UNIVERSITY OF
THESSALY

Συλλογή μετρήσεων και σύνδεση τοπικών κοινοτήτων μέσω DIY ασύρματων δικτύων

Φοιτητής:
ΜΑΥΡΙΔΗΣ ΙΩΑΝΝΗΣ

Επιβλέπων:
ΚΟΡΑΚΗΣ ΑΘΑΝΑΣΙΟΣ
Επίκουρος καθηγητής

Συν επιβλέπων:
ΑΡΓΥΡΙΟΥ ΑΝΤΩΝΙΟΣ
Επίκουρος καθηγητής

UNIVERSITY OF THESSALY

Sensing and Connecting Local Communities through DIY Wireless Networks

By:
Mayridis Ioannis

Supervisor:
Korakis Athanasios
Assistant Professor

Co-Supervisor:
Argyrioy Antonios
Assistant Professor

Διπλωματική Εργασία για την απόκτηση του Διπλώματος του Μηχανικών Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

....
Μαυρίδης Ιωάννης
Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας

Acknowledgments

The work of this thesis has been one of the most significant academic challenge I have faced so far. Without the support, patience and guidance of the people around me this thesis would not have been completed. First of all, my enormous debt of gratitude goes to my thesis advisor and mentor Assistant Professor Athanasios Korakis. I am thankful for his knowledge and his belief in me, which gave me the opportunity to work in the company's facilities and also for letting me use their technological equipment in order to accomplish this thesis purposes.

I would also like to thank my colleague Charilao Niavi, for providing important feedback for my thesis and the support required to overcome the initial problems encountered and keep working with passion. Also, throughout the period of my studies, he was there to actively support and guide me toward taking the best possible decisions.

Especially, I would like to thank my friends and my family, for supporting me every day. I totally believe that great achievements require friends to celebrate, and throughout my lifetime I was always trying to meet new people and make new friends.

Most importantly, I would like to thank my wife Anastasia and my little son Odysseus for their support. There are no words that can express my gratitude and appreciation for all they have done for me. Without their patience and support, I would not have been able to finish my thesis. The least I can do in recognition is to dedicate this thesis to them.

Dedicated to

My family

Περίληψη

Στην σύγχρονη εποχή τα αυτό-δημιουργούμενα ασύρματα κοινοτικά δίκτυα συνεχώς αυξάνονται, προσπερνώντας με αυτό τον τρόπο την ανάγκη ενός κεντρικού συντονιστή για την υλοποίηση του. Η δημιουργία και η ύπαρξη ενός κοινοτικού ασύρματου δικτιού, οπού οι χρήστες του θα την δυνατότητα αλληλεπίδρασης μεταξύ τους αποτελεί επιτακτική κοινωνική ανάγκη.

Σε αυτή την διπλωματική εργασία μελετήσαμε τον τρόπο δημιουργίας και λειτουργίας ασύρματων κοινοτικών δικτυών με απτότερο σκοπό την ανάπτυξη μια αυτό-δημιουργούμενης δικτυακής εργαλειοθήκης. Αυτή η εργαλειοθήκη παρέχει την δυνατότητα σε πολίτες που δεν έχουν ειδικευμένη γνώση στην επιστήμη των ασύρματων δικτυών να δημιουργήσουν ένα για να καλύψουν τις ανάγκες της τοπικής τους κοινότητας. Η εργαλειοθήκη αυτή περιλαμβάνει διάφορες εφαρμογές ανοιχτού λογισμικού, αισθητήρες και υπηρεσίες και με σκοπό να βελτιώσει την καθημερινή αλληλεπίδραση των πολιτών μια κοινότητας.

Abstract

Nowadays, Wireless community networks (or wireless communities) are constantly growing, thus overcoming the need for a central coordinator to implement it. The creation and existence of a wireless community network where its users will be able to interact each other is a vital civil urgency.

In this diploma thesis we studied how to create and operate wireless community networks with the tangible purpose of developing a Do-It-Yourself (DIY) network Toolkit. This toolkit enables citizens who have no specialized knowledge in the science of wireless networks to create one to cover the needs of their local community. This toolkit includes a variety of open-source software, sensors and services, in order to improve the daily interaction of community citizens.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past decade, the proliferation of open source technology such as Arduino, Raspberry Pi and the growing popularity of crowdfunding platforms have fostered the design of new sensing technologies that allow citizens to collect and share all sorts of data, from temperature and humidity to air quality or noise pollution and nuclear radiation.

These new technologies aim to empower self-organized groups of citizens by providing Do-it-yourself (DIY) or more open systems that they can appropriate to tackle matters of concern, learn about technology and the environment.

While affordable technologies could foster the emergence of new forms of environmental monitoring, empowering citizens to create a wireless local network in order to collect, share and use data in their own terms, previous studies have revealed a number of technical and social issues that can hinder the appropriation of sensing practices at the grassroots level.

A key concern is that laypeople often fail to effectively create a wireless local network by using low-cost hardware because they lack technical skills and the systems' user experience is poorly designed. Naturally, failing to set up and connect a sensor on these platforms often leads to frustration and disengagement; to missed opportunity to contribute valuable data.

Understanding the above problems, in this thesis we have implemented a device where is responsible for creating and managing a local wireless network to meet the needs of a local community

1.2 Thesis Objective

This thesis objective is a comprehensive investigation on the wireless community networks existed in these days around the world, studying their specials characteristics, as well as the innovation of these, resulting in design and implementation of a compact device which allows to no-technical citizens to create a local wireless network for their local communities. That network called Local Community Network, and the members in this network could have the ability to interact, synchronize and inform. According to fulfill this scope, we worked on the following:

- A comprehensive investigation of the low-cost hardware has been used to implement this device.

- A comprehensive investigation appropriate software meeting the needs of the local community.

- Designing and implementation of a Low-Level Command Line Interface that is the backbone of this device.

- Designing and implementation of a Data Collection Framework that offers the ability to collect useful information from a local community.

- Conducting wireless experiments in NITOS [1] Future Internet Facility in order to integrate various USB dongle antennas.

Chapter 2

Background

In this chapter we will delve deeper into the Community Wireless Network sector, descripting the architecture, the special characteristics and the innovation of these. We will also refer to a related project that we studied, understanding in that way the requirements for the community wireless network implementation.

2.1 Community wireless network

Community wireless networks (CWNs) have long been a testing-ground for the precepts of community informatics (CI), defined as "a sustainable approach to community enrichment that integrates participatory design of information technology resources, popular education, and asset-based development to enhance citizen empowerment and quality of life" (Stoecker, 2005). Depending on their design, CWN projects may serve many of these goals in different ways; yet it depends on how the networks are built and used, and who is doing the building and using.

Thus, CI scholars have asked fundamental questions about how to understand or define the "communities" of CWNs: are they geographically or virtually composed and defined? And how do they function to build relationships, or to foster collective visioning and collaboration toward achieving collective goals - that is, to build community?

In his editorial in the Journal of Community Informatics special issue "Wireless Networking for Communities, Citizens and the Public Interest," Michael Gurstein (2008) asks "whether there is a wireless community informatics beyond that of facilitating access. Perhaps, the broader concept of a wireless CI is nothing more than a peculiar artifact arising from the circumstance that many of the early wireless innovators, while having somewhat parallel backgrounds to the early community networking innovators (progressive politics, university education, technical proficiency, youthful) often referred to themselves as 'community networkers'."

This paper will evaluate claims about social and community benefits emerging from CWNs by synthesizing the findings of three empirical research projects looking in-depth at the impact of particular networks. Thought these studies come from different disciplines, each examines two fundamental questions: what, if any, social utility do these networks have beyond or by virtue of facilitating Internet access; and what is the "community" of the CWN(s) examined. These studies have not previously been compared, since they are drawn from across communities of practice; yet comparison of their findings provides a framework to evaluate claims of broad social benefit emerging from CWNs as well as to understand, leverage, and work from actual documented outcomes.

Practitioners seeking to design and implement networking projects, as well as policymakers in the US, Europe, and developing world contexts searching for local connectivity solutions, can build realistic expectations by exploring the CWN efforts discussed here. Additionally, this paper will introduce a collaborative research methodology used by the CONFINE Consortium that could be adapted to introduce a set of intentional, collective goals - beyond broadband access - into the practice and process of community networking. As we move into another era of investment in local wireless networking as a potential broadband solution, it is important to understand the successes and failures of past efforts, and to create a framework for evaluating social impact and finding the most suitable and appropriate use cases for this technology.

CWN advocates have projected benefits - from digital inclusion to economic development - that should emerge by virtue of creating free public Wi-Fi access (Schuler, 1994,1996; Antoniadis et al., 2008). Yet in the 2008 JoCI special issue on wireless networking, Tapia and Ortiz's article demonstrates how, in practice, many municipal-community wireless projects did not live up the rhetoric that was employed to "sell" them to local governments and stakeholders. Specifically, Tapia and Ortiz illustrate how the two cases they focus on (Portland and Tempe) foundered on the shoals of sustainability (in particular, lack of sufficient funds for maintenance and upgrades) and adoption (lack of interest or buy-in by intended beneficiaries). In these cases - as in Philadelphia, San Francisco, Santa

Clara, East Palo Alto, and other localities where local actors tried to provide free public Wi-Fi in the mid-2000s - failures of sustainability and adoption ended the projects, in some cases with millions of (public) dollars and effort lost along the way.

Yet at the same time, many networks have thrived, mostly in Europe. These include some of the largest and longest-standing CWNs in the world: ii for example, guifi.net, at approximately 25,000 nodes and 15 years old, covering large swathes of Catalonia, Spain, and now and reaching into other parts of the world; the Athens Wireless Municipal Network, at 3000 nodes and 13 years old (AWMN, 2014), covering Athens and broad swathes of Greece; Freifunk Berlin, at 12 years old and approximately 400 nodes; and Wireless België, at 10 years old and approximately 700 nodes (data via Avonts, Braem, and Blondia, 2013).

None of these successful networks is municipal or city-led, though they tend to fall mostly within particular geographic areas and depend for expansion and maintenance on local offline populations, and in some cases are supported or leveraged by local municipalities. In fact, each of these networks depends on the efforts of a local volunteer workforce of technologically savvy individuals. All are also part of the CONFINE (Community Networks for the Future Internet) Consortium, a federated community network test bed and research effort supported by the European Union's Seventh Research Framework Programme to explore and demonstrate the social and technological potential of CWNs for practitioners, researchers, and policymakers.

Meanwhile, in the US, the Obama administration is currently working with the National Telecommunications and Information Agency (NTIA) and representatives from other federal agencies, along with municipal stakeholders, to pilot and support muni-community broadband solutions. Along with municipal governments and NGOs, the US Department of Housing and Urban Development (HUD) and other agencies are investing in local wireless along with other solutions, as privately supplied communications infrastructure has not been able to close the broadband gap: 28% of Americans still do not have broadband access at home (NTIA, 2014). As we move into another era of potential investment in local wireless networking as a potential solution to this problem, it is important to understand the successes and failures of past efforts, and to create a framework for evaluating their social impact across different fields of practice.

2.2 Community Wireless Network of the Sarantaporo.gr

Sarantaporo.gr is a non-profit civil society that was established in September 2013 but as a team volunteer since 2010 in the wider region of Sarantaporo Elassona, building a community wireless network connecting 15 villages in the area, while providing free access to internet for residents, residents and visitors to the area.

The character of the community network of Sarantaporo.gr is public, open and social in the sense that our network:

- implemented and developed by Sarantaporo.gr but with the contribution of local associations and active residents of the villages in the area,
- is free to use by any resident, operator or visitor of those areas if its use is in accordance with its operating rules and does not hinder the sharing of available network resources,
- access to the Community network is not the subject of speculative activity.

The community network Sarantaporo.gr is a “common”, a resource open to all and manageable by all those who participate in it. It is an important example of wireless / wired Community electronic communications networks, which are now being developed in many parts of the world. It is a promising alternative to rapidly bridge the digital divide in the most deserted regions of the country, while at the same time promoting co-operation and equal participation in the digital age and the wealth of knowledge and opportunities it offers. Our ultimate goal, with regard to Community networks, is to create a model for the development of such infrastructures, which can be reproduced in other areas of 'digital isolation'.

For the future we are envisioning and working towards the creation of a modern community network that combines wireless technologies with fiber optic technologies that will connect all the villages of the Municipality of Elassona and will provide high-speed internet connection and other Community networks in Greece and abroad. For this purpose, we work collaboratively with local societies and use technological knowledge and scientific methods in collaboration with various institutions: cultural associations, educational institutions, NGOs, research centers, local authorities etc.

The name of the team, Sarantaporo.gr, arose from the website we started to build for our village, which would be a reference point for our childhood years from the village, as well as the information and cooperation tool for the organization of cultural and voluntary activities.

Along the way, however, we realized that the villagers could not visit this site because they did not have access to the internet! Seeking a solution to this problem, we were soon able to implement a wireless community network in our village, which provides free access to all residents and visitors, in cooperation with the LTE and Open Wi-Fi. The news was quickly traced to the surrounding villages and our team responded with pleasure and offer to the request of clubs and active residents of 14 neighboring villages for the construction of corresponding networks in their villages, some of which even today have no alternative access to internet via the Sarantaporo.gr community wireless network.

2.3 Single Board Computers

These devices are typically used with backplanes, which allow a single board computer to be easily connected to systems and in a very compact arrangement.

While single board computers are typically less powerful than standard, tower computers, they are more than adequate for the purposes they are used toward in industry. Some of these devices are also excellent experimentation and teaching tools, owing to their low cost, simplicity and the fact that many of them can easily be connected to other hardware.

A single board computer is differentiated from a desktop or personal computer not by what it is capable of, but by the way it is designed. A single board computer has all of the elements of a complete computer contained within one single circuit board. Conversely, desktop computers have motherboards with slots for the basic components of the computer. On a desktop computer, the processor, memory, storage, input and output devices and other basic components attach via sockets on the motherboard. On a single board design, everything is self-contained.

Single board computers are also expanded in their capabilities differently than desktop computers. On a desktop computer, adding capability to the motherboard is typically accomplished by plugging a peripheral device into a slot on the motherboard. A high-end video card, for instance, will be plugged into a specialized slot on the motherboard. On a single board computer, this and the other components required on the computer are simply a part of the circuit board and the device is ready to go as is.

Single board computers come in a huge range of different capacities. Because some of them are used to control very simple processes, some single board computers are very slow and limited compared to the average desktop computer.

Single board computers are frequently employed in embedded applications. An embedded computer cannot be expanded upon and contains only the input and output capabilities it needs for the task for which it is designed. For example, a vending machine might have an embedded single board computer in it to control the functions of the vending machine, but there would be no provision to add more hardware to the computer to expand its capabilities.

In many cases, single board computers are plugged into a backplane. The backplane allows for input and output devices to be attached to the computer. Single board computers are frequently used in rack systems, which allows for reliable and fast integration into a system.

Chapter 3

Implementation

In this thesis has been created a compact device named Toolkit, connecting members of a community using a wireless technology. In this chapter we will describe the operation of the Toolkit, as well as the cases where a device like this can become vital to a community. We will also mention the hardware and software required for building this device and the available features.

3.1 The Toolkit overview

The Toolkit it is a combination of low-cost hardware, open source software and wireless technology, for building local network, known as community wireless networks. The local community can benefit from Internet access or from local services and applications where the Toolkit is hosted.

It is not intended for technology experts, but for communities and individuals such as researchers, activists, artists, social scientists and in general most citizens who are willing to take control of their digital world. It offers a user-friendly interface that allows the creation and customization of a local wireless network covering the needs of the local proximity. The coverage of the physical space could be extended as the Toolkit can operated either one single connecting the users in the low physical proximity or more Toolkits forming a wireless mesh network, covering range to a whole neighborhood

There are three main users' categories where interacted with the device Toolkit and involved in its design, deployment, and usage:

- The owner: an individual (researcher, activist), community (a nonprofit, a neighborhood), or local authority.
- The administrator: The person responsible for choosing the various configuration and customization options. The administrator could be the same as the owner of the Toolkit.
- The users: individuals or communities who are expected to interact through the Toolkit and consume its outputs.

3.2 Uses cases

In this section will describe the possible daily use cases of the toolkit in order to make it more comprehensible. Firstly, as we mention before the Toolkit was implemented to covering the needs of a local communities. As a community we could define a group of people living in the same place or having a characteristic in common. So, there are a lot of scenarios in our lives where we form a community. The following are some of these:

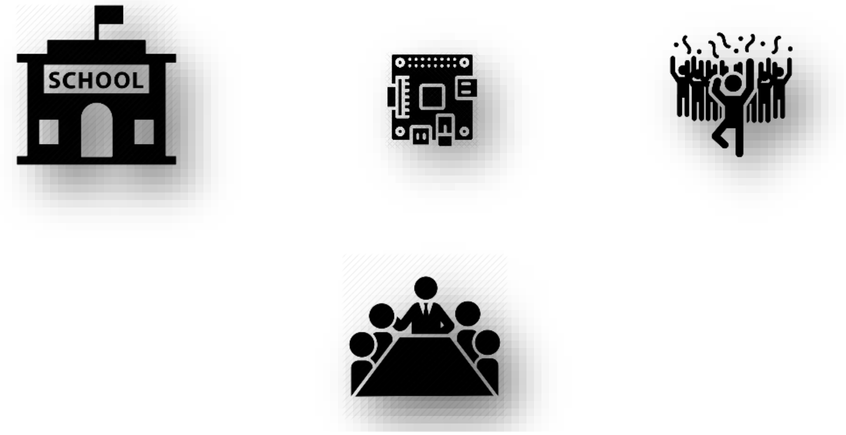


Figure 3 8 uses cases

- Conference/meetings: One of the most common uses of Toolkit is in meetings or conferences. Toolkit users are able to synchronize about the agenda, presentations and more using the appropriate apps and on the other hand the organizers can collect useful data about the event, forming a general aspect.

- Education: A class teacher could use the Toolkit, forming a wireless space where pupils will be informed of the courses, will be able to take small examinations or to pass decisions on future actions.
- Parties: Instead of connecting to a mainstream social media, the Toolkit is ideal for such deployments, where people in physical proximity can upload photos, texts and comments on others' posts.

3.3 Hardware

A comprehensive investigation on the hardware used for this thesis was conducted in order to find the appropriate, covering the needs of the Toolkit. First, we needed a platform that the Toolkit will be based, and this platform should satisfy four vital conditions: low consumption, low purchase cost, open source software support and small dimension. For all above the single board computer Raspberry Pi was chosen offers also a various of its features, such as number of USB ports, extensibility through many GPIO pins and onboard Wi-Fi module.

Moreover, to improve the wireless network provided by the Toolkit for both coverage and more functionality, the integration of USB dongle antennas and routers was indispensable. Also, integration of sensors was vital, collecting atmospheric measurements for our local physical proximity.

3.3.1 Raspberry Pi

The Raspberry Pi3 is the latest version of the credit card sized computer from the Raspberry Pi Foundation. The Raspberry Pi was released by Premier Farnell, the company behind element14, in 2012. The Raspberry Pi 3 is about the 50%

faster than its predecessor the Pi 2. It includes built in WIFI and Bluetooth Low Energy connectivity, making it truly an IOT-ready device.

The Raspberry Pi 3 features a 1.2 GHz quad-core 64-bit Arm Cortex A53 processor, Chip antenna, 4 USB ports, an Ethernet Port, a GPIO, HDMI, 3.5mm Audio Output, WIFI chip, 1GB LPDDR2 for RAM Memory, and a MicroSD slot. The MicroSD card contains the Pi3's operating system and it can also be used for file storage

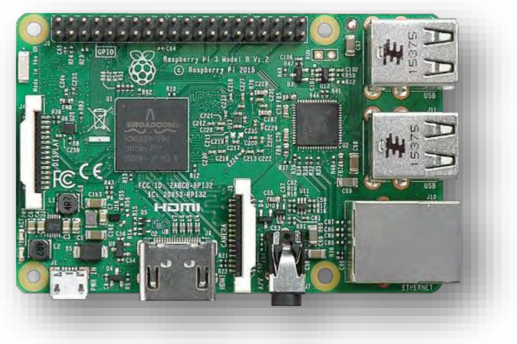


Figure 3 9 Raspberry Pi

3.3.2 USB Dongle Antenna

Two different Wi-Fi protocols of USB dongle antenna have been implemented on the Raspberry Pi. Namely, the 802.11n and the 802.11ac. The first one Wi-Fi protocol is a wireless-networking standard that uses multiple antennas to increase data rates. Sometimes referred to as MIMO, which stands for "multiple input and multiple output". Its purpose is to improve network throughput over the two previous standards—802.11a and 802.11g. (Figure 2).

The last Wi-Fi Protocol provide high-throughput wireless local area networks (WLANs) on the 5 GHz band. The specification has multi-station throughput of at least 1 gigabit per second and single-link throughput of at least 500 megabits per second (500 Mbit/s). This is accomplished by extending the air-interface concepts embraced by 802.11n: wider RF bandwidth (up to 160 MHz), more MIMO spatial streams (up to eight), downlink multi-user MIMO (up to four clients), and high-density modulation (up to 256-QAM). (Figure 3)

Figure 3 10 USB Dongles antennas 802.11ac protocol

Figure 3 11 USB Dongles Antennas 802.11n protocol

3.3.3 Router

One of the most important aspects of the Toolkit is the Wi-Fi Access Point that it provides, trying to empower the characteristic of it such as the number of users that can be connected, its connectivity range and more networking aspects we intergraded the OpenWrt Router TP-LINK TL-MR3020. To this end, an automated functionality has implemented in the Low-level Command Line Interface which migrate the Access Point to the Router and provide improved networking characteristics to the users. You can find more details about the Low-Level Command Line Interface in section 3.3.4.

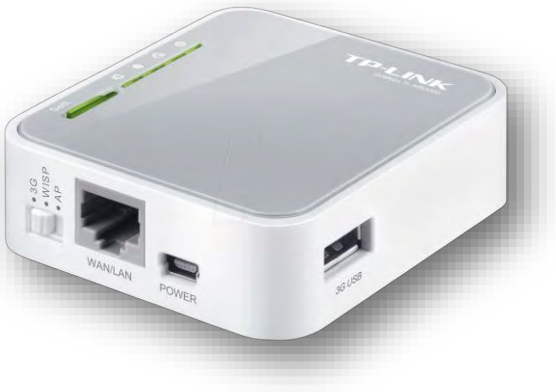


Figure 3 12 OpenWrt Router Tp-Link TL-MR3020

3.3.4 Sensors

Two different types of sensor modules are now integrated on the Raspberry Pi. Namely, the Sense HAT [6] board for Raspberry Pi and the SHT11 module [7] from Sensirion, both shown in 3 6. The later one is a simple relative temperature and humidity sensor that provides a digital output and is individually calibrated in a precision humidity chamber.

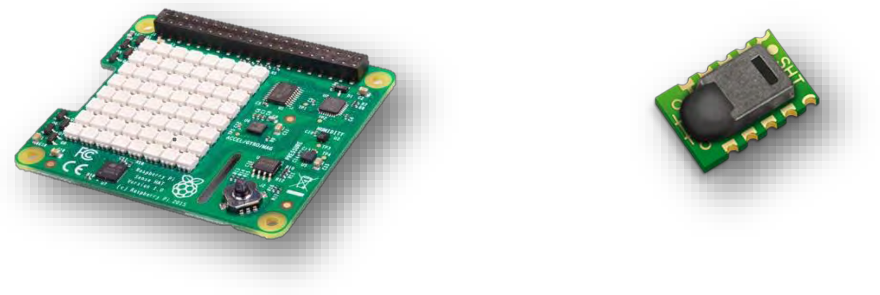


Figure 3 13 Sensors

The Sense HAT features an 8X8 RGB LED matrix, a mini joystick and the following sensors:

- Gyroscope
- Accelerometer
- Magnetometer

- Temperature
- Humidity
- Barometric pressure

3.4 Software

3.4.1 Operating system

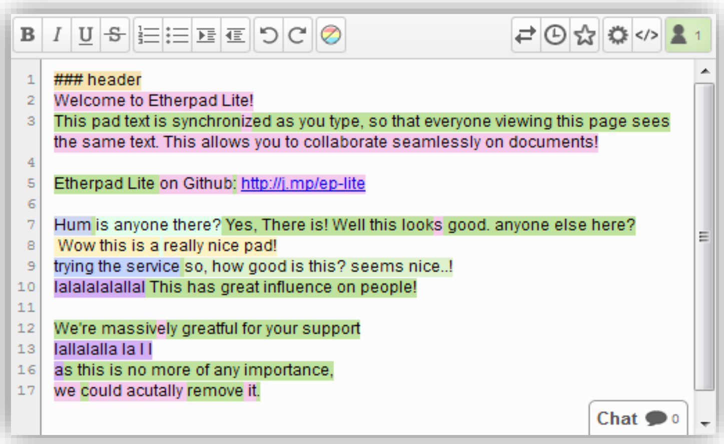
The implementation of the Toolkit was necessary to rely on an open source operating system as we had to adapt its function according to our environment. That's why we worked at Raspbian Jessie OS. The Raspbian Jessie is a Debian-based computer operating system for Raspberry Pi. Since 2015 it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the family of Raspberry Pi single-board computers.

3.4.2 Applications

In this section we represent the applications where have been installed on the Toolkit device. The basic requirement of these applications was to be an open source as we would like to provide free to the citizens and programming these in that way to collect useful data. The applications were selected towards the goal of enriching the Toolkit with the service such as file sharing, collaborative editing or survey management.

Etherpad

Etherpad [8] is an open source, web-based collaborative real-time editor, allowing authors to simultaneously edit a text document, and see all of the participants' edits in real-time, with the ability to display each author's text in their own color. There is also a chat box in the sidebar to allow meta communication.



```
### header
Welcome to Etherpad Lite!
This pad text is synchronized as you type, so that everyone viewing this page sees
the same text. This allows you to collaborate seamlessly on documents!

Etherpad Lite on Github: http://j.mp/ep-lite

Hum is anyone there? Yes, There is! Well this looks good. anyone else here?
 Wow this is a really nice pad!
trying the service so, how good is this? seems nice..!
lalalalalallal This has great influence on people!

We're massively greatful for your support
lallalalla la l l
as this is no more of any importance,
we could acutally remove it.
```

Figure 3 14 Etherpad

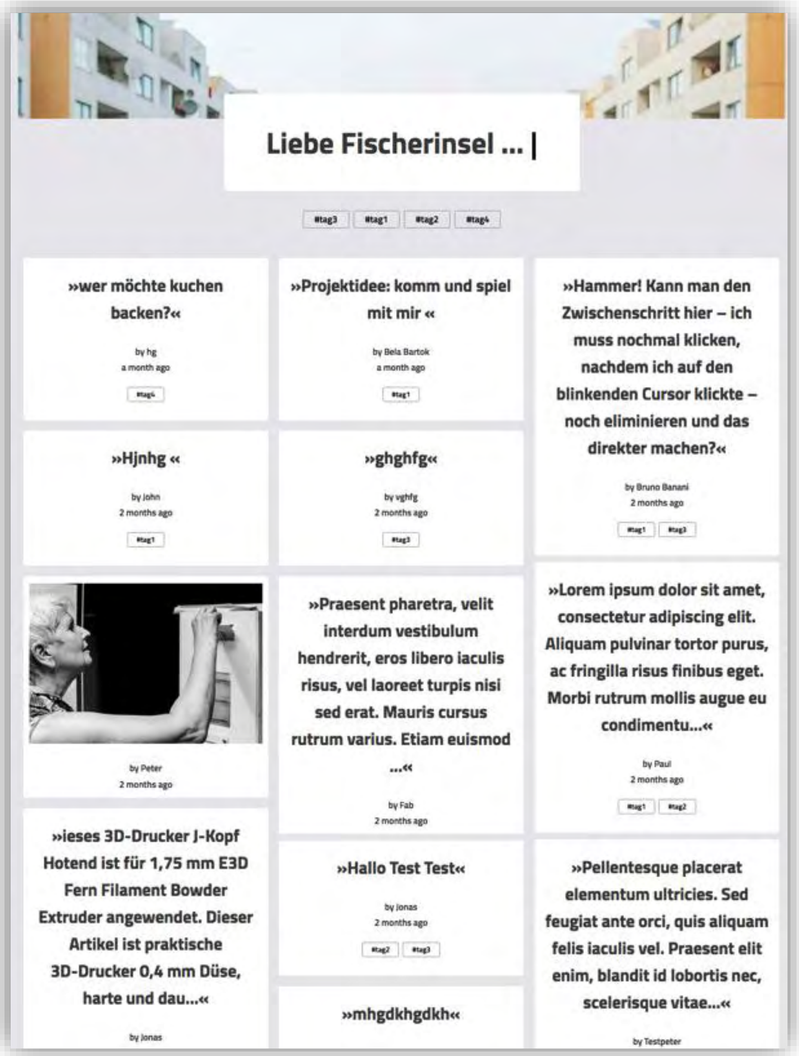
Nextcloud

Nextcloud is a suite of client-server software for creating and using file hosting services[9]. It is functionally similar to Dropbox, although Nextcloud is free and open-source, allowing anyone to install and operate it on a private server. In contrast to proprietary services like Dropbox, the open architecture allows adding functionality to the server in the form of applications and enables users to have full control of their data.

Figure 3 15 Nextcloud

Guestbook

The Guestbook[10] is a web-based application that allows users to post short texts and images without needing to sign-up. The posts are displayed in a composite view ordered by how recently they were uploaded. This application is part of the Toolkit and is being developed as a generic application for sharing ideas, photos, and more, related to the specific place where a Toolkit deployed.



Liebe Fischerinsel ... |

»wer möchte kuchen backen?«

by hg
a month ago

»Hjnhg «

by John
2 months ago

by Peter
2 months ago

»ieses 3D-Drucker J-Kopf Hotend ist für 1,75 mm E3D Fern Filament Bowder Extruder angewendet. Dieser Artikel ist praktische 3D-Drucker 0,4 mm Düse, harte und dau...«

by Jonas

»Projektidee: komm und spiel mit mir «

by Bela Bartok
a month ago

»ghghfg«

by vghfg
2 months ago

»Praesent pharetra, velit interdum vestibulum hendrerit, eros libero iaculis risus, vel laoreet turpis nisi sed erat. Mauris cursus rutrum varius. Etiam euismod ...«

by Fab
2 months ago

»Hallo Test Test«

by Jonas
2 months ago

»mhgdkhgdkh«

»Hammer! Kann man den Zwischenschritt hier – ich muss nochmal klicken, nachdem ich auf den blinkenden Cursor klickte – noch eliminieren und das direkter machen?«

by Bruno Banani
2 months ago

»Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam pulvinar tortor purus, ac fringilla risus finibus eget. Morbi rutrum mollis augue eu condimentu...«

by Paul
2 months ago

»Pellentesque placerat elementum ultricies. Sed feugiat ante orci, quis aliquam felis iaculis vel. Praesent elit enim, blandit id lobortis nec, scelerisque vitae...«

by Testpeter

Figure 3 16 Guestbook

WordPress

WordPress is an open-source and free Web publishing application, content management system (CMS) and blogging tool built by a community of developers and contributors. WordPress allows users to build dynamic websites and blogs that may be updated, customized and managed from its back-end CMS and integrated application and components.

Figure 3 17 WordPress

FramaDate

Doodle is an online scheduling tool that can be used quickly and easily to find a date and time to meet with multiple people. First you suggest dates and times for your event participants to choose from, then Doodle creates a polling calendar that can be sent to participants for feedback. As each participant selects the dates and times from the polling calendar that he or she is free, Doodle aggregates the responses and tells you which option works best for everyone.

Figure 3 18 Framadate

3.5 Low-Level Command Line Interface

The Low-level Command Line Interface is a collection of Linux-shell scripts which have been created to serve the following needs:

- Middleware: it operates as middleware between the hardware and a higher-level application interface.
- Coordinator: it acts to interconnect all the hardware where we presented above
- Advance User Interface: an advance user could use this interface for debugging if something gone wrong with the Toolkit. He could also understand the function of the tools at the code level and then add new features and functions where he satisfies his situation.

The Linux-shell scripts are divided to three categories according to each operation. Namely, System, Networking and Collection. The System are consisted of these scripts where interact with the system of the Toolkit, they give us information about the current state of the Toolkit. The Networking scripts starts all the necessary services that the local wireless network needs and finally the Collection scripts gather useful data from applications, sensors and statistics of the system to help the Data Collection Framework.

Following is depicted the collection of the Linux-shell scripts where they are form the Low-level Command Line Interface, it presented the functionality and the usage of each script separately.

3.5.1 The scripts

In this section we will describe the all available scripts of the Low-level Command Line Interface, providing for each script an operation description and one usage table. Moreover, for better comprehension of each script there is a collection of examples.

app.sh

app.sh	
-a, --action [start, stop, status] <application>	Controls the status of the installed applications

Table 3 3 Script Usage

Description:

The app.sh script enables the control of the status of the installed applications such as the Etherpad, the Guestbook, and the Interview-archive. You can start, stop or display the status of the above applications.

Examples:

- Start the application Etherpad
  sh app.sh -a start etherpad

- Stop the application Guestbook
  sh app.sh -a stop guestbook

- Display the status of the application Interview-archive
  sh app.sh -a status interview

appstat.sh

appstat.sh		
-n, --name	--store [enable, disable, flush]	Controls the status of the storage process
	-d, --domain	Sets the server domain to be used for storage (Default is localhost)
--status		Shows the status of storage process

Table 3 4 Script Usage

Description:

The mazi-appstat.sh script enables the collection of statistical data from the applications installed on the Raspberry Pi and the storage of these data in a local or remote database. In addition, you could flush these data from the database in case you do not need them. Now, you can collect data from the following applications, Guestbook, Etherpad and FramaDate.

Examples:

- Start collecting data for Etherpad and store them in a local database
  sh appstat.sh -n etherpad --store enable

- Start collecting data for Guestbook and store them in a remote database (demo.mazizone.eu)
  sh appstat.sh -n guestbook --store enable -d demo.mazizone.eu

- Stop collecting data for Framadate
  sh appstat.sh -n framadate --store disable

- Flush the statistical data of Etherpad in the local database
  sh appstat.sh -n etherpad --store flush

- Flush the statistical data of Etherpad in a remote database (demo.mazizone.eu)
  sh appstat.sh -n etherpad --store flush -d demo.mazizone.eu

- Display the status of the collection service
  sh appstat.sh --status

sense.sh

bash sense.sh [SenseName] [Options] [SensorOptions]	
[SenseName]	
-n, --name (sensehat, sht11)	The name of the sensor
[Options]	
-s, --store	Stores the measurements in a database
-d, --duration	Duration in seconds to take measurements
-i, --interval	Seconds between periodic measurements
-a, --available	Displays the status of the available sensors
-D, --domain	Sets a remote server domain (default is localhost)
--status	Displays the status of the storage process
[SensorOptions]	
-t, --temperature	Get the temperature
-h, --humidity	Get the humidity
-p, --pressure	Get the current pressure in Millibars
-m, --magnetometer	Get the direction of North
-g, --gyroscope	Get a dictionary object indexed by the strings x, y, z. The values are floats representing the angle of the axis in degrees
-ac, --accelerometer	Get a dictionary object indexed by the strings x, y, z. The values are floats representing the acceleration intensity of the axis in Gs

Table 3 5 Script Usage

Description:

The mazi-sense.sh script has been created to manage various sensors where are connected to the Raspberry Pi. This script can detect the connected sensors and consequently collect measurements periodically with a specific duration and interval between measurements. In addition, it can store these measurements in a local or remote database and check the status of the storage procedure, as well.

Examples:

- Display which sensors are connected to the Raspberry Pi
  bash sense.sh -a

- Display the atmospheric temperature and humidity from Sensehat sensor
  bash sense.sh -n sensehat -t -h

- Display the atmospheric pressure from Sensehat sensor and store it in a local database
  bash sense.sh -n sensehat -p –store

- Display the Accelerometer x y z raw data in Gs from Sensehat sensor and store them in a remote database (demo.mazizone.eu)
  bash sense.sh -n sensehat -ac --store -D demo.mazizone.eu

- Display the direction of North in degrees from Sensehat sensor for 60 secs
  bash sense.sh -n sensehat -m -d 60

- Display the atmospheric temperature from Sht11 sensor every 5 sec for 60 secs
  bash sense.sh -n sht11 -t -i 5 -d 60

- Display the status of the storage procedure
  bash sense.sh –status

domain.sh

domain.sh	
-d, --domain	Sets a new network domain for the portal
-s, --splash	Sets a new splash page

Table 3 6 Script Usage

Description:

The mazi-domain.sh script enables the modification of the MAZI Portal’s domain and the change of the splash page.

Examples:

- Change the domain of the portal page as “new.example.eu”
  sh domain.sh -d new.example.eu

- Set the Guestbook application as a splash page
  sh domain.sh -s guestbook

internet.sh

internet.sh	
-m [offline/dual/managed]	Sets the mode of the Wi-Fi Access Point

Table 3 7 Script Usage

Description:

The mazi-internet.sh script is able to modify the mode of your Wi-Fi Access Point – currently - between offline and dual as the managed mode has not been implemented yet. In the offline mode, clients of the Wi-Fi Access Point have not access to the Internet and are permanently redirected to the Portal splash page. In the dual mode, the Raspberry Pi provides Internet access through either the Ethernet cable or an external USB Wi-Fi adapter.

Examples:

- Set the mode to offline
  sh internet.sh -m offline

- Set the mode to dual
  sh internet.sh -m dual

antenna.sh

antenna.sh	
-a, --active	Shows if a USB Wi-Fi adapter exists
-s, --ssid	Sets the SSID of the Wi-Fi network
-p, --password	Sets the password of the Wi-Fi network
-l, --list	Displays a list of the available Wi-Fi networks in range
-h, --hidden	Connect to hidden Wi-Fi network
-d, --disconnect	Disconnect the USB adapter from Wi-Fi network

Table 3 8 Script Usage

Description:

The mazi-antenna.sh script has been created in order to manage an external USB adapter that is connected to the Raspberry Pi. This script is able to check if a USB adapter is connected to the Raspberry Pi. In addition, you can discover the available networks in range and connect to one of them. Finally, you can disconnect the USB adapter from the connected Wi-Fi network.

Examples:

- Check for existing USB adapter
  sh antenna.sh –a

- List the available Wi-Fi networks in range of the USB adapter
  sh antenna.sh –l

- Connect to the Wi-Fi “mazi-network” with password “mazipass”
  sh antenna.sh -s mazi-network -p mazipass

- Connect to the hidden Wi-Fi “mazi-hidden”
  sh antenna.sh -s mazi-hidden -h

- Disconnect the USB adapter from the Wi-Fi network
  sh antenna.sh -d

router.sh

router.sh	
-s, --status	Displays if the OpenWRT router exists
-a, --activate	Activates the OpenWRT router as the Wi-Fi AP of this MAZI Zone
-d, --deactivate	Disconnects the router and restores the initial settings of the Raspberry pi built-in Wi-Fi module

Table 3 9 Script Usage

Description:

The router.sh script is used for the management of the OpenWrt Router connected to this MAZI Zone. After connecting an OpenWrt Router this script is able to detect it and control the status of the connection, (activate/deactivate).

Examples:

- Display the status of the router (connected or not)
  sh router.sh –s

- Activate the Router with the current Wi-Fi settings
  sh router.sh –a

- Deactivate the Router and move the Access Point to the on-board Wi-Fi module of the Raspberry
  sh router.sh –d

wifi.sh

wifi.sh	
-s, --ssid	Sets the name of the Wi-Fi network
-c, --channel	Sets the Wi-Fi channel
-p, --password	Sets the Wi-Fi password
-w, --wpa [OFF/off]	Turns off wireless network security

Table 3 10 Script Usage

Description:

The mazi-wifi.sh script is responsible for creating the Wi-Fi Access Point on the Raspberry Pi. With this script, you can also modify the settings of your Wi-Fi Access Point.

Examples:

- Set the Wi-Fi SSID to “mazizone”
  sh wifi.sh -s mazizone

- Set the Wi-Fi channel to 6
  sh wifi.sh -c 6

- Set the Wi-Fi network password to mazipass
  sh wifi.sh -p mazipass

- Remove the password from the Wi-Fi network
  sh wifi.sh -w off

current.sh

current.sh	
-i, --interface	Shows the name of the active wireless interface
-c, --channel	Shows the Wi-Fi channel in use
-m, --mode	Shows the mode of the Wi-Fi network
-p, --password	Shows the password of the Wi-Fi network
-s, --ssid	Shows the name of the Wi-Fi network
-d, --domain	Shows the network domain of the MAZI Portal
-w, --wifi	Shows the device that broadcasts the Wi-Fi AP (pi or OpenWRT router)

Table 3 11 Script Usage

Description:

The mazi-current.sh script displays the settings of the Wi-Fi Access Point that has been created in this MAZI Zone. You can view information such as the name, the password and the channel of the Wi-Fi Access Point. You can also see the domain you are using for the portal page, as well as the active interface that broadcasts the Wi-Fi Access Point - in case you have plugged in an OpenWRT router. Finally, this script informs you about the mode of your Wi-Fi Access Point (offline, dual, managed).

Examples:

- Display the wireless interface of your Wi-Fi Access Point
  sh current.sh –i

- Display the name of your Wi-Fi network
  sh current.sh –s

- Display the password of your Wi-Fi Access Point
  sh current.sh –p

- Display the channel of your Wi-Fi Access Point
  sh current.sh –c

- Display the mode of your Wi-Fi Access Point (offline, dual, managed)
  sh current.sh –m

- Display the domain of the portal page
  sh current.sh –d

- Display the device that broadcast the Wi-Fi Access Point (Raspberry Pi or OpenWrt router)
  sh current.sh -w

stat.sh

stat.sh	
-t, --temp	Displays the CPU core temperature
-u, --users	Displays the number of connected users
-c, --cpu	Displays the CPU usage
-r, --ram	Displays the RAM usage
-s, --storage	Displays the card storage in use
--sd	Displays information about the SD card
-n, --network	Displays the download/upload speed
-d, --domain	Sets a remote server domain (default is localhost)
--store	Controls the status of the storage process
--status	Shows tha status of the storage process

Table 3 12 Script Usage

Description:

The mazi-stat.sh script enables the observation of system activity data of the Raspberry Pi such as the CPU temperature, the CPU usage, the RAM usage, the Storage usage, the Download/Upload speed and the number of users connected to the Wi-Fi network. You can also see information about the SD card such as capacity and whether or not the filesystem has been expanded. Another functionality is the storage of these data in a local or remote database. In addition, you have the ability to flush these data from the database in case you do not need them.

Examples:

- Display the CPU core temperature and the online users
  sh stat.sh -t -u

- Display the CPU usage and the RAM usage
  sh stat.sh -c -r

- Display the available storage and the Download/Upload speed
  sh stat.sh -s -n

- Display information about SD card
  ```
  sh stat.sh --sd
  ```

- Start collecting CPU core temperature and store this in a local database
  ```
  sh stat.sh -t --store enable
  ```

- Start collecting the RAM usage and store this in a remote database (demo.mazizone.eu)
  ```
  sh stat.sh -r --store enable -d demo.mazizone.eu
  ```

- Stop the storing procedure
  ```
  sh stat.sh --store disable
  ```

- Flush the statistical data in the local database
  ```
  sh stat.sh --store flush
  ```

- Flush the statistical data in a remote database
  ```
  sh stat.sh --store flush -d demo.mazizone.eu
  ```

- Display the status of the storage service
  ```
  sh stat.sh --status
  ```

3.6 Data Collection Framework

A Data Collection Framework was implemented on the Toolkit, with the purpose of collecting and storing data from sensors, applications and system of the Toolkit in a local or remote Data Collection Server. These data could give to the administrator a general information from the local wireless network, understanding the needs and forming an opinion about his local community.

More specifically, the administrator is able for starting this framework to collect the following data:

- System activity: CPU usage, RAM usage, Storage, connected users
- Sensor measurements: temperature, humidity
- Application statistics: application clicks, application users,

All these data collecting used the corresponding script of the Low-Level Command Line Interface and then using a REST API are stored in a database. This database could be locally or remotely according to the context.

3.6.1 Use cases

There are two possible scenarios where the Data Collection Framework could be operated. The first one is depicted on Figure 1.1, the Toolkit collecting data (e.g. system activity) and storing in the local SQL database, in this case the Toolkit is operated as a Data Collection Server.

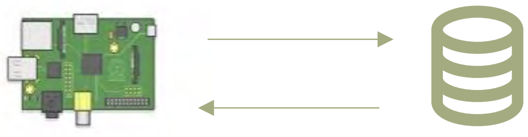


Figure 3 19 Toolkit and Data Collection Sever

Another scenario could be a local authority that would collect data shared voluntarily by multiple MAZI zone administrators building this way collective awareness at larger scale (than the coverage of a single MAZI node), through a truly bottom-up process. For example, a collection of Toolkits around the world collecting data and storing them to a public Data Collection Server.

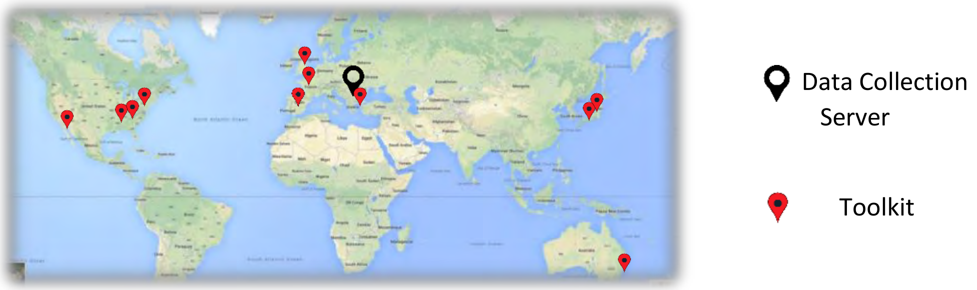


Figure 3 20 Public Data collection Server

3.6.2 Framework Operation

Both of two scenarios the operation of this framework is similar. Initially, the Toolkit communicates with the data collection server by sending a unique information (administrator name, deployment, tooltip, description) to find out if it is registered in the data collector database. The Data Collection Server responds to the Toolkit with a unique ID if it is existing in the database otherwise with the message “no such device”, then the Toolkit sends ones again its unique information to register in the database, obtaining its ID. After that, the Toolkit could start a repeated data collection process using the corresponding script of the Low-Level Command Line Interface, then this data encapsulated with the unique ID and sending to the Data Collection Server.

This Data Collection Framework was implemented on REST API protocol, benefit from following advantages. Initially, the REST protocol totally separates the user interface from the server and the data storage. This has some advantages when making developments. For example, it improves the portability of the interface to other types of platforms, it increases the scalability of the projects, and allows the different components of the developments to be evolved independently.

Moreover, the separation between client and server has one evident advantage, and that is that each development team can scale the product without too much problem. They can migrate to other servers or make all kinds of changes in the database, provided the data from each request is sent correctly. The separation makes it easier to have the front and the back on different servers, and this makes the apps more flexible to work with. To this end, it was created the following REST API calls.

- POST /monitoring/register
  Registers the Toolkit device and then responds with the Toolkit ID.

- GET /device/id
  Returns the Toolkit ID if it is registered, otherwise the message "no such device"

- POST /sensor/register
  Registers the sensor and then responds with the sensor ID.

- GET /sensor/id
  Returns the sensor ID if it is registered otherwise the message "no such device".

- POST /create/measurements
  Creates the appropriate table for the sensor measurements.

- POST /update/measurements
  Stores the sensor's measurements to the corresponding table.

- POST /create/application
  Create the appropriate table for the application statistic data.

- POST /update/application
  Store the application statistics data to the corresponding table.

- POST /create/system
  Create the appropriate table for system statistic data.

- POST /update/system
  Store the statistics data to the corresponding table.

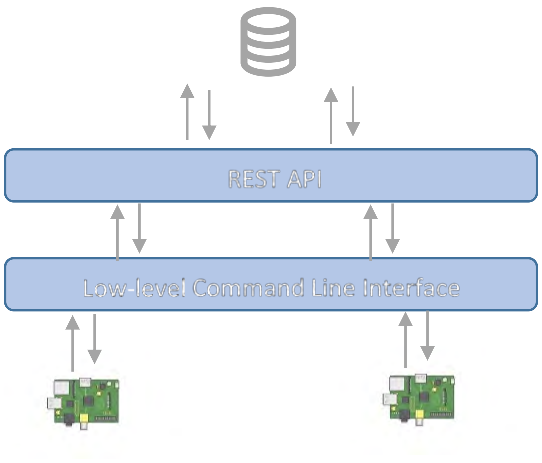


Figure 3 21 Data Collection Framework Layers

3.7 Mesh Network Functionality

The range of a Wi-Fi computer network depends primarily on the type of wireless Access Points used to build it. The Wi-Fi signal range of any given Access Point also varies significantly from device to devices. Factors that determine the range of one access point include:

- The specific 802.11 protocol it runs.
- The strength of its device transmitter.
- The nature of physical obstructions and/or radio interference in the surrounding area

A general rule of thumb in home networking says that Wi-Fi routers operating on the traditional 2.4 GHz band reach up to 150 feet (46 m) indoors and 300 feet (92 m) outdoors. Older 802.11a routers that ran on 5 GHz bands reached approximately one-third of these distances. Newer 802.11n and 802.11ac routers that operate on both 2.4 GHz and 5 GHz bands vary in the reach similarly.

Trying to eliminate the above restrictions of the Toolkit's Access Point, we integrated a mesh network functionality, in order to empower its coverage.

3.7.1 B.A.T.M.A.N-adv

The implementation of the mesh network functionality based on the B.A.T.M.A.N.-adv routing protocol. The Better Approach to Mobile Adhoc Networking (B.A.T.M.A.N.)[13] is a routing protocol for multi-hop mobile ad hoc networks which is under development by the German "Freifunk" community and intended to replace the link state routing protocol - OLSR.

Most other wireless routing protocol implementations (e.g. the batman daemon) operate on layer 3 which means they exchange routing information by sending UDP packets and bring their routing decision into effect by manipulating the kernel routing table. Batman-adv operates entirely on ISO/OSI Layer 2 - not only the routing information is transported using raw ethernet frames but also the data traffic is handled by batman-adv. It encapsulates and forwards all traffic until it reaches the destination, hence emulating a virtual network switch of all nodes participating. Therefore, all nodes appear to be link local and are unaware of the network's topology as well as unaffected by any network changes.

This design bears some interesting characteristics:

- Network-layer agnostic - you can run whatever you wish on top of batman-adv: IPv4, IPv6, DHCP, IPX ..
- Nodes can participate in a mesh without having an IP
- Easy integration of non-mesh (mobile) clients (no manual HNA fiddling required)
- Roaming of non-mesh clients
- Optimizing the data flow through the mesh (e.g. interface alternating, multicast, forward error correction, etc)
- Running protocols relying on broadcast/multicast over the mesh and non-mesh clients (Windows neighborhood, mDNS, streaming, etc)

3.7.2 Functionality Operation

There are two available modes of the mesh network functionality where the Toolkit can be operated, expending the Wi-Fi signal coverage.

- Gateway: It is the master node of the network. The Toolkit output will be visible to all users of this network.
- Relay: It is a relay node of the mesh network, forwards its users to the gateway node using the shortest path.

There is one limitation for the smooth operation of the Toolkit on the mesh network, is that of the wireless interfaces. The Toolkit device it should have two wireless interfaces, one for the Wi-Fi Access Point that it provides and one for the mesh network connectivity.

The operation creating a mesh network using a collection of Toolkits is the following. Initially, the Toolkit that we want to be visible to everyone in the mesh we create, acts as a gateway node, creating in that way an ad-hoc network with specific SSID(service set identifier). After that, each Toolkit that wants to be joined in the mesh network should be set as Relay node in order to connect to the gateway if it is visible, and to extend the wireless network. At the figure 4. Is depicted a topology with one Toolkit gateway node and four Toolkit relay nodes.

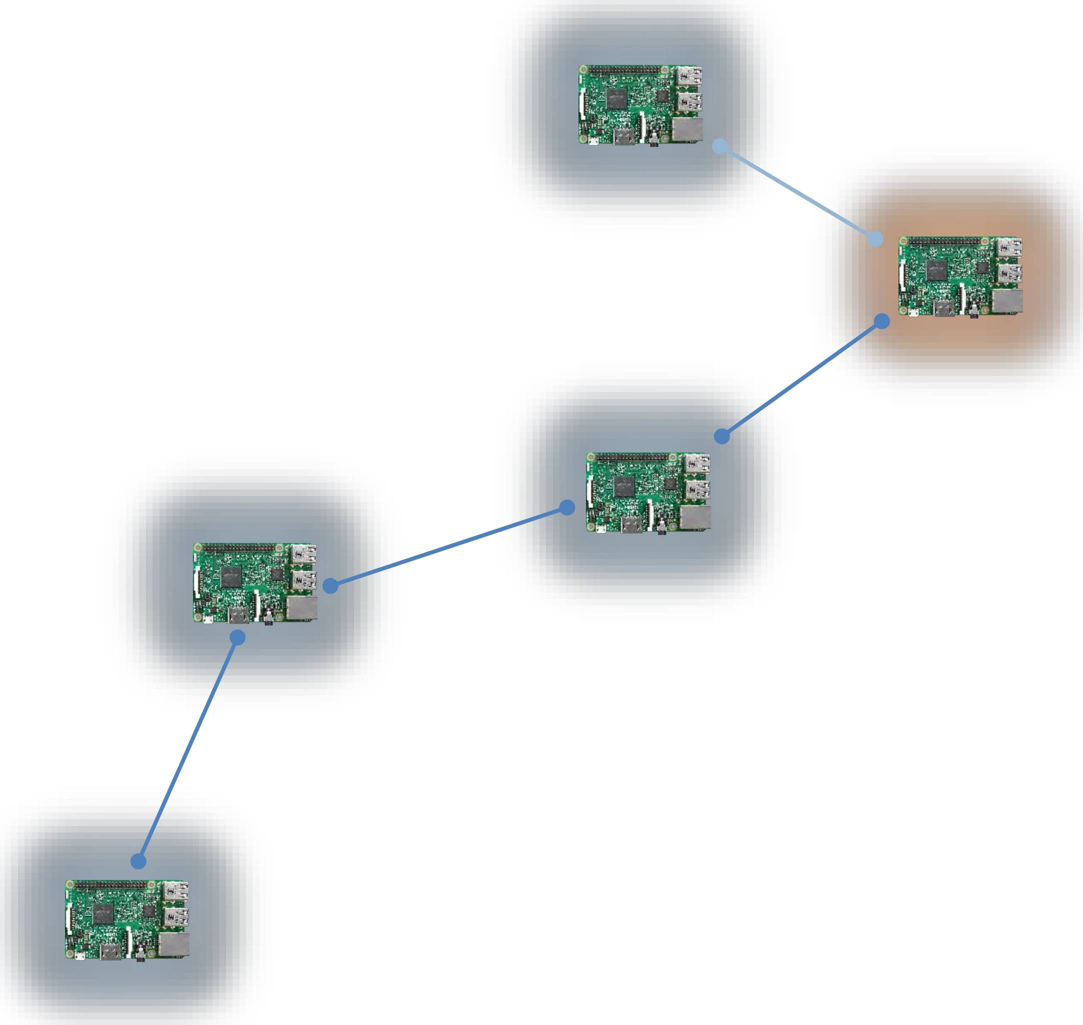


Figure 3 22 Mesh Network Topology

Chapter 4

Experiments

In this chapter we are going to describe the NITOS indoor testbed where we used in this thesis, processing the wireless experiments in a real environmental condition. In addition, we will depict the experiments process as well as the results.

4.1 NITOS Indoor RF Isolated Testbed

The NITOS RF Isolated Indoor Deployment [1]consists of 50 Icarus nodes that feature multiple wireless interfaces (Wi-Fi, WiMAX, LTE) and is deployed in an isolated environment at a University of Thessaly's campus building. Experimenters are able to run and evaluate power demanding processing algorithms and protocols in a large-scale testbed. It is also equipped with directional antennas and other advanced prototypes.

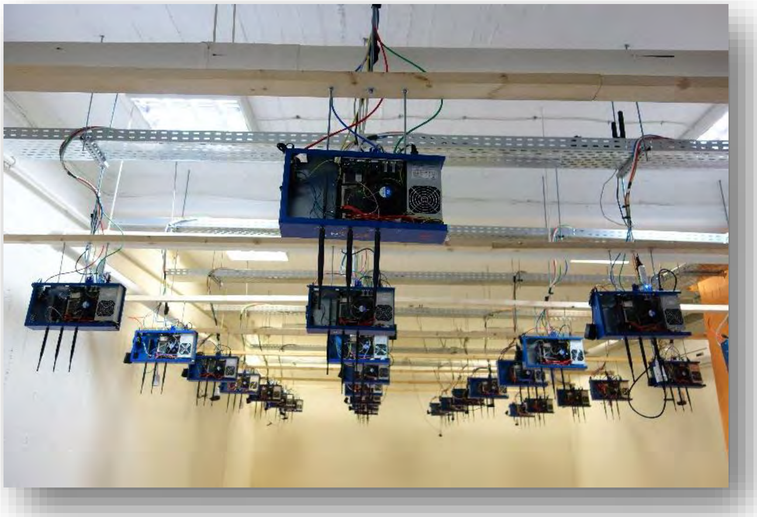


Figure 4 3 NiTOS Indoor RF Isolated Testbed

The Icarus nodes are placed in a symmetrical way around the isolated environment of NITOS indoor testbed forming a grid topology. The distance amongst the nodes is fixed to 1.2 meters and the height level is identical for all them as well. As a result, a uniform environment is created where all nodes are isobaric and of equal capabilities. You can find below a picture of the topology of the NITOS indoor testbed.

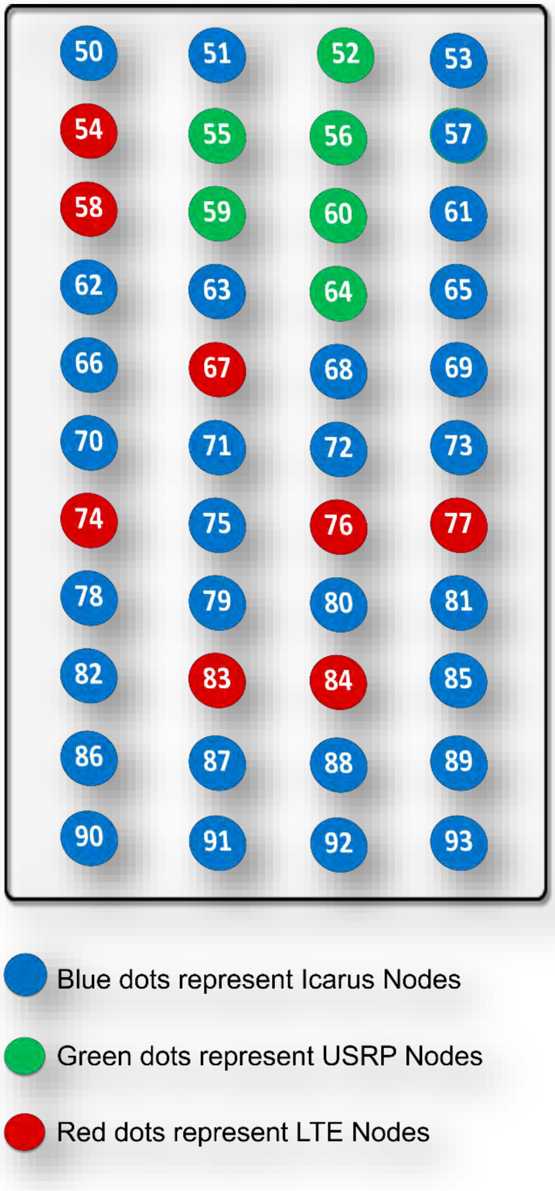


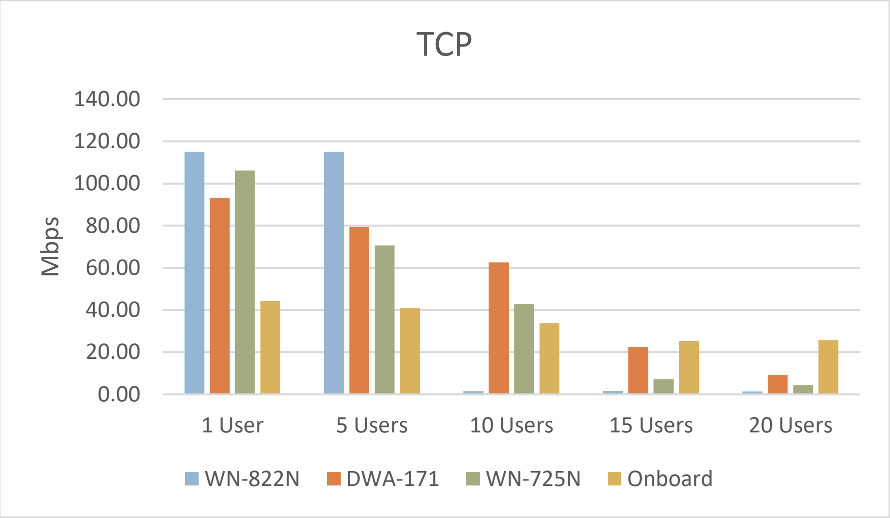
Figure 4 4 Testbed Topology

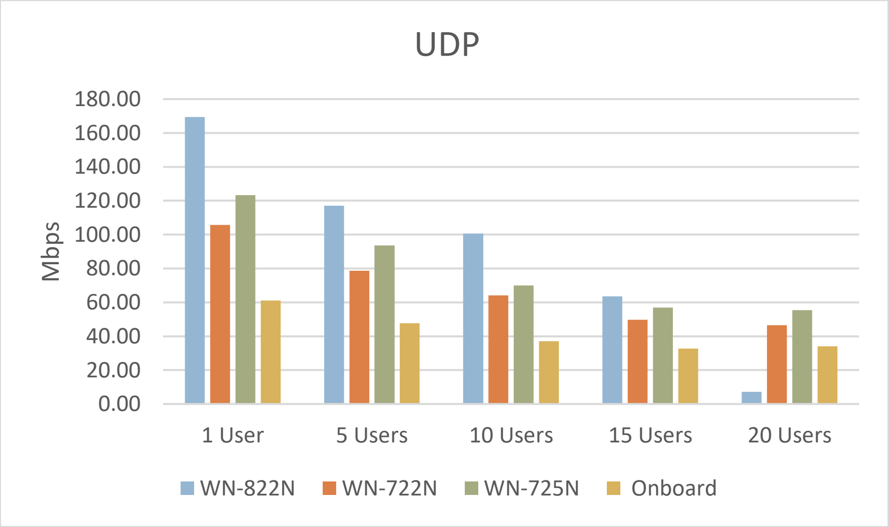
4.2 Experimental Procedure

Towards assisting the Toolkit users to choose between the available hardware modules before they assemble and deploy their MAZI toolkit, we conducted experiments in real-world settings in order to test and evaluate the network performance of diverse DIY setups and suggest the most appropriate solution for each use case. The experimentation was conducted in NITOS Future Internet Facility [1] where we integrated the hardware, namely a Raspberry Pi 3 model B together with multiple USB Wi-Fi adapters and conducted the research. The USB Wi-Fi adapters integrated are four of the 802.11n mode (TL-WN822N [14], TL-WN725N [15], TL-WN722N [16] and the onboard WI-Fi module) and three of the 802.11ac mode (EW-7811UTC [17], TL-Archer T4U [18], DWA-171[19]).

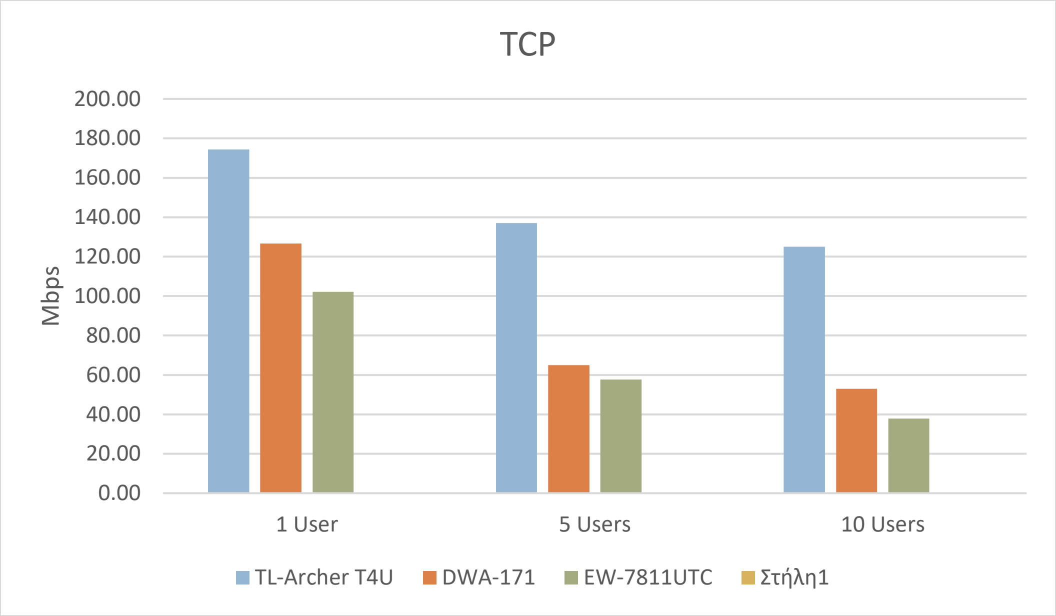
During the experiments we attached the USB Wi-Fi adapters to the USB port of the Raspberry, consequently connected a number of NITOS nodes to them and started sending TCP and UDP traffic in order to stress each one and discover its network limits in terms of throughput and number of concurrently connected users. We started with just one client connected, in order to check the capacity of the link and continued adding more Wi-Fi clients. We repeated the experiments for our seven different network modules and depicted from Fig to Fig and can be used as reference point for candidate Toolkit users in order to estimate their expected performance according to their network equipment. It should mention that the throughput show is achieved during concurrent traffic in each category of experiments (5 nodes, 10 nodes etc.) which is not the case in an ordinary Toolkit, where more than 30 users (on each Toolkit) will be able to connect and browse the local services.

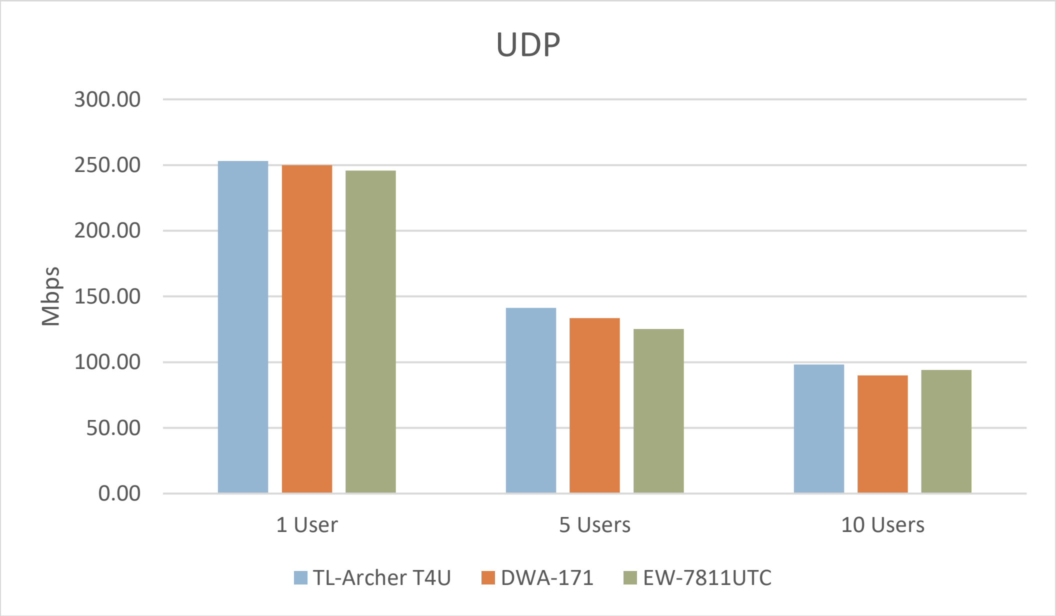
At the figure is depicted the results for experiments using the 802.11n protocol, for TCP and UDP respectively.





At the figure is depicted the results for experiments using the 802.11ac protocol, for TCP and UDP respectively.





Chapter 5

Conclusion

The toolkit has the potential to become a game changer in the empowerment of people to build location-based services hosted on infrastructures that are locally owned and managed. As WordPress enabled practically anyone with basic digital skills to build their own web site, the Toolkit will make it similarly easy to build local wireless networks hosting a wide variety of free software applications. For this, experts with very diverse backgrounds need to collaborate and better understand each other's domains. This paper introduced the Toolkit from an engineering perspective, which will hopefully inspire and empower researchers and activists to develop their own local experiments.

Bibliography

[1] NITlab. 2008-2018. NITOS Future Internet Facility.
http://nitlab.inf.uth.gr/NITlab/nitos

[2] Sarantaporo.gr.
http://www.sarantaporo.gr/node/382

[3] Single Board Computers
https://en.wikipedia.org/wiki/Single-board_computer

[4] Raspberry Pi Foundation. 2018. Raspberry Pi.
https://www.raspberrypi.org/.

[5] TP-Link. 2018. TL-MR3020
https://www.tp-link.com/il/products/details/cat-14_TL-MR3020.html

[6] Raspberry Pi Foundation 2018. Sensehat.
https://www.raspberrypi.org/products/sense-hat/

[7] Sensirion 2018. SHT11
https://www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-humidity-sensors-for-accurate-measurements/

[8] Etherpad 2018. Open-source collaborative editor.
http://etherpad.org/

[9] Nextcloud 2018. Open-source file sharing application.
https://nextcloud.com/

[10] MAZI Guestbook. 2016-2018. Open-source digital guestbook application.
https://github.com/mazi-project/guestbook

[11] WordPress. 2018. Content Management System.
https://wordpress.org/

[12] FramaSoft 2018. Open-source poll creation application
https://framadate.org/

[13] Open-Mesh 2018. B.A.T.M.A.N routing protocol
https://www.open-mesh.org/projects/open-mesh/wiki

[14] TP-Link. 2018. TL-WN822N.
https://www.tp-link.com/us/products/details/cat-5520_TL-WN822N.html

[15] TP-Link 2018. TL-WN725N.
https://www.tp-link.com/us/products/details/cat-5520_TL-WN725N.html

[16] TP-Link.2018. TL-WN722N.
https://www.tp-link.com/us/products/details/cat-5520_TL-WN722N.html

[17] Edimax 2018. EW-7811UTC
https://www.edimax.com/edimax/merchandise/merchandise_detail/data/edimax/au/wireless_adapters_ac600_dual-band/ew-7811utc

[18] TP-Link 2018. TL-Archer-T4U.
https://www.tp-link.com/il/products/details/cat-11_Archer-T4U.html

[19] D-Link 2018. DWA-171.
https://eu.dlink.com/uk/en/products/dwa-171-wireless-ac-dual-band-usb-adapter