A mobile streaming solution for IEEE 802.11 infrastructure and mesh networks

Fernando Gros González

upf. **Universitat Pompeu Fabra** *Barcelona*

To my family, specially my parents and sisters for supporting me in any situation and making possible this degree and this thesis.

Acknowledgments

I would like to thank to my thesis supervisors, Jaume Barceló and Efraín Foglia for their guidance and persistent help. In addition, a special thank you for the QMP team, specially Pau Escrich for his help on the firmware study.

Also, to my department partners, Nacho Justel, Jorge Beltrán and Alejandro Andreu for the collaboration and the knowledge exchange we have had during the thesis.

At last, it is important to say that this work was partially supported by the European Commission through the project Commons for Europe (CIP-ICT-PSP-2011-5-297191)

Abstract

The Mobile Node is a platform that is being developed to provide temporary wireless network coverage to events. The Mobile Node uses IEEE 802.11 technology combined with the "quick mesh project" (QMP) firmware and advanced mesh routing protocols (BMX, OLSR). In this thesis we have documented how does this platform works and how can be used to improve existing fixed mesh networks. Furthermore, we have developed an audio and video streaming application for android devices. This application can be used in combination with the Mobile Node technology and also on its own by connecting to any IEEE 802.11 network. When we combine this platform with the open mesh networks already deployed, we are providing the users with new solutions to continue improving their networks and to run applications over them.

Resum

El Node Mòbil es una plataforma que està essent desenvolupada per a proporcionar cobertura de xarxa sense fils en esdeveniments. El Node Mòbil utilitza la tecnologia del IEEE 802.11 combinada amb el firmware "Quick Mesh Project" (QMP) i protocols d'enrutament avançats (BMX, OLSR). En aquest treball hem documentat com funciona aquesta plataforma i com es pot utilitzar per a millorar les xarxes mesh fixes ja existents. A més, hem desenvolupat una aplicació de streaming de àudio y vídeo per a dispositius Android. Aquesta aplicació es pot utilitzar en combinació amb la tecnologia de Node Mòbil o en qualsevol xarxa IEEE 802.11. Quan combinem aquesta plataforma amb xarxes mesh obertes que ja estan desplegades, estem oferint als usuaris noves solucions per a continuar millorant les seves xarxes i per a executar aplicacions sobre elles.

Contents

List of Figures

List of Tables

List of Abbreviations

AP Access Point

BMX6 BatMan-eXperimental version 6

BSS Basic Service Set

DRP Dynamic Routing Protocol

GSF Gracia Sense fils

IID Internal IDentifier

LAN Local Area Network

MANET Mobile Ad-Hoc Network

NAT Network Address Translation

NCD Network Characterization Daemon

OLSR6 Optimized Linked State Routing version 6

PoE Power over Ethernet

QMP Quick Mesh Project

RIPE Réseaux IP Européens

RTSP Real Time Streaming Protocol

ULA Unique Local Address

WAN Wide Area Network

WMN Wireless Mesh Networks

Chapter 1

INTRODUCTION

The mobile node is a portable and auto-configurable transmission unit with wireless technology that offers mobility in the urban space. This node is designed to contribute to the digital mesh through existing networks. Furthermore, it provides connectivity with a wide range of different hardware: fixed nodes, other mobile nodes, smartphones, pc, tablets, etc.

The basic idea that lies behind the concept of the mobile node, is the socialization and modeling of the networks according to the interests and needs of communities and citizens, completely changing the paradigm. At present, networks are designed and deployed based on corporate and/or government interests, this fact does not allow the home user to control its data flows and, much less, get involved in the process.

Thanks to the low costs of the equipments and to the fact that they are completely self-configuring, it is now possible to model the urban space based on people or communities who are the ones that will be using these infrastructures. It allows users to take part in the process from beginning to end, from bottom to top. Thus, the city and the data flows that govern it will be totally adapted to people, breaking the barriers that are currently imposed by the well established designs we have to deal with.

The access to independent and self-managed transmission channels in the public space has very high associated costs. Most communities are unable to meet these expenses and therefore, they cannot increase their representativeness within the society controlling their traffic. This is the particular problem that this work

addresses. Using bottom-up models we try to respond to this necessity by creating a new concept of "social routing" agreed on consensus, with open protocols that provide access to the physical and digital public space. This configuration offers a model with low costs, high social representativeness and ease of reproduction.

This is directly related with open mesh networks and how are they being deployed nowadays. We can control our traffic and model the networks according to our interests, using this platform to expand these networks in that cases that we need it.

The streaming solution we propose, perfectly fits those networks and specially a scenario with the mobile node where we may want to share the content we are watching at that moment.

Chapter 2

STATE OF THE ART

This project consists on studying how are, currently, being build the open mesh networks in cities and what mechanisms we have to contribute or improve them. There are many different ways to improve this networks, namely, we can use different hardware, different software, build applications which run over them, etc.

First of all, we will analyze how these networks are, and how they are operating to have a better idea of what we want to improve.

2.1 Mesh and MANET networks

2.1.1 Definition and properties

When we talk about mesh networks, we refer to networks where all the participants are also routers. If we had to set a single definition the following can be a good one:

"*A Mesh network is one where all nodes (participants) are routers, meaning that all the nodes accept and forward packets from other nodes according to the routing rules.*" [Escrich, 2012a]

More specifically, we want to talk about Wireless Mesh Networks (WMN) which may refer also to the users of the network, and can be defined as follows:

"*Wireless mesh networks often consist of mesh clients, mesh routers and gateways. The mesh clients are often laptops, cell phones and other wireless devices while the mesh routers forward traffic to and from the gateways which may but need not connect to the Internet.*" [Huynh, 2011]

To summarize, we have that mesh networks are basically networks which are not defined by the topology (physical layer) or the kind of links between two nodes (link layer). They are defined by the way the nodes operate among them, there are not master/slave or node/supernode distinctions and so, all the nodes have a similar function. In addition, the clients (end users) do not notice any difference (between mesh or other kind of networks) when they connect to the mesh, it is totally transparent for them.

What was mentioned above was that there were three kinds of nodes: clients, routers and gateways. The routers and gateways operate in exactly the same way when they have to route packets within the network, the only difference is that the gateways may be connected to a wider network, namely, the Internet and they can route packets to this other network. So, mesh routers just route packets inside the mesh while mesh gateways can also route packets to the outside.
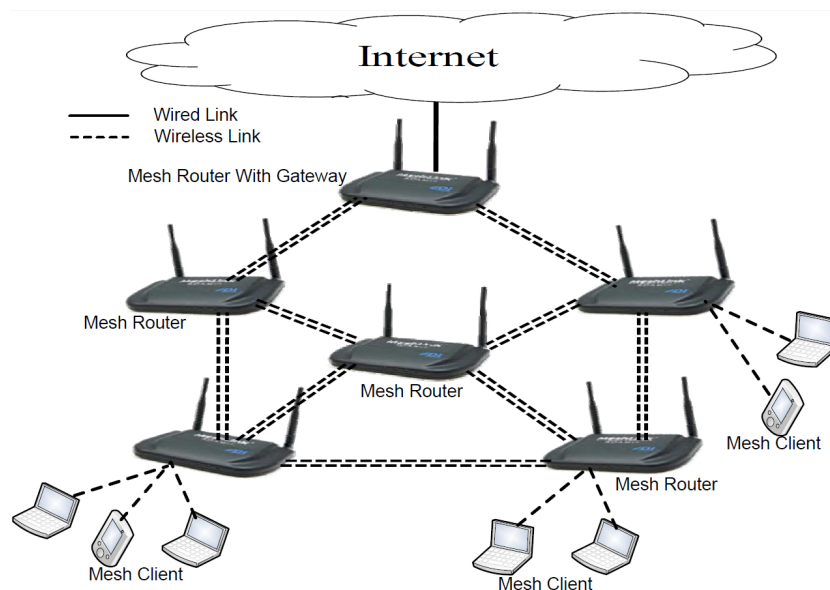

Figure 2.1: Wireless Mesh Network Example

http://www.intechopen.com/source/html/37888/media/wmn11.jpg

4

Then, we can say that WMN are a subtype of mesh networks. They have all the properties of these networks with the only difference that all the nodes are connected wirelessly.

2.1.2 IEEE 802.11 Operating modes

IEEE 802.11 networks can operate in two different modes: Infrastructure mode and Ad-Hoc mode.

- **Infrastructure Mode:** In this mode we have a central point named Access Point (AP) that creates a Basic Service Set (BSS) zone in which all the packets have to go through the AP. So, when two nodes in the network want to exchange information, the sender will transmit the packet to the AP and the AP will forward the packet to the receiver. The BSS zone is identified by the MAC address of the AP, which is called BSSID in this context. Furthermore, we can say that a master/slave model is followed in infrastructure mode.
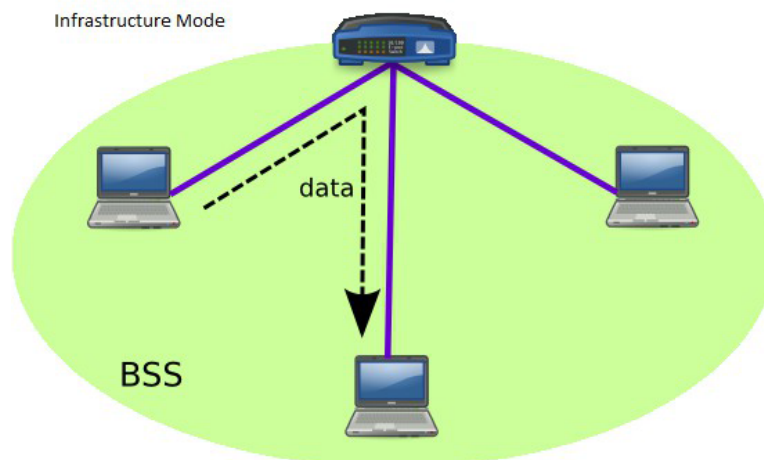

Figure 2.2: Infrastructure Mode BSS

[Escrich, 2012b]

- **Ad-Hoc Mode:** In Ad-Hoc mode, all the participants play the same role. Therefore, every single node connects with all the nodes it can, and so the central point idea disappears. To identify which participants are in the same network we just need to find those who have the same BSSID. At this point, all the machines directly connected can exchange information but, a layer 3 routing protocol is required to allow the communication between nodes which are not connected directly.
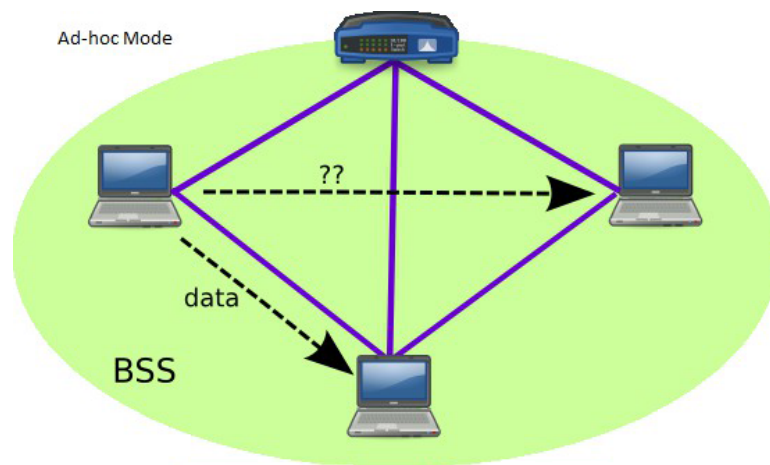
Figure 2.3: Ad-Hoc Mode BSS

[Escrich, 2012b]

2.1.3 MANET Networks

As mentioned in previous sections a WMN network is a network where all or part of the devices can route packets. We can distinguish three different kind of nodes in these networks:

- Client: Simple device which cannot route packets. This devices just use the networks to get access to The Internet.

- Router: Route packets throughout the network.

- Gateway: Route packets throughout the network but also to the outside.

So, a MANET network is a WMN with mobile devices.

"A Mobile Ad hoc NETwork (MANET) is a kind of wireless ad-hoc network, and is a self-configuring network of mobile routers (and associated hosts) connected by wireless links - the union of which forms an arbitrary topology."[Ramesh et al., 2010]

As mentioned, when we talk about MANET we are also referring to networks which have a self-configuring property. This property relies on the fact that the routers within the network are free-to-move anytime and to anywhere.

2.2   Open MANET Networks in Barcelona

Nowadays, in Barcelona (and in some other cities) some MANET networks are being deployed. It is important to say, that according to many definitions in the literature these networks would be defined as WMN but, the owners of these networks define them as MANET and we will use this name for consistency.

We will study this case in particular because is the one we are more familiar with. In general terms, all the deployments work in the same manner and so, studying one single case can give us a general a idea of all of them.

Normally, all these deployments only differ in some points, these are the more relevant ones:

- The funding scheme

- The kind of hardware

- The firmware used

2.2.1   Funding scheme in Barcelona's MANET networks

There are some examples of MANET networks successfully deployed in Barcelona:

- Gràcia

- Sants

- Poble Nou

7

- Sant Joan Despí

All these networks have been deployed using the same funding scheme, promoted by the Guifi.net foundation, and is, somehow, based on crowd-funding. Basically, anyone can install a new node and join to the mesh. There is just one restriction: you have to have direct vision with, at least, another node in the network. If you achieve this requirement, you can buy and install the node yourself and you expand the network. So, this is crowd-funding because every person joining the network funds its own equipment, which has the only requirement of being compatible with the firmware used. Since many people has no experience on installing and configuring the nodes, and despite the fact that both the hardware and software are specially designed to allow non-technical people to use it without problems, the foundation provides technical support for those who need it.

2.2.2   The Kind of hardware in Barcelona's MANET networks

Being that the users are the owners of the network and so, the people who buy the equipment, usually low-cost hardware is used. Low-cost does not mean, low-quality, in fact some of the hardware used have very high features and a very good performance. Normally, the hardware used is from Ubiquity Networks (NanoStation, Rocket, Bullet, etc.), but it is not strictly necessary. Actually, any hardware compatible with the linux distribution openWRT is accepted.

There are more information about some of this devices in Appendix 2.

2.2.3   The Firmware used in Barcelona's MANET networks

This is maybe the most important difference between all the deployments around the world. In most of them, there is a common point: all the firmwares are openWRT based, but they have some differences. Particularly, in Barcelona the first deployments used the Gracia Sense fils (GSF) firmware, developed since 2003 for the Gracia Sense Fils Wireless Community[1]. Some years later, the firmware became outdated since the protocol versions were old and the configuration of the nodes was hard. Then, they decided to create a new firmware with bigger scope, they named it QMP and is the one being used currently (some nodes have to be migrated to QMP yet).

---

[1]More information at: http://graciasensefils.net/

QMP has been chosen in Barcelona because is a firmware that covers perfectly the needs in the deployments carried out there, and has been developed by themselves. Nowadays, in most of the deployments a new firmware is designed specially for them, and despite the fact that many of them could be used in other deployments, it does not normally happen. It is not an efficient way to work, and because of it a new initiative has just started, it is called "libre-mesh"[2]. This initiative tries to build a new firmware based on three existent ones: QMP (Spain), AlterMesh (Argentina) and eigenNet (Italy). It can be a good opportunity to merge all the best points of every firmware a create a new version that fits in many different scenarios.

2.3   QMP firmware basics

2.3.1   Main Features

QMP [3] is an Operating System designed for embedded devices, a firmware. The main features of this firmware are:

- OpenWRT based

- 802.11a/b/g/n support

- IPv6 native

- IPv4 tunneled over IPv6

- Auto configuration system

- Web GUI to monitor and configure

- Visualization tools (maps, graphs, etc.)

- Automatic dynamic routing (zero-conf)

- BGP (Border Gateway Protocol) support

- Open Source

---

[2]More information at: http://libre-mesh.org/
[3]http://qmp.cat

2.3.2 Quick Deployments

QMP has been specially designed and developed to achieve quick network deployments. To do so, they have created the auto configuration feature which plays a very important role within the firmware. When we talk about quick deployments, we mean that we have to accomplish these requirements:

- The deployment must be performed as fast as possible.

- It must be able to be done by non-technical people.

- It must be possible in most situations.

2.3.3 Addressing

QMP uses three different kind of IP addresses:

- IPv6 Unique Local Address (ULA): IPv6 private range to be used internally in the mesh. These IPs are used for the communication among the nodes in the mesh network, and so they are not neither valid nor routable outside.

- IPv6 Réseaux IP Européens (RIPE): IPv6 public IPs range (6to6 tunneling). These are globally valid and routable.

- IPv4: IPv4 private range to connect with the final user (4to6 tunneling). They are assigned to the final users attached to a node in the mesh, when they transmit any packet that has to travel throughout the mesh is encapsulated in an IPv6 packet (tunneling).
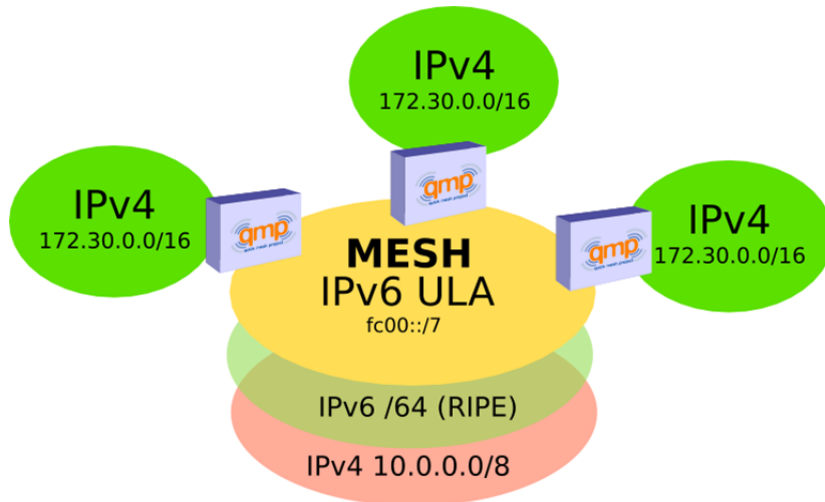
Figure 2.4: QMP Addressing

[Escrich, 2012b]

2.3.4 Operating modes

The firmware has two different operating modes, depending on the scenario we should choose one or the other:

- Roaming for fast deployments: All the access points in this mode will have the same IP and the same ESSID in order to allow users mobility, namely, they will not lose the connection although they switch from an AP to another. Every AP implements a Network Address Translation (NAT) and so, two users attached to different APs will not have direct vision between them.

- Community: Every node will have a randomly assigned IPs range and will announce this range through the mesh. There is not NAT, every user has direct vision with the others (1 hop away from the IPv4 network layer point of view), but mobility is not allowed (no roaming).
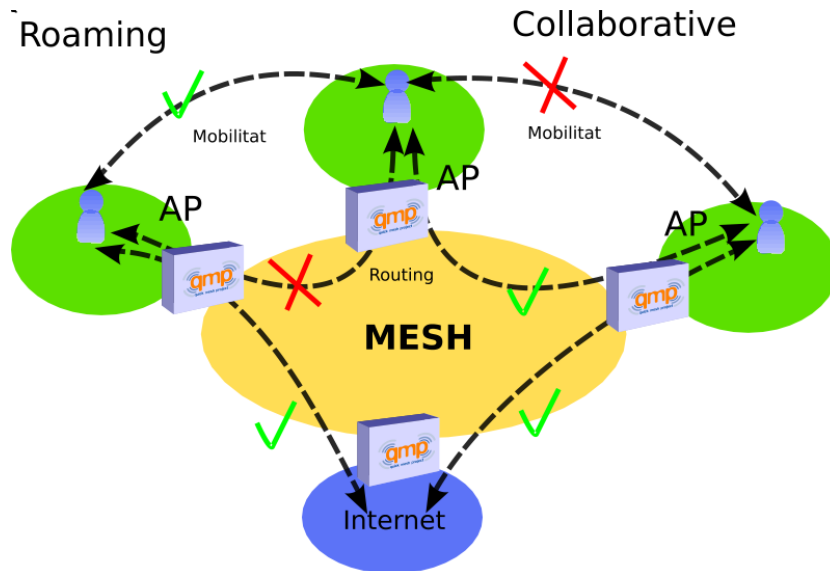
11

Figure 2.5: QMP Operating Modes

[Escrich, 2012b]

2.3.5 DRP

The firmware uses different protocols:

- BMX6 as the main DRP.

- Optimized Linked State Routing version 6 (OLSR6) as a backup DRP.

- Babel as a backup DRP but optional.

Usually, two networks are built in parallel the main one using BMX6 and the backup on using OLSR6. There are two reasons for that:

1. To prevent that a node gets isolated (no neighbors) due to a single BMX6 failure.

2. To make performance measurements of both protocols in exactly the same environment. This was very useful at the beginning of the project to decide which protocol was better in every scenario.

All these three protocols use IPv6 ULA to talk to other nodes and are isolated at the link layer (MAC) using VLANs. It allows the protocols to work in parallel (if necessary) without interfering with the other protocols.

2.3.6 BMX6

Since BMX6 is the main DRP we are going to analyze it deeply:

"BMX6 is a table-driven routing protocol for wireless mesh networks. [...] its goal is to compose a path from source to destination by deciding on each node which will be the next hop. BMX6 is a distance vector protocol, since the information each node manages is [...] destination node, next hop and cost. " [Neumann et al., 2012]

In terms of dissemination - how the nodes exchange the information - we can distinguish between two different states: transient and steady.

- Transient phase: the nodes exchange information related to the environment: Internal IDentifier (IID), nodes description, links, etc. Thanks to this information, ever single node builds a dictionary table that translates between IID to the global hashes of the full node description.

- Steady phase: every node has local information state (IID-to-hash dictionary) and global information state as hash-to-description. So, during this phase there is just a small exchange of packets that informs about link metrics and network changes. Thank to the tables set up in transient phase, those packets do not need to use the 128 bit IPv6 address as the identifier, they can use the 16 bit IID which produces a much lower overhead.

To summarize, we can say say that BMX6 controls the overhead better than other protocols (some experiments demonstrate that, for example: [Neumann et al., 2012]). There is a big overhead at the very beginning (transient state) and later it becomes very low during the steady state. So the main features of this protocol are:

- Pro-active: Uses UDP flooding to periodically send Originator Messages (OGM) and build a routing table.

- Destination-sequenced, Distance-vector (DSDV): Every node just knows which neighbor is better to reach another, namely, they do not need to know the entire topology, just the best paths.

- Does not use IP as node identifier, it uses global identifiers using SHA2 hashing.

In terms of the kind of frames used by BMX6 we have two different types: periodic messages, periodically generated on every node and occasional messages, exchanged only when necessary. The most used are the periodical which are the following:

Frame name	Description
HELLO_ADV	Hello advertisement. Used for letting neighboring nodes detect the link quality in transmit direction (from sending to receiving node).
RP_ADV	Rx probe advertisement. Used for reporting about reception rate of hello messages from neighboring nodes.
OGM_ADV	OGM advertisement. Used for updating periodically route and metric information over the mesh.
OGM_ACK	OGM acknowledgment. Used for acknowledging the previously reception of a full OGM_ADV frame.

Table 2.1: BMX6 Frames

[Escrich, 2012a]

Chapter 3

CONTRIBUTION

As we mentioned in previous chapters, we wanted to find ways to contribute to current open mesh networks, by providing ideas, introducing new hardware (or new ways to use the existing one) or building applications that run over the network.

We can divide the contribution of this project in three different points:

1. Study and documentation of the Mobile Node project

2. Installation of a node within Poblenou Mesh

3. Development of an audio and/or video streaming Android application

3.1   Mobile Node Project

"*The Mobile Node is an independent transmission unit made up of open protocols; it is a device with its natural space in the urban environment, which enables a digital fabric to be created using wireless networks. The project has become a process of constant research and development which explores the possibilities of interactivity in the digital public space.*"[1]

The idea is the socialization of the networks, allowing the users to get involved in the process and prioritize their interests and needs. Nowadays, the paradigm is

---

[1]Extracted from Sonar+D presentation text: `http://www.sonar.es/es/2013/prg/sm/guifi-net-mobilitylab-net-qmp-cat-exo-cat:-nodo-mC3B3vil_27`

completely the opposite, ergo, governments or organizations decide how the networks have to be designed and where is better to deploy them. So it does not allow the user neither to control its data flows nor to feel part of the network.

In other words, what we are doing with the Mobile Node, is making the most of the hardware and firmware we have, combining them and using the resulting device in a different way. So, a mobile node, in our context, is a device that can be used to expand networks already deployed and give coverage to a zone where there was no coverage at this time. Therefore, we are taking advantage of the "Auto-configuration" feature of QMP to provide Internet Access rapidly.

The main advantage of this device, is the speed and freedom that offers. Given this features, the most common usage will be as a temporary solution to cover a particular zone during a certain period of time. It can be very useful in events, congresses, concerts, parties, etc. Which will take just a few hours or days. Using this device in cities like Barcelona, where fixed mesh networks are quite extended, will allow the organizers of the events to provide Internet Access to the attendants in a very fast, easy and cheap manner.

Although covering events is, perhaps, the main usage we can give to our mobile node it is not the only one. Since we are talking about open networks, with open firmware we have the freedom to use them as we prefer. Normally, the mobile nodes are powered using batteries which will last just some hours, but it is not strictly necessary to use them that way. We could join a community or create a new one and model our own networks using mobile nodes, there are no restrictions considering that a mobile node is nothing but the same hardware and the same firmware used in a different way.

Given all these facts, that is the reason why we talk about changing the paradigm and start modeling the networks according to the interests of those who are going to use them.

So, the biggest contribution we did in that field, was to write a document talking about the project, its origins, how to build our first mobile node, the hardware we need, how to configure some devices and build a small mesh, the prototypes that are being tested nowadays, etc. Since this is a project which is being carried out by Efraín Foglia with the support of Mobility Lab, we just took all the infor-

mation and started doing experiments to build the networks and the nodes, but did not do anything new. So we created a document gathering all the information that will allow beginners to start learning and working in that field. This document can be found at appendix 2.

The latest prototype of the mobile node, will be presented on June 13th,14th and 15th in Sónar event in Barcelona, here we have some pictures of it:
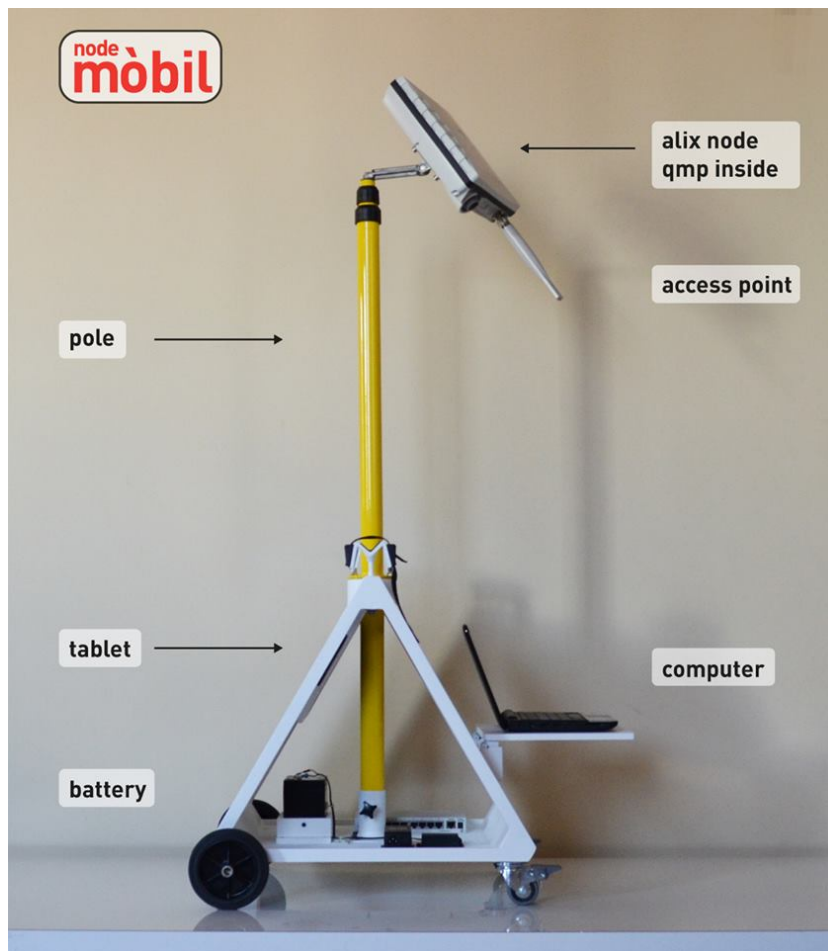
Figure 3.1: Latest Mobile Node Prototype 1

Figure 3.2: Latest Mobile Node Prototype 2

3.2 New node in Poblenou Mesh

As we already mentioned in previous chapters, nowadays, Barcelona has plenty of open mesh networks working, one of them is the one placed at Poblenou neighborhood. Most of the nodes of the network, are still using the old firmware GSF and some members of guifi.net and some volunteers are trying to update all of them to QMP. Since our campus is placed in Poblenou and the main building in one of the tallest of the neighborhood we decided to contribute to the mesh by placing a node on the roof of that building.

Being that it was the first time we were working with this kind of firmware and hardware we followed a set of steps before deploying the node.

3.2.1 Choice of Hardware

Given the position and height of the building, and the physical situation of the other nodes, we made a first approximation which said that we should cover an angle between 70 and 100 degrees. Thanks to the previous experience of the QMP and guifi.net teams, the decision did not take too much time since they solved many similar situations in the past. Their recommendation was to use a Rocket M5 from Ubiquity Networks together with a 90 degrees sector antenna.

The main features of these hardware are:

- Rocket M5:

    - Description: Outdoor base station which transmits in the 5GHz band

    - Protocols: 802.11a, 802.11n

    - Outside reach: more than 50km (depending on the antenna)

    - Transmission frequency: 4900-5800MHz

    - Max. Transmitted power: 27 dBm

Figure 3.3: Rocket M5

- Airmax 5G-17-90:

    - Description: 90 Degree, 5 Ghz, MIMO, 17 dbi Sector antenna for ROCKET M5

Figure 3.4: Airmax 5G-17-90

3.2.2 Lab Testbed

In order to test the hardware and firmware before deploying the node, we created a testbed in the lab. We took three different nodes, installed QMP and built a small mesh to get familiar with this kind of networks.

The devices we chose were: Nanostation M5, Alix 2D2 and Rocket M5. We flashed[2] them with the most recent version of the firmware and we got an indoor mesh:

BMX6 (5GHz)
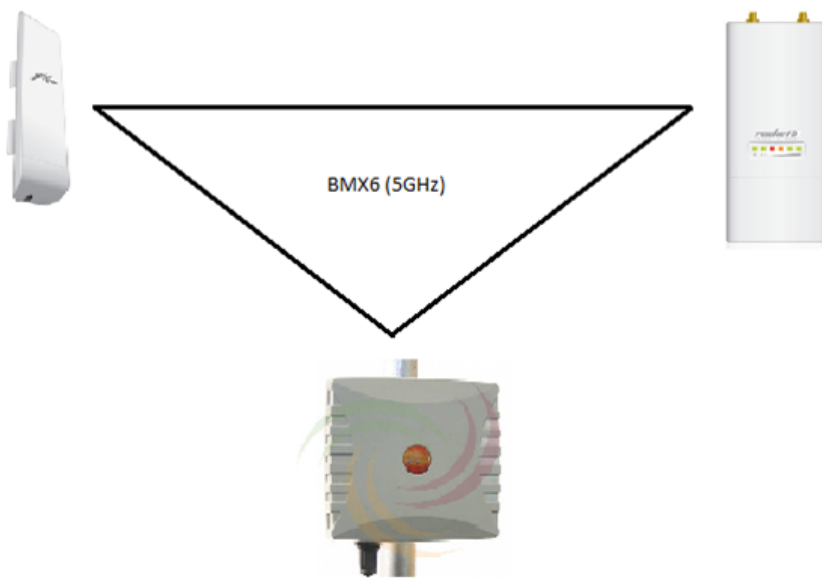
Figure 3.5: First testing mesh

Thanks to the firmware, the nodes recognized each other in a few seconds and the network was deployed automatically. The next step was to set at least one of the nodes as a gateway to a wider network, the Internet in this case. We connected the Wide Area Network (WAN) interface of the Nanostation to a switch with an output to the Internet and waited for a result. In a few seconds, the Nanostation had announced this gateway throughout the mesh and so, all the nodes added a new entry to their tables showing the path to get this network.

---

[2]More details on how to flash the nodes in appendix 2
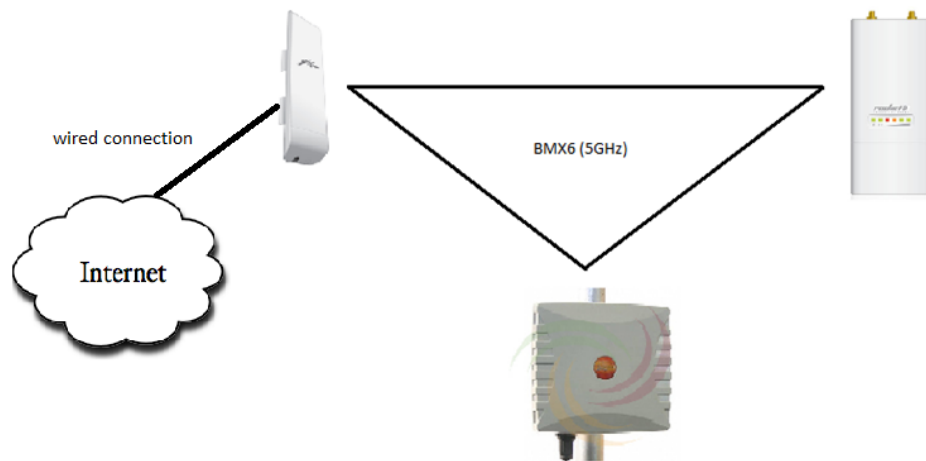
Figure 3.6: First testing mesh with gateway

Afterwards, we wanted to test if this gateway was actually working or not. To do so, we connected some clients via Wi-Fi to the Alix and tried to access to the Internet from them.
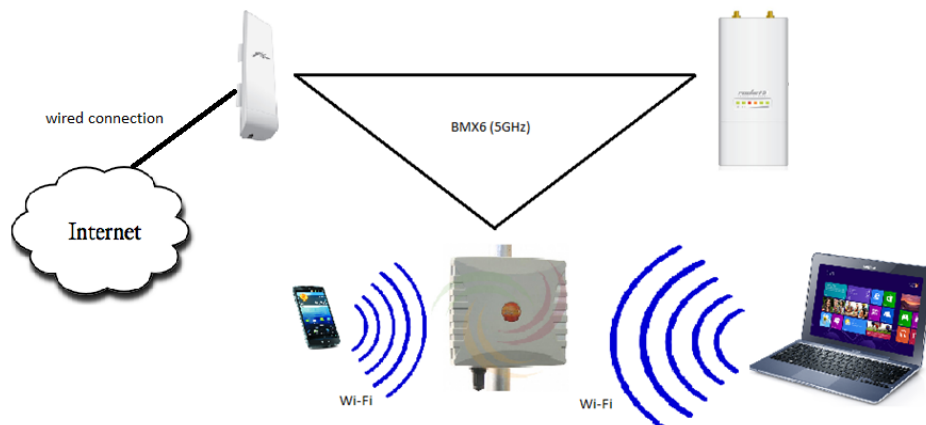
Figure 3.7: First testing mesh with gateway and clients

It worked as expected, and we could access the Internet without problems, fact that proved that we had built the mesh properly and that both hardware and firmware were doing their job perfectly.

3.2.3 Physical installation of the Rocket M5

Once we had tested that the hardware was working well and that we could build a mesh without problems, we proceeded to the installation of the device on the roof of the building. To do so, we just needed to point the antenna to the right direction and set the same transmission channel of the other nodes. Here we have some pictures of the result:

Figure 3.8: Rocket M5 installed on the roof of Roc Boronat Building (1)

Figure 3.9: Rocket M5 installed on the roof of Roc Boronat Building (2)

To be able to supply the node, and access to it to change the configuration or check any feature, we connected both the Power over Ethernet (PoE) and Local Area Network (LAN) interfaces to a switch which allowed us to do both things:
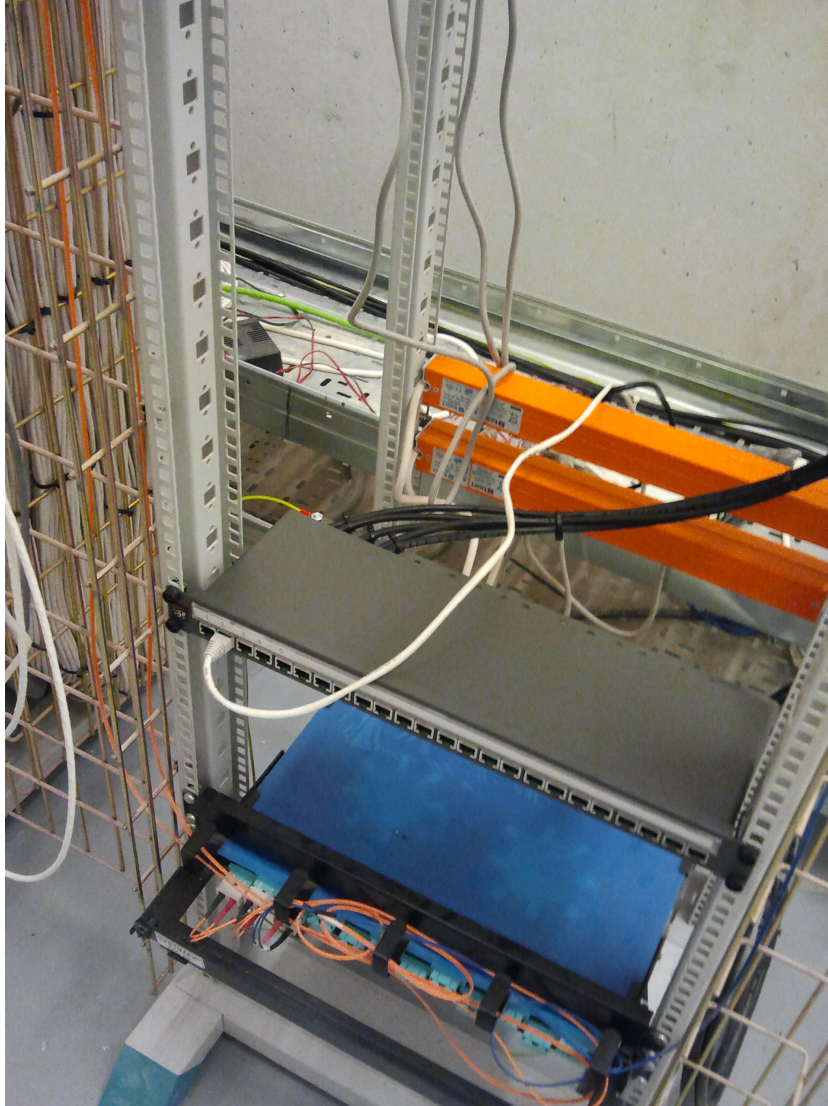
Figure 3.10: Rocket M5 connected to the Switch

At the moment, we have direct vision with one other node. At the same time, this node is connected to another one, so we have a mesh of three nodes deployed.
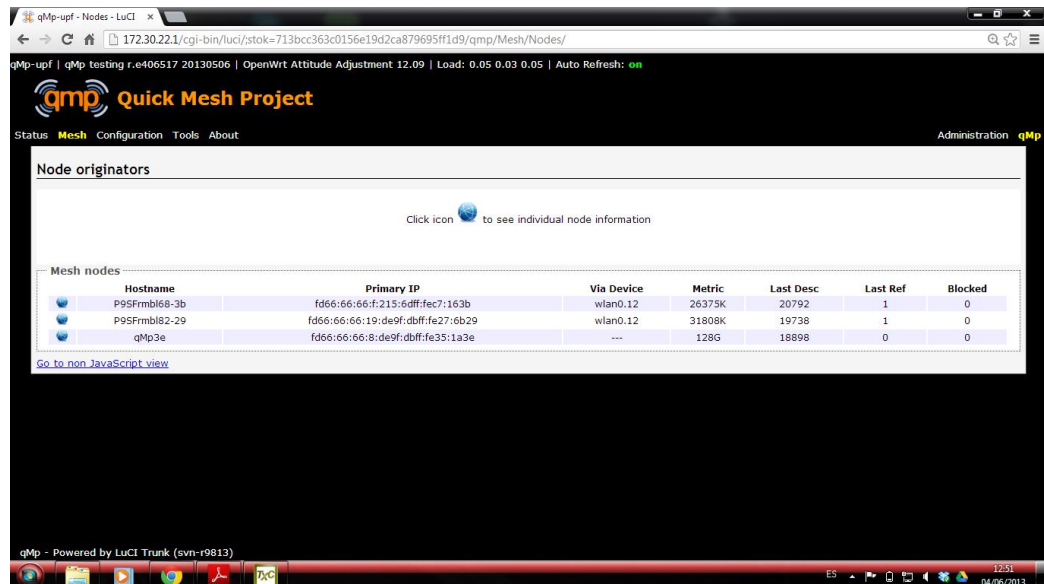
Figure 3.11: Poblenou Mesh nodes

We can also see the graph that displays how the nodes are connected among them:
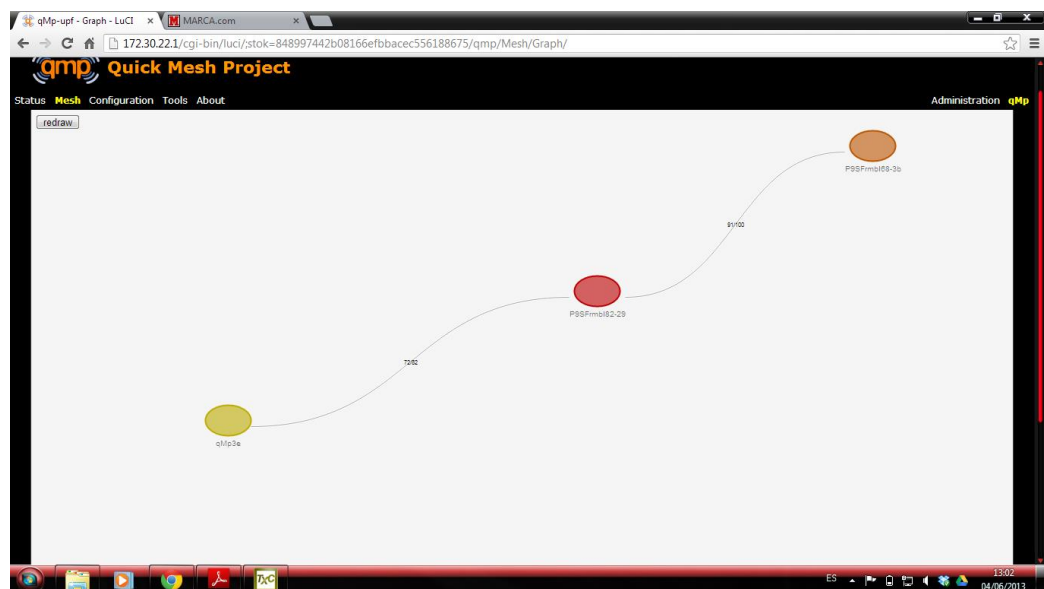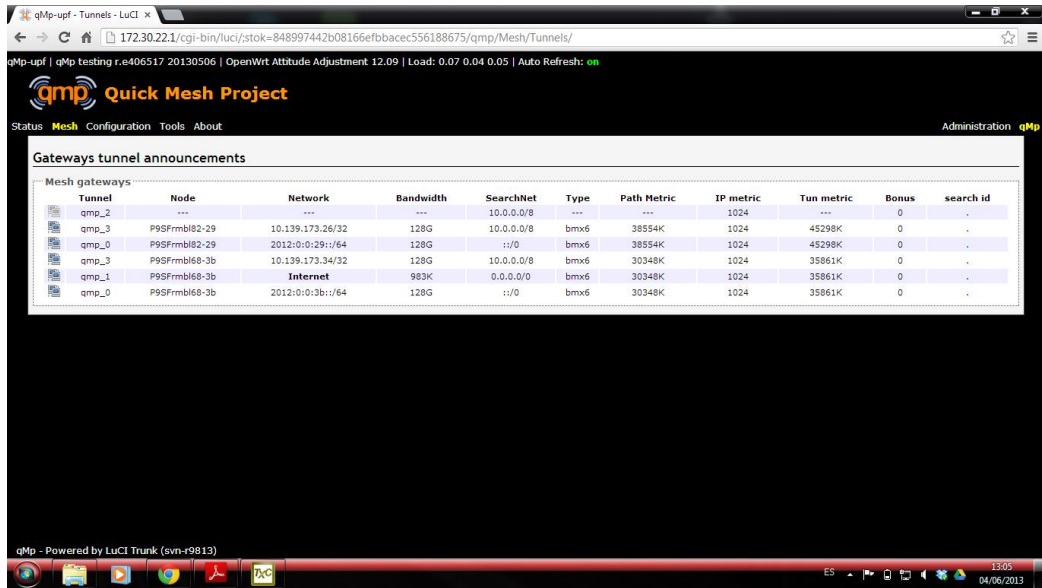
Figure 3.12: Poblenou Mesh nodes distribution

26

The main purpose of these networks is to provide Internet Access, and in this particular network one of the nodes is acting as a gateway to the Internet and we can access to it from any of the nodes in the Mesh.

Figure 3.13: Poblenou Mesh announced links

3.3 Mobile Node Android Application

3.3.1 Other streaming solutions

There are some other streaming solutions for android devices. Each of them has its pros and cons, and the reason why we decided to make a solution ourselves is that any of the existent ones fitted well with our project requirements. It is important to mention that most of these solutions are designed to video surveillance purposes instead of sharing contents.

Spydroid

Spydroid is an application that streams the audio and video captured from the smartphone to a computer in the same network. In addition, it is an open source application quite powerful but, at the same time, very complex for our purposes. The design did not fit with our specifications either. So, we decided to take some ideas and some parts of the code from this app, but we modified and added many other things.

Figure 3.14: Spydroid Logo

Ip Webcam

Ip Webcam is a very professional and powerful video surveillance application which basically streams the video and audio from the smartphone to a computer in the same network. The concept is the same as ours (streaming), but the goal is different. The main problem with this app is that is a closed solution and we do not have access to the source code. Despite this fact, we have used it to take ideas on the design and the settings to configure the camera.

Figure 3.15: Ip Webcam Logo

Wifi Camera

Wifi Camera is a very simple application that streams video from the smartphone to a computer in the same network. This application is an open source project which is too simple for our purposes. Nevertheless, we have taken some small parts of the source code to improve our application, and some ideas like the "stop" button.

Figure 3.16: Wifi Camera Logo

3.3.2   Our solution

This application, is specially designed to work in a scenario together with the mobile node, however, it may work in any LAN. The application basically streams the video and audio captured by the smartphone camera throughout the LAN and any client within the network can receive it. So, we have two separated entities acting in this scenario: the app which is the server and a computer which is the client.

3.3.3   Server side

The server side, will run the app in a smartphone and it will be usually done by the final user. The user, will just need to choose the options (frame rate, bit rate, height, etc) or leave the default ones and accept them. At this moment, the app is ready to receive incoming connections from any client. When a connection is received, the server will start the streaming session.

Generic Use cases

Case	Open the app and choose the options
Context	The users wants to choose a particular configuration before starting the stream
Primary Actors	User
Preconditions	The user has the app installed in the Smartphone and has launched it.
Post-conditions success	The user chooses and accepts the settings and the app is ready to start the stream
Post-conditions failure	The settings cannot be accepted due to an internal error or a user's mistake. The streaming session cannot start.
Main stage for Success	1. The user launches the app. 2. All the options (resolution, bitrate, framerate, etc) are chosen or left by default. 3. The settings are accepted. 4. The app is ready to start streaming sessions.
Extensions	-

Table 3.1: Server Side Use Case 1

Case	Share the URL on Twitter before accepting the settings
Context	The users wants to share its URL on twitter
Primary Actors	User
Preconditions	The User presses the "Share Url" button
Post-conditions success	The twitter app (if installed) or the browser launch with a tweet ready to be posted.
Post-conditions failure	Neither the twitter app nor the browser launch, or they do it wrong.
Main stage for Success	1. The user launches the app. 2. The user presses "Share Url" button. 3. Twitter app or browser launch. 4. The user presses the "send tweet" button. 5. The tweet is posted correctly.
Extensions	3.a Neither twitter app nor browser launch 3.a.1 Return to Step 2 5.a The tweet is not posted 5.a.1 Return to Step 4

Table 3.2: Server Side Use Case 2

Case	Stop a streaming session once started
Context	The user wants to close a session
Primary Actors	User
Preconditions	A streaming session is running
Post-conditions success	The streaming session is closed correctly
Post-conditions failure	The session keeps alive
Main stage for Success	1. The user launches the app. 2. The user accepts the options. 3. An clients connects. 4. The streaming session begins. 5. The user presses the "stop" button. 6. The session closes.
Extensions	6.a The session does not close 6.a.1 Return to Step 5 if it is a button problem 6.a.2 Close the app, and return to step 1 if it is a client problem or a generic app problem

Table 3.3: Server Side Use Case 3

Layout

These are the main layouts of the application:
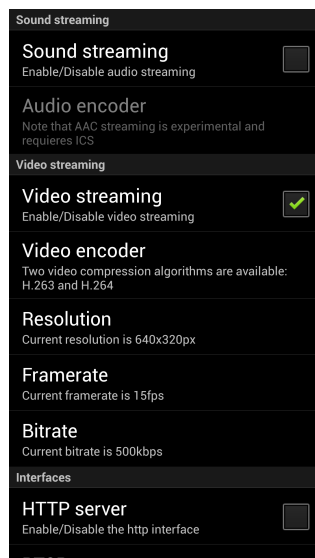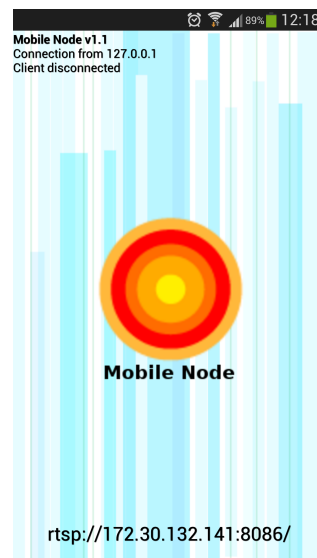

Figure 3.17: App Options Screen

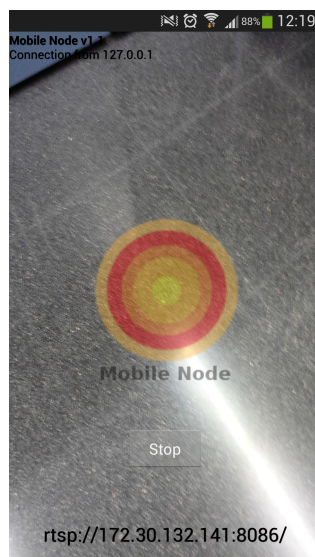Figure 3.18: App Main layout screen

Figure 3.19: App Main layout screen with camera on

3.3.4 Client side

For this side of the full system, we have used some open source software to receive the stream. It is important to say, that since this part of the system will not be used by the final user (in normal situations), but for the administrator, it is a little bit harder to make it work right.

We may want two different things when we receive the stream:

1. Play back the stream.

2. Record the stream.

Play Back

The simplest way to connect to the server and play the stream is using a software compatible with Real Time Streaming Protocol (RTSP) which is the protocol that uses the app to send the data. The most common one, and the one we have used in our tests, is VLC[3]. During the tests, we have realized that in some cases the app and the player are not fully compatible and there are some problems during the first seconds of the session causing an unexpected timeout.

Whenever we encounter this problem, we can use an intermediary between the app and vlc. This intermediary is a proxy server that establishes the connection with the RTSP Server in the app and allows multiple clients to connect to it an receive the stream. We have used an open source software called Live555 Proxy Server[4]. Using this intermediary, vlc is able to receive the stream much better (in some cases) than connecting directly to the app, and there are no errors. So the logic to make it work would be:

1. Have an app with the server running

2. Open the proxy server and introduce the app's URL

3. Open VLC and introduce the new URL generated by the proxy server
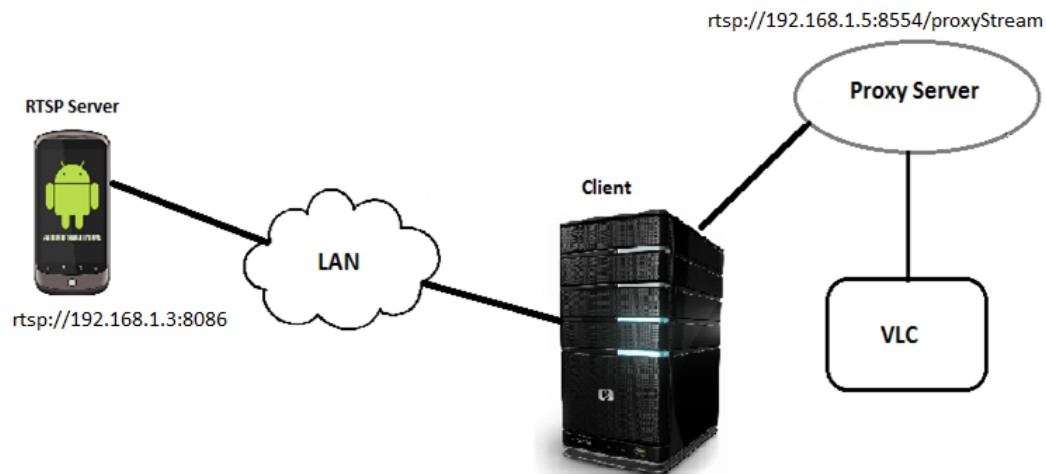
4. Start watching the stream

---

[3]http://www.videolan.org/vlc/
[4]http://www.live555.com/proxyServer/

Figure 3.20: App with proxy and vlc diagram

Record

In order to record the stream, we have been using a very lightweight open source software called openRTSP[5]. This software allows to connect directly with the app and save into a file all the data received. It is very useful in case we want to have a local copy of the stream to process it later in any way. It runs in a terminal an we just need to type a simple command indicating the record time, the height and width of the video, the bit rate and frame rate and the url of the server. An example of command would be this one:

openRTSP -d 40 -4 -f 15 -w 640 -h 480 -b 400000 "rtsp://192.168.1.3:8086" >video.avi

3.3.5 Internal Work flow and challenges

Now that we know how do the application work from the outside, we will take a look at how does it work internally and we will explain some of the challenges we have had to achieve in order to make everything function well.

---

[5]http://www.live555.com/openRTSP/

Internal Work Flow

The logic that the application follows is this one:

1. First of all, an activity that allows to choose the preferences is launched.

2. The user chooses the preferences and presses "Accept" button.

3. Internally, the app sends a message announcing its RTSP URL and some information about the video settings.

4. A new activity is launched, and a connection with localhost is carried out in order to configure the camera with the preferences of the user.

5. Then the server is ready to receive connections.

6. When a connection is received, the Server creates a unique session ID for this client and exchanges some control data with the client.

7. Then, the Android Media Recorder and other classes are used to record and send the data captured by the camera.

8. When either the client or the server (pressing stop button) decide to close the session, all the information is removed in the server and the app becomes ready to receive other connections.

Challenges

The main challenges that we had to face when developing the application were:

- How to stop a session from server side. This was a difficult part because, RTSP does not implement a way to close a session from server side by default. In addition, there was a complex multithreading scheme working behind and closing a session was not trivial. In the end, we found a way to stop all the threads and subthreads that allowed us to achieve our goal.

- How does the client side know the RTSP URL and the video features of the server? This was maybe the biggest challenge of the project, it is very easy to know the URL when you have the smartphone in front of you, but the idea of this app is that the server and the client are run by different people. So, we had to, somehow, let the client know the IP address of the server. We tried different solutions, like sharing the URL on twitter or implementing an XMPP based chat between server and client. Since any of these solutions were not exactly what we were looking for, we decided to implement a

multicast client-server application to share the information. IP Multicast is a technique for one-to-many or many-to-many real time communication over IP networks, the main idea is that all the participants join to a multicast group, and all the messages are sent just once setting the destination IP as the group IP. With this, and thanks to some protocols like Internet Group Management Protocol(IGMP), whenever a message is sent to this IP, all the members of the group will receive it. It was perfect to solve our problem, because we did not need to know the IP address of the others to exchange information.

3.3.6 Other information about the app

Some parts of the code, have been derived from some open-source projects such as: spydroid[6], android-eye[7]

The source code of the application can be found on github on the following site: https://github.com/ferry91/MobileNode-App

The application is under a GPL license, which means that any individual, organization or company has the freedoms to use, study, share(copy) and modify the software. And, since GPL is a copyleft license, all the derived works have to be distributed under the same license terms.

The application will be presented at Sonar in the Market Lab within the Sonar+D section[8] in Friday 15th of June together with the Mobile Node and the Mobility Lab platform.

---

[6]https://code.google.com/p/spydroid-ipcamera/
[7]https://github.com/Teaonly/android-eye
[8]http://www.sonar.es/es/2013/pg/market-place_221

Chapter 4

CONCLUSIONS

During this thesis we have analyzed and studied some different topics, which can work independently or together. This is maybe the most important feature of this field, mesh networks can be open or not, can run application over them or not, can use a any compatible hardware and in any imaginable way.

In terms of open mesh networks, we have studied some real cases which are currently deployed and working well. So, we could say that the model they have followed is, mostly, successful. As in any starting model, we always have some thing that can be improved, in particular in the cases we have studied we could highlight some of them:

- The networks have been deployed in some neighborhoods with a relatively small amount of nodes, and most people do not even know them. To become known and the process of involving more people can be one of the things that need to be improved. The profile of the people involved is usually technical, but to make this networks grow and become important in people everyday life, something has to be changed. The dissemination of this projects is usually small and only people who is working on that fields or is really interested on them get to know them. We should start looking at open networks as a benefit for the society since they can be modeled and deployed following our needs and not the needs of corporations or organizations. It is true that it is still difficult to find ways to get people really involved with this kind of projects, only a small percentage of the population participates actively on them because they normally think that these are things that will benefit others but not themselves.

- In terms of hardware there is no problem because there is a wide variety of

devices which can be used and they have similar performance and compatibility with the others.

- The firmware is different at this point, the main problem (as we mentioned before) is that every team develops its own firmware with the features they think are better. Therefore, for these people who is not used to this projects it can be a problem, being that they do not really know the differences among the different firmwares. It would be important to, somehow, standardize all the firmwares in one. Despite the fact that every deployment can have different needs, a single firmware may be able to be configured in different ways to reach that needs without having to develop a new one.

As we have seen during this thesis, making a network grow has a lot of different connotations. It is not strictly necessary to make it grow in terms of size or number of nodes, you can just improve the performance or add a new functionality by choosing the most optimal firmware or building an application.

The mobile node is a perfect example of this, a workstation which adds new possibilities to the existing networks with a combination of hardware and firmware. The mobile node application increases these possibilities even more, although it can be used without the mobile node itself.

To conclude, we can say that all this initiatives try to improve they way we access the Internet nowadays. Not necessarily in terms of performance or speed, but in terms of covering the needs of the final users and offering them the opportunity to decide what is better for them in every single situation.

Chapter 5

FUTURE WORK

This field is very open, and we will always be able to continue working and developing new features. Just to mention some of the possibilities that are being considered nowadays:

- Developing a Network Characterization Daemon (NCD) to publish some information about the nodes such as, WiFi links, Routing Neighbors, throughput of the links, number of routes, number of associated clients to a node, etc. This daemon may run inside every node and would publish the data periodically to a server in HTTP-JSON format. This can be a useful tool because we can derive lots of applications from it, for example, a mobile/web application to show the network topology in real-time, which would allow to find the best places to install new nodes.

- Try to develop a firmware which covers most of the needs of the users and becomes, somehow, a standard. This would avoid the current problem of having this variety of different firmware.

- Think about new ways to combine hardware, firmware and apps to offer new possibilities to this networks.

- Find ways to get more people involved on that initiatives, it is a difficult task but also a very important one.

As we can see, there are no limitations on that field. Any new combination, any new application related to this field can always improve the whole ecosystem.

Bibliography

[Diot et al., 2000] Diot, C., Levine, B. N., Lyles, B., Kassem, H., and Balensiefen, D. (2000). Deployment issues for the ip multicast service and architecture. *Network, IEEE*, 14(1):78–88.

[Escrich, 2012a] Escrich, P. (2012a). Quick deployment network using MANET. Degree thesis.

[Escrich, 2012b] Escrich, P. (2012b). Xarxes mesh. curs guifi.net 2012. Presentation.

[Foglia, 2010] Foglia, E. (2010). Mobile Node: Open Portable Infrastructure, Overlapping Digital Paths. *MediaCity: Interaction of Architecture, Media and Social Phenomena.*, pages 557–570.

[Huynh, 2011] Huynh, P. (2011). Simulation and analysis of wireless mesh network in advanced metering infrastructure.

[Mogaibel et al., 8 14] Mogaibel, H. A., Othman, M., Subramaniam, S., and Hamid, N. A. W. A. (2012-08-14). Wireless mesh networks - efficient link scheduling, channel assignment and network planning strategies.

[Neumann et al., 2012] Neumann, A., Lopez, E., and Navarro, L. (2012). An evaluation of bmx6 for community wireless networks. *International Workshop on Community Networks and Bottom-up-Broadband (CNBuB 2012)*.

[Ramesh et al., 2010] Ramesh, V., Subbaiah, D. P., Rao, N. K., and Raju, M. J. (2010). Performance comparison and analysis of dsdv and aodv for manet. *International Journal on Computer Science and Engineering*, 2(02):183–188.

Chapter 6

APPENDIXES

.1 Appendix 1: Pilot Charter

Pilot Charter - Mobile Node

Fernando Gros González

6 de diciembre de 2012

1.   Opportunity/Idea

The opportunity of carrying out this project resides in the BuB branch of the Commons for Europe initiative. This project was initially impulsed by Guifi.net and exo.cat among others and now has been introduced in this BuB4EU in order to receive this «last push». The main idea is to create a free transmission workstation that can be used in the urban space and contributes to the digital mesh through other networks.

2.   Description/Purpose

The project will consist on creating a mobile wireless workstation that creates or fits in an existent telecommunications infrastructure. It will have to be an auto-configurable device and will have the capability to interconnect with a wide range of different hardware. Furthermore, it will be able to work in many different spaces and without the need of an ISP.

3.   Goal

Improve the actual design of the mobile node, test it, create some documentation and find a way to make it easy-to-replicate.

4.   Objectives/Results

- Improve or create a new design of the mobile node.
- Allow citizens to be able to access low-cost technology with a big social return.
- Create documentation of the whole process.

5. Scope

Deploy and test this technology in Barcelona (Raval) and in London (Decision pending) and observe the results to replicate it in other cities.

6. Stakeholders

- guifi.net

- exo.cat

- qmp.cat

- ESADE

- UPF

- mobilitylab.net

- commons for europe

- Locality UK

- Creativecoop UK

7. Risks

- Prototype failures or firmware bugs that can delay the whole project.

- People unacceptance of the system or its social returns.

It is important to mention that this project is in an advanced stage and so, many of the risks that could have affected it are already mitigated.

.2 Appendix 2: Documentation

Universitat Pompeu Fabra

Mobile Node. Detailed documentation

Fernando Gros González
3/27/2013

Table of Contents

What is the Mobile Node?

The mobile node is a portable and auto-configurable transmission unit with wireless technology that offers mobility in the urban space. This node is designed to contribute to the digital mesh through existing networks. Furthermore, it provides connectivity with a wide range of different hardware: fix nodes, other mobile nodes, smartphones, pc, tablets, etc.

How was it born?

The basic idea that lies behind the concept of the mobile node, is the socialization and modeling of the networks according to the interests and needs of communities and citizens, completely changing the paradigm. At present, networks are designed and deployed based on corporate and/or government interests, this fact does not allow the home user to control its data flows and, much less, get involved in the process.

Thanks to the low costs of the equipments and to the fact that they are completely self-configuring, it is now possible to model the urban space based on people or communities who are the ones that will be using these infrastructures. It allows users to take part in the process from beginning to end, from bottom to top. Thus, the city and the data flows that govern it will be totally adapted to people, breaking the barriers that are currently imposed by the well established designs we have to deal with.

What for?

The access to independent and self-managed transmission channels in the public space has very high associated costs. Most communities are unable to meet these expenses and therefore, they cannot increase their representativeness within the society controlling their traffic. At this point, is where the project is intended to emphasize. Using bottom-up models we try to respond to this necessity by creating a new concept of "social routing" agreed on consensus, with open protocols that provide access to the physical and digital public space. This configuration, offers a model with low costs, high social representativeness and ease of reproduction.

Related texts

Foglia, E. (2011). Colisiones en la MediaCity, Prácticas sociales y artísticas sobre el trazado digital. En: Impresión Expandida / Expanded Print. (ed.) Eloi Puig, Alicia Vela y Antonia Vilà. ED: Barcelona, Universidad de Barcelona.

Foglia, E. (2010). Mobile Node: Open Portable Infrastructure, Overlapping Digital Paths. MediaCity: Interaction of Architecture, Media and Social Phenomena. ED: Weimar, Bauhaus-Universität.

Component Description

Hardware

Below there is a list of the components that form the mobile node and the other nodes that make possible to create mesh networks and its main features:

- Ubiquiti Airmax Bullet M5:
    - Description: Wireless outdoors device that transmits in the 5GHz band.
    - Details:
        - Protocols: 802.11a, 802.11n
        - Outdoor range (depending on the antenna): more than 50km
        - Transmission Frequency: 5470MHz-5825MHz
        - Maximum transmitted power: 27 dBm
        - Bandwidth: Until 100Mbps

- Ubiquiti Nanostation M5:
    - Description: Outdoors CPE (*Customer Premises Equipment*) device that transmits in the 5GHz band.
    - Details:
        - Protocols: 802.11a, 802.11n
        - Outdoor range: more than15km
        - Transmission Frequency: 4900-5800MHz
        - Maximum transmitted power: 27 dBm
        - Bandwidth: Until 150Mbps

- Ubiquiti Aircam dome:
    - Description: IP cammera Ethernet cable supplied.
    - Details:
        - Supported protocols: IPv4/v6, HTTP, UPnP, DNS, NTP, RTSP, DHCP, TCP, UDP, IGMP, RTCP, ICMP, ARP
        - Image quality: 30 FPS, HDTV 720p
        - Video compression: H.264/MPEG-4/MJPEG

- Ubiquity Aircam mini:
    - Description: IP cammera Ethernet cable supplied.
    - Details:

- Supported protocols: IPv4/v6, HTTP, UPnP, DNS, NTP, RTSP, DHCP, TCP, UDP, IGMP, RTCP, ICMP, ARP
- Image quality: 30 FPS, 1MP/HDTV 720p
- Video compression: H.264/MPEG-4/MJPEG

- Arduino DUE:
  - Description: Electronic board based on a core ARM 32-bit processor that enhances the standard Arduino functionalities.
  - Details:
    - Microcontroller: AT91SAM3X8E
    - Digital I/O pins: 54 (12 have PVM output)
    - Analog input pins: 12
    - Analog output pins: 2

- Arduino WiFi Shield:
  - Description: Board that connects an Arduino to the Internet wirelessly.
    - Protocols: 802.11b, 802.11g
    - Encryption types: WEP y WPA2

- 2 batteries: Rechargeable Li-ion Emergency Power Battery:
  - Description: Battery for video camera, walkie talkie, camera, etc.
    - Capacity: 9000mAh
    - Input voltage: 12.6V
    - Output voltage: 12V

- 2 black passive screwable POE modules:
  - Description: It supplies electrical power to the device through an Ethernet port that supports PoE.
    - Connector: Ethernet RJ-45
    - Bandwidth: 10/100 Mbps

- Network cable and Ethernet RJ-45 connectors:
  - Description: Wire and connectors to assemble direct and crossover Ethernet cables.

- Female faston connectors:
  - Description: Connectors to be added to an electrical wire.

- Box with IP65 antenna:
  - Description: Outdoor compact case with an integrated antenna that operates across the 5GHz band.
  - Details:
    - Transmission frequency: 4900MHz-5900MHz
    - Maximum Gain: 20 dBi

- PC Engines ALIX 2D2:
  - Description: Low power computer board:
  - Details
    - Memory: 256 MB DDR DRAM
    - Processor: 500 MHz AMD Geode LX800

- Compex radio:
  - Description: Compact radio with a high performance and a low power consumption.
    - Transmission frequency: 2,4GHz y 5GHz
    - Protocols: 802.11 a/b/g/n
    - Chipset: Atheros AR5414

- Pigtail 15cm. UFL/N-H (female) Bulkhead:
  - Description: Connector good for most of Mini-PCI radios.
  - Details:
    - Connector type: UFL (IPEX, Hi-Rose) to N Female Bulkhead.

- Pigtail 5 GHz. UFL-SMA plug right angle 30 cm:
  - Description: Connector for box-integrated antennas and most of Mini-PCI radios.
  - Details:
    - Connector type: UFL (IPEX, Hi-Rose) to SMA right angle.

- Power supply 18v. 0,8 A (15 W) Alix 2C/3C):
  - Description: Switching power supply unit for Ubiquiti devices, Alix boards, etc.

- DMD: Dot Matrix Display
  - Description: LED panel that easily displays clocks, status screens, graphs, etc. Compatible with Arduino.

Software

The software used by the mobile node is a firmware called Quick Mesh Project (QMP). It is a GNU/Linux operating system based on OpenWRT for embedded devices. The main features of this firmware are:

- OpenWRT based
- 802.11a/b/g/n support
- IPv6 native
- IPv4 tunneled over IPv6
- Auto configuration system
- Web GUI to monitor and configure
- Visualization tools (maps, graphs, etc.)
- Automatic dynamic routing (zero-conf)
- BGP (Border Gateway Protocol) support (half implemented)
- Open Source

Types of Mobile Nodes

Due to the different needs identified during the development process of the prototypes, it was decided to create four different design options. Each with its own characteristics:

1) **Panzer**. Robust with a large electronic box which can hold more batteries. It has greater stability. 4 wheels.
2) **Tarantula**. Lighter than Panzer. It has extendable legs for stability. 4 wheels.
3) **Trip**. Special for trips. Based on existing plastic boxes. Tracker wheels very robust and computer equipment container.
4) **Cam**. Wireless high definition video camera which interconnects with mobile nodes. Lightweight and with the possibility of streaming video.
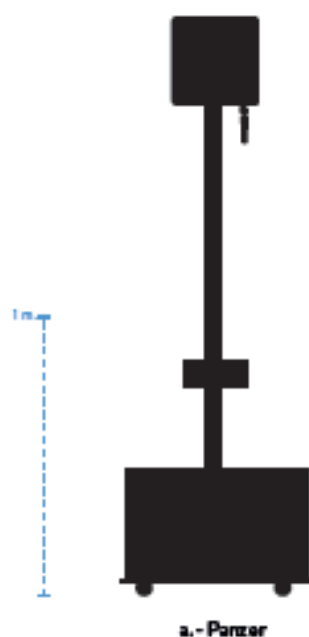
a.- Panzer
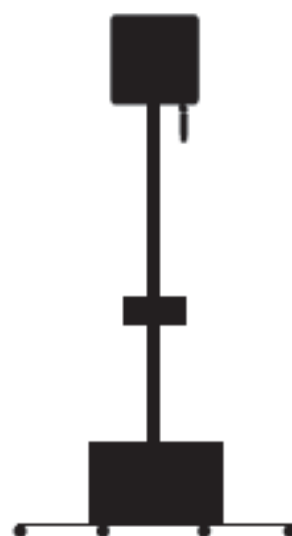
b.- Tarántula
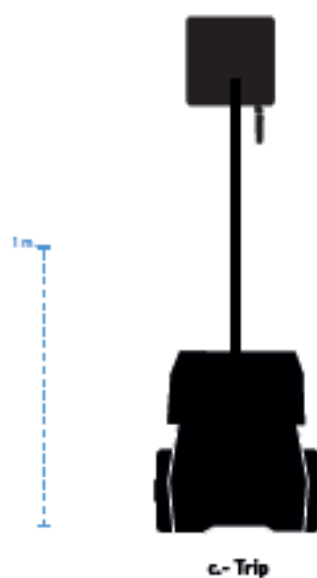
c.- Trip

d.- Cam

Assembly

We will divide the process in several steps

Box drilling and pigtail installation

The following pictures show the process:
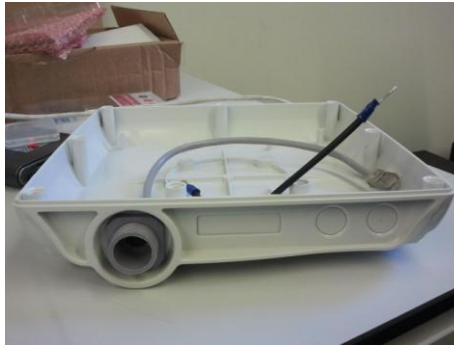
Undrilled box
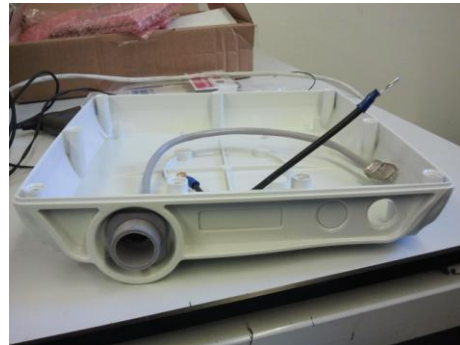
Drilled box
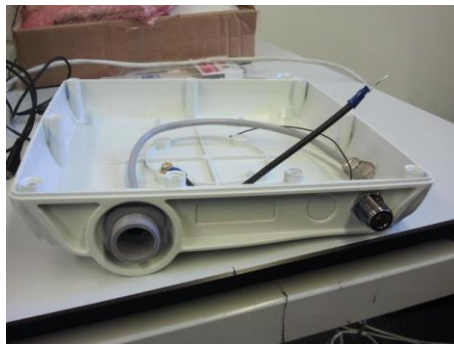
Box with pigtail installed

For this process, it is necessary a drill (or similar) to perforate the box and a cutter (or similar) to scratch and remove the remaining burr.

Installation of the board

It basically consists on adhere and affix the board to the box.

Affix the brackets to the board

Orientation and location of the board in the box
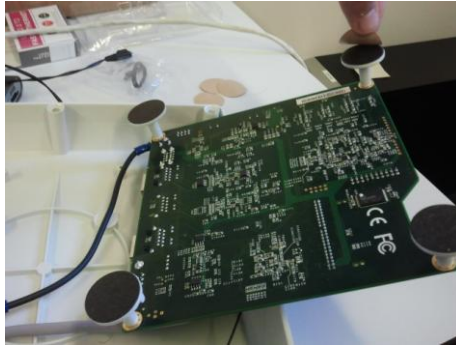
Remove the protectors of the brackets

Board affixed and wires orientation

For this process it is simply necessary a screwdriver.

Installation of the radios

Fit the radios into the appropriate slots of the motherboard:
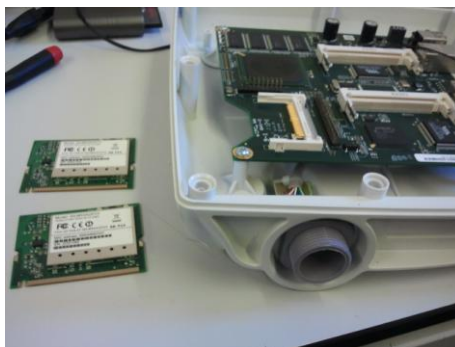
Compex radios

Radios installed on the board

Connection of the pigtails to the radios

Pigtails connected to the radios
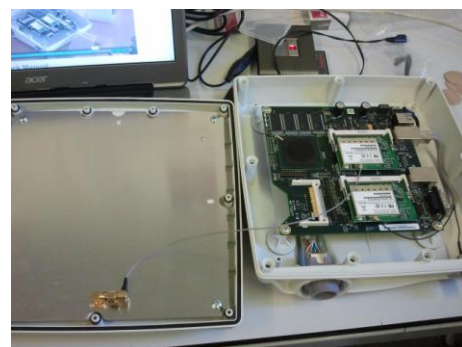
Pigtail connected to the upper cover of the box (antenna)

Installation of the flash card and switching-on

Card inserted into the slot

Power supply connected

Box closing

---

[1] For more information see the video tutorial: http://vimeo.com/58127395

Implementation

The following illustration shows a basic implementation of the Mobile Node with Alix antenna:
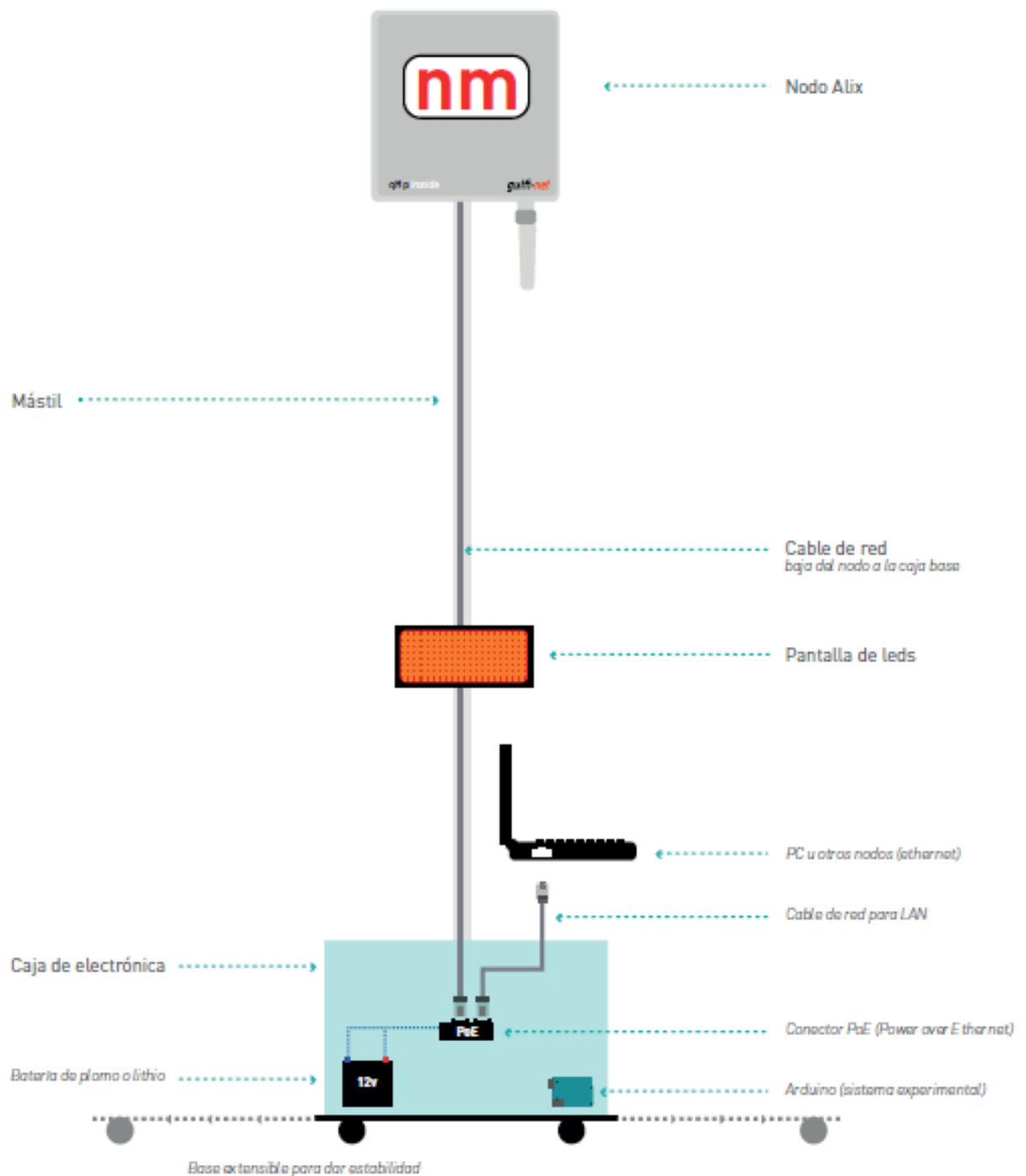
It is noted that we, basically, have the alix box fixed to a mast and connected to a battery. On the other hand, we have the LED screen that displays node status information and other relevant information using arduino and a PC that lets you quickly set the node.

With this implementation we will be able to offer coverage to a particular area by expanding an existing fixed network. A basic example of mesh network could be the following:
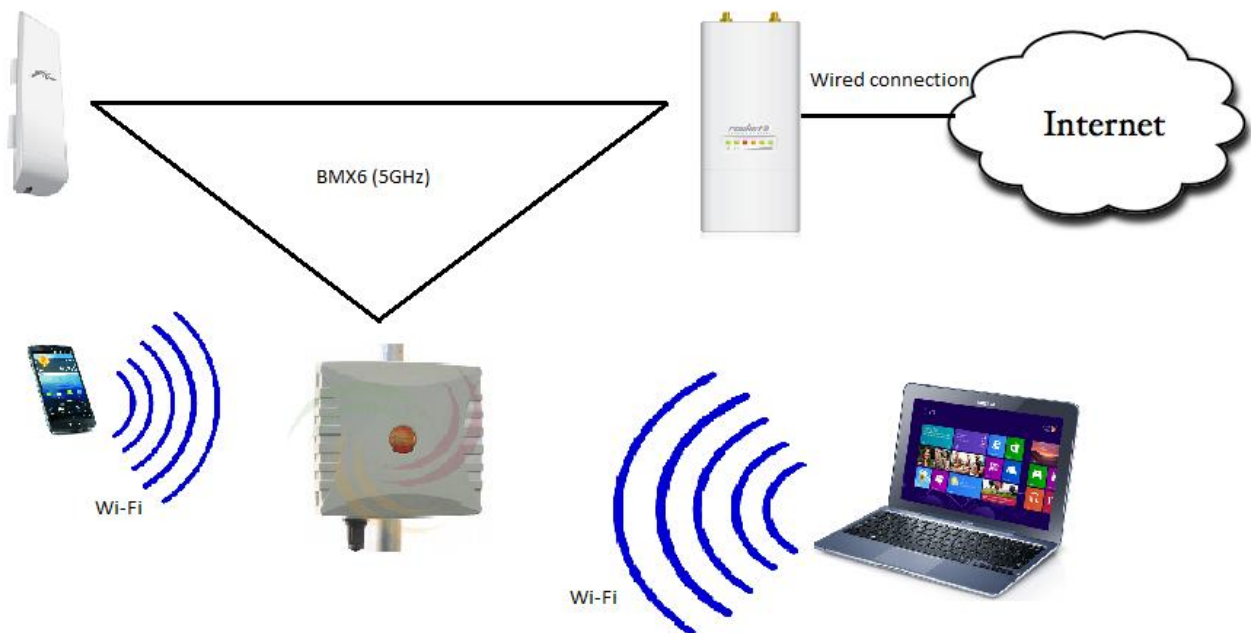
There is a fixed network composed for two antennas, thanks to our mobile node we can expand it and provide Wi-Fi connection to the final users.

Configuration of the nodes

To download and compile the source code of QMP we have to follow these steps:

1) Download the source code from the repository:
   a. (Recommended) `git clone git://qmp.cat/qmpfw.git qmpfw`
   b. wget -c -q -O - "http://qmp.cat/gitrevision_download?project_id=7&rev=anonymous" | tar zxvf -

2) Go to the source folder:
   a. cd qmpfw

3) (Recommended) Make a "checkout" of the "testing" branch:
   a. make .checkout_qmp QMP_GIT_BRANCH=testing

4) Compile specifying the target node type:
   a. make build T=alix
   b. To see the available targets: make list_targets
   c. If you have more than one core in your processor you can execute: make build T=alix J=N, where N is the number of cores that you want to use.

5) Finally, you will be able to find the compiled image in the images/ folder.

Nanostation M5

The steps to flash and configure the Nanostation M5 are the following:

1) Reset the device by pressing "reset" button during 10 second, until the lights start to flicker alternatively.
2) Install a tftp client, such as tftp-hpa.
3) We assign an Ip address to the wired connection of our PC within the 192.168.1.x/24 range.
4) From a terminal:
   a. `tftp 192.168.1.20` (Defaul IP of the Nanostation)
   b. `mode octet`
   c. `trace`
   d. `put *.bin` (Image compiled previously)
   e. `quit` (Exit the tftp client)

14

5) Change the IP address to on belonging to the 172.30.x.x/16 range

6) We open a browser and go to http://172.30.22.1 or http://admin.qmp

7) We use user: root and password: 13f

8) We can now utilize the GUI of QMP.

Rocket M5

We need to follow exactly the same steps that we followed with Nanostation M5.

Alix 2D2

Once assembled and mounted into its box, we need to install the firmware in the flash card:

1) Connect the flash card to a PC with a card reader (internal or external).

2) We execute the command `df -lh` to see the drives list connected to the machine.

3) We identify the name associated to our flash card, for example /dev/sda

4) From the terminal we go to the folder where we have QMP firmware

5) We execute the command: `if=firmware.bin of=/dev/sda,` to dump the firmware to the flash card.

6) We execute the command: `sync,` to empty the buffers and make sure that there are not more pending operations.

7) We extract the flash card and insert it into the Alix board.

8) We switch it on and wait a reasonable time until the firmware is loaded.

9) Later, we should be able to access the GUI connecting an Ethernet wire and going to the following URL: http://172.30.22.1 (We do not need to configure any Ip since it is assigned with dhcp)

If it does not work, we will need to repeat the dump process since it is possible that an error occurred during the copy process.

Basic features of the QMP interface

Below is a brief introduction to the main aspects of the interface qmp.

First, we have the welcome screen, where we will be asked for the username and password (U: root and P: 13f by default).
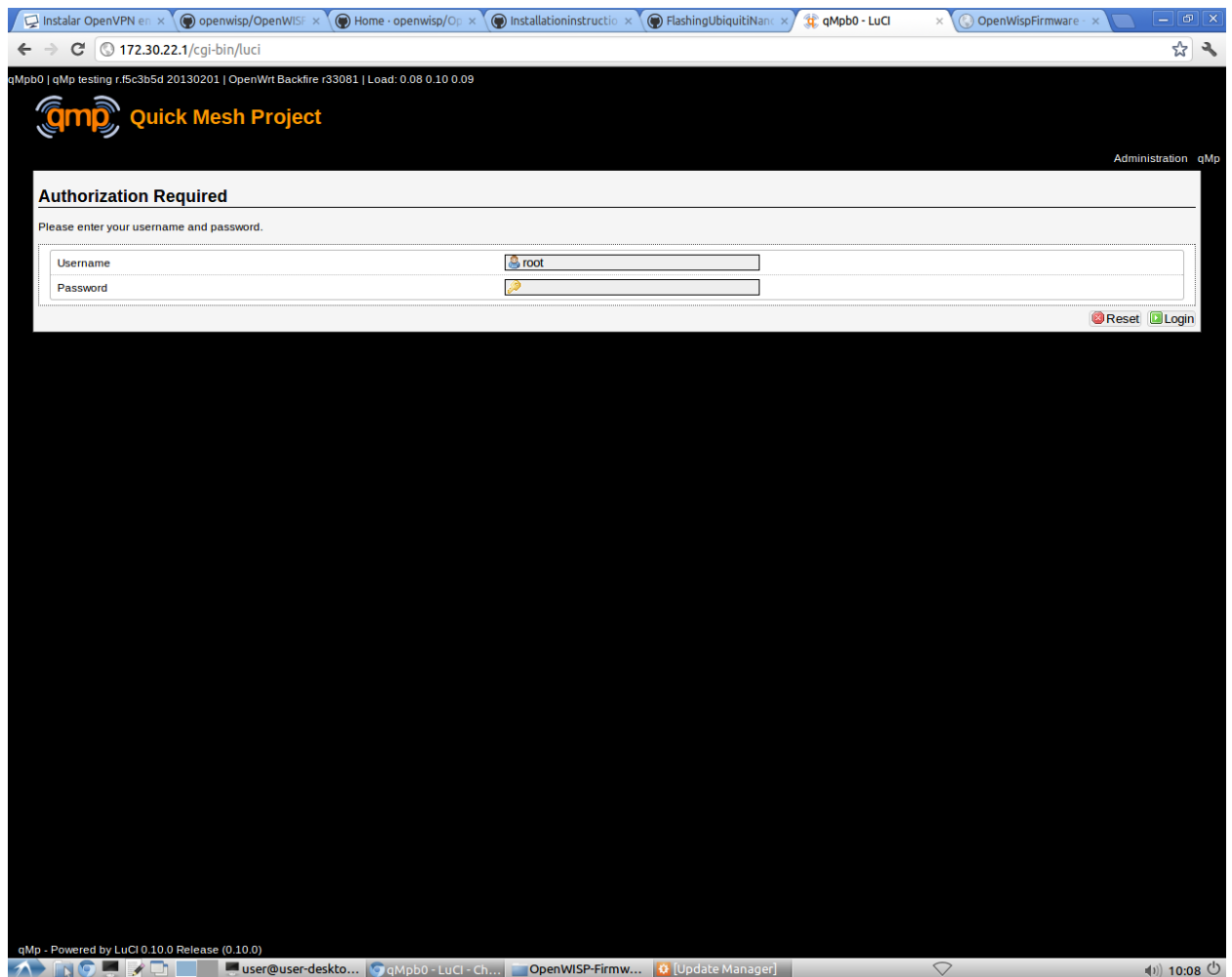
Once you have logged in, we can see the tabs: "Status", "Mesh", "Configuration", "Tools" and "About". We also have the option to access in "QMP" or "Administrator" mode, in the first we have only some basic options that we can control, but are sufficient for many cases. In the second, we have many more options than we can manage, and customize in a more detailed way.

Status tab

It displays general system status: name, hardware model, firmware and kernel versions, memory status, etc. It is a purely informational tab, but it is very useful to get a general idea.

Mesh tab

This tab displays the status of the mesh network. We can observe its current status, check what nodes belong to it, see the links established among them, see a visual graph of nodes and topology, etc.
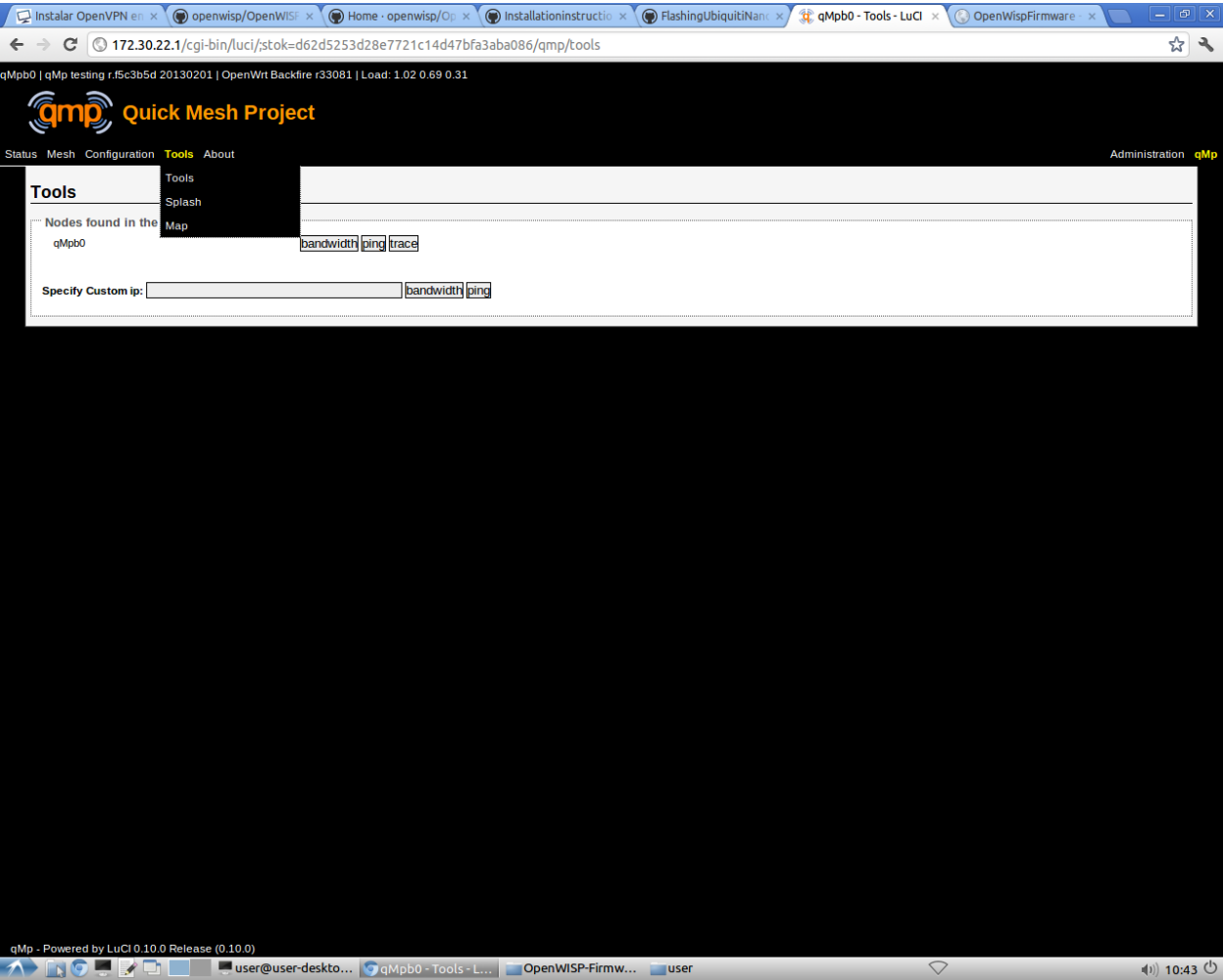
Configuration tab

This tab allows to configure the network interfaces, the Wi-Fi, the node itself and to assign the appropriate parameter in every case.
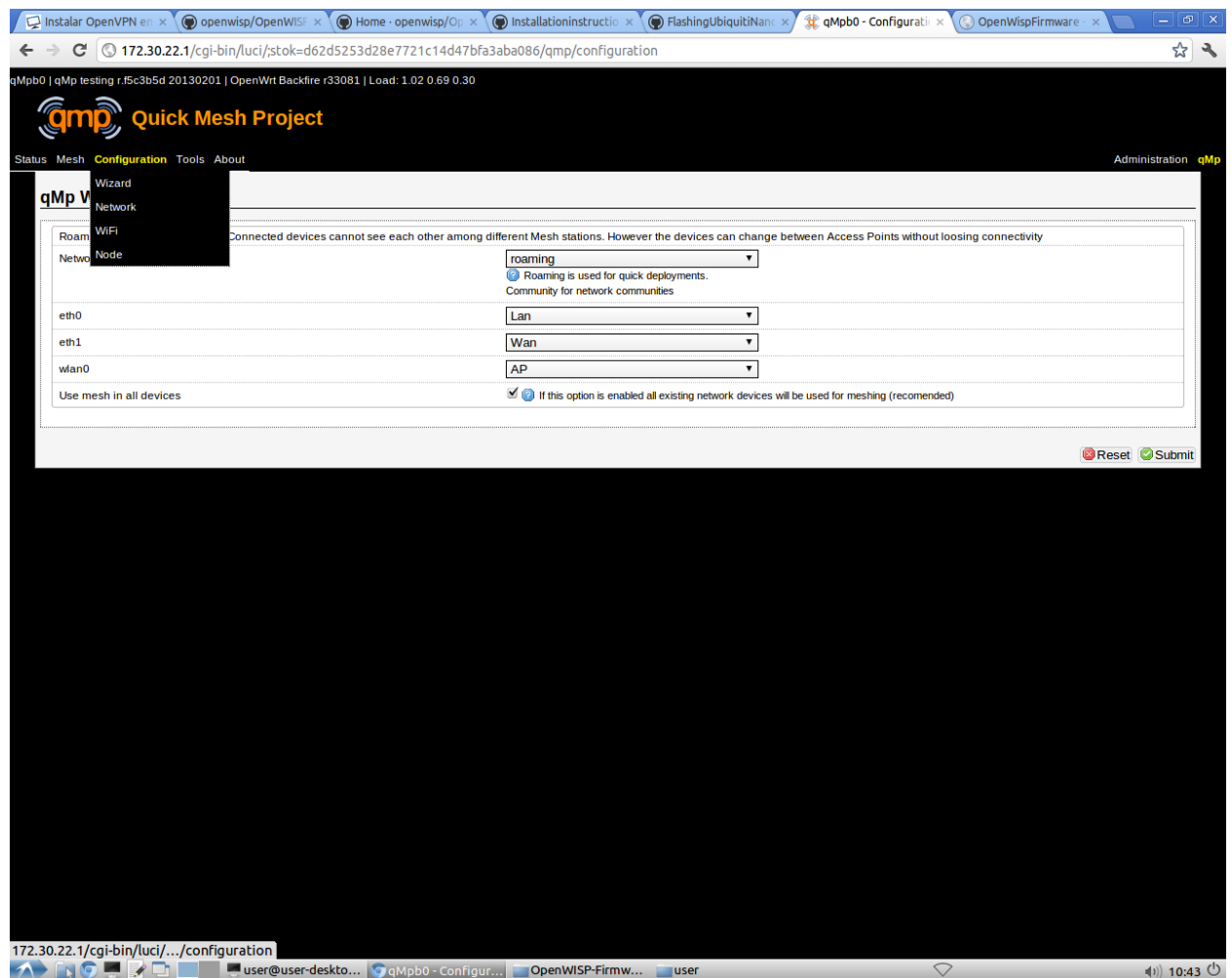
Tools tab

It allows to configure the options, the splash screen, etc.

.3 Appendix 3: QMP Presentation

Mobile Node Pilot

Fernando Gros González

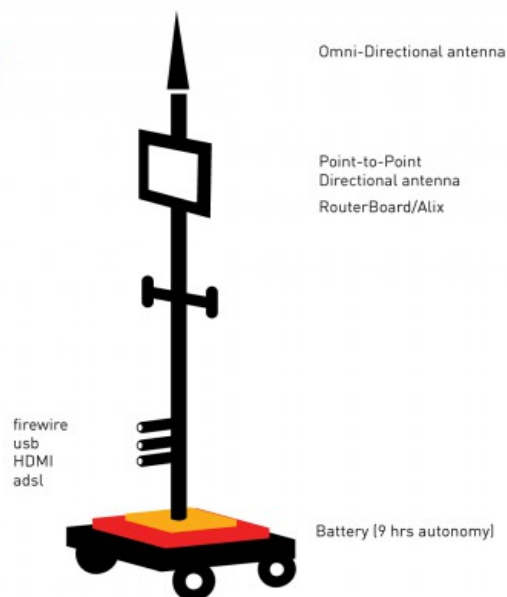04-03-2013

Outline

- Introduction
- Mesh and Manet Networks
- QMP
    - Main features
    - Quick deployments
    - Addressing
    - Modes
    - Dynamic Routing Protocol
    - BMX6
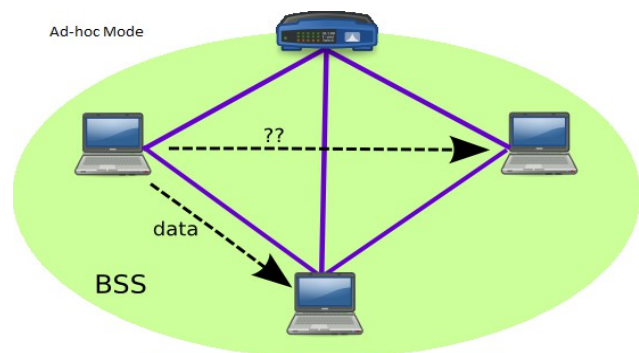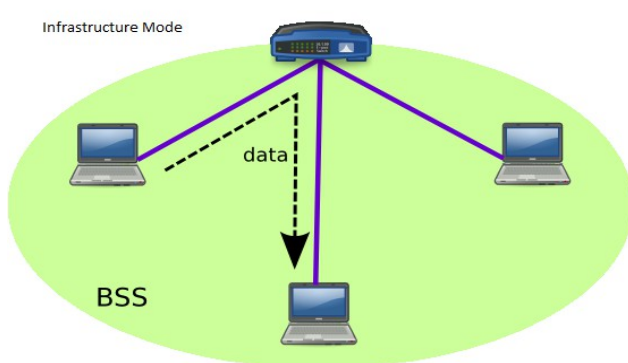- Mobile Node
- Information sources

Introduction

- The main idea within the Mobile Node Pilot is to create a free transmission workstation that can be used in the urban space and contributes to the digital mesh through other networks.

Mobile Node Prototype V_0.01

Omni-Directional antenna

Point-to-Point
Directional antenna

RouterBoard/Alix

firewire
usb
HDMI
adsl

Battery (9 hrs autonomy)

4

Mesh and Manet Networks (I)

- A mesh network is a network where all the participants are also routers. They are, normally, level 3 networks (Network layer) although they can also be link layer networks.

- Every node has to be connected, at least, to another node in the network, independently of the method (Cable, Wi-Fi, OF, etc.)

- It is not strictly necessary that this kind of networks use the Ad-hoc mode, they can use infrastructure mode too, although it is useless in many cases.

Mesh and Manet Networks (II)

- When we have a mesh network which uses a wireless system to interconnect the nodes and is built in Ad-hoc mode, we talk about MANET (Mobile Ad-hoc Network).

- So a MANET network:

  - " *is a self-configuring infrastructureless network of mobile devices connected by wireless.*" (Wikipedia)

- In order to allow this self-configuring feature, we use Dynamic Routing protocols (DRP) which are specially optimized for MANET deployments.
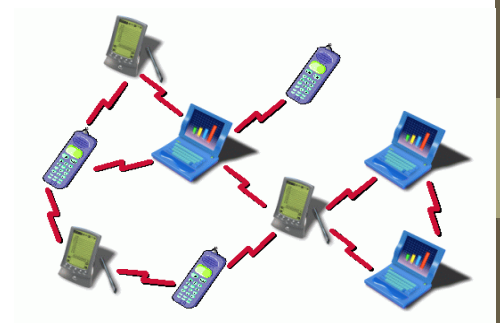
QMP

Main Features

* OpenWRT based
* 802.11a/b/g/n support
* IPv6 native
* IPv4 tunneled over IPv6
* Auto configuration system
* Web GUI to monitor and configure
* Visualization tools (maps, graphs, etc.)
* Automatic dynamic routing (zero-conf)
* BGP (Border Gateway Protocol) support (half implemented)
* Open Source

QMP

Quick Deployments

- One of the scenarios QMP has been developed for is: Quick deployments.
- Achieve the following requirements:
    - The deployment must be performed as fast as possible.
    - It must be able to be done by non-technical people.
   - It must be possible in most situations.

QMP

Addressing

- Qmp uses three different kind of IP addresses:
  - IPv6 ULA: IPv6 private range to be used internally in the mesh. These IPs are used for the communication among the nodes in the mesh network, and so they are not neither valid nor routable outside.
  - IPv6 RIPE: IPv6 public IPs range (6to6 tunneling). These are globally valid and routable.
  - IPv4: IPv4 private range to connect with the final user (4to6 tunneling). They are assigned to the final users attached to a node in the mesh, when they transmit any packet that has to travel throughout the mesh it is encapsulated in an IPv6 packet (tunneling).

QMP
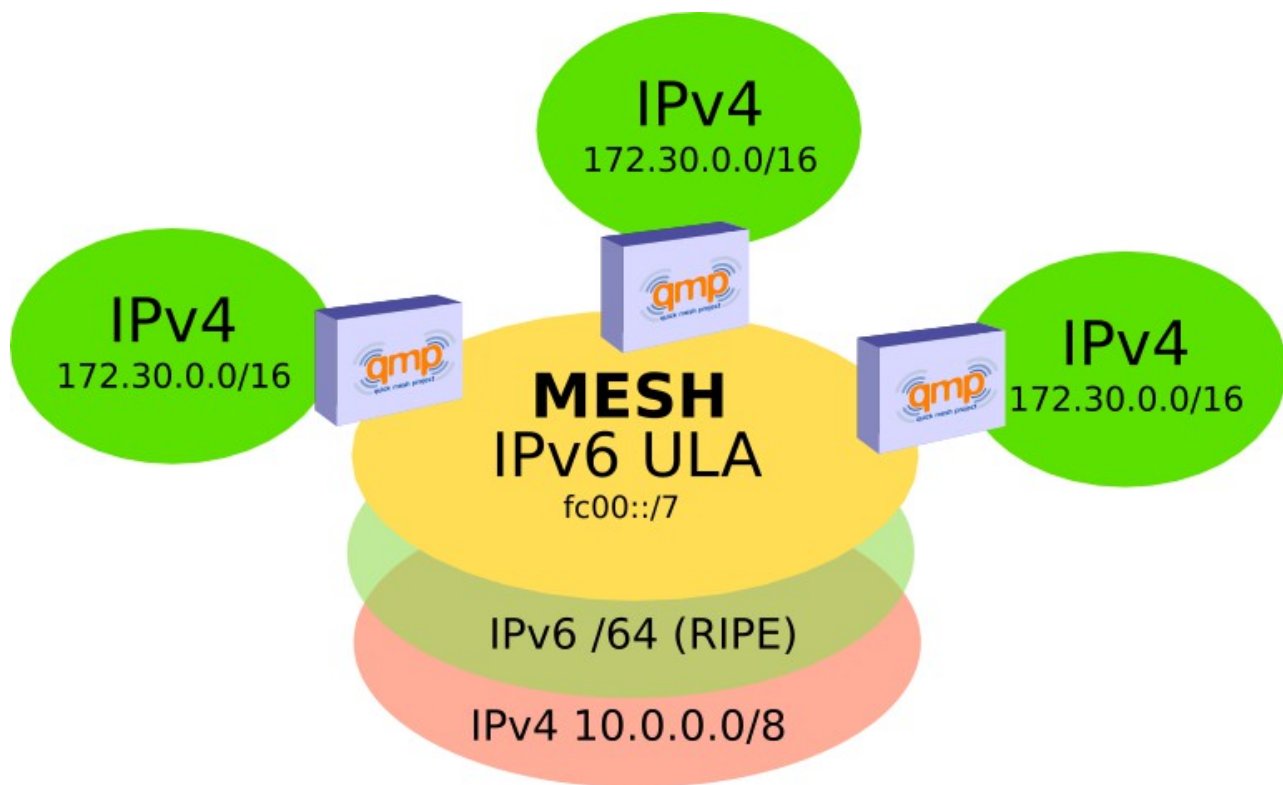
QMP

Modes

- *Roaming* for fast deployments:
  - All the access points in this mode will have the same IP and the same ESSID in order to allow users mobility, namely, they won't lose the connection although they switch from an AP to another.
  - Every AP implements a NAT and so, two users attached to different APs won't have direct vision between them.
- *Community:*
  - Every node will have a randomly assigned IPs range and will announce this range through the mesh.
  - There is not NAT, every user has direct vision with the others (1 hope away from the IPv4 network layer point of view), but mobility is not allowed (no roaming).

QMP
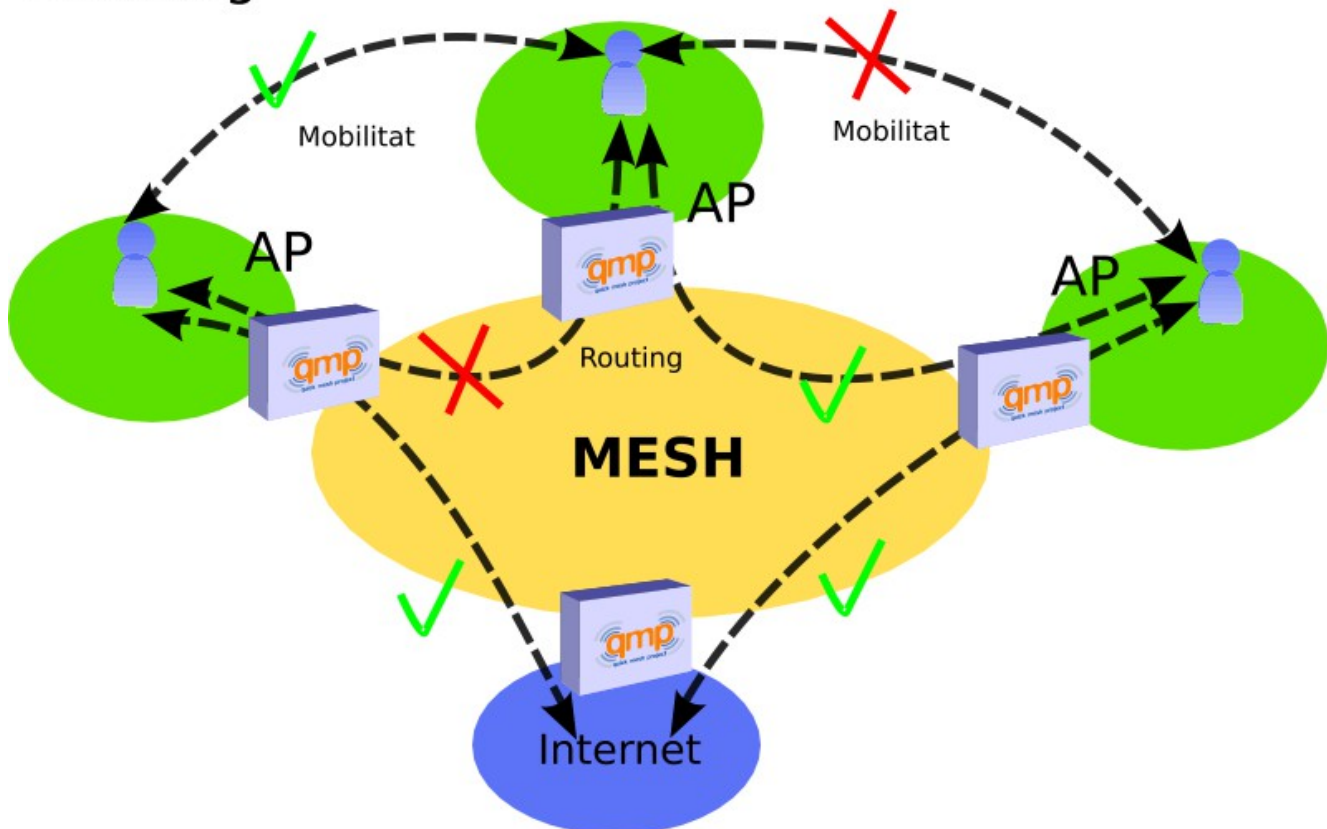
QMP

Dynamic Routing Protocol

- The QMP firmware, uses the following protocols:
  - BMX6 as the main DRP.
  - OLSR6 as a backup DRP.
  - Babel as a backup DRP but optional.

-

- All three use IPv6 ULA to talk to other nodes and are isolated at the link layer (MAC) using VLANs.

QMP

BMX6

- Pro-active: Uses UDP flooding to periodically send Originator Messages (OGM) and build a routing table.

- Destination-sequenced, Distance-vector (DSDV): Every node just knows which neighbor is better to reach another, namely, they do not need to know the entire topology, just the best paths.

- Does not use IP as node identifier, it uses global identifiers using SHA2 hashing.
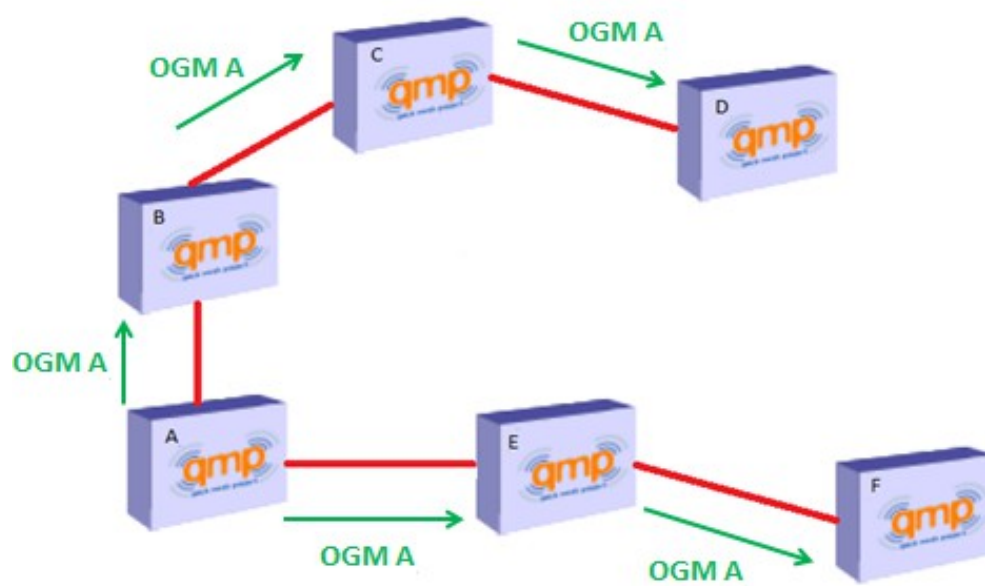
Best Path to A					
Step	B	C	E	D	F
1	A	-	A	-	-
2	A	B	A	-	E
3	A	B	A	C	E

QMP

| Frame name	Description
HELLO ADV	Hello advertisement. Used for letting neighboring nodes detect the link quality in transmit direction (from sending to receiving node).
RP ADV	Rx probe advertisement. Used for reporting about reception rate of hello messages from neighboring nodes.
OGM ADV	OGM advertisement. Used for updating periodically route and metric information over the mesh.
OGM ACK	OGM acknowledgement. Used for acknowledging the previously reception of a full OGM ADV frame.

Mobile Node

- The main goal of creating a node which has mobility is giving coverage to zones or event that are not covered with the existing fixed infrastructure. And so it has a lot of social implications, for citizens and cities:
  - Giving Wi-FI access to people in a concert or any other event (indoor or outdoors).
  - Allow people broadcast something taking place with their Smartphone (apps like bambuser) or using IP cameras.
  - Giving coverage in a protest, moving the node along with the people.
  - Etc!!
- We always need a fixed infrastructure to be able to expand the mesh by adding this node. Cities like Barcelona are creating this kind of infrastructures.

Mobile Node

BMX6 (5GHz)

Wired connection

Internet

Wi-Fi

Wi-Fi
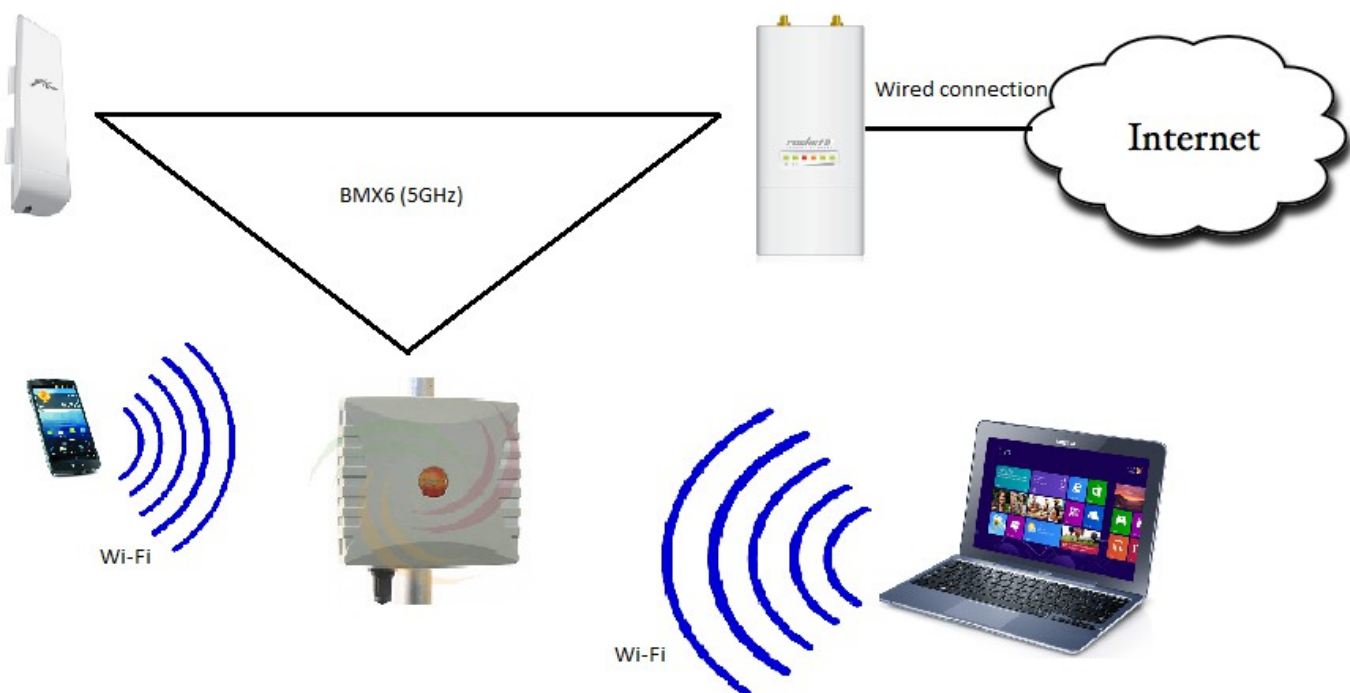
Information Sources

- Escrich, P. (2012, January). Quick deployment network using MANET. Retrieved March 4, 2013

- Escrich, P. (2012, November 17). Xarxes Mesh. Curs Guifi.net 2012. Retrieved March 5, 2013

- Mobile ad hoc network. (n.d.). In *Wikipedia*. Retrieved March 4, 2013, from http://en.wikipedia.org/wiki/Mobile_ad_hoc_network

- Starting a Community. (2013, January 11). In *Quick Mesh Project*. Retrieved March 5, 2013, from http://qmp.cat/projects/qmp/wiki/Starting

- Roaming-Community/Collaborative. (2012, June 26). In *Quick Mesh Project*. Retrieved March 5, 2013, from http://qmp.cat/projects/qmp/wiki/Roaming-Collaborative

- QMP users mailing list