# SCALABILITY OF MESH NETWORK TELEMETRY FOR SWARMS OF UNMANNED AERIAL SYSTEMS

Shawn Keshmiri[1], Dustin Hauptman[2], Daksh Shukla[1], Aaron Blevins[1]
[1]Department of Aerospace Engineering
[2]Department of Electrical Engineering & Computer Science
University of Kansas
Lawrence, KS, 66045
keshmiri@ku.edu, dhauptman.dh@gmail.com, daksh@ku.edu, aarontblevins@yahoo.com

## ABSTRACT

Swarms of autonomous unmanned aerial systems (UASs) are becoming increasingly popular as efficient replacement for manned aircraft. The major component that makes the swarm of UASs possible is an efficient exchange of aircrafts states (e.g. position & velocity) for all agents and the ground station. Advanced communication technologies are required to be implemented on each agent to enable real-time communication at high frequencies (e.g. 20 Hz) to avoid inter collisions and holding formations. To assess mesh network limitations and to identify bottlenecks, a series of simulations are carried out using actual hardware that is used for swarms of UASs, which are: (1) Amount of bandwidth that can be guaranteed given the communication system being used (XBee-900HP), each plane that the KU team uses, transmits 127 variables, 4 bytes each, at 20 Hz which means each plane needs 10 KBps and the mesh network might be able to support 53 UASs theoretically (2) Range limitations (3) Latency issues.

## INTRODUCTION

Autonomous swarms of small UASs are becoming increasingly popular among the research community as efficient remote sensing platforms that can effectively scan a desired field [1]. They can also be used for missions such as reconnaissance and strike, precision agriculture, etc. The availability of small embedded computational platforms and other hardware make it feasible to design a swarm of UASs. Implementation of complex nonlinear Guidance, Navigation and Control (GNC) algorithms becomes practical on parallel computing platforms and graphical processing units (GPUs). The major component that makes the swarm of UASs possible is an efficient exchange of position and velocity data for each agent amongst themselves and the ground station. For this, advanced communication technologies are required to be implemented on each agent to enable real time communication at high frequencies such as 20 Hz. However, such a system is inherently susceptible to unprecedented issues like intra and inter agent interference, latency, range, and limited baud rate.

There are several limitations to consider when dealing with telecommunications within a swarm.

The first limitation is the amount of bandwidth that can be guaranteed given the communication system being used. Each plane that the KU team is using is transmitting 127 variables, each consisting of 4 bytes, at 20 Hz. Calculating the minimum bandwidth per plane per second leaves us with each plane needing 10 KBps. Depending on the frequency of the communication system (900 Mhz, 2.4 Ghz, etc.) this bandwidth requirement could potentially be met very quickly. For example the KU team currently uses the Digi XBee-PRO 900HP, which has an upper bandwidth limit of 200 Kbps (up to 4 miles) [2]. In that system, only 2 planes could be fully supported. To potentially address this, the number of variables could be reduced. For instance if the number of variables was changed to 6 (only position and velocity information), then the total bandwidth per plane would drop down to 480 Bps. This, in theory, would allow us to support $\sim 53$ planes.

The second limitation is the range. One potential swarm formation is one where a few planes are outside the communication range of the ground station because of the chosen frequency for the communication medium. In this case the planes will need to be able to dynamically change their information routing path, and transmit through another plane to the ground station. One could also switch between different frequencies to establish long range communication as the swarm drifts apart.

A third potential problem is latency. Specifically, there are three problems within latency: The processors capability to read and write information, the speed of light, and the speed at which the transmitters and receivers can read information to and from the processor. The first and the third problems would be the main issues to address when dealing with long range communication in a swarm that needs to maintain a constant real-time frequency of 20 Hz.

## RELATED WORK

There has been quite a bit of work on mesh networks. Such as [3, 4] which both evaluated some form of a 802.11 mesh network. However, our network consists of XBee Pro S3B on a Digimesh network. There has been some research on the performance of the XBee devices when meshed together [5, 6], but not when in a moving formation or under the time demands of a real time system. Other papers have focused more on the technical aspect of mesh networking such as the effectiveness of RTS/CTS [7] or even the power consumption of different protocols [8].

## EVALUATION

One of the main concerns when flying in a swarm is knowing if the information being transmitted is being received. If it is not, then any action an individual plane in the swarm takes will have some risk. Therefore it becomes of utmost importance to evaluate how much information is being dropped on the network. It also becomes important to maintain the real time requirements of the system. In order to evaluate these two requirements the latency and packet loss between a sender and receiver was measured. The way this was measured was through a simple program that sent a ”Ping” with a timestamp and waited for a ”Pong” that contained the same timestamp. It should be noted this isn’t the ICMP Ping, but a custom program. The timestamp was then subtracted from the current time and the difference was considered the latency. There were problems (as in the sender stopped sending) when the responder didn’t send back a message, due to corruption or dropped packets, so there was a timeout set to allow another ping to be sent after 100 ms.

### *A. HARDWARE IN THE LOOP SETUP*

In order to conduct close to reality and practical telemetry tests that perform data packets' exchange in real time, wireless over the mesh network, a Hardware in The Loop (HiTL) setup is built to perform aircraft six-degrees-of-freedom (6DOF) simulations. This setup essentially consists of multiple aircraft avionics and the ground control station (GCS) laptop as the major simulator component end devices. The aircraft avionics is equipped with a computational platform (Odroid XU4) that runs all the autopilot software (Guidance, Navigation & Control) onboard in a ROS (Robot Operating System) framework. It also consists of aircraft 6DOF nonlinear equations of motion that emulate aircraft states in real time. The aircraft state information generated by 6DOF equations is continuously sent wireless through a XBee module both to the GCS computer and any other participating agents. Among the aircraft agents, only GPS (Global Positioning System) position (North, East, Down - NED) and velocity (NED) information are exchanged in order to perform a collaborative swarm flight. The details of hardware and avionics used, and a complete description of swarm technology and the details of the GNC software running onboard the aircraft can be found in reference [9].

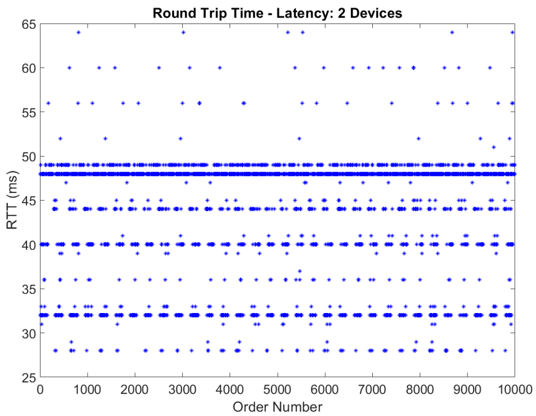
### *B. TESTS*

There were three tests conducted. The first test was to mimic the default settings that were present on the current swarm source code setup. This test included setting the stop bits to 1, waiting for only 1 byte in the read buffer before reading, and enabling up to 1 retransmissions within the network. The second test was simply seeing the performance impact when setting the number of stop bits to 2, and retransmissions to 0. The third test kept the settings of the second test as well as increased the minimum number of bytes to read from 1 to 17, which was the smallest the ping message would be.

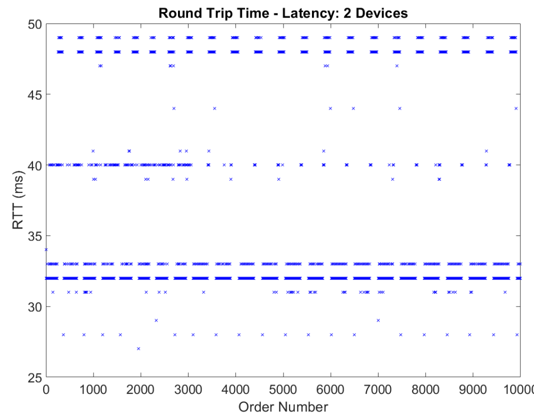
These three tests were then conducted under two different environments. The first was a single sender and single receiver, two devices total. The other environment was with a single sender and three receivers, four devices total. A quick note, a sender refers to the device that sends the first ping while the receivers are the ones waiting for the ping.

There was an additional of three tests performed with three Xbee devices. The first being a stationary test for a baseline. The second and third were tests where either the two receivers moved or the sender moved. These tests were performed using 2 stop bits, a minimum read of 1 byte, and 0 retransmissions. The first moving test had the receivers switch positions at the same time, over 60 meters. The second moving test involved moving the sender to receiver 1 and then to receiver 2. Receiver 1 was at 38 meters from the sender, while receiver 2 was at 26 meters initially. After these tests, the distance at which the devices were communicating, was measured.

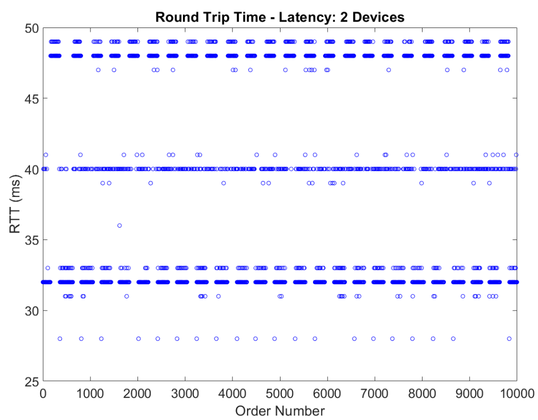
Finally a test was performed under HiTL simulation to simulate the actual data being transfered and decoded in the swarm. In this test, no new information was transmitted with the default messages, but instead one of the messages was hijacked and the "Ping-Pong" messages were transfered using that message.




(a) Experiment 1




(b) Experiment 2




(c) Experiment 3

Figure 1: Round Trip Times - 2 Devices

### *C. HARDWARE*

The tests were run using five XBee-Pro S3B 900 MHz RF modules. Each module was connected to a separate device for the tests. There were in total six devices used for testing, a Dell Latitude 3550, a Dell Latitude 5580, a Dell Studio 1735, a Dell Inspiron 17, a HP Pavilion, and a single Odroid XU4.

## RESULTS

Figure 1a contains the latency of the first test where the default settings of the swarm were mimicked using only two devices (Latitude 5580 and Latitude 3550). Figure 1b contains the latency for when the number of stop bits is set to 2 and the retransmission value is set to 0. Figure 1c is an further extension of 1b to see what would occur when the minimum read size was increased from 1 byte to 17 bytes. The packet loss for the same experiments can be seen in Table 1a, Table 1b, and Table 2 respectively.

| | Pings | Pongs |
|---|---|---|
| Sent | 10000 | 9134 |
| Received | 9134 | 8186 |
| Loss | 8.66% | 10.38% |

| | Pings | Pongs |
|---|---|---|
| Sent | 10000 | 9539 |
| Received | 9539 | 9127 |
| Loss | 4.61% | 4.32% |

Table 1: (a) Packet Loss of Sender and Receiver for Two Devices with 1 Stop Bit, a Minimum Read of 1 byte, and a Retransmission value of 1. (b) Packet Loss of Sender and Receiver for Two Devices with 2 Stop Bits, a Minimum Read of 1 byte, and a Retransmission value of 0

| | Pings | Pongs |
|---|---|---|
| Sent | 10000 | 9214 |
| Received | 9214 | 9102 |
| Loss | 7.86% | 1.22% |

Table 2: Packet Loss of Sender and Receiver for Two Devices with 2 Stop Bits, a Minimum Read of 17 bytes, and a Retransmission value of 0

The next experiment simply increased the number of devices on the network from 2 to 4 (Latitude 5580, Latitude 3550, Studio 1735, and Inspiron 17) to see what the affects of an increased throughput and noise would be. The latency can be seen in Figure 2 with packet loss information in Tables 3 and 4.

The next three experiments were to simply see if the affect of moving within close proximity of each other would cause any dramatic changes in the latency or packet loss. The initial stationary test can be seen in Figure 2d while the two moving tests can be seen in 3. The devices used were the Latitude 5580, Latitude 3350, and Inspiron 17.

The final test was integrating the code into an HiTL simulation of the swarm and testing the latency of messages that are being processed. This was using the Odroid, Inspiron 17, and Latitude 3550.

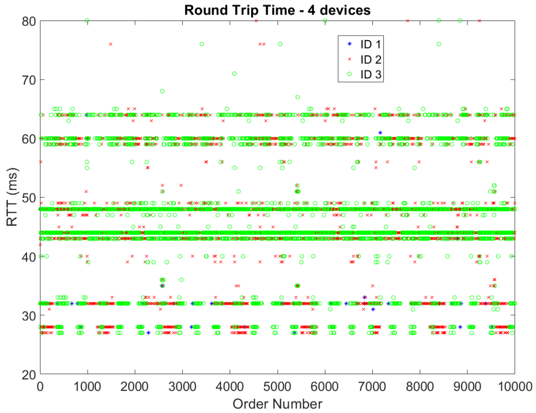
## DISCUSSION

It is interesting to note that all the data appears to have specific latencies that the packets conform to. In Figure 1 the packets appear to conform to $32ms \sim 48ms$. However, during testing the latency measurements were varying between these intervals ($32ms \sim 48ms$) on a fairly regular basis. There is no particular interpretation or reasoning behind this finding. Therefore, further test-

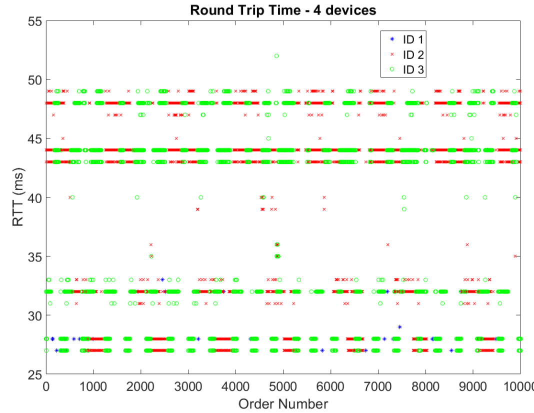
| ID | Pings Received | Loss |
|---|---|---|
| 1 | 136 | 98.64% |
| 2 | 8589 | 14.11% |
| 3 | 8613 | 13.87% |

| ID | Pongs Sent | Received | Loss |
|---|---|---|---|
| 1 | 136 | 83 | 38.971% |
| 2 | 8589 | 5504 | 38.918% |
| 3 | 8613 | 5512 | 36.004% |

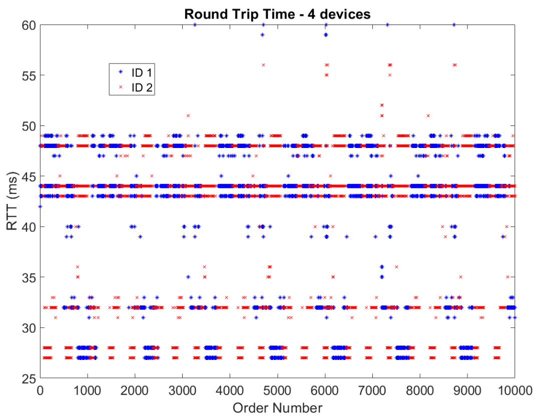
Table 3: Total Sent Pings = 10000. Packet Loss of Sender and Receivers for Four Devices with 1 Stop Bit, a Minimum Read of 1 byte, and a Retransmission value of 1



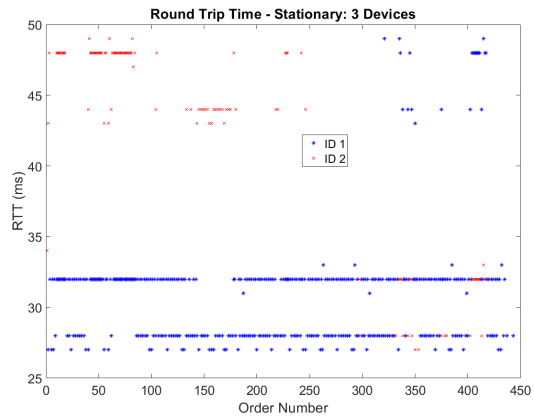

(a) Experiment 1




(b) Experiment 2




(c) Experiment 3




(d) Round Trip Times - 3 Stationary Devices

Figure 2: Round Trip Times - 4 Devices (a,b,c) and 3 Devices (d)

| ID | Pings Received | Loss |
|---|---|---|
| 1 | 50 | 99.5% |
| 2 | 8597 | 14.03% |
| 3 | 8261 | 17.39% |

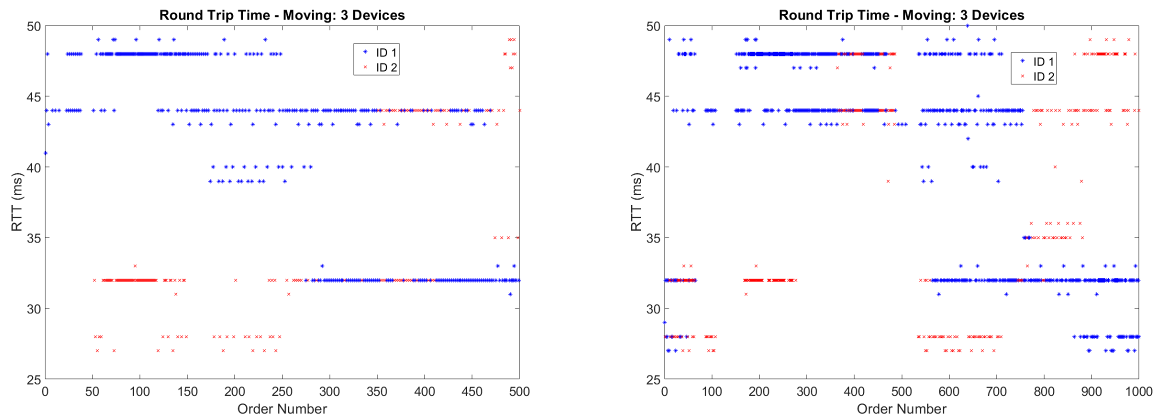
| ID | Pongs Sent | Received | Loss |
|---|---|---|---|
| 1 | 50 | 32 | 36% |
| 2 | 8597 | 5723 | 33.43% |
| 3 | 8261 | 5306 | 35.77% |

Table 4: Total Sent Pings = 10000. Packet Loss of Sender and Receivers for Four Devices with 2 Stop Bits, a Minimum Read of 1 byte, and a Retransmission value of 0

| ID | Pings Received | Loss |
|---|---|---|
| 1 | 471 | 5.8% |
| 2 | 150 | 70% |

| ID | Pongs Sent | Received | Loss |
|---|---|---|---|
| 1 | 471 | 398 | 15.5% |
| 2 | 150 | 113 | 24.67% |

Table 5: Total Sent Pings = 500. Packet Loss of Sender and Receivers for Three Devices during Stationary Test




(a) Experiment 1 (b) Experiment 2

Figure 3: Round Trip Times - 3 Moving Devices

| ID | Pings Received | Loss |
|---|---|---|
| 1 | 458 | 8.4% |
| 2 | 265 | 47% |

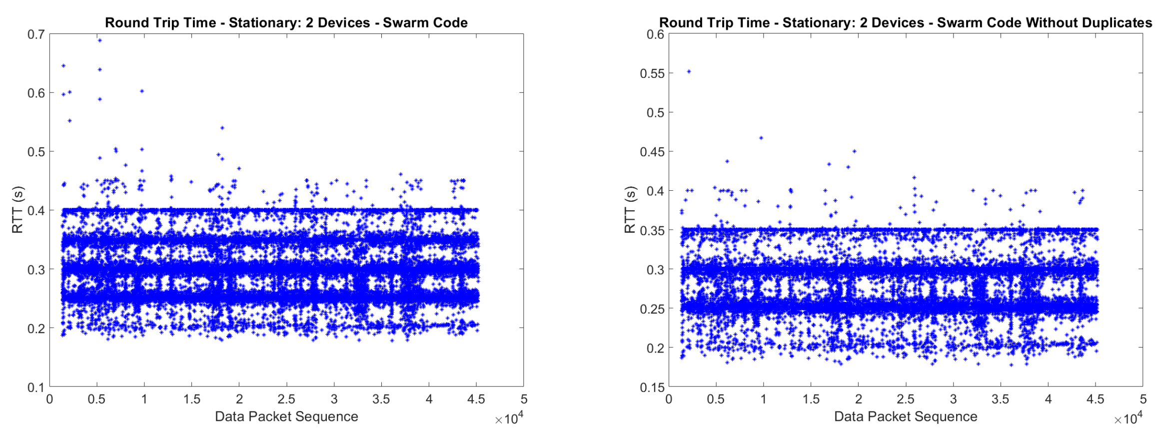
| ID | Pongs Sent | Received | Loss |
|---|---|---|---|
| 1 | 458 | 370 | 21.44% |
| 2 | 265 | 172 | 35.09% |

Table 6: Total Sent Pings = 500. Packet Loss of Sender and Receivers for Three Devices during First Moving Test

| ID | Pings Received | Loss |
|---|---|---|
| 1 | 906 | 9.4% |
| 2 | 484 | 51.6% |

| ID | Pongs Sent | Received | Loss |
|---|---|---|---|
| 1 | 906 | 679 | 25.06% |
| 2 | 484 | 334 | 30.99% |

Table 7: Total Sent Pings = 1000. Packet Loss of Sender and Receivers for Three Devices during Stationary Test




(a) With Duplicate Data Packets (b) Without Duplicate Packets

Figure 4: Round Trip Times - 2 Stationary Devices Exchanging Data During HiTL Simulations

ing was conducted where a spectrum analyzer was hooked up to the Xbee antennas and waveforms were monitored around the 900 MHz band. It was determined that when the packets would start to switch from one latency to another, the RF module appeared to be turning off all, but $2-4$ of its channels. During the switching period, either there were a few packets that were successfully transmitted or none whatsoever. This seems to be a characteristic of the mesh networking protocol and should be analyzed more extensively in order to ensure robust communication.

Another crucial aspect to point out is that in Figure 2c there was a $100\%$ packet loss to one of the devices. However, after later experiments it was discovered that this was due to the computational hardware used for experimentation. In the experiments, three modern Dell laptops (both Latitudes and one Inspiron) along with an older Dell laptop (Studio 1735) were used. The more modern laptops were able to perform the task relatively well, but the older laptop could not. Isolated experiments on the Studio 1735 laptop, without any interference from other devices, showed that the data packets' loss was immensely high. This loss was replicated on the HP Pavilion as well. A fix for this could be that all packets would have to be some standard size, and the minimum read would be fixed to this size as well. However, exchange of data packets with dynamic sizing is more pragmatic to the swarm communication system requirements, as there are a handful of messages of varying sizes that can be parsed at any moment by an end device.

During the moving tests it was noticed that one of the devices (ID 2 in Figure 4) was performing below the expectations. It was found that this occurred due to simply a loose USB connection to the Xbee device. After tightening the connection, the packet loss dropped from $\sim 70\%$ to $\sim 50\%$. In a UAV, vibrations are to be expected, so it is possible that the advantage of a "Plug and Play" device could be lost. A potential solution could be to use soldered XBee USB connectors to be placed in a drone.

After performing the moving tests, the range at which the Xbee devices could communicate and reliably transmit and receive data, was investigated. Using a laser range finder and GPS data in these experiments, the effective range was found to be approximately 750 meters. The 750 meters range limit was determined by taking one device down a road until the sender wasn't receiving any replies.

One of the final analysis is for the HiTL simulations in Figure 4. In these two figures it can be seen that the typical packet latency is between $4 \sim 8$ times higher as the real time requirement. One of the explanations for this is that the program which is being currently used in the system, requires that an entire message be read, and then processed byte by byte in a for loop. This could be sped up by simply reducing the amount of data being transfered (if the for loop is necessary) or simply the processing code needs to be reevaluated.

## CONCLUSIONS

In this paper we measured and report our initial findings on the latency delays and packet loss in a host of different settings, that can be potentially utilized in designing robust, efficient and reliable communication systems for swarms of flying unmanned aerial systems. We showed that the XBee modules can perform under certain real time constraints when needed. However, the additional overhead of decoding the data seems to be the main bottleneck. In the future, actual swarm flights will be conducted while recording data communication statistics, experiments involving distance to latency measurements at distances that exceed 300 meters will be performed, and a higher number of UAVs will be integrated into the swarm framework. Additionally, the size of data can be reduced to the minimum amount needed to successfully fly in a swarm, more measurements can be added (such as monitoring the number of bytes transfered), and code can be improved to try and eliminate the processing bottleneck. This work analyzes various communication statistics in detail collected by performing different types of simulations, that are pertinent to multi-agent collaborative systems moving at high speeds with substantial electromagnetic interference within the avionics systems of the agents.

## REFERENCES


[1] C. Leuschen, R. Hale, S. Keshmiri, J. Yan, F. Rodriguez-Morales, A. Mahmood, and S. Gogineni, "Uas-based radar sounding of the polar ice sheets," *IEEE Geoscience and Remote Sensing Magazine*, vol. 2, no. 1, pp. 8–17, 2014.

[2] Digi International Inc., Global Headquarters Digi International 11001 Bren Road East Minnetonka, MN 55343, *XBee-PRO 900HP/XSC RF Modules S3 and S3B - User Guide*, n ed., June 2014.

[3] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11 b mesh network," in *Proceedings of the 11th annual international conference on Mobile computing and networking*, pp. 31–42, ACM, 2005.

[4] A. Sheth, S. Nedevschi, R. Patra, S. Surana, E. Brewer, and L. Subramanian, "Packet loss characterization in wifi-based long distance networks," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pp. 312–320, May 2007.

[5] A. Kumar, P. G. Namboothiri, S. Deshpande, S. Vidhyadharan, K. M. Sivalingam, and S. S. Murty, "Testbed based throughput analysis in a wireless sensor network," in *Communications (NCC), 2012 National Conference on*, pp. 1–5, IEEE, 2012.

[6] V. Mayalarp, N. Limpaswadpaisarn, T. Poombansao, and S. Kittipiyakul, "Wireless mesh networking with xbee," in *2nd ECTI-Conference on Application Research and Development (ECTI-CARD 2010), Pattaya, Chonburi, Thailand*, pp. 10–12, 2010.

[7] K. Xu, M. Gerla, and S. Bae, "Effectiveness of rts/cts handshake in ieee 802.11 based ad hoc networks," *Ad hoc networks*, vol. 1, no. 1, pp. 107–123, 2003.

[8] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pp. 46–51, Ieee, 2007.

[9] S. Keshmiri, A. R. Kim, A. Blevins, D. Shukla, and E. Mark, "Control of multi-agent collaborative fixed-wing uass in unstructured environment," in *Unmanned Aircraft Systems (ICUAS), 2018 International Conference on*, IEEE, 2018.