Networking Optimization for IoT Mesh Networks

Manvik Nanda
mbnanda2@illinois.edu

Gerard Matthew
gerardm3@illinois.edu

Abstract

Rural communities are affected by the ability to obtain internet connectivity at a lower cost. Internet providers are not incentivised to invest in extending their network for financial reasons. In the presence of disasters, mobile providers focus on maximizing their profits by delaying repair of critical infrastructure used to provide connectivity. At the end, the rural communities are left with either no connectivity or very expensive internet connectivity. The goal of this project is to better understand wireless mesh networking and how it could be used to solve the internet connectivity problem of rural communities. An evaluation of options to build a wireless mesh network was done. The wireless mesh network was built using 6 Raspberry PI nodes implementing BATMAN protocol in an outdoor environment. The network was observed to achieve up to 12 Mbps on the best link with re-routing occuring about 60 seconds after the introduction of packet loss.

1 Introduction

Traditional wireless networks rely on a centralized infrastructure to enable connectivity between the nodes which are connected on the network. The centralized infrastructure introduces a resilience problem where there is a single point of failure within the network. The nodes participating in the network are required to be within range of the base station in order to connect to the network. The base stations are typically fixed while users of the network may be mobile. The expansion of such a network is difficult to deploy as the base stations need to be provided with their own connection in order to expand the network. For this reason, there has been research done to solve some of these problems of the traditional network.

Mobile Ad-Hoc networks were built to address some of the known challenges with the traditional wireless networks. There is no longer a need for fixed nodes within the network, nor is there the need for direct communication to a base station. Instead, the network is constructed of mobile nodes which are able to speak directly to each other. This allows the network to be easily deployed and adaptive to changes in the network topology with the mobility of nodes. One drawback to mobile ad hoc networks is that they were not built to provide internet connectivity. The use cases for this type of network were related to military use. Therefore, a lot of pioneering of this type of network was done by the Defense Advanced Research Projects Agency (DARPA).

The Mobile Ad-Hoc networks had their own set of challenges. These included challenges related to quality of service, throughput, energy and routing to name a few. As depicted in Figure 1, the nodes within the network are mobile. With mobile nodes, it is important to keep track of the link state in order to make the best routing decisions at any point in time. The quality of service obtained from the network is dependent on how well the network is able to keep track of the link states. Similarly, with mobility, the throughput of the network is no longer consistent. Nodes are able to move closer to each other while using the same channel to communicate. The use of the same channel in close proximity will introduce interference and degrade the throughput of the network. An additional challenge is energy management. As the nodes are mobile, they operate on battery and need to continuously be listening for communication. With limited power, the nodes are able to be on the network for a limited duration of time.

Another critical challenge with this type of network is routing. The traditional routing algorithms like distance vector do not work well in this network when the mobility rate is very high and when there are a high number of nodes in the network. Therefore, to solve some of the routing challenges it is important to take a closer look at the set of metrics which should be used. In a subsequent section of this report, we will present a set of the metrics which were evaluated as part of the project.

While Mobile Ad-Hoc networks were great for use in the military, there was a similar need for such a type of network to address challenges with the civilian population. Specifically it raised the question of "How can learnings from mobile ad-hoc networks be used to solve civilian related problems?" As mentioned above, the traditional networks are difficult to deploy and costly to scale. For some of these reasons, wireless mesh networks were created to apply the concepts from mobile ad-hoc networks which are able to address civilian related challenges such as connectivity in rural communities.

A wireless mesh network is constructed of various components in a decentralized manner. The decentralized nature of wireless mesh networks allow them to be more resilient to failure and easier to scale and deploy. From Fig 2, there are multiple mesh routers which are interconnected to create a "mesh backbone". Some of the mesh routers are able to connect to the internet and are designated as internet gateway routers. The mesh clients are able to connect to the network and subsequently to the internet by connecting to a mesh router. The mesh routers are able to be located at varying distances from each other and are mostly static. In order to obtain good throughput on the network, it is necessary to route packets on the best links while being able to be adaptive to failures in the network paths. In a subsequent section, we will discuss some of the different types of metrics which have been evaluated in literature.

In this project, our focus was on building a wireless mesh network to understand how they operate in order to solve some of the challenges faced in rural communities.

2 Motivation

Rural communities are at the mercy of internet and mobile service providers for connectivity. These communities are low in population and do not provide a profitable opportunity for the service providers. As internet service providers operate a business denture, they are not incentivized to improve connectivity in these communities. Therefore, these communities are left with more expensive options such as the use of mobile data plans in order to obtain access to basic internet connectivity.

This issue has been observed in small island territories within the Caribbean when disasters such as hurricanes affect pre-existing wired infrastructure. In the aftermath of these natural disasters, the internet service providers who are also the mobile service providers refuse to repair the existing land-line infrastructure which provides internet connectivity to homes. This lack of connectivity thus forces the communities to utilize the more expensive data plans for their mobile devices to connect with their friends and family. With the existing use of mobile devices which also have the ability to connect to a WiFi network, it is practical to deploy a wireless mesh network to solve this problem in rural communities.

While it is agreed that the use of a wireless mesh network is a viable option to solve the problem, there needs to be a better understanding of how routing would occur within a wireless mesh network when deployed in a large rural area. There are numerous metrics which are used to determine the best route for packets within a mesh network. As a mesh network is adaptive to faults, the routing decisions need to be made within a relatively short time frame in order to reliable communication within the network.

The following a few of the metrics that can be used to perform routing decisions:

1. Packet Loss Rate: Packet Loss Rate measures the percentage of lost packets during transmission. A low packet loss rate is advantageous for reliable communication, prompting nodes to select paths with minimal losses to enhance overall network performance. However, the drawback lies in the potential for decreased performance and reliability when selecting paths solely based on this metric, as other factors affecting network efficiency may be overlooked.
2. Throughput: Throughput signifies the actual data transfer rate between nodes, a critical factor for applications requiring high data rates. Routing decisions based on throughput aim to optimize data transmission efficiency, ensuring timely and effective communication. Nevertheless, focusing solely on throughput may neglect considerations of latency or other performance metrics, potentially leading to suboptimal routing decisions.
3. Round Trip Time (RTT): Round Trip Time [8] measures the time for a data packet to travel from source to destination and back. Selecting the path with the least RTT optimizes factors like link load and bandwidth. While easy to implement, RTT-based decisions may be susceptible to self-interference caused by queuing, impacting the overall reliability of the chosen route.
4. Packer Pair: In Packet Pair [8], nodes send two consecutive probes to neighbors, and the path is chosen based on the least sum of delays measured by these neighbors. This approach excels in accounting for various factors but comes at the cost of increased computational complexity and resource utilization, making it more expensive than methods relying on RTT measurements.
5. Expected Transmission Count (ETX): ETX [8] represents the expected number of transmissions for a packet to be successfully received. Lower ETX values indicate more reliable links, making it valuable for wireless mesh networks. However, relying solely on ETX may oversimplify the evaluation of link reliability, potentially ignoring other critical factors influencing overall network performance.
6. Expected Transmission Time (ETT): ETT improves on ETX by considering differences in link transmission rates and incorporating throughput. While capturing the impact of link capacity on path performance, ETT falls short in fully addressing intra-flow and interflow interference, limiting its ability to provide a comprehensive solution to network optimization challenges.
7. Weighted Cumulative ETT (WCETT): WCETT [9] enhances ETT by incorporating intra-flow interference considerations. It evaluates end-to-end delay and channel diversity, striking a balance between throughput and delay for route optimization. However, a key limitation is the uncertainty about its computability in polynomial time, potentially impacting its feasibility for practical implementation in certain network environments.

Each of the metrics presented above has its own strengths and weaknesses. For the project, the focus was to utilize a preexisting protocol which relied on packet loss and throughput as the metric to help in routing decisions. The following sections outline the protocol used during the experiment in detail.

2.1 Better Approach to Mobile Ad-hoc Networking (BATMAN)

A wireless mesh network can be built using systems which communicate via the wireless medium. In this project, we utilized Raspberry Pi to build the mesh network using the BATMAN protocol. The first goal of the project was to enable connectivity between all the nodes in the wireless mesh network. The BATMAN protocol utilizes two routing algorithms, BATMAN v4 [6] and BATMAN v5 [7]. The aim of creating the wireless mesh network using BATMAN was to allow us to understand what each algorithm is good at and where there may be challenges. The overall goal was to ensure that this protocol could be effectively used for reliable data transmission in rural areas.

BATMAN serves as a vital component in our project. It is an open-source routing protocol designed for networks where devices communicate with each other without a fixed structure. We selected BATMAN because it provides flexibility and its code can be adjusted to meet our specific needs. In addition, as it is open-sourced, in the event there are adjustments to be made in routing, there would be the opportunity to perform such modifications. This adaptability is crucial as we aim to create a solution tailored to work effectively in rural areas. Due to the limited time allocated in the semester for the project, we saw it best to spend time understanding how to build a wireless mesh network by learning from an existing protocol instead of creating a new protocol.

BATMAN operates by each node continuously broadcasting originator messages (OGM) to its neighbors. The neighbors processes the OGM, updates the address to its own address, then rebroadcasts the message to its neighbors while keeping the sender as the original sender. When a node needs to deliver a message to another node, it first looks at its routing table to determine the best link to send the packet to. In the original version of BATMAN, the count of the number of OGMs to the destination from a link was used to indicate the link quality. However, the link quality has been enhanced in BATMAN v4 to take into consideration packet loss on the link in both receiving and transmitting. Similarly, in BATMAN v5, throughput was used instead of only packet loss for situations where there are links with the same quality but varying throughput capabilities.

3 Related Work

There is pre-existing work in the area of wireless mesh networking. This section covers the related work where BATMAN was used for the wireless mesh networks.

There is pre-existing work in the area of wireless mesh networking. This section covers the related work where BATMAN was used for the wireless mesh networks.

As discussed in Kiran, K., et al. [1], BATMAN and BATMAN-Adv differ in the layer (ISO/OSI model) in which they are implemented, with BATMAN in layer 3 and BATMAN-Adv in layer 2. The paper explores the impact of this differentiation on their performance and stability. The evaluation is conducted using a testbed with mobile nodes, and the results reveal that BATMAN-Adv, implemented in layer 2, provides significant improvements in throughput, especially in scenarios with mobile nodes. However, when considering stability in terms of jitter and packet loss, BATMAN performs well in many scenarios compared to BATMAN-Adv. The paper concludes that BATMAN-Adv offers better throughput and faster routing optimization, especially in mobile scenarios, but the stability comparison is inconclusive. The study used iperf3 as a tool for performance measurement and analyzed the routing path in real-time using batmand and batctl for BATMAN and BATMAN-Adv, respectively. The authors suggest future work involving larger networks and exploring the impact of nodes going offline on network performance. Using this as an inspiration, we conduct our tests in an uncontrolled outdoor environment.

Meanwhile, Liu, Ligang, et al. [2] discusses the performance evaluation of BATMAN-adv wireless mesh network routing algorithms, focusing on its new variant, version V, and comparing it with version IV. The study aims to analyze the performance of version V, which uses a throughput-based metric, in comparison to the well-established version IV, which employs a transmit quality (TQ) mechanism. The authors conduct tests in two scenarios: one to evaluate packet loss and delay, and the other to assess route update performance. In scenario 1, version V exhibits higher packet loss rates and delays compared to version IV. In scenario 2, which focuses on route updating, version IV demonstrates better average throughput and route recovery, while version V shows degraded throughput performance and struggles with route recovery. The analysis suggests that the main reason for the performance disparity lies in the high algorithm cost of version V. The transmission of management packets, particularly Echo Location Protocol (ELP) packets for neighbor discovery and metric data collection, consumes significant airtime, leading to a decrease in effective bandwidth. Comparison with BATMAN V will remain as a future scope of our study, although our findings indicate that re-routing in BATMAN IV is probably not very efficient as it takes a lot of time for re-routing to take place. This suggests that the time interval for OGM's might need to be decreased.

Chissungo, Edmundo, et al. [3] investigates the performance of the Batman-adv protocol on an indoor Mesh Potato (MP) testbed. The Mesh Potato is a device used for voice communications over a wireless medium, supporting data as well. The study aims to understand the network performance of MPs by measuring delay, packet loss, jitter, and throughput. Experiments involve packets of varying sizes over multiple hops, and the data is analyzed to check if the network latency for up to four hops adheres to ITU-Recommendation G.114. The impact of varying packet sizes on network performance is also explored. Results are presented and discussed, covering packet loss ratio, delay/latency, jitter, and throughput for both 73-byte and 1500-byte packets across different hop scenarios. The findings suggest that while delay values are within acceptable limits for VoIP, packet loss and jitter increase significantly with more than two hops. Throughput decreases sharply after two hops, indicating potential limitations in the network's performance. The conclusion emphasizes the need for further exploration of network anomalies and suggests future work, including running experiments on different floors to validate results. As stated in the paper, conclusive results can only be drawn by testing the performance of BATMAN IV, V on different floors with varying number of hops.

The focus is on providing connectivity to under-serviced rural areas using low-cost wireless devices and mesh networking technology. As part of a case study in [4], the authors present pilot deployments of wireless networks in Peebles Valley, South Africa, and Macha, Zambia, highlighting the social observations, challenges faced, and future directions for development. Key findings include the importance of local champions, skills development, and the need for innovative solutions such as an "ISP in a box," IPv6 adoption, real-time flow over mesh networks, and energy-efficient routing. The paper emphasized the potential of wireless mesh networking to empower disadvantaged communities in developing regions. Thus, it would be very important to also keep in mind the non-technical challenges that will be encountered when actually deploying a mesh network based on the BATMAN routing algorithm to a rural area.

A lot of inspiration was drawn from a very interesting case study [5]. The wireless mesh network (WMN), using LocustWorld mesh nodes, provided the community with internet service, surpassing urban offerings. The deployment faced challenges such as unexpected behavior, blind spots, and signal propagation issues. The network used IEEE 802.11b technology with the Ad-Hoc On-Demand Distance Vector (AODV) protocol for routing. The nodes were strategically placed, and adjustments were made to improve coverage. The positive impact this has had on the local community has been immense and in summary, the deployment of the WMN demonstrated the viability of mesh networks in rural settings, offering a real-world testbed for research.

Finally, it was very interesting to see how VillageNet built a low cost omnidirectional mesh network [10]. The paper primarily discusses challenges in rural areas with significant populations but constrained financial resources. A lot of focus was given to designing low cost antennas and a cost function was introduced, considering two types: masts for heights less than 20 meters and steel towers for greater heights. Thus, deploying a solution requires careful consideration beyond the technology employed.

4 Approach

In this section, we present the approach which was used to build the wireless mesh network using Raspberry PIs. Below is a description of each planned step of the experiment.

1. Set up the wireless mesh network so that each node is able to communicate with each other.
2. Introduce packet loss to observe how the network reacts to the poor link in its routing decisions.
3. Repeat 1 and 2 for each version of BATMAN.

4.1 Node Configuration

A python script was written to configure BATMAN protocol on each Raspberry Pi. The script performed the following functions:

1. Install batctl tool - 'sudo apt-get install -y batctl'
2. Configure the batman virtual interface
3. Configure the wireless lan interface to channel 6 and mesh-network SSID
4. Assign static ip address to each node
5. Load BATMAN kernel module
6. Disable DHCP for wlan0
7. Configure the DNS

The function below handles tasks 1-4.

```
def create_start_batman_adv_script():
    script_content = '''#!/bin/sh
    # batman-adv interface to use
    sudo ifconfig bat0 down
    sudo batctl if add wlan0
    sudo ifconfig bat0 mtu 1468

    # Tell batman-adv this is a gateway client
    sudo batctl gw_mode client

    # Activates batman-adv interfaces
    sudo ifconfig wlan0 up
    sudo ifconfig bat0 up

    #assign a static IP
    sudo ifconfig bat0 192.168.199.11 netmask
        ↪ 255.255.255.0
    sudo route add default gw 192.168.199.1
    sudo iwconfig wlan0 channel 6 essid mesh-
        ↪ network mode ad-hoc
    '''
    with open(os.path.expanduser(dir + 'start-
        ↪ batman-adv.sh'), 'w') as f:
        f.write(script_content)
    run_command('chmod␣+x␣' + dir + 'start-
        ↪ batman-adv.sh')
```

This function generates a shell script (start-batman-adv.sh) with commands to configure and activate batman-adv interfaces, set up a gateway client, assign static IP addresses, and configure wireless network parameters such as the channel and SSID.

The function below handles tasks 5-7.

```
def configure_batman_adv_module():
    run_command('echo␣"batman-adv"␣|␣sudo␣tee␣--
        ↪ append␣/etc/modules')
```

This function adds the "batman-adv" module to the list of kernel modules, ensuring it is loaded at boot.

```
def configure_dhcpcd():
    run_command('echo␣"denyinterfaces␣wlan0"␣|␣
        ↪ sudo␣tee␣--append␣/etc/dhcpcd.conf')
```

This function adds a configuration to dhcpcd to deny automatic DHCP configuration for the wlan0 interface.

```
def setup_nameserver():
    run_command('echo␣"nameserver␣192.168.199.1"
        ↪ ␣|␣sudo␣tee␣--append␣/etc/resolv.c
```

This function adds a nameserver configuration to the /etc/resolv.conf file so that when internet connectivity is available, the nodes are able to resolve the hostnames correctly.

With all of the above functions defined, the

```
if __name__ == "__main__":
    install_batctl()
    create_start_batman_adv_script()
    configure_batman_adv_module()
    setup_nameserver()

    print("All␣configurations␣done.␣You␣can␣now␣
        ↪ reboot␣the␣Pi.")
```

The main block calls the functions in sequence to perform the setup which automates the setup process for configuring a Raspberry Pi to operate within the network.

5 Experiment

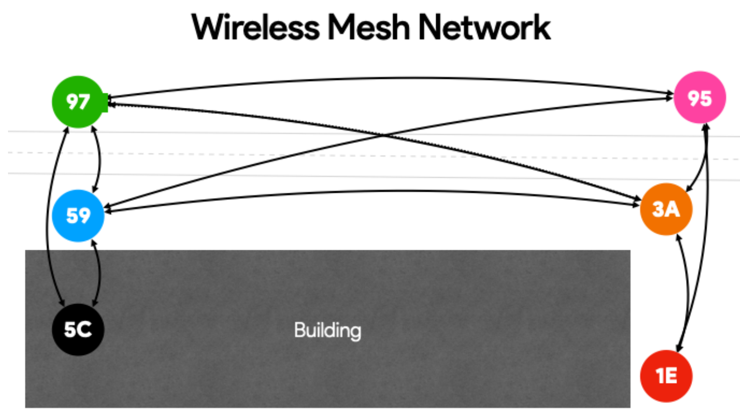

Figure 1. Experiment Setup with 6 Nodes

A total of 6 nodes were used during the experiment. The nodes were arranged in the form as illustrated in the figure. The aim of the experiment was to measure the throughput between green node (97) to the red (1E)

At the start of the experiment, node 97 communicated to node 1E through node 3A. At the 68 second mark of the experiment, packet loss was introduced on the 97 to 3A link for 90 seconds. After 90 seconds the link was reinstated without packet loss. This setup allows us to study how the network adapts during challenges and reverts to normalcy after interference.

A small script was written which introduced the packet loss, slept for 90 seconds then removed the packet loss on the link.

```
sudo tc qdisc add dev wlan0 root netem loss 75%
```

This line adds a network emulation (netem) queueing discipline (qdisc) to the wlan0 interface. The netem module is used for network emulation, and in this case, it introduces packet loss. The loss 75

```
sleep 90
```

This line pauses the script for 90 seconds, simulating a period of packet loss.

```
sudo tc qdisc del dev wlan0 root
```

After the 90-second delay, this line removes the previously added netem qdisc from the wlan0 interface, effectively stopping the simulated packet loss

6 Evaluation

The first evaluation of the network was to understand the furthest distance that two nodes were able to communicate in an uncontrolled environment. It was observed that the two nodes were able to communicate up-to a distance of 61.2 meters. Although at this distance, the throughput values were 150KB/sec, hence a major degradation. This experiment was important to understand the placement of nodes, going forward in our research.

The second evaluation of the network was to understand how well BATMAN v4 reacted to a link which started experiencing packet loss. This evaluation was done using a widely-used network performance tool called iperf. In order to perform the tests, the iperf server was required to be running on the target node.

This was done using the following command. In this command there are two flags used -s and -u.

```
iperf -s -u &
```

Below is a description of each flag used for the server.

- **-s**: This flag indicates that iperf should run in server mode. The server waits for clients to connect and measures the performance of the network between the server and the client.
- **-u**: This flag specifies that the server should use UDP instead of the default TCP protocol. UDP is a connectionless protocol that is often used for real-time applications where a small amount of data loss is acceptable.

As the server was running on the target node, to run the test a client was started from the testing node using the following command.

```
iperf -c server_ip -i 2 -t 300 -u -b 1000m | tee
    ↪ log_file.log
```

Below is a description of each flag used for the client.

- **-c**: The "-c" flag designates the command as a client, and "SERVER_IP" is a placeholder for the actual IP address or hostname of the iperf server to which the client will connect.
- **-i**: The "-i" flag sets the interval for periodic bandwidth reports. In this case, the interval is set to 2 seconds, meaning that iperf will provide a bandwidth report every 2 seconds during the test.
- **-t**: The "-t" flag specifies the duration of the test in seconds. In this example, the test will run for 300 seconds (5 minutes).
- **-b**: The "-b" flag sets the bandwidth for the test. In this case, it is set to 1000m, indicating a bandwidth of 1000 megabits per second. This was done to avoid the default bandwidth of 1Mb/s set by iperf.
- `| tee`: The "|" symbol is a pipe, which redirects the output of the iperf command to the "tee" command. "tee" is a Unix command that reads from standard input and writes to both standard output and one or more files. In this case, it writes the output to a file named "log_file.log".

Below are the results obtained from the tests conducted while introducing packet loss on one of the links within the mesh network.

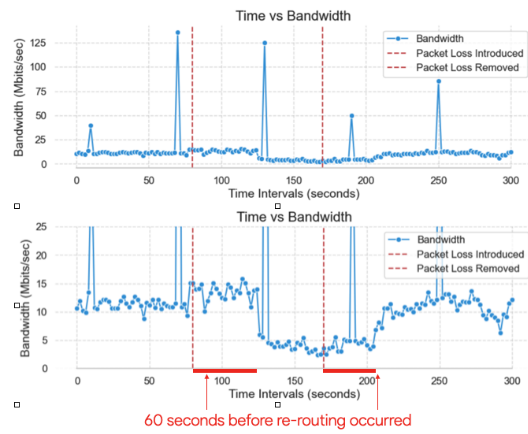

Figure 2. Experiment Setup with 6 Nodes

It is observed that the bandwidth decreased after the introduction of packet loss; however, the bandwidth decrease occurred approximately one minute after the packet loss introduction. The introduction of packet loss shifted the traffic to a healthier route. The delay in the shift is due to the sliding window of metric collection which is used in the routing algorithm of BATMAN. The number of Originator Messages (OGM) received along a path slowly decayed over time as the link became degraded. This however would have taken time to decrease below the threshold of the other nodes. OGMs are emitted by each node every second, and hence the delay in time that was observed. Similar observation was noticed when the packet loss was removed from the link.

The major challenge in this experiment was finding the right placement for the nodes where there is reduced interference, but sufficient distance where direct communication is not possible while still maintaining a high throughput. This challenge was mitigated by intentionally positioning the target node in an area where there was sufficient obstruction where direct communication would be the least efficient route.

7 Future Work

In order to better understand the existing wireless mesh network systems, there are additional evaluations that we would like to do in the future. Below is a list of work that we see should be done in the future to help us better understand the state of wireless mesh networks and how they operate in a rural environment.

1. **Comparison with BATMAN v5**: In the future, this work can be expanded to do a comparative analysis on the performance of BATMAN v5, which measures performance using throughput. This comparison would

help understand the strengths and weaknesses of both BATMAN v4 and BATMAN v5 while providing insights into future potential enhancements. By doing this, we can figure out which protocol works better in different real-world situations.
2. **Long-Range Antennas**: Addition of long-range antennas, this upgrade would allow our mesh network to cover even greater distances. The idea is to see how well our network performs over long distances, especially in rural areas where people are spread out. This step is crucial for making our mesh network effective and reliable in diverse geographic settings.
3. **Machine Learning for Smarter Decisions**: Looking ahead, we're exploring the use of machine learning techniques to help our network make smarter decisions. This means the network could adapt in real-time, avoiding paths that usually have low bandwidth. By doing this, we aim to make our network more intelligent and capable of optimizing itself. It's all about using advanced technology to solve challenges and improve connectivity in rural areas.
4. **Energy-Efficiency Optimization**: Explore ways to optimize the energy efficiency of the mesh network nodes. This could involve developing algorithms or protocols that dynamically adjust power consumption based on network demand, helping extend the battery life of devices in areas with unreliable power sources.
5. **Community Engagement and Training**: Extend the project scope to include community engagement and training initiatives. Work on creating educational materials and conducting training sessions to empower local communities in rural areas to understand, maintain, and contribute to the sustainability of the mesh network.

8 Conclusion

A number of wireless mesh network metrics were evaluated to gain a better understanding of the impact of routing. To better understand how wireless networks operate, a wireless mesh network was built using Raspberry Pi nodes with BATMAN as the routing protocol. During the experiment, an evaluation of the impact of packet loss and recovery while using BATMAN v4 was presented. From the results, the network was observed to achieve up to 12 Mbps on the best link with re-routing occurring about 60 seconds after the introduction of packet loss and with the final throughput values never reaching the initial values before the introduction of packet loss. These observations lead us to believe there are optimizations within BATMAN protocol which can be achieved in order to be more reactive to degraded links. Nonetheless, based on the results, the use of a wireless mesh network can be utilized in a rural environment to enable connectivity to communities.

9 References

1. Kiran, K., et al. "Experimental evaluation of BATMAN and BATMAN-Adv routing protocols in a mobile testbed." TENCON 2018-2018 IEEE Region 10 Conference. IEEE, 2018.
2. Liu, Ligang, et al. "Performance evaluation of BATMAN-adv wireless mesh network routing algorithms." 2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom). IEEE, 2018.
3. Chissungo, Edmundo, Edwin Blake, and Hanh Le. "Investigation into batman-adv protocol performance in an indoor mesh potato testbed." 2011 Third International Conference on Intelligent Networking and Collaborative Systems. IEEE, 2011.
4. Johnson, David Lloyd, and Kobus Roux. "Building rural wireless networks: Lessons learnt and future directions." Proceedings of the 2008 ACM workshop on Wireless networks and systems for developing regions. 2008.
5. Ishmael, Johnathan, et al. "Deploying rural community wireless mesh networks." IEEE Internet Computing 12.4 (2008): 22-29.
6. https://www.open-mesh.org/doc/batman-adv/BATMAN_IV.html
7. https://www.open-mesh.org/projects/batman-adv/wiki/BATMAN_V
8. Draves, Richard, Jitendra Padhye, and Brian Zill. "Comparison of routing metrics for static multi-hop wireless networks." ACM SIGCOMM Computer Communication Review 34.4 (2004): 133-144.
9. Draves, Richard, Jitendra Padhye, and Brian Zill. "Routing in multi-radio, multi-hop wireless mesh networks." Proceedings of the 10th annual international conference on Mobile computing and networking. 2004.
10. Dutta, Partha, Sharad Jaiswal, and Rajeev Rastogi. "VillageNet: A low-cost, IEEE 802.11-based mesh network for connecting rural areas." Bell Labs Technical Journal 12.2 (2007): 119-131.