Abstract— In this paper, we describe the performance of the B.A.T.M.A.N advanced (Batman-adv) protocol on an indoor Mesh Potato (MP) testbed. The MPs are small devices used for voice communications over the wireless medium but also supports data. The Batman-adv protocol is designed for ad hoc wireless networks. We measure delay and packet loss, jitter and throughput in order to understand the MPs network performance. The experiments used packets of varying sizes over multiple hops. We analyze the data to see if the network latency for up to four hops is within the recommended boundaries set by ITU-Recommendation G. 114. We also observe how the network’s performance is affected by the varying packet sizes. Finally the experiments also reveal the common issues found on the wireless medium and also indoor testbeds.

Keywords-component; Batman-adv; Testbed

I. INTRODUCTION

Wireless nodes in ad hoc wireless mesh networks lack the capability for communicating with nodes not directly connected to it. Due to a limited communication range routing protocols exist as a mechanism to overcome this problem and thus are in charge of performing data forwarding between nodes helping to form an ad hoc network. There exists an abundant number of routing protocols [1] each fitting into a pre-existing taxonomy. However even with so many protocols one has still to be developed that is better than all others in all aspects. Previous works into one protocol called Better Approach to Mobile ad hoc Networking (B.A.T.M.A.N or Batman) suggests that “Batman is the panacea that community wireless mesh networks have been waiting for” [2].

Batman-adv is an open-source wireless routing protocol and is the predominant implementation of B.A.T.M.A.N routing algorithm as it used extensively as the routing protocol in a wireless communication device called the Mesh Potato (MP) [3]. The MP is used as an alternative communication device for communities. MPs use Voice over IP (VoIP) to allow users to wirelessly make calls between connected nodes on the ad hoc wireless network. The Batman-adv protocol’s performance on the device (MP) has yet to measure. This would be useful as it would give us valuable insight into the real-world performance of this protocols when used as a solution for community wireless networks. Here we present a practical insight into a real-world performance of Batman-adv.

This work is structured as follows: we start in Section 2 with the background on the protocols. This is followed by Section 3 which describes the experiment set up. The results section is follows in Section 4. Finally conclusions are drawn in Section 5.

II. RELATED WORK

An overview of the Batman-adv routing protocol is presented next. This is followed by an overview of the MP and finally the relevant literature.

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

Batman is a simple and robust algorithm for establishing multi-hop routes in ad hoc networks [4]. As explained by Johnson, D., et al [2] Batman does not maintain the full route to the destination, each node along the route only maintains the information about the next link through which the node can find the best route. The objective is to maximize the probability of delivering a message. Batman does not attempt to check the quality of each the link, it just checks its existence. The protocol does these checks by having all nodes periodically broadcasts hello packets to its neighbours, these packets are known as originator messages (OGM). Broadcasting is when a single source sends messages to all available nodes in the broadcast domain/network. This is in contrast to unicast where a node sends messages to one specific node in the network.

The structure of the OGM packet periodically sent is here presented:

- originator address
- sending node address: this is changed by receiving nodes and then the packet is re-broadcasted
- unique sequence number: The sequence number is used to check the concurrency of the message
- bidirectional link flag: used when the OGM packet received is its own and the sender is someone else
- time to leave (TTL)

When a node receives an OGM there are two possibilities, either the originator is or is not already in its routing table. If the originator is not in the routing table then a new entry is made for it and the sender node is added as a one hop neighbour to it and its count incremented. If the
Figure 1: shows the physical network topology for the test bed used in the experiments conducted in this work. On the far left (bottom left) and far right (bottom right) are the Unix boxes which generate and receive the network traffic. In between are the MP nodes that perform the routing. Each node ran the Batman-adv protocol. Each dotted line represented a hop in the network.
mimicked techniques described by P. Gunningberg, et al. [9] and B. Hagelstein, et al. [10]. The authors use techniques such as intentional attenuation of the signal level at each node in the testbed to enable some nodes to be out of range of others and thus creating multi-hop network topologies.

A. Physical Testbed

The physical testbed used in our experiments was achieved by deploying a MP network in the Computer Science building at the University of Cape Town (UCT). Figure 1 shows the connections achievable in the largest implementation of the MP testbed given the space available and signal propagation issues caused by the close proximity of the nodes. We used two Unix boxes and MPs all running the Batman-adv routing protocol. One Unix box was placed in the farthest room on the floor used in the building. This is shown on the far left of Figure 1 (bottom left corner) Node (0). In the opposite direction, we placed the second Unix box also in the farthest room. This is show of the farthest right of Figure 1 (bottom right corner) Node (6). In between these two Unix boxes are the MP Nodes (1–5). The MPs did all the routing on the network. The dotted lines in Figure 1, between the network nodes, represent the existing links between nodes. Each link (dotted line) represents a hop in the network. The Unix boxes generated and received the packet traffic on the network and are passive network nodes from a routing perspective.

B. Scenarios

The testbed was rolled out as need and eventually looked like Figure 1. Each of the hops included two Unix boxes and zero or more MPs placed in between the Unix machines as need to achieve the desired number of hops. This is shown on the Figures 2, 3 and 4 these were a few of the scenarios used in the experiments. We note that the one hop scenario does not use any MPs. The data gathered from it serve for comparison purposes with the scenarios that include the MPs.

![Figure 2: Scenario 1 (1Hop), one meter distance between the Unix boxes.](image)

![Figure 3: Scenario 2 (2Hop), one meter distance between the left Unix box and the MP. 15 Meters between the MP and the Unix boxes.](image)

C. Testing

The testing was conducted on the testbed matching the physical topology mentioned in Figure 1. In the testbed the Unix box nodes generate traffic in the form of data packets. We use packets of size 73 bytes and 1500 bytes, each representing voice and standard Ethernet packets respectively. In doing this we hoped to compare the performance of the network when dealing with voice and data packets sizes.

In each of the experiments conducted we varied the load which were packets generated and sent by the Unix box on the far left on Figure 1. We sent 1000 UDP packets of size 73 bytes, this was repeated 60 times, referred here as iteration. We then increased the packets size to 1500 bytes. We also iterated this 60 times as well. We did each of the experiments for each independent number of hops represented by the scenarios in Figures 2, 3 and 4. In each hop we observe how load and number of hops traversed affects each of the metrics chosen to be scrutinized. The chosen metrics are Throughput (Tp), Jitter (J), Packet Loss Ratio (PLR) and Delay (D). We believe that these are all essential metrics we need to analyse in order understand the performance of the Batman protocol.

IV. RESULTS

Next we discuss the results of the experiments described in Section 3.

A. Packet Loss Ratio

VoIP is not tolerant of packet loss to the extent that high packet loss can degrade the call quality. In VoIP, high packets loss will cause a call to break up, and too much of this will result in an incomprehensible conversation [11]. Table 1, below, shows the average percentage packet loss in each hop throughout the experiments.

Table 1, below, shows us what we expected to see, the larger the amount of hops traversed the higher the packet loss. The same idea is also true for packet sizes. Larger packet sizes can also generate higher packet losses. Larger packets are broken down into smaller chunks to be sent; therefore, larger packets have larger number of chunks to be sent which increase the probability of loss, aggravated by the increasing number of hops traversed.

Table 1 shows us that for 1500 byte packets the loss rate rises sharply from 0% with the first hop to 71% on the second hop and 84% on the fourth hop. The data suggests that perhaps the MP network is not well suited for services with large data packets such as Ethernet. The data collected shows us that there are less packet losses when the MP
network routes 73 byte sized packets then when it routes 1500 byte sized packets on all the hops. The data shows that packet loss affects all packet sizes at the fourth hop as 73 byte and 1500 byte packets experience 71% and 84% loss respectively. We can also point out that for the one hop scenario the packet losses are so low that the percentage packet loss experienced is virtually zero for both packet sizes. This can be attributed to the fact that the one hop scenario does not use any MP routers only really serves to compare MP networks to Unix machine network using the same protocol.

Finally, we note that there is a sharp rise in packet loss on the fourth hop for the 73 byte packets. This rise suggests that even for the smaller packet sizes communication hops higher than three hops are not suitable for the Batman-adv network.

### TABLE I. Average (Avg) and Standard Deviation (STD) for Packet loss

<table>
<thead>
<tr>
<th>Hop</th>
<th>Avg 73 Byte Data</th>
<th>STD for 73 Byte Data</th>
<th>Avg 1500 Byte Data</th>
<th>STD for 1500 Byte Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>8.179516</td>
<td>8.254515</td>
<td>71.2907</td>
<td>7.838142</td>
</tr>
<tr>
<td>3</td>
<td>17.019266</td>
<td>6.193529</td>
<td>73.624716</td>
<td>15.245518</td>
</tr>
<tr>
<td>4</td>
<td>37.115779</td>
<td>45.354644</td>
<td>84.481537</td>
<td>21.712366</td>
</tr>
</tbody>
</table>

After packet loss, delay is considered the "second most disruptive impairment in VoIP networks" [12] and we address delay on the MP testbed next.

### B. Delay/Latency

Delay is the time taken to transmit a packet from a source to a destination (one-way latency) in milliseconds (ms). The effects of delay to the caller generally appear as echo and talker overlap. Talker overlap occurs when the end-to-end delay between a packet transmission and reception is so great that one caller cuts off the speech of another caller due to excessive delay. Acceptable and unacceptable delay values for voice applications where established by the International Telecommunication Union G series (ITU-G) [13]. According to ITU-Recommendation G. 114 [14] delay values below 150ms are acceptable, values between 150ms and 400ms are acceptable provided callers are aware of the impairment. Values above 400ms are deemed unacceptable.

Table 2, below, shows the values we measured on our testbed. The values depict the expected effects of increasing number of hops and packet sizes on the network. Delay was expected to increase with the increasing hops and packets sizes simply because it takes longer to send more data over an increasing number of hops.

We found our delay values to fall within the ITU-Recommendation G. 114 boundaries for acceptable delay. Voice applications on the MP network seem well suited as even values at the fourth hop level are well within the acceptable range, having an average of 32ms at the fourth hop. Whether the delay on MP networks for other applications that may run on it are within acceptable boundaries is applications depended. Relevant tests will have to be carried out for those applications. In the case of Ethernet, the values are within the boundaries.

### TABLE II. Average (Avg) and Standard Deviation (STD) Delay/Latency for the 73 and 1500 Byte Packets

<table>
<thead>
<tr>
<th>Hop</th>
<th>Avg 73 Byte Data</th>
<th>STD for 73 Byte Data</th>
<th>Avg 1500 Byte Data</th>
<th>STD for 1500 Byte Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.889084</td>
<td>0.602789</td>
<td>3.2467451</td>
<td>1.479338</td>
</tr>
<tr>
<td>2</td>
<td>20.671616</td>
<td>50.097983</td>
<td>51.89275</td>
<td>39.684970</td>
</tr>
<tr>
<td>3</td>
<td>3.753710</td>
<td>50.3063</td>
<td>14.306119</td>
<td>73.59119</td>
</tr>
<tr>
<td>4</td>
<td>31.715779</td>
<td>96.655718</td>
<td>68.389355</td>
<td>16.599199</td>
</tr>
</tbody>
</table>

### C. Jitter

Jitter is defined as latency variations measured in milliseconds. Jitter is usually caused by queuing, contention and changes in the path through the network [11]. Jitter is particular important on network links supporting voice over IP (VoIP) because high jitter causes fluctuations in the call quality causing calls to be choppy and may even cause breaks in calls. Essentially the important jitter values recorded are for the 73 byte packets as those represent the voice packets used in the MP network.

Table 3, below, shows us the average jitter experienced in each hop for both the 73 byte and 1500 byte packets.

### TABLE III. Average (Avg) and Standard Deviation (STD) Jitter in Milliseconds for Hop 1 to 4 Over 60 Iterations for the 73 and 1500 Byte Packets

<table>
<thead>
<tr>
<th>Hop</th>
<th>Avg 73 Byte Data</th>
<th>STD for 73 Byte Data</th>
<th>Avg 1500 Byte Data</th>
<th>STD for 1500 Byte Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.16675</td>
<td>0.053383</td>
<td>0.4512</td>
<td>0.212506</td>
</tr>
<tr>
<td>2</td>
<td>4.11985</td>
<td>0.854593</td>
<td>59.7014</td>
<td>50.774851</td>
</tr>
<tr>
<td>3</td>
<td>6.563133</td>
<td>1.388631</td>
<td>129.886366</td>
<td>161.568123</td>
</tr>
<tr>
<td>4</td>
<td>199.34048</td>
<td>189.36153</td>
<td>260.625002</td>
<td>356.588547</td>
</tr>
</tbody>
</table>

### D. Throughput

Throughput is the average rate of successful messages/packets delivered over a communication channel per unit time. We measure throughput in bytes per second (bytes/sec).

This data is better understood when represented graphically as we have done on Figure 5 below. The X-axis represents the number of iterations of the experiment for each hop and the Y-axis shows the throughput values as bytes/sec. What we expected to see is the throughput decrease with each increasing hop packets have to traverse. We expect this because network bandwidth is essentially halved with each hop [12]. This is shown by the graph in Figure 5. The graph shows throughput decreasing with each hop. Hop 1 (blue line) has the highest throughput averaging 124587 bytes/sec. It is then followed by the hop 2 line (red line) which averaged 14325 bytes/sec, then hop 3 (green line) which averaged 1777 bytes/sec and finally hop 4 (purple line) which averaged 295 bytes/sec.
We note the extreme gap between the first hop data and the second hop data. This can be attributed to the fact that the one hop scenario is composed purely of Unix boxes. The one hop scenario serves as a comparison of the MP network and the Unix network when using the same protocol. Unix boxes performed better as they have more resource than the MPs.

The next graph shown in Figure 6, below shows us the data for the scenario where the network is handling and forwarding Ethernet sized packets. We expected the throughput to decrease in the same way we expected the voice data to however since the data packets are larger we expected throughput to decrease faster for each hop.

Throughput started out really high for the first hop (blue line) averaging 128656 bytes/sec which is slightly higher than the voice data at one hop throughput. This is followed by a sharp drop in the throughput with the second hop throughput (red line) averaging 1963 bytes/sec which is much lower than the voice data at two hops. The last two hops (green - three hops, purple - four hops) have a small difference between them. Here the throughput is extremely low, averaging 356 bytes/sec and 146 bytes/sec respectively.

Finally, both Figure 5 and Figure 6, demonstrate that the use of smaller sized packets increases the networks performance. In other words for the MP network small packet sizes are more suitable then the larger sized packets such as Ethernet.

Table 4, shows the standard deviations and the means of the throughputs data graphed above.

<table>
<thead>
<tr>
<th>Hop</th>
<th>Avg 73 Byte Data</th>
<th>STD for 73 Byte Data</th>
<th>Avg 1500 Byte Data</th>
<th>STD for 1500 Byte Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>124586.666</td>
<td>1.027E-10</td>
<td>128656.41</td>
<td>5.869E-11</td>
</tr>
<tr>
<td>2</td>
<td>14325.3875</td>
<td>1267.03790</td>
<td>1963.6079</td>
<td>2947.8040</td>
</tr>
<tr>
<td>3</td>
<td>1777.00100</td>
<td>5034.31041</td>
<td>356.02668</td>
<td>463.73697</td>
</tr>
<tr>
<td>4</td>
<td>295.170658</td>
<td>147.076593</td>
<td>145.98199</td>
<td>77.398387</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In our research through the literature surrounding the Batman-adv routing protocol we did not see any evidence of tests run on the one device that uses it the most, the MP. We chose to perform tests on an actual MP testbed.

We focused our attention on packet loss, delay, jitter, throughput in order to help us understand the performance of the MP network with increasing hops and packet sizes. The results we obtained for delay suggest that even at higher hops the network can support VoIP as the values fall well within the boundaries recommended by the ITU-Recommendation G.114. However the packet loss and jitter values above two hops suggest the opposite. This is further supported by the throughput and data gathered which show that networks performance decreases sharply after two hops for both voice and Ethernet sized packets.

We witnessed a few network anomalies which we attributed to the nature of radio packet networks. In future we will re run the same experiments on a different floor of the building in order to see if these anomalies are really due to the nature of the communication medium or of the network itself. Furthermore, comparing Batmand and Batman-adv would give us insights into the performance.
differences between layers 2 and layer 3 routing protocols. This would be a valuable contribution and previous work done on this [5] had inconclusive results.

REFERENCES