ST. MARY'S UNIVERSITY

School of Science, Engineering and Technology
Department of Engineering

UAV Communications Protocol Project

By

Jesus Gutierrez

Senior Design Project Presented to the Department of Engineering
In Partial Fulfillment of the Requirements
For the Degree of

Bachelor of Science
In
SOFTWARE ENGINEERING

San Antonio, Texas
May 2021

Supervising Advisor:
Dr. Ben Abbott
INSTRUCTOR OF ELECTRICAL ENGINEERING

ABSTRACT

The UAV Communications Protocol project provided the St. Mary's University Drone Lab with a communications protocol that will be used to communicate between the St. Mary's University's ground station and drone(s). This protocol makes use of a customer-specified radio to communicate between the ground station and the drone. In other words, the protocol connects the ground station and the drone using a radio that is onboard the drone and a radio that is in the ground station. In order to fulfill the customer requirements, however, and develop a proper communications protocol, this radio connection between the ground station and the drone needed to change. Therefore, the radio connection between the ground station and the drone was converted to a network connection. In doing so, the connection between the ground station and the drone was able to implement network-level security using the OpenSSL library. Unlike many commercially available communication protocols, this protocol utilizes the OpenSSL library to implement security and encrypt the communications between the ground station and the drone. All of this allows the drone and the ground station to send and receive encrypted messages within about 20 milliseconds. As per the customer requirements, this protocol also provides a means to monitor and track its' usage. The protocol keeps a record of its' usage and provides information about its' communication. This ability for self-monitoring was used to generate the tables that verify the performance of the protocol found further below in this report.

ACKNOWLEDGMENTS

The UAV communications Protocol Team would like to thank the professors who have made it their goal to ensure that we grow as students and as people. We are sincerely grateful for meeting all the wonderful staff at St. Mary's University and I am especially thankful for those who have helped in this project. Thank you to Dr. Abbott for providing his knowledge and skills to ensure that this team grew as students and as engineers. Thank you to Dr. Tezza, Nicole Webb, and Dr. Rezaie for the feedback and assistance throughout the project. Thank you also to the entire School of Science Engineering and Technology, and the St. Mary's University Drone Lab.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. PROBLEM STATEMENT

St. Mary's University has recently added a new addition to its' offerings to students. It has created a Drone Lab to further the education of students and to provide the resources necessary to learn about Drones. As part of the St. Mary's University Drone lab, Dr. Tezza and Dr. Rezaie have requested that a UAV Communications Protocol be developed for its' usage in the St. Mary's University Drone Lab. This protocol is to allow for communication between the St. Mary's University ground station, as well as its' drones. It must also provide secure communications, operate within a timely manner, run on specified hardware, and overall provide adequate communication and operation of the drones as per the established requirements for this project.

1.2. OBJECTIVES

The objective was to develop a UAV Communications Protocol that implemented security to allow for the encryption and decryption of messages that are sent between the ground station and the drone. The delivered artifacts are listed below and can be found in the references section. The artifacts include the UAV Communications Protocol's source code, documentation, requirements list, and other artifacts as needed per the requirements. Additionally, test cases and statistical data of the UAV Communications Protocol were included in the results section of this document.

List of Deliverables:

Documentation

Requirements List and Jira (Closed as this is classified)

https://jnseniordesign.atlassian.net/secure/RapidBoard.jspa?projectKey=SEN&rapidView=2&view=planning.nodetail&atlOrigin=eyJpIjoiZDMxMmNkNjU4ODUzNDA1YjhiOTA1YTE1YWY4MTQxNzAiLCJwIjoiaiJ9

Design Documents

Context Diagrams

Source Code (Classified Source Code) https://github.com/JG907/UavDroneProtocol.git

1.3. LITERATURE SEARCH

A highly popular existing UAV Communication Protocol on the market is the Micro Air Vehicle Link Protocol or MAVLink Protocol. This is often used by people ranging from drone enthusiasts to drone professionals. Yet, this protocol is often modified because of its' lack of security and can pose an overhead given its' size and the vast number of features. MAVLink, although having many potential uses, is a type of one size fits all protocol, and therefore may not be the best for all implementations. MAVLink is a very big and complex protocol, and although it is possible to add custom security, it might not be practical. Consider the research conducted by Auburn University and Appalachian State University in which students claimed that, "While this research uses a 16MHz processor that can perform encryption and decryption fairly quickly, the aforementioned extremely high overhead is unacceptable for an already resource-constrained system." (Butcher Page 3).

In short, although MAVLink is a widely used protocol, and although it likely has a lot of positives, it lacks built-in security, and any attempt to customize MAVLink might result in an undesirable Protocol.

For a detailed list of literature sources, please refer to the References Page.

1.4. PROBELEM CONTRAINTS, REQUIREMENTS, AND SPECIFICATIONS

Some obstacles and constraints this Project Team faced were as follows. When it came to the requirements, there were many which we did not know how we were going to fulfill. You see, our team was inexperienced when it came to networking. As such, we did not know where to start when it came to this project. The result was that the first semester for this project was mainly used for researching, as well as for the planning of the project. While the second, although having a slow start, slowly picked up speed. Another obstacle was that this project was being conducted in the middle of the COVID-19 Pandemic, and it was even more difficult to make progress. The need to adapt to a new learning environment made it harder to work as a team and to learn. Of course, we are grateful that we have the technology to work from home, but it was still something that we had to adapt to, and it was not something we had ever done before. Another obstacle involves the team itself. This project began with two individuals yet, due to unforeseen obstacles, it resulted in the team being modified to just one individual. When it comes to the requirements, we originally created a Gantt Chart and tracked the requirements in various applications until settling for Jira. The UAV communications protocol was developed on a Linux Operating System, and its' source code was managed on GitHub. The only thing we had familiarity with was GitHub and the Gantt Chart.

Although this project had a slow start, and many knowledge gaps had to be filled, seeing these obstacles as opportunities to grow as students rather than obstacles made it easier to tackle.

1.5. PROPOSED SOLUTION

To meet the customer requirements for making the UAV Communications Protocol for the St. Mary's University Drone Lab, the following was done. During the first part of the senior design class, we (the project team) regularly met with the customers and project sponsor (Dr. Abbott) to be able to come up with an agreed upon design. The second part of the senior design was then primarily focused on implementing this agreed upon design as per the requirements. The design that was agreed upon is shown below in Figure 1: A high-level overview (Level 0).

Table 1: Division of Labor

Member	*Task*
Jesus Gutierrez (SE)	Research, Documentation, and Development
Nevin Tran (SE)	Research and Documentation

2. SUMMARY OF ENGINEERING METHOD

This project follows common Software Engineering practices for developing software. We began by Eliciting Requirements and refining those Requirements. From there, we began designing and seeking customer approval and satisfaction. After that, we began development and testing. Lastly, we focused on testing and then on delivery. To properly manage the project, we used tools such as Jira to manage the requirements, Project Libre to manage scheduling, and GitHub for source code management.

3. ITERATIVE DESIGN SUMMARY

This project was developed in an iterative manner following an Agile life cycle. We held weekly meetings with the customer and the project sponsor to advance the project as a whole. Requirements were elicited at the beginning and then were refined. From these refined Requirements, designs were made and also refined. These designs were then the guide to developing the communications protocol. The designs are also the guide for testing the protocol to ensure that it works as intended by the requirements.

4. OVERVIEW OF SYSTEM

The UAV communications protocol as a whole can be pictured as is as shown in

Figure 1: A high-level overview (Level 0). This protocol is intended to be used to communicate from the ground station to the drone. This communication will also support the standard TLS security to protect the communications being done. Please note that the usage of the TLS library was chosen because it is an industry-standard among many projects and is known to provide robust security that is hard to tamper with.

This protocol that was described above was designed in that fashion so as to satisfy the customer's needs and requirements. The documentation of these requirements can be found in the requirements link listed below in the references section.

4.1. SYSTEM ARCHITECTURE

The customer requested a UAV communications Protocol to be used in the St. Mary's University Drone Lab. This protocol addresses issues associated with some existing protocols on the market. As discussed in the literature section, one of the most popular commercially available communications protocols is MAVLink. A drawback, however, with MAVLink is that it does not provide all of the features that the customer requested. It is, however, possible to modify MAVLink and add the missing features, but it would create an undesirable overhead that defeats the purpose of trying to use MAVLink. To better align with the customer needs, the UAV communications protocol that this team developed provides a protocol based on the customer requirements while avoiding the unnecessary drawbacks of other protocols. The design and architecture of the developed protocol are shown in the figures below.

4.1.1. Context Diagram – A high-level overview (Level 0)

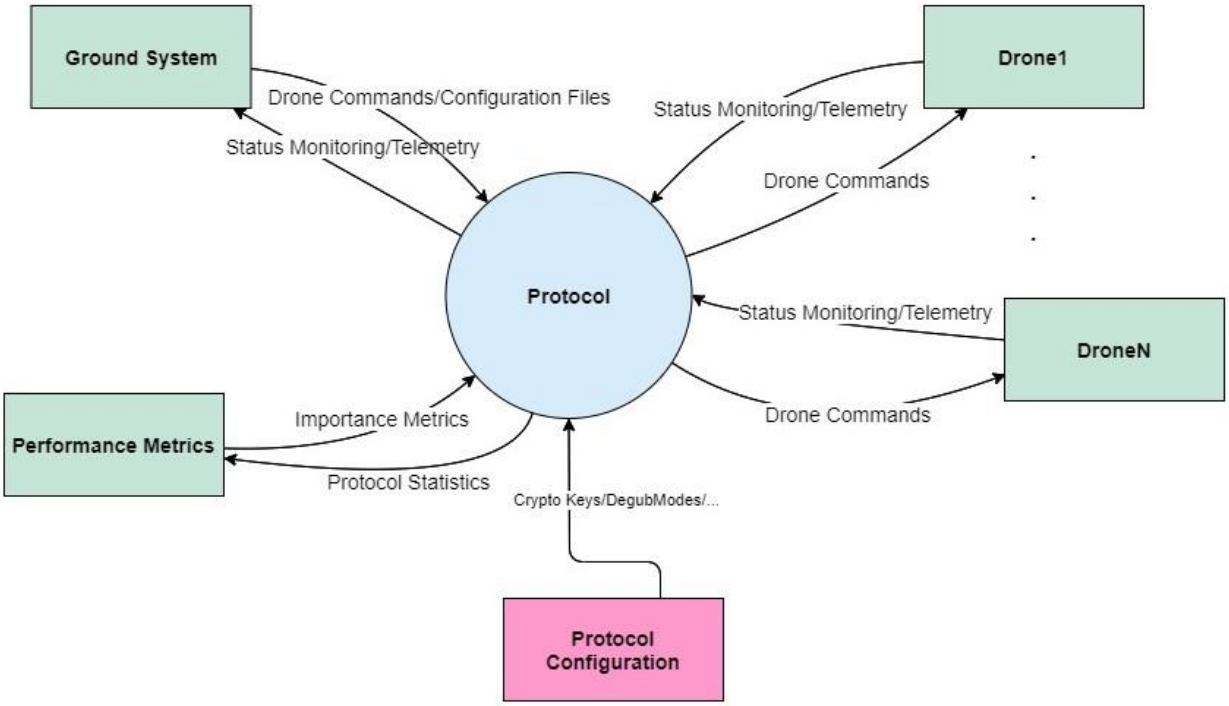

Figure 1: A high-level overview (Level 0)

This Context Diagram provides a high-level overview of the entire UAV communications protocol that was developed. It shows how the Ground System intends to communicate with the drones it will fly. It also shows how the protocol will maintain its' statistics and be customized through the protocol configurations.

4.2 Context Diagram - Ground Station (Level 1)

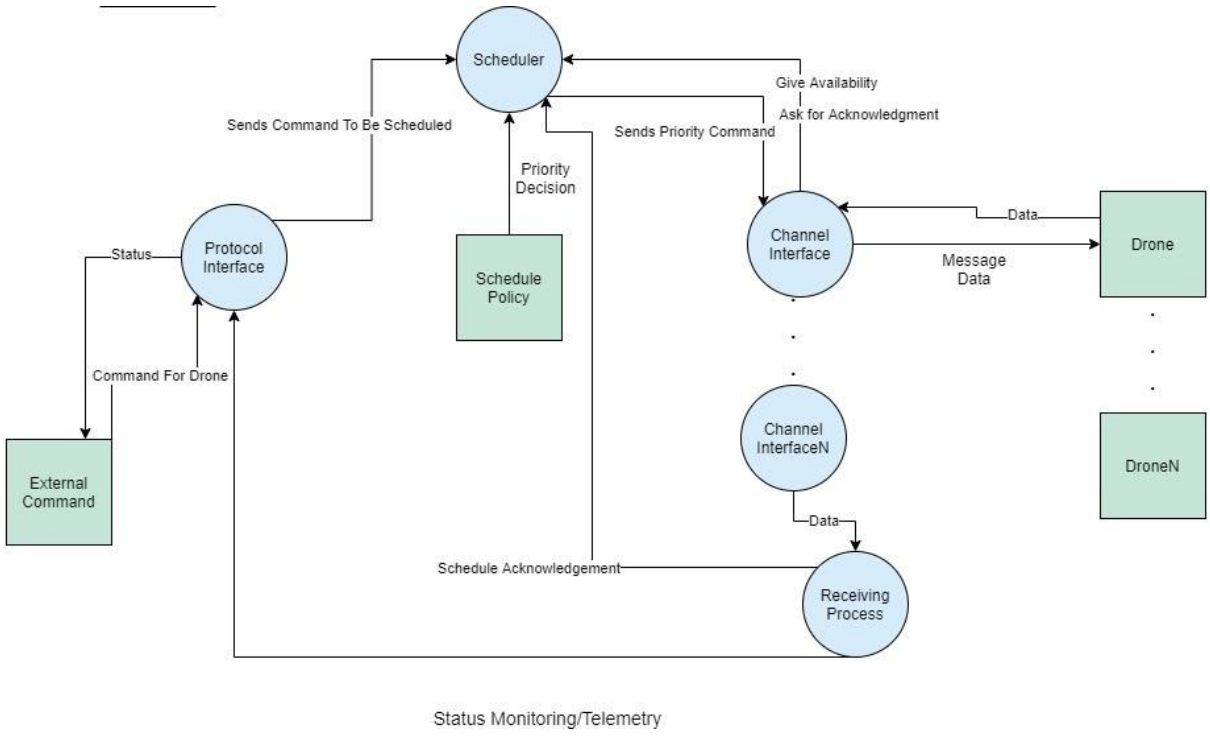

Figure 2: Ground Station (Level 1)

This Context Diagram represents how the Ground Station will communicate with the Drones. This design implements the possibility to have more than one drone and is represented by the "DroneN" box. Although it was not a requirement to implement the functionality to support multiple drones, the customer requested it be included in the designs, and that is why it is presented in this Figure.

4.3. Context Diagram – Drone (Level 1)

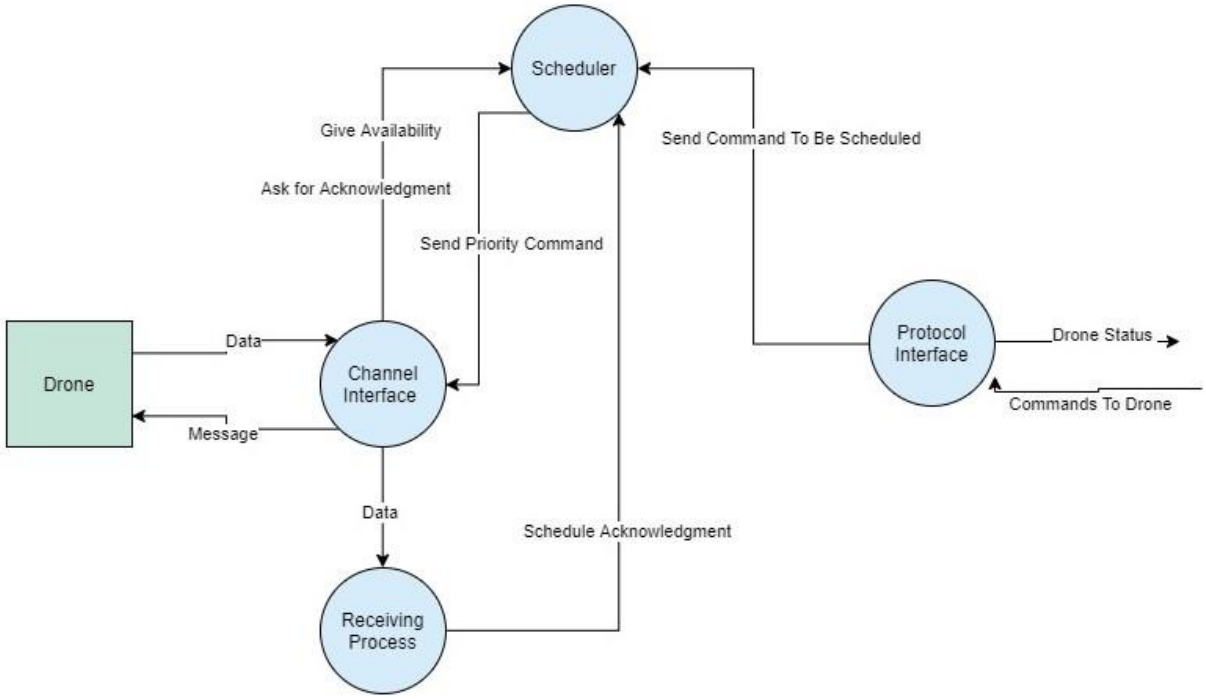

Figure 3: Drone (Level 1)

This Context Diagram demonstrates how the drone responds to the Ground Station in a secure and scheduled way. Further details on the security are found below.

4.4. Context Diagram – Inside Channel Interface (Level 2)

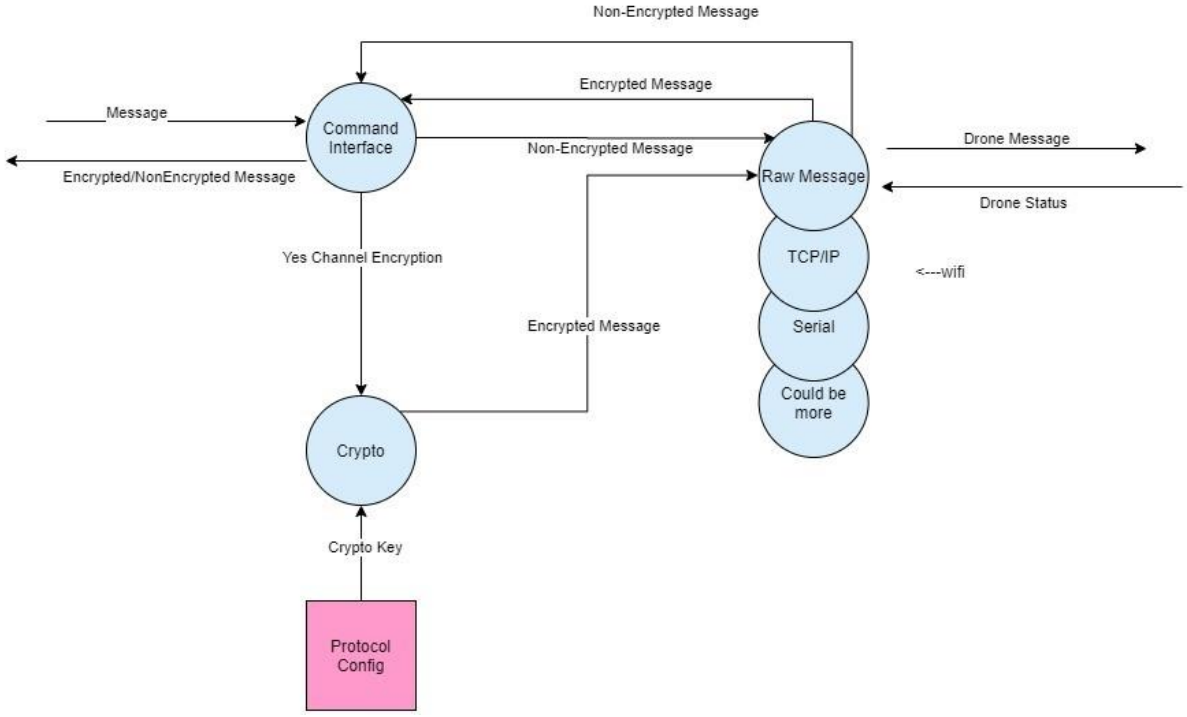

Figure 4: Inside Channel Interface (Level 2)

This Context Diagram shows the inner workings of the Channel Interface bubble shown in the previous sections. This Figure shown here is a representation of how the messages in the UAV communications protocol are encrypted. The actual security being used is TLS. TLS was chosen because it is used in various projects across industry and is known to be reliable and robust.

5. UNEXPECTED PROBLEMS AND SOLUTIONS

There were some unfortunate events that took place throughout this project. For one, this project was undertaken during the COVID-19 Pandemic, and as such, it is inherently a challenge. Also, this project dove into a lot of things that the project team had little to no experience with. This project was a learning experience as much as it was an opportunity to demonstrate what we had learned during our time at St. Mary's University. Yet, for unexpected circumstances, this project ended up losing the other member, and as such, the project ended up with only one member. These problems had to be solved in the order in which they came. Just as they unexpectedly popped up, they needed to be dealt with accordingly. Otherwise, the entire project could have lagged behind, and this project would have never gotten anywhere.

6. RESULTS

This section is meant to communicate the outcome of the development of the UAV communications protocol and how it is used to send secure messages between the drone and the ground station, and vice versa. This section also contains a series of Figures aimed to show the outcome of this project. The figures below will have data that pertains to the ground station and the drone. The primary hardware used to develop and test for this project was a laptop to represent the ground station, and a Raspberry Pi 3 to represent the drone. The exact specification of each device is listed here:

6.1 SPECIFICATIONS OF HARDWARE USED

The device used to represent the Ground Station:

- Device name: MSI
- Processor: Intel® Core™ i7-10750H CPU @ 2.6GHz 2.59 GHz
- Installed RAM: 16.0GB RAM
- System Type: 64-bit OS, x64-based processor

The device used to represent the Drone:

- Device name: Raspberry Pi 3 B v1.2
- Processor: Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- Installed RAM: 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

6.2 SINGLE MESSAGE COMMUNICATIONS

In this section, a series of figures are presented. Data gathered from the communication between the Ground Station and the Drone are shown below. Each figure will show data either from one device, (from the device specifications listed above), or from using both devices. In the case where only one device is listed, the single device acted as both the ground station and the drone. This was done to simulate the protocol on one device and get an idea of the protocol's performance. In the case where there isn't one device listed, then the figure was generated using both devices, specifically using the Laptop to act as the Ground Station, and the raspberry Pi 3 to act as the Drone.

The UAV communications protocol automatically records the data that is seen in the Figures below by storing it in a comma-separated-values (CSV) file. Two files pertain to the ground station, and two files pertain to the drone. Together, these files can then be further processed to produce graphs, tables, etc., and gain a deeper insight into what is going on with the protocol. This functionality ultimately allows the drone lab to process the data as they may see fit according to their needs. This also fulfills the monitorability requirement that was requested by the customer and can be seen in the requirements link in the references section.

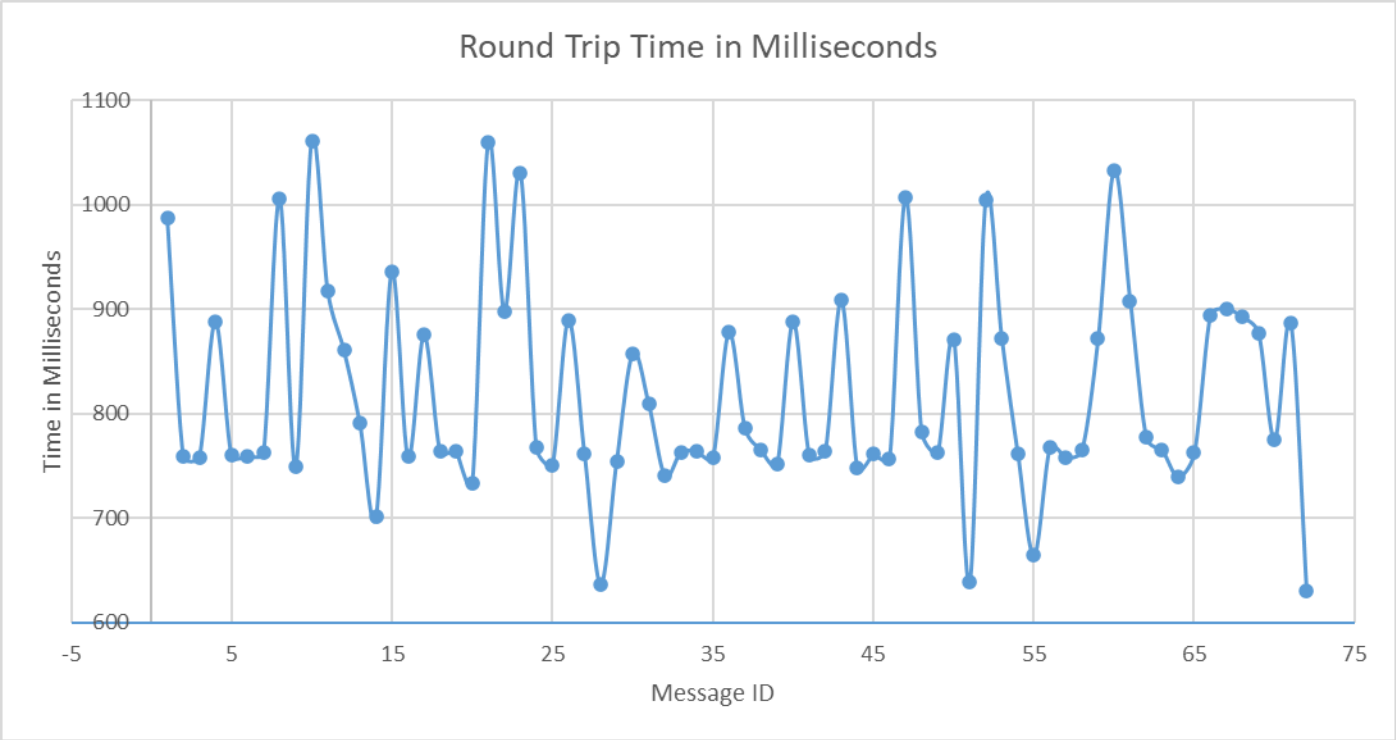

Figure 5: Round-Trip Time for single message communications

Figure 5 shows the round-trip time for sending and receiving a single message between the Ground Station and the Drone. The hardware used to obtain this data is the Laptop and the Raspberry Pi 3. This data is useful because it allows one to see the worst-case and best-case scenarios for a message's round-trip time. Round trip time is defined as the total time it takes for a message sent by the ground station to come back from the drone. This time is useful to track as it provides insight as to how fast the protocol is. It can be seen that the time never exceeds 1100 milliseconds. This should not be happening as the time it takes to send and receive messages should be faster as seen below. The reason this is happening is due to messages being buffered and not being flushed. The solution to this is already underway. However this graph nonetheless

is useful as it proves the monitorability requirement of the protocol is fulfilled, and that the messages get queued up properly as given by the shape of the graph.

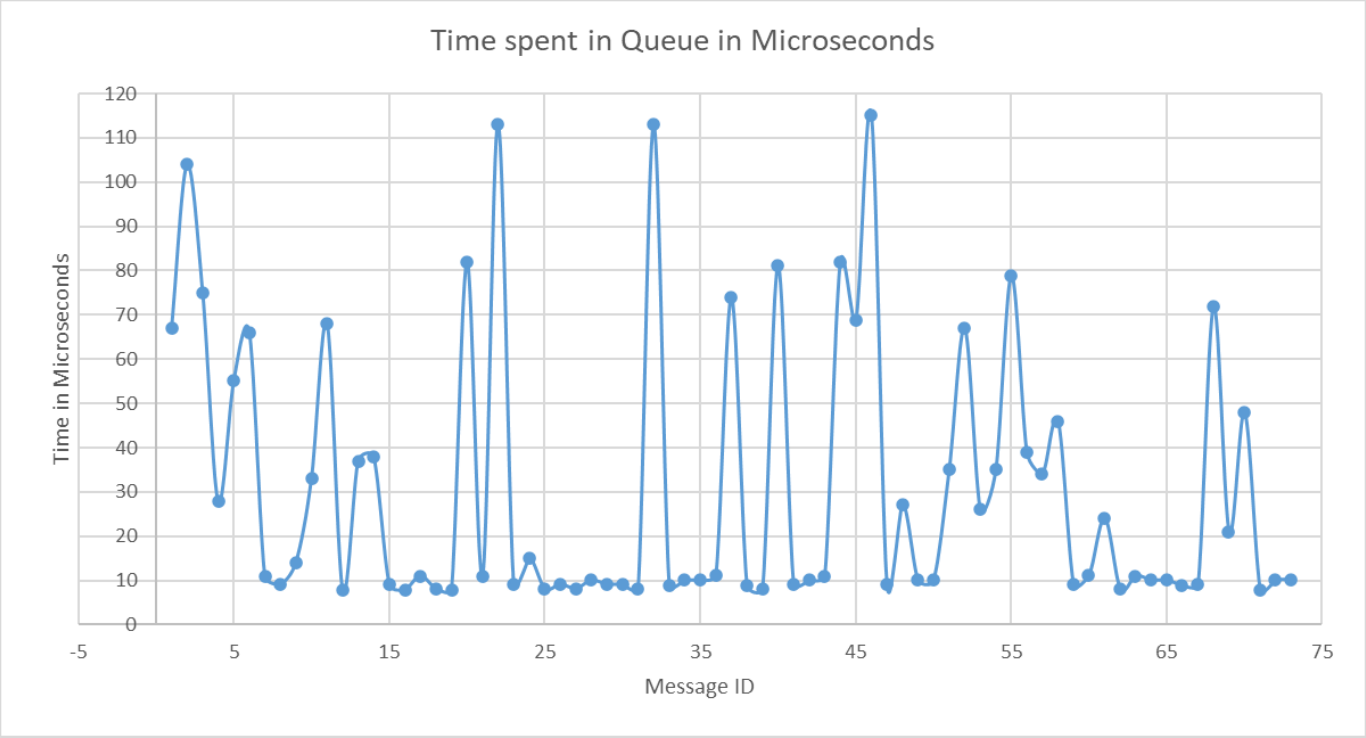

Figure 6: Time spent in queue for single message communications

This graph here goes with Figure 5 above. Figure 5 shows the round-trip time for a message being sent and received by the ground station to the drone. In this graph, we see the time it takes for the message to leave the ground station before heading off to the drone. Specifically, before the ground station's message gets transmitted to the drone, the message is placed in a message queue. The communications protocol uses a message queue to collect the messages that will be sent, then sends them off one by one according to their priority. It is important to know the amount of time a given message spends in the message queue to track when high-priority messages get sent. The message queue will send the highest priority messages

first before sending off the lower priority messages. This, however, is very fast if there is only one message being sent at a time. The reason for this is because when sending individual messages, the queue is occupied by only one message at any given moment. Therefore, when this happens, the queue only has one message to send and, due to it being the only message, it has the highest priority. This, in turn, means the message is sent off almost immediately after being placed in the queue. For this reason, the graph above shows that there was never a message that spent more than 120 microseconds in the queue before being transmitted.

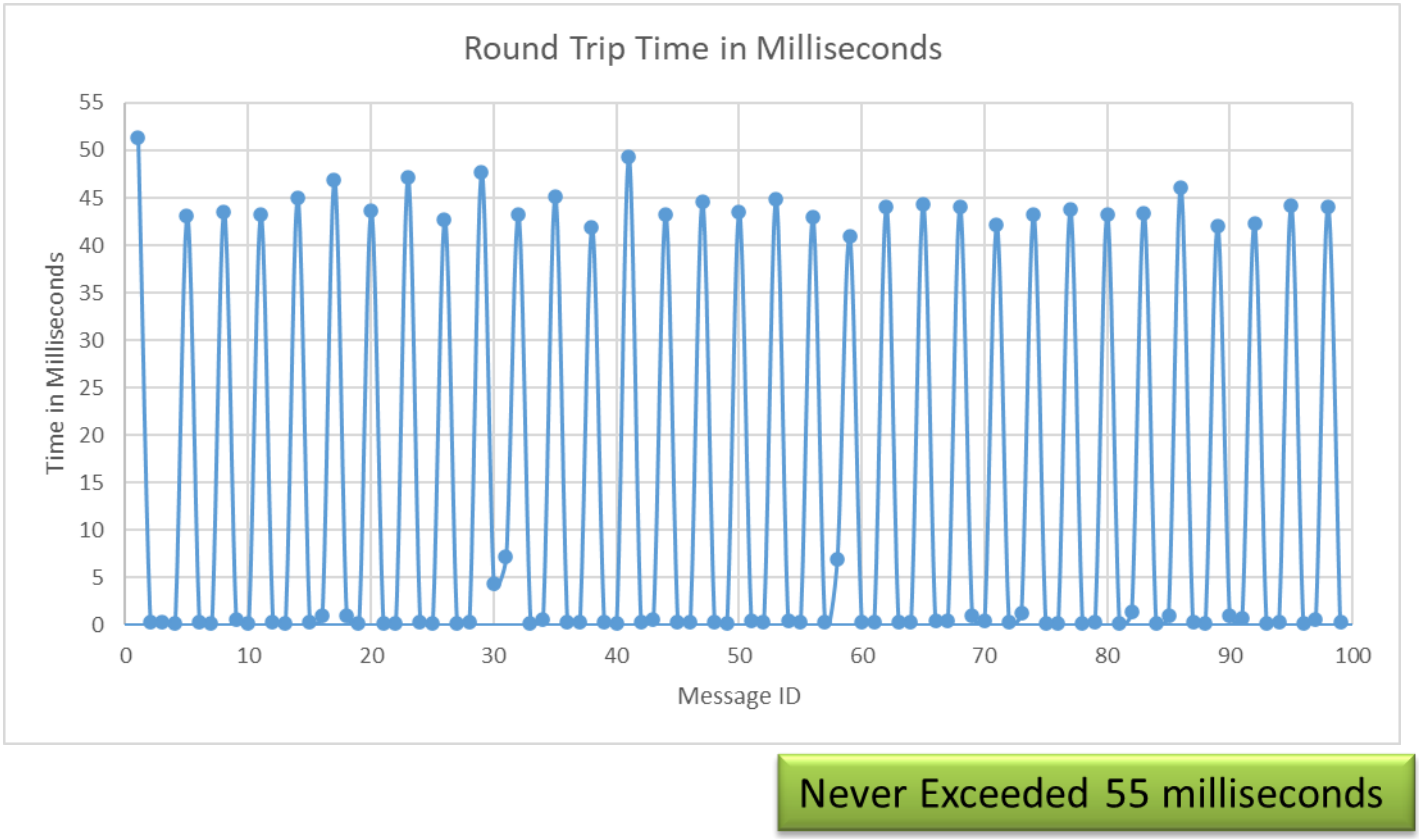

Figure 7: Round-Trip Time for single message communications - Laptop

Figure 7 shows the round-trip time for each message that was sent between the ground station and the drone. In this case, however, a single machine was used to act as the ground station and the drone. Specifically, the laptop mentioned above in the "Specifications of Hardware used for Results" was used to act as both the ground station and the drone. It can be seen from this graph that the round-trip time was under 55-millisecond. It is important to note that this time includes the encryption and decryption of the messages using the OpenSSL library.

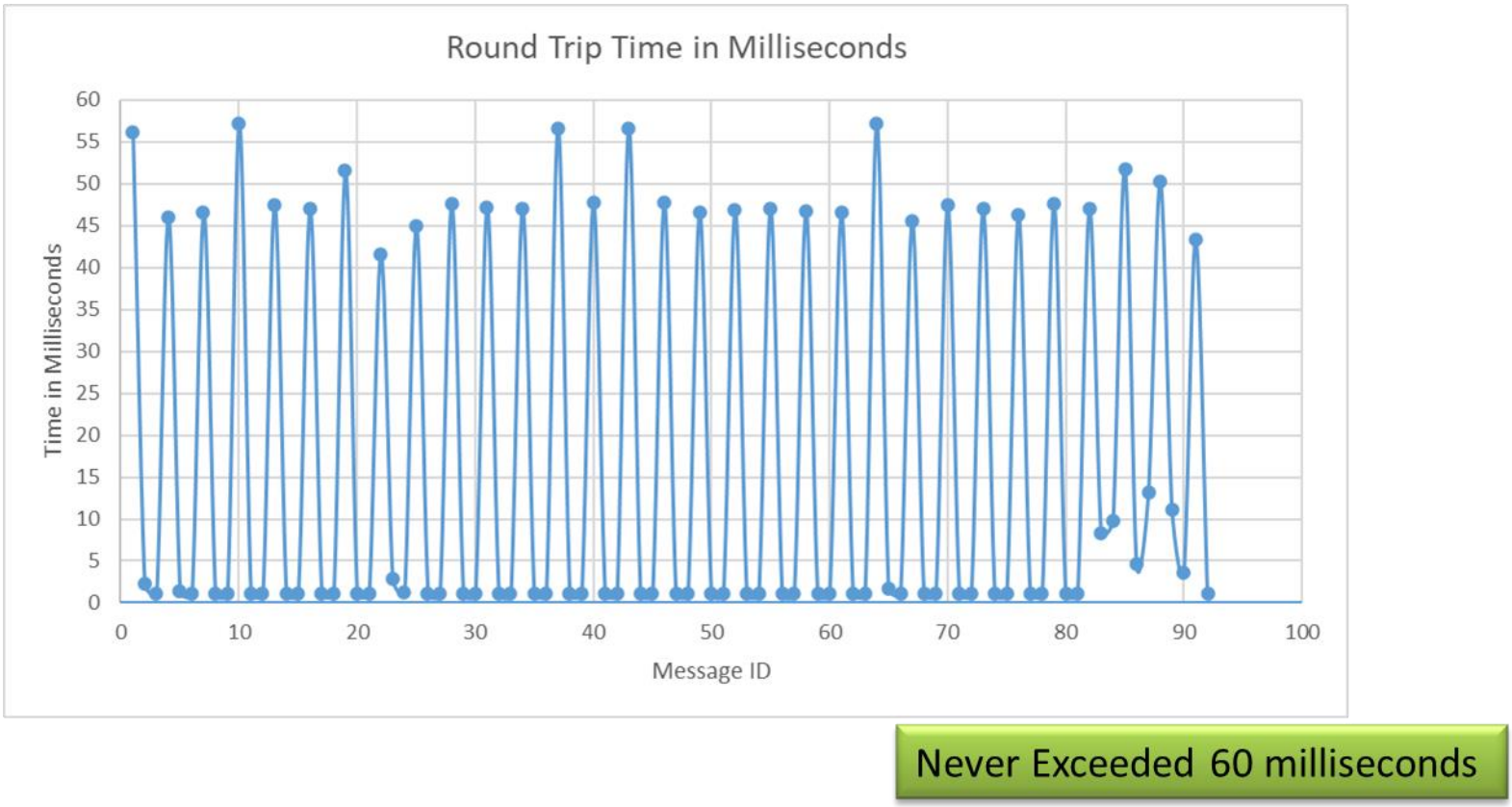

Figure 8: Round-Trip Time for single message communications – Raspberry Pi 3

In this figure, the ground station and the drone were simulated on the Raspberry Pi 3. The Raspberry Pi 3 communicated with itself and sent messages to itself to be able to generate this

graph seen here. It can be seen how the graph indicates that the round-trip time, or the time it takes a message to be sent to the drone and received back to the ground, never exceeded 60 milliseconds. It is important to note that this time includes the encryption and decryption of the messages using the OpenSSL library.

After seeing both Figures, Figure 7 and Figure 8, it can bee seen that they are both about the same, indicating that the protocol functions as expected. We can also gain a little bit of insight into the overhead required on each device to run the protocol, yet it appears that there isn't much of a difference.

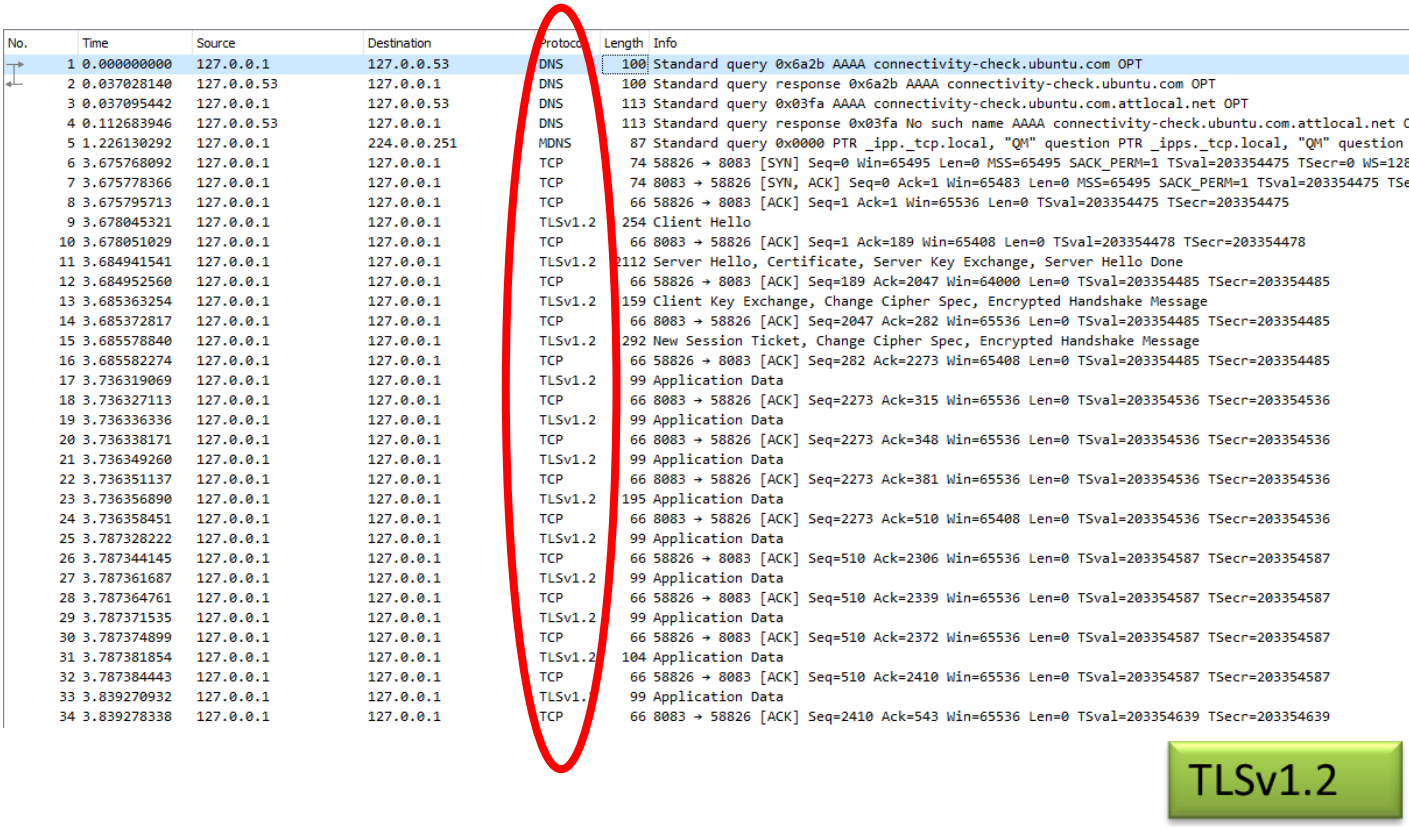

Figure 9: Encrypted Communications - Wireshark

In this figure, it can be seen under the circled “Protocol” column that there are instances of TLSv1.2, which is also known as Transport Layer Security, and is what gives the communications protocol its’ ability to encrypt and decrypt messages. Here it can be seen using a piece of software called Wireshark that the communications protocol does indeed encrypt and decrypt the messages being sent and received as required by the customer.

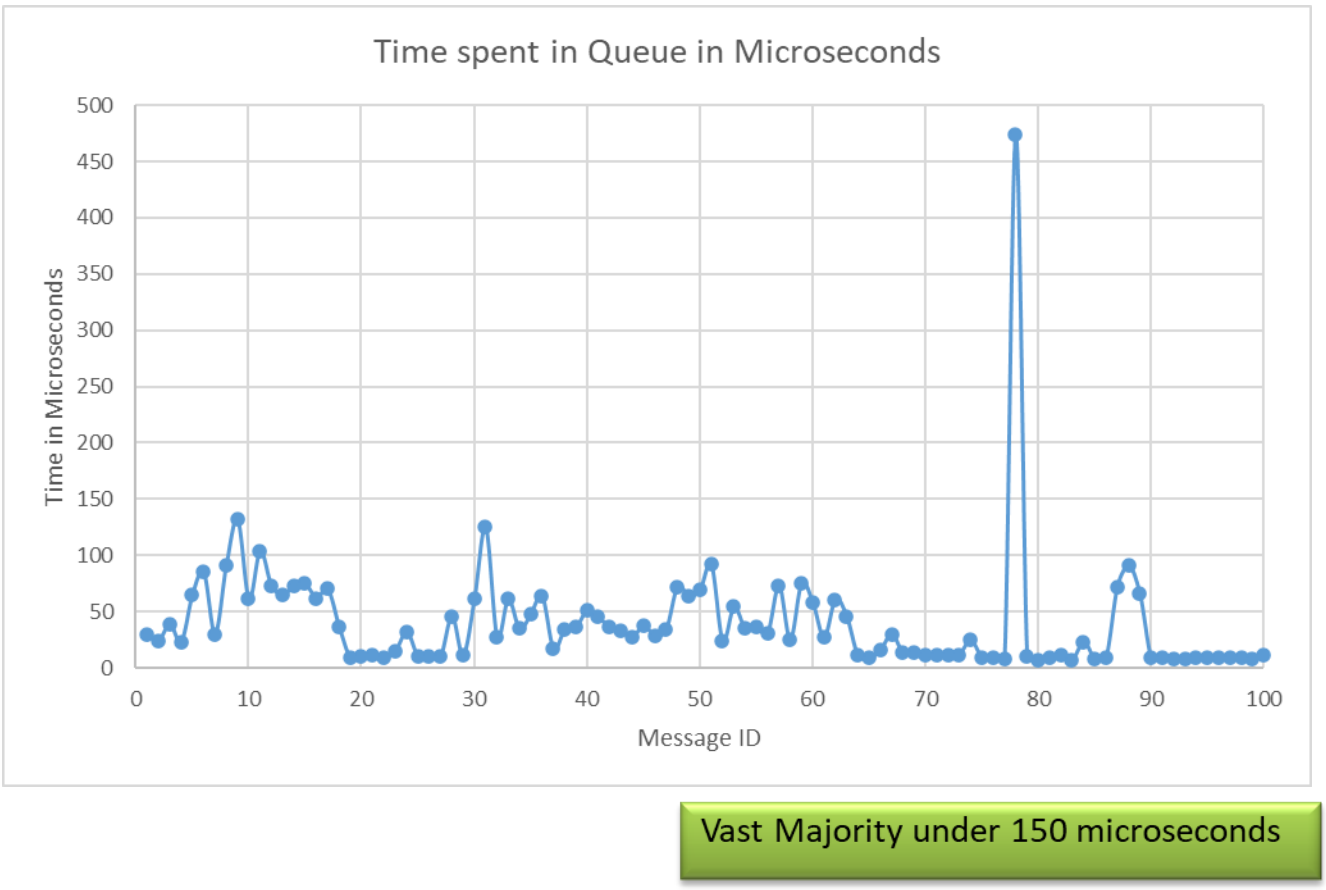

Figure 10: Time spent in queue for single message communications - Laptop

This Figure shows the time that each message spent in the queue before being sent off to the drone. In this case, it is important to remember that the ground station and the drone are being simulated to run on the same machine, specifically in this case, the Laptop mentioned

above in the hardware specifications. Yet, it is also important to note that the laptop is being used to host a guest operating system, which also happens to be the operating system that is running the communication protocol. This can be a reason for the random spike in time spent in queue around message 80. This, however, was only for one message. In fact, all other messages were under 150 microseconds, therefore it was assumed to be caused by the operating system which was being run within another operating system and therefore unable to utilize its resources as well as in the case of the Raspberry Pi 3 in Figure 13 below.

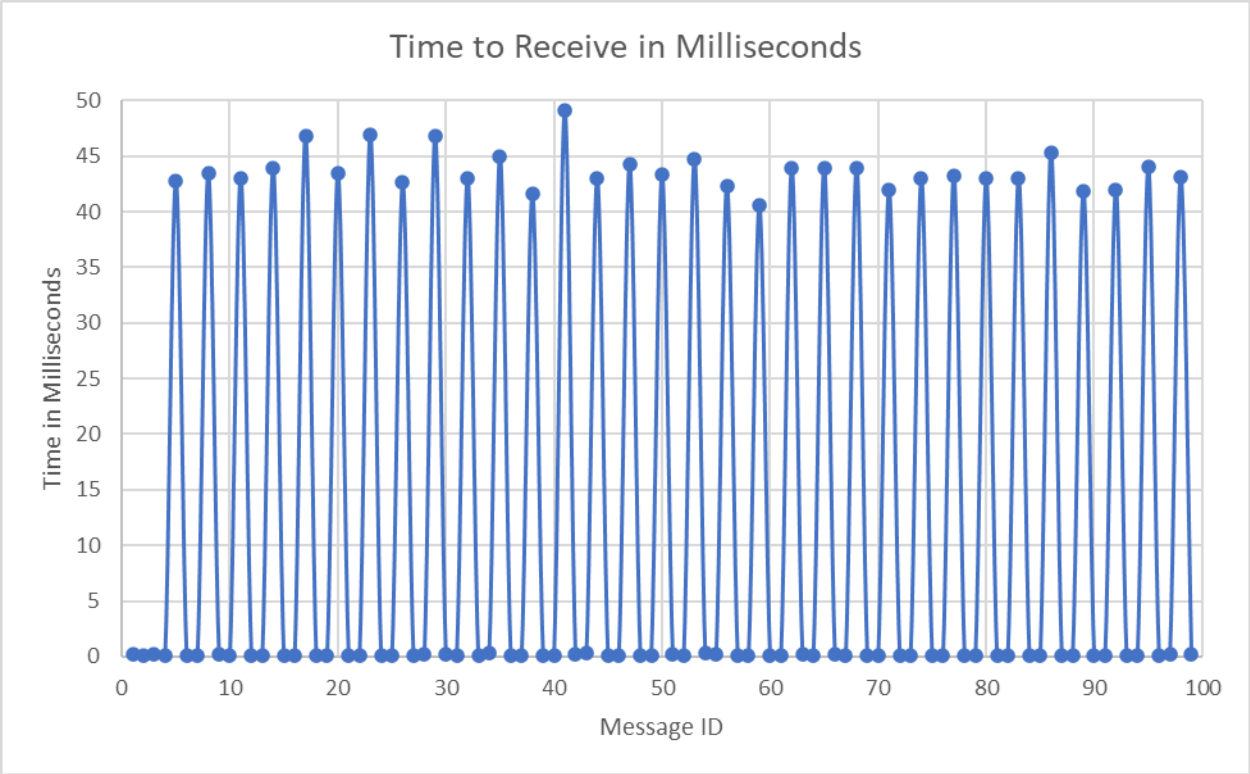

Figure 11: Time it takes Drone to receive a message - Laptop

This Figure was generated using the laptop to send messages between the ground station and the drone (the laptop talking to itself). This graph helps gain a deeper understanding of the

overhead that is a part of the transmission of messages when the ground station and the drone (with two devices) are communicating with each other. Despite this being data that comes from only one device, it can be used in conjunction with the corresponding graph on the Raspberry Pi 3's single message data to provide insight into the overhead for communications between the two devices when acting as the ground station and the drone.

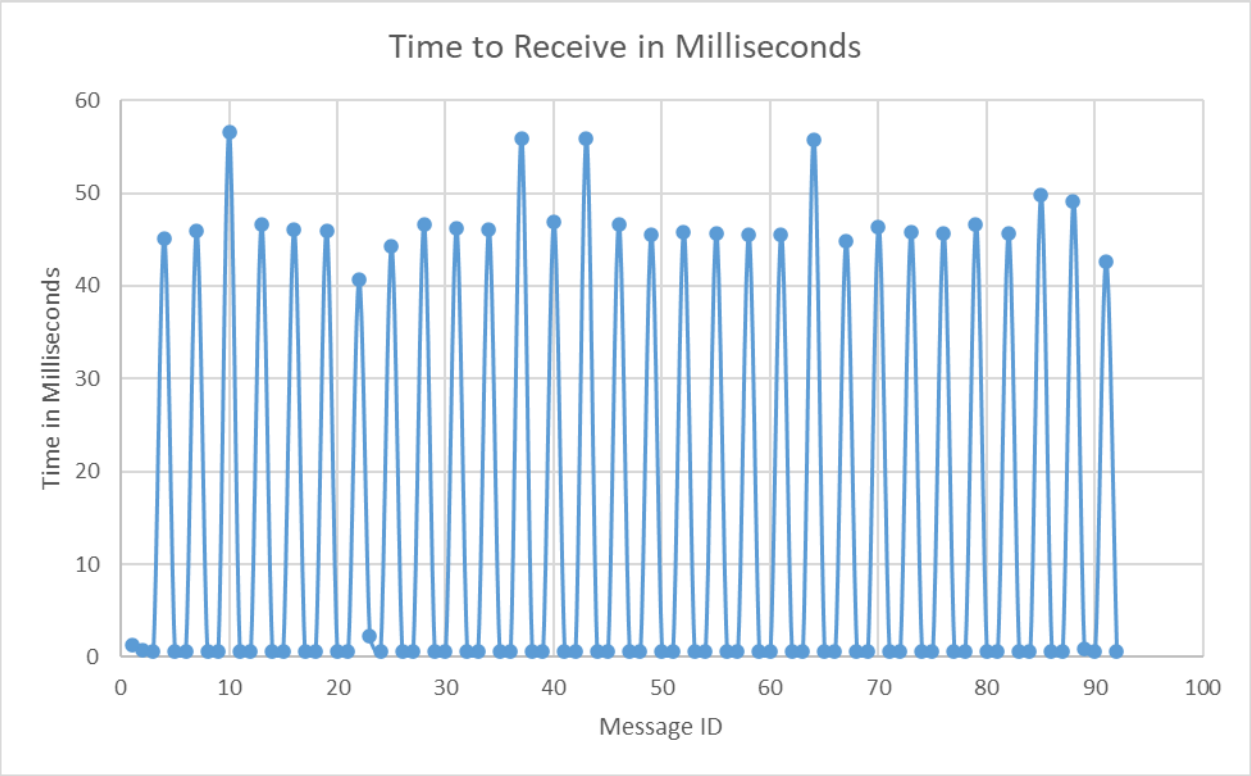

Figure 12: Time it takes Drone to receive a message – Raspberry Pi 3

This graph is the graph that was mentioned in Figure 11 above. This graph shows the time it takes for a message to travel from being sent from the ground station to the drone. It is important to note that the raspberry pi 3 was used to collect this data, and it was used to act as

both the ground station and the drone. Figure 11 and Figure 12 can be used to gain insight into the overhead required sending messages from the ground station to the drone. Using this insight, one can then get a better understanding of the latency and overhead that occurs when a message is sent from the laptop (ground station) to the raspberry pi 3 (drone). It can also be used to provide insight on the real ground station and drones in the St. Mary's University drone lab.

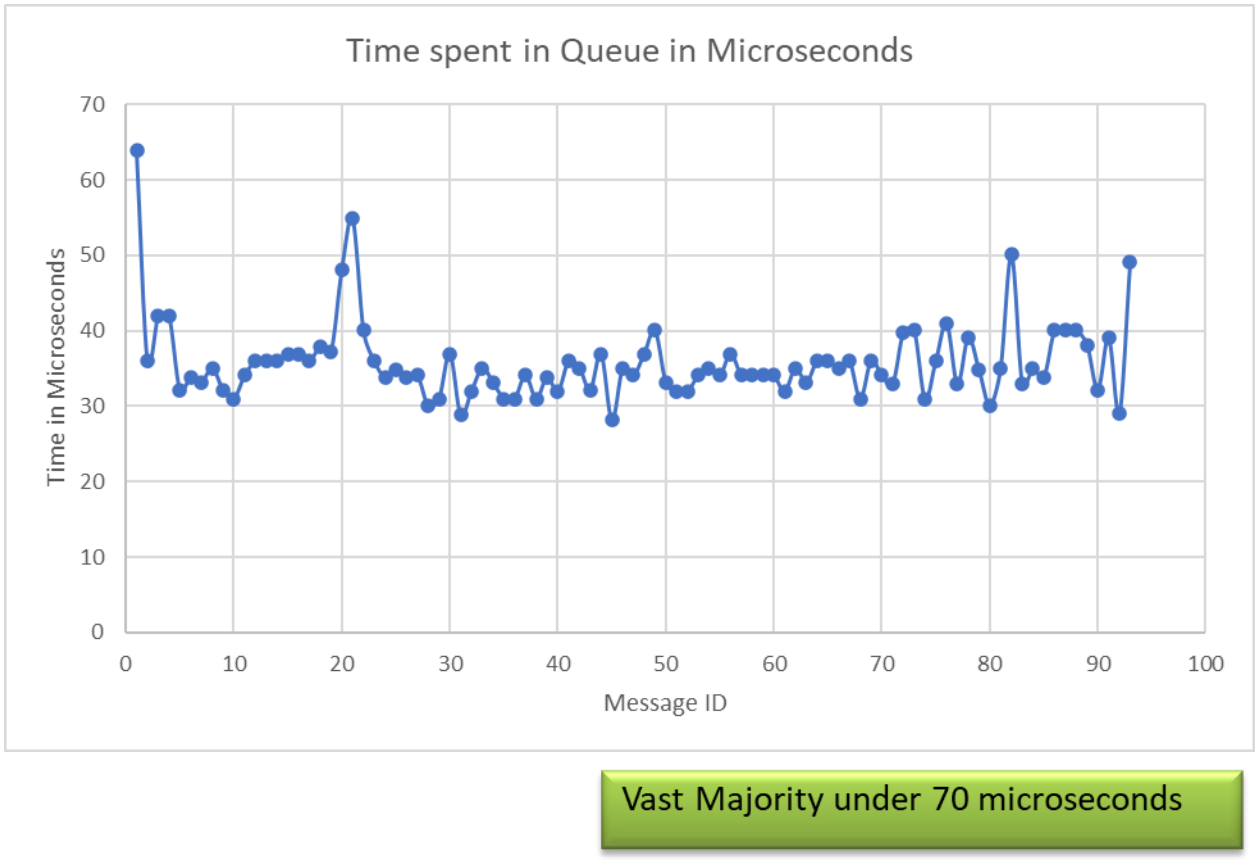

Figure 13: Time spent in queue for single message communications – Raspberry Pi 3

Figure 13 shows the time a single message spends in the queue before being sent off to the drone. In this Figure, the ground station and the drone are being simulated on the raspberry pi 3. The communications described in figure 13 are between the raspberry pi 3 and itself, not with the

laptop or any other external computer. This is just like in Figure 10 except, in that case, the ground station and the drone were running solely on the laptop, while here it's on the raspberry pi 3. It can be seen that the time spent in the queue was much lower on the raspberry pi 3 than that of the laptop. It was concluded that the raspberry pi 3's performance was superior because the raspberry pi 3 was using its' own operating system and not a virtualization like in the case of the laptop.

6.3 MULTIPLE MESSAGE COMMUNICATIONS

In this section, a series of figures are presented. Data gathered from the communication between the Ground Station and the Drone are shown below. Each figure will show data either from one device, (from the device specifications listed above), or from using both devices. In the case where only one device is listed, the single device acted as both the ground station and the drone. This was done to simulate the protocol on one device and get an idea of the protocol's performance. In the case where there isn't one device listed, then the figure was generated using both devices, specifically using the Laptop to act as the Ground Station, and the raspberry Pi 3 to act as the Drone.

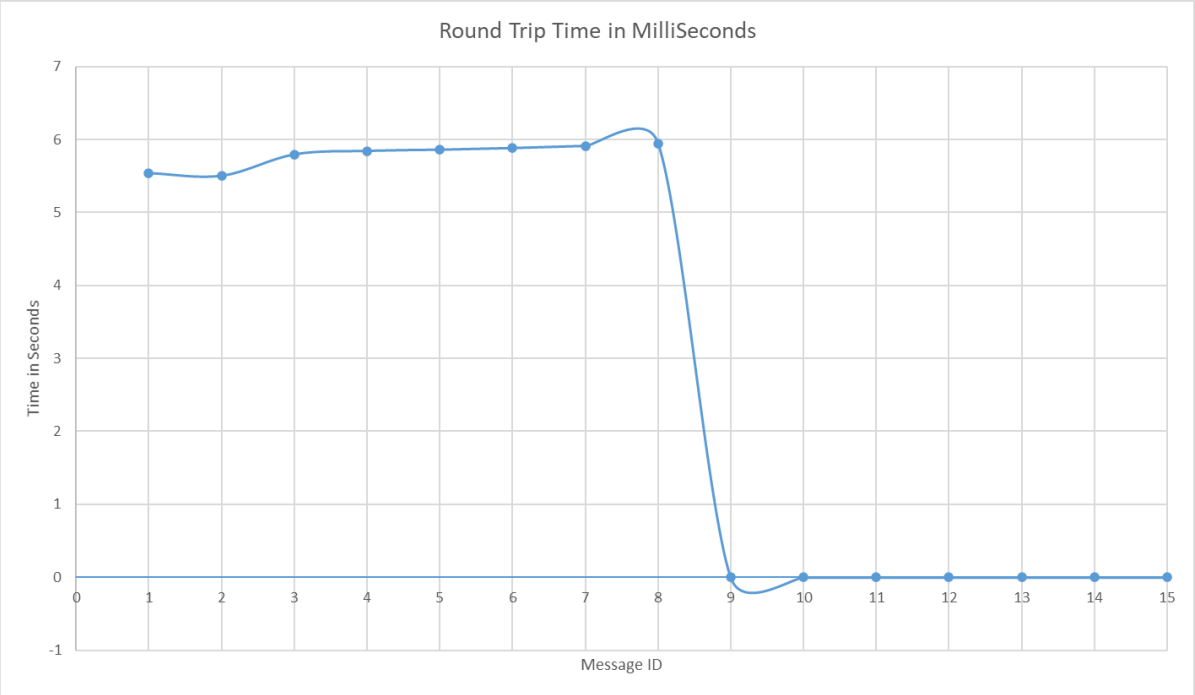

Figure 14: Round-Trip Time for multiple message communications

This was mentioned in the presentation about an error that was occurring when utilizing the radios to transmit the messages from one machine to the other, there is a fix in the works for this and is why this graph appears to be incorrect. It can be seen however that this error appears to only be present when communicating between two machines such as the Laptop and the Raspberry Pi 3, and not when the protocol is run on a single machine. This indicates that the protocol is functioning and that the only issue is the actual connection between the two machines that requires extra attention.

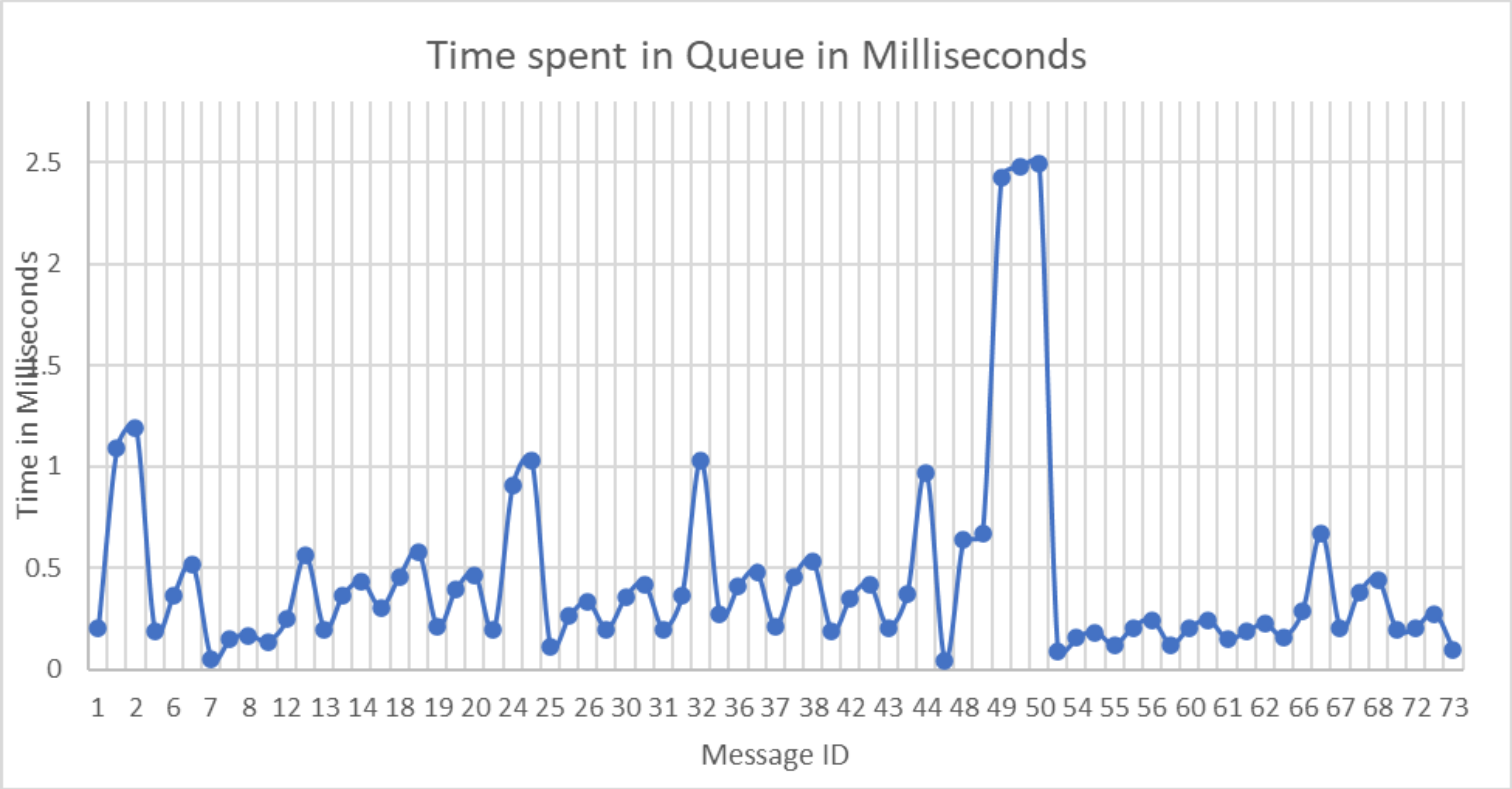

Figure 16: Time spent in queue for multiple message communications

Figure 16 shows the time spent in queue for each message when sending multiple messages from the laptop to the raspberry pi 3. Please note that in this case there were multiple messages being sent. It can be seen how the time each message spent in the queue increased as more messages were being sent. This shows evidence of the message queue at work making sure that the highest priority message gets sent prior to any lower priority messages. This is to ensure that the messages are received in a timely manner.

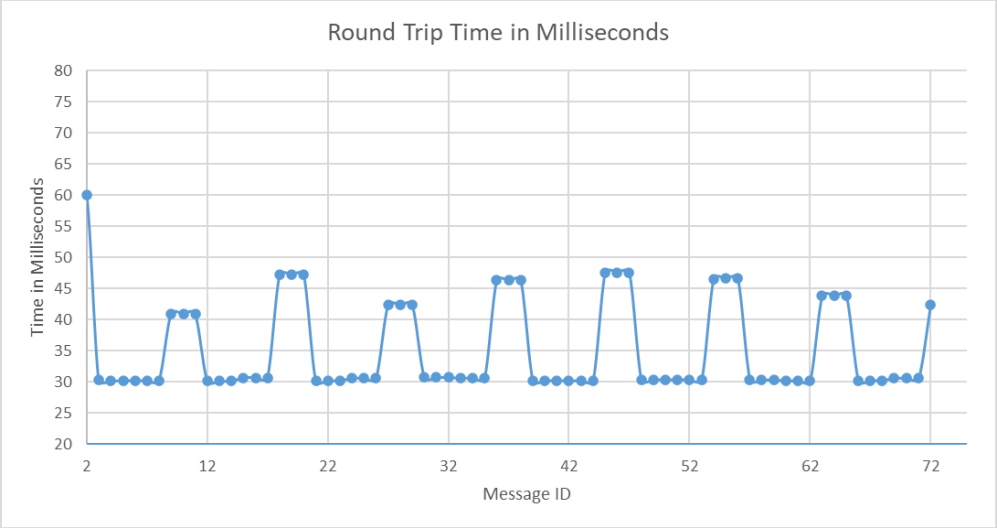

Figure 17: Round-Trip Time for multi-message communications – Laptop

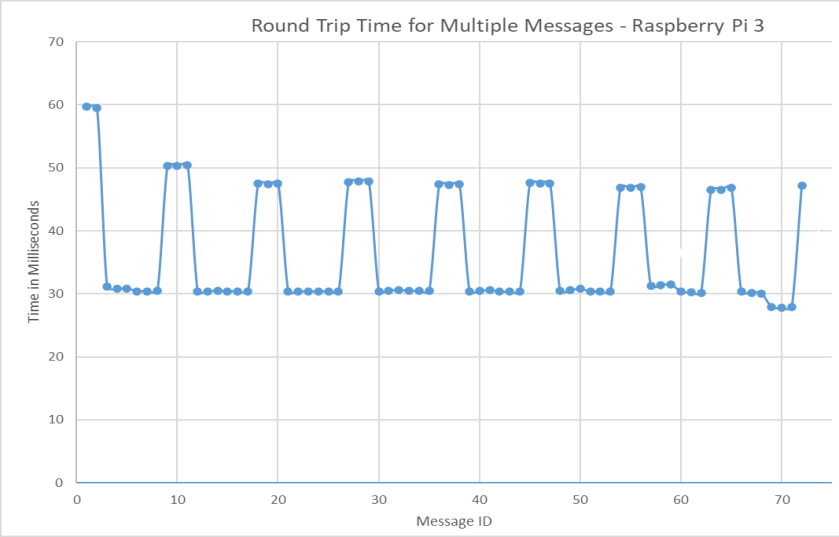

Figure 18: Round-Trip Time for multi-message communications – Raspberry Pi 3

Figure 17, shows the round-trip time for each message when sending messages from the ground station to the drone as simulated on the laptop, while Figure 18 shows the round-trip time for sending from the ground station to the drone as simulated on the raspberry pi. Please note that in this case there were multiple messages being sent. It can be seen in each figure that the round-

trip time would increase, then decrease, and repeat. This shows evidence of the messages being queued so that the protocol sends the highest priority message before sending any lower priority messages. For this reason, there is a fluctuation in the round trip time each message took, yet this is normal and expected.

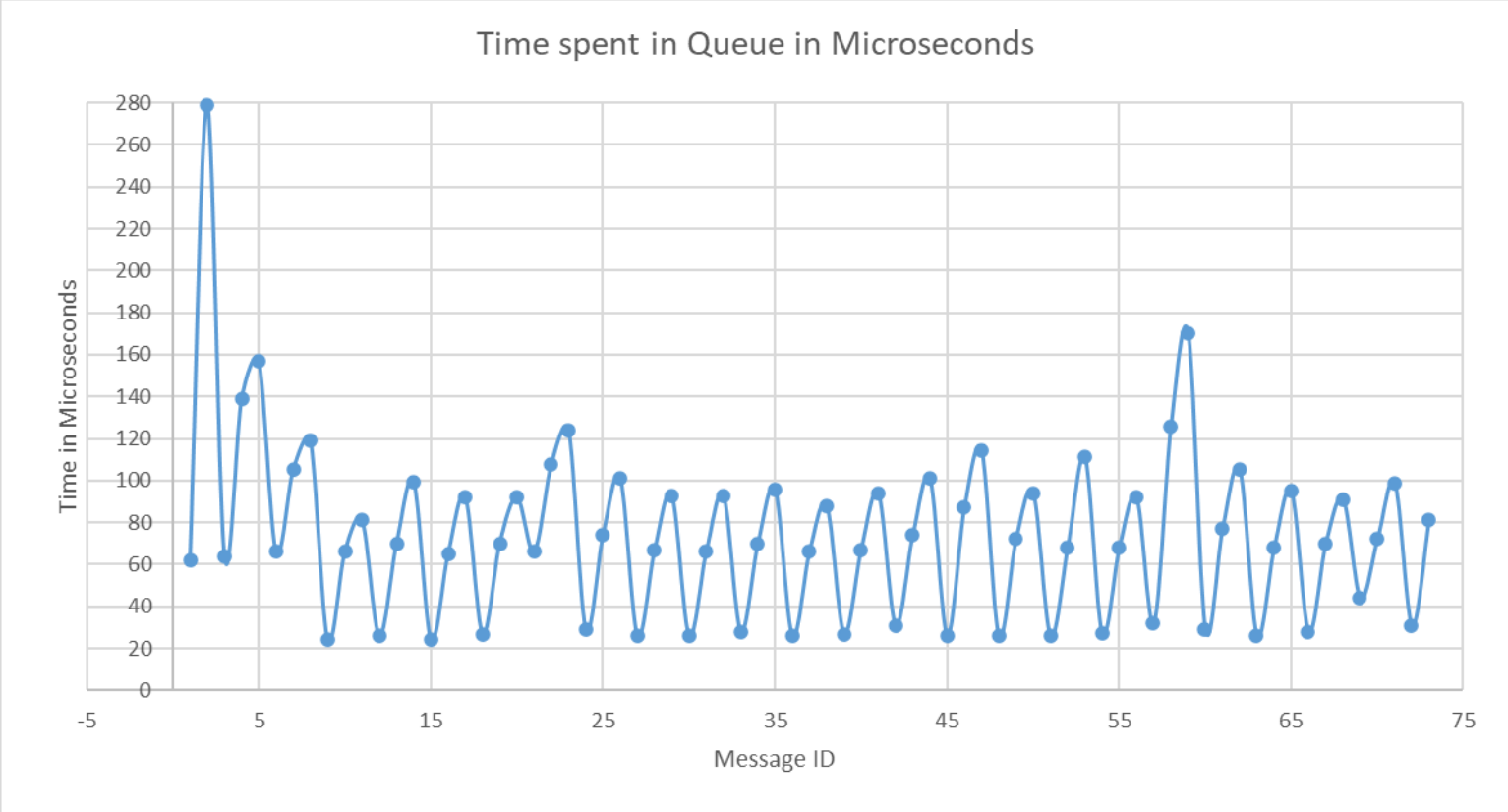

Figure 19: Time spent in queue for multiple message communications- Laptop

In Figure 19, we have the time spent in the queue for each message that was sent from the ground station to the drone. It should be noted that this is being simulated on the Laptop. One can see how the messages are being placed in the queue as more of them come in. When one is sending multiple messages from the ground station to the drone, there are too many wanting to

be sent at once, so to send them off one by one, they first need to be first placed in a wait queue and then sent off based on their priority. One can see evidence of that happening here by the shape of the graph. As more messages come in, the time spent in the queue increases. One will notice that after the messages are sent, then the next group of messages to be sent follow the same pattern.

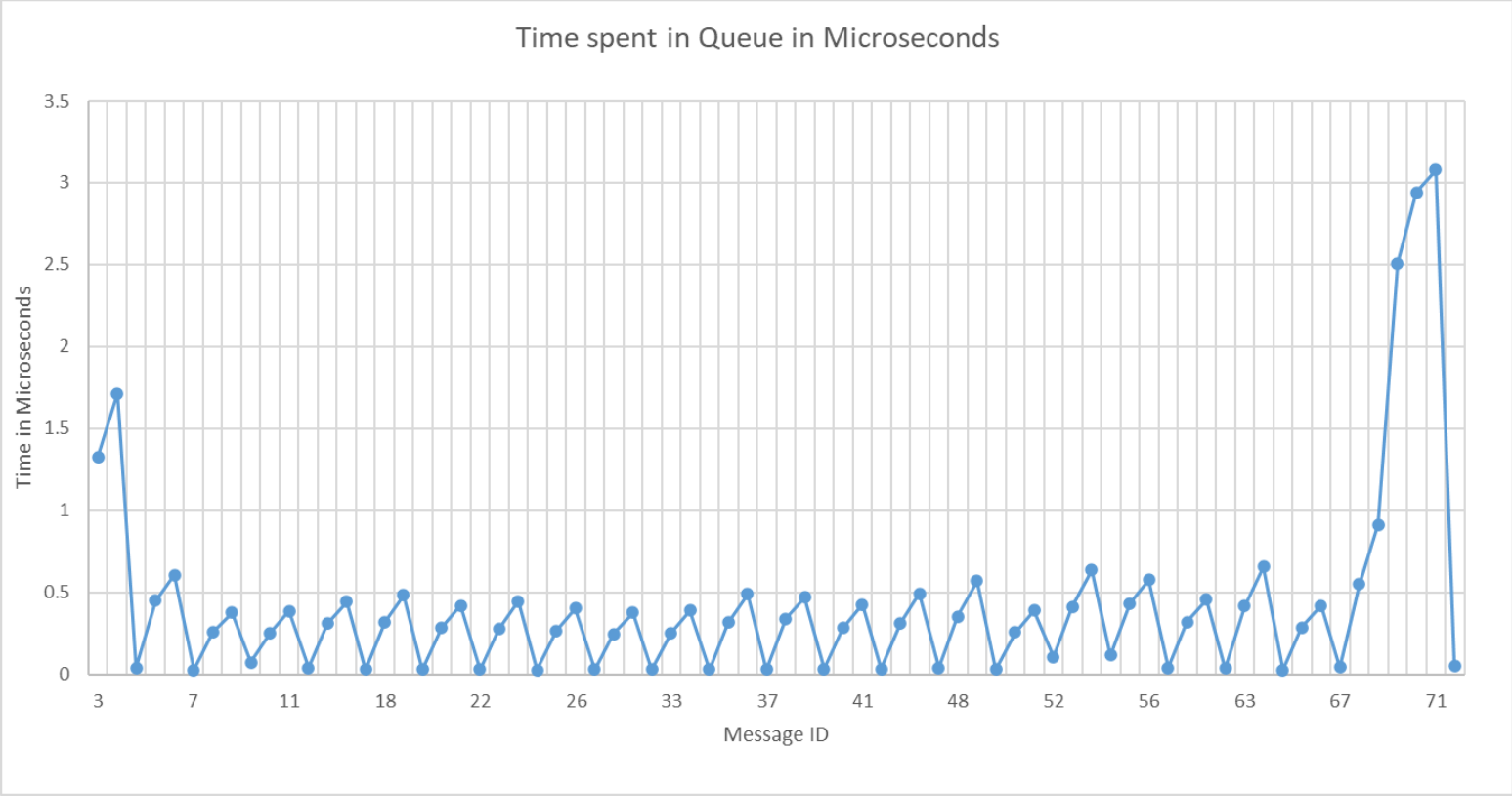

Figure 20: Time spent in the queue for multiple message communications- Raspberry Pi 3

In figure 20, we have the time spent in the queue for each message that was sent from the ground station to the drone. It should be noted that this is being simulated on the Raspberry Pi 3. One can see how the messages are being placed in the queue as more of them come in. You see, because the protocol is sending multiple messages from the ground station to the drone, there are

too many wanting to be sent, so they need to be first placed in a wait queue and then sent off based on their priority. One can see evidence of that happening here by the shape of the graph. As more messages come in, the time spent in the queue increases. One will notice that after the messages are sent, then the next group of messages to be sent follow the same pattern.

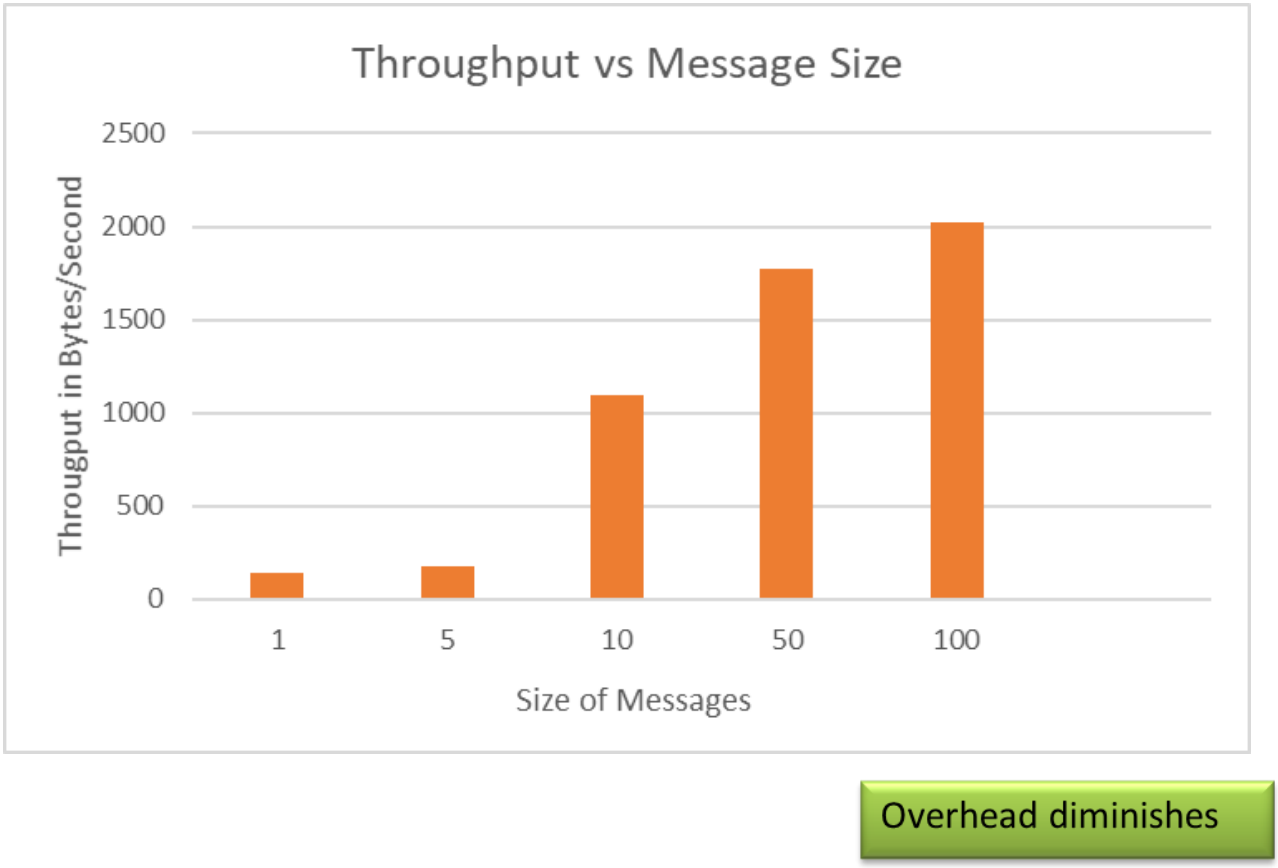

Figure 21: Throughput - Raspberry Pi 3

This graph provides insight as to the throughput for sending messages to the drone from the Ground Station. In this case, the Ground station represented the Laptop, and the Drone represented the Raspberry Pi 3. It can be seen from the graph, that once the message size approached 100 Bytes, it begins to level out and therefore hint at the diminishing of the overhead

needed to send data. In other words, the bigger the message size, the smaller the overhead required to send the message. This is seen as the message size approaches 100 bytes. First the data was gathered using message sizes of 1 byte, then 5 bytes, and so on until reaching a point in which the overhead would begin to diminish. It was found as previously noted that 100 bytes would be the point in which the overhead would diminish.

7. ECONOMIC, PUBLIC HEALTH, SAFETY, WELFARE, AND ENVIRONMENTAL ANALYSIS OF RESULTS

This project was undertaken with the intention of providing the St. Mary's University Drone lab with a communication protocol it can call its' own. This protocol directly impacts economic, public health, and safety because it involves the ordered delivery of critical messages that the ground station needs to send and receive to the drones it will fly. If these messages are not well protected, then safety is impacted, and as such, so is public health. Ultimately, if the protocol cannot function as intended, then it gains a potentially negative economic impact since the protocol will not be able to be used for any sort of commercial purposes if it cannot be reliable. Therefore, the project team sought to positively impact the aspects of public health, safety, and economic factors, through the UAV communications protocol.

8. FURTHER IMPLEMENTATION

The UAV communications Protocol will be utilized as the St. Mary's University Drone Lab sees fit. It is not known for what purposes the drone lab will use the protocol specifically, but

the protocol's usage will be dependent on the implementation in which the drone lab wishes to use the protocol. Since it was foreseen that there could be many possible usages for this protocol, a way to monitor the usage of the protocol was provided, as requested by the customer, to ensure that the protocol can keep its' statistics and better assist in whatever implementation the future may hold for the protocol. This, along with the other requirements requested by the customer, such as encrypted communication, allows for the protocol to be ready for the drone lab's needs.

9. REFERENCES

ArduPilot Documentation https://ardupilot.org/ and https://ardupilot.org/dev/docs/mavlink-basics.html

Beginning C++ Programming - From Beginner to Beyond https://www.udemy.com/course/beginning-c-plus-plus-programming/

Communication and Networking Technologies for UAVs: A Survey https://www.sciencedirect.com/science/article/abs/pii/S1084804520302137

CompTIA Network+ Certification Video Course https://www.youtube.com/watch?v=vrh0epPAC5w

Computer Networking Course - Network Engineering [CompTIA Network+ Exam Prep] https://www.youtube.com/watch?v=qiQR5rTSshw

Drone Programming Primer for Software Development https://www.udemy.com/course/drone-programming-primer-for-software-development/

OpenSSL Foundation, Inc. "OpenSSL." */Index.html*, OpenSSL Software Foundation., 2018, www.openssl.org/

Communication and Networking Technologies for UAVs: A Survey https://www.sciencedirect.com/science/article/abs/pii/S1084804520302137

Securing the MAVLink Communication Protocol for Unmanned Aircraft Systems Technical Report #CSSE14-02 https://www.semanticscholar.org/paper/Securing-the-MAVLink-Communication-Protocol-for-%23-Butcher-Stewart/4ce068b40089549f3d445d30e45fe8b53a141c88?p2df

Secure Shell Fundamentals - Learn SSH By Configuring It https://www.udemy.com/course/secure-shell-fundamentals-learn-ssh-by-configuring-it/learn/lecture/16548080#overview

10. APPENDICES

Link to GitHub

https://github.com/JG907/UavDroneProtocol.git

Link to Jira

https://jnseniordesign.atlassian.net/secure/RapidBoard.jspa?projectKey=SEN&rapidView=2&view=planning.nodetail&atlOrigin=eyJpIjoiNWIwZjJlNGI0M2E2NGEzN2FjZmY5ZDgxN2U5N2M1ZTUiLCJwIjoiaiJ9

11. SMC CAPSTONE REFLECTIONS

The senior design project was a project that was unlike any I had ever participated in. Prior to the senior design project, I only ever had classroom projects that were in a much smaller scale. The senior design project given its size and complexity, was valuable because it allowed for me gain a lot of insight into the potential future work I will participate in once graduating. I really enjoyed the number of topics touched in this course and will value this learning experience from here on.

This project has allowed me to gain an insight into how the industry will be like. I had a great time putting together the knowledge I have gathered throughout my four years here at St. Mary's University. It was eye opening to see how all my classes can come together to make something concrete. Despite it being a lot of work, I really enjoyed this Senior Design class. This class was a great experience, and I am certain it will benefit me for the rest of my career and studies. Not only did I gain experience with designing and with requirement elicitation, but also with talking to people and working together to get things done. I appreciate the efforts my professors and classmates made to help in this project because without their support, it would not have been possible to complete this project. This class promoted a good environment for learning and growing, and I certainly felt that. I hope the UAV communications protocol is used for something good in the future, for the benefit of helping humanity. There was a ton of learning involved, and I am happy to have learned as much as I did.

There were times where the project was stuck and did not move along, COVID as-well as personal obstacles got in the way for each of us and it caused setbacks that needed to be overcome. There was a lot of things that happened throughout this project, good and bad, but the overall outcome is one which I think is one to be taken as a lesson and to be reflected upon. I believe that although there were many obstacles, there were also a lot of good moments, along eye-opening lessons. I am grateful for this project and will be affected by it the rest of my career and studies. I am thankful to my professors, classmates, and those at St. Mary's University, as-well as the audience who took time from their days to watch us all give our presentations and read our reports.