

UC Santa Cruz

UC Santa Cruz Electronic Theses and Dissertations

Title

Wireless Relay Communication System for Multiple Small Robots

Permalink

<https://escholarship.org/uc/item/5d85q0jh>

Author

Baumgartner, Jeremy Gordon

Publication Date

2015

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-NonCommercial License, available at <https://creativecommons.org/licenses/by-nc/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

**WIRELESS RELAY COMMUNICATION SYSTEM FOR MULTIPLE
SMALL ROBOTS**

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

JEREMY GORDON BAUMGARTNER

March 2015

The Thesis of JEREMY GORDON
BAUMGARTNER
is approved:

Professor Mircea Teodorescu, Chair

Professor Gabriel Elkaim

Professor Ricardo Sanfelice

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Copyright © by
JEREMY GORDON BAUMGARTNER
2015

Table of Contents

List of Figures	v
Abstract	vi
Dedication	vii
Acknowledgments	viii
1 Introduction	1
1.1 Overview	1
1.2 Thesis Structure	2
2 Background Information	3
2.1 Introduction	3
2.2 Small Ground-Based Robots	3
2.3 Mobile Ad-Hoc Networks	5
2.4 Relay Systems	7
2.5 Radio Propagation	9
2.6 Simulation Tools	9
2.6.1 Electromagnetic Propagation Simulation Tools	10
2.6.2 Robot Simulation Tools	11
2.6.3 Network Simulation Tools	11
3 Motivation and Goals	13
4 Modeling	15
4.1 Introduction	15
4.2 Wireless InSite	15
4.2.1 Simulation Configuration	16
4.2.2 Simulation Results	18
4.3 Webots	20
4.3.1 3D Design	21
4.3.2 Software Design	24

4.3.3	Wireless Design	26
4.3.4	Wireless Analysis	27
4.3.5	Simulation Results	29
4.4	QualNet	29
4.4.1	Simulation Configuration	30
4.4.2	Sequence Number Implementation	31
4.4.3	Simulation Results	33
5	Validation and Results	36
5.1	Introduction	36
5.2	XBee System	36
5.2.1	Lab Testing	37
5.2.2	Field Testing	39
6	Conclusion	46
7	Future Work	48
A	Appendix	50
A.1	Webots Controller Code	50
A.2	XBee Radio Configuration	59
	Bibliography	64

List of Figures

4.1	A visualization of the received transmission power of a radio for various transmitter heights.	19
4.2	A visualization of the received power of the lower-left transmitter. The upper-right radio experiences poor signal due to the line-of-sight obstruction. . . .	21
4.3	A visualization of the received power of the upper-right transmitter. The lower-left radio experiences poor signal due to the line-of-sight obstruction.	22
4.4	A visualization of the received power of two transmitters used together to establish line-of-sight with the upper-right radio.	23
4.5	The robot 3D design.	24
4.6	Functional diagram of the Webots simulation.	26
4.7	Structure of a transmitted packet in the Webots simulation.	26
4.8	Wireless link budget of a point-to-point XBee radio system.	28
4.9	Plot and regression of the link budget over distance.	28
4.10	Graph Topologies	31
4.11	RipResponse Structure	32
4.12	Overhead from RIP in Static Environment	34
4.13	Overhead from RIP in Mobile Environment	34
5.1	Functional diagram for lab testing of the XBee radio system.	37
5.2	Lab bench testing setup of the XBee radio system.	38
5.3	UAV in flight during testing. Red circle highlights the plane, which is zoomed in the inset image.	40
5.4	The fully instrumented Bixler 2 plane.	41
5.5	The fully instrumented Flyzone Calypso plane.	41
5.6	Typical set of results obtained from one UAV.	42
5.7	Both planes on the ground on opposite ends of the field. No line-of-sight. No communication link.	44
5.8	Plane 1 operating as a relay node for Plane 2. Direct line-of-sight.	45
5.9	Plane 1 is directly above the base station. No line-of-sight. No communication link.	45

Abstract

Wireless Relay Communication System for Multiple Small Robots

by

JEREMY GORDON BAUMGARTNER

Small terrestrial robots have the advantage of being inexpensive and small in size. However, this introduces notable disadvantages, including limited communication range and restricted tasking capabilities. To make up for these shortcomings, multiple robots can be utilized in a mobile ad-hoc network (MANET) in order to extend transmission range as a function of robot quantity and ultimately achieve more difficult tasks.

Three types of software models are created, which simulate the electromagnetic propagation of multiple radios, the 3D physical environment of multiple robots, and the network-level implementation of a routing protocol. A hardware implementation of a wireless relay communication system is developed to test the real world affects of such a system. Tests are conducted in a small lab setting and scaled up to field tests of multiple unmanned aerial vehicles (UAVs). The results show that a MANET in a relay topology can improve communication range and overall mission capability.

To my endlessly supportive and loving family.

Acknowledgments

I would like to thank my advisor, Mircea Teodorescu, whose early interest in my goals and constant enthusiasm for my work made my research what it is today. I would also like to thank Katia Obraczka, whose support was greatly appreciated throughout my time at UC Santa Cruz. Special thanks to Gabriel Elkaim and Ricardo Sanfelice for serving on my committee and providing support for anything related to UAVs.

I would like to thank my lab mates in the DANSER lab for their support, including my friend Sebastian Hening, who helped me greatly throughout UAV testing. I thank Corey Ippolito for the guidance and opportunities he provided regarding UAVs. And I thank Lincoln Thurlow for his expertise on networking and simulations.

I would also like to thank Daphne, my best friend and distraction during my years here and beyond.

And finally I'd like to thank my parents Bill and Vicki and my brother Zach for their loving support. I would not be here without them.

Chapter 1

Introduction

1.1 Overview

This thesis investigates the use of mobile ad-hoc networks (MANETs) to improve aspects of multi-robot systems. Models and simulations are developed in order to explore and quantify electromagnetic propagation affects, multi-robot communication and collaboration, and network-level routing of mobile nodes. Electromagnetic wave propagation is taken into account for radio line-of-sight obstacle avoidance. Relay topologies are examined in particular, in order to simplify the dynamic movement of mobile nodes and offer more focused robotic tasking. A MANET protocol is studied and modified in order to facilitate simpler and more efficient routing.

Insights gained from the models are applied to real-world scenarios. A small radio with MANET capabilities is selected and tested in a variety of environments and platforms. Relay topology networks are created in both a lab setting and an outdoor setting. Radios are installed on multiple small unmanned aerial vehicles (UAVs) in order to test long range

communication via strategically-placed relay nodes.

1.2 Thesis Structure

This thesis, “Wireless Relay Communication System for Multiple Small Robots”, has six chapters. Chapter 1 gives an introductory overview and structure of the thesis.

Chapter 2 provides the necessary background information spanning the scope of this thesis. Literature reviews on small ground-based robots, mobile ad-hoc networks, relay systems, radio propagation, and simulation tools are completed. It establishes the state-of-the-art and gives context to the points discussed later in the thesis.

Chapter 3 lays out the motivation and goals for the thesis.

Chapter 4 details the modeling accomplished. It introduces the various types of simulations used, followed by a detailing of the specific tools and results. This involves Wireless InSite, Webots, and QualNet.

Chapter 5 explores the real life validation of the models and results therein. It begins with an introduction, followed by the implementation of a specific radio system. This covers two phases of testing: lab testing and field testing.

Chapter 6 concludes the thesis by revisiting the goals and summarizing the results.

Chapter 7 discusses future work for an extended application.

Chapter 2

Background Information

2.1 Introduction

A literature review was conducted to provide adequate background on the multiple fields represented by this research. Several key topics were investigated: small ground-based robots, mobile ad-hoc networks, relay systems, radio propagation, and simulation tools.

2.2 Small Ground-Based Robots

The current state-of-the-art of small ground-based robots spans across low-cost consumer products and expensive government-funded robotics. Consumer robotics targets a lower price point, so innovations in the area are often made for cost reduction. Research robotics is low volume and does not necessarily require a business case, so advances can be made in expensive cutting-edge technologies. Government-funded robotics can involve large contracts, also allowing for state-of-the-art development.

Companies have been making strong business cases for consumer robotics. Small robots are becoming more accessible to average consumers as technologies improve and cheapen. Neato Robotics and iRobot Corporation have had success with their lines of autonomous home cleaning robots, including the XV and Roomba series. These robots feature advanced autonomous mapping and navigation for less than 500 dollars. Suitable Technologies and Double Robotics develop telepresence robots for home or office environments. They feature advanced mobility and communication systems. Willow Garage is a robotics research lab that develops hardware and software for personal robotics. They developed and maintain several open-source software tools that are widely used, including the Robot Operating System (ROS) and the OpenCV computer vision library.[28] These companies are a sample of the growing small-robot consumer market. The industry will grow as technology improves, utility increases, cost decreases, and more companies enter.

Research in small robots at the university level has been strong for decades. Specific research topics include localization, navigation, locomotion, power usage, computer vision, wireless communication, cooperation, and task planning. Many institutions are making large impacts in the field, including Stanford University, MIT, Carnegie Mellon University, University of Michigan, and Georgia Institute of Technology. The current specific projects are too numerous to describe individually.

Government-funded robots are built for an array of applications, such as warfare or space exploration. The US government establishes contracts with companies to develop robots for specific purposes. One example is the iRobot 510 PackBot [38], which was funded by a Small Business Innovative Research (SBIR) grant to help soldiers in dangerous

missions. The PackBot serves as a platform for research and development that supports a wide range of payloads.

NASA has also produced many purpose-built rovers for missions exploring the Moon and Mars, such as the Apollo Lunar Roving Vehicle, Sojourner, Spirit, Opportunity, and Curiosity. Russia and China have also developed rovers, including Lunokhod 1 and Chang'e 3. These rovers vary in size from the size of a microwave to the the size of a car.

2.3 Mobile Ad-Hoc Networks

Mobile ad-hoc networks (MANET) are networks that are infrastructure-less, wireless, and consist of multiple mobile nodes. These nodes exist in topologies that are multi-hop and dynamic.[30]

MANET protocols can be divided into two categories: distance-vector routing protocols and link-state routing protocols. Distance-vector protocols periodically broadcast their complete routing tables to their immediate neighbors, leading to a propagation of updates throughout the network. The Bellman-Ford algorithm can then be used to determine the best path between any two points. Common protocols include Routing Information Protocol (RIP)[17], Destination-Sequenced Distance Vector (DSDV)[25], and Ad-hoc On-Demand Distance Vector (AODV)[24]. Link-state routing protocols have each node share information about itself and its immediate links. This is propagated throughout the network and copied by each router, thus giving each node an identical map of the network. Each node can then calculate their best paths independently. Common ones include Open Shortest Path First (OSPF)[20], Intermediate System to Intermediate System (IS-IS)[23], and

Optimized Link State Routing (OLSR)[13]. This thesis focuses on distance-vector routing protocols, specifically RIP and AODV.

RIP is an older routing protocol with disadvantages to modern protocols. However, it has the advantage of being relatively simple and easily modified. RIP employs hop counts as a metric to determine paths through a network. Disadvantages of RIP include subpar looping properties, high message overhead, poor scalability, and slow convergence time. The inherent properties of the basic Bellman-Ford algorithm are to blame for the poor looping properties, so RIP uses a technique called split horizon with poison reverse to help eliminate looping issues. RIP has gone through multiple revisions, such that there is RIPv1, RIPv2, RIPng.

AODV is a more modern MANET that improves upon RIP in many ways. The main attribute of AODV is that the network remains silent until a connection needs to be established (hence “on demand”). When a connection needs to be made to the desired node, a request is broadcasted. Nodes throughout the network forward the request message. Each node records the node they receive the request from. When the desired node is reached, a message is returned back to the source node, via the established route. This route is used as the shortest communication path until it fails or gets recycled.

Research in MANETs spans many applications, especially within the field of robotics. Such applications include search and rescue, hazardous situations, military, urban environments, planetary exploration, and any other situations where multiple mobile robots can achieve collaborative tasks.[21] Possible benefits include rapid deployment, increased mobility, increased communication range, higher redundancy, and unique task

functionality.[26] However, these benefits come at the cost of increased complexity, bandwidth overhead, and power consumption. This has slowed the research and development of such systems.[5]

There are many MANET protocols in use today, but each has particular shortcomings that must be taken into consideration.[1] No existing protocol offers the best of all worlds[29]. Research in this field is attempting to develop a highly capable, multi-functional MANET. Until then, design choices must be made based on the specific application and allowable trade-offs.

2.4 Relay Systems

Rapid changes in network topology (i.e. mobile nodes disconnecting and reconnecting with others) can lead to routing failures and looping. Many protocols have solved these issues, but at the cost of increased message overhead since updated routing tables must continuously flood through the network. This reduces the bandwidth available for data transmission. Large, dynamic MANETs that require fast data transfers push the limits of any MANET protocol.[30]

Applying the concepts of MANETs towards simplified or bounded topologies allows researchers to more easily solve problems in targeted applications. One such simplified topology is a relay system, where the nodes are lined up in a queue and forward messages to the next-in-line.[21] This simplifies the challenges found in many MANETs, since routing decisions are based on predictable scenarios that are unlikely to change dramatically. A relay system has many benefits over a highly dynamic topology: decreased control message

overhead, increased data transmission bandwidth, increased overall transmission distance, system redundancy, and general simplicity.

Examples of such work include the relay node-dropping tactical robots researched by at the Space and Naval Warfare Systems Center (SPAWAR).[26] A similar system is the Scout robot with relay capabilities researched at the University of Minnesota.[7] Another system is the Ad Hoc UAV Ground Network (AUGNet) from University of Colorado, Boulder.[3] Army Communications Electronics Command (CECOM) did early work in building an airborne relay system using UAVs.[27] Harvard University did research in relay communication systems for moving vehicles on highways using store and forward techniques.[4]

One inspiration for this thesis is the work done on the relay node-dropping robots by Pezeshkian and Nguyen.[26] The research in this thesis changes the architecture by using small robots or UAVs as nodes, rather than static nodes dropped from behind a larger robot. This is advantageous because the relay robots can also participate in overall tasks, such as reconnaissance or sample collection. The Scout robot [7] is also similar in architecture, but the work focuses more on the physical design of the robot, including locomotion improvements and grappling hook design. This thesis places more emphasis on the MANET and electromagnetic propagation analysis. The AUGNet system [3] is advanced, coordinating multiple robots and UAVs in a large MANET. The network improves the communication range of any of the nodes. This thesis implements smaller, homogeneous nodes in order to improve mobility and decrease complexity of the system.

2.5 Radio Propagation

Radio waves, a form of electromagnetic radiation, exhibit many effects as they propagate through a medium. The scope of this thesis covers the ultra high frequency (UHF) band, which is between the frequencies of 300 MHz and 3 GHz. At these frequencies, radio waves propagate primarily by line-of-sight. Thus large buildings or geographical features, such as mountains, can completely block transmissions. Transmissions can also be attenuated by any physical objects or atmospheric phenomena.[18]

However, line-of-sight is more than just a straight line from radio to radio. More realistically, it is an elliptical envelope from radio to radio, called the Fresnel zone.[2] In order for transmissions to propagate, no more than 40% of the Fresnel zone can be obstructed. This becomes problematic with small robots that have short radios (that is, antennas that are close to the ground).[10] At such low heights, the ground itself becomes a major obstruction for the Fresnel zones. Thus, short robots have limited communication ranges.[37] Measures can be taken to counter the affects of Fresnel zones on small robot wireless communication. Antenna height can be increased, but this can be physically difficult and limit mobility. Transmit power can be increased, but this puts a burden on the low-power systems of a small robot. This thesis focuses on bringing nodes closer together and utilizing relay nodes.

2.6 Simulation Tools

Software simulators are an important tool to expedite initial design and limit obstacles that might arise later in the development cycle. There are tools that simulate

computer networks, electromagnetic propagation, three-dimensional physical interaction, and much more. They allow the user to catch unforeseen affects early in the development cycle and help avoid expensive and time-consuming hardware implementations. It is important to identify these tools and understand how to leverage them to achieve the desired goal. There is also no one tool that simulates everything at once, although research-based solutions exist [15][33], so the tools must be utilized individually.

2.6.1 Electromagnetic Propagation Simulation Tools

To understand the effects and limitations of using multiple radios in certain environments, it is necessary to implement electromagnetic (EM) propagation simulations. Such tools make use of ray tracing propagation models to quantify radio transmissions.

One tool is Wireless InSite by Remcom, a popular EM propagation simulator with many useful features and support for students. It offers complete control to characterize multiple radio transceivers and place them within complicated natural, urban, and indoor environments. There are propagation models available, including full 3D. The full 3D model is very complete (and naturally takes Fresnel zones into account).[39]

Other tools include WinProp by AWE Communications and Terrain Analysis Package (TAP) by Softwright.[14] These software packages have similar features to Wireless InSite, but are more targeted toward industrial applications, such as the installation of large cell towers and other larger scale projects. Therefore the software is less suitable the smaller physical scale of this research.

2.6.2 Robot Simulation Tools

Robot simulators can provide insight regarding the interaction of between the environment, objects, and other robots. These interactions can include physical collisions, object sensing, and wireless communications.

One popular robot simulator is called Webots.[19] Within the simulation, robots have a detailed physical design, some method of mobility, many choices of sensors, and user-defined software control code. Multiple robots can intersect with each other within the same environment. Webots was used in this thesis.

Another popular simulator is Stage, a 2D multi-robot simulator that is part of the open-source Player Project.[9] The tool is used to create interactions between hundreds of robots in a single environment, complete with sensors and control programs.

Gazebo is also part of the Player Project, and the 3D alternative to Stage. It allows for complete 3D design and integrates a more powerful physics engine. Like other components of the Player Project, it can have a high learning curve.

2.6.3 Network Simulation Tools

Many tools exist for simulating computer networks, including simulators with mobile ad-hoc network support. Popular network simulators include ns-2, QualNet, and OM-Net++. They have varying levels of complexity, ease-of-use, completeness, availability, and support.

The most popular tool is ns-2 [6], a discrete event simulator that was created to simulate wired networks. Wireless network support was added via extensions to the core

program. ns-2 is complex and has limited modularity, making it difficult to use. It also has high resource usage and poor scalability in large networks. However, it is open-source and widely supported and documented, making it a popular choice.

QualNet is a commercial discrete event simulator based on the GloMoSim core [40], a popular wireless network simulator developed at UCLA. QualNet provides a more cohesive experience, good documentation, and a wide selection of protocols. It is also parallelized, allowing it to run relatively fast. QualNet was used in this thesis.

OMNet++ [36] is a network simulation library and framework used to build network simulators. It is highly modular and has wireless and mobility extensions that add MANET capabilities.

Chapter 3

Motivation and Goals

A small robot offers special advantages, including low cost and small size. This however introduces significant disadvantages, such as limited communication range and restricted mission capabilities. To address these issues, multiple small robots can be employed to communicate and accomplish tasks collaboratively.

Mobile ad-hoc networks can improve communication distances by relaying transmissions across multiple robots to reach the furthest nodes, increasing communication range as a function of robot quantity. Multiple small robots can then operate cooperatively to achieve unique tasks that one large robot may not achieve. This strength in numbers has the added benefit of system redundancy. If an individual robot stops functioning, the impact on the overall system can be relatively minimal. Furthermore, the low cost of such robots would allow for greater production quantity and the small size would provide greater transportation convenience.

The goal of this thesis is to research the affects of multiple mobile wireless nodes in

a relay communication system and apply the insights by testing real life implementations. Simulations will be completed from multiple perspectives of the system. The models and protocols will be modified to improve performance. And a physical realization of a system will be developed in order to validate the work. The work will show that a relay communication system can be leveraged to overcome the disadvantages of small robots, such that communication distance is increased and mission tasks are accomplished.

Chapter 4

Modeling

4.1 Introduction

With advances in software simulation tools, the design, implementation, and testing of robotic systems can be accomplished quickly, without the overhead of traditional hardware iterations. Three simulation tools were chosen to pursue the goals of this thesis. Each tool serves a specific purpose, since no one tool simulates every facet of an entire multi-robot system. Wireless InSite, to simulate the communication system on an electromagnetic level; Webots, to simulate the robots on a system level; and QualNet, to simulate the communication system on a networking level.

4.2 Wireless InSite

Simulations are carried out in Wireless InSite in order to better understand the expected complex electromagnetic propagation effects of multiple radios. The software

incorporates the characteristics of the radio, the presence of physical obstacles, and complicated electromagnetic transmission effects. The simulations are meant to provide an early look at the affects that might not arise in the Webots and QualNet simulations, as well as any affects that might be unexpected in general.

Two key simulations are carried out. The first simulation contains a radio placed on an infinite flat plane. The purpose of this simulation is to explore the affects of Fresnel zones and how the height of the radio transmitter with respect to the ground plane affects the maximum transmission distance. The second simulation contains multiple radios placed within an L-shaped office building hallway. The purpose of this simulation is to explore how line-of-sight obstacles and enclosed spaces affect radio transmissions.

4.2.1 Simulation Configuration

A complete simulation in Wireless InSite contains many attributes, including physical features, transmitters, receivers, communication systems, antennas, waveforms, and propagation models.

For all Wireless InSite simulations, the design is approached with small robots and XBee radios in mind (for more information on the radios, see Section 5.2.1). The antennas are vertically polarized and the carrier frequency is set to a 2.4 GHz sinusoid. The transmitters are generally placed at a height of 0.25 meters from the ground, since that is an appropriate height for the type of small robot explored in this thesis. The receivers are placed at the same height as the transmitters in a grid pattern dense enough to capture the detailed constructive and destructive areas of the propagation patterns. The term “receiver” in this case refers to the point at which the propagation model calculates a value

for the power received. Denser grids have more receivers, thus increasing the resolution of the results (while increasing computation time).

Wireless InSite offers many propagation models. The Full 3D model is the most accurate and complete. It functions by propagating rays throughout the environment, taking into account transmissions, reflections, and diffractions in the electromagnetic field. Physical features in the environment can have complicated geometries and transmissions are allowed to propagate through surfaces.

The ray tracing method used in the Full 3D model is called the Shooting and Bouncing Ray (SBR) method. SBR traces rays throughout the three-dimensional geometry, starting from the source points and propagating outwards in multiple directions. Rays reflect off objects specularly, and continue to reflect off each subsequent object, until the maximum number of allowable reflections is reached. Diffractions are also taken into account by examining a discontinuous ray neighboring a continuous ray. The SBR procedure is more complete described in [31] and [32]. After defining the rays, Full 3D implements an electric field evaluation defined in [12] and [16].

For the first simulation, a flat 4 km square plane is created to model the ground. The radio transmitter is placed at the center of the plane at a height of 0.25 meters. The receiver grid is also placed at a height of 0.25 meters, since it is assumed that the transmitting robot would be at the same height as a receiving robot. The grid density is set to 30 meters between nodes, which provides adequate resolution for the visualization. The simulation is run and the results are collected. The transmitter and receiver grid are then moved to a height of 0.1 meters and simulated again. Then again for 0.5 meters and

1 meter.

For the second simulation, a standard office building environment was imported as the physical environment. An indoor L-shaped hallway was selected as an area of study, since it introduces an obstruction in the line-of-sight. The walls, floors, and ceiling are constructed of reinforced concrete. A transmitter is placed at one end of the hallway at a height of 0.25 meters. A second transmitter is placed at the opposite point the hallway, out of the line-of-sight of the first transmitter. A third transmitter is placed at the corner of the L-shape such that it is within line-of-sight for the other two transmitters. The receiver grid is placed at a height of 0.25 meters and covers the entire area of the hallway. The simulation is run and the results are collected.

4.2.2 Simulation Results

The figures in this section are visualizations created after running the Wireless InSite Full 3D propagation model in the desired environment and configuration. The color scales represent the magnitude of the received transmission power (red is strongest and violet is weakest).

Figure 4.1 shows the results of the first set of simulations, where a single transmitter was placed on a flat plane with varying heights. Four different transmitter heights are shown: 0.1, 0.25, 0.5, and 1 meter. The color scale represents the received power at each individual point across the plane, measured at the same height as the transmitter height. The plane in each of the four images is 4 km by 4 km.

It is clear that increasing the height of the transmitter significantly increases transmission distance. If a small robot is estimated to have a 0.25 meter transmitter, and a large

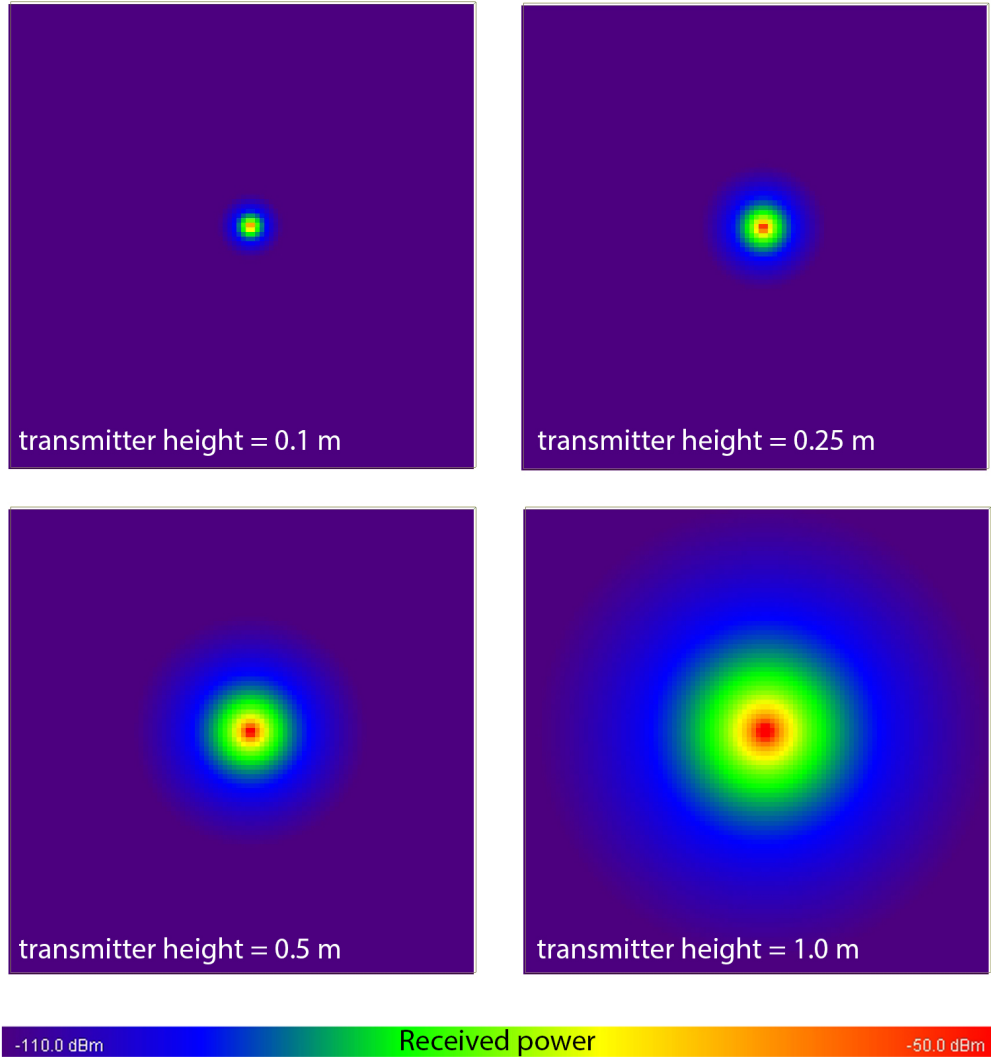


Figure 4.1: A visualization of the received transmission power of a radio for various transmitter heights.

robot has a 1 meter transmitter, then the difference in communication distance between the two is drastic.

Figure 4.2, Figure 4.3, and Figure 4.4 show the results of the second set of simulations, where multiple transmitters were placed in an L-shaped office hallway with line-of-

sight obstructions. Figure 4.2 shows the received power of the lower-left transmitter. It is evident that the upper-right radio experiences poor signal due to the line-of-sight obstruction. Figure 4.3 shows a different perspective with the received power of the upper-right transmitter. It is also evident that the lower-left radio experiences poor signal due to the line-of-sight obstruction. Figure 4.4 makes use of a third transmitter to relay transmissions from the lower-left transmitter to the upper-right radio. This significantly improves the signal of the upper-right radio, and serves as confirmation that using multiple radios to bridge line-of-sight losses is beneficial.

Another interesting observation is the pattern of destructive interference, where out-of-phase waves cancel each other out, creating significant signal holes. It is evident that the hallway environment is both good and bad. The enclosed space allows signal vectors to reflect off of the walls, floor, and ceiling to eventually reach the radio, providing improved signal. However, the destructive interference of the varying reflections still creates areas of poor signal.

4.3 Webots

Webots is a professional robot simulator that can take into account many properties of a multi-robot system. Rigid bodies of the robots can be designed and simulated in a 3D space. Individual robots can be characterized with radios and sensors and multiple robots can interact within the simulation. Webots provides a software integrated development environment, making it ideal for the rapid prototyping of control software.

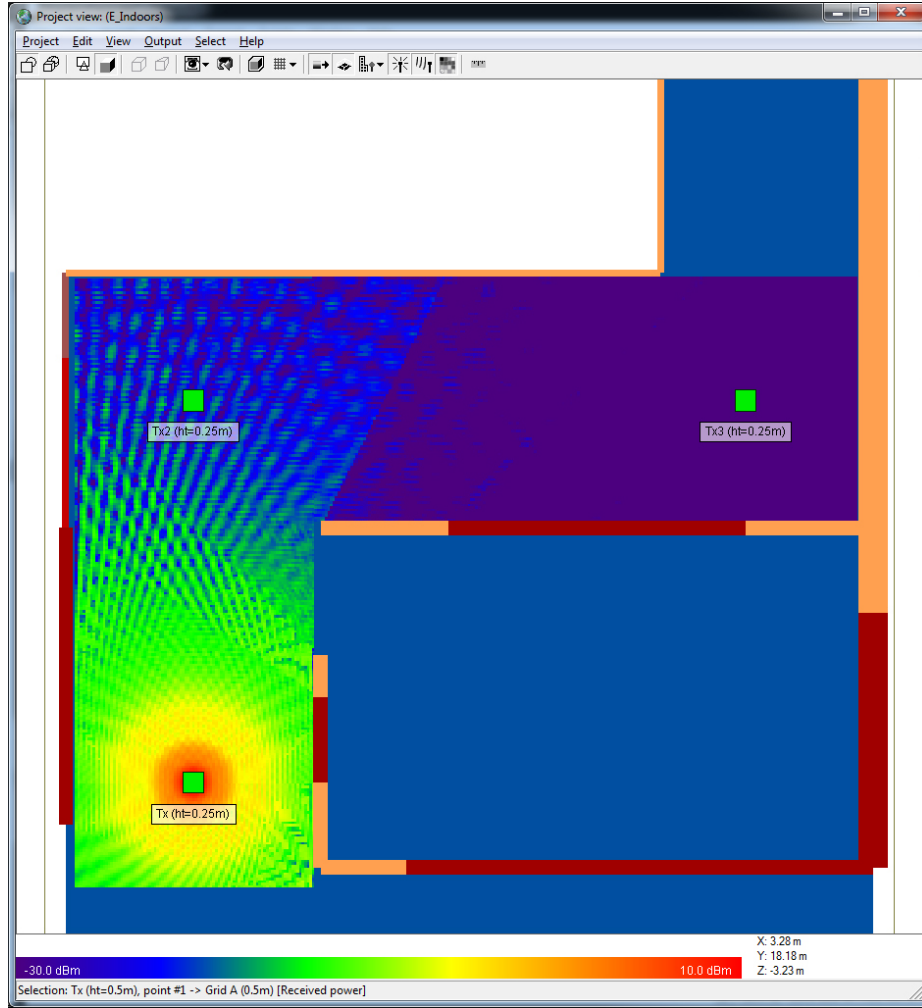


Figure 4.2: A visualization of the received power of the lower-left transmitter. The upper-right radio experiences poor signal due to the line-of-sight obstruction.

4.3.1 3D Design

Since this research focuses on the wireless communication between the robots, rather than the physical robots themselves, a simplified CAD was developed to satisfy only the key components of the system. The robot body is a cylinder 8 centimeters tall and 9 centimeters in diameter, with the flat sides parallel to the ground plane. A pair of differential

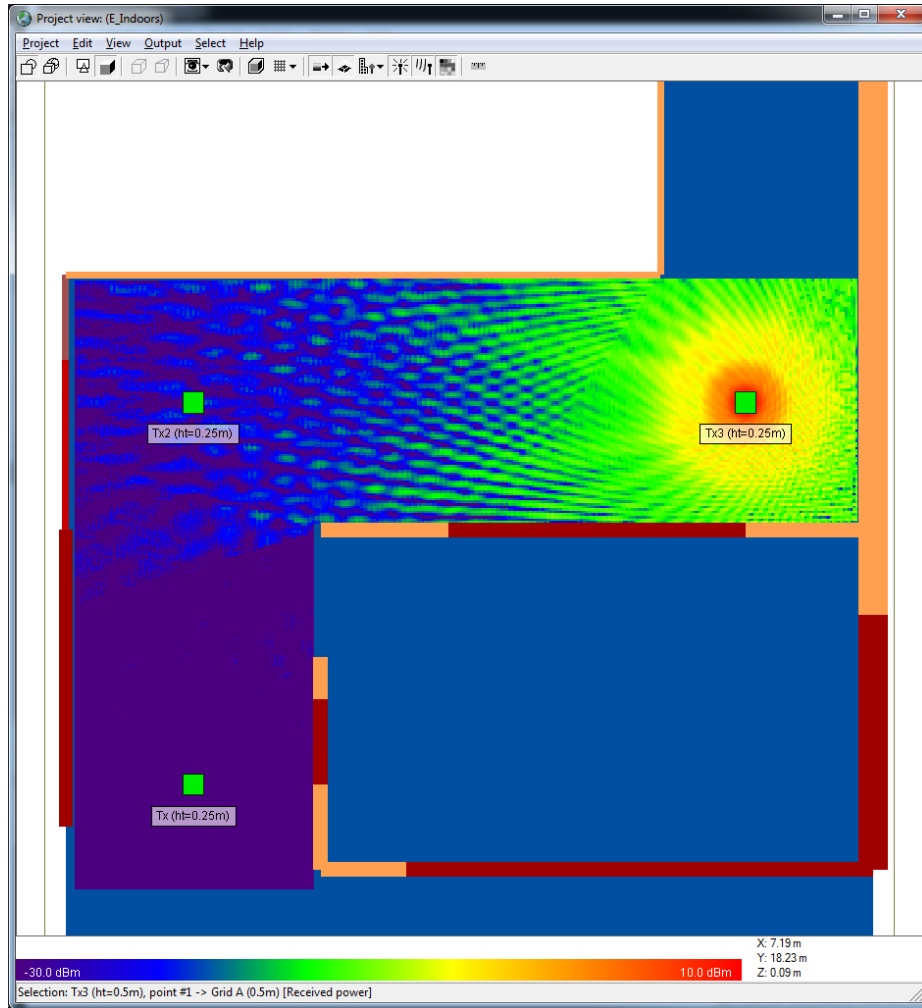


Figure 4.3: A visualization of the received power of the upper-right transmitter. The lower-left radio experiences poor signal due to the line-of-sight obstruction.

wheels 5 centimeters in diameter are mounted on the sides of the body. On the flat top of the body are two radio antennas (one for transmit, one for receive). These combined elements result in a stable robot with the features needed for a wireless communication simulation. The 3D mock-up of the robots can be seen in Figure 4.5. The simulation environment contains three robots: one Coordinator, one Router, and one End Device.

There is also a physical environment in which the robots operate, called a world.

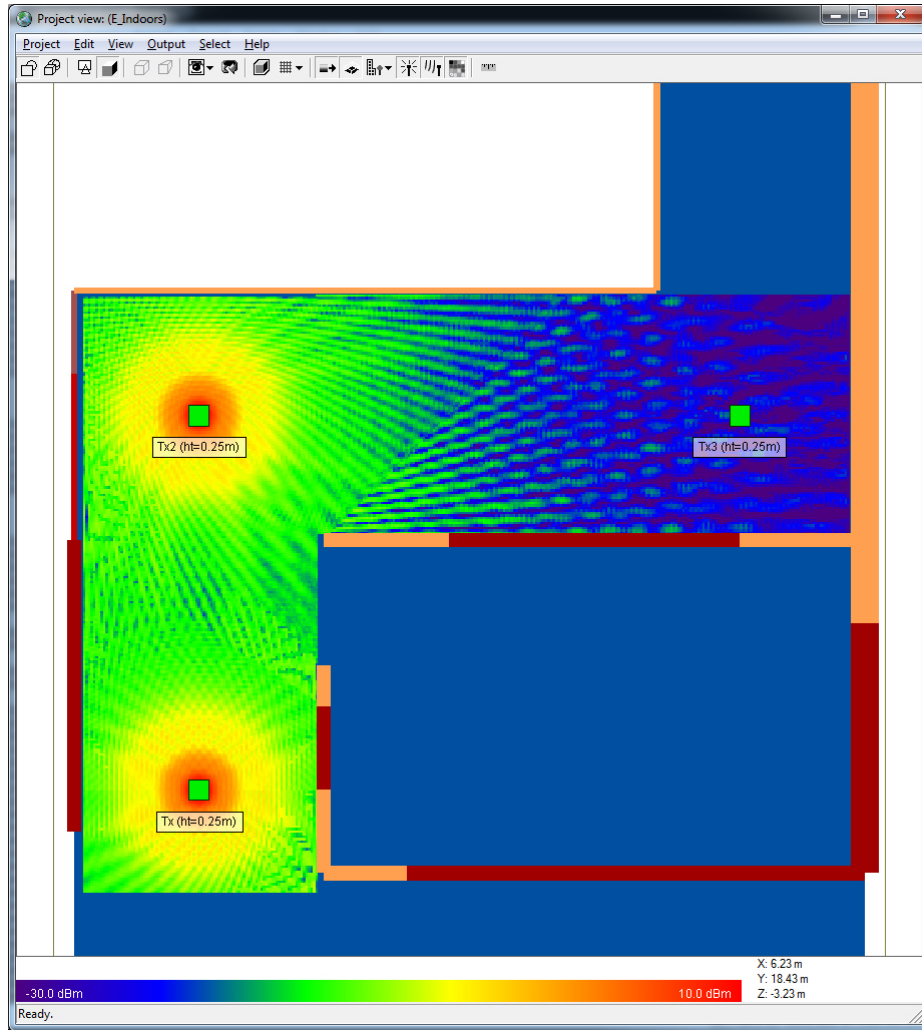


Figure 4.4: A visualization of the received power of two transmitters used together to establish line-of-sight with the upper-right radio.

A flat checkerboard plane, approximately 26 meters by 13 meters, represents the ground. Various solids sitting on this plane represent obstacles in the world, which can hinder both mobility and wireless communication.

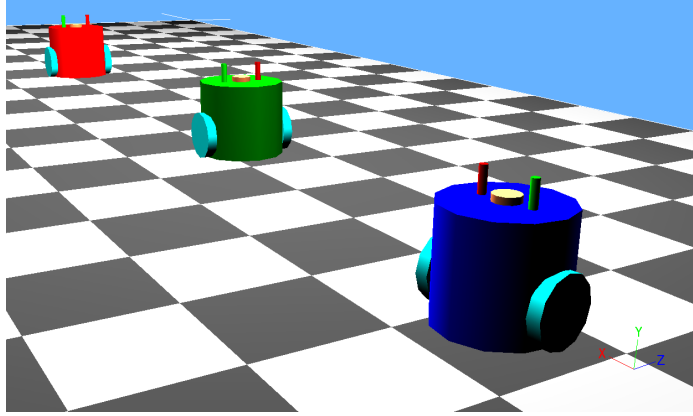


Figure 4.5: The robot 3D design.

4.3.2 Software Design

For each robot physical model there is a corresponding controller program (Figure 4.6). This is programmed in C and includes several Webots libraries along with the standard C libraries. The controller program initializes the robots, sensors, and motors. Control loops then read various sensor data, control the output to the wheels, and transmit packets between robots. Basic Webots controller code for each individual robot type can be found in Appendix A.1.

Algorithm 1 shows the pseudocode for each robot’s controller program. During each iteration of the controller’s main loop, several actions occur. First the robot checks the receiver device buffer to obtain any incoming packets. If there are packets waiting, then the controller decodes and stores the contents. Based on the received message, the controller can take action, such as forwarding the packet to the next node or carrying out a command (ex. turn left). Packets have a well-defined structure, as shown in Figure 4.7.

An omnipresent program called the supervisor can also be programmed to control

Algorithm 1 Robot controller program

```
1: initialize robot, devices, variables

2: loop

3:   if packet received then

4:     read and store packet

5:   end if

6:   decide action

7:   do action

8:   create packet

9:   transmit packet to next hop

10:  compute motor speeds

11:  set motor speeds

12: end loop
```

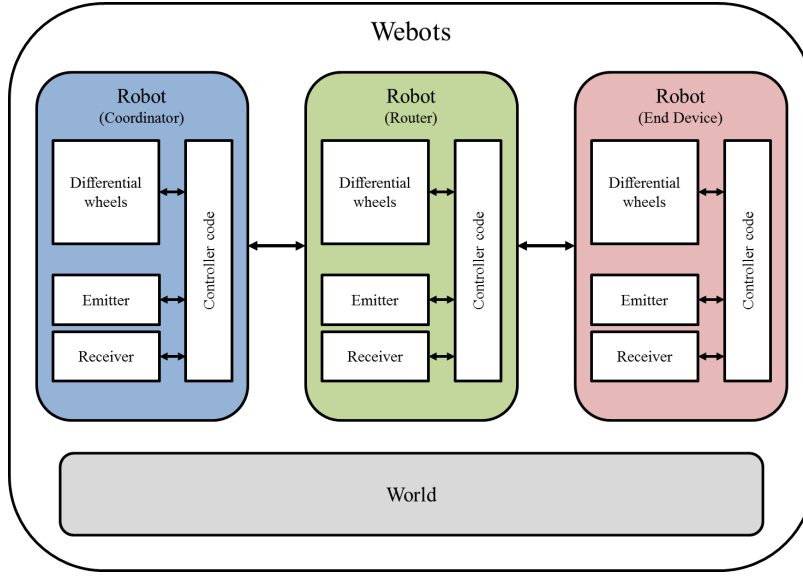


Figure 4.6: Functional diagram of the Webots simulation.

Start delimiter	Length		Frame Data	Checksum
Byte 1	Byte 2	Byte 3	Byte 4 ... Byte N	Byte N+1
0x7E	MSB	LSB	API-specific structure	Single byte

Figure 4.7: Structure of a transmitted packet in the Webots simulation.

every facet of the system. It is up to the programmer how much control the supervisor is given. In this simulation the supervisor took no role, since a goal of this system was to have multiple robots operating in a long-distance queue, without a shared connection to a master node.

4.3.3 Wireless Design

The wireless communication system is another aspect of the simulation. Each robot has two antennas, one for transmitting and one for receiving. When a message is generated by the Coordinator robot, it is transmitted to the Router robot. The Router

robot then forwards this message to the End robot. The End robot can then send a reply message back to the Coordinator via the Router. All communication is bidirectional.

Webots models signal strength with the inverse square law. Indeed, Webots calculates the signal strength between two transmitters with equation (4.1), where d is the point-to-point distance between the two transmitters.

$$\text{signal strength} = 1/d^2 \tag{4.1}$$

Webots handles line-of-sight obstacles in two trivial ways: the signal remains the same, or the signal is completely lost. In reality, the signal would degrade but not necessarily disappear in the presence of obstacles. Fresnel zones are also not taken into account, since transmissions in Webots are merely vectors from transmitter to receiver. Thus, Webots is not capable of simulating real-life wireless signal strength in a world of obstacles. This is why the Wireless InSite simulation described previously is important.

4.3.4 Wireless Analysis

Webots's wireless signal strength calculation was found to be a simple inverse square calculation, which is simply a ratio with no quantified values. Thus in order to obtain values closer to real life, it is necessary to analyze the link budget of a desired radio system and apply those values to the trivial Webots calculation.

The chosen system consists of three XBee-PRO S2B radios made by Digi, which operate at 2.4Ghz and have a 63mW output power. The link budget for a point-to-point XBee wireless system (under ideal conditions) is shown in Figure 4.8. The free space path

loss (FSPL) is given by equation (4.2), and assumes an ideal system.

Transmit Power	18 dBm	Xbee-PRO 63mW Series 2B:	Antenna:
Tx Antenna Gain	2 dBi	Frequency	2.4 GHz
Rx Antenna Gain	2 dBi	Tx Power	18 dBm
		Receiver Sensitivity	-102 dBm
Total Gain	22 dBm		
Free Space Path Loss (FSPL)	100.0542 dB	FSPL:	
Expected Rx'd Signal Level	-78.0542 dBm	distance	1 km
		frequency	2.4 GHz
Receiver Sensitivity	-102 dBm	constant	92.45
Antenna Cable Loss	0		
Link Margin	23.94578 dBm		

Figure 4.8: Wireless link budget of a point-to-point XBee radio system.

$$\text{FSPL(dB)} = 10 \log_{10} \left(\left(\frac{4\pi}{c} df \right)^2 \right) \quad (4.2)$$

By varying the distance value in the link budget, a plot of distance versus link margin is created (Figure 4.9). Performing a power regression on the data gives a line of best fit of $y = 0.248/x^2$. This constant can then be applied to Webots simulations to give absolute ideal quantities for inverse square wireless signal calculations.

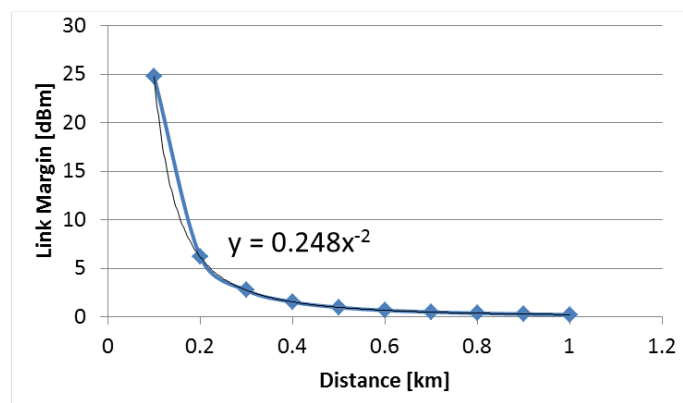


Figure 4.9: Plot and regression of the link budget over distance.

4.3.5 Simulation Results

The Webots simulation of a simple mobile ad-hoc network in Webots yields promising results. A network of transmitters attached to small rovers is established, and message packets are generated, transmitted, read, stored, and forwarded.

Webots's treatment of wireless signal propagation is subpar since it is purely line-of-sight. It is moderately improved by analyzing the ideal link budget of a desired radio and applying the resulting scalar to the trivial inverse square calculation that Webots provides. Ideally, Webots would be able to integrate an electromagnetic propagation model into the robot simulation environment, but that is likely outside the scope of the tool. Thus the simulation capability of Wireless InSite remains important.

The simple communication protocol used in the robot controller code is far from the complexity of a full MANET protocol, such as the AODV protocol used in the XBee radios. This Webots simulation used simple store and forward techniques, rather than routing tables, path algorithms, or other common MANET features. Therefore the following QualNet simulation becomes useful.

4.4 QualNet

QualNet is selected as the simulator of choice due to its completeness, documentation, and ready availability. Using the QualNet models, a modified version of RIPv2 is implemented, which will be referred to as Modified RIP. The goal of Modified RIP is to create a protocol with decreased traffic overhead, with minimal impact on protocol robustness.

RIPv2 has certain advantages that make it well-suited for simulation and modification. It is relatively simple, so modifications to the protocol can be more easily approached. The QualNet RIPv2 model is also well-documented, so issues can be more quickly corrected.

RIPv2 implements a key feature called split horizon with poison reverse. The purpose of this is to prevent looping within the network, but at the cost of protocol complexity. Thus to simplify RIPv2, Modified RIP first removes split horizon with poison reverse. To correct for the loss of these features, a technique using sequence numbers is implemented. This makes Modified RIP simpler while maintaining the overall functionality of RIPv2.

4.4.1 Simulation Configuration

QualNet is a recognized standard for networking simulation. It is a discrete event simulator, meaning that events occur at discrete moments in time. Events are scheduled at the packet level (as opposed to flow level), such that an event may be scheduled when a packet is encoded on the wire, decoded, processed, etc. The tool also comes with a fully functional TCP/IP stack that requires no modifications to run. QualNet can also implement node mobility, so that nodes move throughout the network. This makes it well-suited for MANET simulations.

The simulation uses QualNet 5.2 running on a 4-core 2.6GHz AMD processor and 12GB of RAM. Default simulations are provided for both a static topology (Figure 4.10a) and a mobile topology (Figure 4.10b).

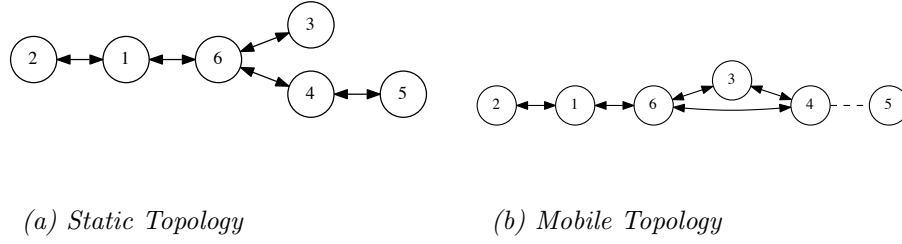


Figure 4.10: Graph Topologies

4.4.2 Sequence Number Implementation

To implement sequence numbers, two QualNet files were modified: `routing_rip.cpp` and `routing_rip.h`. For the key simplification of the protocol, split horizon is disabled in the code. This will also show that the sequence number implementation does not require split horizon to be functional.

The structure of the nodes is modified by adding a sequence number for each node transmission and a map $\langle destination, sequence\ number \rangle$. The `RipResponse` structure is also modified (Figure 4.11) to include a sequence number, to allow the passing of messages between RIP nodes.

Algorithm 2 shows the algorithm for sending sequence numbers. When a node transmits a RIP update, it increments the value of its own sequence number and inserts that value into the `RipResponse` message to be flooded to other nodes on the network.

Algorithm 3 shows the algorithm for receiving sequence numbers. When a node receives a new RIP message, the node compares the new sequence number presented in the `RipResponse` message with the existing value stored in the map. If the value of the new sequence number is greater than the existing one, then it processes the request and


```

typedef struct {
    unsigned char command;
    unsigned char version;
    short mustBeZero;
    long seq_num; //seq num
    RipRte rtes[25]; //route tables
    MPR msg;
    char* me; //IP sender
} RipResponse;

```

Figure 4.11: RipResponse Structure

overwrites the map's outdated sequence number. If the new sequence number is the same value as the existing one, then it checks the metric field of the RipRte structure for each entry in the routing table. If any metrics in the message are smaller than what is currently in the routing table, then it updates the routing table. Otherwise, no changes are made. If the new sequence number is smaller than the existing one, this message is out of date and the message is discarded without forwarding. This is a very similar approach to the one DSDV takes regarding sequence numbers.

Algorithm 2 Sequence Number Sending

```

1: for each RIP message do
2:   if message is a FULL UPDATE then
3:     increment Sequence Number
4:   end if
5: end for

```

Algorithm 3 Sequence Number Receiving

```
1: for each RIP message do

2:   if entry exists from sender then

3:     if this_seqnum > last_seqnum then

4:       accept update, broadcast to all neighbors

5:     else if this_seqnum == last_seqnum then

6:       for each entry in message's route table do

7:         if message metric > current metric then

8:           update routing table entry

9:         else

10:          do nothing

11:        end if

12:      end for

13:    else

14:      Outdated packet, throw away, do not propagate.

15:    end if

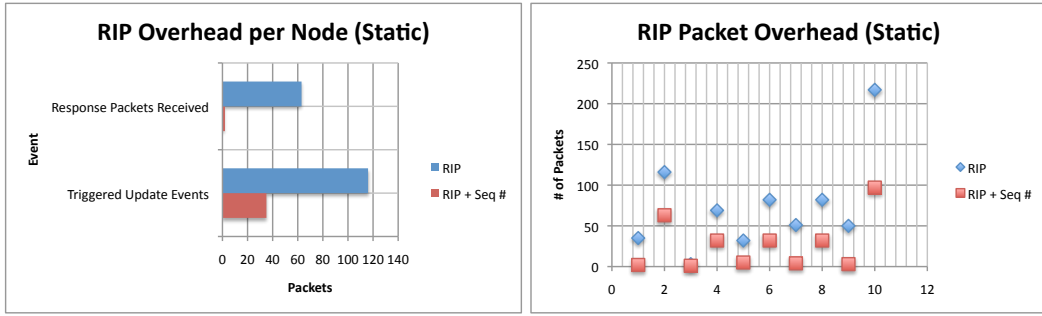
16:  end if

17: end for
```

4.4.3 Simulation Results

Two simulations are completed: one with a static topology and one with a mobile topology. The mobile topology was set to 5% mobility, which is comparable to the simple-topology of the slow-moving small robots in mind for this thesis.

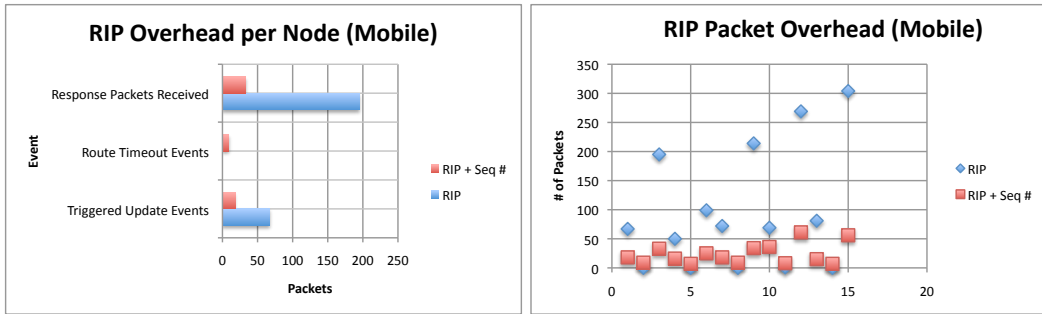
Throughout the simulations, Modified RIP did not allow any loops to form. This shows the success of the sequence number implementation. Furthermore, Modified RIP reduced overhead by 60% in the static topology and by 75% in the mobile topology. This is evident in Figures 4.13 and 4.12. This bodes well for the usefulness of Modified RIP. The reduced overhead makes the protocol very efficient for these types of topologies.



(a) Static node

(b) Static RIP

Figure 4.12: Overhead from RIP in Static Environment



(a) Mobile node

(b) Mobile RIP

Figure 4.13: Overhead from RIP in Mobile Environment

The sequence number method does have a drawback. Throwing out old messages caused RIP routes to timeout where they previously would not (Figure 4.13a). This may cause highly mobile networks to take longer to converge when using sequence numbers

versus split horizon. However, the relatively low mobility of a relay topology does not make this as big of an issue.

Chapter 5

Validation and Results

5.1 Introduction

Software simulations of theoretical wireless networks cannot fully capture the goals of this thesis. In order to form a complete picture of the system, a hardware implementation must be accomplished. This will test the real-life application of a MANET within varying environments.

5.2 XBee System

XBee radios were selected for this project. They are characterized by their small size, low power usage, and low cost. Notably, multiple XBees can be configured to run as a mobile ad-hoc network. The following sections describe the various tests and results completed using these radios.

5.2.1 Lab Testing

The system consists of three XBee-PRO S2B radios made by Digi, which operate at 2.4Ghz, have a 63mW output power, and a 1 mile range line-of-sight range. These radios utilize the ZigBee networking protocol, which rides atop the IEEE 802.15.4 radio protocol. They run Ember ZB firmware, which uses the AODV MANET routing protocol. AODV is a reactive routing protocol that discovers routes on-demand, maintains routing tables, eliminates loops, finds the shortest route, and accounts for node changes.[24] Thus, when the radios are placed in a static relay topology (point to point to point), the AODV protocol will relay the messages from beginning to end.

Figure 5.1 shows a functional diagram of the XBee radio system. Two laptops were set up with USB serial terminals. One XBee was plugged into the USB port of the first laptop. Another XBee was plugged into a power supply. The last XBee was plugged into the USB port of the second laptop. Figure 5.2 shows a picture of the lab bench setup.

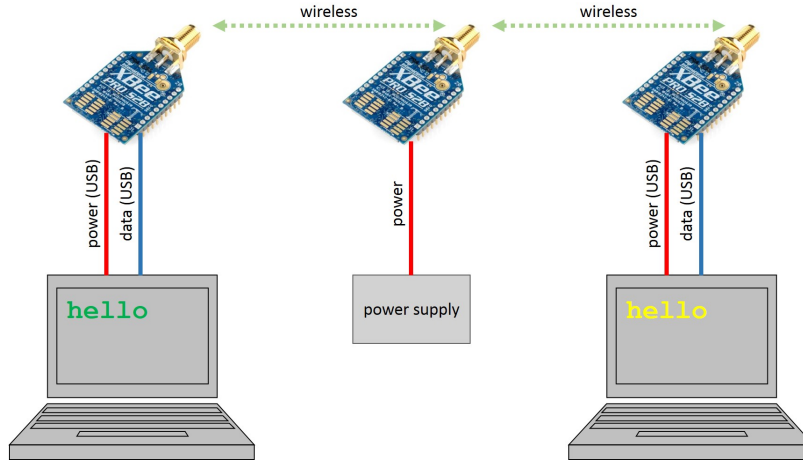


Figure 5.1: Functional diagram for lab testing of the XBee radio system.

As in the Webots simulation, the XBees were configured as Coordinator, Router,

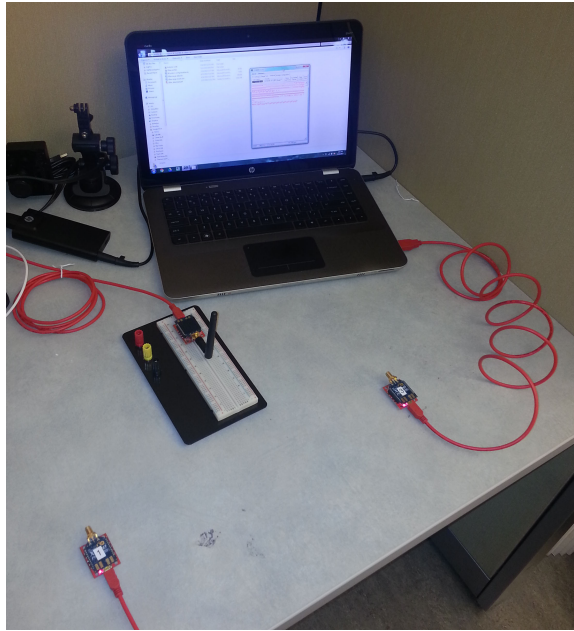


Figure 5.2: Lab bench testing setup of the XBee radio system.

and End Device nodes. By configuring each node with proper destination addresses, network identifications, and specific functionalities, a mobile ad-hoc network was established. The XBee configuration files can be found in Appendix A.2.

With standard 2 dBi ducky antennas, communication range between two XBees spanned the entire length of the engineering building. Thus, forcing the radios to communicate in a relay topology was a challenge. To make two radios exceed their transmission range, either the distance needs to be increased, or the signal needs to be attenuated. The former was not possible in a lab setting, and the latter requires expensive equipment. However, it was found that disconnecting the antenna and transmitting through the bare RPSMA connector, the signal is attenuated enough such that the maximum transmission range is approximately half a meter. This solution is convenient, inexpensive, and yields consistent results.

The Coordinator and End Device nodes are verified to work point-to-point by sending transparent messages from the serial terminal on one computer to the serial terminal on another. The two radios are then moved apart until signal was lost (verified by a failure to send messages). Connection was lost at approximately 1 meter. Then a third XBee radio, configured as a Router node, is powered on equidistant between the first two nodes. Within a few seconds the node is discovered by the existing two-radio network and the network begins to again transmit messages using the third radio as a relay. The transmission distance of the three-radio network is approximately double that of the two-radio network.

5.2.2 Field Testing

In order to test a relay communication system, a test must be implemented to allow full-power XBees with antennas to exceed their maximum range (which can be up to one mile). Also, testing on the ground introduces many inconsistent obstructions, such as terrain, Fresnel zones, and urban features. Furthermore, physically moving nodes across the ground at such distances can be tedious and time consuming.

To address these issues, a two unmanned aerial vehicles (UAVs) are developed in order to break transmission ranges and test MANETs in a relay topology under more favorable circumstances.

To decrease overall cost and streamline the building of UAVs, they are constructed using off-the-shelf components. Two foam plane platforms are selected. First is a Bixler 2, which has a 1.5 meter wingspan and is powered by a 1300kv brushless motor. The stripped airplane weighs 0.76 kg and has a 0.5 kg payload potential. The plane is powered by one 4000 mAh 11.1 V Lithium polymer battery with an estimated flight time of 50-60 minutes.

The second plane is a Flyzone Calypso, which has similar weight and power to the Bixler 2, except for a slightly wider wingspan of 1.8 meters. The overall design of both planes is sufficient to reach and maintain waypoints for the purpose of these tests. Figure 5.3 shows the Bixler 2 plane in-flight during testing.



Figure 5.3: UAV in flight during testing. Red circle highlights the plane, which is zoomed in the inset image.

The on-board electronics of both planes consist of two main components: (1) the autopilot with sensors and (2) the microcontroller with XBee radio.

1. The autopilot is an ArduPilot Mega 2.5. The board includes a 3-axis gyro, accelerometer, and magnetometer as well as a barometer. It also interfaces with the on-board GPS and Pitot tube. This autopilot has the benefits of being open source, low cost, and widely documented. It has multiple modes of operation, including waypoint nav-

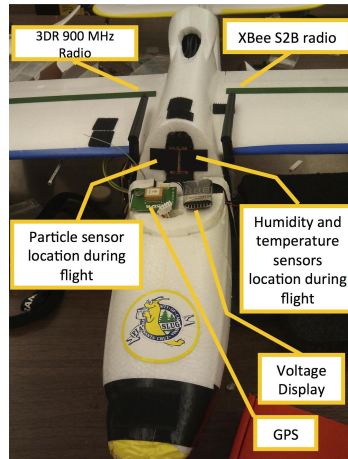


Figure 5.4: The fully instrumented Bixler 2 plane.

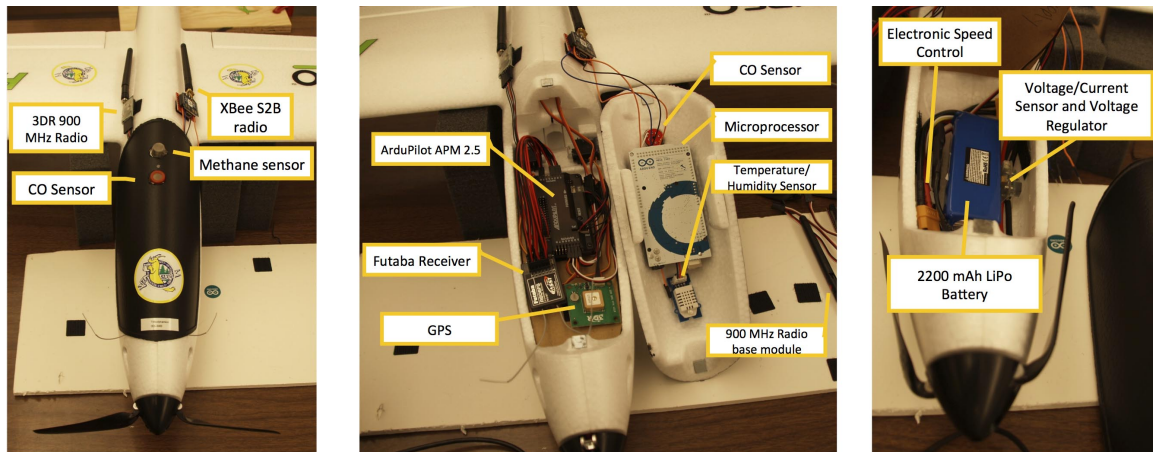


Figure 5.5: The fully instrumented Flyzone Calypso plane.

igation, stabilized flight (pilot-in-the-loop), and RTL (return to launch).

2. The microcontroller is an Arduino Mega 2560, which can communicate over serial to the XBee radio module and autopilot. This will enable the plane to serve as a node in the relay communication system.

All test flights take place on an open field, far from any hazards. This field is mostly flat with a moderate hill in the center. The hill is situated such that the opposite

end of the field cannot be seen from the other end.

Figure 5.6 shows the statistics from a successful single plane test flight. The upper-left plot shows the altitude of the plane over time. The upper-right plot shows a post-processed visualization of the GPS data within Google Earth. The plot on the lower-left shows the humidity and temperature data gathered via the XBee radio over the course of the flight.

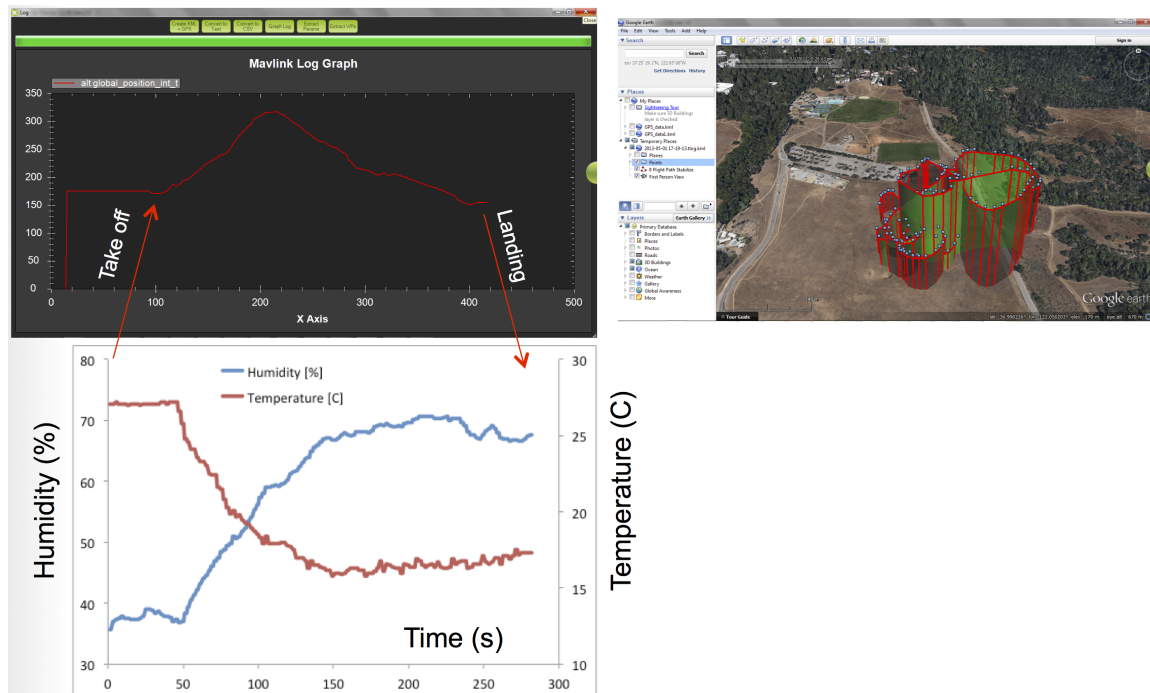


Figure 5.6: Typical set of results obtained from one UAV.

The next step is to fly two planes to validate the communication relay ability of the XBee radios. Both the Bixler and Calypso planes are fully instrumented for flight, complete with XBee radio packages. The base station is a laptop connected to an XBee running the UAV software and a serial terminal to receive XBee transmissions. The goal of this test is to begin with no radio connection between the base station and a distant UAV.

A second UAV will then fly between the base station and the distant UAV in order to relay transmissions.

The XBee radios are configured in the same way as lab testing. The base station uses the Coordinator-configured XBee, the relay plane uses the Router-configured XBee, and the distant plane uses the End Device-configured XBee.

Figures 5.7, 5.8, and 5.9 are live base station screenshots taken during the flight tests, showing the GPS locations of the two planes (“Plane 1” and “Plane 2”), the hill, and the base station, as well as the XBee radio terminal (the white window on the right). The terminal is blank when the communication link is broken and is displaying sensor data when the link is good.

Figure 5.7 shows the initial state of the test. The base station and Plane 1 are on the ground at one end of the field and Plane 2 is on the other end of the field. The hill is directly between the two points and completely obstructs the line-of-sight. Therefore, there is no connection to Plane 2.

In Figure 5.8, Plane 1 is flying a triangular waypoint pattern over the hill and base station. When Plane 1 is directly above the hill, there is line-of-sight between the base station and Plane 1, and line-of-sight between Plane 1 and Plane 2. The XBee communication system is then able to establish a link between the base station and Plane 2 via Plane 1. This is evident in the terminal read-out, as the humidity and temperature data is streaming into the base station from Plane 2.

In Figure 5.9, Plane 1 is still flying the triangular waypoint pattern, but is now directly above the base station. In this situation, the hill again blocks line-of-sight between

Plane 2 and the other two radios. The connection to Plane 2 is again lost. This scenario was repeated multiple times and yielded consistent results. As Plane 1 flew its pattern, connection to Plane 2 would cut in and out, depending on the line-of-sight.

The field testing successfully validated the early models and simulations completed, as well as the testing done in the lab.



Figure 5.7: Both planes on the ground on opposite ends of the field. No line-of-sight. No communication link.



Figure 5.8: Plane 1 operating as a relay node for Plane 2. Direct line-of-sight.

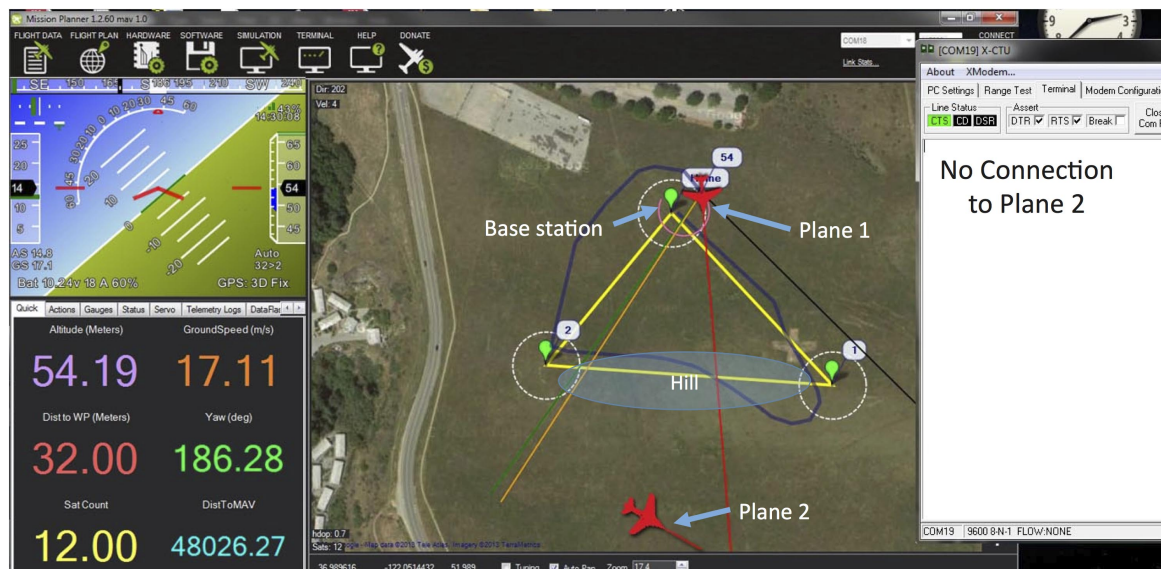


Figure 5.9: Plane 1 is directly above the base station. No line-of-sight. No communication link.

Chapter 6

Conclusion

The results in this thesis illustrate the benefits of using a relay communication system with small robots. Communication distance is increased as a function of robot quantity by relaying transmissions across a queue of multiple robots. This enables small, inexpensive robots to carry out mission objectives that were previously impossible. This is evident in UAV flight testing, where a relay node successfully establishes communication with another node located too far away. This distant node is then able to send sensor data that was previously going unread.

Simulation tools are utilized to explore the electromagnetic propagation affects, 3D physical modeling, and network-level simulations involved in a multi-robot mobile ad-hoc network (before real world validations are pursued). Simulations of radio wave propagation in various scenarios illustrates the affects of Fresnel zones and line-of-sight obstructions and how they significantly impede radio communications. 3D physical modeling of a multi-robot system shows how complicated a complete simulation solution needs to be, and how spe-

cialized simulation tools are especially necessary. And network simulations allow protocols to be modified to improve desired characteristics of an existing standard.

Chapter 7

Future Work

The early foundations of this thesis are based on a concept of a fleet of small lunar rovers. A large lander would launch from Earth carrying many small rovers. Upon landing on the moon, these rovers would individually exit the lander and drive in a straight queue toward a desired destination. Along the way, the momentarily trailing rover would stop and hold position. When the leading rover reaches the destination, the rest of the rovers would be roughly equally spaced between the lander and the destination to overcome Fresnel effects and line-of-sight obstructions. With this topology of wireless nodes, a simple MANET can be implemented to establish communication between the lander and the rest of the rovers. This is especially useful on the rough terrain of the moon, where rocks and craters can easily block communication without strategically-placed relays.

With an established mobile ad-hoc network of small rovers, tasks on the lunar surface can be completed while maintaining full communications. One such task could be prospecting for lunar resources. A fleet of specialized small rovers can prospect an area more

completely than one large rover. And the rovers can work together to efficiently transport materials from remote locations to a home base. The system would also have excellent redundancy, since the loss of one small rover can be acceptable.

Appendix A

Appendix

A.1 Webots Controller Code

```
/*  
  
 * File:      robot1.c  
  
 * Date:      5/14/2013  
  
 * Description: COORDINATOR  
  
 * Author:    Jeremy Baumgartner  
  
 *  
 */  
  
  
#include <webots/robot.h>  
  
#include <webots/differential_wheels.h>  
  
#include <webots/receiver.h>  
  
#include <webots/emitter.h>  
  
#include <stdlib.h>
```

```

#include <stdio.h>

#include <string.h>

#include <time.h>


#define ROBOTS 3

#define TIME_STEP 64


#define DELAY 100 //ms


//A sample message for an XBee radio

#define START_BYTE "7E"

#define DATA_LEN_MSB "00"

#define DATA_LEN_LSB "05"

#define DEST_ADDR "0003" //16-bit

#define MAX_HOPS "15"

#define DATA "FFFF"

#define CHECKSUM "00"


int main() {

    wb_robot_init();


    WbDeviceTag receiver;

    WbDeviceTag emitter;

    const char *name;

```

```

char message_tx[24];

int left_speed, right_speed;

name = wb_robot_get_name();

receiver = wb_robot_get_device("receiver");

emitter = wb_robot_get_device("emitter");

wb_receiver_enable(receiver, 100);

//FILE *file;

//file = fopen("output1.txt","w");

while(wb_robot_step(TIME_STEP)!=-1) {

    /* Transmit messages */

    //Sample message: 7E0005000315FFFF00

    snprintf(message_tx, strlen(message_tx)+1, "%s%s%s%s%s%s", START_BYTE,

        DATA_LEN_MSB, DATA_LEN_LSB, DEST_ADDR, MAX_HOPS, DATA, CHECKSUM);

    printf("%s\n", message_tx[0]);

    wb_emitter_send(emitter, message_tx, strlen(message_tx)+1);

    //fprintf(file,"%s\n",message_tx);

    //system("pause");

    /* Compute the motor speeds */

    left_speed=1;

```

```
right_speed=0;

/* Set the motor speeds. */

wb_differential_wheels_set_speed(left_speed, right_speed);


//fclose(file);

}


//fclose(file);


return 0;

}
```

```
/*  
  
 * File:      robot2.c  
  
 * Date:      5/14/2013  
  
 * Description: RELAY/REPEATER  
  
 * Author:    Jeremy Baumgartner  
  
 *  
 */  
  
#include <webots/robot.h>  
  
#include <webots/differential_wheels.h>  
  
#include <webots/receiver.h>  
  
#include <webots/emitter.h>  
  
#include <stdlib.h>  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <time.h>  
  
#include <unistd.h>  
  
#define ROBOTS 3  
  
#define TIME_STEP 64  
  
int main() {  
  
    wb_robot_init();
```

```

WbDeviceTag receiver;

WbDeviceTag emitter;

const char *name;

char message_rx[64];

char message_tx[64];

int left_speed=0;

int right_speed=0;


name = wb_robot_get_name();

receiver = wb_robot_get_device("receiver");

emitter = wb_robot_get_device("emitter");

wb_receiver_enable(receiver, 100); //unsure about ms time


//FILE *file;

//file = fopen("output2.txt","w");


while(wb_robot_step(TIME_STEP)!=-1) {

    /* Receive messages */

    while (wb_receiver_get_queue_length(receiver) > 0) {

        const char *message_rx = wb_receiver_get_data(receiver);

        const double *dir = wb_receiver_get_emitter_direction(receiver);

        double signal = wb_receiver_get_signal_strength(receiver);

        //printf("received: %s (signal=%g, dir=[%g %g %g])\n",message_rx, signal,

            dir[0], dir[1], dir[2]);
    }
}

```



```

    //printf("2received: %s\n", message_rx);

    wb_receiver_next_packet(receiver);

    int left_speed=-1;

    fprintf(file,"%s\n",message_rx);
}

//fprintf(file,"%s\n","TEST");

/* Transmit messages */

//sprintf(message_tx, "Hello%d");

wb_emitter_send(emitter, message_rx, strlen(message_rx) + 1);

//printf("2sent: %s\n", message_rx);

/* Compute the motor speeds */

left_speed=0;

right_speed=1;

/* Set the motor speeds. */

wb_differential_wheels_set_speed(left_speed, right_speed);

//fclose(file);
}

return 0;
}

```

```
/*

* File:      robot3.c

* Date:      5/14/2013

* Description: END DEVICE

* Author:    Jeremy Baumgartner

*

*/

#include <webots/robot.h>

#include <webots/differential_wheels.h>

#include <webots/receiver.h>

#include <webots/emitter.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <time.h>

#define ROBOTS 3

#define TIME_STEP 64

int main() {

    wb_robot_init();

    WbDeviceTag receiver;
```

```

WbDeviceTag emitter;

const char *name;

char message_rx[64];

int left_speed, right_speed;


name = wb_robot_get_name();

receiver = wb_robot_get_device("receiver");

emitter = wb_robot_get_device("emitter");

wb_receiver_enable(receiver, 100);


while(wb_robot_step(TIME_STEP)!=-1) {

    /* Emit data

    sprintf(message, "Hello%d");

    wb_emitter_send(emitter, message, strlen(message) + 1);*/

    /* Receive messages */

    while (wb_receiver_get_queue_length(receiver) > 0) {

        const char *message_rx = wb_receiver_get_data(receiver);

        const double *dir = wb_receiver_get_emitter_direction(receiver);

        double signal = wb_receiver_get_signal_strength(receiver);

        printf("received: %s (signal=%g, dir=[%g %g %g])\n",message_rx, signal,

            dir[0], dir[1], dir[2]);

        //printf("%s\n", message_rx);

        wb_receiver_next_packet(receiver);
    }
}

```

```

    }

    /* Compute the motor speeds */

    left_speed=1;

    right_speed=0;

    /* Set the motor speeds. */

    wb_differential_wheels_set_speed(left_speed, right_speed);

}

return 0;
}

```

A.2 XBee Radio Configuration

Coordinator configuration

```

XBP24-ZB_20A7_S2B.mxi
80
0
301
20A7
0
[A] ID=2001
[A] SC=7FFF
[A] SD=3
[A] ZS=0
[A] NJ=FF
[A] DH=13A200
[A] DL=409E95C7
[A] NI=

```

[A] NH=1E
[A] BH=0
[A] AR=FF
[A] DD=30000
[A] NT=3C
[A] NO=0
[A] CR=3
[A] SE=E8
[A] DE=E8
[A] CI=11
[A] PL=4
[A] PM=1
[A] EE=0
[A] EO=0
[A] BD=3
[A] NB=0
[A] SB=0
[A] R0=3
[A] D7=1
[A] D6=0
[A] CT=64
[A] GT=3E8
[A] CC=2B
[A] SP=20
[A] SN=1
[A] D0=1
[A] D1=0
[A] D2=0
[A] D3=0
[A] D4=0
[A] D5=1
[A] P0=1
[A] P1=0
[A] P2=0
[A] PR=1FFF
[A] LT=0
[A] RP=28
[A] D0=1
[A] IR=0
[A] IC=0
[A] V+=0

Router configuration

XBP24-ZB_22A7_S2B.mxi

80

0

301

22A7

0

[A] ID=2001

[A] SC=7FFF

[A] SD=3

[A] ZS=0

[A] NJ=FF

[A] NW=0

[A] JV=0

[A] JN=0

[A] DH=13A200

[A] DL=40A04D7E

[A] NI=

[A] NH=1E

[A] BH=0

[A] AR=FF

[A] DD=30000

[A] NT=3C

[A] NO=0

[A] CR=3

[A] SE=E8

[A] DE=E8

[A] CI=11

[A] PL=4

[A] PM=1

[A] EE=0

[A] EO=0

[A] BD=3

[A] NB=0

[A] SB=0

[A] RO=3

[A] D7=1

[A] D6=0

[A] CT=64

[A] GT=3E8

[A] CC=2B

[A] SM=0

[A] SN=1

[A] SO=0

[A] SP=20
[A] ST=1388
[A] P0=0
[A] D0=1
[A] D1=0
[A] D2=0
[A] D3=0
[A] D4=0
[A] D5=1
[A] P0=1
[A] P1=0
[A] P2=0
[A] PR=1FFF
[A] LT=0
[A] RP=28
[A] D0=1
[A] IR=0
[A] IC=0
[A] V+=0

End Device configuration

XBP24-ZB_28A7_S2B.mxi

80

0

301

28A7

0

[A] ID=2001

[A] SC=7FFF

[A] SD=3

[A] ZS=0

[A] NJ=FF

[A] JN=0

[A] DH=0

[A] DL=0

[A] NI=

[A] NH=1E

[A] BH=0

[A] DD=30000

[A] NT=3C

[A] NO=0

[A] CR=3

[A] SE=E8
[A] DE=E8
[A] CI=11
[A] PL=4
[A] PM=1
[A] EE=0
[A] EO=0
[A] BD=3
[A] NB=0
[A] SB=0
[A] RO=3
[A] D7=1
[A] D6=0
[A] CT=64
[A] GT=3E8
[A] CC=2B
[A] SM=4
[A] ST=1388
[A] SP=20
[A] SN=1
[A] S0=0
[A] P0=0
[A] D0=1
[A] D1=0
[A] D2=0
[A] D3=0
[A] D4=0
[A] D5=1
[A] P0=1
[A] P1=0
[A] P2=0
[A] PR=1FFF
[A] LT=0
[A] RP=28
[A] D0=1
[A] IR=0
[A] IC=0
[A] V+=0

Bibliography

- [1] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2(1):1–22, January 2004.
- [2] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. IEEE 802.11b ad hoc networks: Performance measurements. *23rd International Conference on Distributed Computing Systems Workshops 2003 Proceedings*, pages 135–145, 2005.
- [3] TX Brown, B Argrow, and C Dixon. Ad Hoc UAV Ground Network (Augnet). *AIAA*, pages 1–11, 2004.
- [4] Zong Da Chen, HT Kung, and Dario Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 247–250. ACM, 2001.
- [5] Imrich Chlamtac, Marco Conti, and Jennifer J.-N. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, July 2003.
- [6] Ian T Downard. Simulating sensor networks in ns-2. Technical report, DTIC Document, 2004.

- [7] Andrew Drenner, Ian Burt, Bradley Kratochvil, Bradley J Nelson, Nikos Panikolopoulos, and KB Yesom. Communication and mobility enhancements to the scout robot. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 865–870. IEEE, 2002.
- [8] Robert Faludi. *Building wireless sensor networks: with ZigBee, XBee, arduino, and processing.* ” O’Reilly Media, Inc.”, 2010.
- [9] Brian Gerkey, Richard T Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th international conference on advanced robotics*, volume 1, pages 317–323, 2003.
- [10] DB Green and AS Obaidat. An accurate line of sight propagation performance model for ad-hoc 802.11 wireless LAN (WLAN) devices. *2002 IEEE International Conference on Communications*, pages 3424–3428, 2002.
- [11] Sebastian Hening, Jeremy Baumgartner, Candace Walden, Russell Kirmayer, Mircea Teodorescu, Nhan Nguyen, and Corey Ippolito. Distributed sampling using small unmanned aerial vehicles (uavs) for scientific missions. *AIAA Infotech@Aerospace Conference*, 2013.
- [12] Remcom Incorporated. Wireless insite 2.7.1 reference manual. 2014.
- [13] P. Jacquet, P. Muhlethaler, and T Clausen. Optimized link state routing protocol for ad hoc networks. 2001.
- [14] Marko Korkalainen and Mikko Sallinen. A survey of rf-propagation simulation tools for

- wireless sensor networks. In *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pages 342–347. IEEE, 2010.
- [15] David Kotz, Calvin Newport, Robert S Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 78–82. ACM, 2004.
- [16] Raymond J Luebbers. Finite conductivity uniform gtd versus knife edge diffraction in prediction of propagation path loss. *Antennas and Propagation, IEEE Transactions on*, 32(1):70–76, 1984.
- [17] Gary Scott Malkin. Rip version 2. *IETF RFC 2453*, 1998.
- [18] Vachirapol Mayalarp, Narisorn Limpaswadpaisarn, Thanachai Poombansao, and Somsak Kittipiyakul. Wireless mesh networking with xbee. In *2nd ECTI-Conference on Application Research and Development (ECTI-CARD 2010), Pattaya, Chonburi, Thailand*, pages 10–12, 2010.
- [19] Olivier Michel. Webots: Symbiosis between virtual and real mobile robots. In *Virtual Worlds*, pages 254–263. Springer, 1998.
- [20] John Moy. Ospf version 2. *IETF RFC 2328*, 1997.
- [21] HG Nguyen, N Pezeshkian, A Gupta, and N Farrington. Maintaining Communications Link for a Robot Operating in a Hazardous Environment. *ANS 10th International Conference on Robotics and Remote Systems for Hazardous Environments*, 2004.

- [22] HG Nguyen, N Pezeshkian, Michelle Raymond, Anoop Gupta, and Joseph M Spector. Autonomous communication relays for tactical robots. Technical report, DTIC Document, 2003.
- [23] David Oran. Osi is-is intra-domain routing protocol. *IETF RFC 1149*, 1990.
- [24] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. *IETF RFC 3561*, pages 1–38, 2003.
- [25] Charles Perkins and Pravin Bhagwat. Highly Dynamic (DSDV) for Mobile Computers Routing. *SIGCOMM Comput Commun Rev*, pages 234–244, 1994.
- [26] Narek Pezeshkian, HG Nguyen, and Aaron Burmeister. Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations. *IASTED Int Conf on Robotics and Applications*, 2007.
- [27] Maj FJ Pinkney, Dan Hampel, and Stef DiPierro. Unmanned aerial vehicle (uav) communications relay. In *Military Communications Conference, 1996. MILCOM’96, Conference Proceedings, IEEE*, volume 1, pages 47–51. IEEE, 1996.
- [28] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5, 2009.
- [29] Ram Ramanathan and Jason Redi. A brief overview of ad hoc networks: challenges and directions. *IEEE Communications Magazine*, 2002.

- [30] E.M. Royer and CK Toh. A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, (April):46–55, 1999.
- [31] J Schuster and R Luebbers. Hybrid sbr/gtd radio propagation model for site-specific predictions in an urban environment. *12th Ann. Rev. of Progress in Applied Computational Electromagnetics*, 1:84–92, 1996.
- [32] Joseph Schuster and Raymond Luebbers. Comparison of site-specific radio propagation path loss predictions to measurements in an urban area. In *Antennas and Propagation Society International Symposium, 1996. AP-S. Digest*, volume 2, pages 1210–1213. IEEE, 1996.
- [33] Illya Stepanov, Daniel Herrscher, and Kurt Rothermel. On the impact of radio propagation models on manet simulation results. In *Proceedings of 7th International Conference on Mobile and Wireless Communications Networks (MWCN 2005), Marrakech, Morocco, 2005*.
- [34] Cagatay Tanil, Chirag Warty, and Esam Obiedat. Collaborative mission planning for uav cluster to optimize relay distance. In *Aerospace Conference, 2013 IEEE*, pages 1–11. IEEE, 2013.
- [35] Mauro Tortonesi, Cesare Stefanelli, Erika Benvegna, Ken Ford, Niranjan Suri, and Mark Linderman. Multiple-uav coordination and communications in tactical edge networks. *Communications Magazine, IEEE*, 50(10):48–55, 2012.
- [36] András Varga et al. The omnet++ discrete event simulation system. In *Proceedings of the European simulation multiconference (ESM2001)*, volume 9, page 65. sn, 2001.

- [37] Howard Xia, Henry L Bertoni, Leandro R Maciel, Andrew Lindsay-Stewart, and Robert Rowe. Radio propagation characteristics for line-of-sight microcellular and personal communications. *Antennas and Propagation, IEEE Transactions on*, 41(10):1439–1447, 1993.
- [38] Brian M Yamauchi. Packbot: a versatile platform for military robotics. In *Defense and Security*, pages 228–237. International Society for Optics and Photonics, 2004.
- [39] Chang Fa Yang, Boau Cheng Wu, and Chuen Jyi Ko. A ray-tracing method for modeling indoor wave propagation and penetration. *IEEE Transactions on Antennas and Propagation*, 46(6):907–919, 1998.
- [40] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. Glomosim: a library for parallel simulation of large-scale wireless networks. In *Parallel and Distributed Simulation, 1998. PADS 98. Proceedings. Twelfth Workshop on*, pages 154–161. IEEE, 1998.