Universidade de Aveiro

Redes e Sistemas Autónomos

Project Report

André Clérigo (98485), Pedro Rocha (98256)

Departamento de Eletrónica, Telecomunicações e Informática

June 16th, 2023

Contents

1 Introduction **3**
- 1.1 Overview . . . 3
- 1.2 Motivation . . . 3
- 1.3 Project Objectives . . . 4

2 Architecture **5**
- 2.1 Project Architecture . . . 5
- 2.2 Technological view . . . 5
- 2.3 Envisioned Scenario . . . 6

3 Implementation **7**
- 3.1 Configurations . . . 7
- 3.2 Dashboard Webapp . . . 7
- 3.3 Message Flow . . . 8

4 Testbed Experiments **9**
- 4.1 Test Methodology . . . 9
- 4.2 Results . . . 9
  - 4.2.1 RTT Results . . . 10
  - 4.2.2 Jitter Results . . . 13
  - 4.2.3 PDR Results . . . 14
  - 4.2.4 Throughput Results . . . 16
- 4.3 Demo . . . 17
  - 4.3.1 Demo Architecture . . . 17
  - 4.3.2 Dashboard Results . . . 17
  - 4.3.3 Messaging Service . . . 18

5 Conclusion **21**

6 Future Work **22**

7 Resources **24**

List of Figures

2.1 Physical architecture. 5
2.2 Envisioned scenario. 6

3.1 Visual message timeline. 8

4.1 Maximum distance achieved with LoS. 10
4.2 RTT in Function of Distance (No LoS Inside). 10
4.3 RTT in Function of Distance (No LoS Outside). 11
4.4 RTT in Function of Distance (1 Hop). 11
4.5 RTT in Function of Distance (2 Hop). 12
4.6 RTT in Function of Distance (2 Hop to Internet). 12
4.7 Jitter in Function of Distance (No LoS). 13
4.8 Jitter in Function of Distance (LoS). 13
4.9 PDR in Function of Distance (No LoS). 14
4.10 PDR in Function of Distance (LoS). 15
4.11 Throughput in Function of Distance (No LoS). 16
4.12 Throughput in Function of Distance (LoS). 16
4.13 Architecture for the demo. 17
4.14 Dashboard Results for the demo. 18
4.15 Contact list. 19
4.16 Message chat. 19
4.17 Message and state acknowledgment. 20

6.1 Raspberry Pi 4 used. 22
6.2 5G module. 23

Chapter 1

Introduction

1.1 Overview

The project is focused on establishing an emergency network infrastructure in Aveiro by integrating the existing infrastructure of the Aveiro Tech City Living Lab (ATCLL) through the utilization of smart lamp posts and drones. This integration aims to expand an ad-hoc network that provides both emergency communication and Internet access to the population during critical times This involved setting up a B.A.T.M.A.N.[1] network between Raspberry Pi devices. With tests conducted both inside and outside a building, incorporating line-of-sight (LoS) and non-line-of-sight (No LoS) conditions as well as connections to the Internet.
This project aims to provide a reliable and effective emergency communication network to the population of Aveiro, ensuring their safety and well-being during critical situations.

1.2 Motivation

In today's ever changing world, where emergencies and crises can strike unexpectedly, having a robust and resilient network infrastructure is of the most importance. Traditional networks may face challenges during emergencies, leading to high delays in this critical communications or even no communication at all. Recognizing that it is critical to provide assistance to affected individuals, the project aims to establish an emergency network infrastructure that can effectively handle such events to the population and emergency workforce of Aveiro.

The backbone of this project relies on integrating the pre-existing infrastructure of the ATCLL. This integration will utilize both the smart lamp posts available within ATCLL and the drones owned by the lab. The ATCLL is a innovation hub that combines cutting-edge technology, research, and development, making it an ideal platform for implementing an emergency network. By leveraging the infrastructure already in place, the project can significantly reduce costs and implementation time.

The use of smart lamp posts and drones to expand the ad-hoc network offers several advantages. Street poles, which are already present in some parts of the city and many more to come, can be equipped with ad-hoc modules to expand our ad-hoc network. The smart lamp posts can also be equipped with 5G modules to provide high-speed connectivity during emergencies. The integration of drones complements the network infrastructure by providing additional coverage and mobility specially in conditions with no LoS. Drones can be deployed quickly to areas that are inaccessible or heavily damaged areas, facilitating seamless communication among emergency personnel within the same range. This efficient coordination of emergency services leads to faster response times and, ultimately, helps save lives.

Overall, the project's motivation lies in providing Aveiro with a state-of-the-art emergency network infrastructure that leverages existing infrastructure and resource integrating modern technologies. By establishing this network, Aveiro can become a model city for other regions, inspiring similar initiatives to create resilient and efficient emergency communication systems.

[1] https://www.open-mesh.org/projects/batman-adv/wiki

1.3 Project Objectives

The principal objectives of our project are as follows:

- Utilize the ATCLL Infrastructure: The project aims to leverage the existing infrastructure of Aveiro Tech City Living Lab (ATCLL) to establish an ad-hoc network for emergency entities such as firefighters, law enforcement, and other relevant organizations. By utilizing the ATCLL infrastructure, which includes smart lamp posts and other resources such as drones, the project seeks to create a reliable and robust backbone for this network.

- Provide Internet Access: Whenever possible, the project intends to offer a internet access that complements the ad-hoc network. We could use 5G modules strategically placed on smart lamp posts to provide strong network coverage near the infrastructure. This network coverage will gradually increase as it is extended to areas covered by drones.

- Extended Network Range: Drones will be used to expand the range of both the internet access and the ad-hoc network serving as a border router and a gateway. These drones will establish connections between smart lamp posts, devices that are not physically close to each other or not in LoS. By being in high altitudes the probability of these devices having a good LoS to the drone is very high which makes it excellent to bridge the gaps in connectivity in areas that are difficult to access.

Overall, the objectives of the project revolve around taking advantage of the existing ATCLL infrastructure, with the possibility of integrating 5G technology, and utilizing drones to establish a robust emergency network. By doing so, the project aims to enhance communication and coordination among emergency entities, ensure reliable network connectivity during crises, and bridge the connectivity gaps in hard-to-reach areas.

Chapter 2

Architecture

2.1 Project Architecture

The final achieved architecture for this project consisted of three Raspberry Pis, one laptop computer and one smartphone. Firstly we had three Raspberry Pis that were running B.A.T.M.A.N. to create an ad-hoc network. One of the Raspberry Pis (RPI-3) was connected to the laptop via an Ethernet cable, and the laptop itself was connected to the smartphone's WiFi hotspot as depicted in Figure 2.1. In addition to that, our RPI-3 was configured as a gateway server on B.A.T.M.A.N.. In the context of B.A.T.M.A.N., a gateway server acts as a central point that connects multiple networks together. It facilitates communication between different devices within the network. On the other hand, the other devices (RPI-1 & RPI-2) were configured as gateway clients. Gateway clients rely on the gateway server to forward their network traffic to other network, in this case, the Internet. This setup allows for efficient routing and communication among the connected devices. With this setup we can have an ad-hoc network that is proactive. On top of that all the Raspberry Pi devices have access to Internet through the RPI-3 which only has Internet because it is connected to a device with access to it, inherently RPI-3 does not have Internet access.

In the building, we had access to the eduroam network, but it wasn't suitable for conducting tests because it blocked the NAT translation occurring on the laptop. When performing tests outside we also needed an alternative solution. To overcome this, we relied on the laptop being connected to a smartphone's WiFi hotspot. This allowed us to establish an internet connection and carry out the necessary tests.

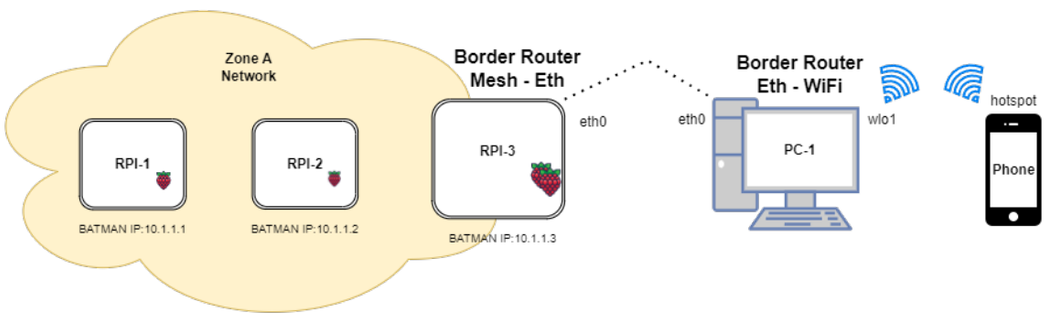

Figure 2.1: Physical architecture.

2.2 Technological view

The technologies used for this project were B.A.T.M.A.N. protocol running on the wireless interfaces of Rapsberry Pi devices with the addition of a laptop with an UNIX-based operating system capable of running an iptables shell script. In addition to that we could easily migrate to 5G if we added 5G modules to the Raspberry Pi that serves as a gateway, giving it the capability of not needing the laptop to have Internet connection.

B.A.T.M.A.N. is a routing protocol for wireless mesh ad-hoc networks. It works by discovering neighboring nodes, evaluating link quality, and selecting routes based on the strength and reliability of connections. When a node needs to send data to a destination, it broadcasts a Route Request (RREQ) message, which propagates through the network until it reaches a node with a route or a neighbour to the destination. A

Route Reply (RREP) message is then sent back to establish the route. B.A.T.M.A.N. continuously monitors link quality and dynamically adapts routes by updating routing tables to avoid using degraded connections.

2.3 Envisioned Scenario

In the envisioned scenario, the objective is to use the ATCLL infrastructure, including smart lamp posts and drones, to establish a robust emergency network. This network would serve critical personnel and their vehicles during emergency situations, but can also enable citizens of Aveiro to communicate with each other and have Internet access which is shown on Figure 2.2.

The smart lamp posts are ideal for forming the backbone of the network. Because they are stationary and in an elevated position, these posts provide a good LoS to the drones, allowing for a stable communication and minimizing signal disruption. On top of that, these posts are interconnect by fiber optic cable, therefore, all posts can be a reliable node to our network providing high-bandwidth.

In complement to the smart lamp posts, drones can play a crucial role in bridging communication gaps between devices and expanding greatly the range of the network. Drones can be deployed at higher altitudes, taking advantage of their ability to maintain LoS with a greater number of devices. By acting as aerial relays, drones establish connections between devices that are located further apart and may not have direct LoS or are in areas with challenging conditions.

Another thing to take into consideration is that there are a lot of vehicles specially the ones used by firefighters that normally stay parked in critical areas, by being stationary they provide a stable and reliable network. We recon that taking advantage of these vehicles will greatly improve the coverage of the network.

The correlation between the project's architecture depicted in Figure 2.1 and the envisioned scenario illustrated in Figure 2.2 is readily apparent. While the prototype architecture has a physical interface for connecting the gateway to the Internet, we could have incorporated an additional wireless interface with a 5G module. Although our current prototype lacks static devices at high altitudes, conducting experiments and tests readily demonstrates that such additions would enhance the network range and performance.

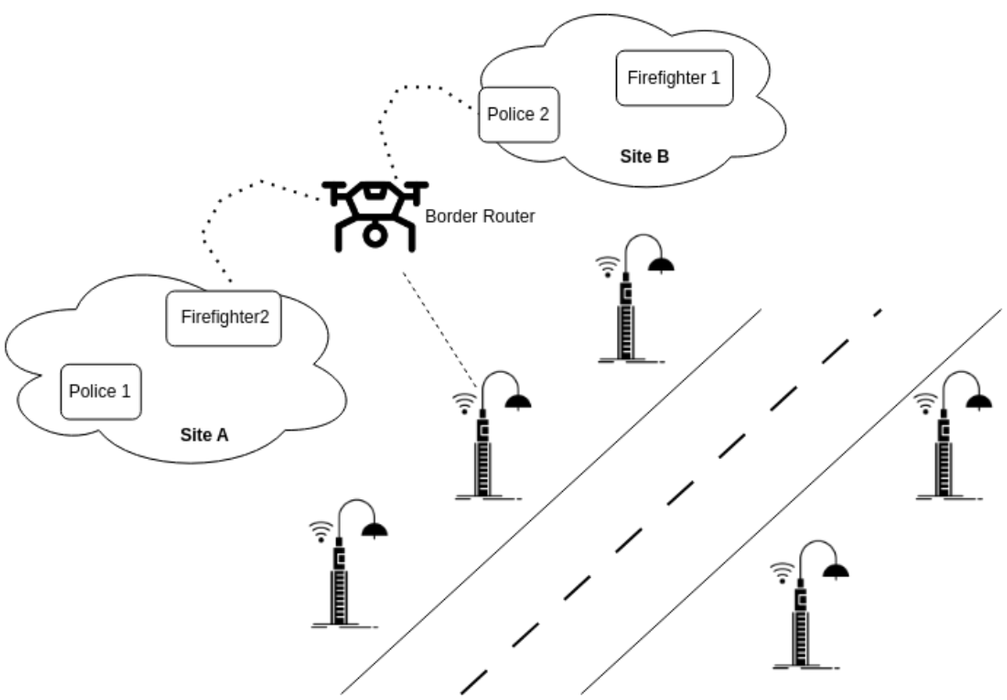

Figure 2.2: Envisioned scenario.

Chapter 3

Implementation

3.1 Configurations

These were the B.A.T.M.A.N. configurations:

- All Raspberry Pi devices are running B.A.T.M.A.N. on their WiFi interface (wlan0) which is the only wireless interface available to the device.
- The B.A.T.M.A.N. ad-hoc network SSID is "adhoctest" using channel 11 and a frequency of 2462MHz.
- RPI-3 must run `sudo batctl gw_mode server` so it will be a gateway and act as a central point that connects multiple networks.
- RPI-1 & RPI-2 must run `sudo batctl gw_mode client` which will rely on the gateway server to forward their network traffic to other networks.
- It is recommended to run `sudo batctl gwl` on RPI-1 & RPI-2 to verify that they have a gateway and this gateway is the RPI-3.

Device specific configurations:

- Devices have a static IP "192.168.2.X" with X being the Id of the device.
- The laptop directly connected to the RPI-3 has a static IP (192.168.2.10/24).
- The laptop must be connected to the a network with an internet connection other than eduroam to be functional, this is because eduroam.
- RPI-1 & RPI-2 must have a default route via the RPI-3's IP.

Internet configurations:

- Run the iptables script[1] (`sudo ./iptables.sh wlo1 eth0`) on the laptop computer bridging the interface connected to RPI-3 via Ethernet cable and the interface that has connection to the Internet (probably a WiFi interface).
- Run the iptables script (`sudo ./iptables.sh eth0 bat0`) on RPI-3 device bridging the B.A.T.M.A.N. interface (wlan0) and the interface connected directly to the laptop via Ethernet cable (eth0).

3.2 Dashboard Webapp

For the dashboard presented on the Demo section, we developed a React Webapp and a NodeJS API. The NodeJS API acts as a bridge between the device that sends statistics and the dashboard itself. It has two endpoints, one for storing data on POST /batman and another GET /batman to retrieve the information.

The device that is processing statistics is using POST /batman to send the data to the API which will store the received data on a MongoDB database with the current timestamp. The dashboard will call GET /batman every 500ms to retrieve the last 30 values from the database and plot the values on three dynamic graphs, one for RTT vaues, one for Jitter and another for PDR.

[1] This script sets up network forwarding and masquerading rules to enable network communication between devices on a local network and the internet, with appropriate security checks to allow established connections and traffic between specified interfaces.

3.3 Message Flow

In our network communication, various messages are exchanged that include routing information dissemination and data. The following is a concise explanation of these messages and their functionalities:

Originator Message (OGM): OGM serves as the primary means of disseminating routing information throughout the B.A.T.M.A.N. network. It is a broadcast message sent by nodes and contains important details such as transmission quality, translation table, multicast information, and distributed ARP table. OGMs are used to discover and maintain routes to other nodes within the network.

OGM Rebroadcast: This process involves nodes forwarding OGMs they receive from their neighbors. By rebroadcasting OGMs, nodes actively contribute to the dissemination of routing information across the network. This mechanism helps nodes discover and maintain routes to other nodes, ensuring efficient and reliable routing throughout the network.

Unicast TVLV (Type-Length-Value) Messages: Unicast TVLV messages are used in B.A.T.M.A.N. for exchanging unicast information between nodes. These messages can take various types, such as Translation Table Request and Translation Table Response. They employ the TVLV structure to request and share translation table entries enabling targeted communication between specific nodes and facilitating the exchange of essential information.

In addition to these routing and control messages, data is simultaneously exchanged between devices using protocols such as UDP, TCP and others.

Furthermore, during the configuration of the network gateway, OGMs sent by the gateway will include TVLV Gateway Information. These OGMs serve the purpose of announcing the node as a gateway and sharing the download and upload speeds associated with it.

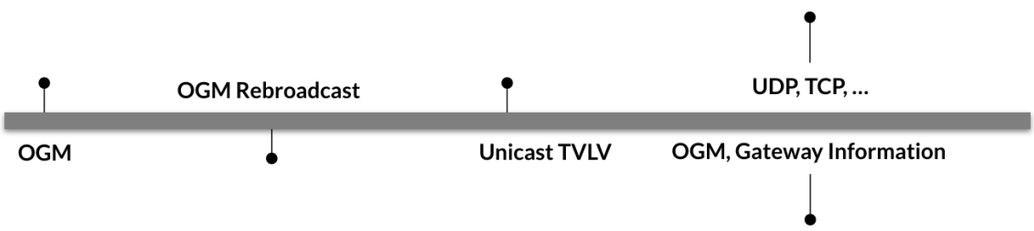

Figure 3.1: Visual message timeline.

Chapter 4

Testbed Experiments

4.1 Test Methodology

In this project, we conducted experiments to evaluate device communication at a height of approximately 1.5 meters. The objective was to assess the effectiveness of the network connectivity and communication capabilities between devices within the ad-hoc network and the Internet.

To determine the distance between devices, we used the GPS functionality available on our smartphones and the coordinates obtained from the GPS were used to calculate the distance between the devices. It is important to note that the measuring equipment used for these experiments has limitations in terms of precision.

However, the team can assure a certain degree of consistency in the tests conducted. This means that although the measurements might not be absolutely precise, they are reliable and provide consistent results that can be used to evaluate the performance of the network .

These were the test scenarios:

- 1 Hop - This scenario uses two devices (one running Iperf server and the other Iperf client). We only tested the ad-hoc network capabilities and the devices where in LoS.
- 2 Hop - This scenario uses three devices (one running Iperf server, one running Iperf client and another in the middle serving as a middle hop). We tested both ad-hoc network capabilities and access to Internet, keeping in mind that the devices where in LoS.
- No LoS Inside - This scenario uses two devices (one running Iperf server and the other Iperf client). We only tested the ad-hoc network capabilities and the devices where in not in LoS, they were inside a building having on between the devices a closed door, walls and hallways.
- No LoS Outside - This scenario uses two devices (one running Iperf server and the other Iperf client). We only tested the ad-hoc network capabilities and the devices where in not in LoS, they were outdoors and having a building corner between the devices.

4.2 Results

One of the first things we tested was the maximum distance achieved. We quickly realised that under LoS conditions, we achieved a maximum distance of roughly 200-230 meters with a single hop. With the addition of an intermediate device, serving as an extra hop, this range was extended to approximately 280-300 meters visualized in Figure 4.1. However, these distances can vary depending on the atmospheric conditions.

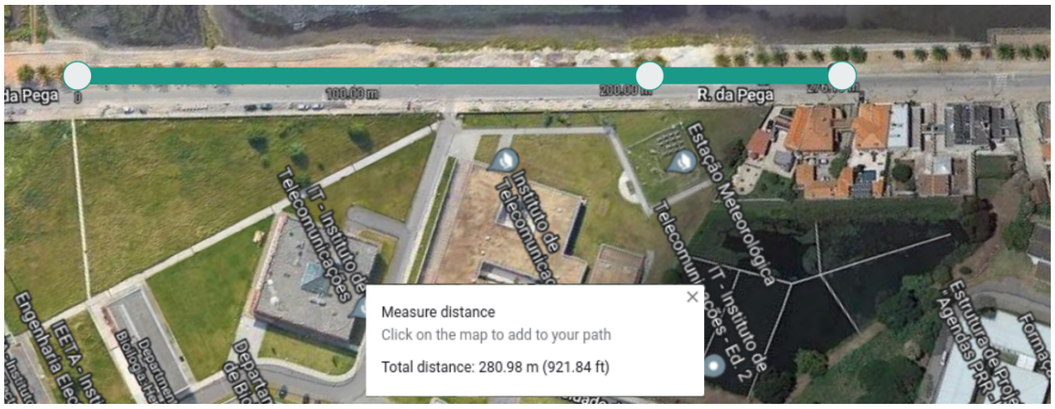

Figure 4.1: Maximum distance achieved with LoS.

4.2.1 RTT Results

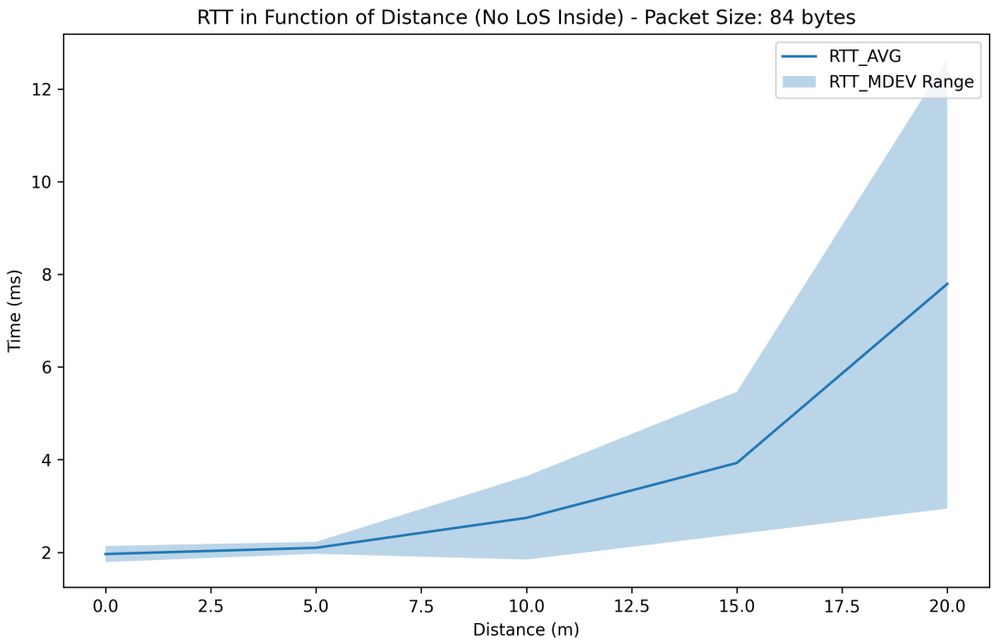

Figure 4.2: RTT in Function of Distance (No LoS Inside).

From the graph depicted in Figure 4.2 we see the RTT values in function of the distance between devices in a scenario of no LoS inside a building. It is evident that there is an exponential increase in RTT values has the distance between devices increases. The maximum distance between devices in this scenario was approximately 20 meters. It is worth noting that despite this trend, the RTT values remain consistently low, below 15ms.

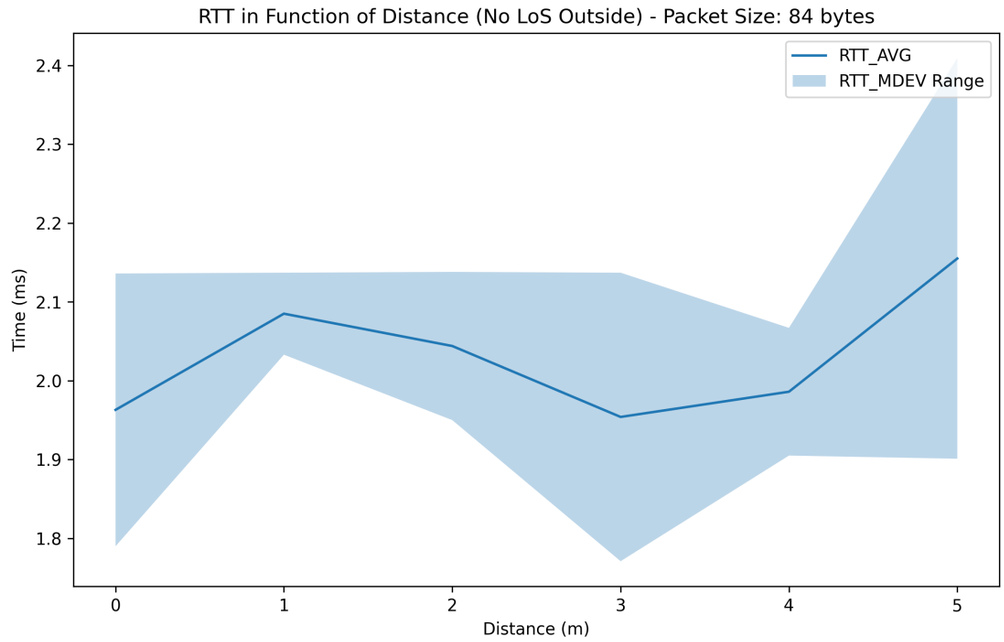

Figure 4.3: RTT in Function of Distance (No LoS Outside).

From the graph depicted in Figure 4.3 we see the RTT values in function of the distance between devices in a scenario of no LoS outside a building. In this case the variation of the RTT values is not so evident as before. It is also relevant to add that in this case the Raspberry Pis were positioned in the corner of a building and that the maximum distance that we achieved between devices was 5 meters.

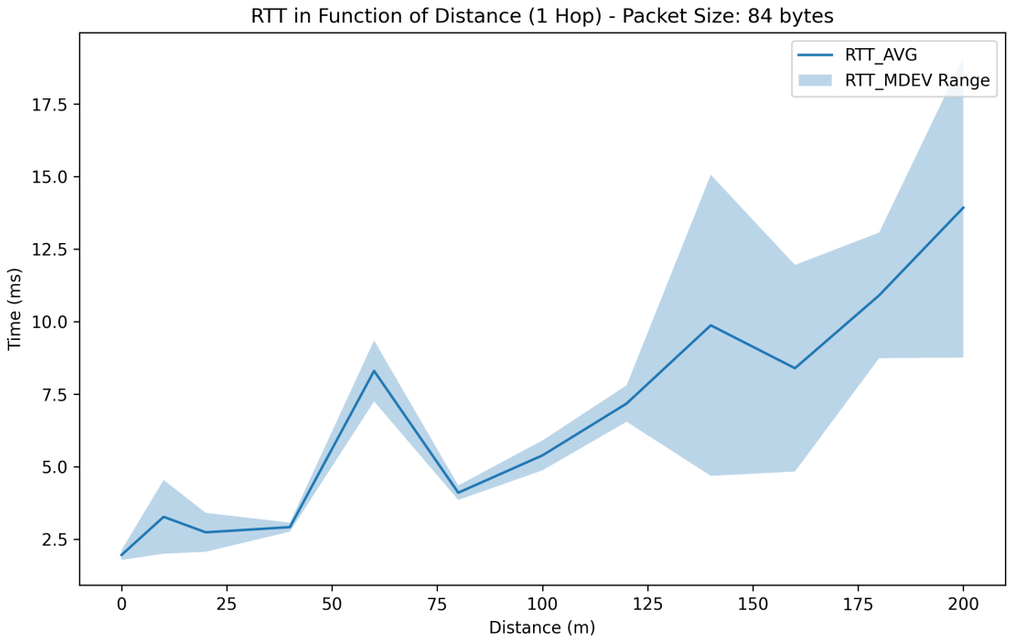

Figure 4.4: RTT in Function of Distance (1 Hop).

From the graph depicted in Figure 4.4 we see the RTT values in function of the distance between devices in a scenario of LoS with 1 Hop. In this case the variation of the RTT values is clearly trending exponentially as the distance between devices increases. However, there are some points where the values obtained don't follow the expected trend. This can be explained by our test methodology, as the distance between devices increases so did the time for our tests, and therefore we only did one measurement per distance tested which could cause the variation observed. In this scenario we were capable of achieving 200 meters between devices.

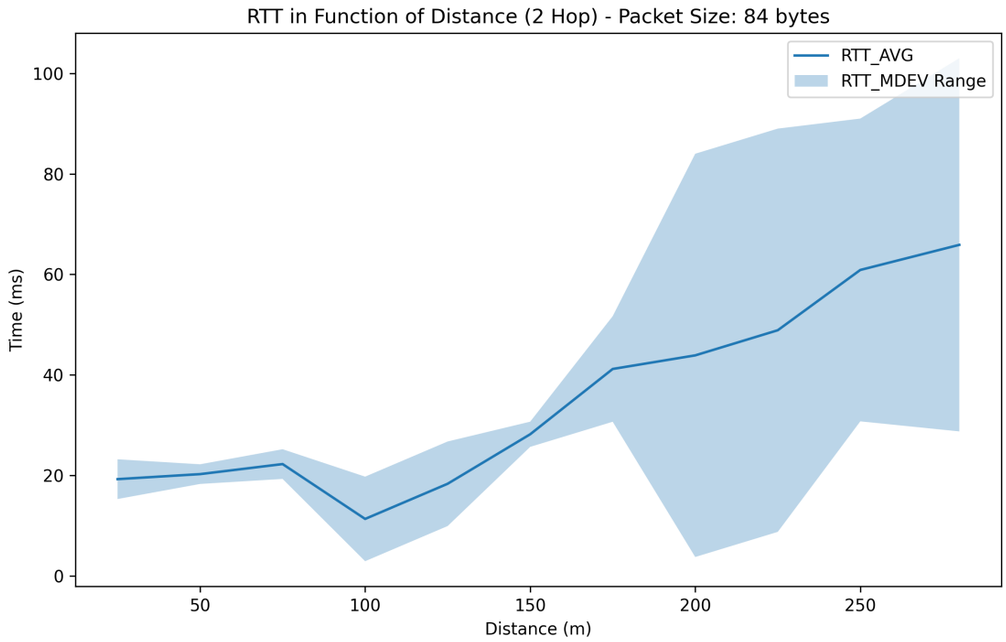

Figure 4.5: RTT in Function of Distance (2 Hop).

From the graph depicted in Figure 4.5 we see the RTT values in function of the distance between devices in a scenario of LoS with 2 Hops. In this case the variation of the RTT values observed follow the same trends and issues as before. One thing to note is that we can see a clear increase on the RTT values when comparing the same distances with the scenario with 1 Hop. In this scenario we were capable of achieving 280 meters between devices.

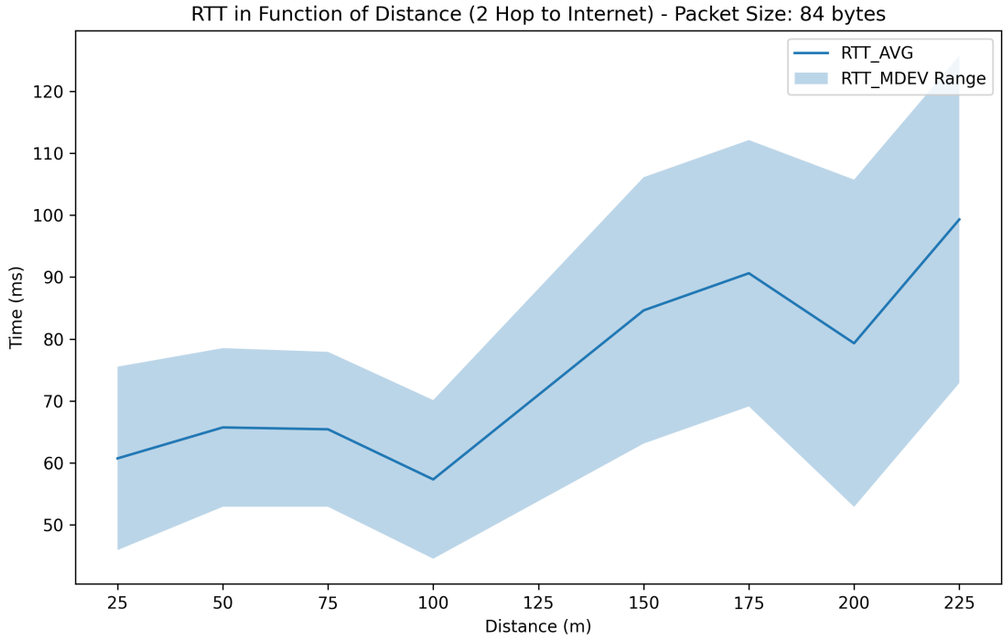

Figure 4.6: RTT in Function of Distance (2 Hop to Internet).

As for the same test done previously but now using an Iperf public server on the Internet we can see that on the graph depicted in Figure 4.6 the RTT values observed follow the same trend and issues as the other tests in LoS. The only thing to note is that the RTT values obtained in this scenario were the highest obtained for the LoS scenario. This is to be expected because not only do we have 2 Hops to communicate between the device and the gateway but we are also communicating with the Internet which adds a constant delay in that process.

4.2.2 Jitter Results

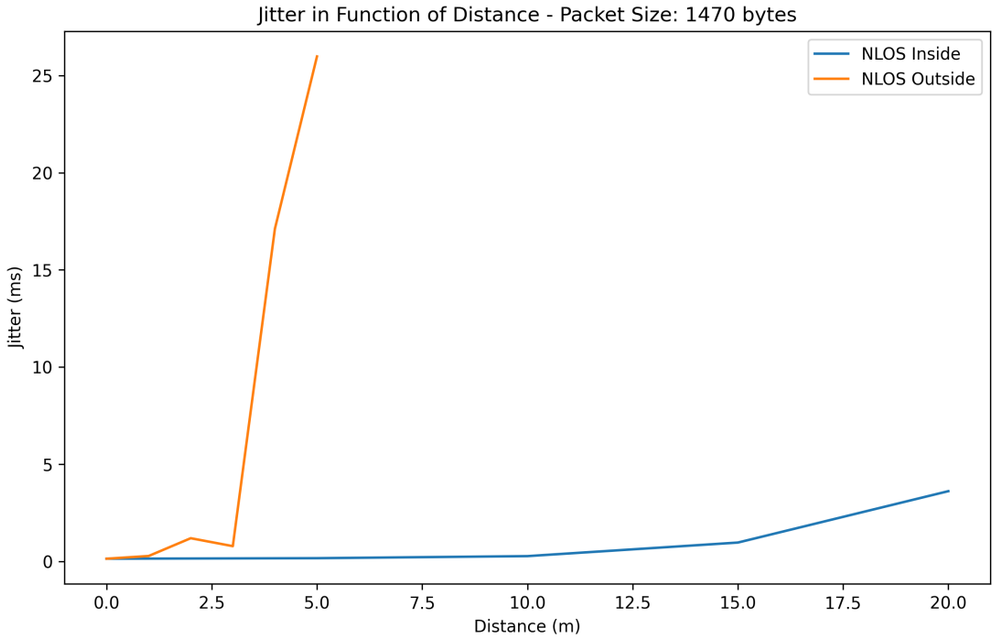

Figure 4.7: Jitter in Function of Distance (No LoS).

In Figure 4.7 we can see a graph which shows the Jitter obtained in function of the distance between devices in both scenarios of no LoS. Has expected, we observe that as the distance between devices increases so does the Jitter value. With the indoor scenario producing significantly lower values when compared with the outdoor scenario. This is to be expected since the indoor scenario consists only one wall and a spacious hallway, enabling signal bounce and reflection on the walls, resulting in improved distances and values. Additionally, it is worth mentioning that in the outdoor scenario, the LoS is obstructed by the corner of a building rather than a smaller object like a car, further degrading the signal quality.

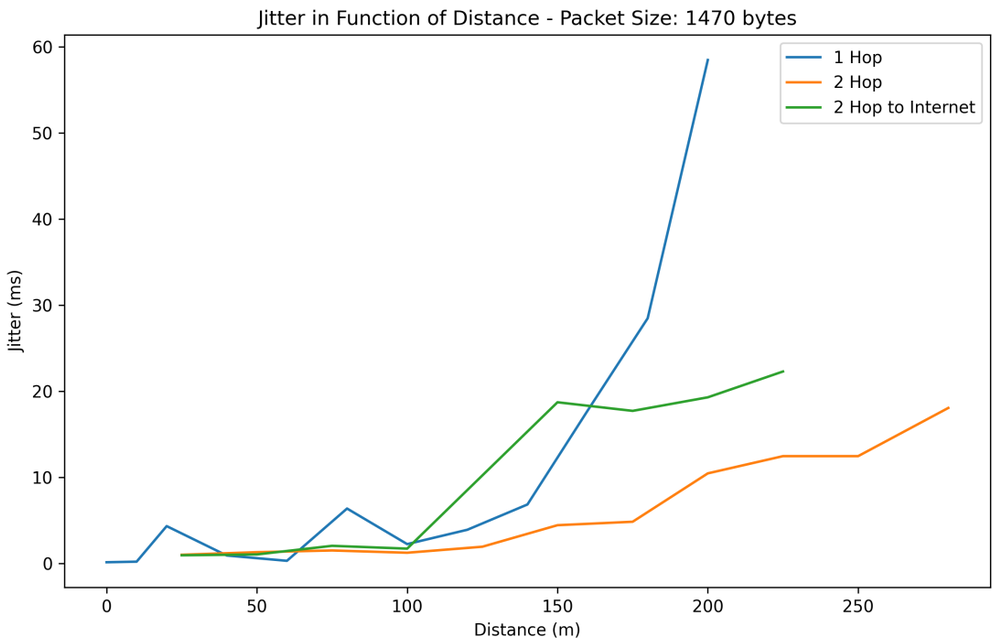

Figure 4.8: Jitter in Function of Distance (LoS).

In Figure 4.7 we can see a graph which shows the Jitter obtained in function of the distance between devices on all LoS scenarios. There is a clear trend that shows

that as the devices go further from each other, the Jitter value increases, we also note that generally the values obtained for scenario with 1 Hop are worse than the values obtained for the scenario with 2 Hops. This is to be expected because even though the RTT values are worse for the multi hop scenario, more hops generally mean more stability, especially when comparing to a scenario that only involves two nodes (1 Hop scenario).

When comparing the values between the 2 Hops to the Internet and 2 Hops we can observe the expected behaviour where Jitter is higher when the route includes the Internet.

4.2.3 PDR Results

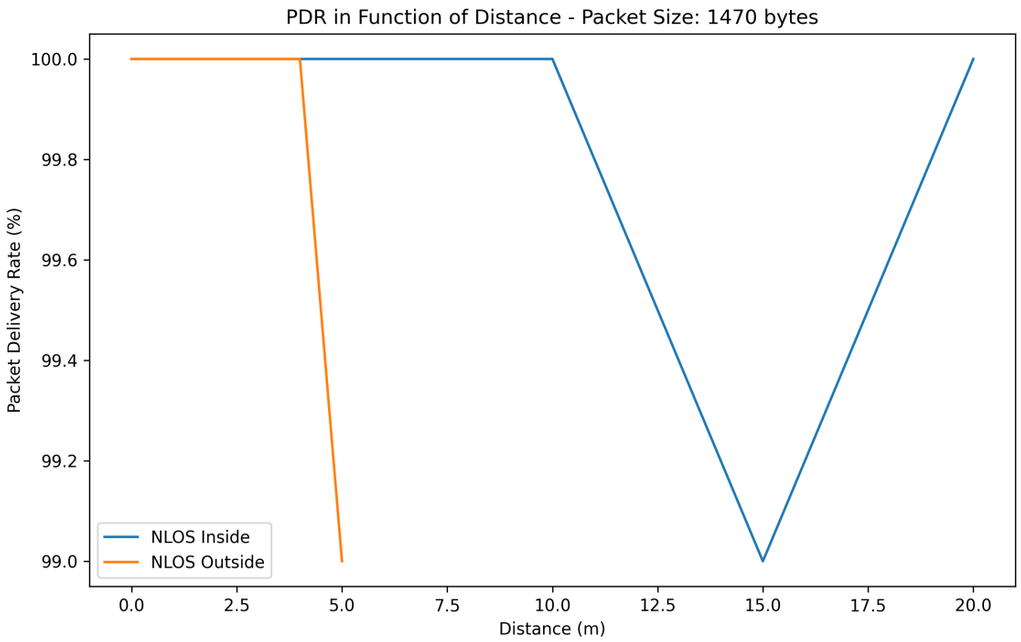

Figure 4.9: PDR in Function of Distance (No LoS).

In Figure 4.9 we can observe the Packet Delivery Ratio in function of the distance between devices in both scenarios of no LoS. It is evident that the PDR remains consistently high, occasionally dropping to 99%. The reasons behind these fluctuations have been explained previously.

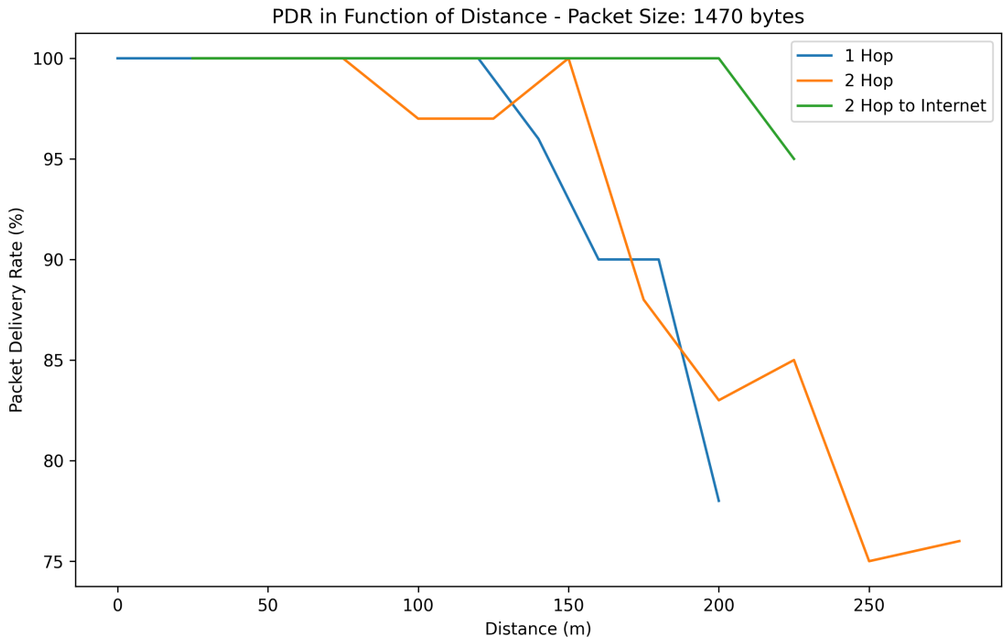

Figure 4.10: PDR in Function of Distance (LoS).

In Figure 4.10 we can observe the Packet Delivery Ratio in function of the distance between devices on all LoS scenarios. The observed trend indicates that as the distance between devices increases, the PDR tends to decrease. It is worth noting that the PDR values for the scenario with only 1 Hop show a faster decline compared to the values for 2 Hop. However, it is important to highlight that even though the minimum PDR value for 2 Hop is lower than that of 1 Hop, using 2 Hops allows for greater distances between devices which would normally produce higher PDR values.

The values obtained for the scenario with 2 Hops to the Internet do not match what the team was expecting. To make sense of these results, the team speculated that as the Internet test was conducted after the 2 Hop scenario, it could have led to a more stable B.A.T.M.A.N. network, decreasing the PDR values. This could be again verified if more tests were conducted.

4.2.4 Throughput Results

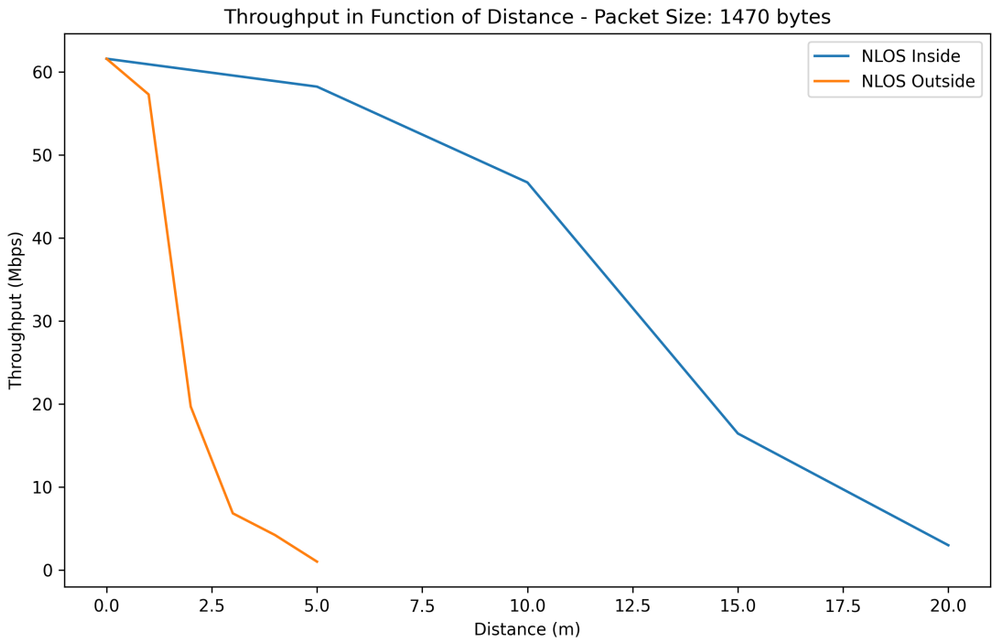

Figure 4.11: Throughput in Function of Distance (No LoS).

The Figure 4.11 above depicts the Throughput obtained in the communication in function of the distance between devices in both scenarios of no LoS. It is evident that the Throughput values decrease as the distance between devices increases, which aligns with our expectations. Furthermore, as previously explained in the section analyzing Figure 4.7, it was anticipated that the indoor scenario would generally get better Throughput values compared to the outdoor scenario.

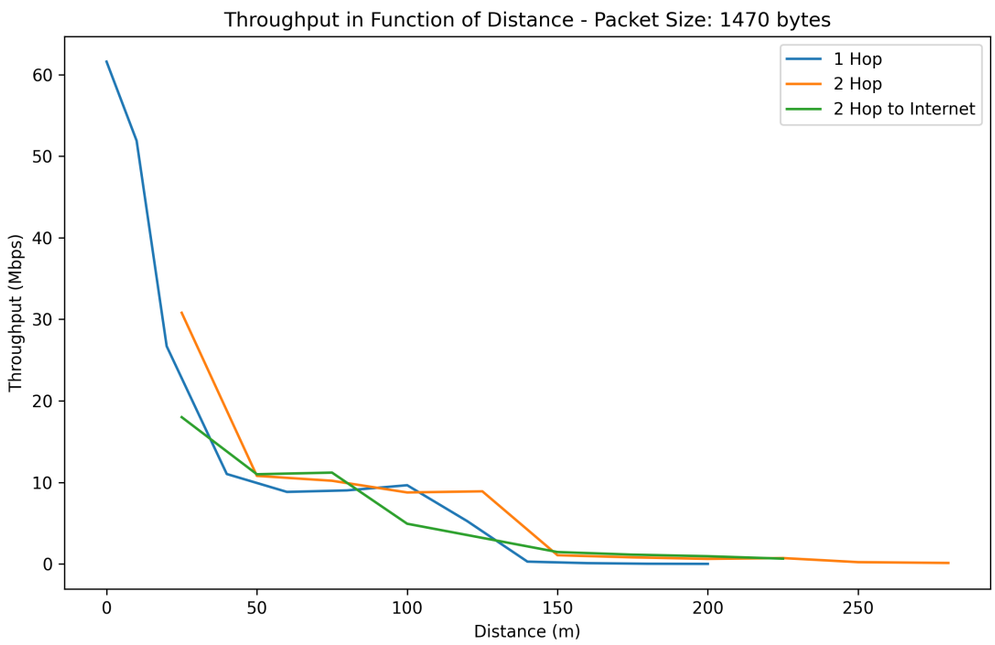

Figure 4.12: Throughput in Function of Distance (LoS).

The Figure 4.12 above depicts the Throughput obtained in the communication in function of the distance between devices on all LoS scenarios. As anticipated, the results depicted in this graph align with expectations. Generally, the throughput values in a 2 Hops scenario tend to be better when compared to the same distance covered by only 1

Hop. Additionally, when comparing the Throughput values in the same scenario, those obtained for an Iperf server in the Internet are slightly lower. This can be attributed to the constant overhead introduced by the additional time required for communication with the Internet.

On a particular note, the team finds confidence in the results and the distances obtained due to the observation that the minimum Throughput values for both scenarios are similar. These minimum values occur at the maximum distance between devices, which reinforces the validity of the obtained results.

4.3 Demo

We had two approaches for the demo of our project. Firstly we will setup two Raspberry Pi devices as depicted in Figure 4.13 inside a building one of them (the client) will go further away from the other (the server) that is transmitting the values of RTT, Jitter and PDR. The second part of the demo consists in the integration of a service called "Toxcore" that enables us to simulate a real-time messaging service between the devices on the ad-hoc network.

4.3.1 Demo Architecture

In this section of the demo we will have RPI-2 running an Iperf Server and RPI-3 runing an Iperf client having it's server to be the one on RPI-2. The RPI-3 is stationary and connected to Web Server API which will store the values of RTT, Jittter and PDR. The RPI-2 will not be stationary and will move away from the cient going through some walls and hallways. The dashboard will periodically fetch data sent from RPI-3 with the desired metrics.

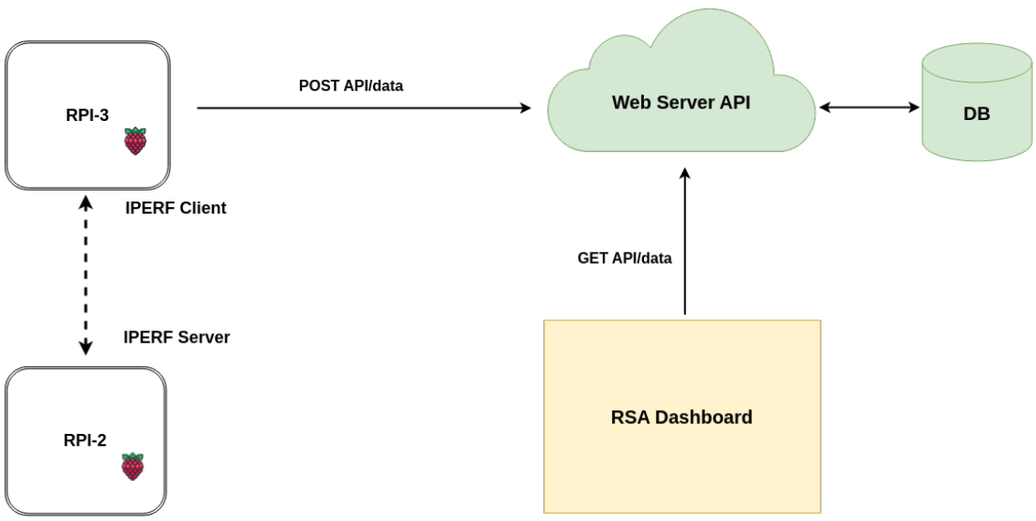

Figure 4.13: Architecture for the demo.

4.3.2 Dashboard Results

In Figure 4.14 we can clearly observe that as the devices grow further apart we see a substantial increase on the RTT and Jitter values, while the PDR values decrease slightly.

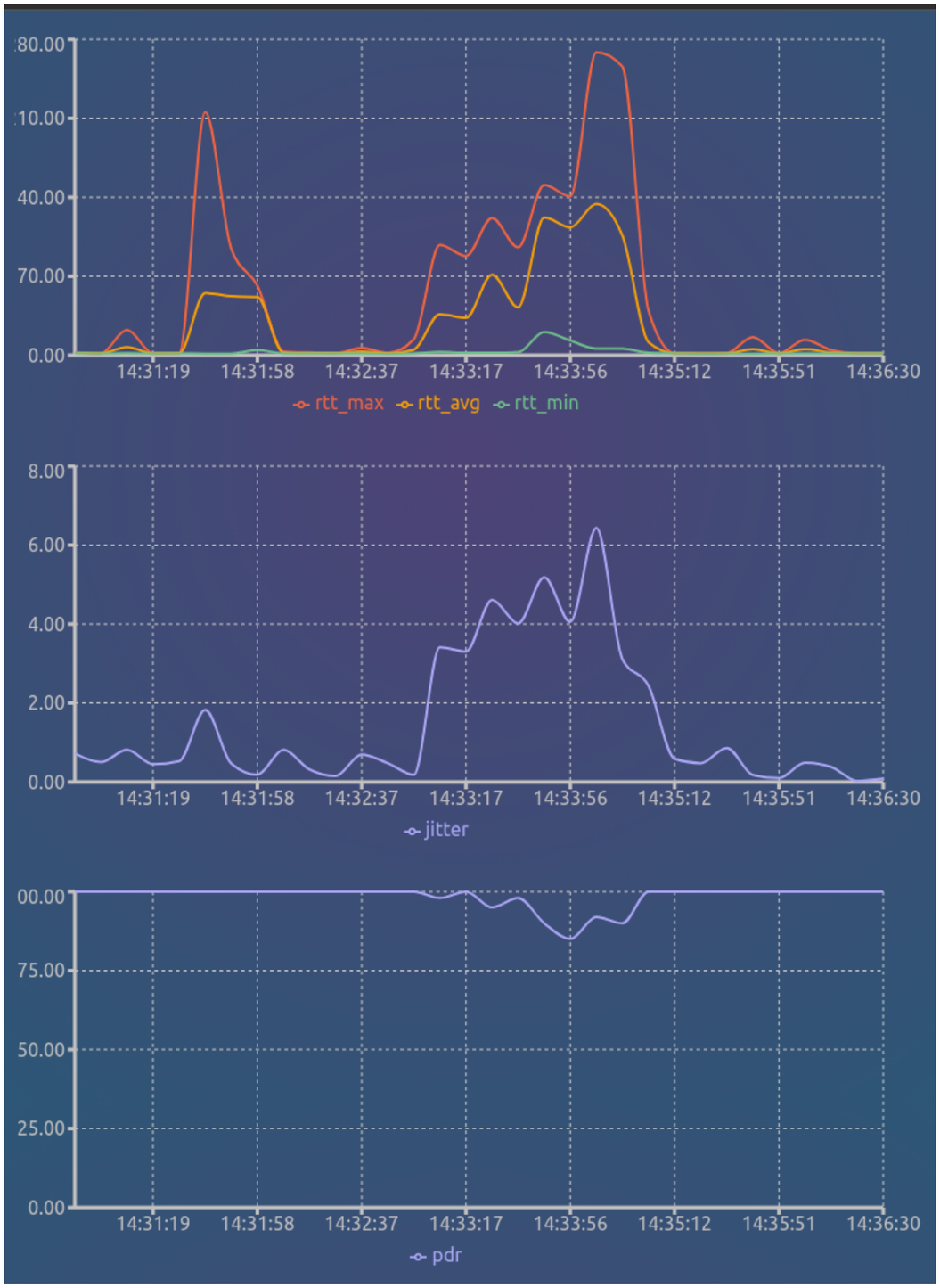

Figure 4.14: Dashboard Results for the demo.

4.3.3 Messaging Service

In our project, we successfully integrated a service called "Toxcore" into the emergency network infrastructure. This service enables real-time communication between devices through text messages and voice, similar to a normal social media application. By incorporating Toxcore, we aimed to showcase the capabilities of the network and explore potential use cases.

It is worth mentioning that although Toxcore showcases the network's communication capabilities, it may not meet specific requirements, such as ultra-low delay, reliability, and user privacy, which are essential in sensitive emergency situations. Therefore, it may not be suitable for a production-level scenario in those cases. However, the inclusion of Toxcore serves as a valuable indicator of the network's potential and opens up possibilities for future enhancements and features.

We will demonstrate the following scenarios:

- Contact list - As depicted in Figure 4.15 Toxcore enables users to create a list of contacts. This contact list allows users to establish individual communication channels with their contacts. Additionally, Toxcore provides information

on the last seen online status of each contact, offering valuable insights into the availability and presence of users within the network.

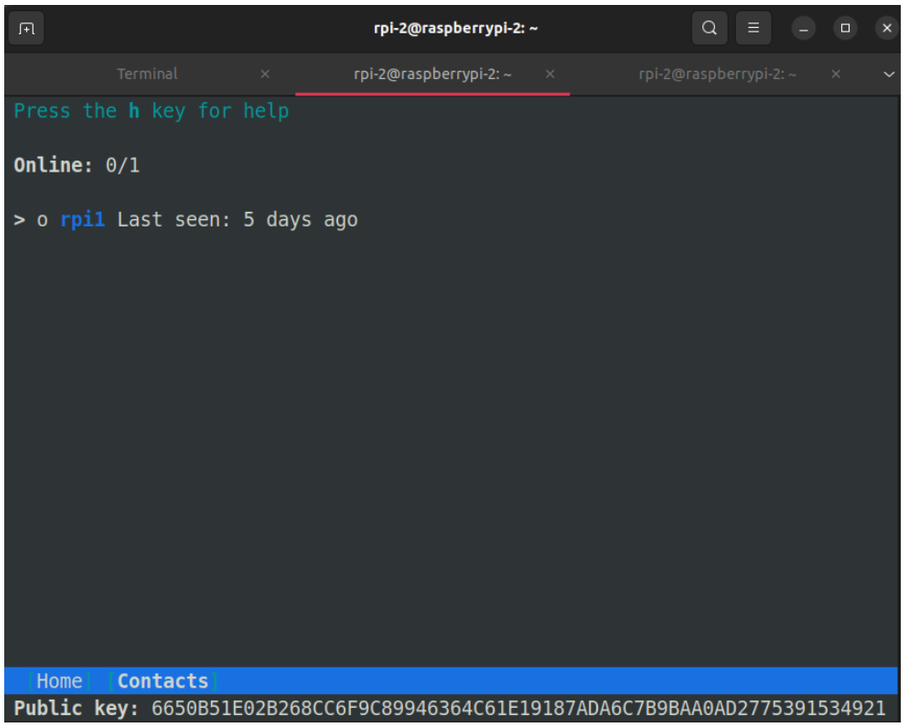

Figure 4.15: Contact list.

- Message chat - As depicted in Figure 4.16 Toxcore facilitates message chat, allowing users to engage in real-time text-based conversations with their contacts individualy and in groups.

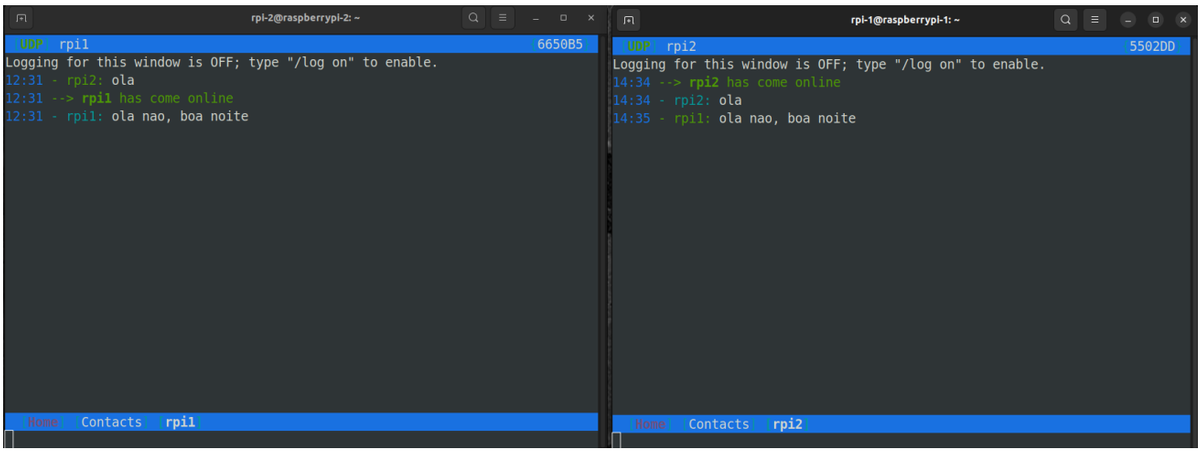

Figure 4.16: Message chat.

- Message acknowledgement - As depicted in Figure 4.17 Toxcore includes message acknowledgement capabilities, providing users with confirmation that their sent messages have been received by the intended recipients.

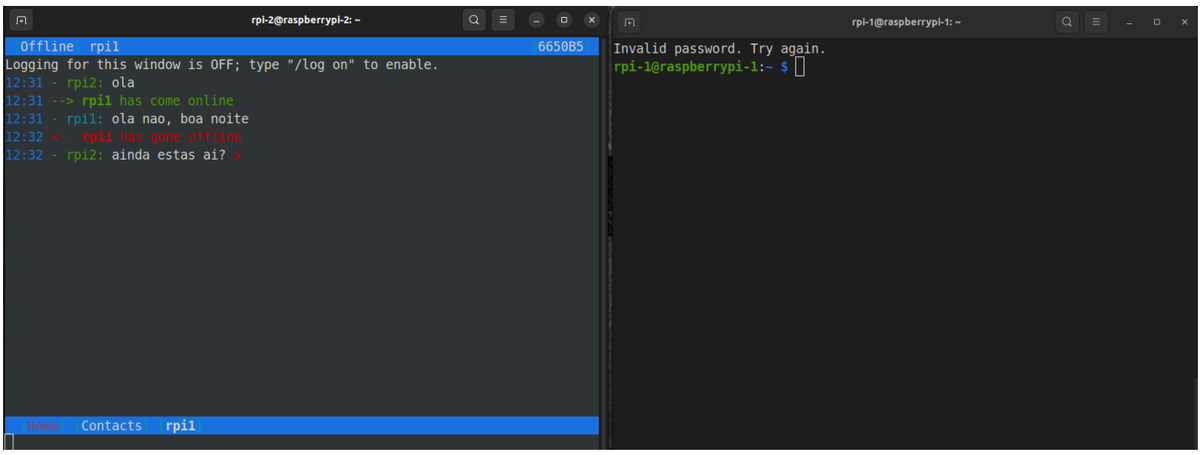

Figure 4.17: Message and state acknowledgment.

The important takeaway from this demonstration is that we have successfully demonstrated the feasibility of UDP communication between devices within the network. This means that it is highly plausible to deploy a dedicated communication service that allows for targeted messaging to specific users, groups of users, broadcast messages to the nearest zone for sending alerts or emergency notification or even voice communication.

Chapter 5

Conclusion

In conclusion, this project has successfully constructed a scenario that closely aligns with the intended use case of the emergency network infrastructure depicted in Figure 2.2. Through testing and evaluation, the implementation of the network has demonstrated its ability to handle emergency situations and deliver efficient communication services to emergency entities such as firefighters, police, and other relevant organizations.

In addition to the core functionality of the emergency network, the project has also incorporated an additional feature with the integration of "toxcore". This service emulates a social chatting platform, enabling seamless communication among users across devices within the ad-hoc network. By integrating this service, the project not only provides essential emergency communication but also shows the feasibility of sending alerts and other related actions that have the same communication principal and are useful in an emergency scenario.

In short, this project has demonstrated the feasibility and effectiveness of the proposed emergency network infrastructure. It provides a reliable communication system for emergency entities while incorporating additional features to enhance social interaction during crises and system wide alerts. The tests realized have ensured that the network can perform in real-life scenarios, contributing to the safety and well-being of the population of Aveiro in emergency situations.

Chapter 6

Future Work

Based on the current project state, there are several areas for future work and enhancements that can improve the emergency network infrastructure:

- Upgrading the Network Card: The devices used were Raspberry Pis, these devices have average network cards, to enhance performance and coverage, it is possible to integrate a more advanced network card. This should offer improved signal strength, faster data transmission rates, and broader coverage capabilities significantly. On top of that, with a better network card we could test the Received Signal Strength Indicator (RSSI) values that express the quality and strength of the wireless signal between devices.
- Adding a Small Antenna: Our specific model of Raspberry Pis 4 have an enclosed case that further degrades the wireless signal as depicted in Figure 6.1. Introducing a small antenna will extend the network's range and improve signal reception enabling devices to overcome obstacles and reach devices that are located further away.

Figure 6.1: Raspberry Pi 4 used.

- Adding a 5G Module: To enhance Internet connectivity and reduce reliance on a laptop, integrating a dedicated 5G module (similar to the one depicted in Figure 6.2) on the device to communicate with the infrastructure. The addition of a 5G module would provide high-speed internet access to devices within the network, enabling access to online resources during emergencies.

Figure 6.2: 5G module.

- DHCP Server on gateway: Implementing a DHCP server on device that server as a gateway enables the other devices in the network to receive a dynamic IP addresses automatically. The DHCP server can also be configured to assign the appropriate default route to the connected devices. This ensures that each device has the correct routing information to access external networks, such as the Internet.

  This eliminates the need for manual IP and default route configuration on each device, making it easier to scale the network as more devices are added or removed.

Chapter 7

Resources

Link to code repository with the demo: `https://github.com/andreclerigo/mect_1ano/tree/main/RSA/Projeto`