Better Approach To Mobile Ad-hoc Networking in MaritimeManet

Michał Rączkiewicz, University of Twente, The Netherlands

Thales S.A. has created an experimental mobile ad-hoc network (MANET) node called MaritimeManet. To improve MaritimeManet's performance, directional antennas are used, as opposed to omnidirectional antennas. There already exist some routing protocols used in MANETs, however it is not clear how they perform in nodes with directional antennas. This research investigated whether Better Approach To Mobile Ad-hoc Networking (BATMAN) is a suitable routing protocol to be used in conjunction with MaritimeManet. BATMAN has been found to be able to fully support multiple network interfaces (antennas), and to maintain distinct connections using multiple antennas at once. BATMAN has also proved to recognize and adopt to dynamic network topology. Convergence time in BATMAN has been found to be linearly proportional to BATMAN Originator Message (OGM) interval.

1 INTRODUCTION

Ship-to-ship communication serves a crucial role in maritime operations. It can be used in search and rescue missions, optimizing maritime traffic, as well as fleet coordination. An efficient communication solution is also vital for innovation, for example it is important for developing autonomous navigation technology [1].

There already exist multiple means for ship-to-ship communication. Commonly used is the very high frequency radio band (VHF) [2]. There are various ship-to-ship systems operating on VHF band, however they all suffer from limited bandwidth [2]. Data could also be transferred between ships by means of broadband internet, based on technologies such as long-term evolution (LTE) or 802.11g IEEE standard (Wi-Fi) [2]. Whereas this is possible, there are some drawback related to those approaches.

BLUECOM+, is a system based on 802.11g standard, that aims at providing broadband internet on distances up to 150 km from shore. To achieve that however, fixed infrastructure needs to be put into place, namely offshore platforms for accommodating radio relays [2]. Similarly, LTE-maritime requires base stations located on land [2]. Moreover, unless expensive satellite technology is used, broadband internet is only available within about 100 kilometers from the shore [2].

There does not exist a system that would allow for a ship-to-ship high throughput data transfer. A potential solution for this communication problem could be a Mobile Ad-hoc Network (MANET). MANETs can be deployed anywhere at sea, since they require no fixed infrastructure and their operation is not dependent on geography [3].

Thales S.A. is experimenting with a MANET concept called MaritimeManet that utilizes directional antennas. Each node uses directional antennas, which allow to increase the propagation range as compared to omnidirectional antennas. Antennas in MaritimeManet are arranged in a sunflower pattern to allow communication in any direction (see Figure 1). The fundamental principle in MaritimeManet is to discover other nodes and to set up the strongest possible wireless connection between two nodes by selecting the 'best' antenna at both nodes (see Figure 1). To maintain connectivity while nodes move with respect to each other, each node periodically selects the 'best' antenna at all nodes, which may result in handover of a connection to an adjacent antenna.

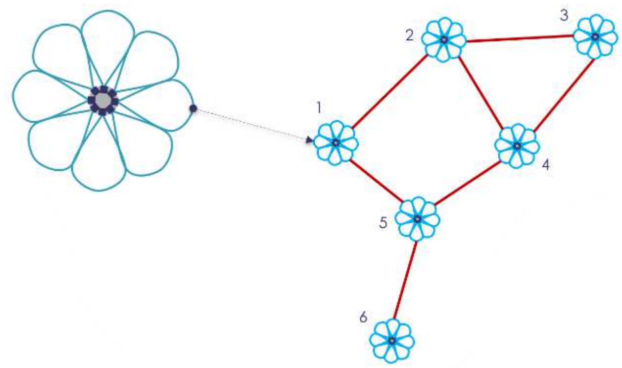


Figure 1. MaritimeManet in a network

Directional antennas in MANETs have been found to have greater range than omnidirectional antennas used in the same network [4]. Since not all nodes in a MaritimeManet network are directly connected, a routing protocol is needed. For this task the Better Approach To Mobile Ad-hoc Networking (BATMAN) protocol was proposed. BATMAN was designed for operating with maximum two hardware network interfaces on a device, each MaritimeManet node however, has multiple directional antennas. It is therefore unclear, whether BATMAN will correctly perform in MaritimeManet.

Some routing protocols for MANETs with directional antennas have already been created and implemented in a test environment, however they are purely experimental [5, 6]. The purpose of this research is to verify whether BATMAN is suitable as a routing protocol on top of MaritimeManet.

2 RELATED WORK

To obtain insight into already existing maritime communication technologies, [1, 2] were investigated. Those research papers have revealed that maritime communication capabilities are lacking in range and bandwidth as compared to communications on land. Especially insightful was [2] which laid out all existing technologies and their drawbacks. An overview of Ad-hoc network technology and related protocols was also conducted [3, 4, 5, 6, 7]. Whereas [3] focused on security, it also gave a good overview of advantages of MANET. Some literature regarding directional antennas in MANET was found [4, 5, 6], however those papers describe systems treat

TScIT 39, July 7, 2022, Enschede, The Netherlands

© 2022 University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

each antenna as a distinct system. Lastly, BATMAN itself was examined [8, 9], to obtain an understanding of protocol's operation.

The literature review has revealed that there are numerous articles about routing in MANET with omnidirectional antennas and few articles focusing on routing in MANET with directional antennas. No article was found that investigated BATMAN's suitability for operations in MANET with directional antennas.

3 RESEARCH PROBLEM

The support for the existing protocols would more than likely be lacking as they are merely experimental solutions. There exists a need for a protocol to be used in conjunction with MaritimeManet.

This paper will investigate whether BATMAN is a suitable routing protocol for MaritimeManet. The purpose of the research is to demonstrate how well BATMAN handles a network built with multiple MaritimeManet nodes and to measure its performance. MaritimeManet is a system whose nodes travel independently and arbitrarily from one another, BATMAN will therefore also be tested for its capability for routing in dynamic network topologies. The problem statement leads to the following research questions:

How effectively can BATMAN routing protocol be used in a MaritimeManet, i.e. in wireless networks in which nodes have multiple directional antennas? The above will be investigated by means of the following sub-questions:

1. What is the minimal feasible testbed with which BATMAN can be tested in static and dynamic topology?
2. How can BATMAN operate in a node with multiple directional antennas each having only one neighbor?
3. How well does BATMAN handle dynamic changes in topology?

In order to answer the research question, four experiments will be executed. For experimentation, a suitable testing environment needs to be developed. Firstly, the minimum feasible testbed will be described, answering the research sub-question 1. Then, in experiment 1, BATMAN will be tested in a static environment, with three distinct antennas. Then, experiments 2, 3 and 4 will test BATMAN in a dynamic topology.

4 TESTBED

To verify if BATMAN is a suitable choice for operations in MaritimeManet, a testbed needs to be established. The testbed must meet certain prerequisites and be representative of MaritimeManet. Firstly, the hardware choices will be described, then network configuration will be discussed. After network is configured, experimentation tools and BATMAN kernel module will be installed. Finally, the configuration and operation of BATMAN will be explained.

4.1 Hardware

Whereas MaritimeManet node is a wireless network device, BATMAN can also be tested in a wired environment. Ultimately, there is no difference in operation of a network protocol in a wireless or wired medium. Moreover, experimenting in a wired environment will allow for a much greater degree of control over the network topology. Since the environment is not wireless, antennas are not needed. Instead, they will be abstracted away to virtual network interfaces. BATMAN doesn't perceive a difference in a hardware network interface (antenna) and a virtual network interface [10].

Since BATMAN is a software process, which runs in an operating system (OS) kernel, or in an OS application, a computer supporting a suitable OS is needed. For that purpose PCEngines Alix2d2 was chosen. Alix is a single board computer with 256MB of random access memory (RAM) and two ethernet ports, which will be enough to satisfy the networking experiments. OpenWrt was chosen as the operating system for Alix. It is a lightweight Linux distribution targeting embedded devices [11]. On one hand OpenWrt offers a fully writable file system, which will be needed to conduct the experiments, on the other hand it doesn't have high hardware requirements. Additionally, a built-in package manager allows to install BATMAN and other tools necessary to conduct the experiments.

To build a BATMAN network, multiple Alixes are needed, each one acting as a single node in the network. At least four Alixes are needed to verify if BATMAN is capable of maintaining more than two connections using distinct hardware interfaces. To meaningfully test nontrivial dynamic topologies, even more Alixes are needed. Five Alixes will therefore be used to build the testbed.

4.2 Network Configuration

Each Alix only has two ethernet ports, therefore BATMAN data streams need to be separated to simulate the existence of multiple hardware interfaces. This can be done with the help of virtual local area network (VLAN). VLAN separates network traffic on the layer 2 of the network stack. It is accommodated by the Ethernet protocol, with an extended header assigning an identifier. Even though network traffic is propagated on one ethernet cable, VLANs provide their separation.

Since the most complex topology consists of every Alix connecting with all other Alixes with an individual link, four VLANs per Alix are needed. Each Alix will be configured with 5 VLAN networks; VLAN 100 for control purposes and four VLANs x1, x2, x3 and x4 where x stands for the identification number of an Alix. Only those four VLANs will be used for experimentation. Experimentation VLANs will terminate at four distinct virtual network interfaces on every Alix. Those interfaces are eth0.x1, eth0.x2, eth0.x3 and eth0.x4, where x stands for the identification number of the Alix.

To form connections between BATMAN nodes, experimentation VLANs will be bridged. For this purpose OpenVSwitch (OvS) is used. OvS maps network interfaces onto an internal software switch, which can be used to configure and modify the network topology. To operate this switch, a separate device is needed. For this purpose, PC Engines APU is used, which is a more capable device than an Alix. APU running the OvS switch is configured with 20 experimentation VLANs, 4 for each Alix, and the control VLAN. The only point where two VLANs can interchange, is the OvS, which ensures total control over the network topology. The complete VLAN setup can be seen in Figure 2.

To control network devices a control laptop is needed. The control laptop runs both Windows 10 and Ubuntu operating systems. Windows is used for internet sharing to Alixes, as its configuration is much simpler than on other operating systems. Ubuntu is a Linux distribution that will be used during the

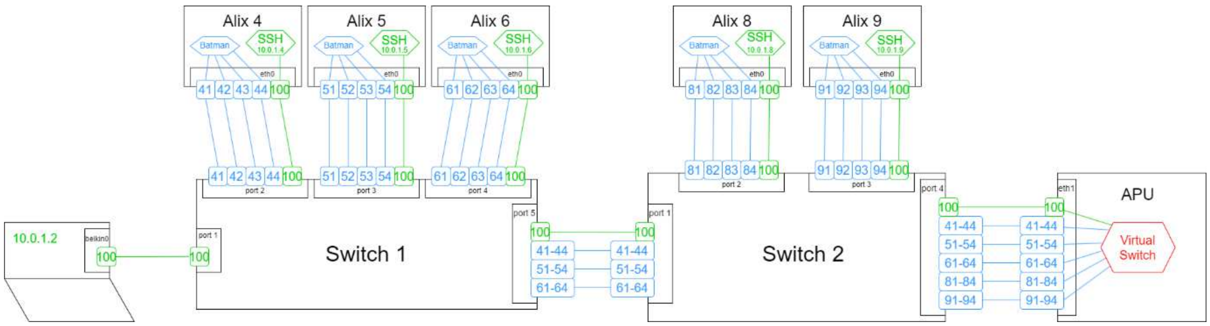


Figure 2. VLANs in the testbed

experimentation phase. Linux is needed to create a virtual interface to access VLAN 100, which wasn't possible on Windows.

To access network devices, Secure Shell (SSH) is used. SSH requires Internet Protocol (IP) addressing to operate, therefore, each interface operating on VLAN 100 received an IP address on ethernet port 0 (eth0). The control laptop was assigned the IP address 10.0.1.2 and 10.0.1.1 was reserved for the default gateway, as it is required when configuring network connections on Windows 10 OS. To connect all devices, two Mikrotik RB260GSP switches are used. These switches have five ports each, four of them capable of Power over Ethernet (PoE). PoE provides connected device with power via an ethernet cable, so that an external power supply is not needed. All Alixes in the system will be powered with PoE. Each switch is configurable with RouterOS which is the firmware installed on all the Mikrotik Switches. RouterOS has a graphic interface that can be accessed with a browser, with the IP address of the switch.

The first switch was assigned address 10.0.1.3 and connected to Alixes with addresses 10.0.1.4, 10.0.1.5 and 10.0.1.6. The second switch addressed 10.0.1.7 connected Alixes with addresses 10.0.1.8, 10.0.1.9, as well as the APU with address 10.0.1.13. The complete overview of the system can be seen in Figure 2.

To route VLANs appropriately in switches, VLAN port forwarding rules must be established using the RouterOS. VLANs routing is configured by selecting ports to which VLAN has access. Experimentation VLANs only have access to the port to which the Alix is connected on the switch, and to the port where APU is connected. Additionally, experimentation VLANs from Alix 4, 5 and 6 can cross to switch 2, to access APU (see Figure 2). Control VLAN 100 can access all devices in the network.

Additionally, Internet connection needed to be established to download the packages. This has been done by the means of Windows network sharing, with the control laptop acting as a gateway. Internet sharing will be disabled after all Alixes are fully configured.

4.3 Installation of Experimentation Tools

Finally, all Alixes and APU are accessible with SSH. The installation of BATMAN and various experimentation tools can thus begin. OpenWrt has a built-in package manager tool (Opkg) which can be used to download and install all of the above.

Table 1 presents tools that are needed to check the operation of BATMAN. Batctl-full will be used to display the routing table, as well as to perform runtime tweaks to frequency of Originator Messages (OGM) and Echo Location Protocol messages (ELP). Tcpdump will be used to capture the network traffic and save it to files. Those files will later be analyzed on the control laptop with Wireshark. To generate network traffic, iPerf will be used.

Table 1. Installed Opkg Packages

Package	Description
kmod-batman-adv	Implementation of BATMAN protocol
batctl-full	BATMAN diagnostic tool
tcpdump	Network traffic capture
iperf	Network traffic generator

4.4 BATMAN Configuration

With BATMAN installed, it needs to be configured. BATMAN abstracts the routing decisions by separating its interfaces into soft interfaces and hard interfaces. Note the naming confusion here, BATMAN hard interfaces are not to be confused with hardware interfaces. BATMAN hard interfaces can also be virtual interfaces, which indeed is the case in the testbed. To avoid further confusion, BATMAN's interfaces will be referred to as soft and hard, and network interfaces as hardware and virtual. BATMAN soft interface serves as a gateway to the BATMAN network. Each bat0 interface must have at least one hard interface, associated with a hardware or virtual network interface. Whenever a packet arrives at bat0, BATMAN decides which hard interface to forward it to (in case of transiting or originating traffic) or whether to strip it of the BATMAN header and pass it through bat0 to the process using bat0 as a network interface [10]. Configuration of BATMAN in an Alix is presented in

Figure 3. In this case, eth0.x1 – eth0.x4 are BATMAN hard interfaces. In every Alix, a bat0 soft interface was created. Then the four (eth0.x1 – eth0.x4) virtual network interfaces associated with VLANs were assigned as BATMAN hard interfaces.

BATMAN uses two types of control packets. The Echo Location Protocol (ELP) packets [12] and the Originator Message (OGM) [13]. ELP's task is to discover new nodes in the BATMAN network, and OGM is responsible for creating the originator table, which is used for routing. Both ELP packets and OGMs are flooded to the network on all hard interfaces. When ELP is received by a BATMAN node, it updates its neighbor table and drops the packet. When an OGM is received, the originator table is updated and it is propagated on all other hard interfaces. OGMs therefore traverse the entire network, whereas ELPs can only go within the range of one hop.

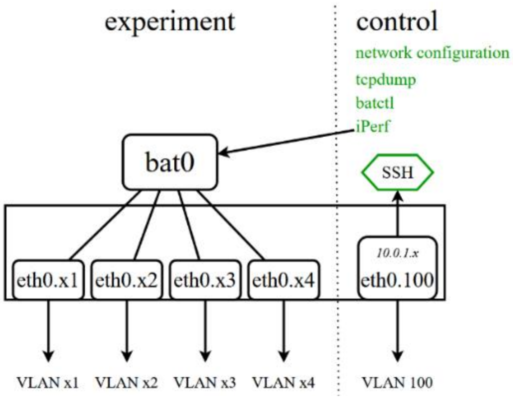


Figure 3. eth0 and its virtual interfaces on an Alix

The correct operation of the testbed was verified by opening SSH sessions to each Alix and APU in parallel, while inspecting the RouterOS on both switches. This proved that all network devices can be accessed at any point thought the experiment. Additionally, OvS was configured with the topology in Figure 4. Then, batctl was used to show routing tables on all participating BATMAN nodes. Routing tables confirmed the correct operation of BATMAN.

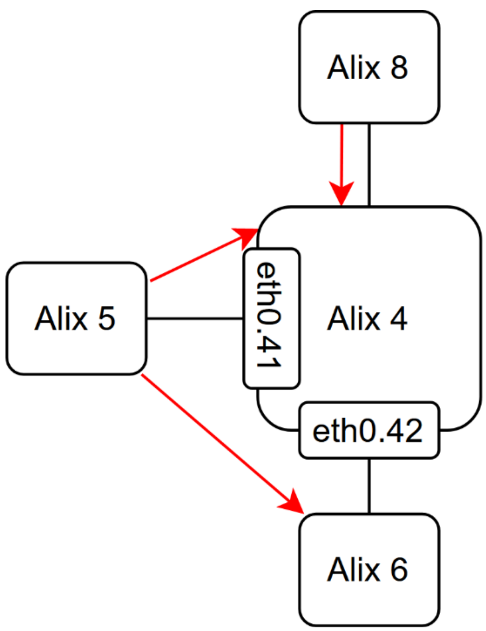


Figure 4. Experiment 1 topology with selected BATMAN hard interfaces

5 EXPERIMENTS

In order to investigate BATMAN's performance, multiple experiments will be conducted. Experiment 1 will verify if BATMAN is capable of operating using multiple network hardware interfaces. Experiments 2, 3 and 4 will reveal BATMAN's performance in dynamic topologies. Experiment 2 will investigate the network convergence time for a selected node, based on a handover scenario. Converge time is the time it takes for a network node to update its routing table so that it correctly represents the network topology. Experiment 3, will verify if BATMAN selects the most efficient link to the destination node. Finally, Experiment 4 will find BATMAN configuration that yields the shortest convergence time. The overview of experiments and their characteristics can be seen in Table 2.

Table 2: Overview of experiments and their characteristics

Exp. #	Topology	Number of intermediate hops	BATMAN settings
1	Static	fixed	OGM = 1000ms, ELP = 100ms
2	Dynamic (handover)	variable	OGM = 1000ms, ELP = 100ms
3	Dynamic (lost link, new link)	variable	OGM = 1000ms, ELP = 100ms
4	Dynamic (handover)	Dependent on topology	OGM variable, ELP variable

5.1 Static Topology

To check if BATMAN is capable of operating in a node with multiple directional antennas, each having only one neighbor, experiment 1 was conducted. Topology presented in Figure 4 was used in the experiment and is the minimal topology to answer the posed question. Alix 4 will be the node maintaining three connections with distinct BATMAN hardware interfaces. The topology was configured statically using the OvS.

Three iPerf sessions were run simultaneously. Each iPerf session will be set to transmit User Datagram Protocol (UDP). UDP does not confirm the successful packet delivery, therefore it doesn't retransmit traffic in case of a packet loss. IPerf was set to transmit packets of 1460 bytes, which is the biggest possible size to be sent without packet fragmentation. The above settings aim at making the tcpdump files analysis easier by eliminating retransmitting traffic and packets fragmentation. IPerf sessions were set to be 10s long and to transmit at 30Mb/s. Sessions can be seen in Figure 4, indicated with red arrows.

During the experiment, tcpdump was used to capture traffic passing through the eth0.41 and eth0.42 BATMAN hard interfaces on Alix 4 (see Figure 4). The tcpdump files were later analyzed with Wireshark. Additionally, iPerf on Alix 4 and Alix 6 will save its log files, which were examined for packet loss.

The investigation of inter arrival times of ELP packets have revealed that each BATMAN hard interface transmits an ELP packet every 114ms. This shows that ELP packets have slightly lower frequency than 100ms configured. Whereas the inter arrival time of packets is slightly above the desired period, this doesn't negate the correct operation of BATMAN as the delay between the ELP packets is negligible.

BATMAN has been configured to send an OGM once every second from each of the batman hardware interfaces. In every network, n-1 OGMs per second are expected to be seen on each hard interface, where n is the number of nodes in the network. This is because, BATMAN node always drops the OGM if the source address is the same as the address of the node [13]. In topology 1,

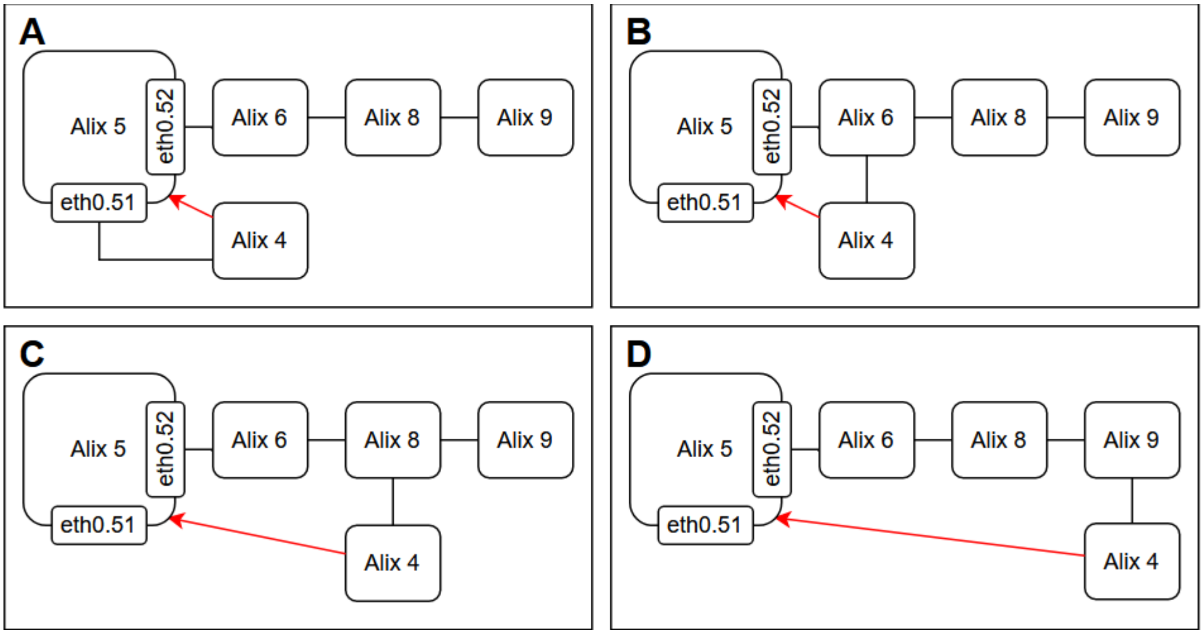


Figure 5: Topology 2A – 2D with selected BATMAN hard interfaces

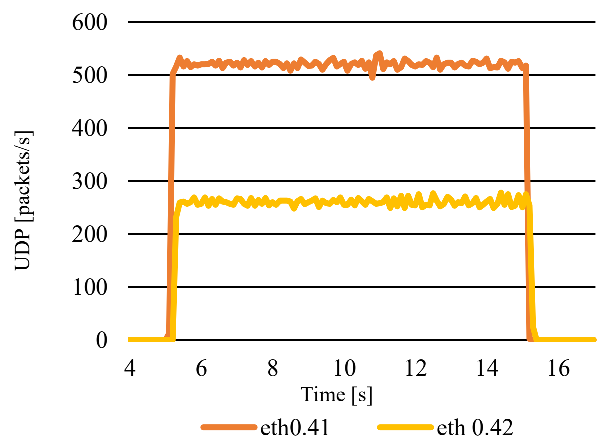
three OGMs per second are therefore expected on each interface or one originating in each node of the network. On average 3.15 OGMs per second were seen on eth0.41 and 3.47 on eth0.42. The OGMs occurred slightly more often than expected. The routing algorithm of BATMAN performed well, creating the originator table presented in Figure 6. The originator column displays the origin of the OGM message. This column is also the list of available destinations. The last-seen column is the time since the last OGM message has been received. Throughput is BATMAN's estimation of link quality, given in Mb/s. In case there are multiple entries in the originator column for one destination, BATMAN firstly picks the entry with the lowest last-seen value, if that value is identical with other entries, then the entry with the highest throughput is chosen.

```
  Originator        last-seen ( throughput)  Nexthop          [outgoingIF]
          Alix8    0.640s (       77.8)              Alix5 [    eth0.41]
          Alix8    0.640s (       77.8)              Alix6 [    eth0.42]
*         Alix8    0.640s (      100.0)  *           Alix8 [    eth0.43]
          Alix5    0.470s (       77.8)              Alix8 [    eth0.43]
          Alix5    0.470s (       77.8)              Alix6 [    eth0.42]
*         Alix5    0.470s (      100.0)  *           Alix5 [    eth0.41]
          Alix6    0.180s (       77.8)              Alix5 [    eth0.41]
          Alix6    0.180s (       77.8)              Alix8 [    eth0.43]
*         Alix6    0.180s (      100.0)  *           Alix6 [    eth0.42]
```

Figure 6. Originator table in Alix 4 in experiment 1

Lastly, packet loss was examined in iPerf log files. IPerf maintained a steady transmission rate of 30Mb/s in all sessions. Almost all packets have arrived at destination, with packet loss of 0% rounded to the nearest decimal. Traffic in Alix 4 can be seen in Graph 1. There are twice as many UDP packets seen on eth0.41, than eth0.42. This is because eth0.41 handles traffic originating in Alix 5, with destinations to both Alix 4 and Alix 6. The UDP packets on eth0.41 are therefore both packets terminating in Alix4 and packets transiting to Alix 6. On eth0.42, only packets outgoing towards Alix 6 can be seen. This shows that BATMAN correctly identifies the terminating and transiting traffic.

Graph 1. UDP traffic on Alix 4 in Experiment 1



BATMAN has proved to be able to handle more than two connections with distinct hardware interfaces. It correctly propagates and receives ELP messages and OGMs which results in the originator table being correctly set up. Additionally, BATMAN recognizes all created hardware interfaces, and knows which interface to use for each route. Finally, BATMAN has proved its capability to send data streams with negligible packet loss.

5.2 Handovers

Handover is a process in which the route, from a node of a system, that maintains communication with a different node in the system, is altered. An example of handover can be seen in Figure 5. In topology 2A Alix 4 transmits to Alix 5 without any intermediate nodes. When topology changes to 2B, Alix 4 is handed over to Alix 6, but the connection to Alix 5 is maintained. To verify if BATMAN is capable of maintaining a connection when transmitter is handed over to a different node in case of a network topology change, experiment 2 was conducted.

Topology presented in Figure 5 is a dynamic topology. Each box represents the topology after a change. Experiment 2 will be started with the topology 2A configured statically. Then, three topology changes will be performed. All topologies are configured using the OvS. The virtual switch will be configured using a shell script run on the APU before the experiment. Topology changes will be executed using three separate shell scripts in 10 second intervals.

To test network throughput iPerf tool will be used. An iPerf session will be run continuously through all topology changes. The session can be seen in Figure 5 indicated with a red arrow. IPerf was used with the same settings as in experiment 1, apart from the session duration, which was set to 45 seconds to cover all topology changes.

During the experiment, tcpdump will be used to capture traffic passing through the eth0.51 and eth0.52 interfaces on Alix 5 (see Figure 5). This will effectively allow to eavesdrop on all traffic in the network (excluding ELP and OGM), since Alix 5 is the only receiver of packets. The tcpdump file will be later analyzed with Wireshark. Additionally, iPerf on Alix 4 will save its log file, which will also be later examined for packet loss.

During the experiment BATMAN originator table will be displayed on Alix 4 every second. Originator table contains last seen column, which indicates the last time this node has received an OGM packet from other nodes. This will allow to know the exact moment when Alix 4 updates its originator table. Sadly, batctl does not allow to save its output to a file, therefore a separate terminal window will be used and then the terminal log will be saved to a file.

During experiment, it took BATMAN about 4.5 seconds to update the entire originator table after each topology change. Some entries in the table have been updated quicker, the entry for Alix 5, which was the destination node, was updated last in all topology changes, however. Table 3 presents the time in milliseconds, from when topology change was initiated, to the time the next hop entry in the originator table of Alix 4 was updated. The leftmost column contains destinations in the routing table. Neighboring nodes, after the topology change, have their time marked in **bold**.

Table 3. Period from topology change to originator table entry

Originator	A → B	B → C	C → D
Alix 5	5144	4307	4672
Alix 6	**981**	4307	4577
Alix 8	1080	**245**	4577
Alix 9	388	560	**729**

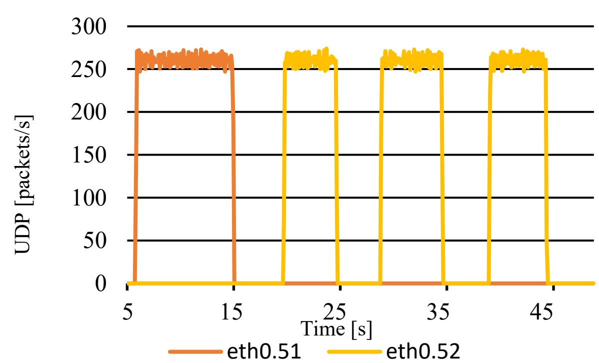
Such long update times are achieved, despite OGMs arriving on eth0.52 frequently. The two outliers are OGMs from Alix 5 and OGMs from Alix 6, which had arrival frequency of slightly less than 1/s. The time it takes for BATMAN to update the originator table could be decreased, by increasing the frequency of OGMs. This will be tested in experiment 4. Despite large convergence times of the originator table, BATMAN correctly identified each topology after it has changed.

Because of the long convergence times, iPerf session reported a significant packet loss of 35%. This is because BATMAN selects the hard interface to transmit data packets to, from the originator table that has not yet been updated. UDP packets arriving on Alix 5 can be seen in Graph 2. Topology changes have been executed at 15, 25 and 35 seconds into the experiment. Note the downtimes in packet arrival after each topology change contributing to significant packet loss.

ELP was configured with the interval of 100ms. Both, interface eth0.51 and eth0.52 see ELP packets with the same interval as in experiment 1. Interface eth0.51 is cut off from any other node after the first topology change. This is reflected by only observing ELP packets originating from Alix 5.

Whereas BATMAN is quite inefficient when updating its originator table, it successfully managed to update its originator table, albeit causing significant packet loss. Experiment 4 will therefore aim at selecting the best frequency for OGMs in order to improve efficiency of BATMAN. Nevertheless, after the originator table was updated, BATMAN kept delivering packets at the speed of 30Mb/s as ordered by iPerf.

Graph 2. UDP traffic on Alix 5 in experiment 2



5.3 Alternative Routes

Experiment 3 will verify if BATMAN can select an already existing, alternative route to the destination node if the original route is broken. Additionally, it will be tested if BATMAN selects the more efficient route compared to the currently selected route.

The topology presented in Figure 7 is a dynamic topology. Each box represents the topology after a change. The experiment will be started with the statically configured topology 3A. Then two topology changes will be performed. The initial topology, as well as topology changes, are performed using the OvS. During the experiment, topology changes will be executed in 10 second intervals.

To test the network throughput the iPerf tool will be used. Two iPerf sessions will be run continuously through all transitions. Sessions can be seen in Figure 7, indicated by the red arrow. IPerf sessions will be configured with the same settings as in experiment 1, albeit with small changes. Namely, the throughput will be decreased to 20 Mb/s, to decrease tcpdump file sizes, thus saving valuable Alix disk space. Propagation time will be set to 40s to cover all topology changes.

During the experiment, tcpdump will be used to capture traffic passing through the BATMAN hardware interfaces eth0.81 and eth 0.91 (see Figure 7). The above tcpdump configuration will allow to eavesdrop on all possible routes from Alix 4 to Alix 8 and from Alix

6 to Alix 8 in all topologies. Tcpdump files will be later analyzed with Wireshark. In the first topology, the amount of packets passing interfaces eht0.81 and eth0.91 will be compared. This will allow to determine which route packets from Alix 6 to Alix 8 take. The route taken by traffic from Alix 6 to Alix 8 will reveal if BATMAN correctly routes traffic with less busy link. During the experiment, BATMAN originator table will be displayed on Alix 4 and Alix 6 in a separate SSH session every second. Originator tables will be used to confirm routing decisions made by BATMAN.

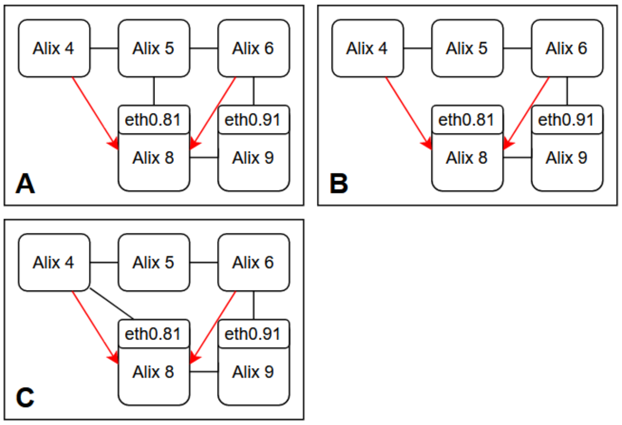


Figure 7. Topology 3A – 3C with selected BATMAN hard interfaces

During the experiment no packet loss was recorded for data stream from Alix 6 to Alix 8. A data loss of 12% was recorded for data stream from Alix 4 to Alix 8. The packet loss is attributed to the topology changes. An approximately 5 second period of throughput below 20Mb/s was observed in the iPerf log file for the data stream originating from Alix 4. As mentioned in experiment 2, BATMAN takes around 4.5s to update the originator table entries, for destination nodes not being immediate neighbors. Eventually, when the originator table was updated, the amount of packets/s on eth0.91 doubled. This shows that BATMAN has correctly rerouted packets from Alix 4 via Alix 9.

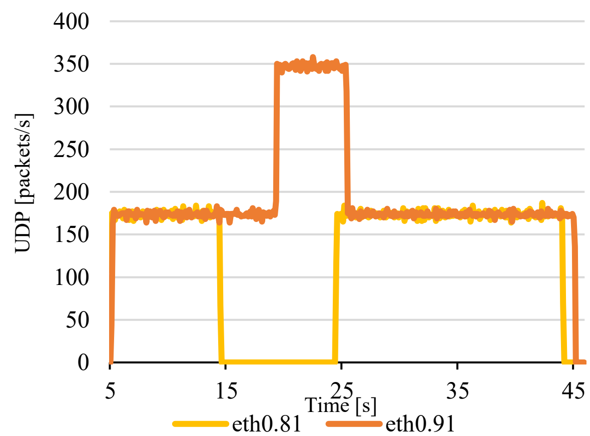
The second topology change, however, didn't affect either of the iPerf data streams. Packets originating in Alix 6 kept traversing Alix 9 as it remained the most efficient route to Alix 8. Data stream from Alix 4 to Alix 8 was not affected either, despite the topology change. This is because the route from Alix 4 to Alix 8 via Alix 9 was not interrupted in the second topology change. The UDP packets from Alix 4 were therefore, still routed via Alix 9, despite a shorter route being available.

It took BATMAN about a second to update the originator table on Alix 4, for the route to Alix 8. This moment can be seen on Graph 3 by the return of the traffic to eth0.81. The increased level of traffic can still be observed on eth0.91 at that moment. This is because UDP packets from Alix 4 routed via Alix 9 are still traveling to their destination. The route from Alix 9 is much longer than the direct link from Alix 4 to Alix 8, therefore an overlap can be seen in the graph. For a moment, Alix 8 receives data from Alix 4 directly (arriving at eth0.81) and via Alix 9. Packets arriving at eth0.81 effectively overtake packets propagated via Alix 9.

Through the experiment packets from Alix 6 keep traversing Alix 9. Whereas in topology 3B, this is the only route available to Alix 8, and in topology 3C the alternative route has two intermediate nodes, as opposed to one, BATMAN still made the correct choice in topology 3A. The only difference distinguishing the two routes from Alix 6 to Alix 8 in topology 3A, is the presence of traffic from Alix 4 to Alix 8 on the alternative route. BATMAN has identified that fact and chose to route Alix 6 to Alix 8 traffic via Alix 9.

BATMAN has performed well in finding an alternative route to the destination node in case of a broken link. Whereas, with default OGM frequency it suffers from slow originator table updates, it nevertheless found a detour to the destination node. BATMAN correctly identifies the less busy route when two otherwise identical routes are available. Whenever a shorter route than the currently used one appears, BATMAN updates its originator table correctly and reroutes current traffic by way of the shorter route.

Graph 3. UDP traffic in experiment 3



5.4 Optimal Configuration

Experiment 4 will be conducted to find the OGM and ELP interval values, for which the originator table convergence time is the shortest. For this purpose, dynamic topology 2 will be used (see Figure 5). Additionally, to check if the convergence time can be generalized to an arbitrary topology, the convergence time will be measured in the alternative topology presented in Figure 8, and then compared with topology 2. The iPerf sessions in both topologies are indicated with red arrows in Figure 5 and Figure 8.

Experiment 4 will be repeated multiple times in dynamic topology 2, each time changing OGM and ELP interval. Initial values will be 100ms for ELP interval and 1000ms for OGM interval. OGM interval will be decreased by 100ms, until it reaches 100ms. If lower convergence time with OGM interval of 100 millisecond than with the OGM interval of 200 millisecond are achieved, that would imply the OGM interval can be shortened still to yield even shorter convergence time of the originator table. In this case ELP will be decreased to 10ms and another 10 iterations of the experiment will be repeated with OGMs interval decreasing by 10ms starting at 10ms. Convergence time in alternative topology will be measured for selected interval lengths of OGM and ELP. OGM and ELP interval lengths in the alternative topology will be selected to confront the convergence time achieved in topology 2.

Each iteration of the experiment will produce iPerf log files indicating throughput and tcpdump capture on eth0.51 and eth0.52. Convergence times will be identified using Wireshark, by comparing the arrival time of the last UDP packet before the topology change and the first UDP packet after the topology change. Subtracting those two values will yield the convergence time over the topology change. The same will be performed in the alternative topology, albeit with tcpdump listening on eth0.51, eth0.52 and eth0.53 (indicated interfaces in Figure 8).

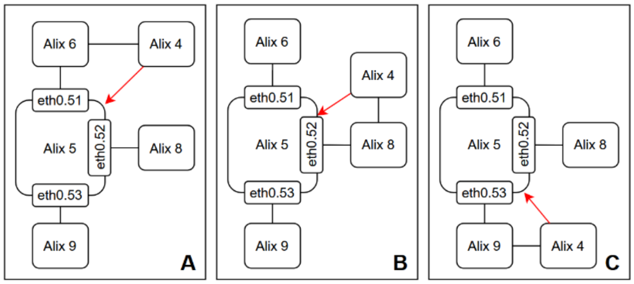
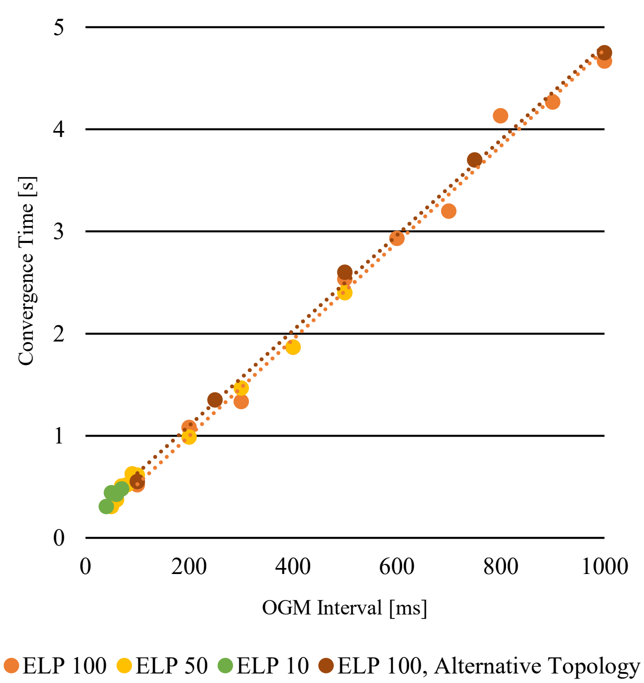


Figure 8: Alternative topology in experiment 4 with selected BATMAN hard interfaces

The BATMAN convergence time has been greatly reduced by decreasing the interval of OGM. A linear relationship has been found between the OGM interval and the convergence time of the originator table (see Graph 4). Increasing the ELP interval while preserving the OGM interval didn't yield any improvements in convergence time, further confirming the originator table update issue relates to low frequency of OGMs. Similar results were achieved in the alternative topology, establishing the same linear relationship.

Graph 4. Converge time of Alix 4 originator table entry with destination of Alix 5 with respect to OGM interval



6 CONCLUSIONS

BATMAN can be tested for correct operation in a wired environment. Wired testbed is a much better choice than a wireless environment. Wired testbed requires much less space and its network topology is easily configurable with high degree of confidence. Additionally, wired medium suffers from almost no network interference, as opposed to a wireless medium.

BATMAN is a suitable routing protocol for operation in a node with multiple hardware interfaces, each interface maintaining one peer to peer link. This makes BATMAN a suitable routing algorithm for operation in MaritimeManet. BATMAN correctly routes traffic to neighboring and non-neighboring nodes. Convergence time of a BATMAN originator table is variable per destination. Destinations that are neighbors of the node, tend to have their originator table entries updated the fastest. BATMAN originator table convergence time is linearly proportional to the OGM interval. The OGM interval should be set as short as possible to achieve the shortest originator table convergence times.

In this research, BATMAN was tested with default configuration, only with OGM and ELP interval altered. It is possible other configuration options influence convergence time and thus performance of BATMAN. Most notably, the OGM aggregation option. Additionally, in all experiments, the ELP interval was set either equal or shorter than OGM interval. Further research should focus on investigating BATMAN's performance with OGM aggregation disabled and with ELP intervals longer than OGM intervals.

ACKNOWLEDGMENTS

I would like to thank Jan Laarhuis, the supervisor of this research thesis. Jan has always provided me with high quality feedback on everything I worked on throughout the research. His guidance has contributed to my development as a researcher.

I would also like to thank Omar Gadelmawla, whose experience with Alixes, OpenWrt and Mikrotik switches was of great help. Omar has made himself available and provided me with valuable help through the research.

REFERENCES

[1] X. Zhang, C. Wang, L. Jiang, L. An and R. Yan, "Collision-avoidance navigation systems for Maritime Autonomous Surface Ships: A state of the art survey," *Ocean Engineering,* vol. 235, 2021.

[2] F. S. Alqurashi, A. Trichili, N. Saeed, B. S. Ooi and M.-S. Alouini, "Maritime Communications: A Survey on Enabling Technologies, Opportunities, and Challenges," *IEEE Internet of Things Journal,* vol. 10, pp. 3525-3547, 2023.

[3] M. Yadav and N. Uparosiya, "Survey on MANET: Routing Protocols, Advantages, Problems and Security," *International Journal of Innovative Computer Science & Engineering,* vol. 1, no. 2, pp. 12-17, 2014.

[4] T. Cakan and V. Wieser, "QoS parameters enhancement by using directional antennas in MANET," in *2012 ELEKTRO,* Rajecke Teplice, Slovakia, 2012.

[5] R. Ramanathan, J. Redi, S. Member, C. Santivanez, D. Wiggins and S. Polit, "Ad Hoc Networking With Directional Antennas: A Complete System Solution," *IEEE Journal on Selected Areas in Communications,* pp. 496-506, 2005.

[6] R. J. Mohsin, J. Woods and M. Q. Shawkat, "Density and mobility impact on MANET routing protocols in a maritime environment," in *2015 Science and Information Conference (SAI)*, London, Gread Britain, 2015.

[7] T. Korakis, G. Jakllari and L. Tassiulas, "CDR-MAC: A Protocol for Full Exploitation of Directional Antennas in Ad Hoc Wireless Networks," *IEEE Transactions on Mobile Computing,* vol. 7, pp. 145-155, 2008.

[8] I. A. Voinov, A. C. Cedeño, J. Chung and G. González, "A Performance Analysis of Wireless Mesh Networks Implementations Based on Open Source Software," in *Open Source Software: Mobile Open Source Technologies*, Berlin, Germany, Springer Berlin Heidelberg, 2014, pp. 107-110.

[9] D. Johnson, N. Ntlatlapa and C. Aichele, "Simple pragmatic approach to mesh routing using BATMAN," in *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries*, Pretoria, South Africa, 2008.

[10] "Open-Mesh.org," [Online]. Available: https://www.open-mesh.org/projects/batman-adv/wiki/Wiki. [Accessed 20 06 2023].

[11] OpenWrt, "Welcome to the OpenWrt Project," [Online]. Available: https://openwrt.org/. [Accessed 19 06 2023].

[12] "Open-Mesh.org," [Online]. Available: https://www.open-mesh.org/doc/batman-adv/ELP.html. [Accessed 22 06 2023].

[13] "Open-Mesh.org," [Online]. Available: https://www.open-mesh.org/projects/batman-adv/wiki/OGM. [Accessed 22 06 2023].