# Towards A Mobile Ad-Hoc Mesh Network Establishing Emergency Drone System

A. R. Justman, R. K. Fischer, R. A. Khayat, J. M. Rasure, G. Mazzaro and R. Integlia

*Abstract*-- **A drone based mobile ad-hoc mesh network for emergency communications is discussed. The in-progress project seeks to improve emergency and disaster area communication systems by creating a mobile, ad-hoc wireless network with an array of microcomputers, a GPS receiver, IMU, network adapter and drone. The Linux based platform includes network management, data collection, and integration with visualization. The expected outcome of this project is the establishment of a wireless mesh network capable of self-healing to support emergency response.**

***Index Terms*—Mobile ad-hoc mesh network, drone, GPS, IMU, Batman-adv, Open-WRT, Machine Learning, Linear Regression, Exponential regression, K-means clustering, Self-healing, emergency response**

## I. Introduction

THROUGHOUT history, humans have been developing ways to communicate effectively with each other, and Wi-Fi networks are the most recent technological innovation in this field. Wi-Fi mesh networks were first introduced in the 1980's and had the advantage of being more reliable than traditional topologies [2]. However, due to cost and complexity it did not see any major advancements. In recent years due to the lowering costs of computing power there has been a renewed interest in mesh network technologies [2]. Team spider colony is working to further this field of study through the development of the Network Establishing Emergency Drone System (NEEDS).

The goal of this project is to provide a wireless intranet for ad hoc deployment wherever emergency service personnel would need it. The target application of this technology is nonprofit emergency service but there are also opportunities in the military and private sectors.

As NEEDS advances, it will move progressively integrate the use of autonomous drones to establish and monitor a self-healing wireless mesh network in any location. In its current state, a user places network nodes in the intended coverage area to establish the initial network. These nodes are access points into the network and allow team members to communicate through a Wi-Fi connection using a Rocketchat server. Once the wireless network is established, a Scout drone is navigated through the coverage area by a user and collects GPS and signal strength data. CIC analyzes the gathered information using data interpolation and machine learning to predict the best additional node locations. These new nodes would increase network coverage and "heal" the network. Using this technology, team Spider Colony is confident that we can provide reliable wireless intranet.

## II. Hardware

This section describes the components used in the NEEDS design. The system consists of four separate units: the network gateway control information center (NCIC), the network bridge (Worker), the network Scout and Data Integration Control Center (DICC). Fig 1 is a level 1 diagram of the network and shows the interconnections between each component. Each network unit consists of a power source, a Raspberry Pi 3B+ and Wi-Fi network adapter. The Scout is also equipped with a Berry GPS unit. Information is exchanged between network devices across the mesh using a typical 2.4 GHz Wi-Fi radio frequency. A photo of all network components can be seen in Fig 2.

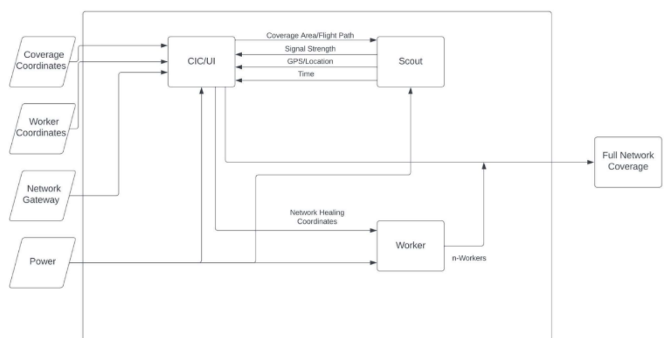



Fig 1 NEEDS level 1 diagram

### A. Raspberry Pi 3B+

The Raspberry Pi 3B+ is a micro-computer used to control each network unit. NCIC uses the Raspberry Pi to maintain the network gateway, assign network IP addresses and establish a chat server. In the Workers, the Raspberry Pi is used to establish bridges to the network which creates an expanding network area. The Scout uses a Raspberry Pi to control the collection of GPS and signal strength data.

### B. Typhoon H drone

The Typhoon H drone is an off-the-shelf drone with a detachable gimbal. The drone is used as the carrier for the Scout unit, allowing for faster data collection in the coverage area. For this project, the gimbal is removed and replaced with a 3D printed housing for the Raspberry Pi 3B+, a battery, and the other components that comprise the Scout.

### *C. GPS*

The Berry GPS is a module for the Raspberry Pi 3B+ which processes GPS satellite data. The GPS unit is attached to the Raspberry Pi inside of the 3D printed housing on the Scout. The signal quality is improved by an external GPS antenna mounted to the Typhoon H drone. The Berry GPS also streams IMU, altitude, and barometric pressure data to the Raspberry Pi.

### *D. Wi-Fi Adapter*

The Wi-Fi network adapter is an external Wi-Fi antenna that serves multiple purposes in this project. The Canakit Wi-Fi antenna allows access to the network when attached to the Worker and the ALFA network adaptor monitors signal strength when attached to the Scout. The ALFA is used on the Scout for its better data collection property. The Wi-Fi network adapter can also be attached to the NCIC allowing access to the internet if an internet router is available.

Fig 2 NEEDS Hardware

## III. SOFTWARE

This section explains the various software codes developed and used to create and monitor a wireless mesh network and process data for self-healing. It describes Linux scripting to enable Open-Wrt (batman-adv), and the use of Python to develop the DICC, and NCIC graphical user interface (GUI).

### *A. Network*

The network is created using the Batman-adv implementation of Open-WRT. Batman-adv is a mobile ad-hoc network that is part of the Linux operating kernel and can be set up on any device running the Linux OS. Each Raspberry Pi has to have open-source software installed and network parameters set. An installation script was developed to streamline this process, reducing errors and drastically lowering the time required to create network nodes.

NEEDS uses three different start-up scripts to initialize the Raspberry Pi as either a gateway for NCIC, a bridge for the Workers or as the Scout. The flow chart for these scripts is shown in Fig 3 and are added to the boot sequence of each Raspberry Pi. By adding this script to the boot sequence, a Worker, Scout, or NCIC is initialized on startup. The scripts dedicate the built in Wi-Fi network adapter (wlan0) to bat0 which establishes the mesh network. The bridge node script enables a second Wi-Fi network adapter (wlan1) to act as an access point into the mesh network and the Scout script starts the data collection subroutine discussed in Section III B.

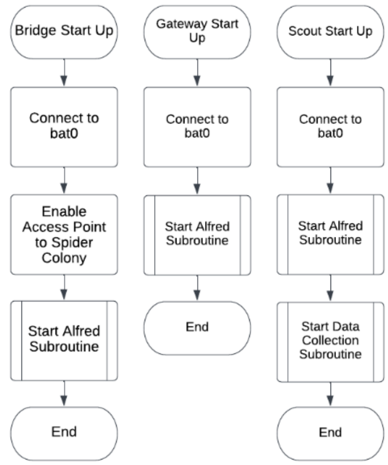



Fig 3 Network Start Up Flow Charts

Each script also enables the Alfred subroutine which takes advantage of batman-adv's Alfred protocol. Alfred is a Linux daemon integrated with the Batman-adv network. Alfred stores and distributes a small amount of data, typically one line, throughout the entire network. A user classifies system files as Alfred files and Alfred catalogs them. These files can be accessed by querying Alfred for the file through different nodes. Alfred handles the operations and file transfer. When a file is received from Alfred it contains the information in the file along with the host device's MAC address. This daemon is an essential component allowing the user to control the network from NCIC.

The Alfred subroutine flow charts are shown in Fig 4 and enable control of the wireless mesh network from NCIC. The subroutine continuously monitors Alfred files and will update the node if a change was made. The subroutine also prevents file time out (typically 10 mins) by refreshing Alfred every 9 minutes. The subroutine for each bridge monitors the Echo Location Protocol (ELP) and Originator interval levels. These levels adjust the time between network routing table updates allowing for better control of network overhead. This allows NEEDS to shorten the time between updates during initial setup while nodes are moving. Since establishment of the network for emergency services is critical, routing tables must be updated quickly. Once the network is stable, routing tables can update less frequency allow for higher network overhead.

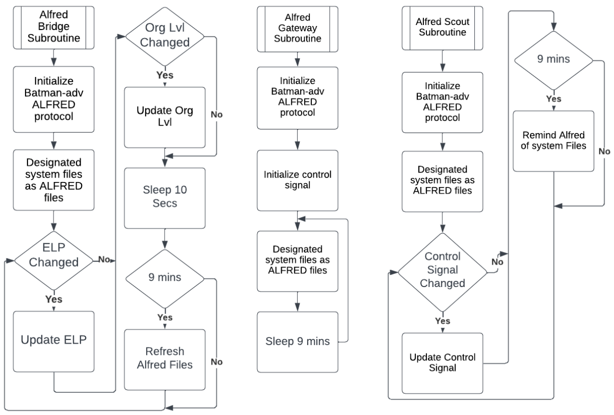



Fig 4 Alfred subroutine flow charts

### B. Scout

The Scout combines GPS coordinates provided by a Berry GPS with the strongest available signal strength of the mesh network obtained through network scan features included with the Linux operating system. While navigating through the user defined coverage area, data is logged into a csv file once per second. This produces the required data used in determining the network's integrity. This file is then available to be extracted with a flash drive or ported over the intranet to DICC.

Upon boot, the Scout initializes the data collection sub-routine and waits for the start collection command to be sent from NCIC, as depicted in the Fig 5. Starting from boot allows for the Berry GPS unit to obtain a first fix and start collection of data without the need of a screen. Once data collection is initiated, the Scout will create a timestamped csv file and begin to log GPS position and satellite data followed immediately by a network scan to obtain the signal strength data. Additional raw data from the IMU is logged for future development.

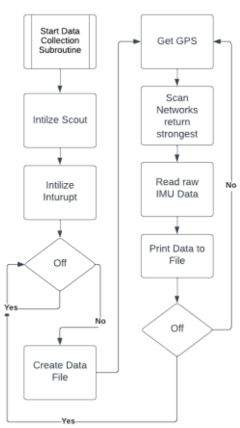



Fig 5 Scout Data Collection Flow Chart

GPS data collection relies on the GPSD daemon to interface with the Berry and collect the desired information over the Raspberry Pi's serial port. The serial port is defined during startup of the Raspberry Pi. This daemon automatically detects many different GPS devices, protocols, and configurations, simplifying the connection to the Berry GPS. GPSD does calculations efficiently and does not add any measurable error to the position data. The design and physics of the Berry GPS typically produces an error under 5 meters and needs to be combined with an IMU for location corrections. This location correction can be used to help with the navigation of a drone. NEEDS is currently only using the GPS location for network mapping purposes.

The Berry GPS uses a u-blox CAM M8Q GPS module, a LSM6DSL gyroscope/accelerometer module, a LIS3MDL Magnetometer module, and a BMP388 barometric module. The IMU module utilizes the i2c bus to communicate with the Raspberry Pi and is critical for further development. U-center is a software developed by the u-blox manufacturer and was used to collect accuracy information as seen in Fig 6. U-center

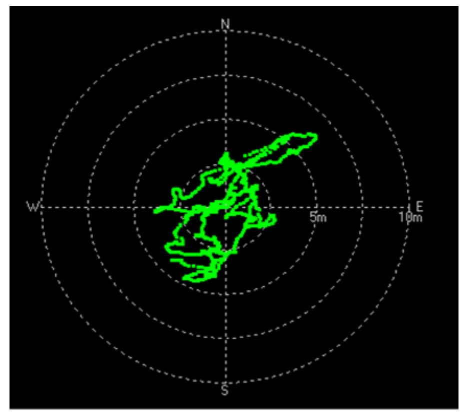
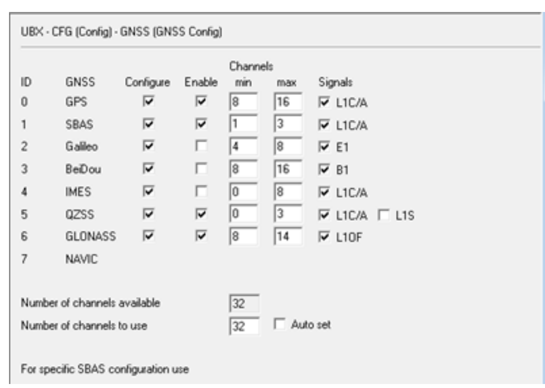



Fig 6 U-blox CAM M8Q GPS module

was also used to determine the best configuration for the GPS module shown in the Fig 7. It can track 3 GNSS satellites concurrently or track 2 GNSS's and implement SBAS. No major difference in error was detected in either configuration.



UBX - CFG (Config) - GNSS (GNSS Config)

| ID | GNSS | Configure | Enable | Channels min | max | Signals |
|---|---|---|---|---|---|---|
| 0 | GPS | ☑ | ☑ | 8 | 16 | ☑ L1C/A |
| 1 | SBAS | ☑ | ☑ | 1 | 3 | ☑ L1C/A |
| 2 | Galileo | ☑ | ☐ | 4 | 8 | ☑ E1 |
| 3 | BeiDou | ☑ | ☐ | 8 | 16 | ☑ B1 |
| 4 | IMES | ☑ | ☐ | 0 | 8 | ☑ L1C/A |
| 5 | QZSS | ☑ | ☑ | 0 | 3 | ☑ L1C/A ☐ L1S |
| 6 | GLONASS | ☑ | ☑ | 8 | 14 | ☑ L1OF |
| 7 | NAVIC | | | | | |

Number of channels available 32
Number of channels to use 32 ☐ Auto set
For specific SBAS configuration use

Fig 7 UBX Configuration Menu

### C. Data Analysis

All data analysis, processing and decision making is managed by multiple python programs, written in an object-oriented style, on a laptop at DICC. The scout.py class is set so that multiple objects can be created based on the use case of the mesh network. If multiple meshes need to be established, multiple scout objects can be run to manage each mesh. Oracle, a scout object, initializes the scout properties for data processing. This initialization imports all necessary libraries such as Pandas, NumPy, SciPy, Scikit-Learn (sklearn), and Matplotlib. The time stamped csv data file is manually pulled from the scout, renamed to "Oracle.csv", and placed in the program's venv file.

The program starts by initializing preset variables. A variable named threshold is a signal strength of -57 dBm. Team Spider Colony determined -57 dBm to be the threshold for accepting good data transfer. This value was chosen to ensure each node had optimal network connection. The learning threshold is set to -60 dBm for decision making on whether or not the data can be added to the database for increasing model accuracy.

After the user inputs the number of nodes deployed and decides whether the data should be added to the learning model, the data from Oracle.csv is imported into the program. The data is then copied, sorted, and variables are created for the various data cleaning, manipulation and prediction models to be used by the various methods.

Different columns from each DF are written into individual arrays. The latitude and longitude arrays are interpolated into a grid, using Numpy. The signal strength array is interpolated linearly between latitude and longitude positions. This creates a signal mapping where the initial data points are interpolated into 10,000 datapoints. The percentage of datapoints exceeding the learning threshold of -60 dBm is saved to a modeling database. Fig 8 runs the majority of the above methods and data cleaning.

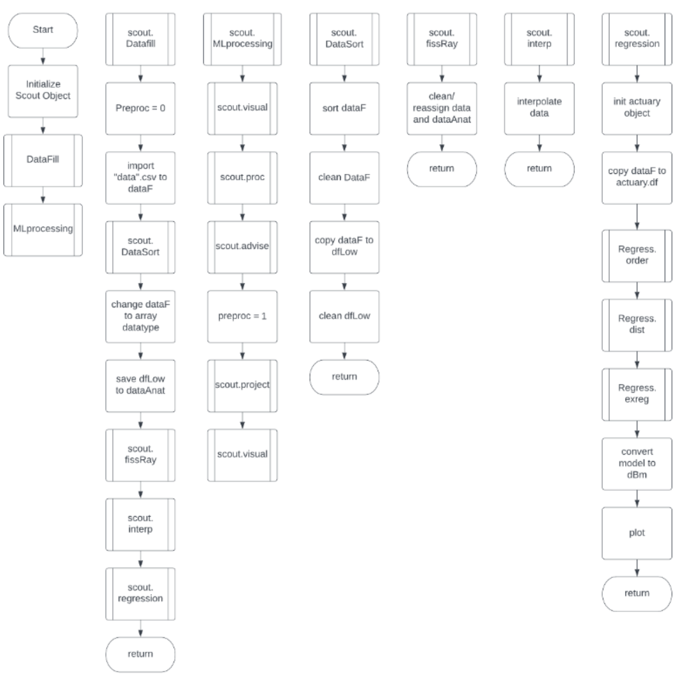



Fig 8 Scout Class Flow Chart

The program has the ability to take two different paths based on the nature of the data. If the data has a percentage that equals or exceeds 80% of the learning threshold level, has not been used for a learning model before, and ensures there is only 1 deployed node, the program will run the new data through the learning portion of the Regress class.

This starts the machine learning portion of the code. With the parameters passed, Regress will find the max value of signal strength and use the Haversine distance calculation program to find the distance between each datapoint and the node. Regress will convert each dBm measurement into Watts and run an exponential regression by fitting a linearized exponential function using the transformation as seen below:

$$y = ab^x$$

$$\ln(y) = \ln(ab^x)$$
$$\ln(y) = \ln(a) + x\ln(b)$$

Linear regression from sklearn is run to calculate the coefficients for the exponential model. These coefficients and data points are saved to their respective databases and the model coefficients are passed back to the main scout class. By increasing the population of data that the model draws from the accuracy of the model will increase with every iteration of the learning model being run. If the data is not being used for the learning process, the program will pull coefficients from the model in the database. The regression model is visualized, after being transformed back to exponential form, for the user in Fig 9.

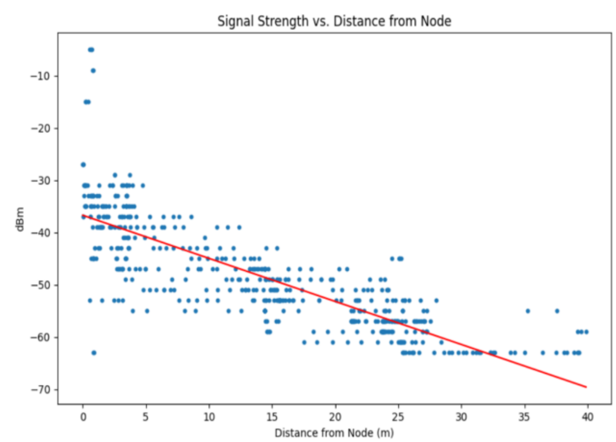



Fig 9 Exponential Regression

The interpolated signal mapping is visualized into two separate color graphs for the user in latitude and longitude as seen in Fig 14. If the data did not trigger the learning parameters then it had more than one node, see Fig 15, or it did not pass all parameters required.

Using the signal mapping data, the user decides how many nodes they wish to deploy. These nodes are used as clusters for K-Means. The K-Means algorithm runs 20 seed sets with maximum 300 iterations per set to cluster the areas of low signal integrity and find the centroids. The centroids are used with the regression model to simulate new nodes, Fig 10. Predicted values are generated for latitude and longitude positions and compared with the original data, generating a new predicted signal mapping that is visualized for the user, Fig 11.

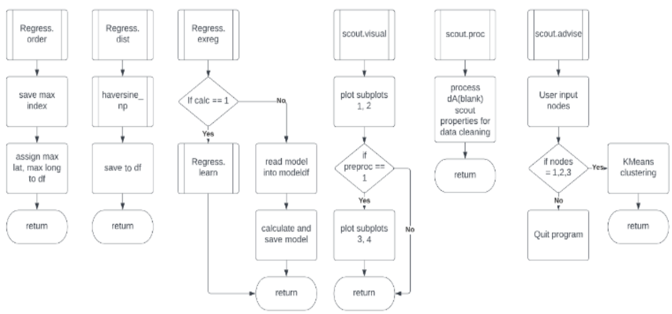



Fig 10 Scout Class Flow Chart Machine Learning Portion

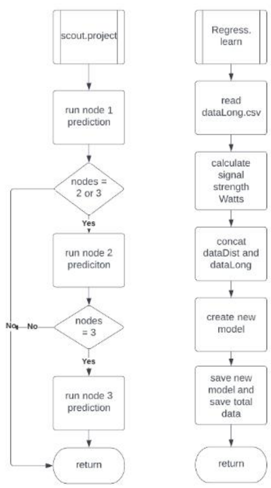



Fig 11 Scout Class Flow Chart w/ learn

## D. Network Control GUI

The Network control GUI has many functions allowing NEEDS to control the network from one central location. NCIC is established using a batman-adv gateway which gives an IP address to any device connected to the network. These IP addresses allow for the operation of a Rocketchat server. Rocketchat is an open-source chat server which allows text communications for any user connected to the network.

Fig 12 is the CIC GUI flowcharts which initializes all GUI functions, establishes program interrupts, and waits for user input. The GUI pulls a list of known client MAC addresses, and random MACs that appear on system start up from system files. The Gui also receives a list of node names and GPS coordinates from Alfred. CIC uses these lists to increase readability of batman-adv outputs. The GUI has five different functions and is shown in Fig 13 with the flow chart of these functions as shown in Fig 12. Starting/Stopping Scout data collection and updating ELP/ORG levels both take advantage

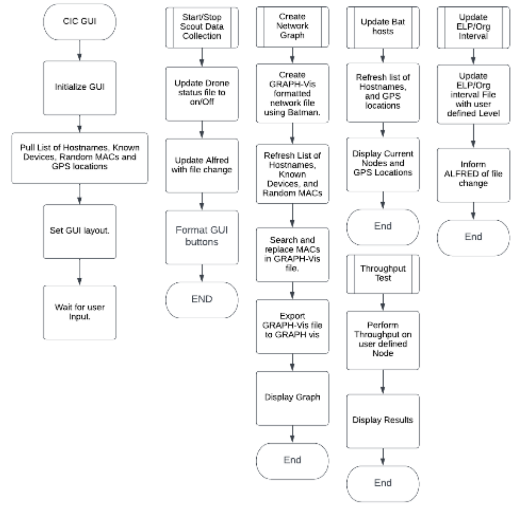



Fig 12 CIC GUI Flow Charts

of the Alfred subroutine discussed in Section III A. When the user initiates these functions, Alfred files associated with the function, are changed, and Alfred is updated. These files are then checked by each node using the Alfred subroutine causing an interrupt to trigger if any changes were made. The throughput test performs a batman-adv throughput test on the selected node and displays it in the GUI. Updating batman hosts displays the current nodes attached to the network and their current GPS location. Finally, the generate network graph function produces a graphical representation of the network by first pulling the routing table information stored on Alfred and then comparing these graphs to the lists of host names, known devices and random MACs. The program then replaces the MAC address with readable device names and creates a network graph using Graphviz.

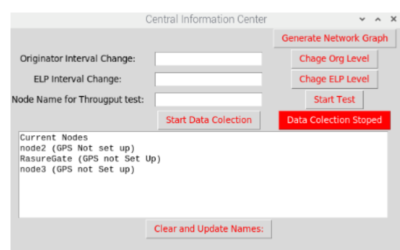



Fig 13 CIC GUI

# IV. Field Testing

## A. Field Test

Field tests were performed on a single and double Worker network. For the single Worker network, a node was placed near the center of a football field, and the Scout was carried through the coverage area collecting signal strength and GPS locations. This data was then imported into DICC for data analysis which produced Fig 14. The top left of Fig 14 is an interpolated heat map of signal strength signal based on GPS location. The bottom left graph of Fig 14 indicates areas that do not pass threshold specification. The top right graph is the predicted location for new Workers that heal the network based on the K-means clustering algorithm. The bottom right graph is the predicted coverage area once the new Workers are placed into the coverage area. This process of monitoring the area, signal mapping, and predicting node locations demonstrates the self-healing capabilities of NEEDS. Fig 15 is the results of a double Worker network and demonstrates NEEDS ability to monitor a multi-nodal network.

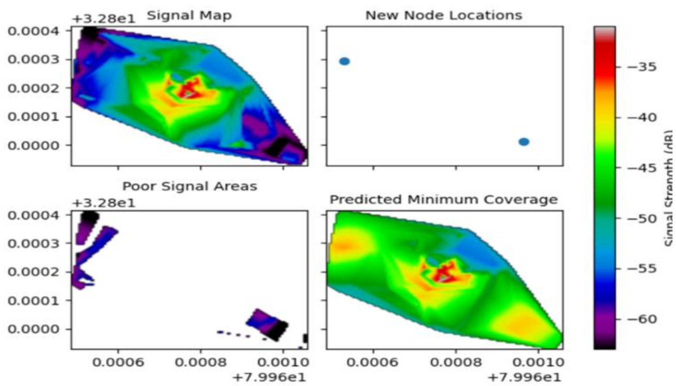



Fig 14 Data Analysis Output

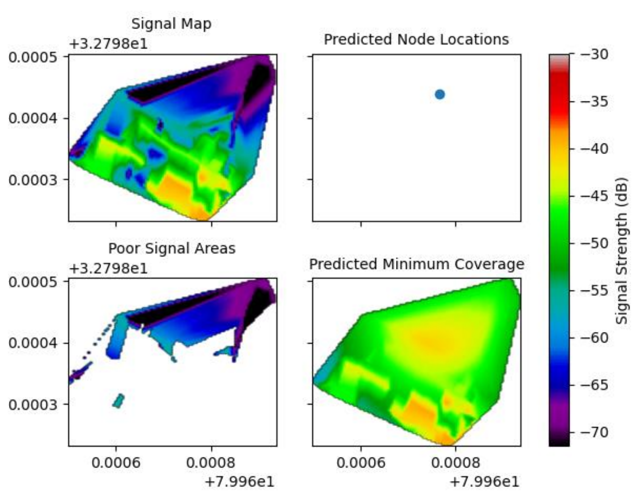



Fig 15 Multi-Nodal Field Test

During these field tests a network connection graph was created by NCIC and is shown in Fig 16. This figure represents the ability for NCIC to pull network data from the Batman-adv routing tables, filter/replace MAC addresses and inform the user of network connections. The other functions of the CIC were also tested successfully to demonstrate the ability to control the network. Lastly, test users connect to the Rocketchat server and send text communications over the network which demonstrated the ability for emergency response personnel to communicate across the network.

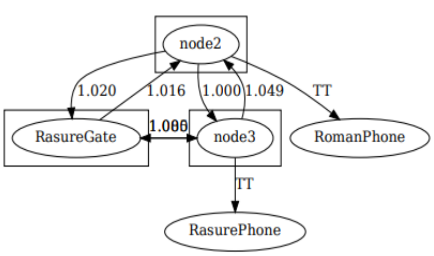



Fig 16 Generated Network Graph

## V. Sources of Error

### A. Berry GPS

The GPS data is an integral part of the project. When initial GPS data was collected, the program ran sporadically. After some trial, error and research, it was determined that the Berry GPS unit required 2.4 amps to operate reliably instead of the 2.1 amps that had previously been supplied. The solution implemented was to use a battery with a larger output current. There is a chance that the output current of the new battery could drop below the 2.4 amp threshold as it discharges to a low state. However, this is unlikely to be realized on account of the drone having a 20 minute flight time. Another source of error that can be attributed to the GPS is unstable and unreliable data. Due to constantly changing conditions and random anomalies, occasionally a lack of GPS data causes the program to end its loop or lag in the data collection. This is due to the Berry GPS not being as suitable for drone use as other GPS units. A fix for this would be to change the GPS unit to a different make.

### B. Signal strength

Gathering signal strength presents another potential source of error to the project. During testing, the signal strength data would intermittently stop refreshing and continuously output identical data. The cause is the way the Raspberry Pi stores its signal strength data. The solution to this was found in a Linux command that force refreshes the signal strength data before sampling the data. The final piece of acquiring reliable signal strength was to test the system and determine how long it took for the data to update, and then making the program wait that set length of time before retrieving the data.

## VI. Future Work

This project acts as a proof-of-concept demonstration that it is possible to establish and monitor a mobile ad-hoc mesh network and has many opportunities for continued development. Future work in hardware includes advancements to data collection, automation of drones, and increased coverage area. Automation can be advanced through the implementation of neural networks for image processing, increasing model accuracy and expanding to 3 dimensions.

### A. Drone Automation

The automation of the Scout allows the drone to fly along a preprogrammed path through the network coverage area, freeing up the operator for other responsibilities. Automated drones can also be used as a delivery system for network Workers. Further improvement to automation involves the development of an algorithm for Worker nodes. This algorithm would allow Workers to adjust their topology from predictive models made at DICC to ensure proper coverage. This combination of automated drones and prediction models will help be less invasive for the user due to its properties of self establishment and self-healing.

### *B. Data Collection*

GPS accuracy is important for this project and can be increased through the integration of the Berry GPS's built in IMU. Integration of this IMU would give access to the Scout's direction of travel and acceleration allowing for GPS location correction. IMU altitude integration would expand signal mapping and data analysis into the third dimension.

Further work on the signal strength detection would involve adding multiple Wi-Fi network adapters with different antenna orientations. This would allow the Scout to determine the max signal it receives regardless of the polarization of the incoming signal. Increasing the number of Scouts either through drones or have network users send back signal strength and position data, will decrease the amount of time to generate accurate signal maps.

### *C. Data Processing*

Automating the entire data processing portion is the ultimate goal. Taking more measurements that pass parameters will increase the accuracy of the regression model. The next step would be to have a user input an area based in latitude and longitude and have the program automatically determine node locations for the best coverage and deploy. Neural networks can be applied in two ways. One is a weighting system for the model to further increase the accuracy of the predictive model and possibly allow it to be scalable. The second is using image processing to identify physical barriers to signal coverage and do proper predictive modeling based on those barriers which would lead to eventual adaptation to modeling signal propagation through interior spaces.

Live data processing and continually updating the signal mapping will allow for a dynamic system that can continually operate and conform to changing scenarios and needs of the user. Integrating the NCIC into the DICC will allow for active data acquisition and manage NEEDS as a complete Central Information and Control Center.

### *D. Increased Coverage Area*

To increase network coverage area, the first step is to increase the size and power of the network adapter. Once the network coverage area has been increased a plan will need to be implemented for the deployment of the nodes. One idea involves utilizing multiple heavy weight drones. These drones would deliver Workers to the locations specified by DICC. A new system for monitoring the coverage area would also need to be addressed as the current drone only has a battery life of 20 mins. An idea was to develop a battery recharging station that the Scout would return to on low power.

## VII. Conclusion

The goal of this project was to develop an autonomous mobile ad-hoc mesh network that could be deployed for emergency communications in any location. The initial steps towards this goal were achieved with the implementation of open-source software, programming languages and the integration of GPS and network adapters with Raspberry Pis. The field tests of NEEDS demonstrate its ability to create and monitor an emergency communications system.

Future work on this project will focus on fully integrating autonomous drone flight and removing errors associated with gathering GPS and signal strength data. These efforts will continue to progress mesh network technologies and provide better communication in emergency situations. When the goal of this system is achieved it will have a major impact on the way emergency situations are handled and has the potential to save many lives.

## VIII. Acknowledgements

The authors gratefully acknowledge the contributions of:

Dr. Gregory Mazzaro and Dr. Ryan Integlia

## IX. References

[1] T. Frank, "Cell Phone Service Must Be Restored Quicker after Hurricanes", *Scientific American*, 2019, October 8, https://www.scientificamerican.com/article/cell-phone-service-must-be-restored-quicker-after-hurricanes/

[2] G Fleishman, "Wireless mesh networks: Everything you need to know", *PC World*, MAY 5, 2020

[3] M. Lindner, S. Wunderlich, L. Lüssing, A. Quartulli, M. Hundebøll, S, Eckelmann, "Better Approach to Mobile Ad-hoc Networking Wiki", https://www.open-mesh.org/projects/open-mesh/wiki, Jan 26, 2023

[4] H. Gupta, Zongheng Zhou, S. R. Das and Q. Gu, "Connected sensor cover: self-organization of sensor networks for efficient query execution," in IEEE/ACM Transactions on Networking, vol. 14, no. 1, pp. 55-67, Feb. 2006, doi: 10.1109/TNET.2005.863478.

[5] N. Tatebe, K. Hattori, T. Kagawa, Y. Owada and K. Hamaguchi, "Energy-efficient construction algorithm for mobile mesh networks," The 20th Asia-Pacific Conference on Communication (APCC2014), 2014, pp. 73-77, doi: 10.1109/APCC.2014.7091608

[6] P. Zhou, X. Fang, Y. Fang, R. He, Y. Long and G. Huang, "Beam Management and Self-Healing for mmWave UAV Mesh Networks," in IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1718-1732, Feb. 2019, doi: 10.1109/TVT.2018.2890152.

[7] S. Morgenthaler, T. Braun, Z. Zhao, T. Staub and M. Anwander, "UAVNet: A mobile wireless mesh network

using Unmanned Aerial Vehicles," 2012 IEEE Globecom Workshops, 2012, pp. 1603-1608, doi: 10.1109/GLOCOMW.2012.6477825.

[8] Babu, K M Dinesh, Geetha Vaidyanathan, Jvn Ravikumar and P. Sunil. "NavIC and GPS State Vector as Tracking Sources for Flight Safety." 2019 International Conference on Range Technology (ICORT) (2019): 1-4.

[9] Winkler Chris, Comerford Richard. "The critical role of sensors in drone innovation." Electronic Products, June 19, 2016

[11] "API reference#," *API reference - pandas 2.0.0 documentation*. [Online]. Available: https://pandas.pydata.org/docs/reference/index.html#api. [Accessed: 08-Apr-2023].

[12] "API reference," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/classes.html. [Accessed: 08-Apr-2023].

[13] G. Sanderson, "Neural networks," *YouTube*, 05-Oct-2017. [Online]. Available: https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi. [Accessed: 08-Apr-2023].

[14] "Lost but lovely: The haversine," *Plus Maths*, 04-Jul-2014. [Online]. Available: https://plus.maths.org/content/lost-lovely-haversine#:~:text=The%20term%20%22haversine%22%20apparently%20comes,from%20%22half%20versed%20sine%22. [Accessed: 08-Apr-2023].

[15] "Numpy reference#," *NumPy Reference - NumPy v1.24 Manual*. [Online]. Available: https://numpy.org/doc/1.24/reference/index.html. [Accessed: 08-Apr-2023].

[16] S. Potisuk, "Introduction to Machine Learning (ML)," in *330 - Grimsley Hall - ELEC 413-81*, 2023.

[17] "Scipy API#," *SciPy API - SciPy v1.10.1 Manual*. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/index.html#scipy-api. [Accessed: 08-Apr-2023].

## X. Biographies

**Andrew Justman** earned an Associates in Electronic Engineering Technology in 2014 from Trident Technical College while working full-time at the Naval Weapon Station Commissary. He then took an internship with Venture Aerobearings from 2014-2015. He moved onto a full-time position with Cummins Inc in 2016 through present as an Instrumentation/Emissions Technician working on electro-mechanical systems. Andrew Justman is expected to graduate from The Citadel with a BSEE in 2023.

**Randel Fischer** has worked as a defense contractor for more than 15 years. His career has focused mainly on installing and testing radio systems but also includes work leading to expertise in pneumatic, hydraulic, and mechanical systems. He has been married for over ten years to Carrison Fischer and has two children, Chloe and Mason Fischer. Randall Fischer is expected to graduate from the Citadel in May of 2023 with his BSEE degree and plans to use it to further his career in the defense contracting industry.

**Roman Khayat** started in the United States Coast Guard at Small Boat Station Elizabeth City, NC training and operating in Search and Rescue and Law Enforcement. He went on to attend Electronics Technician school at TRACEN Petaluma, in Petaluma California. After completing training, he was stationed at Base Charleston where he supported Cutters, Search and Rescue Stations, FLETC, MLEA, and ANT's from Georgetown, SC down to Tybee, GA. After almost 6 years in the Coast Guard and he separated and pursued an education in Electrical Engineering at The Citadel. Graduating Summer 2023 he has been already accepted to the Accelerated Masters Program and expected to graduate May 2024.

**Jacob Rasure** started his career as a Navy Nuclear technician in 2009 when he attended Navel Nuclear Power Training Command. He spent 5 years as an electrical operator on the USS Greeneville and was then stationed as an instructor at Navel Nuclear Power Training Command for 3 years. After separating from the Navy, he began interning at Navel Information Warfare Center as a wireless network engineer and attending The Citadel Military College of South Carolina. Jacob Rasure is expected to graduate with a BSEE in 2023. He has also been accepted into The Citadel's accelerated Masters program and intends to further his education after graduation.