Dynamic Mesh Network for Telemetry Propagation and Communications in Coordinated Drone Swarms

Eric A. Cai
Donald Bren School of Information and Computer Science
(*Computer Science and Engineering*)
University of California, Irvine
Irvine, CA , USA

Davis K. Furukawa
Donald Bren School of Information and Computer Science
(*Computer Science and Engineering*)
University of California, Irvine
Irvine, CA , USA

Dylan C. Leighton
Donald Bren School of Information and Computer Science
(*Computer Science and Engineering*)
University of California, Irvine
Irvine, CA , USA

Gustavo A. Velazquez
Henry Samueli School of Engineering
(*Computer Science and Engineering*)
University of California, Irvine
Irvine, CA , USA

Haowei Zhang
Henry Samueli School of Engineering
(*Electrical Engineering*)
University of California, Irvine
Irvine, CA , USA

Davide Callegaro
Donald Bren School of Information and Computer Science
(*Graduate Student (Ph. D)*)
University of California, Irvine
Irvine, CA , USA

Marco Levorato
Donald Bren School of Information and Computer Science
(*Professor*)
University of California, Irvine
Irvine, CA , USA

***Abstract*—Unmanned aerial vehicles (UAV) are emerging as a new technology for innovative applications. A dynamic mesh or dynamic ad-hoc network is one interesting network model that is applicable to the high mobility of nodes within these UAV swarms. *BATMAN advanced* was chosen from multiple routing protocols as its flexibility and portability creates the perfect foundation for our own more application specific protocol. Our project aims to create a dynamic mesh and implement a protocol that is application specific to data transfer in networks with high mobility nodes. The current design contains 5 Raspberry Pi's that are configured with the BATMAN adv protocol. We have conducted a Static Node Test that is without interference and Aerial Interference Test with simulated aerial interference by gently shaking the Raspberry Pi's. The results of the test show that the aerial interference tests for bandwidth is better than the tests with no interference. The general range of interest for distances and their respective bandwidths is 0 to 60 meters. However, the maximum effective one hop range of BATMAN adv. is approximately 130 meters. We are not finished with our project yet so there are more refined tests and results that are in future consideration.**

***Index Terms*—Mesh Network, Dynamic Mesh Network, BATMAN, TCP, UAV, Drones, Raspberry Pi, Wireless, WiFi, Bandwidth, Iperf,**

I. Introduction

Drones or unmanned aerial vehicles (UAVs) are becoming more commonplace in today's technologically integrated society. Their uses range from search and rescue in natural disasters, law enforcement, firefighting, mapping, agriculture, etc. However, as the use of drones becomes more prevalent, there is an increasing demand for more efficient and effective communications between these aerial devices. To satisfy the necessary specifications of a robust communication network, research is required for various topological solutions. One specific solution to approach this issue is the use of a mesh network to establish communication between the devices.

A mesh network is a local network topology in which its nodes are connected directly to one another. This interconnectedness avoids the dependency on a central or critical link like that present within a STAR topology. In any communication network, delay and congestion will always be present which facilitates the need for proper routing algorithms to ensure the effective and timely transfer of data. Depending on the numerous scenarios, making use of the proper routing algorithm is essential for the functionality of the mesh network.

Due to the fact that drones are highly mobile devices, the need for a mesh network that compensates for this phenomenon is necessary. Thus, a dynamic mesh network or a dynamic ad-hoc network is the most appropriate network topology for this situation. However, high node mobility causes the configuration of the network topology to constantly change. This constant change of configuration increases the difficulty for routing and data transfer as previously used routes become meaningless. Thus, there was a need for constantly updating proactive routing protocol to handle the bulk of the routing within the dynamic mesh. Better Approach To Mobile Ad-hoc

Network advanced (BATMAN adv) is one proactive routing protocol that is commonly used as a basis for routing and data transfer within a mesh network.

This project aims to ensure productive communications and data transfer between UAV's and/or to other devices outside the associated wireless ad hoc mesh network, such as a computer. This is will be built upon inspiration from existing protocols of network configuration, such as B.A.T.M.A.N (Better Approach to Mobile Ad-Hoc Networks) Advanced, to create a configuration that is application specific to high mobility nodes, such as drones. One particular prior example of work with mesh networks was an experiment done by Benjamin Sliwa et al. [1] on the use of multiple relay nodes, a server node, and a mobile node and running routing tests with various routing protocols. This experiment used a vehicular ad-hoc network (VANET) and tests the performance of BATMAN V routing in comparison to other routing algorithms in this configuration. This experiment showed that BATMAN V was comparatively better than its counterparts in tests for packet delivery against speed, traffic load, and number of parallel bit streams. Our project aims to allow all the nodes to mobile, but the performance and experimentation on a single mobile node in the mesh network is a perfect foundation for our own experiments and design.

In this paper we will give:

1) A background on:
   - Wireless Mesh Networks (WMN)
   - Ad-hoc on demand Distance Vector (AODV)
   - Optimal Link State Routing
   - BABEL
   - BATMAN
2) The Project Design including:
   - Materials
   - Standards
   - Current Design
3) The Experimental Design including:
   - Testing
   - Metrics Used
   - Data
   - Results
4) Summary and Conclusion
5) Future Considerations for Design and Project

II. Background Information

A. *Wireless Mesh Networks/Dynamic Mesh Networks:*

Wireless Mesh Networks are a form of wireless ad-hoc networks which means that the network is not centralized in nature. Mesh networks tend to be strongly connected, which means that all nodes within the network are connected to each other in one hop. However, this is not a necessity as long as all nodes can reach every other node through some path and there is not a centralized node that connects to every node, like that of a star topology. Another important component is that mesh networks are self-healing and self-configuring which allows them to be much more fault tolerant [2]. The self-configuring and self-healing qualities allow for the nodes in the network to remain connected when a node within the mesh is disconnected or disappears from the network. Specially configured nodes that are called gateways and bridges are needed to facilitate communications between devices that are outside the mesh with devices that are in the network. A gateway is connected to some network, i.e. a local area network (LAN), and allows any device that is connected to the LAN to communicate with devices that are in the mesh. A bridge allows devices, that are outside the mesh network and not connected to networks that the gateways of the mesh are connected, to communicate with devices within the mesh network [3].

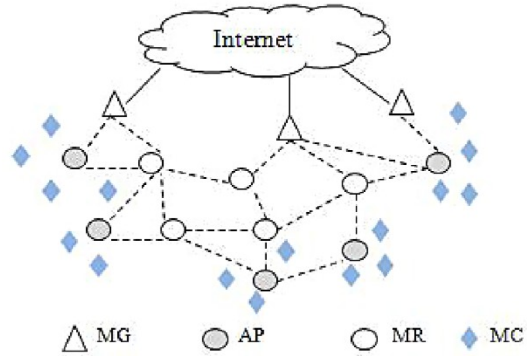

Fig. 1. Topological design of a mesh network (MG: Mesh Gateway, AP: Access Point, MR: Mesh Router, MC: Mesh Client) [3]

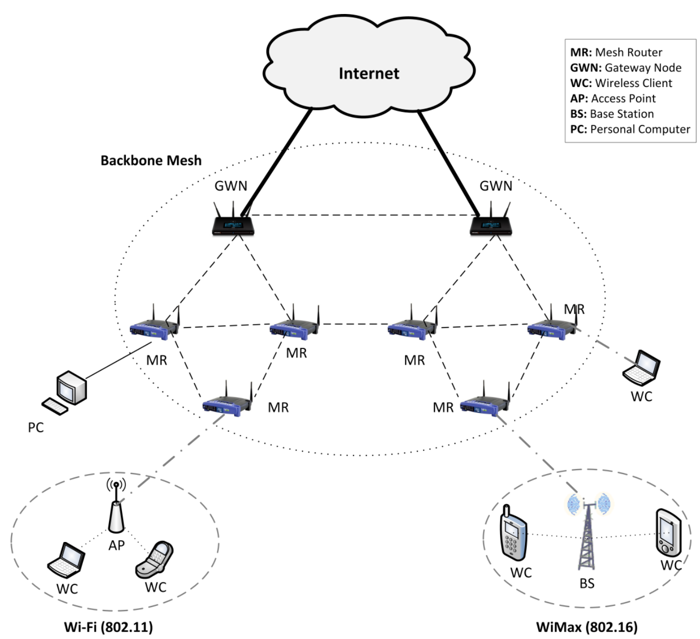

Fig. 2. More Visual Representation of Mesh Network

There are three main model for mesh networks: the infrastructure model, client model, and the hybrid model which is a combination of the infrastructure model and client model.

Infrastructure WMNs: This type of WMN forms a backbone of mesh routers that create a mesh of self-configuring, self-healing links that can be connected to the internet and other networks through routers configured as gateways or bridges. This allows clients outside the mesh network to communicate with the devices that are present within the router but only through the specific gateway or bridge routers. This is the most common type of mesh network that is used. Fig. 3 is a visual representation of this WMN model [2].

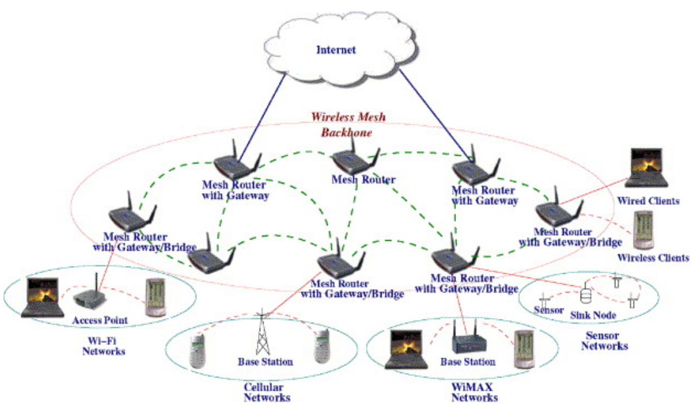

Fig. 3. Representation of Infrastructure Mesh[2]

Client WMNs: This type of WMN provides peer-to-peer (P2P) networking among the devices within the mesh network. As the client nodes facilitate the routing and configuration functionalities there is no requirement for mesh routers in this model. Packets that are transferred within this network tend to travel in multi-hop paths through multiple nodes within the network. Since Client WMNs provide a P2P network, the devices commonly use one type of radio to form the network. However, since there are no designated mesh routers in this network the end nodes are responsible for calculations for routing and configuration which in turn increases the requirements of these end nodes as compared to an infrastructure model. Fig. 4 is a visual representation of the client model [2].

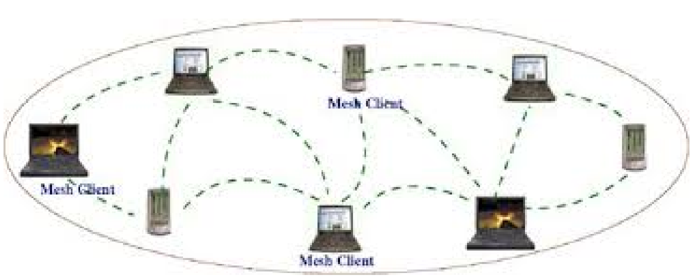

Fig. 4. Representation of Client Mesh[2]

Hybrid WMNs: This type of WMN is a combination of the infrastructure and client WMNs. Hybrid WMNs allow devices outside the mesh network to interact with devices inside the network by accessing a gateway or bridge node. However, other devices that are connected to a client that is connected to the mesh network can also use the client as a gateway and access devices in the network. The infrastructure of the hybrid model allows for connections to other networks such as Internet or Wi-Fi. Interestingly, the addition of clients and their routing capabilities provides improved connectivity and coverage within the network. Fig. 5 is a visual representation of the hybrid model[2].

Fig. 5. Representation of Hybrid Mesh[2]

Characteristics:

- **Efficiency:** WMNs have very minimal needs in terms of physical infrastructure. This allows for WMNs to be perfect for applications in developing countries or during natural disasters as a network created with minimal physical infrastructure can still be deployed or used after physical damage is done to most other infrastructure. The movement or changing of nodes and routing protocol that are implemented within the mesh network affect the overhead within the mesh network and can affect the efficiency of the network as a whole. How one approaches the splitting and transmission of data will also affect the efficiency of the network. For example, transmission over too many hops can lead to large delays and decreased throughput.
- **Limits of Usability:** The nodes within the mesh network and the protocols implemented are the main sources of limitations for a mesh network. The capacity or bandwidth capabilities and the memory that is available in each node is a limiting factor. These factors are fairly obvious as a node with more memory and better bandwidth is most likely better than one with less memory and bandwidth. Another limiting factor is that the mesh network needs a minimum number of nodes to function. This implies that under certain circumstances, it is not a viable solution if one does not have the requisite number of devices. Protocols and other algorithms implemented will also cause limitation due to inherent need for calculations and storage.
- **Fault Tolerance:** Since mesh networks are decentralized networks, there is not single point of failure within the mesh network. Some mesh networks use techniques such as flooding, broadcasting packets/information to every

single node within the network, to create redundancy and prevent loss. However, this does not mean that there is not point of failure that can occur. Within infrastructure and hybrid models, if there is a single gateway used, the gateway can be seen as a single point of failure. This is because it is the only link between outside devices and devices within the mesh causing it to be critical. Another interesting characteristic is that the mesh network is highly dependent on routing. This means that the efficiency and effectiveness of the network is dependent on the ability of the routing protocol to handle sudden issues such as a disconnected node.

B. Routing Protocols

There are many approaches to routing within networks which leads to a variety of solutions and methods. In this paper, we are concerned with the difference in performance between common reactive, algorithms that calculate routes after a request for transfer has been made, and proactive, algorithms that calculate routing table before requests, protocols in mesh networks and their usability and effectiveness in a network with high node mobility. With this in mind, we researched and compared the reactive solution AODV and three proactive solutions OSLR, BABEL and BATMAN.

Reactive: Reactive protocols wait until a request is sent before allowing the nodes to initiate a route discovery process. This means that the nodes do not need to constantly update their routing tables to predict the best routes within the network like proactive protocols do. This causes reactive protocols to have higher latency but lower overhead when compared to proactive protocols.

- **AODV:** A reactive protocol that discovers routes as necessary and does not maintain routes from every node to every other[5]. These routes are maintained for a certain period of time, if it is not used again it is considered expired and discarded. This minimizes the number of active routes between source and destination as well as stale routes. This helps with reducing the computations needed for route maintenance in the protocol. Routes are discovered in the network by flooding the network with a route request packet. The nodes maintain a list of precursor nodes for each destination, that should be routed through [4].

Proactive: Proactive protocols use various techniques to maintain necessary routing tables on every node within the mesh to represent the entire topology. This means that the route between nodes can be given immediately when requested but incurs higher overhead as the tables must be stored and maintained. The maintenance of the tables also forces the nodes within the mesh to need to constantly communicate to ensure that the topology of the network has not changed or check if there is a better path.

- **OSLR:** OLSR was developed for MANETs and is a table-driven proactive protocol[5]. The nodes select a set of neighbors to be multipoint relay (MPR). These MPRs are responsible for forwarding control traffic and are an efficient mechanism for flooding control traffic by reducing the number of required transmissions. OLSR provides the shortest path routes by using MPR nodes and their declarations of link-state information for their MPR selectors. MPRs periodically announce their information in control messages and the route is calculated by finding the nodes that are reachable from the various MPRs [6],[8].
- **BABEL:** Proactive routing based on distance vector approach to routing that is an evolution of the Expected Transmission count ETX algorithm [7]. This selects routes more intelligently than using single hop-count approach and has two distinctive characteristics that optimize relay performance. BABEL uses history-sensitive route selection to ensure that nodes do not continuously change preferred route between source and destination as it can lead to route instability. BABEL will prefer past routes over routes that have just been established when there is a choice. BABEL also executes a reactive update and forces a request for routing when a link failure is detected in the network [8].
- **BATMAN:** Proactive routing protocol that is more aligned to the minimal resources available in embedded hardware [8]. There is no central knowledge of routing information such that no node knows the topology of the network. They also do not know the routes to all other destinations, but only have a list of nearest neighbors as well as their best neighbor. Information is routed by forwarding through nodes and their subsequent best neighbor. BATMAN also prefers better links on the idea that better links provide faster and more reliable communication [12]. The best neighbor of a node is determined by how well the link is between the node and that neighbor. The protocol seems to be less complex than link-state calculation and has modest hardware requirements.

There are various factors that must be taken into account when deciding the type of solution to use for routing. For example, mesh networks with high number of nodes may find that a reactive solution is better than a proactive solution as there is a high amount of overhead in proactive routing tables when node counts are higher. For our project, we have a relatively low node count, between 5-10 nodes which is favors a proactive approach. Since BATMAN is built upon the logic of an OSLR approach and has many configurations that can be modified, we decided that BATMAN would be the best option to base our solution on.

C. B.A.T.M.A.N.

BATMAN was born out of a response to shortcomings that OLSR presented [9],[13]. As the number of nodes increased in the network, OLSR had a tendency to flush routing tables unnecessarily causing routes to regularly go up and down and routing loops due to out of sync routing tables. BATMAN combats these issues with the use of Originator Messages

(OGM) that are flooded throughout the network to notify other nodes within the mesh of the existence of a node and to also test the strength of the links. The OGMs are used in BATMAN's calculations of the best neighbor for each node [9],[10],[11]. The Transmission Quality (TQ) of a link is based on the number of received OGMs from different nodes to get the receive quality (RQ) and the number of their own OGMs that are resent by their neighbors or the echo quality (EQ). EQ is divided by RQ to get TQ and the higher the TQ the better quality the link is[10]. This allows the nodes within the mesh to know which link is the best and therefore know to which node it needs to propagate data when needed. The nodes within a mesh configured with BATMAN use TQ to find its best neighbor and maintains a routing table of these best neighbors and nearest neighbors. This means that none of the nodes within a BATMAN mesh know the topology of the network but can still propagate data in the correct direction. BATMAN also considers number of hops in the calculations by removing a set percentage for every hop from the TQ of a particular node. For example, every extra hop besides the first could incur a four percent (4%) penalty to the TQ. This helps prevent routes that are stronger in strength but have significantly larger hop counts [11],[12].

BATMAN was originally a layer routing protocol and used UDP to handle the exchange of routing information. However, BATMAN adv is a layer 2 routing protocol and increased efficiency by pushing most of the overhead calculations into the Linux kernel. This shift moved the routing protocol fro the network layer to the data link layer. This change cause a shift from the use of IP addresses to MAC addresses an allows the protocol to be easily portable between IPv4 an IPv6. This feature makes the protocol more lightweight an futureproof. Another advantage of moving routing to the dat link layer is that any desired transfer protocol such as TCP o UDP can be implemented on top of the current BATMA routing protocol, increasing the versatility of the protoco [11],[12].

BATMAN adv has default settings that run the device within the network on a single frequency channel and offe TDMA as the solution to communication. Based on the 802.1 standard, the frequency channels that would be used are 1, 6 or 11. However, BATMAN adv also allows for changes in the default configuration to allow multicast and the use of multiple channels instead of the default broadcast and single channel approach [11],[12].

BATMAN adv is a routing protocol that is extremely versatile in nature and allows for the tweaking of configurations to fit the needs of the user. This makes BATMAN adv the perfect foundation for our own application specific protocol to ensure data transmission in high mobility mesh networks.

III. Project Design

A. Materials

Hardware:

- Raspberry Pi 3b
- Microsd Cards (8, 32, 64 GB)
- HDMI Cables
- Ethernet Cable
- UAVs - Drones
- Router/Network Switch
- Computers

Software:

- Python 3.6.8
- BATMAN-adv (Better Approach to Mobile Ad-Hoc Network)

B. Standards

- IEEE 802.11: IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking
- Wi-Fi 2.4 GHz (IPv4)
- MicroSD
- USB 2.0 / HDMI 2.0
- TCP

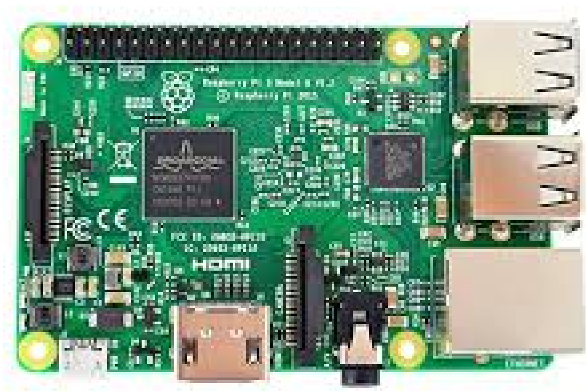

Fig. 6. Raspberry Pi Model 3b

C. Current Design

Nodes:

- 4 Raspberry Pi Model 3b
  - 1 Raspberry Pi Model 3b is the Gateway (GW)
- 1 Raspberry Pi Zero

The design of our project is a simulated UAV mesh network with Raspberry Pi's as the stand ins for real UAVs. Our final design proposes the use of around 5-7 UAVs that are coordinated in a swarm. We represent these UAVs with the Raspberry Pi's in the current design.

Five Raspberry Pi's have been configured to function as nodes in our mobile Ad-Hoc network. To relay messages and establish connection(s) between these nodes we utilized an open-source protocol known as B.A.T.M.A.N advanced to configure the Raspberry Pi's. One Raspberry Pi serves as the gateway that allows access between devices on the Wireless Network that the gateway is connected to via Ethernet and the nodes within the Mesh network. The remainder of the Raspberry Pi's behave as nodes to our Mobile Ad-Hoc Network. All these mesh nodes are connected to one another in the same fashion as the Mesh networks that were previously described. Figure 7 illustrates our group's tentative plans to use the B.A.T.M.A.N advanced protocol to simplify our process of propagating data between nodes connected to the same Raspberry Pi.

The Mesh Network that is currently in use is of the infrastructure model as we are trying to communicate between the mesh devices as well as allow an outside device to send commands. Fig. 8 shows an implementation of our mesh network with 3 nodes that was established for testing and proof of concept.

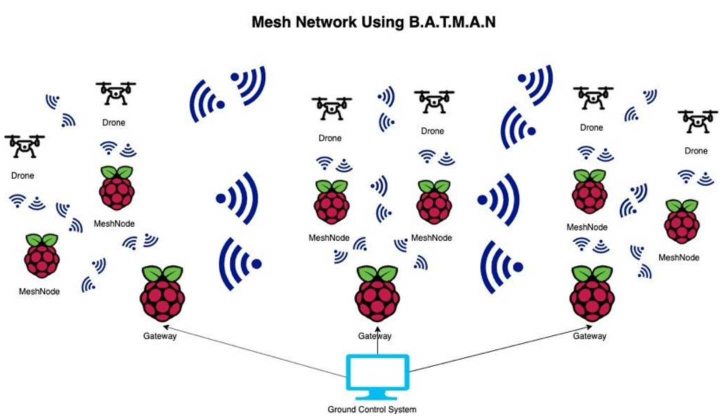

Fig. 7. Mesh Network Tentative Drone Plans

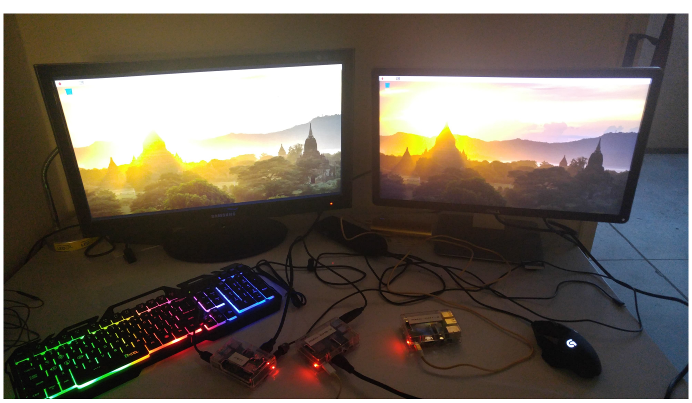

Fig. 8. Static Mesh Network Using 3 Nodes (Raspberry Pi)

IV. Experimental Design

A. Tests

1) Static or no Interference Node Bandwidth Test: The first test that was conducted was the using 2 nodes, one stationary the gateway and one mobile node. The gateway acted as the server and the mobile node was the client for the TCP Iperf test. The test was conducted by having a stationary GW and walking the mobile node away from the GW. An Iperf test was conducted at zero-distance from the gateway and in intervals of approximately ten steps. At every stop, a GPS coordinate was taken and an Iperf test was conducted. The test was conducted in a Line of Sight (LOS) manner and outside on an open field. The nodes were held above the ground to simulate drones off the ground and to decrease interference. These nodes were supposed to be static so the Raspberry Pi's were held as still as possible. Table 1 has the raw data for this test. Table 3 is the processed data for this test. Fig. 9 is a graphical representation of the processed data for this test. Fig 10 is a graphical representation of the processed data without the zero distance bandwidth to allow for more readability for the rest of the data in this first test.
2) Aerial Interference Bandwidth Test: The second test that was conducted used the same exact methods that were used in the first test. However, to simulate the interference that a drone would experience in flight we gently shook the Raspberry Pi's. Table 2 has the raw data for this test. Table 4 has the processed data for this test. Fig. 11 is a graphical representation of the processed data for this test.

B. Metrics Used

1) Distance: Distance in meters between two points; Calculated with a distance formula for two GPS coordinates. Distance calculated by gps-coordinates.org
2) Bandwidth: Kbits/sec Bandwidth report given by Iperf software tests

C. Data

Table 1 has the raw data for the static node test.
Table 3 is the processed data for the static node test.
Table 2 has the raw data for the aerial node test.
Table 4 has the processed data for the aerial node test.

TABLE I
Raw Data For Static Node Tests

Static Raw Data			
GPS Coordinates	Interval (s)	Size (MB)	Bandwidth (kbits/sec)
33.650119,-117.8340	0.0-10.2	23.9	19,700
33.650389,-117.8337	0.0-12.8	0.896	575
33.650543,-1178337	0.0-20.8	0.181	71.4
33.650670,-1178835	0.0-22.1	0.271	100
33.650663,-1178336	0.0-15.2	0.375	202
33.651008,-117.8333	0.0-390.2	0.0691	1.45

TABLE II
RAW DATA FOR AERIAL NODE TESTS

Aerial Raw Data			
GPS Coordinates	Duration (s)	Size (MB)	Bandwidth (kbits/sec)
33.650119,-117.8341	10.8	3.0	2320
33.650438,-117.8337	10.6	0.896	692
33.650319,-117.8337	11.0	1.380	1050
33.650565,-117.8337	10.9	2.880	2200
33.650657,-117.8336	10.5	0.640	502
33.650663,-117.8336	11.9	0.181	124
33.650872,-117.8334	608.7	0.102	1.38

TABLE III
PROCESSED DATA FOR STATIC NODE TESTS

Static Processed Data	
Distance (m)	Bandwidth (kbits/sec)
0	19700
45.61	575
58.33	71.4
74.56	202
80.92	100
130.43	1.45

D. Results

Fig. 9 is a graphical representation of the processed data for the static node test.

Fig 10 is a graphical representation of the processed data without the zero distance bandwidth to allow for more readability for the rest of the data in the static node test.

Fig. 11 is a graphical representation of the processed data for the aerial node test.

V. SUMMARY AND CONCLUSION

These tests were rather rudimentary and not the most accurate but from the collected data there are some conclusions that can be made. For example, the range of interest for the effective distance seems to be between 0 and approximately 50 to 60 meters for both the Static Node Test and the Aerial Node Test. There is a sharp decrease in the bandwidth of the node after the first 70 to 80 meters. From the graphs and data it appears that the network performed better during simulated aerial interference than for the static node. At the 130.43 meter mark, the results for the bandwidth were 1.45 kbits/sec and 1.38 kbits/sec for the static node test and aerial node test

TABLE IV
PROCESSED DATA FOR AERIAL NODE TESTS

Aerial Processed Data	
Distance (m)	Bandwidth (kbits/sec)
0	2320
40.91	1050
49.37	692
66.03	2200
73.05	502
74.56	124
130.43	1.38

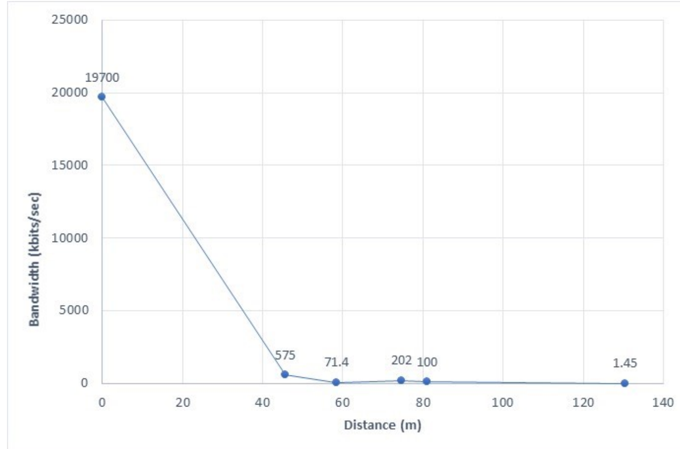

Fig. 9. Static Node Bandwidth Test

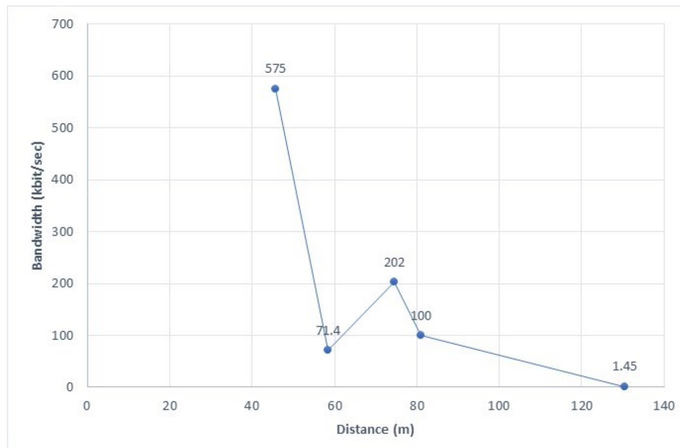

Fig. 10. Static Node Bandwidth Test without zero outlier to make the other data more readable.

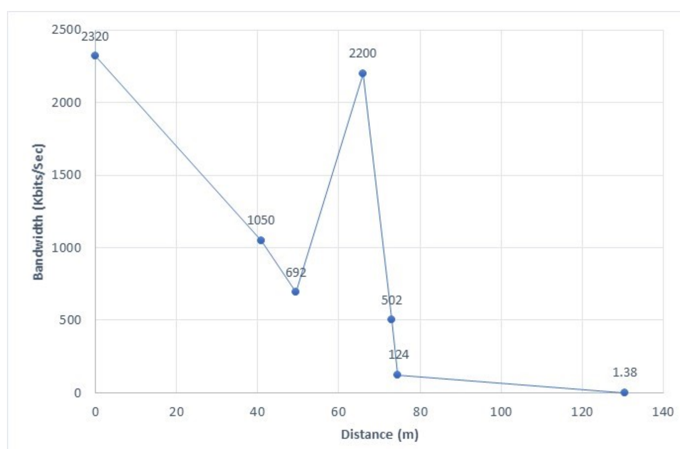

Fig. 11. Aerial Interference Node Bandwidth Tests

respectively. From this metric, we know that 130.43 meters is still within the one-hop distance for the BATMAN adv routing protocol but it is not necessarily a useful distance. This helps us to conclude that approximately 130 meters is the maximum range for one hop routing for BATMAN adv implemented on Raspberry Pi's without supplemental enhancers like a WiFi Antenna or Adapter.

In summary, this paper gave background information on WMNs and BATMAN adv which is the foundation for the current project design. The experimental design led to two TCP Iperf tests, one for a static or no interference and one for aerial interference. Each of the tests were conducted in a field in an LOS manner and the results were interesting. The maximum one hop range within our network has been proved to be approximately 130 meters in distance and under conditions of little physical interference such as walls. We are confident that the design for our simulated network is compatible and easily portable to an actual UAV network.

VI. Future Considerations

A. Integration with Hydra Lab

The collaboration with Prof. Levorato's laboratory will ensure that, once the network is functioning on the ground, we will move our experiments on the drones themselves. We will test different type of traffic and configurations to assure the deployment to be flexible for different applications. Such collaboration will also help us operating the drones in a safe manner.

Note that such study is original, since the current state of the art is not studying swarms with independent entities, but rather using broadcasting communications.

To evaluate the accomplishment of the project, we will prove that the autonomous swarm is able to set different formations given a visual input. For example, given a swarm of 5 UAVs equipped with embedded computers, and providing only one with a camera, we will stimulate the swarm with an input (such as a star) and observe the swarm propagate information and move accordingly to form the required shape. The mission described will require the network to support the information flow not only reliably, but also in a real time fashion. For example, the telemetry information of drones must be available to the others when planning their trajectories.

B. Testing

We understand that the current iteration of tests and experiments that were conducted were not the most effective and accurate. There are many improvements that we have considered for the current tests.

Improvements on Tests:

1) **Better Metric or Standard for Distance:** In our current iteration, we used GPS coordinates to measure distance. However, GPS coordinates have a relatively high margin of error, thus we decided to change this metric. Rather than the use of coordinates and calculation of distance using meters or something similar, we are considering the use of an object such as an ethernet cable or rope as the standard unit distance. Though it may not be as universal it will be more accurate and consistent. This metric can also be relatively easily converted to a metric such as meters if necessary.
2) **Use of WiFi adapters:** WiFi adapters can be used to enhance WiFi signals within the network as well as change the frequency from 2.4 GHz to 5.0 GHz. This will allow us to compare the difference between using 2.4 GHz and 5.0 GHz for our network. The signals can also potentially increase the effective range of the data transfer in the network.
3) **More samples:** For our current iteration we only took one test calculation at each stop. For the next iteration of testing we plan to take multiple samples and use the average as a better metric. This can be enhanced with a Python script that takes many samples and returns an averaged result. This should lead to more accurate results and be more informative.

New Tests:

1) **Multi-Hop routes:** We will be moving on to multi-hop routes and converting a one-hop route into multi-hop routes if necessary. This will allow us to test our network for robustness and efficiency in terms of data transfer over larger distances and multiple hops.
2) **Different Data:** We will be moving on to testing with actual packets of useful data such as images or text to test the strength of the network. This will lead to the use of other metrics such as frame delay or packet loss and inform us on the reliability and effectiveness of the current network design.

Acknowledgment

This project is supported by the Henry Samueli School of Engineering and Donald Bren School of Information and Computer Sciences at the University of California, Irvine.

Special thanks to our Project Advisor, Associate Professor Marco Levorato, Davide Callegaro, and the HYDRA (Resilient Computation for Heterogeneous Autonomous Drone sYstems) Lab.

VII. Appendix

1) **Standards and Importance:** Standards: IEEE 802.11: IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking Wi-Fi 2.4GHz/5GHz, USB 2.0 / HDMI 2.0, TCP, Category 5 Ethernet cable, Standard 8 GB/32GB/64GB MicroSD cards, IPv4 IP address, Secure Shell (SSH). The Wifi protocol and all standards relevant to Wifi and networking are imperative to our project as we are trying to build on top of this a protocol for data propagation. We used these network standards because our nodes use Wifi on the 802.11 protocol using 2.4GHz

and 5GHz. We use ethernet cables to connect as a client to our mesh network as well as allow us to use SSH to monitor nodes within the mesh network. Our nodes are booted off of the microSD cards that act as their storage and boot drives. Our project is compliant with these protocols as they all run in standard mode when used in our project and all hardware is used as it was intended. Also all of these standard protocols are used in every day network devices and all hardware is easily accessible to most people making our project realistic as well as replicable.

2) **Constraints and Solutions:** Constraints we have run into during our project are availability and cost of materials and availability of large amounts of area. Our overall project is meant to work with unmanned aerial vehicles (UAVs), however we cannot afford UAVs comparable to how our project is supposed to work as well as being affordable. We currently can only work on Raspberry Pi computing systems and have humans walk around acting as nodes instead of real drones flying around each other. The other major constraint we've been facing is the availability of large amounts of land with little to no obstacles to obstruct line of sight to each node. Since our project has the ideal scenario have line of sight to all nodes we are trying to test within line of sight of all nodes. However there are few places accessible to us with large amounts of land with no obstructions. This has led to inconsistent metric testing causing a big delay in getting basic metrics of our current mesh network. We worked to solve this by just choosing the largest area with the least amount of obstructions however it is not an ideal amount of land with no obstructions.

3) **Security and Protections:** Our project, being largely wireless network based, warrants large amounts of software and network security. One such security risk would be what clients can connect to our mesh network and access the nodes and possibly other clients. To reduce this risk, the protocol we are using to connect the nodes and clients in our mesh network requires gateway nodes to announce clients accessible to the nodes in the network. This reduces the amount of possible ill intended connections since in order to send and receive data from the network the client must be approved and announced to the rest of the network. Another security risk is someone hacking into the nodes themselves. To reduce this risk we have set log ins for each node individually so if someone were to remotely try to access the node they would need to know the log in and password, even if they had access to the node itself they would need to log in to access the node's data and network.

References

[1] B. Sliwa, S. Falten, and C. Wietfeld, "Performance evaluation and optimization of B.A.T.M.A.N. v routing for aerial and ground-based mobile ad-hoc networks," IEEE Veh. Technol. Conf., vol. 2019-April, pp. 1–7, 2019.

[2] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks Journal 47 (4) (2005) 445–487

[3] Benyamina, D., Hafid, A., & Gendreau, M. (2011). Wireless mesh networks design—A survey. IEEE Communications surveys & tutorials, 14(2), 299-310.

[4] John Hopkins University. Ad hoc On Demand Distance Vector (AODV) Routing Protocol. [Online]. Available: https://www.cs.jhu.edu/ cs647/aodv.pdf

[5] K. Holter, "Comparing AODV and OLSR," Comparing AODV and OLSR, 28-Apr-2005. [Online]. Available: https://folk.uio.no/kenneho/studies/essay/essay.html.

[6] Clausen, Thomas Heide & Jacquet, Philippe& Adjih, Cedric & Laouiti, Anis & Minet, Pascale & Muhlethaler, Paul & Qayyum, Amir & Viennot, Laurent. (2003). Optimized link state routing protocol (OLSR).

[7] V. Vesel´y, V. Rek, and O. Rysav´y, "Babel routing protocol for omnet++ more than just a new simulation module for INET framework," CoRR, vol. abs/1609.05215, 2016.[Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[8] M. Abolhasan, B. Hagelstein and J. C. -. Wang, "Real-world performance of current proactive multi-hop mesh protocols," 2009 15th Asia-Pacific Conference on Communications, Shanghai, 2009, pp. 44-47. doi: 10.1109/APCC.2009.5375690

[9] D. Johnson, N. Ntlatlapa, C. Aichel, "Simple pragmatic approach to mesh routing using BATMAN," 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, Oct 2008, pp. 1–10

[10] D. Seither, A. Konig, M. Hollick, "Routing performance of wireless mesh networks: A practical evaluation of BATMAN advanced", Proc. IEEE 36th Conf. Local Comput. Netw., pp. 897-904, Oct. 2011.

[11] A. Neumann, C. Aichele, M. Lindner, S. Wunderlich, "Better approach to mobile ad-hoc networking (B.A.T.M.A.N.)", IETF Draft, 2008.

[12] Open-Mesh.net, "B.A.T.M.A.N. (better approach to mobile ad-hoc networking)." [Online]. Available: http://www.open-mesh.net/

[13] J. Pojda, A. Wolff, M. Sbeiti and C. Wietfeld, "Performance analysis of mesh routing protocols for UAV swarming applications," 2011 8th International Symposium on Wireless Communication Systems, Aachen, 2011, pp. 317-321. doi: 10.1109/ISWCS.2011.6125375