Long-range Communication Framework for Autonomous UAVs

by

Elchin Mammadov

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc. degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

The communication range between a civilian Unmanned Aerial Vehicle (UAV) and a Ground Control Station (GCS) is affected by the government regulations that determine the use of frequency bands and constrain the amount of power in those frequencies. The application of multiple UAVs in search and rescue operations for example demands a reliable, long-range inter-UAV communication. The inter-UAV communication is the ability of UAVs to exchange data among themselves, thus forming a network in the air. This ability could be used to extend the range of communication by using a decentralized routing technique in the network. To provide this ability to a fleet of autonomous dirigible UAVs being developed at the University of Ottawa, a new communication framework was introduced and implemented. Providing a true mesh networking based on a novel routing protocol, the framework combines long-range radios at 900 MHz Industrial, Scientific and Medical (ISM) band with the software integrated into the electronics platform of each dirigible. With one radio module per dirigible the implemented software provides core functionalities to each UAV, such as exchanging flight control commands, telemetry data, and photos with any other UAV in a decentralized network or with the GCS. We made use of the advanced networking tools of the radio modules to build capabilities into the software for route tracing, traffic prioritization, and minimizing self-interference. Initial test results showed that without acknowledgements, packets can be received in the wrong order and cause errors in the transmission of photos. In addition, a transmission in a presence of a third broadcasting node slows down by 4-6 times. Based on these results our software was improved to control to flow of transmit data making the fragmentation, packetization, and reassembly of photos more reliable. Currently, using radios with half-wavelength dipole antennas we can achieve a one-hop communication range of up to 5 km with the radio frequency line-of-sight (RF LOS). This can be extended further by adding as many radio nodes as needed to act as intermediate hops.

Acknowledgements

The research and implementation work presented ahead would not exist without the motivation and help of many.

First and foremost, I would like to thank my thesis supervisor, Dr. Wail Gueaieb, for giving me the opportunity in gaining new skills and his guidance throughout the research project.

I would like to thank, Dr. Derek McNamara, who helped me to find my research area during the starting stages of my Master studies and provided his opinion about my work throughout the years.

Dr. Eric Lanteigne, who helped me acquiring all the required equipment on time and NSERC for funding the research.

Dr. Abdulmotaleb El Saddik, for his advice on troubleshooting the problem of multimedia data transmission.

Dr. Brad Jackson from Defence R&D Ottawa, for his feedback on the summary of my completed work.

My fellow graduate students working in my lab for helping me with troubleshooting the software.

Last, but not least, my family and my fiancée, Eugenia, who helped me with drawing the diagrams, and always believing in me and always keeping me focused on reaching my goals.

Contents

List of Tables

List of Figures

List of Acronyms

AODV Ad hoc On-demand Distance Vector

API Application Programming Interface

AUGNet Ad hoc UAV Ground Network

bps Bits per second

COTS Commercial Off-The-Shelf

DSSS Direct-sequence Spread Spectrum

DSDV Destination Sequenced Distance Vector

DSR Dynamic Source Routing

ERP Effective Radiated Power

FHSS Frequency-hopping Spread Spectrum

FSK Frequency Shift Keying

GCC GNU Compiler Collection

GCS Ground Control Station

GPS Global Positioning System

ISM Industrial, Scientific and Medical

MANET Mobile Ad hoc Network

MAV Micro Air Vehicles

MIMO Multiple-Input and Multiple-Output

OLSR Optimized Link-State Routing

OSPF Open Shortest Path First

QAM Quadrature Amplitude Modulation

RF LOS radio frequency line-of-sight

RS-232 Recommended Standard 232

RSS Received Signal Strength

SPOT Sensor Platform for Observation and Tracking

UART Universal Asynchronous Receiver/Transmitter

UAV Unmanned Aerial Vehicle

USB Universal Serial Bus

Chapter 1

Introduction

A UAV is a flying platform without a pilot present on-board. Most common representatives of a UAV is fixed-wing and rotary-wing aircrafts. Figure 1.1 shows an example of these aircrafts. A modern UAV uses global positioning system (GPS) to determine the coordinates and the speed with respect to Earth. The spatial orientation and the acceleration of a UAV is determined using gyroscopes and accelerometers. These are the key sensors used in controlling a UAV's movement in space. The control of a UAV is implemented using specialized digital signal processing microcontrollers called autopilots or computers with PC/104 or MicroPC form factors. The flight control software is usually written in C or C++.

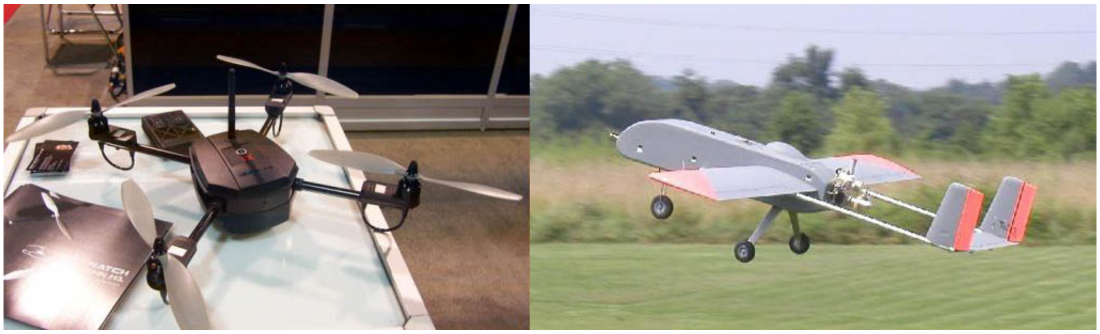


Figure 1.1: Rotary-wing (left) and fixed-wing (right) UAVs [35]

As the wireless technology started to advance and became available for open-source development, civilian UAVs started to transmit telemetry and multimedia data to the GCS. A survey of fixed-wing and rotary-wing civilian UAVs in [45] reveals that their flight duration is 1 hr at most. In the context of this thesis, a civilian UAV refers to an unmanned flying platform with a payload of up to 1 kg (payload is directly related to cost). The short flight duration is another reason why civilian UAVs do not require long range communication. For communication over a limited range and a high data rate most UAVs use the IEEE 802.11 a/b/g/n standard based Wi-Fi technology [30], [11]. The range of Wi-Fi is limited to 70 and 300 m in an indoor and outdoor environment, respectively. To increase the range up to 1-1.5 km a high-gain directional antenna can be used, but in a mobile environment it would require some kind of antenna tracking mechanism. The other solution used for increasing the range between a UAV and the GCS is using the 900 MHz band with high-gain antennas. The 900 MHz band is not part of the 802.11 standard and it is widely used with personal communication standards at a lower data rates. Some of the developed civilian UAVs use both bands, the 900 MHz for the long-range telemetry acquisition and the movement control, and the 2.4 GHz for the close-range multimedia streaming.

1.1 Motivation

Throughout the 20th century the UAV technology has undergone the research and development mainly in the military environment. As the price of electronic devices decreased and the wireless communication became open for use to public, UAVs picked up applications in aerial monitoring, specifically the search and rescue operations at the beginning of the 21st century. The current research and development of UAVs is shifting from remotely controlled to automated systems. This in itself introduces new challenges which need to be addressed.

The possibilities of using a lightweight and a small UAV is limited by fewer resources such as a small payload, a short battery life, and a limited power in wireless communication [34]. These limitations can be compensated for by utilizing several UAVs as a group. In addition, a group of effectively communicating light and small UAVs have useful properties. Based on the circumstances taking places worldwide, it is expected that the near future applications of large UAVs will require multi-agent systems [25]. It will be simpler and cheaper to conduct the refinement of the emerging multi-agent control algorithms on a smaller scale UAVs first. Another useful property is in an effective communication between UAVs. Reliable wireless connections among multiple UAVs can be used to extend the range of the communication compared to a single UAV's capability.

Most of the modern multiple-UAV control systems lack autonomous decision making to effectively reach a solution for a given task [24], [14]. Typical examples causing such decision making are: the emergence of new beneficial information to avoid obstacles earlier and the requirement for selecting a new route due to failure of some intermediate communication nodes. The higher the uncertainty of the environment, the more likely is occurrence of unplanned events, the lower the efficiency of the existing control systems, unable to make decisions and automatically adapt to the environment. The basis of the effective autonomous decision making in a group of UAVs is the effective inter-UAV communication. In a review of distributed autonomous robots, Parker articulated the importance of implementing and maintaining a reliable multi-robot communication [39]. A simulation study concluded that sharing just a small amount of data (e.g. position of each dirigible UAV) among multiple robots can make their application more effective [2].

1.2 Thesis Objectives

The main objective of this research project is to build a communication framework to support the development of autonomous search and rescue dirigible UAV fleet at the University of Ottawa. The dirigible UAV, called Sensor Platform for Observation and Tracking (SPOT), addresses the key issues in the current civilian UAV systems: endurance and low cost. A single uOttawa SPOT is capable of flying for 24+ hrs with a cost of $5000 CAD and requires a long-range communication in the range of 1-10 kilometres [45]. In addition, multiple SPOTs can be utilized for an effective coverage of a search and rescue area. Given the on-board electronic resources of a single SPOT, the initial set of design requirements for the communication framework were acquired. The only feasible way to satisfy the requirements for long-range communication, network scalability, and throughout is by building a decentralized mesh radio network between a group of SPOTs.

The proposed communication framework can also be used as a testbed for multi-UAV, also referred to as multi-agent, control systems. The majority of the existing multi-UAV control algorithms have been demonstrated through simulation based models because their physical implementations require inter-UAV communication which is simply assumed to be granted in such models.

1.3 Scope of Thesis and Contributions

In order to support the future development and testing of the SPOT fleet, a communication framework have been designed and built. The main design and development goal for the framework is to establish and maintain long-range reliable wireless links in a decentralized mesh. The desired range for communication is in the range of 1-10 kilometres between two SPOT dirigibles or the GCS and a SPOT. The focus is on maintaining reliable links over a long range rather than finding ways to achieve faster data rates. Because this is a real-time design work and not a simulation based project, it carries equipment cost constraints, needs performance validation in a real environment (preferably at high altitude), and make use of the unlicensed ISM frequency bands. In order to begin with the development of the framework and start collecting initial test data three radios are selected to build the communication framework (minimum number required for establishing a network with mesh topology). The following work summarizes the scope and contributions of the thesis:

1. Design and development of data processing and connection management layer, with a control over the transmit power. Using a novel proprietary mesh routing protocol based on the 802.15.4 standard, the framework achieves a decentralized network topology. The literature review have not shown that this protocol has been used in the context of civilian UAV applications before. Making use of serial radios with high sensitivity, the implemented framework provides a communication support to the autonomous dirigible UAVs requiring a communication range, with a GCS, between 1-10 km.

2. The framework combines transmission of multiple data types into a single communication system. Determining the wireless communication requirements: the amount of data being sent and the frequency of the transmission. The dirigible SPOT is equipped with multiple sensors including a digital camera. In order to help a receiving node to identify what type of data is being received, we have designed a protocol which works by tagging each packet with data type identifier. This allows the dirigibles communicate autonomously.

3. Developing and maintaining reliable connections between dirigible UAVs when existing connections get broken and new ones are established automatically. By maintaining a maximum reliability, the ongoing transmission of data in DigiMesh is not affected by new routes replacing the existing ones.

The real-time video transmission is not supported by the current design for the long-range framework because even a highly compressed video format such as MPEG-4 Part 10 (H.264) requires data rates above 1 Mbps. However, the framework does support a semi-real time transmission of the recorded video files between SPOTs and the GCS.

1.4 Thesis Outline

The next sections of this thesis are organized as follows: Chapter 2 provides the review of the related work and different implementations of ad hoc networks with different topologies. The main focus is kept on the implementations applicable to civilian UAV. Chapter 3 provides an overview of the hardware-software configuration and the integration into the SPOT. The description of the software shows how information is processed. Chapter 4 presents the analysis of the results collected from real world test scenarios. Finally, Chapter 5 draws the conclusions and discusses possible future improvements which can be made to the framework by using antenna diversity techniques.

Chapter 2

Related Studies - State of the Civil UAV Communication Technology

2.1 Synopsis

This chapter discusses current implementations of the communication systems for the civil UAVs. The term civil UAV refers to a low-cost ($1000 - $10,000), limited capability, and a small size unmanned aerial vehicle. The most commonly used civil UAVs in research are the Commercial Off-The-Shelf (COTS) radio control model airplanes, and the quad-rotor platforms. The main goal of these UAVs are autonomous operation for collection of the telemetry and the multimedia data from remote locations. For the wireless communication with the UAVs, operators often use radio modems and the IEEE 802.11x (Wi-Fi) compliant devices. Many countries have regulations for operating these devices, which limit the communication distance with a civilian UAV. This is the case in a point-to-point communication link between a GCS and a single civilian UAV. The most practical solution to extending the communication range is introducing additional routing nodes, forming a wireless ad hoc network. Another solution, using cellular technology, has been demonstrated to work for altitudes up to 500 meters but it depends on the existence of a cellular infrastructure. The challenge remains in developing a communication framework for multi-UAV environments. In this chapter special focus is made on the implementations of the wireless networks involving multiple civilian UAVs.

2.2 Analog vs. Digital Communication for Radio Control

Today one can still find remaining civilian UAV models employing analog communication techniques, called modulation in wireless communication, working on 35 MHz or 40 MHz frequencies. In the analog communication the information is transmitted by changing one of the time-varying properties of an electrical signal, such as the amplitude, frequency, or phase. From the wireless ad hoc network point of view, as the analog signal is re-transmitted, maintaining the originally sent signal becomes harder. Each radio node introduces a noise degrading the original signal. There are no practical techniques that would allow to recover the original signal; therefore, analog communication techniques are not used in a multi-UAV environment. On the other hand, the digital communication techniques allow to recover part of the corrupted information or at least minimize the noise effects as the signal is re-transmitted. In the case of the digital communication the main indicator of errors is the bit error rate. When data is transmitted over-the-air, the digital signal processing techniques provide more immunity to noise and interference compared to the analog signal processing methods [46] [60]. Also the electronic circuitry for digital signal processing is more economical solution compared to the analog variants. Therefore, all the modern radio control is implemented using the digital signal processing techniques.

2.2.1 Digital Systems in Radio Control

Radio control models is a hobby apparatus of the 21st century, readily available to general public for purchase. Modern digital radio control system has long been a necessity in this field, but to buy high-quality and low-cost system is difficult. Radio-controlled helicopters, propeller-powered airplanes, and similar models are all favourite hobby tools of many people. Radio control of models makes it possible to perform complex tasks at a distance. In the past five years purchasing and installing various digital equipment on multi-copter platforms became very popular. An example of such UAV platform is shown on Figure 2.1

Applications of civilian UAVs are vast ranging from agriculture and construction to the oil and gas sector and the security sector [31], [1], [32], [51], [8]. Among these applications, Sarris in [52] mentions using civilian UAVs as a communication satellites,

Figure 2.1: An example of a multicopter with a digital camera [31]

connecting locations over a long-range. Limited in payload due to their size, each civil UAV is designed for a specific task. In cooperation these civil UAVs can accomplish a wide range of multiple tasks in a short period of time. However, the main research focus so far is to develop and improve various control algorithms for the civil UAVs. The control algorithms assume that a communication link is in place between multiple UAVs and the GCS. The communication may not be important for simulations or close-range operation because technology such as Wi-Fi provide quick, secure, and reliable links. Wi-Fi technology is good for operation within 50-100 meters range with omni-directional antennas and 800-1000 meters with high-gain directional antennas. However, it can not provide long-range communication in the order of several kilometres because the maximum transmit power is limited by the regulations in many countries. In addition to regulations, it is not feasible to install high-gain antennas on UAVs due to their payload and power limitations.

2.2.2 Radio Frequency for the Civil UAV Control

The 72 MHz is a radio frequency dedicated specifically to control civil UAVs. This is a globally accepted standard. Using radio controllers, such as the one shown on Figure 2.2, operators can control the movement of the UAV in space. When an operator moves the knobs on the controller, a sequence of electrical pulses representing the command is generated. This sequence is then mapped onto a 72 MHz wave via a pulse code modulation. The 72 MHz transmitter then transmits the sequence of pulses into the air. It is similar to a broadcast message in the network. Any receiver on the same

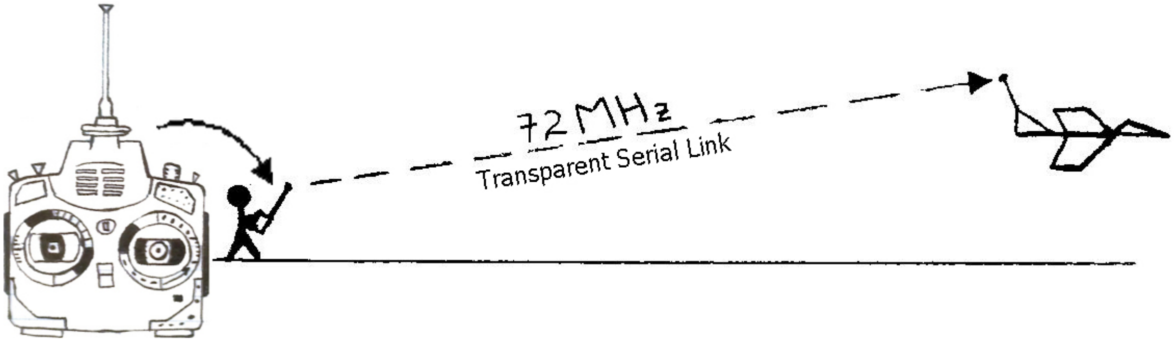


Figure 2.2: Typical radio control of a model aircraft

radio channel as the transmitter will receive the transmitted sequence and decode it as a control command. Having two controllers and two UAVs in a close vicinity creates interference to each other. To avoid this, the 72 MHz band is broken into 60 channels, each having a bandwidth of 10 kHz. A complete list of channels with frequencies for the 72 MHz band is listed in [37]. Note that there is a 20 kHz safety margin between any two adjacent channels to account for non-ideal signal processing circuits like filters. Every radio controller needs to use several channels, with each channel controlling one function of the UAV. For example, the throttle is controlled on the radio channel 72.01 MHz, the pitch is controlled on the radio channel 72.02 MHz, so one controller could require use of frequencies between 72.01 MHz and 72.11 MHz. If an additional UAV must be operated in the same vicinity then it would be assigned frequencies between 72.21 MHz and 72.31 MHz depending on number of controls. A typical radio controller (e.g. the ones manufactured by Futaba) can support up to 8-10 channels. Assuming that the UAV uses all of the supported channels this would allow the operation of no more than 7-10 UAVs in a close vicinity.

The advantage of the 72 MHz transparent link is the UAV's fast response to the movement control commands. On Figure 2.2 the wireless link between the operator and the UAV is referred to as the transparent serial link and replaces a serial cable such as the USB or the RS-232/485 for an asynchronous data transmission. The link is transparent in a sense that there is no security, error correction, or routing used. Once the control signal reaches the receiver it is demodulated and sent to the appropriate motor controller. Because there are no error checking or security protocols, there are no additional delays. The disadvantage of the 72 MHz transparent link is having no data encoding against interference. The link is designed to transmit control signals over a small bandwidth. It is not possible to transmit multiple commands in a single transmission or compressed images over the 72 MHz radio frequency. There are other frequency allocations which can be used for the data transmission.

2.3 Emerging Civilian UAV Communication Architectures

Historically, UAVs were referenced to military applications only. However, since the beginning of the 21st century "miniature-UAVs", also called Micro Air Vehicles (MAV), started gaining paramount importance in civilian applications. The miniature-UAVs began to gain popularity starting around 2003, approximately the same time the wireless technology started to advance forward providing higher data rate wireless network cards. Specifically, the advancement of wireless technology operating in license free frequency bands called ISM, such as 433 MHz, 900 MHz, 2.4 GHz and 5 GHz. The ISM bands allow designing, testing, and research of communication links with UAVs in the civil airspace, without any license requirements. In comparison to the 72 MHz band, the ISM bands provide more bandwidth. The emerging civilian UAVs use the radios operating at one of the ISM bands to transmit the data (commands, telemetry, and images) to the GCS. Based on the type of control used for a civilian UAV they can be categorized into three different groups:

Manual During its entire flight, the UAV is controlled by the operator.

Semi-autonomous Only some aspects of the UAV's flight is controlled by the operator (e.g. take-off and landing).

Autonomous UAV is capable of performing the entire assignment without any input from the operator.

The simplest way to control a UAV while streaming sensory data from it can be achieved by using the 72 MHz controllers for the movement control, with 900 MHz and/or 2.4 GHz radios for the data transmission. The UAV communication systems used in [40], [54], and [66] show that a combination of 900 MHz and 2.4 GHz is a field-proven implementation to work for data links on civilian UAVs. The autonomous UAVs use only the data transmission radios. The 900 MHz radio modems are often used for the transmission of telemetry from the civil UAVs. Zhou et al. in [66] built a model aircraft using the

72 MHz and 900 MHz frequency bands. The system uses a 72 MHz controller for the manual flight control while the 900 MHz radio modem is transmitting and receiving low resolution video stream. The authors do not mention the specifications of their radio modems, and the video stream. The parameters such as the type of the modulation used, whether the data rate is constant or adaptive, and the transmitted frames per second would provide enough information to get an idea of how the same system would perform over a longer distance or if multiple UAVs were used as relays. There are other designs, such as [40], where the authors decided to store the recorded video on an on-board memory card and perform data processing once the UAV lands. The stored videos have high resolution but are not supported with real-time processing capability.

In [54] a group of researchers took a similar approach by using two 900 MHz and a 2.4 GHz radio modems on the Yamaha RMAX civilian UAV helicopter. Compared to [66], the data link in [54] provides higher data rates at a closer range because it switches from 900 MHz serial transmission to 2.4 GHz Wi-Fi. The authors specify a 20 miles radius as the achievable communication range when 900 MHz modems are used. Such range is usually achievable by using high-gain directional antennas on the GCS. Independent of antenna type and with fixed transmit power the 900 MHz wireless link can operate at much longer distances than the 2.4 GHz; however, data rates are much lower compared to 2.4 GHz. The authors specify data rates to be less than 11500 Bits per second (bps) for the 900 MHz radios at the maximum distance. This data rate is sufficient for streaming telemetry or controlling the helicopter. However, streaming 50-300 kilobyte compressed images would take 34-208 seconds at this rate. Within this time the environment around the helicopter would change. One of the goals for our framework in this thesis is transmitting compressed images with a minimal delay, in the order of few seconds, to support semi real-time data exchange between a dirigible UAV and the operator.

2.3.1 Regulations for ISM Bands

The amount of power allowed for wireless transmission of data inside the ISM bands is limited by government regulations. Because the communication distance directly depends on the transmission power, it is a challenge in achieving the long distance using the ISM bands with limited power. One way to overcome this challenge is using high-gain directional antennas at the GCS and keep them pointing at the UAV. The next two sections will discuss in more detail the use of high-gain directional antennas and the related challenges of their use. According to Industry Canada regulations, the maximum allowable Effective Radiated Power (ERP) is 4W for the frequencies between 902 and 928 MHz [20], which accounts for the gain of the transmit antenna. This regulation also holds for North and South America [13]. The ERP is calculated as follows:

$$ERP = P_{transmit} G_{transmit} = P_{transmit}[dBm] + G_{transmit}[dBm] \tag{2.1}$$

where $P_{transmit}$ is the power level of the transmitted signal and $G_{transmit}$ is the gain of the transmit antenna. Both of these parameters can be changed during the configuration, as long as their product satisfies the regulation. For example, a regular dipole antenna has $G_{transmit} \approx 2.1\ dB$ and can be used to transmit a signal with power no more than 33.9 dBm. The same system can use a directional antenna with $G = 16\ dB$ to transmit a signal with power no more than 20 dBm.

The regulations for the 900 MHz and 2.4 GHz bands are the same, which triggers the following question. Given a fixed amount of transmit power, why 900 MHz is a more suitable frequency for the long-range communication? The answer can be shown using the Friis transmission equation, which relates the transmitted and the received powers:

$$P_{receive} = P_{transmit} G_{transmit} G_{receive} \left(\frac{\lambda}{4\pi R} \right)^2 \tag{2.2}$$

where R is the separation distance between a transmitter and a receiver in meters, $G_{receive}$ is the gain of the receiving antenna, and λ is the wavelength in meters. From equation 2.2, the received power depends on the wavelength, hence the frequency. Assuming all the other variables are constant, different frequencies will yield different received power levels.

2.3.2 900 MHz vs. 2.4 GHz for Longer Range

Having shown that various distances can be achieved with different frequencies we now consider the 900 MHz and 2.4 GHz bands. A modern approach to exchanging data with a civilian UAV uses both ISM bands. The 2.4 GHz link use Wi-Fi technology and the 900 MHz uses radio modems with serial data transmission. When exchanging data with the GCS at a distance less than 100 meters Wi-Fi is used. Whereas for longer distances the radio modems are used. The signal power at 900 MHz frequency undergoes less attenuation than at the 2.4 GHz. This can be shown by calculating the signal loss or path loss due to the radio wave propagation in the line-of-sight through free space. The free space path loss is the ratio of the transmit power to the received power in dB:

$$\text{Path Loss} = \frac{P_{transmit}}{P_{receive}} = \left(\frac{4\pi R}{\lambda}\right)^2 \tag{2.3}$$

The wavelength, λ, is inversely proportional to the frequency, $\lambda = \frac{C}{f}$, where c is the speed of light and f is the frequency. Hence, increasing the frequency in equation 2.3 also increases the path loss in free space. Substituting $\lambda = 0.33$ (wavelength at 900 MHz in meters) and $\lambda = 0.125$ (wavelength at 2.4 GHz in meters) in equation 2.3, then equating the two expressions shows that the difference in path loss between the two frequencies is 8.6 dB (Path Loss at 900 MHz $= (2.7)^2$ Path Loss at 2.4 GHz, and $2.7^2 = 8.6$ dB). Therefore, for the same distance the wireless link at 2.4 GHz will experience 8.6 dB lower received signal power than at 900 MHz. In terms of the distance, the 900 MHz wireless link would operate over seven times longer range than the 2.4 GHz link.

The antenna gains in equations 2.2 and 2.3 depend on the type of antennas. There are two types of antennas used in wireless communication; omni-directional and directional. Directional antennas are used in the GCS and omni-directional antennas are used on-board of the UAV. The challenge in designing a system with a directional antenna is that the GCS must precisely track the UAV in the air in order to maintain a reliable link. This scenario is illustrated on Figure 2.3. On the Figure 2.3 the dashed line illustrates directional antenna's signal beam. The point marked as a target must always point at the UAV, otherwise it is very hard to detect the received signal from the UAV. Therefore, the signal beam of the antenna must follow UAV's movement in space. One of the most common and simple tracking techniques is pointing antenna at the Global Positioning System (GPS) coordinates reported by the UAV. There are challenges associated with building a tracking mechanism and the next section present the results of existing tracking mechanism designs.

2.3.3 Antenna Types in UAV Communication

Directional antennas are preferred in a long-range communication because they decrease the interference by focusing the received or transmitted power in a specific direction. Omni-directional antennas are subjected to greater interfering signals since they receive signals from all directions around them. There are designs which use an omni-directional antenna on the fixed-wing model airplane and a tracking high-gain directional antenna

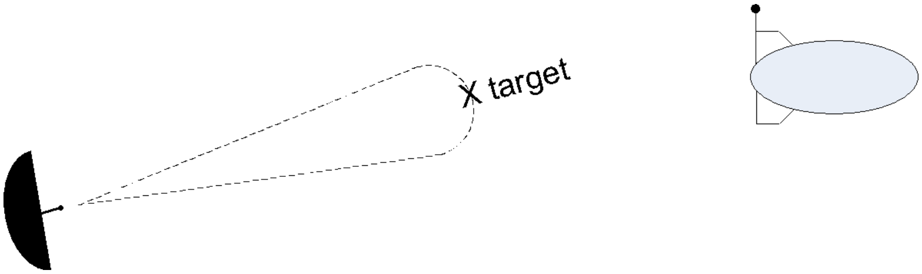


Figure 2.3: UAV tracking high-gain directional antenna scenario

on the GCS. Such design is demonstrated by Jenvey et al. in [50]. In their system design, the authors use a rotating mechanical platform with a 2.4 GHz reflective antenna to keep a track of the UAV in space and point the antenna at it. As the above results show there are several challenges associated with such approach. The main challenge is keeping the antenna's signal beam always pointing at the flying UAV, as illustrated on Figure 2.3. At a distance between 50 to 700 meters their tracking was successful. At longer distances than 700 meters, the tracking mechanism kept losing the UAV due to the very weak received signal. Because the tracking solely relies on the received signal, the antenna must maintain a constant communication with the UAV. This suggests that this tracking system is well suited for an open-field type of environment with a constant line-of-sight. Their tests were done when the tracking started from the UAV's take off. There are not enough results to show the performance of the system when the signal from the UAV is lost. With our design we intend to restore the broken communication by introducing a new intermediate node either on the ground or by lunching another dirigible.

Another important issue to consider with a tracking directional antenna is the stabilization of the rotating platform. With an unstable rotating platform the data communication with a UAV will keep breaking because the directional antenna's signal beam can not constantly point at the target. Balzano et al. solve this problem in [3] by using an advanced gimbal system. The estimated cost of their tracking system starts at $9,000.

In order to gain a better understanding of which antenna type is suitable for a moving UAV, Park and Jung have performed a simulation study in [38]. In their study they used a stationary directional antenna and an omni-directional antenna. In the simulation they analyse the performance of these antennas when a UAV is passing by. This scenario is illustrated on Figure 2.4. In their test, they recorded the duration in which the

UAV was communicating with the antennas. This time frame was used to calculate the transmission range of each antenna. The authors concluded that the omni-directional antenna maintained the communication with the passing by UAV 15% longer than the directional antenna. From these result we conclude that in a mobile UAV environment omni-directional antennas perform better in maintaining a communication at a close range.

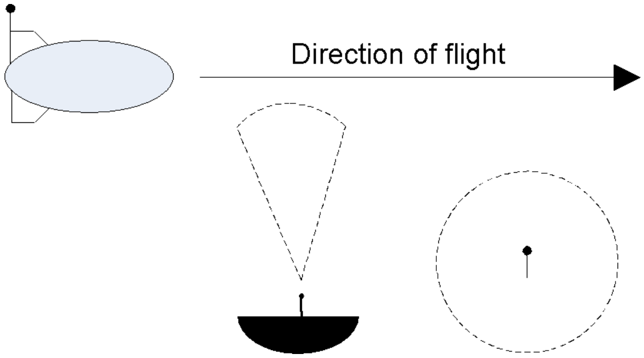


Figure 2.4: UAV passing by two stationary antennas

When analysing the combination of the above results from [50], [38], and [3] the resulting conclusion is that the directional antennas require maintaining a constant line-of-sight and must be precisely pointed at the UAV. Such accuracy is required because the directional antenna's signal beam is narrow, maximum 10 degrees. Maintaining the UAV within this 10 degrees at a line-of-sight becomes a mechanical engineering task. In the communication framework built for the dirigible UAV fleet we have used omni-directional antennas and assumed a line-of-sight due to the high flying altitude. Even if the line-of-sight is not available (e.g. dense area with high-rise buildings like a downtown), data can be rerouted through the mesh network without a need to keep tracking each dirigible.

2.4 Beyond Line-of-sight Communication - Multiple UAVs

Limited transmit power and the fixed frequency bands are the main limiting factors in achieving beyond line-of-sight communication links. Commercial solutions offer satellite communication equipment. Satellites are used as relay stations; transmitting data between two points on the ground. This allows achieving beyond LOS communication between a GCS and a UAV. However, communication with a satellite depends on satellite's location for a reliable communication and a clear view of the sky. It also requires frequency license in each country for operation and carries monthly costs for transmitted data. Alternative to satellite communication, beyond LOS communication can be achieved using multiple UAVs for relaying data among themselves. Multiple UAVs cooperate to relay the transmitted data to the destination using the 900 MHz or 2.4 GHz links. Such networks can be referred to as multi-hop ad hoc networks, wireless sensor networks, or wireless embedded networks. The achievable distance depends on the ad hoc routing protocols used, the power management configuration, and type of antennas used on the individual wireless nodes.

2.4.1 Cooperative UAV Networks

The real-world application results show that individually there are many civilian UAVs which are reliable in operation but lack communication with other autonomous vehicles. A group of UAVs can communicate among themselves using an ad hoc network. In this case each UAV plays a role of a "smart repeater". "Smart" in this context means that each node knows how to reach the destination using some kind of a routing method. Most of the COTS radio modems support point-to-point, point-to-multipoint, and repeater type connections. However, they are unable on their own of finding a route between a source and a destination, i.e. determine the next hop to the destination. An external software and hardware are required for performing the routing. The performance of the ad hoc routing protocols and network's scalability depend on the number of UAVs in the network, the network topology, whether transmit power is adjustable or not, the UAV's speed, and the environment in which they are operated.

The need to have cooperating UAVs mainly stems from search and rescue missions. These situations have the requirement to cover multiple geographical ground and aerial regions simultaneously. For example, Rohde et al. in [47] have developed a system for optimized spatial sensor coverage using an autonomous swarm of Micro UAVs. Their system relies on the Received Signal Strength (RSS) and GPS data being shared between neighbouring nodes and they point out that the existing control system and the communication method are still not working together effectively. Goddemeier et al. in [16] have proposed a distributed algorithm that achieves fast exploration of a given area and also relies on inter-UAV connectivity. Vachtsevanos et al. in [61] have summarized a list of data generally required to be shared among multiple UAVs in their control systems. This also opens possibility of collaboration between ground and aerial robots. An excellent example of search and rescue is the earthquake that happened in the Spring of 2011 in Japan. As Levy summarizes in [28], the rescue teams experienced problems with communication between air and ground autonomous vehicles and emphasizes the requirement for more research in this area. The article mentions the successful application of the Honeywell's T-hawk MAV and Helipse's unmanned helicopters, which use military-level radios manufactured by Cobham for the transmission of data. The article then concludes with expressing a need for having a relay communication between multiple non-military autonomous vehicles. In this thesis we attempt to address this problem by building a communication framework which can help multiple robots collaborate using a multi-hop ad hoc network with routing capabilities.

2.4.2 Inter-UAV Communication - Problem Definition

The inter-UAV communication challenges are described in more detail by Ryan et al. in [49] where the existing control algorithms for the non-military UAVs are summarized. Authors point out that the wireless communication between individual UAVs remain an issue and a verified implementation is yet to be found. The requirements for the UAV-to-UAV links are:

- Software deployable into multiple UAVs
- Reliable transmission with low transmit power and omni-directional antennas
- Using variable size data packets for transmission of telemetry and image data
- Maintaining link reliability as the network scales with the number of UAVs in the group

The first step in establishing a communication among multiple UAVs is implementing a software that handles data transmission between them. Today, most of the autonomous civilian UAVs use autopilot systems, the most common of which are ArduPilot, based on ATMEGA microcontrollers and Paparazzi based on STMicroelectronics STM32F series microcontrollers. Both types are hobby level controllers that connect between the on-board servo motors and the 72 MHz receiver. This provides on-board processing of data from sensors while maintaining manual flight control link with the 72 MHz transmitter on GCS. Depending on the choice of radios and the programming these autopilot systems can be made to communicate among themselves. This is left as an open source task for the designers. A more advanced autopilot system called Piccolo, built for military UAVs, working on 900 MHz, and providing centralized communication architecture was used by Matczynski in [33]. He proposed an embedded software architecture for use in multiple UAV control. His design was based on a Linux platform. The author points out that the main design limitation came from the network's star topology of the Piccolo autopilot communication; all the messages must be routed through a central node that manages the mission from the ground. A suggestion was made for future development to have direct links among UAVs, eliminating the need of routing through a central node. A similar communication system for multiple UAVs has been proposed by Chen et al [22]. Using radio modems at 900 MHz the authors introduce a communication method which attempts to resolve issues of data collision and timing control when multiple UAVs are transmitting. In their design the central node traverses a list of UAVs and allows each one to transmit one packet within 100 milliseconds on the network. Every node that wants to transmit information must wait for its turn and consecutive packets are delayed for at least 100 milliseconds. In both designs, [33] and [22], data collision is avoided but the speed of communication is limited by the central node that controls all the traffic. The goal for our communication framework is to allow the parallel transmission of data among multiple pairs of dirigible UAVs in a decentralized network. Collision of data packets in our network is handled at each node.

The authors in [49] suggest that a communication framework must be built to provide the application programming-level control and cooperation among several UAVs. Such framework would allow point-to-point communication between any two UAVs and should be simple to integrate into different UAV platforms. The authors do not provide details about the possible design of the framework. Instead, they focus on studying existing multiple UAV flight control algorithms. Vachtsevanos et al. in [61] later performed a similar review but with the addition of creating simulation models for the control algorithms of multiple-UAVs. However, the models suggested do not include the simulation of the communication links. These studies, show that currently there is a necessity for a reliable communication framework for multiple civilian UAVs. Thus, part of our goal is to design and implement the wireless communication framework with the appropriate Application Programming Interface (API) that addresses some of the above mentioned requirements.

The performance of the cellular links in the UAV environment has been explored by Goddemeier et al. [15]. The authors performed field-trials using an aerostat with sensors and a fixed-wing UAV to measure cellular link performance at different altitudes. Their analysis is based on RSS values. The article describes changes in the RSS values as the UAV's height changes. The authors show that as the UAV flies higher the RSS values decrease and the maximum height for a reliable link was determined to be around 500 m. In a cellular network, the transmit antennas used on towers are directional and point toward the ground; therefore, as the cellular receiver is raised it goes outside of the coverage area. This result reveals how cellular technologies can be good candidates for the inter-UAV communication in altitudes up to 500 meters. Wzorek et al. in [64] demonstrated a control of a Yamaha RMAX helicopter through a mobile phone application. A helicopter was operated at an altitude of 25-35 meters, which is in harmony with the cellular link measurements in [15]. It is important to point out that the above mentioned multi-UAV communication techniques rely on the availability of cellular service in their area.

The ad hoc network routing protocols and the network topology characterize the network's performance. The routing in an ad hoc network is challenging because the network's physical infrastructure is not constant in space. One of ad hoc network's characteristics is the fact that its topology is constantly changing due to the movement of nodes in the space or the change in radio signal's propagation conditions. Propagation conditions affect the RSS. The analyses in [15], [64] indicate that a wireless link's performance, such as transmission delay and successful delivery rate, depend on the RSS. In some situations there is no accurate way of measuring the delay or the success rate; therefore it is sufficient to measure the RSS values to estimate the performance of the link. In addition, the ad hoc networks have limited bandwidth and may not have a radio line-of-sight. As a result, a centralized routing hierarchy with preassigned routing functionality nodes are ineffective. All of the nodes should be able to detect their neighbours and perform the routing [5]. This provides alternative propagation paths, increasing the reliability of data transmission.

2.5 Network Communication Standards

The network standards implemented by the lower layers in a multi-layer system have an impact on the achievable communication range. Currently there are two widely accepted network standards for 900 MHz and 2.4 GHz that provide specifications for the physical and data link layers. For the purpose of this research project, the key aspects that differentiate these standards are the achievable data rates and the communication range.

2.5.1 IEEE 802.11

The IEEE 802.11 standard, known for use in Wi-Fi technology, is a collection of specifications for development of physical and media access control (MAC) layers in networks. The MAC layer has a common implementation across several IEEE 802.11 version for accessing channel. At the physical layer the standard uses two techniques for the transmission of radio signals: direct sequence spread spectrum (DSSS) and frequency hopping spread spectrum (FHSS). In the context of this thesis, the difference between the two techniques is the data rates. The processing of radio signals with FHSS can be done for data rates under 1-2 megabits per second. For higher data rates DSSS is used. The DSSS standard started with support for 1-2 Mbps and now it has been increased to 54 Mbps. As the standard was revised to increase the data rates, the manufacturers of Wi-Fi network equipment use DSSS.

The IEEE 802.11 standard defines two modes in which Wi-Fi equipment can operate - “Ad hoc” and client/server (also referred to as infrastructure mode). It is the Ad hoc mode that made the 802.11 based equipment attractive for multi-agent system applications and their research. Iqbal et al. in [21] provides a list of mesh networks which have been implemented using the 802.11 standard. One of the implementations, MobiMESH in [7], talks about the challenges of practical implementation of the mesh networks based on 802.11, specifically the design limitations of the standard and interference between two routing nodes.

The design limitation of the 802.11 standard is the adaptive modulation techniques which affect the data rate. Briefly, the adaptive modulation measures the RSS and chooses the appropriate modulation technique that minimizes errors in data transmission - the closer a receiver to a transmitter, the higher the data rate can be used. The different modulation techniques used in Wi-Fi do not perform well in a power limited environment, which was demonstrated by the results in [7] and the long-range communication is a power-limited environment. Wirtz et al. in [63] implemented an IP-only based multi-hop ad hoc network using the infrastructure mode of the 802.11 standard. In this setup the routing nodes do not perform route discovery because the routes are already known through the use of static routing. The paper also mentions that a new 802.11s standard is currently under development to support a mesh network structure.

The final remark about the existing Wi-Fi based long-range mesh designs is the fixed node locations. Today Wi-Fi mesh implementations are mainly used to provide connectivity to rural regions. In this application the distance and location of the points are fixed. Examples of this Wi-Fi based mesh implementations are Tegola Tiered network in Scotland [4] and WiLDNET [41]. In both implementations high-gain directional antennas are used. Authors in [41] mention that with omni-directional antennas Wi-Fi mesh can work over distances 1-2 km at the most.

2.5.2 IEEE 802.15.4

Wireless networks based on IEEE 802.15.4 standard are an alternative to wired connections in the distributed systems for monitoring and control. Similar to 802.11, the 802.15.4 standard provides specifications for physical and MAC layer development, but it uses an IEEE 64-bit addressing scheme. The 802.15.4 based networks provide a more flexible architecture, simpler in installation and operation compared to the 802.11. Similar to Wi-Fi being the accepted technology for development with 802.11, ZigBee is the accepted technology for development with the 802.15.4. The difference is that ZigBee targets low power consumption to save energy because it is designed to be powered from a battery. Hence this results in a low transmit power and a highest data rate of 250 kbps in the 2.4 GHz band.

In comparison to Wi-Fi, ZigBee works in one mode and it uses a tree mesh topology. ZigBee's low-power consumption and the mesh network topology has been used to implement wireless sensor networks and the related network testbeds based on 802.15.4 standard. Due to a low power consumption the achievable communication range is 10-50 meters. The main applications are in home appliance automation, alarm systems, remote control, and sensor networks. Singh et al. in [55] describes a civilian UAV system with 2.4 GHz embedded radios, built by Digi International, that employs ZigBee standard for transmitting data from sensors. Pekhteryev et al. tested a transmission of JPEG and JPEG-2000 compressed images over a ZigBee network in [42]. His conclusion was that JPEG-2000 compressed images were transmitted with less errors.

The 802.15.4 standard with ZigBee has been used for the development of wireless sensor network testbeds. In particular the CC2430/CC2420 radios in 2.4 GHz by Texas Instruments have been used in numerous research projects on developing ZigBee wireless sensor networks [62], [44], [65]. So far the developed testbeds based on 802.15.4 can work over distances up to 100 meters or multiple floors. The radios being used in our communication network implement the 802.15.4 physical layer standard because it allows making use of the binary Frequency Shift Keying (FSK) modulation for achieving longer communication distance than with the 802.11 standard.

2.6 Ad hoc Routing Protocols

Ad hoc routing protocols started to receive more attention when several mobile computers equipped with wireless interfaces had to be interconnected. Ad hoc routing implies a union of several mobile devices with wireless network adapters into a single data network. By definition there is no centralized network management, in addition, such networks should be self-organizing (self-adjusting). Achieving a self-organizing ad hoc network can be accomplished by implementing a routing protocol on some or all network nodes. The are three categories of routing protocols for the ad hoc networks [48], [27], [36], [17]:

1. Proactive protocols - routing tables are used on all of the nodes and periodically updated. Routes are already known to the destination before the data is ready.

2. Reactive protocols - routes are established “on-demand”; when data is ready for transmission. To establish a route, the sending node transmits a broadcast request message, which must reach the destination node. In response, the recipient sends a confirmation message from which the sending node acquires the route information and saves it in its routing table. For the consequent transmission of data the routing table is used. If a route is found to be broken, the protocol initiates a route maintaining procedure, which finds a new route to the same destination.

3. Hybrid protocols - a combination of the proactive and the reactive protocols. An example of that is a large network broken down into smaller subnets. Each subnet uses the proactive protocol while the reactive protocol interconnects the subnets.

Each of the three categories has its own advantages and disadvantages. The proactive protocols use the least time to prepare routes. In comparison to the reactive protocols, the proactive protocols take away part of the total bandwidth due to periodic routing table updates, which use broadcast messages. Another important fact to consider is the power consumed when using these protocols. The reactive protocol uses less power compared to proactive because it only performs updates when required. The proactive protocol consumes power whether transmitting actual data or periodically updating the routes.

At the moment, there are several protocols developed for the first two categories [5], [9], [19]:

- Ad hoc On-demand Distance Vector (AODV) - A reactive protocol
- Dynamic Source Routing (DSR) - A reactive protocol
- Optimized Link-State Routing (OLSR) - A proactive protocol
- Destination Sequenced Distance Vector (DSDV) - A proactive protocol
- Open Shortest Path First (OSPF) Mobile Ad hoc Network (MANET) - A proactive protocol

Brown et al. in [6] implemented an “Ad hoc UAV Ground Network (AUGNet)” using the DSR protocol and the IEEE 802.11b wireless interface. In their paper they also demonstrate that there is no significant mutual interference between 900 MHz and 2.4 GHz equipments mounted in the same system. Their tests were directed at measuring the performance of the ad hoc network for the transmission of voice and image data. They performed tests with and without a UAV and the results show that using just one UAV had improvement on the network’s reliability and the range due to a better radio line-of-sight. The authors point out that the DSR protocol took a long time for finding new routes when the previous ones were broken. Each node in the AUGNet has routing capabilities; therefore, it uses a mesh network topology. Larsson and Hedman had shown in their simulation studies [26] that the DSR and the AODV protocols’ performance depend on the lower layer’s implementation and the network’s topology. They varied the number of the radio nodes from 4 to 10 with randomly chosen nodes moving at speeds from 0 m/s to 20 m/s. Their conclusion was that in a scenario with 95% of nodes moving at speeds of 1 to 5 m/s and the remaining 5% of nodes moving at a speed of 20 m/s the received number of packets was in the range 52% to 58%. This is due to their design, which performs routing based on the IP only, without any support from physical or link layer.

2.6.1 Power Management in Ad hoc Networks - Self-interference

One of the important issues to consider in a multi-hop ad hoc network is the self-interference, arising from all the nodes communicating on the same frequency channel and using the same transmit power. Elliot et al. in [12] have introduced the self-interference issue in the application of the proactive and the reactive routing categories in scalable networks. The paper also summarizes one of the important issues in ad hoc networks; interference caused by transmitting data at the highest possible power level. Using the highest transmit power allows reaching the destination directly from the source. However, this may create interference caused by the radio nodes communicating closer to the receiver than the transmitting node. If the transmitter power is constant in a multi-hop ad hoc network, the nearby radios produce signals with higher power levels making it harder to detect weaker signals arriving from a further distance. This interference could be reduced if the radios could adjust the transmit power before transmitting each packet or a group of packets representing a single data structure [53], [57], [59]. By reducing the transmit power the data is transmitted through the intermediate nodes, multiple hops, to the destination. Other solutions include physical channel access techniques for determining best time periods for transmission of data. Adjusting the transmit power for each packet can be done at the network or application layers while the channel access techniques requires access to the physical layer, which is not always available for programming. In our communication framework we provide the dirigible UAV with the capability of programming the transmit power at the application layer.

2.6.2 Ad hoc Network Topologies

There are four network topologies that are possible to implement with the currently available commercial radio units. Figure 2.5 summarizes their configurations.

The star and the tree topologies pre-assign functions to the nodes. In the star topology the centre node represents a “master” through which all the transmission takes place. This topology was used in [33] and [22]. The obvious disadvantage of this topology is that if the master node is down the entire network collapses. Another popular standard is the Bluetooth protocol stack. It is designed to use the star topology. According to the Bluetooth specifications, up to seven “slaves” can connect to one master node [56]. Hoffmann et al. from Stanford University [18] developed a test bed for multiple UAV control using the Bluetooth standard. The Bluetooth can provide up to 2 Mbps rate, but the communication range per link is at most 10-15 meters. Another disadvantage for

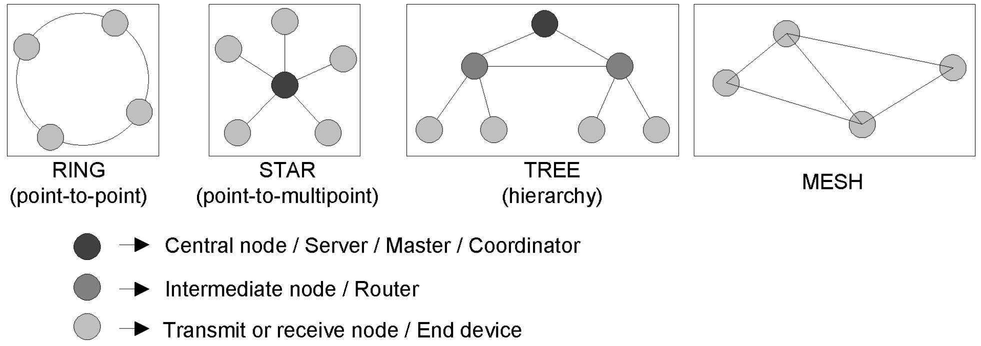


Figure 2.5: Network Topologies

the Bluetooth's application in multiple UAVs is having no backup links, since all links have a single path. If the propagation path is obstructed, the communication is lost.

The tree topology is used by the ZigBee, IEEE 802.15.4, standard. ZigBee also has a pre-assigned functionalities to the nodes. The root of the tree is called "coordinator" in ZigBee, the intermediate nodes which have routing capabilities are called "routers", and the nodes connected to routers are called "end devices". The end devices can communicate only with the parent nodes; therefore, communication between end devices require presence of the router nodes. ZigBee provides data rates up to 250 Kbps, which is less than the Bluetooth, but the communication range per link can be on the order of few hundred meters. A research team from India demonstrated their attempt in [55] to build an intra-UAV communication network using ZigBee modules, which are the Xbee radios manufactured by the Digi International. The team uses the Xbee modules with directional YAGI antennas for the transmission of the GPS, accelerometer, and gyroscope data between computing platforms working on an Atmega 128 micro-controller. They achieved an increase of 70 meters by using YAGI antenna instead of a monopole wire antenna. However, they do not explain how they will be tracking their UAV in the air with the directional YAGI. Their research work is still in progress.

The ZigBee standard uses the AODV protocol and provides a scalable network. A larger area can be covered by adding more routers with end devices. This is an improvement on the Bluetooth. The disadvantage of ZigBee is similar to that of the Bluetooth's. The ZigBee router with several connected end devices can be viewed as having a star topology. In addition, increasing the number of end devices without increasing the number of routers degrades the overall network's performance. Xiao et al. in [65] simulated ZigBee's performance with a variable number of end devices. The authors compared ZigBee against the normal AODV. The difference between the two is that, in the normal AODV, each node can perform routing, while in ZigBee only the router and the coordinator nodes perform routing. The results of the simulation from [65] show that as the number of ZigBee end devices is increased from 20% to 50% the successful packet delivery drops below 20%. The normal AODV routing maintains almost a constant rate around 60-80%. Since each node in the AODV can route data, it provides alternative propagation paths, which improves the reliability of the network.

2.7 Chapter Summary

The advancement of the wireless technology is making it possible to operate multiple civilian UAVs over extended distances with obstructed line of communication. The traditional radio controlled hobby model aircrafts are evolving into a technology that can be applied for critical missions such as security monitoring, cooperative search and rescue missions. As it has been shown, several research teams have successfully built civilian UAVs using unlicensed spectrum of frequencies for the wireless data transmission. The regulations for the wireless communication in the ISM bands prevent achieving longer distances with a single UAV.

Multiple UAVs are starting to be used for extending the communication range by forming ad hoc networks for achieving scalable communication links. Since there are no accepted standard for ad hoc networks, each of the designs is different. Due to the unpredictable operating environment one of the main challenges in ad hoc communication is maintaining link reliability and the capability of fixing broken links. Several approaches and implementations have been demonstrated, each one addressing specific issues of ad hoc communication. Despite these advances, there is still a need for building a software application-level communication framework that is UAV platform independent and which provides options for further development in the multiple UAV networking for the civilian airspace.

Despite the large volume of research conducted in different aspects of UAVs, such as control and pilot systems for example, there is still an urgent need for a long-range wireless communication coverage to secure the transmission of data between UAVs and/or between them and a ground base station. In the following chapter, we present our implemented communication framework's hardware and software structure.

Chapter 3

Proposed Framework and Methodology

The challenges and requirements of creating a multiple UAV communication environment were outlined in Chapter 2. This chapter provides the details of the design and the development of a new communication framework which allows multiple UAVs to exchange data and provide support to their existing control algorithms. Also, the framework can be used as a testbed for an evaluation of new emerging routing protocols in a real environment. The research process consists of drawing up a set of high-level requirements for the framework, investigating and choosing, based on these requirements, a radio technology, developing the software for the framework's integration into a dirigible UAV control system, and running real-time tests for measuring some key performance parameters. Real-time tests are performed indoors and outdoors to find out the scalability of the network.

In order to support the development of an autonomously operated fleet of civilian UAV dirigibles at the University of Ottawa, a first version of the communication framework for the control of the fleet was designed and built. The framework does not use the existing communication infrastructure such as cellular towers or satellites. As it was explained in section 2.4, satellite communication can get expensive for transmitting images and requires equipment with more weight than the allowed payload per the dirigible UAV. Cellular technology is not available everywhere (e.g. operating the dirigible UAV fleet over the sea or in rural areas). Therefore, our design incorporates the hardware and the software components required for establishing a long-range wireless communication between all the dirigibles and the GCS, independent of the surrounding infrastructure.

The framework is not designed to operate as a stand-alone program. It requires a main control process such as one of the previously mentioned UAV autopilot systems or an on-board operating system in charge of the flight control. The main task of the framework is extending the radio control of a civilian UAV by relaying the commands in an ad hoc network using a novel mesh routing protocol called DigiMesh$^{\text{TM}}$. When a transmitter and a receiver are separated far enough, and in the case when the direct communication link does not work (e.g. no line-of-sight), a multi-hop pattern is used for relaying the data. The other task of the framework is providing each of the dirigibles in the fleet with the following capabilities:

- Establishing connection with any of the dirigibles in the wireless network, provided that its within the range
- Identifying itself by sending its own network address and GPS information to the dirigible requesting it or to the GCS
- Exchanging text commands, pre-recorded audio and video data with a required dirigible in the network using a reliable connection
- Handling the reception of data from multiple sources - aggregate node functionality
- Adjusting the transmit power on the locally connected radio unit or remotely on an other dirigible
- Notifying the control station when the communication with it is about to break, based on RSS value

In addition to the above, the framework allows the GCS at any given time to find out all the dirigibles connected to the ad hoc network. This functionality is useful for determining the geographical area currently being covered by the fleet based on the reported GPS data. Another application of this function is estimating the maximum range of communication that can be achieved between the GCS and the farthest dirigible when the fleet is aligned in a daisy chain formation. In this scenario the assumption is that the RF LOS exists or can be achieved between mutually connected dirigibles, while the source and destination nodes may not have a direct RF LOS.

This thesis focuses on the system design and integration of the proposed wireless communication framework into the existing dirigible UAV architecture. All the hardware components used in the framework are commercially available. Once the framework structure is finalized it can then be optimized at a later stage.

3.1 System Hardware

The initial plan for communication with the SPOT was to use a walkie-talkie based system because its frequency band does not require a license. The advantage of this technology is the simplicity of radios and ability to communicate with all the radios within a certain range (broadcast). The walkie-talkie technology is similar to using point-to-multipoint radios at 900 MHz. The disadvantage of this setup is that the transmit power in the frequency band for a walkie-talkie is limited by regulations to 0.5 W and there is no routing to address messages to a specific radio node (unicast). For this reason the design goal was made to find a COTS radio technology working on the 900 MHz frequency band and provides some type of mesh routing.

3.1.1 UAV Dirigible - SPOT

The proposed communication framework for the control of multiple UAVs can be employed in different applications with different aerial platforms. But above all, this solution, is developed for use on the dirigible SPOTs being developed at the University of Ottawa and shown on Figure 3.1. A full description of SPOT dirigible's design, including the power system parameters is available in [45]. Each SPOT dirigible is equipped with a compact desktop computer manufactured by VIA Embedded, called ARTiGO A1100 in a Pico-ITX form factor. Most of the on-board electronics except for the Wi-Fi utilize a Universal Serial Bus (USB) connection as illustrated on the system block diagram in Figure 3.2. Having both types of serial interface ports Recommended Standard 232 (RS-232) and USB allows the connection to the radio through either one with a proper conversion between the radio's asynchronous data output and the serial interfaces on the computer.

As it can be seen from Figure 3.1 dirigible's gondola houses the engine and the required electronics for the flight mission. Initial tests have shown that the engine caused electromagnetic interference affecting digital compass readings. For this reason a decision was made to place the omni-directional antenna on the top most fin, pointing upward. Placing the antenna far from the receiver and data processing computer and connecting them with a long RF cable from the antenna will introduce an insignificant signal at-

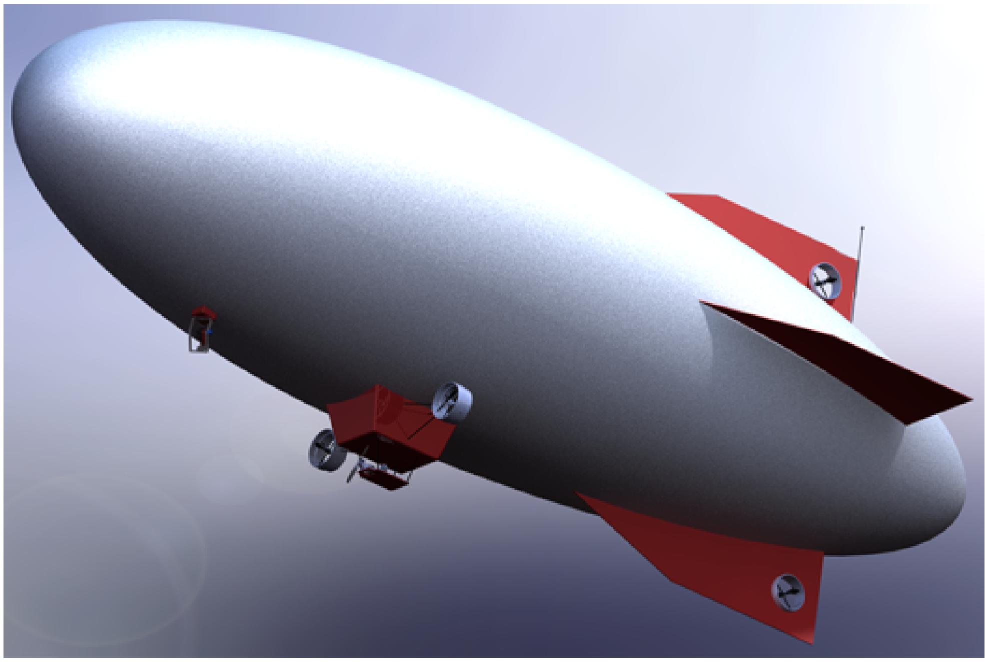


Figure 3.1: SPOT UAV dirigible

tenuation. The radio and the interface converter units are small enough that they can be attached to the fin. Then either USB or RS-232 cable can be installed between the communication units at the fin and the ARTiGO computer. Since the dirigibles operate at high altitudes, such configuration would allow each dirigible to communicate with any other dirigible within RF LOS.

Once the design for a single SPOT is finalized, multiple SPOTs can be used to cover a large geographical area. Multiple dirigibles in the air can be used for developing a long-range communication by relaying the signal through a multi-hop wireless network between the GCS and the fleet. This is the concept used in the proposed framework. The design goal is maximizing the distance between SPOT dirigibles while maintaining a reliable communication link without violating government regulations for radio transmission in the ISM bands. The design process starts with approximating the amount of data to be transferred to/from each SPOT and finding appropriate COTS radio technology which can be used with some type of decentralized routing techniques.

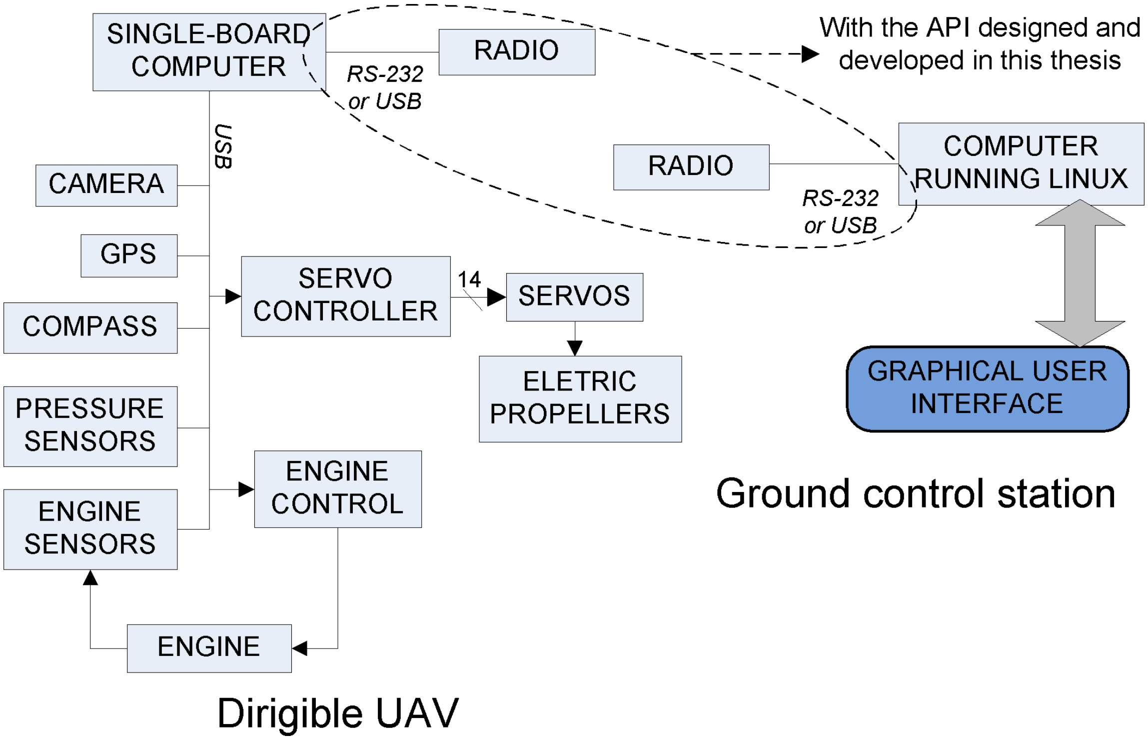


Figure 3.2: System block diagram of the SPOT

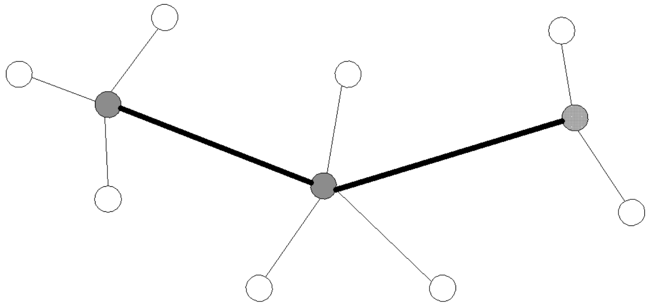
3.1.2 Radio System Selection

Choosing radio technology depends on the required throughput and the communication range the radio can provide. The system block diagram on Figure 3.2 shows the sensors from which data must be transmitted. This includes the GPS, the JPEG compressed photos, and the telemetry. Real-time video transmission is not supported by the framework for now. A real-time video stream requires high bandwidth, even with a maximum compression such as MPEG-4 part 10, the required data rate for a compressed video from a high-definition camera is around 1-3 Mbps and requires an additional radio unit dedicated to this task. The 2.4 GHz ISM band is typically used for video transmission at close range because it provides more bandwidth compared to 900 MHz band. There are commercially available radio technologies with charge-coupled cameras designed for wireless video streaming at the 900 MHz band. However, they are not designed for multi-hop networks. The implemented framework supports transmission of pre-recorded video files, which are stored on the on-board computer. The focus is on a low-bandwidth data exchange between any two nodes in the fleet of the SPOT dirigibles.

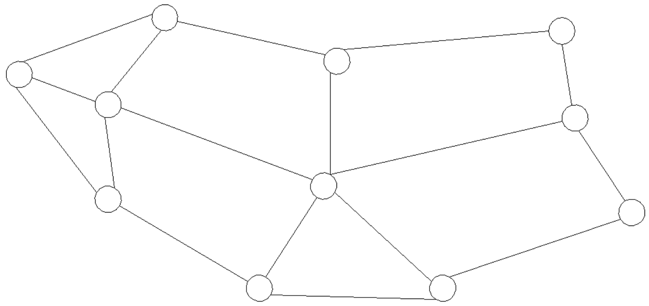
After a discussion with the SPOT dirigible's designers it was decided that all the data from the sensor would be sent to the GCS as frequently as the lowest data update rate among the sensors. The GPS unit has the lowest update rate of one second. Every second it automatically outputs the longitude and latitude coordinates. Together with all other sensors the expected data load for transmission is between 70 to 100 bytes per single transmit request, excluding high-bandwidth data (e.g. images and videos). Comparing this rate with expected data rates of 1-100 kbps for wireless embedded networks, as stated in [56], the SPOT's requirements for long-range communication can be satisfied using commonly supported serial link rates between 9600 bps and 115200 bps. There are two places in our system where the data rate can be controlled. The rate at which the on-board computer communicates with the radio through the USB or RS-232 and the rate of over-the-air transmission. It is important to keep track of the two rates because if they are not equal the software must use data flow control to avoid loss of data, which slows down the overall communication. The issue of configuring the two rates is discussed further in the section 3.2.3.

Multi-hop ad hoc networks usually are not designed to transmit any multimedia data. However, in a scenario when the fleet is operating beyond LOS it is helpful to send photos from different SPOTs to GCS. The serial link data rates may be low for a real-time video transmission but compressed photos can be received within few seconds. When transmitting still images the goal is to transmit as fast as possible with a reliable serial link design. Compressed images in the JPEG format are used by the camera on SPOT but this format is not error resilient. If the reliability of the link or the radio of successful data delivery to total data sent is low, then the received images will be corrupted.

The fleet of SPOT dirigibles is being designed to operate in different and unknown environments. This is a challenge for reliable data transmission and in order to provide reliable communication among individual dirigibles the design goal was made to use a mesh topology for a wireless network. As Robert Poor explains in [43], the mesh topology is well suited for networks independent of the environment. A decentralized mesh on Figure 3.3 illustrates how every radio node connects with all other nodes within its communication range. Such configurations provide more than one transmission path between a source and a destination, thus increasing reliability of data delivery at the physical layer. The design goal for the SPOT mesh radio network is better satisfied with the decentralized topology. A backbone would force transmitting radios to forward the message to the nearest backbone node every time before the message will be routed to the destination, this reduces the networks flexibility.



CENTRALIZED MESH CONFIGURATION
with a backbone route



DECENTRALIZED MESH CONFIGURATION
(true mesh)

Figure 3.3: Centralized vs. decentralized mesh topology

Currently there are two accepted standards which can be used for implementing a mesh network. The 802.11 standard mainly used in Wi-Fi technology or the 802.15.4 standard commonly used for building low-powered personal wireless networks with ZigBee. The Wi-Fi technology is often used for high data rate communication and for the ad hoc network research. The wireless network cards already come with support for ad hoc mode connectivity. However, this mode is not suitable for building mesh networks using Wi-Fi. The ad hoc mode can establish mutual links between nodes at the physical layer but it does not perform any of the networking functionalities. For example in a Windows or Linux operating system one can create multiple separate peer-to-peer connections, but the operating system does not link them. In other words it does not route packets between the connections. An additional software is required to resolve addressing and routing between these links. Authors of Ad hoc UAV Ground Network

(AUGNet) (decentralized mesh) implemented a custom DSR routing by using a software called "The Click Modular Router" from [23]. The click modular router is a software which provides programming tools for connecting the individual peer-to-peer connections, making it possible to develop custom routing techniques. Such mesh design can be summarized in terms of implemented layers as shown on Figure 3.4. Each of the nodes in the AUGNet network implements these three layers.

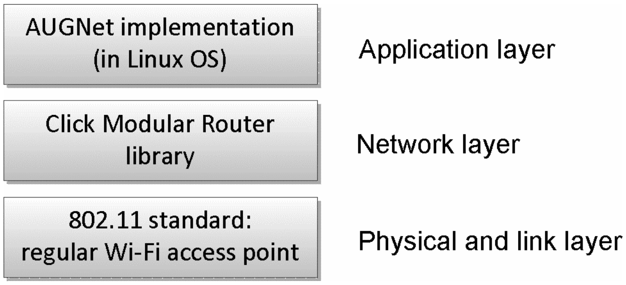


Figure 3.4: Summary of AUGNet

Wirtz et al. demonstrated a simpler approach to creating Wi-Fi based centralized mesh in [63]. Rather than building a custom routing protocol the authors used Wi-Fi cards with special drivers and software from Ubuntu Linux that allowed them to setup a mesh network using laptops and mobile phones only. Wi-Fi based mesh networks can not operate over the range larger than 3 kilometres based on the results achieved from AUGNet and [6] testing. The main limiting factor is the transmit power and the modulation techniques used in Wi-Fi, like Quadrature Amplitude Modulation (QAM), which are power inefficient but provide high data rates for a fixed bandwidth. For these reasons Wi-Fi based design for the SPOT has not been chosen. Instead a new mesh routing protocol based on 802.15.4 standard is used.

In contrast to Wi-Fi technology, there are many different higher level implementation available for the the 802.15.4 standard. ZigBee became an accepted implementation among many manufacturers of radios. The ZigBee is a centralized mesh routing technology which supports multi-hop links. However as the discussion in section 2.6.2 concludes, ZigBee does not scale successfully because only router and coordinator type nodes have routing capabilities. Furthermore, ZigBee is designed to use fixed transmit power levels between 50-100mW, as indicated in the specifications of the ZigBee radio manufacturers RFM$^{\text{TM}}$ and Digi International®. ZigBee technology provides a centralized mesh network design but achieving a long-range with ZigBee radios requires a use of the high-gain antennas. As an alternative to ZigBee, the same two companies offer different mesh implementations with different routing protocols. Among these the DigiMesh protocol together with the radio supporting it was identified as the best candidate for the development of the SPOT's communication framework.

3.1.3 DigiMesh Routing Protocol

The DigiMesh is a propriety protocol, developed and maintained by Digi International. The firmware implementing the protocol can be used either on Xbee® or on 9XTend™ radios, both compatible with ZigBee. Using either one will allow to switch between ZigBee or DigiMesh if required. This is achieved by changing the firmware on the radio. The physical and link layers are implemented according to the 802.15.4 standard. While Xbee radio modules are focused on providing fixed low power and over-the-air data transmission rate of 250 kbps, 9XTend radio provides adjustable power between 1mW to 1W, the longest range in its class of radios without high-gain antennas, and higher payload size per radio packet. The distinguishing feature of Xbee and 9XTend is the ability of configuring the radio parameters or exchanging data in the mesh without a need of switching radio's operating mode. The communication and the radio configuration are implemented using API provided by the radio's firmware. The 9Xtend radio connected to an external half-wave dipole antenna is shown on Figure 3.5.

The radio together with the DigiMesh was chosen based on the following criteria:

1. **Antenna type vs. Range and Data Rate**. Using a regular half-wave dipole antenna the 9XTend radio can provide a maximum communication range up to 10 kilometres per hop with the transmission rate of 115000 bps [10]. The radio achieves such long-range with a limited power because it uses a binary FSK modulation scheme, which is power-efficient [60].

2. **Packetization**. All the communication, whether it is with the radio or over-the-air, has been implemented through packets (datagrams). In comparison to Xbee radios which offer only 100 bytes of payload per packet, 9XTend offers 256 bytes of payload per packet.

3. **Routing Capability**. The DigiMesh routing has been implemented by the manufacturer based on a "Quiet protocol". As the specification in [10] states, this

Figure 3.5: 9XTend radio

protocol reduces the routing overhead. Consequently, this helps to maintain a higher data throughput in the mesh.

4. **In-flight Configuration**. The SPOT can change the configuration settings of the 9XTend radio at any time. This capability is mainly useful for adjusting transmit power levels in different network configuration scenarios. In addition to changing configuration of a locally connected radio, using the DigiMesh protocol a SPOT can remotely change configuration of other radios in the network. This capability can be used by multiple UAV control algorithms for minimizing the self-interference issue.

5. **Advanced Features**. The API for the 9XTend radio provides several features which can be used to manage the fleet and run diagnostic test of the network. These features include enabling per packet acknowledgements for a reliable data transmission, data collision handling algorithms that work at the physical layer, route tracing tools, making each radio node capable of discovering all the neighbours within one hop, and changing DigiMesh network ID as a security feature because only the nodes with the same ID can communicate.

Overall, Digi International provided the most detailed open-source technical information about the DigiMesh and the 9XTend radio's API, which was crucial for the development of our communication framework. The firmware that provides all of the above functionalities with the DigiMesh protocol is located inside the radio's embedded memory and the provided API allowed to develop the framework's software in C programming language.

3.1.4 9XTend® Long-range Radio Module

Three 9XTend radios have been used for the development of the framework. The radio can not be connected directly to the USB or RS-232 port of the on-board computer because it uses a Universal Asynchronous Receiver/Transmitter (UART) interface for the connection. The UART is a single-ended signalling method of data transmission while the USB uses a differential-signalling method and the RS-232 uses different logic voltage levels. A serial data converter must be used between the 9XTend radio and the on-board computer. In order to speed up framework's development and testing, two interface boards were used from the manufacturer. The interface board together with the connected radio is shown on Figure 3.6.

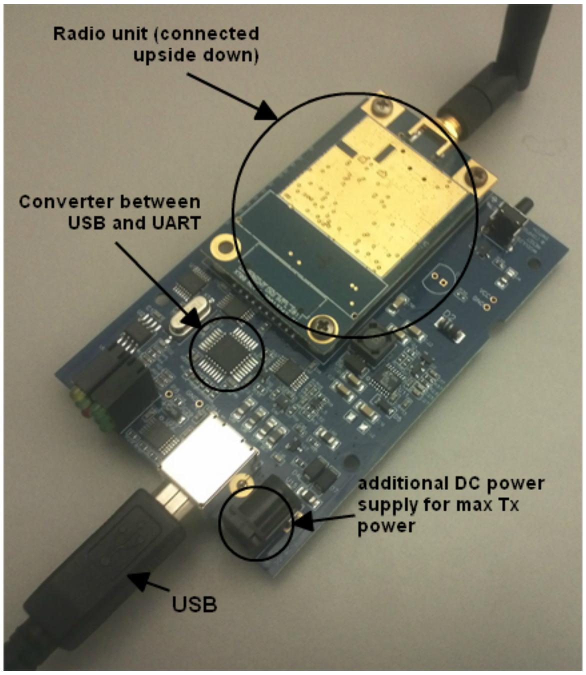


Figure 3.6: Interface board for 9XTend radio

Considering the limited space on-board of the dirigible, the interface board takes up unnecessary space for the main function it is performing. It was mainly purchased as a convenient development and testing tool. For the third radio we used a different and less expensive solution for the connection to the computer. We used a converter chip to connect from the UART to the RS-232, together with building the power supply circuit for the radio. The final circuit showing the power distribution and data communication is depicted on Figure 3.7.

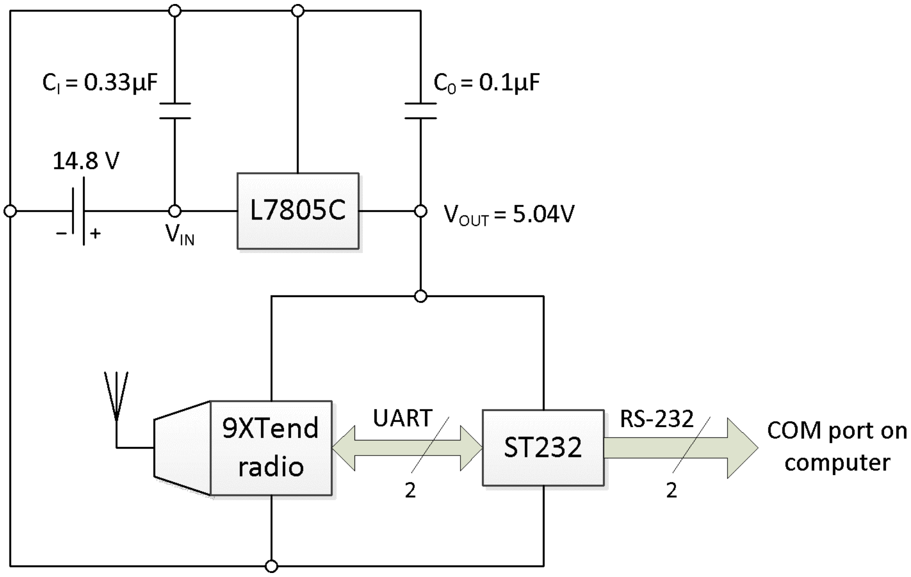


Figure 3.7: Circuit of a single radio node in testbed

The connection pins on the radio are grouped into a header with 2 millimetre pitch. This is a non-standard pitch (standard pitch is 2.5 millimetre) so a corresponding socket with a 2 millimetre pitch was obtained, as shown on Figure 3.8.

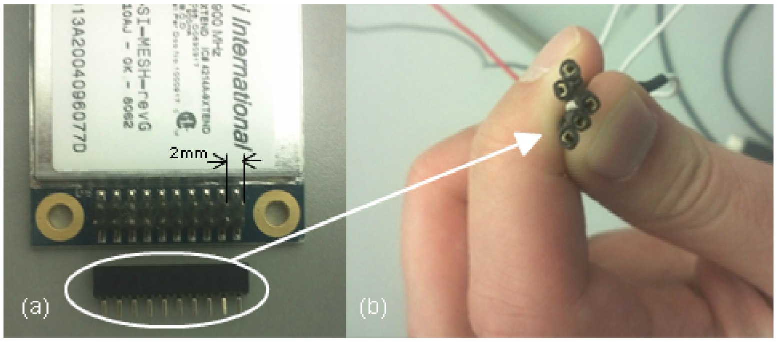


Figure 3.8: 9XTend radio custom connection

3.2 Hardware-Software Interfacing

Using the on-board ARTiGO computer the SPOT operates using its own implemented flight controller program, which runs on a Linux operating system. The flight controller program is written in C, therefore, the communication framework's software has also been implemented in C. In order to integrate the hardware setup from the previous section into the existing electronic platform of the SPOT, an API in C was developed. The API works as a bridge between the flight control program and the radio's API, as illustrated on Figure 3.9.

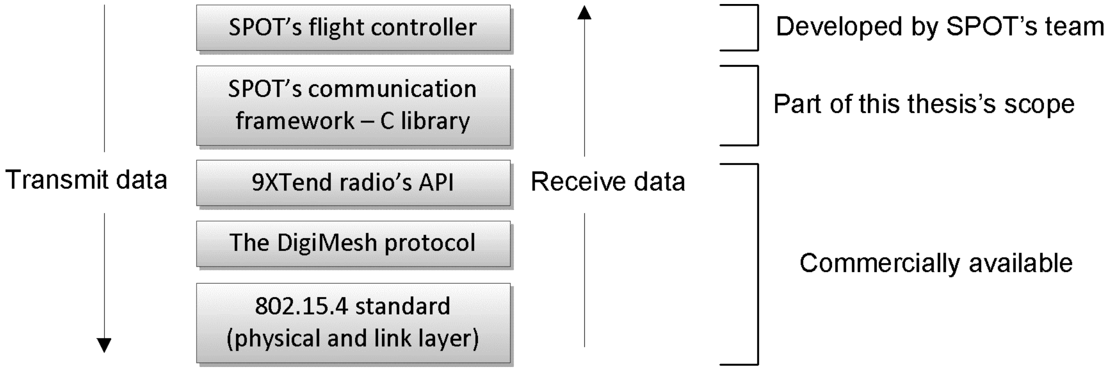


Figure 3.9: Software layers of the radio communication

3.2.1 Radio-packet Based Communication

The 9XTend radio provides an API, which is used for radio configuration and data transmission. The API allows the flight controller of the SPOT to monitor the state of communication by comparing the RSS against the radio's sensitivity, changing the transmit power level, and prioritizing the transmission without the operator's input. In comparison to the API, a transparent communication (section 2.2.2) is more suited for a manual control of the SPOT. Combined with the DigiMesh routing, the flight controller can specify the destination address through API per packet, which leads to a network of autonomously communicating dirigible UAVs. In the API mode the on-board communication with the radio and between the dirigibles is achieved through data packets. As the radio specification describes in [10], the general structure of an API packet is shown on Figure 3.10.

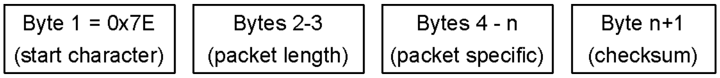


Figure 3.10: Structure of an API packet for 9XTend radio

In the structure shown in Figure 3.10, the start character, the packet length, and the checksum bytes are present in all types of packets. The packet specific bytes change depending on the type of message sent/received from the radio. The following types of messages are used by the framework: radio configuration commands, transmit SPOT's data, acknowledgement for a configuration change, status of the transmitted packets, and the received data. The configuration commands can be sent either to a locally or remotely connected radio. In the case of a remote configuration the packets include network address of the remote radio. The packet structure used for the radio's configuration is shown on Figure 3.11. The main application of the remote configuration is reducing the self-interference in the mesh. The transmitting node can lower the transmit power of the adjacent nodes not participating in the routing. This attempts to address the issue of dynamic power adjustment for each packet, which was described in section 2.6.1 and in [12].

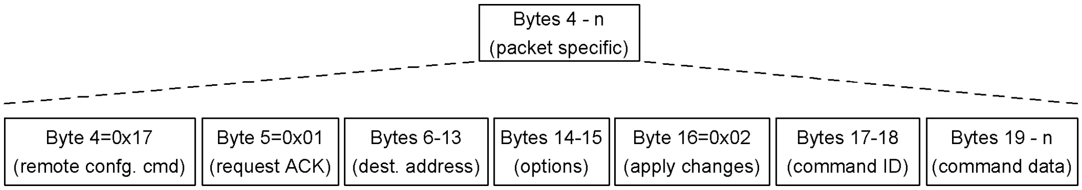


Figure 3.11: Structure of a command packet

The packet structure for data transmission is similar to the configuration command packets up to the first 15 bytes (Figure 3.11). After that the structure shown on Figure 3.12 is used. Each transmit packet can carry a maximum of 256 bytes of payload data. This is more than 128 bytes of payload data specified by the 802.15.4 standard [65]. Therefore, the data files exceeding this size are broken down into 256 byte portions. In our C API we design how the payload portion of the packet is structured to transmit different types of data and allow a receiver to determine the type automatically. The API uses the first byte of the payload portion to label different data types. Since this mesh is a multi-hop network, the API provides a control over number of hops for each packet sent. For SPOT's application the number of hops is not limited in order to achieve the longest range. The networking options allow disabling transmission retries, enabling packet encryption, or using extended transmission timeouts.

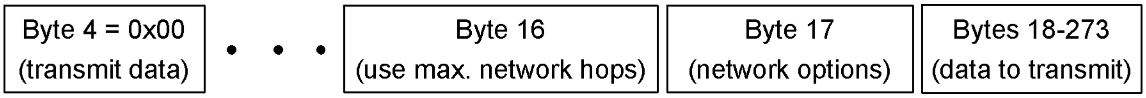


Figure 3.12: Structure of a data transmit packet

The acknowledgement messages are received from the locally connected radio for different types of packets sent to the radio. When either a configuration command or data is sent, the radio notifies the connected on-board computer of the success or failure of the action. The structure of the acknowledgement packets is the same as shown on Figure 3.11, except that the destination address is changed to the responder's address and one extra byte indicating success or failure is added. When a destination radio receives data, it is presented to the framework in the packet structure shown on Figure 3.13. This packet also provides an RSS value in dBm, which is used for detecting weak links and notifying the GCS about their location by including the GPS data in the packet payload. However, if the packet has traversed multiple hops, the reported RSS value is only that of the last hop. The information about the last hop is still important because it could be the only connection to the destination node.

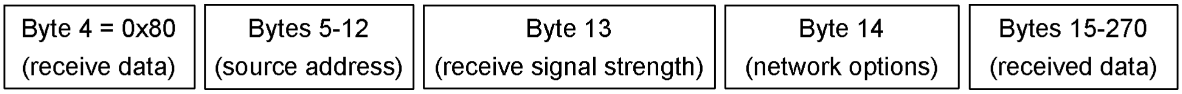


Figure 3.13: Structure of a received data packet

The next section discusses how the framework builds all these different packets and achieves data transmission between the dirigible UAV nodes using the DigiMesh protocol.

3.2.2 Proposed and Implemented API in C

In order to setup a mesh network using 9XTend radios an API in C was developed to facilitate the communication between the flight controller of the SPOT and the 9XTend radio. The API was developed for the Linux operating system, therefore, it uses the C POSIX library for accessing the serial ports of the computer, handling data files, and the system's time functions for the testing purposes. The implemented code provides the following functions to the flight control program:

- Configuring the settings of the USB port connected to the radio
- Receiving flight control commands from GCS
- Transmitting/receiving files (still images, log files, pre-recorded video)
- Breaking large files for transmission and rebuilding them at the receiver end
- Choosing a central node (GCS) in the network while the fleet of UAVs is airborne

The designed and developed API can easily be extended further based on the possible future new requirements of the SPOT developers. Currently, the implemented functions achieve the basic reliable and secure communication links in the mesh. Further details about each function and its development is discussed next.

3.2.3 Serial Interface Configuration

Before the flight control program can communicate with the radio, the serial interface should be configured such that it achieves the maximum data throughput between the on-board computer and the radio. The 9XTend radio connects to the computer through one of the available USB or RS-232 ports. On the computer, the Linux operating system detects the connection on the COM port. There is no need for installing any device drivers because the Linux system already contains them (many USB devices use the same converter circuits). Using the <termios.h> file of the C POSIX library, access is gained to the virtual COM port and its configuration in Linux (this will work for any Linux distribution). Figure 3.14 shows how the API configures the USB port settings.

Once the port is connected the input and output data streams of the COM port can be processed either using canonical or non-canonical mode [58]. The serial interface baud rates supported by the radio range from 9600 to 230400 baud per second but the over-the-air rate is fixed to 115200 baud/s. Therefore, the serial interface rate is set to 115200 baud/s. For the over-the-air transmission the radio uses binary-FSK modulation, which transmits one bit per one baud, hence the effective data transmission rate in the framework is 115200 bps. If the serial interface rate is not equal to the over-the-air rate, a data flow control must be implemented over the serial interface to avoid loss of data. The 9XTend radio has an internal 2.1 kilobyte data buffer, which overflows if either of the transmission rates is not controlled. The flow control would monitor additional signal on the computer connection to stop the transmission when the radio buffer gets

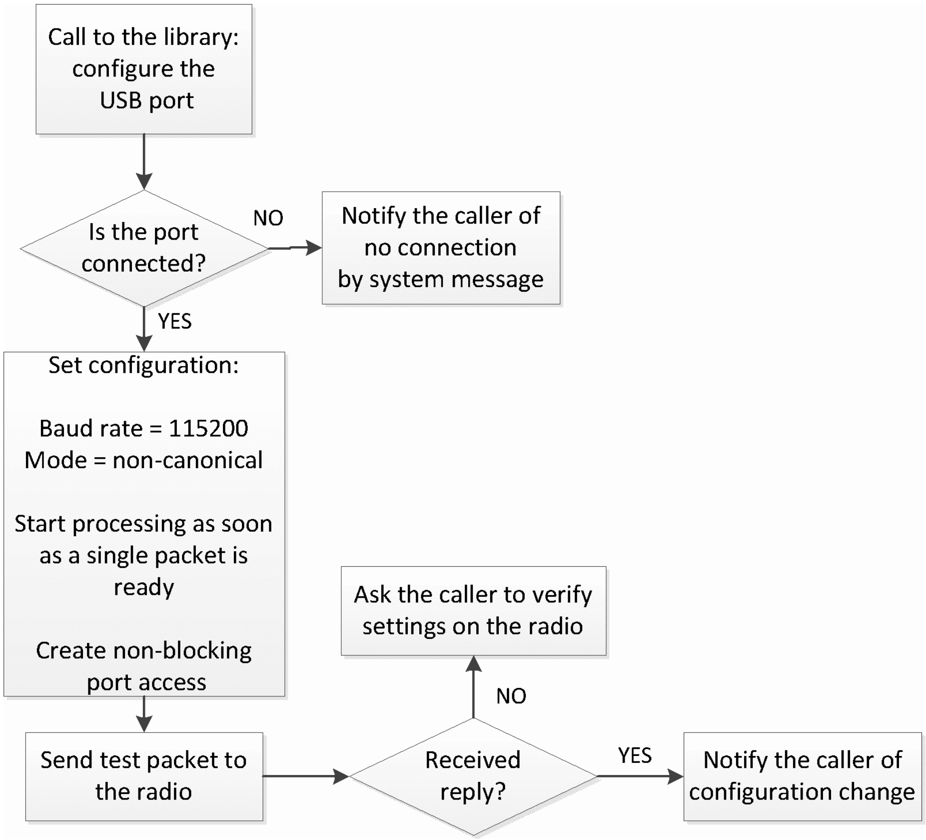


Figure 3.14: Algorithm for USB port configuration for 9XTend radio

full and resume it afterwards. This would complicate the implemented code and for this reason other baud rates are not used in the framework. Rather than processing a single byte at a time, the API works with variable size arrays that represent packets used for communication with the 9XTend radios. This helps to minimize the delay for processing each packet.

The final step in the configuration makes sure that the API does not block the run of other programs on the SPOT. By default, if a flight controller system requests the API to check for incoming data and there is no data available, the API freezes until some data becomes available. This behaviour could block the rest of the flight control program from continuing its work. By setting the port access to non-blocking the API checks if there is at least one packet available from the radio. If it is available, the API reads the entire packet and sends it for further processing. If there is no data available, the API continues with other requests from the flight controller.

3.2.4 Output Data Processing

The API developed in C uses different functions to process the different types of the messages received from the flight controller. At this stage of the SPOT project the focus is on transmitting the flight control commands and telemetry data. When the flight control program makes a call to the API to send commands, it provides the API with a destination address, the command in the form of an array of characters, and the size of the command in bytes to the API. The API then goes through the process depicted on Figure 3.15 to transmit the command.

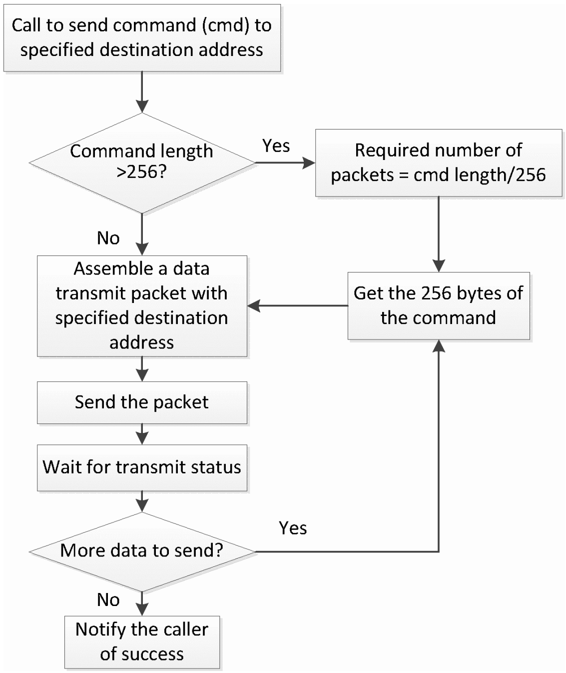


Figure 3.15: Algorithm for processing data for transmit request

Currently the maximum payload of 256 bytes per packet is enough to send all the sensor data from a SPOT to any other node using one packet. The possibility of having longer commands was implemented to accommodate future addition of sensors. When sending images, the API uses a function with similar processing steps as shown on Figure 3.15, except the data is taken from an external file and the payload per packet is changed to 255 bytes. One byte is used by the API to indicate to the receiver the order of the packets. It is important to keep track of the order for the correct assembly of the image file at the receiving node.

For each transmit packet sent to the radio, the radio responds with a transmit status message. This message indicates whether the transmission was successful or not. When the source node sends data to the destination, it waits for a response from the destination. The destination radio forwards the data to the API on the computer and sends back an acknowledgement to the source radio. Once the source radio receives it, a status message is sent to the source node's API. To better explain the transmission of data the interaction diagram on Figure 3.16 provides a summary of where acknowledgements are generated and how they are received through the mesh.

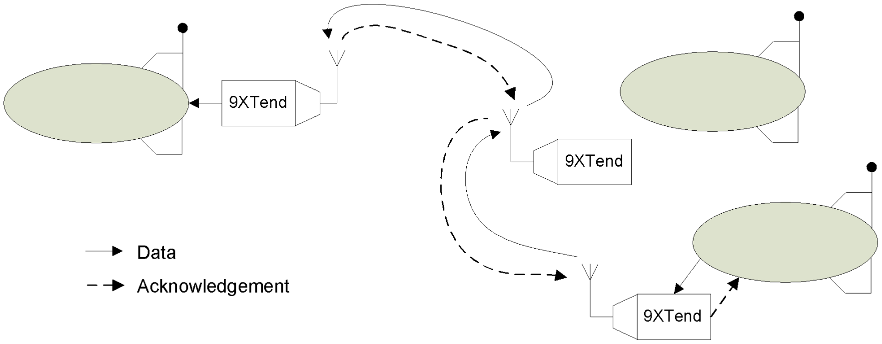


Figure 3.16: Data transmission using DigiMesh protocol

3.2.5 Input Data Processing

The processing of the input data to the API from the radio, is implemented differently compared to the outgoing data. In case of sending data to the radio, every function in the API can generate and send data packets but there is only one way for the data to enter the API. Inside the API the decision is made on the type of the received data and then the appropriate API flag is set. The functions waiting for the acknowledgements or other data always check the flags instead of accessing directly the serial port. The flags in the API are implemented for two reasons: to eliminate simultaneous access to the port by parallel requests, and to prevent a function from reading data that does not belong to it. To illustrate this concept consider the following scenario. The API has sent a new configuration setting for transmit power and is waiting for a confirmation. Meanwhile, the radio receives a data packet from the network and forwards it to the API. If the function waiting for confirmation reads the data packet, the data will be discarded, resulting in a loss of information. To manage the data correctly the API has a single function that checks for incoming data and follows the steps shown on Figure 3.17 for processing it.

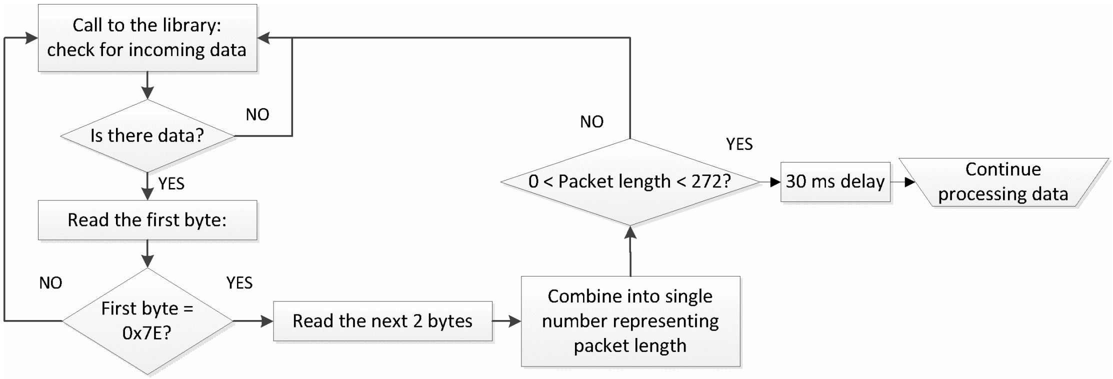


Figure 3.17: Algorithm for incoming data processing

When there is data available from the radio, the API checks if it starts with a valid packet (byte 1 on Figure 3.10) and then checks the packet's length. If both values are correct, the packet is read and processed. Notice the 30 millisecond delay between reading the packet's length and the rest of the packet. This is the amount of time it takes for the Linux OS to forward the longest packet to the API. This value was determined empirically by sending packets with maximum payloads and changing the delay until the minimum value was achieved to guarantee reliable data delivery. Decreasing this value will result in unreliable communication by reading unknown data. However, this delay does not affect over-the-air data rate, as the next chapter on performance evaluation shows.

Once the packet is in the temporary buffer, the API analyses its content. Figure 3.18 shows how the API processes each received packet's content. There are three types of messages which can be received: image data, flight commands, and acknowledgements. The images (compressed photos) or pre-recorded videos are broken into several packets, which are sent and received in the correct order. Since the receiver does not know if the data has been received earlier from the current source node or not, it checks for the first byte in the payload part of the packet. If the first byte is `0x11` indicating a new file, the API takes source node's network address and uses it to name a file where the consecutive data from the current source node will be stored. For identification purposes we are using the IEEE 64-bit unique addresses which are assigned to the 9XTend radios by the manufacturer. By using the senders network address as the file name, data can be properly received from multiple sources. Also the operator may need to retrieve the GPS data where the photo was taken by sending a request to the saved network address.

The `0x50` byte would indicate that file already exists for this sender. The `0x99` would indicate that this is the last piece of image data from this sender. The API would read it and release the file on the local computer so that the operator at the GCS can view it. The flight commands could be either a control message for the on-board electronic devices or the data arriving from the sensors. In this case, the payload content of the packet is saved in a separate buffer and the flight controller is notified of the awaiting data in the buffer. The acknowledgements could be either for successfully sent data or for a successfully applied new configuration on the requested radio. These processing functions represent the main communication in the mesh.

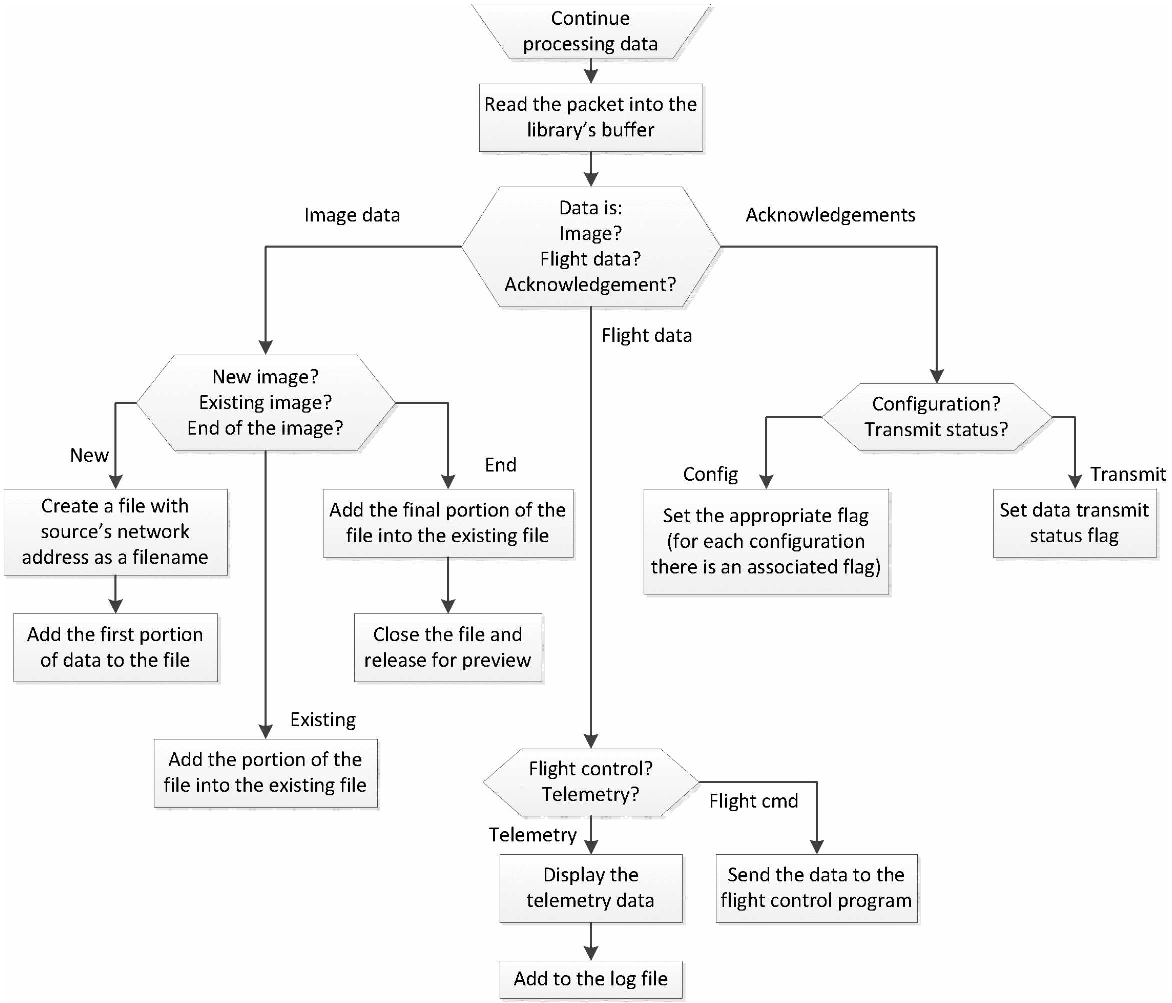


Figure 3.18: Algorithm for processing input data (part 2)

3.3 Advanced Features - Mobile Central Node Function

The mobile central node role refers to the ability of the mesh network to designate a central node at any point in time. For example if for some reason the GCS stops responding, one of the dirigible UAVs can take its role. One of the key functionalities of a central node is collecting GPS data from each dirigible for the purpose of fleet management. By collecting and storing the GPS data from the dirigibles a geographical coverage area can be determined by plotting the acquired data on a map. In most ad hoc networks the roles are preassigned to the wireless nodes. A recent research implementing a new token-based ad hoc network for multiple UAV communication assigns a “backbone UAV” role, similar to the central node, to one of the UAVs [29]. From their test results it appears that the backbone assignment is done before the testing, which implies that if the backbone is disconnected during the flight the GCS will loose track of the fleet. In our mesh the central node can be automatically reassigned to another dirigible while the fleet is airborne.

The central node role is assigned when a dirigible sends a special broadcast command on the network called “aggregator support”. Every dirigible that receives this command builds a route back to the aggregator and stores the aggregator’s network address locally. After the aggregator has been assigned in the network, any dirigible can send data to it using a regular data transmit packet with the aggregator’s address.

3.4 Route Tracing

One of the useful capabilities of our implemented framework is that all the transmission routes can be traced. As more radio nodes are added to the communication framework it will become important for the transmitter to trace the route. An important application of route tracing is avoiding data collision and long delays at the radio nodes which are already busy. For example, a single broadcasting node can slow down the transmission of images between other nodes by 4-6 times. A similar case is when an intermediate node is used simultaneously for two different transmissions. This scenario is illustrated on Figure 3.19. In both cases the transmission is delayed because some of the radio nodes have to process multiple streams of serial data.

In Figure 3.19 the node marked with darker shade causes a delay in both data streams

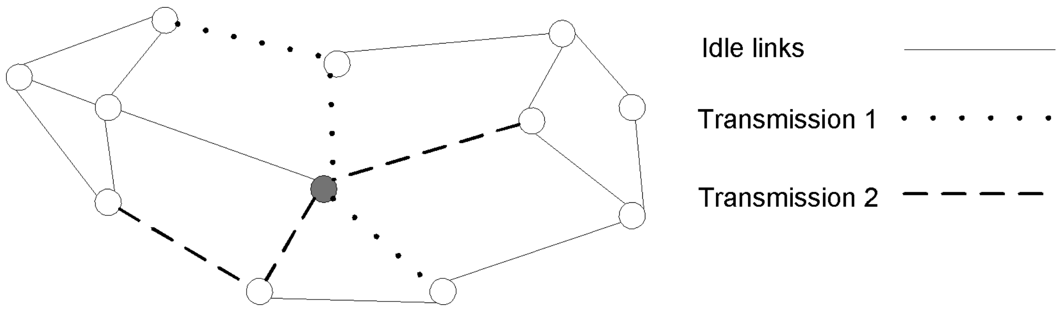


Figure 3.19: Two data streams using same intermediate node

because it is processing different routes. Assuming that the route marked with dashes is more important we can temporarily isolate it (create a cluster) from the general network. First, we trace each route to identify intermediate nodes common to both routes. Then we assign different network IDs to these nodes. This will cause all the other traffic to be rerouted around the priority route, thus avoiding the delay issues, as illustrated on Figure 3.20. After the transmission the network IDs are changed back to restore the connections to the overall network.

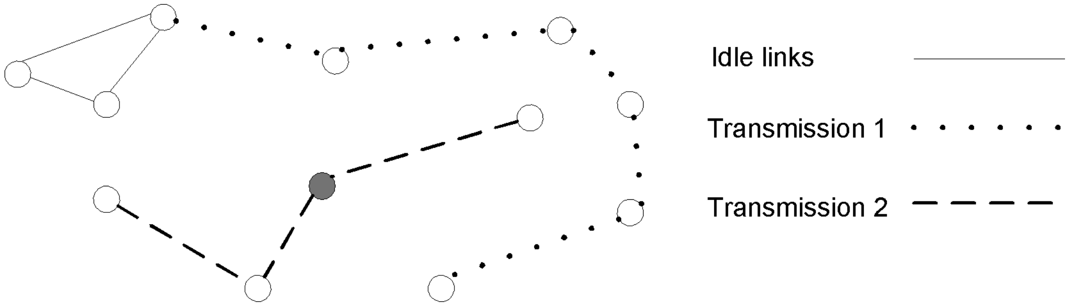


Figure 3.20: Two data streams separate by different network IDs

The route tracing is implemented in the following way. When the flight controller requests a route trace to a given destination address, the API assembles a special packet which is tracked by DigiMesh. Every time the data packet hops, the summary of this hop is sent back to the source node in a delivery status message. If the data packet makes two hops, the API receives two status messages. Each status packet tells the source which node in the mesh last received the data and where it was forwarded. After receiving each status message, the API extracts the address of the intermediate node and stores it in an n by 8 byte array (64-bit address), where n represents the number of hops. The ascending order represents the sequence of intermediate nodes. This feature will become more useful as more radio nodes are added to the network.

3.5 Chapter Summary

The proposed communication framework's hardware and software configuration have been presented in this chapter. It explains how the framework's radios are integrated into the SPOT dirigible UAV's on-board electronics and connects to the computer. It was shown that the existing mesh network designs using 802.11 standards are not capable of achieving communication ranges longer than 3 kilometres and require custom routing implementation before data can be sent. Using DigiMesh and its associated long-range 900Mhz 9XTend radios with half-dipole antennas, the framework achieves a scalable mesh network of 5 to 10 kilometres communication range with a LOS and maximum transmit power of 1 W. The section on hardware setup briefly mentions the antenna placement on the dirigible but does not provide the exact final setup because this will be performed once the first dirigible is ready to fly. By using custom serial interface converters instead of the ones from the manufacturer, the cost of the framework can be kept to a minimum.

The software side of the framework takes care of the radio control, data and fleet management. Through radio-packet based communication and using proprietary packet structures, the developed API in C establishes reliable links among dirigible UAVs. The packet based communication takes up some of the processing time because large files must be broken down into smaller pieces that fit into packets and then each packet must be acknowledged. But the packets allow autonomous reliable communication among multiple dirigibles and the transmit power can be changed on a per packet basis. The developed packet-based mesh can be used as a testbed for multi-UAV control algorithms and new ad hoc protocols being developed by other researchers.

Chapter 4

Performace Evaluation and Results

In order to verify the overall expected performance of the framework several experiments have been carried out. The tests were conducted in indoor and outdoor real world environments. Three 9XTend 900 MHz radios supporting the DigiMesh protocol were used to create a small mesh network for testing. The completed tests were designed to evaluate the following aspects of the framework:

- Power requirements during the receive and the transmit modes
- Individual link and mesh testing
- Self-healing functionality of the mesh
- Reliability of mobile mesh connections
- Transmission of images in the mesh
- Effective throughput measurements for single and two hops
- Data collision and interference handling

The tests are not performed airborne because the dirigible UAV is not ready to fly yet. The performance of these framework functions are captured at the application layer (Figure 3.9). The individual test scenarios are described in the following sections.

4.1 Radio Power Requirements

Depending on the selected transmit power level, the radio requires different supplied power. In the transmit mode the radio uses the amount of power specified in the table 4.1.

Table 4.1: 9XTend radio power requirements for transmitting data [10]

Transmit Power	**1 mW**	**10 mW**	**100 mW**	**500 mW**	**1 W**
Achievable range	0.5-1* km				5-6** km
Supply Voltage	2.8 - 5.5 V			3 - 5.5 V	4.75 - 5.5 V
Supply Current at 5V	130 mA	160 mA	270 mA	500 mA	680 mA
Supply Current at 3.3V	125 mA	150 mA	260 mA	600 mA	N/A

* has been verified through tests

** has been verified to provide the highest RSSI at 1.1 km

When receiving data, the radio draws 88.9 mA from the power supply. The receive mode is also a default mode for the radio, meaning that when there is no transmission 88.9 mA will be used by the radio. When the radio detects the incoming data from the on-board computer, it automatically switches to the transmit mode. The supply for the transmit power levels of 1-100 mW was provided from the USB port of a laptop. For 500 mW - 1 W transmit power, external power sources such as a DC wall adapter (Figure 3.6) or a 6-cell lithium-polymer battery was used.

4.2 General Test Procedure

Our experimental testbed configuration consists of 3 9XTend radios, running firmware with the DigiMesh routing. Each radio was connected to a laptop, which runs a Linux OS. The actual on-board computer of SPOT will eventually be running TinyOS version of Linux. It does not matter which Linux distribution is running because one can easily install the same system libraries for accessing the serial ports on each distribution. The photos on Figure 4.1 show the three radio nodes during tests. The radio nodes on Figure 4.1(a) and (c) were used as a transmitter/receiver while the node on 4.1(b) was used as an intermediate router. The 9XTend radios on (a) and (c) were used with the development boards which allowed fast software development and testing.

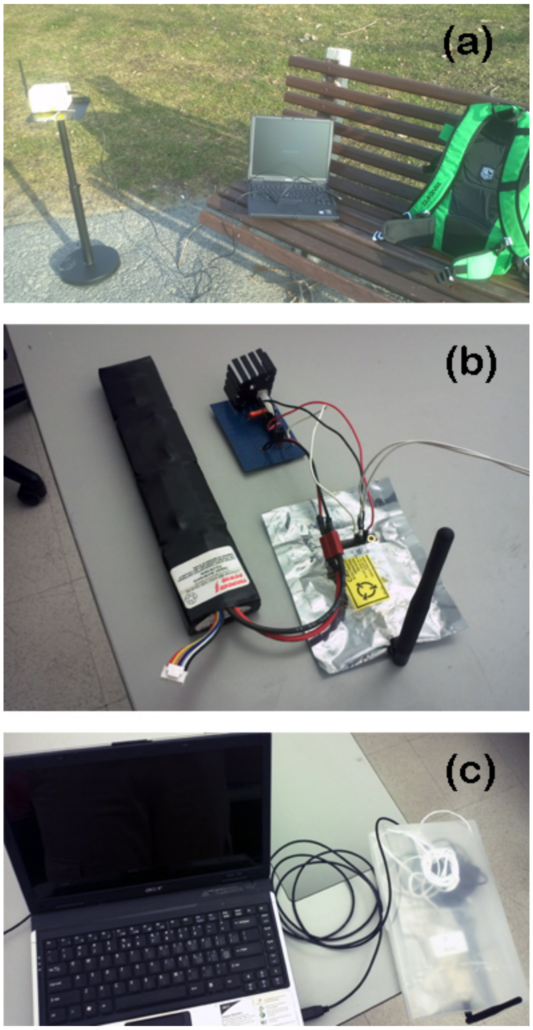


Figure 4.1: Setup of three 9XTend radio nodes

In order to test the image and text-based command transmission, additional functions such as timing and route tracing have been added to measure the framework's overall performance. As it was mentioned in Chapter 3, the developed API is not a standalone application. Therefore, a small program was written to perform the same function calls to the API that the SPOT's flight controller would. During the tests each laptop executes this program from its Linux terminal for sending data and checking the serial ports for incoming data by making calls to the API. For the test images, we took few photos of the University of Ottawa campus and the Rideau Canal from an elevated viewpoint. These photos were prestored in JPEG format and were about 1-4 Megabyte each. The photos were compressed by reducing the resolution until their size was less than 100 kilobytes. It is possible that in certain scenarios the operators may need to acquire a higher resolution, uncompressed JPEG photos. For these tests we used the uncompressed. As the following discussion will show, the format and the size of the stored images do not affect the reliability of the framework. This way our system can transmit any file type without additional image processing libraries.

4.3 Operational Range of The Radio

In order to validate that the communication range can be extended by increasing the transmit power while maintaining a constant throughput, a basic test scenario was performed by transmitting data at different power levels and capturing the RSS at the receiver. The captured RSS values are shown in Figure 4.2. Increasing the transmitter power to 1 W from 1 mW results in 30 dBm RSS increase, as expected. The conclusion is that as long as the RSS is above a threshold (e.g. threshold in our test is -95 dBm in Figure 4.2), the distance between the receiver and the transmitter can be safely increased. By performing this kind of analysis of the RSS, while the fleet is airborne, the distance in each link of the DigiMesh can be maximized.

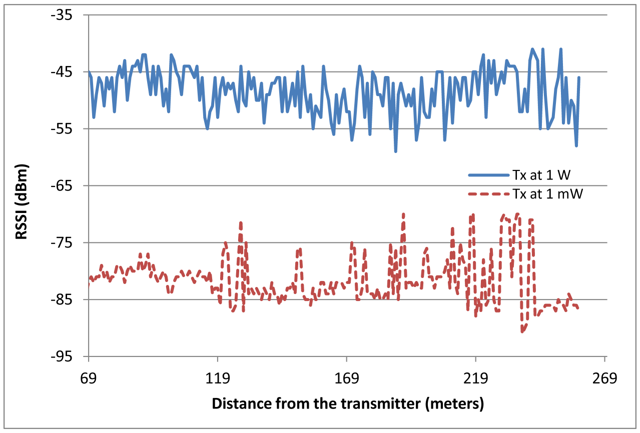


Figure 4.2: Operating Range of the 9XTend radio from received power perspective

4.4 Effective Data Throughput Measurements

Our goal is to evaluate the effective throughout for a single and two-hop transmission. First, we measure the throughput when there is no movement of the nodes. Then we verify the same results with mobile scenarios. The effective throughput is calculated as follows:

$$\text{Effective throughput} = \frac{\text{Total number of bits (or bytes)}}{\text{Time it takes to send and receive a file (seconds)}} \tag{4.1}$$

We are interested in the real (wall-clock) time it takes to transmit a given amount of data. The accuracy of the measurements and calculations are ± 1 second. For our application this is an acceptable accuracy because the environment around the dirigible does not change significantly within a second. All of the on-board sensors are sampled every second. The algorithm on Figure 4.3 shows where we added the timing function to the API. The captured transmission duration includes the time it takes to acknowledge each packet. Without the acknowledgements the transmission becomes unreliable and as explained in section 4.5.1, the transmitted photos may get corrupted at the receiver end.

4.4.1 One-hop Throughput

To prepare the photos for a transmission, we take each photo shown on Figure 4.4 and compress it into three different size. Each of the compressed photos are sent 3-5 times in order to verify that the duration of the transmission does not change. The duration is then recorded for a single hop and the effective throughput is calculated in Kbps using equation 4.1. During the calculation of the throughput we take into account the packet header and the checksum bits, which constitute an additional 128 bits. The maximum payload per packet is 255 bytes, which is twice more than specified by 802.15.4 (maximum of 127 bytes).

First, we measure the effective throughput of a one hop transmission inside the laboratory and then perform the same test using a daisy chain connection with an intermediate radio node to extend the range of the communication. Our goal is to use the one-hop throughput as a reference for the two-hop transmission later, in order to find by how much the effective end-to-end throughput decreases in a multi-hop network using DigiMesh. If we assume that a one hop transmission takes y seconds, we expect that the worst case in a two hop transmission would take 2×y seconds. Table 4.2 summarizes our test results for a one-hop transmission.

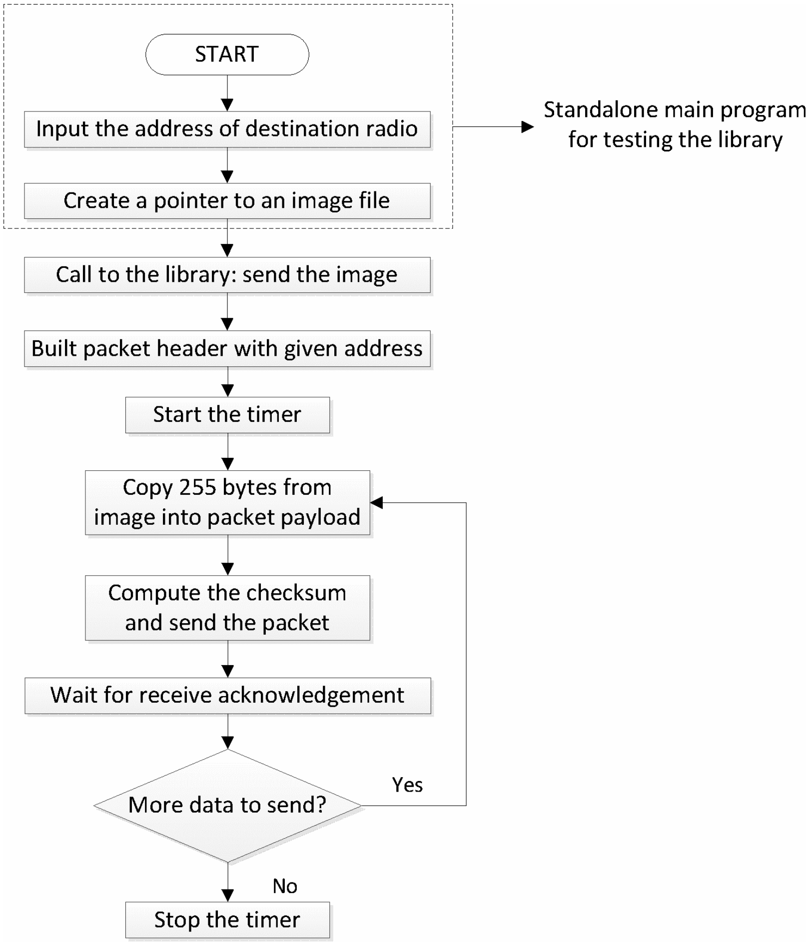


Figure 4.3: Program structure for timing image transmission

From Table 4.2, the image size indicates the size of an image file prestored on the computer. The total number of packets is $\frac{Image\ size\ (bytes)}{255}$, rounded to the next integer. The header indicates the total number of bytes used to transmit an image, calculated as $Total\ packets\ \times 16\ header\ bytes\ per\ packet$. The total amount of data transmitted is the sum of the header, image size, and a checksum. The throughput is then calculated as $\frac{Sent\ data}{Tx\ time}$. The difference of a second in Tx time column of Table 4.2 is due to recording the time on different laptops, which is related to processing data through a serial port and Linux OS. On average our communication framework achieves a throughput of 31017-33935 bps. Comparing this result to the throughput of 40 Kbps, specified by the 802.15.4 standard [65], we conclude that the two values are close. The 802.15.4 standard specifies the throughput at the physical layer, while we were measuring the throughout at the application layer because this is the amount of time the framework takes to transmit

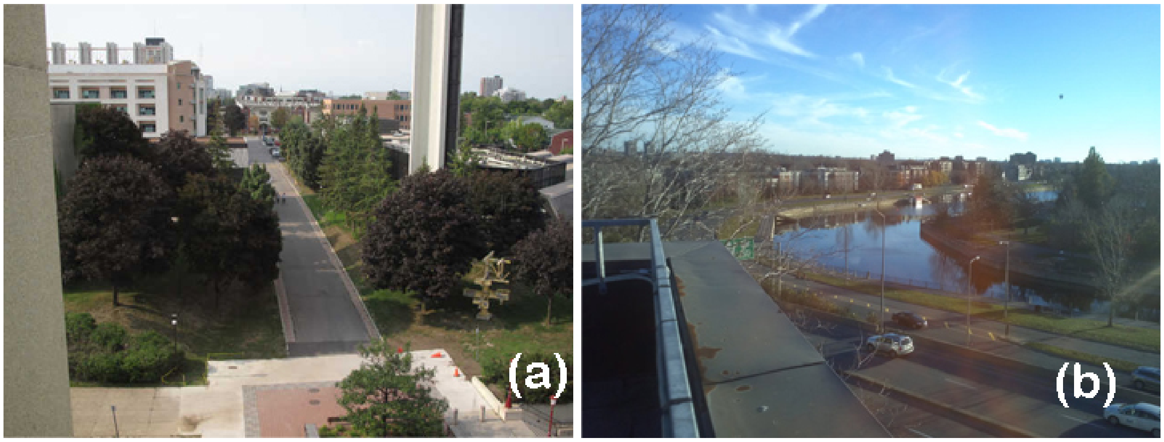


Figure 4.4: Photos used for the performance evaluation

Table 4.2: One-hop effective throughput (with ACKs)

Image Size (bytes)	**Total packets**	**Header (bytes)**	**Sent data (bytes)**	**Tx time (seconds)**	**Throughput (bps)**
22953	91	1456	24410	6-7	27897 - 32546
45397	179	2864	48262	11-12	32175 - 35100
112482	442	7072	119555	28-29	32981 - 34159

an image. The DigiMesh routing and the data handling by the Linux OS contribute additional delays, which can explain the difference between the achieved and the specified throughputs.

4.4.2 Two-hop Throughput

In order to measure the effective throughput for a two hop transmission, we needed to make sure that all the packets are routed through an intermediate node. For this, we setup the testbed on the floor outside the laboratory. On this floor the transmitter and the receiver nodes are separated from each other sufficiently enough that they can not communicate (with the minimum transmit power of 1 mW). The setup is illustrated by the diagram on Figure 4.5.

All the nodes were stationary during throughput measurements. We take the point A, on Figure 4.5, as a reference for distances and there is no LOS because the radio at point

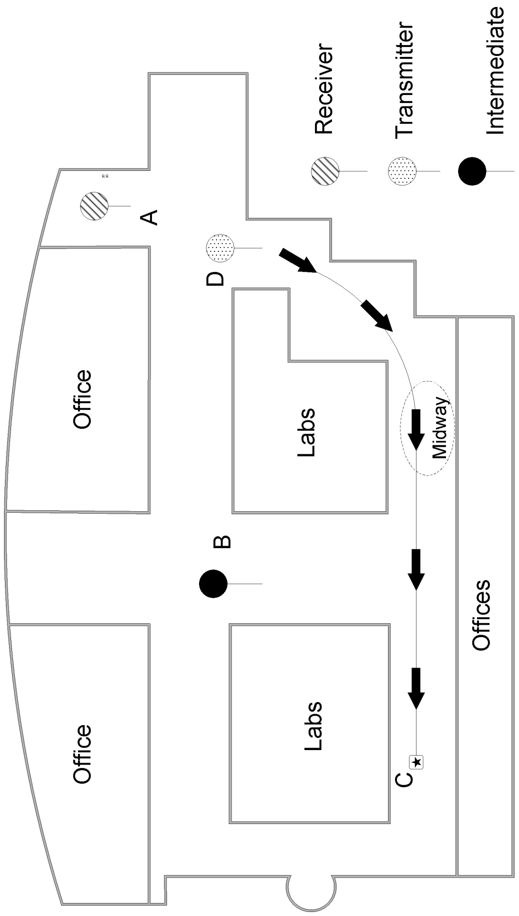


Figure 4.5: Testbed setup on the floor

A is inside an open-door room. The LOS distance from point A to the intermediate node B is 45 meters. The LOS distance to point C from point A is 73 meters. First we verified that the radios at point A and C can not communicate by keeping the intermediate radio powered off. Then we power it on and record the two-hop transmission time. The results are summarized in Table 4.3.

Comparing results from Table 4.2 with Table 4.3, we observe that the transmission time increases, resulting in end-to-end throughput decrease by 41-42%, so it is not the expected full 50%. The expected decrease in throughput is 50% because the same transmission is repeated twice. The achieved decrease in throughput is less than that because each hop exhibits a better performance than the equivalent one hop over the same distance. The final point is that the transmitter power level does not affect the throughput. We performed the one-hop and two-hop tests with different power levels and found out that

Table 4.3: Two-hop effective throughput (with ACKs)

Image Size (bytes)	**Total packets**	**Header (bytes)**	**Sent data (bytes)**	**Tx time (seconds)**	**Throughput (bps)**
22953	91	1456	24410	10	19528
45397	179	2864	48262	15-17	22711 - 25740
112482	442	7072	119555	39-41	23328 - 24524

the throughput does not change as the transmit power levels change. As long as all the packets are received with RSS above the radio's sensitivity of -100 dBm, the throughput remains constant. This is expected because the 9XTend radio does not implement an adaptive modulation scheme, which would provide higher throughput at higher power levels and vice versa. The movement of the nodes also did not affect the throughput while the packets were received with an RSS above -100 dBm. This means that as long as the SPOTs are moving within the communication range the throughput will not change.

4.4.3 Transmission Delays

When working with data converters between UART, USB, and RS-232 the operating system can be a source of an additional delay. In Windows or Linux there is a delay called "serial port latency timer". Not all device drivers use this timer but in our tests we determined that changing its value made a difference on the effective throughput as the image sizes were increased. The purpose of this timer is to buffer the data coming through a serial port before forwarding it to the application layer. By default this timer is set to 16 milliseconds. For a real-time processing application this timer can not be removed but it can be set to a minimum of 1 millisecond. In our application, the converters between UART and USB integrated into the development boards (Figure 3.6) are affected by this timer. The one-hop and two-hop throughput measurements were done with the timer set to 1 millisecond, which does not effect the transmission time significantly.

In addition to the latency timer, another delay occurs in transmission when new routes are built at each radio node. In the DigiMesh network the routes are built when the transmitter has data to send, similar to the AODV routing. The building of each point-to-point route, when the radios are powered up, can take 1-3 seconds based on our test results. However, once the route has been built it is stored in the routing table of each radio and the consequent transmissions take place without the initial route finding delay. Upon the power cycle of the 9XTend radio, the routes are deleted. In order to find out this delay we power cycle all three radios, then record the transmission time and find the additional delay by comparison to one-hop and two-hop results from Tables 4.2 and 4.3. We found that the additional delay from route finding on each radio is about 1-3 seconds.

4.5 Reliability of The Mesh

The communication framework is being designed to operate on the mobile SPOT in an uncontrolled environment. Therefore, it is important to maintain a reliable connection between the radio nodes in the network for achieving the measured throughputs and receiving compressed photos without errors. The reliability of the wireless connections can be improved using software techniques and/or diversity techniques in the space domain. The latter is a future goal in the ongoing development and its design is discussed in the section 5.2. In this section we focus on the implemented software techniques with acknowledgements and checksum. The checksum (Figure 3.10) is computed for each packet as: `0xFF`−(sum of all bytes excluding the start character and the packet length). The checksum is verified at the DigiMesh protocol layer (Figure 3.9) and if the verification fails the packet is discarded and no acknowledgement is generated. Section 3.2.4 described the structure of the acknowledgement packets. In the next section we will show the benefit of using acknowledgements, at the cost of potentially lower effective throughput, in order to control the serial flow of data, especially when transferring photos in the network. Acknowledgements become the only reliable way for controlling the SPOT in a no-LOS environment.

4.5.1 Transmission of Image Data

When sending photos, the designed API performs fragmentation, packetization, and reassembly of images. We are not using any additional image processing library and only work with the binary data. The transfer of any files in the binary data format, including a prerecorded videos, is easier than using image processing libraries. We look at the reliability of image transmission with and without acknowledgements, transmission of an image while one of the nodes is in motion, and transmission of different formats such as JPEG-2000 and BMP.

Our first design and development of the image fragmentation and packetization used only delays. The framework splits the JPEG encoded photos into packets and then assembles them back at the receiver. After sending each packet to the radio from the computer, the API pauses for 10-30 milliseconds. This delay was used to give the radio enough time for processing and sending the packet to the receiver. We assumed that the packets were arriving at the receiver in the same order as they were being sent by the transmitter. This assumption simplifies the reassembly of the photos. Examples of the received images are shown on Figure 4.6. As the results show this was not a reliable way of transmitting large files. Out of 10 transmissions of each photo, 5 to 7 of the received photos would be corrupted even if the link reliability is high.

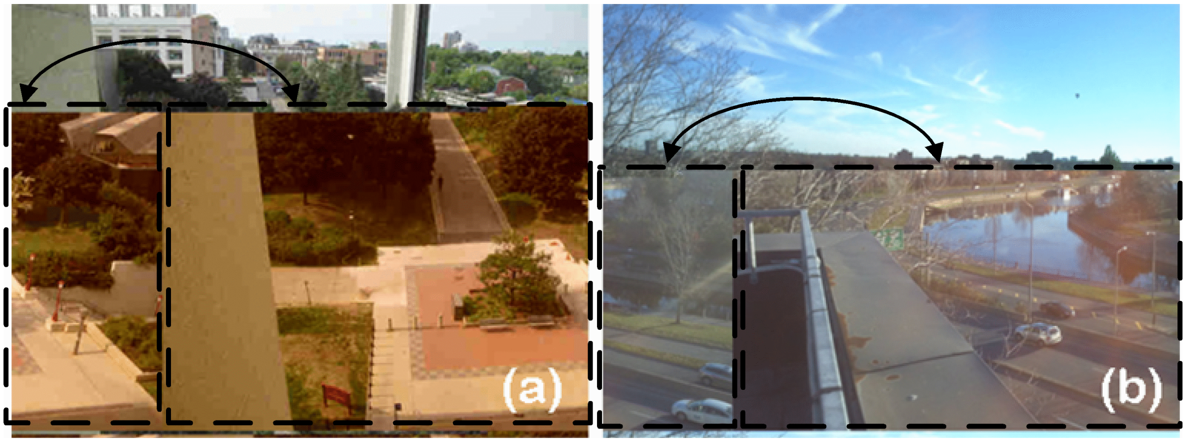


Figure 4.6: Received corrupted images

The reason behind this was that some of the packets were not received in the right order. Consequently the reassembly of the photos was erroneous. The image (a) on Figure 4.6 was received over a link with a reliability of 99.04%. The reliability is computed as: $\frac{Size\ of\ received\ image}{Size\ of\ sent\ image}$. The image (b) was received over a link with a reliability of 98.87%. This result shows that JPEG format has no tolerance for error. Looking at the nature of the error it looks that the images were received correctly at the beginning but then some parts were flipped horizontally. Because our design relies on a correct sequence of packets this suggests that some of the packets were received in the wrong order while the 1-2% of total packets were lost (missing intensity values). In a mobile wireless mesh it is possible that some packets can reach the receiver earlier than the others because the routes can change for each packet depending on the link's strength. For the next design revision we made use of the acknowledgement instead of delays to control the flow of data. Now, every time the packet is sent the API waits for the acknowledgement before sending the next one. This way it is guaranteed that all the packets will be received in the correct order before being assembled into the image. This solved the problem with the corrupted images. The next section explains that increasing number of hops also increases the transmission delay and the reliability of the connections because the new routes can the replace broken ones.

4.5.2 Self-healing Links - Best Route Selection

Our goal is building a mesh that can rearrange its topology to maintain connections in a mobile radio network. In this section we perform a test scenario where a an existing link breaks during the transmission of an image and a new route is chosen. It is expected that the dirigibles will be moving within a fleet. The links may get broken while transmitting data. Even though the flight controller can monitor the RSS the SPOT can fly behind large obstacles that may create large fluctuations in the RSS and cause sudden failures of links. In situations like this a mesh network should reroute the data to maintain a reliable link. Hence, the purpose of this experiment is to analyze the capability of the framework to maintain the transmission of the data while repairing a broken link by selecting a new route.

In our test setup, we start the transmission of an image at point D of Figure 4.5 and move the transmitter as indicated by the arrows. As the transmitter gets close to the intermediate node at point B, the link between the transmitter and receiver becomes weak and eventually breaks before the transmitter can reach point C. When the link breaks the packets are routed through an intermediate node. During the transmission the API captures the RSS of each packet and stores it into a file. We then generate the plot shown on Figure 4.7, where each point indicates a single received packet. The graph on the Figure shows that as the transmitter is moved further away from the receiver the RSS drops. Up to about 45 meters one-hop transmission is taking place. After that there is a positive jump in the RSS values, indicating that the transmitter started to send the remaining of the image packets through a closer intermediate node. During the two-hop transmission and rerouting of the remaining packets no loss of data occurred because the images were received successfully. This result shows that combining our API with the radios, the framework maintains the shortest path during a continuous transmission as long as the received RSS is above the threshold of -100 dBm. When the transmitter fails to receive an acknowledgement it starts a new route finding procedure.

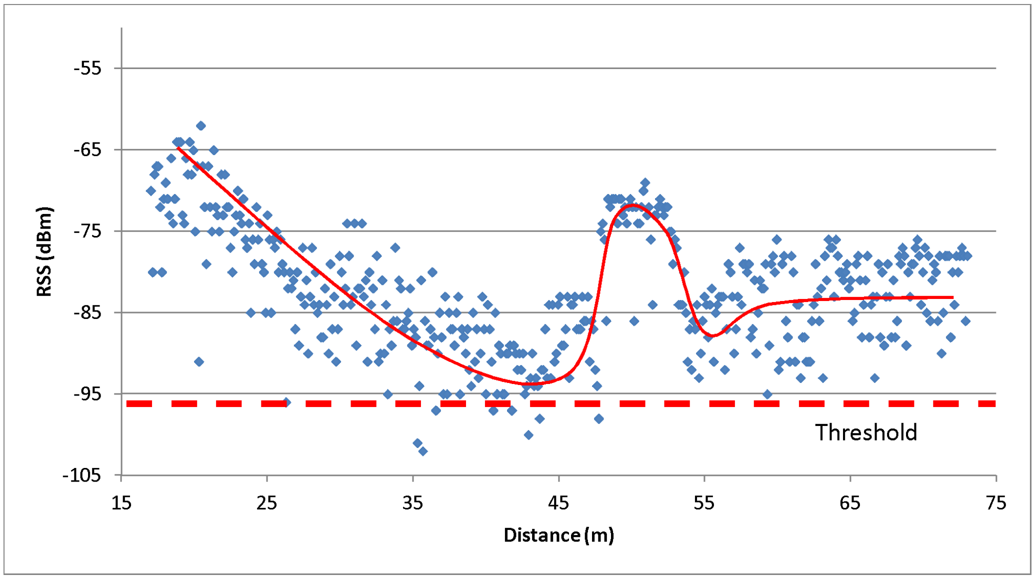


Figure 4.7: Automatic passing from one-hop to two-hop transmission

In order to establish a new route through an intermediate node, the transmitter sends a route request packet through the network as a broadcast message. When the request reaches the receiver, it sends a reply packet back to the transmitter through the same route used by the request packet. In a large network there is a possibility that several replies be received. In that case, the transmitter selects the best route based on the end-to-end link quality. The quality is determined by the DigiMesh protocol using the RSS values along the routes.

4.6 Data Collision Handling

Recalling the issue of self-interference described in section 2.6.1, we are now going to look at the RSS values in the presence of an interfering node. From Figure 4.5, we place the transmitter midway between the receiver and the intermediate node such that all three radios can communicate. We then perform the following steps:

1. We obtain the reference RSS values for the transmission between point A and D. The transmitter sends an image to the receiver while the intermediate node is turned off. The resulting graph is shown on the Figure 4.8. Notice that the weakest RSS is -53 dBm.

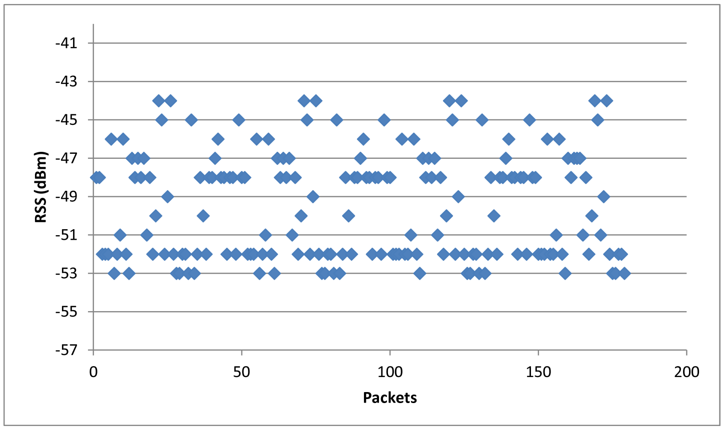


Figure 4.8: RSS vs. distance with no interference

2. After we have obtained these reference values, we start sending broadcast messages from the intermediate node B. This generates some traffic in the mesh and adds self-interference when the other two nodes are transmitting data.

3. We then start the image transmission from the transmitter to the receiver. The receiver captures the RSS values which are later used for comparison with the reference values. The resulting graph is shown on the Figure 4.9.

Notice that in the presence of the interfering node, the weakest RSS is -56 dBm. Compared to the case with no interference on the Figure 4.8, the lowest RSS value decreased by 2-3 dBm. In addition, the transmission time was multiplied by 4-6 because the transmitter and receiver had to process extra data from the broadcast. A possible solution to this situation is to assign different network IDs to the transmitter and the receiver. In the DigiMesh network, only the radio nodes with the same network ID can exchange data (clusters). By default all radios are assigned `3332` network ID. We changed this to `1111` on the transmitter and the receiver node, which resulted in no self-interference from the broadcasting intermediate node. Once the interference was eliminated, we achieved the same one-hop throughput discussed earlier (section 4.4.1) and the same RSS values found earlier in Figure 4.8.

4.7 Long-range Communication

Up to now we have demonstrated that the communication framework uses the DigiMesh routing protocol in order to successfully implement and maintain a transmission in a

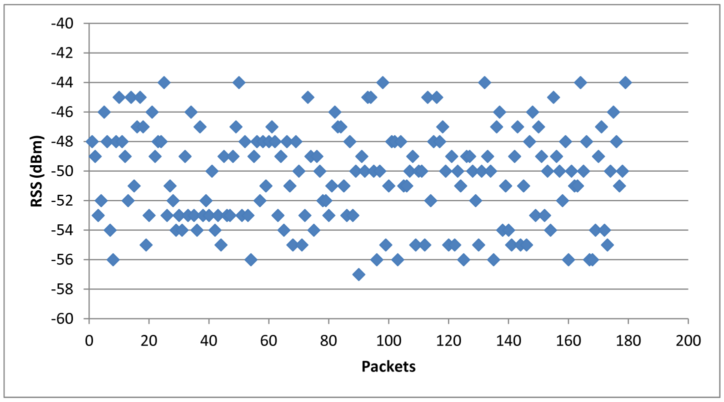


Figure 4.9: RSS vs. distance with interference

network with mobile nodes. This result can also be achieved over long range communication links between the two dirigibles. The purpose of the following experiment is to demonstrate that a single hope within our framework has the capability of achieving a much longer communication distance with less power and a half-wave dipole antennas, than the regular Wi-Fi technology used in the existing sensor networks.

The section over the Ottawa River between the Victoria Island and the Alexandra Bridge was chosen as the test site for our long-range experiment. A detailed diagram demonstrating the test site is shown on Figure 4.10. The photos of our setup are shown on Figure 4.11. These sites were chosen for two reasons: the line-of-site between the two locations and the elevation of 5-10 meters above the river on the bridge.

Figure 4.11: Hardware setups on the test sites

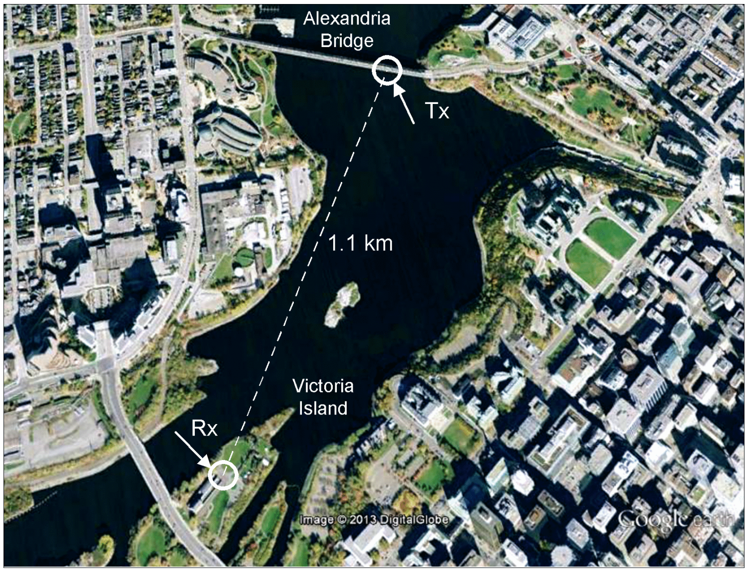


Figure 4.10: Long-range communication test site

The test was performed by setting the transmit power to 1 W on both radios while exchanging images between the two locations. The framework recorded the RSS values of each packet in order to find out the minimum transmit power required over the indicated 1.1 kilometre distance. The resulting plot of the RSS values is shown on the Figure 4.12. From the plot, we note that the weakest RSS is -65 dBm. We have already demonstrated in section 4.1 and Figure 4.2 that changing the transmit power from 1 W to 1 mW results in 30 dB loss in the RSS. Changing the transmit power from 1 W to 1 mW in our long-range communication experiment results in a -30 dB shift of the plot on Figure 4.12. The resulting plot is shown on Figure 4.13.

The weakest RSS shifts to -93 dBm when the transmit power of 1 mW was used. We have also performed the one-hop throughput measurements for different power levels. At 1 mW transmit power the transmission delay was increased by 30% compared to the transmission at 1 W. The transmission delay was higher because at 1 mW some of the data packets were retransmitted due to the RSS values below -100 dB. We increased the transmit power to 10 mW and achieved the same one-hop throughput results shown in Table 4.2. From this experiment the first conclusion we can draw is that operating the radios close to their sensitivity of -100 dBm may increase the transmission delay (depending on outdoor environment), which is resolved by increasing the transmit power or by using a half-wave dipole antenna with higher gain at the receiver. This means

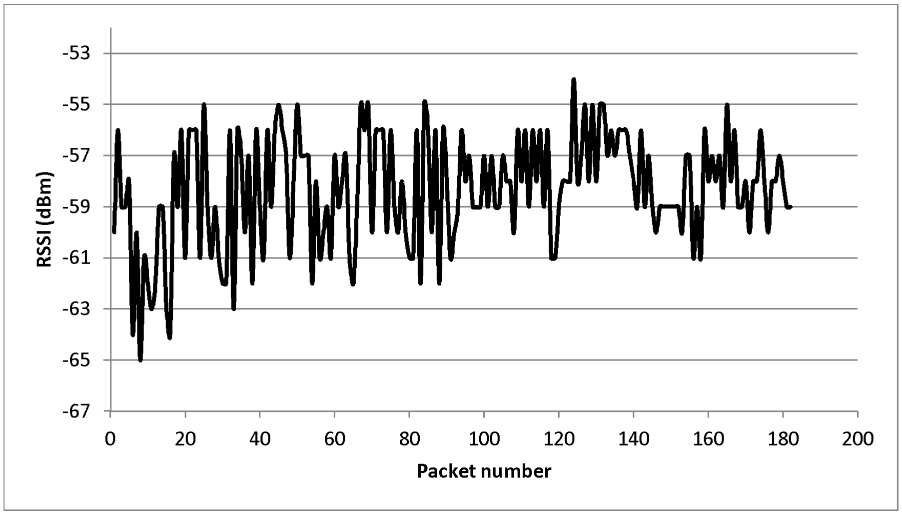


Figure 4.12: RSS values captured over the distance of 1.1 km with Tx at 1 W

that it is more optimal operating the framework such that the distance between any two dirigibles is maximum based on the captured RSS and the acceptable transmission delay. The second conclusion is that using just 1 mW transmit power with a half-wave dipole antenna, the framework works over a distance that is at least 3 times longer than existing sensor networks using Wi-Fi.

4.8 Chapter Summary

The performance evaluation of our communication network has been presented in this chapter. We have explained how we have modified the original hardware setup in order to satisfy power requirements for long range communication. Experimental testbed setups were used to measure one-hop and two-hop data throughout in the mesh. We used larger image sizes than what will be used by the SPOT in order to verify the reliability of our software. Our designed and developed communication API can achieve up to 35 kbps over one-hop using acknowledgements and an over-the-air transmission rate of 115200 bps. Adding the second hop increases the transmission delay by 41-42%. We have successfully verified that with our designed and developed API, images are received without errors when the transmitter detects a connection loss and selects a new route.

Although, the network is designed to transmit the photos and the sensory data, the latter was not tested in the network because the sensory platform of the SPOT is currently

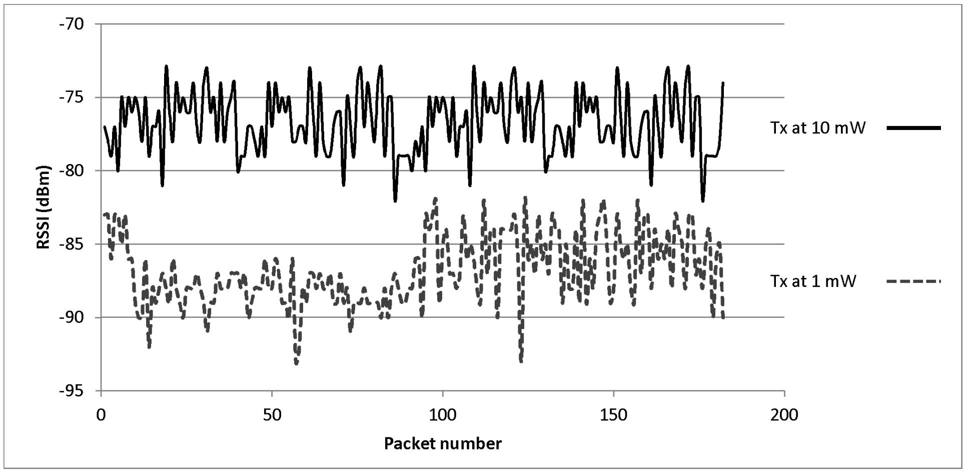


Figure 4.13: RSS values captured over the distance of 1.1 km with Tx at 1 mW and 10 mW

being assembled and tested. However, the sensory data requires for less bandwidth than the images, therefore, by testing the network with the photos we are effectively testing its transmission limits. In addition to the main performance evaluation we have shown how our design makes use of the advanced features provided by the XTend and the DigiMesh protocol layers. With our API, each SPOT can prioritize each data stream to minimize self-interference problems that could occur from the concurrent SPOTs communicating nearby. The communication framework is now ready for deployment on the first SPOT which will be launched in the next few months.

Chapter 5

Conclusion

At the beginning of this research project we explained our main goal of building a long-range communication framework. Initially, our main application for the developed framework was integrating it into the fleet of autonomous search and rescue dirigible UAVs because it offers a longer endurance than most of the existing UAVs for the same price range of \$1000-\$5000. The existing civilian UAV platforms do not have the capability of flying for longer than 30-45 minutes with a payload of 1+ kilograms and most of them lacked long range communication systems. SPOT's longer endurance and high payload allows flying it further and requires long communication links. We went from drafting communication system requirements for the SPOT, surveying the market and selecting the radio technology that could support our requirements, and building the remaining system components for integration with the flight controller. Our communication framework finds applications over a long distance using multi-hop reliable communication with mobile radio nodes.

In addition, our implemented framework can be used to support research projects involving the performance evaluation of new emerging ad hoc routing techniques, distributed wireless sensor platforms, and multiple UAV control algorithms. As the main research problem stated in section 2.4.2, currently there is an urgent need for a decentralized mesh communication in several projects working on the development of cooperative multi-robot systems.

5.1 Thesis Contributions and Achievements

This thesis introduces a communication framework that would enable multiple long endurance UAVs autonomously operate individually or as a group in search and rescue missions with minimal supervision from the GCS. As part of the framework, the long-range 9XTend radios allow operating a group of SPOTs at greater distances by making full use of the allowed transmit power range. The radios are integrated into the SPOT through a serial data converter between UART, RS-232, and USB interfaces. The designed API manages connections and data for the flight controller on the VIA Embedded ARTiGO Pico-ITX form factor single-board computer. In addition to the core functionalities, the designed and developed API provides functions for remotely adjusting power on any desired UAV in the network, route tracing, traffic prioritization, and subnet management within the main mesh network. The main contributions of this thesis are:

1. Design and development of the communication framework consisting of the data management layer, the decentralized routing protocol DigiMesh, and the associated radios. Using transmit power between 1 mW and 1 W the achievable one-hop communication range, with half-wave (17 cm for 900 MHz) dipole antennas, is 1-6 km. The framework maintains an average constant throughput of 31-34 Kbps, as long as the RSS is kept above the radio's sensitivity.

2. Design and development of the data management layer as an API, in C programming language, for the flight controller. This API combines transmission of control commands, sensory data, and photos. At the transmitter this layer splits photos and at the receiver node reassembles them. The designed and implemented algorithm for packetizing data uses a simple protocol between a receiver and a transmitter that automatically determines when a transmission has been completed and the link can disconnected.

3. The framework achieves maximum reliability in each connection and photos with high resolution are received without errors. By using acknowledgements for each transmitted packet, the data management API controls the flow of transmitted data while DigiMesh establishes routes to destinations. At the receiver node, the algorithm that reassembles the data makes sure that multiple photos, received from the same transmitter, are not overwritten while the dirigibles are airborne.

5.2 Future Work

The focus of the upcoming research should be using antenna diversity techniques for improving the stability of links operating at their marginal values. The issue of the link stability occurs in the space domain, therefore trying to create new coding techniques in the software is not as efficient as using the diversity techniques. The main principle behind the antenna diversity is that the same signal can be detected through one antenna while the other antenna shows no signal, even though the separation between antennas is a fraction of a wavelength [60]. It is the nature of the radio signal propagation called fading that makes antenna diversity techniques beneficial with simple design. So the goal for the antenna diversity design in our communication framework would be to use multiple antennas placed on the SPOT and receive data through the one providing the highest RSS. In addition to link stability improvements and depending on the hardware setup, the antenna diversity can make sure that the maximum allowed transmit power settings are used under the government regulations. The maximum ERP can be maintained at all time by using different antennas for different transmit power levels.

Bibliography

[1] Lehmann Aviation. UAV aerial-monitor oil and gas wells, facilities, pipelines. *Oil, Gas & Petrochem Equipment*, 58:11, 2012.

[2] Tucker Balch and Ronald C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1:27–52, 1994.

[3] Quirino Balzano, John Rzasa, Stuart Milner, and Christopher Davis. High capacity tactical networks with reconfigerable, steerable, narrow-beam agile point-to-point rf links. In *IEEE Military Communications Conference*, 2007.

[4] Giacomo Bernardi, Peter Buneman, and Mahesh Marina. Tegola tiered mesh network testbed in rural scotland. In *ACM workshop on Wireless networks and systems for developing regions*, 2008.

[5] Azzedine Boukerche. *Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks*. Wiley-IEEE Press, 2008.

[6] Timothy X Brown, Brian Argrow, Cory Dixon, Sheetalkumar Doshi, Roshan-George Thekkekunnel, and Daniel Henkel. Ad hoc ua v ground network (augnet). In *3rd Unmanned Unlimited Technical Conference*, 2004.

[7] Antonio Capone, Stefano Napoli, and Alberto Pollastro. Mobimesh: An experimental platform for wireless mesh networks with mobility support. In *Workshop on "Wireless mesh: moving towards applications"*, 2006.

[8] Timothy H. Cox, Christopher J. Nagy, Mark A. Skoog, Ivan A. Somers, and Ryan Warner. Civil UAV capability assessment. Technical report, NASA, 2004.

[9] Carlos de Morais Cordeiro and Dharma Prakash Agrawal. *Ad Hoc and Sensor Networks: Theory and Applications (2nd ed.)*. World Scientific Publishing Company, 2011.

[10] Digi International Inc. *XTend^{TM} RF Module*, 90002166a edition, 09 2012.

[11] Nathan Douglass, Lorne Hengst, Charlie Huddleston, and Anna Spicer. Unmanned aerial vehicle report. Technical report, Oklahoma State University, December 2004.

[12] Chip Elliott and Bob Heile. Self-organizing, self-healing wireless networks. In *2000 IEEE Aerospace Conference Proceedings*, 2000.

[13] Kursat Eroglu. The worldwide approval status for 900 mhz and 2.4 ghz spread spectrum radio products. In *IEEE International Symposium on Electromagnetic Compatibility*, 1998.

[14] Jeremi Gancet, Gautier Hattenberger, Rachid Alami, and Simon Lacroix. Task planning and control for a multi-uav system: architecture and algorithms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[15] Niklas Goddemeier, Kai Daniel, and Christian Wietfeld. Coverage evaluation of wireless networks for unmanned aerial systems. In *IEEE GLOBECOM Workshops*, 2010.

[16] Niklas Goddemeier, Sebastian Rohde, Kai Daniel, and Christian Wietfeld. Communication constrained mobility and topology management for relay sensor networks. In *IEEE WCNC 2011 - Network*, 2011.

[17] Zuraida Binti Abdullah Hani and Mohd Dani Bin Baba. Designing routing protocols for mobile ad hoc network. In *IEEE National Conference on Telecommunication Technology Proceedings*, 2003.

[18] G.W. Hoffmann, D.G Rajnarayan, S.L. Waslander, D. Dostal, J.S. Jang, and C.J. Tomlin. The stanford testbed of autonomous rotorcraft for multi agent control (starmac). In *The 23rd Digital Avionics Systems Conference*, 2004.

[19] Julian Hsu. Performance of mobile ad hoc networking routing protocols in large scale scenarios. In *IEEE Military Communications Conference*, 2004.

[20] Industry Canada. *Radio Standards Specification 210, Issue 7, Low-power Licence-exempt Radiocommunication Devices (All Frequency Bands): Category I Equipment.*, 2007.

[21] Muddesar Iqbal, Xingheng Wang, David Wertheim, and Xu Zhou. Swanmesh: A multicast enabled dual-radio wireless mesh network for emergency and disaster recovery services. *Journal of Communications*, 4:298–306, 2009.

[22] Chen J., Fei Q., and Geng Q. The design of data link for multi-UAVs. In *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 2012.

[23] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The click modular router. In *ACM Transactions on Computer Systems*, 2000.

[24] Simon Lacroix, Rachid Alami, Thomas Lemaire, Gautier Hattenberger, and Jérémi Gancet. Decision making in multi-uavs systems: Architecture and algorithms. In *Multiple Heterogeneous Unmanned Aerial Vehicles*, 2007.

[25] Thomas Laengle, Tim C. Lueth, and Ulrich Rembold. A distributed control architecture for autonomous robot systems. In *International Conference on Informatics and Control*, volume 4, pages 411–432, 1997.

[26] Tony Larsson and Nicklas Hedman. Routing protocols in wireless ad-hoc networks - a simulation study. Master's thesis, Lulea University of Technology, Stockholm, 1998.

[27] Liza Abdul Latiff and Norsheila Fisal. Routing protocols in wireless mobile ad hoc network - a review. In *IEEE The 9th Asia-Pacific Conference on Communication*, 2003.

[28] Stephanie Levy. Aftershock: Unmanned systems help japan recover from disaster. *Mission Critical*, 1:18–22, 2011.

[29] Jun Li, Zhexiong Wei, Yifeng Zhou, Mathieu Deziel, Louise Lamont, and F. Richard Yu. A token-based connectivity update scheme for unmanned aerial vehicle ad hoc networks. In *IEEE 2012 Ad-hoc and Sensor Networking Symposium*, 2012.

[30] Mariano I. Lizarraga, David M. Ilstrup, Gabriel Hugh Elkaim, and James Davis. Aerial photography using a nokia n95. In *World Congress on Engineering and Computer Science*, October 2008.

[31] Danielle Lucey. Minicopters take the art of photography to the skies. *Mission Critical*, 2:29–34, August 2012.

[32] Danielle Lucey. Virginia tech researchers seek out mysterious blankets of biomaterials. *Mission Critical*, 2:17–93, May 2012.

[33] Michael J. Matczynski. A distributed embedded software architecture for multiple unmanned aerial vehicles. Master's thesis, Massachusetts Institue of Technology, 2006.

[34] I. Maza, K. Kondak, M. Bernard, and A. Ollero. Multi-uav cooperation and control for load transportation and deployment. *Journal of Intelligent and Robotic Systems*, 57:417–449, 2010.

[35] IntelliTech Microsystems. Vector p fixed-wing uav. `http://www.uavm.com/uavfixedwing.html`. Online; accessed 1-April-2013.

[36] Hassen A. Mogaibel and Mohamed Othman. Review of routing protocols and its metrics for wireless mesh networks. In *IEEE International Association of Computer Science and Information Technology*, 2009.

[37] The Model Aeronautics Association of Canada. Canadian frequency chart. `http://www.maac.ca/freq_chart.php`, 2012. Online; accessed 24-December-2012.

[38] MyungGu Park and Jaeil Jung. An analysis of communication performance according to antenna directionality in UAV operation environment. In *2nd IEEE International Conference on Network Infrastructure and Digital Content*, 2010.

[39] Lynne E. Parker. Current state of the art in distributed autonomous mobile robotics. In *Distributed Autonomous Robotic Systems 4*, 2000.

[40] E. Pastor, J. Lopez, and P. Royo. A hardware/software architecture for UAV payload and mission control. In *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, 2006.

[41] Rabin Patra, Sergiu Nedevschi, Sonesh Surana, Anmol Sheth, Lakshminarayanan Subramanian, and Eric Brewer. Wildnet: design and implementation of high performancewifi based long distance networks. In *4th USENIX Symposium on Networked Systems Design & Implementation*, 2007.

[42] Georgiy Pekhteryev, Zafer Sahinoglu, Philip Orlik, and Ghulam Bhatti. Image transmission over ieee 802.15.4 and zigbee networks. In *IEEE International Symposium on Circuits and Systems*, 2005.

[43] Robert D. Poor. Wireless embedded networks: Connectivity for everyday objects. In *International Conference on Consumer Electronics, 2005: Digest of Technical Papers*, 2005.

[44] Chen Qiaoyun, Liu Junmin, and Ning Bingwu. Design and test of zigbee network in storehouse management. In *4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008.

[45] Steven Recoskie, Atef Fahim, Wail Gueaieb, and Eric Lanteigne. Experimental testing of a hybrid power plant for a dirigible UAV. *Journal of Intelligent & Robotic Systems*, 69:69–81, 2013.

[46] Wolfgang W. Rochus. UAV data-links: Tasks, types, technologies and examples. In *Development and Operation of UAVs for Military and Civil Applications*, September 1999.

[47] Sebastian Rohde, Niklas Goddemeier, Kai Daniel, and Christian Wietfeld. Link quality dependent mobility strategies for distributed aerial sensor networks. In *IEEE Globecom 2010 Workshop on Wireless Networking for Unmanned Aerial Vehicles*, 2010.

[48] Elizabeth M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. In *IEEE Personal Communications*, 1999.

[49] Allison Ryan, Marco Zennaro, Adam Howell, Raja Sengupta, and J. Karl Hedrick. An overview of emerging results in cooperative UAV control. In *43rd IEEE Conference on Decision and Control*, 2004.

[50] Jenvey S., Gustafsson J., and Henriksson F. A portable monopulse tracking antenna for UAV communications. In *22nd International Unmanned Air Vehicle Systems Conference*, 2007.

[51] D. Frousheger S. Hrabar, T. Merz. Development of an autonomous helicopter for aerial powerline inspections. In *IEEE 1st International Conference on Applied Robotics for the Power Industry*, pages 1–6, October 2010.

[52] Z. Sarris. Survey of UAV applications in civil markets. In *IEEE Mediterranean Conference on Control and Automation*, June 2001.

[53] T. Shepherd. *Decentralized Channel Management in Scalable Multihop Spread-Spectrum Packet Radio Networks.* PhD thesis, MIT, Laboratory for Computer Science, 1995.

[54] D. H. Shim, H. J. Kim, and S. Sastry. Hierarchical control system synthesis for rotorcraft-based unmanned aerial vehicles. In *American Institute of Aeronautics & Astronautics Guidance, Navigation and Control Conference*, 2000.

[55] S. Singh and P. Ranjan. Towards a new low cost, simple implementation using embedded system wireless networking for UAVs. In *IEEE 5th International Conference on Advanced Networks and Telecommunication Systems*, 2011.

[56] Katayoun Sohrabi, Jay Gao, Vishal Ailawadhi, and Gregory J. Pottie. Protocols for self-organization of a wireless sensor network. In *IEEE Personal Communications*, 2000.

[57] James Stevens. Spatial reuse through dynamic power and routing control in common-channel random-access packet radio networks. SURAN Program Technical Note (SRNTN) 59, 1998.

[58] Michael R. Sweet. *Serial Programming Guide for POSIX Operating Systems.* Michael R. Sweet, 2010.

[59] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, COM-32 (3):246–257, 1984.

[60] Rappaport T.S. *Wireless Communications: Principles and Practice (2nd Edition).* Prentice Hall, 2002.

[61] George Vachtsevanos, Ben Ludington, Johan Reimann, Panos Antsaklis, and Kimon Valavanis. Modeling and control of unmanned aerial vehicles current status and future directions. In *Workshop on Modeling and Control of Complex Systems (MCCS)*, 2005.

[62] Geoffrey Werner-Allen, Patrick Swieskowski, and Matt Welsh. Motelab: A wireless sensor network testbed. In *Fourth International Symposium on Information Processing in Sensor Networks*, 2005.

[63] Hanno Wirtz, Tobias Heer, Robert Backhaus, and Klaus Wehrle. Establishing mobile ad-hoc networks in 802.11 infrastructure mode. In *WiNTECH '11 Proceedings of the 6th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, 2011.

[64] Mariusz Wzorek, David Landen, and Patrick Doherty. GSM technology as a communication media for an autonomous unmanned aerial vehicle. In *21th Bristol UAV Systems Conference*, 2006.

[65] Yang Xiao. *Handbook on Sensor Networks.* World Scientific Publishing Company, 2010.

[66] Guoqing Zhou and Deyan Zang. Civil UAV system for earth observation. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, 2007.