Fall 2013

Intelligent aerial store & foreword packet repeater

William P. McDonnell
University of New Hampshire, Durham

Recommended Citation

McDonnell, William P., "Intelligent aerial store & foreword packet repeater" (2013). *Master's Theses and Capstones*. 821.
https://scholars.unh.edu/thesis/821

INTELLIGENT AERIAL STORE & FOREWORD PACKET REPEATER

BY

William P. McDonnell

BSEE, University of New Hampshire, 2011

THESIS

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Master of Science in

Electrical Engineering

September 2013

This thesis has been examined and approved.

Michael J Carter

Thesis Director, Michael J. Carter,
Associate Professor of Electrical Engineering

W. Thomas Miller, III,
Professor of Electrical Engineering

L G Kraft

L. Gordon Kraft,
Professor Emeritus of Electrical Engineering

1 August 2013

Date

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

INTELLIGENT AERIAL STORE & FORWARD PACKET REPEATER

by

William P. McDonnell

University of New Hampshire, September, 2013

A communication framework capable of rapid deployment and adaptive wireless support was designed and implemented using an unmanned aerial vehicle equipped with a 900 MHz, frequency- hopping transceiver configured as a store and forward packet repeater. Users with or without line of sight propagation between one another can automatically connect through the packet repeater and employ the aerial platform for extended data transfer. The airborne vehicle accommodates dynamic re-positioning in response to varying radio link conditions, thus supporting communication between highly mobile and/or line of sight-obstructed users even as the network topology evolves. Using open source and custom written software applications, as well as specially modified radio firmware, a command and data-logging environment was designed to monitor, control and initialize radio network conditions and vehicle platforms in real time. Careful real world evaluation of the developed system has demonstrated a robust platform capable of improvement to a user's communication performance.

Chapter 1

ISSUES WITH LOW POWERED WIRELESS COMMUNICATION

1.1 Line of Sight Reliance

Wireless communication systems provide a means of exchanging information and synchronizing actions at a distance. These accomplishments are only made possible with a persistent connection between isolated users. The reliability of these communication systems is of paramount concern in order to guarantee safety of operators and correct interpretation of user intentions. In many situations the propagation environment plays a pivotal role in determining the quality of a wireless channel. Specifically, in cases involving low powered transmission there is a significant degree of reliance on line of sight (LOS) between the source and destination. When LOS cannot be provided, the link quality is brought into question and reliability may suffer

Wireless communication is traditionally facilitated through the use of an existing infrastructure such as repeaters, cell towers or satellites. Wireless networks often employ one or more of these cooperative techniques between the source and destination in an attempt to mitigate the aforementioned line of sight reliance. These methods serve the task of bolstering messages en route, thereby mitigating the surrounding environmental effects. While these techniques of network enhancement can significantly improve a static wireless channel, an existing communication infrastructure is required. The development of such an infrastructure generally takes time and money to complete, which places strict limitation on the scale and location of such developments. Furthermore, these systems are only effective while users remain in the general vicinity of such infrastructure, which places strict limitations on mobility.

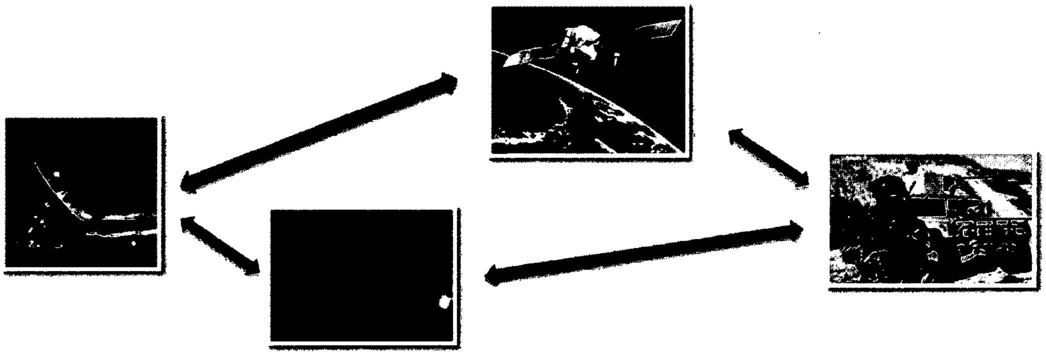


Figure 1.1 – Wireless User Reliant on Existing Infrastructure

There are many scenarios in which users require long range low power data transmission in environments devoid of existing infrastructure. Operation of semi-autonomous vehicles in hazardous environments often requires traveling inside of a structure while the operator remains outside at a safe distance. Generally the propagation path between the user and robot is obstructed, thus limiting the achievable safe distance from the hazard. Search and rescue often takes place in complex rural environments (e.g., mountainous terrain) that make team coordination difficult. Constant contact between first responders is absolutely necessary for a coordinated and efficient effort to take place. Military exercises are frequently carried out in hostile environments devoid of friendly communication amenities. These are situations where communication is absolutely critical, yet typical equipment performs poorly.

1.2 Features of a Communication Relay

A communication relay is one of the simplest forms of cooperative data transfer in a network setting. Unlike a typical point to point connection, information does not flow directly from user to user. Instead, data is first routed to a relay node, digitally regenerated and bolstered in strength and then transmitted onward to the next node. This process can be continued through a series of daisy chained relays creating networks that can be specifically tailored to the necessary transmission range. The repeater action may be accomplished through a store and forward method. Data received from one user is recorded in local memory at the relay point and transmitted onward during the next available timeslot. The relay retransmits this data at full power down to the next user creating the illusion that users are closer than they are actually spaced.

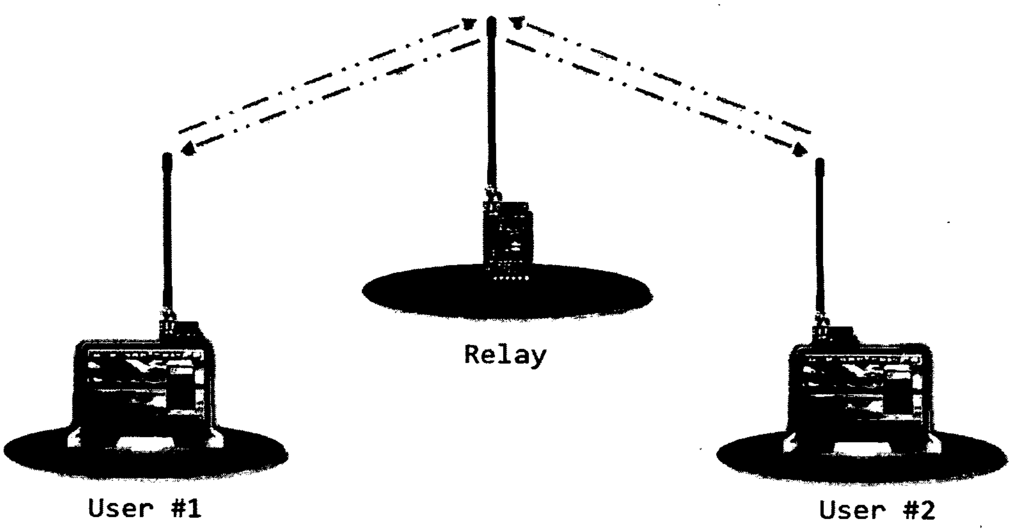


Figure 1.2 - Single Hop Relay Configuration

While there exist many cooperative network topologies, the relay structure benefits from a number of key features. The simplicity of the relay architecture makes it easy to implement and less prone to errors prevalent in systems of greater complexity. Data is simply rebroadcasted by each repeater without the need to route traffic. While not unique to relays, the communication protocol supports merged serial data streams (e.g. control, sensor, link statistics etc.) Relays can be linked together to form networks which facilitate short, medium or long distance communications. Relays provide a significant boost in range capabilities between users, thus creating the possibility of low power operation. Furthermore relays can be used to overcome physical obstacles by creating emulated line of sight between users (e.g. satellite repeaters). Relays support communications between users who would normally be unable to exchange information. Through cooperative store and forwarding actions, relay networks provide critical improvement to existing wireless networks through increased reliability and range

1.3 Possible Relay Platforms

A typical relay platform involves an integrated transceiver module fixed to a static platform such as a tower, wall fixture or a naturally elevated object. The position of the relay relative to users is central to the device's ability to create reliable range extension. For example, a relay that is placed low to the ground or obstructed by obstacles will provide little utility to the user. Ideal relay function usually involves elevating and positioning the relay platform at the midpoint between users. In this configuration a trade-off between the reliability and transmitted power levels is achievable. While balance is obtained in static scenarios, users generally move about their environment creating difficulty in maintaining a constant signal quality on either side of the relay link.

While a fixed relay cannot to alter its position under time-varying link conditions, a mobile platform can. Fixed wing or lighter than air vehicles pose a viable relay platform given their ability to vary altitude and position over time. Fixed wing aerial vehicles generate flight through a combination of rotor thrust and wing shape. The aerodynamics of this vehicle allows for gliding over distances with or without power as long as the vehicle continues along a forward trajectory. Unlike a rotary winged device such as a helicopter, a fixed wing vehicle is unable to loiter about a point by hovering. To remain in a particular area of interest a fixed wing vehicle must circle about a point. This creates variation in the network propagation conditions as the repeater deviates from the point of link performance balance. On the other hand, a fixed wing vehicle can remain aloft for a longer period of time since less thrust is needed to generate lift than for a rotary wing aircraft.

A lighter than air vehicle could also serve as a mobile platform in a similar fashion to the fixed wing example. This vehicle benefits from the ability to generate lift through a contained gas at lower density than the external air as opposed to rotors or rocket propellant. This type of vehicle can stay aloft for extended periods of time. Under fair flight conditions the vehicle can remain about a fixed point without the need to move forward at a constant rate. Recently Google has come forward with plans to equip blimps with Wi-Fi routers in order to extend the internet to remote locations of the world [8]. These high altitude platforms allows for millions of people over hundreds of square miles to gain access to data services in areas without communication infrastructure. While a lighter than air vehicle does exhibit positive characteristics, there is still a number of issues preventing this platform from being ideal as a radio relay host. Network conditions can fluctuate rapidly due to motion of ground vehicles while the airship may be unable to respond at a quick enough rate to retain a continuous communication link. Blimps are also highly visible and could easily be targeted in a military scenario.

1.4 Selected Relay Platform

Figure 1.3 – Quadcopter Platform

A low cost quadcopter was selected to serve as the relay platform due to a number of key features lacking in other aerial vehicles. This highly maneuverable aerial vehicle supports flight through the combination of four planar rotors. The quadcopter can operate with six degrees of freedom by translating along its orthogonal axes as well as rotating about them at a rapid rate. Since lift is generated through vertically oriented rotors as opposed to a fixed wing, the quadcopter can loiter about a position indefinitely (subject to power source constraints) and provide superior positioning capabilities. The vehicle is extremely stable and can oppose strong wind gusts due to high thrust motors and limited cross sectional area. The quadcopter can operate in dense urban environments, or even indoors, where traditional aerial vehicles would be unable to navigate. If used in a military theater the quadcopter's high mobility and low radar cross section is an additional benefit. This combination of high maneuverability and stability makes for an ideal vehicle to satisfy the needs of a relay node.

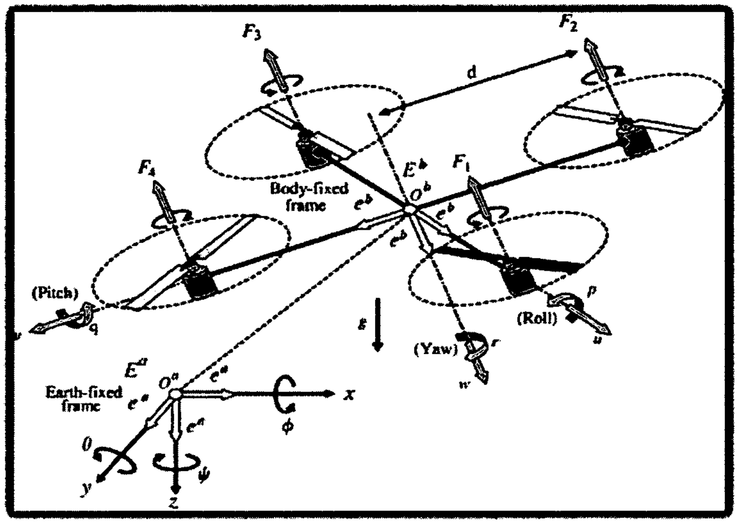


Figure 1.4 – Quadcopter Six-Degrees of Freedom Diagram

(Figure Credit: IEEE Aerospace and Electronic Systems [6])

The quadcopter platform provides a low cost vehicle capable of autonomous or guided flight through the use of an integrated flight controller. This component houses an inertial measurement unit (IMU), global positioning system (GPS), barometric altimeter and ultrasonic proximity sensors in addition to a built-in microcontroller. The flight controller performs the critical task of sensing the vehicle's orientation, acceleration, position and altitude. From these measurements decisions can be made as to the correct thrust levels of motors to accomplish complex maneuvers such as waypoint tracking, hovering and circling. The quadcopter's onboard flight controller supports a powerful communication protocol known as MAVlink, which enables reporting of the vehicle's current operating condition and reception of high level user commands. The use of this protocol allows for access to the controller system without the need to alter the existing embedded applications.

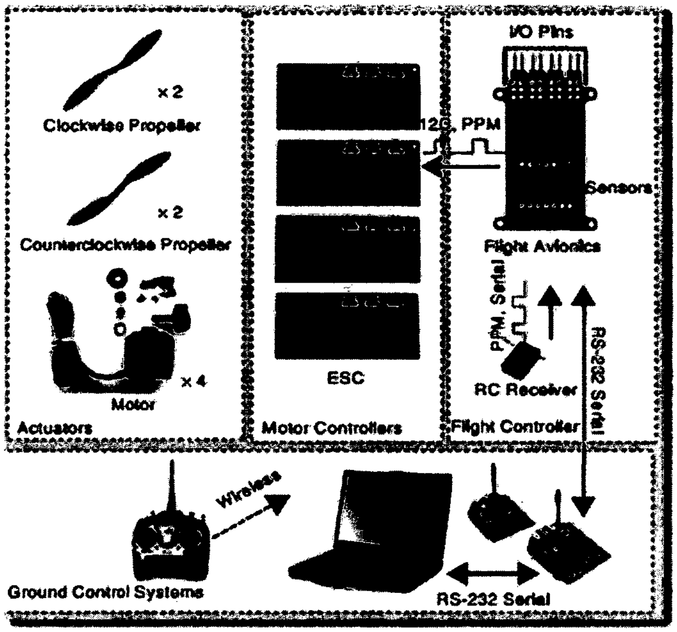


Figure 1.5 - Quadcopter Components and Control Methods

Figure Description: The quadcopter can be manually operated with a traditional radio controller-receiver combination. High level commands (e.g., waypoints, arm, land, hold position) can also be sent to the vehicle over wireless telemetry.
(Figure Credit: IEEE Aerospace and Electronic Systems [6])

The quadcopter can be controlled in a number of ways including direct manual control and high level user commands. A standard wireless RC controller can be used to maneuver the vehicle by supplying raw pulse width modulation signals, which are directly applied to the quadcopter's motor controllers. The difficulty of manual control can be relaxed through a series of semi-autonomous flight modes such as stabilization, altitude fix, and loiter. Users can transmit mission commands to the quadcopter through a wireless serial link to implement semi-autonomous or scripted flight patterns. The quadcopter provides a powerful platform where users can leverage control over the system through the established vehicle command and control framework.

1.5 Remote Ground Rover

A treaded vehicle ("tank") platform was built to demonstrate the principles of a remotely operated vehicle. All commands and sensor reports exchanged between the ground station and ground rover are routed through the relay radio. This utilization of the relay platform allows the tank to remain connected to the ground station over great distances and without line of sight propagation. The tank platform controller houses a number of subsystems including a GPS, microcontroller, motors, IMU and radio transceiver module. Mobility is provided by two brushed DC motors connected to rubber treads through toothed wheels. The tank was programmed to report back to the ground station with data pertaining to its health, position and orientation, thus providing detailed information on the rover's current status.

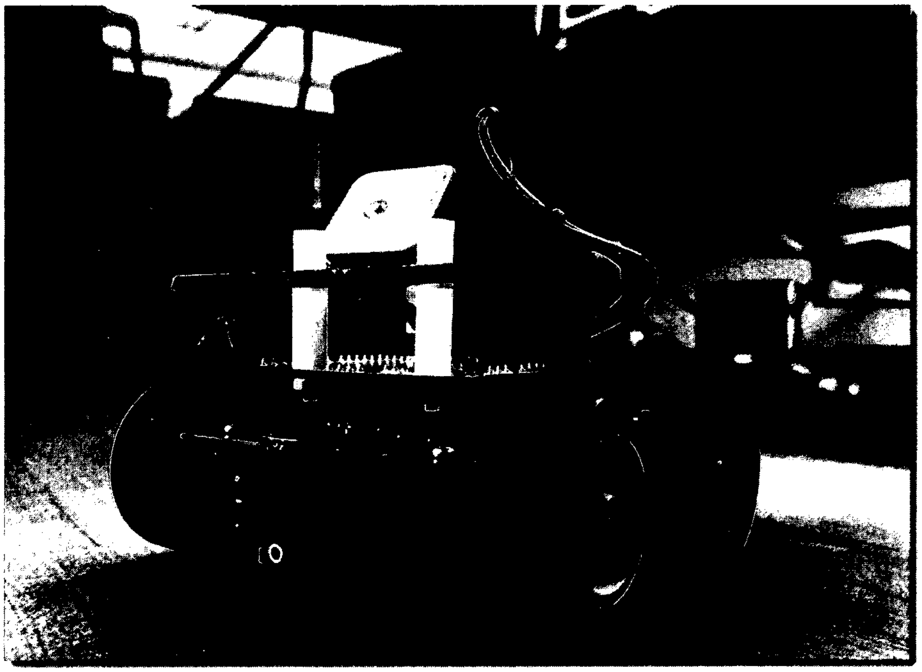


Figure 1.6 - Tank Platform Serving as End User

1.6 Structure of Thesis

Chapter two discusses the evolution of repeater systems, related research and how this undertaking improved upon preceding work. This section starts with a brief history of repeaters and the recent desire to create aerial versions. A number of papers are summarized on topics related to the designed intelligent aerial relay network (IARN). Finally the improvement IARN provides compared to previous research is discussed. Chapter two places the project in context with the current state of aerial relay research. Chapter three serves to recount the technical implementation of the relay network. The section starts with an introduction to the networks purpose and features. This is followed by a description of the radios physical and data layers. The chapter concludes with a discussion of the control functionality of the quadcopter and data logging provided through ground station.

In Chapter 4, the reader is presented with a series of experiments to evaluate the network and application's true capabilities. Preliminary data link layer tests are discussed to demonstrate the initial link issues and approach with which these problems were resolved. Next, a set of outdoor experiments are detailed and results are displayed. Chapter 4 ends with an explanation of outdoor tests with the quadcopter in flight and initial dynamic positioning investigations. Chapter five describes the project summary followed by possible future work and system improvements. For the curious reader, additional appendices are included with further detail on project resources such as references, data sheets and source code.

Chapter 2

STORE & FORWARD DIGITAL REPEATERS

2.1 Evolution of Repeaters in Airborne Platforms

The modern digital store and forward repeater can trace its origins back thousands of years to the times of the ancient Romans. A repeater is defined as "a relayer of a message" and in Roman times warnings of approaching armies were accomplished with smoke and fire. While the manner in which information is transferred has changed drastically, the concept has not. Fast forward a couple thousand years, and the same ideas were implemented with magnetism through the invention of the telegraph and Morse code signaling. This method of data transfer is simply an extension of the Roman signal fire, improving upon the original in rate, range and information density. In modern times information can be exchanged through binary data modulated RF waveforms. We can now send pictures, voice, video and text information at blistering rates over great distances with little overhead.

The relay concept is certainly not a new idea as demonstrated in the previous discussion. With an ever present desire to improve our technology, there has been interest expressed in using repeaters in a number of non-standard scenarios. As detailed in the first chapter, the positioning of a relay platform is pivotal in determining the system's effectiveness. Naturally, engineers realized that repeaters placed at high altitudes can provide coverage over larger areas than system located near the ground. A satellite is therefore an ideal platform to support terrestrial relaying of data over extreme distances. Unfortunately the cost of such systems prohibits their use by individuals and small user groups in all but the most life-critical contexts.

As detailed in Chapter One, there is simply not enough existing infrastructure to support users operating via line of sight in rural environments. These are scenarios where repeaters stand to significantly improve wireless channels, especially when they are located at high altitudes and are capable of movement. This desire for improvement has sparked the design of aerial platforms with embedded relays to support communication on a large scale [2, 3, 4]. Until recently, the operating cost and logistics prevented general deployment of such systems. A number of advancements in various fields of engineering has allowed for progress toward affordable adaptive aerial relays.

With the reduction of transistor footprints in integrated circuits, the physical size of many communication systems can be decreased drastically, thereby requiring less mass, volume and power consumption. This is a critical criterion to meet as many aerial platforms have strict upper limits on their acceptable payload dimensions and weight. UAVs have become more and more commonplace in military and commercial regimes.

While these platforms are typically utilized for surveillance and reconnaissance, their greater availability has permitted the re-purposing of many platforms for communication support [1]. The current research in the field of airborne relays is in an early state with a number of simple simulations and early prototypes. These researched topics are similar in nature to this thesis and provide context for the work performed.

2.2 Related Aerial Relay Research

As with many commercially available technologies, the first aerial relays were developed for use in military theaters. With any conflict there is a need for isolated ground units to stay connected. The soldier is dependent upon communication for coordination and friend-foe identification. These needs have generated military interest in allocating manned and unmanned aerial platforms for communication purposes. The classified nature of these projects makes it difficult to accurately pinpoint the absolute first application of aerial communication relays. Modern utilization of an unmanned airborne vehicle was most likely to have taken place after 2006 to support operations in Afghanistan. This system involved an RQ-7B Shadow aerial vehicle with a Falcon III radio installed within the UAVs tail boom [1]. In mid-2007 the US Army first implemented a Communication Relay Package-Light (CRP-L) onboard a similar UAV platform. This system was able to support communication over significantly longer ranges than those typically experienced by line of sight VHF-UHF radio transceivers.

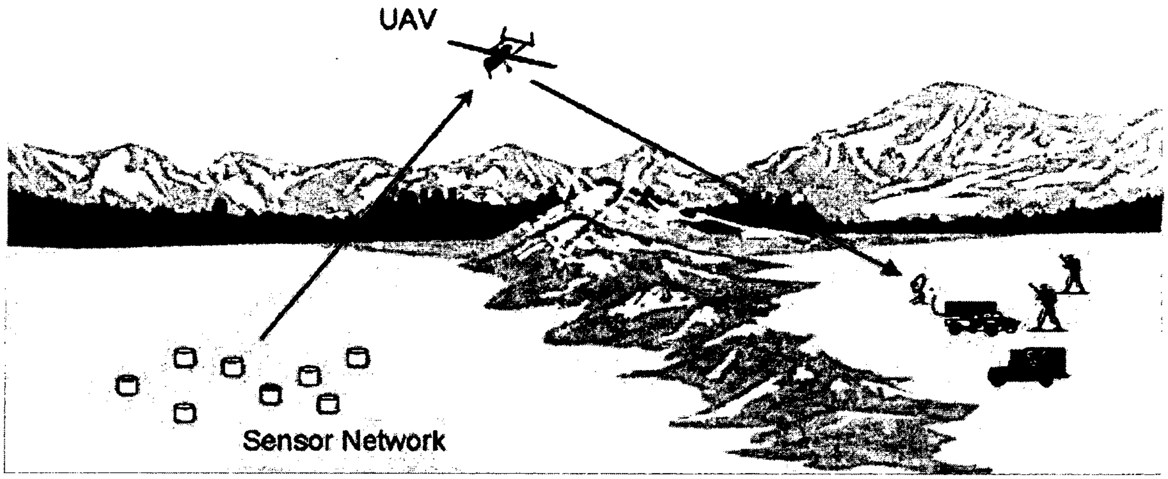


Figure 2.1 - Military Application of Airborne Relays

(Figure Credit: A New Method for Distributing Power Usage Across a Sensor Network [9])

While these early airborne relays provide improved communication reliability for soldiers operating with LOS radio equipment, these vehicles were most often manually controlled by human operators. Constant supervision of the UAV is required to properly place the relay within range of units as they navigate the environment. In an effort to eliminate the human operator, there is a need for intelligent positioning algorithms to be researched and implemented. A few months after the deployment of the CRP-L system, engineers from UCLA published results for such an algorithm in a military communication conference [2]. The team researched a positioning algorithm designed to aid a wireless *ad hoc* network composed of UAV platforms. Based on the desire to improve throughput performance and reliability, the algorithm calculates the three dimensional placement of nodes in order to maximize communication performance. While this research is an exciting first step towards intelligent aerial relays, it is simply a program without physical incarnation. Performance tests were performed within the confines of a simulation without real world experiments, thus calling into question the algorithm's true capability.

In a similar vein, Zhan *et al.* investigated the benefits of an unmanned aerial vehicle as a platform for a digital repeater [3]. The authors derived the expected environment propagation influences and simulated communication performance. The link quality was graded based on the average data rate versus the symbol error rate, which provided a metric to maximize. The developed algorithm was tasked with adjusting the vehicle's commanded heading in an attempt to optimize communication performance for ground nodes. A handoff process was implemented to enable improved communication as the network topology evolves due to user divergence. The results of this investigation demonstrate an algorithm capable of maintaining a link-optimizing heading under simulation conditions. As the vehicles are highly mobile, a handoff procedure was shown to greatly benefit network performance. This research was performed within the confines of computer simulation and mathematical modeling. A series of simplifying assumptions was made to reduce system complexity at the cost of accuracy. Communication between all nodes is assumed to be error free and data transfer overhead is considered nonexistent. Many assumptions were made during the derivation of channel models, which further reduced the simulation outcome reliability.

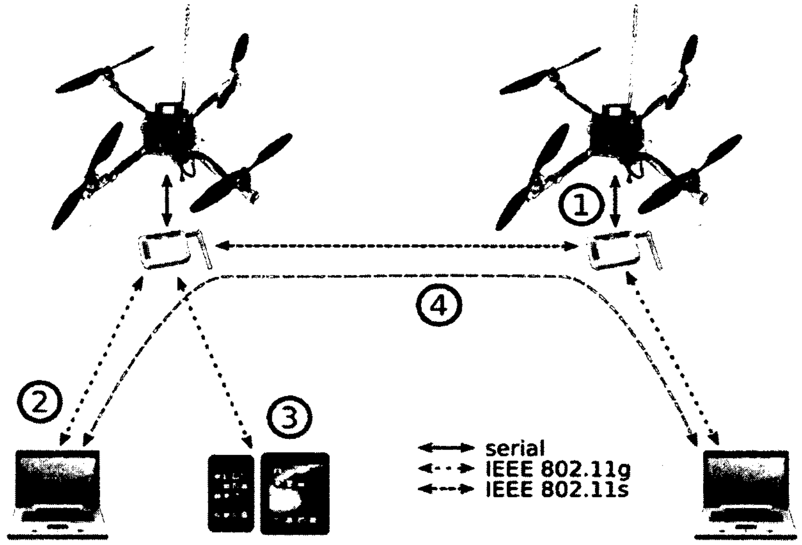


Figure 2.2 - UAVNet Wireless Network Utilizing Quadcopter Platforms

(Figure Credit: UAVNet: A mobile wireless mesh network using Unmanned Arial Vehicles [4])

The papers previously discussed took a theoretical approach to the relay positioning problem at hand. In these papers a number of simplifying assumptions are made about the channel model, propagation environment, user movement and link overhead in an effort to reduce complexity. While a significant reduction in complexity is achieved, the designed system cannot be directly applied for real world operation.

At the midpoint of this thesis research (December 2012), a paper was published by a group of engineers at the University of Bern in Switzerland [4] that partially overcomes the limitations of the previous studies on intelligent aerial relay networks. However, the author of this thesis did not become aware of the publication until the conclusion of his research. This very recent paper bridges the gap between the theoretical basis provided in previous work and actual practice of a real world solution. UAVNet is a framework for automatically establishing IEEE 802.11g/s wireless mesh networks with a number of aerial relay nodes (quadcopters). The network functions to provide reliable wireless access to isolated users who would otherwise be unable to communicate. UAVNet serves as a platform upon which complex control and data routing algorithms can be implemented and thoroughly tested.

In an effort to support connected devices with a reliable network, the airborne nodes are designed to actively position themselves based on the current propagation environment. UAVNet is capable of two basic positioning modes to place relay nodes between end users. A geometric mode uses the global position information of end users to calculate a midpoint and direct the relay platforms to this location. A more rigorous positioning method utilizes the link signal strength between active users to account for variation in the network propagation environment. These two positioning modes allows for dynamic link support, thus ensuring a stable link even when the network topology evolves. These positioning modes and associated algorithms were also developed independently by this thesis author prior to learning of the publication by the University of Bern group.

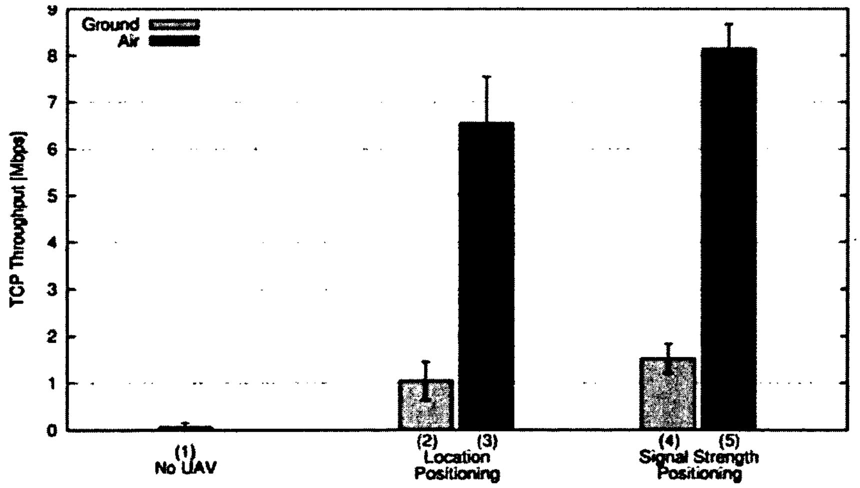


Figure 2.3 - UAVNet Data Rate Performance for Three tests

(Figure Credit: UAVNet: A mobile wireless mesh network using Unmanned Arial Vehicles [4])

Evaluation of the relay platform was performed by placing the network nodes in a variety of locations by manual means while network throughput was recorded. The initial test had the UAV placed on the ground directly between the two netbooks to simulate a ground relay system. Next the quadcopter was placed at the geographic midpoint to simulate the UAVNet geometric positioning method. Finally the link received signal strength was balanced between the two endpoints (quadcopter was walked to this location and fixed at an elevated level) to test the networks capacity to dynamically position itself based on the current link conditions. After these simple tests were run the network throughput was analyzed to determine the differences between each geometric node placement. Results proved that the inclusion of the UAV relay platforms provided a significant improvement to network reliability and throughput over a simple static ground relay. When the two positioning modes were compared versus one another, it was determined that the signal-strength based positioning provided better performance than the simple midpoint-based location algorithm as seen in Figure 2.3.

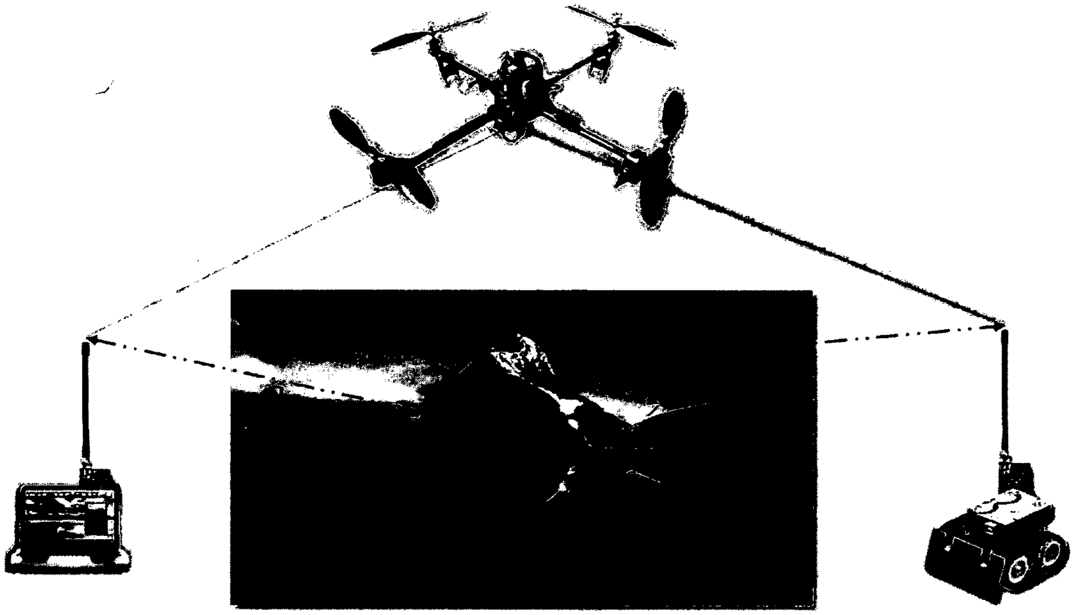


Figure 2.4 - Intelligent Aerial Relay Network (IARN) Configuration

Figure Description: The intelligent aerial relay involves the following components (ground station, tank and quadcopter vehicle). The quadcopter incorporates the relay onboard, thus providing the advantage of height and mobility. (Note: Actual platforms vary in appearance)

The research reported in this thesis also establishes an intelligent aerial relay platform, and some important contrasts are drawn with the prior work in the literature. The intelligent aerial relay network (IARN) was developed and implemented over the course of this thesis project. Units operating in rough terrain typically experience reduced range and transmission reliability due to multipath, as well as geometric path loss caused by the intrinsic complexity of the propagation environment. IARN aims to automatically acquire and dynamically support users nominally communicating through line of sight methods.

The recent surge of interest in aerial platforms for use as communication relay nodes indicates that this thesis topic is worthy of investigation. While the developed framework yields a number of improvements over previous research in the field of aerial networks, it is necessary to discuss similarities and differences between the related studies and the present work.

A number of themes are consistent throughout the prior work reported in the literature and the present thesis research. In all cases the researchers worked on a facet of a communication scenario involving wireless end users and an aerial network access point. Optimizing and dynamically supporting wireless users served as the underlying motivator for each project. The various teams all worked toward this overarching goal, with a slightly different approach taken in each case. While the first two papers [2,3] focused on the theoretical positioning of UAV network nodes, there were many simplification made throughout the modeling and simulation process that limited the limited the applicability of the results to the practical real world environment. For a useable prototype to be developed, the whole problem must be approached without seeking generality or by decreased scope of applicability.

IARN builds upon the concept of improved network access through a dynamic aerial relay platform. Whereas previous research narrowly focused on a subset of the overall problem and with only simulations, IARN was implemented and tested as a fully operational platform. Of all prior research on this topic, the UAVNet paper bears the strongest resemblance to the developed IARN framework. These two projects were developed independently without knowledge of one another, thus demonstrating the practicality of the achieved research. While these two systems accomplish similar feats (automatic networking, dynamic node placement, aerial mobility etc.) there are a number of notable differences and improvements with IARN. The open mesh network nodes of UAVNet utilize the IEEE 802.11g standard with orthogonal frequency-division multiplexing at 2.4 GHz to implement wireless networks. While this protocol yields a significantly higher throughput than the (IEEE 802.11g: 54 Mb/s versus 3D Robotics radio module: 256 kb/s) these speeds are only obtainable at short distances.

The use of orthogonal frequency-division multiplexing causes increased interference between adjacent nodes requires users to operate beyond a strict minimum separation distance. The open source communication protocol utilized for IARN was designed to minimize interference through time division multiple access with protective guard intervals. This protocol allows for networking with any range of user proximity. Additionally, the lower 915 MHz carrier frequency selected for IARN experiences less path loss than 2.4 GHz, which extends communications range. Finally, UAVNet was developed by a team of four individuals while this thesis research was performed solely by the author. In all, IARN provides a cost effective, low power and interference free framework for supporting the communication requirements of end users.

Chapter 3

IMPLEMENTATION OF AN INTELLIGENT AERIAL RELAY

3.1 High Level Link Description

A relay platform was developed to serve the essential role of connecting two remote user nodes through a transparent serial link. A tank unit, quadcopter and ground station serve user roles, sinking and sourcing data that propagates over the link. The designed network satisfies communication needs of users, keeping them updated and in synch. Through a combination of software design, hardware selection and data level organization the network supports the circulation of multiple message types to and from multiple users without issue. Messages can contain commands, user reports, link statistics and other information. This versatility lends itself to a wide range of potential applications for the established communication framework.

Unlike a typical point to point wireless link where data flows directly between users, the relay platform operates in a cooperative "store and forward" fashion. Data transmitted between the ground station and tank platform endpoints are first routed through the radio repeater before arriving at their destination during the next communication round. The repeater located on the quadcopter is the midpoint in the link joining the end users while simultaneously providing the quadcopter with a critical connection to the ground station. This cooperative setup was designed to mitigate issues with multipath propagation and line of sight obstructions, which improves, in turn, overall link quality.

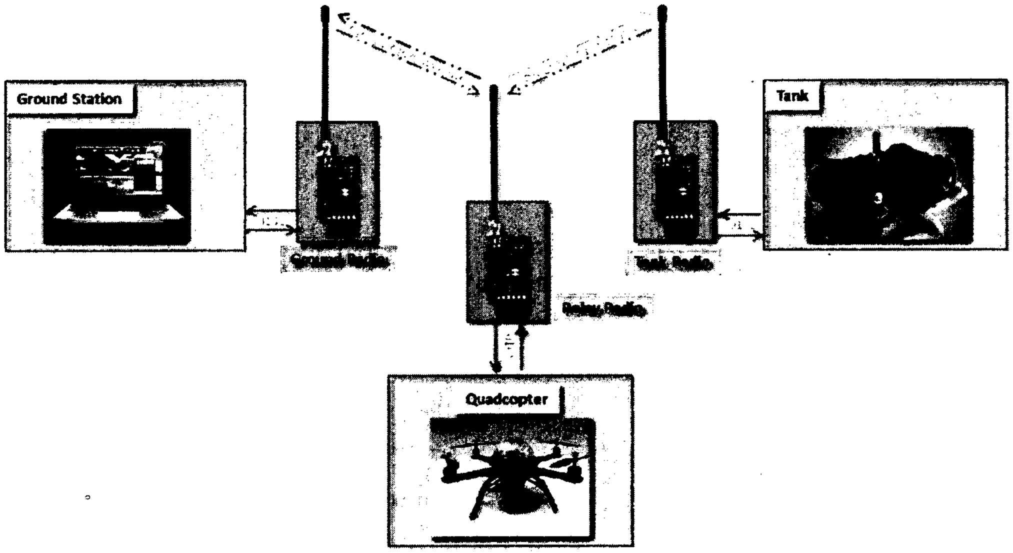


Figure 3.1 - High Level Communication Diagram

The network supports two simultaneous traffic serial streams through a merged data channel. Communication needs between the ground station and tank as well as the ground station and quadcopter can be provided concurrently. Data transmitted from the ground station is a combination of commands intended for the tank and quadcopter while data received at the ground station is a fusion of the quadcopter and tank sensor status and radio link statistic streams. A merged serial structure onboard the relay and ground platforms reduces the computational load on the embedded applications through reduced system complexity. This provides higher reliability of the radios in addition to lower system latency. Parsing of the serial stream is therefore performed by the end user, which partially alleviates the load on the relay node.

3.2 Radio Physical Layer

The selection of a radio transceiver was made on the basis of the desire for low power consumption, useful data rate, bidirectional communication, programming ease and modest cost. After careful review of relevant products on the market, the Si1000 transceiver system on a chip was selected to serve as the hardware upon which the link was to be built. The Si1000 is a powerful mixed signal microcontroller connected over a serial parallel interface (SPI) to a Si4430 integrated transceiver chip. The chip is powered by a custom 8051 high speed controller core that provides the system with basic instruction execution, interrupt handling and code storage at a high rate. The Si1000 also contains a number of configurable peripherals that allow for temperature sensing, UART communications, high precision timing and voltage comparison.

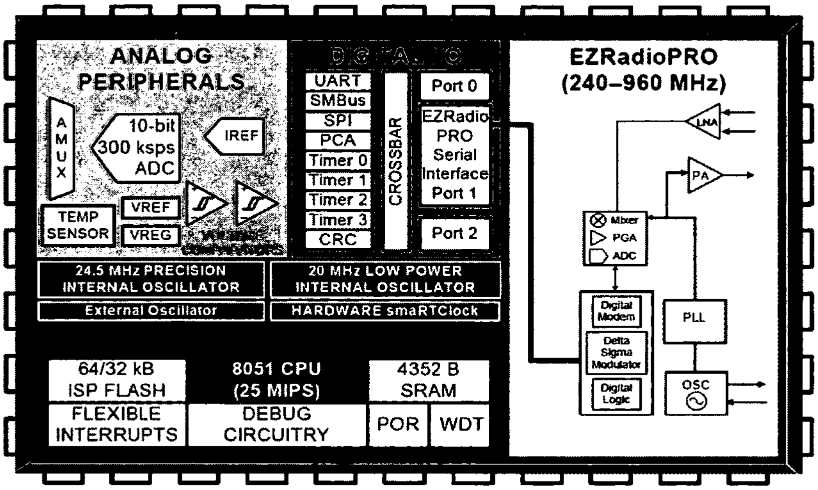


Figure 3.2 - Si1000 System Architecture

As mentioned previously, the Si1000 interfaces with a Si4430 wireless transceiver module over a serial parallel interface. This provides high speed communication between the two devices as well as code execution and task isolation. The main microcontroller unit (MCU) can operate on instructions independent of the transceiver and vice versa. Only when cases arise where the two need to exchange data does main code execution need to be halted. The Si4430 contains the necessary radio frequency hardware for the reception and transmission of frequency shift keying (FSK) digital modulation. Mixers, a low noise preamplifier, power amplifier, phase lock loop, RF switch and delta sigma digital to analog converter are integrated in the chip package. The Si1000 fulfills the requirements necessary to construct the communication relay link in addition to many other features that could be fully utilized in future firmware development efforts.

While the Si1000 can be purchased as a standalone product, it was decided early in the thesis research that a radio module utilizing the embedded Si1000/Si4430 system was more desirable. A number of companies provide "all in one" radio modules built around the Si1000 that require no assembly. Telemetry kits were selected from 3D Robotics (San Diego, CA) in order serve as the radio link hardware. These boards were chosen over other options due to their compact package, convenient debug interface, status LEDs and included preconfigured firmware. This highly flexible telemetry module natively supports the embedded C language and its memory class extensions. A high level of hardware configuration and algorithm complexity can be accomplished through this software environment.

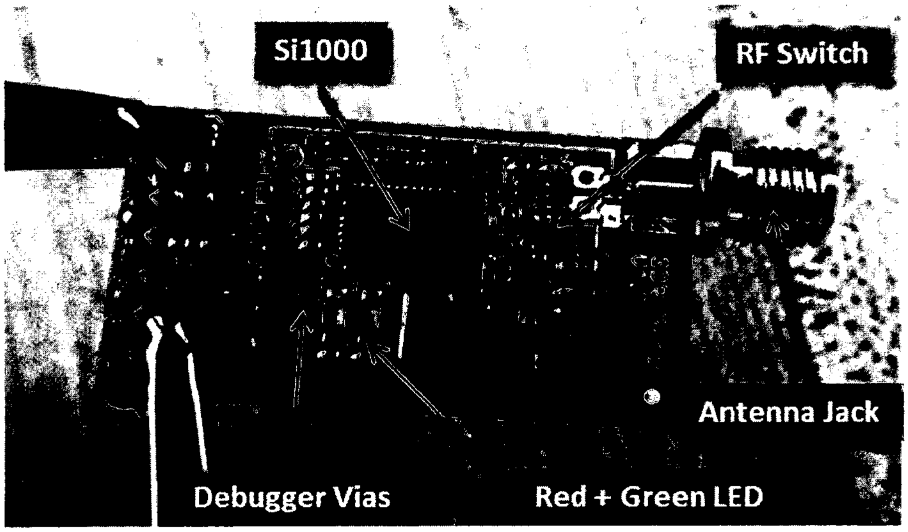


Figure 3.3 - 3D Robotics Radio Board

3D Robotics sells two such radio modules operating in either the 915 MHz or 433 MHz ISM (Industrial, Scientific and Medical) bands. For the purposes of the relay network a 915 MHz radio was chosen. Transmitting devices in the ISM bands are unlicensed, but most operate at low transmitted power levels with modest antennas, and no protection from interference by other users of the band is provided. The ISM bands, however, place no restrictions on duty cycle or application type, which lends them wide range of potential uses. The ISM range about 915 MHz specifically allows for higher output power and greater bandwidth than other ISM spectrum allocations. Furthermore, this frequency range has received less usage as compared to other ISM bands such as 2.4 GHz or 433 MHz (the 915 MHz range was widely used for cordless telephone system, but has largely been supplanted by the transition to use of cellular phones instead of wireline phones with cordless handsets in residential and small office settings). This combination creates an ideal portion of the spectrum in which to operate a wireless relay without many of the issues found at other frequencies.

While 915 MHz provides many benefits it is also worth noting a few issues associated with the band. Operating at a higher frequency than the 433MHz radio creates greater path loss, higher reflectivity in the environment and greater susceptibility to multipath fading. Without strict operating guidelines (duty cycle, access to unlicensed users) at 915 MHz, therein lies the potential for channel overcrowding and static interferers. Power and antenna gain limitation placed on users of this band prevent the relay from supporting communication beyond one kilometer. These constraints create interesting problems from an engineering standpoint. Greater path loss and multipath fading creates even more of a necessity for cooperative "store and forward" operation. Static interferers can be overcome by smart frequency hopping.

Data transmission over the radio link is performed through binary frequency shift keying (BFSK) about a center frequency. The Si4430 transceiver provides a wide range of configurable radio parameters such as data rate, transmission frequency deviation, channel frequency spacing, hopping rate and power level. This range of parameters provides a great deal of flexibility to the designer and allows many of the aforementioned spectrum issues to be overcome. A frequency hopping spread spectrum approach was taken to satisfy FCC regulations and minimize multiple user interactions on a particular channel. This frequency hop takes place at the end of each transmission round as opposed to a per packet basis.

The link was configured with a 40 kb/s data rate, 100 kHz hopping channel spacing, 20 kHz frequency deviation and a +20 dBm transmit power level. These parameters were chosen in order to maximize range while reducing the potential for link interference. While the data rate of the Si1000 can be configured up to 256 kb/s, the expense is further spectral spreading and reduced transmission range. The true upper limitation on network speed is generated by the simple fact that the transceiver modules are half-duplex. At any given time, data transfer can only occur in one direction. The radio modules must switch between receive and transmit modes to redirect the flow of data, which has a significant limitation on the overall rate at which data can propagate to the network extremities. Due to this fact, the radio modem data rate was selected to be much lower than the maximum possible value. This decision was made as the effect on network throughput was negligible while the improvement of system range was considerable.

The received radio modules arrived preloaded with an open source embedded application which provides a bidirectional transparent serial link between two paired units. This software is written to enable frequency hopping, time division multiplexing, configurable data rate, power, Mavlink framing, link statistic reporting and automatic link acquisition. Although feature-rich, the included application serves only point to point communication and does not readily allow relay functionality. Alteration of the source code was difficult without a thorough understanding of the intricate firmware architecture. For the relay functionality only a subset of the existing radio features were necessary. It was therefore more beneficial to build a custom embedded application from scratch and simply use the existing source code as a guideline for development.

The conversion of the radio module from a point-to-point serial link to a store and forward relay network was an involved process requiring the development of three embedded software applications. Each radio runs a custom set of instructions that is designed to accomplish a precise task in the overall network architecture. Applications run concurrently onboard their respective platforms and interface with one another through precisely timed wireless application programming interfaces (API). For the relay link to function properly, predictable state transitions, tolerance to timing variations, frequency hopping, automatic acquisition and transparent bidirectional serial communication are necessary.

These previously listed features were implemented onboard the radio platforms through three embedded applications. The design process started with the creation of a communication framework that would serve as the underlying code structure for every radio on the network. Each radio application was custom tailored to serve the needs of its platform but contains the same baseline structure. This provided a significant boost in coding efficiency as many methods could be reworked or simply reused as each piece of the network was created. Furthermore, troubleshooting and upgrades to the code platform could be applied to each application in many instances.

Due to the complexity of the radio firmware, a modular coding technique was utilized. The various tasks performed by the firmware were spread among these modules, thereby decreasing the code complexity of individual functions. Each separate source file contained related methods that provided similar services or relied upon one another to accomplish a task. The major modules are termed *serial*, *radio*, *physical*, *initialize* and *application*. These files are supplemented by other components that provide additional non-critical functionality. Modules serve a range of associated tasks necessary for the overall operation of the radio hardware.

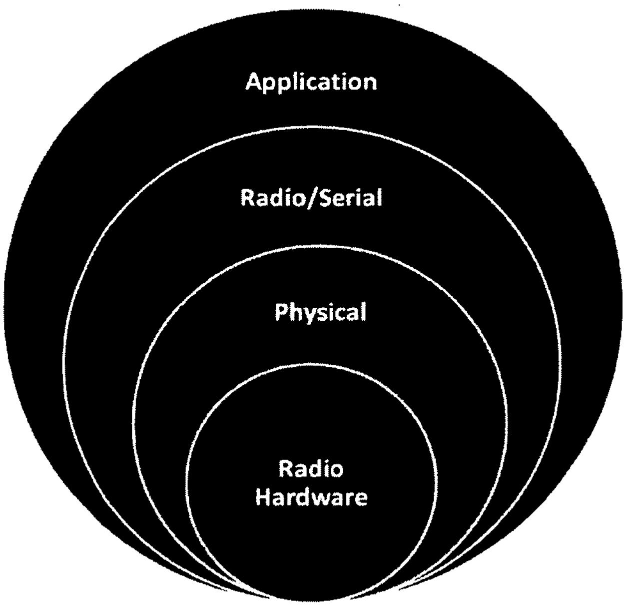


Figure 3.4 - Radio Firmware Modules

Figure Description: The four major firmware modules are shown in the above figure. The innermost circle represents the physical radio hardware components controlled by the higher level firmware modules. The circle order (Outer/highest to inner/lowest) represents the level of abstraction from the hardware details. Outer firmware modules can call inner modules but the reverse operation is prohibited.

The serial module serves as an interface between the Si1000's internal 8051 serial port and firmware buffers. The utilization of an interrupt driven architecture allows for bidirectional serial data processing in the background while applications run concurrently. All data entering or leaving the radio board is routed through this firmware module. Other parts of the radio program can access the serial module and its related functions to retrieve or transmit serial data on demand. The physical module provides the critical task of controlling and communicating with the radio transceiver. Run time calculation of all radio register settings is supported, allowing for a wide range of radio configurations such as modulation type, channel spacing and data rate. An API allows access to the radio resources through a simple protocol. Control over the radio is used to change its state between idle, transmit, and receive. Bidirectional data transfer over the wireless link is provided through this module allowing for simple data exchanges.

The radio firmware module is composed of a series of higher level methods which call functions associated with serial and physical components. The functions contained within the radio module serve the task of handling data flow between the serial and physical modules. Working together these three modules are capable of transferring bidirectional data between the serial port and the wireless network. The order in which radio functions are called is dictated by the highest level module known as the application. The application code handles the order in which the radio transitions between states and monitors the link activity. The application layer was designed to provide abstraction from the physical and radio modules through a high level interface. This design was implemented to provide powerful customization options for the radio with little alteration to underlying functions.

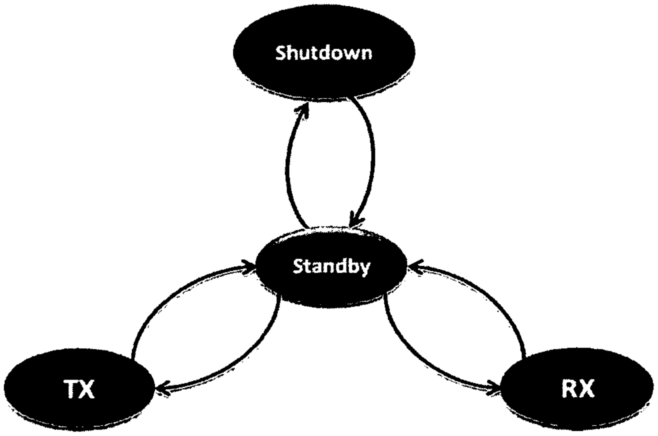


Figure 3.5 - Radio Hardware (Si4430) State Machine Diagram

Before discussing the firmware functionality of the relay, it is worthwhile detailing the communication protocol between two end nodes as the relay platform is simply an extension of this protocol. The radio platform has four modes of operation: transmit, receive, standby and silence. Transmit and receive simply configure hardware to supply serial data to the radio front end or to listen and acquire incoming radio packets. In standby mode the Si4430 chip is placed in a low power state while register settings are preserved. Transitions between transmit and receive modes can only be accomplished by first occupying standby. Due to the half-duplex nature of the radio, only one state is occupied at any given time. The underlying radio application serves to control the current radio state, transition timing, data flow and link reporting. Firmware was organized to provide methodical code flow in order to support the overarching communication system objective.

At any given time a single radio has control of the link in transmit mode. All other radios remain in receive mode unless instructed otherwise. The flow diagram representing radio state control is given in Figure 3.6 below. The transmitting radio is given a slice of time and permission to send any data existing within its internal buffers. With the exhaustion of residual data, the round is ended by transmitting an end of transmission (EOT) command. The EOT packet is only seen by the seen by the radio firmware and is filtered from the serial data stream. The packet anatomy contains a total of four ASCII encoded bytes, three start bytes (plus sign) followed by the command payload (EOT or 0x04 in Hex). Once the packet EOT is sent this radio switches into standby followed by receive mode. Upon reception of the EOT command the receiving radio switches into standby succeeded by the transmit state and the process is repeated for the duration of link activity. This simple handshaking mechanism allows for predictable state transitions without the possibility for packet collisions.

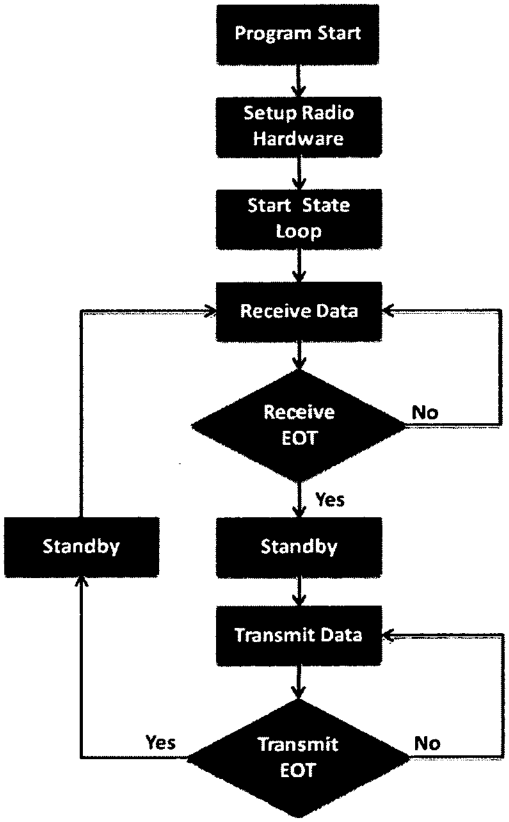


Figure 3.6 – Radio Code Execution Flow

The firmware provides a solid foundation upon which to build a communication network. Basic amenities are provided such as serial transfer, wireless reception/transmission, state transition handling and frequency hopping. As mentioned the application module provides abstraction from the lower firmware layers. This allows for relatively little alteration of the firmware while creating a radio with unique behavior. To implement the relay functionality, a series of modifications were made to this underlying firmware and associated modules on a per radio basis. The first platform developed was the ground station radio. As this radio platform only served to bridge data between the ground station's quadcopter control application and the quadcopter relay node, the software solution was relatively straightforward. The underlying firmware was utilized with little alteration as it provided the necessary amenities for serial transfer and interaction with other radio platforms.

The next system developed was the relay radio as this provides an interface between the tank and ground station. While the baseline radio firmware provided the framework of the platform, a number of modifications were necessary to implement the relay architecture. The relay communicates with two users and therefore additional states in its application layer were needed. Four states were included to perform the transfer of data between both ends of the link. Code was added to the radio module to implement behavioral alterations when the quadcopter talks to the tank versus the ground station. For example, when transmitting to the ground station, both data originating from the quadcopter as well as the tank is merged and sent. On the other hand when the relay transmits to the tank, only data coming from the ground station is repeated onward. This store and forward action was accomplished by reserving a large section of system memory for the storage of received wireless data. In the transmission phase, this buffer was used as a source of data for completing the store and forward operation.

These behavioral changes also pertain to the reception phase. The relay performs a different set of operations when receiving from the tank versus the ground station. As the ground station sends a merged serial stream that can contain instructions for the tank or quadcopter, the relay must handle the exchange properly. When the tank uses the relay to transmit information, the repeater firmware must guarantee that all tank data is forwarded to the ground station while avoiding confusing the quadcopter. In addition, the connection between the quadcopter and ground station is considered the more critical link as all flight data and commands are handled through this link. It is important that this side of the network is preserved over that of the tank link. To implement this biasing of link preservation, the firmware was altered further. This allows the relay and ground station to exchange data whether or not the tank is present on the network. Furthermore, this design renders the probability of a connection loss caused by the tank unit to be very low. These alterations to the core radio firmware serve the purpose of generating a custom radio application capable of store and forward operation.

A frequency hopping pattern was utilized to reduce system interference and provide channel separation between receiving radios. With frequency orthogonality there is no need to include addressing information in the header sent from a particular radio with destination instructions. The hopping pattern illustrated in Figure 3.7 demonstrates that only two radios will be on the same carrier channel at any given time. All packets exchanged during these states will be heard by the two participating parties while the third radio will be none the wiser. For the purpose of this network a linearly increasing frequency hopping pattern was selected.

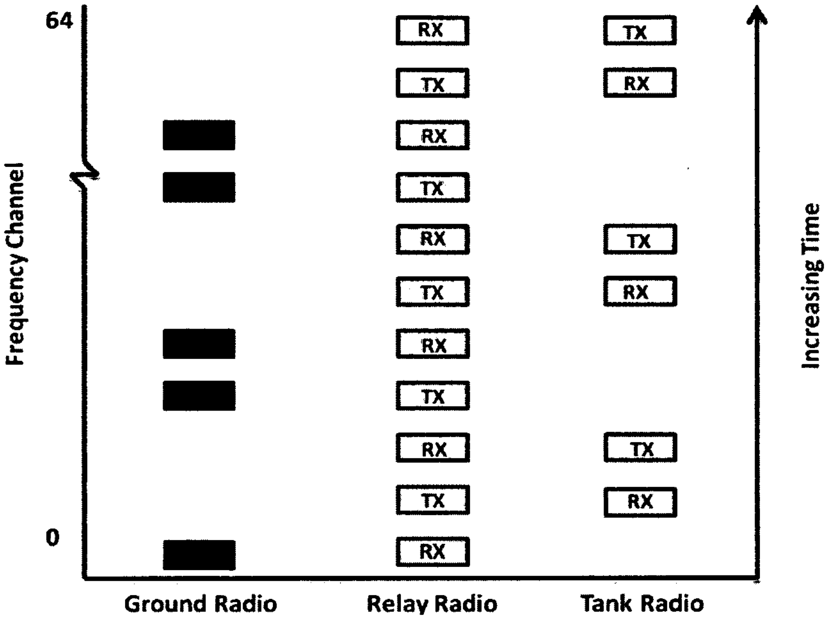


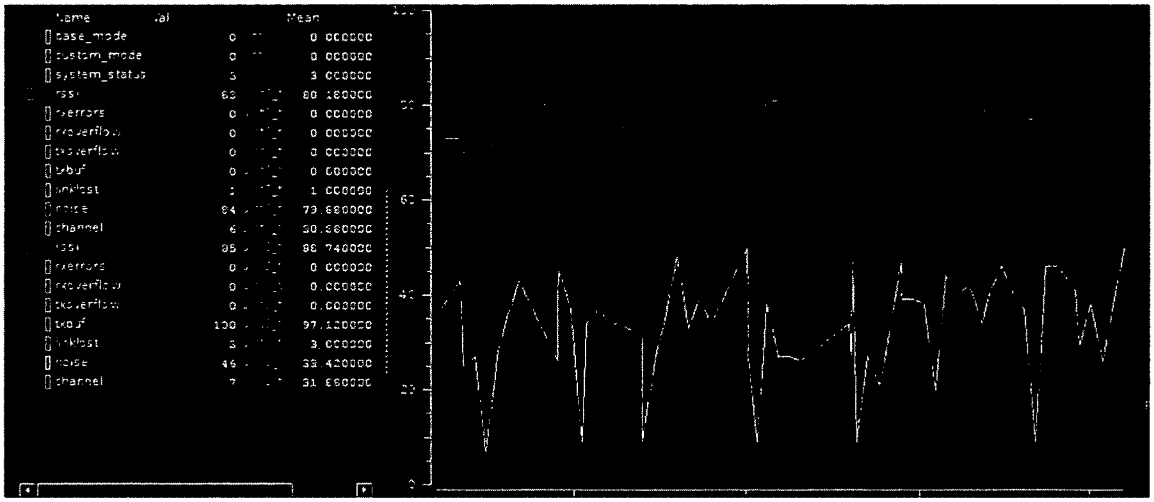
Figure 3.7 - Frequency Hopping Pattern with Time

Figure Description: The illustrated hopping pattern demonstrates the center frequency of each radio module with time. Note the relay hops one channel per transmit/receive round while the ground and tank radios hop in a 1 then 3 pattern (1 frequency hop after a reception round, 3 frequency hops after a transmission round).

The hopping pattern is designed to have the ground station and tank unit receivers always separated in frequency by at least one carrier channel spacing. This prevents the two receivers from picking one another up by direct path communication in cases where multipath or close proximity is a factor. The ground station and tank radios therefore will never "see" one another and it is up to the relay to alternate between the two end nodes and ferry the data across. Each radio is given a default channel to return to upon a link interruption or a power reset. The ground station and relay always start together on the same channel and upon their first handshake begin hopping linearly in frequency. The tank radio always starts one channel up from the ground-relay start channel. Even when the tank has not established contact with the relay it still hops linearly while it waits for the other two radios to catch up. This reduces the possibility of a static noise/interference source either preventing acquisition or inducing false acquisition.

A frequency hop takes place on each and every RX/TX round guaranteeing that no single carrier channel is used continuously for a long data exchange. Upon startup or link loss the ground station and relay start on channel zero while the tank sits on channel one. The ground station always starts off by transmitting to the relay before hopping up three channels. The relay then moves up one channel and transmits to the tank unit. Before receiving data from the tank, the radio pair hops by one channel before exchanging data. At this point the tank hops up three channels to wait for the next round of data transfer while the relay hops up by one channel. The relay and ground station are now on the same channel and the process repeats itself. Upon reaching the maximum frequency channel the pattern wraps around to the lower end of the carrier channel range and repeats itself. This repetition is accomplished through the use of a modulo operator and a base two channel count. Furthermore, repetitive frequency hopping provides guaranteed channel isolation even in the event of link losses and de-synchronization.

In addition to serial data originating at user nodes, radio link statistics packets are also generated onboard the tank and ground radio platforms. These packets are formed such that they can utilize the existing relay network without the need for a separate serial stream. The radio report payload contains valuable link statistics useful for real time visualization of the network's condition as well as diagnostics. Received Signal Strength Indication (RSSI), background noise level, receiver errors, link losses and buffer overflows are sampled on each communication round. The received signal strength indicator is sampled during the preamble of an incoming packet while there is radio frequency energy at the system front end. On the other hand, the background noise level is simply a measure of the energy present at a receiver's input during a silence period of the network. These two measurements, when properly combined, provide an estimate of signal to noise ratio (SNR) and a user's proximity to disconnection.



Name	Val	Mean
base_mode	0	0.000000
custom_mode	0	0.000000
system_status	5	3.000000
rssi	63	80.180000
rxerrors	0	0.000000
rxoverflow	0	0.000000
txoverflow	0	0.000000
txbuf	0	0.000000
linklost	1	1.000000
noise	84	79.880000
channel	6	30.360000
rssi	85	88.740000
rxerrors	0	0.000000
rxoverflow	0	0.000000
txoverflow	0	0.000000
txbuf	100	97.120000
linklost	3	3.000000
noise	46	39.420000
channel	7	31.880000

Figure 3.8 - Real Time Link Statistics Report by Radios

Figure Description: In Qgroundcontrol a real-time graph of the incoming radio statistics are displayed. In this case the tank and quadcopter noise level and signal strength is displayed according to the legend on the left side. Any of the listed data types can be checked for display and recording if desired.

The receiver error counter is incremented within the serial firmware module each time a radio packet fails a cyclical redundancy check (CRC), which generates a measurement of packet error rate. The link loss counter is increased inside of the application firmware module each time a radio returns to the acquisition phase after the connection has been lost, which does not automatically occur upon a CRC failure. Lastly, the buffer overflow measurements update each time a serial buffer runs out of space to contain data. Once a radio report is requested, all current values are encapsulated in a packet frame and transmitted at the end of a transmission round. Taken together these statistics provide useful metrics upon which to judge link quality and stability on both sides of the relay node.

The relay network state machine was designed to accomplish the required communication needs while remaining simple in nature. Providing system reliability and reasonable data rate were dominant influences throughout the design process. This delicate balance was achieved with the state machine as shown in Figure 3.9. Each end radio operates in either transmit, receive or standby mode depending on current state of the link. The relay operates in a slightly different fashion with five states required for the store and forward operation. These states consist of a transmit/receive tank, transmit/receive ground station and standby. The radios only transition between states when specific conditions are met. A radio in receive mode will only advance beyond this point if an end of transmission (EOT) command is received or a link timeout occurs. This procedural transition preserves critical link timing and eliminates issues with clock drift that might affect a more traditional Time Division Multiple Access (TDMA) protocol. The handshake protocol also supports adaptive telemetry for asymmetrical traffic loads by intelligently returning link control to other transceivers when transmission is unnecessary.

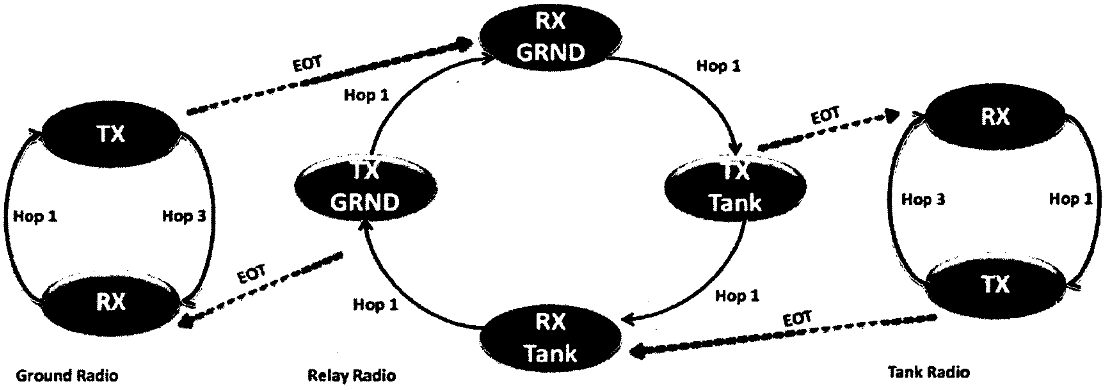


Figure 3.9 - Relay Network State Machine

Figure Description: The transition of radio states are displayed above. The network state machine transitions each time a radio transmits an EOT command and hands over control

When a radio operates in transmit mode it effectively has control over the entire link while all other radios remain in receive mode. The master radio stays in this state while data remains within its internal buffers. Upon exhaustion of its residual data the radio ends this state by transmitting the EOT command. This cues the receiving radio that it has been given control of the link and can freely transmit. The transfer of control to only one unit at a time prevents simultaneous broadcasting and the necessity to address packets to their destined user (MAVlink does contain address information on the source system and components but this is invisible to the radio firmware at the link level). While the network is operating under normal conditions, all radio states update in this simple fashion. The relay interfaces with either end of the link (tank or ground station) similar to a point-to-point connection. A transmit state takes place followed by a receive state according to the handoff procedure described previously.

In contrast to a simple point-to-point connection, data received during the previous receive mode is stored aboard the relay and utilized as a data source in the next transmission round. This operation serves to transfer data from one end of the network to the downstream user, stitching the two ends into one continuous link. The relay must therefore not only receive and transmit data, it must also be aware of the particular unit to which it is talking. Four relay states perform bidirectional communication as well as identification of a particular platform as the target destination. These states are designed to handle interfacing the relay with either end of the link and altering state behavior accordingly. The order of relay states allow for data to be passed from reception states quickly through the storage buffer, thus providing fast and reliable repeater operation.

The relay network is designed to stay connected and operating under a variety of link conditions. Each radio must remain in state synchronism with one another, but also within a timing tolerance margin as well. When no data is transmitted, the network time synchronization is a trivial matter as there is no load on the embedded systems. During periods of heavy data transfer, the radio microcontroller undergoes a significant number of additional instruction executions. This creates latency shift between the radio under load and the connected systems. If the latency is great enough there is the potential for radios to lose connectivity due to a link timeout or a missed state transition window. The link is made tolerant to time latency variation by placing silence windows (guard intervals) between RX-TX mode transitions. The length of these silence periods was selected to account for the largest possible transfer of data in a given communication round.

In order to prevent the link from hanging in the event of receiver errors or signal loss, a contingency plan is utilized. After a defined period of radio inactivity each radio goes into an acquisition phase to search and find one another. The tank and ground radios simply move into receive mode on a predetermined carrier channel of which the relay is aware. The relay goes into a search pattern by hopping to the ground radio's selected channel and broadcasting the relay radio's presence. Switching between listening and transmitting modes allows the relay to check for any active radios. If the ground radio is on and in its acquisition mode, it will respond to the relay with an acknowledgement packet.

Upon the successful completion of this exchange, the link between the ground and relay is considered active and the hunt is on for the tank radio. While the ground-relay link is down, the tank waits in receive mode, periodically hopping in frequency. Once the link between the ground and relay is established, the two radios will hop through frequency and exchange data bidirectionally while the relay simultaneously searches for the tank radio. Once the relay's transmit channel matches that of the tank's receiver, the two exchange a series of messages before that link is considered active. At this point both ends of the radio network are active and capable of data transfer.

While the relay provides more than enough data bandwidth for the connected vehicles, the option of higher data rates is limited by a number of network bottlenecks. As the system is half-duplex, transmission and reception cannot occur simultaneously, which requires a lengthy transition between the states. This issue is aggravated by the need for the relay to perform a store and forward operation requiring double the number of state changes as compared to a point-to-point system. Additional delays are inserted as guard intervals to prevent de-synchronization when the radio systems come under heavy data traffic loads. The modulation scheme is binary frequency shift keying, which encodes one bit of data per transmitted symbol. While the radio hardware can be configured to transmit at up to 256 kb/s, a significant decrease in range prevents this ability from being utilized in the scenarios envisioned for use of the quadcopter-based relay node.

3.3 Data Link Layer

The relay network facilitates communication between nodes by acting as a virtual serial port between the ground station and the remote tank unit. The system was designed to be transparent without the need for data users to be aware of the network configuration. Through network abstraction, users simply transmit and receive serial data as though they were directly connected to one another over a wired COM port. Any device with a UART configured to the correct baud rate can utilize the network without issue. This simple transparent setup alleviates the user from needing to implement complex handshaking protocols or time synchronization.

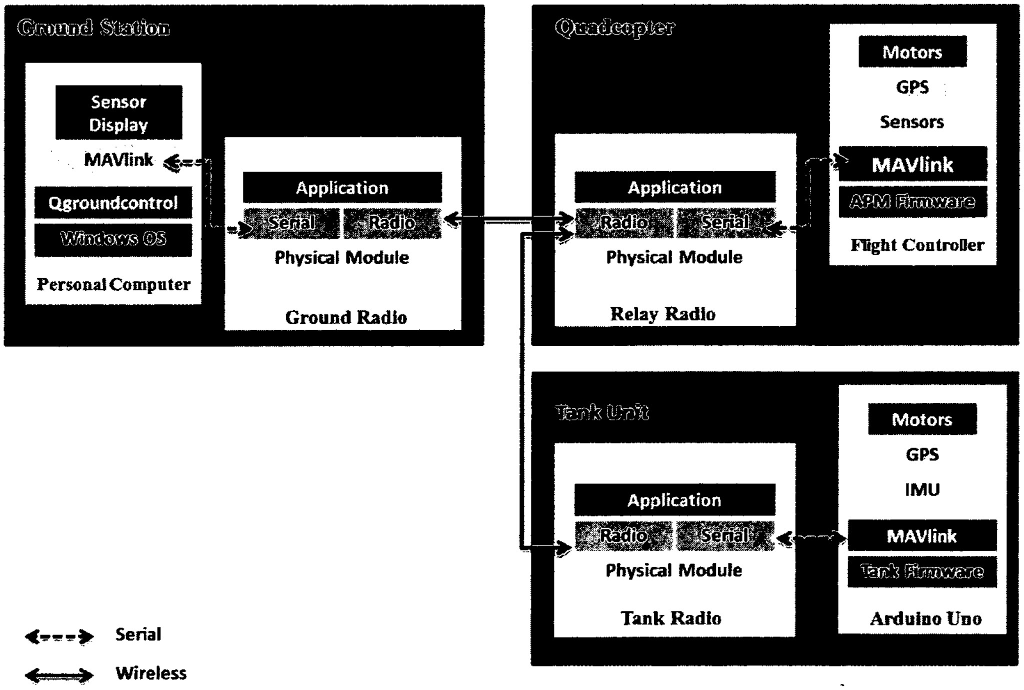


Figure 3.10 - Architecture and Communication Interfaces of IARN

The flow of data works to seamlessly transport information between connected users. Data originating on the ground station is transferred over a communication port to the ground radio UART and copied into internal buffers by an interrupt service routine. Data is then copied on a packet basis from the serial buffers into "first in, first out" (FIFO) buffers on board the integrated transceiver. Bytes stored in the FIFOs are transmitted over the wireless link sequentially and received by the radio relay hardware. As packets arrive at the relay radio front end, data is copied on a byte basis into a local receiver FIFO. The FIFO is then emptied into two large buffers residing onboard the 8051 microcontroller. From here the data is sent to two destinations, the quadcopter platform via the UART and the tank radio over the radio link. Transmission of data from the radio over the UART is accomplished by a second interrupt service routine that transmits serial data while other tasks are performed in the background (buffering received serial data, state transition).

Data intended for the ground station originates onboard the tank or quadcopter unit respectively. While tank data is transmitted wirelessly to the relay, quadcopter information is simply transferred via the connected UART to onboard relay buffers. These two serial streams are merged into one and transmitted to the ground station during the following communication round. This constant flow of data serves user communication needs while relieving them of computational burdens associated with other network topologies.

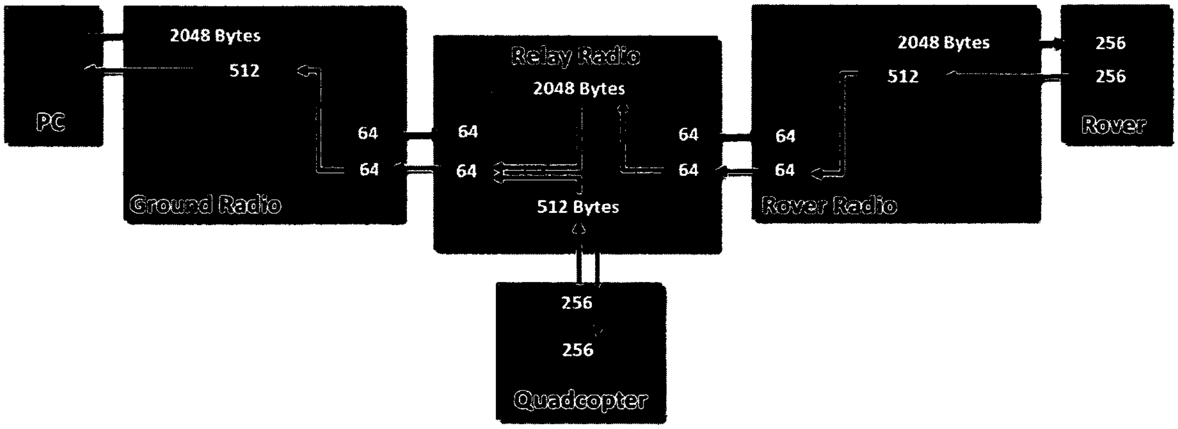


Figure 3.11 – Link Level Data Flow Diagram

Data transmitted over the wireless connection occurs predictably with defined state ordering. Conversely, the data arriving over the UART is random and the connected device is free to generate data packets at any time (refer to figure 3.11 for timing). This creates problems as a radio entering the transmit state is unaware of exactly where in the current packet frame the connected device is. The radio simply saves the received buffer size at the start transmission and sends this payload over the wireless link. The transmission round ends when the current payload has been copied and sent along with the EOT command. Data arriving over the UART during this transmission period is not sent out during this communication round but instead buffered internally.

This simple transmission format can cause issues as a packet can become fragmented when a UART device hasn't sent a complete packet frame but the radio ends its transmission round. While this isn't a problem for a point to point connection (fragmented data is simply stitched together on the receiving side) the nature of the relay's merged relay serial stream can inadvertently inject a message frame within a split packet. This can occur when the relay merges the tank and quadcopter data streams and transmits back to the ground station. The probability of splitting a packet is reduced by utilizing a serial link layer protocol that contains delays between packet frames but transmits single frames all at once. A high baud rate (115,200 b/s) is also utilized to reduce the serial transmission time, further decreasing the chance a radio will end its transmission round part way through a packet frame.

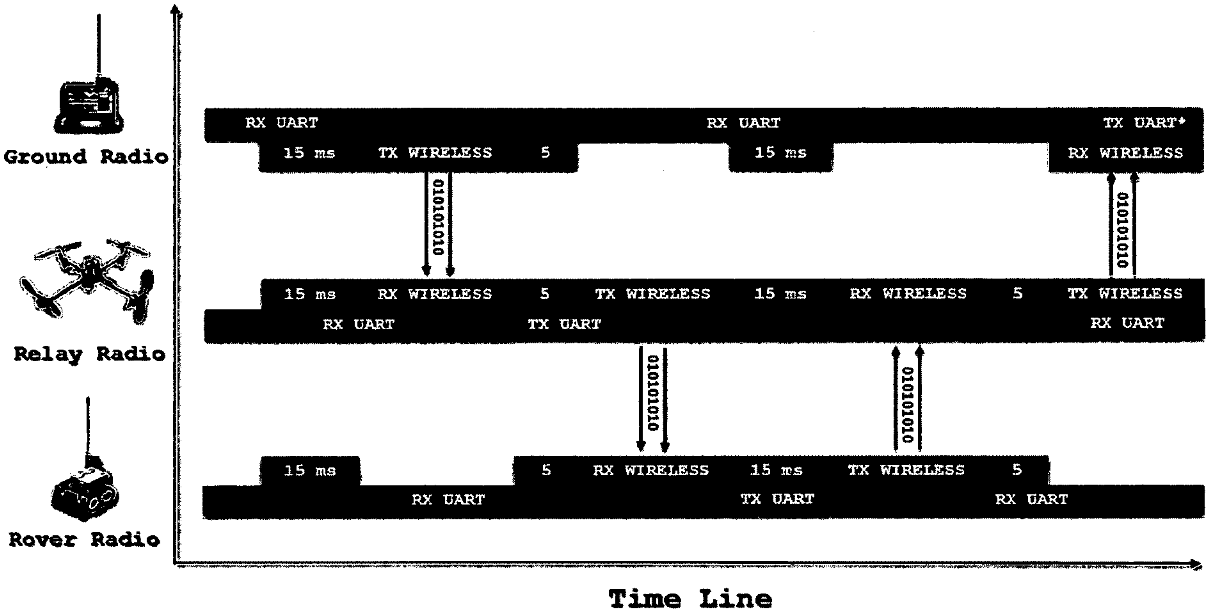


Figure 3.12 – Serial and Wireless Data Transfer Timing Diagram

Figure Description: The figure demonstrates the timing of data transfers and the method of transmission. When a radio is given a turn to transmit all buffered serial data is transmitted. Serial data arriving during this transmission period is buffered internally but does not get transmitted until the next communication round. Transmission of serial data is synchronized with wireless reception. Once a radio has ended the receive mode, all received data is immediately transmitted over the serial port. (* Occurs after RX Wireless)

Whereas the timing of serial reception and wireless transmission is not fully synchronized, wireless reception and serial transmission is. The timing shown in figure 3.12 demonstrates that serial data is transmitted over the UART immediately following a wireless reception. Until the wireless reception round ends, the transmit port remains inactive.

Data flowing from a source to destination must pass through a series of wired (SPI, UART) and wireless (Binary FSK) communication links in addition to undergoing numerous memory copies. When serial data is transferred over a connection or copied between memory locations, execution time is required. This time delay is proportional to the amount of data traffic present on the network. Processing time places a fixed upper limit on the achievable data rate and as well as network latency. The primary sources of network latency are listed in table 3.1 under minimum and maximum payload scenarios. Tracing the flow of data through the network from source to destination allows for the calculation of total system latency. Table 3.2 lists the total time required for data to propagate from one network node to another under minimum and maximum data payloads.

Delay Type	Min Payload (64 Bytes)	Max Payload (2048 Bytes)
t_{UART}	.5 ms	17 ms
t_{SPI}	.005 ms	.16 ms
$t_{wireless}$	1.6 ms	51 ms
t_{copy}	.47 ms	15 ms
t_{guard}	15 ms	15 ms

Table 3.1 – Network Delays under Maximum and Minimum Payloads

Network Route	Min Payload	Max Payload
Ground Station - Tank	26.13 ms	218.32 ms
Ground Station - Quadcopter	18.54 ms	130 ms
Tank - Ground Station	26.13 ms	218.32 ms
Quadcopter - Ground Station	8.54 ms	120 ms

Table 3.2 – Network Latency from Source to Destination

While the relay network provides the vessel upon which data moves between users it does not contain a standard format for serial data. Beyond the need for data to be encoded in bytes, the network does not require a serial protocol at the link level. This is both good and bad, as any serial stream type can propagate through the network without issue. On the other hand, a lack of protocol leaves the network exposed to improper message routing, data corruption and message incompatibility. Without a link level standard, the relay network cannot be considered reliable or safe for connected users. Therefore it is necessary to implement a protocol in order to improve system capabilities and the services provided to users.

MAVlink [7] was chosen as the link level protocol to prevent the aforementioned issues. MAVlink is a powerful packet-based communication standard predominantly used for networks involving semi-autonomous wireless vehicles. Any standard serial or wireless technology which fulfills communication needs can be used with this standard. MAVlink uses a familiar packet frame for data transmission, which contains a number of necessary data fields. The packet anatomy as shown in Figure 3.13 contains a start frame delimiter, payload length, sequence number, sending system ID, component ID, message ID, payload and cyclic redundancy check (CRC). This allows for a number of critical protocol features such as error detection, message rejection, packet parsing and dropped packet discovery. A significant boost in system reliability and safety is achieved through this protocol with little to no loss in transmission efficiency.

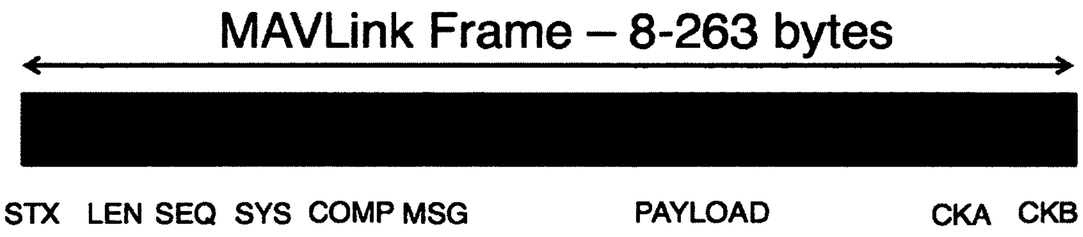


Figure 3.13 - Mavlink Packet Frame Anatomy

(Figure Credit: Qgroundcontrol.org [7])

MAVlink is typically used in scenarios where multiple autonomous vehicles are reporting their status in parallel to a main subscriber application. Each vehicle's communication software contains an embedded MAVlink protocol layer with the ability to encode/decode messages in the serial stream. The subscriber application handles parsing of received vehicle data packets and formation of system commands. Enforcing the MAVlink protocol on all connected devices enables a series of benefits for the network beyond simply safety. Vehicles can reject improperly routed messages and the subscriber application can distinguish between multiple vehicles on the network. This is extremely important as up to 255 devices can be connected concurrently and therefore the ability to resolve multiple systems is absolutely critical.

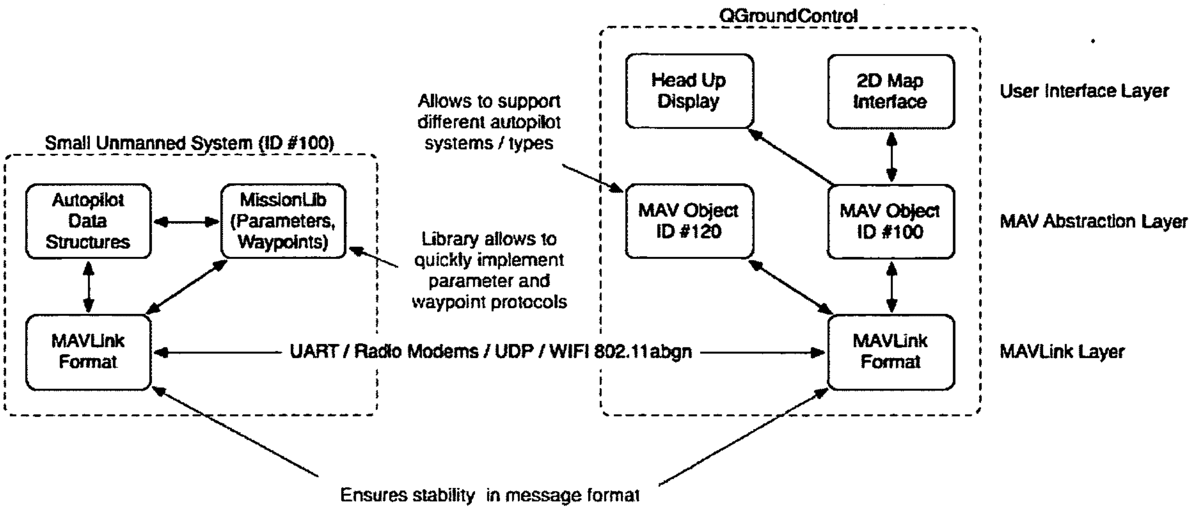


Figure 3.14 - Typical MAVlink Communication Scenario

(Figure Credit: Qgroundcontrol.org [7])

MAVlink is highly configurable and compatible with many embedded controllers and vehicle types. The protocol comes with a built-in library of commonly used commands and messages such as GPS and IMU reporting as well as waypoint, motor arming and manual commands. The existing messages also come with encoding/decoding functions allowing for easy generation of packets and access to data fields of received packets. Users can opt to utilize the stock message frames for their applications or extend the common library by generating additional custom messages. This feature lends to easy adaptation of MAVlink to work with a wide range of unique vehicles and system components.

The tank and quadcopter contain a number of integrated sensors and micro-devices that provide a range of information on each platform's current condition. In order to reduce bandwidth requirements, only a subset of these integrated systems was selected for MAVlink reporting. Global position, orientation and network activity provided the metrics upon which to interpret a vehicle's current condition. These messages were chosen to be sampled and transmitted back to the ground station for processing. Careful selection of transmitted packets minimizes network use while providing desired system indicators.

Retaining a high level of radio link quality is a predominant objective of the relay network. Knowledge of the past and present link statistics is a metric upon which to judge network quality. It was necessary to provide the framework for monitoring link parameters and generating radio reports pertaining to the current network conditions. This reporting was accomplished by extending the standard MAVlink library to include a custom radio message. Two major steps were required to extend MAVlink and enable transmission/reception of radio report packets. First, a message definition was created in the extended markup language (XML) to describe various components of the message such as name, payload, variable data types and message ID. From this XML file a series of C header files could be generated. These files were then included onboard all systems involved in encoding and decoding these reports.

Figure 3.15 - Radio Report Payload Anatomy

Radio reports are formed onboard the tank and ground station transceivers and transmitted back to the base application. Since MAVlink frames are used to assemble these packets, the existing network and serial data stream can be utilized to return data to the base application. The payload of these generated messages contains pertinent link statistics such as received signal strength, receiver errors, buffer overflows, link drops, background noise and current hopping channel. By monitoring these values, a measure of the network's reliability and proximity to disconnection can be obtained.

Since the MAVlink protocol does not contain an address for the destination system in every packet, routing data to the correct user can become an issue. Without knowledge of all possible Mavlink messages, the relay radio has no way of parsing packets intended for the quadcopter versus the tank. Fortunately the end user (tank and quadcopter) is aware of the acceptable MAVlink messages and can ignore any incorrectly routed messages. In an effort to improve the reliability of the radio network and decrease system traffic load, broadcast routing was implemented. This frees the radio from needing to persistently monitor the serial stream and decide where packets are destined in the network.

As data arrives at the ground station, the returned packets are parsed and examined. Each vehicle connected to the ground station via the relay link is instantiated by a class of related functions and data structures. The data field corresponding to the system ID is used to associate received packets with the correct vehicle class. The message ID provides another level of detail to determine the payload type and choose which data structures to update within a class. The MAVlink library contains a number of decoding functions to recover particular fields of a packet retrieved from the serial stream, which allows easy access to message data. This structure of data on the ground station side offers a powerful method of representing vehicles and keeping their associated information organized.

3.4 Control and Data Logging Support

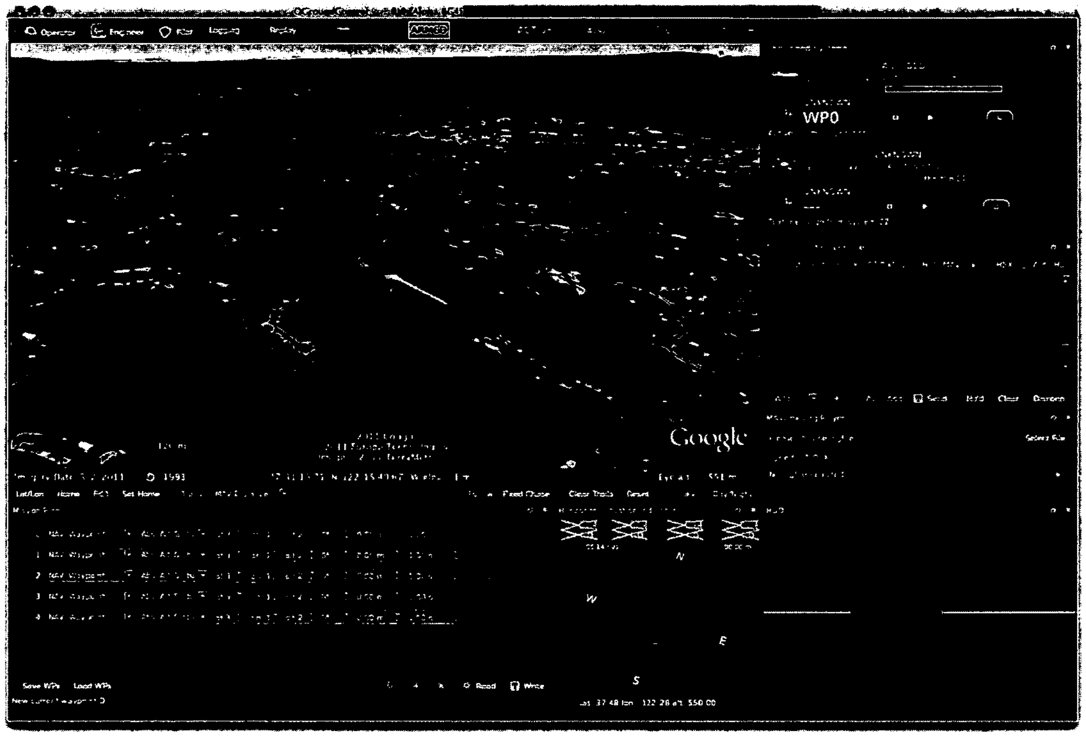
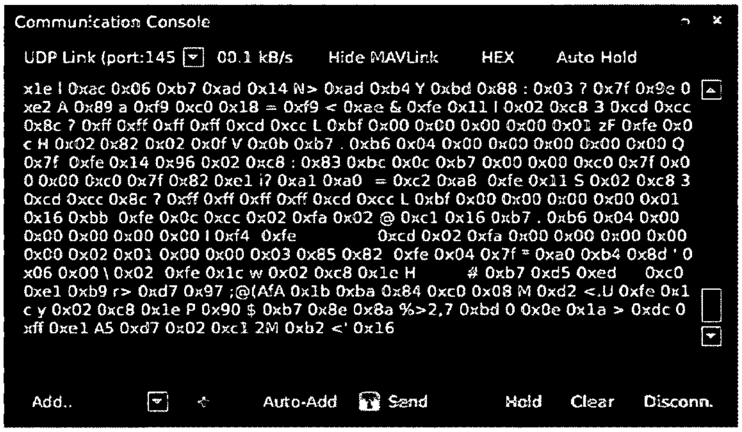


Figure 3.16 - Graphical User Interface of Ground Station (Qgroundcontrol)

Qgroundcontrol is a powerful open source application for control and processing of autonomous vehicle systems. Data received from vehicles over a serial link is parsed on a user by user basis and displayed in various useful ways. Qgroundcontrol provides a powerful combination of low level access to data on the communication link in addition to high level visual representation and modeling. For example, orientation data and global position can be collected from a vehicle's inertial measurement unit (IMU) or global positioning system. This, in turn, is used to update instruments such as a three dimensional heads up display and vehicle location on a digital map. Qgroundcontrol facilitates the user's understanding of their vehicles' current condition, thus significantly improving the safety of unmanned systems. Received packets can also be logged in real time and stored locally for future replay. Vehicle commands such as waypoints, motor arming and calibration can be generated from within the application allowing for full control over connected systems. Qgroundcontrol uses MAVlink as the link level protocol for increased reliability and vehicle safety. Utilizing the existing radio network, commands, vehicle data and radio reports can all propagate simultaneously.



```
x1e l 0xac 0x06 0xb7 0xad 0x14 N> 0xad 0xb4 Y 0xbd 0x88 : 0x03 ? 0x7f 0x9e 0
xe2 A 0x89 a 0xf9 0xc0 0x18 = 0xf9 < 0xae & 0xfe 0x11 l 0x02 0xc8 3 0xcd 0xcc
0x8c ? 0xff 0xff 0xff 0xff 0xcd 0xcc L 0xbf 0x00 0x00 0x00 0x00 0x01 zF 0xfe 0x0
c H 0x02 0x82 0x02 0x0f V 0x0b 0xb7 . 0xb6 0x04 0x00 0x00 0x00 0x00 0x00 Q
0x7f  0xfe 0x14 0x96 0x02 0xc8 : 0x83 0xbc 0x0c 0xb7 0x00 0x00 0xc0 0x7f 0x0
0 0x00 0xc0 0x7f 0x82 0xe1 i? 0xa1 0xa0  = 0xc2 0xa8  0xfe 0x11 S 0x02 0xc8 3
0xcd 0xcc 0x8c ? 0xff 0xff 0xff 0xff 0xcd 0xcc L 0xbf 0x00 0x00 0x00 0x00 0x01
0x16 0xbb  0xfe 0x0c 0xcc 0x02 0xfa 0x02 @ 0xc1 0x16 0xb7 . 0xb6 0x04 0x00
0x00 0x00 0x00 0x00 l 0xf4  0xfe            0xcd 0x02 0xfa 0x00 0x00 0x00 0x00
0x00 0x02 0x01 0x00 0x00 0x03 0x85 0x82  0xfe 0x04 0x7f " 0xa0 0xb4 0x8d ' 0
x06 0x00 \ 0x02  0xfe 0x1c w 0x02 0xc8 0x1e H         # 0xb7 0xd5 0xed     0xc0
0xe1 0xb9 r> 0xd7 0x97 ;@(AfA 0x1b 0xba 0x84 0xc0 0x08 M 0xd2 <.U 0xfe 0x1
c y 0x02 0xc8 0x1e P 0x90 $ 0xb7 0x8e 0x8a %>2,7 0xbd 0 0x0e 0x1a > 0xdc 0
xff 0xe1 AS 0xd7 0x02 0xc1 2M 0xb2 <' 0x16
```

Figure 3.17 - Low Level Communication Console

Figure Description: The communication console provides low level access to the received/transmitted serial stream. Users can transmit ASCII characters via the terminal and also view basic debug information such as data rate.

User code can be generated and injected into the application allowing for the inclusion of additional features. As Qgroundcontrol is fully compatible with MAVlink, custom radio report messages were included along with the prepackaged Mavlink library. This allows for viewing and even logging of link statistics in real time. Both the tank and ground radio link information can be viewed simultaneously, thus providing powerful insight on the link's current health, SNR balance and proximity to disconnection. Global positioning data can used to determine distances between radios, thus affording comparison of node distances to link health.

3.5 Dynamic Relay Positioning

While the propagation environment remains static (i.e. users remain in place, the surrounding geometry and multipath structure are fixed) the existing relay framework is adequate to sustain a quality communication link. Mobile users will experience a range of network performance as they traverse the environment. Variation in atmospheric conditions and the environmental setting play a significant role in determining network performance. These actions over time will cause imbalance in the communication link quality or even disconnection of users. If a static relay is employed for communication purposes, network performance will be determined solely by these external influences (e.g. atmospheric conditions, communication environment and distance to users).

Figure 3.18 - Dynamic Relay Positioning based on Link Statistics

It is hypothesized that a relay platform capable of intelligent positioning would retain a consistently balanced communication link even under time varying environmental conditions. Network conditions and vehicle locations could be monitored; coordinates could be generated and transmitted to the relay creating a closed loop control system. The devices connected via the relay network provide a wealth of information on vehicle position, orientation and link quality. While Qgroundcontrol by default only contains the ability to display this data in useful ways, additional code can be added to further extend the application. By encapsulating this user code within the Qgroundcontrol source, code level access to received data is gained and signal processing can take place. This lends itself to the exciting possibility of autonomous decision-making based on the current state of vehicles and the quality of the links. Commands can be generated by the ground station and transmitted via the network to linked vehicles, thus effectively creating a closed loop position control system. With proper processing of received data, decision making and formation of commands it is possible to create a relay positioning algorithm within the existing network framework.

In an effort to preserve the connectivity of the relay network with time-varying channel parameters, a simple positioning algorithm was developed. The algorithm monitors the received signal strength indicator data reported by the ground station and tank unit radios. From these reports the SNR balance of the link and absolute received power can be determined. Additionally the global position can be obtained from both the quadcopter and tank in real time, which allows for distance estimation between all transceiver nodes. The RSSI coupled with location data provides important metrics upon which to judge the links' quality and the end users' proximity to disconnection. As the quadcopter is moved about the network environment, a scanning subroutine samples the signal strength reported by each user and attaches a global position. Samples are only recorded when at least 500 ms have elapsed and the quadcopter has moved a distance of three meters. This provides enough spatial and temporal variation during the scan to properly characterize the signal strength over an area of the network environment.

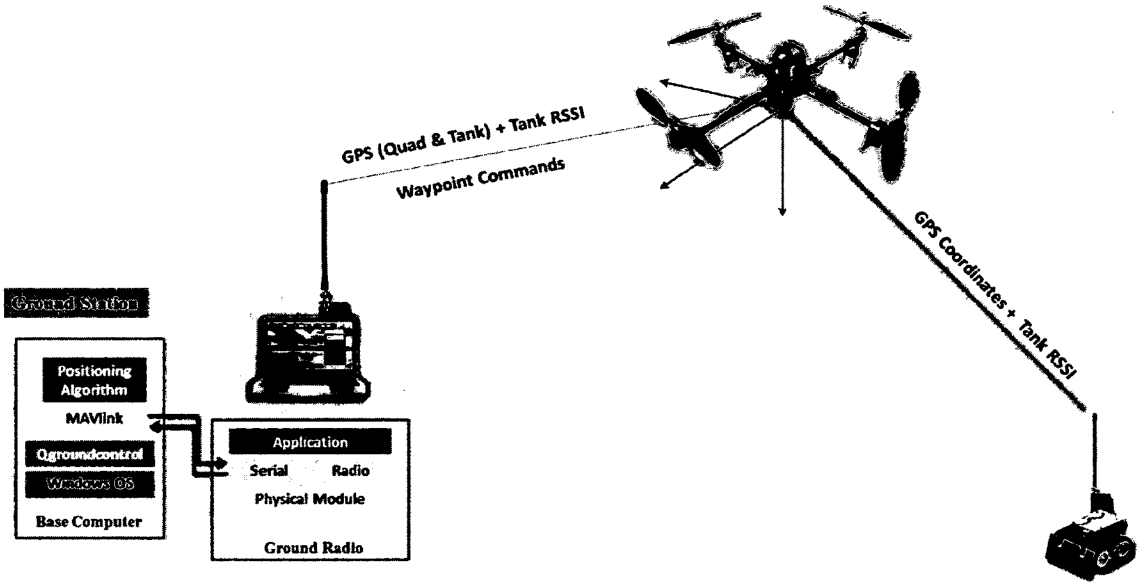


Figure 3.19 - Proposed Position Algorithm Framework

Figure Description: The proposed position algorithm is designed for easy integration with the existing communication protocol and Qgroundcontrol framework. Access to vehicle data is achieved through the decoding of MAVlink packets returned through the serial data stream. After processing of said values, the algorithm produces global relay coordinates which are transmitted to the quadcopter's flight controller.

After the scanning procedure concludes the resulting information is processed to determine the highest average RSSI pair (one RSSI value for each side of the link) and the attached position index. From the recorded data the corresponding position that maximizes signal strength can be obtained by simply passing the calculated index to the latitude/longitude array. The resulting latitude/longitude pair is then packaged into a command intended for the quadcopter flight controller. This message is decoded onboard the quadcopter and the route is flown by the vehicle. Upon reaching this new destination the quadcopter loiters about this point while the link quality is monitored for fluctuation.

If the link quality decreases below a predetermined threshold, action must be taken in order to maintain the link. Since the connection between the quadcopter and ground station is critical for safety and command flow, preservation of this end of the link is weighted more heavily. By setting the acceptable received signal level threshold higher than that of the tank unit, the algorithm favors preserving the critical link. When thresholds are crossed due to fluctuation in link signal strength, the algorithm must determine the best course of action to preserve the link.

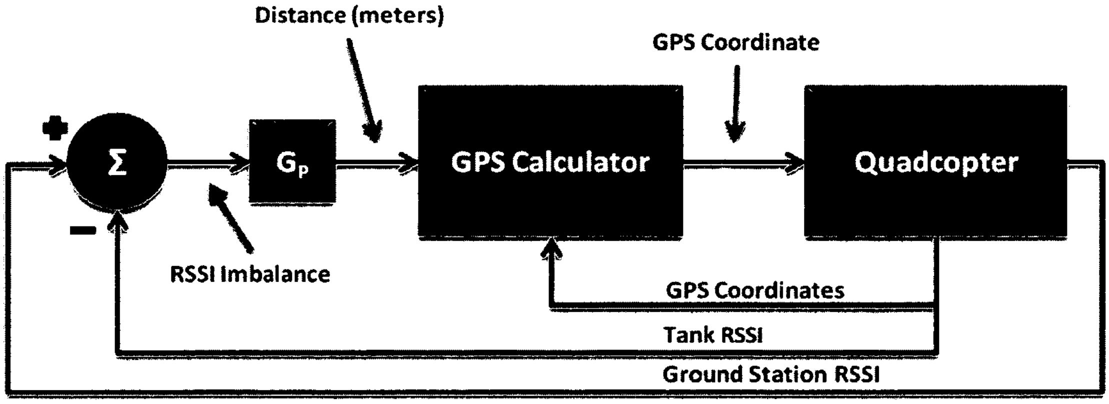


Figure 3.0-20 - Proportional Relay Position Controller

Figure Description: The proportional controller is utilized while the link imbalance remains within a reasonable margin. The calculated distance to the next GPS waypoint is proportional to the magnitude of the link imbalance. The bearing of the destination waypoint is based on the angle between the current relay location and the end user with the lower signal strength.

While the network link quality is balanced, the quadcopter is instructed to remain about its current position for an indefinite period of time. As the connection quality becomes biased toward one end due to environmental alterations or user movement, the quadcopter must quickly react to prevent a lost link. When the link is biased toward one end by a certain margin, a simple proportional feedback controller is used to shift the quadcopter's position between the two vehicles. The quadcopter's shift distance is proportional to the imbalance in the link thus, allowing for dynamic relocation under time varying link conditions. When the imbalance is deemed too large and disconnection is imminent, the quadcopter is quickly relocated to the current geographic midpoint calculated from user GPS coordinates. The scan subroutine is once again executed and in an effort to determine the new link equilibrium location about this midpoint and the proportional controller is enforced. This process continues while it is necessary to retain a network connection and the quadcopter remains capable of flight (battery depletion being the primary concern).

The rate at which the network conditions vary and by what amount are critical factors in determining the effectiveness of an algorithmic positioning system. If the closed loop positioning system cannot sample network conditions, calculate a position, transmit commands and relocate the relay in a timely manner, the controller will be unable to respond quick enough to maintain signal conditions. As the controller must make decisions based off sampled data which propagates over the network, there is inherent latency in feedback and command output. GPS data for example arrives every 200 ms while RSSI data is approximately 65 ms. Network latency can also generate delays up to 218 ms for all sampled data to be transmitted and returned to the ground station. Transmitting a command back to the quadcopter also requires approximately 18.54 ms before the flight controller will obtain the packet. The quadcopter also cannot move at an unlimited rate and therefore requires a reasonable time to move from location to location (max speed set to 10 m/s). While the control loop speed will vary due to network conditions, it takes more than 430 ms for the control system to monitor conditions and produce output. The controller will therefore only be effective in scenarios where the signal conditions variation occurs slowly.

Chapter 4

EXPERIMENTAL VALIDATION

4.1 Preliminary Lab Tests

Purpose of Experiments

Before moving the relay and related platforms outside the laboratory environment it was necessary to characterize the link quality and compatibility of systems under idealistic conditions. The link reporting capabilities were tested and verified for correct functionality. Radios needed to be able to power up or down while still automatically acquiring one another. A series of tests were performed to guarantee the radios abilities to acquire once outside the confines of the lab. This would provide a gauge on expected performance and a metric to determine whether further improvements were necessary. It is desired to determine the expected error rates, send/receive serial capabilities, data rate and re/acquisition under processor load.

Acquisition Test

Purpose of Experiment

For the radios to function properly outside of the lab when disconnected from the debugging equipment, an acquisition phase was developed. Radios need to be capable of automatically detecting and acquiring one another in any possible power-up sequence order.

Experimental Setup

Three networked radios were programmed with acquisition instructions and powered on and off in all possible orders.

Results:

While the systems did connect properly a number of times, there were enough inconsistencies observed to warrant further investigation. In a number of instances the radios were unable to connect to one another after power was applied. Utilizing debug units, the acquisition code was run and breakpoints were set at critical locations to test for issues. It was quickly determined that radios were receiving packets in error during this initial synchronization. This caused the radio hardware to consider the link broken because retrieved packets didn't correspond to the expected message. After careful examination of the radio startup code it was determined the radio hardware was improperly initialized, thus causing intermittent errors and unstable operation.

This issue was resolved by adding a series of radio configuration functions to properly initialize the hardware and switch between states. Once these upgrades were applied to each radio the acquisition test was run once more. Radios were booted up and powered down in all possible sequence orders to verify that automatic acquisition was functioning. The radios performed well, acquiring one another quickly in the majority of cases. However, under circumstances where the host device transferred a large amount of serial data to the radio during the acquisition phase, synchronization became an issue. The delicate timing needed for radio synchronization was disturbed by the inrush of serial data which caused latency between systems. This problem was resolved fixed by disabling serial interrupts until after the link was up and running properly. After this final issue was addressed, the radios performed perfectly in the lab, finding and connecting with one another.

Serial Data Transmission Test

Purpose of Experiment

Network quality is of paramount concern as the reliability and safety of users rests on the shoulders of this connection. Before utilizing the network in more demanding situations the relay serial transmission capabilities were thoroughly tested.

Experimental Setup

The end radios which would eventually serve the tank and ground station were connected to computer serial COM ports. Onboard the host computer a basic serial terminal application was used to view received data and transmit user characters. The relay radio was connected to a separate communication COM port in the same fashion. This configuration allowed for data to be transmitted or received over each serial terminal to check for any corruption in the serial stream and relay operation.

Experimental Procedure

A number of variable length character patterns were transmitted from each terminal and viewed at the receiving COM port.

Results

When packets fewer than sixty four bytes in length were generated, the network performed admirably with data flowing to the correct users while remaining fully intact. There were no apparent issues with state timing, data routing or packet corruption gathered from this test. As users may require the transmission of large blocks of data well over sixty four bytes in length, further testing was necessary to deem the network fully functional. A series of large data sets were compiled and transmitted through the serial terminal. Upon viewing the serial output, it became apparent that data was being improperly handled by the radio link.

At a maximum, sixty four bytes were transferred via the link regardless of the original payload size. As the link remained synched and there were no detectable packet losses, this indicated that somewhere along the link a transmission round was ending prematurely. After running each radio application concurrently with the aid of the software development environment it became obvious the relay was ignoring data. This was caused by a software flag being overwritten. This resulted in a termination of the transmission round before all residual data in the transmit buffer had been sent. Once this flag was moved to a safe memory location, further tests demonstrated that large blocks of data could propagate through the network intact.

Network Synchronization under Load

Purpose of Experiment

At this point it had been confirmed that radios were exchanging data, remaining synchronized and properly acquiring one another under light network conditions (e.g. only one user transmitting a serial stream at a time). It was yet to be seen how well the network functions under a heavy user load. A guarantee that radios remain synchronized was critical before the relay link could be fully utilized

Experimental Setup

The ground station application was executed onboard the host PC and connected to the ground radio. The quadcopter and ground rover were powered on and a wireless connection was established between all units.

Experimental Procedure

For the first test the data streams generated onboard each vehicle platform were reduced to only generate the most vital messages. This was accomplished on board the tank's microcontroller by removing IMU and GPS reporting while allowing radio reports and tank heartbeat messages to be sent. The quadcopter platform contains a highly configurable range of serial streams that can be altered from the ground station or programmed into system EEPROM. This creates a great deal of variability in required bandwidth for the quadcopter platform. All but the most basic serial streams were turned off in this first test to reduce system load.

Once each platform was configured to decrease their individual network usage, radios were attached and the link was established. Qgroundcontrol performed the task of allowing low level access to received serial data returned by the link in addition to high level visual representation. From this multi-level view of data, the link capabilities could readily be assessed. While network traffic remained relatively light there was no indication of system errors. Radios remained synchronized and all chosen messages arrived at the ground terminal without issue. Through the Qgroundcontrol application the number of serial data streams emanating from the quadcopter was increased for the quadcopter to determine if the link remained connected under increased data traffic. While the utilized data bandwidth was increased incrementally, status LEDs and heartbeat messages were monitored to verify radio activity. The received instantaneous serial data rate was calculated at the ground station COM port, providing a measure of the merged quadcopter and tank data traffic.

As the data rate of the merged serial stream (quadcopter and tank transmissions) approached 1 kb/s at the ground station, link conditions quickly deteriorated and resulted in a series of disconnects. At this point the network became unreliable and data flowed sporadically between units. Radios were connected to debug units and the data traffic load was once again increased to re-create this behavior. After running the radios under debugger control it became apparent that large transfers of data required a significant number of memory copies and the execution of interrupt service routines. As the link data rate was increased, this causes a noticeable postponement of normal system operation. If this delay is great enough, state mismatches can occur, or even network timeouts, which render the link inactive. Fortunately, the solution to these issues was as straightforward as adding guard intervals between receive to transmit state changes on all radios. This simple addition provides a tolerance to time shifts between radio applications brought on by an increased data load.

Once again, the radio platforms were connected to their respective users and synchronized for data transfer. The number of serial streams was ramped up incrementally while the network was monitored for reliability. When the received data rate at the ground station reached 5 kb/s, the link quality began to exhibit similar symptoms of failure. While a higher data rate could be achieved by configuring the modulation scheme with a higher symbol rate, this was an acceptable network bandwidth to support all user needs. At this point, the combined serial data rate ceiling was set to 5 kb/s and users (i.e. individual traffic generators) were configured properly to conform to this limitation. Further tests demonstrated an extremely robust connection while data rates remained below this level, proving the utility of the designed network.

User Compatibility and MAVlink detection

Purpose of Experiment

While the previous experiments demonstrated the network's physical layer functionality, it was uncertain whether users were compatible with one another and able to decode/encode messages properly. Therefore, it was necessary to transmit a series of messages from users and check whether these packets were decoded without error at the ground station.

Experimental Setup

Each platform was linked together over the existing network connection and vehicles were configured to produce a series of MAVlink messages. Qgroundcontrol ran onboard the host PC to provide necessary low level access to packets and visual interpretation of incoming data.

Experimental Procedure

It was desired to determine if the ground station was properly detecting active users and decoding their respective message payloads. The quadcopter was configured to provide a single serial stream back to the ground station. Packets arriving at the ground station were examined for frame completeness and data corruption. The tank was then configured to provide another source of MAVlink packets for viewing and comparison. These serial streams were merged onboard the relay and parsed at the ground station into the proper vehicle classes. A series of commands were transmitted to the quadcopter to verify response and bidirectional serial data flow.

Results

As each vehicle was brought online and transmission began, Qgroundcontrol responded accordingly. A vehicle class was generated for each separate device on the network and corresponding packets were routed to this data structure. As a vehicle's position or orientation changed, Qgroundcontrol correctly updated the equivalent instruments within each application. When a vehicle was disconnected from the network, a proper warning message was generated to notify the user of connection loss. Commands transmitted from the ground station invoked a proper response from the quadcopter. These results show that the MAVlink protocol was implemented correctly onboard each respective platform and it is now possible to monitor, record and command vehicles from within Qgroundcontrol.

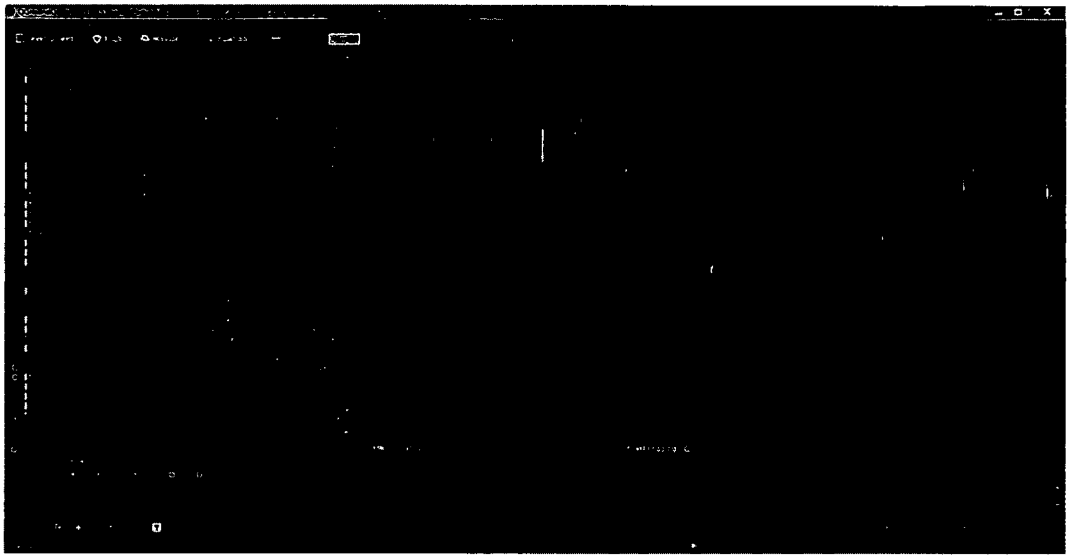


Figure 4.1 - Display of Received User Data (IMU Gyro output vs. time)

Radio Report Detection and Logging

Purpose of Experiment

In order to properly monitor the network health, the ground station must be able to perceive MAVlink radio reports generated by the tank and ground station radios. Furthermore, assurance of radio link statistics accuracy was critical before future experiments could take place.

Experimental Setup

Three radios were configured to implement the relay network without any vehicle-generated data streams. The tank radio was programmed to transmit its respective link statistics report over the network through the existing relay connection. The ground radio was configured to report statistics over the local serial connection directly to the ground station. The radios were placed in close proximity to one another on the laboratory desk. Qgroundcontrol was run onboard the host PC and connected to the link through the ground radio.

Experimental Procedure

Each radio was powered on and synchronized into the network while Qgroundcontrol monitored the radio link statistic serial stream. As data arrived at the ground station it was checked for validity and the correct system response. Once data had been received, payloads were logged from within Qgroundcontrol and exported to a comma delimited (CSV) file format. Radios were then moved apart and link parameters were monitored for variation.

Results

With radios connected and reporting link statistics back to the ground station, a steady stream of serial data was seen in the communication terminal console. These character patterns were verified as the correct packets according to the original radio report message definition. Qgroundcontrol was nonresponsive to this incoming data beyond simply displaying strings of characters within the debug console. This anomaly was caused by the omission of a heartbeat packet that needed to be sent periodically in addition to the radio report packets. The heartbeat message alerts Qgroundcontrol to the presence of a connected device. Without this message Qgroundcontrol will disregard any data not attached to an active platform.

Once a “heartbeat” message was generated onboard each radio, Qgroundcontrol responded by declaring a vehicle class. A response was noted for the heartbeat. The radio report was still ignored. This issue was caused by the improper specification of a CRC generator polynomial onboard each radio system. As the ground station performs a CRC on each received data packet, packets are simply dropped upon failing. Once the generator polynomial was updated with correct values a notable change took place within Qgroundcontrol. Incoming data now populated a real time plot and radio statistics could be monitored as numerical values. This was a very promising result because reported radio link data was now being properly interpreted and displayed by Qgroundcontrol, thus allowing data logging and post-processing.

The received signal strength (RSSI) values of the tank and ground radios were monitored as radios were moved about the room. While a slight variation in signal strength occurred as separation distance increased, there were inconstancies in the magnitude of returned values. When operating under short range, the RSSI encoder should theoretically return 200-250 counts (the maximum value being 255 due to the 8 bit resolution of the RSSI register). The current configuration only displayed 100 counts. Upon further inspection of the radio firmware it became clear that an issue with signal strength sample timing was to blame. The sampling routine was moved from its present location in code execution into an interrupt service routine called during the packet preamble. The signal strength sample time now occurs within a valid packet frame while RF energy is present at the radio’s front end and provides an accurate depiction of present signal strength.

Early Outdoor Tests without Quadcopter Flight

Purpose of Experiment

With the network functioning under laboratory conditions it was now possible to move experiments outside. This series of tests was designed to investigate the radio's statistics monitoring abilities in a true operational environment. Variation of received signal strength with distance to a source was of great interest as this would indicate link sensitivity to user movement. Determining the accuracy of vehicle sensor output was required before future experiments could utilize their produced results. GPS and magnetometer readings needed an outdoor environment to work properly and therefore had not been fully examined as of this point. A rigorous test of system data logging within Qgroundcontrol was needed to detect any formatting or data corruption issues. These explorations provided a characterization of the link's true operating capabilities, signal quality and achievable distance range, which would allow future investigations to be productive.

Experimental Setup

The tank and ground station were placed adjacent to each other on a support at an altitude of 2 meters above the ground. The Quadcopter was placed at a nearby location as depicted in Figure 4.2. The network connection was established and all returned data was selected for logging within Qgroundcontrol.

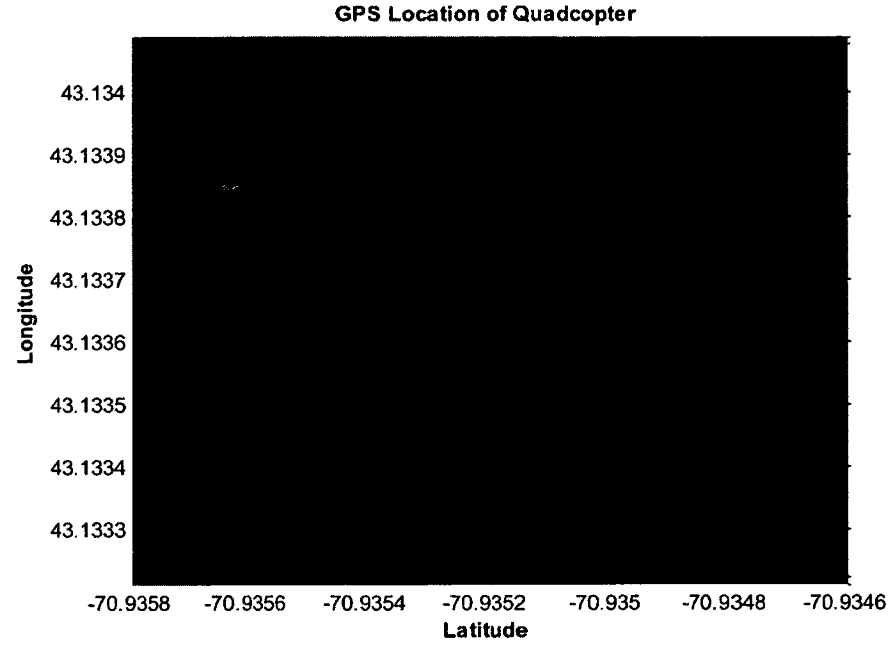


Figure 4.2 - Position of Vehicles during Early Outdoor Experiment

Figure Description: Green X corresponds to tank and ground station location, Blue X demonstrates the quadcopter's start location and while the black X is the end position. Location of Experiment is South side of Kingsbury Hall adjacent to McDaniel Drive.

Experiment Procedure

With Qgroundcontrol sampling and logging all returned MAVlink packets, the experiment was started with all units sitting in close proximity to one another. The tank and ground station remained fixed in place for the duration of the experiment while the quadcopter was moved through the environment. A straight path was traversed with the quadcopter starting at the ground station and ending 70 meters away. Upon reaching its final destination the quadcopter was placed on the ground and data logging was halted. Exporting the payload fields of logged MAVlink packets into a CSV file format provided access to the embedded data. The resulting values were thoroughly inspected for validity to ensure that the network and vehicles had functioned properly. A decoding script was developed in Matlab and used to interpret and process the exported Qgroundcontrol log to provide critical feedback on experiments.

Results

Time Stamp	System ID	Data Title	Value
0	84	M84:radio_report.rxerrors	4
0	84	M84:radio_report.rxoverflow	0
0	84	M84:radio_report.txoverflow	0
0	84	M84:radio_report.txbuf	0
0	84	M84:radio_report.linklost	255
0	84	M84:radio_report.noise	78
0	84	M84:radio_report.channel	30
16	71	M71:HEARTBEAT.base_mode	0
16	71	M71:HEARTBEAT.custom_mode	0
16	71	M71:HEARTBEAT.system_status	3
16	71	M71:radio_report.rssi	114

Figure 4.3 – Qgroundcontrol Log File from Early Outdoor Experiment

Figure Description: Vehicle IDs allowed data to be sorted according to platform, titles were used to automatically create name fields of structs and associate numerical results with their corresponding label. Time stamps provided millisecond accuracy for sampled values.

A number of useful results were provided by this simple investigation. Qgroundcontrol proved to be a fantastic application for data logging with no apparent formatting or decoding issues. All selected data types were recorded, labeled and given a time stamp according to the layout depicted in figure 4.3. This demonstrated that the relay link was functioning properly and that data was being transferred as expected. Upon inspection of the raw quadcopter GPS data, it was observed that its spatial resolution was much greater than initially expected based on published accuracies of (civilian accessible) GPS units. A movement of as little as two meters provided enough translational displacement for the quadcopter to detect and update its position. This result is very promising as coupling between vehicle position and received signal strength may be strong enough to make feasible an automatic positioning algorithm. Furthermore, a radio link balancing algorithm supplemented by GPS data is facilitated by this level of location accuracy.

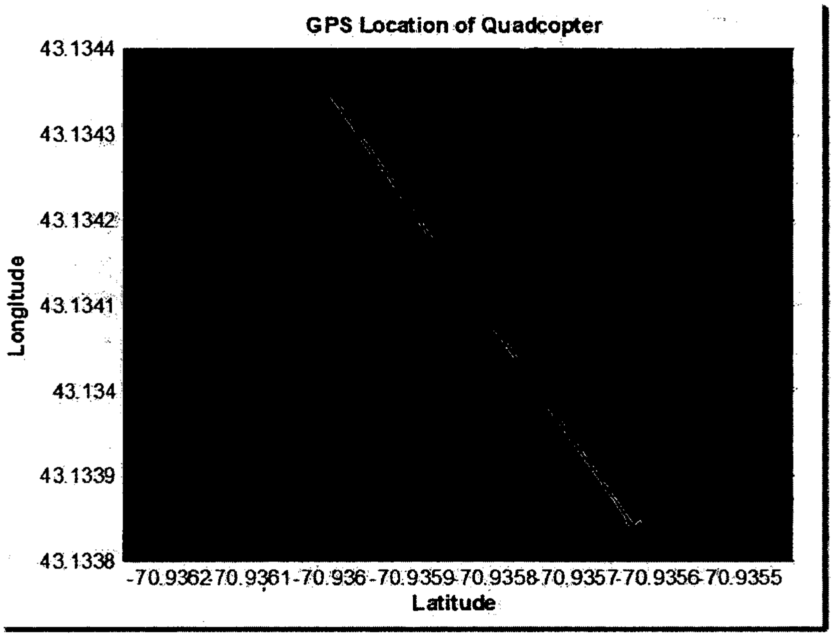


Figure 4.4 - Example of High Accuracy Global Position Information

Figure Description: Sub 3-meter horizontal accuracy of the quadcopter afforded by the MediaTek-3329 GPS module at 10 Hz sampling rate.

The quadcopter was shifted away from the tank and ground unit as the experiment progressed, thus creating a mostly linearly increasing separation between network radios. The resulting RSSI data showed a steady decrease in received signal strength as time passed. This correlated well with the resulting signal strength versus vehicle global position.

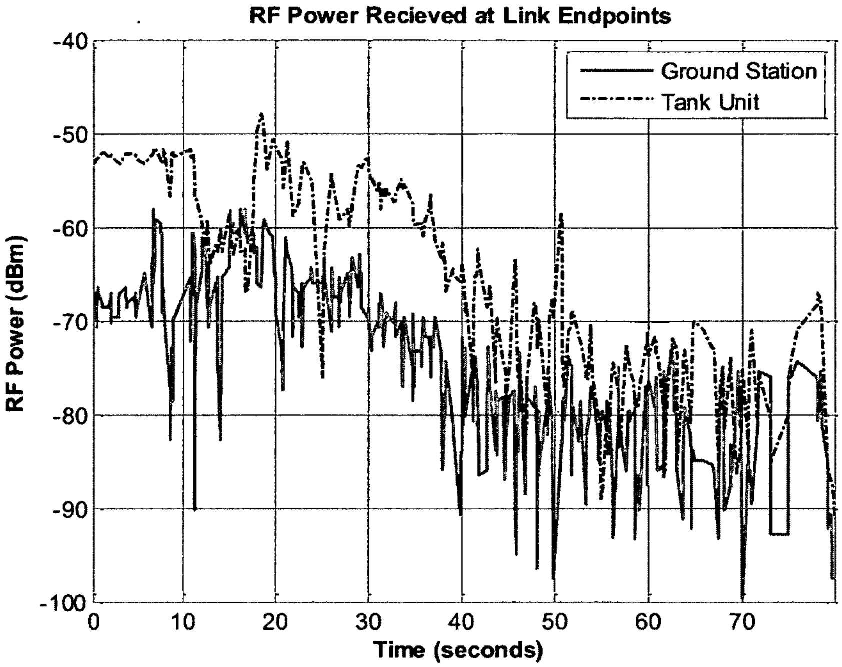


Figure 4.5 - Received Power (dBm) Level at Both Link Endpoints

Figure Description: The absolute RF power derived from the RSSI encoder count at the ground station and tank unit is displayed in the above figure.

As the quadcopter was moved further away from the other units, the received signal steadily decreased. The rate at which the signal strength dropped off also demonstrated a high level of sensitivity to translational movement. While the total signal variation is caused by multipath propagation and antenna orientation effects, the general trend is the result of increased transmission range. The average slope of this RSSI graph indicated a movement of as little as 1.38 meters is enough to produce a detectable variation in signal strength. While influences such as multipath, refraction and environmental obstacles could create significant variation in signal strength even without radio platform movement, this result shows promise for use of RSSI data in an automated quadcopter relay positioning system.

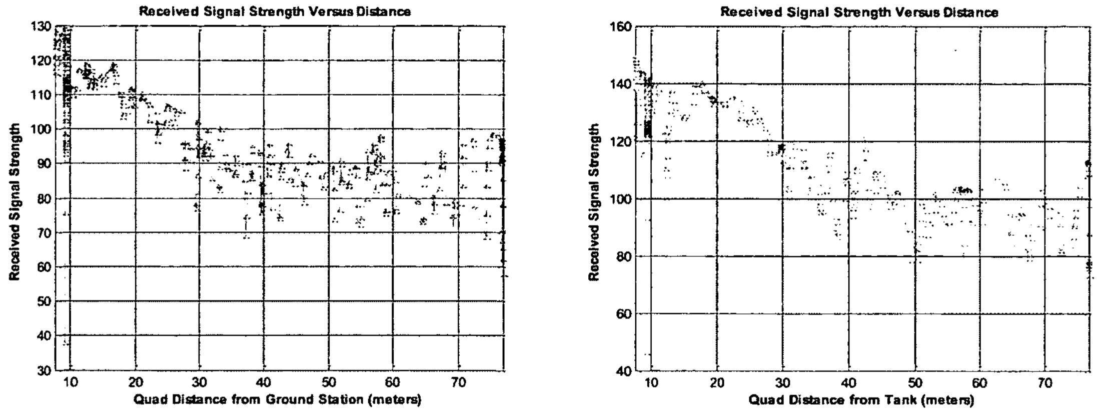


Figure 4.6 - Raw Signal Strength Encoder Count vs. Distance to Relay

Figure Description: With only 8 bits of RSSI encoder resolution, minute horizontal movement resulted in a detectable signal strength variation. Ripple demonstrates significant multipath structure peaks and nulls.

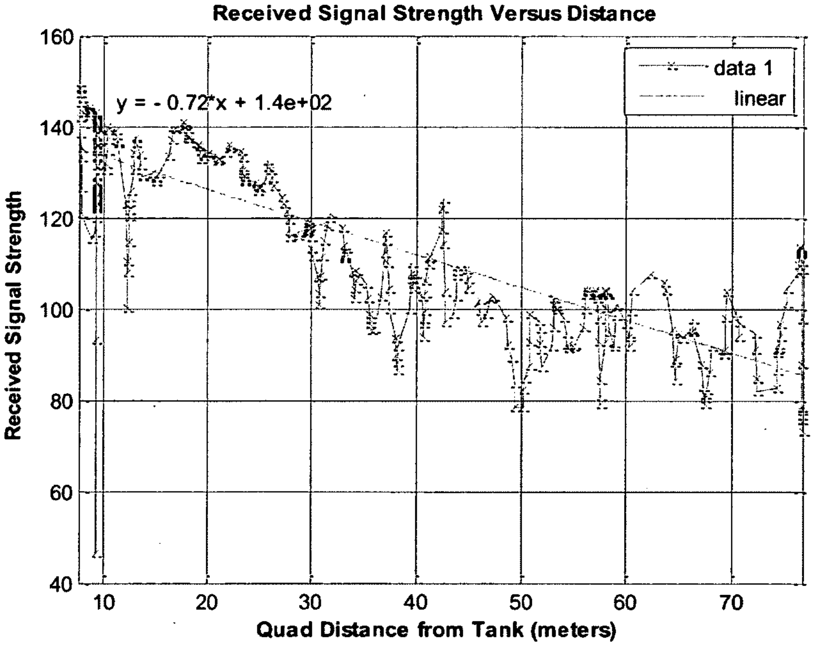


Figure 4.7 - Linear Curve Fit to Tank Raw RSSI Encoder Count vs. Distance

Figure Description: A significant downward trend is noted as the separation between the tank and relay radio are spaced further apart.

4.2 Received Signal Strength Variation with Patterned Movement

Linear Displacement Test

Purpose of Experiment

It was desired to investigate the variation in received signal strength as the relay platform performs a series of choreographed maneuvers. The effectiveness of a positioning algorithm is largely determined by the networks ability to detect changes in link quality as users move about their environment. Therefore, it is necessary to determine the variation in network link quality caused by the execution of these simple maneuvers.

Experiment Setup

Three simple patterns were flown with the quadcopter while the ground station and tank unit were fixed at either corner of Bremner Field, a large artificial turf playing field at the University of New Hampshire. The experiment environment was chosen to minimize multipath issues associated with urban locations. This helped to reduce test variance due to these higher order effects while retaining the variables of interest. The quadcopter was placed at the same starting location before each run and before the relay link was brought online. All reported MAVlink packets returned by vehicles and radios were logged within Qgroundcontrol.

Experimental Procedure:

The first quadcopter maneuver involved walking with quadcopter in hand along the line between the ground station and tank positions at a constant rate. Upon reaching the tank at the far end of the field, logging was brought to a halt.

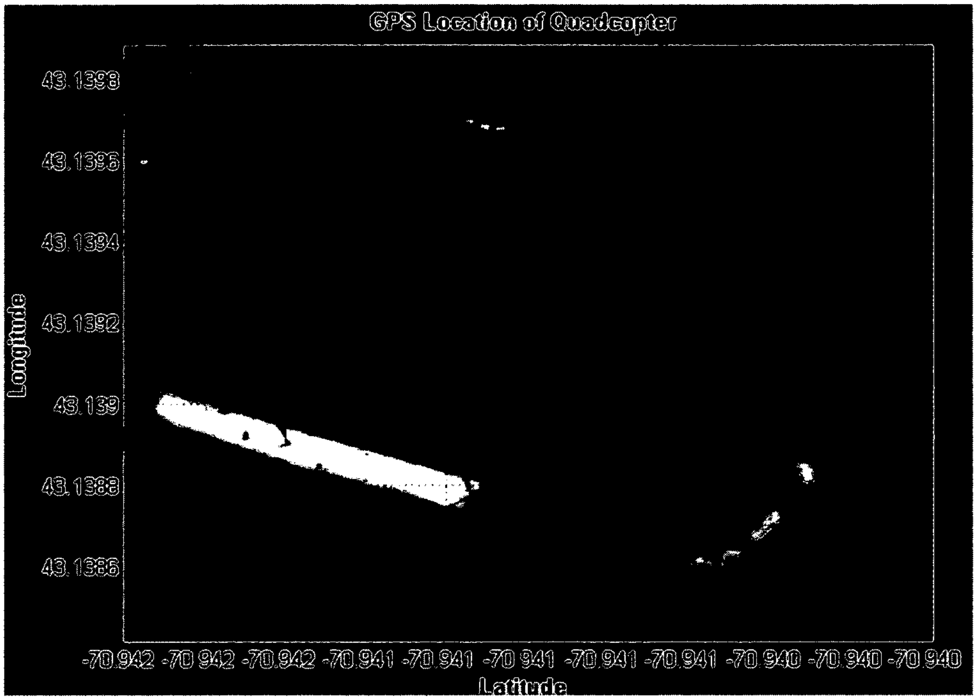


Figure 4.8 - Movement of Quadcopter during Experiment

Figure Description: Green line donates straight line path between ground station and tank (located to far right), red circles indicate reported GPS coordinates of quadcopter.

Results:

As the quadcopter was moved from the ground station to the tank, the signal strength responded as expected. The RSSI gradually decreased at the ground station while simultaneously increasing at the tank, which demonstrated the path loss proportional relationship with separation distance. As demonstrated in the previous investigation, the receiver was sensitive to relatively small displacements of position.

This sensitivity was further improved due to the differential nature of the link with units on either side of the quadcopter. Unfortunately the link did drop a number of times during the experiment but was able to recover and reestablish the network. While drops rarely occurred within the lab or at short distances, this present environment proved to be troublesome for the link. As Figures 4.9 and 4.10 make clear, there is rapid fluctuation of RSSI due to multipath, primarily ground reflection, on both sides of the relay link over short horizontal displacements. However, the network performed well enough to capture all necessary data for a valid experiment.

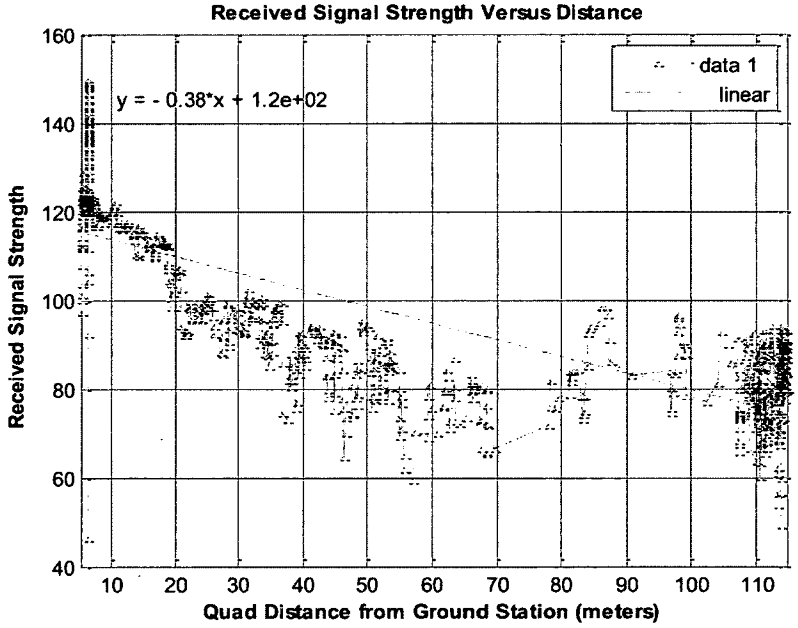


Figure 4.9 - Ground Station Signal Strength Variation with Distance (Raw Encoder Count)

Figure Description: Linear downward trend is noted as separation distance between relay and ground station is increased. The signal slope is much less than that of previous tests in urban environments. This data correlates with a reduced path loss exponent found in open environments. Note the large ripple in signal strength between 30 and 70 meters, this is most likely caused by multipath nulls present in the environment.

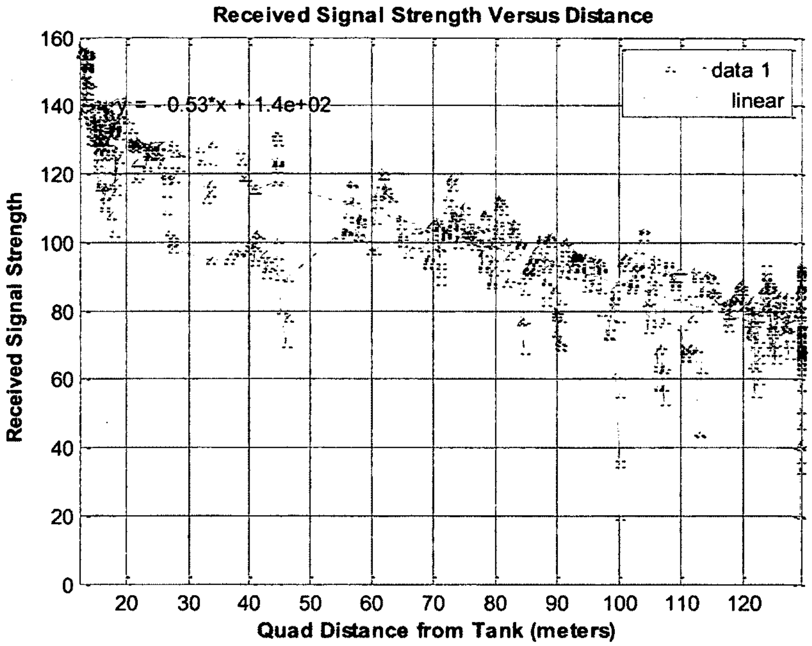


Figure 4.10 - Tank Signal strength Variation with Distance (Raw Encoder Count)

Figure Description: A similar downward slope of signal strength is observed for the tank as distance is increased. Note that the tank unit experienced little RSSI variance about the linear trend line, but a greater mean path loss versus distance, indicating an altered multipath environment at this end of Bremner Field versus the ground station's side.

Circular Pattern Test

Purpose of Experiment:

The goal of this investigation was to determine the scale of RSSI variation about a circular pattern. If there is enough signal deviation with vehicle movement, it may be possible to infer a location at which to best support the relay link or a direction toward a particular end user. For a RSSI-based positioning algorithm to be realizable there needs to be a detectable signal strength variation within a relatively limited region of operation.

Experiment Setup:

The ground station and tank were once again fixed at diagonal corners of Bremner Field. The quadcopter was placed in the center of the field approximately equidistant to the two end units. The link was established and Qgroundcontrol was configured to log all available data.

Experimental Procedure:

About the midfield point a circular path was followed with a radius of approximately 10 meters. A slow pace was set in order to provide enough time for global position coordinates to be sampled in order to improve spatial resolution. Upon completing a 360 degree traversal a closed loop path was achieved and the logging of experimental data was stopped.

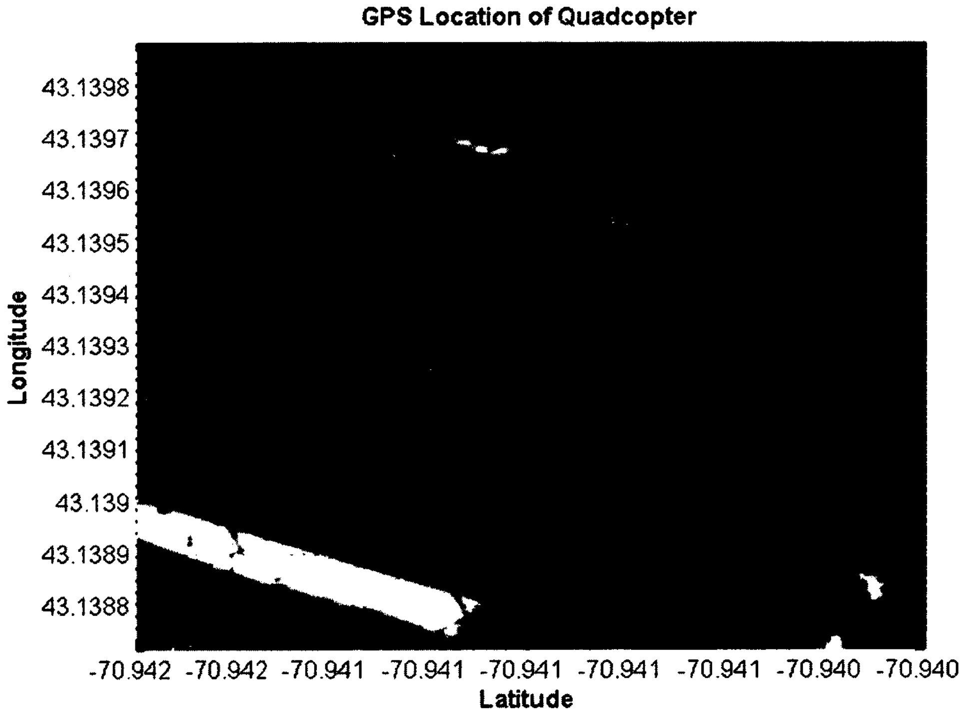


Figure 4.11 - Circular Path of Quadcopter during Test

Experimental Results

The circular maneuver generated two sets of signal strength indicators corresponding to the perspective of the tank and ground station receivers. Plotting each unit's RSSI versus the quadcopter's position provided a glimpse of the overall environment signal strength. As shown in Figure 4.12, a circular maneuver at a 10 meter radius provided enough spatial variation to create a detectable change in signal strength at both ends of the relay link. The two signal rings display a rotation about the axis perpendicular to the vector between the ground station and tank. This result coincides with the previous linear displacement test. As the quadcopter moves closer to any particular unit, the received signal level is increased while movement away results in a drop off.

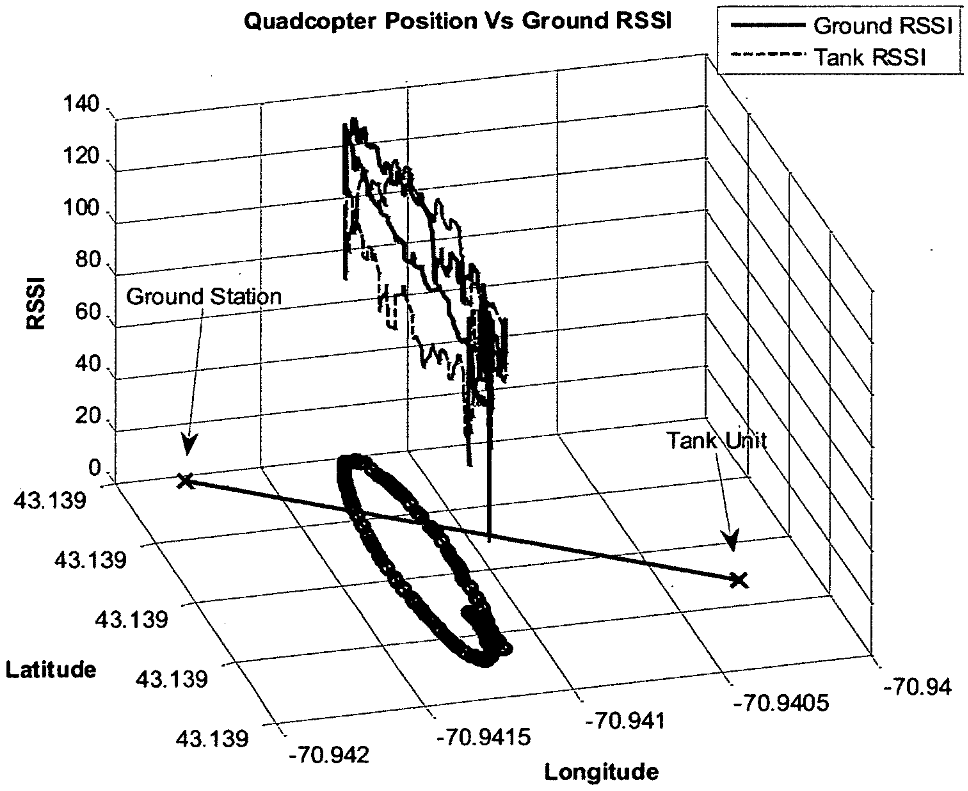


Figure 4.12 - End Point Signal Strength with Circular Movement

As seen in Figure 4.13, the circular path provided a quasi-sinusoidal variation in distance between the quadcopter and the end units. The signal strength followed a similar trend demonstrating the strong coupling between relay-end user separation distance and received power levels. As the quadcopter was moved 18 meters towards the ground station on its circular path, the received signal varied by 20 encoder counts (10 dB). On the tank end of the link the same 10 dB deviation was observed in the resulting data. If necessary, larger radius circular paths can be followed to increase the signal strength variation measured by either user.

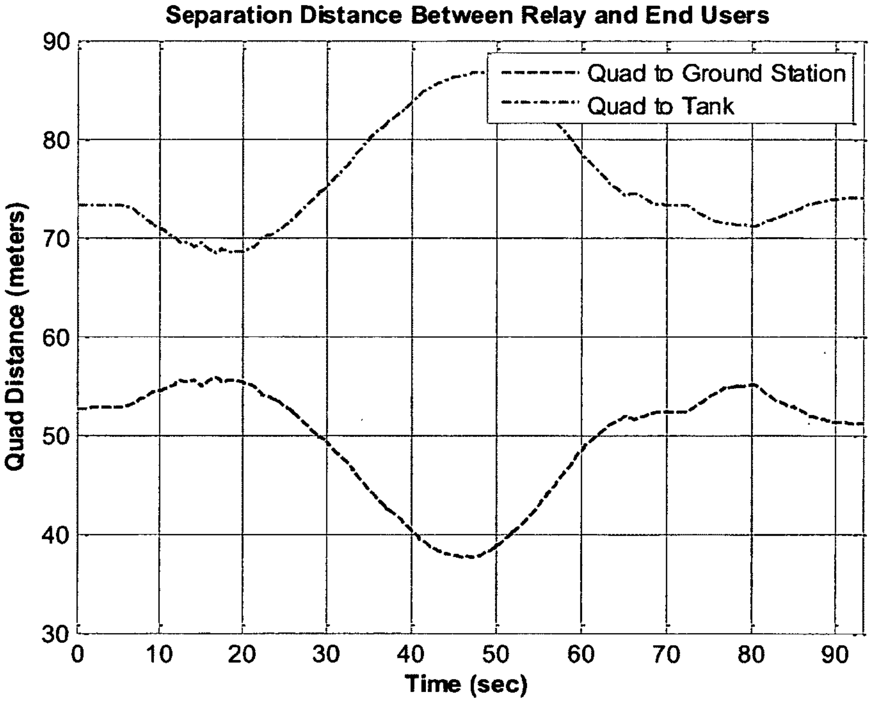


Figure 4.13 - Separation Distance between Relay and End Users

Figure Description: The traversed circular path created a quasi-sinusoidal variation in link distance between the relay and end users.

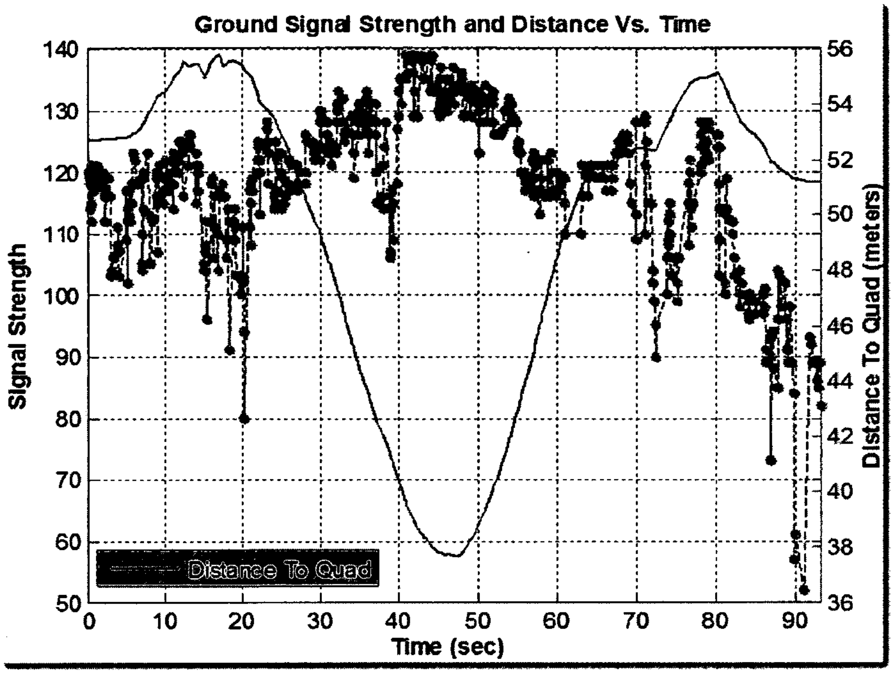


Figure 4.14 - Ground Station Signal Strength and Distance from Quadcopter vs. Time (Raw Encoder Count)

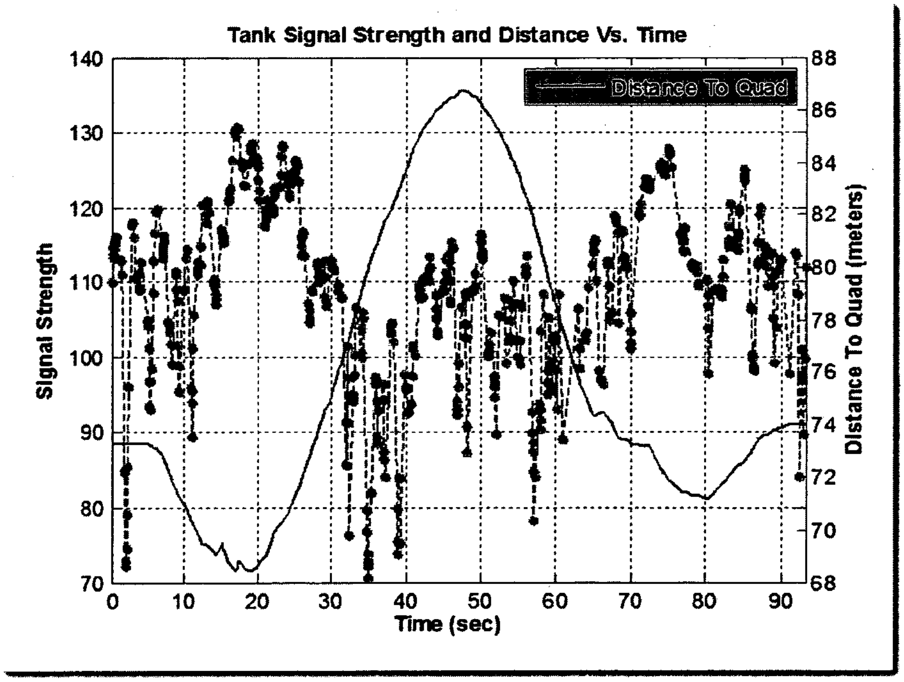


Figure 4.15 - Tank Signal Strength & Distance from Relay vs. Time (Raw Encoder Count)

Altitude Variation Test

Experiment Purpose

The purpose of this experiment is to determine the relationship between the supportable link range and the quadcopter's height above the ground.

Experimental Setup

The two end users (ground station and tank) were separated from one other by approximately 130 meters and the quadcopter was placed at the link's geometric midpoint. The relay connection was established and, upon verification of vehicle data generation activity, Qgroundcontrol was configured to log all available data streams.

Experimental Procedure

The quadcopter was flown off the ground and maneuvered to maintain as little horizontal movement as possible. Meanwhile the altitude was varied repeatedly from near 0 to approximately 20 meters off the ground.

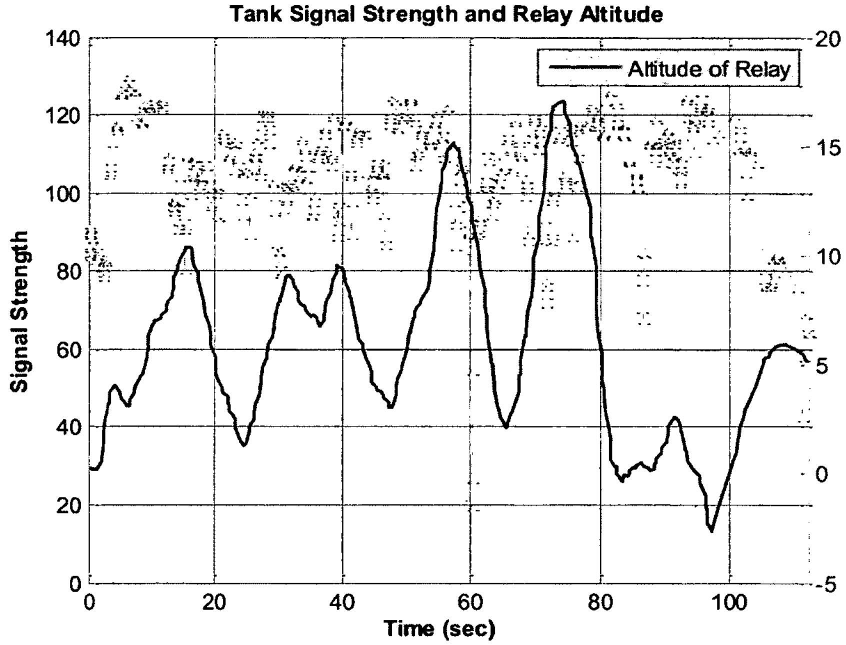


Figure 4.16 - Tank Signal Strength (Raw encoder counts) and Relay Altitude vs. Time

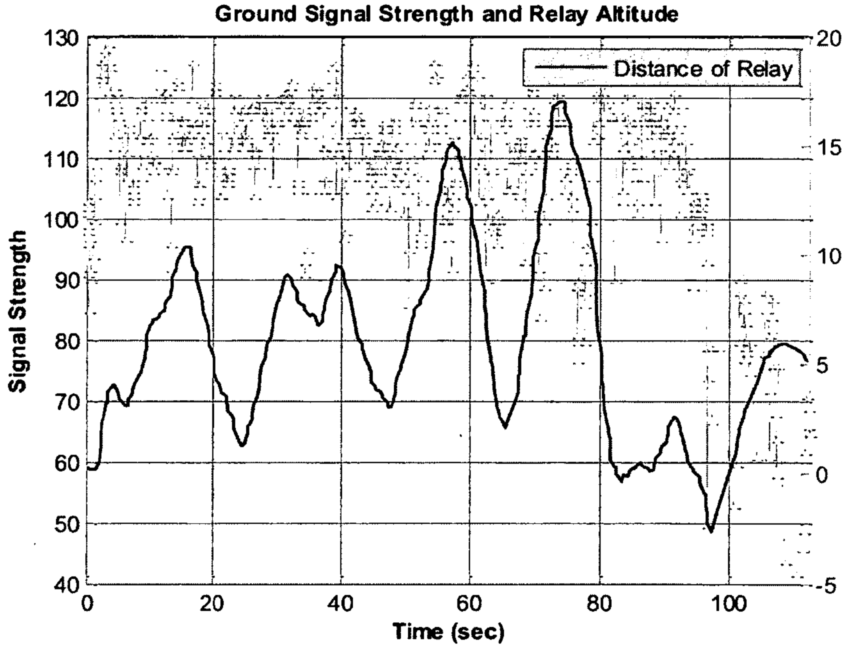


Figure 4.17 - Ground Station Signal Strength (Raw encoder counts) and Relay Altitude vs. Time

Results:

Altitude proved to drastically affect the link quality and maximum link range. When the relay remained on the ground, there were many instances when the link would drop and have trouble reconnecting. This issue was solved once the platform was simply raised by a half meter. Once the relay was flying in the air, the link quality improved even more with a significant boost in received signal on either end of the link. As the quadcopter fluctuated up and down the signal strength fluctuated wildly. A direct proportional relationship between altitude and signal strength was not seen in the data, indicating that the relationship between the two is more complex than originally thought. This entanglement can be mostly attributed to the phenomena commonly known as "two ray ground reflection" [5]. As the relay's height is altered, the phase between the line of sight and ground signal will shift between 0-180 degrees resulting in height dependent constructive/destructive interference. The radio horizon as seen by the relay node is $d = \sqrt{h(D + h)}$ where D is the diameter of the earth and h is the altitude of the repeater platform. The relay stands to increase its line of sight range greatly as its altitude increases, thus providing larger coverage area even in environments with vertical obstacles. As the transceiver module's theoretical maximum communication range is 1 km under line of sight conditions, there are obviously diminishing returns beyond a certain range of vertical movements. The results obtained are promising in demonstrating the value of an aerial relay platform.

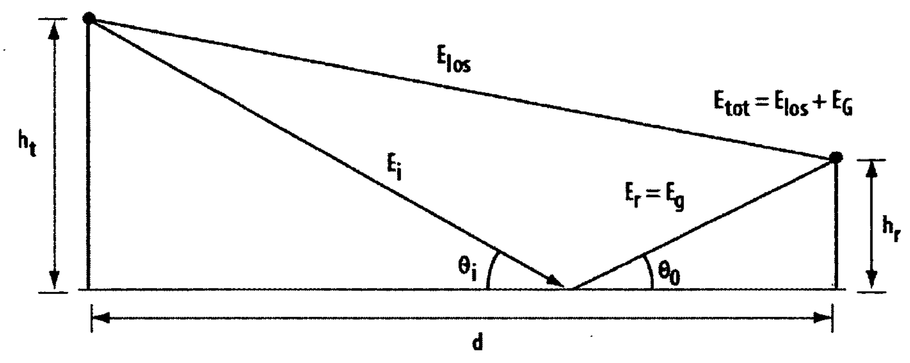


Figure 4.18 - Two Ray Propagation Model

Figure Description: This model accounts for the major source of signal fluctuation during the altitude test. As the reflected waveform shifted in phase relative to the line of sight signal, the receiver experienced constructive and destructive interference. **Figure Credit:** Gary Ybarra [5])

4.3 Outdoor Flight Tests beyond Line of Sight

Beyond Line of Sight Test

Experiment Purpose:

This investigation was designed to determine the capability of the relay to improve user connections in situations where line of sight between endpoints is unavailable. The test should provide an indication of the relay's value in providing communication between users who would otherwise be unable to connect directly.

Experiment Setup:

The tank and ground station were placed approximately 107 meters apart from one another with a large building (Kingsbury Hall) obstructing their line of sight. The quadcopter was placed near the ground station, and Qgroundcontrol was set up for logging of all pertinent data. In the first test only the quadcopter and ground station were utilized while the tank remained turned off. In the second experiment all three units were connected and logged.

Experiment Procedure:

The first test started with the quadcopter and ground station sitting side by side. Once data logging had begun, the quadcopter was carried away and around the corner of Kingsbury according to the path shown in Figure 4.19. When the link became unstable, the path was retraced and logging was stopped upon arrival at the ground station.

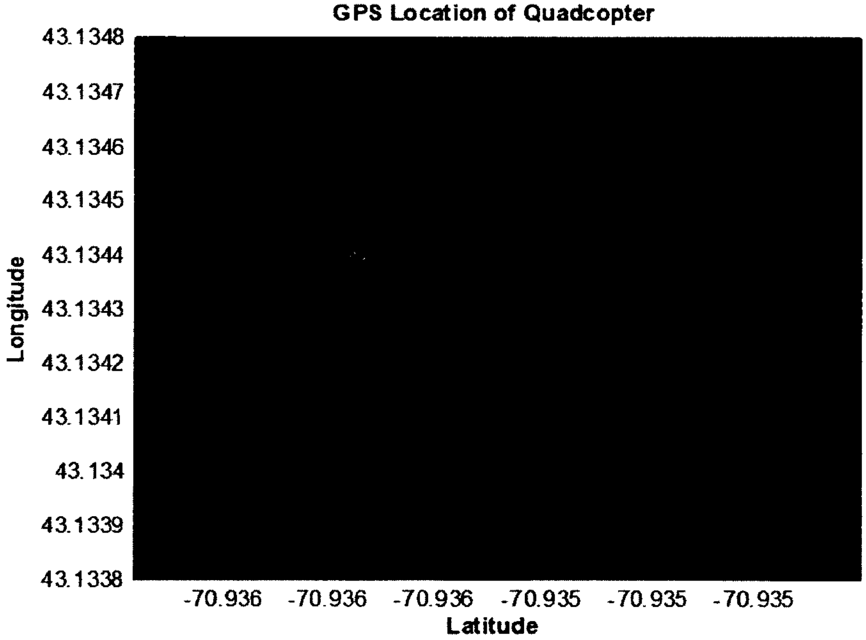


Figure 4.19 - Path Taken by Relay Platform during Point-to-Point Test

Figure Description: Note that the reported relay position clips through Kingsbury Hall on the return path. This error most likely resulted from a decrease in visible GPS satellites multipath reflection caused by the GPS receiver's close proximity to Kingsbury Hall.

This scenario served to simulate a point-to-point link without a cooperative store and forward node. In the second test all three units were connected to the network and the quadcopter once again served as the relay in the network. The quadcopter was placed near the building's southwest corner (toward the lower left side of the image in Figure 4.19) in order to provide line of sight between the relay and both end points. The tank was located near the position in which the link began to deteriorate in the previous test (northern corner of Kingsbury Hall in Figure 4.19 as denoted by green X). After data logging had begun, the quadcopter was flown about its start position for approximately two minutes. After this period was over, the quadcopter was returned to the start position and logging was halted.

Results:

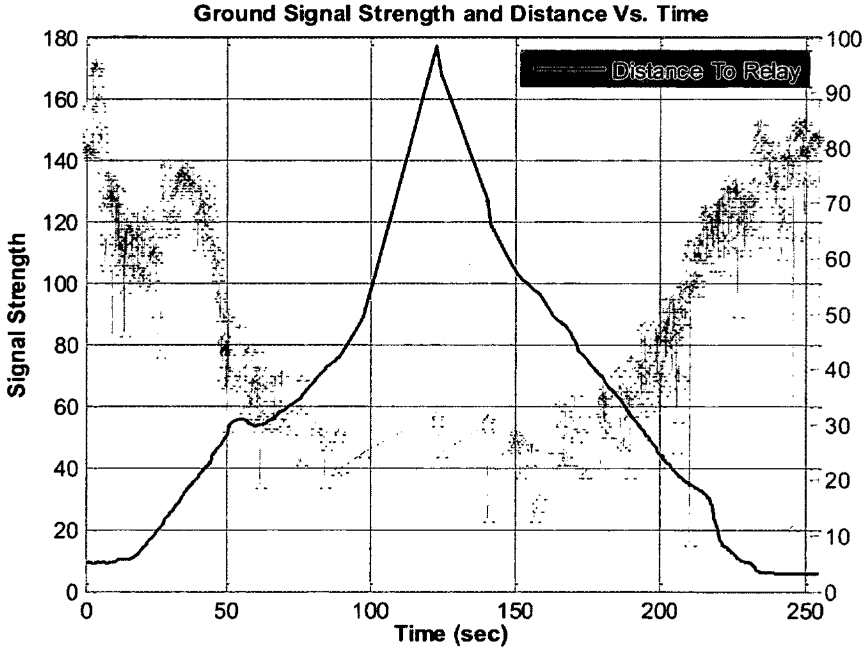


Figure 4.20 - Single Ended Link Test Distance and RSSI (Raw Encoder Counts) vs. Time

Figure Description: Rapid drop off in signal strength is observed as the relay rounds the Northwest corner of Kingsbury Hall. Once line of sight is broken, only non-direct modes of propagation maintain the link, resulting in sporadic communication performance.

The single ended point-to-point investigation demonstrated a rapid deterioration in signal strength as the ground station and quadcopter were separated from one another. Once line of sight was broken, the rapid RSSI drop-off became even more apparent while the link began to exhibit disconnections. Figure 4.20 shows a substantial loss of signal strength with a relatively small movement of only 100 meters. This demonstrates the radio transceivers' heavy reliance on line of sight propagation for reliable communications. When users operate outside of each other's radio horizon, due to distance or obstacles, the link will rapidly deteriorate.

While the point-to-point configuration signal quality dropped quickly with minimal user movement, the inclusion of the relay produced a much better connection for end users. The tank was placed at the location in which the link became unsupportable in the previous test (near the northwest corner of Kingsbury Hall, shown near the upper center of the picture in Figure 4.19). Once again the adjacent building prevented line of sight between these two units rendering a point-to-point connection unobtainable. While the RSSI in the previous test dipped below -97 dBm for the ground station, the incorporation of the relay boosted the connection strength significantly at both ends. The ground station link signal strength hovered about -67 dBm while the tank unit varied greatly but remained near -90 dBm (Figure 4.21). More importantly, the link remained active and functioning even with the tank and ground station outside line of sight and at a great separation distance. This verifies the capability of the relay to support connections between otherwise unserviceable users.

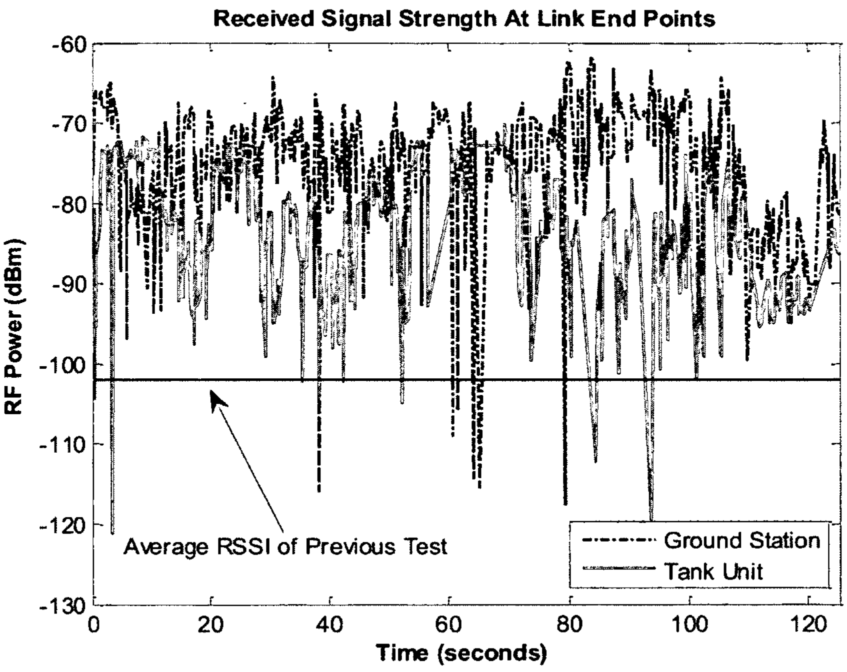


Figure 4.21 - Received Signal Strength at End Users with Relay

Figure Description: The received signal strength at either end of the link (tank and quadcopter) is displayed for the test involving the relay. The green line is the average received signal strength of the point-to-point test when the quadcopter-ground station link began to fail.

4.4 Automatic Link Balancing Test

Purpose of Experiment

It was desired to check for proper midpoint and scanning coordinate calculation and conditional code flow (in deciding which direction to move or when to move to midpoint) to determine if the positioning algorithm is capable of generating valid global coordinates in response to an imbalance in the signal strengths at either end point.

Experimental Setup

As shown is Figure 4.22, two computers were connected to one another over a wireless transparent serial link. Computer #1 ran the relay position algorithm embedded within the Qgroundcontrol source code. Pseudo-GPS coordinates were created for both the ground station and tank unit as shown in Figure 4.23 within the positioning algorithm. The signal strength at both link end points was simulated within the algorithm through the combination of a random deviation generator, fixed offset and a function of separation distance (separation distance in meters between the relay and the respective link end point). This provides a reasonable signal variation which one could expect within an urban environment. An unaltered version of Qgroundcontrol was executed and configured to display Google Maps onboard computer #2. Google Maps was initialized to display global position data retrieved from the incoming serial stream (Position algorithm calculated coordinates). Qgroundcontrol was configured on both PCs to interface with the attached radio module and connect both applications over a transparent serial data link. With both versions of Qgroundcontrol properly configured and connected through the serial data link, a series of breakpoints were set within the positioning algorithm source code.

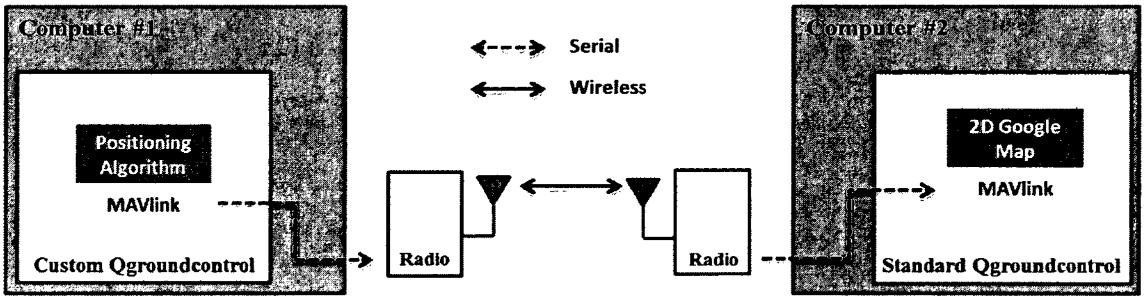


Figure 4.22 - Setup for Testing Positioning Algorithm

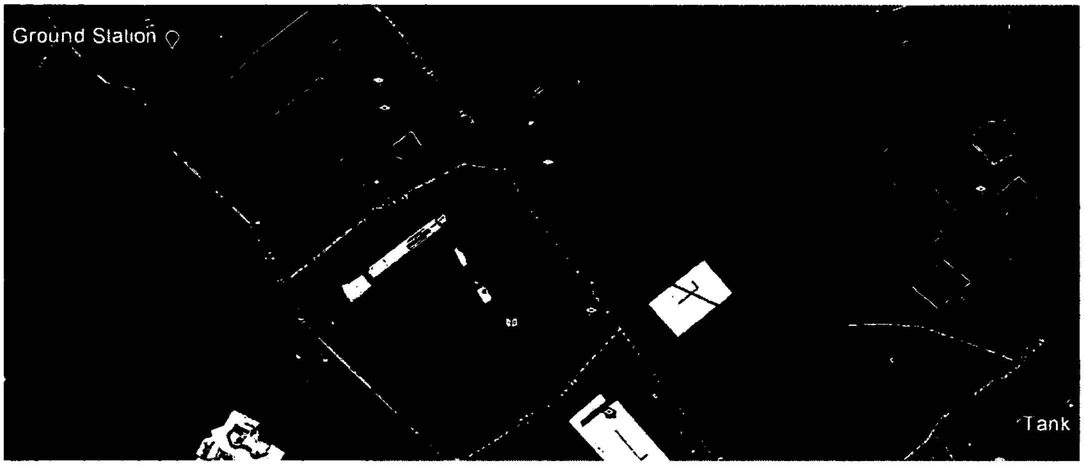


Figure 4.23 - Simulated GPS coordinates for Positioning Test

Figure Description: Pseudo-GPS coordinates for the ground station and tank unit during position algorithm test. Ground station located at western corner of Kingsbury Hall, adjacent to McDaniel Drive. Tank unit located next to Englehardt Hall near Quad Way.

Experimental Procedure

The positioning algorithm was executed, producing a series of waypoints in response to the program's state and the simulated link end point signal strengths. Generated global coordinates were transmitted over the transparent serial link to the second instance of Qgroundcontrol where the Google Map was updated with waypoint indicators. Breakpoints halted code execution each time simulated link conditions required a new position to be calculated. The link signal strengths, vehicle positions, and destination coordinates were inspected at this point to verify the validity of the current algorithms output.

Results

The positioning algorithm functioned in a consistent manner with no notable outliers as shown in Figure 4.24. In each case that an imbalance was present, a global coordinate was generated with the correct bearing and distance, consistent with the severity of the link imbalance. The algorithm demonstrated proper decision making in all cases where code execution was brought to a halt and statistics were checked. These results are promising as the current code structure surrounding the positioning algorithm already allows for a direct interface to vehicles connected to the serial data link. The current relay positioning algorithm (RPA) supports communications with the true relay platform outside of simulation, and true flight controller commands. Further testing is necessary to determine if the RPA is stable, and capable of correct functioning outside of the simulation environment. This test was put off due to concern for the safety of the quadcopter as this platform was essential in future tests.

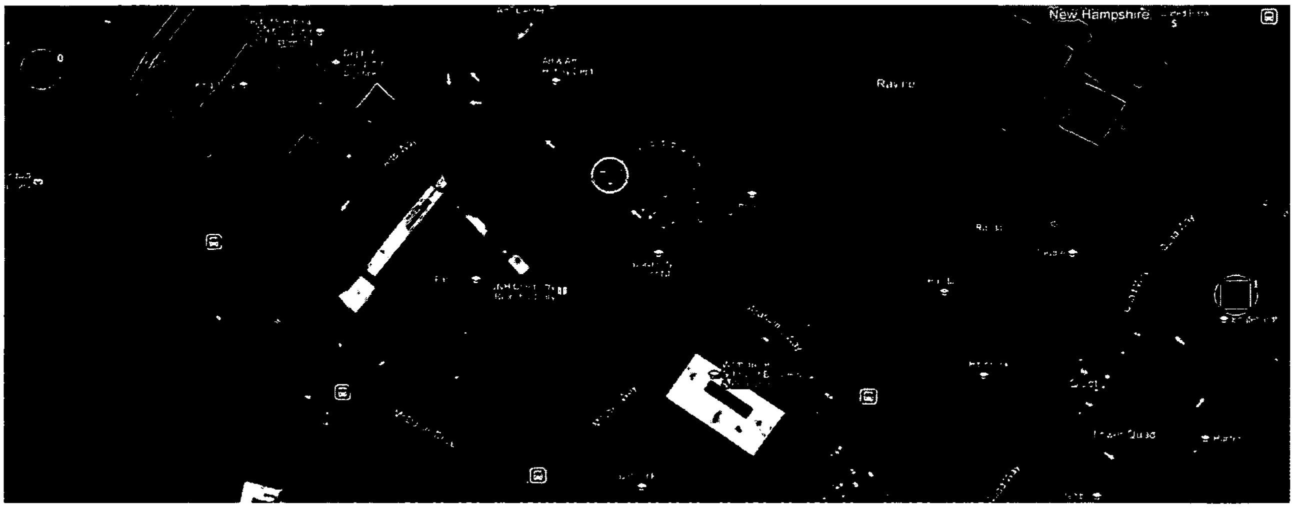


Figure 4.24 - Position Algorithm Output over Time

Figure Description: The orange line traced between waypoint 0 (ground station) and 1 (tank) is the direct path between link endpoints. The orange triangle surround by a yellow circle represents the quadcopter's current position. Red circular indicators were updated each time the positioning algorithm generated a new coordinate to which the relay platform should move. The circular path represents the scanning routine pattern used to sample the RSSI. Once the best location is selected (based on the highest average RSSI at both end points) from the scan candidates, the vehicle moves to this location and the proportional positioning controller takes over. All calculated positions from then on are located on the path between end points.

Chapter 5

SUMMARY & FUTURE WORK

5.1 Summary of Capabilities

The intelligent aerial relay platform described in this thesis is capable of improvement to communication performance between isolated users whom would otherwise be unable to communicate. A transparent serial data link is provided to connected users allowing asynchronous data transfer without the need for complex handshaking. IARN can automatically acquire and support the associated users as they come into range of the access point node. Lost connections are reestablished automatically when signal quality reaches a stable, suitable level. While the relay network provides the physical channel upon which serial data can flow, MAVlink contributes compatibility and safety among users. IARN supports simultaneous flow of commands, vehicle reports and radio statistics over the radio link. Vehicles can be monitored in real time with interactive displays and sent high level commands such as waypoints. All reported serial data received over the link can be logged in real time for later viewing or data processing. A dynamic aerial relay positioning algorithm has been developed to direct the quadcopter platform based on the current network communication performance. This system can predict three dimensional locations in which to place the relay and better support end users under time varying link conditions. This provides an intelligent, dynamic positioning system capable of supporting mobile users in complex environments even while the network topology evolves.

5.2 Limitations of Work

The effectiveness of the relay platform is governed by a number of factors. Quadcopters afford hovering capabilities and high mobility at the cost of a reduced flight time. This places strict limits on the duration over which the network can remain effective in supporting users. The short flight time would be aggravated with the inclusion of a range-extending, high power transmitter. To alleviate the already heavy power requirements placed on the existing battery, an efficient transceiver was selected at the cost of range. The selection of a half-duplex radio transceiver also places strict limits on the available network bandwidth as radios must switch between TX and RX modes. For users requiring high rate data transmission (video, high quality voice) the current network is inadequate. With that said, the current network configuration supplies more than enough bandwidth for the subsystems onboard both the tank and quadcopter platforms, which include IMU, GPS, magnetometer, and ultrasonic proximity sensor modules as well as a sophisticated flight controller module on the quadcopter.

The radio firmware developed in this thesis research is built on top of a physical layer module without compatibility with existing IEEE network protocols. Users attempting to access the radio link with other communication technologies will be unable to do so without first acquiring the proper firmware and suitable hardware. The current relay configuration only supports three concurrent devices (tank, quadcopter and ground station) connected through a merged serial stream. While it is possible to add capability to share the link between more devices, this would require further code development and bandwidth allocation. While many existing network protocols contain features that could solve the aforementioned issues, a significant amount of overhead was required. For the purpose of this thesis research it was decided that the communication protocol would be built from the ground up to minimize the associated overhead and signal processing complexity.

The current algorithmic positioning system is limited in its ability to predict the best location in which to support end users. Simply traversing the line between the tank and ground station will not always yield the optimal locations. Often time, the best solution to the relay position is located elsewhere.

5.3 Suggested Future Work

The current communication framework can be improved in a number of areas. Future system enhancements can be organized into alteration of the data link, flight platform and positioning algorithm capabilities. Reorganization of the data link to include an existing physical/link layer standard (e.g. IEEE 802.11) would increase compatibility and bandwidth over that of the current protocol stack, albeit at the likely expense of supportable relay range and necessary source code footprint. Support of multiple users could also be obtained by the proper selection of physical layer signaling protocol, although this would entail significant increase in software complexity and power consumption.

The flight platform could also be altered to increase its viable time aloft, thus enabling communications to be supported over a longer period of time. Selection of higher efficiency motors and batteries with greater energy density would support longer duration flights and network access.

Further testing of the RPA is necessary to determine the controller's stability and real world capabilities outside of the previous simulation. The existing Qgroundcontrol software structure makes this process straightforward with only limited additional code development. Improvement to the positioning algorithm capabilities is another worthwhile undertaking. The current algorithm relies heavily on global positions of all users and simple proportional control methods to generate three dimensional coordinates for desired relay positioning. The existing framework could be altered to include more robust control methods (e.g., adaptive control, neural network) that render the position calculation less susceptible to RSSI fluctuations due to multipath. Reliability could be improved by implementing the algorithm onboard the quadcopter platform, thus eliminating the need for calculated coordinates provided by the ground station to propagate over the wireless connection.

Appendix A

References

1. "Airborne Communications Relay Could Become Primary Mission for Tactical UAVs, Defense Update", *Defense Update*, 11 Jan 2010, Available: defense-update.com/features/2010/january/airborne_relays_for_uavs_110110.html
2. Izhak Rubin, Runhe Zhang, "Placement of UAVS as Communication Relays Aiding Mobile Ad Hoc Wireless Networks", *Military Communication Conference*, Oct 2007 Issue, pp. 1-7,
3. Pengcheng Zhan, Kai Yu, A. Lee Swindlehurst, "Wireless Relay Communications with Unmanned Aerial Vehicles: Performance and Optimization", *Aerospace and Electronic Systems*, July 2010 Issue, pp. 2068-2085
4. Morgenthaler, S. ,Braun, T. ; Zhongliang Zhao ; Staub, T. ; Anwander, M., UAVNet: A mobile wireless mesh network using Unmanned Aerial Vehicles, Globecom workshops, Dec 2012, pp. 1603-1608
5. Professor Gary Ybarra, Two Ray Ground Reflection Model, Duke University Dept. of Electrical and Computer Engineering, Dec 2001, pp. 1-4, Available: http://people.ee.duke.edu/~gary/ECE486/ECE186lecture8.pdf
6. Hyon Lim, Jaemann Park, Daewon Leed and H.J. Kim, Multirotor Aerial Vehicle, IEEE Robotics and Automation Magazine, Sept. 2012, http://online.qmags.com/RAM0912?pg=1&mode=2#pg1&mode2

7. Lorenz Meier, Andrew Tridgell. (2010) MAVlink, Available: http://qgroundcontrol.org/mavlink/start

8. Matthew Stibbe, "Google's Next Cloud Product: Google Blimps To Bring Wireless Internet to Africa", Forbes Magazine, June 2013, Available: http://www.forbes.com/sites/matthewstibbe/2013/06/05/googles-next-cloud-product-google-blimps-to-bring-wireless-internet-to-africa/

9. Patrick Vincenta, Murali Tummalab, John McEachen, A new method for distributing power usage across a sensor network, IEEE Sensor and Ad Hoc Communications and Networks, Sept. 2006, pp. 518-526, Available: http://www.sciencedirect.com/science/article/pii/S1570870507001680

Definition of Terms

1. **IMU:** (Intertial measurement unit) - Typically used to measure orientation of a device through the combination of an accelerometer, magnetometer and gyroscope.
2. **GPS:** (Global positioning system) – calculates global coordinates (Latitude, Longitude) based off timing of received satellite signals.
3. **UAVNet**: A communication framework created with highly mobile aerial vehicles equipped with IEEE 802.11g mesh network nodes.
4. **MAVlink:** (Micro Aerial Vehicle Link) – packet based link level communication protocol and message library. Provides error detection and packet encoding/decoding of transmitted data. Open source design allows for further customization and inclusion of addition library messages.

5. **Qgroundcontrol:** Open source ground station application. Provides real-time monitoring and controlling of connected vehicles and devices.
6. **Pseudo Data:** fake or simulated serial data that contains no actual information but may be useful in testing the physical link layer.
7. **IARN:** (Intelligent Aerial Relay Network) – a communication framework that supports a communication link between a remote ground rover and ground station through a dynamic aerial relay. IARN supports vehicle reports/commands and real time link monitoring simultaneously. Dynamic positioning algorithm positions relay platform in real time to retain the rover-ground station connection as the propagation environment changes.
8. **UART:** (Universal Asynchronous Receiver Transmitter) – duplex serial communication protocol for asynchronous data transmission. Utilized for serial transfer onto and off of radio boards.
9. **SPI** – (Serial Peripheral interface) - A high data rate full duplex serial communication protocol. Used onboard the Si1000 to transfer data between the embedded transceiver module and 8051 microcontroller at a rapid rate.
10. **IEEE 802.11g:** wireless communication protocol used for Wi-Fi enabled devices (laptops, routers etc.). Utilized orthogonal frequency-division multiplexing as the modulation scheme.