SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: June 12, 2021

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Mark Rizko
Cameron Burdsall

ENTITLED

Drone-based Wireless Communications for Disaster Recovery

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

B. Dezfouli

Thesis Advisor

Nam Ling
Nam Ling (Jun 14, 2021 11:47 PDT)

Department Chair

Drone-based Wireless Communications for Disaster Recovery

by

Mark Rizko
Cameron Burdsall

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 12, 2021

Drone-based Wireless Communications for Disaster Recovery

Mark Rizko
Cameron Burdsall

Department of Computer Science and Engineering
Santa Clara University
June 12, 2021

ABSTRACT

This project aims to establish a drone system that can deploy a wireless mesh network over a disaster area, which would aid in the process of finding survivors by using wireless communications to identify where victims are and allow authorities to send out alerts to people in the area. We also seek to add a device detection feature that would allow the drones to passively look for devices through WiFi and Bluetooth Low Energy, giving disaster responders the ability to actively identify specific devices, locate zones where victims lie, and discern a rough population estimate of that area. Those that find themselves in the middle of the disaster situation are able to use their own devices to establish a connection with the outside world. Under a mesh topology, the drones will act as network nodes that will dynamically connect to each other which enables a more resilient configuration. The drones will be flying above an area to enable wireless connectivity for first responders as well as those in distress, ensuring reliable detection and a critical communications pipeline when it is needed the most.

Table of Contents

List of Figures

Chapter 1

Introduction

In this thesis, we will discuss various modern drone and computer technologies and how they might be used from the point of view of a disaster response team. First, we will examine the evolution of drone technologies and what is available on the market today. We also discuss the importance of wireless communications in a disaster scenario and how a basic network can be implemented and deployed using software packages. Finally, we observe how the signals that our mobile devices use to find WiFi networks and Bluetooth devices can be used to quickly and reliably check the presence of people in an area.

1.1 Disaster Scenarios

Disasters have unfortunately become a prevalent part of modern life for many people. Earthquakes, floods, hurricanes, and their consequences are expected to become more common as climate change worsens. Professional disaster response has been a critical component in mitigating the damage done by these events and saving lives. However, there is still room for vast improvement, and we anticipate the opportunities to improve disaster response will continue to grow as technology advances.

1.2 Drone-Based Recovery

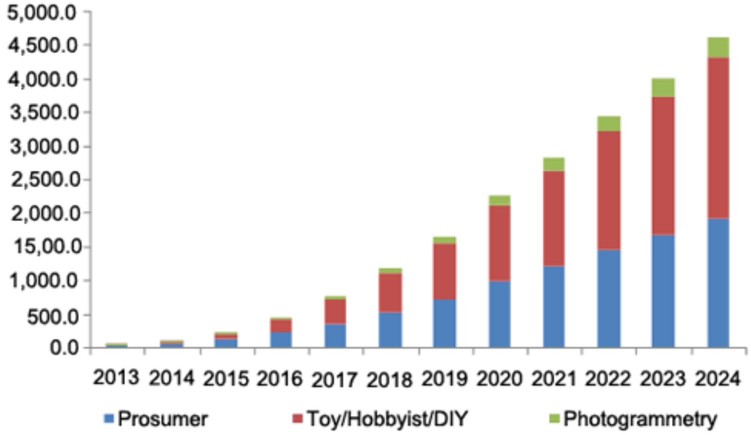


Figure 1.1: North America consumer drone market by technology, 2012 - 2022 (USD Million)

Commercial drone technology has grown substantially over the past decade through many technological developments signifying the emergence of an ’innovation shock’ (6). Industry research has projected that ”the global drone market is estimated to grow from $2 billion in 2016 to nearly $127 billion in 2020” (6). These projections draw attention to the increasing interest and possibilities of drone technologies to achieve much greater outcomes for our society than the simple RC toys that commercial drones were known for only a decade ago. Through these advancements, many applications for drone technology have emerged, such as one home automation drone that uses a weight-sensing drone ”to maintain an Eco-friendly environment by removing the garbage dumpsters located outside the home” (12). Systems like these work in collaboration with IoT microprocessors such as the Raspberry Pi, where such an open-ended system has no limit on the possibilities of applications that can be made. This project aims to show how advanced networking techniques can be used in conjunction with drones to create a mesh network system and wireless device sniffer for disaster recovery.

1.3 The Role of Wireless Communication

Wireless communications have become one of the backbones of today’s communications infrastructure, and is one of the primary ways people receive news and other critical information (5). In disaster scenarios, often times wireless communications are taken down when cell towers are destroyed or the power grid is disabled. Despite this, people will still likely carry their mobile devices in hopes of communications being restored at some point.

Reestablishing wireless communications during an emergency can help deliver important information to those who need it most. Maps, warnings, and evacuation areas can be communicated with those in a disaster area without any physical interaction, allowing manpower to be saved and reallocated. Our mobile devices are also constantly emitting signals in order to facilitate wireless communications. These signals can be used in a disaster scenario to help pinpoint the location of survivors who may be carrying a mobile phone or other devices. A drone equipped with ability to detect these signals could be leveraged by disaster recovery teams to help locate people, especially those who live in isolated towns in hard to reach terrain. There is a high likelihood of an area being populated if many wireless signals are detected by the drone.

1.4 Thesis Structure

This thesis will first highlight related works that helped inspire the design choices made during this project. These projects show the effectiveness of using wireless technology to aid in disaster recovery. Next, we will review the existing network and drone technologies that this project was built on before transitioning into our proposed solution. After presenting our solution, we will review performance results and discuss further improvements that can be made on this project to increase the reliability and effectiveness of our system. We will then conclude with some final words, acknowledgments, and references.

Chapter 2

Related Work

Project OWL (2) is an open source initiative to develop wireless network technology for rugged environments, especially those affected by natural disasters. They deployed a mesh network during Hurricane Maria which consisted of access points enclosed in buoyant and water proof ”ducks,” which would float around the affected area and enable rudimentary communications. This project leverages LoRa technology to allow nodes to connect to each other from much farther distances than supported by WiFi. The downside is that the throughput of the network is severely hampered by LoRa’s data rate of 300bps - 37.5kbps. This means that only essential communications can really be permitted on the network, and high throughput applications are unfeasible.

The author in (13) discuss the main challenges of estimating occupancy of an area by analyzing WiFi Probe requests received from unique devices. They first propose a method of occupancy estimation indoors using a ”WiFi Pineapple”, and then dip into estimating population outdoors by having the ”WiFi Pineapple” fly around on a drone and pairing the received probe packets with a GPS coordinate. Unfortunately, they did not make much progress solving the issue of MAC randomization, which at the time was only a feature of iOS devices. They were able to overcome the issue by having many WiFi Pineapples collecting packets simultaneously, so devices are only considered ”detected” when more than a certain number of nodes see its probes. Since then, MAC randomization has become industry standard with its incorporation in iOS 8 and Android 10, and is a big consideration when attempting to do any sort of tracking via probe requests.

The authors of (11) estimate the population of a live concert by having ten ”agents” walk around acting as Bluetooth probes. They were able to detect almost 80% of all Bluetooth devices at the concert. While they found that only 8.2% of people actually had visible Bluetooth devices, the potential of estimating the population in an area based on random samplings of wireless signals becomes apparent.

Chapter 3

Existing Technology

3.1 Wireless Network Technology

3.1.1 WiFi and Mesh Networks

A computer network can be described as a group of computers communicating with each other over common established protocols via physical, optical, or wireless medium. In order for an end device, like a personal computer or mobile phone to connect to the Internet, it must first connect to a Local Area Network, or LAN, that has an external connection to the Internet. This external connection is managed by a router, which directs connections between the LAN's connected devices and the larger Internet. In order to connect to a LAN wirelessly, a technology known as WiFi is used. The protocols for WiFi are defined by the IEEE 802.11 standard and its derivatives (3).

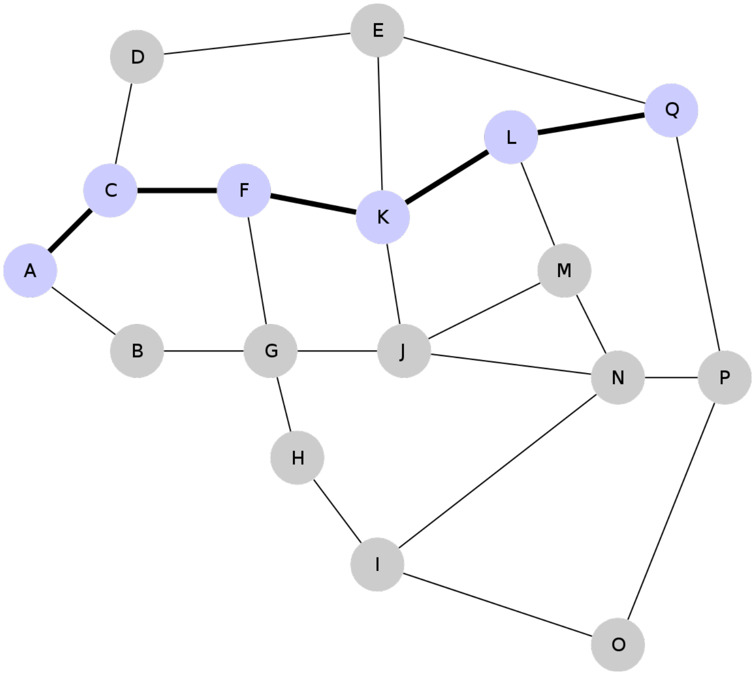


Figure 3.1: Example of a Mesh Network Topology

Networks can be configured in a variety of ways; however, many of these ways are prone to massive network failures if single nodes in the network fail. Consider only the nodes A, C, D, and F in Figure 3.1. If node C were to fail, all others would lose connectivity to each other. Now consider the entire network with all nodes. All the nodes in the network are connected to at least two other nodes, resulting in no single point of failure in the network. This ensures that if one node fails, the network should still work. This is referred to as a mesh network topology (14).

This type of topology is ideal when a reliable network is needed that can tolerate several node failures. For our project, we will be deploying a network with a mesh topology for this exact reason. If one drone were to lose power or get taken out of the sky, the network should be able to tolerate the loss of the node and reconfigure as necessary.

3.1.2 Packets and Frames

A packet is a formatted piece of data that is sent across a computer network. A packet contains information called the source and destination IP address. A frame is essentially a packet that holds additional information about how the packet is to travel across a physical medium, such as an Ethernet cable or over a radio wave (WiFi).

In order for an end device to know what wireless access points (APs) are nearby, it must send out a probe request frame. These requests are received by APs and are responded to with a probe response, which contains information about the wireless access point. APs can also broadcast their presence by sending beacons, which is typically how a smartphone or computer identifies the wireless networks around it.

A wireless interface can be set to detect and collect these probe frames by using a software such as TCPDump in conjunction with a wireless interface that is capable of being put into monitor mode (aka Radio Frequency Monitor or RFMON). This allows us to detect the probes sent out by devices when they are looking for a wireless network to connect to. In this thesis, we will investigate how we can use these detected signals to have an indicator on where people might be located in a disaster based on the signals constantly emitted by their devices.

3.1.3 MAC Addresses and Randomization

Media Access Control addresses, or MAC addresses, are unique addresses assigned to a Network Interface Chip (NIC) based on the manufacturer of the chip, allowing devices to be uniquely identified by their manufacturer. Network administrators have used this technique to track and identify individual's activities on their network. Like IP Addresses for packets, MAC Addresses are used in frames to show what interface a frame is coming from and where it is going to.

Modern mobile devices currently employ a newer security method known as MAC randomization. This method ensures that a device changes the MAC address it broadcasts when sending WiFi or Bluetooth probe requests and connecting to networks in order to reduce the chance that it can be tracked across or in between networks. MAC randomization helps ensure device anonymity and reduce the effectiveness of packet sniffing to determine meaningful information about a device.

3.1.4 Bluetooth Low Energy

Bluetooth has become an industry standard wireless communication technology. It is used for short range wireless communications between Bluetooth enabled devices, with applications including hands free phone systems, car infotainment systems, as well as wireless speakers and headphones. Bluetooth Low Energy (BLE) was introduced in 2011 as a more lightweight and power-efficient rendition of Bluetooth that could be used for smaller applications where power consumption is important. Many different applications of IoT technology depend on Bluetooth Low Energy to achieve their desired behavior and length of operation. A very important consideration and design constraint for Bluetooth Low Energy is the ability to last a reasonably long amount of time on coin cell batteries (7). Through many optimizations such as the use of 3 channels instead of 79, simplified packets, and reduction in receiving time, Bluetooth Low Energy sacrifices some performance from Bluetooth Classic for the purpose of reducing power consumption as much as possible.

The devices using Bluetooth Low Energy are typically resource-constrained peripherals such as simple sensors, however, Bluetooth Low Energy is supported on complex devices like smartphones which allow us to sniff user devices through this protocol. The rate at which devices broadcast BLE advertisement signals is significantly faster compared to WiFi signals, which justifies the inclusion of this technology in our sniffer. The trade-off for this accelerated sniffing rate is considerably less range than WiFi, however, formulating conclusions based on the correlation between the WiFi and BLE signals will be sufficient for detecting victim devices.

3.2 Low Power IoT Technology

One of the main goals for our design was to limit the amount of power used as much as possible. Obviously, the drone will take up a considerable amount of power, by far the most out of the entire system. There is not much improvement to make to the power the drone will consume, as that is outside of the scope of this project. However, we employ some power saving techniques in order to reduce the amount of power the Raspberry Pi system takes.

The first method employed is to disable any and every unnecessary hardware feature on the Raspberry Pi. Disabled modules include the on-board Bluetooth interface, the unused Ethernet port, and unused USB ports. In addition, the choice of operating system also reduces power draw from the system. We chose to deploy a Raspberry Pi OS Lite image on the Raspberry Pi. This version of the Raspberry Pi OS (formerly known as Raspbian) is stripped of all unnecessary software packages and services. This allows us to not only save storage space, but also have less processes running in the background. This OS also does not have a GUI (Graphical User Interface), and can only be interacted with via CLI, or Command Line Interface (1).

3.3 Drone Technology

3.3.1 MAVLink Protocol

For a very long time, drones and other MAV (micro air vehicle) devices have been mainly controlled using a physical RC controller which would require the user to be in the vicinity and line of sight of the vehicle. Eventually, however, the MAVLink protocol was developed in order to allow developers to communicate with MAV devices remotely, advancing the autopilot technologies for these vehicles. The MAVLink protocol is a very lightweight means of communication between unmanned MAVs and a ground control system (8). One of the beauties of the MAVLink protocol is that it allows communication between different transport layers and mediums due to its lightweight structure (8).

3.3.2 Pixhawk 4

Pixhawk 4 is an advanced, state of the art autopiloting system for academic and commercial drone developers. Developed by the PX4 team and Holybro, who manufactures the drone kit used for this project, Pixhawk 4 has progressed as the standard for open-source framework for MAV (micro air vehicles) (10).The purpose of Pixhawk for commercial drone users and companies developing drone technologies is to reduce overhead in developing autopiloting systems, which can sometimes mean the exclusion of an entire research and development team. The portability of Pixhawk 4 is impressive as well, allowing the technology to be used on a wide range of MAV applications with the open-source potential to constantly improve over time. The reliability and interoperability of Pixhawk 4 as well as the years of development behind the technology make it the right choice for the drone autopilot component of this project.

Holybro made the setup process for Pixhawk on their drone kit very easy, requiring that the included Pixhawk 4 board be plugged into the rotors through a PWM cable, which allows precise control over the power provided through the cable in order to adjust rotor speeds. Additionally, the Pixhawk board is also connected to the included power management board (which is plugged into the battery), the GPS module, and finally the telemetry radio attached to the drone. The board uses the telemetry radio to communicate (through MAVLink protocol) with ground control, feeding in data from the attached GPS module as well as receiving data to the rotors. Holybro's easy-to-use configuration of the Pixhawk allowed us to get the drone up and running with very little problems regarding the autopilot system.

3.3.3 QGroundControl

The author in (4) discusses the function of Ground Control systems (GCS) as ”architectures comprising necessary hardware to operate communications or control input to remote UAVs, and a software interface enabling pilots to operate out of line-of-sight” (4). As the industry for professional and commercial drone technology is constantly expanding, QGroundControl is another breakthrough made in drone technology which reduces the development time and overhead of drone projects. QGroundControl operates using the MAVLink protocol, meaning it is compatible with

Pixhawk 4 devices and the combination of the two drone technologies is not uncommon at all in the drone development community today.

With QGroundControl, the user is able to remotely control the drone from many miles away (this of course depends on the autopilot transmitter being used) and allows drone routes and missions to be planned from the ground control station. This functionality allows us to deploy the network to a wide range of locations, including remote ones, so long as the radio transmitter on the drones can reach. This gives the project the potential to have one operating base that is able to monitor and provide a network for entire countries.

Chapter 4

Proposed Solution

4.1 Drone Construction and Operation

The drone kit we used for this project was cleverly designed to be built and operated without much experience or difficulty, which presents an advantage regarding the manufacturability of our system. Holybro provided clear instructions for the assembly of the S500 drone that we believe is suitable for anyone capable of using simple tools. The first challenge in regards to the operation of the drone was the radio calibration process, which was prone to slight malfunctions with the RF transmitters provided. This issue was resolved through the installation of updated drivers, and the mission planning process begun. A number of test flights were needed to collect performance data as well as verify that the drone was fully functional. Planning missions in the QGroundControl software was not a challenge due to the intuitive nature of the program. Using a live map (similar to Google Earth) with the position of the drone provided by the onboard GPS module, the operator of the drone simply places points on the map while setting parameters for each point. These parameters include altitude, hold duration, yaw, and speed.

After configuring the missions in QGroundControl the drone was ready for the first test flight, which is where the next challenge presented itself regarding the storage of the battery. Originally, the plan was to design a 3D printed chassis for the battery and Raspberry Pi that would unify and ensure the security of our components. Our lack of expertise with CAD and limited access to 3D printers during COVID derailed this plan, prompting us to pivot towards securing these components to a metal strip under the drone that was initially designed for a smaller battery. Upon securing the components, we began recording our first flights for the power consumption measurements while moving and hovering.

4.2 Deploying Wireless Network Infrastructure using Drones

In order to achieve our use case of deploying a wireless network from a drone, we leveraged a software called hostapd, or Host Access Point Daemon. This software allowed us to turn a normal wireless interface into a wireless access point that other devices could connect to. The software also facilitates routing the connections within the created network to another network, meaning that if we route the connections from the Raspberry Pi to another router with internet access, then so will the network deployed by the Raspberry Pi. We propose using hostapd with a Raspberry Pi device coupled with a drone and battery to facilitate the deployment of a wireless network from one of the wireless interfaces on the Raspberry Pi, as shown in Figure 4.1.

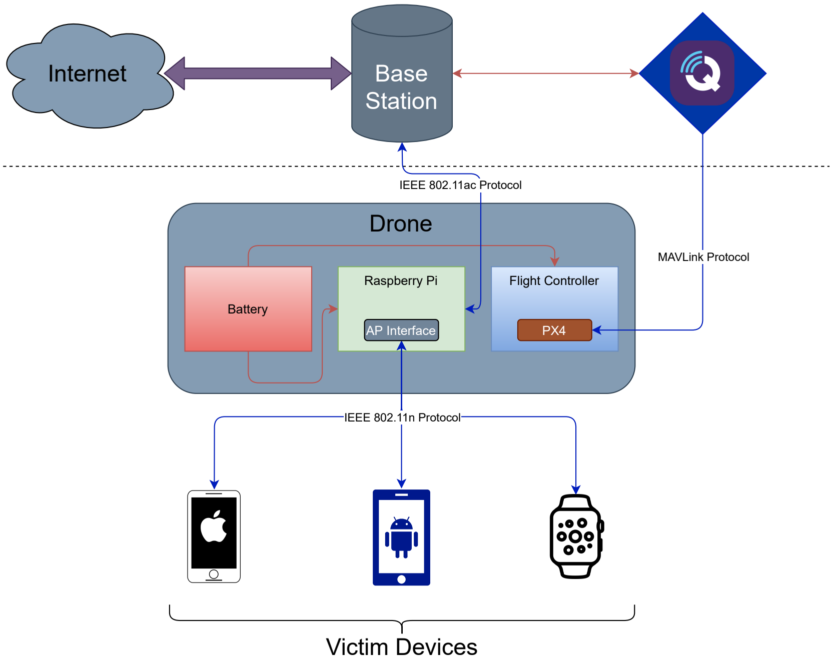


Figure 4.1: System Architecture for Wireless Network

Figure 4.1 describes the system architecture for our wireless network configuration. Starting with the victim devices, they are connected to the network being facilitated by the Raspberry Pi over the IEEE 802.11n protocol. The Raspberry Pi and drone components are being powered by the same battery and are all part of a single physical unit. The Pixhawk 4 flight controller board on the drone allows it to communicate with the QGroundControl software running on the base station computer through the MAVLink protocol. The Raspberry Pi facilitates the network connection through the 802.11ac protocol to the base station, which is the only component that is actually connected to the internet.

4.3 Estimating Nearby Population Via WiFi Probe Request Packets

As mentioned in Chapter II, there has been much progress made in regards to estimating the population in a room or area by analyzing the probe request packets being broadcasted. However, none of the mentioned works were able to overcome the issues in population estimation that are caused by the widespread adoption of MAC randomization technology, namely how accurately we can estimate the number of people nearby if each device detected is not guaranteed to send the same MAC addresses with each probe request they send. Previous methods, such as in (11) have estimated population by actively scanning for BLE frames and reporting the number of unique addresses detected within a certain RSS range.

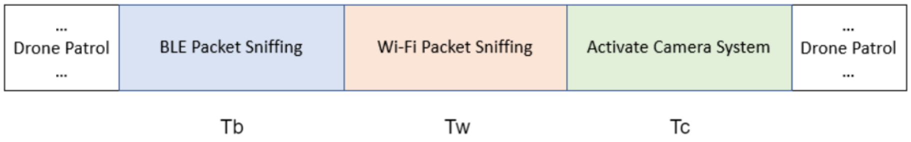


Figure 4.2: Drone scanning phases

We propose using the ideas from these previous works coupled with our drone system to allow the system to be able to record the number of nearby devices, which can be used as a metric of how many actual people may be nearby. Our sister project involves using cameras and FLIR (Forward Looking InfraRed) imaging on a drone to identify people using machine learning and computer vision concepts. Our method of finding people can be used as a precursor to decide whether or not to spend the large amount of power on activating and using the camera system. Since BLE uses much less power compared to WiFi, we propose the flow in Figure 4.2. We start with the drone system on patrol, as described in the QGroundControl section. At a designated point during the patrol, the drone will stop and transition to idle. During this idle phase, packet sniffing will commence. First, we activate the BLE sniffer to detect any BLE traffic in the area for a period of time *Tb*. If there are a sufficient number of packets detected, we then continue to log 802.11 probe request packets for a period of *Tw*. If there are a sufficient number of packets detected at this point, the drone system would then continue on to begin scanning with the cameras for a period *Tc*. As discussed in the wireless technologies section, detecting these packets will need to be done in a particular way due to the newly widespread adoption of MAC randomization technologies.

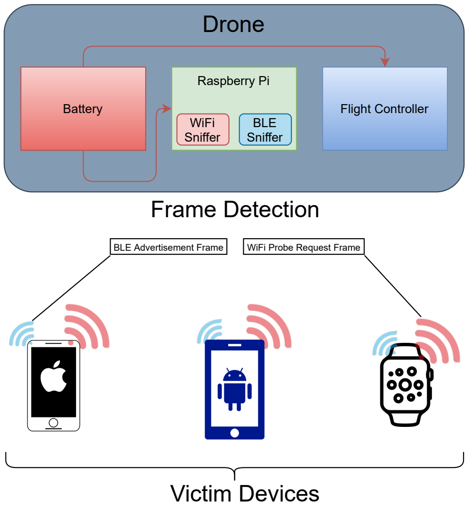


Figure 4.3: System Architecture for Sniffing Interface

Figure 4.3 describes how the sniffing is performed. There are two additional interfaces on the Raspberry Pi: a wireless interface capable of being put into ”Monitor” or ”Promiscuous” mode, and a BLE interface that is capable of sniffing BLE frames. The WiFi sniffer is looking for WiFi probe request frames, and the BLE sniffer is looking for BLE advertisement frames. The idea is that if the drone is within the broadcasting range of these signals, then the sniffer will be able to detect their presence.

Due to MAC randomization, it is almost guaranteed that we will detect the same device but with a different MAC address if we do a long continuous scan for packets. To circumvent this issue, we propose doing several shorter scans, and considering the average number of packets detected. We also exploit the fact that for the most part, MAC randomization is only applied to mobile and smart devices. Simpler IoT devices (TVs, Tiles, speakers, etc) will primarily broadcast their actual MAC address without randomization. In addition, Windows devices and most Linux distributions do not have MAC randomization turned on by default, but usually do have it implemented as an option to turn on. Therefore, if a detected packet falls outside of the block of manufacturer dedicated MAC addresses, it will most likely be a mobile or smart device.

In order to perform a scan for probe frames in a given area, we first must program in a patrol route for the drone via QGroundControl. We found planning mission routes in QGroundControl very intuitive, simply placing points on the map for the drone to travel, setting altitude at those points, and the speed of the drone during travel. This is advantageous for our system because it allows the operation of the drone to be taught easily to someone who may know the area very well but does not have experience with drone technologies. During the patrol, the drone will stop at designated places for a set period T to perform a scan.

T needs to be a value that will minimize the chance of detecting the same device broadcasting a different MAC address for WiFi probes or BLE advertisements. This can be done by making *T* less than the time it takes on average for a device to transmit a probe or advertisement with a new address. There are many different factors that can affect how frequently a device transmits a probe, such as whether or not it is actively being used, whether or not battery saving methods are turned on, and so on. We assume that most of the time, devices will be inactive (in sleep mode) but otherwise in normal operation (Wireless and Bluetooth enabled, no Airplane mode, etc). The *T* that we are looking for regarding WiFi will henceforth be called *Tw*.

For BLE, we will also need to establish a period in which to scan. Since BLE advertisements are known to broadcast more frequently than WiFi probes, we anticipate that the period for BLE, *Tb*, would likely be less than *Tw*. Despite the sizes on Figure 4.2, the periods of *Tb*, *Tw*, and *Tc* all vary significantly, as each ”scan” will take a different amount of time to reliably accomplish its goal of locating devices.

Chapter 5

Performance Evaluation

This section will review the measurements taken for our system. The probe measurements will display the effectiveness of our sniffing interface with the drone power consumption measurements reviewing the efficiency of our drone while offering suggestions for improvement. All data has been measured multiple times over to check for irregularities and only the final, most accurate of our data sets are displayed below.

5.1 Sniffing

One of the challenges of doing our evaluation was trying to find a location that would minimize the ”noise” from other devices not included in our test. We discovered that it was not feasible to perform these tests at home, and had to go to a remote location with no other devices around in order to get more accurate and readable results.

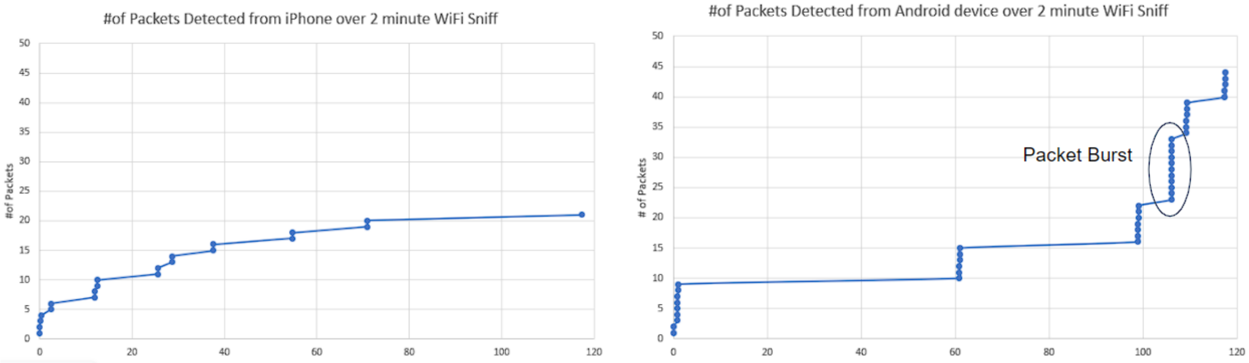


Figure 5.1: WiFi Probes detected during scan on iPhone and Android

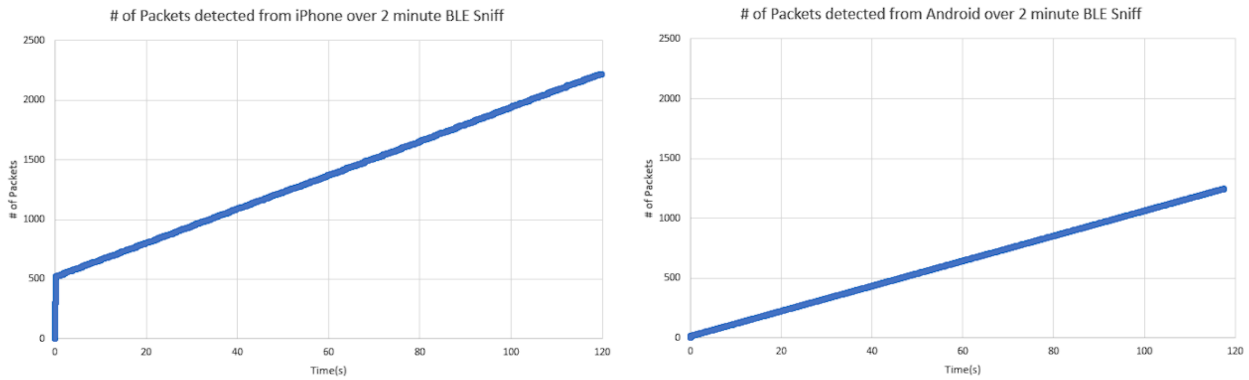


Figure 5.2: BLE Advertisements detected during scan on iPhone and Android

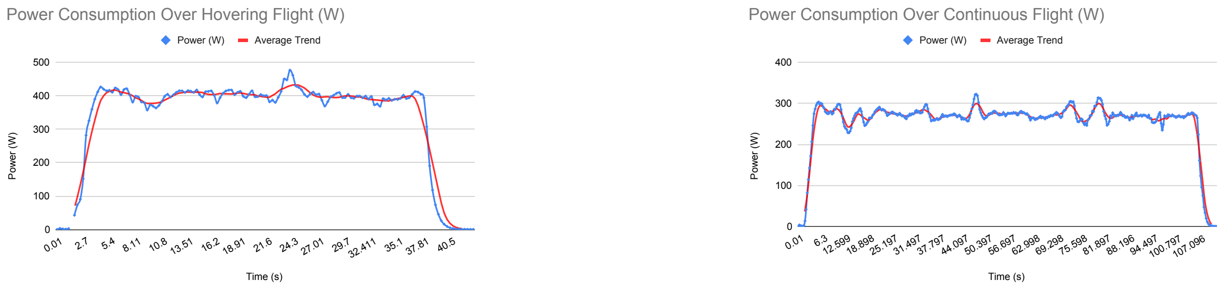
During testing, we attempted to find out how frequently WiFi probes and BLE advertisements are sent by mobile devices. Our goal was to determine a period of time, *Tw* for WiFi and *Tb* for BLE, which maximizes the chance of detecting the signals emitted by devices. This is the period of time that the drone would have to spend either hovering or slowed down. As will be discussed in the next section, we want to minimize the amount of time spent hovering, as it was found to be much more expensive in terms of power usage.

We found that BLE advertisements are emitted far more frequently than WiFi probes, and were thus a more reliable signal to look for when attempting to detect the presence of devices. *Tb* would need to be at most one second, and could reliably assume less without the loss of detection ability. As seen in Figure 5.2, BLE advertisements are sent out far more frequently than the WiFi probes in Figure 5.1. This means that if a device is within the shorter BLE range, it is has a very high chance of having at least one of its advertisements be detected.

Tw would need to be at minimum 1 minute, as the devices that we tested only emitted roughly 1-4 packet ”bursts” per minute (see Figure 5.1). Even with this longer scan period, there is still a likely chance that some WiFi probes could be missed by the scan. This is due to the fact that TCPDump will constantly change the radio channel it is listening on in order to hit the full spectrum of 802.11, which inevitably causes some of the broadcasted probes to be missed. This means that even if a device is within WiFi range, there is a decent chance that broadcasted probes will not even be seen by the sniffer.

5.2 Drone Power Consumption

Measuring the power consumption of the drone was not very straightforward. First, we decided that the comparison would be made between the two phases of operation that the drone would undergo, hovering while facilitating the network and moving while scanning for devices. Due to the three hundred miles that separated the group members, the power consumption was not able to be measured on the drone with the Raspberry Pi attached, which allowed us to use the log files from the drone rather than having to test the hardware directly.



(a) Hovering Flight

(b) Continuous Flight

Figure 5.3: Power Consumption (W) of the Drone

For both measurements, the instantaneous voltage and current during these phases of flights were extracted from the flight logs stored with every mission on the Pixhawk 4 board. The moving flight was measured over a period of roughly over two minutes while the hovering flight was measured over thirty seconds. The measured time on the moving flight was increased to account for turning and maneuvering the drone as it would in a real-life scenario. The hovering flight data measurement period did not require extra time, as the behavior of flight while hovering should not change very much. Referring to the graphs above, more power was consumed over hovering flight than over moving flight, with an average difference of about 120W. This roughly translates to 27 minutes of flight while moving and 19 minutes of flight time while hovering. The moving flight required less power on average due to the translational lift of the drone while moving, which brings an added efficiency to the rotors which allow the drone to travel while consuming less power than expected (9).

These current power readings for the drone imply that the power consumption of the Raspberry Pi is nearly negligible in comparison, as the Raspberry Pi would only consume a maximum of roughly 7W under the heaviest loads. With the drone consuming the majority of the power for this project, we believe sharing the battery between the drone and Raspberry Pi will not bottleneck the flight time of the drone too heavily. Combining the mentioned flight times of hovering and moving flight, we discern a rough estimate of 20-23 minutes of operational flight time. There is definitely room for improvement regarding the power efficiency of the system, which will be discussed more thoroughly in the following section.

Chapter 6

Future Improvements

In this section, we will be describing some of the shortcomings of our project, and improvements to be made in the future. With the difficult circumstances of 2020, a few minor ambitions and goals set at the beginning of the project were not fully met; however, we believe these improvements will further increase the usefulness and effectiveness of our system.

6.1 Power and Battery Considerations

The first component of our system that we would like to see improved in the future is reduced power consumption of not only the drone, but network components as well. Regarding the drone, there are more than a few improvements to be made that would contribute to reducing the power consumption overall. Starting with the battery, we believe a lighter battery could potentially be used without sacrificing too much battery life. Currently, we are using a high capacity 8000mAh battery that adds considerable weight to the drone and we believe there is room for compromise between the weight and capacity of the battery that could make our system more energy-efficient.

Aside from the power consumption alone, we had ideas early in the design phase of our project about using solar-powered chargers for the battery while the system is idle. With our research on this topic mostly finished, the large capacity of our battery and considerably lower charging speed of solar power was not going to be realistically useful for our current system, as the drone would spend far too much time idle charging with solar power. With that being said, we believe there is a place for solar power in the project; some ideas include an on-board solar panel to continuously provide power to the raspberry pi (leaving all battery power to the drone) and using a lower-capacity battery for the drone that would cut down the idle time of the system.

Finally, it was mentioned earlier that we planned on 3D printing a chassis for the battery, Raspberry Pi, and antennas. We believe this would increase the uniformity of our system and protect the valuable components from exposure to outside conditions. These minor improvements are the first step to weatherproofing the system and ensuring the most reliable and efficient operation in all conditions.

6.2 Mesh Network

Another area where future work could be done is to implement a mesh network for the drones using LoRa technology. LoRa was a technology that we researched early in development as a way to greatly extend the range of wireless communications at the cost of throughput. We eventually decided that this would be out of the scope of what we wanted to accomplish in our development time frame, however it is an interesting idea to expand upon for future iterations of this project. LoRa would allow the mesh network to function without having to be within range of the base station, offering a truly portable system prepared for the worst possible terrain and conditions. Additionally, it is possible that future 802.11 protocols may be of use to enhance WiFi range and network throughput.

6.3 Data Collection and Processing

An interesting feature we also wanted to implement during this project was a mobile landing page to collect user information. This landing page would function similar to that of a public WiFi access point, where users are prompted to authenticate themselves for WiFi access. The plan for us was to have a landing page that required the names and condition of those in a victim group to be collected and processed in a database. We believe this would help authorities perform accurate head counts of those affected by the disaster to further aid in recovery.

Chapter 7

Conclusion

This thesis demonstrates that drone technology has great potential for the future of disaster response and its ability to aid search and rescue teams to more effectively save lives. Using drone technology to perform scans on disaster-affected areas reduces the risk of these search and rescue teams and more effectively locates victims through their devices. The novelty of this project lies in our system's ability to sniff WiFi and Bluetooth packets to locate where survivors may be, and ultimately save lives.

We found that detecting the presence of devices was easiest to do by looking for BLE advertisements. Although having a much shorter range compared to WiFi, it is still much easier to detect the signals due to the sheer frequency in which they are transmitted. An effective way to utilize this would be performing longer-range scans on WiFi to discern a radius of potential victims, then using the more responsive BLE signals to closely pinpoint the location.

Another idea which would both lower power usage and per unit cost would be to exclude the WiFi sniffing feature from the system. While looking for WiFi probes does provide useful insight into how many people may be close by, it is not as accurate and high fidelity as the BLE advertisements are, and thus someone looking for the lowest power configuration or lowest cost unit may be interested in cutting this feature.

In conclusion, we hope this thesis shed some light on the endless possibilities of combining modern drone systems with smart IoT technology, and the potential these systems have to save lives. We also hope to have provided some insight into how disaster recovery and safety authorities can leverage the basic signals that our mobile devices send to help locate or at least estimate how many people are nearby a given area. We are honored to work on a project with this much potential, and hope our work will be continued in the future to create a system that contributes to the betterment of our world.

Chapter 8

Acknowledgments

This project would not have been possible without the contributions of many individuals. We would first like to thank our advising Professor, Dr. Behnam Dezfouli, for guiding us along our journey through this project and giving us ideas when we hit roadblocks. We would also like to thank Ron Snyder and Jason Hoac from the Cisco Tactical Operations Disaster Recovery Team for giving us their time and expertise in disaster response technologies and for giving us the opportunity to work with Cisco and learn about so many new and fascinating technologies.

Bibliography

[1] *Raspberry pi os*. https://www.raspberrypi.org/documentation/raspbian/.

[2] *Project owl*. https://www.project-owl.com, 2017. Accessed: 2020-10-12.

[3] F. Cali, M. Conti, and E. Gregori, *Ieee 802.11 wireless lan: capacity analysis and protocol enhancement*, in Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98, vol. 1, 1998, pp. 142–149 vol.1.

[4] T. Dardoize, N. Ciochetto, J. Hong, and H. Shin, *Implementation of ground control system for autonomous multi-agents using qgroundcontrol*, in 2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS), 2019, pp. 24–30.

[5] A. Geiger, *Key findings about the online news landscape in america*, in Pew Research, 2019.

[6] F. Giones and A. Brem, *From toys to tools: The co-evolution of technological and entrepreneurial developments in the drone industry*, Business Horizons, 60 (2017), pp. 875–884. THE GENERATIVE POTENTIAL OF EMERGING TECHNOLOGY.

[7] R. Heydon and N. Hunn, *Bluetooth low energy*, CSR Presentation, Bluetooth SIG https://www. bluetooth. org/DocMan/handlers/DownloadDoc. ashx, (2012).

[8] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, *Micro air vehicle link (mavlink) in a nutshell: A survey*, IEEE Access, 7 (2019), pp. 87658–87680.

[9] T. McAdams, *Translational lift*, 2012.

[10] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, *The pixhawk open-source computer vision framework for mavs*, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXVIII-1/C22 (2011), pp. 13–18.

[11] F. M. Naini, O. Dousse, P. Thiran, and M. Vetterli, *Population size estimation using a few individuals as agents*, in 2011 IEEE International Symposium on Information Theory Proceedings, 2011, pp. 2499–2503.

[12] N. Ravi, R. Chitanvis, and M. El-Sharkawy, *Applications of drones using wireless sensor networks*, in 2019 IEEE National Aerospace and Electronics Conference (NAECON), 2019, pp. 513–518.

[13] E. G. Vattapparamban, *People counting and occupancy monitoring using wifi probe requests and unmanned aerial vehicles*, Master's thesis, FIU Electronic Theses and Dissertations, 2016.

[14] Y. Wu, D. Sun, P. Yi, and J. Tang, *Design and implementation of wireless mesh network testbed sjtu-mesh*, in 2010 International Conference on Internet Technology and Applications, 2010, pp. 1–4.

Drone-Based_Wireless_Communications_for_Disaster_Recovery

Final Audit Report 2021-06-14

Created:	2021-06-14
By:	Bioengineering Department (bioengineering@scu.edu)
Status:	Signed
Transaction ID:	CBJCHBCAABAAwUXjDtkWzm2x2l1qqoK67uM5cwZebAXH

"Drone-Based_Wireless_Communications_for_Disaster_Recovery" History

Document created by Bioengineering Department (bioengineering@scu.edu)
2021-06-14 - 5:29:41 PM GMT- IP address: 24.6.105.116

Document emailed to Nam Ling (nling@scu.edu) for signature
2021-06-14 - 5:30:04 PM GMT

Email viewed by Nam Ling (nling@scu.edu)
2021-06-14 - 6:47:01 PM GMT- IP address: 74.125.214.7

Document e-signed by Nam Ling (nling@scu.edu)
Signature Date: 2021-06-14 - 6:47:24 PM GMT - Time Source: server- IP address: 75.4.202.62

✓ Agreement completed.
2021-06-14 - 6:47:24 PM GMT