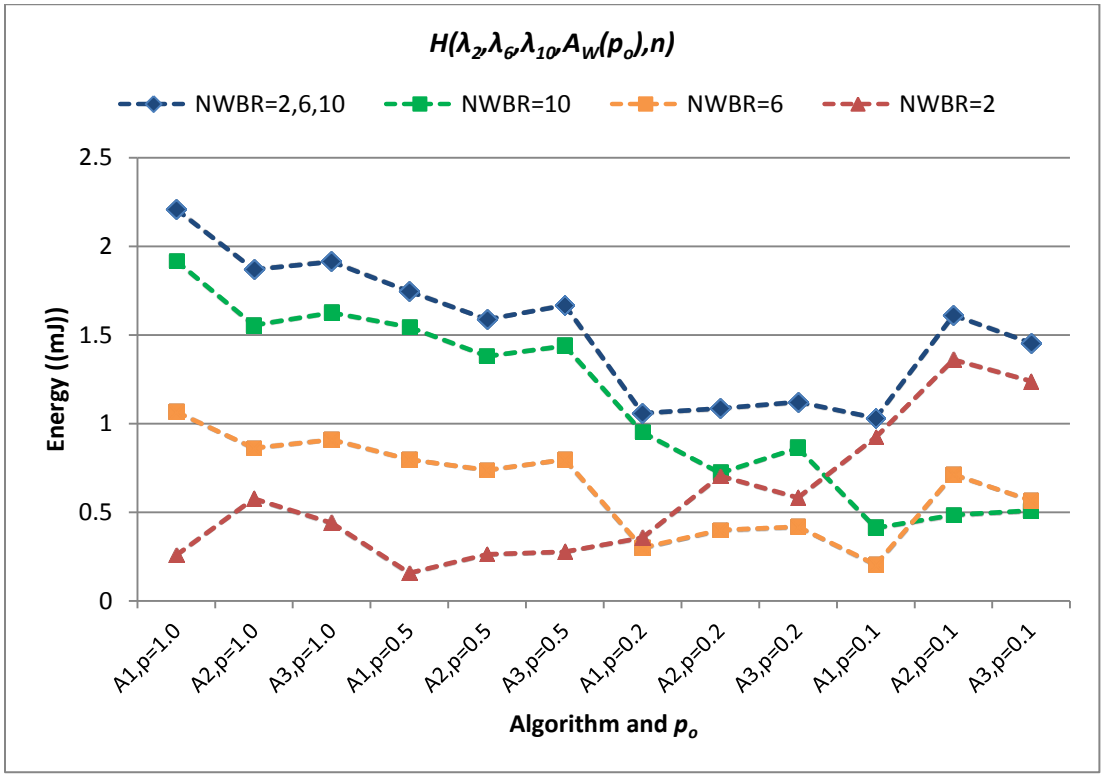
where the O_X^T values are calculated using the values from figure 5.29 in equations 5.4-5.6. In equation 5.9, λ_2, λ_6, and λ_{10} are weighting coefficients that set the NWBR value for which the cost function *H* is being optimized. The 3 tested distinct NWBR values were $NWBR_{10}$, $NWBR_6$, and $NWBR_2$, which correspond to [λ_2=0, λ_6=0, λ_{10}=1], [λ_2=0, λ_6=1, λ_{10}=0], and [λ_2=1, λ_6=0, λ_{10}=1], respectively. In this experiment, since only 3 NWBR values were experimentally tested, $NWBR_{ALL}$ corresponds to a value of $NWBR_{2\text{-}6\text{-}10}$, which is the linear combination of all 3 tested NWBR values, and can be set by [λ_2=1, λ_6=1, λ_{10}=1]. The 4 NWBR values and their corresponding lambda coefficients are shown in figure 5.30.

NWBR Value	Lambda Coefficients [λ_2-λ_6-λ_{10}]
2	[1-0-0]
6	[0-1-0]
10	[0-0-1]
2-6-10	[1-1-1]

Figure 5.30 - Table of 4 different NWBR values and their corresponding λ coefficients.

Using all the experimental results shown in sections 5.4.1.1 and 5.4.1.2 and the expected results from figure 5.29, figure 5.31 shows several plots of *H* for the 4 different NWBR combinations as a function of algorithm and p_o value. The minimum of each *H* function curve is the optimal algorithm and p_o value for that particular NWBR combination. The plot shows that based on 3 min-terms, the $NWBR_2$ curve has different optimal parameters than the other 3 NWBR values. Therefore, if a network has different traffic patterns at different times that occur at or near a particular NWBR value, the cluster head can dynamically adjust the algorithm's

parameters to reduce power consumption according to the recognized traffic pattern.



NWBR Value	Optimal Algorithm
2	$A_1(p_o = 0.5)$
6	$A_1(p_o = 0.1)$
10	$A_1(p_o = 0.1)$
2-6-10	$A_1(p_o = 0.1)$

Figure 5.31 - Several plots of the cost function H for 4 different NWBR values as a function of algorithm and p_o. The plots were generated with equation 5.9 using the experimental results shown in sections 5.4.1.1 and 5.4.1.2 and the expected results shown in figure 5.29.

In particular, the blue curve corresponds to the $NWBR_{2\text{-}6\text{-}10}$, which represents the case where network traffic is occuring at 3 different burst rates. The *H* function shows similar energy consumption with the following algorithms: [A1, p=0.2], [A1, p=0.1], [A2, p=0.2], and [A3, p=0.2]. A clearer picture of how the energy varies for the $NWBR_{2\text{-}6\text{-}10}$ value is shown in figure 5.32 which plots simulated *H* curves for the 3

algorithms using every p_o value. In this figure, the experimental and simulated values fit very well for the A_1 algorithm, pretty well for the A_2 algorithm, but not so well for the A_3 algorithm. The deviations can be more clearly seen in figure 5.33, which plots each algorithm separately. The deviation in fit for A3 can be accounted for by the fact that the results and simulated values shown in figure 5.27 – 5.28 did not match very well, but their relative differences matched, which suggests that the simulated values for A3 can provide information about the trends of the implemented A_3 algorithm. Overall, these results show that based on a minimization process of 3 min-terms where network traffic is expected equally across all NWBR values, the algorithm of W=1 and $p_o = \frac{1}{8}$ provides the best performance in terms of minimizing energy consumed by the RA protocol.

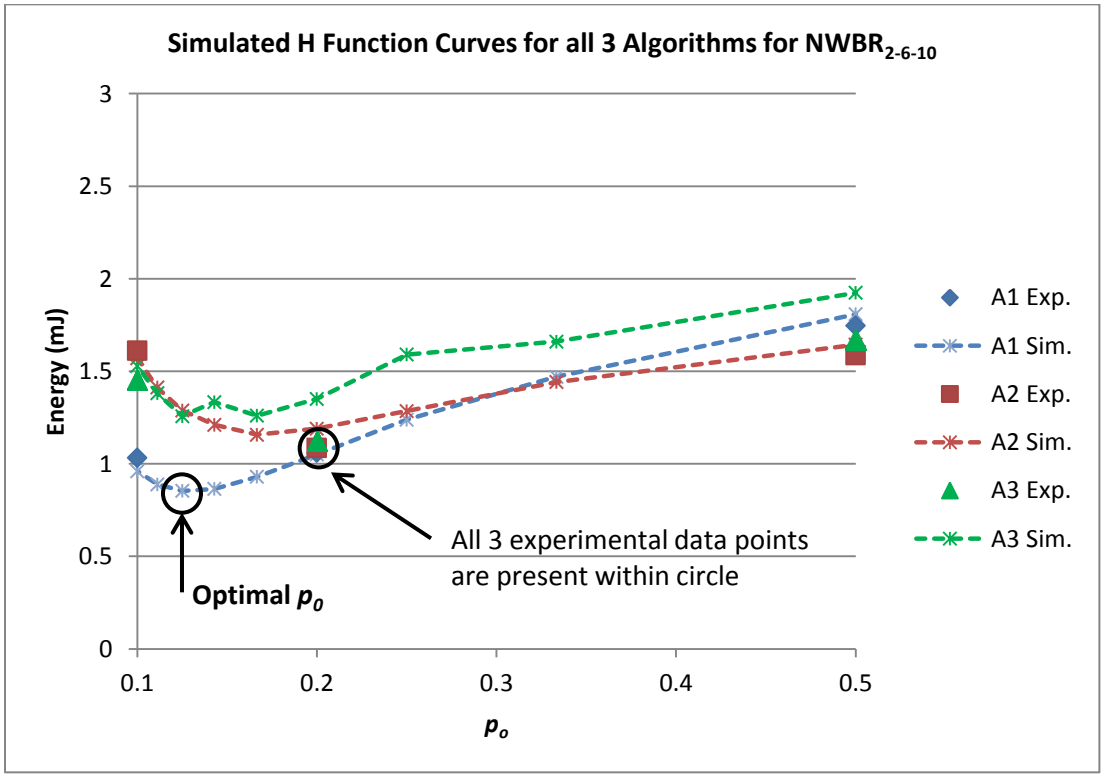


Figure 5.32 - Differences in energy consumption for the $NWBR_{2\text{-}6\text{-}10}$ as a function of algorithm and p_o value. The dotted lines should only be interpreted as guides for the eye.

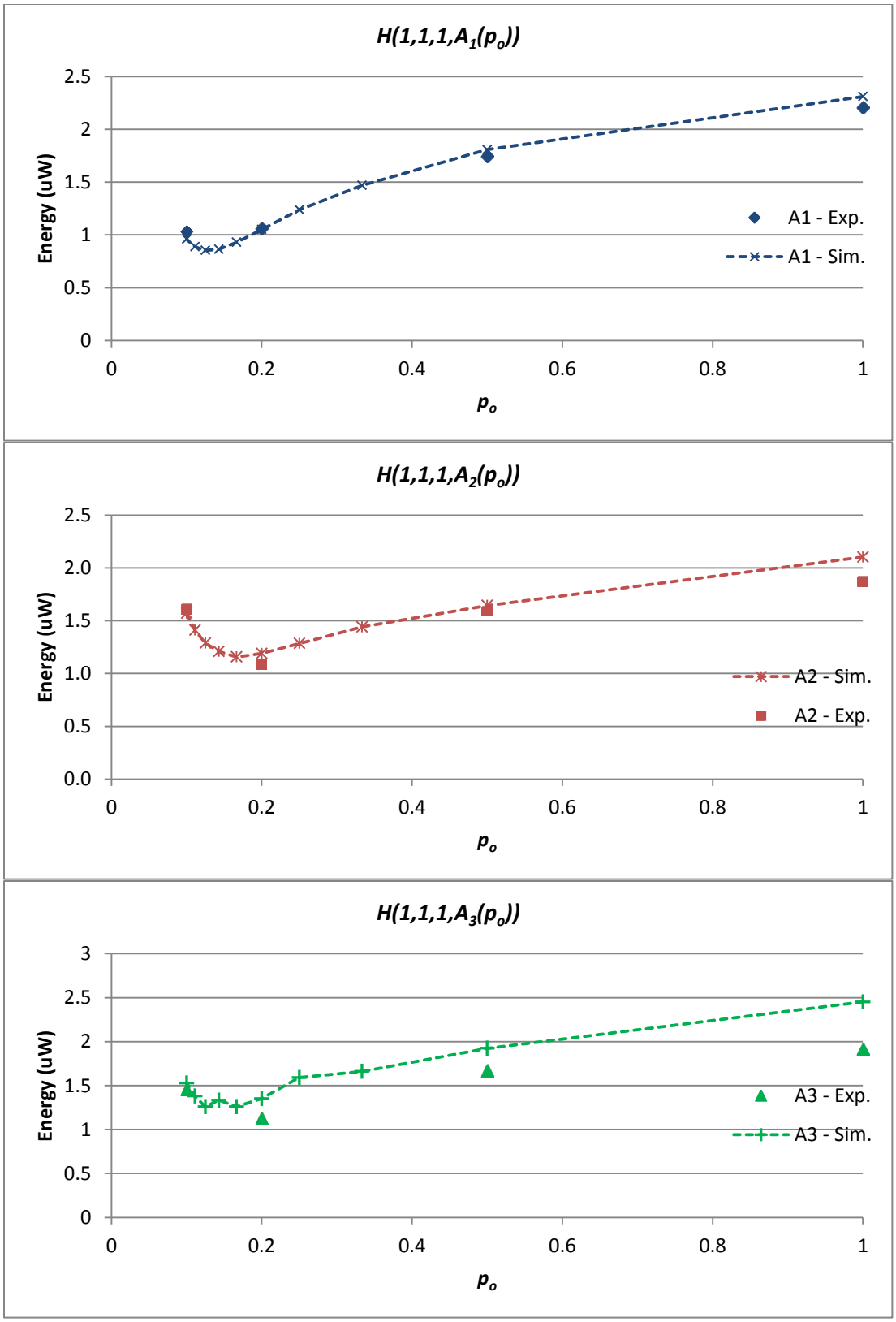


Figure 5.33 - The curves from figure 5.32 are plotted separately. Each plot corresponds to a different algorithm window size and shows the differences in energy consumption for the $NWBR_{2\text{-}6\text{-}10}$ as a function of p_o. The dotted lines are intended to be guides for the eye.

5.4.2.3 – Energy Optimization Using all 10 Min-Terms

In the previous example, only 3 min-terms were used to optimize the network. In general, the optimization process becomes more accurate as more min-terms are included in the optimization process. Figure 5.34 shows the simulation results of *H* being optimized with all 10 min-terms for the different algorithms as p_o is varied. It turns out that the best performance comes from the W=1, $p_o = \frac{1}{7}$ algorithm, which is pretty close to the W=1, $p_o = \frac{1}{8}$ algorithm determined experimentally when only using 3 min-terms in the optimization process. For large p_o values, all 3 algorithms consume the most energy because many collisions are occurring, especially for large NWBR values. As the p_o value decreases, the energy begins to decrease because collisions become less abundant. Instead, the chances of a success or timeout increase. As the p_o value decreases further, the energy continues to decrease because now more timeouts are occurring, which consume less energy than a collision. However, with more timeouts, the RA-protocol needs additional SFs to resolve contention, which increases energy consumption at the cluster head. Thus, at even smaller p_o values, the energy begins to rise again as the additional SFs consume more energy than the timeouts were saving. The minimum of the curve is the optimal point.

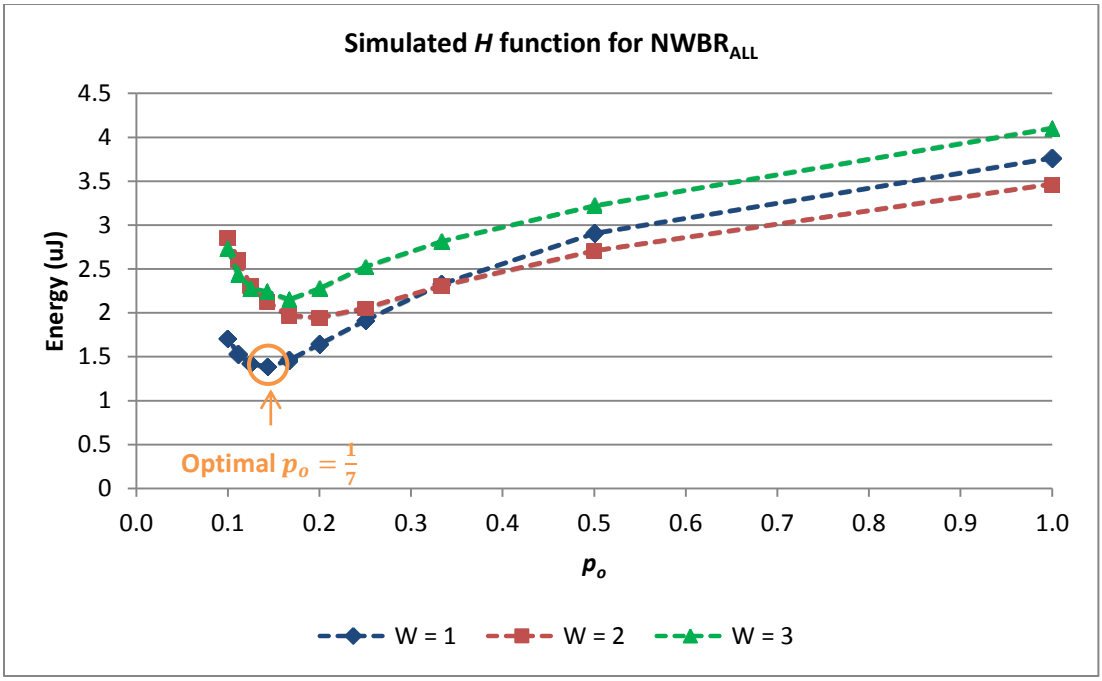


Figure 5.34 - Shows the simulation results of *H* being optimized with all 10 min-terms minimizing terms for the different algorithms as p_o is varied. The dotted lines should only be interpreted as guides for the eye.

Figure 5.35 shows how the different p_o values affect the average energy consumption of the A_1 algorithm by plotting the percent error between the algorithm and optimal average energy for different NWBR values, which is given by

$$R_E\left(X, A_W(p_o)\right) = \frac{(E_X^T - O_X^T)}{O_X^T} * 100 \tag{5.10}$$

In the figure, curves of A_1 with p_o values of $\frac{1}{5}, \frac{1}{6}, \frac{1}{7}$, and $\frac{1}{8}$ are plotted. Each algorithm maintains a R_E value under 40% at NWBR values≥ 4, which shows that most of the energy minimization comes from the larger NWBR values. For smaller NWBR values, the R_E increases rapidly, indicating that most energy is consumed when only a few motes try to access the channel. This means that when all NWBR values are equally likely to occur, corresponding to the $NWBR_{ALL}$ case, the protocol minimizes most of

the energy at larger NWBR values by reducing the amount of collisions by starting at a smaller p_o value. However, the small p_o value makes the protocol less efficient at smaller NWBR values. Specifically, the energy minimized algorithm, A_1 with $p_o = \frac{1}{7}$ (shown as a green solid line), has its minimum R_E value occurring at $NWBR_6$. It maintains less than 30% R_E as long as 4 or more motes are trying to access the channel simultaneously.

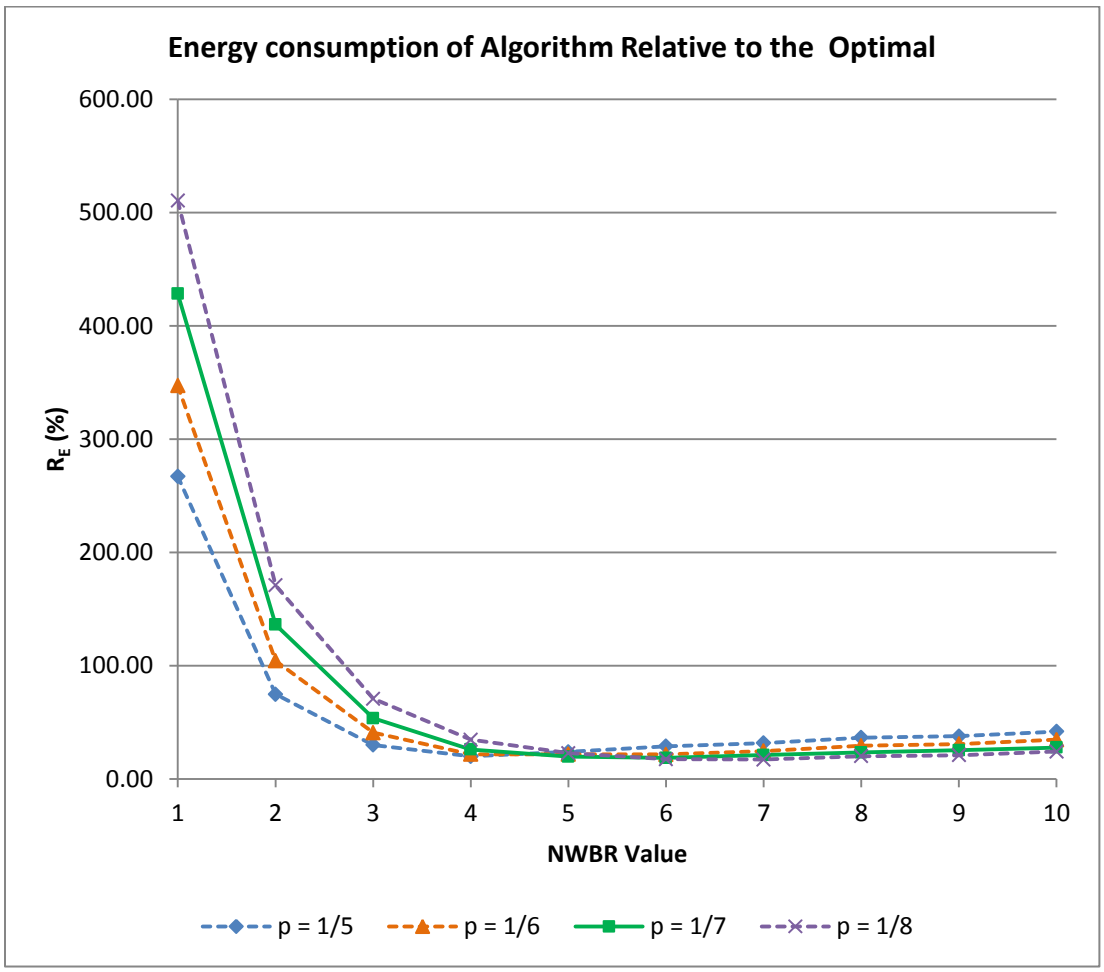


Figure 5.35 - Shows how the different p_o values affect the average energy consumption of the A_1 algorithm by plotting the percent error between the algorithm and optimal average energy for different NWBR values.

5.4.3 – Random Access Performance in Single Cluster Networks

Using $p_o = \frac{1}{7}$, figure 5.36 shows the RA protocol results in resolving mote contention for different NWBR values within a 10 mote network cluster. As can be seen from the figure, the simulated results match very well with the experimental results. The expected optimal performance based on the cluster head transmitting p_{opt} every RAS signal is shown as *<Optimal>* to provide a relative comparison in the algorithm's performance. The algorithm resolves contention within 3 SFs of optimal values for NWBR values of 3 and larger. The algorithm does not work as well for NWBR values smaller than 4 because the initial p value is small to begin with. The average error in SFs between the A_1 and <Optimal> is 2.58, with the largest error of ~5.14 SFs occurring at $NWBR_1$ and the smallest error of ~1.55 SFs occurring at NWBR values of 5 and 6. The standard deviation in SFs for both curves is shown in figure 5.37. The experimental results matched very well with the simulation. The average error between the two standard deviation curves is .51 SFs.

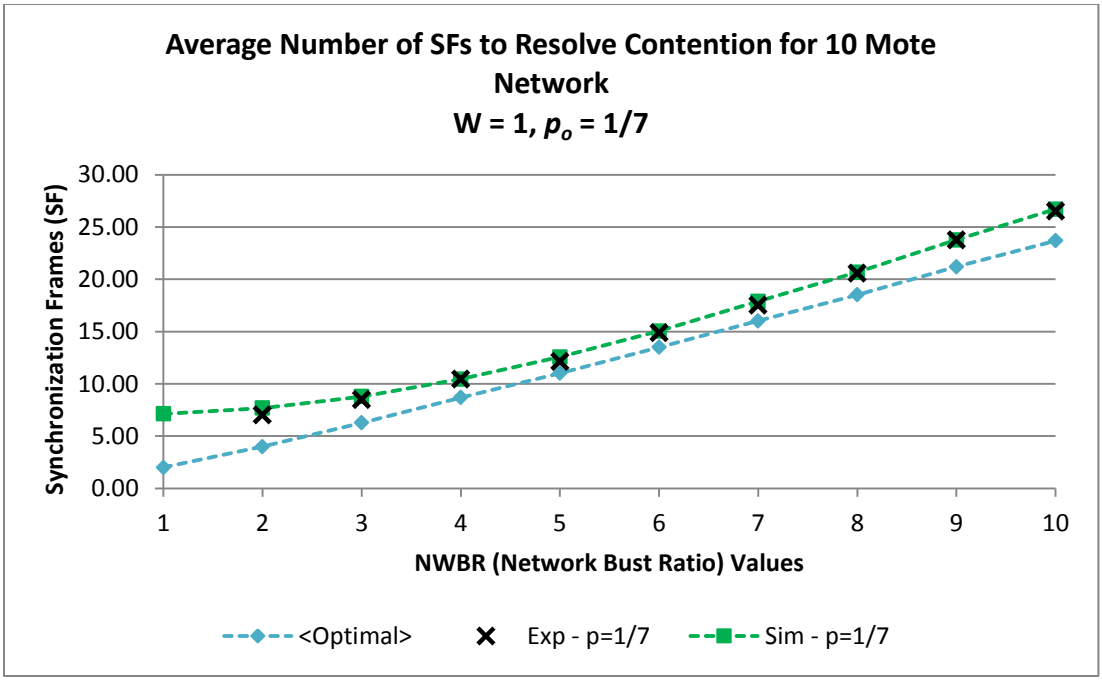


Figure 5.36 - RA protocol results in resolving mote contention for different NWBR values within a 10 mote network cluster when p_o = 1/7. The dotted lines are meant to be interpreted as guides for the eye.

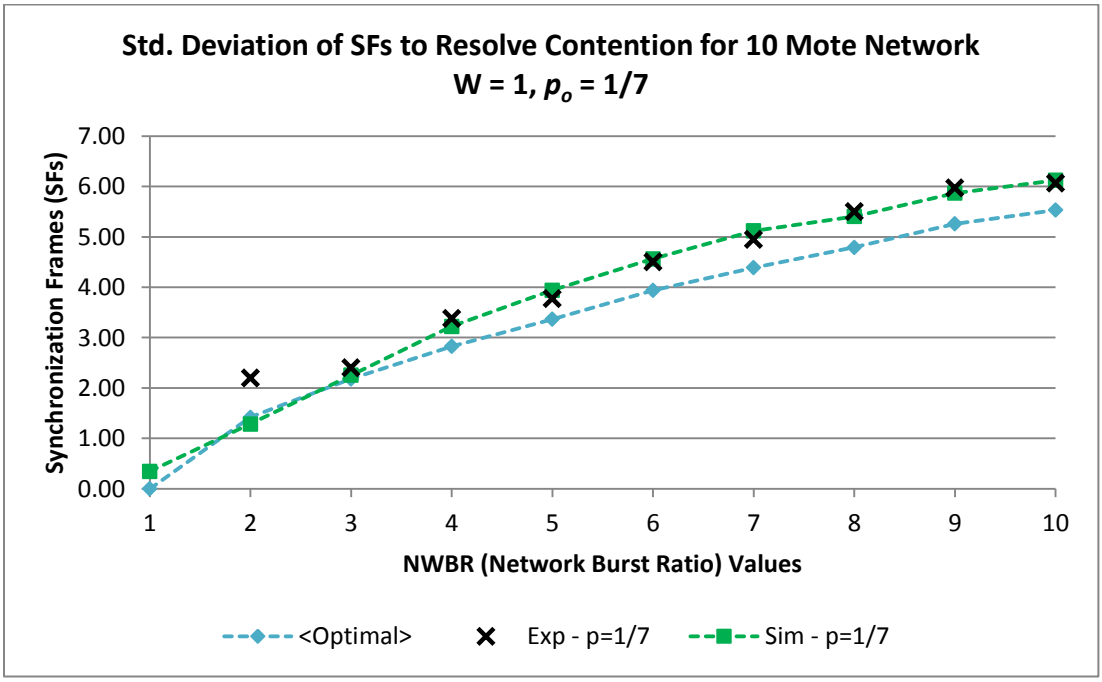


Figure 5.37 - Standard deviation between experimental, simulated, and optimal (analytical) results. The dotted lines are meant to be interpreted as guides for the eye.

5.5 – Single Cluster Network Activity Experiments

Thus far, the RA protocol has been analyzed and optimized based on minimizing energy consumption and random access times in network configurations where all the motes were initially sleeping and then triggered to wake up by a cluster head signal. Upon waking, the motes tried to access the channel, which caused collisions and forced the RA protocol to run. Therefore, the RA protocol results reflected a network traffic pattern that consisted of a single "distinct" NWBR value.

In realistic applications, network data traffic can potentially occur at many different NWBR values. The objective of the following experiment is to simulate a more practical network and measure how well the FSOC sensor network communication protocols can handle data communication and random access contention.

In this experiment, all the motes in the network wake up on their own at random times and then try to transmit 10 packets of data to the cluster head. Due to the random wake up times, the cluster head will detect network traffic that occurs at multiple NWBR values instead of a single one. In addition, the random wake up times will affect the number of packet collisions during D-TDMA communication and cause the RA protocol to run multiple times which, depending on the set p_o value, can potentially increase the total number of SFs. Together, these events will affect the network's efficiency, which is a measure of the number of data packets vs. non-data packets being transmitted.

5.5.1 – Experimental Configuration

The experiment was conducted on a 10 mote configuration, which is the same setup as shown in figures 5.1 – 5.8. An experimental run begins with the cluster head transmitting a signal to all the motes that synchronizes all of the clocks. Then, each mote wakes up after a random time delay (R_D) in ms given by,

$$R_D = 8.2 * n \quad \text{(ms)} \tag{5.11}$$

where *n* is a randomly generated integer within certain bounds. Different bounds were used to generate different types of network congestion patterns. Small bounds mean that the average time difference between motes waking up $\langle T_{Diff}^{M} \rangle$ will be small, while large bounds mean that the $\langle T_{Diff}^{M} \rangle$ will be large. If the $\langle T_{Diff}^{M} \rangle$ is large, then on average, whenever the RA protocol runs, it will only assign a small number of motes a D-TDMA time slot. However, as the $\langle T_{Diff}^{M} \rangle$ is reduced, there will be more motes trying to access the channel in a shorter time interval, so the RA protocol will assign a larger number of motes D-TDMA time slots each time it's run. The different bounds and their corresponding network congestion level, which is a measure of how many motes are expected to be assigned a D-TDMA time slot each RA protocol run, are shown in figure 5.38. A low network congestion level means that a small number of motes will be assigned to the queue, while higher congestion levels indicate larger numbers of motes.

Network Congestion	Random integer *n* bounds	Delay Bounds (ms)
Low	$0 \leq n \leq 127$	$0 \leq R_D \leq 1041.4$
Medium	$0 \leq n \leq 63$	$0 \leq R_D \leq 516.6$
High	$0 \leq n \leq 31$	$0 \leq R_D \leq 254.2$
Very High	$0 \leq n \leq 15$	$0 \leq R_D \leq 123.0$

Figure 5.38 - Definition of the different network congestion levels.

After each mote wakes up, the mote transmits 10 data packets to the cluster head. One data packet can be transferred every D-TDMA time slot (TS_{Data}). In this experiment, the time duration of a RA protocol SF is the same as a TS_{Data}.

Since a total of 10 motes woke up during an experimental run, a total of 100 packets were collected at the cluster head. Based on the number of interruptions and packet collisions, several data packets had to be re-transmitted in order to arrive successfully at the cluster head. In addition, depending on how many times the RA protocol ran, multiple SFs were transmitted. The total number of SF and retransmitted data packets determines how efficient the network is operating. The network efficiency (NE) percentage is given by.

$$NE = \frac{100}{100 + SF + Retransmits} * 100 \qquad (5.12)$$

The experimental run finishes after each mote successfully transmits its 10 packets to the cluster head. The experiment consisted of approximately 1000 runs per network congestion level per algorithm. The measurements consisted of:

- Number of packets successfully received
- Number of packet retransmissions
- Number of times the RA protocol ran

- Number of SFs per RA protocol run
- Number of motes assigned a time slot per RA protocol run

5.5.2 – Experimental Results

Figures 5.39 and 5.40 provide a measure of the differences between the different congestion levels. Specifically, figure 5.39 shows the average number of times the RA protocol ran during each of the different congestion levels for the different algorithms, while figure 5.40 shows the average number of motes that were assigned a time slot within each RA protocol run. The protocol runs multiple times at lower congestion levels. As the congestion level increases, motes begin to wake up within a few frames from one another, so the RA protocol can assign time slots to multiple motes within a single run.

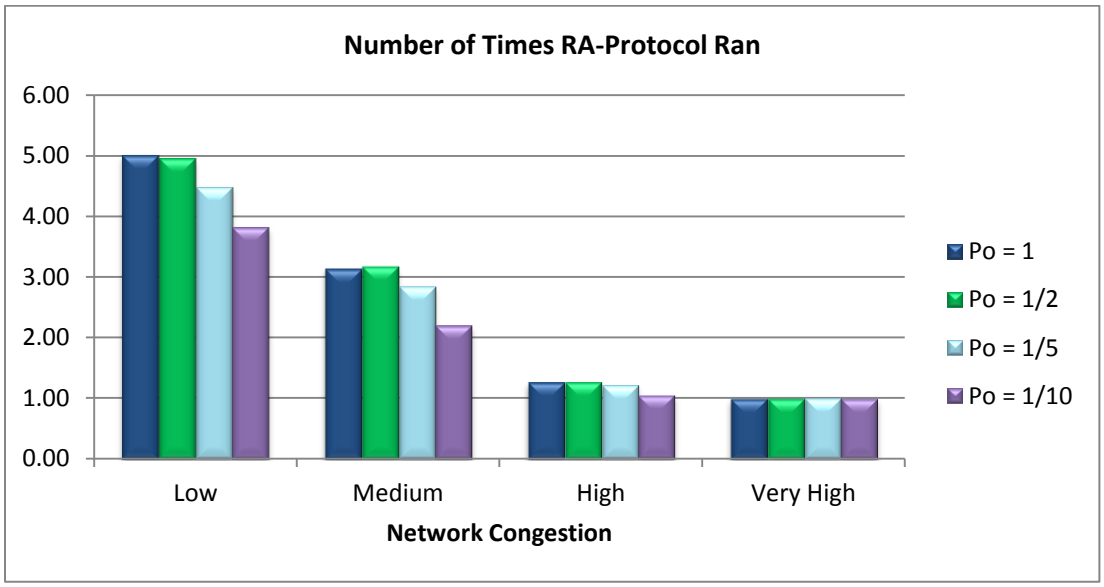


Figure 5.39 - The number of times the RA protocol ran at each network congestion level for different p_o values.

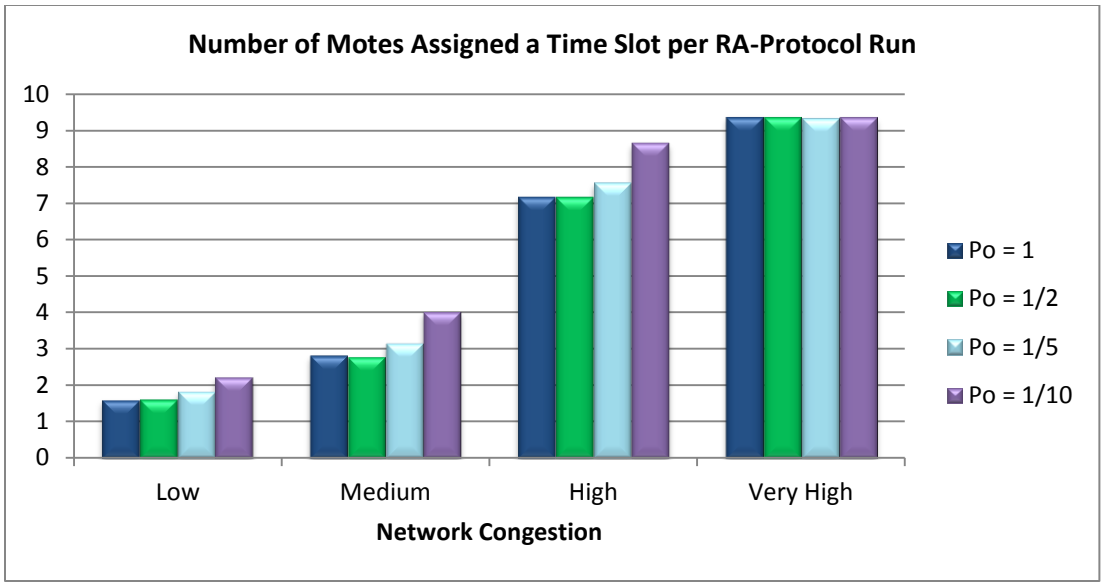


Figure 5.40 - The number of motes assigned a time slot per RA protocol run at the different network congestion levels for different p_o values.

Figures 5.41 and 5.42 shows the total number of SFs required to acquire all 10 motes during each experimental run and the number of SFs within each protocol run, respectively. The different p_o values performed slightly different at the low and medium congestion levels. At these lower congestion levels, the smaller p_o values are less efficient, because fewer motes are contending for channel access. The result is that protocols take more SFs to finish. Hence, $p_o = \frac{1}{10}$ requires the most SFs at low and medium levels.

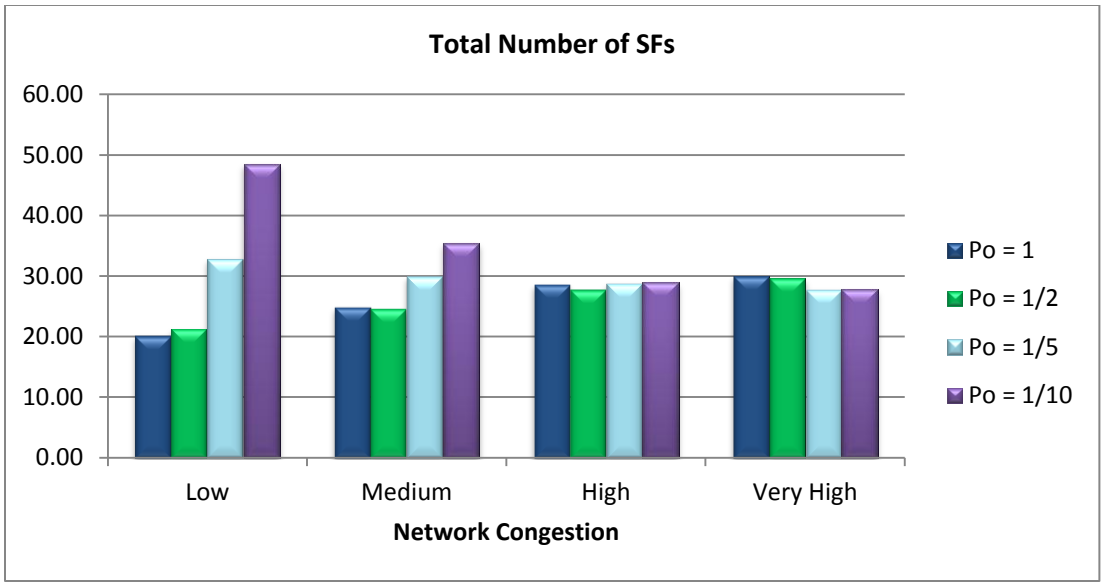


Figure 5.41 - Total number of SFs required to assign all motes for the different network congestion levels.

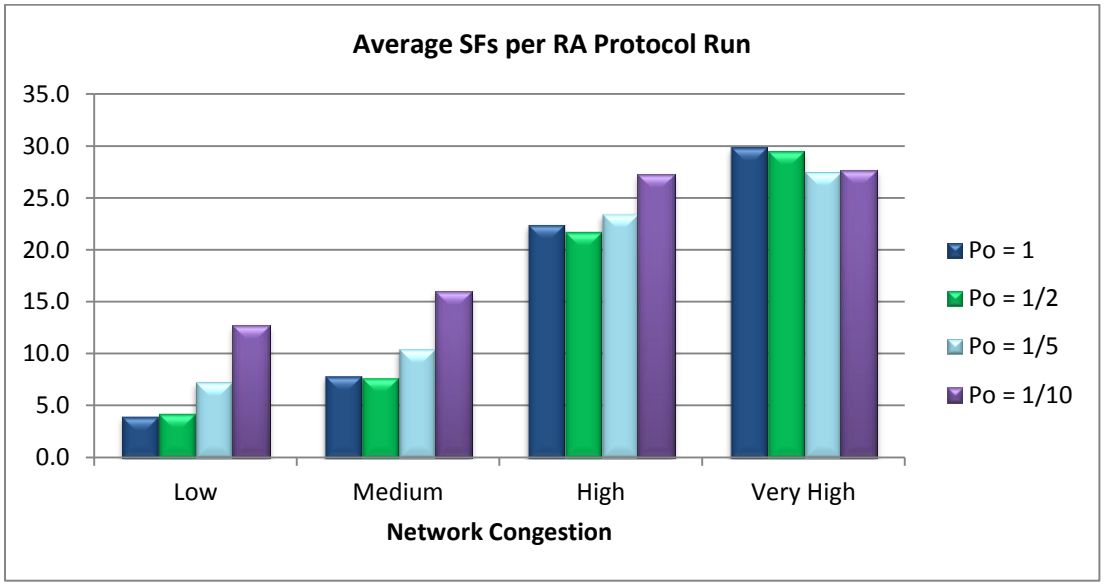


Figure 5.42 - Average number of SFs per RA protocol run at each of the different network congestion levels.

Figure 5.43 shows the total number of data packets that needed to be retransmitted from all the motes to the cluster head at the different congestion levels. The largest number of retransmitted packets occurred at the low congestion level because most motes were waking up and accessing the channel in widely spaced time intervals,

thereby forcing multiple RA protocol runs by causing packet collisions at the cluster head.

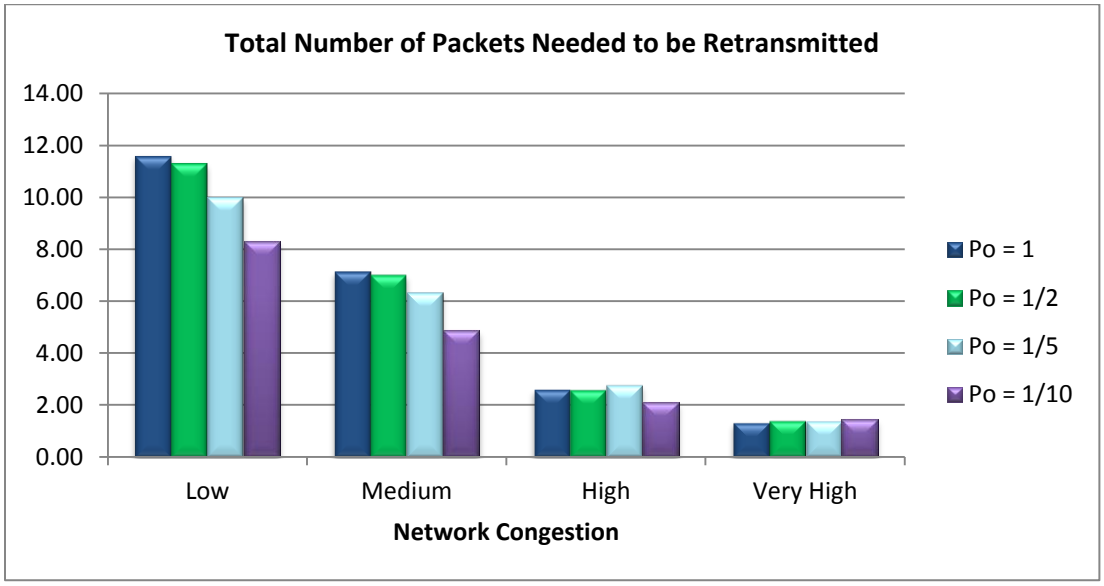


Figure 5.43 - The total number of data packets that needed to be retransmitted from all the motes to the cluster head at the different congestion levels.

The network efficiency, which was defined in equation 5.12, is shown in figure 5.44. At high and very high congestion levels, all algorithms perform relatively the same, operating at ~77% efficiency. As the congestion level decreases, p_o begins to affect the network efficiency. Specifically, at the low level with $p_o = 1.0$, the efficiency is ~77% but with $p_o = 0.1$ the efficiency drops to ~64%. At each congestion level, the best protocol performs close to ~77% efficiency. These efficiency values are based on the motes transmitting 10 data packets each. If motes transmit more than 10 data packets when waking up, the network efficiency will rise. Figure 5.45 shows how the NE depends on the number of data packets that each mote (within a 10 mote network cluster) transmits to the cluster head after waking up.

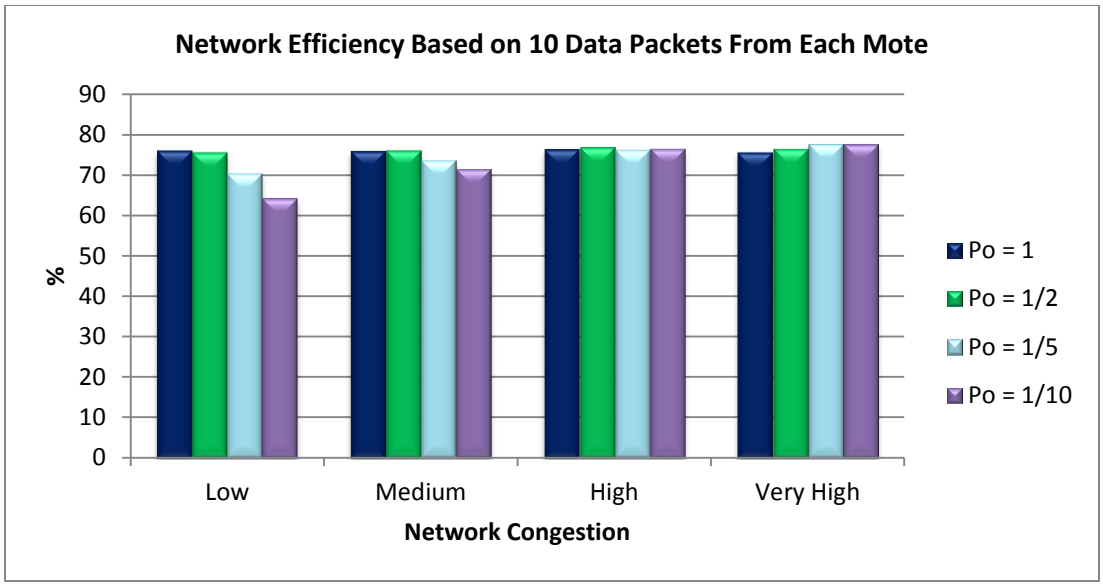


Figure 5.44 - Experimental results of network efficiency when each mote transmits 10 data packets.

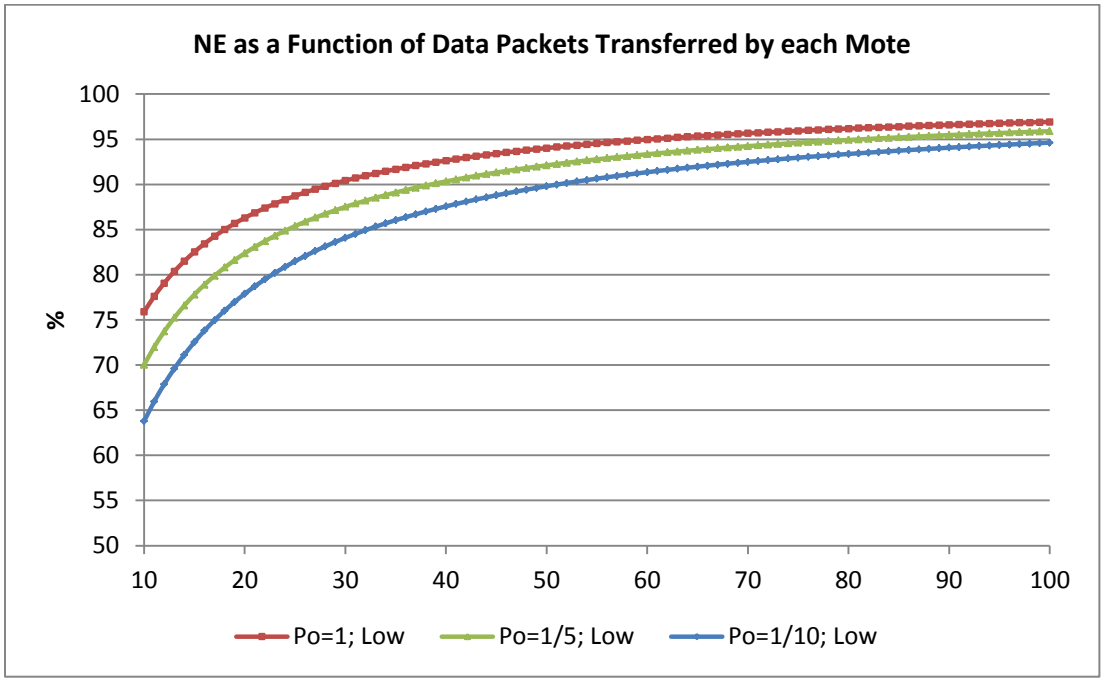


Figure 5.45 - Network efficiency increases as the number of data packets that each mote (within a 10 mote network cluster) transmits to the cluster head after waking up. The curve $p_o = 0.5$ is not shown because it is very similar to $p_o = 1.0$.

The NE% asymptotically approaches 100% as the number of data packets approaches infinity. The curves for the other congestion levels and algorithms fall in

between the "$p_o = 1.0$; Low" and "$p_o = 0.1$; Low" curves. The main result is that the FSOC sensor network DMAC protocol becomes more efficient as the number of data packets increases.

5.5.3 – Probability Distributions of Experimental Results

From these measurements, several probability distribution functions (PDFs) were generated from the 1000 experimental runs. These distributions provide meaningful data on how efficient the FSOC sensor network is at resolving contention at the different congestion levels and different p_o values.

SFs to Assign a Single Mote PDF: These distributions are shown in figure 5.46 for $p_o = 1, \frac{1}{5}, \frac{1}{10}$ for each of the congestion levels. The distributions show the probability of how many SFs are required to assign a mote a time slot. The distribution provides a measure of comparison between random access times of the RA protocol at different congestion levels and different p_o values. At low congestion, $p_o = 1$ provides a significant advantage over the $p_o = \frac{1}{5}, \frac{1}{10}$ algorithms, but as the congestion increases, each p_o value assigns motes at relatively similar rates. At very high congestion, the smaller p_o values slightly outperform $p_o = 1$.

An interesting characteristic of the distributions for the low and medium congestion levels is that the $p_o = \frac{1}{5}, \frac{1}{10}$ curves have 2 peaks. The first peak corresponds to a smaller SF value, which represents the situation where multiple motes are assigned

a time slot during a RA protocol run. The second peak corresponds to a larger SF value, and represents the situation where only 1 mote is assigned a time slot when the RA protocol runs.

SFs to Assign all Motes PDF: These distributions are shown in figure 5.47, and they show the total number of SFs required to assign all motes a time slot. The distribution provides a measure of how long it takes to acquire all motes over an experimental run. At low levels of congestion, there is a clear difference in total SFs between all three p_o values, but as congestion increases, all three curves get closer together. At very high congestion, the $p_o = \frac{1}{5}, \frac{1}{10}$ curves perform slightly better than the $p_o = 1$ curve in terms of random access time.

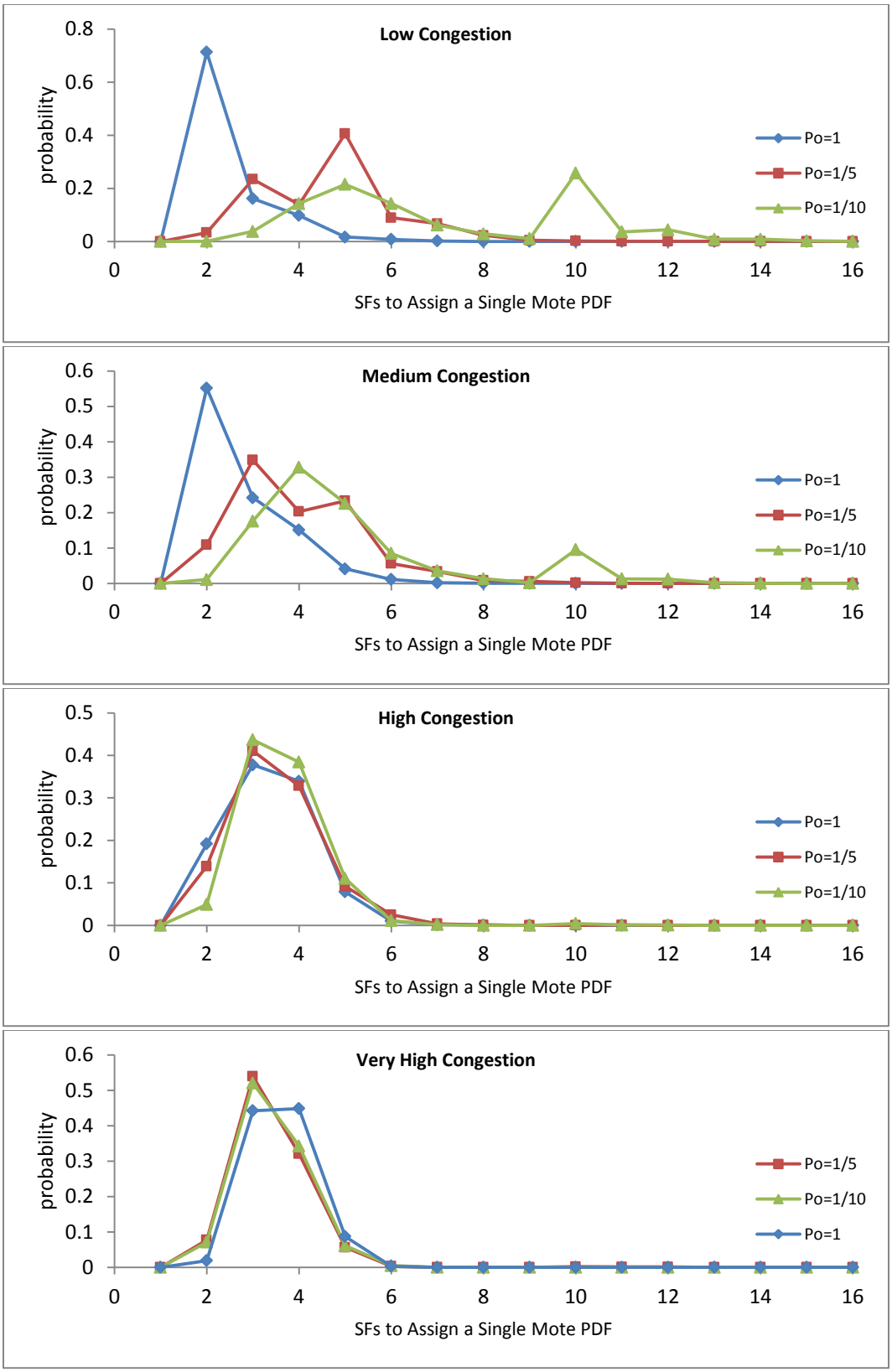


Figure 5.46 - Probability distribution functions of the number of SFs required to assign a single mote a time slot whenever the RA algorithm runs. The plotted lines are guides for the eye.

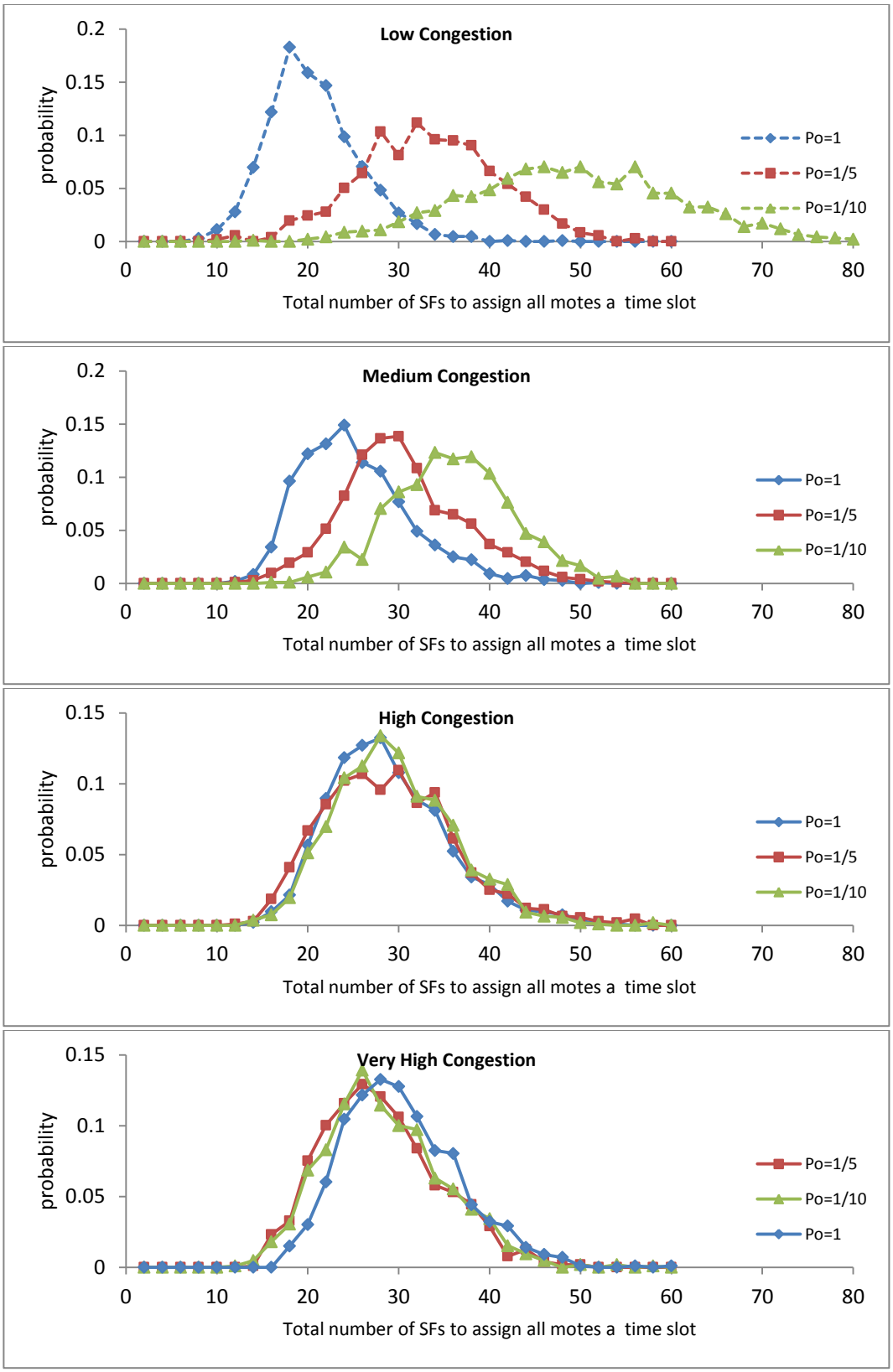


Figure 5.47 - Probability distribution functions of the total number of synchronization frames (SFs) to assign all 10 time slots to the motes. The plotted lines are guides for the eye.

Network Efficiency PDF: These distributions are shown in figure 5.48, and they plot the probability distribution of network efficiency for the different congestion levels and different p_o values. At low congestion, the $p_o = 1$ algorithm has the best efficiency. As the congestion levels increase, all three curves begin to perform similarly, but at very high congestion, the $p_o = \frac{1}{5}, \frac{1}{10}$ algorithms slightly outperform the $p_o = 1$ algorithm.

NWBR PDF: These distributions are shown in figure 5.49. The distributions represent the probability distribution of the different network burst rate (NWBR) values that were detected by the cluster head. An important result is that regardless of the algorithm's p_o value, each algorithm generated a similar NWBR PDF at each of the different congestion levels. The low congestion level was mostly dominated with NWBR values of 1, 2, 3, and 4, with 1 having the highest probability. At medium congestion, the probabilities of all NWBR values slightly increased, but the largest probabilities remained in the lower range of NWBR values. At high congestion, the probability of $NWBR_9$ value dominated each PDF, and at very high congestion, the probabilities of NWBR values of 9 and 10 dominated each PDF.

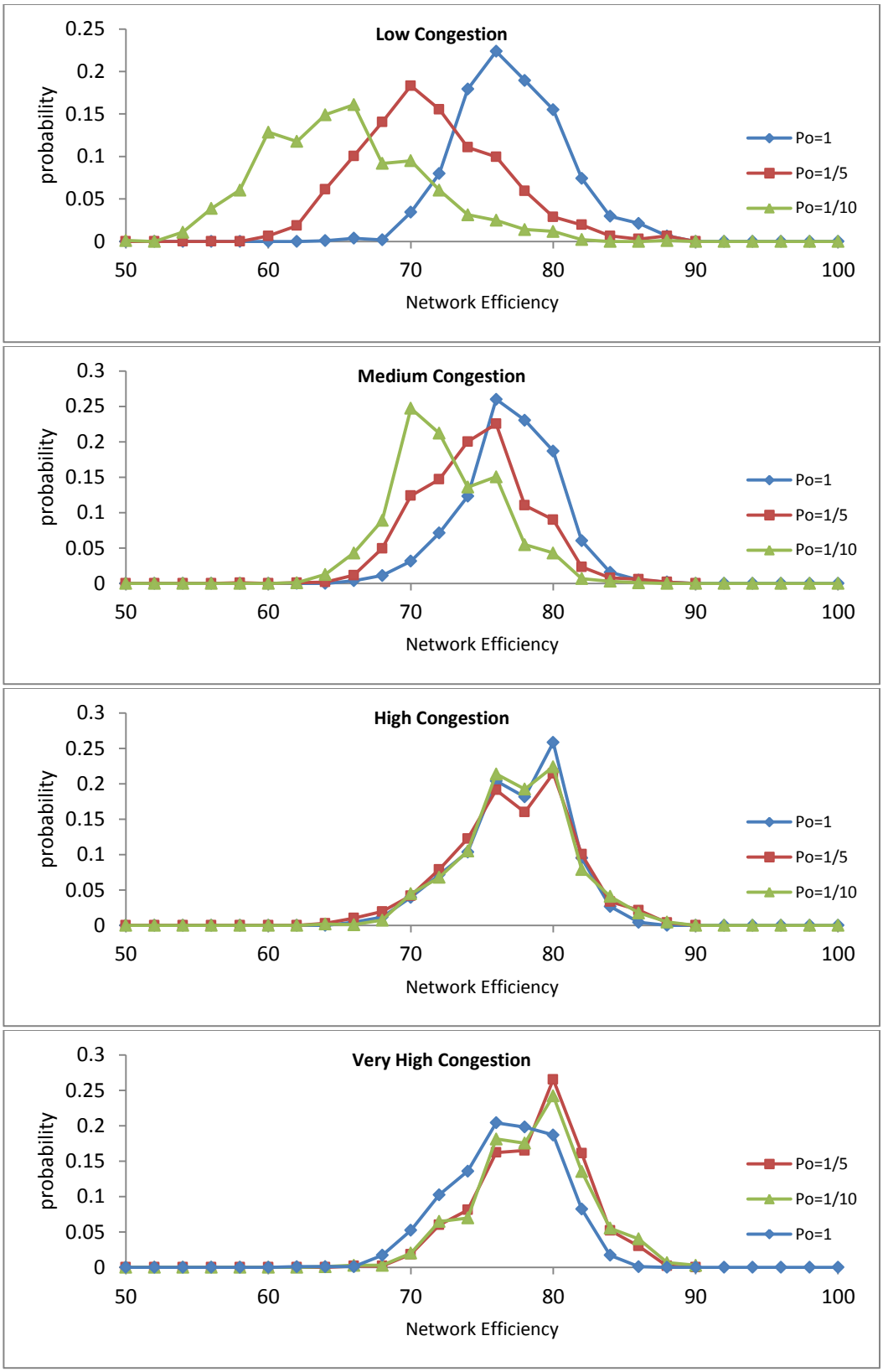


Figure 5.48 - Probability distribution functions of the network efficiency when the motes transmit 10 packets each. The plotted lines are guides for the eye.

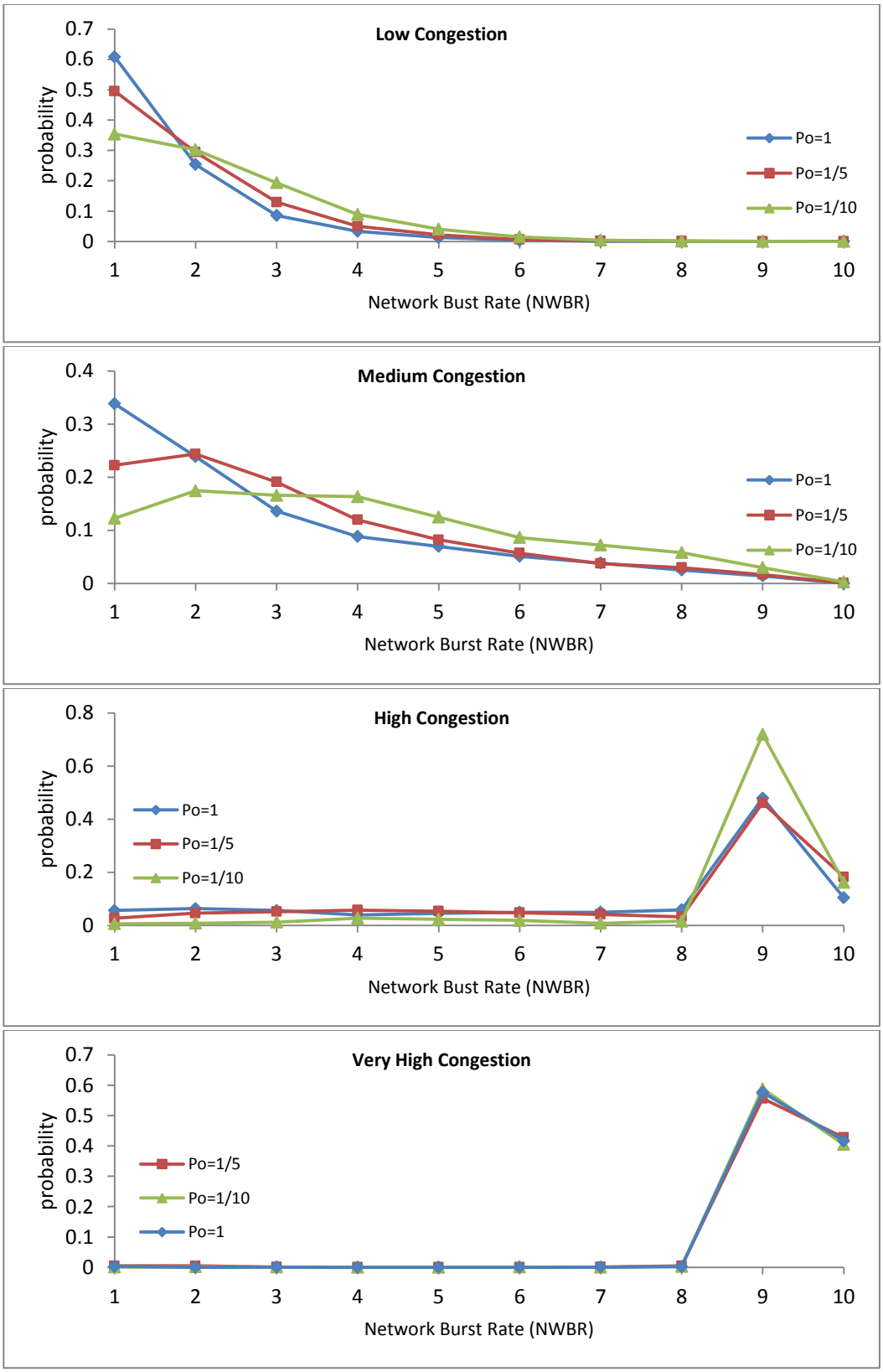


Figure 5.49 - Probability distribution functions of the NWBR. As congestion increases, the probability for larger NWBR values increases. The plotted lines are guides for the eye.

5.5.4 - Network Performance Analysis

The two main considerations when optimizing the RA algorithm parameters are:

1) Random Access Times
2) Energy Consumption

These two quantities can now be minimized by updating the cost function *H* to include the probability scaled values from the generated NWBR PDF. In this manner, based on the cluster head's detected network traffic patterns, the cluster head can update its NWBR PDF, and extract the probability scaled values of the different NWBR values. These extracted values can then be incorporated into the cost function *H* by,

$$H\left(A_1, A_2, A_3 \dots A_N, \lambda_1, \lambda_2, \lambda_3, \dots \lambda_N, A_W(p_o)\right) = \sqrt{\left(\sum_{X=1}^{N} A_X \lambda_X (E_X^T - O_X^T)^2\right)} \quad (5.13)$$

where $A_1, A_2, A_3 \dots A_N$ represent the scaled probability values from the NWBR PDF.

Using the scaled values from figure 5.49 into equation 5.13, along with the min-term and optimal values, figure 5.50 shows the computed *H* optimized p_o value for each of the NWBR PDF curves.

Congestion Level	Low			Medium			High			Very High		
Algorithm's Preset p_o	1	$\frac{1}{2}$	$\frac{1}{10}$	1	$\frac{1}{2}$	$\frac{1}{10}$	1	$\frac{1}{2}$	$\frac{1}{10}$	1	$\frac{1}{2}$	$\frac{1}{10}$
H optimized	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$

Figure 5.50 - Computed *H optimized* p_o values using scaled NWBR values from figure 5.49 into equation 5.13.

Regardless of the algorithm's initial parameters, each algorithm gets optimized to nearly the same p_o for low and medium congestion, and the same p_o value for high and very high congestion. Thus, the cluster head can converge to the optimal settings regardless of its initial parameters.

The results from this experiment show that the cluster head can dynamically optimize the RA protocols parameters based on the history of detected traffic patterns as long as the traffic patterns are not completely random. For completely random traffic patterns, the cost function *H* needs to set all coefficients $A_1, A_2, A_3 \ldots A_N$, and $\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_N$ equal to 1.

5.6 – Scalability to Larger Single Cluster Network Sizes

Figure 5.51 shows simulations results of how the algorithm A_1, $p_o = \frac{1}{7}$ would perform when the number of motes within a network cluster equals 100. The algorithm deviates from the expected optimal values linearly at a rate approximately equal to $.6121 * NWBR$ value. The largest difference in performance is approximately 62 SF.

When moving to larger network sizes, the algorithm's performance can be improved by simply scaling p_o

$$p_o = \frac{1}{7 * k} \tag{5.14}$$

where k is an integer that is determined by

$$k = ceil\left(\frac{NWBR}{10}\right) \tag{5.15}$$

where $ceil$ is the mathematical ceiling function that rounds the argument up to the nearest integer, and the NWBR value can be determined from the cluster head's NWBR PDF.

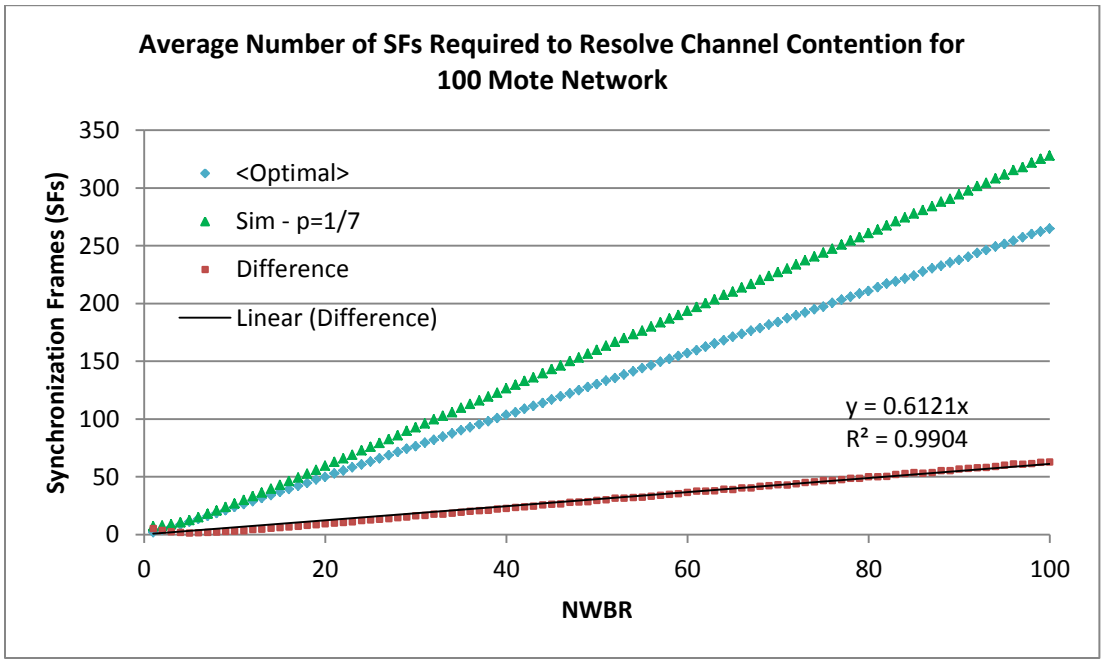


Figure 5.51 - Simulation results of how algorithm A1 with p_o=1/7 would perform on a 100 mote network.

Figure 5.52 shows the increase in performance of the scaled algorithm in comparison to the non-scaled algorithm and the optimal values.

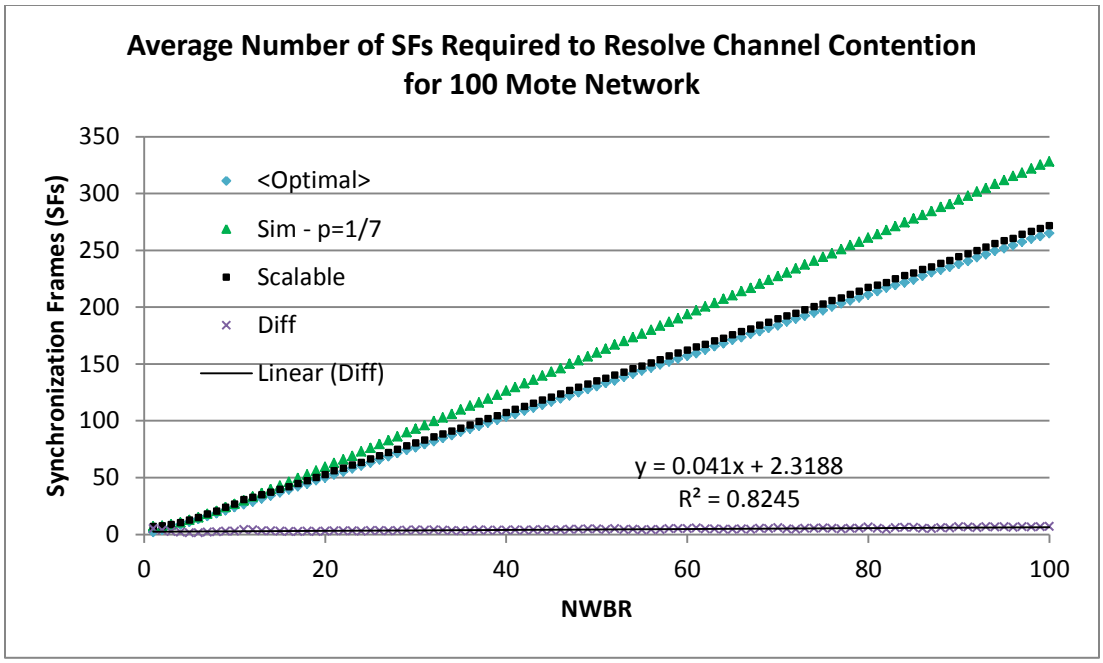


Figure 5.52 - Simulation results showing how performance can be increased from using the scaled algorithm.

The deviation in performance between the scalable algorithm and the optimal values can be linearly approximated by the equation

$$\Delta_{Diff} = .041 * NWBR + 2.3188, \quad (5.16)$$

resulting in a significant increase in performance, which is tabulated in figure 5.53.

	Scaled Algorithm	Non-scaled Algorithm
Average Error Difference	4.39 SFs	30.189 SFs
Std. Dev. in Errr Difference	1.31 SFs	19.06 SFs
Maximum Error Difference	6.92 SFs	63.013 SFs

Figure 5.53 - Performance comparison between scaled and non-scaled algorithm on a 100 mote network.

The simplistic nature of this scalable algorithm leads to an easy implementation into the microcontroller.

Chapter 6 – Conclusions

6.1 – Closing Thoughts

While radio frequency systems have been the traditional solution to free space communications, optical communication systems are attractive in applications where physical security, low levels of interference, and potential for extremely high data rates are critical. These characteristics result from the wavelength and corresponding high directionality of free space optical communication (FSOC). Because of these characteristics, the design and architecture of a FSOC system is different from traditional RF communication systems. Thus, the objective of this dissertation was to show the design, analysis, and optimization of a FSOC system and directional media access control (DMAC) protocol that could provide multipoint communication with highly directional beams.

The developed DMAC algorithm was implemented on a prototype network of 10 motes and 1 cluster head. One of the key factors of the algorithm is how fast it can add motes onto the dynamic time division multiple access (D-TDMA) queue, and this factor was measured by the number of synchronization frames (SFs) required to resolve channel contention between the recently awaken motes. In order to achieve theoretical performance, the algorithm needs to transmit the optimal transmission probability (p_{opt}) during each SF, but this would require the cluster

head to know the network load *a priori*, which is not a realistic assumption for a practical network. Instead, the cluster head must assume that any number of motes may potentially wake up simultaneously and request channel access at any given time. Based off this assumption, the algorithm tries to estimate the p_{opt} values to transmit during each SF by analyzing the number of successful transmissions, timeouts, and packet collisions during the SFs. The algorithm performs best when the initial transmission probability (p_0) is set closest to the p_{opt}, but since the network load is not known *a priori*, there is no straightforward way of doing this.

However, using an energy optimization method that was based on the number of successful transmissions, timeouts, and collisions during a SF, it was shown that a cluster head can dynamically update its initial transmission probability so that the algorithm could achieve near-theoretical performance without knowing the network load *a priori*. This was experimentally shown for the case of 10 motes and 1 cluster head. In this network configuration, if there is an equal probability of all network burst rate (NWBR) values to occur, near-theoretical performance can be obtained with p_0 set to $\frac{1}{7}$.

Furthermore, if the NWBR values are not equally probable, the cluster head can monitor the network traffic and obtain probability values for each NWBR value so that a more efficient p_0 value can be configured. It was shown that regardless of the algorithm's $p_0$value, each algorithm was optimized to nearly the same p_0 value for low and medium congestion, and the same p_0 value for high and very high

congestion. Thus, the cluster head can dynamically adjust the algorithm's parameters in real-time to increase network performance without any knowledge of the network load *a priori*.

In general, the optimization process requires a lot of computations, so applying the method for network sizes up to 100 motes would not be practical for microcontroller based systems. Instead, a simple scalable version of the algorithm was developed in which p_0 gets scaled by a factor of $1/k$, where k is determined from the aggregated network burst rate probability distribution function. This scaled algorithm was simulated on a network size of 100 motes, and the algorithm performed very close to the theoretical ideal performance. The average deviation in SFs between the scaled algorithm and the theory was 4.39 SFs with a maximum deviation of 6.92 SFs and a standard deviation of 1.31 SFs across the 100 NWBR values. The simulations and experimental results agreed very well for a network size of 10 motes, which adds validity to the simulation results for a network size of 100 motes.

In summary, the objectives of this dissertation were to present:

1. the design of small, inexpensive, and fully functional SR-FSOC systems capable of transmitting and receiving packets
2. a MAC protocol for FSOC that allows point to multipoint communication using directional links

3. a method to optimize the MAC algorithm's parameters to increase network performance without knowing the network traffic load *a priori*

6.2 – Future Research Directions

The core of the system was developed in "C" and imbedded into ATmega644p microcontrollers. The advantage of the MCU is that it can directly modulate and drive multiple VCSEL lasers, saving overall space and complexity in design. However, the drawback is that it limits the optical bitrate. The MCU operated at 8 MHz, and was able to provide software-driven modulation up to 100 kb/s. The data rate can be increased by moving to faster microcontrollers or field programmable gate arrays.

The hardware that was experimentally tested was designed with 10° optical beams. In principle, the beam can be much narrower if the overall size of the transceiver was not a limitation or if higher density integrated VCSEL arrays became more readily available. Also, the developed hardware only had 1 photodiode. By increasing the number of photodiodes, the number of independent transmitter and receiver channels would increase and spatial multiplexing could be used to further increase the total data throughput.

In general, there are still areas where research and development is needed to make low power SR-FSOC sensor networks more attractive as compared to traditional RF systems. Some of these areas include:

1. Increase the optical data rate of the hardware to much faster speeds than radio frequency communication
2. Increase signal coverage while maintaining small transceiver sizes
3. Increase the range of the devices
4. Implement solar/laser recharging capability to increase energy efficiency
5. Increase the number of independent Rx/Tx channels so that multiple pathways can exist between a mote and the end destination to ensure link stability in the case of a link obstruction
6. Further evaluation of the DMAC protocol in dynamically changing environments where multiple motes move in and out of different network clusters

Works Cited

1. *Wireless infrared communications.* **Khan, J. M. and Barry, J. R.** 1997. Proc. IEEE. Vol. 85, pp. 265-298.

2. **Barry, J. R.** *Wireless Infrared Communications.* Boston : Kluwer, 1994.

3. *Short-range optical wireless communications.* **O'Brien, D.C., et al.** 2005. Wireless World Research Forum.

4. *Free space optical sensor network for fixed infrastructure sensing.* **Agrawal, N., Milner, S. D. and Davis, C. C.** 2009. Proc. SPIE, Free-Space Laser Communications IX. Vol. 7464.

5. *Gigabit class high-speed indoor optical wireless; system design, challenges and results.* **O'Brien, D. C., et al.** 2010. Proc. SPIE, Free-Space Laser Communications X. Vol. 7814.

6. *Wireless Optical Network for a Home Network.* **Bouchet, O., et al.** 2010. Proc. SPIE, Free-Space Laser Communications X. Vol. 7814, pp. 781406:1-9.

7. *High data-rate optical wireless communications in passenger aircraft: Measurements and simulations.* **O'Brien, D. C., et al.** 2008. 6th Internation Symposium on Communication Systems, Networks and Digital Signal Processing, 2008. CNSDSP 2008. pp. 68-71.

8. *Free space optical sensor networking for underwater sensing applications.* **Agrawal, N., Milner, S. D. and Davis, C. C.** 2009. 2009 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). pp. 475-480.

9. *A 2.4-Ghz CMOS Short-Range Wireless-Sensor-Network Interface for Automotive Applications.* **Carmo, J. P. and Couto, C.** 5, 2010, IEEE Transactions on Industrial Electronics, Vol. 57, pp. 1764-1771.

10. *Channel modeling for FSO communications and sensor networking inside structures.* **Davis, C. C., Eslami, M. and Agrawal, N.** 2009. Proc. SPIE, Free-Space Laser Communications IX. Vol. 7464.

11. *Spot-diffusing and fly-eye receivers for indoor infrared wireless communications.* **Yun, G. and Kavehrad, M.** 1992. Proc. 1992 IEEE International Conference on Selected Topics in Wireless Communications. pp. 262-265.

12. *Performance Evaluation of Experimental 50-Mb/s Diffuse Infrared Wireless Link Using On-Off Keying with Decision-Feedback Equalization.* **Marsh, G. W. and Kahn, J. M.** 11, 1996, IEEE Transactions on Communications, Vol. 44, pp. 1496-1504.

13. *Analysis of infrared wireless links employing multibeam transmitters and imaging diversity receivers.* **Djahani, P. and Kahn, J. M.** 2000, IEEE Transactions on Communications, Vol. 48, pp. 2077-2088.

14. *A Physical Model of the Wireless Infrared Communication Channel.* **Jungnickel, V., et al.** 3, 2002, IEEE Journal on Selected Areas in Communications, Vol. 20, pp. 631-640.

15. *Receiver Designs and Channel Characterization for Multi-Spot High-Bit-Rate Wireless Infrared Communications.* **Jivkova, S. and Kavehrad, M.** 12, 2001, IEEE Transactions on Communications, Vol. 49, pp. 2145-2153.

16. *Electronic Tracking for Wireless Infrared Communications.* **Jungnickel, V., et al.** 2003, IEEE Transactions on Wireless Communications, Vol. 2, pp. 989-999.

17. *Angle diversity for nondirected wireless infrared communications.* **Carruthers, J. B. and Khan, J. M.** 2000, IEEE Transactions on Communications, Vol. 48, pp. 960-069.

18. *Free-space Optics based Sensor Network Design using Angle-diversity Photodiode Arrays.* **A. K. Ghosh, S. Kunta, P. Verma, R. C. Huck.** 2010. Proc. SPIE, Free-Space Laser Communications X. Vol. 7814.

19. *Analysis of compound parabolic concentrators and aperture averaging to mitigate fading on free space objects.* **Wasiczko, L. M., Smolyaninov, I. I. and Davis, C. C.** 2004. Proc. SPIE, Free-Space Laser Communications and Active Laser Illimuniation III. Vol. 5160, pp. 133-142.

20. **Welford, W. and Winston, R.** *High Collection Nonimaging Optics.* New York : Academic Press, 1989.

21. **Winston, R., Minano, J.C. and Benitez, P.** *Nonimaging Optics.* San Diego : Academic Press, 2005.

22. *Compound parabolic concentrators for narrowband wireless infrared receivers.* **Ho, K. and Kahn, J. M.** 1995, Optical Engineering, Vol. 34, pp. 1385-1395.

23. *Design of free space optical omnidirectional transceivers for indoor applications using non-imaging optical devices.* **Agrawal, N. and Davis, C. C.** 2008. Proc. SPIE, Free-Space Laser Communications VIII. Vol. 7091.

24. *Cellular tracked optical wireless demonstration.* **Parand, F., Faulkner, G.E. and O'Brien, D.C.** 5, 2003, IEE Proc. - Optoelectronics, Vol. 150.

25. *Single-channel imaging receiver for optical wireless communications.* **Castillo-Vazquez, M. and Puerta-Notario, A.** 2005, IEEE Communications Letters, Vol. 9, pp. 897-899.

26. *Optical Wireless Sensor Network System Using Corner Cube Retroreflectors.* **Teramoto, S. and Ohtsuki, T.** 2005, EURASIP Journal on Applied Signal Processing, Vol. 1, pp. 39-44.

27. *Design and Implementation of Optical Wireless Communications with Optically Powered Smart Dust Motes.* **O'Brien, D. C., et al.** 9, 2009, IEEE Journal on Selected Areas in Communication, Vol. 27, pp. 1646-1653.

28. *High-Speed Integrated Transceivers for Optical Wireless.* **O'Brien, D.C., et al.** 3, 2003, IEEE Communications Magazine, Vol. 41, pp. 58-62.

29. *Integrated Transceivers for Optical Wireless Communications.* **O'Brien, D.C., et al.** 1, 2005, IEEE Journal of Selected Topics in Quantum Electronics, Vol. 11, pp. 173-183.

30. *IrDA: Past, Present and Future.* **Williams, S.** 1, 2001, IEEE Personal Communications, Vol. 7, pp. 11-19.

31. *Performance Analysis of the Advanced Infrared (AIr) CSMA/CA MAC Protocol for Wireless LANs.* **Vistsas, V. and Boucouvalas, A. C.** 5, 2003, Wireless Networks, Vol. 9, pp. 495-507.

32. **Davis, Christopher C.** *Lasers and Electro-Optics.* Cambridge : Cambridge University Press, 1996.

33. *TDMA Protocol Requirements for Wireless Sensor Networks.* **Cionca, V., Newe, T. and Dadarlat, V.** 2008. The Second International Conference on Sensor Technologies and Applications. pp. 30-35.

34. *SpeckMAC: low-power decentralised MAC protocols for low data rate transmissions in specknets.* **Wong, K-J. and Arvind, D. K.** Florence, Italy : Assocication for Computing Machinery, 2006. Proc. of the 2nd international workship on Multi-hop ad hoc networks: from theory to reality. pp. 71-78.

35. *A MAC Protocol with Directional Antennas for Deafness Avoidance in Ad Hoc Networks.* **Takata, M., Bandai, M. and Watanabe, T.** 2007. IEEE Global Telecommunications Conference, GLOBECOM '07. pp. 620-625.

36. *Network Coded ALOHA for Wireless Multihop Networks.* **Lee, H-K. and Kim, S-L.** 2009. IEEE Conference WCNC 2009, Wireless Communications and Networking .

37. *Throughput and Delay Analysis of Unslotted IEEE 802.15.4.* **Latre, B., et al.** 1, 2006, Journal of Networks, Vol. 1, pp. 20-28.

38. **Toh, C. K.** *Ad Hoc Mobile Wireless Networks.* Upper Saddle River : Prentice Hall PTR, 2002.

39. **Kurose, J. F. and Ross, K. W.** *Computer Networking.* New York; Boston; San Fransisco : Addison-Wesley, 2009.

40. *A Cross-Layer Neighbor-Discovery Algorithm for Directional 60-GHz Networks.* **Yildirim, F. and Liu, H.** 8, 2009, IEEE Transactions on Vehicular Technology, Vol. 58, pp. 4598-4604.

41. *A Directional CSMA/CA Protocol for mmWave Wireless PANs.* **Gong, M. X., et al.** 2010. IEEE Wireless Communications and Networking Conference (WCNC).

42. *A Distributed Asynchronous Directional-to-Directional MAC Protocol for Wireless Ad Hoc Networks.* **Shihab, E., Cai, L. and Pan, J.** 9, 2009, IEEE Transactions on Vehicular Technology, Vol. 58, pp. 5214-5134.

43. *Directional MAC Protocol for Millimeter Wave based Wireless Personal Area Networks.* **An, X. and Hekmat, R.** 2008. IEEE Vehicular Technology Conference (VTC). pp. 1636-1640.

44. ATmega644P Automotive Datasheet. *www.atmel.com.* [Online] http://www.atmel.com/dyn/resources/prod_documents/doc7674.pdf.

45. **Streetman, B. G. and Banerjee, S.** *Solid State Electronic Devices.* Upper Saddle River : Prentice Hall Inc., 2000.

46. S6775-01 Si PIN Photodiode Datasheet. *www.hamamatsu.com.* [Online] http://sales.hamamatsu.com/assets/pdf/parts_S/s2506-02_etc_kpin1048e04.pdf.

47. AD8603: Precision Single MicroPower Rail to Rail Input/Output Low Noise CMOS Operational Amplifier. *www.analog.com.* [Online] http://www.analog.com/static/imported-files/data_sheets/AD8603_8607_8609.pdf.

48. **Pachchigar, M.** *Design Considerations for a Transimpedance Amplifier.* National Semiconductor Coorporation. 2008. Application Note 1803. http://www.national.com/nationaledge/files/national_AN-1803.pdfNHerWJU.

49. **Wang, T. and Erhman, B.** *Compensate Transimpedance Amplifiers Intuitively.* Texas Instruments. 2005. Application Report. http://www.ti.com/litv/pdf/sboa055a.

50. **Gray, P. R., et al.** *Analysis and Design of Analog Integrated Circuits (4th Edition).* New York : John Wiley & Sons, 2001.

51. **Agrawal, G. P.** *Fiber-Optic Communication Systems.* New York : John Wiley & Sons, Inc., 2002.

52. **Palais, J.C.** *Fiber Optic Communications.* New Jersey : Pearson Prentice Hall, 2004.

53. VCSEL-980 Datasheet. *www.thorlabs.com.* [Online] http://www.thorlabs.com/Thorcat/11900/11923-M01.pdf.

54. *Advaned Circuits.* [Online] http://www.4pcb.com/.

55. Manual SMT Pick and Place Machine: ezPick. *www.abacom-tech.com.* [Online] http://www.abacom-tech.com/assets/data_sheets/ezPick.pdf.

56. *Free space optical sensor network for short-range applications.* **Agrawal, N., Milner, S. D. and Davis, C. C.** 2010. Proc. SPIE, Free-Space Laser Communications X. Vol. 7814.